

# Χαροκόπειο Πανεπιστήμιο

Τμήμα Πληροφορικής & Τηλεματικής



## Πτυχιακή εργασία

Ανάπτυξη εφαρμογής και δικτυακής πλατφόρμας μέτρησης απόδοσης  
ασύρματων δικτύων για συσκευές Android

**Νικόλαος Κορωνιώτης**

AM: 21015

**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 από τον Οκτώβριο του 2013 έως τον Ιούνιο του 2014.

Στα πλαίσια της εργασίας αυτής ήρθα σε επαφή με ένα πλήθος τεχνολογιών και ταυτόχρονα κλίθηκα να εφαρμόσω 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 Θανάση Κουντουρά με τον οποίο είχαμε μια εξαιρετική 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 εργάστηκαν πάνω στην υλοποίηση και υποστήριξη μιας τόσο απαραίτητης υπηρεσίας.

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

## Πρόλογος

Στο παρόν κείμενο θα παρουσιαστεί η διαδικασία υλοποίησης της πτυχιακής εργασίας με τίτλο: Ανάπτυξη εφαρμογής και δικτυακής πλατφόρμας μέτρησης απόδοσης ασυρμάτων δικτύων για συσκευές Android.

Αρχικά στο πρώτο κεφάλαιο παρουσιάζουμε το πρόβλημα που επιχειρεί να λύσει η παρούσα πτυχιακή. Στην συνέχεια, στο δεύτερο κεφάλαιο παρουσιάζονται τα επιμέρους τμήματα της αρχιτεκτονικής του συστήματος που υλοποιήσαμε. Έπειτα στα κεφάλαια τρία και τέσσερα παρουσιάζονται οι τεχνικές που χρησιμοποιήσαμε για την υλοποίηση, εφαρμογή και την δικτυακή πλατφόρμα αντιστοίχως. Στα κεφάλαια αυτά εμφανίζονται, λεπτομέρειες σχετικά με τις επιλογές μας ως προς τις τεχνολογίες και τα εργαλεία που χρησιμοποιήσαμε, παρέχεται κώδικας πλήρως σχολιασμένος, όπου αυτό κρίθηκε απαραίτητο, ενώ συμπεριλαμβάνουμε και παράδειγμα εκτέλεσης για την εφαρμογή σε πιθανά σενάρια εκτέλεσης συνοδευόμενα από οπτικό υλικό. Τέλος στο κεφάλαιο πέντε, αναφέρουμε τα συμπεράσματα που αποκομίσαμε από της διαδικασίες που ακολουθήσαμε στα πλαίσια της παρούσας πτυχιακής εργασίας, ενώ αναλύουμε και πιθανές επεκτάσεις που είναι δυνατό να γίνουν σε μεταγενέστερη περίοδο.

Η υλοποίηση της παρούσας πτυχιακής, αποτέλεσε μία αρκετά απαιτητική διαδικασία, μέσω της οποίας, απέκτησα μεγαλύτερη οικειότητα όσον αφορά τον προγραμματισμό κινητών συσκευών Android. Παράλληλα, απέκτησα πρακτική γνώση πάνω στην λειτουργία και την παραμετροποίηση διαφόρων εξυπηρετητών που χρησιμοποιήθηκαν στην υλοποίηση της πλατφόρμας, καθώς και στον δικτυακό προγραμματισμό μέσω του οποίου συνδέσαμε την εφαρμογή με την δικτυακή πλατφόρμα.

Το τμήμα της διαδικτυακής πλατφόρμας, υλοποιήθηκε σε συνεργασία με τον συμφοιτητή μου Αθανάσιο Κουντουρά.



## Περιεχόμενα

|  |     |
|--|-----|
| Ευχαριστίες.....   | 3   |
| Πρόλογος.....  | 4   |
| Περίληψη.....  | 7   |
| Abstract.....  | 9   |
| 1. Εισαγωγή.....   | 11  |
| 1.1 Ορισμός του προβλήματος.....   | 11  |
| 1.2 Σκοπός της πτυχιακής.....  | 13  |
| 2 Αρχιτεκτονική του προτεινόμενου συστήματος.....  | 15  |
| 3 Υλοποίηση της εφαρμογής.....   | 19  |
| 3.1 Τεκμηρίωση επιλογών για την υλοποίηση του Client της πλατφόρμας.....                   | 19  |
| 3.2 Εγκατάσταση Eclipse και παραμετροποίηση για ανάπτυξη λογισμικού Android.....           | 24  |
| 3.3 Λεπτομέρειες υλοποίησης Android Client.....  | 30  |
| 3.4 Λειτουργία της εφαρμογής LBSpeedAndroid.....   | 43  |
| 3.5 Σύγκριση άλλων εργαλείων μέτρησης.....   | 47  |
| 4 Υλοποίηση της διαδικτυακής πλατφόρμας.....   | 51  |
| 4.1 Εξήγηση των τεχνολογιών που χρησιμοποιούνται στον Server.....                          | 51  |
| 4.2 Εγκατάσταση και παραμετροποίηση των τμημάτων Server της πλατφόρμας.....                | 53  |
| 4.3 Δημιουργία χρήστη στην MySQL βάση.....   | 60  |
| 4.4 Δημιουργία πίνακα σε MySQL βάση για αποθήκευση μετρήσεων και γεωχωρικών δεδομένων..... | 61  |
| 4.5 Υλοποίηση της υπηρεσίας διαδικτύου.....  | 64  |
| 4.6 Υλοποίηση και λειτουργία του εξυπηρετή για χρήση στην εξαγωγή μετρήσεων.....           | 65  |
| 4.7 Είσοδος και χρήση του Geoserver.....   | 70  |
| 4.8 Οπτική αναπαράσταση των μετρήσεων μας μέσω SLD.....                                    | 82  |
| 4.9 Απεικόνιση αποτελεσμάτων στο διαδίκτυο.....  | 84  |
| 5 Συμπεράσματα και Επεκτάσεις.....   | 85  |
| Βιβλιογραφία.....  | 87  |
| A. Appendices.....   | 91  |
| Appendix A: Κώδικας MainActivity Κλάσης.....   | 91  |
| Appendix B: Κώδικας SpeedTester, ServerComm και RunTest Κλάσεων.....                       | 101 |
| Appendix C: Κώδικας activity_getting_location.xml.....                                     | 110 |



## Περίληψη

Στις μέρες μας, είναι σπάνιο να βρεις κάποιον στις αναπτυγμένες χώρες του πλανήτη που να μην έχει στην κατοχή του τουλάχιστον μία συσκευή με δυνατότητα πρόσβασης στο διαδίκτυο, είτε αυτή είναι ένα “έξυπνο” κινητό, ένας υπολογιστής (Laptop, Desktop, Notebook) ή ακόμα και μία κονσόλα (Game console). Είτε μας αρέσει, είτε όχι το Internet (διαδίκτυο), έχει γίνει ένα αναπόσπαστο κομμάτι της καθημερινότητας μας, προσφέροντας γρήγορους τρόπους επικοινωνίας {μέσω των social media, την ηλεκτρονική αλληλογραφία (e-mail) και την δυνατότητα συνομιλίας μέσω του διαδικτύου (Voice over IP)}, την δυνατότητα on-line αγοραπωλησιών (e-commerce) καθώς και ταχύτατη προσπέλαση τεράστιων αποθεμάτων πληροφορίας. Γίνεται συνεπώς προφανές ότι καθώς αυξάνεται η εξάρτηση μας στις υπηρεσίες του διαδικτύου για την επίλυση των καθημερινών μας προβλημάτων, θα πρέπει να αναπτυχθούν αξιόπιστες μέθοδοι για την μέτρηση και την αξιολόγηση των συνδέσεων που προσφέρουν οι πάροχοι υπηρεσιών Διαδικτύου στους πελάτες τους.

Αντικείμενο της εργασίας αυτής, είναι η ανάπτυξη δικτυακής πλατφόρμας και εφαρμογής για Android συσκευές με απώτερο σκοπό την εξαγωγή μετρήσεων σχετικά με την σύνδεση στο διαδίκτυο και την μεταγενέστερη απεικόνιση αυτών σε χάρτη τόσο στην εφαρμογή, όσο και σε ιστοσελίδα δικιάς μας επινοήσης.

Ειδικότερα, στα πλαίσια αυτής της πτυχιακής εργασίας, αναπτύχθηκε μια πλατφόρμα λογισμικού για την γεωγραφική απεικόνιση της ταχύτητας σύνδεσης αλλά και άλλων μετρικών απόδοσης (key performance indicators, KPIs) κινητών τελευταίας τεχνολογίας (smartphones) που χρησιμοποιούν το λειτουργικό σύστημα Android. Σκοπός μας είναι η πλατφόρμα αυτή να μπορεί να χρησιμοποιηθεί ως ένα εργαλείο για το χαρακτηρισμό ασυρμάτων δικτύων κινητής τηλεφωνίας ή δικτύων βασισμένων στο πρότυπο IEEE 802.11 (wifi). Μια άλλη χρήση της θα μπορούσε να είναι και η πιθανή εύρεση ελαττωμάτων των υποδομών, βάση γεωγραφικών περιοχών και συγκεκριμένων ποιοτικών χαρακτηριστικών. Η αρχιτεκτονική της πλατφόρμας αυτής χωρίζεται σε 2 μέρη: 1)εφαρμογή πελάτης (client application), και 2)εξυπηρετές (servers). Ειδικότερα η εφαρμογή πελάτης η οποία είναι στοχευμένη για συσκευές Android 4.0.3 Ice Cream Sandwich MR1 με μικρότερη επιτρεπτή έκδοση την Froyo (2.2-2.2.3), μετράει και αποστέλλει σε μία βάση δεδομένων, βασισμένη σε MySQL, γενικές ανώνυμες πληροφορίες για την σύνδεση στο διαδίκτυο του εκάστοτε χρήστη, μαζί με την γεωγραφική θέση του χρήστη η οποία υπολογίζεται μέσω GPS κυρίως και απεικονίζεται στο κινητό του χρήστη μέσω Google Maps. Ειδικότερα εκτελεί ελέγχους ταχύτητας αποστολής και λήψης προς το Server κομμάτι της πλατφόρμας. Το δεύτερο κομμάτι της πλατφόρμας, είναι βασισμένο σε ένα εικονικό μηχάνημα,

που βρίσκεται στην υπηρεσία Ωκεανός, εκτελεί Linux (Ubuntu 13.10) και αποτελείται από έναν εξυπηρετή υλοποιημένο με χρήση της γλώσσας Java, ο οποίος συνεργάζεται με την εφαρμογή για την λήψη μετρήσεων, μία βάση δεδομένων MySQL στην οποία αποθηκεύονται τα δεδομένα από τις μετρήσεις, έναν application server ανοιχτού λογισμικού τον οποίο πήραμε έτοιμο (geoserver 2.5) και παραμετροποιήσαμε κατάλληλα τόσο για να δέχεται δεδομένα από την βάση όσο και για να αποστέλλει τα δεδομένα αυτά βάση προτύπων σε μία ιστοσελίδα, η οποία εμφανίζει στοιχεία από την βάση πάνω σε ένα χάρτη βάση των επιλογών του επισκέπτη.

## Abstract

Nowadays, it is rare to find someone in the developed countries of the world that does not have possession of at least one device with Internet access capabilities, whether it is a "smart" cell phone, a computer (Laptop, Desktop, Notebook) or even a console (Game console). Whether we like it or not, the Internet has become an integral part of our daily routine, offering quick ways of communication, the possibility of on-line purchases and sales (e-commerce) and rapid access to huge reserves of information. It is therefore evident, that as our reliance on Internet services to solve our daily problems increases, we need to develop reliable methods for the measurement and evaluation of the connections that are offered by Internet service providers to their customers.

The subject of this thesis, is the development of a web platform and an application for Android devices with the final goal being to receive measurements regarding one's connection to the Internet and later displaying them both on a map of our application, as well as a site of our own devising.

More particularly concerning this dissertation, we developed a software platform for the geographical visualization of Internet connection speed, as well as other key performance indicators (KPI) for mobile devices running the Android operating system (OS). Our aim is the platform that we developed to be able to be used as a tool to characterize wireless networks, both mobile based networks, as well as networks based on the IEEE 802.11 standard (WIFI). Another possible use of our platform could be finding defects in infrastructures, based on geographical regions and specific quality characteristics. The architecture of this platform is divided into two parts: 1) client application, and 2) servers. Specifically, the client application which is targeted for devices having the Android 4.0.3 Ice Cream Sandwich MR1 release, with the lowest allowed version being Froyo (2.2-2.2.3), measures and sends to a database general anonymous information about the internet connection of each user along with the user's geographic location, calculated either via the usage of GPS technology or through one's Internet connection and displayed to the user through Google Maps. In particular the application runs speed tests regarding the user's uplink and downlink speed with the help of the Server, which is part of the platform. The second part of the platform is based on a virtual machine, which was created by using the Okeanos service, runs Linux (Ubuntu 13.10) and consists of a server developed by using the programming language Java, which cooperates with the application to obtain measurements, a MySQL database, an open source application server which we downloaded and customized so that it could host Geoserver, another open source application server which serves

geographical data to a standard-based website, which displays the measurements kept in the database, on a map based on the user's choices.

# 1. Εισαγωγή

## 1.1 Ορισμός του προβλήματος.

Στη σημερινή εποχή, ολοένα και περισσότεροι άνθρωποι ανακαλύπτουν την χρησιμότητα του διαδικτύου, τόσο σαν ένα εργαλείο στην επαγγελματική τους σταδιοδρομία, όσο και σαν μέσο ψυχαγωγίας. Παράλληλα, η αγορά ηλεκτρονικών συσκευών προσωπικής χρήσης, έχει κατακλυστεί από τις αποκαλούμενες έξυπνες κινητές συσκευές (smartphones), οι οποίες προσφέρουν μία πληθώρα από λειτουργίες, ένα σημαντικό ποσοστό των οποίων σχετίζεται με το διαδίκτυο. Είναι λοιπόν προφανές, πως πρέπει να εξασφαλίζεται τόσο η αξιοπιστία, όσο και μία ικανοποιητική ταχύτητα στις συνδέσεις στο διαδίκτυο.

Ως εκ τούτου, δεν είναι περίεργο το ότι έχουν αναπτυχθεί πολλές τεχνολογίες που παρέχουν διαδικτυακή πρόσβαση σε κινητές συσκευές. Γενικώς οι εν λόγω τεχνολογίες μπορούν να ομαδοποιηθούν σε δυο βασικές κατηγορίες: α) Ένα σύνολο κινητών τεχνολογιών βασιζόμενες στο σύστημα GSM (Global System for Mobile Communications), οι οποίες κατηγοριοποιούνται σε τεχνολογίες δεύτερης γενιάς (2G), τρίτης γενιάς (3G) και τέταρτης γενιάς (4G). β) Τεχνολογίες που βασίζονται στο πρότυπο IEEE 802.11, κοινώς γνωστές ως Wi-Fi. Η διαφορετική φύση των τεχνολογιών αυτών, έχουν σαν αποτέλεσμα την σημαντική εξάρτιση της απόδοσης αυτών σε διάφορους παράγοντες, όπως: περιβαλλοντικές συνθήκες, ποιότητα υποδομών του παρόχου, ένταση και παρεμβολή σημάτων καθώς και το πλήθος των ενεργών συνδέσεων.

Παράλληλα όπως αναφέρεται στα [1] και [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.

Συνεπώς, είναι αναγκαία η ύπαρξη αξιόπιστων εργαλείων, βάση των οποίων μπορούν να ληφθούν μετρήσεις για τον χαρακτηρισμό των συνδέσεων των κινητών συσκευών, με εύκολη

και πλήρως κατανοητή από τους χρήστες απεικόνιση των μετρήσεων αυτών και της γεωγραφικής τους συσχέτισης.



## 1.2 Σκοπός της πτυχιακής

Η παρούσα πτυχιακή, έχει σαν κύριο σκοπό να δημιουργήσει μία εφαρμογή για έξυπνα κινητά τηλέφωνα, η οποία θα μπορεί να λαμβάνει μετρήσεις σχετικές με την σύνδεση στο διαδίκτυο και να τις παρουσιάζει στον χρήστη, μαζί με αντίστοιχα αποτελέσματα άλλων χρηστών, πάνω σε ένα χάρτη που θα εμφανίζει την τοποθεσία κάθε μέτρησης. Η χρήση της εφαρμογής θα είναι πλήρως κατανοητή από τον εκάστοτε χρήστη, ενώ τα αποτελέσματα θα απεικονίζονται με απλό και κατανοητό τρόπο.

Επίσης μέρος της πτυχιακής, είναι η ανάπτυξη μιας διαδικτυακής πλατφόρμας, η οποία θα επικοινωνεί με την εφαρμογή για την λήψη των μετρήσεων, θα συλλέγει τα δεδομένα από τις εφαρμογές και τέλος θα επεξεργάζεται τα δεδομένα αυτά με απώτερο σκοπό την αποστολή τους τόσο στις εφαρμογές για την απεικόνιση τους πάνω σε χάρτες, όσο και σε μία ιστοσελίδα η οποία θα είναι προσβάσιμη από τους χρήστες μέσω των ηλεκτρονικών υπολογιστών τους. Ακόμη, η πλατφόρμα θα σχεδιαστεί για να συνεργάζεται με εφαρμογές ανεξαρτήτως το λειτουργικό σύστημα των κινητών συσκευών στα οποία τρέχουν.

Η εφαρμογή στοχεύει σαν χρήστη τον μέσο χρήστη έξυπνων κινητών συσκευών. 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 επαγγελματικά εργαλεία που έχουν αναπτυχθεί από μεγάλες εταιρίες με έτη εμπειρίας σε μετρήσεις δικτύων.



## 2 Αρχιτεκτονική του προτεινόμενου συστήματος

Η πλατφόρμα που υλοποιήθηκε για την λήψη μετρήσεων, καθώς και την αποθήκευση αυτών σε βάση δεδομένων και την μεταγενέστερη απεικόνιση τους στις υλοποιημένες εφαρμογές, αποτελείται γενικώς από δύο βασικά τμήματα τα οποία στην συνέχεια χωρίζονται σε περαιτέρω υποκατηγορίες:

- α) Την εφαρμογή πελάτη για συσκευές Android(Android client)
- β) Τους εξυπηρετές (servers) που διαχειρίζονται τα δεδομένα.

Αρχικά διαθέτουμε την εφαρμογή πελάτη (client) για Android συσκευές η οποία παίρνει μετρήσεις για την ταχύτητα αποστολής (uplink speed) και λήψης (downlink speed) σε συνεργασία με τον εξυπηρετή που υλοποιήθηκε αποκλειστικά για αυτόν τον σκοπό, καθώς και 2 χρόνους ping, έναν με τους εξυπηρετές της Google και έναν με τον εξυπηρετή στον οποίο βρίσκονται όλα τα επιμέρους κομμάτια του (β) προαναφερθέντος τμήματος της πλατφόρμας μέτρησης. Αυτά τα δεδομένα, καθώς και άλλα όπως τον χρόνο διεξαγωγής των μετρήσεων, την έκδοση του λειτουργικού συστήματος, τις συντεταγμένες της συσκευής και την τεχνολογία σύνδεσης στο διαδίκτυο, αποστέλλονται στο τμήμα της πλατφόρμας για αποθήκευση και μεταγενέστερη απεικόνιση μέσω των εφαρμογών.

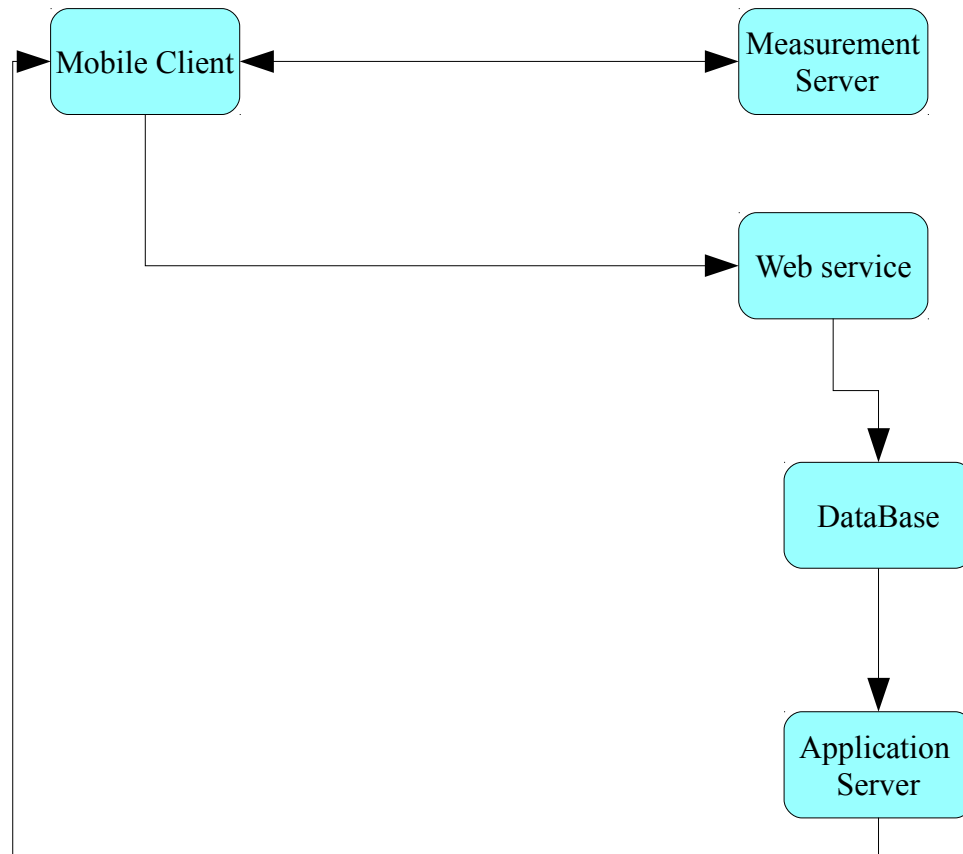
Εν συνεχεία διαθέτουμε το β) τμήμα του όλου συστήματος το οποίο αποτελείται από τα εξής:

1. Έναν πολυνηματικό εξυπηρετή (multithreaded server) υλοποιημένο με χρήση της προγραμματιστικής γλώσσας Java, ο οποίος συνεισφέρει στην λήψη των μετρήσεων.
2. Μία υπηρεσία διαδικτύου (web service) που υφίσταται σε μορφή σελίδας PHP(Hypertext Preprocessor) η οποία δέχεται τις μετρήσεις από τις εφαρμογές πελάτη σε μορφή POST request.
3. Μία βάση MySQL στην οποία καταλήγουν τα δεδομένα από τις μετρήσεις από τις εφαρμογές μέσω του 2).
4. Ένας δικτυακός εξυπηρετής ( web server ) Apache Tomcat, μέσω του οποίου.
5. Ένας εξυπηρετής εφαρμογή (application server) Geoserver έκδοση 2.5, ο οποίος μορφοποιεί την απεικόνιση των μετρήσεων που ελήφθησαν από τις εφαρμογές για την απεικόνιση τους σε χάρτη.

Γενικώς η ορθή λειτουργία της πλατφόρμας μπορεί να περιγραφεί ως εξής:

Μέσω της εφαρμογής, ο χρήστης παίρνει μετρήσεις σχετικές με την ταχύτητα σύνδεσης στο διαδίκτυο. Αν οι μετρήσεις ληφθούν επιτυχώς, τότε προμηθεύονται στην βάση δεδομένων μέσω

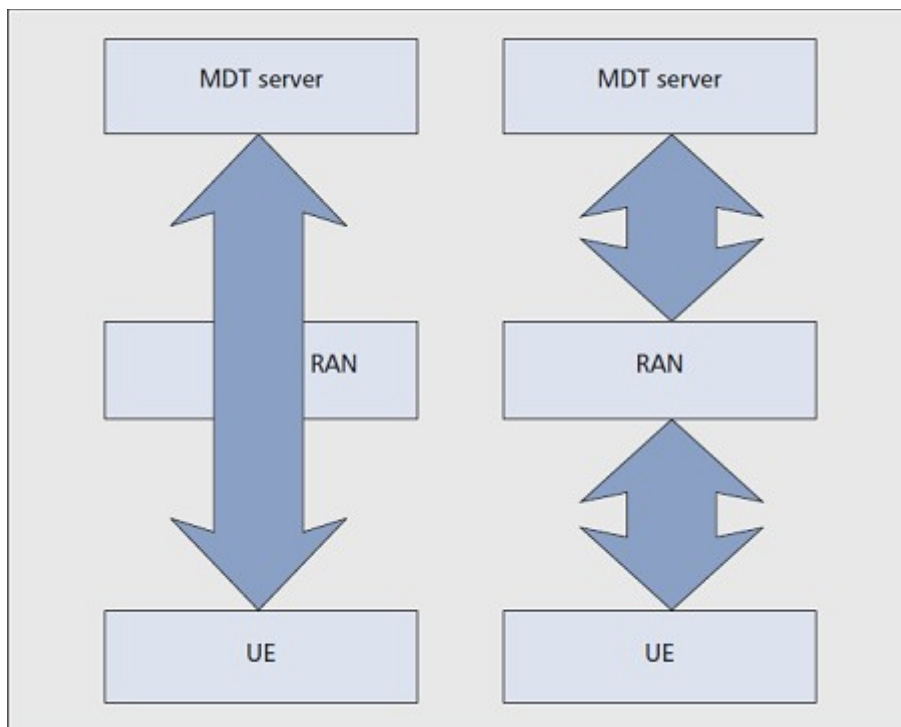
ενός HTTP POST request (Hypertext Transfer Protocol) προς την σελίδα index.php που βρίσκεται στο εικονικό μηχάνημα για τον σκοπό αυτό, ειδάλλως η εφαρμογή δεν στέλνει τίποτα. Έπειτα ο Geoserver, ο οποίος έχει πρόσβαση στην βάση δεδομένων, επιστρέφει τα δεδομένα των μετρήσεων χρησιμοποιώντας το πρωτόκολλο WMS (Web Map Service) στις εφαρμογές για να απεικονιστούν κατόπιν αιτήματος.



Σχεδίαση 2.1: Αρχιτεκτονική πλατφόρμας

Να σημειωθεί εδώ πως η πλατφόρμα που υλοποιήσαμε, λαμβάνει μετρήσεις όμοιες με αυτές που λαμβάνονται κατά την διάρκεια των Drive Testing των κινητών δικτύων. Τα Drive Testings, είναι διαδικασίες που εκτελούν οι πάροχοι κινητής τηλεφωνίας για τον χαρακτηρισμό του δικτύου τους ανά περιοχή. Η διεξαγωγή τους, συνήθως γίνεται μέσω χρήσης εξειδικευμένου υλικού για την λήψη μετρήσεων και συσκευών GPS (Global Positioning System) για την λήψη της τοποθεσίας. Τον τελευταίο καιρό όμως, και εξαιτίας της εμφάνισης των έξυπνων κινητών συσκευών (smartphones) έχει παρατηρηθεί η τάση για ελαχιστοποίηση των Drive Test που διεξάγονται από τους παρόχους κυρίως λόγω του κόστους διεξαγωγής αυτών. 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 ακολουθεί την αρχιτεκτονική μέτρησης τύπου U-Plane MDT.

Το MDT (minimization of Drive Tests) αποτελεί ένα σύνολο τεχνολογιών για την μέτρηση απόδοσης κινητών δικτύων χρησιμοποιώντας τις κινητές συσκευές των χρηστών (User Equipment, UE). Η αρχιτεκτονική U-plane περιγράφει μία διαδικασία μέτρησης απόδοσης δικτύων που εκτελείται με πλήρη άγνοια του παρόχου και το κινητό δίκτυο (Radio Access Network, RAN). 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. [1],[2]



Εικόνα 2.2: U-plane αρχιτεκτονική (αριστερά), C-plane αρχιτεκτονική (δεξιά)

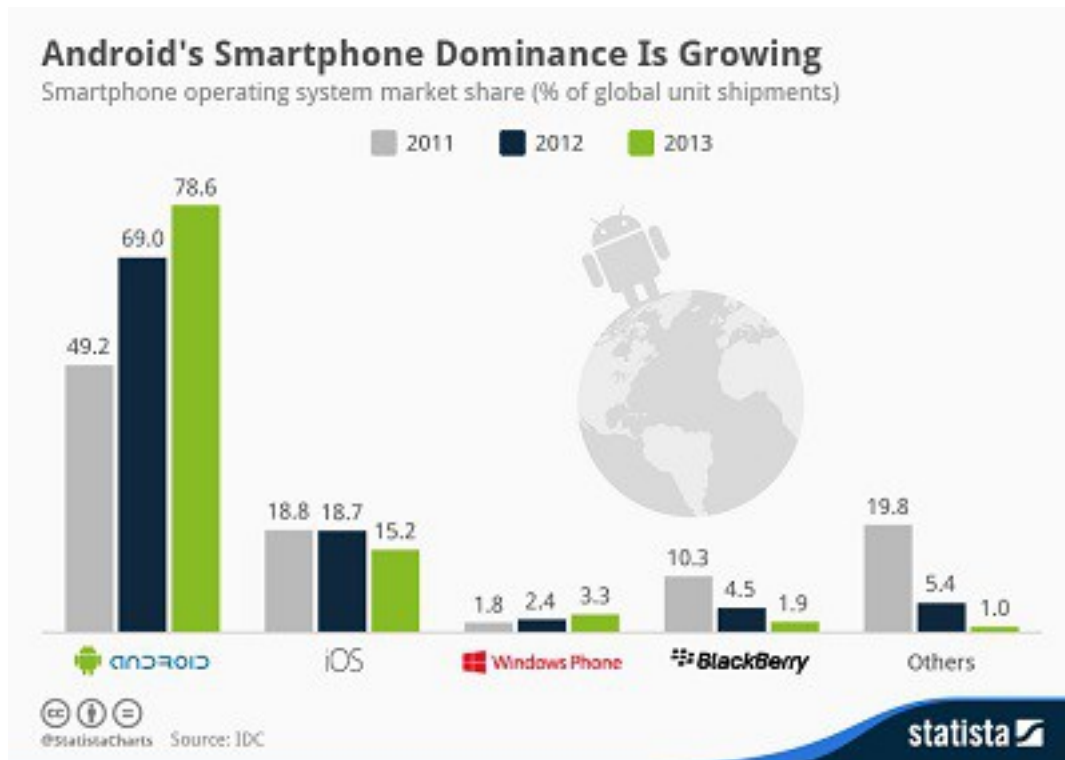
Πηγή: [2]



### 3 Υλοποίηση της εφαρμογής.

#### 3.1 Τεκμηρίωση επιλογών για την υλοποίηση του Client της πλατφόρμας.

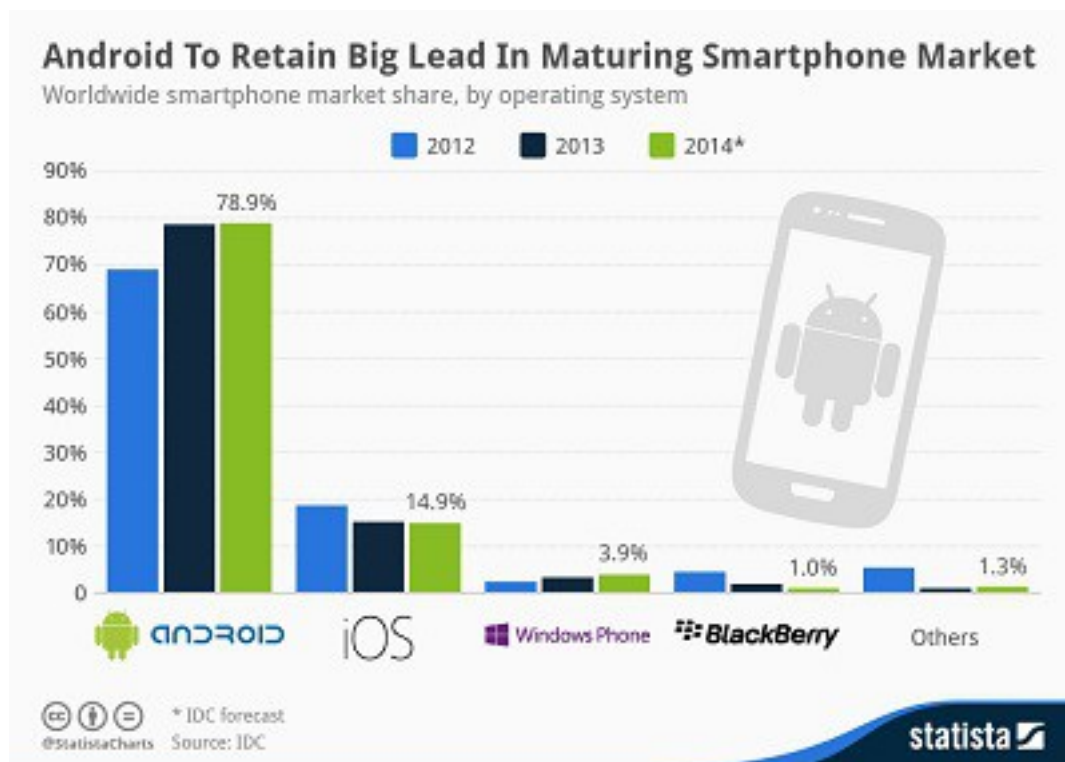
Καθώς υπάρχουν αρκετά λειτουργικά συστήματα για έξυπνα κινητά τηλέφωνα (smartphones), είχαμε την δυνατότητα να επιλέξουμε από ένα εύρος λειτουργικών για την υλοποίηση της εφαρμογής της πλατφόρμας. Οι κύριοι υποψήφιοι για τον σκοπό αυτό, ήταν τα λειτουργικά: Android, IOS, Windows Mobile και Blackberry. Μέσω σχετικής έρευνας που διέπραξε η IDC (International Data Corporation) (<http://www.idc.com/>) την οποία και εντοπίσαμε στις σελίδες [3] και [4], παρατηρήσαμε πως παγκοσμίως τα ινία στην αγορά των έξυπνων κινητών τηλεφώνων (smartphones), κατείχαν οι συσκευές Android καθ' όλη την διάρκεια των ερευνών με ποσοστό που ξεκίνησε στο 49% το 2011 και προβλέπεται να ανέρθει στο 79% το 2014, με τις συσκευές IOS να κατέχουν την δεύτερη θέση με τα ποσοστά τους να μην ξεπερνάνε το 20% και τις συσκευές Windows Mobile και Blackberry να βρίσκονται στην τρίτη και τέταρτη θέση αντιστοίχως. Βασιζόμενοι σε αυτά τα στατιστικά στοιχεία που παρουσιάζουν τις συσκευές Android ως τις πλέον δημοφιλέστερες συσκευές της αγοράς, την πληθώρα υποστηρικτικού υλικού για προγραμματιστές, καθώς και το γεγονός ότι κατείχαμε προηγούμενη εμπειρία στην ανάπτυξη λογισμικού για συσκευές Android, επιλέξαμε τις συσκευές αυτές για να αναπτύξουμε την εφαρμογή της πλατφόρμας μετρήσεων που υλοποιήσαμε.



Εικόνα 3.1: Έρευνα ποσοστού αγοράς smartphone βάση λειτουργικού 2011-2013

Πηγή <http://www.statista.com/chart/1899/smartphone-market-share/>

Αρχική προέλευση IDC (International Data Corporation)



Εικόνα 3.2: Έρευνα ποσοστού αγοράς smartphone βάση λειτουργικού 2012-2014\*

Πηγή <http://www.statista.com/chart/1961/smartphone-market-share-2014/>

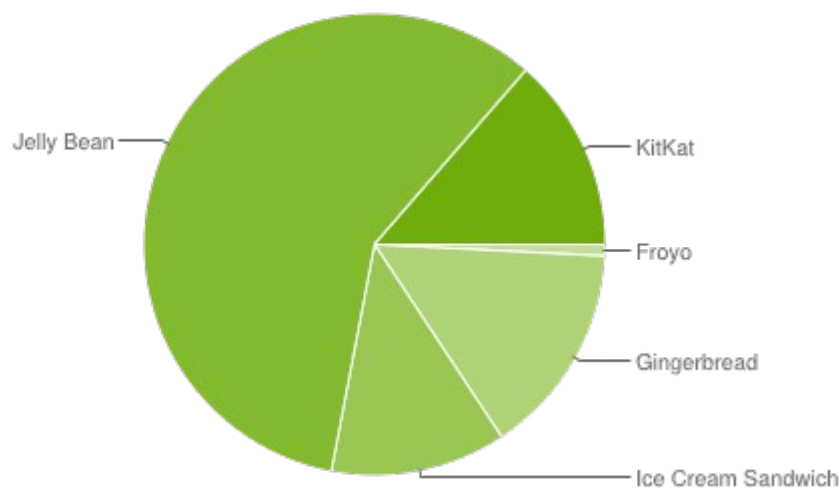
Αρχική προέλευση IDC (International Data Corporation)

\* Τα ποσοστά του 2014 αποτελούν προβλέψεις.



Το Android [5], αποτελεί ένα λειτουργικό σύστημα, κατά πλειοψηφία ανοιχτού κώδικα, για έξυπνες κινητές συσκευές (smartphones), βασισμένο σε Linux πυρήνα (Kernel). Αναπτύχθηκε αρχικά από την Google και εν συνεχεία από την Open Handset Alliance (<http://www.openhandsetalliance.com/>). Η ανάπτυξη λογισμικού για την πλατφόρμα Android γίνεται με την χρήση της προγραμματιστικής γλώσσας Java, ενώ οι εφαρμογές εκτελούνται μέσω της εικονικής μηχανής Dalvik [6],[7] η οποία αποτελεί μία εξειδικευμένη εικονική μηχανή για χρήση σε κινητές συσκευές.

Επιλέξαμε να υλοποιήσουμε την εφαρμογή μας με στόχο την έκδοση Ice Cream Sandwich (API level 15) την οποία και διέθεταν περίπου 12.3% των συσκευών, ενώ καθώς και οι νεότερες εκδόσεις του Android (Jelly Bean, KitKat,...) υποστηρίζουν εφαρμογές υλοποιημένες για παλιότερες, συνολικά η εφαρμογή αυτή θα υποστηριζόταν από περίπου το 84.3 % των συσκευών στην αγορά [8].



Εικόνα 3.3: Αναπαράσταση χρήσης εκδόσεων Android σε συσκευές της αγοράς.

Δεδομένα ελήφθησαν σε μία περίοδο 7 ημερών με τελευταία μέρα τις 4/6/2014

Πηγή: [https://developer.android.com/about/dashboards/index.html?utm\\_source=ausdroid.net](https://developer.android.com/about/dashboards/index.html?utm_source=ausdroid.net)

Ακολουθεί πίνακας που δίνει μια πιο συγκεκριμένη ερμηνεία του διαγράμματος της εικόνας 3.3

| Version       | Codename           | API | Distribution |
|---------------|--------------------|-----|--------------|
| 2.2           | Froyo              | 8   | 0.8%         |
| 2.3.3 - 2.3.7 | Gingerbread        | 10  | 14.9%        |
| 4.0.3 - 4.0.4 | Ice Cream Sandwich | 15  | 12.3%        |
| 4.1.x         | Jelly Bean         | 16  | 29.0%        |
| 4.2.x         |                    | 17  | 19.1%        |
| 4.3           |                    | 18  | 10.3%        |
| 4.4           | KitKat             | 19  | 13.6%        |

Για την ανάπτυξη του Android πελάτη (client) λογισμικού, επιλέξαμε να χρησιμοποιήσουμε το ολοκληρωμένο περιβάλλον ανάπτυξης Eclipse (<http://www.eclipse.org/>). Το Eclipse είναι ένα δωρεάν λογισμικό ανοιχτού κώδικα, γραμμένο στην προγραμματιστική γλώσσα Java, με επεκτάσιμο σύστημα Plug-in που προσφέρει ένα πλήρες προσαρμόσιμο περιβάλλον με δυνατότητα χρήσης για την ανάπτυξη εφαρμογών σε μια μεγάλη ποικιλία γλωσσών. Δημιουργημένο για μεγάλη ευελιξία, με την προσθήκη ενός Plug-in ,το Eclipse μετατρέπεται σε ένα ικανό περιβάλλον ανάπτυξης λογισμικού για Android συσκευές.

Επιλέξαμε να χρησιμοποιήσουμε το Eclipse ως περιβάλλον ανάπτυξης του Android client στα πλαίσια της παρούσας πτυχιακής, καθώς κατείχαμε σχετική οικειότητα με την χρήση του περιβάλλοντος αυτού για ανάπτυξη εφαρμογών για το εν λόγω λειτουργικό. Ένας άλλος λόγος για τον οποίο πήραμε αυτή την απόφαση είναι ότι συγκριτικά με άλλα περιβάλλοντα ανάπτυξης Android εφαρμογών, το Eclipse σε συνδυασμό με το Android Development Tools Plug-in, προσφέρει ένα σταθερό και δοκιμασμένο περιβάλλον με πλούσιο υλικό τεκμηρίωσης (documentation), ενώ το ανερχόμενο εργαλείο της Google, Android studio αν και δημιουργήθηκε αποκλειστικά για την ανάπτυξη λογισμικού για Android συσκευές, βρίσκεται ακόμα σε δοκιμαστική φάση με πολλές από τις λειτουργίες του να είναι είτε ατελείς, είτε μη υλοποιημένες. [9], [10], [11].

Ένα σημαντικό κομμάτι της εφαρμογής, πέρα από τον μηχανισμό λήψης των μετρήσεων, είναι η απεικόνιση αυτών σε χάρτες. Για τον σκοπό αυτό, επιλέξαμε να χρησιμοποιήσουμε τους χάρτες της Google (Google Maps API v2) [12]. Αν και η εγκατάσταση των χαρτών καθώς και η διαδικασία ενεργοποίησης τους απαιτήσε λίγη προσπάθεια, άξιζε τον κόπο, καθώς καταλήξαμε

με ένα φιλικό και αξιόπιστο προς τον χρήστη περιβάλλον με ποικιλία στα είδη χαρτών.

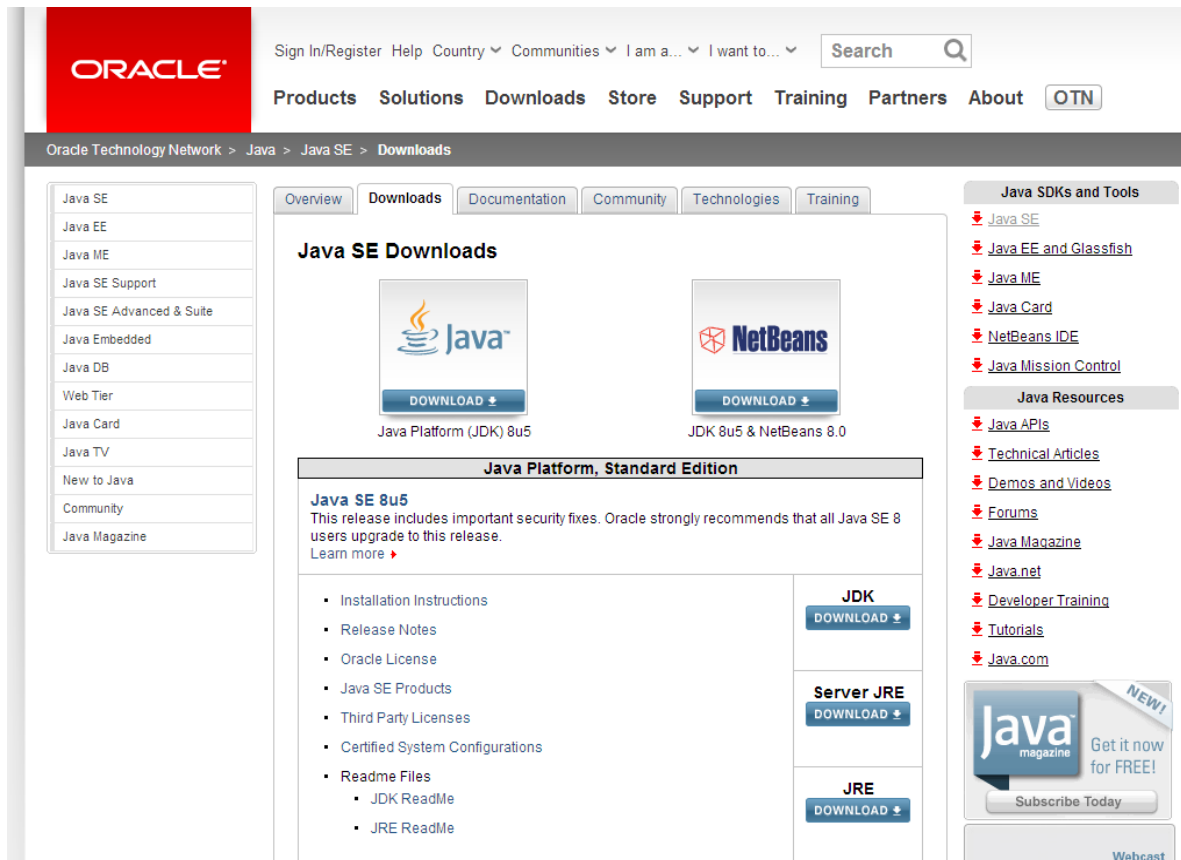
Η πτυχιακή εργασία υλοποιήθηκε σε φορητό ηλεκτρονικό υπολογιστή με εγκατεστημένο λειτουργικό σύστημα Windows 7 64-bit.

### 3.2 Εγκατάσταση Eclipse και παραμετροποίηση για ανάπτυξη λογισμικού Android

Για να ξεκινήσουμε την όποια ανάπτυξη κώδικα για Android στο λειτουργικό σύστημα της επιλογής μας, αρχικά έπρεπε να εγκαταστήσουμε κάποια πακέτα λογισμικού που χρησιμοποιούνται από τα εργαλεία ανάπτυξης της εφαρμογής.

#### Java Development Kit Download

Καθώς η δημιουργία εφαρμογών για Android συσκευές, βασίζεται σε μεγάλο βαθμό στην προγραμματιστική γλώσσα Java, απαιτείται η εγκατάσταση του περιβάλλοντος ανάπτυξης (Java SE Development Kit) το οποίο βρήκαμε στον ακόλουθο σύνδεσμο [Java SE Development Kit](http://www.oracle.com/technetwork/java/javase/downloads/index.html). Για την εγκατάσταση αυτού, μεταβήκαμε στον προαναφερθέν σύνδεσμο



Εικόνα 3.4: Επιλογή του Java Development Kit προς λήψη.

Πηγή <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

και επιλέξαμε να κατεβάσουμε το JDK(Download Java Platform (JDK)).

Έπειτα μεταφερθήκαμε σε σελίδα στην οποία, αφού πρώτα αποδεχτήκαμε την άδεια χρήσης του λογισμικού, κληθήκαμε να επιλέξουμε την κατάλληλη έκδοση του λογισμικού για το λειτουργικό μας σύστημα.

**Oracle Technology Network > Java > Java SE > Downloads**

**Java SE Development Kit 8 Downloads**

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- Java Developer Newsletter (tick the checkbox under Subscription Center > Oracle Technology News)
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK MD5 Checksum

**Looking for JDK 8 on ARM?**  
JDK 8 for ARM downloads have moved to the JDK 8 for ARM download page.

**Java SE Development Kit 8u5**

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

| Product / File Description          | File Size | Download                                       |
|-------------------------------------|-----------|--|
| Linux x86                           | 133.58 MB | <a href="#">jdk-8u5-linux-i586.rpm</a>         |
| Linux x86                           | 152.5 MB  | <a href="#">jdk-8u5-linux-i586.tar.gz</a>      |
| Linux x64                           | 133.87 MB | <a href="#">jdk-8u5-linux-x64.rpm</a>          |
| Linux x64                           | 151.64 MB | <a href="#">jdk-8u5-linux-x64.tar.gz</a>       |
| Mac OS X x64                        | 207.79 MB | <a href="#">jdk-8u5-macosx-x64.dmg</a>         |
| Solaris SPARC 64-bit (SVR4 package) | 135.68 MB | <a href="#">jdk-8u5-solaris-sparcv9.tar.Z</a>  |
| Solaris SPARC 64-bit                | 95.54 MB  | <a href="#">jdk-8u5-solaris-sparcv9.tar.gz</a> |
| Solaris x64 (SVR4 package)          | 135.9 MB  | <a href="#">jdk-8u5-solaris-x64.tar.Z</a>      |
| Solaris x64                         | 93.19 MB  | <a href="#">jdk-8u5-solaris-x64.tar.gz</a>     |
| Windows x86                         | 151.71 MB | <a href="#">jdk-8u5-windows-i586.exe</a>       |
| Windows x64                         | 155.18 MB | <a href="#">jdk-8u5-windows-x64.exe</a>        |

Εικόνα 3.5: Επιλογή κατάλληλου λογισμικού για την πλατφόρμα μας

Πηγή <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

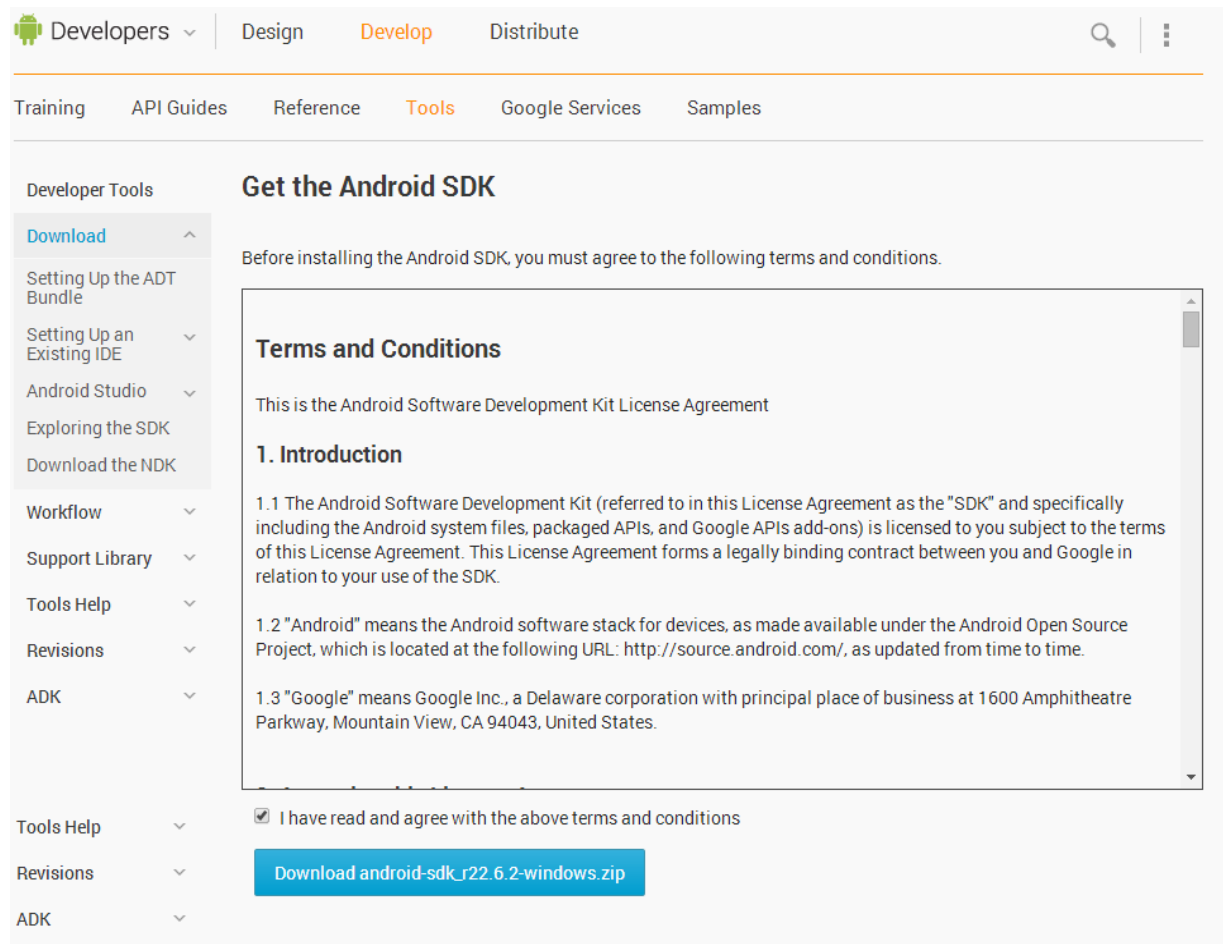
Καθώς η παρούσα πτυχιακή υλοποιήθηκε σε υπολογιστικό μηχάνημα που διέθετε εγκατεστημένο λειτουργικό Windows 64-bit, επιλέξαμε να κατεβάσουμε το πακέτο με την ανάλογη ένδειξη.

Στη συνέχεια, αφότου κατεβάσαμε το εκτελέσιμο αρχείο στον υπολογιστή μας το εκτελέσαμε και ακολουθώντας τις απαραίτητες ενέργειες στο παράθυρο που ανοίγει εγκαταστήσαμε το λογισμικό.



και στην συνέχεια στην κατηγορία SDK Tools Only, το συμπιεσμένο (.zip) πακέτο (package) που αντιστοιχεί στο λειτουργικό μας σύστημα.

Η επιλογή αυτή μας οδήγησε σε ένα παράθυρο στο οποίο έπρεπε να αποδεχτούμε τους όρους χρήσης του λογισμικού για να προχωρήσουμε με την λήψη.



Εικόνα 3.7: Αποδοχή όρων χρήσης και λήψη συμπιεσμένου αρχείου SDK

Πηγή <http://developer.android.com/sdk/index.html#download>

Αποδεχόμενοι τους όρους χρήσης, ξεκινήσαμε την λήψη του SDK συμπιεσμένου αρχείου για τα Windows. Στη συνέχεια, μετά την λήψη του αρχείου, το αποσυμπίεσαμε σε τυχαίο φάκελο, κρατώντας παράλληλα την τοποθεσία αυτή, καθώς μας χρειάστηκε παρακάτω.

### Λήψη πλατφόρμας Android για την ανάπτυξη λογισμικού.

Σε αυτό το σημείο, έπρεπε να κατεβάσουμε ένα σύνολο από πλατφόρμες (Εκδόσεις του Android λειτουργικού), μία από τις οποίες θα χρησιμοποιούνταν για την ανάπτυξη της εφαρμογής. Οπότε μεταβήκαμε στον φάκελο που αποσυμπίεσαμε το SDK (Standard Development Kit) και εκτελέσαμε το αρχείο "SDKManager.exe". Στην συνέχεια και αφότου άνοιξε ο SDKManager, επιλέξαμε τις εκδόσεις Android πλατφόρμας τις οποίες επιθυμούσαμε να κατεβάσουμε, τα πακέτα της κατηγορίας Tools, το πακέτο Google Play services που βρισκόταν στην κατηγορία

Extras και επιλέξαμε install x packages. Έπειτα, για να ολοκληρωθεί η διαδικασία, αποδεχτήκαμε τις άδειες επιλέγοντας το Accept License και πατήσαμε το install κουμπί. Στην συνέχεια κατεβάσαμε το πρόγραμμα Eclipse στο οποίο και αναπτύξαμε την εφαρμογή.

### **Eclipse Download**

Το επόμενο βήμα ήταν να κατεβάσουμε το Eclipse Integrated Development Kit. Το Eclipse το βρήκαμε στον ακόλουθο σύνδεσμο <http://www.eclipse.org/downloads/>. Μεταβήκαμε μέσω αυτού σε σελίδα στην οποία απεικονίζονταν πιθανές διανομές και επεκτάσεις του Eclipse που διατίθεντο προς λήψη. Η πρώτη επιλογή σε αυτήν την λίστα, ήταν η νεώτερη διανομή του λογισμικού. Εντοπίσαμε την διανομή που θα μεταφορτώνουμε στον υπολογιστή μας και επιλέξαμε την κατάλληλη έκδοση για το λειτουργικό μας σύστημα, μέσω ενός επιλογέα που βρισκόταν πάνω δεξιά. Επιλέξαμε την διανομή για Windows και στην συνέχεια βάση της αρχιτεκτονικής του μηχανήματος μας, επιλέξαμε τον σύνδεσμο Windows 64-bit. Έπειτα μέσω ενός από τους mirror συνδέσμους κατεβάσαμε σε συμπιεσμένη μορφή το Eclipse. Μετά την λήψη, μέσω ενός λογισμικού αποσυμπίεσης, αποσυμπίεσαμε το (.zip) αρχείο σε έναν νέο φάκελο με όνομα eclipse τον οποίο και τοποθετήσαμε στον φάκελο "C:\". Να σημειωθεί εδώ, ότι η έκδοση του Eclipse που κατεβάσαμε εμείς, είναι η Kepler Service Release 2. Επίσης, την πρώτη φορά που ενεργοποιείται το Eclipse, είναι απαραίτητο να εισάγουμε την διαδρομή προς τον φάκελο χώρο εργασίας (workspace) στον οποίο αποθηκεύεται λογισμικό που υλοποιούμε με το πρόγραμμα αυτό. Στην δικιά μας περίπτωση, τοποθετήσαμε το φάκελο αυτό μέσα στον φάκελο του χρήστη του υπολογιστικού μας συστήματος και επιλέξαμε να τον χρησιμοποιήσουμε σαν προεπιλεγμένο χώρο εργασίας.

### **Ρύθμιση Eclipse για ανάπτυξη εφαρμογών για Android**

Έχοντας κατεβάσει τα εργαλεία μας, έμενε μόνο να συνδέσουμε το Eclipse με το Android Standard Development Kit, κατεβάζοντας πρώτα το Android Development Tools plugin. Ανοίξαμε το Eclipse και επιλέξαμε το Install new software που βρισκόταν στην επιλογή βοήθεια (help) του menu. Στο Install παράθυρο επιλέξαμε το κουμπί Add και εισάγαμε ένα τυχαίο όνομα (για παράδειγμα ADT Plug-in) και στην περιοχή με την υπόδειξη Location τον ακόλουθο σύνδεσμο <https://dl-ssl.google.com/android/eclipse/>. Πατώντας το πλήκτρο OK, εμφανίστηκαν περιεχόμενα προς λήψη στο παράθυρο Install, επιλέξαμε το checkbox δίπλα στο Developer Tools και πατήσαμε Next. Διαβάσαμε και αποδεχτήκαμε τις άδειες χρήσης και πατήσαμε Finish. Με την ολοκλήρωση της εγκατάστασης επανεκινήσαμε το Eclipse. Όταν ανοίξε το Eclipse, έπρεπε να καθορίσουμε την τοποθεσία του Android Standard Development Kit, οπότε μεταφερθήκαμε στην επιλογή Window του menu του Eclipse και στις επιλογές που



εμφανίστηκαν πατήσαμε την επιλογή Preferences. Στο παράθυρο Preferences, επιλέξαμε Android από την αριστερή λίστα και στην δεξιά επιφάνεια στο πεδίο SDK Location εισάγαμε το πλήρες μονοπάτι προς τον φάκελο στον οποίο αποθηκεύσαμε το Android Standard Development Kit. Έτσι ολοκληρώθηκε η ρύθμιση του Eclipse και είμαστε έτοιμοι να ξεκινήσουμε την υλοποίηση του Android Client για την πλατφόρμα λήψης μετρήσεων.

### 3.3 Λεπτομέρειες υλοποίησης Android Client

Πριν αναλύσουμε τις μεθόδους που δημιουργήσαμε και χρησιμοποιήσαμε για την υλοποίηση των μηχανισμών μέτρησης των τεσσάρων Key Performance Indicators (Uplink ταχύτητα αποστολής, Downlink ταχύτητα λήψης, Ping our personal server, Ping Google Servers), θα αναφερθούμε στα βήματα που ακολουθήσαμε για να αποκτήσουμε πρόσβαση στους χάρτες της Google (Google Maps), μέσω της δεύτερης έκδοσης της προγραμματιστικής διεπαφής (Google Maps API v2, η οποία είναι μία βιβλιοθήκη) που προσφέρεται στους προγραμματιστές (developers) μέσω των Google Play services. Για την διαδικασία αυτή, καθώς και για οποιαδήποτε διαδικασία αναφέρεται στην υλοποίηση και ανάπτυξη του Android client χρησιμοποιήθηκε το Eclipse IDE (Intergrated Development Environment) Kepler που κατεβάσαμε σε προηγούμενο στάδιο. **Να σημειωθεί εδώ, ότι για να λειτουργήσει η τελική εφαρμογή μας θα πρέπει να έχουμε εγκατεστημένα τα Google Play Services στην Android συσκευή μας, καθώς και ενεργοποιημένες τις υπηρεσίες τοποθεσίας Google.** Γενικώς η εφαρμογή μας επιχειρεί να ενημερώσει τον χρήστη αν κάποιες από τις δύο προαναφερθέν παράμετροι δεν λειτουργούν σωστά.

#### Προσθήκη της google play services βιβλιοθήκης στην εφαρμογή μας

Το πρώτο βήμα της διαδικασίας προσθήκης των Google Maps στην εφαρμογή μας, ήταν η δημιουργία ενός νέου Android application με Project και Application name **LBSpeedAndroid** και όνομα πακέτου **com.koros.speedtester**, μέσω του Eclipse. Στην συνέχεια, χρειαστήκαμε την βιβλιοθήκη Google Play services. Για την εισαγωγή της βιβλιοθήκης, μεταβήκαμε στην επιλογή File του menu του Eclipse και στην συνέχεια επιλέξαμε την υποκατηγορία Import... . Καθώς η βιβλιοθήκη αυτή αποτελεί Android κώδικα, στο επόμενο παράθυρο επιλέξαμε να εισάγουμε ήδη υπάρχων Android κώδικα στον χώρο εργασίας(Existing Android Code Into Workspace) που εμφανίζεται όταν επιλέξουμε το εικονίδιο με την υπόδειξη Android και στην συνέχεια πατάμε το κουμπί με την ένδειξη Next. Έπειτα στο πεδίο Root Directory, εισάγαμε την διαδρομή (path) προς την βιβλιοθήκη, η οποία βρίσκεται στο φάκελο που αποθηκεύσαμε το Standard Development Kit του Android που κατεβάσαμε σε προηγούμενο βήμα και πιο συγκεκριμένα στον υποφάκελο “<sdk-path>\extras\google\google\_play\_services”. Ύστερα, εντοπίσαμε και επιλέξαμε την καταχώριση libproject\google-play-services\_lib στην επιφάνεια με την ένδειξη “Projects:”, ενεργοποιήσαμε την λειτουργία Copy project into workspace με σκοπό να αντιγραφεί η βιβλιοθήκη στον φάκελο workspace και τέλος πατήσαμε το κουμπί με την ένδειξη Finish. Τελικό βήμα στην διαδικασία αυτή αποτέλεσε η σύνδεση της Android εφαρμογής με την βιβλιοθήκη που εισάγαμε στο workspace. Κατόπιν, στον Package Explorer κάναμε δεξί κλικ στην εφαρμογή μας και επιλέξαμε Properties. Στο επόμενο παράθυρο μεταβήκαμε στην

κατηγορία Android και εν συνεχεία στην επιφάνεια με την ένδειξη Library. Πατώντας το κουμπί Add μεταβαίνουμε σένα παράθυρο με όλες τις βιβλιοθήκες που διαθέτουμε στο φάκελο workspace και συνεπώς μπορούμε να προσθέσουμε στην εφαρμογή μας. Στο σημείο αυτό εντοπίσαμε το google-play-services\_lib και πατήσαμε το κουμπί OK για να ολοκληρωθεί η διαδικασία προσθήκης της βιβλιοθήκης στην εφαρμογή μας.

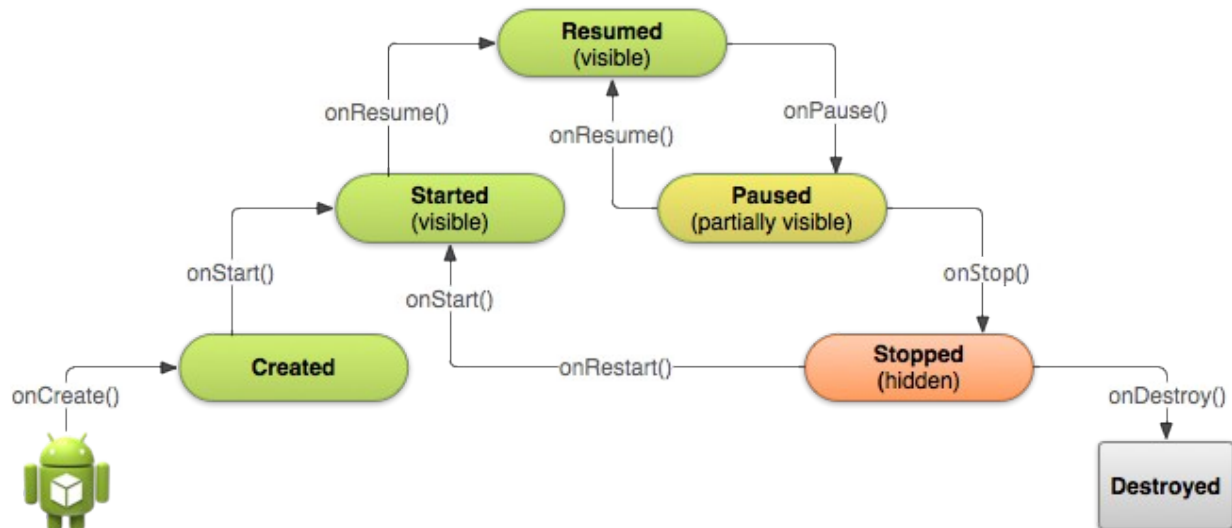
### **Σύνδεση της εφαρμογής μας με τους εξυπηρέτες της Google**

Έχοντας συνδέσει την εφαρμογή μας με την βιβλιοθήκη google play services, το επόμενο βήμα στην υλοποίηση της Android client εφαρμογής ήταν η απόκτηση πρόσβασης στους εξυπηρέτες των χαρτών της Google, δηλώνοντας την εφαρμογή μας στο Google API Console <https://console.developers.google.com/project>. Για να εισέλθουμε στο Google API Console, απαιτείται η χρήση των αναγνωριστικών (username, password) ενός λογαριασμού ηλεκτρονικού ταχυδρομείου (e-mail) της Google. Στην σελίδα που μεταβήκαμε μέσω του παραπάνω συνδέσμου, δημιουργήσαμε ένα νέο Project ονομάζοντάς το API Project 2. Ύστερα ενεργοποιήσαμε την επιλογή Google Maps Android API V2 και στην επόμενη σελίδα επιλέξαμε Create New Android Key. Στην συνέχεια, στο παράθυρο διαλόγου που ανοίγει καλούμαστε να εισάγουμε το SHA1 fingerprint (αποτύπωμα) και το όνομα πακέτου της εφαρμογής μας, χωρίζοντας τα με τον χαρακτήρα ';'. Το αποτύπωμα αυτό είναι ένα μοναδικό σύνολο από χαρακτήρες που προκύπτει από τον αλγόριθμο SHA-1, ο οποίος παίρνει σαν είσοδο ένα πιστοποιητικό που χρησιμοποιεί ο προγραμματιστής για να υπογράψει ψηφιακά την εφαρμογή του. Στην περίπτωση της παρούσας πτυχιακής, χρησιμοποιήθηκε το Debug keystore καθώς δεν κρίθηκε απαραίτητο προσθήκη της εφαρμογής στο Play Store. Καθώς το SHA-1 αποτύπωμα είναι μοναδικό, οι εξυπηρέτες της Google το χρησιμοποιούν ως μία μορφή αναγνωριστικού από όλες τις εφαρμογές που χρησιμοποιούν τις υπηρεσίες που προσφέρουν. Για τον εντοπισμό του SHA 1 fingerprint μεταβήκαμε στην επιλογή Window στο Eclipse και στην συνέχεια στο Preferences. Έπειτα στο επόμενο παράθυρο, πατήσαμε Android, Build και εντοπίσαμε το SHA 1 fingerprint. Αφού εισάγαμε τις πληροφορίες που μας ζητήθηκαν (SHA-1 fingerprint ; Application package name), η κονσόλα μας επιστρέφει το κλειδί που θα χρησιμοποιήσει η εφαρμογή μας για την επικοινωνία της με τον εξυπηρέτη (server) της Google, το οποίο και εισάγαμε στο Android Manifest αρχείο της εφαρμογής μας.

### **Υλοποίηση της εφαρμογής**

Κάθε εφαρμογή στο περιβάλλον Android αποτελείται από δραστηριότητες (Activities) που θεωρούνται τα βασικά δομικά συστατικά μίας εφαρμογής. Κάθε εφαρμογή διαθέτει μία βασική δραστηριότητα (main Activity) η οποία καλείτε κατά την εκκίνηση της εκάστοτε εφαρμογής.

Κάθε δραστηριότητα διαθέτει την δική της γραφική διεπαφή και είναι υπεύθυνη τόσο για την διαχείριση αυτής, όσο και για την διαχείριση του κύκλου ζωής της. Σε γενικές γραμμές ο κύκλος ζωής μίας δραστηριότητας περιγράφεται από την ακόλουθη εικόνα:



Εικόνα 3.8: Κύκλος ζωής Ενεργειών (Activities) σε περιβάλλον Android

Πηγή <http://developer.android.com/training/basics/activity-lifecycle/pausing.html>

Για την υλοποίηση της εφαρμογής **LBSpeedAndroid**:

1. Δημιουργήσαμε τις παρακάτω κλάσεις (classes) για την ανάπτυξη της λειτουργικότητας της εφαρμογής:

#### **MainActivity:**

Επεκτείνει την κλάση `FragmentActivity` για την διαχείριση του αντικείμενου των χαρτών της Google και υλοποιεί την `OnItemSelectedListener` για την . Η κλάση που καλείται κατά την εκκίνηση της εφαρμογής μας και σε αυτήν υλοποιήθηκαν λειτουργίες που διαχειρίζονται γεγονότα σχετικά με τον κύκλο ζωής της βασικής δραστηριότητας (Main Activity) στο περιβάλλον Android, όπως `onCreate()`, `onPause()`, `onRestart()`, `onResume()`. Η `onCreate()` μέθοδος καλείται κατά την εκκίνηση και επομένως σε αυτήν ρυθμίζουμε το γραφικό περιβάλλον και γενικότερα τα επιμέρους τμήματα της εφαρμογής. Η `onPause()` μέθοδος καλείται όταν η δραστηριότητα (Activity) βρίσκεται στο παρασκήνιο και μεταβαίνει στην κατάσταση παύσης (pause). Η `onResume()` μέθοδος εκτελείται όταν η δραστηριότητα (Activity) μεταβαίνει από την κατάσταση παύσης στην οποία βρίσκεται όταν βρίσκεται στο παρασκήνιο σε ενεργή κατάσταση στο προσκήνιο, όταν δηλαδή δεν καλύπτεται από άλλες δραστηριότητες (Activities). Η `onRestart()` μέθοδος εκτελείται σε περίπτωση που η δραστηριότητα (Activity) έχει σταματήσει και επιχειρεί να ξεκινήσει πάλι. Προστέθηκαν έλεγχοι τόσο για την ύπαρξη των Google Play Services όσο και για

την κατάσταση των υπηρεσιών τοποθεσίας (location services) στην μέθοδο startTester η οποία εκκινεί την διαδικασία λήψης των μετρήσεων όταν επιλεγεί το αντίστοιχο κουμπί στην τρίτη καρτέλα (tab) με τίτλο Test Connection της εφαρμογής. Επίσης στην κλάση αυτή υλοποιήθηκαν μέθοδοι που αφορούν την ρύθμιση και λειτουργία του γραφικού περιβάλλοντος της εφαρμογής, καθώς και ανάκτηση της διεύθυνσης IP(Internet Protocol) του μηχανήματος στο οποίο βρίσκονται οι εξυπηρετές της εφαρμογής μας.

Στο **Appendix A: Κώδικας MainActivity Κλάσης** μπορεί να βρεθεί ο κώδικας της MainActivity κλάσης πλήρως σχολιασμένος.

### NetworkUtility:

Προσφέρει πληροφορίες τόσο στο γραφικό περιβάλλον, όσο και στην κλάση NetworkChangeReceiver σχετικά με την σύνδεση μας στο διαδίκτυο(wifi, mobil, no connection).

### Κώδικας της NetworkUtility κλάσης:

```
package com.koros.speedtester;
import android.content.Context;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.net.wifi.WifiInfo;
import android.net.wifi.WifiManager;
import android.telephony.TelephonyManager;
import android.text.format.Formatter;
/*The following Class was based upon this guide
http://viralpatel.net/blogs/android-internet-connection-status-network-change/ */
/*This class as well as the NetworkChangeReceiver are used to check our Internet
connection status, that is if we are connected and what means we use for this
connection */

public class NetworkUtility {
    public static int TYPE_WIFI = 1;
    public static int TYPE_MOBILE = 2;
    public static int TYPE_NOT_CONNECTED = 0;
    static String info=null;
    public static int getConnectivityStatus(Context context) {
        ConnectivityManager cm = (ConnectivityManager) context
            .getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo activeNetwork = cm.getActiveNetworkInfo();
        if (activeNetwork!=null) {
            if(activeNetwork.getType() == ConnectivityManager.TYPE_WIFI) {
                return TYPE_WIFI;
            }
            if(activeNetwork.getType() == ConnectivityManager.TYPE_MOBILE) {
                return TYPE_MOBILE;
            }
        }
        return TYPE_NOT_CONNECTED;
    }
    @SuppressWarnings("deprecation")
    public static String getConnectivityStatusString(Context context) {
        int conn = NetworkUtility.getConnectivityStatus(context);
        String status = null;
        //If wifi connection Present
        if (conn == NetworkUtility.TYPE_WIFI) {
            WifiManager wfm=(WifiManager)context
                .getSystemService(Context.WIFI_SERVICE);
            WifiInfo wfi=wfm.getConnectionInfo();
            //saving connection type (wifi) to send to server
        }
    }
}
```

```

SpeedTester.connectionType="wifi";
status = "Wifi enabled\nSSID: "+wfi.getSSID()+"\nIP: "+Formatter.formatIpAddress(wfi.getIpAddress())
+"\n"+WifiInfo.LINK_SPEED_UNITS+": "+wfi.getLinkSpeed();
//If mobile connection present
} else if (conn == NetworkUtility.TYPE_MOBILE) {
    TelephonyManager tm=(TelephonyManager)context
    .getSystemService(Context.TELEPHONY_SERVICE);
    //saving specific connection type (mobile) to send to server
    SpeedTester.connectionType="mobile";
    switch(tm.getNetworkType()) {
        case TelephonyManager.NETWORK_TYPE_1xRTT:
            SpeedTester.connectionType="mobile: 1xRTT";
            break;
        case TelephonyManager.NETWORK_TYPE_CDMA:
            SpeedTester.connectionType="mobile: CDMA";
            break;
        case TelephonyManager.NETWORK_TYPE_EDGE:
            SpeedTester.connectionType="mobile: EDGE";
            break;
        case TelephonyManager.NETWORK_TYPE_EHRPD:
            SpeedTester.connectionType="mobile: EHRPD";
            break;
        case TelephonyManager.NETWORK_TYPE_EVDO_0:
            SpeedTester.connectionType="mobile: EVDO_0";
            break;
        case TelephonyManager.NETWORK_TYPE_EVDO_A:
            SpeedTester.connectionType="mobile: EVDO_A";
            break;
        case TelephonyManager.NETWORK_TYPE_EVDO_B:
            SpeedTester.connectionType="mobile: EVDO_B";
            break;
        case TelephonyManager.NETWORK_TYPE_GPRS:
            SpeedTester.connectionType="mobile: GPRS";
            break;
        case TelephonyManager.NETWORK_TYPE_HSDPA:
            SpeedTester.connectionType="mobile: HSDPA";
            break;
        case TelephonyManager.NETWORK_TYPE_HSPA:
            SpeedTester.connectionType="mobile: HSPA";
            break;
        case TelephonyManager.NETWORK_TYPE_HSPAP:
            SpeedTester.connectionType="mobile: HSPAP";
            break;
        case TelephonyManager.NETWORK_TYPE_HSUPA:
            SpeedTester.connectionType="mobile: HSUPA";
            break;
        case TelephonyManager.NETWORK_TYPE_IDEN:
            SpeedTester.connectionType="mobile: IDEN";
            break;
        case TelephonyManager.NETWORK_TYPE_LTE:
            SpeedTester.connectionType="mobile: LTE";
            break;
        case TelephonyManager.NETWORK_TYPE_UMTS:
            SpeedTester.connectionType="mobile: UMTS";
            break;
        case TelephonyManager.NETWORK_TYPE_UNKNOWN:
            SpeedTester.connectionType="mobile: UNKNOWN";
            break;
    }
    //Getting the carrier name
    status = "Mobile data enabled\nOperator name: "
    +tm.getNetworkOperatorName();
    //If no connection present
} else if (conn == NetworkUtility.TYPE_NOT_CONNECTED) {
    status = "Not connected to Internet";
}
return status;
}
}

```

### NetworkChangeReceiver:

Επεκτείνει την κλάση **BroadcastReceiver**. Παίρνει πληροφορίες από την

**NetworkUtility** κλάση εκτελώντας περαιτέρω ελέγχους όσον αφορά την επικοινωνία με το διαδίκτυο και ανάλογα με το αποτέλεσμα επηρεάζει το γραφικό περιβάλλον, απενεργοποιώντας πλήκτρα που εκτελούν δικτυακές λειτουργίες όταν δεν υπάρχει κάποια σύνδεση ή όταν δεν είναι ενεργοποιημένες οι υπηρεσίες τοποθεσίας (location services).

### Κώδικας της NetworkChangeReceiver κλάσης:

```
package com.koros.speedtester;
import com.google.android.gms.common.ConnectionResult;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.graphics.Color;
import android.location.LocationManager;
import android.net.ConnectivityManager;
import android.widget.ProgressBar;
/*The following Class was based upon this guide http://viralpatel.net/blogs/android-
internet-connection-status-network-change/ */
/*This class as well as the NetworkUtility are used to check our Internet connection
status, that is if we are connected and what means we use for this connection */
public class NetworkChangeReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(final Context context, final Intent intent) {
        final ConnectivityManager connMgr = (ConnectivityManager)
            context.getSystemService(Context.CONNECTIVITY_SERVICE);
        final android.net.NetworkInfo wifi =
            connMgr.getNetworkInfo(ConnectivityManager.TYPE_WIFI);
        final android.net.NetworkInfo mobile =
            connMgr.getNetworkInfo(ConnectivityManager.TYPE_MOBILE);
        LocationManager manager = (LocationManager)
            MainActivity.alpha.getSystemService(Context.LOCATION_SERVICE);
        String status = NetworkUtility.getConnectivityStatusString(context);
        MainActivity.connection.setText(status);
        MainActivity.connection.invalidate();
        if((wifi.isConnected()) || (mobile.isConnected())){//If either WIFI or mobile
            Internet connection is present
            if(!MainActivity.testUnderway){//If test is underway do not alter the GUI
                MainActivity.TestStart.setClickable(true);//Enables test starting
                    button when connection is present
                MainActivity.TestStart.setBackgroundColor(Color.LTGRAY);
                MainActivity.TestStart.setTextColor(Color.BLACK);
                MainActivity.TestStart.setText("Start Test");
                MainActivity.TestStart.invalidate();
                MainActivity.status.setTextColor(Color.BLACK);
                MainActivity.status.setText("Test Inactive");
                MainActivity.downlink.setBackgroundColor(Color.TRANSPARENT);
                MainActivity.downlink.invalidate();
                MainActivity.uplink.setBackgroundColor(Color.TRANSPARENT);
                MainActivity.uplink.invalidate();
                MainActivity.ping1.setBackgroundColor(Color.TRANSPARENT);
                MainActivity.ping1.invalidate();
                MainActivity.ping2.setBackgroundColor(Color.TRANSPARENT);
                MainActivity.ping2.invalidate();
                MainActivity.pb.setVisibility(ProgressBar.GONE);
                MainActivity.pb.invalidate();
            }
        }else{//If neither WIFI nor mobile Internet connection is present, Set the GUI
            appearance to display the lack of connection.
            MainActivity.TestStart.setClickable(false);//disable test starting button
                to avoid exceptions
            MainActivity.TestStart.setBackgroundColor(Color.TRANSPARENT);
            MainActivity.TestStart.setTextColor(Color.RED);
            MainActivity.TestStart.setText("No Internet Connection");
            MainActivity.status.setText("Terminating Test!");
            MainActivity.status.setTextColor(Color.RED);
            MainActivity.status.setBackgroundColor(Color.TRANSPARENT);
            MainActivity.status2.setBackgroundColor(Color.TRANSPARENT);
            MainActivity.ping1.setBackgroundColor(Color.TRANSPARENT);
            MainActivity.ping2.setBackgroundColor(Color.TRANSPARENT);
            MainActivity.downlink.setBackgroundColor(Color.TRANSPARENT);
            MainActivity.uplink.setBackgroundColor(Color.TRANSPARENT);
        }
    }
}
```

```

        MainActivity.pb.setVisibility(ProgressBar.INVISIBLE);
        MainActivity.pb.invalidate();
        MainActivity.TestStart.invalidate();
    }
    if(!manager.isProviderEnabled(LocationManager.NETWORK_PROVIDER) && !
MainActivity.LocationErrorMessage){//If network location provider is disable and no
other messages regarding this event are present (the
MainActivity.LocationErrorMessage variable is used to indicate that a
message is displayed)
        MainActivity.LocationServicesNeededMessage();//then display a message that
sends the user to options to enable location services
    }
    if((!wifi.isConnected()) && (!mobile.isConnected()) && (!
manager.isProviderEnabled(LocationManager.GPS_PROVIDER) || (!
manager.isProviderEnabled(LocationManager.NETWORK_PROVIDER)))){// if no connection
is enables and no location provider is enabled
        if(MainActivity.GPSAvailable==ConnectionResult.SUCCESS){//And Google Play
Services are installed
            MainActivity.map.setMyLocationEnabled(false);//Disable my position button
        }
    }
    else{//else if Google Play Services are installed and there is at least one
location provider enabled
        if(MainActivity.GPSAvailable==ConnectionResult.SUCCESS&&
(manager.isProviderEnabled(LocationManager.GPS_PROVIDER) ||
(manager.isProviderEnabled(LocationManager.NETWORK_PROVIDER)))){
            MainActivity.map.setMyLocationEnabled(true);//Enable my position
button
        }
    }
}
}
}

```

### Ping:

Διαθέτει μεθόδους που προσομοιώνουν την λειτουργία του εργαλείου **Ping** παίρνοντας τέσσερεις μετρήσεις και εξάγοντας έναν μέσο όρο. Επιχειρήσαμε αρχικά να χρησιμοποιήσουμε το Ping εργαλείο που βρίσκεται σε κάθε Linux περιβάλλον, δεδομένου ότι το Android είναι βασισμένο σε Linux πυρήνα (Linux Kernel) αλλά, αν και εντοπίσαμε την λειτουργία, παρατηρήσαμε πως για να πραγματοποιηθούν ICMP (Internet Control Message Protocol) echo requests πακέτα χρειαζόμασταν δικαιώματα χρήστη root, τα οποία δεν είναι διαθέσιμα σε όλες τις συσκευές. Συνεπώς υλοποιήσαμε μια λειτουργία με ίδια λογική η οποία χρησιμοποιεί HTTP (Hypertext Transfer Protocol) requests (αιτήματα) και response (ανταποκρίσεις) και εξάγει ρεαλιστικά αποτελέσματα. Στην σύνδεση ενημερώνουμε τον εκάστοτε εξυπηρέτη ότι η εφαρμογή που συνδέεται είναι εφαρμογή Android καθώς και ότι επιθυμούμε η σύνδεση να κλίση μετά την σύνδεση.

### Κώδικας της Ping κλάσης:

```

package com.koros.speedtester;
import java.io.IOException;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
/*Class Ping implements a proper Ping operation*/
public class Ping {
    String privateServerAddress;
    String otherServerAddress;
    public Ping() {
        privateServerAddress=MainActivity.getDefaultServer();
        otherServerAddress="www.google.gr";
    }
}

```



```

    }
    private String ping(boolean local){
        URL url;
        HttpURLConnection urlc;
        double timeMean=0;
        try {
            for(int i=0;i<4;i++){
                url = new URL("http://" + (local?
                    privateServerAddress:otherServerAddress)); //URL of the server to
                    perform a ping
                urlc = (HttpURLConnection) url.openConnection();
                urlc.setRequestProperty("User-Agent", "Android Application:");
                urlc.setRequestProperty("Connection", "close");
                urlc.setConnectTimeout(1000 * 30); // Timeout is in seconds
                long time1=System.nanoTime(); //Takes time before connection
                    measured in nanoseconds
                urlc.connect(); //Connect to remote server
                long time2=System.nanoTime(); //Takes time after connection measured
                    in nanoseconds
                if (urlc.getResponseCode() == 200) { //Checks if response code was
                    OK (request was complete
                    urlc.disconnect(); //Closes connection
                    timeMean=((time2-time1)/1000000.0+timeMean)/2.0);
                    //Calculate the mean time of all 4 iterations and change
                        time to milliseconds
                }
            }
            return timeMean+""; //return mean time
        } catch (MalformedURLException el) {
            el.printStackTrace();
            return null;
        } catch (IOException e) {
            e.printStackTrace();
            return null;
        } catch (Exception e){
            return null;
        }
    }
    String pingRun(boolean local){
        return ping(local);
    }
}

```

### SpeedTester, ServerComm, RunTest:

Συνολικά, αυτές οι τρεις κλάσεις σε συνδυασμό με την Ping εμπεριέχουν το σύνολο των λειτουργιών που αποτελούν το τεστ της ταχύτητας σύνδεσης στο διαδίκτυο (Internet Speed Test). Η SpeedTester, η οποία ενθυλακώνει τις άλλες δύο (ServerComm, RunTest), καταγράφει την τοποθεσία του χρήστη είτε χρησιμοποιώντας λειτουργίες των χαρτών της Google, είτε αν υπάρχει κάποιο πρόβλημα με τις λειτουργίες αυτές, μέσω μεθόδων του Android, την ημερομηνία και ώρα εκκίνησης του τεστ και τέλος εκκινεί την RunTest κλάση η οποία επεκτείνει την κλάση AsyncTask (όλες οι δικτυακές ενέργειες δεν επιτρέπονται μέσα στο βασικό νήμα εκτέλεσης της εφαρμογής [13], οπότε εκτελούνται μέσα σε ξεχωριστό νήματα εκτέλεσης) και συνεπώς αποτελεί νήμα εκτέλεσης. Στην RunTest, υλοποιήθηκε η λειτουργία επικοινωνίας με τον Java εξυπηρέτη που αναπτύξαμε για την απόκτηση των μετρήσεων ταχύτητας αποστολής και λήψης (uplink and downlink). Κατά το τέλος της εκτέλεσης του, το RunTest νήματα εκκινεί ένα δεύτερο νήμα εκτέλεσης το ServerComm (η κλάση ServerComm επεκτείνει την AsyncTask οπότε συμπεριφέρεται και αυτή σαν νήμα), το οποίο και στέλνει τις μετρήσεις μαζί με τις

υπόλοιπες πληροφορίες που καταγράφονται (τύπος σύνδεσης στο διαδίκτυο, τύπος λειτουργικού συστήματος συσκευής) στην υπηρεσία web (web service) που υλοποιήσαμε και βρίσκεται στο εικονικό μηχάνημα όπου τρέχουν οι εξυπηρετές της πλατφόρμας μας.

Στο **Appendix B: Κώδικας SpeedTester, ServerComm και RunTest Κλάσεων** μπορεί να βρεθεί ο κώδικας των κλάσεων SpeedTester, ServerComm και RunTest Κλάσεων κλάσης πλήρως σχολιασμένος.

### **WMSTileProvider:**

Επεκτείνει την κλάση **UrlTileProvider** και προσφέρει υπηρεσίες στην κλάση **TileProviderFactory** εκτελώντας υπολογισμούς σχετικά με την αναπαράσταση των μετρήσεων που στέλνονται από τον Geoserver.

### **Κώδικας της WMSTileProvider κλάσης:**

```
package com.koros.speedtester;
import com.google.android.gms.maps.model.UrlTileProvider;
/*The following Class was based upon this guide
http://www.azavea.com/blogs/labs/2013/01/wms-on-android/ */
/*This class as well as the TileProviderFactory class are used to create and return a
TileProvider with the appropriate URL for the layer and style to be displayed by Google
Maps */
public abstract class WMSTileProvider extends UrlTileProvider {

    // Web Mercator n/w corner of the map.
    private static final double[] TILE_ORIGIN = {-20037508.34789244, 20037508.34789244};
    //array indexes for that data
    private static final int ORIG_X = 0;
    private static final int ORIG_Y = 1;

    // Size of square world map in meters, using WebMerc projection.
    private static final double MAP_SIZE = 20037508.34789244 * 2;

    // array indexes for array to hold bounding boxes.
    protected static final int MINX = 0;
    protected static final int MAXX = 1;
    protected static final int MINY = 2;
    protected static final int MAXY = 3;

    // Construct with tile size in pixels, normally 256, see parent class.
    public WMSTileProvider(int x, int y) {
        super(x, y);
    }

    // Return a web Mercator bounding box given tile x/y indexes and a zoom
    // level.
    protected double[] getBoundingBox(int x, int y, int zoom) {
        double tileSize = MAP_SIZE / Math.pow(2, zoom);
        double minx = TILE_ORIGIN[ORIG_X] + x * tileSize;
        double maxx = TILE_ORIGIN[ORIG_X] + (x+1) * tileSize;
        double miny = TILE_ORIGIN[ORIG_Y] - (y+1) * tileSize;
        double maxy = TILE_ORIGIN[ORIG_Y] - y * tileSize;

        double[] bb = new double[4];
        bb[MINX] = minx;
        bb[MINY] = miny;
        bb[MAXX] = maxx;
        bb[MAXY] = maxy;

        return bb;
    }
}
```

```
}
```

### TileProviderFactory:

Χρησιμοποιεί την κλάση **WMSTileProvider** και τις λειτουργίες της για την σωστή δόμηση τεσσάρων URL (Universal Resource Locator) που αποστέλλονται ως αίτημα στον Geoserver για την παραλαβή των οπτικοποιημένων μετρήσεων ταχύτητας αποστολής και λήψης (Uplink and Downlink) καθώς και των δυο ping χρόνων (Ping personal server, Ping Google Server)

### Κώδικας της TileProviderFactory κλάσης:

```
package com.koros.speedtester;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.Locale;
import com.google.android.gms.maps.model.TileProvider;
/*The following Class was based upon this guide
http://www.azavea.com/blogs/labs/2013/01/wms-on-android/ */
/*This class as well as the WMSTileProvider class are used to create and return a
TileProvider with the appropriate URL for the layer and style to be displayed by Google
Maps */
public class TileProviderFactory{
    static String GEOSERVER_FORMAT;
    private static final String GEOSERVER_FORMAT_DOWNLINK =
        "http://83.212.111.56:8080/geoserver/cite/wms" +
        "?service=WMS" +
        "&version=1.1.1" +
        "&request=GetMap" +
        "&layers=cite:measurements" +
        "&query_layers=cite:measurements" +
        "&styles=Downlink"+
        "&bbox=%f,%f,%f,%f" +
        "&width=%d" +
        "&height=%d" +
        "&srs=EPSG:900913" +
        "&format=image/png" +
        "&transparent=true";
    private static final String GEOSERVER_FORMAT_UPLINK =
        "http://83.212.111.56:8080/geoserver/cite/wms" +
        "?service=WMS" +
        "&version=1.1.1" +
        "&request=GetMap" +
        "&layers=cite:measurements" +
        "&query_layers=cite:measurements" +
        "&styles=Uplink"+
        "&bbox=%f,%f,%f,%f" +
        "&width=%d" +
        "&height=%d" +
        "&srs=EPSG:900913" +
        "&format=image/png" +
        "&transparent=true";
    private static final String GEOSERVER_FORMAT_PING1 =
        "http://83.212.111.56:8080/geoserver/cite/wms" +
        "?service=WMS" +
        "&version=1.1.1" +
        "&request=GetMap" +
        "&layers=cite:measurements" +
        "&query_layers=cite:measurements" +
        "&styles=Ping1"+
        "&bbox=%f,%f,%f,%f" +
        "&width=%d" +
        "&height=%d" +
        "&srs=EPSG:900913" +
        "&format=image/png" +
        "&transparent=true";
    private static final String GEOSERVER_FORMAT_PING2 =
        "http://83.212.111.56:8080/geoserver/cite/wms" +
        "?service=WMS" +
        "&version=1.1.1" +
        "&request=GetMap" +
```

```

        "&layers=cite:measurements" +
        "&query_layers=cite:measurements" +
        "&styles=Ping2"+
        "&bbox=%f,%f,%f,%f" +
        "&width=%d" +
        "&height=%d" +
        "&srs=EPSG:900913" +
        "&format=image/png" +
        "&transparent=true";

// return a geoserver wms tile layer
static TileProvider getTileProvider(String tileType) {

    if(tileType.equalsIgnoreCase("Downlink")){
        GEOSERVER_FORMAT=GEOSERVER_FORMAT_DOWNLINK;
    }
    else if(tileType.equalsIgnoreCase("Uplink")){
        GEOSERVER_FORMAT=GEOSERVER_FORMAT_UPLINK;
    }
    else if(tileType.equalsIgnoreCase("Ping1")){
        GEOSERVER_FORMAT=GEOSERVER_FORMAT_PING1;
    }
    else if(tileType.equalsIgnoreCase("Ping2")){
        GEOSERVER_FORMAT=GEOSERVER_FORMAT_PING2;
    }

    TileProvider tileProvider = new WMSTileProvider(256,256) {

        @Override
        public synchronized URL getTileUrl(int x, int y, int zoom) {
            double[] bb = getBoundingBox(x, y, zoom);

            String s = String.format(Locale.US, TileProviderFactory.GEOSERVER_FORMAT,
bb[MiNX],bb[MiNY], bb[MaXX],
bb[MaXY],256,256);//MainActivity.Map.getWidth(),MainActivity.Map.getHeight());

            URL url = null;
            try {
                url = new URL(s);
            } catch (MalformedURLException e) {
                throw new AssertionError(e);
            }

            return url;
        }
    };

    return tileProvider;
}

```

2. Δημιουργήσαμε ένα αρχείο για τον ορισμό και οργάνωση του γραφικού περιβάλλοντος με όνομα `activity_getting_location.xml`. Το αρχείο αυτό ορίζει 3 καρτέλες με τίτλους `Maps`, `Connection Type(os)`, `Test Connection`. Στην πρώτη καρτέλα απεικονίζονται δεδομένα μετρήσεων πάνω σε χάρτη με τον χρήστη να επιλέγει τόσο τα δεδομένα που θα απεικονιστούν, όσο και τον τύπο του χάρτη που θα χρησιμοποιηθεί. Στην δεύτερη καρτέλα, εμφανίζονται πληροφορίες για την κατάσταση σύνδεσης του χρήστη που προμηθεύονται στο γραφικό περιβάλλον μέσω λειτουργιών του Android λειτουργικού. Στην τρίτη καρτέλα, έχουμε την γραφική διεπαφή μέσω της οποίας ο χρήστης, έχει την δυνατότητα να διεξάγει ελέγχους για την λήψη μετρήσεων της κατάστασης της σύνδεσης του στο διαδίκτυο.

Στο Appendix C: Κώδικας activity\_getting\_location.xml μπορεί να βρεθεί ο κώδικας του αρχείου activity\_getting\_location.xml .

3. Στο αρχείο Strings.xml προσθέσαμε καταχωρίσεις τύπου <string-array> για να δηλώσουμε τιμές για τα Spinners που έχουμε πάνω από τον χάρτη.

#### **Κώδικας Strings.xml:**

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">LBSpeedAndroid</string>
    <string name="action_settings">Settings</string>
    <string name="hello_world">Hello world!</string>
    <string name="title_activity_getting_location">GettingLocation</string>
    <string name="Label">GoogleMap location view</string>
    <string name="ConnectionType">Type of Internet Connection</string>
    <string name="Maps">Display Map</string>
    <string name="Exit">Exit</string>
    <string name="title_activity_options">Options</string>

    <string-array name="mapType">
        <item>Hybrid</item>
        <item>Satelite</item>
        <item>Terrain</item>
        <item>Map</item>
    </string-array>
    <string-array name="tileLayers">
        <item>No Data</item>
        <item>Downlink Speed</item>
        <item>Uplink Speed</item>
        <item>Ping Personal Server</item>
        <item>Ping Google</item>
    </string-array>
</resources>
```

4. Τροποποιήσαμε το αρχείο AndroidManifest.xml για να αποκτήσουμε δικαιώματα χρήσης συγκεκριμένων πόρων που απαιτούνται τόσο από τους χάρτες της Google, όσο και από τις υπόλοιπες λειτουργίες της εφαρμογής.

#### **Κώδικας AndroidManifest.xml:**

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.koros.speedtester"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="19" />

    <!-- Permission for Google Maps -->
    <permission
        android:name="com.koros.speedtester.permission.MAPS_RECEIVE"
        android:protectionLevel="signature" />

    <uses-feature android:name="android.hardware.wifi" />

    <!-- Use permission for Google Maps -->
    <uses-permission android:name="com.koros.speedtester.permission.MAPS_RECEIVE" />
    <!-- Permission for accessing the Internet -->
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <!-- Permission for writing to external storage -->
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <!-- Use permission for Google Maps -->
```

```

<uses-permission
android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<!-- Permissions for user's location -->
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>

<!-- Use OpenGL ES version 2 for Google Maps -->
<uses-feature
    android:glEsVersion="0x00020000"
    android:required="true" />
<!-- We request a larger heap to support the memory usage of our application -->
<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme"
    android:largeHeap="true">

    <meta-data
        android:name="com.google.android.gms.version"
        android:value="@integer/google_play_services_version" />
    <!-- Android key for Google Maps -->
    <meta-data
        android:name="com.google.android.maps.v2.API_KEY"
        android:value="AIzaSyAdZfcjtEOhX4ntthMmFiEwAWemsN9f7Ys" />

    <activity
        android:name="com.koros.speedtester.MainActivity"
        android:label="@string/app_name"
        android:configChanges="orientation|screenSize">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity
        android:name="com.koros.speedtester.GettingLocation"
        android:label="@string/title_activity_getting_location" >
    </activity>
    <activity
        android:name="com.koros.speedtester.Options"
        android:label="@string/title_activity_options" >
    </activity>
</application>
</manifest>

```

### 3.4 Λειτουργία της εφαρμογής LBSpeedAndroid

Η εφαρμογή μας αποτελείται από τρεις καρτέλες (tabs):

1. Maps: Στην οποία και απεικονίζονται τα δεδομένα μας στον χάρτη μαζί με την τοποθεσία του χρήστη όποτε αυτό είναι δυνατό.
2. Connection Type (OS): Στην οποία απεικονίζονται γενικές πληροφορίες σχετικά με την κατάσταση σύνδεση του χρήστη.
3. Test Connection: Στην οποία βρίσκεται το γραφικό περιβάλλον που επιτρέπει στον χρήστη να εκτελέσει τον έλεγχο της σύνδεσης του και τον ενημερώνει σχετικά με τα αποτελέσματα.

Κατά την εκκίνηση της εφαρμογής, εμφανίζεται η καρτέλα Maps στην οποία, αν η συσκευή του χρήστη τηρεί τις προϋποθέσεις

α) Η συσκευή διαθέτει εγκατεστημένα τα Google Play Services,

β) Είναι ενεργοποιημένα τα Location Services και

γ) Υπάρχει ανοιχτή σύνδεσή στο διαδίκτυο,

εμφανίζεται η τοποθεσία του χρήστη πάνω στον χάρτη, ενώ παράλληλα του δίνεται η δυνατότητα, μέσω δυο επιλογών, να αλλάξει τον τύπο του χάρτη και να απεικονίσει τα αποτελέσματα των μετρήσεων σχετικά με ένα από τα τέσσερα μεγέθη που μετρούνται. Σε περίπτωση που κάποια από τις παραπάνω προϋποθέσεις δεν πληρούνται, ο χρήστης ειδοποιείται αναλόγως:

α) Αν η συσκευή του δεν διαθέτει τα Google Play Services, εμφανίζεται μήνυμα που παραπέμπει τον χρήστη στο play store για να τα κατεβάσει,

β) Αν δεν έχει ενεργοποιημένα τα Location Services, μέσω μηνύματος ο χρήστης μεταφέρεται στις επιλογές (options) και μπορεί να τα ενεργοποιήσει και

γ) Αν δεν υπάρχει ανοιχτή σύνδεση στο διαδίκτυο, ενέργειες εξαρτώμενες από αυτή, όπως λήψη μετρήσεων απενεργοποιούνται.



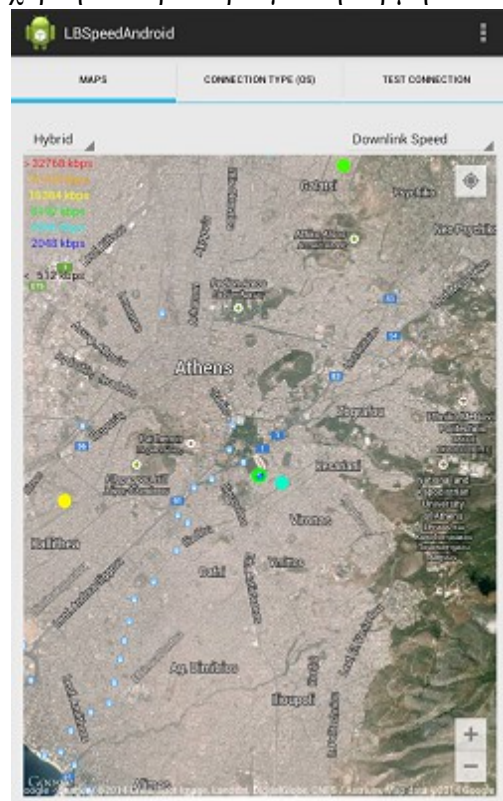


Εικόνα 3.9: Επιλογή τύπου χάρτη

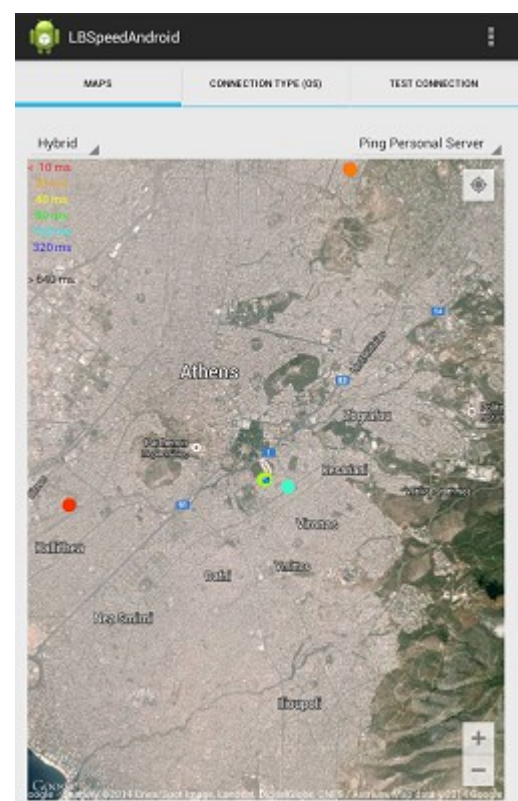


Εικόνα 3.10: Επιλογή μετρήσεων προς απεικόνιση

Η απεικόνιση των μετρήσεων πάνω στον χάρτη, μπορεί να αλλάξει μέσω του επιλογέα στα δεξιά του χάρτη, ενώ ανάλογα με την επιλογή του χρήστη εμφανίζεται και μία κλίμακα πάνω στον χάρτη και αριστερά για την ερμηνεία των απεικονίσεων.



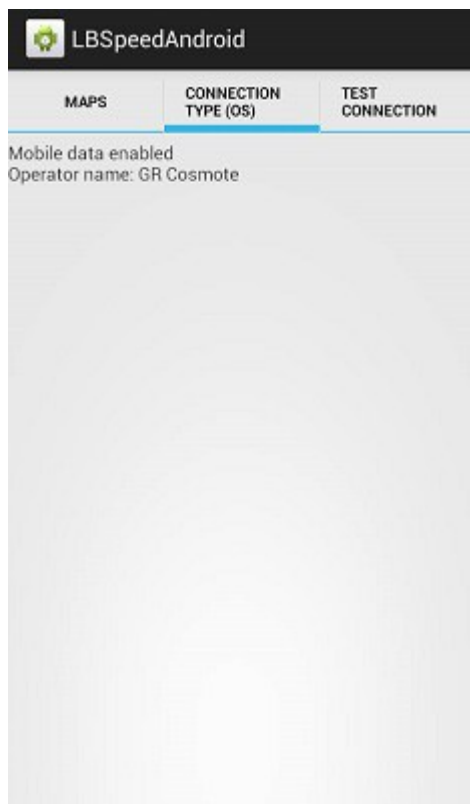
Εικόνα 3.11: Απεικόνιση Downlink



Εικόνα 3.12: Απεικόνιση Ping χρόνου



Στην καρτέλα Connection Type (OS), ο χρήστης παίρνει πληροφορίες σχετικά με τον τύπο σύνδεσης στο διαδίκτυο (WIFI, Mobile Data), ενώ στην περίπτωση που ο χρήστης συνδέεται μέσω δικτύων κινητής τηλεφωνίας (Mobile Data), απεικονίζεται και ο πάροχος του.

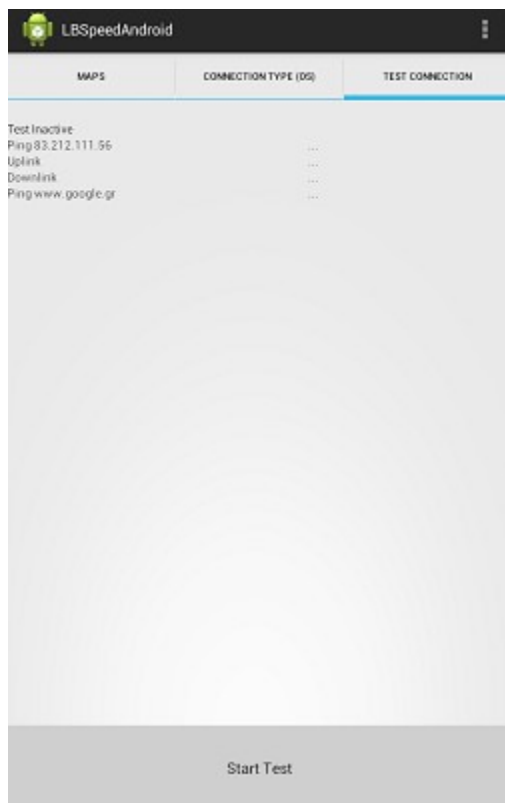


Εικόνα 3.13: Καρτέλα πληροφοριών σύνδεσης (mobile Network)



Εικόνα 3.14: Καρτέλα πληροφοριών σύνδεσης (Wi-Fi)

Ο χρήστης έχει την δυνατότητα να εκτελέσει το τεστ που υλοποιήσαμε λαμβάνοντας μετρήσεις σχετικά με την ταχύτητα αποστολής και λήψης καθώς και δύο χρόνους ping, έναν από το εικονικό μηχάνημα στο οποίο βρίσκονται οι εξυπηρετές της πλατφόρμας, και έναν από τους εξυπηρετές της Google. Σε περίπτωση που χαθεί η σύνδεση στο διαδίκτυο, είτε κατά την εκτέλεση του τεστ, είτε κατά οποιαδήποτε στιγμή της λειτουργίας, ο χρήστης ενημερώνεται με αντίστοιχο μήνυμα. Το τεστ εκκινείται όταν ο χρήστης πατήσει το κουμπί Start Test, το οποίο εν συνεχεία απενεργοποιείται κατά την διάρκεια εξαγωγής των αποτελεσμάτων και μέσω αλλαγής της ένδειξης του σε Test Started, ενημερώνει τον χρήστη για την εκκίνηση της διαδικασίας. Κατά την επιτυχή ολοκλήρωση του τεστ, ο χρήστης ενημερώνεται για την αποστολή των αποτελεσμάτων του και επανενεργοποιείται το κουμπί Start Test.



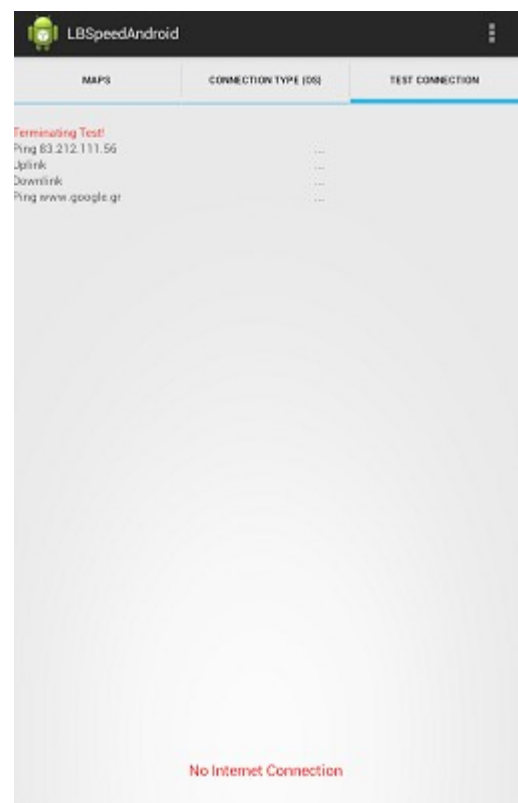
Εικόνα 3.15: Καρτέλα λήψης μετρήσεων



Εικόνα 3.16: Διεξαγωγή ελέγχου για λήψη μετρήσεων



Εικόνα 3.17: Ολοκλήρωση του ελέγχου και απεικόνιση δεδομένων



Εικόνα 3.18: Συμπεριφορά εφαρμογής σε περίπτωση έλλειψης σύνδεσης στο διαδίκτυο

### 3.5 Σύγκριση άλλων εργαλείων μέτρησης.

Στα πλαίσια της παρούσας πτυχιακής, πραγματοποιήθηκε έρευνα πάνω στα χαρακτηριστικά άλλων πιθανών εργαλείων που προσφέρονται στους χρήστες κινητών συσκευών Android μέσω του Play Store δίχως πληρωμή. Βασικός σκοπός της έρευνας αυτής, ήταν η σύγκριση τόσο των αποτελεσμάτων, όσο και των δυνατοτήτων που προσφέρουν οι εφαρμογές αυτές στους χρήστες. Όλες οι εφαρμογές που συγκρίθηκαν, ελήφθησαν δίχως πληρωμή και μπορούν να βρεθούν στο Play Store μέσω του ακόλουθου συνδέσμου <https://play.google.com/store> ή μέσω της αντίστοιχης εφαρμογής στα κινητά Android.

Οι εφαρμογές που ελέγχθηκαν, είναι οι ακόλουθες:

1. Ookla SpeedTest, Ookla
2. Speed Test Light, V-SPEED.eu
3. SpeedChecker, Optimal Software s.r.o
4. Open Signal 3G/4G/WiFi Maps & Speedtest, OpenSignal.com

Η εφαρμογή **Ookla SpeedTest** της Ookla (<https://www.ookla.com/>), αποτελεί ένα γνωστό στους χρήστες εργαλείο μέτρησης ταχύτητας σύνδεσης στο διαδίκτυο. Προσφέρει στους χρήστες, μέτρησης ταχύτητας αποστολής και λήψης δεδομένων καθώς και μία τιμή χρόνου Ping μέσα από ένα εύχρηστο και κατανοητό γραφικό περιβάλλον. Επίσης η εφαρμογή κρατά ιστορικό μετρήσεων με δυνατότητα ένδειξης της θέσης στην οποία ελήφθησαν αυτές, χωρίς όμως να απεικονίζονται τα αντίστοιχα αποτελέσματα άλλων χρηστών μέσω της εφαρμογής. Οι χρήστες, διαθέτουν την δυνατότητα να αλλάζουν τις μονάδες μέτρησης των απεικονιζόμενων μετρήσεων, καθώς και να επιλέγουν αν επιθυμούν χειροκίνητα τον εξυπηρέτη που θα χρησιμοποιηθεί για τις μετρήσεις. Δίδεται επίσης η δυνατότητα να ληφθούν μετρήσεις και μέσω του συνδέσμου <http://www.speedtest.net/> που είναι προσβάσιμος μέσω του διαδικτύου.

Η εφαρμογή Ookla SpeedTest μπορεί να βρεθεί στον ακόλουθο σύνδεσμο του Google Play Store <https://play.google.com/store/apps/details?id=org.zwanoo.android.speedtest> .

Η εφαρμογή **Speed Test Light** της V-SPEED.eu (<http://www.v-speed.eu/>), είναι μια εφαρμογή που προσφέρει στους χρήστες δυνατότητα μετρήσεων ταχύτητας αποστολής και λήψης προς το διαδίκτυο, καθώς και μέτρηση χρόνου Ping. Πλεονέκτημα της εφαρμογής αυτής, (όπως αναφέρετε στην σελίδα της στο Google Play Store) είναι η δυνατότητα της να προσαρμόζει τους αλγορίθμους που χρησιμοποιούν για την απόκτηση των μετρήσεων βάση του τύπου του δικτύου στο οποίο είμαστε συνδεδεμένοι (Wi-Fi, 2G, 3G,...). Διαθέτει ένα εύχρηστο γραφικό περιβάλλον

και κρατάει ιστορικό με παλαιότερες μετρήσεις. Δεν προσφέρει ικανότητα απεικόνισης μετρήσεων στον χάρτη.

Η εφαρμογή Speed Test Light μπορεί να βρεθεί στον ακόλουθο σύνδεσμο του Google Play Store <https://play.google.com/store/apps/details?id=eu.vspeed.android>

Η εφαρμογή **SpeedChecker** της Optimal Software s.r.o είναι μία σχετικά απλή στην χρήση εφαρμογή για την λήψη μετρήσεων σχετικά με την ταχύτητα σύνδεσης στο διαδίκτυο. Προσφέρει τιμές για την ταχύτητα λήψης και αποστολής δεδομένων καθώς και μία τιμή χρόνου Ping. Το γραφικό περιβάλλον δεν προσφέρει δυνατότητα απεικόνισης των δεδομένων σε χάρτη, ενώ αρκετές λειτουργίες του απαιτούν πληρωμή για να ενεργοποιηθούν.

Η εφαρμογή SpeedChecker μπορεί να βρεθεί στον ακόλουθο σύνδεσμο του Google Play Store <https://play.google.com/store/apps/details?id=uk.co.broadbandspeedchecker>

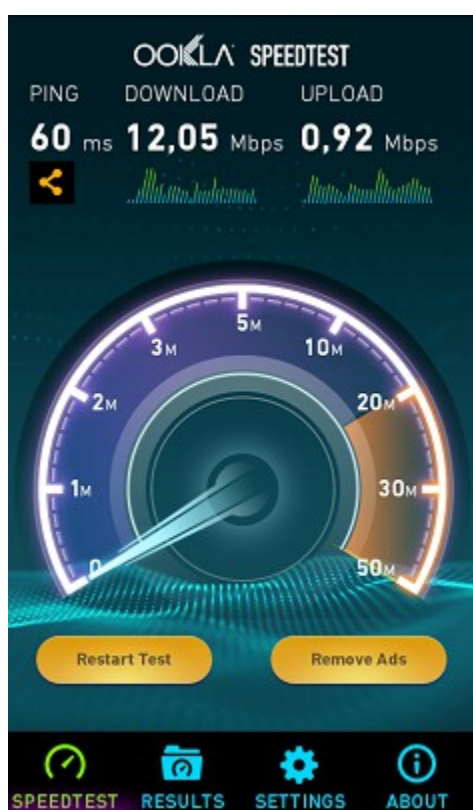
Η εφαρμογή **Open Signal 3G/4G/WiFi Maps & Speedtest** της OpenSignal.com (<http://opensignal.com/>) είναι μία άριστα δημιουργημένη εφαρμογή λήψης μετρήσεων με ένα φιλικό προς τον χρήστη γραφικό περιβάλλον, μέσω του οποίου προσφέρονται αρκετές πληροφορίες σχετικά με την σύνδεση στο διαδίκτυο. Κρατάει ιστορικό των μετρήσεων, το οποίο στην συνέχεια απεικονίζει πάνω σε χάρτη, ενώ παράλληλα διαθέτει μηχανισμούς απεικόνισης κάλυψης σημάτων κινητής τηλεφωνίας, μέσω αξιοποίησης μετρήσεων από όλους τους χρήστες της εφαρμογής. Προσφέρει τιμές για την ταχύτητα λήψης και αποστολής δεδομένων καθώς και μία τιμή χρόνου Ping. Το γραφικό περιβάλλον δεν προσφέρει δυνατότητα απεικόνισης των δεδομένων σε χάρτη, ενώ αρκετές λειτουργίες του απαιτούν πληρωμή για να ενεργοποιηθούν.

Η εφαρμογή Open Signal μπορεί να βρεθεί στον ακόλουθο σύνδεσμο του Google Play Store <https://play.google.com/store/apps/details?id=com.staircase3.opensignal>

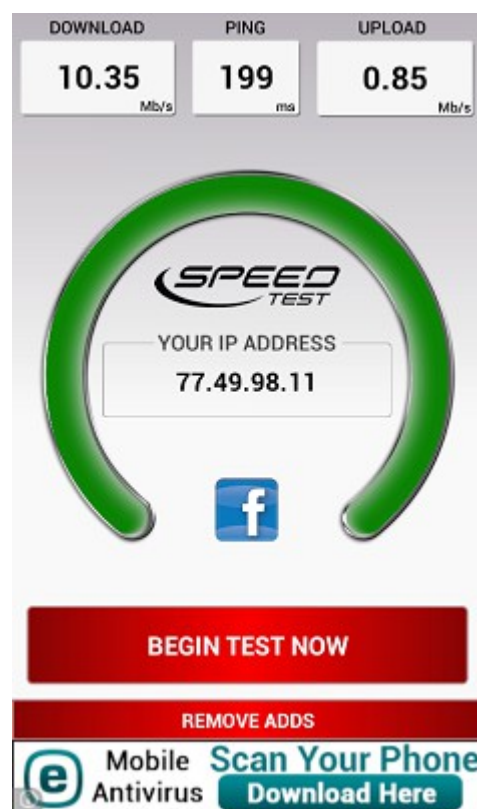
### Σύγκριση μετρήσεων

Οι μετρήσεις ελήφθησαν την ίδια ημέρα και σε συνεχόμενες χρονικές στιγμές. Παρακάτω απεικονίζονται οι οθόνες των εφαρμογών που συγκρίθηκαν με τα αποτελέσματα.

| Εφαρμογές                          | Downlink Speed (Mbps) | Uplink Speed (Mbps) |
|------------------------------------|-----------------------|---------------------|
| Ookla SpeedTest                    | 12,05                 | 0,92                |
| Speed Test Light                   | 10,35                 | 0,85                |
| SpeedChecker                       | 5,62                  | 0,61                |
| Open Signal                        | 0,737                 | 2,1                 |
| LBSPeetAndroid<br>(Η εφαρμογή μας) | 10,31                 | 0,81                |



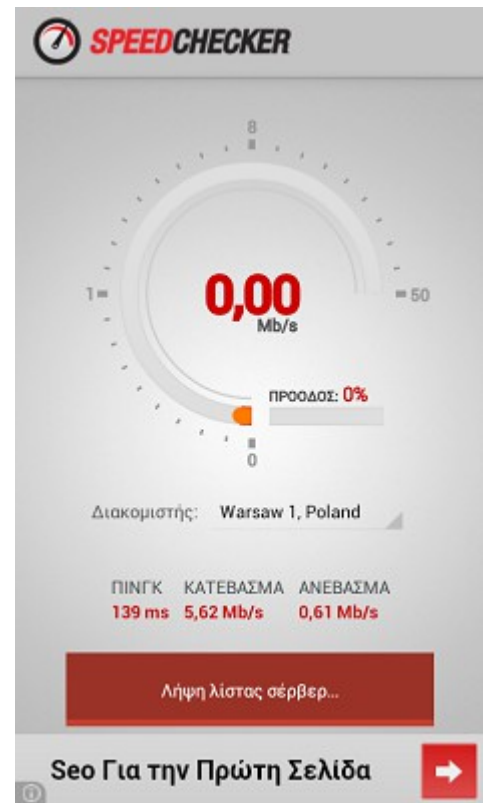
Εικόνα 3.19: Οθόνη Ookla SpeedTest



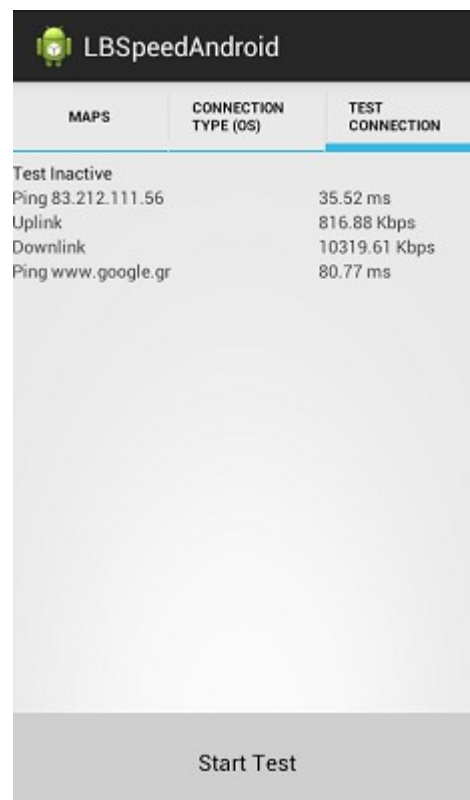
Εικόνα 3.20: Οθόνη Speed Test Light



Εικόνα 3.21: Οθόνη OpenSignal



Εικόνα 3.22: Οθόνη SpeedChecker



Εικόνα 3.23: Οθόνη LBSpeedAndroid

## 4 Υλοποίηση της διαδικτυακής πλατφόρμας

### 4.1 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 server της πλατφόρμας στην υπηρεσία cloud του okeanos ([okeanos.grnet.gr](http://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 μας 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](http://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](http://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). Για 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](http://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 [14]. 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 [15] και το Facebook [16]. Αν και γενικότερα ο Geoserver συνήθως συνδέεται με διαφορετικά συστήματα βάσεων δεδομένων τα οποία είναι σχεδιασμένα συγκεκριμένα για την παροχή υπηρεσιών για αποθήκευση και διαχείριση γεωχωρικών δεδομένων όπως είναι η PostGis [17] 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.

Καθώς η πλατφόρμα που υλοποιήσαμε, χρησιμοποιεί έναν εξυπηρέτη υλοποιημένο με χρήση της γλώσσας Java για την λήψη των μετρήσεων, κρίθηκε αναγκαίο να βρούμε έναν τρόπο να θέσουμε τον εξυπηρέτη αυτό σε συνεχόμενη λειτουργία στο παρασκήνιο. Για τον λόγο αυτό εγκαταστήσαμε το screen [18] πακέτο λογισμικού. Το screen αποτελεί έναν πολυπλέκτη συνόδων του τερματικού (terminal session multiplexer) που δίνει την δυνατότητα στον χρήστη να διαθέτει πολλαπλές συνόδους (sessions) παράλληλα, να τις διαχειρίζεται καθώς και να αποσυνδέσει μία σύνοδο αφήνοντας την να λειτουργεί στο παρασκήνιο, που είναι και η βασική λειτουργία για την οποία εγκαταστήσαμε το λογισμικό.



## 4.2 Εγκατάσταση και παραμετροποίηση των τμημάτων 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 τρέξαμε 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 επιτρέψει να εκκινούμε, να σταματάμε και να επανεκκινούμε 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 εκτελέσιμα 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
```

Για την εγκατάσταση του πακέτου `screen`, εκτελέσαμε την παρακάτω εντολή

```
apt-get install screen
```

### 4.3 Δημιουργία χρήστη στην MySQL βάση

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 'ezio'.( Στον χρήστη αυτόν, δώσαμε δικαιώματα προσπέλασης και εγγραφής αποκλειστικά και μόνο προς τους πίνακες της βάσης db1.)

Δημιουργήσαμε τον χρήστη 'ezio':

```
create user 'ezio' @ '%' 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 'ezio':

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

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

```
flush privileges;
```



#### 4.4 Δημιουργία πίνακα σε MySQL βάση για αποθήκευση μετρήσεων και γεωχωρικών δεδομένων

Με το παρακάτω MySQL query, δημιουργούμε τον πίνακα που θα κρατά τις μετρήσεις μας, καθώς και άλλα δεδομένα, όπως τύπος δικτύου (WIFI, GPRS, UMTS...), την ημερομηνία και ώρα κατά την οποία πήραμε τις μετρήσεις (προμηθεύονται από την εφαρμογή που παίρνει τις μετρήσεις), τον τύπο του λειτουργικού που διαθέτει η συσκευή (Android\_VersionNumber, IOS).

```
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 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. 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. 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 εξυπηρετή που διεξάγει η συσκευή. Μπορεί να πάρει κενή τιμή (null).
- Ping\_DN: Πεδίο τύπου κινητής υποδιαστολής (float) στο οποίο αποθηκεύονται τα εκάστοτε αποτελέσματα της μέτρησης ping προς τον εξυπηρετή της Google που διεξάγει η συσκευή. 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 [19] 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.



## 4.5 Υλοποίηση της υπηρεσίας διαδικτύου

Για τον σκοπό της επικοινωνίας των εφαρμογών πελατών (Clients) με την βάση δεδομένων στην οποία αποθηκεύονται όλες οι μετρήσεις, επιλέξαμε να υλοποιήσουμε μία υπηρεσία διαδικτύου υπό την μορφή μίας σελίδα που δημιουργήθηκε με χρήση της γλώσσας PHP (<http://www.php.net/>). Μέσω της σελίδας αυτής, η οποία λαμβάνει HTTP (Hypertext Transfer Protocol) Post αιτήσεις, εισάγουμε τα δεδομένα στην βάση με τον ακόλουθο τρόπο:

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

```
<?php $con=mysqli_connect("83.212.111.56","ezio","*****","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>
```

## 4.6 Υλοποίηση και λειτουργία του εξυπηρέτη για χρήση στην εξαγωγή μετρήσεων

Η εξαγωγή των μετρήσεων ταχύτητας δικτύου, πραγματοποιείται μέσω επικοινωνίας της εφαρμογής που υλοποιήσαμε για συσκευές Android με έναν πολυνηματικό εξυπηρέτη που δημιουργήσαμε χρησιμοποιώντας την γλώσσα Java. Στον εξυπηρέτη αυτό συνδεόμαστε χρησιμοποιώντας TCP (Transmission Control Protocol) υποδοχής (sockets) και επικοινωνούμε μέσω ανταλλαγής μηνυμάτων. Καθώς όπως προαναφέραμε ο εξυπηρέτης είναι πολυνηματικός, υποστηρίζεται παράλληλη σύνδεση πολλαπλών χρηστών σε αυτόν. Χρησιμοποιήσαμε το περιβάλλον NetBeans IDE 8.0 (<https://netbeans.org/features/index.html>) για την δημιουργία του εξυπηρέτη και την εξαγωγή αυτού σε συμπιεσμένη εκτελέσιμη μορφή την οποία και έχουμε θέσει σε λειτουργία στο ίδιο μηχάνημα που έχουμε εγκαταστήσει όλα τα τμήματα εξυπηρέτες της πλατφόρμας, ενώ για να εξασφαλίσουμε ότι θα λειτουργεί όσο είναι ενεργοποιημένο το μηχάνημα, χρησιμοποιήσαμε το εργαλείο screen των Ubuntu, την λειτουργία του οποίου θα εξηγήσουμε παρακάτω.

### Υλοποίηση του εξυπηρέτη

Στο NetBeans, δημιουργήσαμε ένα project με όνομα SpeedTestServer με τις παρακάτω κλάσεις. Για να δημιουργήσουμε ένα εκτελέσιμο (.jar) αρχείο για να ανεβάσουμε στο εικονικό μηχάνημα και να το θέσουμε σε λειτουργία, κάνουμε δεξί κλικ στο project SpeedTestServer και επιλέγουμε Clean and Build. Το εκτελέσιμο αρχείο βρίσκεται στον φάκελο dist του φακέλου που περιέχει τον κώδικα του project.

Ο εξυπηρέτης αποτελείται από τρεις κλάσεις: Server, SubServerThread, TestingServer.

Αρχικά έχουμε την **Server** κλάση στην οποία και υλοποιείτε η πολυνηματική φύση του εξυπηρέτη. Χρησιμοποιούμε μία δομή επανάληψης while, μέσα στην οποία δημιουργούμε ένα νήμα εκτέλεσης της κλάσης SubServerThread κάθε φορά που δεχόμαστε ένα αίτημα σύνδεσης στην υποδοχή εξυπηρέτη (Server Socket) η οποία περιμένει τέτοια αιτήματα στην θύρα 9222.

### Κώδικας της Server:

```
package SocketServer;
import java.io.IOException;
import java.net.ServerSocket;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
public class Server {
    static ServerSocket ss;
    static Thread t;
    static boolean looping;
    public static void main(String[] args) {
        try {
            looping=true;
```

```

        ss=new ServerSocket(9222);//We create a server socket object that listens for request
        at port 9222

        do{
            if(looping){

                new Thread(new SubServerThread(ss.accept())).start();//If we receive a
                connection request we create a separate thread of execution and start the speed testing procedure

            }
        }while(looping);
    } catch (IOException ex) {
        Logger.getLogger(Server.class.getName()).log(Level.SEVERE, null, ex);
    }
}
}

```

Στην συνέχεια, έχουμε την κλάση SubServerThread, η οποία υλοποιεί (implements) την κλάση Runnable, εμπεριέχει το σύνολο των μηνυμάτων που ανταλλάσσονται με την εφαρμογή και διαχειρίζεται λειτουργίες αποστολής και λήψης δεδομένων της κλάσης TestingServer. Η κλάση SubServerThread διαθέτει την μέθοδο bufferSendFillArray η οποία φορτώνει την φωτογραφία random3000x3000.jpg:



*Εικόνα 4.1: Εικόνα που χρησιμοποιούμε για μέτρηση ταχύτητας λήψης*

που διαθέτει μέγεθος 17 Mega Bytes και παράλληλα χρησιμοποιείται και από την Ookla.

Η βασική λειτουργία της κλάσης αυτής είναι η ανταλλαγή μηνυμάτων “ack” καθώς και η αμφίδρομη ενημέρωση των 2 μερών της σύνδεσης (client – server) σχετικά με το μέγεθος των πληροφοριών που επρόκειτο να αποσταλούν.

#### **Κώδικας της SubServerThread:**

```
package SocketServer;
```

```

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.net.Socket;
import java.util.Random;
import java.util.logging.Level;
import java.util.logging.Logger;

public class SubServerThread implements Runnable{
    Socket connection;
    DataInputStream dis;
    DataOutputStream dos;
    int RandomPort;
    static byte bufferSend[];
    public SubServerThread(Socket s){
        connection=s;
        System.out.println("Socket is"+(connection.isBound()? "bound":"not bound"));
        try {
            dis=new DataInputStream(connection.getInputStream());
            dos=new DataOutputStream(connection.getOutputStream());
        } catch (IOException ex) {
            Logger.getLogger(SubServerThread.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    static void bufferSendFillArray(){//Read the random3000x3000.jpg file to be send to
application
        try {

            File f=new File("random3000x3000.jpg");
            FileInputStream fis=new FileInputStream(new File("random3000x3000.jpg"));
            SubServerThread.bufferSend=new byte[ (int)f.length()];
            fis.read(bufferSend);
            fis.close();

        } catch (FileNotFoundException ex) {
            Logger.getLogger(SubServerThread.class.getName()).log(Level.SEVERE, null, ex);
        } catch (IOException ex) {
            Logger.getLogger(SubServerThread.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    @Override
    public void run() {
        try {
            bufferSendFillArray();
            if(dis.readUTF().equals("ack")){
                int bufferRec=dis.readInt();//We receive the size of the file that the
application is about to send
                TestingServer ts=new TestingServer(connection,bufferRec);
                //uplink
                System.out.println("Preparing to recieve data\nIP: "+connection.getLocalAddress()+"Mark
II\n");

                ts.recieveData();

                System.out.println("Recieved data\n");
                dos.writeInt(bufferSend.length);//We inform the connected application of the size
of the file we are about to send
                if(dis.readUTF().equals("ack")){//downlink

                    ts.sendData();
                    dis.close();
                    dos.close();
                    connection.close();

                }

            }

        } catch (IOException ex) {
            Logger.getLogger(SubServerThread.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}

```

```

    }
}

```

Τέλος, έχουμε την κλάση **TestingServer**. Σε αυτήν υλοποιήθηκαν μέθοδοι για την αποστολή και λήψη των δεδομένων που ανταλλάσσονται κατά την διάρκεια των μετρήσεων.

### Κώδικας της TestingServer:

```

package SocketServer;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.Socket;
import java.net.SocketException;
import java.util.Random;
import java.util.logging.Level;
import java.util.logging.Logger;

public class TestingServer {
    DataInputStream dis;
    DataOutputStream dos;

    InetAddress servAdd;
    Socket s;

    byte[] buffer;
    int bufferSizeRec;

    public TestingServer(Socket s,int bufferSizeRec){
        try {

            this.s=s;
            dis=new DataInputStream(s.getInputStream());
            dos=new DataOutputStream(s.getOutputStream());

            this.bufferSizeRec=bufferSizeRec;
            buffer=new byte[(int)bufferSizeRec];

        } catch (IOException ex) {
            Logger.getLogger(TestingServer.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    public void packetHandeling(boolean receiv){
        if(receiv){//We receive data from application for uplink measurement
            try {

                dis.readFully(buffer);

            } catch (IOException ex) {
                Logger.getLogger(TestingServer.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
        else{//We send data to application for downlink measurement
            try {
                buffer=SubServerThread.bufferSend;

                int i=0;
                while(i<buffer.length){

                    dos.write(buffer,i, (i+50000)>buffer.length?buffer.length-i:50000);
                    dos.flush();
                    i+=(i+50000)>buffer.length?buffer.length-i:50000;

                }
            }
        }
    }
}

```



```
        } catch (IOException ex) {  
            Logger.getLogger(TestingServer.class.getName()).log(Level.SEVERE, null, ex);  
        }  
    }  
    public void sendData(){//Method For Sending Data to Client  
        packetHandeling(false);  
    }  
  
    public void recieveData() {//Method For Recieving Data from Client  
        packetHandeling(true);  
    }  
}
```

### **Εκτέλεση του εξυπηρέτη**

Για να θέσουμε σε λειτουργία τον εξυπηρέτη στο παρασκήνιο και συνεπώς ανεξάρτητα από την σύνοδο με την οποία συνδεόμαστε στο εικονικό μηχάνημα, χρησιμοποιήσαμε το πακέτο λογισμικού screen ως εξής:

Στο τερματικό μέσω του οποίου έχουμε συνδεθεί στο εικονικό μηχάνημα, και αφού μεταφέρουμε σε συμπιεσμένη μορφή (.jar) τον εξυπηρέτη που υλοποιήσαμε σε Java και αποκτήσουμε δικαιώματα εκτέλεσης σε αυτό, ενεργοποιούμε το screen με την παρακάτω εντολή:

```
screen
```

Και τρέχουμε το αρχείο του εξυπηρέτη στον φάκελο που τον κατεβάσαμε:

```
./SpeedTestServer.jar
```

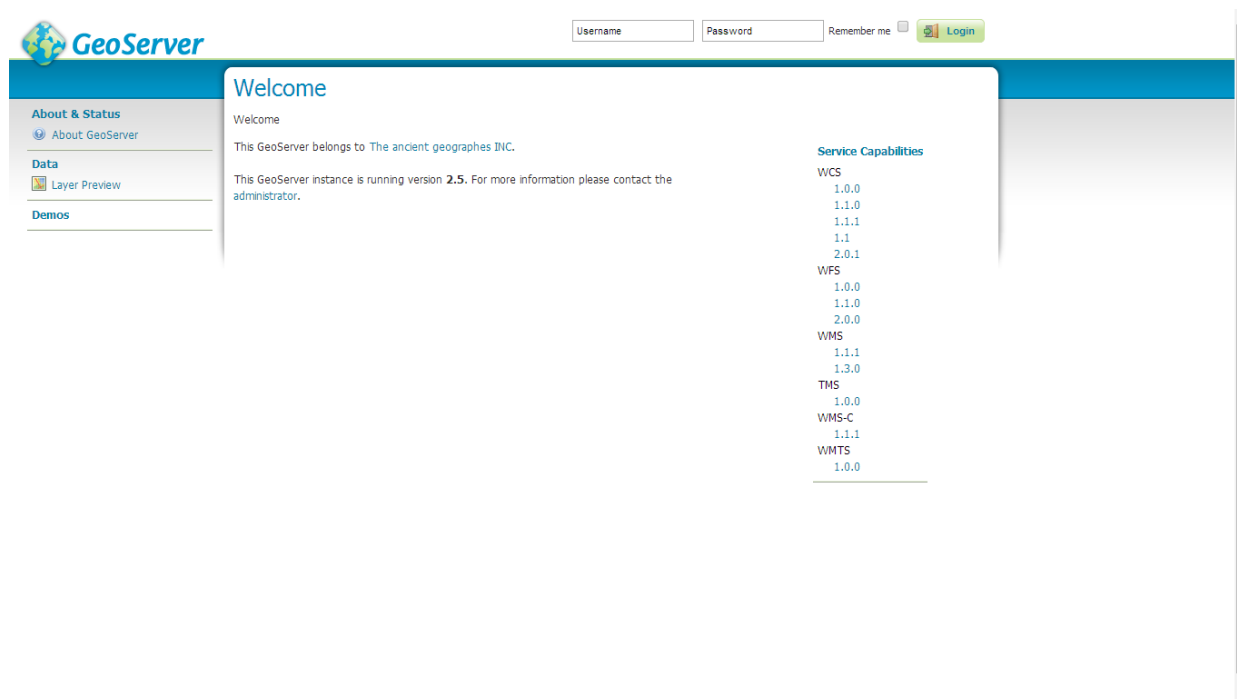
Ενώ για να αποκολλήσουμε την σύνοδο απλά πληκτρολογούμε :

```
Ctrl+A+d
```

Και επιστρέφουμε στην αρχική σύνοδο.

## 4.7 Είσοδος και χρήση του Geoserver

Για να εισέλθουμε στην δικτυακή διεπαφή του Geoserver, δεδομένου ότι έχουμε μία απaráλλαχτη εγκατάσταση του Tomcat [20] που λειτουργεί ως web container [21] και δηλαδή δεν έχουμε αλλάξει τον αριθμό πόρτας στον οποίο ανταποκρίνεται ο Geoserver σε αιτήσεις σύνδεσης, μεταβαίνουμε με τον διαδικτυακό μετρητή της επιλογής μας στην διεύθυνση **http://<IP-address-of-Geoserver-host>:8080/geoserver** (**http://<IP-Διεύθυνση-του-Geoserver-μηχανήματος>:8080/geoserver**)



Εικόνα 4.2: Log in σελίδα του Geoserver

Στη συνέχεια και αφού εισάγουμε σωστά αναγνωριστικά για τον χρήστη (όνομα χρήστη και κωδικό), μεταβαίνουμε στην κεντρική σελίδα όπου μας δίδεται η δυνατότητα να διαχειριστούμε τις υπηρεσίες που προσφέρει ο Geoserver.

The screenshot displays the GeoServer web interface. At the top left is the GeoServer logo. The top right corner shows the user is logged in as 'nick' with a 'Logout' button. The left sidebar contains several sections: 'About & Status' (Server Status, GeoServer Logs, Contact Information, About GeoServer), 'Data' (Layer Preview, Workspaces, Stores, Layers, Layer Groups, Styles), 'Services' (WCS, WFS, WMS), 'Settings' (Global, JAI, Coverage Access), 'Tile Caching' (Tile Layers, Caching Defaults, Gridsets, Disk Quota), 'Security' (Settings, Authentication, Passwords, Users, Groups, Roles, Data, Services), 'Demos', and 'Tools'. The main content area features a 'Welcome' message, stating 'This GeoServer belongs to The ancient geographies INC.' Below this, there are statistics: 20 Layers (with an 'Add layers' button), 10 Stores (with an 'Add stores' button), and 7 Workspaces (with a 'Create workspaces' button). Two warning messages are displayed: one about the master password not being changed from the default, and another about strong cryptography. At the bottom, it states 'This GeoServer instance is running version 2.5. For more information please contact the administrator.' On the right side of the main content area, there is a 'Service Capabilities' section listing various services and their versions: 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), and WMTS (1.0.0).

Εικόνα 4.3: Κεντρική σελίδα του Geoserver

Η πρώτη μας δουλειά είναι να δημιουργήσουμε μία σύνδεση με την βάση δεδομένων στην οποία βρίσκονται τα δεδομένα γεωγραφικής τοποθεσίας που επιθυμούμε να απεικονίσουμε. Για να το κάνουμε αυτό, μεταβαίνουμε στο αριστερό menu και στην κατηγορία data (δεδομένα) επιλέγουμε την υποκατηγορία Stores. Η επιλογή αυτή μας οδηγεί στην σελίδα διαχείρισης των Stores.

GeoServer

Logged in as nick. [Logout](#)

### Stores

Manage the stores providing data to GeoServer

[Add new Store](#)  
[Remove selected Stores](#)

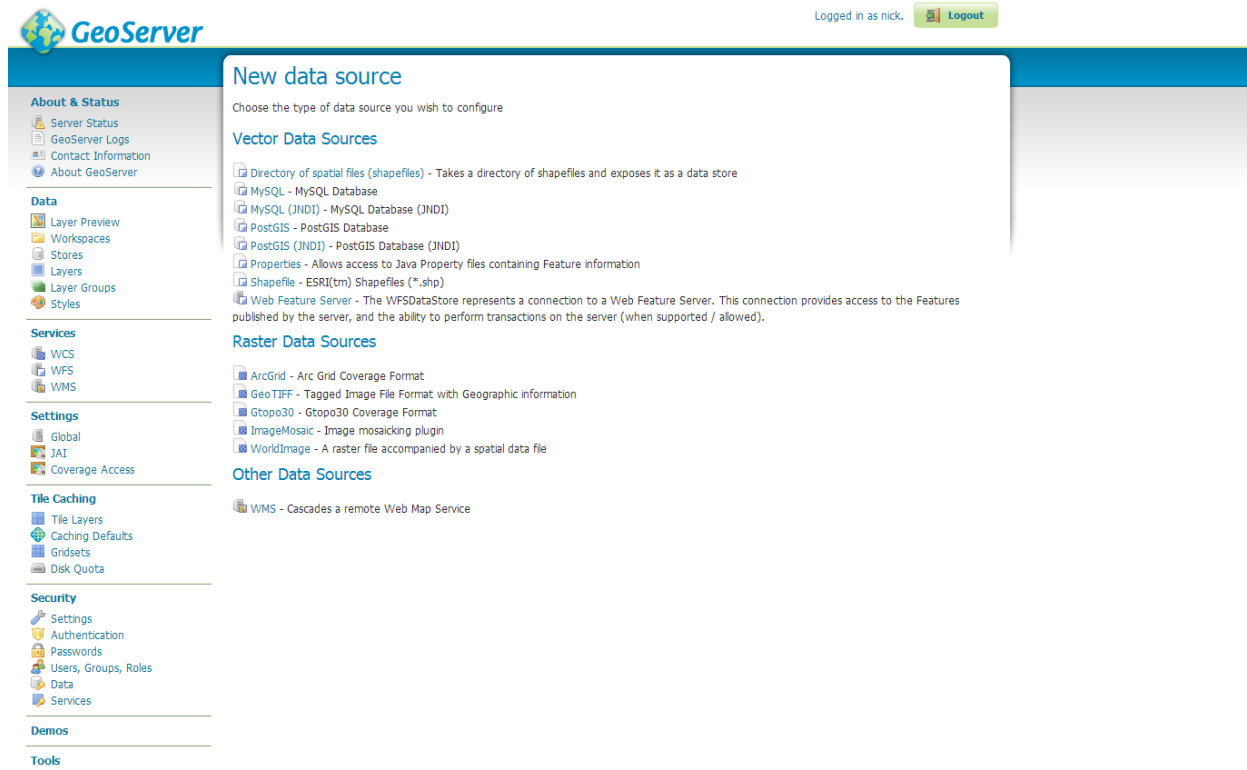
<< < > >> Results 1 to 10 (out of 10 items)

| <input type="checkbox"/> | Data Type | Workspace | Store Name       | Type        | Enabled? |
|--------------------------|-----------|-----------|------------------|-------------|----------|
| <input type="checkbox"/> |           | sf        | sfdem            | GeoTIFF     | ✓        |
| <input type="checkbox"/> |           | nurc      | worldImageSample | WorldImage  | ✓        |
| <input type="checkbox"/> |           | nurc      | arcGridSample    | ArcGrid     | ✓        |
| <input type="checkbox"/> |           | nurc      | mosaic           | ImageMosaic | ✓        |
| <input type="checkbox"/> |           | nurc      | img_sample2      | WorldImage  | ⚠        |
| <input type="checkbox"/> |           | sf        | sf               | Shapefile   | ✓        |
| <input type="checkbox"/> |           | tiger     | nyc              | Shapefile   | ✓        |
| <input type="checkbox"/> |           | topp      | taz_shapes       | Shapefile   | ✓        |
| <input type="checkbox"/> |           | topp      | states_shapefile | Shapefile   | ✓        |
| <input type="checkbox"/> |           | cite      | MySQLDB          | MySQL       | ✓        |

<< < > >> Results 1 to 10 (out of 10 items)

Εικόνα 4.4: Σελίδα διαχείρισης Stores του Geoserver

Για την δημιουργία ενός νέου Store, μεταφερόμαστε μέσω του συνδέσμου Add new Store στην σελίδα προσθήκης νέας πηγής δεδομένων, στην οποία μας δίνεται η δυνατότητα να επιλέξουμε μέσα από μια μεγάλη ποικιλία διαφορετικών βάσεων δεδομένων.



Εικόνα 4.5: Σελίδα δημιουργίας νέας πηγής πληροφοριών του Geoserver

Εμείς επιλέξαμε την MySQL, καθώς τέτοιου τύπου βάση χρησιμοποιήσαμε.

Στην συνέχεια, συμπληρώσαμε κάποια πεδία τα οποία είναι σημαντικά στην ορθή σύνδεση της βάσης με τον Geoserver όπως:

- Workpsace: Ποιο workspace θα καταχωρηθεί η σύνδεση αυτή (εμείς επιλέξαμε το cite).
- host: Η IP διεύθυνση του μηχανήματος στο οποίο υφίσταται η βάση δεδομένων .
- port: Σε ποιόν αριθμό πόρτας θα περιμένει αίτησης για σύνδεση η βάση δεδομένων.
- database: Το αναγνωριστικό της βάσης δεδομένων, στους πίνακες της οποίας θέλουμε να συνδεθούμε.
- user: Αναγνωριστικό ενός χρήστη της βάσης δεδομένων.
- passwd: Κωδικός πρόσβασης που συνδέεται με το παραπάνω αναγνωριστικό.

Και πατώντας το Save κουμπί, αποθηκεύουμε το νέο data source.

The screenshot displays the 'Edit Vector Data Source' interface in GeoServer. The left sidebar contains navigation menus for 'About & Status', 'Data', 'Services', 'Settings', 'Tile Caching', 'Security', 'Demos', and 'Tools'. The main content area is titled 'Edit Vector Data Source' and includes the following sections:

- Basic Store Info**: Workspace (cite), Data Source Name (MySQLDB), Description, and an 'Enabled' checkbox.
- Connection Parameters**: host (83.212.111.56), port (3306), database (db1), user (ezio), password (masked), and namespace (http://www.opengeospatial.net/cite). It also includes an 'Expose primary keys' checkbox, max connections (10), min connections (1), fetch size (1000), connection timeout (20), and a 'validate connections' checkbox.
- Primary key metadata table**: A text input field.
- Session startup SQL**: A text input field.
- Session close-up SQL**: A text input field.
- storage engine**: MySQLAM.

At the bottom of the form are 'Save' and 'Cancel' buttons.

Εικόνα 4.6: Σελίδα διαχείρισης σύνδεσης σε βάση δεδομένων του Geoserver

Έπειτα, το επόμενο βήμα στην ρύθμιση του μηχανισμού απεικόνισης των μετρήσεων μας, είναι η δημιουργία ενός Layer [22]. Αρχικά μεταβαίνουμε μέσω της επιλογής Layers του αριστερού στην επιφάνεια διαχείρισης των Layers.

GeoServer

Logged in as nick. Logout

**Layers**

Manage the layers being published by GeoServer

Add a new resource

Remove selected resources

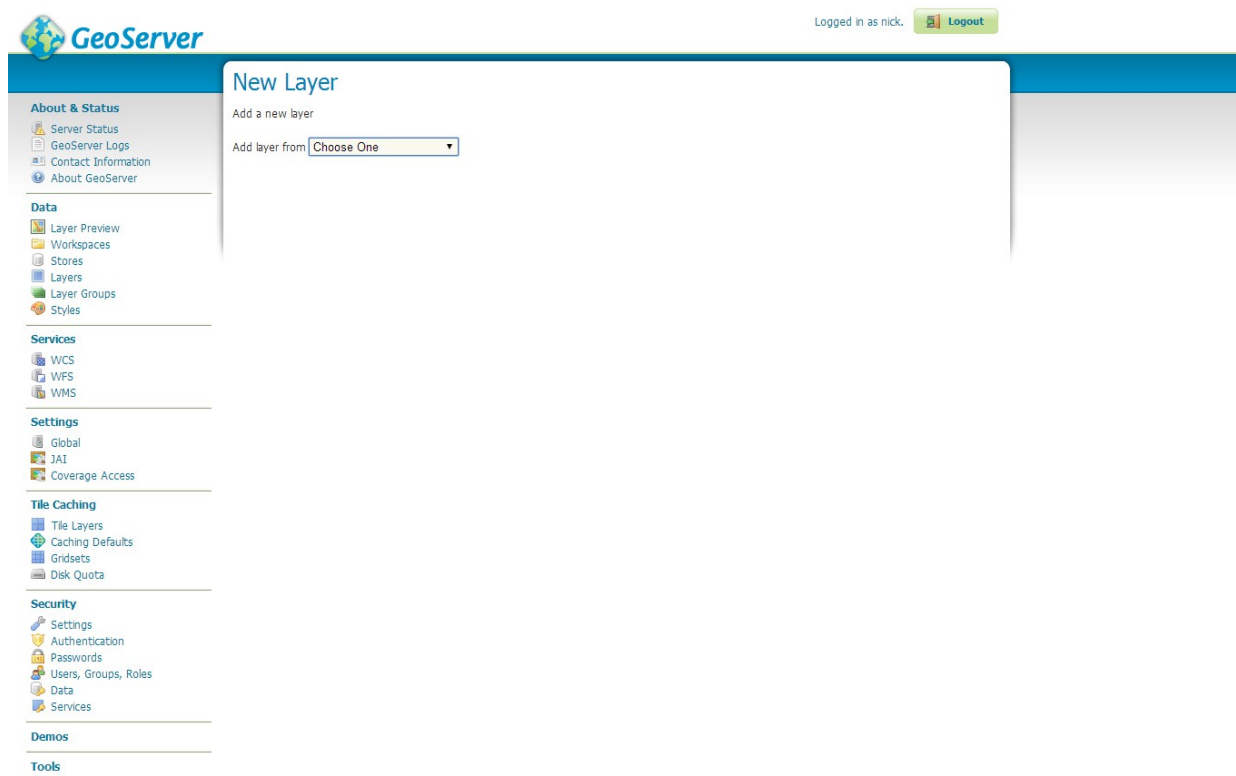
Results 1 to 20 (out of 20 items)

| Type | Workspace | Store            | Layer Name                | Enabled? | Native SRS |
|------|-----------|------------------|---------------------------|----------|------------|
|      | sf        | sf               | restricted                | ✓        | EPSG:26713 |
|      | sf        | sf               | roads                     | ✓        | EPSG:26713 |
|      | sf        | sf               | streams                   | ✓        | EPSG:26713 |
|      | sf        | sf               | archsites                 | ✓        | EPSG:26713 |
|      | sf        | sf               | bugsites                  | ✓        | EPSG:26713 |
|      | sf        | sf               | sfdem                     | ✓        | EPSG:26713 |
|      | nurc      | worldImageSample | Img_Sample                | ✓        | EPSG:4326  |
|      | nurc      | arcGridSample    | Arc_Sample                | ✓        | EPSG:4326  |
|      | nurc      | mosaic           | mosaic                    | ✓        | EPSG:4326  |
|      | nurc      | img_sample2      | Pk50095                   | ⚠        | EPSG:32633 |
|      | tiger     | nyc              | poi                       | ✓        | EPSG:4326  |
|      | tiger     | nyc              | poly_landmarks            | ✓        | EPSG:4326  |
|      | tiger     | nyc              | tiger_roads               | ✓        | EPSG:4326  |
|      | tiger     | nyc              | giant_polygon             | ✓        | EPSG:4326  |
|      | topp      | taz_shapes       | tasmania_water_bodies     | ✓        | EPSG:4326  |
|      | topp      | taz_shapes       | tasmania_cities           | ✓        | EPSG:4326  |
|      | topp      | taz_shapes       | tasmania_roads            | ✓        | EPSG:4326  |
|      | topp      | taz_shapes       | tasmania_state_boundaries | ✓        | EPSG:4326  |
|      | topp      | states_shapefile | states                    | ✓        | EPSG:4326  |
|      | cite      | MySQLDB          | measurements              | ✓        | EPSG:4326  |

Results 1 to 20 (out of 20 items)

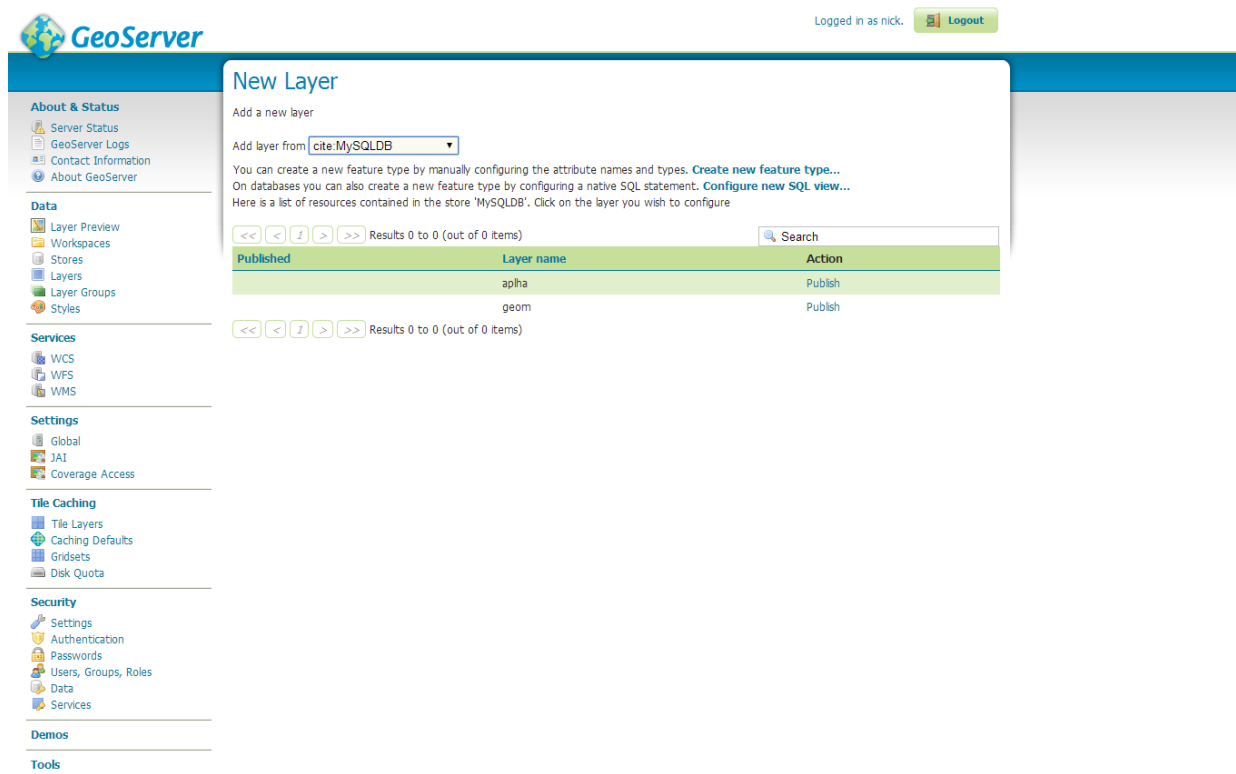
Εικόνα 4.7: Σελίδα διαχείρισης Layers του Geoserver

Και μέσω του add new resource συνδέσμου εκκινούμε την διαδικασία δημιουργίας νέου Layer.



Εικόνα 4.8: Σελίδα επιλογής data source για δημιουργία νέου Layer του Geoserver

Καθορίζουμε μέσω του επιλογέα από ποιο Store θα εισάγουμε το Layer



Εικόνα 4.9: Σελίδα επιλογής data source για δημιουργία νέου Layer του Geoserver

Και επιλέγουμε το Configure new SQL view... που μας οδηγεί στην ακόλουθη επιφάνεια, στην οποία και εισάγουμε τα έξεις δεδομένα:

- View name: Το όνομα του Layer.
- SQL statement: Ερώτημα SQL για την λήψη δεδομένων από το Store που επιλέξαμε.



**GeoServer** Logged in as nick. [Logout](#)

**About & Status**

- Server Status
- GeoServer Logs
- Contact Information
- About GeoServer

**Data**

- Layer Preview
- Workspaces
- Stores
- Layers
- Layer Groups
- Styles

**Services**

- WCS
- WFS
- WMS

**Settings**

- Global
- JAI
- Coverage Access

**Tile Caching**

- Tile Layers
- Caching Defaults
- Gridsets
- Disk Quota

**Security**

- Settings
- Authentication
- Passwords
- Users, Groups, Roles
- Data
- Services

**Demos**

**Tools**

### Create new SQL view

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

**View Name**

**SQL statement**

**SQL view parameters**

[Guess parameters from SQL](#) [Add new parameter](#) [Remove selected](#)

| <input type="checkbox"/>            | Name                          | Default value | Validation regular expression |
|-------------------------------------|-------------------------------|---------------|-------------------------------|
| <input checked="" type="checkbox"/> | Escape special SQL characters |               |                               |


**Attributes**

[Refresh](#) ☐ Guess geometry type and srid

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

[Save](#) [Cancel](#)

Εικόνα 4.10: Σελίδα δημιουργίας *SQL view* δημιουργία νέου *Layer* του *Geoserver*  
Πατώντας το *Save* κουμπί, μεταφερόμαστε στην *Edit Layer* οθόνη.


Logged in as nick. [Logout](#)

**About & Status**

- Server Status
- GeoServer Logs
- Contact Information
- About GeoServer

**Data**

- Layer Preview
- Workspaces
- Stores
- Layers
- Layer Groups
- Styles

**Services**

- WCS
- WFS
- WMS

**Settings**

- Global
- JAI
- Coverage Access

**Tile Caching**

- Tile Layers
- Caching Defaults
- Gridsets
- Disk Quota

**Security**

- Settings
- Authentication
- Passwords
- Users, Groups, Roles
- Data
- Services

**Demos**

**Tools**

## 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

**New Keyword**  

▼

**Vocabulary**

**Metadata links**

No metadata links so far

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  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**

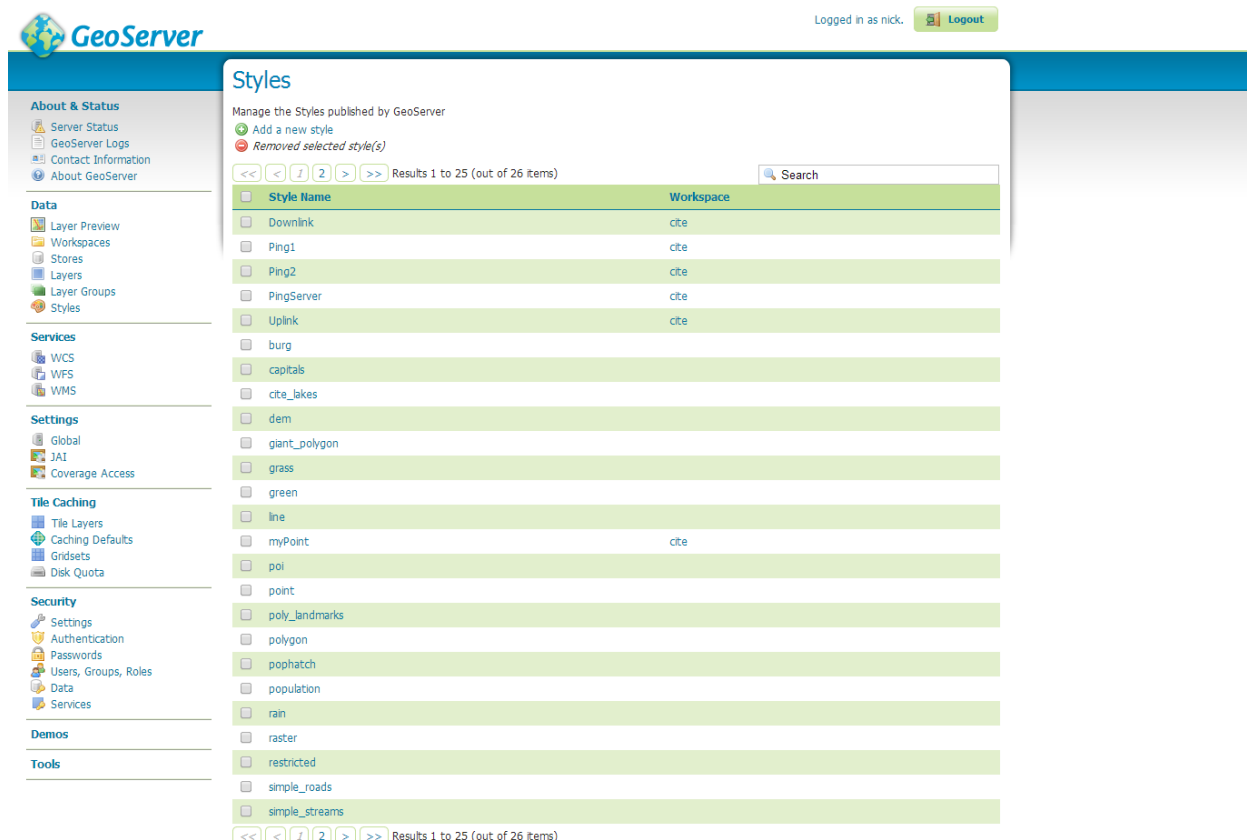
| Property         | Type      | Nullable | Min/Max Occurrences |
|------------------|-----------|----------|---------------------|
| Downlink         | Float     | false    | 1/1                 |
| ID               | Integer   | false    | 1/1                 |
| latitude         | Float     | false    | 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](#)

Εικόνα 4.11: Σελίδα επεξεργασίας του νέου Layer του Geoserver

Σε αυτήν μας δίνεται η δυνατότητα να καθορίσουμε περαιτέρω παραμέτρους, όπως το σύστημα συντεταγμένων (εμείς εισάγαμε το EPSG:4326), το Native Bounding Box (το οποίο συμπληρώθηκε αυτόματα μέσω της επιλογής Compute from data), το Lat/Lon Bounding Box (οι τιμές του οποίου είναι ίδιες με του Native Bounding Box). Για να αποθηκεύσουμε τις αλλαγές μας απλά πατάμε το κουμπί Save που βρίσκεται στο κάτω μέρος της επιφάνειας.

Τέλος δημιουργούμε ένα Style, μέσω του οποίου θα απεικονίσουμε τα δεδομένα μας από τα Layers. Επιλέγουμε τον σύνδεσμο Style στο αριστερό menu και ανοίγει ο Styles manager.



Εικόνα 4.12: Σελίδα διαχείρισης Styles του Geoserver

Έπειτα, επιλέγουμε το **Add new style** για να μεταφερθούμε στην σελίδα δημιουργίας νέου Style. Σε αυτήν εισάγουμε το όνομα του νέου Style που επιθυμούμε να δημιουργήσουμε, το workspace στο οποίο θα αποθηκευτεί το Style καθώς και τον SLD [23] κώδικα ο οποίος περιγράφει την μορφοποίηση απεικόνισης των δεδομένων.

GeoServer

Logged in as nick. Logout

### New style

Type a new SLD definition, or use an existing one as a template, or upload a ready made style from your file system. 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**

Choose One [Copy ...](#)

12pt

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30

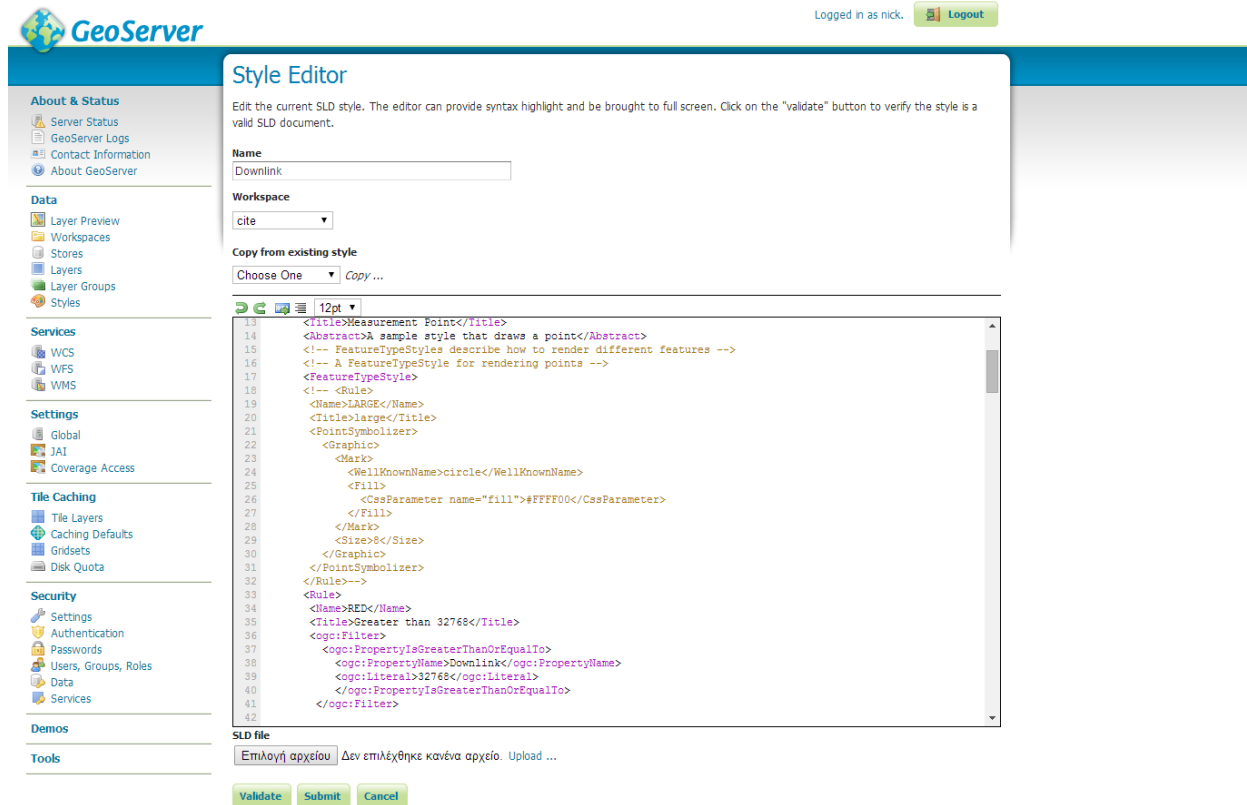
**SLD file**

Επιλογή αρχείου Δεν επιλέχθηκε κανένα αρχείο. Upload ...

Validate Submit Cancel

Εικόνα 4.13: Σελίδα δημιουργίας νέου Style του Geoserver

Το παρακάτω, είναι ένα παράδειγμα ενός Style που υλοποιήσαμε για την απεικόνιση των μετρήσεων ταχύτητας λήψης .



Εικόνα 4.14: Παράδειγμα ενός Style του Geoserver

Μέσω των Styles που δημιουργήσαμε, αντιστοιχήσαμε μετρήσεις και χρώματα για απεικόνιση όπως φέnete στις εικόνες 4.15 και 4.16.

- Greater than 32768
- 16384 to 32768
- 8192 to 16384
- 4096 to 8192
- 2048 to 4096
- 1024 to 2048
- 512 to 1024
- Less than 512

Εικόνα 4.15:  
Αντιστοίχιση χρωμάτων  
για μετρήσεις  
Downlink, Uplink  
(Kbps)

- Greater than 640
- 320 to 640
- 160 to 320
- 80 to 160
- 40 to 80
- 20 to 40
- 10 to 20
- Less than 10

Εικόνα 4.16:  
Αντιστοίχιση  
χρωμάτων για τις  
μετρήσεις Ping (ms)

## 4.8 Οπτική αναπαράσταση των μετρήσεων μας μέσω SLD

To 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). Μέσω αυτού μπορέσαμε στον Geoserver να οπτικοποιήσουμε τα δεδομένα μας και να τα απεικονίσουμε 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 συγκεκριμένο τρόπο.

Μέσα στους κανόνες, διαθέτουμε τα εξής πεδία:

- Name: Όνομα του κανόνα.
- Title: Τίτλος του κανόνα.
- Filter: Πεδίο που περιέχει συνθήκης, μέσω των οποίων επιλέγονται τα δεδομένα που θα αναπαρασταθούν με τον τρόπο που δηλώνει ο κανόνας. Πιο συγκεκριμένα έχουμε συνθήκες, όπως η ιδιότητα είναι μεγαλύτερη ή ίση (ogc:PropertyIsGreaterThanOrEqual) μιας τιμής. Μία ιδιότητα (ogc:Property) που προσφέρεται από το εκάστοτε store. Μία τιμή (ogc:Literal) η οποία συγκρίνεται με την ιδιότητα στην συνθήκη.
- Point Symbolizer: Ρυθμίσεις σχετικά με την απεικόνιση του συγκεκριμένου κανόνα, όπως το σχήμα που θα χρησιμοποιηθεί για την αναπαράσταση, το χρώμα του σχήματος και το μέγεθος σε pixels.

Ένα παράδειγμα ενός SLD κώδικα που χρησιμοποιήσαμε για να αναπαραστήσουμε την μέγιστη μετρήσει της ταχύτητας λήψης.

```
<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>  
</Rule>
```

#### **4.9 Απεικόνιση αποτελεσμάτων στο διαδίκτυο**

Για την απεικόνιση των μετρήσεων μας, εκτός των χαρτών που εισάγαμε στην εφαρμογή που υλοποιήσαμε, δημιουργήσαμε και μία σελίδα προσβάσιμη από όλους η οποία βρίσκεται στο εικονικό μηχάνημα που δημιουργήσαμε μέσω της υπηρεσίας oceanos. Στη σελίδα αυτή, ο χρήστης διαθέτει την δυνατότητα να επιλέξει τον τύπο του χάρτη, ανάμεσα από ένα πλήθος χαρτών συμπεριλαμβανομένων και των Google maps, πάνω στον οποίο θα απεικονιστούν οι μετρήσεις. Για την απεικόνιση των μετρήσεων, η σελίδα επικοινωνεί με τον Geoserver, ο οποίος ανάλογα με τις επιλογές του χρήστη, αποστέλλει εικόνες με χρήση του πρωτοκόλλου WMS (Web Map Service), μέσω των οποίων απεικονίζονται οι μετρήσεις μας, με παράλληλη υπέρθεση των εικόνων αυτών στον χάρτη.

Η σελίδα για την απεικόνιση των αποτελεσμάτων μέσω του διαδικτύου μπορεί να βρεθεί στον ακόλουθο σύνδεσμο: <http://83.212.111.56/mapsDisplay.wms> .



## 5 Συμπεράσματα και Επεκτάσεις

Στα πλαίσια της παρούσας πτυχιακής, ήρθαμε σε επαφή με ένα πλήθος τεχνολογιών και τεχνικών για την μέτρηση χαρακτηριστικών της δικτυακής σύνδεσης σύγχρονων κινητών συσκευών. Γενικώς παρατηρήσαμε πως οι κατασκευαστές των συσκευών αυτών, θέτουν διάφορους περιορισμούς στην δικτυακή λειτουργία τους οι οποίες σε ορισμένες περιπτώσεις δυσκόλεψαν την υλοποίηση της εφαρμογής μας. Βασιζόμενοι όμως σε καθιερωμένα πρότυπα πάνω στην δικτύωση, καταφέραμε και υλοποιήσαμε τον στόχο της πτυχιακής, να δημιουργήσουμε δηλαδή μία λειτουργική εφαρμογή εύκολη προς την χρήση με λήψη αξιόπιστων μετρήσεων για τον χαρακτηρισμό σύνδεσης στο διαδίκτυο και μία δικτυακή πλατφόρμα εύκολα προσβάσιμη από εφαρμογές συσκευών ανεξαρτήτως του λειτουργικού της συσκευής που τις εκτελεί.

Η εφαρμογή που δημιουργήσαμε, καλύπτει πλήρως τους στόχους που τέθηκαν στα πλαίσια της πτυχιακής, ενώ είναι δυνατή η διανομή της μέσω της πλατφόρμας Play Store. Μέσω ενός φιλικό προς τον χρήστη περιβάλλον διεπαφής, παρουσιάζονται με εύκολο τρόπο τα αποτελέσματα των ελέγχων, ενώ το σύνολο αυτών είναι προσβάσιμο από όλους τους χρηστές. Ωστόσο, υπάρχει πάντοτε η δυνατότητα βελτίωσης ορισμένων χαρακτηριστικών της πτυχιακής. Οπότε παρακάτω θα αναφερθούμε σε ορισμένες πιθανές επέκτασης της εργασίας αυτής.

Αρχικά, μία πιθανή επέκταση θα μπορούσε να είναι η προσθήκη και άλλων ειδών μετρήσεων, με σκοπό τον καλύτερο χαρακτηρισμό της σύνδεσης του χρήστη. Παράδειγμα τέτοιων μετρήσεων, είναι η ένταση του λαμβανόμενου σήματος. Επίσης, μία άλλη πιθανή επέκταση, είναι η εμπλούτιση του γραφικού περιβάλλοντος για την διεξαγωγή μετρήσεων. Παρατηρήσαμε, ότι σε αντίστοιχες εφαρμογές που προσφέρονται στο Play Store, το γραφικό περιβάλλον που διέθεταν ήταν πιο επιδεικτικό, σε σύγκριση με την εφαρμογή μας, στην οποία δημιουργήσαμε ένα απλό και λιτό περιβάλλον πλήρως ικανό να ενημερώσει τον χρήστη για την κατάσταση των ελέγχων που τρέχει. Μία άλλη επέκταση, αποτελεί η προσθήκη περισσότερης ασφάλειας στην ανταλλαγή των δεδομένων με τους εξυπηρέτες της δικτυακής πλατφόρμας. Καθώς στην παρούσα φάση, η εργασία μας δεν συλλέγει δεδομένα ικανά να ταυτοποιήσουν έναν χρήστη, δεν κρίθηκε απαραίτητη η προσθήκη ιδιαίτερης ασφάλειας κατά την μεταφορά των μετρήσεων από την εφαρμογή προς την βάση δεδομένων μέσω της υπηρεσίας διαδικτύου που υλοποιήσαμε, κάτι που ίσως να αλλάξει σε πιθανή μεταγενέστερη επέκταση της εργασίας αυτής. Επιπρόσθετα, θα ήθελα να αναφέρω, πως μία σοφή επέκταση, θα ήταν η χρήση του νέου Google Maps API V3 στην θέση του Google Maps API V2 που ήταν διαθέσιμο κατά την υλοποίηση μας, καθώς παρατηρήσαμε με χρήση εργαλείων που προσφέρει το Eclipse ότι οι χάρτες της Google, καταναλώνουν αρκετά μεγάλο κομμάτι μνήμης κατά την εκτέλεση, κάτι που από πολλούς έχει

χαρακτηρίζεται ως διαρροή μνήμης (Memory leak) και μπορεί να αποτελέσει επιβλαβή παράγοντα στην ορθή εκτέλεση της εφαρμογής μας. Τέλος, θα ήθελα να αναφέρω, πως η παρούσα πλατφόρμα είναι δυνατό να χρησιμοποιηθεί και για συλλογή μετρήσεων και άλλων μεγεθών τα οποία είτε μπορούν να λαμβάνονται από τους αισθητήρες των σύγχρονων κινητών συσκευών, είτε μπορούν να εισάγονται από τους χρήστες με παραδείγματα τέτοιων μεγεθών, να είναι η μέτρηση της ηχορύπανσης για την πρώτη κατηγορία και την καταγραφή γεγονότων που συμβαίνουν στην καθημερινότητα μας (εγκληματικότητα, πυρκαγιές, ...) για την δεύτερη κατηγορία.

## 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 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 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6211483&isnumber=6211473>
- [3] Felix Richter. Android's Smartphone Dominance is Growing [Ηλεκτρονικό] 2/2014. [Παραπομπή: 25/6/2014] Σύνδεσμος: <http://www.statista.com/chart/1899/smartphone-market-share/>
- [4] Felix Richter. Android To Retain Big Lead In Maturing Smartphone Market [Ηλεκτρονικό] 3/2014. [Παραπομπή: 25/6/2014] Σύνδεσμος: <http://www.statista.com/chart/1961/smartphone-market-share-2014/>
- [5] Wikipedia. Android (operating System) [Ηλεκτρονικό] 6/2014. [Παραπομπή: 25/6/2014] Σύνδεσμος: [http://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))
- [6] Android Developers. Dalvic Technical Information [Ηλεκτρονικό]. [Παραπομπή: 25/6/2014] Σύνδεσμος: <http://source.android.com/devices/tech/dalvik/>
- [7] Wikipedia. Dalvik (software) [Ηλεκτρονικό] 6/2014. [Παραπομπή: 25/6/2014] Σύνδεσμος: [http://en.wikipedia.org/wiki/Dalvik\\_\(software\)](http://en.wikipedia.org/wiki/Dalvik_(software))
- [8] Android Developers. Platform Versions [Ηλεκτρονικό]. [Παραπομπή: 25/6/2014] Σύνδεσμος: [https://developer.android.com/about/dashboards/index.html?utm\\_source=ausdroid.net#Platform](https://developer.android.com/about/dashboards/index.html?utm_source=ausdroid.net#Platform)
- [9] Android Developers. Getting started with Android Studio, Early Access Preview, Caution [Ηλεκτρονικό]. [Παραπομπή: 25/6/2014] Σύνδεσμος: <http://developer.android.com/sdk/installing/studio.html>

- [10] HalfApp. Eclipse vs. Android Studio: The battle of the Android IDEs [Ηλεκτρονικό] 15/12/2013. [Παραπομπή: 25/6/2014] Σύνδεσμος: <http://halfapp.com/blog/eclipse-vs-android-studio-battle-android-ides/>
- [11] Eclipse Plugins, Bundles and Products. Android Development Tools for Eclipse [Ηλεκτρονικό]. [Παραπομπή: 25/6/2014] Σύνδεσμος: [http://marketplace.eclipse.org/content/android-development-tools-eclipse#.U5TQHvI\\_vIM](http://marketplace.eclipse.org/content/android-development-tools-eclipse#.U5TQHvI_vIM)
- [12] Android Developers. Google Maps Android API V2 [Ηλεκτρονικό]. [Παραπομπή: 25/6/2014] Σύνδεσμος: <http://developer.android.com/google/play-services/maps.html>
- [13] Android Developers. Connection to the Network [Ηλεκτρονικό]. [Παραπομπή: 25/6/2014] Σύνδεσμος: <http://developer.android.com/training/basics/network-ops/connecting.html>
- [14] Netcraft. June 2013 Web Server Survey [Ηλεκτρονικό] 6/2013. [Παραπομπή: 25/6/2014] Σύνδεσμος: <http://news.netcraft.com/archives/2013/06/06/june-2013-web-server-survey-3.html>
- [15] M. Zack Urlocker. Google Runs MySQL [Ηλεκτρονικό] 12/2005. [Παραπομπή: 25/06/2014] Σύνδεσμος: [http://zurlocker.typepad.com/theopenforce/2005/12/googles\\_use\\_of\\_.html](http://zurlocker.typepad.com/theopenforce/2005/12/googles_use_of_.html)
- [16] Om Malik. Facebook's Insatiable Hunger for Hardware [Ηλεκτρονικό] 25/04/2008. [Παραπομπή: 25/06/2014] Σύνδεσμος: <http://gigaom.com/2008/04/25/facebooks-insatiable-hunger-for-hardware/>
- [17] Geoserver User manual. Working with Databases [Ηλεκτρονικό] 17/09/2013. [Παραπομπή: 25/06/2014] Σύνδεσμος: <http://ares.boundlessgeo.com/geoserver/master/doc/en/user/data/database/index.html>
- [18] Ubuntu Documentation. Screen [Ηλεκτρονικό]. Παραπομπή: 25/06/2014] Σύνδεσμος: <https://help.ubuntu.com/community/Screen>
- [19] MySQL Development. 14.1 The MyISAM Storage Engine [Ηλεκτρονικό]. [Παραπομπή: 25/6/2014] Σύνδεσμος: <http://dev.mysql.com/doc/refman/5.0/en/myisam-storage-engine.html>

[20] Wikipedia. Apache Tomcat [Ηλεκτρονικό] 6/2014. [Παραπομπή: 25/6/2014] Συνδεσμος: [http://en.wikipedia.org/wiki/Apache\\_Tomcat](http://en.wikipedia.org/wiki/Apache_Tomcat)

[21] Wikipedia. Web Container [Ηλεκτρονικό] 5/2014. [Παραπομπή: 25/6/2014] Συνδεσμος: [http://en.wikipedia.org/wiki/Web\\_container](http://en.wikipedia.org/wiki/Web_container)

[22] GeoServer Concepts. 4.4 Layer Group [Ηλεκτρονικό]. <http://girona-geoserver-workshop.readthedocs.org/en/latest/overview/concepts.html#layer-group>

[23] Wikipedia. Styled Layer Descriptor [Ηλεκτρονικό] 2/2014. [Παραπομπή: 25/6/2014] Συνδεσμος: [http://en.wikipedia.org/wiki/Styled\\_Layer\\_Descriptor](http://en.wikipedia.org/wiki/Styled_Layer_Descriptor)



## A. Appendices

### Appendix A: Κώδικας MainActivity Κλάσης.

#### Κώδικας της MainActivity κλάσης:

```
package com.koros.speedtester;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.Scanner;
import com.koros.speedtester.R;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.GooglePlayServicesUtil;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.MapFragment;
import com.google.android.gms.maps.model.TileOverlay;
import com.google.android.gms.maps.model.TileOverlayOptions;
import com.google.android.gms.maps.model.TileProvider;
import android.location.LocationManager;
import android.net.ConnectivityManager;
import android.os.Bundle;
import android.provider.Settings;
import android.app.AlertDialog;
import android.app.Dialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.DialogInterface.OnDismissListener;
import android.content.Intent;
import android.content.IntentFilter;
import android.content.res.Configuration;
import android.graphics.Color;
import android.support.v4.app.FragmentActivity;
import android.view.Menu;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ProgressBar;
import android.widget.Spinner;
import android.widget.TabHost;
import android.widget.TabHost.TabSpec;
import android.widget.TextView;
import android.widget.Toast;
import android.view.*;
public class MainActivity extends FragmentActivity implements OnItemSelectedListener {

    public static GoogleMap map=null;
    public static TextView connection;//Type of connection
    public static Button TestStart;//Starting Test button Variable
    public TileOverlay mapOverlay;
    public static boolean testUnderway;//Boolean variable to signal that a measurement is
    underway and that the Test Connection tab is not to be altered by some
    NetworkChangeReceiver operations.
    public static Spinner mapType;
    // Value 1-8 are TextViews for displaying the scale of measurements
    public TextView value1;
    public TextView value2;
    public TextView value3;
    public TextView value4;
    public TextView value5;
    public TextView value6;
    public TextView value7;
    public TextView value8;
    //
    public static boolean LocationErrorMessage=false;
    public static Spinner tileType;
    public static boolean ncrRegistered;//Network Changed Receiver variable
    public static TextView ping1;
    public static TextView status2;
    public static TextView ping2;
    public static TextView downlink;
    public static TextView uplink;
```

```

public static TextView status;
public static TextView pserver;
public static TextView rserver;
public static ProgressBar pb;
static int GPSAvailable;//Globally accessible variable for determining if Google Play
services a vital part for Google Maps to function are present
public static MainActivity alpha;
//public static TextView connectionInfo;
public static View Map;
NetworkChangeReceiver ncr=null;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_getting_location);
    ncrRegistered=false;
    alpha=this;//Instance of this Object
    OptionsFileHandler();
    GPSAvailable=GooglePlayServicesUtil.isGooglePlayServicesAvailable(alpha);//Check
    if Google Play Services are available
    if(GPSAvailable==ConnectionResult.SUCCESS){//If Google Play Services are available
        if(ncrRegistered==false){//If Network Change Receiver is not registered
            ncr=new NetworkChangeReceiver();
            IntentFilter ifr=new IntentFilter();
            ifr.addAction("android.net.conn.CONNECTIVITY_CHANGE");
            ifr.addAction("android.location.PROVIDERS_CHANGED");
            ifr.addAction("android.net.wifi.WIFI_STATE_CHANGED");
            this.registerReceiver(ncr, ifr);//Register the Network Change
            Receiver
            ncrRegistered=true;
        }
        map=(MapFragment)getFragmentManager().
        findFragmentById(R.id.map).getMap();//Get instance of map fragment
        map.setMapType(GoogleMap.MAP_TYPE_HYBRID);
        //Check if you can view My Location on the map
        ConnectivityManager connMgr = (ConnectivityManager)
        getBaseContext().getSystemService(Context.CONNECTIVITY_SERVICE);
        android.net.NetworkInfo wifi =
        connMgr.getNetworkInfo(ConnectivityManager.TYPE_WIFI);
        LocationManager manager = (LocationManager)
        getSystemService(Context.LOCATION_SERVICE);
        android.net.NetworkInfo mobile =
        connMgr.getNetworkInfo(ConnectivityManager.TYPE_MOBILE);
        if(!wifi.isConnected() && !mobile.isConnected() && (!
        manager.isProviderEnabled(LocationManager.GPS_PROVIDER) || (!
        manager.isProviderEnabled(LocationManager.NETWORK_PROVIDER))){
            map.setMyLocationEnabled(false);
        }
        else{
            map.setMyLocationEnabled(true);
        }
        map.getUiSettings().setMyLocationButtonEnabled(true);
        Map=findViewById(R.id.map);
    }
    else{//If Google Play Services are not available
        int requestCode = 10;
        Dialog dialog =GooglePlayServicesUtil.getErrorDialog(GPSAvailable, this,
        requestCode);
        dialog.show();//Display message that takes you to play store to download
        Google Play Services
    }
    setUpSpinners();
    connection=(TextView)findViewById(R.id.ConnectionType);
    TestStart=(Button)findViewById(R.id.ipb);
    //Initialize and Setup the 3 tabs
    TabHost tabHost=(TabHost)findViewById(R.id.tabHost);
    tabHost.setup();
    TabSpec spec1=tabHost.newTabSpec("Tab 1");
    spec1.setContent(R.id.tab1);
    spec1.setIndicator("Maps");
    TabSpec spec2=tabHost.newTabSpec("Tab 2");
    spec2.setIndicator("Connection Type (os)");
    spec2.setContent(R.id.tab2);
    TabSpec spec3=tabHost.newTabSpec("Tab 3");
    spec3.setIndicator("Test Connection");
    spec3.setContent(R.id.tab3);
    tabHost.addTab(spec1);
    tabHost.addTab(spec2);
    tabHost.addTab(spec3);
    tabHost.invalidate();
    //End of Tab initialization and setup

```



```

        MainActivity.connection.setText( NetworkUtility.
        getConnectivityStatusString(getApplicationContext()));
        MainActivity.connection.invalidate();
        setUpTesterUI();
    }

    public static void ErrorMessageDisplay(){//Error message for aborting the test
        Toast.makeText(MainActivity.alpha, "Aborting Test due to connection issues",
        Toast.LENGTH_SHORT).show();
    }

    public void setUpTesterUI(){//Initialize the Speed Tester GUI
        ping1=(TextView)findViewById(R.id.ping1);
        ping2=(TextView)findViewById(R.id.ping2);
        downlink=(TextView)findViewById(R.id.downlink);
        uplink=(TextView)findViewById(R.id.uplink);
        status=(TextView)findViewById(R.id.status);
        status2=(TextView)findViewById(R.id.status2);
        pserver=(TextView)findViewById(R.id.pserver);
        rserver=(TextView)findViewById(R.id.rserver);
        pb=(ProgressBar)findViewById(R.id.ProgressBar);
        pb.setVisibility(ProgressBar.INVISIBLE);
        pb.invalidate();
        resetTesterUI();
    }

    public static void resetTesterUI(){//Reset the Speed Tester GUI
        MainActivity.ping1.setText("...");
        MainActivity.ping1.invalidate();
        MainActivity.ping2.setText("...");
        MainActivity.ping1.invalidate();
        MainActivity.downlink.setText("...");
        MainActivity.downlink.invalidate();
        MainActivity.uplink.setText("...");
        MainActivity.downlink.invalidate();
        MainActivity.pb.setVisibility(ProgressBar.GONE);
        MainActivity.pb.invalidate();
        pserver.setText("Ping "+getDefaultServer());
        pserver.invalidate();
        rserver.setText("Ping www.google.gr");
        rserver.invalidate();
    }

    public void setUpSpinners(){//Sets up the spinners for viewing multiple kinds o data as
    well as numerous kinds of maps
        value1=(TextView)findViewById(R.id.Value1);
        value2=(TextView)findViewById(R.id.Value2);
        value3=(TextView)findViewById(R.id.Value3);
        value4=(TextView)findViewById(R.id.Value4);
        value5=(TextView)findViewById(R.id.Value5);
        value6=(TextView)findViewById(R.id.Value6);
        value7=(TextView)findViewById(R.id.Value7);
        value8=(TextView)findViewById(R.id.Value8);
        //tileSpinner initialization
        tileType=(Spinner)findViewById(R.id.TileSpinner);
        ArrayAdapter<CharSequence> adapter =
        ArrayAdapter.createFromResource(this,R.array.tileLayers,
        android.R.layout.simple_spinner_item);
        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        tileType.setAdapter(adapter);
        tileType.setOnItemSelectedListener(this);
        mapType=(Spinner)findViewById(R.id.MapSpinner);
        ArrayAdapter<CharSequence> adapter1 =
        ArrayAdapter.createFromResource(this,R.array.mapType,
        android.R.layout.simple_spinner_item);
        adapter1.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        mapType.setAdapter(adapter1);
        mapType.setOnItemSelectedListener(this);
        //tileSpinnerInitialization ended
    }

    @Override
    protected void onRestart(){//Method call when application is replaced in focus.
        super.onRestart();
        if(ncrRegistered==false){//If NetworkChangeReceiver is not registered
            IntentFilter ifr=new IntentFilter();
            ifr.addAction("android.net.conn.CONNECTIVITY_CHANGE");
            ifr.addAction("android.location.PROVIDERS_CHANGED");
            ifr.addAction("android.net.wifi.WIFI_STATE_CHANGED");

            ncr=new NetworkChangeReceiver();

            this.registerReceiver(ncr, ifr); //Re-register the Reciever

```

```

        ncrRegistered=true; //The receiver is registered
    }

    }

    @Override
    protected void onResume() {
        super.onResume();
        if (ncrRegistered==false) { //If NetworkChangeReceiver is not registered
            IntentFilter ifr=new IntentFilter();
            ifr.addAction("android.net.conn.CONNECTIVITY_CHANGE");
            ifr.addAction("android.location.PROVIDERS_CHANGED");
            ifr.addAction("android.net.wifi.WIFI_STATE_CHANGED");

            ncr=new NetworkChangeReceiver();

            this.registerReceiver(ncr, ifr); //Re-register the Reciever
            ncrRegistered=true; //The receiver is registered
        }
        GPSAvailable=GooglePlayServicesUtil.isGooglePlayServicesAvailable(alpha); //Check
if Google Play Services are available
        if (GPSAvailable!=ConnectionResult.SUCCESS) { //If Google Play Services are not
available
            int requestCode = 10;
            Dialog dialog =GooglePlayServicesUtil.getErrorDialog(GPSAvailable, this,
requestCode);
            dialog.show(); //Display message that takes you to play store to download Google
Play Services
        } else if (map==null) { //If Google Play Services are available but we haven't instantiated
the map object
            map=( (MapFragment) getFragmentManager().findFragmentById(R.id.map)).getMap(); //Get
instance of map fragment
            map.setMapType(GoogleMap.MAP_TYPE_HYBRID);
            //Check if you can view My Location on the map
            ConnectivityManager connMgr = (ConnectivityManager)
getBaseContext().getSystemService(Context.CONNECTIVITY_SERVICE);
            android.net.NetworkInfo wifi =
connMgr.getNetworkInfo(ConnectivityManager.TYPE_WIFI);
            LocationManager manager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);
            android.net.NetworkInfo mobile =
connMgr.getNetworkInfo(ConnectivityManager.TYPE_MOBILE);
            if ((!wifi.isConnected()) && (!mobile.isConnected()) && (!
manager.isProviderEnabled(LocationManager.GPS_PROVIDER)) || (!
manager.isProviderEnabled(LocationManager.NETWORK_PROVIDER))) {
                map.setMyLocationEnabled(false);
            }
            else {
                map.setMyLocationEnabled(true);
            }

            map.getUiSettings().setMyLocationButtonEnabled(true);
            Map=findViewById(R.id.map);
        }
    }

    private void MapDownlinkData() { //Display Downlink speed results
        if (map!=null) {
            if (mapOverlay!=null) {
                mapOverlay.remove();
                mapOverlay.clearTileCache();
            }

            TileProvider tileProvider = TileProviderFactory.getTileProvider("Downlink");
            mapOverlay = map.addTileOverlay(new
TileOverlayOptions().tileProvider(tileProvider)); //Here I add the appropriate tile
            value1.setTextColor(Color.RED);
            value1.setText("> 32768 kbps");
            value1.setVisibility(TextView.VISIBLE);
            value1.invalidate();
            value2.setTextColor(Color.rgb(255,165, 0));
            value2.setText(" 32768 kbps");
            value2.setVisibility(TextView.VISIBLE);
            value2.invalidate();
            value3.setTextColor(Color.YELLOW);
            value3.setText(" 16384 kbps");
            value3.setVisibility(TextView.VISIBLE);
            value3.invalidate();
            value4.setTextColor(Color.GREEN);
            value4.setText(" 8192 kbps");
            value4.setVisibility(TextView.VISIBLE);
            value4.invalidate();

```

```

        value5.setTextColor(Color.rgb(0, 255, 255));
        value5.setText("    4096 kbps");
        value5.setVisibility(View.VISIBLE);
        value5.invalidate();
        value6.setTextColor(Color.BLUE);
        value6.setText("    2048 kbps");
        value6.setVisibility(View.VISIBLE);
        value6.invalidate();
        value7.setTextColor(Color.LTGRAY);
        value7.setText("    1024 kbps");
        value7.setVisibility(View.VISIBLE);
        value7.invalidate();
        value8.setTextColor(Color.BLACK);
        value8.setText("< 512 kbps");
        value8.setVisibility(View.VISIBLE);
        value8.invalidate();
        System.gc();
    }

}

private void MapUplinkData() { //Display Uplink speed Results
    if (map != null) {
        if (mapOverlay != null) {
            mapOverlay.remove();
            mapOverlay.clearTileCache();
        }
        TileProvider tileProvider = TileProviderFactory.getTileProvider("Uplink");
        mapOverlay = map.addTileOverlay(new
TileOverlayOptions().tileProvider(tileProvider)); //Here I add the appropriate tile
        value1.setTextColor(Color.RED);
        value1.setText("> 32768 kbps");
        value1.setVisibility(View.VISIBLE);
        value1.invalidate();
        value2.setTextColor(Color.rgb(255, 165, 0));
        value2.setText("    32768 kbps");
        value2.setVisibility(View.VISIBLE);
        value2.invalidate();
        value3.setTextColor(Color.YELLOW);
        value3.setText("    16384 kbps");
        value3.setVisibility(View.VISIBLE);
        value3.invalidate();
        value4.setTextColor(Color.GREEN);
        value4.setText("    8192 kbps");
        value4.setVisibility(View.VISIBLE);
        value4.invalidate();
        value5.setTextColor(Color.rgb(0, 255, 255));
        value5.setText("    4096 kbps");
        value5.setVisibility(View.VISIBLE);
        value5.invalidate();
        value6.setTextColor(Color.BLUE);
        value6.setText("    2048 kbps");
        value6.setVisibility(View.VISIBLE);
        value6.invalidate();
        value7.setTextColor(Color.LTGRAY);
        value7.setText("    1024 kbps");
        value7.setVisibility(View.VISIBLE);
        value7.invalidate();
        value8.setTextColor(Color.BLACK);
        value8.setText("< 512 kbps");
        value8.setVisibility(View.VISIBLE);
        value8.invalidate();
        System.gc();
    }

}

private void MapPing1Data() { //Display personal server Ping results
    if (map != null) {
        if (mapOverlay != null) {
            mapOverlay.remove();
            mapOverlay.clearTileCache();
        }
        TileProvider tileProvider = TileProviderFactory.getTileProvider("Ping1");
        mapOverlay = map.addTileOverlay(new
TileOverlayOptions().tileProvider(tileProvider)); //Here I add the appropriate tile
        value1.setTextColor(Color.RED);
        value1.setText("< 10 ms");
        value1.setVisibility(View.VISIBLE);
        value1.invalidate();
        value2.setTextColor(Color.rgb(255, 165, 0));

```

```

        value2.setText(" 20 ms");
        value2.setVisibility(TextView.VISIBLE);
        value2.invalidate();
        value3.setTextColor(Color.YELLOW);
        value3.setText(" 40 ms");
        value3.setVisibility(TextView.VISIBLE);
        value3.invalidate();
        value4.setTextColor(Color.GREEN);
        value4.setText(" 80 ms");
        value4.setVisibility(TextView.VISIBLE);
        value4.invalidate();
        value5.setTextColor(Color.rgb(0, 255, 255));
        value5.setText(" 160 ms");
        value5.setVisibility(TextView.VISIBLE);
        value5.invalidate();
        value6.setTextColor(Color.BLUE);
        value6.setText(" 320 ms");
        value6.setVisibility(TextView.VISIBLE);
        value6.invalidate();
        value7.setTextColor(Color.LTGRAY);
        value7.setText(" 640 ms");
        value7.setVisibility(TextView.VISIBLE);
        value7.invalidate();
        value8.setTextColor(Color.BLACK);
        value8.setText("> 640 ms");
        value8.setVisibility(TextView.VISIBLE);
        value8.invalidate();
        System.gc();
    }

}

private void MapPing2Data() { //Display Google Ping results
    if (map != null) {
        if (mapOverlay != null) {
            mapOverlay.remove();
            mapOverlay.clearTileCache();
        }
        TileProvider tileProvider = TileProviderFactory.getTileProvider("Ping2");
        mapOverlay = map.addTileOverlay(new
TileOverlayOptions().tileProvider(tileProvider)); //Here I add the appropriate tile
        value1.setTextColor(Color.RED);
        value1.setText("< 10 ms");
        value1.setVisibility(TextView.VISIBLE);
        value1.invalidate();
        value2.setTextColor(Color.rgb(255, 165, 0));
        value2.setText(" 20 ms");
        value2.setVisibility(TextView.VISIBLE);
        value2.invalidate();
        value3.setTextColor(Color.YELLOW);
        value3.setText(" 40 ms");
        value3.setVisibility(TextView.VISIBLE);
        value3.invalidate();
        value4.setTextColor(Color.GREEN);
        value4.setText(" 80 ms");
        value4.setVisibility(TextView.VISIBLE);
        value4.invalidate();
        value5.setTextColor(Color.rgb(0, 255, 255));
        value5.setText(" 160 ms");
        value5.setVisibility(TextView.VISIBLE);
        value5.invalidate();
        value6.setTextColor(Color.BLUE);
        value6.setText(" 320 ms");
        value6.setVisibility(TextView.VISIBLE);
        value6.invalidate();
        value7.setTextColor(Color.LTGRAY);
        value7.setText(" 640 ms");
        value7.setVisibility(TextView.VISIBLE);
        value7.invalidate();
        value8.setTextColor(Color.BLACK);
        value8.setText("> 640 ms");
        value8.setVisibility(TextView.VISIBLE);
        value8.invalidate();
        System.gc();
    }

}

private void MapNoData() { //Display no data and remove any labels
    if (map != null) {
        if (mapOverlay != null) {
            mapOverlay.remove();

```

```

        mapOverlay.clearTileCache();
    }
    map.clear();
    value1.setVisibility(View.INVISIBLE);
    value2.setVisibility(View.INVISIBLE);
    value3.setVisibility(View.INVISIBLE);
    value4.setVisibility(View.INVISIBLE);
    value5.setVisibility(View.INVISIBLE);
    value6.setVisibility(View.INVISIBLE);
    value7.setVisibility(View.INVISIBLE);
    value8.setVisibility(View.INVISIBLE);
    System.gc();
}

@Override
protected void onPause() { //Method called when application is out of focus
    super.onPause();
    if (ncrRegistered) {
        this.unregisterReceiver(ncr);
        ncrRegistered=false;
    }
}

@Override
public void onConfigurationChanged(Configuration newConfig) {
    super.onConfigurationChanged(newConfig);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);

    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) { //Handle options selection
    switch (item.getItemId()) {

        case R.id.exit:
            finish();
            break;

    }

    return false;
}

}

public static void OptionsFileHandler() { //Set directories and IPs in internal storage
    File f=new
File("/data/data/com.koros.speedtester/MapsLocationServicesDir");//directory that only our
application can access
    if (!f.exists()) {
        f.mkdir();
    }
    File f1=new File(f, "Measurement");
    if (!f1.exists()) {
        f1.mkdir();
    }
    f=new File(f, "Options");
    if (!f.exists()) {
        f.mkdir();
    }
    File buff=f;
    f=new File(f, "options.txt");
    if (!f.exists()) {
        try {
            f.createNewFile();
            FileOutputStream fos=new FileOutputStream(f);
            fos.write("83.212.111.56".getBytes());

```

```

        fos.close();
        fos=new FileOutputStream(new File(buff,"DefaultOptions.txt"));
        fos.write("83.212.111.56".getBytes());
        fos.close();

    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

}

}

static public String getDefaultServer(){//Retrieve IP of server
    String serverAddress=null;
    try {
        File f=new
File("/data/data/com.koros.speedtester/MapsLocationServicesDir/Options/DefaultOptions.txt");
        Scanner rf = new Scanner(f);
        serverAddress=rf.nextLine();
        rf.close();
    } catch (FileNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return serverAddress;
}

static public String getServerAddress(){
    File f=new
File("/data/data/com.koros.speedtester/MapsLocationServicesDir/Options/options.txt");
    String serverAddress=null;
    boolean Found=false;
    do{
        try {
            if(Found){
                f=new
File("/data/data/com.koros.speedtester/MapsLocationServicesDir/Options/DefaultOptions.txt");
                Found=false;
            }
            Scanner fr=new Scanner(f);
            serverAddress=fr.nextLine();
            fr.close();
        } catch (FileNotFoundException e) {
            Toast msg=new Toast(MainActivity.alpha);
            msg.setText("Unable to find "+f.getName()+"\nAttempting to retry.");
            msg.setDuration(Toast.LENGTH_SHORT);
            Found=true;
        }
    }while(Found);

    return serverAddress;
}

public static void LocationServicesNeededMessage(){
    MainActivity.LocationErrorMessage=true;
    AlertDialog ad= new
AlertDialog.Builder(MainActivity.alpha).setMessage("Location Services Needed For The Speed
Test").setPositiveButton("Enable Location Services",
        new DialogInterface.OnClickListener() {

            public void onClick(DialogInterface dialog,int id) {

                MainActivity.alpha.startActivity(new
Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS));
                MainActivity.LocationErrorMessage=false;

            }

        }).setNegativeButton("Exit",new DialogInterface.OnClickListener() {

            public void onClick(DialogInterface dialog,int id) {
                dialog.cancel();
                MainActivity.LocationErrorMessage=false;
            }

        }).create();
    ad.setCancelable(false);
}

```

```

        ad.setOnDismissListener(new OnDismissListener() {

            @Override
            public void onDismiss(DialogInterface

dialog) {

        MainActivity.LocationErrorMessage=false;

        });
        ad.show();

    }

    public void startTester(View view){//Start Test Button Listener

        //callClientServerCommunicationMethod
        LocationManager manager = (LocationManager) MainActivity.alpha.getSystemService(
Context.LOCATION_SERVICE );
        if(!manager.isProviderEnabled(LocationManager.NETWORK_PROVIDER)){
            if(!MainActivity.LocationErrorMessage)
                MainActivity.LocationServicesNeededMessage();

        }
        else{
            testUnderway=true;
            MainActivity.TestStart.setClickable(false);
            MainActivity.TestStart.setBackgroundColor(Color.BLACK);
            MainActivity.TestStart.setTextColor(Color.GREEN);
            MainActivity.TestStart.setText("Test Started");
            MainActivity.TestStart.invalidate();
            MainActivity.ping1.setText("...");
            MainActivity.ping1.invalidate();
            MainActivity.ping2.setText("...");
            MainActivity.ping1.invalidate();
            MainActivity.downlink.setText("...");
            MainActivity.ping1.invalidate();
            MainActivity.uplink.setText("...");
            MainActivity.ping1.invalidate();
            status.setText("Test Active");
            status.setTextColor(Color.GREEN);
            status.setBackgroundColor(Color.BLACK);
            status2.setBackgroundColor(Color.BLACK);
            status.invalidate();
            System.gc();
            SpeedTester st=new SpeedTester();
            st.startCommand();

        }
    }
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position,long id)
    { //Handler the item selection for both Spinners

        switch((((Spinner)parent).getId())){
            case R.id.TileSpinner:
                String TileType=((String)((TextView)view).getText());
                if(TileType.equalsIgnoreCase("Downlink Speed")){
                    MapDownlinkData();
                }
                else if(((String)
((TextView)view).getText()).equalsIgnoreCase("Uplink Speed")){
                    MapUplinkData();
                }
                else if(((String)((TextView)view).getText()).equalsIgnoreCase("Ping Personal
Server")){
                    MapPing1Data();
                }
                else if(((String)((TextView)view).getText()).equalsIgnoreCase("Ping Google")){
                    MapPing2Data();
                }
                else if(((String)((TextView)view).getText()).equalsIgnoreCase("No Data")){
                    MapNoData();
                }
                Toast.makeText(alpha,"Displaying: "+((TextView)view).getText()+"
", Toast.LENGTH_SHORT).show();
                break;

```

```
        case R.id.MapSpinner:
            String MapType=((String)((TextView)view).getText());
            if(MapType.equalsIgnoreCase("Hybrid")){
                if(map!=null){
                    map.setMapType(GoogleMap.MAP_TYPE_HYBRID);
                }
            }
            else if(MapType.equalsIgnoreCase("Satelite")){
                if(map!=null){
                    map.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
                }
            }
            else if(MapType.equalsIgnoreCase("Map")){
                if(map!=null){
                    map.setMapType(GoogleMap.MAP_TYPE_NORMAL);
                }
            }
            else if(MapType.equalsIgnoreCase("Terrain")){
                if(map!=null){
                    map.setMapType(GoogleMap.MAP_TYPE_TERRAIN);
                }
            }
            break;
    }
}
@Override
public void onNothingSelected(AdapterView<?> parent) {
    // TODO Auto-generated method stub
}
}
```



## Appendix B: Κώδικας SpeedTester, ServerComm και RunTest Κλάσεων.

### Κώδικας SpeedTester, ServerComm, RunTest κλάσεων:

```
package com.koros.speedtester;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.net.Socket;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.Random;
import java.util.Scanner;
import java.util.StringTokenizer;

import org.apache.http.NameValuePair;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;

import android.content.Context;
import android.graphics.Color;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationManager;
import android.os.AsyncTask;
import android.os.Build;
import android.widget.ProgressBar;
import android.widget.Toast;
public class SpeedTester {
    //Name of file that holds measurements
    final String fileName="Measurement.txt";
    public static String connectionType;
    String command;
    String serverAddress=null;
    //time stamp for database
    Date tStamp=new Date();
    String outputS=new String();
    final File output=new
File("/data/data/com.koros.speedtester/MapsLocationServicesDir/Measurement/"+fileName);
    FileOutputStream fos;
    private int serverPort;
    Ping ping=new Ping();

    public void startCommand(){
        RunTest rt=new RunTest();
        getDefaultServer();
        try {
            //The server port at which our remote Java server awaits a connection
            serverPort=9222;
            //Opening stream to file with file descriptor output with 'false' in order
not to append
            fos = new FileOutputStream(output,false);

            //From here
            MainActivity.ping1.setBackgroundColor(Color.LTGRAY);
            MainActivity.ping1.invalidate();
            MainActivity.pb.setVisibility(ProgressBar.VISIBLE);
            MainActivity.pb.invalidate();
            //to here we enable a progress bar to show the measurements process has
started
            //double variable for holding the longitude
            double longitude;
```

```

//double variable for holding the latitude
double latitude;
//If we do not have a reference of the Google Maps object
if (MainActivity.map==null) {

    LocationManager lm = ((LocationManager)
MainActivity.alpha.getSystemService(Context.LOCATION_SERVICE)); //Instantiate location manager
    //Instantiate Criteria Object for selecting best Provider
    Criteria criteria = new Criteria();
    // Getting the name of the best provider. The second parameter is set
to true so we get the best available provider
    String provider = lm.getBestProvider(criteria, true);
    // Getting Location
    Location location = lm.getLastKnownLocation(provider);
    //We retrieve longitude from the Location object
    longitude=location.getLongitude();
    //We retrieve longitude from the Location object
    latitude=location.getLatitude();
} //If we can not get our location from the Google Maps we use this
alternative

else if (MainActivity.map.getLocation()==null) {

    LocationManager lm = ((LocationManager)
MainActivity.alpha.getSystemService(Context.LOCATION_SERVICE)); //Instantiate location manager
    //Instantiate Criteria Object for selecting best Provider
    Criteria criteria = new Criteria();

    // Getting the name of the best provider. The second parameter is set
to true so we get the best available provider
    String provider = lm.getBestProvider(criteria, true);
    // Getting Location
    Location location = lm.getLastKnownLocation(provider);
    //We retrieve longitude from the Location object
    longitude=location.getLongitude();
    //We retrieve longitude from the Location object
    latitude=location.getLatitude();
} //if we can get our location from Google Maps
else{
    longitude=MainActivity.map.getLocation().getLongitude();
    latitude=MainActivity.map.getLocation().getLatitude();
}
//we write longitude to the file holding all the data to be send to the
server after the measurements are taken
fos.write(("long:"+longitude+"\n").getBytes());
//we write latitude to the file holding all the data to be send to the server
after the measurements are taken
fos.write(("lat:"+latitude+"\n").getBytes());
//we change the format of the Date object
DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
Date date = new Date();
//we write the newly formatted date into the file
fos.write(("timeDate:"+dateFormat.format(date)+"\n").getBytes());
//we close the file
fos.close();
// we initiate the RunTest thread execution
rt.execute(new Void[]{});

//in every catch block in this file we perform GUI operations for visual
representation of any mistakes
} catch (FileNotFoundException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
    MainActivity.ErrorMessageDisplay();
    MainActivity.testUnderway=false;
    //Enables test starting button when connection is present
    MainActivity.TestStart.setClickable(true);
    MainActivity.TestStart.setBackgroundColor(Color.LTGRAY);
    MainActivity.TestStart.setTextColor(Color.BLACK);
    MainActivity.TestStart.setText("Start Test");
    MainActivity.TestStart.invalidate();
    MainActivity.status.setTextColor(Color.BLACK);
    MainActivity.status.setText("Test Inactive");

    MainActivity.status.setBackgroundColor(Color.TRANSPARENT);
    MainActivity.status2.setBackgroundColor(Color.TRANSPARENT);
    MainActivity.downlink.setBackgroundColor(Color.TRANSPARENT);
    MainActivity.downlink.invalidate();
    MainActivity.uplink.setBackgroundColor(Color.TRANSPARENT);
    MainActivity.uplink.invalidate();
}

```



```

        e.printStackTrace();
    }

    } //Class extends AsyncTask so that we can perform network operations. ServerComm class
    performs the measurements transport via HTTP Post request in our php Web Service.
    public class ServerComm extends AsyncTask<Void,Void,Void>{
        //Method that connects to index.php and sends the measurements.
        public void sendDataToServer() throws Exception{

            HttpClient client=new DefaultHttpClient();
            //URL of device to receive the post request
            HttpPost post=new HttpPost("http://"+serverAddress+"/index.php");

            //List of name value pairs of data to be sent
            List<NameValuePair> pairs=new ArrayList<NameValuePair>();
            //Start scanning file that contains the data to be sent
            Scanner scanner=new Scanner(output);
            String linebuff;
            //Loop for retrieving all data
            while(scanner.hasNext()){
                linebuff=scanner.nextLine();
                //tokenization of line
                StringTokenizer st=new StringTokenizer(linebuff,":");
                //If we encounter the timeDate field the tokenization procedure is
                altered due to the format of timedate
                if(linebuff.contains("timeDate")){
                    String td=st.nextToken();
                    String tdValue=new String();
                    while(st.hasMoreTokens()){
                        tdValue+=st.nextToken();
                        if(st.hasMoreTokens()){
                            tdValue+=":";
                        }
                    }
                    pairs.add(new BasicNameValuePair(td,tdValue));
                }else{
                    pairs.add(new
BasicNameValuePair(st.nextToken(),st.nextToken()));
                }
            }
            //Connect the Post request with the pairs of values
            post.setEntity(new UrlEncodedFormEntity(pairs));
            //Send the Post request
            client.execute(post);

        }
        @Override
        protected Void doInBackground(Void... arg0) {
            try { //if clause to check if there was an error with the measurement
                taking.
                    if(!
((String)MainActivity.status.getText().toString()).equalsIgnoreCase("Aborting Test due to
Connection Issues!")){
                        sendDataToServer();
                    }
            } catch (Exception e) {
                MainActivity.ErrorMessageDisplay();
                MainActivity.status.setText("Terminating Test!");
                MainActivity.status.setTextColor(Color.RED);

                MainActivity.status.setBackgroundColor(Color.TRANSPARENT); MainActivity.status2.setBackgroundColor
                (Color.TRANSPARENT);

                MainActivity.status.invalidate();
                MainActivity.TestStart.setClickable(true);
                MainActivity.TestStart.setBackgroundColor(Color.LTGRAY);
                MainActivity.TestStart.setTextColor(Color.BLACK);
                MainActivity.TestStart.setText("Start Test");
                MainActivity.TestStart.invalidate();
                MainActivity.testUnderway=false;
            }
            return null;
        }
        @Override
        protected void onPostExecute(Void arg0){
            //Operations performed after the measurements are sent to the
            server.Mostly GUI transitions of the Test Connection Tab.

```

```

        if(!
((String)MainActivity.status.getText().toString()).equalsIgnoreCase("Terminating Test!")){
            //Toast message to inform user that measurement were sent to the
server.
            Toast.makeText(MainActivity.alpha, "Measurements sent",
Toast.LENGTH_SHORT).show();
            MainActivity.TestStart.setClickable(true);
            MainActivity.TestStart.setBackgroundColor(Color.LTGRAY);
            MainActivity.TestStart.setTextColor(Color.BLACK);
            MainActivity.TestStart.setText("Start Test");
            MainActivity.TestStart.invalidate();
            MainActivity.status.setText("Test Inactive");
            MainActivity.status.setTextColor(Color.BLACK);
            MainActivity.status.setBackgroundColor(Color.TRANSPARENT);
            MainActivity.status2.setBackgroundColor(Color.TRANSPARENT);
            MainActivity.status.invalidate();
            MainActivity.pb.setVisibility(ProgressBar.INVISIBLE);
            MainActivity.pb.invalidate();
            MainActivity.testUnderway=false;
        }
        else{
            Toast.makeText(MainActivity.alpha, "Measurements not sent",
Toast.LENGTH_SHORT).show(); //Toast message to inform user that were not sent to the server
            MainActivity.testUnderway=false;
        }
    }

    //Class extends AsyncTask and therefore is executed as a thread. Performs measurement
taking.
    class RunTest extends AsyncTask<Void,String,Void>{
        //Socket that connects with Java Server
        Socket s;
        DataInputStream dis;
        DataOutputStream dos;
        //Buffer byte array for sending data to measure Uplink Speed
        byte[] buffer = new byte[1200000];
        //Buffer byte array for receiving data to measure Downlink Speed
        byte[] bufferrec ;
        //Fill the buffer with random bytes
        public void randomDataGenerator(){
            Random r=new Random();
            r.nextBytes(buffer);
        }

        //Method that handles the receiving and sending of data to measure Uplink and
Downlink
        public void packetHandeling(boolean receiv){
            //If we are measuring downlink
            if(receiv){
                try {
                    //Reads data from server
                    dis.readFully(bufferrec);
                } catch (IOException ex) {
                    publishProgress("error");
                }

                //If we are measuring Uplink
            } else{
                try {
                    int i=0;
                    OutputStream os=s.getOutputStream();
                    //Send data in bursts. Repeat until i reaches the same value as length
buffer
                    while(i<buffer.length){
                        //writes bytes from buffer starting from i. The number of bytes
that are written starts at 50000 bytes until the time when i+50000 exceeds the limit of the
array, then we send the rest of the data
                        os.write(buffer,i, (i+50000)>buffer.length?buffer.length-i:50000);
                        os.flush();
                        i+=(i+50000)>buffer.length?buffer.length-i:50000;
                    }

                } catch (IOException ex) {
                    publishProgress("error");
                }
            }
        }
    }
}

```

```

@Override
protected Void doInBackground(Void... arg0) {
    // TODO Auto-generated method stub
    try {
        //Ping Personal Server
        String ping1=ping.pingRun(true);
        if(ping1==null){
            //if ping returns null terminate measurement Procedure
            publishProgress("error");
        }else{
            //else send ping1, value pair to onProgressUpdate to be written
            publishProgress("ping1:"+ping1);
        }
        //Open socket to server
        s=new Socket(serverAddress,serverPort);
        //Enable TCPNoDelayOption
        s.setTcpNoDelay(true);
        //Set the send BufferSize of the device to the same size as the
        server
        s.setSendBufferSize(8192);
        dis=new DataInputStream(s.getInputStream());
        dos=new DataOutputStream(s.getOutputStream());
        //Send Ack to server
        dos.writeUTF("ack");
        //Fill buffer with random bytes
        randomDataGenerator();
        //Send the buffer size to the server
        dos.writeInt(buffer.length);
        //time before uplink measurement in milliseconds
        long timeBefor=System.currentTimeMillis();
        //Perform uplink speed test
        packetHandeling(false);
        //time after uplink measurement in milliseconds
        long timeAfter=System.currentTimeMillis();
        //getting bufferrec size from server
        int recBuffSize=dis.readInt();
        //Instantiating bufferrec for downlink measurement
        bufferrec=new byte[recBuffSize];
        //Send measured speed of uplink in Mbps to onProgressUpdate
        publishProgress("uplink:" + ((buffer.length*8)/1000.0)/((timeAfter-
timeBefor)/1000.0 ));
        //Send Ack to server
        dos.writeUTF("ack");
        //time before downlink measured in milliseconds
        long timeBefor1=System.currentTimeMillis();
        //Perform downlink speed test
        packetHandeling(true);
        //time after measurement in milliseconds
        long timeAfter1=System.currentTimeMillis();
        //close connections to server
        dos.close();
        dis.close();
        s.close();
        //Send measured speed of downlink in Mbps to onProgressUpdate
        publishProgress("downlink:" + ((bufferrec.length*8)/1000.0)/
((timeAfter1-timeBefor1)/1000.0));
        //Ping Google Server
        String ping2=ping.pingRun(false);
        if(ping2==null){
            //if ping returns null terminate measurement Procedure
            publishProgress("error");
        }else{
            //else send ping2, value pair to onProgressUpdate to be
            publishProgress("ping2:"+ping2);
        }
    } catch(Exception e){
        e.printStackTrace();
        //If there is an exception terminate measurement Procedure
        publishProgress("error");
    }

    return null;
}
@SuppressWarnings("finally")
@Override

```

```

public void onProgressUpdate(String... strings) {
    //Make Progress Bar invisible
    MainActivity.pb.setVisibility(ProgressBar.INVISIBLE);
    MainActivity.pb.invalidate();
    //If we get an error we terminate procedure and reset the GUI
    if(strings[0].equalsIgnoreCase("error")){
        MainActivity.ErrorMessageDisplay();
        MainActivity.status.setText("Terminating Test!");
        MainActivity.status.setTextColor(Color.RED);
        MainActivity.status.setBackgroundColor(Color.TRANSPARENT);
        MainActivity.status2.setBackgroundColor(Color.TRANSPARENT);
        MainActivity.status.invalidate();
        MainActivity.TestStart.setClickable(true);
        MainActivity.TestStart.setBackgroundColor(Color.LTGRAY);
        MainActivity.TestStart.setTextColor(Color.BLACK);
        MainActivity.TestStart.setText("Start Test");
        MainActivity.TestStart.invalidate();
        MainActivity.downlink.setBackgroundColor(Color.TRANSPARENT);
        MainActivity.downlink.setText("...");
        MainActivity.downlink.invalidate();
        MainActivity.uplink.setBackgroundColor(Color.TRANSPARENT);
        MainActivity.uplink.setText("...");
        MainActivity.uplink.invalidate();
        MainActivity.ping1.setBackgroundColor(Color.TRANSPARENT);
        MainActivity.ping1.setText("...");
        MainActivity.ping1.invalidate();
        MainActivity.ping2.setBackgroundColor(Color.TRANSPARENT);
        MainActivity.ping2.setText("...");
        MainActivity.ping2.invalidate();
        MainActivity.pb.setVisibility(ProgressBar.GONE);
        MainActivity.pb.invalidate();
        MainActivity.resetTesterUI();
        MainActivity.testUnderway=false;
        //Informing the VM that it is a good time to call Garbage collector
        to avoid OutOfMemoryError
        buffer=null;
        bufferrec=null;
        System.gc();
        try {
            //If we haven't terminated connection we do so here
            if(s!=null){
                if(s.isConnected()){
                    dos.close();
                    dis.close();
                    s.close();
                    //attempt to terminate thread
                    this.cancel(true);
                    return;
                }
            }
            } catch (IOException e1) {
                e1.printStackTrace();
                this.cancel(true);
                return ;
            }finally{
                this.cancel(true);
                return ;
            }
        }
    }else{
        //If we get a name,value pair
        //Tokenize the String we receive
        StringTokenizer st=new StringTokenizer(strings[0],":");
        String measurementType=st.nextToken();
        String measurementValue=st.nextToken();
        //According to the measurement taken we display it differently in
        the GUI
        if(measurementType.contains("uplink")){
            //We reduce the decimal numbers to 2
            measurementValue=((int)((double)Double.parseDouble(
            measurementValue))*100)/100.0+"";
            MainActivity.uplink.setText(measurementValue+" Kbps");
            MainActivity.uplink.invalidate();
            MainActivity.pb.setVisibility(ProgressBar.VISIBLE);
            MainActivity.pb.invalidate();
            MainActivity.uplink.setBackgroundColor(Color.TRANSPARENT);
            MainActivity.uplink.invalidate();
            MainActivity.downlink.setBackgroundColor(Color.LTGRAY);
            MainActivity.downlink.invalidate();
        }else if(measurementType.contains("downlink")){
            //We reduce the decimal numbers to 2

```

```

        measurementValue=((int)((double)Double.parseDouble(
        measurementValue))*100)/100.0+"";
        MainActivity.downlink.setText(measurementValue+" Kbps");
        MainActivity.downlink.invalidate();
        MainActivity.pb.setVisibility(ProgressBar.VISIBLE);
        MainActivity.pb.invalidate();
        MainActivity.downlink
        .setBackgroundColor(Color.TRANSPARENT);
        MainActivity.downlink.invalidate();
        MainActivity.ping2.setBackgroundColor(Color.LTGRAY);
        MainActivity.ping2.invalidate();
    }else if(measurementType.contains("ping1")){
        //We reduce the decimal numbers to 2
        measurementValue=((int)((double)Double.parseDouble(
        measurementValue))*100)/100.0+"";
        MainActivity.ping1.setText(measurementValue+" ms");
        MainActivity.ping1.invalidate();
        MainActivity.pb.setVisibility(ProgressBar.VISIBLE);
        MainActivity.pb.invalidate();
        MainActivity.ping1.setBackgroundColor(Color.TRANSPARENT);
        MainActivity.ping1.invalidate();
        MainActivity.uplink.setBackgroundColor(Color.LTGRAY);
        MainActivity.uplink.invalidate();
    } else if(measurementType.contains("ping2")){
        //We reduce the decimal numbers to 2
        measurementValue=((int)((double)Double.parseDouble(
        measurementValue))*100)/100.0+"";
        MainActivity.ping2.setText(measurementValue+" ms");
        MainActivity.ping2.invalidate();
        MainActivity.pb.setVisibility(ProgressBar.VISIBLE);
        MainActivity.pb.invalidate();
        MainActivity.ping2.setBackgroundColor(Color.TRANSPARENT);
        MainActivity.ping2.invalidate();
    }

    try {
        //Open connection to file holding measurements with append option
        fos=new FileOutputStream(output,true);
        //We write the measurement taken to the file
        fos.write((strings[0]+"\\n").getBytes());
        //close the file
        fos.close();
    } catch (Exception e) {
        e.printStackTrace();
        MainActivity.ErrorMessageDisplay();
        MainActivity.status.setText("Terminating Test!");
        MainActivity.status.setTextColor(Color.RED);
        MainActivity.status.setBackgroundColor(Color.TRANSPARENT);
        MainActivity.status2.setBackgroundColor(Color.TRANSPARENT);
        MainActivity.status.invalidate();
        MainActivity.TestStart.setClickable(true);
        MainActivity.TestStart.setBackgroundColor(Color.LTGRAY);
        MainActivity.TestStart.setTextColor(Color.BLACK);
        MainActivity.TestStart.setText("Start Test");
        MainActivity.TestStart.invalidate();
        MainActivity.testUnderway=false;
        this.cancel(true);
        return ;
    }
}

}
@SuppressWarnings("deprecation")
@Override
public void onPostExecute(Void a) {
    //Informing the VM that it is a good time to call Garbage collector to
    avoid OutOfMemoryError
    buffer=null;
    bufferrec=null;
    System.gc();
    //If an error didn't occur
    if(!((String)MainActivity.status.getText().toString())
    .equalsIgnoreCase("Terminating Test!")){
        MainActivity.status.setText("Test Completed.");
        MainActivity.status.invalidate();
        String tech;
        try {
            fos = new FileOutputStream(output,true);
            //Add the type of OS that the device has
            fos.write(("os:Android_" +Build.VERSION.SDK+ "\\n").
            getBytes());

```





## Appendix C: Κώδικας activity\_getting\_location.xml.

### Κώδικας activity\_getting\_location.xml

```
<?xml version="1.0" encoding="utf-8"?>

<TabHost android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:id="@+id/tabHost"
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    >
    <TabWidget
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:id="@android:id/tabs"
    />
    <FrameLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:id="@android:id/tabcontent"
    >

        <RelativeLayout

            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:paddingBottom="@dimen/activity_vertical_margin"
            android:paddingLeft="@dimen/activity_horizontal_margin"
            android:paddingRight="@dimen/activity_horizontal_margin"
            android:id="@+id/tab1"
            android:paddingTop="80px"

            tools:context=".MainActivity" >

                <fragment
                    android:id="@+id/map"
                    android:name="com.google.android.gms.maps.MapFragment"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:layout_below="@+id/label"
                />
                <TextView
                    android:text=""
                    android:layout_alignParentLeft="true"
                    android:id="@+id/Value1"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_alignTop="@+id/map"
                    android:visibility="invisible"/>
                <TextView
                    android:text=""
                    android:layout_alignParentLeft="true"
                    android:id="@+id/Value2"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_below="@+id/Value1"
                    android:visibility="invisible"/>
                <TextView
                    android:text=""
                    android:layout_alignParentLeft="true"
                    android:id="@+id/Value3"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_below="@+id/Value2"
                    android:visibility="invisible"/>
                <TextView
                    android:text=""
                    android:layout_alignParentLeft="true"
                    android:id="@+id/Value4"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_below="@+id/Value3"
                    android:visibility="invisible"/>
                <TextView
                    android:text=""
                    android:layout_alignParentLeft="true"
```

```

        android:id="@+id/Value5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/Value4"
        android:visibility="invisible"/>
<TextView
    android:text=""
    android:layout_alignParentLeft="true"
    android:id="@+id/Value6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/Value5"
    android:visibility="invisible"/>
<TextView
    android:text=""
    android:layout_alignParentLeft="true"
    android:id="@+id/Value7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/Value6"
    android:visibility="invisible"/>
<TextView
    android:text=""
    android:layout_alignParentLeft="true"
    android:id="@+id/Value8"

    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/Value7"
    android:visibility="invisible"/>
<RelativeLayout
    android:id="@+id/label"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" >
    <Spinner
        android:id="@+id/MapSpinner"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true">
    </Spinner>
    <Spinner
        android:id="@+id/TileSpinner"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true">
    </Spinner>
</RelativeLayout>
</RelativeLayout>
<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:id="@+id/tab2"
    android:orientation="vertical"
    android:paddingTop="80px">
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="100px"
        android:id="@+id/ConnectionType"
        android:text="@string/ConnectionType"
    />
    </LinearLayout>
<RelativeLayout
    android:id="@+id/tab3"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:paddingTop="80px">
<TableLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_alignParentBottom="true"
    android:id="@+id/svcf"
    android:stretchColumns="*"
    android:shrinkColumns="*"
    android:orientation="vertical"
    android:layout_marginBottom="100px">
    <TableRow
        android:id="@+id/tableRow1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content">

```

```

<TextView
    android:id="@+id/status"
    android:text="Test Inactive"
    android:layout_width="fill_parent"/>
<TextView
    android:id="@+id/status2"
    android:layout_width="fill_parent"/>
</TableRow>
<TableRow
    android:id="@+id/tableRow1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
    <TextView
        android:id="@+id/pserver"
        android:text="Personal Server"/>
    <TextView
        android:id="@+id/ping1"
        android:text="...">
    </TableRow>
    <TableRow
        android:id="@+id/tableRow2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
        <TextView
            android:text="Uplink"/>
        <TextView
            android:id="@+id/uplink"
            android:text="...">
        </TableRow>
        <TableRow
            android:id="@+id/tableRow3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content">
            <TextView
                android:text="Downlink"/>
            <TextView
                android:id="@+id/downlink"
                android:text="...">
            </TableRow>
            <TableRow
                android:id="@+id/tableRow4"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content">
                <TextView
                    android:id="@+id/rserver"
                    android:text="Remote Server"/>
                <TextView
                    android:id="@+id/ping2"
                    android:text="...">
                </TableRow>
        </TableLayout>
        <ProgressBar
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/ProgressBar"
            android:indeterminate="false"
            android:layout_centerInParent="true"
            style="@android:style/Widget.ProgressBar.Large.Inverse"/>
        <Button
            android:id="@+id/ipb"
            android:layout_width="fill_parent"
            android:layout_height="100px"
            android:layout_alignParentBottom="true"
            android:onClick="startTester"
            android:text="Start Test" />
    </RelativeLayout>
</FrameLayout>
</TabHost>

```