



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Εύρεση πληροφοριών στο Web μέσω Twitter

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

των

ΑΘΑΝΑΣΙΟΥ ΑΝΔΡΕΟΥ
ΔΗΜΗΤΡΙΟΥ ΣΕΡΜΠΙΟΥ

Επιβλέπων : Τιμολέων Σελλής
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2012

Η σελίδα αυτή είναι σκόπιμα λευκή.



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
Τομέας τεχνολογίας ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Εύρεση πληροφοριών στο Web μέσω Twitter

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

των

ΑΘΑΝΑΣΙΟΥ ΑΝΔΡΕΟΥ
ΔΗΜΗΤΡΙΟΥ ΣΕΡΜΠΟΥ

Επιβλέπων : Τιμολέων Σελλής
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 22η Οκτωβρίου 2012.

.....
Τιμολέων Σελλής
Καθηγητής Ε.Μ.Π.

.....
Ιωάννης Βασιλείου
Καθηγητής Ε.Μ.Π.

.....
Γιάννης Σταύρακας
Ερευνητής Β' ΙΠΣΥ/Ε.Κ.
«Αθηνά»

Αθήνα, Οκτώβριος 2012

.....
Αθανάσιος Ανδρέου

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

.....
Δημήτριος Σέρμπος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

© 2012 – Με επιφύλαξη παντός δικαιώματος All rights reserved

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τους συγγραφείς.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν των συγγραφέων και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Ευχαριστίες

Θα θέλαμε να ευχαριστήσουμε όλους όσους μας βοήθησαν στη εκπόνηση της διπλωματικής εργασίας.

Ευχαριστούμε τον επιβλέποντα Καθηγητή κ. Τιμολέοντα Σελλή, ο οποίος μας έδωσε την ευκαιρία να ασχοληθούμε με τη διπλωματική αυτή εργασία.

Επίσης θα θέλαμε να ευχαριστήσουμε θερμά τον συν-επιβλέποντα ερευνητή κ. Γιάννη Σταύρακα για τη πολύτιμη καθοδήγηση που χάρη σε αυτή ολοκληρώθηκε αυτή η διπλωματική.

Τέλος θα θέλαμε να ευχαριστήσουμε τους γονείς μας για τη στήριξη τους, καθώς και τους φίλους μας Ειρήνη και Νίκο που με τις συζητήσεις που κάναμε μαζί τους στα διάφορα στάδια αυτής της εργασίας, άφησαν και αυτοί το στίγμα τους πάνω της.

Η σελίδα αυτή είναι σκόπιμα λευκή

Περίληψη

Τα διαδικτυακά κοινωνικά δίκτυα (OSN) είναι ένας καινούριος τομέας παροχής υπηρεσιών ο οποίος λόγω της αλματώδους ανάπτυξής του, αποτελεί ένα δυναμικό παράγοντα με ισχυρή επίδραση στη παγκόσμια οικονομία, τις κατευθύνσεις της τεχνολογικής ανάπτυξης καθώς και τις κοινωνικές εξελίξεις. Για αυτό το λόγο τα OSN παρέχουν μια μοναδική ευκαιρία για ένα ευρύ φάσμα έρευνας που οπωσδήποτε πρέπει να περιλαμβάνει μεθοδολογίες για τη συλλογή αυτών των δεδομένων καθώς και την επεξεργασία τους. Αυτή την περιοχή προσπαθεί να ενισχύσει η συγκεκριμένη πτυχιακή εργασία, η οποία επικεντρώνεται στο Twitter , ένα από τα μεγαλύτερα OSN, που λόγω των ιδιοτήτων του έχει αναδειχτεί σε μια πολύ σημαντική πηγή πληροφοριών. Η εφαρμογή που αναπτύχθηκε στα πλαίσια της πτυχιακής εργασίας είναι μια Web εφαρμογή. Η εφαρμογή αυτή έχει κύριο στόχο να εκκινεί «Campaigns» για την συνεχιζόμενη αναζήτηση στην υπηρεσία του Twitter , εξάγοντας πληροφορίες, όπως Tweets, hashtags και κυρίως URLs μέσα στα Tweets. Στη συνέχεια να τις επεξεργάζεται με ένα τρόπο ώστε να αποκαλύπτονται κάποια δεδομένα που το ίδιο το Twitter δεν έχει αναπτύξει λειτουργία για να τα εμφανίζει. Και τέλος να παρέχει στο χρήστη ένα Web γραφικό περιβάλλον για να μελετήσει όλες αυτές τις πληροφορίες, αλλά και να τις εξάγει σε αρχεία που να μπορεί μετά να τα επεξεργαστεί μέσα σε ένα άλλο, μεγαλύτερο και πιο σημαντικό πλαίσιο.

Στη διαδικασία του σχεδιασμού και τις υλοποίησης, αξιοποιήθηκαν σε ένα μεγάλο βαθμό οι δυνατότητες που προσφέρει το εξαιρετικά οργανωμένο και λειτουργικό Twitter API καθώς και το LongURL API, χρησιμοποιήθηκαν δημοφιλή εργαλεία και πλατφόρμες ανάπτυξης, όπως η γλώσσα προγραμματισμού Java, η πλατφόρμα σχεδίασης Web εφαρμογών HTML/CSS/JavaScript/JQuery και η σχεσιακή βάση MySQL με στόχο η εφαρμογή μας να είναι αυτόνομη αλλά να μπορεί να εισαχθεί και σε κάποια άλλη πλατφόρμα.

Λέξεις Κλειδιά: twitter, campaign, διαδικτυακά κοινωνικά δίκτυα

Η σελίδα αυτή είναι σκόπιμα λευκή

Abstract

The online social networks (OSN) is an new service sector that due to its rapid development has a strong impact on the global economy, the direction of technological development and social change. These are the reasons why the OSN provide a unique opportunity for research that should include methodologies for the collection of data and its processing. The purpose of this thesis is to enhance this section of research, focusing on twitter, one of the biggest OSN, which due to its specificity, has emerged as an important source of information. The application developed is a Web application. Its main purpose is to start 'campaigns' which crawl in the twitter service, extracting information like Tweets, hashtags and mainly URLs within Tweets. Then it processes the extracted information to disclose specific data, in ways that Twitter has not developed yet any functions to do it. Finally, it provides a web graphic user interface for the user to study all the information and to export it to files that can be later edited in a different context.

During the process of planning and implementation, the potential of the highly organized and functional Twitter API were used to a large extent, as well as those of the LongURL API. Popular tools and development platforms were used, such as the programming language JAVA, the web applications design platform HTML / CSS / JavaScript / JQuery and MySQL relational database so that the application can be autonomous but still able to be inserted in another platform.

Keywords: twitter, campaigns, online social networks

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πίνακας περιεχομένων

1	Εισαγωγή.....	1
1.1	Εύρεση URL και πληροφοριών μέσω του twitter.....	1
1.2	Αντικείμενο διπλωματικής.....	2
1.2.1	Πρόβλημα.....	2
1.2.2	Σκοπός.....	2
1.2.3	Βασικές Λειτουργίες.....	3
1.2.4	Συνεισφορά.....	3
1.3	Οργάνωση κειμένου.....	4
2	Σχετικές εργασίες.....	5
2.1	Πλατφόρμα υποστήριξης ανάλυσης δεδομένων στο twitter.....	5
3	Υπόβαθρο.....	7
3.1	Twitter APIs.....	7
3.1.1	Twitter for Websites.....	7
3.1.2	REST API.....	8
3.1.3	Streaming API.....	8
3.1.4	Search API.....	8
3.2	LongURL API.....	11
4	Ανάλυση Απαιτήσεων Συστήματος.....	13
4.1	Αρχιτεκτονική.....	13
4.2	Περιγραφή Λειτουργιών.....	14
4.2.1	Απλή αναζήτηση.....	14
4.2.2	Campaign Refinement.....	16
4.2.3	Campaign Management.....	20
4.2.4	Campaign viewer.....	22
4.3	Μοντέλο Οντοτήτων Συσχετίσεων.....	23
5	Σχεδίαση Συστήματος.....	27
5.1	Αρχιτεκτονική.....	27

5.2	Περιγραφή Κλάσεων	28
5.2.1	<i>InitialScreen.java</i>	28
5.2.2	<i>ManageRefresh.java</i>	29
5.2.3	<i>ManageTabs.java</i>	29
5.2.4	<i>MainNavigation.java</i>	29
5.2.5	<i>ListHandler.java</i>	29
5.2.6	<i>FinalPanel.java</i>	29
5.2.7	<i>CampaignSearch.java</i>	29
5.2.8	<i>FileDownloader.java</i>	29
5.2.9	<i>ManageFinalTabs.java</i>	29
5.2.10	<i>Campaign.java</i>	30
5.2.11	<i>CampaignFactory.java</i>	30
5.2.12	<i>JSONFactory.java</i>	30
5.2.13	<i>LongUrl.java</i>	30
5.2.14	<i>MailSender.java</i>	30
5.2.15	<i>Tuple.java</i>	30
5.2.16	<i>QueryMaker.java</i>	30
5.2.17	<i>CampaignManager.java</i>	35
5.3	Βάση Δεδομένων	35
5.4	Κωδικοποίηση αρχείων.....	36
6	Υλοποίηση	41
6.1	Λεπτομέρειες υλοποίησης.....	41
6.1.1	<i>Multi-threading Handling</i>	41
6.1.2	<i>Δυναμικό περιεχόμενο στο Campaign Management</i>	45
6.1.3	<i>Υλοποίηση του θεματικού νέφους</i>	48
6.2	Πλατφόρμες και προγραμματιστικά εργαλεία	49
6.2.1	<i>Πλατφόρμες ανάπτυξης</i>	50
6.2.2	<i>Εργαλεία ανάπτυξης Συστήματος</i>	56
6.3	Οδηγίες εγκατάστασης.....	56
6.3.1	<i>Εγκατάσταση Java SE</i>	56
6.3.2	<i>Εγκατάσταση Tomcat</i>	56

6.3.3	Εγκατάσταση MySQL	58
7	Έλεγχος.....	63
7.1	Μεθοδολογία ελέγχου.....	63
7.2	Αναλυτική παρουσίαση ελέγχου.....	63
8	Επίλογος.....	69
8.1	Σύνοψη και συμπεράσματα.....	69
8.2	Μελλοντικές επεκτάσεις	70
9	Βιβλιογραφία.....	71

Εικόνα 1 Αρχιτεκτονική της πλατφόρμας [TWP].....	6
Εικόνα 2 Block διάγραμμα.....	14
Εικόνα 3 Κεντρική Σελίδα	15
Εικόνα 4 Φόρμα αναζήτησης με απενεργοποιημένο κουμπί	15
Εικόνα 5 Φόρμα αναζήτησης με ενεργοποιημένο κουμπί.....	16
Εικόνα 6 Campaign Refinement.....	17
Εικόνα 7 Campaign Refinement-Καρτέλα URLs	17
Εικόνα 8 Campaign Refinement-Καρτέλα Posts.....	18
Εικόνα 9 Campaign Refinement-Καρτέλα ExpandedURLs.....	18
Εικόνα 10 Campaign Refinement- Θεματικό νέφος	19
Εικόνα 11 Campaign Refinement – Query Terms	19
Εικόνα 12 Campaign Refinement – Launch Campaign	20
Εικόνα 13 Campaign Management	21
Εικόνα 14 Campaign Management- κουμπιά.....	22
Εικόνα 15 Campaign Viewer	22
Εικόνα 16 PopUp.jsp.....	23
Εικόνα 17 Μοντέλο οντοτήτων-συσχετίσεων	26
Εικόνα 18 Δομή των κλάσεων	28
Εικόνα 19 Σχεσιακό σχήμα	36
Εικόνα 20 Περιγραφή ενός αντικειμένου.....	52
Εικόνα 21 Περιγραφή ενός πίνακα.....	52
Εικόνα 22 Περιγραφή μιας τιμής	52
Εικόνα 23 Αναλυτικός τρόπος λειτουργίας των JSP.....	55
Εικόνα 24 Εγκατάσταση Tomcat	57
Εικόνα 25 Webapps directory του Tomcat.....	58
Εικόνα 26 Initial Screen	64
Εικόνα 27 Campaign Refinement.....	65
Εικόνα 28 Campaign Viewer	66
Εικόνα 29 Campaign Viewer - Θεματικό νέφος με 100 το πολύ στοιχεία.....	67
Εικόνα 30 - Campaign Viewer - Export JSON	67

Η σελίδα αυτή είναι σκόπιμα λευκή

1

Εισαγωγή

1.1 Εύρεση URL και πληροφοριών μέσω του twitter

Μία ιδιαίτερα ενδιαφέρουσα και ευρέως χρησιμοποιούμενη υπηρεσία είναι το twitter . Πρόκειται για μια ιστοσελίδα κοινωνικής δικτύωσης η οποία δημιουργήθηκε το 2006 και πάρα πολύ γρήγορα απέκτησε παγκόσμια φήμη, έχοντας εκατομμύρια ενεργούς χρήστες κάθε μέρα. Οι χρήστες μπορούν να διαβάσουν tweets - υποχρεωτικά μικρά μηνύματα - , να αναρτούν καθώς και να ακολουθούν άλλους χρήστες που τους προκαλούν το ενδιαφέρον. Αυτή η γρήγορη μορφή επικοινωνίας με μικρά μηνύματα ενός προς πολλούς, έδωσε το χαρακτηρισμό στο twitter σαν ένα <διαδικτυακό SMS>.

Ξεχωριστή σημασία παρουσιάζει το γεγονός ότι οι χρήστες πολλές φορές περιλαμβάνουν URLs στα tweets τους καθώς και hashtags, λέξεις κλειδιά που ξεκινούν με το πρόθεμα # και χρησιμοποιούνται για να ομαδοποιούν και να κατηγοριοποιούν δεδομένα. Με την επιλογή ενός hashtag, ο χρήστης έχει τη δυνατότητα να δει τα τελευταία tweets που το περιλαμβάνουν.

Ο τεράστιος όγκος πληροφοριών που βρίσκεται στο twitter και κυρίως τα URLs που αναρτώνται σε αυτό, το καθιστά κατάλληλο για να ενημερωθεί κάποιος για θέματα που τον ενδιαφέρουν.

1.2 Αντικείμενο διπλωματικής

Σκοπός της παρούσας διπλωματικής εργασίας είναι να εξαχθούν και να αξιοποιηθούν οι πληροφορίες που αναφέρθηκαν παραπάνω. Αυτό θα γίνει μέσω της ανάπτυξης μιας εφαρμογής που θα δίνει τη δυνατότητα στο χρήστη να ψάξει απευθείας στο twitter για κάποιο θέμα που τον αφορά χωρίς να χρησιμοποιήσει κάποια άλλη εξωτερική μηχανή αναζήτησης. Ο χρήστης μπορεί με την εισαγωγή πολλαπλών hashtags να ανακτήσει σχετικά posts και URLs, να προσαρμόσει και να περιορίσει την αναζήτηση του σε πράγματα που τον ενδιαφέρουν πιο πολύ, και τελικά να εκκινήσει μια campaign, για χρονικό διάστημα που ο ίδιος έχει ορίσει, η οποία θα του δώσει πρόσβαση σε όλα τα επιθυμητά δεδομένα που έχουν εμφανιστεί στο twitter .

1.2.1 Πρόβλημα

Λόγω των μεγάλων συσχετίσεων των URLs αυτών με παραμέτρους, όπως ο χρόνος ανάρτησης τους, η ταυτότητα και η γεωγραφική θέση των χρηστών που τα αναρτούν κλπ, η εξαγωγή αυτών των πληροφοριών θα ήταν προφανώς εξαιρετικά χρήσιμη καθώς από την ανάλυση τους προκύπτουν πολύτιμα συμπεράσματα χρήσιμα για ερευνητές, opinion miners, διαφημιστές, δημοσιογράφους κ.α.

Επομένως μια εφαρμογή που εντοπίζει τα URLs καθώς και ορισμένες από αυτές τις πληροφορίες και συσχετίσεις θα είχε διπλή σημασία: Από τη μια θα είχε μεγάλο ενδιαφέρον και πολλά περιθώρια ανάπτυξης από μόνη της σαν εναλλακτική μηχανή αναζήτησης για το web, ιδίως για γεγονότα και πρόσωπα της επικαιρότητας μιας και στο Twitter οι ειδήσεις εξαπλώνονται ταχύτατα. Από την άλλη η λειτουργικότητα της θα μπορούσε να γίνει μέρος κάποιας μεγαλύτερης πλατφόρμας, για πιο ολοκληρωμένη ανάλυση όλων των πληροφοριών του twitter .

1.2.2 Σκοπός

Σκοπός της διπλωματικής εργασίας είναι η εξαγωγή όλων αυτών των πληροφοριών από το twitter , η επεξεργασία μέρους αυτών –κυρίως των URLs - έτσι ώστε να είναι η πρόσβαση και η εποπτεία τους, φιλική και εύχρηστη και τέλος η μετατροπή τους σε μορφή αξιοποιήσιμη για περαιτέρω επεξεργασία. Αυτό έγινε μέσω της ανάπτυξης μιας εφαρμογής που να δίνει τη δυνατότητα στο χρήστη να ψάξει απευθείας στο Twitter για κάποιο θέμα που τον αφορά χωρίς να χρησιμοποιήσει κάποια άλλη εξωτερική μηχανή αναζήτησης. Ο χρήστης μπορεί με την εισαγωγή πολλαπλών hashtags να ανακτήσει σχετικά tweets και URLs, να προσαρμόσει και να περιορίσει ή να επεκτείνει την αναζήτηση του σε πράγματα που τον

ενδιαφέρουν πιο πολύ, με απώτερο σκοπό να εκκινήσει μια campaign, για χρονικό διάστημα που ο ίδιος έχει ορίσει, στο οποίο πραγματοποιείται περιοδική αναζήτηση για τη συλλογή των απαιτούμενων πληροφοριών

1.2.3 Βασικές Λειτουργίες

Αφού μελετήθηκε διεξοδικά το API του twitter, στα πλαίσια της διπλωματικής αναπτύχθηκε μια εφαρμογή που δίνει την δυνατότητα για :

1.μια γρήγορη αναζήτηση στο twitter, που επιστρέφει τα 50 τελευταία tweets που περιλαμβάνουν τα hashtags που έχει δώσει ο χρήστης. Τα URLs και τα posts θα αποθηκευτούν σε μια βάση δεδομένων για επεξεργασία και εύκολη ανάκτηση και θα παρουσιαστούν με τη μορφή λίστας.

2.προσαρμογή της ερώτησης έτσι ώστε να παρθούν τα δεδομένα που αφορούν περισσότερο το χρήστη. Αυτό γίνεται μέσω ενός θεματικού νέφους που δημιουργείται από τα πιο δημοφιλή hashtags που βρέθηκαν στη γρήγορη αναζήτηση που έγινε στο 1 καθώς και με τη χρήση φίλτρων που περιορίζουν τα tweets που θα επιστραφούν ως προς τη γλώσσα και τον τύπων τους (mixed, recent, popular).

3.εκκίνηση μιας campaign με μοναδικό όνομα την οποία θα ξεκινήσει ο χρήστης μετά την παραπάνω διαδικασία προσαρμογής. Η campaign θα τρέχει στον server για συγκεκριμένο χρονικό διάστημα που θα έχει δοθεί από το χρήστη, θα κάνει αίτηση στο twitter κάθε 15 λεπτά για ανάκτηση των 1500 τελευταίων tweet με τα εισαχθέντα hashtags, θα τα αποθηκεύει στη βάση δεδομένων και όταν τελειώσει, θα ενημερώνει το χρήστη μέσω mail. Στο τέλος τα urls και τα posts είναι διαθέσιμα προς παρουσίαση σε μορφή λίστας.

1.2.4 Συνεισφορά

Η συνεισφορά της διπλωματικής μας εργασίας μπορεί να συνοψιστεί ως εξής:

- Διεξοδική μελέτη των δυνατοτήτων που προσφέρει το twitter API, και ιδιαίτερα το twitter Search API, το οποίο και αποφασίσαμε να χρησιμοποιήσουμε για την εφαρμογή μας (έναντι του Rest API), λόγω τις πιο ακριβής αναζήτησης του για πρόσφατα tweets.
- Σύνδεση των URLs που εξάγουμε από το twitter με τις υπηρεσίες που προσφέρονται από το API του ιστοτόπου <http://longUrl.org> έτσι ώστε να γίνεται επέκταση όσων short URLs δε δημιουργήθηκαν από το twitter αλλά από άλλες υπηρεσίες.
- Crawling στο twitter για ανίχνευση πληροφοριών που η μετέπειτα επεξεργασία τους μπορεί να είναι ιδιαίτερης σημασίας.

- Αξιοποίηση των δυνατοτήτων και των πληροφοριών που μπορεί να προσφέρει το twitter για να χρησιμοποιηθεί ως μηχανή αναζήτησης για άμεσα αποτελέσματα.
- Δημιουργία μιας γραφικής διεπαφής χρήστη από το μηδέν με χρήση HTML, CSS, JSP, JQuery, με υλοποίηση δυναμικών πλαισίων, προσαρμοσμένων στη φύση της εφαρμογής έτσι ώστε να γίνεται η πλοήγηση σε αυτή με έναν εύκολο, πρακτικό και χρηστικό τρόπο.
- Υλοποίηση θεματικού νέφους έτσι ώστε να γίνεται η προσαρμογή του ερωτήματος της campaign ευκολότερη εποπτικά.
- Τέλος η επιστροφή όλων των δεδομένων που συλλέχθηκαν ή δημιουργήθηκαν κατά τη διάρκεια της campaign σε μορφή JSON έτσι ώστε ο χρήστης να έχει πληθώρα εργαλείων για να κάνει διεξοδική ανάλυση αυτών των δεδομένων.

1.3 Οργάνωση κειμένου

Το επόμενο κεφάλαιο αναλύει μια ευρύτερη εργασία πάνω στο twitter , μέρος της πλατφόρμας της οποίας ορίζεται υλοποιείται στη παρούσα διπλωματική. Στο κεφάλαιο 3 μελετούνται τα twitter API και LongUrl API. Στο 4 παρουσιάζονται οι απαιτήσεις συστήματος της εφαρμογής και στο 5 η αρχιτεκτονική της. Ενδιαφέρουσες λεπτομέρειες υλοποίησης, όπως επίσης και εγχειρίδιο εγκατάστασης δίνονται στο 6. Τέλος στο 7 παρατίθεται ένα σενάριο λειτουργίας ως μέρος του έλεγχου, ακολουθούμενο από τον επίλογο στο 8.

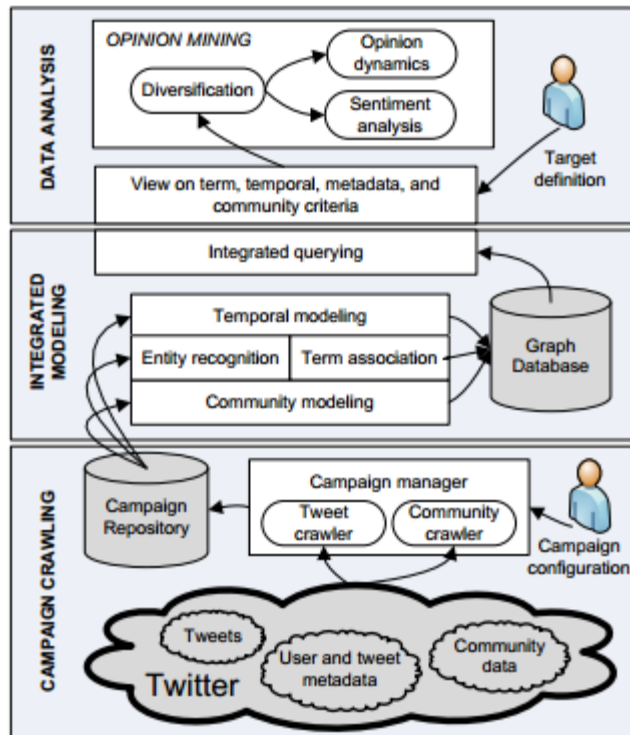
2

Σχετικές εργασίες

Η ραγδαία ανάπτυξη των OSN (online social networks) και ειδικότερα του twitter και η συνεχής παραγωγή περιεχομένου από αυτό έχει οδηγήσει σε μια πληθώρα πρωτοποριακών εφαρμογών που βασίζονται στην επεξεργασία αυτού του περιεχομένου. Παρόλα αυτά για την μέγιστη αξιοποίηση όλων αυτών των πληροφοριών πρέπει να αναπτυχθεί μια ολοκληρωμένη πλατφόρμα που να υποστηρίζει την ανάλυση πληροφοριών στο twitter . Η εφαρμογή που δημιουργήθηκε στα πλαίσια αυτής της διπλωματικής εργασίας, έχει σχεδιαστεί έτσι ώστε να μπορεί να ενταχθεί σε μια τέτοια γενικότερη πλατφόρμα .

2.1 Πλατφόρμα υποστήριξης ανάλυσης δεδομένων στο twitter

Η προσέγγιση και ο σκοπός αυτής της πλατφόρμας είναι να μπορεί να αναλαμβάνει ποικίλα καθήκοντα ανάλυσης δεδομένων και να παρέχει την απαραίτητη δομή και τις υπηρεσίες έτσι ώστε να μπορούν τα δεδομένα αυτά να συλλεχθούν και να διαχειριστούν εύκολα. Η διαστρωματική δομή αυτής της πλατφόρμας φαίνεται καθαρά στο παρακάτω σχήμα:



Εικόνα 1 Αρχιτεκτονική της πλατφόρμας [TWP]

Η εφαρμογή που αναπτύχθηκε μπορεί να ενταχτεί στη πλατφόρμα σαν μέρος του campaign crawling και συγκεκριμένα για την εξαγωγή Tweets και ορισμένων user και tweet metadata τα οποία μετά την κατάλληλη τροποποίηση μπορούν να εισαχθούν στο Campaign Repository από το οποίο μετά εισάγονται στα ανώτερα επίπεδα. Επιπλέον η πτυχιακή μας ακολουθεί τη μεθοδολογία αυτής της πλατφόρμας όσον αφορά την εκκίνηση μιας campaign. Δίνει δηλαδή πολύ μεγάλη σημασία στο ερώτημα αναζήτησης. Έτσι υποστηρίζει και επιβάλλει την ακόλουθη λειτουργία:

Σε κάθε αναζήτηση πρέπει αρχικά να αναλυθεί ένα μικρό δείγμα από tweets έτσι ώστε να δώσει μια γενική εικόνα στο χρήστη για τα αποτελέσματα που μπορεί να περιμένει και να του προσφέρει ποικίλα εργαλεία έτσι ώστε να προσαρμόσει την αναζήτηση του και όταν είναι έτοιμος να εκκινήσει την campaign να έχει σχηματίσει μια ικανοποιητική εικόνα για το τι αποτελέσματα μπορεί να περιμένει [TWP].

Τέλος η πλατφόρμα αυτή επιβάλλει την εκκίνηση αλλά και τη λειτουργία πολλαπλών campaign ταυτόχρονα [TWP].

3

Υπόβαθρο

Σε αυτό το κεφάλαιο θα αναφερθούμε στη δομή των υπηρεσιών που αξιοποιήσαμε για την υλοποίηση της εφαρμογής μας.

3.1 Twitter APIs

Η υπηρεσία του twitter , προσφέρει περισσότερα από ένα APIs που το καθένα εξυπηρετεί διαφορετικούς σκοπούς και απευθύνεται στους developers για συγκεκριμένες δουλειές. Αναλυτικότερα υπάρχουν τα εξής APIs:

3.1.1 Twitter for Websites

Η συγκεκριμένη διεπαφή προσφέρει μια σειρά από προϊόντα που προσφέρουν τη δυνατότητα σε ιστοσελίδες να χρησιμοποιούν το Twitter για πολύ βασικές λειτουργίες [TWT1].

3.1.2 *REST API*

Το REST API δίνει τη δυνατότητα στους προγραμματιστές να έχουν πρόσβαση σε μερικές από τις βασικές λειτουργίες όπως οι πληροφορίες χρήστη και οι ανανεώσεις κατάστασης. Προσφέρει δυνατότητες χειρισμού όλων αυτών των στοιχείων καθώς και άλλων παρεμφερών, όπως πχ την αλλαγή εικόνας και την προσθήκη σε αγαπημένα συγκεκριμένων Tweets [TWT1].

3.1.3 *Streaming API.*

Το Streaming Api απευθύνεται σε όσους θέλουν να συλλέγουν μεγάλη ποσότητα πληροφοριών σε πραγματικό χρόνο. Συνήθως χρησιμοποιείται για εφαρμογές που συλλέγουν δεδομένα για αναλυτική έρευνα. Ο λόγος που δεν το χρησιμοποιήσαμε στην εφαρμογή μας είναι γιατί ψάχνει μόνο στο 1% των tweets που δημιουργούνται και γιατί δε μπορεί να ψάξει μόνο για hashtag (μια αναζήτηση για hashtag επιστρέφει και τις λέξεις χωρίς το «#») [TWT1].

3.1.4 *Search API.*

Το Search Api, που είναι και το Api που χρησιμοποιήσαμε για την ανάπτυξη της εφαρμογής μας, χρησιμοποιείται για αναζήτηση πραγματικού χρόνου στα πρόσφατα Tweets. Το Search Api έχει τις πιο πολλές επιλογές για αναζήτηση tweets, αλλά έχει και ορισμένους περιορισμούς. Επιστρέφει μόνο Tweets που έχουν δημιουργηθεί το πολύ μια εβδομάδα από τη στιγμή της αναζήτησης, ορισμένες φορές δε μπορεί να επεξεργαστεί περίπλοκες ερωτήσεις, οι ερωτήσεις είναι υποχρεωτικά ανώνυμες και έχουν το πολύ 1000 χαρακτήρες μήκος. Τέλος μπορεί να βρει το πολύ τα 1500 τελευταία tweets που σχετίζονται με την αναζήτηση του. Ας μελετήσουμε όμως λίγο πιο εξονυχιστικά το συγκεκριμένο API [TWT2].

Η διεύθυνση URL για να χρησιμοποιήσουμε το Search API έχει τη μορφή <http://search.twitter.com/search.format>, όπου format είναι ο τύπος επιστροφής των αρχείων. Εμείς χρησιμοποιήσαμε για τύπο, αρχεία JSON. Στη συνέχεια προσθέτουμε τις ακόλουθες παραμέτρους:

- **q (υποχρεωτική).** Το βασικό ερώτημα το οποίο πρέπει να είναι σε μορφή UTF8.
- **callback.** Αν χρησιμοποιηθεί, η απάντηση θα χρησιμοποιήσει τον τύπο JSONP. Χρησιμοποιείται μόνο σε αιτήσεις JSON.

- **geocode.** Επιστρέφει Tweets τα οποία μπορούν να εντοπιστούν μέσα στην ακτίνα ενός συγκεκριμένου σημείου δοσμένου του γεωγραφικού μήκους/πλάτους του. Η παράμετρος καθορίζεται από τη τιμή «γεωγραφικό πλάτος, γεωγραφικό μήκος, ακτίνα» και η ακτίνα καθορίζεται αν είναι μίλια «mi» ή χιλιόμετρα «km». Παράδειγμα τιμής: : 37.781157,-122.398720,1mi.
- **lang.** Επιστρέφει τα tweets που είναι γραμμένα στην επιθυμητή γλώσσα. Η γλώσσα δίνεται με κωδικοποίηση ISO 639-1.
- **locale.** Καθορίζει τη γλώσσα στην οποία στέλνεται η ερώτηση (υποστηρίζει μόνο Ιαπωνική γλώσσα).
- **page.** Ο αριθμός της σελίδας.
- **result_type.** Επιστρέφει τον τύπο των Tweets που προτιμά κάποιος να του επιστραφούν. Η προκαθορισμένη τιμή είναι «mixed» και οι τιμές που μπορεί να πάρει είναι «mixed», «recent» (επιστρέφει μόνο τα πιο πρόσφατα Tweets) και «popular» (επιστρέφει μόνο τα πιο δημοφιλή Tweets).
- **rpp.** Καθορίζει τον αριθμό των Tweets που θα υπάρχουν σε σελίδα με μέγιστο το 100.
- **show_user.** Όταν έχει τιμή true τοποθετεί στην αρχή του Tweet το σύμβολο «:».
- **until.** Επιστρέφει Tweets που δημιουργήθηκαν πριν από τη δοσμένη ημερομηνία. Η ημερομηνία έχει μορφή YYYY-MM-DD. Πχ 2010-03-28.
- **since_id.** Επιστρέφει αποτελέσματα με ID μεγαλύτερο από (δηλαδή πιο πρόσφατο από) το συγκεκριμένο ID. Εάν το όριο των Tweets έχει ξεπεράσει το τελευταίο id, το since_id θα καλέσει το παλιότερο ID που είναι διαθέσιμο.
- **max_id.** Επιστρέφει αποτελέσματα με ID μικρότερο από (δηλαδή παλιότερο από) ή ίσο με το δοσμένο ID.
- **include_entities.** Όταν πάρει τιμή true ή t ή 1, κάθε Tweet περιλαμβάνει ένα κόμβο που λέγεται entities. Ο κόμβος προσφέρει μια ποικιλία από μετα-δεδομένα σχετικά με το Tweet σε διακριτή δομή περιλαμβάνοντας: URLs, media, hashtags. Να σημειωθεί ότι δεν υπάρχει πρόσβαση στην πληροφορία «user_mentions» και ότι η συγκεκριμένη παράμετρος υποστηρίζεται μόνο για format JSON. [TWT3]

Ένα παράδειγμα χρησιμοποίησης του API είναι το ακόλουθο:

http://search.twitter.com/search.json?q=USElections&rpp=1&page=7&since_id=12345

Εδώ επιλέγουμε τύπο JSON με ερώτημα USElections που να επιστρέφει Tweets με id μεγαλύτερο από 12345, ένα Tweet ανά σελίδα και εμείς θέλουμε την 7^η σελίδα. Η υπηρεσία του Twitter επιστρέφει το εξής JSON:

```
{
  "completed_in":0.055,
  "max_id":258878586503696384,
  "max_id_str":"258878586503696384",

  "next_page":"?page=8&max_id=258878586503696384&q=USElections&rpp=1",
  "previous_page":
    "?page=6&max_id=258878586503696384&q=USElections&rpp=1",
  "page":7,
  "query":"USElections",
  "refresh_url":
    "?since_id=258878586503696384&q=USElections",
  "results":[{"created_at":"Thu, 18 Oct 2012 10:33:59+0000",
    "from_user":"EstherStanhope1",
    "from_user_id":465559145,
    "from_user_id_str":"465559145",
    "from_user_name":"Esther Stanhope", "geo":null,
    "id":258878586503696384,
    "id_str":"258878586503696384",
    "iso_language_code":"en",
    "metadata":{"result_type":"recent"}],

  "profile_image_url":"http://a0.twimg.com/profile_images/1759156450/esther_small_normal.jpg",

  "profile_image_url_https":"https://si0.twimg.com/profile_images/1759156450/esther_small_normal.jpg",
  "source":"&lt;a href=&quot;http://twitter.com/&quot;&gt;web&lt;/a&gt;",
  "text":"@LoveTweeting1 Thank You For RT:Analysis: Michelle Obama Vs. Ann Romney ~ #PresenterCoach #Presentation #USElections ~ http://t.co/xuFFmOA4",
  "to_user":"LoveTweeting1",
  "to_user_id":323948655,
  "to_user_id_str":"323948655", "to_user_name":"LoVe"}],
  "results_per_page":1,
  "since_id":12345,
  "since_id_str":"12345"
}
```

Παράδειγμα χρήσης του Twitter Search API

Τέλος αξίζει να αναφέρουμε ότι αναφερόμαστε στην έκδοση 1.0 του Search API, γιατί χρησιμοποιώντας αυτή τη διεπαφή υλοποιήσαμε την εφαρμογή μας, ενώ πρόσφατα, το Twitter δημιούργησε και καινούρια έκδοση την 1.1 που έχει διαφορετική δομή και οργάνωση. Αυτή τη στιγμή το Twitter προσφέρει και ως 2 εκδόσεις διαθέσιμες ως προγραμματιστές.

3.2 LongURL API

Για την επέκταση των URL αξιοποιήθηκε η διεπαφή ας υπηρεσίας LongURL (<http://longurl.org>).

Ο τρόπος με τον οποίο μπορεί κάποιος να χρησιμοποιήσει την υπηρεσία είναι να κάνει μια HTTP αίτηση του τύπου: <http://api.longurl.org/v2/expand> [LU].

Οι παράμετροι που δέχεται το API είναι οι παρακάτω:

- **url (υποχρεωτικό).** Το URL που πρόκειται να επεκταθεί. Πρέπει να ξεκινά με http:// ή με https:// και να είναι κωδικοποιημένο σα URL.
- **all-redirects.** Αν η παράμετρος πάρει τη τιμή 1 ας οι HTTP ανακατευθύνσεις περιλαμβάνονται στην απάντηση.
- **content-type.** Όταν η τιμή του γίνεται 1, τότε περιλαμβάνεται το MIME type του ζητούμενου URL στην απάντηση.
- **response-code.** Όταν η τιμή του γίνεται 1, τότε περιλαμβάνεται ο HTTP κωδικός απόκρισης του ζητούμενου URL στην απάντηση.
- **title.** Όταν η τιμή του γίνεται 1, τότε περιλαμβάνεται και ο τίτλος HTML του ζητούμενου URL στην απάντηση.
- **rel-canonical.** Όταν η τιμή του γίνεται 1, τότε περιλαμβάνεται και το Canonical URL του ζητούμενου URL στην απάντηση.
- **meta-keywords.** Όταν η τιμή του γίνεται 1, τότε περιλαμβάνονται και οι λέξεις κλειδιά του ζητούμενου URL στην απάντηση.
- **meta-description.** Όταν η τιμή του γίνεται 1, τότε περιλαμβάνεται και η περιγραφή του ζητούμενου URL στην απάντηση.
- **format.** Αυτή η παράμετρος καθορίζει τον τύπο ας απάντησης. Μπορεί να είναι xml(προκαθορισμένο), json ή php.
- **callback.** Χρησιμοποιείται για JSONP. Απαιτεί η παράμετρος **format** να είναι ίση με JSON [LU].

Ένα άλλο σημαντικό μέρος του LongURL API είναι οι απαντήσεις λάθους:

- **400 Bad Request.** Σημαίνει ότι υπάρχει λάθος στην αίτηση.
- **404 Not Found.** Σημαίνει ότι ζητάμε κάτι που δεν υπάρχει ή ότι χρησιμοποιούμε άκυρη έκδοση του API.
- **500 Internal Server Error.** Σημαίνει ότι υπήρξε κάποιο πρόβλημα στον Server.

- **503 Service Unavailable.** Σημαίνει ότι τη συγκεκριμένη χρονική στιγμή η υπηρεσία είναι μη διαθέσιμη. Συνίσταται σε περίπτωση λάθους τέτοιου τύπου ο αιτών να δοκιμάζει αργότερα.
- **504 Bad Gateway.** Σημαίνει ότι υπήρξε κάποιο πρόβλημα στην επικοινωνία των servers. Αυτό συμβαίνει συνήθως όταν η υπηρεσία που δημιουργεί τα short URL είναι μη διαθέσιμη [LU].

Ας εξετάσουμε τώρα την απάντηση στη συγκεκριμένη αίτηση:

<http://api.longurl.org/v2/expand?url=http://bit.ly/RynMwP&format=json&title=1>

Θέσαμε **format** JSON, τη παράμετρο τίτλου και προφανώς εισάγαμε και το URL. Ακολουθεί η απάντηση που λάβαμε:

```
{  
  «long-  
url»:»http://www.guardian.co.uk/commentisfree/2012/oct/16/us-  
presidential-debates-us-elections-2012-candy-crowley-obama-romney»,  
  "title": "Ungovernable reality, not Candy Crowley, threatens Obama and  
Romney | Comment is free | guardian.co.uk"  
}
```

Παράδειγμα χρήσης του LongURL API

4

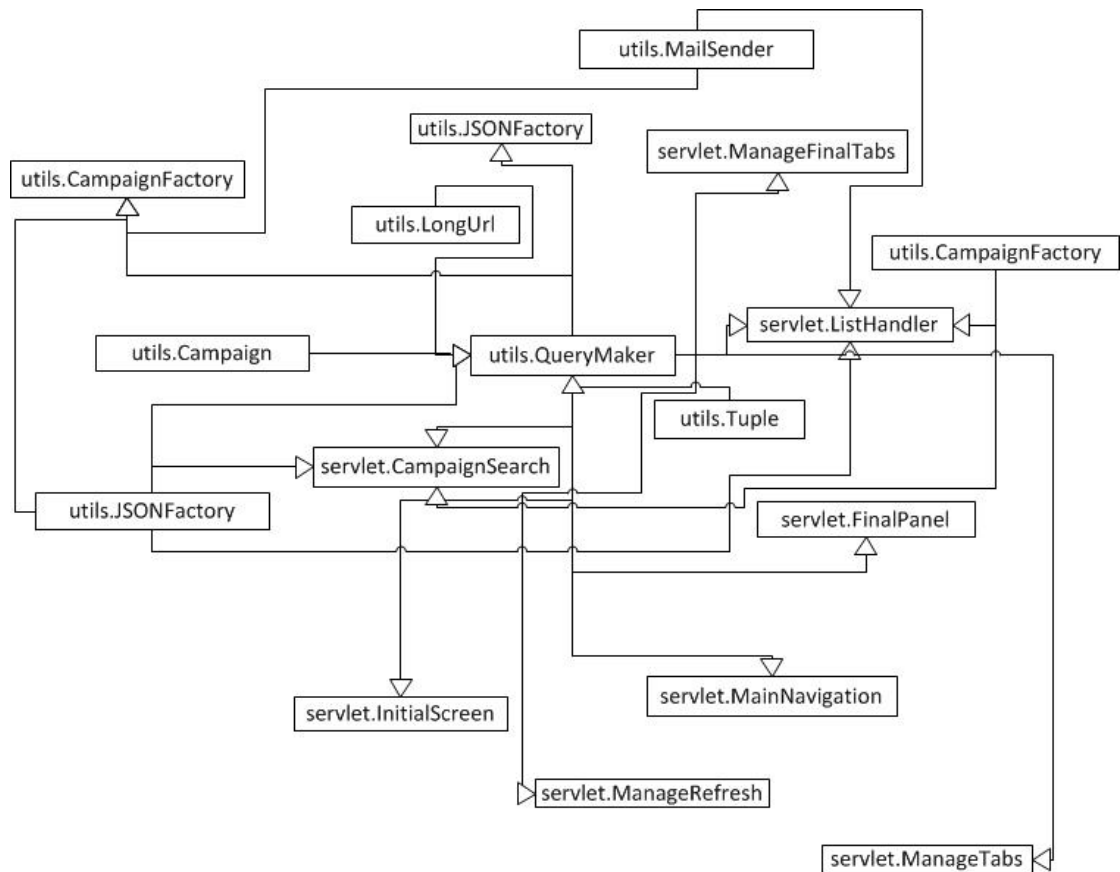
Ανάλυση Απαιτήσεων Συστήματος

Ακολουθεί η περιγραφή της αρχιτεκτονικής του συστήματος και η ανάλυση απαιτήσεων για τις λειτουργίες του.

4.1 Αρχιτεκτονική

Το σύστημα μας αποτελείται από τα εξής επιμέρους κομμάτια:

- Απλή αναζήτηση στο Twitter με την εισαγωγή ενός ή περισσότερων hashtags.
- Campaign refinement, όπου η ερώτηση μπορεί να αλλάξει και να προσαρμοστεί ώστε να είναι πιο κοντά σε αυτό που θέλει να βρει ο χρήστης. Με την ολοκλήρωση του δίνεται η δυνατότητα για εκκίνηση μιας campaign.
- Campaign management, όπου γίνεται διαχείριση όλων των campaigns που έχουν μπει στη βάση δεδομένων. Η διαχείριση είναι διαφορετική και εξαρτάται με το αν η εκάστοτε campaign έχει τελειώσει, αν έχει διακοπεί η αν τρέχει ακόμα.
- Campaign view, όπου δίνεται η δυνατότητα στο χρήστη να δει τα διάφορα δεδομένα που έχουν συλλεχτεί με το πέρας μιας campaign.



Εικόνα 2 Block διάγραμμα

4.2 Περιγραφή Λειτουργιών

Περιγράφονται παρακάτω αναλυτικά οι λειτουργίες που εκτελεί το κάθε υποσύστημα της εφαρμογής:

4.2.1 Απλή αναζήτηση

Σε αυτό το αρχικό σημείο, ο χρήστης καλείται να εισάγει ένα ή περισσότερα hashtags για μια διερευνητική αναζήτηση στο twitter. Η αναζήτηση αυτή θα επιστρέψει τα 50 τελευταία tweets που τα περιέχουν και είναι καθαρά εποπτική. Σκοπός της δεν είναι στο να ικανοποιήσει τον χρήστη ως προς την ερώτηση, αλλά να τον παρακινήσει στο να ξεκινήσει τη διαδικασία προσαρμογής, την campaign refinement που θα αναλυθεί παρακάτω. Η φόρμα φαίνεται στο παρακάτω screenshot. Στο κενό πεδίο συμπληρώνονται τα hashtags και με το πάτημα του κουμπιού διεκπεραιώνεται η αναζήτηση και γίνεται μετάβαση στο επόμενο βήμα.



Campaign Management



Εικόνα 3 Κεντρική Σελίδα

Στο σημείο αυτό αξίζει να σημειωθεί ότι όταν το πεδίο είναι κενό, το κουμπί εκκίνησης είναι απενεργοποιημένο ώστε να μην επιτρέπεται η μετάβαση στην επόμενη σελίδα. Με την εισαγωγή χαρακτήρων, το κουμπί μπορεί πλέον να πατηθεί, όπως φαίνεται χαρακτηριστικά στις παρακάτω εικόνες.



Campaign Management

Εικόνα 4 Φόρμα αναζήτησης με απενεργοποιημένο κουμπί



Campaign Management

Εικόνα 5 Φόρμα αναζήτησης με ενεργοποιημένο κουμπί

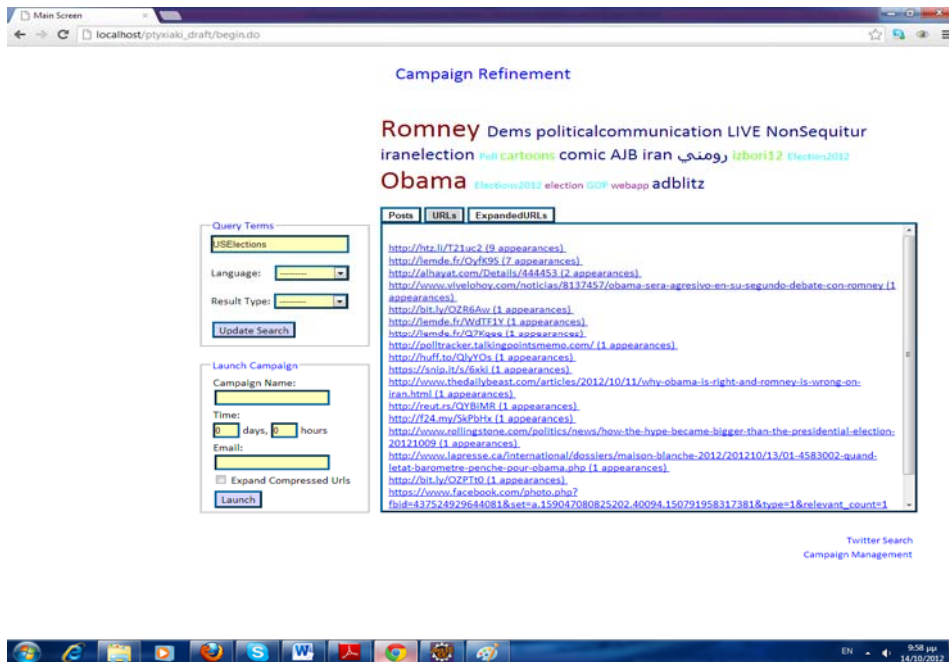
Ένα άλλο ενδιαφέρον χαρακτηριστικό της αναζήτησης είναι ότι είτε ο χρήστης κάνει αναζήτηση για μια λέξη-κλειδί με το πρόθεμα «#» είτε όχι, η εφαρμογή μετατρέπει αυτόματα την ερώτηση αναζήτησης σε ερώτηση για hashtags και όχι οποιοδήποτε λέξη. Επίσης αν γραφούν δυο λέξεις ή και περισσότερες χωριζόμενες με κενό τότε το εύρος αναζήτησης επεκτείνεται για όλα τα αντίστοιχα hashtags με τη πράξη του λογικού «ή» μεταξύ τους.

Τέλος να αναφέρουμε ότι υπάρχει ένας σύνδεσμος στο Campaign Management, λειτουργεία που θα αναλυθεί αργότερα.

4.2.2 Campaign Refinement

Ο χρήστης τώρα καλείται να προσαρμόσει το ερώτημα του, εισάγοντας πιο κατάλληλα hashtags και φιλτράροντας το. Αφού εκτελέσει πολλαπλές μικρές αναζητήσεις και βεβαιωθεί ότι τα αποτελέσματα (δηλαδή τα posts και τα urls που επιστρέφονται) ικανοποιούν τις απαιτήσεις του ως προς αυτό που θέλει να ψάξει, συμπληρώνει μια φόρμα και εκκινεί τη campaign του.

Στο παρακάτω screenshot βλέπουμε τη σελίδα του Campaign Refinement, όπως προκύπτει μετά από μια απλή αναζήτηση (όπως περιγράφηκε στην 4.2.1). Η αναζήτηση έγινε για το hashtag USElections.

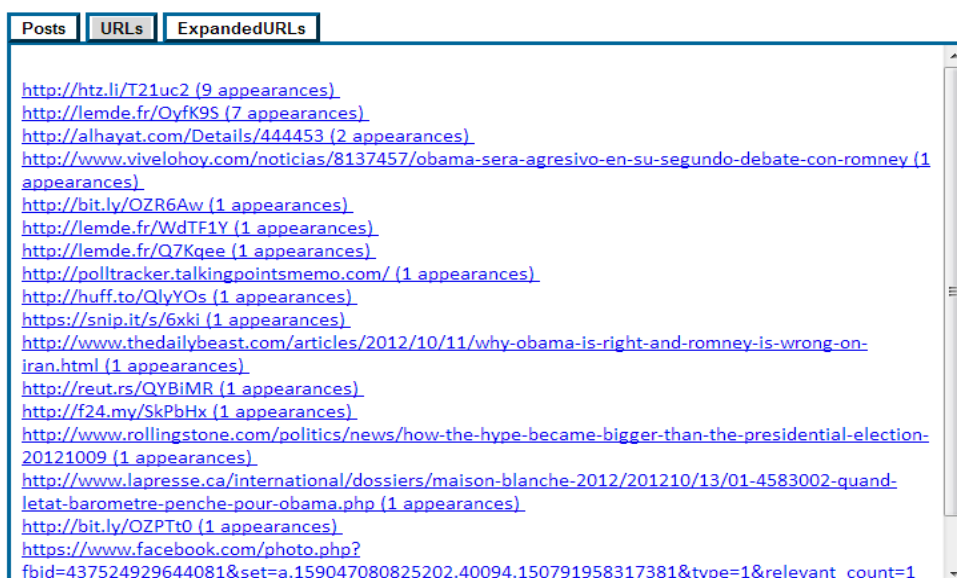


Εικόνα 6 Campaign Refinement

4.2.2.1 Εμφάνιση url και post

Στο συγκεκριμένο σημείο της διεπαφής, ο χρήστης βλέπει τα url και τα posts που επέστρεψε η απλή αναζήτηση. Πρόκειται για ένα παράθυρο που χωρίζεται σε 3 καρτέλες, με την πρώτη να απεικονίζει τα posts, τη δεύτερη (και προεπιλεγμένη) τα urls και την τρίτη τα urls αφού περάσουν από μία λειτουργία επέκτασης. Η λειτουργία της επέκτασης μπορεί να θεωρηθεί απαραίτητη καθώς πολλές φορές τα urls έχουν συμπιεσθεί έτσι ώστε να απεικονίζονται με λιγότερους χαρακτήρες στο εκάστοτε tweet.

Παρακάτω παρατίθενται 3 screenshots που δείχνουν τις 3 αυτές καρτέλες



Εικόνα 7 Campaign Refinement-Καρτέλα URLs

Posts	URLs	ExpandedURLs
dampniat88: RT @LeMonde_USA: A l'intérieur du QG de campagne d'Obama http://t.co/mTsG3oBb #USelections		
jeanJacquesNGOM: RT @LeMonde_USA: A l'intérieur du QG de campagne d'Obama http://t.co/mTsG3oBb #USelections		
MA_L_F: RT @LeMonde_USA: A l'intérieur du QG de campagne d'Obama http://t.co/mTsG3oBb #USelections		
Wilfried_Uchiha: RT @LeMonde_USA: A l'intérieur du QG de campagne d'Obama http://t.co/mTsG3oBb #USelections		
Doremipunkeke: RT @LeMonde_USA: A l'intérieur du QG de campagne d'Obama http://t.co/mTsG3oBb #USelections		
virgile57: RT @LeMonde_USA: A l'intérieur du QG de campagne d'Obama http://t.co/mTsG3oBb #USelections		
lemondefr: RT @LeMonde_USA: A l'intérieur du QG de campagne d'Obama http://t.co/mTsG3oBb #USelections		
jreyest: #Obama será 'agresivo' en su segundo debate con #Romney http://t.co/9O55E302 via @vivelohoy #Elections2012 #USElections		
Martens_Erik: This year, #GOP and #Dems together air over 800,000 ads, of which 65% have negative tone #USelections #adblitz #politicalcommunication		
wespeaknews: #uselections Biden's Bite, by @JamilMShawwa. http://t.co/dII8bzTQ		
Nissy0743: "Des foules de plus en plus importantes aux meetings de Romney http://t.co/vMV5FEcA #USelections"		

Εικόνα 8 Campaign Refinement-Καρτέλα Posts

Posts	URLs	ExpandedURLs
	http://www.haaretz.com/news/u-s-elections-2012/poll-obama-grabs-wide-lead-among-early-voters-1.469922 (9 appearances)	
	http://www.lemonde.fr/elections-americaines/twitter/2012/10/14/a-l-interieur-du-qq-de-campagne-d-obama_1775202_829254.html?utm_source=dlvr.it&utm_medium=twitter (7 appearances)	
	http://alhayat.com/Details/444453 (2 appearances)	
	http://www.vivelohoy.com/noticias/8137457/obama-sera-agresivo-en-su-segundo-debate-con-romney (1 appearances)	
	http://www.wespeaknews.com/politics/bidens-bite-103162.html (1 appearances)	
	http://www.lemonde.fr/elections-americaines/breve/2012/10/13/des-foules-de-plus-en-plus-importantes-aux-meetings-de-romney_1774998_829254.html?utm_source=dlvr.it&utm_medium=twitter (1 appearances)	
	http://www.lemonde.fr/elections-americaines/breve/2012/10/13/une-marche-pour-sauver-big-bird-bientot-a-washington_1774999_829254.html?utm_source=dlvr.it&utm_medium=twitter (1 appearances)	
	http://polltracker.talkingpointsmemo.com/ (1 appearances)	
	http://www.huffingtonpost.com/2012/10/14/black-people-obama-2012-support-prejudice_n_1964609.html?ncid=edlinkusaolp00000003 (1 appearances)	
	https://snip.it/collections/31086-US-Political-Comedy?highlight=323442&ref=external (1 appearances)	
	http://www.thedailybeast.com/articles/2012/10/11/why-obama-is-right-and-romney-is-wrong-on-iran.html (1 appearances)	
	http://www.reuters.com/article/2012/10/14/us-usa-campaign-poll-idUSBRE8991FR20121014 (1 appearances)	
	http://www.france24.com/en/elections-us-mobile?	

Εικόνα 9 Campaign Refinement-Καρτέλα ExpandedURLs

Αξίζει να αναφερθεί ότι τα τόσο τα urls όσο και τα expanded urls εμφανίζονται ταξινομημένα με βάση το πόσες φορές συναντιούνται στα tweets που μας δίνει η απλή αναζήτηση. Ο αριθμός εμφανίσεων φαίνεται δίπλα από κάθε url.

4.2.2.2 Προσαρμογή της ερώτησης

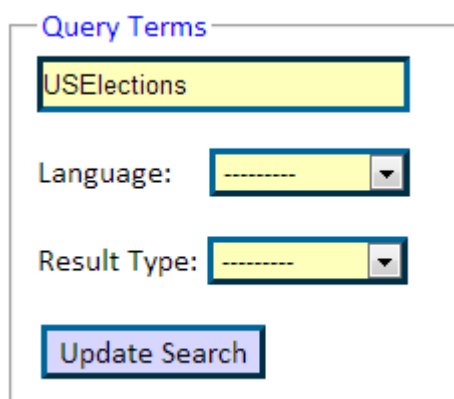
Το υποσύστημα αυτό έχει σκοπό να βοηθήσει τον χρήστη στο να προσαρμόσει την αναζήτηση του περισσότερο στο αντικείμενο που θέλει και να πάρει πιο ικανοποιητικά αποτελέσματα. Το παραπάνω καθίσταται δυνατό με επιλογή καλύτερου συνδυασμού hashtags προς αναζήτηση.

Η συγκεκριμένη λειτουργία πραγματοποιείται με τη δημιουργία ενός θεματικού νέφους κατά τη διάρκεια της πρώτης διερευνητικής αναζήτησης. Το νέφος αυτό περιλαμβάνει σχετικά με την αναζήτηση hashtags και συγκεκριμένα τα 20 πιο συχνά εμφανιζόμενα στα tweets που επιστραφήκαν.

Election2012 رومني Dems NonSequitur Poll cartoons iranelection LIVE
Romney GOP izbori12 iran election Elections2012 comic AJB
politicalcommunication adblitz webapp Obama

Εικόνα 10 Campaign Refinement- Θεματικό νέφος

Με την επιλογή οποιουδήποτε από αυτά, το hashtag προστίθεται στην πεδίο αναζήτησης και δίνεται με το πάτημα ενός κουμπιού η δυνατότητα νέας αναζήτησης.



Query Terms

USElections

Language: [-----]

Result Type: [-----]

Update Search

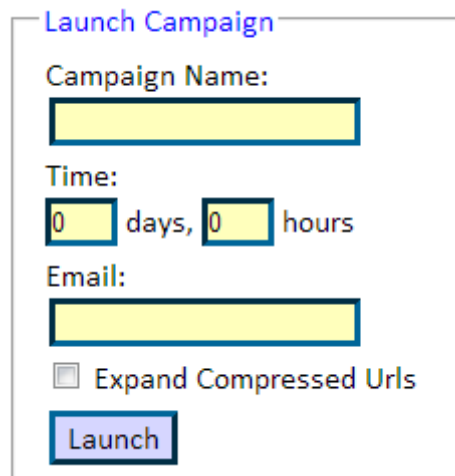
Εικόνα 11 Campaign Refinement – Query Terms

Στο ίδιο υποσύστημα υπάρχουν και 2 φίλτρα για περιορισμό τους αναζήτησης ως τους τη γλώσσα των tweets που θα επιστραφούν και ως προς τον τύπο τους (mixed,popular,recent).

4.2.2.3 Εκκίνηση του campaign

Όταν μετά από πολλαπλές αναζητήσεις, τα αποτελέσματα ικανοποιούν το χρήστη, αυτός καλείται να ξεκινήσει μια campaign η οποία θα κάνει κάθε 15 λεπτά αίτηση στο Twitter για

την ανάκτηση Tweets, για συγκεκριμένο χρονικό διάστημα. Το διάστημα αυτό δίνεται από χρήστη, μαζί με ένα μοναδικό όνομα, ένα Mail για ενημέρωση με το πέρας τους campaign, καθώς και με την επιλογή για μετατροπή των Url στην επεκταμένη τους μορφή.



The image shows a web form titled "Launch Campaign". It contains the following elements:

- A text input field for "Campaign Name".
- A "Time:" section with two numeric input fields: "0" for "days" and "0" for "hours".
- A text input field for "Email".
- A checkbox labeled "Expand Compressed Urls".
- A blue "Launch" button.

Εικόνα 12 Campaign Refinement – Launch Campaign

4.2.3 Campaign Management

Πρόκειται για ένα πολύ σημαντικό υποσύστημα, κύριος ρόλος του οποίου είναι η διαχείριση των διαφόρων campaigns που βρίσκονται στη βάση. Οι campaign αυτές μπορούν να βρίσκονται σε 3 καταστάσεις και μπορούν να διαχειριστούν αναλόγως. Είναι οι εξής:

1. Pending
2. Finished
3. Interrupted

Στο παρακάτω παράθυρο φαίνονται 3 campaigns με διαφορετικές καταστάσεις, όπως επίσης και οι λειτουργίες που μπορεί ο χρήστης να κάνει σε αυτές.

Campaign Management

The screenshot displays a 'Campaign Management' interface. It features a list of three campaigns, each with its name, search quote, start date, duration, and status. Below the list are several control buttons: 'view', 'delete', 'stop now', and 'export JSON'. Additionally, there are two control panels for 'update deadline' and 'relaunch interrupted', each with a 'Time:' label and input fields for 'days' and 'hours'.

Campaign Name	Search Quote	Start Date	Duration	Status
obama	obama	Sun Oct 14 18:49:31 EEST 2012	0 days and 2 hours	Finished
pokemon	pokemon	Sun Oct 14 18:52:12 EEST 2012	0 days and 2 hours	Interrupted
USElections	USElections	Tue Oct 16 22:53:48 EEST 2012	0 days and 1 hours	Pending

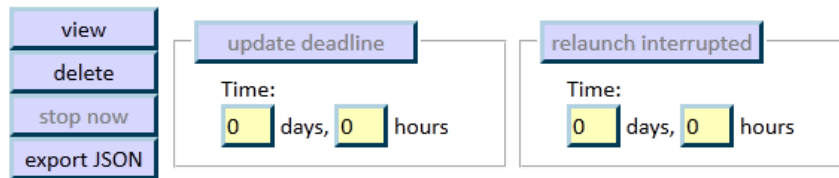
[Twitter Search](#)

Εικόνα 13 Campaign Management

Στην πρώτη κατηγορία (Pending) βρίσκονται οι campaigns που δεν έχουν ακόμα ολοκληρωθεί και τρέχουν ακόμα στον server. Πάνω σε αυτές, ο χρήστης μπορεί να επενεργήσει με 3 τρόπους: μπορεί να τις σταματήσει πατώντας το κουμπί stop now, να ανανεώσει το χρονικό διάστημα για το οποίο θα τρέχουν, εισάγοντας νέα ώρα η να στις διαγράψει.

Στη δεύτερη κατηγορία (Finished) βρίσκονται αυτές που έχουν ολοκληρωθεί. Με το πάτημα του κουμπιού view, όταν είναι επιλεγμένη μια τέτοια campaign, ο χρήστης μεταβαίνει σε ένα άλλο παράθυρο και βλέπει τα αποτελέσματα που επιστράφηκαν. Η λειτουργία αυτή θα αναλυθεί μετά. Επίσης του δίνεται η δυνατότητα να δει κάποια JSON αρχεία, λειτουργία που στις θα αναλυθεί και αυτή αργότερα. Τέλος, δίνεται η δυνατότητα διαγραφής.

Στη Τρίτη κατηγορία (Interrupted) βρίσκονται οι campaigns που σταμάτησαν πριν ολοκληρωθεί ο χρόνος που έπρεπε να εκτελεστούν και στις οποίες δεν έχει επέμβει ο χρήστης πατώντας το κουμπί stop now. Ο λόγος διακοπής είναι απρόβλεπτος και συνήθως αφορά προβλήματα που εμφανιστήκαν στη πλευρά του server (π.χ. προσωρινή παύση λειτουργίας του server ενώ η campaign είναι σε κατάσταση pending). Ο χρήστης μπορεί είτε να διαγράψει μια τέτοια campaign, ή να την επανεκκινήσει, εισάγοντας νέα ώρα και πατώντας το κουμπί relaunch interrupted.



Twitter Search

Εικόνα 14 Campaign Management- κουμπιά

4.2.4 Campaign viewer

Το υποσύστημα αυτό αποτελεί και το πιο βασικό υποσύστημα, στο οποίο καταφεύγει ο χρήστης για να δει τα αποτελέσματα με την ολοκλήρωση μιας campaign. Επιλέγοντας στο campaign management μια finished campaign και πατώντας το κουμπί view, μεταβαίνει στην παρακάτω φόρμα.

Campaign Viewer



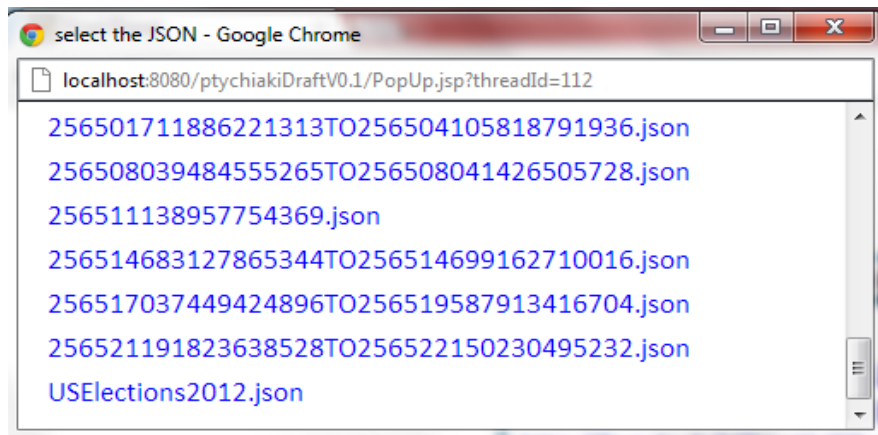
Twitter Search
Campaign Management

Debate2012 USA USA2012 iran America Romney
 JewishTimes Obama2012 Palestinians
 vicepresidentialdebate MiddleEast US Elections2012 VPdebate Biden
 Elecciones2012 israel Elections election2012 PaulRyan MittRomney
 IsraeltheRegion GOP debate Poll Democrats Obama forward
 BarackObama politics

Εικόνα 15 Campaign Viewer

Το κεντρικό παράθυρο έχει λειτουργία όμοια με αυτό στην απλή αναζήτηση στο campaign refinement και απεικονίζει τα urls, posts και expanded urls (αν υπάρχουν) που επέστρεψε η αναζήτηση. Δεξιά φαίνονται πληροφορίες για την campaign, όπως το όνομα της, τα hashtags αναζήτησης, η ημερομηνία εκκίνησης διάρκειας, η κατάσταση της, το mail ενημέρωσης του

χρηστή και αν είχε ζητηθεί επέκταση των urls. Επίσης υπάρχει ένα κουμπί export JSON που ανοίγει ένα παράθυρο το οποίο περιέχει JSON αρχεία, τόσο με τα αποτελέσματα του campaign, όσο και με τα αποτελέσματα που επέστρεψε το twitter. Τέλος, υπάρχει ένα θεματικό νέφος με όλα τα συναφή hashtags του campaign και ένα κουμπί για το καθορισμό του πόσα από αυτά θα εμφανιστούν.



Εικόνα 16 PopUp.jsp

4.3 Μοντέλο Οντοτήτων Συσχετίσεων

Η εφαρμογή μας χρησιμοποιεί μια βάση δεδομένων τόσο για την απλή αναζήτηση, όσο και για τα campaigns. Για τις απλές αναζητήσεις η βάση έχει προσωρινό χαρακτήρα και τα δεδομένα σβήνονται με κάθε νέο session. Για τα campaigns όμως ,τα δεδομένα αποθηκεύονται μόνιμα, ώστε να είναι συνεχώς προσπελάσιμα.

Τα σύνολα οντοτήτων που σχεδιάσαμε για τη βάση μας είναι τα εξής:

- **Basic.** Αναφέρεται σε ένα tweet που επιστρέφει μια απλή αναζήτηση. Έχει ως πρωτεύον κλειδί ένα id. Έχει γνωρίσματα το post που βρίσκεται στο tweet, τον χρήστη που το έκανε, τη γλώσσα του, το sessionId του χρήστη της εφαρμογής και το id του tweet όπως μας το δίνει το twitter .
- **BasicUrl.** Αναφέρεται σε ένα url που επιστρέφει μια απλή αναζήτηση. Έχει ως πρωτεύον κλειδί ένα id. Έχει γνωρίσματα το url που βρίσκεται στο tweet, το id του tweet όπως μας το δίνει το twitter , το sessionId του χρήστη της εφαρμογής και ένα ακόμα id που χρησιμοποιείται ως ξένο κλειδί και έχει το ρόλο να συνδέσει το url με το tweet στο οποίο βρίσκεται (μέσω του id του basic).
- **RelatedHash.** Αναφέρεται σε ένα σχετικό hashtag που επιστρέφει μια απλή αναζήτηση. Έχει ως πρωτεύον κλειδί ένα id. Έχει γνωρίσματα το hashtag αναζήτησης, το σχετικό hashtag που βρέθηκε, πόσες φορές εμφανίστηκε ,το sessionId του χρήστη της εφαρμογής και ένα ακόμα id που χρησιμοποιείται ως ξένο κλειδί και

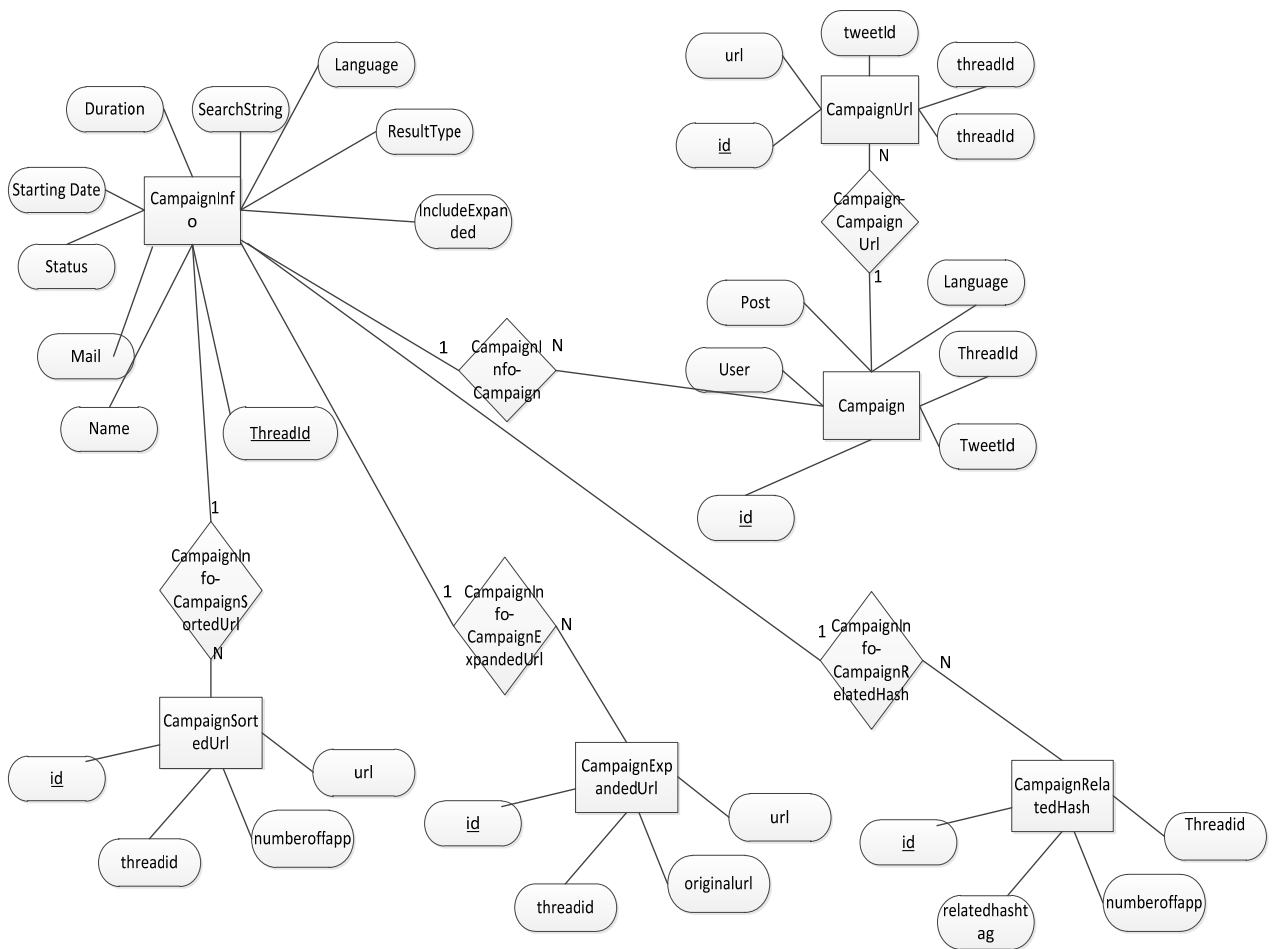
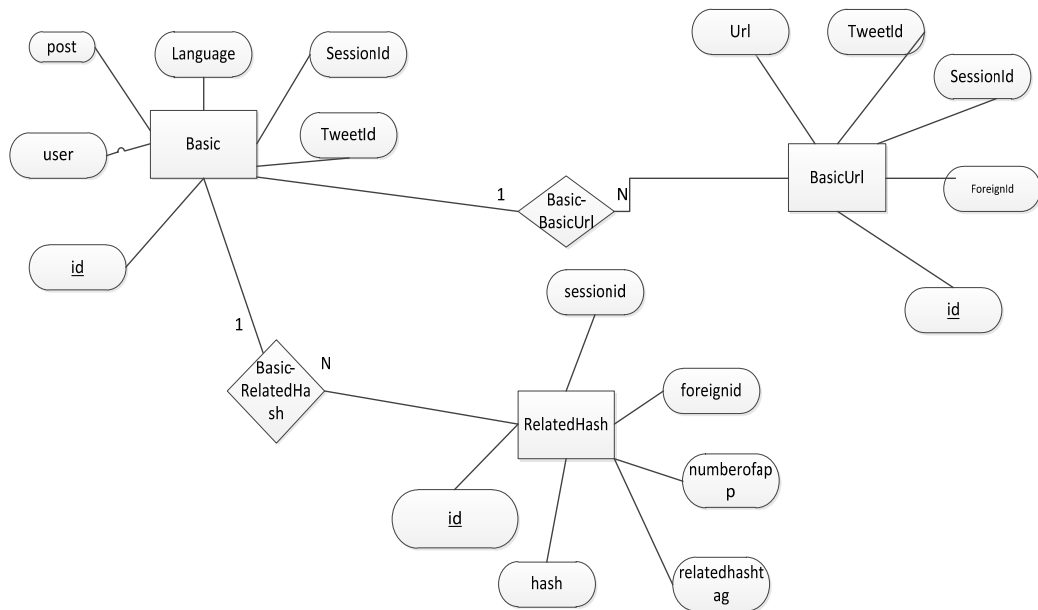
έχει το ρόλο να συνδέσει το σχετικό hashtag με το tweet στο οποίο βρίσκεται (μέσω του id του basic).

- **CampaignInfo.** Αναφέρεται σε μια campaign. Έχει ως πρωτεύον κλειδί ένα ThreadId και γνωρίσματα ένα όνομα, ένα mail, την κατάσταση της, την ημερομηνία εκκίνησης, τη διάρκεια, το κείμενο αναζήτησης, τη γλώσσα, το τύπο αποτελέσματος και μια τιμή Boolean για το αν είναι επιθυμητή η μετατροπή των urls στην επεκταμένη τους μορφή.
- **Campaign.** Αναφέρεται σε ένα tweet που ανήκει σε μια campaign. Έχει ως πρωτεύον κλειδί ένα id. Έχει γνωρίσματα το post που βρίσκεται στο tweet, τον χρήστη που το έκανε, τη γλώσσα του, το id του tweet όπως μας το δίνει το twitter και ένα ακόμα id που χρησιμοποιείται ως ξένο κλειδί και έχει το ρόλο να συνδέσει το tweet με το campaign στο οποίο βρίσκεται (μέσω του threadid του CampaignInfo).
- **CampaignUrl.** Αναφέρεται σε ένα url που βρίσκεται σε ένα tweet που ανήκει σε μια campaign. Έχει ως πρωτεύον κλειδί ένα id. Έχει γνωρίσματα το url που βρίσκεται στο tweet, το id του tweet όπως μας το δίνει το twitter, το threaded και ένα ακόμα id που χρησιμοποιείται ως ξένο κλειδί και έχει το ρόλο να συνδέσει το url με το tweet στο οποίο βρίσκεται (μέσω του id του Campaign).
- **CampaignSortedUrl.** Έχει σκοπό να κρατεί κάποιες πληροφορίες με βάση το πόσες φορές εμφανίζεται ένα url σε μια campaign. Έχει ως πρωτεύον κλειδί ένα id. Έχει γνωρίσματα το url, το πόσες φορές εμφανίζεται και ένα ακόμα threadid που χρησιμοποιείται ως ξένο κλειδί και έχει το ρόλο να συνδέσει το url με το campaign στο οποίο βρίσκεται (μέσω του threadid του CampaignInfo).
- **CampaignExpandedUrl.** Έχει σκοπό να κρατά τα urls στην επεκταμένη τους μορφή. Έχει ως πρωτεύον κλειδί ένα id. Έχει γνωρίσματα το url, το αρχικό url πριν την επέκταση και ένα ακόμα threadid που χρησιμοποιείται ως ξένο κλειδί και έχει το ρόλο να συνδέσει το url με το campaign στο οποίο βρίσκεται (μέσω του threadid του CampaignInfo).
- **CampaignRelatedHash.** Έχει σκοπό να κρατά όλα τα σχετικά hashtag που βρίσκονται και ανήκουν σε μια campaign. Έχει ως πρωτεύον κλειδί ένα id. Έχει γνωρίσματα, το σχετικό hashtag που βρέθηκε, πόσες φορές εμφανίστηκε και ένα ακόμα threadid που χρησιμοποιείται ως ξένο κλειδί και έχει το ρόλο να συνδέσει το σχετικό hashtag με την campaign στην οποία βρίσκεται (μέσω του threadid του CampaignInfo).

Τα σύνολα συσχετίσεων προκύπτουν από τα ξένα κλειδιά και είναι τα :

- **Basic-BasicUrl.**
- **Basic-RelatedHash.**
- **CampaignInfo-Campaign.**
- **Campaign-CampaignUrl.**
- **CampaignInfo-CampaignSortedUrl.**
- **CampaignInfo-CampaignExpandedUrl.**
- **CampaignInfo-CampaignRelatedHash.**

Όλα έχουν γνωρίσματα τα αντίστοιχα κλειδιά και εν' τέλει θα παραλειφθούν στο αντίστοιχο σχεσιακό διάγραμμα καθώς αποτελούν συσχετίσεις ένα προς πολλά.



Εικόνα 17 Μοντέλο οντοτήτων-συσχετίσεων

5

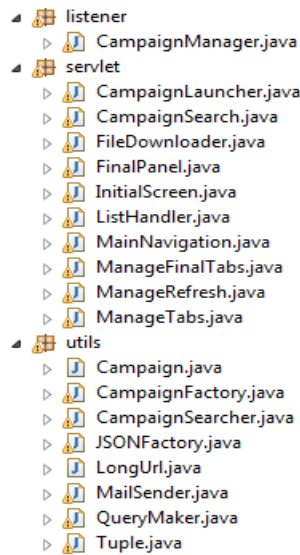
Σχεδίαση Συστήματος

Ακολουθεί η περιγραφή της σχεδίασης του συστήματος.

5.1 Αρχιτεκτονική

Η εφαρμογή μας έχει 19 κλάσεις οι οποίες χωρίζονται σε 3 πακέτα:

- Πακέτο listener που περιέχει μια κλάση, η χρήση της οποίας θα εξηγηθεί μετά.
- Πακέτο servlet που περιέχει όλα τα servlets της εφαρμογής μας.
- Πακέτο utils που περιέχει βοηθητικές κλάσεις που χρησιμοποιούνται ευρέως από τα servlets μας.



Εικόνα 18 Δομή των κλάσεων

Επίσης χρησιμοποιούνται 5 JSP σελίδες για επικοινωνία με τα servlets και εμφάνιση των αποτελεσμάτων στον browser. Αυτές είναι οι: InitialScreen.jsp, MainScreen.jsp, ListHandler.jsp, FinalPanel.jsp, PopUp.jsp.

Τέλος για τη πρόσβαση στο Twitter χρησιμοποιήθηκε η βιβλιοθήκη της Java με BSD άδεια Twitter 4j v2.2.6, για το parsing JSON αρχείων η βιβλιοθήκη Gson της Google και για τη σύνδεση με τη βάση δεδομένων η βιβλιοθήκη mysqlconnector.

5.2 Περιγραφή Κλάσεων

Ακολουθεί συνοπτική εξήγηση των κλάσεων της εφαρμογής και των μεθόδων τους:

Servlets:

5.2.1 InitialScreen.java

Servlet που επικοινωνεί με το InitialScreen.jsp, παίρνει τα hashtags αναζήτησης στην απλή αναζήτηση με τη βοήθεια κλάσεων από το πακέτο utils, λαμβάνει τα αποτελέσματα από το twitter, τα αποθηκεύει στη βάση, τα επεξεργάζεται, και στέλνει τα προς εμφάνιση αποτελέσματα στο MainScreen.jsp για να ξεκινήσει η διαδικασία του Campaign Refinement.

5.2.2 *ManageRefresh.java*

Servlet που επιτελεί την ίδια λειτουργία με το InitialScreen.java και ενεργοποιείται κατά το Campaign Refinement, όταν ο χρήστης ξαναεκτελεί αναζήτηση για καλύτερη προσαρμογή των αποτελεσμάτων.

5.2.3 *ManageTabs.java*

Servlet που είναι υπεύθυνο για τη διαχείριση των καρτελών κατά το Campaign Refinement. Παρατηρεί την εναλλαγή ανάμεσα στις 3 καρτέλες (posts,urls ,expandedurls) και κάνοντας τις απαραίτητες διεργασίες, ανανεώνει το MainScreen.jsp αναλόγως.

5.2.4 *MainNavigation.java*

Servlet που είναι υπεύθυνο για την άμεση μεταπήδηση από το Campaign Refinement, είτε στην αρχική σελίδα για απλή αναζήτηση, είτε στην σελίδα του Campaign Management ,χωρίς εκκίνηση νέας campaign.

5.2.5 *ListHandler.java*

Servlet που χειρίζεται όλα τα κουμπιά σελίδας listhandler.jsp. Είναι δηλαδή υπεύθυνο για τις λειτουργίες του Campaign Management, και για τις μεταπηδήσεις στις αντίστοιχες σελίδες.

5.2.6 *FinalPanel.java*

Servlet που χειρίζεται τη παραμετροποίηση του μεγέθους του Cloud από τον χρήστη.

5.2.7 *CampaignSearch.java*

Servlet που ελέγχει αν μια Campaign είναι ορισμένη σωστά και αν δεν είναι στέλνει τα μηνύματα λάθους, αλλιώς την εκκινεί.

5.2.8 *FileDownloader.java*

Servlet που χειρίζεται τη διαδικασία του κατεβάσματος των JSON αρχείων. Καλείται από το PopUp.jsp.

5.2.9 *ManageFinalTabs.java*

Servlet που είναι υπεύθυνο για τη διαχείριση των καρτελών κατά το Campaign Viewer. Παρατηρεί την εναλλαγή ανάμεσα στις 3 καρτέλες (posts,urls ,expandedurls) και κάνοντας τις απαραίτητες διεργασίες, ανανεώνει το FinalPanel.jsp αναλόγως.

Utils:

5.2.10 Campaign.java

Η κλάση αυτή υλοποιεί τον τύπο `campaign` ο οποίος περιέχει όλες τις βασικές πληροφορίες για μια `campaign`.

5.2.11 CampaignFactory.java

Η κλάση αυτή παρέχει τις μεθόδους για την εκκίνηση και τη διαγραφή των `Campaigns`.

5.2.12 JSONFactory.java

Η κλάση αυτή παρέχει τις μεθόδους για τη δημιουργία και τη διαγραφή των `JSON` αρχείων και των φακέλων στους οποίους αντιγράφονται/βρίσκονται.

5.2.13 LongUrl.java

Κλάση που είναι υπεύθυνη για την επικοινωνία με την υπηρεσία `Longurl` και χρησιμοποιείται για την μετατροπή των `urls` στην επεκταμένη τους μορφή.

5.2.14 MailSender.java

Κλάση που είναι υπεύθυνη για την αποστολή `mail` στο χρήστη με το πέρας της `campaign` που αυτός ξεκίνησε.

5.2.15 Tuple.java

Κλάση που υλοποιεί την τουπλά στη `java` και χρησιμοποιείται από την `QueryMaker` για τη δημιουργία του θεματικού νέφους, την ταξινόμηση των `url` κ.α.

5.2.16 QueryMaker.java

Αποτελεί μια από τις πιο σημαντικές κλάσεις της εφαρμογής. Καλείται σχεδόν από όλα τα `servlets` και είναι υπεύθυνη για την επικοινωνία με το `Twitter`, για την εγγραφή, ανάκληση, διαγραφή δεδομένων από τη βάση καθώς και για τη δημιουργία των θεματικών νεφών. Όλα τα παραπάνω γίνονται με τις εξής μεθόδους:

5.2.16.1 Destroyhashtags

Μέθοδος που σβήνει τα πολλαπλά # μπροστά από ένα hashtag και κρατά μόνο ένα.

5.2.16.2 processSearchString

Μέθοδος που παίρνει ως όρισμα ένα String και χωρίζει τα διαδοχικά hashtags με λογικό OR ώστε να γίνει δυνατή η διενέργεια ερώτησης στο Twitter , όπως μας επιτρέπει το API του.

5.2.16.3 createQuery

Μέθοδος που παίρνει ως ορίσματα τα hashtags, μια γλώσσα, και έναν τύπο, ώστε να δημιουργηθεί ένα κατάλληλο ερώτημα αναζήτησης σύμφωνα με API του Twitter .

5.2.16.4 getResult

Παίρνει ως όρισμα ένα ερώτημα που έχει δημιουργηθεί με την προηγούμενη μέθοδο, κάνει την ερώτηση στο Twitter και επιστρέφει το αποτέλεσμα.

5.2.16.5 part of

Παίρνει ως όρισμα 2 strings και τσεκάρει αν το πρώτο βρίσκεται στο δεύτερο.

5.2.16.6 writeToTemporaryDatabase

Μέθοδος με πολλαπλά ορίσματα που είναι υπεύθυνη να γράψει τα αποτελέσματα που σχετίζονται με μια απλή αναζήτηση στη βάση δεδομένων.

5.2.16.7 checkIfInDB

Βοηθητική μέθοδος που ελέγχει αν ένα hashtag υπάρχει ήδη στη βάση.

5.2.16.8 CloudMaker

Μέθοδος που φτιάχνει το θεματικό νέφος από hashtags στο Campaign Refinement.

5.2.16.9 Findmax

Βοηθητική μέθοδος που χρησιμοποιείται στην κατασκευή του νέφους.

5.2.16.10 Findmaxindex

Βοηθητική μέθοδος που χρησιμοποιείται στην κατασκευή του νέφους.

5.2.16.11 *Fontlist*

Βοηθητική μέθοδος που χρησιμοποιείται στην κατασκευή του νέφους. Καλείται από την CloudMaker και την CampaignCloudMaker για να καθοριστούν τα fonts των διαφόρων hashtags.

5.2.16.12 *CampaignCloudMaker*

Μέθοδος που φτιάχνει το θεματικό νέφος από hashtags στο Campaign Management.

5.2.16.13 *getPostsFromTemp*

Μέθοδος που επιστρέφει από τη βάση τα posts που αφορούν μια απλή αναζήτηση.

5.2.16.14 *getUrlsFromTemp*

Μέθοδος που επιστρέφει από τη βάση τα urls που αφορούν μια απλή αναζήτηση.

5.2.16.15 *Getposts*

Μέθοδος που παίρνει τα posts από ένα tweet, όπως αυτό μας επιστράφηκε έπειτα από ερώτηση στο Twitter .

5.2.16.16 *writeToCampaignDatabase*

Μέθοδος με πολλαπλά ορίσματα που είναι υπεύθυνη να γράψει τα αποτελέσματα που σχετίζονται με μια campaign στη βάση δεδομένων.

5.2.16.17 *writeToCampaignInfo*

Μέθοδος με πολλαπλά ορίσματα που εισάγει μια νέα campaign στη βάση δεδομένων. Η writeToCampaignDatabase εισάγει στη βάση τα urls, posts ,σχετικά hashtags που την αφορούν.

5.2.16.18 *updateCampaignInfo*

Μέθοδος που χρησιμοποιείται για να ανανεώσει την κατάσταση μιας campaign στη βάση.

5.2.16.19 *writeToCampaignExpandedUrl*

Μέθοδος που χρησιμοποιείται για να αποθηκεύσει σε συγκεκριμένο πίνακα της βάσης τα url που σχετίζονται με μια campaign στην επεκταμένη τους μορφή.

5.2.16.20 *writeToCampaignSortedUrl*

Μέθοδος που χρησιμοποιείται για να αποθηκεύσει σε συγκεκριμένο πίνακα της βάσης τα url που σχετίζονται με μια campaign, ταξινομημένα βάσει τον αριθμό εμφάνισης τους.

5.2.16.21 *extrackLastTweetId*

Βοηθητική μέθοδος που παίρνει το τελευταίο tweet που σχετίζεται με μια campaign.

5.2.16.22 *extrackLastTweetId*

Βοηθητική μέθοδος που παίρνει το επόμενο ThreadId που σχετίζεται με μια campaign.

5.2.16.23 *getCampaignInfo*

Μέθοδος που διαβάζει από τη βάση τις πληροφορίες για μια campaign (όνομα, διάρκεια, κατάσταση κλπ).

5.2.16.24 *deleteCampaign*

Μέθοδος που διαγράφει μια campaign από τη βάση.

5.2.16.25 *getUrlsFromCampaign*

Μέθοδος που παίρνει από τη βάση τα urls που σχετίζονται με μια campaign.

5.2.16.26 *getExpandedUrlsFromCampaign*

Μέθοδος που παίρνει από τη βάση τα urls στην επεκταμένη τους μορφή που σχετίζονται με μια campaign.

5.2.16.27 *getCampaignInfoSize*

Μέθοδος που βλέπει πόσες campaigns υπάρχουν στη βάση.

5.2.16.28 *ChangeStatus*

Αλλάζει την κατάσταση των pending μεθόδων σε Interrupted όταν ξεκινά ο server.

5.2.16.29 *isOld*

Ελέγχει αν η campaign βρίσκεται σε κάποιο thread που τρέχει.

5.2.16.30 *changeThreadId*

Αλλάζει το threadid μιας campaign που βρίσκεται στη βάση.

5.2.16.31 *getCampaignInfoByThreadId*

Παίρνει τις πληροφορίες για μια campaign, σύμφωνα με το threadid της.

5.2.16.32 *updateDate*

Αλλάζει την ημερομηνία εκκίνησης μιας campaign στη βάση.

5.2.16.33 *getPostsFromCampaign*

Παίρνει από τη βάση τα posts που σχετίζονται με μια campaign.

5.2.16.34 *getJSONInfoFromCampaign*

Παίρνει τις πληροφορίες που σχετίζονται με μια campaign και τις επιστρέφει σε μορφή JSON.

5.2.16.35 *getJSONExpandedUrl*

Παίρνει τα urls που σχετίζονται με μια campaign στην επεκταμένη τους μορφή και τα επιστρέφει σε μορφή JSON.

5.2.16.36 *getExpandedUrls*

Επιστρέφει τα urls που σχετίζονται με μια campaign στην επεκταμένη τους μορφή.

5.2.16.37 *checkIfNameTaken*

Ελέγχει αν το όνομα που πάει να δώσει ο χρήστης σε μια campaign είναι μοναδικό.

Listener

5.2.17 *CampaignManager.java*

Κλάση που είναι υπεύθυνη για το multithreading της εφαρμογής, δηλαδή για τη δυνατότητα να τρέχουν παράλληλα πολλές campaign στον server. Η λειτουργία του θα εξηγηθεί αναλυτικά αργότερα.

5.3 *Βάση Δεδομένων*

Εδώ δίνεται το σχεσιακό σχήμα της βάσης δεδομένων μας, όπως αυτό προκύπτει από το μοντέλο οντοτήτων-συσχετίσεων που δόθηκε παραπάνω. Να σημειωθεί ότι οι συσχετίσεις έχουν παραληφθεί καθώς αποτελούν συσχετίσεις ένα προς πολλά. Θα γίνονται εμφανείς μέσω ξένων κλειδιών.

Οι σχέσεις της βάσης είναι:

Basic(id,user,post,language,SessionId,TweetId)

Basicurl(id,url,TweetId,SessionId,ForeignId)

Campaign(id,user,post,language,ThreadId,TweetId)

campaignExpandedUrl(id,url,ThreadId,originalUrl,numberOfapp)

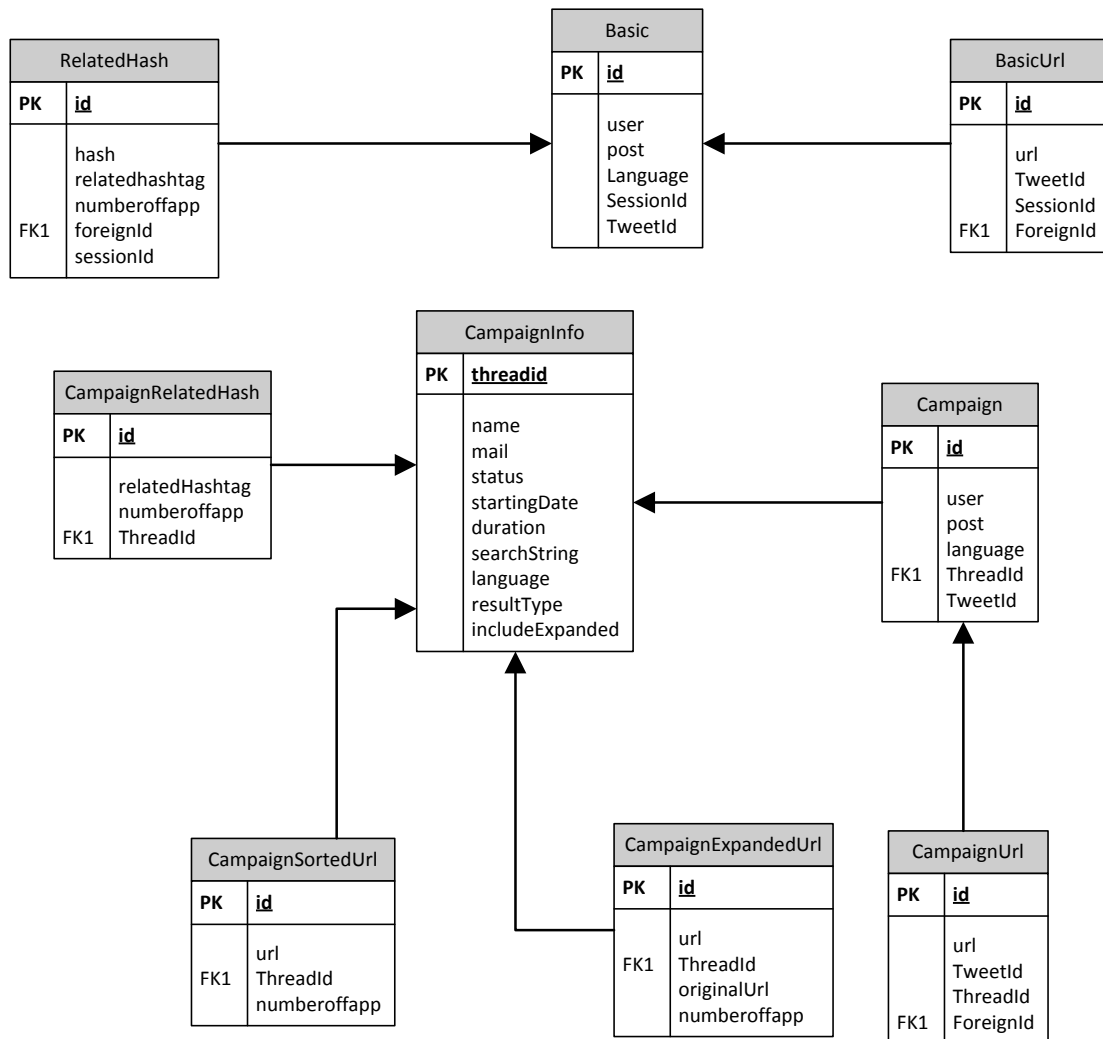
campaignInfo(threadId,name,mail,status,startingDate,duration,searchString,language,resultType,includeExpanded)

campaignRelatedHash(id,relatedHashtag,numberOfapp,ThreadId)

campaignSortedUrl(id,url,ThreadId,numberOfapp)

campaignurl(id,url,TweetId.ThreadId,ForeignId)

relatedhash(id,hash,relatedhashtag,numberOffapp,foreignId,sessionId)



Εικόνα 19 Σχεσιακό σχήμα

5.4 Κωδικοποίηση αρχείων

Η εφαρμογή μας επιστρέφει 2 τύπους JSON αρχείων. Ο ένας περιλαμβάνει τα Tweets της αναζήτησης όπως ακριβώς τα δέχεται η εφαρμογή από το Twitter . Το όνομα αυτού του αρχείου έχει τη μορφή FirstTweetIdTOLastTweetId.json. ή αν περιλαμβάνεται μόνο ένα Tweet, TweetId.json. Η μορφή του αρχείου είναι:

```
{
  "results": [TweetInformation *]
}
```

Στο κομμάτι TweetInformation υπάρχει η δομή του Tweet όπως επιστρέφεται από το Twitter

:

```
{  
  "text": ,  
  "from_user_id": ,  
  "from_user_name": ,  
  "geo": ,  
  "profile_image_url_https": ,  
  "iso_language_code": ,  
  "to_user_name": ,  
  "entities":  
    [{  
      "expanded_url": ,  
      "indices": ,  
      "display_url": ,  
      "url":  
    }],  
  "hashtags":  
    [{  
      "text": ,  
      "indices":  
    }],  
  "user_mentions": ,  
  "id": ,  
  "to_user_id_str": ,  
  "source": ,  
  "from_user_id_str": ,  
  "from_user": ,  
  "created_at": ,  
  "to_user": ,  
  "id_str": ,  
  "profile_image_url": ,
```

```
“metadata”:  
  {  
    “result_type”:  
  }  
}
```

Ο άλλος τύπος JSON αρχείου που επιστρέφει η εφαρμογή μας όταν τελειώσει μια Campaign είναι ένα αρχείο με όνομα της μορφής CampaignName.json. Η μορφή του αρχείου είναι η ακόλουθη:

```
{  
  “Name”: ,  
  “Mail”: ,  
  “Status”: ,  
  “StartingDate”: ,  
  “Duration”: ,  
  “SearchString”: ,  
  “Language”: ,  
  “ResultType”: ,  
  “Tweets”: [CampaignTweets*],  
  “UrlExpansion”:  
  [{  
    “url”: ,  
    “extendedurl”: ,  
  }*],  
  “RelatedHashtags”:  
  [{  
    “hashtag”:  
  }*]  
}
```

Η μορφή που έχει το κομμάτι CampaignTweets είναι το ακόλουθο:

```
{  
  "User": ,  
  "TweetId": ,  
  "Post": ,  
  "Language": ,  
  "URL":  
}
```


6

Υλοποίηση

Σε αυτό το κεφάλαιο θα αναφέρουμε κάποιες λεπτομέρειες υλοποίησης που είναι αξιοπρόσεκτες καθώς και κάποιες οδηγίες εγκατάστασης των πλατφόρμων και των προγραμματιστικών εργαλείων που χρησιμοποιήσαμε.

6.1 Λεπτομέρειες υλοποίησης

Όσον αφορά την υλοποίηση της εφαρμογής μας υπάρχουν τρεις λεπτομέρειες που θα πρέπει να εξηγηθούν ξεχωριστά γιατί ο τρόπος υλοποίησης τους είναι κάπως πολύπλοκος ή όχι προφανής.

6.1.1 *Multi-threading Handling*

Η εφαρμογή μας είναι φτιαγμένη έτσι ώστε να υποστηρίζει τη ταυτόχρονη λειτουργία περισσότερων από μιας Campaigns.

Για να επιτευχθεί αυτό αρχικά χρησιμοποιούμε έναν listener (την κλάση CampaignManager), ο οποίος τρέχει μαζί με την εκκίνηση του Server. Εκεί πέρα από κάποιες άλλες αρχικοποιήσεις δημιουργούμε έναν κενό Vector τύπου Timer, όπου Timer είναι μια κλάση της Java στο πακέτο java.util η οποία προσφέρει μεθόδους για τη μελλοντική εκτέλεση ενεργειών μέσα σε κάποιο Thread [TMR].

Αφού δημιουργηθεί ο Vector χρησιμοποιούμε τη μέθοδο findCampaignInfoSize() της κλάσης QueryMaker προκειμένου να μάθουμε πόσα campaigns υπάρχουν ήδη στη βάση. Στη συνέχεια αποθηκεύουμε τις δύο αυτές δομές στον Servlet Container.

```
this.sc= arg0.getServletContext();
Vector<Timer> activeCampaigns = null;
activeCampaigns = new Vector<Timer>();

QueryMaker qm = new QueryMaker();

int existingThreads =qm.findCampaignInfoSize();

sc.setAttribute("campaigns", activeCampaigns);
sc.setAttribute("existingThreads", existingThreads);
```

Κώδικας για τη δημιουργία Vector που περιλαμβάνεται στην CampaignManager.Java

Εφόσον αυτά τα στοιχεία υπάρχουν στον container, όταν θέλουμε να εκκινήσουμε μια Campaign, αρχικά ανακτούμε το attribute «campaigns», δημιουργούμε έναν καινούριο Timer χρησιμοποιώντας τη μέθοδο createTimer() της κλάσης Campaign Factory και στη συνέχεια τοποθετούμε τον καινούριο Vector στον Container.

```
Vector<Timer> activeCampaigns =
    (Vector<Timer>) getServletContext().getAttribute("campaigns");
CampaignFactory cf = new CampaignFactory();
timer =
    cf.createTimer(hours,days,threadID,searchString, language, resultType,
        mail,name,includeExpanded);
activeCampaigns.add(timer);
ServletContext sc = request.getServletContext();
sc.setAttribute ("campaigns", activeCampaigns);
```

Κώδικας για τη εκκίνηση μιας Campaign μέσα από την CampaignSearch.java

Στη περίπτωση που ο χρήστης επιθυμεί να σταματήσει μια campaign πρόωρα, ανακτώνται από τον Container τα attributes «campaigns» και «existingThreads», βρίσκουμε μέσω της πληροφορίας «existingThreads» το index της campaign μας στον Vector «campaigns», μέσω της μεθόδου findIndexOfTimer της κλάσης CampaignFactory, σταματάμε τον Timer και τον αφαιρούμε από τον Vector, στη συνέχεια επιτελούμε τις λειτουργίες που πρέπει να εκτελεστούν όταν τερματιστεί μια Campaign (π.χ. αποστολή Mail) και ύστερα προσθέτουμε στον Container τον ανανεωμένο Vector.

```

int indexInTimer =
    cf.findIndexOfTimer(campaignVector, threadId, numberOfOlds);

Vector<Timer> activeCampaigns =

    (Vector<Timer>) getServletContext().getAttribute("campaigns");

activeCampaigns.elementAt(indexInTimer).cancel();

activeCampaigns.elementAt(indexInTimer).purge();

ServletContext sc = request.getServletContext();

sc.setAttribute ("campaigns", activeCampaigns);

```

Κώδικας για την πρόωρη διακοπή μιας Campaign (παραλείπονται ενέργειες όπως η αποστολή mail)

Αν ο χρήστης επιθυμεί να ανανεώσει μια τρέχουσα campaign ή να επανεκκινήσει μια Campaign που έχει διακοπεί αυτόματα (έχει δηλαδή status Interrupted) χρησιμοποιείται σχεδόν πανομοιότυπη διαδικασία.

Αν η Campaign έχει διακοπεί ζητάμε ένα καινούριο ThreadId, ξεκινάμε μια καινούρια Campaign με τα ίδια στοιχεία που είχε η παλιά (εκτός από τον καινούριο χρόνο τερματισμού), τη προσθέτουμε στον Vector και στη συνέχεια ανανεώνουμε τη βάση με τα καινούρια μας δεδομένα καθώς και τους φακέλους των αρχείων JSON. Επίσης σε αυτή τη περίπτωση πρέπει να αλλάξει το attribute «existingThreads» του Container γιατί πλέον οι παλιές Campaigns έχουν μειωθεί κατά μια.

```

long maxThreadId = qm.getNextThreadId();
qm.changeThreadId(threadId, maxThreadId);

Vector<Timer> activeCampaigns =

    (Vector<Timer>) getServletContext().getAttribute("campaigns");
Campaign campaign = qm.getCampaignInfoByThreadId(maxThreadId);
JSONFactory jf = new JSONFactory();
jf.renameFolder(threadId, maxThreadId);
qm.deleteCampaign(threadId);
Timer timer =
    cf.createTimer(hours,days,(int) maxThreadId,
        campaign.getSearchString(),
        campaign.getMail(),campaign.getName(),campaign.getLanguage(),
        campaign.getResultType(),campaign.isExpandedUrls());
activeCampaigns.add(timer);
qm.updateDate(campaign.getDate(), maxThreadId);
ServletContext sc = request.getServletContext();
sc.setAttribute ("campaigns", activeCampaigns);
campaignVector = qm.getCampaignInfo();
int vectorLength = campaignVector.size();
numberOfOlds--;
sc.setAttribute ("existingThreads", numberOfOlds);

```

Κώδικας για την επανεκκίνηση Interrupted Campaign.

Αν η Campaign έχει status Pending, η διαδικασία είναι η ίδια με μόνη διαφορά το ότι ο το attribute «existingThreads» δεν ανανεώνεται, οπότε ο κώδικας παραλείπεται.

Τέλος αν κάποιος επιλέξει να διαγράψει μια Campaign, πάλι θα πρέπει να ελέγξουμε αν αυτή έχει status Pending, Finished ή Interrupted.

Αν είναι Finished ή Interrupted, δηλαδή δεν υπάρχει στον Vector «campaigns», αυτό σημαίνει ότι θα πρέπει να αλλάξει το attribute «existingThreads». Αν το status είναι Pending θα πρέπει να σταματήσουμε τον Timer που χειρίζεται τη Campaign και να τον αφαιρέσουμε από τον Vector. Σε κάθε περίπτωση πάντως θα πρέπει να διαγράψουμε τις πληροφορίες από τη βάση δεδομένων καθώς και τους φακέλους που έχουν δημιουργηθεί.

```
Vector<Campaign> campaignVector = qm.getCampaignInfo();

int numberOfOlds =
    (int) getServletContext().getAttribute("existingThreads");
boolean isOld = qm.isOld(numberOfOlds, threadId);
ServletContext sc = request.getServletContext();

//delete the folder with the JSON's
JSONFactory jf = new JSONFactory();
jf.deleteFolder(String.valueOf(threadId));
if (isOld) {
    qm.deleteCampaign(threadId);
    numberOfOlds--;
    sc.setAttribute ("existingThreads", numberOfOlds);
} else {
    int indexInTimer = cf.findIndexOfTimer(campaignVector, threadId,
        numberOfOlds);
    Vector<Timer> activeCampaigns =
        (Vector<Timer>) getServletContext().getAttribute("campaigns");
    String status =
        campaignVector.elementAt(indexInTimer+numberOfOlds).getStatus();
    if (status.equals("Pending")) {

        activeCampaigns.elementAt(indexInTimer).cancel();

        activeCampaigns.elementAt(indexInTimer).purge();

    }
    activeCampaigns.remove(indexInTimer);
    sc.setAttribute ("campaigns", activeCampaigns);
    qm.deleteCampaign(threadId);
}
```

Κώδικας για τη διαγραφή των Campaigns

Αυτές ήταν οι βασικές λειτουργίες του Multi-Threading handling. Επίσης αξίζει να σημειωθεί ότι ο listener πρέπει να δηλωθεί και στο αρχείο web.xml για να λειτουργήσει.

6.1.2 Δυναμικό περιεχόμενο στο Campaign Management

Όπως έχει ειπωθεί και πιο πριν, στη λειτουργία Campaign Management κάποιος χρήστης μπορεί να δει όλες τις Campaign, να επιλέξει κάποια από αυτές και να τη χειριστεί όπως επιθυμεί. Για να επιλέξει την Campaign που θέλει ο χρήστης, θα μπορούσε να υπάρχει ένα HTML <select> Tag. Όμως ακόμα και αν με τη χρήση CSS θα μπορούσαμε να το προσαρμόσουμε σε μεγάλο βαθμό υπάρχει ένας σοβαρός περιορισμός. Τα πεδία <option> στο <select> Tag μπορούν να περιέχουν μόνο απλό κείμενο και όχι HTML. Έτσι, επειδή εμείς θέλαμε για λόγους ευκολίας χειρισμού αλλά και εμφάνισης να μορφοποιήσουμε το περιεχόμενο, δημιουργήσαμε με χρήση CSS και JavaScript μια δική μας δομή εμφάνισης.

Στο ListHandler.jsp δημιουργούμε ένα <div> Tag με id = "content". Μέσα σε αυτό, με ένα scriptlet JSP τυπώνουμε όλες τις campaigns τις οποίες έχουμε ανακτήσει από το attribute της request «Vector» και το έχουμε περάσει σε έναν Vector<Campaign>, με <div> tag element με id κάθε φορά το ThreadId της εκάστοτε Campaign.

```
<div id="content">
<%
ListIterator iter = campaignVector.listIterator();
while (iter.hasNext()) {
    Campaign campaign = (Campaign) iter.next();
    out.print("<div id = \""+campaign.getThreadId() + "\" class = \"elem\">
");
    out.print("<font color = \"BLUE\"> Campaign Name: </font> " +
campaign.getName()+ "<br /> " +
"<font color = \"BLUE\"> Search Quote: </font> " +
campaign.getSearchString() +
" <font color = \"BLUE\"> Start Date: </font> " + campaign.getDate() +
" <font color = \"BLUE\"> Duration: </font> " + campaign.getDuration() +
"<br /> <font color = \"BLUE\"> Mail: </font> " + campaign.getMail() +
" <font color = \"BLUE\"> UrlExpansion: </font> " +
campaign.isExpandedUrls() +
" <font color = \"BLUE\"> Status: </font> " + campaign.getStatus() +
"</div> " + "<hr style=\"color:#006699; background-color:#006699;\" />\"
);
}
%>
</div>
```

Scriptlet που τυπώνει τις Campaigns με την κατάλληλη μορφοποίηση

Εφόσον τώρα όλες οι Campaign έχουν τυπωθεί με τον επιθυμητό τρόπο μπορεί να γίνει και ο επιθυμητός χειρισμός από μια σειρά συναρτήσεων σε JavaScript χρησιμοποιώντας τη βιβλιοθήκη jquery-1.8.0.js. Όταν το έγγραφο έχει φορτώσει πλήρως (δηλαδή έχουν τυπωθεί και οι διαθέσιμες campaigns), θέτονται όλα τα κουμπιά χειρισμού των Campaigns ανενεργά. Αυτό γίνεται πολύ εύκολα χρησιμοποιώντας την JQuery:

```

$("#viewButton").attr("disabled", "disabled");
$("#deleteButton").attr("disabled", "disabled");
$("#info").attr("disabled", "disabled");
$("#updateButton").attr("disabled", "disabled");
$("#relaunchButton").attr("disabled", "disabled");
$("#stopButton").attr("disabled", "disabled");
$("#JSONButton").attr("disabled", "disabled");
$("#mainSearchButton").show();
$("#initialScreenButton").show();

```

Κώδικας για την απενεργοποίηση των κουμπιών

Στη συνέχεια έχουμε δημιουργήσει μια συνάρτηση `selectHighlight(element,number,status)`. Η συγκεκριμένη συνάρτηση όταν γίνεται click στο στοιχείο `element` που δίνεται ως παράμετρος, όλα τα `<div>` Tags μέσα στο `content`, αποκτούν άσπρο χρώμα για το φόντο τους ενώ το στοιχείο μέσα στο οποίο έγινε click αποκτά κάποιο άλλο χρώμα για να ξεχωρίζει. Παράλληλα αφαιρούνται από τη φόρμα που καλεί το servlet `ListHandler.java` δύο `<input>` tags, τα οποία είναι ορισμένα να μη φαίνονται και έχουν `id hidden1` και `hidden2` αντίστοιχα. Αυτά περιελάμβαναν τις τιμές `ThreadId` και `Status` κάποιας `Campaign` που είχε επιλεγθεί παλιότερα (ή είναι κενά αν είναι η πρώτη φορά που επιλέγεται `Campaign`). Επομένως στη συνέχεια ανατοποθετούνται τα δυο αυτά `<input>` Tags με τις καινούριες τους τιμές. Τέλος ελέγχεται το `status` αν είναι 0 (`Pending`), 1 (`Interrupted`), 2 (`Finished`) και ενεργοποιούνται/απενεργοποιούνται τα αντίστοιχα κουμπιά. Στη συνέχεια υλοποιούμε τη συνάρτηση `selectHighlightAll(max,array,status)`, που δέχεται σαν ορίσματα ένα πίνακα με τα `ThreadId's` με τη σειρά που εκτυπώνονται, ένα πίνακα με το `status` της κάθε `Campaign` και το μέγεθος των 2 πινάκων (που προφανώς είναι ίσο με τον αριθμό των `Campaigns`), και εφαρμόζεται η `selectHighlightAll`, μέσα σε αυτή. Τέλος, δημιουργούμε με ένα scriptlet JSP αυτές τις παραμέτρους και καλούμε τη συνάρτηση με αυτά τα ορίσματα. Οι πίνακες δημιουργούνται σε `Strings` από το `Scriptlet` και στην ουσία μέσα από αυτό γράφουμε κώδικα σε `JavaScript` έτσι ώστε να τυπωθεί στην `HTML` σελίδα που θα παραχθεί από το `JSP`.

Ακολουθεί ο κώδικας για τη παρακάτω διαδικασία, ο οποίος σε συνδυασμό με τα προηγούμενα κομμάτια που υλοποιήθηκαν δημιουργεί μια εναλλακτική λύση που αντικαθιστά στη συγκεκριμένη περίπτωση το `<select>` Tag της `HTML` προσφέροντας πολύ περισσότερες δυνατότητες προσαρμογής. Αν και η λύση που βρήκαμε είναι στενά συνδεδεμένη με την εφαρμογή μας, θα μπορούσε να γενικευτεί με λίγες αλλαγές έτσι ώστε να είναι μια γενικότερη εναλλακτική λύση που να μπορεί να εφαρμοστεί και σε περισσότερες περιπτώσεις.

```

element.click(function(){
    $(".elem").css("background","none");
    element.css("background-color","#D6D6FF");
    $("#hidden1").remove();
    $("#hidden2").remove();
    $("<input type =\"hidden\" id=\"hidden1\"
name=\"threadId\" value=\"\" + number + \"\" />").appendTo("#buttons");
    $("<input type =\"hidden\" id=\"hidden2\"
name=\"status\" value=\"\" + status + \"\" />").appendTo("#buttons");

    if (status == 0) {
        $("#viewButton").attr("disabled","disabled");
        $("#deleteButton").removeAttr("disabled");
        $("#info").removeAttr("disabled");
        $("#updateButton").removeAttr("disabled");
        $("#relaunchButton").attr("disabled","disabled");
        $("#stopButton").removeAttr("disabled");
        $("#mainSearchButton").removeAttr("disabled");
        $("#JSONButton").attr("disabled","disabled");
        $("#initialScreenButton").removeAttr("disabled");
    }

    if (status == 1) {
        $("#viewButton").attr("disabled","disabled");
        $("#deleteButton").removeAttr("disabled");
        $("#info").removeAttr("disabled");
        $("#updateButton").attr("disabled","disabled");
        $("#relaunchButton").removeAttr("disabled");
        $("#stopButton").attr("disabled","disabled");
        $("#JSONButton").removeAttr("disabled");
        $("#mainSearchButton").removeAttr("disabled");
        $("#initialScreenButton").removeAttr("disabled");
    }

    if (status == 2) {
        $("#viewButton").removeAttr("disabled");
        $("#deleteButton").removeAttr("disabled");
        $("#info").attr("disabled","disabled");
        $("#updateButton").attr("disabled","disabled");
        $("#relaunchButton").attr("disabled","disabled");
        $("#stopButton").attr("disabled","disabled");
        $("#JSONButton").removeAttr("disabled");
        $("#mainSearchButton").removeAttr("disabled");
        $("#initialScreenButton").removeAttr("disabled");
    }

    }
});

var selectHighlightAll_ = function (max,array,status) {
    for (var i = 0;i<max;i++) {

        var id = "#" +array[i];
        var statusCond_ = status[i];

        selectHighlight($(id),array[i],status[i]);};};

```

```

<%
    int length =( Integer)
request.getAttribute("length").intValue();

    String arrayJS = "[";
    String statusJS = "[";

    while (iterJs.hasNext()) {
        Campaign campaign = (Campaign) iterJs.next();
        String status = campaign.getStatus();
        int stat = -1;
        if (status.equals("Pending"))
            stat = 0;
        else if (status.equals("Interrupted"))
            stat = 1;
        else if (status.equals("Finished"))
            stat = 2;
        arrayJS = arrayJS + campaign.getThreadId() + ",";
        statusJS = statusJS + stat + ",";
    }

    if (arrayJS.length() == 1) {
        arrayJS = arrayJS + "]";
        statusJS = statusJS + "]";
    }
    else {
        arrayJS= arrayJS.substring(0, arrayJS.length() -1 ) +
"]";
        statusJS= statusJS.substring(0, statusJS.length() -1 )
+ "]";
    }

    out.print("selectHighlightAll(" + length + "," + arrayJS +
"," + statusJS + ");");
%>

```

Κώδικας για τη δημιουργία του δυναμικού περιεχομένου στη λειτουργία Campaign Management

6.1.3 Υλοποίηση του θεματικού νέφους.

Για την υλοποίηση του θεματικού νέφους, εντοπίζονται πρώτα όλα τα σχετικά hashtags που βρίσκονται στα tweets που επιστράφηκαν από κάποια αναζήτηση, είτε αυτή είναι απλή αναζήτηση είτε campaign. Έπειτα αποθηκεύονται στη βάση δεδομένων σε ξεχωριστό πίνακα, μαζί με τον αριθμό εμφανίσεων τους. Τέλος επιλέγονται τα πιο δημοφιλή (τα 20 δημοφιλέστερα για απλή αναζήτηση στο Campaign Refinement και όσα επιλέξει ο χρήστης για το campaign) και εκχωρούνται σε αυτά μέγεθος γραμματοσειράς και χρώμα. Η παραπάνω διαδικασία γίνεται με τον εξής αλγόριθμο:

- Δημιουργούμε πέντε κατηγορίες, μια για κάθε ξεχωριστό font που θα χρησιμοποιηθεί.
- Στην πρώτη κατηγορία θα ενταχτεί το 10% των hashtags, στη δεύτερη το 20%,στην τρίτη το 40%,στη τέταρτη το 10% και στην πέμπτη το 10%.Αυτο γίνεται με τη λογική ότι η μεσαία κατηγορία θα έχει και τα πιο πολλά στοιχεία.

```
int noKeys3 = (int) (cloudlistsize * perCent3);
int noKeys4 = (int) (cloudlistsize * perCent4);
int noKeys2 = (int) (cloudlistsize * perCent2);
int noKeys5 = (int) (cloudlistsize * perCent5);
int noKeys1 = (int) (cloudlistsize * perCent1);
```

Κώδικας για την κατηγοριοποίηση των hashtags

- Αν αριθμός που προκύπτει από την πρόσθεση των παραπάνω δεν είναι ίδιος με τον αριθμό των προς εμφάνιση hashtags, αυξάνουμε την μεσαία κατηγορία ως προς τη διαφορά τους,

```
int rest = cloudlistsize - (noKeys1 + noKeys2 + noKeys3 + noKeys4 +
noKeys5);
noKeys3=noKeys3+rest;
```

Κώδικας για την αύξηση της μεσαίας κατηγορίας

- Έχοντας ταξινομήσει τα hashtags ως προς τη σειρά εμφάνισης τους, εκχωρούμε fonts. Ξεκινώντας από τα πιο δημοφιλή, δίνουμε στα noKeys1 πρώτα το μεγαλύτερο font, στα επόμενα noKeys2 το επόμενο κλπ.

```
for(int z=0;z<noKeys1;z++){
    myfontlist.add("tag1");
}
for(int z=noKeys1;z<noKeys2+noKeys1;z++){
    myfontlist.add("tag2");
}
for(int z=noKeys2+noKeys1;z<noKeys3+noKeys2+noKeys1;z++){
    myfontlist.add("tag3");
}
for(int z=noKeys3+
noKeys2+noKeys1;z<noKeys4+noKeys3+noKeys2+noKeys1;z++){
    myfontlist.add("tag4");
}
for(intz=noKeys4+noKeys3+noKeys2+noKeys1;z<noKeys5+noKeys4+noKeys3+n
oKeys2+noKeys1;z++){
    myfontlist.add("tag5");
}
```

Κώδικας για τη προσθήκη των fonts

6.2 Πλατφόρμες και προγραμματιστικά εργαλεία

Σε αυτή την ενότητα θα περιγράψουμε τη πλατφόρμα ανάπτυξης και εκτέλεσης της εφαρμογής μας καθώς και τα προγραμματιστικά εργαλεία που χρησιμοποιήσαμε.

6.2.1 Πλατφόρμες ανάπτυξης

Οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής μας είναι όσον αφορά το client side computing οι γλώσσες HTML/CSS και JavaScript στην οποία χρησιμοποιήθηκε η βιβλιοθήκη JQuery και όσον αφορά το server side computing η γλώσσα Java με τις μεθόδους των Servlets και των JSP, ο server Tomcat, και για τη σχεσιακή βάση η MySql.

6.2.1.1 Web Design Tools

Τα πρότυπα και οι γλώσσες που χρησιμοποιήθηκαν για το σχεδιασμό της γραφικής διεπαφής με το χρήστη είναι τα ακόλουθα:

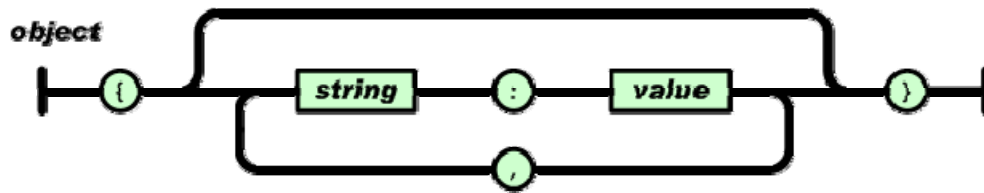
- **HTML.** Η HTML είναι μια γλώσσα που χρησιμοποιείται για την περιγραφή της δομής των ιστοσελίδων. Προσφέρει τη δυνατότητα στους σχεδιαστές να δημοσιεύσουν έγγραφα με επικεφαλίδες, κείμενο, πίνακες λίστες κλπ, αλλά και τη δυνατότητα να αποκτήσουν online πληροφορίες μέσω συνδέσμων υπερκειμένου. Επίσης παρέχει τα μέσα για να σχεδιάσει κανείς φόρμες για επικοινωνία με απομακρυσμένες υπηρεσίες σε διάφορους τομείς . Τέλος προσφέρει τη δυνατότητα να περιλαμβάνει κάποιος στο διαδικτυακό του έγγραφο και τύπους πολυμέσων όπως video και ήχο [HTCS].

Αυτή η περιγραφή της δομής των σελίδων γίνεται με τη χρήση σήμανσης (markup), δηλαδή με υποσημειώσεις σε ένα έγγραφο που είναι συντακτικά διαχωρίσιμες από το υπόλοιπο κείμενο [MRKP].

- **CSS.** Η CSS είναι η γλώσσα που χρησιμοποιείται για τη περιγραφή της παρουσίασης των ιστοσελίδων. Δηλαδή επεμβαίνει στο χρώμα , το σχέδιο και τις γραμματοσειρές τις. Προσφέρει τη δυνατότητα να προσαρμοστεί το παρουσιαστικό μιας ιστοσελίδας στους διαφορετικούς τύπους συσκευών, όπως για παράδειγμα μεγάλες ή μικρές οθόνες και εκτυπωτές. Η CSS είναι ανεξάρτητη της HTML και μπορεί να χρησιμοποιηθεί με κάθε γλώσσα σήμανσης (markup language) που βασίζεται σε XML. Ο διαχωρισμός HTML και CSS έχει ως συνέπεια να διευκολύνεται η συντήρηση ιστοσελίδων και να είναι προσιτό το ίδιο περιεχόμενο σε διαφορετικά

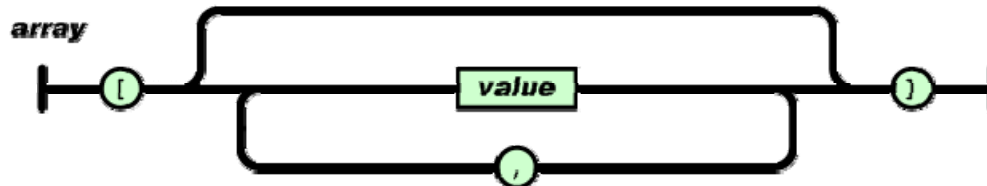
περιβάλλοντα. Αυτό σαν ιδέα αναφέρεται σαν διαχωρισμός δομής από παρουσίαση [HTCS].

- **XML.** Η XML είναι μια γλώσσα σήμανσης που καθορίζει ένα σύνολο κανόνων για τη κωδικοποίηση εγγράφων με ένα τρόπο ώστε να είναι εύκολα αναγνωρίσιμο και από άνθρωπο και από μηχανή. Καθορίζεται από το XML 1.0 Specification που δημιουργήθηκε από το W3C καθώς και από διάφορους άλλους σχετικούς προσδιορισμούς. Είναι σχεδιασμένο έτσι ώστε να είναι απλό, γενικό και χρηστικό μέσα στο διαδίκτυο. Επίσης έχει σα σκοπό την υποστήριξη μιας ευρείας γκάμας από εφαρμογές, και την ευκολία συγγραφής κώδικα σε αυτό, ενώ δεν έχει δοθεί σημασία στο να είναι λακωνικό. Επίσης αξίζει να σημειωθεί ότι υποστηρίζει όλες τις γλώσσες μέσω του προτύπου Unicode. Πάρα πολλές προγραμματιστικές διεπαφές εφαρμογής (APIs) έχουν σχεδιαστεί έτσι ώστε οι προγραμματιστές να πρέπει να επεξεργαστούν XML δεδομένα. Σε αυτό έχει παίξει ρόλο ότι ένα αρχείο XML είναι αυστηρά ορισμένο με ένα σύνολο κανόνων. Τέλος, από το XML πηγάζουν πολλές δημοφιλείς γλώσσες, όπως οι RSS, Atom, SOAP, XHTML [XML].
- **JSON.** Επειδή όμως η XML έχει δεχθεί ιδιαίτερη κριτική για το βερμπαλισμό της και επίσης το βασικό δένδρικό μοντέλο της XML σε συστήματα τύπων γλωσσών προγραμματισμού ή βάσεων δεδομένων μπορεί να είναι ιδιαίτερα δύσκολο έχουν προκύψει και γλώσσες σήμανσης με διαφορετική φιλοσοφία για να καλύψουν αυτά τα προβλήματα. Μια από αυτές, η οποία υποστηρίζεται και από το API του Twitter είναι η JSON. Η JSON είναι ένα ελαφρύ πρότυπο με κύριο σκοπό να μπορεί να διαβαστεί και να γραφεί εύκολα από ανθρώπους, αλλά και να αναλύεται και να δημιουργείται γρήγορα και από μηχανές. Βασίζεται σε ένα υποσύνολο του JavaScript Programming Language, Standard ECMA-262 3rd Edition – December 1999. Χρησιμοποιεί συμβάσεις των γλωσσών προγραμματισμού που μοιάζουν στη C για να είναι εύκολη η εξοικείωση με αυτή. Η JSON χτίζεται πάνω σε 2 δομές: μια συλλογή από ζευγάρια ονόματος/τιμής στα πρότυπα ενός αντικειμένου ή εγγραφής σε άλλες γλώσσες και μία διατεταγμένη λίστα Τιμών στα πρότυπα της δομής του πίνακα ή της λίστας. Αυτές είναι καθολικές δομές δεδομένων που με κάποιο τρόπο υποστηρίζονται σε όλες τις σύγχρονες γλώσσες προγραμματισμού. Στη JSON παίρνουν την ακόλουθη μορφή: Ένα αντικείμενο είναι ένα μη ταξινομημένο σύνολο ζευγαριών ονόματος/τιμής που ξεκινάει με το σύμβολο «{» και τελειώνει με το σύμβολο «}». Κάθε όνομα ακολουθείται από «:» και τη τιμή του αντικειμένου και κάθε ζευγάρι διαχωρίζεται από τα υπόλοιπα με «,» [JSON].



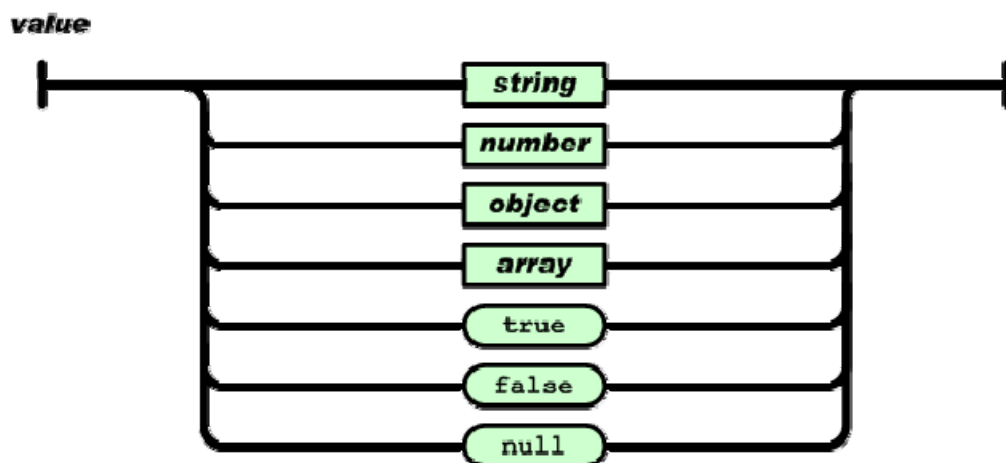
Εικόνα 20 Περιγραφή ενός αντικειμένου

Ένας πίνακας ξεκινάει με το σύμβολο «[» και ολοκληρώνεται με το σύμβολο «]». Ενδιάμεσα οι τιμές διαχωρίζονται με «,» [JSON].



Εικόνα 21 Περιγραφή ενός πίνακα

Μία τιμή μπορεί να είναι μία συμβολοακολουθία ανάμεσα σε «"» ή ένας αριθμός ή true ή false ή null ή αντικείμενο ή πίνακας [JSON].



Εικόνα 22 Περιγραφή μιας τιμής

Τέλος η δομή μιας συμβολοακολουθίας ορίζεται με τον ίδιο τρόπο που ορίζεται και στις περισσότερες γλώσσες προγραμματισμού [JSON].

- **JavaScript/JQuery.** Η JavaScript είναι μια δυναμική γλώσσα σεναρίων που υποστηρίζει αντικειμενοστραφή, προστακτικό και συναρτησιακό προγραμματισμό. Είναι φτιαγμένη στα πλαίσια του ECMAScript στάνταρ και χρησιμοποιείται κυρίως για client-side προγραμματισμό αφού τρέχει πάνω στον Web Browser. Αυτό δίνει τη δυνατότητα για πολύ εύκολη δημιουργία δυναμικού περιεχομένου χωρίς να στέλνονται δεδομένα στον Web Server [JS].

Η λειτουργικότητα και η αποτελεσματικότητα της JavaScript στον client-side προγραμματισμό, απογειώνονται όμως με τη χρήση της βιβλιοθήκης JQuery, που

είναι μια βιβλιοθήκη φτιαγμένη για να δουλεύει σε όλους τους δημοφιλείς Web Browser. Με τις μεθόδους που παρέχει στους προγραμματιστές καθίσταται εξαιρετικά εύκολη η δημιουργία δυναμικού περιεχομένου, αλλά και η επικοινωνία με τον Web Server για την άντληση ή επεξεργασία ορισμένων πληροφοριών [JQ].

Να σημειώσουμε ότι όλα αυτά τα εργαλεία τρέχουν στους σύγχρονους φυλλομετρητές κανονικά, οπότε ο χρήστης αρκεί να έχει κάποιον σύγχρονο φυλλομετρητή όπως ο Google Chrome ή ο Mozilla Firefox.

6.2.1.2 *Java Enterprise Edition*

Η γλώσσα Java δημιουργήθηκε στα μέσα της δεκαετίας του 90 από την εταιρία Sun. Είχε ως πρωταρχικό στόχο να βελτιώσει ορισμένες αδυναμίες της δημοφιλούς αντικειμενοστραφούς γλώσσας C++. Αν και στη πρώτη της έκδοση τα χαρακτηριστικά της γλώσσας ήταν πρωτόγονα σε σχέση με αυτά της C++, τα προγράμματα που γράφονταν σε αυτή -που σήμερα αποκαλούμε μικροεφαρμογές (applets) – μπορούσαν να εκτελούνται σαν μέρος των ιστοσελίδων από τον φυλλομετρητή Netscape Navigator. Αυτή ήταν και η πρώτη μορφή web διαδραστικού προγραμματισμού και για αυτό το λόγο η γλώσσα έγινε πολύ γρήγορα δημοφιλής. Με τον καιρό εξελίχθηκε, βελτιώθηκε και σήμερα τη χρησιμοποιούν περισσότεροι προγραμματιστές σε σχέση με τους προγραμματιστές που χρησιμοποιούν C++.

Είναι μια αντικειμενοστραφής γλώσσα με κύρια χαρακτηριστικά την ανεξαρτησία πλατφόρμας, τον ασφαλή προγραμματισμό και την εύκολη εκμάθηση από προγραμματιστές που έχουν εμπειρία σε γλώσσες που μοιάζουν με τη C [JV6].

Είναι μια πλήρως αντικειμενοστραφής γλώσσα με χαρακτηριστικά όπως η μονή κληρονομικότητα έτσι ώστε να διευκολύνεται η χρήση της [JV6].

Η ασφάλεια της Java οφείλεται κυρίως στο γεγονός ότι δε περιλαμβάνει pointers καθώς και στο ότι η δέσμευση και στη συνέχεια η αποδέσμευση μνήμης γίνονται δυναμικά (garbage collector) [JV6]

Η ανεξαρτησία από τη πλατφόρμα υλοποιείται με τον εξής τρόπο:

Όταν ένα πρόγραμμα σε Java ολοκληρωθεί, μεταγλωττίζεται σε μια μορφή που αποκαλείται bytecode. Ένα αρχείο σε bytecode μπορεί να εκτελείται μετά σε έναν interpreter που αποκαλείται JVM (Java Virtual Machine), ο οποίος είναι διαθέσιμος για πολλές αρχιτεκτονικές και λειτουργικά συστήματα, οπότε έτσι ο ίδιος κώδικας μπορεί να εκτελεστεί σε διαφορετικές αρχιτεκτονικές. Συγκεκριμένα η Sun είχε δηλώσει ότι υπάρχουν πάνω από 5.5 δισεκατομμύρια συσκευές που έχουν ενεργοποιημένο JVM [JVM].

6.2.1.3 Java Servlets/Java Server Pages

Η γλώσσα προγραμματισμού Java προσφέρεται για τη χρήση ενός web server σε πλατφόρμα για την ανάπτυξη εφαρμογών. Για αυτόν το σκοπό υπάρχουν τα servlets, τα οποία χρησιμοποιώντας τα μπορούμε να δεχόμαστε είσοδο από χρήστες, να παρουσιάζουμε εγγραφές από κάποια βάση δεδομένων και να δημιουργούμε ιστοσελίδες με δυναμικό τρόπο. Αυτή η προσέγγιση εμπλουτίζεται και με τη χρήση των Java Server Pages (JSP), μιας τεχνολογίας κατασκευής ιστοσελίδων που συνδυάζει στατικό περιεχόμενο HTML με την έξοδο που παρέχεται από servlets και εκφράσεις Java [JV6].

Τα servlets είναι κλάσεις της Java που εκτελούνται από ένα web server πάνω σε ένα διερμηνευτή, τη μηχανή εκτέλεσης servlet (servlet engine), ο οποίος υποστηρίζει τη προδιαγραφή Java Servlet και είναι βελτιωμένος έτσι ώστε η εκτέλεση των servlets να καταναλώνει ελάχιστους πόρους. Συγκεκριμένα, υποστηρίζουν λειτουργία με νήματα και έτσι για κάθε servlet ένα αντίγραφο του φορτώνεται στη μνήμη, «γεννώντας» νήματα για τον χειρισμό περισσότερων από έναν χρηστών. Τα πακέτα της Java που χρησιμοποιούνται για τη δημιουργία των servlets είναι τα javax.servlet και javax.servlet.http τα οποία αποτελούν στάνταρ μέρος της Java Enterprise Edition, μιας διευρυμένης έκδοσης της βιβλιοθήκης κλάσεων της Java [JV6].

Κάθε servlet κληρονομεί τα χαρακτηριστικά του από τη κλάση HttpServlet η οποία περιλαμβάνεται στο πακέτο javax.servlet. Αυτή η κλάση περιλαμβάνει μεθόδους οι οποίες αντιπροσωπεύουν τον κύκλο ζωής ενός servlet και συλλέγουν πληροφορίες από τον server στον οποίο αυτό τρέχει [JV6].

Συγκεκριμένα, η μέθοδος init(ServletConfig) καλείται αυτόματα όταν ένας web server θέτει για πρώτη φορά ένα servlet σε κατάσταση online, για να χειριστεί την αίτηση ενός χρήστη. Αντίστοιχα υπάρχει και η μέθοδος destroy η οποία θέτει ένα servlet σε κατάσταση offline. Προκειμένου ένα servlet να μη «κολλήσει», η μέθοδος αυτή καλείται και αυτόματα αν περάσει ένα συγκεκριμένο χρονικό διάστημα [JV6].

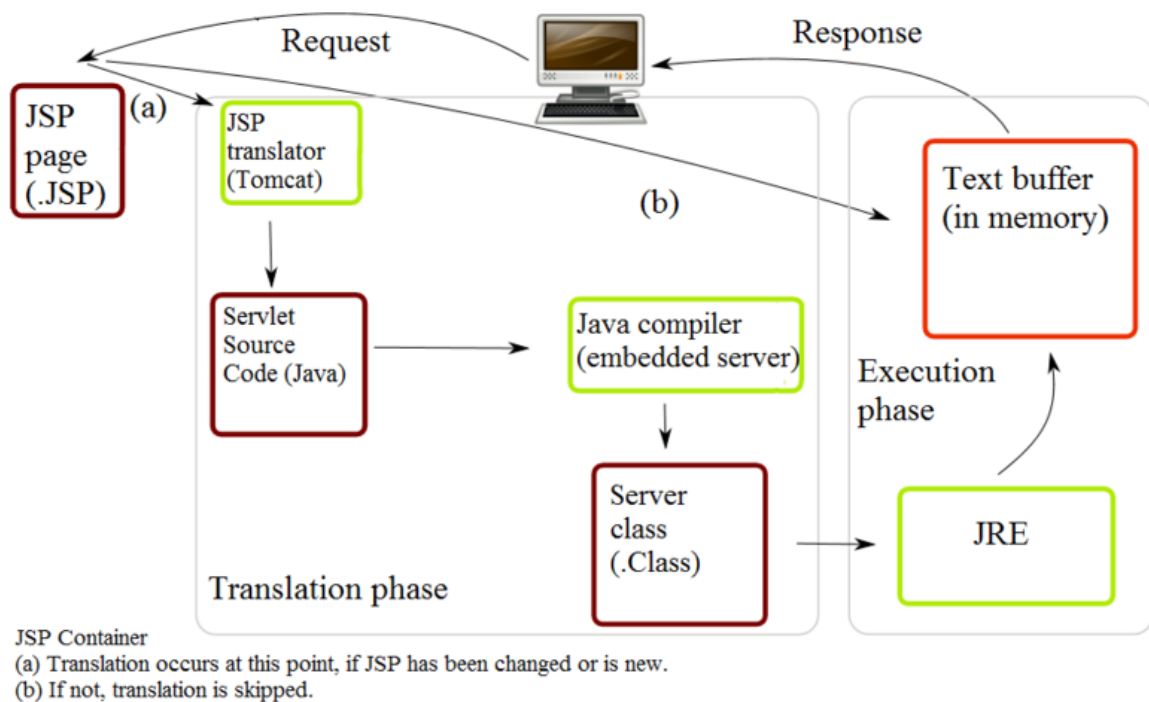
Επίσης υπάρχουν και οι μέθοδοι doGet() και doPost() για τον χειρισμό των αιτήσεων http, ανάλογα με τον τύπο τους. Η doGet() χρησιμοποιείται για αιτήσεις HTTP Get, στις οποίες όλα τα δεδομένα μιας φόρμας προσαρτώνται στο τέλος μιας διεύθυνσης Url. Αντίστοιχα η doPost() χειρίζεται αιτήσεις HTTP Post στις οποίες τα δεδομένα στέλνονται σε ξεχωριστό header και όχι στη Url [JV6].

Τα Servlets όμως αν και διευκολύνουν τη παραγωγή Html περιεχομένου με δυναμικό τρόπο δημιουργώντας σελίδες οι οποίες αλλάζουν σαν απάντηση στην είσοδο που τους έστειλε ο χρήστης, είναι ιδιαίτερα αντιπαραγωγικά όσον αφορά τη παραγωγή στατικού Html

περιεχομένου. Επίσης επειδή για τη παραγωγή του στατικού Html περιεχομένου απαιτείται προφανώς χρήση της Java και άρα γνώσεις προγραμματισμού, η σχεδίαση των ιστοσελίδων θα έπρεπε να γίνεται και αυτή από προγραμματιστές. Για να αντιμετωπίσει αυτά τα προβλήματα καθώς και τον ανταγωνισμό από υπηρεσίες όπως η PHP και η ASP, η Sun δημιούργησε τα JSP. Οι σελίδες JSP λειτουργούν συμπληρωματικά ως προς τα Servlet αφού διαχωρίζουν το περιεχόμενο Web, σε στατικό περιεχόμενο που δεν χρειάζεται να αλλάξει ποτέ και σε δυναμικό το οποίο παράγεται μέσα από Servlets. Έτσι στην πράξη, οι σελίδες JSP είναι κυρίως σελίδες HTML με κομμάτια από κώδικα Java μέσα σε αυτές (scriptlets) [JV6].

Στην ουσία δηλαδή τα JSP μεταφράζονται αυτόματα σε Servlets με χρήση του JavaServer Pages compiler όπως φαίνεται και στη παρακάτω εικόνα [JVSRVLT].

Τέλος αξίζει να σημειωθεί ότι η δημοφιλέστερη επιλογή web server για νέους δημιουργούς servlets είναι ο Apache Tomcat ο οποίος παρέχεται δωρεάν για μεταφορά στα συστήματα των χρηστών μπορεί να τρέχει παράλληλα με έναν άλλο web server ή σαν αυτόνομος server [JV6].



Εικόνα 23 Αναλυτικός τρόπος λειτουργίας των JSP

6.2.1.4 MySQL

Η βάση δεδομένων που χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής μας είναι η MySQL. Είναι η πιο δημοφιλής βάση δεδομένων ανοιχτού κώδικα, με πάνω από 100

εκατομμύρια υπολογιστές να την έχουν εγκατεστημένη. Αναπτύχθηκε με σκοπό να είναι πολύ γρήγορη, αξιόπιστη και εύκολη. Τη χρησιμοποιούν τεράστιες ιστοσελίδες, συστήματα επιχειρήσεων και κορυφαίες εταιρίες όπως η Yahoo!, Alcatel-Lucent, Google, Nokia, Youtube, Wikipedia και Booking.com. Είναι μέρος κλειδί του LAMP (Linux, Apache, MySQL, PHP/Perl/Python) ενός πακέτου λογισμικού που χρησιμοποιούν όλο και περισσότερες εταιρίες λόγω του φθηνότερου κόστους του από ιδιόκτητο λογισμικό [SQL].

6.2.2 Εργαλεία ανάπτυξης Συστήματος

Ολόκληρη η εφαρμογή αναπτύχθηκε και δοκιμάστηκε χρησιμοποιώντας το Eclipse IDE, ένα ολοκληρωμένο περιβάλλον ανάπτυξης που χρησιμοποιείται κυρίως για ανάπτυξη εφαρμογών σε Java αν και λόγω του γεγονότος ότι οι δυνατότητες του επεκτείνονται με plug-ins υποστηρίζει την ανάπτυξη σε πάρα πολλές γλώσσες [ECL].

6.3 Οδηγίες εγκατάστασης

Στο παρακάτω κεφάλαιο δίνονται οι οδηγίες εγκατάστασης της εφαρμογής σε λειτουργικό σύστημα Windows.

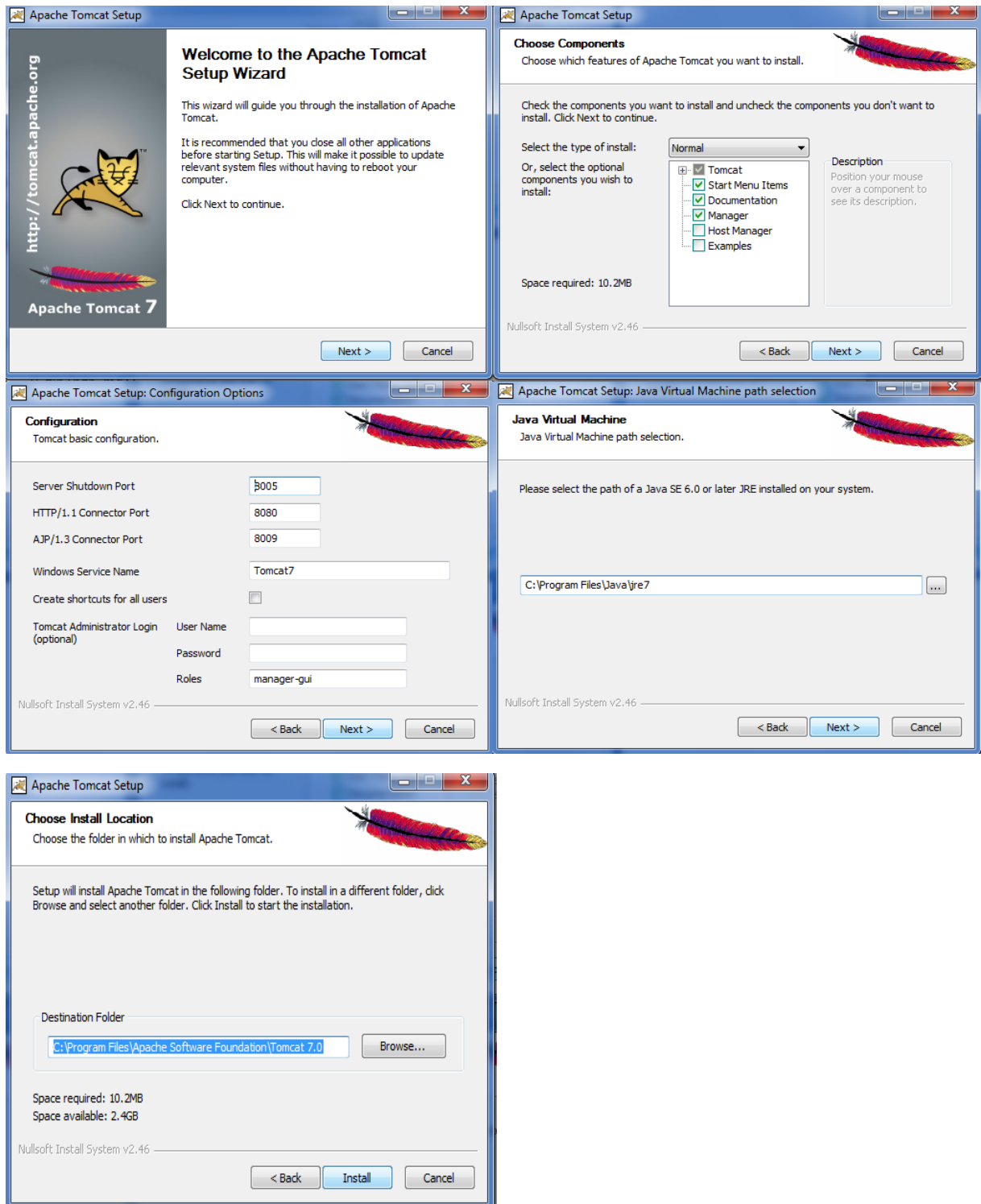
6.3.1 Εγκατάσταση Java SE

Μεταβαίνουμε στην επίσημη ιστοσελίδα της Oracle όπου δίνεται δυνατότητα λήψης της τελευταίας έκδοσης (Java 7) :

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>. Επιλεγούμε την κατάλληλη έκδοση για το σύστημα μας και προχωράμε στην εγκατάσταση

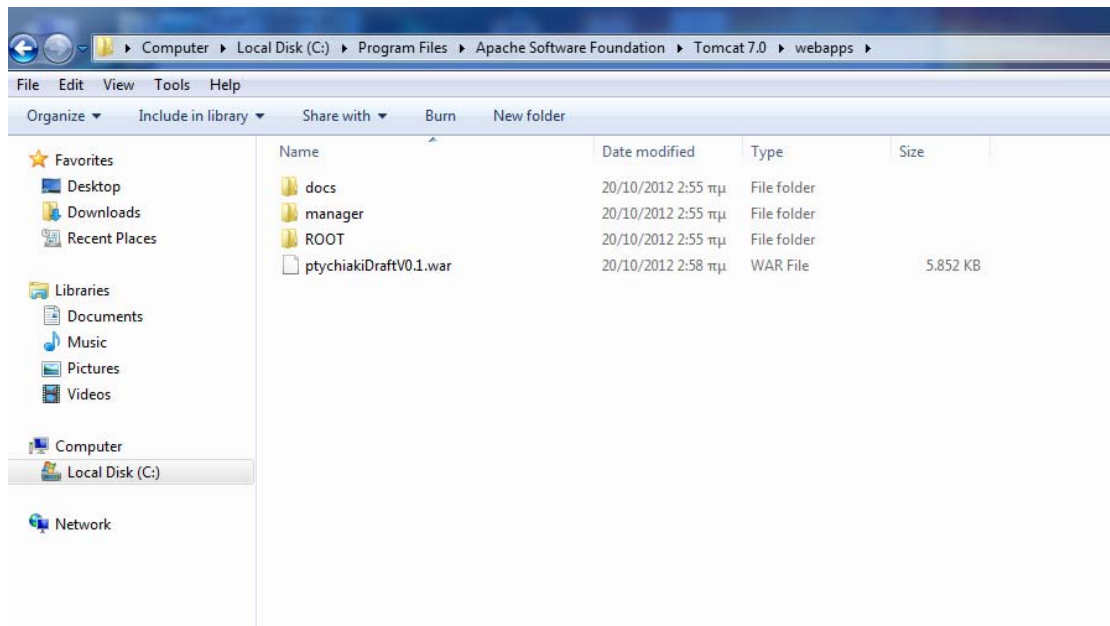
6.3.2 Εγκατάσταση Tomcat

Μεταβαίνουμε στην ιστοσελίδα <http://tomcat.apache.org/download-70.cgi> όπου κατεβάζουμε την τελευταία έκδοση του Windows Service Installer..Μετά το κατέβασμα ,τον εγκαθιστούμε. Αφήνουμε τις default επιλογές χωρίς να βάλουμε κάποιον κωδικό.



Εικόνα 24 Εγκατάσταση Tomcat

Αντιγράφουμε το WAR file στο webapps φάκελο του Tomcat που βρίσκεται στο directory εγκατάστασης του.



Εικόνα 25 Webapps directory του Tomcat

Τέλος μεταβαίνουμε στην ιστοσελίδα http://localhost:8080/ptychiaki_draft/InitialScreen.jsp.

6.3.3 Εγκατάσταση MySQL

Μεταβαίνουμε στην ιστοσελίδα <http://dev.mysql.com/downloads/mysql/5.5.html> όπου κατεβάζουμε την τελευταία έκδοση του MySQL Community Server. Ακλουθούμε την εγκατάσταση με τις default ρυθμίσεις, επιλέγοντας μόνο το checkbox με label 'Include Bin Directory in Windows Path'. Για Server Host βάζουμε 127.0.0.1, για port 3306, για username root και για password megathan.

Με τον παρακάτω κώδικα φτιάχνουμε εύκολα τη βάση δεδομένων μας. Τον αποθηκεύουμε αρχικά σε ένα αρχείο txt, έστω mydatabase.txt. Έπειτα ανοίγουμε γραμμή εντολών και πληκτρολογούμε την εξής εντολή:

```
mysql -u root -pmegathan <mydatabase.txt
```

Η βάση δεδομένων έχει έτσι δημιουργηθεί και η εφαρμογή πλέον μπορεί να έχει κανονική πρόσβαση σε αυτή.

```

DROP DATABASE IF EXISTS `ptychiaki_draft`;
CREATE DATABASE `ptychiaki_draft`;
CREATE TABLE `ptychiaki_draft`.`basic` (
  `id` BIGINT UNSIGNED NOT NULL AUTO_INCREMENT,
  `User` VARCHAR(45) NOT NULL,
  `Post` LONGTEXT NOT NULL,
  `Language` VARCHAR(45) NOT NULL,
  `SessionId` LONGTEXT NOT NULL,
  `TweetId` BIGINT UNSIGNED NOT NULL,
  PRIMARY KEY (`id`)
)
ENGINE = InnoDB
CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
CREATE TABLE `ptychiaki_draft`.`basicurl` (
  `id` BIGINT UNSIGNED NOT NULL AUTO_INCREMENT,
  `Url` LONGTEXT NOT NULL,
  `TweetId` BIGINT UNSIGNED NOT NULL,
  `SessionId` LONGTEXT NOT NULL,
  `ForeignId` BIGINT UNSIGNED NOT NULL,
  PRIMARY KEY (`id`)
)
ENGINE = InnoDB
CHARACTER SET utf8 COLLATE utf8_general_ci;
CREATE TABLE `ptychiaki_draft`.`relatedhash` (
  `id` BIGINT UNSIGNED NOT NULL AUTO_INCREMENT,
  `hash` VARCHAR(45) NOT NULL,
  `relatedhashtag` VARCHAR(45) NOT NULL,
  `numberofapp` VARCHAR(45) NOT NULL,
  `foreignid` BIGINT UNSIGNED NOT NULL,
  `sessionid` LONGTEXT NOT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `FK_relatedhash_1` FOREIGN KEY `FK_relatedhash_1`
(`foreignid`)
  REFERENCES `basic` (`id`)
  ON DELETE CASCADE
  ON UPDATE CASCADE
)
ENGINE = InnoDB
CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;

CREATE TABLE `ptychiaki_draft`.`campaignInfo` (
  `ThreadId` BIGINT UNSIGNED NOT NULL AUTO_INCREMENT,
  `Name` VARCHAR(45) NOT NULL,
  `Mail` VARCHAR(45) NOT NULL,
  `Status` VARCHAR(45) NOT NULL,
  `StartingDate` VARCHAR(255) NOT NULL,
  `Duration` VARCHAR(45) NOT NULL,
  `SearchString` VARCHAR(45) NOT NULL,
  `Language` VARCHAR(45) NOT NULL,
  `ResultType` VARCHAR(45) NOT NULL,
  `IncludeExpanded` TINYINT(1) UNSIGNED NOT NULL,
  PRIMARY KEY (`ThreadId`)
)
ENGINE = InnoDB
CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;

```

```

CREATE TABLE `ptychiaki_draft`.`campaign` (
  `id` BIGINT UNSIGNED NOT NULL AUTO_INCREMENT,
  `User` VARCHAR(45) NOT NULL,
  `Post` VARCHAR(255) NOT NULL,
  `Language` VARCHAR(45) NOT NULL,
  `threadid` BIGINT UNSIGNED NOT NULL,
  `TweetId` BIGINT UNSIGNED NOT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `FK_campaign_1` FOREIGN KEY `FK_campaign_1` (`threadid`)
    REFERENCES `campaigninfo` (`ThreadId`)
    ON DELETE CASCADE
    ON UPDATE CASCADE
)
ENGINE = InnoDB
CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;

CREATE TABLE `ptychiaki_draft`.`campaignurl` (
  `id` BIGINT UNSIGNED NOT NULL AUTO_INCREMENT,
  `Url` VARCHAR(500) NOT NULL,
  `TweetId` BIGINT UNSIGNED NOT NULL,
  `ThreadId` VARCHAR(45) NOT NULL,
  `ForeignId` BIGINT UNSIGNED NOT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `FK_campaignurl_1` FOREIGN KEY `FK_campaignurl_1`
    (`ForeignId`)
    REFERENCES `campaign` (`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE
)
ENGINE = InnoDB
CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;

CREATE TABLE `ptychiaki_draft`.`campaignsortedurl` (
  `id` BIGINT UNSIGNED NOT NULL AUTO_INCREMENT,
  `Url` VARCHAR(500) NOT NULL,
  `ThreadId` BIGINT UNSIGNED NOT NULL,
  `numberofapp` BIGINT UNSIGNED NOT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `FK_campaignsortedurl_1` FOREIGN KEY
    `FK_campaignsortedurl_1` (`ThreadId`)
    REFERENCES `campaigninfo` (`ThreadId`)
    ON DELETE CASCADE
    ON UPDATE CASCADE
)
ENGINE = InnoDB
CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
CREATE TABLE `ptychiaki_draft`.`campaignextendedurl` (
  `id` BIGINT UNSIGNED NOT NULL AUTO_INCREMENT,
  `Url` VARCHAR(500) NOT NULL,
  `ThreadId` BIGINT UNSIGNED NOT NULL,
  `OriginalUrl` VARCHAR(500) NOT NULL,
  `numberofapp` BIGINT UNSIGNED NOT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `FK_campaignextendedurl_1` FOREIGN KEY
    `FK_campaignextendedurl_1` (`ThreadId`)
    REFERENCES `campaigninfo` (`ThreadId`)
    ON DELETE CASCADE
    ON UPDATE CASCADE
)
ENGINE = InnoDB
CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;

```

```

CREATE TABLE `ptychiaki_draft`.`campaignrelatedhash` (
  `id` BIGINT UNSIGNED NOT NULL AUTO_INCREMENT,
  `relatedhashtag` VARCHAR(45) NOT NULL,
  `numberofapp` BIGINT UNSIGNED NOT NULL,
  `threadId` BIGINT UNSIGNED NOT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `FK_campaignrelatedhash_1` FOREIGN KEY
`FK_campaignrelatedhash_1` (`threadId`)
  REFERENCES `campaigninfo` (`ThreadId`)
  ON DELETE CASCADE
  ON UPDATE CASCADE
)
ENGINE = InnoDB
  CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;

```

Κώδικας για τη δημιουργία της βάσης

7

Έλεγχος

7.1 Μεθοδολογία ελέγχου

Ο Δημήτρης είναι Ελληνοαμερικάνικος και λόγω του καλού καιρού στην Ελλάδα, αποφάσισε να επεκτείνει τις διακοπές του και τον Οκτώβριο. Όμως ενδιαφέρεται πολύ για τις πολιτικές εξελίξεις στη πατρίδα του, ενόψει των εκλογών του Νοεμβρίου. Έτσι αποφασίζει να χρησιμοποιήσει την εφαρμογή μας για να ενημερωθεί για τη διαμάχη των δύο αντιπάλων.

7.2 Αναλυτική παρουσίαση ελέγχου

Αρχικά ο Δημήτρης ανοίγει την εφαρμογή μας και κάνει μια διερευνητική αναζήτηση, εισάγοντας ως hashtag το USElections.

USElections

Campaign Management

Εικόνα 26 Initial Screen

Με την ολοκλήρωση της αναζήτησης, μεταβαίνει στην επόμενη σελίδα όπου βλέπει τα αποτελέσματα του.

Campaign Refinement

women glassceiling Israel EEUU Politics MagGh debates barackobama
MittRomney GWU **Obama** mitromney Elections2012 Romnesia World USA2012
izboriSAD USelections12 **Romney** earlyvoting

Posts | **URLs** | ExpandedURLs

<http://washdiplomat.com/DPouch/2012/October/story1uselections.html> (3 appearances)
<http://lemde.fr/OMVeSV> (3 appearances)
<http://lemde.fr/S1caD8> (2 appearances)
<http://lemde.fr/OMVeT9> (2 appearances)
http://www.liberation.fr/monde/2012/10/19/en-cas-d-echec-romney-arretera-la-politique_854352 (2 appearances)
http://lnkd.in/94TU_U (2 appearances)
<http://lemde.fr/ONHk2U> (2 appearances)
<http://lemde.fr/ONJz6C> (2 appearances)
<http://lemde.fr/RXbKgZ> (1 appearances)
<http://www.youtube.com/watch?v=hDTT1yRNsFE&list=ULhDTT1yRNsFE> (1 appearances)
<http://latinosusa.blog.lemonde.fr/2012/10/19/les-hispaniques-de-moins-en-moins-conservateurs-en-matiere-de-moeurs/> (1 appearances)
<http://is.gd/1AwHM9> (1 appearances)
<http://bit.ly/QBIBXe> (1 appearances)
<http://usat.ly/WwkIHH> (1 appearances)
<http://dld.bz/bQgTU> (1 appearances)
<http://lemde.fr/S1caDa> (1 appearances)
<http://lemde.fr/WFgFr4> (1 appearances)
<http://lemde.fr/WFFV0k> (1 appearances)
http://www.lavie.fr/actualite/billets/elections-americales-le-vote-tres-convoite-des-hispaniques-protestants-19-10-2012-32332_288.php (1 appearances)

Query Terms
USElections
Language:
Result Type:

Launch Campaign
Campaign Name:
Time: days, hours
Email:
 Expand Compressed Urls

Twitter Search
Campaign Management

Εδώ του κεντρίζει το ενδιαφέρον το hashtag Obama από το θεματικό νέφος, οπότε αποφασίζει να το συμπεριλάβει στην αναζήτηση του. Με απλό πάτημα το επιλεγεί και το προσθέτει στο Query Terms. Επίσης επιλέγει τη γλώσσα των tweets που θα επιστραφούν, να

είναι αγγλικά. Πατά Update Search για να του παρουσιαστούν τα νέα αποτελέσματα. Αφού τα ελέγξει και δει ότι τον ικανοποιούν, συμπληρώνει τα απαραίτητα πεδία και είναι έτοιμος να ξεκινήσει μια campaign πατώντας το κουμπί Launch!

Campaign Refinement

RonPaul [Romnesia](#) progress VOTE [Desperate](#) Drudge Poll [GOP](#) Libya
 barackobama [obama2012](#) Virginia [dsc](#) [tlot](#) [tcot](#) [CantAfford4More](#)
 Romney [dccc](#) [p2](#) [Longoria](#)

Posts
URLs
ExpandedURLs

Query Terms

USElections Obama

Language: English

Result Type: -----

[Update Search](#)

Launch Campaign

Campaign Name: USElections2012c

Time: 0 days, 4 hours

Email: electrothan@hotmail.com

Expand Compressed Urls

[Launch](#)

[PrgrsvArchitect](#): Vote for Progress to 2050 not Regression towards the 1950s. Vote Obama & for your local Democrats! #progress #Obama #dccc #dsc

[ObamaNewsTweetr](#): Obama talks housing...on The Daily Show - CNN <http://t.co/ocZfEp8y> #barackobama #obama

[SCPioneer](#): Top Lib Calls for Eva Longoria's Resignation from #Obama Campaign <http://t.co/E3H2M8zA> via @BreitbartNews #Desperate #Longoria insults women

[NancyJakubiak](#): RT @RCdeWinter: #Obama is prepared to #veto any legislation re #fiscalcliff unless #GOP agrees to tax hike for the wealthy. I say fine! <http://t.co/CdDHD5aq>

[angeliquerene](#): "@CarrollStandard: #Obama Crew Siphoned \$9B -- Your Money -- to Create 910 'GreenScheme' #Jobs at \$9.8M Each <http://t.co/bwBZ3m0t>" #tcot

[Bigdog327](#): @CherDuncombe Thanks for the follow, we need this election, more than we ever needed one before. #Obama

[LitMargaretNan](#): RT @RonPaul2012FTW: #Obama should be impeached & incarcerated. That said, #Romney should be disqualified & arrested. VoteLiberty #tcot #RonPaul

[ghhshirley](#): Chronicling Mitt Romney's lies, Vol. XXXIX <http://t.co/HlPhYgtQ> #GOP #tcot #p2 #VOTE #DEM #Obama

[cmpnwtr](#): RT @ak2sandiego: #Virginia - Vets orgs endorsing #Obama because he's PRO Vet,VA,Military,Families. Republicans send U to war, vote against your benefits.VOTE

[geogwa](#): Lol Obamacare covers Romnesia.....that made my day. #Romney #Obama #Romnesia

[johnmatthew19](#): #Breaking #News> > Latest Presidential Gallup #Poll>> > #Romney 51% -- #Obama 45% and falling..> <http://t.co/8bqpFFTX> #Ohio #Iowa #Wisconsin

[Twitter Search](#)
[Campaign Management](#)

Εικόνα 27 Campaign Refinement

Αφού περάσουν οι 4 ώρες που είχε βάλει ως διάρκεια, λαμβάνει ένα mail, που τον ενημερώνει ότι η campaign έχει ολοκληρωθεί. Μπαίνει αμέσως στη σελίδα του Campaign Management ,την επιλέγει και πατά view.

Campaign Management

Campaign Name: USElections2012
 Search Quote: USElections Start Date: Thu Oct 11 22:06:39 EEST 2012 Duration: 0 days and 4 hours
 Mail: electrothan@hotmail.com UrlExpansion: true Status: Finished

Campaign Name: name
 Search Quote: google Start Date: Fri Oct 12 13:17:17 EEST 2012 Duration: 0 days and 6 hours
 Mail: electrothan@hotmail.com UrlExpansion: false Status: Finished

Campaign Name: USElections2012b
 Search Quote: USElections Obama2012 Start Date: Fri Oct 12 15:10:43 EEST 2012 Duration: 0 days and 6 hours
 Mail: electrothan@hotmail.com UrlExpansion: true Status: Interrupted

Campaign Name: USElections2012c
 Search Quote: USElections Obama Start Date: Fri Oct 19 22:50:31 EEST 2012 Duration: 0 days and 4 hours
 Mail: electrothan@hotmail.com UrlExpansion: true Status: Finished

view

delete

stop now

export JSON

update deadline

Time:
 days, hours

relaunch interrupted

Time:
 days, hours

Twitter Search

Εδώ βλέπει όλα τα αποτελέσματα που επιστραφήκαν .Καθώς δεν επιθυμεί να δει τα short urls, επιλέγει την καρτέλα expandedUrls.

Campaign Viewer

Posts

URLs

ExpandedURLs

<http://www.sltrib.com/sltrib/opinion/55019844-82/romney-obama-state-president.html.csp> (66 appearances)

<http://www.carolineglick.com/e/2012/10/justice-for-ft-hood-heroes.php> (22 appearances)

<http://www.sltrib.com/sltrib/opinion/55019844-82/romney-obama-state-president.html.csp> (18 appearances)

http://www.aljazeera.com/indepth/features/2012/10/2012101912039948199.html?utm_content=automate&utm_campaign=Trials&utm_source=NewSocialFlow&utm_term=plustweets&utm_medium=twitter (17 appearances)

http://www.aljazeera.com/indepth/features/2012/10/2012101912039948199.html?utm_content=automate&utm_campaign=Trials&utm_source=NewSocialFlow&utm_term=plustweets&utm_medium=twitter (13 appearances)

<http://www.sltrib.com/sltrib/opinion/55019844-82/romney-obama-state-president.html.csp> (12 appearances)

<http://hinterlandgazette.com/2012/10/republicans-benghazi-consulate-attack-partisan-voted-cut-state-dept-funding.html> (11 appearances)

<http://www.politicususa.com/obama-exposed-romney-brat-challenged-corrected.html> (11 appearances)

Definition

Campaign Name: USElections2012c

Search Quote: USElections Obama

Start Date: Fri Oct 19 22:50:31 EEST 2012

Duration: 0 days and 4 hours

Status: Finished

Mail: electrothan@hotmail.com

UrlExpansion: true

export JSON

Hashtag Cloud:

Twitter Search
Campaign Management

USA Syria p2 tlot notoptimal gop News Virginia Obamnesia
 DNC DEM Libya election terrorist FtHood drones OWS Muslim
 Romney

Εικόνα 28 Campaign Viewer

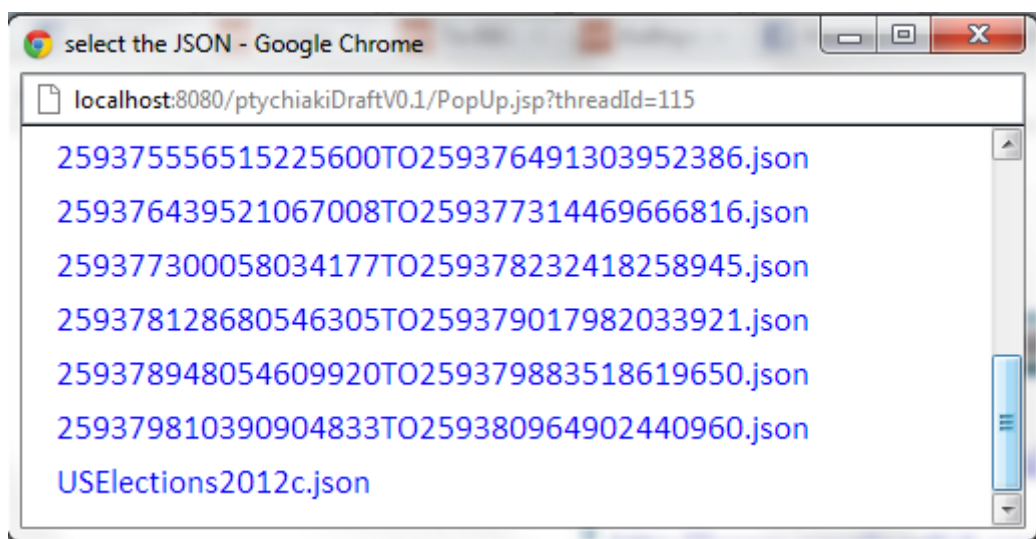
Αποφασίζει επίσης να δει ένα μεγαλύτερο θεματικό νέφος, οπότε γράφει 100 στο πεδίο hashtag cloud και πατά go.

Hashtag Cloud:

USDOR terrorist cot **teaparty** ows romneyryan Jobs US POLLS
dems LGBT Florida USelection2012 epicfail ccot POLL Muslim
VoteDEM USA China women RNC DEM debate bahrain tromney2012
Virginia presidentialdebate tiot News 2012election notoptimal Pelosi
democrat gmu US2012 Ohio **RomneyRyan2012**
FtHood lnyhbt **obama2012** Newspaper Syria DNC
Romnesia Congress **tlot** bcot Reid waronwomen **Obamnesia**
uk Msnbc Stop topprog TaggThePussy jcot 4moreyears
tcot OR TheView MLK election2012 war election Bush biden NATO
Romney Romney2012 **vote** RonPaul mitt potus **gop**
politics **p2** MSM president debates mittromney prolife
America Forward bigbird CantAfford4More binder GOTV
wiunion Muslims Libya fem2 drones PityPartyAnn sensata Benghazi
World TURKEY **Obamnesia** Gallup

Εικόνα 29 Campaign Viewer - Θεματικό νέφος με 100 το πολύ στοιχεία

Τέλος αποφασίζει να δει και ένα JSON αρχείο. Πατά το κουμπί export JSON με το οποίο του εμφανίζεται ένα παράθυρο με τα διάφορα αρχεία που μπορεί να κατεβάσει.



Εικόνα 30 - Campaign Viewer - Export JSON

Επιλέγει ένα από αυτά, το κατεβάζει στον υπολογιστή του, το βλέπει και το επεξεργάζεται όπως επιθυμεί.

```
1 { "Name": "USElections2012c", "Mail": "electrothan@hotmail.com", "Status": "Finished", "StartingDate": "Fri O
2 { "User": "KyllikiT", "TweetId": "259380964902440960", "Post": "#obama supporter: #Romney is 'socialist & c
3 { "User": "tigrs99", "TweetId": "259380963665145856", "Post": "RT @CNNVideo: President #Obama on a campaign
4 { "User": "Flag_of_Freedom", "TweetId": "259380955029061632", "Post": "Any #Liberals who will vote for #Oba
5 { "User": "ACLJ", "TweetId": "259380940650995712", "Post": "With Libya scandal the crime is worse than the
6 { "User": "taxdude77", "TweetId": "259380924641329152", "Post": "My bank account is #notoptimal. Thanks #Ob
7 { "User": "BryceStawski", "TweetId": "259380923865374720", "Post": "RT @RasmussenPoll: #Ohio: #Obama 49%, #
8 { "User": "ImNotScotWalker", "TweetId": "259380916835717120", "Post": "RT @SpaceCheef: #TaggThePussy's Dad
9 { "User": "PeggyStuart", "TweetId": "259380914000367616", "Post": "RT @JeffersonObama: Only one Romney Poll
10 { "User": "MAPSTER5500", "TweetId": "259380894027087873", "Post": "RT @bmmwash374: Yup! MT @mpf2011: Knowing
11 { "User": "TBreul", "TweetId": "259380870996172802", "Post": "ύ@SeekingDisciple: While I oppose President #
12 { "User": "AAIUSA", "TweetId": "259380868861276160", "Post": "#Obama coins #Romnesia as #Romney's "conditio
13 { "User": "omarbaddar", "TweetId": "259380868831928321", "Post": "#Obama coins #Romnesia as #Romney's "cond
14 { "User": "thinkoutsideth1", "TweetId": "259380860346839040", "Post": "RT @JilleBond: Yeah, you're reading
15 { "User": "PatriotLand", "TweetId": "259380855598891008", "Post": "#Forward #Obama donor is the Calif. offi
16 { "User": "YouDrill_WeKill", "TweetId": "259380850431500288", "Post": "U knw u doing something wen Morgan F
17 { "User": "KingSouthBSB", "TweetId": "259380848330162176", "Post": ""@KINGof334MOBB: s/o 2 @KingSouthBSB th
18 { "User": "OAN_SheFine", "TweetId": "259380845620637696", "Post": "#obama #romney #2012election #president
19 { "User": "_newsAggregator", "TweetId": "259380811856498688", "Post": "RT @CNNVideo: President #Obama on a
20 { "User": "katspixels", "TweetId": "259380804667445248", "Post": "Cris and his new pin!!! #obama http://t.
21 { "User": "LucyD_OnlyMe", "TweetId": "259380792038420481", "Post": "You know youre a boss if Morgan Freeman
```

Εικόνα 31 - USElections2012c.json

8

Επίλογος

Σε αυτό το σημείο θα συνοψίσουμε τη παρουσίαση και θα εκθέσουμε τα συμπεράσματα μας σχετικά με το αντικείμενο της και τέλος θα προτείνουμε μελλοντικές επεκτάσεις προκειμένου η εφαρμογή που αναπτύχθηκε να βελτιωθεί.

8.1 Σύνοψη και συμπεράσματα

Το Twitter από τη στιγμή που πρωτοεμφανίστηκε σαν υπηρεσία, έχει γνωρίσει τεράστια ανάπτυξη και είναι μια από τις πρωταρχικές υπηρεσίες social network. Οι επίδραση που είχε στην κοινωνία είναι ενδεικτική του πόσο δημοφιλής είναι. Λόγω αυτής της μαζικής χρήσης, τα δεδομένα που διακινούνται στην υπηρεσία είναι πάρα πολλά και η επεξεργασία τους είναι προφανώς μεγάλης σημασίας. Κινούμενη προς αυτή την κατεύθυνση, η εφαρμογή αυτή, προσπαθεί να συλλέξει δεδομένα από το Twitter, να ξεχωρίσει κάποια από αυτά (URLs και hashtags) και να τα αξιοποιήσει ως ένα βαθμό προκειμένου να προκύψουν κάποια επιπρόσθετα χρήσιμα δεδομένα. Χαρακτηριστικό παράδειγμα είναι η χρησιμοποίηση του LongURL API για την επέκταση των short-URLs. Ύστερα η εφαρμογή προσπαθεί να τα επεξεργαστεί έτσι ώστε να τα παρουσιάσει με κάποιο πρακτικό και κατανοητό τρόπο στο χρήστη. Τέλος, βασική λειτουργία της είναι να εξάγει τα αποτελέσματα αυτής της επεξεργασίας σε αρχεία για την περαιτέρω αξιοποίησή τους. Με αυτό τον τρόπο η εφαρμογή μπορεί να γίνει και μέρος κάποιας πιο μεγάλης πλατφόρμας που να έχει σκοπό την ανάλυση των δεδομένων του Twitter .

8.2 Μελλοντικές επεκτάσεις

Η διπλωματική μας εργασία, πέρα από τα περιθώρια ένταξης της σε κάποια πλατφόρμα για ανάλυση δεδομένων του Twitter , θα μπορούσε να βελτιωθεί λειτουργικά στους παρακάτω τομείς:

- Στο τρόπο που επιτελείται η συνεχόμενη αναζήτηση. Η εφαρμογή κάνει αναζήτηση ανά συγκεκριμένο χρονικό διάστημα (15 λεπτά) το οποίο παρατηρήσαμε ότι ήταν αρκετά μικρό ώστε να μη χάνονται Tweets μέσω του Twitter Search API το οποίο δε μας επιτρέπει να παίρνουμε μεγαλύτερο αριθμό από τα 1500 τελευταία Tweets που περιλαμβάνουν το/τα Hashtag που αναζητούμε. Όμως από την άλλη είναι πολύ συχνό φαινόμενο στις Campaigns που τρέξαμε για να ελέγξουμε την εφαρμογή να γίνονται περιττές αναζητήσεις που θα μπορούσαν να αποφευχθούν. Αυτό το στοιχείο σε συνδυασμό με το γεγονός ότι το Twitter δεν αποκαλύπτει το μέγιστο όριο αιτήσεων http που δικαιούται να κάνει προς αυτό κάποιος, μας κάνει να πιστεύουμε ότι η υλοποίηση δυναμικής αυξομείωσης του χρονικού διαστήματος μεταξύ 2 αναζητήσεων, προσαρμοσμένης για τη κάθε Campaign ξεχωριστά, θα ήταν μια χρήσιμη βελτίωση. Το συγκεκριμένο κομμάτι είναι και ιδιαίτερα σημαντικό για την εισαγωγή της λειτουργικότητας της εφαρμογής στη πλατφόρμα που αναφέρθηκε σε προηγούμενο κεφάλαιο γιατί αποτελεί βασική προϋπόθεσή της.
- Στη ταξινόμηση των Url που επιστρέφονται. Αυτή τη στιγμή η εφαρμογή μας ταξινομεί τα Url με βάση τον αριθμό εμφάνισης τους. Θα μπορούσε στη ταξινόμηση τους παρόλα αυτά να παίζουν ρόλο και γεωγραφικά κριτήρια, ή η ταυτότητα των χρηστών που τα αναρτούν κα.
- Επιπρόσθετη επεξεργασία των αποτελεσμάτων αναζήτησης, με αυτόματη εξαγωγή κάποιων στατιστικών στοιχείων που θα ήταν χρήσιμα.

9

Βιβλιογραφία

- [TWP] Y. Stavrakas, V. Plachouras, A Platform for Supporting Data Analytics on Twitter : Challenges and Objectives, IMIS/RC “ATHENA”, Athens, Greece
- [JV6] R. Cadenhead, L. Lemay, Πλήρες εγχειρίδιο της Java 6(μτφ), Εκδόσεις Γκιούρδας pp10-12, 555-575, 2007
- [JVSRLT] http://en.wikipedia.org/wiki/Java_Servlet
- [HTCS] <http://www.w3.org/standards/webdesign/htmlcss>
- [MRKP] http://en.wikipedia.org/wiki/Markup_language
- [XML] <http://www.w3.org/TR/2008/REC-xml-20081126/#NT-AttValue>
- [JSON] <http://www.json.org/>
- [JS] <http://en.wikipedia.org/wiki/JavaScript>
- [JQ] <http://jquery.com/>
- [TWT1] <https://dev.twitter.com/start>
- [TWT2] <https://dev.twitter.com/docs/using-search>
- [TWT3] <https://dev.twitter.com/docs/api/1/get/search>
- [LU] <http://longurl.org/api#expand-url>
- [TMR] <http://docs.oracle.com/javase/1.5.0/docs/api/java/util/Timer.html>
- [JVM] http://en.wikipedia.org/wiki/Java_virtual_machine

[SQL] <http://www.mysql.com/about/>

[ECL] [http://en.wikipedia.org/wiki/Eclipse_\(software\)](http://en.wikipedia.org/wiki/Eclipse_(software))