



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ
ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ**

**«Ανάπτυξη Περιβάλλοντος Κεντρικής Αποστολής
Μηνυμάτων Πολιτικής σε Χρήστες Πολλαπλών Κοινωνικών
Μέσων WEB 2.0 και Συλλογής και Οπτικοποίησης της
Ανταπόκρισης»**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
των

Κουλιζάκη Ιωάννη
Πιτσιώνα Αντώνη

Επιβλέπων: Λουκής Ευριπίδης
Μόνιμος Επίκουρος Καθηγητής
Πανεπιστήμιο Αιγαίου

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή:

Λουκής Ευριπίδης
Μόνιμος Επίκουρος Καθηγητής
Πανεπιστήμιο Αιγαίου

Χαραλαμπίδης Ιωάννης
Επίκουρος Καθηγητής
Πανεπιστήμιο Αιγαίου

Μαραγκουδάκης Μανώλης
Λέκτορας
Πανεπιστήμιο Αιγαίου

ΣΑΜΟΣ, Σεπτέμβριος 2011

Ευχαριστίες

Πρώτα απ' όλα θα θέλαμε να εκφράσουμε τις θερμές μας ευχαριστίες στον επιβλέποντα καθηγητή της διπλωματικής εργασίας, κ. Ευριπίδη Λουκή, Μόνιμο Επίκουρο Καθηγητή του τμήματος Μηχανικών Πληροφοριακών & Επικοινωνιακών Συστημάτων του Πανεπιστημίου Αιγαίου για την πολύτιμη βοήθεια και καθοδήγηση κατά τη διάρκεια εκπόνησης της εργασίας.

Επίσης ευχαριστούμε τον κ. Ιωάννη Χαραλαμπίδη, Επίκουρο Καθηγητή του τμήματος Μηχανικών Πληροφοριακών & Επικοινωνιακών Συστημάτων του Πανεπιστημίου Αιγαίου και τον κ. Μανώλη Μαραγκουδάκη, Λέκτορα του τμήματος Μηχανικών Πληροφοριακών & Επικοινωνιακών Συστημάτων του Πανεπιστημίου Αιγαίου, μέλη της εξεταστικής επιτροπής, για τις πολύτιμες υποδείξεις τους, αλλά και τους υπόλοιπους καθηγητές του τμήματος Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων για τη συμβολή τους στην διεύρυνση των γνώσεών μας.

Επιπλέον, θα θέλαμε να εκφράσουμε την ευγνωμοσύνη μας στην κ. Αγγελική Ανδρουτσοπούλου, Επιστημονική Συνεργάτη και Υποψήφια Διδάκτωρ του Πανεπιστημίου Αιγαίου, για την προθυμία και τη διαθεσιμότητα που επέδειξε σε όλη τη διάρκεια εκπόνησης της εργασίας παρέχοντας χρήσιμες συμβουλές και δεδομένα.

Τέλος, θα θέλαμε να ευχαριστήσουμε τις οικογένειές μας και τους φίλους μας, για την συμπαράσταση και ψυχολογική υποστήριξη που μας προσφέρουν εδώ και χρόνια και για τη συμβολή τους στη διεκπεραίωση των σπουδών μας.

Πίνακας Περιεχομένων

Πίνακας Περιεχομένων.....	3
Πίνακας Εικόνων.....	5
Περίληψη.....	7
Abstract.....	8
Κεφάλαιο 1 ^ο - Εισαγωγή.....	9
Κεφάλαιο 2 ^ο - Web 2.0 & Social Media.....	13
2.1. - Νέες τάσεις στο Web - Έμφαση στον άνθρωπο.....	13
2.2. - Το Web 2.0 ως ο Νέος Παγκόσμιος Ιστός.....	18
2.3. - Web 1.0 vs Web 2.0: βασικά σημεία διαφοροποίησης.....	20
2.4. - Web 3.0 ή Σημασιολογικός Ιστός (Semantic Web).....	23
2.5. - Επισκόπηση στις νέες τεχνολογίες του Web 2.0.....	27
2.5.1. - Ιστολόγια (Blogs).....	27
2.5.2. - WIKIS.....	29
2.5.3. - Κοινωνική Σήμανση (Social Tagging).....	31
2.5.4. - Ιστοχώροι Κοινωνικής Δικτύωσης (SNS).....	32
2.5.5. - Podcasts και Videocasts.....	33
2.5.6. - Εικονικοί Κόσμοι (Virtual Worlds).....	35
2.5.7. - Photo Sharing.....	36
2.5.8. - Web Conferencing & Electronic meetings.....	36
2.6. - Ιδιωτικότητα στο Web 2.0 (Security Issues).....	37
2.7. - Οι δημοφιλέστερες Web 2.0 Ιστοσελίδες Κοινωνικής Δικτύωσης.....	39
2.7.1. - Facebook.....	39
2.7.2. - YouTube.....	39
2.7.3. - LinkedIn.....	40
2.7.4. - Twitter.....	40
2.7.5. - Picasa.....	40
2.7.6. - Blogger.....	41
Κεφάλαιο 3 ^ο - Βασικοί στόχοι του Padgets.....	42
3.1. - Βασική ιδέα.....	42
3.2. - Όφελος στην πολιτική διαδικασία.....	45
3.3. - Καινοτομία όσον αφορά τις υφιστάμενες πρακτικές.....	47
3.3.1. - Καινοτομία στο τομέα της ηλεκτρονικής διαβούλευσης.....	47
3.3.2. - Καινοτομία στα συστήματα υποστήριξης αποφάσεων.....	48
3.4. - Ρόλοι που αλληλεπιδρούν μέσω του Padgets.....	50
3.4.1. - Φορέας χάραξης πολιτικής.....	50
3.4.2. - Πολίτης.....	53
3.5. - Μελέτη περίπτωσης - debate για τη μετανάστευση.....	55
3.5.1. - Με την προοπτική του policy maker.....	55
3.5.2. - Με την προοπτική του πολίτη.....	60
3.6. - Use Case ανάλυση.....	63
Κεφάλαιο 4 ^ο - Ανάλυση των API των social media.....	81
4.1. - Blogger.....	82
4.2. - Picasa.....	88
4.3. - YouTube.....	101
4.4. - Twitter.....	112
4.5. - Facebook.....	118

4.6. - LinkedIn.....	138
4.7. - Google Charts API.....	142
Κεφάλαιο 5 ^ο - Μεθοδολογία.....	154
5.1. - Μελέτη Concept Padgets.....	154
5.2. - Μελέτη Social Media APIs.....	155
5.3. - Μελέτη Visualization APIs.....	158
5.4. - Ανάλυση απαιτήσεων εφαρμογής.....	158
5.4.1. - Λειτουργικές απαιτήσεις.....	158
5.4.2. - Μη Λειτουργικές απαιτήσεις.....	161
5.5 - Μελέτη τεχνολογιών - Τεχνολογίες που χρησιμοποιήθηκαν.....	163
Κεφάλαιο 6 ^ο - Υλοποίηση Εφαρμογής SocialCampaign	166
6.1. - Publishing.....	166
6.1.1. - Facebook.....	166
6.1.2. - Twitter.....	178
6.1.3. - YouTube.....	185
6.1.4. - LinkedIn.....	193
6.1.5. - Blogger.....	201
6.1.6. - Picasa.....	209
6.2. - Retrieving.....	217
6.2.1. - Facebook.....	217
6.2.2. - Twitter.....	236
6.2.3. - YouTube.....	244
6.2.4. - LinkedIn.....	251
6.2.5. - Blogger.....	258
6.2.6. - Picasa.....	263
6.3. - Visualization.....	268
Κεφάλαιο 7 ^ο - Εφαρμογή SocialCampaign	290
7.1. - Publishing.....	290
7.2. - Visualization.....	295
Κεφάλαιο 8 ^ο - Συμπεράσματα.....	301
Βιβλιογραφία.....	304
Παράρτημα.....	308
I. - Λεξικό βασικών όρων του Padgets.....	308
II. - Ανάλυση των διασημότερων Social Media.....	310

Πίνακας Εικόνων και Πινάκων

1.1. - Διαγραμματική απεικόνιση της εφαρμογής.....	10
1.2. - Ροή του Padgets.....	11
1.3. - Λογότυπο της εφαρμογής.....	12
2.1. - Συλλογική Ευφυΐα ως βασική φιλοσοφία του Νέου Ιστού.....	13
2.2. - Λογότυπο της Κοινοπραξίας για τον Παγκόσμιο Ιστό.....	17
2.3. - Χρηστό-κεντρική θεώρηση του Web 2.0.....	18
2.4. - Σχηματική αντιπαραβολή του Web 1.0 και του Web 2.0.....	21
2.5. - Ιστός 1.0 VS Ιστού 2.0 κατά τον Tim O'Reilly.....	21
2.6. - Αλλαγές της μετάβασης του Web 1.0 στο Web 2.0.....	22
2.7. - Η εξέλιξη του διαδικτύου.....	25
2.8. - Συγκριτική Αξιολόγηση των τριών εκδόσεων του διαδικτύου.....	26
2.9. - Συσχετισμός Κοινωνικής Σήμανσης με τα αυτοκόλλητα post it.....	31
3.1. - Οι συνιστώσες του Padgets.....	44
3.2. - Αξία που προκύπτει από το padgets για όλους τους εμπλεκόμενους.....	45
3.3. - Διαφορετικές προοπτικές του ρόλου του policy maker.....	51
3.4. - Policy makers - αρμοδιότητές τους.....	53
3.5. - Κατηγοριοποίηση του ρόλου του πολίτη.....	54
3.6. - Οπτική του πολιτικού για τη διαβούλευση μέσω του padget.....	55
3.7. - Ροή της άποψης σχετικά με τη μετανάστευση που διαμορφώνεται δυναμικά.....	59
3.8. - Συνολική εικόνα της εκστρατείας από την οπτική του policy maker.....	59
3.9. - Εμφάνιση καθαρών αριθμητικών στοιχείων.....	60
3.10. - Οπτική της διαβούλευσης από τη μεριά του πολίτη.....	61
3.11. - Ροή της αλληλεπίδρασης.....	63
3.12. - Πίνακας χρηστών συστήματος.....	64
3.13. - Use Case διάγραμμα του End User.....	66
3.14. - Use Case διάγραμμα του Padgets Initiator / Consultant.....	68
3.15. - Use Case διάγραμμα του System Administrator.....	71
3.16. - Use Case διάγραμμα του Padgets Initiator / Consultant (Web Application).....	73
3.17. - Use Case διάγραμμα του Padgets Initiator / Consultant (Decision Model).....	75
3.18. - Περιπτώσεις χρήσεις.....	77
4.1. - YouTube Interface.....	101
4.2. - Twitter Interface.....	112
4.3. - Facebook Interface.....	118
4.4. - LinkedIn Interface.....	138
4.5. - Πίνακας με τα βασικότερα Google Charts.....	142
4.6. - query σε data source.....	147
4.7. - query response.....	148
4.8. - Τύπος δεδομένων ανά γράφημα.....	149
5.1. - Φιλοσοφία Padgets	154
5.2. - Προσομοίωση μιας καμπάνιας μέσω του Padgets.....	155
6.1. - Ο γράφος του Facebook.....	166
6.2. - Chart το οποίο παράγεται - average age.....	273
6.3. - Chart το οποίο παράγεται - likes per type.....	276
6.4. - Chart το οποίο παράγεται - αριθμητικά.....	280
6.5. - Chart το οποίο παράγεται - word cloud.....	283
6.6. - Chart το οποίο παράγεται - geochart.....	287

6.7. - Chart το οποίο παράγεται - map chart.....	289
7.1. - Signing in.....	290
7.2. - Δημιουργία καμπάνιας.....	291
7.3. - Επιλογή των SM.....	291
7.4. - Αυθεντικοποίηση.....	292
7.5. - Πεδία κειμένου.....	293
7.6. - Πολυμέσα.....	294
7.7. - Επιλογή καμπάνιας για παρουσίαση.....	295
7.8. - General numbers.....	296
7.9. - Numbers per SM.....	296
7.10. - Average age.....	297
7.11. - Map with tags.....	297
7.12. - Map per country.....	298
7.13. - General gender.....	298
7.14. - #Comments per SM.....	299
7.15. - #Likes per type.....	299
7.16. - Comments Picasa.....	300
7.17. - Comments on video.....	300

Περίληψη

Η παρούσα εργασία χωρίζεται σε δύο διακριτές ενότητες και εντάσσεται στην προσπάθεια διεύρυνσης της συμμετοχικής δημοκρατίας.

Στην πρώτη ενότητα πραγματοποιείται ενδελεχής διερεύνηση της υπάρχουσας κατάστασης του web 2.0 ιστού και καταγράφονται τα δημοφιλέστερα μέσα κοινωνικής δικτύωσης. Έπειτα επιλέγονται έξι από αυτά (Facebook, Twitter, LinkedIn, YouTube, Picasa και Blogger) και μελετώνται οι δυνατότητες που προσφέρουν τα API (Application Programming Interface) τους, καθώς επίσης και οι δυνατότητες που προσφέρει το Google Charts API.

Στη δεύτερη ενότητα της εργασίας χρησιμοποιώντας τα social media που αναφέρθηκαν παραπάνω πραγματοποιείται ανάλυση και υλοποίηση μίας εφαρμογής, η οποία μέσω μίας κεντρικής διαδικασίας θα προσφέρει τις παρακάτω δυνατότητες:

1. Δημοσίευση ποικίλου περιεχομένου σε διάφορα μέσα κοινωνικής δικτύωσης κατ' επιλογή του χρήστη
2. Συλλογή της συνολικής ανταπόκρισης του δημοσιευμένου περιεχομένου από όλα τα προεπιλεγμένα μέσα κοινωνικής δικτύωσης
3. Παρουσίαση της ανταπόκρισης με χρήση βασικών τεχνικών οπτικοποίησης

Σκοπός της εφαρμογής είναι να δοθεί η δυνατότητα σε χρήστες που είναι σε θέση να διαμορφώνουν πολιτικές να μπορούν γρήγορα και εύκολα μέσω μόνο μίας κεντρικής διαδικασίας να γνωστοποιούν τις προωθούμενες πολιτικές τους απευθείας στους πολίτες.

ΚΟΥΛΙΖΑΚΗΣ ΙΩΑΝΝΗΣ

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

ΠΙΤΣΙΩΝΑΣ ΑΝΤΩΝΗΣ

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

© 2011

Abstract

This paper is divided into two distinct sections and is part of the attempt to extend participatory democracy.

In the first section a thorough research carried out on the current state of web 2.0 web and recorded the most popular social media. Then selected six of them (Facebook, Twitter, LinkedIn, YouTube, Picasa, and Blogger) and studied the potential of the API (Application Programming Interface), and also the potential of the Google Charts API.

In the second section using the social media mentioned above shall be analyzed and implemented an application which through a centralized procedure offers the following features:

1. Post varied content across different social media at the discretion of the user
2. Collection of the overall response of the published content from all the default social media
3. Presentation of the response using basic visualization techniques

The purpose of the application is to enable users who are able to formulate policies to publish quickly and easily via a single centralized disclose their policies directly to citizens.

KOULIZAKIS IOANNIS

Department of Information and Communication Systems Engineering

UNIVERSITY OF THE AEGEAN

PITSIONAS ANTONIS

Department of Information and Communication Systems Engineering

UNIVERSITY OF THE AEGEAN

© 2011

Κεφάλαιο 1^ο - Εισαγωγή

Καθώς η web 2.0 φιλοσοφία αγκαλιάστηκε από τους χρήστες του παγκόσμιου ιστού τα τελευταία χρόνια με αποτέλεσμα την εκρηκτική αύξηση των χρηστών παγκοσμίως, παρατηρήθηκαν δύο σημαντικές μεταβολές στα μέχρι πρότινος δεδομένα: η εμφάνιση των μέσων κοινωνικής δικτύωσης και η προσπάθεια των κυβερνήσεων να αναπτύξουν την ηλεκτρονική συμμετοχή στα κοινά (e-participation).

Αρχικά, έχουμε την εμφάνιση των πρώτων μέσων κοινωνικής δικτύωσης τα οποία έγιναν με μεγάλο ενθουσιασμό αποδεκτά από τους χρήστες, δημιουργώντας έτσι ένα νέο κοινωνικό παγκόσμιο ιστό. Πλέον υπάρχουν αμέτρητοι ιστόχωροι κοινωνικής αλληλεπίδρασης, οι οποίοι προσφέρουν κάθε είδους κοινωνικής μορφής αλληλεπίδραση μεταξύ των χρηστών, από ανταλλαγή απλών μηνυμάτων δημόσια, στον διαμοιρασμό ποικίλου είδους περιεχομένου (φωτογραφίες, βίντεο), μέχρι τη δημιουργία υπο-δικτύων κοινού ενδιαφέροντος. Έτσι περάσαμε στην υιοθέτηση του διαδικτύου ως βασικό μέσο κοινωνικοποίησης και ψυχαγωγίας, δημιουργώντας νέες διαστάσεις στην επικοινωνία μεταξύ των χρηστών:

- ✓ Ελεύθερη δημοσιοποίηση δεξιοτήτων, έκφρασης δημιουργικότητας, ευκαιρίες ανάδειξης
- ✓ Εκμετάλλευση της συλλογικής γνώσης κι εμπειρίας
- ✓ Δημοκρατικότητα, αίσθηση ένταξης σε κοινότητα, συνεργασία και συνεισφορά
- ✓ Αμφίδρομη επικοινωνία χρήστη-επιχειρήσεων, ενίσχυση της διαπραγματευτικής δύναμης των χρηστών στις εμπορικές συναλλαγές
- ✓ Νέοι τρόποι άσκησης Κρατικής Πολιτικής

Παρακάτω πραγματοποιείται μια ενδεικτική κατηγοριοποίηση των Ιστοχώρων Κοινωνικής Δικτύωσης σύμφωνα με τη φιλοσοφία των υπηρεσιών που παρέχουν:

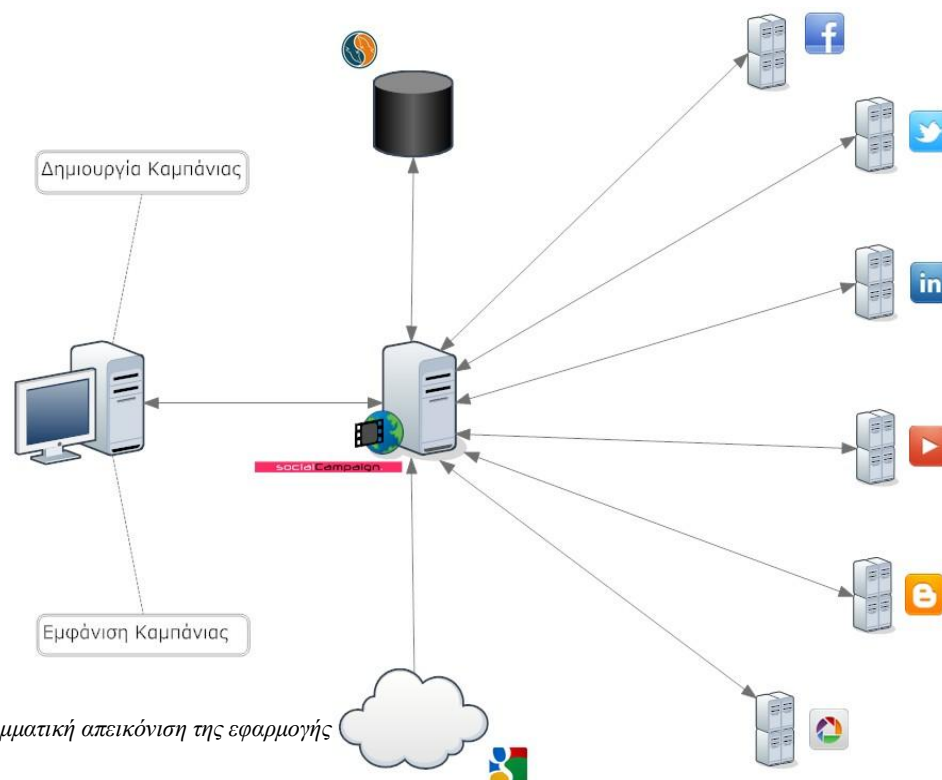
- ✓ Δίκτυα με έμφαση στις προσωπικές σχέσεις (πχ Facebook)
- ✓ Δίκτυα επαγγελματικών κοινοτήτων (πχ LinkedIn)
- ✓ Δίκτυα οπτικοακουστικού υλικού (πχ YouTube)
- ✓ Ιστολόγια (Blogs) (πχ Blogger)
- ✓ Ιστολόγια προσαρμοσμένα σε κινητές συσκευές (micro-blogs) (πχ Twitter)
- ✓ Δίκτυα συλλογής κι ανταλλαγής εικόνων (πχ Picasa)

Οι κυβερνήσεις αντιλαμβανόμενες τις νέες δυνατότητες του νέου ιστού προσπαθούν να αυξήσουν την ηλεκτρονική συμμετοχή των πολιτών ξοδεύοντας μεγάλα ποσά για την ανάπτυξη ιστοτόπων με σκοπό τη διαβούλευση των πολιτών σε θέματα πολιτικής. Η φιλοσοφία όμως των ιστοτόπων αυτών συνήθως ακολουθεί τη φιλοσοφία του web 1.0 ιστού, με αποτέλεσμα συνήθως η χρήση τους να μην επιφέρει τα επιθυμητά αποτελέσματα.

Η αλματώδης όμως ανάπτυξη των μέσων κοινωνικής δικτύωσης προσφέρει τη δυνατότητα η ηλεκτρονική συμμετοχή να επιτευχθεί αποτελεσματικά μέσα από το feedback των χρηστών που μπορεί να προκύψει από τις δραστηριότητές τους στα διάφορα social media.

Αρχικά το αντικείμενο της εργασίας θα είναι η ανάλυση της υπάρχουσας κατάστασης στον ιστό σχετικά με τα μέσα κοινωνικής δικτύωσης. Έπειτα αφού μελετηθούν οι δυνατότητες που προσφέρονται μέσα από τα API τους θα προχωρήσουμε στο δεύτερο κομμάτι της εργασίας όπου χρησιμοποιώντας κάποια από τα πιο δημοφιλή μέσα κοινωνικής δικτύωσης πραγματοποιείται ανάλυση και υλοποίηση μίας εφαρμογής, η οποία μέσω μίας κεντρικής διαδικασίας θα προσφέρει τις παρακάτω δυνατότητες:

1. Δημοσίευση ποικίλου περιεχομένου σε διάφορα μέσα κοινωνικής δικτύωσης κατ' επιλογή του χρήστη
2. Συλλογή της συνολικής ανταπόκρισης του δημοσιευμένου περιεχομένου από όλα τα προεπιλεγμένα μέσα κοινωνικής δικτύωσης
3. Παρουσίαση της ανταπόκρισης με χρήση βασικών τεχνικών οπτικοποίησης



1.1. - Διαγραμματική απεικόνιση της εφαρμογής

Η εφαρμογή που θα υλοποιηθεί θα προσπαθήσει να εισάγει μια νέα προσέγγιση στην προώθηση πολιτικής εφαρμόζοντας τη φιλοσοφία του Web 2.0, πηγαίνοντας δηλαδή ακριβώς εκεί όπου βρίσκονται οι χρήστες στο διαδίκτυο.

Η διάδοση των πολιτικών μηνυμάτων θα γίνεται μέσω πολλαπλών μέσων κοινωνικής δικτύωσης ταυτόχρονα, χρησιμοποιώντας μια ενιαία, ολοκληρωμένη διεπαφή επικοινωνίας με σκοπό τη λήψη της ανταπόκρισης των χρηστών.



1.2. - Ποή του Padgets

Σκοπός λοιπόν της εφαρμογής είναι να δοθεί η δυνατότητα σε χρήστες που είναι σε θέση να διαμορφώνουν πολιτικές να μπορούν γρήγορα και εύκολα μέσω μόνο μίας κεντρικής διαδικασίας να γνωστοποιούν τις προωθούμενες πολιτικές τους έτσι ώστε:

- ✓ Αρχικά να επιτευχθεί διεύρυνση της συμμετοχικής δημοκρατίας μέσω του web 2.0 καθώς θα ανοίξει ένας επαναλαμβανόμενος κύκλος προώθησης - ανταπόκρισης
- ✓ Να δώσει στο χρήστη τη δυνατότητα να παρακολουθήσει συγκεντρωμένη τη συνολική ανταπόκριση της προωθούμενης πολιτικής του
- ✓ Τελικά να μπορέσει να υποβοηθήσει το χρήστη στη λήψη των αποφάσεων που θα καθορίσουν την πολιτική που θα ακολουθήσει

Παρακάτω ακολουθεί μια σύντομη περιγραφή των κεφαλαίων της εργασίας:

Στο πρώτο κεφάλαιο γίνεται μια εισαγωγή στο αντικείμενο και το σκοπό της εργασίας.

Στο δεύτερο κεφάλαιο παρουσιάζονται οι δεδομένες συνθήκες που επικρατούν στον παγκόσμιο ιστό και οι δυνατότητες που προσφέρονται μέσα από την αξιοποίηση των διάφορων μέσων κοινωνικής δικτύωσης.

Στο τρίτο κεφάλαιο παρουσιάζουμε της φιλοσοφία και τις βασικές αρχές του έργου Radgets, το οποίο χρησιμοποιήσαμε ως πρότυπο για την ολοκλήρωση της παρούσης εργασίας.

Στο τέταρτο κεφάλαιο προχωρούμε σε αναλυτική περιγραφή των δυνατοτήτων που προσφέρουν τα API των έξι βασικών social media που θα χρησιμοποιήσουμε στην εφαρμογή καθώς και επίσης των δυνατοτήτων αναπαράστασης δεδομένων γραφικά του Charts API της Google.

Στο πέμπτο αναφέρεται η μεθοδολογία που ακολουθήθηκε για την εκπόνηση της διπλωματικής εργασίας.

Στο έκτο κεφάλαιο παρουσιάζεται αναλυτικά η υλοποίηση της εφαρμογής χωρισμένη σε τρεις ενότητες, μία για τη διαδικασία του publishing, μία για τη διαδικασία του retrieving και μία για τη διαδικασία του visualization.

Στο έβδομο κεφάλαιο πραγματοποιείται παρουσίαση της εφαρμογής που υλοποιήθηκε με οθόνες εκτέλεσης από τις βασικές λειτουργίες της, έτσι ώστε να δοθεί μία γεύση από το σχεδιασμό της διεπαφής και να εκτιμηθούν οι δυνατότητές της.

Η εργασία ολοκληρώνεται με το όγδοο κεφάλαιο, όπου παραθέτονται τα συμπεράσματα που προέκυψαν από την μελέτη και την ανάπτυξη της εφαρμογής που υλοποιήθηκε.

The logo for 'socialCampaign.' is displayed in white text on a red rectangular background. The word 'social' is in a lowercase, sans-serif font, and 'Campaign.' is in a larger, bold, uppercase, sans-serif font.

1.3. - Λογότυπο της εφαρμογής

Κεφάλαιο 2^ο - Web 2.0 & Social Media

2.1. - Νέες τάσεις στο Web - Έμφαση στον άνθρωπο

Η ανάπτυξη του Παγκόσμιου Ιστού, όπως είναι φυσικό, συνέπεσε με την καθιέρωση νέων υπηρεσιών που καθιστούν το Διαδίκτυο μια μεγάλη πλατφόρμα επικοινωνίας και διαδραστικότητας. Οι νέες αυτές υπηρεσίες δεν αποτελούν απλά πακέτου λογισμικού, αλλά έχουν χαρακτήρα χρηστό-κεντρικό. Οι χρήστες δηλαδή συνεισφέρουν ενεργά στην ανάπτυξη των εφαρμογών, ενώ παράλληλα δίνεται ιδιαίτερη σημασία στην δυνατότητα δημιουργίας δικτυακών κοινοτήτων. Σύμφωνα με τον Tim O'Reilly, οι περισσότερο επιτυχημένες από αυτές τις εφαρμογές, είναι εκείνες που περιγράφονται ως «σημασιολογικά εκπαιδευόμενα συστήματα» και μοχλεύουν τελικώς τα μέχρι πρότινος υποκρυπτόμενα και ανεκδήλωτα μετα-δεδομένα (metadata). Αυτό ακριβώς το επόμενο βήμα στην θεώρηση που έχουμε για το Διαδίκτυο, ονομάζεται κατά σύμβαση «Νέος Παγκόσμιος Ιστός» ή ίσως πιο δόκιμα Web 2.0.



2.1. - Εκμετάλλευση της Συλλογικής Ευφυΐας ως βασική φιλοσοφία του Νέου Παγκόσμιου Ιστού

Στην εποχή του Νέου Παγκόσμιου Ιστού λοιπόν, όπως διαμορφώνεται και επεκτείνεται με την ενεργή συμμετοχή των χρηστών, η «εκπαίδευση» των συστημάτων βάση της δραστηριότητας των χρηστών θα πρέπει να θεωρείται αυτονόητη. Οι προδιαγραφές και γενικότερα η οργάνωση και λειτουργία των Web 2.0 εφαρμογών ακολουθούν μινιμαλιστική φιλοσοφία και ενισχύουν τη διαλειτουργικότητα, την επικοινωνία και την καινοτομία. Βάσει των παραπάνω αρχών, αναπτύσσεται η κατεύθυνση προς τον Σημασιολογικό Ιστό (Semantic Web),

έναν Ιστό φιλικό σε μετατροπές, ευκολονόητο, με έξυπνες μηχανές αναζήτησης που προσαρμόζονται στις ανάγκες του χρήστη και δυνατότητες ανάπτυξης «ανθρώπινης» λογικής. Ο Σημαιολογικός Ιστός αποκαλείται από πολλούς Web 3.0 και ήδη από τις αρχές του 2010 θεωρείται ότι βρισκόμαστε στο μεταίχμιο μετάβασης από την 2.0 στην 3.0 έκδοση.

Γίνεται λοιπόν σαφές ότι ένας Παγκόσμιος Ιστός που βασίζεται στη συνεργασία και την επικοινωνία, επιφέρει επαναστατικές αλλαγές στη μέχρι πρότινος στατική δομή του Διαδικτύου. Το Διαδίκτυο θαρραλέα και δυναμικά εισέβαλε από το 1994 κι έπειτα, σε τομείς που αρχικά φάνταζε να μην έχει θέση. Σε αυτή τη βάση, θα μπορούσε κανείς να ισχυριστεί ότι το Web 2.0 αποτελεί την εξέλιξη του διαδικτύου που έχει ως κινητήριο δύναμη τους ίδιους τους χρήστες και διαμορφώνεται από τις ανάγκες τους - ανάγκες για αρτιότερη, ευκολότερη και πιο αποτελεσματική επικοινωνία. Ίσως μάλιστα είναι η πρώτη φορά που οι χρήστες καθορίζουν τις εξελίξεις σε τόσο μεγάλο βαθμό, γεγονός που αποδεικνύεται και από την επιλογή του περιοδικού Time να ανακηρύξει σημαντικότερο πρόσωπο της χρονιάς 2006 τους ίδιους τους χρήστες.

Αυτό που ουσιαστικά διαφοροποιεί το Web 2.0 λοιπόν, από την πρότερη στατική δομή του Παγκόσμιου Ιστού είναι η ικανότητα του να συμβάλλει στη βελτίωση της καθημερινής ζωής του μέσου ανθρώπου, οπουδήποτε κι αν βρίσκεται, έχοντας ίσες ευκαιρίες πρόσβασης στην επικοινωνία και στη γνώση- άμεσα, γρήγορα και αποτελεσματικά. Ο Bill Gillis (από το “Center to Bridge the Digital Divide” : Κέντρο για την ισοστάθμιση της πρόσβασης σε ψηφιακές πληροφορίες και πόρους) αναφέρεται στο Νέο Ιστό ως *«εργαλείο που καθιστά δυνατή την καινοτομία, την επικοινωνία και τον μετασχηματισμό πληροφοριών»*. Έτσι, η Τεχνολογία Πληροφοριών κι Επικοινωνιών αποκτά νέες διαστάσεις ως Τεχνολογία Καινοτομιών και Μετασχηματισμού δίνοντας στον τελικό χρήστη ρόλο ενεργό και κυρίαρχο.

Σε μια προσπάθεια καταγραφής των αλλαγών που σηματοδότησε η έλευση του Web 2.0, παραθέτουμε στη συνέχεια επιγραμματικά στοιχεία που αφορούν κυρίως στους τελικούς αποδέκτες της τεχνολογίας του Νέου Παγκόσμιου Ιστού, δηλαδή τους ίδιους τους χρήστες:

- **Εφαρμογές προσαρμοσμένες στα συμφέροντα – ανάγκες των χρηστών:** Η εμπειρία χρήσης του διαδικτύου γίνεται άμεση, ευκολότερη και πιο αποδοτική μέσω μιας σειράς τεχνολογικών απλοποιήσεων. Μιλάμε πλέον για μια νέα γενιά «ελαφριάς» τεχνολογίας σε πρωτόκολλα (REST), open source εφαρμογές σχεδιασμένες πολλές φορές με τη συμβολή των ίδιων των χρηστών και απλό προγραμματιστικό και λειτουργικό σχεδιασμό (π.χ. αντικατάσταση των C# ή Java από γλώσσα PHP, εφαρμογές σε τεχνολογία Ajax). Επιπλέον παρέχεται δυνατότητα παραμετροποίησης ιστοσελίδων σύμφωνα με τις προτιμήσεις των ίδιων των χρηστών (π.χ. Netvibes.com), καθώς και ευκολότερη αναζήτηση πληροφορίας μέσω tagging. Τέλος, πολλές

παραδοσιακές εφαρμογές που μέχρι πρότινος ήταν διαθέσιμες μόνο κατόπιν αγοράς (λειτουργικά συστήματα, προγράμματα ηλεκτρονικού ταχυδρομείου, Office κλπ) έχουν πλέον αντικατασταθεί με αντίστοιχες εφαρμογές ανοιχτού κώδικα (Linux, gmail, Google docs κ.λπ.).

- **Νέες διαστάσεις στην επικοινωνία μεταξύ των χρηστών, υιοθέτηση του διαδικτύου ως βασικό μέσο κοινωνικοποίησης και ψυχαγωγίας:** Μια νέα γενιά διαδικτυακών εφαρμογών έχει καταστήσει την επικοινωνία των χρηστών άμεση, προσβάσιμη από παντού και με σχεδόν μηδενικό κόστος. Προγράμματα όπως το Skype και το MSN Messenger παρέχουν δυνατότητες ανταλλαγής γραπτών μηνυμάτων, εικόνων, ήχου ή βίντεο σε πραγματικό χρόνο. Παράλληλα, η πρωτοφανής διάδοση των ιστολογίων καθιστά την επικοινωνία μαζικότερη και πιο ουσιαστική, δεδομένου ότι ο καθένας μπορεί να εκφράσει τις σκέψεις και τις απόψεις του απευθυνόμενος σε ένα τεράστιο κοινό – το κοινό του διαδικτύου στο σύνολό του. Στις ειδικά διαμορφωμένες ιστοσελίδες social bookmarking (π.χ. del.icio.us) ο κάθε μεμονωμένος χρήστης μπορεί εύκολα να επικοινωνήσει και να εντοπίσει άτομα που έχουν κοινά ενδιαφέροντα. Τέλος, οι ιστογόνοι κοινωνικής δικτύωσης αποτελούν πλέον έναν από τους βασικότερους τρόπους κοινωνικοποίησης των νεαρών κυρίως ατόμων, καθιστώντας το φαινόμενο του social networking ένα από τα πιο φλέγοντα ζητήματα της επικαιρότητας. Οι έφηβοι εκμεταλλεύονται στο έπακρο τις δυνατότητες του Παγκόσμιου Ιστού, ενημερώνονται, ψυχαγωγούνται, αναπτύσσουν κοινωνικές σχέσεις, ακούνε μουσική και δημιουργούν τις δικές τους προσωπικές ιστοσελίδες. Το Web 2.0 δημιουργεί νέες εμπειρίες που σχετίζονται με την οικογένεια, τους φίλους, τα χόμπι, τις αγορές, την εργασία, την υγεία, τα οικονομικά θέματα και γενικότερο το σύνολο των δραστηριοτήτων που πραγματοποιούνται σε καθημερινή βάση.
- **Ελεύθερη δημοσιοποίηση δεξιοτήτων, έκφρασης δημιουργικότητας, ευκαιρίες ανάδειξης:** Το YouTube.com και το MySpace.com είναι δύο από τις πλέον δημοφιλείς ιστοσελίδες, δεδομένου ότι βρίσκονται πάντα στις 10 πρώτες σε κίνηση παγκοσμίως. Οι ιστοχώροι αυτοί παρέχουν τη δυνατότητα στους χρήστες να δημοσιοποιήσουν βίντεο με τις ατομικές τους δεξιότητες (π.χ. καλλιτεχνικές, μουσική, βίντεο, φωτογραφία) έχοντας πρόσβαση σε ένα ευρύ κοινό, το οποίο δε θα μπορούσαν να προσεγγίσουν διαφορετικά. Χαρακτηριστικό παράδειγμα της «γενιάς καλλιτεχνών του MySpace» είναι το Βρετανικό συγκρότημα Arctic Monkeys (www.myspace.com/arcticmonkeys), που κατάφερε να γίνει γνωστό μέσω του διαδικτύου και να επιτύχει τις περισσότερες πωλήσεις σε μια εβδομάδα (363.735 αντίτυπα) στην ιστορία της Βρετανικής μουσικής.

- **Εκμετάλλευση της συλλογικής γνώσης κι εμπειρίας:** Το διαδίκτυο στη νέα του μορφή γίνεται ολοένα και πιο πλούσιο από εμπειρίες, γνώσεις και πληροφορίες που καταθέτονται μαζικά κάθε νέο λεπτό σε blogs, wikis και forums και ιστοχώρους κοινωνικής δικτύωσης. Οι χρήστες έτσι μπορούν ανά πάσα στιγμή να ενημερώσουν ή να ενημερωθούν για εμπειρίες σχετικά με καταναλωτικά προϊόντα, ιατρικά ζητήματα, επιστημονικά, πρακτικά και κοινωνικά θέματα. Ιστοσελίδες όπως το netmums.com αποδεικνύεται ότι επιτελούν σημαντικό κοινωνικό έργο και αντικαθιστούν παραδοσιακές μορφές κοινωνικής μέριμνας.
- **Δημοκρατικότητα, αίσθηση ένταξης σε κοινότητα, συνεργασία και συνεισφορά:** Οι χρήστες ανεξαρτήτως γεωγραφικής θέσης, κοινωνικών και φυλετικών χαρακτηριστικών μπορούν να ενταχθούν σε κοινότητες που ασχολούνται με θέματα που τους ενδιαφέρουν, να ανταλλάξουν απόψεις, να συνεργαστούν και να συνεισφέρουν στην επίτευξη ενός κοινού σκοπού. Αξίζει να σημειώσουμε ότι η εθελοντική συνεισφορά είναι τόσο διαδεδομένη ανάμεσα στους χρήστες του διαδικτύου, όσο ίσως σε κανέναν άλλο τομέα της κοινωνικής ζωής. Το μέγεθος αυτού του φαινομένου και η κατάργηση των παραδοσιακών φραγμών, κάνει πολλούς να υποστηρίζουν ότι συντελείται μια νέα κοινωνική επανάσταση.
- **Αμφίδρομη επικοινωνία χρήστη-επιχειρήσεων, ενίσχυση της διαπραγματευτικής δύναμης των χρηστών στις εμπορικές συναλλαγές:** Το καταναλωτικό κοινό αποκτά περισσότερο αξιόπιστη και ταχύτερη πρόσβαση σε μια ευρύτερη γκάμα προϊόντων κι υπηρεσιών. Μέσα από τις νέες εφαρμογές του Web 2.0 οι χρήστες έχουν λόγο κι άποψη τόσο στα προϊόντα όσο και στην πολιτική των επιχειρήσεων, συμβάλλοντας ενεργά στην αναδιαμόρφωση των υπηρεσιών προς όφελός τους. Η δυνατότητα ανάγνωσης της εμπειρίας-εκτίμησης για ένα προϊόν από άτομα που το έχουν ήδη αγοράσει ενισχύουν ακόμα περισσότερο τη θέση των καταναλωτών έναντι των εταιριών. Η παραδοσιακή διαφήμιση δεν έχει πλέον την ίδια δυναμική και οι επιχειρήσεις αναγκάζονται να αντιμετωπίσουν τους καταναλωτές πιο υπεύθυνα.
- **Νέοι τρόποι άσκησης Κρατικής Πολιτικής:** Στο πλαίσιο μιας κοινωνικής και οικονομικής διάρθρωσης που χαρακτηρίζεται από την άνιση κατανομή των φυσικών πόρων κι αγαθών, η Γνώση κι ελεύθερη διακίνησή της δε μπορεί παρά να αποτελεί το «πετρέλαιο του 21^{ου} αιώνα» όσον αφορά στα Πολιτικά δρώμενα. Το διαδίκτυο αποτελεί μια νέα βάση και υποδομή πάνω στην οποία αναπτύσσονται οι σημαντικότεροι δίαυλοι επικοινωνίας κι ενημέρωσης, δίνοντας έτσι εφελκυστήρι για μια πολιτική οργάνωση που συνοδεύεται από νέους τρόπους επιρροής της κοινής γνώμης. Ειδικότερα για την άσκηση Κρατικής Πολιτικής, ανοίγονται νέες δυνατότητες που δίνουν εντελώς

διαφορετική χροιά στο μέχρι τώρα σκηνικό. Ξεκινώντας από κυβερνητικές ιστοσελίδες και διαδικτυακές πολιτικές καμπάνιες - προεκλογικές κι επικοινωνιακές- και συνεχίζοντας με πηγές πληροφόρησης κι ενημέρωσης γύρω από πολιτικά ζητήματα είναι μόνο μερικές από τις προοπτικές που μπορούμε να εντοπίσουμε. Ενδυναμώνεται δηλαδή η επικοινωνία πολίτη-κράτους, ενισχύονται πρακτικές ηλεκτρονικής ψηφοφορίας, διεξάγονται έρευνες κοινής γνώμης, διαμορφώνονται στρατηγικές και αξιολογούνται σενάρια μεταβολής ισχυουσών πολιτικών τακτικών. Παράλληλα, Η διεκπεραίωση υποθέσεων μέσω του διαδικτύου και η δυνατότητα καταχώρησης των στοιχείων των πολιτών σε κοινή φόρμα δεδομένων, εξυπηρετεί τους πολίτες και διευκολύνει τις καθημερινές τους συναλλαγές με δημόσιες υπηρεσίες.

Αν και το πλέγμα της επιρροής του Παγκόσμιου Ιστού στα πλαίσια της κοινωνίας είναι πολύ ευρύτερο απ' ό τι παρουσιάστηκε ακροθιγώς στις προηγούμενες παραγράφους, ίσως τελικά οι έννοια κι οι διαστάσεις του Παγκόσμιου Ιστού να συνοψίζονται ακριβώς στους τίτλους των τριών ενοτήτων του Συμποσίου για τον εορτασμό της 10^{ης} επετείου του W3C (World Wide Web Consortium – Κοινοπραξία για τον Παγκόσμιο Ιστό):

- Διαδίκτυο Εννοιών (Web of Meaning)
- Διαδίκτυο επί παντός επιστητού (Web on Everything)
- Διαδίκτυο για Όλους (Web for Everyone)

Στους τρεις αυτούς τίτλους συνοψίζεται κι η ιδέα του Νέου Παγκόσμιου Ιστού 2.0 (Web 2.0), της νέας εξελιγμένης γενιάς του Συνεργατικού Διαδικτύου για την οποία γίνεται εκτενής αναφορά στην επόμενη ενότητα.



2.2. - Λογότυπο της Κοινοπραξίας για τον Παγκόσμιο Ιστό

2.2. - Το Web 2.0 ως ο Νέος Παγκόσμιος Ιστός

Ο Παγκόσμιος Ιστός από μια «καταπληκτική ιδέα» της δεκαετίας του 1990, μέσα σε διάρκεια μικρότερη από δέκα έτη, εξελίχθηκε σε καθημερινή πρακτική για σημαντικό ποσοστό του παγκόσμιου πληθυσμού, ενώ ως τα τέλη του 2010 υπολογίζεται ότι το 80% της υφηλίου θα συμμετέχει ενεργά στο διαδίκτυο. Αν η θεώρηση αυτή μοιάζει για κάποιους ουτοπική, μια αναδρομή στο παρελθόν και στην τότε συζήτηση για τη ραγδαία εξάπλωση του Παγκόσμιου Ιστού, αποδεικνύει ότι στον ψηφιακό ηλεκτρονικό κόσμο η έννοια του χρόνου είναι σχετική, και τα μεγέθη εξωπραγματικά τουλάχιστον σε σύγκριση με την ανθρώπινη καθημερινότητα. Ο Eric Schmidt χαρακτήρισε το διαδίκτυο ως «το πρώτο πράγμα που η ανθρωπότητα δημιούργησε και το οποίο η ίδια ανθρωπότητα δε κατανοεί, το μεγαλύτερο πείραμα στην αναρχία που είχαμε ποτέ».

Στη μέχρι σήμερα πορεία του Παγκόσμιου Ιστού οι εξελίξεις από τεχνολογικής σκοπιάς είναι ραγδαίες, όπως άλλωστε συμβαίνει γενικότερα στον κλάδο της Πληροφορικής και Τηλεπικοινωνιών. Την τελευταία δεκαετία όμως, παρατηρείται μια τάση ισχυροποίησης της δυνατότητας διασύνδεσης των επιμέρους χρηστών, με έμφαση στην ηλεκτρονική τους συνεργασία, την επικοινωνία, την ασφαλή ανταλλαγή πληροφοριών και την διαλειτουργικότητα. Ο Παγκόσμιος Ιστός (Web 1.0) λαμβάνει λοιπόν νέες διαστάσεις κι εμπλουτίζεται με σύγχρονους τεχνολογικούς μηχανισμούς, προκειμένου να προσφέρει ένα σύνολο νέων δυνατοτήτων. Αυτή ακριβώς η τάση δημιούργησε αυτό που ονομάζουμε Νέο Παγκόσμιο Ιστό ή Ιστό 2.0 (Web 2.0) θεσπίζοντας μια νέα εξελιγμένη γενιά διαδικτύου αναβαθμισμένων υπηρεσιών.



2.3. - Χρηστό-κεντρική θεώρηση του Web 2.0

Η έννοια του Web 2.0 εισήχθη για πρώτη φορά το 2004 στη διάρκεια ενός συνεδρίου που οργανώθηκε από την O' Reilly Media. Στα πλαίσια των εργασιών του συνεδρίου με θέμα προτάσεις για την αναβάθμιση του Παγκόσμιου Ιστού, ο Tim O'

Reilly αναφέρθηκε στον όρο Web 2.0 ως «την επιχειρηματική επανάσταση στη βιομηχανία των υπολογιστών, που προκλήθηκε από την θεώρηση του Διαδικτύου ως Πλατφόρμα και από την προσπάθεια κατανόησης των κανόνων στους οποίους οφείλεται η επιτυχία της νέας αυτής πλατφόρμας». Ουσιαστικά ο Ιστός 2.0 σηματοδοτεί τη νέα γενιά ανάπτυξης και σχεδιασμού του Παγκόσμιου Ιστού, με στόχο τη διευκόλυνση της επικοινωνίας και εν γένει τη συνεργασία, τη συμμετοχή και τη δημιουργικότητα στα πλαίσια του νέου αυτού περιβάλλοντος. Κάποια από τα βασικά σενάρια του Νέου Παγκόσμιου Ιστού ή Ιστού 2.0 περιλαμβάνουν την ανάπτυξη δικτυακών κοινοτήτων και εφαρμογές όπως οι ιστοχώροι κοινωνικής δικτύωσης (Social Networks Sites-SNS), τα Wikis, τα ιστολόγια (blogs) και τα ειδικά συστήματα καταχώρησης ετικετών δεδομένων (social tagging systems-STS).

Ολοκληρώνοντας την εισαγωγική θεώρηση για το Web 2.0, παρουσιάζουμε συνοπτικά τις σημαντικότερες τεχνολογίες που χρησιμοποιούνται και το διαφοροποιούν ως προς τον τρόπο λειτουργίας και παρουσίασης των ιστοσελίδων σε σχέση με τον παραδοσιακό Παγκόσμιο Ιστό:

- **Πλούσια και διαδραστικά interfaces χρηστών (Rich Internet Applications-RIA):** Το Web 2.0 αντιπροσωπεύεται από την τάση για όσο το δυνατόν αποδοτικότερη εκμετάλλευση του διαδικτύου με χρήση απλών τεχνολογιών. Οι RIA εφαρμογές χρησιμοποιούν τεχνολογία Flash, Javascript, Ajax κ.λπ., ενώ οι ιστοσελίδες ανανεώνονται μόνο τμηματικά για τα δεδομένα που αλλάζουν όσο ο χρήστης βρίσκεται ή επανέρχεται σε αυτές. Για παράδειγμα, στο Gmail ο υπολογισμός του διαθέσιμου αποθηκευτικού χώρου ανανεώνεται σε πραγματικό χρόνο και από όλη την ιστοσελίδα αλλάζει μόνο αυτός. Έτσι η εμπειρία χρήσης του διαδικτύου γίνεται καλύτερη και πιο ευέλικτη.
- **Χρήση CSS (Cascading Style Sheets):** Τα CSS χρησιμοποιούνται για το διαχωρισμό των δεδομένων καθαρής πληροφορίας από τα δεδομένα μορφοποίησης σε μια ιστοσελίδα. Η εφαρμογή της τεχνολογίας αυτής, πέρα από την οικονομία στο εύρος ζώνης του δικτύου, προσφέρει και ευελιξία στον τρόπο παρουσίασης των δεδομένων, αφού ο χρήστης βλέπει τα δεδομένα σύμφωνα με τα CSS που ο ίδιος έχει ορίσει. Για παράδειγμα, τα ίδια δεδομένα μπορούν να παρουσιαστούν σε οθόνη υπολογιστή, κατευθείαν σε εκτυπωτή, σε μορφή ανάγνωσης για τυφλούς ή ακόμα και να μετατραπούν σε φωνή με χρήση κατάλληλου λογισμικού.
- **Χρήση σημασιολογικών δεδομένων και microformats:** Με αυτό τον τρόπο περιγράφεται η σημασία των δεδομένων που περιέχουν οι ιστοσελίδες. Έτσι, αυτά κατηγοριοποιούνται και η αναζήτησή τους γίνεται ευκολότερη και αποδοτική.
- **Χρήση RSS feeds:** Η τεχνολογία RSS (Really Simple Syndication) αποτελεί έναν απλό τρόπο για την αποστολή πληροφοριών από μια ιστοσελίδα, χωρίς να χρειάζεται ο χρήστης να μπαίνει στη διαδικασία συχνών επισκέψεων και αναζήτησης των πληροφοριών αυτών από το site. Το

μόνο που χρειάζεται είναι να δηλώσει στο πρόγραμμα RSS -ή στην online υπηρεσία RSS- τι ακριβώς επιθυμεί να εντοπίσει στο Διαδίκτυο και αυτομάτως αποστέλλονται οι νέες πληροφορίες στον υπολογιστή του.

- **Χρήση ελεύθερου λογισμικού:** Για παράδειγμα Linux σαν λειτουργικό σύστημα, Apache σαν Web server, MySQL σαν βάση δεδομένων και PHP, Pearl, Python σαν γλώσσες προγραμματισμού. Οι όροι «Ελεύθερο Λογισμικό» (Free Software) και «Λογισμικό Ανοικτού Κώδικα» (Open Source Software) αναφέρονται σε προγράμματα των οποίων ο πηγαίος κώδικας είναι προσβάσιμος σε άτομα εκτός της εταιρίας παραγωγής τους και συνεργατών της. Οι όροι αυτοί δεν αναφέρονται σε λογισμικό που διατίθεται δωρεάν (freeware), καθώς το ελεύθερο λογισμικό μπορεί να έχει τιμή πώλησης σε πολλές περιπτώσεις μεγάλη.
- **«Ελαφρά» πρωτόκολλα δικτύου REST και SOAP:** Τα πρωτόκολλα αυτά χρησιμοποιούν απλές HTTP εντολές (get, post, put κ.λπ.) για ανάκτηση δεδομένων από τους servers.
- **Αρχιτεκτονικές SOA (Service Oriented Architecture):** Επιτρέπουν το διαμοιρασμό και την επαναχρησιμοποίηση υπηρεσιών-εφαρμογών από διαφορετικά προγράμματα λογισμικού και SaaS (Software as a Service) όπου οι εφαρμογές είναι εγκατεστημένες σε κεντρικό server στο δίκτυο και οι χρήστες τις χρησιμοποιούν μέσω browser ανεξαρτήτως ηλεκτρονικού υπολογιστή, τόπου και χρονικής στιγμής.

2.3. - Web 1.0 vs Web 2.0: βασικά σημεία διαφοροποίησης

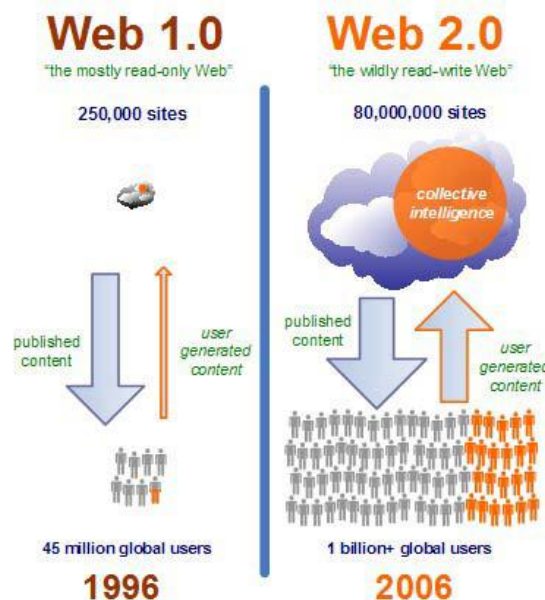
Συζητώντας τώρα περί Ιστού 2.0 κι αντιπαραβάλλοντάς τον με τον Ιστό 1.0 εύλογα δημιουργείται το ερώτημα περί του «προφίλ» της κάθε έκδοσης και των πιθανών διαφοροποιήσεων μεταξύ τους. Είναι προφανές καταρχήν ότι ως **Ιστό 1.0 (Web 1.0)** θεωρούμε την τάξη πραγμάτων που επικρατούσε στον Παγκόσμιο Ιστό πριν την εμφάνιση του **Ιστού 2.0 (Web 2.0)**. Παρά το προφανές στην διαφοροποίηση των όρων δεν είναι εξίσου προφανείς οι λόγοι που τους διαφοροποιούν.

Ο **Ιστός 1.0** χαρακτηριζόταν μεταξύ άλλων από στατικές ιστοσελίδες, η ανάπτυξη των οποίων βασιζόταν σε μεγάλο βαθμό στη χρήση πλαισίων (frames), ενώ η HTML επέτρεπε –έστω και περιορισμένα- προσωπική χρήση διαφόρων ετικετών, κατά τη βούληση κάθε χρήστη. Πέρα όμως από το περιεχόμενο και το αποτέλεσμα των ιστοσελίδων, εξίσου στατική ήταν και η ανάδραση των εφαρμογών με το χρήστη, όπου ακόμα και στην περίπτωση επικοινωνίας μέσω φόρμας σε μια ιστοσελίδα, η φόρμα αυτή θα αποστέλλοταν στον αποδέκτη με χρήση ηλεκτρονικού ταχυδρομείου.

Το περιβάλλον άρχισε να μεταβάλλεται και να εξελίσσεται σε αυτό που σήμερα ονομάζεται **Ιστός 2.0**, όταν τόσο οι προγραμματιστές πακέτων κι εφαρμογών λογισμικού όσο και οι τελικοί χρήστες των εφαρμογών αυτών, άρχισαν να τροποποιούν τις μεθόδους χρήσης και τη φιλοσοφία αξιοποίησης των δυνατοτήτων

του Παγκόσμιου Ιστού. Έτσι οι μεταβολές του σκηνικού δεν εντοπίζονται μόνο στο τεχνολογικό πεδίο, αλλά περιλαμβάνουν μια καινούρια διαφορετική αντίληψη των πραγμάτων από τους ίδιους τους χρήστες.

Αν κι ο όρος “**Web 2.0**” φαίνεται να έχει επικρατήσει ο ίδιος ο Tim Berners – Lee στάθηκε επιφυλακτικός αναφορικά με την ουσιαστική διαφοροποίηση μεταξύ Web 1.0 και Web 2.0. Παρά του ότι οι υποστηρικτές του Web 2.0 διατείνονται ότι το «*Web 1.0 συνέδεε υπολογιστές ενώ το Web 2.0 συνδέει ανθρώπους*», ο επινοητής του Παγκόσμιου Ιστού σωστά απαντά ότι ο Ιστός 2.0 στηρίζεται τεχνολογικά στον προϋπάρχοντα Ιστό 1.0, οπότε στην πραγματικότητα δε μπορούμε να μιλήσουμε για μια μεγάλη τεχνολογική επανάσταση. Υπάρχουν κι άλλοι εξίσου επιφυλακτικοί με τη νέα αυτή έκδοση του Ιστού, όπως ο Best που προβληματίζεται αν ο Ιστός 2.0 καταλήξει ως η νέα «Διαδικτυακή Φούσκα». Παράλληλα οι Madden et al ξεκαθαρίζουν ότι αφενός το Web 2.0 δεν είναι ένα internet 2.0 που λειτουργεί σε εντελώς ξεχωριστή βάση από την προϋπάρχουσα κι αφετέρου, ότι ένα μεγάλο σύνολο χρηστών είναι απλά γνώστες του όρου, χωρίς όμως να γνωρίζουν έστω κι αμυδρά το περιεχόμενο της αναφοράς. Υπάρχουν τέλος αυτοί που υποστηρίζουν ότι το Web 2.0 είναι όρος του μάρκετινγκ.



2.4. - Σχηματική αντιπαραβολή του Web 1.0 και του Web 2.0

Παρά το σκεπτικισμό του εφευρέτη του Παγκόσμιου Ιστού κι άλλων επιστημόνων κι ειδικών σχετικά με την έννοια του Νέου Παγκόσμιου Ιστού, ο όρος Web 2.0 κατέστη δημοφιλής κι απόδειξη αυτού αποτελούν οι 9,5 εκατομμύρια αναφορές στον όρο στις σελίδες του Google μέσα σε διάστημα 1,5 έτους από την παρουσίασή του. Ο Tim O’ Reilly στο άρθρο του “What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software” (2009) κάνει μια πρακτική προσπάθεια αντιπαραβολής και πινακοποίησης των διαφορών των δύο αυτών εκδόσεων.

Web 1.0	Web 2.0
DoubleClick	Google AdSense
Ofoto	Flickr

Akamai	BitTorrent
Mp3.com	Napster
Britannica Online	Wikipedia
Personal websites	blogging
evite	Upcoming.org and EVDB
Domain name speculation	Search engine optimization
Page views	Cost per click
Screen scraping	Web services
publishing	participation
Content management systems	wikis
Directories (taxonomy)	Tagging (folksonomy)
stickiness	syndication

2.5. - Ιστός 1.0 VS Ιστός 2.0 κατά τον Tim O'Reilly

Ίσως η φράση του Miller (2005), «*To Web 1.0 οδήγησε τους ανθρώπους στην πληροφορία. Το Web 2.0 οδηγεί την πληροφορία στους ανθρώπους*», συμπυκνώνει εύστοχα όλους τους λόγους που διαφοροποιούν το Νέο Παγκόσμιο Ιστό και δικαιολογούν την επέκταση «2.0». Συμπληρωματικά με την θεώρηση του Tim O'Reilly και προς αποσαφήνιση των λόγων αυτών, συνοψίζονται στον ακόλουθο πίνακα οι κυριότερες αλλαγές της μετάβασης του Web 1.0 στο Web 2.0.

Web 1.0	Web 2.0
Διαδικτυακές υπηρεσίες που προσφέρονται μέσω του WWW (Online κείμενο με εικόνα, ήχο, υπερσυνδέσμους)	Υπηρεσίες κοινωνικού περιεχομένου, Κοινωνική Δικτύωση, Κοινότητες (Online Communities)
Στατικότητα	Δικτύωση κι αυτοματισμός
Προϊόντα	Υπηρεσίες
Σύνδεση ιστοσελίδας με άλλες μόνο μέσω υπερσυνδέσμων	Real-time εμπλουτισμός ιστοσελίδας με περιεχόμενο από άλλες ιστοσελίδες (π.χ. RSS feeds, widgets)
Ιστοσελίδες μόνο για ανάγνωση ("the mostly read only web")	Συμμετοχή, διαδραστικότητα, συνεισφορά ("read-write web")
Οργανωμένος, στοχευμένος, ελεγχόμενος «μονόλογος»	Διάλογος, συνεργασία, συλλογικότητα, αλληλεπίδραση, εμπιστοσύνη
Περιορισμένες δυνατότητες αμφίδρομης επικοινωνίας (π.χ. web forms, e-mails)	Διαδραστικότητα και συμμετοχή του χρήστη (IM, comments, sharing κ.λπ.)
owning	Sharing and remix
HTML	XML, RSS, Ajax - Δυναμική αλλαγή περιεχομένου ιστοσελίδων
portals	Απολύτως παραμετροποιήσιμη start page από τον χρήστη (π.χ. igoogle, Netvibes, pageflakes)

Προσωπική ή εταιρική ιστοσελίδα	Ιστολόγιο (blog) ή wiki
Συστήματα διαχείρισης περιεχομένου (CMS)	wiki
Στατική διαφήμιση (banner, flash, page views)	Κόστος ανά επίσκεψη (cost per click), χρήση rich media (ήχος, εικόνα κ.λπ.), στοχευόμενη διαδικτυακή στρατηγική μάρκετινγκ (viral marketing)
Online πηγές πληροφόρησης (π.χ. εγκυκλοπαίδειες)	Δημιουργία wikis (π.χ. Wikipedia)
Σελιδοδείκτες (bookmarks /favorites)	RSS feeds , Social bookmarking
Directories – taxonomy (ελεγχόμενα λεξιλόγια)	Tags – folksonomy (δημιουργία από τον ίδιο το χρήστη)
Χρήση Netscape και Internet Explorer	Χρήση Chrome, Firefox, Flock, IE8 κ.λπ.
Χρήση λογισμικού κλειστού κώδικα	Εκτεταμένη χρήση open source λογισμικού
Χρήση τοπικά εγκατεστημένων στον Η/Υ εφαρμογών	Κοινές εφαρμογές Web (Network as a platform), υβριδικές εφαρμογές
Αποθήκευση δεδομένων στον τοπικό δίσκο	Online storage
Χρήση desktop PC για πρόσβαση στο web, καλωδιακή σύνδεση	Χρήση πολλών μέσων πρόσβασης στο web (PDAs, laptops, mobile phones κ.λπ.)

2.6. - Αλλαγές της μετάβασης του Web 1.0 στο Web 2.0

2.4. - Η «έξυπνη» γενιά του Διαδικτύου: Web 3.0 ή Σημασιολογικός Ιστός (Semantic Web)

Με τον όρο **Web 3.0** αναφερόμαστε στην επερχόμενη νέα εξέλιξη του Παγκόσμιου Ιστού που εισήχθη από τον ίδιο το δημιουργό του τον Tim Berners- Lee. Ο νέος ευφυής Ιστός με χρήση της τεχνητής νοημοσύνης, των σύγχρονων γραφικών και των ταχύτερων συνδέσεων θα προσφέρει μια ακόμα πιο ικανοποιητική και συναρπαστική εμπειρία στον τελικό χρήστη. Στο επίκεντρο θα βρεθούν οι «έξυπνες μηχανές αναζήτησης» που θα έχουν την δυνατότητα εξόρυξης και συνδυασμού δεδομένων με ανθρώπινη λογική, σε αντιδιαστολή με τον απλό εντοπισμό «λέξεων - κλειδιών» που ισχύει μέχρι σήμερα. Σύμφωνα με τους ειδικούς ήδη από τα τέλη του 2009 διανύουμε τα πρώτα βήματα του Σημασιολογικού Ιστού. Οι σημαντικότερες των αλλαγών ωστόσο αναμένονται τα επόμενα έτη. Πιο συγκεκριμένα θα μπορούσαμε να εντοπίσουμε τις εξής βασικές διαφοροποιήσεις:

- **Πρόσβαση από παντού με κινητές ή ασύρματες συσκευές:** οι απλές μόντεμ γραμμές αντικαταστάθηκαν από το Ethernet, από το Ethernet προχωρήσαμε στο Wifi κι ακολουθούν τα 3G, 4G και WiMax (LTE) δίκτυα. Υπάρχουν μάλιστα σχέδια για metropolitan WiMax δίκτυα μεγαλύτερης εμβέλειας, καθώς τα σημερινά καλύπτουν απόσταση

μόλις 35 χιλιομέτρων (μικρή σε σχέση με τα 100 χιλιόμετρα εμβέλεια των Wifi. Πλέον το Διαδίκτυο θα είναι διαθέσιμο παντού, γρηγορότερο και φθηνότερο από ποτέ, ενώ η χρήση του θα είναι το ίδιο εύκολη με την απάντηση μιας εισερχόμενης κλήσης σε κινητό τηλέφωνο.

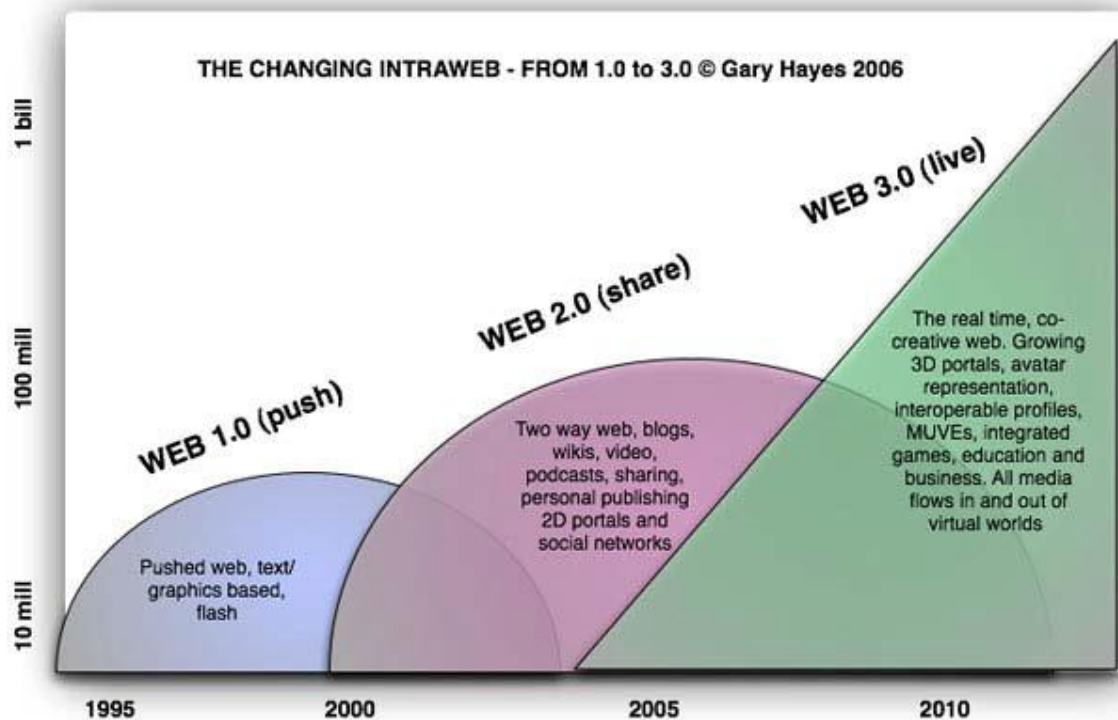
- **Δυνατότητα προσδιορισμού της θέσης (location) με GPS-awarded εφαρμογές :** ως αποτέλεσμα της ενίσχυσης των δυνατοτήτων πρόσβασης στο διαδίκτυο μέσω κινητών συσκευών, αναμένεται αντίστοιχα η ενσωμάτωση σε αυτές GPS τεχνολογίας (Global Positioning System – Παγκόσμιο Σύστημα Εντοπισμού Θέσης) που βασίζεται σε ένα πλέγμα 24 δορυφόρων της Γης, πάνω στους οποίους είναι ειδικά προσαρμοσμένοι δέκτες ικανοί να εντοπίζουν οποιαδήποτε θέση. Θα έρθουμε αντιμέτωποι επομένως, με μια καινοτομική σειρά κινητών συσκευών (GPS - awarded) που θα αλλάξουν τον καθεαυτό τρόπο που αντιλαμβανόμαστε το χώρο γύρω μας. Ασύρματες αλληλεπιδράσεις μεταξύ μηχανών, οχημάτων, συσκευών, αισθητήρων και πολλών άλλων διατάξεων θα δώσουν εντελώς νέα χροιά στη καθημερινότητά μας. Έτσι καθίσταται δυνατή η χρήση ηλεκτρονικών καρτών εισιτηρίων, η επικοινωνία κινητών συσκευών για ανταλλαγή πληροφοριών κι εκτέλεση πληρωμών, ακόμα κι η ανάκτηση πληροφοριών από διαφημιστικές πινακίδες. Ήδη βρίσκεται υπό μελέτη, αν και σε εκκολαπτόμενο στάδιο, η εισαγωγή “location – based” εφαρμογών σε Κοινωνικά Δίκτυα όπως το Twitter, σε μια προσπάθεια των διαφημιστών να εκμεταλλευτούν αυτή την τεράστια νέα πρόκληση που ανοίγεται στην αγορά. Το Google έχει ήδη ακολουθήσει το παιχνίδι των εξελίξεων με την εγκαίνιαση του Buzz ως τμήμα της υπηρεσίας ηλεκτρονικού ταχυδρομείου Gmail. Προβλέπεται μάλιστα, ότι η τεχνολογία αυτού του είδους θα έχει εγκατασταθεί σε περισσότερο από 1 δισεκατομμύρια τηλέφωνα μέχρι το έτος 2015.
- **Κοινωνικά Δίκτυα στο επίκεντρο:** ο άνθρωπος είναι ως γνωστόν κοινωνικό ον. Ιστοχώροι Κοινωνικής Δικτύωσης όπως το Facebook, το Twitter, το Flickr είναι ήδη καθημερινή πρακτική για τη συντριπτική πλειοψηφία των χρηστών και αναμένεται να ενσωματωθούν ακόμα περισσότερο στη ζωή όλων μας, αλλάζοντας ριζικά τον τρόπο της επικοινωνίας μεταξύ μας. Τα επόμενα χρόνια λοιπόν, ο περισσότερος διαδικτυακός (online) χρόνος θα ξοδεύεται σε διάφορων μορφών «κοινωνικές» δραστηριότητες στα πλαίσια σύγχρονων διαδραστικών κοινωνικών πλατφορμών, που θα υποστηρίζουν «έξυπνες» εφαρμογές τρισδιάστατων γραφικών.

Υπάρχουν μάλιστα ήδη ερευνητικά προγράμματα που βρίσκονται σε εξέλιξη μελετώντας αυτή ακριβώς την προοπτική εκσυγχρονισμού του Παγκόσμιου Ιστού. Στο πανεπιστήμιο της Ουάσιγκτον το “KnowItAll” , ένα πρόγραμμα που υποστηρίζεται από το Google, έχει τη δυνατότητα εντοπισμού κι εξαγωγής πληροφοριών που έχουν αναρτηθεί από τους χρήστες σε διάφορους ιστοχώρους. Στην ίδια κατεύθυνση, ως πρόδρομος των εργαλείων του επερχόμενου Web 3.0 είναι το λογισμικό “Adaptive Blue” που έχει τη δυνατότητα αποθήκευσης ιστοσελίδων που

έχει επισκεφθεί ο χρήστης και χρήση αυτών για πιο αποδοτική κι εξατομικευμένη χρήση του διαδικτύου.

Όπως χαρακτηριστικά λέει η Viviane Reding, το Web 3.0 σημαίνει «αδιάλειπτη, οποτεδήποτε και οπουδήποτε, επιχειρηματική δραστηριότητα, ψυχαγωγία και κοινωνική δικτύωση με ταχύτητα, αξιοπιστία και ασφαλή δίκτυα. Σημαίνει το τέλος μεταξύ κινητών και σταθερών γραμμών. Σηματοδοτεί άλμα δεκαπλασιασμού της κλίμακας του ψηφιακού σύμπαντος που θα έχει συντελεστεί το έτος 2015». Ο διευθύνων σύμβουλος της Google, Eric Schmidt, με τη σειρά του, έδωσε έναν από τους πιο σαφείς ορισμούς του Web 3.0 όταν ρωτήθηκε από μέλος του κοινού στο Digital Forum της Σεούλ. Δήλωσε ότι το Web 3.0 «θα απαρτίζεται από εφαρμογές που θα λειτουργούν όλες μαζί και θα διατίθενται μέσα από κοινωνικά δίκτυα, ηλεκτρονικό ταχυδρομείο κλπ. Οι εφαρμογές θα είναι σχετικά μικρές, εξαιρετικά γρήγορες και θα μπορούν να τρέξουν σε οποιονδήποτε υπολογιστή και να τροποποιηθούν από οποιονδήποτε χρήστη».

Τελικά πολύ σωστά κάποιος θα αναρωτηθεί τι ακριβώς είναι αυτό που δίνει την επέκταση «1.0, 2.0 ή 3.0» και διαφοροποιεί την κάθε γενιά Διαδικτύου. Αυτό που πρέπει να διευκρινίσουμε είναι ότι οι επεκτάσεις αυτές χρησιμοποιούνται κατά σύμβαση απλά για να περιγράψουν νέα πακέτα υπηρεσιών ή ακόμα καλύτερα τη νέα φιλοσοφία που αντιπροσωπεύει την κάθε εξέλιξη. Δε μιλάμε δηλαδή για εντελώς καινούριο Παγκόσμιο Ιστό, αλλά για εμπλουτισμό της υπάρχουσας τεχνολογικής πλατφόρμας. Ο Gary Hayes με ένα αρκετά ευκρινές διάγραμμα κάνει μια προσπάθεια να διασαφηνίσει τα όρια μεταξύ των εκδόσεων, τόσο χρονικά όσο κι από άποψη περιεχομένου, ενώ παράλληλα δίνει μια εικόνα για τη προστιθέμενη αξία του Διαδικτύου στο πέρασμα του χρόνου.



2.7. - Η εξέλιξη του διαδικτύου

Ο Ιστός 1.0 λοιπόν παρουσιάζεται στατικός καθώς αφορά στην απλή «προώθηση» (push) πληροφοριών μέσω γραφικών και flash εφαρμογών. Ήδη από το 2000 κάνει την πρώτη εμφάνιση του ο Ιστός 2.0 που βασίζεται στη «διαμοίραση κι επικοινωνία» (share). Πρόκειται για ένα διαδίκτυο δύο κατευθύνσεων, όπου ο χρήστης κάνει προσωπικές δημοσιεύσεις σε ιστολόγια, ιστοχώρους διαμοίρασης φωτογραφιών ή βίντεο, ενώ παράλληλα είναι μέλος διαδικτυακών κοινοτήτων. Ο Ιστός 3.0 τέλος, του οποίου τα όρια είναι ασαφή και θεωρητικά ήδη διανύουμε τα πρώτα βήματά του, αφορά σε μια «εμπειρία πραγματικού χρόνου» (live). Πρόκειται για ένα διαδίκτυο που υποστηρίζει ιστοχώρους Εικονικής Πραγματικότητας, οι χρήστες έχουν βιωματικές εμπειρίες ως avatars ενώ οι επιχειρήσεις εμπλέκονται πλέον ενεργά αυξάνοντας κατακόρυφα την αξία του Παγκόσμιου Ιστού.

Χρήσιμη θεωρούμε τέλος και την θεώρηση της Marta Strickland όπως αυτή παρουσιάζεται στη μελέτη της με τίτλο “The Evolution of Web 3.0”, στην οποία δίδονται πινακοποιημένα τα βασικά σημεία διαφοροποίησης των τριών εκδόσεων του Διαδικτύου.

Web 1.0	Web 2.0	Web 3.0
Διαδίκτυο απλής ανάγνωσης (read only)	Ανοιχτό Διαδίκτυο γραφής κι ανάγνωσης (read and write)	Προσωποποιημένο διαδίκτυο
Επικέντρωση στις εταιρίες	Επικέντρωση στις κοινότητες	Επικέντρωση στο άτομο
Προσωπικές ιστοσελίδες	Ιστολόγια	Lifestream
Κτήση περιεχομένου (Owning content)	Διαμοίραση περιεχομένου (Sharing content)	Παγίωση δυναμικού περιεχομένου (Consolidating dynamic content)
HTML, portals	XML, RSS	Ο Σημασιολογικός Ιστός (Semantic Web)
Web forms	Web applications	Widgets, drag & drop mashups
Directories (taxonomy)	Tagging (folksonomy)	Κατανόηση συμπεριφοράς χρηστών (User behaviour)
Netscape	Google	iGoogle, NetVibes
Αριθμός προβολών ιστοσελίδας	Κόστος ανά κλικ (Cost per click)	User engagement
Διαφήμιση	Rich media, viral marketing	Ψυχαγωγική διαφήμιση (Advertainment)

2.8. - Συγκριτική Αξιολόγηση των τριών εκδόσεων του διαδικτύου

2.5. - Επισκόπηση στις νέες τεχνολογίες του Web 2.0

Στο σημείο αυτό, κρίνεται σκόπιμη η αναφορά στις βασικές υπηρεσίες ή εφαρμογές που αναδύονται με τον επαναπροσδιορισμό των τεχνολογιών του Web 1.0, προς την κατεύθυνση της νέας φιλοσοφίας συμμετοχικότητας και συνεργατικότητας που χαρακτηρίζει το Web 2.0. Ανάμεσα σε αυτές τις εφαρμογές ξεχωρίζουν τέσσερις που όχι μόνο έτυχαν ένθερμης αποδοχής από πλευράς των χρηστών, αλλά δημιούργησαν κι εξακολουθούν να αναπτύσσονται και να εξελίσσονται εκείνο το δικτυακό περιβάλλον που ενισχύει τη συνεργασία και τη διάδραση των ανά τον κόσμο χρηστών. Έτσι αναφέρονται οι **ιστοχώροι κοινωνικής δικτύωσης** (social networks sites SNS), τα **Wikis**, τα **ιστολόγια** (blogs) και τα ειδικά συστήματα καταχώρησης **ετικετών δεδομένων** (social tagging systems STS). Για πληρέστερη όμως εικόνα του Νέου Παγκόσμιου Ιστού και προκειμένου να καταστεί σαφές τι είναι αυτό που τον διαφοροποιεί από την προηγούμενη έκδοση, παραθέτουμε και μερικές επιπλέον τεχνολογίες όπως τα **Podcasts**, οι **Κόσμοι Εικονικής Πραγματικότητας** (Virtual Worlds), οι **Ηλεκτρονικές Συνεδριάσεις** (Web Conferences) και οι **Ιστοχώροι Διαμοίρασης Φωτογραφιών** (Photo Sharing Sites).

2.5.1. - Ιστολόγια (Blogs)

Η ετυμολογία του όρου “**blog**” υποδηλώνει μεταξύ άλλων την ίδια την έννοια της εφαρμογής. Προερχόμενος από σύντμηση των λέξεων “web log”, δηλαδή «δικτυακή εγγραφή», επικράτησε σαν “blog” όταν ο Peter Merholz με διάθεση αστεϊσμού διέσπασε τη λέξη weblog δημιουργώντας τη φράση “we blog” (δηλαδή εμείς blogάρουμε- διατηρούμε ή καταχωρούμε επαφές στα blogs). Προκάτοχοι των blogs ήταν οι λίστες ηλεκτρονικού ταχυδρομείου, τα Usenet, τα bulletin board συστήματα και οι εμπορικές δικτυακές υπηρεσίες. Το πρώτο γνωστό blog δημιουργήθηκε και λειτούργησε το 1992 από τον πρωτοπόρο κι εμπνευστή του Παγκόσμιου Ιστού, τον Tim Berners-Lee, με το όνομα “What’s New Page”. Τα επόμενα χρόνια οι ρυθμοί ανάπτυξης των ιστολογίων είναι παραδόξως αργοί, αφού ως το τέλος του 1999 καταμετρώνται μόλις 23 γνωστά blogs. Από το σημείο αυτό και μετά ωστόσο, οι εξελίξεις είναι αλματώδεις. Ενδεικτικά αναφέρουμε ότι το 2006 καταμετρήθηκαν 50 εκατομμύρια ιστολόγια με νέα να μπαίνουν σε λειτουργία κάθε ένα δευτερόλεπτο, σύμφωνα με έρευνα των Tapscott και Williams (2006).

Σύμφωνα με τη Wikipedia, ένα blog αποτελεί μια ιστοσελίδα στην οποία καταχωρούνται εγγραφές χρηστών διάφορου περιεχομένου και θεματολογίας, που παρουσιάζονται με χρονολογική σειρά. Θα πρέπει να τονίσουμε όμως ότι τα blogs διαφοροποιούνται από μια απλή ιστοσελίδα, κυρίως στο γεγονός ότι η ανανέωσή τους είναι ιδιαίτερα απλή και θα μπορούσαμε να πούμε τετριμμένη. Χαρακτηριστικά, το να καταχωρηθεί μια νέα εγγραφή σε ένα ιστολόγιο είναι τόσο απλή όσο η αποστολή ενός μηνύματος ηλεκτρονικού ταχυδρομείου, το γνωστό και καθημερινό σε όλους e-mail. Μπορεί βέβαια χάρη στις νέες τεχνολογίες η διατήρηση κι ανανέωση μιας ιστοσελίδας να έχει απλουστευτεί ιδιαίτερα, ωστόσο ακόμα κι έτσι παραμένει συνθετότερη και σίγουρα περισσότερο χρονοβόρα διαδικασία. Επιπρόσθετα, ένα ιστολόγιο – blog συνοδεύεται από ένα μηχανισμό ανατροφοδότησης (feedback) από πλευράς του αναγνώστη – επισκέπτη κάτι που καθιστά τη χρήση blog άμεση και απλή και το ίδιο το ιστολόγιο ένα «φόρουμ επικοινωνίας». Αντίθετα, στις περισσότερες ιστοσελίδες το αμεσότερο σύστημα επικοινωνίας που παρέχεται, είναι κάποια φόρμα

αποστολής e-mail στο διαχειριστή ή στον κάτοχο της σελίδας. Τέλος, αυτό που ουσιαστικά διαφοροποιεί τα ιστολόγια από μια κοινή τυπική ιστοσελίδα, είναι το προσωπικό στυλ του συγγραφέα και των επισκεπτών του δίνοντας έτσι στο blog έναν ιδιαίτερο εξατομικευμένο χαρακτήρα.

Εκτός όμως από τα blogs συναντώνται και τα **micro-blogs** που δε διαφέρουν παρά μόνο στην έκταση των εγγραφών, καθώς λειτουργούν βάση πολύ σύντομων καταχωρήσεων. Τόσο τα blogs όσο και τα micro-blogs δίνουν τη δυνατότητα εγγραφών σε μορφή όχι μόνο κειμένου αλλά και εικόνας, μουσικής ή βίντεο αποτελώντας έτσι κομμάτι των κοινωνικών μέσων (social media). Υπάρχουν διάφορες κατηγορίες blogs. Σύμφωνα με τη Wikipedia τα πιο συνήθη είναι τα προσωπικά blogs. Το τύπου προσωπικό ημερολόγιο, αποτελεί αυτή την απλούστερη περίπτωση επικοινωνίας, άλλες φορές σα δημοσίευση προσωπικών μηνυμάτων και απόψεων και άλλοτε σα καθρέφτισμα σκέψεων και συναισθημάτων. Τα επιχειρηματικά blogs όλο και αυξάνονται αποτελώντας μέσο επικοινωνίας για τα μέλη της επιχείρησης, εργαλείο δημοσίων σχέσεων και μάρκετινγκ, ή ακόμα και στοιχείο ενίσχυσης της μάρκας (branding).

Συνεχίζοντας την κατηγοριοποίηση, ενδιαφέρον παρουσιάζει η κατάταξη των blogs βάσει του τύπου δεδομένων που κατά προτεραιότητα αξιοποιούν και φιλοξενούν. Έτσι συναντώνται τα **vlogs** (video & blogs) που περιλαμβάνουν κυρίως βίντεο, τα **linklogs** (link & blogs) που αποτελούνται κυρίως από υπερσυνδέσεις προς εσωτερικούς συνδέσμους στο διαδίκτυο, τα **sketchlogs** (sketch & blogs) που περιλαμβάνουν κυρίως σκίτσα και κατ' αντιστοιχία τα **photoblogs** (photo & blogs) που περιλαμβάνουν κατά βάση φωτογραφίες. Μια περισσότερο διαφοροποιημένη κατηγορία είναι αυτή των **artlogs** (art & blogs) που αποτελούν κανάλια δημοσιοποίησης τέχνης.

Στα πλαίσια της διαρκώς εξελισσόμενης και αναπτυσσόμενης blogόσφαιρας, ο Tim O' Reilly προσπαθώντας να συμβάλει στην αρτιότερη διαμόρφωσή της πρότεινε το 2007 το Κώδικα Δεοντολογίας των bloggers, αναφέροντας μεταξύ άλλων ότι «ο κώδικας δεοντολογίας στοχεύει στη δημιουργία μιας κουλτούρας που θα ενθαρρύνει την ελεύθερη προσωπική έκφραση παράλληλα με την ανάπτυξη δημιουργικής συζήτησης στα πλαίσια της blogόσφαιρας».

Σε μια προσπάθεια ένδειξης του μεγέθους των διαστάσεων που έχει λάβει το φαινόμενο των blogs αξίζει να σημειώσουμε μερικά αριθμητικά στοιχεία. Σύμφωνα με την αναφορά της Technorati, με τίτλο “State of the Blogosphere 2008”, καταμετρήθηκαν σε διάστημα έξι ετών (2002-2008) περίπου 113 εκατομμύρια blogs. Σε διάστημα μάλιστα 120 ημερών, καταχωρήθηκαν περισσότερα από 7,4 εκατομμύρια blogs ενώ σε διάρκεια 24 ωρών περίπου 900 χιλιάδες. Εκτιμάται ότι η blogόσφαιρα διπλασιάζεται σε μέγεθος κάθε 6 μήνες. Οι αριθμοί καταμαρτυρούν ότι τα blogs είναι ένα μαζικό δικτυακό φαινόμενο, ικανοποιώντας την ανάγκη των ανθρώπων να «ακούνε και να ακούγονται», ελεύθερα, αβίαστα, ισότιμα. Σταδιακά αναδεικνύονται σε ένα μέσο μαζικής ενημέρωσης, το οποίο αξιοποιούν οι πολίτες προκειμένου να ενημερωθούν, ενώ την ίδια στιγμή μειώνεται η επιρροή των παραδοσιακών μέσων, όπως η τηλεόραση και οι εφημερίδες.



2.5.2. - WIKIS

Τα **Wikis** είναι ιστοχώροι που δίνουν τη δυνατότητα, σε οποιονδήποτε τις επισκέπτεται, να παρέμβει στο περιεχόμενο τους τροποποιώντας το. Αναλόγως την πολιτική του ιστοχώρου, οι χρήστες θα πρέπει να διατηρούν κάποιο λογαριασμό χρήστη ή μπορεί ελεύθερα να έχουν δικαίωμα συνεισφοράς. Από το δικαίωμα αυτό της συμμετοχής εξαιρούνται οι χρήστες που για κάποιο λόγο βρίσκονται στη λίστα αποκλεισμού, ενδεχομένως επειδή παρέβησαν κάποιον Κανόνα Λειτουργίας στο παρελθόν. Το αποτέλεσμα αυτής ακριβώς της λογικής δίνει μια αξιόπιστη βάση δεδομένων η οποία ανανεώνεται συνεχώς με τη συλλογική προσπάθεια όλων των χρηστών της. Γενικεύοντας, και σύμφωνα με τους Klamma et al.(2008) τα Wikis *«αποτελούν κατά κάποιο τρόπο κοινωνικά δίκτυα καθώς στα πλαίσιά τους παρατηρούνται αλλαγές τόσο στους κόμβους του δικτύου (τις σελίδες των Wikis) όσο και στις σχέσεις μεταξύ των κόμβων»*.

Τα Wikis έχουν πολλές εφαρμογές, όπως η δημιουργία συνεργατικών ιστοσελίδων και οι ιστοσελίδες δικτυακών κοινοτήτων. Χαρακτηριστικό και γνωστότερο παράδειγμα Wiki είναι η μεγάλη δικτυακή, ελεύθερη, συνεργατική εγκυκλοπαίδεια Wikipedia που υποστηρίζεται από τον μη Κερδοσκοπικό Οργανισμό Wikimedia Foundation και υλοποιήθηκε το 2001. Ωστόσο, εφαρμογές Wiki συναντώνται και σε επιχειρησιακό επίπεδο, κυρίως στις περιπτώσεις των ενδοδικτύων (intranets) και των συστημάτων διαχείρισης γνώσης και πληροφοριών.

Η πρώτη εφαρμογή Wiki, το **WikiWikiWeb** (γνωστό και σαν WordsWeb) αναπτύχθηκε από τον Ward Cunningham, έναν Αμερικάνο προγραμματιστή ηλεκτρονικών υπολογιστών, το 1994. Σύμφωνα με το δημιουργό του το WikiWikiWeb χρησιμοποιήθηκε για τη διευκόλυνση της ανταλλαγής ιδεών μεταξύ των προγραμματιστών και είναι *«η απλούστερη βάση δεδομένων η οποία θα μπορούσε να λειτουργήσει»*. Η ονομασία του Wiki προέρχεται από τη χαβανέζικη λέξη “Wiki” που σημαίνει γρήγορα, οπότε ο Cunningham με τον όρο WikiWikiWeb παραπέμπει σε ένα “Quick Web” δεδομένου ότι υποστηρίζει την εύκολη και ταχεία επεξεργασία των ιστοσελίδων του από τους χρήστες. Σε αντικατάσταση της λέξης “quick”, ο δημιουργός επέλεξε τη χαβανέζικη λέξη, ενθυμούμενος το Wiki Wiki Shuttle, το τακτικό δρομολόγιο που συνέδεε τους τερματικούς σταθμούς του Διεθνούς

Αεροδρομίου της Honolulu. Με την επικράτηση του όρου Wiki, προτάθηκε η επέκταση των αρχικών στη φράση “**What I Know Is**”.

Από το 2000 και μετά τα Wikis έτυχαν ευρείας αποδοχής από τις επιχειρήσεις με κύριες εφαρμογές την υποστήριξη επικοινωνίας μεταξύ των μελών διαχείρισης ενός έργου (project communication), τα εσωτερικά δίκτυα επιχειρήσεων (intranets) και τη συνεργατική σύνταξη εγγράφων (κυρίως για τεχνικούς σκοπούς). Οι Adobe, Intel, IBM, Microsoft αλλά και το FBI είναι μόνο μερικά από τα παραδείγματα εφαρμογών Wikis σε επιχειρήσεις (enterprise - corporate wikis). Κατ’ αυτόν τον τρόπο επιτυγχάνεται ευκολότερη, ταχύτερη και οικονομικότερη διαμοίραση πληροφοριών μεταξύ των χρηστών.

Τα Wikis λειτουργούν πολύ απλά, στα πλαίσια ενός web browser, χωρίς να απαιτούν επιπλέον εφαρμογές ή επεκτάσεις λογισμικών προγραμμάτων. Υποστηρίζοντας μάλιστα τη συσχέτιση μεταξύ των δεδομένων που καταχωρείται στις ιστοσελίδες τους, κάνουν χρήση της υπερσύνδεσης περιεχομένου (όπως για παράδειγμα ανακατεύθυνση, μέσω υπερσυνδέσμου, από μια λέξη που υπάρχει σε μια εγγραφή, στην αντίστοιχη εγγραφή που αφορά αυτή τη λέξη) υποδεικνύοντας απλοϊκά (ενδεχομένως με αλλαγή χρώματος ή υπογράμμιση) αν υπάρχουν επιπλέον διαθέσιμες εγγραφές για την εκάστοτε λέξη. Θα πρέπει όμως να ξεκαθαριστεί ότι τα Wikis δεν απευθύνονται σε εξεζητημένους χρήστες, αλλά αντιθέτως, επιχειρούν να ενεργοποιήσουν και να εμπλέξουν τους επισκέπτες τους σε μια διαρκή διαδικασία ενημέρωσης, δημιουργίας και συνεργατικής τροποποίησης των ιστοσελίδων που χρησιμοποιούν.

Έντονη συζήτηση γίνεται γύρω από το θέμα της αξιοπιστίας και της ασφάλειας των εγγραφών στα Wikis, με τους επικριτές να υποστηρίζουν ότι η ελευθερία επεξεργασίας των εγγραφών από τον κάθε χρήστη τελικώς επιβαρύνει το σύστημα στο σύνολό του, ενώ η ίδια ακριβώς ελευθερία, σύμφωνα με τους θιασώτες, είναι αυτή που εξασφαλίζει έγκαιρη και έγκυρη αντιμετώπιση οποιασδήποτε προβληματικής περίπτωσης επεξεργασίας. Ειδικότερα, αναφορικά με το ζήτημα της αξιοπιστίας, είναι λογικό ότι δεδομένης της ελευθερίας που εξασφαλίζεται στους χρήστες να παρεμβαίνουν στο περιεχόμενο, είναι πιθανό να παρατηρηθούν περιπτώσεις ανάρμοστης ή εσφαλμένης επεξεργασίας ή τροποποίησής του. Εκεί όμως ακριβώς καλείται να δράσει η κοινότητα που δημιουργείται στα πλαίσια του Wiki, η οποία στη συντριπτική πλειοψηφία περιπτώσεων παραβατικής συμπεριφοράς ενεργεί άμεσα για την επαναφορά της ισορροπίας. Όσο μάλιστα πιο μεγάλη και προσηλωμένη είναι η κοινότητα του Wiki, τόσο πιο άμεσα αντιμετωπίζονται τέτοια περιστατικά. Με την ίδια φιλοσοφία λειτουργεί και το θέμα της ασφάλειας. Οι δημιουργοί των Wikis επενδύουν στην άμεση, συλλογική και έγκυρη διόρθωση κάποιας «ζημιάς ή βλάβης», με την ευρεία έννοια, παρά στην προσπάθεια αποτροπής της.

Κλείνοντας, μια σύντομη αναφορά στην έκταση των Wikis στον Παγκόσμιο Ιστό, περιλαμβάνει, σύμφωνα με τον WikiIndex.org (που αποτελεί ένα Wiki για τα Wikis), περισσότερες από 4.300 σελίδες που σχετίζονται με Wikis, χρήστες και πληροφορίες Wiki. Σε αυτές περιλαμβάνονται περισσότερα από 3.700 αρχεία, ενώ έχουν διενεργηθεί περισσότερες από 10.208.260 επισκέψεις.

Στην ουσία, η «Ταξινόμηση από Ανθρώπους» (Folksonomy) είναι το αποτέλεσμα της «Κοινωνικής Σήμανσης» (Social Tagging) και η καινοτομία έγκειται στο γεγονός ότι η πληροφορία αποκτά δομή και κατηγοριοποιείται όχι μόνο από τον αρχικό δημιουργό της, αλλά και από κάθε μεμονωμένο τελικό χρήστη. Έτσι στις περιπτώσεις της Συνεργατικής Σήμανσης επιλέγονται ελεύθερα «λέξεις - κλειδιά» σύμφωνα με τη θεώρηση των χρηστών, οι οποίες θα χρησιμεύσουν στη μεταγενέστερη αναζήτηση κι ανεύρεση πληροφορίας. Πρόκειται δηλαδή για μια ταξινόμηση που προκύπτει από τους ίδιους τους χρήστες, χωρίς μάλιστα να απαιτείται συμμόρφωση σε κάποιους κανόνες.

Η δημοτικότητα κι η υιοθέτηση των τεχνολογιών και των πρακτικών των Folksonomies καθιερώθηκε το 2004, αν και σε πρώιμο στάδιο εμφανίζονται ήδη από το 2002, όταν αποτέλεσαν εφαρμογή κοινωνικών λογισμικών (social software) πρωτίστως για σήμανση φωτογραφιών και σελιδοδεικτών (social bookmarking). Γίνεται αντιληπτό ότι και η τεχνολογία της Κοινωνικής Σήμανσης αποτελεί υπηρεσία του Web 2.0, επιτρέποντας στον κάθε χρήστη να κατηγοριοποιεί και να ανευρίσκει πληροφορίες, στον Παγκόσμιο Ιστό χωρίς να είναι ειδικός και χωρίς να απαιτούνται εξειδικευμένες γνώσεις.

Στα πλαίσια οπτικοποίησης των Folksonomies συνήθη πρακτική αποτελεί η παράθεση ενός «ενεργού σύννεφου από tags». Σε αυτό το σύννεφο παρουσιάζεται υπό μορφή λέξεων ή σύντομων φράσεων το περιεχόμενο και η ουσία του, διαβαθμισμένων κατά μέγεθος και πιθανόν με αλφαβητική σειρά, υποδηλώνοντας τη σημαντικότητα ή τη δημοτικότητά τους. Ο χρήστης επιλέγοντας κάποιον όρο, μεταφέρεται μέσω υπερσυνδέσμου στο αντίστοιχο σημείο της ίδιας ή κάποιας άλλης ιστοσελίδας, στην οποία γίνεται αναφορά στο συγκεκριμένο όρο.

Με τη χρήση των Folksonomies η αναζήτηση κι ανεύρεση πληροφοριών γίνεται ταχύτερη κι απλούστερη με το αποτέλεσμα της σύμπραξης κάθε μεμονωμένου χρήστη, προάγοντας έτσι τη φιλοσοφία του συνεργατικού Web 2.0 ως πεδίο κοινοπραξίας. Δε λείπουν όμως και οι επικριτές της ιδέας, που υποστηρίζουν ότι η ελευθερία που προσδίδεται στο κάθε χρήστη να εισάγει κάποιο tag (σηματοδοτώντας έτσι το περιεχόμενο κάποιας πληροφορίας) ενδεχομένως να οδηγήσει στην εισαγωγή αδόκιμου ή παραπλανητικού όρου που θα οδηγήσει σε ασυνεπή αποτελέσματα. Καθώς λοιπόν οι ετικέτες επιλέγονται αυθαίρετα και κατά βούληση του χρήστη, περιπτώσεις εννοιολογικών συνωνυμιών, ομωνυμιών και πολυσημιών είναι πιθανό να ανακύψουν, καθιστώντας αναποτελεσματικές τις προσπάθειες κατηγοριοποίησης κι αναζήτησης πληροφοριών.

2.5.4. - Ιστοχώροι Κοινωνικής Δικτύωσης (Social Network Sites SNS)

Ένας **Ιστοχώρος Κοινωνικής Δικτύωσης (SNS)** αποσκοπεί στη δημιουργία μιας Διαδικτυακής Κοινότητας (Online Community) αποτελούμενης από μέλη με κοινά ενδιαφέροντα ή δραστηριότητες, που επιθυμούν να έρθουν σε επαφή με άλλα μέλη των οποίων τα ενδιαφέροντα και τις δραστηριότητες ενδιαφέρονται επίσης να ανακαλύψουν. Από την απαρχή της εμφάνισής τους το 1994, τα Κοινωνικά Δίκτυα χαίρουν μεγάλης αποδοχής από ευρύ ποσοστό των χρηστών του Διαδικτύου, ενώ ο αριθμός αυτός αυξάνει εκθετικά τα τελευταία χρόνια.

Οι κύριοι τύποι Υπηρεσιών Κοινωνικής Δικτύωσης (Social Network Services) περιλαμβάνουν μεταξύ άλλων ευρετήρια διαφόρων ειδών (όπως για παράδειγμα παλιοί συμμαθητές ή μέλη ενός οργανισμού) αλλά και υποστηρικτικά μέσα για τη διασύνδεση με φίλους (όπως προσωπικές αυτοπεριγραφικές σελίδες ή υπηρεσίες επικοινωνίας), δομικά στοιχεία που υιοθετεί η πλειονότητα των Ιστοχώρων Κοινωνικής Δικτύωσης. Στα SNS μεμονωμένοι χρήστες ή ακόμα ολόκληροι οργανισμοί ή επιχειρήσεις έχουν τη δυνατότητα να συνθέσουν ένα προσωπικό δημόσιο προφίλ στο οποίο θα έχουν πρόσβαση τόσο οι ίδιοι όσο και η «λίστα» των μελών που εκείνοι θα έχουν επιλέξει.

Ιδιαίτερο χαρακτηριστικό των Ιστοχώρων Κοινωνικής Δικτύωσης αποτελεί το γεγονός ότι η ανάπτυξή τους ξεκίνησε από τους ίδιους τους χρήστες. Όπως εξηγεί ο Δρ. Γιώργος Μητακίδης, καθηγητής του Πανεπιστημίου Πατρών, «τα κοινωνικά δίκτυα γεννήθηκαν από κάτω προς τα πάνω, δεν είναι προϊόν κάποιας εταιρείας και άλλαξαν το Διαδίκτυο ποικιλοτρόπως». Από ένα σημείο και μετά έγιναν δημοφιλή κι απέκτησαν την ιδιαίτερη δυναμική που τα χαρακτηρίζει σήμερα, καθιστώντας τη σημαντικότερη διαφορά μεταξύ των Web 1.0 και Web 2.0. Ένα από τα ερωτήματα που δημιουργούνται εξετάζοντας τα Κοινωνικά Δίκτυα είναι ο βαθμός στον οποίο συνιστούν μια μόδα, μια τάση της εποχής και πρόσκαιρο φαινόμενο της εξελικτικής πορείας του ιντερνέτ. Τα μέχρι τώρα στοιχεία ωστόσο, δεν αφήνουν περιθώριο αμφισβήτησης. Τα SNSs συνεχώς αυξάνουν, παγιώνονται κι εξελίσσονται αποκτώντας όλο και μεγαλύτερη μερίδα φανατικών χρηστών. Έχουν ξεπεράσει πλέον τον στενό κλοιό μιας περιορισμένης αριθμητικά ομάδας και αφορούν άτομα κάθε ηλικίας λιγότερο ή περισσότερο εξοικειωμένα με την τεχνολογία του Διαδικτύου.

Ενδεικτικά αναφέρουμε στο σημείο αυτό μια κατηγοριοποίηση των Ιστοχώρων Κοινωνικής Δικτύωσης (SNS), σύμφωνα με τη φιλοσοφία των υπηρεσιών που παρέχουν. Ιδιαίτερα δημοφιλή είναι τα Δίκτυα Κοινωνικοποίησης κι Επικοινωνίας με έμφαση στις προσωπικές σχέσεις (Person - oriented) όπως το MySpace, Facebook, Friendster, Bebo, Orkut, WindowsLiveSpace και το Hi5. Ακολουθούν οι Επαγγελματικές Κοινότητες (Professional Communities) που αφορούν επιχειρήσεις κι επαγγελματίες, όπως το LinkedIn και το itLinkz. Σημαντικής αποδοχής είναι τα Δίκτυα Οπτικοακουστικού Υλικού (Media - oriented) όπως το YouTube στη προβολή βίντεο και το Flickr ως τεράστια βάση δεδομένων εικόνων. Ως επιτομή της νέας τάσης που καθιστά τη χρήση ιστοχώρων Κοινωνικής Δικτύωσης μέσω κινητών τηλεφώνων ιδιαίτερα δημοφιλή, το Twitter έρχεται να καθιερωθεί ως ο πλέον χαρακτηριστικός “micro - blog” Ιστοχώρος αυτού του είδους. Τέλος, συναντάμε Ψηφιακούς Κόσμους Εικονικής Πραγματικότητας (Virtual- World Communities) όπως το SecondLife και το Warcraft που αναμένεται να εδραιωθούν ακόμα περισσότερο τα επόμενα χρόνια καθώς θα σημειωθούν σημαντικές εξελίξεις στην τεχνολογία της Εικονικής 3D Αναπαράστασης.

2.5.5. - Podcasts και Videocasts

Τα **Podcasts** είναι ψηφιακές ηχητικές εγγραφές που ενδεχομένως συνοδεύονται από βίντεο, για τα οποία παρέχεται η δυνατότητα μέσω ειδικού λογισμικού να αποθηκευτούν αυτόματα ως αρχεία για τον κάθε μεμονωμένο χρήστη. Διαφοροποιούνται δηλαδή από ένα κοινό ηχητικό αρχείο στο σημείο του τρόπου αποθήκευσής του, καθώς κάθε χειρισμός σε αυτή τη περίπτωση θα πρέπει να γίνει

χειροκίνητα (McBride & Wingfield, 2005). Τα Podcasts προέκυψαν ως συγκερασμός των κοινών ηχητικών αρχείων και των ιστολογίων (blogs). Όπως έχουμε ήδη αναφέρει, τα blogs είναι μια νέα μορφή επικοινωνίας, όπου ο δημιουργός τους μπορεί να δημοσιεύσει οποιαδήποτε πληροφορία και να τη διαθέσει προς ανάγνωση και σχολιασμό σε όλο το κοινό του Παγκόσμιου Ιστού. Μέσω της τεχνολογίας των Podcasts, τα blogs μπορούν να εμπλουτιστούν με ηχητικά δεδομένα τα οποία εν συνεχεία είναι διαθέσιμα για κάθε χρήστη που είτε επιθυμεί να τα αρχειοθετήσει στον προσωπικό του υπολογιστή, είτε να τα αναπαράγει μέσω ενός mp3 player.

Η απαρχή της τεχνολογίας των Podcasts γίνεται το 2003 όταν ένας ραδιοφωνικός παραγωγός ο Christopher Lydon χάνει την εκπομπή στο ραδιοφωνικό σταθμό που δούλευε και διοχετεύοντας την ανάγκη του να «ακούγεται» στο κοινό του φτιάχνει το πρώτο audio blog για το BloggerCon του Harvard το 2003, ύστερα από προτροπή του Dave Winer. Η δημοτικότητα αυτού του τύπου των blogs, που εμπλουτίζονται με Podcasts, άρχισε να αυξάνεται όταν όλο και περισσότεροι χρήστες εξοικειώθηκαν με τη νέα αυτή τεχνολογία και συνειδητοποίησαν τα πλεονεκτήματα που τη συνοδεύουν. Τα βασικά τους πλεονεκτήματα εντοπίζονται αφενός στην απλή τους χρήση κι αφετέρου στη δωρεάν δυνατότητα για αποθήκευση (download) στον προσωπικό υπολογιστή κάθε χρήστη.

Τα Podcasts που συνοδεύονται από κάποια εγγραφή βίντεο ονομάζονται **Videocasts** και είναι ακόμα πιο ελκυστικά για τους χρήστες. Οι δυνατότητες χρήσης τους είναι απεριόριστες, καθώς μπορεί να χρησιμοποιηθούν για απλή δημοσίευση κάποιου θέματος προσωπικού ενδιαφέροντος αλλά ακόμα και για εκπαιδευτικούς λόγους. Θα μπορούσαμε να πούμε ότι τα Podcasts είναι μια τεχνολογία που μπορεί να ακουστεί οπουδήποτε (στο αυτοκίνητο, στο κινητό τηλέφωνο, στον υπολογιστή) και να εμπεριέχει οτιδήποτε (από ένα μάθημα Ισπανικών ως ένα επιμορφωτικό πρόγραμμα για τους εργαζομένους μιας επιχείρησης).

Είναι εμφανές λοιπόν ότι το Podcasting προσφέρει μια σειρά από εφαρμογές στη καθημερινή ζωή, περιλαμβάνοντας μεταξύ άλλων ραδιοφωνικές εκπομπές για φίλους, διαφημίσεις και γενικότερα ειδήσεις για το ευρύ κοινό. Υποστηρίζει όμως και επιχειρηματικούς σκοπούς, όπως εκπαίδευση προσωπικού, επικοινωνία μεταξύ των μελών μιας εταιρίας, διοργάνωση εκδηλώσεων κι επαφή με τους πελάτες. Από επιμορφωτική σκοπιά, το Podcasting μπορεί να χρησιμοποιηθεί ως μέσο διευκόλυνσης και αύξησης της ικανότητας εκμάθησης.

Ανακεφαλαιώνοντας λοιπόν, τόσο το Podcasting όσο και το Videocasting είναι αδιαμφισβήτητα πολύτιμα εργαλεία για επιχειρήσεις, επαγγελματίες αλλά και για κάθε απλό χρήστη του Διαδικτύου. Τα επόμενα χρόνια αναμένεται να αναπτυχθούν περισσότερες εφαρμογές τέτοιου τύπου, καθιστώντας τη νέα μορφή του Παγκόσμιου Ιστού ακόμα πιο συναρπαστική κι ενδιαφέρουσα. Αξίζει να σημειωθεί ότι ακόμα βρισκόμαστε σε εμβρυακό στάδιο κι ίσως δε μπορούμε να αναλογιστούμε την επανάσταση που θα φέρει η σωστή αξιοποίηση των Podcasts για επιχειρήσεις κι οργανισμούς.

2.5.6. - Εικονικοί Κόσμοι (Virtual Worlds)

Από την πρώτη εμφάνισή τους το 1974 οι **Εικονικοί Κόσμοι** (Virtual Worlds) έχουν απαλλαχτεί του καθαρά ψυχαγωγικού χαρακτήρα και χρησιμοποιούνται πλέον τόσο από επιχειρήσεις όσο και για εκπαιδευτικούς σκοπούς. Στη παρούσα φάση οι Ιστοχώροι Εικονικής Πραγματικότητας βρίσκουν εφαρμογές ως Ψηφιακές Τάξεις, Δικτυακά επιμορφωτικά προγράμματα νέων υπαλλήλων σε επιχειρήσεις αλλά ακόμα κι ως σύγχρονα εργαλεία μάρκετινγκ που απευθύνονται στο Web 2.0 καταναλωτικό κοινό. Οι αναλυτές πιστεύουν ότι οι Εικονικοί Κόσμοι θα εδραιωθούν ως ο νέος καθολικός τρόπος επικοινωνίας. Οι επιχειρήσεις που θα καταφέρουν άμεσα να αξιοποιήσουν και να κατανοήσουν τις δυνατότητες που συνοδεύουν αυτού του τύπου τους ιστοχώρους, αναμένεται να ηγηθούν των υπολοίπων στον τομέα τους.

Η έννοια του Εικονικού Κόσμου ξεκίνησε από τον Steve Colley το 1974 και το παιχνίδι του “Maze War” , 3-Διαστάσεων. Ακολούθησαν πολλές εφαρμογές τέτοιου τύπου με τις πιο σύγχρονες να δημιουργούν ένα σκηνικό Εικονικής Πραγματικότητας που δίνουν τη δυνατότητα στον παίκτη- χρήστη να φτιάξει το δικό του Ψηφιακό Χαρακτήρα (Avatar). Στα πλαίσια των Ιστοχώρων Εικονικής Πραγματικότητας αναπτύσσεται μια ευρεία γκάμα εφαρμογών όχι μόνο απλού ψυχαγωγικού χαρακτήρα, αλλά αμιγώς εμπορικού ή και εκπαιδευτικού. Ο επιχειρηματικός κόσμος λοιπόν, βρίσκει σε αυτού του τύπου τους Ιστοχώρους, μια νέα σειρά δυνατοτήτων και προοπτικών κερδοφορίας.

Παρόλα τα οφέλη που έχει προσφέρει αυτή η ιδέα των Εικονικών Κόσμων πρέπει να σημειώσουμε ότι τη πραγματική διάσταση των δυνατοτήτων που ανακύπτουν θα τη βιώσουμε τα επόμενα χρόνια, καθώς για την ώρα συναντώνται λειτουργικά προβλήματα. Σε πολλές ιστοσελίδες λόγω κυρίως του μεγάλου πλήθους των επισκεπτών, παρουσιάζονται «καθυστερήσεις» (slowdowns) που δυσχεραίνουν τη χρήση των εφαρμογών. Αλλά ακόμα και σε επίπεδο ασφάλειας κι ελέγχου, ακριβώς επειδή η ανάπτυξη τόσο σε μέγεθος όσο και σε αξία, αυτού του τύπου των Ιστοχώρων είναι ραγδαία, υπάρχουν αρκετά ζητήματα προς επίλυση προκειμένου να είμαστε σε θέση να μιλήσουμε για μια τελειοποιημένη γενιά υπηρεσιών Εικονικής Πραγματικότητας.

Η τεχνολογική αυτή εξέλιξη αναμένεται να ακολουθήσει την ταχύτητα εξέλιξης του ίδιου του Διαδικτύου, καθώς κολοσσιαίες εταιρίες όπως η Coca - Cola, η IBM και η Toyota επενδύουν ήδη τεράστια χρηματικά ποσά στην Έρευνα κι Ανάπτυξη που αφορά τους Ιστοχώρους Εικονικής Πραγματικότητας. Με την επιστήμη να προοδεύει ταχύτατα, υπολογίζεται ότι ως το τέλος του 2011 τουλάχιστον το 80% των εταιρικών χρηστών θα έχουν έστω και μια παρουσία σε κάποιον Εικονικό Κόσμο, εκμεταλλευόμενοι την προοπτική βελτίωσης των επιχειρηματικών διαδικασιών, ενίσχυσης της συνεργατικότητας, ενδυνάμωσης των δημοσίων σχέσεων και οπτικοποίησης των τάσεων της αγοράς μέσα από μοντέλα 3-Διαστάσεων.

Η πρώτη γνωστή πλατφόρμα αυτού του είδους είναι το SecondLife που γνωρίζει όμως παρακμή τα τελευταία χρόνια, καθώς πιο εξελιγμένες κι απλές στη χρήση Ιστοσελίδες έχουν εμφανιστεί στο προσκήνιο κι απειλούν να το εκτοπίσουν εντελώς. Ενδεικτικά αναφέρουμε τα Club Penguin, Webkinz, Haboo Hotel, There, Kaneva και Languna Beach. Παράλληλα, τεράστια ονόματα όπως η Microsoft, το Google και το Yahoo! ετοιμάζονται να προωθήσουν τον δικό τους Ιστοχώρο Εικονικής Πραγματικότητας στο άμεσο μέλλον.

2.5.7. - Βάσεις συγκέντρωσης κι ανταλλαγής φωτογραφιών (Photo Sharing)

Η πρώτη εμφάνιση ιστοσελίδων που αφορούσαν φωτογραφικό υλικό συναντάται στα μέσα της δεκαετίας του 1990 και σε πρώιμο στάδιο υποστήριζε την δυνατότητα τελικής εκτύπωσης ψηφιακών φωτογραφιών. Οι δυνατότητες πολλαπλασιάστηκαν ακολούθως και περιελάμβαναν μεταξύ άλλων την επεξεργασία εικόνας, δυνατότητα αγοράς προϊόντων σχετικών με το φωτογραφικό υλικό, δημόσια και ιδιωτική ανταλλαγή εικόνων, και γενικότερα Κοινωνικά Δίκτυα που υποστηρίζουν ένα σύνολο εφαρμογών κι υπηρεσιών σχετικών με φωτογραφίες.

Ένα από τα πιο δημοφιλή **Κοινωνικά Δίκτυα Συλλογής κι Ανταλλαγής Εικόνων** (Photo- Sharing Website) είναι το Flickr, που σχεδιάστηκε από το Yahoo! το 2005. Διατίθεται σε 8 γλώσσες και προσφέρει δυνατότητα παγκόσμιας διακίνησης εικόνων και φωτογραφιών. Υπολογίζεται ότι κάθε ώρα περίπου 8.000 φωτογραφίες προστίθενται στην ήδη τεράστια βάση δεδομένων, δημιουργώντας έτσι έναν ανυπολόγιστης αξίας πυρήνα γνώσης κι επικοινωνίας. Ειδικά για επιμορφωτικούς σκοπούς, η εικόνα διευκολύνει σημαντικά και εμπλουτίζει τη διαδικασία εκμάθησης, οπότε αυτού του τύπου οι ιστοχώροι αναμένεται να εξαπλωθούν σε σχολεία και βιβλιοθήκες.

Οι μελλοντικές τάσεις περιλαμβάνουν Photo- Sharing Sites με περισσότερες δυνατότητες επεξεργασίας εικόνων τύπου PhotoShop που θα κάνουν την εμπειρία χρήσης τους πιο ενδιαφέρουσα, διασκεδαστική και συνεργατική. Επιχειρήσεις κι οργανισμοί θα ευνοηθούν ακολούθως αν αναπτύξουν σωστά μια Photo- Sharing παρουσία στο Διαδίκτυο είτε για ιδιωτική - ενδοεταιρική χρήση, είτε για μια αμεσότερη επικοινωνία με το πελατειακό κοινό.

2.5.8. - Διαδικτυακές Συνεδριάσεις (Web Conferencing & Electronic meetings)

Μπορούμε να αναζητήσουμε τις ρίζες των **Διαδικτυακών Συνεδριάσεων** (Web Conferencing) πριν την εμφάνιση του Διαδικτύου και του Παγκόσμιου Ιστού, σε ένα ηλεκτρονικό πρόγραμμα με το όνομα PLATO που δημιουργήθηκε το 1960, στο Πανεπιστήμιο Illinois από τον καθηγητή Don Bitzer. Ο Wooley το 1994 αναφέρει ότι το PLATO «υποστήριζε την ανταλλαγή άμεσων μηνυμάτων, κοινή χρήση οθόνης (remote screen sharing) και διαδικτυακά παιχνίδια αποτελώντας έτσι την αφετηρία γι' αυτό που ονομάζουμε σήμερα Κοινωνικά Δίκτυα».

Οι ηλεκτρονικές Διαδικτυακές Συνεδριάσεις εξελίχθηκαν με το πέρασμα των χρόνων και μπορούν να οριστούν σήμερα ως ένας συνεργατικός τρόπος επικοινωνίας μέσω ενός δικτύου σε πραγματικό χρόνο (Peer-to- peer meetings). Περιλαμβάνουν δυνατότητα άμεσης παρουσίασης σε μορφή βίντεο και κειμένου, ανταλλαγής μηνυμάτων γραπτών ή φωνητικών, διαμοίρασης αρχείων κι εφαρμογών, κοινή χρήση οθόνης και φυσικά υπηρεσίες ασφάλειας, όπως ακριβώς συμβαίνει σε μια εταιρική συνεδρίαση.

Για το ηχητικό κομμάτι η κυρίαρχη παρούσα τάση είναι να χρησιμοποιείται η τηλεφωνική οδός, αναμένεται όμως άμεσα να ενισχυθεί η τεχνολογία του VoIP, που επιτρέπει τον έλεγχο του ομιλητή μέσω της οθόνης κι θωρακίζει καλύτερα το

σύστημα από θέμα ασφάλειας. Όσον αφορά την επέκταση των υπηρεσιών αυτού του είδους υπάρχουν τρεις επιλογές: το μοντέλο «Λογισμικό ως Υπηρεσία» (SaaS Model- Software as a Service) κατά το οποίο ο τελικός χρήστης αποκτά πρόσβαση στο διαδίκτυο μέσω του συστήματος του προμηθευτή της υπηρεσίας, το μοντέλο “On-Premise” όπου ο χρήστης έχει αγοράσει την άδεια χρήσης της υπηρεσίας κι έχει τη δυνατότητα να τη χρησιμοποιεί στο δικό του σύστημα και τέλος το “Blended model” που είναι συγκερασμός των ανωτέρω. Οι περισσότερες επιχειρήσεις χρησιμοποιούσαν ως τώρα το πρώτο μοντέλο, παρατηρείται όμως μια τάση προς το δεύτερο “On- Premise” μοντέλο καθώς πλεονεκτεί τόσο από άποψη κόστους όσο και ασφάλειας.

Σύμφωνα με τον Mann, μέχρι το τέλος του 2010, το 75% των εταιρικών χρηστών θα έχει πρόσβαση σε Διαδικτυακές Συνεδριάσεις, βελτιστοποιώντας έτσι τα κέρδη της επιχείρησης (λόγω μείωσης κόστους σε ταξίδια) αλλά και την αποδοτική επικοινωνία και συνεργασία των μελών της. Οι συνεδριάσεις θα μπορούν να γίνονται πλέον οπουδήποτε και να περιλαμβάνουν εταιρικά μέλη από ολόκληρο τον κόσμο, γεγονός που αποτελεί σημαντικό πλεονέκτημα για τις πολυεθνικές εταιρίες.

Από εκπαιδευτική άποψη οι Διαδικτυακές Συνεδριάσεις έχουν δώσει τη δυνατότητα σε οποιονδήποτε άνθρωπο το επιθυμεί να παρακολουθήσει μαθήματα μέσω Ανοικτού Πανεπιστημίου, έχοντας μια άμεση και αποδοτική εμπειρία εκμάθησης. Δίνεται μάλιστα η δυνατότητα για ερωτήσεις σε πραγματικό χρόνο κι επικοινωνία με καθηγητές ή συμφοιτητές, όπως ακριβώς συμβαίνει σε ένα κανονικό Πανεπιστήμιο. Ένα επιτυχές πρόγραμμα τέτοιου τύπου είναι η ενσωμάτωση του λογισμικού “WiredRed’s e/Pop web conferencing software” στο Kent College Preparatory School, προκειμένου να ενισχύσει την επικοινωνία μεταξύ επιχειρήσεων και πανεπιστημίου για την επίλυση σύνθετων προβλημάτων, μειώνοντας αντιστοίχως περιττά έξοδα για ταξίδια.

Για να παρουσιάσουμε μερικά αριθμητικά δεδομένα, σύμφωνα με έρευνα (Gartner 2007) η αγορά των προγραμμάτων για Διαδικτυακές Συνεδριάσεις ανερχόταν σε 1,13 δισεκατομμύρια δολάρια το 2007, με αναμενόμενο ετήσιο ποσοστό αύξησης 19,5% τα επόμενα χρόνια. Συμπεραίνουμε επομένως, πως σε μια εποχή παγκοσμιοποίησης που διανύουμε, οι Ηλεκτρονικές Συνεδριάσεις θα αποτελούν καθημερινή πρακτική, δεδομένης της ανάγκης για επικοινωνία των εργαζομένων της επιχείρησης και των συνεργατών ανά τον κόσμο.

2.6. - Ιδιωτικότητα στο Web 2.0 (Security Issues)

Ένας βασικός αποτρεπτικός παράγοντας υιοθέτησης των νέων κοινωνικών υπηρεσιών τόσο από ιδιώτες όσο κι από επιχειρήσεις δεν είναι άλλος από την Ιδιωτικότητα και το βαθμό στον οποίο αυτή διακυβεύεται κατά την έκθεση στον Νέο Παγκόσμιο Ιστό. Ίσως το σημαντικότερο γεγονός που πρέπει να έχουμε υπόψη κατά τη χρήση του Διαδικτύου είναι πως οτιδήποτε δημοσιεύεται δικτυακά θα παραμείνει για πάντα εκεί, κατά το λατινικό *Scripta Manent*. Εργοδότες, ανταγωνιστικές εταιρείες αλλά ακόμα και η Κυβέρνηση μπορούν να συλλέξουν πληροφορίες για επιχειρήσεις, οργανώσεις, αλλά και για τον κάθε χρήστη χωριστά καθώς περιηγείται στο διαδίκτυο.

Το φαινόμενο αυτό γίνεται ιδιαίτερα έντονο όσον αφορά στη στοχευόμενη διαφήμιση. Αξιοσημείωτο είναι το παράδειγμα των **στοχευόμενων διαφημίσεων** “Phorm”, από την British Telecom την περίοδο 2006-2007, που είχε ως αποτέλεσμα η Ευρωπαϊκή Επιτροπή να δεχτεί χιλιάδες καταγγελίες από τους χρήστες της εταιρείας, οι οποίοι δήλωναν ότι η καταγραφή των online δραστηριοτήτων τους για διαφημιστικούς σκοπούς έγινε εν αγνοία τους. Ως αποτέλεσμα, η Κομισιόν κάλεσε τη Μεγάλη Βρετανία να συμμορφωθεί με την κοινοτική οδηγία για την προστασία των προσωπικών δεδομένων στις ηλεκτρονικές επικοινωνίες και να αναθεωρήσει την εθνική νομοθεσία της σε αυτή την κατεύθυνση, απειλώντας τη χώρα ακόμη και με προσφυγή στο Ευρωπαϊκό Δικαστήριο.

Παρόμοια υπηρεσία στοχευόμενης διαφήμισης, ωστόσο, έθεσε σε λειτουργία και η Google. Βέβαια, η εταιρεία ανακοίνωσε ότι η online παρακολούθηση δεν θα αφορά ευαίσθητα προσωπικά δεδομένα που αφορούν θρησκευτικές πεποιθήσεις, σεξουαλικό προσανατολισμό, θέματα υγείας, κ.λπ. Επιπλέον, η πρόσφατη δημιουργία της μηχανής αναζήτησης rip1.com φαίνεται να περιορίζει και ίσως να θέτει σε κίνδυνο τα επίπεδα της ιδιωτικότητας. Αρκεί να αναφέρουμε ότι η rip1 αποτελεί μηχανή αναζήτησης μόνο για ανθρώπους. Αξίζει να σημειωθεί ότι σύμφωνα με κοινοτική οδηγία της Ε.Ε όλοι οι πάροχοι ηλεκτρονικών υπηρεσιών είναι υποχρεωμένοι να κρατούν αρχεία με τα e-mail και τις ιστοσελίδες που επισκέπτονται οι συνδρομητές τους για ένα χρόνο για λόγους ασφαλείας. Στο σημείο αυτό, ας ορίσουμε ποια στοιχεία νοούνται ως **προσωπικά δεδομένα**: Ως προσωπικό δεδομένο χαρακτηρίζεται οποιαδήποτε αντικειμενική ή υποκειμενική πληροφορία αφορά ένα άτομο. Αναλυτικότερα, το όνομα, η διεύθυνση, προσωπικοί αριθμοί, το γένος, η ομάδα αίματος, εισόδημα, τραπεζικοί αριθμοί, σεξουαλικές προτιμήσεις, πολιτικές πεποιθήσεις, θρησκευτικά πιστεύω είναι μερικά παραδείγματα ιδιωτικών στοιχείων.

Παρακάτω, διατυπώνονται εν συντομία τα δικαιώματα που έχουν οι χρήστες του Διαδικτύου σύμφωνα με το Μανιφέστο για την Ιδιωτικότητα στο Νέο Παγκόσμιο Ιστό 2.0:

1. Κάθε χρήστης έχει δικαίωμα να γνωρίζει αν προσωπικές πληροφορίες συλλέγονται από κάποια ιστοσελίδα και σε τι βαθμό.
2. Κάθε χρήστης έχει δικαίωμα να γνωρίζει το σκοπό για τον οποίο συλλέγονται οι πληροφορίες. Ο σκοπός οφείλει να γίνει ξεκάθαρος από την αρχή και να μην αφήνεται ως συμπέρασμα εκ του περιεχομένου της ιστοσελίδας.
3. Καθένας έχει την κυριότητα των προσωπικών του δεδομένων. Αυτό συνεπάγεται ότι οι οργανισμοί δε μπορούν να χρησιμοποιήσουν τα στοιχεία αυτά χωρίς τη συγκατάθεση του χρήστη. Επιπλέον ο χρήστης μπορεί να αλλάξει τα στοιχεία αυτά αν το επιθυμεί.

2.7. - Οι δημοφιλέστερες Web 2.0 Ιστοσελίδες Κοινωνικής Δικτύωσης

Ολοκληρώνοντας την εισαγωγική ενότητα που αφορά στο πλαίσιο λειτουργίας κι ανάπτυξης του Νέου Παγκόσμιου Ιστού, κρίνεται σκόπιμη η συνοπτική αναφορά στις πιο επιτυχημένες Web 2.0 Ιστοσελίδες Κοινωνικής Δικτύωσης που συγκεντρώνουν καθημερινά τα βλέμματα εκατομμυρίων χρηστών του Διαδικτύου.

2.7.1. - Facebook

Είναι ο δημοφιλέστερος Ιστοχώρος Κοινωνικής Δικτύωσης και ιδρύθηκε από τον Mark Zuckerberg, τριτοετή τότε φοιτητή του Harvard, στις 4 Φεβρουαρίου 2004. Η κύρια λειτουργία του σήμερα, προσανατολίζεται στην κοινωνική προσέγγιση χρηστών διαδικτυακά, μέσω της δημιουργίας ενός προσωπικού προφίλ. Το Facebook



καταμετράει περισσότερα από 400.000.000 ενεργά μέλη παγκοσμίως, που κατά μέσο όρο διασυνδέονται με 130 φίλους και ξοδεύουν παραπάνω από 55 λεπτά ημερησίως. Εκτιμάται ότι το 1/3 του πληθυσμού των ΗΠΑ διατηρεί

προφίλ στην υπηρεσία, ενώ το αντίστοιχο ποσοστό για την Ελλάδα ανέρχεται στο 22,9% του πληθυσμού (περί τα 2.515.220 μέλη), αριθμός που αυξάνεται με ρυθμούς γεωμετρικής προόδου. Με αφορμή τη δημοτικότητά του, το Facebook έχει υποστεί κριτική και κατηγορείται για θέματα που αφορούν στα προσωπικά δεδομένα των χρηστών και τις πολιτικές απόψεις των ιδρυτών του. Ωστόσο, η συγκεκριμένη ιστοσελίδα παραμένει η πιο διάσημη εφαρμογή κοινωνικής δικτύωσης.

2.7.2. - Youtube

Το YouTube είναι ένας δημοφιλής διαδικτυακός τόπος, που επιτρέπει αποθήκευση, αναζήτηση κι αναπαραγωγή ψηφιακών ταινιών. Ιδρύθηκε το 2005, ενώ το 2006 η εταιρεία εξαγοράστηκε από την Google έναντι του αστρονομικού ποσού των 1,65 δις δολαρίων. Η υπηρεσία παρέχει τη δυνατότητα σε όλους του επισκέπτες να προβάλλουν τα αποθηκευμένα βίντεο, ενώ τα εγγεγραμμένα μέλη μπορούν να αποθηκεύουν απεριόριστο αριθμό ταινιών με χρονικό όριο δέκα λεπτών το καθένα. Για κάθε εγγραφή βίντεο παρέχεται ο αριθμός των επισκεπτών που το έχουν προβάλλει, καθώς και σχόλια χρηστών προκειμένου να μπορεί να αξιολογηθεί. Ενδεικτικά, κάθε λεπτό της ώρα υπολογίζονται 24 νέες ώρες βίντεο που προστίθενται από χρήστες, αριθμός ασύλληπτος αφού για να τα παρακολουθήσει κανείς χρειάζεται 4 συνεχόμενα χρόνια.



2.7.3. - LinkedIn

Το LinkedIn είναι ένας Ιστοχώρος Κοινωνικής Δικτύωσης επαγγελματικού χαρακτήρα. Από την ίδρυσή του, το Φεβρουάριο του 2002, καταμετράει σήμερα περισσότερα από 60 εκατομμύρια συνδεδεμένα μέλη προερχόμενα από 200 περίπου διαφορετικές χώρες παγκοσμίως (τα 30 εκατομμύρια εντοπίζονται στις ΗΠΑ). Στο



πλαίσιο της υπηρεσίας μεμονωμένοι επαγγελματίες αλλά κι ολόκληρες επιχειρήσεις μπορούν να δημιουργήσουν το προσωπικό τους προφίλ που χρησιμεύει στη συνέχεια για ανεύρεση

εργασίας ή προσωπικού, ανταλλαγή επιστημονικών απόψεων και κάθε μορφή επικοινωνίας. Ο χρήστης έχει υψηλό επίπεδο ελέγχου στην ψηφιακή του ταυτότητα και στις προσωπικές του πληροφορίες. Από το Φεβρουάριο του 2008, το LinkedIn διατίθεται και σε ειδική έκδοση για κινητές συσκευές, προκειμένου να είναι ανά πάσα στιγμή διαθέσιμο στους απανταχού χρήστες του.

2.7.4. - Twitter

Το Twitter αντιπροσωπεύει μια νέα μορφή διαδραστικότητας, όπου ολόκληρη η επικοινωνία διεξάγεται με μόλις 140 χαρακτήρες, βασισμένο στη λογική του λεγόμενου micro blogging. Όπως όλα τα sites κοινωνικής δικτύωσης, έτσι και το Twitter, στα πρώτα βήματά του, «αφοσιώθηκε» κυρίως στην προσέλκυση των περισσότερων δυνατών χρηστών, παρά στην παραγωγή και εξασφάλιση κερδών,



γεγονός που έκανε τους αναλυτές να αναρωτιούνται, ως προς το πώς θα μπορούσε να «μεταμορφωθεί» σε προσοδοφόρα επιχείρηση. Η απάντηση ήρθε με τη λέξη “Twitter” να χαρακτηρίζεται ως η κορυφαία του 2009 (σύμφωνα με έρευνα του Global Language Monitor), ενώ εταιρείες που επιθυμούν να τη χρησιμοποιούν ως διαφημιστικό μέσο οφείλουν να πληρώνουν χρηματικό αντίτιμο. Από συμφωνίες που υπογράφηκαν με την Microsoft και τη Google, οι ιδρυτές της υπηρεσίας εξασφάλισαν κέρδη της τάξης των 25

εκατομμυρίων δολαρίων, ενώ από το 2006 οπότε και ιδρύθηκε υπολογίζονται περίπου έσοδο που ανέρχονται στα 155 εκατομμύρια δολάρια.

2.7.5. - Picasa

Οι ιστοσελίδες διαμοίρασης φωτογραφιών έχουν ξεκινήσει την λειτουργία τους ήδη από τις αρχές της δεκαετίας του '90. Παρόλα αυτά όμως, χρειάστηκαν μερικά χρόνια ακόμα (Φεβρουάριος 2004) για να κάνει την εμφάνισή της μια υπηρεσία με το όνομα Picasa και να ενσωματώσει τη λογική της διαμοίρασης κι ανταλλαγής φωτογραφιών σε μια Διαδικτυακή Κοινότητα. Τα τελευταία χρόνια, το Picasa έχει γίνει η δημοφιλέστερη και ταχύτερα αναπτυσσόμενη υπηρεσία

διαμοίρασης φωτογραφιών στο Διαδίκτυο (φιλοξενούσε πάνω από 4 δισεκατομμύρια φωτογραφίες τον Οκτώβριο του 2009), καθώς είναι από τις πρώτες που καθιέρωσαν τις ετικέτες σήμανσης (tags) με σκοπό τη δημιουργία δεσμών και σχέσεων μεταξύ των φωτογραφιών των ίδιων των χρηστών.



2.7.6. - Blogger

Το Blogger είναι μια υπηρεσία δημοσίευσης blog που επιτρέπει ιδιωτικά ή δημόσια blogs με χρόνο-σφραγίδες καταχώρησης. Δημιουργήθηκε από τη Pyra Labs, η οποία αγοράστηκε από την Google το 2003. Σε γενικές γραμμές, τα ιστολόγια φιλοξενούνται από τη Google σε ένα subdomain, το



blogspot.com. Μέχρι την 1η Μαΐου του 2010, το Blogger επέτρεπε στους χρήστες να δημοσιεύουν τα blogs και σε άλλες υπηρεσίες φιλοξενίας, μέσω FTP. Όλα αυτά τα blogs έπρεπε (ή ακόμα πρέπει) να μεταφερθούν στους διακομιστές της Google. Εκτός από blogspot.com domain επιτράπηκε domain μέσω προσαρμοσμένης διεύθυνσης URL. Τον Ιούλιο του 2011 ανακοινώθηκε χωρίς να επιβεβαιωθεί ότι η Google σκοπεύει να αλλάξει το όνομα της υπηρεσίας από "Blogger" σε "Blogs Google," ως μέρος ενός

ευρύτερου σχεδίου μετονομασίας ή απόσυρσης όλων των ονομάτων των υπηρεσιών που δεν περιέχουν τον όρο Google.

Κεφάλαιο 3^ο - Βασικοί στόχοι του Padgets

Κοινωνικές και τεχνολογικές τάσεις που παράγονται από Web2.0 εφαρμογές και από μέσα κοινωνικής δικτύωσης έχουν ήδη οδηγήσει σε δραματικές κοινωνικο-πολιτισμικές εξελίξεις. Οι εξελίξεις αυτές ασκούν πίεση στις κυβερνητικές οργανώσεις να καινοτομήσουν στις συναλλαγές τους με τους πολίτες, εισάγοντας νέες μορφές ατομικής συμμετοχής στα κοινωνικά και πληροφοριακά γενικότερα δίκτυα και να παρέχουν τη δυνατότητα για συμπαραγωγή, ακόμη και συν-δημιουργία μεταξύ των Κρατικών υπηρεσιών. Οι εν λόγω δυνατότητες θα πρέπει να είναι ευπρόσδεκτες από τους διαμορφωτές πολιτικής (policy makers), οι οποίοι επιδιώκουν διευκόλυνση των διαδικασιών των κρατικών υπηρεσιών που θα μπορούσαν να οδηγήσουν σε ένα νέο επίπεδο ανάπτυξης της συμμετοχικής δημοκρατίας μέσω των συγχρόνων μοντέλων Ηλεκτρονικής Διακυβέρνησης.

Το έργο PADGETS στο σύνολό του αποτελεί μια έγκυρη απάντηση στις ασάφειες που εξακολουθούν να περιβάλλουν τέτοια θέματα, παρέχοντας στους κυβερνητικούς φορείς εργαλεία Πληροφοριακής και Επικοινωνιακής Τεχνολογίας (ΤΠΕ) με την ικανότητα να αναλύουν άμορφα δεδομένα κοινωνικής ανταπόκρισης και να τα μετατρέπουν σε προβλέψεις του πιθανού αντίκτυπου των πολιτικών τους σε συγκεκριμένες κοινωνικές ομάδες που διακρίνονται κάτω από σαφώς ορισμένα κοινωνικά κριτήρια.

Πλέον, είναι κοινώς αποδεκτή άποψη ότι η διαθεσιμότητα των πλήρως λειτουργικών συμμετοχικών εργαλείων αποτελεί απαραίτητη προϋπόθεση για τη χρηστή διακυβέρνηση (Good Governance). Συνεπώς, η δημιουργία καινοτόμων εργαλείων πολιτικής, όπως τα gadgets πολιτικής, προτίθεται να παράσχουν καθοριστική συμβολή στην διάδοση των ορθών πρακτικών της χρηστής διακυβέρνησης σε όλα τα επίπεδα διακυβέρνησης στην Ευρώπη.

3.1. Βασική ιδέα

Στην ουσία, ένα padget μπορεί να οριστεί ως ένας πόρος που δημιουργήθηκε από έναν ενδιαφερόμενο πολιτικής συνήθως ως στιγμιότυπο μέσα σε μια πλατφόρμα κάποιου social media. Ένα padget παρέχει δυνατότητα αλληλεπίδρασης με άτομα τα οποία ενεργούν ατομικά και οικειοθελώς είτε εκπροσωπούν μια ομάδα κοινού ενδιαφέροντος και παρακολουθεί το περιεχόμενο και τις δραστηριότητες των χρηστών αυτών που τελικά συνδέονται με αυτό.

Από την οπτική ενός policy maker ένα padget θα πρέπει να μπορεί να πάρει τη μορφή μιας μικρής εφαρμογής που ο ίδιος θα μπορεί να εγκαταστήσει εύκολα και χωρίς εξειδικευμένες τεχνικές γνώσεις σε περιβάλλον web. Ένα padget θα πρέπει να του επιτρέπει να εισάγει σε αυτό το πολιτικό μήνυμά του, μέσω διάφορων τύπων περιεχομένου, και να διαχέονται ομοίμορφα σε όλα τα προεπιλεγμένα από αυτόν

μέσα κοινωνικής ενημέρωσης, αποφεύγοντας έτσι ένα σημαντικό βαθμό κατακερματισμού. Επιπλέον, ένα padget επιτρέπει τον policy maker να παρακολουθεί και να αναλύει τις αντιδράσεις και τις ανατροφοδοτήσεις των χρηστών στα μηνύματα της πολιτικής του.

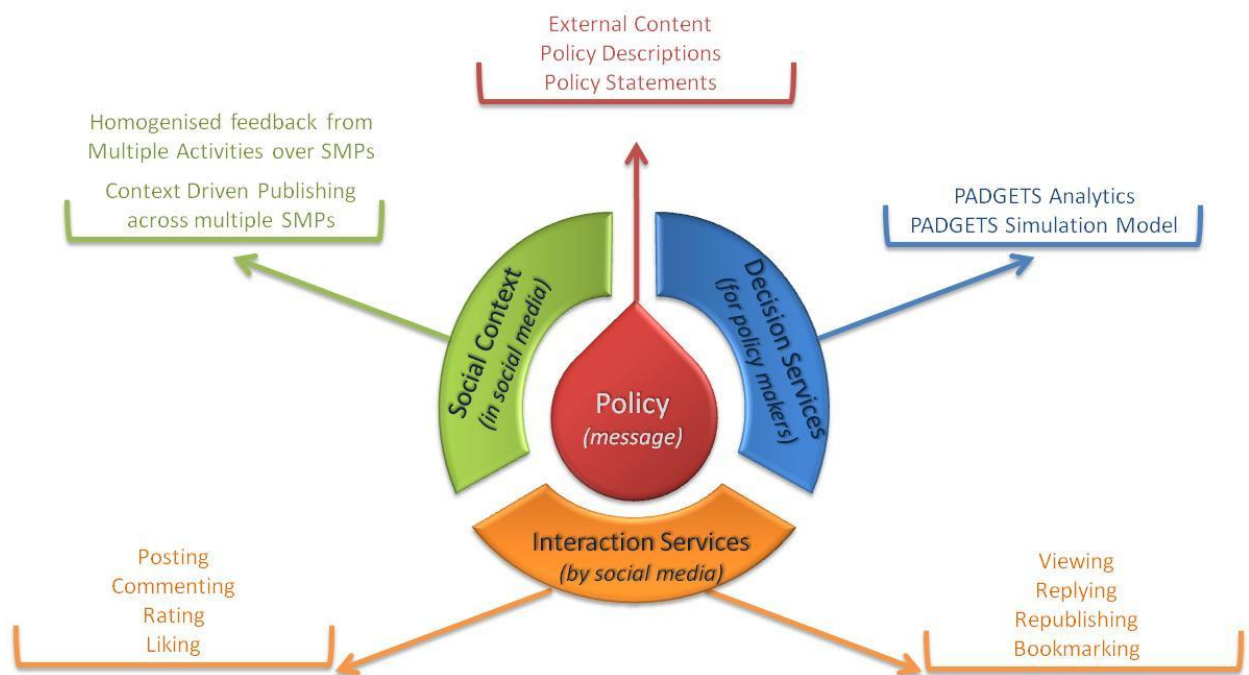
Από τη σκοπιά του τελικού χρήστη, ένα padget αναπαρίσταται ως ένα πλήρως συνδεδεμένο περιεχόμενο (τίτλο, posts, συνδέσμους, εικόνες, βίντεο) που παράγεται από έναν άλλο χρήστη (το δημιουργό του padget) και εκφράζει μια συγκεκριμένη πολιτική (καμπάνια πολιτικής). Οι τελικοί χρήστες μπορούν να έχουν πρόσβαση σε αυτές τις πληροφορίες εύκολα μέσα από τα social media της επιλογής του (π.χ. Facebook, Twitter, Youtube) και να αλληλεπιδράσει σε αυτό χρησιμοποιώντας το πρότυπο αλληλεπίδρασης που παρέχεται από το εκάστοτε social media. Ως εκ τούτου μπορούν να επιλέξουν να εκφράσουν τη γνώμη τους ή να εκτελέσουν διάφορες άλλες δραστηριότητες όπως αυτές επιτρέπονται από το συγκεκριμένο κοινωνικό μέσο απρόσκοπτα χωρίς να χρειάζεται να έχουν πρόσβαση σε οποιοδήποτε εξωτερικό περιβάλλον ή εργαλείο.

Ένα padget αποτελείται από τέσσερις κύριες συνιστώσες:

1. - Ένα μήνυμα σε οποιαδήποτε δυνατή μορφή περιεχομένου πολιτικής. Το πολιτικό μήνυμα παρουσιάζεται σε μια καθορισμένη δομή (με τη χρήση διαφορετικών τύπων περιεχομένου), προκειμένου να λαμβάνονται υπόψη το ενδιαφέρον που επιδεικνύεται από τους τελικούς χρήστες στο παρόν μήνυμα πολιτικής αλλά και επιπλέον κατά πόσο αυτό το μήνυμα θα σχολιαστεί και θα αναδημοσιευτεί τελικώς από αυτούς. Συνήθως, το πολιτικό μήνυμα θα μπορούσε να διακρίνεται σε τρία κύρια μέρη: ένα σύντομο μήνυμα πολιτικής που προκαλεί ιδιαίτερη αίσθηση, μια σύντομη πολιτική περιγραφή και μια σειρά από πιο εκτεταμένη τεκμηρίωση που μπορεί να συνδέεται με το μήνυμα σε διάφορες μορφές (κειμένου ή πολυμέσων).
2. - Μια σειρά από υπηρεσίες αλληλεπίδρασης, που επιτρέπουν στους χρήστες να αλληλεπιδρούν με το gadget της πολιτικής (να το βρίσκουν, να αποκτούν πρόσβαση στο περιεχόμενό του, να σχολιάσουν το περιεχόμενό του, να το αναδημοσιεύσουν κ.λπ.). Οι διεπαφές αυτές μπορούν να παρέχονται είτε από τις ίδιες τις κοινωνικές πλατφόρμες στις οποίες δημοσιεύεται η πολιτική εκστρατεία είτε από το ίδιο το padget όταν λαμβάνει τη μορφή μιας μικρο-εφαρμογής.
3. - Το κοινωνικό περιεχόμενο, που είναι το πλαίσιο που περιγράφει την κοινωνική δραστηριότητα σχετικά με το gadget της πολιτικής σε κάθε πλατφόρμα κοινωνικής δικτύωσης ξεχωριστά, όπου βέβαια το gadget πολιτικής είναι δημοσιευμένο. Αυτό το στοιχείο επιτρέπει στο gadget πολιτικής να παρακολουθεί το περιεχόμενο που δημιουργείται από τις δραστηριότητες των χρηστών στα μέσα κοινωνικής δικτύωσης, όπου η πολιτική εκστρατεία είναι παρούσα. Τα στοιχεία αυτά αντιπροσωπεύουν την

αλληλεπίδραση του χρήστη με το μήνυμα πολιτικής που το padget μεταφέρει, για παράδειγμα, ο αριθμός των σχολίων που έγιναν από τους χρήστες, το περιεχόμενο των σχολίων τους, ο αριθμός των ευνοϊκών θεωρήσεών τους, κ.λπ..

4. - Οι υπηρεσίες υποστήριξης απόφασης, οι οποίες προσφέρονται μέσα από δύο διαφορετικές λειτουργίες. Το PADGETS analytics, το οποίο επεξεργάζεται αριθμητικά, δημογραφικά και κειμένους δεδομένα που συλλέγονται από τη δημοσιευμένη καμπάνια του padgets, έτσι ώστε να μπορέσουν να εξαχθούν συμπεράσματα που προκύπτουν από την αλληλεπίδραση με το μήνυμα πολιτικής. Το μοντέλο προσομοίωσης PADGETS, το οποίο αναλύει και προβλέπει τη μελλοντική πορεία διάχυσης της προωθούμενης πολιτικής σε βαθμό ενδιαφέροντος και αποδοχής. Οι υπηρεσίες υποστήριξης απόφασης είναι υπεύθυνες για την παραγωγή των εξόδων με τις πληροφορίες καθολικά συγκεντρωμένες που θα παρουσιαστούν στον δημιουργό του συγκεκριμένου padget, του policy maker συνήθως.



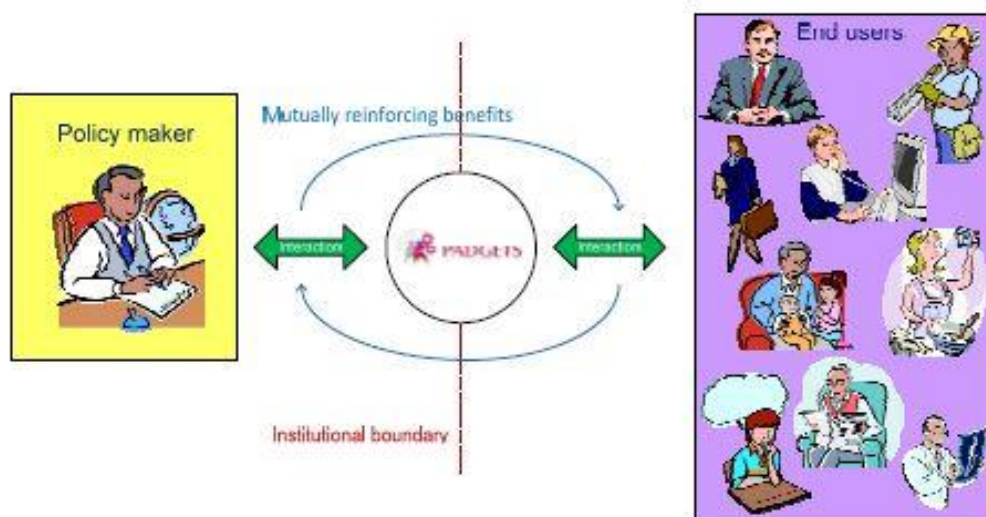
3.1. - Οι συνιστώσες του Padgets[5]

Τελικώς, τα gadgets πολιτικής δημιουργούνται, διαμοιράζονται, παρακολουθούνται και διακόπτονται μέσα από μια ενιαία εφαρμογή, που από εδώ και στο εξής θα αναφέρεται ως πλατφόρμα padgets, ή πιο απλά Padgets.

3.2. Όφελος στην πολιτική διαδικασία

Το Padgets αντιπροσωπεύει μια γέφυρα πάνω από τα θεσμικά σύνορα των κυβερνήσεων επιτρέποντάς τους να εδραιώσουν μια αμφίδρομη ροή επικοινωνίας μεταξύ των φορέων χάραξης πολιτικής και της κοινωνίας των πολιτών. Ουσιαστικά, μπορεί να αντιληφθεί με τη μείωση της απόστασης μεταξύ της διαδικασίας χάραξης πολιτικής και τις ανάγκες της κοινωνίας, τόσο από άποψη χρόνου όσο και από τα εργαλεία που απαιτούνται. Με άλλα λόγια, η χρήση των gadgets πολιτικής επιτρέπει την καλύτερη ενημέρωση της διαδικασίας λήψης πολιτικών αποφάσεων, παρέχοντας μια πιο σαφής και δυναμική εικόνα των απόψεων και των προτεραιοτήτων των διαφόρων ενδιαφερομένων. Παρέχοντας στους υπεύθυνους χάραξης πολιτικής ένα κανάλι από το οποίο μπορούν να αφουγκράζονται τη φωνή της κοινωνίας από εκεί όπου η ίδια επιλέγει να εκφράσει τις απόψεις της, το padgets δημιουργεί έναν καινοτόμο τρόπο για τη συγκέντρωση, αξιολόγηση και τελικώς απόφαση που προκύπτει από το input της ίδιας της κοινωνίας.

Το όφελος που προκύπτει από τη διαδικασία του padget μπορεί να συνοψιστεί σε μερικές λέξεις: «πολύπλευρη και πολλαπλών οφελών» (σχήμα που ακολουθεί). Με άλλα λόγια, η δράση της πλατφόρμας Padget δημιουργεί τόσο άμεσες όσο και έμμεσες θετικές επιπτώσεις για τις διάφορες κατηγορίες φορέων που ασχολούνται με τη διαδικασία («πολύπλευρη»), όπως και επίσης διάφορα οφέλη για όλους τους άμεσα εμπλεκόμενους (actors) της συνολικής διαδικασίας που προκύπτει από τη χρήση του padgets: εύκολη και άνετη συμμετοχή των πολιτών στις πολιτικές ζυμώσεις, και επίσης χρήσιμο και χαμηλού κόστους input για τους φορείς χάραξης πολιτικής («πολλαπλών οφελών»).



3.2. - Αξία που προκύπτει από το padgets για όλους τους εμπλεκόμενους

Μια καμπάνια padget αντιπροσωπεύει ένα σύνολο δραστηριοτήτων και περιεχομένου μέσω κοινωνικών μέσων μαζικής ενημέρωσης που καλύπτουν την δημιουργία, τη διανομή, την αλληλεπίδραση, την παρακολούθηση και την λήξη της ισχύος ενός (ή περισσότερων) padget ενός συγκεκριμένου πολιτικού περιεχομένου. Η δημιουργία μιας εκστρατείας padget συνεπάγεται μια σειρά από προϋποθέσεις κυριότερη εκ των οποίων μπορεί να θεωρηθεί η ισχυρή παρουσία του δημιουργού ενός padget σε επιλεγμένες κοινωνικές πλατφόρμες μέσω μαζικής ενημέρωσης. Συνεπώς, αυτό αποτελεί μια ζωτικής σημασίας απαίτηση λόγω της ίδιας της φύσης της επικοινωνίας που εφαρμόζεται πάνω από τα μέσα κοινωνικής δικτύωσης και τις πλατφόρμες δικτύωσης γενικότερα. Ως εκ τούτου, η αποτελεσματική και ταχεία διάδοση του μηνύματος πολιτικής σε τέτοια μέσα, εξαρτάται από την ύπαρξη μιας ποιοτικής και μεγάλης κοινότητας που θα έχει άμεση σχέση με τον εντολέα του μηνύματος.

Padget καμπάνιες μπορεί να ξεκινήσουν κατά τη διάρκεια μιας ή και περισσότερων φάσεων του κύκλου διαμόρφωσης της προωθούμενης πολιτικής: τον καθορισμό της ημερήσιας διάταξης, την πολιτική ανάλυση, τη χάραξη της πολιτικής, την εφαρμογή της πολιτικής και την πολιτική παρακολούθηση και αξιολόγηση. Ο σκοπός, η λειτουργία και κατά συνέπεια το όφελος της κάθε εκστρατείας μπορεί να ποικίλλει ανάλογα με το στάδιο του κύκλου της πολιτικής στην οποία η εκστρατεία έχει ξεκινήσει.

Στη διαδικασία του καθορισμού της ημερήσιας διάταξης, η καμπάνια μπορεί να χρησιμοποιηθεί ως ένα σημαντικό εργαλείο για την εκμείωση των αναγκών και των προτεραιοτήτων επιτρέποντας έτσι την αύξηση της σύγκλισης μεταξύ των δράσεων της κυβέρνησης και τις προσδοκίες των πολιτών. Κατά τη διάρκεια της φάσης της πολιτικής ανάλυσης, η καμπάνια μπορεί να χρησιμοποιηθεί για να συγκεντρώσει απόψεις των πολιτών σχετικά με τις κυριότερες πτυχές που αφορούν την πολιτική που συζητείται ή ακόμα και απόψεις σχετικές με τις πιθανές λύσεις για την εφαρμογή της. Κατά τη διαδικασία διαμόρφωσης της πολιτικής, η καμπάνια μπορεί να αποτελέσει εργαλείο για την εκτίμηση της αποδοχής του επιλεγμένου τρόπου δράσης, καθώς και για τη συλλογή χρησίων ανατροφοδοτήσεων ώστε να βελτιωθεί η αρχική διατύπωση. Κατά τη φάση της εφαρμογής της πολιτικής, η καμπάνια μπορεί να χρησιμοποιηθεί ως ένα γρήγορο και άκρως διαδραστικό εργαλείο που επιτρέπει στον policy maker να διαπιστώσει κατά πόσο η πολιτική που εφαρμόζεται έχει παράξει το επιθυμητό επίπεδο ευαισθητοποίησης και ενδιαφέροντος στις ομάδες στόχους των πολιτών. Τέλος, στη φάση της παρακολούθησης και αξιολόγησης της πολιτικής, η καμπάνια μπορεί να χρησιμοποιηθεί ως ένα εργαλείο αξιολόγησης της πολιτικής ώστε να εκτιμηθεί η αντίληψη των επιπτώσεων που δημιουργούνται από την εφαρμογή της πολιτικής δράσης, δημιουργώντας έτσι μια πολύτιμη διαδικασία μάθησης για το policy maker.

Τέλος, θα ήταν ίσως χρήσιμο να διευκρινιστεί ότι οι εκστρατείες μέσω του padget προβλέπονται επί του παρόντος ως κυρίως εργαλείο διαβούλευσης. Οι policy makers μπορεί να αποφασίσουν να επιλέξουν εναλλακτικές λύσεις που δεν είναι

πλήρως σύμφωνες με τα αποτελέσματα της διαβούλευσης. Παρ' όλα αυτά, η αύξηση του επιπέδου της διαφάνειας που εισάγει η χρήση του Padget θα τους υποχρεώνουν να παρέχουν πιο πειστικά επιχειρήματα για την επιλογή του τρόπου δράσης που πιθανόν να αποκλίνει από τη δηλωμένη γνώμη των πολιτών.

3.3. Καινοτομία όσον αφορά τις υφιστάμενες πρακτικές

Στην ακόλουθη ενότητα, εξηγείται η καινοτομία της πλατφόρμας padget. Πιο συγκεκριμένα, η καινοτομία του εξετάζεται βάσει πρακτικών που έχουν να κάνουν με τη διάδοση της ηλεκτρονικής διαβούλευσης αλλά επίσης και με τη χρήση συστημάτων υποστήριξης αποφάσεων στο δημόσιο τομέα.

3.3.1. Καινοτομία στο τομέα της ηλεκτρονικής διαβούλευσης

Η ανάπτυξη ενός πραγματικού εργαλείου επικεντρωμένο στη συμπεριφορά του πολίτη απαιτεί από τους φορείς χάραξης πολιτικής να κάνουν το πρώτο βήμα προς τη κατεύθυνση των πολιτών αντί να αναμένουν αυτοί να γνωστοποιήσουν το περιεχόμενο των απόψεων και των δραστηριοτήτων τους στους επίσημους ηλεκτρονικούς χώρους που δημιουργούνται για την ηλεκτρονική διαβούλευση και γενικότερα για την ηλεκτρονική συμμετοχή. Για αυτό το λόγο το padgets, που βασίζεται σε υπάρχουσες πλατφόρμες κοινωνικών μέσων μαζικής ενημέρωσης, αποτελούν μια καλή ευκαιρία για να προσφερθεί άνετη και χωρίς προβλήματα ηλεκτρονική συμμετοχή.

Οι κύριες καινοτομίες που εισήγαγε η πλατφόρμα PADGETS μπορούν να συνοψιστούν ως εξής: χαλάρωση των υφιστάμενων περιορισμών όσον αφορά το μέγεθος, τη συχνότητα και την ποιότητα της συμμετοχής και επιπλέον μια ολοκληρωμένη διαχείριση των πολλαπλών καναλιών των μέσων κοινωνικής δικτύωσης και ενημέρωσης. Μια σύντομη παρουσίαση αυτών παρέχεται παρακάτω.

Ας φανταστούμε ότι, προκειμένου να βελτιωθεί η διαδικασία χάραξης πολιτικής αποτελεί απαραίτητη προϋπόθεση η αύξηση του αριθμού των ατόμων που συμμετέχουν στη διαδικασία αυτή. Εάν οι ομάδες εστίασης, μικρά δείγματα των ενδιαφερομένων πολιτών, και η μεσολάβηση της συμμετοχής μέσω εκπροσώπων δεν επαρκούν ώστε να εγγυηθεί μια ποιοτική και ποσοτική βελτίωση της συμμετοχής, τότε ποιο είναι το κόστος (σε χρόνο και χρήμα) για την αύξηση των δειγμάτων αλληλεπίδρασης; Ας φανταστούμε επίσης ότι λόγω των πολλών δημόσιων προωθούμενων πολιτικών που διαχειρίζεται μια συγκεκριμένη κυβερνητική αρχή, οι ενδιαφερόμενοι θα πρέπει να κληθούν να συμμετάσχουν με μια πολύ υψηλή συχνότητα. Πώς θα μπορούσαν συνεπώς τα χρονοδιαγράμματα να οργανωθούν και

πόσοι πόροι θα πρέπει να διατεθούν για την αποζημίωση των πολιτών για την απώλεια χρόνου; Τέλος, ας φανταστούμε ότι η ανατροφοδότηση που ζητείται να δοθεί από τους πολίτες δεν μπορεί να προκύψει από τη χρησιμοποίηση ενός ερωτηματολογίου, αλλά τους ζητά μια πιο ενεργητική προσπάθεια. Πόσο ακριβή είναι η συλλογή μιας τέτοιας υψηλής ποιότητας πληροφόρηση σε χρόνο και χρήμα τόσο για τους πολίτες όσο και για τους φορείς χάραξης πολιτικής;

Η πλατφόρμα PADGETS επιτρέπει να ξεπεραστούν τέτοιοι περιορισμοί, επιτρέποντας την εφαρμογή της συμμετοχικής προσέγγισης για την χάραξη πολιτικής σε έναν απλό και γρήγορο τρόπο που συμβάλλει στην εξοικονόμηση πόρων. Πρώτον, όλοι οι πολίτες είναι ελεύθεροι να συμμετέχουν σε οποιαδήποτε διαδικασία χάραξης πολιτικής τους ενδιαφέρει, κατά το χρόνο που προτιμούν, με την προσπάθεια συμμετοχής που είναι πρόθυμοι να δαπανήσουν, και πάνω απ' όλα με τη βοήθεια του μέσου κοινωνικής δικτύωσης με το οποίο είναι πιο εξοικειωμένοι και έχουν συνηθίσει να περνούν χρόνο σε αυτό. Με άλλα λόγια, η πλατφόρμα αποτελεί ένα εργαλείο για την άνετη και χωρίς δυσκολίες συμμετοχή. Από την αντίθετη πλευρά, οι φορείς χάραξης πολιτικής μπορούν να έχουν συνεχώς πρόσβαση στις πληροφορίες σχετικά με τη γνώμη των πολιτών δηλαδή ότι μπορούν όχι μόνο να έχουν πρόσβαση σε αυτού του είδους την πληροφορία όποτε το επιθυμούν, αλλά και ότι μια τέτοια πληροφορία είναι πάντα ενημερωμένη κατά τη στιγμή της συλλογής της. Επιπλέον, οι φορείς χάραξης πολιτικής μπορούν συνεχώς να αλληλεπιδράσουν με το ευρύτερο κοινό επιτυγχάνοντας έτσι τη γρήγορη τροποποίηση και προσαρμογή των θεμάτων πολιτικής που είναι υπό συζήτηση.

Η ύπαρξη μιας μοναδικής κεντρικής διεπαφής στην πλατφόρμα του padgets επιλύει τα προβλήματα που προκύπτουν με τη διαχείριση των διαφορετικών κοινωνικών μέσων μαζικής ενημέρωσης, καθένα από τα οποία παρουσιάζει ιδιαιτερότητες όσον αφορά τους στόχους, τις διασυνδέσεις, τα λειτουργικά χαρακτηριστικά, τον τύπο του περιεχομένου και το βαθμό ανταλλαγής περιεχομένου.

3.3.2. Καινοτομία στον τομέα των συστημάτων υποστήριξης αποφάσεων στο δημόσιο τομέα

Συστήματα υποστήριξης αποφάσεων κερδίζουν αναγνώριση στο δημόσιο τομέα, ο οποίος επιδιώκει λύσεις για διάφορα προβλήματα σε μια σειρά από διάφορους τομείς. Τον τελευταίο καιρό, λόγω του αναπροσανατολισμού της πολιτικής προς την συμμετοχή του κοινού και τη συνεργασία σε διαδικασίες λήψης αποφάσεων, ο αριθμός των λύσεων στον τομέα της ηλεκτρονικής διακυβέρνησης έχει αυξηθεί. Τα συστήματα στήριξης και συνεργασίας στη λήψη αποφάσεων, ωστόσο, εξακολουθούν να χρησιμοποιούνται κυρίως σε στενά επαγγελματικούς κύκλους και δεν έχουν βρει ακόμα τον δρόμο τους στους φορείς διαμόρφωσης πολιτικής ή στο ευρύ κοινό γενικότερα.

Η ανάπτυξη της πλατφόρμας PADGETS αποτελεί ένα βήμα προς τη δημιουργία ενός συστήματος υποστήριξης αποφάσεων ικανό να επεξεργάζεται και να παρουσιάζει το input της κοινωνίας με έναν δυναμικό τρόπο. Οι κύριες καινοτομίες που εισήγαγε η πλατφόρμα PADGETS έχουν να κάνουν με: την δημιουργία ενός συστήματος υποστήριξης αποφάσεων που συνδυάζει μοντέλα προσομοίωσης με τα διάφορα μέσα κοινωνικής δικτύωσης αλλά επίσης και της καλύτερης αξιοποίησης των δεδομένων που προκύπτουν από την αλληλεπίδραση με τα ίδια τα μέσα κοινωνικής δικτύωσης. Μια σύντομη παρουσίαση αυτών των πτυχών παρέχεται παρακάτω.

Η χρήση εργαλείων που στηρίζονται στη Τεχνολογία Πληροφορικής και Επικοινωνιών για την υποστήριξη της λήψης αποφάσεων ήταν ανέκαθεν μια «κλειστή» δραστηριότητα που πραγματοποιούνταν συνήθως με στατική εξωτερικών εισροών με τη μορφή αδόμητων δεδομένων που προέρχονται από διαφορετικές πηγές (π.χ. στατιστικές υπηρεσίες). Αυτή η προσέγγιση έχει μια σειρά από σημαντικούς περιορισμούς: η έλλειψη άμεσης σύνδεσης με την εξωτερική πραγματικότητα με την οποία η απόφαση πολιτικής πρέπει να αλληλεπιδράσει, την εγγενή καθυστέρηση στην υποστήριξη της απόφασης πολιτικής, λόγω του χρόνου για τη συλλογή και την επεξεργασία των σχετικών στοιχείων που απαιτούνται για την ανάλυση. Η καινοτομία που προέρχεται από το PADGETS συνίσταται στη σύνδεση της διαδικασίας λήψης αποφάσεων με τις δραστηριότητες που ασκούνται στα διάφορα μέσα κοινωνικής δικτύωσης. Αυτό επιτρέπει να διαπιστωθεί μια άμεση σχέση μεταξύ της διαδικασίας λήψης απόφασης και του εξωτερικού κόσμου, ο οποίος αλληλεπιδρά ουσιαστικά και συνεχώς με τα μέσα κοινωνικής δικτύωσης. Συνεπώς, μόλις εξασφαλιστεί ότι οι απαραίτητες οργανωτικές διαδικασίες έχουν τεθεί σε εφαρμογή, θα πρέπει να συμβάλλουν στην παραγωγή ενός πολύ πιο αποδοτικού και αποτελεσματικού τρόπου λήψης αποφάσεων στους φορείς πολιτικής.

Οι αναλύσεις και οι προσομοιώσεις χρησιμοποιώντας περιεχόμενο από τις βασικές δραστηριότητες των χρηστών, συσσωρευμένο στα διάφορα μέσα κοινωνικής δικτύωσης αλλά και στα εργαλεία του web 2.0 ιστού, έχουν πολλά να προσφέρουν προς την κατεύθυνση μιας πιο προσανατολισμένης προς το πολίτη χάραξη πολιτικής. Από αυτή την άποψη, το εργαλείο υποστήριξης λήψης αποφάσεων του padget παρέχει μια σειρά από καινοτόμες λειτουργίες που παράγουν χρήσιμες γνώσεις ώστε να χρησιμοποιηθούν για την ενημέρωση της διαδικασίας λήψης αποφάσεων. Ειδικότερα, επιτρέπει να: δημιουργηθούν στιγμιότυπα με τα επίπεδα ευαισθητοποίησης, ενδιαφέροντος και αποδοχής μιας δεδομένης πολιτικής μέσα από το εσωτερικό των μέσων κοινωνικής δικτύωσης, τονιστούν συγκεκριμένα χαρακτηριστικά που μπορεί να οδηγήσουν τις εκτιμήσεις σε πιθανές αποκλίσεις (π.χ. ηλικία, φύλο, κ.λπ.), να δημιουργηθούν πιθανά σενάρια για το πώς τέτοια επίπεδα ευαισθητοποίησης, ενδιαφέροντος και αποδοχής μπορεί να μεταβάλλονται με την πάροδο του χρόνου (π.χ. επόμενους 12 μήνες) και τέλος να ξεχωρίσει σχετικές γνώμες που προκύπτουν από την αλληλεπίδραση των τελικών χρηστών με το μήνυμα της πολιτικής.

3.4. Ρόλοι που αλληλεπιδρούν μέσω του Padgets

Είναι αναγκαίο να καθοριστούν κατηγορίες υψηλού επιπέδου των φορέων που εμπλέκονται σε μια καμπάνια του Padget, έτσι ώστε με τον τρόπο αυτό να προκύψει μια βασική κατηγοριοποίηση των κλάσεων που θα περιλαμβάνουν μεταξύ άλλων τις κλάσεις των φορέων χάραξης πολιτικής και τις κλάσεις των ενδιαφερόμενων πολιτών που αλληλεπιδρούν με την κυβερνητική πλευρά, μέσω της διαδικασίας προώθησης πολιτικής.

3.4.1. Φορέας χάραξης πολιτικής

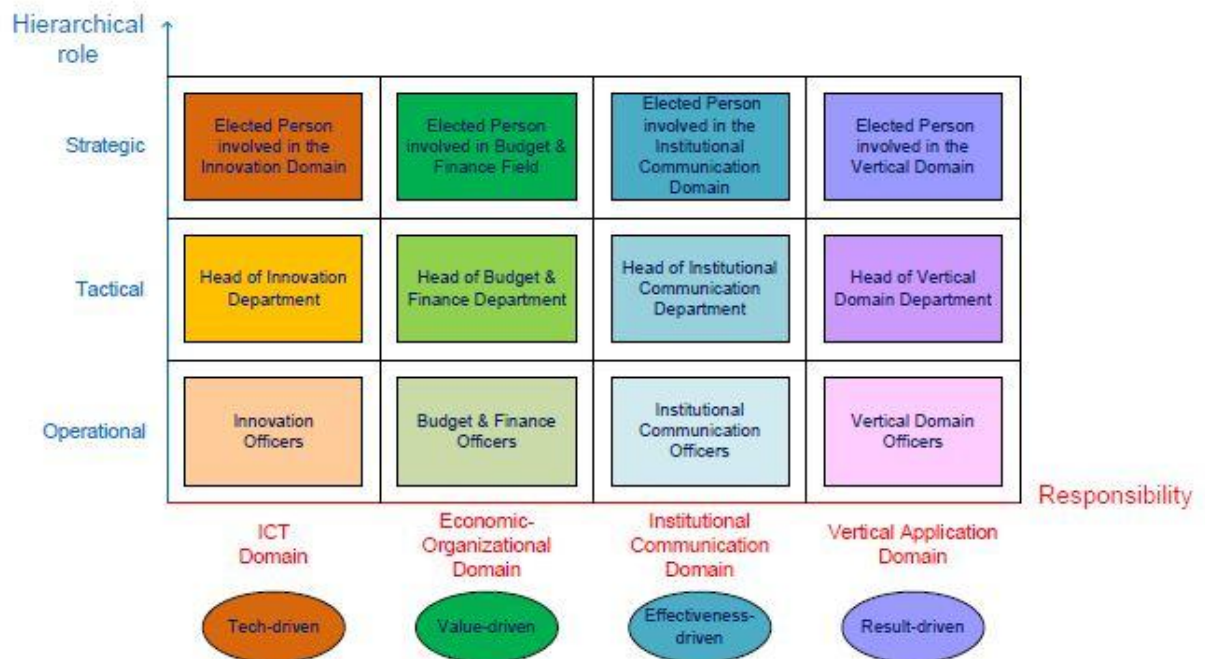
Ο ρόλος του policy maker θα μπορούσε ουσιαστικά να αναλυθεί σε δύο αλληλένδετες προοπτικές, μια κατακόρυφη διάσταση και μια οριζόντια διάσταση, όπως θα παρουσιαστούν παρακάτω.

Η θέση του policy maker κατά μήκος της κατακόρυφης διάστασης σκιαγραφεί τον ιεραρχικό ρόλο που έπαιξε στην κυβερνητική οργάνωση που είναι υπό εξέταση. Θέσεις της ιεραρχίας μέσα στους κυβερνητικούς οργανισμούς μπορούν να ταξινομηθούν κατά τον ακόλουθο τρόπο:

- Στρατηγικούς ρόλους, οι οποίοι συνεπάγονται μακροπρόθεσμη λήψη αποφάσεων και μακροπρόθεσμο σχεδιασμό. Προσδιορισμός των κατευθυντήριων γραμμών και αρχών της πολιτικής που οδηγούν σε δραστηριότητες που χαρακτηρίζονται ως υψηλής ευθύνης.
- Τακτικούς ρόλους, οι οποίοι συνεπάγονται μέσης διάρκειας λήψη αποφάσεων με γνώμονα την επιδίωξη του ευελιξίας και ευκινησίας. Δραστηριότητες προγραμματισμού περιλαμβάνουν το συνδυασμό των διαθέσιμων πόρων, την εξέταση των εμποδίων και τον επανέλεγχο των εναλλακτικών έτσι ώστε να εξασφαλιστεί η εφαρμογή των στρατηγικών σχεδίων.
- Επιχειρησιακούς ρόλους, οι οποίοι συνεπάγονται βραχυπρόθεσμη λήψη αποφάσεων, μικρής εμβέλειας σχεδιασμό και καθημερινή επίβλεψη. Υποχρεώσεις που συνδέονται με αυτούς τους ρόλους αφορούν την πραγματική εκτέλεση των στρατηγικών σχεδίων μέσα από τους επιλεγμένους μηχανισμούς.

Η θέση του policy maker κατά μήκος της οριζόντιας διάστασης αντανακλά την περιοχή ευθύνης του στις διαδικασίες επίβλεψης με τις οποίες είναι επιφορτισμένος στα πλαίσια της διοικητικής διαίρεσης. Η πολυπλοκότητα μιας ολοκληρωμένης καμπάνιας padget απαιτεί μια διεπιστημονική ολοκλήρωση των συνεισφορών που προέρχονται από διάφορους τομείς εμπειρογνωμοσύνης. Εξετάζοντας σε υψηλό επίπεδο, οι βασικές περιοχές ευθύνης μπορούν να περιγραφούν ως εξής:

- Τομέας Τεχνολογίας Πληροφορικής και Επικοινωνιών, συμφέροντα, τα οποία σχετίζονται αποκλειστικά με την τεχνολογική διάσταση, σε αυτόν τον τομέα έχουν σχέση με τη σκοπιμότητα και τη βιωσιμότητα των διαδικασιών του Padget, με μια στενή εστίαση σε θέματα που βρίσκονται στον τομέα της Τεχνολογίας Πληροφορικής και Επικοινωνιών.
- Οικονομικο-Οργανωτικός Τομέας, σχετίζεται όχι μόνο με τις οικονομικές και χρηματοπιστωτικές διαστάσεις μιας padget εφαρμογής, αλλά και με τις αλλαγές που συμβαίνουν στη δομή των κυβερνητικών φορέων, ως συνέπεια της έναρξης της συμμετοχικής διαδικασίας.
- Τομέας Επικοινωνίας, οι δράσεις στον τομέα αυτό αποσκοπούν στην ενίσχυση της εμπιστοσύνης των πολιτών στα κυβερνητικά όργανα και την τόνωση της ενεργού συμμετοχής του κοινού στη λήψη αποφάσεων που συμβαίνουν στο εσωτερικό της δημόσιας διοίκησης.
- Τομέας Εφαρμογής, δηλαδή ο τομέας που επηρεάζεται άμεσα από μια padget διαβούλευση. Τα συμφέροντα σε αυτόν τον τομέα περιλαμβάνουν την επιτυχία του policy maker, του οποίου ο βαθμός της ευαισθητοποίησης, του ενδιαφέροντος και της υποστήριξης διερευνώνται μέσα από την καμπάνια του padget.

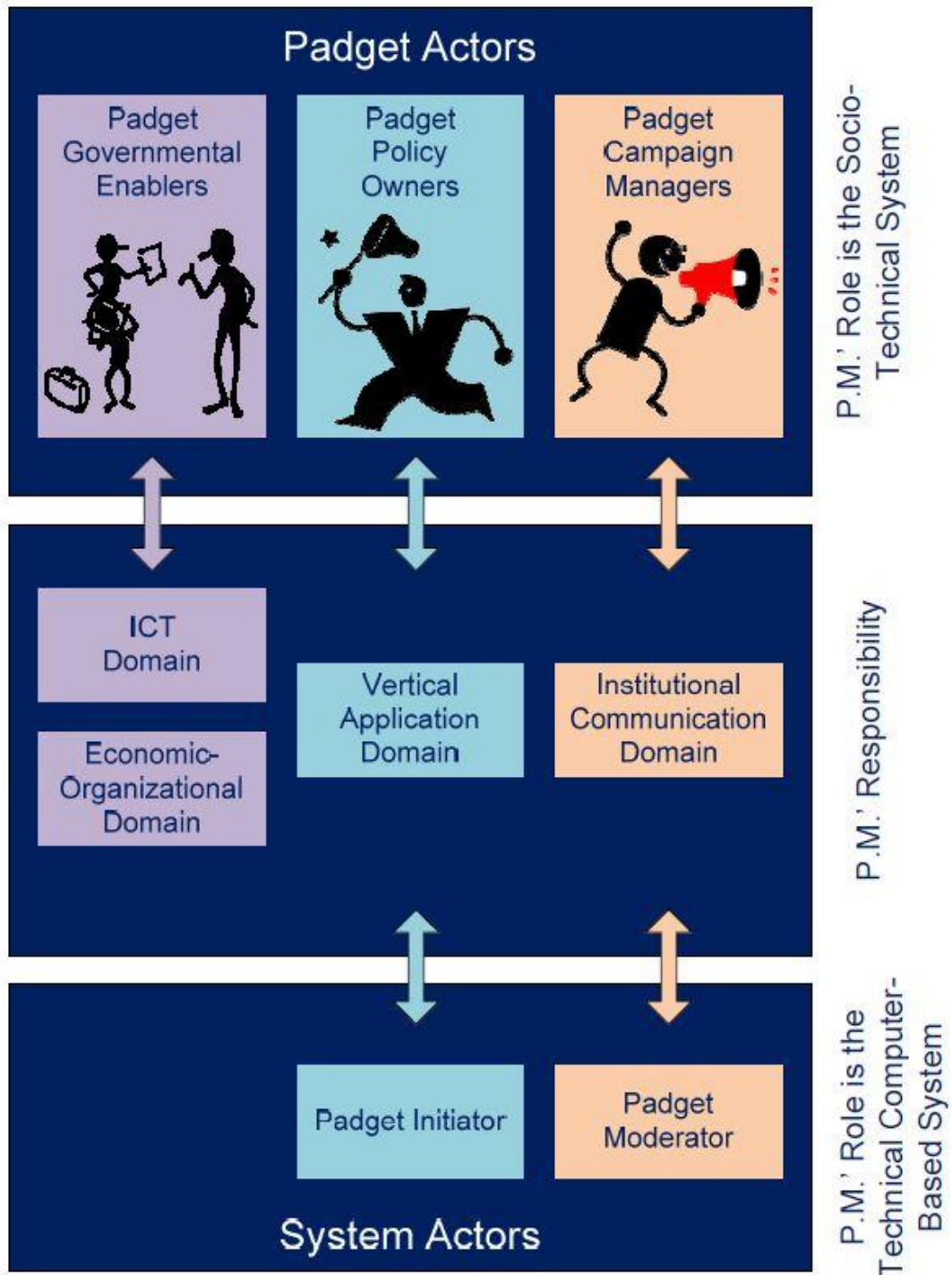


3.3. - Διαφορετικές προοπτικές του ρόλου του policy maker

Άτομα που ανήκουν στα προαναφερθέντα πεδία και θέσεις, παρά το γεγονός ότι κατέχουν ρόλους σχετικούς με την εφαρμογή και τη χρήση της πλατφόρμας Padget, καλύπτουν ετερογενείς λειτουργίες κατά τη διάρκεια του κύκλου ζωής μιας εκστρατείας. Επίσης δεν επιτρέπεται σε όλους, παρόλο που περιλαμβάνονται στο τεχνικό τομέα του Padgets, να έχουν πρόσβαση ως τεχνικοί του ηλεκτρονικού συστήματος. Βάσει της εν λόγω πολυπλοκότητας και λαμβάνοντας υπόψη τις ευθύνες του φορέα χάραξης πολιτικής, τρεις διακριτοί ρόλοι έχουν εντοπιστεί μέχρι σήμερα, προκειμένου να μπορέσουν να κατηγοριοποιηθούν οι φορείς ενός Padget.

- Εκκινήτης Padget πολιτικής, ο οποίος μετά από μέτρηση της σκοπιμότητας των πρωτοβουλιών που βρίσκονται υπό εξέταση υποστηρίζει την υλοποίηση της Padget καμπάνιας, καθιστώντας δυνατή την επίτευξη της δημιουργίας μιας ολοκληρωτικά διαδραστικής και με γνώμονα τη κοινωνία πολιτικής διαβούλευσης.
- Ιδιοκτήτης Padget πολιτικής, δηλαδή τα άτομα με τη δύναμη να επηρεάσουν ή να καθορίσουν τις πολιτικές και τις πρακτικές σε διεθνές, εθνικό, περιφερειακό ή και τοπικό επίπεδο, σε ένα σαφώς καθορισμένο τομέα. Είναι υπεύθυνοι για τη διαμόρφωση της πολιτικής, τον καθορισμό των αρχών για τις δημόσιες πράξεις και ταυτόχρονα την αξιολόγηση των πτυχών που αξίζουν να συζητηθούν κατά τη συμμετοχική διαδικασία.
- Διαχειριστής μιας καμπάνιας Padget, ο οποίος δρα απευθείας στο κυβερνητικό κομμάτι, προκειμένου να ενισχυθεί η κοινωνική αλληλεπίδραση και να τονωθεί η εκμαίευση απόψεων. Επιπρόσθετα, οι αρμοδιότητές του, όπου αυτό είναι αναγκαίο, θα μπορούσαν να αφορούν τη διαμεσολάβηση των ανοικτών συζητήσεων και τη ρύθμιση του συστήματος σε πραγματικό χρόνο.

Οι policy makers στους οποίους επιτρέπεται η πρόσβαση στις τεχνικές ρυθμίσεις του συστήματος που λειτουργεί η εφαρμογή του Padgets μπορούν να κατηγοριοποιηθούν ως «system actors».

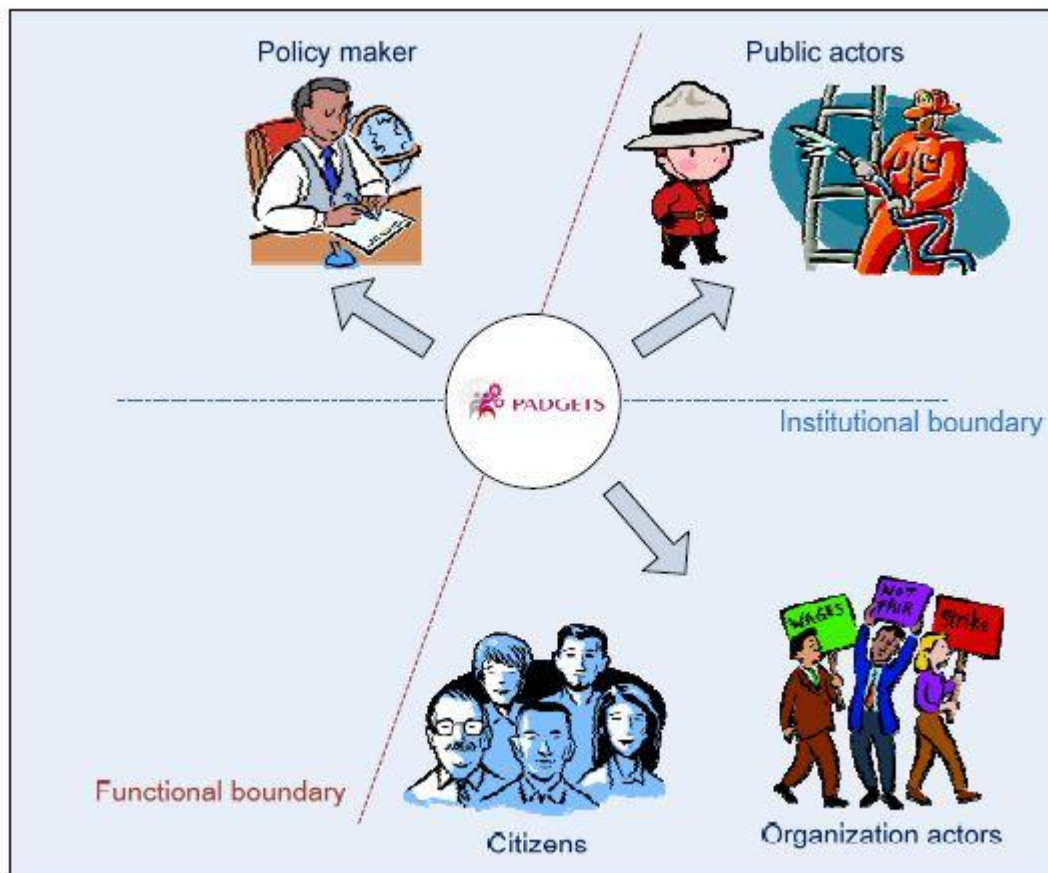


3.4. - Συσχετισμός μεταξύ των ρόλων των policy makers με τις αρμοδιότητές τους

3.4.2. Πολίτης

Η πληθώρα των πολιτών που ενδέχεται να συμμετάσχουν σε Padget εκστρατείες μπορούν να κατηγοριοποιηθούν λαμβάνοντας υπόψη ότι ανήκουν σε τρεις κύριες κατηγορίες των μακροοικονομικών φορέων.

- Πολίτες, δηλαδή απλοί πολίτες ή άτομα που είναι μέλη κοινωνικών (π.χ. μαθητές σχολείου, καθηγητές σχολείου, φοιτητές, πανεπιστημιακοί καθηγητές) ή διοικητικών (π.χ. άτομα που ζουν σε ένα συγκεκριμένο δήμο, επαρχία, περιοχή) ομάδων που συμμετέχουν σε διαδικασίες αξιολόγησης κάποιας εκστρατείας Padget.
- Οργανωτικοί παράγοντες, δηλαδή τα μέλη μιας ευρύτερης κοινωνίας που επιδιώκουν την επίτευξη συλλογικών στόχων και έχουν ένα όριο που τους χωρίζει από το περιβάλλον τους (π.χ. επιχειρήσεις, φιλανθρωπικές οργανώσεις, μη κερδοσκοπικές ομάδες, συνεταιρισμούς, πολιτικά κόμματα, εμπορικά συνδικάτα). Αυτά τα άτομα, αναλαμβάνουν δράση για λογαριασμό της οργάνωσης που εκπροσωπούν, προκειμένου να προωθήσουν τα πιστεύω της και να στηρίξουν τις πρωτοβουλίες που είναι συνυφασμένες με την αποστολή και τις αξίες της οργάνωσης.
- Δημόσιοι λειτουργοί, δηλαδή οι δημόσιοι υπάλληλοι που εργάζονται για ένα τμήμα της κυβέρνησης ή για έναν οργανισμό που εμπλέκεται άμεσα με μια συγκεκριμένη εκστρατεία Padget. Οι δημόσιοι υπάλληλοι μπορούν να παρέχουν μια πολύτιμη συμβολή που βασίζεται στην καθημερινή εμπειρία τους σχετικά με τον τομέα τους και, εκτός αυτού, με τις γνώσεις που έχουν αποκτήσει με την πάροδο του χρόνου.



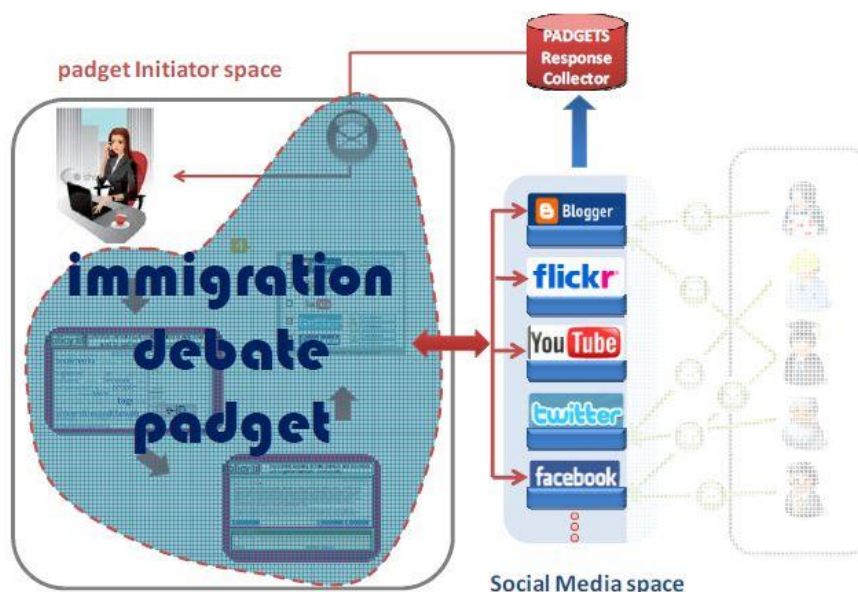
3.5. - Κατηγοριοποίηση του ρόλου του πολίτη

2.5. Μελέτη περίπτωσης - debate για τη μετανάστευση/5/

Στη συνέχεια θα παρουσιαστεί ένα σενάριο περίπτωσης χρήσης μιας εκστρατείας padget. Το σενάριο παρουσιάζεται τόσο από την οπτική του εκκινήτη της καμπάνιας όσο και από την οπτική των τελικών χρηστών και έχει ως σκοπό να περιγράψει ένα παράδειγμα εκστρατείας μέσω Padget βασισμένο σε δεδομένα της πραγματικής ζωής.

2.5.1. Με την προοπτική του policy maker

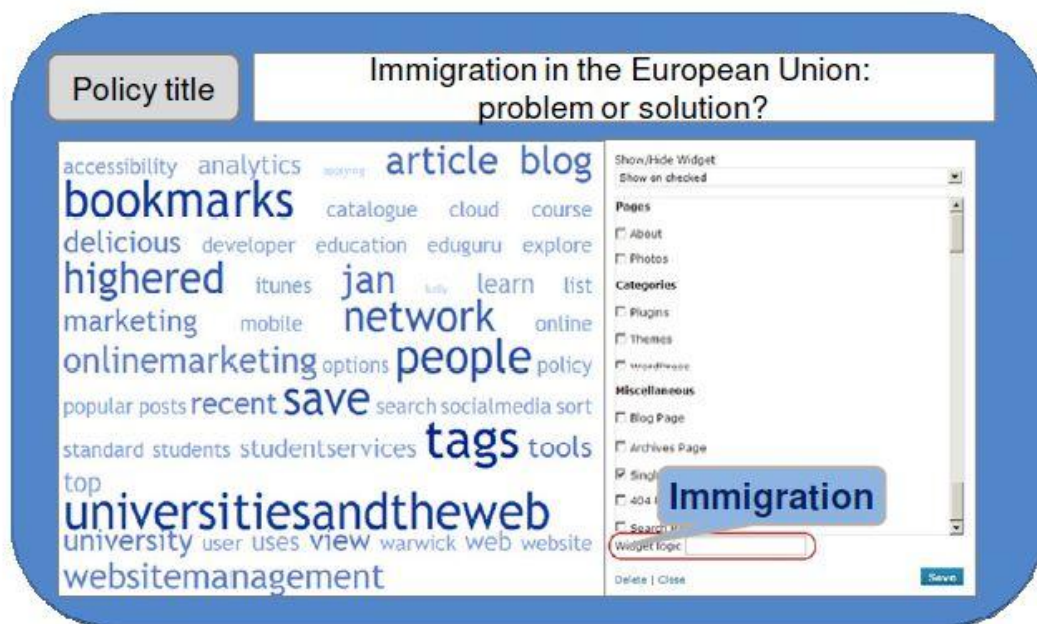
Ένα μέλος του Ευρωπαϊκού Κοινοβουλίου (Tanja) θέλει να συζητήσει το θέμα της μετανάστευσης με τους Σλοβένους πολίτες. Η νόμιμη μετανάστευση και η ένταξη των υπηκόων τρίτων χωρών αποτελούν μέρος μιας σημαντικής διαβούλευσης σήμερα σε ολόκληρη την Ευρωπαϊκή Ένωση. Τα περισσότερα κράτη μέλη τώρα βιώνουν μεταναστευτικά φαινόμενα και αντιμετωπίζουν προβλήματα ένταξης. Ορισμένες χώρες, συμπεριλαμβανομένων των νέων κρατών μελών, αντιμετώπισαν μόλις πρόσφατα το φαινόμενο της μετανάστευσης. Άλλα κράτη μέλη έχουν ασχοληθεί με τη μετανάστευση και τα προβλήματα ένταξης για δεκαετίες, αλλά όχι πάντα με ικανοποιητικά αποτελέσματα, και κατόπιν αυτών αναθεωρούν τις πολιτικές τους. Λαμβάνοντας υπόψη τις διαφορετικές ιστορίες, παραδόσεις και θεσμικές ρυθμίσεις, υπάρχει μια ευρεία ποικιλία προσεγγίσεων που λαμβάνονται για την εξεύρεση λύσεων στα προβλήματα τα οποία πρέπει να αντιμετωπιστούν. Η ΕΕ προσπαθεί για την ανάπτυξη κοινών προσεγγίσεων για την ένταξη και προωθεί την ανταλλαγή των βέλτιστων πρακτικών. Η οικονομική βοήθεια είναι απαραίτητη σε αυτό το πλαίσιο. Ειδικότερα στις περιόδους οικονομικής κρίσης αυτό είναι ένα θέμα που χρίζει ιδιαίτερης διαβούλευσης με τους πολίτες της ΕΕ.



3.6. - Οπτική του πολιτικού για τη διαβούλευση μέσω του padget

Ροή της μελέτης περίπτωσης

1. Η Tanja θέλει να συζητήσει και να συγκεντρώσει απόψεις σχετικά με τη μεταναστευτική πολιτική με τους Σλοβένους πολίτες και με εκπροσώπους μη-κυβερνητικών οργανώσεων. Είναι προς το συμφέρον του ευρωβουλευτή ότι το θέμα συζητείται επίσης στα μέσα ενημέρωσης και έτσι το μήνυμα θα μπορεί να προσεγγιστεί από ένα ευρύτερο κοινό. Χρησιμοποιεί τον Web Editor του PADGETS για να θέσει το εξής ερώτημα: «Η μετανάστευση στην Ευρωπαϊκή Ένωση: πρόβλημα ή λύση;», με τη δημιουργία μιας αντίστοιχης καμπάνιας Padget. Επιπλέον, επισημαίνει τη καμπάνια της με την ετικέτα Immigration Rights Draft Resolution.



2. Στη συνέχεια, προβαίνει στη δήλωση της πολιτικής της θέσης, περιγράφοντας την εμφανή σημεία και επισυνάπτοντας διάφορους τύπους υποστηρικτικού υλικού.



Policy title Immigration in the European Union:
problem or solution?

Characters left: 11

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam justo sapien, mollis id, varius et, suscipit in, leo. In eget orci et nibh lobortis feugiat. Mauris nisi. In pede ante, fringilla eu, elementum vitae, accumsan ac, eros. Nunc vulputate, nibh vel commodo molestie, magna nulla dapibus risus, ut egestas magna tellus nec nunc. Aenean at turpis. Sed laoreet elit quis risus. Maecenas euismod nibh ac felis. Duis nibh mauris, nonummy id, lacinia et, dignissim nec, orci. Nam suscipit bibendum dolor. Nullam tincidunt iaculis sapien. Etiam ornare porta risus. Etiam ultricies, erat a consequat ullamcorper, orci augue laoreet lectus, nec mattis erat tortor id magna. Etiam scelerisque lectus et sem. Sed sit amet quam. Quisque interdum est in velit. Suspendisse egestas viverra est. Donec libero nisi, sollicitudin id, vehicula in, adipiscing eget, enim.

Sed id eros at sapien posuere euismod. Cras id purus sit amet odio pellentesque ultrices. Donec dignissim. Curabitur eu dolor a pede varius laoreet. Nunc laoreet pede ac metus. Vestibulum quis mauris a magna tincidunt tincidunt. Cras malesuada. Ut augue dolor, auctor vel, rhoncus vel, consequat in, diam. Nunc sit amet lorem.

Attach File(s)

Mv Computer

Attach:

3. Η Tanja αφού έχει γράψει το μήνυμά της και έχει επισυνάψει το υπόλοιπο περιεχόμενό της στην κεντρική οθόνη της εφαρμογής, επιλέγει τα μέσα κοινωνικής δικτύωσης στα οποία θα εμφανιστεί το περιεχόμενο της καμπάνιας της.

Blogger ID: ***** PW: *****

Picasa

YouTube

Twitter ID: ***** PW: *****

facebook ID: ***** PW: *****

Remember this login account

Αφού κάνει login στα social media που έχει επιλέξει εμφανίζεται μια οθόνη επιβεβαίωσης.



4. Αφού επιλεγεί η δημοσίευση του περιεχομένου η εφαρμογή διασπά παρασκευαστικά το μήνυμα στα διάφορα στοιχεία που το συνθέτουν και δημιουργώντας διαφορετικά στιγμιότυπα που είναι κατάλληλα για τα επιλεγμένα κανάλια - δηλαδή μια σύντομη περιγραφή 140 χαρακτήρες το πολύ για το Twitter, μια εκτενέστερη περιγραφή για την κατάσταση στο Facebook, κ.λπ.. Κάθε στιγμιότυπο δημοσιεύεται στην αντίστοιχη πλατφόρμα κάποιου social media ανάλογα με τη συμβατότητα.
5. Η Tanja δεν επιθυμεί να συμμετάσχει σε μακρές συζητήσεις, αυτή απλά ενδιαφέρεται να πάρει την πρώτη αντίδραση των πολιτών στις ερωτήσεις της. Η ανταπόκριση των πολιτών που σχετίζεται με την εκστρατεία της Tanja συλλέγονται από την εφαρμογή. Αυτό μπορεί να είναι πολύ χρήσιμο για την Tanja, η οποία θα πρέπει να ψηφίσει, να συζητήσει και να δώσει προτάσεις για το θέμα της διαβούλευσής της στην Επιτροπή και στις συνεδριάσεις της ολομέλειας του Ευρωπαϊκού Κοινοβουλίου. Μέσω της καμπάνιας ένα ευρύτερο κοινό μπορεί να συμμετέχει ολόενα και περισσότερο στη διαδικασία χάραξης πολιτικής και να γίνει ουσιαστικά εμπνευστής νέων πολιτικών.

Μέσα από το κομμάτι του Padgets για την υποστήριξη λήψης απόφασης η Tanja τροφοδοτείται με ανεξάρτητους κοινωνικούς δείκτες που προκύπτουν από την αλληλεπίδραση της εκστρατείας μέσα από το Padget, δηλαδή πληροφορίες σε ατομικό επίπεδο σχετικά με τις τάσεις της ευαισθητοποίησης, του ενδιαφέροντος, των απόψεων, των προτάσεων κ.λπ..

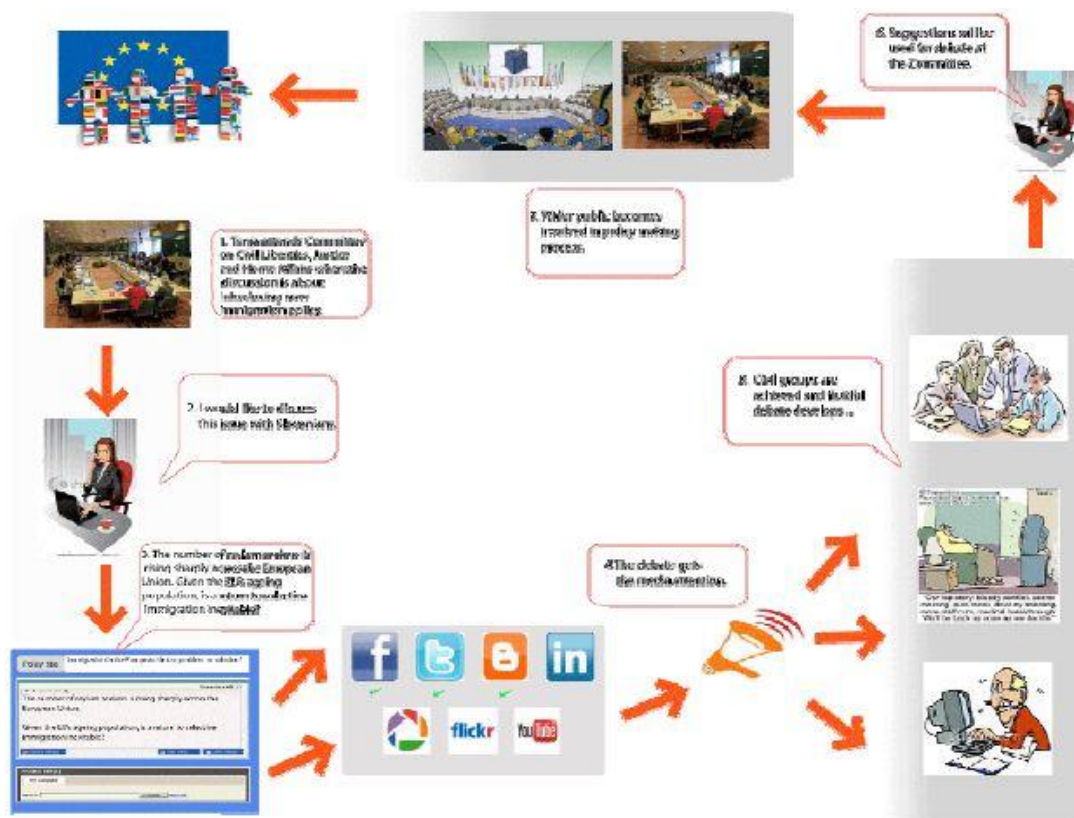
- πλήθος από ατομικά reaches της καμπάνιας
- πλήθος από ατομικά views της καμπάνιας
- πλήθος από likes/dislikes
- πλήθος από comments
- περιεχόμενο των comments

- στοιχεία που αφορούν τις γενικότερες συνθήκες επισκεψιμότητας (π.χ. browsers περιήγησης, τύπος συσκευής περιήγησης)
- χρονικές ετικέτες στα παραπάνω στοιχεία
- ατομικά δημογραφικά στοιχεία (π.χ. ηλικία, φύλλο, τοποθεσία, επίπεδο μόρφωσης, κ.λπ.)

Μέσα από το radgets analytics η Tanja λαμβάνει αναλυτική αναφορά με όλα τα παραπάνω δεδομένα της διαβούλευσής της, γεγονός που τη διευκολύνει για περαιτέρω διαβούλευση στην Επιτροπή.

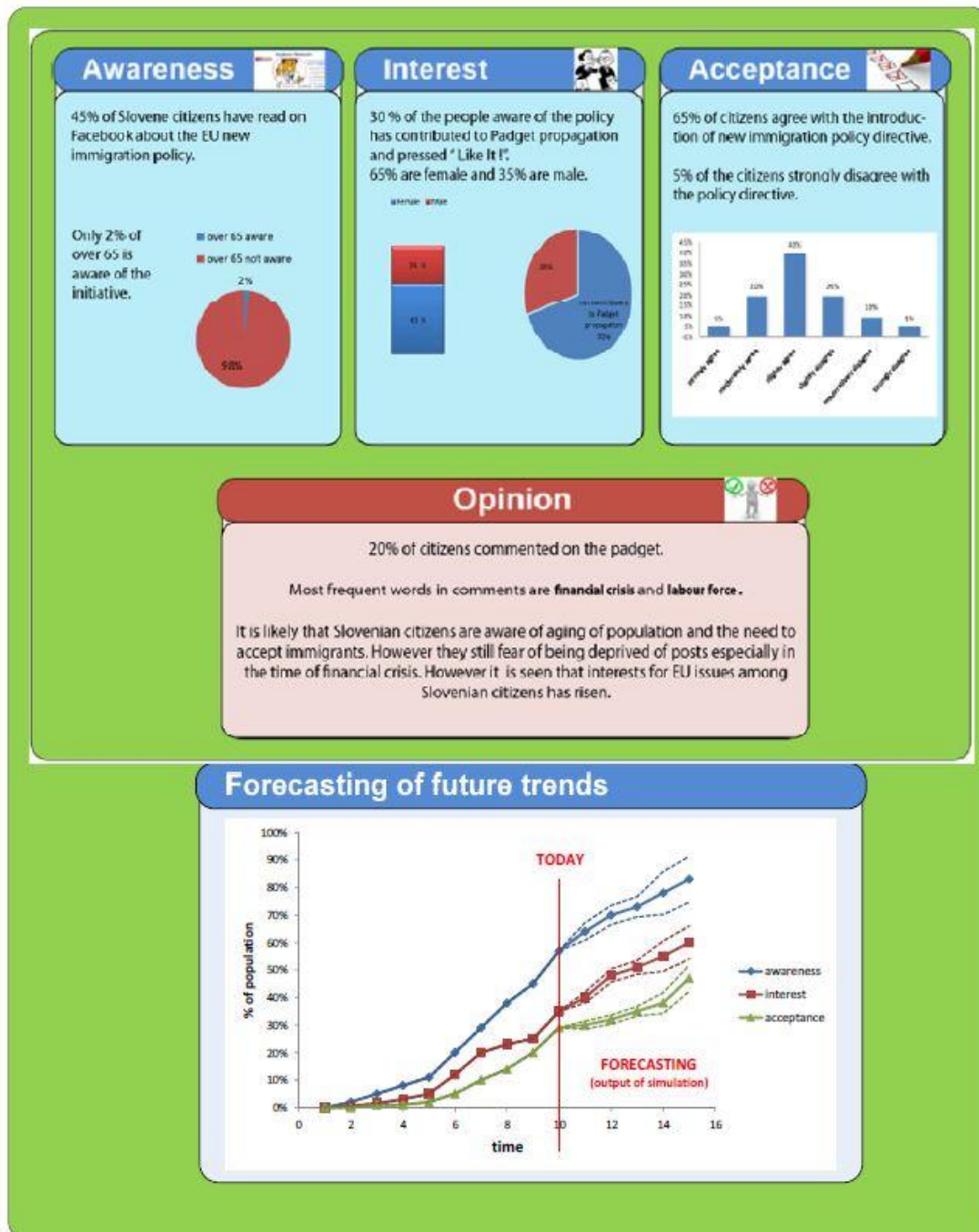


3.7. - Ροή της άποψης σχετικά με τη μετανάστευση που διαμορφώνεται δυναμικά



3.8. - Συνολική εικόνα της εκστρατείας από την οπτική του policy maker

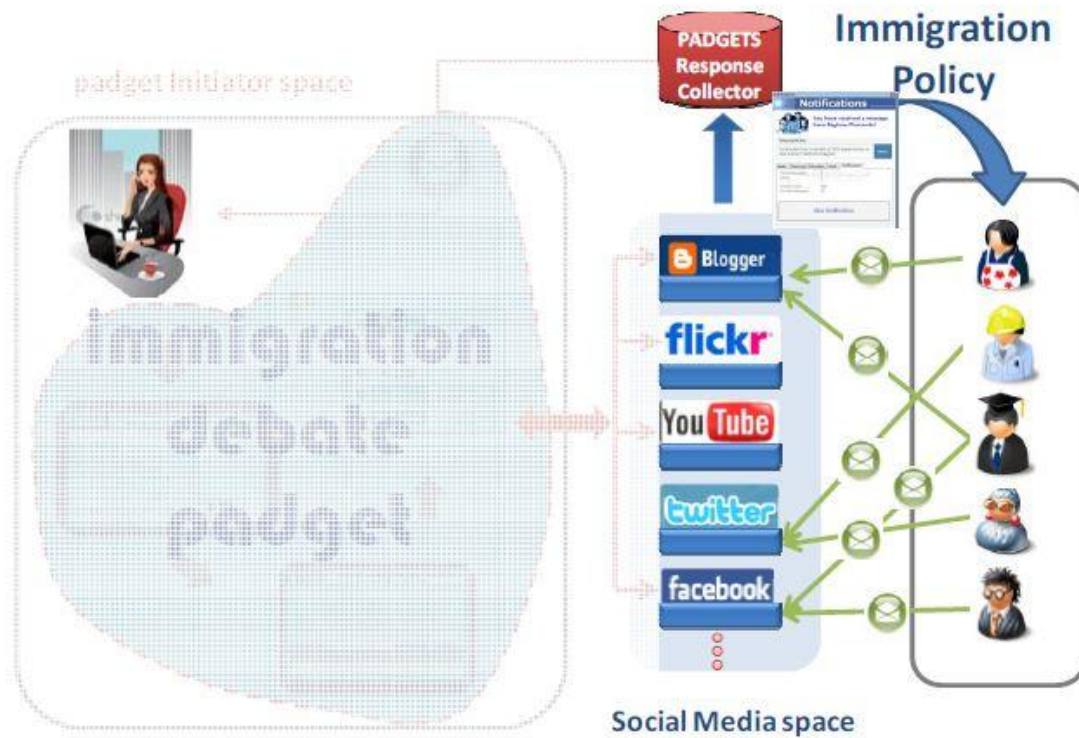
Μετά από την αλληλεπίδραση των πολιτών με την καμπάνια ο policy maker βλέπει καθαρά αριθμητικά δεδομένα βάσει τεσσάρων δεικτών.



3.9. - Εμφάνιση καθαρών αριθμητικών στοιχείων

2.5.2. Με την προοπτική του πολίτη

Ο Peter είναι ένας από τους πολυάριθμους πολίτες που ενδιαφέρθηκε για την έντονη διαβούλευση που έλαβε χώρα πρόσφατα μεταξύ του Σλοβενικού κοινού.



3.10. - Οπτική της διαβούλευσης από τη μεριά του πολίτη

Ροή της μελέτης περίπτωσης

1. Βαθμός εναισθητοποίησης

Ο Peter, ως τελικός χρήστης του Padgets, ενημερώνεται με μια κοινοποίηση από την Tanja, η οποία είναι ο εκκινήτης της εκστρατείας. Τώρα έχει την ευκαιρία να εκφράζει τη γνώμη του σχετικά με τη δήλωση και, επιπλέον, να εκφράσει την προτίμησή του για το προσχέδιο του ψηφίσματος για τα δικαιώματα μετανάστευσης.



2. Κοινωνική αλληλεπίδραση

Ο Peter εκφράζει το επίπεδο της αποδοχής σχετικά με το προσχέδιο του ψηφίσματος για τα δικαιώματα μετανάστευσης και γράφει τα σχόλιά του για το προσχέδιο του ψηφίσματος με το Front End εργαλείο του Padgets.

1. Would you support the EU Immigration Rights draft resolution?
Please indicate your level of agreement.

2. Would you like to comment on this ?
I think we should limit immigration to EU member states.

End users select their level of agreement.....

Ένας άλλος τελικός χρήστης του Padgets, μια σλοβενική μη κυβερνητική οργάνωση, έρχεται σε επαφή με τη διαβούλευση του μεταναστευτικού μέσα από το Twitter και δίνει τη γνώμη της και την πρότασή της στην Ευρωβουλευτή.



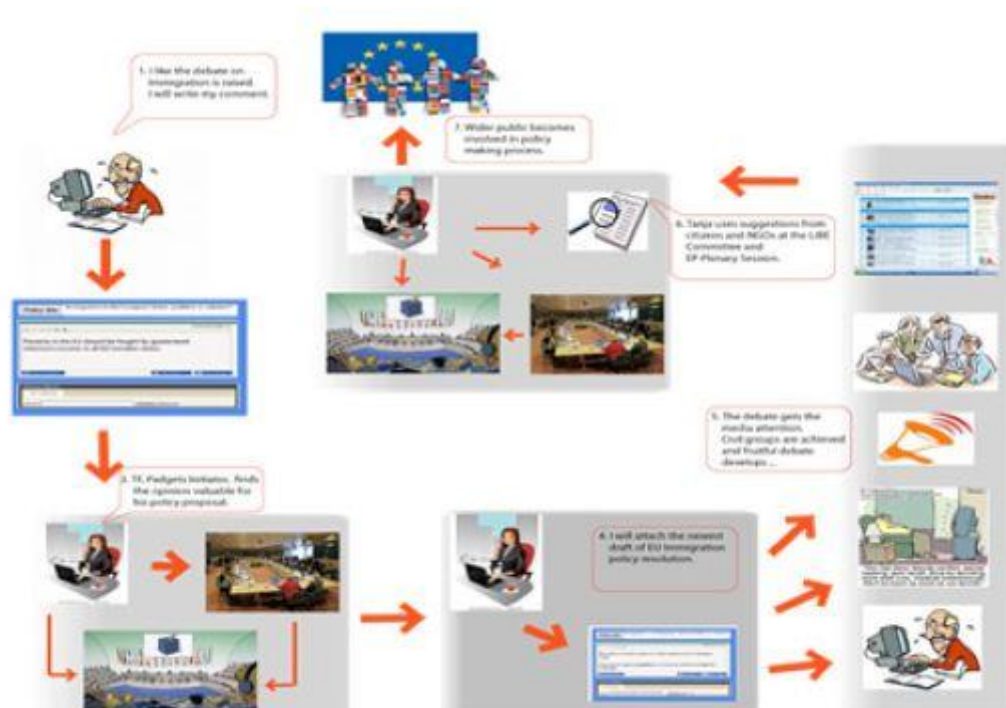
twitter



Μια άλλη ΜΚΟ της Σλοβενίας, το Ινστιτούτο για την ηλεκτρονική συμμετοχή (INePA), ενεργώντας ως τελικός χρήστης του Padget, έρχεται σε επαφή με τη διαβούλευση μέσω του Blogger. Υιοθετεί τη διαβούλευση για τη μετανάστευση στο δικό της forum.



Γόνιμη συζήτηση λαμβάνει χώρα και οι τελικούς χρήστες του Padget αυξάνονται αισθητά: οι δημοσιογράφοι ενδιαφέρονται για τις απόψεις των ατόμων και των ΜΚΟ, οι οποίες δημοσιεύονται στους ευρωβουλευτές με αποτέλεσμα την εξάπλωσή τους στο ευρύτερο κοινό.






3.11. - Ροή της αλληλεπίδρασης[5]


Το κοινό έχει κίνητρα για να συζητήσει το θέμα, εφόσον το θέμα αυτό αποτελεί μέρος της καθημερινής τους ζωής. Μερικοί χρησιμοποιούν το Facebook, το Twitter, το Blogger, ακόμη και το Youtube για μηνύματα βίντεο. Οι Σλοβένοι πολίτες

έδειξαν μεγαλύτερο ενδιαφέρον στα Ευρωπαϊκά θέματα: το γεγονός αυτό αποτελεί υψίστης σημασίας δεδομένου ότι το τρέχον ενδιαφέρον για τα ζητήματα ΕΕ στη Σλοβενία είναι εξαιρετικά χαμηλό (η προσέλευση στις ευρωπαϊκές εκλογές του 2009 δεν έχει υπερβεί το 20%).

3.6. Use Case ανάλυση

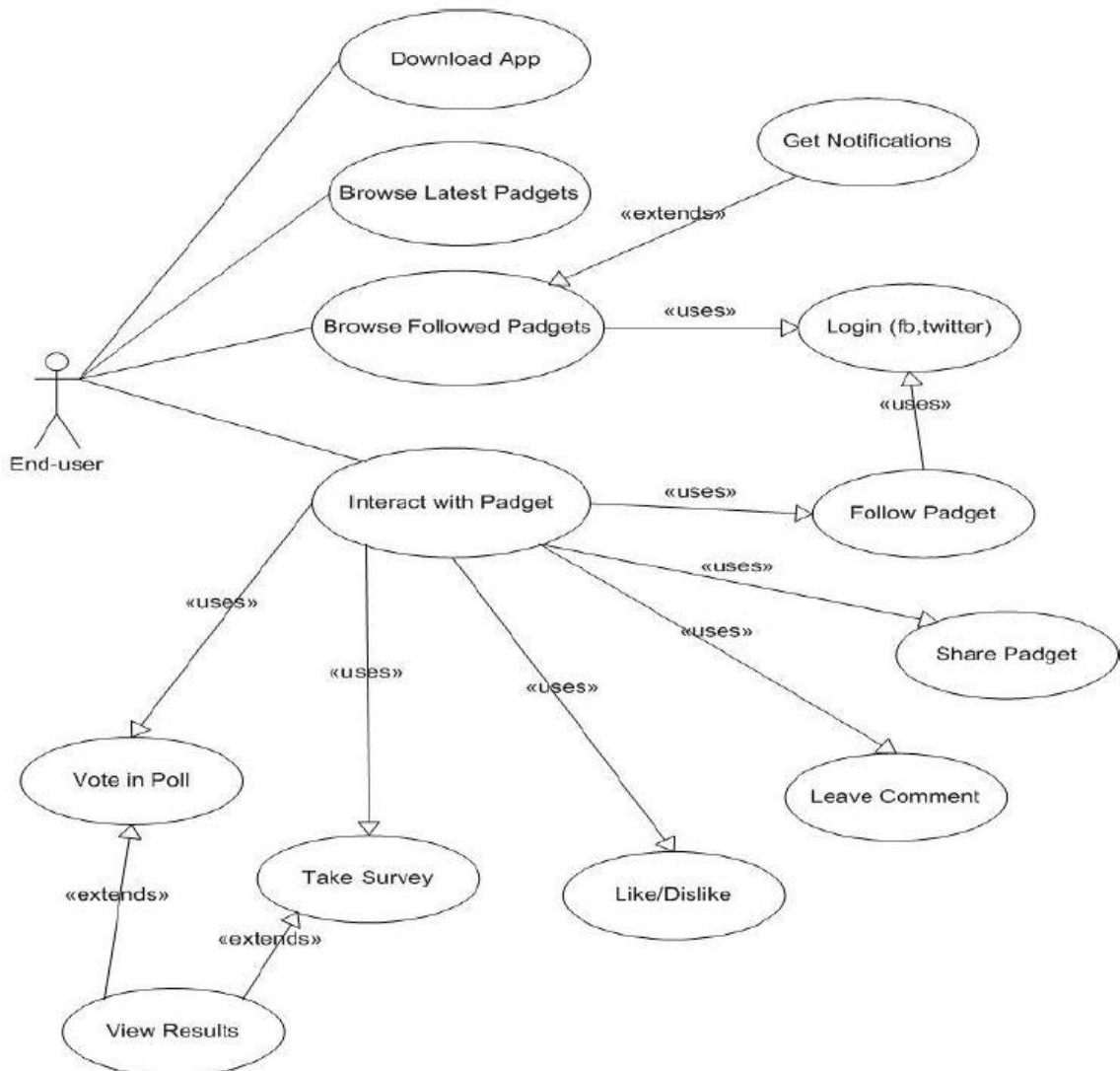
Προκειμένου να προσδιορίσουμε τους κύριους παράγοντες που σχετίζονται με την πλατφόρμα PADGETS, διαφορετικοί τύποι χρηστών θα παρουσιαστούν σε αυτήν την ενότητα ως actors. Ένας actor είναι όποιος ανταλλάσει δεδομένα με την εφαρμογή. Ένας actor μπορεί να είναι ένας εσωτερικός ή εξωτερικός χρήστης. Οι βασικοί παράγοντες (κατηγορίες χρηστών) της πλατφόρμας παρουσιάζονται ως εξής:

Actors	Ρόλος	Περιγραφή
 Citizen	End User	Ο "Citizen" είναι ο τελικός χρήστης που θα είναι σε θέση να έχει πρόσβαση και να διαβάζει τις πληροφορίες μέσα από τους δικούς του λογαριασμούς των μέσων κοινωνικής δικτύωσης, όπως το Facebook, το Twitter, το Youtube, κ.λπ.. Σε κάθε μία από τις αντίστοιχες πλατφόρμες όπου ο "Citizen" θα έχει πρόσβαση στο περιεχόμενο της πολιτικής, θα είναι σε θέση να χρησιμοποιήσει τις δυνατότητες που παρέχονται από την εκάστοτε πλατφόρμα για να αλληλεπιδράσει και να συνεισφέρει την άποψή του.
 Decision Maker	Padgets Initiator	Ο "Decision Maker" είναι ο εκκινητής του Gadget πολιτικής με τον Web Editor του PADGETS παρέχοντας όλες τις σχετικές πληροφορίες και το περιεχόμενο για την έναρξη της padget εκστρατείας καθώς και τη λήψη της ανατροφοδότησης όπως αυτή συσσωρεύεται από την σύστημα.
 Consultant	Padgets Supporter	Ο "Consultant" είναι ένας εξωτερικός χρήστης του συστήματος που παρέχει υποστήριξη στο φορέα λήψης απόφασης σχετικά με τη χρήση της πλατφόρμας PADGETS και των επιμέρους τμημάτων της εφαρμογής. Είναι σε θέση να χρησιμοποιήσει τα εργαλεία λήψης αποφάσεων της

		εφαρμογής προχωρώντας σε μεθοδολογική προσέγγιση και αξιολόγηση της γνώμης των τελικών χρηστών.
 System Administrator	System Supporter	<p>Ο "System Administrator" είναι κάθε εξουσιοδοτημένο πρόσωπο που παρακολουθεί το σύστημα από τεχνικής άποψης, εξασφαλίζει την ορθή λειτουργία της και διαχειρίζεται τους χρήστες, τους ρόλους και τα στατιστικά στοιχεία. Η πλατφόρμα του radgets ως ένα σύστημα που έχει ανάγκη συντήρησης και ελέγχου, πρέπει να έχει ένα πρόσωπο ή έναν οργανισμό που θα είναι υπεύθυνοι για τη συνολική διαχείριση του συστήματος, τη διαχείριση των χρηστών (δικαιώματα, περιορισμοί, λογαριασμοί) και την επίλυση των προβλημάτων. Ο System Administrator είναι επίσης υπεύθυνος για τη συνολική λειτουργία της πλατφόρμας και κυρίως για τους τρόπους με τους οποίους το περιεχόμενο της αλληλεπίδρασης παρέχεται στους τελικούς χρήστες μέσα από διαφορετικά κανάλια επικοινωνίας με έναν εύκολο και αποτελεσματικό τρόπο.</p>

3.12. - Πίνακας χρηστών συστήματος[51]

Σε αυτό το κομμάτι δίνεται μια βασική περιγραφή της αρχιτεκτονικής του συστήματος μέσα από τη μελέτη Use Cases. Κυρίως περιγράφεται το σύνολο των περιπτώσεων χρήσης που αντιπροσωπεύουν ορισμένες από τις πιο σημαντικές, κεντρικές λειτουργίες. Παρακάτω ακολουθούν οι περιπτώσεις χρήσης της πλατφόρμας radgets και των σχετιζόμενων εφαρμογών.

End User[5]

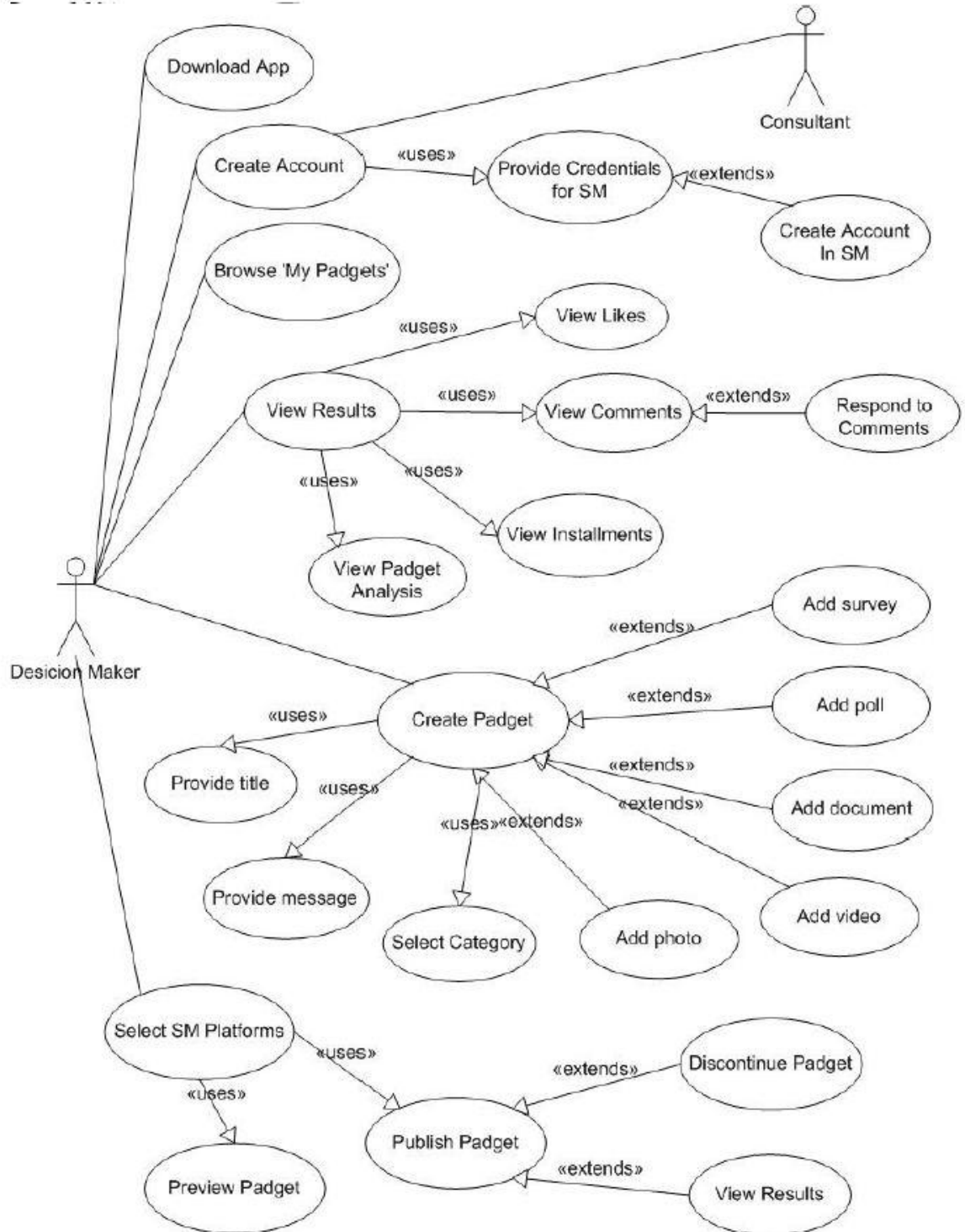
3.13. - Use Case διάγραμμα του End User

- Download App (UC 01)
 - Find app in Android market / iGadget Directory.
 - Install app on device / Add gadget to iGoogle page.
- Browse Latest Padgets (UC 02)
 - View a list of all the Padgets.
 - Sort by date, subject and decision maker.
 - Search.
- Browse Followed Padgets (UC 03)
 - View a list of the padgets that the user follows in app and other SM.

- Login (UC 04)
User has to login in order to be able to follow Padgets and get notifications
 - Google account.
 - Facebook account.
 - OpenId.
 - Twitter account.
- Get Notifications (UC 05)
 - Receive notification about followed padgets
 - * New comment.
 - * New material.
 - * Other.
- Interact with Padget (UC 06)
 - The user will be able to perform a set of actions depending on how the padget has been built.
- Follow Padget (UC 07)
 - Requires Login.
 - View it in a separate tab (followed padgets).
 - Receive notifications.
- Share Padget (UC 08)
 - Share the padget to SM.
- Leave Comment (UC 09)
 - Write a free text comment in response to the statement of the padget or to another comment.
- Like/Dislike (UC 10)
 - One click functionality to express approval /disapproval of the padget statement.
- Take Survey (UC 11)
 - Answer to a set of questions.
 - Demographic info may be asked in a survey.
- Vote in Poll (UC 12)
 - Select one answer from a predefined set of options.
- View Results (UC 13)

- Aggregated results of surveys and polls.

Padgets Initiator / Consultant[5]

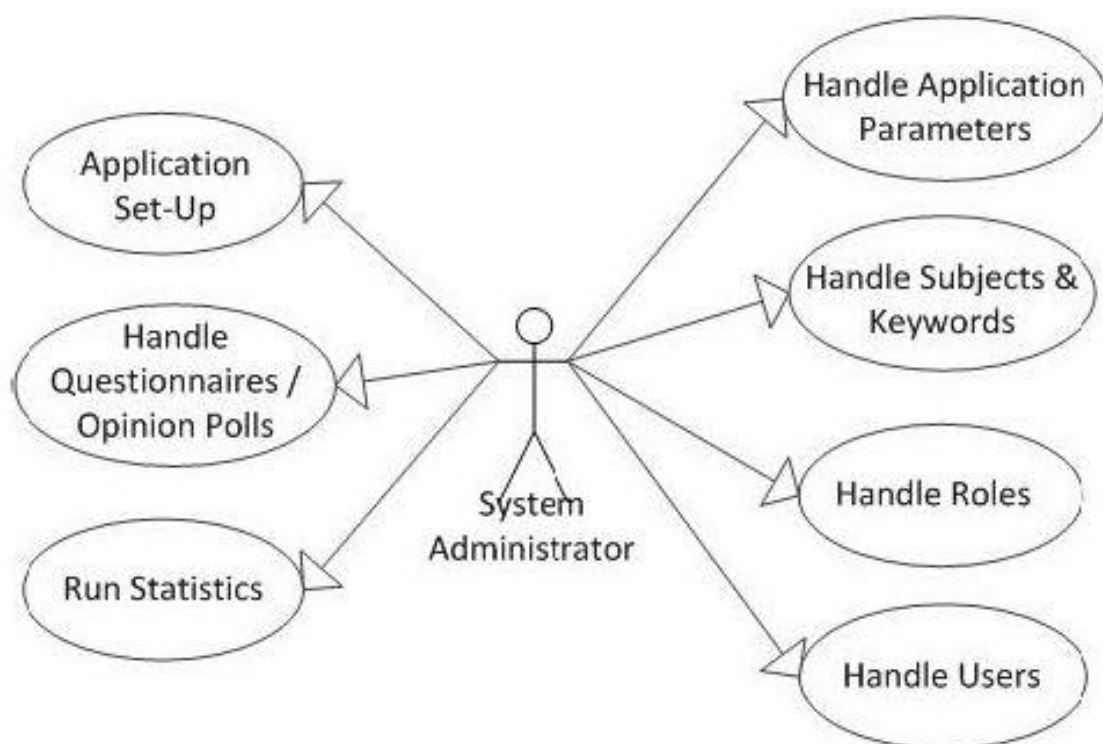


3.14. - Use Case διάγραμμα του Padgets Initiator / Consultant

- Download App (UC 14)

- Find app in Android market / iGadget Directory.
- Install app on device / Add gadget to iGoogle page.
- Create Account (UC 15)
 - The PADGETS Initiator will have to create an account in order to be able to create/manage his padgets. A consultant will assist him to setup the account and demonstrate the padgets functionalities.
- Provide Credentials for SM (UC 16)
 - The PM has to provide her credentials for the various SM platforms that she wants to publish her padgets to.
- Create Account in SM (UC 17)
 - The PM might have to create new accounts in SM in case he doesn't already have.
- Browse 'My Padgets' (UC 18)
 - A list view of all the padgets that the PM has initiated will be available.
- View Results (UC 19)
 - A dashboard of the PM's padgets will provide results about the published padgets.
- View Likes (UC 20)
 - The Likes/Dislikes to a published padget.
- View Comments (UC 21)
 - The list of comments for a published padget.
- Respond to comments (UC 22)
 - The PM will be able to send a respond to a comment made to one of her padgets.
- View installations (UC 23)
 - The number of installations of the padget that can serve as an indicator of the infiltration of the padget.
- View Padgets Analysis (UC 24)
 - This is a general use case containing all other information that Padgets will provide.
- Create Padget (UC 25)
 - The creation of the padget will be an easy process of graphically adding the pieces of a padget together.
- Provide title (UC 26)

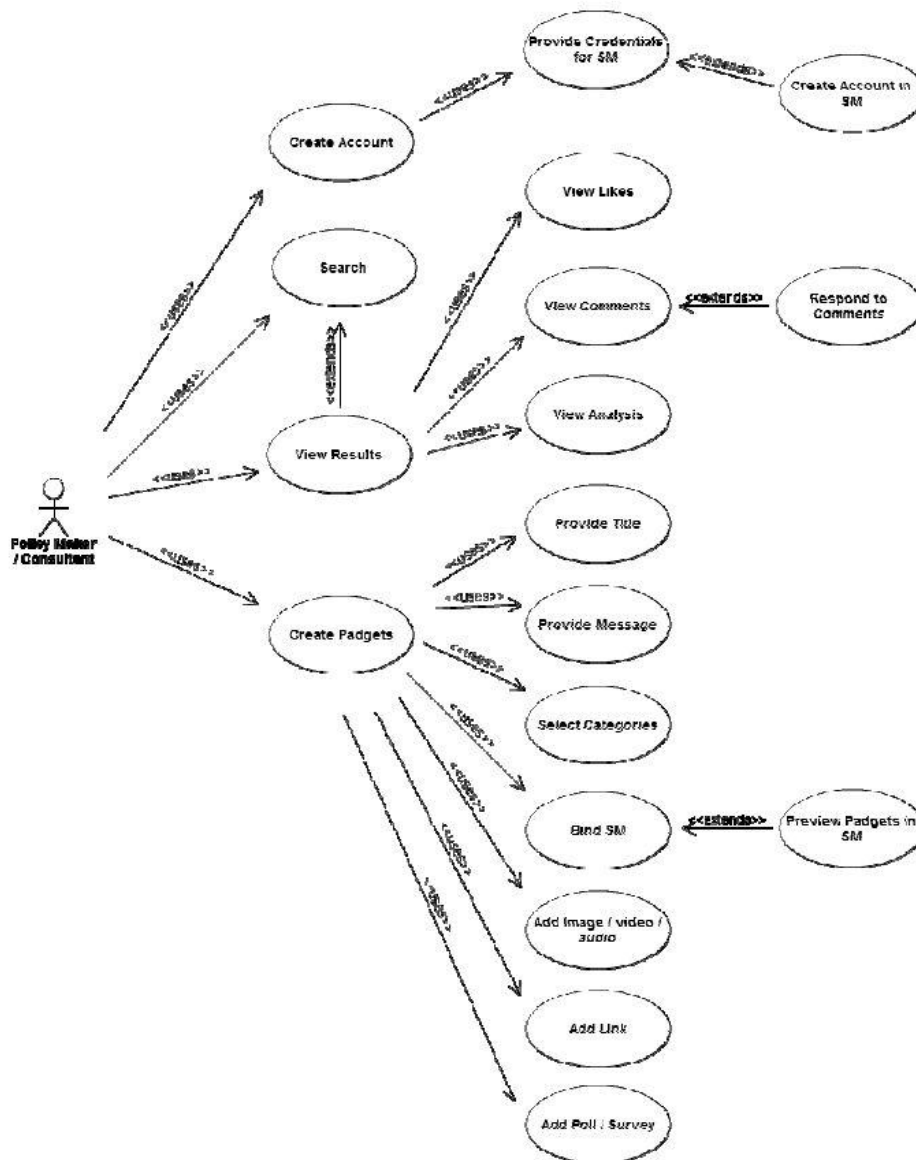
- The title of the padget is the first and maybe more important part of the padget.
- Provide message (UC 27)
 - The message is also an obligatory field that must be provided.
- Select category (UC 28)
 - The category of the policy message must be defined in order to help categorization and search functionality of the system
- Add photo (UC 29)
 - A photo can be attached to the padget taken by a photo gallery or at the same moment if the padget is instantiated via a mobile device.
- Add video (UC 30)
 - A video can be also part of a padget in a similar way with the photo.
- Add document (UC 31)
 - Documents supporting the message of the padget will be allowed.
- Add poll (UC 32)
 - A poll can be associated with a padget in order to get people' s opinion on an issue.
- Add survey (UC 33)
 - A survey can be attached to a Padget providing an extra tool to gather information.
- Select SM Platforms (UC 34)
 - The Decision Maker can select any of the supported SM platforms to publish his padget.
- Preview Padget (UC 35)
 - The Decision Maker will be able to see how the padget will look like in the selected SM platforms before publishing it.
- Publish Padget (UC 36)
 - The padget is published to the selected SM platforms.
- Discontinue Padget (UC 37)
 - The Decision Maker will be able to stop a padget from being monitored by the system.

System Administrator[5]

3.15. - Use Case διάγραμμα του System Administrator

- Handle Application Parameters/ Item Metadata (UC 38)
 - Handle Countries (View, Add / Modify, Search, Delete).
 - Handle Cities (Add / Modify Cities, Search and Delete).
 - Handle Languages (View, Add / Modify, Search, Delete).
- Handle Subjects & Keywords (Application Parameters) (UC 39)
 - Insert / Modify subjects.
 - Save Subjects & Keywords.
 - Delete Subjects & Keywords.
 - Print List of Subjects & Keywords.
 - Assign tags and keyword into subjects.
 - Link subjects based on similarity.
- Handle Roles (Security & Permissions) (UC 40)
 - Create New Role.
 - Modify existing Roles.
 - Assign permissions to roles.

- Remove Role.
- Print User Role (Users, Permissions).
- Handle Users (UC 41)
 - Assign roles to users.
 - Add New / Modify Existing User (User Role, Security Level, Login ID, Password, Personal Information).
 - Save New / Modified User Data.
 - Search Users (Fill various criteria and search).
 - Store Search Criteria (for future use).
 - Remove User.
 - Print Preview / Print List of Users.
- Run Statistics (Traffic Statistics) (UC 42)
 - Run & Preview Report.
 - Print Report.
 - Save Report.
 - Platform activity statistics
 - * Generate statistics for a poll/survey.
 - * Generate statistics regarding the visibility of a section.
 - * Most active issues.
- Handle questionnaires/opinion polls (UC 43)
 - Publish new evaluation questionnaire to the portal.
 - Generate reports regarding the results of a poll/survey.
 - Post feedback from relevant organizations in the platform.
- Application Setup (UC 44)
 - Handle General options.
 - Handle Repository (Path of Repository).

Padgets Initiator and Consultant (Web Application) [5]

3.16. - Use Case διάγραμμα του Padgets Initiator / Consultant (Web Application)

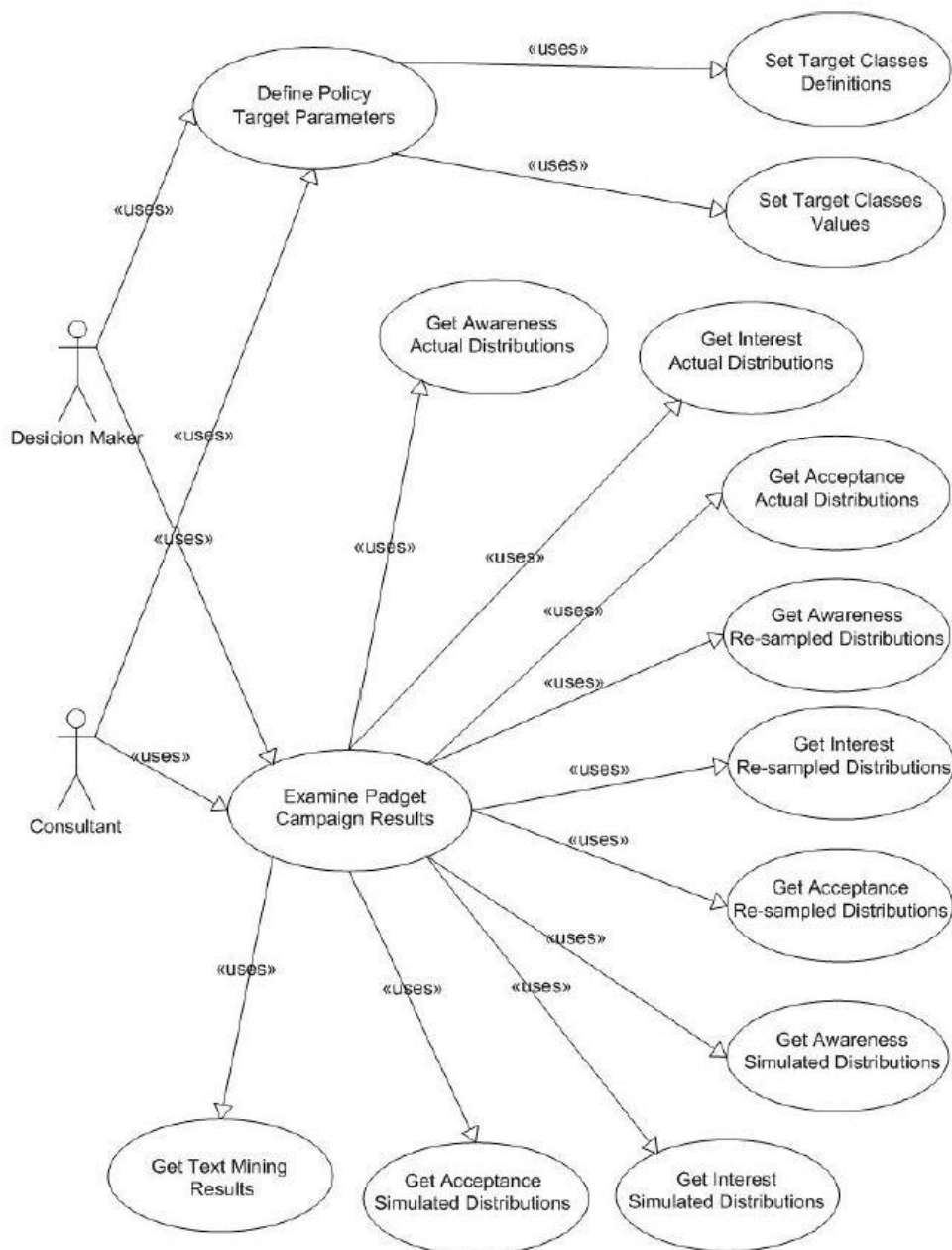
- Create Account (UC 45)
 - The PADGETS Initiator will have to create an account in order to be able to create/manage his padgets. A consultant will assist him to setup the account and demonstrate the padgets functionalities.
- Provide Credentials for SM (UC 46)
 - The PM has to provide his credentials for the various SM platforms that he wants to publish his padgets to.
- Create Account in SM (UC 47)
 - The PM might have to create new accounts in SM in case he doesn't already have.

- Search (UC 48)
 - A search page permits to filter padgets that the PM has instantiated.
 - A list view of the padgets that the PM has initiated will be available.
- View Results (UC 49)
 - A dashboard of the PM' s padgets will provide results about the published padgets.
- View Likes (UC 50)
 - The Likes/Dislikes to a published padget.
- View Comments (UC 51)
 - The list of comments for a published padget.
- Respond to comments (UC 52)
 - The PM will be able to send a respond to a comment made to one of her padgets.
- View Analysis (UC 53)
 - This is a general use case containing all other information that Padgets will provide.
- Create Padget (UC 54)
 - The creation of the padget will be an easy process of graphically adding the pieces of a padget together.
- Provide title (UC 55)
 - The title of the padget is the first and maybe more important part of the padget.
- Provide message (UC 56)
 - The message is also an obligatory field that must be provided.
- Select categories (UC 57)
 - The padget can be associated to many relevant categories.
- Bind SM (UC 58)
 - The PM will be able to bind message to many different SM platforms.
- Preview padgets in SM (UC 59)
 - The PM will be able to preview the message in each SM platforms.
- Add image/video/audio (UC 60)
 - An image/video/audio can be attached to the padget taken by a gallery or a web page.
- Add link (UC 61)

- A link can be inserted in a padget typing it or selecting page from internet.

- Add poll/survey (UC 62)
 - A poll/survey can be associated with a padget in order to get people's opinion on an issue.

Padgets Initiator / Consultant (Decision Model Service)[5]



3.17. - Use Case διάγραμμα του Padgets Initiator / Consultant (Decision Model Service)

- Define Policy Target Parameters (UC 63)
 - Settlement of parameters which are necessary in light of data analysis, stratification and re-sampling.
- Set Target Classes Definitions (UC 64)
 - Setting of the number of classes to be used in the stratification of sample data and subsequent specification of socio-demographic features pertaining to each class.
- Set Target Classes Value (UC 65)
 - Input of static data referring to socio-demographic variables of actual End User (stakeholders) affected by the public policy under examination.
- Examine Padget Campaign Results (UC 66)
 - View of outcomes stemming from Padget consultation and subsequent data elaboration.
- Get Awareness Actual Distributions (UC 67)
 - Request for indicators and metrics related to the mere passive reception of the policy message in Social Media environments evaluated over the main categories of stakeholders identifiable according to socio-demographic specified variables (raw data).
- Get Interest Actual Distributions (UC 68)
 - Request for indicators and metrics related to the proactive spreading of the policy message in Social Media domain evaluated over the main categories of stakeholders identifiable according to socio-demographic specified variables.
- Get Acceptance Actual Distributions (UC 69)
 - Request for indicators and metrics related to data collected inside Padget evaluated over the main categories of stakeholders identifiable according to demographic specified variables.
- Get Awareness Re-sampled Distributions (UC 70)
 - Request for indicators and metrics related to the mere passive reception of the policy message in Social Media environments evaluated through a data projection into the real world (raw data re-sampling).
- Get Awareness Re-sampled Distributions (UC 71)
 - Request for indicators and metrics related to the mere passive reception of the policy message in Social Media environments evaluated through a data projection into the real world (raw data re-sampling).

- Get Interest Re-sampled Distributions (UC 72)
 - Request for indicators and metrics related to the proactive spreading of the policy message in Social Media domain evaluated through a data projection into the real world.
- Get Acceptance Re-sampled Distributions (UC 73)
 - Request for indicators and metrics related to data collected inside Padget evaluated through a data projection into the real world.
- Get Awareness Simulated Distributions (UC 74)
 - Request for scenario information related to the mere passive reception of the policy message in Social Media environments evaluated through simulation of phenomena evolution in the near future.
- Get Interest Simulated Distributions (UC 75)
 - Request for scenario information related to the proactive spreading of the policy message in Social Media domain evaluated through simulation of phenomena evolution in the near future.
- Get Acceptance Simulated Distributions (UC 76)
 - Request for scenario information related to themes debated in Padget consultation evaluated through simulation of phenomena evolution in the near future.
- Get Text Mining Results (UC 77)
 - Request for results of public opinion analysis which are performed through automated processing of opinions, sentiments and emotions found, expressed and implied in text (e.g. pattern recognition, subjectivity classification, and sentence level sentiment classification).

Παρακάτω παρουσιάζονται με αναλυτική περιγραφή οι περιπτώσεις χρήσεις[5] που καταγράφηκαν παραπάνω.

UC Code	Actor / Roles	Περιγραφή
UC14 UC15 UC45	PADGET Initiator	<p><u>Pre-Conditions:</u> Download the application</p> <p><u>Flow:</u> The policy-maker registers himself with the system providing Username, Password, contact details, etc. and accepts the system's Terms of Use.</p> <p><u>Post-Conditions:</u> The Policy Maker can login to the system assuming the role of PADGET Initiator.</p>

<p>UC16 UC17 UC46 UC47</p>	<p>PADGET Initiator</p>	<p><i>Pre-Conditions:</i> The Policy Maker is already registered with the system</p> <p><i>Flow:</i> The Policy Maker logs into the system. The policy maker uses his Social Media accounts, by selecting the corresponding platform (i.e. Facebook, Twitter, Youtube, etc.) and provides his login credentials (i.e. Username/Password)</p> <p><i>Post-Conditions:</i> The system can access the Policy Maker's Social Media Accounts when he is logged in.</p>
<p>UC25 UC26 UC27 UC28 UC29 UC30 UC31 UC32 UC33 UC54 UC55 UC56 UC57 UC60 UC61 UC62</p>	<p>PADGET Initiator</p>	<p><i>Pre-Conditions:</i> The Policy Maker is already registered with the system</p> <p><i>Flow:</i> The Policy Maker logs into the system. The policy-maker creates a PADGET for his policy. He categorizes his PADGET according to the policy it contains. He creates a (short) title. Optionally, he creates an analysis for his policy, or he attaches a Doc or a link to an external site/blog or a link to a Youtube Video or a link to Picasa Photo-stream. If the Policy Maker selects a Native application (the mobile app.) he will be able to add Polls and Surveys</p> <p><i>Post-Conditions:</i> Through the system, the Initiator will be able to monitor the new 'padget' and the related content in the various SM platforms where this is published.</p>
<p>UC34 UC35 UC36 UC58 UC59</p>	<p>PADGET Initiator Consultant</p>	<p><i>Pre-Conditions:</i> The Policy Maker is already registered in the system and he has created a new 'padget'.</p> <p><i>Flow:</i> The Policy Maker selects the targeted Social Media and/or native applications in order to publish his new 'padget'. The system 'breaks-down' the relevant padget content into different content-parts that fit the requirements of each target platform. The system creates a preview of what part of the message will be published in the corresponding social media. The system publishes each content part to the targeted platform.</p> <p><i>Post-Conditions:</i> The system can monitor all interactions with the several 'padget' content-parts on the various social media platforms where these are published.</p>
		<p><i>Pre-Conditions:</i> n/a</p>

UC1 UC2 UC3 UC4 UC5 UC6 UC7 UC8 UC9 UC10 UC11 UC12 UC13 UC22 UC52	Citizen Decision Maker	<p><i>Flow:</i> A Citizen - that follows any of the Policy Maker's Social Media accounts where the padget is published - comes across a specific padget content-part in a particular platform. This could be a status update in the Policy Maker's Facebook page, a tweet in his Twitter account, a post in his Blog, a video in his Youtube channel, etc. The Citizen decides to interact and offer his opinion on the fraction of the policy message he comes across. For that end he is using any of the available interaction capabilities provided by the platform where he comes across the content part of the padget. For example this could be "liking" a post or "commenting" on it in the Policy Maker's Facebook Page, 'retweeting' or "re-plying" at it in Twitter, "commenting" or "rating" a video in the Youtube channel, etc. The Policy Maker is notified when a Citizen interacts with his padget. This can happen by all different kinds of notifications provided natively by the social media platforms and dedicated client apps that he might be using. The Policy Maker decides to respond to it. This can happen also by all kinds of different tools available today to interact with social media platforms.</p> <p><i>Post-Conditions:</i> The system can track all (Citizen and Policy Maker) interactions with the several padget content-parts from the various social media platforms.</p>
UC18 UC19 UC20 UC21 UC23 UC24 UC48 UC49 UC50 UC51 UC53		<p><i>Pre-Conditions:</i> The Policy Maker is already registered. He has created a padget. He has published this padget.</p> <p><i>Flow:</i> The Policy Maker logs into the system. He reviews the 'padgets' that he has published and which are currently monitored by the system. He selects a particular padget. He is able to get an aggregated view of all citizens' interactions with the particular his 'padgets' - i.e. "comments" from Facebook, "comments" from Blog post, "comments" in Youtube, results from the mobile app etc.</p> <p><i>Post-Conditions:</i> n/a</p>
UC63 UC64		<p><i>Pre-Conditions:</i> The Policy Maker is already registered. He has created a padget. He has</p>

UC65 UC66 UC67 UC68 UC69 UC70 UC71 UC72 UC73 UC74 UC75 UC76 UC77		<p>published this padget.</p> <p><i>Flow:</i> The Policy Maker logs into the system. He reviews the ‘padgets’ that he has published and which are currently monitored by the system. He selects a particular padget. He is able to get an aggregated view of all citizens’ interactions through the services provided by the PADGETS Decision Model Service</p> <p><i>Post-Conditions:</i> n/a</p>
UC18 UC37		<p><i>Pre-Conditions:</i> The Policy Maker is already registered. He has created a padget. He has published this padget.</p> <p><i>Flow:</i> The Policy Maker logs into the system. The Policy Maker logs into the system. He reviews the ‘padgets’ that he has published. He selects a particular padget. He discontinues the padget.</p> <p><i>Post-Conditions:</i> The system stops keeping track of any kind of citizen interaction with the particular padget. The system discontinues also the corresponding iGoogle Gadget if one exists.</p>
UC38 UC39 UC40 UC41 UC42 UC44	System Administrator	<p><i>Pre-Conditions:</i> To have administration rights</p> <p><i>Flow:</i> The System administrator will ensure the systems proper operation. He will handle roles, users, run statistics etc.</p> <p><i>Post-Conditions:</i> n/a</p>

3.18. - Περιπτώσεις χρήσεις

Κεφάλαιο 4^ο - Ανάλυση των API των social media

Στην ακόλουθη ενότητα θα προβούμε σε ενδελεχή μελέτη της τρέχουσας κατάστασης των APIs[6] των μέσων κοινωνικής δικτύωσης που αποτελούν άμεσους στόχους της εφαρμογής **SocialCampaign** καθώς και επίσης του Google Charts API[7] το οποίο θα χρησιμοποιηθεί για την παρουσίαση των αποτελεσμάτων. Επιπλέον γίνεται προσπάθεια για τον προσδιορισμό και την καταγραφή των διαθέσιμων στο κοινό δυνατοτήτων που προσφέρουν τα API προκειμένου να επιτρέψει την αυτοματοποιημένη αλληλεπίδραση μεταξύ των social media με ανεξάρτητες εξωτερικές εφαρμογές.

Για κάθε μία από τις επιλεγμένες πλατφόρμες μέσων κοινωνικής δικτύωσης πραγματοποιείται ανάλυση προκειμένου να προσδιοριστούν οι τρόποι για την αλληλεπίδραση κάποιας εξωτερικής εφαρμογής με τις λειτουργίες των μέσων κοινωνικής δικτύωσης μέσα από τα προσφερόμενα APIs τους. Ο στόχος είναι να διευκρινιστούν για κάθε πλατφόρμα τα ακόλουθα χαρακτηριστικά:

- Διαθέσιμα APIs και τα είδη επικοινωνίας (δραστηριότητες) που επιτρέπουν.
- Δυνατότητες για την προώθηση του περιεχομένου σε αυτά μέσω των APIs τους, όπου με τον όρο της προώθησης περιλαμβάνονται όλες οι πιθανές δραστηριότητες των χρηστών που έχουν ως αποτέλεσμα τη προσθήκη κάποιου είδους περιεχόμενο σε αυτές τις πλατφόρμες που αντιπροσωπεύουν τη γνώμη τους ή τις θέσεις τους.
- Δυνατότητες για την ανάκτηση του περιεχομένου από αυτά μέσω των APIs τους, όπου με τον όρο της ανάκτησης εννοείται κάθε είδους πληροφορία που προκύπτει από τις δραστηριότητες της αλληλεπίδρασης των χρηστών με το δημοσιευμένο περιεχόμενο.
- Δυνατότητες για την ανάπτυξη εφαρμογών στο περιβάλλον τους προσφέροντας στους χρήστες την απευθείας αλληλεπίδραση μαζί τους.
- Τα πρότυπα που υιοθετούν για να περιγράψουν το περιεχόμενό τους.

Οι λειτουργίες των APIs ταξινομούνται ως μέθοδοι για την προώθηση του περιεχομένου (ανέβασμα, τροποποίηση του περιεχομένου κ.λπ.) και μέθοδοι για την ανάκτηση του περιεχομένου (λήψη περιεχομένου, αναζήτηση, κλπ.). Για κάθε μέθοδο, η ανάλυση περιέχει πληροφορίες σχετικά με τη λειτουργικότητά της και τις παραμέτρους της. Όπου κρίνεται σκόπιμο ένα μικρό δείγμα κώδικα συμπεριλαμβάνεται.

Τα μέσα κοινωνικής δικτύωσης που επιλέχτηκαν για ανάλυση των APIs τους είναι τα εξής: Blogger, Picasa, Youtube, Twitter, Facebook και LinkedIn.

4.1. - Blogger

Η σύνδεση εξωτερικών εφαρμογών με τις δυνατότητες του Blogger επιτυγχάνεται μέσα από τη χρήση του Blogger Data API. Το Blogger Data API επιτρέπει σε εξωτερικές εφαρμογές να παρακολουθούν και να ενημερώνουν τα περιεχόμενα του Blogger βασιζόμενο στα πρότυπα του Google Data API. Μια εξωτερική εφαρμογή μπορεί να χρησιμοποιήσει το API του Blogger για να δημιουργήσει νέα blog posts, να επεξεργαστεί ή να διαγράψει ήδη υπάρχοντα και να θέσει ποικίλα ερωτήματα που σχετίζονται με το δημοσιευμένο post.

Πιο συγκεκριμένα οι δυνατότητες που υποστηρίζει η τρέχουσα έκδοση του API είναι:

- Δημιουργία posts
- Ανάκτηση posts
- Επεξεργασία posts
- Διαγραφή posts
- Καταχώρηση comments
- Ανάκτηση comments

Η τρέχουσα έκδοση του Blogger Data API υποστηρίζει διάφορες βιβλιοθήκες, όπως Java, .NET, PHP, JavaScript, Python). Το πρωτόκολλο που χρησιμοποιείται είναι η XML και HTTP. Στην ενότητα αυτή, θα περιγραφεί μόνο η πλατφόρμα του πρωτοκόλλου. Θα πρέπει να σημειωθεί, ωστόσο, ότι και οι δύο πλατφόρμες προσφέρουν τις ίδιες δυνατότητες για την προώθηση και την ανάκτηση του περιεχομένου. Το κύριο σημείο αναφοράς που χρησιμοποιείται για το API και των συναφών βιβλιοθηκών είναι το Google Developer's Guide Version 2.0, το οποίο περιέχει την πλήρη τεκμηρίωση και δείγματα κώδικα.



API Method:	Create & Publish post [3]		
Description:	Creates and publishes new blog entries, as well as drafts of entries. First, an XML representation of the post to publish must be created. This XML needs to be in the form of an Atom <entry> element.		
URL:	http://www.blogger.com/feeds/blogID/posts/default		
Formats:	Xml, Atom 1.0		
HTTP Method(s):	PUT		
Requires Authentication:	Yes		
API restrictions:	No		
Parameters:	Required	Type	Description
blogID	Yes	int	The blog id
Title (in xml)	Yes	string	The post title
Content (in xml)	Yes	html	The post content
Category (in xml)	Optional	Category Term	One or more categories the post is related to.
<app:control> (in xml) [Blogger2]	Optional	Yes/No	"Yes" Specifies that the post should be archived as a draft rather than a normal post. "No" turns an <u>existing</u> draft post into a published post.
			<pre> <app:control xmlns:app='http://www.w 3.org/2007/app'> <app:draft>yes</app:draft> </app:control> </pre>

API Method:	Update post [5]
Description:	To update an existing blog post, the client must retrieve the entry to be updated, modify it and send a PUT request, with the updated entry in the message body, to the edit URL of the post. The <id> value in the entry must exactly match the <id> of the existing entry. The updated entry should preserve all the elements that were in the original entry in order to comply with the Google Data API (To ensure forward compatibility)
URL:	http://www.blogger.com/feeds/blogID/posts/default/postID
Formats:	Xml, Atom 1.0
HTTP Method(s):	PUT
Requires Authentication:	Yes
API restrictions:	No

Parameters:	Required	Type	Description
blogID	Yes	int	The blog id
postID	Yes	int	The post id
id (in xml)	Yes	int	The post id
Title (in xml)	Yes	string	The post title
Content (in xml)	Yes	html	The post content
Category (in xml)	Optional	Category Term	One or more categories the post is related to.
Author.Name(in xml)	Optional	string	The post author's name.
Author.email(in xml)	Optional	email	The post author's email.
Author.uri(in xml)	Optional	uri	The post author's uri.
<app:control> (in xml)	Optional	Yes/No	"Yes" Specifies that the post should be archived as a draft rather than a normal post. "No" turns an <u>existing</u> draft blog post into a published post.
			<pre> <app:control xmlns:app="http://www.w 3.org/2007/app"> <app:draft>yes</app:draft> </app:control> </pre>

API Method:	Delete post [6]		
Description:	Deletes a post		
URL:	http://www.blogger.com/feeds/blogID/posts/default/postID		
Formats:	http		
HTTP Method(s):	DELETE (Some firewalls block HTTP DELETE messages. Alternatively, the client can include a X-HTTP-Method-Override: DELETE header in a POST request.)		
Requires Authentication:	Yes		
API restrictions:	No		
Parameters:	Required	Type	Description
blogID	Yes	int	The blog id
postID	Yes	int	The post id

API Method:	Create & Publish Comment [7]		
Description:	Creates and publishes a comment. The client can only post comments to a blog owned by the authenticated user. All new comments will appear as if they were created by the currently authenticated user.		
URL:	http://www.blogger.com/feeds/blogID/postID/comments/default		
Formats:	Xml, Atom 1.0		
HTTP Method(s):	POST		
Requires Authentication:	Yes		
API restrictions:	No		
Parameters:	Required	Type	Description
blogID	Yes	int	The blog id
postID	Yes	int	The post id
Title (in xml)	Yes	string	The comment title.
content (in xml)	Yes	html	The comment content.

API Method:	Delete Comment [8]		
Description:	Deletes a comment		
URL:	http://www.blogger.com/feeds/blogID/postID/comments/default/commentID		
Formats:	http		
HTTP Method(s):	DELETE (Some firewalls block HTTP DELETE messages. Alternatively, the client can include a X-HTTP-Method-Override: DELETE header in a POST request.)		
Requires Authentication:	Yes		
API restrictions:	No		
Parameters:	Required	Type	Description
blogID	Yes	int	The blog id
postID	Yes	int	The post id
CommentID	Yes	int	The comment id

API Method:	Retrieve List of blogs [9]		
Description:	Provides a feed that lists the blogs for a particular user; that feed is known as a "metafeed"		
URL:	http://www.blogger.com/feeds/profileID/blogs		
Formats:	Xml, Atom 1.0		
HTTP Method(s):	GET		
Requires Authentication:	No -It is possible to query the API without authentication. Therefore, there is no need to set the Authorization parameter when retrieving blog posts from a public blog.		
API restrictions:	No		
Parameters:	Required	Type	Description
profileID	Yes	int	The number in the URL for the user's profile page. One can also substitute "default" for the user ID, which tells Blogger to return the list of blogs for the user whose credentials accompany the request.

API Method:	Retrieve blog posts using query parameters [11]		
Description:	Allows requesting a set of entries that match specified criteria, such as requesting blog posts published or updated in a given date range.		
URL:	http://www.blogger.com/feeds/blogID/posts/default		
Formats:	Xml, Atom 1.0		
HTTP Method(s):	GET		
Requires Authentication:	No (It is possible to query a Blogger public feed without authentication. Therefore, there is no need to set the "Authorization" parameter when retrieving blog posts from a public blog.)		
API restrictions:	No		
Parameters:	Required	Type	Description
blogID	Yes	int	The blog id
alt	Optional	atom/rss(default)	The type of feed to return
/category	Optional	string	Specifies categories to filter the feed results.
orderby	Optional	lastmodified(default), starttime, updated	The order in which to return entries
published-min	Optional	datetime	Lower bound on entry publication dates.
published-max	Optional	datetime	Upper bound on entry publication dates.
start-index	Optional	int(1-based)	The 1-based index of the first result to be
			retrieved (for paging).
updated-min	Optional	int	Lower bound on entry update dates. (This query parameter is ignored unless the "orderby" parameter is set to "updated".)
updated-max	Optional	int	Upper bound on entry update dates. (This query parameter is ignored unless the "orderby" parameter is set to "updated".)

API Method:	Retrieve all blog comments [12]		
Description:	Gets the comments for all posts of a blog.		
URL:	<i>http://www.blogger.com/feeds/blogID/comments/default</i>		
Formats:	Xml, Atom 1.0		
HTTP Method(s):	GET		
Requires Authentication:	No (It is possible to query a Blogger public feed without authentication. Therefore, there is no need to set the "Authorization" parameter when retrieving blog posts from a public blog.)		
API restrictions:	No		
Parameters:	Required	Type	Description
blogID	Yes	int	The blog id

API Method:	Retrieve all comments from a post [12]		
Description:	Gets the comments for the specified post of a blog.		
URL:	<i>http://www.blogger.com/feeds/blogID/postID/comments/defaultz</i>		
Formats:	Xml, Atom 1.0		
HTTP Method(s):	GET		
Requires Authentication:	No (It is possible to query a Blogger public feed without authentication. Therefore, there is no need to set the "Authorization" parameter when retrieving blog posts from a public blog.)		
API restrictions:	No		
Parameters:	Required	Type	Description
blogID	Yes	int	The blog id
postID	Yes	int	The post id

4.2. - Picasa

Το Picasa Web Albums Data API επιτρέπει στις client εφαρμογές να δουν και να ενημερώσουν λευκώματα, φωτογραφίες και σχόλια σύμφωνα με το πρότυπο του Google Data API. Μια client εφαρμογή μπορεί να χρησιμοποιήσει το Picasa Web Albums Data API για τη δημιουργία νέων άλμπουμ, τη φόρτωση νέων φωτογραφιών, τη προσθήκη σχολίων, την επεξεργασία ή διαγραφή άλμπουμ, φωτογραφιών και σχολίων και την αναζήτηση φωτογραφιών με συγκεκριμένα κριτήρια.

Υπάρχουν πολλές υποστηριζόμενες εκδόσεις του Picasa Web Albums Data API. Η τεκμηρίωση για κάθε έκδοση περιλαμβάνει:

- Γράφοντας κώδικα που χρησιμοποιεί το Google Data πρωτόκολλο άμεσα. Ο κώδικας αυτός έχει πρόσβαση στο API χρησιμοποιώντας HTTP αιτήματα, όπως GET ή POST. Με αυτά τα αιτήματα, τα δεδομένα που αποθηκεύονται από το προϊόν της Google μεταφέρονται μπρος - πίσω με τη μορφή feeds δεδομένων. Οι πηγές δεδομένων είναι απλά δομημένες λίστες που περιέχουν τα σχετικά δεδομένα. Για ιστορικούς λόγους, η αρχική μορφή του feed ήταν σε AtomPub XML, αλλά τώρα πλέον, JavaScript Object Notation Object, (JSON) υποστηρίζεται επίσης ως μια εναλλακτική μορφή.
- Χρησιμοποιώντας μια βιβλιοθήκη client για μια συγκεκριμένη πλατφόρμα ανάπτυξης. Αντί να γραφεί κώδικας που θα κάνει HTTP requests απευθείας, η εφαρμογή client μπορεί αντί αυτού να προγραμματιστεί χρησιμοποιώντας μια από τις γλώσσες προγραμματισμού που είναι διαθέσιμες στο σύνολο των client βιβλιοθηκών που παρέχονται. Ο κώδικας είναι γραμμένος σε ένα πιο θεωρητικό επίπεδο χρησιμοποιώντας τις μεθόδους και τις κλάσεις της συγκεκριμένης γλώσσας όπως αυτά παρέχονται από τη client βιβλιοθήκη. Υπάρχουν Developer's guides για διάφορες γλώσσες (Java, Objective-C, NET, PHP, Python)

Για να στείλει ένα request ή να ανεβάσει ένα αρχείο, μια client εφαρμογή χρειάζεται ένα διακριτικό ταυτότητας (token). Η επιλογή της μεθόδου αυθεντικοποίησης πρέπει να επιλέγεται ανάλογα από τον τύπο της client εφαρμογής: single-user desktop εφαρμογές θα πρέπει να χρησιμοποιούν την ClientLogin μέθοδο, ενώ οι multi-user web εφαρμογές θα πρέπει να χρησιμοποιούν τη μέθοδο AuthSub. Ανεξάρτητα από τη μέθοδο που θα επιλεγεί, υπάρχει η δυνατότητα είτε να στέλνονται "raw" HTTP requests είτε να χρησιμοποιούνται οι client βιβλιοθήκες της Google η οποία θα χειριστεί την εξακρίβωση της γνησιότητας.

API Method:	Post photo [24]		
Description:	<p>There are two ways to add an image to an album using the Data API:</p> <ul style="list-style-type: none"> Uploading the binary image data along with its metadata. To do this, a MIME content type "multipart/related" must be used; send image metadata in one part of the POST body, and binary-encoded image data in another part. This is the preferred approach. The message format for the body of the POST is as following: <pre> Content-Type: multipart/related; boundary="END_OF_PART" Content-Length: 423478347 MIME-version: 1.0 Media multipart posting --END_OF_PART Content-Type: application/atom+xml <entry xmlns='http://www.w3.org/2005/Atom'> <title>pls-to-love-realcat.jpg</title> <summary>Real cat wants attention too.</summary> <category scheme="http://schemas.google.com/g/2005#kind" term="http://schemas.google.com/photos/2007#photo"/> </entry> --END_OF_PART Content-Type: image/jpeg ...binary image data... </pre> <ul style="list-style-type: none"> Uploading the binary image data without the metadata. The message's format for the body of the POST is as following: <pre> Content-Type: image/jpeg Content-Length: 47899 Slug: pls-to-love-realcat.jpg ...binary image data goes here... </pre>		
URL:	<i>http://picasaweb.google.com/data/feed/api/user/userID/albumid/albumID</i>		
Formats:	MIME , XML/Atom 1.0 While all images appearing on the Picasa Web Albums site are in the JPEG format, images of any of the following types can be uploaded using the API:		
	<ul style="list-style-type: none"> image/bmp image/gif image/jpeg image/png 		
HTTP Method(s):	POST		
Requires Authentication:	Yes [33]		
API restrictions:	No		
Parameters:	Required	Type	Description
userID	Yes	string	The user where the image will be posted. This parameter is passed through the HTTP POST request.

albumID	Yes*	string	The album where the image will be posted. This parameter is passed through the HTTP POST request. * Wanted to post an image, but not to the hassle of requiring the user of the application to choose an album, the image can be posted to the "Drop Box." This special album will automatically be created the first time it is used to store an image. To post to the Drop Box, it should be used an <i>albumID</i> value of default: <i>http://picasaweb.google.com/data/feed/api/user/default/albumid/default.</i>
Entry Title	Yes	text	Contains the desirable filename to be used for the image. This parameter is passed through the <entry> tag in the MIME message.
Entry Summary	No	text	A sort description for the new entry that is posted. This parameter is passed through the <entry> tag in the MIME message.
Entry Author	No	text	The author of the entry. This parameter is passed through the <entry> tag in the MIME message.
Entry Category	No	text	The category of the entry, referencing to the schemas.google. This parameter is passed through the <entry> tag in the MIME message.
Content	Yes	binary data	The RAW data for the image to be uploaded. It is on the "Datatype: image/..." part of the MIME message.

API Method:	Update photo [25]
Description:	This command replaces the metadata and/or the binary image data for an image. In each case, to make sure that the update doesn't overwrite another client's changes, an HTTP <i>If-Match</i> header, which contains the original image ETag value, should be included. The original image ETag value can be determined by examining the <entry> element's <i>gd:etag</i> attribute. If wanted to update the image regardless of whether someone else has updated it since it was retrieved, then <i>If-Match: *</i> must be used and the ETag should not be included.
	<ul style="list-style-type: none"> • Updating an image and its metadata: To replace both the binary data and the metadata for the image that was inserted earlier, an HTTP request like the following must be used: <pre>PUT http://picasaweb.google.com/data/media/api/user/userID/albumid/albumID/photoID/photoID</pre> <p>The URL is the value of the <link rel="edit-media"> tag that was returned after the earlier POST.</p> <p>In the body of the PUT, the updated metadata and image data must be included, in the same multipart format that was used to do the POST with metadata.</p> <p>Note: With PUT, a partial update of an entry cannot be done; the full entry data have to be sent to replace the existing data. However, a PUT method to send only the metadata or only the image can be used, as described in the following sections.</p>

- **Updating only the photo**

To replace only the photo's binary data, the following HTTP request can be used:

```
PUT
http://picasaweb.google.com/data/media/api/user/userID/albumID/albumID/photoid/photoID
```

The URL is the value of the `<link rel="edit-media">` tag that was returned after the earlier POST command.

In the body of the PUT, the replacement image data must be included, in the same format that was used to do the POST message without metadata.

- **Updating only the metadata**

To replace only the image metadata (and not the image itself), the steps that would be followed to send an update to a non-media feed must be followed. In particular, the following HTTP request must be sent:

```
PUT
http://picasaweb.google.com/data/entry/api/user/userID/albumID/albumID/photoid/photoID
```

The URL is the value of the `<link rel="edit">` tag that was returned after the earlier POST.

Parameters:	Required	Type	Description	
<i>userID</i>	Yes	string	The user where the image will be updated. This parameter is passed through the HTTP PUT request.	These are included in the response feed of the POST methods, as Feed HTML Link: <code>/feed/link[[]@rel="alternate"]\ [[]@type="text/html"]/@href</code>
<i>albumID</i>	Yes	string	The album where there is the image to be updated. This parameter is passed through the HTTP PUT request.	
<i>photoID</i>	Yes	string	The image that will be updated. This parameter is passed through the HTTP PUT request.	
Entry Title	Yes	text	Contains the desirable filename to be updated for the image.	
Entry Summary	No	text	A sort description for the entry that will be updated. This parameter is passed through the <code><entry></code> tag in the MIME message.	
Entry Author	No	text	The updated author of the entry. This parameter is passed through the <code><entry></code> tag in the MIME message.	
Entry Category	No	text	The updated category of the entry, referencing to the <code>schemas.google</code> . This parameter is passed through the <code><entry></code> tag in the MIME message.	
Content	Yes	binary data	The RAW data for the image to be updated. It is on the "Datatype: image/..." part of the MIME message.	

API Method:	Delete photo [26]		
Description:	<p>To delete an image and its metadata, a HTTP DELETE request must be sent to either the edit link or the edit-media link. In both cases, the image and metadata are deleted.</p> <p>For example, to delete the image from the previous methods:</p> <pre>DELETE http://picasaweb.google.com/data/entry/api/user/userID/albumid/albumID/photoID/photoID</pre> <p>Wanting to make sure that it is not deleted an image that has been changed by another client since was retrieved, an HTTP <i>If-Match</i> header that contains the original image ETag value must be included. The original image ETag value can be determined by examining the <entry> element's <i>gd:etag</i> attribute.</p> <p>Wanting to delete the image regardless of whether someone else has updated it since it was retrieved, then an <i>If-Match: *</i> must be included and not the ETag. (In this case, you don't need to retrieve the image before deleting it.)</p>		
URL:	http://picasaweb.google.com/data/entry/api/user/ <i>userID</i> /albumid/ <i>albumID</i> /photoID/ <i>photoID</i>		
Formats:	XML/Atom 1.0		
HTTP Method(s):	DELETE		
Requires Authentication:	Yes [33]		
API restrictions:	No		
Parameters:	Required	Type	Description
<i>userID</i>	Yes	string	The user where the image will be deleted. This parameter is passed through the HTTP PUT request.
<i>albumID</i>	Yes	string	The album where there is the image to be deleted. This parameter is passed through the HTTP PUT request.
<i>photoID</i>	Yes	string	The image that will be deleted. This parameter is passed through the HTTP PUT request.

API Method:	Add comment [28]			
Description:	<p>Example: The following request adds the comment "great photo!" to the image identified by <i>photoID</i> in the album with ID <i>albumID</i> owned by <i>userID</i>.</p> <pre> POST http://picasaweb.google.com/data/feed/api/user/userID/albumid/albumID /photoID/photoID <entry xmlns="http://www.w3.org/2005/Atom"> <content>great photo!</content> <category scheme="http://schemas.google.com/g/2005#kind" term="http://schemas.google.com/photos/2007#comment"/> </entry> </pre>			
URL:	http://picasaweb.google.com/data/feed/api/user/userID/albumid/albumID/photoID/photoID			
Formats:	MIME , XML/Atom 1.0			
HTTP Method(s):	POST			
Requires Authentication:	Yes [33]			
API restrictions:	No			
Parameters:	Required	Type	Description	
userID	Yes	string	The user the image/video belongs to. This parameter is passed through the HTTP POST request.	There are included in the response feed of the image POST method, as Feed HTML Link: /feed/link[[]@rel="alternate"]\ [[]@type="text/html"]/@href
albumID	Yes	string	The album where the image/video is located. This parameter is passed through the HTTP POST request.	
photoID	Yes	string	The image/video that will be commented. This parameter is passed through the HTTP POST request.	
Entry Category	No	text	The category of the entry, referencing to the schemas.google. This parameter is passed through the <entry> tag in the MIME message.	
Content	Yes	binary data	The RAW data for the image to be uploaded. It is on the "Datatype: image/..." part of the MIME message.	

API Method:	Create Album [32]
Description:	<p>Example: This request creates an album by sending an authenticated POST request with an appropriately formed entry. To authenticate, one of the authentication mechanisms discussed in the Authentication section must be used. [33]</p> <pre data-bbox="491 577 1311 1146"> POST http://picasaweb.google.com/data/feed/api/user/userID <entry xmlns='http://www.w3.org/2005/Atom' xmlns:media='http://search.yahoo.com/mrss/' xmlns:gphoto='http://schemas.google.com/photos/2007'> <title type='text'>Trip To Italy</title> <summary type='text'>This was the recent trip I took to Italy.</summary> <gphoto:location>Italy</gphoto:location> <gphoto:access>public</gphoto:access> <gphoto:timestamp>1152255600000</gphoto:timestamp> <media:group> <media:keywords>italy, vacation</media:keywords> </media:group> <category scheme='http://schemas.google.com/g/2005#kind' term='http://schemas.google.com/photos/2007#album'></category> </entry> </pre> <p>Picasa Web Albums creates a new album using the data the user sent, then returns an HTTP 201 status code, along with a copy of the new album in the form of an <entry> element. The returned entry is similar to the one the users sent, but the returned one contains various elements added by the server, such as an <id> element.</p> <p>If the request fails for some reason, then a different status code may be returned.</p>
	<p>For information about the status codes used by the API, there is the HTTP status codes section of the Google Data APIs protocol reference document.</p>
URL:	<i>http://picasaweb.google.com/data/feed/api/user/userID</i>
Formats:	MIME , XML/Atom 1.0
HTTP Method(s):	POST
Requires Authentication:	Yes [33]
API restrictions:	No

Parameters:	Required	Type	Description
userID	Yes	string	The user where the image will be posted. This parameter is passed through the HTTP POST request.
Entry Title	Yes	text	Contains the desirable filename to be used for the image. This parameter is passed through the <entry> tag in the MIME message.
Entry Summary	No	text	A sort description for the new entry that is posted. This parameter is passed through the <entry> tag in the MIME message.
Entry Author	No	text	The author of the entry. This parameter is passed through the <entry> tag in the MIME message.
Entry Category	No	text	The category of the entry, referencing to the schemas.google. This parameter is passed through the <entry> tag in the MIME message.
gphoto:location	No	string	The user-specified location associated with the album. The <gphoto:location> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:location> element is valid with the album kind. [93]
gphoto:access	No	string	The album access level. In this document, access level is also referred to as "visibility." Valid values are public or private. The <gphoto:access> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:access> element is valid with the album kind. [94]
gphoto:timestamp	No	readonly Date	The time when the location was established.
media:keywords	No	string	Lists the tags associated with the entry. Contains a comma-separated list of tags that have been added to the image, or all tags that have been added to images in the album. Tags are limited to 128 characters and 100 tags per item. Tags use the UTF-8 character set, so they are not limited to ASCII. The <media:keywords> element appears as a child of a <media:group> element. i.e.:
			<media:keywords>kitty, cat, big dog, yarn, fluffy</media:keywords> [91],[92]

API Method:	Modify Album [98]		
Description:	<p>After retrieving an album entry, it can be modified by sending a PUT request, containing the new album data, to the URL given in the <link rel="edit"> element.</p> <p>To make sure that the update does not overwrite another client's changes, an HTTP <i>If-Match</i> header that contains the original album ETag value must be included. The original album ETag value can be determined by examining the <entry> element's gd:etag attribute.</p> <p>Wanting to update the album regardless of whether someone else has updated it since being retrieved, then <i>If-Match: *</i> must be used and not the ETag.</p>		
URL:	http://picasaweb.google.com/data/entry/api/user/ <i>userID</i> /albumid/ <i>albumID</i>		
Formats:	MIME , XML/Atom 1.0		
HTTP Method(s):	PUT		
Requires Authentication:	Yes [33]		
API restrictions:	No		
Parameters:	Required	Type	Description
userID	Yes	string	The user whose is the album to be modified. This parameter is passed through the HTTP PUT request.
albumID	Yes	string	The album to be modified. This parameter is passed through the HTTP PUT request.
Entry Title	Yes	text	Contains the desirable filename to be used for the image. This parameter is passed through the <entry> tag in the MIME message.
Entry Summary	No	text	A sort description for the new entry that is posted. This parameter is passed through the <entry> tag in the MIME message.
Entry Author	No	text	The author of the entry. This parameter is passed through the <entry> tag in the MIME message.
Entry Category	No	text	The category of the entry, referencing to the schemas.google. This parameter is passed through the <entry> tag in the MIME message.
gphoto:location	No	string	The user-specified location associated with the album. The <gphoto:location> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:location> element is valid with the album
			kind. [93]
gphoto:access	No	string	The album access level. In this document, access level is also referred to as "visibility." Valid values are public or private. The <gphoto:access> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:access> element is valid with the album kind. [94]
gphoto:timestamp	No	readonly Date	The time when the location was established.
media:keywords	No	string	Lists the tags associated with the entry. Contains a comma-separated list of tags that have been added to the image, or all tags that have been added to images in the album. Tags are limited to 128 characters and 100 tags per item. Tags use the UTF-8 character set, so they are not limited to ASCII. The <media:keywords> element appears as a child of a <media:group> element. i.e.: <media:keywords>kitty, cat, big dog, yarn, fluffy</media:keywords> [91][92]

API Method:	Delete Album [99]		
Description:	You can delete an album by sending an authenticated HTTP DELETE request to the URL inside of the <code><link rel="edit"></code> .		
URL:	http://picasaweb.google.com/data/entry/api/user/userID/albumid/albumID		
Formats:	XML/Atom 1.0		
HTTP Method(s):	DELETE		
Requires Authentication:	Yes [33]		
API restrictions:	No		
Parameters:	Required	Type	Description
userID	Yes	string	The user whose is the album to be modified. This parameter is passed through the HTTP PUT request.
albumID	Yes	string	The album to be modified. This parameter is passed through the HTTP PUT request.

API Method:	Request photos by Community Search[32]		
Description:	<p>There are different ways to retrieve images. The most common is to get a list of all of the images in an album, but you can also retrieve recent images from a user, or search images from the public albums of other users</p> <p>Listing images by community search: With the API, it is able to search images uploaded by other users, as long as they are in a public album, defining the query parameters.</p>		
URL:	http://picasaweb.google.com/data/feed/api/all?q=string&max-results=int		
Formats:	XML/Atom 1.0/RSS 2.0/JSON All images appearing on the Picasa Web Albums site are in the JPEG format.		
HTTP Method(s):	GET		
Requires Authentication:	Yes [33]		
API restrictions:	No		
Parameters:	Required	Type	Description
<code>/-/category</code>	No	string	<p>Lists each category as if it were part of the resource's URI, in the form <code>/categoryname/</code>—this is an exception to the usual <code>name=value</code> form. These parameters (all categories) must be listed before any other query parameters.</p> <p>Precede the first category with <code>/-/</code> to make clear that it's a category. For example, if Jo's feed has a category for entries about Fritz, it could request those entries like this: <code>"http://www.example.com/feeds/jo/-/Fritz"</code>.</p>

			<p>This allows the implementation to distinguish category-predicated query URIs from resource URIs.</p> <p>Queries are allowed on multiple categories by listing multiple category parameters, separated by slashes. The service returns all entries that match all of the categories (like using AND between terms). For example: <code>http://www.example.com/feeds/jo-/Fritz/Laurie</code> returns entries that match both categories.</p> <p>To do an OR between terms, use a pipe character (<code> </code>), URL-encoded as <code>%7C</code>. For example: <code>"http://www.example.com/feeds/jo-/Fritz%7CLaurie"</code> returns entries that match either category.</p> <p>An entry matches a specified category if the entry is in a category that has a matching term or label, as defined in the Atom specification. (Roughly, the "term" is the internal string used by the software to identify the category, while the "label" is the human-readable string presented to a user in a user interface.)</p> <p>To exclude entries that match a given category, use the form <code>/-categoryname/</code>. To query for a category that has a scheme—such as <code><category scheme="urn:google.com" term="public"/></code>—the scheme must be placed in curly braces before the category name. For example: <code>/{urn:google.com}public</code>. If the scheme contains a slash character (<code>/</code>) it should be URL-encoded as <code>%2F</code>. To match a category that has no scheme, use an empty pair of curly braces. Not specify in curly braces, then categories in any scheme will match.</p> <p>The above features can be combined. For example: <code>/A%7C-{urn:google.com}B/-C</code> means (A OR (NOT B)) AND (NOT C).</p>
category	No	string	<p>An alternative way to perform a category filter. The two methods are equivalent:</p> <ul style="list-style-type: none"> • To do an OR between terms, it needs a pipe character (<code> </code>), URL-encoded as <code>%7C</code>. For example: <code>"http://www.example.com/feeds?category=Fritz%7CLaurie"</code> returns entries that match either category. • To do an AND between terms, it needs a comma

			character (,). For example: "http://www.example.com/feeds?category=Fritz,Laurie" returns entries that match both categories.
max-results	No	int	Maximum number of results to be retrieved. For any service that has a default max-results value (to limit default feed size), you can specify a very large number if you want to receive the entire feed.
alt	No	string	Alternative representation type. <ul style="list-style-type: none"> • Not specifying an alt parameter, the service returns an Atom feed. This is equivalent to alt=atom. • alt=rss returns an RSS 2.0 result feed (for reads only). • alt=json returns a JSON representation of the feed. • alt=json-in-script Requests a response that wraps JSON in a script tag. • alt=atom-in-script Requests an Atom response that wraps an XML string in a script tag. • alt=rss-in-script Requests an RSS response that wraps an XML string in a script tag. • alt=atom-service Requests an Atom service document that describes the feed.
fields	No	string	Returns only the requested portions of the resource representation for improved performance and efficiency. See Partial response section. [97]
prettyprint	No	Boolean	If <i>prettyprint=true</i> , the XML returned by the server will be human readable (pretty printed). Default: <i>prettyprint=false</i>
author	No	string	The service returns entries where the author name and/or email address match the query string.
q	No	string	When creating a query, list search terms separated by spaces, in the form q=term1 term2 term3. (As with all of the query parameter values, the spaces must be URL encoded.) The service returns all entries that match all of the search terms (like using AND between terms). Like Google's web search, a service searches on complete words (and related words with the same stem), not substrings. To search for an exact phrase, enclose the phrase in quotation marks: q="exact phrase". To exclude entries that match a given term, use the form q=-term. The search is case-insensitive. Example: to search for all entries that contain the exact phrase "Elizabeth Bennet" and the word "Darcy" but don't contain the word "Austen", use the following
			query: ?q="Elizabeth Bennet" Darcy -Austen
strict	No	Boolean	Set <i>strict=true</i> to verify that each of your query parameters are recognized by the service. An error will be returned if a parameter is not recognized. Default: <i>strict=false</i>
published-min,published-max	No	Timestamp	Use the RFC 3339 timestamp format. For example: 2005-08-09T10:57:00-08:00. The lower bound is inclusive, whereas the upper bound is exclusive.

API Method:	Retrieve comments [101]		
Description:	<ul style="list-style-type: none"> • By user: This request returns a feed of the most recent comments on a userID's images. The string "default" can be used in place of a real userID, in which case the server will use the userID of the user credentials used to authenticate the request. • By photo: This request retrieves all of the comments associated with a particular image. The following response retrieves a feed of the comments on the image identified by photoID, inside of the album with ID albumID, owned by the user userID. 		
URL:	<i>by_user</i> : http://picasaweb.google.com/data/feed/api/user/default?kind=comment&max-results=int <i>by_photo</i> : http://picasaweb.google.com/data/feed/api/user/userID/albumid/albumID/photoID/photoID?kind=comment		
Formats:	XML/Atom 1.0/RSS 2.0/JSON All images appearing on the Picasa Web Albums site are in the JPEG format.		
HTTP Method(s):	GET		
Requires Authentication:	Yes [33]		
API restrictions:	No		
Parameters:	Required	Type	Description
userID	Yes	string	The user where the image will be posted. This parameter is passed through the HTTP GET request. If the value is "default", the authorized user is passed as parameter
albumID	Yes*	string	The album where the image is located. This parameter is passed through the HTTP GET request. *Only for the "by_photo" method.
photoID	Yes*	string	The image which comments will be retrieved. This parameter is passed through the HTTP GET request. *Only for the "by_photo" method.
start-index	No	int	1-based index of the first result to be retrieved
max-results	No	int	Maximum number of results to be retrieved. For any service that has a default max-results value (to limit default feed size), you can specify a very large number if you want to receive the entire feed.
alt	No	string	Alternative representation type. <ul style="list-style-type: none"> • Not specifying an alt parameter, the service returns an Atom feed. This is equivalent to alt=atom. • alt=rss returns an RSS 2.0 result feed (for reads only). • alt=json returns a JSON representation of the feed. • alt=json-in-script Requests a response that wraps JSON in a script tag. • alt=atom-in-script Requests an Atom response that wraps an XML string in a script tag. • alt=rss-in-script Requests an RSS response that wraps an XML string in a script tag. • alt=atom-service Requests an Atom service document that describes the feed.
fields	No	string	Returns only the requested portions of the resource representation for improved performance and efficiency. See Partial response section. [97]

4.3. - YouTube

Το YouTube ως υπηρεσία της Google χρησιμοποιεί το Google Data API όπως αυτό περιγράφεται στο Developers' Guide v.2.0. Το Data API επιτρέπει σε έναν εξωτερικό χρήστη να πραγματοποιήσει τις περισσότερες δυνατότητες από αυτές που μπορεί να χρησιμοποιήσει ένας YouTube χρήστης μέσα από τη σελίδα της υπηρεσίας. Επιπλέον είναι διαθέσιμα και επιπρόσθετα APIs, τα οποία υποστηρίζουν λιγότερες δυνατότητες, όπως το Player API που επιτρέπει σε χρήστες να χρησιμοποιήσουν τον YouTube Player και τα Widgets του YouTube τα οποία έχουν φορτωθεί σε μια εξωτερική σελίδα. Παρόλα αυτά, αυτού του είδους τα APIs είναι μόνο για να μεταφορτώνουν της λειτουργίες του YouTube σε τρίτες σελίδες και όχι για να προγραμματίζουν σε εξωτερικές εφαρμογές.



4.1. - YouTube Interface

API Method:	Upload video (Browser-based uploading) [103][104]
Description:	<p>The YouTube Data API provides two methods for uploading videos to YouTube. A newly uploaded video will be included in a user's uploaded videos feed a few minutes after the upload completes and YouTube finishes processing the video. In addition, a newly uploaded video will typically be included in search results a couple of hours after the upload completes and YouTube finishes processing the video. However, this delay may be longer during periods of heavy API server loads.</p> <p>Browser-based uploading allows programmers to accept video uploads from their users without having to handle, host or proxy the actual video files. They should choose a browser-based uploading process if they do not want to host or store the uploaded video files. The browser – based uploading works in 3 steps:</p> <p>Step 1-Uploading Video metadata: The example below shows the format of an API request that uses the AuthSub [105] authentication scheme for browser-based uploading :</p> <pre>POST /action/GetUploadToken HTTP/1.1 Host: gdata.youtube.com Authorization: AuthSub token="<authentication_token>" GData-Version: 2 X-GData-Key: key=<developer_key> Content-Length: <content_length> Content-Type: application/atom+xml; charset=UTF-8</pre> <p>API_XML_request</p> <p>Step 2 - Extracting values from the API response: When an Upload API request is submitted, the API returns an XML response that contains an upload URL and upload token that enable the user to upload the actual video file. It will be needed to extract the URL and token from the response and include these values in the form on the web page where the user submits the video file.</p>

	<pre><?xml version='1.0' encoding='UTF-8'?> <response> <url>http://uploads.gdata.youtube.com/action/FormDataUpload/AEF3 087AUD</url> <token>AEwbFAQEvf3xox...</token> </response></pre>		
	<p>Step 3- Uploading the video file: After extracting the upload URL and upload token from the API response, it is needed to display a form so that the user can upload the actual video file. The form must use the upload URL as the value of the <form> tag's action attribute and have a hidden input field containing the upload token. In addition, the form should verify that the user has selected a file to upload before allowing the user to actually submit the form.</p>		
	<pre><form action="URL?nexturl=http%3A%2F%2Fwww.example.com method="post" enctype="multipart/form-data" onsubmit="return checkForFile();" > <input id="file" type="file" name="file"/> <div id="errMsg" style="display:none;color:red"> You need to specify a file. </div> <input type="hidden" name="token" value="TOKEN"/> <input type="submit" value="go" /> </form></pre>		
	<p>Note: The form must follow some guidelines:</p> <ul style="list-style-type: none"> • The nexturl parameter must be added to the form's target URL. This parameter specifies the URL to which YouTube will redirect the user's browser when the user uploads his video file. After the video is uploaded in the browser, the user will be redirected to the nexturl URL. • The value of the <form> tag's enctype must be set to the attribute to multipart/form-data. • The <input> tag that identifies the file must be named file. • The <input> tag that contains the token must be named token. • The application should verify that the user has selected a file to upload before allowing the user to submit the form to upload the file. 		
URL:	It is extracted from the API response		
Formats:	XML/Atom 1.0		
HTTP Method(s):	POST		
Requires Authentication:	Yes [33]		
API restrictions:	No		
Parameters:	Required	Type	Description
Step1			

authentication_token	Yes	string	This value contains the authentication token for the programmer's request. The token will either be a ClientLogin token, an AuthSub single-use token, an AuthSub session token or an OAuth access token.
developer_key	Yes	string	This value uniquely identifies the application that is submitting the request to upload the video.
content_length	Yes	bytes	This value contains the length of the entire body of the HTTP POST request.
API_XML_Request: Author	No	text	(child of: entry. Children: name, uri?) Video metadata, uploaded on the first step. The <author> tag usually contains information about the YouTube user who owns a piece of video content.
API_XML_Request: Category	No	text	(child of: entry) Video metadata uploaded on the first step. The <category> tag specifies the category to which the entry belongs. <ul style="list-style-type: none"> When the <category> tag appears as a subtag of either <feed> or <entry>, the tag can identify the type of item described in each feed entry. The value of the scheme attribute is http://schemas.google.com/g/2005#kind and the value of the term attribute indicates whether feed entries describe videos, playlist links, subscriptions, contacts or another entity type. When the <category> tag appears as a subtag of <entry> and the entry describes a video, the tag can also specify a particular keyword or category associated with that video. In this case, the scheme attribute indicates whether the term attribute value refers to a keyword or a category.
API_XML_Request: <media:group>	Yes	text	(child of: entry, children: media:title, media:description?, media:keywords?, media:category, media:content*, media:credit?, media:player?, media:rating?, media:restriction?, media:thumbnail*, yt:aspectratio?, yt:duration?, yt:private?, yt:uploaded?, yt:videoid?) Video metadata, uploaded on the first step. The <media:group> tag encapsulates metadata about a video resource. [106] Its XML children can encapsulate all the desired metadata information.
Step 3			

<i>Upload URL</i>	Yes	string	Target URL for video file upload, extracted on the second step and needed on the third one.
<i>Upload token</i>	Yes	string	Necessary for video file upload, extracted on the second step and needed on the third one. It is the user's identification.

API Method:	Upload video (Direct Uploading) [103][107]
Description:	<p>Uploading a video by sending an HTTP POST request containing the video and associated metadata to the method's URL.</p> <p>The string <code><youtube_username></code> must be replaced with a YouTube username or with the string "default", indicating an upload to the account of the currently logged user.</p> <p>The currently logged user is identified by the authentication token.</p> <p>The Upload API request has the following format:</p> <pre>POST /feeds/api/users/default/uploads HTTP/1.1 Host: uploads.gdata.youtube.com Authorization: AuthSub token="<authentication_token>" GData-Version: 2 X-GData-Key: key=<developer_key> Slug: <video_filename> Content-Type: multipart/related; boundary="<boundary_string>" Content-Length: <content_length> Connection: close --<boundary_string> Content-Type: application/atom+xml; charset=UTF-8 API_XML_request --<boundary_string> Content-Type: <video_content_type> Content-Transfer-Encoding: binary <Binary File Data> --<boundary_string>--</pre> <p>The Authorization header is formatted differently for a ClientLogin authentication token at a single point[108]. The following example demonstrates how to format the Authorization request header for a ClientLogin authentication token:</p> <pre>Authorization: GoogleLogin auth=<authentication_token></pre>

	Note: The hostname for uploading a video via the API, uploads.gdata.youtube.com , is different from the hostname used for all other API functions. However, the same authentication token can be used in all cases.		
URL:	<i>http://uploads.gdata.youtube.com/feeds/api/users/<youtube_username>/uploads</i>		
Formats:	<p>XML/Atom 1.0</p> <p>YouTube accepts a wide range of video file formats. The video files below are all accepted by YouTube. Some video formats work best and are most compatible with certain computers, software or devices.</p> <ul style="list-style-type: none"> • Windows Media Video (.WMV) • .3GP (cell phones) • .AVI (windows) • .MOV (mac) • .MP4 (ipod/psp) • .MPEG • .FLV (adobe flash) • .MKV (h.264) <p>After submitting an Upload API request, the API returns an Atom entry that contains information about the uploaded video. The entry has the same format as an entry that would appear in a user's uploaded videos feed. The response may contain properly escaped HTML tags. The entry will contain one <link> tag for which the value of the rel attribute is self. To check the status of the uploaded video, an HTTP GET request must be submitted to the URL identified in this <link> tag.</p>		
HTTP Method(s):	POST		
Requires Authentication:	Yes		
API restrictions:	No		
Parameters:	Required	Type	Description
youtube_username	Yes*	string	This value contains the username for the user's YouTube account. The uploaded video will be associated with this account. * This variable is only relevant for requests that use ClientLogin authentication.
authentication_token	Yes	string	This value contains the authentication token for programmer's request. The token will either be a ClientLogin token, an AuthSub single-use token, an AuthSub session token or an OAuth access token.
developer_key	Yes	string	This value uniquely identifies the application that is submitting the request to upload the video.
video_filename	Yes	string	This value contains the name of the video file that the content creator is uploading.
boundary_string	Yes	string	This value contains a string of non-space characters.

			Typically, the string is a series of hyphens followed by a random set of alphanumeric characters. Note: This value appears multiple times in the HTTP POST request.
content_length	Yes	bytes	This value contains the length of the entire body of the HTTP POST request.
API_XML_Request: Author	No	text	(child of: <i>entry</i> . Children: <i>name, uri</i> ?) Video metadata uploaded on the first step. The <author> tag usually contains information about the YouTube user who owns a piece of video content.
API_XML_Request: Category	No	text	(child of: <i>entry</i>) Video metadata, uploaded on the first step. The <category> tag specifies the category to which the entry belongs. <ul style="list-style-type: none"> When the <category> tag appears as a subtag of either <feed> or <entry>, the tag can identify the type of item described in each feed entry. The value of the <i>scheme</i> attribute is http://schemas.google.com/g/2005#kind and the value of the <i>term</i> attribute indicates whether feed entries describe videos, playlist links, subscriptions, contacts or another entity type. When the <category> tag appears as a subtag of <entry> and the entry describes a video, the tag can also specify a particular keyword or category associated with that video. In this case, the <i>scheme</i> attribute indicates whether the <i>term</i> attribute value refers to a keyword or a category.
API_XML_Request: <media:group>	Yes	text	(child of: <i>entry</i> , children: <i>media:title, media:description?, media:keywords?, media:category, media:content*, media:credit?, media:player?, media:rating?, media:restriction?, media:thumbnail*, yt:aspectratio?, yt:duration?, yt:private?, yt:uploaded?, yt:videoid?</i>) Video metadata, uploaded on the first step. The <media:group> tag encapsulates metadata about a video resource. [106] Its XML children can encapsulate all the desired metadata information.
video_content_type	Yes	string	The MIME type of the uploaded video file. The MIME type can be a video media type, such as video/mpeg or video/mp4, or it can be application/octet-stream.
Binary File Data	Yes	binary data	This value contains the binary code for the video file that is being uploaded.

API Method:	Add Comment reply [113]		
Description:	<p>A comment is a text response to a video. Logged-in users can add comments to a video but cannot modify or delete those comments. In addition, please note that YouTube will convert any HTML mark-up that appears in a comment into plain text. Typically, a user would add a comment to a video after watching that video.</p> <p>The API request for adding a comment in response to a video is nearly identical to the API request for adding a comment in reply to another comment. The only difference is that the latter request contains an additional <link> tag that identifies the comment being replied to. The following instructions explain how to format the <link> tag:</p> <ol style="list-style-type: none"> 1. The rel attribute value must be set to <code>http://gdata.youtube.com/schemas/2007#in-reply-to</code>. 2. The type attribute value must be set to <code>application/atom+xml</code>. 3. The href attribute value must be set to the self-link for the comment being replied to. <p>The sample XML API request below demonstrates how to add a comment in</p>		
	<p>reply to another comment. The <link> tag that indicates that the comment is a reply to another comment is highlighted in bold and underlined text:</p> <pre>POST /feeds/api/videos/VIDEO_ID/comments HTTP/1.1 Host: gdata.youtube.com Content-Type: application/atom+xml Content-Length: CONTENT_LENGTH Authorization: AuthSub token="AUTHORIZATION_TOKEN" GData-Version: 2 X-GData-Key: key=DEVELOPER_KEY <?xml version="1.0" encoding="UTF-8"?> <entry xmlns="http://www.w3.org/2005/Atom" xmlns:yt="http://gdata.youtube.com/schemas/2007"> <link_rel="http://gdata.youtube.com/schemas/2007#in-reply-to" type="application/atom+xml" href="http://gdata.youtube.com/feeds/api/videos/VIDEO_ID/comments/COMMENT_ID"/> <content>This is a comment about another comment.</content> </entry></pre> <p>Note: The POST URL for this API request specifies the video associated with the new comment. The <link> tag identifies the original comment to which the new comment replies. The new comment and the original comment must be associated with the same video.</p>		
URL:	<code>http://gdata.youtube.com/feeds/api/videos/video_id/comments/comment_id</code>		
Formats:	XML/Atom 1.0		
HTTP Method(s):	POST		
Requires Authentication:	Yes		
API restrictions:	No		
Parameters:	Required	Type	Description
video_id	Yes	string	This is the id of the video where the target comment belongs to.
comment_id	Yes	string	The id of the target comment to which the reply is posted.
authentication_token	Yes	string	This value contains the authentication token for programmer's request. The token will either be a ClientLogin token, an AuthSub single-use token, an AuthSub session token or an OAuth access token.

API Method:	Standard Video Feeds ^[129]		
Description:	<p>Standard feeds contain lists of videos that either reflect YouTube user behavior, such as top-rated and most viewed video feeds, or were selected by YouTube staff, such as recently featured and mobile video feeds. Many of these feeds are shown on the Videos tab of the YouTube website. Standard feeds are updated every few minutes. To retrieve a standard feed, a HTTP GET request is sent to the URL associated with that feed.</p> <p>In addition, many of these feeds support the time query parameter, which allows restricting the feed to only contain relevant results from the previous day, week or month. For example, to retrieve top-rated videos from the previous day, the HTTP GET request looks like the following URL:</p> <pre>http://gdata.youtube.com/feeds/api/standardfeeds/top_rated?time=today</pre> <p>Note: The top_rated, top_favorites, most_viewed, most_discussed and most_responded standard feeds support the time parameter.</p>		
URL:	<i>http://gdata.youtube.com/feeds/api/standardfeed/{parameter}</i>		
Formats:	XML/Atom 1.0 YouTube returns an Atom feed that contains one <entry> tag for each video in the result set. ^[207]		
HTTP Method(s):	GET		
Requires Authentication:	No		
API restrictions:	No		
Parameters:	Required	Type	Description
developer_key	Yes	string	This value uniquely identifies the application that is submitting the request.
top_rated	No	URL	This feed contains the most highly rated YouTube videos.
top_favorites	No	URL	This feed contains videos most frequently flagged as favourite videos.
most_viewed	No	URL	This feed contains the most frequently watched YouTube videos.
most_popular	No	URL	This feed contains the most popular YouTube videos, selected using an algorithm that combines many different signals to determine overall popularity.
most_recent	No	URL	This feed contains the videos most recently submitted to YouTube.
most_discussed	No	URL	This feed contains the YouTube videos that have received the most comments.
most_responded	No	URL	This feed contains YouTube videos that receive the most video responses.
recently_featured	No	URL	This feed contains videos recently featured on the YouTube home page or featured videos tab.
watch_on_mobile	No	URL	This feed contains videos suitable for playback on mobile devices.

API Method:	Get Comments ^[212]		
Description:	<p>Each video entry contains a <gd:comments> tag, which encapsulates the URL to which API requests is sent to retrieve the list of comments from the video. The sample XML below shows how this URL appears in an API response:</p> <pre><feed> <entry> ... <media:group> ... </media:group> <gd:comments> <gd:feedLink href='http://gdata.youtube.com/feeds/api/videos/VIDEO_ID/comments' /> </gd:comments> </entry></pre>		
	<pre></feed></pre> <p>YouTube users may respond to a comment with another comment. In an XML response, a comment that was posted in response to another comment will contain an in-reply-to link as shown in the following <link> tag:</p> <pre><link rel="http://gdata.youtube.com/schemas/2007#in-reply-to" type="application/atom+xml" href="http://gdata.youtube.com/feeds/api/videos/2cd.../comments/978..." /></pre>		
URL:	<i>http://gdata.youtube.com/feeds/api/videos/video_id/comments</i>		
Formats:	XML/Atom 1.0		
HTTP Method(s):	GET		
Requires Authentication:	No		
API restrictions:	No		
Parameters:	Required	Type	Description
developer_key	Yes	string	This value uniquely identifies the application that is submitting the request.
video_id	Yes	string	Identifies the video for which the comments are being retrieved.

API Method:	Get Video Responses ^[213]		
Description:	<p>This section explains how to retrieve a feed of video responses for a video. Note that some videos may not have any video responses.</p> <p>Each video entry in an API response contains a series of <link> tags. The <link> tag that has a rel attribute value of http://gdata.youtube.com/schemas/2007/#video.responses identifies the URL for retrieving video responses for that video entry. (The <link> tag's href attribute identifies the URL.)</p> <pre><link rel="http://gdata.youtube.com/schemas/2007#video.responses" type="application/atom+xml" href="http://gdata.youtube.com/feeds/api/videos/ZTUVgYoEN_b/responses" /></pre>		
URL:	<i>http://gdata.youtube.com/feeds/api/videos/video_id/responses</i>		
Formats:	XML/Atom 1.0		
HTTP Method(s):	GET		
Requires	No		

Authentication:			
API restrictions:	No		
Parameters:	Required	Type	Description
developer_key	Yes	string	This value uniquely identifies the application that is submitting the request.
video_id	Yes	string	Identifies the video for which responses are being retrieved.

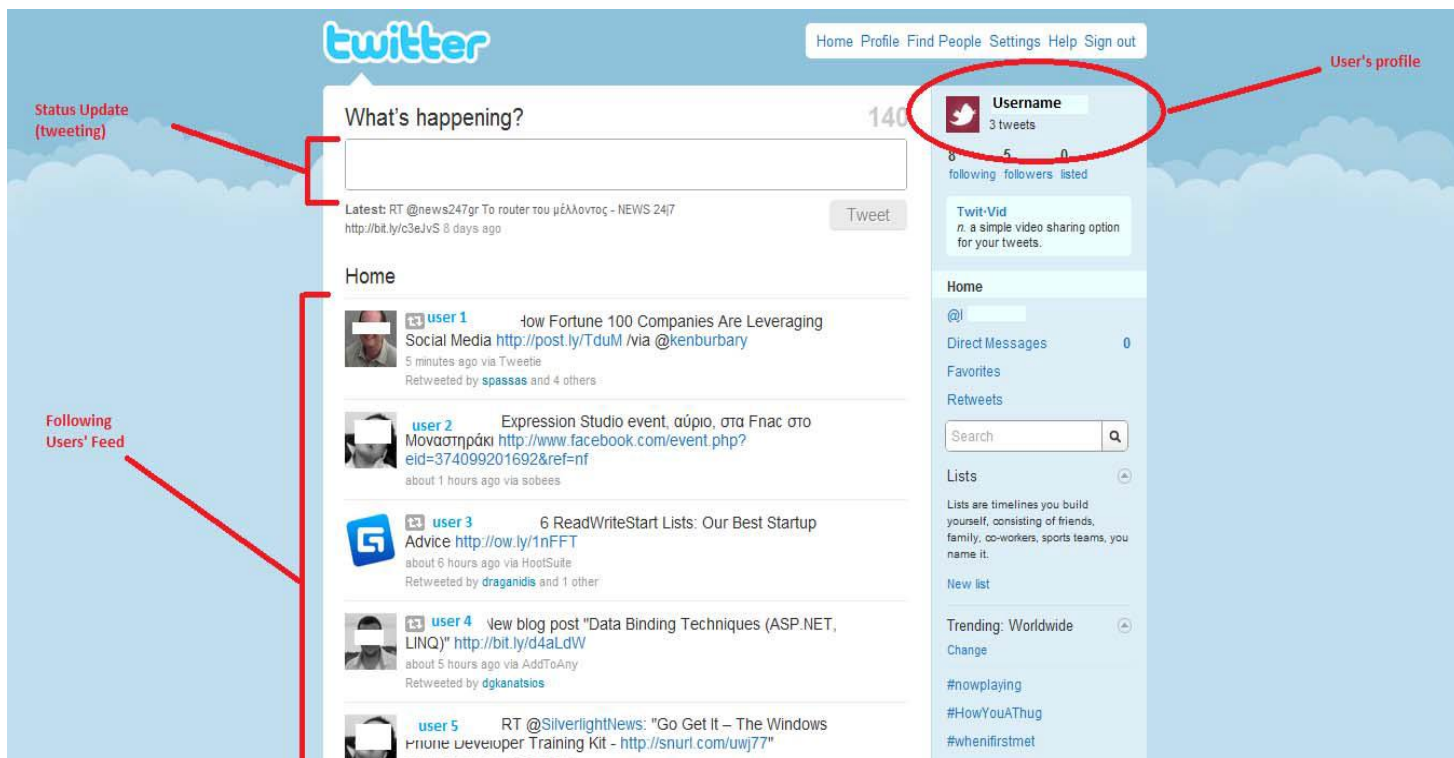
API Method:	Get user profile [220]		
Description:	<p>This section explains how to retrieve an entry containing information about a specific user's profile.</p> <p>To request another user's profile, a HTTP GET request must be sent to the following URL. Note that this request does not require authentication.</p> <pre>http://gdata.youtube.com/feeds/api/users/username</pre>		
URL:	http://gdata.youtube.com/feeds/api/users/username		
Formats:	<p>XML/Atom 1.0</p> <p>An API response to a request for a user profile contains a single <entry> tag. The profile entry contains the following information:</p> <ul style="list-style-type: none"> • Several tags in the yt namespace identify personal information that the user added to his YouTube profile, such as his age, favorite books, occupation or school. • The <media:thumbnail> tag contains a picture that the user has uploaded to his profile or, if no such picture exists, a default image. • The <yt:statistics> tag contains statistics about the user, including the number of people who have viewed the user's profile, the number of videos the user has watched and the number of people who subscribe to the user's uploaded videos. • A series of <gd:feedLink> tags provide URLs for retrieving the user's uploaded videos, favorite videos, playlists, subscriptions and contacts. 		
HTTP Method(s):	GET		
Requires Authentication:	No/ Yes(username has a default value)		
API restrictions:	No		
Parameters:	Required	Type	Description
developer_key	Yes	string	This value uniquely identifies the application that is submitting the request.
username	Yes	string	Identifies for whom the profile is being requested. If the request is made with "default" as the username, the API assumes that the request refers to the current user.

4.4 - Twitter

Το Twitter έχει ένα σχετικά απλό πρωτόκολλο επικοινωνίας. Πυρήνας της τεχνολογίας του Twitter είναι μια συσκευή αγνωστικιστής δρομολόγησης μηνυμάτων με στοιχειώδη κοινωνικά χαρακτηριστικά δικτύωσης. Με την αποδοχή μηνυμάτων sms, Web, Mobile Web, instant message, ή ακόμα και από εξωτερικά API, το Twitter καθιστά εύκολο για στους χρήστες να παραμένουν συνδεδεμένοι.

Επί του παρόντος, το Twitter API αποτελείται σήμερα από δύο διακριτά REST APIs και ενός Streaming API. Πλέον πολλοί προγραμματιστές εφαρμογών συνδυάζουν APIs ανάλογα με τις ιδιαιτερότητες της εφαρμογής τους. Ο διαχωρισμός μεταξύ των REST και Streaming APIs μπορεί να θεωρηθεί ως μια όχι ιδανική λύση και υπάρχει αποκλειστικά για λόγους backwards συμβατότητας. Υπάρχει συνεχής προσπάθεια από το Twitter για τη βελτίωση του API βαθμιαία συνδυάζοντας τα κομμάτια του Search και REST API. Το Streaming API παρέχει χαμηλού χρόνου αναμονής και υψηλού όγκου πρόσβαση στο Twitter.

Το API του Twitter είναι HTTP-based και, με εξαίρεση το Streaming API, επιχειρεί να συμμορφώνεται με τις αρχές του Representational State Transfer (REST). Υπάρχουν βιβλιοθήκες Twitter API για σχεδόν οποιοδήποτε γλώσσα προγραμματισμού.



4.2. - Twitter Interface

API Method:	statuses/update [34]		
Description:	Updates the authenticating user's status .Request must be a POST. A status update with text identical to the authenticating user's current status will be ignored to prevent duplicates. The user can submit posts (' tweet') and his 'followers' will receive and read them. A 'tweet' consists of a text post of 140 characters which can contain links to other websites or media.		
URL:	http://api.twitter.com/1/statuses/update.format		
Formats:	xml, json		
HTTP Method(s):	POST		
Requires Authentication:	Yes		
API restrictions:	Yes (Updates: 1,000 per day on top of other API restrictions)		
Parameters:	Required	Type	Description
status	Yes	Text	The text of the status update. URL encode as necessary. Statuses over 140 characters will cause a 403 error to be returned from the API.
in_reply_to_status_id	<i>optional</i>	int	The ID of an existing status that the update is in reply to. This parameter will be ignored unless the author of the tweet (that this parameter references) is mentioned within the status text. Therefore, @[username] must be included, where username is the author of the referenced tweet, within the update.
lat	<i>optional</i>	Decimal (from -90.0 to +90.0)	The location's latitude that this tweet refers to. The valid ranges for latitude is -90.0 to +90.0 (North is positive) inclusive. This parameter will be ignored if outside that range, if it is not a number, if geo_enabled is disabled, or if there not a corresponding long parameter with this tweet.
long	<i>optional</i>	Decimal (from -180.0 to +180.0)	The location's longitude that this tweet refers to. The valid ranges for longitude is -180.0 to +180.0 (East is positive) inclusive. This parameter will be ignored if outside that range, if it is not a number, if geo_enabled is disabled, or if there not a corresponding lat parameter with this tweet.

place_id	<i>optional</i>	int	The place to attach to this status update. Valid place_ids can be found by querying <code>geo/reverse_geocode</code> .
display_coordinates	<i>optional</i>	Boolean	By default, geo-tweets will have their coordinates exposed in the status object (to remain backwards compatible with existing API applications). To turn off the display of the precise latitude and longitude (but keep the contextual location information), pass <code>display_coordinates=false</code> on the status update.

API Method:	statuses/retweet [35]		
Description:	The user has also the ability to retransmit ('Retweet' - RT) a received 'tweet' thus extending the reach of a message. This API command performs this action. Requires the id parameter of the original tweet. The request must be a POST or PUT. Returns the original tweet with retweet details embedded.		
URL:	http://api.twitter.com/1/statuses/retweet/id.format		
Formats:	xml, json		
HTTP Method(s):	POST, PUT		
Requires Authentication:	Yes		
API restrictions:	Yes (Updates/Retweets: 1,000 per day on top of other API restrictions)		
Parameters:	Required	Type	Description
id	Yes	int	The numerical ID of the tweet you are retweeting.

API Method:	friendships create [45]		
Description:	Allows the authenticating users to follow the user specified in the ID parameter. Returns the "befriended" user in the requested format when successful. Returns a string describing the failure condition when unsuccessful. If the originating user is already a friend of the target user an HTTP 403 will be returned.		
URL:	http://api.twitter.com/1/friendships/create/id.format		
Formats:	xml, json		
HTTP Method(s):	POST		
Requires Authentication:	Yes		
API restrictions:	Yes- The technical follow limit is 1,000 per day. Also Once an account is following 2,000 other users, additional follow attempts are limited by account-specific ratios.		
Parameters:	Required	Type	Description
id	Yes*	String/Int	The ID or screen name of the user to befriend
user_id	Yes*	int	Specifies the ID of the user to befriend
screen_name	Yes*	string	Specifies the screen name of the user to befriend.
follow	<i>Optional</i>	Boolean	Enable delivery of statuses from this user to the authenticated user's device

API Method:	Search [51]		
Description:	Returns tweets that match a specified query.		
URL:	http://search.twitter.com/search.format		
Formats:	json, atom		
HTTP Method(s):	GET		
Requires Authentication:	No		
API restrictions:	1 call per request.		
Parameters:	Required	Type	Description
callback	Optional	String	Only available for JSON format. If supplied, the response will use the JSONP format with a callback of the given name.
lang	Optional	String	Restricts tweets to the given language, given by an ISO 639-1 code.
locale	Optional	String	Specify the language of the query
			you are sending (only ja is currently effective).
max_id	Optional	int	Returns tweets with status ids less than the given id.
q	Optional	String	The text to search for.
rpp	Optional	int	The number of tweets to return per page, up to a max of 100.
page	Optional	int	The page number (starting at 1) to return, up to a max of roughly 1500 results (based on rpp * page).
since	Optional	YYYY-MM-DD	Returns tweets with since the given date.
since_id:	Optional	int	Returns tweets with status ids greater than the given id.
geocode	Optional	latitude,longitude,radius(mi/km)	Returns tweets by users located within a given radius of the given latitude/longitude.
show_user	Optional	Boolean	When true, prepends "<user>:" to the beginning of the tweet. This is useful for readers that do not display the Atom author field.
until	Optional	YYYY-MM-DD	Returns tweets generated before the given date.

API Method:	statuses retweeted-to-me [61]		
Description:	Returns the 20 most recent retweets posted by the authenticated user's friends .		
URL:	http://api.twitter.com/1/statuses/retweeted_to_me.format		
Formats:	xml, json, atom		
HTTP Method(s):	GET		
Requires Authentication:	Yes		
API restrictions:	1 call per request.		
Parameters:	Required	Type	Description
since_id	Optional	int	Returns only statuses with an ID greater than (that is, more recent than) the specified ID.
max_id	Optional	int	Returns only statuses with an ID less than (that is, older than) or equal to the specified ID.
count	Optional	int	Specifies the number of statuses to retrieve. May not be greater than 200.
page	Optional	int	Specifies the page of results to retrieve (there are pagination limits).

API Method:	statuses retweets-of-me [62]		
Description:	Returns the 20 most recent retweets posted by the authenticating user's friends		
URL:	http://api.twitter.com/1/statuses/retweets_of_me.format		
Formats:	xml, json, atom		
HTTP Method(s):	GET		
Requires Authentication:	Yes		
API restrictions:	1 call per request.		
Parameters:	Required	Type	Description
since_id	Optional	int	Returns only statuses with an ID greater than (that is, more recent than) the specified ID.
max_id	Optional	int	Returns only statuses with an ID less than (that is, older than) or equal to the specified ID.
count	Optional	int	Specifies the number of statuses to retrieve. May not be greater than 200.
page	Optional	int	Specifies the page of results to retrieve (there are pagination limits).

API Method:	friends ids [82]		
Description:	Returns an array of numeric IDs for every user the specified user is following.		
URL:	http://api.twitter.com/1/friends/ids.format		
Formats:	xml, json		
HTTP Method(s):	GET		
Requires Authentication:	No, unless requesting it from a protected user; if getting this data of a protected user, the user must auth (and be allowed to see that user).		
API restrictions:	1 call per request		
Parameters:	Required	Type	Description
id	Yes*	String/Int	The id or screen name of the user for whom the friends list is returned.
user_id	Yes*	int	Specifies the ID of the user for whom the friends list is returned.
screen_name	Yes*	string	Specifies the screen name of the user for whom the friends list is returned.
cursor	Optional	int	Breaks the results into pages. A single page contains 5000 identifiers. A value of -1 begins paging. The cursor option is used to access older followers

* one of the fields is required.

API Method:	followers ids [83]		
Description:	Returns an array of numeric IDs for every user following the specified user.		
URL:	http://api.twitter.com/1/followers/ids.format		
Formats:	xml, json		
HTTP Method(s):	GET		
Requires Authentication:	No, unless requesting it from a protected user; if getting this data of a protected user, the user must auth (and be allowed to see that user).		
API restrictions:	1 call per request		
Parameters:	Required	Type	Description
id	Yes*	String/Int	The id or screen name of the user for whom the followers list is returned.
user_id	Yes*	int	Specifies the ID of the user for whom the followers list is returned.
screen_name	Yes*	string	Specifies the screen name of the user for
			whom the followers list is returned.
cursor	Optional	int	Breaks the results into pages. A single page contains 5000 identifiers. A value of -1 begins paging. The cursor option is used to access older followers

4.5. - Facebook

Οι προσφερόμενες από το API του Facebook δυνατότητες σχετίζονται άμεσα με τη δημιουργία και τη διαχείριση εφαρμογών. Η πρόσβαση στο API του Facebook είναι δυνατή μόνο μέσα από πιστοποιημένες «Facebook App» εφαρμογές. Ο προγραμματισμός στο Facebook είναι εφικτός με δύο τρόπους:

- Ως μια εφαρμογή που φιλοξενείται στο προφίλ κάποιου λογαριασμού Facebook. Σε αυτή την περίπτωση η εφαρμογή εγκαθίσταται στο προφίλ του χρήστη.
- Ως μια εφαρμογή έξω από το Facebook συνδεδεμένη με την πλατφόρμα του μέσω της διαδικασίας «Facebook connect». Με άλλα λόγια, μια εξωτερική εφαρμογή συνδέεται με το Facebook μέσα από το API του εκτελώντας API κλήσεις μαζί του.

Η βάση του API του Facebook χρησιμοποιεί ένα REST-like interface. Για αυτό το λόγο, οι κλήσεις στο Facebook πραγματοποιούνται στέλλοντας HTTP GET ή POST requests στον server του Facebook API REST (<http://api.facebook.com/restserver.php>). Σχεδόν οποιαδήποτε γλώσσα προγραμματισμού μπορεί να χρησιμοποιηθεί για να επικοινωνήσει με το πρότυπο HTTP με αυτόν το server. Το Facebook ακόμα παρέχει μια επιπρόσθετη βιβλιοθήκη JavaScript API και ένα πλήθος από επιπλέον βοηθητικές γλώσσες προγραμματισμού όπως: FQL (Facebook Query Language), XFBML, FBML (Facebook Markup Language; an extension to HTML) and FBJS (JavaScript into Facebook Applications).



4.3. - Facebook Interface

API Method:	Links.post [132]		
Description:	Lets a user post a link on their Wall through the application. Before the application can post a link to a user's profile, the user must grant the application the <code>share_item</code> extended permission.		
URL:	Not transparent –dependent on the library used(i.e. PHP5, JavaScript, iPhone, ActionScript 3.0, Microsoft SDK, Force.com are the official libraries)		
Formats:	XML/JSON (This method returns a <code>link_id</code> if successful.)		
HTTP Method(s):	POST		
Requires Authentication:	Yes. (This function does not require a <code>session_key</code> , unless it is being called by a desktop application. This method can be called using a session secret, and not the application secret)		
API rate limited:	No		
Parameters:	Required	Type	Description
<code>api_key</code>	Yes	string	The application key associated with the calling application. If the API key is specified in the client, it doesn't have to be passed with every call.
<code>session_key</code>	Optional	string	The session key of the logged user or the session key provided when the user granted the application the <code>offline_access</code> extended permission. The session key is automatically included by the PHP client.
<code>Call_id</code>	Yes	float	The sequence number of the request. Each successive call for a session must use a sequence number greater than the last. Facebook suggests using the current time in milliseconds, such as PHP's <code>microtime(true)</code> function. If the call ID is specified in the client, it doesn't have to be passed with every call.
<code>sig</code>	Yes	string	An MD5 hash of the current request and the application secret key. Facebook computes the signature automatically.
<code>v</code>	Yes	string	This must be set to 1.0 to use this version of the API. If the version is specified in the client it doesn't have to be passed with every call.
<code>url</code>	Yes	string	The URL for the link.
<code>comment</code>	Yes	string	The comment that the user included with the link.
<code>uid</code>	Yes	int	The user ID of the user posting the link.
<code>image</code>	Optional	string	The absolute URL to an image on the linked page. It can be determined using <code>links.preview</code> . If the image URL does not match one of the images on the page, or if this parameter is not specified, the first image specified in <code>metadata</code> is displayed with the post.
<code>format</code>	Optional	string	The desired response format, which can be either XML or JSON. (Default value is XML.)
<code>callback</code>	Optional	string	Name of a function to call. This enables cross-domain JavaScript requests using the <code><script></code> tag, (also known as JSONP), and works with both the XML and JSON formats. The function will be called with the response passed as the parameter.

API Method:	Notes.create [134]		
Description:	Lets a user write a Facebook note through an application. Before a user can write a note through the application, the user must grant it the "create_note" extended permission.		
URL:	Not transparent –dependent on the library used (PHP5, JavaScript, iPhone, ActionScript 3.0, Microsoft SDK, Force.com are the official libraries)		
Formats:	XML/JSON (If successful, this method returns the "note_id" of the note that was just created)		
HTTP Method(s):	POST		
Requires Authentication:	Yes. (This function does not require a session_key, unless it is being called by a desktop application.)		
API rate limited:	No		
Parameters:	Required	Type	Description
api_key	Yes	string	The application key associated with the calling application. If the API key is specified in the client, it is not necessary to be passed with every call.
Call_id	Yes	float	The sequence number of the request. Each successive call for any session must use a sequence number greater than the last. Facebook suggests using the current time in milliseconds, such as PHP's "microtime (true)" function. If call ID is specified in the client, it doesn't have to be passed with every call.
sig	Yes	string	An MD5 hash of the current request and the application's secret key. Facebook computes the signature automatically.
v	Yes	string	This must be set to 1.0 to use this version of the API. If the version is specified in the client, it doesn't have to be passed with every call.
uid	Yes	int	The user ID of the user posting the link.
title	Yes	string	The title of the note.
content	Yes	string	The content of the note
session_key	Optional	string	The session key of the logged user, or the session key provided when the user granted the "offline_access" extended permission to the application. The session key is automatically included by the PHP client.
format	Optional	string	The desired response format, which can be either XML or JSON. (Default value is XML.)
callback	Optional	string	Name of a function to call. This enables cross-domain JavaScript requests using the <script> tag (also known as JSONP) and works with both the XML and JSON formats. The function will be called with the response passed as the parameter.

API Method:	Status.set [137]		
Description:	<p>Updates a user's Facebook status through an application. Before the application can set a user's status, the user must grant it the "status_update" extended permission. This call is a streamlined version of users.setStatus, as it takes fewer arguments.</p> <p>For Web applications, the call must pass either the ID of the user on whose behalf it is being made or the session key for that user. If a user is not specified with the uid parameter, then the user associated with the relevant session will be the target of the call. To set the status of a Facebook Page, the call must pass its uid.</p> <p>However, if the application is a desktop application, a valid session key is needed for security reasons. A uid parameter is not necessary.</p>		
URL:	Not transparent –dependent on the used library (i.e. PHP5, JavaScript, iPhone, ActionScript 3.0, Microsoft SDK, Force.com are the official libraries)		
Formats:	XML/JSON (1 on success, 0 on failure, or an error code)		
HTTP Method(s):	POST		
Requires Authentication:	Yes. (This method can be called using a session secret, and not the application secret (for example, for a Facebook Connect site or desktop application).		
API rate limited:	No		
Parameters:	Required	Type	Description
api_key	Yes	string	The application key associated with the calling application. If the API key is specified in the client, it doesn't have to be passed with every call.
Call_id	Yes	float	The sequence number of the request. Each successive call for any session must use a sequence number greater
			than the last. Facebook suggests using the current time in milliseconds, such as PHP's "microtime (true)" function. If call ID is specified in the client, it doesn't have to be passed with every call.
sig	Yes	string	An MD5 hash of the current request and the application's secret key. Facebook computes the signature automatically.
v	Yes	string	This must be set to 1.0 to use this version of the API. If the version is specified in the client, it doesn't have to be passed with every call.
session_key	Optional	string	The session key of the user whose status is being set. Note: A session key is always required for desktop applications. It is required for Web applications only when the uid is not specified.
format	Optional	string	The desired response format, which can be either XML or JSON. (Default value is XML.)
callback	Optional	string	Name of a function to call. This enables cross-domain JavaScript requests using the <script> tag, (also known as JSONP), and works with both the XML and JSON formats. The function will be called with the response passed as the parameter.

status	<i>Optional</i>	<i>string</i>	The status message to set. Note: The maximum message length is 255 characters; messages longer than that limit will be truncated and appended with "...".
uid	<i>Optional</i>	<i>int</i>	The user ID of the user posting the link. If a user is not specified, the current session user's ID is used.

API Method:	Video.upload[145]
Description:	<p>Uploads a video owned by the current session user and returns the new video. In order to upload a video from an application, the user must grant this application the "video_upload" extended permission.</p> <p>Important: Upload requests must be formed as a MIME multi-part message sent using POST data. Each argument, including the raw video data, should be specified as a separate chunk of form data.</p> <p>The details of preparing a MIME multi-part message depend on your particular request library, but typically the library allows you to specify POST, set various MIME headers, then add additional data. Note that all line breaks in the message should be a carriage return followed by a newline ('\r\n') rather than just a newline ('\n'). For example, after specifying each Content-Disposition line, '\r\n\r\n' should be printed.</p> <p>Each user is subject to limits on the length and size of the video files they can upload, just like they are when uploading through Facebook.</p> <p>video.getUploadLimits can be used to determine a specific user's limits. To increase video upload limits, users can verify their accounts with Facebook. One simple way to verify an account is by confirming a mobile phone number with Facebook. It can be determined whether a user has verified their account querying for verified on the user FQL table or by calling users.isVerified, but in general it should be determined a user's video limits independently with video.getUploadLimits. It might be desirable to display these limits to the user before that user tries to upload a video to Facebook.</p>
URL:	Not transparent –dependent on the library used (PHP5, JavaScript, iPhone, ActionScript 3.0, Microsoft SDK, Force.com are the official libraries)
Formats:	<p>XML/JSON , MIME</p> <p>The following video file formats can be uploaded through this call:</p> <ul style="list-style-type: none"> ▪ 3g2 (Mobile Video) ▪ 3gp (Mobile Video) ▪ 3gpp (Mobile Video)

	<ul style="list-style-type: none"> ▪ asf (Windows Media Video) ▪ avi (AVI Video) ▪ dat (MPEG Video) ▪ flv (Flash Video) ▪ m4v (MPEG-4 Video) ▪ mkv (Matroska Format) ▪ mod (MOD Video) ▪ mov (QuickTime Movie) ▪ mp4 (MPEG-4 Video) ▪ mpe (MPEG Video) ▪ mpeg (MPEG Video) ▪ mpeg4 (MPEG-4 Video) ▪ mpg (MPEG Video) ▪ nsv (Nullsoft Video) ▪ ogm (Ogg Format) ▪ ogv (Ogg Video Format) ▪ qt (QuickTime Movie) ▪ tod (TOD Video) ▪ vob (DVD Video) ▪ wmv (Windows Media Video) 		
HTTP Method(s):	POST		
Requires Authentication:	Yes. (This method can be called using a session secret, and not the application secret (for example, for a Facebook Connect site or desktop application).)		
API rate limited:	No		
Parameters:	Required	Type	Description
api_key	Yes	string	The application key associated with the calling application. If the API key is specified in application's client, it doesn't have to be passed with every call.
session_key	Yes	string	The session key of the logged user or the session key provided when the user granted the application the "offline_access" extended permission. The

			session key is automatically included by Facebook's PHP client.
call_id	Yes	float	The request's sequence number. Each successive call for any session must use a sequence number greater than the last. Facebook suggests using the current time in milliseconds, such as PHP's "microtime (true)" function. If the call ID is specified in application's client, it's not needed to be passed with every call.
sig	Yes	string	An MD5 hash of the current request and the application's secret key. Facebook computes the signature automatically.
v	Yes	string	This must be set to 1.0 to use this version of the API. If the version is specified in the client, it doesn't have to be passed with every call.
format	Optional	string	The desired response format, which can be either XML or JSON. (Default value is XML.)
callback	Optional	string	Name of a function to call. This enables cross-domain JavaScript requests using the <script> tag, sometimes known as "JSONP". This works with both XML and JSON.
[no name]	Yes	data	The file data for the video.
title	Optional	string	The name of the video. The name cannot be longer than 65 characters. Longer titles will get truncated and will not return an error.
description	Optional	string	A description of the video. There is no limit to the length of the description.
privacy	Optional	object	A JSON-encoded object that defines the privacy setting for a post, video, or album. It contains the following fields: <ul style="list-style-type: none"> ▪ value (string): The privacy value for the object, specify one of EVERYONE, CUSTOM, ALL_FRIENDS, NETWORKS_FRIENDS, FRIENDS_OF_FRIENDS, SELF. ▪ friends (string): For CUSTOM settings, this indicates which users can see the object. Can be one of EVERYONE, NETWORKS_FRIENDS (when the object can be seen by networks and friends), FRIENDS_OF_FRIENDS, ALL_FRIENDS,

			<p>SOME_FRIENDS, SELF, or NO_FRIENDS (when the object can be seen by a network only).</p> <ul style="list-style-type: none"> ▪ networks (string): For CUSTOM settings a comma-separated list of network IDs that can see the object or 1 for all of a user's networks needs to be specified. ▪ allow (string): When friends is set to SOME_FRIENDS a comma-separated list of user IDs and friend list IDs that <i>can</i> see the post needs to be specified. ▪ deny (string): When friends is set to SOME_FRIENDS, a comma-separated list of user IDs and friend list IDs that <i>cannot</i> see the post needs to be specified. <p>Only the user can specify the privacy settings for the post. An interface that lets the user specify the privacy setting can be created. For CUSTOM settings, friends.get and friends.getLists should be used to get the user's friends and friend lists to populate the interface, and the selections passed along to the privacy object.</p> <p>Privacy Policy: Any non-default privacy setting must be intentionally chosen by the user. It may not set a custom privacy setting unless the user has proactively specified that they want this non-default setting.</p>
--	--	--	--

API Method:	Photos.createAlbum[158]		
Description:	<p>Creates and returns a new album owned by the specified user or the current session user. See photo uploads [159] for a description of the upload workflow. The only storable values returned from this call are aid and owner. No relationships between them are storable.</p> <p>For Web applications, either the ID of the user (on whose behalf this call is being made) or the session key for that user must be passed (but not both). If a user is specified with the uid parameter, then the session user will be the target of the call.</p> <p>However, if the application is a desktop application, a valid session key must be passed for security reasons. I must not be passed a uid parameter.</p>		
URL:	Not transparent –dependent on the library used (i.e. PHP5, JavaScript, iPhone, ActionScript 3.0, Microsoft SDK, Force.com are the official libraries)		
Formats:	XML/ JSON (The returned cover_pid is always 0)		
HTTP Method(s):	POST		
Requires Authentication:	Yes (This method can be called by using a session secret, and not the application secret (for example, for a Facebook Connect site or desktop application))		
API rate limited:	No		
Parameters:	Required	Type	Description
api_key	Yes	string	The application key associated with the calling application. If the API key is specified in the client, it doesn't have to be passed with every call.

session_key	<i>Optional</i>	string	The session key of the logged user or the session key provided when the user granted the application the <code>offline_access</code> extended permission. The session key is automatically included by Facebook's PHP client. Note: A session key is always required for desktop applications. It is required for Web applications only when the <code>uid</code> is not specified.
call_id	Yes	float	The request's sequence number. Each successive call for any session must use a sequence number greater than the last. Facebook suggests using the current time in milliseconds, such as PHP's "microtime (true)" function. If the call ID is specified in application's client, it doesn't have to be passed with every call.
sig	Yes	string	An MD5 hash of the current request and the application's secret key. Facebook computes the signature automatically.
v	Yes	string	This must be set to 1.0 to use this version of the API. If the version is specified in application's client, it doesn't have to be passed with every call.
format	<i>Optional</i>	string	The desired response format, which can be either XML or JSON. (Default value is XML.)
callback	<i>Optional</i>	string	Name of a function to call. This enables cross-domain JavaScript requests using the <code><script></code> tag, sometimes known as "JSONP". This works with both XML and JSON.
name	Yes	string	The album name.
location	<i>optional</i>	string	The album location.
description	<i>optional</i>	string	The album description.
visible	<i>optional</i>	string	Visibility of the album. One of <code>friends</code> , <code>friends-of-friends</code> , <code>networks</code> , <code>everyone</code> . Note: This parameter remains for legacy reasons. The <code>privacy</code> parameter should be used instead, as it is more powerful and dynamic.
uid	<i>optional</i>	int	The user ID of the user for whom the album is being created. If this parameter is not specified, then it defaults to the session user. Note: This parameter applies only to Web applications and is required by them only if the <code>session_key</code> is not specified. Facebook ignores this parameter if it is passed by a desktop application.
privacy	<i>optional</i>	object	A JSON-encoded object that defines the privacy setting for a post, video, or album. It contains the following fields. <ul style="list-style-type: none"> value (string): The privacy value for the object, specify one of <code>EVERYONE</code>, <code>CUSTOM</code>, <code>ALL_FRIENDS</code>, <code>NETWORKS_FRIENDS</code>, <code>FRIENDS_OF_FRIENDS</code>, <code>SELF</code>. friends (string): For <code>CUSTOM</code> settings, this indicates

			<p>which users can see the object. Can be one of EVERYONE, NETWORKS_FRIENDS (when the object can be seen by networks and friends), FRIENDS_OF_FRIENDS, ALL_FRIENDS, SOME_FRIENDS, SELF, or NO_FRIENDS (when the object can be seen by a network only).</p> <ul style="list-style-type: none"> ▪ networks (string): For CUSTOM settings, specify a comma-separated list of network IDs that can see the object, or 1 for all of a user's networks. ▪ allow (string): When friends is set to SOME_FRIENDS, specify a comma-separated list of user IDs and friend list IDs that <i>can</i> see the post. ▪ deny (string): When friends is set to SOME_FRIENDS, specify a comma-separated list of user IDs and friend list IDs that <i>cannot</i> see the post. <p>Only the user can specify the privacy settings for the post. You can create an interface that lets the user specify the privacy setting. For CUSTOM settings, it should be used friends.get and friends.getLists to get the user's friends and friend lists to populate the interface, then pass along the selections to the privacy object.</p> <p>Privacy Policy: Any non-default privacy setting must be intentionally chosen by the user. It may not be set a custom privacy setting unless the user has proactively specified that they want this non-default setting.</p>
--	--	--	--

API Method:	Photos.upload[160]
Description:	<p>Uploads a photo owned by the specified user or the current session user and returns the new photo. See photo uploads for a description of the upload workflow. The only storable values returned from this call are pid, aid, and owner. All applications can upload photos with a "pending" state, which means that the photos must be approved by the user before they are visible on the site. Photos uploaded by applications with the photo_upload extended permission are visible immediately.</p> <p>If no album is specified, the photo is uploaded to a default album for the application, which can be created if necessary. Regular albums have a size limit of 200 photos. Default application albums have a size limit of 1000 photos.</p> <p>The application can upload to a specific user's album by calling photos.getAlbums.</p>

	<p>However, it cannot upload to a user's profile picture album.</p> <p>Important: Upload requests must be formed as a MIME multi-part message sent using POST data. Each argument, including the raw image data, should be specified as a separate chunk of form data.</p> <p>The details of preparing a MIME multi-part message depend on the particular request library, but typically the library allows to specify POST, set various MIME headers, then add additional data. Note that all line breaks in the message should be a carriage return followed by a newline ("\r\n") rather than just a newline ("\n"). For example, after specifying each Content-Disposition line, it should print "\r\n\r\n".</p> <p>It is strongly recommended to scale the image in the application before adding it to the request. The largest dimension should be at most 720 pixels (the largest display size Facebook supports).</p> <p>For Web applications, it must be passed either the ID of the user on whose behalf it is being made this call or the session key for that user, but not both. If a user is not specified with the uid parameter, then that user whose session it is will be the target of the call.</p> <p>However, if the application is a desktop application, a valid session key must be passed for security reasons. Do not pass a uid parameter.</p>
URL:	Not transparent –dependent on the library used (i.e. PHP5, JavaScript, iPhone, ActionScript 3.0, Microsoft SDK, Force.com are the official libraries)
Formats:	<p>XML/ JSON , MIME</p> <p>The following image file formats can be uploaded through this call:</p> <ul style="list-style-type: none"> ▪ GIF ▪ JPG ▪ PNG ▪ PSD ▪ TIFF ▪ JP2 ▪ IFF ▪ WBMP ▪ XBM
HTTP Method(s):	POST
Requires Authentication:	Yes (This method does not require a session_key, unless it is being called by a desktop application. This method can be called by using a session secret, and not

	the application secret (for example, for a Facebook Connect site or desktop application).		
API rate limited:	No		
Parameters:	Required	Type	Description
api_key	Yes	string	The application key associated with the calling application. If the API key is specified in the client, it doesn't have to be passed with every call.
call_id	Yes	float	The request's sequence number. Each successive call for any session must use a sequence number greater than the last. Facebook suggests using the current time in milliseconds, such as PHP's "microtime (true)" function. If the call ID is specified in application's client, it doesn't have to be passed with every call.
sig	Yes	string	An MD5 hash of the current request and the application's secret key. Facebook computes the signature automatically.
v	Yes	string	This must be set to 1.0 to use this version of the API. If the version is specified in application's client, it doesn't have to be passed with every call.
[no name]	Yes	data	The raw image data for the photo.
session_key	Optional	string	The session key of the logged user, or the session key provided when the user granted the application the offline_access extended permission. The session key is automatically included by Facebook's PHP client. Note: A session key is always required for desktop applications. It is required for Web applications only when the uid is not specified.
format	Optional	string	The desired response format, which can be either XML or JSON. (Default value is XML.)
callback	Optional	string	Name of a function to call. This enables cross-domain JavaScript requests using the <script> tag, sometimes known as "JSONP". This works with both XML and JSON.
aid	Optional	string	The album ID of the destination album. The aid cannot be longer than 50 characters.
caption	Optional	string	The caption of the photo.
uid	optional	int	The user ID of the user for whom the album is being created. If this parameter is not specified, then it defaults to
			the session user. Note: This parameter applies only to Web applications and is required by them only if the session_key is not specified. Facebook ignores this parameter if it is passed by a desktop application.

API Method:	Comments.get[167]		
Description:	<p>This method returns all the comments, for a given XID, posted through either fb:comments or the Comments Box (which is created with the fb:comments (XFBML) tag). This method is a wrapper for the FQL query on the "comment" FQL table.</p> <p>Only one XID can be specified with this call. In order to retrieve comments for multiple XIDs, fql.query should be run against the comment FQL table.</p>		
URL:	Not transparent –dependent on the library used(i.e. PHP5, JavaScript, iPhone, ActionScript 3.0, Microsoft SDK, Force.com are the official libraries)		
Formats:	XML/ JSON		
HTTP Method(s):	GET		
Requires Authentication:	Yes (session_key)		
API rate limited:	No		
Parameters:	Required	Type	Description
api_key	Yes	string	The application key associated with the calling application. If the API key is specified in the client, it doesn't have to be passed with every call.
call_id	Yes	float	The sequence number of the request. Each successive call for any session must use a sequence number greater than the last. Facebook suggests using the current time in milliseconds, such as PHP's "microtime (true)" function. If the call ID is specified at the client, it doesn't have to be passed with every call.
sig	Yes	string	An MD5 hash of the current request and the secret key of the application, as described in the "How Facebook Authenticates The application" documentation. Facebook computes the signature automatically.
v	Yes	string	This parameter must be set to 1.0 to use the current version of the API. If the version is specified at the client of the application, it doesn't have to be passed with every call.
session_key	Yes	string	The session key of the logged user, or the session key provided when the user granted the application the "offline_access" extended permission. The session key is automatically included by the PHP client of Facebook.
xid	Yes*	int	The comment xid to retrieve. For a Comments Box, the xid can be determined on the admin panel or in the application settings editor in the Facebook Developer application. * Either a xid or an object_id parameter must be specified.
object_id	Yes*	string	The object_id of an object on Facebook. This can be a video, note, link, photo, or photo album. Note that for photos and albums, the object_id must be queried from the "photo" and "album" FQL tables. * Either an xid or an object_id parameter must be specified

format	<i>Optional</i>	<i>string</i>	The desired response format, which can be either XML or JSON. (Default value is XML.)
callback	<i>Optional</i>	<i>string</i>	The name of a function to call. This enables cross-domain JavaScript requests using the <script> tag (also known as JSONP) and works both with the XML and JSON formats. This function will be called with the response passed as the parameter.

method	Users.getInfo [190]
Description:	<p>Returns a wide array of user-specific information for each user identifier passed, limited by the view of the current user. The current user is determined from the <code>session_key</code> parameter. The only storable values returned from this call are those under the <code>affiliations</code> element, the <code>notes_count</code> value, the <code>proxied_email</code> address, and the contents of the <code>profile_update_time</code> element.</p> <p>Use this call to get user data that intend to display to other users (of the application, for example). If it is needed some basic information about a user for analytics purposes, call <code>users.getStandardInfo</code> instead.</p> <p>This call no longer requires a session key. However, if this method is called without a session key, it only returns the following information:</p> <ul style="list-style-type: none"> ▪ <code>uid</code> ▪ <code>first_name</code> ▪ <code>last_name</code> ▪ <code>name</code> ▪ <code>locale</code> ▪ <code>current_location</code> ▪ <code>affiliations</code> (regional type only) ▪ <code>pic_square</code> ▪ <code>profile_url</code> ▪ <code>sex</code> <p>This method can be called as soon as a user interacts with the application, before she has authorized the application to access his information. If so, it is possible to get the same information as without a session (see above).</p>

	<p>Important: Depending upon the user's privacy settings (including whether the user has decided to opt out of Platform completely), there is access to certain user data. For any user submitted to this method, the following user fields are visible to an application only if that user has authorized that application:</p> <ul style="list-style-type: none"> ▪ meeting_for ▪ meeting_sex ▪ religion ▪ significant_other_id <p>In addition, the visibility of all fields, with the exception of first_name, last_name, name, pic, sex, current_location, affiliations, and uid may be restricted by the user's Facebook privacy settings in relation to the calling user (the user associated with the current session).</p> <p>If a field is not visible for either of the above reasons, then that field's corresponding element will be empty with a nil attribute set, in the following manner: <significant_other_id xsi:nil="true"/></p> <p>Finally, a user profile query will not return any data if the user has turned off access to Facebook Platform.</p> <p>FQL equivalent: This function is similar (but returned in a slightly different format) to doing the following FQL query, with the appropriate parameters filled in:</p> <pre>SELECT uid, current_location FROM user WHERE <uid> IN (uid)</pre>
URL:	Not transparent –dependent on the library used(i.e. PHP5, JavaScript, iPhone, ActionScript 3.0, Microsoft SDK, Force.com are the official libraries)
Formats:	<p>XML/ JSON</p> <p>(Following is a list of data returned by this call:</p> <ul style="list-style-type: none"> ▪ uid - The user ID corresponding to the user info returned. This is always returned, whether included in fields or not, and always as the first subelement. ▪ about_me - text element corresponding to Facebook 'About Me' profile section. May be blank. ▪ activities - User-entered "Activities" profile field. No guaranteed formatting. ▪ affiliations - list of network affiliations, as affiliation elements, each of which contain year, type, status, name, and nid child elements. If no affiliations are returned, this element will be blank. The user's primary network (key: nid) will be listed first. <p>- type takes the following values:</p>

	<ul style="list-style-type: none"> • college: college network • high school: high school network • work: work network • region: geographical network <ul style="list-style-type: none"> - year may be blank, depending on the network type. - name is the name of the network. - nid is a unique identifier for the network. The user-to-network relation may be stored. - status describes the user's graduate status if the network is a college network. Otherwise, it is blank. <ul style="list-style-type: none"> ▪ birthday - User-entered "Birthday" profile field. No guaranteed formatting as it's based on the user's locale. The "birthday_date" must be used, needed to be sure of the format. ▪ birthday_date - The user's birthday, rendered as a machine-readable string. The format of this date never changes. ▪ books - User-entered "Favorite Books" profile field. No guaranteed formatting. ▪ contact_email - is a string containing the user's primary Facebook email address. If the user shared his or her primary email address with the application, this address also appears in the email field (see below). ▪ current_location - User-entered "Current Location" profile fields. Contains four children: city, state, country, and zip. <ul style="list-style-type: none"> - city is user-entered, and may be blank. - state is a well-defined name of the state, and may be blank. - country is well-defined, and may be blank. - zip is an integer, and is 0 if unspecified by the user. ▪ education_history - list of school information, as education_info elements, each of which contain name, year, and concentration child elements. If no school information is returned, this element will be blank. <ul style="list-style-type: none"> - year is a four-digit year, and may be blank. - name is the name of the school, and is user-specified. - concentrations is a list of concentration elements, and may be an empty list. - degree is the name of the degree, and may be blank. ▪ email is a string containing the user's primary Facebook email address or the user's proxied email address, whichever address the user granted the application. ▪ email_hashes is an array containing a set of confirmed email hashes for the user. Emails are registered via the connect.registerUsers API call and are only
--	---

	<p>confirmed when the user adds the application. The format of each email hash is the crc32 and md5 hashes of the email address combined with an underscore (_).</p> <ul style="list-style-type: none"> ▪ family is an array containing the user's family members, listing the user ID and relationship for each member. ▪ first_name is generated from the user-entered "Name" profile field. ▪ has_added_app - [Deprecated] Bool (0 or 1) indicating whether the user has authorized the application. This value is now equivalent to <code>is_app_user</code>. ▪ hometown_location - User-entered "Hometown" profile fields. Contains three children: city, state, and country. <ul style="list-style-type: none"> - city is user-entered, and may be blank - state is a well-defined name of the state, and may be blank. - country is well-defined, and may be blank. ▪ hs_info - User-entered high school information. Contains five children: <code>hs1_name</code>, <code>hs2_name</code>, <code>grad_year</code>, <code>hs1_key</code>, and <code>hs2_key</code>. <ul style="list-style-type: none"> - hs1_name is well-defined, and may be left blank - hs2_name is well-defined, and may be left blank, though may not have information if <code>hs1_name</code> is blank. - grad_year is a four-digit year, or may be blank - hs1_id is a unique ID representing that school, and is not zero if and only if <code>hs1_name</code> is not blank. - hs2_id is a unique ID representing that school, and is not zero if and only if <code>hs2_name</code> is not blank. ▪ interests - User-entered "Interests" profile field. No guaranteed formatting. ▪ is_app_user - Bool (0 or 1) indicating whether the user has used the calling application. ▪ is_blocked - Bool that returns true if the user is blocked to the viewer/logged user. The user must be logged in or the application must have a valid session key to get this value. ▪ last_name - The user's last name, which is generated from the user-entered "Name" profile field. ▪ locale - The current locale in which the user has chosen to browse Facebook. The basic format is LL_CC, where LL is a two-letter language code, and CC is a two-letter country code. Country codes are taken from the ISO 3166 alpha 2 code list. For instance, 'en_US' represents US English. ▪ meeting_for - list of desired relationship types corresponding to the "Looking For" profile element. If no relationship types are specified, the <code>meeting_for</code> element is empty. Otherwise represented as a list of seeking child text elements, which may each contain one of the following strings: Friendship, A
--	---

	<p>Relationship, Dating, Random Play, Whatever I can get.</p> <ul style="list-style-type: none"> ▪ meeting_sex - list of desired relationship genders corresponding to the "Interested In" profile element. If no relationship genders are specified, the meeting_sex element is empty. Otherwise represented as a list of sex child text elements, which may each contain one of the following strings: male, female. ▪ movies - User-entered "Favorite Movies" profile field. No guaranteed formatting. ▪ music - User-entered "Favorite Music" profile field. No guaranteed formatting. ▪ name - User-entered "Name" profile field. May not be blank. ▪ notes_count - Total number of notes written by the user. ▪ pic - URL of user profile picture, with max width 100px and max height 300px. May be blank. ▪ pic_with_logo - URL of user profile picture with a Facebook logo overlaid, with max width 100px and max height 300px. May be blank. ▪ pic_big - URL of user profile picture, with max width 200px and max height 600px. May be blank. ▪ pic_big_with_logo - URL of user profile picture with a Facebook logo overlaid, with max width 200px and max height 600px. May be blank. ▪ pic_small - URL of user profile picture, with max width 50px and max height 150px. May be blank. ▪ pic_small_with_logo - URL of user profile picture with a Facebook logo overlaid, with max width 50px and max height 150px. May be blank. ▪ pic_square - URL of a square section of the user profile picture, with width 50px and height 50px. May be blank. ▪ pic_square_with_logo - URL of a square section of the user profile picture with a Facebook logo overlaid, with width 50px and height 50px. May be blank. ▪ political - User-entered "Political View" profile field. It's a free-form text field. ▪ profile_blurb - A free-form text field under a user's profile picture. ▪ profile_update_time - Time (in seconds since epoch) that the user's profile was last updated. If the user's profile was not updated within the past 3 days, 0 is returned. ▪ profile_url - URL of the Facebook profile of the user. If the user has specified a username, the username is included in the URL, not profile.php?id=UID. ▪ proxied_email - A proxied wrapper alternative for contacting the user through email, instead of directly calling notifications.sendEmail. If the user shared his or her proxied email address with the application, this address also
--	---

	<p>appears in the email field (see below).</p> <ul style="list-style-type: none"> ▪ quotes - User-entered "Favorite Quotes" profile field. No guaranteed formatting. ▪ relationship_status - User-entered "Relationship Status" profile field. Is either blank or one of the following strings: Single, In a Relationship, In an Open Relationship, Engaged, Married, It's Complicated, Widowed. ▪ religion - User-entered "Religious Views" profile field. No guaranteed formatting. ▪ sex - User-entered "Sex" profile file. This is a translated string, so the gender returned depends on the user's locale. This field may be blank. ▪ significant_other_id - the id of the person the user is in a relationship with. Only shown if both people in the relationship are users of the application making the request. ▪ status - Contains a "message" child with user-entered status information, as well as a "time" child with the time (in seconds since epoch) at which the status message was set. ▪ timezone - offset from GMT (e.g. California is -8). ▪ tv - User-entered "Favorite TV Shows" profile field. No guaranteed formatting. ▪ username - The user's Facebook username, if one was specified. Otherwise, this field is blank. ▪ wall_count - Total number of posts to the user's wall. Note that this does not include items with attachments, i.e. wall photos, wall videos, posted links, etc. Only items that show up on http://www.facebook.com/wall.php are included in this count. ▪ website - User-entered personal website profile field. No guaranteed formatting. ▪ work_history - list of work history information, as work_info elements, each of which contain location, company_name, position, description, start_date and end_date child elements. If no work history information is returned, this element is blank. <ul style="list-style-type: none"> - location is user-entered, and has a similar format to current_location and hometown_location above. - company_name is user-entered, and does not necessarily correspond to a Facebook work network. - description is user-entered, and may be blank. - position is user-entered, and may be blank. - start_date is of the form YYYY-MM, YYYY, or MM. It may be blank. - end_date is of the form YYYY-MM, YYYY, or MM. It may be blank.)
--	--

HTTP Method(s):	GET		
Requires Authentication:	Yes (This method can be called by using a session secret, and not the application secret (for example, for a Facebook Connect site or desktop application))		
API rate limited:	No		
Parameters:	Required	Type	Description
api_key	Yes	string	The application key associated with the calling application. If the API key is specified in the client, it is not needed to pass it with every call.
call_id	Yes	float	The sequence number of the request. Each successive call for any session must use a sequence number greater than the last. Facebook suggests using the current time in milliseconds, such as PHP's "microtime (true)" function. If the call ID is specified at the client, it doesn't have to be passed with every call.
sig	Yes	string	An MD5 hash of the current request and the secret key of the application, as described in the "How Facebook Authenticates The application" documentation. Facebook computes the signature automatically.
v	Yes	string	This parameter must be set to 1.0 to use the current version of the API. If the version is specified at the client of the application, it doesn't have to be passed with every call.
uids	Yes	array	List of user IDs. This is a comma-separated list of user IDs.
fields	Yes	array	List of desired fields in return. This is a comma-separated list of field strings.
session_key	Optional	string	The session key of the logged user, or the session key provided when the user granted the application the "offline_access" extended permission. The session key is automatically included by the PHP client of Facebook.
format	Optional	string	The desired response format, which can be either XML or JSON. (Default value is XML.)
callback	Optional	string	The name of a function to call. This enables cross-domain JavaScript requests using the <script> tag (also known as JSONP) and works both with the XML and JSON formats. This function will be called with the response passed as the parameter.

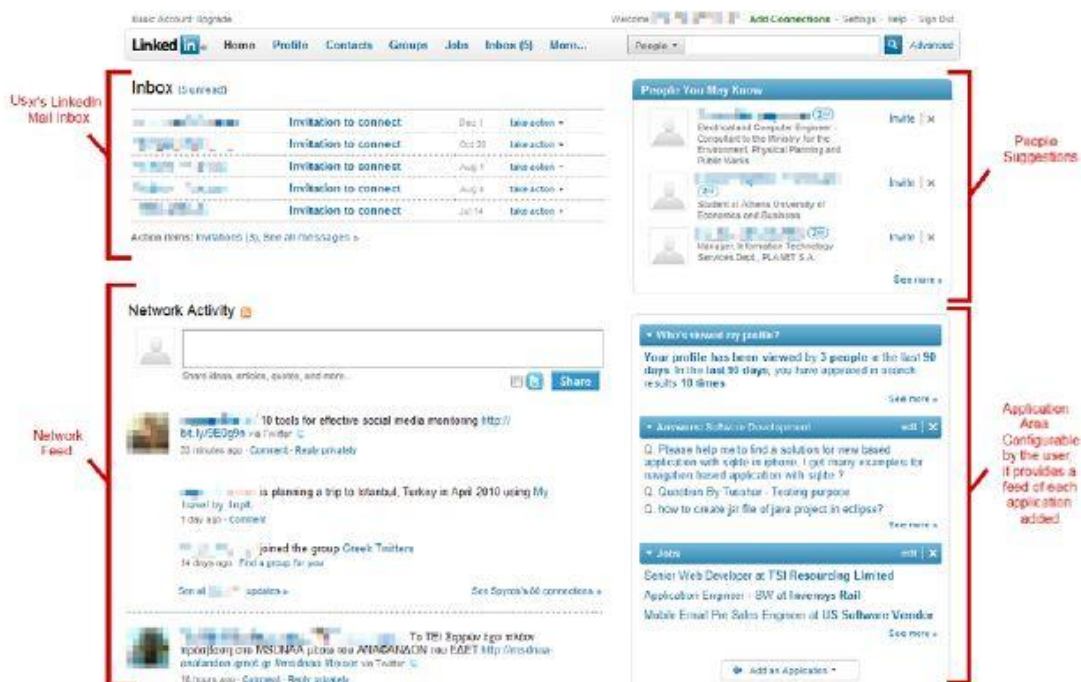
4.6. - LinkedIn

Το LinkedIn έχει το δικό του API που επιτρέπει στους χρήστες να συνδέσουν τα επαγγελματικά τους δίκτυα με οποιαδήποτε τοποθεσία ή υπηρεσία. Το API παρέχει πρόσβαση σε σχεδόν κάθε στοιχείο δεδομένων της υπηρεσίας. Είναι ένα REST - like API και απαιτεί έλεγχο αυθεντικοποίησης για τα περισσότερα από τα αιτήματα χρησιμοποιώντας το πρότυπο OAuth.

Εκτός από το βασικό API που αναφέραμε προηγουμένως, το LinkedIn παρέχει widgets μέσα από το LinkedIn Widget API. Το Widget API επιτρέπει την ενσωμάτωση μερικών από των λειτουργιών του LinkedIn σε τρίτες σελίδες. Μερικά από τα πιο αξιοσημείωτα προσφερόμενα widgets είναι τα εξής:

- Company Insider: Επιτρέπει στους χρήστες να ανακαλύψουν τον τρόπο που συνδέονται με τις εταιρείες σε μια τοποθεσία.
- Profile: Εμφανίζει προφίλ από χρήστες του LinkedIn σε ένα blog, ένα website ή μια εφαρμογή.
- Share On LinkedIn: Προσθέτει την επιλογή Share on LinkedIn σε οποιαδήποτε σελίδα ή blog επιτρέποντας στους χρήστες να μοιραστούν οποιοδήποτε περιεχόμενο με τις συνδέσεις τους στο LinkedIn.

Το Widget API δεν θα αναλυθεί καθόλου καθώς δεν πραγματοποιείται χρήση στα πλαίσια ανάπτυξης της εφαρμογής SocialCampaign.



4.4. - LinkedIn Interface

API Method:	Post Network Update [398]		
Description:	<p>The Post Network Update API allows sending in an activity from a site or application to be posted to the first degree connections of the LinkedIn user. Network updates are the LinkedIn term for news feed activities and appear on the centre column of the home page.</p> <p>Example XML Body:</p> <pre><activity locale="en_US"> <content-type> linkedin-html </content-type> <body> Something to post </body> </activity></pre>		
URL:	http://api.linkedin.com/v1/people/~person-activities		
Formats:	A successful response will be indicated by a 201 Created HTTP status code and an empty response body.		
HTTP Method(s):	POST		
Requires Authentication:	true		
API restrictions:	<u>Daily Limits:</u> Application: 5,000 Per User: 5		
Parameters:	Required	Type	Description
API key	Yes	Key	The application's LinkedIn API key
Access Token	Yes	Token	The Access Token for any API calls to LinkedIn to identify the user on whose behalf the client is making the call.
activity	Yes	Nodes for the activity being created	Parent node for the update. Should include a locale attribute indicating the IETF Language code of the contents. If unsure, the client should specify "American English" with "en_US".
content-type	Yes	'linkedin-html'	Indicates that there may be a limited subset of HTML within the contents of the update.
body	Yes	string	The actual content of the update. It is possible to use HTML to include links to the user name and the content the user created. Other HTML tags are not supported. All body text should be HTML entity escaped and UTF-8 compliant.

API Method:	Get Network Updates API [403]		
Description:	Returns the user's network updates, which is the LinkedIn term for the user's feed. This call returns most of what shows up in the middle column of the LinkedIn.com home page.		
URL:	http://api.linkedin.com/v1/people/~network		
Formats:	xml		
HTTP Method(s):	GET		
Requires Authentication:	true		
API restrictions:	<u>Daily Limits:</u> Application: 500.000 Per User: 300		
Parameters:	Required	Type	Description
API key	Yes	Key	The application's LinkedIn API key
Access Token	Yes	Token	The Access Token for any API calls to LinkedIn to identify the user on whose behalf the client is making the call.
type	Optional	Network Update Type	Any valid Network Update Type from the table below. Types must be in all capital letters.
count	Optional	int	Ranges are specified with a starting index and a number of results to return. The default value is 10 and that is the maximum number returned. Requesting more will return 10.
start	Optional	int	The offset by which to start Network Update pagination

API Method:	Get More Comments on a Specific Network Update [404]		
Description:	By default, a network update containing more than 5 comments will only return the most recent five. This method is designed to retrieve all comments for a given network update		
URL:	http://api.linkedin.com/v1/people/~network/updates/key={NETWORK UPDATE KEY}/update-comments		
Formats:	xml		
HTTP Method(s):	GET		
Requires Authentication:	true		
API restrictions:	<u>Daily Limits:</u> Application: 500.000 Per User: 300		
Parameters:	Required	Type	Description
API key	Yes	Key	The application's LinkedIn API key
Access Token	Yes	Token	The Access Token for any API calls to LinkedIn to identify the user on whose behalf the client is making the call.
NETWORK UPDATE KEY	Yes	update/update-key	An update/update-key representing an update.

API Method:	Profile API [405]		
Description:	The Profile API returns a member's LinkedIn profile. The access token will be used to determine the profile to return.		
URL:	http://api.linkedin.com/v1/people/Prof_API_profile_type_field_selector		
Formats:	xml		
HTTP Method(s):	GET		
Requires Authentication:	true		
API restrictions:	Daily Limits: Application: 100.000 Per User: 50		
Parameters:	Required	Type	Description
API key	Yes	Key	The application's LinkedIn API key
Access Token	Yes	Token	The Access Token for any API calls to LinkedIn to identify the user on whose behalf the client is making the call.
Prof_API	Yes	string	~ : the tilde character returns the profile of the current user. The access token will be used to determine the profile to return. id=member_id: indicates the use of a member token. Requesting by member token only returns a profile from the logged in person's three degree network and doesn't return any out-of-network profiles. A member token cannot be used to get a public profile. url=API URL: Requests profile from a search result. url=Public Profile URL: returns a profile from inside or outside the logged in member's three degree network. The Public Profile URL must be URL escaped.
profile_type	Optional	':public'	To return a public profile, the client must add :public to the end of the profile. Without that, the call will return a standard profile..
field_selector	Optional	:(field1,field2...)	Is used to select specific fields. All fields must be at the same level of depth in the XML.

4.7. - Google Charts API

Η Google μέσα από το Google Chart API παρέχει έναν τέλειο τρόπο για να απεικονιστούν τα δεδομένα στις web εφαρμογές. Από απλά line charts, σε πολύπλοκα ιεραρχικά tree maps, το Google Charts API παρέχει ένα μεγάλο αριθμό από καλά σχεδιασμένους τύπους γραφημάτων. Η ομαδοποίηση και εμφάνιση των δεδομένων είναι μια αρκετά εύκολη υπόθεση χρησιμοποιώντας μια πλήρη σειρά από εργαλεία που το επιτρέπουν.

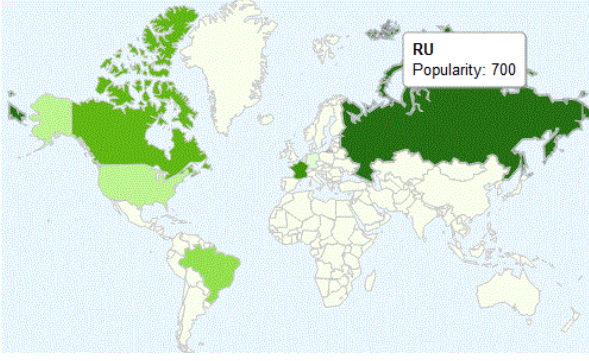
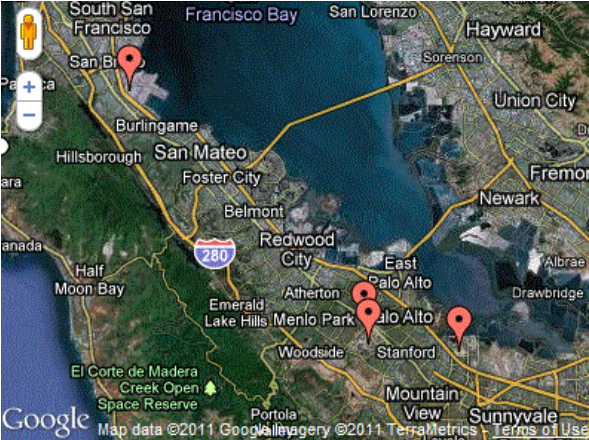
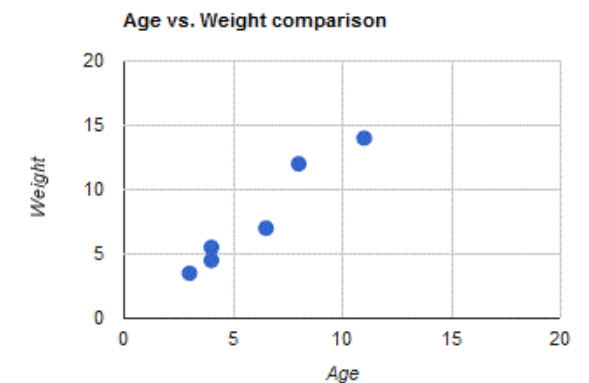
Τα γραφήματα παρουσιάζονται ως JavaScript κλάσεις. Το Google Charts API παρέχει πολλούς τύπους γραφημάτων. Αν και η προεπιλεγμένη εμφάνιση είναι η προτεινόμενη για τις περισσότερες περιπτώσεις, δίνεται η δυνατότητα τα γραφήματα να μπορούν να προσαρμοστούν εύκολα έτσι ώστε να ταιριάζουν καλύτερα στις εφαρμογές στις οποίες χρησιμοποιούνται. Τα γραφήματα είναι άκρως διαδραστικά και επιτρέπουν την μεταξύ τους σύνδεση με σκοπό τη δημιουργία πιο σύνθετων δομών παρουσίασης των δεδομένων της εφαρμογής. Τα γραφήματα εμφανίζονται με τη χρήση της τεχνολογίας HTML5/SVG για να παρέχουν συμβατότητα με μεγάλο εύρος από browsers (συμπεριλαμβανομένης και της VML για τους παλιές εκδόσεις των browsers) όπως και επίσης φορητότητα σε διάφορες σταθερές και φορητές πλατφόρμες. Δεν απαιτούνται πρόσθετα plug-ins.

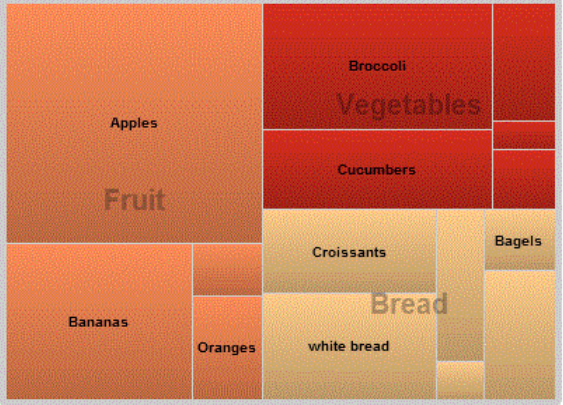
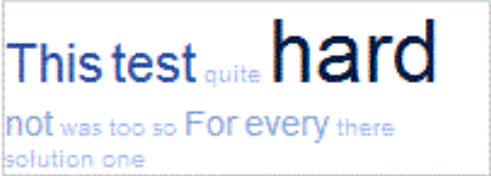
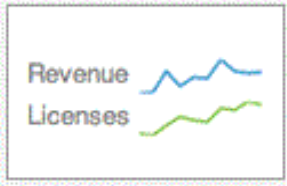
Το Datasource πρωτόκολλο του Google Charts API περιλαμβάνει μια γλώσσα ερωτημάτων που βασίζεται στο λεξιλόγιο της SQL, επιτρέποντας έτσι την απευθείας σύνδεση των γραφημάτων με πίνακες δεδομένων αλλά ταυτόχρονα και την επιλογή των δεδομένων με διάφορα κριτήρια επιλογής.

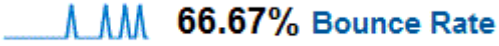
Το Google Charts API παρέχει διάφορα είδη γραφημάτων, τα βασικότερα από τα οποία παρουσιάζονται στον παρακάτω πίνακα[7].

Visualisation Type	Description	Example												
Pie Chart	A circle graph divided in slices, whose arc length is proportional to the quantity it represents.	<p>My Daily Activities</p> <p>A pie chart titled "My Daily Activities" showing the distribution of time spent on various activities. The largest slice is Work at 45.8%, followed by Sleep at 29.2%. The remaining activities (Watch TV, Commute, and Eat) each account for 8.3% of the total time.</p> <table border="1"> <thead> <tr> <th>Activity</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>Work</td> <td>45.8%</td> </tr> <tr> <td>Sleep</td> <td>29.2%</td> </tr> <tr> <td>Watch TV</td> <td>8.3%</td> </tr> <tr> <td>Commute</td> <td>8.3%</td> </tr> <tr> <td>Eat</td> <td>8.3%</td> </tr> </tbody> </table>	Activity	Percentage	Work	45.8%	Sleep	29.2%	Watch TV	8.3%	Commute	8.3%	Eat	8.3%
Activity	Percentage													
Work	45.8%													
Sleep	29.2%													
Watch TV	8.3%													
Commute	8.3%													
Eat	8.3%													

<p>Column Chart</p>	<p>A vertical bar chart where the height of each bar is controlled by the value it represents. Each color corresponds to a data series.</p>	<p>Company Performance</p> <table border="1"> <thead> <tr> <th>Year</th> <th>Sales</th> <th>Expenses</th> </tr> </thead> <tbody> <tr> <td>2004</td> <td>1000</td> <td>400</td> </tr> <tr> <td>2005</td> <td>1170</td> <td>450</td> </tr> <tr> <td>2006</td> <td>650</td> <td>1100</td> </tr> <tr> <td>2007</td> <td>1000</td> <td>500</td> </tr> </tbody> </table>	Year	Sales	Expenses	2004	1000	400	2005	1170	450	2006	650	1100	2007	1000	500
Year	Sales	Expenses															
2004	1000	400															
2005	1170	450															
2006	650	1100															
2007	1000	500															
<p>Bar Chart</p>	<p>The horizontal version of the Column Chart.</p>	<p>Company Performance</p> <table border="1"> <thead> <tr> <th>Year</th> <th>Sales</th> <th>Expenses</th> </tr> </thead> <tbody> <tr> <td>2004</td> <td>1000</td> <td>400</td> </tr> <tr> <td>2005</td> <td>1170</td> <td>450</td> </tr> <tr> <td>2006</td> <td>650</td> <td>1100</td> </tr> <tr> <td>2007</td> <td>1000</td> <td>500</td> </tr> </tbody> </table>	Year	Sales	Expenses	2004	1000	400	2005	1170	450	2006	650	1100	2007	1000	500
Year	Sales	Expenses															
2004	1000	400															
2005	1170	450															
2006	650	1100															
2007	1000	500															
<p>Line Chart</p>	<p>A line chart displays information as a series of data points connected by straight line segments. Each data series is displayed as a separate line. A line chart is often used to visualize a trend in data over intervals of time – a time series.</p>	<p>Company Performance</p> <table border="1"> <thead> <tr> <th>Year</th> <th>Sales</th> <th>Expenses</th> </tr> </thead> <tbody> <tr> <td>2004</td> <td>1000</td> <td>400</td> </tr> <tr> <td>2005</td> <td>1170</td> <td>450</td> </tr> <tr> <td>2006</td> <td>650</td> <td>1100</td> </tr> <tr> <td>2007</td> <td>1000</td> <td>500</td> </tr> </tbody> </table>	Year	Sales	Expenses	2004	1000	400	2005	1170	450	2006	650	1100	2007	1000	500
Year	Sales	Expenses															
2004	1000	400															
2005	1170	450															
2006	650	1100															
2007	1000	500															
<p>Area Chart</p>	<p>An area chart is based on the line chart and displays graphically quantitative data. The area between axis and line is commonly emphasized. Area charts are used to represent and compare two or more series of cumulated totals using numbers or percentages (stacked area charts in this case) over time.</p>	<p>Company Performance</p> <table border="1"> <thead> <tr> <th>Year</th> <th>Sales</th> <th>Expenses</th> </tr> </thead> <tbody> <tr> <td>2004</td> <td>1000</td> <td>400</td> </tr> <tr> <td>2005</td> <td>1170</td> <td>450</td> </tr> <tr> <td>2006</td> <td>650</td> <td>1100</td> </tr> <tr> <td>2007</td> <td>1000</td> <td>500</td> </tr> </tbody> </table>	Year	Sales	Expenses	2004	1000	400	2005	1170	450	2006	650	1100	2007	1000	500
Year	Sales	Expenses															
2004	1000	400															
2005	1170	450															
2006	650	1100															
2007	1000	500															

<p>Geo Chart</p>	<p>A Geo Chart is an SVG¹-based map of a country, continent, or region map, with colors and values assigned to specific regions. Values are displayed as a color scale, and optional hover text can be specified for regions.</p>													
<p>Map</p>	<p>A map displayed using the Google Maps API. Data values are displayed as points on the map and can be specified as coordinates (Latitude-Longitude pairs) or addresses.</p>													
<p>Scatter Chart</p>	<p>A scatter chart is used to map correlation between sets of numbers. Each data series is displayed with a separate color.</p>	 <table border="1"> <caption>Age vs. Weight comparison</caption> <thead> <tr> <th>Age</th> <th>Weight</th> </tr> </thead> <tbody> <tr> <td>3</td> <td>4</td> </tr> <tr> <td>4</td> <td>5</td> </tr> <tr> <td>5</td> <td>6</td> </tr> <tr> <td>8</td> <td>12</td> </tr> <tr> <td>11</td> <td>14</td> </tr> </tbody> </table>	Age	Weight	3	4	4	5	5	6	8	12	11	14
Age	Weight													
3	4													
4	5													
5	6													
8	12													
11	14													

<p>Tree Map Chart</p>	<p>A visual representation of a data tree, where each node can have zero or more children, and one parent (except for the root, which has no parents). Each node is displayed as a rectangle, sized and colored according to values that has been assigned. Sizes and colors are valued relative to all other nodes in the graph. The size and color of a leaf node can be specified whereas a non-leaf node will be displayed as a bounding box for leaf nodes. The default behavior is to move down the tree when a user left-clicks a node, and to move back up the tree when a user right-clicks the graph.</p>	
<p>Term Cloud</p>	<p>A list of terms, where the size and color of each word is determined by a specified frequency value (typically the number of times it appears in some text and the positive/negative/neutral value).</p>	
<p>Sparkline</p>	<p>Single or multiple sparklines present trends and variations associated with some measurement in a simple and condensed way. A sparkline is characterized</p>	

	by its small size and high data density.	
Metric	A percentage or simple number that is associated with the value of a measurement.	

4.5. - Πίνακας με τα βασικότερα Google Charts

Όλα τα γραφήματα απαιτούν την παροχή στοιχείων που προέρχονται από ένα αρχείο προέλευσης δεδομένων που υποστηρίζει το Visualisation API (π.χ. Google spreadsheets). Ένα αρχείο προέλευσης δεδομένων είναι μια διαδικτυακή υπηρεσία που μπορεί να παρέχει δεδομένα σε ένα γράφημα Javascript. Ένα αρχείο προέλευσης δεδομένων εκθέτει ένα URL, που ονομάζεται URL προέλευσης δεδομένων, στο οποίο το διάγραμμα στέλνει HTTP GET αιτήσεις. Σε απάντηση, το αρχείο προέλευσης δεδομένων επιστρέφει τα δεδομένα σε κατάλληλη μορφή έτσι ώστε το διάγραμμα να μπορεί να τα χρησιμοποιήσει για να δημιουργήσει το γράφημα στη σελίδα. Αυτό το αίτημα-απάντηση πρωτόκολλο είναι γνωστό ως Google Visualization API wire πρωτόκολλο. Τα δεδομένα που εξυπηρετούνται από ένα αρχείο προέλευσης δεδομένων μπορούν να εξαχθούν από διάφορους πόρους, όπως ένα αρχείο ή βάση δεδομένων.

- Request Format

Ο client στέλνει ένα HTTP GET αίτημα με διάφορες παραμέτρους, συμπεριλαμβανομένων των προσαρμοσμένων στοιχείων, μια προαιρετική σειρά ερωτημάτων, υπογραφή, και άλλα στοιχεία. Η πηγή των δεδομένων λαμβάνει και αναλύει την αίτηση, όπως περιγράφεται στη request φόρμα. Η μορφή της αντίδρασης εξαρτάται από την παράμετρο *out* του αιτήματος, η οποία τελικά καθορίζει τον τύπο της αντίδρασης. Η προκαθορισμένη τιμή της παραμέτρου *out* είναι μια συμβολοσειρά JSON.

- JSON Response Format

Η πηγή δεδομένων προετοιμάζει τα δεδομένα στη μορφή που ζητούνται, δημιουργώντας έτσι έναν πίνακα JSON. Η πηγή δεδομένων δημιουργεί μια απόκριση HTTP που περιλαμβάνει τα δεδομένα αλλά και λοιπές παραμέτρους απάντησης και τα στέλνει πίσω στον client σε response μορφή. Μια απάντηση JSON περιλαμβάνει τα δεδομένα σε ένα αντικείμενο JavaScript που μπορεί να μεταδοθεί απευθείας στον κατασκευαστή του γραφήματος και να το συμπληρώσει.

Παρακάτω ακολουθεί ένα απλό παράδειγμα ενός πίνακα δεδομένων και ενός ζεύγους JSON request και response.

```
{cols:[{id:'Col1',label:'',type:'number'}],
rows:[{c:[{v:1.0,f:'1'}]},
      {c:[{v:2.0,f:'2'}]},
      {c:[{v:3.0,f:'3'}]},
      {c:[{v:1.0,f:'1'}]}
    ]
}
```

Returns the basic information with a three column, four row table.

Request:

`http://www.example.com/mydatasource?tqx=reqId:0;sig:4641982796834063168;out:JSON`

Response

```
google.visualization.Query.setResponse({version:'0.6',reqId:'0',status:'ok',sig:'5982206968295329967',table:{cols:[{id:'Col1',label:'',type:'number'},{id:'Col2',label:'',type:'number'},{id:'Col3',label:'',type:'number'}],rows:[{c:[{v:1.0,f:'1'},{v:2.0,f:'2'},{v:3.0,f:'3'}]},{c:[{v:2.0,f:'2'},{v:3.0,f:'3'},{v:4.0,f:'4'}]},{c:[{v:3.0,f:'3'},{v:4.0,f:'4'},{v:5.0,f:'5'}]},{c:[{v:1.0,f:'1'},{v:2.0,f:'2'},{v:3.0,f:'3'}]}}});
```

google.visualization.Query[7]

Method	Return Value	Description
<code>abort()</code>	None	Stops the automated query sending that was started with <code>setRefreshInterval()</code> .
<code>setRefreshInterval(seconds)</code>	None	Sets the query to automatically call the <code>send</code> method every specified duration (number of seconds), starting from the first explicit call to <code>send</code> . <code>seconds</code> is a number greater than or equal to zero. If you use this method, you should call it before calling the <code>send</code> method. Cancel this method either by calling it again with zero (the default), or by calling <code>abort()</code> .
<code>setTimeout(seconds)</code>	None	Sets the number of seconds to wait for the data source to respond before

		raising a timeout error. <code>seconds</code> is a number greater than zero. The default timeout is 30 seconds. This method, if used, should be called before calling the <code>send</code> method.
<code>setQuery(string)</code>	None	Sets the query string. The value of the <code>string</code> parameter should be a valid query. This method, if used, should be called before calling the <code>send</code> method.
<code>send(callback)</code>	None	Sends the query to the data source. <code>callback</code> should be a function that will be called when the data source responds. The callback function will receive a single parameter of type <code>google.visualization.QueryResponse</code> .

4.6. - query σε data source

google.visualization.QueryResponse[7]

Method	Return Value	Description
<code>getDataTable()</code>	DataTable	Returns the data table as returned by the data source. Returns <code>null</code> if the query execution failed and no data was returned.
<code>getDetailedMessage()</code>	String	Returns a detailed error message for queries that failed. If the query execution was successful, this method returns an empty string. The message returned is a message that is intended for developers, and may contain technical information, for example 'Column {salary} does not exist'.
<code>getMessage()</code>	String	Returns a short error message for queries that failed. If the query execution was successful, this method returns an empty string. The message returned is a short message that is

		intended for end users, for example 'Invalid Query' or 'Access Denied'.
<code>getReasons()</code>	Array of strings	Returns an array of zero or more entries. Each entry is a short string with an error or warning code that was raised while executing the query. Possible codes: <ul style="list-style-type: none"> • <code>access_denied</code> The user does not have permissions to access the data source. • <code>invalid_query</code> The specified query has a syntax error. • <code>data_truncated</code> One or more data rows that match the query selection were not returned due to output size limits. (warning). • <code>timeout</code> The query did not respond within the expected time.
<code>hasWarning()</code>	Boolean	Returns <code>true</code> if the query execution has any warning messages.
<code>isError()</code>	Boolean	Returns <code>true</code> if the query execution failed, and the response does not contain any data table. Returns <code><false></code> if the query execution was successful and the response contains a data table.

4.7. - query response

Ο πίνακας[7] που ακολουθεί συνοψίζει τις διαφορετικές μορφές των δεδομένων που κάθε τύπος γραφήματος αναμένει να παρουσιάσει.

Visualisation Type	Data Table		
Pie Chart	Column	Type	Description
		0	string
	1	number	slice value
Column Chart Bar Chart	Column	Type	Description
		1	string

	2	number	bar value (data series 1)
	N	number	bar value (data series N)
	<i>Each row in the table represents a group of adjacent bars. The value at a given row and column controls the height of the single bar represented by this row and column.</i>		
Area Chart	Column	Type	Description
	0	string	category label
	1	number	value (data series 1)
	N	number	value (data series N)
	<i>Each numeric column represents a separate series in the chart.</i>		
Line Chart	Column	Type	Description
	0	string	category label
	1	number	value (data series 1)
	N	number	value (data series N)
	<i>Each column is displayed as a separate line.</i>		
Geo Chart	Column	Type	Description
	0	string	A map location. The following formats are accepted: <ul style="list-style-type: none"> · A specific address (for example, "1600 Pennsylvania Ave").

			<ul style="list-style-type: none"> · A country name as a string (for example, "England"), or an uppercase ISO-3166 code or its English text equivalent (for example, "GB" or "United Kingdom"). · An uppercase ISO-3166-2 region code name or its English text equivalent (for example, "US-NJ" or "New Jersey"). · A metropolitan area code. These are three-digit metro codes used to designate various regions; US codes only supported. Note that these are <i>not</i> the same as telephone area codes.
	1 [optional]	number	value
Map	Column	Type	Description
	0	number	latitude
	1	number	longitude
	2 [optional]	string	description of the location specified in the first two columns.
Scatter Chart	Column	Type	Description
	0	number	x-axis value
	1	number	y-axis value data series 1
	N	number	y-axis value data series N
	<i>Each column is displayed with a separate color.</i>		

Tree Map Chart	Column	Type	Description
	0	string	ID for this node. It can be any valid JavaScript string, including spaces, and any length that a string can hold. This value is displayed as the node header.
	1	string	The ID of the parent node. If this is a root node, it is blank.
	2	number	The size of the node. Any positive value is allowed. This value determines the size of the node, computed relative to all other nodes currently shown. This value is ignored for non-leaf nodes (it is actually calculated from the size of all its children).
	3 [optional]	number	An optional value used to calculate a color for this node. Any value, positive or negative, is allowed. The color value is first recomputed on a scale from minColorValue to maxColorValue, and then the node is assigned a color from the gradient between minColor and maxColor.
	<i>Each row in the data table describes one node (a rectangle in the graph). Each node (except the root node) has one or</i>		

	<i>more parent nodes. Each node is sized and colored according to its values relative to the other nodes currently shown.</i>		
Term Cloud	Column	Type	Description
	0	string	term
	1	number	frequency of term
	2 [optional]	string	url that should be opened when the term is clicked
Sparkline	Column	Type	Description
	0	number	value
	N	number	value

4.8. - Τύπος δεδομένων ανά γράφημα

Κεφάλαιο 5^ο - Μεθοδολογία

Προτού προχωρήσουμε στην περιγραφή της υλοποίησης της εφαρμογής **Social Campaign** θα περιγράψουμε εν συντομία σε αυτό το κεφάλαιο τα βήματα που ακολουθήθηκαν καθ' όλη τη διάρκεια εκπόνησης της παρούσης διπλωματικής εργασίας.

5.1. - Μελέτη Concept Padgets

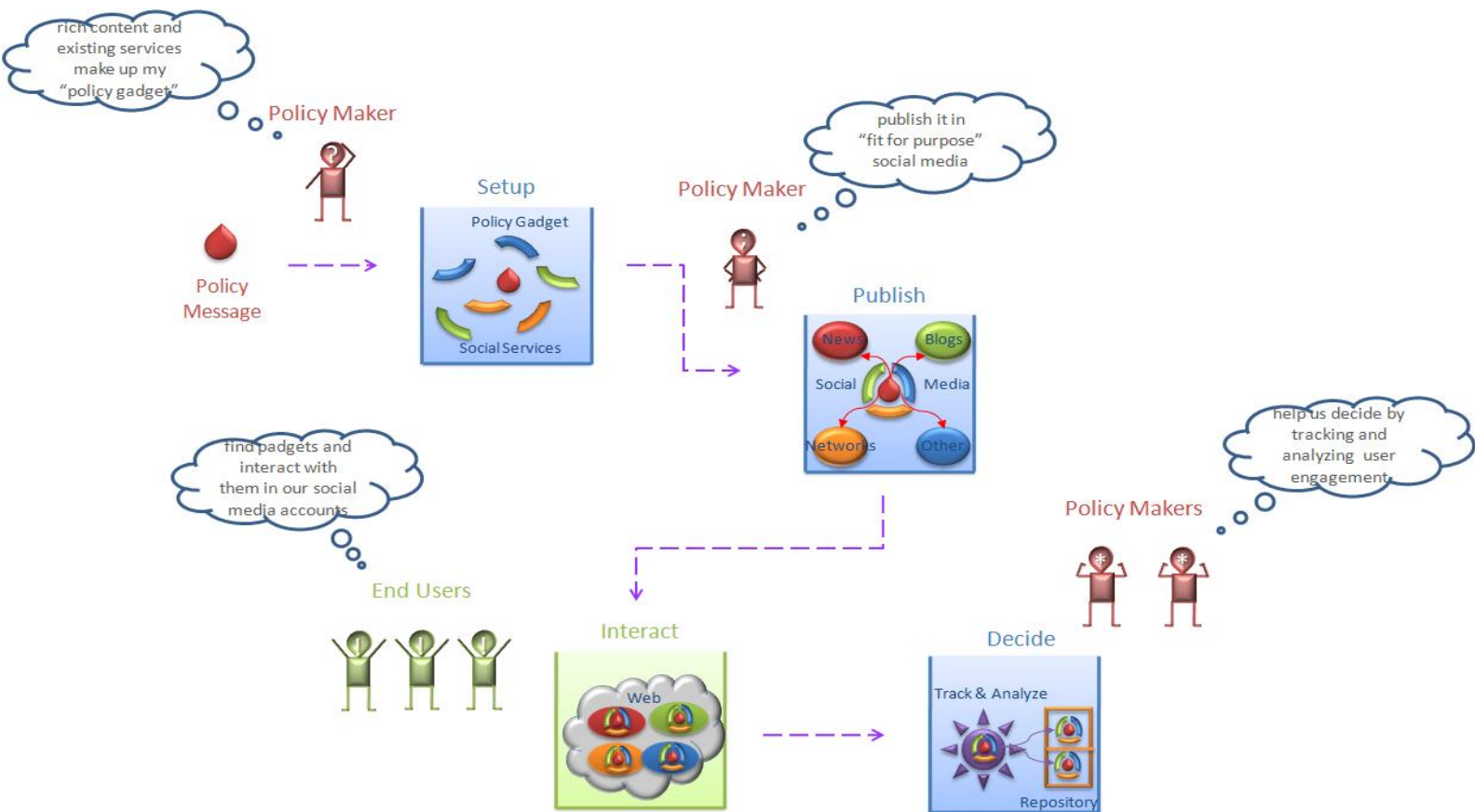
Το πρώτο βήμα που έγινε κατά τη διάρκεια εκπόνησης της εργασίας ήταν η μελέτη της φιλοσοφίας του έργου Padgets. Η φιλοσοφία του Padgets περιγράφεται εν συντομία παρακάτω: (αναλυτική περιγραφή του Padgets ακολουθεί στο κεφάλαιο 3)

- Οι κυβερνήσεις ξοδεύουν μεγάλα ποσά για την ανάπτυξη ιστοτόπων με σκοπό τη ηλεκτρονική συμμετοχή των πολιτών σε θέματα πολιτικής, η χρήση των οποίων όμως δεν επιφέρει τα επιθυμητά αποτελέσματα.
- Η αλματώδης ανάπτυξη των μέσων κοινωνικής δικτύωσης προσφέρει τη δυνατότητα η ηλεκτρονική συμμετοχή να επιτευχθεί αποτελεσματικά μέσα από το feedback των χρηστών.
- Το Έργο Padgets εισάγει μια νέα προσέγγιση στην προώθηση πολιτικής εφαρμόζοντας τη φιλοσοφία του Web 2.0 πηγαίνοντας ακριβώς εκεί όπου βρίσκονται οι χρήστες στο διαδίκτυο.



5.1. - Φιλοσοφία Padgets

- Η διάδοση των πολιτικών μηνυμάτων γίνεται μέσω πολλαπλών μέσων κοινωνικής δικτύωσης ταυτόχρονα, χρησιμοποιώντας μια ενιαία, ολοκληρωμένη διεπαφή επικοινωνίας λαμβάνοντας επιπλέον την ανταπόκριση των χρηστών.



5.2. - Προσομοίωση μιας καμπάνιας μέσω του Padgets

5.2. - Μελέτη Social Media APIs

Αφού μελετήθηκε η φιλοσοφία του προγράμματος Padgets και τα APIs των Social Media (Facebook, Twitter, LinkedIn, Youtube, Blogger, Picasa), ξεκίνησε η δοκιμαστική χρήση των APIs των παραπάνω Social Media με τη δημιουργία μικροεφαρμογών οι οποίες εκτελούσαν ορισμένες λειτουργίες στο κάθε Social Media ξεχωριστά, χωρίς κάποιο User Interface και μόνο με χρήση της γλώσσας προγραμματισμού PHP.

Για την κάθε εφαρμογή η οποία υλοποιήθηκε, χρησιμοποιήθηκε και μια αντίστοιχη βιβλιοθήκη για την γλώσσα PHP. Πιο συγκεκριμένα, χρησιμοποιήθηκαν αντικειμενοστραφείς βιβλιοθήκες για το κάθε Application Programming Interface οι οποίες ήταν οι παρακάτω:

- Facebook: Για τη χρήση του Facebook API χρησιμοποιήθηκε η βιβλιοθήκη Facebook PHP SDK. Η συγκεκριμένη βιβλιοθήκη, έχει υλοποιηθεί από την ομάδα του Facebook, είναι μια βιβλιοθήκη ανοιχτού κώδικα και διανέμεται υπό την άδεια Apache License 2.0.

- Twitter: Για τη χρήση του Twitter API χρησιμοποιήθηκε η βιβλιοθήκη TtitterOauth. Η συγκεκριμένη βιβλιοθήκη, έχει υλοποιηθεί από τον Abraham Williams, είναι μια βιβλιοθήκη ανοιχτού κώδικα και διανέμεται υπό την άδεια MIT License.
- LinkedIn: Για τη χρήση του LinkedIn API χρησιμοποιήθηκε η βιβλιοθήκη Simple LinkedIn. Η συγκεκριμένη βιβλιοθήκη, έχει υλοποιηθεί από τον Paul Menega, είναι μια βιβλιοθήκη ανοιχτού κώδικα και διανέμεται υπό την άδεια MIT License.
- Youtube: Για τη χρήση του Youtube API χρησιμοποιήθηκε η βιβλιοθήκη Youtube PHP client API. Η συγκεκριμένη βιβλιοθήκη, έχει υλοποιηθεί από την ομάδα του Youtube, είναι μια βιβλιοθήκη ανοιχτού κώδικα και διανέμεται υπό όρους χρήσης που έχει δημοσιεύσει η ομάδα του Youtube API.
- Blogger: Για τη χρήση του Blogger API χρησιμοποιήθηκε η βιβλιοθήκη Blogger Blogger Data API for PHP. Η συγκεκριμένη βιβλιοθήκη, έχει υλοποιηθεί από την ομάδα του Blogger, είναι μια βιβλιοθήκη ανοιχτού κώδικα και διανέμεται υπό όρους χρήσης που έχει δημοσιεύσει η ομάδα του Blogger API.
- Picasa: Για τη χρήση του Twitter API χρησιμοποιήθηκε η βιβλιοθήκη Picasa PHP Client Library. Η συγκεκριμένη βιβλιοθήκη, έχει υλοποιηθεί από την ομάδα του Picasa, είναι μια βιβλιοθήκη ανοιχτού κώδικα και διανέμεται υπό όρους χρήσης που έχει δημοσιεύσει η ομάδα του Picasa API.

Με τη βοήθεια των παραπάνω βιβλιοθηκών, οι μικροεφαρμογές οι οποίες δημιουργήθηκαν, βοήθησαν στην πρακτική κατανόηση των APIs και των προγραμματιστικών δυνατοτήτων που προσφέρουν σε τρίτες εφαρμογές ώστε να στείλουν και να λάβουν δεδομένα από αυτά τα έξι Social Media.

Πιο συγκεκριμένα οι μικρές εφαρμογές που υλοποιήθηκαν πραγματοποιούσαν τις εξής λειτουργίες:

Μικροεφαρμογή Facebook :

Σύνδεση στο Facebook ενός χρήστη με το λογαριασμό του, και αποστολή στο Facebook (στο προσωπικό του προφίλ), κείμενο μικρού μήκους (status update), κείμενο μεγάλου μήκους (note), δημιουργία φωτογραφικού άλμπουμ και αποστολή φωτογραφίας, αποστολή βίντεο. Επίσης δινόταν η δυνατότητα στο χρήστη να ονοματίσει με κάποιον τίτλο και κάποια περιγραφή το βίντεο το οποίο είχε αποστείλει, τη φωτογραφία, το φωτογραφικό άλμπουμ και το μεγάλο κείμενο.

Στη συνέχεια όλα αυτά τα στοιχεία τα οποία είχαν αποσταλεί στο Facebook, παρακολουθούνταν από την εφαρμογή, η οποία μετά από εντολή του συνδεδεμένου στο Facebook χρήστη συνέλεγε στοιχεία για τα likes τα οποία αντιστοιχούσαν στην κάθε

ανάρτηση (αριθμό, φύλο χρηστών, ηλικία χρηστών, τοποθεσία χρηστών) και τα comments (αριθμό, φύλο χρηστών, ηλικία χρηστών, τοποθεσία χρηστών, περιεχόμενο comment).

Μικροεφαρμογή Twitter :

Σύνδεση στο Twitter ενός χρήστη με το λογαριασμό του, και αποστολή κειμένου μικρού μήκους (tweet). Η ανάρτηση αυτή παρακολουθούνταν και μετά από επιλογή του χρήστη η μικροεφαρμογή επέστρεφε τον αριθμό των αναδημοσιεύσεων που αφορούσαν το μικρό κείμενο, καθώς και την τοποθεσία των χρηστών οι οποίοι αναδημοσίευσαν το μικρό αυτό κείμενο που είχε αναρτηθεί.

Μικροεφαρμογή LinkedIn :

Σύνδεση στο Twitter ενός χρήστη με το λογαριασμό του, και αποστολή κειμένου μικρού μήκους (Network Update). Η ανάρτηση αυτή παρακολουθούνταν και μετά από επιλογή του χρήστη η μικροεφαρμογή συνέλεγε στοιχεία για τα likes τα οποία αντιστοιχούσαν στην κάθε ανάρτηση (αριθμό, ηλικία χρηστών, τοποθεσία χρηστών) και τα comments (αριθμό, ηλικία χρηστών, τοποθεσία χρηστών, περιεχόμενο comment).

Μικροεφαρμογή Youtube :

Σύνδεση στο Youtube ενός χρήστη με το λογαριασμό του, και αποστολή ενός βίντεο (video upload). Η ανάρτηση αυτή παρακολουθούνταν και μετά από επιλογή του χρήστη η μικροεφαρμογή συνέλεγε στοιχεία για το πλήθος των εμφανίσεων του βίντεο, το πλήθος των comments που έγιναν στο συγκεκριμένο βίντεο, τη βαθμολογία του βίντεο, και το περιεχόμενο των comments. Επίσης, από τους χρήστες οι οποίοι είχαν κάνει κάποιο comment, η εφαρμογή συνέλλεγε την ηλικία τους, το φύλο τους και την τοποθεσία τους.

Μικροεφαρμογή Blogger :

Σύνδεση στο Blogger ενός χρήστη με το λογαριασμό του, και αποστολή ενός κειμένου μεγάλου μήκους (blog post). Η ανάρτηση αυτή παρακολουθούνταν και μετά από επιλογή του χρήστη η μικροεφαρμογή συνέλεγε στοιχεία για το περιεχόμενο των comments τα οποία την αφορούσαν και για τον αριθμό αυτών των comments.

Μικροεφαρμογή Picasa :

Σύνδεση στο Picasa ενός χρήστη με το λογαριασμό του, και αποστολή μιας φωτογραφίας. Η ανάρτηση αυτή παρακολουθούνταν και μετά από επιλογή του χρήστη η μικροεφαρμογή συνέλεγε στοιχεία για το περιεχόμενο των comments τα οποία την αφορούσαν και για τον αριθμό αυτών των comments.

5.3. - Μελέτη Visualization APIs

Τα δεδομένα τα οποία θα δεχόταν η εφαρμογή δεν θα είχαν κάποιο νόημα εάν δεν παρουσιαζόταν στον τελικό χρήστη με κάποια μορφή, η οποία θα ήταν απλή, κατανοητή αλλά παράλληλα περιγραφική και περιεκτική.

Για την αναπαράσταση των δεδομένων επιλέχθηκε το Google chart API το οποίο προσφέρει μεγάλη ποικιλία από είδη διαγραμμάτων, τα οποία μπορούσαν να περιγράψουν τόσο τα αριθμητικά δεδομένα - αποτελέσματα της εφαρμογής, όσο και τα δεδομένα τα οποία περιείχαν κείμενο και τοποθεσίες.

Αντίστοιχα, δημιουργήθηκαν ορισμένες μικροεφαρμογές, μία για κάθε είδος διαγράμματος από αυτά τα οποία μελετήθηκαν στην πράξη οι οποίες αντλούσαν δεδομένα από μια έτοιμη βάση δεδομένων με δοκιμαστικά δεδομένα και τα παρουσίαζαν σχηματικά.

5.4. - Ανάλυση απαιτήσεων εφαρμογής

Αφού έγινα όλες οι δοκιμές με τα APIs τα οποία είχε αποφασισθεί πως θα χρησιμοποιηθούν, διαδικασία η οποία διήρκησε αρκετό χρονικό διάστημα για το λόγο ότι η κάθε βιβλιοθήκη ήταν γραμμένη με διαφορετικό τρόπο και στυλ, ανάλογα με το προσωπικό στυλ του κάθε προγραμματιστή ο οποίος την είχε υλοποιήσει και στην πλειοψηφία τους συνοδευόταν από ελάχιστη έως και καθόλου τεκμηρίωση, πραγματοποιήθηκε η ανάλυση των απαιτήσεων της τελικής εφαρμογής η οποία ουσιαστικά θα ενοποιούσε όλες τις παραπάνω δοκιμαστικές μικροεφαρμογές σε μια ενιαία εφαρμογή.

Οι απαιτήσεις οι οποίες η εφαρμογή ικανοποιεί χωρίζονται σε δύο κατηγορίες, τις λειτουργικές και τις μη λειτουργικές απαιτήσεις. Ως λειτουργικές, χαρακτηρίζονται οι απαιτήσεις η οποίες αφορούν καθαρά τη λειτουργικότητα της εφαρμογής και το τι θα πρέπει να κάνει ώστε να ικανοποιεί στο ελάχιστο το σκοπό για τον οποίο έχει υλοποιηθεί. Οι μη λειτουργικές απαιτήσεις αφορούν σε θέματα τα οποία και να μην υλοποιηθούν η εφαρμογή και πάλι θα ικανοποιεί το σκοπό για τον οποίο θα έχει υλοποιηθεί.

Έπειτα από συζήτηση με τους επιβλέποντες τις διπλωματικής, οι απαιτήσεις διαμορφώνονται ως εξής:

5.4.1. - Λειτουργικές απαιτήσεις

- Ο χρήστης για να χρησιμοποιήσει την εφαρμογή θα πρέπει να κάνει εγγραφή δίνοντας Όνομα, επώνυμο, διεύθυνση ηλεκτρονικού ταχυδρομείου, όνομα χρήστη, και κωδικό πρόσβασης.

- Το όνομα, το επώνυμο, η διεύθυνση ηλεκτρονικού ταχυδρομείου, το όνομα χρήστη, και ο κωδικός πρόσβασης του χρήστη θα πρέπει να αποθηκεύονται σε βάση δεδομένων και να τα γνωρίζει μόνο ο χρήστης και ο διαχειριστής της εφαρμογής.
- Ο χρήστης θα έχει τη δυνατότητα να δημιουργεί καμπάνιες για το οτιδήποτε θέλει οι οποίες θα αποτελούνται από κείμενο μικρού μήκους, κείμενο μεγάλου μήκους, εικόνα και βίντεο.
- Ο χρήστης θα έχει τη δυνατότητα να επιλέξει σε ποια Social Media θα εμφανίζεται η καμπάνια η οποία θα έχει δημιουργήσει.
- Ο χρήστης θα μπορεί να δημιουργήσει καμπάνια σε όποια γλώσσα θέλει.
- Εάν ο χρήστης επιλέξει να δημιουργήσει καμπάνια η οποία θα δημοσιευθεί στο Facebook, θα πρέπει να εισάγει των κωδικό πρόσβασης και τα στοιχεία του στην εφαρμογή μέσω Facebook.
- Εάν ο χρήστης επιλέξει να δημιουργήσει καμπάνια η οποία θα δημοσιευθεί στο Twitter, θα πρέπει να εισάγει των κωδικό πρόσβασης και τα στοιχεία του στην εφαρμογή μέσω Twitter.
- Εάν ο χρήστης επιλέξει να δημιουργήσει καμπάνια η οποία θα δημοσιευθεί στο LinkedIn, θα πρέπει να εισάγει των κωδικό πρόσβασης και τα στοιχεία του στην εφαρμογή μέσω LinkedIn.
- Εάν ο χρήστης επιλέξει να δημιουργήσει καμπάνια η οποία θα δημοσιευθεί στο Youtube, θα πρέπει να εισάγει των κωδικό πρόσβασης και τα στοιχεία του στην εφαρμογή μέσω του λογαριασμού του στη Google.
- Εάν ο χρήστης επιλέξει να δημιουργήσει καμπάνια η οποία θα δημοσιευθεί στο Blogger, θα πρέπει να εισάγει των κωδικό πρόσβασης και τα στοιχεία του στην εφαρμογή μέσω του λογαριασμού του στη Google.
- Εάν ο χρήστης επιλέξει να δημιουργήσει καμπάνια η οποία θα δημοσιευθεί στο Picasa, θα πρέπει να εισάγει των κωδικό πρόσβασης και τα στοιχεία του στην εφαρμογή μέσω του λογαριασμού του στη Google.
- Εάν ο χρήστης επιλέξει να δημοσιεύσει περιεχόμενο στο Facebook, τότε θα πρέπει να ανεβάσει φωτογραφία, βίντεο, κείμενο μικρού μήκους, κείμενο μεγάλου μήκους.
- Εάν ο χρήστης επιλέξει να δημοσιεύσει περιεχόμενο στο Twitter, τότε θα πρέπει να ανεβάσει κείμενο μικρού μήκους.
- Εάν ο χρήστης επιλέξει να δημοσιεύσει περιεχόμενο στο LinkedIn, τότε θα πρέπει να ανεβάσει κείμενο μικρού μήκους.

- Εάν ο χρήστης επιλέξει να δημοσιεύσει περιεχόμενο στο Youtube, τότε θα πρέπει να ανεβάσει βίντεο.
- Εάν ο χρήστης επιλέξει να δημοσιεύσει περιεχόμενο στο Blogger, τότε θα πρέπει να ανεβάσει κείμενο μεγάλου μήκους.
- Εάν ο χρήστης επιλέξει να δημοσιεύσει περιεχόμενο στο Picasa, τότε θα πρέπει να ανεβάσει φωτογραφία.
- Εάν ο χρήστης επιλέξει να ανακτήσει δεδομένα από το Facebook, θα πρέπει να εισάγει των κωδικό πρόσβασης και τα στοιχεία του στην εφαρμογή μέσω Facebook.
- Εάν ο χρήστης επιλέξει να ανακτήσει δεδομένα από το Twitter, θα πρέπει να εισάγει των κωδικό πρόσβασης και τα στοιχεία του στην εφαρμογή μέσω Twitter.
- Εάν ο χρήστης επιλέξει να ανακτήσει δεδομένα από το LinkedIn, θα πρέπει να εισάγει των κωδικό πρόσβασης και τα στοιχεία του στην εφαρμογή μέσω LinkedIn.
- Εάν ο χρήστης επιλέξει να ανακτήσει δεδομένα από το Youtube, θα πρέπει να εισάγει των κωδικό πρόσβασης και τα στοιχεία του στην εφαρμογή μέσω του Google λογαριασμού του.
- Εάν ο χρήστης επιλέξει να ανακτήσει δεδομένα από το Blogger, θα πρέπει να εισάγει των κωδικό πρόσβασης και τα στοιχεία του στην εφαρμογή μέσω του Google λογαριασμού του.
- Εάν ο χρήστης επιλέξει να ανακτήσει δεδομένα από το Picasa, θα πρέπει να εισάγει των κωδικό πρόσβασης και τα στοιχεία του στην εφαρμογή μέσω του Google λογαριασμού του.
- Τα αριθμητικά δεδομένα τα οποία θα ανακτώνται από τα Social Media τα οποία έχουν επιλεχθεί, θα πρέπει να παρουσιάζονται σε Column charts και σε Bar charts
- Τα δεδομένα περιεχομένου γραπτού λόγου (σχόλια χρηστών) θα πρέπει να παρουσιάζονται με χρήση διαγραμμάτων Word Cloud και Table chart.
- Τα δημογραφικά δεδομένα (ηλικία χρηστών που έχουν αλληλεπιδράσει με το δημοσιευμένο περιεχόμενο, τοποθεσία χρηστών που έχουν αλληλεπιδράσει με το δημοσιευμένο περιεχόμενο, φύλο χρηστών που έχουν αλληλεπιδράσει με το δημοσιευμένο περιεχόμενο), θα πρέπει να παρουσιάζονται με χρήση διαγραμμάτων Geochart, Map, Pie chart, Column chart, Table και Word Cloud.

5.4.2. - Μη λειτουργικές απαιτήσεις

- Η φόρμα εγγραφής στο σύστημα να εμφανίζεται και να αποκρύπτεται με κινούμενο εφέ.
- Η φόρμα εισόδου στο σύστημα να εμφανίζεται και να αποκρύπτεται με κινούμενο εφέ.
- Κατά την εγγραφή να ζητείται από το χρήστη να εισάγει τον μυστικό κωδικό του δύο φορές για επιβεβαίωση.
- Κατά την εγγραφή, το όνομα του χρήστη θα πρέπει να αποτελείται από τρεις έως είκοσι χαρακτήρες και να περιέχει μόνο λατινικούς χαρακτήρες και τον χαρακτήρα της κάτω παύλας.
- Κατά την εγγραφή, το επώνυμο του χρήστη θα πρέπει να αποτελείται από τρεις έως είκοσι χαρακτήρες και να περιέχει μόνο λατινικούς χαρακτήρες και τον χαρακτήρα της κάτω παύλας.
- Κατά την εγγραφή, το username του χρήστη θα πρέπει να αποτελείται από τρεις έως είκοσι χαρακτήρες και να περιέχει μόνο λατινικούς χαρακτήρες, αριθμούς και τον χαρακτήρα της κάτω παύλας.
- Κατά την εγγραφή, το username του χρήστη θα πρέπει να αποτελείται από τρεις έως είκοσι χαρακτήρες και να περιέχει μόνο λατινικούς χαρακτήρες, αριθμούς και τον χαρακτήρα της κάτω παύλας.
- Κατά την εγγραφή, ο μυστικός κωδικός (password) του χρήστη θα πρέπει να αποτελείται από έξι έως είκοσι χαρακτήρες και να περιέχει μόνο λατινικούς χαρακτήρες, αριθμούς και τους χαρακτήρες !, @, #, \$, %, &, *, (,).
- Κατά την εγγραφή, το e-mail του χρήστη θα πρέπει να έχει έγκυρη μορφή διεύθυνσης ηλεκτρονικού ταχυδρομείου.
- Κατά την είσοδο του χρήστη στην εφαρμογή, το username του χρήστη θα πρέπει να αποτελείται από τρεις έως είκοσι χαρακτήρες και να περιέχει μόνο λατινικούς χαρακτήρες, αριθμούς και τον χαρακτήρα της κάτω παύλας.
- Κατά την είσοδο του χρήστη στην εφαρμογή, Κατά την εγγραφή, ο μυστικός κωδικός (password) του χρήστη θα πρέπει να αποτελείται από έξι έως είκοσι χαρακτήρες και να περιέχει μόνο λατινικούς χαρακτήρες, αριθμούς και τους χαρακτήρες !, @, #, \$, %, &, *, (,).

- Κατά την είσοδο του χρήστη στην εφαρμογή, το username του χρήστη θα πρέπει να είναι ήδη αποθηκευμένο στη βάση δεδομένων, αλλιώς να εμφανίζεται μήνυμα λάθους στο χρήστη.
- Κατά την είσοδο του χρήστη στην εφαρμογή, το password του χρήστη θα πρέπει να είναι ήδη αποθηκευμένο στη βάση δεδομένων, αλλιώς να εμφανίζεται μήνυμα λάθους στο χρήστη.
- Όλα τα μενού της εφαρμογής και οι φόρμες θα πρέπει να εμφανίζονται και να αποκρύπτονται με κινούμενο εφέ.
- Κάθε φορά όπου θα εκτελείται μια εργασία στο backend θα πρέπει να ειδοποιείται ο χρήστης με κατάλληλο εικονίδιο.
- Εάν κατά την διαδικασία αποστολής δεδομένων δεν έχει επιλεγεί κάποιο Social Media, τότε η διαδικασία δε θα επιτρέπεται να συνεχιστεί και θα πρέπει να παροτρύνεται ο χρήστης να επιλέξει κάποιο Social Media προκειμένου να συνεχιστεί η διαδικασία.
- Στη φόρμα συγκέντρωσης του περιεχομένου που θα αποσταλεί στα επιλεγμένα Social Media, εάν δεν έχει αποσταλεί κάποιο στοιχείο, τότε η διαδικασία θα πρέπει να σταματά και να παροτρύνεται ο χρήστης να εισάγει το απαραίτητο στοιχείο προκειμένου να συνεχιστεί η διαδικασία.

5.5 - Μελέτη τεχνολογιών - Τεχνολογίες που χρησιμοποιήθηκαν

Apache HTTP Server

Ο Apache HTTP Server είναι ένας διακομιστής ιστοσελίδων ο οποίος διαδραμάτισε προεξέχοντα ρόλο στην αρχική ανάπτυξη του παγκόσμιου ιστού. Μέχρι σήμερα, είναι ο πιο διαδεδομένος διακομιστής με περισσότερα από 100 εκατομμύρια ιστοσελίδες να βασίζονται τη λειτουργία τους σε αυτόν. Τα χαρακτηριστικά που διαθέτει του επιτρέπουν να ανταγωνίζεται τους υπόλοιπους διαθέσιμους διακομιστές τόσο σε όρους παρεχόμενων λειτουργιών όσο και σε απόδοση. Ο Apache διατίθεται για μία μεγάλη πληθώρα λειτουργικών συστημάτων, όπως Windows, Linux, FreeBSD, κλπ.. Παρόλα αυτά στις περισσότερες των περιπτώσεων προτιμάται η λειτουργία του σε συνδυασμό με το λειτουργικό σύστημα Linux. Αποτελεί ανοικτό λογισμικό υπό την άδεια Apache License και η ανάπτυξη και διαχείρισή του πραγματοποιείται από μία ανοικτή κοινότητα προγραμματιστών υπό την αιγίδα του Apache Software Foundation. Ο Apache HTTP Server υποστηρίζει ένα ευρύ σύνολο χαρακτηριστικών, πολλά από τα οποία υλοποιούνται ως ξεχωριστά modules τα οποία και επεκτείνουν το βασικό σύνολο λειτουργιών του διακομιστή. Ενδεικτικά αναφέρονται οι εξής δυνατότητες:

- Υποστήριξη γλωσσών προγραμματισμού στην πλευρά του διακομιστή, όπως είναι PHP, Perl, Python, κλπ.
- Δυνατότητα ταυτοποίησης χρηστών μέσω διάφορων διαδεδομένων συστημάτων ταυτοποίησης.
- Τροποποίηση των διευθύνσεων ώστε να είναι φιλικότερες στον χρήστη και στις μηχανές αναζήτησης.
- Συμπίεση των αποστελλόμενων δεδομένων για βελτίωση των επιδόσεων του διακομιστή.
- Υποστήριξη ασφαλούς μεταφοράς της πληροφορίας μέσω του πρωτοκόλλου TLS.

Αν και ο βασικός σχεδιαστικός στόχος του Apache δεν ήταν η επίτευξη των μέγιστων επιδόσεων, παρόλα αυτά εξακολουθεί να είναι ένας από τους πιο γρήγορους διακομιστές που κυκλοφορούν σήμερα. Έχει τη δυνατότητα μέσω του αντίστοιχου module να υποστηρίζει διαφορετικές αρχιτεκτονικές υπολογιστικών συστημάτων με αποτέλεσμα να εκμεταλλεύεται κατά το μέγιστο δυνατό τρόπο την εκάστοτε υπολογιστική υποδομή.

Έχοντας υπόψη το σύνολο των παραπάνω αναφερόμενων πληροφοριών για τον Apache HTTP Server πραγματοποιήθηκε η επιλογή του ως διακομιστή για τη δικτυακή εφαρμογή **SocialCampaign** που αναπτύχθηκε στα πλαίσια της παρούσης διπλωματικής εργασίας.

MySQL

Η MySQL είναι ένα διαδεδомένο σύστημα σχεσιακής βάσης δεδομένων που αριθμεί σήμερα περισσότερες από 6 εκατομμύρια εγκαταστάσεις παγκοσμίως. Λειτουργεί ως server και, επομένως, μπορεί να εξυπηρετήσει την πρόσβαση πολλαπλών χρηστών σε πολλαπλές βάσεις δεδομένων ταυτόχρονα. Είναι ανοικτό λογισμικό μιας και διανέμεται υπό την άδεια GNU General Public License. Ανήκει στην κερδοσκοπική εταιρία MySQL AB, η οποία πλέον είναι θυγατρική εταιρία της Sun Microsystems, η οποία με τη σειρά της βρίσκεται στη διαδικασία απορρόφησης από την Oracle Corporation. Η MySQL χρησιμοποιείται πολύ συχνά σε έργα ανοικτού λογισμικού και, συνήθως, συνδυάζεται με το λειτουργικό σύστημα Linux και το διακομιστή ιστοσελίδων Apache HTTP Server. Ένας σημαντικός αριθμός από μεγάλης κλίμακας διαδικτυακών προϊόντων χρησιμοποιεί τη MySQL, όπως Wikipedia, Google και Facebook. Η MySQL είναι σχεδιασμένη ώστε να υποστηρίζει ένα μεγάλο αριθμό από λειτουργικά συστήματα περιλαμβανομένων των Linux, Windows, NetBSD, OpenBSD, κλπ.. Επιπλέον, διατίθενται βιβλιοθήκες πρόσβασης στο σύστημα της βάσης δεδομένων για τις περισσότερες γνωστές γλώσσες προγραμματισμού. Τα χαρακτηριστικά που διαθέτει καλύπτουν ένα ευρύ φάσμα εφαρμογών. Ενδεικτικά παρατίθενται τα εξής:

- Υποστήριξη του ANSI SQL 99.
- Υποστήριξη για stored procedures, cursors και triggers.
- Υποστήριξη για updatable views.
- Παρέχει πολλαπλές ανεξάρτητες μηχανές αποθήκευσης δεδομένων, όπως MyISAM, InnoDB, κλπ.. Η κάθε μηχανή είναι βελτιστοποιημένη για συγκεκριμένη χρήση.
- Υποστήριξη Full-text indexing.

Με βάση τα παραπάνω, επιλέχθηκε να χρησιμοποιηθεί η MySQL ως το σύστημα σχεσιακής βάσης δεδομένων το οποίο ανέλαβε το ρόλο της διαχείρισης των δεδομένων της εφαρμογής διαχείρισης κειμένου.

PHP

Η PHP είναι μία ευρέως διαδεδομένη scripting γλώσσα προγραμματισμού, η οποία ειδικεύεται και χρησιμοποιείται κατά κόρον στην ανάπτυξη εφαρμογών για τον παγκόσμιο ιστό. Είναι συμβατή με μία ευρεία συλλογή από λειτουργικά συστήματα και διακομιστές ιστοσελίδων με αποτέλεσμα να χρησιμοποιείται σε περισσότερες από 20 εκατομμύρια ιστοσελίδες παγκοσμίως. Αποτελεί ανοικτό λογισμικό και διανέμεται υπό την άδεια PHP License. Αξίζει να σημειωθεί ότι, αν και ο αρχικός στόχος της γλώσσας ήταν αποκλειστικά η υποστήριξη δημιουργίας δυναμικών ιστοσελίδων, σήμερα μπορεί να χρησιμοποιηθεί για τη συγγραφή ανεξάρτητων εφαρμογών είτε γραμμής εντολών είτε γραφικού interface.

Από την έκδοση 4 και έπειτα, τα αρχεία πηγαίου κώδικα μεταγλωττίζονται αυτόματα σε bytecode και εκτελούνται από την αφηρημένη μηχανή Zend Engine με αποτέλεσμα αυξημένες επιδόσεις σε σχέσεις με τις προγενέστερες εκδόσεις. Παράλληλα, υπάρχουν λύσεις σε επίπεδο λογισμικού που αξιοποιούν ένα σύνολο από τεχνικές για να επιτύχουν ακόμη μεγαλύτερες επιδόσεις όταν αυτό κρίνεται αναγκαίο για την εύρυθμη εξυπηρέτηση του συνόλου των χρηστών της δικτυακής εφαρμογής. Άλλο ένα σημαντικό χαρακτηριστικό της γλώσσας προγραμματισμού PHP είναι ότι από την έκδοση 5 και έπειτα υποστηρίζει πλήρως το αντικειμενοστραφές μοντέλο προγραμματισμού με αποτέλεσμα ο προγραμματιστής να έχει την ευχέρεια να αξιοποιήσει σύγχρονες τεχνικές ανάπτυξης λογισμικού για την συγγραφή της εκάστοτε εφαρμογής.

Στη βάση όλων των παραπάνω χαρακτηριστικών, η PHP επιλέχθηκε ως η γλώσσα ανάπτυξης του συστήματος διαχείρισης κειμένου της παρούσας διπλωματικής εργασίας. Επιπλέον, κρίθηκε ως θετικό στοιχείο η δυνατότητα άριστης συνεργασίας τόσο με τον Apache HTTP Server όσο και με το σύστημα σχεσιακής βάσης δεδομένων MySQL.

JavaScript

Η JavaScript είναι γλώσσα προγραμματισμού η οποία έχει σαν σκοπό την παραγωγή δυναμικού περιεχομένου και την εκτέλεση κώδικα στην πλευρά του πελάτη (client-side) σε ιστοσελίδες.

Η αρχική έκδοση της JavaScript βασίστηκε στη σύνταξη στη γλώσσα προγραμματισμού C, αν και έχει εξελιχθεί, ενσωματώνοντας πια χαρακτηριστικά από νεότερες γλώσσες. Αρχικά χρησιμοποιήθηκε για προγραμματισμό από την πλευρά του πελάτη (client), που ήταν ο φυλλομετρητής (browser) του χρήστη, και χαρακτηρίστηκε σαν *client-side* γλώσσα προγραμματισμού. Αυτό σημαίνει ότι η επεξεργασία του κώδικα JavaScript και η παραγωγή του τελικού περιεχομένου HTML δεν πραγματοποιείται στο διακομιστή, αλλά στο πρόγραμμα περιήγησης των επισκεπτών, ενώ μπορεί να ενσωματωθεί σε στατικές σελίδες HTML. Αντίθετα, άλλες γλώσσες όπως η PHP εκτελούνται στο διακομιστή (*server-side* γλώσσες προγραμματισμού).

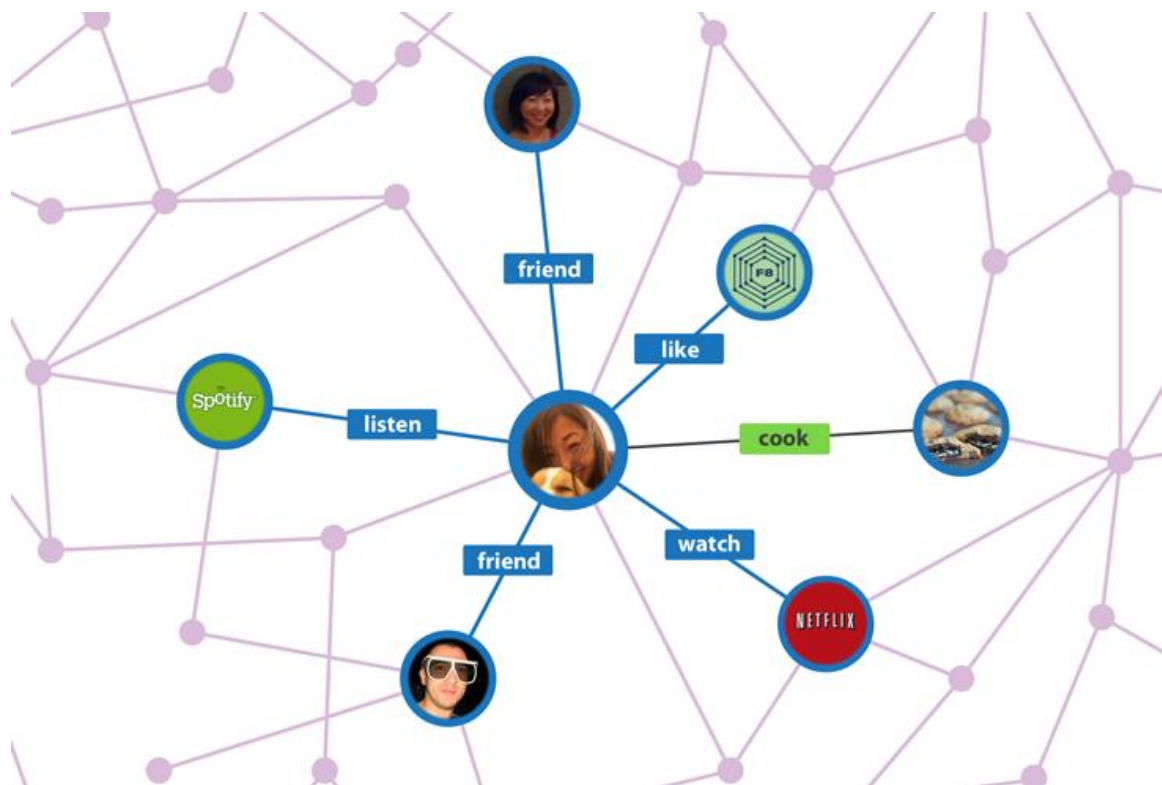
Κεφάλαιο 6^ο - Υλοποίηση Εφαρμογής Social Campaign

Μετά από μελέτη και εξοικείωση με τα APIs των Social Media με τα οποία επιλέξαμε να ασχοληθούμε και των web τεχνολογιών που αναφέραμε παραπάνω, ξεκίνησε η υλοποίηση της τελικής εφαρμογής, η οποία αποτελούσε το στόχο της διπλωματικής εργασίας. Οι λειτουργίες της εφαρμογής, χωρίζονται σε δύο κύριες λειτουργίες, αυτή του Publishing περιεχομένου και αυτή του Retrieving περιεχομένου, οι οποίες αναφέρονται αναλυτικά παρακάτω. Τέλος, έπειτα από το retrieving της πληροφορίας πραγματοποιείται το visualization μέρος της εφαρμογής, το οποίο παρουσιάζει τη συλλεγμένη πληροφορία οπτικά με χρήση διαγραμμάτων.

6.1. - Publishing

6.1.1 Facebook

Η βασική ιδέα γύρω από το Facebook είναι ότι όλα τα αντικείμενα τα οποία βρίσκονται σε αυτό (όπως χρήστες, εφαρμογές κ.α.) ανήκουν σε ένα μεγάλο γράφο. Τα αντικείμενα αυτά, συνδέονται μεταξύ τους με διάφορες δράσεις (actions). Η παρακάτω εικόνα περιγράφει το γράφο αυτό με πιο κατανοητό τρόπο.



6.1. - Ο γράφος του Facebook

Για παράδειγμα, το αντικείμενο (object) John έχει φίλο (friend) το αντικείμενο Mary.

Στην παραπάνω πρόταση, τα δύο αντικείμενα John και Mary, συνδέονται μεταξύ τους στο γράφο του Facebook με μια δράση (action) η οποία είναι η δράση friend.

Έτσι λοιπόν μπορεί κανείς, με χρήση actions να εισάγει στον γράφο του Facebook διάφορα objects. Η διαδικασία γίνεται πραγματοποιώντας HTTP requests στον server του Facebook και εκείνος με τη σειρά του απαντάει κατάλληλα στα requests τα οποία πραγματοποιούμε.

Παρακάτω φαίνεται πως εισάγονται προγραμματιστικά στο Facebook κείμενο μικρού μήκους (status update), κείμενο μεγάλου μήκους (note), φωτογραφία (photo) και βίντεο (video).

Αρχικά, εφόσον έχει επιλεγεί το Facebook για να δημοσιευθεί σε αυτό περιεχόμενο θα πρέπει ο χρήστης να κάνει είσοδο στην εφαρμογή μέσω του λογαριασμού του στο Facebook. Η διαδικασία ξεκινάει από το αρχείο createCampaign.php το οποίο εμφανίζει μια HTML φόρμα η οποία έχει τροποποιηθεί χρησιμοποιώντας JavaScript και πιο συγκεκριμένα τη βιβλιοθήκη jQuery, και εμφανίζει τα διαθέσιμα Social Media στα οποία ο εγγεγραμμένος χρήστης μπορεί να προωθήσει την καμπάνια του.

```
<form method="post" id="selectSM" action="selection.php" >
<div><label for="check1" >Facebook_____</label><input id =
"check1" class = "right" type="checkbox" name="facebook" value="yes" /></div>
<br/>
<div><label for="check2">Twitter_____</label><input id =
"check2" class = "right" type="checkbox" name="twitter" value="yes" /></div>
<br/>
<div><label for="check3">Linkedin_____</label><input id =
"check3" class = "right" type="checkbox" name="linkedin" value="yes" /></div>
<br/>
```

```

<div><label for="check4">Youtube_____</label><input id =
"check4" class = "right" type="checkbox" name="youtube" value="yes" /></div>

<br/>

<div><label for="check5">

Blogger_____</label><input id
= "check5" class = "right" type="checkbox" name="blogger" value="yes" /></div>

<br/>

<div><label for="check6">

Picasa_____</label><input id =
"check6" class = "right" type="checkbox" name="picasa" value="yes" /></div>

<br/>

<div><input id="choosebtn" value="Choose SM" type="submit" /></div>

</form>

```

Η φόρμα από την οποία επιλέγει ο χρήστης τα Social Media στα οποία θα δημοσιευθεί η καμπάνια του.

Η παραπάνω φόρμα αποστέλλεται στον server, και πιο συγκεκριμένα στο αρχείο selection.php, και γίνεται μια καταχώρηση στη βάση δεδομένων με μια νέα καμπάνια η οποία θα δημοσιευθεί στα Social Media τα οποία έχουν επιλεγεί.

```

if( isset( $_POST['facebook'] )
$facebook = $_POST['facebook'];
else
$facebook = 'no';

if( isset( $_POST['twitter'] ) )
$twitter = $_POST['twitter'];
else
$twitter = 'no';

```



```
if( isset( $_POST['linkedin'] )
$linkedin = $_POST['linkedin'];

else
$linkedin = 'no';

if( isset( $_POST['youtube'] )
$youtube = $_POST['youtube'];

else
$youtube = 'no';

if( isset( $_POST['blogger'] )
$blogger = $_POST['blogger'];

else
$blogger = 'no';

if( isset( $_POST['picassa'] )
$picassa = $_POST['picassa'];

else
$picassa = 'no';

$finaluseridee = $_SESSION['userid'];

$current = 1;

$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
or die('Could not connect to database: ' . mysql_error());

mysql_select_db('ptyxiaki_vasi') or die('Could not select database');

mysql_query("INSERT INTO campaigns (userid, facebook, twitter, linkedin, youtube,
blogger, picasa, current)
VALUES ( '$finaluseridee', '$facebook', '$twitter', '$linkedin', '$youtube', '$blogger',
'$picassa', '$current')");
```

Έλεγχος της φόρμας και δημιουργία μια νέας εγγραφής στη βάση δεδομένων με τη νέα καμπάνια που δημιουργείται αυτή τη στιγμή.

		id	userid	facebook	twitter	linkedin	youtube	blogger	picasa	campaign_name	current
		690	50	yes	yes	yes	yes	yes	yes	Δοκιμαστικό - συλλογή δεδομένων	0

Εγγραφή νέας καμπάνιας στη βάση, η οποία θα δημοσιευθεί σε όλα τα Social Media τα οποία υποστηρίζει η εφαρμογή.

Αφού γίνει η εγγραφή της καμπάνιας στη βάση δεδομένων, ο χρήστης οδηγείται στο αρχείο auth.php το οποίο ευθύνεται για την αυθεντικοποίηση του χρήστη μέσω του λογαριασμού στο καθένα από τα Social Media που έχουν επιλεγεί για την καμπάνια.

Εδώ, για το Facebook, τα στοιχεία του λογαριασμού του χρήστη είναι γνωστά μόνο σε αυτόν για αυτό και η αυθεντικοποίηση μέσω Facebook γίνεται με redirect από την εφαρμογή στο Facebook και από το Facebook στην εφαρμογή.

Αρχικά η εφαρμογή διαβάζει τη βάση και στη συνέχεια ελέγχει εάν ο χρήστης έχει επιλέξει το Facebook ώστε να δημοσιευθεί σε αυτό περιεχόμενο της καμπάνιας του.

```

$link = mysql_connect('localhost', 'ptychiaki_vasi', '34fg90j')
or die('Could not connect to database: ' . mysql_error());

mysql_select_db('ptychiaki_vasi') or die('Could not select database');

$checkSelectedMedia = mysql_query("SELECT facebook, twitter, linkedin, youtube,
blogger, picasa
FROM campaigns
WHERE id = '$current_campaign_id'
LIMIT 1;
");

mysql_close($link);

$row = mysql_fetch_array($checkSelectedMedia);

$facebookChecked = $row['facebook'];

```

Ανάγνωση της βάσης δεδομένων και καταχώρηση της τιμής yes ή no στη μεταβλητή facebookChecked. Εάν η μεταβλητή facebookChecked έχει την τιμή yes τότε

περιεχόμενο της καμπάνιας θα δημοσιευθεί στο Facebook. Εάν η τιμή της μεταβλητής `facebookChecked` είναι `no` τότε περιεχόμενο της καμπάνιας δε θα δημοσιευθεί στο Facebook.

```
if($facebookChecked == 'yes'){
require '../SMauthentications/facebookauth.php';
}
```

Εάν η μεταβλητή `facebookChecked` έχει την τιμή `yes`, ο χρήστης καλείται να συνδεθεί στην εφαρμογή μέσω το λογαριασμού του στο Facebook

Το αρχείο `facebookauth.php` είναι υπεύθυνο για την αυθεντικοποίηση του χρήστη στην εφαρμογή μέσω Facebook, προκειμένου να αποστείλει στο Facebook το περιεχόμενο που επιθυμεί μέσω της εφαρμογής. Καλείται όταν ο χρήστης επιλέξει να δημοσιεύσει περιεχόμενο στο Facebook και παρόλο του ότι είναι πολύ μικρό σε έκταση, παίζει καθοριστική σημασία γιατί είναι αυτό που συνδέει την εφαρμογή με το PHP SDK του Facebook.

Το PHP SDK του Facebook αποτελείται από μια μεγάλη κλάση, την κλάση `Facebook.class.php`.

Για να γίνει το οποιοδήποτε API call, θα πρέπει να δημιουργηθεί ένα αντικείμενο της κλάσης αυτής το οποίο στη συνάρτηση δημιουργίας του δέχεται δύο ορίσματα, το `id` του application το οποίο υλοποιούμε και ένα μυστικό κωδικό. Το απαραίτητο αυτό αντικείμενο, δημιουργείται στο αρχείο `facebookauth.php`.

```
require 'facebook_required_files/facebook.php';
$facebook = new Facebook(array(
    'appId' => '173278596073417',
    'secret'=>'f5fc60352cb1ef235019b1b84be36615',
));
```

Δημιουργία ενός αντικειμένου της κλάσης Facebook

Στη συνέχεια, η εφαρμογή ελέγχει εάν κάποιος χρήστης είναι ήδη συνδεδεμένος στο Facebook ή όχι. Εάν είναι, τότε απλά επιστρέφει στην εφαρμογή πατώντας στο link που του εμφανίζεται. Εάν δεν είναι, τότε θα πρέπει να συνδεθεί με το λογαριασμό του και να συμφωνήσει εάν η εφαρμογή θα πρέπει να έχει πρόσβαση σε

κάποια στοιχεία του ή όχι. Για να έχει η εφαρμογή πρόσβαση στα στοιχεία αυτά, και συγκεκριμένα στην άδεια να δημοσιεύει περιεχόμενο μικρού κειμένου στον τοίχο του (status update), να δημοσιεύει περιεχόμενο φωτογραφίας (photo upload), περιεχόμενο βίντεο (video upload) και περιεχόμενο μεγάλου κειμένου (note). Πιο συγκεκριμένα, για καθένα από τα στοιχεία τις παραπάνω πρότασης η εφαρμογή ζητάει από το χρήστη τις άδειες publish_stream, user_photos, user_videos και user_notes ώστε να εκτελέσει τις διαδικασίες που χρειάζονται για να δημοσιεύσει περιεχόμενο στο λογαριασμό του συνδεδεμένου χρήστη. Τις άδειες αυτές τις εμφανίζει στο χρήστη μέσω ενός λινκ με το οποίο θα γίνει το login μέσω Facebook.

```
$loginUrl = $facebook-
>getLoginUrl(array('scope'=>'publish_stream,user_photos,user_videos,user_notes'));
```

Οι άδειες μέσω του login url

Αφού ολοκληρωθεί η παραπάνω διαδικασία, ο χρήστης θα πρέπει να εισάγει τα στοιχεία τα οποία θα δημοσιευθούν στο Facebook. Αυτό γίνεται μέσω μιας HTML φόρμας η οποία έχει διαμορφωθεί κατάλληλα χρησιμοποιώντας CSS και ελέγχεται πριν αποσταλεί το οτιδήποτε στο server με JavaScript. Η αποστολή της φόρμας γίνεται με την JavaScript τεχνική AJAX. Η εφαρμογή ανατρέχει στη βάση δεδομένων, εντοπίζει την καμπάνια η οποία διαμορφώνεται αυτή τη στιγμή και διαμορφώνει δυναμικά τη φόρμα ανάλογα με τις επιλογές του χρήστη.

```
$current_campaign_id = $_SESSION['curent_campaign_id'];

$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
or die('Could not connect to database: ' . mysql_error());

mysql_select_db('ptyxiaki_vasi') or die('Could not select database');

$checkSelectedMedia = mysql_query("SELECT facebook, twitter,
linkedin, youtube, blogger, picasa
FROM campaigns
WHERE id = '$current_campaign_id'
LIMIT 1;
");

mysql_close($link);
```

```
$row = mysql_fetch_array($checkSelectedMedia);
$facebookfinal = $row['facebook'];
```

Η εφαρμογή ανατρέχει στη βάση και εντοπίζει την καμπάνια η οποία διαμορφώνεται την παρούσα χρονική στιγμή.

Το κομμάτι τις φόρμας το οποίο αφορά το Facebook φαίνεται παρακάτω

```
$publishform = '<form id = "finalforme" enctype = "multipart/form-data"
method="post" action="posting.php">
<div><label for="Campaignname">Campaign name: </label><br/>
<input type = "text" name="Campaignname" id="Campaignname"/>
<br/>';
if($facebookfinal == 'yes'){
$publishform .= '<div><label for="shortmessage">Short message: </label><br/>
<textarea rows="2" cols="40" name="shortmessage"
id="shortmessage"></textarea></div>
<br/>
<div><label for="longmessage">Long message: </label><br/>
<textarea rows="10" cols="40" name="longmessage"
id="longmessage"></textarea></div>
<br/>
<div class="upload">
<input type="file" name="photo" id="photo"/>
<span id="upload_text">Photo</span><br/>
</div>
<br/>
<div class="upload">
<input type="file" name="video" id="video"/>
```

```

<span id="upload_text2">Video</span><br/>
</div>
<br/>';//echo
}

```

Το κομμάτι τις φόρμας το οποίο αφορά το Facebook

Στη συνέχεια, αφού η φόρμα έχει συμπληρωθεί από το χρήστη, τα δεδομένα μεταφέρονται στο server και στο αρχείο posting.php. Το αρχείο posting.php είναι ένα από τα δύο βασικά αρχεία της εφαρμογής το οποίο αναλαμβάνει τη δημοσίευση περιεχομένου στα Social media με τα οποία η εφαρμογή αλληλεπιδρά. Παρακάτω, παρουσιάζεται το κομμάτι του κώδικα το οποίο αποστέλλει περιεχόμενο στο Facebook.

```
$facebookShortMessage = $_POST['shortmessage'];
```

```
if ($user) {
```

```
    try {
```

```
        // Proceed knowing you have a logged in user who's
        // authenticated.
```

```
        $user_post = $facebook->api('/me/feed', 'Post', array(
            'message' => $facebookShortMessage
        ));
```

```
    } catch (FacebookApiException $e) {
```

```
        echo $e ->getMessage();
```

```
        $user = null;
```

```
    }
```

```
}
```

```
$status_update_id = $user_post['id'];//get immediately the status update that was
created
```

```
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
```

```

or die('Could not connect to database: ' . mysql_error());

mysql_select_db('ptyxiaki_vasi') or die('Could not select database');

mysql_query("INSERT INTO facebook_posting (campaign_id, id_facebook,
type_facebook)
VALUES ('$current_campaign_id', '$status_update_id', 'status_update'");

mysql_close($link);

```

Δημιουργία ενός νέου status update στο προφίλ του συνδεδεμένου στο Facebook χρήστη και απόθήκευσή του id του status update που μόλις ανέβηκε, στη βάση δεδομένων

```

//photo upload

if ($user) {

    try {

        $args = array('message' => $CampaignName);

        $args['image'] = '@' . realpath($photoPath);

        $photo = $facebook->api('/me/photos', 'post', $args);

        $photo_id = $photo['id'];

    } catch (FacebookApiException $e) {

        echo $e ->getMessage();

        $user = null;

    }

}

$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
or die('Could not connect to database: ' . mysql_error());

mysql_select_db('ptyxiaki_vasi') or die('Could not select database');

mysql_query("INSERT INTO facebook_posting (campaign_id, id_facebook,
type_facebook)
VALUES ('$current_campaign_id', '$photo_id', 'photo'");

mysql_close($link);

```

Αποστολή μιας νέας φωτογραφίας στο προφίλ του συνδεδεμένου στο Facebook χρήστη και αποθήκευσή του id της φωτογραφίας που μόλις ανέβηκε, στη βάση δεδομένων

```
if ($user) {  
    try {  
        $video_path = $videoPath;  
        $video_args = array('description' => $CampaignName);  
        $video_args['video'] = '@' . realpath($video_path);  
        $create_video = $facebook->api('/me/videos', 'post', $video_args);//epistrefei array  
  
        $create_video_id = $create_video['id'];  
        unlink($video_path);  
    } catch (FacebookApiException $e) {  
        echo $e ->getMessage();  
        $user = null;  
    }  
}  
  
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')  
or die('Could not connect to database: ' . mysql_error());  
  
mysql_select_db('ptyxiaki_vasi') or die('Could not select database');  
  
mysql_query("INSERT INTO facebook_posting (campaign_id, id_facebook,  
type_facebook)  
VALUES ('$current_campaign_id', '$create_video_id', 'video')");  
  
mysql_close($link);
```

Αποστολή ενός νέου βίντεο στο προφίλ του συνδεδεμένου στο Facebook χρήστη και αποθήκευση του id του βίντεο που μόλις ανέβηκε, στη βάση δεδομένων


```
//note upload
$facebookLongMessage = $_POST['longmessage'];
if ($user) {
    try {
        $create_note = $facebook->api('/me/notes', 'post', array(
            'message' => $facebookLongMessage,
            'subject' => $CampaignName
        ));//epistrefei array

        $note_id = $create_note['id'];

    } catch (FacebookApiException $e) {
        echo $e ->getMessage();
        $user = null;
    }
}

$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
or die('Could not connect to database: ' . mysql_error());

mysql_select_db('ptyxiaki_vasi') or die('Could not select database');

mysql_query("INSERT INTO facebook_posting (campaign_id, id_facebook,
type_facebook)
VALUES ('$current_campaign_id', '$note_id', 'note')");

mysql_close($link);
```

Αποστολή ενός νέου note (κειμένου μεγάλης έκτασης) στο προφίλ του συνδεδεμένου στο Facebook χρήστη και αποθήκευση του id του note που μόλις ανέβηκε, στη βάση δεδομένων

Το Facebook δίνει στην εφαρμογή τη δυνατότητα να αποστέλλει και τους τέσσερις τύπους δεδομένων (κείμενο μικρού μεγέθους, κείμενο μεγάλου μεγέθους, βίντεο, φωτογραφία).

6.1.2. - Twitter

Το API του Twitter υλοποιεί μια REST προσέγγιση, γεγονός που το κάνει εύχρηστο και εύκολο στο να υλοποιηθούν εφαρμογές οι οποίες να ενσωματώνουν λειτουργίες του Twitter.

Παρακάτω φαίνεται πως εισάγεται προγραμματιστικά στο Twitter κείμενο μικρού μήκους (tweet).

Αρχικά, εφόσον έχει επιλεγεί το Twitter για να δημοσιευθεί σε αυτό περιεχόμενο θα πρέπει ο χρήστης να κάνει είσοδο στην εφαρμογή μέσω του λογαριασμού του στο Twitter. Η διαδικασία ξεκινάει από το αρχείο createCampaign.php το οποίο εμφανίζει μια HTML φόρμα η οποία έχει τροποποιηθεί χρησιμοποιώντας javascript και πιο συγκεκριμένα τη βιβλιοθήκη jQuery, και εμφανίζει τα διαθέσιμα Social Media στα οποία ο εγγεγραμμένος χρήστης μπορεί να προωθήσει την καμπάνια του.

```
<form method="post" id="selectSM" action="selection.php" >
<div><label for="check1" >Facebook_____</label><input id =
"check1" class = "right" type="checkbox" name="facebook" value="yes" /></div>
<br/>
<div><label for="check2">Twitter_____</label><input id =
"check2" class = "right" type="checkbox" name="twitter" value="yes" /></div>
<br/>
<div><label for="check3">Linkedin_____</label><input id =
"check3" class = "right" type="checkbox" name="linkedin" value="yes" /></div>
<br/>
<div><label for="check4">Youtube_____</label><input id =
"check4" class = "right" type="checkbox" name="youtube" value="yes" /></div>
```

```

<br/>
<div><label for="check5">
Blogger_____</label><input id
= "check5" class = "right" type="checkbox" name="blogger" value="yes" /></div>
<br/>
<div><label for="check6">
Picassa_____</label><input id =
"check6" class = "right" type="checkbox" name="picassa" value="yes" /></div>
<br/>
<div><input id="choosebtn" value="Choose SM" type="submit" /></div>
</form>

```

Η φόρμα από την οποία επιλέγει ο χρήστης τα Social Media στα οποία θα δημοσιευθεί η καμπάνια του.

Η παραπάνω φόρμα αποστέλλεται στον server, και πιο συγκεκριμένα στο αρχείο selection.php, και γίνεται μια καταχώρηση στη βάση δεδομένων με μια νέα καμπάνια η οποία θα δημοσιευθεί στα Social Media τα οποία έχουν επιλεγεί.

```

if( isset( $_POST['facebook'] )
$facebook = $_POST['facebook'];
else
$facebook = 'no';
if( isset( $_POST['twitter'] )
$twitter = $_POST['twitter'];
else
$twitter = 'no';
if( isset( $_POST['linkedin'] )

```

```
$linkedin = $_POST['linkedin'];  
  
else  
  
$linkedin = 'no';  
  
if( isset( $_POST['youtube'] ) )  
  
$youtube = $_POST['youtube'];  
  
else  
  
$youtube = 'no';  
  
if( isset( $_POST['blogger'] ) )  
  
$blogger = $_POST['blogger'];  
  
else  
  
$blogger = 'no';  
  
if( isset( $_POST['picassa'] ) )  
  
$picassa = $_POST['picassa'];  
  
else  
  
$picassa = 'no';  
  
$finaluseridee = $_SESSION['userid'];  
  
$current = 1;  
  
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')  
  
or die('Could not connect to database: ' . mysql_error());  
  
mysql_select_db('ptyxiaki_vasi') or die('Could not select database');  
  
mysql_query("INSERT INTO campaigns (userid, facebook, twitter, linkedin, youtube,  
blogger, picasa, current)  
  
VALUES ( '$finaluseridee', '$facebook', '$twitter', '$linkedin', '$youtube', '$blogger',  
'$picassa', '$current')");
```

Έλεγχος της φόρμας και δημιουργία μια νέας εγγραφής στη βάση δεδομένων με τη νέα καμπάνια που δημιουργείται αυτή τη στιγμή.

		id	userid	facebook	twitter	linkedin	youtube	blogger	picasa	campaign_name	current
		690	50	yes	yes	yes	yes	yes	yes	Δοκιμαστικό - συλλογή δεδομένων	0

Εγγραφή νέας καμπάνιας στη βάση, η οποία θα δημοσιευθεί σε όλα τα Social Media τα οποία υποστηρίζει η εφαρμογή.

Αφού γίνει η εγγραφή της καμπάνιας στη βάση δεδομένων, ο χρήστης οδηγείται στο αρχείο auth.php το οποίο ευθύνεται για την αυθεντικοποίηση του χρήστη μέσω του λογαριασμού στο καθένα από τα Social Media που έχουν επιλεγεί για την καμπάνια.

Εδώ, για το Twitter, τα στοιχεία του λογαριασμού του χρήστη είναι γνωστά μόνο σε αυτόν για αυτό και η αυθεντικοποίηση μέσω Twitter γίνεται με redirect από την εφαρμογή στο Twitter και από το Twitter στην εφαρμογή.

Αρχικά η εφαρμογή διαβάζει τη βάση και στη συνέχεια ελέγχει εάν ο χρήστης έχει επιλέξει το Twitter ώστε να δημοσιευθεί σε αυτό περιεχόμενο της καμπάνιας του.

```
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
or die('Could not connect to database: ' . mysql_error());
mysql_select_db('ptyxiaki_vasi') or die('Could not select database');
$checkSelectedMedia = mysql_query("SELECT facebook, twitter, linkedin, youtube,
blogger, picasa
FROM campaigns
WHERE id = '$current_campaign_id'
LIMIT 1;
");
mysql_close($link);
$row = mysql_fetch_array($checkSelectedMedia);
$twitterChecked = $row['twitter'];
```

Ανάγνωση της βάσης δεδομένων και καταχώρηση της τιμής yes ή no στη μεταβλητή twitter Checked. Εάν η μεταβλητή twitterChecked έχει την τιμή yes τότε περιεχόμενο

της καμπάνιας θα δημοσιευθεί στο Twitter. Εάν η τιμή της μεταβλητής `twitterChecked` είναι no τότε περιεχόμενο της καμπάνιας δε θα δημοσιευθεί στο Twitter.

```
if($twitterChecked == 'yes'){
require './SMauthentications/twitterauth.php';
}
```

Εάν η μεταβλητή `twitterChecked` έχει την τιμή `yes`, ο χρήστης καλείται να συνδεθεί στην εφαρμογή μέσω το λογαριασμού του στο Twitter

Το αρχείο `twitterauth.php` είναι υπεύθυνο για την αυθεντικοποίηση του χρήστη στην εφαρμογή μέσω Twitter, προκειμένου να αποστείλει στο Twitter το περιεχόμενο που επιθυμεί μέσω της εφαρμογής. Καλείται όταν ο χρήστης επιλέξει να δημοσιεύσει περιεχόμενο στο Twitter και παρόλο του ότι είναι πολύ μικρό σε έκταση, παίζει καθοριστική σημασία γιατί είναι αυτό που συνδέει την εφαρμογή με τη βιβλιοθήκη `twitteroauth`.

Η βιβλιοθήκη `twitteroauth` αποτελείται από μια μεγάλη κλάση, την κλάση `TwitterOAuth.class.php`.

Για να γίνει το οποιοδήποτε API call, θα πρέπει να δημιουργηθεί ένα αντικείμενο της κλάσης αυτής το οποίο στη συνάρτηση δημιουργίας του δέχεται τέσσερα ορίσματα, το `access token`, το `access token secret`, το `oauth token` και το `oauth token secret`. Το απαραίτητο αυτό αντικείμενο, δημιουργείται στο αρχείο `posting.php`.

```
require('./SMauthentications/twitter_required_files/twitteroauth/twitteroauth.php'
)
$connection = new TwitterOAuth('WkcWcGjPXDFQ6Z3Uz2QA' ,
'ssaq0aC36Tfi4UDTbJo42MTBTHXd6E2XatpqsCLoI', $access_token['oauth_token'],
$access_token['oauth_token_secret']);
```

Δημιουργία ενός αντικειμένου της κλάσης `twitteroauth`

Στη συνέχεια, η εφαρμογή ελέγχει εάν κάποιος χρήστης είναι ήδη συνδεδεμένος στο Twitter ή όχι. Και στις δύο περιπτώσεις, θα πρέπει ο χρήστης να εξουσιοδοτήσει την εφαρμογή, ώστε να μπορεί αυτή να έχει πρόσβαση στα tweets του, να παρακολουθεί ποιούς χρήστες του Twitter ακολουθεί ο συνδεδεμένος χρήστης αλλά και

ποιοί τον ακολουθούν, να ανανεώνει το προφίλ του χρήστη και να δημοσιεύει tweets για αυτόν.

Μόλις ο χρήστης δώσει εξουσιοδότηση στην εφαρμογή για αυτές τις ενέργειες, μεταφέρεται και πάλι στην εφαρμογή.

Αφού ολοκληρωθεί η παραπάνω διαδικασία, ο χρήστης θα πρέπει να εισάγει τα στοιχεία τα οποία θα δημοσιευθούν στο Twitter. Αυτό γίνεται μέσω μιας HTML φόρμας η οποία έχει διαμορφωθεί κατάλληλα χρησιμοποιώντας CSS και ελέγχεται πριν αποσταλεί το οτιδήποτε στο server με javascript. Η αποστολή της φόρμας γίνεται με την javascript τεχνική AJAX. Η εφαρμογή ανατρέχει στη βάση δεδομένων, εντοπίζει την καμπάνια η οποία διαμορφώνεται αυτή τη στιγμή και διαμορφώνει δυναμικά τη φόρμα ανάλογα με τις επιλογές του χρήστη.

```
$current_campaign_id = $_SESSION['curent_campaign_id'];
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
    or die('Could not connect to database: ' . mysql_error());
mysql_select_db('ptyxiaki_vasi') or die('Could not select database');
$checkSelectedMedia = mysql_query("SELECT facebook, twitter,
linkedin, youtube,          blogger, picasa
FROM campaigns
WHERE id = '$current_campaign_id'
LIMIT 1;
");
mysql_close($link);
$row = mysql_fetch_array($checkSelectedMedia);
$twitterfinal = $row['twitter'];
```

Η εφαρμογή ανατρέχει στη βάση και εντοπίζει την καμπάνια η οποία διαμορφώνεται την παρούσα χρονική στιγμή.

Το κομμάτι τις φόρμας το οποίο αφορά το Twitter φαίνεται παρακάτω

```
if($twitterfinal == 'yes' && ($facebookfinal == 'no' && $linkedinfinal == 'no')){
    $publishform .= '<div><label for="shortmessage">Short message: </label><br/>
    <textarea rows="2" cols="40" name="shortmessage"
    id="shortmessage"></textarea></div><br/>';//echo
    }
<br/>';//echo
}
```

Το κομμάτι τις φόρμας το οποίο αφορά το Twitter

Στη συνέχεια, αφού η φόρμα έχει συμπληρωθεί από το χρήστη, τα δεδομένα μεταφέρονται στο server και στο αρχείο posting.php. Το αρχείο posting.php είναι ένα από τα δύο βασικά αρχεία της εφαρμογής το οποίο αναλαμβάνει τη δημοσίευση περιεχομένου στα Social media με τα οποία η εφαρμογή αλληλεπιδρά. Παρακάτω, παρουσιάζεται το κομμάτι του κώδικα το οποίο αποστέλλει περιεχόμενο στο Twitter.

```
$twitterShortMessage = $_POST['shortmessage'];
$tweet = $connection->post('statuses/update', array('status' =>
    $twitterShortMessage));
$id_twitter = $tweet->id_str; // and finally i get the tweet's id
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
or die('Could not connect to database: ' . mysql_error());

mysql_select_db('ptyxiaki_vasi') or die('Could not select database');
mysql_query("INSERT INTO twitter_posting (campaign_id, id_twitter, type_twitter)
            VALUES ('$current_campaign_id',
            '$id_twitter', 'tweet')");
mysql_close($link);
```

Αποστολή κειμένου μικρού μήκους (tweet) στο twitter και αποθήκευση του id του στη βάση δεδομένων

6.1.3 - YouTube

Το API του Youtube υλοποιεί μια REST προσέγγιση, γεγονός που το κάνει εύχρηστο και εύκολο στο να υλοποιηθούν εφαρμογές οι οποίες να ενσωματώνουν λειτουργίες του.

Παρακάτω φαίνεται πως εισάγεται προγραμματιστικά στο Youtube βίντεο (video upload).

Αρχικά, εφόσον έχει επιλεγεί το Youtube για να δημοσιευθεί σε αυτό περιεχόμενο θα πρέπει ο χρήστης να κάνει είσοδο στην εφαρμογή μέσω του λογαριασμού του στο Google. Η διαδικασία ξεκινάει από το αρχείο createCampaign.php το οποίο εμφανίζει μια HTML φόρμα η οποία έχει τροποποιηθεί χρησιμοποιώντας javascript και πιο συγκεκριμένα τη βιβλιοθήκη jQuery, και εμφανίζει τα διαθέσιμα Social Media στα οποία ο εγγεγραμμένος χρήστης μπορεί να προωθήσει την καμπάνια του.

```
<form method="post" id="selectSM" action="selection.php" >

<div><label for="check1" >Facebook_____</label><input id =
"check1" class = "right" type="checkbox" name="facebook" value="yes" /></div>

<br/>

<div><label for="check2">Twitter_____</label><input id =
"check2" class = "right" type="checkbox" name="twitter" value="yes" /></div>

<br/>

<div><label for="check3">Linkedin_____</label><input id =
"check3" class = "right" type="checkbox" name="linkedin" value="yes" /></div>

<br/>

<div><label for="check4">Youtube_____</label><input id =
"check4" class = "right" type="checkbox" name="youtube" value="yes" /></div>

<br/>

<div><label for="check5">
```

```

Blogger_____</label><input id
= "check5" class = "right" type="checkbox" name="blogger" value="yes" /></div>

<br/>

<div><label for="check6">

Picassa_____</label><input id =
"check6" class = "right" type="checkbox" name="picassa" value="yes" /></div>

<br/>

<div><input id="choosebtn" value="Choose SM" type="submit" /></div>

</form>

```

Η φόρμα από την οποία επιλέγει ο χρήστης τα Social Media στα οποία θα δημοσιευθεί η καμπάνια του.

Η παραπάνω φόρμα αποστέλλεται στον server, και πιο συγκεκριμένα στο αρχείο selection.php, και γίνεται μια καταχώρηση στη βάση δεδομένων με μια νέα καμπάνια η οποία θα δημοσιευθεί στα Social Media τα οποία έχουν επιλεγεί.

```

if( isset( $_POST['facebook'] )
$facebook = $_POST['facebook'];

else
$facebook = 'no';

if( isset( $_POST['twitter'] ) )
$twitter = $_POST['twitter'];

else
$twitter = 'no';

if( isset( $_POST['linkedin'] ) )
$linkedin = $_POST['linkedin'];

else
$linkedin = 'no';

```

```
if( isset( $_POST['youtube'] )  
$youtube = $_POST['youtube'];  
else  
$youtube = 'no';  
if( isset( $_POST['blogger'] ) )  
$blogger = $_POST['blogger'];  
else  
$blogger = 'no';  
if( isset( $_POST['picassa'] ) )  
$picassa = $_POST['picassa'];  
else  
$picassa = 'no';  
$finaluseridee = $_SESSION['userid'];  
$current = 1;  
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')  
or die('Could not connect to database: ' . mysql_error());  
mysql_select_db('ptyxiaki_vasi') or die('Could not select database');  
mysql_query("INSERT INTO campaigns (userid, facebook, twitter, linkedin, youtube,  
blogger, picasa, current)  
VALUES ( '$finaluseridee', '$facebook', '$twitter', '$linkedin', '$youtube', '$blogger',  
'$picassa', '$current')");
```

Έλεγχος της φόρμας και δημιουργία μια νέας εγγραφής στη βάση δεδομένων με τη νέα καμπάνια που δημιουργείται αυτή τη στιγμή.

	id	userid	facebook	twitter	linkedin	youtube	blogger	picasa	campaign_name	current
	690	50	yes	yes	yes	yes	yes	yes	Δοκιμαστικό - συλλογή δεδομένων	0

Εγγραφή νέας καμπάνιας στη βάση, η οποία θα δημοσιευθεί σε όλα τα Social Media τα οποία υποστηρίζει η εφαρμογή.

Αφού γίνει η εγγραφή της καμπάνιας στη βάση δεδομένων, ο χρήστης οδηγείται στο αρχείο auth.php το οποίο ευθύνεται για την αυθεντικοποίηση του χρήστη μέσω του λογαριασμού στο καθένα από τα Social Media που έχουν επιλεγεί για την καμπάνια.

Εδώ, για το Youtube, τα στοιχεία του λογαριασμού του χρήστη είναι γνωστά μόνο σε αυτόν για αυτό και η αυθεντικοποίηση μέσω Google γίνεται με redirect από την εφαρμογή στο Google accounts και από το Google accounts στην εφαρμογή.

Αρχικά η εφαρμογή διαβάζει τη βάση και στη συνέχεια ελέγχει εάν ο χρήστης έχει επιλέξει το Youtube ώστε να δημοσιευθεί σε αυτό περιεχόμενο της καμπάνιας του.

```
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
or die('Could not connect to database: ' . mysql_error());

mysql_select_db('ptyxiaki_vasi') or die('Could not select database');

$checkSelectedMedia = mysql_query("SELECT facebook, twitter, linkedin, youtube,
blogger, picasa
FROM campaigns
WHERE id = '$current_campaign_id'
LIMIT 1;
");

mysql_close($link);

$row = mysql_fetch_array($checkSelectedMedia);

$youtubeChecked = $row['youtube'];
```

Ανάγνωση της βάσης δεδομένων και καταχώρηση της τιμής yes ή no στη μεταβλητή youtubeChecked. Εάν η μεταβλητή youtubeChecked έχει την τιμή yes τότε περιεχόμενο

της καμπάνιας θα δημοσιευθεί στο Youtube. Εάν η τιμή της μεταβλητής `youtubeChecked` είναι no τότε περιεχόμενο της καμπάνιας δε θα δημοσιευθεί στο Youtube.

```
if($youtubeChecked == 'yes'){
require '../SMauthentications/googleauth.php';
}
```

Εάν η μεταβλητή `youtubeChecked` έχει την τιμή `yes`, ο χρήστης καλείται να συνδεθεί στην εφαρμογή μέσω το λογαριασμού του στο Google

Το αρχείο `googleauth.php` είναι υπεύθυνο για την αυθεντικοποίηση του χρήστη στην εφαρμογή μέσω Google. Περιέχει κώδικα που είναι υπεύθυνος και για την αυθεντικοποίηση μέσω Google, του χρήστη που δημιουργεί τη δεδομένη χρονική στιγμή καμπάνια και αφορά την αυθεντικοποίηση και για τα τρία προϊόντα της Google (youtube, Blogger, Picasa) με τα οποία αλληλεπιδρά η εφαρμογή η οποία αναπτύχθηκε.

Η βιβλιοθήκη Youtube PHP client API κάνει χρήση μιας μεγάλης βιβλιοθήκης, της βιβλιοθήκης Zend GData. Για να γίνει το οποιοδήποτε API call, θα πρέπει να δημιουργηθεί ένα αντικείμενο της κλασης Zend Gdata YouTube το οποίο στη συνάρτηση δημιουργίας του δέχεται τέσσερα ορίσματα, το `httpClient`, το `applicationId`, το `clientId`, και το `developerKey`. Το απαραίτητο αυτό αντικείμενο, δημιουργείται στο αρχείο `posting.php`.

```
$developerKey = 'A139si5hMq_qXwym1kmm9WkxWNsKtM8AJ3WxIz-lyB3ej8zrTk01QPZ4OUOTPMnlcmXdPyk6c0XUIxOtXR0W79-ibBXpUy-vg';

$applicationId = 'velas';

$clientId = 'Social Campaign';

Zend_Loader::loadClass('Zend_Gdata_YouTube');

Zend_Loader::loadClass('Zend_Gdata_AuthSub');

Zend_Loader::loadClass('Zend_Gdata_App_Exception');

$httpClient =
Zend_Gdata_AuthSub::getHttpClient($_SESSION['sessionToken']);
```

```
$ytf = new Zend_Gdata_YouTube($httpClient, $applicationId, $clientId,
$developerKey); //object for api calls
```

Δημιουργία ενός αντικειμένου της κλάσης Zend Gdata YouTube

Στη συνέχεια, η εφαρμογή ελέγχει εάν κάποιος χρήστης είναι ήδη συνδεδεμένος με το Google ή όχι. Και στις δύο περιπτώσεις, θα πρέπει ο χρήστης να εξουσιοδοτήσει την εφαρμογή, ώστε να μπορεί αυτή να αποστέλλει βίντεο στο λογαριασμό του και να μπορεί να έχει πρόσβαση στα στοιχεία των χρηστών (ηλικία, φύλο, τοποθεσία,) που έχουν αλληλεπιδράσει με το βίντεο που ανέβασε .

Αφού ολοκληρωθεί η παραπάνω διαδικασία, ο χρήστης θα πρέπει να εισάγει τα στοιχεία τα οποία θα δημοσιευθούν στο Youtube. Αυτό γίνεται μέσω μιας HTML φόρμας η οποία έχει διαμορφωθεί κατάλληλα χρησιμοποιώντας CSS και ελέγχεται πριν αποσταλεί το οτιδήποτε στο server με javascript. Η αποστολή της φόρμας γίνεται με την javascript τεχνική AJAX. Η εφαρμογή ανατρέχει στη βάση δεδομένων, εντοπίζει την καμπάνια η οποία διαμορφώνεται αυτή τη στιγμή και διαμορφώνει δυναμικά τη φόρμα ανάλογα με τις επιλογές του χρήστη.

```
$current_campaign_id = $_SESSION['curent_campaign_id'];
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
        or die('Could not connect to database: ' . mysql_error());
mysql_select_db('ptyxiaki_vasi') or die('Could not select database');
$checkSelectedMedia = mysql_query("SELECT facebook, twitter,
linkedin, youtube, blogger, picasa
FROM campaigns
WHERE id = '$current_campaign_id'
LIMIT 1;
");
mysql_close($link);
$row = mysql_fetch_array($checkSelectedMedia);
$youtubefinal = $row['youtube'];
```

Η εφαρμογή ανατρέχει στη βάση και εντοπίζει την καμπάνια η οποία διαμορφώνεται την παρούσα χρονική στιγμή.

Το κομμάτι τις φόρμας το οποίο αφορά το Youtube φαίνεται παρακάτω

```
if($youtubefinal == 'yes' && $facebookfinal == 'no'){
    $publishform .= '<div class="upload">
                                <input type="file" name="video"
id="video" />
                                <span
id="upload_text2">Video</span><br/>
                                </div>';//echo
}
```

Το κομμάτι τις φόρμας το οποίο αφορά το Youtube

Στη συνέχεια, αφού η φόρμα έχει συμπληρωθεί από το χρήστη, τα δεδομένα μεταφέρονται στο server και στο αρχείο posting.php. Το αρχείο posting.php είναι ένα από τα δύο βασικά αρχεία της εφαρμογής το οποίο αναλαμβάνει τη δημοσίευση περιεχομένου στα Social media με τα οποία η εφαρμογή αλληλεπιδρά. Παρακάτω, παρουσιάζεται το κομμάτι του κώδικα το οποίο αποστέλλει περιεχόμενο στο Youtube.

```
$myVideoEntry = new Zend_Gdata_YouTube_VideoEntry();
// create a new Zend_Gdata_App_MediaFileSource object
$filesorce = $ytf-
>newMediaFileSource($_FILES["video"]["tmp_name"]);
$filesorce->setContentType('video/as'); //----- auto ypo testarisma
$filesorce->setSlug('as.as');
$myVideoEntry->setMediaSource($filesorce);
```

```
$myVideoEntry->setVideoTitle($CampaignName);
$myVideoEntry->setVideoDescription($CampaignName);
$myVideoEntry->SetVideoTags('social campaign');
$uploadUrl = 'http://uploads.gdata.youtube.com/feeds/api/users/default/uploads';
try {
    $newEntry=$ytf-
>insertEntry($myVideoEntry,$uploadUrl,'Zend_Gdata_YouTube_VideoEntry');
        $newEntry->setMajorProtocolVersion(2);
    } catch (Zend_Gdata_App_HttpException $httpException) {
        echo $httpException->getRawResponseBody();
    } catch (Zend_Gdata_App_Exception $e) {
        echo $e->getMessage();
    }
}
$youtube_video_id = $newEntry->getVideoId();
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
or die('Could not connect to database: ' . mysql_error());

mysql_select_db('ptyxiaki_vasi') or die('Could not select database');

mysql_query("INSERT INTO youtube_posting (campaign_id, id_youtube,
type_youtube)
VALUES ('$current_campaign_id', '$youtube_video_id', 'video')");

mysql_close($link);
```

Αποστολή βίντεο στο Youtube και αποθήκευση του id του στη βάση δεδομένων

6.1.4 - LinkedIn

Το API του LinkedIn υλοποιεί μια REST προσέγγιση, γεγονός που το κάνει εύχρηστο και εύκολο στο να υλοποιηθούν εφαρμογές οι οποίες να ενσωματώνουν λειτουργίες του.

Παρακάτω φαίνεται πως εισάγεται προγραμματιστικά στο LinkedIn κείμενο μικρού μήκους(network update).

Αρχικά, εφόσον έχει επιλεγεί το LinkedIn για να δημοσιευθεί σε αυτό περιεχόμενο θα πρέπει ο χρήστης να κάνει είσοδο στην εφαρμογή μέσω του λογαριασμού του στο Google. Η διαδικασία ξεκινάει από το αρχείο createCampaign.php το οποίο εμφανίζει μια HTML φόρμα η οποία έχει τροποποιηθεί χρησιμοποιώντας javascript και πιο συγκεκριμένα τη βιβλιοθήκη jQuery, και εμφανίζει τα διαθέσιμα Social Media στα οποία ο εγγεγραμμένος χρήστης μπορεί να προωθήσει την καμπάνια του.

```
<form method="post" id="selectSM" action="selection.php" >

<div><label for="check1" >Facebook_____</label><input id =
"check1" class = "right" type="checkbox" name="facebook" value="yes" /></div>

<br/>

<div><label for="check2">Twitter_____</label><input id =
"check2" class = "right" type="checkbox" name="twitter" value="yes" /></div>

<br/>

<div><label for="check3">Linkedin_____</label><input id =
"check3" class = "right" type="checkbox" name="linkedin" value="yes" /></div>

<br/>

<div><label for="check4">Youtube_____</label><input id =
"check4" class = "right" type="checkbox" name="youtube" value="yes" /></div>

<br/>

<div><label for="check5">
```

```

Blogger_____</label><input id
= "check5" class = "right" type="checkbox" name="blogger" value="yes" /></div>

<br/>

<div><label for="check6">

Picassa_____</label><input id =
"check6" class = "right" type="checkbox" name="picassa" value="yes" /></div>

<br/>

<div><input id="choosebtn" value="Choose SM" type="submit" /></div>

</form>

```

Η φόρμα από την οποία επιλέγει ο χρήστης τα Social Media στα οποία θα δημοσιευθεί η καμπάνια του.

Η παραπάνω φόρμα αποστέλλεται στον server, και πιο συγκεκριμένα στο αρχείο selection.php, και γίνεται μια καταχώρηση στη βάση δεδομένων με μια νέα καμπάνια η οποία θα δημοσιευθεί στα Social Media τα οποία έχουν επιλεγεί.

```

if( isset( $_POST['facebook'] )
$facebook = $_POST['facebook'];

else
$facebook = 'no';

if( isset( $_POST['twitter'] ) )
$twitter = $_POST['twitter'];

else
$twitter = 'no';

if( isset( $_POST['linkedin'] ) )
$linkedin = $_POST['linkedin'];

else
$linkedin = 'no';

```

```
if( isset( $_POST['youtube'] )
$youtube = $_POST['youtube'];
else
$youtube = 'no';
if( isset( $_POST['blogger'] ) )
$blogger = $_POST['blogger'];
else
$blogger = 'no';
if( isset( $_POST['picassa'] ) )
$picassa = $_POST['picassa'];
else
$picassa = 'no';
$finaluseridee = $_SESSION['userid'];
$current = 1;
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
or die('Could not connect to database: ' . mysql_error());
mysql_select_db('ptyxiaki_vasi') or die('Could not select database');
mysql_query("INSERT INTO campaigns (userid, facebook, twitter, linkedin, youtube,
blogger, picasa, current)
VALUES ( '$finaluseridee', '$facebook', '$twitter', '$linkedin', '$youtube', '$blogger',
'$picassa', '$current')");
```

Έλεγχος της φόρμας και δημιουργία μια νέας εγγραφής στη βάση δεδομένων με τη νέα καμπάνια που δημιουργείται αυτή τη στιγμή.

←T→	id	userid	facebook	twitter	linkedin	youtube	blogger	picasa	campaign_name	current
Edit Copy Delete	690	50	yes	yes	yes	yes	yes	yes	Δοκιμαστικό - συλλογή δεδομένων	0

Εγγραφή νέας καμπάνιας στη βάση, η οποία θα δημοσιευθεί σε όλα τα Social Media τα οποία υποστηρίζει η εφαρμογή.

Αφού γίνει η εγγραφή της καμπάνιας στη βάση δεδομένων, ο χρήστης οδηγείται στο αρχείο auth.php το οποίο ευθύνεται για την αυθεντικοποίηση του χρήστη μέσω του λογαριασμού στο καθένα από τα Social Media που έχουν επιλεγεί για την καμπάνια.

Εδώ, για το LinkedIn, τα στοιχεία του λογαριασμού του χρήστη είναι γνωστά μόνο σε αυτόν για αυτό και η αυθεντικοποίηση μέσω LinkedIn γίνεται με redirect από την εφαρμογή στο LinkedIn και από το LinkedIn στην εφαρμογή.

Αρχικά η εφαρμογή διαβάζει τη βάση και στη συνέχεια ελέγχει εάν ο χρήστης έχει επιλέξει το LinkedIn ώστε να δημοσιευθεί σε αυτό περιεχόμενο της καμπάνιας του.

```

$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
or die('Could not connect to database: ' . mysql_error());

mysql_select_db('ptyxiaki_vasi') or die('Could not select database');

$checkSelectedMedia = mysql_query("SELECT facebook, twitter, linkedin, youtube,
blogger, picasa
FROM campaigns
WHERE id = '$current_campaign_id'
LIMIT 1;
");

mysql_close($link);

$row = mysql_fetch_array($checkSelectedMedia);

$linkedinChecked = $row['linkedin'];

```

Ανάγνωση της βάσης δεδομένων και καταχώρηση της τιμής yes ή no στη μεταβλητή `linkedinChecked`. Εάν η μεταβλητή `linkedinChecked` έχει την τιμή `yes` τότε περιεχόμενο

της καμπάνιας θα δημοσιευθεί στο Youtube. Εάν η τιμή της μεταβλητής `linkedinChecked` είναι no τότε περιεχόμενο της καμπάνιας δε θα δημοσιευθεί στο LinkedIn.

```
if($linkedinChecked == 'yes'){
require './SMauthentications/linkedin.php';
}
```

Εάν η μεταβλητή `youtubeChecked` έχει την τιμή `yes`, ο χρήστης καλείται να συνδεθεί στην εφαρμογή μέσω το λογαριασμού του στο LinkedIn.

Το αρχείο `linkedinauth.php` είναι υπεύθυνο για την αυθεντικοποίηση του χρήστη στην εφαρμογή μέσω Google. Περιέχει κώδικα που είναι υπεύθυνος και για την αυθεντικοποίηση μέσω LinkedIn, του χρήστη που δημιουργεί τη δεδομένη χρονική στιγμή καμπάνια.

Η βιβλιοθήκη `simplelinkedin` αποτελείται από μια μεγάλη κλάση, την κλάση `linkedin_3.1.1.class.php`.

Για να γίνει το οποιοδήποτε API call, θα πρέπει να δημιουργηθεί ένα αντικείμενο της κλάσης αυτής το οποίο στη συνάρτηση δημιουργίας του δέχεται τέσσερα ορίσματα, το `access token`, το `access token secret`, το `oauth token` και το `oauth token secret`. Το απαραίτητο αυτό αντικείμενο, δημιουργείται στο αρχείο `posting.php`.

```
require('./SMauthentications/linkedin_required_files/linkedin_3.1.1.class.php');

$API_CONFIG = array(
    'appKey' => '5ny9d6pikdzz',
    'appSecret' => 'TmB1FyF8LTkVldID',
    'callbackUrl' => NULL
);

$OBJ_linkedin = new LinkedIn($API_CONFIG);
```

Δημιουργία ενός αντικειμένου της κλάσης LinkedIn

Στη συνέχεια, η εφαρμογή ελέγχει εάν κάποιος χρήστης είναι ήδη συνδεδεμένος με το LinkedIn ή όχι. Εάν είναι ήδη συνδεδεμένος η γίνεται ανακατεύθυνση πίσω στην εφαρμογή ενώ εάν δεν είναι εισάγει τα στοιχεία του λογαριασμού του και η ανακατεύθυνση γίνεται στη συνέχεια.

Αφού ολοκληρωθεί η παραπάνω διαδικασία, ο χρήστης θα πρέπει να εισάγει τα στοιχεία τα οποία θα δημοσιευθούν στο LinkedIn. Αυτό γίνεται μέσω μιας HTML φόρμας η οποία έχει διαμορφωθεί κατάλληλα χρησιμοποιώντας CSS και ελέγχεται πριν αποσταλεί το οτιδήποτε στο server με javascript. Η αποστολή της φόρμας γίνεται με την javascript τεχνική AJAX. Η εφαρμογή ανατρέχει στη βάση δεδομένων, εντοπίζει την καμπάνια η οποία διαμορφώνεται αυτή τη στιγμή και διαμορφώνει δυναμικά τη φόρμα ανάλογα με τις επιλογές του χρήστη.

```
$current_campaign_id = $_SESSION['curent_campaign_id'];
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
    or die('Could not connect to database: ' . mysql_error());
mysql_select_db('ptyxiaki_vasi') or die('Could not select database');
$checkSelectedMedia = mysql_query("SELECT facebook, twitter,
linkedin, youtube,
                                blogger, picasa
FROM campaigns
WHERE id = '$current_campaign_id'
LIMIT 1;
");
mysql_close($link);
$row = mysql_fetch_array($checkSelectedMedia);
$linkedinfinal = $row['linkedin'];
```

Η εφαρμογή ανατρέχει στη βάση και εντοπίζει την καμπάνια η οποία διαμορφώνεται την παρούσα χρονική στιγμή.

Το κομμάτι τις φόρμας το οποίο αφορά το LinkedIn φαίνεται παρακάτω

```
if($linkedinfinal == 'yes' && $facebookfinal == 'no'){
    $publishform .= '<div><label for="shortmessage">Short message:
</label><br/>
    <textarea rows="10" cols="40" name="shortmessage"
id="shortmessage"></textarea></div><br/>';//echo
}
```

Το κομμάτι τις φόρμας το οποίο αφορά το LinkedIn

Στη συνέχεια, αφού η φόρμα έχει συμπληρωθεί από το χρήστη, τα δεδομένα μεταφέρονται στο server και στο αρχείο posting.php. Το αρχείο posting.php είναι ένα από τα δύο βασικά αρχεία της εφαρμογής το οποίο αναλαμβάνει τη δημοσίευση περιεχομένου στα Social media με τα οποία η εφαρμογή αλληλεπιδρά. Παρακάτω, παρουσιάζεται το κομμάτι του κώδικα το οποίο αποστέλλει περιεχόμενο στο LinkedIn.

```
$linkedinShortMessage = $_POST['shortmessage'];
    if($linkedinChecked == 'yes' && $twitterChecked == 'yes')
require('../SMauthentications/linkedin_required_files/linkedin_3.1.1.class.p
hp');
    if($linkedinChecked == 'yes' && $twitterChecked == '')
require('../SMauthentications/linkedin_required_files/linkedin_3.1.1.class.ph
p');
    $API_CONFIG = array(
        'appKey' => '5ny9d6pikdzz',
        'appSecret' => 'TmB1FyF8LTkVldID',
        'callbackUrl' => NULL
    );
    $OBJ_linkedin = new LinkedIn($API_CONFIG);
```

```

$OBJ_linkedin->setTokenAccess($_SESSION['oauth']['linkedin']['access']);

$contentl = array();

$contentl['comment'] = $linkedinShortMessage;

$network_update = $OBJ_linkedin->share('new', $contentl);

$profile_id = $OBJ_linkedin->profile('~:(id)');

$final_profile_id = $profile_id['linkedin'];

$response=$OBJ_linkedin>updates('?scope=self&type=STAT',$final_profile_id);

$response['linkedin'] = new SimpleXMLElement($response['linkedin']);

$apantisi = $response['linkedin'];

$key = 'update-key';

$network_update_id = $apantisi->update->$key;// to id tou network update
pou molis anevike

$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
or die('Could not connect to database: ' . mysql_error());

mysql_select_db('ptyxiaki_vasi') or die('Could not select database');

mysql_query("INSERT INTO linkedin_posting (campaign_id, id_linkedin,
type_linkedin)
VALUES ('$current_campaign_id', '$network_update_id',
'network_update')");

mysql_close($link);

```

Αποστολή κειμένου μικρού μήκους στο LinkedIn και αποθήκευση του id του στη βάση δεδομένων

6.1.5. - Blogger

Το API του Blogger υλοποιεί μια REST προσέγγιση, γεγονός που το κάνει εύχρηστο και εύκολο στο να υλοποιηθούν εφαρμογές οι οποίες να ενσωματώνουν λειτουργίες του.

Παρακάτω φαίνεται πως εισάγεται προγραμματιστικά στο Blogger κείμενο μεγάλης έκτασης.

Αρχικά, εφόσον έχει επιλεγεί το Blogger για να δημοσιευθεί σε αυτό περιεχόμενο θα πρέπει ο χρήστης να κάνει είσοδο στην εφαρμογή μέσω του λογαριασμού του στο Google. Η διαδικασία ξεκινάει από το αρχείο createCampaign.php το οποίο εμφανίζει μια HTML φόρμα η οποία έχει τροποποιηθεί χρησιμοποιώντας javascript και πιο συγκεκριμένα τη βιβλιοθήκη jQuery, και εμφανίζει τα διαθέσιμα Social Media στα οποία ο εγγεγραμμένος χρήστης μπορεί να προωθήσει την καμπάνια του.

```
<form method="post" id="selectSM" action="selection.php" >

<div><label for="check1" >Facebook_____</label><input id =
"check1" class = "right" type="checkbox" name="facebook" value="yes" /></div>

<br/>

<div><label for="check2">Twitter_____</label><input id =
"check2" class = "right" type="checkbox" name="twitter" value="yes" /></div>

<br/>

<div><label for="check3">Linkedin_____</label><input id =
"check3" class = "right" type="checkbox" name="linkedin" value="yes" /></div>

<br/>

<div><label for="check4">Youtube_____</label><input id =
"check4" class = "right" type="checkbox" name="youtube" value="yes" /></div>

<br/>

<div><label for="check5">
```

```

Blogger_____</label><input id
= "check5" class = "right" type="checkbox" name="blogger" value="yes" /></div>

<br/>

<div><label for="check6">

Picassa_____</label><input id =
"check6" class = "right" type="checkbox" name="picassa" value="yes" /></div>

<br/>

<div><input id="choosebtn" value="Choose SM" type="submit" /></div>

</form>

```

Η φόρμα από την οποία επιλέγει ο χρήστης τα Social Media στα οποία θα δημοσιευθεί η καμπάνια του.

Η παραπάνω φόρμα αποστέλλεται στον server, και πιο συγκεκριμένα στο αρχείο selection.php, και γίνεται μια καταχώρηση στη βάση δεδομένων με μια νέα καμπάνια η οποία θα δημοσιευθεί στα Social Media τα οποία έχουν επιλεγεί.

```

if( isset( $_POST['facebook'] )
$facebook = $_POST['facebook'];
else
$facebook = 'no';
if( isset( $_POST['twitter'] ) )
$twitter = $_POST['twitter'];
else
$twitter = 'no';
if( isset( $_POST['linkedin'] ) )
$linkedin = $_POST['linkedin'];
else

```

```
$linkedin = 'no';  
  
if( isset( $_POST['youtube'] )  
  
$youtube = $_POST['youtube'];  
  
else  
  
$youtube = 'no';  
  
if( isset( $_POST['blogger'] ) )  
  
$blogger = $_POST['blogger'];  
  
else  
  
$blogger = 'no';  
  
if( isset( $_POST['picassa'] ) )  
  
$picassa = $_POST['picassa'];  
  
else  
  
$picassa = 'no';  
  
$finaluseridee = $_SESSION['userid'];  
  
$current = 1;  
  
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')  
  
or die('Could not connect to database: ' . mysql_error());  
  
mysql_select_db('ptyxiaki_vasi') or die('Could not select database');  
  
mysql_query("INSERT INTO campaigns (userid, facebook, twitter, linkedin, youtube,  
blogger, picasa, current)  
  
VALUES ( '$finaluseridee', '$facebook', '$twitter', '$linkedin', '$youtube', '$blogger',  
'$picassa', '$current'");
```

Έλεγχος της φόρμας και δημιουργία μια νέας εγγραφής στη βάση δεδομένων με τη νέα καμπάνια που δημιουργείται αυτή τη στιγμή.

←T→	id	userid	facebook	twitter	linkedin	youtube	blogger	picasa	campaign_name	current
Edit Copy Delete	690	50	yes	yes	yes	yes	yes	yes	Δοκιμαστικό - συλλογή δεδομένων	0

Εγγραφή νέας καμπάνιας στη βάση, η οποία θα δημοσιευθεί σε όλα τα Social Media τα οποία υποστηρίζει η εφαρμογή.

Αφού γίνει η εγγραφή της καμπάνιας στη βάση δεδομένων, ο χρήστης οδηγείται στο αρχείο auth.php το οποίο ευθύνεται για την αυθεντικοποίηση του χρήστη μέσω του λογαριασμού στο καθένα από τα Social Media που έχουν επιλεγεί για την καμπάνια.

Εδώ, για το Blogger, τα στοιχεία του λογαριασμού του χρήστη είναι γνωστά μόνο σε αυτόν για αυτό και η αυθεντικοποίηση μέσω Google γίνεται με redirect από την εφαρμογή στο Google accounts και από το Google accounts στην εφαρμογή.

Αρχικά η εφαρμογή διαβάζει τη βάση και στη συνέχεια ελέγχει εάν ο χρήστης έχει επιλέξει το Blogger ώστε να δημοσιευθεί σε αυτό περιεχόμενο της καμπάνιας του.

```

$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
or die('Could not connect to database: ' . mysql_error());

mysql_select_db('ptyxiaki_vasi') or die('Could not select database');

$checkSelectedMedia = mysql_query("SELECT facebook, twitter, linkedin, youtube,
blogger, picasa
FROM campaigns
WHERE id = '$current_campaign_id'
LIMIT 1;
");

mysql_close($link);

$row = mysql_fetch_array($checkSelectedMedia);

$bloggerChecked = $row['blogger'];

```

Ανάγνωση της βάσης δεδομένων και καταχώρηση της τιμής yes ή no στη μεταβλητή bloggerChecked. Εάν η μεταβλητή bloggerChecked έχει την τιμή yes τότε περιεχόμενο

της καμπάνιας θα δημοσιευθεί στο Blogger. Εάν η τιμή της μεταβλητής `bloggerChecked` είναι no τότε περιεχόμενο της καμπάνιας δε θα δημοσιευθεί στο Blogger.

```
if($bloggerChecked == 'yes'){
require './SMauthentications/googleauth.php';
}
```

Εάν η μεταβλητή `bloggerChecked` έχει την τιμή `yes`, ο χρήστης καλείται να συνδέθει στην εφαρμογή μέσω το λογαριασμού του στο Google

Το αρχείο `googleauth.php` είναι υπεύθυνο για την αυθεντικοποίηση του χρήστη στην εφαρμογή μέσω Google. Περιέχει κώδικα που είναι υπεύθυνος και για την αυθεντικοποίηση μέσω Google, του χρήστη που δημιουργεί τη δεδομένη χρονική στιγμή καμπάνια και αφορά την αυθεντικοποίηση και για τα τρία προϊόντα της Google (youtube, Blogger, Picasa) με τα οποία αλληλεπιδρά η εφαρμογή η οποία αναπτύχθηκε.

Η βιβλιοθήκη Youtube PHP client API κάνει χρήση μιας μεγάλης βιβλιοθήκης, της βιβλιοθήκης Zend GData. Για να γίνει το οποιοδήποτε API call, θα πρέπει να δημιουργηθεί ένα αντικείμενο της κλάσης Zend Gdata το οποίο στη συνάρτηση δημιουργίας του δέχεται τέσσερα ορίσματα, το `httpClient`, το `applicationId`, το `clientId`, και το `developerKey`. Το απαραίτητο αυτό αντικείμενο, δημιουργείται στο αρχείο `posting.php`.

```
Zend_Loader::loadClass('Zend_Gdata');

    Zend_Loader::loadClass('Zend_Gdata_Query');

    Zend_Loader::loadClass('Zend_Gdata_ClientLogin');

    Zend_Loader::loadClass('Zend_Gdata_AuthSub');

$client = Zend_Gdata_AuthSub::getHttpClient($_SESSION['sessionToken']);
$gdClient = new Zend_Gdata($client);
```

Δημιουργία ενός αντικειμένου της κλάσης Zend Gdata

Στη συνέχεια, η εφαρμογή ελέγχει εάν κάποιος χρήστης είναι ήδη συνδεδεμένος με το Google ή όχι. Και στις δύο περιπτώσεις, θα πρέπει ο χρήστης να εξουσιοδοτήσει την εφαρμογή, ώστε να μπορεί αυτή να αποστέλλει κείμενο μεγάλου μεγέθους στο λογαριασμό του.

Αφού ολοκληρωθεί η παραπάνω διαδικασία, ο χρήστης θα πρέπει να εισάγει τα στοιχεία τα οποία θα δημοσιευθούν στο Blogger. Αυτό γίνεται μέσω μιας HTML φόρμας η οποία έχει διαμορφωθεί κατάλληλα χρησιμοποιώντας CSS και ελέγχεται πριν αποσταλεί το οτιδήποτε στο server με javascript. Η αποστολή της φόρμας γίνεται με την javascript τεχνική AJAX. Η εφαρμογή ανατρέχει στη βάση δεδομένων, εντοπίζει την καμπάνια η οποία διαμορφώνεται αυτή τη στιγμή και διαμορφώνει δυναμικά τη φόρμα ανάλογα με τις επιλογές του χρήστη.

```
$current_campaign_id = $_SESSION['curent_campaign_id'];
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
    or die('Could not connect to database: ' . mysql_error());
mysql_select_db('ptyxiaki_vasi') or die('Could not select database');
$checkSelectedMedia = mysql_query("SELECT facebook, twitter,
linkedin, youtube,
                                blogger, picasa
FROM campaigns
WHERE id = '$current_campaign_id'
LIMIT 1;
");
mysql_close($link);
$row = mysql_fetch_array($checkSelectedMedia);
$bloggerfinal = $row['blogger'];
```

Η εφαρμογή ανατρέχει στη βάση και εντοπίζει την καμπάνια η οποία διαμορφώνεται την παρούσα χρονική στιγμή.

Το κομμάτι τις φόρμας το οποίο αφορά το Blogger φαίνεται παρακάτω

```
if($bloggerfinal == 'yes' && $facebookfinal == 'no'){
    $publishform .= '<div><label for="longmessage">Long message:
</label><br/>
<input type="text" value="" name="longmessage" id="longmessage" />
</div>
<div><label for="blogname">Blog name
: </label><br/>
<input type="text" name="blogname"
id="blogname"></text></div><br/>'; //echo
}
```

Το κομμάτι τις φόρμας το οποίο αφορά το Blogger

Στη συνέχεια, αφού η φόρμα έχει συμπληρωθεί από το χρήστη, τα δεδομένα μεταφέρονται στο server και στο αρχείο posting.php. Το αρχείο posting.php είναι ένα από τα δύο βασικά αρχεία της εφαρμογής το οποίο αναλαμβάνει τη δημοσίευση περιεχομένου στα Social media με τα οποία η εφαρμογή αλληλεπιδρά. Παρακάτω, παρουσιάζεται το κομμάτι του κώδικα το οποίο αποστέλλει περιεχόμενο στο Blogger.

```
$blogname = $_POST['blogname'];
$content = $_POST['longmessage'];
$query = new Zend_Gdata_Query('http://www.blogger.com/feeds/default/blogs');
$feed = $gdClient->getFeed($query);
$blogID = getblogid($feed, $blogname);
$url = 'http://www.blogger.com/feeds/' . $blogID . '/posts/default';
$entry = $gdClient->newEntry();
$entry->title = $gdClient->newTitle($campaignName);
$entry->content = $gdClient->newContent($content);
```

```
$entry->content->setType('text');

$createdPost = $gdClient->insertEntry($entry, $uri);

$idText = explode('-', $createdPost->id->text);

$newPostID = $idText[2];

$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
or die('Could not connect to database: ' . mysql_error());

mysql_select_db('ptyxiaki_vasi') or die('Could not select database');

mysql_query("INSERT INTO blogger_posting (campaign_id, id_blogger, type_blogger,
blog_id)
VALUES ('$current_campaign_id', '$newPostID', 'blog-post', '$blogID')");

mysql_close($link);
```

Αποστολή κειμένου μεγάλης έκτασης στο Blogger και αποθήκευση του id του στη βάση δεδομένων

6.1.6 - Picasa

Το API του Picasa υλοποιεί μια REST προσέγγιση, γεγονός που το κάνει εύχρηστο και εύκολο στο να υλοποιηθούν εφαρμογές οι οποίες να ενσωματώνουν λειτουργίες του.

Παρακάτω φαίνεται πως εισάγεται προγραμματιστικά στο Blogger κείμενο μεγάλης έκτασης.

Αρχικά, εφόσον έχει επιλεγεί το Picasa για να δημοσιευθεί σε αυτό περιεχόμενο θα πρέπει ο χρήστης να κάνει είσοδο στην εφαρμογή μέσω του λογαριασμού του στο Google. Η διαδικασία ξεκινάει από το αρχείο createCampaign.php το οποίο εμφανίζει μια HTML φόρμα η οποία έχει τροποποιηθεί χρησιμοποιώντας javascript και πιο συγκεκριμένα τη βιβλιοθήκη jQuery, και εμφανίζει τα διαθέσιμα Social Media στα οποία ο εγγεγραμμένος χρήστης μπορεί να προωθήσει την καμπάνια του.

```
<form method="post" id="selectSM" action="selection.php" >

<div><label for="check1" >Facebook_____</label><input id =
"check1" class = "right" type="checkbox" name="facebook" value="yes" /></div>

<br/>

<div><label for="check2">Twitter_____</label><input id =
"check2" class = "right" type="checkbox" name="twitter" value="yes" /></div>

<br/>

<div><label for="check3">Linkedin_____</label><input id =
"check3" class = "right" type="checkbox" name="linkedin" value="yes" /></div>

<br/>

<div><label for="check4">Youtube_____</label><input id =
"check4" class = "right" type="checkbox" name="youtube" value="yes" /></div>

<br/>

<div><label for="check5">
```

```

Blogger_____</label><input id
= "check5" class = "right" type="checkbox" name="blogger" value="yes" /></div>

<br/>

<div><label for="check6">

Picasa_____</label><input id =
"check6" class = "right" type="checkbox" name="picassa" value="yes" /></div>

<br/>

<div><input id="choosebtn" value="Choose SM" type="submit" /></div>

</form>

```

Η φόρμα από την οποία επιλέγει ο χρήστης τα Social Media στα οποία θα δημοσιευθεί η καμπάνια του.

Η παραπάνω φόρμα αποστέλλεται στον server, και πιο συγκεκριμένα στο αρχείο selection.php, και γίνεται μια καταχώρηση στη βάση δεδομένων με μια νέα καμπάνια η οποία θα δημοσιευθεί στα Social Media τα οποία έχουν επιλεγεί.

```

if( isset( $_POST['facebook'] )
$facebook = $_POST['facebook'];
else
$facebook = 'no';
if( isset( $_POST['twitter'] ) )
$twitter = $_POST['twitter'];
else
$twitter = 'no';
if( isset( $_POST['linkedin'] ) )
$linkedin = $_POST['linkedin'];
else

```

```
$linkedin = 'no';  
  
if( isset( $_POST['youtube'] )  
  
$youtube = $_POST['youtube'];  
  
else  
  
$youtube = 'no';  
  
if( isset( $_POST['blogger'] ) )  
  
$blogger = $_POST['blogger'];  
  
else  
  
$blogger = 'no';  
  
if( isset( $_POST['picassa'] ) )  
  
$picassa = $_POST['picassa'];  
  
else  
  
$picassa = 'no';  
  
$finaluseridee = $_SESSION['userid'];  
  
$current = 1;  
  
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')  
  
or die('Could not connect to database: ' . mysql_error());  
  
mysql_select_db('ptyxiaki_vasi') or die('Could not select database');  
  
mysql_query("INSERT INTO campaigns (userid, facebook, twitter, linkedin, youtube,  
blogger, picasa, current)  
  
VALUES ( '$finaluseridee', '$facebook', '$twitter', '$linkedin', '$youtube', '$blogger',  
'$picassa', '$current')");
```

Έλεγχος της φόρμας και δημιουργία μια νέας εγγραφής στη βάση δεδομένων με τη νέα καμπάνια που δημιουργείται αυτή τη στιγμή.

	id	userid	facebook	twitter	linkedin	youtube	blogger	picasa	campaign_name	current
	690	50	yes	yes	yes	yes	yes	yes	Δοκιμαστικό - συλλογή δεδομένων	0

Εγγραφή νέας καμπάνιας στη βάση, η οποία θα δημοσιευθεί σε όλα τα Social Media τα οποία υποστηρίζει η εφαρμογή.

Αφού γίνει η εγγραφή της καμπάνιας στη βάση δεδομένων, ο χρήστης οδηγείται στο αρχείο auth.php το οποίο ευθύνεται για την αυθεντικοποίηση του χρήστη μέσω του λογαριασμού στο καθένα από τα Social Media που έχουν επιλεγεί για την καμπάνια.

Εδώ, για το Picasa, τα στοιχεία του λογαριασμού του χρήστη είναι γνωστά μόνο σε αυτόν για αυτό και η αυθεντικοποίηση μέσω Google γίνεται με redirect από την εφαρμογή στο Google accounts και από το Google accounts στην εφαρμογή.

Αρχικά η εφαρμογή διαβάζει τη βάση και στη συνέχεια ελέγχει εάν ο χρήστης έχει επιλέξει το Picasa ώστε να δημοσιευθεί σε αυτό περιεχόμενο της καμπάνιας του.

```
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
or die('Could not connect to database: ' . mysql_error());
mysql_select_db('ptyxiaki_vasi') or die('Could not select database');
$checkSelectedMedia = mysql_query("SELECT facebook, twitter, linkedin, youtube,
blogger, picasa
FROM campaigns
WHERE id = '$current_campaign_id'
LIMIT 1;
");
mysql_close($link);
$row = mysql_fetch_array($checkSelectedMedia);
$picasaChecked = $row[' picasa '];
```

Ανάγνωση της βάσης δεδομένων και καταχώρηση της τιμής yes ή no στη μεταβλητή picasaChecked. Εάν η μεταβλητή picasaChecked έχει την τιμή yes τότε περιεχόμενο

της καμπάνιας θα δημοσιευθεί στο Picasa.. Εάν η τιμή της μεταβλητής `picasaChecked` είναι no τότε περιεχόμενο της καμπάνιας δε θα δημοσιευθεί στο Picasa.

```
if($picasaChecked == 'yes'){
require '../SMauthentications/googleauth.php';
}
```

Εάν η μεταβλητή `bloggerChecked` έχει την τιμή `yes`, ο χρήστης καλείται να συνδεθεί στην εφαρμογή μέσω το λογαριασμού του στο Google

Το αρχείο `googleauth.php` είναι υπεύθυνο για την αυθεντικοποίηση του χρήστη στην εφαρμογή μέσω Google. Περιέχει κώδικα που είναι υπεύθυνος και για την αυθεντικοποίηση μέσω Google, του χρήστη που δημιουργεί τη δεδομένη χρονική στιγμή καμπάνια και αφορά την αυθεντικοποίηση και για τα τρία προϊόντα της Google (youtube, Blogger, Picasa) με τα οποία αλληλεπιδρά η εφαρμογή η οποία αναπτύχθηκε.

Η βιβλιοθήκη Youtube PHP client API κάνει χρήση μιας μεγάλης βιβλιοθήκης, της βιβλιοθήκης Zend GData. Για να γίνει το οποιοδήποτε API call, θα πρέπει να δημιουργηθεί ένα αντικείμενο της κλάσης Zend Gdata Photos το οποίο στη συνάρτηση δημιουργίας του δέχεται τέσσερα ορίσματα, το `httpClient`, το `applicationId`, το `clientId`, και το `developerKey`. Το απαραίτητο αυτό αντικείμενο, δημιουργείται στο αρχείο `posting.php`.

```
Zend_Loader::loadClass('Zend_Gdata_Photos');

    Zend_Loader::loadClass('Zend_Gdata_ClientLogin');

    Zend_Loader::loadClass('Zend_Gdata_AuthSub');

    Zend_Loader::loadClass('Zend_Gdata_App_Exception');

    $client =
Zend_Gdata_AuthSub::getHttpClient($_SESSION['sessionToken']);

    $gp = new Zend_Gdata_Photos($client, "Google-DevelopersGuide-
1.0");
```

Δημιουργία ενός αντικειμένου της κλάσης Zend Gdata

Στη συνέχεια, η εφαρμογή ελέγχει εάν κάποιος χρήστης είναι ήδη συνδεδεμένος με το Google ή όχι. Και στις δύο περιπτώσεις, θα πρέπει ο χρήστης να εξουσιοδοτήσει την εφαρμογή, ώστε να μπορεί αυτή να αποστέλλει κείμενο μεγάλου μεγέθους στο λογαριασμό του.

Αφού ολοκληρωθεί η παραπάνω διαδικασία, ο χρήστης θα πρέπει να εισάγει τα στοιχεία τα οποία θα δημοσιευθούν στο Picasa. Αυτό γίνεται μέσω μιας HTML φόρμας η οποία έχει διαμορφωθεί κατάλληλα χρησιμοποιώντας CSS και ελέγχεται πριν αποσταλεί το οτιδήποτε στο server με javascript. Η αποστολή της φόρμας γίνεται με την javascript τεχνική AJAX. Η εφαρμογή ανατρέχει στη βάση δεδομένων, εντοπίζει την καμπάνια η οποία διαμορφώνεται αυτή τη στιγμή και διαμορφώνει δυναμικά τη φόρμα ανάλογα με τις επιλογές του χρήστη.

```
$current_campaign_id = $_SESSION['curent_campaign_id'];
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
    or die('Could not connect to database: ' . mysql_error());
mysql_select_db('ptyxiaki_vasi') or die('Could not select database');
$checkSelectedMedia = mysql_query("SELECT facebook, twitter,
linkedin, youtube,
                                blogger, picasa
FROM campaigns
WHERE id = '$current_campaign_id'
LIMIT 1;
");
mysql_close($link);
$row = mysql_fetch_array($checkSelectedMedia);
$picasafinal = $row['picasa'];
```

Η εφαρμογή ανατρέχει στη βάση και εντοπίζει την καμπάνια η οποία διαμορφώνεται την παρούσα χρονική στιγμή.

Το κομμάτι τις φόρμας το οποίο αφορά το Picasa φαίνεται παρακάτω.

```
if($picasafinal == 'yes' && $facebookfinal == 'no'){  
    $publishform .= '<div class="upload">  
        <input type="file" name="photo"  
id="photo"/>  
        <span  
id="upload_text">Photo</span><br/>  
        </div><br/>';//echo  
}
```

Το κομμάτι τις φόρμας το οποίο αφορά το Picasa

Στη συνέχεια, αφού η φόρμα έχει συμπληρωθεί από το χρήστη, τα δεδομένα μεταφέρονται στο server και στο αρχείο posting.php. Το αρχείο posting.php είναι ένα από τα δύο βασικά αρχεία της εφαρμογής το οποίο αναλαμβάνει τη δημοσίευση περιεχομένου στα Social media με τα οποία η εφαρμογή αλληλεπιδρά. Παρακάτω, παρουσιάζεται το κομμάτι του κώδικα το οποίο αποστέλλει περιεχόμενο στο Picasa.

```
$username = "default";  
$filename = $photoPath;  
$photoName = $CampaignName;  
$photoCaption = $CampaignName;  
$photoTags = "social campaign";  
$albumId = $newAlbumID;  
$fd = $gp->newMediaFileSource($filename);  
$fd->setContentType("image/jpeg");  
$photoEntry = $gp->newPhotoEntry();  
$photoEntry->setMediaSource($fd);  
$photoEntry->setTitle($gp->newTitle($photoName));
```

```

    $photoEntry->setSummary($gp->newSummary($photoCaption));

    $keywords = new Zend_Gdata_Media_Extension_MediaKeywords();

    $keywords->setText($photoTags);

    $photoEntry->mediaGroup = new
Zend_Gdata_Media_Extension_MediaGroup();

    $photoEntry->mediaGroup->keywords = $keywords;

    $albumQuery = $gp->newAlbumQuery();

    $albumQuery->setUser($username);

    $albumQuery->setAlbumId($albumId);

    $insertedEntry = $gp->insertPhotoEntry($photoEntry, $albumQuery-
>getQueryUrl());

    //echo var_dump($insertedEntry->id->text);

    $idPhotoText = explode('/', $insertedEntry->id->text);

    $newPhotoID = $idPhotoText[11]; //just created albums id

    $link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
or die('Could not connect to database: ' . mysql_error());

    mysql_select_db('ptyxiaki_vasi') or die('Could not select database');

    mysql_query("INSERT INTO picasa_posting (campaign_id, id_picasa,
type_picasa, album_id)
                VALUES ('$current_campaign_id',
'$newPhotoID', 'photo', '$newAlbumID')");

    mysql_close($link);

```

Αποστολή φωτογραφίας στο Picasa και αποθήκευση του id της στη βάση δεδομένων.

6.2 Retrieving

Αρχικά, μόλις συνδεθεί στην εφαρμογή ο χρήστης επιλέγει να του παρουσιασθούν οι καμπάνιες τις οποίες έχει δημιουργήσει. Κατόπιν, ακολουθώντας τις διαδικασίες οι οποίες περιγράφονται παρακάτω, γίνεται η συλλογή των δημοσιευμένων δεδομένων της καμπάνιας τα οποία έχουν προκύψει από την αλληλεπίδραση που είχαν οι χρήστες των κοινωνικών μέσων, Facebook, Twitter, LinkedIn, Youtube, Blogger και Picasa.

6.2.1. - Facebook

Όσο αφορά το Facebook η συλλογή δεδομένων αφορά δύο κατηγορίες, τα αριθμητικά δεδομένα (πλήθος likes, πλήθος comments), το περιεχόμενο των comments και τα δημογραφικά δεδομένα που αφορούν το φύλο, την ηλικία και την τοποθεσία των χρηστών που έχουν αλληλεπιδράσει με το δημοσιευμένο περιεχόμενο.

Μόλις ο χρήστης επιλέξει να του παρουσιαστούν δεδομένα για την καμπάνια που τον ενδιαφέρει και εάν το Facebook ανήκει στα Social media τα οποία έχουν επιλεγεί για να παρουσιαστεί σε αυτά το περιεχόμενο της καμπάνιας του, τότε η εφαρμογή ακολουθεί την εξής διαδικασία.

Αρχικά παρουσιάζεται στο χρήστη μια φόρμα η οποία περιέχει τις καμπάνιες του. η φόρμα δημιουργείται δυναμικά ανάλογα με το πόσες καμπάνιες έχει δημιουργήσει ο χρήστης. Η δημιουργία τις φόρμας γίνεται στο αρχείο viewCampaigns.php το οποίο ανατρέχει στη βάση δεδομένων και παρουσιάζει τις καμπάνιες που αντιστοιχούν στο χρήστη.

```
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j') or die('mysql error' .
mysql_error());

mysql_select_db('ptyxiaki_vasi') or die ('could not connect to database');

mysql_query("SET NAMES 'utf8'");

$userid = $_SESSION['userid'];

$user_campaigns = mysql_query("SELECT * from campaigns WHERE userid = $userid
");

echo '<div id = "viewcamps">

<form action = "campaignid.php" method = "post">
```

```

<label for="campaign">Campaign Name: </label><br/>
<select name="campaign" id="campaign">;
while( $row = mysql_fetch_array($user_campaigns, MYSQL_BOTH)){
    echo '<option>'. $row['campaign_name'] .'</option>';
}
echo '</select>
<input type="submit" id = "viewstats" value = "View stats"/>
</form>
<p>Select one of your campaigns to view charts</p>
</div>;

```

Δημιουργία της φόρμας παρουσίασης των καμπανιών του χρήστη

Αφού επιλεγεί η καμπάνια για την οποία θα πρέπει να γίνει η ανάκτηση δεδομένων, και εάν το Facebook ανήκει στα Social media στα οποία έχει δημοσιευθεί περιεχόμενο της καμπάνιας, τότε θα πρέπει ο χρήστης να συνδεθεί στην εφαρμογή μέσω Facebook, πράγμα το οποίο προτρέπεται από της εφαρμογή να κάνει.

Ο χρήστης οδηγείται στο αρχείο auth.php το οποίο ευθύνεται για την αυθεντικοποίηση του χρήστη μέσω του λογαριασμού στο καθένα από τα Social Media που έχουν επιλεγεί για την καμπάνια.

Εδώ, για το Facebook, τα στοιχεία του λογαριασμού του χρήστη είναι γνωστά μόνο σε αυτόν για αυτό και η αυθεντικοποίηση μέσω Facebook γίνεται με redirect από την εφαρμογή στο Facebook και από το Facebook στην εφαρμογή.

Αρχικά η εφαρμογή διαβάζει τη βάση και στη συνέχεια ελέγχει εάν ο χρήστης έχει επιλέξει το Facebook ώστε να δημοσιευθεί σε αυτό περιεχόμενο της καμπάνιας του.

←T→	id	userid	facebook	twitter	linkedin	youtube	blogger	picasa	campaign_name	current
  	690	50	yes	yes	yes	yes	yes	yes	Δοκιμαστικό - συλλογή δεδομένων	0

Καμπάνια καταχωρημένη στη βάση δεδομένων

```

$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
or die('Could not connect to database: ' . mysql_error());

mysql_select_db('ptyxiaki_vasi') or die('Could not select database');

$checkSelectedMedia = mysql_query("SELECT facebook, twitter, linkedin, youtube,
blogger, picasa
FROM campaigns
WHERE id = '$current_campaign_id'
LIMIT 1;
");

mysql_close($link);

$row = mysql_fetch_array($checkSelectedMedia);

$facebookChecked = $row['facebook'];

```

Ανάγνωση της βάσης δεδομένων. Εάν η μεταβλητή facebookChecked έχει την τιμή yes τότε περιεχόμενο της καμπάνιας έχει δημοσιευθεί στο Facebook. Εάν η τιμή της μεταβλητής facebookChecked είναι no τότε περιεχόμενο της καμπάνιας δεν έχει δημοσιευθεί στο Facebook.

```

if($facebookChecked == 'yes'){
require '../SMauthentications/facebookauth.php';
}

```

Εάν η μεταβλητή facebookChecked έχει την τιμή yes, ο χρήστης καλείται να συνδεθεί στην εφαρμογή μέσω το λογαριασμού του στο Facebook

Το αρχείο facebookauth.php είναι υπεύθυνο για την αυθεντικοποίηση του χρήστη στην εφαρμογή μέσω Facebook, προκειμένου να αποστείλει στο Facebook το περιεχόμενο που επιθυμεί μέσω της εφαρμογής. Καλείται όταν ο χρήστης επιλέξει να δημοσιεύσει περιεχόμενο στο Facebook και παρόλο του ότι είναι πολύ μικρό σε έκταση, παίζει καθοριστική σημασία γιατί είναι αυτό που συνδέει την εφαρμογή με το PHP SDK του Facebook.

Το PHP SDK του Facebook αποτελείται από μια μεγάλη κλάση, την κλάση Facebook.class.php.

Για να γίνει το οποιοδήποτε API call, θα πρέπει να δημιουργηθεί ένα αντικείμενο της κλάσης αυτής το οποίο στη συνάρτηση δημιουργίας του δέχεται δύο ορίσματα, το id του application το οποίο υλοποιούμε και ένα μυστικό κωδικό. Το απαραίτητο αυτό αντικείμενο, δημιουργείται στο αρχείο facebookauth.php.

```
require 'facebook_required_files/facebook.php';

$facebook = new Facebook(array(

    'appld' => '173278596073417',
    'secret'=>'f5fc60352cb1ef235019b1b84be36615',

));
```

Δημιουργία ενός αντικειμένου της κλάσης Facebook

Στη συνέχεια, η εφαρμογή ελέγχει εάν κάποιος χρήστης είναι ήδη συνδεδεμένος στο Facebook ή όχι. Εάν είναι, τότε απλά επιστρέφει στην εφαρμογή πατώντας στο link που του εμφανίζεται. Εάν δεν είναι, τότε θα πρέπει να συνδεθεί με το λογαριασμό του και να συμφωνήσει εάν η εφαρμογή θα πρέπει να έχει πρόσβαση σε κάποια στοιχεία του ή όχι. Για να έχει η εφαρμογή πρόσβαση στα στοιχεία αυτά, και συγκεκριμένα στην άδεια να λάβει πληροφορίες για περιεχόμενο μικρού κειμένου στον τοίχο του (status update), να λάβει πληροφορίες για περιεχόμενο φωτογραφίας (photo upload), λάβει πληροφορίες για περιεχόμενο βίντεο (video upload) και περιεχόμενο μεγάλου κειμένου (note). Πιο συγκεκριμένα, για καθένα από τα στοιχεία τις παραπάνω πρότασης η εφαρμογή ζητάει από το χρήστη τις άδειες publish_stream, user_photos, user_videos, read_stream, user_birthday, friends_birthday και user_notes ώστε να εκτελέσει τις διαδικασίες που χρειάζονται για να δημοσιεύσει περιεχόμενο στο λογαριασμό του συνδεδεμένου χρήστη. Τις άδειες αυτές τις εμφανίζει στο χρήστη μέσω ενός link με το οποίο θα γίνει το login μέσω Facebook.

```
$loginUrl = $facebook-
>getLoginUrl(array('scope'=>'publish_stream,user_photos,user_videos,user_notes,read_stream,user_birthday,friends_birthday,'));
```

Οι άδειες μέσω του login url

Αφού ολοκληρωθεί η παραπάνω διαδικασία, ο χρήστης δίνει εντολή στην εφαρμογή να συλλέξει δεδομένα. Παρακάτω παρουσιάζεται ο κώδικας ο οποίος βρίσκεται στο αρχείο `retrieving.php` και ανακτά δεδομένα για τα στοιχεία τις καμπάνιας τα οποία είναι δημοσιευμένα στο Facebook.

```
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
    or die('Could not connect to database: ' . mysql_error());

mysql_select_db('ptyxiaki_vasi') or die('Could not select database');

mysql_query("SET NAMES 'utf8'");

$facebook_retrieve = mysql_query("SELECT * FROM facebook_posting WHERE
campaign_id = '$selectedCampaign'");
```

Η εφαρμογή ανατρέχει στη βάση δεδομένων και εντοπίζει την καμπάνια η οποία επιλέχθηκε

Κατόπιν, εντοπίζει τα id των στοιχείων τα οποία δημοσιεύτηκαν στο Facebook και με κατάλληλα API calls ανακτά τον αριθμό των comments και των likes τα οποία έχουν γίνει στα δημοσιευμένα στο Facebook στοιχεία της καμπάνιας όπως φαίνεται παρακάτω.

```
while ($row = mysql_fetch_array($facebook_retrieve, MYSQL_ASSOC))
{
    if ( $row['type_facebook'] == 'status_update'){

        $status_update_id = $row['id_facebook'];

        $likes_status_up = $facebook->api('/') . $status_update_id .
'/likes');//epistrefei ena associative array me tis plirofories tou sigkekrimenou status
update pou eksetazetai
```

```

$comments_status_up = $facebook->api('/', $status_update_id .
'/comments',array('limit'=>10000));//epistrefei ena associative array me tis
plirofories tou sigkekrimenou status update

$number_likes_status = count($likes_status_up['data']);//o arithmos twn likes gia to
status update pou eksetazetai

$number_comments_status = count($comments_status_up['data']);//o arithmos
twon comments gia to status update pou eksetazetai

mysql_query("INSERT INTO facebook_retrieving (id_fp, campaign_id, type_facebook,
no_of_comments, no_of_likes) VALUES ('$status_update_id', '$selectedCampaign',
'status_update', '$number_comments_status', '$number_likes_status')");

}

else if ( $row['type_facebook'] == 'photo' ){

$photo_id = $row['id_facebook'];

$likes_photo = $facebook->api('/', $photo_id . '/likes');//epistrefei ena associative
array me tis plirofories tou sigkekrimenou photo pou eksetazetai

$comments_photo = $facebook->api('/', $photo_id .
'/comments',array('limit'=>10000));//epistrefei ena associative array me tis
plirofories tou sigkekrimenou status update

$number_likes_photo = count($likes_photo['data']);//o arithmos twon likes gia to
photo pou eksetazetai

$number_comments_photo = count($comments_photo['data']);//o arithmos twon
comments gia to photo pou eksetazetai

mysql_query("INSERT INTO facebook_retrieving (id_fp, campaign_id, type_facebook,
no_of_comments, no_of_likes) VALUES ('$photo_id', '$selectedCampaign', 'photo',
'$number_comments_photo', '$number_likes_photo')");

}

else if ( $row['type_facebook'] == 'note'){

$note_id = $row['id_facebook'];

$likes_note = $facebook->api('/', $note_id . '/likes');//epistrefei ena associative
array me tis plirofories tou sigkekrimenou note pou eksetazetai

```

```

$comments_note = $facebook->api('/' . $note_id .
'/comments',array('limit'=>10000));//epistrefei ena associative array me tis
plirofories tou sigkekrimenou status update

$number_likes_note = count($likes_note['data']);//o arithmos twn likes gia to note
pou eksetazetai

$number_comments_note = count($comments_note['data']);//o arithmos twn
comments gia to note pou eksetazetai

mysql_query("INSERT INTO facebook_retrieving (id_fp, campaign_id, type_facebook,
no_of_comments, no_of_likes) VALUES ('$note_id', '$selectedCampaign', 'note',
'$number_comments_note', '$number_likes_note')");

    }

else if ( $row['type_facebook'] == 'video'){

    $video_id = $row['id_facebook'];

    $likes_video = $facebook->api('/' . $video_id . '/likes');//epistrefei ena
associative array me tis plirofories tou sigkekrimenou video pou eksetazetai

$comments_video = $facebook->api('/' . $video_id .
'/comments',array('limit'=>10000));//epistrefei ena associative array me tis
plirofories tou sigkekrimenou status update

$number_likes_video = count($likes_video['data']);//o arithmos twn likes gia to
video pou eksetazetai

$number_comments_video = count($comments_video['data']);//o arithmos twn
comments gia to video pou eksetazetai

mysql_query("INSERT INTO facebook_retrieving (id_fp, campaign_id, type_facebook,
no_of_comments, no_of_likes) VALUES ('$video_id', '$selectedCampaign', 'video',
'$number_comments_video', '$number_likes_video')");

    }

}

```

Ανάκτηση του αριθμού των comments και των likes τα οποία έχουν γίνει στα δημοσιευμένα στο Facebook στοιχεία της καμπάνιας

Επίσης, στο παραπάνω κομμάτι κώδικα τα στοιχεία που συλλέγονται καταχωρούνται στη βάση δεδομένων όπως φαίνεται παρακάτω.

campaign_id kampanias	to id tis kip	type_facebook status update, post	no_of_comments status update)	aforun sigkekrimeno postarisma(px aforoun sigkekrimeno	no_of_likes status update)	aforoun sigkekrimeno postarisma(px aforoun sigkekrimeno
850	status_update				0	0
850	photo				0	2
850	note				0	1
850	video				3	1

Αριθμητικά στοιχεία για status update, φωτογραφία, βίντεο και note καμπάνιας δημοσιευμένης στο Facebook.

Στη συνέχεια ανακτώνται δημογραφικά δεδομένα (ηλικία, φύλο, τοποθεσία) των χρηστών οι οποίοι αλληλεπίδρασαν με τα στοιχεία τις καμπάνιας η οποία είχε δημοσιευθεί.

Παρακάτω, φαίνεται πως ανακτώνται προγραμματιστικά από το Facebook τα δημογραφικά δεδομένα για κάθε τύπο ανάρτησης(status update, note ,video, photo).

```

if ( $row['type_facebook'] == 'status_update'){//pairnri ta ids tw n xrhstwn pou
kanane like kai pairnei ilikia genos kai perioxi kai ta vazei sti vasi

    $status_update_id = $row['id_facebook'];

    $likes_status_up = $facebook->api('/') . $status_update_id .
'/likes');//epistrefei ena associative array me tis plirofories tou sigkekrimenou status
update pou eksetazetai

    $comments_status_up = $facebook->api('/') .
$status_update_id . '/comments',array('limit'=>10000));//epistrefei ena associative
array me tis plirofories tou sigkekrimenou status update

foreach ($likes_status_up as $likes){

    foreach($likes as $content){

        $user_profile = $facebook->api('/') . $content['id'];

        $gender = 'unknown';

        $birthday = '01/01/1900';
    }
}

```



```

        $location = 'unknown';

        $user_profile = $facebook->api('/', $content['id']);

    if (isset($user_profile['gender']))
        $gender = $user_profile['gender'];

    if (isset($user_profile['birthday']))
        $birthday = $user_profile['birthday'];

    if (isset($user_profile['location']['name']))
        $location = $user_profile['location']['name'];

    mysql_query("INSERT INTO facebook_users (campaign_id, type, gender, birthday,
    location) VALUES ('$selectedCampaign', 'status_update', '$gender', '$birthday',
    '$location')");
}

```

Ανάκτηση ηλικίας, γένους και περιοχής των χρηστών που πραγματοποίησαν like στο status update(κείμενο μικρού μήκους) της καμπάνιας.

← T →	campaign_id	type	gender	birthday	location
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	850	status_update	female	11/04/1986	unknown

Εγγραφή στη βάση δεδομένων με τα δεδομένα για τους χρήστες όπου πραγματοποίησαν like στο status update.

Έπειτα, συλλέγονται η ηλικία, το γένος και η περιοχή των χρηστών που πραγματοποίησαν comment στο status update(κείμενο μικρού μήκους) της καμπάνιας, και επίσης συγκεντρώνεται και το περιεχόμενο των comments τα οποία έχουν πραγματοποιήσει οι χρήστες που αλληλεπίδρασαν με το συγκεκριμένο status update.

```

foreach ($comments_status_up as $comments){//pairni ta ids twm xrhstwn pou
kanane comment kai pairnei ilikia genos kai perioxi kai ta vazei sti vasi
foreach($comments as $content){

```

```
$user_profile = $facebook->api('/', $content['id']);

$gender = 'unknown';

$birthday = '01/01/1900';

$location = 'unknown';

$user_profile = $facebook->api('/', $content['id']);

if (isset($user_profile['gender']))
    $gender = $user_profile['gender'];

if (isset($user_profile['birthday']))
    $birthday = $user_profile['birthday'];

if (isset($user_profile['location']['name']))
    $location = $user_profile['location']['name'];

mysql_query("INSERT INTO facebook_users (campaign_id, type, gender, birthday,
location) VALUES ('$selectedCampaign', 'status_update', '$gender', '$birthday',
'$location')");
    }

}

foreach ($likes_status_up as $likes)

foreach ($comments_status_up as $comments){
    foreach($comments as $content){
        $message = $content['message'];

        mysql_query("INSERT INTO facebook_comments (campaign_id,
type_facebook, comment) VALUES ('$selectedCampaign', 'status_update',
'$message' )");
    }
}
}
```

Ανάκτηση ηλικίας, γένους και περιοχής των χρηστών που πραγματοποίησαν comment στο status update (κείμενο μικρού μήκους) της καμπάνιας, συλλογή του περιεχομένου του comment και καταχώρησή τους στη βάση δεδομένων.

	campaign_id	type_facebook	comment
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	850	status_update	θέλω λίγο να ημερήσεις.....
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	850	status_update	χαχαχα! κάνε ένα σχόλιο επί της ουσίας αν θέλεις v...
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	850	status_update	α ρε Τόνυ....πάντα εύστοχος....
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	690	status_update	giannnnnnnnnnnnnn... antonnnnnnnnnnnnni

Καταχώρηση στη βάση δεδομένων των comments που έγιναν στο status update της καμπάνιας.

	campaign_id	type	gender	birthday	location
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	850	status_update	female	11/04/1986	unknown

Εγγραφή στη βάση δεδομένων με τα δεδομένα για τους χρήστες όπου πραγματοποίησαν comment στο status update.

```

else if ( $row['type_facebook'] == 'photo' ){

    $photo_id = $row['id_facebook'];

    $likes_photo = $facebook->api('/' . $photo_id .
'/likes');//epistrefei ena associative array me tis plirofories tou sigkekrimenou photo
pou eksetazetai

    $comments_photo = $facebook->api('/' . $photo_id .
'/comments',array('limit'=>10000));//epistrefei ena associative array me tis
plirofories tou sigkekrimenou status update

    foreach ($likes_photo as $likes){

        foreach($likes as $content){

            $user_profile = $facebook->api('/' . $content['id']);

$gender = 'unknown';

$birthday = '01/01/1900';

$location = 'unknown';

```

```

$user_profile = $facebook->api('/', $content['id']);
if (isset($user_profile['gender']))
$gender = $user_profile['gender'];
if (isset($user_profile['birthday']))
$birthday = $user_profile['birthday'];
if (isset($user_profile['location']['name']))
$location = $user_profile['location']['name'];

mysql_query("INSERT INTO facebook_users (campaign_id, type, gender, birthday,
location) VALUES ('$selectedCampaign', 'status_update', '$gender', '$birthday',
'$location')");

    }
}

```

Ανάκτηση ηλικίας, γένους και περιοχής των χρηστών που πραγματοποίησαν like στη φωτογραφία της καμπάνιας.

	campaign_id	type	gender	birthday	location
<input type="checkbox"/> Edit Inline Edit Copy Delete	850	status_update	female	11/04/1986	unknown

Εγγραφή στη βάση δεδομένων με τα δεδομένα για τους χρήστες όπου πραγματοποίησαν like στη φωτογραφία.

Έπειτα, συλλέγονται η ηλικία, το γένος και η περιοχή των χρηστών που πραγματοποίησαν comment στη φωτογραφία της καμπάνιας, και επίσης συγκεντρώνεται και το περιεχόμενο των comments τα οποία έχουν πραγματοποιήσει οι χρήστες που αλληλεπίδρασαν με τη συγκεκριμένη φωτογραφία.

```
foreach ($comments_photo as $comments){//pairri ta ids tw n xrhstwn pou kanane
comment kai pairnei ilikia genos kai perioxi kai ta vazei sti vasi

        foreach($comments as $content){

$user_profile = $facebook->api('/', $content['id']);

$gender = 'unknown';

$birthday = '01/01/1900';

$location = 'unknown';

$user_profile = $facebook->api('/', $content['id']);

if (isset($user_profile['gender']))

$gender = $user_profile['gender'];

if (isset($user_profile['birthday']))

$birthday = $user_profile['birthday'];

if (isset($user_profile['location']['name']))

$location = $user_profile['location']['name'];

mysql_query("INSERT INTO facebook_users (campaign_id, type, gender, birthday,
location) VALUES ('$selectedCampaign', 'status_update', '$gender', '$birthday',
'$location')");

        }

}

foreach ($likes_status_up as $likes)

foreach ($comments_photo as $comments){

foreach($comments as $content){

$message = $content['message'];

mysql_query("INSERT INTO facebook_comments (campaign_id, type_facebook,
comment) VALUES ('$selectedCampaign', 'status_update', '$message' )");

        }

    }

}
```

Ανάκτηση ηλικίας, γένους και περιοχής των χρηστών που πραγματοποίησαν comment στη φωτογραφία της καμπάνιας, συλλογή του περιεχομένου του comment και καταχώρησή τους στη βάση δεδομένων .

		campaign_id	type_facebook	comment
<input type="checkbox"/>	Edit Inline Edit Copy Delete	850	status_update	θέλω λίγο να ημερήσεις.....
<input type="checkbox"/>	Edit Inline Edit Copy Delete	850	status_update	χαχαχα! κάνε ένα σχόλιο επί της ουσίας αν θέλεις v...
<input type="checkbox"/>	Edit Inline Edit Copy Delete	850	status_update	α ρε Τόνυ....πάντα εύστοχος....
<input type="checkbox"/>	Edit Inline Edit Copy Delete	690	status_update	giannnnnnnnnnnnnn... antonnnnnnnnnnnnn

Καταχώρηση στη βάση δεδομένων των comments που έγιναν στη φωτογραφία καμπάνιας.

		campaign_id	type	gender	birthday	location
<input type="checkbox"/>	Edit Inline Edit Copy Delete	850	status_update	female	11/04/1986	unknown

Εγγραφή στη βάση δεδομένων με τα δεδομένα για τους χρήστες όπου πραγματοποίησαν comment στη φωτογραφία.

Έπειτα, συλλέγονται η ηλικία, το γένος και η περιοχή των χρηστών που πραγματοποίησαν comment στο κείμενο μεγάλης έκτασης(note) της καμπάνιας, και επίσης συγκεντρώνεται και το περιεχόμενο των comments τα οποία έχουν πραγματοποιήσει οι χρήστες που αλληλεπίδρασαν με το συγκεκριμένο note.

```

else if ( $row['type_facebook'] == 'note'){

    $note_id = $row['id_facebook'];

    $likes_note = $facebook->api('/', $note_id .
'/likes');//epistrefei ena associative array me tis pliروفories του sigkekrimenou note
που eksetazetai

    $comments_note = $facebook->api('/', $note_id .
'/comments',array('limit'=>10000));//epistrefei ena associative array me tis
pliروفories του sigkekrimenou status update

```

```
foreach ($likes_note as $likes){
    foreach($likes as $content){

        $user_profile = $facebook->api('/', $content['id']);

        $gender = 'unknown';
        $birthday = '01/01/1900';
        $location = 'unknown';

        $user_profile = $facebook->api('/', $content['id']);

        if (isset($user_profile['gender']))
            $gender = $user_profile['gender'];

        if (isset($user_profile['birthday']))
            $birthday = $user_profile['birthday'];

        if (isset($user_profile['location']['name']))
            $location = $user_profile['location']['name'];

        mysql_query("INSERT INTO facebook_users (campaign_id, type, gender, birthday,
        location) VALUES ('$selectedCampaign', 'status_update', '$gender', '$birthday',
        '$location')");

    }
}
```

Ανάκτηση ηλικίας, γένους και περιοχής των χρηστών που πραγματοποίησαν like στο note της καμπάνιας.

←T→	campaign_id	type	gender	birthday	location
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	850	status_update	female	11/04/1986	unknown

Εγγραφή στη βάση δεδομένων με τα δεδομένα για τους χρήστες όπου πραγματοποίησαν like στο note.

```
foreach ($comments_note as $comments){//pairnri ta ids tw n xrhstwn pou kanane
comment kai pairnei ilikia genos kai perioxi kai ta vazei sti vasi
```

```
foreach($comments as $content){
```

```
$user_profile = $facebook->api('/', $content['id']);
```

```
$gender = 'unknown';
```

```
$birthday = '01/01/1900';
```

```
$location = 'unknown';
```

```
$user_profile = $facebook->api('/', $content['id']);
```

```
if (isset($user_profile['gender']))
```

```
$gender = $user_profile['gender'];
```

```
if (isset($user_profile['birthday']))
```

```
$birthday = $user_profile['birthday'];
```

```
if (isset($user_profile['location']['name']))
```


Έπειτα, συλλέγονται η ηλικία, το γένος και η περιοχή των χρηστών που πραγματοποίησαν comment στο βίντεο της καμπάνιας, και επίσης συγκεντρώνεται και το περιεχόμενο των comments τα οποία έχουν πραγματοποιήσει οι χρήστες που αλληλεπίδρασαν με το συγκεκριμένο βίντεο.

```
else if ( $row['type_facebook'] == 'video'){  
    $video_id = $row['id_facebook'];  
    $likes_video = $facebook->api('/' . $video_id .  
'/likes');//epistrefei ena associative array me tis plirofories tou sigkekrimenou video  
    pou eksetazetai  
    $comments_video = $facebook->api('/' . $video_id .  
'/comments',array('limit'=>10000));//epistrefei ena associative array me tis  
    plirofories tou sigkekrimenou status update  
  
    foreach ($likes_video as $likes){  
        foreach($likes as $content){  
  
            $user_profile = $facebook->api('/' . $content['id']);  
  
            $gender = 'unknown';  
            $birthday = '01/01/1900';  
            $location = 'unknown';  
  
            $user_profile = $facebook->api('/' . $content['id']);  
  
            if (isset($user_profile['gender']))  
                $gender = $user_profile['gender'];  
  
            if (isset($user_profile['birthday']))
```

```

$birthday = $user_profile['birthday'];

if (isset($user_profile['location']['name']))
    $location = $user_profile['location']['name'];

mysql_query("INSERT INTO facebook_users (campaign_id, type, gender, birthday,
location) VALUES ('$selectedCampaign', 'status_update', '$gender', '$birthday',
'$location')");
    }
}

```

Ανάκτηση ηλικίας, γένους και περιοχής των χρηστών που πραγματοποίησαν like στο βίντεο της καμπάνιας.

← T →	campaign_id	type	gender	birthday	location
<input type="checkbox"/> Edit Inline Edit Copy Delete	850	status_update	female	11/04/1986	unknown

Εγγραφή στη βάση δεδομένων με τα δεδομένα για τους χρήστες όπου πραγματοποίησαν like στο βίντεο.

```

foreach ($comments_video as $comments){//pairnri ta ids tw n xrhstwn pou kanane
comment kai pairnei ilikia genos kai perioxi kai ta vazei sti vasi

foreach($comments as $content){

$user_profile = $facebook->api('/'. $content['id']);

$gender = 'unknown';

$birthday = '01/01/1900';

$location = 'unknown';

```

```

$user_profile = $facebook->api('/', $content['id']);

if (isset($user_profile['gender']))
    $gender = $user_profile['gender'];

if (isset($user_profile['birthday']))
    $birthday = $user_profile['birthday'];

if (isset($user_profile['location']['name']))
    $location = $user_profile['location']['name'];

mysql_query("INSERT INTO facebook_users (campaign_id, type, gender, birthday,
location) VALUES ('$selectedCampaign', 'status_update', '$gender', '$birthday',
'$location')");
    }

} //foreach ($likes_status_up as $likes)

foreach ($comments_video as $comments){
    foreach($comments as $content){
        $message = $content['message'];

        mysql_query("INSERT INTO facebook_comments (campaign_id,
type_facebook, comment) VALUES ('$selectedCampaign', 'status_update',
'$message' )");
    }
}
}

```

Ανάκτηση ηλικίας, γένους και περιοχής των χρηστών που πραγματοποίησαν comment στο βίντεο της καμπάνιας, συλλογή του περιεχομένου του comment και καταχώρησή τους στη βάση δεδομένων .

	campaign_id	type_facebook	comment
<input type="checkbox"/> Edit Inline Edit Copy Delete	850	status_update	θέλω λίγο να ημερήσεις....
<input type="checkbox"/> Edit Inline Edit Copy Delete	850	status_update	χαχαχα! κάνε ένα σχόλιο επί της ουσίας αν θέλεις ν...
<input type="checkbox"/> Edit Inline Edit Copy Delete	850	status_update	α ρε Τόνυ....πάντα εύστοχος....
<input type="checkbox"/> Edit Inline Edit Copy Delete	690	status_update	gianniiiiiiiiiiii... antoniiiiiiiiiiiiii

Καταχώρηση στη βάση δεδομένων των comments που έγιναν στο βίντεο της καμπάνιας.

	campaign_id	type	gender	birthday	location
<input type="checkbox"/> Edit Inline Edit Copy Delete	850	status_update	female	11/04/1986	unknown

Εγγραφή στη βάση δεδομένων με τα δεδομένα για τους χρήστες όπου πραγματοποίησαν comment στο βίντεο.

6.2.2 - Twitter

Όσο αφορά το Twitter η συλλογή δεδομένων αφορά δύο κατηγορίες, τα αριθμητικά δεδομένα (πλήθος retweets) και τα δημογραφικά δεδομένα που αφορούν το φύλο και την τοποθεσία των χρηστών που έχουν αλληλεπιδράσει με το δημοσιευμένο περιεχόμενο.

Μόλις ο χρήστης επιλέξει να του παρουσιαστούν δεδομένα για την καμπάνια που τον ενδιαφέρει και εάν το Twitter ανήκει στα Social media τα οποία έχουν επιλεγεί για να παρουσιαστεί σε αυτά το περιεχόμενο της καμπάνιας του, τότε η εφαρμογή ακολουθεί την εξής διαδικασία.

Αρχικά παρουσιάζεται στο χρήστη μια φόρμα η οποία περιέχει τις καμπάνιες του. η φόρμα δημιουργείται δυναμικά ανάλογα με το πόσες καμπάνιες έχει δημιουργήσει ο χρήστης. Η δημιουργία τις φόρμας γίνεται στο αρχείο viewCampaigns.php το οποίο ανατρέχει στη βάση δεδομένων και παρουσιάζει τις καμπάνιες που αντιστοιχούν στο χρήστη.

```
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j') or die('mysql error' .
mysql_error());
```

```
mysql_select_db('ptyxiaki_vasi') or die ('could not connect to database');
```

```
mysql_query("SET NAMES 'utf8'");

$userid = $_SESSION['userid'];

$user_campaigns = mysql_query("SELECT * from campaigns WHERE userid = $userid
");

echo '<div id = "viewcamps">

    <form action = "campaignid.php" method = "post">

        <label for="campaign">Campaign Name: </label><br/>

        <select name="campaign" id="campaign">';

while( $row = mysql_fetch_array($user_campaigns, MYSQL_BOTH)){

    echo '<option>'. $row['campaign_name'] . '</option>';

}

echo '</select>

<input type="submit" id = "viewstats" value = "View stats"/>

</form>

<p>Select one of your campaigns to view charts</p>

</div>';
```

Δημιουργία της φόρμας παρουσίασης των καμπανιών του χρήστη

Αφού επιλεγεί η καμπάνια για την οποία θα πρέπει να γίνει η ανάκτηση δεδομένων, και εάν το Twitter ανήκει στα Social media στα οποία έχει δημοσιευθεί περιεχόμενο της καμπάνιας, τότε θα πρέπει ο χρήστης να συνδεθεί στην εφαρμογή μέσω Twitter, πράγμα το οποίο προτρέπει από της εφαρμογή να κάνει.

Ο χρήστης οδηγείται στο αρχείο auth.php το οποίο ευθύνεται για την αυθεντικοποίηση του χρήστη μέσω του λογαριασμού στο καθένα από τα Social Media που έχουν επιλεγεί για την καμπάνια.

Εδώ, για το Twitter, τα στοιχεία του λογαριασμού του χρήστη είναι γνωστά μόνο σε αυτόν για αυτό και η αυθεντικοποίηση μέσω Facebook γίνεται με redirect από την εφαρμογή στο Twitter και από το Twitter στην εφαρμογή.

Αρχικά η εφαρμογή διαβάζει τη βάση και στη συνέχεια ελέγχει εάν ο χρήστης έχει επιλέξει το Twitter ώστε να δημοσιευθεί σε αυτό περιεχόμενο της καμπάνιας του.

←T→	id	userid	facebook	twitter	linkedin	youtube	blogger	picasa	campaign_name	current
	690	50	yes	yes	yes	yes	yes	yes	Δοκιμαστικό - συλλογή δεδομένων	0

Καμπάνια καταχωρημένη στη βάση δεδομένων

```

$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')

or die('Could not connect to database: ' . mysql_error());

mysql_select_db('ptyxiaki_vasi') or die('Could not select database');

$checkSelectedMedia = mysql_query("SELECT facebook, twitter, linkedin, youtube,
blogger, picasa
FROM campaigns
WHERE id = '$current_campaign_id'
LIMIT 1;
");

```

```
mysql_close($link);

$row = mysql_fetch_array($checkSelectedMedia);

$twitterChecked = $row['twitter'];
```

Ανάγνωση της βάσης δεδομένων. Εάν η μεταβλητή `twitterChecked` έχει την τιμή `yes` τότε περιεχόμενο της καμπάνιας έχει δημοσιευθεί στο Twitter. Εάν η τιμή της μεταβλητής `twitterChecked` είναι `no` τότε περιεχόμενο της καμπάνιας δεν έχει δημοσιευθεί στο Twitter.

```
if($twitterChecked == 'yes'){
require '../SMauthentications/twitterauth.php';
}
```

Εάν η μεταβλητή `twitterChecked` έχει την τιμή `yes`, ο χρήστης καλείται να συνδεθεί στην εφαρμογή μέσω το λογαριασμού του στο Twitter

Το αρχείο `twitterkauth.php` είναι υπεύθυνο για την αυθεντικοποίηση του χρήστη στην εφαρμογή μέσω Twitter, προκειμένου να αποστείλει στο Twitter το περιεχόμενο που επιθυμεί μέσω της εφαρμογής. Καλείται όταν ο χρήστης επιλέξει να δημοσιεύσει περιεχόμενο στο Twitter και παρόλο του ότι είναι πολύ μικρό σε έκταση, παίζει καθοριστική σημασία γιατί είναι αυτό που συνδέει την εφαρμογή με τη βιβλιοθήκη `twitteroauth` του Twitter.

Η βιβλιοθήκη `twitteroauth` του Twitter αποτελείται από μια μεγάλη κλάση, την κλάση `twitteroauth.class.php`.

Για να γίνει το οποιοδήποτε API call, θα πρέπει να δημιουργηθεί ένα αντικείμενο της κλάσης αυτής το οποίο στη συνάρτηση δημιουργίας του δέχεται δύο ορίσματα, το `id` του application το οποίο υλοποιούμε και ένα μυστικό κωδικό. Το απαραίτητο αυτό αντικείμενο, δημιουργείται στο αρχείο `retrieving.php`.


```
$connection = new TwitterOAuth('WkcWcGjPXDFQ6Z3Uz2QA',
'ssaq0aC36Tfi4UDTbJo42MTBTHXd6E2XatpqsCLoI', $access_token['oauth_token'],
$access_token['oauth_token_secret']);
```

Δημιουργία ενός αντικειμένου της κλάσης Twitteroauth

Στη συνέχεια, η εφαρμογή ελέγχει εάν κάποιος χρήστης είναι ήδη συνδεδεμένος στο Twitter ή όχι. Και στις δύο περιπτώσεις, θα πρέπει ο χρήστης να εξουσιοδοτήσει την εφαρμογή, ώστε να μπορεί αυτή να έχει πρόσβαση στα tweets του, να παρακολουθεί ποιούς χρήστες του Twitter ακολουθεί ο συνδεδεμένος χρήστης αλλά και ποιοί τον ακολουθούν, να ανανεώνει το προφίλ του χρήστη και να δημοσιεύει tweets για αυτόν.

Μόλις ο χρήστης δώσει εξουσιοδότηση στην εφαρμογή για αυτές τις ενέργειες, μεταφέρεται και πάλι στην εφαρμογή.

Αφού ολοκληρωθεί η παραπάνω διαδικασία, ο χρήστης δίνει εντολή στην εφαρμογή να συλλέξει δεδομένα. Παρακάτω παρουσιάζεται ο κώδικας ο οποίος βρίσκεται στο αρχείο `retrieving.php` και ανακτά δεδομένα για τα στοιχεία τις καμπάνιας τα οποία είναι δημοσιευμένα στο Twitter.

```
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')

or die('Could not connect to database: ' . mysql_error());

mysql_select_db('ptyxiaki_vasi') or die('Could not select database');

mysql_query("SET NAMES 'utf8'");

$ twitter _retrieve = mysql_query("SELECT * FROM twitter _posting WHERE
campaign_id = '$selectedCampaign');
```

Η εφαρμογή ανατρέχει στη βάση δεδομένων και εντοπίζει την καμπάνια η οποία επιλέχθηκε

Κατόπιν, εντοπίζει τα id των στοιχείων τα οποία δημοσιεύτηκαν στο Twitter και με κατάλληλα API calls ανάκτά το πλήθος των retweets που έχουν γίνει στο tweet της καμπάνιας, όπως επίσης το φύλο και την τοποθεσία των χρηστών που έχουν κάνει retweet το συγκεκριμένο tweet.

```
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')

    or die('Could not connect to database: ' . mysql_error());

mysql_select_db('ptyxiaki_vasi') or die('Could not select database');

mysql_query("SET NAMES 'utf8'");

    $twitter_retrieve = mysql_query("SELECT * FROM twitter_posting
WHERE campaign_id =          '$selectedCampaign'");

    $row = mysql_fetch_array($twitter_retrieve, MYSQL_ASSOC);

    $tweet_id = $row['id_twitter'];

    $retweets = $connection->get('statuses/retweets/' . $tweet_id);

    $no_of_retweets = count($retweets);

    mysql_query("INSERT INTO twitter_retrieving (id_twitter,
campaign_id, type_twitter,          no_of_retweets) VALUES ('$tweet_id',
'$selectedCampaign', 'tweet', '$no_of_retweets' )");
```

```

$retweets_by = $connection->get('statuses/' . $tweet_id .
'/retweeted_by');

foreach( $retweets_by as $users){

$location = $users->location;

mysql_query("INSERT INTO twitter_users (campaign_id, location)
VALUES
                ('$selectedCampaign', '$location' )");
}

```

Ανάκτηση του πλήθους των retweets που έχουν γίνει στο tweet της καμπάνιας και του φύλου και της τοποθεσίας των χρηστών που έχουν κάνει retweet το συγκεκριμένο tweet. Στη συνέχεια όλα αυτά τα στοιχεία αποθηκεύονται στη βάση δεδομένων.

	id_tr to id tou pinaka	id_twitter to id tou tweet	campaign_id id tis kampanias	type_twitter	no_of_retweets
<input type="checkbox"/> Edit Copy Delete	12	119809051403091969	850	tweet	0
<input type="checkbox"/> Edit Copy Delete	14	114781619012911104	690	tweet	3
<input type="checkbox"/> Edit Copy Delete	15	119786464153313281	848	tweet	0

Εγγραφή αριθμού retweets για ένα συγκεκριμένο tweet στη βάση δεδομένων

	campaign_id	type	gender	birthday	location
<input type="checkbox"/> Edit Copy Delete	690	tweet	unknown	unknown	Trikala Greece
<input type="checkbox"/> Edit Copy Delete	690	tweet	unknown	unknown	xios
<input type="checkbox"/> Edit Copy Delete	690	tweet	unknown	unknown	samos - tumba

Εγγραφές στη βάση δεδομένων για τους χρήστες που έχουν πραγματοποιήσει retweet για το tweet της καμπάνιας

6.2.3. - YouTube

Όσο αφορά το Youtube η συλλογή δεδομένων αφορά δύο κατηγορίες, τα αριθμητικά δεδομένα (πλήθος views, πλήθος comments, video rating) και τα δημογραφικά δεδομένα που αφορούν το φύλο, την ηλικία και την τοποθεσία των χρηστών που έχουν αλληλεπιδράσει με το δημοσιευμένο περιεχόμενο, καθώς και το περιεχόμενο των comments των χρηστών που έχουν αλληλεπιδράσει με το βίντεο που έχει αποστείλει η εφαρμογή για λογαριασμό του χρήστη στο Youtube.

Μόλις ο χρήστης επιλέξει να του παρουσιαστούν δεδομένα για την καμπάνια που τον ενδιαφέρει και εάν το Youtube ανήκει στα Social media τα οποία έχουν επιλεγεί για να παρουσιαστεί σε αυτά το περιεχόμενο της καμπάνιας του, τότε η εφαρμογή ακολουθεί την εξής διαδικασία.

Αρχικά παρουσιάζεται στο χρήστη μια φόρμα η οποία περιέχει τις καμπάνιες του. η φόρμα δημιουργείται δυναμικά ανάλογα με το πόσες καμπάνιες έχει δημιουργήσει ο χρήστης. Η δημιουργία τις φόρμας γίνεται στο αρχείο viewCampaigns.php το οποίο ανατρέχει στη βάση δεδομένων και παρουσιάζει τις καμπάνιες που αντιστοιχούν στο χρήστη.

```
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j') or die('mysql error' .
mysql_error());

mysql_select_db('ptyxiaki_vasi') or die ('could not connect to database');

mysql_query("SET NAMES 'utf8'");

$userid = $_SESSION['userid'];

$user_campaigns = mysql_query("SELECT * from campaigns WHERE userid = $userid
");

echo '<div id = "viewcamps">

    <form action = "campaignid.php" method = "post">

    <label for="campaign">Campaign Name: </label><br/>

    <select name="campaign" id="campaign">;

while( $row = mysql_fetch_array($user_campaigns, MYSQL_BOTH)){

    echo '<option>'. $row['campaign_name'] . '</option>';

}
```

```

echo '</select>

<input type="submit" id = "viewstats" value = "View stats"/>

</form>

<p>Select one of your campaigns to view charts</p>

</div>';

```

Δημιουργία της φόρμας παρουσίασης των καμπανιών του χρήστη

Αφού επιλεγεί η καμπάνια για την οποία θα πρέπει να γίνει η ανάκτηση δεδομένων, και εάν το Youtube ανήκει στα Social media στα οποία έχει δημοσιευθεί περιεχόμενο της καμπάνιας, τότε θα πρέπει ο χρήστης να συνδεθεί στην εφαρμογή μέσω Google, πράγμα το οποίο προτρέπει από της εφαρμογή να κάνει.

Ο χρήστης οδηγείται στο αρχείο auth.php το οποίο ευθύνεται για την αυθεντικοποίηση του χρήστη μέσω του λογαριασμού στο στο καθένα από τα Social Media που έχουν επιλεγεί για την καμπάνια.

Εδώ, για το Youtube, τα στοιχεία του λογαριασμού του χρήστη είναι γνωστά μόνο σε αυτόν για αυτό και η ευθεντικοποίηση μέσω Google γίνεται με redirect από την εφαρμογή στο Google accounts και από το Google accounts στην εφαρμογή.

Αρχικά η εφαρμογή διαβάζει τη βάση και στη συνέχεια ελέγχει εάν ο χρήστης έχει επιλέξει το Youtube ώστε να δημοσιευθεί σε αυτό περιεχόμενο της καμπάνιας του.

←T→	id	userid	facebook	twitter	linkedin	youtube	blogger	picasa	campaign_name	current
 Edit  Inline Edit  Copy  Delete	690	50	yes	yes	yes	yes	yes	yes	Δοκιμαστικό - συλλογή δεδομένων	0

Καμπάνια καταχωρημένη στη βάση δεδομένων

```

$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
or die('Could not connect to database: ' . mysql_error());

mysql_select_db('ptyxiaki_vasi') or die('Could not select database');

```

```
$checkSelectedMedia = mysql_query("SELECT facebook, twitter, linkedin, youtube,
blogger, picasa
FROM campaigns
WHERE id = '$current_campaign_id'
LIMIT 1;
");

mysql_close($link);

$row = mysql_fetch_array($checkSelectedMedia);

$youtubeChecked = $row['youtube'];
```

Ανάγνωση της βάσης δεδομένων. Εάν η μεταβλητή youtubeChecked έχει την τιμή yes τότε περιεχόμενο της καμπάνιας έχει δημοσιευθεί στο Youtube. Εάν η τιμή της μεταβλητής youtubeChecked είναι no τότε περιεχόμενο της καμπάνιας δεν έχει δημοσιευθεί στο Youtube.

```
if($youtubeChecked == 'yes'){
require '../SMauthentications/googleauth.php';
}
```

Εάν η μεταβλητή youtubeChecked έχει την τιμή yes, ο χρήστης καλείται να συνδέθει στην εφαρμογή μέσω το λογαριασμού του στο Google

Το αρχείο googleauth.php είναι υπεύθυνο για την αυθεντικοποίηση του χρήστη στην εφαρμογή μέσω Google, προκειμένου να αποστείλει στο Youtube το περιεχόμενο που επιθυμεί μέσω της εφαρμογής. Καλείται όταν ο χρήστης επιλέξει να δημοσιεύσει περιεχόμενο στο Twitter και παρόλο του ότι είναι πολύ μικρό σε έκταση, παίζει καθοριστική σημασία γιατί είναι αυτό που συνδέει την εφαρμογή με τη βιβλιοθήκη Youtube PHP client API του Youtube.

Η βιβλιοθήκη Youtube PHP client API κάνει χρήση μιας μεγάλης βιβλιοθήκης, της βιβλιοθήκης Zend GData. Για να γίνει το οποιοδήποτε API call, θα πρέπει να δημιουργηθεί ένα αντικείμενο της κλασης Zend Gdata YouTube το οποίο

στη συνάρτηση δημιουργίας του δέχεται τέσσερα ορίσματα, το httpClient, το applicationId, το clientId, και το developerKey. Το απαραίτητο αυτό αντικείμενο, δημιουργείται στο αρχείο retrieving.php.

```
$developerKey = 'A139si5hMq_qXwym1kmm9WkxWNsKtM8AJ3Wxlz-lyB3ej8zrTk01QPZ4OUOTPMnlcmmXdPyk6c0XUIxOtXR0W79-ibBXpUy-vg';

$applicationId = 'velas';

$clientId = 'Social Campaign';

Zend_Loader::loadClass('Zend_Gdata_YouTube');

        Zend_Loader::loadClass('Zend_Gdata_AuthSub');

        Zend_Loader::loadClass('Zend_Gdata_App_Exception');

        $httpClient =
Zend_Gdata_AuthSub::getHttpClient($_SESSION['sessionToken']);

$ytf = new Zend_Gdata_YouTube($httpClient, $applicationId, $clientId,
$developerKey); //object for api calls
```

Δημιουργία ενός αντικειμένου της κλάσης Zend_Gdata_YouTube

Στη συνέχεια, η εφαρμογή ελέγχει εάν κάποιος χρήστης είναι ήδη συνδεδεμένος με το Google ή όχι. Και στις δύο περιπτώσεις, θα πρέπει ο χρήστης να εξουσιοδοτήσει την εφαρμογή, ώστε να μπορεί αυτή να έχει πρόσβαση στο πλήθος των views, πλήθος comments, περιεχόμενο comments, video rating και στα στοιχεία των χρηστών (ηλικία, φύλο, τοποθεσία) που έχουν αλληλεπιδράσει με το βίντεο που ανέβασε.

Παρακάτω παρουσιάζεται πως γίνεται η ανάκτηση αυτών των στοιχείων από το Youtube προγραμματιστικά.

```
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')

or die('Could not connect to database: ' . mysql_error());

mysql_select_db('ptyxiaki_vasi') or die('Could not select database');

mysql_query("SET NAMES 'utf8'");

$youtube_retrieve = mysql_query("SELECT * FROM youtube_posting WHERE
campaign_id = '$selectedCampaign'");

$row = mysql_fetch_array($youtube_retrieve, MYSQL_BOTH);

$youtube_id = $row['id_youtube'];

//AVEGAVE RATING AND VIDEO VIEWS AND HOW MANY PEOPLE INTERACTED AND
VIEWS

$videoEntry = $ytf->getVideoEntry($youtube_id);

$rating = $videoEntry->getVideoRatingInfo();

$views = $videoEntry->getVideoViewCount();//number of views

$average_video_rating = $rating['average'];//average rating

$number_of_people_rated = $rating['numRaters'];//number of people rated

//AVEGAVE RATING AND VIDEO VIEWS AND HOW MANY PEOPLE INTERACTED AND
VIEWS

//COMMENTS AND USER INFO

$commentFeed = $ytf->getVideoCommentFeed($youtube_id);

$count = 0;// o counter tha deixnei to number of comments
```



```

foreach ($commentFeed as $commentEntry) {

    $comment_text = $commentEntry->title->text;// to text tou comment

    $comment_user = $commentEntry->author[0]->name->text;//o xrhsths pou ekane
    to comment

    $userProfileEntry = $ytf->getUserProfile($comment_user);

    $comment_user_age = $userProfileEntry->getAge();

    $comment_user_gender = $userProfileEntry->getGender();

    $comment_user_location = $userProfileEntry->getLocation();

    mysql_query("INSERT INTO google_comments (campaign_id,type_google, comment,
    social_media) VALUES ('$selectedCampaign', 'video', '$comment_text', 'youtube')");

    mysql_query("INSERT INTO google_users (campaign_id, type, gender, birthday,
    location, social_media) VALUES ('$selectedCampaign', 'video',
    '$comment_user_gender', '$comment_user_age', '$comment_user_location',
    'youtube')");

    $count++;

    }

    mysql_query("INSERT INTO youtube_retrieving (id_youtube, campaign_id,
    type_youtube, no_of_comments, no_of_views, no_of_likes, rating) VALUES
    ('$youtube_id', '$selectedCampaign', 'video', '$count', '$views',
    '$number_of_people_rated', '$average_video_rating')");

    //COMMENTS AND USER INFO

    mysql_close($link);

```

Ανάκτηση πλήθους των views, πλήθους comments, περιεχόμενου comments, video rating και στα στοιχεία των χρηστών (ηλικία, φύλο, τοποθεσία) και στη συνέχεια αποθήκευσή τους στη βάση δεδομένων

id_youtube	to id tou pinaka youtube_posting	campaign_id	id tou pinaka campaigns	type_youtube	no_of_comments	no_of_views	no_of_likes
a-pZK50ofZQ		690		video	7	14	2
Kml0x7dsCoE		848		video	0	0	0

Εγγραφή στη βάση δεδομένων με αριθμητικά στοιχεία για το βίντεο της καμπάνιας

← T →	campaign_id	type	gender	birthday	location	social_media
<input type="checkbox"/> Edit Inline Edit Copy Delete	690	video	m	24	GR	youtube
<input type="checkbox"/> Edit Inline Edit Copy Delete	690	video	m	24	GR	youtube
<input type="checkbox"/> Edit Inline Edit Copy Delete	690	video	m	44	US	youtube
<input type="checkbox"/> Edit Inline Edit Copy Delete	690	video	m	24	GR	youtube
<input type="checkbox"/> Edit Inline Edit Copy Delete	690	video	m	24	GR	youtube
<input type="checkbox"/> Edit Inline Edit Copy Delete	690	video	m	24	GR	youtube
<input type="checkbox"/> Edit Inline Edit Copy Delete	690	video	m	24	GR	youtube

Εγγραφή στη βάση δεδομένων με δημογραφικά στοιχεία για τες χρήστες που αλληλεπίδρασαν με βίντεο της καμπάνιας

6.2.4 - LinkedIn

Όσο αφορά το LinkedIn, η συλλογή δεδομένων αφορά δύο κατηγορίες, τα αριθμητικά δεδομένα (πλήθος likes, πλήθος comments), το περιεχόμενο των comments και τα δημογραφικά δεδομένα που αφορούν την ηλικία και την τοποθεσία των χρηστών που έχουν αλληλεπιδράσει με το δημοσιευμένο περιεχόμενο.

Μόλις ο χρήστης επιλέξει να του παρουσιαστούν δεδομένα για την καμπάνια που τον ενδιαφέρει και εάν το LinkedIn ανήκει στα Social media τα οποία έχουν επιλεγεί για να παρουσιαστεί σε αυτά το περιεχόμενο της καμπάνιας του, τότε η εφαρμογή ακολουθεί την εξής διαδικασία.

Αρχικά παρουσιάζεται στο χρήστη μια φόρμα η οποία περιέχει τις καμπάνιες του. η φόρμα δημιουργείται δυναμικά ανάλογα με το πόσες καμπανιες έχει δημιουργήσει ο χρήστης. Η δημιουργία τις φόρμας γίνεται στο αρχείο viewCampaigns.php το οποίο ανατρέχει στη βάση δεδομένων και παρουσιάζει τις καμπάνιες που αντιστοιχούν στο χρήστη.

```

$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j') or die('mysql error' .
mysql_error());

mysql_select_db('ptyxiaki_vasi') or die ('could not connect to database');

mysql_query("SET NAMES 'utf8'");

$userid = $_SESSION['userid'];

$user_campaigns = mysql_query("SELECT * from campaigns WHERE userid = $userid
");

echo '<div id = "viewcamps">

    <form action = "campaignid.php" method = "post">

    <label for="campaign">Campaign Name: </label><br/>

    <select name="campaign" id="campaign">;

while( $row = mysql_fetch_array($user_campaigns, MYSQL_BOTH)){

    echo '<option>'. $row['campaign_name'] . '</option>';

}

```

```

echo '</select>

<input type="submit" id = "viewstats" value = "View stats"/>

</form>

<p>Select one of your campaigns to view charts</p>

</div>';

```

Δημιουργία της φόρμας παρουσίασης των καμπανιών του χρήστη

Αφού επιλεγεί η καμπάνια για την οποία θα πρέπει να γίνει η ανάκτηση δεδομένων, και εάν το LinkedIn ανήκει στα Social media στα οποία έχει δημοσιευθεί περιεχόμενο της καμπάνιας, τότε θα πρέπει ο χρήστης να συνδεθεί στην εφαρμογή μέσω LinkedIn, πράγμα το οποίο προτρέπεται από της εφαρμογή να κάνει.

Ο χρήστης οδηγείται στο αρχείο auth.php το οποίο ευθύνεται για την αυθεντικοποίηση του χρήστη μέσω του λογαριασμού στο στο καθένα από τα Social Media που έχουν επιλεγεί για την καμπάνια.

Εδώ, για το LinkedIn, τα στοιχεία του λογαριασμού του χρήστη είναι γνωστά μόνο σε αυτόν για αυτό και η ευθεντικοποίηση μέσω LinkedIn γίνεται με redirect από την εφαρμογή στο LinkedIn και από το LinkedIn στην εφαρμογή.

Αρχικά η εφαρμογή διαβάζει τη βάση και στη συνέχεια ελέγχει εάν ο χρήστης έχει επιλέξει το LinkedIn ώστε να δημοσιευθεί σε αυτό περιεχόμενο της καμπάνιας του.

←T→	id	userid	facebook	twitter	linkedin	youtube	blogger	picasa	campaign_name	current
 Edit  Copy  Delete	690	50	yes	yes	yes	yes	yes	yes	Δοκιμαστικό - συλλογή δεδομένων	0

Καμπάνια καταχωρημένη στη βάση δεδομένων

```

$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
or die('Could not connect to database: ' . mysql_error());

```

```
mysql_select_db('ptyxiaki_vasi') or die('Could not select database');

$checkSelectedMedia = mysql_query("SELECT facebook, twitter, linkedin, youtube,
blogger, picasa
FROM campaigns
WHERE id = '$current_campaign_id'
LIMIT 1;
");

mysql_close($link);

$row = mysql_fetch_array($checkSelectedMedia);

$linkedinChecked = $row['linkedin'];
```

Ανάγνωση της βάσης δεδομένων. Εάν η μεταβλητή linkedinChecked έχει την τιμή yes τότε περιεχόμενο της καμπάνιας έχει δημοσιευθεί στο LinkedIn. Εάν η τιμή της μεταβλητής linkedinChecked είναι no τότε περιεχόμενο της καμπάνιας δεν έχει δημοσιευθεί στο LinkedIn.

```
if($linkedinChecked == 'yes'){
require '../SMauthentications/linkedinauth.php';
}
```

Εάν η μεταβλητή youtubeChecked έχει την τιμή yes, ο χρήστης καλείται να συνδέθει στην εφαρμογή μέσω το λογαριασμού του στο LinkedIn

Το αρχείο linkedinauth.php είναι υπεύθυνο για την αυθεντικοποίηση του χρήστη στην εφαρμογή μέσω LinkedIn. Περιέχει κώδικα που είναι υπεύθυνος και για την αυθεντικοποίηση μέσω LinkedIn, του χρήστη που δημιουργεί τη δεδομένη χρονική στιγμή καμπάνια.

Η βιβλιοθήκη simplelinkedin αποτελείται από μια μεγάλη κλάση, την κλάση linkedin_3.1.1.class.php.

Για να γίνει το οποιοδήποτε API call, θα πρέπει να δημιουργηθεί ένα αντικείμενο της κλάσης αυτής το οποίο στη συνάρτηση δημιουργίας του δέχεται τέσσερα ορίσματα, το access token, το access token secret, το oauth token και το oauth token secret. Το απαραίτητο αυτό αντικείμενο, δημιουργείται στο αρχείο posting.php.

```
$API_CONFIG = array(
    'appKey' => '5ny9d6pikdzz',
    'appSecret' => 'TmB1FyF8LTkVldID',
    'callbackUrl' => NULL
);

$obj_linkedin = new LinkedIn($API_CONFIG);
```

Δημιουργία ενός αντικειμένου της κλάσης SimpleLinkedIn

Στη συνέχεια, η εφαρμογή ελέγχει εάν κάποιος χρήστης είναι ήδη συνδεδεμένος με το LinkedIn ή όχι. Εάν είναι ήδη συνδεδεμένος η γίνεται ανακατεύθυνση πίσω στην εφαρμογή ενώ εάν δεν είναι εισάγει τα στοιχεία του λογαριασμού του και η ανακατεύθυνση γίνεται στη συνέχεια.

Παρακάτω παρουσιάζεται πως γίνεται η ανάκτηση των αριθμητικών και δημογραφικών στοιχείων από το LinkedIn προγραμματιστικά.

```
mysql_select_db('ptyxiaki_vasi') or die('Could not select database');

mysql_query("SET NAMES 'utf8'");

$linkedin_retrieve = mysql_query("SELECT * FROM
linkedin_posting WHERE campaign_id = '$selectedCampaign'");

$row = mysql_fetch_array($linkedin_retrieve, MYSQL_ASSOC);

$linkedin_id = $row['id_linkedin'];
```

```

    $likes = $OBJ_linkedin->likes($linkedin_id);

    $likess = new SimpleXMLElement($likes['linkedin']);

    $comments = $OBJ_linkedin->comments($linkedin_id);

    $comments2 = new SimpleXMLElement($comments['linkedin']);

foreach ($likess as $sup)//looparisma mesa sta comments
{
    $finale = $sup;

    $person_id = $finale->person->id;

    $person = $OBJ_linkedin->profile2('(id=' . $person_id . '):(location)');

    $person2 = $OBJ_linkedin->profile2('(id=' . $person_id . '):(date-of-birth)');

    $person = new SimpleXMLElement($person['linkedin']);

    $person2 = new SimpleXMLElement($person2['linkedin']);

    $date = 'date-of-birth';

    $year = $person2->person->$date->year;

    $loc = $person->person->location->name;

    mysql_query("INSERT INTO linkedin_users (campaign_id, birthday, location)
VALUES ('$selectedCampaign', '$year', '$loc' )");

    //echo '<br/>' . $person->person->location->name;

    //echo '<br/>' . $person2->person->$date->year;

}

    $number_of_likes = $likess['total'];

    $number_of_comments = $comments2['total'];//YEAH!

    mysql_query("INSERT INTO linkedin_retrieving (id_linkedin, campaign_id,
no_of_comments, no_of_likes) VALUES ('$linkedin_id', '$selectedCampaign',
'$number_of_comments', '$number_of_likes' )");

```

```
//GETTING NUMBER OF COMMENTS

//GETTING TEXT OF THE COMMENTS

$comments3 = $OBJ_linkedin->comments($linkedin_id);
$comments4 = new SimpleXMLElement($comments3['linkedin']);
foreach ($comments4 as $up)//looparisma mesa sta comments
{
    $finale = $up;
    $person_id = $finale->person->id;
        $person = $OBJ_linkedin->profile2('(id=' . $person_id . '):(location)');
        $person2 = $OBJ_linkedin->profile2('(id=' . $person_id . '):(date-of-
birth)');
        $person = new SimpleXMLElement($person['linkedin']);
        $person2 = new SimpleXMLElement($person2['linkedin']);
        $date = 'date-of-birth';
        $year = $person2->person->$date->year;
        $loc = $person->person->location->name;
        mysql_query("INSERT INTO linkedin_users (campaign_id, birthday,
location) VALUES ('$selectedCampaign', '$year', '$loc' )");
    }
foreach ($comments4 as $up)//looparisma mesa sta comments //taking comments
text
{
    $finale = $up;
    $text = $finale->comment;
```



```
mysql_query("INSERT INTO linkedin_comments (campaign_id,
comment) VALUES ('$selectedCampaign', '$text' );")
}

//GETTING TEXT OF THE COMMENTS

mysql_close($link);
```

Ανάκτηση πλήθους των likes, πλήθους comments, περιεχόμενου comments, video και στα στοιχεία χρηστών (ηλικία, τοποθεσία) και στη συνέχεια αποθήκευσή τους στη βάση δεδομένων

	campaign_id	type_linkedin	comment
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	690	network_update	Δοκιμαστικό
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	690	network_update	dsgsggh
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	690	network_update	pare ena akoma

Αποθηκευμένα comments στη βάση δεδομένων, χρηστών που έχουν αλληλεπιδράσει με το κείμενο μικρού μήκους που αναρτήθηκε στο LinkedIn

	campaign_id	type	gender	birthday	location
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	690	network_update	unknown	1987	Greece
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	690	network_update	unknown	1986	Greece
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	690	network_update	unknown	1987	Greece
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	690	network_update	unknown	1987	Greece
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	690	network_update	unknown	1986	Greece

Αποθηκευμένα δημογραφικά στοιχεία χρηστών που έχουν αλληλεπιδράσει με το κείμενο μικρού μήκους που αναρτήθηκε στο LinkedIn

6.2.5 - Blogger

Όσο αφορά το Blogger η συλλογή δεδομένων αφορά τη συλλογή του πλήθους των comments και τη συλλογή του περιεχομένου των comments που έχουν υποβληθεί στο κείμενο μεγάλου μήκους της καμπάνιας.

Μόλις ο χρήστης επιλέξει να του παρουσιαστούν δεδομένα για την καμπάνια που τον ενδιαφέρει και εάν το Blogger ανήκει στα Social media τα οποία έχουν επιλεγεί για να παρουσιαστεί σε αυτά το περιεχόμενο της καμπάνιας του, τότε η εφαρμογή ακολουθεί την εξής διαδικασία.

Αρχικά παρουσιάζεται στο χρήστη μια φόρμα η οποία περιέχει τις καμπάνιες του. η φόρμα δημιουργείται δυναμικά ανάλογα με το πόσες καμπάνιες έχει δημιουργήσει ο χρήστης. Η δημιουργία τις φόρμας γίνεται στο αρχείο viewCampaigns.php το οποίο ανατρέχει στη βάση δεδομένων και παρουσιάζει τις καμπάνιες που αντιστοιχούν στο χρήστη.

```
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j') or die('mysql error' .
mysql_error());

mysql_select_db('ptyxiaki_vasi') or die ('could not connect to database');

mysql_query("SET NAMES 'utf8'");

$userid = $_SESSION['userid'];

$user_campaigns = mysql_query("SELECT * from campaigns WHERE userid = $userid
");

echo '<div id = "viewcamps">

    <form action = "campaignid.php" method = "post">

    <label for="campaign">Campaign Name: </label><br/>

    <select name="campaign" id="campaign">;

while( $row = mysql_fetch_array($user_campaigns, MYSQL_BOTH)){

    echo '<option>' . $row['campaign_name'] . '</option>';

}

echo '</select>
```

```

<input type="submit" id = "viewstats" value = "View stats"/>
</form>
<p>Select one of your campaigns to view charts</p>
</div>;

```

Δημιουργία της φόρμας παρουσίασης των καμπανιών του χρήστη

Αφού επιλεγεί η καμπάνια για την οποία θα πρέπει να γίνει η ανάκτηση δεδομένων, και εάν το Blogger ανήκει στα Social media στα οποία έχει δημοσιευθεί περιεχόμενο της καμπάνιας, τότε θα πρέπει ο χρήστης να συνδεθεί στην εφαρμογή μέσω Google, πράγμα το οποίο προτρέπεται από της εφαρμογή να κάνει.

Ο χρήστης οδηγείται στο αρχείο auth.php το οποίο ευθύνεται για την αυθεντικοποίηση του χρήστη μέσω του λογαριασμού στο στο καθένα από τα Social Media που έχουν επιλεγεί για την καμπάνια.

Εδώ, για το Blogger, τα στοιχεία του λογαριασμού του χρήστη είναι γνωστά μόνο σε αυτόν για αυτό και η ευθεντικοποίηση μέσω Google γίνεται με redirect από την εφαρμογή στο Google accounts και από το Google accounts στην εφαρμογή.

Αρχικά η εφαρμογή διαβάζει τη βάση και στη συνέχεια ελέγχει εάν ο χρήστης έχει επιλέξει το Blogger ώστε να δημοσιευθεί σε αυτό περιεχόμενο της καμπάνιας του.

←T→	id	userid	facebook	twitter	linkedin	youtube	blogger	picasa	campaign_name	current
	690	50	yes	yes	yes	yes	yes	yes	Δοκιμαστικό - συλλογή δεδομένων	0

Καμπάνια καταχωρημένη στη βάση δεδομένων

```

$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
or die('Could not connect to database: ' . mysql_error());
mysql_select_db('ptyxiaki_vasi') or die('Could not select database');

```

```
$checkSelectedMedia = mysql_query("SELECT facebook, twitter, linkedin, youtube,
blogger, picasa
FROM campaigns
WHERE id = '$current_campaign_id'
LIMIT 1;
");

mysql_close($link);

$row = mysql_fetch_array($checkSelectedMedia);

$bloggerChecked = $row['blogger'];
```

*Ανάγνωση της βάσης δεδομένων. Εάν η μεταβλητή `bloggerChecked` έχει την τιμή `yes` τότε περιεχόμενο της καμπάνιας έχει δημοσιευθεί στο *Blogger*. Εάν η τιμή της μεταβλητής `bloggerChecked` είναι `no` τότε περιεχόμενο της καμπάνιας δεν έχει δημοσιευθεί στο *Blogger*.*

```
if($bloggerChecked == 'yes'){
require '../SMauthentications/googleauth.php';
}
```

*Εάν η μεταβλητή `bloggerChecked` έχει την τιμή `yes`, ο χρήστης καλείται να συνδέθει στην εφαρμογή μέσω το λογαριασμού του στο *Google**

Το αρχείο `googleauth.php` είναι υπεύθυνο για την αυθεντικοποίηση του χρήστη στην εφαρμογή μέσω *Google*, προκειμένου να αποστείλει στο *Youtube* το περιεχόμενο που επιθυμεί μέσω της εφαρμογής. Καλείται όταν ο χρήστης επιλέξει να δημοσιεύσει περιεχόμενο στο *Blogger* και παρόλο του ότι είναι πολύ μικρό σε έκταση, παίζει καθοριστική σημασία γιατί είναι αυτό που συνδέει την εφαρμογή με τη βιβλιοθήκη *Youtube PHP client API* του *Youtube*.

Η βιβλιοθήκη Blogger Data API for PHP κάνει χρήση μιας μεγάλης βιβλιοθήκης, της βιβλιοθήκης Zend GData. Για να γίνει το οποιοδήποτε API call, θα πρέπει να δημιουργηθεί ένα αντικείμενο της κλάσης Zend Gdata το οποίο στη συνάρτηση δημιουργίας του δέχεται ένα όρισμα, το gdClient. Το απαραίτητο αυτό αντικείμενο, δημιουργείται στο αρχείο retrieving.php.

```
$client = Zend_Gdata_AuthSub::getHttpClient($_SESSION['sessionToken']);  
  
$gdClient = new Zend_Gdata($client);  
  
$draft = new Zend_Gdata_App_Extension_Draft();
```

Δημιουργία ενός αντικειμένου της κλάσης Zend_Gdata

Στη συνέχεια, η εφαρμογή ελέγχει εάν κάποιος χρήστης είναι ήδη συνδεδεμένος με το Google ή όχι. Και στις δύο περιπτώσεις, θα πρέπει ο χρήστης να εξουσιοδοτήσει την εφαρμογή, ώστε να μπορεί αυτή να έχει πρόσβαση στο πλήθος των comments και στο περιεχόμενο των comments των χρηστών που έχουν αλληλεπιδράσει με το κείμενο μεγάλου μήκους που ανέβασε.

Παρακάτω παρουσιάζεται πως γίνεται η ανάκτηση αυτών των στοιχείων από το Youtube προγραμματιστικά.

```
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')  
  
or die('Could not connect to database: ' . mysql_error());  
  
mysql_select_db('ptyxiaki_vasi') or die('Could not select database');  
mysql_query("SET NAMES 'utf8'");  
  
$blogger_retrieve = mysql_query("SELECT id_blogger, blog_id FROM  
blogger_posting WHERE campaign_id = '$selectedCampaign'");  
  
$row = mysql_fetch_array($blogger_retrieve, MYSQL_ASSOC);  
  
$blogID = $row['blog_id'];  
  
$post_id = $row['id_blogger'];
```

```

$query = new Zend_Gdata_Query('http://www.blogger.com/feeds/'
. $blogID . '/' . $post_id . '/comments/default');

$feed = $gdClient->getFeed($query);

$comment_counter = 0;

foreach($feed->entries as $entry) {

    $comment_text = $entry->title->text; // τι $i είναι το πλήθος των
    σχολίων και το $entry->title->text είναι το κείμενο του κάθε σχολίου

    mysql_query("INSERT INTO google_comments (campaign_id,
    type_google, comment, social_media) VALUES ('$selectedCampaign', 'blog-post',
    '$comment_text', 'blogger')");

    $comment_counter++;

}

mysql_query("INSERT INTO blogger_retrieving (id_blogger, campaign_id,
    type_blogger, no_of_comments) VALUES ('$post_id', '$selectedCampaign', 'blog-
    post', '$comment_counter')");

mysql_close($link);

```

Ανάκτηση πλήθους και περιεχομένου comments και στη συνέχεια αποθήκευσή τους στη βάση δεδομένων

	id_br	id_blogger to id tou blog post	campaign_id	type_blogger	no_of_comments
<input type="checkbox"/> Edit Inline Edit Copy Delete	8	7511832905107595451	690	blog-post	4
<input type="checkbox"/> Edit Inline Edit Copy Delete	9	2627205815399269972	848	blog-post	0

Εγγραφές στη βάση δεδομένων με το πλήθος των comments

<input type="checkbox"/> Edit Inline Edit Copy Delete	690	blog-post	akoma ena	blogger
<input type="checkbox"/> Edit Inline Edit Copy Delete	690	blog-post	σχόλιο δοκιμαστικό	blogger
<input type="checkbox"/> Edit Inline Edit Copy Delete	690	blog-post	kai edw!!	blogger
<input type="checkbox"/> Edit Inline Edit Copy Delete	690	blog-post	dokimastiko sxolio!!! :P	blogger

Εγγραφές στη βάση δεδομένων με comments που έχουν γίνει στο κείμενο μεγάλου μήκους

6.2.6 - Picasa

Όσο αφορά το Picasa η συλλογή δεδομένων αφορά τη συλλογή του πλήθους των comments και τη συλλογή του περιεχομένου των comments που έχουν υποβληθεί στο κείμενο μεγάλου μήκους της καμπάνιας.

Μόλις ο χρήστης επιλέξει να του παρουσιαστούν δεδομένα για την καμπάνια που τον ενδιαφέρει και εάν το Picasa ανήκει στα Social media τα οποία έχουν επιλεγεί για να παρουσιαστεί σε αυτά το περιεχόμενο της καμπάνιας του, τότε η εφαρμογή ακολουθεί την εξής διαδικασία.

Αρχικά παρουσιάζεται στο χρήστη μια φόρμα η οποία περιέχει τις καμπάνιες του. η φόρμα δημιουργείται δυναμικά ανάλογα με το πόσες καμπανιες έχει δημιουργήσει ο χρήστης. Η δημιουργία τις φόρμας γίνεται στο αρχείο viewCampaigns.php το οποίο ανατρέχει στη βάση δεδομένων και παρουσιάζει τις καμπάνιες που αντιστοιχούν στο χρήστη.

```
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j') or die('mysql error' .
mysql_error());

mysql_select_db('ptyxiaki_vasi') or die ('could not connect to database');

mysql_query("SET NAMES 'utf8'");

$userid = $_SESSION['userid'];

$user_campaigns = mysql_query("SELECT * from campaigns WHERE userid = $userid
");

echo '<div id = "viewcamps">

    <form action = "campaignid.php" method = "post">

    <label for="campaign">Campaign Name: </label><br/>

    <select name="campaign" id="campaign">;

while( $row = mysql_fetch_array($user_campaigns, MYSQL_BOTH)){

    echo '<option>' . $row['campaign_name'] . '</option>';

}

echo '</select>
```

```
<input type="submit" id = "viewstats" value = "View stats"/>
</form>
<p>Select one of your campaigns to view charts</p>
</div>;
```

Δημιουργία της φόρμας παρουσίασης των καμπανιών του χρήστη

Αφού επιλεγεί η καμπάνια για την οποία θα πρέπει να γίνει η ανάκτηση δεδομένων, και εάν το Picasa ανήκει στα Social media στα οποία έχει δημοσιευθεί περιεχόμενο της καμπάνιας, τότε θα πρέπει ο χρήστης να συνδεθεί στην εφαρμογή μέσω Google, πράγμα το οποίο προτρέπεται από της εφαρμογή να κάνει.

Ο χρήστης οδηγείται στο αρχείο auth.php το οποίο ευθύνεται για την αυθεντικοποίηση του χρήστη μέσω του λογαριασμού στο στο καθένα από τα Social Media που έχουν επιλεγεί για την καμπάνια.

Εδώ, για το Picasa, τα στοιχεία του λογαριασμού του χρήστη είναι γνωστά μόνο σε αυτόν για αυτό και η ευθεντικοποίηση μέσω Google γίνεται με redirect από την εφαρμογή στο Google accounts και από το Google accounts στην εφαρμογή.

Αρχικά η εφαρμογή διαβάζει τη βάση και στη συνέχεια ελέγχει εάν ο χρήστης έχει επιλέξει το Picasa ώστε να δημοσιευθεί σε αυτό περιεχόμενο της καμπάνιας του.

	id	userid	facebook	twitter	linkedin	youtube	blogger	picasa	campaign_name	current
	690	50	yes	yes	yes	yes	yes	yes	Δοκιμαστικό - συλλογή δεδομένων	0

Καμπάνια καταχωρημένη στη βάση δεδομένων

```
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
or die('Could not connect to database: ' . mysql_error());

mysql_select_db('ptyxiaki_vasi') or die('Could not select database');

$checkSelectedMedia = mysql_query("SELECT facebook, twitter, linkedin, youtube,
blogger, picasa
FROM campaigns
```



```

WHERE id = '$current_campaign_id'

LIMIT 1;

");

mysql_close($link);

$row = mysql_fetch_array($checkSelectedMedia);

$picasaChecked = $row['picasa'];

```

Ανάγνωση της βάσης δεδομένων. Εάν η μεταβλητή `picasaChecked` έχει την τιμή `yes` τότε περιεχόμενο της καμπάνιας έχει δημοσιευθεί στο Picasa. Εάν η τιμή της μεταβλητής `picasaChecked` είναι `no` τότε περιεχόμενο της καμπάνιας δεν έχει δημοσιευθεί στο Picasa.

```

if($picasaChecked == 'yes'){
require '../SMauthentications/googleauth.php';
}

```

Εάν η μεταβλητή `picasaChecked` έχει την τιμή `yes`, ο χρήστης καλείται να συνδεθεί στην εφαρμογή μέσω το λογαριασμού του στο Google

Το αρχείο `googleauth.php` είναι υπεύθυνο για την αυθεντικοποίηση του χρήστη στην εφαρμογή μέσω Google, προκειμένου να αποστείλει στο Picasa το περιεχόμενο που επιθυμεί μέσω της εφαρμογής. Καλείται όταν ο χρήστης επιλέξει να δημοσιεύσει περιεχόμενο στο Picasa και παρόλο του ότι είναι πολύ μικρό σε έκταση, παίζει καθοριστική σημασία γιατί είναι αυτό που συνδέει την εφαρμογή με τη βιβλιοθήκη Picasa PHP Client Library του Picasa.

Η βιβλιοθήκη Picasa PHP Client Library κάνει χρήση μιας μεγάλης βιβλιοθήκης, της βιβλιοθήκης Zend GData. Για να γίνει το οποιοδήποτε API call, θα πρέπει να δημιουργηθεί ένα αντικείμενο της κλάσης `Zend_Gdata_Photos` το οποίο στη συνάρτηση δημιουργίας του δέχεται ένα όρισμα, το `gdClient`. Το απαραίτητο αυτό αντικείμενο, δημιουργείται στο αρχείο `retrieving.php`.

```

$client = Zend_Gdata_AuthSub::getHttpClient($_SESSION['sessionToken']);

$gp = new Zend_Gdata_Photos($client, "Google-DevelopersGuide-1.0");

```

Δημιουργία ενός αντικειμένου της κλάσης `Zend_Gdata_Photos`

Στη συνέχεια, η εφαρμογή ελέγχει εάν κάποιος χρήστης είναι ήδη συνδεδεμένος με το Google ή όχι. Και στις δύο περιπτώσεις, θα πρέπει ο χρήστης να εξουσιοδοτήσει την εφαρμογή, ώστε να μπορεί αυτή να έχει πρόσβαση στο πλήθος των comments και στο περιεχόμενο των comments των χρηστών που έχουν αλληλεπιδράσει με το κείμενο μεγάλου μήκους που ανέβασε.

Παρακάτω παρουσιάζεται πως γίνεται η ανάκτηση αυτών των στοιχείων από το Picasa προγραμματιστικά.

```
$link = mysql_connect('localhost', 'ptyxiaki_vasi', '34fg90j')
    or die('Could not connect to database: ' . mysql_error());

mysql_select_db('ptyxiaki_vasi') or die('Could not select database');

mysql_query("SET NAMES 'utf8'");

$picasa_retrieve = mysql_query("SELECT * FROM picasa_posting WHERE
campaign_id = '$selectedCampaign'");

$row = mysql_fetch_array($picasa_retrieve, MYSQL_ASSOC);

$album_id_picasa = $row['album_id'];
$photo_id_picasa = $row['id_picasa'];

$query = $gp->newPhotoQuery();

$username = "default";

$query->setUser($username);

$query->setPhotoId($photo_id_picasa);

$query->setAlbumID($album_id_picasa);

    $query->setKind("comment");

    $query->setMaxResults("10000");

    $photoFeed = $gp->getPhotoFeed($query);

$count=0;

foreach ($photoFeed as $commentEntry) {
```

```

        $comment_text = $commentEntry->content->text;

        mysql_query("INSERT INTO google_comments
(campaign_id, type_google, comment, social_media) VALUES ('$selectedCampaign',
'photo', '$comment_text', 'picasa')");

        $count++;

    }

mysql_query("INSERT INTO picasa_retrieving (id_picasa, campaign_id, type_picasa,
no_of_comments) VALUES ('$photo_id_picasa', '$selectedCampaign', 'photo',
'$count')");

mysql_close($link);

```

Ανάκτηση πλήθους και περιεχόμενου comments και στη συνέχεια αποθήκευσή τους στη βάση δεδομένων

	id_pr	id_picasa	campaign_id	type_picasa	no_of_comments
<input type="checkbox"/> Edit Inline Edit Copy Delete	8	5653771534275434242	690	photo	2
<input type="checkbox"/> Edit Inline Edit Copy Delete	9	5658165421728726450	848	photo	0

Εγγραφές στη βάση δεδομένων με το πλήθος των comments όπου έγιναν στη φωτογραφία

<input type="checkbox"/> Edit Inline Edit Copy Delete	690	photo	δοκιμαστικό	picasa
<input type="checkbox"/> Edit Inline Edit Copy Delete	690	photo	testarw twra to api	picasa

Εγγραφές στη βάση δεδομένων με comments όπου έγιναν στη φωτογραφία

6.3 Visualization

Για την οπτικοποίηση της συλλογής των δεδομένων των καμπανιών που δημιουργούνται, χρησιμοποιήθηκαν τα Google charts. Αποτελούν charts τα οποία δημιουργούνται με javascript στην HTML σελίδα την οποία παρουσιάζονται και χρησιμοποιώντας το Google chart API μπορούμε να δημιουργήσουμε διαγράμματα εύκολα, γρήγορα και κυρίως αποδοτικά.

Στην εφαρμογή η οποία αναπτύχθηκε στα πλαίσια της παρούσης διπλωματικής εργασίας, τα δεδομένα τα οποία έχουν ανακτηθεί από τη διαδικασία του retrieving, μετατρέπονται σε JSON μορφή και αποστέλλονται στον κώδικα ο οποίος δημιουργεί το κάθε chart.

Παρακάτω, θα παρουσιαστούν τα php αρχεία τα οποία ανατρέχουν στη βάση δεδομένων, ανακτούν τα στοιχεία τα οποία έχουν συλλεχθεί και τα μετατρέπουν στη μορφή JSON, οι κώδικες javascript οι οποίοι δημιουργούν τα charts και τα τελικά charts τα οποία παράγονται.

Τα είδη των Google charts τα οποία χρησιμοποιήθηκαν για την παρουσίαση των δεδομένων τα οποία συνέλεξε η εφαρμογή ήταν τα παρακάτω :

- Column Chart
- Pie Chart
- Bar Chart
- Word Cloud
- Table
- Geochart
- Map

Διαδικασία δημιουργίας Column Chart

Παρακάτω παρουσιάζεται η διαδικασία δημιουργίας του Column chart το οποίο παρουσιάζει τη μέση ηλικία όλων των χρηστών που έχουν αλληλεπιδράσει με τα στοιχεία της καμπάνιας στο σύνολό τους.

Τα δεδομένα τα οποία αναπαριστά το συγκεκριμένο chart ανακτώνται από τη βάση δεδομένων από το αρχείο data_age.php

```
$selectedCampaign = $_SESSION['retrievingid'];
$con = mysql_connect("localhost","ptyxiaki_vasi","34fg90j");
mysql_select_db("ptyxiaki_vasi");
mysql_query("SET NAMES 'utf8'");
$counter = 0;
$age = 0;
$av_age = 0;
$av_age_male = 0;
$av_age_female = 0;
$av_age_unknown = 0;

//EMFANISI TOU AGE GIA OLI TIN KAMPANIA

$result = mysql_query("SELECT age FROM total_final_users WHERE
campaign_id = $selectedCampaign");

while ($row = mysql_fetch_array($result)) {
    if($row[0] != 0){
        $age = $age + $row[0];
        $counter++;
    }
}

$av_age = $age / $counter;
$counter = 0;
```

```
$age = 0;

//EMFANISI TOU AGE GIA GENDER MALE

$result = mysql_query("SELECT age FROM total_final_users WHERE
(campaign_id = $selectedCampaign && (gender = 'male' || gender = 'm'))");

while ($row = mysql_fetch_array($result)) {

    if($row[0] != 0){

        $age = $age + $row[0];

        $counter++;

    }

    $sav_age_male = $age / $counter;

    $counter = 0;

    $age = 0;

//EMFANISI TOU AGE GIA GENDER FEMALE

$result = mysql_query("SELECT age FROM total_final_users WHERE
(campaign_id = $selectedCampaign && (gender = 'female' || gender = 'f'))");

while ($row = mysql_fetch_array($result)) {

    if($row[0] != 0){

        $age = $age + $row[0];

        $counter++;

    }

    $sav_age_female = $age / $counter;

    $counter = 0;

    $age = 0;

//EMFANISI TOU AGE GIA GENDER UNKNOWN
```

```

$result = mysql_query("SELECT age FROM total_final_users WHERE
(campaign_id = $selectedCampaign && gender = 'unknown')");

while ($row = mysql_fetch_array($result)) {

    if($row[0] != 0){

        $age = $age + $row[0];

        $counter++;

    }

    $sav_age_unknown = $age / $counter;

    $counter = 0;

    $age = 0;

echo '{"result":["Total Campaign", "" . $sav_age . "", "Male", "" . $sav_age_male .
"", "Female", "" . $sav_age_female . "", "Unknown", "" . $sav_age_unknown .
""], "rows":4}';

mysql_close($con);

```

Ο κώδικας του αρχείου data_age.php

Στη συνέχεια, ο javascript κώδικας ο οποίος δημιουργεί το column chart, καλεί το αρχείο data_age.php και δημιουργεί το Column chart.

```

<script type="text/javascript">

//demographics - average age - column chart

google.load("visualization", "1", {packages:["corechart"]});

google.setOnLoadCallback(drawChart);

function drawChart() {

    $.ajax({

        type: "POST",

        url: 'data_age.php',

        success: function(data) {

```

```
var obj = jQuery.parseJSON(data);

var data = new google.visualization.DataTable();

data.addColumn('string', 'type');

data.addColumn('number', 'Average Age - General');

data.addRows(obj.rows);

for(i=0,j=0;i<obj.rows;i++){

                                                                    data.setValue(i, 0,
obj.result[j++]);
                                                                    data.setValue(i, 1,
Number(obj.result[j++]));
                                                                    }

var chart = new
google.visualization.ColumnChart(document.getElementById('agecolumn'));

chart.draw(data, {width: 800, height: 440, title: 'Demographics - Average
Age', colors: ['red', '#004411'],

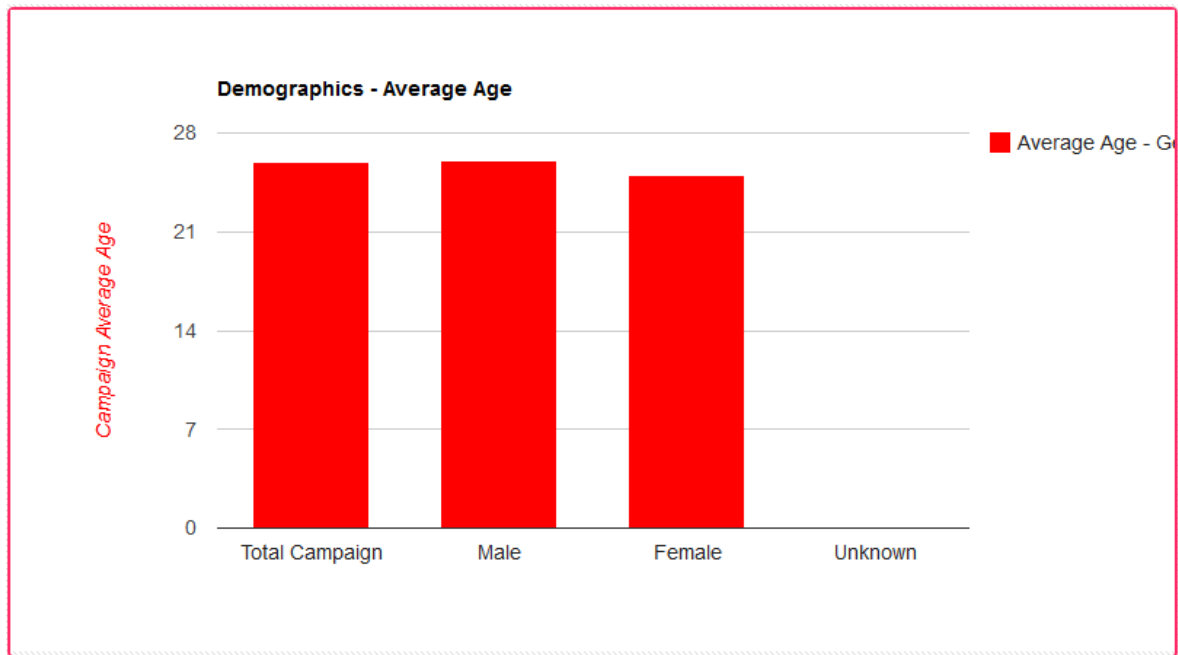
vAxis: {title: 'Campaign Average Age', titleTextStyle: {color: 'red'}}

});

}

</script>
```

Ο javascript κώδικας ο οποίος δημιουργεί το Column chart καλώντας το αρχείο data_age.php



6.2. - Το τελικό Chart το οποίο παράγεται - average age

Διαδικασία δημιουργίας Pie Chart

Παρακάτω παρουσιάζεται η διαδικασία δημιουργίας του Pie chart το οποίο παρουσιάζει τα likes τα οποία έγιναν σε περιεχόμενο κειμένου και σε περιεχόμενο video.

Τα δεδομένα τα οποία αναπαριστά το συγκεκριμένο chart ανακτώνται από τη βάση δεδομένων από το αρχείο data_campaign_likes.php

```

$selectedCampaign = $_SESSION['retrievingid'];
$con = mysql_connect("localhost","ptyxiaki_vasi","34fg90j");
mysql_select_db("ptyxiaki_vasi");
mysql_query("SET NAMES 'utf8'");
$no_of_text = 0;
$no_of_video = 0;
$no_of_photo = 0;

```

```
//ΕΜΦΑΝΙΣΗ ΤΟΥ ΠΛΗΘΟΥΣ ΤΩΝ LIKES ΓΙΑ ΟΛΗ ΤΙΝ ΚΑΜΠΑΝΙΑ ΑΝΑ POSTING_TYPE

$result = mysql_query("SELECT no_of_likes FROM total_retrieving WHERE
(campaign_id = $selectedCampaign && (type_post = 'network_update' || type_post
= 'status_update' || type_post = 'note' || type_post = 'blog-post' || type_post =
'tweet'))");

while ($row = mysql_fetch_array($result)) {

    $no_of_text = $no_of_text + $row[0];

}

$result = mysql_query("SELECT no_of_likes FROM total_retrieving WHERE
(campaign_id = $selectedCampaign && (type_post = 'photo'))");

while ($row = mysql_fetch_array($result)) {

    $no_of_photo = $no_of_photo + $row[0];

}

$result = mysql_query("SELECT no_of_likes FROM total_retrieving WHERE
(campaign_id = $selectedCampaign && (type_post = 'video'))");

while ($row = mysql_fetch_array($result)) {

    $no_of_video = $no_of_video + $row[0];

}

echo '{"result":["Text-Likes", "" . $no_of_text . "" , "Photo-Likes", "" .
$no_of_photo . "" , "Video-Likes", "" . $no_of_video . ""], "rows":3}';

mysql_close($con);
```

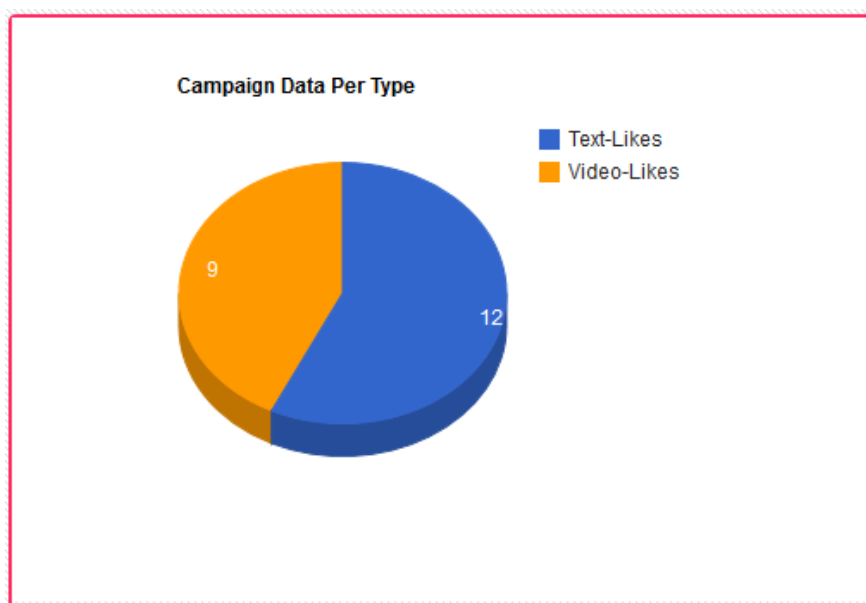
Ο κώδικας του αρχείου data_campaign_likes.php

Στη συνέχεια, ο javascript κώδικας ο οποίος δημιουργεί το Pie chart, καλεί το αρχείο data_campaign_likes.php και δημιουργεί το Pie chart.

```
<script type="text/javascript">
//ΕΜΦΑΝΙΣΗ ΤΟΥ ΠΛΗΘΟΥΣ ΤΩΝ ΛΙΚΕΣ ΓΙΑ ΟΛΗ ΤΙΝ ΚΑΜΠΑΝΙΑ ΑΝΑ POSTING_TYPE
google.load("visualization", "1", {packages:["corechart"]});
google.setOnLoadCallback(drawChart);
function drawChart() {
    $.ajax({
        type: "POST",
        url: 'data_campaign_likes.php',
        success: function(data) {
var obj = jQuery.parseJSON(data);
var data = new google.visualization.DataTable();
data.addColumn('string', 'type');
data.addColumn('number', 'Number');
data.addRows(obj.rows);
for(i=0,j=0;i<obj.rows;i++){
                                data.setValue(i, 0,
obj.result[j++]);
                                data.setValue(i, 1,
Number(obj.result[j++]));
                                }
                                var chart = new
google.visualization.PieChart(document.getElementById('div21'));
```

```
chart.draw(data, {width: 500,  
height:340, title: 'Campaign Data Per Type', legend: 'right', is3D:true,  
pieSliceText:'value'});  
  
}  
  
});  
  
}
```

Ο javascript κώδικας ο οποίος δημιουργεί το Pie chart καλώντας το αρχείο data_campaign_likes.php



6.3. - Το τελικό Chart το οποίο παράγεται - Likes per type

Διαδικασία δημιουργίας Bar Chart

Παρακάτω παρουσιάζεται η διαδικασία δημιουργίας του Bar chart το οποίο παρουσιάζει τα likes τα οποία έγιναν σε περιεχόμενο κειμένου και σε περιεχόμενο video.

Τα δεδομένα τα οποία αναπαριστά το συγκεκριμένο chart ανακτώνται από τη βάση δεδομένων από το αρχείο data_number_campaign.php.

```
$selectedCampaign = $_SESSION['retrievingid'];
$con = mysql_connect("localhost","ptyxiaki_vasi","34fg90j");

mysql_select_db("ptyxiaki_vasi");
mysql_query("SET NAMES 'utf8'");

$no_of_text = 0;
$no_of_video = 0;
$no_of_photo = 0;

//EMFANISI TOU PLITHOUS TWN LIKES GIA OLI TIN KAMPANIA ANA POSTING_TYPE

$result = mysql_query("SELECT no_of_likes FROM total_retrieving WHERE
(campaign_id = $selectedCampaign && (type_post = 'network_update' || type_post
= 'status_update' || type_post = 'note' || type_post = 'blog-post' || type_post =
'tweet'))");

while ($row = mysql_fetch_array($result)) {
    $no_of_text = $no_of_text + $row[0];
}

$result = mysql_query("SELECT no_of_likes FROM total_retrieving WHERE
(campaign_id = $selectedCampaign && (type_post = 'photo'))");

while ($row = mysql_fetch_array($result)) {
    $no_of_photo = $no_of_photo + $row[0];
}

$result = mysql_query("SELECT no_of_likes FROM total_retrieving WHERE
(campaign_id = $selectedCampaign && (type_post = 'video'))");

while ($row = mysql_fetch_array($result)) {
    $no_of_video = $no_of_video + $row[0];
}
```

```

        echo '{"result":["Text-Likes"," . $no_of_text . ","Photo-Likes"," .
        $no_of_photo . ","Video-Likes"," . $no_of_video . ""],"rows":3}';
mysql_close($con);

```

Ο κώδικας του αρχείου data_number_campaign.php

Στη συνέχεια, ο javascript κώδικας ο οποίος δημιουργεί το Pie chart, καλεί το αρχείο data_number_campaign.php και δημιουργεί το Pie chart.

```

<script type="text/javascript">

//EMFANIZEI GIA OLOKLIRI TIN KAMPANIA ARITHMITIKA

google.load("visualization", "1", {packages:["corechart"]});

google.setOnLoadCallback(drawChart);

function drawChart() {

    $.ajax({

        type: "POST",

        url: 'data_number_campaign.php',

        success: function(data) {

                                                    var obj =
jQuery.parseJSON(data);

                                                    var data = new
google.visualization.DataTable();

                data.addColumn('string', 'SM');

                data.addColumn('number', 'Number of Comments');

                data.addColumn('number', 'Number of Likes');

                data.addColumn('number', 'Number of Views');

//if trexei mono gia sinolika tis kampanias -> kanonika oi grammes

if(obj.rows == 1){

                data.addRows(obj.rows);

                for(i=0,j=0;i<obj.rows;i++){

```

```

data.setValue(i, 0,
obj.result[j++]);
    data.setValue(i, 1, Number(obj.result[j++]));

data.setValue(i, 2,
Number(obj.result[j++]));

data.setValue(i, 3,
Number(obj.result[j++]));
    }
}

//else trexei gia per social media -> mia grammi epipleon eksaitias tou
facebook

else{

data.addRows(obj.rows+1);

for(i=0,j=0;i<obj.rows+1;i++){

data.setValue(i, 0,
obj.result[j++]);

data.setValue(i, 1, Number(obj.result[j++]));

data.setValue(i, 2, Number(obj.result[j++]));

data.setValue(i, 3, Number(obj.result[j++]));

}
}

var chart = new google.visualization.BarChart(document.getElementById('div16'));
chart.draw(data, {width: 800, height: 640, title: 'Total Numbers General',

vAxis: {title: 'SocialMedia', titleTextStyle: {color: 'red'}}

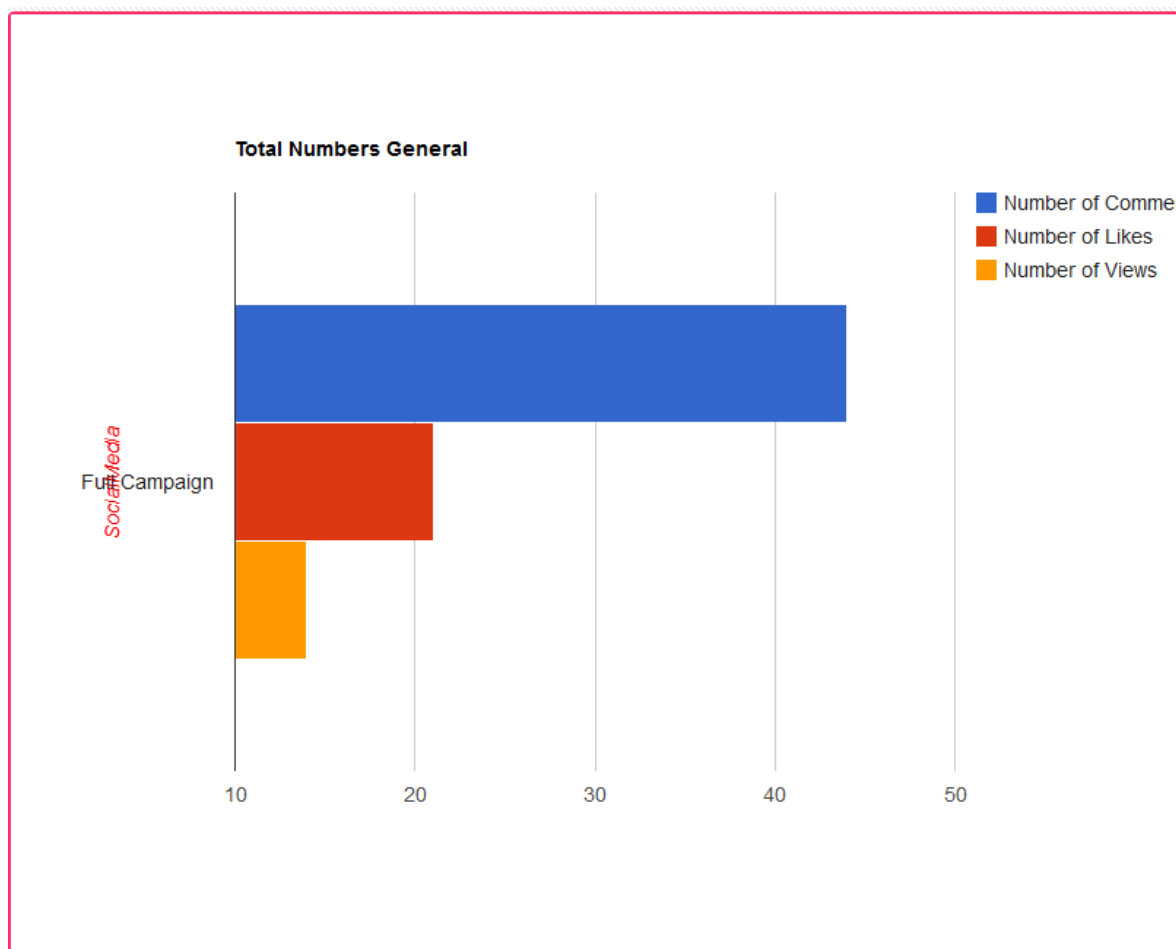
});

});

}
</script>

```

Ο javascript κώδικας ο οποίος δημιουργεί το Pie chart καλώντας το αρχείο `data_campaign_likes.php`



6.4. - Το τελικό Chart το οποίο παράγεται - αριθμητικά

Διαδικασία δημιουργίας Word cloud chart και table chart

Παρακάτω παρουσιάζεται η διαδικασία δημιουργίας των word cloud chart και table chart το οποίο παρουσιάζει τα comments που έχουν γίνει σε videos της καμπάνιας. Ο λόγος που παρουσιάζονται και τα δύο είδη charts μαζί είναι ότι στην εφαρμογή χρησιμοποιήθηκαν και τα δύο είδη μαζί για να παρουσιάσουν περιεχόμενο comments.

Τα δεδομένα τα οποία αναπαριστά το συγκεκριμένο chart ανακτώνται από τη βάση δεδομένων από το αρχείο data_comment_video.php

```
$selectedCampaign = $_SESSION['retrievingid'];  
  
$con = mysql_connect("localhost","ptyxiaki_vasi","34fg90j");  
  
    mysql_select_db("ptyxiaki_vasi");  
  
    mysql_query("SET NAMES 'utf8'");  
  
//EMFANISI OLWN TWN COMMENTS POY AFOROYN video (facebook kai youtube)  
  
    $result = mysql_query("SELECT comment FROM total_comments WHERE  
(campaign_id = $selectedCampaign && type = 'video')");  
  
    $x = 0;  
  
    $str = array();  
  
    while ($row = mysql_fetch_array($result)) {  
        array_push($str, $row['comment']);  
  
        $x++;  
    }  
  
    echo json_encode(array("result"=>$str,"rows"=>mysql_num_rows($result));  
  
mysql_close($con);
```

Ο κώδικας του αρχείου data_comment_video.php

Στη συνέχεια, ο javascript κώδικας ο οποίος δημιουργεί το column chart, καλεί το αρχείο data_comment_video.php και δημιουργεί το word cloud chart και το table chart.

```
<script type="text/javascript">

    //EMFANISI OLWN TWN COMMENTS POY AFOROYN video (facebook kai
youtube)

    google.load("visualization", "1",{packages: ['table']});

    google.setOnLoadCallback(draw);

    function draw() {

        $.ajax({

            type: "POST",

            url: 'data_comment_video.php',

            success: function(data) {

var obj = jQuery.parseJSON(data);

var data = new google.visualization.DataTable();

                data.addColumn('string', 'List of selected Comments for
facebook and youtube video');

data.addRows(obj.rows);

for(i=0;i<obj.result.length;i++){

                    data.setCell(i, 0, obj.result[i]);

                }

var outputDiv = document.getElementById('wcdiv9');

var wc = new WordCloud(outputDiv);

wc.draw(data, null);

wc.draw(data, {stopWords: 'a an and is or the of for to'});

table = new google.visualization.Table(document.getElementById('tablediv9'));
```

```
table.draw(data);
```

```
    }
```

```
});
```

```
}
```

```
</script>
```

Ο javascript κώδικας ο οποίος δημιουργεί το word cloud chart και το table chart καλώντας το αρχείο data_comment_video.php



6.5. - Τα τελικά Charts τα οποία παράγονται - word cloud

Διαδικασία δημιουργίας Geochart

Παρακάτω παρουσιάζεται η διαδικασία δημιουργίας του Geochart το οποίο παρουσιάζει την τοποθεσία των χρηστών που αλληλεπίδρασαν με το περιεχόμενο της καμπάνιας.

Τα δεδομένα τα οποία αναπαριστά το συγκεκριμένο chart ανακτώνται από τη βάση δεδομένων από το αρχείο data_geochart.php

```
$selectedCampaign = $_SESSION['retrievingid'];  
  
$con = mysql_connect("localhost","ptyxiaki_vasi","34fg90j");  
  
mysql_select_db("ptyxiaki_vasi");  
  
mysql_query("SET NAMES 'utf8'");  
  
$result = mysql_query("SELECT location FROM total_users WHERE  
campaign_id = $selectedCampaign");  
  
$x = 0;  
  
$str = array();  
  
while ($row = mysql_fetch_array($result)) {  
    if($row['location'] != 'unknown'){  
        $counter = 0;  
        $place = $row['location'];  
        $result_2 = mysql_query("SELECT location FROM total_users  
WHERE campaign_id = $selectedCampaign");  
        while ($row_2 = mysql_fetch_array($result_2)){  
            if($row_2['location'] == $place){  
                $counter++;  
            }  
        }  
        array_push($str, $row['location'], "$counter");  
        $x++;  
    }  
}
```

```
    }  
  }  
  
  echo json_encode(array("result"=>$str,"rows"=>$x));  
  
mysql_close($con);
```

Ο κώδικας του αρχείου data_geochart.php

Στη συνέχεια, ο javascript κώδικας ο οποίος δημιουργεί το Geochart, καλεί το αρχείο data_geochart.php και δημιουργεί το Geochart.

```
<script type="text/javascript">  
  //general map  
  google.load("visualization", "1", {packages:["geomap"]});  
  google.setOnLoadCallback(drawChart);  
  function drawChart() {  
    $.ajax({  
type: "POST",  
url: 'data_geochart.php',  
success: function(data) {  
var obj = jQuery.parseJSON(data);  
var data = new google.visualization.DataTable();  
data.addColumn('string', 'Location - General');  
data.addColumn('number', 'Number of Responses');  
data.addRows(obj.rows);  
for(i=0,j=0;i<obj.rows;i++){  
data.setValue(i, 0,  
obj.result[j++]);
```

```
        data.setValue(i, 1, Number(obj.result[j++]));
    }

    var options = {};
    options['dataMode'] = 'regions';
    options['width'] = '780px';
    options['height'] = '488px';

    var container = document.getElementById('map_canvas');
    var geomap = new google.visualization.GeoMap(container);
    geomap.draw(data, options);
    }
    });
}

</script>
```

Ο javascript κώδικας ο οποίος δημιουργεί το Geochart καλώντας το αρχείο data_geochart.php



6.6. - Το τελικό Chart το οποίο παράγεται - geochart

Διαδικασία δημιουργίας Map chart

Παρακάτω παρουσιάζεται η διαδικασία δημιουργίας του Map chart το οποίο παρουσιάζει την τοποθεσία των χρηστών που αλληλεπίδρασαν με το περιεχόμενο της καμπάνιας.

Τα δεδομένα τα οποία αναπαριστά το συγκεκριμένο chart ανακτώνται από τη βάση δεδομένων από το αρχείο data_locationt.php

```

$selectedCampaign = $_SESSION['retrievingid'];
$con = mysql_connect("localhost","ptyxiaki_vasi","34fg90j");
mysql_select_db("ptyxiaki_vasi");
mysql_query("SET NAMES 'utf8'");
$result = mysql_query("SELECT location FROM total_users WHERE
campaign_id = $selectedCampaign");
$x = 0;
$str = array();
while ($row = mysql_fetch_array($result)) {

```

```
if($row['location'] != 'unknown'){  
    array_push($str, $row['location']);  
    $x++;  
}  
  
echo json_encode(array("result"=>$str,"rows"=>$x));  
  
mysql_close($con);mysql_close($con);
```

Ο κώδικας του αρχείου data_location.php

Στη συνέχεια, ο javascript κώδικας ο οποίος δημιουργεί το Map chart, καλεί το αρχείο data_location.php και δημιουργεί το Map chart.

```
<script type="text/javascript">  
    //google map with tags  
  
    google.load("visualization", "1", {packages:["map"]});  
    google.setOnLoadCallback(drawChart);  
  
    function drawChart() {  
        $.ajax({  
            type: "POST",  
            url: 'data_location.php',  
            success: function(data) {  
  
var obj = jQuery.parseJSON(data);  
var data = new google.visualization.DataTable();  
            data.addColumn('string', 'Location');  
            data.addRows(obj.rows);  
            for(i=0;i<obj.rows;i++){  
                data.setValue(i, 0, obj.result[i]);  
            }  
        }  
    }  
}
```



```

    }

    map = new google.visualization.Map(document.getElementById('map_div'));
    map.draw(data, {showTip: true});

    }

    });

}

</script>

```

Ο javascript κώδικας ο οποίος δημιουργεί το Map chart καλώντας το αρχείο `data_location.php`



6.7. - Το τελικό Chart το οποίο παράγεται - map chart

Κεφάλαιο 7^ο - Εφαρμογή SocialCampaign

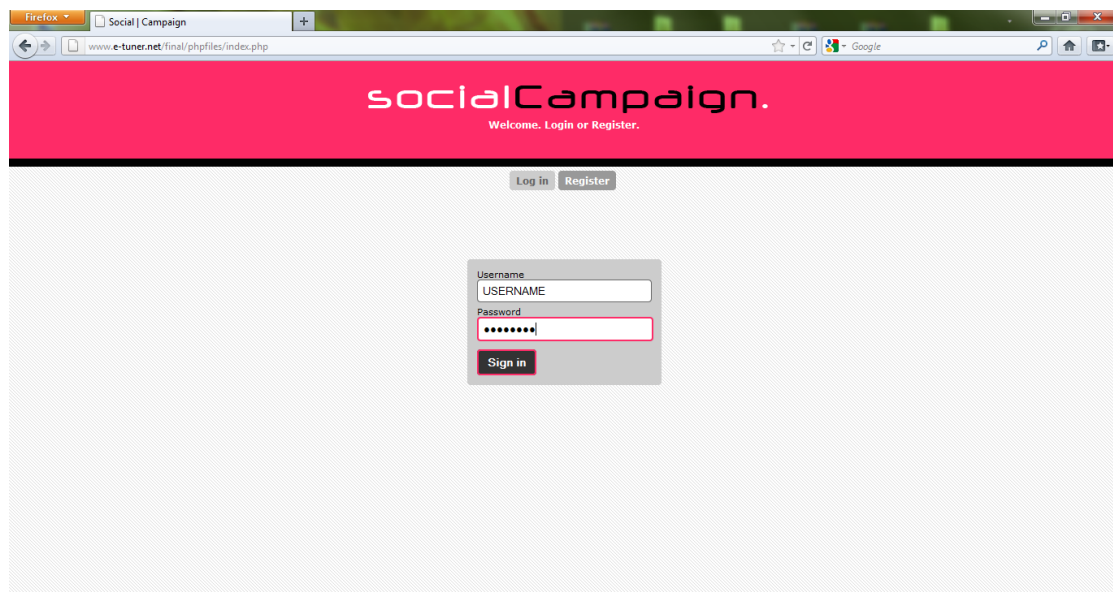
Σε αυτή την ενότητα θα πραγματοποιηθεί μια σύντομη παρουσίαση της εφαρμογής **SocialCampaign** που υλοποιήθηκε στα πλαίσια της παρούσης διπλωματικής εργασίας.

Σκοπός του περιεχομένου αυτής της ενότητας είναι απλώς να παρουσιαστεί το κομμάτι της διεπαφής της εφαρμογής και όχι να εκτιμηθεί η αποτελεσματικότητά του σε συνθήκες προώθησης κάποιας πραγματικής πολιτικής καμπάνιας.

Η ενότητα αυτή χωρίζεται σε δύο διακριτά τμήματα, το πρώτο για τη διαδικασία της δημιουργίας καμπάνιας και το δεύτερο για τη διαδικασία παρουσίασης των αποτελεσμάτων μιας επιλεγμένης από το χρήστη καμπάνιας.

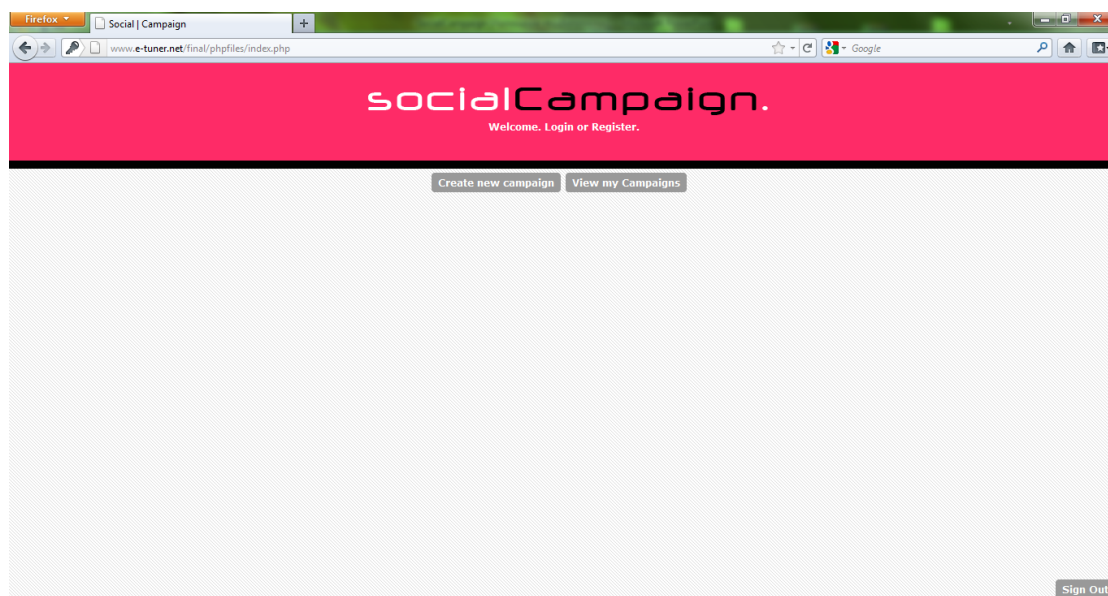
7.1. - Publishing

Στην αρχική σελίδα της εφαρμογής ο χρήστης μπορεί να εγγραφεί ή αν έχει ήδη λογαριασμό απλά να εισάγει τα στοιχεία του για να αποκτήσει πρόσβαση.

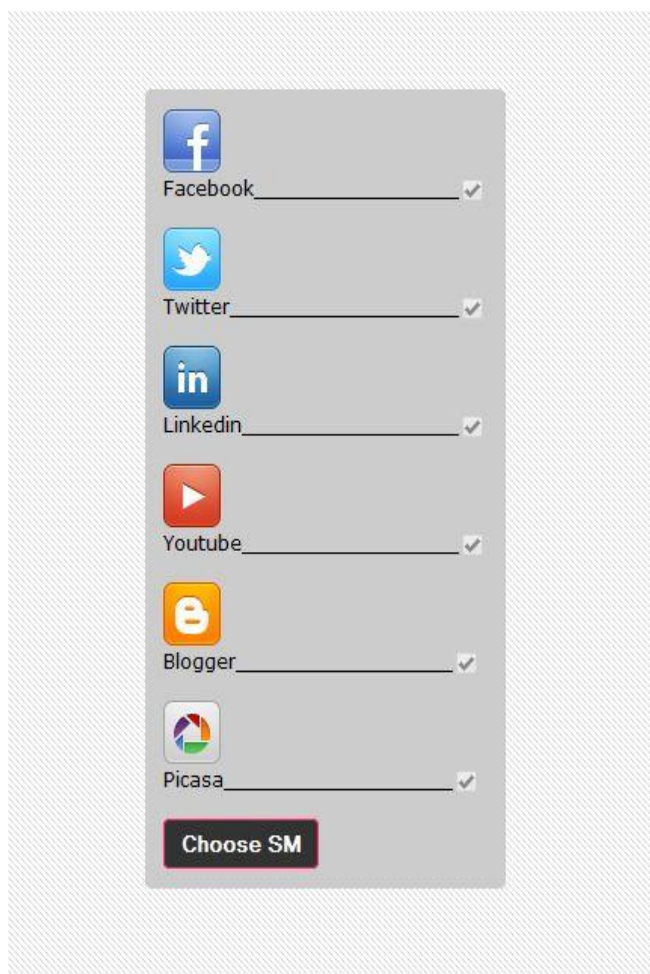


7.1. - Signing in

Αφού ο χρήστης αποκτήσει πρόσβαση στην εφαρμογή τότε μπορεί να επιλέξει μεταξύ δημιουργίας νέας καμπάνιας ή παρουσίασης υπάρχουσας καμπάνιας. Σε αυτό το βήμα θα επιλέξουμε δημιουργία νέας καμπάνιας.



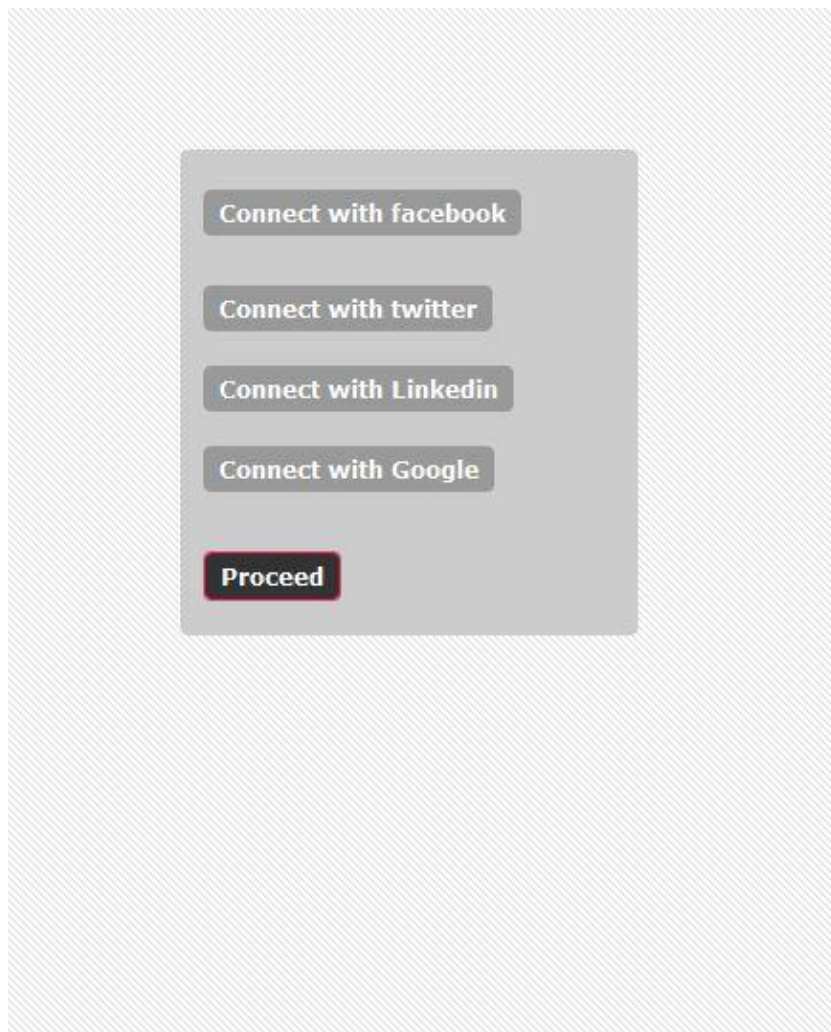
7.2. - Δημιουργία καμπάνιας



7.3. - Επιλογή των SM

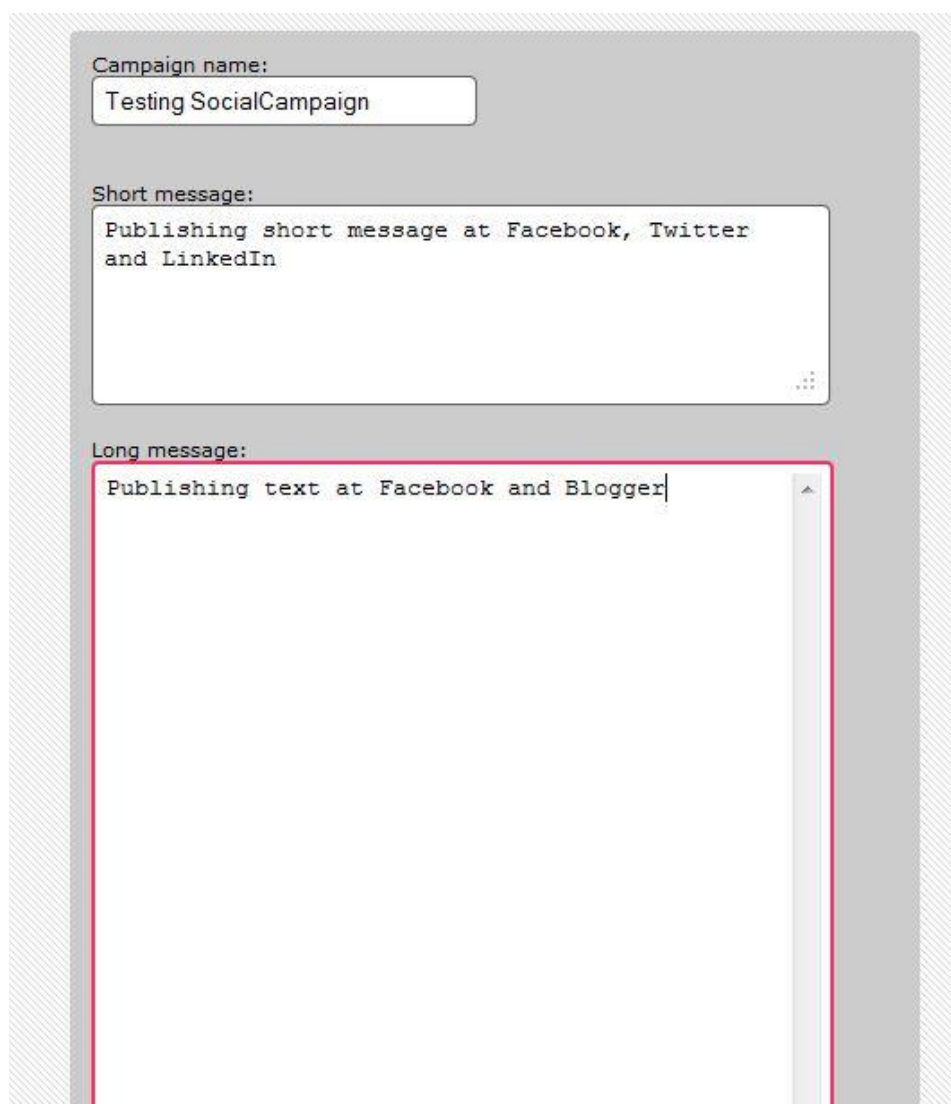
Ο χρήστης αφού επιλέξει τη δημιουργία νέας καμπάνιας μεταβαίνει στη φόρμα επιλογής των social media στα οποία επιθυμεί να δημοσιευτεί το περιεχόμενό του.

Στην αμέσως επόμενη οθόνη πρέπει να προχωρήσει σε αυθεντικοποίηση των προσωπικών του λογαριασμών στα social media ώστε η εφαρμογή να αποκτήσει πρόσβαση στις δυνατότητες και στα δεδομένα του κάθε λογαριασμού του στα social media ξεχωριστά.



7.4. - Αυθεντικοποίηση

Στη συνέχεια εμφανίζεται η φόρμα στην οποία ο χρήστης δίνει το όνομα της καμπάνιας του. Ανάλογα με τα social media που έχει επιλέξει εμφανίζονται τα κατάλληλα πεδία. Σε μια καμπάνια με όλα τα social media θα έχουμε πεδίο για μικρό κείμενο, πεδίο για μεγάλο κείμενο και φόρτωση φωτογραφίας και βίντεο.



The image shows a screenshot of a web application interface for creating a social media campaign. It features three distinct text input areas:

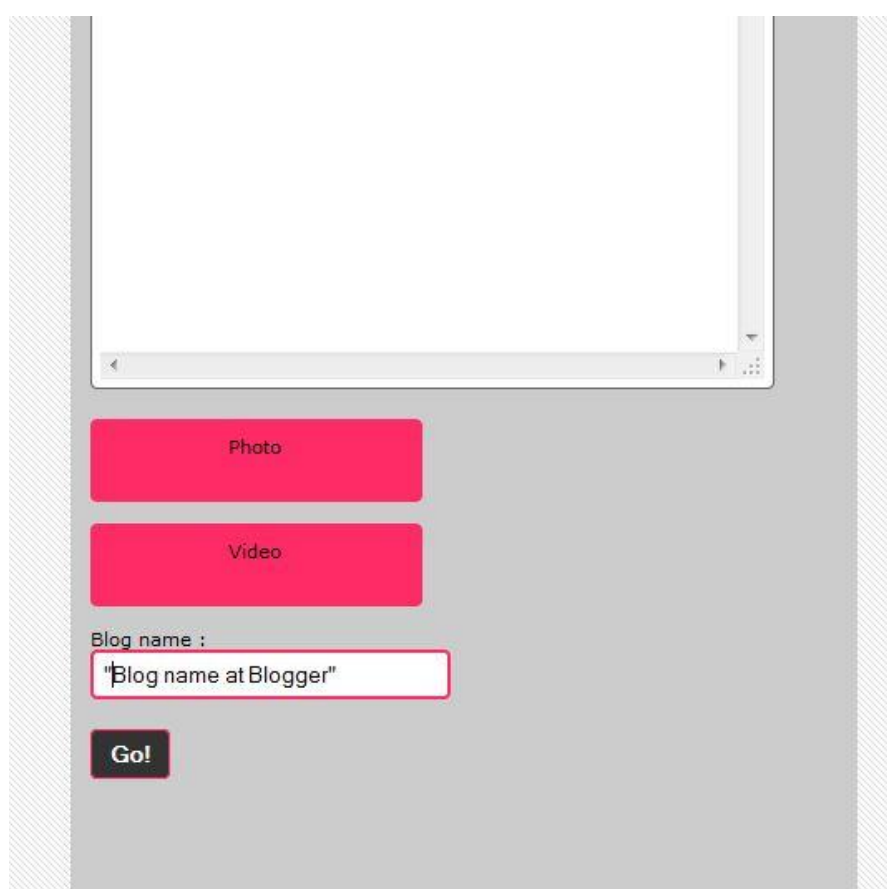
- Campaign name:** A single-line text box containing the text "Testing SocialCampaign".
- Short message:** A larger text box containing the text "Publishing short message at Facebook, Twitter and LinkedIn".
- Long message:** A very large text box containing the text "Publishing text at Facebook and Blogger".

7.5. - Πεδία κειμένου

Στο πρώτο πεδίο ο χρήστης δίνει το όνομα της καμπάνιας του. Αυτό το όνομα θα είναι τόσο το αναγνωριστικό για την εφαρμογή αλλά επίσης θα είναι και το όνομα το νέων αντικειμένων που θα δημιουργηθούν στα social media (π.χ. το όνομα του άλμπουμ στο picasa). Το πεδίο του μικρού κειμένου αναφέρεται στο *status update* του Facebook, στο *tweet* του Twitter και στο *status update* του LinkedIn. Το πεδίο του μεγάλου κειμένου αναφέρεται στη δημιουργία νέου *note* στο Facebook και στη δημιουργία νέου *post* σε blog του Blogger.

Στη συνέχεια της φόρμας ο χρήστης έχει τη δυνατότητα να ανεβάσει τόσο φωτογραφίες όσο και βίντεο.

Η φωτογραφία θα φορτωθεί στο Facebook και στο Picasa. Στο Facebook την πρώτη φορά χρήσης του εργαλείου θα δημιουργηθεί ένα άλμπουμ φωτογραφιών με το όνομα SocialCampaign όπου εκεί μέσα θα αποθηκεύονται οι φωτογραφίες με το όνομα της καμπάνιας. Στο Picasa θα δημιουργείται νέο άλμπουμ με το το όνομα της καμπάνιας όπου εκεί θα αποθηκεύεται η φωτογραφία επίσης με το όνομα της καμπάνιας.

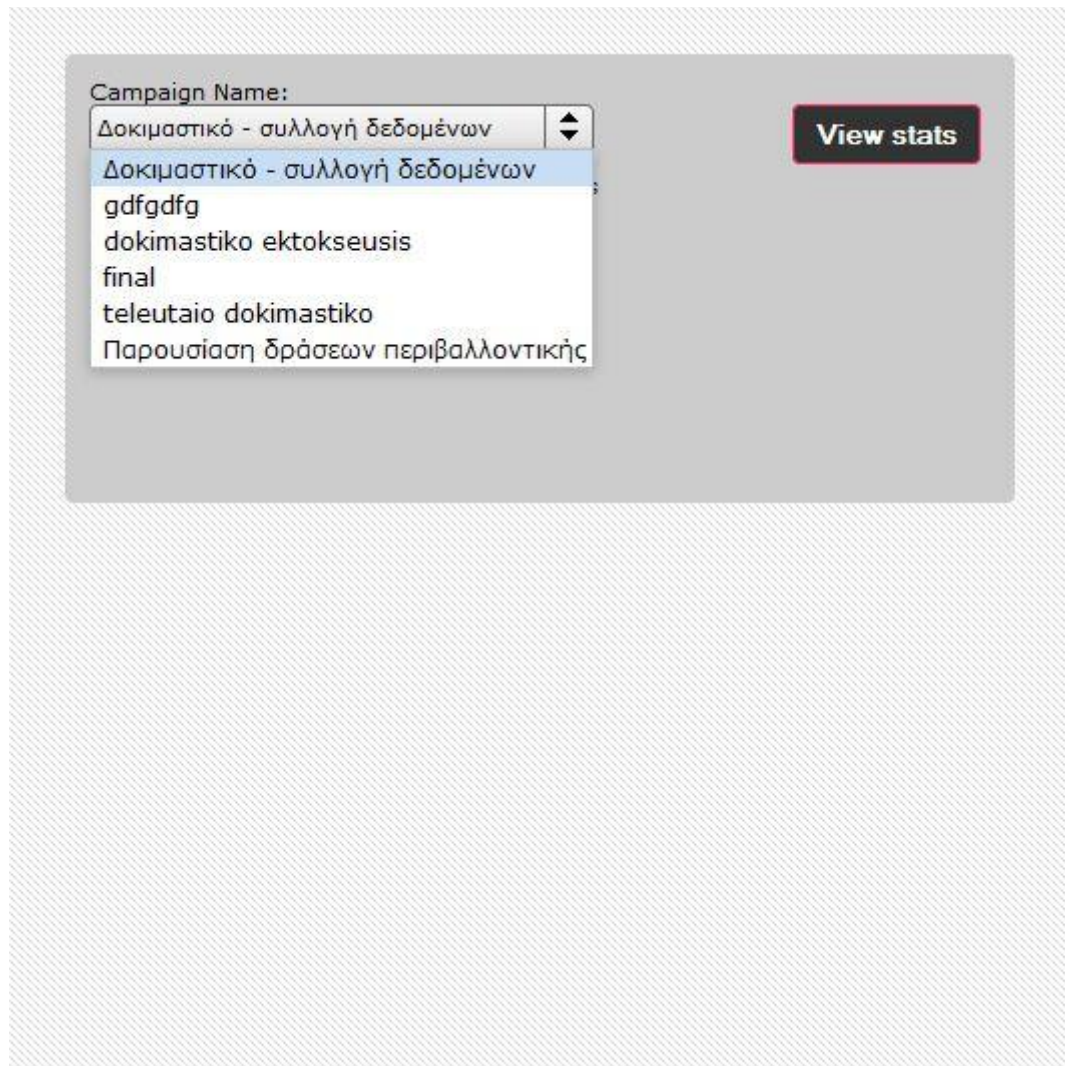


7.6. - Πολυμέσα

Στο τελευταίο πεδίο ο χρήστης πρέπει να εισάγει το όνομα του blog του στο Blogger καθώς εκεί μέσα θα δημιουργηθεί το νέο post.

7.2. - Visualization

Αν ο χρήστης επιλέξει αντί για δημιουργία νέας καμπάνιας την παρουσίαση υπάρχουσας καμπάνιας θα του εμφανίσει μια λίστα με όλες τις καμπάνιες που έχει δημιουργήσει. Εκεί ο χρήστης επιλέγει την επιθυμητή καμπάνια και η εφαρμογή ξεκινάει τη συλλογή των δεδομένων που αφορούν την καμπάνια από όλα τα social media.



7.7. - Επιλογή καμπάνιας για παρουσίαση

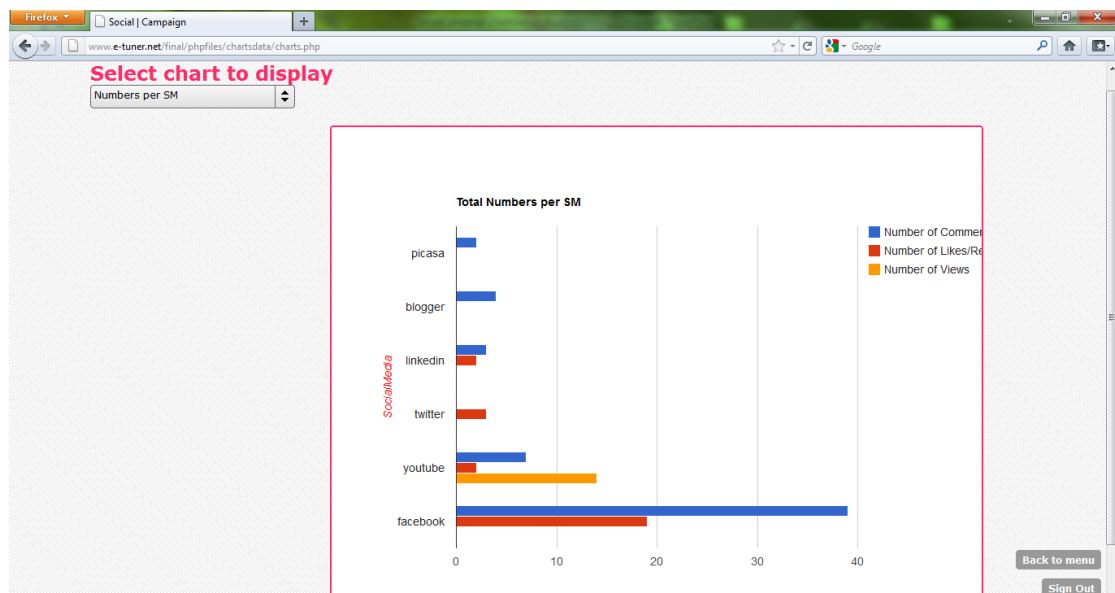
Στη συνέχεια η εφαρμογή θα παρουσιάσει τα αποτελέσματα της αλληλεπίδρασης των περιεχομένων της καμπάνιας στα social media. Παρακάτω παρουσιάζονται όλοι οι τύποι των γραφημάτων που χρησιμοποιούνται αλλά όχι όλες οι όψεις καθώς παρέχονται πολλές δυνατότητες φιλτραρίσματος με προκαθορισμένα κριτήρια.

Στο παρακάτω γράφημα παρουσιάζεται το πλήθος των comments και των likes που αφορούν όλη την καμπάνια. Επίσης εμφανίζεται και το πλήθος των views που αφορά μόνο το YouTube.



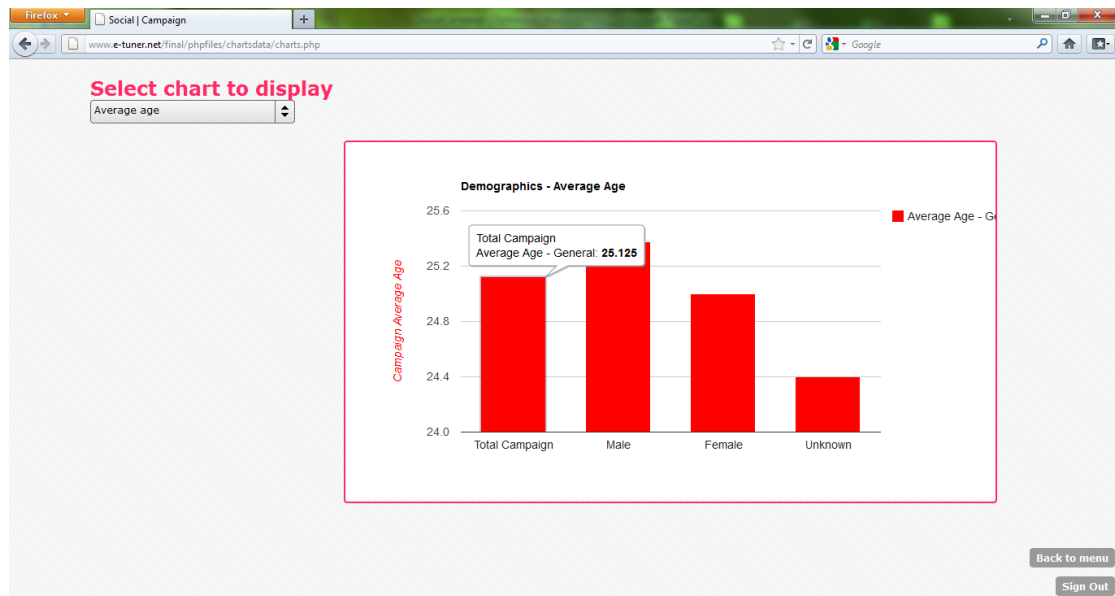
7.8. - General numbers

Στη συνέχεια εμφανίζονται για κάθε social media ξεχωριστά.



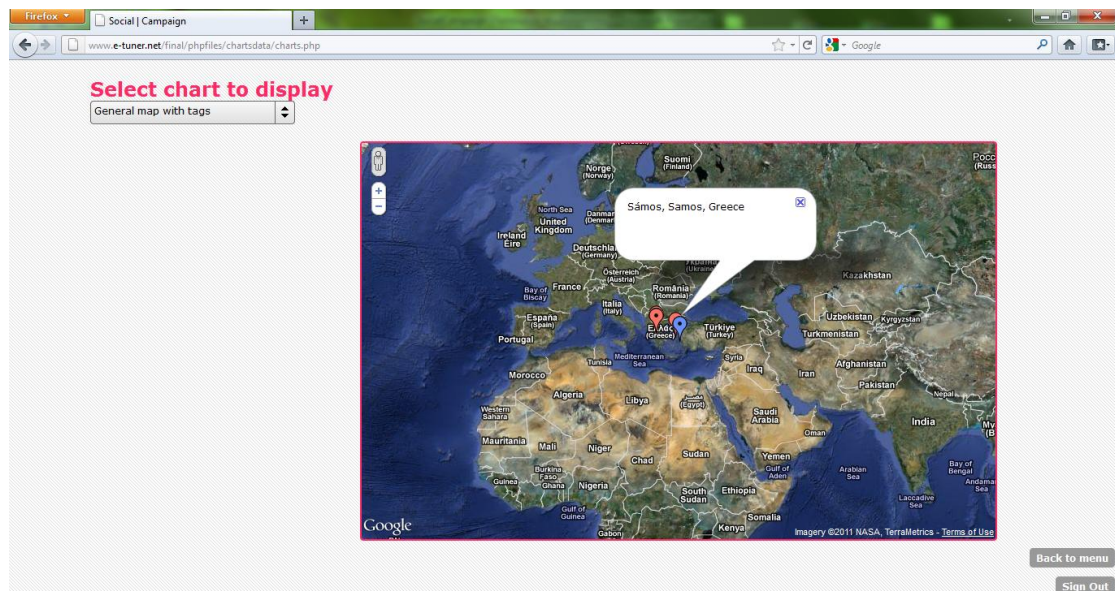
7.9. - Numbers per SM

Μέση ηλικία για όλη την καμπάνια αλλά και για male - female.



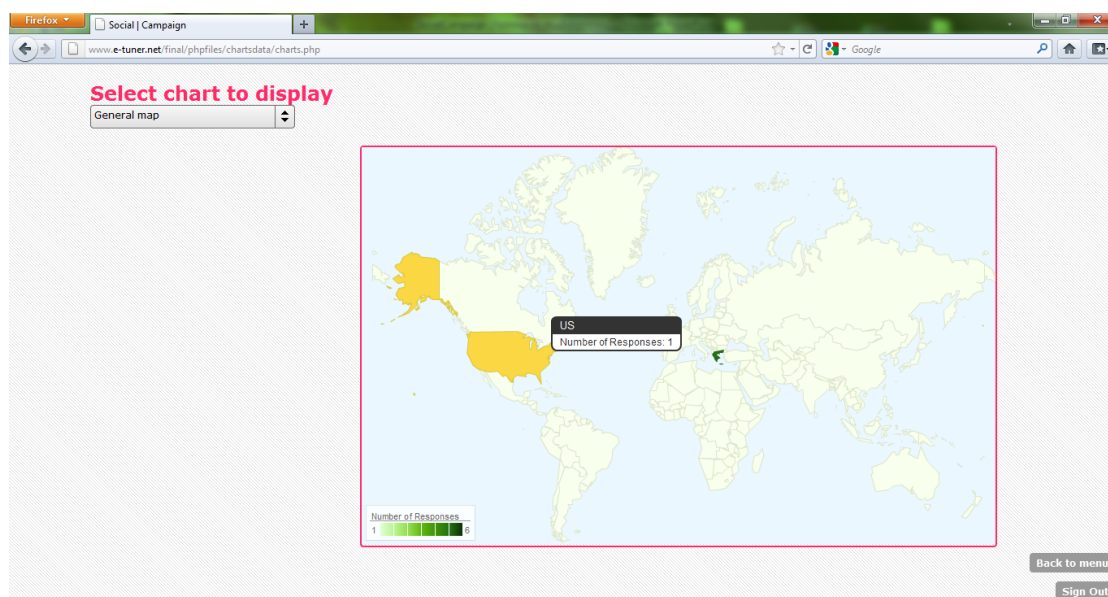
7.10. - Average age

Παρουσίαση με tags σε χάρτη τα σημεία αλληλεπίδρασης με την καμπάνια.



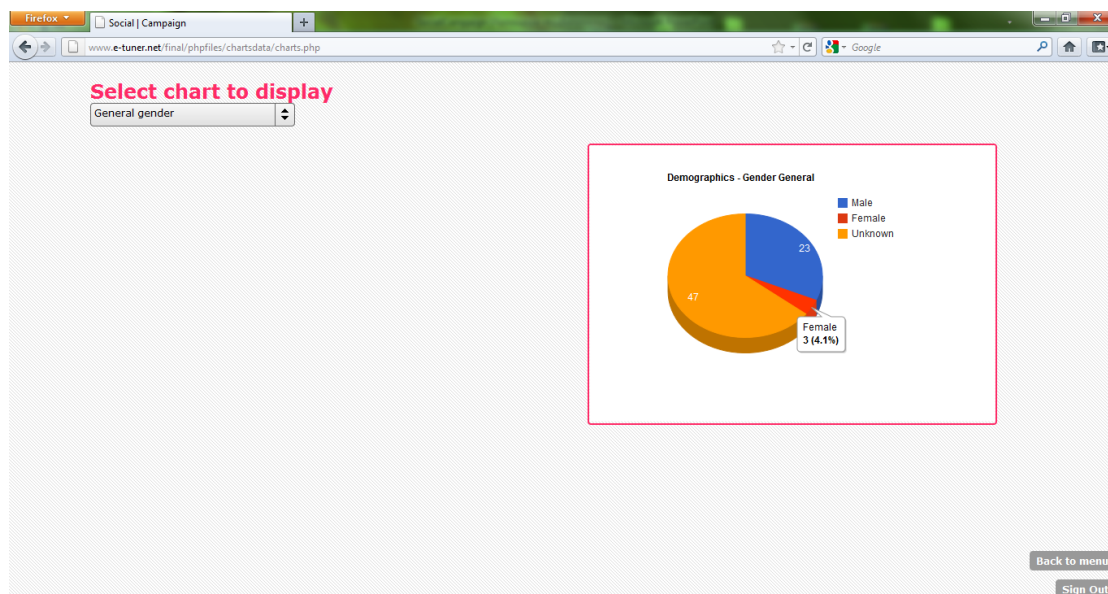
7.11. - Map with tags

Στη συνέχεια εμφανίζεται η πυκνότητα της αλληλεπίδρασης ανά χώρα ξεχωριστά.



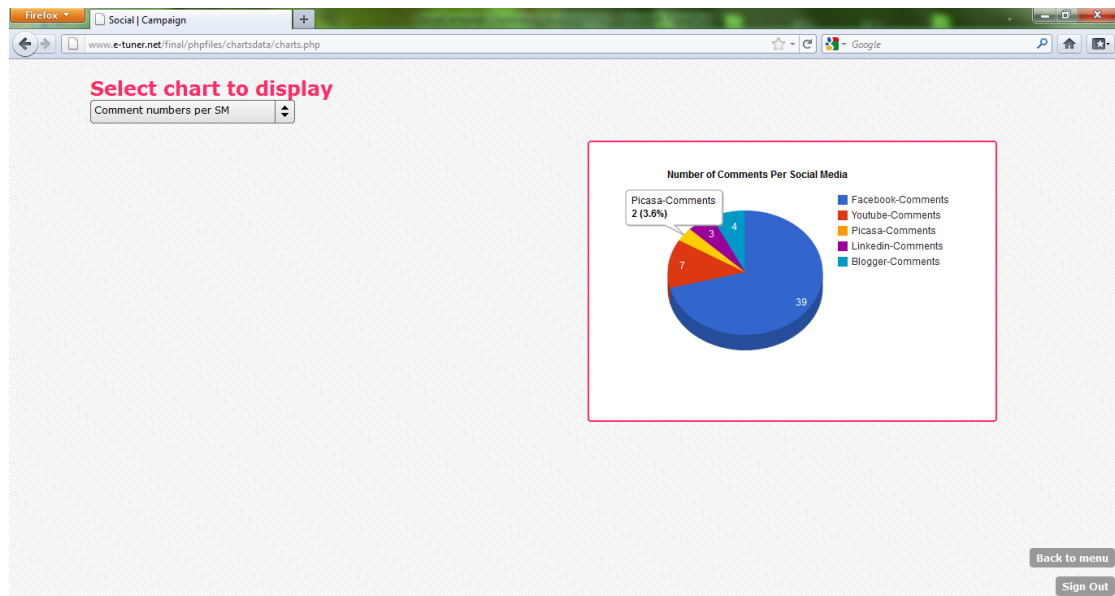
7.12. - Map per country

Στη συνέχεια έχουμε pie chart με δημογραφικά στοιχεία που αφορούν το γένος.



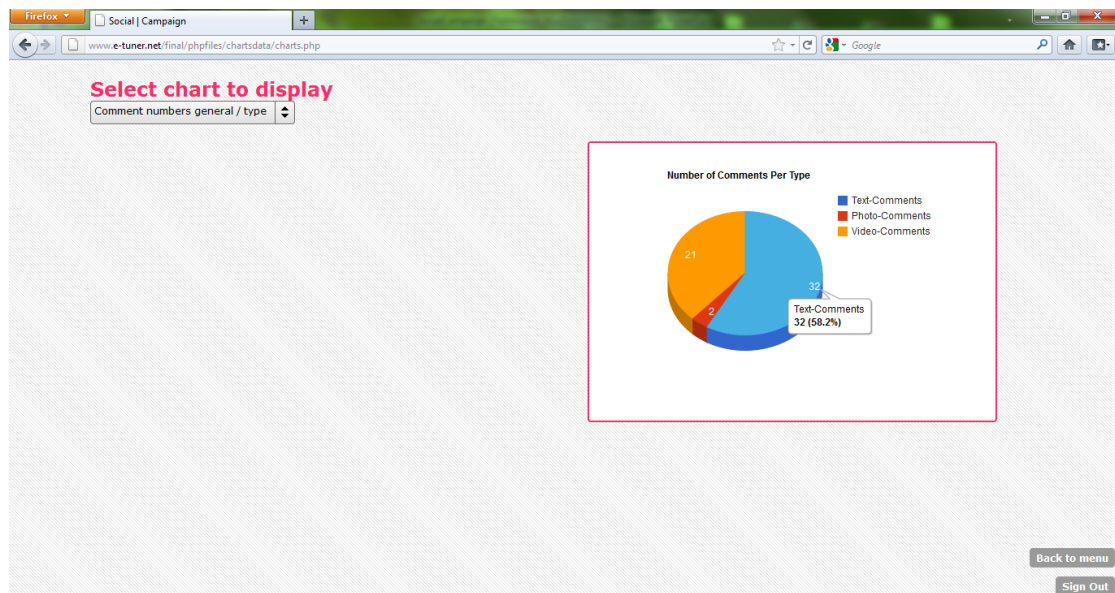
7.13. - General gender

Στη συνέχεια έχουμε το πλήθος των comments ανά social media.



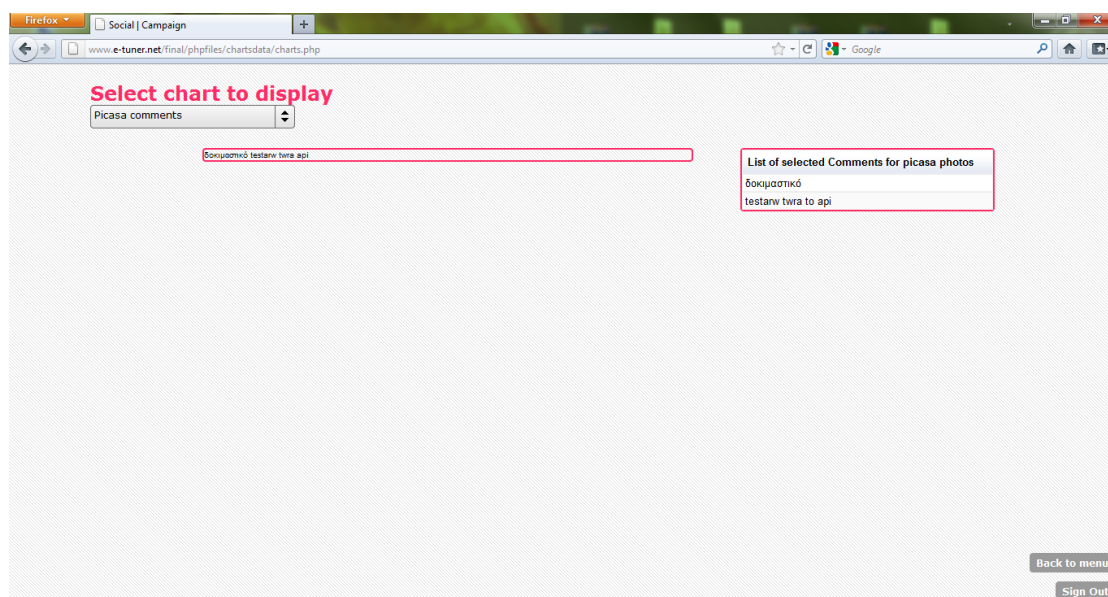
7.14. - #Comments per SM

Στη συνέχεια έχουμε το πλήθος των likes ανά τύπο περιεχομένου.



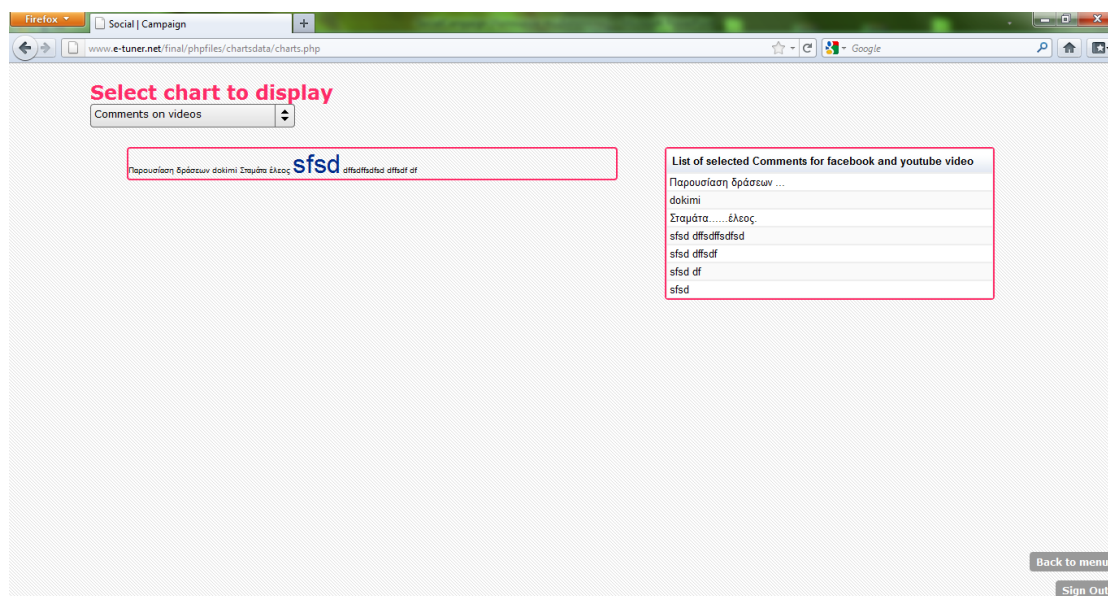
7.15. - #Likes per type

Στη συνέχεια έχουμε την εμφάνιση των comments ανά social media, εδώ picasa.



7.16. - Comments Picasa

Τέλος έχουμε την εμφάνιση των comments ανά τύπο περιεχομένου, εδώ έχουμε για βίντεο, comments από facebook - βίντεο και από youtube.



7.17. - Comments on video

Κεφάλαιο 8^ο - Συμπεράσματα

Η παρούσα εργασία επιχειρεί αρχικά να δώσει μια γενική εικόνα των διασυνδέσεων που εφαρμόζουν οι πλατφόρμες των μέσων κοινωνικής δικτύωσης με τον «έξω κόσμο», δηλαδή να μελετήσει και εν τέλει να χρησιμοποιήσει τα API τους. Μέσω της προσφερόμενης τεκμηρίωσης και της προσεκτικής μελέτης των APIs προχωρούμε σε μια πρώτη πιλοτική προσπάθεια υλοποίησης μιας εφαρμογής που θα συνεργάζεται και θα χρησιμοποιεί τις δυνατότητες των μέσων κοινωνικής δικτύωσης έτσι ώστε να αποκτηθεί μια πρώτη βασική εμπειρία και γνώση που θα ανοίξει το δρόμο για την υλοποίηση πλήρως λειτουργικών εφαρμογών που θα βασίζονται στο μοντέλο του Padgets.

Προχωρήσαμε σε βασική αξιολόγηση της εφαρμογής που δημιουργήθηκε στα πλαίσια της εργασίας. Μια περαιτέρω συστηματική αξιολόγηση του εργαλείου με βάση μια οργανωμένη μεθοδολογία θα αποτελέσει αντικείμενο μελλοντικής μελέτης. Σε αυτό το στάδιο η χρήση της εφαρμογής είχε ως σκοπό τον έλεγχο της ορθής συνεργασίας της με τα API των έξι social media που χρησιμοποιεί.

Παρακάτω ακολουθούν κάποια από τα βασικότερα σημεία στα οποία κατέληξε η μελέτη που πραγματοποιήθηκε μέσα από τη διαδικασία της εξερεύνησης των προσφερόμενων APIs μερικών εκ των βασικότερων social media αλλά και της χρησιμοποίησης δημοφιλών τεχνολογιών υλοποίησης που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής.

Μια από τις σημαντικότερες δυσκολίες και, ταυτόχρονα, ένα σημαντικό στοιχείο των APIs των μέσων κοινωνικής δικτύωσης, είναι η «ρευστότητά» τους. Οι περισσότερες πλατφόρμες είναι σχετικά νέες και οι διεπαφές προγραμματισμού τους δεν έχουν ακόμη φτάσει σε ικανοποιητικό βαθμό λειτουργικότητας. Ταυτόχρονα, προσπαθούν κυρίως να συμμορφώνονται καλύτερα με τις πραγματικές ή τουλάχιστον αντιληπτές ανάγκες των χρηστών τους και με τις απαιτήσεις των προγραμματιστών τους. Από τεχνική σκοπιά τα Social Media είναι ένας δυναμικά μεταβαλλόμενος και συνεχώς εξελισσόμενος τομέας. Τα χαρακτηριστικά τους μεταβάλλονται μέρα με τη μέρα, καθώς θέλουν να συμφωνούν με τις ανάγκες των χρηστών τους. Έτσι ως επακόλουθο τα APIs τους μπορούν να αλλάζουν μέρα με τη μέρα, για την τροποποίηση της λειτουργικότητας ή της ασφάλειάς τους.

Από την άλλη πλευρά, όλο και περισσότερα κοινωνικά δίκτυα δείχνουν να συμμορφώνονται με τα ανοικτά πρότυπα των APIs επιθυμώντας έτσι να είναι πιο ανοιχτά και προσβάσιμα. Σε αυτό το πλαίσιο, παρέχουν όλο και περισσότερες λειτουργίες μέσα από τα APIs τους, ενώ ταυτόχρονα προσπαθούν να καταφέρουν να πείσουν να εμπλακούν όλο και περισσότεροι προγραμματιστές ώστε να δημιουργήσουν εφαρμογές που θα βασίζονται στις υπηρεσίες τους. Σε αυτό το σημείο πρέπει να τονιστεί ότι η εμπειρία μας από αυτή την μελέτη μας έδειξε ότι όλο και

περισσότερα μέσα κοινωνικής δικτύωσης επιλέγουν να εκθέτουν όλο και με μεγαλύτερη λεπτομέρεια τις μεθόδους τους μέσα από τα API τους, ώστε να επικεντρώνονται σε ενδότερες λειτουργίες τους προσφέροντας έτσι στους προγραμματιστές ένα συνεχώς αυξανόμενο σύνολο δυνατοτήτων.

Κατά κύριο λόγο, τα APIs ασχολούνται με περιεχόμενο διαφόρων τύπων. Αυτό το περιεχόμενο μπορεί να είναι κείμενο, εικόνες, βίντεο ή πιο σύνθετες μορφές, όπως γεγονότα ή albums. Ένα μεγάλο μέρος των APIs είναι αφιερωμένο στη δημιουργία (ή φόρτωση), τροποποίηση και διαγραφή τέτοιου περιεχομένου. Φυσικά, υπάρχουν επίσης λειτουργίες που υποστηρίζουν την άμεση ανάκτηση συγκεκριμένου περιεχομένου. Αυτό που είναι πιο ενδιαφέρον, σε σχέση με την εφαρμογή μας, είναι η ύπαρξη λειτουργιών που επιτρέπουν τη συλλογή δεδομένων σχετικά με οποιοδήποτε τύπο περιεχομένου της πλατφόρμας του social media, όπως "user ratings", "unique visits" ή αναμετάδοσή τους σε άλλους κόμβους του κοινωνικού δικτύου. Η άλλη κύρια κατηγορία περιλαμβάνει API στοιχεία που εστιάζουν στους χρήστες, όπως η αυθεντικοποίηση και τα προσωπικά στοιχεία των χρηστών. Το τελευταίο είναι ιδιαίτερα ενδιαφέρον για τους σκοπούς της παρούσης εργασίας, καθώς δίνει τη δυνατότητα να ανακτηθούν πληροφορίες για τους χρήστες του κοινωνικού δικτύου. Σε συνδυασμό με δεδομένα σχετικά με πραγματικό περιεχόμενο, τέτοιου είδους είσοδος έχει τη δυνατότητα να παρέχει πληροφορίες για τις δραστηριότητες μέσα σε ένα κοινωνικό δίκτυο, όταν αυτές σχετίζονται με συγκεκριμένες εισροές, όπως πολιτικά μηνύματα και συζητήσεις. Οποιαδήποτε εφαρμογή παρόμοιας φιλοσοφίας με αυτή του έργου PADGETS θα μπορούσε κάλλιστα να χρησιμοποιήσει αυτές τις λειτουργίες για την εξαγωγή σημαντικών συμπερασμάτων για την υποβοήθηση και υποστήριξη της διαδικασίας λήψης αποφάσεων.

Όσο αφορά την εφαρμογή που υλοποιήθηκε στα πλαίσια της διπλωματικής εργασίας, όπως αναφέρθηκε και παραπάνω η χρήση της είχε ως σκοπό τη δοκιμή της ορθής λειτουργίας και συνεργασίας της με τα APIs των social media. Διαισθητικά, ύστερα από την εκτέλεση πλήθους δοκιμών με την εφαρμογή μπορούμε να τονίσουμε ότι η εφαρμογή μας προσφέρει τις παρακάτω δυνατότητες:

- Εύκολη, γρήγορη και κυρίως αποτελεσματική προώθηση περιεχομένου πολιτικής
- Επιτυχημένη διείσδυση στους κύκλους των πολιτών
- Άμεση αξιολόγηση περιεχομένου
- Εξαγωγή συμπερασμάτων και υποβοήθηση λήψης αποφάσεων
- Προοπτική ενίσχυσης του e-participation
- Εκμετάλλευση εφαρμογής για προώθηση παντός τύπου περιεχομένου (πχ διαφημιστικό περιεχόμενο)

Οι δυνατότητες των προσφερόμενων APIs των διαφόρων social media σε συνδυασμό με μια εφαρμογή που τις εκμεταλλεύεται δημιουργούν ιδανικές συνθήκες για την ανάπτυξη της διαδραστικής συμμετοχής που επιτρέπει από τη μία τους πολιτικούς να υποβοηθούνται στη διαδικασία λήψης αποφάσεων πολιτικής και από την άλλη ταυτόχρονα τους πολίτες να παρακολουθούν με πλουσιότερη πληροφόρηση τις πολιτικές ζυμώσεις και κυρίως τους επιτρέπει να μπορούν να τις επηρεάσουν δυναμικά.

Βιβλιογραφία

- [1]Charalabidis, Y., Gionis, G., Ferro, E., Loukis, E., ‘Towards a Systematic Exploitation of Web 2.0 and Simulation Modeling Tools in Public Policy Process’, IFIP Second International Conference on e-Participation - ePart 2010, August 29 - September 2, 2010, Lausanne, Switzerland.
- [2]Charalabidis, Y., Loukis, E., ‘Transforming Government Agencies’ Approach to eParticipation through Efficient Exploitation of Social Media’, European Conference on Information Systems (ECIS) 2011, 9-11 June 2011, Helsinki, Finland.
- [3]Charalabidis, Y., Loukis, E., ‘Towards New Web 2.0 Based Multi-Channel Approaches to e-Participation’, in Yogesh Dwivedi (Ed) ‘Transformational Government through eGov: Socio-economic, Cultural, and Technological issues’, Emerald Publications, 2012.
- [4]Padgets - “Deliverable 1.1 - Categorization of Web 2.0 Social Media and Stakeholder Characteristics”, <http://www.padgets.eu/>, 28/06/2010.
- [5]Padgets - “Deliverable 1.2 – Standards, Interfaces and APIs for interplatform communication in Web2.0 Social Media”, <http://www.padgets.eu/>, 30/06/2010.
- [6]Padgets - “D2.1: Padget Design and Decision Model for Policy Making”, <http://www.padgets.eu/>, 31/01/2011.
- [7]Padgets - “Deliverable 3.2 - PADGETS Platform Individual Components”, <http://www.padgets.eu/>, 17/05/2011.
- [8]Dunleavy, P. and Margetts, H. ‘The second wave of digital era governance.’ In: American Political Science Association Conference, 4 September 2010, Washington DC, USA. (Unpublished).
- [9]Benkler, Y., ‘The Wealth of Networks: How Social Production Transforms Markets and Freedom’, Yale University Press.
- [10]Von Hippel, E., ‘Democratizing Innovation’, MIT Press, Cambridge.
- [11]Howe, J., ‘The Rise of Crowdsourcing’ , Wired, 14 June 2006.
- [12]Tapscott, D. and Williams, A., ‘Wikinomics: How Mass Collaboration Changes Everything (Portfolio)’, 2006
- [13]Shirky, C., ‘Cognitive Surplus’, Allen Lane, New York, 2010.
- [14]Christakis, N. and Fowler, J., ‘Connected: The Surprising Power of Our Social Networks and How they Shape Our Lives’, Brown & Co, Little, 2009.

- [15]Escher, T., Margetts, H., Petricek, V. & Cox, I., ‘Governing from the centre? Comparing the nodality of digital governments’ , paper to the 2006 Annual Meeting of the American Political Science Association, 31 Aug-4 Sept2006.
- [16]Hood, C. and Margetts, H., ‘The Tools of Government in the Digital Age, Palgrave’, Basingstoke, 2007.
- [17]Shirky, C., ‘Here Comes Everybody. The power of organizing without organizations’, Allen Lane, New York, 2008.
- [18]UNESCAP, “What is Good Governance? “ Retrieved April, 2010, from <http://www.unescap.org/pdd/prs/ProjectActivities/Ongoing/gg/governance.asp>
- [19]Wimmer, M.A.; Stadach, M.; Morland, P., “Glossary” . In (Zechner, A. Hrsg.): E-Government Guide Germany. Strategies, Solutions and Efficiency, Fraunhofer IRB Verlag, Stuttgart, 2007; S. 431-456
- [20]Repnik, H.P., Mohs, R.M., "Good Governance", Democracy and Development Paradigms. In: Intereconomics, Band 27, Nr. 1, 1992
- [21]Levy, P., ‘Collective Intelligence: Mankind’ s Emerging World in Cyberspace’ (R. Bononno, Trans.), Plenum, New York, 1997.
- [22] “Policy Brief: Engaging Citizens Online for Better Policy-making” , OECD Observer, March, 2003.
- [23]Ferro E., Molinari F., “Framing Web 2.0 in the Process of Public Sector Innovation: Going Down the Participation Ladder” in European Journal of ePractice www.epracticejournal.eu, No 9 March 2010, pg.23 ISSN: 1988-625X.
- [24]Gammack, J., Barker, M., ‘E-democracy and public participation: a global overview of policy and activity’. School of Management, Griffith University Queensland, 2003.
- [25]Benčina J., ‘Web-based Decision Support System for the Public Sector Comprising Linguistic Variables’. Informatica 31 311-323, 2007.
- [26]Dunleavy, Patrick and Margetts, Helen, ‘The second wave of digital era governance.’ In: American Political Science Association Conference, 4 September 2010, Washington DC, USA (Unpublished).
- [27]Richardson, J. J., ‘European Union : power and policy-making’, Routledge, Oxon, England, 2006.
- [28]Immergut, E., “The Rules of the Game: The Logic of Health Policy-Making in France, Switzerland, and Sweden.” In Steinmo S., Thelen K., Longstreth F., (eds.), Structuring politics, 1992, Cambridge University Press.

- [29]Weiss, J. A., 'Pathways to cooperation among public agencies'. *Journal of Policy Analysis and Management*, 1987, vol. 7, issue 1, pages 94-117.
- [30]Bardach, E., 'Getting Agencies to Work Together: The Practice and Theory of Managerial Craftsmanship', 1998, The Brookings Institution.
- [31]Strang D. and Meyer J.W., 'Institutional Conditions for Diffusion, Theory and Society', Vol. 22, No. 4. (1993).
- [32]Bomberg, E. and Peterson, J., 'Policy Transfer and Europeanization: Passing the Heineken Test?' *Queen's On-Line Papers on Europeanization*, No. 2/2000.
- [33]Lodge, Martin and James, Oliver, 'The limitations of 'policy transfer' and 'lesson drawing' for public policy research'. *Political studies review*, 1 (4). pp. 179-193, 2003.
- [34]Dolowitz, D. and Marsh, D., 'Who Learns What from Whom: a Review of the Policy Transfer Literature', *Political Studies*, 44, 343-57, 1996. Evans, M. and Davies, J., 'Understanding Policy Transfer: a Multi-Level, Multi-Disciplinary Perspective', *Public Administration*, 77 (2), 361-85.1999.
- [35]Stone, D., 'Lesson Drawing and Policy Transfer', *Politics*, 19 (1), 51-59, 1999.
- [36]Dolowitz, D., 'Introduction', *Governance*, 13 (1), 1-4, 2000.
- [37]Dolowitz, D. and Marsh, D. (2000) 'Learning from Abroad: the Role of Policy Transfer in Contemporary Policy-Making', *Governance*, 13 (1), 5-24, 2000.
- [38]Rose, R., 'What is Lesson Drawing?', *Journal of Public Policy*, 11 (1), 3-30, 1991.
- [39]Bennett, C. J., 'Review Article: What is Policy Convergence and What Causes It?', *British Journal of Political Science*, 21, 215-33, 1991.
- [40]Bennett, C. J., 'Understanding Ripple Effects: the Cross-National Adoption of Policy Instruments for Bureaucratic Accountability', *Governance*, 10 (3), 213-33, 1997.
- [41]Jacoby, W., 'Imitation and Politics', Cornell University Press, Ithaca NY, 2000.
- [42]Witte, B., 'Eisenbahn und Staat: Ein Vergleich der europäischen und nordamerikanischen Eisenbahnorganisationen in ihrem Verhältnis zum Staat', Gustav Fischer, Jena, 1932.
- [43]Barker, E., 'The Development of Public Services in Western Europe'. Oxford University Press, London, 1944.

- [44]Lerner, D., ‘The Transfer of Institutions’ , in W. Hamilton (ed.) The Transfer of Institutions. Duke University Press, Durham NC, 1964.
- [45]Subramaniam, V., ‘Transplanted Indo-British Administration’. Ashish, New Delhi, 1977.
- [46]Wolman, H., ‘Understanding Cross-National Policy Transfers: the Case of Britain and the US’, Governance, 5 (1), 27-45, 1992.
- [47]Berry, F. C. and Berry, W. D., ‘Innovation and Diffusion Models in Policy Research’ , in P. A. Sabatier (ed.) Theories of the Policy Process. Westview, Boulder CO, 1999.
- [48]Most, B. A. and Starr, H., ‘Theoretical and Logical Issues in the Study of International Diffusion’ , Journal of Theoretical Politics, 2 (4), 391-412, 1990.
- [49]DiMaggio, P. J. and Powell, W. W., ‘The Iron Cage Revisited: Institutional Isomorphism and Collective Rationality in Organisational Fields’ , in W. W. Powel and P. J. DiMaggio, The New Institutionalism in Organisational Analysis. University of Chicago Press, Chicago IL, pp. 63-82, 1991.
- [50]Beer, S., ‘Decision and Control’. Wiley, London, 1966.
- [51]S.-M. Kim and E. Hovy, “Determining the sentiment of opinions,” Proceedings of the International Conference on Computational Linguistics (COLING), 2004.
- [52]S.-M. Kim and E. Hovy, “Automatic identification of pro and con reasons in online reviews,” Proceedings of the COLING/ACL Main Conference Poster Sessions, pp. 483-490, 2006.
- [53]S.-M. Kim and E. Hovy, “Crystal: Analyzing predictive opinions on the web,” Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP/CoNLL), 2007.
- [54] T. Wilson, J. Wiebe, and P. Hoffmann, “Recognizing contextual polarity in phrase-level sentiment analysis.” Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), pp. 347-354, 2005.
- [55] L. Polanyi and A. Zaenen, “Contextual lexical valence shifters,” Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text, 2004.
- [56]S. Morinaga, K. Yamanishi, K. Tateishi, and T. Fukushima, “Mining product reputations on the Web,” Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), pp. 341-349, 2002. (Industry track).

Παράρτημα

I. - Λεξικό βασικών όρων του Padgets[5]

Term	Definition
PADGETS	An EU - FP7 STREP project developing novel social media applications for policy making.
Padget	A resource (application or content), typically instantiated within a social media platform, created by a policy stakeholder, providing interactivity with the citizens.
Padget main components	A padget is composed by four main components: a policy message, the social context, the decision services and the interaction services.
PADGETS platform	The software application supporting the Padgets creation, distribution, monitoring and withdrawal.
Padget Initiator	The creator and owner of a Padget, typically a policy maker. (Other stakeholders such as lobbying organisations, citizens' groups may act as Padget Initiators).
Padget (End) User	The individual reached by a Padget, through a Social Media Platform.
Padget Functions	The various functions provided by a Padget towards the users, typically ranging from simple to complex, as following: <ul style="list-style-type: none"> • information publishing (read only) • one or more, single of multiple choice, pre-defined questions with pre-defined answers (e.g. polls) • open text response • inclusion of attachments in response.
Padget Campaign	A Padget Campaign is a set of activities covering creation, distribution, interaction, monitoring and termination of one or more Padgets for a specific goal.
Padget Campaign Set-up	The activities before the launching of a Padget, typically including the creation or enhancement of the Padget initiator presence in the targeted SMP's.
Social Media Platform API's	The application programmable interfaces provided by a specific SMP, for performing actions through the invocation of a web service. Similar: the Social Networking Platform API's.

Social Media Platform (SMP)	The term denoting any social and media related internet-based system (such as YouTube, Picasa, Blogger, etc.) The term includes SNP's.
Social Network(ing) Platform (SNP)	The term denoting the social networking, internet-based systems (such as Facebook, MySpace, LinkedIn, etc.).
PADGETS Front-End	The part of PADGETS platform exposed to the Padgets initiator or the (end) user, through the appropriate interface.
PADGETS Back-End	The part of PADGETS platform not exposed to the Padgets initiator or the (end) user.
PADGETS Privacy Component	The part of PADGETS platform that manages user privacy settings or preferences.
PADGETS Decision Support Component	The part of PADGETS platform that formulates the final output of a Padget Campaign, towards the Padgets initiator.
PADGETS Simulation Model	The module of the PADGETS platform analyzing and projecting into the future the diffusion process of the policy message in terms of awareness, interest and acceptance.
PADGETS Analytics	The module of the PADGETS decision support component processing textual and demographic data gathered through Padgets Campaigns to extract the opinions expressed about the policy message.
Padget Context	The specific "neighbourhood" of a Padget, e.g. the area where a Padget is instantiated. It has three aspects: technical, policy and social.
Padget Technical Context	The technical context where a Padget is instantiated, i.e. the specific SMP or part of SMP.
Padget Policy Context	The subject or theme a Padget is addressing, i.e. economic policy, environmental policy, a specific legal element, a specific municipal plan, etc.
Padget Social Context	The network of individuals a Padget is reaching through the social network connections of its initiator.
Padget Life-cycle	The set of states a Padget may have.

II. - Ανάλυση των διασημότερων Social Media[4]

SOCIAL MEDIA		Unique Visitors	Unique Users	GENDER	
				W	M
Bebo.com		15000000	5600000		
	USA	3500000	1100000	66	34
	UK			55	45
	Germany			38	62
	Italy			28	72
	Spain			38	62
	France			26	74
	<i>EU5 average</i>			37	63
Blogger.com		73000000	41000000		
	USA	21000000	8100000	62	38
	UK			36	64
	Germany			30	70
	Italy			30	70
	Spain			38	62
	France			30	70
	<i>EU5 average</i>			32.8	67.2
Chatroulette.com		6700000	2600000		
	USA	2000000	620000	43	57
	UK			38	62
	Germany			22	78
	Italy			18	82
	Spain			15	85
	France			24	76
	<i>EU5 average</i>			23.4	76.6
Classmates.com		19000000	6700000		
	USA	17000000	6200000	66	34
	UK			48	52
	Germany			28	72
	Italy			34	66
	Spain			36	64
	France			28	72
	<i>EU5 average</i>			34.8	65.2
Delicious.com		8200000	5100000		
	USA	2900000	1200000	55	45
	UK			30	70
	Germany			13	87
	Italy			21	79
	Spain			26	74
	France			17	83
	<i>EU5 average</i>			21.4	78.6

SOCIAL MEDIA		Unique Visitors	Unique Users	GENDER	
				W	M
Digg.com		25000000	14000000		
	USA	12000000	5100000	43	57
	UK			30	70
	Germany			17	83
	Italy			18	82
	Spain			18	82
	France			18	82
	<i>EU5 average</i>			20.2	79.8
Facebook.com		1100000000	370000000		
	USA	490000000	130000000	60	40
	UK			55	45
	Germany			45	55
	Italy			45	55
	Spain			45	55
	France			48	52
	<i>EU5 average</i>			47.6	52.4
Flickr.com		110000000	55000000		
	USA	37000000	14000000	55	45
	UK			45	55
	Germany			32	68
	Italy			38	62
	Spain			41	59
	France			32	68
	<i>EU5 average</i>			37.6	62.4
Friendfeed.com		3800000	2400000		
	USA	990000	430000	57	43
	UK			28	72
	Germany			11	89
	Italy			21	79
	Spain			15	85
	France			9	91
	<i>EU5 average</i>			16.8	83.2
Friendster.com		14000000	13000000		
	USA	2400000	740000	57	43
	UK			45	55
	Germany			45	55
	Italy			34	66
	Spain			48	52
	France			26	74
	<i>EU5 average</i>			39.6	60.4
Hi5.com		50000000	35000000		
	USA	5000000	1800000	55	45
	UK			43	57
	Germany			38	62

SOCIAL MEDIA		Unique Visitors	Unique Users	GENDER	
				W	M
	Italy			34	66
	Spain			50	50
	France			45	55
	<i>EU5 average</i>			42	58
Last.fm		24000000	11000000		
	USA	7600000	2600000	55	45
	UK			41	59
	Germany			36	64
	Italy			28	72
	Spain			34	66
	France			32	68
	<i>EU5 average</i>			34.2	65.8
Linkedin.com		82000000	41000000		
	USA	42000000	16000000	50	50
	UK			41	59
	Germany			26	74
	Italy			32	68
	Spain			32	68
	France			28	72
	<i>EU5 average</i>			31.8	68.2
Livejournal.com		26000000	6800000		
	USA	16000000	2300000	55	45
	UK			43	57
	Germany			28	72
	Italy			34	66
	Spain			36	64
	France			22	78
	<i>EU5 average</i>			32.6	67.4
Myspace.com		210000000	80000000		
	USA	140000000	41000000	66	34
	UK			48	52
	Germany			45	55
	Italy			38	62
	Spain			41	59
	France			38	62
	<i>EU5 average</i>			42	58
Netvibes.com		2400000	1200000		
	USA	390000	170000	52	48
	UK			30	70
	Germany			13	87
	Italy			22	78
	Spain			30	70
	France			21	79
	<i>EU5 average</i>			23.2	76.8

SOCIAL MEDIA		Unique Visitors	Unique Users	GENDER	
				W	M
Ning.com		37000000	13000000		
	USA	17000000	4600000	64	36
	UK			43	57
	Germany			38	62
	Italy			38	62
	Spain			45	55
	France			32	68
	<i>EU5 average</i>			39.2	60.8
Reddit.com		5100000	2400000		
	USA	2800000	1000000	34	66
	UK			24	76
	Germany			12	88
	Italy			15	85
	Spain			19	81
	France			24	76
	<i>EU5 average</i>			18.8	81.2
Stumbleupon.com		11000000	5500000		
	USA	6200000	2400000	52	48
	UK			32	68
	Germany			12	88
	Italy			16	84
	Spain			19	81
	France			16	84
	<i>EU5 average</i>			19	81
Tagged.com		28000000	16000000		
	USA	8100000	2800000	64	36
	UK			52	48
	Germany			38	62
	Italy			24	76
	Spain			45	55
	France			36	64
	<i>EU5 average</i>			39	61
Twitter.com		150000000	80000000		
	USA	54000000	21000000	60	40
	UK			43	57
	Germany			26	74
	Italy			26	74
	Spain			32	68
	France			26	74
	<i>EU5 average</i>			30.6	69.4
Vimeo.com		18000000	8300000		
	USA	920000	320000	50	50
	UK			32	68
	Germany			21	79

SOCIAL MEDIA		Unique Visitors	Unique Users	GENDER	
				W	M
	Italy			26	74
	Spain			30	70
	France			22	78
	<i>EU5 average</i>			26.2	73.8
Wikipedia.org		550000000	310000000		
	USA	160000000	56000000	52	48
	UK			45	55
	Germany			38	62
	Italy			38	62
	Spain			43	57
	France			43	57
	<i>EU5 average</i>			41.4	58.6
Wordpress.com		210000000	110000000		
	USA	59000000	21000000	57	43
	UK			45	55
	Germany			34	66
	Italy			38	62
	Spain			43	57
	France			34	66
	<i>EU5 average</i>			38.8	61.2
Xanga.com		4600000	2200000		
	USA	2000000	630000	66	34
	UK			50	50
	Germany			43	57
	Italy			32	68
	Spain			36	64
	France			21	79
	<i>EU5 average</i>			36.4	63.6

SOCIAL MEDIA		AGE GROUPS						
		0-17	18-24	25-34	35-44	45-54	55-64	>65
Bebo.com								
	USA	41	8	11	16	16	6	2
	UK	38	14	18	15	10	4	1
	Germany	17	11	19	22	20	7	4
	Italy	10	18	35	22	14	0	0
	Spain	9	18	46	18	10	0	0
	France	24	5	20	23	23	4	0
	<i>EU5 average</i>	19.6	13.2	27.6	20	15.4	3	1
Blogger.com								
	USA	7	7	22	27	20	13	4
	UK	4	15	33	21	15	8	3
	Germany	9	8	26	20	24	8	4

SOCIAL MEDIA		AGE GROUPS						
		0-17	18-24	25-34	35-44	45-54	55-64	>65
	Italy	4	12	31	29	16	7	1
	Spain	6	11	41	29	9	4	0
	France	10	5	24	32	18	9	2
	<i>EU5 average</i>	6.6	10.2	31	26.2	16.4	7.2	2
Chatroulette.com								
	USA	14	17	20	25	17	6	2
	UK	8	33	33	14	8	3	1
	Germany	12	11	34	16	21	5	2
	Italy	0	17	36	30	14	4	0
	Spain	5	12	56	16	6	3	0
	France	15	11	25	25	17	5	1
	<i>EU5 average</i>	8	16.8	36.8	20.2	13.2	4	0.8
Classmates.com								
	USA	11	3	7	18	31	23	8
	UK	0	12	25	26	23	14	0
	Germany	0	0	34	41	25	0	0
	Italy	0	0	41	31	28	0	0
	Spain	0	0	57	43	0	0	0
	France	0	0	42	39	20	0	0
	<i>EU5 average</i>	0	2.4	39.8	36	19.2	2.8	0
Delicious.com								
	USA	5	8	20	32	18	14	4
	UK	3	16	33	21	17	7	3
	Germany	4	5	44	17	21	6	3
	Italy	3	10	44	28	12	5	0
	Spain	2	7	54	28	6	2	0
	France	6	2	32	33	20	7	0
	<i>EU5 average</i>	3.6	8	41.4	25.4	15.2	5.4	1.2
Digg.com								
	USA	10	9	19	28	19	12	4
	UK	8	18	31	22	13	6	2
	Germany	7	8	39	17	22	5	2
	Italy	6	13	37	28	12	4	0
	Spain	4	11	57	20	6	2	0
	France	11	5	27	33	18	5	1
	<i>EU5 average</i>	7.2	11	38.2	24	14.2	4.4	1
Facebook.com								
	USA	17	7	14	21	25	12	4
	UK	10	12	31	24	13	7	3
	Germany	13	13	26	21	16	7	4

SOCIAL MEDIA		AGE GROUPS						
		0-17	18-24	25-34	35-44	45-54	55-64	>65
	Italy	6	9	34	26	16	6	2
	Spain	7	7	45	26	10	5	1
	France	20	15	18	23	14	8	2
	<i>EU5 average</i>	11.2	11.2	30.8	24	13.8	6.6	2.4
Flickr.com								
	USA	13	8	19	23	21	12	4
	UK	9	16	27	23	16	8	3
	Germany	12	8	27	23	21	6	3
	Italy	6	12	34	28	14	4	1
	Spain	8	11	45	24	8	3	0
	France	15	5	22	30	19	8	1
	<i>EU5 average</i>	10	10.4	31	25.6	15.6	5.8	1.6
Friendfeed.com								
	USA	3	12	24	35	15	8	3
	UK	0	20	42	16	18	4	0
	Germany	0	5	51	20	20	5	0
	Italy	1	10	49	28	9	3	0
	Spain	0	7	71	19	4	0	0
	France	0	0	40	33	23	4	0
	<i>EU5 average</i>	0.2	8.4	50.6	23.2	14.8	3.2	0
Friendster.com								
	USA	27	8	15	19	23	6	2
	UK	11	21	40	13	10	4	2
	Germany	29	13	18	14	20	5	0
	Italy	8	15	43	22	9	3	0
	Spain	11	18	54	18	0	0	0
	France	29	5	27	20	18	0	0
	<i>EU5 average</i>	17.6	14.4	36.4	17.4	11.4	2.4	0.4
Hi5.com								
	USA	19	8	28	20	18	5	2
	UK	8	15	43	18	10	4	2
	Germany	12	12	21	21	21	7	5
	Italy	9	15	37	19	14	4	1
	Spain	13	11	42	20	10	3	1
	France	17	10	26	24	16	6	1
	<i>EU5 average</i>	11.8	12.6	33.8	20.4	14.2	4.8	2
Last.fm								
	USA	21	11	16	21	19	9	3
	UK	10	20	26	20	14	8	3
	Germany	16	15	26	15	18	6	2

SOCIAL MEDIA		AGE GROUPS						
		0-17	18-24	25-34	35-44	45-54	55-64	>65
	Italy	8	16	32	24	15	4	1
	Spain	8	12	46	24	8	3	0
	France	25	6	17	28	16	6	1
	<i>EU5 average</i>	13.4	13.8	29.4	22.2	14.2	5.4	1.4
LinkedIn.com								
	USA	3	4	16	28	28	17	4
	UK	2	8	32	27	18	10	3
	Germany	2	4	35	27	18	8	5
	Italy	2	7	31	38	16	6	1
	Spain	2	6	43	36	8	4	1
	France	5	4	37	28	14	11	1
	<i>EU5 average</i>	2.6	5.8	35.6	31.2	14.8	7.8	2.2
Livejournal.com								
	USA	12	16	23	23	16	8	2
	UK	6	27	30	17	12	6	2
	Germany	9	9	34	26	16	4	2
	Italy	6	25	36	21	9	4	0
	Spain	5	22	47	18	6	2	0
	France	21	4	20	26	24	4	0
	<i>EU5 average</i>	9.4	17.4	33.4	21.6	13.4	4	0.8
Myspace.com								
	USA	34	8	12	16	20	6	2
	UK	10	21	30	19	13	5	2
	Germany	16	16	20	20	18	6	2
	Italy	6	14	37	23	14	4	1
	Spain	7	13	49	19	8	3	0
	France	18	9	22	26	18	6	1
	<i>EU5 average</i>	11.4	14.6	31.6	21.4	14.2	4.8	1.2
Netvibes.com								
	USA	5	10	23	34	16	10	3
	UK	0	18	39	19	15	6	3
	Germany	3	3	48	16	20	5	6
	Italy	0	14	36	33	13	5	0
	Spain	3	6	52	27	8	3	0
	France	6	4	31	31	18	10	2
	<i>EU5 average</i>	2.4	9	41.2	25.2	14.8	5.8	2.2
Ning.com								
	USA	15	9	17	25	20	11	3
	UK	11	20	28	20	13	6	2
	Germany	18	9	22	21	21	6	3

SOCIAL MEDIA		AGE GROUPS						
		0-17	18-24	25-34	35-44	45-54	55-64	>65
	Italy	10	15	34	24	12	4	1
	Spain	8	11	42	24	10	4	0
	France	21	6	20	27	20	6	1
	<i>EU5 average</i>	13.6	12.2	29.2	23.2	15.2	5.2	1.4
Reddit.com								
	USA	6	12	29	27	14	9	3
	UK	4	20	32	20	15	6	2
	Germany	0	5	50	18	20	7	0
	Italy	0	14	43	25	18	0	0
	Spain	0	11	67	22	0	0	0
	France	16	0	23	34	18	9	0
	<i>EU5 average</i>	4	10	43	23.8	14.2	4.4	0.4
Stumbleupon.com								
	USA	8	12	21	28	17	10	4
	UK	6	20	32	18	15	8	3
	Germany	3	7	43	17	22	4	3
	Italy	0	16	44	30	10	0	0
	Spain	0	10	64	21	6	0	0
	France	8	0	32	36	18	6	0
	<i>EU5 average</i>	3.4	10.6	43	24.4	14.2	3.6	1.2
Tagged.com								
	USA	28	5	12	18	28	7	2
	UK	15	16	33	17	13	4	2
	Germany	8	12	24	20	22	9	4
	Italy	4	12	38	16	24	5	0
	Spain	14	10	41	19	11	4	2
	France	14	16	23	18	24	5	0
	<i>EU5 average</i>	11	13.2	31.8	18	18.8	5.4	1.6
Twitter.com								
	USA	10	9	18	27	20	11	3
	UK	6	14	32	22	15	7	3
	Germany	7	7	34	21	19	8	4
	Italy	4	12	35	30	12	5	1
	Spain	4	10	54	23	7	2	0
	France	11	4	25	31	19	8	1
	<i>EU5 average</i>	6.4	9.4	36	25.4	14.4	6	1.8
Vimeo.com								
	USA	8	11	21	26	18	12	4
	UK	6	20	31	20	15	7	2
	Germany	6	10	36	19	21	6	3

SOCIAL MEDIA		AGE GROUPS						
		0-17	18-24	25-34	35-44	45-54	55-64	>65
	Italy	6	11	37	28	12	5	0
	Spain	4	9	53	23	7	4	0
	France	12	6	25	31	17	7	1
	<i>EU5 average</i>	6.8	11.2	36.4	24.2	14.4	5.8	1.2
Wikipedia.org								
	USA	13	6	15	22	24	15	4
	UK	10	13	26	24	15	9	3
	Germany	12	8	19	24	21	11	6
	Italy	7	10	26	29	20	6	1
	Spain	10	10	36	27	12	5	1
	France	18	6	20	27	15	12	2
	<i>EU5 average</i>	11.4	9.4	25.4	26.2	16.6	8.6	2.6
Wordpress.com								
	USA	15	8	18	24	20	11	3
	UK	12	18	28	21	13	6	2
	Germany	13	9	25	23	21	6	3
	Italy	7	11	30	28	17	5	1
	Spain	9	10	39	27	10	4	0
	France	19	5	21	28	19	7	1
	<i>EU5 average</i>	12	10.6	28.6	25.4	16	5.6	1.4
Xanga.com								
	USA	21	18	17	19	16	7	2
	UK	9	32	26	17	11	4	0
	Germany	23	17	31	12	17	0	0
	Italy	0	24	52	24	0	0	0
	Spain	0	27	58	15	0	0	0
	France	28	0	20	32	20	0	0
	<i>EU5 average</i>	12	20	37.4	20	9.6	0.8	0