



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

Τμήμα Πληροφορικής και Τηλεματικής

**Κατεύθυνση: Υπολογιστικές και
Διαδικτυακές Τεχνολογίες
και εφαρμογές**

**Διαδικτυακή Πλατφόρμα Συντονισμού για Σύ-
ντηξη Γεωχωρικών Δεδομένων από Μετρήσεις
Απόδοσης Ασύρματων Δικτύων**

**Sensing Web Application for Location Based
Data Fusion on QoE of Wireless Networks**

Μια Διπλωματική Εργασία του

GERTI NURKA

Επιβλέποντες:

Δαλάκας Βασίλειος

Τσαδήμας Ανάργυρος

Μιχαλακέλης Χρήστος

ΑΘΗΝΑ 2015

Πρόλογος

Η εκπόνηση της παρούσας διπλωματικής εργασίας, νιώθω, πως είχε ως κύριο σκοπό την ουσιαστική έρευνα, επαφή και εφαρμογή στην πράξη, τεχνολογιών διαδικτύου που διδάχτηκα ως μεταπτυχιακός φοιτητής, αποκτώντας με αυτόν τον τρόπο πολύτιμες εμπειρίες. Πρακτικός όμως στόχος της παρούσας διπλωματικής εργασίας είναι να καλύψει τις ανάγκες πληροφόρησης των χρηστών στον τομέα της απόδοσης των ασύρματων δικτύων, δίνοντάς τους την δυνατότητα να συντονίζουν τις μετρήσεις που πρέπει να πραγματοποιηθούν για να εξαχθούν χρήσιμα συμπεράσματα που αντανakλούν την ποιότητα της υπηρεσίας όπως την αντιλαμβάνονται οι χρήστες αυτής.

Όλα τα παραπάνω δεν θα είχαν συμβεί χωρίς τους επιβλέποντες της διπλωματικής εργασίας που ήταν οι Δαλάκας Βασίλειος και Τσαδήμας Ανάργυρος, Διδάσκοντες στο ΠΜΣ του Τμήματος Πληροφορικής και Τηλεματικής στο Χαροκόπειο Πανεπιστήμιο. Τους ευχαριστώ ιδιαίτερα για την καθοδήγηση που μου προσέφεραν κατά την διάρκεια εκπόνησης της διπλωματικής, για τις συνεχείς και “φρέσκες” ιδέες που μου παρείχαν, για την πολύτιμη βοήθειά τους σε τεχνικά ζητήματα και τέλος για τον χρόνο που μου αφιέρωσαν σε πολύωρες συζητήσεις.

Δεν θα μπορούσα όμως να παραλείψω τις ιδιαίτερες ευχαριστίες μου προς τον συμφοιτητή μου και συνεργάτη μου κατά τη διάρκεια της εκπόνησης της διπλωματικής εργασίας, τον Μωυσιάδη Αναστάσιο. Με τη σειρά του ο Μωυσιάδης Αναστάσιος είχε αναλάβει, στη δική του διπλωματική εργασία, να δημιουργήσει την εφαρμογή HuaServiceTester2 η οποία είναι υπεύθυνη για την συλλογή όλων των δεδομένων που χρησιμοποιούμε στην δική μας διαδικτυακή εφαρμογή. Επομένως η εργασία του Μωυσιάδη Αναστάσιου και η εφαρμογή που ανέπτυξε είναι αναπόσπαστα κομμάτια της δικής μας εργασίας, καθώς πραγματοποιούν ένα σημαντικό και καθοριστικό έργο για εμάς το οποίο είναι η συλλογή όλων των δεδομένων. Θα ήθελα λοιπόν να τον ευχαριστήσω θερμά για όλη αυτή την εποικοδομητική και ευχάριστη συνεργασία που είχαμε κατά την διάρκεια της εκπόνησης των διπλωματικών μας εργασιών.

Τέλος, θα ήθελα να ευχαριστήσω έναν δικό μου άνθρωπο την Ornela Senja που μου προσέφερε πολύτιμη βοήθεια στην παρούσα διπλωματική εργασία, αυτή της συντακτικής και ορθογραφικής διόρθωσης του κειμένου. Την ευχαριστώ λοιπόν ιδιαίτερα για

την υπομονή της να διαβάζει και να διορθώνει ένα κείμενο άσχετο με το αντικείμενο της.

Περίληψη

Στο παρόν κείμενο παρουσιάζεται η διαδικασία σχεδιασμού και υλοποίησης της διπλωματικής εργασίας με αντικείμενο την ανάπτυξη διαδικτυακής πλατφόρμας συντονισμού για σύντηξη γεωχωρικών δεδομένων από μετρήσεις απόδοσης ασύρματων δικτύων. Στα πλαίσια του πεδίου αυτού έχει αναπτυχθεί η διαδικτυακή εφαρμογή MyWiFiSpeed, που έχει ως απώτερο σκοπό την παροχή λειτουργιών προς τους χρήστες, για το συντονισμό των μετρήσεων της απόδοσης των ασύρματων δικτύων. Επίσης, στόχος της διαδικτυακής εφαρμογής είναι η σύντηξη γεωχωρικών δεδομένων που προκύπτουν από τις μετρήσεις και παρουσίαση της παραγόμενης πληροφορίας στον χρήστη.

Οι μετρήσεις από τις οποίες προκύπτουν τα γεωχωρικά δεδομένα πραγματοποιούνται από crowdsourcing^[1] υποδομή. Ειδικότερα, υπάρχει διαθέσιμη η εφαρμογή HuaServiceTester2 για κινητές συσκευές μέσω της οποίας οι χρήστες έχουν την δυνατότητα να εκτελούν μετρήσεις που αφορούν την απόδοση των ασύρματων δικτύων και δικτύων κινητής τηλεφωνίας. Οι μετρήσεις δύναται να λαμβάνονται οποιαδήποτε στιγμή το επιθυμεί ο χρήστης της κινητής συσκευής ή συμμετέχοντας σε καμπάνιες (campaigns). Τις καμπάνιες τις δημιουργεί και τις διαχειρίζεται ο διαχειριστής μέσω της διαδικτυακής εφαρμογής και αποστέλλονται στις κινητές συσκευές με τη χρήση του GCM server της google. Τα γεωχωρικά δεδομένα που προκύπτουν από τις μετρήσεις των κινητών συσκευών δομούνται και αποθηκεύονται σε μια βάση δεδομένων από την οποία τις αντλεί η διαδικτυακή εφαρμογή μας για σύντηξη. Ο όρος σύντηξη δεδομένων αναφέρεται στις διαδικασίες και τεχνικές που ακολουθούνται με σκοπό την επίτευξη της καλύτερης δυνατής εκτίμησης και την πλήρη και έγκυρη αξιολόγηση των δεδομένων. Στη δική μας περίπτωση για τη σύντηξη των γεωχωρικών δεδομένων χρησιμοποιούμε τον GeoServer και για την παρουσίαση της παραγόμενης πληροφορία στον χρήστη, χάρτες μέσω της τεχνολογίας OpenLayers και τα Java Prime faces. Συνεπώς, στα κεφάλαια που ακολουθούν γίνεται μια ανάλυση των τεχνολογιών και των τεχνικών που είναι διαθέσιμες για την ανάπτυξη μιας διαδικτυακής πλατφόρμας. Επίσης αναφερόμαστε σε υπάρχουσες ανταγωνιστικές πλατφόρμες, καθώς και στις απαιτήσεις περιβάλλοντος για την ανάπτυξη τους. Επιπλέον παρουσιάζουμε τη δική μας υλοποίηση, τη διαδικτυακή εφαρμογή MyWiFiSpeed.

Συγκεκριμένα στο κεφάλαιο 1 κάνουμε αρχικά μια εισαγωγική αναφορά στα προβλήματα που μας ώθησαν στην δημιουργία της διπλωματικής εργασίας. Στη συνέχεια δίνουμε τον ορισμό της έννοιας του crowdsourcing καθώς και της έννοιας της υπηρεσίας βασισμένη στην τοποθεσία (LBS). Τέλος, στο κεφάλαιο 1 παρουσιάζεται η πρόταση της διπλωματικής για την επίλυση των προβλημάτων. Στο κεφάλαιο 2 καταγράφονται οι τεχνολογίες που είναι διαθέσιμες για την ανάπτυξη μιας διαδικτυακής πλατφόρμας βασισμένη στην τοποθεσία. Ιδιαίτερη όμως έμφαση, δίνεται στις τεχνολογίες που έχουν χρησιμοποιηθεί για την υλοποίηση της δικής μας πρότασης. Αυτές οι τεχνολογίες είναι από τις πιο θεμελιώδεις όπως η HTML μέχρι τις πιο απαιτητικές που είναι η χρήση κάποιων Framework όπως τα Java PrimeFaces και η χρήση βιβλιοθηκών όπως τα OpenLayers για την JavaScript. Στο κεφάλαιο 3 ορίζεται η έννοια της «σύντηξης δεδομένων» (Data Fusion) και αναφέρονται οι τεχνικές που χρησιμοποιούνται. Επιπλέον παρουσιάζουμε τα εργαλεία που υλοποιούν και προσφέρουν αλγόριθμους σύντηξης δεδομένων και συγκεκριμένα γίνεται μια εκτενής αναφορά στον GeoServer και στον αλγόριθμο Barnes Surface. Θα αναφερθούμε ειδικά σε αυτό το εργαλείο – εφαρμογή και σε αυτόν τον αλγόριθμο, γιατί είναι αυτοί που χρησιμοποιούνται στην υλοποίηση της διαδικτυακής πλατφόρμας MyWiFiSpeed. Στο κεφάλαιο 4 γίνεται μια σύντομη αναφορά σε υπάρχουσες “ανταγωνιστικές” πλατφόρμες που στοχεύουν στην παρουσίαση στατιστικών στοιχείων σχετικά με τις επιδόσεις των ασύρματων δικτύων. Συγκεκριμένα παρουσιάζονται δυο εφαρμογές το Sensorly και το OpenSignal και επιπλέον γίνεται μια σύγκριση με τη δική μας προσέγγιση, παρουσιάζοντας τις ομοιότητες και τις διαφορές. Στο επόμενο κεφάλαιο (κεφάλαιο 5) παρουσιάζονται οι απαιτήσεις περιβάλλοντος για την ανάπτυξη της διαδικτυακής μας πλατφόρμας και αντίστοιχων εφαρμογών. Δηλαδή αναφέρονται και αναλύονται όλα τα εργαλεία και οι εφαρμογές που χρησιμοποιήσαμε και εγκαταστήσαμε στο σύστημα μας για την ανάπτυξη και τη λειτουργία της διαδικτυακής εφαρμογής. Ουσιαστικά η ενότητα 5 έχει περισσότερο το χαρακτήρα ενός οδηγού (tutorial) που θα μας καθοδηγήσει μελλοντικά να μπορέσουμε να “στήσουμε” εμείς ή κάποιος τρίτος το κατάλληλο περιβάλλον για την ανάπτυξη μιας παρόμοιας πλατφόρμας ή την επέκταση της υφιστάμενης. Στο κεφάλαιο 6 γίνεται μια εκτενής τεχνική ανάλυση της εφαρμογής MyWiFiSpeed και παρουσιάζεται ο οδηγός χρήσης της. Κατά την τεχνική ανάλυση εξηγούμε τη δομή της διαδικτυακής εφαρμογής και των επιμέρους στοιχείων της όπως είναι η βάση δεδομένων, ο Geoserver και ο GCM application server. Από την άλλη, στον οδηγό χρήσης εξηγούμε τις λειτουργίες που μπορεί να εκτελέσει ο χρήστης στο περιβάλλον της εφαρμογής, αλλά και με ποιον τρόπο μπορεί να

τις εκτελέσει. Και στις δύο περιπτώσεις η ανάλυση και η περιγραφή των λειτουργιών του περιβάλλοντος της εφαρμογής θα είναι σε τεχνικό επίπεδο. Τέλος στο κεφάλαιο 7 παρατίθενται τα συμπεράσματα που αφορούν κυρίως τις γνώσεις που αποκτήσαμε από την επαφή, με τις τεχνολογίες, που είχαμε κατά την διαδικασία ανάπτυξης της διπλωματικής εργασίας. Επιπλέον αναφέρονται τα προβλήματα που αντιμετωπίσαμε, αλλά και πιθανές μελλοντικές επεκτάσεις.

Abstract

Developing a sensing Web Application for Location Based Data Fusion on QoE of Wireless Networks is the subject of this thesis. To this end, we have developed a web application named MyWiFiSpeed with primary objective to provide functionalities to users, such as coordinating performance measurements of wireless networks. Also, purpose of this web application is the fusing of geospatial data, already collected from the measurements, and presenting added value information to users.

In order to collect geospatial data we use a crowdsourcing^[1] infrastructure. In particular, HuaServiceTester2 application is available for mobile devices and through this application users of mobile devices are able to perform measurements and collect data about user quality of experience of wireless networks. Users can perform their measurements either on demand either by participating in campaigns. Campaigns can be created only by the administrator of the web application that we have developed, that is also responsible for their management. When a new campaign is created it can be sent to mobile devices through Google's GCM server. Geospatial data are received from the mobile devices through the internet and are stored in a main database from which our web application collects them for further treatment. According to Joint Directors of Laboratories (JDL) workshop^[45], the term data fusion refers to a multi-level process dealing with the association, correlation, combination of data and information from single and multiple sources to achieve refined position, identify estimates and complete and timely assessments of situations, threats and their significance. Here, GeoServer is employed to complete the data fusion process and framework of OpenLayers and PrimeFaces for presenting the extracted information to the users. Consequently, in next chapters we are providing an analysis of available technologies and techniques for developing similar web applications. In addition, existing competitive platforms and their environmental requirements during development are descust. Last but not least we are presenting our implementation, MyWiFiSpeed.

In chapter 1, an introductory reference to problems that led us to develop this thesis is presented. Next, the definition for crowdsourcing and location based services (LBS)

is given. Finally, in the end of chapter 1 a description of how our implementation solves these problems is offered. In chapter 2 we are listing available technologies that can be used to develop location based web applications (LBWA), but special emphasis will be given to technologies that have been used for the implementation of our proposal. These technologies may vary from the most fundamental as HTML, to the most demanding which may be the usage of some framework as PrimeFaces for Java and OpenLayers for JavaScript. On the other hand, in chapter 3 we are mentioning the meaning and the techniques of data fusion. Furthermore, in this chapter we are presenting the tools that implement and offer data fusion algorithms with extensive reference to GeoServer and the algorithm Barnes Surface. We will specifically refer to GeoServer and Barnes Surface, because these two are the ones we use in the implementation of MyWiFiSpeed. Moreover, chapter 4 is a brief reference to current competitive platform, aimed at presenting statistics on the performance of wireless networks. Specifically, in this chapter we will show the similarities and differences between the two applications Sensorly and OpenSignal plus a comparison with our own approach. Next chapter (Chapter 5) presents the environmental requirements for the development of our Internet platform and related applications. Moreover in this chapter we are listing and analyzing all the tools and applications that we have used and installed in our system for the development and operation of our web application. Essentially, chapter 5 plays the role of a tutorial that will guide us in the future to set-up the right environment expand this platform or to develop a similar one. In Chapter 6 we are providing an extensive technical analysis of the way we implement MyWiFiSpeed and at the end we are presenting a user manual. Specifically, in section technical analysis we will explain the structure of MyWiFiSpeed and its individual elements such as database, GeoServer and GCM application server. On the other hand, in the user manual we will explain the functionalities that can be performed and how these can be performed by the user. Finally, in Chapter 7 we are concluding, focusing on the knowledge gained from the development process of this thesis. Moreover, we reporting the problems we encountered and possible future extensions.

Περιεχόμενα

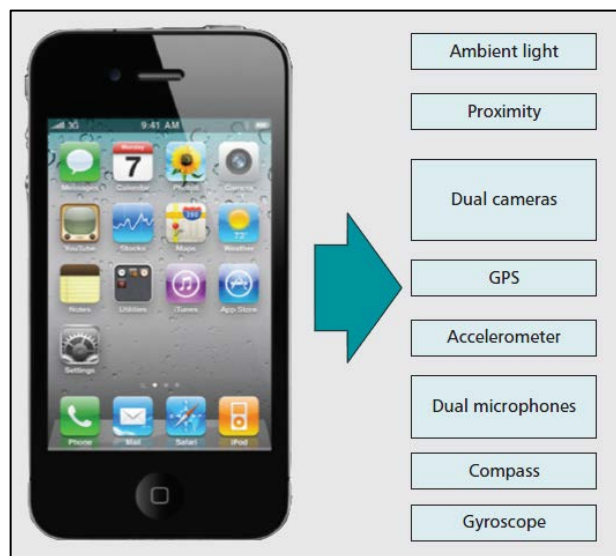
ΠΡΟΛΟΓΟΣ.....	I
ΠΕΡΙΛΗΨΗ	III
ABSTRACT.....	VII
ΠΕΡΙΕΧΟΜΕΝΑ.....	IX
1 ΕΙΣΑΓΩΓΗ	1
1.1 ΟΡΙΣΜΟΣ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ.....	1
1.2 ΠΡΟΤΑΣΗ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ	4
2 ΑΝΑΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΗΣ ΠΛΑΤΦΟΡΜΑΣ ΒΑΣΙΣΜΕΝΗ ΣΤΗΝ ΤΟΠΟΘΕΣΙΑ.....	7
2.1 ΑΝΑΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΗΣ ΠΛΑΤΦΟΡΜΑΣ & ΤΕΧΝΟΛΟΓΙΕΣ	7
2.1.1 Client Side Τεχνολογίες.....	8
2.1.2 Server Side Τεχνολογίες	18
2.2 ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΤΟΠΟΘΕΣΙΑΣ & ΤΕΧΝΟΛΟΓΙΚΕΣ ΛΥΣΕΙΣ	25
2.2.1 OpenLayers.....	26
3 ΣΥΝΤΗΞΗ ΔΕΔΟΜΕΝΩΝ (DATA FUSION)	29
3.1 GEOSERVER.....	30
3.2 BARNES SURFACE.....	33
4 ΥΠΑΡΧΟΥΣΕΣ ΠΛΑΤΦΟΡΜΕΣ.....	37
5 ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ ΕΦΑΡΜΟΓΗΣ.....	41
5.1 GLASSFISH	41
5.2 PHPMYADMIN	44
5.3 NETBEANS.....	48
5.4 GEOSERVER.....	51
5.5 GOOGLE CLOUD MESSAGING (GCM).....	54
6 ΠΑΡΟΥΣΙΑΣΗ ΕΦΑΡΜΟΓΗΣ.....	59

6.1	ΤΕΧΝΙΚΗ ΑΝΑΛΥΣΗ	61
6.1.1	<i>Βάση Δεδομένων.....</i>	<i>61</i>
6.1.2	<i>GeoServer.....</i>	<i>65</i>
6.1.3	<i>GCM Application Server.....</i>	<i>70</i>
6.1.4	<i>MyWiFiSpeed</i>	<i>77</i>
6.2	ΟΔΗΓΟΣ ΧΡΗΣΗΣ.....	91
6.2.1	<i>Σύνδεση – Εγγραφή (Log In – Sing Up)</i>	<i>91</i>
6.2.2	<i>Κεντρική Σελίδα (Home Page)</i>	<i>92</i>
6.2.3	<i>Διαχείριση Λογαριασμού Χρήστη (User Account Management).....</i>	<i>93</i>
6.2.4	<i>Στατιστικά (Statistics).....</i>	<i>94</i>
6.2.5	<i>Campaigns.....</i>	<i>98</i>
6.2.6	<i>Users.....</i>	<i>101</i>
7	ΣΥΜΠΕΡΑΣΜΑΤΑ.....	103
	ΠΑΡΑΠΟΜΠΕΣ	107

1 Εισαγωγή

1.1 Ορισμός του προβλήματος

Τα τελευταία χρόνια, σύμφωνα με το άρθρο^[2], έχει υπάρξει μια ραγδαία αύξηση των έξυπνων κινητών συσκευών, καθιερώνοντας τα ως τις κύριες υπολογιστικές συσκευές και μέσα επικοινωνίας στις ζωές των ανθρώπων. Σε αυτό κατά ένα σημαντικό ποσοστό συνέβαλε η ύπαρξη διαδικτυακών διανομέων εφαρμογών όπως το AppStore της Apple και το PlayStore της Google, μετατρέποντας έτσι τις κινητές συσκευές σε συσκευές εφαρμογών οι οποίες μπορούν να κατεβάσουν σε μικρό χρονικό διάστημα εκατοντάδες εφαρμογές. Τα δυο σημαντικότερα όμως γεγονότα που καθιέρωσαν τις έξυπνες κινητές συσκευές είναι η δυνατότητα τους να προγραμματίζονται και ο συνεχώς αυξανόμενος αριθμός των ενσωματωμένων αισθητήρων τους. Σήμερα, για παράδειγμα μια έξυπνη κινητή συσκευή συνήθως στον βασικό της εξοπλισμό μπορεί να έχει αισθητήρα ταχύτητας, ψηφιακή πυξίδα, γυροσκόπιο, GPS, μικρόφωνα, κάμερες κτλ. (Εικόνα 1). Συλλογικά, αυτοί οι αισθητήρες είναι που προσφέρουν την προοπτική ανάπτυξης νέων εφαρμογών σε ένα ευρύ φάσμα τομέων, όπως η υγειονομική περίθαλψη^[4], τα κοινωνικά δίκτυα^[5], την ασφάλεια, την παρακολούθηση του περιβάλλοντος^[6] και τις μεταφορές^{[7][8]}.



Εικόνα 1: Βασικός εξοπλισμός σημερινού έξυπνου κινητού.

Κατά την διαδικασία της ανάπτυξης των εφαρμογών για τους τομείς που προαναφέρθηκαν προκύπτουν δύο βασικές ανάγκες – προβλήματα, είτε αυτές οι εφαρμογές είναι διαδικτυακές είτε όχι. Το πρώτο πρόβλημα αφορά τη συλλογή των δεδομένων, καθώς για να είναι χρήσιμη μια εφαρμογή σε έναν τομέα πρέπει να υπάρχουν πληροφορίες διαθέσιμες στους χρήστες οι οποίες όμως πληροφορίες προέρχονται από την επεξεργασία δεδομένων.

Σύμφωνα με το άρθρο^[1] η διαδικασία της επεξεργασίας και της συλλογής των δεδομένων καταλαμβάνει ένα μεγάλο κομμάτι ποσοστού από το χρηματικό και χρονικό κόστος της ανάπτυξης της εκάστοτε εφαρμογής. Για τον λόγο αυτό, η πλειοψηφία των εφαρμογών που προορίζονται ή βασίζονται σε κινητές συσκευές χρησιμοποιούν για την συλλογή και την επεξεργασία των δεδομένων υποδομές crowdsourcing. Ο ορισμός του crowdsourcing δίνεται, από το ίδιο άρθρο, ως η ανάθεση μιας εργασίας από μια εταιρία ή ένα ίδρυμα, που εκτελούνται από τους εργαζόμενους της, σε ένα απροσδιόριστο δίκτυο ανθρώπων με την μορφή ανοιχτής πρόσκλησης. Η εργασία αυτή μπορεί να πραγματοποιηθεί σε συνεργασία ατόμων ή από μεμονωμένα άτομα. Ωστόσο το σημαντικό προαπαιτούμενο για να υπάρξει μια crowdsourcing πλατφόρμα είναι η ανοιχτή πρόσκληση και ένα μεγάλο δίκτυο από πιθανούς εργαζόμενους, μισθωτούς ή εθελοντές. Χρησιμοποιώντας crowdsourcing υποδομή για την συλλογή και επεξεργασία των δεδομένων υπάρχει ένα σημαντικό κέρδος σε χρήματα και χρόνο. Με βάση αυτά, σήμερα έχουν αναπτυχθεί πολλές εφαρμογές για έξυπνες κινητές συσκευές που βασίζονται στο crowdsourcing για την συλλογή των δεδομένων. Ενδεικτικά μερικές από αυτές είναι το sensorly^[9] για την δημιουργία χαρτών με την κάλυψη των ασύρματων και κινητών δικτύων. Το Nericell^[10] και VTrack^[7] για την παροχή πληροφοριών σχετικά με την κυκλοφοριακή κίνηση. Το PIER^[11] για τον υπολογισμό των εξατομικευμένων περιβαλλοντικών επιπτώσεων και το Ear-Phone^[12] για την δημιουργία χαρτών με αναπαράσταση της ηχορύπανσης.

Το δεύτερο πρόβλημα, είναι ο εντοπισμός και η αναπαράσταση της τοποθεσίας των έξυπνων συσκευών που συμμετέχουν στην υποδομή crowdsourcing. Με άλλα λόγια είναι η ανάγκη παράδοσης, στον χρήστη, υπηρεσιών βασισμένων στην τοποθεσία (Location Based Services – LBS)^[13], την παρουσίαση και την βελτίωση της εμπορίας χρήσης αυτών των υπηρεσιών. Οι υπηρεσίες βασισμένες στην τοποθεσία ορίζονται, σύμφωνα με το βιβλίο^[13], ως οι υπηρεσίες που ενσωματώνουν την τοποθεσία ή την θέση μιας κινητής συσκευής σε άλλα δεδομένα ώστε να προσφέρουν επιπρόσθετη πληροφορία στον χρήστη. Ο εντοπισμός της τοποθεσίας και της θέσης του χρήστη πραγματοποιείται κυρίως μέσω του παγκόσμιου συστήματος εντοπισμού θέσεις το γνωστό GPS^[15].

Ένα βήμα πιο πέρα, και αφού έχουμε συλλέξει τις πληροφορίες μέσω ενός LBS, αυτό πλέον που μας απασχολεί κατά την ανάπτυξη μιας διαδικτυακής εφαρμογής βασισμένη στην τοποθεσία είναι η παρουσίαση των πληροφοριών αυτών στον χρήστη. Για την επίτευξη του σκοπού αυτού χρησιμοποιούνται συστήματα γεωγραφικής πληροφορίας (Geographical Information System – GIS)^[16]. Το άρθρο^[16] μας εξηγεί πως μια κοινή διαδικασία για την παρουσίαση πληροφοριών με γεωγραφική αναφορά σε μια διαδικτυακή εφαρμογή περιλαμβάνει τα εξής βήματα. Πρώτα δημιουργούνται οι πληροφορίες που πρό-

κειται να παρουσιαστούν. Δεύτερον πρέπει τις πληροφορίες αυτές να τις ανεβάσουμε σε έναν εξυπηρετητή χάρτη όπως τον MapServer^[17], GeoServer^[18] κ.α. Σαν τρίτο βήμα πρέπει να κατασκευαστεί το διαδικτυακό περιβάλλον στο οποίο θα παρουσιάζονται οι πληροφορίες αυτές στους χρήστες. Σήμερα για κάθε ένα από τα βήματα αυτά υπάρχουν εργαλεία που μας βοηθάνε, όπως για τη παραγωγή των δεδομένων υπάρχουν οι εφαρμογές οι οποίες μπορούν να απεικονίζουν, αναζητούν, ενημερώνουν και να επεξεργάζονται γεωγραφικά δεδομένα. Τέτοιες εφαρμογές είναι το Quantum GIS^[19] και το GRASS^[20]. Για τη φιλοξενία των δεδομένων και τη δημιουργία χαρτών υπάρχουν οι εξυπηρετητές χαρτών δικτύου (Web Map Servers) όπως οι MapServer^[17], GeoServer^[18] κ.α. Για τη δημιουργία διαδικτυακής εφαρμογής με γεωγραφική αναφορά υπάρχουν οι Mapbender3^[21], GeoMajas^[22], MapFish^[23], GeoMOOSE^[24] και OSGIS^[16].

Τέλος, όλα τα παραπάνω πρέπει να συνδυάζονται και να παρουσιάζονται κατά τέτοιο τρόπο ώστε η εμπειρία χρήσης να είναι εύκολη και ευχάριστη στο χρήστη, σύμφωνα με το άρθρο^[14]. Επίσης, το άρθρο αυτό μας εξηγεί πως η εμπειρία χρήστη (User experience - UX) είναι μια καθιερωμένη έρευνα και χώρος εξάσκησης για την αντιμετώπιση προβλημάτων κατά το σχεδιασμό της διεπαφής χρήστη. Συγκεκριμένα, τα μεγαλύτερα προβλήματα στην εμπειρία χρήσης αντιμετωπίζονται πρώτα στον εντοπισμό της θέσης του χρήστη καθώς θέλουμε να γίνει αυτόματα χωρίς να τον απασχολούμε με τέτοιες ενέργειες και δεύτερον στα κινητά λόγω της μικρής οθόνης και της δυσκολίας στην εισαγωγή κειμένου. Για την αντιμετώπιση τέτοιων προβλημάτων το άρθρο προτείνει το location based UX (LbUX). Επιπλέον το άρθρο εμβαθύνει αναφέροντας μεταξύ άλλων πως το LbUX είναι το σύνολο των αντιλήψεων του χρήστη σχετικά με τις εφαρμογές που βασίζονται στην τοποθεσία. Μπορούμε να κατανοήσουμε αυτές τις αντιλήψεις λαμβάνοντας υπόψη τις εξής ερωτήσεις που βασίζονται στην τοποθεσία:

- Αποτελεσματικότητα: καλύπτει η εφαρμογή τις ανάγκες του χρήστη παρέχοντας λειτουργικότητα βασισμένη στη τοποθεσία;
- Αποδοτικότητα: Πόση εργασία θα πρέπει οι χρήστες να εκτελούν για να επωφεληθούν από την τοποθεσία; Σε ποιο βαθμό μπορεί η εφαρμογή να λειτουργεί αυτόματα για λογαριασμό των χρηστών;
- Διαφάνεια: κατανοούν οι χρήστες τι πληροφορίες σχετικά με τη τοποθεσία επεξεργάζεται η εφαρμογή για αυτούς; Πώς συλλέγει αυτές τις πληροφορίες και γιατί τις χρειάζεται;
- Προστασία Προσωπικών Δεδομένων: Σε ποιο βαθμό μπορούν οι χρήστες να ελέγχουν τη διανομή των προσωπικών πληροφοριών που αφορούν την τοποθεσία;

- Ικανοποίηση: Οι χρήστες βρίσκουν τη συνολική εμπειρία ικανοποιητική;

Από τον τρόπο που θα απαντηθούν αυτές οι ερωτήσεις καθορίζεται το LbUX που θα δημιουργηθεί. Όμως, η δημιουργία και η παράδοση ενός LbUX δεν είναι μια ενιαία πρόκληση, αλλά αφορά πολλούς τομείς σχεδίασης:

- Αρχιτεκτονική πληροφορίας (Information Architecture): Η αρχιτεκτονική για το πώς πρέπει μια εφαρμογή να αναπαραστήσει την πληροφορία που αφορά τη τοποθεσία (LIA).
- Σχεδιασμός υπηρεσίας (Service design): Σχεδιασμός για το πώς μια εφαρμογή αποκτά και διαχειρίζεται την τοποθεσία του χρήστη.
- Σχεδιασμός αλληλεπίδρασης (Interaction design): Σχεδιασμός για την επικοινωνία μεταξύ των εφαρμογών του χρήστη και των πηγών (εφαρμογών) παροχής της τοποθεσίας του.
- Σχεδιασμός διεπαφής (Interface design): Ο σχεδιασμός της διεπαφής θα πρέπει να υποστηρίζει και τα τρία προηγούμενα και να παρουσιάζει στο χρήστη την τοποθεσία που έχει αντιληφθεί η εφαρμογή και να του δώσει τη δυνατότητα να τη διορθώσει.

Συνεπώς, η διαδικασία σχεδιασμού και ανάπτυξης μιας διαδικτυακής πλατφόρμας υποδομής, βασισμένη στην τοποθεσία για τη σύντηξη δεδομένων απαιτεί μια πληθώρα συνεργασιών από τεχνολογίες και εφαρμογές. Πρέπει κανείς να ξεκινήσει από την βάση η οποία είναι η συλλογή των δεδομένων και να ακολουθήσει τα στάδια που προαναφέρθηκαν ώστε να φτάσει στην παρουσίαση τους στον τελικό χρήστη. Κάποια από τα στάδια αυτά μπορούν να παραληφθούν και κάποια άλλα μπορούν να αντικατασταθούν από άλλα παρόμοια. Όμως η κάθε παράληψη ή παραλλαγή των βημάτων αυτών έχει τις δικές της συνέπειες που κατά τη σχεδίαση της εφαρμογής πρέπει να ληφθούν υπόψη και να γίνουν οι αναγκαίες παραδοχές. Σε καμία όμως περίπτωση η παράλειψη ή η αντικατάσταση ενός εκ των σταδίων δεν είναι απαγορευτικό για την υλοποίηση της επιθυμητής εφαρμογής, καθώς υπάρχουν πολύ τρόποι για την επίτευξη του ίδιου σκοπού.

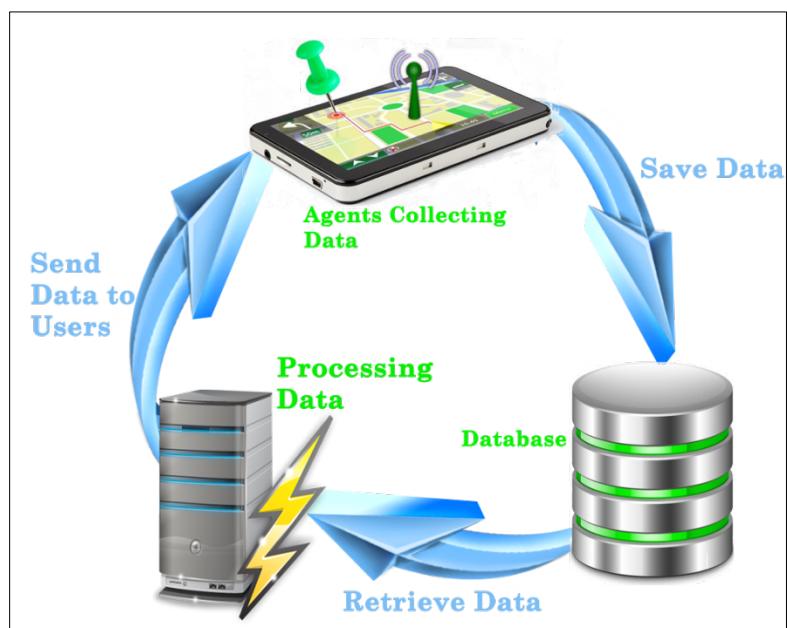
1.2 Πρόταση της Διπλωματικής

Στην παρούσα διπλωματική η πρόταση που παρουσιάζεται για την προσέγγιση και κατά το δυνατόν καλύτερη αντιμετώπιση των προβλημάτων που παρουσιάστηκαν στο κεφάλαιο 1.1, είναι η διαδικτυακή πλατφόρμα myWIFIspeed. Η πλατφόρμα αυτή δραστηριοποιείται στον τομέα της σύντηξης δεδομένων τα οποία αφορούν την απόδοση των ασύρματων δικτύων που χρησιμοποιούν οι κινητές συσκευές. Ειδικότερα, σε συνεργασία με την crowdsourcing εφαρμογή HuaServiceTester2 η οποία είναι συμβατή με κινητές συσκευές

που έχουν Android λειτουργικό σύστημα, συλλέγονται δεδομένα σχετικά με την απόδοση των ασύρματων δικτύων από φορείς παροχής όπως OTE, Forthnet, Wind κ.α καθώς και των δικτύων κινητής τηλεφωνίας Cosmote, Vodafone, Wind κ.α. Συγκεκριμένα με τον όρο απόδοση αναφερόμαστε στη ταχύτητα του εκάστοτε δικτύου κατά το κατέβασμα (downloading), ανέβασμα (uploading), ping στην google και ping στον εξυπηρετητή που δημιουργήθηκε για της ανάγκες της διπλωματικής. Στην συνέχεια η πλατφόρμα myWIFIspeed επεξεργάζεται τα δεδομένα αυτά και παρουσιάζει την παραγόμενη πληροφορία διαδικτυακά πλέον στον χρήστη. Κατά την διαδικασία της επεξεργασίας η πλατφόρμα κάνει χρήση, συνδυάζει και εκμεταλλεύεται την προσφορά και τα πλεονεκτήματα από τις τεχνολογίες και τεχνικές που αναφέρθηκαν στο κεφάλαιο 1.1.

Συνεπώς οι χρήστες έξυπνων κινητών συσκευών με λειτουργικό android μπορούν να εγκαταστήσουν την εφαρμογή HuaServiceTester2_και μέσω αυτής να λαμβάνουν μετρήσεις για το ασύρματο δίκτυο WIFI τους ή για το δίκτυο του φορέα τους με ελάχιστες έως καθόλου ενέργειες. Οι μετρήσεις αυτές, που όπως προαναφέρθηκε αφορούν τις ταχύτητες κατεβάσματος, ανεβάσματος και ping, αποθηκεύονται σε κατάλληλα διαμορφωμένη βάση δεδομένων. Στη συνέχεια η διαδικτυακή πλατφόρμα myWIFIspeed ανεβάζει αυτά τα δεδομένα σε έναν map server, συγκεκριμένα στο GeoServer (αναλυτικότερη παρουσίαση του στο κεφ. 3). Στο σημείο αυτό έχουν δημιουργηθεί τα κατάλληλα layers και styles για την σύντηξη των δεδομένων (αναλυτικότερα στο κεφ. 6). Μετά τη σύντηξη των δεδομένων, η πληροφορία μεταφέρεται μέσω HTTP κλήσεων στον διαδικτυακό τόπο και παρουσιάζεται στον χρήστη με την χρήση των OpenLayers maps (κεφ. 2). Στον διαδικτυακό τόπο οι χρήστες μπορούν να συνδεθούν είτε ως απλοί χρήστες για να δουν απλώς την παραγόμενη πληροφορία είτε ως διαχειριστές. Τέλος η πληροφορία από τον διαδικτυακό τόπο πίσω στην εφαρμογή HuaServiceTester2 γίνεται με την χρήση του GCM^[25] Server της google.

Συμπερασματικά, είδαμε με συνοπτικό τρόπο τον κύκλο ζωής των δεδομένων (Εικόνα 2) από την στιγμή της δημιουργίας τους, την επεξεργασία τους, την εμφάνιση τους στον χρήστη διαδικτυακά και τέλος την επιστροφή τους πίσω στην android εφαρμογή. Καθώς και τις βασικές υπάρχουσες τεχνολογίες που χρησιμοποιούνται κατά την διαδρομή. Όλα αυτά που ειπώθηκαν με περιγραφικό τρόπο σε αυτό το κεφάλαιο, παρουσιάζονται και αναλύονται στα επόμενα κεφάλαια.



Εικόνα 2: Κύκλος ζωής δεδομένων

2 Ανάπτυξη Διαδικτυακής Πλατφόρμας Βασισμένη στην Τοποθεσία

2.1 Ανάπτυξη Διαδικτυακής Πλατφόρμας & Τεχνολογίες

Για την δημιουργία μιας τέτοιας διαδικτυακής πλατφόρμας δεν μπορεί να γίνει χρήση μόνο μιας τεχνολογίας, αλλά απαιτείται συνδυασμός τεχνολογιών ανάπτυξης διαδικτύου. Οι τεχνολογίες αυτές έχουν αναπτυχθεί και διακρίνονται σε δύο βασικές κατηγορίες:

- Τεχνολογίες ανάπτυξης στην πλευρά του πελάτη (client side)
- Τεχνολογίες ανάπτυξης στον εξυπηρετητή (server side)

Οι client side τεχνολογίες δημιουργούνται και τρέχουν στη συσκευή του χρήστη με βασικότερο εκφραστή τον φυλλομετρητή (browser) και έχουν ως στόχο την υποστήριξη της εμπειρίας χρήσης. Δηλαδή οι client side τεχνολογίες στοχεύουν στη δημιουργία δυναμικών και αλληλεπιδραστικών εφαρμογών διαδικτύου ώστε ο χρήστης να μπορεί να τις παραμετροποιήσει κατά πως επιθυμεί ή να κάνει τις λειτουργίες που θέλει με απλό και ουσιαστικό τρόπο. Από την άλλη πλευρά, οι server side τεχνολογίες απευθύνονται στον προγραμματισμό διαδικτυακών εφαρμογών, αλλά στην πλευρά του server. Αυτό σημαίνει πως στοχεύουν στην υλοποίηση δύσκολων λειτουργιών, που δεν μπορούν να δημιουργηθούν ή να τρέξουν στον πελάτη όπως η σύνδεση με βάση δεδομένων για εγγραφή, ανάγνωση ή ενημέρωση, η επικοινωνία με άλλη διαδικτυακή εφαρμογή για λήψη ή αποστολή δεδομένων κ.α.

Συνεπώς ο συνδυασμός των client side και server side τεχνολογιών είναι αυτός που μας προσφέρει τα εργαλεία και καθιστά δυνατή την ανάπτυξη μιας διαδικτυακής πλατφόρμας. Για τον λόγο αυτό στα δύο επόμενο κεφάλαια αναφέρονται διαθέσιμες τεχνολογίες client side καθώς και server side. Πρέπει να σημειωθεί πως δεν αναφέρονται όλες οι υπάρχουσες τεχνολογίες πράγμα το οποίο θα ήταν αδύνατον σε μια διπλωματική εργασία. Αναφέρονται όμως αυτές που χρησιμοποιήθηκαν ή στις οποίες βασιστήκαμε για την ανάπτυξη της δικής μας διαδικτυακής πλατφόρμας

2.1.1 Client Side Τεχνολογίες

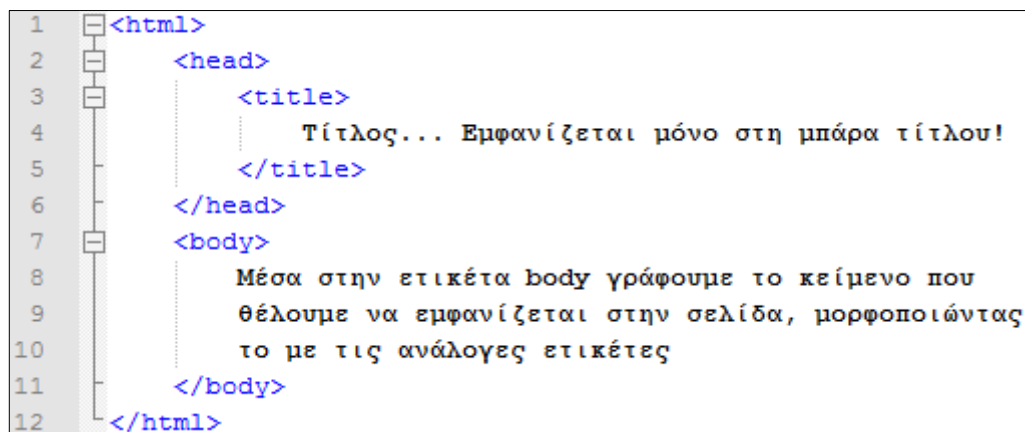
HTML

Η ονομασία προέρχεται από τα αρχικά των Αγγλικών λέξεων Hyper Text Markup Language^{[26][27]} ή στα ελληνικά Γλώσσα Σήμανσης Υπερκειμένου. Έχουν δημοσιευτεί πολλές εκδόσεις-πρότυπα της HTML από το 1990, άλλα από το 2008 μέχρι και σήμερα είμαστε στο πρότυπο HTML5. Η χρήση, στον παγκόσμιο ιστό, μιας γλώσσας σήμανσης σημαίνει ότι γράφεται πρώτα το κείμενο και έπειτα προσθέτονται ειδικά σύμβολα γύρω από τις λέξεις ή από ολόκληρες προτάσεις ώστε να καθοριστεί ο τρόπος εμφάνισης των εγγράφων στον φυλλομετρητή. Με τον όρο έγγραφα εννοούμε κάθε είδους αρχείου, όπως αρχείο κειμένου, εικόνες, animation και βίντεο. Τα αρχεία της HTML είναι απλά αρχεία κειμένου στα οποία υπάρχουν ειδικά σύμβολα που λέγονται ετικέτες (tags). Οι ετικέτες ελέγχουν την δομή και την μορφή του περιεχόμενου της ιστοσελίδας. Επίσης παρέχουν πληροφορίες στους φυλλομετρητές για την σελίδα που πρόκειται να εμφανίσουν, όπως ο τίτλος της σελίδας ή ο συγγραφέας της, κ.α.

Η γενική δομή μιας ετικέτας είναι ως εξής: <όνομα_ετικέτας> κείμενο </όνομα_ετικέτας>. Οι περισσότερες HTML ετικέτες αποτελούνται από μια ετικέτα αρχής και μια ετικέτα τέλους και ανάμεσα σε αυτές υπάρχει το κείμενο που χαρακτηρίζεται από τις ετικέτες αυτές. Η ετικέτα τέλους περιέχει τον χαρακτήρα / πριν το όνομα της ετικέτας. Με την ετικέτα αρχής δηλώνεται στον φυλλομετρητή η αρχή της ενέργειας που θα εκτελεστεί στο κείμενο. Για να δηλώσουμε το σημείο που θα σταματήσει να εκτελείται η ενέργεια αυτή χρησιμοποιούμε την ετικέτα τέλους. Υπάρχουν περιπτώσεις ετικετών HTML που δεν έχουν ετικέτα κλεισίματος, π.χ. όταν θέλουμε να εισάγουμε μία εικόνα απλά δηλώνουμε την ενέργεια αυτή με την αντίστοιχη HTML ετικέτα . Στο όνομα ετικέτας δε γίνεται διάκριση ανάμεσα σε πεζούς και κεφαλαίους χαρακτήρες. Οποιοσδήποτε συνδυασμός είναι αποδεκτός Π.χ. οι ετικέτες <html>, <HTML> ή <HtMl> είναι ισοδύναμες.

Ένα αρχείο HTML περιέχεται μέσα στις ετικέτες αρχής και τέλους <HTML>...</HTML> και αποτελείται από δύο ξεχωριστά μέρη: το <HEAD>...</HEAD> στο οποίο περιλαμβάνονται πληροφορίες που αφορούν το ίδιο το αρχείο και όχι το τι θα εμφανιστεί στον φυλλομετρητή, και το <BODY>...</BODY> στο οποίο περιλαμβάνεται ό,τι θα εμφανιστεί. Πριν δείξουμε τα παραπάνω με ένα παράδειγμα, θα αναφερθούμε σε μία άλλη ετικέτα, την <TITLE>...</TITLE>. Η ετικέτα αυτή εμφανίζει τον τίτλο στην μπάρα τίτλου του

φυλλομετρητή. Η ετικέτα αυτή δεν εμφανίζει κάποιο κείμενο στην οθόνη και βρίσκεται μέσα στην ετικέτα <HEAD>...<HEAD>. Όπως αναφέραμε και παραπάνω ένα αρχείο HTML είναι ένα αρχείο κειμένου (ASCII). Επομένως, για να δημιουργήσουμε ένα αρχείο HTML αρκεί να ανοίξουμε έναν επεξεργαστή κειμένου, να πληκτρολογήσουμε το κείμενό μας, να το αποθηκεύσουμε σε ένα αρχείο με κατάληξη .htm ή .html και στη συνέχεια να το εμφανίσουμε κάνοντας χρήση ενός φυλλομετρητή. Στην Εικόνα 3: Δομή αρχείου HTML παρουσιάζεται στην πράξη η δομή ενός αρχείου HTML και στον Πίνακα 1 φαίνονται οι πιο συνηθισμένες ετικέτες.



Εικόνα 3: Δομή αρχείου HTML

Πίνακας 1: Συνηθέστερες ετικέτες στην HTML

ΕΤΙΚΕΤΑ	ΠΕΡΙΓΡΑΦΗ	ΕΤΙΚΕΤΑ	ΠΕΡΙΓΡΑΦΗ
<html>	Το αρχικό tag για κάθε σελίδα HTML	<u>	Υπογράμμιση χαρακτήρων
<head>	Η επικεφαλίδα της σελίδας	<strike>	Διακριτή διαγραφή (strike)
<meta>	Στοιχεία για τις μηχανές αναζήτησης	<blockquote>	Αριστερό περιθώριο σε μία παράγραφο
<title>	Ο τίτλος της σελίδας που θα εμφανιστεί στο Browser.	<sup>	Εκθέτης
<body>	Το κύριο σώμα της σελίδας	<sub>	Δείκτης
<!-->	Σχόλιο	<small>	Μικρά γράμματα
 	Αλλαγή γραμμής (δεν έχει tag τέλους)	<big>	Μεγάλα γράμματα
<p>	Αλλαγή παραγράφου		Μορφοποίηση γραμματοσειράς
<h1>	Επικεφαλίδα (από 1 έως 6)	<hr>	Οριζόντια γραμμή
	Δημιουργία έντονων χαρακτήρων (bold)	<a>	Σύνδεσμος (link)
<i>	Δημιουργία πλάγιων χαρακτήρων		Εισαγωγή εικόνας

XML

Extensible Markup Language^[28] (XML) είναι γλώσσα σήμανσης, που περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση κειμένων. Η XML σχεδιάστηκε δίνοντας έμφαση στην απλότητα, τη γενικότητα και τη χρησιμότητα στο διαδίκτυο. Είναι μία μορφοποίηση δεδομένων κειμένου, με ισχυρή υποστήριξη Unicode για όλες τις γλώσσες. Αν και η σχεδίαση της XML εστιάζει στα κείμενα, χρησιμοποιείται ευρέως για την αναπαράσταση αυθαίρετων δομών δεδομένων, που προκύπτουν για παράδειγμα στις υπηρεσίες ιστού. Για αυτόν το λόγο η XML χρησιμοποιείται κυρίως για την δόμηση, αποθήκευση και την μεταφορά δεδομένων μεταξύ των ιστοσελίδων στον ιστό.

Η σύνταξη της XML είναι παρόμοια με την σύνταξη της HTML, αλλά υπάρχουν μεταξύ τους δύο μεγάλες και σημαντικές διαφορές που καθορίζουν τον διαφορετικό τομέα χρήσης τους. Η πρώτη είναι στη λειτουργία τους, η XML όπως προείπαμε απλά αποθηκεύει δεδομένα εστιάζοντας στο είδος τους, αλλά για να διαβαστούν τα δεδομένα, να σταλούν μεταξύ ιστοσελίδων ή να παρουσιαστούν, κάποιος θα πρέπει να γράψει το κατάλληλο λογισμικό που να εκτελεί αυτές τις λειτουργίες. Ενώ η HTML παρουσιάζει τα δεδομένα εστιάζοντας στην μορφοποίηση τους χωρίς ανάγκη επιπρόσθετου λογισμικού. Δεύτερον η XML, σε αντίθεση με την HTML, δεν έχει προκαθορισμένες ετικέτες (tags) που να ορίζονται από κάποιο XML πρότυπο, αλλά την ώρα που γίνεται η συγγραφή του αρχείου ο συγγραφέας μπορεί να δημιουργήσει δικές του ετικέτες όπως στην Εικόνα 4.

```
1      <?xml version="1.0" encoding="ISO-8859-1"?>
2      <note>
3          <to>Tove</to>
4          <from>Jani</from>
5          <heading>Reminder</heading>
6          <body>Don't forget me this weekend!</body>
7      </note>
```

Εικόνα 4: Δομή XML αρχείου. Όλα τα tags είναι δημιουργία του συγγραφέα

Συνεπώς από τα προηγούμενα βλέπουμε πως με την χρήση των XML αρχείων μπορούμε να έχουμε πλεονεκτήματα^[29] όπως:

- Διαχωρισμός δεδομένων από την HTML: Εάν θέλουμε να εμφανίζονται δυναμικά δεδομένα σε μια σελίδα HTML, θα χρειαστεί πολλή δουλειά για να επεξεργαστούμε τον κώδικα HTML κάθε φορά που αλλάζουν τα δεδομένα. Ενώ με το XML, τα δεδομένα μπορούν να αποθηκεύονται σε ξεχωριστά αρχεία XML. Με αυτό τον τρόπο μπορούμε να επικεντρωθούμε στην ανάπτυξη HTML / CSS για

την εμφάνιση και τη διάταξη, και να είμαστε σίγουροι ότι οι αλλαγές στα δεδομένα δεν θα απαιτήσουν αλλαγές στην HTML. Στη συνέχεια με λίγες γραμμές κώδικα JavaScript, μπορούμε να διαβάσουμε το εξωτερικό αρχείο XML και να ενημερώσουμε το περιεχόμενο των δεδομένων της σελίδας.

- Απλοποίηση κοινόχρηστης χρήσης δεδομένων: Στον πραγματικό κόσμο, τα συστήματα ηλεκτρονικών υπολογιστών και βάσεις δεδομένων περιέχουν δεδομένα σε ασύμβατες μορφές. Ενώ τα XML δεδομένα αποθηκεύονται σε μορφή απλού κειμένου. Αυτό παρέχει ένα τρόπο αποθήκευσης δεδομένων ανεξάρτητου λογισμικού και υλικού. Καθιστώντας έτσι πολύ πιο εύκολη την δημιουργία δεδομένων που μπορούν να χρησιμοποιηθούν από διαφορετικές εφαρμογές.
- Απλοποίηση διακίνησης δεδομένων: Μια από τις πιο χρονοβόρες προκλήσεις για τους προγραμματιστές είναι η ανταλλαγή δεδομένων μεταξύ ασύμβατων συστημάτων μέσω του Διαδικτύου. Όμως η ανταλλαγή δεδομένων ως XML αρχεία μειώνει σημαντικά την πολυπλοκότητα αυτή, δεδομένου ότι τα δεδομένα μπορούν να διαβαστούν από διαφορετικές ασύμβατες εφαρμογές.
- Εύκολη διαχείριση στις αλλαγές πλατφόρμας: Η αναβάθμιση σε νέα συστήματα (hardware ή software), είναι πάντα χρονοβόρα. Μεγάλες ποσότητες δεδομένων πρέπει να μετατραπούν, και συνήθως δεδομένα μη συμβατά χάνονται. Τα δεδομένα όμως που είναι αποθηκευμένα σε XML αρχεία είναι σε μορφή απλού κειμένου και αυτό καθιστά ευκολότερη την επέκταση ή την αναβάθμιση του συστήματος μας χωρίς την απώλεια δεδομένων.
- Διαθεσιμότητα δεδομένων: Διαφορετικές εφαρμογές μπορούν να έχουν πρόσβαση στα δεδομένα, όχι μόνο σε σελίδες HTML, αλλά επίσης και από τις XML πηγές δεδομένων. Με την XML, τα δεδομένα μπορούν να είναι διαθέσιμα σε όλα τα είδη των "μηχανών ανάγνωσης".

CSS

Τα Cascading Style Sheets^[30], αλλιώς στα ελληνικά Διαδοχικά Φύλλα Στυλ ή αλλιώς λουχία φύλλων στυλ, είναι μια γλώσσα φύλλων στυλ που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Συγκεκριμένα με το CSS ελέγχουμε την εμφάνιση ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML, δηλαδή μπορούμε να πούμε ότι ελέγχουμε την εμφάνιση ενός ιστότοπου. Τα CSS τα χρειαζόμαστε γιατί γράφοντας τις σελίδες μας μόνο με HTML κώδικα, μπορούμε να ορίσουμε το χρώμα και το μέγεθος του κειμένου αλλά και άλλων

στοιχείων της σελίδας (όπως πίνακες, συνδέσμους, λίστες κτλ). Για να αλλάξουμε όμως το χρώμα κάποιου κειμένου ή το χρώμα ενός πίνακα, θα πρέπει να βρούμε το χρώμα αυτό μέσα στον κώδικα και να το αλλάξουμε. Η διαδικασία αυτή μπορεί να φαντάζει εύκολη όταν έχουμε να διαχειριστούμε μια μόνο σελίδα, αλλά ένας ιστότοπος αποτελείται από δεκάδες σελίδες τις οποίες χρειάζεται να διαχειριζόμαστε εύκολα και γρήγορα. Με την χρήση CSS μπορούμε να ορίζουμε χρώματα και μεγέθη οργανωμένα σε στυλ και έπειτα να εφαρμόζουμε τα στυλ αυτά στα στοιχεία (tags) των σελίδων της εφαρμογής μας. Με αυτόν τον τρόπο, κάθε φορά που αλλάζουμε το χρώμα ενός στυλ, αλλάζει το χρώμα όλων των στοιχείων που έχουν αναφορά στο στυλ αυτό. Έτσι αν έχουμε ορίσει ένα στυλ για το κύριο μενού της εφαρμογής, τότε θα χρειάζεται να αλλάξουμε το χρώμα του στυλ αυτού και αυτόματα θα εφαρμοστεί σε όλες τις σελίδες. Εκτός από την ευκολία στην διαχείριση ενός ιστότοπου, ένα άλλο σημαντικό πλεονέκτημα της χρήσης CSS στις σελίδες είναι ο "καθαρότερος" κώδικας, χωρίς πολλές ιδιότητες στις ετικέτες οι οποίες τον κάνουν δυσανάγνωστο. Επιπλέον κάνει γρηγορότερη την πλοήγηση καθώς το αρχείο, μέσα στο οποίο ορίζονται τα στυλ, "διαβάζεται" από τον φυλλομετρητή μόνο μια φορά και έπειτα αποθηκεύεται στην προσωρινή μνήμη (cache memory), μειώνοντας έτσι το μέγεθος της πληροφορίας που "κατεβαίνει" (download) στους φυλλομετρητές. Έτσι ήθελες έλα

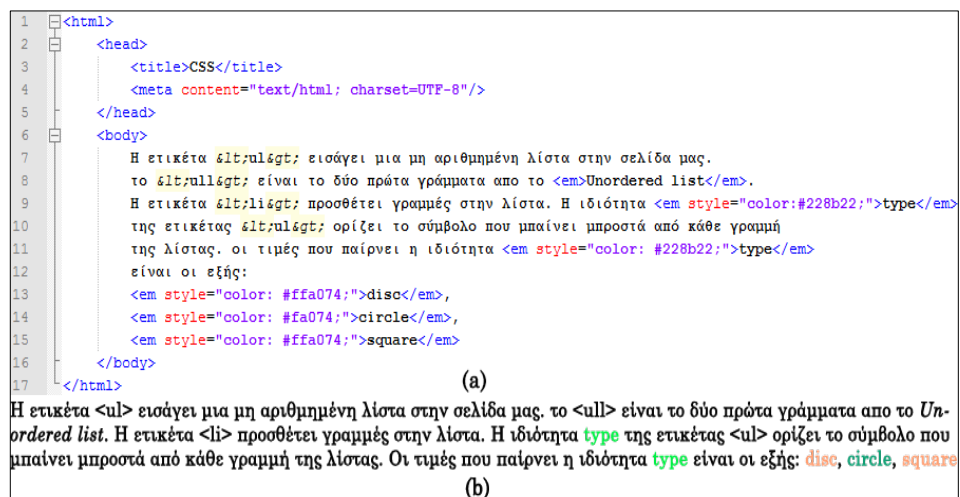
Τα CSS σε μια σελίδα μπορούν να οριστούν με 3 διαφορετικούς τρόπους έχοντας το ίδιο αποτέλεσμα.

1. **ορισμός των στυλ μέσα στην σελίδα (Internal Style Sheet)**^[31]: στον τρόπο αυτό ορίζουμε το στυλ στην επικεφαλίδα (HEAD) της σελίδας χρησιμοποιώντας την HTML ετικέτα <style> όπου μέσα γράφουμε τα στυλ που θα εφαρμοστούν στην σελίδα, όπως ο παρακάτω κώδικας στην Εικόνα 5.

```
1 <html>
2   <head>
3     <title>CSS</title>
4     <meta content="text/html; charset=UTF-8"/>
5     <style type="text/css">
6       body{
7         background-color: blue;
8       }
9     </style>
10  </head>
11  <body>
12    Κείμενο...
13  </body>
14 </html>
```

Εικόνα 5: Internal style sheet

2. Ορισμός στυλ μέσα στις ετικέτες html^[31]: Μπορούμε να ορίσουμε στυλ στις ετικέτες που επιθυμούμε, χρησιμοποιώντας την κοινή ιδιότητα style της HTML. Η ιδιότητα style μπορεί να μπει σχεδόν σε όλες τις HTML ετικέτες της ενότητας <body>. Ένα παράδειγμα είναι ο κώδικας στην Εικόνα 6. Με αυτόν τον τρόπο όμως χάνουμε το μεγάλο πλεονέκτημα των CSS, δηλαδή αυτό της εύκολης διαχείρισης των στυλ των σελίδων μας, καθώς στην περίπτωση αυτή το στυλ εφαρμόζεται μόνο στην ετικέτα στην οποία ορίζεται και πουθενά αλλού. Δεν συνιστάται η συχνή χρήση του τρόπου αυτού, ιδιαίτερα σε περιπτώσεις που το στυλ θα χρησιμοποιηθεί και σε κάποιο άλλο σημείο του ιστότοπου.
3. Ορισμός των στυλ σε εξωτερικό αρχείο (External Style Sheet)^[31]: στον τρόπο αυτό δημιουργούμε ένα εξωτερικό αρχείο στυλ με επέκταση *.css στο οποίο γράφουμε τα στυλ που θέλουμε το ένα κάτω από το άλλο. Η σύνδεση του εξωτερικού αρχείου στυλ και της σελίδας HTML γίνεται με την χρήση της HTML ετικέτας <link> στο τμήμα HEAD της HTML σελίδας, όπως στον παρακάτω κώδικα. Η ετικέτα <link> εισάγει στην σελίδα τα στυλ που βρίσκονται στο εξωτερικό αρχείο *.css. Παράδειγμα εισαγωγής εξωτερικού αρχείου CSS σε μια σελίδα φαίνεται στην Εικόνα 7. Έχοντας ορίσει τα στυλ σε ένα εξωτερικό αρχείο *.css μπορούμε να τα εισάγουμε μέσα σε όσες σελίδες html επιθυμούμε, δημιουργώντας έτσι ένα ενιαίο στυλ για τις σελίδες αυτές.



Εικόνα 6: (a) στυλ μέσα στις ετικέτες, (b) αποτέλεσμα του στυλ

```

1 <html>
2 <head>
3   <title>CSS</title>
4   <meta content="text/html; charset=UTF-8"/>
5   <link type="text/css" rel="stylesheet" href="/mystyle.css">
6 </head>
7 <body>
8   Κείμενο...
9 </body>
10 </html>

```

Εικόνα 7: Εισαγωγή εξωτερικού αρχείου css σε σελίδα html

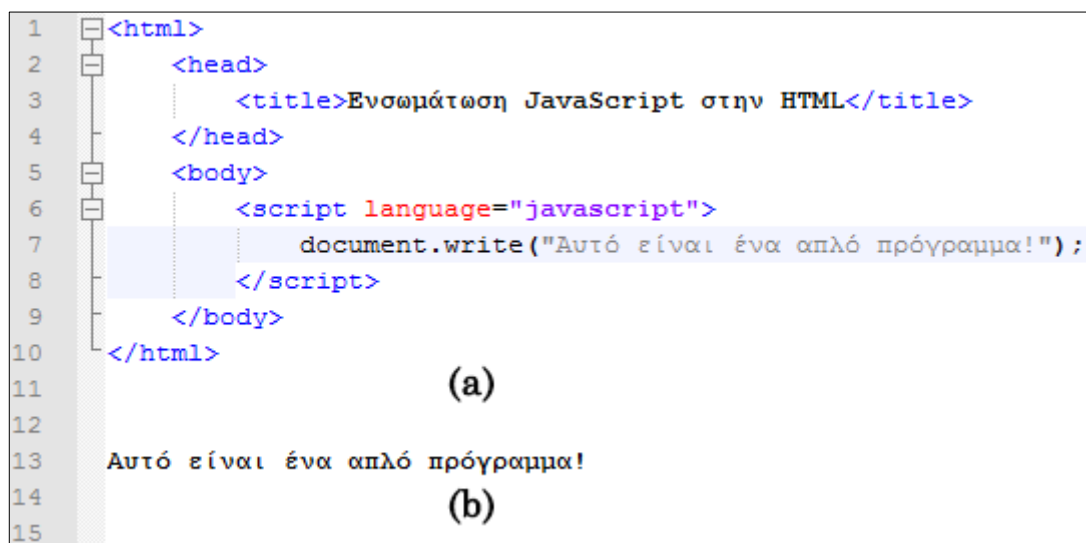
Από τους τρεις τρόπους που παρουσιάστηκαν για την εισαγωγή CSS σε μια HTML σελίδα, ο προτιμότερος τρόπος είναι η εισαγωγή από εξωτερικό αρχείο CSS. Ο τρόπος αυτός σε σχέση με τους υπόλοιπους μας προσφέρει τα πλεονεκτήματα ότι: τα στυλ γράφονται σε εξωτερικό αρχείο το οποίο μπορούμε να εισάγουμε σε όσες σελίδες επιθυμούμε δημιουργώντας έτσι ένα ενιαίο στυλ για όλες τις σελίδες της εφαρμογής μας. Επιπλέον ο πηγαίος κώδικας HTML της σελίδας είναι "καθαρότερος" γιατί δεν περιέχει τα στυλ στο τμήμα head, όπως στον πρώτο τρόπο, αλλά ούτε περιέχει τα στυλ μέσα στις ετικέτες, όπως στον δεύτερο τρόπο. Ο "καθαρότερος" κώδικας είναι πιο εύκολα διαχειρίσιμος, αλλά κυρίως "κατεβαίνει" γρηγορότερα στον φυλλομετρητή του επισκέπτη, λόγω του μικρότερου σε μέγεθος περιεχομένου. Τέλος το εξωτερικό αρχείο που περιέχει τα στυλ, αποθηκεύεται στην προσωρινή μνήμη του φυλλομετρητή του επισκέπτη της σελίδας όταν αυτός ανοίγει την κεντρική σελίδα του ιστότοπου. Έτσι όποια σελίδα και να ανοίξει στην συνέχεια, ο φυλλομετρητής θα διαβάσει τα στυλ από την προσωρινή μνήμη του, μειώνοντας έτσι τον όγκο δεδομένων που κατεβάζει από τον εξυπηρετητή που φιλοξενεί την σελίδα.

JavaScript

Ο σχεδιασμός του ιστού δεν αφορά μόνο το περιεχόμενο και την παρουσίαση. Οι χρήστες εκτός από αυτά περιμένουν και διαδραστικές εφαρμογές, που θα ανταποκρίνονται στις επιλογές τους. Αυτή την δουλειά έρχεται να την κάνει η JavaScript. Η JavaScript είναι μια γλώσσα προγραμματισμού από την πλευρά του πελάτη, η οποία δημιουργήθηκε από την εταιρεία Netscape και χρησιμοποιείται ευρύτατα για να προσθέσει δυναμικά εφέ και διαδραστικότητα στις ιστοσελίδες. Η Javascript υποστηρίζεται από τους πιο γνωστούς φυλλομετρητές όπως τον Netscape Navigator, τον Internet Explorer, το Mozilla Firefox, Google Chrome, Safari, Opera καθώς και από άλλους λιγότερο γνωστούς. Η Javascript χαρακτηρίζεται και ως γλώσσα συγγραφής σεναρίων (scripting

language). Τα σενάρια (scripts) που γράφονται σε JavaScript, είναι μικρά προγράμματα που αλληλεπιδρούν με τον φυλλομετρητή και τον κώδικα HTML μιας σελίδας. Ο κώδικας της JavaScript γράφεται σε καθαρό κείμενο (ASCII μορφή) και ενσωματώνεται μέσα στον κώδικα της HTML. Μπορεί να εκτελεστεί αμέσως με το φόρτωμα της σελίδας ή όταν λαμβάνει χώρα ένα συμβάν (event), όπως η πίεση ενός πλήκτρου του ποντικιού ή η τοποθέτηση του ποντικιού πάνω σε ένα αντικείμενο. Οι εφαρμογές της JavaScript στο δυναμικό προγραμματισμό παγκόσμιου ιστού είναι πάρα πολλές. Ως χαρακτηριστικά παραδείγματα αναφέρουμε τον έλεγχο της εγκυρότητας των δεδομένων που πληκτρολογούνται σε μια φόρμα του πελάτη πριν αυτά αποσταλούν στον εξυπηρετητή, τη διεξαγωγή διάφορων υπολογισμών και μετατροπών μεγεθών και την προσθήκη δυναμικών μενού στις ιστοσελίδες.

Το σενάριο JavaScript ενσωματώνεται στην ιστοσελίδα, ανάμεσα στις ετικέτες <SCRIPT> και </SCRIPT>. Μέσα σε ένα αρχείο HTML μπορούν να υλοποιηθούν πολλά σενάρια JavaScript, χρησιμοποιώντας πολλαπλές ετικέτες SCRIPT. Ένα παράδειγμα κομματιού κώδικα, μέσα σε μια HTML σελίδα, είναι η Εικόνα 8.



Εικόνα 8: (a) κώδικας JavaScript στην html, (b) αποτέλεσμα του κώδικα

Τα σενάρια JavaScript δεν είναι υποχρεωτικά ενσωματωμένα στη σελίδα HTML. Αντίθετα, μπορούν να γραφτούν ως ξεχωριστά αρχεία με κατάληξη *.js, τα οποία καλούνται και εκτελούνται από τη σελίδα HTML με τη βοήθεια σχετικής αναφοράς. Σε αυτή τη περίπτωση, η ετικέτα SCRIPT συνοδεύεται από την παράμετρο SRC, η οποία χρησιμοποιείται για τη δήλωση του εξωτερικού αρχείου στον κώδικα HTML, όπως φαίνεται στο παρακάτω τμήμα κώδικα στην Εικόνα 9, στο οποίο το αρχείο global.js δηλώνεται ως εξωτερικό σενάριο JavaScript.

```

1 <html>
2   <head>
3     <title> Εισαγωγή εξωτερικού αρχείου JavaScript </title>
4     <script language="JavaScript" src="global.js"></script>
5   </head>
6   <body>
7     Κείμενο...
8   </body>
9 </html>

```

Εικόνα 9: Εισαγωγή εξωτερικού αρχείου JavaScript στην html

Η JavaScript βασίζεται στην έννοια του αντικειμένου. Το αντικείμενο είναι μια οντότητα η οποία κατέχει ένα σύνολο από ιδιότητες, οι οποίες μπορούν να τροποποιηθούν με κατάλληλες μεθόδους. Ένα καλό παράδειγμα αντικειμένου είναι το αντικείμενο document (η επιφάνεια του φυλλομετρητή), το οποίο έχει αρκετές ιδιότητες, όπως το χρώμα παρασκήνιου, το χρώμα κειμένου και τον τίτλο. Οι περισσότερες ιδιότητες της Javascript σχετίζονται άμεσα ή έμμεσα με αλλαγές μιας ή περισσότερων ιδιοτήτων των αντικειμένων. Η σύνταξη αναφοράς σε κάποια ιδιότητα είναι η εξής: <όνομα αντικειμένου>.<ιδιότητα>. Για παράδειγμα, το document.bgcolor είναι το χρώμα παρασκήνιου για το αντικείμενο εγγράφου (document). Όπως προαναφέραμε, η έννοια του αντικειμένου είναι στενά συνδεδεμένη και με την έννοια της μεθόδου. Οι μέθοδοι είναι συναρτήσεις ή διαδικασίες που χρησιμοποιούνται για να εκτελούν μια λειτουργία σ' ένα αντικείμενο. Η κλήση και εκτέλεση των μεθόδων των αντικειμένων γίνεται με τη δήλωση αντικείμενο.μέθοδος(). Χαρακτηριστικό παράδειγμα κλήσης μεθόδου είναι η εντολή document.write("Αυτό είναι ένα απλό πρόγραμμα!").

Τέλος το σύνολο των προκαθορισμένων αντικειμένων του φυλλομετρητή είναι γνωστό με το όνομα DOM (Document Object Model). Τα DOM είναι ένα σύνολο συμβάσεων για το χειρισμό, την πλοήγηση και την επεξεργασία XML – HTML αρχείων χρησιμοποιώντας ένα σύνολο από API που μπορούν να εφαρμοστούν σε πολλές γλώσσες. Με τα DOM μπορούμε να κατεβάσουμε τη δομή ενός αρχείου XHTML και να κάνουμε αλλαγές και προσθήκες δυναμικά χωρίς να ανανεώσουμε την σελίδα.

AJAX

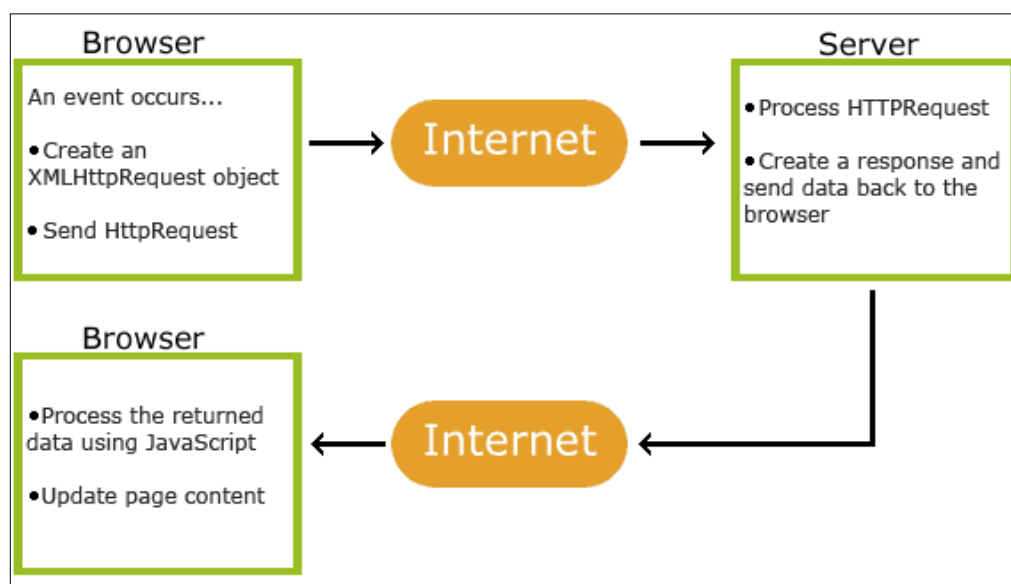
Η AJAX (Asynchronous JavaScript and XML^{[32][33]}) δεν είναι μια τεχνολογία^{[34][34]}, αλλά είναι ένας όρος που αντιπροσωπεύει μια ομάδα από τεχνολογίες διαδικτύου. Οι τεχνολογίες αυτές μπορούν να χρησιμοποιηθούν για την υλοποίηση διαδικτυακών εφαρμογών που επικοινωνούν με τον εξυπηρετητή στο παρασκήνιο, χωρίς να γίνεται παρεμβολή στην τρέχουσα κατάσταση της σελίδας. Ο όρος AJAX αναφέρθηκε πρώτη

φορά στο άρθρο^[34], στο οποίο ο Jesse James Garrett εξηγεί ότι χρησιμοποιούνται οι εξής τεχνολογίες:

- HTML (ή XHTML) και CSS για την αναπαράσταση
- Το Document Object Model (DOM) για δυναμική αναπαράσταση και αλληλεπίδραση με τα δεδομένα.
- Η XML για την ανταλλαγή των δεδομένων, και η XSLT για τον χειρισμό τους.
- Το XMLHttpRequest αντικείμενο για την ασύγχρονη επικοινωνία.
- Η JavaScript για να ενώσει όλες αυτές τις τεχνολογίες.

Ωστόσο υπήρξαν ορισμένες εξελίξεις στις τεχνολογίες που χρησιμοποιούνται σε μια εφαρμογή AJAX και στον ορισμό του όρου. Για παράδειγμα η χρήση της XML δεν είναι απαραίτητη για την ανταλλαγή των δεδομένων, στην θέση του μπορεί να χρησιμοποιηθεί η JSON (JavaScript Object Notation).

Τι μας προσφέρει λοιπόν ο συνδυασμός όλων αυτών των τεχνολογιών υπό το όνομα AJAX; Η AJAX επιτρέπει στις διαδικτυακές σελίδες να ανανεώνονται ασύγχρονα ανταλλάσσοντας μικρό όγκο δεδομένων με τον εξυπηρετητή στο παρασκήνιο. Αυτό σημαίνει πως είναι δυνατόν να ανανεώνονται μέρη των σελίδων, χωρίς να φορτώνεται ξανά όλη η σελίδα. Πριν από την AJAX, όταν γινόταν μια αλλαγή στο περιεχόμενο της σελίδας, ο περιηγητής φόρτωνε από την αρχή όλα τα δεδομένα. Ενώ τώρα με την χρήση της AJAX φορτώνει μόνο το κομμάτι της σελίδας που γίνεται αλλαγή του περιεχομένου. Στην Εικόνα 10 φαίνεται το μπλοκ διάγραμμα^[33] που δείχνει την αρχή λειτουργίας της AJAX.



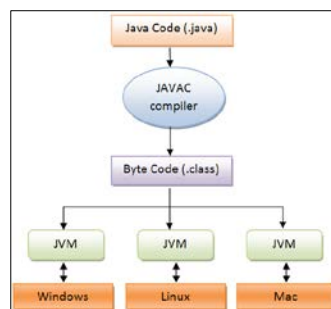
Εικόνα 10: Μπλοκ διάγραμμα λειτουργίας της AJAX (πηγή: w3shcools.com)

2.1.2 Server Side Τεχνολογίες

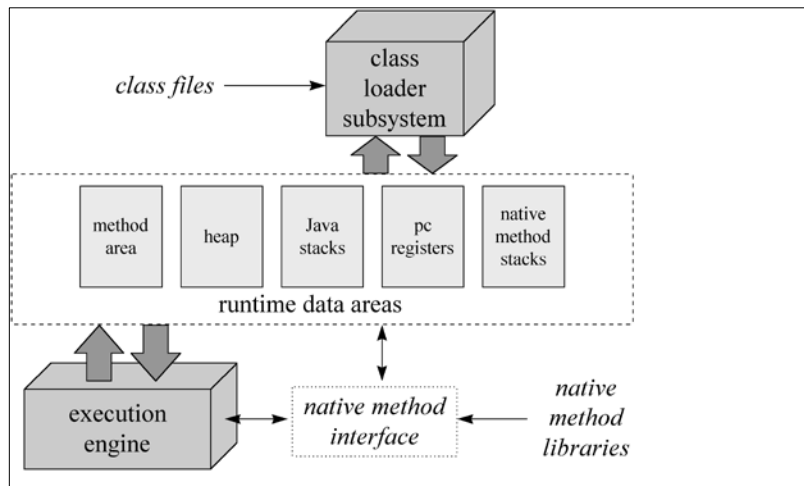
Java

Σύμφωνα με το εγχειρίδιο «The Java Language Specification»^[35] της Oracle η Java αποτελεί μια από τις ευρέως χρησιμοποιούμενες γλώσσες προγραμματισμού υψηλού επιπέδου η οποία βασίζεται στον αντικειμενοστραφή προγραμματισμό και στις κλάσεις, και είναι γλώσσα γενικού σκοπού. Η Java σχεδιάστηκε από τους James Gosling, Mike Sheridan, Patrick Naughton και η πρώτη έκδοση της έγινε το 1996 από την Oracle. Από τότε έχουν ακολουθήσει πολλές εκδόσεις με σημαντικότερες τις Java SE (από το 2014 βρισκόμαστε στη Java SE 8) και Java EE (από το 2013 βρισκόμαστε στη Java EE 7).

Η Java σχεδιάστηκε με σκοπό να προσφέρει ανεξαρτησία από το υλικό και το λειτουργικό σύστημα, δηλαδή να παρέχει την δυνατότητα στους προγραμματιστές εφαρμογών να γράψουν τον πηγαίο κώδικα της εφαρμογής μόνο μια φορά και να τρέχει παντού^[36]. Πρακτικά αυτό σημαίνει πως ο πηγαίος κώδικας μπορεί να μεταγλωττιστεί μόνο μια φορά και να εκτελείται σε όλα τα λειτουργικά συστήματα ανεξαρτήτως υλικού χωρίς να υπάρχει η ανάγκη να μεταγλωττίζεται ξεχωριστά για το κάθε ένα. Η ανεξαρτησία αυτή επιτυγχάνεται μεταγλωττίζοντας των πηγαίο κώδικα σε μια ενδιάμεση αναπαράσταση, η οποία ονομάζεται Java bytecode, αντί να μεταγλωττιστεί απευθείας σε γλώσσα μηχανής εξαρτημένη από την αρχιτεκτονική (Εικόνα 11). Ο Java bytecode (ενδιάμεσος κώδικας) είναι ισοδύναμος με την γλώσσα μηχανής, με βασικότερη διαφορά πως εκτελείται από την εικονική μηχανή της Java (virtual machine - JVM) η οποία είναι ειδικά γραμμένη για την αρχιτεκτονική του μηχανήματος που τρέχει τον κώδικα (Εικόνα 12). Συνεπώς οι χρήστες για να τρέξουν εφαρμογές java θα πρέπει απλώς να εγκαταστήσουν στο σύστημά τους ή στον φυλλομετρητή τους το Java Runtime Environment (JRE). Το JRE είναι η διανομή της JVM και κάποιων βιβλιοθηκών που παρέχει η Oracle στους χρήστες ώστε να μπορούν να εκτελούν τις εφαρμογές που είναι γραμμένες σε Java.



Εικόνα 11: Αρχιτεκτονική εκτέλεσης κώδικα της JAVA (πηγή: wikipedia.com)



Εικόνα 12: Αρχιτεκτονική JVM (πηγή: wikipedia.com)

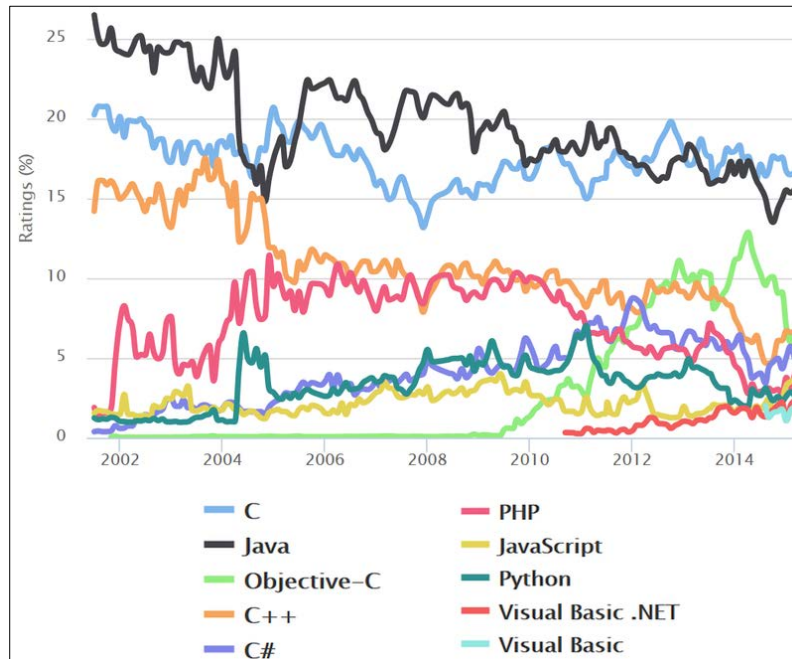
Επιπλέον οι προγραμματιστές επιλέγοντας τις κατάλληλες βιβλιοθήκες και κλάσεις που παρέχει το Java SE ή το Java EE τους δίνεται η δυνατότητα ανάπτυξης εφαρμογών για πληθώρα περιβάλλοντα όπως είναι το διαδικτυακό περιβάλλον, το γραφικό περιβάλλον αλλά και το περιβάλλον γραμμής εντολών. Στην διπλωματική αυτή χρησιμοποιούνται κυρίως κλάσεις και βιβλιοθήκες διαδικτυακού περιβάλλοντος καθώς η πλατφόρμα που αναπτύσσεται αφορά το διαδίκτυο. Συνεπώς μερικές από τις διαθέσιμες και ευρέως χρησιμοποιούμενες κλάσεις για διαδικτυακό περιβάλλον είναι τα Java Servlet, μια τεχνολογία ικανή να παρέχει στους προγραμματιστές εργαλεία για την επέκταση των λειτουργιών ενός διαδικτυακού εξυπηρετητή, αλλά και την επικοινωνία του με υπάρχοντα συστήματα. Επίσης μια άλλη τεχνολογία ανάπτυξης διαδικτυακών εφαρμογών είναι τα Java Server Pages (JSP), τα οποία απαντούν σε HTTP αιτήσεις που πραγματοποιούν οι πελάτες παράγοντας συνήθως HTML σελίδες. Τα JSP καταφέρνουν αυτή τη λειτουργία ενσωματώνοντας στην HTML σελίδα, Java κώδικα. Τέλος, μετά την εμφάνιση και την διάδοση αυτών των δύο τεχνολογιών έχουν αναπτυχθεί και κυκλοφορήσει και άλλες παρόμοιες τεχνολογίες που προσφέρουν περισσότερα εργαλεία και πιο εύχρηστα. Δυο τέτοιες τεχνολογίες είναι τα Java Server Faces και Java Prime Faces τα οποία αναλύονται στην συνέχεια.

Συμπερασματικά μπορούμε να πούμε πως η Java είναι ένα σημαντικό εργαλείο ανάπτυξης εφαρμογών για τους προγραμματιστές οι οποίοι την προτιμάνε ιδιαίτερα, λαμβάνοντας υπόψη μας και το γράφημα στην Εικόνα 13. Η προτίμηση που υπάρχει προς την Java οφείλεται κυρίως στα εξής πλεονεκτήματα:

- Είναι απλή στην χρήση και αντικειμενοστραφής
- Προσφέρει ασφάλεια

- Είναι ανεξάρτητη αρχιτεκτονικής
- Έχει υψηλές επιδόσεις

Με βάση αυτά τα πλεονεκτήματα και τις προβλέψεις της tiobe.com θα συνεχίσει να είναι ψηλά στις προτιμήσεις των προγραμματιστών.



Εικόνα 13: Δημοτικότητα γλωσσών προγραμματισμού από το 2002 – 2015 όπως έχει δημοσιευθεί στη σελίδα tiobe.com

Java Server Faces (JSF)

Τα JavaServer Faces δημιουργήθηκαν από την Java Community Process (JCP) στην οποία συμμετείχαν πρωτοπόρες στην τεχνολογία εταιρίες (π.χ Sun Microsystems, Oracle, IBM κτλ) και ειδικοί στο διαδίκτυο και στην Java. Συνεπώς ο πυρήνας των JSF είναι ένα Java Framework για την δημιουργία διεπαφής χρήστη σε διαδικτυακές εφαρμογές^[37], το οποίο πρώτη φορά εμφανίστηκε το 2004 στην έκδοση J2EE 1.4. Ειδικότερα αυτό που προσφέρουν τα JSF ως βασικό στοιχείο είναι η απλοποίηση της διαδικασίας ανάπτυξης διεπαφών χρήστη, η οποία συνήθως αποτελεί το πιο δύσκολο στάδιο κατά την ανάπτυξη μιας διαδικτυακής πλατφόρμας. Σήμερα τα JSF χρησιμοποιούνται παρότι υπάρχει η δυνατότητα δημιουργίας διεπαφών χρήστη χρησιμοποιώντας θεμελιώδεις τεχνολογίες διαδικτύου της Java, όπως τα Java servlets και τα JavaServer Pages. Σύμφωνα όμως με το βιβλίο «JavaServer Faces 2.0 The complete reference» χωρίς την ύπαρξη ενός ολοκληρωμένου Framework, όπως τα JSF σχεδιασμένο για την ανάπτυξη διαδικτυακών εφαρμογών, αυτές οι τεχνολογίες μπορούν να οδηγήσουν σε πλήθος προβλημάτων ανάπτυξης και συντήρησης. Τα JSF αποτρέπουν αυτά τα προβλήματα προ-

σφέροντας καθιερωμένα πρότυπα ανάπτυξης, χτισμένα πάνω σε προϋπάρχοντα Framework της Java για ανάπτυξη διαδικτυακών εφαρμογών.

Τα JSF κατορθώνουν να πετύχουν τον σκοπό τους ο οποίος, όπως προείπαμε, είναι η απλοποίηση ανάπτυξης της διεπαφής χρήστη, προσφέροντας:

- Μια προσέγγιση ανάπτυξης ανεξάρτητη από τον πελάτη για την δημιουργία διαδικτυακών διεπαφών χρήστη. Η προσέγγιση αυτή βελτιώνει την αποδοτικότητα και την ευχρηστία κατά την ανάπτυξη.
- Απλοποιημένη πρόσβαση και διαχείριση των δεδομένων που προέρχονται από την διαδικτυακή διεπαφή χρήστη.
- Αυτοματοποιημένη διαχείριση των καταστάσεων της διεπαφής χρήστη μεταξύ πολλαπλών αιτήσεων και χρηστών.
- Ένα Framework ανάπτυξης το οποίο είναι φιλικό σε διάφορους προγραμματιστές με διαφορετικές ικανότητες.
- Μια περιγραφή των αρχιτεκτονικών προτύπων για την ανάπτυξη μιας διαδικτυακής πλατφόρμας.

Επιπλέον τα JSF προσφέρουν ακόμα ένα πλεονέκτημα, ίσως το σημαντικότερο. Συλλέγουν τα καλύτερα στοιχεία απαραίτητα για την ανάπτυξη στο διαδίκτυο, που έχουν δημιουργηθεί σε βάθος χρόνου, και τα συνδυάζουν σε μια ολοκληρωμένη διεπαφή προγραμματισμού εφαρμογών (API) για την δημιουργία διεπαφής χρήστη σε Java διαδικτυακές εφαρμογές.

Η αρχιτεκτονική των JSF βασίζεται στο Java EE Web Application Standard με λίγες ειδικές διαμορφώσεις.

- Ο JSF Controller (Ελεγκτής) ενεργοποιείται μέσω του αρχείου web.xml[38] ορίζοντας μια συγκεκριμένη διαδρομή (URL) όπως την /faces/*.
- Όλα τα στοιχεία μιας εφαρμογής JSF μπορούν να οριστούν στο προαιρετικό αρχείο διαμόρφωσης faces-config.xml. Το αρχείο αυτό αντιμετωπίζεται ισάξια με το web.xml και τοποθετείται στον κατάλογο WEB-INF/.
- Όταν μια εφαρμογή τρέχει σε έναν web container ο οποίος δεν υποστηρίζει τα JSF, τότε οι βιβλιοθήκες που συνθέτουν το JSF πρέπει να τοποθετούνται στον κατάλογο WEB-INF/lib.

Εφόσον μια Java EE διαδικτυακή εφαρμογή έχει ρυθμιστεί σωστά για να υποστηρίξει τα JSF, τότε μπορούμε να αρχίσουμε την δόμηση των views^[39] (σελίδες πελάτη). Η

ανάπτυξη μιας σελίδας JSF γίνεται με την χρήση κανόνων και την σύνταξη της XHTML, για τον λόγο αυτό πρέπει να γίνει εισαγωγή των βιβλιοθηκών που φαίνονται στην Εικόνα 14: JSF XML namespaces Εικόνα 14. Επειδή σε μια JSF σελίδα γράφουμε XHTML, τα HTML στοιχεία πρέπει να είναι όλα με μικρούς χαρακτήρες. Ένα στοιχείο παραδείγμα κώδικα JSF σελίδας φαίνεται στην Εικόνα 15.

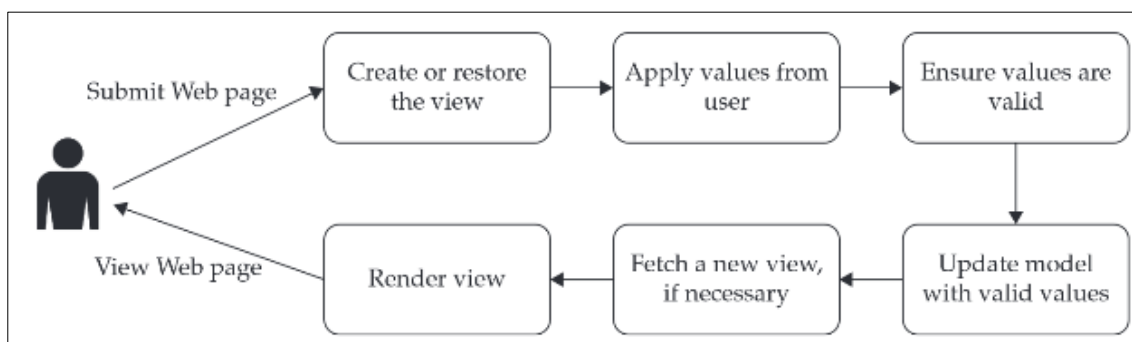
```
<html xmlns=http://www.w3.org/1999/xhtml
      xmlns:h=http://java.sun.com/jsf/html
      xmlns:f="http://java.sun.com/jsf/core">
```

Εικόνα 14: JSF XML namespaces

```
<html xmlns=http://www.w3.org/1999/xhtml
      xmlns:h=http://java.sun.com/jsf/html
      xmlns:f="http://java.sun.com/jsf/core">
  <body>
    <h:form>
      <h2>A Simple JSF Page</h2>
      <h:inputText value="#{modelBean.username}"/>
      <h:commandButton value="Click Here"/>
    </h:form>
  </body>
</html>
```

Εικόνα 15: Κώδικας μιας JSF σελίδας

Όταν κάποιος χρήστης στείλει αίτηση για μια JSF σελίδα ή κάνει μια ενέργεια πάνω σε αυτήν τότε ξεκινάει μια σειρά από events (συμβάντα) στον server για την δημιουργία της σελίδας ή για την υποδοχή των δεδομένων από αυτήν. Η σειρά των events που δημιουργούνται κατά την αίτηση μιας JSF σελίδας είναι γνωστά ως ο κύκλος ζωής της JSF (Εικόνα 16). Στο σημείο αυτό αξίζει να σημειωθεί πως ο ίδιος κύκλος ζωής με τα JSF εφαρμόζεται σε όλες τις διαδικτυακές εφαρμογές. Κάθε διαδικτυακή εφαρμογή πρέπει να εκτελεί και να υποστηρίζει αυτόν τον κύκλο ζωής αλλιώς θα πάσχει από έλλειψη παραγωγικής ποιότητας.

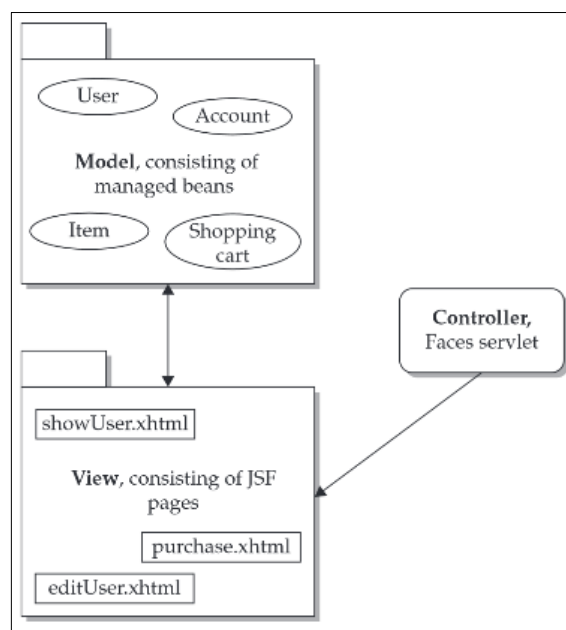


Εικόνα 16: Κύκλος ζωής JSF (πηγή: wikipedia.com)

Τέλος πρέπει να αναφερθούμε και στο μοντέλο πλοήγησης (navigation model) το οποίο ακολουθεί το κλασικό μοντέλο Model – View – Controller^[39]. Συνεπώς όπως μια MVC εφαρμογή, έτσι και μια JSF εφαρμογή είναι χωρισμένη σε τρία ξεχωριστά στοιχεία:

- Το μοντέλο, περιέχει την λογική των δεδομένων ή τον κώδικα που δεν σχετίζεται με το περιβάλλον χρήστη.
- Την σελίδα χρήστη (View), περιέχει τον απαραίτητο κώδικα για το περιβάλλον του χρήστη.
- Τον ελεγκτή (Controller), χαρίζεται τις αιτήσεις του χρήστη, παίρνει τα κατάλληλα δεδομένα από το μοντέλο και επιστρέφει σε αυτόν το κατάλληλο view.

Ο συνδυασμός των τριών αυτών στοιχείων, που αναπαρίστανται και γραφικά στην Εικόνα 17, παράγουν μια αρχιτεκτονική που αποδίδει μοναδικά, ξεχωριστά και διαχειρίσιμα κομμάτια κώδικα.



Εικόνα 17: Στοιχεία του MVC μοντέλου (πηγή: wikipedia.com)

Συνεπώς σύμφωνα με τα παραπάνω τα JSF είναι ένα Java Framework που προσφέρει την δυνατότητα ανάπτυξης διαδικτυακών εφαρμογών διευκολύνοντας όμως την διαδικασία δημιουργίας διαδικτυακής διεπαφής χρήστη.

Java PrimeFaces

Τα PrimeFaces είναι μια ανεξάρτητη ανοιχτού κώδικα βιβλιοθήκη, γραμμένη σε Java, με γραφικά στοιχεία για το περιβάλλον του χρήστη. Η ανάπτυξη των PrimeFaces άρχισε το 2008 και έγιναν διαθέσιμα το 2009, από τότε μέχρι σήμερα έχουν εκδοθεί πολλές

εκδόσεις. Συγκεκριμένα το 2014 εκδόθηκε η έκδοση PrimeFaces 5.0 η οποία όπως θα δούμε και στο κεφάλαιο 6 χρησιμοποιείται στην ανάπτυξη της διαδικτυακής πλατφόρμας. Σήμερα το 2015, βρισκόμαστε στην έκδοση 5.2.

Τα PrimeFaces μπορούν να εφαρμοστούν σε διαδικτυακές εφαρμογές που έχουν αναπτυχθεί με το JSF Framework. Η χρήση της βιβλιοθήκης PrimeFaces σε μια JSF διαδικτυακή εφαρμογή, σύμφωνα με το βιβλίο «PrimeFaces Beginners Guide» προσφέρει

- Πάνω από 100 διαφορετικά γραφικά στοιχεία.
- Υποστήριξη της τεχνολογίας AJAX.
- Ευκολία στην χρήση, δεν χρειάζεται καμία ρύθμιση.
- Ανεξαρτησία από άλλες βιβλιοθήκες
- Υποστήριξη των πιο γνωστών φυλλομετρητών (IE8+, Chrome, Firefox, Safari, και Opera)

Συνεπώς για να χρησιμοποιήσουμε τα PrimeFaces το μόνο που πρέπει να κάνουμε είναι να εισάγουμε το jar^[40] αρχείο τους στην εφαρμογή μας και να αρχίσουμε να εκμεταλλευόμαστε τις δυνατότητες που μας προσφέρει. Παράδειγμα χρήσεις των PrimeFaces μέσα από JSF εφαρμογή φαίνεται στην Εικόνα 18. Οι γραμμές που πρέπει να εξηγήσουμε στον κώδικα της ίδιας σελίδας είναι δύο

- `xmlns:p="http://primefaces.org/ui"` όπου στην γραμμή αυτή γίνεται χρήση του xml namespace των PrimeFaces.
- `<p:editor value="Hello World, PrimeFacesRocks!!"/>` όπου σε αυτή την γραμμή γίνεται χρήση του PrimeFaces στοιχείου editor για την εμφάνιση στον browser του κειμένου Hello world

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:p="http://primefaces.org/ui">
  <f:view contentType="text/html">
    <h:head>
      <title>First PrimeFaces Page</title>
    </h:head>
    <body>
      <h:form>
        <p:editor value="Hello World, PrimeFaces
          Rocks!!"/>
      </h:form>
    </body>
  </f:view>
</html>
```

Εικόνα 18: PrimeFaces, Hello World εφαρμογή

Ο κώδικας που παρουσιάστηκε στην Εικόνα 18 αποτελεί και την γενική δομή μιας JSF εφαρμογής που χρησιμοποιεί την βιβλιοθήκη των PrimeFaces. Για τον εμπλουτισμό της με περεταίρω στοιχεία των PrimeFaces και για την κατανόηση τους μπορούμε να απευθυνθούμε στο εγχειρίδιο χρήσης των PrimeFaces^[41].

2.2 Προσδιορισμός Τοποθεσίας & Τεχνολογικές Λύσεις

Μια από τις μεγαλύτερες δυνατότητες των κινητών συσκευών είναι ότι μπορούν να βρίσκονται οπουδήποτε μαζί μας. Για αυτό και το “που” είναι τόσο σημαντική πληροφορία και μας κάνει να θέλουμε να την ενσωματώσουμε στις διαδικτυακές εφαρμογές. Το να ξέρουμε την τοποθεσία των χρηστών, μας δίνει την δυνατότητα να τους παρέχουμε χρησιμότερες πληροφορίες. Για παράδειγμα αν κάποιος ζει στην Ελλάδα δεν έχει λόγο να βλέπει πληροφορίες για την ταχύτητα των ασύρματων δικτύων μιας άλλης χώρας, αν δεν το επιλέξει ο ίδιος φυσικά. Για τους λόγους αυτούς οι υπηρεσίες βασισμένες στην τοποθεσία (Location – based services –LBS) αποτελούν ένα από τα βασικότερα χαρακτηριστικά των διαδικτυακών εφαρμογών βασισμένων στην τοποθεσία. Κατά την εφαρμογή όμως των υπηρεσιών αυτών στην ιστοσελίδα μας εμφανίζονται δυο προβλήματα, το πρόβλημα του εντοπισμού της θέσης και το πρόβλημα της απεικόνισης της στην διαδικτυακή πλατφόρμα.

Αναφορικά με τον εντοπισμό της θέσης του χρήστη, μπορεί να πραγματοποιηθεί είτε στον εξυπηρετητή είτε στον πελάτη. Ο προσδιορισμός της θέσης στον εξυπηρετητή μπορεί να γίνει μέσω της IP του πελάτη. Ωστόσο η μέθοδος αυτή δεν είναι τόσο απλή όσο ακούγεται. Ανάλογα στο τι τύπο σύνδεσης (2G, 3G, 4G, WIFI) χρησιμοποιεί ο χρήστης, η διεύθυνση IP που λαμβάνουμε μπορεί να είναι η διεύθυνση του παρόχου δικτύου ή της ασύρματης σύνδεσης. Χρησιμοποιώντας αυτή τη μέθοδο, η ακρίβεια στον εντοπισμό της θέσης είναι στο επίπεδο της χώρας ή της πόλης. Άλλη μια μέθοδος που μπορούμε να χρησιμοποιήσουμε στον εξυπηρετητή είναι η αναγνώριση των ρυθμίσεων του φυλλομετρητή, αν ο χρήστης έχει κάνει τις σωστές. Αλλά σε αυτή την μέθοδο εκτός από το μειονέκτημα στην ακρίβεια του εντοπισμού προστίθεται και η πιθανότητα λάθους από την μεριά του χρήστη, σε περίπτωση που δεν έχει ρυθμίσει σωστά τον φυλλομετρητή.

Από την άλλη πλευρά, όταν ο εντοπισμός της θέσης γίνεται στον πελάτη υπάρχουν πιο εξελιγμένες μέθοδοι. Με την πιο διαδεδομένη μέθοδο, το γνωστό σε όλους μας GPS (Global Positioning System). Το σύστημα GPS έχει ενσωματωμένο έναν GPS δέκτη, σε κάθε συσκευή που την υποστηρίζει. Ο δέκτης για να προσδιορίσει την θέση της συ-

σκευής διαβάζει δεδομένα από δορυφόρο. Στα κινητά, το σφάλμα ακρίβειας του GPS, είναι μεταξύ 2m και 100m. Το κινητό όμως θα πρέπει να κοιτάζει ελεύθερο ουρανό και ο χρόνος που χρειάζεται για να υπολογιστεί η τοποθεσία είναι μεταξύ 5 δευτερολέπτων και 5 λεπτών. Επίσης υπάρχουν και άλλες μέθοδοι εντοπισμού θέσης που όμως δεν χρησιμοποιούνται τόσο σε διαδικτυακές εφαρμογές. Τέτοιες είναι το A-GPS, ο εντοπισμός θέσης μέσω πληροφοριών κυψέλης (Cell information) και τέλος το ασύρματο σύστημα θέσης (WiFi Positioning System). Τι γίνεται όμως αν δεν υπάρχουν ή δεν τηρούνται οι κατάλληλες συνθήκες και οι παραπάνω τεχνικές προσδιορισμού θέσης αποτύχουν; η απάντηση σε αυτό το ερώτημα είναι ότι απευθυνόμαστε στον χρήστη. Δηλαδή η εφαρμογή μας, ρωτάει τον χρήστη να προσδιορίσει την τοποθεσία του.

Αναφορικά με το πρόβλημα της απεικόνισης της τοποθεσίας σε διαδικτυακές εφαρμογές έχουν αναπτυχθεί τεχνολογίες και εργαλεία που μπορούμε να χρησιμοποιήσουμε για αυτόν τον σκοπό. Τα πιο γνωστά στο ευρύ κοινό είναι το GoogleMaps, OpenLayers^[43] και BingMaps^[44] για τα οποία υπάρχουν διαθέσιμα API που οι προγραμματιστές μπορούν να χρησιμοποιήσουν για να παρουσιάσουν στους χρήστες γεωχωρικές πληροφορίες. Στην διπλωματική αυτή χρησιμοποιείται το API των OpenLayers επειδή είναι ανοιχτού κώδικα και αυτό το API θα παρουσιάσουμε στην συνέχεια.

2.2.1 OpenLayers

Τα OpenLayers είναι μια Javascript βιβλιοθήκη ανοιχτού κώδικα για την αναπαράσταση πληροφοριών χάρτη στους φυλλομετρητές. Ουσιαστικά η βιβλιοθήκη αυτή παρέχει API^[43] στους προγραμματιστές για την δημιουργία διαδικτυακών εφαρμογών βασισμένων στην τοποθεσία. Όμως, πριν αρχίσει κανείς να χρησιμοποιεί τα OpenLayers υπάρχουν δύο βασικά στοιχεία που πρέπει να γίνουν κατανοητά, ο χάρτης (Map) και το επίπεδο (Leyer). Συγκεκριμένα το στοιχείο Map των Openlayers αποθηκεύει πληροφορίες σχετικά με τα projection (προβολή), extents (επεκτάσεις), units (μονάδες) κ.α. του χάρτη. Ενώ από την άλλη, μέσα στον χάρτη τα δεδομένα παρουσιάζονται μέσω layers (επίπεδα). Ένα Layer παρέχει την πηγή δεδομένων και πληροφορίες στα OpenLayers για το πώς θα γίνει η αίτηση των δεδομένων και πώς θα απεικονιστούν. Συνεπώς σε συνδυασμό των στοιχείων Map και Layer γίνεται δυνατή από τα OpenLayers η αίτηση και η απεικόνιση των δεδομένων στον browser.

Για να γίνει όμως όσο το δυνατόν καλύτερη η κατανόηση της χρήσης των OpenLayers σε μια διαδικτυακή εφαρμογή θα παρουσιάσουμε και θα εξηγήσουμε στο σημείο αυτό, τον βασικό κώδικα που χρειάζεται (Εικόνα 19). Αρχικά για να δημιουρ-

γήσουμε έναν OpenLayer χάρτη στην εφαρμογή μας θα πρέπει να δημιουργήσουμε την HTML σελίδα μέσω της οποίας οι χρήστες θα βλέπουν αυτόν τον χάρτη. Τα OpenLayers υποστηρίζουν την εισαγωγή χάρτη μέσα σε οποιοδήποτε επίπεδο στοιχείων στον HTML κώδικα. Πρακτικά αυτό σημαίνει πως μπορούμε να εισάγουμε τον χάρτη σε οποιοδήποτε στοιχείο HTML με μοναδικό απαιτούμενο να υπάρχει ένα script tag με το οποίο θα γίνεται η εισαγωγή της βιβλιοθήκης των OpenLayers στην HTML σελίδα.

```
<!doctype html>
<html lang="en">
  <head>
    <link rel="stylesheet" href="http://openlayers.org/en/v3.0.0/css/ol.css" type="text/css">
    <style>
      .map {
        height: 400px;
        width: 100%;
      }
    </style>
    <script src="http://openlayers.org/en/v3.0.0/build/ol.js" type="text/javascript"></script>
    <title>OpenLayers 3 example</title>
  </head>
  <body>
    <h2>My Map</h2>
    <div id="map" class="map"></div>
    <script type="text/javascript">
      var map = new ol.Map({
        target: 'map',
        layers: [
          new ol.layer.Tile({
            source: new ol.source.MapQuest({layer: 'sat'})
          })
        ],
        view: new ol.View({
          center: ol.proj.transform([37.41, 8.82], 'EPSG:4326', 'EPSG:3857'),
          zoom: 4
        })
      });
    </script>
  </body>
</html>
```

Εικόνα 19: Βασική χρήση OpenLayers 3

Στη συνέχεια θα πρέπει να δημιουργηθεί μέσω JavaScript ο χάρτης. Για τον σκοπό αυτό υπάρχει ο `ol.Map` κατασκευαστής (constructor) στον οποίο μπορούμε να ορίσουμε παραμέτρους με βασικότερες να είναι το `target`, το `layers` και το `view`. Στη παράμετρο `target` μπορούμε να ορίσουμε το HTML στοιχείο στο οποίο έχουμε αποφασίσει να εμφανιστεί ο χάρτης, δίνοντας το `id` του. Ενώ στην παράμετρο `layers` ορίζονται τα layer που θα εμφανιστούν στον χάρτη. Συγκεκριμένα στην Εικόνα 19 καλείται το layer που αναπαριστά γραφικά τον χάρτη σε μορφή *satellite*. Επιπλέον στην παράμετρο `layers` μπορούμε να ορίσουμε και οτιδήποτε άλλο θέλουμε να εμφανιστεί στον χάρτη, π.χ. δρόμους, σύνορα χωρών κτλ. Από την άλλη η παράμετρος `view` αφορά τον τρόπο εμφάνισης του χάρτη στην οθόνη και σε αυτήν μπορούμε να ορίσουμε ένα σύνολο από παραμέτρους με σημαντικότερες να είναι το σύστημα υπολογισμού συντεταγμένων που θα

χρησιμοποιείται και η μεγέθυνση του χάρτη. Συνεπώς οι τρεις αυτοί παράμετροι είναι οι βασική για την δημιουργία ενός OpenLayers χάρτη και ο κώδικας στην Εικόνα 19 είναι ο ελάχιστος απαιτούμενος για να εμφανιστεί αυτός ο χάρτης στον browser. Περισσότερο κώδικα και λειτουργίες των OpenLayers Maps παρουσιάζονται στο κεφάλαιο 6 όπου θα γίνει και η παρουσίαση της διαδικτυακής εφαρμογής που κάνει χρήση των OpenLayers. Επίσης μεγάλη βοήθεια στην κατανόηση της ανάπτυξης διαδικτυακής εφαρμογής με τη χρήση των OpenLayers, μπορεί κανείς να βρει στον επίσημο διαδικτυακό τόπο του OpenLayers API^[43].

3 Σύντηξη Δεδομένων (Data Fusion)

Το άρθρο “Μια Ανασκόπηση στις Τεχνικές Σύντηξης Δεδομένων” του Federico Castanedo μας εξηγεί πως όλες οι εργασίες οι οποίες απαιτούν εκτίμηση παραμέτρων από πολλαπλούς αισθητήρες μπορούν να επωφεληθούν από τις μεθόδους σύντηξης δεδομένων. Επίσης αναφέρει πως οι διαθέσιμες τεχνικές σύντηξης δεδομένων μπορούν να κατηγοριοποιηθούν σε τρεις κατηγορίες:

- Συσχέτισης δεδομένων
- Εκτίμησης κατάστασης
- Λήψης αποφάσεων

Επιπλέον στο ίδιο άρθρο δίνονται δυο γνωστοί και ευρέως αποδεκτοί ορισμοί του όρου σύντηξη δεδομένων. Στον πρώτο ορισμό η έννοια της σύντηξης δεδομένων ορίζεται ως μια πολύ-επίπεδη διαδικασία που ασχολείται με τη σύνδεση, το συσχετισμό και το συνδυασμό των δεδομένων και πληροφοριών από μια ή πολλές πηγές, με σκοπό την επίτευξη της καλύτερης δυνατής εκτίμησης και την πλήρη και έγκυρη αξιολόγηση των δεδομένων αυτών^[45]. Ο δεύτερος ορισμός που παρατίθεται μας λέει πως οι τεχνικές σύντηξης δεδομένων συνδυάζουν δεδομένα από πολλαπλούς αισθητήρες και πληροφορίες από συσχετιζόμενες βάσεις δεδομένων, για να επιτύχουν τη βέλτιστη ακρίβεια υπολογισμού η οποία δεν θα μπορούσε να επιτευχθεί με τη χρήση ενός αισθητήρα μόνο^[46]. Ο δεύτερος ορισμός είναι και αυτός που εκφράζει περισσότερο την εργασία σε αυτή τη διπλωματική. Εν συντομία λοιπόν, μπορούμε να πούμε ότι η σύντηξη δεδομένων είναι ο συνδυασμός δεδομένων από πολλαπλές πηγές με σκοπό την δημιουργία βέλτιστης πληροφορίας. Ο όρος βέλτιστη πληροφορία αναφέρεται στην δημιουργία φθηνής, καλής ποιότητας και σχετικής με το πρόβλημα πληροφορίες.

Στη παρούσα εργασία γίνεται χρήση των τεχνικών σύντηξης δεδομένων (η θεωρητική και μαθηματική ανάλυσή τους υπάρχει στο άρθρο^[47]) και ειδικότερα των τεχνικών εκείνων που ανήκουν στην κατηγορία εκτίμησης μιας κατάστασης. Η χρήση τεχνικών σύντηξης δεδομένων κρίθηκε απαραίτητη καθώς, σύμφωνα και με τους ορισμούς που δόθηκαν, είναι οι καταλληλότερες τεχνικές για την παραγωγή πληροφορίας από δεδομένα με γεωχωρική αναφορά τα οποία συλλέγονται μέσω πολλαπλών αισθητήρων. Στην περίπτωση μας ως αισθητήρες μπορούν να οριστούν οι έξυπνες κινητές συσκευές,

μέσω των οποίων πραγματοποιείται η συλλογή των δεδομένων για την απόδοση των ασύρματων δικτύων. Όμως για την επεξεργασία των δεδομένων αυτών αναγκαίο προαπαιτούμενο είναι η ανάπτυξη αλγόριθμων βασισμένων στις τεχνικές σύντηξης. Στην παρούσα διπλωματική δεν αναπτύσσεται κάποιος νέος αλγόριθμος σύντηξης δεδομένων, άλλα για την επίτευξη των επιθυμητών αποτελεσμάτων χρησιμοποιούνται διαθέσιμα εργαλεία που προσφέρουν αλγόριθμους βασισμένους σε αυτές τις τεχνικές.

Ιδανικές εφαρμογές για τη σύντηξη γεωχωρικών δεδομένων, όπως στην περίπτωση μας, είναι οι Map Servers (εξυπηρετητές χαρτών) όπως ο GeoServer^[18] και ο MapServer^[17]. Ένας Map Server μας δίνει την δυνατότητα να τον συνδέσουμε με μια βάση δεδομένων και να επεξεργαστεί τα δεδομένα εφαρμόζοντας αλγόριθμους βασισμένους στις τεχνικές σύντηξης δεδομένων. Η πληροφορία που προκύπτει, από τους Map Server μπορεί να έχει τη μορφή κειμένου (π.χ json) ή εικόνων (π.χ png). Και οι δύο, GeoServer και MapServer, παρέχουν πλήθος αλγόριθμων κατάλληλων για σύντηξη δεδομένων. Η δική μας πλατφόρμα όμως συνεργάζεται με τον GeoServer για τον οποίον γίνεται μια συνοπτική παρουσίαση στο κεφάλαιο 3.1. Οι λόγοι για τους οποίους επιλέχτηκε ο GeoServer για τη σύντηξη των δεδομένων μας είναι δύο. Ο πρώτος λόγος έχει να κάνει με τον αλγόριθμο BarnesSurface (κεφάλαιο 3.2) που μας προσφέρει και τον οποίο έχουμε επιλέξει ως τεχνική σύντηξης των δεδομένων μας και ο δεύτερος λόγος είναι η δυνατότητα παραμετροποίησης του, καθώς αποτελεί μια εφαρμογή ανοιχτού κώδικα.

3.1 GeoServer

Ο GeoServer είναι μια εφαρμογή που επιτρέπει στους χρήστες να μοιράζονται και να επεξεργάζονται γεωχωρικά δεδομένα. Επίσης είναι ανοιχτού κώδικα εφαρμογή γραμμένη σε Java και σχεδιασμένη να προσφέρει διαλειτουργικότητα, επεξεργάζοντας γεωχωρικά δεδομένα που προέρχονται από οποιαδήποτε πηγή δεδομένων χρησιμοποιώντας ανοιχτά πρότυπα. άρχισε να αναπτύσσεται το 2001 από τον μη κερδοσκοπικό οργανισμό The Open Planning Project^[53] (TOPP). Επιπλέον ο GeoServer, σύμφωνα με το εγχειρίδιο χρήσης του^[48], αποτελεί την υλοποίηση αναφοράς των προτύπων Open Geospatial Consortium^[49] (OGC), Web Feature Service^[50] (WFS) και Web Coverage Service^[51] (WCS), καθώς και ένα πιστοποιημένο υψηλής απόδοσης Web Map Service^[52] (WMS).

Ο GeoServer μπορεί να εγκατασταθεί ως εφαρμογή στα λειτουργικά συστήματα Windows, Mac OS και Linux, μπορεί όμως να χρησιμοποιηθεί και με υπάρχον servlet container application όπως το Apache Tomcat και Jetty. Επιπλέον για να τρέξει ο Ge-

οServer προαπαιτεί να υπάρχει εγκαταστημένη η Java στο σύστημα. Για τις ανάγκες της δικής μας εργασίας ο Geoserver τρέχει στον Apache Tomcat. Ο λόγος που έγινε επιλογή αυτής της εγκατάστασης είναι η ευκολία στην απομακρυσμένη διαχείριση του, καθώς μας προσφέρει την διαδικτυακή διεπαφή διαχείρισης από την οποία μπορούμε να δημοσιοποιήσουμε τα δεδομένα μας.

Τα βασικά που μας προσφέρει ο GeoServer για την συλλογή, επεξεργασία και δημοσιοποίηση των δεδομένων μας είναι

- Επικοινωνία με την πηγή των δεδομένων
- Δημιουργία Layers για την κατάλληλη επεξεργασία των δεδομένων
- Δημιουργία Styles για την κατάλληλη μορφοποίηση της πληροφορίας
- Δημοσιοποίηση της πληροφορίας μέσω τυποποιημένων πρωτοκόλλων

Συνεπώς για να εκμεταλλευτούμε τις δυνατότητες του GeoServer πρέπει αρχικά να καταστήσουμε δυνατή τη σύνδεση του με την πηγή των δεδομένων μας. Αυτό επιτυγχάνεται μέσω της διεπαφής διαχείρισης και δημιουργώντας ένα νέο Data Store. Κατά την δημιουργία ενός data store έχουμε να επιλέξουμε μεταξύ δύο βασικών κατηγοριών πηγών δεδομένων:

- Vector Data Sources (διανυσματικές πηγές δεδομένων)
- Raster Data Sources (πηγές δεδομένων με εικόνες γεωχωρική αναφορά)

Ως vector data sources μπορούν να οριστούν οι πηγές που περιέχουν μεμονωμένα σημεία, τα οποία (για 2D δεδομένα) αποθηκεύονται ως ζεύγη (x,y) συντεταγμένων. Τα σημεία αυτά μπορούν να ενωθούν σε μια συγκεκριμένη σειρά ή να ενταχθούν σε κλειστό δακτύλιο για την δημιουργία γραμμών ή πολυγώνων αντίστοιχα. Δηλαδή όλα τα διανυσματικά δεδομένα συνοδεύονται ουσιαστικά από λίστες συντεταγμένων που ορίζουν τις κορυφές και τους κανόνες που καθορίζουν αν και το πώς ενώνονται αυτές οι κορυφές. Από την άλλη τα raster data αποτελούνται από εικονοστοιχεία που το κάθε ένα σχετίζεται με μια τιμή. Ως απλουστευμένο παράδειγμα raster δεδομένων αποτελεί μια ψηφιακή φωτογραφία όπου κάθε τιμή εικονοστοιχείου αντιστοιχεί σε ένα συγκεκριμένο χρώμα. Στο γεωγραφικό σύστημα πληροφοριών (ΓΣΠ – GIS), οι τιμές των εικονοστοιχείων μπορεί να αντιπροσωπεύουν υψόμετρο πάνω από την επιφάνεια της θάλασσας, ή χημικές συγκεντρώσεις, ή βροχοπτώσεις κ.λπ. Το βασικό σημείο είναι ότι όλα αυτά τα δεδομένα αναπαρίστανται ως ένα πλέγμα (συνήθως τετράγωνο) στοιχείων. Η διαφορά μεταξύ ενός ψηφιακού μοντέλου εδάφους (digital elevation model – DEM) στο ΓΣΠ και μια ψηφιακή φωτογραφία είναι ότι η DEM περιλαμβάνει πρόσθετες πλη-

ροφορίες που περιγράφουν την τοποθεσία των άκρων της εικόνας στον πραγματικό κόσμο, μαζί με το πόσο μεγάλο κάθε εικονοστοιχείο είναι στο έδαφος. Αυτό σημαίνει πως οι raster εικόνες (εικόνες με γεωγραφική αναφορά) μπορούν να τοποθετηθούν σωστά η μία δίπλα στην άλλη, επιτρέποντας μας να αναπαραστήσουμε τον χάρτη. Στο σημείο αυτό αξίζει να σημειωθεί πως από την αποθήκευση των δεδομένων ως Vector ή Raster προκύπτουν, για τα δυο αυτά μοντέλα, κάποια πλεονεκτήματα και μειονεκτήματα από τα οποία τα πιο σημαντικά είναι:

- Τα Raster datasets καταγράφουν μια τιμή για όλα τα σημεία της περιοχής που καλύπτουν το οποίο μπορεί να απαιτήσει περισσότερο αποθηκευτικό χώρο από όσο τα δεδομένα σε vector μορφή. που μπορεί να αποθηκεύουν δεδομένα μόνο όπου χρειάζεται.
- Τα Raster δεδομένα είναι υπολογιστικά λιγότερο ακριβά κατά την απεικόνιση τους σε σχέση με τα διανυσματικά γραφικά
- Τα vector δεδομένα είναι απλούστερο να ενημερωθούν και να διατηρηθούν, ενώ μια εικόνα raster θα πρέπει να αναπαραχθεί πλήρως από την αρχή σε περίπτωση αλλαγής

Μετά την επίτευξη της σύνδεσης με την πηγή των δεδομένων μας πρέπει να αντλήσουμε τα δεδομένα, να τα επεξεργαστούμε και να τα μορφοποιήσουμε, για τον σκοπό αυτό στον GeoServer υπάρχουν τα Layers και τα styles. Τα Layers έχουν σαν σκοπό την επεξεργασία ενός συνόλου των vector ή raster δεδομένων που προέρχονται από την data store που περιγράψαμε, λαμβάνοντας υπόψη παραμέτρους που μπορούμε να ορίσουμε εμείς. Από την άλλη τα styles χρησιμοποιούνται για την μορφοποίηση του συνόλου των vector ή raster δεδομένων ενός layer. Η μορφοποίηση μπορεί να αφορά το χρώμα, το πάχος και άλλες οπτικές παραμέτρους που χρησιμοποιούνται για την αναπαράσταση των δεδομένων στο χάρτη. Τα styles δημιουργούνται με την χρήση μιας γλώσσας σήμανσης που λέγεται Styled Layer Descriptor^[54].

Σαν τελευταίο βήμα, μετά την σύνδεση με το data source και την δημιουργία των layers και styles, είναι η δημοσιοποίηση της πληροφορίας στους πελάτες. Για δημοσιοποίηση της πληροφορίας χρησιμοποιούνται τυποποιημένα πρότυπα που έχει καθιερώσει ο οργανισμός Open Geospatial Consortium^[49] (OGC), τα οποία είναι:

- Το Web Feature Service (WFS). Υποστηρίζει αιτήσεις για δεδομένα με γεωγραφικά χαρακτηριστικά (με διανυσματικές γεωμετρίες και παραμέτρους)

- Το Web Map Service (WMS). Υποστηρίζει αιτήσεις για εικόνες χάρτη (και άλλα format) που παράγονται από γεωγραφικά δεδομένα.
- Το Web Coverage Service (WCS). Υποστηρίζει αιτήσεις για raster δεδομένα.
- Το Web Processing Service (WPS). Είναι μια υπηρεσία της OGC για την δημοσιοποίηση γεωχωρικών διεργασιών, αλγόριθμων και υπολογισμών. Το WPS επεκτείνει τον GeoServer ώστε να παρέχει γεωχωρικές αναλύσεις. Επίσης το WPS δεν είναι μέρος του GeoServer εξορισμού, αλλά είναι διαθέσιμο σαν επέκταση (extension).

Από τις προαναφερθείσες υπηρεσίες αυτή που μας προσφέρει αλγόριθμους για τη σύνηξη δεδομένων είναι το WPS και συγκεκριμένα τον Barnes Surface που εξετάζουμε στη συνέχεια. Άλλοι παρόμοιοι αλγόριθμοι είναι οι Heatmap^[55] και Point Stacker^[56].

3.2 Barnes Surface

Ο αλγόριθμος Barnes Surface^[57] είναι ένας Vector-σε-Raster μετασχηματισμός ο οποίος υπολογίζει την περιβάλλουσα επιφάνεια μεταξύ ακανόνιστων σημείων παρατήρησης. Συνήθως χρησιμοποιείται για την δημιουργία χαρτών καιρού καθώς και για άλλα μετεωρολογικά δεδομένα. Η επιφάνεια που παράγεται από τον μετασχηματισμό υπολογίζεται δυναμικά από το σύνολο των δεδομένων, ώστε να μπορεί να χρησιμοποιηθεί για να απεικονίσει τις αλλαγές που τυχόν συμβαίνουν. Η άποψη της επιφάνειας δημιουργείται από τη διαμόρφωση ενός Layer και ενός SLD style το οποίο καλεί τον Barnes Surface μετασχηματισμό.

Για την καλύτερη κατανόηση του αλγόριθμου πρέπει να αναφέρουμε πως ο Barnes Surface λειτουργεί σε ένα πλέγμα εικονοστοιχείων που καλύπτουν μια συγκεκριμένη έκταση στο χώρο των δεδομένων εισόδου και υπολογίζει μια εκτίμηση για την κατάσταση για το κάθε εικονοστοιχείο που δεν περιέχει πληροφορίες. Η εκτίμηση του κάθε εικονοστοιχείου υπολογίζεται από τα σημεία παρατήρησης εισόδου, χρησιμοποιώντας άθροισμα εκθέτη (Gaussian) για την απόσβεση του κάθε σημείου παρατήρησης. Όσα περισσότερα είναι τα σημεία παρατήρησης στην είσοδο τόσο μειώνεται το σφάλμα της υπολογιζόμενης τιμής σε σχέση με τις πραγματικές παρατηρήσεις. Τέλος η εκτίμηση παρουσιάζεται παράγοντας ως έξοδο μια raster επιφάνεια.

Εμβαθύνοντας στο τεχνικό κομμάτι, μπορούμε να χρησιμοποιήσουμε τον Barnes Surface σε ένα style (SLD) αρχείο, ορίζοντας ένα πλήθος από παραμέτρους όπως παρουσιάζει ο Πίνακας 2. Ένα παράδειγμα χρήσης του αλγόριθμου είναι η Εικόνα 20. Η

εισαγωγή γίνεται με τη χρήση της ετικέτας <Transformation> (γραμμή 14) μέσα στην ετικέτα <FeatureTypeStyle> (γραμμή 13) στο SLD αρχείο. Στη συνέχεια το SLD αρχείο μπορεί να εφαρμοστεί σε οποιοδήποτε Layer που περιέχει τα κατάλληλα δεδομένα. Όπως φαίνεται και στην εικόνα στις γραμμές 15 – 71 γίνεται χρήση του μετασχηματισμού ορίζοντας τις κατάλληλες παραμέτρους, για τα δεδομένα εισαγωγής. Ενώ στις γραμμές 72 – 89 ορίζεται ο συμβολισμός που θα χρησιμοποιηθεί για τη μορφοποίηση του αποτελέσματος που παράγει ο μετασχηματισμός.

Περισσότερες τεχνικές λεπτομέρειες για την χρήση του Barnes Surface θα δοθούν στο κεφάλαιο 6.1. Στο σημείο αυτό θα αρκεστούμε στο να αναφέρουμε τι μπορεί να μας προσφέρει η χρήση του αλγόριθμου στην παρούσα εργασία. Αυτό λοιπόν που μας προσφέρει είναι ο υπολογισμός της περιβάλλουσας επιφάνειας μεταξύ ακανόνιστων παρατηρήσεων για την απόδοση των ασύρματων δικτύων. Με άλλα λόγια, θα υπολογίζει μια εκτίμηση της απόδοσης των ασύρματων δικτύων σε χώρο που δεν υπάρχει παρατήρηση, αλλά βρίσκεται ενδιάμεσα σε τουλάχιστον δύο παρατηρήσεις. Ο υπολογισμός της εκτίμησης αυτής είναι αναγκαίος διότι πρακτικά είναι αδύνατον να υπάρχουν παρατηρήσεις σε κάθε σημείο του χώρου ώστε να έχουμε εικόνα της απόδοσης των δικτύων. Συνεπώς ο Barnes Surface καθιστά δυνατή την ορθή στατιστική απεικόνιση της απόδοσης των ασύρματων δικτύων στο χώρο ανεξάρτητα αν σε αυτό το σημείο υπάρχει μέτρηση ή όχι, αρκεί ο χώρος αυτός να βρίσκεται ενδιάμεσα σε τουλάχιστον δύο μετρήσεις.

Πίνακας 2: Παράμετροι Barnes Surface αλγόριθμου

Name	Required	Description
data	Yes	Σύνολο δεδομένων που εμφανίζονται στον χάρτη
valueAttr	Yes	Το όνομα των δεδομένων προς επεξεργασία (π.χ σε μια βάση δεδομένων το όνομα της στήλης προς επεξεργασία)
dataLimit	No	Αριθμός ορίου παρατηρήσεων προς επεξεργασία
scale	Yes	Μέγεθος κλίμακα για τον μετασχηματισμό.
convergence	No	Συντελεστής σύγκλισης
passes	No	Ο αριθμός των εισαγόμενων σημείων που θα υπολογιστούν.
minObservations	No	Ο ελάχιστος αριθμός των παρατηρήσεων που χρειάζονται για να παραχθεί ένα κελί (εικόνα) σαν αποτέλεσμα.
maxObservationDistance	No	Μέγιστη απόσταση των παρατηρήσεων για την δημιουργία ενός κελιού (εικόνας).
noDataValue	No	Τιμή προς εμφάνιση σε ένα κελί όταν δεν υπάρχουν δεδομένα για αυτό.

pixelsPerCell	No	Ανάλυση του παραγόμενου κελιού (εικόνας)
queryBuffer	No	Απόσταση επέκτασης αιτήματος
outputBBOX	Yes	Γεω –ωαναφορικές συντεταγμένες του αποτελέσματος
outputWidth	Yes	Μήκος παραγόμενης εικόνας
outputHeight	Yes	Ύψος παραγόμενης εικόνας

Εικόνα 20: Παράδειγμα χρήσης Burnes Surface σε SLD αρχείο

```

1  <?xml version="1.0" encoding="ISO-8859-1"?>
2  <StyledLayerDescriptor version="1.0.0"
3    xsi:schemaLocation="http://www.opengis.net/sld StyledLayerDescriptor.xsd"
4    xmlns="http://www.opengis.net/sld"
5    xmlns:ogc="http://www.opengis.net/ogc"
6    xmlns:xlink="http://www.w3.org/1999/xlink"
7    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
8    <NamedLayer>
9      <Name>Burnes surface</Name>
10     <UserStyle>
11       <Title>Burnes Surface</Title>
12       <Abstract>A style that produces a Barnes surface using a rendering transformation</Abstract>
13       <FeatureTypeStyle>
14         <Transformation>
15           <ogc:Function name="gs:BurnesSurface">
16             <ogc:Function name="parameter">
17               <ogc:Literall>data</ogc:Literall>
18             </ogc:Function>
19             <ogc:Function name="parameter">
20               <ogc:Literall>valueAttr</ogc:Literall>
21               <ogc:Literall>value</ogc:Literall>
22             </ogc:Function>
23             <ogc:Function name="parameter">
24               <ogc:Literall>scale</ogc:Literall>
25               <ogc:Literall>15.0</ogc:Literall>
26             </ogc:Function>
27             <ogc:Function name="parameter">
28               <ogc:Literall>convergence</ogc:Literall>
29               <ogc:Literall>0.2</ogc:Literall>
30             </ogc:Function>
31             <ogc:Function name="parameter">
32               <ogc:Literall>passes</ogc:Literall>
33               <ogc:Literall>3</ogc:Literall>
34             </ogc:Function>
35             <ogc:Function name="parameter">
36               <ogc:Literall>minObservations</ogc:Literall>
37               <ogc:Literall>1</ogc:Literall>
38             </ogc:Function>
39             <ogc:Function name="parameter">
40               <ogc:Literall>maxObservationDistance</ogc:Literall>
41               <ogc:Literall>10</ogc:Literall>
42             </ogc:Function>
43             <ogc:Function name="parameter">
44               <ogc:Literall>pixelsPerCell</ogc:Literall>
45               <ogc:Literall>10</ogc:Literall>
46             </ogc:Function>
47             <ogc:Function name="parameter">
48               <ogc:Literall>queryBuffer</ogc:Literall>
49               <ogc:Literall>40</ogc:Literall>
50             </ogc:Function>
51             <ogc:Function name="parameter">
52               <ogc:Literall>outputBBOX</ogc:Literall>
53               <ogc:Function name="env">
54                 <ogc:Literall>wms_bbox</ogc:Literall>
55               </ogc:Function>
56             </ogc:Function>
57             <ogc:Function name="parameter">
58               <ogc:Literall>outputWidth</ogc:Literall>
59               <ogc:Function name="env">
60                 <ogc:Literall>wms_width</ogc:Literall>
61               </ogc:Function>
62             </ogc:Function>
63             <ogc:Function name="parameter">
64               <ogc:Literall>outputHeight</ogc:Literall>
65               <ogc:Function name="env">
66                 <ogc:Literall>wms_height</ogc:Literall>
67               </ogc:Function>
68             </ogc:Function>
69           </ogc:Function>
70         </Transformation>
71         <Rule>
72           <RasterSymbolizer>
73             <!-- specify geometry attribute of input to pass validation -->
74             <Geometry><ogc:PropertyName>point</ogc:PropertyName></Geometry>
75             <Opacity>0.8</Opacity>
76             <ColorMap type="ramp">
77               <ColorMapEntry color="#FFFFFF" quantity="-990" label="nodata" opacity="0"/>
78               <ColorMapEntry color="#2E4AC9" quantity="-9" label="nodata"/>
79               <ColorMapEntry color="#41A0FC" quantity="-6" label="values"/>
80               <ColorMapEntry color="#58CCFB" quantity="-3" label="values"/>
81               <ColorMapEntry color="#76F9FC" quantity="0" label="values"/>
82               <ColorMapEntry color="#6AC597" quantity="3" label="values"/>
83               <ColorMapEntry color="#479364" quantity="6" label="values"/>
84               <ColorMapEntry color="#2E6000" quantity="9" label="values"/>
85               <ColorMapEntry color="#579102" quantity="12" label="values"/>
86               <ColorMapEntry color="#9AF20C" quantity="15" label="values"/>
87               <ColorMapEntry color="#B7F318" quantity="18" label="values"/>
88             </ColorMap>
89           </RasterSymbolizer>
90         </Rule>
91       </FeatureTypeStyle>
92     </UserStyle>
93   </NamedLayer>
94 </StyledLayerDescriptor>

```


4 Υπάρχουσες Πλατφόρμες

Όπως είναι φυσικό δεν είμαστε οι μόνοι, και πόσο μάλλον οι πρώτοι, που παρατηρήσαμε τα προβλήματα τα οποία περιγράφηκαν στο κεφάλαιο 1 και τις τεχνολογικές και τεχνικές λύσεις των κεφαλαίων 2 και 3. Το πρόβλημα της παρατήρησης, της μέτρησης, του υπολογισμού και της παρουσίασης της απόδοσης των ασύρματων δικτύων είναι ένα πρόβλημα που απασχολεί τους τεχνικούς και τους χρήστες πολλά χρόνια. Για τον λόγο αυτό στην παρούσα ενότητα θα αναφέρουμε τις υλοποιημένες προσεγγίσεις (που καταφέραμε να εντοπίσουμε) για τα προβλήματα τα οποία προσπαθεί και η δική μας διαδικτυακή πλατφόρμα να αντιμετωπίσει. Όμως πριν συνεχίσουμε πρέπει να σημειωθεί πως όλες οι υπάρχουσες υλοποιημένες λύσεις δεν ικανοποιούν τις ίδιες ανάγκες ή δεν προσφέρουν λύσεις στα ίδια προβλήματα, αν τα συγκρίνουμε μεταξύ τους ή με την δική μας προσέγγιση. Άρα μπορούμε να συνεχίσουμε λέγοντας πως δεν είναι πλεονασμός η ανάπτυξη μιας ακόμα πλατφόρμας μέτρησης της απόδοσης των ασύρματων δικτύων, καθώς η κάθε προσέγγιση εμβαθύνει και προσπαθεί να αντιμετωπίσει διαφορετικό πεδίο του προβλήματος. Αξίζει όμως και θεωρούμε απαραίτητο την παρουσίαση των “ανταγωνιστικών” εφαρμογών καθώς και την αντικειμενική σύγκριση τους με την δική μας διαδικτυακή πλατφόρμα. Στη συνέχεια, μέσα από την παρουσίαση και σύγκριση των δύο βασικών διαδικτυακών εφαρμογών που είναι κοντά στη δική μας υλοποίηση, θέλουμε και ευελπιστούμε να πετύχουμε την κατανόηση της διαφορετικότητας και τον σκοπό της κάθε προσέγγισης.

Μετά από έρευνα που πραγματοποιήσαμε, παρατηρήσαμε πως οι δύο κυριότερες εφαρμογές που δραστηριοποιούνται στον ίδιο τομέα με την δική μας υλοποίηση είναι οι διαδικτυακές πλατφόρμες Sensorly^[9] και OpenSignal^[58]. Οι δύο αυτές πλατφόρμες έχουν μεταξύ τους και με τη δική μας υλοποίηση αρκετές ομοιότητες και διαφορές. Αρχικά όμως θα επικεντρωθούμε στο σκοπό τους, στις λειτουργίες τους και στις δυνατότητες που προσφέρουν στους χρήστες. Ο σκοπός λοιπόν του Sensorly και του OpenSignal είναι η απεικόνιση στατιστικών στοιχείων για τις επιδόσεις των δικτύων κινητής τηλεφωνίας σε διαδικτυακούς χάρτες με βάση δεδομένα που συλλέγονται από crowdsourcing υποδομή. Αναλυτικότερα, παρέχουν στους χρήστες τους εφαρμογές για κινητά τηλέφωνα συμβατές με τα λειτουργικά Android και iOS. Οι χρήστες έχουν την δυνατότητα μέσω των κινητών τους και των εφαρμογών αυτών να κάνουν μετρήσεις για την απόδοση των δικτύων κινητής τηλεφωνίας τους. Στη συνέχεια οι δύο πλατφόρ-

μες εφαρμόζουν τεχνικές σύντηξης στα δεδομένα από τις μετρήσεις των χρηστών και με την βοήθεια των Map Servers παράγουν την επιθυμητή πληροφορία. Τέλος την παραγόμενη αυτή πληροφορία την παρουσιάζουν διαδικτυακά στους χρήστες με τη χρήση του Google Maps API.

Όπως μπορούμε να δούμε από την παραπάνω σύντομη περιγραφή των Sensorly και OpenSignal η λειτουργία τους και οι δυνατότητες που παρέχουν είναι παρόμοιες με τη δική μας υλοποίηση. Για το λόγο αυτό υπάρχει η ανάγκη να σημειώσουμε ρητά τις ομοιότητες και τις διαφορές μεταξύ των τριών υλοποιήσεων. Οι ομοιότητες λοιπόν είναι πως και οι τρεις πλατφόρμες βασίζονται στο crowdsourcing για την συλλογή των δεδομένων, παρέχοντας στους χρήστες εφαρμογή κατάλληλη για τα κινητά τηλέφωνα τους ώστε να μπορούν να κάνουν τις μετρήσεις σχετικά με την απόδοση των δικτύων κινητής τηλεφωνίας. Η σύντηξη των δεδομένων, που συλλέγονται από τις μετρήσεις, γίνεται με τη χρήση των Map Servers (π.χ MapServer, GeoServer) και η παραγόμενη πληροφορία παρουσιάζεται στον χρήστη με τη χρήση των Map API (π.χ GoogleMaps, OpenLayers) από όπου ο χρήστης έχει τη δυνατότητα να δει την παραγόμενη πληροφορία.

Από την άλλη πλευρά υπάρχουν σημαντικές διαφορές που ξεχωρίζουν τις τρεις εφαρμογές τη μια από την άλλη. Με σημαντικότερη διαφορά τη δυνατότητα δημιουργίας καμπάνιας (campaigns) για το συντονισμό των συσκευών στις μετρήσεις, που παρέχει η πλατφόρμα η οποία αναπτύσσεται στη παρούσα διπλωματική. Δηλαδή προσφέρεται επιπλέον σε σχέση με τις άλλες πλατφόρμες η δυνατότητα να υπάρχει κεντρική διαχείριση των συσκευών που επιθυμούν να συμμετέχουν στις μετρήσεις. Αυτό σημαίνει πως ο διαχειριστής μπορεί να ορίσει μέσω της πλατφόρμας μας περιοχές ενδιαφέροντος προς μέτρηση, τις λεγόμενες καμπάνιες, και να τις στείλει στις κινητές συσκευές. Η ύπαρξη της δυνατότητας ο διαχειριστής να δημιουργεί καμπάνιες προσφέρει τα πλεονεκτήματα:

- Κατανομή των μετρήσεων σε όλο το χώρο. Απαλείφοντας το φαινόμενο σε μια περιοχή να υπάρχουν πολλές μετρήσεις και σε άλλες περιοχές να υπάρχουν λίγες έως καθόλου μετρήσεις.
- Παρατήρηση της απόδοσης των δικτύων σε μια συγκεκριμένη περιοχή και για συγκεκριμένο χρονικό διάστημα.
- Απαλλαγή των χρηστών (αν το επιθυμούν) από την χειροκίνητη εκτέλεση μετρήσεων.

- Αποφυγή επιπλέον χρέωσης (σε bandwidth) του χρήστη καθώς οι μετρήσεις θα εκτελούνται μόνο όταν ο χρήστης θα εισέρχεται στην περιοχή που έχει επιλέξει να κάνει μετρήσεις και θα σταματούν αυτόματα μόλις εξέλθει από αυτήν.

Επίσης, μια άλλη σημαντική διαφορά είναι πως το Sensorly και το OpenSignal δεν προσφέρουν τη δυνατότητα μέτρησης της απόδοσης των ασύρματων δικτύων, αλλά μόνο δυνατότητα μέτρησης της απόδοσης των δικτύων κινητής τηλεφωνίας. Αυτό έχει δύο αρνητικές συνέπειες. Η πρώτη έχει να κάνει με τον περιορισμό του χρήστη να κάνει μόνο μετρήσεις από το κινητό τηλέφωνο του και όχι από άλλες συσκευές που δεν υποστηρίζουν δίκτυα κινητής τηλεφωνίας (π.χ Tablets). Η δεύτερη αρνητική συνέπεια έχει να κάνει με τη μη παροχή πληροφορίας στους χρήστες που θέλουν να ξέρουν τις ταχύτητες του ασύρματου internet στην περιοχή τους. Τέλος άλλη μια διαφορά που κάποιος μπορεί να παρατηρήσει είναι πως το Sensorly και το OpenSignal μπορούν, σε αντίθεση με τη δική μας πλατφόρμα, να εντοπίσουν διαδικτυακά τη θέση του χρήστη και αυτό προσφέρει ως πλεονέκτημα την απαλλαγή του χρήστη από τη διαδικασία του να πάει χειροκίνητα στη περιοχή του χάρτη στην οποία βρίσκεται εκείνη τη στιγμή, ώστε να δει την πληροφορία.

Συμπερασματικά η διαδικτυακή πλατφόρμα που αναπτύξαμε στοχεύει σε τρία κύρια σημεία τα οποία δεν προσφέρουν οι υπάρχουσες εφαρμογές. Πρώτον γίνεται μια προσπάθεια, με την δημιουργία καμπάνιας, να υπάρχουν συντονισμένες μετρήσεις από τους χρήστες σε περιοχές και σε χρονικό διάστημα που θα ορίζονται από τον διαχειριστή. Οι συντονισμένες μετρήσεις θα γίνονται χωρίς να επιβαρύνουν τον χρήστη με περαιτέρω ενέργειες, αλλά παράλληλα θα διασφαλίζεται η δυνατότητα του να κάνει όποτε και όπου επιθυμεί μετρήσεις. Δεύτερον οι μετρήσεις που θα κάνουν οι χρήστες θα αφορούν όχι μόνο τις επιδόσεις των δικτύων κινητής τηλεφωνίας αλλά και τα ασύρματα δίκτυα WIFI. Τρίτον και τελευταίον θα παρουσιάζονται στον χρήστη πλήθος στατιστικών πληροφοριών για τις επιδόσεις των δικτύων στον χάρτη και σε γραφήματα. Άρα μπορούμε να πούμε πως η διαδικτυακή πλατφόρμα MyWIFISpeed προσφέρει ότι και οι υπάρχουσες πλατφόρμες, αλλά και κάποιες επιπλέον σημαντικές επεκτάσεις.

5 Λειτουργικές Απαιτήσεις Εφαρμογής

Σε αυτό το κεφάλαιο παρουσιάζουμε το περιβάλλον στο οποίο έγινε η ανάπτυξη της διαδικτυακής πλατφόρμας παραθέτοντας όλα τα εργαλεία που απαιτήθηκαν για να καταστήσουν δυνατή τη λειτουργία της. Δηλαδή η παρούσα ενότητα αποτελεί ένα σύντομο οδηγό χρήσης των εργαλείων που χρησιμοποιήθηκαν για να στηθεί το περιβάλλον ανάπτυξης της εφαρμογής μας. Επομένως, ο τελικός στόχος της ενότητας δεν είναι η παράθεση της υλοποίησης της διαδικτυακής πλατφόρμας (αυτό θα γίνει στο κεφάλαιο 6), αλλά τι χρειαζόμαστε και τι χρησιμοποιήσαμε για να την ολοκληρώσουμε και να επιτύχουμε τις επιθυμητές λειτουργίες.

5.1 GlassFish

Ο GlassFish^[60] είναι ένας εξυπηρετητής εφαρμογών (Application Server), δηλαδή προσφέρει την δυνατότητα ανάπτυξης διαδικτυακών εφαρμογών, αλλά και την δυνατότητα να τις εξυπηρετεί κατά την εκτέλεση τους. Επίσης ο GlassFish είναι εφαρμογή ανοιχτού κώδικα και αρχικά αναπτύχθηκε από την Sun Microsystems, ενώ σήμερα χρηματοδοτείται από την Oracle. Η Sun Microsystems διέθεσε την πρώτη έκδοση του GlassFish το 2006 και από τότε μέχρι σήμερα έχουν ακολουθήσει πολλές εκδόσεις με τελευταία την 4.1 από το 2014. Επιπλέον ο GlassFish υποστηρίζει εφαρμογές που έχουν αναπτυχθεί σε τεχνολογίες της Java όπως Enterprise JavaBeans, JPA, JavaServer Faces, Java PrimeFacs, JMS, RMI, JavaServer Pages, servlets κ.α. Αυτός είναι και ο λόγος που τον προτιμήσαμε, σε σχέση με άλλους εξυπηρετητές (π.χ Apache Tomcat), να τον εγκαταστήσουμε και να τον χρησιμοποιήσουμε ως εξυπηρετητή για την εκτέλεση της διαδικτυακής εφαρμογής μας.

Βασική προϋπόθεση για να χρησιμοποιήσουμε τον GlassFish είναι να υπάρχει εγκαταστημένη η Java στο περιβάλλον μας και στη συνέχεια να τον κατεβάσουμε και να τον εγκαταστήσουμε στον υπολογιστή που θέλουμε να λειτουργεί ως εξυπηρετητής. Υπάρχουν όμως πολλές δυνατές επιλογές για τον τρόπο εγκατάστασης του ανάλογα με το τι προτιμάμε και το περιβάλλον που θέλουμε να τον εγκαταστήσουμε. Όλους τους δυνατούς τρόπους εγκατάστασης μπορούμε να τους βρούμε στο εγχειρίδιο εγκατάστα-

σης^[61] του GlassFish. Εμείς στη δική μας περίπτωση και επειδή το περιβάλλον που θέλαμε να τον εγκαταστήσουμε ήταν περιβάλλον Linux, ακολουθήσαμε τα παρακάτω βήματα αρχίζοντας με την εγκατάσταση της Java (αν υπάρχει η Java εγκατεστημένη τότε μπορούμε να παραλείψουμε τα βήματα 2 έως 4).

1. Ενημέρωση συστήματος (για να σιγουρευτούμε πως έχουμε τις τελευταίες εκδόσεις)

- `sudo apt-get update`

2. Εγκατάσταση python-software-properties

- `sudo apt-get install python-software-properties`

- `sudo add-apt-repository ppa:webupd8team/java`

3. Εγκατάσταση της Java (στην περίπτωση μας της έκδοσης 7)

- `sudo apt-get install oracle-java7-installer`

4. Επιβεβαίωση εγκατάστασης

- `java -version`

πρέπει να δούμε κάτι αντίστοιχο:

- `java version "1.7.0_51"`

`Java(TM) SE Runtime Environment (build 1.7.0_51-b13)`

`Java HotSpot(TM) 64-Bit Server VM (build 24.51-b03, mixed mode)`

5. Λήψη zip αρχείου του GlassFish

- `wget download.java.net/glassfish/4.0/release/glassfish-4.0.zip`

6. Εγκατάσταση της εφαρμογής unzip για να αποσυμπιέσουμε το zip αρχείο του Glassfish (αν υπάρχει ήδη εγκατεστημένο μπορούμε να παραλείψουμε αυτό το βήμα)

- `apt-get install unzip`

7. Δημιουργία του καταλόγου opt και αποσυμπιέζουμε σε αυτό το αρχείο zip του GlassFish

- `unzip glassfish-4.0.zip -d /opt`

8. Η εγκατάσταση ολοκληρώθηκε οπότε μπορούμε να ξεκινήσουμε τον εξυπηρετητή

- `asadmin start-domain`

αν όλα πήγαν καλά θα πρέπει να δούμε σαν απάντηση

- `Waiting for domain1 to start`

Successfully started the domain : domain1

domain Location: /opt/glassfish4/glassfish/domains/domain1

Log File: /opt/glassfish4/glassfish/domains/domain1/logs/server.log

dmin Port: 4848

command start-domain executed successfully.

9. Για να μπορέσουμε να επισκεφτούμε απομακρυσμένα τη σελίδα διαχείρισης του GlassFish στη διεύθυνση «MyServerIP:4848» θα πρέπει να ενεργοποιήσουμε το secure admin (ασφαλής διαχείριση)

- `asadmin enable-secure-admin`

πρέπει να δούμε

- `Enter admin user name> admin`

`Enter admin password for user "admin">`

`You must restart all running servers for the change in secure admin to take effect.`

`Command enable-secure-admin executed successfully.`

10. Πρέπει να γίνει επανεκκίνηση του domain για να εφαρμοστεί η ενεργοποίηση του secure admin

- `asadmin restart-domain`

πρέπει να δούμε

- `Successfully restarted the domain`

`Command restart-domain executed successfully.`

Μετά από αυτά τα βήματα μπορούμε να επισκεφτούμε (και απομακρυσμένα) τη σελίδα διαχείρισης του GlassFish στη διεύθυνση “MyServerIP:4848” και να το παραμετροποιήσουμε. Εκτός όμως από την εγκατάσταση και την παραμετροποίηση του GlassFish, άλλη μια βασική και συχνή ανάγκη είναι η εκτέλεση (deploy) μιας διαδικτυακής εφαρμογής από αυτόν. Για να κάνουμε deploy την εφαρμογή μας πρέπει αρχικά να έχουμε διαθέσιμο το war^[62] αρχείο της και να ακολουθήσουμε τα εξής βήματα

11. Εφόσον βρισκόμαστε στο κατάλογο που έχουμε τοποθετήσει το αρχείο war εκτελούμε

- `asadmin deploy sample-dir/hello.war`

πρέπει να δούμε

- `Enter admin user name> admin`

```
Enter admin password for user "admin">
```

```
Application deployed with name hello.
```

```
Command deploy executed successfully.
```

Συμπερασματικά με την εκτέλεση όλων αυτών των βημάτων έχουμε καταφέρει να εγκαταστήσουμε τον GlassFish και να είναι διαθέσιμος ώστε να ανεβάζουμε σε αυτόν τις διαδικτυακές εφαρμογές μας προς εκτέλεση. Επίσης είδαμε τον τρόπο με τον οποίο ανεβάζουμε μια διαδικτυακή εφαρμογή στον GlassFish και επιπλέον πως μπορούμε να επιτρέψουμε την απομακρυσμένη σύνδεση στη σελίδα διαχείρισης και σε όλες τις εφαρμογές που έχουμε κάνει deploy.

5.2 phpMyAdmin

Η phpMyadmin είναι μια διαδικτυακή εφαρμογή ανοιχτού κώδικα γραμμένο στη γλώσσα προγραμματισμού PHP, που προορίζεται για τη διαχείριση της MySQL βάσης δεδομένων στο διαδίκτυο. Επίσης υποστηρίζει ένα ευρύ φάσμα λειτουργιών για τις βάσεις δεδομένων MySQL, MariaDB και Drizzle. Οι πιο συνηθισμένες και συχνά χρησιμοποιούμενες λειτουργίες της (διαχείριση βάσης δεδομένων, πινάκων, στηλών, σχέσεων, χρηστών, δικαιωμάτων κτλ.) μπορούν να εκτελούνται μέσω της διεπαφής χρήστη, από την οποία μπορούμε να εκτελέσουμε και SQL ερωτήματα. Πληροφοριακά η phpMyadmin αναπτύχθηκε πρώτη φορά το 1998 από τον Tobias Ratschiller και την ίδια χρονιά έγινε η διάθεση της έκδοσης 0.9.0. Σήμερα βρισκόμαστε στην έκδοση 4.2.0 από το 2014. Το όνομα της προέρχεται από τη γλώσσα προγραμματισμού php με την οποία έχει γίνει η ανάπτυξη της, τη βάση δεδομένων MySQL την οποία διαχειρίζεται και τη διαχείριση (administration) ως η βασική λειτουργία της.

Για να χρησιμοποιήσουμε την phpMyAdmin θα πρέπει να έχουμε εγκατεστημένο στο σύστημα μας έναν εξυπηρετητή διαδικτύου (web Server) όπως οι Apache, nginx, IIS (εμείς εγκαταστήσαμε τον Apache). Επίσης πρέπει να έχουμε εγκαταστήσει την PHP 5.3.0 ή νεότερη έκδοση και μια βάση δεδομένων από τις MySQL 5.5 ή νεότερη, MariaDB 5.5 ή νεότερη και Drizzle (εμείς έχουμε εγκαταστήσει την MySQL). Την phpMyAdmin μπορούμε να την εγκαταστήσουμε στα windows αλλά και στα linux. Εμείς στο σημείο αυτό θα παραθέσουμε τα βήματα εγκατάστασης στο linux. Για τα βήματα εγκατάστασης στα windows μπορείτε να απευθυνθείτε στον εξής οδηγό εγκατάστασης^[63]. Πριν εγκαταστήσουμε όμως την phpMyAdmin στο linux σύστημα μας θα πρέπει όπως προαναφέραμε να εγκαταστήσουμε πρώτα την ομάδα εφαρμογών LAMP

(Linux, Apache, MySQL, PHP) ακολουθώντας τα εξής βήματα (αν υπάρχουν ήδη εγκατεστημένα στο σύστημα μπορούμε να παραλείψουμε αυτά τα βήματα)

1. Εγκατάσταση του Apache

- `sudo apt-get install apache2`

για να ελέγξουμε αν η εγκατάσταση ολοκληρώθηκε με επιτυχία μπορούμε να πλοηγηθούμε μέσω του browser μας στη διεύθυνση «http://myIP» ή αν είμαστε τοπικά στην διεύθυνση «http://localhost» και το αποτέλεσμα που πρέπει να δούμε είναι παρόμοιο με αυτό που φαίνεται στην Εικόνα 21



Εικόνα 21: Μήνυμα επιτυχής εγκατάστασης του Apache web server

2. Εγκατάσταση MySQL

- `sudo apt-get install mysql-server libapache2-mod-auth-mysql php5-mysql`

κατά την εγκατάσταση θα κληθούμε να ορίσουμε τον κωδικό του διαχειριστή, αν δεν το κάνουμε κατά την εγκατάσταση θα πρέπει να το κάνουμε αργότερα μέσα από την MySQL. Εφόσον έχει ολοκληρωθεί η εγκατάσταση θα πρέπει να την ενεργοποιήσουμε με την εξής εντολή

- `sudo mysql_install_db`

Για να ολοκληρωθεί η εγκατάσταση θα πρέπει να τρέξουμε το MySQL script εγκατάστασης

- `sudo /usr/bin/mysql_secure_installation`

Αφού τρέξουμε αυτή την εντολή θα μας ζητηθεί να εισάγουμε τον υφιστάμενο κωδικό. Στο σημείο αυτό μπορούμε να εισάγουμε τον κωδικό μας. Στη συνέχεια θα μας ζητηθεί να αλλάξουμε τον κωδικό μας, εμείς επιλέξαμε όχι. Τέλος θα μας ζητηθούν κάποιες επιλογές, εμείς επιλέξαμε σε όλες ναι.

3. Εγκατάσταση PHP

- `sudo apt-get install php5 libapache2-mod-php5 php5-mcrypt`

Μετά την εκτέλεση της εντολής θα μας ζητηθεί δύο φορές η άδεια μας και στις δύο φορές επιλέγουμε ναι. Επίσης θα είναι χρήσιμο να εισάγουμε την php στον κατάλογο index με την εξής εντολή

- `sudo nano /etc/apache2/mods-enabled/dir.conf`

και γράφοντας στην αρχή του αρχείου τα εξής

- `<IfModule mod_dir.c> DirectoryIndex index.php index.html index.cgi index.pl index.php index.xhtml index.htm </IfModule>`

Επιπλέον η php έχει ένα σύνολο από χρήσιμες βιβλιοθήκες και μπορούμε να τις δούμε με την εντολή

- `apt-cache search php5-`

και μπορούμε να εγκαταστήσουμε τις βιβλιοθήκες που θέλουμε (εμείς τις εγκαταστήσαμε όλες) με την εντολή

- `sudo apt-get install name of the module`

αν θέλουμε να εγκαταστήσουμε περισσότερες από μια βιβλιοθήκες ταυτόχρονα μπορούμε στην τελευταία εντολή να παραθέσουμε τα ονόματα τους χωρισμένα μεταξύ τους με κενά.

Αφού έχουμε εγκαταστήσει την ομάδα εφαρμογών LAMP είμαστε σε θέση να εγκαταστήσουμε την phpMyAdmin ακολουθώντας τα παρακάτω βήματα:

1. Εντολή εγκατάστασης

- `sudo apt-get install phpmyadmin apache2-utils`

κατά την εγκατάσταση η phpMyAdmin θα μας ζητήσει κάποιες ρυθμίσεις, για τις ρυθμίσεις αυτές ακολουθούμε τα εξής:

- Επιλέγουμε τον Apache2 ως εξυπηρετητή
- Επιλέγουμε ναι όταν ερωτηθούμε για την διαμόρφωση phpMyAdmin με το dbconfig-common
- Εισάγουμε τον κωδικό της MySQL όταν μας ζητηθεί
- Εισάγουμε τον κωδικό που θέλουμε να χρησιμοποιήσουμε για να συνδεθούμε στην phpMyAdmin.

2. Εφόσον η εγκατάσταση έχει ολοκληρωθεί, ανοίγουμε το αρχείο παραμέτρων του Apache.

- `sudo nano /etc/apache2/apache2.conf`

3. Εισάγουμε στο αρχείο, το αρχείο παραμέτρων της phpMyAdmin

- `Include /etc/phpmyadmin/apache.conf`
4. Επανεκκινούμε τον Apache
- `sudo service apache2 restart`
5. Επισκεπτόμαστε την σελίδα εισόδου της phpMyAdmin πληκτρολογώντας στον browser την διεύθυνση myIP/phpmyadmin ή αν είμαστε τοπικά την διεύθυνση localhost/phpmyadmin και θα δούμε το αποτέλεσμα που φαίνεται και στην Εικόνα 22. Στα πεδία Username και Password εισάγουμε τους κωδικούς που ορίσαμε προηγουμένως ή αν δεν ορίσαμε κανένα οι εξορισμού κωδικοί είναι user: admin, pass: admin.

Εικόνα 22: Σελίδα εισόδου της phpMyAdmin

Στο σημείο αυτό και αφού συνδεθούμε μπορούμε να δημιουργήσουμε και να διαχειριστούμε τις βάσεις και τους πίνακες μας.

5.3 NetBeans

Το NetBeans^[59] IDE (integrated development environment) είναι μια εφαρμογή που έχει αναπτυχθεί με τη γλώσσα προγραμματισμού Java και χρησιμοποιείται κυρίως για την ανάπτυξη εφαρμογών σε Java. Μπορεί όμως να υποστηρίζει και την ανάπτυξη εφαρμογών σε PHP, C/C++ και HTML5. Επιπλέον το NetBeans είναι ανεξάρτητο λειτουργικού συστήματος και μπορεί να εγκατασταθεί σε Microsoft Windows, Mac OS X, Linux, Solaris και σε άλλες πλατφόρμες οι οποίες υποστηρίζουν το JVM (Java Virtual Machine). Η ανάπτυξη του NetBeans ξεκίνησε το 1996 από ομάδα φοιτητών και το 1999 εξαγοράστηκε από την Sun Microsystems η οποία με τη σειρά της εξαγοράστηκε το 2010 από την Oracle. Η τελευταία σήμερα υποστηρίζει την ανάπτυξη του NetBeans και έχει κυκλοφορήσει από το 2014 την έκδοση NetBeans IDE 8.0.

Για να χρησιμοποιήσουμε το NetBeans θα πρέπει αρχικά να το κατεβάσουμε από την επίσημη ιστοσελίδα επιλέγοντας την έκδοση που χρειαζόμαστε μεταξύ των:

- Java SE
- Java EE
- C/C++
- PHP
- ALL

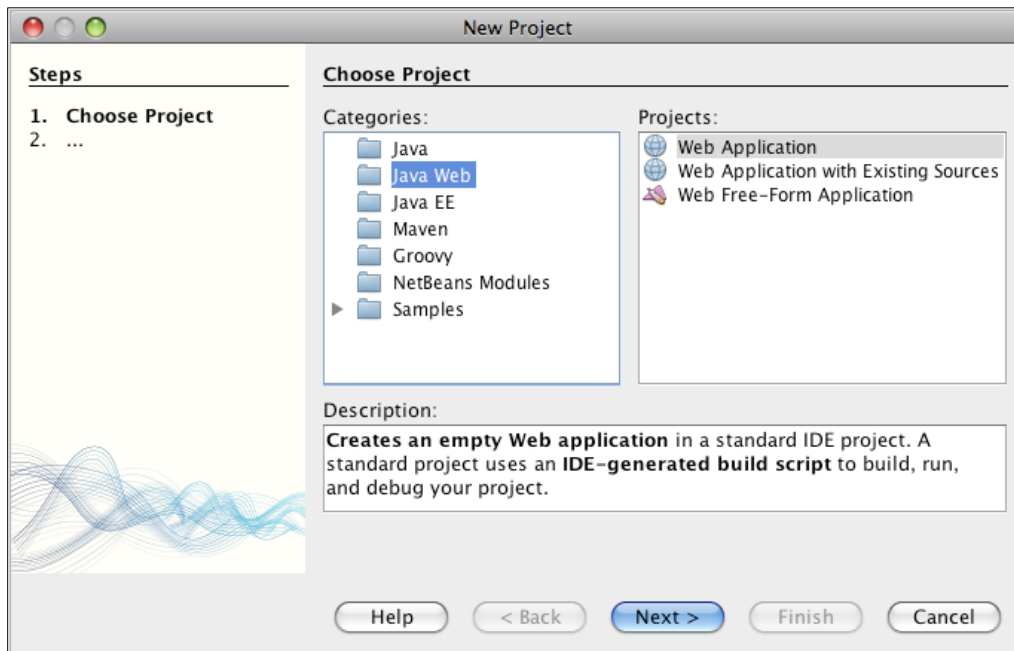
Εμείς έχουμε κατεβάσει την έκδοση all που τα συμπεριλαμβάνει όλα. Στη συνέχεια πρέπει να το εγκαταστήσουμε στον υπολογιστή μας ακολουθώντας τα παρακάτω βήματα για τα λειτουργικά Microsoft Windows και Linux:

1. Αφού έχει ολοκληρωθεί η λήψη, πρέπει να τρέξουμε το εκτελέσιμο αρχείο με επέκταση *.exe για τα windows και *.sh για τα linux. Συγκεκριμένα για τα linux το αρχείο εγκατάστασης θα πρέπει πρώτα να γίνει εκτελέσιμο με την εξής εντολή: `chmod +x <installer-file-name>`.
2. Αν έχει γίνει λήψη των εκδόσεων all ή Java EE, τότε υπάρχει η δυνατότητα να παραγοντοποιήσουμε την εγκατάσταση, κάνοντας τα εξής:
 1. Επιλέγουμε το Customize (προσαρμογή)
 2. Μέσα στο παράθυρο διαλόγου που εμφανίζεται επιλέγουμε τα εργαλεία που επιθυμούμε
 3. Επιλέγουμε OK
3. Στη σελίδα καλωσορίσματος επιλέγουμε επόμενο.

4. Στη σελίδα άδειας χρήσης διαβάζουμε την άδεια χρήσης και επιλέγουμε αποδοχή και πατάμε next
5. Στη σελίδα άδειας χρήσης του JUnit, αποφασίζουμε αν χρειαζόμαστε το JUnit (εμείς αποφασίσαμε ναι), κάνουμε την κατάλληλη επιλογή και επιλέγουμε επόμενο.
6. Στη σελίδα εγκατάστασης του NetBeans κάνουμε τα εξής:
 1. Αποδεχόμαστε την εξορισμού διεύθυνση για τον κατάλογο εγκατάστασης ή ορίζουμε μια καινούργια.
 2. Αποδεχόμαστε την εξορισμού εγκατάσταση του JDK ή επιλέγουμε μια διαφορετική από τη λίστα. Αν δεν υπάρχει διαθέσιμη καμία JDK εγκατάσταση τότε μπορεί να μην βρίσκεται στη σωστή διεύθυνση όποτε την ορίζουμε ή δεν υπάρχει το JDK εγκατεστημένο στο σύστημα μας όποτε την εγκαθιστούμε
7. Αν η σελίδα εγκατάστασης του GlassFish ανοίξει, αποδεχόμαστε την εξορισμού διεύθυνση εγκατάστασης ή ορίζουμε καινούργια.
8. Αν η σελίδα εγκατάστασης του Apache Tomcat ανοίξει, αποδεχόμαστε την εξορισμού διεύθυνση εγκατάστασης ή ορίζουμε καινούργια
9. Στη σελίδα περίληψης ελέγχουμε αν υπάρχει αρκετός διαθέσιμος αποθηκευτικός χώρος και επιλέγουμε Install.
10. Στη σελίδα ολοκλήρωσης επιλέγουμε τέλος.

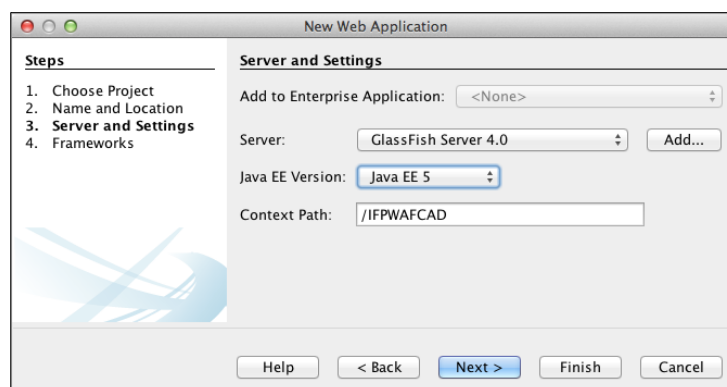
Αφού έχει ολοκληρωθεί η εγκατάσταση μπορούμε πλέον να συνεχίσουμε και να δημιουργήσουμε projects. Για τη δημιουργία projects και ειδικότερα για τη δημιουργία ενός Java web application project θα πρέπει να ακολουθήσουμε τα εξής βήματα:

1. Ανοίγουμε την εφαρμογή NetBeans από το κατάλληλο εικονίδιο.
2. Στη συνέχεια στο κεντρικό μενού επιλέγουμε File > New Project. Στο αναδυόμενο παράθυρο (Εικόνα 23) πρέπει να επιλέξουμε τη κατηγορία Java Web και μετά το Java Web Application. Τέλος επιλέγουμε next.

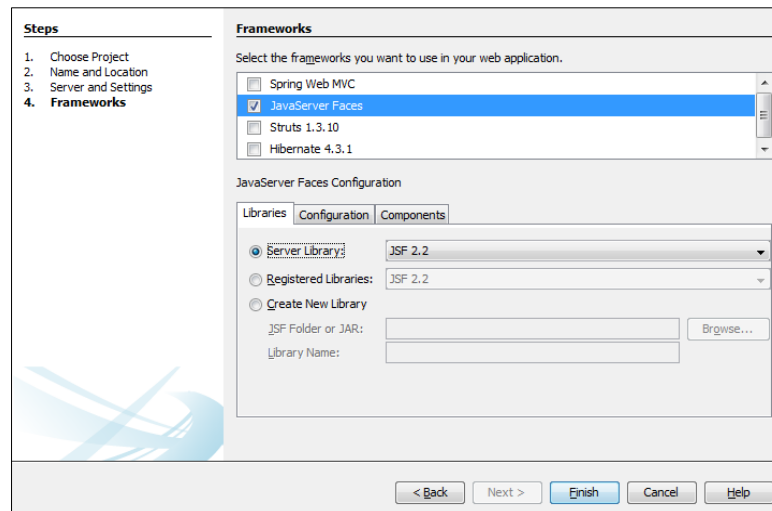


Εικόνα 23: Δημιουργία νέου Java Web Application

3. Στο επόμενο παράθυρο πρέπει να επιλέξουμε όνομα για το project και την τοποθεσία που θα αποθηκευτεί στον υπολογιστή μας. Επιλέγουμε next
4. Στο παράθυρο Server and Settings (Εικόνα 24) επιλέγουμε τον GlassFish σαν τον Server που θα χρησιμοποιηθεί για να εκτελέσει την εφαρμογή.
5. Στο πεδίο Java EE Version (Εικόνα 24) επιλέγουμε την έκδοση της Java με την οποία θέλουμε να αναπτύξουμε την εφαρμογή (εμείς επιλέξαμε την έκδοση 6).
6. Αν θέλουμε να ολοκληρώσουμε την διαδικασία χωρίς να προσθέσουμε κάποιο Framework τότε επιλέγουμε τέλος. Αν θέλουμε να προσθέσουμε κάποιο Framework (εμείς προσθέσαμε) τότε επιλέγουμε επόμενο.
7. Στο παράθυρο Frameworks κάνουμε τις κατάλληλες επιλογές όπως στην Εικόνα 25 και επιλέγουμε τέλος.



Εικόνα 24: Παράθυρο Server and Settings



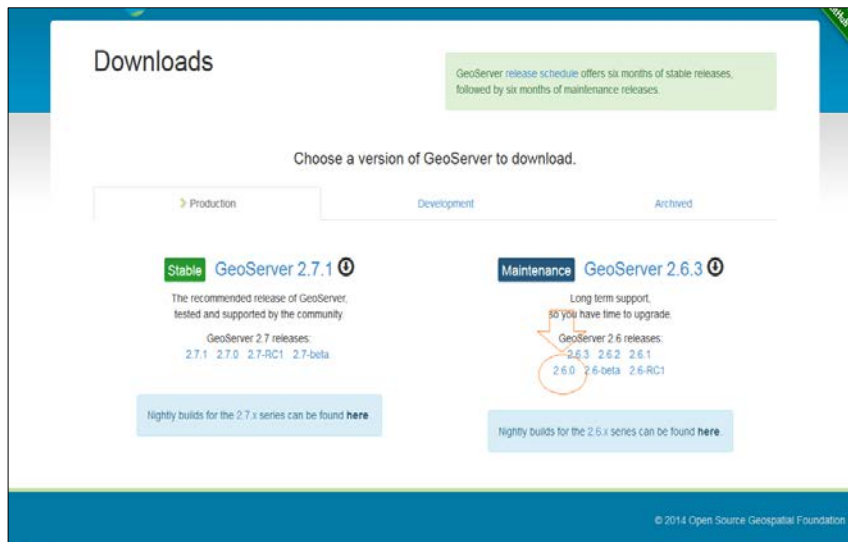
Εικόνα 25: Εισαγωγή του Framework JavaServer Faces

Στο σημείο αυτό το project έχει δημιουργηθεί και μπορούμε να αρχίσουμε να αναπτύσσουμε τον κώδικα της εφαρμογής.

5.4 GeoServer

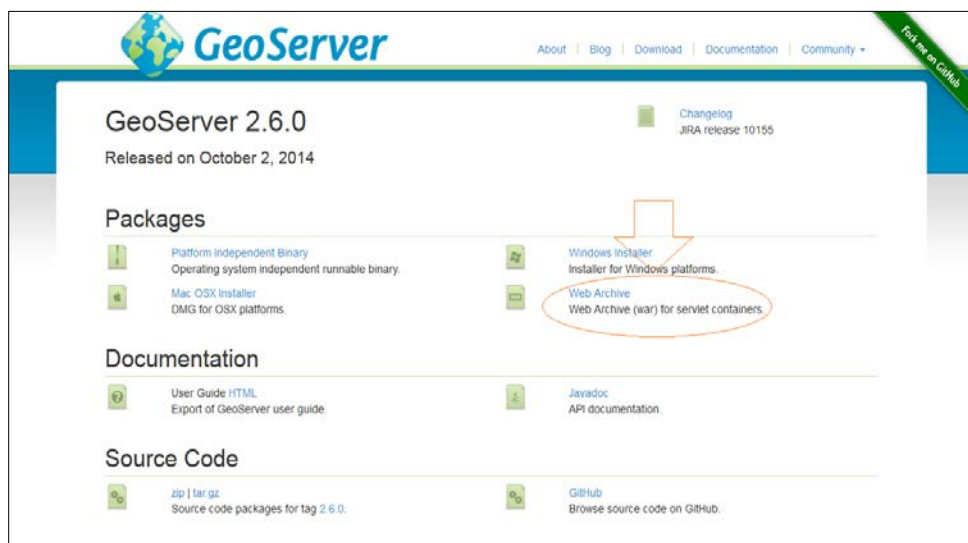
Για τον GeoServer είδαμε στο κεφάλαιο 3.1 τι είναι, τι κάνει και για ποιο λόγο τον χρησιμοποιούμε εμείς στην εργασία μας. Στην παρούσα ενότητα αυτό που θα αναφέρουμε για τον GeoServer είναι ο τρόπος με τον οποίο μπορούμε να τον χρησιμοποιήσουμε. Δηλαδή θα δούμε πως γίνεται η εγκατάσταση του, πως δημιουργούμε τα layers και πως τα styles. Επομένως τον Geoserver μπορούμε να τον εγκαταστήσουμε ως εφαρμογή στο λειτουργικό μας σύστημα ή ως διαδικτυακή εφαρμογή σε έναν εξυπηρετητή εφαρμογών όπως ο Apache. Για την εγκατάσταση στα διάφορα λειτουργικά συστήματα μπορούμε να ανατρέξουμε στις οδηγίες^[64] που βρίσκονται στην επίσημη ιστοσελίδα. Για την εισαγωγή του GeoServer στον Apache ακολουθούμε τα παρακάτω βήματα:

1. Κατεβάζουμε τον GeoServer από την ιστοσελίδα geoserver.org/download/ επιλέγοντας την έκδοση που μας ενδιαφέρει, εμείς έχουμε επιλέξει την έκδοση 2.6.0 (**Error! Reference source not found.**) και στη συνέχεια επιλέγουμε το war αρχείο (**Error! Reference source not found.**).

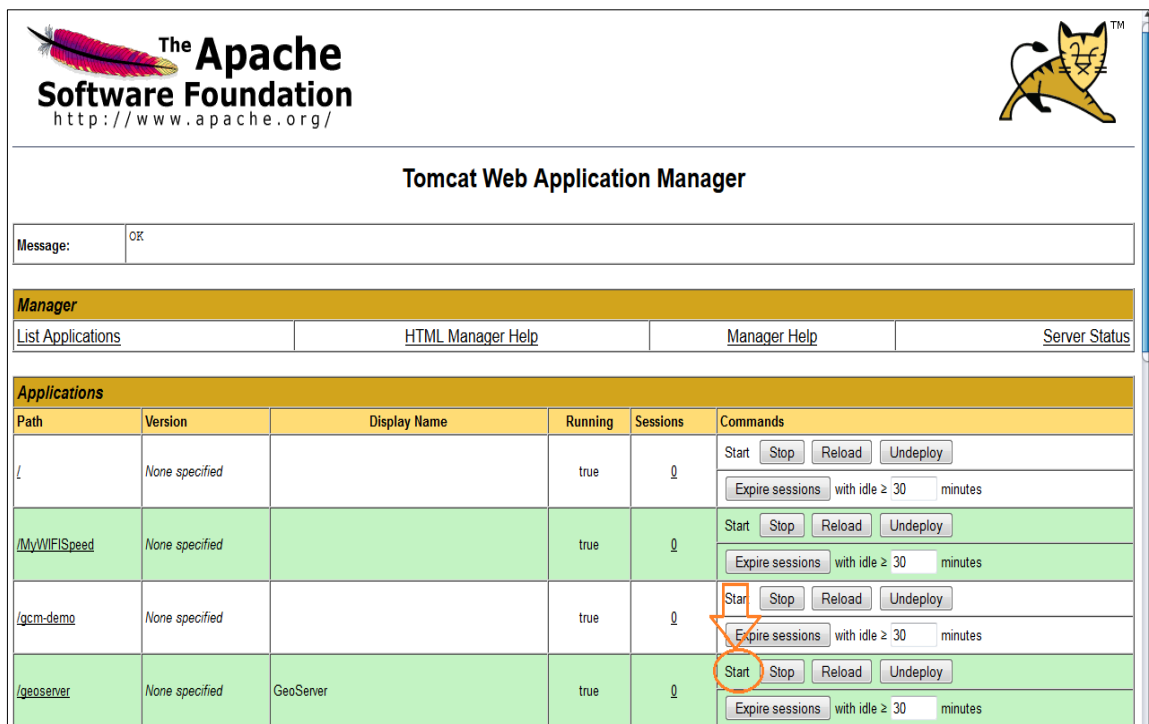


Εικόνα 26: GeoServer έκδοση λήψης

2. Εφόσον ολοκληρωθεί η λήψη αποσυμπιέζουμε το αρχείο
3. Αντιγράφουμε το geoserver.war αρχείο στον κατάλογο webapps του apache
4. Κάνουμε επανεκκίνηση τον Apache
5. Πηγαίνουμε στη σελίδα διαχείρισης του apache στην διεύθυνση `myIP:portNumber/manager/html`, εισάγουμε τους προσωπικούς κωδικούς εισόδου και στην κεντρική σελίδα διαχείρισης εντοπίζουμε την εφαρμογή Geoserver και την εκκινούμε (Εικόνα 28).
6. Τέλος μπορούμε να συνδεθούμε στην σελίδα διαχείρισης του GeoServer μέσω της διεύθυνσης `http://myIP:portNumber/geoserver` και να συνδεθούμε με τους κωδικούς μας (Εικόνα 29).

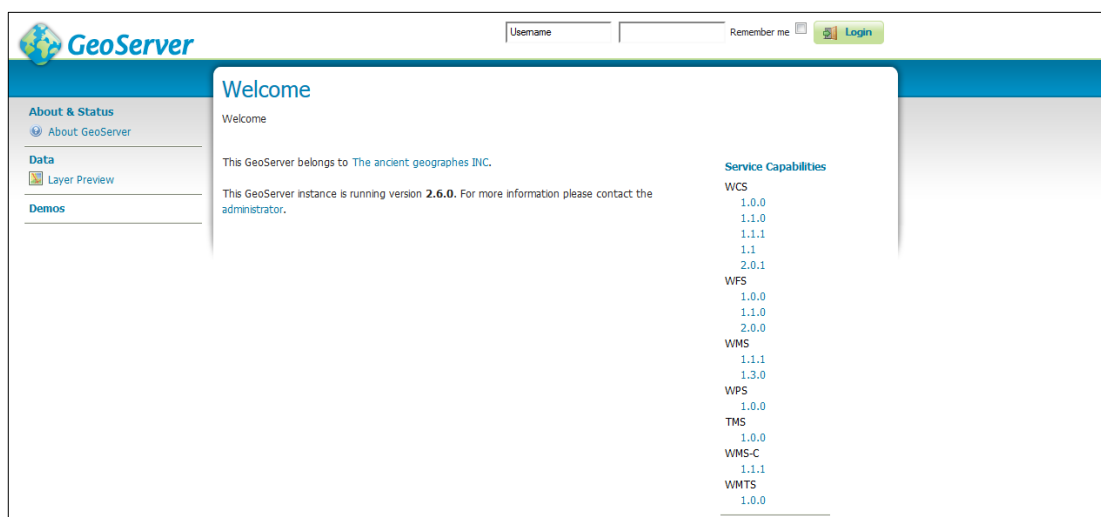


Εικόνα 27: GeoServer λήψη war αρχείου



Εικόνα 28: Apache έναρξη GeoServer

Εφόσον συνδεθούμε στην σελίδα διαχείρισης μπορούμε πλέον να δημιουργήσουμε τη σύνδεση με τη βάση δεδομένων, τα layers και τα styles που επιθυμούμε για την απεικόνιση της πληροφορίας. Για την δημιουργία των παραπάνω εμείς ακολουθήσαμε και προτείνουμε τα βήματα που παρατίθενται αναλυτικά στο κεφάλαιο 4.7 στην πτυχιακή εργασία του Νικόλαου Κορωνιώτη^[65].



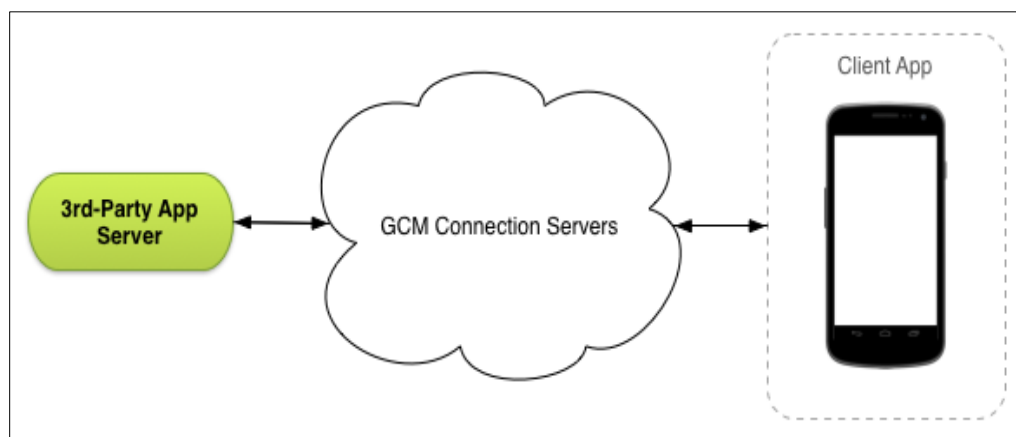
Εικόνα 29: Σελίδα Εισόδου GeoServer

5.5 Google Cloud Messaging (GCM)

Το Google Cloud Messaging^[25] (GCM) είναι μια δωρεάν υπηρεσία της google η οποία προσφέρει στους προγραμματιστές τη δυνατότητα αποστολής και λήψης μηνυμάτων μεταξύ εξυπηρετητή και πελάτη. Η αποστολή των μηνυμάτων μπορεί να γίνει από τον εξυπηρετητή προς τον πελάτη, αλλά και από τον πελάτη προς τον εξυπηρετητή παρακάμπτοντας τον παραδοσιακό τρόπο ανταλλαγής μηνυμάτων που λάμβανε χώρα μόνο μετά από αιτήσεις (request) του πελάτη. Ένα παράδειγμα εφαρμογής της GCM υπηρεσίας είναι η πληροφόρηση της εφαρμογής του πελάτη από τον εξυπηρετητή για την ύπαρξη νέων πληροφοριών για αυτήν, όπως στην περίπτωση μιας ειδοποίησης ενός νέου email, χωρίς να έχει προηγηθεί αίτημα από τον πελάτη. Η δυνατότητα αυτή είναι και ο λόγος που προτιμήσαμε τη χρήση της GCM υπηρεσίας για την αποστολή της καμπάνιας που θα δημιουργεί ο διαχειριστής, από την διαδικτυακή εφαρμογή στις κινητές συσκευές.

Για να μπορέσουμε όμως να κάνουμε χρήση της υπηρεσίας GCM θα πρέπει να υλοποιήσουμε την αρχιτεκτονική που φαίνεται στην Εικόνα 30 η οποία απαιτεί την ανάπτυξη μιας εφαρμογής που θα έχει τον ρόλο του εξυπηρετητή και θα επικοινωνεί με την GCM υπηρεσία μέσω των HTTP ή XMPP πρωτοκόλλων και μιας εφαρμογής πελάτη. Η επικοινωνία των τριών αυτών στοιχείων (GCM υπηρεσίας, Εξυπηρετητή, Εφαρμογή πελάτη) γίνεται ως εξής:

- Οι GCM εξυπηρετητές σύνδεσης (GCM Connection Servers) της google λαμβάνουν μηνύματα από τον εξυπηρετητή μας και τα στέλνουν στην εφαρμογή πελάτη.



Εικόνα 30: Αρχιτεκτονική υπηρεσίας GCM (πηγή: google.com)

- Στον εξυπηρετητή μας (App Server) υλοποιούμε αρχικά τα πρωτόκολλα HTTP ή XMPP για την επικοινωνία με την GCM υπηρεσία. Στη συνέχεια στέλνουμε τα μηνύματα στους GCM εξυπηρετητές σύνδεσης, οι οποίοι με τη σειρά τους τα αποθηκεύουν σε ουρά και τα στέλνουν στην εφαρμογή πελάτη (Client App).
- Η εφαρμογή πελάτη πρέπει να είναι μια εφαρμογή υλοποιημένη κατά τέτοιο τρόπο ώστε να υποστηρίζει τα GCM μηνύματα και να έχει γίνει η εγγραφή της στην GCM υπηρεσία ώστε να λάβει ένα μοναδικό κωδικό αναγνώρισης.

Επομένως τα βασικά που χρειαζόμαστε για την υλοποίηση και την χρήση της GCM υπηρεσίας χωρίζονται σε δύο κατηγορίες τα στοιχεία (Components) και τα διαπιστευτήρια (Credentials) όπως μας δείχνει και ο Πίνακας 3.

Τεχνικά, στην παρούσα εργασία για να χρησιμοποιήσουμε την GCM υπηρεσία χρειάστηκε να υλοποιήσουμε τον εξυπηρετητή που θα στέλνει τις καμπάνιες και τον πελάτη που θα τις δέχεται. Και οι δύο περιπτώσεις προσφέρονται από την google σχετικά έτοιμες και το μόνο που χρειάζεται να κάνουμε είναι να τις εφαρμόσουμε στο δικό μας περιβάλλον και να τις παραμετροποιήσουμε κατάλληλα. Εμείς όμως στην παρούσα ενότητα θα αρκεστούμε στο να αναφερθούμε μόνο στον τρόπο εφαρμογής του εξυπηρετητή εφαρμογών στο δικό μας περιβάλλον και την παραμετροποίηση του σύμφωνα με τις δικές μας ανάγκες, καθώς η εφαρμογή πελάτη έχει υλοποιηθεί και περιγραφεί στα πλαίσια της διπλωματικής εργασίας του Μωυσιάδη Αναστάσιου^[66] στην οποία μπορούμε να ανατρέξουμε για περισσότερες πληροφορίες. Επομένως η ανάπτυξη του εξυπηρετητή εφαρμογών μας θα πρέπει γίνει με κριτήρια τις εξής δυνατότητες που θα πρέπει να προσφέρει:

- Επικοινωνία με την εφαρμογή πελάτη.

Πίνακας 3: Αναγκαία στοιχεία και διαπιστευτήρια για την υλοποίηση της GCM υπηρεσίας

Στοιχεία (Components)	
GCM Εξυπηρετητές Σύνδεσης (Connection Servers)	Οι εξυπηρετητές της google που εμπλέκονται κατά την αποστολή μηνυμάτων μεταξύ του εξυπηρετητή μας και της εφαρμογής πελάτη.

Εφαρμογή Πελάτη (Client App)	Εφαρμογή στον πελάτη για την επικοινωνία με τον εξυπηρετητή μας και που υποστηρίζει τα GCM μηνύματα.
Εξυπηρετητής (App Server)	Ο εξυπηρετητής που αναπτύσσουμε στο δικό μας περιβάλλον για να υλοποιηθεί η GCM επικοινωνία.
Διαπιστευτήρια (Credentials)	
Κωδικός Αποστολέα (Sender ID)	Ο μοναδικός κωδικός του αποστολέα τον οποίο τον αποκτά μέσω της API κονσόλας της google. Χρησιμοποιείται κατά την διαδικασία εγγραφής για να αναγνωριστεί ο εξυπηρετητής μας ως ο έχων δικαίωμα να στέλνει μηνύματα στην εφαρμογή πελάτη.
Sender Auth Token	Είναι το μοναδικό κλειδί του εξυπηρετητή που το επιτρέπει την πρόσβαση στην GCM υπηρεσία της google.
Κωδικός εφαρμογής (Application ID)	Ο μοναδικός κωδικός της εφαρμογής πελάτη που έχει κάνει εγγραφή για να λαμβάνει μηνύματα.
Registration Token	Μοναδικός κωδικός που οι GCM εξυπηρετητές αντιστοιχούν σε μια εφαρμογή πελάτη ώστε να λαμβάνει μηνύματα.

- Αποστολή κατάλληλα διαμορφωμένων αιτημάτων προς τον εξυπηρετητή σύνδεσης της υπηρεσίας GCM.
- Δυνατότητα διαχείρισης των αιτημάτων και αποστολής τους ξανά όταν είναι αναγκαίο
- Δυνατότητα αποθήκευσης του API key και του κωδικού εγγραφής του πελάτη.
- Δυνατότητα παραγωγής μοναδικών κωδικών ανά μήνυμα για να καταστήσει μοναδικό το κάθε μήνυμα που αποστέλλεται.

Ειδικότερα έχουμε την δυνατότητα να επιλέξουμε μεταξύ δύο υλοποιήσεων του εξυπηρετητή εφαρμογών, της υλοποίησης που προσφέρει αποστολή μηνυμάτων μέσω

HTTP και εκείνης που προσφέρει αποστολή μηνυμάτων μέσω XMPP^[67]. Οι διαφορές και οι ομοιότητες τους διακρίνονται ως εξής:

- Upstream/Downstream μηνύματα (μηνύματα από και προς τον εξυπηρετητή)
 - HTTP: μηνύματα μόνο από τον εξυπηρετητή προς τον πελάτη, δυνατότητα αποστολής δεδομένων με μέγεθος πάνω από 4KB.
 - XMPP: μηνύματα από τον εξυπηρετητή προς τον πελάτη, αλλά και από τον πελάτη προς τον εξυπηρετητή. Δυνατότητα αποστολής δεδομένων με μέγεθος πάνω από 4KB.
- Ασύγχρονη ή συγχρονισμένη αποστολή μηνυμάτων
 - HTTP: συγχρονισμένη αποστολή μηνυμάτων.
 - XMPP: ασύγχρονη αποστολή μηνυμάτων.
- JSON μηνύματα
 - HTTP: αποστέλλονται μέσω HTTP Post.
 - XMPP: ενσωματώνονται μέσα στο XMPP μήνυμα.
- Μήνυμα απλού κείμενου
 - HTTP: αποστέλλονται μέσω HTTP Post.
 - XMPP: δεν υποστηρίζονται.
- Πολλαπλή αποστολή μηνυμάτων σε πολλαπλές εγγεγραμμένες συσκευές
 - HTTP: υποστηρίζεται σε αποστολή μηνυμάτων JSON.
 - XMPP: δεν υποστηρίζεται.

Εμείς λαμβάνοντας υπόψη τα παραπάνω επιλέξαμε την υλοποίηση που στέλνει μηνύματα μέσω του HTTP πρωτόκολλου, καθώς μας ενδιαφέρει να στείλουμε συγχρονισμένα JSON μηνύματα σε πολλαπλές συσκευές χωρίς να λαμβάνουμε κάποια απάντηση πίσω από την εφαρμογή του πελάτη.

Στο σημείο αυτό και αφού έχουμε αναφέρει τις βασικές προϋποθέσεις που ο εξυπηρετητής εφαρμογών μας πρέπει να τηρεί μπορούμε να παραθέσουμε τα βήματα που πρέπει να ακολουθήσουμε για την υλοποίηση του.

1. Από το github.com^[68] κατεβάζουμε τα εξής αρχεία: gcm-server, samples/gcm-demo-server, and samples/gcm-demo-appengine
2. Ανοίγουμε ένα συντάκτη κειμένου (text editor) και σε αυτό ανοίγουμε το αρχείο samples/gcm-demo-server/WebContent/WEB-INF/classes/api.key. αντικαθιστούμε το υπάρχων κείμενο με το API key που έχουμε αποκτήσει από την πλατφόρμα google console^[69] (για βοήθεια σχετικά με την απόκτηση API key

από την google console μπορούμε να ανατρέξουμε στην βοήθεια σχετικά με αυτή^[70]).

3. Δημιουργούμε το war αρχείο του εξυπηρετητή. Για να δημιουργήσουμε το war αρχείο του εξυπηρετητή μπορούμε:
 - a. Να χρησιμοποιήσουμε κάποιο IDE όπως το NetBeans για να κάνουμε build τις κλάσεις που βρίσκονται στον κατάλογο samples/gcm-demo-server
 - b. Ή μέσα από ένα παράθυρο εντολών να πλοηγηθούμε στον κατάλογο samples/gcm-demo-server και να εκτελέσουμε την εντολή ant war.
4. Εφόσον έχουμε δημιουργήσει το war αρχείο του εξυπηρετητή τότε το ανεβάζουμε στον εξυπηρετητή εφαρμογών μας, εμείς το ανεβάσαμε στον GlassFish.
5. Τέλος αφού έχουμε ανεβάσει το war αρχείο στον GlassFish (ή κάποιον αντίστοιχο εξυπηρετητή εφαρμογών) μπορούμε να τρέξουμε τον GCM εξυπηρετητή μας μέσω της διεύθυνσης myIP:portNumber/gcm-demo/home.

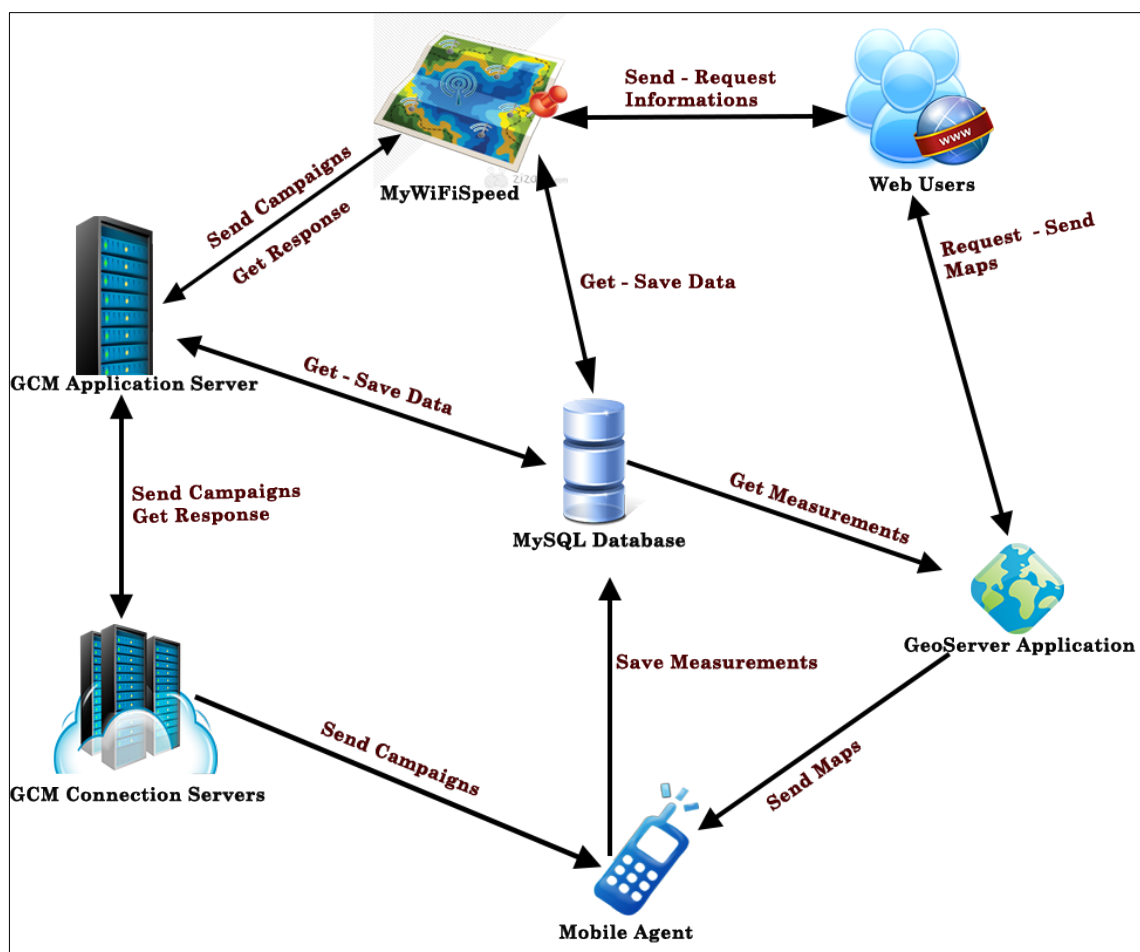
Ακολουθώντας τα παραπάνω βήματα έχουμε καταφέρει να εγκαταστήσουμε στο δικό μας περιβάλλον τον έτοιμο GCM εξυπηρετητή εφαρμογών που μας προσφέρει η google. Μέχρι αυτό το σημείο και με τα βήματα που έχουμε εκτελέσει μπορούμε να στείλουμε στοιχειώδη μηνύματα στις εφαρμογές πελάτη μας, με την προϋπόθεση ότι υπάρχει εφαρμογή πελάτη. Αυτό που απομένει είναι η αλλαγή και προσαρμογή του κώδικα, στο έτοιμο GCM εξυπηρετητή, στις δικές μας ανάγκες. Τις τροποποιήσεις που έχουν γίνει στον κώδικα έτσι ώστε να στέλνουμε τα δικά μας μηνύματα θα τις δούμε στο επόμενο κεφάλαιο όπου θα γίνει και η τεχνική ανάλυση της όλης πλατφόρμας που έχουμε αναπτύξει.

6 Παρουσίαση Εφαρμογής

Στο παρόν κεφάλαιο παρουσιάζεται:

- Σε τεχνικό επίπεδο η υλοποίηση της διαδικτυακής εφαρμογής (κεφάλαιο 6.1)
- Και ο οδηγός χρήσης της εφαρμογής (κεφάλαιο 6.2)

Πριν όμως προχωρήσουμε στα προαναφερθέντα, κρίνουμε απαραίτητο να γίνει μια ανάλυση της αρχιτεκτονικής της συνολικής πλατφόρμας (Εικόνα 31), για να καταστεί ευκολότερη η κατανόηση των λειτουργιών της και της τεχνικής ανάλυσης.



Εικόνα 31: Αρχιτεκτονική Διαδικτυακής Πλατφόρμας MyWiFiSpeed.

Συνεπώς στην αρχιτεκτονική στην Εικόνα 31 φαίνονται η ροή των δεδομένων και τα επιμέρους στοιχεία της εφαρμογής που χρησιμοποιούνται και συνεργάζονται μεταξύ τους για την παραγωγή της επιθυμητής πληροφορίας. Αναλυτικότερα, η όλη διαδικασία ξεκινάει από την κινητή συσκευή (**Mobile Agent**) η οποία εφόσον εγκαταστήσει την εφαρμογή **HuaServiceTester2**^[66], που αποτελεί σημαντικό και αναπόσπαστο κομμάτι

για την λειτουργία της διαδικτυακής εφαρμογής μας, μπορεί να συλλέξει δεδομένα για την απόδοση των ασύρματων δικτύων και να τα αποθηκεύσει στην βάση δεδομένων μας. Τα δεδομένα που συλλέγει αφορούν την ταχύτητα κατεβάσματος (download) και ανεβάσματος (upload), ταχύτητα ping προς τον εξυπηρετητή της google και τον δικό μας των ασύρματων δικτύων. Ο τρόπος συλλογής των δεδομένων από την εφαρμογή HuaServiceTester2 και ο τρόπος αποθήκευσής τους στην βάση δεδομένων μας, παρουσιάζεται αναλυτικά στην διπλωματική εργασία του Μωυσιάδη Αναστάσιου^[66].

Στη συνέχεια, δεύτερος σταθμός στη ροή των δεδομένων είναι η βάση δεδομένων MySQL (Εικόνα 31). Η βάση δεδομένων, στην αρχιτεκτονική μας, αποτελεί κομβικό σημείο στη ροή των δεδομένων καθώς τα αποθηκεύει και επιτρέπει την πρόσβαση στα υπόλοιπα στοιχεία για την ανάκτηση και την επεξεργασία τους. Από το σημείο αυτό και πέρα η ροή των δεδομένων γίνεται σε ασύγχρονες παράλληλες διαδρομές, καθώς μπορούν να σταλθούν από την βάση δεδομένων προς τον Geoserver, από και προς τον GCM application Server και από και προς την διαδικτυακή εφαρμογή MyWiFiSpeed.

Στον GeoServer (Εικόνα 31) εφόσον ανακτηθούν τα δεδομένα από την βάση δεδομένων γίνεται η επεξεργασία τους και συγκεκριμένα εφαρμόζεται σε αυτά ο αλγόριθμος σύντηξης δεδομένων Barnes Surface. Σαν επόμενο βήμα, ο GeoServer μετά από requests στέλνει τους χάρτες, που παρήχθησαν μετά την εφαρμογή του Barnes Surface αλγόριθμου και που περιλαμβάνουν την οπτική πληροφορία για την απόδοση των ασύρματων δικτύων, στους χρήστες της εφαρμογής HuaServiceTester2 και στους χρήστες της διαδικτυακής εφαρμογής MyWiFiSpeed.

Με την σειρά της η εφαρμογή MyWiFiSpeed (Εικόνα 31) έχει ως σκοπό αρχικά την λήψη των δεδομένων από την βάση δεδομένων και την επεξεργασία τους για την παραγωγή στατιστικής πληροφορίας σχετικά με την απόδοση των ασύρματων δικτύων. Στη συνέχεια την παράδοση στον χρήστη μιας διαδικτυακής διεπαφής μέσω της οποίας θα μπορεί:

1. Να κάνει εγγραφή (Sign Up) και να συνδέεται (Log In).
2. Να παρακολουθήσει την απόδοση των ασύρματων δικτύων μέσω χαρτών, που παρήχθησαν από τον GeoServer.
3. Να παρακολουθήσει στατιστικά στοιχεία για τα ασύρματα δίκτυα.
4. Να διαχειριστεί τον λογαριασμό του (αλλαγή όνομα χρήστη και κωδικού χρήστη).
5. Επιπλέον αν έχει τον ρόλο του διαχειριστή (administrator) να:

- a. Διαχειριστεί τους χρήστες.
- b. Δημιουργήσει, να διαχειριστεί και να αποστείλει, στις κινητές συσκευές, καμπάνιες για μετρήσεις σε συγκεκριμένες περιοχές.
- c. Παρακολουθήσει την απόδοση των ασύρματων δικτύων μέσω χαρτών ανά καμπάνια.
- d. Παρακολουθήσει στατιστικά στοιχεία για τα ασύρματα δίκτυα ανά καμπάνια.

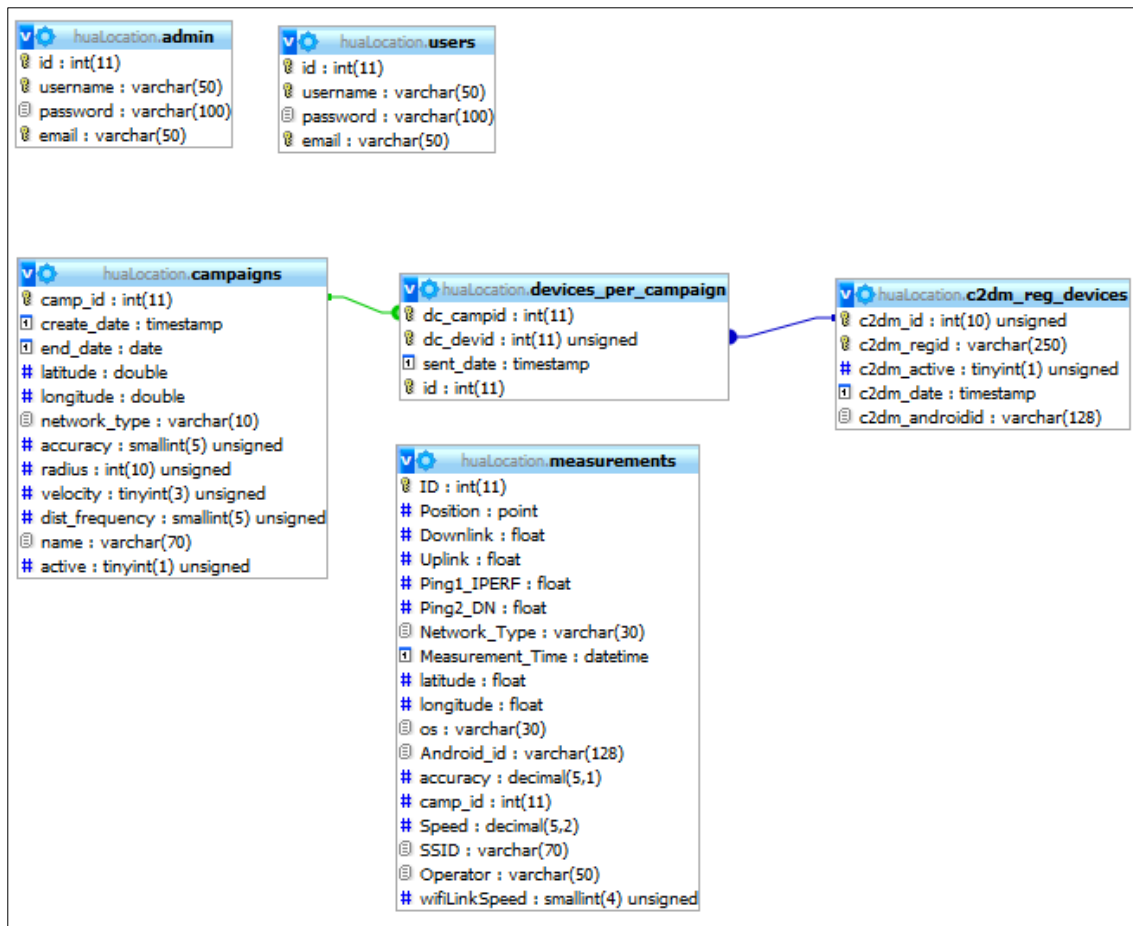
Για την επίτευξη των παραπάνω λειτουργιών η εφαρμογή MyWiFiSpeed κάνει εγγραφή δεδομένων πίσω στην βάση δεδομένων, αλλά και επικοινωνεί και με τον GCM Application Server μέσω του οποίου αποστέλλει τις καμπάνιες στην εφαρμογή HuaServiceTester2. Τέλος ο GCM Application Server, αφού επικοινωνήσει με την βάση δεδομένων για την εύρεση των εγγεγραμμένων κινητών συσκευών, αποστέλλει τα στοιχεία τους και τις καμπάνιες στους GCM Connection Servers που με την σειρά τους αναλαμβάνουν να τις στείλουν στην κινητή εφαρμογή.

Συνοπτικά, η περιγραφή αυτή εκφράζει την συνολική λειτουργία και την ροή των δεδομένων στα επιμέρους στοιχεία της αρχιτεκτονικής που παρουσιάζεται στην Εικόνα 31. Τα δύο επόμενα κεφάλαια στην ενότητα αυτή παρουσιάζουν τα σημαντικότερα σημεία του κώδικα που αναπτύχθηκε και τον οδηγό χρήσης της διαδικτυακής εφαρμογής MyWiFiSpeed.

6.1 Τεχνική Ανάλυση

6.1.1 Βάση Δεδομένων

Την τεχνική ανάλυση της διαδικτυακής πλατφόρμας MyWiFiSpeed κρίνουμε πως πρέπει να την ξεκινήσουμε αναλύοντας το κομβικότερο σημείο της πλατφόρμας μας, το “εσωτερικό” της βάσης δεδομένων. Ειδικότερα η βάση δεδομένων μας είναι μια MySQL βάση την οποία την διαχειριζόμαστε μέσω της phpMyAdmin. Η phpMyAdmin με τη σειρά της είναι εγκατεστημένη στον servlet container Apache Tomcat. Στην Εικόνα 32 φαίνεται το σύνολο των πινάκων που έχουν δημιουργηθεί στην βάση δεδομένων μας.



Εικόνα 32: Πίνακες βάσης δεδομένων.

Πίνακας “admin”

Χρησιμοποιείται για την αποθήκευση των στοιχείων των χρηστών που κατέχουν το ρόλο του διαχειριστή και περιλαμβάνει τα πεδία:

- **id:** αποτελεί τον μοναδικό κωδικό για την κάθε εγγραφή του πίνακα (θετικός αυξανόμενος ακέραιος αριθμός με μέγεθος 11 ψηφίων).
- **username:** το όνομα χρήστη του διαχειριστή (αλφαριθμητικό με μέγεθος 50 χαρακτήρων)
- **password:** ο κωδικός εισόδου του διαχειριστή (αλφαριθμητικό με μέγεθος 100 χαρακτήρων)
- **email:** ηλεκτρονικό ταχυδρομείο του διαχειριστή (αλφαριθμητικό με μέγεθος 50 χαρακτήρων)

Πίνακας “users”

Χρησιμοποιείται για την αποθήκευση των στοιχείων των απλών χρηστών και περιλαμβάνει τα ίδια πεδία και τύπο δεδομένων με τον πίνακα “admin”.

Πίνακας “measurements”

Χρησιμοποιείται για την αποθήκευση των μετρήσεων που πραγματοποιούνται από τις κινητές συσκευές μέσω της εφαρμογής HuaServiceTester2 και περιλαμβάνει τα πεδία:

- **ID:** αποτελεί τον μοναδικό κωδικό για την κάθε εγγραφή του πίνακα (θετικός αυξανόμενος ακέραιος αριθμός με μέγεθος 11 ψηφίων).
- **Position:** πεδίο γεωγραφικών συντεταγμένων του σημείου που έλαβε χώρα η εκάστοτε μέτρηση (τύπος δεδομένων point, ειδικός τύπος για την αποθήκευση συντεταγμένων).
- **Downlink:** πεδίο απόδοσης της ταχύτητας κατεβάσματος (downlink) των ασύρματων δικτύων (τύπος δεδομένων θετικός δεκαδικός αριθμός).
- **Uplink:** πεδίο απόδοσης της ταχύτητας ανεβάσματος (uplink) των ασύρματων δικτύων (θετικός δεκαδικός αριθμός).
- **Ping1_IPERF:** πεδίο απόδοσης της ταχύτητας ping των ασύρματων δικτύων προς τον εξυπηρετητή της google (θετικός δεκαδικός αριθμός).
- **Ping2_DN:** πεδίο απόδοσης της ταχύτητας ping των ασύρματων δικτύων προς τον εξυπηρετητή που δημιουργήθηκε για την ανάγκες της εργασίας (θετικός δεκαδικός αριθμός).
- **Network_Type:** ο τύπος του δικτύου (WIFI, GSM, LTE) μέσω του οποίου έγινε η μέτρηση (αλφαριθμητικό με μέγεθος 30 χαρακτήρων).
- **Measurment_Time:** ημερομηνία και ώρα που πραγματοποιήθηκε η μέτρηση (datetime).
- **latitude:** γεωγραφικό πλάτος του σημείου που πραγματοποιήθηκε η μέτρηση (θετικός δεκαδικός αριθμός).
- **Longitude:** γεωγραφικό μήκος του σημείου που πραγματοποιήθηκε η μέτρηση (θετικός δεκαδικός αριθμός).
- **os:** λειτουργικό σύστημα (operating system) μέσω του οποίου πραγματοποιήθηκε η μέτρηση (αλφαριθμητικό με μέγεθος 30 χαρακτήρες).
- **Android_id:** μοναδικό αναγνωριστικό του λειτουργικού συστήματος android για την κάθε εφαρμογή (αλφαριθμητικό με μέγεθος 128 χαρακτήρων).
- **accuracy:** ακρίβεια σε μέτρα που απαιτείται για να γίνει δεκτή μια μέτρηση (θετικός δεκαδικός αριθμός με 5 ακέραια ψηφία και ένα δεκαδικό).
- **camp_id:** κωδικός της καμπάνιας στην οποία πραγματοποιήθηκε η μέτρηση (θετικός ακέραιος αριθμός).

- **Speed:** ταχύτητα με το οποίο κινούνταν η κινητή συσκευή κατά την πραγματοποίηση της μέτρησης (θετικός δεκαδικός αριθμός με 5 ακέραια ψηφία και 2 δεκαδικά).
- **SSID:** όνομα ασύρματου δικτύου μέσω του οποίου πραγματοποιήθηκε η μέτρηση (αλφαριθμητικό με μέγεθος 70 χαρακτήρων).
- **Operator:** όνομα παρόχου του δικτύου κινητής τηλεφωνίας (αλφαριθμητικό με μέγεθος 50 χαρακτήρων).
- **wifiLinkSpeed:** Η ταχύτητα σύνδεσης με το WiFi σημείο πρόσβασης (μικρός ακέραιος αριθμός 4^{ov} ψηφίων - smallint (4)).

Πίνακας “campaigns”

Χρησιμοποιείται για να αποθηκεύσει ο διαχειριστής τα στοιχεία για τις καμπάνιες που δημιουργεί μέσω της εφαρμογής MyWiFiSpeed και περιλαμβάνει τα πεδία:

- **camp_id:** αποτελεί τον μοναδικό κωδικό για την κάθε εγγραφή του πίνακα (θετικός αυξανόμενος ακέραιος αριθμός με μέγεθος 11 ψηφίων).
- **create_date:** ημερομηνία δημιουργίας της καμπάνιας (timestamp).
- **end_date:** ημερομηνία λήξης της καμπάνιας (date).
- **latitude:** γεωγραφικό πλάτος του κέντρου της περιοχής της καμπάνιας (δεκαδικός αριθμός).
- **longitude:** γεωγραφικό μήκος του κέντρου της περιοχής της καμπάνιας (δεκαδικός αριθμός).
- **radius:** η μέγιστη απόσταση σε μέτρα (ακτίνα) από το κέντρο της περιοχής της καμπάνιας που θα πραγματοποιούνται μετρήσεις (ακέραιος αριθμός 10 ψηφίων).
- **network_type:** ο τύπος δικτύου (WIFI, Mobile, Both) για τον οποίο θα πραγματοποιηθούν μετρήσεις στην συγκεκριμένη καμπάνια (αλφαριθμητικό με μέγεθος 10 χαρακτήρων).
- **accuracy:** η μέγιστη ακρίβεια σε μέτρα για την οποία θα γίνουν δεκτές οι μετρήσεις στην συγκεκριμένη καμπάνια (μικρός ακέραιος 5 ψηφίων – smallint(5)).
- **velocity:** η μέγιστη ταχύτητα κίνησης της συσκευής για την οποία θα γίνουν δεκτές μετρήσεις (tinyint(3)).
- **dist_frequency:** συχνότητα σε απόσταση κατά την οποία θα λαμβάνονται μετρήσεις (smallint(5)).
- **name:** όνομα καμπάνιας (αλφαριθμητικό με μέγεθος 70 χαρακτήρων)

- **active:** ένδειξη για το αν η καμπάνια είναι ενεργή ή όχι (Boolean).

Πίνακας “c2dm_reg_devices”

Χρησιμοποιείται για την αποθήκευση των κινητών συσκευών που έχουν κάνει εγγραφή στην υπηρεσία GCM και περιλαμβάνει τα πεδία:

- **c2dm_id:** αποτελεί τον μοναδικό κωδικό για την κάθε εγγραφή του πίνακα (θετικός αυξανόμενος ακέραιος αριθμός με μέγεθος 10 ψηφίων).
- **c2dm_regid:** ο μοναδικός κωδικός που αντιστοιχεί στην συσκευή κατά την διαδικασία εγγραφής της στην GCM υπηρεσία (αλφαριθμητικό με μέγεθος 250 χαρακτήρων).
- **c2dm_active:** ένδειξη για το αν η συσκευή είναι ενεργή ή όχι (Boolean)
- **c2dm_date:** ημερομηνία εγγραφής της συσκευής στην υπηρεσία GCM (timestamp).
- **c2dm_androidid:** μοναδικός κωδικός του λειτουργικού της κάθε συσκευής (αλφαριθμητικό με μέγεθος 128 χαρακτήρες).

Πίνακας “devices_per_campaign”

Πίνακας που χρησιμοποιείται για να συνδέσει τις καμπάνιες με τις συσκευές στις οποίες έχουν σταλθεί και περιλαμβάνει τα πεδία:

- **id:** αποτελεί τον μοναδικό κωδικό για την κάθε εγγραφή του πίνακα (θετικός αυξανόμενος ακέραιος αριθμός με μέγεθος 11 ψηφίων).
- **dc_campid:** το πεδίο που λειτουργεί ως ξένο κλειδί και δείχνει την αντίστοιχη καμπάνια στον πίνακα campaigns (θετικός ακέραιος αριθμός με μέγεθος 11 ψηφία).
- **dc_devid:** το πεδίο που λειτουργεί ως ξένο κλειδί και δείχνει την αντίστοιχη συσκευή στον πίνακα c2dm_regdevices (θετικός ακέραιος αριθμός με μέγεθος 11 ψηφία).
- **send_date:** η ημερομηνία αποστολής της καμπάνιας (dc_campid) στην συσκευή (dc_devid) (timestamp).

6.1.2 GeoServer

Σχετικά με τον GeoServer έχουμε αναφέρει στο κεφάλαιο 3.1 τι είναι, τι μας προσφέρει και ποιες οι λειτουργίες του, επίσης στο κεφάλαιο 3.2 έχουμε κάνει μια αναφορά στον αλγόριθμο σύντηξης δεδομένων Barnes Surface. Επιπλέον για τον GeoServer έχουμε παραθέσει στο κεφάλαιο 5.4 τον τρόπο εγκατάστασης και χρήσης του. Στην παρούσα

ενότητα αυτό που θα παρουσιάσουμε είναι η δημιουργία SLD (Styled Layer Descriptor) αρχείων, στον GeoServer, για τη σύντηξη δεδομένων με την χρήση του αλγόριθμου Barnes Surface. Για να δημιουργήσουμε όμως τα SLD αρχεία θα πρέπει αρχικά να έχουμε δημιουργήσει τα κατάλληλα Layers που να συνδέονται στην βάση δεδομένων μας. Εμείς στην παρούσα εργασία χρησιμοποιήσαμε το Layer “measurements” που είχε δημιουργηθεί στα πλαίσια της πτυχιακής εργασίας του Νικόλαου Κορωνιώτη^[65]. Αυτό που χρειάστηκε να προσθέσουμε εμείς, εκτός από το να δημιουργήσουμε τα SLD αρχεία που θα χρησιμοποιούν τον αλγόριθμο Barnes Surface, ήταν η επέκταση της WPS υπηρεσίας που προσφέρει τον αλγόριθμο Barnes Surface.

Εισαγωγή WPS επέκτασης

Για να εισάγουμε την επέκταση της WPS υπηρεσίας στον GeoServer, καθώς δεν είναι προ-εγκαταστημένη, ακολουθούμε τα παρακάτω βήματα:

1. Πλοηγούμαστε στην σελίδα geoserver.org/download/ από την οποία μπορούμε να κατεβάσουμε τον GeoServer.
2. Επιλέγουμε την έκδοση του GeoServer που χρησιμοποιούμε (Εικόνα 26), εμείς χρησιμοποιούμε την έκδοση 2.6.0.
3. Κάνουμε λήψη (download) της επέκτασης WPS. Ο σύνδεσμος για την λήψη της επέκτασης WPS βρίσκεται στην ενότητα Extensions (επεκτάσεις) κάτω από την υποενότητα Services υπηρεσίες.
4. Αποσυμπιέζουμε το αρχείο, που κάναμε λήψη, στον κατάλογο WEB-INF/lib της εγκατάστασης του GeoServer.
5. Επανεκκινούμε τον GeoServer.
6. Συνδεόμαστε στην σελίδα διαχείρισης του GeoServer (Εικόνα 29) και αν η επέκταση της WPS υπηρεσίας εισήχθη σωστά τότε θα πρέπει να δούμε το αποτέλεσμα που φαίνεται στην Εικόνα 29. Αν δεν δούμε αυτό το αποτέλεσμα θα πρέπει να ελέγξουμε το αρχείο συμβάντων (log file) του GeoServer για πιθανά λάθη.

Από την στιγμή που εκτελέσουμε με επιτυχία αυτά τα βήματα μπορούμε να δημιουργήσουμε τα SLD αρχεία και να χρησιμοποιήσουμε τον Barnes Surface αλγόριθμο για τον μετασχηματισμό των δεδομένων μας.

Δημιουργία SLD αρχείων

Για τις ανάγκες της εργασίας δημιουργήθηκαν τέσσερα SLD αρχεία:

- **DownLinkBarnesSurface:** το SLD αρχείο (Εικόνα 33) που εφαρμόζει τον αλγόριθμο Barnes Surface για τον μετασχηματισμό των δεδομένων που αφορούν την απόδοση της ταχύτητας λήψης (Downlink) των ασύρματων δικτύων.
- **UpLinkBarnesSurface:** το SLD αρχείο Εικόνα 33 που εφαρμόζει τον αλγόριθμο Barnes Surface για τον μετασχηματισμό των δεδομένων που αφορούν την απόδοση της ταχύτητας ανεβάσματος (Uplink) των ασύρματων δικτύων.
- **PingGoogleBarnesSurface:** το SLD αρχείο Εικόνα 33 που εφαρμόζει τον αλγόριθμο Barnes Surface για τον μετασχηματισμό των δεδομένων που αφορούν την απόδοση της ταχύτητας με την οποία εκτελείται το ping στον εξυπηρετητή της Google.
- **PingOurServerBarnesSurface:** το SLD αρχείο Εικόνα 33 που εφαρμόζει τον αλγόριθμο Barnes Surface για τον μετασχηματισμό των δεδομένων που αφορούν την απόδοση της ταχύτητας με την οποία εκτελείται το ping στο δικό μας εξυπηρετητή.

Ο κώδικας όλων των SLD αρχείων είναι παρόμοιος για αυτό και εδώ, για εξοικονόμηση χώρου θα παραθέσουμε ως παράδειγμα προς ανάλυση μόνο τον XML κώδικα του DownLinkBarnesSurface SLD αρχείου (Εικόνα 33). Τα τμήματα του κάθε αρχείου διακρίνονται στα εξής:

- Ορισμός του αλγόριθμου που θα χρησιμοποιηθεί για τον μετασχηματισμό (γραμμές 14 – 74).
- Ορισμός του κανόνα (rule) που θα χρησιμοποιηθεί για τον έλεγχο του πεδίου που μας ενδιαφέρει και για την μορφοποίηση του αποτελέσματος (γραμμές 75 – 98)
 - ο Ορισμός του φίλτρου το οποίο καθορίζει την πράξη ελέγχου που θέλουμε να γίνει μεταξύ του πεδίου της βάσης δεδομένων μας με μια τιμή (γραμμές 79 – 84)
 - ο Ορισμός του συμβολισμού του raster το οποίο καθορίζει την μορφοποίηση ανάλογα με την τιμή του πεδίου που ορίσαμε στον φίλτρο προς έλεγχο (γραμμές 75 – 97).

Αναλυτικότερα οι σημαντικότερες γραμμές μεταξύ των άλλων είναι:

- **Γραμμή 9:** Όνομα του SLD αρχείου. Αυτό το όνομα χρησιμοποιείται και στα request ως τιμή της παραμέτρου style.
- **Γραμμή 11:** Ο τίτλος του SLD αρχείου.

- **Γραμμή 14:** Ορίζεται ότι πρόκειται να χρησιμοποιηθεί μετασχηματισμός.
- **Γραμμή 15:** Ορισμός του μετασχηματισμού Barnes Surface ως ο μετασχηματισμός που θα χρησιμοποιηθεί.
- **Γραμμές 16 – 72:** Ορισμός των κατάλληλων τιμών για τις παραμέτρους που είναι απαραίτητες για την σωστή λειτουργία του μετασχηματισμού Barnes Surface. Τις παραμέτρους αυτές τις περιγράφει ο Πίνακας 2.
- **Γραμμή 75:** Ορισμός ενός κανόνα.
- **Γραμμή 79:** Ορισμός ενός φίλτρου.
- **Γραμμές 80 - 83:** Ορισμός σύγκρισης της παραμέτρου που μας ενδιαφέρει με μια τιμή.
- **Γραμμή 81:** Ορισμός του ονόματος του πεδίου στην βάση δεδομένων μας που θέλουμε να συγκρίνουμε με κάποια τιμή.
- **Γραμμή 82:** Ορισμός της τιμής με την οποία θέλουμε να συγκρίνουμε το πεδίο της βάσης δεδομένων μας.
- **Γραμμή 85:** Ορισμός της μορφοποίησης του raster.
- **Γραμμή 86:** Ορισμός του βαθμού διαφάνειας του raster (όσο μικρότερο τόσο πιο διαφανές).
- **Γραμμές 87 – 96:** Ορισμός του τρόπου που θα χρωματίζεται το raster ανάλογα με τις τιμές του πεδίου της βάσης δεδομένων μας που έχουμε ορίσει στο φίλτρο (γραμμή 81).

Εφόσον έχουμε δημιουργήσει ένα παρόμοιο αρχείο μπορούμε πλέον να εκτελέσουμε αιτήματα (request) προς τον GeoServer από τον πελάτη, αιτούμενοι χάρτες που περιλαμβάνουν τον μετασχηματισμό των δεδομένων μας ορίζοντας ως τιμή στην παράμετρο style το όνομα του SLD αρχείου.


```

1  <?xml version="1.0" encoding="ISO-8859-1"?>
2  <StyledLayerDescriptor version="1.0.0"
3    xsi:schemaLocation="http://www.opengis.net/sld StyledLayerDescriptor.xsd"
4    xmlns="http://www.opengis.net/sld"
5    xmlns:ogc="http://www.opengis.net/ogc"
6    xmlns:xlink="http://www.w3.org/1999/xlink"
7    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
8    <NamedLayer>
9      <Name>DownLinkBarnesSurface</Name>
10     <UserStyle>
11       <Title>DownLinkBarnesSurface</Title>
12       <Abstract>A style that produces a Barnes surface using a rendering transformation</Abstract>
13       <FeatureTypeStyle>
14         <Transformation>
15           <ogc:Function name="gs:BarnesSurface">
16             <ogc:Function name="parameter">
17               <ogc:Literal>data</ogc:Literal>
18             </ogc:Function>
19             <ogc:Function name="parameter">
20               <ogc:Literal>valueAttr</ogc:Literal>
21               <ogc:Literal>Downlink</ogc:Literal>
22             </ogc:Function>
23             <ogc:Function name="parameter">
24               <ogc:Literal>dataLimit</ogc:Literal>
25               <ogc:Literal>1000000</ogc:Literal> <!-- 1000000 -->
26             </ogc:Function>
27             <ogc:Function name="parameter">
28               <ogc:Literal>scale</ogc:Literal>
29               <ogc:Literal>.0080</ogc:Literal> <!-- 0.000025 -->
30             </ogc:Function>
31             <ogc:Function name="parameter">
32               <ogc:Literal>convergence</ogc:Literal>
33               <ogc:Literal>0.6</ogc:Literal>
34             </ogc:Function>
35             <ogc:Function name="parameter">
36               <ogc:Literal>passes</ogc:Literal>
37               <ogc:Literal>10</ogc:Literal> <!--2-->
38             </ogc:Function>
39             <ogc:Function name="parameter">
40               <ogc:Literal>minObservations</ogc:Literal>
41               <ogc:Literal>2</ogc:Literal> <!--2-->
42             </ogc:Function>
43             <ogc:Function name="parameter">
44               <ogc:Literal>maxObservationDistance</ogc:Literal>
45               <ogc:Literal>0.0025</ogc:Literal> <!--0.0005-->
46             </ogc:Function>
47             <ogc:Function name="parameter">
48               <ogc:Literal>pixelsPerCell</ogc:Literal>
49               <ogc:Literal>10</ogc:Literal> <!--8-->
50             </ogc:Function>
51             <ogc:Function name="parameter">
52               <ogc:Literal>outputBBOX</ogc:Literal>
53               <ogc:Function name="env">
54                 <ogc:Literal>wms_bbox</ogc:Literal>
55               </ogc:Function>
56             </ogc:Function>
57             <ogc:Function name="parameter">
58               <ogc:Literal>outputWidth</ogc:Literal>
59               <ogc:Function name="env">
60                 <ogc:Literal>wms_width</ogc:Literal>
61               </ogc:Function>
62             </ogc:Function>
63             <ogc:Function name="parameter">
64               <ogc:Literal>outputHeight</ogc:Literal>
65               <ogc:Function name="env">
66                 <ogc:Literal>wms_height</ogc:Literal>
67               </ogc:Function>
68             </ogc:Function>
69             <ogc:Function name="parameter">
70               <ogc:Literal>queryBuffer</ogc:Literal>
71               <ogc:Literal>0.5</ogc:Literal> <!--0.001-->
72             </ogc:Function>
73           </ogc:Function>
74         </Transformation>
75         <Rule>
76           <Name>rule1</Name>
77           <Title>Give Color to Raster</Title>
78           <Abstract>A raster with 100% opacity</Abstract>
79           <ogc:Filter>
80             <ogc:PropertyIsGreaterThan>
81               <ogc:PropertyName>Downlink</ogc:PropertyName>
82               <ogc:Literal>100</ogc:Literal>
83             </ogc:PropertyIsGreaterThan>
84           </ogc:Filter>
85           <RasterSymbolizer>
86             <Opacity>0.5</Opacity>
87             <ColorMap type="ramp">
88               <ColorMapEntry color="#000000" quantity="512" label="Less than 512" opacity="0"/>
89               <ColorMapEntry color="#C0C0C0" quantity="1024" label="512 to 1024" opacity="1"/>
90               <ColorMapEntry color="#0000FF" quantity="2048" label="1024 to 2048" opacity="1"/>
91               <ColorMapEntry color="#00FFFF" quantity="4096" label="2048 to 4096" opacity="1"/>
92               <ColorMapEntry color="#00FF00" quantity="8192" label="4096 to 8192" opacity="1"/>
93               <ColorMapEntry color="#FFFF00" quantity="16384" label="8192 to 16384" opacity="1"/>
94               <ColorMapEntry color="#FF9900" quantity="32768" label="16384 to 32768" opacity="1"/>
95               <ColorMapEntry color="#FF0000" quantity="100000" label="Greater than 32768" opacity="1"/>
96             </ColorMap>
97           </RasterSymbolizer>
98         </Rule>
99       </FeatureTypeStyle>
100     </UserStyle>
101   </NamedLayer>
102 </StyledLayerDescriptor>
103

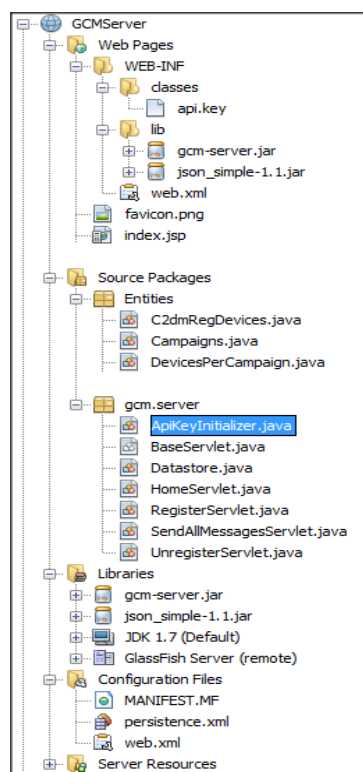
```

Εικόνα 33: Κώδικας DownLinkBarnesSurface SLD αρχείου

6.1.3 GCM Application Server

Στην ενότητα αυτή παρουσιάζουμε την παραμετροποίηση που έγινε στον GCM application server για να καταστεί δυνατή η χρήση του από την διαδικτυακή εφαρμογή MyWiFiSpeed με σκοπό να αποσταλούν οι καμπάνιες στις κινητές συσκευές. Την GCM (Google Cloud Messaging) υπηρεσία, τα επιμέρους στοιχεία της και τι χρειάζεται να υλοποιήσουμε προκειμένου να την χρησιμοποιήσουμε τα έχουμε περιγράψει στο κεφάλαιο 5.5. Σύμφωνα με τις περιγραφές αυτές εμείς σαν πρώτο βήμα ήταν να κάνουμε λήψη της εφαρμογής ανοιχτού κώδικα GCM application server^[68] που προσφέρει η google ως παράδειγμα και να τον παραμετροποιήσουμε. Εμείς (για λόγους που αναφέρουμε στο κεφάλαιο 5.5) επιλέξαμε να υλοποιήσουμε τον GCM application server που κάνει χρήση του πρωτοκόλλου HTTP για την επικοινωνία του με τους GCM connection servers.

Εμβαθύνοντας σε τεχνικά ζητήματα ο GCM application server είναι μια διαδικτυακή εφαρμογή που κάνει χρήση των τεχνολογιών της Java και ειδικότερα της τεχνολογίας JavaServlets. Στην Εικόνα 34 παρουσιάζεται η δομή του GCM application server η οποία συμπεριλαμβάνει τα servlets από το παράδειγμα υλοποίησης της google, αλλά και τις δικές μας προσθήκες και τροποποιήσεις. Το κάθε ένα στοιχείο από αυτά αναλύεται στη συνέχεια.



Εικόνα 34: Δομή GCM application server

api.key

Είναι αρχείο κειμένου στο οποίο αποθηκεύεται (σε μορφή απλού κειμένου) ο μοναδικός κωδικός που έχουμε λάβει από την υπηρεσία Google Cloud Messaging κατά την εγγραφή του GCM application server μας σε αυτήν. Έχουμε αλλάξει το αρχικό περιεχόμενο του αρχείου αυτού αντικαθιστώντας το ενδεικτικό κλειδί που υπήρχε, στην υλοποίηση που κατεβάσαμε από την google, με το δικό μας μοναδικό κλειδί.

gcm-server.jar

Είναι η βιβλιοθήκη που περιλαμβάνει τις κλάσεις του GCM application server, έτσι όπως έχουν υλοποιηθεί από την google.

json-simple.jar

Είναι η βιβλιοθήκη που περιλαμβάνει τις κλάσεις για την κωδικοποίηση των μηνυμάτων σε json μορφή με σκοπό την αποστολή τους στους GCM connection servers. Η βιβλιοθήκη αυτή δεν έχει τροποποιηθεί από εμάς.

favicon.png

Είναι το λογότυπο του GCM server.

Index.jsp

Είναι η αρχική σελίδα του GCM application server (ουσιαστικά περιλαμβάνει μόνο μια γραμμή κώδικα που μας ανακατευθύνει (redirect) στην HomeServlet.java (βλέπε παρακάτω)). Εμείς δεν έχουμε τροποποιήσει αυτή τη σελίδα.

Entities Package

Στο πακέτο αυτό έχουν υλοποιηθεί από εμάς οι Java κλάσεις που αντιστοιχούν στους πίνακες και στα πεδία της βάσης δεδομένων μας (Εικόνα 32), που χρειαζόμαστε ώστε να αποσταλούν οι καμπάνιες στις κινητές συσκευές. Δηλαδή οι κλάσεις αυτές παρέχουν τις ιδιότητες και τις συναρτήσεις που είναι απαραίτητες ώστε με τη χρήση της τεχνολογίας JPA^[72] να καταφέρουμε να υλοποιήσουμε τις CRUD^[73] (Create, Read, Update, Delete) λειτουργίες προς την βάση δεδομένων. Συνεπώς οι οντότητες που δημιουργήθηκαν (ακολουθώντας τον εξής οδηγό^[71]) είναι:

C2dmRegDevices.java

Κλάση που αντιστοιχεί στον πίνακα c2dm_reg_devices της βάσης δεδομένων μας.

Campaigns.java

Κλάση που αντιστοιχεί στον πίνακα campaigns της βάσης δεδομένων μας.

DevicesPerCampaign.java

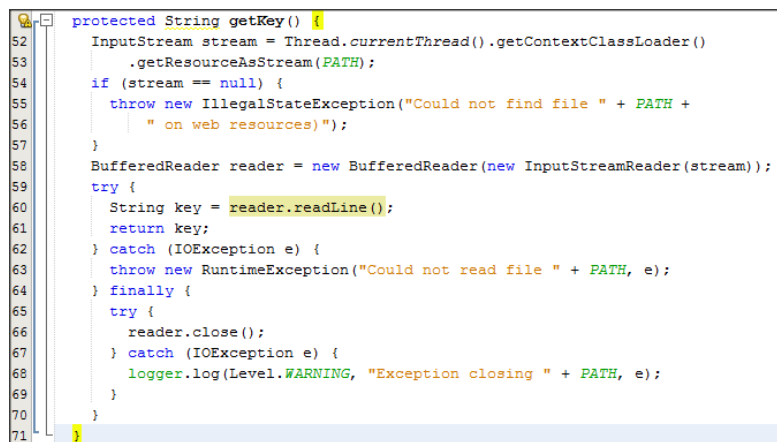
Κλάση που αντιστοιχεί στον πίνακα devices_per_campaign της βάσης δεδομένων μας.

gcm.server Package

Στο πακέτο αυτό έχουν υλοποιηθεί τα Servlets που εκτελούν τις κύριες λειτουργίες του GCM application server. Συγκεκριμένα τα Servlets έχουν ως εξής:

ApiKeyInitializer.java

Το Servlet αυτό ανακτά τον μοναδικό κωδικό εγγραφής του GCM application server μας στο Google Cloud Message από το αρχείο api.key που προαναφέραμε. Ο σημαντικότερος κώδικας της σελίδας, που εκτελεί αυτή την λειτουργία είναι η συνάρτηση που φαίνεται στην Εικόνα 35 και δεν έχει τροποποιηθεί από εμάς.



```
52 protected String getKey() {
53     InputStream stream = Thread.currentThread().getContextClassLoader()
54         .getResourceAsStream(PATH);
55     if (stream == null) {
56         throw new IllegalStateException("Could not find file " + PATH +
57             " on web resources");
58     }
59     BufferedReader reader = new BufferedReader(new InputStreamReader(stream));
60     try {
61         String key = reader.readLine();
62         return key;
63     } catch (IOException e) {
64         throw new RuntimeException("Could not read file " + PATH, e);
65     } finally {
66         try {
67             reader.close();
68         } catch (IOException e) {
69             logger.log(Level.WARNING, "Exception closing " + PATH, e);
70         }
71     }
72 }
```

Εικόνα 35: Συνάρτηση της Servlet ApiKeyInitializer.java σελίδας για την ανάκτηση του κωδικού εγγραφής από το αρχείο api.key

BaseServlet.java

Αποτελεί τη βασική Servlet σελίδα που υλοποιεί βασικές συναρτήσεις οι οποίες χρησιμοποιούνται (γίνονται extend) από τα υπόλοιπα servlet.

Datastore.java

Είναι η κλάση στην οποία υλοποιούνται οι συναρτήσεις μέσω των οποίων χρησιμοποιώντας τις Entity classes, που περιγράφηκαν πιο πάνω, πραγματοποιείται η επικοινωνία με τη βάση δεδομένων. Οι συναρτήσεις που εμείς προσθέσαμε και χρησιμοποιούμε είναι:

- **getDevicesFromDatabase** (Εικόνα 36): ανακτά όλους τους κωδικούς εγγραφής των εγγεγραμμένων συσκευών στην υπηρεσία Google Cloud Messages που υπάρχουν στον πίνακα c2dm_reg_devices της βάσης δεδομένων μας.

```

94 public static List<String> getDevicesFromDatabase() {
95     List<String> DatabaseRegIds = new ArrayList<>();
96
97     EntityManagerFactory emf = Persistence.createEntityManagerFactory("GSMServerPU");
98     EntityManager em = emf.createEntityManager();
99     em.getTransaction().begin();
100
101     Query mQuery = em.createNativeQuery("SELECT c2dm_regid FROM c2dm_reg_devices"
102         + " where c2dm_active = '1'");
103     DatabaseRegIds = mQuery.getResultList();
104     em.close();
105     emf.close();
106     return DatabaseRegIds;
107 }

```

Εικόνα 36: Συνάρτηση getDevicesFromDatabase της κλάσης Datastore.java για την ανάκτηση των κωδικών εγγραφής των εγγεγραμμένων συσκευών.

- **UnregisterDevice** (Εικόνα 37): καταργεί τις συσκευές, που έχουν διαγραφεί από την υπηρεσία Google Cloud Messages, από τον πίνακα c2dm_reg_devices της βάσης μας.

```

108 public static void UnregisterDevice(String regId) {
109     logger.info("Unregistering " + regId);
110     EntityManagerFactory emf = Persistence.createEntityManagerFactory("GSMServerPU");
111     EntityManager em = emf.createEntityManager();
112     em.getTransaction().begin();
113
114     C2dmRegDevices mDevice = em.createNamedQuery("C2dmRegDevices.findByC2dmRegid",
115         C2dmRegDevices.class).setParameter("c2dmRegid", regId).getSingleResult();
116     mDevice.setC2dmActive(false);
117
118     em.persist(mDevice);
119     em.getTransaction().commit();
120
121     em.close();
122     emf.close();
123 }

```

Εικόνα 37: Συνάρτηση UnregisterDevice της κλάσης Datastore.java για την κατάργηση των μη εγγεγραμμένων συσκευών.

- **UpdateRegistrationDevice** (Εικόνα 38): ανανεώνει (update) τον κωδικό εγγραφής των συσκευών στον πίνακα c2dm_reg_devices της βάσης μας όταν αυτές έχουν αλλάξει κωδικό εγγραφής στην υπηρεσία Google Cloud Messages.

```

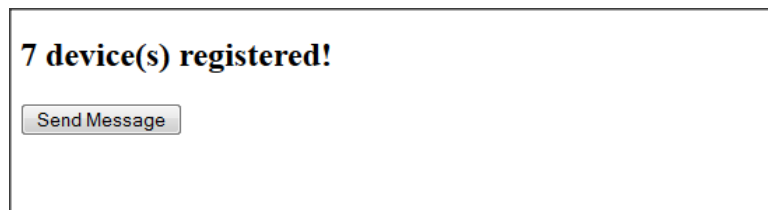
124 public static void UpdateRegistrationDevice(String oldId, String newId) {
125     logger.info("Updating " + oldId + " to " + newId);
126
127     EntityManagerFactory emf = Persistence.createEntityManagerFactory("GSMServerPU");
128     EntityManager em = emf.createEntityManager();
129     em.getTransaction().begin();
130
131     C2dmRegDevices mDevice = em.createNamedQuery("C2dmRegDevices.findByC2dmRegid",
132         C2dmRegDevices.class).setParameter("c2dmRegid", oldId).getSingleResult();
133     mDevice.setC2dmRegid(newId);
134
135     em.persist(mDevice);
136     em.getTransaction().commit();
137
138     em.close();
139     emf.close();
140 }

```

Εικόνα 38: Συνάρτηση UpdateRegistrationDevice της κλάσης Datastore.java για την ανανέωση του κωδικού εγγραφής των συσκευών που έχουν αλλάξει κωδικό εγγραφής.

HomeServlet.java

Η HomeServlet.java αποτελεί την αρχική servlet σελίδα του GCM application server όπως έχει υλοποιηθεί από την google στο παράδειγμά της. Πρακτικά η σελίδα αυτή εμφανίζει στην οθόνη του browser (Εικόνα 39) το σύνολο των εγγεγραμμένων κινητών συσκευών στη βάση δεδομένων μας και ένα κουμπί το οποίο προσφέρει τη δυνατότητα αποστολής ενός στοιχειώδους μηνύματος προς αυτές. Για την αποστολή του μηνύματος προς τις εγγεγραμμένες κινητές συσκευές καλεί τη σελίδα SendAllMessagesServlet.java (βλ. Παρακάτω). Εμείς τη σελίδα αυτή δεν την χρησιμοποιούμε σε κάποιο σημείο απλώς την έχουμε αφήσει για λόγους επίδειξης.



Εικόνα 39: Παράδειγμα αποτελέσματος από την εκτέλεση της σελίδας HomeServlet.java

RegisterServlet.Java

Είναι η servlet σελίδα, όπως έχει υλοποιηθεί από την google στο παράδειγμα της, που πραγματοποιεί την εγγραφή των νέων κινητών συσκευών στην βάση δεδομένων μας. Εμείς την σελίδα αυτή δεν την χρησιμοποιούμε, καθώς η λειτουργία της εγγραφής των νέων κινητών συσκευών στην βάση εκτελείται από διαδικασία που έχει υλοποιηθεί στην διπλωματική εργασία του Μωυσιάδης Αναστάσιου^[66].

UnregisterServlet.java

Είναι η servlet σελίδα, όπως έχει υλοποιηθεί από την google στο παράδειγμα της, που καταργεί τις συσκευές, που έχουν διαγραφεί από την υπηρεσία Google Cloud Messages, από την βάση δεδομένων μας. Εμείς την σελίδα αυτή δεν την χρησιμοποιούμε, καθώς η κατάργηση κινητών συσκευών στην βάση εκτελείται από νέα διαδικασία που έχουμε υλοποιήσει (βλ. παρακάτω).

SendAllMessagesServlet.java

Είναι ίσως η πιο σημαντική servlet σελίδα του GCM application server, καθώς συνδυάζει τις υπόλοιπες servlet σελίδες και χρησιμοποιεί συναρτήσεις από διάφορες κλάσεις για την αποστολή των μηνυμάτων στις κινητές συσκευές. Η SendAllMessagesServlet.java είναι η σελίδα που έχει τροποποιηθεί από εμάς και είναι αυτή που καλείται από την διαδικτυακή εφαρμογή MyWiFiSpeed για να σταλθούν οι καμπάνιες στις

κινητές συσκευές. Συγκεκριμένα οι συναρτήσεις που υλοποιούνται σε αυτήν την servlet σελίδα είναι

- **doPost:** είναι η γνωστή συνάρτηση η οποία σε όλα τα servlet είναι υπεύθυνη για την διαχείριση των αιτημάτων (requests). Στην δική μας περίπτωση εκτελεί επιπλέον τις λειτουργίες όπως είναι η ανάκτηση του json μηνύματος που συμπεριλαμβάνει τις καμπάνιες και έχει σταλεί από την διαδικτυακή εφαρμογή MyWiFiSpeed μέσω HTTP αιτήματος (request) στην παράμετρο “msg” που έχουμε ορίσει εμείς (Εικόνα 40). Σαν δεύτερο βήμα ανακτά σε μια λίστα όλους τους κωδικούς των εγγεγραμμένων συσκευών που υπάρχουν στην βάση δεδομένων μας (Εικόνα 41) χρησιμοποιώντας την συνάρτηση “getDevicesFromDatabase” της κλάσης “Datastore” που έχουμε περιγράψει πιο πάνω. Χρειαζόμαστε τους κωδικούς αυτούς ώστε να στείλουμε τις καμπάνιες στις κινητές συσκευές. Στη συνέχεια η συνάρτηση μας, στον κώδικα που φαίνεται στην Εικόνα 42, κάνει έλεγχο για το αν βρέθηκαν κωδικοί εγγεγραμμένων συσκευών στην βάση δεδομένων. Αν όχι τότε επιστρέφει κατάλληλο μήνυμα στην διαδικτυακή εφαρμογή MyWiFiSpeed με το οποίο την ενημερώνει για τη μη ύπαρξης εγγεγραμμένων συσκευών στην βάση δεδομένων. Αλλιώς αν βρεθούν εγγεγραμμένες συσκευές ελέγχει για το αν είναι μόνο μια. Αν υπάρχει μόνο μια εγγεγραμμένη συσκευή τότε στέλνει το μήνυμα σε αυτήν κάνοντας χρήση της συνάρτησης “send” της κλάσης sender που έχει υλοποιηθεί από την google στο παράδειγμα της. Αν υπάρχουν πάνω από μια εγγεγραμμένες συσκευές τότε χωρίζονται σε ομάδες των χιλίων (1000) και το μήνυμα στέλνεται σε αυτές με τη χρήση της συνάρτησης “asyncSend” (βλ. παρακάτω) περνώντας της ως παραμέτρους την ομάδα των χιλίων (1000) συσκευών και το json μήνυμα που πρέπει να σταλεί προς αυτές. Τέλος η doPost απαντάει πίσω στην διαδικτυακή εφαρμογή MyWiFiSpeed με το κατάλληλο μήνυμα, για το αν στάλθηκαν οι καμπάνιες στις συσκευές ή όχι.

```
90  
91 String jsonMessage = getParameter(req, PARAMETER_MESSAGE);  
92
```

Εικόνα 40: Ανάκτηση του json μηνύματος που έχει σταλεί από την διαδικτυακή εφαρμογή MyWiFiSpeed μέσω HTTP αιτήματος (request) στην doPost.

```

90
91     List<String> devices = Datastore.getDevicesFromDatabase();
92

```

Εικόνα 41: Ανάκτηση όλων των κωδικών εγγραφής των συσκευών στην doPost.

```

91     if (devices.isEmpty()) {
92         errorMsg = "Message ignored as there is no device registered!";
93         resp.setStatus(-1);
94     } else {
95         // NOTE: check below is for demonstration purposes; a real application
96         // could always send a multicast, even for just one recipient
97         if (devices.size() == 1) {
98             // send a single message using plain post
99             String registrationId = devices.get(0);
100             Message message = new Message.Builder().addData("msg", jsonMessage).build();
101             Result result = sender.send(message, registrationId, 5);
102         } else {
103             // send a multicast message using JSON
104             // must split in chunks of 1000 devices (GCM limit)
105             int total = devices.size();
106             List<String> partialDevices = new ArrayList<String>(total);
107             int counter = 0;
108             int tasks = 0;
109             for (String device : devices) {
110                 counter++;
111                 partialDevices.add(device);
112                 int partialSize = partialDevices.size();
113                 if (partialSize == MULTICAST_SIZE || counter == total) {
114                     asyncSend(partialDevices, jsonMessage);
115                     partialDevices.clear();
116                     tasks++;
117                 }
118             }
119             successMsg = "Asynchronously sending " + tasks + " multicast messages to " +
120                 total + " devices";
121             resp.setStatus(HttpServletResponse.SC_OK);
122         }
123     }

```

Εικόνα 42: Κώδικας της doPost για την αποστολή των μηνυμάτων.

- asyncSend** (Εικόνα 43): η συνάρτηση αυτή δέχεται ως παραμέτρους μια λίστα με τους κωδικούς των συσκευών στις οποίες θα πρέπει να στείλει το json μήνυμα που δέχεται ως δεύτερη παράμετρο. Με άλλα λόγια, η κύρια λειτουργία της συνάρτησης αυτής είναι να στείλει τις καμπάνιες που έρχονται από την διαδικτυακή εφαρμογή MyWiFiSpeed στην εφαρμογή HuaServiceTester2 των κινητών συσκευών. Όπως μπορεί να γίνει κατανοητό η συνάρτηση αυτή “υλοποιεί” τους λόγους για τους οποίους χρησιμοποιήσαμε τον GCM application server. Για να γίνει αυτή η εργασία, πίσω στον κώδικα στην Εικόνα 43, η συνάρτηση “asyncSend” κάνει και αυτή χρήση της συνάρτησης “send” της κλάσης “sender” αλλά αυτή τη φορά ασύγχρονα, δηλαδή την καλεί από διαφορετικά νήματα (threads) κάθε φορά. Τέλος αφού ολοκληρωθεί η αποστολή του μηνύματος στις συσκευές ελέγχει την απάντηση από τους GCM connection servers και ανάλογα με αυτή επιστρέφει μήνυμα επιτυχής αποστολής και αν χρειαστεί εκτελεί την συνάρτηση “UpdateRegistrationDevice” της κλάσης “Datastore” (εμείς εδώ δεν

την χρησιμοποιούμε, αλλά την έχουμε αφήσει για εκπαιδευτικούς σκοπούς ή για μελλοντική χρήση) ή την συνάρτηση “UnregisterDevice” της ίδιας κλάσης που έχουμε περιγράψει πιο πάνω.

```
133 private void asyncSend(List<String> partialDevices, String jsonMessage) {
134     // make a copy
135     final List<String> devices = new ArrayList<String>(partialDevices);
136     final String jsonMsg = jsonMessage;
137     threadPool.execute(new Runnable() {
138
139         public void run() {
140             Message message = new Message.Builder().addData("msg", jsonMsg).build();
141             MulticastResult multicastResult;
142             try {
143                 multicastResult = sender.send(message, devices, 5);
144             } catch (IOException e) {
145                 logger.log(Level.SEVERE, "Error posting messages", e);
146                 return;
147             }
148             List<Result> results = multicastResult.getResults();
149             // analyze the results
150             for (int i = 0; i < devices.size(); i++) {
151                 String regId = devices.get(i);
152                 Result result = results.get(i);
153                 String messageId = result.getMessageId();
154                 if (messageId != null) {
155                     logger.fine("Successfully sent message to device: " + regId +
156                               "; messageId = " + messageId);
157                     String canonicalRegId = result.getCanonicalRegistrationId();
158                     if (canonicalRegId != null) {
159                         // same device has more than one registration id: update it
160                         logger.info("canonicalRegId " + canonicalRegId);
161                         Datastore.UpdateRegistrationDevice(regId, canonicalRegId);
162                     }
163                 } else {
164                     String error = result.getErrorCodeName();
165                     if (error.equals(Constants.ERROR_NOT_REGISTERED)) {
166                         // application has been removed from device - unregister it
167                         logger.info("Unregistered device: " + regId);
168                         Datastore.UnregisterDevice(regId);
169                     } else {
170                         logger.severe("Error sending message to " + regId + ": " + error);
171                     }
172                 }
173             }
174         }
175     });
176 }
```

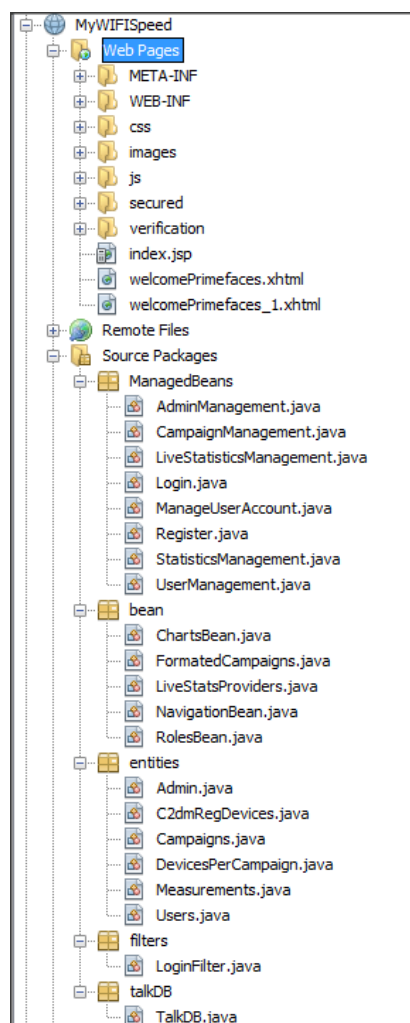
Εικόνα 43: Συνάρτηση asyncSend.

6.1.4 MyWiFiSpeed

Στην ενότητα αυτή παρουσιάζουμε τις τεχνικές λεπτομέρειες των επιμέρους στοιχείων (Εικόνα 44) που δημιουργήσαμε για την υλοποίηση της διαδικτυακής εφαρμογής MyWiFiSpeed και τη ροή – τρόπο επικοινωνίας μεταξύ τους (Εικόνα 45). Συγκεκριμένα αναλύουμε όλες τις κλάσεις και τις σελίδες, οι οποίες εξυπηρετούν τον χρήστη, εξηγώντας τα σημαντικότερα τμήματα του κώδικά τους. Πριν όμως προχωρήσουμε στην ανάλυση αυτή θα αναφέρουμε ένα ενδεικτικό παράδειγμα της ροής που λαμβάνει χώρα στην εφαρμογή μας κατά τη σύνδεση ενός χρήστη. Έστω λοιπόν ότι ο χρήστης είναι εγγεγραμμένος και συνδέεται στην εφαρμογή μας. Αρχικά με τον που επισκεφθεί ο χρήστης τη διεύθυνση της εφαρμογής μας, το αίτημα (request) του πηγαίνει στο φίλτρο

μας (LoginFilter.java) για να αποφασιστεί αν ο χρήστης είναι ήδη συνδεδεμένος. Αν είναι από πριν συνδεδεμένος τότε το αίτημα του προωθείται στην κεντρική σελίδα (index.xhtml), αλλιώς προωθείται στη σελίδα σύνδεσης. Από την σελίδα σύνδεσης ο χρήστης μπορεί να συνδεθεί αν έχει λογαριασμό ή να κάνει εγγραφή αν δεν έχει. Στη συνέχεια εφόσον ο χρήστης προσπαθήσει να συνδεθεί γίνεται έλεγχος από την login.java αν τα διαπιστευτήρια του είναι σωστά. Αν οι κωδικοί που έδωσε ο χρήστης είναι σωστοί τότε ελέγχεται ποιος είναι ο ρόλος του (διαχειριστής ή απλός χρήστης) και προωθείται στη κεντρική σελίδα δίνοντας του τα κατάλληλα δικαιώματα, αλλιώς προωθείται πάλι στην σελίδα σύνδεσης προβάλλοντας του το σχετικό μήνυμα λάθους. Εφόσον τελικά ο χρήστης συνδέθηκε μπορεί από την κεντρική σελίδα να κάνει τα εξής ανάλογα με τα δικαιώματα του.

- Να διαχειριστεί τον λογαριασμό του (δυνατότητα διαθέσιμη και στους δυο ρόλους)



Εικόνα 44: Κλάσεις που δημιουργήθηκαν για την υλοποίηση της διαδικτυακής εφαρμογής MyWiFiSpeed.

- Να δει στατιστικά στοιχεία σχετικά με τις αποδόσεις των ασύρματων δικτύων (δυνατότητα διαθέσιμη και στους δύο ρόλους).
- Να δημιουργήσει νέες καμπάνιες, να διαχειριστεί τις υπάρχουσες και να δει στατιστικά στοιχεία για αυτές (δυνατότητα διαθέσιμη μόνο για τον ρόλο του διαχειριστή).
- Να εισάγει νέους χρήστες και για τους δύο ρόλους και να διαχειριστεί και τους ήδη υπάρχοντες (δυνατότητα διαθέσιμη μόνο για τον ρόλο του διαχειριστή)

Τέλος ο χρήστης μπορεί να αποσυνδεθεί από την εφαρμογή. Στις επόμενες παραγράφους γίνεται η ανάλυση της κάθε κλάσης ξεχωριστά και παρατίθενται τα σημαντικότερα τμήματα του κώδικά τους.

Entities Package

Όπως αναφέραμε και στην ανάλυση του GCM application server (κεφάλαιο 6.1.3), έτσι και εδώ στο πακέτο αυτό έχουμε δημιουργήσει τις κλάσεις που υλοποιούν σε προγραμματιστικό επίπεδο τη δομή των πινάκων της βάσης δεδομένων μας. Ο σκοπός των Java αυτών κλάσεων είναι να παρέχουν τη δυνατότητα, χρησιμοποιώντας την τεχνολογία JPA, να επιτευχθεί η επικοινωνία της εφαρμογής μας με την βάση δεδομένων για την εγγραφή, ανάγνωση...

Οι κλάσεις που συμπεριλαμβάνονται στο πακέτο αυτό δημιουργήθηκαν και εδώ ακολουθώντας τον οδηγό^[71] και αντιστοιχούν στους πίνακες της βάσης μας όπως περιγράφεται παρακάτω.

Admin.java

Αντιστοιχεί στον πίνακα admin (Εικόνα 32) της βάσης δεδομένων μας. Η κλάση αυτή χρησιμοποιείται για τη διαχείριση των ανεργιών στον ρόλο του διαχειριστή.

Users.java

Αντιστοιχεί στον πίνακα users (Εικόνα 32) της βάσης δεδομένων μας. Η κλάση αυτή χρησιμοποιείται για τη διαχείριση των ανεργιών στον ρόλο του απλού χρήστη.

Measurements.java

Αποτελεί ίσως την βασικότερη κλάση του πακέτου Entities καθώς υλοποιεί τον σημαντικότερο πίνακα της βάσης δεδομένων μας, τον measurements (Εικόνα 32). Ο πίνακας αυτός περιέχει όλες τις τιμές των μετρήσεων που αποστέλλονται από τις κινητές συσκευές και έτσι μέσω της κλάσης Measurements.java διαχειριζόμαστε αυτές τις τιμές με σκοπό την εξαγωγή στατιστικής πληροφορίας.

Campaigns.java

Όπως και στον GCM application server (κεφάλαιο 6.1.3) έτσι και εδώ είναι η κλάση που αντιστοιχεί στον πίνακα campaigns (Εικόνα 32) της βάσης δεδομένων μας. Χρησιμοποιείται για να καταστεί δυνατή η διαχείριση στις καμπάνιες.

C2dmRegDevices.java

Αντιστοιχεί στον πίνακα c2dm_reg_devices (Εικόνα 32) της βάσης δεδομένων μας. Ο πίνακας αυτός περιέχει τις εγγεγραμμένες συσκευές στην Google Cloud Message υπηρεσία στις οποίες θα αποσταλούν οι καμπάνιες.

DevicesPerCampaign.java

Αντιστοιχεί στον πίνακα devices_per_campaign (Εικόνα 32) της βάσης δεδομένων μας. Ο πίνακας αυτός περιέχει ποιες καμπάνιες έχουν σταλεί σε ποιες συσκευές.

TalkDB.java

Την κλάση TalkDB.java την δημιουργήσαμε με σκοπό να έχουμε συγκεντρωμένες όλες τις συναρτήσεις (functions) που καθιστούν δυνατή την επικοινωνία με τις entity classes. Δηλαδή στην κλάση αυτή έχουν δημιουργηθεί οι συναρτήσεις που χρησιμοποιώντας την τεχνολογία JPA^[72] και μέσω των entity classes καθιστούν δυνατή την υλοποίηση των CRUD^[73] λειτουργιών με την βάση δεδομένων μας.

ChartsBean.java

Είναι η κλάση στην οποία έχουν υλοποιηθεί οι συναρτήσεις που δημιουργούν, με τη χρήση των chart μοντέλων των primefaces, τα στατιστικά γραφήματα που παρουσιάζονται στον χρήστη στη κεντρική σελίδα. Ως παράδειγμα μπορούμε να δούμε στην Εικόνα 46 την συνάρτηση GetMeasurementsPie που δημιουργεί το γράφημα “πίτα” για τις συνολικές μετρήσεις των δικτύων, των λειτουργικών συστημάτων και των παρόχων. Η συνάρτηση αυτή δέχεται ως παράμετρο ποιες μετρήσεις θα αφορά το γράφημα (των δικτύων, των λειτουργικών συστημάτων ή των παρόχων) και επιστρέφει ένα αντικείμενο τύπου PieChartModel το οποίο είναι και το γράφημα που αναπαρίσταται στη κεντρική σελίδα.

```

200 public PieChartModel GetMeasurementsPie(String pieType){
201     PieChartModel mPieChartModel = new PieChartModel();
202
203     TalkDB mTalkDB = new TalkDB();
204     List mList = mTalkDB.CountMeasurements(pieType);
205
206     for(Object obj : mList){
207         Object[] mObject = (Object[]) obj;
208         //mObject[0] = network type, mObject[1] = measurements sum
209         mPieChartModel.set(mObject[0].toString(),
210             Double.valueOf(mObject[1].toString()));
211     }
212     switch (pieType) {
213         case "Network_Type":
214             mPieChartModel.setTitle("Measurements By Network");
215             break;
216         case "os":
217             mPieChartModel.setTitle("Measurements BY OS");
218             break;
219         case "Operator":
220             mPieChartModel.setTitle("Measurements BY Provider");
221             break;
222     }
223     mPieChartModel.setLegendPosition("w");
224     mPieChartModel.setShowDataLabels(true);
225
226     return mPieChartModel;
227 }

```

Εικόνα 46: Συνάρτηση GetMeasurementsPie.

LiveStatsProviders.java

Είναι η κλάση της οποίας τα αντικείμενα τα χρησιμοποιούμε για να ανακτήσουμε και να αναπαραστήσουμε στατιστικά, σε πραγματικό χρόνο, όπως ο συνολικός αριθμός των μετρήσεων και το μέγιστο downlink και uplink αυτή τη στιγμή στη βάση δεδομένων μας.

FormattedCampaigns.java

Είναι η κλάση της οποίας τα αντικείμενα τα χρησιμοποιούμε για να διαμορφώσουμε κατάλληλα τις καμπάνιες που υπάρχουν στη βάση δεδομένων με σκοπό την αποστολή τους στον GCM application server.

NavigationBean.java

Είναι η κλάση στην οποία αποθηκεύουμε συγκεντρωτικά τις διαδρομές (paths) για την πλοήγηση μεταξύ των σελίδων μας.

RolesBean.java

Είναι η κλάση της οποίας τα αντικείμενα χρησιμοποιούμε για να ορίσουμε και να ανακτήσουμε σε ποιον ρόλο ανήκει ο κάθε χρήστης.

ManagedBeans Package

Στο πακέτο αυτό έχουν δημιουργηθεί όλες οι κλάσεις που αλληλεπιδρούν με την κεντρική σελίδα και διαχειρίζονται τις λειτουργίες που εκτελεί ο χρήστης στην διαδικτυακή διεπαφή. Οι κλάσεις αυτές είναι κλάσεις διαχείρισης (Managed Beans) της τεχνολο-

γίας Java Server Faces (JSF) και επίσης επικοινωνούν με τα στοιχεία των Prime Faces που υλοποιούνται στην σελίδα “index.xhtml”. Συγκεκριμένα οι κλάσεις που έχουν δημιουργηθεί για αυτόν τον σκοπό είναι:

Register.java

Είναι υπεύθυνη για την εγγραφή των χρηστών στην βάση δεδομένων μας. Αλληλεπιδρά με την σελίδα εγγραφής “singup.xhtml” σύμφωνα με τους κανόνες της JSF τεχνολογίας και δέχεται από την ίδια σελίδα τα στοιχεία εγγραφής (όνομα χρήστη, κωδικό, ηλεκτρονικό ταχυδρομείο) του χρήστη. Σε αυτή τη κλάση η πιο βασική συνάρτηση είναι η “RegisterUser” (Εικόνα 47) που εκτελεί την λειτουργία της εγγραφής του χρήστη.

```
68 public String RegisterUser() throws InterruptedException{
69     boolean usernameExist = mTalkDB.UsernameExist(username);
70     boolean emailExist = mTalkDB.EmailExist(email);
71     if(!emailExist && !usernameExist){
72         if(mTalkDB.UserRegistration(username, password, email)){
73             RequestContext.getCurrentInstance().update("growl");
74             FacesContext context = FacesContext.getCurrentInstance();
75             context.getExternalContext().getFlash().setKeepMessages(true);
76             context.addMessage("growl",
77                 new FacesMessage(FacesMessage.SEVERITY_INFO,
78                     "Registration Successfull",
79                     "Log In with your username and password"));
80             return navigationBean.toLogin();
81         }
82         RequestContext.getCurrentInstance().update("growl");
83         FacesContext context = FacesContext.getCurrentInstance();
84         context.addMessage(null,
85             new FacesMessage(FacesMessage.SEVERITY_ERROR,
86                 "Error",
87                 "Registration Faild. DataBase Error Occurred. "
88                     + "Please Try Later"));
89         return "";
90     }else{
91         RequestContext.getCurrentInstance().update("growl");
92         FacesContext context = FacesContext.getCurrentInstance();
93         context.addMessage(null,
94             new FacesMessage(FacesMessage.SEVERITY_ERROR,
95                 "Error", "User already exists. Please select "
96                     + "another Username or E-mail"));
97         return "";
98     }
```

Εικόνα 47: Συνάρτηση RegisterUser.

Login.java

Πραγματοποιεί τη σύνδεση και αποσύνδεση των εγγεγραμμένων χρηστών στην εφαρμογή μας. Επίσης σε αυτή τη κλάση αποφασίζεται σε πιο ρόλο ανήκουν (απλός χρήστης ή διαχειριστής) οι χρήστες κατά τη σύνδεσή τους, ανάλογα με τα διαπιστευτήρια που παραθέτουν. Οι δύο βασικότερες συναρτήσεις που υλοποιούνται στη κλάση αυτή είναι:

- **LoginControl** (Εικόνα 48):

```

82 public String LoginControl() {
83     if (loggedIn) {
84         RequestContext.getCurrentInstance().update("growl");
85         FacesContext context = FacesContext.getCurrentInstance();
86         context.addMessage(null, new FacesMessage(FacesMessage.
87             SEVERITY_WARN, "Warning", "You Are Already Logged In!!! "
88             + "<br> <a href='\"/MyWIFISpeed/faces/secured/\"
89             + \"index.xhtml\">Home Page</a> <br> "
90             + "<a href='\"/MyWIFISpeed/faces/secured/\"
91             + \"/index.xhtml?logout=true\">Log Out</a>"));
92         return "";
93     }
94     else if (mTalkDB.AdministratorLoginControl(username, password)) {
95         loggedIn = true;
96         rolesBean.setAdminRole(true);
97         rolesBean.setUsername(username);
98         return navigationBean.redirectToIndex();
99     }
100     else if (mTalkDB.UserLoginControl(username, password)) {
101         loggedIn = true;
102         rolesBean.setUserRole(true);
103         rolesBean.setUsername(username);
104         return navigationBean.redirectToIndex();
105     }
106     RequestContext.getCurrentInstance().update("growl");
107     FacesContext context = FacesContext.getCurrentInstance();
108     context.addMessage(null, new FacesMessage(FacesMessage.SEVERITY_ERROR,
109         "Error", "Invalid Username or Password!!!"));
110     return "";
111 }

```

Εικόνα 48: Συνάρτηση LoginControl

- **LogoutControl** (Εικόνα 49):

```

111 public String LogoutControl() {
112     loggedIn = false;
113     if (rolesBean.isAdminRole()) {
114         rolesBean.setAdminRole(false);
115     } else if (rolesBean.isUserRole()) {
116         rolesBean.setUserRole(false);
117     }
118
119     RequestContext.getCurrentInstance().update("growl");
120     FacesContext context = FacesContext.getCurrentInstance();
121     context.getExternalContext().getFlash().setKeepMessages(true);
122     context.addMessage("growl", new FacesMessage(FacesMessage.
123         SEVERITY_INFO, "Logged Out Successfully", ""));
124
125     return navigationBean.redirectToLogin();
126 }
127

```

Εικόνα 49: Συνάρτηση LogoutControl

ManageUserAccount.java

Υλοποιούνται όλες οι συναρτήσεις που προσφέρουν την δυνατότητα στον χρήστη, ανεξαρτήτως ρόλου, να διαχειριστεί τον λογαριασμό του. Οι δύο βασικότερες συναρτήσεις που υπάρχουν σε αυτή τη κλάση είναι η “UpdateUsername” (Εικόνα 50) για την αλλαγή του ονόματος χρήστη και η “UpdatePassword” για την αλλαγή του κωδικού πρόσβασης του χρήστη.


```

58 public void UpdateUsername(ActionEvent e){
59     boolean usernameExist = mTalkDB.UsernameExist(username);
60     if(!usernameExist){
61         if(rolesBean.isAdminRole()){
62             if(mTalkDB.UpdateAdministratorUsername(rolesBean.getUsername(),
63                 username)){
64                 rolesBean.setUsername(username);
65                 RequestContext.getCurrentInstance().
66                     update("updateAccountGrowl");
67                 RequestContext.getCurrentInstance().update("mainForm");
68                 FacesContext context = FacesContext.getCurrentInstance();
69                 context.addMessage(null, new FacesMessage(FacesMessage.
70                     SEVERITY_INFO, "Your Username Changed", ""));
71             }
72         }
73         else if(rolesBean.isUserRole()){
74             if(mTalkDB.UpdateUserUsername(rolesBean.getUsername(),
75                 username)){
76                 rolesBean.setUsername(username);
77                 RequestContext.getCurrentInstance().
78                     update("updateAccountGrowl");
79                 RequestContext.getCurrentInstance().update("mainForm");
80                 FacesContext context = FacesContext.getCurrentInstance();
81                 context.addMessage(null, new FacesMessage(FacesMessage.
82                     SEVERITY_INFO, "Your Username Changed", ""));
83             }
84         }
85     }else{
86         RequestContext.getCurrentInstance().update("updateAccountGrowl");
87         FacesContext context = FacesContext.getCurrentInstance();
88         context.addMessage(null, new FacesMessage(FacesMessage.
89             SEVERITY_ERROR, "Error", "Username already exists. "
90             + "Please select another one"));
91     }
92 }

```

Εικόνα 50: Συνάρτηση UpdateUsername.

AdminManagement.java

Υλοποιούνται όλες οι συναρτήσεις που προσφέρουν την δυνατότητα στον διαχειριστή της εφαρμογής μας να μπορεί να διαχειρίζεται τους λογαριασμούς των υπόλοιπων χρηστών που ανήκουν στον ίδιο ρόλο. Οι βασικότερες συναρτήσεις που προσφέρουν τις δυνατότητες αυτές και συμπεριλαμβάνονται σε αυτή τη κλάση είναι οι:

- **GetAllAdmins:** επιστρέφει όλους τους διαχειριστές.
- **RegisterAdmin:** παρέχει τη δυνατότητα στον διαχειριστή να δημιουργήσει έναν νέο χρήστη αναθέτοντας του τον ρόλο του διαχειριστή.
- **DeleteAdmin:** παρέχει τη δυνατότητα στον διαχειριστή να διαγράψει έναν χρήστη που κατέχει τον ρόλο του διαχειριστή.
- **SaveAdminChanges:** παρέχει τη δυνατότητα στον διαχειριστή να εκτελεί αλλαγή των στοιχείων ενός χρήστη, που κατέχει τον ρόλο του διαχειριστή.
- **AjaxSearchAdmins:** ασύγχρονη αναζήτηση των χρηστών που έχουν τον ρόλο του διαχειριστή.
- **SearchAdmins:** αναζήτηση ενός διαχειριστή.

UserManagement.java

Είναι κλάση που προσφέρει τις ίδιες λειτουργίες και είναι παρόμοια υλοποιημένη με την κλάση “AdminManagement.java”. Η μόνη διαφορά μεταξύ των δύο αυτών κλάσεων είναι πως η “UserManagement.java” προσφέρει στον διαχειριστή τις λειτουργίες για την διαχείριση των λογαριασμών των χρηστών που κατέχουν τον ρόλο του απλού χρήστη, ενώ η “AdminManagement.java” των χρηστών που κατέχουν τον ρόλο του διαχειριστή.

StatisticsManagement.java

Υλοποιούνται όλες οι συναρτήσεις που υποστηρίζουν τη δυνατότητα παραγωγής στατιστικών στοιχείων, για τις μετρήσεις που έχουν γίνει από τις κινητές συσκευές ,και εμφάνισης τους στην σελίδα “index.xhtml”.

LiveStatisticsManagement.java

Υλοποιούνται όλες οι συναρτήσεις που υποστηρίζουν τη δυνατότητα παραγωγής στατιστικών στοιχείων **σε πραγματικό χρόνο**, για τις μετρήσεις που έχουν γίνει από τις κινητές συσκευές, και εμφάνισης τους στην σελίδα “index.xhtml”.

CampaignManagement.java

Στην κλάση αυτή υλοποιούνται όλες οι συναρτήσεις εκείνες που παρέχουν τις λειτουργίες για να διαχειριστούμε τις καμπάνιες και να τις στείλουμε στις κινητές συσκευές. Οι σημαντικότερες συναρτήσεις της κλάσης αυτής είναι:

- **SaveCampaign:** προσφέρει τη δυνατότητα να αποθηκεύσουμε προσωρινά τις καμπάνιες, που δημιουργεί ο διαχειριστής μέσω της διαδικτυακής εφαρμογής, σε μια λίστα στη μνήμη του εξυπηρετητή μας.
- **SendCampaign (Εικόνα 51):** στέλνει τις καμπάνιες που έχει δημιουργήσει ο διαχειριστής και που υπάρχουν στη βάση δεδομένων, στον GCM application server. Τα βήματα που ακολουθούμε σε αυτή τη συνάρτηση είναι τα εξής:
 1. Αν υπάρχουν καμπάνιες που είναι αποθηκευμένες στην προσωρινή λίστα του εξυπηρετητή μας τότε τις αποθηκεύουμε στην βάση δεδομένων
 2. Θέτουμε ως μη ενεργές τις καμπάνιες που υπάρχουν στη βάση δεδομένων μας και έχει παρέλθει η ημερομηνία λήξης τους.
 3. Ανακτούμε τη λίστα με τις ενεργές καμπάνιες από την βάση δεδομένων μας.
 4. Μορφοποιούμε τη λίστα με τις καμπάνιες μας κατάλληλα προς αποστολή
 5. Μετατρέπουμε τη μορφοποιημένη λίστα με τις καμπάνιες μας σε Json αντικείμενο.

6. Στέλνουμε το Json αντικείμενο στον GCM application server.

```
1 public void SendCampaign(ActionEvent e) throws Exception {
2     if(!this.campaignsForSending.isEmpty()){
3         mTalkDB.InsertCampaignsInDB(this.campaignsForSending);
4         this.campaignsForSending.clear();
5     }
6
7     mTalkDB.DeactiveExpiredCampaigns();
8     GetCampaignsToShow();
9
10    List<Campaigns> mCampaignsList = mTalkDB.GetCampaignsByActive(true);
11
12    if (mCampaignsList != null) {
13        //format sql date
14        DateFormat createDateFormat = new SimpleDateFormat("yyyy-MM-dd hh:mm:ss");
15        DateFormat endDateFormat = new SimpleDateFormat("yyyy-MM-dd");
16
17        List<FormattedCampaigns> formattedCampaignsList = new ArrayList<>();
18        for (Campaigns camp : mCampaignsList) {
19            FormattedCampaigns mFormattedCampaigns = new FormattedCampaigns();
20            mFormattedCampaigns.setCamp_id(camp.getCampId());
21            mFormattedCampaigns.setName(camp.getName());
22            mFormattedCampaigns.setCreate_date(createDateFormat.format(camp.getCreateDate()));
23            mFormattedCampaigns.setEnd_date(endDateFormat.format(camp.getEndDate()));
24            mFormattedCampaigns.setLatitude(camp.getLatitude());
25            mFormattedCampaigns.setLongitude(camp.getLongitude());
26            mFormattedCampaigns.setNetwork_type(camp.getNetworkType());
27            mFormattedCampaigns.setAccuracy(camp.getAccuracy());
28            mFormattedCampaigns.setRadius(camp.getRadius());
29            mFormattedCampaigns.setVelocity(camp.getVelocity());
30            mFormattedCampaigns.setDist_frequency(camp.getDistFrequency());
31
32            formattedCampaignsList.add(mFormattedCampaigns);
33        }
34
35        Gson mGson = new Gson();
36        String jsonCampaign = "{\"campaigns\": "+ mGson.toJson(formattedCampaignsList)+"}";
37        String gcmAnswer = SendHttpRequest(jsonCampaign);
38
39        RequestContext.getCurrentInstance().update("growl");
40        FacesContext context = FacesContext.getCurrentInstance();
41        context.addMessage(null, new FacesMessage(FacesMessage.SEVERITY_INFO, "Campaigns Se
42    } else {
43        RequestContext.getCurrentInstance().update("growl");
44        FacesContext context = FacesContext.getCurrentInstance();
45        context.addMessage(null, new FacesMessage(FacesMessage.SEVERITY_INFO, "No Campaigns
46    }
47 }
```

Εικόνα 51: Συνάρτηση SendCampaign.

- **SendHttpRequest(Εικόνα 52):** πραγματοποιεί την HTTP κλήση προς τον GCM application server, για να αποσταλούν οι καμπάνιες.

```
218 private String SendHttpRequest(String msgToSend) throws Exception {
219     URL url = new URL("http://83.212.113.47:8080/GCMserver/sendAll");
220     URLConnection conn = url.openConnection();
221     conn.setDoOutput(true);
222     BufferedReader reader;
223     String returnGCMMsg = "";
224     try (OutputStreamWriter writer = new OutputStreamWriter(conn.
225         getOutputStream())) {
226         writer.write("msg="+msgToSend);
227         writer.flush();
228         String line;
229         reader = new BufferedReader(new InputStreamReader(conn.
230             getInputStream()));
231         while ((line = reader.readLine()) != null) {
232             System.out.println(line);
233             returnGCMMsg += line;
234         }
235     }
236     reader.close();
237     return returnGCMMsg;
238 }
```

Εικόνα 52: Συνάρτηση SendHttpRequest.

Loginfilter.java

Είναι η πρώτη κλάση που καλείται όταν κάποιος επισκέπτεται την εφαρμογή μας. Ο σκοπός της ύπαρξης της είναι η αποφυγή εισόδου μη συνδεδεμένων χρηστών στην κεντρική σελίδα μας, index.xhtml (βλ. πιο κάτω). Όπως μπορούμε να δούμε και από τον κώδικα στην Εικόνα 49 ελέγχουμε μέσω της συνάρτησης isLoggedIn της κλάσης Login αν ο χρήστης είναι συνδεδεμένος. Αν ναι τότε τον κατευθύνουμε στην κεντρική μας σελίδα. Αν όχι ελέγχουμε αν ο χρήστης έχει στείλει αίτημα για να κάνει αποσύνδεση. Αν ναι (ο χρήστης αιτείται αποσύνδεση) τότε αποσυνδέεται κάνοντας χρήση των συναρτήσεων setLoggedIn και setAdminRole ή setUserRole και κατευθύνεται στην σελίδα σύνδεσης. Αν και αυτός ο έλεγχος (αίτημα του χρήστη για αποσύνδεση) δεν είναι αληθής τότε εμφανίζουμε στον χρήστη τη σελίδα σύνδεσης.

```
36  @Override
37  public void doFilter(ServletRequest request, ServletResponse response,
38                      FilterChain chain) throws IOException, ServletException {
39      // Get the loginBean from session attribute
40      Login loginController = (Login)((HttpServletRequest)request).
41          getSession().getAttribute("login");
42
43      // For the first application request there is no loginBean in the
44      // session so user needs to log in
45      // For other requests loginBean is present but we need to check if user
46      // has logged in successfully
47      if (loginController == null || !loginController.isLoggedIn()) {
48          String contextPath = ((HttpServletRequest)request).getContextPath();
49          ((HttpServletResponse)response).sendRedirect(contextPath +
50              "/faces/verification/login.xhtml");
51          return;
52      }
53
54      String logOutRequest = request.getParameter("logOut");
55      if(logOutRequest != null && logOutRequest.equals("true")){
56          loginController.setLoggedIn(false);
57          if(loginController.getRolesBean().isAdminRole()){
58              loginController.getRolesBean().setAdminRole(false);
59          }else if(loginController.getRolesBean().isUserRole()){
60              loginController.getRolesBean().setUserRole(false);
61          }
62          String contextPath = ((HttpServletRequest)request).getContextPath();
63          ((HttpServletResponse)response).sendRedirect(contextPath +
64              "/faces/verification/login.xhtml");
65          return;
66      }
67
68      chain.doFilter(request, response);
69  }
70 }
```

Εικόνα 53: Loginfilter.java κύρια συνάρτηση.

singup.xhtml

Σελίδα μέσω της οποίας ο χρήστης μπορεί να πραγματοποιήσει την εγγραφή του στην διαδικτυακή εφαρμογή. Η εγγραφή είναι ελεύθερη αρκεί ο χρήστης να ορίσει το όνομα χρήστη και κωδικό πρόσβασης που επιθυμεί να έχει και να δώσει τη διεύθυνση του ηλεκτρονικού ταχυδρομείου του.

login.xhtml

Σελίδα μέσω της οποίας ο χρήστης μπορεί να πραγματοποιήσει την σύνδεση του στην διαδικτυακή εφαρμογή. Το μόνο που χρειάζεται είναι να δώσει το όνομα χρήσης και τον κωδικό του που όρισε κατά την εγγραφή.

Index.xhtml

Είναι η κεντρική σελίδα η οποία εμφανίζεται στον χρήστη και αλληλεπιδρά μαζί του με σκοπό να παρέχει όλες τις λειτουργίες (χάρτης, στατιστικά στοιχεία, διαχείριση καμπάνιας, διαχείριση χρηστών) της διαδικτυακής εφαρμογής. Για την επίτευξη του σκοπού αυτού η “index.xhtml” κάνει χρήση των τεχνολογιών Java Server Faces και Java Primefaces. Επιπλέον χρησιμοποιεί όλες τις κλάσεις από το “ManagedBeans” πακέτο.

CSS

Στον φάκελο css που φαίνεται στην Εικόνα 44 συμπεριλαμβάνονται τα αρχεία με τον css κώδικα για την μορφοποίηση της κεντρικής σελίδας “Index.xhtml”.

Js

Στον φάκελο js, που μπορούμε να δούμε στην Εικόνα 44, συμπεριλαμβάνονται τα αρχεία με τον JavaScript – JQuery κώδικα τα οποία χρησιμοποιούνται από την “Index.xhtml” για την δυναμική λειτουργικότητα της. Επίσης μέσω των ίδιων αρχείων γίνεται χρήση της βιβλιοθήκης των OpenLayers για την εμφάνιση των χαρτών και η επικοινωνία με τον Geoserver. Συγκεκριμένα έχουν υλοποιηθεί δύο αρχεία JavaScript:

main.js

Περιέχει των JavaScript - JQuery κώδικα που χειρίζεται τις ενέργειες του χρήστη προσδίδοντας έτσι δυναμικότητα στα στοιχεία της “Index.xhtml”.

map.js

Όπως προδίδει και το όνομά του το αρχείο αυτό περιλαμβάνει τον κώδικα που είναι υπεύθυνο για την εμφάνιση και τη λειτουργία του χάρτη. Επίσης στο αρχείο αυτό γίνεται και η επικοινωνία με τον Geoserver. Συγκεκριμένα κάποιες από τις κυριότερες λειτουργίες που πραγματοποιούνται στο αρχείο αυτό είναι:

- Η χρήση των OpenLayers για την δημιουργία και την εμφάνιση του χάρτη στην “Index.xhtml” (Εικόνα 54).

```

17 $(document).ready(function () {
18     Map = new ol.Map({
19         view: new ol.View({
20             center: ol.proj.transform([23.728007884898, 38.100327431707],
21                 'EPSG:4326', 'EPSG:3857'),
22             zoom: 7
23         }),
24         layers: [
25             new ol.layer.Tile({
26                 source: new ol.source.MapQuest({layer: 'osm'})
27             }),
28             _CircleVectorLayer
29         ],
30         renderer: 'canvas',
31         target: 'map'
32     });

```

Εικόνα 54: Δημιουργία χάρτη

- Η δημιουργία και η εμφάνιση των Layers που περιέχουν την πληροφορία από τον Geoserver που πραγματοποιείται όπως ο κώδικας στην Εικόνα 55.
- Η δυνατότητα δημιουργίας καμπάνιας επιλέγοντας περιοχή στον χάρτη με γραφικό τρόπο (Εικόνα 56).

```

356 function ShowBarnesSurface(provider,style) {
357     ClearWmsBarnesSurfaceLayer();
358
359     var filter;
360     if (provider !== '' && provider !== 'wifi') {
361         filter = "Operator = '" + provider + "'";
362     } else if(provider === 'wifi'){
363         filter = "Network_Type = '" + provider + "'";
364     }else {
365         filter = '1=1';
366     }
367
368     WmsBarnesSurface = new ol.layer.Tile({
369         source: new ol.source.TileWMS({
370             url: 'http://83.212.113.47:8888/geoserver/cite/wms',
371             params:{
372                 LAYERS: 'cite:huaLocation',
373                 FORMAT: 'image/png',
374                 STYLES: style,
375                 CQL_FILTER: filter
376             },
377             serverType: 'geoserver'
378         })
379     });

```

Εικόνα 55: συνάρτηση δημιουργίας Layer πού περιέχει τον μετασχηματισμό του αλγόριθμου Barnes Surface, που στέλνει ο Geoserver.

```

68 function AddInteraction() {
69     if (!(_NewCampaignPanelIsOpen || _NewCampaignPanelIsMinimize) &&
70         _Draw !== null) {
71         return;
72     }
73     _Draw = new ol.interaction.Draw({
74         features: featureOverlay.getFeatures(),
75         type: ("Polygon")
76     });
77     _Map.addInteraction(_Draw);
78     _Map.addInteraction(modify);
79
80     _Draw.on('drawstart',
81         function (evt) {
82             }, this);
83
84     _Draw.on('drawend',
85         function (evt) {
86             var coords = _Draw.f[0][0];
87
88             var boundingExtentObj = ol.extent.boundingExtent(_Draw.f[0]);
89             var center = ol.extent.getCenter(boundingExtentObj);
90             var width = ol.extent.getWidth(boundingExtentObj);
91             var radius = width / 2;
92
93             var transformedCenter = ol.proj.transform([center[0],
94                 center[1]], 'EPSG:3857', 'EPSG:4326');
95             var lon = transformedCenter[0];
96             var lat = transformedCenter[1];
97
98             $(".newCampaignLatitude").val(lat.toFixed(4));
99             $(".newCampaignLongitude").val(lon.toFixed(4));
100             $(".newCampaignRadius").val(Math.round( radius ));
101         }, this);
102 }

```

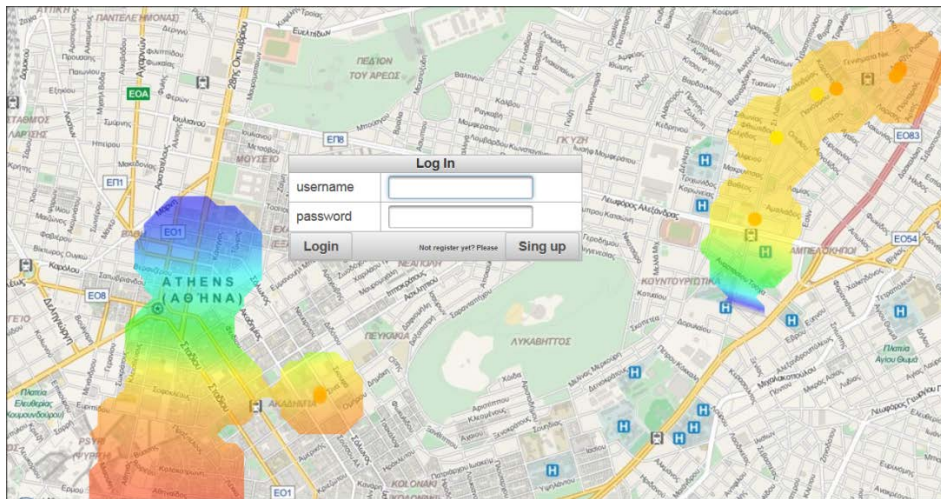
Εικόνα 56: Συνάρτηση γραφικής δημιουργίας καμπάνιας.

6.2 Οδηγός χρήσης

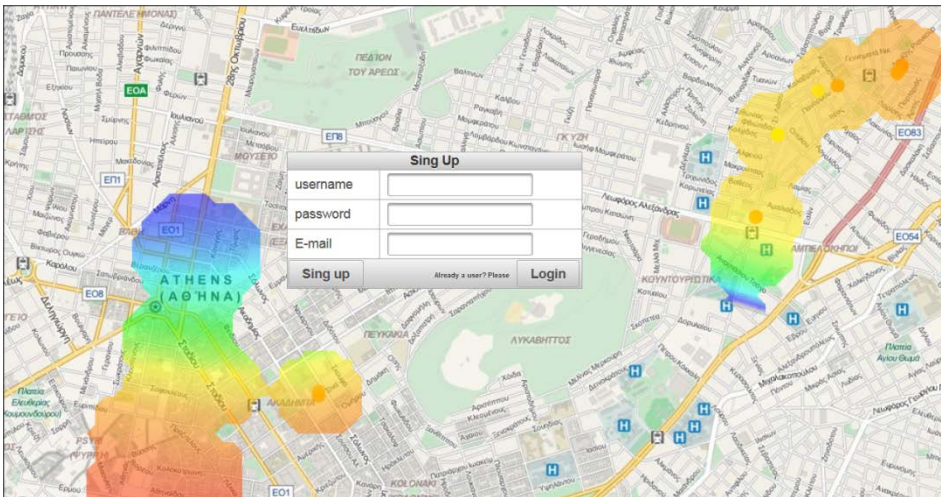
Η εφαρμογή MyWiFiSpeed είναι διαθέσιμη και μπορούμε να την επισκεφτούμε στην διεύθυνση <http://83.212.113.47:8484/MyWIFISpeed>

6.2.1 Σύνδεση – Εγγραφή (Log In – Sing Up)

Η πρώτη σελίδα που θα δει ο χρήστης κατά την πλοήγηση του είναι η σελίδα εισόδου (Log In) που φαίνεται στην Εικόνα 57. Στην σελίδα σύνδεσης μπορεί να εισάγει το όνομα χρήστη του και τον κωδικό εισόδου για να συνδεθεί στην διαδικτυακή εφαρμογή, ανεξάρτητα από τον ρόλο του. Από την άλλη αν ο χρήστης δεν είναι εγγεγραμμένος τότε για να εγγραφεί μπορεί να επιλέξει το κουμπί Sing Up ώστε να πλοηγηθεί στην σελίδα εγγραφής. Στην σελίδα εγγραφής μπορούν ελεύθερα να εγγραφούν μόνο χρήστες που θα έχουν τον ρόλο του απλού χρήστη, ορίζοντας μόνο ένα όνομα χρήσης, κωδικό εισόδου και έγκυρη διεύθυνση ηλεκτρονικού ταχυδρομείου. Την εγγραφή χρηστών με ρόλο διαχειριστή μπορεί να την πραγματοποιήσει μόνο ένας διαχειριστής μέσα από την εφαρμογή.



Εικόνα 57: Σελίδα σύνδεσης.

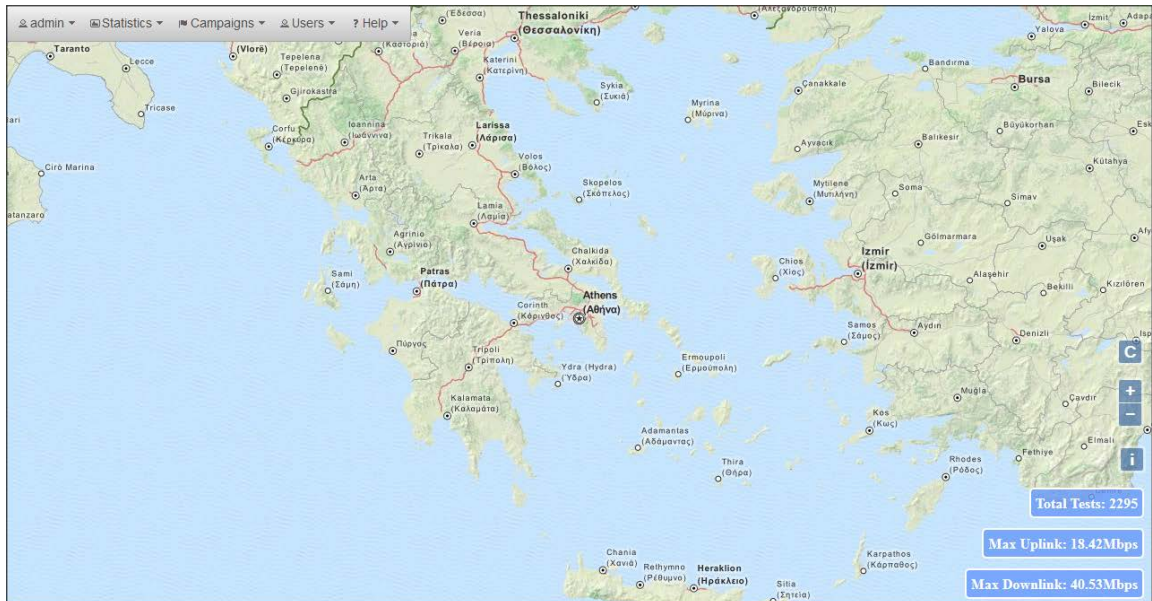


Εικόνα 58: Σελίδα εγγραφής.

6.2.2 Κεντρική Σελίδα (Home Page)

Είναι η σελίδα η οποία δίνει πρόσβαση στον χρήστη στο χάρτη, σε στατιστικά στοιχεία και στο κεντρικό μενού για την εκτέλεση των διάφορων λειτουργιών. Δηλαδή, όπως μπορούμε να δούμε και στην Εικόνα 59 ο χρήστης μπορεί να δει ως κεντρικό στοιχείο τον χάρτη και επίσης το κεντρικό μενού που βρίσκεται πάνω αριστερά. Επιπλέον στην κάτω δεξιά πλευρά υπάρχουν τα κουμπιά ελέγχου του χάρτη και το κουμπί με την υπόδειξη “C” που χρησιμεύει στον καθαρισμό του χάρτη από πιθανόν ενέργειες του χρήστη. Κάτω από τα κουμπιά ελέγχου του χάρτη υπάρχουν τρεις στατιστικές ενδείξεις που λειτουργούν σε πραγματικό χρόνο. Η πρώτη ένδειξη αφορά τις συνολικές μετρήσεις απόδοσης ασύρματων δικτύων που έχουν πραγματοποιηθεί από όλους τους παροχείς δικτύου, αλλά και από τον καθένα πάροχο τους ξεχωριστά. Η δεύτερη ένδειξη αφορά το μέγιστο ρυθμό ανεβάσματος σε Mbps για όλες τις μετρήσεις, αλλά και για τον

κάθε πάροχο ξεχωριστά. Παρόμοια, και η τρίτη ένδειξη αφορά τον μέγιστο ρυθμό ανεβάσματος σε Mbps για όλες τις μετρήσεις, αλλά και για τον κάθε πάροχο ξεχωριστά.



Εικόνα 59: Κεντρική σελίδα.

6.2.3 Διαχείριση Λογαριασμού Χρήστη (User Account Management)

Οι χρήστες που συνδέονται στην διαδικτυακή εφαρμογή μπορούν, ανεξαρτήτως ρόλου, να διαχειριστούν τον λογαριασμό τους (Εικόνα 60). Επιλέγοντας από το μενού την επιλογή με το όνομα χρήστη (στην περίπτωση μας admin) και στη συνέχεια την επιλογή “My Account” ο χρήστης μπορεί να αλλάξει το όνομα χρήστη και τον κωδικό εισόδου του. Για παράδειγμα για την αλλαγή του ονόματος χρήστη πρέπει να κάνουμε click πάνω στο όνομα χρήστη μας, να γράψουμε ένα νέο όνομα χρήστη και στη συνέχεια να επιλέξουμε το κουμπί επιβεβαίωσης δεξιά ή το κουμπί ακύρωσης. Αν επιλέξουμε το κουμπί επιβεβαίωσης τότε για την ολοκλήρωση της διαδικασίας θα πρέπει να κάνουμε click και στο κουμπί “Change Username”.

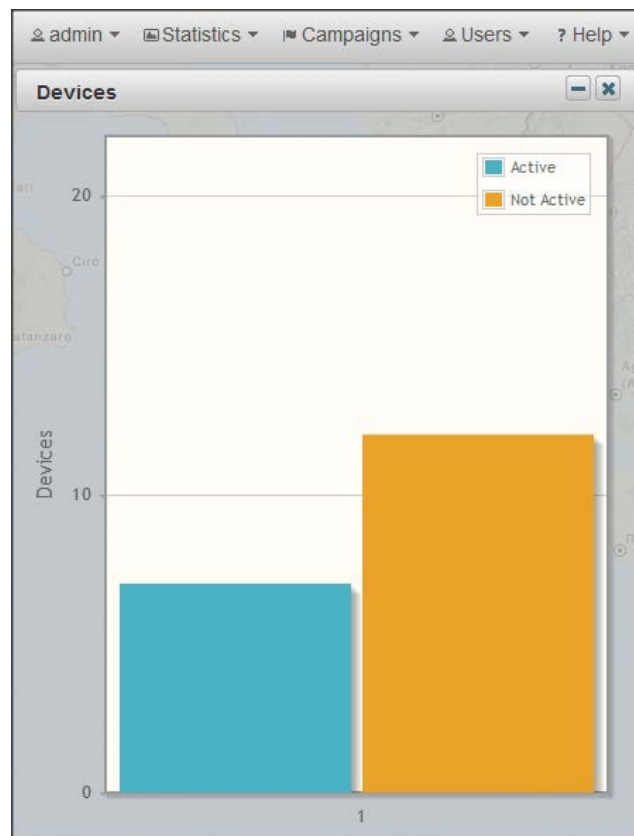
Εικόνα 60: Διαχείριση λογαριασμού χρήστη.

6.2.4 Στατιστικά (Statistics)

Το μενού “Statistics” προσφέρει τη δυνατότητα στους χρήστες να δουν, ανεξαρτήτως ρόλου, στατιστικά γραφήματα και στοιχεία πάνω στον χάρτη σχετικά με το εύρος ζώνης ασύρματων δικτύων, τις μετρήσεις και τις κινητές συσκευές που έχουν εγκατεστημένη την εφαρμογή HuaServiceTester2. Συγκεκριμένα κάτω από το μενού “Statistics” υπάρχουν τα εξής υπομενού:

Devices

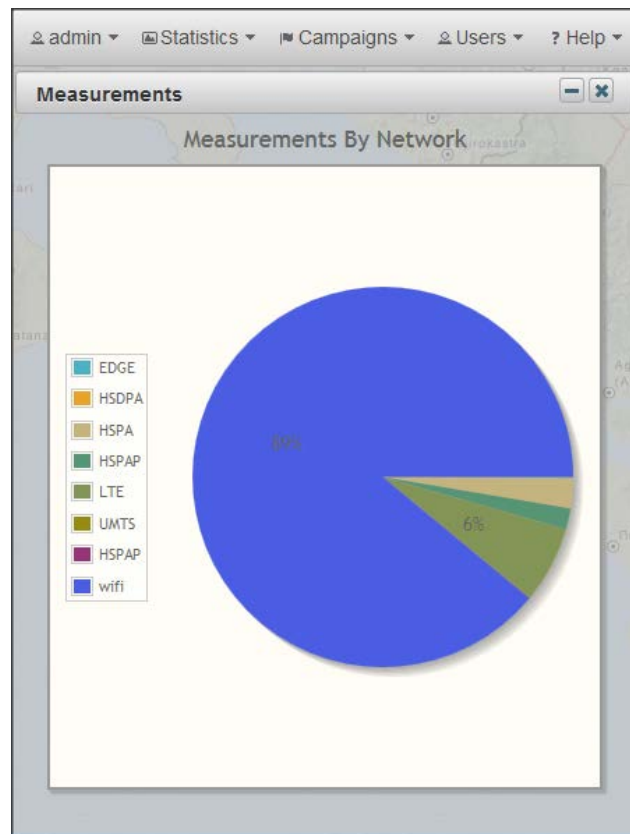
Προσφέρει το γράφημα που παρουσιάζεται στην Εικόνα 61 και το οποίο πληροφορεί τον χρήστη για τις συνολικές ενεργές και ανενεργές συσκευές που έχουν κάνει μετρήσεις.



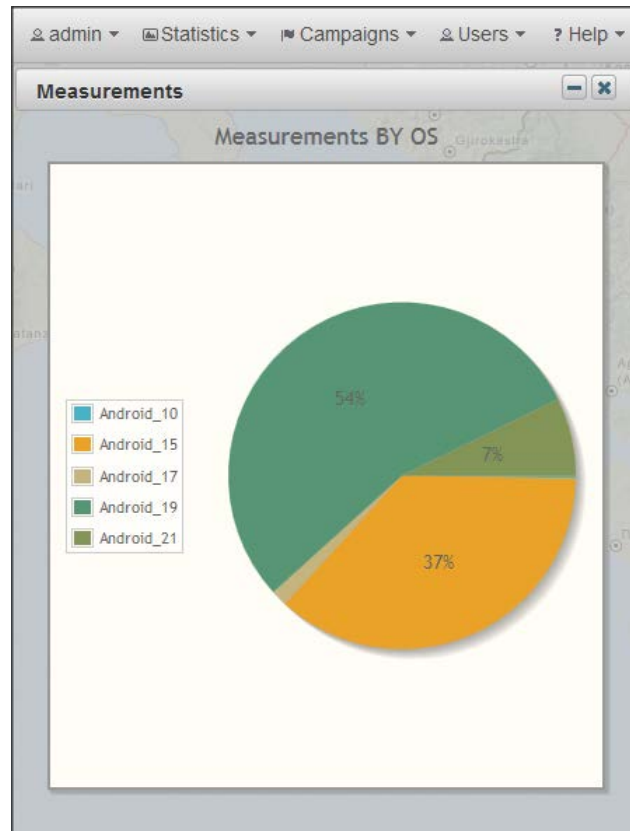
Εικόνα 61: Γράφημα συνολικών ενεργών και ανενεργών συσκευών.

Measurements

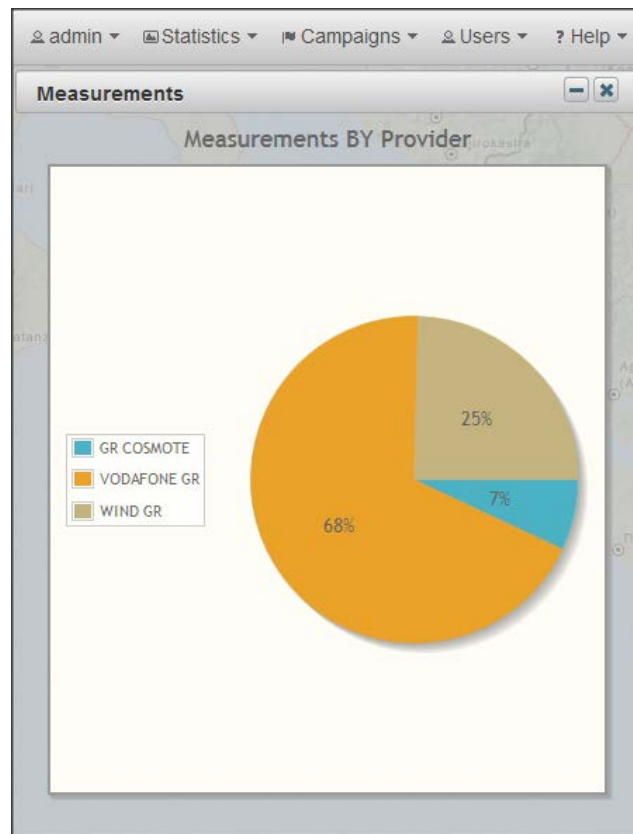
Προσφέρει γραφήματα για το ποσοστό μετρήσεων που έχουν πραγματοποιηθεί από τις κινητές συσκευές ανά τύπο δίκτυο (Εικόνα 62), ανά λειτουργικό σύστημα (Εικόνα 63) και ανά πάροχο δικτύου (Εικόνα 64).



Εικόνα 62: Γράφημα συνολικού ποσοστού μετρήσεων ανά τύπο δικτύου.



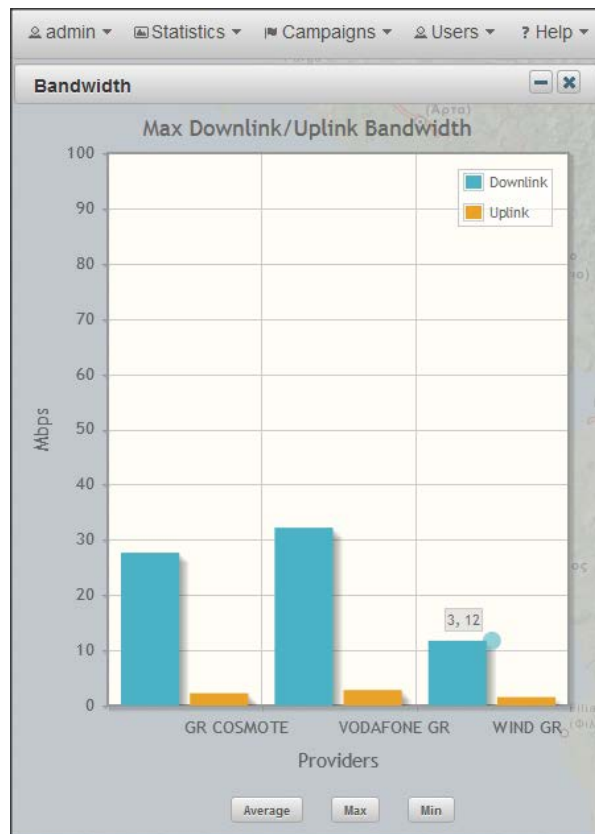
Εικόνα 63: Γράφημα συνολικού ποσοστού μετρήσεων ανα λειτουργικό σύστημα.



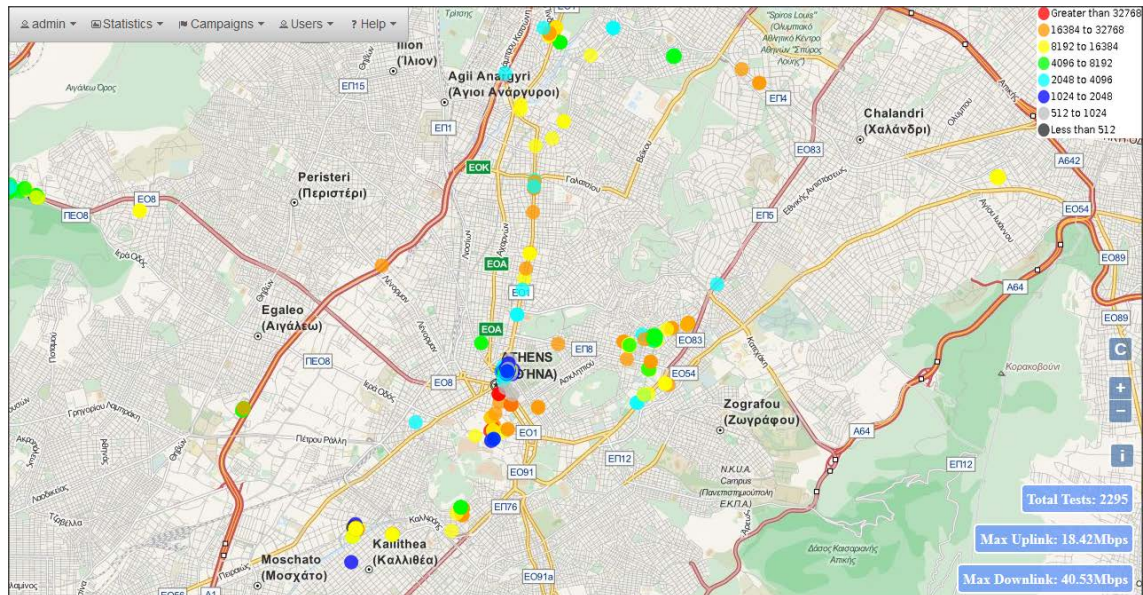
Εικόνα 64: Γράφημα συνολικού ποσοστού μετρήσεων ανα πάροχο δικτύου.

Bandwith

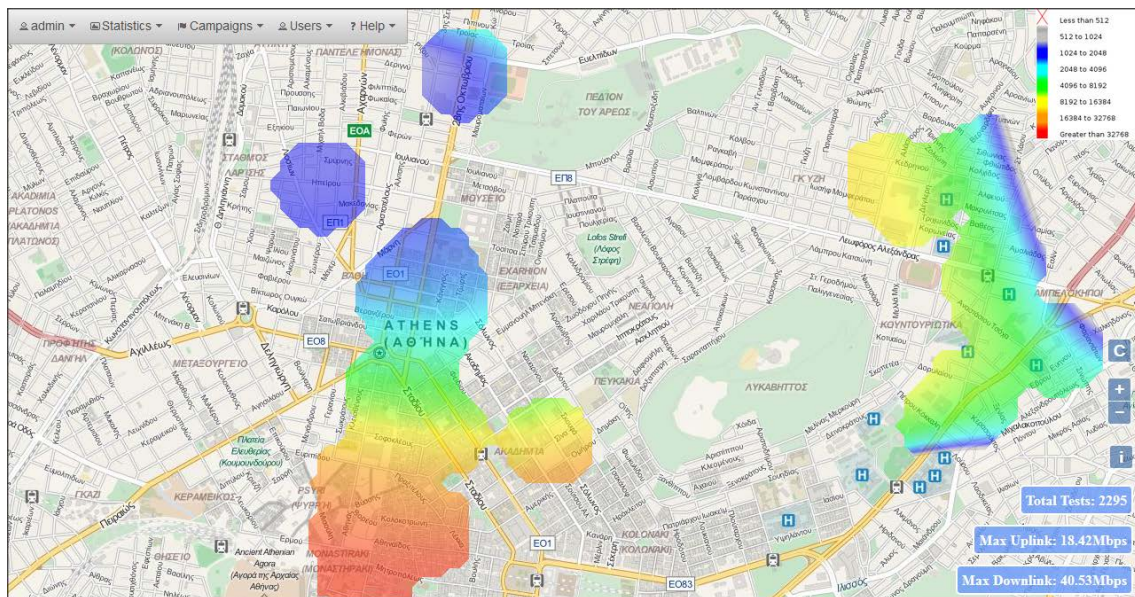
Προσφέρει τη δυνατότητα αναπαράστασης στατιστικών στοιχείων για την απόδοση των ασύρματων δικτύων σε γραφήματα και στον χάρτη. Αναλυτικότερα στα γραφήματα αναπαρίστανται η μέγιστη, ελάχιστη και μέση ταχύτητα κατεβάσματος (downlink) και ανεβάσματος (Uplink) για τους τελευταίους μήνες, ανά τύπο δικτύου και ανά πάροχο δικτύου όπως φαίνεται για παράδειγμα στην Εικόνα 65. Από την άλλη στον χάρτη αναπαρίστανται τα σημεία που έχουν γίνει οι μετρήσεις δείχνοντας τις ταχύτητες ανεβάσματος, κατεβάσματος, ping του εξυπηρετητή της google και ping στον δικό μας εξυπηρετητή (Εικόνα 66). Επιπλέον στον χάρτη αναπαρίστανται οι ταχύτητες κατεβάσματος, ανεβάσματος και ping όπως έχουν προκύψει από την εφαρμογή του μετασχηματισμού Barnes Surface (Εικόνα 67).



Εικόνα 65: Γράφημα μέγιστου εύρους ζώνης ανα πάροχο δικτύου.



Εικόνα 66: Ταχύτητα κατεβάσματος ανά σημείο μέτρησης.



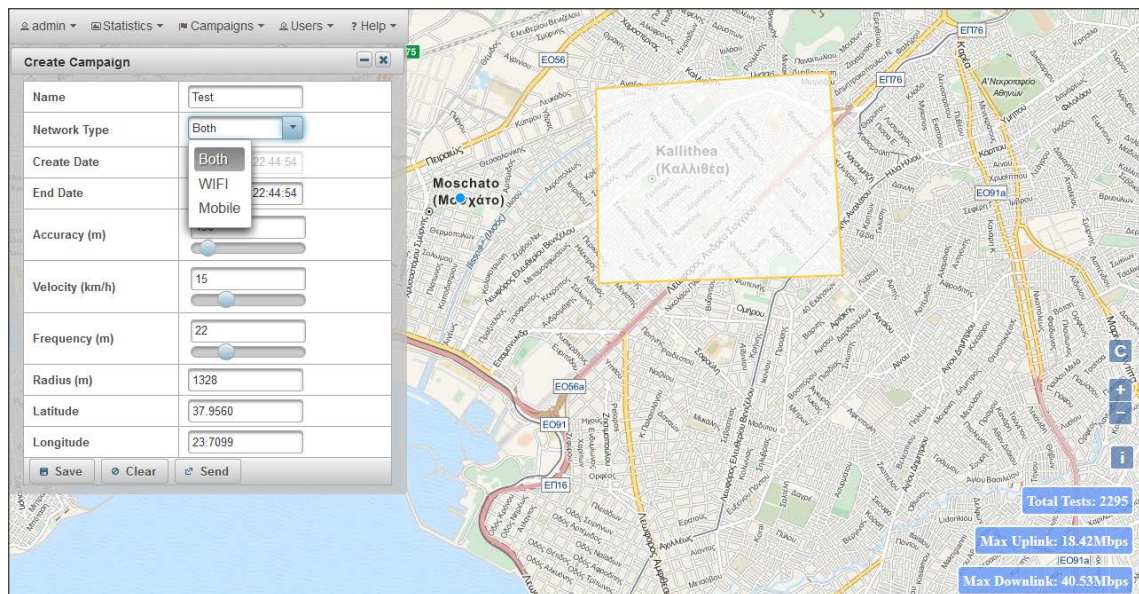
Εικόνα 67: Ταχύτητα κατεβάσματος ύστερα από την εφαρμογή του μετασχηματισμού Barnes Surface.

6.2.5 Campaigns

Μέσω του μενού “Campaigns” οι χρήστες που ανήκουν στον ρόλο του διαχειριστή μπορούν να δημιουργήσουν και να διαχειριστούν τις καμπάνιες που έχουν σταλεί στις κινητές συσκευές. Συγκεκριμένα κάτω από το μενού “Statistics” υπάρχουν τα εξής υπομενού:

New

Μέσω του υπομενού αυτού ο διαχειριστής μπορεί να δημιουργήσει τις νέες καμπάνιες που επιθυμεί να στείλει στις κινητές συσκευές (Εικόνα 68). Συγκεκριμένα για τη δημιουργία μιας νέας καμπάνιας ο διαχειριστής θα πρέπει να συμπληρώσει τη λίστα τιμών που φαίνεται στην Εικόνα 68 σύμφωνα με τις οδηγίες που παρουσιάζει ο Πίνακας 4, να επιλέξει την περιοχή που επιθυμεί στον χάρτη και να στείλει τις καμπάνιες επιλέγοντας το κουμπί “Send”. Επιπλέον αν ο διαχειριστής επιθυμεί να δημιουργήσει πάνω από μια καμπάνιες τότε, αφού συμπληρώσει τις τιμές και επιλέξει την περιοχή στον χάρτη, θα πρέπει να επιλέξει το κουμπί “Save” για να σώσει την καμπάνια προσωρινά. Στη συνέχεια και αφού επαναλάβει αυτή τη διαδικασία για κάθε καμπάνια που θέλει να δημιουργήσει μπορεί να επιλέξει το κουμπί “Send” για να αποστείλει όλες τις νέες καμπάνιες που δημιούργησε, αλλά και τις ενεργές καμπάνιες που υπάρχουν στην βάση δεδομένων.



Εικόνα 68: Μενού για την δημιουργία νέας καμπάνιας.

Πίνακας 4: Λίστα τιμών για τις καμπάνιες.

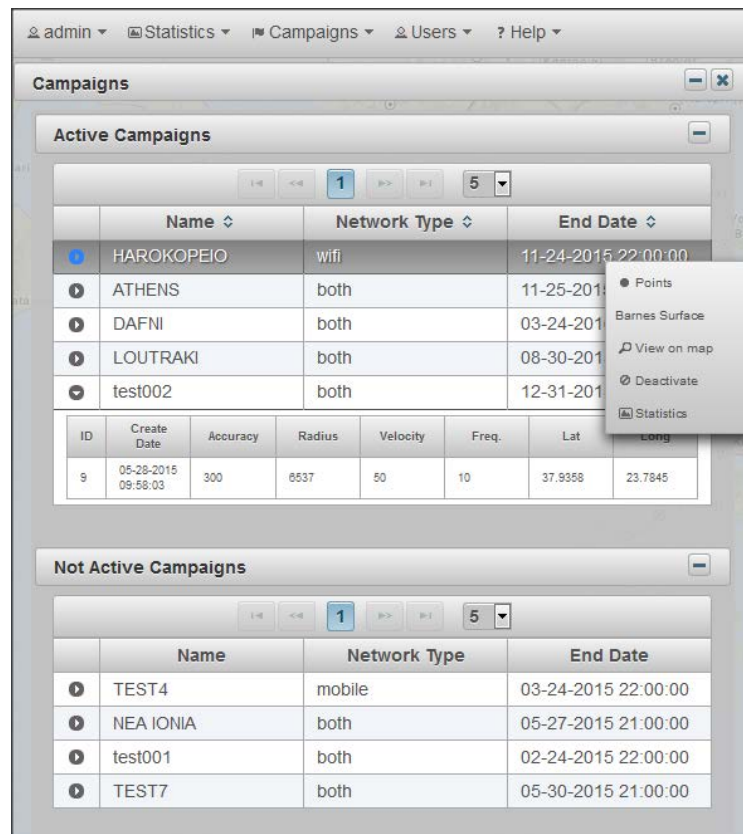
Όνομα Πεδίου	Περιγραφή	Τύπος	Απαιτείτε
Όνομα/Name	Το όνομα της καμπάνιας	Αλφαριθμητικό (String)	Ναι
Τύπος δικτύου/Network Type	Τύπος δικτύου για το οποίο θα λαμβάνονται μετρήσεις	Επιλογές: {Both, WIFI, Mobile}	Ναι
Ημερομηνία δημιουργίας/Create Date	Ημερομηνία έναρξης της καμπάνιας. Η τιμή αυτή ορίζεται αυτόματα κατά τη δημιουργία της καμπάνιας.	Ημερομηνία (Date)	Ναι
Ημερομηνία λήξης/ End Date	Ημερομηνία λήξης της καμπάνιας	Ημερομηνία (Date)	Ναι
Ακρίβεια/Accuracy	Ακρίβεια στον εντοπισμό της θέσης του χρήστη από την οποία θα γίνονται δεκτές οι μετρήσεις.	Αριθμός σε μέτρα (integer in meters)	Ναι
Ταχύτητα/Velocity	Μέγιστη ταχύτητα με την οποία ο χρήστης μπορεί να μετακινείται και να λαμβάνει μετρήσεις.	Αριθμός σε χιλ./ώρα (number in km/h)	Ναι
Συχνότητα/Frequency	Κάθε πόση απόσταση θα πρέπει να λαμβάνει μετρήσεις ο χρήστης	Αριθμός σε μέτρα (Number in meters)	Ναι
Ακτίνα/Radius	Η ακτίνα της περιοχής	Αριθμός σε μέτρα (Num-	Ναι

	που έχει επιλεγεί για την καμπάνια. Η τιμή αυτή ορίζεται αυτόματα με την επιλογή της περιοχής στον χάρτη.	ber in meters)	
Γεωγραφικό πλάτος/Latitude	Το γεωγραφικό πλάτος του κέντρου της περιοχής που έχει επιλεγεί για την καμπάνια. Η τιμή αυτή ορίζεται αυτόματα με την επιλογή της περιοχής στον χάρτη.	Αριθμός (Number)	Ναι
Γεωγραφικό μήκος/ Longitude	Το γεωγραφικό μήκος του κέντρου της περιοχής που έχει επιλεγεί για την καμπάνια. Η τιμή αυτή ορίζεται αυτόματα με την επιλογή της περιοχής στον χάρτη.	Αριθμός (Number)	Ναι

See All

Από το υπομενού αυτό ο διαχειριστής μπορεί να δει όλες τις ενεργές και ανενεργές καμπάνιες και να τις διαχειριστεί. Δηλαδή όπως μπορούμε να δούμε και στην Εικόνα 69 ο διαχειριστής μπορεί να δει:

- Πληροφορίες για τις καμπάνιες.
- Ποιες είναι ενεργές και ποιες είναι ανενεργές.
- Την περιοχή της κάθε καμπάνιας στον χάρτη (κάνοντας δεξί click πάνω στην καμπάνια και επιλέγοντας την επιλογή “View on map”).
- Στατιστικά στοιχεία.
- Τις ταχύτητες κατεβάσματος, ανεβάσματος και ring ανά μέτρηση (όπως στο υπομενού Στατιστικά)
- Τις ταχύτητες κατεβάσματος, ανεβάσματος και ring όπως προέκυψαν μετά την εφαρμογή του μετασχηματισμού Barnes Surface (όπως στο υπομενού Στατιστικά)
- Να θέσει ως ανενεργές ή ως ενεργές όποιες καμπάνιες επιθυμεί.



Εικόνα 69: Μενού διαχείρισης καμπάνιας.

6.2.6 Users

Το τελευταίο μενού που συναντάμε είναι το μενού “Users” μέσω του οποίου οι χρήστες που ανήκουν στον ρόλο του διαχειριστή μπορούν να δημιουργήσουν και να διαχειριστούν τους υπόλοιπους χρήστες της εφαρμογής ανεξάρτητα σε ποιον ρόλο ανήκουν. Συγκεκριμένα ο διαχειριστής μπορεί να δει (Εικόνα 70), δημιουργήσει (Εικόνα 71) και να επεξεργαστεί (Εικόνα 72) τους λογαριασμούς των απλών χρηστών αλλά και των άλλων διαχειριστών.

admin ▾ Statistics ▾ Campaigns ▾ Users ▾ ? Help ▾	
Users	
Username	E-mail
user	user@hua.gr
vdalakas	vdalakas@hua.gr
test	test@hua.gr

Εικόνα 70: Εμφάνιση όλων των χρηστών.

admin ▾ Statistics ▾ Campaigns ▾ Users ▾ ? Help ▾

Register User [minimize] [close]

username	<input type="text"/>
password	<input type="password"/>
E-mail	<input type="text"/>

+ Add User

Εικόνα 71: Δημιουργία χρήστη.

admin ▾ Statistics ▾ Campaigns ▾ Users ▾ ? Help ▾

Edit User [minimize] [close]

Username/E-mail: **Find**

Username	test
Password	test
E-mail	test@hua.gr

Delete User **Save Changes**

Εικόνα 72: Επεξεργασία χρήστη.

7 Συμπεράσματα

Με την ολοκλήρωση της παρούσας διπλωματικής εργασίας καταλήξαμε στο συμπέρασμα πως το πρόβλημα του συντονισμού, της σύντηξης και της μέτρησης της απόδοσης των ασύρματων δικτύων είναι πολυδιάστατο και για την επίλυση του απαιτείται συνεργασία πολλών τεχνολογιών. Στο συμπέρασμα αυτό καταλήξαμε αφού πρώτα εντοπίσαμε το πρόβλημα, πραγματοποιήσαμε έρευνα, μελετήσαμε την βιβλιογραφία και καταγράψαμε τις διαθέσιμες τεχνολογίες για την ανάπτυξη μιας διαδικτυακής πλατφόρμας που να προσφέρει τη λύση στο παραπάνω πρόβλημα. Το πρόβλημα και οι διαθέσιμες τεχνολογίες, έτσι όπως προέκυψαν από την μελέτη μας, παρουσιάστηκαν αναλυτικά στα παραπάνω κεφάλαια. Επιπλέον στα κεφάλαια 1 – 6 δόθηκε η τεκμηρίωση βάση της οποίας επιλέχτηκαν οι τεχνολογίες και ο τρόπος υλοποίησης και συνεργασίας τους, για την ανάπτυξη της διαδικτυακής πλατφόρμας MyWiFiSpeed.

Κατά την ανάπτυξη της διαδικτυακής πλατφόρμας MyWiFiSpeed μας δόθηκε η ευκαιρία όχι μόνο να έρθουμε σε επαφή με νέες τεχνολογίες και τεχνικές οι οποίες είναι απαραίτητες για ένα τέτοιο εγχείρημα, αλλά και να εμβαθύνουμε στη χρήση και στον τρόπο λειτουργίας τους. Επίσης αποκτήσαμε μια κάποια εμπειρία για το πώς πρέπει να “στηθεί” το περιβάλλον που απαιτείται για την υλοποίηση μιας διαδικτυακής εφαρμογής. Όμως ο τομέας από τον οποίο αποκομίσαμε τα περισσότερα οφέλη σε επίπεδο γνώσης, θεωρούμε πως είναι τα προβλήματα που αντιμετωπίσαμε κατά την διάρκεια της υλοποίησης της διαδικτυακής εφαρμογής και η προσπάθεια εξεύρεσης λύσης για αυτά. Επιγραμματικά μπορούμε να αναφέρουμε στο σημείο αυτό παρατηρήσεις για μερικά από τα προβλήματα που μας δυσκόλεψαν και μας παραξένεψαν ιδιαίτερα κατά την υλοποίηση της διαδικτυακής εφαρμογής:

- **Αδυναμία υποστήριξης των PrimeFaces 5.0 από τον GlassFish 3.1:** Κατά τη χρήση της έκδοσης 5.0 των PrimeFaces παρατηρήσαμε πως μερικά στοιχεία (elements) της έκδοσης αυτής δεν υποστηρίζονταν από τον GlassFish 3.1. Η διαπίστωση αυτή έλαβε χώρα όταν προσπαθήσαμε να κάνουμε deploy την διαδικτυακή εφαρμογή MyWiFiSpeed, που κάνει χρήση του στοιχείου “p:menubar” της συγκεκριμένης έκδοσης των PrimeFaces, στον GlassFish και λάβαμε μήνυμα λάθους. Η λύση που δώσαμε εμείς στο πρόβλημα αυτό ήταν να αναβαθμίσουμε το GlassFish στην έκδοση 4.0. Άλλες λύσεις που θα μπορούσαν να εφαρμο-

στούν είναι πιθανόν η αποφυγή χρήσης στοιχείων της έκδοσης αυτής των PrimeFaces που δεν υποστηρίζονται από την συγκεκριμένη έκδοση του Glass-Fish ή η χρήση παλαιότερης έκδοσης των PrimeFaces.

- **Αδυναμία εμφάνισης των styles του Geoserver στον Openlayer χάρτη:** Καθώς η έκδοση των OpenLayers άλλαξε πρόσφατα στην έκδοση 3, δυσκολευτήκαμε στην ανάκτηση των layers του GeoServer στους χάρτες της διαδικτυακής εφαρμογής μας. Μετά από μελέτη στα εγχειρίδια χρήσης των OpenLayers παρατηρήσαμε πως ο τρόπος ανάκτησης WMS layers από τον GeoServer έχει αλλάξει από την έκδοση 2 στην έκδοση 3. Η JavaScript συνάρτηση που υλοποιήσαμε για να επιτύχουμε αυτή τη λειτουργία φαίνεται στην Εικόνα 73.

```
1 function ShowCite() {  
2     var wmslayer = new ol.layer.Tile({  
3         source: new ol.source.TileWMS({  
4             url: 'http://83.212.113.47:8888/geoserver/wms',  
5             params: {  
6                 LAYERS: 'cite:huaLocation',  
7                 FORMAT: 'image/png',  
8                 TRANSPARENT: true,  
9                 STYLES: 'Downlink'  
10            }  
11        })  
12    });  
13    _Map.addLayer(wmslayer);  
14 }
```

Εικόνα 73: Συνάρτηση για την ανάκτηση των WMS layer στα OpenLayers 3.

- **Ανάκτηση όλων των συντεταγμένων του εμβαδού της περιοχής που έχει επιλεγεί ως περιοχή καμπάνιας:** Όπως παρουσιάστηκε στα προηγούμενα κεφάλαια ο διαχειριστής μπορεί να ορίσει την περιοχή μιας καμπάνιας επιλέγοντας την στον χάρτη. Κατά την επιλογή της περιοχής αυτής μπορεί να σχεδιάσει ένα πολύγωνο, εμείς όμως αυτό που χρειαζόμαστε είναι το κέντρο της περιοχής αυτής και μια ακτίνα καθώς αυτό που περιμένει η εφαρμογή HuaServiceTester2 είναι ένας κύκλος που θα ορίζει την περιοχή της καμπάνιας. Συνεπώς το πρόβλημα ήταν από ένα πολύγωνο σχεδιασμένο στα OpenLayers να βρούμε το σύνολο των συντεταγμένων του εμβαδού του και από αυτά να βρούμε το κέντρο και την ακτίνα (μέγιστο πλάτος πολύγωνου / 2) του. Το κλειδί στην επίλυση του προβλήματος αυτού είναι η παρακάτω τρεις γραμμές JavaScript κώδικα (αφού πρώτα έχουμε σχεδιάσει και έχουμε πάρει σε έναν πίνακα τις συντεταγμένες της κάθε γωνιάς του πολυγώνου):

```
//_Draw.f[0] πίνακας συντεταγμένων όλων των γωνιών του πολυγώνου
```

```
var boundingExtentObj = ol.extent.boundingExtent(_Draw.f[0]);
```

```
var center = ol.extent.getCenter(boundingExtentObj);
```

```
var width = ol.extent.getWidth(boundingExtentObj);
```

Αυτά ήταν μερικά μόνο από τα προβλήματα που ίσως από μια απλή ανάγνωση μπορεί να φαίνονται απλά, αλλά μας πήρε αρκετό χρόνο και διάβασμα σε documentations για την επίλυση τους κατά την υλοποίηση της διαδικτυακής εφαρμογής μας.

Περαιτέρω όμως πρέπει να αναφερθούμε και στο γεγονός πως με το πέρας της υλοποίησης της διαδικτυακής εφαρμογής, στα πλαίσια της παρούσας διπλωματικής εργασίας, έμειναν ζητήματα που δεν λύθηκαν (κυρίως λόγω χρόνου και ότι σε μέγεθος θα βγαίναμε εκτός ορίων διπλωματικής εργασίας) και χρήζουν μελλοντικής υλοποίησης ή επέκτασης. Περιγραφικά αναφέρουμε μερικές πιθανές μελλοντικές υλοποιήσεις και επεκτάσεις που μπορούν να εφαρμοστούν στα όσα έχουμε υλοποιήσει μέχρι τώρα.

- **Εντοπισμός της γεωγραφικής θέσης του χρήστη:** Ο αυτόματος ή μη εντοπισμός της γεωγραφικής θέσης του χρήστη που επισκέπτεται την διαδικτυακή μας εφαρμογή δεν υλοποιήθηκε. Παρόλο που η υλοποίηση μιας τέτοιας λειτουργίας είναι εύκολη, εμείς δεν την υλοποιήσαμε. Καθώς ακολουθήσαμε το σκεπτικό πως αφού υπάρχει διαθέσιμη εφαρμογή για κινητές συσκευές οι χρήστες που θα επισκέπτονται την διαδικτυακή εφαρμογή θα χρησιμοποιούν κατά κύριο λόγο υπολογιστές που συνήθως δεν προσφέρουν πληροφορίες για την γεωγραφική θέση του χρήστη ή αν τις προσφέρουν είναι με μεγάλο σφάλμα. Από την άλλη όμως η τεράστια χρήση των κινητών συσκευών και η συνεχώς αυξανόμενες δυνατότητες τους, καθιστούν επιτακτική ανάγκη την υλοποίηση μιας τέτοιας λειτουργίας. Επιπλέον η ύπαρξη λειτουργίας εντοπισμού θέσης του χρήστη καθιστά την εφαρμογή πολύ πιο φιλική στην χρήση της.
- **Παραμετροποίηση υπάρχουσας καμπάνιας:** αυτή τη στιγμή όσο αφορά τις υπάρχουσες καμπάνιες η διαδικτυακή εφαρμογή, από πλευράς παραμετροποίησης, προσφέρει μόνο τη δυνατότητα απενεργοποίησης και ενεργοποίησης τους. Πιστεύουμε όμως πως είναι μια πολύ καλή επέκταση η υλοποίηση των λειτουργιών που να προσφέρουν τη δυνατότητα περεταίρω παραμετροποίησης για τις υπάρχουσες καμπάνιες. Παραδείγματα τέτοιων παραμετροποιήσεων μπορεί να είναι η δυνατότητα:
 - ο Επέκτασης του χρόνου λήξης της υπάρχουσας καμπάνιας.

- Αλλαγής των στοιχείων της όπως το όνομα της, το δίκτυο το οποίο στοχεύει, τη περιοχή της κτλ.
- Αλλαγή των στοιχείων μιας καμπάνιας, που έχει αποθηκευτεί προσωρινά, πριν γίνει αποστολή της στις κινητές συσκευές.
- **Επιπλέον μετασχηματισμοί:** εφαρμογή και άλλων μετασχηματισμών, εκτός από τον Barnes Surface, στα δεδομένα μέτρησης της απόδοσης των ασύρματων δικτύων. Αυτό θα βοηθούσε ιδιαίτερα στην απόκτηση μιας σφαιρικής άποψης για την απόδοση των ασύρματων δικτύων.
- **Επιπλέον στατιστικά στοιχεία:** οι μετρήσεις που πραγματοποιούνται από τις κινητές συσκευές μπορούν και προσφέρουν πολύ πληροφορία από την οποία μπορούμε να εξάγουμε και να δημιουργήσουμε στατιστικά γραφήματα πραγματικού χρόνου ή μη. Συνεπώς υπάρχει πολύ μεγάλη δυνατότητα για την προσθήκη περισσότερων στατιστικών στοιχείων.

Η λίστα των προβλημάτων που αντιμετωπίσαμε, των εμπειριών που αποκτήσαμε και των όσων μπορούν μελλοντικά να υλοποιηθούν στην εφαρμογή που αναπτύξαμε, είναι πολύ μεγάλη για να την παραθέσουμε ολόκληρη στην παρούσα εργασία. Υπάρχουν όμως καταγεγραμμένα και παρουσιάστηκαν όσο το δυνατόν αναλυτικότερα και επεξηγηματικά στα προηγούμενα κεφάλαια. Παρόλα αυτά μπορείτε να παραμείνετε ενήμεροι για την πορεία της εφαρμογής, αν επισκέπτεστε στην διεύθυνση που λειτουργεί και διαβάζοντας τυχόν μελλοντικές εργασίες που θα πραγματοποιηθούν με βάση τα όσα καταγράψαμε εδώ.

Παραπομπές

- [1] J Howe. The rise of crowdsourcing. In Wired magazine, 2006
- [2] Nicholas D. Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T. Campbell. A Survey of Mobile Phone Sensing. In Communications Magazine, IEEE.
- [3] Daren C. Brabham. Crowdsourcing as a Model for Problem Solving. In University of Utah, USA
- [4] S. Consolvo et al., “Activity Sensing in the Wild: A Field Trial of Ubifit Garden,” Proc. 26th Annual ACM SIGCHI Conf. Human Factors Comp. Sys., 2008, pp. 1797–1806.
- [5] E. Miluzzo et al., “Sensing meets Mobile Social Networks: The Design, Implementation, and Evaluation of the CenceMe Application,” Proc. 6th ACM SenSys, 2008, pp. 337–50.
- [6] M. Mun et al., “Peir, the Personal Environmental Impact Report, as a Platform for Participatory Sensing Systems Research,” Proc. 7th ACM MobiSys, 2009, pp. 55–68.
- [7] A. Thiagarajan et al., “VTrack: Accurate, Energy-Aware Traffic Delay Estimation Using Mobile Phones,” Proc. 7th ACM SenSys, Berkeley, CA, Nov. 2009.
- [8] berkeley.edu, UC Berkeley/Nokia/NAVTEQ, “Mobile Millennium”. [Online]
<http://traffic.berkeley.edu/>
- [9] sensorly.com, Sensorly. [Online]
<http://www.sensorly.com/>
- [10] P. Mohan, V. N. Padmanabhan, and R. Ramjee. Nericell: rich monitoring of road and traffic conditions
- [11] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda. PIER, the personal environmental impact report, as a platform for participatory sensing systems research. In Proceedings of ACM MobiSys, pages 55–68, 2009
- [12] R. Rana, C. Chou, S. Kanhere, N. Bulusu, and W. Hu. Earphone: An end-to-end participatory urban noise mapping. In Proceedings of ACM/IEEE IPSN, pages 105–116, 2010
- [13] Jochen Schiller and Agnès Voisard, Location-Based Services
- [14] Pradeep K. Murukannaiah and Munindar P. Singh: Understanding Location- Based User Experience, North Carolina State University
- [15] Pratap Misra and Per Enge: Global Positioning System: Signals, Measurements and Performance Second Edition

- [16] Olmo Zavala-Romero, Arsalan Ahmed, Eric P. Chassignet, Jorge Zavala-Hidalgo, Agustin Fernandez Eguiarte, Anke Meyer-Baese: An open source Java web application to build self-contained web GIS sites
- [17] Kropla, B., 2005. Beginning MapServer: Open Source GIS Development. Apress
- [18] Deoliveira, J., 2008. GeoServer: uniting the GeoWeb and spatial data infrastructures. In: Proceedings of the 10th International Conference for Spatial Data Infrastructure, St. Augustine, Trinidad. [Online]
- [19] Shekhar, S., Xiong, H., 2007. Quantum GIS. Springer, New York ; Berlin.
- [20] Neteler, M., Bowman, M.H., Landa, M., Metz, M., 2012. GRASS GIS: a multi-purpose open source GIS. Environ. Model. Softw. 31, 124e130. [Online]
<http://linkinghub.elsevier.com/retrieve/pii/S1364815211002775>
- [21] mapbender3.org, Mapbender3. [Online]
<http://mapbender3.org/>
- [22] geomajas.org, Geomajas. [Online]
<http://www.geomajas.org/>
- [23] mapfish.org, MapFish. [Online]
<http://www.mapfish.org/>
- [24] geomoose.org, Geomoose. [Online]
<http://www.geomoose.org/>
- [25] android.com, gcm. [Online]
<https://developers.google.com/cloud-messaging/gcm>
- [26] wikipedia.org, HTML. [Online]
<https://en.wikipedia.org/wiki/HTML>
- [27] w3schools, html. [Online]
<http://www.w3schools.com/html/>
- [28] w3.org, xml. [Online]
<http://www.w3.org/TR/REC-xml/>
- [29] w3schools, xml. [online]
http://www.w3schools.com/xml/xml_usedfor.asp
- [30] wikipedia.org, Cascading Style Sheets. [Online]
https://en.wikipedia.org/wiki/Cascading_Style_Sheets
- [31] w3schools.com, css. [Online]

- http://www.w3schools.com/css/css_howto.asp
- [32] wikipedia.org, Ajax. [Online]
http://en.wikipedia.org/wiki/Ajax_%28programming%29#cite_note-garrett-1
- [33] w3schools.com, ajax intro. [Online]
http://www.w3schools.com/ajax/ajax_intro.asp
- [34] adaptivepath.com, ajax new approach web applications. [Online]
<http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications>
- [35] Gosling James, Joy Bill, Steele Guy, Bracha Gilad and Buckley Alex: The Java® Language Specification
- [36] computerweekly.com, Write once run anywhere. [Online]
<http://www.computerweekly.com/feature/Write-once-run-anywhere>
- [37] Ed Burns, Chris Schalk and Neil Griffin: JavaServer Faces 2.0 The complete reference
- [38] oracle.com, oracle web.xml. [Online]
http://docs.oracle.com/cd/E13222_01/wls/docs81/webapp/web_xml.html
- [39] wikipedia.org, MVC. [Online]
<http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>
- [40] oracle.com, jar file. [Online]
<https://docs.oracle.com/javase/tutorial/deployment/jar/basicsindex.html>
- [41] primefaces.org, PrimeFaces user guide. [Online]
<http://www.primefaces.org/docs>
- [42] google.com, Google Maps API Documentation. [Online]
<https://developers.google.com/maps/documentation/javascript/>
- [43] openlayers.org, OpenLayers API. [Online]
<http://openlayers.org/en/v3.5.0/apidoc/>
- [44] microsoft.com, BingMap API. [Online]
<http://www.microsoft.com/maps/choose-your-bing-maps-API.aspx>
- [45] JDL, Data Fusion Lexicon. Technical Panel For C3, F.E. White, San Diego, Calif, USA, Code 420, 1991
- [46] D. L. Hall and J. Llinas, “An introduction to multisensor data fusion,” Proceedings of the IEEE, vol. 85, no. 1, pp. 6–23, 1997.
- [47] Federico Castanedo: A Review of Data Fusion Techniques
- [48] geoserver.org, GeoServer User Manual, Release 2.4.8. [Online]

- <http://docs.geoserver.org/stable/en/user/>
- [49] opengeospatial.org, Open Geospatial Consortium. [Online]
<http://www.opengeospatial.org/>
- [50] opengeospatial.org, Web Feature Service. [Online]
<http://www.opengeospatial.org/standards/wfs>
- [51] opengeospatial.org, Web Coverage Service. [Online]
<http://www.opengeospatial.org/standards/wcs>
- [52] opengeospatial.org, Web Map Service. [Online]
<http://www.opengeospatial.org/standards/wms>
- [53] theopenplanningproject.org, The Open Planning Project (TOPP). [Online]
<http://www.theopenplanningproject.org/>
- [54] opengeospatial.org, Styled Layer Descriptor. [Online]
<http://www.opengeospatial.org/standards/sld>
- [55] opengeo.org, HeatMap. [Online]
<http://suite.opengeo.org/4.1/cartography/rt/heatmap.html>
- [56] opengeo.org, Point Stacker. [Online]
<http://suite.opengeo.org/4.1/cartography/rt/pointstacker.html>
- [57] opengeo.org, Barnes Surface. [Online]
<http://suite.opengeo.org/4.1/cartography/rt/barnes.html>
- [58] opensignal.com, Open Signal. [Online]
<http://opensignal.com/>
- [59] netbeans.org, NetBeans Official site. [Online]
<https://netbeans.org/>
- [60] glassfish.java.net, Glass Fish Documentation. [Online]
<https://glassfish.java.net/documentation.html>
- [61] glassfish.java.net, GlassFish Installation Guide. [Online]
<https://glassfish.java.net/docs/4.0/installation-guide.pdf>
- [62] oracle.com, WAR File. [Online]
<http://www.oracle.com/technetwork/articles/javase/servletapi-137835.html>

- [63] phpmyadmin.net, phpMyAdmin Windows Installation. [Online]
<http://docs.phpmyadmin.net/en/latest/setup.html>
- [64] geoserver.org, GeoServer Installation
<http://docs.geoserver.org/stable/en/user/installation/>
- [65] Νικόλαος Κορωνιώτης, Πτυχιακή εργασία: Ανάπτυξη εφαρμογής και δικτυακής πλατφόρμας μέτρησης απόδοσης ασύρματων δικτύων για συσκευές Android
- [66] Μωυσιάδης Αναστάσιος, Διπλωματική εργασία: Crowdsourcing Application for Personalized Location Based QoE of Wireless Networks via Background Service on Smart Devices / Εφαρμογή συλλογής γεωχωρικών δεδομένων απόδοσης ασύρματων δικτύων με χρήση υπηρεσίας παρασκηνίου για έξυπνες συσκευές.
- [67] google.com, XMPP Implementation. [Online]
<https://developers.google.com/cloud-messaging/ccs>
- [68] github.com, GitHub Google GCM Server. [Online]
<https://github.com/google/gcm>
- [69] google.com, Google Console. [Online]
<https://cloud.google.com/console>
- [70] google.com, Google Console Help. [Online]
<https://developers.google.com/console/help/new/>
- [71] netbeans.org, Create Entity Classes. [Online]
<https://netbeans.org/kb/docs/javaee/ecommerce/entity-session.html>
- [72] oracle.org, JPA. [Online]
<http://docs.oracle.com/javaee/6/tutorial/doc/bnbpz.html>
- [73] wikipedia.org, CRUD. [Online]
http://en.wikipedia.org/wiki/Create,_read,_update_and_delete