



Χαροκόπειο Πανεπιστήμιο

Τμήμα Πληροφορικής και Τηλεματικής

Πτυχιακή Εργασία

Ανάπτυξη εφαρμογής και δικτυακής πλατφόρμας
(sensing platform) μέτρησης απόδοσης ασύρματων
δικτύων για συσκευές iOS

Κουντουράς Αθανάσιος

ΑΜ: 21018

Επιβλέπων Καθηγήτρια

Νικολαΐδου Μάρα

Μέλη της εξεταστικής επιτροπής

Δαλάκας Βασίλειος

Δημητρακόπουλος Γεώργιος

Ευχαριστίες

Η παρούσα πτυχιακή εργασία υλοποιήθηκε στα πλαίσια του προγράμματος προπτυχιακών σπουδών του τμήματος Πληροφορικής και Τηλεματικής στο Χαροκόπειο Πανεπιστήμιο Αθηνών. Η υλοποίηση της πτυχιακής εργασίας ξεκίνησε τον Οκτώβριο του 2013 και ολοκληρώθηκε τον Ιούνιο του 2014.

Στα πλαίσια της εργασίας αυτής ήρθα σε επαφή με ένα μεγάλο εύρος από τεχνολογίες και εφάρμοσα πολλές από τις δεξιότητες και γνώσεις που απέκτησα κατά την διάρκεια των σπουδών μου.

Επίσης συνεργάστηκα στενά με το προσωπικό του Τμήματος Πληροφορικής και θα ήθελα να ευχαριστήσω όλα τα άτομα που μου παρείχαν την απαραίτητη υποστήριξη για να γίνει εφικτή η υλοποίηση αυτής της πτυχιακής εργασίας.

Πιο συγκεκριμένα θα ήθελα να ευχαριστήσω την Καθηγήτρια Μάρα Νικολαΐδου η οποία με εμπιστεύθηκε για την ανάθεση αυτής της πτυχιακής εργασίας και παρείχε την απαραίτητη υποστήριξη που απαιτούταν για να ολοκληρωθεί αυτή η πτυχιακή.

Ιδιαίτερες ευχαριστίες θα ήθελα να προσφέρω στον Δρ. Βασίλειο Δαλάκα ο οποίος μου προσέφερε απλόχερα τις τεχνικές και ακαδημαϊκές του γνώσεις σχετικά με το αντικείμενο της πτυχιακής και με τον οποίο συνεργάστηκα στενά καθ όλη την περίοδο υλοποίησης αυτής της πτυχιακής.

Ακόμη θα ήθελα να ευχαριστήσω τον συμφοιτητή μου Νίκο Κορωνιώτη με τον οποίο είχαμε μια εξαιρετικά παραγωγική συνεργασία.

Η προσπάθεια υλοποίησης αυτής της πτυχιακής εργασίας θα ήταν επίσης αδύνατη χωρίς την ύπαρξη της πλατφόρμας okeanos. Έτσι θα ήθελα να ευχαριστήσω όλα τα άτομα που δούλεψαν πάνω στην υλοποίηση και υποστήριξη μιας τόσο απαραίτητης υπηρεσίας.

Τέλος η εργασία αυτή ολοκληρώθηκε με την αμέριστη ηθική συμπαράσταση της οικογένειάς μου η οποία με στήριξε σε κάθε δυσκολία που συνάντησα στην πορεία των σπουδών μου.

Πρόλογος

Σε αυτό το κείμενο θα παρουσιαστεί η διαδικασία σχεδιασμού και ανάπτυξης της πτυχιακής εργασίας με τίτλο: Ανάπτυξη εφαρμογής και δικτυακής πλατφόρμας (sensing platform) μέτρησης απόδοσης ασύρματων δικτύων για συσκευές iOS από τον Αθανάσιο Κουντουρά.

- Στο πρώτο κεφάλαιο θα αναφερθεί ο λόγος ύπαρξης αυτής της πτυχιακής και ποιος είναι ο σκοπός της.
- Στο δεύτερο κεφάλαιο θα δοθούν πληροφορίες σχετικές με την κατεύθυνση επίλυσης του προβλήματος καθώς και πληροφορίες που αφορούν τις τεχνολογίες που χρησιμοποιήθηκαν σε αυτή την εργασία μαζί με πληροφορίες σχετικές με την ανάπτυξη εφαρμογών για το λειτουργικό σύστημα iOS.
- Το τρίτο κεφάλαιο περιγράφει την διαδικασία της εγκατάστασης και παραμετροποίησης της διαδικτυακής πλατφόρμας και των επιμέρους τμημάτων της.
- Στο τέταρτο κεφάλαιο περιγράφεται η διαδικασία ανάπτυξης της εφαρμογής για το λειτουργικό σύστημα iOS και γίνεται μία περιγραφή της λειτουργίας της.
- Τελειώνοντας στο πέμπτο κεφάλαιο παρουσιάζονται τα συμπεράσματα στα οποία οδηγήθηκα κατά την διάρκεια της εκπόνησης αυτής της πτυχιακής εργασίας καθώς και πιθανές επεκτάσεις για το σύστημα διαδικτυακή πλατφόρμα- εφαρμογή.

Η πτυχιακή εργασία αυτή ήταν μία πολύ έντονη διαδικασία μάθησης μέσω της οποίας απέκτησα περαιτέρω γνώσεις σχετικά με τον προγραμματισμό για κινητές συσκευές, ιδιαίτερα για συσκευές iOS. Επίσης ενημερώθηκα περισσότερο για την χρήση και παραμετροποίηση εφαρμογών που χρησιμοποιούνται ως εξυπηρετές για διάφορες υπηρεσίες του διαδικτύου. Ακόμη απέκτησα μεγαλύτερη εμπειρία με τον διαδικτυακό προγραμματισμό και τις προκλήσεις του ιδιαίτερα για δίκτυα κινητής τηλεφωνίας.

Το τμήμα της διαδικτυακής πλατφόρμας υλοποιήθηκε σε συνεργασία με τον συμφοιτητή μου Νίκο Κορωνιώτη.

Περιεχόμενα

Ευχαριστίες.....	3
Πρόλογος.....	4
Περίληψη.....	7
Abstract.....	8
Λέξεις Κλειδιά.....	9
1. Εισαγωγή.....	10
1.1 Ορισμός του προβλήματος.....	10
1.2 Σημασία της μέτρησης απόδοσης και των αποτελεσμάτων της για τους χρήστες.....	11
1.3 Σκοπός της εργασίας.....	12
2. Σχεδιασμός.....	14
2.1 Εξήγηση της κατεύθυνσης επίλυσης του προβλήματος.....	14
2.2 Αρχιτεκτονική του συστήματος.....	17
2.3 Εξήγηση των τεχνολογιών που χρησιμοποιήθηκαν στην εφαρμογή για κινητά τηλέφωνα.....	18
2.3.1 Το λειτουργικό σύστημα iOS.....	18
2.3.2 Η γλώσσα προγραμματισμού Objective-C.....	20
2.3.3 Το αρχιτεκτονικό μοτίβο ανάπτυξης λογισμικού (software architectural pattern) Model-View-Controller (MVC).....	22
2.3.4 Το περιβάλλον ανάπτυξης λογισμικού (IDE) Xcode (5.1.1).....	25
2.3.5 Εξήγηση της επιλογής για ανάπτυξη της εφαρμογής στο λειτουργικό σύστημα iOS.....	26
2.4 Εξήγηση των τεχνολογιών που χρησιμοποιήθηκαν στην ηλεκτρονική πλατφόρμα.....	30
2.5 Άλλες υλοποιήσεις και κατευθύνσεις.....	33
3. Υλοποίηση διαδικτυακής πλατφόρμας (Sensing Platform.....)	41
3.1 Πληροφορίες σχετικά με τον Server.....	41
3.2 Εξήγηση των τεχνολογιών που χρησιμοποιούνται στον Server.....	41

3.3 Εγκατάσταση και παραμετροποίηση του τμήματος Server της πλατφόρμας.....	43
3.4 Παραμετροποίηση της Βάσης δεδομένων.....	49
3.4.1 Δημιουργία χρήστη της Βάσης δεδομένων.....	49
3.4.2 Δημιουργία πίνακα σε MySQL βάση για αποθήκευση μετρήσεων και γεωχωρικών δεδομένων.....	50
3.5 Εξήγηση του τρόπου παραλαβής και αποθήκευσης των δεδομένων.....	52
3.6 Παραμετροποίηση του Geoserver.....	53
3.7 Παρουσίαση και εξήγηση της γλώσσας SLD.....	60
3.8 Απεικόνιση των αποτελεσμάτων στο διαδίκτυο.....	62
4. Υλοποίηση εφαρμογής για συσκευές iOS.....	63
4.1 Προετοιμασία του περιβάλλοντος ανάπτυξης.....	63
4.2 Περιγραφή της υλοποίησης της εφαρμογής.....	74
4.3 παρουσίαση της διεπαφής χρήστη της εφαρμογής.....	92
5. Συμπεράσματα και επεκτάσεις.....	96
Βιβλιογραφία.....	99

Περίληψη

Καθώς ο αριθμός των χρηστών έξυπνων κινητών συσκευών συνεχώς αυξάνεται και η πρόσβαση σε υπηρεσίες διαδικτύου γίνεται όλο και πιο διαδεδομένη γίνεται εμφανής η ανάγκη σύγκρισης της ποιότητας των παρεχόμενων υπηρεσιών πρόσβασης στο διαδίκτυο. Παρόλα αυτά δεν υπάρχουν πολλές λύσεις που να δίνουν στους χρήστες πρόσβαση σε πληροφορίες σχετικές με την αναμενόμενη ποιότητα υπηρεσιών που θα συναντήσουν σε διάφορες γεωγραφικές περιοχές. Ο λόγος για τον οποίο τέτοιου είδους πληροφορία είναι απαραίτητη έχει να κάνει με την μεγάλη ποικιλομορφία τεχνολογιών δικτύωσης για κινητές συσκευές καθώς και με τους περιορισμούς που παρέχουν αυτές οι τεχνολογίες.

Αντικείμενο λοιπόν αυτής της εργασίας ήταν η ανάπτυξη μιας διαδικτυακής πλατφόρμας συνοδευόμενη από μια εφαρμογή για κινητά τηλέφωνα με λειτουργικό σύστημα iOS, με σκοπό την παροχή μιας υπηρεσίας που θα επέτρεπε στους χρήστες κινητών τηλεφώνων να μετρήσουν κάποια χαρακτηριστικά του δικτύου στο οποίο βρίσκονται, καθώς και να δουν τα αποτελέσματα των μετρήσεων άλλων χρηστών αποτυπωμένα πάνω σε ένα χάρτη.

Ειδικότερα, κύριος στόχος της εργασίας ήταν η δυνατότητα συλλογής πληροφοριών σχετικών με την ποιότητα της διαδικτυακής σύνδεσης έξυπνων κινητών τηλεφωνικών συσκευών καθώς και γεωγραφικών πληροφοριών που θα επιτρέψουν την δημιουργία μιας απεικόνισης των μετρήσεων όλων των χρηστών της πλατφόρμας. Με αυτό τον τρόπο οι χρήστες θα έχουν την δυνατότητα να αντιληφθούν την ποιότητα των υπηρεσιών διαδικτυακής πρόσβασης που είχαν άλλοι χρήστες της υπηρεσίας σε διάφορες γεωγραφικές τοποθεσίες με έναν εύκολο και κατανοητό τρόπο.

Επίσης αυτή η πλατφόρμα θα μπορεί να χρησιμοποιηθεί και σαν εργαλείο για την σύγκριση διάφορων τεχνολογιών πρόσβασης καθώς και την ποιότητα των δικτύων διαφορετικών εταιριών παροχής κινητών υπηρεσιών.

Με αυτούς τους στόχους υλοποιήθηκε μια διαδικτυακή πλατφόρμα που συλλέγει και επεξεργάζεται αποτελέσματα μετρήσεων από κινητές συσκευές μαζί με γεωγραφικές πληροφορίες. Επίσης αναπτύχθηκε η εφαρμογή για κινητά με λειτουργικό σύστημα iOS η οποία διεξάγει μετρήσεις και αποστέλλει τα αποτελέσματα των μετρήσεων καθώς και γεωγραφικές πληροφορίες καθώς και άλλου τύπου πληροφορίες σχετικά με το κινητό δίκτυο. Ακόμη η εφαρμογή προσφέρει πρόσβαση στα αποτελέσματα μετρήσεων άλλων χρηστών πάνω σε ένα χάρτη για εύκολη κατανόηση αυτών των αποτελεσμάτων. Τέλος αναπτύχθηκε και μια απλή διαδικτυακή σελίδα που δίνει πρόσβαση στα ίδια αποτελέσματα.

Abstract

As the number of users using a modern smartphone continually increases, and access to Internet services becomes more affordable, there is an apparent need for comparison between the quality of Internet access services. In spite of this fact, there aren't many solutions that provide the users with information about the quality of Internet service related with the geographical area. The reason that this kind of information is necessary has to do with the vast variety of cellular networking technologies and their limitations.

The objective of this thesis was the development of a sensing platform and a corresponding application for iOS based mobile phones, with the goal of providing a service that would allow mobile phones' users to measure some characteristics about the network they are connected to, as well as viewing the results of the measurements of other users plotted on a map.

More specifically, the main goal of the thesis was to obtain the ability to collect information about the quality of the Internet connection of modern smartphones as well as their geographical information that allowed us to create a visual representation of the results from all the users with a web platform. This visualization offers the users the ability to perceive the quality of Internet services that other users experienced on specific geographical locations without expert knowledge.

Another possible use of this platform could be as a tool for the comparison of different networking technologies and/or the services that different mobile Internet service providers offer.

Having all these in mind, we developed a web-based sensing platform that collects and processes network measurement results from mobile devices coupled with geographical information. Also, we developed an application for iOS-based mobile phones, that conducts network measurements, gathers geographical information and other data about the mobile network, and reports the results back to the sensing platform. Additionally the mobile application provides access to the results of measurements conducted by the other users of the platform on the user's device. The results are plotted on a map for a clear interpretation of the results. Finally, we also developed a simple web page that displays the same information as the mobile application.

Λέξεις Κλειδιά

iOS εφαρμογή, απόδοση ασύρματου δικτύου, αισθητήρια πλατφόρμα, crowdsourcing, υπηρεσίες με βάση την θέση, ping

Keywords

iOS agent, wireless network throughput, sensing platform, crowdsourcing, location based services, ping

1. Εισαγωγή

1.1 Ορισμός του προβλήματος.

Ζούμε σε μια εποχή που όλο και περισσότεροι άνθρωποι ανακαλύπτουν την χρηστικότητα που τους προσφέρει το διαδίκτυο. Παράλληλα οι έξυπνες κινητές συσκευές (smartphones) γίνονται όλο και πιο διαδεδομένες και ένα από τα κυριότερα χαρακτηριστικά που προσφέρουν αυτές οι συσκευές είναι η δυνατότητα πρόσβασης στο διαδίκτυο. Έτσι βλέπουμε πως όλο και μεγαλύτερη είναι η ανάγκη ύπαρξης γρήγορων και αξιόπιστων συνδέσεων για μεταφορά δεδομένων με την χρήση κινητών συσκευών.

Έτσι στις μέρες μας υπάρχουν πολλές ασύρματες τεχνολογίες που παρέχουν διαδικτυακή συνδεσιμότητα σε κινητές συσκευές. Οι τεχνολογίες αυτές χωρίζονται σε δυο μεγάλες κατηγορίες. Η πρώτη κατηγορία είναι οι κινητές τεχνολογίες πρόσβασης σε δεδομένα βασισμένες στο σύστημα GSM (Global System for Mobile Communications). Αυτές συμπεριλαμβάνουν τις τεχνολογίες που τώρα κατατάσσουμε ως τεχνολογίες δεύτερης γενιάς (2G), τρίτης γενιάς (3G) και τέταρτης γενιάς (4G). Η δεύτερη μεγάλη κατηγορία τεχνολογιών είναι οι τεχνολογίες που βασίζονται στο πρότυπο IEEE 802.11 και που πολλές φορές αναφέρονται και ως Wi-Fi. Σε αυτή την κατηγορία περιέχονται συνήθως εξέλιξης του προτύπου IEEE 802.11 όπως είναι το IEEE 802.11g και IEEE 802.11n τα οποία είναι ευρέως διαδεδομένα.

Η ύπαρξη και χρήση τόσο πολλών και διαφορετικών τεχνολογιών σύνδεσης έχει σαν αποτέλεσμα την πολύ διαφορετική ποιότητα διαδικτυακής πρόσβασης για τις κινητές συσκευές που διαφέρει τόσο σε επίπεδο ταχύτητας αλλά και σε επίπεδο αξιοπιστίας. Επίσης η ποιότητα της σύνδεσης δεν εξαρτάται μόνο από την τεχνολογία σύνδεσης. Σε κινητά δίκτυα τύπου GSM η ποιότητα της σύνδεσης εξαρτάται και από παράγοντες όπως είναι η ένταση του σήματος κινητών δεδομένων, οι ικανότητες του δικτύου του παρόχου, ακόμη και τα εξαρτήματα και η κατασκευή της κινητής συσκευής. Από την άλλη τα δίκτυα τύπου IEEE 802.11 επηρεάζονται και αυτά από παράγοντες όπως η ένταση σήματος αλλά ακόμη περισσότερο επηρεάζονται από την ταχύτητα και την ποιότητα της επίγειας σύνδεσης προς τον σταθμό βάσης που δημιουργεί το ασύρματο δίκτυο.

Παράλληλα όπως αναφέρεται στα Johansson, J. et al, Minimization of Drive Tests in 3GPP Release 11 (1) και Hapsari, W.A et al, Minimization of Drive Tests Solution in 3GPP (2) η ανάγκη για μετρήσεις απόδοσης (performance) των δικτύων κινητών τηλεφώνων με χρήση συσκευών χρηστών είναι μεγάλη και διατυπώνεται και από τον οργανισμό 3GPP (3rd Generation Partnership Project), μια συνεργασία εταιριών που απασχολούνται στον τομέα των τηλεπικοινωνιών για την ανάπτυξη και την εξέλιξη των δυνατοτήτων των δικτύων κινητής τηλεφωνίας που βασίζονται στο GSM. Οι λόγοι για την ανάγκη ύπαρξης μετρήσεων χρηστών είναι αρκετοί όπως αναφέρεται στα (1) και (2) αλλά κυριότεροι είναι η μείωση του λειτουργικού κόστους των εταιριών και η καλύτερη και γρηγορότερη αποτύπωση της εικόνας των δικτύων των εταιριών παροχής υπηρεσιών κινητής τηλεφωνίας και δεδομένων με σκοπό την πιο ορθή δράση σε δραστηριότητες βελτίωσης των δικτύων αυτών.

Είναι οπότε φανερή η ανάγκη ύπαρξης εργαλείων που θα μπορούν να μετρήσουν διάφορα μετρικά δικτύων συγκεκριμένα προσαρμοσμένα στις ανάγκες συνδεσιμότητας των σύγχρονων κινητών συσκευών και να μπορούν να τα παρουσιάσουν με τρόπο που θα αποτυπώνει και θα τα συσχετίζει με τις μεταβολές που παρουσιάζουν ανάλογα με την γεωγραφική τους θέση.

1.2 Σημασία της μέτρησης απόδοσης και των αποτελεσμάτων της για τους χρήστες.

Επειδή υπάρχουν πολλές διαφορετικές τεχνολογίες που η απόδοσή τους εξαρτάται από πάρα πολλούς παράγοντες, είναι εμφανές πως οι χρήστες δεν μπορούν να γνωρίζουν άμεσα και με εύκολο τρόπο τι ποιότητα υπηρεσιών πρόσβασης δικτύων μπορούν να αναμένουν καθώς κινούνται μέσα στην χώρα.

Επίσης, υπάρχουν και πολλές διαφορετικές εταιρίες παροχής υπηρεσιών πρόσβασης στο διαδίκτυο μέσω κινητών τηλεφώνων. Οι χρήστες λοιπόν καλούνται να αποφασίσουν σε θέματα παρόχων η ακόμη και συσκευών χωρίς να έχουν μια γρήγορη και αντικειμενική ενημέρωση σχετικά με την ποιότητα της υπηρεσίας που μπορούν να περιμένουν από κάθε πάροχο υπηρεσιών σε περιοχές που τους ενδιαφέρουν.

Οι υπάρχουσες λύσεις που εκτελούν δοκιμές διαφόρων χαρακτηριστικών ασύρματων δικτύων συνήθως δεν παρέχουν πρόσβαση σε αποτελέσματα άλλων χρηστών και συνήθως δεν εμφανίζουν τα δεδομένα με ένα απλό και εύκολο τρόπο ώστε να μπορεί να γίνει αντιληπτός από τον μέσο χρήστη. Πολλές από αυτές τις λύσεις που παρέχουν τέτοια χαρακτηριστικά τα παρέχουν με ένα περίπλοκο τρόπο που δεν απλοποιεί τα αποτελέσματα με επίπτωση να δημιουργούν περισσότερα ερωτήματα στους χρήστες.

Το πρόβλημα επομένως είναι η έλλειψη μιας πλατφόρμας που θα συλλέγει απλά δεδομένα απόδοσης κινητών και ασύρματων δικτύων και θα τα παρουσιάζει με εύκολο τρόπο συσχετισμένα με γεωγραφικές πληροφορίες, ώστε να μπορούν οι χρήστες να κάνουν σωστές και ενημερωμένες επιλογές.

1.3 Σκοπός της εργασίας

Η εργασία αυτή επομένως, έχει σαν κύριο σκοπό να δημιουργήσει μια μία εφαρμογή για έξυπνα κινητά τηλέφωνα, η οποία θα μπορεί να εκτελεί απλά μετρικά δικτύου και να τα παρουσιάζει στον χρήστη μαζί με αποτελέσματα άλλων χρηστών της εφαρμογής πάνω σε ένα χάρτη που θα εμφανίζει την τοποθεσία κάθε μέτρησης. Όλες οι λειτουργίες της θα υλοποιούνται με ένα εύκολο για τον χρήστη τρόπο και τα αποτελέσματα θα εμφανίζονται απλά και κατανοητά.

Επίσης μέρος της εργασίας είναι η ανάπτυξη μιας διαδικτυακής πλατφόρμας που θα συλλέγει και θα επεξεργάζεται τα αποτελέσματα των μετρήσεων και άλλες σχετικές πληροφορίες από την εφαρμογή. Ακόμη, θα παρέχει τα αποτελέσματα των μετρήσεων στην εφαρμογή με σκοπό την προβολή τους πάνω σε ένα χάρτη καθώς και μια ιστοσελίδα την οποία θα μπορούν να επισκεφθούν χρήστες για να δούνε αποτελέσματα μετρήσεων από τον υπολογιστή τους. Ακόμη θα έχει την δυνατότητα να συνεργάζεται και με άλλες εφαρμογές που θα τρέχουν σε κινητά τηλέφωνα με διαφορετικά λειτουργικά συστήματα.

Η εφαρμογή στοχεύει σαν χρήστη τον μέσο χρήστη έξυπνων κινητών συσκευών και όχι τον τεχνικό δικτύων η εταιρία διεξαγωγής μελετών δικτύων. Ο κυριότερος λόγος για αυτή την κατεύθυνση είναι το γεγονός πως υπάρχουν ήδη λύσεις για ένα πιο τεχνικό κοινό και

δεν θα μπορούσαμε να ανταγωνιστούμε εργαλεία που έχουν αναπτυχθεί από μεγάλες εταιρίες με έτη εμπειρίας σε μετρικά δικτύων.

2. Σχεδιασμός

2.1 Εξήγηση της κατεύθυνσης επίλυσης του προβλήματος.

Η εργασία θα περιέχει 2 ανεξάρτητες οντότητες. Η μία θα είναι η εφαρμογή η οποία έχει αναπτυχθεί για να μπορεί να εκτελεστεί σε κινητές συσκευές που τρέχουν το λειτουργικό σύστημα iOS 7. Η άλλη θα είναι μια διαδικτυακή πλατφόρμα που θα είναι συνέχεια σε λειτουργία και θα εξυπηρετεί την εφαρμογή παρέχοντας τα αποτελέσματα μετρήσεων άλλων χρηστών, ένα web-service που θα παραλαμβάνει τα αποτελέσματα των μετρήσεων από την εφαρμογή και θα τα καταχωρεί στην βάση δεδομένων και επίσης θα περιέχει τα αρχεία που θα μεταφέρονται από την εφαρμογή για την μέτρηση της ταχύτητας του δικτύου.

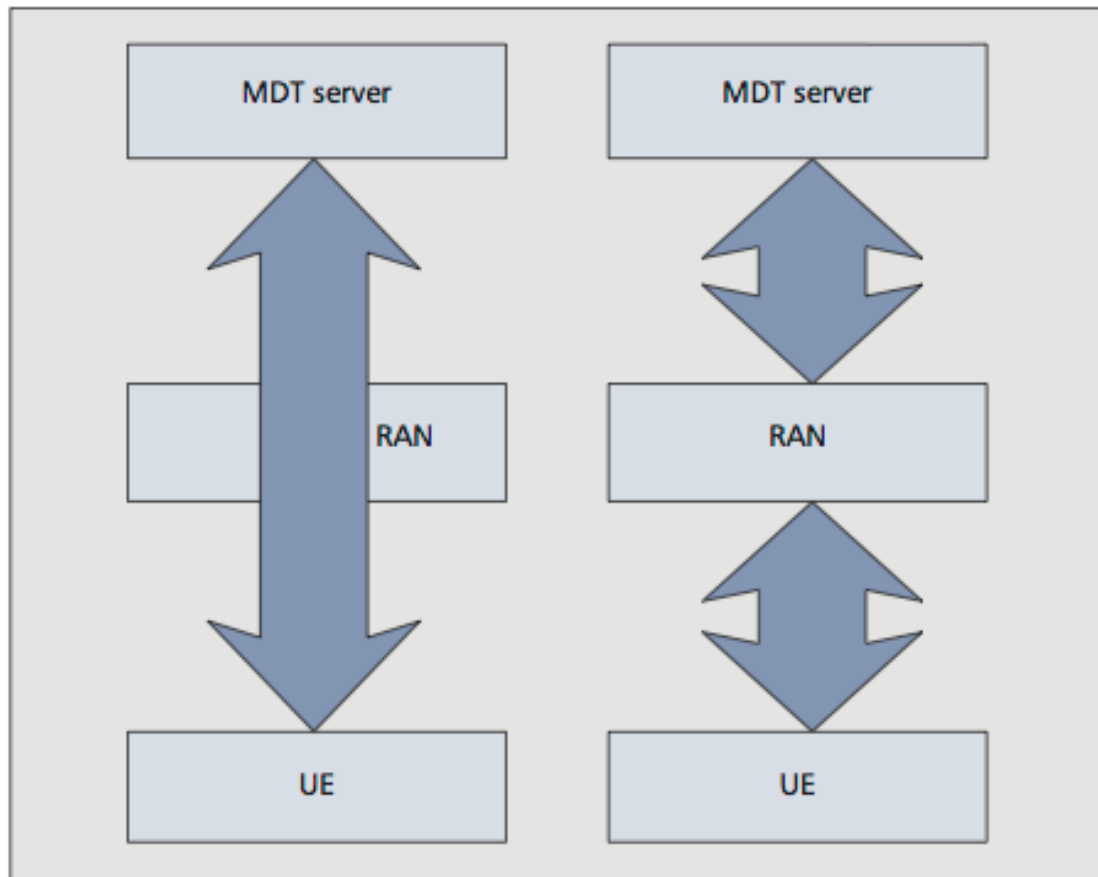
Αυτή η κατεύθυνση ακολουθήθηκε γιατί επιτρέπει την χρήση της πλατφόρμας και από άλλες εφαρμογές που μπορούν να έχουν υλοποιηθεί σε άλλα λειτουργικά συστήματα για κινητά τηλέφωνα όπως το Android. Επίσης με κάποιες αλλαγές και παραμετροποίηση η ίδια πλατφόρμα θα μπορούσε να εξυπηρετήσει και εφαρμογές που εκτελούν άλλου είδους μετρήσεις που θέλουμε να αποτυπωθούν μαζί με γεωγραφικές πληροφορίες. Δηλαδή χρησιμοποιώντας την ίδια εγκατάσταση λογισμικού στον Server και με κάποιες παραμετροποιήσεις ώστε να προσαρμοστεί στους διαφορετικούς τύπους δεδομένων που μπορεί να απαιτεί μια άλλου είδους κατηγορία μετρήσεων και εάν η εφαρμογή υλοποιεί τον ίδιο τρόπο αποστολής και λήψης δεδομένων θα μπορούσαμε να εξυπηρετήσουμε και μια εντελώς διαφορετική κατηγορία εφαρμογών μετρήσεων που θα χρησιμοποιεί το πλήθος των αισθητήρων που βρίσκονται στα σύγχρονα έξυπνα κινητά, όπως θα μπορούσε να είναι για παράδειγμα μια εφαρμογή μέτρησης περιβαλλοντικού θορύβου ανά τις ώρες της ημέρας βασισμένη στην γεωγραφική θέση της συσκευής.

Δόθηκε λοιπόν μεγάλη έμφαση στην δυνατότητα της πλατφόρμας να είναι ικανή να εξυπηρετήσει οποιαδήποτε είδους εφαρμογή κάνει μετρήσεις με γεωγραφικά δεδομένα. Γι' αυτό και η υλοποίηση της πλατφόρμας αλλά και ο τρόπος μεταφοράς δεδομένων από και προς την εφαρμογή βασίστηκαν πάνω σε υπάρχοντα πρότυπα που χρησιμοποιούνται για πλήθος εφαρμογών σε όλο τον κόσμο.

Η εφαρμογή παράλληλα θα είναι προφανώς πλήρως συμβατή με τις τεχνολογίες που χρησιμοποιούνται για την υλοποίηση της διαδικτυακής πλατφόρμας αλλά θα τηρεί και τους όρους και τις οδηγίες σχετικά με την ορθή ανάπτυξη εφαρμογών για το λειτουργικό σύστημα iOS. Έτσι όλα τα δεδομένα που λαμβάνονται από την εφαρμογή θα παρέχονται από επιτρεπόμενες κλήσεις μεθόδων του λειτουργικού συστήματος και για τις μετρήσεις θα χρησιμοποιούνται τα εργαλεία μεταφοράς αρχείων που παρέχει ο κατασκευαστής του λειτουργικού συστήματος iOS, δηλαδή η εταιρία Apple.

Παράλληλα η εργασία θα προσφέρει μετρήσεις οι οποίες θα παρέχουν αρκετά από τα δεδομένα που μπορούν να αποκτηθούν μέσω της διαδικασίας Drive Testing κινητών δικτύων. Το Drive Testing είναι μία διαδικασία που ακολουθείται από παρόχους κινητής τηλεφωνίας με σκοπό την μέτρηση χαρακτηριστικών των δικτύων τους ανά περιοχή. Συνήθως διεξάγεται με ειδικό εξοπλισμό ειδικά σχεδιασμένο για καταμέτρηση δεδομένων με μεγάλη ακρίβεια και περιέχουν και τρόπο καταγραφής της τοποθεσίας της μέτρησης μέσω της τεχνολογίας GPS. Μερικές από τις μετρήσεις που λαμβάνονται σε Drive Test συμπεριλαμβάνουν: ένταση σήματος (Signal intensity), ποιότητα σήματος (Signal quality), παρεμβολές (Interference), Service level statistics και πολλά ακόμη δεδομένα ειδικευμένα για μέτρηση της αποτύπωσης της λειτουργικότητας ενός κινητού δικτύου σε κάποια περιοχή. Όμως τα τελευταία χρόνια και ειδικότερα μετά από την έλευση των έξυπνων κινητών συσκευών (smartphones) εμφανίζεται η τάση για ελαχιστοποίηση των Drive Test που διεξάγονται από τους παρόχους κυρίως λόγω του κόστους διεξαγωγής ενός Drive Test. Η τάση αυτή για μείωση των Drive Tests αποτυπώνεται και στην προτυποποίηση από τον οργανισμό 3GPP τεχνολογιών Minimization of Drive Tests (MDT) που βασίζονται σε μετρήσεις από τις συσκευές των χρηστών για την εξαγωγή μέρους των συμπερασμάτων σχετικά με το δίκτυο που μπορούν να εξαχθούν από την διεξαγωγή ενός Drive Test.

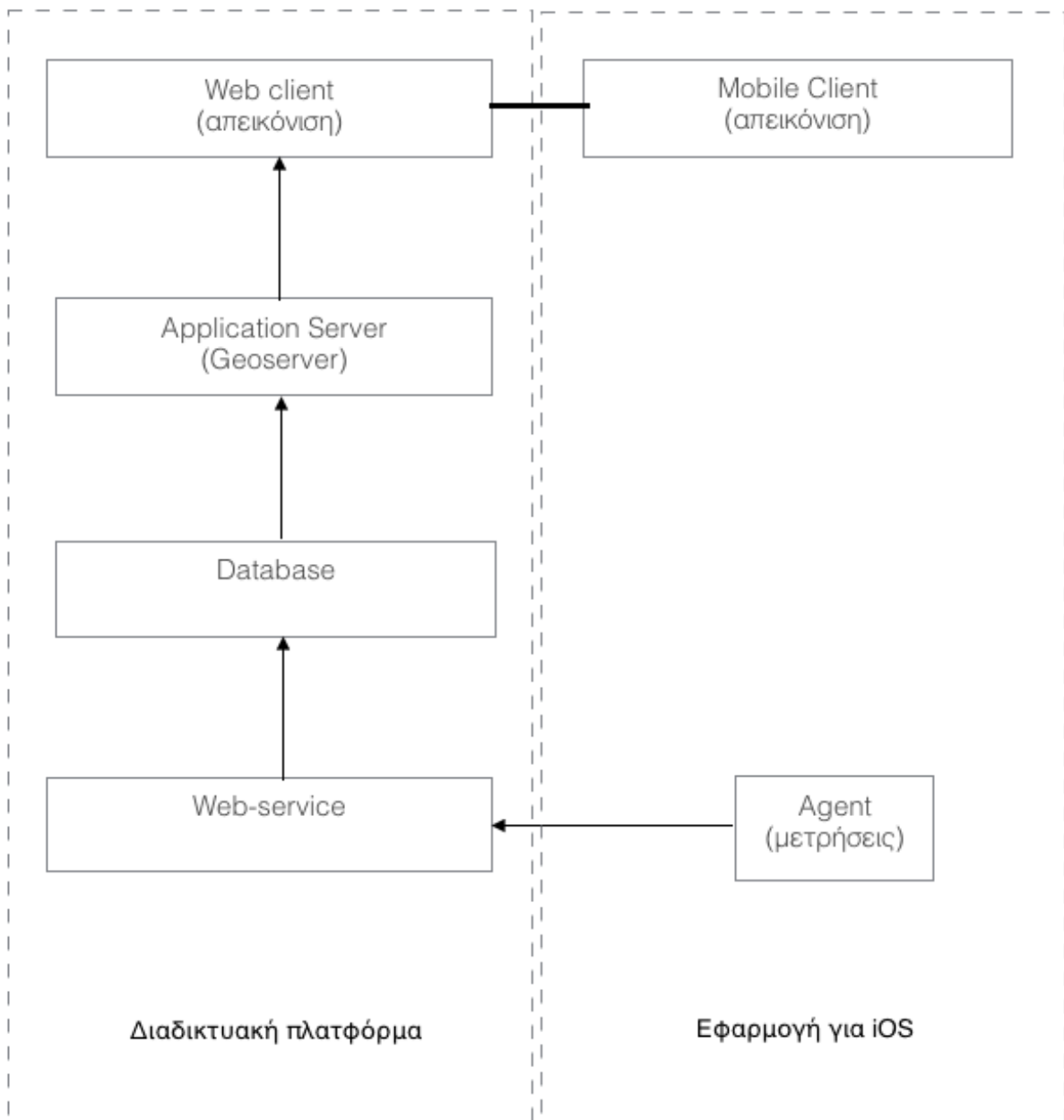
Η εργασία μας μετράει δεδομένα που ανήκουν στην κατηγορία Network Benchmarking των Drive Test ακολουθεί την αρχιτεκτονική μέτρησης τύπου U-Plane MDT η οποία περιγράφεται και συγκρίνεται με άλλες αρχιτεκτονικές μέτρησης απόδοσης (performance) δικτύων. Το παρακάτω σχήμα προσφέρει μία οπτική σύγκριση ανάμεσα στην αρχιτεκτονική U-plane MDT και C-plane MDT.



Εικόνα 1.1 - U-plane αρχιτεκτονική (αριστερά) και C-plane αρχιτεκτονική (δεξιά) (Πηγή (2))

Το MDT (minimization of Drive Tests) αναφέρεται σε τεχνολογίες μέτρησης απόδοσης κινητών δικτύων με την χρήση των κινητών συσκευών των χρηστών (User Equipment, UE) όπως αναφέρεται στα (1) και (2). Η αρχιτεκτονική U-plane περιγράφει μία διάφανη (transparent) για τον πάροχο και το κινητό δίκτυο (Radio Access Network, RAN) διαδικασία μέτρησης απόδοσης. Στη U-plane αρχιτεκτονική η μέτρηση εκτελείται ανάμεσα στον MDT Server και στην συσκευή του χρήστη (UE) με μία άμεση σύνδεση μεταξύ τους, χωρίς την επίγνωση πως εκτελείται δοκιμή του δικτύου από τον πάροχο του δικτύου (RAN). Τα αποτελέσματα μετά την μέτρηση αποστέλλονται στον MDT Server.

2.2 Αρχιτεκτονική του συστήματος



Εικόνα 2.1 - Γραφική απεικόνιση της αρχιτεκτονικής του συστήματος

Στην εικόνα 2.1 απεικονίζεται γραφικά η αρχιτεκτονική των κύριων τμημάτων του συστήματος Διαδικτυακής πλατφόρμας και εφαρμογής. Τα μεγάλα τετράγωνα με τις διακεκομμένες γραμμές υποδηλώνουν το σύστημα που εμπεριέχει τα τμήματα που απεικονίζονται στο εσωτερικό του. Δηλαδή η διαδικτυακή πλατφόρμα περιέχει:

- Το web-service: Είναι υπεύθυνο για την παραλαβή των δεδομένων των μετρήσεων από τον Agent και την καταχώρηση στην βάση δεδομένων (Database)
- Τη Database: Περιέχει τα δεδομένα των μετρήσεων και τα παρέχει στον Application Server

- Τον Application Server: Επεξεργάζεται τα δεδομένα των μετρήσεων που βρίσκονται στην Βάση δεδομένων και κατασκευάζει τις απεικονίσεις που θα παρουσιαστούν στον χρήστη από τους client.
- Τον web client: Προσφέρει ένα απλό περιβάλλον στον χρήστη μέσω web που αποτυπώνει τα αποτελέσματα των μετρήσεων.
- Ενώ η εφαρμογή για iOS περιέχει:
- Τον measurement agent: Συλλέγει πληροφορίες και εκτελεί τις μετρήσεις δικτύου.
- Τον mobile client: Παρουσιάζει με ένα απλό περιβάλλον τα αποτελέσματα στον χρήστη.

Αξίζει επίσης να σημειωθεί πως ο mobile client και ο web client αποτυπώνουν τα ίδια δεδομένα σε διαφορετικές πλατφόρμες, αυτός είναι και ο λόγος που εμφανίζονται συνδεδεμένα στην εικόνα 2.1.

2.3 Εξήγηση των τεχνολογιών που χρησιμοποιήθηκαν στην εφαρμογή για κινητά τηλέφωνα

Για να μπορέσει να γίνουν περισσότερο κατανοητές διάφορες αποφάσεις και διαδικασίες που ακολουθηθήκαν κατά την διάρκεια της υλοποίησης της εφαρμογής καθώς και για λόγους γενικού ενδιαφέροντος θα αναλυθούν οι διάφορες τεχνολογίες που είναι χρήσιμο να γίνουν κατανοητές από οποιονδήποτε ενδιαφέρεται να ασχοληθεί με τον προγραμματισμό για το λειτουργικό σύστημα iOS. Επίσης θα παρουσιαστούν και τα κυριότερα εργαλεία προγραμματισμού που χρησιμοποιούνται για τον προγραμματισμό εφαρμογών για iOS.

2.3.1 Το λειτουργικό σύστημα iOS

Το λειτουργικό σύστημα iOS, αρχικά με το όνομα iPhone OS, ανακοινώθηκε από τον τότε διευθύνων σύμβουλο και ιδρυτή της αμερικανικής εταιρίας Apple Inc., Steve Jobs, τον Ιανουάριου του 2007 στη διάσκεψη του περιοδικού Macworld σαν μέρος της παρουσίασης του καινούργιου προϊόντος της εταιρίας, το iPhone. (3). Στην παρουσίαση ο Steve Jobs ανέφερε πως “το iPhone τρέχει OS X”, και αναφερόταν στο γεγονός πως το λειτουργικό

iPhone OS ήταν βασισμένο στο λειτουργικό σύστημα που χρησιμοποιούσε η εταιρία για τους υπολογιστές της το οποίο ονομάζεται Mac OS X. (4) Παρόλα αυτά έχει γίνει γνωστό πως το λειτουργικό σύστημα iPhone OS που τον Ιούνιο του 2010 μετονομάστηκε σε iOS παρόλο που βασίζεται στο λειτουργικό σύστημα Mac OS X δεν είναι συμβατό με τις εφαρμογές πράγμα που σημαίνει πως υπάρχουσες εφαρμογές έπρεπε να τροποποιηθούν δραματικά για να υποστηρίζονται από το iOS. Αξίζει επίσης να σημειωθεί πως για τον πρώτο χρόνο ύπαρξης και λειτουργίας του λειτουργικού συστήματος iOS δεν υπήρχε η δυνατότητα εγκατάστασης εφαρμογών στη συσκευή και ο μονός τρόπος για να εκτέλεση εφαρμογών ήταν μέσω web με πολλούς περιορισμούς.

Όμως σχεδόν ένα χρόνο μετά από την παρουσίαση ανακοινώθηκε το iPhone Software Development Kit (SDK) που έδινε την δυνατότητα σε προγραμματιστές να αναπτύξουν εφαρμογές που θα εκτελούνται στο κινητό τηλέφωνο. (5)

Όπως αναφέρθηκε το λειτουργικό iOS βασίζεται στο λειτουργικό σύστημα Mac OS X, το οποίο είναι ένα Unix λειτουργικό σύστημα (6). Παρόλα αυτά το λειτουργικό iOS παρουσιάζει αρκετούς περιορισμούς και δεν επιτρέπει την πρόσβαση χρήστη σε λειτουργίες φλοιού (shell access) οπότε δεν μπορεί να θεωρηθεί και αυτό Unix λειτουργικό σύστημα. Παράλληλα το iOS μοιράζεται κάποια από τα προγραμματιστικό πλαίσια (frameworks) που χρησιμοποιούνται στον προγραμματισμό για συστήματα Mac OS X, όπως είναι τα πλαίσια (Frameworks) Foundation και Core Foundation. Όμως το πλαίσιο γραφικών (UI Framework) του iOS είναι το Cocoa Touch που διαφέρει από το πλαίσιο γραφικών το Cocoa που είναι το πλαίσιο γραφικών (UI Framework) του Mac OS X, κάνοντας εφαρμογές γραμμένες για εκτέλεση στο λειτουργικό Mac OS X μη συμβατές με το iOS. Ο λόγος για αυτή την διαφορά είναι ο πολύ διαφορετικός τρόπος αλληλεπίδρασης εφαρμογών με τον χρήστη σε κινητά τηλέφωνα με οθόνη αφής, όπως είναι το iPhone, σε σχέση με τον τρόπο αλληλεπίδρασης σε προσωπικούς υπολογιστές.

Οι τεχνολογίες που χρησιμοποιούνται στο λειτουργικό iOS μπορούν να αναπαρασταθούν και σαν επίπεδα (layers) με τον iOS πυρήνα (kernel) στην βάση και την τεχνολογία cocoa touch στην κορυφή. Πιο συγκεκριμένα το λειτουργικό iOS μπορεί να χωριστεί στα εξής.

1. Cocoa Touch: Περιέχει τις λειτουργίες που διαχειρίζονται την αλληλεπίδραση μέσω αφής, την εμφάνιση των εφαρμογών, μετρήσεις από αισθητήρες όπως οι αισθητήρες κίνησης, καθώς και εφαρμογές web.
2. Media / Application Services: Λειτουργίες για διαχείριση διεπαφής ήχου, προβολή και επεξεργασία εικόνων, κίνηση και γραφήματα καθώς και διαχείριση κάποιων τύπων αρχείων, όπως για παράδειγμα αρχεία τύπου pdf.
3. Core Services: Υπηρεσίες δικτύου, νηματική διαχείριση (Threading), διαχείριση δεδομένων και επιλογών, παροχή υπηρεσιών γεωγραφικής θέσης και διαχείριση αρχείων.
4. Core OS / iOS kernel: Ο πυρήνας του λειτουργικού, διαχείριση ενέργειας, υπηρεσίες ασφάλειας και πιστοποιητικά, το σύστημα αρχείων και POSIX sockets. Ο προγραμματιστής έχει ελάχιστη αλληλεπίδραση με αυτό το επίπεδο. Οι περισσότερες λειτουργίες αυτού του επιπέδου δεν προσφέρουν καμία πρόσβαση στον προγραμματιστή ενώ ακόμη και η λειτουργία των sockets είναι εξαιρετικά περιορισμένη και δεν συνίσταται η χρήση της.

2.3.2 Η γλώσσα προγραμματισμού Objective-C

Η γλώσσα προγραμματισμού Objective-C, είναι μία γλώσσα προγραμματισμού γενικής χρήσης η οποία είναι επέκταση της γλώσσας προγραμματισμού C που προσθέτει την δυνατότητα αντικειμενοστρεφή προγραμματισμού. Επειδή δεν είναι μία από τις γλώσσες προγραμματισμού που διδάσκονται στο πλαίσιο των σπουδών του τμήματος πιστεύω πως αξίζει να αναφερθούν κάποιες πληροφορίες και τα χαρακτηριστικά της γλώσσας Objective-C η οποία είναι η κύρια γλώσσα προγραμματισμού που χρησιμοποιείται στην ανάπτυξη εφαρμογών για το λειτουργικό σύστημα iOS.

Η Objective-C δημιουργήθηκε στις αρχές του 1980 στην εταιρία Stepstone και βασίστηκε στην C και τον τρόπο ανταλλαγής μηνυμάτων που ορίστηκε αρχικά στην γλώσσα προγραμματισμού Smalltalk. Αργότερα παραχωρήθηκε με άδεια χρήσης στην εταιρία NeXT, του συνιδρυτή της Apple Steve Jobs. Εκεί χρησιμοποιήθηκε για την ανάπτυξη προγραμματιστικών βιβλιοθηκών και προγραμματιστικών εργαλείων για την υποστήριξη του λειτουργικού συστήματος της NeXT το NeXTstep που μετά την αγορά της NeXT από

την Apple έγινε η βάση του λειτουργικού συστήματος Mac OS X και αργότερα του iOS.
(7)

Το συντακτικό της Objective-C είναι ένα αυστηρό υπερσύνολο (strict superset) της γλώσσας προγραμματισμού C. Είναι δυνατό να μεταγλωττιστεί (compile) κώδικας γραμμένος σε C από οποιονδήποτε μεταγλωττιστή (compiler) για Objective-C. Επίσης κώδικας C μπορεί να συμπεριληφθεί ελεύθερα σε κάθε πρόγραμμα Objective-C. Το αντικειμενοστρεφές συντακτικό της Objective-C έχει επηρεαστεί, όπως έχει αναφερθεί, από την γλώσσα προγραμματισμού Smalltalk που βασίζεται στο πέρασμα μηνυμάτων σε απεικονίσεις αντικειμένων (object instances). Δηλαδή στην Objective-C κάποιος δεν καλεί μια μέθοδο αλλά στέλνει ένα μήνυμα. Αυτό επιτρέπει μία υλοποίηση γραμμένη σε Objective-C να χειριστεί ένα μήνυμα που δεν έχει προηγουμένως υλοποιηθεί προγραμματιστικά όπως κριθεί απαραίτητο κατά την εκτέλεση του προγράμματος (at runtime) σε αντίθεση με γλώσσες προγραμματισμού όπως η C++ η οποία απαιτεί την υλοποίηση συγκεκριμένης μεθόδου για κάθε κλίση. Αυτό δίνει την δυνατότητα δημιουργίας υλοποιήσεων στις οποίες μηνύματα στέλνονται σε μία συλλογή αντικειμένων και λαμβάνεται απάντηση μόνο από αυτά που αναμένεται να απαντήσουν χωρίς τον φόβο λαθών από τα αντικείμενα που δεν αναγνωρίζουν το μήνυμα.

Όσο αφορά την διαχείριση μνήμης, η έκδοση της Objective-C που χρησιμοποιείται για προγραμματισμό εφαρμογών για iOS υποστηρίζει Αυτόματη Προσμέτρηση Αναφορών (Automatic Reference Counting, ARC) που απαλείφει την ανάγκη για διαχείριση της μνήμης από τον προγραμματιστή. Σε αντίθεση με άλλες λύσεις όπως η συλλογή απορριμάτων (garbage collection) που χρησιμοποιούνται από άλλες γλώσσες προγραμματισμού όπως η Java με την Αυτόματη Προσμέτρηση Αναφορών δεν υπάρχει η ανάγκη για την εκτέλεση μία ξεχωριστής διεργασίας που συνεχώς σαρώνει την μνήμη κατά την διάρκεια της εκτέλεσης (runtime). Αυτό επιτρέπει τα προγράμματα σε Objective-C να κάνουν καλή διαχείριση της μνήμης χωρίς να χάνουν πόρους σε ξεχωριστό σύστημα διαχείρισης, έτσι η Objective-C είναι αρκετά ανταγωνιστική με άλλες γλώσσες όσο αφορά τον προγραμματισμό για συστήματα με περιορισμένους πόρους όπως είναι και τα κινητά τηλέφωνα.

Τελειώνοντας την πολύ σύντομη αναφορά στα χαρακτηριστικά της Objective-C πρέπει να αναφερθεί πως μέσα στην περίοδο εκπόνησης αυτής της πτυχιακής εργασίας η εταιρία Apple ανακοίνωσε μία νέα γλώσσα προγραμματισμού για iOS και Mac OS X που ονομάζεται Swift και είναι βασισμένη στην Objective-C αλλά με την προσθήκη πιο μοντέρνων χαρακτηριστικών, όπως η πρόβλεψη τύπου δεδομένων, και την αφαίρεση της προγραμματιστικής δομής της C. Η Objective-C θα συνεχίσει να υποστηρίζεται για την ανάπτυξη εφαρμογών σε iOS και Mac OS X και θα είναι δυνατή η κλίση μεθόδων της Objective-C από προγράμματα γραμμένα σε Swift.

2.3.3 Το αρχιτεκτονικό μοτίβο ανάπτυξης λογισμικού (software architectural pattern) Model-View-Controller (MVC)

Το αρχιτεκτονικό μοτίβο ανάπτυξης λογισμικού Model-View-Controller (MVC) θα μπορούσε να μεταφραστεί και στα ελληνικά ως Μοντέλο-εμφάνιση-χειριστής αλλά εδώ θα εμφανίζεται με την συντομευμένη αγγλική μορφή MVC. Το MVC όπως αναφέρει και η ονομασία του αποτελείται από 3 διαφορετικές οντότητες που συνολικά δημιουργούν το σύνολο μιας εφαρμογής. Στον αντικειμενοστρεφή προγραμματισμό τα

τρία τμήματα αποτελούν και τα τρία σημεία στα οποία ανήκουν τα αντικείμενα. Τα τρία αυτά τμήματα είναι το μοντέλο (model), η εμφάνιση (view) και ο χειριστής (controller). Παρακάτω εξηγείται η λειτουργία του καθενός.

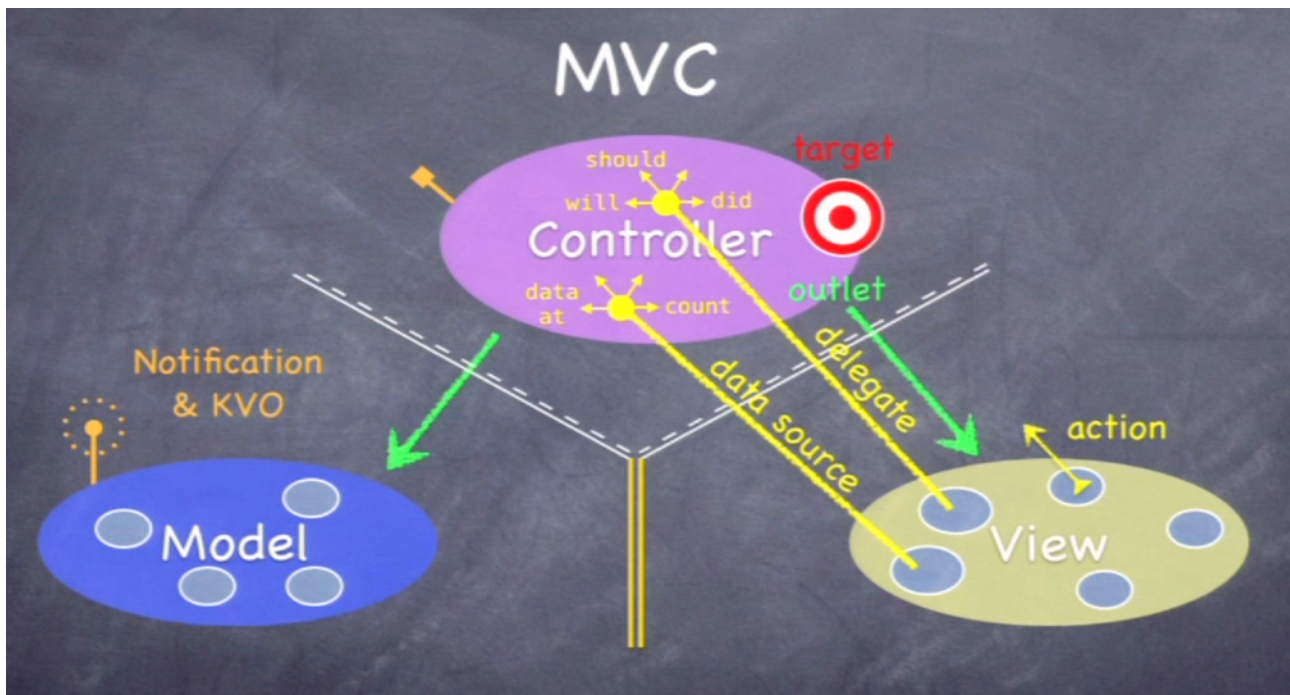
- μοντέλο (model): Περιέχει τις βασικές λειτουργίες της εφαρμογής. Δεν έχει καμία γνώση για τον τρόπο εμφάνισης των δεδομένων στο χρήστη.
- χειριστής (controller): Διαχειρίζεται τον τρόπο με τον οποίο το μοντέλο (model) εμφανίζεται στον χρήστη χρησιμοποιώντας και ελέγχοντας την εμφάνιση (view)
- εμφάνιση (view): Περιέχει απλά γενικά αντικείμενα τα οποία τα χειρίζεται ο χειριστής (controller) για προβολή. Η εμφάνιση (view) δεν έχει καμία επίγνωση της υπόλοιπης υλοποίησης του προγράμματος. Το ιδανικό είναι τα αντικείμενα της εμφάνισης (view) να είναι όσο πιο γενικά γίνεται.

Υπάρχουν συγκεκριμένοι τρόποι με τους οποίους επιτρέπεται να επικοινωνούν μεταξύ τους αυτά τα τρία τμήματα. Πιο συγκεκριμένα, ο χειριστής μπορεί να επικοινωνήσει ανά πάσα στιγμή με το μοντέλο και έχει επίγνωση των ικανοτήτων του μοντέλου. Επίσης ο

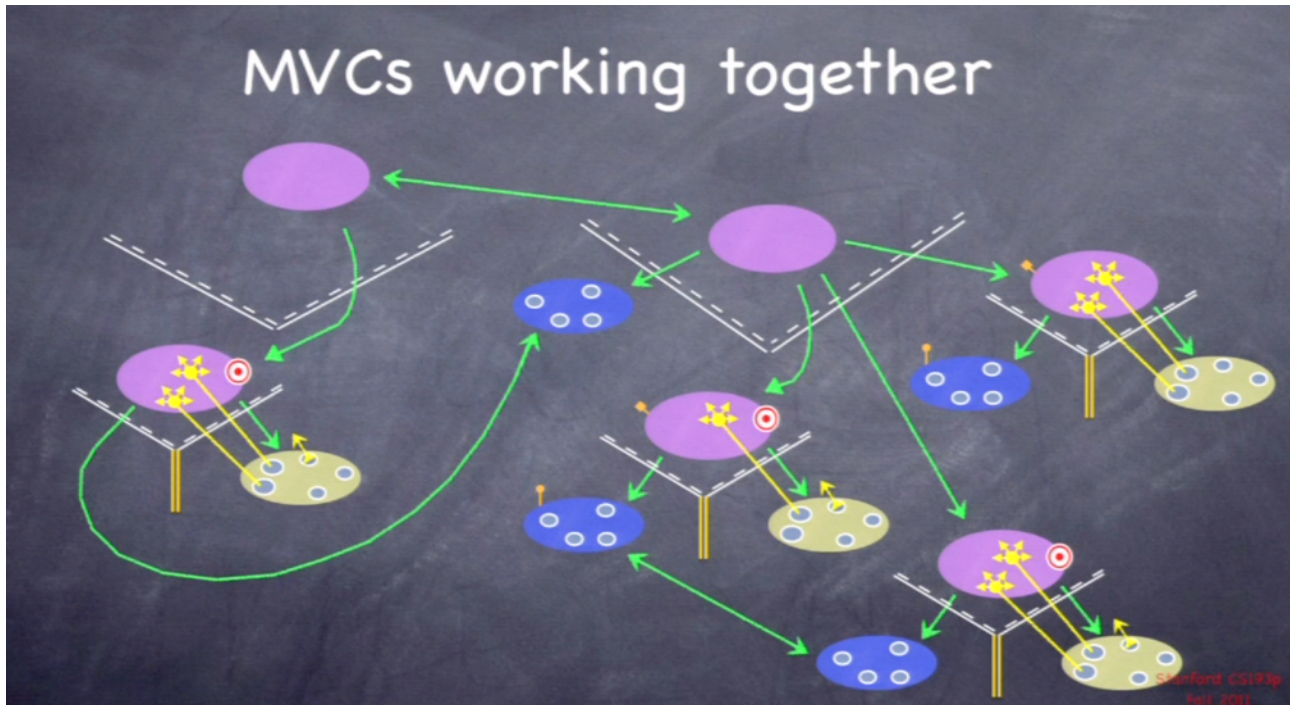
χειριστής έχει την ικανότητα να επικοινωνεί και να αλλάζει την εμφάνιση. Η επικοινωνία αυτή επιτυγχάνεται μέσω διεξόδων (outlets) τα οποία είναι μέρος του χειριστή και τροποποιούν άμεσα την εμφάνιση.

Το μοντέλο και η εμφάνιση δεν έχουν απολύτως καμία άμεση επικοινωνία μεταξύ τους. Το μοντέλο μπορεί να επικοινωνήσει με τον χειριστή αλλά μόνο με πολύ συγκεκριμένους τρόπους. Η επικοινωνία αυτή πρέπει να είναι τυφλή, δηλαδή η εμφάνιση δεν είναι ανάγκη να ξέρει τίποτα σχετικά με τον ελεγκτή και την υλοποίηση του. Ο πρώτος τρόπος επικοινωνίας είναι με την χρήση target-actions. Με αυτό τον τρόπο η εμφάνιση απλά ειδοποιεί τον χειριστή για κάποιο συμβάν, όπως για παράδειγμα την πίεση ενός κουμπιού. Ο άλλος τρόπος είναι με την χρήση delegate. Delegate είναι μία μέθοδος που μπορεί να κληθεί από την εμφάνιση για να ενημερωθεί σε περίπτωση οπου είναι ανάγκη όπως για παράδειγμα όταν η εμφάνιση πρέπει να είναι συγχρονισμένη με κάποιο γεγονός. Ο delegate είναι ο controller οπου απλά υλοποιεί μία προδιατυπωμένη μέθοδο την οποία γνωρίζει πως να καλέσει η εμφάνιση. Αξίζει επίσης να σημειωθεί πως οι εμφανίσεις δεν είναι “ιδιοκτήτες” των δεδομένων που προβάλλουν. Για παράδειγμα, στην περίπτωση που η εμφάνιση προβάλλει ένα πίνακα δεν θα πρέπει ο πίνακας να βρίσκεται αποθηκευμένος σε κάποιο instance της εμφάνισης, η εμφάνιση επιτρέπεται μόνο να υποδεικνύει στα δεδομένα. Το μοντέλο όταν είναι ανάγκη να ειδοποιήσει τον χειριστή μπορεί να στείλει μία ειδοποίηση ώστε να αναλάβει ο χειριστής να διαχειριστεί την αλλαγή.

Για εφαρμογές που απαιτούν μεγάλη πολυπλοκότητα υπάρχει η δυνατότητα να συνδυαστούν πολλά MVC. Ο τρόπος με τον οποίο συνεργάζονται είναι αναλαμβάνοντας ρόλους. Για παράδειγμα σε μια περίπλοκη εφαρμογή μπορεί να υπάρχει μια αρχιτεκτονική MVC η οποία έχει αναλάβει τον ρόλο της εμφάνισης τότε όλο το MVC θα πρέπει να συμπεριφέρεται όπως του επιβάλει ο ρόλος του. Σε κάθε περίπτωση υπάρχει πάντα ένας χειριστής που είναι υπεύθυνος για όλη την εφαρμογή.



Εικόνα 2.2 - Γραφική απεικόνιση μίας αρχιτεκτονικής MVC με επιτρεπτές επικοινωνίες (Πηγή: Paul Hegarty, Stanford University)



Εικόνα 2.3- Γραφική απεικόνιση ενός συστήματος με πολλές MVC (Πηγή: Paul Hegarty, Stanford University)

Ο λόγος για τον οποίο αναφέρεται η αρχιτεκτονική MVC είναι επειδή είναι ο κύριος τρόπος σχεδιασμού εφαρμογών για συσκευές με λειτουργικό σύστημα iOS. Στο προγραμματιστικό περιβάλλον για iOS προσφέρονται όλα τα εργαλεία που απαιτούνται για να πραγματοποιείται η σωστή επικοινωνία ανάμεσα στα διάφορα τμήματα ενός MVC. Εάν ο προγραμματιστής επιχειρήσει να αγνοήσει τους κανόνες της αρχιτεκτονικής σχεδιασμού MVC θα συναντήσει πολλά προβλήματα με τον τρόπο λειτουργίας της εφαρμογής του που συνήθως οδηγούν σε καταστροφική διακοπή λειτουργίας της εφαρμογής. Όλοι οι οδηγοί για προγραμματισμό iOS βασίζονται πάνω στην αρχιτεκτονική MVC και η αρχιτεκτονική αυτή προτείνεται ως η ιδανικότερη από την Apple (8). Άρα και η εφαρμογή της εργασίας υλοποιήθηκε βασισμένη στην αρχιτεκτονική MVC.

2.3.4 Το περιβάλλον ανάπτυξης λογισμικού (IDE) Xcode (5.1.1)

Το Xcode είναι ένα περιβάλλον ανάπτυξης λογισμικού (Integrated Development Environment, IDE) που παρέχεται από την Apple για την ανάπτυξη εφαρμογών που θα εκτελούνται στο λειτουργικό σύστημα iOS ή το Mac OS X. Το Xcode ξεκίνησε να προσφέρεται το 2003 για την υποστήριξη της ανάπτυξης εφαρμογών για Mac OS X. Η πρώτη έκδοση που υποστήριζε το iOS και ήταν μέρος του iOS Software Development Kit (SDK) ήταν η έκδοση 3.1 που ανακοινώθηκε τον Αύγουστο του 2008. Η τελευταία επίσημη έκδοση είναι η έκδοση 5.1 που υποστηρίζει τον προγραμματισμό για συσκευές με iOS 7. Έχει ανακοινωθεί η έκδοση 6 που προσθέτει υποστήριξη για το ερχόμενο iOS 8 και για την νέα γλώσσα προγραμματισμού, Swift.

Το Xcode περιέχει εκτός από το IDE και ένα περιβάλλον σχεδίασης διεπαφής χρήστη (User Interface Builder) σαν μέρος της εφαρμογής. Υποστηρίζει πολλές ευκολίες για τον προγραμματιστή και μπορεί να προβλέψει και να συμπληρώσει αυτόματα ονόματα μεταβλητών και μεθόδων. Επίσης παρέχει γρήγορη πρόσβαση στη βιβλιογραφία κάθε μεθόδου που παρέχεται σαν μέρος του iOS SDK. Έχει ακόμη την δυνατότητα να ανιχνεύσει λάθη και να προτείνει λύσεις χωρίς την ανάγκη compile. Σαν μέρος της δυνατότητας σχεδίασης διεπαφής χρήστη παρέχεται η λειτουργία storyboard η οποία επιτρέπει την ευκολότερη σχεδίαση και σύνδεση των στοιχείων μίας εμφάνισης τύπου MVC με τον χειριστή της.

Ο compiler που προσφέρεται με την έκδοση 6 ονομάζεται LLVM. Ο LLVM είναι γραμμένος με την γλώσσα προγραμματισμού C++ και ο κώδικας του διατίθεται ελεύθερα. Στόχος του LLVM είναι να εκτελεί βελτιστοποιήσεις στο εκτελέσιμο αρχείο που παράγεται κατά την διαδικασία της μεταγλώττισης (compile). Ο LLVM σχεδιάστηκε να μπορεί να λειτουργήσει με πολλές διαφορετικές γλώσσες προγραμματισμού.

2.3.5 Εξήγηση της επιλογής για ανάπτυξη της εφαρμογής στο λειτουργικό σύστημα iOS

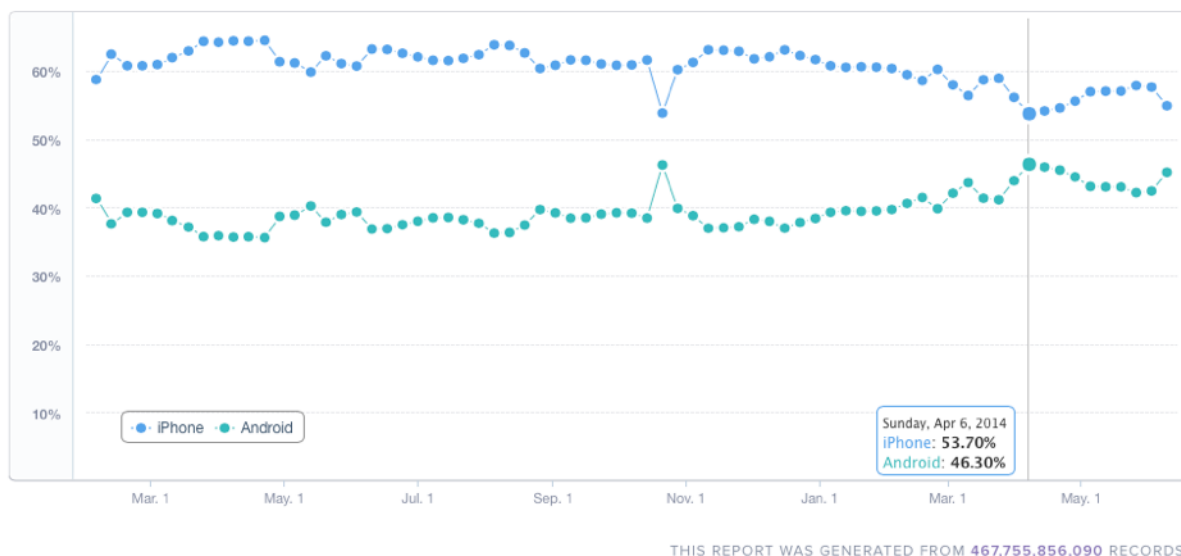
Όπως έχει ήδη αναφερθεί η εφαρμογή για κινητά τηλέφωνα αναπτύχθηκε με σκοπό να μπορεί να εκτελεσθεί σε κινητά τηλέφωνα και κινητές συσκευές που βασίζονται στο λειτουργικό σύστημα iOS 7. Οι λόγοι για τους οποίους επιλέχθηκε να βασιστεί η εφαρμογή στο λειτουργικό σύστημα iOS 7 εξηγούνται παρακάτω.

Το λειτουργικό σύστημα iOS είναι ευρέως διαδεδομένο στην αγορά και έχει ένα αρκετά μεγάλο μερίδιο της αγοράς κινητών τηλεφώνων. Παρόλο που το αντίπαλο λειτουργικό σύστημα Android έχει μεγαλύτερο ποσοστό των ενεργών συσκευών το iOS έχει μεγαλύτερα ποσοστά χρήσης για πρόσβαση στο διαδίκτυο. (9) Αυτό σημαίνει πως παρόλο που υπάρχουν περισσότερες συσκευές που βασίζονται στο λειτουργικό σύστημα Android τα κινητά που βασίζονται στο iOS χρησιμοποιούνται πολύ περισσότερο για πρόσβαση σε εφαρμογές διαδικτύου. Αυτό κάνει την ανάγκη για αποτύπωση της ποιότητας του δικτύου που προσφέρεται σε συσκευές iOS μεγαλύτερη απ' ό,τι σε σχέση με τις συσκευές Android, μιας και οι περισσότεροι χρήστες χρησιμοποιούν συσκευές iOS για να εισέλθουν στο διαδίκτυο.

Το γεγονός πως οι χρήστες iOS χρησιμοποιούν τις συσκευές τους περισσότερο για χρήση εφαρμογών διαδικτύου μπορεί να γίνει ακόμη πιο εμφανής όταν παρατηρήσουμε τα ποσοστά των χρηστών που χρησιμοποίησαν τις κινητές τους συσκευές για να επισκεφθούν ιστοσελίδες καταστημάτων. Σε αυτή την περίπτωση το 60% των επισκεπτών εμπορικών σελίδων από κινητές συσκευές χρησιμοποιούν iOS έναντι 40% των χρηστών Android (10) . Είναι λοιπόν εμφανές πως οι χρήστες που έχουν συσκευές iOS χρησιμοποιούν πολύ περισσότερο τις συσκευές τους για πρόσβαση στο διαδίκτυο.

Android vs. iOS

Feb 5, 2013 - Jun 10, 2014



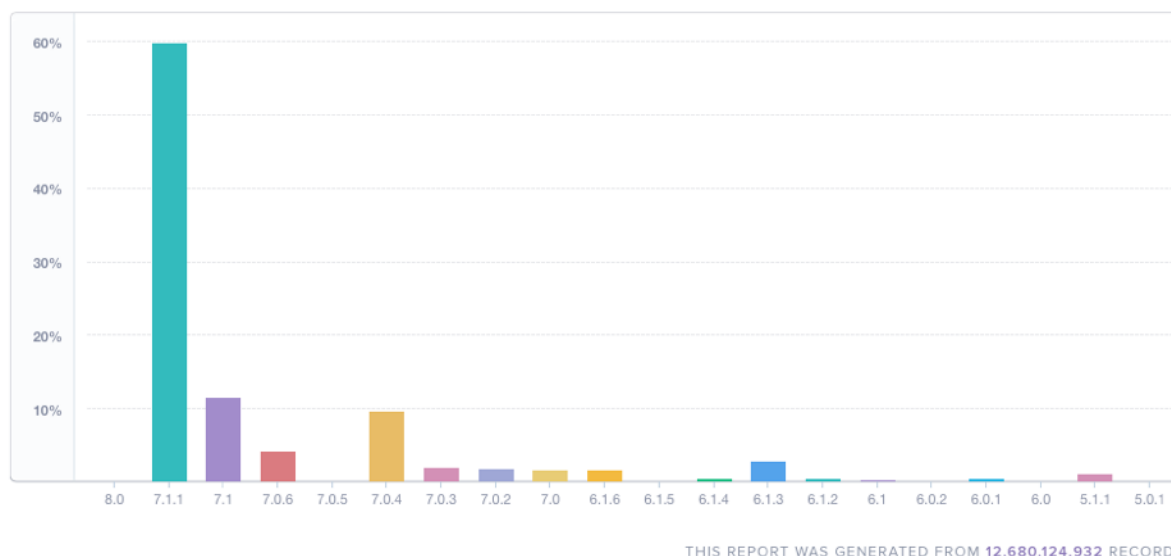
Εικόνα 2.4 - Χρήση κινητών συσκευών (Πηγή: <https://mixpanel.com>)

Άλλο ένα κριτήριο για την επιλογή του συστήματος iOS για την υλοποίηση αυτής της εργασίας είναι και το γεγονός πως οι συσκευές iOS έχουν περιορισμένες διαφορές στο υλικό τους. Αυτό σημαίνει πως υπάρχουν πολύ λίγες διαφορές από συσκευή σε συσκευή όσο αφορά το υλικό τμήμα της συσκευής που εκτελεί την κινητή σύνδεση δεδομένων. Αυτό μας επιτρέπει να έχουμε μεγαλύτερη εμπιστοσύνη στα αποτελέσματα των μετρήσεων επειδή γνωρίζουμε πως το υλικό των συσκευών είναι παρεμφερές με πολύ λίγες αλλαγές σε αντίθεση με τις συσκευές που βασίζονται στο λειτουργικό σύστημα Android οι οποίες αριθμούν εκατοντάδες διαφορετικές συσκευές από διαφορετικούς κατασκευαστές, γεγονός που πιθανός να κάνει τις μετρήσεις σε κινητά με λειτουργικό σύστημα Android λιγότερο αξιόπιστες.

Οι συσκευές με λειτουργικό σύστημα iOS έχουν επίσης το πλεονέκτημα πως υποστηρίζονται με αναβαθμίσεις λογισμικού για πολύ περισσότερο χρόνο σε σύγκριση με τις κινητές συσκευές με λειτουργικό σύστημα Android. Χάρη σε αυτό το γεγονός η πλειοψηφία των χρηστών iOS έχει εγκατεστημένη στη συσκευή του το iOS 7 σε αντίθεση με τις συσκευές Android στις οποίες η πλειοψηφία των χρηστών δεν έχει εγκατεστημένη την πιο πρόσφατη έκδοση του λειτουργικού (κάτω από 25% υιοθέτηση του πιο πρόσφατου Android λειτουργικού, 4.4.2 Kit Kat τον Ιούνιο 2014.)

iOS versions

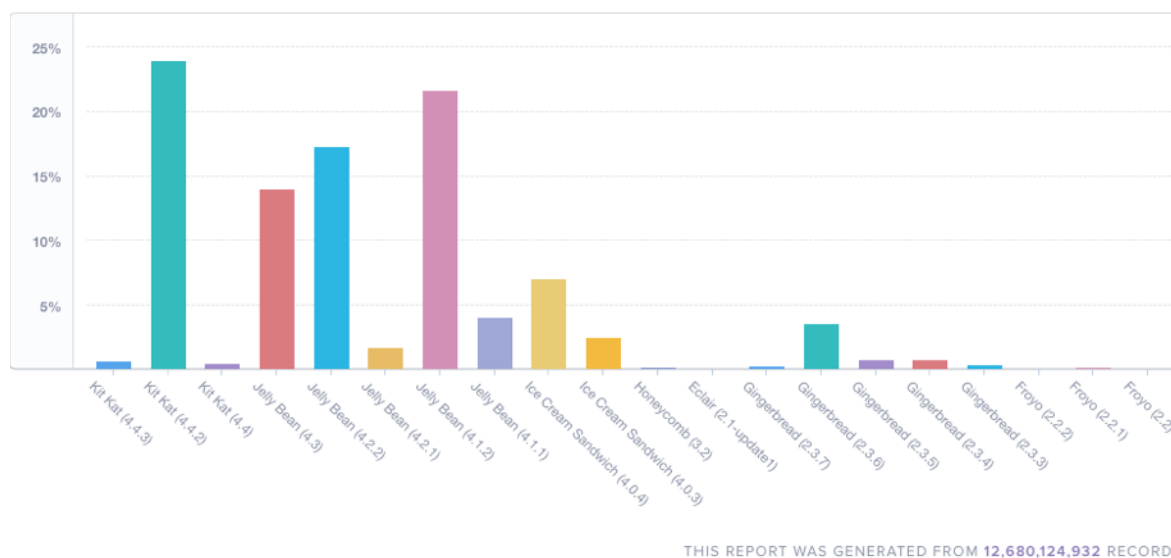
Jun 1, 2014 - Jun 10, 2014



Εικόνα 2.5 - Εγκατεστημένη έκδοση iOS (Πηγή: <https://mixpanel.com>)

Android OS versions

Jun 1, 2014 - Jun 10, 2014



Εικόνα 2.6 - Εγκατεστημένη έκδοση Android (Πηγή: <https://mixpanel.com>)

Παρατηρούμε επίσης πως το ποσοστό των χρηστών που τρέχουν κάποια έκδοση του iOS 7 είναι μεγαλύτερη από 80% ενώ περίπου το 50% των συσκευών Android τρέχουν ακόμη την έκδοση Jelly Bean που διαμοιράστηκε για πρώτη φορά τον Ιούλιο του 2012.

Το γεγονός πως η μεγάλη πλειοψηφία των χρηστών κινητών με λειτουργικό σύστημα iOS έχει εγκατεστημένη την πιο πρόσφατη έκδοση iOS 7 μας δίνει την δυνατότητα να

προγραμματίσουμε με τις πιο σύγχρονες δυνατότητες χωρίς να ανησυχούμε για θέματα συμβατότητας, σε αντίθεση με το λειτουργικό σύστημα Android στο οποίο θα ήμασταν υποχρεωμένοι να προγραμματίσουμε στοχεύοντας ένα λειτουργικό σύστημα δύο ετών και χωρίς την ικανότητα χρήσης των πιο σύγχρονων δυνατοτήτων. Εναλλακτικά θα έπρεπε να προγραμματίσουμε για διαφορετικές εκδόσεις, πράγμα που θα έκανε αυτή την εργασία αδύνατη να υλοποιηθεί λόγω των οικονομικών και χρονικών περιορισμών για την ολοκλήρωση αυτής της εργασίας.

Άλλος ένας παράγοντας που επηρέασε την απόφαση είναι και η ποιότητα και ευκολία χρήσης των προγραμματιστικών εργαλείων που απαιτούνται για προγραμματισμό εφαρμογών κινητών τηλεφώνων. Για τον προγραμματισμό σε iOS χρησιμοποιείται το integrated development environment (IDE) Xcode που το παρέχει ο κατασκευαστής των κινητών τηλεφώνων και του λειτουργικού συστήματος iOS. Για το Android δεν υπάρχει ένα συγκεκριμένο integrated development environment (IDE). Το πιο διάσημο και πιο σύγχρονο integrated development environment (IDE) είναι το Eclipse με την προσθήκη επεκτάσεων που κάνουν δυνατό τον προγραμματισμό για περιβάλλον Android. Οι περισσότεροι που έχουν χρησιμοποιήσει και τα δύο περιβάλλοντα παραδέχονται πως το προγραμματιστικό περιβάλλον Xcode είναι μακράν καλύτερο από τις λύσεις που παρέχονται για προγραμματισμό Android συσκευών. Επίσης ένα ακόμη μεγάλο πλεονέκτημα του Xcode είναι ο πάρα πολύ καλός και γρήγορος device simulator (προσομοιωτής συσκευής) που προσφέρει. Το περιβάλλον ανάπτυξης για εφαρμογές Android προσφέρει και αυτό την δυνατότητα εκτέλεσης εφαρμογών μέσα σε virtual device (εικονικές συσκευές) αλλά δυστυχώς η λύση του Android είναι πάρα πολύ αργή, όπως αναφέρεται από πολλούς προγραμματιστές αλλά και από την δικιά μου εμπειρία. Η δυνατότητα χρήσης εικονικών συσκευών που προσομοιώνουν τις συσκευές τις οποίες θα εκτελείτε ο η εφαρμογή είναι πολύ σημαντική. Επιτρέπει στους προγραμματιστές να ελέγξουν την συμβατότητα των εφαρμογών τους και να ανιχνεύσουν τυχόν λάθη χωρίς την ανάγκη απόκτησης πολλών πραγματικών συσκευών με μεγάλο κόστος. Άλλα πλεονεκτήματα του προγραμματισμού σε iOS είναι η πληθώρα των Application programming interfaces (API) που παρέχονται στον προγραμματιστή που αναπτύσσει εφαρμογές για iOS. Τα APIs επιτρέπουν στον προγραμματιστή να βασίζεται σε ήδη ανεπτυγμένες και δοκιμασμένες προγραμματιστικές λύσεις, πράγμα που αυξάνει την

ποιότητα του πηγαίου κώδικα και μειώνει την συχνότητα λαθών. Αξίζει να σημειωθεί πως πολλά από τα προβλήματα που αντιμετωπίζει ο προγραμματιστής που επιθυμεί να αναπτύξει εφαρμογές για το λειτουργικό σύστημα Android τα έχει αναγνωρίσει και η εταιρία ανάπτυξης του λειτουργικού συστήματος και έχει ξεκινήσει την ανάπτυξη του Android Studio IDE με σκοπό να διορθώσει κάποια από τα προβλήματα. Παρόλα αυτά η ανάπτυξη του Android Studio είναι ακόμη σε πολύ πρώιμο στάδιο και δεν υποστηρίζει πολλές από τις λειτουργίες που είναι απαραίτητες για την ανάπτυξη εφαρμογών στο επίπεδο αυτής της εργασίας. (11)(12)

Σημαντικός, επίσης, παράγοντας για την απόφαση να αναπτυχθεί η εφαρμογή αυτής της εργασίας για συσκευές με λειτουργικό σύστημα iOS ήταν το προσωπικό μου ενδιαφέρον για εκμάθηση καινούργιων προγραμματιστικών δεδομένων. Μέσα στα πλαίσια των σπουδών μου ήρθα σε επαφή με τον προγραμματισμό για συσκευές Android και με την γλώσσα και τα εργαλεία προγραμματισμού που χρησιμοποιούνται. Για την ανάπτυξη εφαρμογών για συσκευές iOS απαιτείται η χρήση διαφορετικής γλώσσας προγραμματισμού, με την οποία είχα ελάχιστη εμπειρία αλλά πολύ ενδιαφέρον για μάθηση, και διαφορετικά εργαλεία και διαδικασίες προγραμματισμού. Έτσι επέλεξα να αναπτύξω την εφαρμογή για κινητές συσκευές με λειτουργικό σύστημα iOS ώστε μέσα στα πλαίσια της εργασίας να αποκτήσω ακόμη περισσότερες καινούργιες γνώσεις.

Τελειώνοντας, είμαι κάτοχος ηλεκτρονικού υπολογιστή τύπου Mac. Οι υπολογιστές Mac είναι οι μόνοι υπολογιστές που υποστηρίζονται από το περιβάλλον ανάπτυξης λογισμικού για iOS επειδή παρέχονται από την ίδια εταιρία. Επίσης το Χαροκόπιο πανεπιστήμιο έχει πρόγραμμα προγραμματισμού για iOS που μου έδινε την δυνατότητα να μπορώ να προγραμματίσω και να εκτελέσω την εφαρμογή μου. Οπότε πληρούσα και όλες τις προϋποθέσεις για ανάπτυξη εφαρμογών για iOS.

2.4 Εξήγηση των τεχνολογιών που χρησιμοποιήθηκαν στην ηλεκτρονική πλατφόρμα.

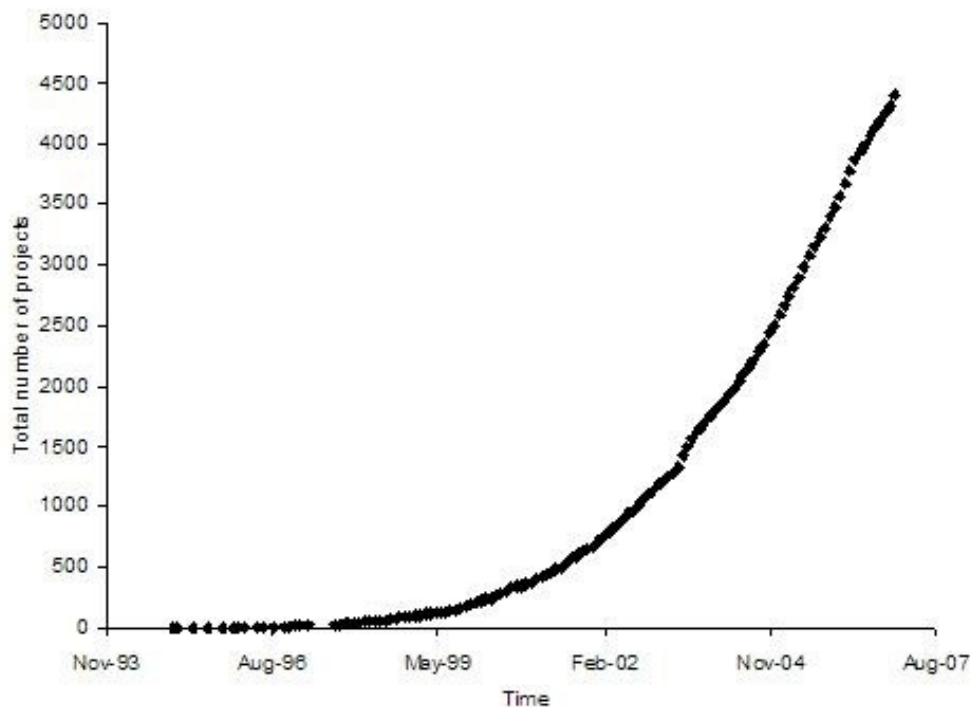
Η ηλεκτρονική πλατφόρμα που υποστηρίζει την εφαρμογή αποτελείται από πολλά τμήματα και πακέτα λογισμικού. Γι' αυτό τον λόγο έπρεπε να είμαστε σίγουροι πως τα διάφορα τμήματα της πλατφόρμας θα μπορούν να λειτουργούν μαζί. Γι' αυτό η υλοποίηση της ηλεκτρονικής πλατφόρμας βασίστηκε πάνω σε πακέτα λογισμικού που υλοποιούν

ανοιχτά πρότυπα. Τα πιο διάσημα πακέτα λογισμικού που υλοποιούν αυτά τα πρότυπα είναι επίσης και λογισμικά ανοιχτού κώδικα που μας προσφέρουν ακόμη περισσότερα πλεονεκτήματα.

Ο όρος λογισμικό ανοιχτού κώδικα (open source software) καθώς και ο όρος ανοιχτό λογισμικό (open software) αναφέρονται σε προγράμματα των οποίων ο πηγαίος κώδικας (source code) είναι προσβάσιμος από οποιονδήποτε έχει το ενδιαφέρον να ασχοληθεί. Επίσης παρέχονται δωρεάν και στην περίπτωση λογισμικού για εξυπηρετές που βασίζονται σε τεχνολογίες web τα προγράμματα ανοιχτού λογισμικού είναι τις περισσότερες φορές τα διασημότερα χάρη στην στήριξη εταιριών όπως η Google και το Facebook. Είναι αξιοσημείωτο πως από το 2000 και μετά υπήρξε μια έκρηξη από προγράμματα ανοιχτού κώδικα που εξυπηρετούν πολλά είδη αναγκών όπως φαίνεται στην εικόνα 2.7.

Έτσι χάρη στην απόφασή μας για την χρήση ανοιχτού λογισμικού και το πλήθος των προγραμμάτων που παρέχονται είχαμε την δυνατότητα να υλοποιήσουμε ολόκληρη την διαδικτυακή πλατφόρμα χρησιμοποιώντας εξ' ολοκλήρου προγράμματα ανοιχτού κώδικα. Δίνοντας έτσι την δυνατότητα στον καθένα να υλοποιήσει χωρίς κόστος την διαδικτυακή πλατφόρμα που υλοποιήσαμε και παραμετροποιήσαμε.

Αναφορικά, τα προγράμματα που χρησιμοποιήθηκαν για την υλοποίηση της διαδικτυακής πλατφόρμας είναι τα εξής: Για λειτουργικό σύστημα επιλέχθηκε το Ubuntu το οποίο είναι βασισμένο στον ανοιχτού κώδικα πυρήνα λογισμικού Linux. Το Linux χρησιμοποιείτε από τις μεγαλύτερες εταιρίες που παρέχουν υπηρεσίες web. Μερικές από αυτές είναι η Google, το Facebook και η Amazon.



Εικόνα 2.7 - Αριθμός των προγραμμάτων ανοιχτού λογισμικού (Πηγή <http://dirkriehle.com>)

Για εξυπηρέτη ιστού χρησιμοποιήσαμε τον Apache HTTP server ο οποίος είναι και ο πιο διάσημος εξυπηρέτης ιστού στον κόσμο. Υπολογίζεται πως το 58,13% όλων των ενεργών ιστότοπων χρησιμοποιούν τον Apache HTTP Server. (13)

Σαν σύστημα διαχείρισης βάσεων δεδομένων χρησιμοποιήσαμε την MySQL η οποία χρησιμοποιείται και αυτή από μεγάλο μέρος του ιστού.

Επιλέγαμε επίσης τον Apache Tomcat για περιβάλλον εκτέλεσης του Application Server. Ο Apache Tomcat μας δίνει την δυνατότητα εκτέλεσης εφαρμογών ιστού που έχουν αναπτυχθεί με την γλώσσα προγραμματισμού Java.

Τέλος για Application Server επιλέξαμε τον Geoserver. Ο Geoserver είναι μια διάσημη εφαρμογή που επιτρέπει την επεξεργασία, προβολή και διαμοιρασμό γεωχωρικών δεδομένων. Υποστηρίζει επίσης το διεθνές πρότυπο εξυπηρέτησης εικόνων γεωαναφοράς (Geo-reference) Web Map Service (WMS) πάνω στο οποίο βασίστηκε η απεικόνιση των αποτελεσμάτων των μετρήσεων μας.

Το Web Map Service (WMS) είναι ένα πρωτόκολλο για την εξυπηρέτηση υπηρεσιών γεωαναφοράς. Προτυποποιήθηκε από τον οργανισμό Open Geospatial Consortium και εκδόθηκε για πρώτη φορά τον Απρίλιο του 2000. Το WMS μπορεί να περιγράψει ένα μεγάλο αριθμό αιτήσεων για γεωαναφορικά δεδομένα με τρόπο ο οποίος επιτρέπει μεγάλη συμβατότητα. Τα δεδομένα συνήθως προσφέρονται με μορφή εικόνας και το WMS μπορεί να περιγράψει τον ακριβή τύπο εικόνας που μπορεί να εξυπηρετηθεί.

2.5 Άλλες υλοποιήσεις και κατευθύνσεις.

Στα πλαίσια του σχεδιασμού αλλά και της δοκιμής αυτής της εργασίας διερευνήσαμε και άλλες εφαρμογές για κινητά τηλέφωνα τύπου iOS που βρίσκονται στο κατάστημα εφαρμογών App Store που προσφέρουν παρόμοιες υπηρεσίες με την δικιά μας εφαρμογή. Τονίζεται επίσης πως όλες οι εφαρμογές που δοκιμάστηκαν αποκτήθηκαν χωρίς την καταβολή χρημάτων. Έτσι αναφέρεται πως δεν ελέγχθηκαν εφαρμογές που απαιτούν την αγορά τους κυρίως λόγω της έλλειψης οικονομικών πόρων της εργασίας.

Οι εφαρμογές που δοκιμάστηκαν είναι:

- speedtest.net Mobile Speed Test, Ookla
- OpenSignal, Opensignal Inc.
- Cell Phone Coverage Map, RootMetrics
- Sensorly-Speedtests & 4G/LTE, CDMA, GSM and Wi-Fi coverage and speed maps, Sensorly
- Network Analyzer Lite - wifi scanner, ping and net info , Technet

Η εφαρμογή speedtest.net Mobile Speed Test της Ookla, είναι μια διάσημη εφαρμογή που προσφέρει μέτρηση ταχύτητας δικτύου και μέτρηση ping. Επίσης προσφέρει και εφαρμογή για εκτέλεση μετρήσεων μέσω web. Η εφαρμογή για κινητά τηλέφωνα iOS επιτρέπει την εκτέλεση δοκιμής ταχύτητας δικτύου και ping. Εμφανίζει τα αποτελέσματα του χρήστη και κρατάει ένα ιστορικό μετρήσεων. Επίσης μετράει και αποθηκεύει δεδομένα σχετικά με την τοποθεσία του χρήστη, αλλά δεν αποτυπώνει τα αποτελέσματα του χρήστη ή άλλων χρηστών σε χάρτη. Δεδομένα σχετικά με τις μετρήσεις άλλων χρηστών παρέχονται μόνο μέσω της σελίδας netindex.com χωρίς γεωγραφικές πληροφορίες και χωρίς την αποτύπωση των μετρήσεων του χρήστη σε χάρτη. Επίσης τα δεδομένα που παρέχονται από την σελίδα

netindex.com είναι στατιστικά επεξεργασμένα δεδομένα που έχουν αποκτηθεί από την χρήση της εφαρμογής. Υπάρχει μία προβολή χάρτη στην σελίδα netindex.com αλλά δεν παρουσιάζει μετρήσεις που έχουν εκτελεστεί από χρήστες αλλά προσφέρει με ένα γραφικό τρόπο στατιστικά επεξεργασμένες πληροφορίες σχετικά με την απόδοση των δικτύων. Επίσης ο χάρτης δεν είναι αναλυτικός και δεν αποτυπώνει γεωγραφικές πληροφορίες. Παράλληλα η εφαρμογή για κινητά iOS υποστηρίζεται από διαφημίσεις που εμφανίζονται στο κάτω μέρος της οθόνης του κινητού τηλεφώνου. Δίνεται η επιλογή στον χρήστη να αφαιρέσει τις διαφημίσεις με καταβολή χρηματικού ποσού αξίας 0,89€. Ένα από τα μεγάλα πλεονεκτήματα που προσφέρει η χρήση της εφαρμογής speedtest.net Mobile Speed Test από την Ookla είναι οι μεγάλης ακρίβειας μετρήσεις που προσφέρει. Οι μετρήσεις εκτελούνται ανάμεσα στην συσκευή του χρήστη και ένα εξαιρετικά μεγάλο δίκτυο από server σε όλο τον κόσμο. Χάρη σε αυτό το πολύ μεγάλο δίκτυο από server η απόσταση του χρήστη από ένα από αυτούς τους εξυπηρετές είναι συνήθως αρκετά μικρή και έτσι τα αποτελέσματα των μετρήσεων ταχύτητας δικτύου και ping επηρεάζονται ελάχιστα από την απόσταση χρήστη-εξυπηρετή.

Η εφαρμογή μπορεί να εγκατασταθεί από αυτή την τοποθεσία: <https://itunes.apple.com/gr/app/speedtest.net-mobile-speed/id300704847?mt=8> (6/2014)

Η εφαρμογή OpenSignal της OpenSignal, Inc είναι μία εφαρμογή που μετράει την ένταση σήματος δικτύων κινητής τηλεφωνίας καθώς και την τεχνολογία υπηρεσίας που προσφέρεται σε κάθε περιοχή και δημιουργεί ένα χάρτη βασισμένο σε μετρήσεις χρηστών που αποτυπώνει την ύπαρξη σήματος ανά περιοχή καθώς και την ένταση του. Επίσης προσφέρει ένα δείκτη στον χρήστη που υποδηλώνει την κατεύθυνση της κεραίας μετάδοσης του σήματος κινητής τηλεφωνίας στο οποίο είναι συνδεδεμένος.

Ο χρήστης όπως δεν έχει την δυνατότητα να εξετάσει τον χάρτη με μεγάλη ακρίβεια γιατί τα δεδομένα παύουν να απεικονίζονται εάν ο χρήστης εστιάσει αρκετά σε κάποια περιοχή. Έχει ακόμη την δυνατότητα απεικόνισης ελεύθερων δικτύων Wi-Fi με βάση δεδομένα των χρηστών. Βασισμένοι στην χρήση της συσκευής και στην περιγραφή της παρατηρούμε πως κύριος σκοπός της εφαρμογής είναι η ενημέρωση των χρηστών για τις περιοχές με το βέλτιστο σήμα κινητής τηλεφωνίας ή την ενημέρωση για ανοικτά δίκτυα τύπου Wi-Fi στα οποία μπορεί ο χρήστης να συνδεθεί. Τα δεδομένα που παρέχει η εφαρμογή δεν αποτυπώνουν στοιχεία όπως η ταχύτητα μετάδοσης δεδομένων ή καθυστέρηση πακέτων του δικτύου (ping). Η εφαρμογή παρέχει λειτουργία μέτρησης της ταχύτητας του δικτύου

αλλά δεν αποτυπώνει τα αποτελέσματα του χρήστη ή άλλων χρηστών της εφαρμογής στον χάρτη. Επίσης αξίζει να σημειωθεί πως αυτή η εφαρμογή μας έδωσε τα μικρότερα αποτελέσματα ταχύτητας σε δοκιμές των εφαρμογών που πραγματοποιήσαμε και θα συζητηθεί παρακάτω. Τέλος η εφαρμογή αναφέρει πως τα αποτελέσματα της δοκιμής μέτρησης της απόδοσης και ταχύτητας του δικτύου αποστέλλονται σε εξυπηρετές της εταιρίας, υπάρχει ωστόσο η δυνατότητα απενεργοποίησης της αποστολής των μετρήσεων. Η εφαρμογή διατίθεται δωρεάν χωρίς διαφημίσεις.

Η εφαρμογή μπορεί να εγκατασταθεί από αυτή την τοποθεσία: <https://itunes.apple.com/gr/app/opensignal/id598298030?mt=8> (6/2014)

Η εφαρμογή Cell Phone Coverage Map από την RootMetrics είναι και αυτή μία εφαρμογή που μετράει την ένταση του δικτύου κινητής τηλεφωνίας στο οποίο είναι συνδεδεμένη η συσκευή του χρήστη καθώς και την ταχύτητα του δικτύου στο οποίο ο χρήστης είναι συνδεδεμένος. Η εφαρμογή προσφέρει ένα χάρτη με δύο επιλογές απεικόνισης. Ο χρήστης μπορεί να επιλέξει να απεικονίζετε η ποιότητα του σήματος ή η ταχύτητα του δικτύου κινητής τηλεφωνίας. Η απεικόνιση των δεδομένων γίνεται χωρίζοντας τον χάρτη σε μεγάλες εξαγωνικές περιοχές και σε κάθε περιοχή συνδυάζονται όλα τα δεδομένα χρηστών και εμφανίζεται η μέση απόδοση με βάση όλες τις μετρήσεις μέσα στην εξαγωνική περιοχή. Δηλαδή η εφαρμογή δεν απεικονίζει ατομικά τα δεδομένα χρηστών αλλά τα ομαδοποιεί. Επίσης η κλίμακα που χρησιμοποιείται στον χάρτη που αποτυπώνει τις ταχύτητες των δεδομένων σε δίκτυα κινητής τηλεφωνίας δεν διαφοροποιεί ταχύτητες πάνω από 1 Mbps. Εμφανίζονται

τέσσερις κατηγορίες ταχύτητας οι οποίες είναι 0-149 kbps, 150-250 kbps, 250-999 kbps και 1 Mbps και άνω. Αυτό σημαίνει πως δεν αποτυπώνονται με μεγάλη ακρίβεια διαφορές ανάμεσα σε ταχύτητες που μπορούν να επιτευχθούν με από κάποια δίκτυα τρίτης (3G) και τέταρτης (4G) γενιάς. Η εφαρμογή δεν αποτυπώνει ταχύτητες δικτύων Wi-Fi αλλά ο χρήστης μπορεί να εκτελέσει μέτρηση ταχύτητας όταν είναι συνδεδεμένος σε δίκτυο Wi-Fi. Ένα μοναδικό χαρακτηριστικό αυτής της εφαρμογής είναι η δυνατότητα που δίνει στον χρήστη να υποβάλει αναφορές σχετικές με την εμπειρία του χρησιμοποιώντας το δίκτυο κινητής τηλεφωνίας. Έτσι ο χρήστης, εάν το επιθυμεί, καλείται να συμπληρώσει ένα απλό ερωτηματολόγιο στο οποίο απλά υποδεικνύει το πρόβλημα, την τοποθεσία και την ώρα του συμβάντος. Η εφαρμογή Cell Phone Coverage Map, RootMetrics διατίθεται δωρεάν και δεν παρουσιάζει διαφημίσεις.

Η εφαρμογή μπορεί να εγκατασταθεί από αυτή την τοποθεσία: <https://itunes.apple.com/gr/app/cell-phone-coverage-map/id399701910?mt=8> (6/2014)

Η εφαρμογή Sensorly-Speedtests & 4G/LTE, CDMA, GSM and Wi-Fi coverage and speed maps η οποία θα αναφέρεται στο εξής ως sensorly, της εταιρίας sensorly είναι μία εφαρμογή που συγκεντρώνει δεδομένα σχετικά με την ένταση σήματος και την τεχνολογία του δικτύου κινητής τηλεφωνίας στο οποίο είναι συνδεδεμένος ο χρήστης. Επίσης πραγματοποιεί μετρήσεις της ταχύτητας του δικτύου. Εμφανίζει τα αποτελέσματα σε ένα χάρτη ο οποίος έχει επιλογές για προβολή έντασης σήματος η προβολή ταχύτητας δικτύου κινητής τηλεφωνίας. Στον χάρτη δεν εμφανίζονται οι μετρήσεις που πραγματοποιούνται χρησιμοποιώντας δίκτυο τύπου Wi-Fi. Ο χρήστης έχει την δυνατότητα να πραγματοποιήσει μέτρηση ταχύτητας του δικτύου Wi-Fi στο οποίο είναι συνδεδεμένη η συσκευή του, παρόλα αυτά δεν υπάρχει δυνατότητα πραγματοποίησης μέτρησης της καθυστέρησης των πακέτων του δικτύου η αλλιώς λειτουργία ping. Στον χάρτη που αποτυπώνονται οι μετρήσεις ταχύτητας δικτύου δεν αποτυπώνετε με κλίμακα η ταχύτητα της μέτρησης αλλά υπάρχει μία κλίμακα με 3 επίπεδα που αποτυπώνει τον αριθμό των μετρήσεων ανά περιοχή. Σκοπός της εφαρμογής είναι η αποτύπωση της κάλυψης σήματος που προσφέρει ο πάροχος της σύνδεσης κινητής τηλεφωνίας βασισμένο σε αποτελέσματα χρηστών. Επίσης προσφέρεται και η λειτουργία μέτρησης της απόδοσης του δικτύου (network performance). Η εφαρμογή sensorly διατίθεται δωρεάν και χωρίς διαφημίσεις.

Η εφαρμογή μπορεί να εγκατασταθεί από αυτή την τοποθεσία: <https://itunes.apple.com/gr/app/sensorly-speedtests-4g-lte/id544269103?mt=8> (6/2014)

Η εφαρμογή Network Analyzer Lite - wifi scanner, ping and net info της Techet είναι μία εφαρμογή που παρέχει στον χρήστη πληροφορίες σχετικά με το δίκτυο το οποίο είναι συνδεδεμένος. Εάν ο χρήστης είναι συνδεδεμένος σε δίκτυο τύπου Wi-Fi εμφανίζονται στον χρήστη πληροφορίες όπως η τοπική διεύθυνση IP, ο DNS Server που χρησιμοποιείται και η διεύθυνση IP του δρομολογητή (router) του δικτύου. Εάν ο χρήστης είναι συνδεδεμένος σε δίκτυο κινητής τηλεφωνίας εμφανίζονται πληροφορίες όπως το όνομα του παρόχου ο τύπος δικτύου και η διεύθυνση IP της συσκευής. Η εφαρμογή δεν εκτελεί μέτρηση ταχύτητας δικτύου αλλά μπορεί να εκτελέσει εκτεταμένη δοκιμή καθυστέρησης δικτύου (ping test) με την δυνατότητα να επιλέξει ο χρήστης την διεύθυνση του server. Επίσης όταν η συσκευή είναι συνδεδεμένη σε δίκτυο Wi-Fi ο χρήστης μπορεί να

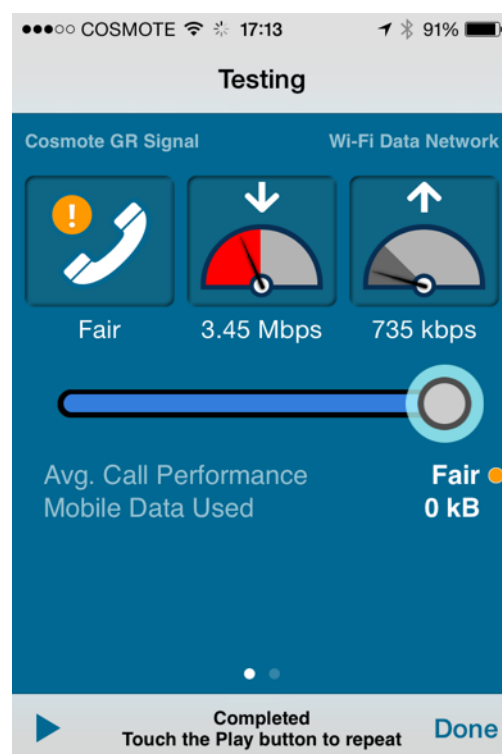
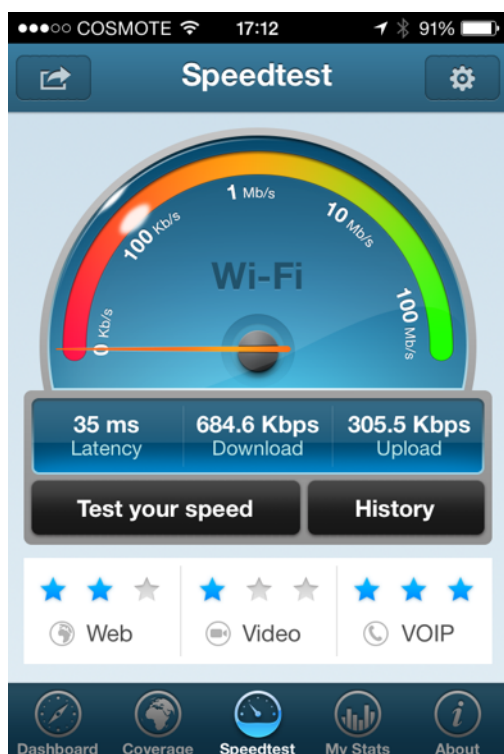
πραγματοποιήσει μία σάρωση του δικτύου για άλλες συνδεδεμένες συσκευές (LAN Scan). Τα αποτελέσματα των μετρήσεων δεν αποστέλλονται και δεν αποθηκεύονται. Δεν υπάρχει κάποια απεικόνιση μετρήσεων των άλλων χρηστών της εφαρμογής. Η εφαρμογή στοχεύει να εξυπηρετήσει τις ανάγκες πιο τεχνικά γνωστικών χρηστών όσο αφορά δικτυακές τεχνολογίες. Η εφαρμογή διατίθεται δωρεάν με παρουσία διαφημίσεων. Επίσης υπάρχει προώθηση για έκδοση της εφαρμογής επί πληρωμή (2,69€) με περισσότερες λειτουργίες. Η εφαρμογή μπορεί να εγκατασταθεί από αυτή την τοποθεσία: <https://itunes.apple.com/gr/app/network-analyzer-lite-wifi/id562315041?mt=8> (6/2014)

Μετά την παρουσίαση των διαφορετικών εφαρμογών θα ήθελα να σχολιάσω πως τα αποτελέσματα των μετρήσεων ταχύτητας, από τις εφαρμογές που προσφέρουν αυτή την λειτουργία, παρουσιάζουν εξαιρετικά μεγάλες διαφορές που παρατηρούνται ακόμα και μετά από μεγάλο αριθμό δοκιμών σε κάθε εφαρμογή. Για να μπορέσω να εξηγήσω το μέγεθος της διαφοράς αποτελεσμάτων θα παραθέσω τα αποτελέσματα που έλαβα μετά από την εκτέλεση δοκιμής ταχύτητας σε κάθε μία από τις εφαρμογές που υποστηρίζουν αυτή την λειτουργία. Η δοκιμή διεξήχθη χρησιμοποιώντας ένα κινητό Apple iPhone 4S που εκτελεί την έκδοση λειτουργικού συστήματος iOS 7.1.1. Το κινητό τηλέφωνο ήταν συνδεδεμένο σε ασύρματο δίκτυο IEEE 802.11n στα 2.4 Ghz. Η σύνδεση στο διαδίκτυο είναι μία φοιτητική σύνδεση OTE ADSL με ονομαστική ταχύτητα κατεβάσματος 24 Mbps (η οποία όμως δεν μπορεί να επιτύχει πάνω από 13 Mbps) και ονομαστική ταχύτητα ανεβάσματος 1Mbps. Την στιγμή που διενεργήθηκαν οι δοκιμές το κινητό τηλέφωνο iPhone ήταν η μόνη συσκευή που ήταν συνδεδεμένη στο δίκτυο. Οι δοκιμές διεξήχθησαν στις 11 Ιουνίου 2014 στο χρονικό διάστημα από τις 17:12 έως τις 17:16. Επίσης διεξήχθη και μία δοκιμή στις 17:23 την ίδια ημέρα από προσωπικό υπολογιστή ο οποίος είναι συνδεδεμένος με το δρομολογητή χρησιμοποιώντας μία σύνδεση Gigabit Ethernet, η δοκιμή έγινε χρησιμοποιώντας την ιστοσελίδα speedtest.net. Ο δρομολογητής ιστού που χρησιμοποιήθηκε ήταν ο Google Chrome version 35.0.1916.114.

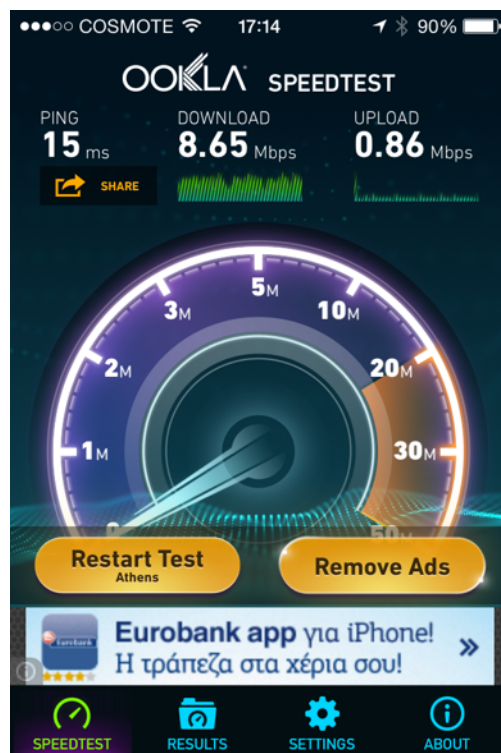
Στον πίνακα αποτυπώνονται τα αποτελέσματα των μετρήσεων. Ο αριθμός δίπλα σε κάθε όνομα υποδηλώνει την αντίστοιχη εικόνα.

	Download (Mbps)	Upload (kbps)
OpenSignal (2.6)	0.684	305
RootMetrics (2.7)	3.45	735
Sensorly (2.8)	5.98	1050
<u>speedtest.net</u> Mobile (2.9)	8.65	860
Εργασία (2.10)	8.13	624
<u>speedtest.net</u> Desktop (2.11)	11.52	850

Παρακάτω παρουσιάζονται τα αποσπάσματα οθόνης που ελήφθησαν μετά από κάθε εκτέλεση μέτρησης (σημειώστε ότι είναι ορατή και η ώρα λήψης κάθε αποσπάσματος οθόνης)



Εικόνες 2.8, 2.9



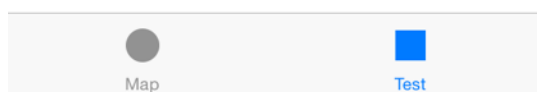
Εικόνες 2.10, 2.11

COSMOTE 17:21 90%

Begin SpeedTest

Download Speed:	8137 kbps
Upload Speed:	624 kbps
Ping:	30 ms
Ping Google:	12 ms

Connected using WiFi



Εικόνα 2.12



Εικόνα 2.13

Είναι ολοφάνερο πως οι μετρήσεις εμφανίζουν δραματικά μεγάλες διαφορές. Μπορούμε να εικάσουμε τους πιθανούς λόγους των διαφορών αυτών αλλά μετά και από την εμπειρία που απέκτησα κατά την διάρκεια της εργασίας πιστεύω πως ο λόγος για τις μεγάλες αυτές διαφορές είναι οι διαφορετικές υλοποιήσεις που χρησιμοποιούνται για την εκτέλεση της μέτρησης ταχύτητας από αυτές τις εφαρμογές. Ο λόγος για τις διαφορετικές υλοποιήσεις είναι η παντελής έλλειψη λειτουργίας στο σύστημα iOS που να παρέχει στον προγραμματιστή την τωρινή ταχύτητα σύνδεσης καθώς και στην δυσκολία υλοποίησης τέτοιας λειτουργίας λόγω περιορισμών που θα εξηγηθούν παρακάτω στο κεφάλαιο (\$ προσθήκη κεφαλαίου που εξηγώ το ios networking). Άλλος λόγος είναι η χρήση διαφορετικών εξυπηρετών (Server) που χρησιμοποιούνται για την πραγματοποίηση της μέτρησης της ταχύτητας δικτύου. Ακόμη υπάρχει και η γενικότερη δυσκολία πραγματοποίησης ακριβούς μέτρησης ταχύτητας δικτύου σε κινητές συσκευές καθώς και άλλοι απρόβλεπτοι παράγοντες που επηρεάζουν την απόδοση ενός δικτύου με την χρήση διαφορετικών τεχνολογιών δικτύωσης.

3 Υλοποίηση διαδικτυακής πλατφόρμας (Server)

3.1 Πληροφορίες σχετικά με τον Server

Ο server της εργασίας συμπεριλαμβάνει ένα μεγάλο πλήθος από διαφορετικές τεχνολογίες που συνδυάζονται για να μας προσφέρουν την λειτουργικότητα που επιθυμούμε.

Πολλές τεχνολογίες από αυτές που περιγράφουμε, μπορούν να αντικατασταθούν από άλλες παρεμφερείς σε λειτουργικότητα επιλογές. Για αυτή την εργασία πάρθηκε η συνειδητή απόφαση να βασιστούμε σε ανοιχτού τύπου λύσεις που συμμορφώνονται με διεθνή ανοιχτά πρότυπα.

Οι κυριότεροι λόγοι για τους οποίους βασιζόμαστε σε ανοιχτού λογισμικού λύσεις είναι αρχικά το κόστος, αλλά επίσης και η ευελιξία που μας προσφέρουν στο να κάνουμε διαφορετικές επιλογές για διαφορετικά τμήματα του server. Επειδή όλες οι λύσεις που χρησιμοποιούνται είναι βασισμένες σε διεθνή ανοιχτά πρότυπα μπορούμε να είμαστε βέβαιοι πως τα διάφορα πακέτα λογισμικού που θα χρησιμοποιήσουμε θα μπορούν να συνεργαστούν με όλα τα άλλα τμήματα της υλοποίησης μας. Επίσης είναι σημαντικό να τονίσουμε πως για όλα τα πακέτα λογισμικού που χρησιμοποιούμε υπάρχει μια αφθονία πληροφοριών που μπορεί ο καθένας να προσπελάσει στο διαδίκτυο, κυρίως λόγω της ανοιχτής τους υλοποίησης.

3.2 Εξήγηση των τεχνολογιών που χρησιμοποιούνται στον Server.

Επειδή η λειτουργικότητα της πλατφόρμας που υλοποιήσαμε εξαρτάται από την ύπαρξη ενός επεξεργαστικά ικανού και δικτυακά ευέλικτου server αποφασίσαμε να υλοποιήσουμε τον sever της πλατφόρμας στην υπηρεσία cloud του okeanos (okeanos.grnet.gr) που μας προσφέρει δωρεάν εικονικά μηχανήματα που έχουν αρκετή υπολογιστική ισχύ για να καλύψουν τις ανάγκες μας καθώς και εξαιρετικά καλή σύνδεση στο Internet μέσω οπτικών ινών καθώς και συνεχή λειτουργία.

Το λειτουργικό σύστημα στο οποίο βασίσαμε την υλοποίηση του εξυπηρέτη της πλατφόρμας μάς είναι το Ubuntu 13.10 που ήταν μέσα στις επιλογές που μας προσέφερε ο okeanos για άμεση δημιουργία του εικονικού μηχανήματος που μας προσφέρει. Άλλοι λόγοι για τους οποίους επιλέξαμε το λειτουργικό σύστημα Ubuntu 13.10 είναι επειδή είναι ένα λειτουργικό σύστημα τύπου Linux που σημαίνει πως είναι λογισμικό ανοιχτού κώδικα που προσφέρεται χωρίς χρέωση. Επίσης είναι πλήρως συμβατό με όλα τα άλλα πακέτα λογισμικού που χρησιμοποιήθηκαν για την υλοποίηση του εξυπηρέτη της ηλεκτρονικής μας πλατφόρμας και είναι γνωστό για την αξιοπιστία του σε περιβάλλοντα server όπου απαιτείται συνεχής λειτουργία.

Για εξυπηρέτη της πλατφόρμας παροχής των γεωχωρικών δεδομένων επιλέξαμε την πλατφόρμα του Geoserver (geoserver.org). Το Geoserver είναι και αυτό λογισμικό ανοιχτού κώδικα, υλοποιημένο στην γλώσσα προγραμματισμού Java, που είναι συμβατό με τις κυριότερες τεχνολογίες παροχής γεωχωρικών δεδομένων όπως αυτές έχουν οριστεί από οργανισμό Open Geospatial Consortium (www.opengeospatial.org). Χρησιμοποιεί open standards για να επιτύχει την λειτουργικότητα του και είναι εύκολο να ενσωματωθεί με τις δημοφιλέστερες υπηρεσίες παροχής χαρτών (όπως Google Maps, OpenStreetMaps και Yahoo Maps). Γι αυτό τον λόγο παρέχει μεγάλη ευελιξία στην δημιουργία απεικονίσεων γεωχωρικών δεδομένων καθώς και στον διαμοιρασμό των δεδομένων.

Ο Geoserver τρέχει σε περιβάλλον Apache Tomcat (tomcat.apache.org). Ο Apache Tomcat είναι ένα πακέτο λογισμικού ανοιχτού κώδικα που συμπεριλαμβάνει τον Apache HTTP server. Ο Apache Tomcat προσφέρει ένα περιβάλλον εκτέλεσης για εφαρμογές τύπου Java που λειτουργούν σαν εξυπηρέτες ιστού και υποστηρίζουν συνδεσιμότητα μέσω HTTP, όπως είναι και ο Geoserver.

Για εξυπηρέτη ιστού χρησιμοποιούμε τον Apache HTTP server. Ο Apache HTTP server είναι ο πιο διάσημος εξυπηρέτης ιστού, υπολογίζεται πως το 58.13% όλων των ενεργών ιστότοπων χρησιμοποιούν τον Apache (13). Ο Apache προσφέρει επίσης το περιβάλλον εκτέλεσης των web services που αναπτύξαμε με σκοπό την μεταφορά των πληροφοριών και μετρήσεων από τους client. Τέλος και ο Apache είναι λογισμικό ανοιχτού κώδικα.

Σαν σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων επιλέξαμε την MySQL. Η MySQL είναι από τα διασημότερα συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων και χρησιμοποιείται από πολλές εξαιρετικά διάσημες ιστοσελίδες όπως είναι η Google (14) και το Facebook (15) .

Αν και γενικότερα ο Geoserver συνήθως συνδέετε με διαφορετικά συστήματα βάσεων δεδομένων τα οποία είναι σχεδιασμένα συγκεκριμένα για την παροχή υπηρεσιών για αποθήκευση και διαχείριση γεωχωρικών δεδομένων όπως είναι η PostGis (16) επιλέξαμε την MySQL επειδή διαθέτουμε εμπειρία στην εγκατάσταση και διαχείριση της καθώς και για το γεγονός πως μπορεί να προσφέρει την λειτουργικότητα που απαιτείται από τον Geoserver με την προσθήκη μιας απλής επέκτασης.

Για εύκολη απομακρυσμένη πρόσβαση και διαχείριση της MySQL εγκαταστήσαμε το phpMyAdmin το οποίο είναι ένα περιβάλλον διαχείρισης της MySQL που προσφέρει ένα περιβάλλον διεπαφής μέσω web.

3.3 Εγκατάσταση και παραμετροποίηση του τμήματος Server της πλατφόρμας.

Αρχικά δημιουργήσαμε ένα εικονικό μηχάνημα μέσω της υπηρεσίας oceanos χρησιμοποιώντας το γραφικό περιβάλλον διεπαφής. Κατά την διαδικασία δημιουργίας του μηχανήματος επιλέξαμε να εγκατασταθεί αυτόματα το λειτουργικό σύστημα Ubuntu 13.10.

Έπειτα συνδεθήκαμε με το απομακρυσμένο μηχάνημα χρησιμοποιώντας την τεχνολογία ssh που προσφέρει μια ασφαλή διεπαφή γραμμής εντολών για την χρήση του λειτουργικού συστήματος.

Για την διαδικασία εγκατάστασης χρειάζεται να εισέλθουμε με δικαιώματα χρήστη root που μπορεί να γίνει με την εντολή `sudo -i`.

Μετά την σύνδεση μας στον server εγκαταστήσαμε την Java 7 που είναι απαραίτητη για την εκτέλεση του Apache Tomcat και του Geoserver. Εκτελέστηκαν οι ακόλουθες εντολές

οι οποίες υποδηλώνουν την πηγή του λογισμικού στο λειτουργικό σύστημα και έπειτα λαμβάνουν τα απαραίτητα αρχεία και εκτελούν την διαδικασία εγκατάστασης:

```
echo "deb http://ppa.launchpad.net/webupd8team/java/ubuntu
precise main" | tee -a /etc/apt/sources.list
echo "deb-src http://ppa.launchpad.net/webupd8team/java/
ubuntu precise main" | tee -a /etc/apt/sources.list
apt-key adv --keyserver keyserver.ubuntu.com --recv-keys
EEA14886
apt-get update
apt-get install oracle-java7-installer
```

Στην συνέχεια ρυθμίζουμε το λειτουργικό ώστε να χρησιμοποιεί την Java 7 ως την προεπιλεγμένη διανομή Java.

```
apt-get install oracle-java7-set-default
update-alternatives --config java
```

Το επόμενο βήμα είναι η παραμετροποίηση του αρχείου /etc/profile για να δηλώσουμε τον κατάλογο που περιέχει Java Runtime Environment της Java 7.

```
JAVA_HOME=/usr/lib/jvm/java-7-oracle
JRE_HOME=/usr/lib/jvm/java-7-oracle/jre
export JAVA_HOME
export JRE_HOME
```

Εν τέλη για να ενεργοποιηθούν οι αλλαγές που δηλώσαμε παραπάνω εκτελούμε την εντολή:

```
source /etc/profile
```

Για την εγκατάσταση του Apache HTTP server και της MySQL εκτελέσαμε την εντολή

```
apt-get install lamp-server^
```

(προσοχή στο ^ στο τέλος)

Η παραπάνω εντολή εγκαθιστά και συνδέει αυτόματα τον Apache HTTP Server την MySQL καθώς και την γλώσσα PHP. Το πακέτο αυτό ονομάζεται LAMP που προέρχεται από τους όρους Linux, Apache, MySQL, PHP. Ο λόγος για τον οποίο υπάρχει το παραπάνω πακέτο είναι επειδή ο συνδυασμός των παραπάνω είναι ένας διάσημος τρόπος για παροχή υπηρεσιών web σε λειτουργικά συστήματα τύπου Linux.

Έπειτα εγκαθιστούμε τον Apache Tomcat μεταβαίνουμε στον φάκελο /usr/local

και κάνουμε λήψη του Apache Tomcat:

```
cd /usr/local
wget http://www.eu.apache.org/dist/tomcat/tomcat-7/v7.0.53/
bin/apache-tomcat-7.0.53.tar.gz
```

Το επόμενο βήμα είναι η αποσυμπίεση του ληφθέντος αρχείου

```
sudo tar xzf apache-tomcat-7.0.53.tar.gz
```

Σε αυτό το σημείο πρέπει να σημειώσουμε πως το παραπάνω αρχείο και ο παραπάνω σύνδεσμος ενδέχεται να μην λειτουργούν. Ο λόγος για τον οποίο μπορεί να μην λειτουργούν είναι επειδή ο Apache Tomcat είναι σε διαρκή ανάπτυξη και συνεχώς παρέχονται καινούργιες εκδόσεις του.

Το επόμενο βήμα στην διαδικασία εγκατάστασης του Apache Tomcat είναι η δημιουργία μιας κατηγορίας χρήστη και ενός εικονικού χρήστη που θα επιτρέπει στον Tomcat να εκτελεί τις λειτουργίες του και να ελέγχει τον αποθηκευτικό του χώρο.

```
groupadd tomcat
useradd -g tomcat -d /usr/local/tomcat -m tomcat
usermod -G www-data tomcat
```

Στη συνέχεια μεταφέρουμε τα αποσυμπιεσμένα αρχεία στον πρόσφατα δημιουργημένο φάκελο `/usr/local/tomcat`, αλλάζουμε τα δικαιώματα του υποφακέλου αυτού ώστε να μπορεί να τα διαχειριστεί ο tomcat και κάνουμε εκτελέσιμα όλα τα scripts που περιέχονται μέσα σε αυτό τον φάκελο

```
mv /usr/local/apache-tomcat-7.0.53 /usr/local/tomcat
chown -R tomcat:tomcat /usr/local/tomcat
chmod +x /usr/local/tomcat/bin/*.sh
```

Έπειτα δημιουργούμε ένα αρχείο script που περιέχει τις παραμέτρους εγκατάστασης και λειτουργίας του Apache Tomcat

```
nano /etc/init.d/tomcat
```

Σε αυτό το αρχείο προσθέτουμε τα παρακάτω:

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:          tomcat
# Required-Start:    $local_fs $remote_fs $network
# Required-Stop:     $local_fs $remote_fs $network
# Should-Start:      $named
# Should-Stop:       $named
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Start Tomcat Server
# Description:       Start the Tomcat Server
### END INIT INFO
# customize values for your needs
JAVA_HOME="/usr/lib/jvm/java-7-oracle"
JRE_HOME="/usr/lib/jvm/java-7-oracle/jre"
TOMCAT_USER=tomcat;
TOMCAT_PATH=/usr/local/tomcat;
export PATH=$JAVA_HOME/bin:$PATH
export CATALINA_HOME=$TOMCAT_PATH
export JAVA_OPTS="$JAVA_OPTS\
-server\
-Xms1024m\
-Xmx1024m\
-XX:+HeapDumpOnOutOfMemoryError\
-XX:HeapDumpPath=/usr/local/tomcat/logs/catalina.out\
-XX:MaxPermSize=256m\
-XX:MaxNewSize=512m\
-XX:NewSize=512m\
-XX:SurvivorRatio=12\
-XX:MaxTenuringThreshold=0\
-XX:+UseConcMarkSweepGC\
-XX:+CMSIncrementalMode\
-XX:+CMSIncrementalPacing\
-XX:+CMSClassUnloadingEnabled\
-XX:+DisableExplicitGC\
-XX:+UseParNewGC\
-XX:+UseTLAB\
-Djava.awt.headless=true\
-Dnewrelic.environment=production"
start() {
    if [ -f $TOMCAT_PATH/bin/startup.sh ];
    then
        echo -n "Starting Tomcat\n"
        /bin/su $TOMCAT_USER -c $TOMCAT_PATH/bin/
startup.sh
    fi
}

stop() {
    if [ -f $TOMCAT_PATH/bin/shutdown.sh ];
    then
        echo -n "Stopping Tomcat\n"
        /bin/su $TOMCAT_USER -c $TOMCAT_PATH/bin/
```

```

shutdown.sh
    fi
}

case "$1" in
    'start')
        start
        ;;
    'stop')
        stop
        ;;
    'restart')
        stop
        start
        ;;
    *)
        echo "Usage: $0 { start | stop | restart}"
        exit 1
        ;;
esac
exit 0

```

Το παραπάνω αρχείο, που δημιουργήσαμε, είναι ένα εκτελέσιμο script που θα μας επιτρέπει να εκκινούμε, να σταματάμε και να επανέκκινούμε τον Apache Tomcat. Επίσης περιέχει πληροφορίες σχετικά με τον κατάλογο που περιέχει το Java Runtime Environment που είναι απαραίτητο για να λειτουργήσει ο Apache Tomcat. Έπειτα δηλώνονται διάφορες παράμετροι για την χρήση της μνήμης του υπολογιστή, για τους καταλόγους αποθήκευσης των αρχείων καταγραφής καθώς και διάφορες άλλες παραμέτρους για την ορθή λειτουργία του Apache Tomcat. Στο τέλος υλοποιείται η λειτουργικότητα της εκκίνησης, της διακοπής λειτουργίας και της επανεκκίνησης του Tomcat.

Μετά δίνουμε πλήρη δικαιώματα στον πρόσφατα δημιουργημένο χρήστη “tomcat” καθώς και δικαιώματα ανάγνωσης και εκτέλεσης σε όλους τους άλλους χρήστες του συστήματος

```
chmod 755 /etc/init.d/tomcat
```

Στην συνέχεια δηλώνουμε τα στοιχεία χρήστη για τη διεπαφή ιστού του tomcat τροποποιώντας το αρχείο /usr/local/tomcat/conf/tomcat-users.xml

```
nano /usr/local/tomcat/conf/tomcat-users.xml
```

Μέσα στο αρχείο προσθέτουμε τα παρακάτω:

```
?xml version='1.0' encoding='utf-8'?>

<tomcat-users>
  <role rolename="manager"/>
    <role rolename="manager-gui"/>
    <role rolename="admin"/>
    <role rolename="admin-gui"/>
    <user username="****" password="****" roles="admin,admin-
gui,manager,manager-gui"/>
</tomcat-users>
```

Με ένα παρόμοιο μοτίβο μπορούμε να δημιουργήσουμε και άλλους χρήστες σε διαφορετικούς ρόλους. Στο πεδίο rolename δηλώνουμε την ιδιότητα του χρήστη ενώ στα πεδία username και password δηλώνουμε τα στοιχεία που θα χρησιμοποιήσει ο χρήστης κατά την διαδικασία login στο web περιβάλλον του Tomcat.

Και τέλος εκκινούμε τον Apache Tomcat

```
etc/init.d/tomcat start
```

Για την εγκατάσταση του Geoserver, αρχικά μεταβαίνουμε στον κατάλογο

/usr/local/tomcat/webapps κάνουμε λήψη των συμπιεσμένων δεδομένων εγκατάστασης

```
cd /usr/local/tomcat/webapps/
wget      http://downloads.sourceforge.net/project/geoserver/
GeoServer/2.5/geoserver-2.5-war.zip
unzip geoserver-2.5-war.zip
```

Έπειτα για να μπορέσει να λειτουργήσει ο Geoserver αυξάνουμε το ελάχιστο μέγεθος εφαρμογής που μπορεί να εκτελέσει ο Apache Tomcat τροποποιώντας το αρχείο

```
nano /usr/local/tomcat/webapps/manager/WEB-INF/web.xml
```

Μέσα στο αρχείο εντοπίζουμε τις γραμμές

```
<multipart-config>
<!-- 50MB max -->
<max-file-size>52428800</max-file-size>
<max-request-size>52428800</max-request-size>
<file-size-threshold>0</file-size-threshold>
</multipart-config>
```

Και τις αντικαθιστούμε με το παρακάτω:

```
<multipart-config>
<!-- 50MB max -->
<max-file-size>626914560</max-file-size>
<max-request-size>626914560</max-request-size>
<file-size-threshold>0</file-size-threshold>
</multipart-config>
```

Με αυτή την αλλαγή επιτρέπουμε εκτελεσιμα αρχεία μεγέθους 626914560 bytes σε αντίθεση με το προεπιλεγμένο όριο των 52428800 bytes.

Τελειώνοντας, μεταφορτώνουμε το απαραίτητο extension για να μπορεί να επικοινωνήσει ο Geoserver με την MySQL και το τοποθετούμε στον κατάλογο :

`/usr/local/tomcat/webapps/geoserver/WEB-INF/lib/`

```
cd /usr/local/tomcat/webapps/geoserver/WEB-INF/lib/
wget      http://sourceforge.net/projects/geoserver/files/
GeoServer%20Extensions/2.5/geoserver-2.5-mysql-plugin.zip
```

```
unzip geoserver-2.5-mysql-plugin.zip
```

Και επανεκκινούμε τον Tomcat

```
etc/init.d/tomcat restart
```

3.4 Παραμετροποίηση της Βάσης δεδομένων

3.4.1 Δημιουργία χρήστη της Βάσης δεδομένων

Για να μπορεί να έχει το web service, το οποίο καταχωρεί τα δεδομένα από τις μετρήσεις , πρόσβαση στον πίνακα measurements της βάσης δεδομένων db1, δημιουργήσαμε έναν χρήστη με αναγνωριστικό 'user'.(Στον χρήστη αυτόν, δώσαμε δικαιώματα προσπέλασης και εγγραφής αποκλειστικά και μόνο προς τους πίνακες της βάσης db1.)

Δημιουργήσαμε τον χρήστη 'user':

```
create user 'user' @ '%' identified by ' users-password';
```

Έπειτα δημιουργήσαμε την βάση δεδομένων db1:

```
create database if not exists 'db1';
```

Τέλος δώσαμε δικαιώματα χρήσης των πινάκων της βάσης db1 στον χρήστη 'user':

```
grant all privileges on 'db1'. * to 'user' @ '%';
```

Και φορτώσαμε εκ νέου τα δικαιώματα των χρηστών για να ενεργοποιηθούν οι αλλαγές μας:

```
flush privileges;
```

3.4.2 Δημιουργία πίνακα σε MySQL βάση για αποθήκευση μετρήσεων και γεωχωρικών δεδομένων

Για την δημιουργία του πίνακα βάσης δεδομένων MySQL ο οποίος θα αποθηκεύει τα δεδομένα που θα αποστέλλονται από την εφαρμογή για κινητά τηλέφωνα, όπως ο τύπος δικτύου(WIFI, GPRS, UMTS κτλ) η ημερομηνία και άλλα καθώς και τα αποτελέσματα μετρήσεων εκτελούμε το παρακάτω SQL query:

```
CREATE TABLE IF NOT EXISTS `measurements` (  
  `ID` int(11) NOT NULL AUTO_INCREMENT,  
  `Position` point NOT NULL,  
  `Downlink` float NOT NULL,  
  `Uplink` float NOT NULL,  
  `Ping1_IPERF` float DEFAULT NULL,  
  `Ping2_DN` float DEFAULT NULL,  
  `Network_Type` varchar(30) NOT NULL,  
  `Measurement_Time` datetime NOT NULL,  
  `latitude` float NOT NULL,  
  `longitude` float NOT NULL,  
  `os` varchar(30) NOT NULL,  
  PRIMARY KEY (`ID`),  
  SPATIAL KEY `Position` (`Position`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Πιο συγκεκριμένα, τα πεδία που χρησιμοποιούμε είναι τα εξής:

- ID: Αναγνωριστικό τύπου ακεραίου (int) που χαρακτηρίζει μοναδικά κάθε καταχώριση, είναι πρωτεύον κλειδί, δεν μπορεί να πάρει κενή τιμή (null) και οι τιμές του καθορίζονται αυτόματα από το σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (ή από το Engine)(Auto increment).
- Position: Πεδίο τύπου point με που αναπαριστά την τοποθεσία στην οποία η συσκευή πήρε την μέτρηση, δεν μπορεί να πάρει κενή τιμή (null) και δημιουργούμε ένα χωρικό

ευρετήριο μέσω της δήλωσης SPATIAL KEY 'KEY_NAME' ('field upon which index is created'), για να λειτουργήσει ορθά (σωστά) με τον Geoserver.

- Downlink: Πεδίο τύπου κινητής υποδιαστολής (float) στο οποίο αποθηκεύονται τα εκάστοτε αποτελέσματα της μέτρησης λήψης που διεξάγει η συσκευή. Τα δεδομένα αποτυπώνουν kbps (kilobit per second). Δεν μπορεί να πάρει κενή τιμή (null).
- Uplink: Πεδίο τύπου κινητής υποδιαστολής (float) στο οποίο αποθηκεύονται τα εκάστοτε αποτελέσματα της μέτρησης αποστολής που διεξάγει η συσκευή. Τα δεδομένα αποτυπώνουν kbps (kilobit per second). Δεν μπορεί να πάρει κενή τιμή (null).
- Ping_IPERF: Πεδίο τύπου κινητής υποδιαστολής (float) στο οποίο αποθηκεύονται τα εκάστοτε αποτελέσματα της μέτρησης ping προς τον προσωπικό μας εξυπηρέτη που διεξάγει η συσκευή. Τα δεδομένα αποτυπώνουν ms (millisecond). Μπορεί να πάρει κενή τιμή (null).
- Ping_DN: Πεδίο τύπου κινητής υποδιαστολής (float) στο οποίο αποθηκεύονται τα εκάστοτε αποτελέσματα της μέτρησης ping προς τον εξυπηρέτη της Google που διεξάγει η συσκευή. Τα δεδομένα αποτυπώνουν ms (millisecond) Μπορεί να πάρει κενή τιμή (null).
- Network_type: Πεδίο τύπου μεταβλητού πλήθους χαρακτήρων με μήκος 30 χαρακτήρων (varchar(30)) στο οποίο αποθηκεύουμε τον τύπο του δικτύου μέσω του οποίου εκτελούμε την μέτρηση. Δεν μπορεί να πάρει κενή τιμή (null).
- Measurement_Time: Πεδίο τύπου μεταβλητού πλήθους χαρακτήρων με μήκος 30 χαρακτήρων (varchar(30)) στο οποίο αποθηκεύουμε τον τύπο του δικτύου μέσω του οποίου εκτελούμε την μέτρηση. Δεν μπορεί να πάρει κενή τιμή (null).
- Latitude: Πεδίο τύπου κινητής υποδιαστολής (float) στο οποίο αποθηκεύουμε το γεωγραφικό πλάτος της τοποθεσίας της συσκευής που εκτέλεσε την μέτρηση. Δεν μπορεί να πάρει κενή τιμή (null).
- Longitude: Πεδίο τύπου κινητής υποδιαστολής (float) στο οποίο αποθηκεύουμε το γεωγραφικό μήκος της τοποθεσίας της συσκευής που εκτέλεσε την μέτρηση. Δεν μπορεί να πάρει κενή τιμή (null).

- OS: Πεδίο τύπου μεταβλητού πλήθους χαρακτήρων με μήκος 30 χαρακτήρων (varchar(30)) στο οποίο αποθηκεύουμε τον τύπο του λειτουργικού συστήματος που διαθέτει η συσκευή η οποία εκτελεί την μέτρηση. Δεν μπορεί να πάρει κενή τιμή (null).

Για την δημιουργία του πίνακα αυτού χρησιμοποιήσαμε την μηχανή αποθήκευσης (Storage Engine) MyISAM καθώς αυτή υποστηρίζει αποθήκευση και διαχείριση γεωγραφικών δεδομένων. Η MyISAM καθώς και αρκετές άλλες storage engines, δημιουργούνε αυτομάτως ένα ευρετήριο (index) όταν δηλώνουμε ένα πεδίο key, οπότε με την δήλωση: SPATIAL KEY `Position` (`Position`)

Δημιουργήθηκε ένα ευρετήριο πάνω στο πεδίο Position του πίνακα, το οποίο είναι απαραίτητο για την επικοινωνία του Geoserver με την βάση.

3.5 Εξήγηση του τρόπου παραλαβής και αποθήκευσης των δεδομένων

Ανοίγουμε σύνδεση προς την βάση δεδομένων με την παρακάτω εντολή, χρησιμοποιώντας τον χρήστη 'user' με τα σωστά αναγνωριστικά.

```
<?php
$con=mysqli_connect("*****","user","*****","db1");
```

Αποθηκεύουμε σε προσωρινές μεταβλητές τα δεδομένα που λαμβάνουμε από το post request.

```
$long=htmlspecialchars($_POST["long"]);
$lat=htmlspecialchars($_POST["lat"]);
$uplink=htmlspecialchars($_POST["uplink"]);
$downlink=htmlspecialchars($_POST["downlink"]);
$ping1=htmlspecialchars($_POST["ping1"]);
$ping2=htmlspecialchars($_POST["ping2"]);
$techType=htmlspecialchars($_POST["techType"]);
$timeDate=htmlspecialchars($_POST["timeDate"]);
$os=htmlspecialchars($_POST["os"]);
if(mysqli_connect_errno()){
    echo "Failed to connect to MySQL db:
    ".mysqli_connect_error();
}
else{
```

Εισάγουμε τα δεδομένα στην βάση μας με ένα insert into query.

```
$err= mysqli_query($con,"insert into measurements values
(null,GeomFromText('POINT('.$long.'" ".$lat.'"')"),".
$downlink.", ".$uplink.", ".$ping1.", ".$ping2.", '".
$techType.'"', '". $timeDate.'"', '". $lat.'"', '". $long.'"', '".
$os.'"')");
if(!$err){
    echo "Failed to connect to MySQL db:
".mysqli_error($con);
}
}
mysqli_close($con); ?>
</br>Passing Data to MySQL DataBase</br></body></html>
```

3.6 Παραμετροποίηση του Geoserver

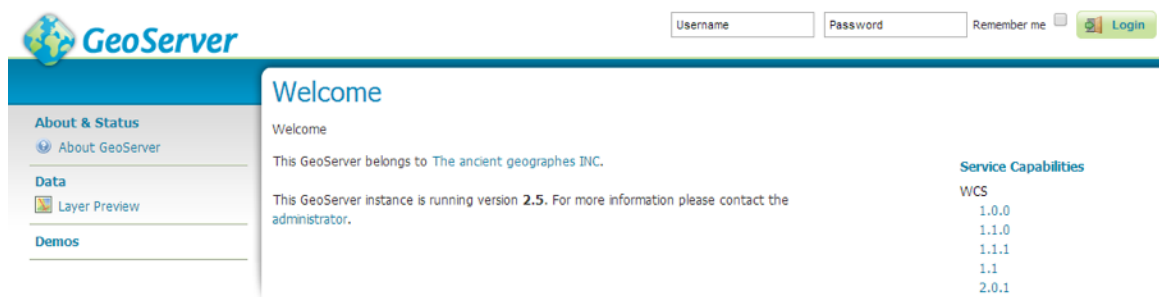
Αρχικά, θα πρέπει να εισέλθουμε στο διαδικτυακό περιβάλλον χρήσης του Geoserver.

Έτσι χρησιμοποιώντας κάποιο web browser επισκεπτόμαστε την διεύθυνση

http://<Server-IP>:8080/geoserver

Στην Θέση του <Server-IP> εισάγουμε την IP διεύθυνση του μηχανήματος στο οποίο έχουμε εγκαταστήσει τον geoserver. Η θύρα 8080 είναι η προεπιλεγμένη θύρα στην οποία ανταποκρίνονται οι εφαρμογές που εκτελούνται με την βοήθεια του Apache Tomcat. Υπάρχει δυνατότητα αλλαγής αυτής της συμπεριφοράς μέσω του web Interface του Apache Tomcat αλλά δεν υπάρχει λόγος να υλοποιηθεί μια τέτοια αλλαγή.

Εάν η εγκατάσταση όλων των τμημάτων του Server ήταν πετυχημένη θα εμφανιστεί η παρακάτω σελίδα όταν θα επισκεφθούμε το διαδικτυακό περιβάλλον χρήσης του Geoserver:



Σε αυτή την σελίδα εισάγουμε τα δεδομένα εισόδου του χρήστη και κάνουμε Login.

Μετά την είσοδο θα παρουσιαστεί η το επόμενο περιβάλλον:

GeoServer

Logged in as nick.

Welcome

Welcome

This GeoServer belongs to The ancient geographies INC.

20 Layers [Add layers](#)

10 Stores [Add stores](#)

7 Workspaces [Create workspaces](#)

Warning: The master password for this server has not been changed from the default. It is **highly** recommended that you change it now. [Change it](#)

Warning: No strong cryptography available, installation of the unrestricted policy jar files is recommended

This GeoServer instance is running version **2.5**. For more information please contact the administrator.

Service Capabilities

WCS	1.0.0
	1.1.0
	1.1.1
	1.1
	2.0.1
WFS	1.0.0
	1.1.0
	2.0.0
WMS	1.1.1
	1.3.0
TMS	1.0.0
WMS-C	1.1.1
WMTS	1.0.0

Σε αυτή την σελίδα επιλέγουμε την επιλογή Stores (που έχουμε συμβολίσει χρησιμοποιώντας ένα μπλε βέλος) από την αριστερή στήλη επιλογών.

Θα εμφανιστεί η σελίδα διαχείρισης των stores. Από αυτή την σελίδα μπορούμε να κάνουμε ρυθμίσεις σχετικά με τις υπηρεσίες που θα προσφέρουν δεδομένα στον Geoserver όπως είναι και η βάση δεδομένων που έχουμε δημιουργήσει.

Επιλέγουμε λοιπόν την επιλογή “Add new Store” όπως εμφανίζεται στην παρακάτω εικόνα.

GeoServer

Logged in as nick.

Stores

Manage the stores providing data to GeoServer

[Add new Store](#)

[Remove selected Stores](#)

Results 1 to 10 (out of 10 items)

☐	Data Type	Workspace	Store Name	Type	Enabled
☐		sf	sfdem	GeoTIFF	✓
☐		nurc	worldImageSample	WorldImage	✓
☐		nurc	arcGridSample	ArcGrid	✓
☐		nurc	mosaic	ImageMosaic	✓
☐		nurc	img_sample2	WorldImage	⚠
☐		sf	sf	Shapefile	✓
☐		tiger	nyc	Shapefile	✓
☐		topp	taz_shapes	Shapefile	✓
☐		topp	states_shapefile	Shapefile	✓
☐		cite	MySQLDB	MySQL	✓

Results 1 to 10 (out of 10 items)

Έπειτα υποδηλώνουμε πως επιθυμούμε να συνδέσουμε μία βάση δεδομένων τύπου MySQL πατώντας την επιλογή MySQL από το πλήθος επιλογών που μας εμφανίζεται.

New data source

Choose the type of data source you wish to configure

Vector Data Sources

-  **Directory of spatial files (shapefiles)** - Takes a directory of shapefiles and exposes it as a data store
-  **MySQL - MySQL Database**
-  **MySQL (JNDI) - MySQL Database (JNDI)**
-  **PostGIS - PostGIS Database**
-  **PostGIS (JNDI) - PostGIS Database (JNDI)**
-  **Properties** - Allows access to Java Property files containing Feature information
-  **Shapefile** - ESRI(tm) Shapefiles (*.shp)
-  **Web Feature Server** - The WFSDataStore represents a connection to a Web Feature Server. This connection provides published by the server, and the ability to perform transactions on the server (when supported / allowed).

Το επόμενο βήμα είναι να ρυθμίσουμε τις παραμέτρους της βάσης δεδομένων μας.

Σε αυτή την σελίδα Θα συμπληρώσουμε τα πεδία:

- **Workspace:** Εδώ δηλώνουμε σε ποιο workspace θα προστεθεί η σύνδεση της βάσης δεδομένων μας. Εμείς επιλέξαμε το workspace cite.
- **host:** Στο πεδίο host συμπληρώνουμε την IP διεύθυνση του μηχανήματος στο οποίο έχουμε εγκαταστήσει την MySQL βάση δεδομένων μας.
- **port:** Στο πεδίο port δηλώνουμε τον αριθμό θύρας στον οποίο αποκρίνεται η βάση δεδομένων μας. Στην περίπτωση μας είναι 3306
- **database:** Το αναγνωριστικό της βάσης δεδομένων η οποία περιέχει τον πίνακα από τον οποίο θέλουμε να αντλήσουμε τα δεδομένα που θα εμφανίζουμε χρησιμοποιώντας τον Geoserver.
- **user:** Το αναγνωριστικό χρήστη της βάσης δεδομένων.
- **passwd:** Ο κωδικός του χρήστη της βάσης δεδομένων.
- **storage engine:** Ο τύπος της μηχανής της βάσης δεδομένων

Μετά την συμπλήρωση των απαραίτητων πεδίων επιλέγουμε την επιλογή save που βρίσκεται στο κάτω μέρος της σελίδας.

Edit Vector Data Source

Edit an existing vector data source

MySQL
MySQL Database

Basic Store Info

Workspace *

cite

Data Source Name *

MySQLDB

Description

☒ Enabled

Connection Parameters

host *

83.212.111.56

port *

3306

database

db1

user *

ezio

passwd

Namespace *

http://www.opengeospatial.net/cite

☒ Expose primary keys

max connections

10

min connections

1

fetch size

1000

Connection timeout

20

☒ validate connections

Primary key metadata table

Session startup SQL

Session close-up SQL

storage engine

MyISAM

Το επόμενο βήμα στην ρύθμιση του μηχανισμού απεικόνισης των μετρήσεων μας, είναι η δημιουργία ενός Layer.

Από την αριστερή στήλη επιλογών επιλέγουμε “Layers”. Εμφανίζεται ένας απλός επιλογέας. Εκεί επιλέγουμε το Store που δημιουργήσαμε νωρίτερα. Στην περίπτωση μας είναι η επιλογή cite:MySQLDB

Στη συνέχεια επιλέγουμε την επιλογή “Configure new SQL view”. Εδώ συμπληρώνουμε το “View name” το οποίο θα είναι το όνομα με το οποίο θα εμφανίζεται το Layer που θα δημιουργήσουμε.

Create new SQL view

Define a new SQL view and configure its identified and geometry columns

View Name

measurementsTakell

SQL statement

```
select * from measurements|
```

SQL view parameters

Guess parameters from SQL Add new parameter Remove selected

☐	Name	Default value	Validation regular expression
☒	Escape special SQL characters		

Attributes

Refresh ☐ Guess geometry type and srid

Name	Type	SRID	Identifier
------	------	------	------------

Save Cancel

Έπειτα συμπληρώνουμε το SQL Statement με το SQL query που θα μας παρέχει τα δεδομένα που θέλουμε να εμφανίσουμε. Στην περίπτωση μας όλα τα δεδομένα.

New Layer

Add a new layer

Add layer from cite:MySQLDB ▼

You can create a new feature type by manually configuring the attribute names and types. [Create new feature type...](#)

On databases you can also create a new feature type by configuring a native SQL statement. [Configure new SQL view...](#)

Here is a list of resources contained in the store 'MySQLDB'. Click on the layer you wish to configure

<<

<

1

>

>>

Results 0 to 0 (out of 0 items)

🔍

Search

Published	Layer name	Action
	alpha	Publish
	geom	Publish

<<

<

1

>

>>

Results 0 to 0 (out of 0 items)

Αφού επιλέξουμε save μεταφερόμαστε στην σελίδα “Edit Layer”.

Edit Layer

Edit layer data and publishing

cite:measurements

Configure the resource and publishing information for the current layer

Data

Publishing

Dimensions

Tile Caching

Basic Resource Info

Name

measurements

☒ Enabled

☒ Advertised

Title

measurements

Abstract

Keywords

Current Keywords

measurements

features

Remove selected

New Keyword

Vocabulary

Add Keyword

Metadata links

No metadata links so far

Add link

Note only FGDC and TC211 metadata links show up in WMS 1.1.1 capabilities

Coordinate Reference Systems

Native SRS

EPSG:4326

EPSG:WGS 84...

Declared SRS

EPSG:4326

Find...

EPSG:WGS 84...

SRS handling

Force declared

Bounding Boxes

Native Bounding Box

Min X	Min Y	Max X	Max Y
23.7008418	37.9622512	23.7473073	37.9664814

Compute from data

Lat/Lon Bounding Box

Min X	Min Y	Max X	Max Y
23.7008418	37.9622512	23.7473073	37.9664814

Compute from native bounds

Feature Type Details

Property	Type	Nullable	Min/Max Occurrences
Downlink	Float	false	1/1
ID	Integer	false	1/1
latitude	Float	false	1/1
longitude	Float	false	1/1
Measurement_Time	Timestamp	false	1/1
Network_Type	String	false	1/1
os	String	false	1/1
Ping1_IPERF	Float	true	0/1
Ping2_DN	Float	true	0/1
Position	Geometry	false	1/1
Uplink	Float	false	1/1
MAX_DL	Double	true	0/1
MAX_UL	Double	true	0/1

Edit sql view

Save

Cancel

Στη σελίδα “Edit Layer” πρέπει να καθορίσουμε τις ακόλουθες παραμέτρους:

- Native SRS: Το SRS (Spatial reference system) είναι το σύστημα συντεταγμένων που θα ακολουθηθεί για την παρουσίαση των γεωγραφικών δεδομένων. Εξαρτάται από την πλατφόρμα που μας παρέχει τη γεωγραφική τοποθεσία. Στην περίπτωση μας χρησιμοποιούμε το EPSG:4326.
- Native Bounding Box: Το Native Bounding Box προσδιορίζει την περιοχή στην οποία θα εμφανίζονται τα δεδομένα μας. Ένας εύκολος τρόπος για την συμπλήρωση αυτού του πεδίου είναι η επιλογή “Compute from data” που θα εξετάσει τις συντεταγμένες που είναι αποθηκευμένες στην βάση δεδομένων μας και θα υπολογίσει αυτόματα τον χώρο. Επίσης μπορεί να λειτουργήσει και σαν εργαλείο για να ανιχνεύσουμε τυχόν λάθη που έγιναν κατά την διάρκεια της παραμετροποίησης το Geoserver Store η της βάσης δεδομένων μας. Εάν υπάρχει κάποιο πρόβλημα θα εμφανιστεί προειδοποιητικό μήνυμα η τα δεδομένα δεν θα αλλάζουν.
- Lat/Lon Bounding Box: Θα είναι παρόμοιο με το Native Bounding Box.

Styles

Manage the Styles published by GeoServer

 Add a new style

 Removed selected style(s)

Επόμενο βήμα είναι η δημιουργία ενός Style.

Για να δημιουργήσουμε ένα καινούργιο Style επιλέγουμε “Styles” από την αριστερή στήλη επιλογών και μετά επιλέγουμε “Add a new style”

Στην επόμενη σελίδα συμπληρώνουμε το όνομα με το οποίο θα αναφέρουμε το καινούργιο Style και επιλέγουμε και το workspace στο οποίο θα αποθηκευτεί και από το οποίο θα αντλεί δεδομένα το Style.

Τέλος στο τελευταίο πεδίο γράφουμε την περιγραφή εμφάνισης του Style.

Η περιγραφή γίνεται χρησιμοποιώντας SLD (Styled Layer Descriptor).

Το SLD είναι ένα XML schema που ορίζετε από τον οργανισμό Open Geospatial Consortium, για την περιγραφή της εμφάνισης layers σε χάρτες.

Παρακάτω εμφανίζεται ο σελίδα δημιουργίας ενός style που χρησιμοποιούμε για την απεικόνιση ταχύτητας λήψης δεδομένων.

Είναι προφανές πως αυτός ο οδηγός προσφέρει απλές οδηγίες για την βασική παραμετροποίηση του Geoserver. Διαφορετικές εργασίες απαιτούν διαφορετικά δεδομένα

και τρόπους απεικόνισης. Γι' αυτό τον λόγο θα είναι απαραίτητη η περαιτέρω παραμετροποίηση του Geoserver ώστε οι απεικόνιση των δεδομένων να είναι η κατάλληλη για κάθε διαφορετικό είδος δεδομένων.

Style Editor

Edit the current SLD style. The editor can provide syntax highlight and be brought to full screen. Click on the "validate" button to verify the style is a valid SLD document.

Name

Workspace

Copy from existing style
 [Copy ...](#)

 12pt

```
13 <Title>Measurement Point</Title>
14 <Abstract>A sample style that draws a point</Abstract>
15 <!-- FeatureTypeStyles describe how to render different features -->
16 <!-- A FeatureTypeStyle for rendering points -->
17 <FeatureTypeStyle>
18 <!-- <Rule>
19 <Name>LARGE</Name>
20 <Title>large</Title>
21 <PointSymbolizer>
22 <Graphic>
23 <Mark>
24 <WellKnownName>circle</WellKnownName>
25 <Fill>
26 <CssParameter name="fill">#FFFF00</CssParameter>
27 </Fill>
28 </Mark>
29 <Size>8</Size>
30 </Graphic>
31 </PointSymbolizer>
32 </Rule>-->
33 <Rule>
34 <Name>RED</Name>
35 <Title>Greater than 32768</Title>
36 <ogc:Filter>
37 <ogc:PropertyIsGreaterThanOrEqualTo>
38 <ogc:PropertyName>Downlink</ogc:PropertyName>
39 <ogc:Literal>32768</ogc:Literal>
40 </ogc:PropertyIsGreaterThanOrEqualTo>
41 </ogc:Filter>
42
```

SLD file
 Δεν επιλέχθηκε κανένα αρχείο. [Upload ...](#)

3.7 Παρουσίαση και εξήγηση της γλώσσας SLD

Το Styled Layer Descriptor (SLD) αποτελεί έναν βασισμένο στο XML (extensive Markup Language) τρόπο οπτικής περιγραφής γεωγραφικών δεδομένων που εκδόθηκε από το Open Geospatial Consortium (OGC). Χρησιμοποιήθηκε για να περιγράψει στον Geoserver τον τρόπο με τον οποίο θα εμφανίζοντα οι τιμές των μετρήσεων που είναι αποθηκευμένες στην βάση δεδομένων. Με την SLD περιγραφή στον Geoserver οπτικοποιούμε τα

δεδομένα μας και τα απεικονίζουμε τόσο στις εφαρμογές μας, όσο και στην σελίδα που δημιουργήσαμε. Πιο συγκεκριμένα, έχουμε:

NamedLayer: Αποτελεί την βασική δομή ενός SLD και ενθυλακώνει κανόνες.

Rules: Κανόνες που εκφράζουν μία συνθήκη για να αναπαραστήσουν μια υποκατηγορία των δεδομένων με ένα συγκεκριμένο τρόπο.

Μέσα στους κανόνες, διαθέτουμε τα εξής πεδία:

- Name: Όνομα του κανόνα.
- Title: Τίτλος του κανόνα.
- Filter: Πεδίο που περιέχει συνθήκες, μέσω των οποίων επιλέγονται τα δεδομένα που θα αναπαρασταθούν με τον τρόπο τον οποίο δηλώνει ο κανόνας. Πιο συγκεκριμένα έχουμε συνθήκες, όπως η ιδιότητα είναι μεγαλύτερη ή ίση (ogc: PropertyIsGreaterThanOrEqualTo) μιας τιμής. Μία ιδιότητα (ogc:Property) που προσφέρεται από το εκάστοτε store. Μία τιμή (ogc:Literal)
- Point Symbolizer: Ρυθμίζεις σχετικά με την απεικόνιση του συγκεκριμένου κανόνα.

Ένα παράδειγμα ενός SLD κώδικα που χρησιμοποιήσαμε για να αναπαραστήσουμε τις μετρήσεις ταχύτητας λήψης. Παρουσιάζουμε μόνο τον πρώτο κανόνα ο οποίος περιγράφει την δημιουργία ενός κύκλου μεγέθους 16, χρώμα κόκκινο, διαφάνεια 0.5 και θα εμφανίζεται για τιμές μεγαλύτερες από 32768 kbps.

Υπάρχουν πολλοί παρόμοιοι κανόνες για κάθε Style.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<StyledLayerDescriptor version="1.0.0"
  xsi:schemaLocation="http://www.opengis.net/sld
StyledLayerDescriptor.xsd"
  xmlns="http://www.opengis.net/sld"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <!-- a Named Layer is the basic building block of an SLD
document -->
  <NamedLayer>
    <Name>Speed Testing Platform Android/IOS</Name>
    <UserStyle>
      <!-- Styles can have names, titles and abstracts -->
      <Title>Measurement Point</Title>
      <Abstract>A sample style that draws a point</Abstract>
      <Rule>
        <Name>RED</Name>
        <Title>Greater than 32768</Title>
```

```

<ogc:Filter>
  <ogc:PropertyIsGreaterThanOrEqualTo>
    <ogc:PropertyName>Downlink</ogc:PropertyName>
    <ogc:Literal>32768</ogc:Literal>
  </ogc:PropertyIsGreaterThanOrEqualTo>
</ogc:Filter>

<PointSymbolizer>
  <Graphic>
    <Mark>
      <WellKnownName>circle</WellKnownName>
      <Fill>
        <CssParameter name="fill">#FF0000</CssParameter>

        <CssParameter name="fill-opacity">0.5</CssParameter>
      </Fill>
    </Mark>
    <Size>16</Size>
  </Graphic>
</PointSymbolizer>

```

3.8 Απεικόνιση των αποτελεσμάτων στο διαδίκτυο

Χρησιμοποιώντας τις ικανότητες που μας παρέχει ο Geoserver για εμφάνιση των αποτελεσμάτων σε χάρτη, δημιουργήθηκε μία απλή σελίδα η οποία παρέχει πρόσβαση στα αποτελέσματα των μετρήσεων μέσω web. Τα αποτελέσματα απεικονίζονται με τον ίδιο μηχανισμό με τον οποίο εμφανίζονται και στην εφαρμογή, δηλαδή μέσω της χρήσης WMS. Όμως στην σελίδα ο χρήστης έχει επιπλέον την δυνατότητα να επιλέξει και από μία μεγάλη επιλογή χαρτών που συμπεριλαμβάνει χάρτες όπως, Google Maps και Open Maps. Η δυνατότητα για προβολή όλων των χαρτών μαζί με τα αποτελέσματα των μετρήσεων μας παρέχεται μέσω της ιδιότητας του Geoserver να σερβίρει δεδομένα και στο web.

Η σελίδα αναπτύχθηκε κυρίως για δοκιμαστικούς σκοπούς και για να μπορούμε να προβάλουμε εύκολα τα αποτελέσματα των μετρήσεων μας κατά την διάρκεια της ανάπτυξης της εργασίας. Η σελίδα βρίσκεται προσωρινά στην σελίδα: <http://83.212.111.56/mapsDisplay.wms> όπου υποστηρίζεται και εκτελείται στον ίδιο εξυπηρέτη όπως και η διαδικτυακή μας πλατφόρμα.

4. Υλοποίηση εφαρμογής για συσκευές iOS

4.1 Προετοιμασία του περιβάλλοντος ανάπτυξης

Το πρώτο βήμα προτού ξεκινήσει η υλοποίηση της εφαρμογής είναι η λήψη της πιο πρόσφατης έκδοσης του Xcode. Το Xcode διατίθεται μέσω του Mac App store και είναι συμβατός με υπολογιστές που εκτελούν το λειτουργικό σύστημα Mac OS X έκδοση 10.8.4 η πιο σύγχρονη. Για να γίνει λήψη του xcode απαιτείται λογαριασμός Apple ID που μπορεί να δημιουργηθεί κατά την διαδικασία λήψης του Xcode και μπορεί να χρησιμοποιηθεί και αργότερα στην δήλωση προγραμματιστή για το πρόγραμμα iOS.

Το xcode μπορεί να βρεθεί σε αυτή την τοποθεσία: <https://itunes.apple.com/gr/app/xcode/id497799835?mt=12> (6/2014)

Μετά την λήψη και την εγκατάσταση του Xcode, πρέπει να εκτελέσουμε το πρόγραμμα και να μεταβούμε στην επιλογή preferences. Εκεί επιλέγουμε Downloads και επιλέγουμε να εγκατασταθούν ο iOS 7 simulator και το iOS 7.1 doc set. Έπειτα προχωράμε στην δήλωση του προγραμματιστή.

Για να μπορεί ο προγραμματιστής να εκτελέσει την εφαρμογή του σε συσκευή με iOS 7 θα πρέπει να είναι μέλος του iOS Developer Program. Υπάρχουν δύο βασικές κατηγορίες μελών, τα ατομικά μέλη που είναι άτομα τα οποία έχουν πληρώσει το αντίτιμο που απαιτείται από την Apple για συμμετοχή στο πρόγραμμα και τα μέλη ομάδων. Οι ομάδες μπορούν να είναι εταιρίες που διατηρούν επαγγελματικό λογαριασμό που συμπεριλαμβάνει τους υπαλλήλους τους και πληρώνουν με βάση ειδικές συμφωνίες. Επίσης μπορεί να είναι και εκπαιδευτικοί οργανισμοί οι οποίοι έχουν το δικαίωμα να διατηρούν ομάδα στο iOS Developer Program και να συμπεριλαμβάνουν τους φοιτητές τους δωρεάν. Αξίζει να σημειωθεί πως όλες οι κατηγορίες μελών εκτός από τους εκπαιδευτικούς οργανισμούς έχουν δικαίωμα να εκδώσουν την εφαρμογή τους στο iOS App Store. Για την ανάπτυξη της εφαρμογής της εργασίας χρησιμοποιήθηκε το εκπαιδευτικό πρόγραμμα μελών του iOS Developer Program στο οποίο προστέθηκα σαν απλό μέλος. Διαχειριστής του λογαριασμού ήταν ο Δρ. Βασίλειος Δαλάκας.

Παρακάτω θα περιγράψω την διαδικασία που πρέπει να εκτελεστεί ώστε να μπορεί ο προγραμματιστής να εκτελέσει την εφαρμογή που αναπτύσσει στην συσκευή του. Περιγράφεται η διαδικασία που εκτελέστηκε για αυτή την εργασία και που πρέπει να εκτελεστεί και για μέλη του εκπαιδευτικού προγράμματος. Η διαδικασία είναι διαφορετική για ατομικά μέλη και επαγγελματίες.

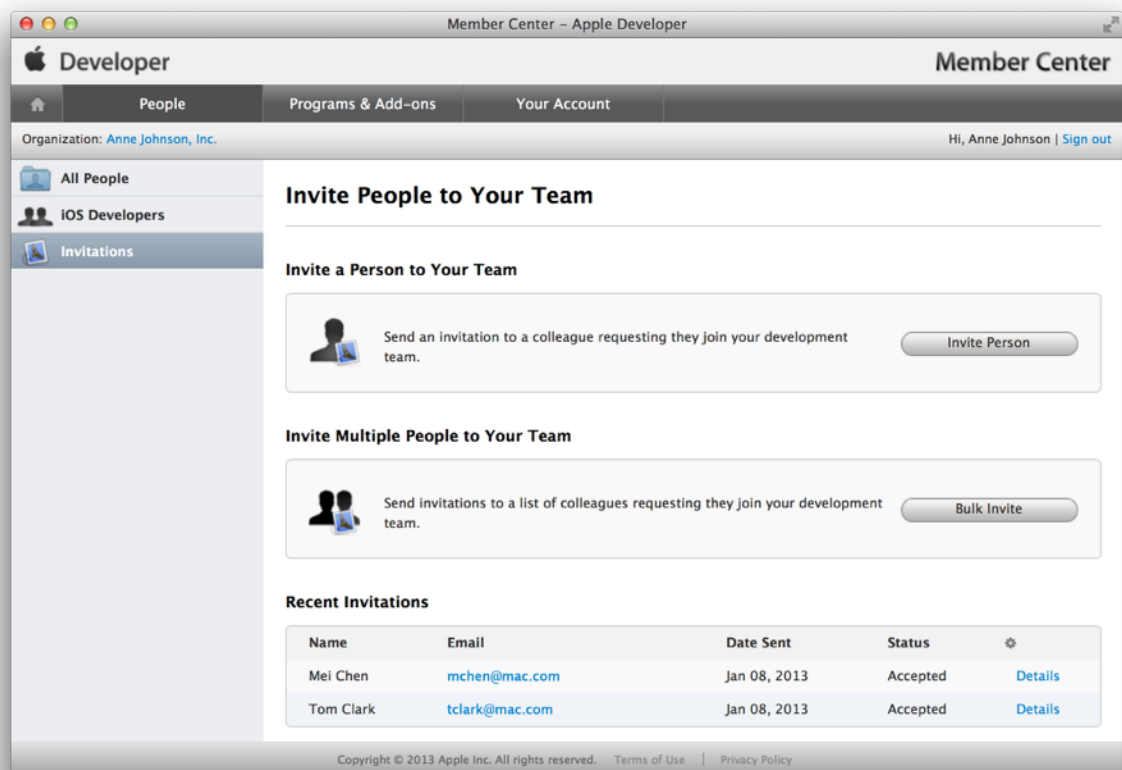
Αρχικά θα πρέπει κάθε ενδιαφερόμενος που θα συμπεριληφθεί στο πρόγραμμα να έχει ένα λογαριασμό Apple ID. Ο λογαριασμός μπορεί να δημιουργηθεί ανά πάσα στιγμή και είναι δωρεάν.

Θα πρέπει ο προγραμματιστής να λάβει πρόσκληση για συμμετοχή στην ομάδα από κάποιον διαχειριστή του ομαδικού λογαριασμού. Υπάρχουν τρεις κατηγορίες χρηστών σε ομαδικούς λογαριασμούς. Ο Agent ο οποίος έχει όλα τα δικαιώματα διαχείρισης του λογαριασμού αλλά επιπλέον έχει και όλες τις νομικές ευθύνες του λογαριασμού. Επίσης υπάρχουν οι Admin που έχουν βασικά δικαιώματα διαχείρισης του λογαριασμού και μπορούν να προσθέσουν νέους χρήστες και να προωθήσουν απλούς χρήστες σε Admin.

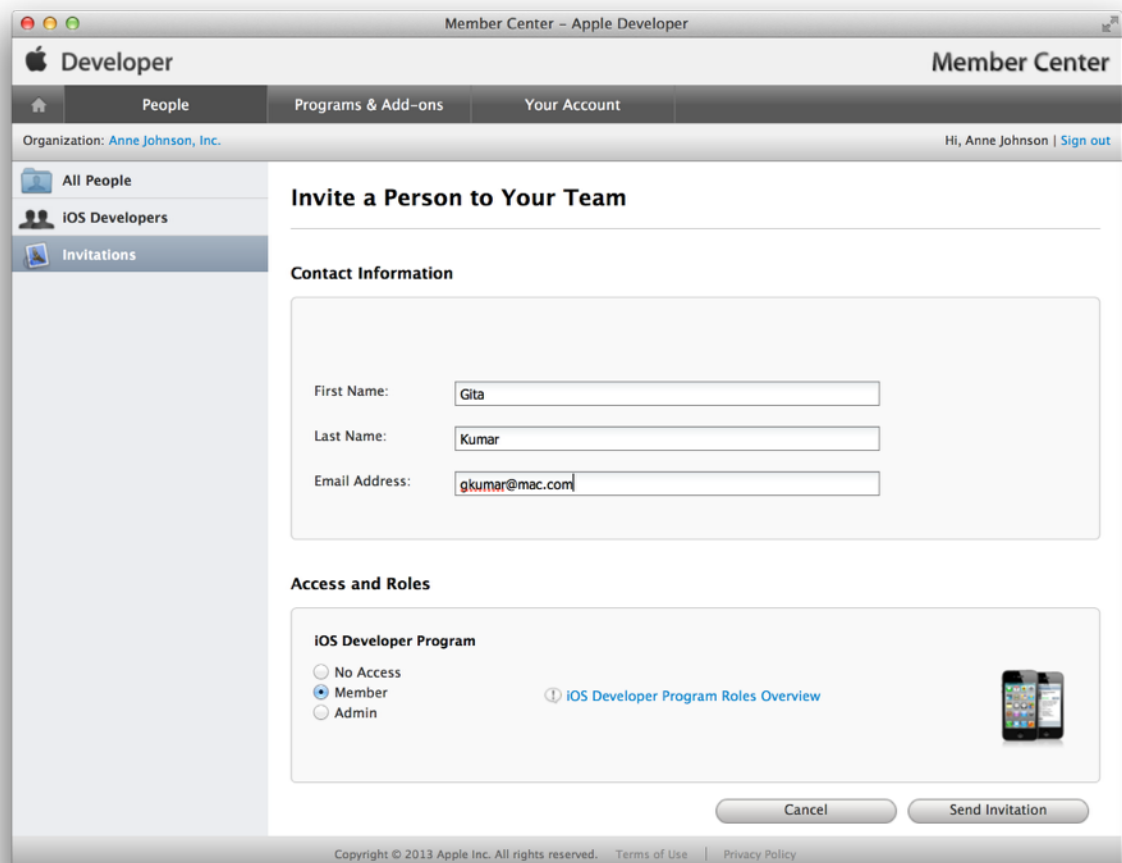
Τέλος υπάρχουν οι απλοί χρήστες που δεν έχουν καθόλου δικαιώματα διαχείρισης του λογαριασμού. Επίσης δεν έχουν ούτε δικαιώματα έκδοσης ψηφιακών πιστοποιητικών, provisioning profiles και δεν μπορούν να δηλώσουν συσκευές ως συσκευές ανάπτυξης. Ο οδηγός θα παρουσιάσει βήματα κυρίως από την πλευρά των χρηστών αλλά παρέχονται και οδηγίες για τους Admin επειδή ουσιαστικά οι Admin πρέπει να εγκρίνουν την κάθε διαδικασία.

Για να σταλεί η πρόσκληση σε ένα νέο μέλος της ομάδας θα πρέπει κάποιος Admin να εισέλθει στην σελίδα διαχείρισης του λογαριασμού (<https://developer.apple.com/membercenter/index.action>) και να επιλέξει την επιλογή “People”.

Σε αυτή την σελίδα ο Admin επιλέγει “Invite Person”, σε περίπτωση πρόσκλησης πολλών ατόμων μπορεί να επιλέξει και “Bulk Invite” αλλά τα βήματα ακολουθούν την διαδικασία προσθήκης ενός ατόμου στην ομάδα. Αφότου επιλέξει ο Admin “Invite Person” θα του εμφανιστεί μία σελίδα στην οποία θα συμπληρωθούν τα στοιχεία του χρήστη που θέλει να προσθέσει στην ομάδα.

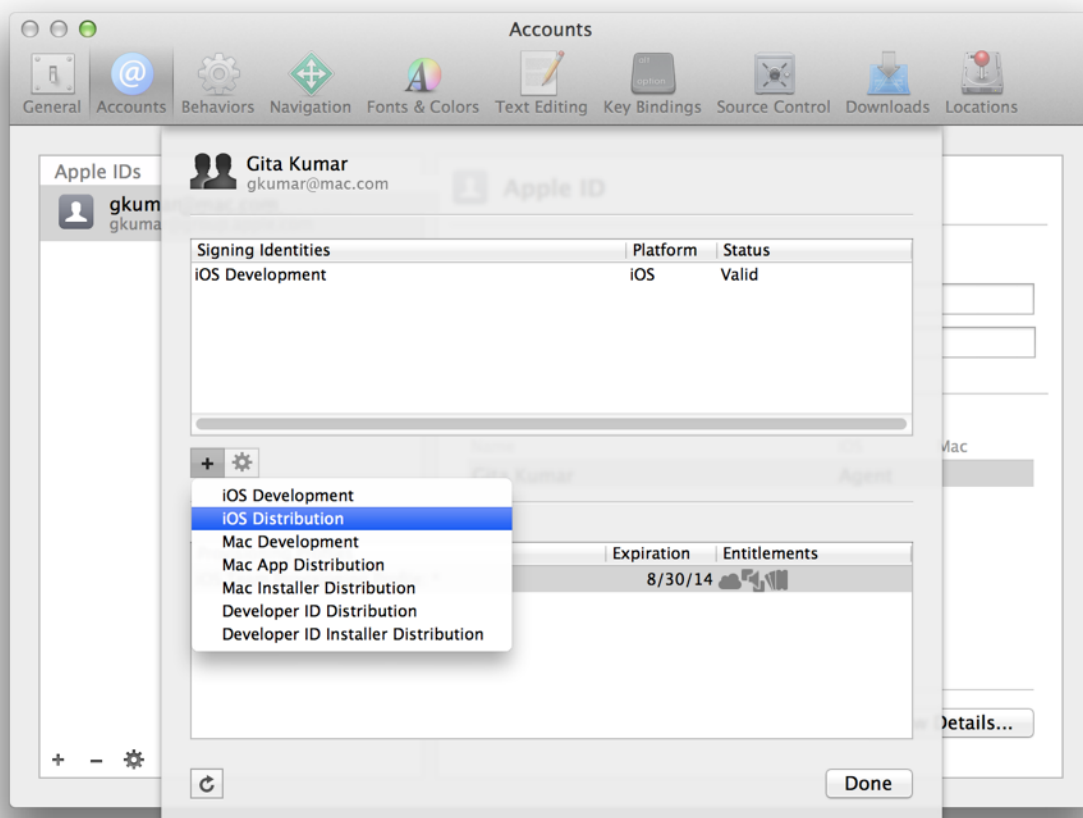


Επίσης σε αυτή την σελίδα ο Admin επιλέγει και τον ρόλο του καινούργιου μέλους, στην περίπτωση μας είναι απλός χρήστης. Υπάρχει η δυνατότητα αλλαγής του ρόλου χρήστη μετά την πρόσκληση αλλά η διαδικασία δεν θα καλυφθεί σε αυτές τις οδηγίες.



Μετά την πρόσκληση, ο νέος χρήστης πρέπει να αποδεχθεί και να δηλώσει τον λογαριασμό και την ομάδα του στην εφαρμογή Xcode. Μετά το άνοιγμα της εφαρμογής ο χρήστης πρέπει να επιλέξει preferences από το μενού του Xcode και στην σελίδα Accounts να προσθέσει τα στοιχεία του και να επιλέξει την ομάδα του πανεπιστημίου του. Έπειτα ο χρήστης πρέπει να ζητήσει την έκδοση ψηφιακού πιστοποιητικού για λόγους δοκιμής από τον Admin. Από την σελίδα Account του Xcode ο χρήστης επιλέγει την ομάδα και μετά την επιλογή “View Details” στο κάτω μέρος του παραθύρου.

Στο νέο παράθυρο που εμφανίζεται πατάει στο σύμβολο “+” και επιλέγει iOS Development. Θα εμφανιστεί ένα παράθυρο διαλόγου που πληροφορεί τον χρήστη πως πρέπει να παρθεί άδεια για την έκδοση του πιστοποιητικού από κάποιο Admin και δίνει την επιλογή να επιλέξει ο χρήστης να αποστείλει αίτηση έκδοσης του ψηφιακού πιστοποιητικού ή ακύρωση της διαδικασίας. Η διαδικασία επιλογής πιστοποιητικού εμφανίζεται παρακάτω.



Μετά την ολοκλήρωση αυτών των βημάτων από τον χρήστη πρέπει κάποιος Admin να εισέλθει στην σελίδα διαχείρισης και να επιλέξει “Certificates, Identities & Profiles”.

Έπειτα επιλέγει Certificates και μετά την υποκατηγορία pending από τα αριστερά της οθόνης. Μετά επιλέγει το πιστοποιητικό του ατόμου που επιθυμεί να εγκρίνει και πατάει Approve.



Τα πιστοποιητικά αυτά είναι προσωπικά και θα πρέπει κάθε ενεργό μέλος της ομάδας ,που θέλει να αναπτύξει κάποιου είδους εφαρμογή, να έχει ένα προσωπικό πιστοποιητικό.

Το πιστοποιητικό μετά μπορεί να ληφθεί από τον χρήστη που το ζήτησε από την σελίδα διαχείρισης στην επιλογή “Certificates, Identities & Profiles”. Εκεί στο ίδιο περιβάλλον με του Admin μπορεί να επιλέξει το πιστοποιητικό του και να κάνει λήψη. Επίσης δίνεται η επιλογή για ανάκληση (revoke) του πιστοποιητικού από τον χρήστη που το ζήτησε και από τον Admin εάν υπάρχει υποψία κακοδιαχείρισης του πιστοποιητικού.



Μετά την λήψη του πιστοποιητικού από τον χρήστη χρειάζεται η εγκατάσταση του στο προσωπικό μηχάνημα του χρήστη. Αυτό γίνεται με ένα διπλό-κλικ πάνω στο αρχείο του πιστοποιητικού. Το πιστοποιητικό προστίθεται στο “keychain” του υπολογιστή mac και μπορεί να διαγραφεί από τον υπολογιστή χρησιμοποιώντας την εφαρμογή keychain access.app που βρίσκεται στον κατάλογο Applications/Utilities σε κάθε mac υπολογιστή. (απαιτούνται δικαιώματα root στον υπολογιστή για κάθε αλλαγή που γίνεται σε πιστοποιητικά).

Το επόμενο βήμα είναι η δήλωση της συσκευής του χρήστη ως συσκευή ανάπτυξης. Για να δηλωθεί η συσκευή του χρήστη ως συσκευή ανάπτυξης έτσι ώστε να μπορούν να εκτελεστούν εφαρμογές με την συσκευή πρέπει να ληφθεί το αναγνωριστικό UDID της συσκευής. Το αναγνωριστικό μπορεί να ληφθεί από την επιλογή Organiser του Xcode. Επιλέγοντας devices και συνδέοντας την συσκευή που θέλουμε να δηλώσουμε με τον υπολογιστή. Η συσκευή θα εμφανιστεί σαν επιλογή στο αριστερό τμήμα του παραθύρου και επιλέγοντας την θα εμφανιστεί μία εικόνα της συσκευής καθώς και τα στοιχεία της. Για να εμφανιστούν όμως στοιχεία όπως το UDID θα πρέπει να επιλεγεί η επιλογή “Trust this device” από την συσκευή όταν θα εμφανιστεί ένα μήνυμα διαλόγου που θα ερωτά τον χρήστη του κινητού εάν εμπιστεύεται την συσκευή στην οποία συνδέθηκε το κινητό.

Εάν ο χρήστης αποδεχθεί τον υπολογιστή ως έμπιστο θα εμφανιστεί η επιλογή Identifier στο παράθυρο Organizer του Xcode κάτω από άλλα στοιχεία σχετικά με το κινητό τηλέφωνο.



Gita Kumar's iPhone

Capacity 14.7 GB

Model iPhone 4

Serial Number 79109MDKA4S

ECID 2611089354229

Identifier 2be702beae2ac34fc0d7f8ae2b5b808a402fc01a

Software Version 6.1.3 (10B329)

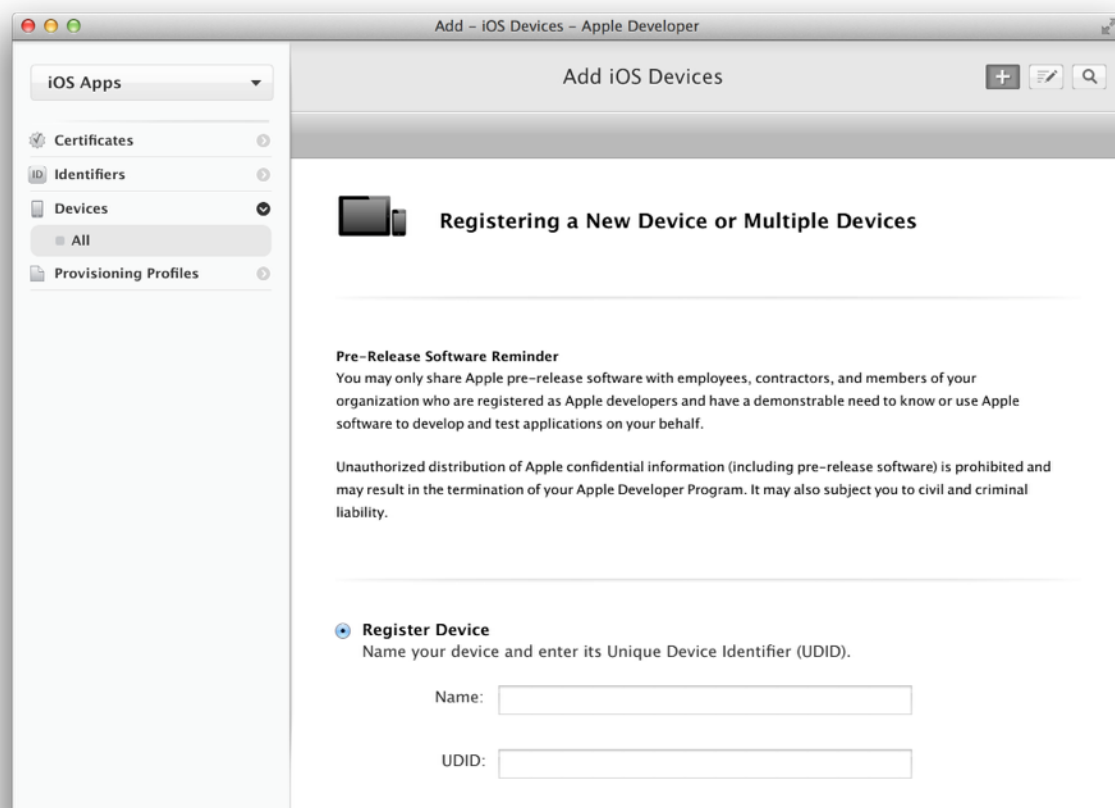
[Restore using iTunes...](#)

Provisioning No provisioning profiles ➕

Applications No developed applications ➕
No FairPlay-encrypted applications

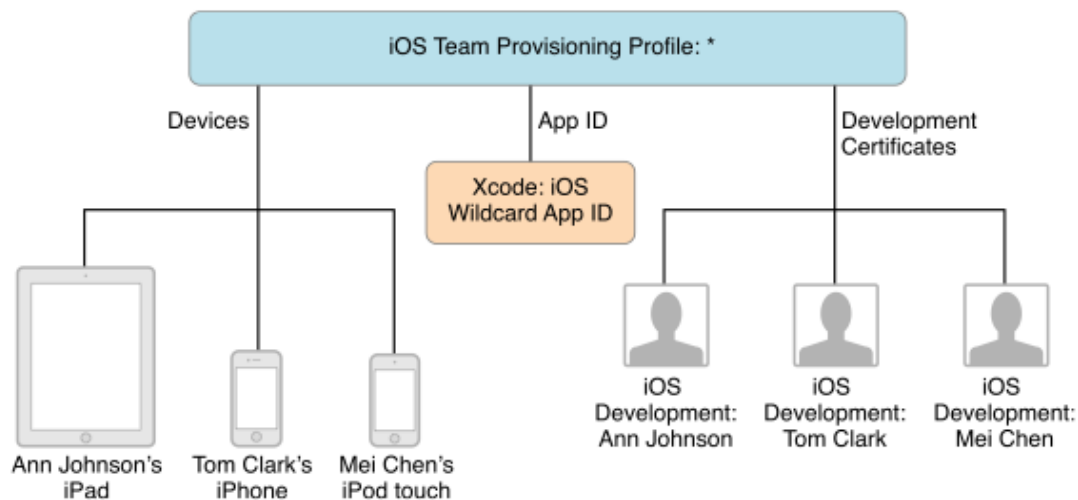
Για να δηλωθεί η συσκευή θα πρέπει να αντιγραφεί ο Identifier και να δοθεί σε κάποιον Admin για να προστεθεί στην λίστα συσκευών. Υπάρχει τρόπος προσθήκης πολλών συσκευών ταυτόχρονα για το επόμενο βήμα αλλά παρουσιάζεται η διαδικασία που ακολουθήθηκε για την προσθήκη μίας συσκευής για αυτή την εργασία.

Ο Admin από την σελίδα “Certificates, Identities & Profiles” του κέντρου διαχείρισης επιλέγει Devices και πατάει το “+” που βρίσκεται στο πάνω δεξιά μέρος της σελίδας. Εκεί θα του παρουσιαστεί μία σελίδα για εισαγωγή ενός αναγνωριστικού για την συσκευή και για το UDID της συσκευής, το οποίο αποκτήθηκε με την παραπάνω διαδικασία.



Το τελικό βήμα είναι η έκδοση και εγκατάσταση στη συσκευή ενός provisioning profile. Το provisioning profile είναι ένα σύνολο οδηγιών που επιτρέπει στην συσκευή να εκτελέσει εφαρμογές που βρίσκονται σε δοκιμαστικό στάδιο. Περιέχει πληροφορίες σχετικά με τις δηλωμένες συσκευές στις οποίες θα λειτουργεί και τα ψηφιακά πιστοποιητικά των προγραμματιστών που μπορούν να εκδίδουν εφαρμογές για δοκιμή. Ακόμη περιέχει και πληροφορίες σχετικά με το αναγνωριστικό της εφαρμογής που θα εκτελείται και ένα μοναδικό αναγνωριστικό σχετικό με την ομάδα που το εκδίδει. Ένα

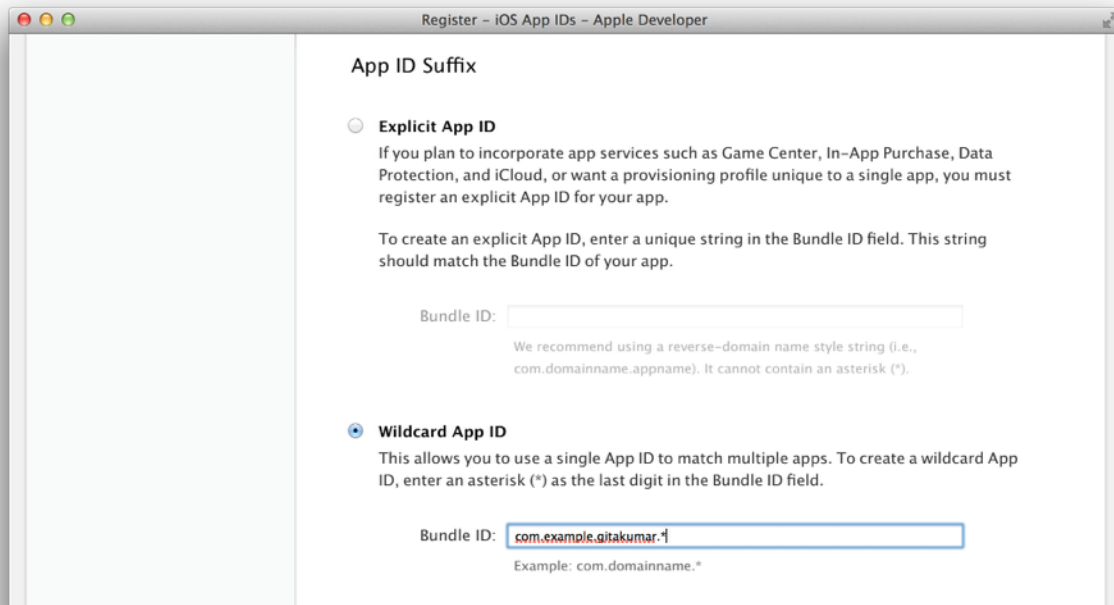
παράδειγμα της δομής και των δεδομένων που περιέχει ένα provisioning profile όπως το παρουσιάζει η Apple εμφανίζεται παρακάτω.



Πρίν την διαδικασία έκδοσης του provisioning profile πρέπει να ελέγξουμε εάν η ομάδα περιέχει κάποια ταυτότητα εφαρμογών (iOS App ID). Τα App ID μπορούν να προβληθούν από την σελίδα “Certificates, Identities & Profiles” στην επιλογή Identities. Εάν δεν υπάρχει κάποιο Identity ο Admin μπορεί να προσθέσει ένα πατώντας το σύμβολο “+” στο πάνω δεξιά μέρος της σελίδας. Έπειτα θα του ζητηθεί ένα όνομα για το App ID

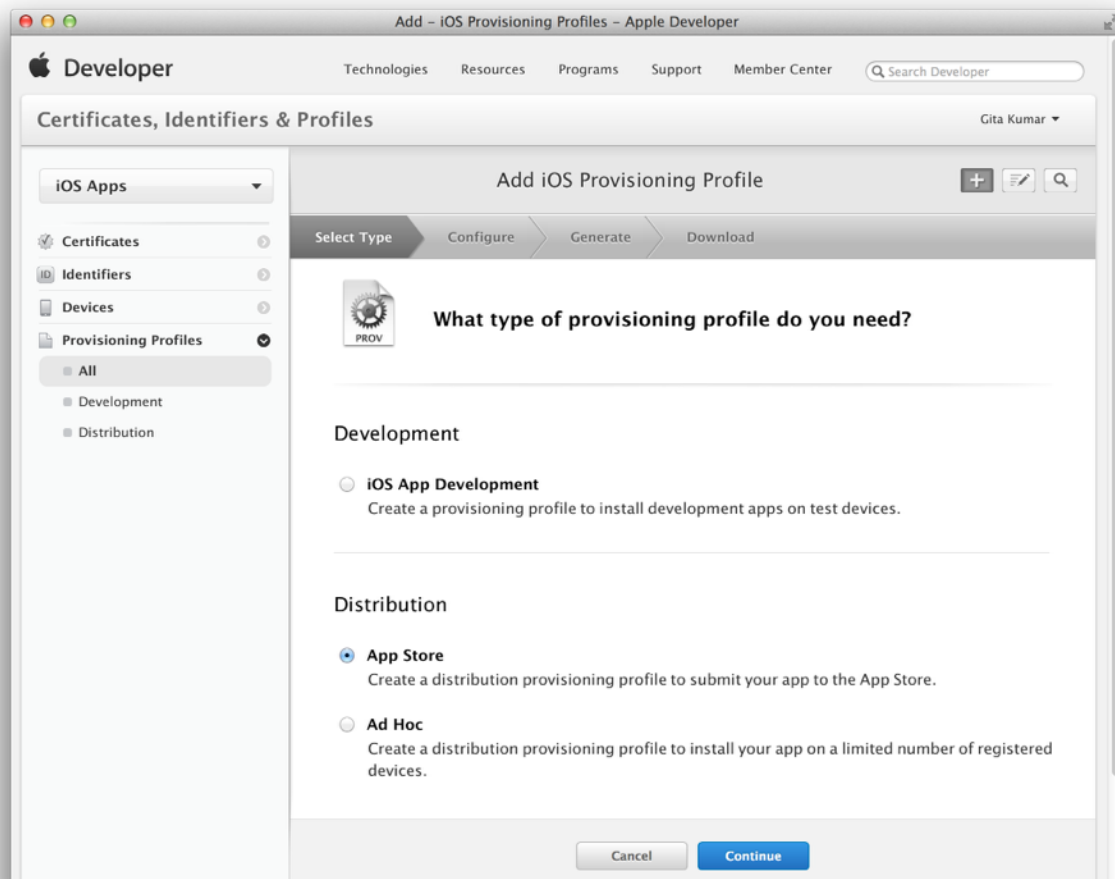


Μετά θα του ζητηθεί να επιλέξει εάν θέλει να δηλώσει ένα συγκεκριμένο όνομα εφαρμογής ή ένα γενικό App ID με χρήση wildcard. Για αυτή την εφαρμογή δημιουργήθηκε ένα γενικό App ID. Τα ονόματα των App ID έχουν την μορφή αντίστροφης url που περιγράφουν από το πιο γενικό προς το πιο ειδικό με τελευταίο πεδίο το όνομα της εφαρμογής. Ένα παράδειγμα wildcard app ID είναι το: gr.hua.dit.*

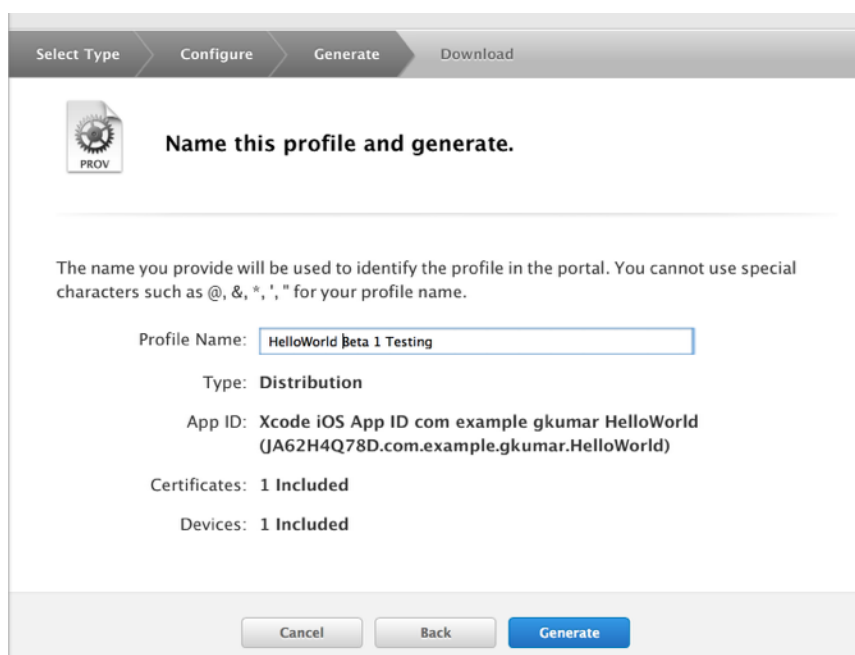


Το τελικό βήμα στην δημιουργία ενός App ID είναι η δήλωση των υπηρεσιών που θα χρησιμοποιεί. Στην εργασία μας δεν χρησιμοποιούμε καμία από τις παρεχόμενες υπηρεσίες.

Μετά την δημιουργία του App ID ο Admin μεταβαίνει στην επιλογή Provisioning Profiles και πατάει το κουμπί “+” στο πάνω δεξιά μέρος της οθόνης για να ξεκινήσει την διαδικασία δημιουργίας ενός Provisioning Profile. Είναι απαραίτητο να έχουν πρώτα δηλωθεί όλες οι συσκευές στις οποίες θα εγκατασταθεί το Provisioning Profile και όλοι οι προγραμματιστές οι οποίοι θα αναπτύξουν εφαρμογές για δοκιμή.



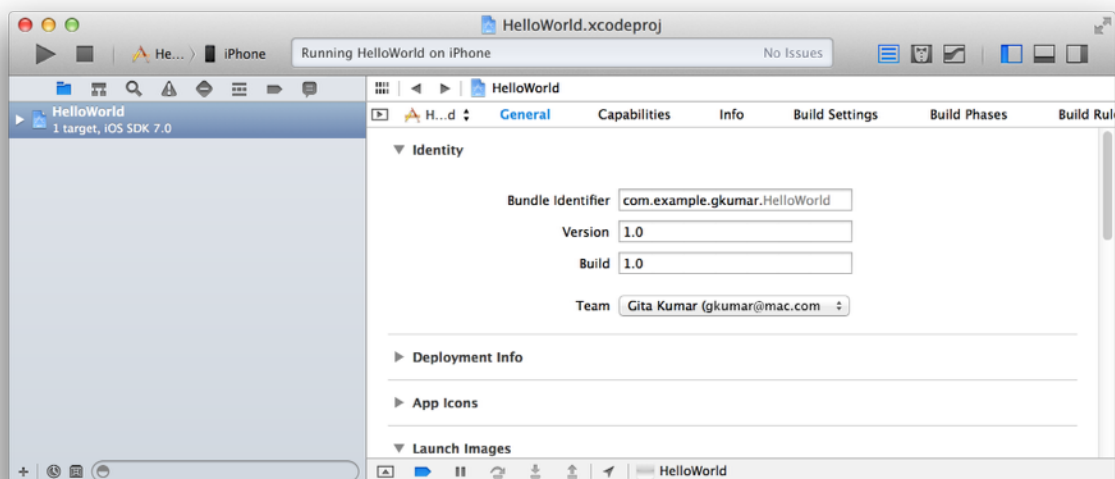
Στην παραπάνω οθόνη επιλέγουμε iOS App Development και μετά την επιλογή Continue. Έπειτα επιλέγουμε το App ID για το οποίο θα εκδοθεί αυτό το Provisioning Profile. Επίσης επιλέγουμε τις συσκευές και τα πιστοποιητικά για τα οποία θα λειτουργεί αυτό το Profile. Πριν το τελικό βήμα δίνουμε ένα αναγνωριστικό όνομα στο Profile και πατάμε Generate.



Μετά την δημιουργία του Provisioning Profile οι χρήστες μπορούν να το κατεβάσουν από τους δικούς τους λογαριασμούς πηγαίνοντας στην σελίδα “Certificates, Identities & Profiles” και επιλέγοντας το Profile που επιθυμούν.

Μετά την λήψη του Profile οι χρήστες πρέπει να το φορτώσουν στις συσκευές τους με την χρήση του Xcode Organizer. Έχοντας συνδεδεμένη την συσκευή όπως σε προηγούμενο βήμα επιλέγουν από το αριστερά μέρος του παραθύρου την επιλογή Provisioning Profiles. Σε αυτό το παράθυρο επιλέγουμε το μεγάλο κουμπί “Add” στο κάτω μέρος της οθόνης και επιλέγουμε το αρχείο που κάναμε λήψη παραπάνω. Εάν όλα έχουν γίνει σωστά θα εμφανιστεί το μήνυμα Valid Profile δίπλα στα στοιχεία του πιστοποιητικού.

Τώρα κατά την διάρκεια δημιουργίας ενός καινούργιου project με το Xcode φροντίζουμε να βάλουμε ένα όνομα που θα συμφωνεί με το δηλωμένο App ID του Provisioning Profile. Έτσι θα μας δίνεται η δυνατότητα να επιλέξουμε να εκτελέσουμε την εφαρμογή στο κινητό μας τηλέφωνο. Σε περίπτωση που θέλουμε να εκτελέσουμε ένα υπάρχον project φροντίζουμε να συμφωνεί το Project Identifier με το App ID του Provisioning Profile η αλλιώς αλλάζουμε το identifier του project. Επίσης πρέπει να έχουμε επιλέξει την ομάδα από την οποία έχει εκδοθεί το Provisioning profile από την σελίδα επιλογών του Project.



Η εικόνα παραπάνω παρουσιάζει ένα παράδειγμα της σελίδας επιλογών του project με συμπληρωμένο Identifier και επιλεγμένη Ομάδα.

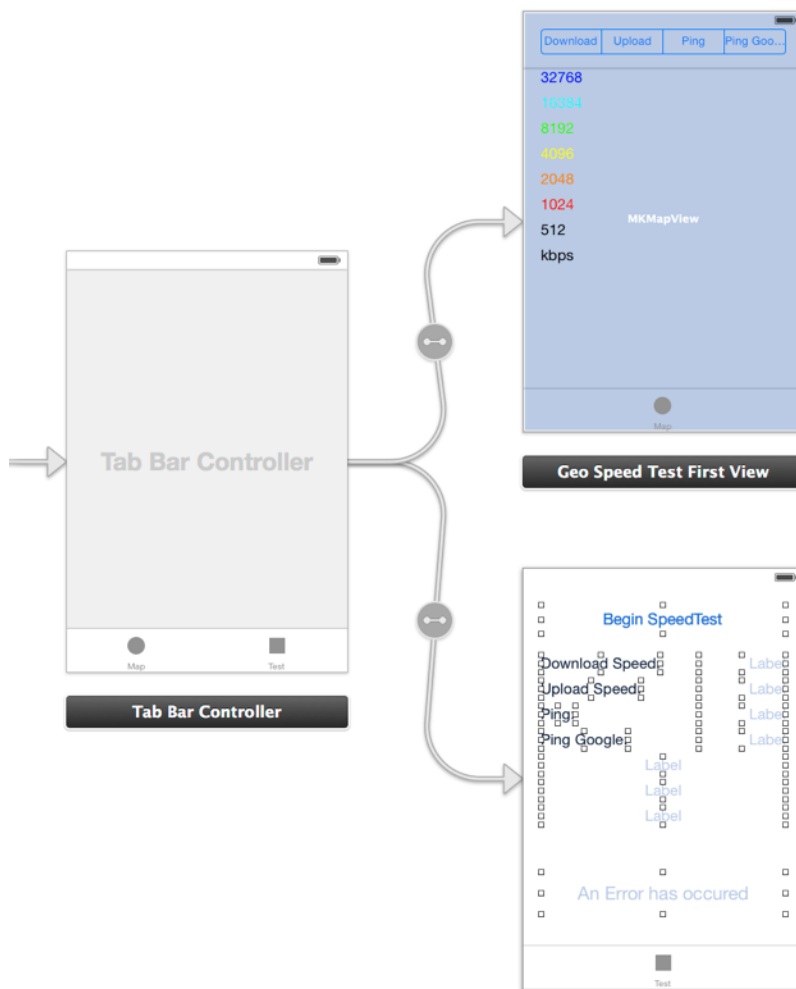
4.2 Περιγραφή της υλοποίησης της εφαρμογής

Όπως αναφέρθηκε στο κεφάλαιο (\$ κεφάλαιο που αναφέρω το MVC) η υλοποίηση της εφαρμογής βασίστηκε στην αρχιτεκτονική σχεδιασμού Model-View-Controller (MVC).

Για την διευκόλυνση της ανάπτυξης με το μοντέλο MVC έγινε χρήση της λειτουργικότητας storyboard του Xcode που βοηθάει στον σχεδιασμό εμφανίσεων (views) και στην σύνδεση των τμημάτων των εμφανίσεων με τον χειριστή.

Το γραφικό περιβάλλον της εφαρμογής περιέχει δύο διαφορετικές εμφανίσεις (views) όπου κάθε μία από τις εμφανίσεις έχει ένα χειριστή (controller). Επίσης υπάρχει και ένας τρίτος controller ο οποίος διαχειρίζεται ποια από τις δύο εμφανίσεις θα προβάλλεται. Αυτή η προσέγγιση είναι μία συνηθισμένη προσέγγιση για τον σχεδιασμό εφαρμογών στο iOS και υποστηρίζεται από την λειτουργικότητα Tab View. Η Tab View όταν επιλεγεί δημιουργεί αυτόματα ένα Tab View Controller που περιέχει την απαραίτητη λειτουργικότητα για την διαχείριση των διαφορετικών εμφανίσεων. Επίσης αυτόματα δημιουργούνται οι δύο εμφανίσεις καθώς και οι χειριστές τους.

Η σχεδίαση της διεπαφής χρήστη (User Interface, UI) μπορεί να επιτευχθεί χρησιμοποιώντας εξ' ολοκλήρου την λειτουργικότητα που προσφέρει ο storyboard του Xcode. Με αυτή την προσέγγιση δεν υπάρχει αρχείο που περιγράφει την εμφάνιση καθώς και την διασύνδεση των τμημάτων της εμφάνισης. Η προσθήκη των τμημάτων της εμφάνισης γίνονται με γραφικό τρόπο και απεικονίζονται μέσω του storyboard. Η διασύνδεση των τμημάτων γίνεται και αυτή μέσω της λειτουργίας του storyboard σε συνδυασμό με την προβολή του header του αντίστοιχου ελεγκτή για κάθε εμφάνιση. Ακολουθεί μία παρουσίαση του storyboard της εφαρμογής.



Εικόνα 4.1 - Απεικόνιση του storyboard της εφαρμογής.

Στην παραπάνω εικόνα είναι ορατές οι δύο εμφανίσεις καθώς και μια αντιπροσώπευση της σύνδεσής τους με τον Tab View Controller ο οποίος ελέγχει το ποια εμφάνιση θα παρουσιάζεται στον χρήστη. Ο Tab View Controller διαχειρίζεται επίσης και τα δυο τμήματα στο κάτω μέρος της οθόνης που επιτρέπουν στον χρήστη της εφαρμογής να επιλέξει ποια εμφάνιση επιθυμεί να προβληθεί. Η λειτουργικότητα του Tab View Controller είναι ήδη προγραμματισμένη από την Apple και οι επιλογές του μπορούν να τροποποιηθούν με γραφικό τρόπο.

Η πρώτη εμφάνιση που παρουσιάζεται στον χρήστη όταν εκτελεστεί η εφαρμογή είναι η εμφάνιση που ονομάζεται Geo Speed Test First View που θα αναφέρεται και ως πρώτη εμφάνιση και ελέγχεται από τον χειριστή Geo Speed Test First View Controller που θα αναφέρεται και ως πρώτος χειριστής. Η εμφάνιση και ο χειριστής δημιουργήθηκαν με

αυτόματο τρόπο αρχικά και έπειτα τροποποιήθηκαν για να εμφανίζουν την λειτουργικότητα που επιθυμούμε.

Η πρώτη εμφάνιση της εφαρμογής περιέχει τον τρόπο με τον οποίο προβάλλονται τα αποτελέσματα των μετρήσεων όλων των χρηστών πάνω από ένα χάρτη. Η εμφάνιση του χάρτη γίνεται μέσω του MKMapView στοιχείου που είναι μέρος του MapKit Framework και παρέχεται από την Apple. Μέσω του MKMapView μπορούμε να επιλέξουμε τον τύπο του χάρτη που επιθυμούμε να εμφανίζεται. Για την εμφάνιση του χάρτη έχουμε τρεις επιλογές: Απλός χάρτης(Map), Δορυφορικός (satellite) και υβριδικός χάρτης (hybrid). Επιλέξαμε υβριδικό χάρτη γιατί εμφανίζει δορυφορικές εικόνες της περιοχής που παρουσιάζει μαζί με δεδομένα χάρτη όπως ονόματα οδών.

Για να μπορέσουμε να ελέγξουμε τον χάρτη ορίσαμε τον πρώτο χειριστή ως delegate του MKMapView. Επίσης συνδέσαμε τον χάρτη και τις ετικέτες (label), οι οποίες προβάλλουν την λεζάντα του χάρτη, με το αρχείο header του πρώτου χειριστή. Όλα αυτά τα στοιχεία συνδέονται σαν Outlets του χειριστή και είναι υπό τον πλήρη έλεγχό του.

Η δήλωση των Outlet έχει την μορφή:

```
@property (weak, nonatomic) IBOutlet MKMapView *mapView;
```

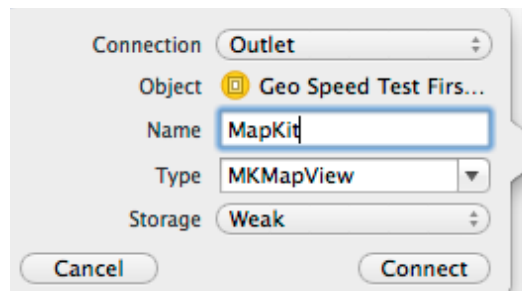
Αυτή η δήλωση δημιουργεί ένα property η αλλιώς μεταβλητή συνόδου η οποία είναι δημόσια, επειδή δηλώθηκε στο αρχείο header του χειριστή. Το property αυτό έχει ακόμη τις ιδιότητες weak και nonatomic. Η ιδιότητα weak υποδηλώνει στο σύστημα διαχείρισης μνήμης του λειτουργικού πως η μεταβλητή αυτή μπορεί να καταστραφεί εάν κάποιο από τα άλλα λειτουργικά τμήματα της εφαρμογής παύει να έχει σύνδεση με την μεταβλητή δηλαδή όταν δεν χρησιμοποιείται πλέον. Όλα τα στοιχεία των εμφανίσεων που συνδέονται ως Outlet με τον χειριστή έχουν συνήθως την ιδιότητα weak επειδή με αυτό τον τρόπο είναι διαθέσιμα για όσο η εμφάνιση προβάλλεται στον χρήστη ενώ διαγράφονται όταν η εμφάνιση παύει να είναι ενεργή. Η εμφάνισης του iOS δεν παύουν να είναι ενεργές ακόμη και όταν δεν προβάλλονται στην ίδια εφαρμογή. Το λειτουργικό παρόλα αυτά μπορεί να διαγράψει την εμφάνιση ακόμη και όταν η εφαρμογή είναι ανοιχτή επειδή υπάρχει έλλειψη μνήμης, πράγμα το οποίο όμως είναι πολύ σπάνιο. Παρόλα αυτά σε καταστάσεις που το λειτουργικό αποδεσμεύει την εμφάνιση είναι επιθυμητό να αποδεσμεύονται και οι

μεταβλητές της και γι αυτό τον λόγο μεταβλητές που συνδέουν εμφανίσεις με χειριστές έχουν την ιδιότητα `weak`. Ό,τι ειπώθηκε για την ιδιότητα `weak` σε αυτή την αναφορά ισχύει και για κάθε άλλη δήλωση που θα περιέχει την ιδιότητα `weak` και δεν θα σχολιαστεί ξανά.

Η ιδιότητα `nonatomic` υποδηλώνει πως η μεταβλητή αυτή δεν θα προσπελαστεί από πολλαπλά νήματα (`threads`). Με αυτό τον τρόπο ενημερώνουμε τον μεταγλωττιστή πως δεν επιθυμούμε να υπάρχουν μηχανισμοί κλειδώματος της πρόσβασης στη μεταβλητή. Εάν επιθυμούμε γενικά να έχουμε κάποια μεταβλητή την οποία θα προσπελαίνουν πολλαπλά νήματα τότε θα πρέπει να δηλώσουμε την μεταβλητή με την ιδιότητα `atomic` η να μη ορίσουμε κάποια ιδιότητα γιατί εάν δεν υποδηλώσουμε κάποια ιδιότητα τότε η προεπιλογή είναι πως η μεταβλητή θα έχει την ιδιότητα `atomic`. Όταν χρειάζεται να διαχειριστούμε μία μεταβλητή η οποία έχει την ιδιότητα `atomic` θα πρέπει να υλοποιήσουμε και μεθόδους ασφαλούς πρόσβασης σε αυτή. Στα πλαίσια αυτής της εργασίας δεν υπάρχουν μεταβλητές που έχουν την ιδιότητα `atomic` λόγω του τρόπου σχεδιασμού της εφαρμογής με τον οποίο αποφεύγουμε την πρόσβαση σε μεταβλητές μέσω άλλων νημάτων. Όταν εκτελούνται άλλα νήματα σε αυτή την υλοποίηση τα αποτελέσματα των εργασιών τους αποτυπώνονται μέσω της χρήσης μεθόδων `delegate` που καλούν το κεντρικό νήμα του ελεγκτή για να διαχειριστεί τα δεδομένα. Άρα όλες οι δηλώσεις μεταβλητών σε αυτή την εργασία περιέχουν την ιδιότητα `nonatomic`.

Μετά ακολουθεί ο τύπος σύνδεσης (`connection`) της μεταβλητής που στην περίπτωση αυτή είναι `IBOutlet` και υποδηλώνει πως αυτή η μεταβλητή είναι δείκτης σε `Outlet`. Τέλος ακολουθεί ο τύπος (`Type`) του δείκτη. Εδώ έχουμε την δήλωση ενός `Outlet` σε `MKMapView` και τέλος έχουμε το όνομα της μεταβλητής. Οι μεταβλητές αυτές που συνδέουν τα στοιχεία της εμφάνισης με τον χειριστή, δηλαδή τα `Outlet` δημιουργούνται με γραφικό τρόπο στο `Xcode`. Η διαδικασία που ακολουθείται για την δημιουργία τους είναι απλός. Αρχικά επιλέγουμε το στοιχείο της εμφάνισης που θέλουμε να συνδέσουμε και μετά κρατώντας πατημένο το πλήκτρο `control` στο πληκτρολόγιο και κάνοντας συνεχόμενο κλικ πάνω στο αντικείμενο τραβάμε μία γραμμή στο σημείο του κώδικα που θέλουμε να εισαχθεί το `Outlet` και απελευθερώνουμε το κλικ. Με την απελευθέρωση του κλικ εμφανίζεται ένας διάλογος με τον οποίο δηλώνουμε το όνομα της μεταβλητής και τις ιδιότητες της. Για γραφικά στοιχεία είναι προεπιλεγμένες οι ιδιότητες `weak` και ανάλογα με

τον τύπο του στοιχείου είναι προεπιλεγμένος και ο κατάλληλος τύπος της μεταβλητής, μπορούμε το να αλλάξουμε εάν έχουμε υλοποιήσει διαφορετικούς τύπους γραφικών στοιχείων για τις εμφανίσεις.. Παρακάτω εμφανίζεται ένα τέτοιο παράθυρο διαλόγου.



Εικόνα 4.2 - Απεικόνιση του διαλόγου δημιουργίας Outlet.

Με τον ίδιο τρόπο δημιουργούμε και τα Outlet για τα στοιχεία τύπου label που εμφανίζουν την λεζάντα του χάρτη. Ο τύπος τους είναι UILabel ο οποίος είναι ο τύπος συστήματος για labels. Για παράδειγμα εμφανίζεται το Outlet του μπλε πεδίου της λεζάντας που ονομάζεται blue label.

```
@property (weak, nonatomic) IBOutlet UILabel *blueLabel;
```

Ο χάρτης της πρώτης εμφάνισης έχει την ικανότητα παρουσίασης των αποτελεσμάτων μετρήσεων. Ο τρόπος με τον οποίο επιτυγχάνουμε την προβολή των μετρήσεων είναι μέσω της δυνατότητας προβολής Tiles τα οποία συμπεριλαμβάνονται στον χάρτη. Τα tiles είναι διάφανες εικόνες που εμφανίζονται πάνω από τα δεδομένα του χάρτη και παρουσιάζουν διάφορα δεδομένα. Ο τύπος Tile που χρησιμοποιείται σε αυτή την εργασία είναι WMS Tiles. Τα WMS Tiles είναι διαφανής εικόνες τύπου png που μεταφορτώνονται χρησιμοποιώντας το πρωτόκολλο WMS. Στην περίπτωση της εφαρμογής αυτής τα δεδομένα των Tiles προσφέρονται από τον Geoserver της διαδικτυακής πλατφόρμας. Για την σωστή εμφάνιση των δεδομένων χρησιμοποιήθηκε η υλοποίηση προσαρμογής εικόνων WMS σε Tile συμβατά με το MapKit framework του iOS 7 που αναπτύχθηκε από τον Stanislav Sumbera και βρίσκεται στην σελίδα: <http://blog.sumbera.com/2014/05/17/wms-on-mapkit-with-ios7/> (6/2014). Στην σελίδα αυτή παρέχονται τρία αρχεία που χρησιμοποιούνται σε αυτή την εφαρμογή. Τα αρχεία αυτά είναι τα: WMSTileOverlay.h το header αρχείο, το WMSTileOverlay.m η υλοποίηση του header και το αρχείο MapViewUtils.h το οποίο περιέχει δηλώσεις μεγεθών για μετατροπή της λειτουργικότητας της υλοποίησης ώστε να είναι συμβατή με διαφορετικά συστήματα χαρτών. Στην

περίπτωση της δικιά μας εφαρμογής δεν υπήρχε ανάγκη μετατροπής της υπάρχουσας υλοποίησης επειδή το σύστημα προβολής χαρτών που χρησιμοποιεί το MapKit framework είναι συμβατό με το σύστημα προβολής χαρτών της Google πάνω στο οποίο αρχικά βασίστηκε η υλοποίηση που παρέχεται για μετατροπή των WMS κλήσεων σε Tiles συμβατά με το MapKit framework. Ο τρόπος με τον οποίο γίνεται η κλήση της υλοποίησης για το WMS είναι πολύ απλός. Αρχικά δηλώνουμε την διεύθυνση URL από την οποία θα μεταφορτωθούν τα δεδομένα με την μορφή που ορίζεται από το WMS. Στην περίπτωση προβολής των αποτελεσμάτων των μετρήσεων ταχύτητας λήψης δεδομένων η δήλωση URL είναι:

```
NSString * url = @"http://*****:8080/geoserver/  
cite/wms?service=WMS&version=1.3.0&request=GetMap&LAYERS=cite  
%3Ameasurements&QUERY_LAYERS=cite  
%3Ameasurements&STYLES=Downlink&srs=EPSG:900913&format=image/  
png&transparent=true&WIDTH=256&HEIGHT=256";
```

Με αυτή την γραμμή δημιουργούμε μία μεταβλητή τύπου NSString, που είναι ο βασικός τρόπος διαχείρισης String του iOS. Ονομάζουμε την μεταβλητή URL και καταχωρούμε σε αυτή το σταθερό τμήμα της URL που χρησιμοποιείται για λήψη των δεδομένων που προβάλλονται στον χάρτη. Το δυναμικό τμήμα της URL προστίθεται αυτόματα από την WMSTileOverlay και περιέχει τις πληροφορίες σχετικά με το ορατό τμήμα του χάρτη. Η κλήση της μεθόδου που αρχικοποιεί και εκτελεί την λειτουργικότητα της WMSTileOverlay είναι:

```
[ self.mkMapView addOverlay:[WMSTileOverlay alloc]  
initWithUrl:url UseMercator:YES];
```

Εδώ καλούμε την μέθοδο addOverlay του MkMapView που είναι το Outlet με το οποίο διαχειριζόμαστε την προβολή του χάρτη. Επίσης αρχικοποιούμε την WMSTileOverlay της υλοποίησης με δεδομένα το παραπάνω URL με τα στατικά δεδομένα της WMS αίτησης και ζητάμε να χρησιμοποιηθεί ο τρόπος προβολής χάρτη που είναι συμβατός με τους χάρτες του συστήματος iOS. Το instance της WMSTileOverlay αποστέλλεται στην addOverlay για προβολή.

Αρχικά δηλώνεται για πρώτη φορά η διεύθυνση με την οποία καλούνται τα δεδομένα των μετρήσεων λήψης, γιατί είναι η προεπιλεγμένη προβολή. Η αρχική δήλωση και έπειτα η

κλήση της μεθόδου που προσθέτει τα Tiles στον χάρτη γίνεται με το που ειδοποιηθεί ο πρώτος χειριστής πως η πρώτη εμφάνιση έχει φορτωθεί στην μνήμη αλλά ακόμη δεν έχει εμφανιστεί στον χρήστη. Ο τρόπος με τον οποίο ειδοποιούμαστε γι αυτό το συμβάν από το λειτουργικό σύστημα είναι μέσω της κλήσης της μεθόδου

```
- (void) viewDidLoad;
```

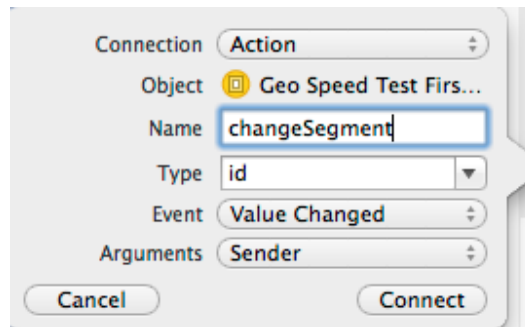
Η οποία υλοποιείται σε κάθε χειριστή ο οποίος διαχειρίζεται κάποια εμφάνιση. Μέσα σε αυτή την μέθοδο λοιπόν προσθέτουμε την αρχικοποίηση της στατικής διεύθυνσης URL για την προβολή Download και μετά εκτελείται η κλήση `addOverlay` που περιγράφηκε παραπάνω. Με αυτό τον τρόπο κάθε φορά που εκτελείται η εφαρμογή θα εμφανίζονται αρχικά τα δεδομένα των μετρήσεων ταχύτητας λήψης.

Για να δοθεί στον χρήστη η δυνατότητα προβολής των αποτελεσμάτων των μετρήσεων ταχύτητας αποστολής δεδομένων, καθώς και της μέτρησης του Ping με την πλατφόρμα μας καθώς και του Ping με τους εξυπηρέτες της Google χρησιμοποιείται το γραφικό στοιχείο διεπαφής (UI Element) που ονομάζεται Segmented Control και παρέχεται από το σύστημα. Το Segmented Control είναι ένα στοιχείο το οποίο δίνει την δυνατότητα στον χρήστη να κάνει μία επιλογή ανάμεσα σε ένα συγκεκριμένο αριθμό επιλογών. Στην περίπτωση αυτής της εφαρμογής το Segmented Control που χρησιμοποιείται περιέχει τέσσερις διαφορετικές επιλογές:

- Download: Παρουσιάζει τα αποτελέσματα των μετρήσεων ταχύτητας λήψης δεδομένων που έχουν εκτελεστεί από τους χρήστες. Είναι η προεπιλεγμένη επιλογή.
- Upload: Παρουσιάζει τα αποτελέσματα των μετρήσεων ταχύτητας αποστολής δεδομένων που έχουν εκτελεστεί από τους χρήστες.
- Ping: Παρουσιάζει τα αποτελέσματα των μετρήσεων Ping ανάμεσα στην συσκευή του χρήστη και την ηλεκτρονική μας πλατφόρμα.
- Ping Google: Παρουσιάζει τα αποτελέσματα των μετρήσεων Ping ανάμεσα στην συσκευή του χρήστη και την ηλεκτρονική μας πλατφόρμα.

Ο τρόπος με τον οποίο συνδέεται το Segmented Control με τον χειριστή είναι μέσω ενός Outlet από το οποίο λαμβάνεται η ενεργή επιλογή και με ένα Action το οποίο ενημερώνει τον χειριστή ότι έχει αλλάξει η κατάσταση του Segmented Control. Η σύνδεση του Outlet είναι παρόμοια με την σύνδεση του χάρτη και των labels. Το Action συνδέεται και αυτό με

το αρχείο header όπως συνδέονται και τα Outlet δηλαδή με γραφικό τρόπο. Παρόλα αυτά η δήλωση του Action δεν δημιουργεί μεταβλητή αλλά δηλώνει μία μέθοδο η οποία θα καλείται όταν αλλάξει η κατάσταση του Segmented Control. Η παρακάτω εικόνα παρουσιάζει τον διάλογο ο οποίος παρουσιάζεται κατά την διαδικασία δημιουργίας του Action.



Εικόνα 4.3 - Απεικόνιση του διαλόγου δημιουργίας Action.

Σε αυτό τον διάλογο επιλέγουμε τύπο connection: Action. Δίνουμε το όνομα της μεθόδου που θα δημιουργηθεί και τα δεδομένα που θα αποστέλλονται στην μέθοδο, που σε αυτή την περίπτωση είναι ένας δείκτης στο στοιχείο Segmented Control που κάλεσε την μέθοδο. Ο διάλογος αυτός παράγει την δήλωση μεθόδου:

```
- (IBAction)changeSegment:(UISegmentedControl *)sender;
```

Στο αρχείο header του πρώτου χειριστή και μία καινή υλοποίηση μεθόδου με την ίδια υπογραφή στο αρχείο της υλοποίησης του χειριστή.

Μέσα στην άδεια υλοποίηση της μεθόδου `changeSegment` πραγματοποιείται ένα case-switch με τα δεδομένα που λαμβάνονται από τον δείκτη του αντικειμένου που κάλεσε την μέθοδο. Ο δείκτης του αποστολέα στην περίπτωση του Segmented Control δίνει την τιμή 0 για την πρώτη επιλογή, την τιμή 1 για την δεύτερη, την τιμή 2 για την τρίτη και την τιμή 3 για την τέταρτη. Έτσι σε κάθε κλήση της μεθόδου δηλώνεται το κατάλληλο URL για την λήψη των αποτελεσμάτων από την ηλεκτρονική πλατφόρμα και επαναφορτίζεται η λειτουργικότητα `addOverlay` για να αλλάξουν τα δεδομένα που προβάλλονται ήδη στον χάρτη. Επίσης αλλάζουμε την λεζάντα του χάρτη αλλάζοντας τις τιμές των label που προβάλλουν την κλίμακα των τιμών των μετρήσεων.

Η τελευταία λειτουργία που παρέχει ο πρώτος χειριστής είναι το κεντράρισμα του χάρτη στην θέση του χρήστη. Αυτή η λειτουργία επιτυγχάνεται με την μέθοδο:

```
-(void)mapView:(MKMapView *)mapView didUpdateUserLocation:
(MKUserLocation *)userLocation {
    [self.mapView setCenterCoordinate:userLocation.coordinate
    animated:YES]; }
```

Αυτή η μέθοδος καλείται αυτόματα από τον χάρτη όταν υπάρχει καινούργια θέση χρήστη και στην υλοποίηση της εκτελείται η λειτουργία setCenterCoordinate με κεντρικό σημείο την θέση του χρήστη που μόλις λήφθηκε. Επίσης υποδηλώνεται πως επιθυμούμε η αλλαγή να γίνει με χρήση animation.

Η δεύτερη εμφάνιση της εφαρμογής μας παρουσιάζει το γραφικό περιβάλλον μέσω του οποίου ο χρήστης ενημερώνεται για τον τύπο της σύνδεσης στο διαδίκτυο που χρησιμοποιεί το κινητό του. Επίσης μέσω της δεύτερης εμφανής ο χρήστης μπορεί να ξεκινήσει την εκτέλεση της μέτρησης της ταχύτητας του δικτύου και του ring που παρουσιάζει. Παράλληλα συλλέγονται κατάλληλα δεδομένα όπως η τοποθεσία του χρήστη και δεδομένα για την τεχνολογία της σύνδεσης δεδομένων. Ο χειριστής της δεύτερης εμφάνισης είναι ο GeoSpeedTestSecondViewController οποίος θα αναφέρεται ως δεύτερος χειριστής. Ο δεύτερος χειριστής είναι αυτός που διαχειρίζεται όλη την διαδικασία πραγματοποίησης μιας δοκιμής ταχύτητας. Για την επίτευξη της δοκιμής χρησιμοποιούνται διαφορετικές υλοποιήσεις για κάθε διαφορετικό τύπο δοκιμής.

Αρχικά όταν ενημερωθεί ο δεύτερος χειριστής πως η εμφάνιση έχει φορτωθεί εκτελούνται κάποιοι έλεγχοι για να επιβεβαιώσουμε την ύπαρξη σύνδεσης στο διαδίκτυο και τον τύπο της σύνδεσης. Έτσι όταν ο δεύτερος χειριστής λάβει την ειδοποίηση πως η δεύτερη εμφάνιση έχει φορτωθεί εκτελείται η μέθοδος

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    self.isFirstTest=YES;
    self.locationManager=[[CLLocationManager alloc] init];

    Reachability* wifiReach = [Reachability
    reachabilityWithHostName: @"*****"];
```

```

    NetworkStatus netStatus = [wifiReach
currentReachabilityStatus];
    NSString *string;
    switch(netStatus) {
        case NotReachable:
            string = @"Server Not Reachable";
            self.displayError.text=string;
            self.displayError.hidden=NO;

            break;
        case ReachableViaWiFi:
            string = @"Connected using WiFi";
            self.connectionType.text=string;
            self.connectionType.hidden=NO;
            self.testButton.enabled=YES;
            break;
        case ReachableViaWWAN:
            string = @"Connected using Mobile Data";
            self.connectionType.text=string;
            self.connectionType.hidden=NO;
            self.testButton.enabled=YES;
            break;
        default:
            string = @"Unknown Network State";
            self.displayError.text=string;
            self.displayError.hidden=NO;
            break;}}

```

Σε αυτή την μέθοδο μεταβιβάζουμε το μήνυμα σε πιθανές άλλες κλάσεις που μπορεί να έχουν κληρονομηθεί. Στην περίπτωση αυτή δεν έχουμε κάποια ιδιαίτερη σχέση κληρονομιάς επειδή χρησιμοποιούμε μία απλή κλάση χειριστή, το μόνο που κληρονομείται είναι η τυπική υλοποίηση χειριστή που έχει το σύστημα. Το μήνυμα μεταβιβάζεται με την [super viewDidLoad]; . Έπειτα αρχικοποιείται ο CLLocationManager ο οποίος θα μας παρέχει την θέση του χρήστη και ο δείκτης του instance που δημιουργείται αποθηκεύεται σε μία ιδιωτική (private) μεταβλητή του δεύτερου χειριστή. Μετά καλείται η μέθοδος Reachability με την διεύθυνση του server μας για να ενημερωθούμε εάν ο server είναι προσβάσιμος.

Η λειτουργικότητα της Reachability παρέχεται από την εταιρία Apple και μπορεί να βρεθεί εδώ: <https://developer.apple.com/Library/ios/samplecode/Reachability/Introduction/Intro.html> (06/2014)

Η βασική λειτουργία της Reachability είναι η ενημέρωση για το εάν υπάρχει σύνδεση με μία διεύθυνση δικτύου και εάν υπάρχει δυνατότητα σύνδεσης ποια τεχνολογία δικτύου χρησιμοποιείται, κινητά δεδομένα ή ασύρματο δίκτυο Wi-Fi. Το πλεονέκτημα της

Reachability είναι πως είναι πολύ αποδοτική ώστε να τρέχει αποτελεσματικά στο μικρό παράθυρο που υπάρχει ανάμεσα στην κλήση της μεθόδου ViewDidLoad και την προβολή της εμφάνισης στον χρήστη. Επίσης είναι μία μέθοδος που παρέχεται από τον κατασκευαστή του λειτουργικού συστήματος iOS που δεν απαιτεί καμία τροποποίηση για την λειτουργία της.

Σε περίπτωση που η εφαρμογή μας έχει πρόσβαση στο διαδίκτυο τότε ενεργοποιείται το κουμπί που ξεκινάει την εκτέλεση της δοκιμής και ανάλογα με τον τύπο του δικτύου εμφανίζουμε και ένα ενημερωτικό μήνυμα στην οθόνη που υποδηλώνει τον τύπο σύνδεσης που χρησιμοποιείται. Εάν δεν υπάρχει σύνδεση ή εάν υπάρξει κάποιο πρόβλημα με την εκτέλεση της εντολής εμφανίζεται μήνυμα λάθους και το κουμπί που ξεκινάει την εκτέλεση της δοκιμής παραμένει απενεργοποιημένο.

Μετά την προβολή της εμφάνισης στον χρήστη και εάν υπάρχει σύνδεση στο δίκτυο ο χρήστης μπορεί να ξεκινήσει την δοκιμή πιέζοντας το κουμπί “Begin SpeedTest”. Η πίεση του κουμπιού ενεργοποιεί μία μέθοδο τύπου Action, όπως και η διαδικασία με την οποία χρησιμοποιείται και στο Segmented Control της πρώτης προβολής. Η εκτέλεση αυτού του Action ξεκινάει την διαδικασία δοκιμής. Αρχικά αλλάζει το μήνυμα που εμφανίζει το κουμπί σε “Test in Progress...” έπειτα τα πεδία τα οποία εμφανίζουν τα αποτελέσματα των μετρήσεων γίνονται αόρατα στην περίπτωση που έχει ήδη εμφανιστεί μια προηγούμενη δοκιμή. Το επόμενο βήμα που εκτελείτε είναι η συλλογή πληροφοριών σχετικά με την σύνδεση κινητής τηλεφωνίας που χρησιμοποιεί η συσκευή. Συλλέγονται πληροφορίες σχετικά με την τεχνολογία της κινητής σύνδεσης καθώς και πληροφορίες σχετικά με τον πάροχο, και τον κωδικό της χώρας που υποδηλώνει η κινητή σύνδεση σύμφωνα με τα δεδομένα του mobile country code (MCC). Οι πληροφορίες αυτές αποθηκεύονται σε ιδιωτικές μεταβλητές του χειριστή. Εκτελείται ξανά δοκιμή πρόσβασης χρησιμοποιώντας την Reachability της Apple για να ενημερωθεί ο χειριστής για τυχόν αλλαγές που μπορεί να συνέβησαν στο δίκτυο από την στιγμή που προβλήθηκε η εμφάνιση μέχρι την στιγμή που πατήθηκε το κουμπί για εκτέλεση δοκιμής. Ανάλογα και πάλι με τις πληροφορίες σύνδεσης που λαμβάνονται από την Reachability εμφανίζονται διαφορετικά μηνύματα στον χρήστη. Εάν δεν υπάρχει σύνδεση εμφανίζεται μήνυμα λάθους στον χρήστη και το κουμπί παραμένει απενεργοποιημένο αλλά τώρα εμφανίζει το μήνυμα “NO Internet Connection” για να υποδηλώσει την ανυπαρξία σύνδεσης. Στην περίπτωση που η σύνδεση

χρησιμοποιεί κινητά δεδομένα τότε εμφανίζεται στον χρήστη το όνομα του παρόχου κινητής τηλεφωνίας και εάν το δίκτυο είναι δεύτερης γενιάς (GPRS, EDGE) τότε εμφανίζουμε ένα μήνυμα που προειδοποιεί τον χρήστη πως η δοκιμή μπορεί να χρειαστεί περισσότερο χρόνο λόγω των εξαιρετικά χαμηλών δυνατοτήτων σύνδεσης που προσφέρουν τα κινητά δίκτυα δεύτερης γενιάς. Εάν λοιπόν υπάρχει σύνδεση συνεχίζεται η εκτέλεση της διαδικασίας της δοκιμής.

Μετά τον έλεγχο συνδεσιμότητας γίνεται αίτηση για λήψη της θέσης του χρήστη. Η αίτηση που κάνουμε είναι ασύγχρονη, δηλαδή δεν λαμβάνουμε άμεσα απάντηση αλλά το σύστημα καλεί μία μέθοδο ανάλογα με το αποτέλεσμα της αίτησης μας όταν έχει αποτελέσματα. Οι μέθοδοι αυτοί είναι προσδιορισμένοι από την τεκμηρίωση της CLLocationManager. Μέσα στις μεθόδους ο προγραμματιστής είναι υπεύθυνος για την διαχείριση της απάντησης του συστήματος. Σε αυτή την εφαρμογή διαχειρίζονται τα μηνύματα λάθους που μπορεί να αποσταλούν από το σύστημα και τα μηνύματα που απαντάνε στην αίτηση για την θέση του χρήστη. Η αίτηση που κάνουμε ζητάει να λάβει την καλύτερη θέση που μπορεί να μας παρέχει το σύστημα, επειδή η απάντηση της μεθόδου CLLocationManager είναι απάντηση τύπου καλύτερης προσπάθειας (best effort) γιατί υπάρχουν πολλοί παράγοντες που επηρεάζουν την ακρίβεια της θέσης που μπορεί να προσδιορίσει η συσκευή. Οι μέθοδοι που διαχειρίζονται τις απαντήσεις του CLLocationManager εμφανίζονται παρακάτω.

```
// handles location error by showing warning (can be
triggered by user denying access
- (void)locationManager:(CLLocationManager *)manager
didFailWithError:(NSError *)error
{
    NSLog(@"didFailWithError: %@", error);
    UIAlertView *errorAlert = [[UIAlertView alloc]
                                initWithTitle:@"Error"
                                message:@"Failed to Get Your Location, Activate location
access" delegate:nil cancelButtonTitle:@"OK"
                                otherButtonTitles:nil];
    [errorAlert show];
}
//updates the current location
- (void)locationManager:(CLLocationManager *)manager
didUpdateToLocation:(CLLocation *)newLocation fromLocation:
(CLLocation *)oldLocation
{
    //log the new location
    self.currentLocation = newLocation;
    self.haveLocation=YES;
}
```

```

        // Stop Location Manager to save resources (we are not
        that interested in pinpoint accuracy)
        [self.locationManager stopUpdatingLocation];
        NSLog(@"New Location: %@", self.currentLocation);}

```

Η πρώτη μέθοδος εκτελείτε σε περίπτωση αποτυχίας λήψης της θέσης του χρήστη. Η μέθοδος αυτή εμφανίζει ένα μήνυμα λάθους που ενημερώνει τον χρήστη και σταματάει την εκτέλεση. Το μήνυμα ενημερώνει τον χρήστη και προτείνει να ελέγξει εάν έχει δώσει δικαιώματα απόκτησης θέσης στην εφαρμογή γιατί η έλλειψη δικαιωμάτων λήψης της θέσης του χρήστη είναι ένα πολύ πιθανό συμβάν.

Η επόμενη μέθοδος αποθηκεύει τα δεδομένα που μας παραχωρεί το σύστημα και σταματάει τον CLLocationManager από το να μας αποστέλλει νέα θέση για την διάρκεια της εκτέλεση αυτής της δοκιμής επειδή η διαδικασία προσδιορισμού της θέσης της συσκευής είναι εξαιρετικά ενεργοβόρα και δεν έχουμε ανάγκη για εξαιρετικά μεγάλη ακρίβεια της θέσης.

Μετά και από την αίτηση για λήψη της γεωγραφικής θέσης της συσκευής ξεκινάμε την εκτέλεση της μέτρησης ping. Παρακάτω παρουσιάζεται η μέθοδος που ενεργοποιείται από το πάτημα του κουμπιού που ξεκινάει την δοκιμή. Μετά θα εξηγηθεί η λειτουργία και ο τρόπος απόκτησης της μέτρησης ping.

```

- (IBAction)beginTest:(UIButton *)sender    //Begin Speedtest
button pressed
{
    NSLog(@"Test Button Pressed");
    sender.enabled=NO;
    [sender setTitle:@"Test in Progress..."
forState:UIControlStateNormal];
    self.displayError.hidden=YES;
    self.downloadSpeed.hidden=YES;
    self.uploadSpeed.hidden=YES;
    self.pingGoogle.hidden=YES;
    self.pingServer.hidden=YES;
    self.mobile.hidden=YES;
    self.carrier.hidden=YES;
    //Find the mobile technology that the phone is using
    CTTelephonyNetworkInfo *telephonyInfo =
[CTTelephonyNetworkInfo new];
    self.mobileTechnology=telephonyInfo;
    NSLog(@"Current Radio Access Technology: %@",
self.mobileTechnology.currentRadioAccessTechnology);
    //find the carrier info
    CTTelephonyNetworkInfo *carrierInfo =
[CTTelephonyNetworkInfo new];
}

```

```

        CTCarrier *carrier = [carrierInfo
subscriberCellularProvider];
        self.phoneCarrier=carrier;
        NSLog(@"Carrier Name: %@ , %@", [self.phoneCarrier
carrierName],[self.phoneCarrier mobileCountryCode]);
        Reachability* wifiReach = [Reachability
reachabilityWithHostName: @"*****"]; //checking server
connectivity and type of connection when user presses the
button
        NetworkStatus netStatus = [wifiReach
currentReachabilityStatus];
        NSString *string;
        switch(netStatus) {
            case NotReachable:
                string = @"Server Not Reachable";
                self.displayError.text=string;
                self.displayError.hidden=NO;
                self.connectionType.text=@"No connection";
                [sender setTitle:@"NO Connection Available"
forState:UIControlStateNormal];
                sender.enabled=NO;
                return;
                break;
            case ReachableViaWiFi:
                string = @"Connected using WiFi";
                self.connectionType.text=string;
                self.connectionType.hidden=NO;
                break;
            case ReachableViaWWAN:
                string = @"Connected using Mobile Data";
                self.connectionType.text=string;
                self.connectionType.hidden=NO;
                self.isMobile=YES;
                self.carrier.text=self.phoneCarrier.carrierName;
                self.carrier.hidden=NO;
                if (![self
isFast:self.mobileTechnology.currentRadioAccessTechnology]) {
                    //If network not fast (GPRS and EDGE) show warning
                    self.mobile.text=@"This might take a while,
please wait";
                    self.mobile.hidden=NO;
                }
                break;
            default:
                string = @"Unknown Network State";
                self.displayError.text=string;
                self.displayError.hidden=NO;
                [sender setTitle:@"NO Connection Available"
forState:UIControlStateNormal];
                sender.enabled=NO;
                return;
                break;
        }
        //Getting Location Data
        self.locationManager.delegate = self;

```

```

        self.locationManager.desiredAccuracy =
kCLLocationAccuracyBest;
        [self.locationManager startUpdatingLocation];
        //Starting the Ping Test
        self.pinger = [[PingTest alloc] init];
        self.pinger.delegate=self;
        [self.pinger runWithHostName:@"*****"];
        return;
    }

```

Η δοκιμή του ping που εκτελείται βασίζεται πάνω στην υλοποίηση Simple Ping που παρέχεται από την Apple και είναι διαθέσιμη εδώ: <https://developer.apple.com/library/mac/samplecode/SimplePing/Introduction/Intro.html> (06/2014)

Στα περισσότερα συστήματα η εκτέλεση ping απαιτεί κάποια συγκεκριμένα δικαιώματα χρήσης που επιτρέπουν την χρήση θυρών τύπου Internet Control Message Protocol (ICMP). Η εφαρμογή μας όμως δεν έχει τα απαραίτητα δικαιώματα και δεν υπάρχει τρόπος να αποκτηθούν. Επίσης δεν υπάρχει κάποια μέθοδο για εκτέλεση ping χρησιμοποιώντας το API του iOS. Οπότε βασιζόμαστε στην υλοποίηση που μας προσφέρει η Apple για εκτέλεση ping που χρησιμοποιεί ειδικές θύρες τύπου ICMP χωρίς την ανάγκη απόκτησης ιδιαιτέρων δικαιωμάτων (non-privileged ICMP socket). Επίσης η Simple Ping προσφέρει και address-resolution. Επειδή η Simple Ping έχει κατασκευαστεί για εκτέλεση σε λειτουργικό σύστημα Mac OS X χρησιμοποιώντας τον terminal του συστήματος έπρεπε να γίνουν κάποιες αλλαγές για να προσαρμόσουμε την λειτουργικότητα που χρειάζεται η εφαρμογή μας.

Η αρχικοποίηση και κλήση της μεθόδου που πραγματοποιεί την δοκιμή ping εμφανίζεται παραπάνω και είναι η τελευταία εντολή που εκτελείται μέσα στο Action που εκτελείται όταν πατηθεί το κουμπί για δοκιμή του δικτύου. Τα αποτελέσματα της δοκιμής αποστέλλονται με τον ίδιο τρόπο που αποστέλλονται και τα αποτελέσματα της αίτησης για την γεωγραφική θέση της συσκευής. Δηλαδή υπάρχει μία μέθοδος στον δεύτερο χειριστή που καλείται από την Simple Ping όταν έχει εκτελεστεί η δοκιμή του ping και μας παρέχει τα αποτελέσματα της δοκιμής. Έπειτα εκτελείται δοκιμή ping και για τους εξυπηρέτες της Google και μετά την λήψη και αυτών των αποτελεσμάτων παρουσιάζονται τα αποτελέσματα στον χρήστη και συνεχίζεται η διαδικασία της δοκιμής με την κλήση της μεθόδου που έχουμε υλοποιήσει για την πραγματοποίηση της μέτρησης ταχύτητας λήψης και την μέτρηση της ταχύτητας αποστολής. Η κλήση της μεθόδου δοκιμής της ταχύτητας γίνεται με τρόπο ο οποίος επιτρέπει την εκτέλεση της μεθόδου στο παρασκήνιο σε διαφορετικό νήμα (thread) εκτέλεσης ώστε να μην αποκοπεί η εκτέλεση του δεύτερου

χειριστή, διότι ο χειριστής είναι υπεύθυνος για την διαδραστικότητα με τον χρήστη και η αποκοπή λειτουργίας του θα κάνει την εφαρμογή να εμφανίζεται ως μη απομακρυνόμενη και μπορεί να έχει σαν αποτέλεσμα τον τερματισμό της εφαρμογής από το σύστημα.

Για την παράλληλη εκτέλεση η κλήση της μεθόδου δοκιμής της ταχύτητας του δικτύου εκτελείται με αυτό τον τρόπο.

```
dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, (unsigned long)NULL), ^{self.speed = [[SpeedTester alloc] init];
    self.speed.delegate=self;
    NSLog(@"Delegate set");
    [self.speed
executeSpeedTestAndReport:@"*****"]});
```

Με την εντολή `dispatch_async` δηλώνεται πως επιθυμούμε να εκτελεστεί η μέθοδος που περιέχει μέσα στην εντολή με τρόπο ασύγχρονο ως προς την εκτέλεση του χειριστή. Θέτουμε την σειρά εκτέλεσης και την προτεραιότητα και καλούμε το block κώδικα που περιέχει την κλήση της μεθόδου εκτέλεσης της δοκιμής ταχύτητας.

Για την διεξαγωγή της δοκιμής ταχύτητας υπήρχε η επιθυμία να χρησιμοποιηθεί κάποιο υπάρχον εξειδικευμένο εργαλείο που θα γνωρίζαμε πως μας παρέχει με ορθά και αποδεκτά αποτελέσματα. Παρόλα αυτά μετά από εκτεταμένη έρευνα απολύθηκε πως αυτή η προσέγγιση δεν θα μπορούσε να λειτουργήσει για την εφαρμογή μας. Αρχικά έγιναν προσπάθειες για να βρεθεί μία έκδοση του γνωστού εργαλείου εκτέλεσης μετρήσεων δικτύου που ονομάζεται `iPerf`. Δυστυχώς δεν υπάρχει κάποια ανοιχτή έκδοση του `iPerf` που είναι διαθέσιμη προς χρήση και συμβατή με το λειτουργικό σύστημα `iOS`. Έγιναν λοιπόν προσπάθειες για μεταφορά του εργαλείου ώστε να γίνει συμβατό με το λειτουργικό σύστημα `iOS`. Αυτή η προσπάθεια όμως αντιμετώπισε πολλά προβλήματα. Ένα από τα κυριότερα είναι η χρήση μεθόδων που παρέχονται από λειτουργικά συστήματα υπολογιστών αλλά είναι απαγορευμένες από κινητά λειτουργικά συστήματα. Επίσης άλλο ένα πρόβλημα ήταν η ανακάλυψη πως οι `POSIX sockets` που χρησιμοποιούνται από το `iPerf` δεν θα μπορούσαν να εκμεταλλευτούν συνδέσεις κινητών δεδομένων και δεν συνιστάται η χρήση τους. (17)

Αυτό θα σήμαινε πως η εφαρμογή μας δεν θα μπορούσε να διεξάγει μετρήσεις της ταχύτητας κινητών δικτύων, το οποίο είναι ένα από τα βασικά χαρακτηριστικά της

εφαρμογής μας. Έπειτα στραφήκαμε σε αναζήτηση εργαλείων του συστήματος αλλά αποκαλύφθηκε πως τα λειτουργικά συστήματα κινητών τηλεφώνων όπως το iOS αλλά και το Android δεν παρέχουν τέτοιου είδους μεθόδους και υπάρχουσες μέθοδοι που υπάρχουν στα λειτουργικά αυτά συστήματα οι οποίες είναι ίδιες με τις μεθόδους που παρέχονται σε παραδοσιακούς προσωπικούς υπολογιστές, είναι περιορισμένες και δεν επιτρέπεται η χρήση τους από τους προγραμματιστές το ζήτημα αυτό αναφέρεται και στην μελέτη του Muckaden Sachit σχετικά με την ανάπτυξη συστήματος εκτέλεσης δοκιμών δικτύου για κινητές συσκευές. (18)

Έτσι στραφήκαμε προς την ανάπτυξη μιας μεθόδου που θα εκτελεί και θα χρονομετρήσει την μεταφορά δεδομένων ανάμεσα στην διαδικτυακή πλατφόρμα και την εφαρμογή. Η ανάγκη για χρονομέτρηση της μεταφοράς πηγάζει από το γεγονός πως το σύστημα iOS δεν μας παρέχει με κάποια μέθοδο με την οποία μπορούμε να λάβουμε τον ρυθμό μετάδοσης των δεδομένων στο δίκτυο. Έτσι η μέθοδος θα χρονομετρούσε την μεταφορά δεδομένων και με το μέγεθος των δεδομένων γνωστό θα μπορούσε να υπολογίσει τον ρυθμό μετάδοσης ο οποίος είναι δεδομένα/χρόνος.

Στο λειτουργικό σύστημα iOS υπάρχουν τρεις τύποι με τους οποίους ο προγραμματιστής μπορεί να ανοίξει μία θύρα (socket) και να μεταφέρει δεδομένα. Ο πρώτος τρόπος είναι μέσω της χρήσης POSIX sockets. Όμως όπως αναφέραμε παραπάνω με αυτό τον τρόπο δεν θα μπορούσε να γίνει μεταφορά δεδομένων σε κινητό δίκτυο οπότε αυτή η μέθοδος δεν μπορούσε να χρησιμοποιηθεί. Οι άλλες δύο μέθοδοι για διαχείριση sockets είναι η CFStream και η NSStream. Η CFStream είναι μία μέθοδος διαχείρισης sockets γραμμένη στην γλώσσα προγραμματισμού C ενώ η NSStream ένας τρόπος διαχείρισης sockets που έχουν δημιουργηθεί με την χρήση CFStream αλλά με αντικειμενοστραφή τρόπο. Το πρόβλημα όμως που παρουσιάζουν αυτοί οι δύο τρόποι διαχείρισης sockets είναι πως η αποστολή και λήψη δεδομένων με την χρήση τους είναι ασύγχρονη και εκτελείται με την αποστολή της αίτησης σε ένα run-loop του συστήματος που αναλαμβάνει να εκτελέσει την μεταφορά στο παρασκήνιο. Αυτό όμως μας αποτρέπει από την δυνατότητα να χρονομετρήσουμε την μεταφορά με αποδοτικό τρόπο επειδή η εντολή μεταφοράς δεδομένων δεν σταματάει την εκτέλεση μέχρι το τέλος της μεταφοράς και επίσης δεν υπάρχει τρόπος να ειδοποιηθούμε πως η μεταφορά έχει τελειώσει η ακόμη πως έχει

αρχίσει, μιας και όπως αναφέραμε η εκτέλεση της μεταφοράς εκτελείται από το σύστημα το οποίο χρησιμοποιεί μία ουρά (queue) για να εξυπηρετήσει όλες τις εφαρμογές. (17)

Έτσι καταλήξαμε στην χρήση των μεθόδων NSMutableURLRequest και NSURLConnection για την αποστολή και λήψη επειδή μας δίνουν την επιλογή μεταφοράς αρχείων με σύγχρονο τρόπο ώστε να μπορούμε να μετρήσουμε την χρονική διάρκεια της μεταφοράς. Έτσι έχουμε σε ένα προσβάσιμο κατάλογο στην ηλεκτρονική πλατφόρμα αρχεία που χρησιμοποιούνται για την εκτέλεση της μέτρησης της ταχύτητας λήψης. Επίσης μέσα στην εφαρμογή υπάρχουν αρχεία που αποστέλλονται στην ηλεκτρονική πλατφόρμα για την μέτρηση της ταχύτητας αποστολής δεδομένων.

Παρακάτω παρουσιάζεται το τμήμα που πραγματοποιεί την μεταφορά δεδομένων από την ηλεκτρονική πλατφόρμα προς την εφαρμογή μας και υπολογίζει τον χρόνο.

```
NSMutableURLRequest *urlRequest = [NSMutableURLRequest
requestWithURL:Address
cachePolicy:NSURLRequestCacheStorageNotAllowed
timeoutInterval:20.0];

NSURLResponse * response = nil;
NSError * error = nil;
NSDate * downStart= [NSDate date];
NSData * data = [NSURLConnection
sendSynchronousRequest:urlRequest
returningResponse:&response
error:&error];
NSDate * downStop= [NSDate date];

NSTimeInterval downTime = [downStop
timeIntervalSinceDate:downStart];
```

Το παραπάνω τμήμα κώδικα ανοίγει μία σύνδεση με την ηλεκτρονική πλατφόρμα. Στην αίτηση ανοίγματος της σύνδεσης προσδιορίζουμε την διεύθυνση με την οποία θα γίνει η σύνδεση, το γεγονός πως δεν επιθυμούμε να γίνει χρήση της δυνατότητας cache για να μην επηρεαστούν τα αποτελέσματα και την χρονική διάρκεια timeout πέρα από την οποία εάν δεν έχει πραγματοποιηθεί η σύνδεση θα υποδείξουμε σφάλμα. Έπειτα κρατάμε την ώρα έναρξης της μεθόδου με την NSDate και αμέσως μετά ξεκινάμε την λήψη των δεδομένων της σύνδεσης που έχουμε ήδη ανοίξει. Στην μέθοδο μεταφοράς δηλώνεται πως η μεταφορά θα είναι σύγχρονη, για να μπορούμε να μετρήσουμε τον συνολικό χρόνο. Αμέσως μετά την ολοκλήρωση της μεταφοράς καταγράφεται ο χρόνος λήξης. Μετά από ελέγχους της

σωστής ολοκλήρωσης υπολογίζουμε την διάρκεια της μεταφοράς με χρήση της `NSTimeInterval`.

Παρόμοια είναι και η διαδικασία με την οποία εκτελείται αποστολή αρχείου στην ηλεκτρονική πλατφόρμα. Η μόνη διαφορά είναι ότι μορφοποιούμε την `NSMutableURLRequest` ώστε να περιέχει το αρχείο προς αποστολή.

Μετά από την εκτέλεση των μετρήσεων επιστρέφουμε στην εκτέλεση του κυρίως νήματος και μαζί παρέχουμε και τα αποτελέσματα των μετρήσεων με την παρακάτω γραμμή.

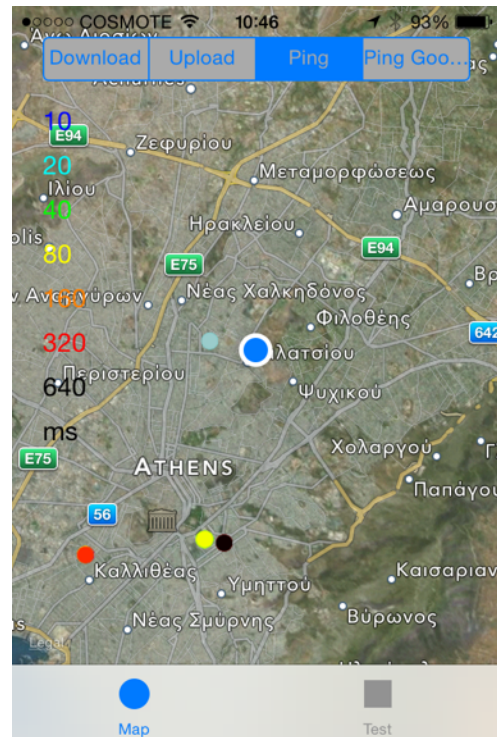
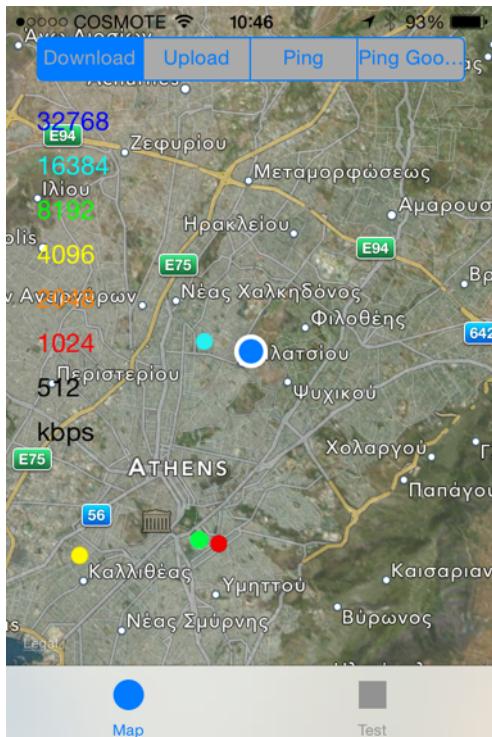
```
dispatch_async(dispatch_get_main_queue(), ^(void) {  
    [self.delegate SpeedTestComplete:self  
upload:uploadSpeed download:downloadSpeed]);});
```

Σε αυτό το σημείο δηλώνουμε πως η κλήση της συμπεριλαμβανομένης μεθόδου θα γίνει στο κυρίως νήμα (main thread) και γίνεται κλήση της μεθόδου του χειριστή που είναι υπεύθυνη για την διαχείριση των αποτελεσμάτων της μέτρησης ταχύτητας.

Η μέθοδος που διαχειρίζεται τα αποτελέσματα της μέτρησης ταχύτητας εμφανίζει αυτά τα αποτελέσματα στον χρήστη και μετά αποστέλλει τα δεδομένα που συλλέχθηκαν κατά την διάρκεια της εκτέλεσης της μέτρησης στην ηλεκτρονική πλατφόρμα χρησιμοποιώντας ένα `PHP POST` που αποστέλλεται στην διαδικασία που περιγράφεται στο κεφάλαιο 3.5.

4.3 Παρουσίαση της διεπαφής χρήστη της εφαρμογής

Σε αυτό το σημείο θα παρουσιαστεί η διεπαφή χρήστη εφαρμογής. Θα παρουσιαστούν παράλληλα και τα διάφορα μηνύματα που λαμβάνει ο χρήστης σε διαφορετικές καταστάσεις. Στην εικόνα 4.5 φαίνεται η πρώτη εμφάνιση της εφαρμογής. Η θέση του χρήστη είναι κεντραρισμένη στον χάρτη και είναι επιλεγμένη η προβολή της ταχύτητας λήψης. Στην εικόνα 4.6 παρουσιάζεται η εμφάνιση των αποτελεσμάτων ping. Ο χρήστης βρίσκεται στην ίδια τοποθεσία και η λεζάντα του χάρτη που υποδηλώνει τα μεγέθη των μετρήσεων έχει προσαρμοστεί ώστε να παρουσιάζει τα μεγέθη μέτρησης του ping. Με παρόμοιο τρόπο παρουσιάζονται και τα αποτελέσματα των μετρήσεων Upload και Ping με τους server της Google.



Εικόνες - 4.5, Προβολή Download (αριστερά) και 4.6 προβολή ping δεξιά

Η δεύτερη εμφάνιση της εφαρμογής περιέχει την διεπαφή χρήστη μέσω της οποίας ο χρήστης ξεκινάει την διαδικασία της μέτρησης χαρακτηριστικών του δικτύου στο οποίο είναι συνδεδεμένο το κινητό και που παρουσιάζονται τα αποτελέσματα μετά την επιτυχή ολοκλήρωση της δοκιμής ή μηνύματα λάθους εάν υπάρξουν προβλήματα κατά την διάρκεια της διεξαγωγής της δοκιμής. Η εικόνα 4.7 παρουσιάζει την εμφάνιση της εφαρμογής πριν ο χρήστης ξεκινήσει την διαδικασία δοκιμής και η συσκευή είναι συνδεδεμένη σε δίκτυο Wi-Fi, παρόμοια είναι και η εμφάνιση στην περίπτωση που υπάρχει σύνδεση με δίκτυο κινητής τηλεφωνίας. Στην εικόνα 4.8 παρουσιάζεται η εμφάνιση στην περίπτωση που κατά την εκκίνηση της εφαρμογής δεν υπάρχει σύνδεση δικτύου ή την περίπτωση να χάθηκε η σύνδεση δικτύου προτού πατηθεί το κουμπί για την εκκίνηση των μετρήσεων. Μπορεί να παρατηρηθεί πως το κουμπί εκκίνησης των μετρήσεων έχει απενεργοποιηθεί και εμφανίζει μήνυμα που μεταδίδει την έλλειψη σύνδεσης, ενώ στο κάτω μέρος της οθόνης εμφανίζεται ξεκάθαρα το μήνυμα λάθους.

Begin SpeedTest

NO Connection Available

Download Speed:

Upload Speed:

Ping:

Ping Google:

Download Speed:

Upload Speed:

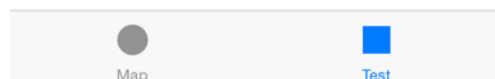
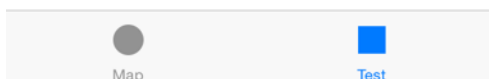
Ping:

Ping Google:

Connected using WiFi

No connection

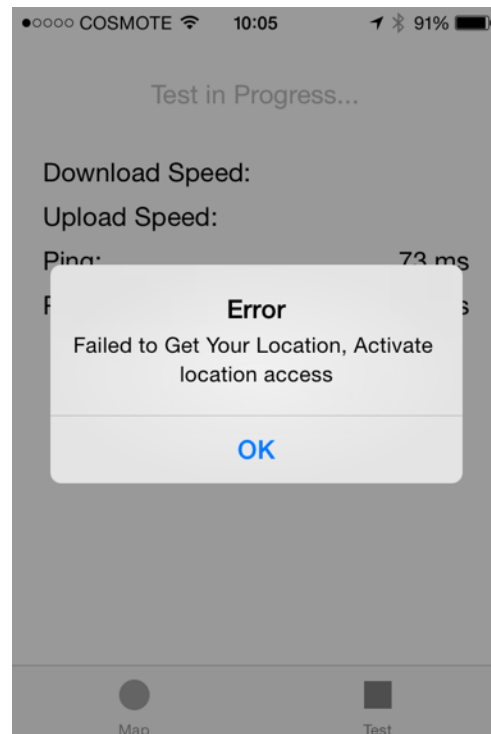
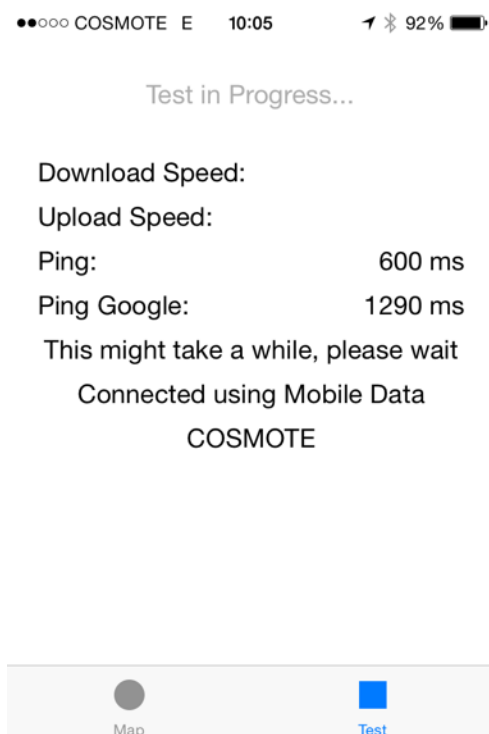
Server Not Reachable



Εικόνες - 4.7 Κανονική εμφάνιση με σύνδεση Wi-Fi και 4.8 εμφάνιση σε περίπτωση μη σύνδεσης.

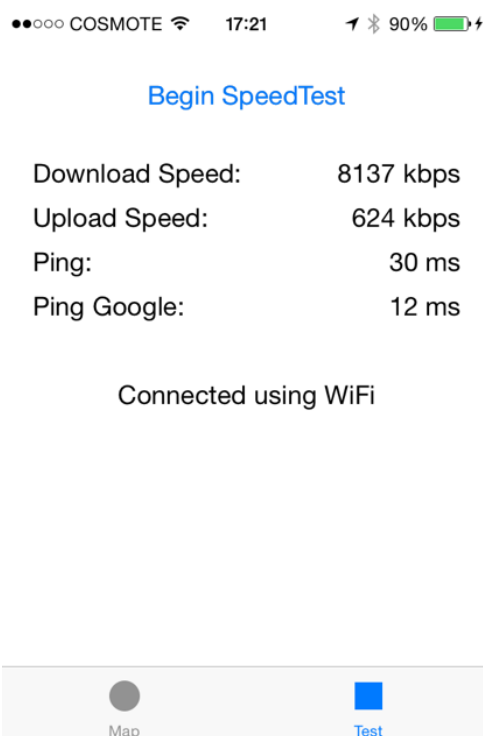
Μετά την εκκίνηση της δοκιμής όπως έχει αναφερθεί και στην υλοποίηση το κουμπί εκκίνησης απενεργοποιείται και εμφανίζονται αποτελέσματα και δεδομένα σχετικά με την σύνδεση της συσκευής. Στην εικόνα 4.9 εμφανίζεται η εμφάνιση της εφαρμογής κατά την διάρκεια εκτέλεσης μίας μέτρησης. Είναι εμφανές το όνομα του παρόχου της κινητής σύνδεσης καθώς και το μήνυμα που ενημερώνει τον χρήστη πως η μέτρηση θα διαρκέσει αρκετό χρονικό διάστημα λόγω της τεχνολογίας σύνδεσης δικτύου, που στην συγκεκριμένη περίπτωση είναι δίκτυο EDGE. Είναι ακόμη ορατά και τα αποτελέσματα των μετρήσεων ping που είναι αρκετά μεγάλα λόγω του τύπου δικτύου στο οποίο είμαστε συνδεδεμένοι. Στην εικόνα 4.10 εμφανίζεται το μήνυμα λάθους που

παρουσιάζεται στον χρήστη στην περίπτωση αδυναμίας απόκτησης της γεωγραφικής θέσης της συσκευής. Η εμφάνιση είναι σκοτεινή επειδή έχει απενεργοποιηθεί η εκτέλεση της και ο χρήστης πρέπει να αναγνωρίσει το μήνυμα.



Εικόνες 4.9 Μέτρηση σε εκτέλεση και 4.10 αδυναμία απόκτησης θέσης

Η εικόνα 4.11 παρουσιάζει την εμφάνιση των αποτελεσμάτων μετά από την εκτέλεση μιας επιτυχημένης δοκιμής.



Εικόνα 4.11 Αποτελέσματα μέτρησης.

5. Συμπεράσματα και επεκτάσεις

Στα πλαίσια αυτής της εργασίας εξερευνήθηκαν και υλοποιήθηκαν διάφορες τεχνικές μέτρησης διάφορων χαρακτηριστικών των δικτύων στα οποία μπορούν να συνδεθούν σύγχρονες κινητές συσκευές. Παρατηρήθηκαν οι δυσκολίες που παρουσιάζονται κατά την διάρκεια ανάπτυξης τεχνικών για την αποτύπωση μεγεθών που αφορούν την ποιότητα του δικτύου. Δυσκολίες οι οποίες αφορούν την πληθώρα των διαφορετικών τεχνολογιών διασύνδεσης αλλά και περιορισμούς που επιβάλλονται από τους κατασκευαστές των λειτουργικών συστημάτων κινητών τηλεφώνων. Όμως παράλληλα αποκαλύφθηκε και η αξία των προτύπων τεχνολογιών που είναι συμβατές με εξαιρετικά πολλές και διαφορετικές υλοποιήσεις. Χάρη σε ανοιχτά πρότυπα επιτεύχθηκε η υλοποίηση μίας διαδικτυακής πλατφόρμας που έχει την δυνατότητα να εξυπηρετεί ένα μεγάλο αριθμό διαφορετικών υλοποιήσεων και συστημάτων και παράλληλα να παραμένει εύκολα επεκτάσιμη για μελλοντικές αλλαγές και προσθήκες.

Η εφαρμογή που αναπτύχθηκε στα πλαίσια της εργασίας αυτής είναι πλήρως λειτουργική και συμμορφώνεται με όλους του κανόνες σχεδιασμού εφαρμογών που ορίζει ο κατασκευαστής του συστήματος iOS, δίνοντας την δυνατότητα για μελλοντική παροχή της εφαρμογής μέσω της διάσημης πλατφόρμας εφαρμογών που παρέχεται στο λειτουργικό σύστημα iOS. Οι μετρήσεις που πραγματοποιούνται είναι σε θέση να παρέχουν μία αρκετά κατατοπιστική εικόνα της κατάστασης και των δυνατοτήτων διάφορων δικτύων που εξυπηρετούν συνδέσεις κινητών συσκευών και τα αποτελέσματα είναι διαθέσιμα σε όλους του πιθανούς χρήστες της εφαρμογής αλλά και της διαδικτυακής πλατφόρμας.

Παρόλα αυτά, υπάρχει πάντα η δυνατότητα βελτίωσης των υπηρεσιών που παρέχονται από τα διάφορα τμήματα αυτής της εργασίας. Γι' αυτό θα παρουσιαστούν διάφορες πιθανές επεκτάσεις αυτής της εργασίας.

Αρχικά μία από τις πιο προφανείς επεκτάσεις θα ήταν η προσθήκη και άλλων τύπων μετρήσεων στην εφαρμογή για μία ακόμη πιο ολοκληρωμένη εικόνα του δικτύου. Αυτές οι μετρήσεις θα μπορούσαν να αφορούν μεγέθη όπως την ένταση του σήματος, την καθυστέρηση ανοίγματος σύνδεσης.

Άλλες πιθανές επεκτάσεις θα μπορούσαν να παρέχουν διαφορετικές υλοποιήσεις δοκίμων. Με βάση την πρόσφατη ανακοίνωση του νέου λειτουργικού συστήματος iOS 8 και με καινούργια τα εργαλεία και την αυξημένη ελευθερία που υπόσχεται πως θα παρέχεται σε αυτό και σε επόμενες εκδόσεις του λειτουργικού συστήματος iOS θα μπορούσαν να αναπτυχθούν καινούργιες υλοποιήσεις δοκιμών που θα μπορούν να εκμεταλλευτούν πιθανούς μειωμένους περιορισμούς. Επίσης με βάση την ανακοίνωση για δυνατότητα χρησιμοποίησης υλοποιήσεων που και υπηρεσιών από άλλες εφαρμογές, που ανακοινώθηκε για το iOS 8, θα μπορούσαν να χρησιμοποιηθούν υλοποιήσεις από εφαρμογές που εξειδικεύονται σε διάφορα μετρικά δικτύου ώστε να υπάρχει μία ομοιομορφία αποτελεσμάτων.

Θα μπορούσε ακόμη να προστεθεί ασφάλεια στον τρόπο μεταφοράς των δεδομένων των μετρήσεων από την εφαρμογή προς την διαδικτυακή πλατφόρμα. Σε περίπτωση που η εφαρμογή αρχίσει να συλλέγει δεδομένα που μπορούν να ταυτοποιήσουν τον χρήστη σε κάποια μελλοντική επέκταση, ίσως να είναι επιθυμητή η ασφαλής μεταφορά των δεδομένων από την εφαρμογή προς την πλατφόρμα με χρήση κρυπτογραφίας.

Ακόμη θα μπορούσε να βελτιωθεί η χρηστικότητα της εφαρμογής ανά τον κόσμο με την προσθήκη περισσότερων εξυπηρετών για την διαδικτυακή πλατφόρμα. Δηλαδή θα μπορούσε να δημιουργηθεί ένα σύνολο από εξυπηρετές που θα έχουν την δυνατότητα να εξυπηρετήσουν με καλύτερο τρόπο την εφαρμογή στην περίπτωση που η εφαρμογή αρχίσει να χρησιμοποιείται σε παγκόσμιο επίπεδο. Με αυτό τον τρόπο θα μπορούσαν επίσης να συλλεχθούν και ακόμη περισσότερα αποτελέσματα που ίσως στο μέλλον να επιτρέψουν μία πιο περίπλοκη ανάλυση των δεδομένων με χρήση τεχνικών εξόρυξης δεδομένων, ώστε να εξαχθούν αποτελέσματα που δεν είναι δυνατό να εξαχθούν από ένα μικρό σύνολο μετρήσεων.

Τελειώνοντας θα ήθελα ακόμη να αναφερθώ και στην δυνατότητα για χρήση της διαδικτυακής πλατφόρμας, με ελάχιστες μετατροπές, με σκοπό την μέτρηση εντελώς διαφορετικών μεγεθών που μπορούν αν αποκτηθούν από τους πολλούς διαφορετικούς αισθητήρες που παρέχουν οι σύγχρονες κινητές συσκευές σε συνδυασμό με την δυνατότητα συσχετισμού αυτών των μετρήσεων με την γεωγραφική θέση της

συσκευής. Ένα τέτοιο μέγεθος θα μπορούσε να είναι η ηχητική ρύπανση μίας περιοχής χρησιμοποιώντας το μικρόφωνο της κινητής συσκευής.

Βιβλιογραφία

1. Johansson, J.; Hapsari, W.A.; Kelley, S.; Bodog, G., "Minimization of drive tests in 3GPP release 11," Communications Magazine, IEEE , vol.50, no.11, pp.36,43, November 2012 doi: 10.1109/MCOM.2012.6353680 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6353680&isnumber=6353668>
2. Hapsari, W.A.; Umesh, A.; Iwamura, M.; Tomala, M.; Gyula, B.; Sebire, B., "Minimization of drive tests solution in 3GPP," Communications Magazine, IEEE , vol.50, no.6, pp.28,36, June 2012 doi: 10.1109/MCOM.2012.6211483 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6211483&isnumber=6211473>
3. Mathew Honan. Apple unveils iPhone [Ηλεκτρονικό] 09/01/2007 [Παραπομπή: 18/06/2014] Σύνδεσμος: <http://www.macworld.com/article/1054769/iphone.html>
4. Brock Kyle. What operating system does the iPhone use? Can it run MacOS X applications like Skype and Photoshop? [Ηλεκτρονικό] 05/07/2010 [Παραπομπή: 18/06/2014] Σύνδεσμος: <http://www.everymac.com/systems/apple/iphone/iphone-faq/iphone-runs-os-x-not-macos-x-cannot-run-macos-x-applications-skype-or-ipod-games.html>
5. Nik Fletcher. Apple: "we plan to have an iPhone SDK in developers' hands in February" [Ηλεκτρονικό] 17/10/2007 [Παραπομπή: 18/06/2014] Σύνδεσμος: <http://www.tuaw.com/2007/10/17/apple-we-plan-to-have-an-iphone-sdk-in-developers-hands-in-fe/>
6. Apple Inc. OS X for UNIX Users [Ηλεκτρονικό] 07/2011 [Παραπομπή: 18/06/2014] Σύνδεσμος: http://movies.apple.com/media/us/osx/2012/docs//OSX_for_UNIX_Users_TB_July2011.pdf
7. CALEB GARLING. iPhone Coding Language Now World's Third Most Popular [Ηλεκτρονικό] 09/07/2012 [Παραπομπή: 18/06/2014] Σύνδεσμος: <http://www.wired.com/2012/07/apple-objective-c/>
8. Apple Inc. Developer Library. Model-View-Controller [Ηλεκτρονικό] 17/09/2013 [Παραπομπή: 18/06/2014] Σύνδεσμος: <https://developer.apple.com/library/ios/documentation/general/conceptual/devpedia-cocoa-core/MVC.html>
9. Jonathan Espinosa. Safari declines in mobile browser usage, others gain traction [Ηλεκτρονικό] 01/07/2013 [Παραπομπή: 18/06/2014] Σύνδεσμος: <http://>

www.insidemobileapps.com/2013/07/01/safari-declines-in-mobile-browser-usage-others-gain-traction/

10. Jim Edwards. 6 Charts Will Convince You That Apple Already Totally Dominates Mobile E-Commerce [Ηλεκτρονικό] 12/03/2014 [Παραπομπή: 18/06/2014] Σύνδεσμος: <http://www.businessinsider.com/data-on-apple-and-mobile-e-commerce-2014-3>
11. Jon Evans. Android vs. iOS Development: Fight! [Ηλεκτρονικό] 06/11/2013. [Παραπομπή: 18/06/2014] Σύνδεσμος: <http://techcrunch.com/2013/11/16/the-state-of-the-art/>
12. Lubos Mikusiac. iOS vs. Android Development Comparison [Ηλεκτρονικό] 03/2013. [Παραπομπή: 18/06/2014] Σύνδεσμος: <http://www.passion4teq.com/articles/ios-android-development-comparison-1/>
13. Netcraft. June 2013 Web Server Survey [Ηλεκτρονικό] 06/2013. [Παραπομπή: 18/06/2014] Σύνδεσμος: <http://news.netcraft.com/archives/2013/06/06/june-2013-web-server-survey-3.html>
14. Zack Urlocker. Google Runs MySQL [Ηλεκτρονικό] 12/2005. [Παραπομπή: 18/06/2014] Σύνδεσμος: http://zurlocker.typepad.com/theopenforce/2005/12/googles_use_of_.html
15. Om Malik. Facebook's Insatiable Hunger for Hardware [Ηλεκτρονικό] 25/04/2008. [Παραπομπή: 18/06/2014] Σύνδεσμος: <http://gigaom.com/2008/04/25/facebook-insatiable-hunger-for-hardware/>
16. Geoserver User manual. Working with Databases [Ηλεκτρονικό] 17/09/2013 [Παραπομπή: 18/06/2014] Σύνδεσμος: <http://ares.boundlessgeo.com/geoserver/master/doc/en/user/data/database/index.html>
17. Apple Inc. Developer Library. Using Sockets and Socket Streams [Ηλεκτρονικό] [Παραπομπή: 18/06/2014] Σύνδεσμος: <https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/NetworkingTopics/Articles/UsingSocketsandSocketStreams.html>
18. Muckaden, Sachit. Myspeedtest: active and passive measurements of cellular data networks [Παραπομπή: 18/06/2014] Σύνδεσμος: <http://hdl.handle.net/1853/47674>