

Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ανάπτυξη Εφαρμογής Android αναγνώρισης δραστηριότητας του χρήστη, σχεδιασμένη βάσει του Material Design της Google Development of Android application recognizing user activity, designed based on Google's Material Design
Όνοματεπώνυμο Φοιτητή	Σμπυράκος Χρήστος
Πατρώνυμο	Νικόλαος
Αριθμός Μητρώου	ΜΠΣΠ/ 13101
Επιβλέπων	Αλέπης Ευθύμιος

Ημερομηνία Παράδοσης **19/11/2015**

(υπογραφή)

Αλέπης Ευθύμιος,
Επίκουρος Καθηγητής

(υπογραφή)

Βίρβου Μαρία
Καθηγήτρια

(υπογραφή)

Τσιχριντζής Γεώργιος
Καθηγητής

ΠΕΡΙΛΗΨΗ

Στόχος της παρούσας Μεταπτυχιακής Διατριβής αποτελεί η μελέτη εφαρμογών για λειτουργικό σύστημα Android που αναγνωρίζουν τη δραστηριότητα των χρηστών (περπάτημα, τρέξιμο, ποδηλασία κ.α.) χρησιμοποιώντας τεχνολογία αναγνώρισης δραστηριότητας και πληροφορίες από τους αισθητήρες της κινητής συσκευής. Αρχικά παρατίθενται συνοπτικά εισαγωγικές πληροφορίες για το λειτουργικό σύστημα Android, καθώς και τα αποτελέσματα της έρευνα για τέτοιου είδους εφαρμογές. Στην συνέχεια στα πλαίσια της εργασίας, είναι η υλοποίηση μιας παρόμοιας εφαρμογής, ο σχεδιασμός της οποίας θα γίνει χρησιμοποιώντας το material design της Google. Η εφαρμογή θα καταγράφει τα αποτελέσματα, θα παρουσιάζει στατιστικά στοιχεία και διαγράμματα και ο χρήστης θα έχει τη δυνατότητα να παρακολουθεί τα αποτελέσματα για την παρούσα ημέρα, καθώς και για τις προηγούμενες. Επίσης ο χρήστης θα μπορεί σε πραγματικό χρόνο να δει που βρίσκεται στον χάρτη, όπου θα βλέπει την δραστηριότητα που το σύστημα αναγνωρίζει για την δεδομένη στιγμή, καθώς και τη πιθανότητα αυτής. Επιπλέον θα δίνεται η δυνατότητα να εξαχθούν οι μετρήσεις σε ένα αρχείο κειμένου το οποίο θα αποθηκεύεται στην μνήμη της συσκευής. Τέλος γίνεται αναλυτική περιγραφή των δυνατοτήτων της εφαρμογής και περιγράφεται ο τρόπος υλοποίησής της.

Abstract

The objective of this Diploma Thesis is the study of applications for Android operating system, which recognize user activity (walking, jogging, cycling, etc.) using activity recognition technology and information from the sensors of the mobile device. At first are listed brief introductory information for the Android operating system, and the results of the research for such applications. Then in the context of this Thesis is the implementation of such an application, the design of which will be using the material design of Google. The application will record the results, presenting statistics and graphs and the user will be able to monitor the records for the present day, as well as the previous ones. Also the user will be able in real time to see where is his position on the map, the activity that the system recognizes the particular time and the probability of the activity. In addition there will be a functionality to export measurements to a text file which will be stored in the memory of the device. Finally there is a detailed description of the application's capabilities and how it is implemented.

Ευχαριστίες

Θα ήθελα να εκφράσω τις ευχαριστίες μου στον επιβλέποντα καθηγητή Κ. Ευθύμιο Αλέπη, για την βοήθεια του, την υποστήριξη του και την πολύ καλή συνεργασία που είχαμε καθ' όλο το χρονικό διάστημα εκπόνησης της διπλωματικής αυτής εργασίας. Οι συμβουλές, οι οδηγίες, οι παρατηρήσεις, η καθοδήγησή του, καθώς και το υλικό που μου παρείχε συνέβαλαν στο μέγιστο στην ολοκλήρωση της εργασίας αυτής.

Πίνακας Περιεχομένων

ΠΕΡΙΛΗΨΗ	3
Abstract	4
Κεφάλαιο 1 ^ο : Εισαγωγή.....	8
1.1 Πληροφορίες για το λειτουργικό σύστημα Android	8
1.2 Ιστορική αναδρομή	9
1.3 Αρχιτεκτονική Συστήματος Android	11
1.4 Material Design	13
Κεφάλαιο 2 ^ο : Μελέτη Παρεμφερών Εφαρμογών στην Ελλάδα και το Εξωτερικό.....	15
2.1 Google Fit	15
2.2 Runtastic Running & Fitness	16
2.3 Nike+ Running	18
2.4 RunKeeper.....	20
Κεφάλαιο 3 ^ο : Παρουσίαση και οδηγίες χρήσης της εφαρμογής	22
3.1 Αρχική οθόνη και διάγραμμα πίτας	22
3.2 Οθόνη χάρτη	28
3.3 Οθόνη με στατιστικά	30
3.4 Πληροφορίες για την εφαρμογή	32
Κεφάλαιο 4 ^ο : Πώς και με ποιά εργαλεία αναπτύχθηκε η εφαρμογή.....	33
4.1 Εργαλεία που χρησιμοποιήθηκαν	33
4.1.1 Java	33
4.1.2 IntelliJ IDE.....	33
4.1.3 Android SKD	33
4.1.4 Εξομοιωτής και φυσική συσκευή	34
4.1.5 Άλλα εργαλεία που χρησιμοποιήθηκαν	34
4.2 Λεπτομέρειες υλοποίησης της εφαρμογής	34
4.2.1 Λειτουργία της εφαρμογής στο παρασκήνιο	34
4.2.2 ActivityRecognitionApi.....	35
4.2.3 Αποθήκευση δεδομένων	37
4.2.4 Τοποθεσία	38
4.2.5 Διάγραμμα πίτας	40
4.2.6 Αποθήκευση δεδομένων σε αρχείο κειμένου.....	40
Κεφάλαιο 5 ^ο : Συμπεράσματα και μελλοντικές επεκτάσεις.....	41
5.1 Συμπεράσματα.....	41
5.2 Μελλοντική επέκταση	41
Κεφάλαιο 6 ^ο : Βιβλιογραφία	42

Κεφάλαιο 1^ο: Εισαγωγή

Ξεκινώντας θα θέλαμε να βοηθήσουμε τον αναγνώστη να κατανοήσει τον λόγο δημιουργίας της εφαρμογής και τις ανάγκες που θέλει να καλύψει. Σκοπός αυτής της διπλωματικής εργασίας είναι η ανάπτυξη μιας εφαρμογής για κινητές συσκευές για λειτουργικό σύστημα Android, μέσω της οποίας ο χρήστης θα μπορεί να παρακολουθεί πως κινήθηκε κατά τη διάρκεια μιας μέρας, καθώς και να συγκρίνει τα αποτελέσματα αυτά με τις προηγούμενες μέρες. Το κοινό στο οποίο απευθύνεται είναι ιδιαίτερα ευρύ καθώς μπορεί να χρησιμοποιηθεί από το γενικό σύνολο, αλλά και εξειδικευμένες κατηγορίες. Συγκεκριμένα οποιοσδήποτε απλός χρήστης θέλει να γνωρίζει πόσο περπατάει κάθε μέρα, πόσες ώρες σπαταλάει τη μέρα στο αυτοκίνητό του ή σε μέσα μαζικής συγκοινωνίας, ακόμα και πόση ώρα έχει το κινητό του τηλέφωνο στο χέρι και το χρησιμοποιεί. Επίσης η εφαρμογή αυτή μπορεί να γίνει ένα χρήσιμο εργαλείο στα χέρια αθλητών που θέλουν να γνωρίζουν πόση ώρα έτρεξαν και να μπορούν να δουν αν πέτυχαν τους στόχους που είχαν θέσει, καθώς και να έχουν τη δυνατότητα να παρακολουθούν την πορεία και την εξέλιξή τους στο πέρασμα χρόνου. Ακόμα επαγγελματίες οδηγοί που κάνουν συγκεκριμένα δρομολόγια, θα μπορούσαν να δουν ποιες μέρες καθυστέρησαν παραπάνω στον δρόμο πιθανότατα λόγω κίνησης και ενδεχομένως να βρίσκουν εναλλακτικές διαδρομές για εκείνες τις μέρες.

Σε αυτό εισαγωγικό κεφάλαιο θα δούμε κάποιες βασικές πληροφορίες για το λειτουργικό σύστημα Android, την ιστορία και την αρχιτεκτονική του, και θα αναφερθούμε στο Material Design, το νέο σύνολο σχεδιαστικών κανόνων που λάνσαρε η Google.

1.1 Πληροφορίες για το λειτουργικό σύστημα Android

Το Android είναι ένα ανοιχτό και ελεύθερο λειτουργικό σύστημα που σχεδιάστηκε αρχικά για έξυπνα κινητά τηλέφωνα και tablets με οθόνη αφής, αλλά πλέον η χρήση του έχει επεκταθεί σε έξυπνες τηλεοράσεις, έξυπνα ρολόγια και αξεσουάρ, μέχρι και έξυπνα αυτοκίνητα. Χρησιμοποιεί πυρήνα (kernel) του λειτουργικού Linux και η διεπαφή χρήστη βασίζεται στην άμεση χειραγώγηση των αντικειμένων και γραφικών επί της οθόνης, χρησιμοποιώντας επαφές αφής που ελάχιστα αντιστοιχούν στις ενέργειες του πραγματικού κόσμου, όπως το σύρσιμο, μαζί με ένα εικονικό πληκτρολόγιο για εισαγωγή κειμένου.

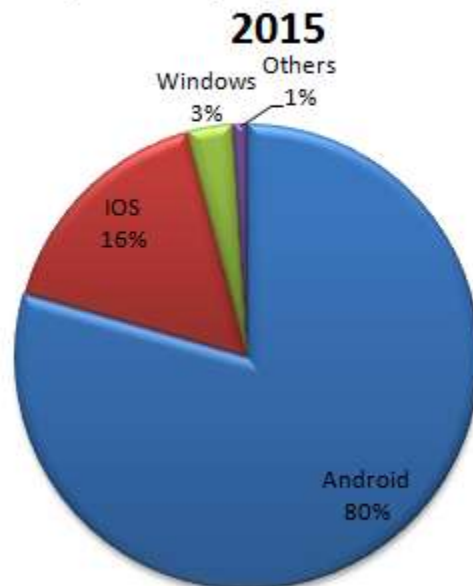
Από τον Ιούλιο του 2013, το Google Play store είχε διαθέσιμες πάνω από ένα εκατομμύριο εφαρμογές Android, και πάνω από 50 δισεκατομμύρια λήψεις εφαρμογών. Μια έρευνα του Απριλίου-Μαΐου 2013 για προγραμματιστές εφαρμογών κινητών συσκευών διαπίστωσε ότι το 71% των developers δημιουργούν εφαρμογές για το Android και μια έρευνα του 2015 έδειξε ότι το 40% επαγγελματιών προγραμματιστών χρησιμοποιούν το Android ως κύρια πλατφόρμα ανάπτυξης λογισμικού, ποσοστό το οποίο είναι συγκρίσιμο με το iOS της Apple που είναι 37%. Κατά το Google I / O 2014, η εταιρεία αποκάλυψε ότι υπήρχαν πάνω από ένα δισεκατομμύριο ενεργοί μηνιαίοι χρήστες του Android, από 538 εκατομμύρια τον Ιούνιο του 2013.

Ο πηγαίος κώδικας του Android δημοσιεύεται από την Google με άδεια χρήσης ανοιχτού κώδικα, αν τελικά οι περισσότερες συσκευές Android κυκλοφορούν με ένα συνδυασμό του ανοιχτού κώδικα και κλειστού λογισμικού, συμπεριλαμβανομένων του ιδιόκτητου λογισμικού που απαιτείται για την πρόσβαση στις υπηρεσίες της Google. Το Android είναι ιδιαίτερα δημοφιλές στις εταιρείες τεχνολογίας που απαιτούν ένα έτοιμο, χαμηλού κόστους και προσαρμόσιμο λειτουργικό σύστημα για συσκευές υψηλής τεχνολογίας, καθώς επίσης ο ανοιχτός κώδικας έχει ενθαρρύνει μια μεγάλη κοινότητα προγραμματιστών να αναπτύσσουν ιδιωτικές και μη εφαρμογές. Αυτές οι εφαρμογές αναπτύσσονται κυρίως στη γλώσσα προγραμματισμού JAVA χρησιμοποιώντας ένα πακέτο ανάπτυξης λογισμικού

Android Software Development Kit (κοινώς γνωστό ως Android SDK). Το πακέτο αυτό χρησιμοποιεί ένα εκτεταμένο σετ από εργαλεία ανάπτυξης, περιέχοντας έναν αποσφαλματωτή (debugger), βιβλιοθήκες λογισμικού, προσομοιωτές συσκευών (emulators), βοηθητικό υλικό, παραδείγματα κώδικα καθώς και βοηθητικούς οδηγούς. Το βασικότερο ολοκληρωμένο περιβάλλον ανάπτυξης (integrated development environment IDE) είναι πλέον το Android Studio της Google, του οποίου η ανάπτυξη στηρίχθηκε στο IntelliJ IDE της JetBrains που είναι το βασικό εργαλείο που χρησιμοποιήθηκε στα πλαίσια ανάπτυξης της εφαρμογής της διπλωματικής εργασίας. Υπάρχουν και άλλα εργαλεία ανάπτυξης (Cross Platform Tools) όπως το Xamarin, όπου η γλώσσα προγραμματισμού είναι η C, C++ ή C#, τα οποία αναπτύσσουν εφαρμογές που παράλληλα με το Android υποστηρίζουν iOS και άλλα λογισμικά.

Παρόλα αυτά έρευνα το 2015 κατέληξε στο συμπέρασμα ότι σχεδόν το 90% των Android συσκευών, έχουν γνωστές αλλά μη διευθετημένες ευπάθειες ασφαλείας λόγω της έλλειψης ενημερώσεων και υποστήριξης.

Global Operating System Market Share



Εικόνα 1.1: Μερίδιο αγοράς λειτουργικών για κινητές συσκευές

Πηγή: www.forbes.com

1.2 Ιστορική αναδρομή

Η εταιρία Android Inc. ιδρύθηκε στη California των Η.Π.Α τον Οκτώβριο του 2003 από τους Andy Rubin, Rich Miner, Nick Sears και Chris White με σκοπό την ανάπτυξη φορητών εφαρμογών που να ενσωματώνουν πληροφορίες σχετικά με τη θέση και τις προτιμήσεις του εκάστοτε χρήστη. Οι αρχικές προθέσεις της εταιρίας ήταν να δημιουργήσει ένα προηγμένο λειτουργικό σύστημα για ψηφιακές

κάμερες αλλά όταν έγινε αντιληπτό ότι το μερίδιο της αγοράς στη συγκεκριμένη κατηγορία ήταν πολύ μικρό αποφασίστηκε να επικεντρωθούν οι προσπάθειες στην ανάπτυξη ενός λειτουργικού για φορητές συσκευές ικανό να ανταγωνιστεί τα εκείνης της εποχής πρωτοπόρα φορητά λειτουργικά συστήματα Symbian και Windows Mobile.

Τον Ιούλιο του 2005 η Google απέκτησε την Android Inc. για περίπου 50 εκατομμύρια δολάρια. Εκείνη την εποχή δεν ήτανε πολλά πράγματα γνωστά για την εταιρία αλλά πολλοί υπέθεσαν ότι με αυτήν την κίνηση η Google επιχειρούσε να εισέλθει στην αγορά των κινητών τηλεφώνων. Είδη από το Δεκέμβριο του 2006 αυξάνονταν οι φήμες για την είσοδο της Google στην αγορά της κινητής τηλεφωνίας κάτι που επιβεβαιώθηκε και επίσημα στις 5 Νοεμβρίου 2007 όταν και ανακοινώθηκε για πρώτη φορά η δημιουργία του Android ως ένα λειτουργικό σύστημα για φορητές συσκευές βασισμένο στο Linux. Το πρώτο κινητό που χρησιμοποιούσε το λογισμικό Android (η συσκευή HTC Dream, της ομώνυμης εταιρείας) διατέθηκε στην αγορά τον Οκτώβρη του 2008. Το 2010 η Google κυκλοφόρησε τη σειρά Nexus, που είναι μια σειρά έξυπνων τηλεφώνων και tablet που τρέχουν το λειτουργικό Android. Το πρώτο smartphone της σειράς, το Nexus One, δημιουργήθηκε σε συνεργασία της Google με την HTC και από τότε η σειρά έχει επεκταθεί με πολλά νέα έξυπνα τηλέφωνα και tablet όπως το Nexus 5 ή το tablet Nexus 7. Στην συνέχεια ακολούθησαν και άλλες μεγάλες και μικρές εταιρείες που εμπιστεύθηκαν το Android ως λειτουργικό και κατάφεραν να το κάνουν το κυρίαρχο λειτουργικό αυτή τη στιγμή στην αγορά. Η συνεργασία με την Samsung, η οποία είναι η πρώτη εταιρεία στις πωλήσεις κινητών παγκοσμίως αυτή την στιγμή κατάφερε και εδραίωσε το Android και η εξάπλωση του οφείλεται κατά κύριο λόγο σε αυτήν.

Από το 2008 έως σήμερα έχουν γίνει αρκετές αναβαθμίσεις στο λειτουργικό σύστημα, οι οποίες έχουν επιφέρει σημαντικές βελτιώσεις προσθέτοντας πολλά νέα χαρακτηριστικά ή διορθώνοντας αρκετά προβλήματα των προηγούμενων εκδόσεων. Κάθε αναβαθμισμένη έκδοση του λειτουργικού παίρνει το όνομά της από κάποιο επιδόρπιο, γλυκό, σνακ με αλφαβητική σειρά. Για παράδειγμα η έκδοση 1.5 με το όνομα «Cupcake» ακολουθήθηκε από την 1.6 με το όνομα «Donut» ή η έκδοση 4.4.4 «kitkat» από την 5.0 με το όνομα «Lollipop» στις 14 Νοεμβρίου 2014. Η έκδοση Lollipop εισήγαγε για πρώτη φορά το material design, ένα σύνολο σχεδιαστικών κανόνων, που προβλέπεται να φέρει επανάσταση στο γραφικό περιβάλλον διεπαφής του χρήστη και θα μας αποσχολήσει σε επόμενη ενότητα της διπλωματικής εργασίας. Το 2014 η Google κυκλοφόρησε το Android One, ένα smartphone που απευθύνεται κυρίως σε πελάτες στις χώρες του αναπτυσσόμενου κόσμου. Η πιο πρόσφατη αναβάθμιση του Android είναι η 6.0 Marshmallow, η οποία αποκαλύφθηκε για πρώτη φορά τον Μάιο του 2015 στο Google I/O υπό την κωδική ονομασία «Android M» και επισήμως τον Οκτώβρη του ίδιου έτους. Το Marshmallow επικεντρώνεται κυρίως στη βελτίωση της συνολικής εμπειρίας του χρήστη του Lollipop, εισάγει νέα δικαιώματα αρχιτεκτονικής, νέα APIs, ένα νέο σύστημα διαχείρισης ενέργειας που μειώνει την δραστηριότητα στο παρασκήνιο όταν μια συσκευή βρίσκεται σε αδράνεια, εγγενή υποστήριξη για την αναγνώριση δακτυλικών αποτυπωμάτων και υποδοχές USB Type-C, τη δυνατότητα αποστολής και αποθήκευσης των δεδομένων σε μια κάρτα microSD και χρησιμοποίηση αυτής ως πρωτεύοντα αποθηκευτικό χώρο, καθώς και άλλες εσωτερικές αλλαγές.



Εικόνα 1.2: Εκδόσεις Android

Πηγή: <http://developer.android.com/>

1.3 Αρχιτεκτονική Συστήματος Android

Το λειτουργικό σύστημα Android μπορεί να χωριστεί σε πέντε επίπεδα: Linux Kernel, μητρικές βιβλιοθήκες (native libraries), Android Runtime, application framework και applications.

Ο Linux Kernel είναι ο πυρήνας στον οποίο βασίζεται το Android και βρίσκεται στο χαμηλότερο επίπεδο, ενώ είναι τροποποιημένος λόγω των ιδιαίτερων αναγκών στη διαχείριση ενέργειας, μνήμης, διεργασιών, οδηγών της συσκευής, δικτύωσης και του περιβάλλοντος λειτουργίας. Ακριβώς «πάνω» από το kernel τρέχουν κάποια τυπικά πακέτα Linux (daemons) όπως το bluez για το Bluetooth και το wpa_supplicant για την κρυπτογράφηση του ασύρματου δικτύου Wi-Fi.

Οι βιβλιοθήκες είναι γραμμένες σε C και C++ και η πρόσβαση σε αυτές γίνεται μέσω κατάλληλης διεπαφής που προσφέρει η Java. Λόγω του ότι το Android προορίζεται για συσκευές με μικρή κύρια μνήμη και χαμηλής ενέργειας επεξεργαστές (CPUs), οι βιβλιοθήκες για εντατικές εργασίες του CPU και του GPU (Graphic Processor Unit) μεταφράζονται σε τροποποιημένες μητρικές βιβλιοθήκες της συσκευής. Βασικές βιβλιοθήκες όπως η libc και η lirm αναπτύχθηκαν ειδικά για χαμηλή κατανάλωση ενέργειας και με βάση το καθεστώς αδειοδότησης του Android. Σε αυτό το επίπεδο ο διαχειριστής επιφάνειας καθορίζει την πρόσβαση στην οθόνη για τον διαχειριστή παραθύρου. Αντίθετα από τα άλλα frameworks, το framework πολυμέσων που περιέχει ήχο και βίντεο πρέπει να τροποποιηθεί εξ ολοκλήρου.

Το Android Runtime αποτελείται από την εικονική συσκευή Dalvik (Dalvik Virtual Machine, DVM) και βασικές βιβλιοθήκες πυρήνα Java. Η εικονική συσκευή Dalvik είναι μεταγλωττιστής για κώδικα byte ο οποίος έχει μετατραπεί από κώδικα byte Java σε κώδικα byte Dalvik. Οι εφαρμογές Android είναι γραμμένες στην υψηλού επιπέδου γλώσσα Java. Η DVM είναι υπεύθυνη για την μεταγλώττιση και την

δημιουργία εκτελέσιμων αρχείων τύπου .dex τα οποία αναγνωρίζονται και εκτελούνται από το λειτουργικό σύστημα. Η εκτέλεσή τους γίνεται παράλληλα και από διαφορετικές εικονικές μηχανές, έτσι ώστε αν παρουσιαστεί ένα σφάλμα σε κάποιο να μην επηρεαστούν τα υπόλοιπα.

Το application framework είναι γραμμένο σε Java και παρέχει αφαιρέσεις από μητρικές βιβλιοθήκες και δυνατότητες του Dalvik στις εφαρμογές. Οι εφαρμογές Android τρέχουν τη δική τους εικονική συσκευή Dalvik και μπορούν να περιέχουν διάφορα στοιχεία όπως Activities, services, broadcast receivers. Τα στοιχεία αυτά αλληλεπιδρούν μεταξύ τους και μπορούν να παρέχουν μια σειρά από υπηρεσίες και λειτουργίες προς άλλες εφαρμογές εφόσον τηρούνται οι πολιτικές ασφαλείας του framework. Μερικές από τις πιο βασικές οντότητες που περιλαμβάνει το application framework είναι το Activity manager, Resource manager, Notification manager, View System, Content Providers.

Είναι το πιο υψηλότερο επίπεδο, είναι εφαρμογές που έρχονται μαζί με την Android συσκευή του χρήστη (όπως τηλέφωνο, επαφές, μουσική, e-mail κ.α.), όπως επίσης εφαρμογές που κάνει εγκατάσταση σύμφωνα με τις ανάγκες του. Οποιαδήποτε εφαρμογή που εγκαθιστά ο χρήστης εκτελείται σε αυτό το επίπεδο. Η χρήση τους γίνεται με διαφανή τρόπο χωρίς να γνωρίζει ο χρήστης τι συμβαίνει από πίσω ή τι απαιτήσεις έχουν από το λειτουργικό σύστημα.

Στην παρακάτω εικόνα απεικονίζεται η αρχιτεκτονική του android και γίνονται ευδιάκριτα τα επίπεδα.



Εικόνα 1.3: Αρχιτεκτονική δομή του Android

Πηγή: www.wikipedia.com

1.4 Material Design

Το Material design είναι ένα σύνολο σχεδιαστικών κανόνων (design language) που δημιουργήθηκε από τη Google και ανακοινώθηκε για πρώτη φορά στις 25 Ιουνίου, 2014 στο συνέδριο Google I/O. Αποτελεί επέκταση του μοντέλου των "καρτών" που είναι σχεδιασμός με βασικό συστατικό στοιχείο τη διάταξη πίνακα (grid layout). Κύριο χαρακτηριστικό του Material design είναι ανταπόκριση των γραφικών στις κινήσεις των χρηστών με τη χρήση διάφορων εφέ, όπως αιώρηση αντικειμένων, σκιές, μεταβαλλόμενο φωτισμό, μεταβάσεις γραφικών, αίσθηση βάθους στην οθόνη, padding, έντονα χρώματα, με σκοπό να υπάρχει μια λογική και φυσική αλληλουχία γεγονότων σε τρεις διαστάσεις.

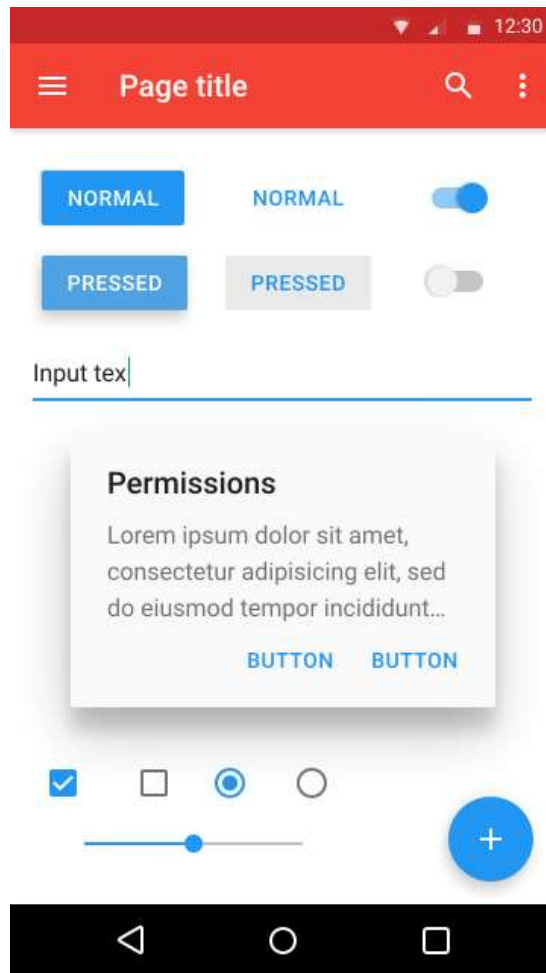
Το Material Design πρωτοεμφανίστηκε με την έκδοση Android 5.0.1 (Lollipop), αλλά πλέον είναι συμβατό να χρησιμοποιηθεί από την έκδοση 2.1 και πάνω, μέσω της βιβλιοθήκης v7 appcompat, που χρησιμοποιείται σε όλες σχεδόν τις συσκευές Android που φτιάχτηκαν από το 2009 και μετά. Το Material Design σταδιακά θα επεκταθεί σε όλα τα προϊόντα της Google, όπως επίσης και σε προϊόντα ανταγωνιστικών και μη εταιριών, τόσο στο web όσο και στο mobile, προσφέροντας ομαλή εμπειρία σε όλες τις πλατφόρμες και εφαρμογές της. Ήδη, το Material Design έχει ενσωματωθεί στις περισσότερες εφαρμογές κινητών τηλεφώνων της Google στο Android όπως τα Gmail, YouTube, Google Drive και πλέον αποτελεί την νούμερο ένα επιλογή για νέες εφαρμογές, καθώς και πάρα πολλές ήδη επιτυχημένες εφαρμογές άλλαξαν το UI τους σε material design, όπως το Skype. Η Google έχει κυκλοφορήσει επίσης APIs (application programming interfaces) για τους προγραμματιστές που θέλουν να ενσωματώσουν την σχεδιαστική αυτή γλώσσα στις εφαρμογές τους, καθώς και την σχετική βιβλιοθήκη Android Design Support Library.

Η απάντηση της Google ως προς το ποιά ήταν η έμπνευσή της για τη δημιουργία του material design, είναι χαρτί και το μολύβι. Χαρακτηριστικά ένας ο σχεδιαστής Matías Duarte εξήγησε ότι «σε αντίθεση με το πραγματικό χαρτί, το ψηφιακό υλικό μπορεί να επεκταθεί και να μορφοποιηθεί έξυπνα. Το material έχει φυσικές επιφάνειες και άκρα. Οι σκιές δίνουν νόημα σε αυτό που αγγίζεις.

Ο σχεδιασμός της εφαρμογής έγινε βάσει των κατευθυντήριων γραμμών του material design που προτείνει η Google στην επίσημη σελίδα για android developers. Τα features που χρησιμοποιήθηκαν παραθέτονται παρακάτω:

- Navigation Drawer
- Navigation View
- Material Toolbar
- Recycler View
- Card View
- Floating Action Buttons
 - a) Single Floating Action Button
 - b) Floating Action Button Toolbar
 - c) Floating Action Button Sheet
- Activity Transition
- Material Icons
- Material Dialogs

- Material Colors (Primary/Primary Dark είναι Indigo και Color Accent Light Blue)



Εικόνα 1.4: Μερικά τυπικά συστατικά του Material Design
 Πηγή: www.wikipedia.com

Κεφάλαιο 2^ο: Μελέτη Παρεμφερών Εφαρμογών στην Ελλάδα και το Εξωτερικό

Σε αυτό το κεφάλαιο θα παρουσιαστούν μερικές από τις πιο επιτυχημένες εφαρμογές για κινητές συσκευές, που η κύρια λειτουργία τους βασίζεται στην αναγνώριση δραστηριότητας του χρήστη. Οι εφαρμογές συνδυάζουν την αναγνώριση δραστηριότητας με διάφορες άλλες πληροφορίες που παρέχονται από τους αισθητήρες της συσκευής (όπως Gps, επιταχυνσιόμετρο) και προσφέρουν εντυπωσιακές μετρήσεις και στατιστικά στοιχεία. Οι εφαρμογές αυτές απευθύνονται κυρίως σε αθλητές, αλλά χρησιμοποιούνται ευρέως και από απλούς χρήστες που ασχολούνται με κάποια αθλητική δραστηριότητα. Μάλιστα μερικές από αυτές δίνουν στον χρήστη της αίσθηση ότι έχει έναν προσωπικό γυμναστή στην τσέπη του. Στο πεδίο αυτό υπάρχουν πάρα πολλές επιλογές που διατίθενται είτε δωρεάν, είτε σε πολύ προσιτές τιμές.

2.1 Google Fit

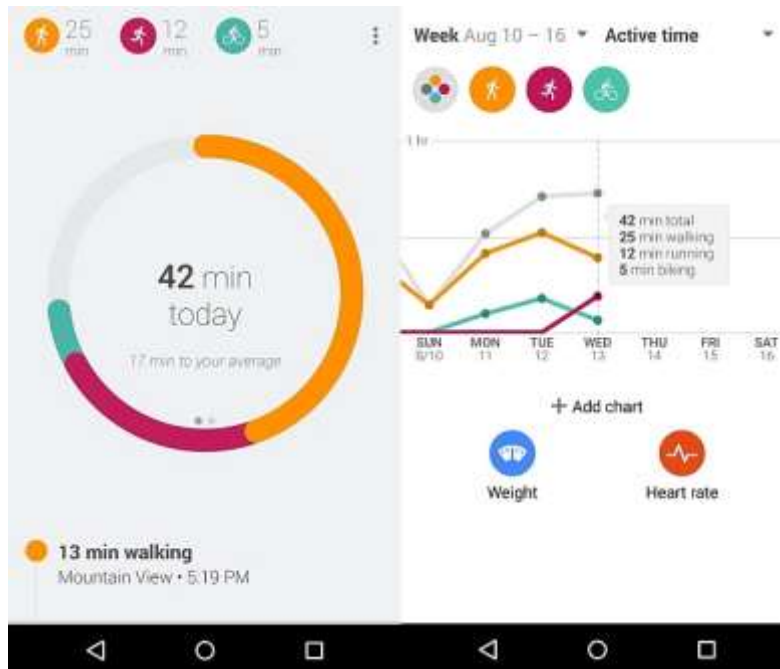
Η Google στις 25 Ιουνίου 2014, λίγο μετά την έκδοση της Apple, iOS 8 HealthKit, ανακοίνωσε το Google fit το οποίο μπορεί να θεωρηθεί ως άμεσος ανταγωνιστής. Το Google Fit είναι μια πλατφόρμα για την παρακολούθηση της υγείας που αναπτύχθηκε από την Google για το λειτουργικό σύστημα Android. Είναι ένα ενιαίο σύνολο από APIs που συνδυάζει δεδομένα από πολλές εφαρμογές και συσκευές. Η Google Fit χρησιμοποιεί τους αισθητήρες και την αναγνώριση δραστηριότητας της κινητής συσκευής ενός χρήστη για την καταγραφή των δραστηριοτήτων φυσικής αγωγής (όπως το περπάτημα, το τρέξιμο ή το ποδήλατο), τα οποία συγκρίνει με τους στόχους του χρήστη για να παρέχει μια ολοκληρωμένη άποψη της φυσικής του κατάστασης. Οι χρήστες έχουν τη δυνατότητα να επιλέγουν με ποιόν θα μοιράζονται τα δεδομένα και επίσης να διαγράφουν αυτές τις πληροφορίες ανά πάσα στιγμή.



Εικόνα: 2.1: Google Fit logo

Πηγή: play.google.com/store

Για την ανάπτυξη της εφαρμογής η Google συνεργάστηκε με κορυφαίες εταιρίες που παρέχουν τέτοιου τύπου εφαρμογές. Σε αυτές περιλαμβάνονται η Nike, η HTC, η LG, η Withings, η Motorola, η Noom, η Runtastic, η RunKeeper και η Polar. Στις εφαρμογές τριών από των παραπάνω εταιριών θα αναφερθούμε στην συνέχεια του κεφαλαίου.



Εικόνα 2.2: Screenshots απο το Google fit

Πηγή: play.google.com/store

2.2 Runtastic Running & Fitness

Η Runtastic είναι μια εφαρμογή αθλητικών δραστηριοτήτων που κατέχει πολύ υψηλή βαθμολογία από τους χρήστες Android στο Google Playstore (4,5) και έχει 10 με 50 εκατομμύρια λήψεις, καθώς είναι ιδιαίτερα δημοφιλής και στις άλλες δυο πλατφόρμες που υποστηρίζεται (Apple iOS και Windows Phones).

Δημιουργήθηκε από την Αυστριακή εταιρία Runtastic που συνδυάζει την παραδοσιακή γυμναστική με εφαρμογές για κινητές συσκευές που εμπεριέχουν στοιχεία από μέσα κοινωνικής δικτύωσης και παιχνίδια. Η αρχική ιδέα γεννήθηκε κατά τη διάρκεια ενός project στο Πανεπιστήμιο Εφαρμοσμένων Επιστημών της Άνω Αυστρίας για την παρακολούθηση αγώνων ιστιοφόρων. Σύντομα οι ιδρυτές κατάλαβαν ότι το κοινό στο οποίο απευθύνονταν ήταν πολύ μικρό και έτσι αποφάσισαν να επικεντρωθούν σε πιο δημοφιλή αθλήματα, όπως τρέξιμο, ποδηλασία, ή περπάτημα.

Η Runtastic αναπτύσσει εφαρμογές αναγνώρισης δραστηριότητας, προϊόντα υλικού και υπηρεσιών, όπως: on-line logs προπόνησης, λεπτομερή ανάλυση των δεδομένων, συγκρίσεις με άλλους χρήστες και πολλές περισσότερες λειτουργίες για να βοηθήσει τους χρήστες να βελτιώσουν τη συνολική φυσική

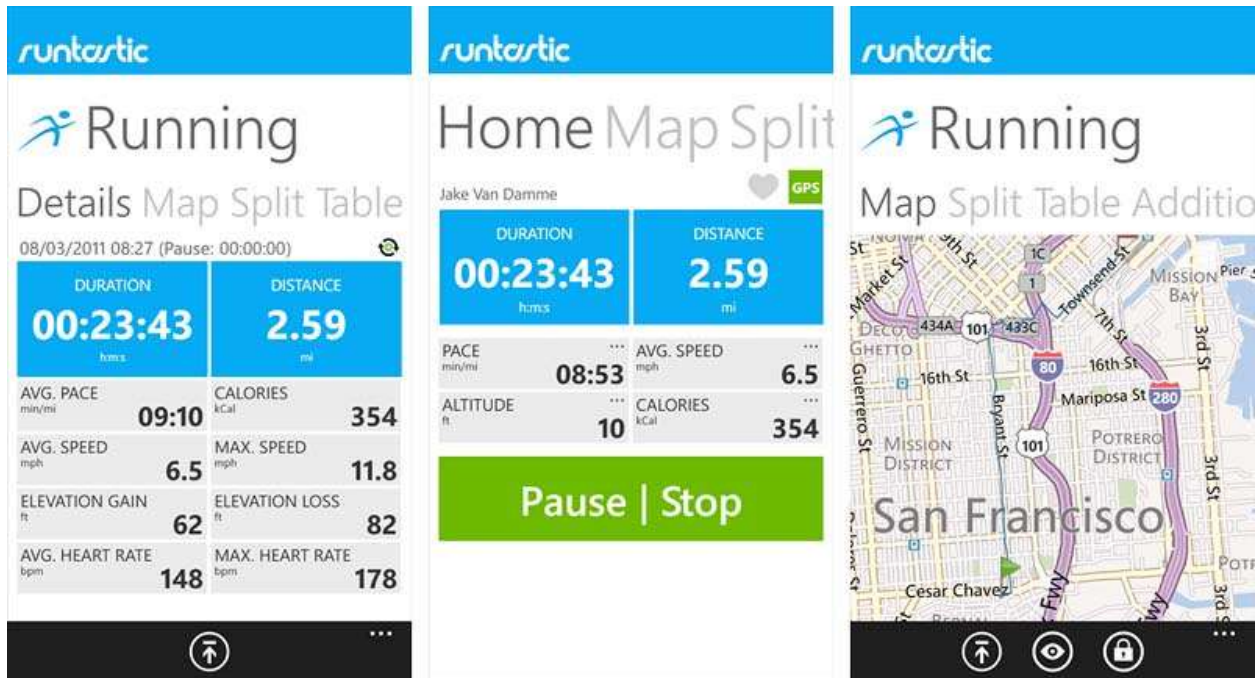
τους κατάσταση. Η εταιρία εξαγοράστηκε τον Αύγουστο του 2015 από την Adidas έναντι του ποσού των 220 εκατομμυρίων ευρώ.



Εικόνα 2.3: Runtastic Running & Fitness App Icon

Πηγή: play.google.com/store

Η εφαρμογή της Runtastic χρησιμοποιεί το GPS για να χαρτογραφήσει και να παρακολουθεί τις αθλητικές δραστηριότητες και την φυσική κατάσταση, όπως τρέξιμο, τζόκινγκ, ποδηλασία ή περπάτημα. Η Runtastic app παρακολουθεί την πρόοδο του χρήστη στην προπόνηση συλλέγοντας πληροφορίες ως προς τον χρόνο, την απόσταση, την υψομετρική διαφορά, την ταχύτητα, τις θερμίδες που έκαψε ο χρήστης, τους καρδιακούς παλμούς, την διαδρομή που ακολούθησε ο χρήστης και τη θέση του στον χάρτη, και ακόμα περισσότερα με στόχο να βοηθήσει τον χρήστη να επιτύχει τους στόχους που έθεσε για την προπόνηση. Επίσης περιλαμβάνει πολλά στατιστικά γραφημάτων και διαγράμματα, προσαρμόσιμο ταμπλό (dashboard) έτσι ώστε ο χρήστης να βλέπει μόνο τις πληροφορίες που θέλει και έχει την δυνατότητα ενσωμάτωσης με πολύ μεγάλες μουσικές εφαρμογές, ώστε να μπορεί ο χρήστης να ακούει μουσική ενώ προπονείται έξω. Ακόμα η εφαρμογή υποστηρίζεται και για android wear (έξυπνα ρολόγια κ.α.), όπως επίσης υπάρχει και μια επέκταση για τα έξυπνα ρολόγια της Sony. Επιπροσθέτως διατίθεται και η επί πληρωμή έκδοση της εφαρμογής (Runtastic Running Pro) έναντι 4,99 ευρώ, η οποία δίνει ακόμα περισσότερες δυνατότητες στον χρήστη.



Εικόνα 2.4: Runtastic screenshots

Πηγή: play.google.com/store

2.3 Nike+ Running

Η εφαρμογή Nike+ Running αναπτύχθηκε από την εταιρία αθλητικών ειδών Nike, inc., η οποία μετρά και καταγράφει την απόσταση και τον ρυθμό που τρέχει ή περπατάει ο χρήστης. Πρόγονος της εφαρμογής ήταν το Nike+ iPod Sports Kit που αποτελείται από μια μικρή συσκευή αποστολής σημάτων που συνδέεται ή είναι ενσωματωμένη σε ένα παπούτσι, το οποίο επικοινωνεί είτε με το Nike + SportBand, έναν δέκτη συνδεδεμένο σε ένα iPod Nano, ή απευθείας με ένα iPod δεύτερης γενιάς, iPhone 3GS και πάνω, ή ένα Nike + SportWatch.

Στις 7 Σεπτεμβρίου 2010, η Nike κυκλοφόρησε το Nike + Running App (αρχικά ονομαζόταν Nike + GPS) για το App Store, το οποίο χρησιμοποίησε μια μηχανή παρακολούθησης που τροφοδοτείται από την εταιρία MotionX που δεν απαιτεί ξεχωριστό αισθητήρα στο παπούτσι ή βηματόμετρο. Η εφαρμογή αυτή λειτουργεί χρησιμοποιώντας το επιταχυνσιόμετρο και το GPS του iPhone και το επιταχυνσιόμετρο του iPod Touch, το οποίο δεν έχει τσιπ GPS. Το Nike + Running είναι συμβατό με το iPhone 6 και το iPhone 6 Plus, καθώς και όλα τα προηγούμενα μοντέλα έως το iPhone 3GS και iPod touch. Στις 21 Ιουνίου του 2012 η Nike κυκλοφόρησε το Nike + Running App και για το Android. Η τρέχουσα εφαρμογή είναι συμβατή με όλα τα τηλέφωνα Android 4.0.3 και πάνω. Αυτή τη στιγμή η εφαρμογή αριθμεί 10 με 50 εκατομμύρια λήψεις στο Google Play Store, με βαθμολογία 4,4 αστέρια στα 5, ενώ αντίστοιχα είναι και τα νούμερα στην πλατφόρμα iOS.



Εικόνα 2.5: Nike+ Running icon

Πηγή: play.google.com/store

Η εφαρμογή καταγράφει με ακρίβεια το ρυθμό σας, τη διανυόμενη απόσταση και το χρόνο, δίνοντας ηχητικές πληροφορίες και χάρτη της διαδρομής, είτε βρίσκεστε σε διάδρομο, είτε σε υπαίθριο χώρο ή γήπεδο. Προσφέρει μεγάλη ποικιλία στατιστικών στοιχείων, μέχρι και την αυξομείωση της απόδοσής σας ανάλογα με τις διάφορες καιρικές συνθήκες. Ακόμα διαθέτει μια λειτουργία προπονητή που προσφέρει προγράμματα εξάσκησης και καθημερινές προπονήσεις για να σας βοηθήσει να προπονήστε πιο έξυπνα.

Η εφαρμογή διαθέτει επίσης με μια σειρά από κοινωνικά στοιχεία όπως η κοινή χρήση διαδρομής, επιτευγμάτων και τη δυνατότητα να προκαλέσετε τους φίλους σας να ξεπεράσουν τον χρόνο σας ή την απόσταση που έχετε διανύσει. Επιπροσθέτως μπορεί να επικοινωνεί με εφαρμογές παρακολούθησης θερμίδων, όπως το Lose It! για να σας βοηθήσει να παρακολουθείτε το ιστορικό καύσης θερμίδων. Ένα επιπλέον χαρακτηριστικό γνώρισμα της εφαρμογής είναι η δυνατότητα να φορτώσετε ένα «δυνατό» τραγούδι της επιλογής σας, για να σας εμπνεύσει όποτε έχουν μείνει μόλις λίγα μέτρα για να πετύχετε τον στόχο σας.



Εικόνα 2.6 Screenshots από Nike+ Running

Πηγή: play.google.com/store

2.4 RunKeeper

Το RunKeeper είναι μια εφαρμογή που χρησιμοποιεί τον εντοπισμού μέσω GPS, που διατίθεται για iOS και Android και έχει πάνω από 40 εκατομμύρια χρήστες. Λανσαρίστηκε για πρώτη φορά το 2008 από τον CEO Jason Jacobs. Στα τέλη του 2011 το RunKeeper εξασφάλισε 10 εκατομμύρια δολάρια σε έναν διαγωνισμό Β σειράς χρηματοδότηση, υπό την αιγίδα της Spark Capital.



Εικόνα 2.7: RunKeeper icon

Πηγή: play.google.com/store

Οι χρήστες με το RunKeeper μπορούν να παρακολουθούν τις αθλητικές τους δραστηριότητες (όπως περπάτημα, τρέξιμο και ποδηλασία) με τη χρήση GPS και για να βλέπουν αναλυτικά στατιστικά στοιχεία που αφορούν το ρυθμό τους, την απόσταση και το χρόνο. Το RunKeeper παρακολουθεί τις επιδόσεις με την πάροδο του χρόνου, επιτρέποντας στους χρήστες να προβάλουν ένα λεπτομερές ιστορικό των δραστηριοτήτων τους, παραθέτει στατιστικά στοιχεία, την πρόοδο του χρήστη και διαθέτει λειτουργία προπόνησης. Η λειτουργία προπόνησης προτρέπει τον χρήστη να αυξήσει στην πρόοδό του ενσωματωμένα ηχητικά μηνύματα στην εφαρμογή. Επίσης μετράει τους καρδιακούς παλμούς και ειδοποιεί τους χρήστες όταν οι παλμοί έχουν ανέβει επικίνδυνα, όπως επίσης τους ειδοποιεί όταν επιτευχθεί προσωπικό ρεκόρ ή κάποιος στόχος.

Η εφαρμογή αυτή επιτρέπει στους χρήστες να ακούν μουσική και να τραβούν φωτογραφίες κατά τη διάρκεια της αθλητικής τους δραστηριότητας, να μοιράζονται το προσωπικό τους περιεχόμενο με τους φίλους τους μέσω δημοσιοποίησης φωτογραφιών, δραστηριοτήτων και επιτευγμάτων στο Facebook και στο Twitter και επιτρέπει ακόμα και την ζωντανή παρακολούθηση του χάρτη των προπονήσεων και αγώνων, καθώς οι χρήστες τρέχουν.



Εικόνα 2.8 Screenshots από RunKeeper

Πηγή: play.google.com/store

Κεφάλαιο 3^ο: Παρουσίαση και οδηγίες χρήσης της εφαρμογής

3.1 Αρχική οθόνη και διάγραμμα πίτας

Αφού αρχικά έχουμε εγκαταστήσει την εφαρμογή, με την πρώτη είσοδό μας βλέπουμε την αρχική οθόνη όπου βλέπουμε την ημερομηνία της παρούσας ημέρας, καθώς και τις πιθανές δραστηριότητες που μπορεί η εφαρμογή να αναγνωρίσει. Από αυτή τη στιγμή και μετά η εφαρμογή καταγράφει είτε είναι στο προσκήνιο είτε στο παρασκήνιο την δραστηριότητα του χρήστη κάθε λεπτό. Οι καταγραφές ανανεώνονται σε πραγματικό χρόνο όταν ο χρήστης χρησιμοποιεί την εφαρμογή και για κάθε δραστηριότητα παρουσιάζονται τα λεπτά, οι ώρες και το ποσοστό που αντιστοιχεί στην συγκεκριμένη δραστηριότητα έναντι του συνολικού χρόνου καταγραφής.



Εικόνα 3.1 Screenshot από την αρχική οθόνη για την παρούσα μέρα

Στο κάτω δεξιά μέρος της οθόνης παρατηρούμε ότι υπάρχει ένα floating action button. Όταν το επιλέξουμε μετατρέπεται σε materia sheet και μας παρέχει τρεις επιλογές. Οι επιλογές αυτές είναι Export Previous Days Results, Reset και Pie Chart και θα τις περιγράψουμε στην συνέχεια.



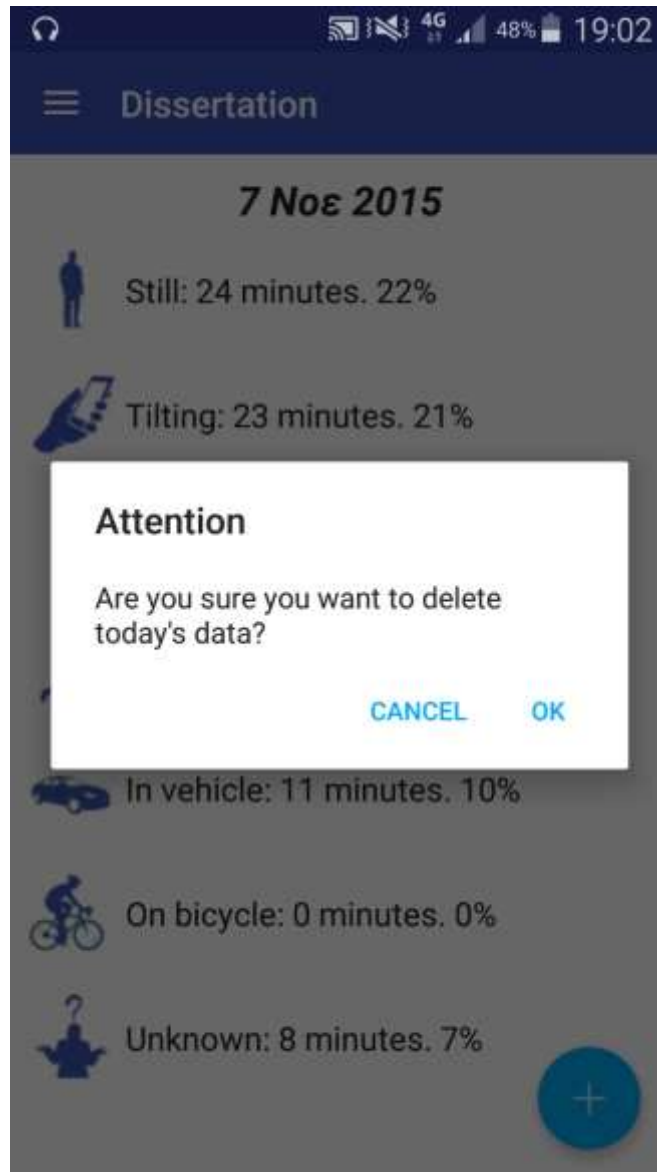
Εικόνα 3.2: Screenshot από την αρχική οθόνη με τις επιλογές του material sheet

1. Export Previous Days Results: με αυτήν την επιλογή μπορούμε να αποθηκεύσουμε τις μετρήσεις των προηγούμενων ημερών στην μνήμη της κινητής μας συσκευής με την μορφή αρχείου .txt. Συγκεκριμένα δημιουργείται το αρχείο «PreviousDaysResults.txt» και αποθηκεύεται στην τοποθεσία «Όλα τα αρχεία» της συσκευής. Όταν δεν υπάρχουν δεδομένα για τις προηγούμενες μέρες εμφανίζεται σχετικό μήνυμα, ενώ αν έχουμε ήδη αποθηκεύσει μια φορά το αρχείο, τις επόμενες φορές που θα πατήσουμε πάλι αυτή την επιλογή το αρχείο θα ενημερώνεται. Στην εικόνα που ακολουθεί φαίνεται η μορφή του αρχείου «PreviousDaysResults.txt»



Εικόνα 3.3: Screenshot από το αρχείο PreviousDaysResults.txt

2. Reset: επιλέγοντας το μπορούμε να διαγράψουμε τις μετρήσεις ως τώρα για την σημερινή ημέρα και να αρχίζει η καταγραφή από την αρχή. Όταν επιλεγεί το reset εμφανίζεται ένας διάλογος τύπου material για να επιβεβαιώσουμε αυτή την ενέργεια, ώστε να αποφευχθεί κατά λάθος διαγραφή των δεδομένων

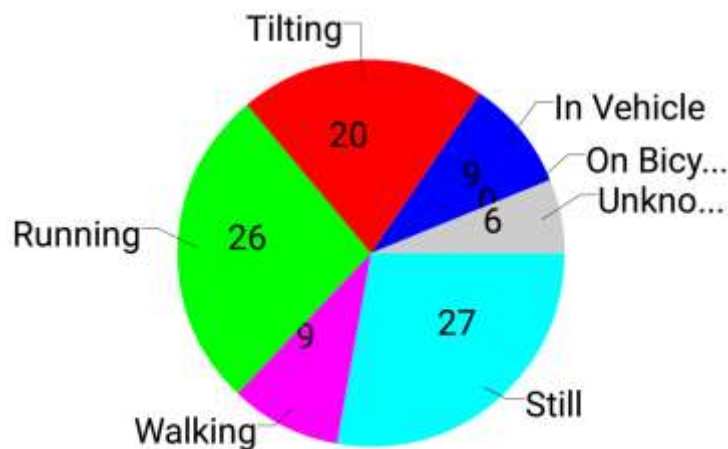


Εικόνα 3.4: Screenshot από την αρχική οθόνη έχοντας επιλέξει το Reset

3. Pie Chart: με αυτή την επιλογή, χρησιμοποιώντας activity transition, εμφανίζεται ένα διάγραμμα πίτας στο οποίο αναπαρίστανται τα ποσοστά των δραστηριοτήτων που έχουν καταγραφεί για την σημερινή ημέρα ως προς τον συνολικό χρόνο της μέχρι στιγμής καταγραφής. Κάθε δραστηριότητα παρουσιάζεται με διαφορετικό χρώμα το οποίο φαίνεται στο κάτω μέρος της εικόνας που ακολουθεί.



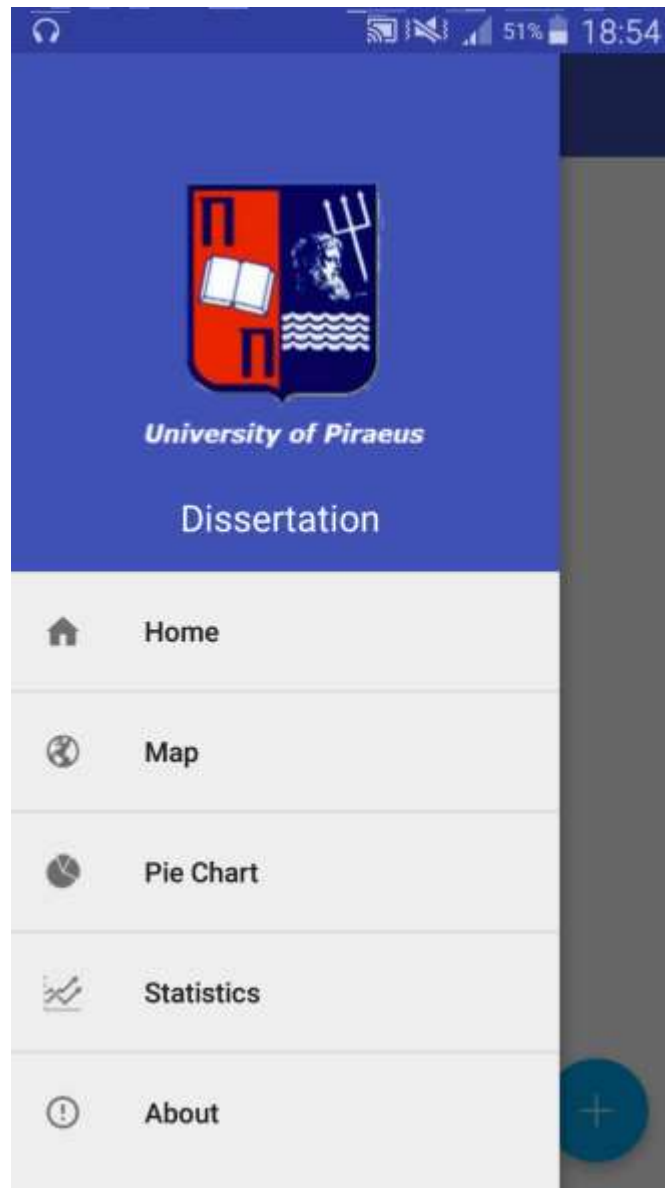
Movement Status Pie Chart.



.Still.Walking.Running.Tilting.In Vehicle
 .On Bicycle.Unknown

Εικόνα 3.5: Screenshot από το διάγραμμα πίτα έχοντας επιλέξει το Pie Chart

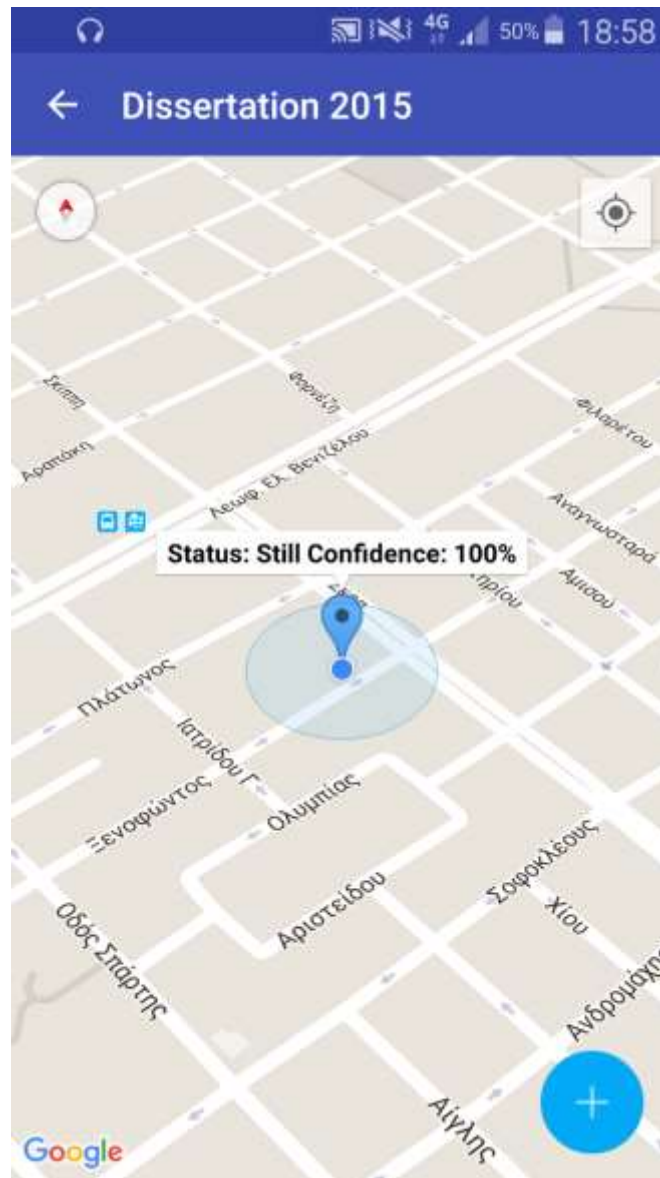
Η αρχική οθόνη διαθέτει επίσης στο δεξιό μέρος Navigation Drawer, από όπου μπορούμε να περιηγηθούμε στις υπόλοιπες οθόνες. Η επιλογή Home μας επιστρέφει στην αρχική οθόνη, ενώ πατώντας το Pie Chart επιστρέφουμε στην οθόνη της προηγούμενης εικόνας. Οι υπόλοιπες τρεις επιλογές παρουσιάζονται παρακάτω.



Εικόνα 3.6 Screenshots από την αρχική οθόνη με ανοιχτό το Navigation Drawer

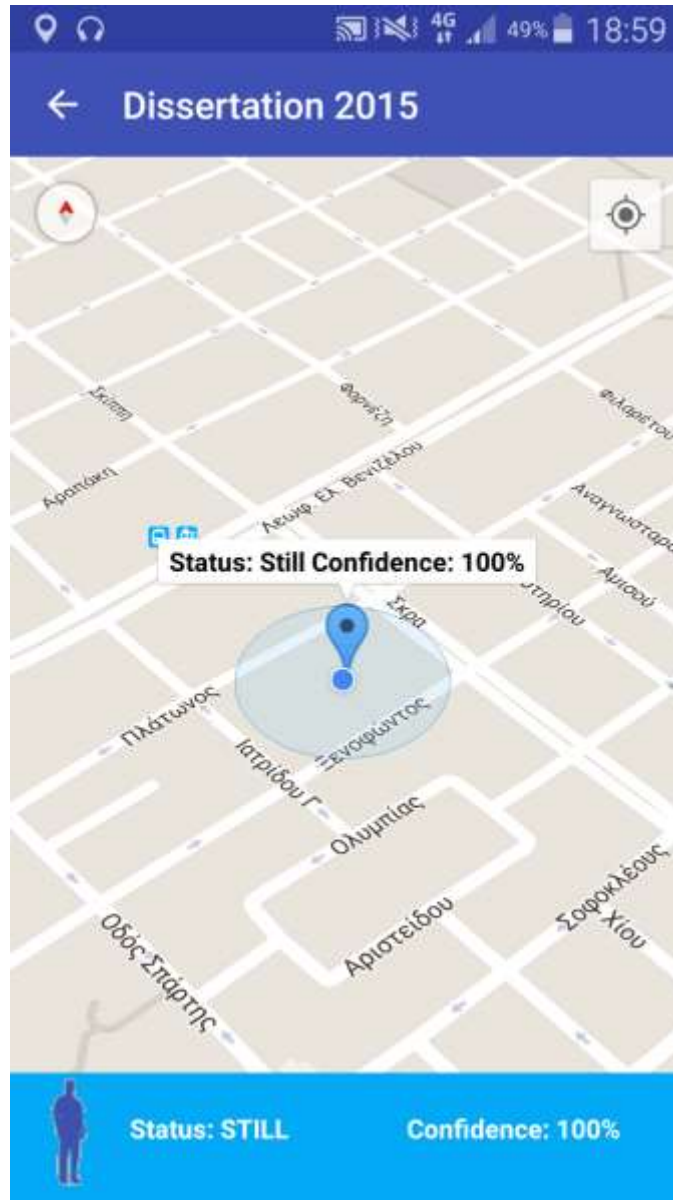
3.2 Οθόνη χάρτη

Επιλέγοντας το Map μεταφερόμαστε στον χάρτη, όπου ο χρήστης βλέπει πού βρίσκεται την δεδομένη στιγμή. Η τοποθεσία του χρήστη επισημαίνεται με τη χρήση ενός δείκτη στον χάρτη, τον οποίο αν επιλέξουμε μας πληροφορεί για την δραστηριότητα του και την αντίστοιχη πιθανότητα της δραστηριότητας. Τόσο η θέση όσο και η δραστηριότητα ανανεώνονται όταν γίνεται καινούργια καταγραφή.



Εικόνα 3.7 Screenshots από την οθόνη χάρτη έχοντας επιλέξει τον δείκτη

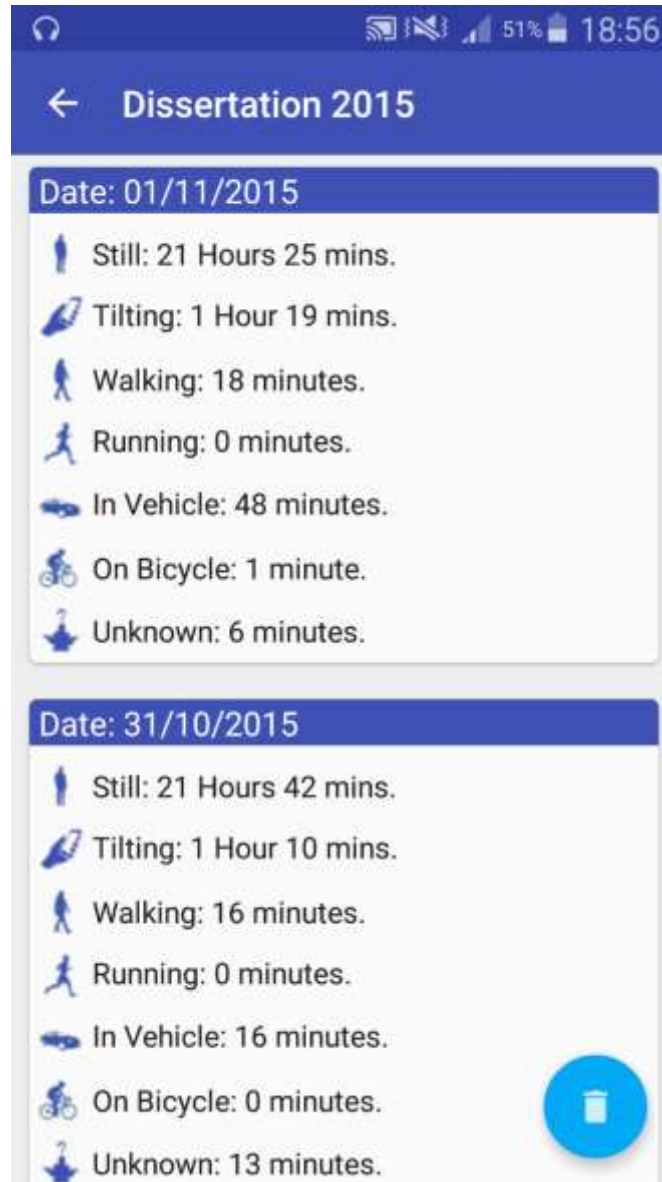
Και σε αυτήν την οθόνη υπάρχει floating action button, το οποίο μετατρέπεται σε toolbar που μας δίνει τις απαραίτητες πληροφορίες για την δραστηριότητα του χρήστη.



Εικόνα 3.8 Screenshot από την οθόνη χάρτη με το toolbar ανοιχτό

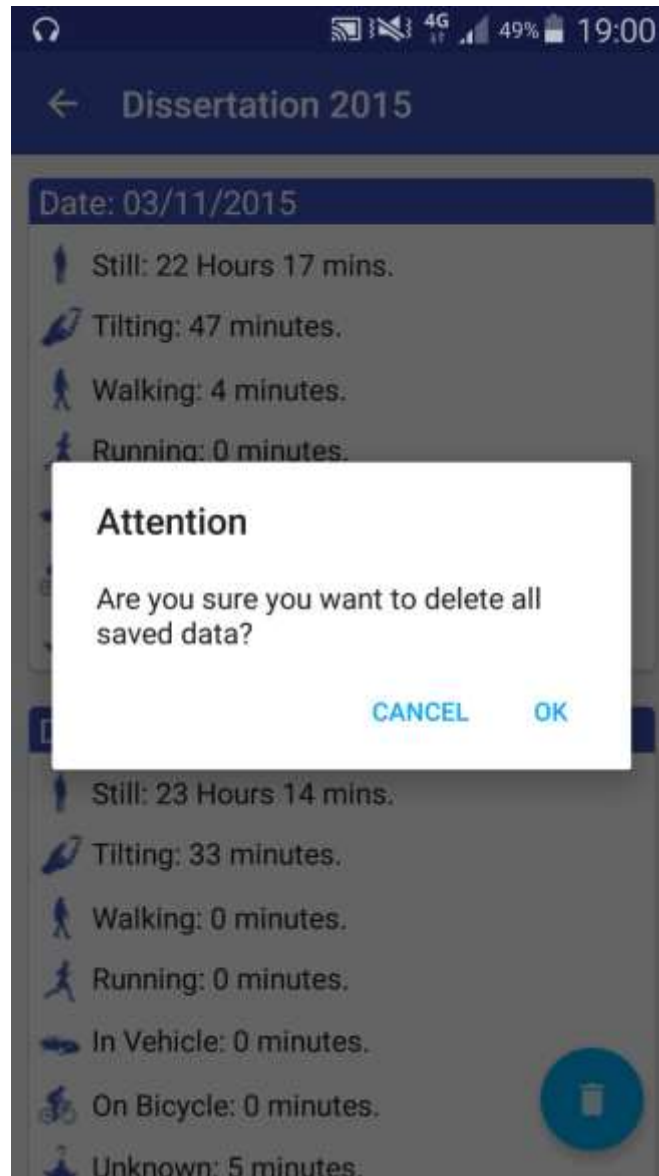
3.3 Οθόνη με στατιστικά

Με την επιλογή του Statistics ο χρήστης βλέπει σε μια λίστα αναλυτικά τα δεδομένα που έχουν καταγραφεί για τη δραστηριότητά του τις προηγούμενες ημέρες. Η λίστα αυτή ανανεώνεται με στο τέλος κάθε μέρας.



Εικόνα 3.8: Screenshot από την οθόνη με τη λίστα με τις καταγραφές των προηγούμενων ημερών

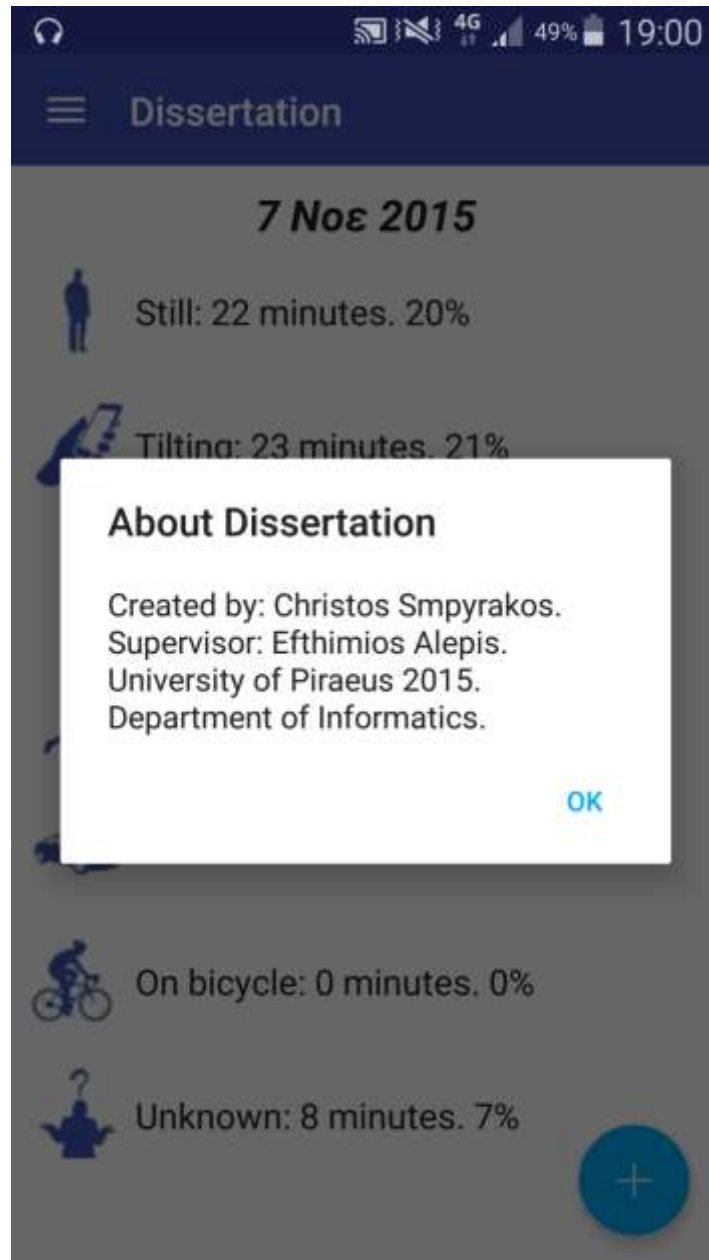
Εάν ο χρήστης επιλέξει το floating action button διαγράφονται όλα τα δεδομένα που έχει συλλέξει η εφαρμογή για τις προηγούμενες μέρες και η λίστα πλέον είναι κενή. Ομοίως και εδώ ζητείται επιβεβαίωση της ενέργειας του χρήστη με διάλογο, για την αποφυγή διαγραφής των δεδομένων σε περίπτωση λάθους.



Εικόνα 3.9: Screenshot από την οθόνη με τη λίστα έχοντας επιλέξει διαγραφή

3.4 Πληροφορίες για την εφαρμογή

Τέλος εάν ο χρήστης επιλέξει το About εμφανίζεται ένας ενημερωτικός διάλογος με πληροφορίες που αφορούν την εφαρμογή.



Εικόνα 3.10: Screenshot από της αρχικής οθόνης έχοντας επιλέξει το About

Κεφάλαιο 4^ο: Πώς και με ποιά εργαλεία αναπτύχθηκε η εφαρμογή

Σε αυτό το κεφάλαιο θα αναφέρουμε τα εργαλεία που χρειάστηκαν για την υλοποίηση της εφαρμογής, και θα προσδιορίσουμε όλο το προγραμματιστικό περιβάλλον. Επίσης θα αναλύσουμε αρκετά τεχνικά σημεία της υλοποίησης και όπου κριθεί απαραίτητο θα παρατεθούν και τα σχετικά τμήματα κώδικα, με σκοπό τη διευκρίνιση τυχών ασαφειών.

4.1 Εργαλεία που χρησιμοποιήθηκαν

4.1.1 Java

Η γλώσσα προγραμματισμού που χρησιμοποιείται για εφαρμογές Android είναι η Java, αν και τον τελευταίο καιρό γίνονται προσπάθειες για να εισέλθουν και άλλες γλώσσες στον χώρο. Η Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού που σχεδιάστηκε από την εταιρεία πληροφορικής Sun Microsystems. Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων άλλων γλωσσών είναι ότι τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα, δηλαδή προσφέρει την ανεξαρτησία λειτουργικού συστήματος και πλατφόρμας.

Απαραίτητο εργαλείο λοιπόν για την ανάπτυξη της εφαρμογής είναι να διαθέτουμε το Java Development Kit (JDK). Για να προχωρήσουμε από εδώ και πέρα χρειαζόμαστε ένα ολοκληρωμένο περιβάλλον ανάπτυξης λογισμικού.

4.1.2 IntelliJ IDE

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε το IntelliJ IDEA Ultimate Edition της εταιρίας JetBrains. Το IntelliJ είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης λογισμικού (integrated development environment, IDE) το οποίο θεωρείται ως ένα από τα καλύτερα της αγοράς. Σύμφωνα με την έκθεση του InfoWorld το 2010 συγκέντρωσε τη μεγαλύτερη βαθμολογία σε σχέση με τα ανταγωνιστικά εργαλεία για προγραμματισμό σε Java (Eclipse, NetBeans, Oracle JDeveloper). Αποτελεί ένα άριστο περιβάλλον για την ανάπτυξη εφαρμογών android και παρέχει ενσωμάτωση πολλών εργαλείων, όπως το Git και βάσεις δεδομένων MySQL. Άλλωστε δεν είναι τυχαίο ότι το Android Studio της Google βασίζεται στο λογισμικό του IntelliJ.

4.1.3 Android SDK

Το Android SDK (Software Development Kit) είναι το επίσημο εργαλείο της Google για την ανάπτυξη εφαρμογών στο Android. Περιέχει μια συλλογή από εργαλεία και βιβλιοθήκες απαραίτητα για τους προγραμματιστές. Πρώτο βήμα στην πορεία της ανάπτυξης είναι η εγκατάσταση και ρύθμισή του. Περιλαμβάνει παραδείγματα εφαρμογών με τον πηγαίο τους κώδικα, βοηθήματα, πληροφορίες και εξομοιωτή για την εκτίμηση της προόδου της εργασίας. Ακόμη, αναλαμβάνει την μεταγλώττιση του κώδικα ώστε να μπορεί να τρέχει στην εικονική μηχανή Dalvik. Οι πλατφόρμες που μπορούν να υποστηρίξουν το Android SDK είναι τα συστήματα που τρέχουν Mac OS X 10.5 ή νεότερο, Windows XP ή νεότερο και οποιαδήποτε σύγχρονη με λειτουργικό Linux. Οι γλώσσες που μπορούν να χρησιμοποιήσουν οι προγραμματιστές για να αναπτύξουν εφαρμογές είναι η Java και C++.

4.1.4 Εξομοιωτής και φυσική συσκευή

Οι εξομοιωτές (emulators) είναι απαραίτητοι για να μπορεί ο προγραμματιστής να παρακολουθεί την πρόοδο και την λειτουργικότητα της εφαρμογής του. Οι εξομοιωτές παρέχονται από το πακέτο Android SDK και υποστηρίζονται άριστα από το IntelliJ IDE. Όμως, αν και ο εξομοιωτής είναι αρκετά αξιόπιστος, στην περίπτωση της εφαρμογής μας χρησιμοποιήθηκε ελάχιστα, καθώς δεν έχει τη δυνατότητα να παρέχει αναγνώριση δραστηριότητας και τοποθεσίας και έτσι η χρήση του περιορίστηκε στη σχεδίαση του γραφικού περιβάλλοντος. Έτσι για τον έλεγχο και ανάπτυξη της λειτουργικότητας χρησιμοποιήθηκε συσκευή τύπου Samsung Galaxy S4 με έκδοση λειτουργικού συστήματος 5.0.1 (Lollipop).

4.1.5 Άλλα εργαλεία που χρησιμοποιήθηκαν

Ως εργαλείο δόμησης (build tool) χρησιμοποιήθηκε το maven. Το maven είναι ένα αυτοματοποιημένο εργαλείο δόμησης που χρησιμοποιείται κυρίως για projects γραμμένα σε Java. Περιγράφει πώς δομείται το λογισμικό και ποιες είναι οι εξαρτήσεις του, απλοποιώντας πολύ την όλη διαδικασία δίνοντας σου τη δυνατότητα να ενσωματώσεις μια βιβλιοθήκη στο project βάζοντας μόνο την εξάρτηση της. Για repository λογισμικού, που είναι μια θέση αποθήκευσης από την οποία τα πακέτα λογισμικού μπορούν να ανακτηθούν και να εγκατασταθεί σε έναν υπολογιστή, χρησιμοποιήθηκε το Sonatype Nexus.

Το version control system που χρησιμοποιήθηκε είναι το Git. Πρόκειται για ένα κατακευματισμένο version control system, που δίνει έμφαση στην ταχύτητα, την ακεραιότητα των δεδομένων, καθώς και την υποστήριξη για κατακευματισμένες και μη γραμμικές ροές εργασίας. Το Git αρχικά σχεδιάστηκε και αναπτύχθηκε από τον Linus Torvalds για ανάπτυξη πυρήνα Linux το 2005.

Επίσης στη μεγάλη πλειοψηφία του κώδικα χρησιμοποιήθηκε το framework androidannotations (μπορείτε να το βρείτε εδώ <http://androidannotations.org/>). AndroidAnnotations είναι ένα framework ανοικτού κώδικα που επιταχύνει την ανάπτυξη λογισμικού για Android. Δημιουργεί αυτόματα κομμάτια κώδικα που επαναλαμβάνονται ή είναι τυποποιημένα σε κάθε κλάση και έτσι επιτρέπει στον προγραμματιστή να επικεντρωθεί στα κομμάτια που είναι πραγματικά σημαντικά. Παράλληλα μικραίνει αρκετά το μέγεθος του κώδικα, τον απλοποιεί και διευκολύνει τη συντήρησή του.

Επιπρόσθετα ενσωματώθηκαν κάποιες εξωτερικές βιβλιοθήκες, κυρίως για την κατασκευή του γραφικού περιβάλλοντος. Αυτές και άλλες πολύ χρήσιμες υπάρχουν στο <https://android-arsenal.com/>.

4.2 Λεπτομέρειες υλοποίησης της εφαρμογής

4.2.1 Λειτουργία της εφαρμογής στο παρασκήνιο

Η εφαρμογή έχει την δυνατότητα να λειτουργεί τόσο στο προσκήνιο όσο και στο παρασκήνιο (background), δηλαδή καταγράφει τιμές ακόμα και όταν ο χρήστης χρησιμοποιεί κάποια άλλη εφαρμογή ή η συσκευή βρίσκεται σε αδράνεια. Αυτό επιτυγχάνεται με την χρήση ενός Alarm Receiver ο οποίος καλείται κάθε ένα λεπτό. Με αυτόν τον τρόπο η εφαρμογή ενεργοποιείται και καταγράφει την δραστηριότητα του χρήστη για την δεδομένη χρονική στιγμή και στην συνέχεια απενεργοποιείται. Εδώ πρέπει να επισημανθεί ότι έχει γίνει η παραδοχή ότι η καταγραφή της δραστηριότητας του χρήστη ανά ένα λεπτό είναι αξιόπιστο χρονικό διάστημα και ότι τα αποτελέσματα ευσταθούν στατιστικά. Επιπροσθέτως στις περιπτώσεις που η συσκευή είναι κλειστή, με το που θα ανοίξει και αρχίσει να

λειτουργεί το σύστημα, η εφαρμογή ενεργοποιείται αυτόματα. Αυτό επιτυγχάνεται με την χρήση των ακόλουθων αδειών στο AndroidManifest.xml.

```
<!-- Keeps the processor from sleeping when a message is received. It is necessary for the WakefulIntentService. -->
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<!-- Allows an application to receive the ACTION_BOOT_COMPLETED that is broadcast after the system finishes booting. -->
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
```

Εικόνα4.1: Τμήμα κώδικα που περιέχει τις απαραίτητες άδειες στο αρχείο AndroidManifest.xml

Κατά την αφύπνιση της εφαρμογής καλείται ένα Service, το οποίο με τη σειρά του καλεί το ActivityRecognitionApi της Google. Το API αυτό είναι ουσιαστικά υπεύθυνο για την αναγνώριση της δραστηριότητας του χρήστη, στο οποίο θα αναφερθούμε αναλυτικότερα στην επόμενη ενότητα.

4.2.2 ActivityRecognitionApi

Η ανίχνευση της κίνησης του χρήστη επιτυγχάνεται με τη χρήση του ActivityRecognitionApi της Google. Το API αυτό αναγνωρίζει την κίνηση του χρήστη διαβάζοντας ανά περιόδους δεδομένα από τους αισθητήρες της συσκευής τα οποία επεξεργάζεται με μοντέλα αναγνώρισης προτύπων και μηχανικής μάθησης. Καλώντας το API αυτό σου επιστρέφει την δραστηριότητα που αναγνώρισε (Detected Activity) για την δεδομένη χρονική στιγμή, καθώς και πόσο σίγουρο είναι για αυτό το αποτέλεσμα (Confidence). Οι τύποι δραστηριοτήτων που υποστηρίζονται αναμένεται να αυξηθούν σύντομα, αλλά αυτή τη στιγμή είναι οι εξής:

- IN_VEHICLE: Ο χρήστης βρίσκεται σε όχημα π.χ. αυτοκίνητο, τρένο.
- ON_BICYCLE: Ο χρήστης κάνει ποδήλατο.
- ON_FOOT: Ο χρήστης μετακινείται με τα πόδια. Αυτή η δραστηριότητα χωρίζεται σε δυο υποδραστηριότητες:
 - a. RUNNING: Ο χρήστης τρέχει.
 - b. WALKING: Ο χρήστης περπατάει.
- STILL: Ο χρήστης είναι ακίνητος
- TILTING: Η γωνία στην οποία βρίσκεται η συσκευή άλλαξε σημαντικά. Αυτό συνήθως συμβαίνει όταν ο χρήστης χρησιμοποιεί εκείνη τη στιγμή τη συσκευή ή καθόταν και σηκώθηκε.
- UNKNOWN: Το API δεν μπόρεσε να αναγνωρίσει την δραστηριότητα.

Επίσης υπάρχουν περιπτώσεις που το API επιστρέφει null τιμές, τις οποίες για τους σκοπούς της εφαρμογής και για να έχουμε όσο το δυνατόν πιο ρεαλιστικά αποτελέσματα τις κατατάσσουμε στην κατηγορία UNKNOWN. Για κάθε μια από τις παραπάνω περιπτώσεις δημιουργήσαμε έναν μετρητή, ο οποίος κάθε φορά που προκύπτει η αντίστοιχη περίπτωση αυξάνεται κατά 1, αντιπροσωπεύοντας το 1 λεπτό και εν συνεχεία η τιμή αυτή αποθηκεύεται στις Shared Preferences στις οποίες γίνεται αναφορά στην επόμενη ενότητα. Επίσης πρέπει να αναφέρουμε ότι όταν ο χρήστης τρέχει ή περπατάει το API

επιστρέφει την τιμή ON_FOOT. Σε αυτή την περίπτωση καλείται μια μέθοδος που επιστρέφει την υποδραστηριότητα με την μεγαλύτερη πιθανότητα, δηλαδή RUNNING ή WALKING. Ακολουθεί το τμήμα κώδικα στο οποίο αναφέρονται τα ανωτέρω.

```
@Override
protected void onHandleIntent(final Intent intent) {
    @Nullable final ActivityRecognitionResult activityRecognitionResult =
ActivityRecognitionResult.extractResult(intent);
    if (activityRecognitionResult != null) {
        DetectedActivity detectedActivity =
activityRecognitionResult.getMostProbableActivity(); // Get the detected activity.
        int confidence = detectedActivity.getConfidence(); // Get the confidence (or
else probability, as a percentage).
        int type = detectedActivity.getType(); // Get the type.
        if (type == DetectedActivity.ON_FOOT) {
            @Nullable final DetectedActivity onFootSubActivity =
walkingOrRunning(activityRecognitionResult.getProbableActivities());
            if (null != onFootSubActivity) {
                detectedActivity = onFootSubActivity;
                confidence = detectedActivity.getConfidence();
                type = detectedActivity.getType();
            }
        }
        movementStatus = MovementStatus.confidence(confidence).
            movementType(getMobilityType(type)).build();
    } else {
        LOGGER.warn("The intent does not contains any activity recognition results.");
        // Null results are counted as unknown.
        preferences.saveOrUpdateCounter(getApplicationContext(),
preferences.UNKNOWN_KEY);
    }
}
```

```
private MovementTypeState getMobilityType(final int type) {
    switch (type) {
        case DetectedActivity.IN_VEHICLE:
            preferences.saveOrUpdateCounter(getApplicationContext(), IN_VEHICLE_KEY);
            return MovementTypeState.DRIVING;
        case DetectedActivity.ON_BICYCLE:
            preferences.saveOrUpdateCounter(getApplicationContext(), ON_BICYCLE_KEY);
            return MovementTypeState.CYCLING;
        case DetectedActivity.ON_FOOT:
            preferences.saveOrUpdateCounter(getApplicationContext(), ON_FOOT_KEY);
            return MovementTypeState.ON_FOOT;
        case DetectedActivity.WALKING:
            preferences.saveOrUpdateCounter(getApplicationContext(), WALKING_KEY);
            return MovementTypeState.WALKING;
        case DetectedActivity.RUNNING:
            preferences.saveOrUpdateCounter(getApplicationContext(), RUNNING_KEY);
            return MovementTypeState.RUNNING;
        case DetectedActivity.STILL:
            preferences.saveOrUpdateCounter(getApplicationContext(), STILL_KEY);
            return MovementTypeState.STILL;
        case DetectedActivity.TILTING:
            preferences.saveOrUpdateCounter(getApplicationContext(), TILTING_KEY);
            return MovementTypeState.TILTING;
        default:
            preferences.saveOrUpdateCounter(getApplicationContext(), UNKNOWN_KEY);
    }
}
```

```

        return MovementTypeState.UNKNOWN;
    }
}

```

Εικόνα4.2: Τμήμα κώδικα που διαχειρίζεται την αναγνώριση δραστηριότητας του χρήστη

4.2.3 Αποθήκευση δεδομένων

Για την αποθήκευση των δεδομένων, αν και η αρχική σκέψη ήταν η δημιουργία μιας Sqlite βάσης δεδομένων, τελικά χρησιμοποιήθηκε το Shared Preferences API που παρέχεται από το Android SDK. Οι Shared Preferences δίνουν τη δυνατότητα αποθήκευσης και ανάκτησης <<πρωτόγονων>> (primitive) τιμών συνόλων δεδομένων, όπως boolean , byte , char , short , int , long , float και double. Εδώ πρέπει να τονιστεί ότι οι τιμές που αποθηκεύονται είναι επίμονες (persistent), δηλαδή ακόμα και αν σταματήσει η εφαρμογή ή κλείσει η συσκευή οι τιμές διατηρούνται.

Η προσέγγιση αυτή ήταν ιδιαίτερα χρήσιμη για την υλοποίηση της συγκεκριμένη εφαρμογής, διότι τα δεδομένα των μετρητών που αποθηκεύονται για την εκάστοτε δραστηριότητα είναι τύπου int. Ακόμα αποθηκεύονται και δυο τιμές τύπου String, μια εκ των οποίων είναι η ημερομηνία και η άλλη ένα Object που περιέχει όλους τους μετρητές που αποθηκεύτηκαν για την συγκεκριμένη ημερομηνία. Το Object μετατρέπεται σε String με τη χρήση της βιβλιοθήκης Google Gson. Η Gson είναι μια βιβλιοθήκη της Java που μπορεί να χρησιμοποιηθεί για τη μετατροπή Java αντικειμένων σε JSON μορφή. Μπορεί επίσης να χρησιμοποιηθεί για να μετατρέψει μια συμβολοσειρά JSON σε ένα ισοδύναμο αντικείμενο της Java.

Ακόμα πρέπει να σημειωθεί ότι με τη χρήση της μεθόδου OnSharedPreferenceChanged, η οποία καλείται όποτε αλλάξει κάποια τιμή, επιτυγχάνεται εκτός της αυτοματοποίησης της όλης διαδικασίας αποθήκευσης και η αυτόματη ανανέωση της βασικής οθόνης ή του χάρτη όταν ο χρήστης έχει την εφαρμογή στο προσκήνιο.

Ειδικότερα χρησιμοποιήθηκαν δυο διαφορετικές Shared Preferences, μια για να αποθηκεύει τους μετρητές και μια για να αποθηκεύει το αντικείμενο που τους περιέχει όταν η μέρα αλλάζει. Κατά την αλλαγή της μέρας τα αποτελέσματα των μετρητών αποθηκεύονται στο αντικείμενο και στη συνέχεια οι μετρητές μηδενίζονται για να καταγράψουν από την αρχή τιμές για την νέα μέρα που μόλις ξεκίνησε. Ο κώδικας που χρησιμοποιήθηκε ακολουθεί παρακάτω.

```

Private void saveObjectAndResetCounters() {
    final SharedPreferences sharedPreferences =
context.getSharedPreferences(Preferences.COUNTERS_PREFERENCES,
Application.MODE_PRIVATE);
    final int stillCounter = sharedPreferences.getInt(STILL_KEY, 0);
    final int onFootCounter = sharedPreferences.getInt(ON_FOOT_KEY, 0);
    final int walkingCounter = sharedPreferences.getInt(WALKING_KEY, 0);
    final int runningCounter = sharedPreferences.getInt(RUNNING_KEY, 0);
    final int tiltingCounter = sharedPreferences.getInt(TILTING_KEY, 0);
    final int inVehicleCounter = sharedPreferences.getInt(IN_VEHICLE_KEY, 0);
    final int onBicycleCounter = sharedPreferences.getInt(ON_BICYCLE_KEY, 0);
    final int unknownCounter = sharedPreferences.getInt(UNKNOWN_KEY, 0);
    final String previousDay = sharedPreferences.getString(PREVIOUS_DAY, "");

    final MovementCounters movementCounters = MovementCounters.date(previousDay).
        still(minutesToHours(stillCounter)).
        onFoot(minutesToHours(onFootCounter)).
        walking(minutesToHours(walkingCounter)).

```

```

        running(minutesToHours(runningCounter)).
        tilting(minutesToHours(tiltingCounter)).
        inVehicle(minutesToHours(inVehicleCounter)).
        onBicycle(minutesToHours(onBicycleCounter)).
        unknown(minutesToHours(unknownCounter)).build();

        final SharedPreferences objectPreferences =
context.getSharedPreferences(Preferences.OBJECT_PREFERENCES,
Application.MODE_PRIVATE);
        final SharedPreferences.Editor objectsEditor = objectPreferences.edit();
        final Gson gson = new Gson();
        final String json = gson.toJson(movementCounters); // myObject - instance of
MyObject
        int dayCounter = objectPreferences.getInt(DAY_COUNTER, 0);
        dayCounter++;
        objectsEditor.putString("MyObject" + String.valueOf(dayCounter), json);
        objectsEditor.putInt(DAY_COUNTER, dayCounter);
        objectsEditor.apply();
        resetCounters();
}

```

Εικόνα4.3: Τμήμα κώδικα που διαχειρίζεται τις Shared Preferences και την αποθήκευση των τιμών στο Object που δημιουργήσαμε

4.2.4 Τοποθεσία

Η εύρεση της τοποθεσίας του χρήστη στον χάρτη, γίνεται με τη χρήση του FusedLocationProviderApi για την ανάκτηση της τελευταίας γνωστής θέσης της συσκευής. Το FusedLocationProviderApi είναι ένα από τα API εντοπισμού τοποθεσίας που παρέχεται από τις υπηρεσίες του Google Play (Google Play Services). Διαχειρίζεται την υποκείμενη τεχνολογία εντοπισμού θέσης και παρέχει ένα απλό API, ώστε να μπορεί ο προγραμματιστής να καθορίσει τις απαιτήσεις σε υψηλό επίπεδο, όπως η υψηλή ή χαμηλή ακρίβεια της τοποθεσίας. Επίσης βελτιστοποιεί την κατανάλωση ισχύος της μπαταρίας της συσκευής.

Για να συνδεθούμε με το API, θα πρέπει πρώτα να δημιουργήσουμε ένα στιγμιότυπο του Google Play Services API Client.

```

@Background(id = "test2")
public void establishGoogleApiConnection() {
    LOGGER.trace("Starting connection with the Google API Client.");
    fusedLocationProviderApi = LocationServices.FusedLocationApi;
    googleApiClient = new GoogleApiClient.Builder(getApplicationContext())
        .addApi(LocationServices.API)
        .addConnectionCallbacks(this)
        .addOnConnectionFailedListener(this)
        .build();
    googleApiClient.connect();
    googleApiClient.blockingConnect(); // Confirm connection existence. If it exists
then continue, if not block till connection is created.
    triggerStatusRequest();
}

```

Εικόνα4.4: Τμήμα κώδικα σύνδεσης με το API

Έπειτα αφού έχουμε συνδεθεί επιλέγουμε τις παραμέτρους του ερωτήματος τοποθεσίας που θέλουμε (ακρίβεια, κόστος μπαταρίας, διάρκεια κ.α.) και ζητάμε την τοποθεσία μέσω της μεθόδου `triggerStatusRequest()`. Με το που βρεθεί η τοποθεσία του χρήστη καλείται η μέθοδος `updateLocation()`, όπου γίνεται ζουμ στον χάρτη και εμφανίζεται ένας δείκτης σε αυτήν. Όπως αναφέρθηκε σε προηγούμενο κεφάλαιο πατώντας στον δείκτη εμφανίζεται η δραστηριότητα του χρήστη την δεδομένη στιγμή. Παρακάτω ακολουθεί ένα τμήμα του αντίστοιχου κώδικα:

```
@UiThread
void triggerStatusRequest() {
    LOGGER.trace("Requesting phone location data.");
    final LocationRequest locationRequest = setUpLocationRequest();
    fusedLocationProviderApi.requestLocationUpdates(googleApiClient, locationRequest,
this);
    updateLocation();
}

/**
 * We need a new location request every time, because each location request has an
 * expiration duration, i.e. a timeout after which it is useless.
 */
LocationRequest setUpLocationRequest() {
    final LocationRequest locationRequest = LocationRequest.create();
    locationRequest.setNumUpdates(LOCATION_REQUEST_NUMBER_OF_LOCATIONS_REQUIRED);
    locationRequest.setExpirationDuration(LocationRequest.ExpirationDurationMillis);
    locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
    locationRequest.setInterval(fetchingInterval);
    return locationRequest;
}

private void updateLocation() {
    googleMap.clear();
    if (currentLocation != null) {
        animateCamera(currentLocation);
    } else {
        final LocationManager locationManager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);
        final Criteria criteria = new Criteria();
        final Location lastKnownLocation =
locationManager.getLastKnownLocation(locationManager.getBestProvider(criteria, true));
        if (lastKnownLocation != null) {
            animateCamera(lastKnownLocation);
        }
    }
}
}
```

Εικόνα4.5: Τμήμα κώδικα που διαχειρίζεται την τοποθεσία του χρήστη

Τόσο η τοποθεσία, όσο και η δραστηριότητα του χρήστη ανανεώνονται στον χάρτη σε πραγματικό χρόνο. Αυτό επιτυγχάνεται με την χρήση των μεθόδων `onLocationChanged` και `onSharedPreferencesChanged` που καλούνται αυτόματα όταν αλλάξει η τοποθεσία του χρήστη ή αυξηθεί ο μετρητής κάποιας δραστηριότητας αντίστοιχα. Παρακάτω φαίνονται οι δυο αυτές μέθοδοι:

```
@Override
public void onLocationChanged(final Location location) {
    currentLocation = location;
}
```

```

        updateLocation();
    }

    @Override
    public void onSharedPreferenceChanged(final SharedPreferences sharedPreferences, final
String key) {
        LOGGER.info("A setting with key {} has changed.", key);
        if (key.equals(Preferences.TOTAL_KEY)) {
            updateLocation();
        }
    }
}

```

Εικόνα4.6: Τμήμα κώδικα ανανέωσης των τιμών με τις μεθόδους onLocationChanged και OnSharedPreferenceChanged

4.2.5 Διάγραμμα πίτας

Η δημιουργία του διαγράμματος πίτας γίνεται με την βοήθεια της βιβλιοθήκης AchartEngine. Πρόκειται για μια βιβλιοθήκη λογισμικού δημιουργίας διαγραμμάτων και γραφημάτων για Android εφαρμογές και είναι διαθέσιμη στην ιστοσελίδα <http://www.achartengine.org/>. Το μόνο που χρειάστηκε στην περίπτωση μας ήταν να δώσουμε τις τιμές των μετρητών και να παραμετροποιήσουμε λίγο την γραφική αναπαράσταση και τα υπόλοιπα τα έκανε η βιβλιοθήκη από μόνη της.

4.2.6 Αποθήκευση δεδομένων σε αρχείο κειμένου

Για την υλοποίηση αυτής της λειτουργίας το βασικότερο που είχαμε να κάνουμε ήταν να ορίσουμε το path που θα αποθηκευτεί το αρχείο και το όνομά του. Από εκεί και πέρα γράφουμε το ζεύγος κλειδιών-τιμών που είναι αποθηκευμένες στις OBJECT_PREFERENCES στο αρχείο και το αποθηκεύουμε. Βασική προϋπόθεση για τα παραπάνω είναι να έχουμε δηλώσει στο AndroidManifest.xml την παρακάτω άδεια.

```

<!-- This gives permission to android logging log4j to write to external storage and
for saving previous days data file. Also required by maps to cache map tile data. -->
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>

```

Εικόνα 4.7: Άδεια εγγραφής σε εξωτερική μνήμη

Κεφάλαιο 5^ο: Συμπεράσματα και μελλοντικές επεκτάσεις

5.1 Συμπεράσματα

Έχοντας ολοκληρώσει την ανάπτυξη της εφαρμογής, είμαστε σε θέση να εξάγουμε διάφορα χρήσιμα συμπεράσματα. Όσον αφορά την εφαρμογή, φαίνεται πως είναι λειτουργική, καθώς τα αποτελέσματα που καταγράφει κρίνονται ικανοποιητικά ως προς την ακρίβειά τους, τουλάχιστον από χρήστες που χρειάζονται τέτοιου είδους εφαρμογές για γενικούς σκοπούς και όχι εξειδικευμένους, όπως εντατική προπόνηση και υγεία. Επίσης η λειτουργία στο παρασκήνιο σε συνδυασμό με τη χαμηλή κατανάλωση μπαταρίας, καθώς και το γραφικό περιβάλλον απέσπασαν θετικά σχόλια από όσους την χρησιμοποίησαν.

Η όλη διαδικασία που απαιτήθηκε από την έρευνα, την αρχή και τον σχεδιασμό της εφαρμογής, μέχρι την ολοκλήρωσή της μπόρεσε να μας δώσει μια συνολική εικόνα για την ανάπτυξη εφαρμογών στην πλατφόρμα Android. Ο προγραμματισμός σε Android είναι μια ευχάριστη πρόκληση που δίνει πολλές δυνατότητες και επιλογές στον προγραμματιστή, του δίνει την αίσθηση της δημιουργίας και τον αναγκάζει να χρησιμοποιήσει διαφορετικά πολύ ενδιαφέροντα εργαλεία. Από την άλλη όμως ο προγραμματιστής καλείται να αντιμετωπίσει αρκετά προβλήματα που προκύπτουν κατά την ανάπτυξη. Συγκεκριμένα αν και φαινομενικά η συγκεκριμένη εφαρμογή δείχνει σχετικά απλή, στην πράξη υπήρξαν διάφορα τεχνικά προβλήματα. Κύριος λόγος είναι ότι στο documentation που παρέχει η Google, αν και καλύπτει σχεδόν όλα τα θέματα, στα περισσότερα οι οδηγίες και τα παραδείγματα είναι πολύ βασικά και για απλές χρήσεις, κάτι το οποίο γίνεται εμφανές όταν προσπαθήσουμε να μπούμε σε λεπτομέρειες και να υλοποιήσουμε λειτουργίες που δεν εμπύπτουν στα παραδείγματα που δίνονται. Αυτό σε αναγκάζει να αναζητήσεις αλλού για πιθανές λύσεις και πολλές φορές θα χρειαστεί να πειραματιστείς μόνος σου και να αυτοσχεδιάσεις μέχρι να φτάσεις στο επιθυμητό αποτέλεσμα. Τέλος ένα επιπλέον πρόβλημα ήταν η διατήρηση της συμβατότητας με παλαιότερες εκδόσεις, κυρίως λόγω της χρήσης material design.

5.2 Μελλοντική επέκταση

Στην εποχή μας που ανάπτυξη των έξυπνων συσκευών και των υπηρεσιών που προσφέρουν είναι ραγδαία και συνεχής κάθε εφαρμογή πρέπει να αναβαθμίζεται και να βελτιώνεται. Έτσι και οι δυνατότητες της εφαρμογής που υλοποιήσαμε μπορούν να επεκταθούν μελλοντικά προς αρκετές κατευθύνσεις. Μερικές ιδέες προς επέκταση μπορούμε να συλλέξουμε από τις εμπορικές εφαρμογές που παρουσιάστηκαν στον κεφάλαιο 2. Υπάρχει η δυνατότητα να ενσωματωθούν αρκετές λειτουργίες που να κάνουν την εφαρμογή πιο ακριβές εργαλείο για αθλητές, όπως η λειτουργία προπόνησης, οι φωνητικές οδηγίες, οι ειδοποιήσεις, η χαρτογράφηση της διαδρομής κ.α. Επίσης ως προς τον τομέα της υγείας, οι δυνατότητες μέτρησης καρδιακών παλμών και καύσης θερμίδων θα μπορούσαν να είναι δυο πιθανές προσθήκες, σε συνδυασμό με την ειδοποίηση του χρήστη ή άλλων χρηστών, όταν οι τιμές αυτές φτάνουν σε επικίνδυνα επίπεδα, ειδικά σε περιπτώσεις που οι χρήστες είναι ηλικιωμένοι. Μια ακόμα οπτική είναι η αξιοποίηση της πληροφορίας όταν ο χρήστης βρίσκεται σε όχημα. Με τη συλλογή πληροφοριών για τις συχνές διαδρομές που κάνει ο χρήστης, η εφαρμογή να μπορεί να προτείνει εναλλακτικές διαδρομές ανάλογα με την μέρα και την ώρα και ιδανικά συγκεντρώνοντας πληροφορίες από πολλούς χρήστες να μπορεί να παρέχει μια πρόβλεψη για την συνολική κίνηση στους δρόμους. Τέλος, καθώς οι δραστηριότητες που θα αναγνωρίζουν οι συσκευές θα αυξηθούν, θα δοθούν επιπλέον δυνατότητες στις εφαρμογές, παρέχοντας υπηρεσίες και σε άλλα πεδία για τα οποία δεν είμαστε σε θέση να γνωρίζουμε ακόμα.

Κεφάλαιο 6^ο: Βιβλιογραφία

- [1] Android Developers: What is Android? <http://developer.android.com/guide/basics/what-is-android.html>
- [2] Λειτουργικό σύστημα Android και αρχιτεκτονική του [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))
- [3] Global operations systems market share 2015 <http://www.forbes.com>
- [4] Material Design <https://www.google.com/design/spec/material-design/introduction.html#>
- [5] Activity Recognition <https://developers.google.com/android/reference/com/google/android/gms/location/ActivityRecognition>
- [6] 12 Android Apps for Athletes of All training levels <http://www.sporttechie.com/2013/10/27/12appsathletes/>
- [7] 5 Best smartphone running apps <http://lifelife.com/5929709/five-best-smartphone-running-apps>
- [8] The Best Running iPhone and Android apps of 2015 <http://www.healthline.com/health/fitness-exercise/top-running-iphone-android-apps#2>
- [9] Activity recognition implementation on android tutorial <http://tutsberry.com/activity-recognition-implementation-on-android/>
- [10] <http://androidannotations.org/>
- [11] <https://android-arsenal.com/>
- [12] <https://en.wikipedia.org>
- [13] <http://developer.android.com/>
- [14] <https://play.google.com>
- [15] <http://stackoverflow.com/>