Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Ανάπτυξη εφαρμογής Android αναγνώρισης της τρέχουσας κατάστασης του χρήστη με την χρήση συστημάτων αισθητήρων κινητών τηλεφώνων** **Android Application development of current user activity using mobile phone sensors**
Ονοματεπώνυμο Φοιτητή	**Πηνελόπη Κόλλια**
Πατρώνυμο	**Κωνσταντίνος**
Αριθμός Μητρώου	**ΜΠΣΠ/13052**
Επιβλέπων	**Ευθύμιος Αλέπης, Επίκουρος καθηγητής**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)	(υπογραφή)	(υπογραφή)
Ευθύμιος Αλέπης	Μαρία Βίρβου	Γεώργιος Τσιχριντζής

Νοέμβριος 2015

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1 : ΕΙΣΑΓΩΓΗ

ΠΕΡΙΛΗΨΗ

Η τεχνολογική εξέλιξη των κινητών και οι δυνατότητες που προσφέρουν πλέον, που οφείλεται κατά κύριο λόγο στο λειτουργικό android έχουν δημιουργήσει πρόσφορο έδαφος για την ανάπτυξη μιας σειράς ενδιαφέρουσων εφαρμογών που βελτιώνουν την καθημερινότητα είτε σε μικρό βαθμό είτε σε μεγάλο.

Στόχος της παρούσας εργασίας είναι να χρησιμοποιήσει τις δυνατότητες των σύγχρονων κινητών και με την βοήθεια των διάφορων ενσωματομένων αισθητήρων να καταλήξει σε χρήσιμα συμπεράσματα για τις δραστηριότητες του χρήστη. Αυτά τα συμπεράσματα θα βοηθήσουν τόσο σε ατομικό όσο και σε συλλογικό επίπεδο.

Σε αυτή την εργασία χρησιμοποιούνται αισθητήρες χαμηλής ενεργειακής κατανάλωσης που βοηθούν στην αναγνώριση πέντε βασικών επιπέδων κίνησης του χρήστη. Ο χρήστης της εφαρμογής έχει την δυνατότητα στο τέλος της ημέρας να δει ποσοτικά το πως δραστηριοποιήθηκε όλη την ημέρα, ενώ του δίνεται η δυνατότητα να δει τα στατιστικά του πληθυσμού με βάση το φύλο και την ηλικία ως προς τις δραστηριότητες του. Πράγμα πολύ χρήσιμο καθ'ότι μπορεί να λειτουργήσει παρακινητικά ως προς την βελτίωση της δικής του ζωής.

ABSTRACT

The evolution of mobile technology and the possibilities that it now offers, mainly due to the android operating system, have created a new ground for a series of applications that improve everyday life.

The objective of this paper is to use the capabilities of modern mobile phones and with the help of various on-board sensors to arrive at useful conclusions about the user's activities. These findings will help both individually and collectively.

Lastly, we use energy efficient sensors that help in the identification of five key user movement activities. The user of the application has the opportunity at the end of the day, to see quantitatively how active he has been during the day while having the possibility to see general statistics of the population grouped by either sex or age for each activity.

We hope that this application will be stimulating to its users to improve their lifestyle.

ΚΕΦΑΛΑΙΟ 2 : ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ ANDROID

Λειτουργικό Σύστημα Android

Το Android είναι ένα λειτουργικό σύστημα που τρέχει κυρίως σε κινητές συσκευές με οθόνη αφής όπως κινητά τηλέφωνα και tablets. Το Android είναι λειτουργικό σύστημα ανοιχτού κώδικα που ανήκει στην Google [20]. Λόγω του ότι είναι λογισμικό ανοικτού κώδικα μπορεί να τροποποιηθεί από οποιοδήποτε όπως για παράδειγμα κατασκευαστές συσκευών, εταιρίες τηλεπικοινωνιών και διάφορους developers. Το λειτουργικό είναι σχεδιασμένο χρησιμοποιώντας της γλώσσες C/C++ και Java, βασίζεται στο Linux και χρησιμοποιεί τον Monolithic kernel (μονολιθικό πυρήνα) που είναι μια τροποποιημένη έκδοση του πυρήνα του λειτουργικού συστήματος Linux (Linux kernel). Ένα επιπλέον πλεονέκτημα του Android είναι πως τρέχει και στις δύο κύριες αρχιτεκτονικές ARM και x86 καθώς επίσης και σε MIPS αρχιτεκτονικές.

2.1 Αρχιτεκτονική Android

Η κύρια πλατφόρμα υλικού για το Android είναι η αρχιτεκτονική ARM, και επίσημα υποστηρίζονται η x86 και MIPSαρχιτεκτονικές. Και οι δύο 64-bit και 32-bit εκδόσεις των τριών αρχιτεκτονικών που υποστηρίζονται από την απελευθέρωση του Android 5.0 ανεπίσημη έργο του Android-x86 είχε παράσχει υποστήριξη για την x86 και MIPS αρχιτεκτονικές πριν από την επίσημη υποστήριξη Από το 2012, το Androidσυσκευές με επεξεργαστές Intel άρχισαν να εμφανίζονται, συμπεριλαμβανομένων των τηλεφώνων και ταμπλέτες. Αν κερδίσουμε την υποστήριξη για πλατφόρμες 64-bit, το Android έγινε για πρώτη φορά να τρέχει σε 64-bit x86 και, στη συνέχεια, στο ARM64.

Οι ελάχιστες απαιτήσεις υλικού έχουν αναβαθμιστεί σε βήματα με την πάροδο του χρόνου, με τις νέες κυκλοφορίες εκδόσεων του Android. Αρχικό το ελάχιστα ήταν 32 MB μνήμης RAM αλλά λιγότερο από 128 MB δεν συνιστάται, με το πρώτο κινητό τηλέφωνο HTC Dream ("ναυαρχίδα") που χρησιμοποιούσε 192 MB RAM με 32 MB μνήμης flash, και μια αρχιτεκτονική ARM με επεξεργαστή 200 MHz (ARMv5). Από το Νοέμβριο του 2013 και με την νεα έκδοση Android 4.4, είναι για συσκευές ARM-based που απαιτούν ARMv7 επεξεργαστή (Android 5.0 υποστηρίζει επίσης ARMv8-A), ενώ η ελάχιστη ποσότητα της μνήμης RAM είναι 512 MB. Η απαιτούμενη ελάχιστη μνήμη που είναι διαθέσιμη για το Android 4.4 είναι 340 MB, και όλες οι συσκευές με λιγότερα από 512 MB μνήμης RAM, πρέπει να αναφέρονται ως "χαμηλής μνήμης RAM" συσκευές.

Τον Οκτώβριο του 2011, με το Android 4.0, η μονάδα επεξεργασίας γραφικών (GPU) υποστηρίζει πλέον το OpenGL ES 2.0 (και το ES 1.0) η επιτάχυνση υλικού είναι υποχρεωτική, ανεξάρτητα από το εάν οι εφαρμογές χρησιμοποιούν άμεσα το OpenGL ES ή όχι. Αργότερα, το Android 4.3 πρόσθεσε υποστήριξη για OpenGL ES 3.0 αν χρησιμοποιείται, με παράλληλη την

υποστήριξη και για τις δύο παλαιότερες εκδόσεις (ES 2.0 και 1.0) που εξακολουθεί να είναι υποχρεωτική.

Εκτός από την εκτέλεση απευθείας σε x86 με βάση το υλικό, το Android μπορεί επίσης να τρέχει σε x86 αρχιτεκτονική με τη χρήση ενός εξομοιωτή Android το οποίο είναι μέρος του Android SDK, ή με τη χρήση BlueStacks ή του Andy.

Το Android αποτελείται από ορισμένες συνιστώσες λογισμικού οι οποίες συνθέτουν ένα ενιαίο και ολοκληρωμένο σύστημα .Έτσι το σύστημα αυτο μπορεί να παρέχει τα μέσα που απαιτούνται για την χρήση νέων εφαρμογών όπως άλλωστε συμβαίνει και με τα λειτουργικά συστήματα των ηλεκτρονικών υπολογιστών.Το android αποτελείται από 4 επίπεδα και από 5 ομάδες συνιστωσών τα οποία περιγράφονται παρακάτω ξεκινώντας από τα χαμηλότερα προς τα υψηλότερα επίπεδα.

Linux Kernel

Το Android βασίζεται στον πυρήνα του Linux για βασικές λειτουργίες όπως είναι η διαχείριση των drivers της συσκευής,διαχείριση μνήμης,διαχείριση διεργασιών καθώς και δικτύωσης που συνεπάγεται την διαχείριση των διεπαφών δικτύου που διαθέτει κάθε συσκευή(πχ. GSM,HSDPA,Wi-Fi,Bluetooth κτλ).

Native Libraries

Οι βιβλιοθήκες του Android είναι γραμμένες στις γλώσσες C και C++ και μπορούν να χρησιμοποιηθούν μέσω κατάλληλου Interface της Java.Μερικές από τις κυριότερες είναι α) η βιβλιοθήκη Surface Manager για την δημιουργία παραθύρων καθώς και διασδιάστατων και τρισδιάστατων γραφικών, β) η βιβλιοθήκη Media Framework που περιέχει αποκωδικοποιητές (codec) για αναπαραγωγή αρχείων πολυμέσων όπως MPEG,MP3 κτλ. Γ) η βιβλιοθήκη SQLite για την υποστήριξη της βάσης δεδομένων SQL και Δ) η βιβλιοθήκη WebKit για την υποστήριξη των φυλλομετρητών(browsers).

Android Runtime

Όπως φαίνεται και στην παρακάτω εικόνα η συνιστώσα του Android Runtime αποτελείται από:

Α)Βασικές βιβλιοθήκες για την διεπαφή των εφαρμογών Java με το περιβάλλον της συσκευής στην οποία εκτελούνται.

Β)Τη Dalvik Virtual Machine η οποία είναι υπεύθυνη για την δημιουργία των εκτελέσιμων αρχείων των εφαρμογών προκειμένου να τα τρέξει το λειτουργικό σύστημα.

Κάθε εφαρμογή του Android είναι γραμμένη σε γλώσσα Java την οποία το λειτουργικό σύστημα δεν την αντιλαμβάνεται απευθείας. Για τον λόγο αυτό η Dalvic Virtual Machine αναλαμβάνει τη δημιουργια των εκτελέσιμων αρχείων *.dex (Dalvic Executable) τα οποία εκτελούνται από το λειτουργικό σύστημα. Κάθε εκτελέσιμο πρόγραμμα εκτελείται από την δική του Virtual Machine, ακόμα και όταν εκτελούνται παράλληλα, με αποτέλεσμα τα διαφορετικά προγράμματα να μην επηρεάζουν το ένα το άλλο και σε περίπτωση που προκύψει κάποιο σφάλμα σε κάποιο από αυτά να μην προκαλέσει προβλήματα στα υπόλοιπα.

Application Framework

Εφόσον το Android προσφέρει μία ανοιχτή πλατφόρμα ανάπτυξης εφαρμογών είναι επόμενο ορισμένες από τις εφαρμογές να είναι αρκετά προχωρημένες και καινοτόμες. Οι εφαρμογές έχουν πρόσβαση στις βασικές βιβλιοθήκες του λειτουργικού συστήματος, μέσω κατάλληλων διεπαφών,και μέσω του Application Framework μπορούν με την σειρά τους να παρέχουν επιπρόσθετες λειτουργίες-υπηρεσίες προς άλλες εφαρμογές,εφόσον κάτι τέτοιο φυσικά δεν περιορίζεται από τις πολιτικές ασφάλειας του Application Framework. Μερικές από τις πιο βασικές οντότητες που περιλαμβάνονται στο πλαίσιο του Application Framework είναι:

- **View System**:
  Επιτρέπει την χρήση λιστών,πλαισίων,πεδίων κειμένου,κουμπιών κτλ.

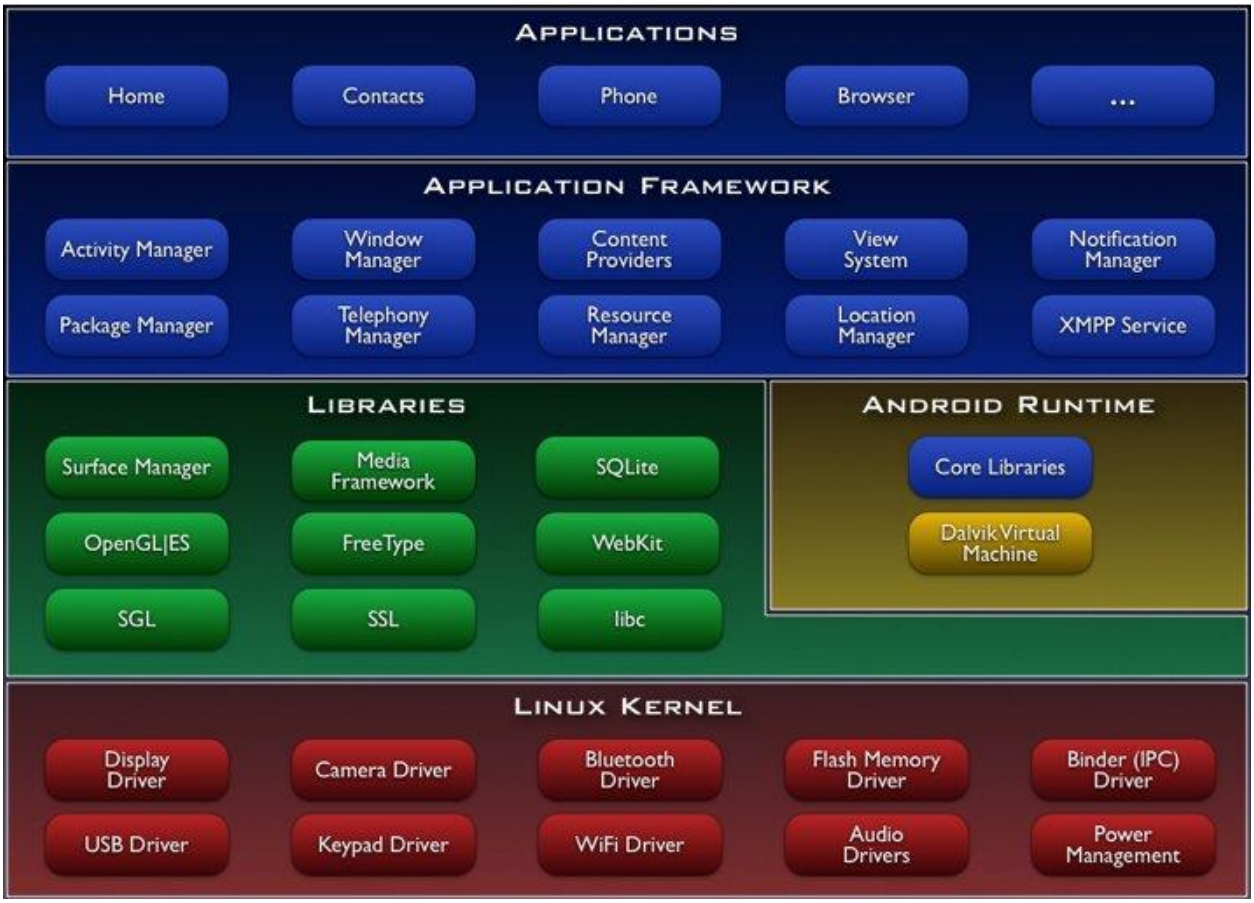
- **Content Providers**:
  Επιτρέπει στις εφαρμογές την πρόσβαση σε δεδομένα άλλων εφαρμογών ή τον διαμοιρασμό των δικών τους δεδομένων,όπως οι επαφές.

- **Resource Manager**:
  Παρέχει την πρόσβαση σε πόρους όπως γραφικά και σε αρχεία σχετικά με την διάταξη των στοιχείων του γραφικού περιβάλλοντος. Απλούστερα ότι δεν είναι κώδικας.

- **Notification Manager**:
  Διαχειρίζεται τα μηνύματα των εφαρμογών που εμφανίζονται στην status bar,όπως εισερχόμενα μυνήματα,ραντεβού κτλ.

- **Activity Manager**:

  Διαχειρίζεται τον κύκλο ζωής των εφαρμογών και παρέχει την δυνατότητα μετάβασης στις προγενέστερες καταστάσεις τους.



Εικόνα : Συνιστώσες λογισμικού λειτουργικού συστήματος Android

Applications

Στην ομάδα των Applications βρίσκονται οι εφαρμογές που θα χρησιμοποιηθούν τελικά οι χρήστες με διαφάνεια ως προς το τι συμβαίνει πίσω από αυτές ή το τι απαιτείται για την εκτέλεση τους από το λειτουργικό σύστημα. Μερικές από τις πιο γνωστές από τις εφαρμογές αυτές είναι ο browser, email client, αποστολή και λήψη SMS,προβολή χαρτών σε συνδιασμό με το στίγμα της συσκευής εάν διαθέτει δέκτη GPS,ημερολόγιο,διαχείριση επαφών,παιχνίδια,RSS readers και πολλές άλλες. Όλες οι εφαρμογές όπως έχει ήδη αναφερθεί είναι γραμμένες σε Java και μπορούν να τρέχουν πολλές παράλληλα χωρίς να επηρεάζει η μια την άλλη.

2.2 Ιστορία εκδόσεων του Android.

Το Android που κυκλοφορεί διάφορες εκδόσεις με ονομασίες που σου ανοίγουν την όρεξη για… νέα χαρακτηριστικά, όπως τα παλαιότερα CupCake (1.5), Donut (1.6), Éclair (2.0, 2.1), GingerBread (2.3) αλλά και FroYo (2.2) Honeycomb (3.0) που υλοποιείται σε ταμπλέτες ενώ υπάρχουν πλέον και οι εκδόσεις Honeycomb (3.1) και Honeycomb (3.2). Δεν πρέπει να ξεχάσουμε να αναφέρουμε και την επερχόμενη έκδοση Ice Cream Sandwich η οποία θα είναι διαθέσιμη από τον Οκτώβρη.

Από την «παρθενική» έκδοση Android 1.0, η οποία κυκλοφόρησε το Σεπτέμβριο του 2008, μέχρι την αμέσως επόμενη, 1.1 που παρουσιάστηκε το Φεβρουάριο του 2009, χρειάστηκε ένας χρόνος για να γίνει η έκρηξη των καινοτόμων εκδόσεων και των σημαντικών αλλαγών που επέφεραν για τον χρήστη.

Το πρώτο smartphone που «έτρεξε» Android είναι το T-Mobile G1 κατασκευασμένο από την HTC με οθόνη αφής TFT–LCD 3,2”, full qwerty πληκτρολόγιο, πρόσβαση σε Gmail, YouTube, Google maps, Google talk, Google calendar, κάμερα 3,2MP με αυτόματη εστίαση και κάρτα μνήμης micro SD.

Μερικές από τις χαρακτηριστικές αλλαγές των εκδόσεων που ακολούθησαν και οι εκπρόσωποί τους:

Εικόνα: Ιστορικό εκδόσεων Android

Android 1.5 Cupcake

Την άνοιξη του 2009 κυκλοφόρησε. Και ουσιαστικά έχουμε και την αρχη των ονοματων με γλυκά. Ηταν η εκδοση με την οποία έχουμε την υποστήριξη των widgets και ώς νέα χαρακτηριστικά του ήταν η εγγραφή video και playback σε μορφή MPEG-4 και 3GP και τα εφέ κίνησης κατά την περιήγηση στις διαφορετικές οθόνες. Η έκδοση 1.5 (CupCake) παρέχει τη δυνατότητα αυτόματης σύνδεσης ακουστικών headset σε συγκεκριμένη απόσταση, ανέβασμα εικόνων στο Picasa και βίντεο στο YouTube κατευθείαν από την κινητή συσκευή του χρήστη, ενώ παρέχει εικονικό πληκτρολόγιο με πρόβλεψη λέξεων και νέα widgets για την αρχική οθόνη. Διαθέτει κάμερα 5MP με αυτόματη εστίαση και κάρτα μνήμης micro SD.Τέλος, το CupCake (1.5) «τρέχει» το HTC Hero με οθόνη αφής TFT-LCD, 3,2”, ανάλυσης 320x480p (HVGA)

Android 1.6 Donut

Το Σεπτεμβριο του 2009 κυκλοφόρησε. Εννοείτε ότι συμπεριελάμβανε νέες βελτιώσεις όπως ευκολότερη αναζήτηση και δυνατότητα προεπισκόπησης εφαρμογών σε όσες συσκευές είχαν Google Play, δείκτες χρήσης της μπαταριάς και αυτόματη περιστροφή οθόνης.Με το Donut δίνεται έμφαση στην αναζήτηση από την αρχική οθόνη με bookmarks, ιστορικό, επαφές κ.ά. αλλά και στη φωνητική αναζήτηση, ενώ υποστηρίζονται και οθόνες αναλύσεων WVGA. Τηλέφωνα της έκδοσης αυτής είναι το LG GT 540, το οποίο υποστηρίζει οθόνη αναλύσεων WVGA και επιπλέον προσφέρει ευκολία στην εύρεση των επαφών, διαθέτει qwerty

πληκτρολόγιο για γρήγορη αποστολή SMS και 3G και Wi-Fi για να είναι ο χρήστης συνδεδεμένος στο διαδύτιο όπου και να βρίσκεται, και το Sony Ericsson X10 με επεξεργαστή Snapdragon 1GHz της Qualcomm και οθόνη αφής 4", 854x480pixels. Στην πίσω όψη βρίσκεται η κάμερα 8,1MP και το LED Flash κάτω ακριβώς από τον φακό της.

Android 2.0 Eclair

Τον Οκτώβριο του 2009 κυκλοφόρησε, μόλις ένα μήνα μετά το Donut. Συμπεριελάμβανε αρκετές μικρές βελτιώσεις, όπως Bluetooth 2.1, κινούμενο φόντο στην οθόνη του home, και πληκτρολόγιο με έξυπνο λεξιλόγιο που μαθαίνει ανάλογα με την χρήση των λέξεων. Με το όνομα Éclair έχουμε και επόμενες επίσης γλυκές εκδόσεις, τις Éclair 2.0 και Éclair 2.1 που προχωρούν ακόμη πιο πολύ, διαθέτοντας υποστήριξη HTML5,νέο browser UI, Google Maps 3.1.2, ψηφιακό ζουμ,ενσωματωμένη υποστήριξη για flash στην κάμερα, βελτιωμένο εικονικό πληκτρολόγιο, δυνατότητα αντίληψης multi-touch, live wallpapers, και bluetooth 2.1. Το Motorola Droid είναι ένας εκπρόσωπος της έκδοσης 2.0 (Éclair), με επεξεργαστή Arm Cortex A8 550MHz. Ο σχεδιασμός του είναι slide με πλήρες qwerty πληκτρολόγιο και οθόνη αφής 3,7", ανάλυσης 480×854p. Επίσης, έχουμε το Samsung galaxy S, με οθόνη αφής 4.0″ Super AMOLED που «τρέχει» Android 2.1 (Éclair) με επεξεργαστή 1GHz, εσωτερική μνήμη 8GB, Wi-Fi και bluetooth 3.0, παρέχει ταχύτητα, χώρο και απαιτούμενες δυνατότητες σύνδεσης. Επιπλέον, έχουμε το HTC Wildfire με οθόνη αφής 3,2″ QVGA, το οποίο υποστηρίζει Adobe flash με άνετη πλοήγηση στο διαδυκτιο και παρέχει λήψη online βίντεο και παιχνίδιών. Διαθέτει κάμερα στα 5MP με αυτόματη εστίαση και LED φλας.

Android 2.2-2.2.3 Froyo

Το Μάιο του 2010 κυκλοφόρησε και το ακρόνυμο της εκδοσης Froyo που αποτελεί συντόμευση της φράσης "Frozen Yogurt" (παγωμένο γιαούρτι). Είναι η πρώτη έκδοση του Android που υποστήριζε Adobe Flash. Μερικές από τις βελτιώσεις ήταν σύνδεση μέσω USB και Wi-Fi hotspot, η γρήγορη εναλλαγή γλώσσας κατά την πληκτρολόγηση και η δυνατότητα απενεργοποίησης της λειτουργίας δικτύου δεδομένων. Η έκδοση 2.2 (FroYo-Frozen Yogurt) αναβάθμισε αισθητά την ταχύτητα του OS, αλλά και τη γενικότερη απόδοση, παρέχει υποστήριξη Adobe flash 10.1 και επιλογή εγκατάστασης εφαρμογών στην κάρτα μνήμης, διαθέτει Market με δυνατότητα αυτόματων updates, ενσωματώνει τον Chrome V8 JavaScript στα browsers applications. Στη FroYo βρίσκουμε και τη δυνατότητα χρήσης της συσκευής για διαμοιρασμό ίντερνετ μέσω Wi-Fi σε άλλες συσκευές (tethering). Εκπροσωπος της έκδοσης αυτής είναι το LG Optimus 2x που διαθέτει οθόνη αφής 4″, λειτουργικό Android Froyo και διπύρηνο επεξεργαστή Nvidia Tegra 2 με ισχυρή κάμερα 8MP, δυνατότητα εγγραφής και αναπαραγωγής βίντεο full HD και συνδεσιμότητα HDMI και DLNA. Ενας άλλος εκπρόσωπος είναι το HTC Desire Z με Android Froyo και οθόνη αφής 3,7″, ανάλυσης WVGA (480x800p), συρόμενο Qwerty πληκτρολόγιο για γρήγορα e-mail, 5MP κάμερα και δέκτη GPS. Χάρη στον ενσωματωμένο Adobe Flash Player και προβάλλει κάθε ιστοσελίδα

Android 2.3-2.3.7 Gingerbread

Τον Δεκέμβρη του 2010 Κυκλοφόρησε η πιο πολυπληθής έκδοση του Android. Ήταν πολύ πιο γρήγορο και εύχρηστο από τις προηγούμενες εκδόσεις και έδινε στους δημιουργούς εφαρμογών μεγαλύτερες δυνατότητες. Οι βελτιώσεις περιλάμβαναν υποστήριξη πολλών καμερών στην

συσκευή όπως και μεγαλύτερης ανάλυσης οθόνη. Η GingerBread (2.3), που κυκλοφόρησε το Δεκέμβριο του 2010, υποστηρίζει πλέον πολύ μεγάλα μεγέθη οθονών και αναλύσεων, διαθέτει επανασχεδιασμένο multi-touch πληκτρολόγιο, προεγκατεστημένη υποστήριξη για τηλεφωνικές κλήσεις μέσω ίντερνετ (VoIP), download manager για κατέβασμα μεγάλων αρχείων, λειτουργίες copy-paste σε όλο το λειτουργικό, καθώς και προεγκατεστημένη υποστήριξη για πολλαπλές κάμερες. Μεγάλος εκπρώσοπος της αποτελεί το Sony Xperia Play, συσκευή προσανατολισμένη στο gaming, «τρέχει» GingerBread (2.3) με επεξεργαστή Scorpion ARM7 και ταχύτητα στα 1GHz. Η οθόνη του είναι 4 ιντσών με ανάλυση 480x854p και η εσωτερική του μνήμη 400MB. Άλλοι εκπρώσοποι είναι τα HTC Cha Cha και HTC Salsa με GingerBread (2.3) που ενσωματώνουν ένα εξειδικευμένο Facebook πλήκτρο, για πρόσβαση με ένα άγγιγμα, στην υπηρεσία του Facebook μέσα από την εμπειρία HTC Sense.

Android 3.0-3.2 Honeycomb

Τον Φεβρουάριο του 2011 κυκλοφόρησε και ήταν διαθέσιμη μόνο για tablets. Οι βελτιώσεις της περιλαμβάνουν γρήγορη πρόσβαση σε χαρακτηριστικά της κάμερας, καλύτερο πληκτρολόγιο κατάλληλο για μεγάλες οθόνες, εκτέλεση πολλαπλών λειτουργιών και εύκολη μετάβαση από την μια στην άλλη. Η έκδοση Honeycomb (3.1) προσέθεσε την επιλογή να μεταφέρεται περιεχόμενο απευθείας από συσκευές USB, ενώ τέλος η έκδοση 3.2 προσέθεσε διάφορες δυνατότητες και ευκολίες για χρήστες και developers όπως τη μεταφορά αρχείων από κάρτες SD και δυνατότητα Zoom to Fill. Αυτή τη στιγμή στην αγορά υπάρχουν αρκετές ταμπλέτες που τρέχουν Android Honeycomb όπως το Motorola Xoom, το Samsung Galaxy Tab 10.1, το επερχόμενο tablet της Sony, Sony Tablet S, το Asus Eee Pad Trasformer και το Toshiba AT200 το οποίο μάλιστα είναι εξοπλισμένο με την τελευταία έκδοση του Android Honeycomb 3.2. Η πρώτη συσκευή που έτρεχε σε Honeycomb ήταν το tablet Motorola Xoom.

Android 4.0-4.0.2 Ice Cream Sandwich

Τον Οκτώβρη του 2011 κυκλοφόρησε και έφερε πληθώρα αλλαγών στο λειτουργικό σύστημα. Η δυνατότητα χρήσης “μαλακών” κουμπιών δηλαδή των κουμπιών πάνω στην οθόνη (πίσω, αρχική, κλπ) είναι πλέον πραγματικότητα καθώς μέχρι τότε όλα τα κινητά είχαν εξωτερικά κουμπιά για αυτές τις λειτουργίες. Μερικές άλλες δυνατότητες ήταν καλύτερη χρήση των φωνητικών εντολών, το Face Unlock, βελτίωση της ταχύτητας απόκρισης και αναδιαμόρφωση του περιβάλλοντος χρήσης. Αυτή η έκδοση διαθέτει καλύτερο web browser με tabs, ανανεωμένο γραφικό περιβάλλον με αρκετά 3D στοιχεία και ανανεωμένο εικονικό πληκτρολόγιο. Δίνει ειδική έκδοση του Gmail για tablets, δυνατότητα βιντεοκλήσεων μέσω εφαρμογής Google Talk, ανανεωμένη έκδοση Google Maps και βελτιστοποιημένη εφαρμογή για ανάγνωση Google e-books.

Android 4.1-4.3.1 Jelly Bean

Τον Ιούνιο του 2012 κυκλοφόρησε και αποτελεί την καλύτερη έκδοση του Android μέχρι σήμερα. Το περιβάλλον χρήσης και η απόκρισή του είναι πιο γρήγορη και καλοφτιαγμένη από ποτέ ενώ περιλαμβάνει πάρα πολλές μικρές βελτιώσεις σε όλο το σύστημα, όπως για παράδειγμα στην κάμερα και στην χρήση φωνής για υπαγόρευση κειμένου. Η έκδοση 4.1 Jelly Bean του OS σκαρφάλωσε στο 28,4% από 25% τον προηγούμενο μήνα και ξεπέρασε το ποσοστό του Ice

Cream Sandwich, το οποίο ανέρχεται πλέον στο 27,5%. Παρόλα αυτά, η έκδοση 2.3 Gingerbread του λειτουργικού παραμένει στην πρώτη θέση και είναι εγκατεστημένη στο 38,5% των Android συσκευών. Το Jelly Bean εφαρμόστηκε για πρωτη φορά στο tablet Google Nexus 7 ενώ η έκδοση Android 4.2 πρωτοεμφανίστηκε στα Nexus 4 και Nexus 10.

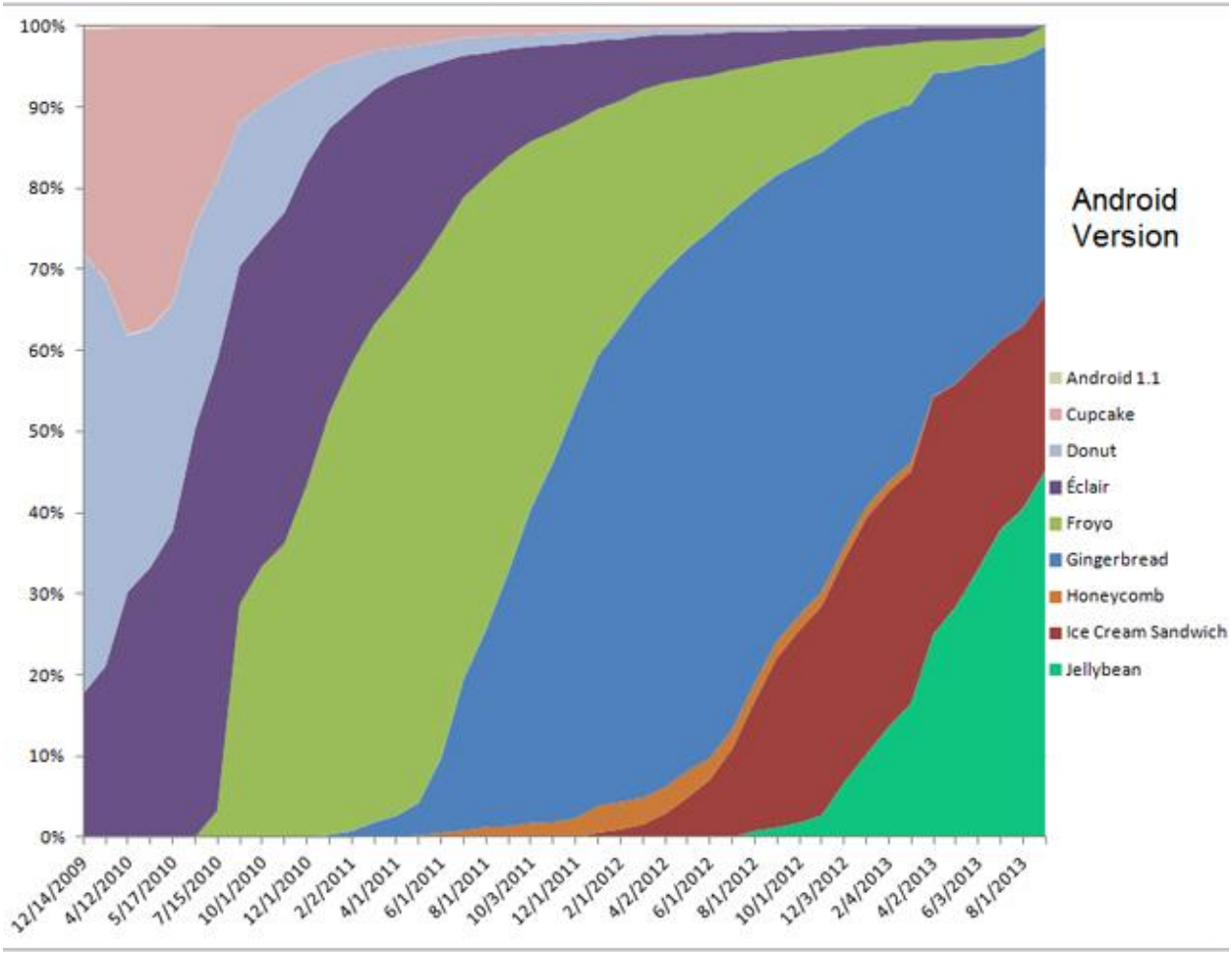
Android 4.4-4.4.4 Kit Kat

Τον Σεπτεμβρη του 2013 κυκλοφόρησε. Αν και αρχικά ήταν να ονομαστεί "Key Lime Pie" ("KLP") κωδική ονομασία, το όνομα άλλαξε καθώς πολύ λίγοι άνθρωποι ξέρουν πραγματικά τη γεύση αυτής της πίτα. Το KitKat έκανε το ντεμπούτο του στο Nexus της Google 5 και έχει βελτιστοποιηθεί για να τρέχει σε ένα μεγαλύτερο εύρος συσκευών από τις προηγούμενες εκδόσεις του Android, αφού έχει ως συνιστώμενη ελάχιστη μνήμη RAM 512 .

Android 5.0 "Lollipop"

Τον Ιούνιο του 2014 ανακοινώθηκε από την Google η νεα έκδοση του Android αλλά θα είναι διαθέσιμη τον τον Νοέμβριο του 2014 για επιλεγμένες συσκευές που τρέχουν Android, συμπεριλαμβανομένων τις συσκευές Nexus. Η Lollipop ως βασική αλλαγή που θα παρουσιάσει είναι ένα επανασχεδιασμένο περιβάλλον εργασίας χρήστη χτισμένο γύρω από μία διαδραστική σχεδιαστική γλώσσα που αποκαλείται ως «υλικό σχεδιασμού». Επιπλέον βελτιώσεις του συστήματος θα είναι οτι το notifications system θα επιτρέπει κοινοποιήσεις που θα μπορούν να προσπελαστούν από την lockscreen, και να εμφανίζονται μαζί με εφαρμογές ως banner πάνω την κορυφή της οθόνης. Εσωτερικές αλλαγές που θα γίνουν επίσης στην πλατφόρμα, πιό συγκεκριμένα, το Android Runtime (ART) θα αντικαταστήσει το Dalvik με μία πιο βελτιωμένη έκδοση την γνωστή ως Project Volta, η οποία θα παρέχει καλύτερη απόδοση των εφαρμογών πράγμα το οποίο σημαίνει ότι θα υπάρξουν και αλλαγές που αποσκοπούν στη βελτίωση και βελτιστοποίηση της χρήσης της μπαταρίας.

Το παρακάτω γράφημα παρουσιάζει όλες τις εκδόσεις του Android και το **ποσοστό** των συσκευών που τις έφεραν κάθε στιγμή, από το τέλος του 2009 μέχρι τις αρχές του 2013. Μέχρι σήμερα η Jelly Bean έχει ξεπεράσει το **50%**, με τις υπόλοιπες συσκευές να τρέχουν την **ICS** αλλά κυρίως την **Gingerbread**! Ελάχιστες, φυσικά, έχουν ξεμείνει με παλαιότερες εκδόσεις, λιγότερο από το 3% των συσκευών. Είθε η πλατφόρμα να συνεχίσει να εξελίσσεται πάνω στο σωστό δρόμο καθώς έχει πολλά να μας προσφέρει ακόμη, αλλά δείχνει τα πρώτα σημεία κόπωσης...



2.3 Η γλώσσα προγραμματισμού Java

Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε *Java* τρέχουνε ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh (σύντομα θα τρέχουν και σε PlayStation καθώς και σε άλλες κονσόλες παιχνιδιών) χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό

λειτουργικό σύστημα. Για να επιτευχθεί όμως αυτό χρειαζόταν κάποιος τρόπος έτσι ώστε τα προγράμματα γραμμένα σε Java να μπορούν να είναι «κατανοητά» από κάθε υπολογιστή ανεξάρτητα του είδους επεξεργαστή (Intel x86, IBM, Sun SPARC, Motorola) αλλά και λειτουργικού συστήματος (Windows, Unix, Linux, BSD, MacOS). Ο λόγος είναι ότι κάθε κεντρική μονάδα επεξεργασίας κατανοεί διαφορετικό κώδικα μηχανής. Ο συμβολικός κώδικας (*assembly*) που μεταφράζεται και εκτελείται σε Windows είναι διαφορετικός από αυτόν που μεταφράζεται και εκτελείται σε έναν υπολογιστή Macintosh. Η λύση δόθηκε με την ανάπτυξη της *Εικονικής Μηχανής* (*Virtual Machine*ή VM ή ΕΜ στα ελληνικά).

Αφού γραφεί κάποιο πρόγραμμα σε Java, στη συνέχεια μεταγλωτίζετται μέσω του μεταγλωττιστή javac, ο οποίος παράγει έναν αριθμό από αρχεία .class (κώδικας byte ή bytecode). Ο κώδικας byte είναι η μορφή που παίρνει ο πηγαίος κώδικας της Java όταν μεταγλωττιστεί. Όταν πρόκειται να εκτελεστεί η εφαρμογή σε ένα μηχάνημα, το Java Virtual Machine που πρέπει να είναι εγκατεστημένο σε αυτό θα αναλάβει να διαβάσει τα αρχεία .class. Στη συνέχεια τα μεταφράζει σε γλώσσα μηχανής που να υποστηρίζεται από το λειτουργικό σύστημα και τον επεξεργαστή, έτσι ώστε να εκτελεστεί (αυτό συμβαίνει με την παραδοσιακή Εικονική Μηχανή (Virtual Machine). Πιο σύγχρονες εφαρμογές της εικονικής Μηχανής μπορούν και μεταγλωττίζουν εκ των προτέρων τμήματα bytecode απευθείας σε κώδικα μηχανής (εγγενή κώδικα ή native code) με αποτέλεσμα να βελτιώνεται η ταχύτητα). Χωρίς αυτό δε θα ήταν δυνατή η εκτέλεση λογισμικού γραμμένου σε Java. Η JVM είναι λογισμικό που εξαρτάται από την πλατφόρμα, δηλαδή για κάθε είδος λειτουργικού συστήματος και αρχιτεκτονικής επεξεργαστή υπάρχει διαφορετική έκδοση του. Έτσι υπάρχουν διαφορετικές JVM για Windows, Linux, Unix, Macintosh, κινητά τηλέφωνα, παιχνιδομηχανές κλπ.

Οτιδήποτε θέλει να κάνει ο προγραμματιστής (ή ο χρήστης) γίνεται μέσω της εικονικής μηχανής. Αυτό βοηθάει στο να υπάρχει μεγαλύτερη ασφάλεια στο σύστημα γιατί η εικονική μηχανή είναι υπεύθυνη για την επικοινωνία χρήστη - υπολογιστή. Ο προγραμματιστής δεν μπορεί να γράψει κώδικα ο οποίος θα έχει καταστροφικά αποτελέσματα για τον υπολογιστή γιατί η εικονική μηχανή θα τον ανιχνεύσει και δε θα επιτρέψει να εκτελεστεί. Από την άλλη μεριά ούτε ο χρήστης μπορεί να κατεβάσει «κακό» κώδικα από το δίκτυο και να τον εκτελέσει. Αυτό είναι ιδιαίτερα χρήσιμο για μεγάλα κατανεμημένα συστήματα όπου πολλοί χρήστες χρησιμοποιούν το ίδιο πρόγραμμα συγχρόνως.

Ακόμα μία ιδέα που βρίσκεται πίσω από τη *Java* είναι η ύπαρξη του συλλέκτη απορριμμάτων (*Garbage Collector*). Συλλογή απορριμμάτων είναι μία κοινή ονομασία που χρησιμοποιείται στον τομέα της πληροφορικής για να δηλώσει την ελευθέρωση τμημάτων μνήμης από δεδομένα που δε χρειάζονται και δε χρησιμοποιούνται άλλο. Αυτή η απελευθέρωση μνήμης στη Java είναι αυτόματη και γίνεται μέσω του συλλέκτη απορριμμάτων. Υπεύθυνη για αυτό είναι και πάλι η εικονική μηχανή η οποία μόλις «καταλάβει» ότι ο σωρός (heap) της μνήμης (στη Java η συντριπτική πλειοψηφία των αντικειμένων αποθηκεύονται στο σωρό σε αντίθεση με τη C++ όπου αποθηκεύονται κυρίως στηστοίβα) κοντεύει να γεμίσει ενεργοποιεί το συλλέκτη απορριμμάτων. Έτσι ο προγραμματιστής δε χρειάζεται να ανησυχεί για το πότε και αν θα ελευθερώσει ένα συγκεκριμένο τμήμα της μνήμης, ούτε και για σφάλματα δεικτών. Αυτό είναι ιδιαίτερα σημαντικό γιατί είναι κοινά τα σφάλματα προγραμμάτων που οφείλονται σε λανθασμένο χειρισμό της μνήμης.

Παρόλο που η εικονική μηχανή προσφέρει όλα αυτά (και όχι μόνο) τα πλεονεκτήματα, η *Java* αρχικά ήταν πιο αργή σε σχέση με άλλες προγραμματιστικές γλώσσες υψηλού επιπέδου

(high-level) όπως η C και η C++. Εμπειρικές μετρήσεις στο παρελθόν είχαν δείξει ότι η C++ μπορούσε να είναι αρκετές φορές γρηγορότερη από την Java. Ωστόσο γίνονται προσπάθειες από τη Sun για τη βελτιστοποίηση της εικονικής μηχανής, ενώ υπάρχουν και άλλες υλοποιήσεις της εικονικής μηχανής από διάφορες εταιρίες (όπως της IBM), οι οποίες μπορεί σε κάποια σημεία να προσφέρουν καλύτερα και σε κάποια άλλα χειρότερα αποτελέσματα. Επιπλέον με την καθιέρωση των μεταγλωττιστών JIT (Just In Time), οι οποίοι μετατρέπουν τον κώδικα byte απευθείας σε γλώσσα μηχανής, η διαφορά ταχύτητας από τη C++ έχει μικρύνει κατά πολύ.

Οι τελευταίες εκδόσεις του javac με τη χρήση της τεχνολογίας Hot Spot έχουν καταφέρει αξιόλογες επιδόσεις που πλησιάζουν ή και ξεπερνούν σε μερικές περιπτώσεις τον εγγενή κώδικα.

Όλα τα εργαλεία που χρειάζεται κάποιος για να γράψει Java προγράμματα έρχονται δωρεάν, από το περιβάλλον ανάπτυξης μέχρι εργαλεία *build* όπως το Apache Ant και βιβλιοθήκες, ενώ υπάρχουν πολλές διαφορετικές υλοποιήσεις της *Εικονικής Μηχανής* και του *μεταγλωττιστή* (πχ the GNU Compiler for Java) της *Java*.

Πολλά εργαλεία και τεχνολογίες σε Java μπορούν να βρεθούν στο Apache Software Foundation αλλά και στοJakarta Project

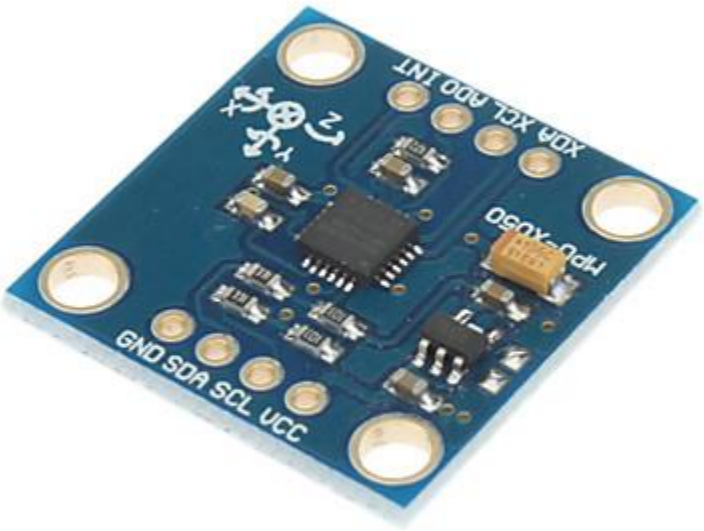
Για να να γράψει κάποιος κώδικα Java δε χρειάζεται τίποτα άλλο παρά έναν επεξεργαστή κειμένου, όπως το Σημειωματάριο (Notepad) των Windows ή ο vi (γνωστός στο χώρο του Unix). Παρ'όλ'αυτά, ένα ολοκληρωμένο περιβάλλον ανάπτυξης (*IDE*) βοηθάει πολύ, ιδιαίτερα στον εντοπισμό σφαλμάτων (debugging). Υπάρχουν αρκετά διαθέσιμα, ενώ πολλά από αυτά έρχονται δωρεάν.

2.4 **Αισθητήρες κινητών τηλεφώνων**

Αισθητήρας είναι μία συσκευή που ανιχνεύει χαρακτηριστικά του περιβάλλον- τός της. Ανιχνεύει γεγονότα ή αλλαγές σε ποσότητες και παρέχει την αντίστοιχη έξοδο ως ένα ηλεκτρικό ή οπτικό σήμα. Οι αισθητήρες χρησιμοποιούνται σε καθημερινά αντικείμενα όπως για παράδειγμα στα κουμπιά των ανελκυστήρων ή στις λάμπες που ανάβουν ακουμπώντας απλά τη βάση τους. Λόγω της τεχνολογικής προόδου, η χρήση των αισθητήρων έχει επεκταθεί πέρα από τα παραδοσιακά πεδία της μέτρησης της θερμοκρασίας ή της πίεσης. Παρόλα αυτά, αναλογικοί αισθητήρες όπως ποτενσιόμετρα χρησιμοποιούνται ευρέως ακόμα και τώρα. Η ευαισθησία ενός αισθητήρα δείχνει πόσο αλλάζει η έξοδός του ανάλογα με τις αλλαγές που υφίσταται η είσοδος που πρόκειται να μετρηθεί. Κάποιοι αισθητήρες μπορεί να έχουν και κάποια επίδραση στην ποσότητα που μετράνε. Για παράδειγμα, ένα θερμόμετρο θερμοκρασίας δωματίου το οποίο εισάγεται σε ένα Cεστό ποτήρι που έχει μέσα κάποιο υγρό, ψύχει το υγρό ενώ παράλληλα Cεσταίνεται από αυτό. Οι αισθητήρες πρέπει να σχεδιάζονται ώστε να έχουν μικρή επίδραση στην ποσότητα που μετρείται. Ως αποτέλεσμα, οι αισθητήρες πια σχεδιάζονται μικροσκοπικοί ώστε να έχουν και μεγαλύτερη ταχύτητα αλλά και ακρίβεια [8]. Σε αυτή την εργασία περισσότερο ενδιαφέρον έχουν οι αισθητήρες που είναι ενσωματωμένοι σε κινητά τηλέφωνα. Οι αισθητήρες αυτοί ανιχνεύουν κυρίως την κίνηση, την τοποθεσία αλλά και τον ήχο. Παρακάτω περιγράφονται λεπτομερώς μερικοί από αυτούς τους αισθητήρες.

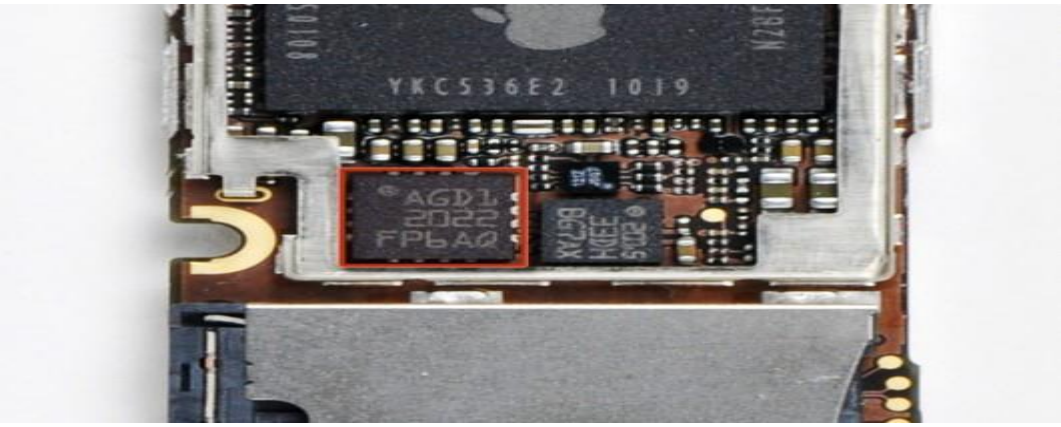
ΕΠΙΤΑΧΥΝΣΙΟΜΕΤΡΑ

Τα επιταχυνσιόμετρα είναι τυπικά ηλεκτρομηχανικά όργανα που μετράνε την επιτάχυνση που ανιχνεύουν κατά μήκος των αξόνων τους. Η επιτάχυνση που μετρείται μπορεί να είναι είτε στατική, όπως η σταθερή δύναμη της βαρύτητας, είτε δυναμική όπως προκαλείται κουνώντας το επιταχυνσιόμετρο. Ανεξάρτητα από τις διαφορές στην κατασκευή τους, η λειτουργία των επιταχυνσιομέτρων έχει να κάνει με τις διάφορες αλλαγές που υφίσταται η μάζα του συστήματος. Η επιτάχυνση είναι ανάλογη του εκτοπίσματος της μάζας όταν εφαρμόζεται κάποια δύναμη. Κατά καιρούς έχουν χρησιμοποιηθεί επιταχυνσιόμετρα που τοποθετούνται πάνω στο σώμα για την αναγνώριση της κίνησης [24].



ΓΥΡΟΣΚΟΠΙΑ

Πρόσφατα, τα κινητά τηλέφωνα έχουν εξοπλιστεί με γυροσκόπια. Τα γυροσκόπια είναι ένα είδος μη περιστρεφόμενων αισθητήρων τα οποία χρησιμοποιούν το Coriolis Effect (η απόκλιση ενός κινούμενου αντικειμένου όταν η κίνηση περιγράφεται σχετικά με ένα σημείο περιστροφής) σε μια μάζα για να ανιχνεύσουν τη γωνία περιστροφής [17]. Τα ενσωματωμένα γυροσκόπια χρησιμοποιούνται κυρίως στην αναγνώριση δραστηριοτήτων [15] αλλά και στην ανίχνευση της θέσης του σώματος [5]. αισθητήρες χαμηλής κατανάλωσης ενέργειας. Παρόλα αυτά, τα γυροσκόπια όταν χρησιμοποιούνται για προσανατολιστικούς σκοπούς είναι επιρρεπή σε λάθη [21].



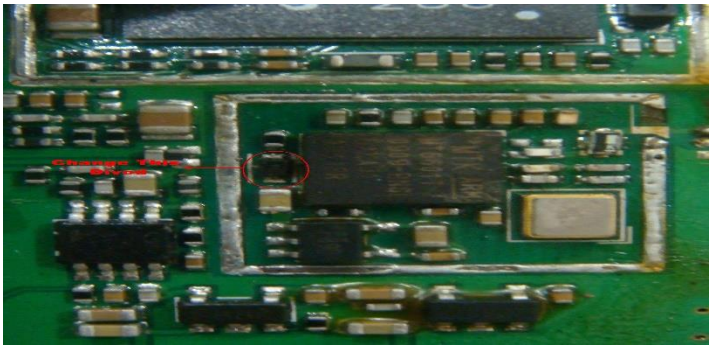
GPS

Το GPS παρέχει τη θέση του χρήστη σχεδόν παντού στη Γη. Το GPS βασίζεται σε μετρήσεις και συγκρίσεις από στιγμιαία σήματα μετάδοσης που προέρχονται από μια κινητή μονάδα. Η θέση ενός κινητού τηλεφώνου μπορεί να μετρηθεί με βάση τη χρονική καθυστέρηση του σήματος από κάθε ένα από ένα αριθμό δορυφόρων μέχρι ένα κινητό δορυφόρο. Η θέση του τηλεφώνου δίνεται σε δύο διαστάσεις όταν ο λήπτης μπορεί να δει τουλάχιστον 3 δορυφόρους. Παρόλο την υψηλή ακρίβεια του GPS όταν χρησιμοποιείται για εξωτερική τοπικότητα, θεωρείται ως μέσο υψηλής κατανάλωσης ενέργειας.

BLUETOOTH

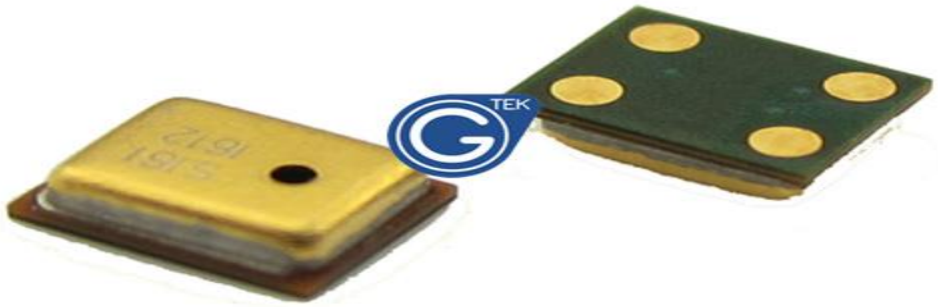
Το bluetooth είναι μια διεπαφή που σχεδιάστηκε για την υποστήριξη της ασύρματης επικοινωνίας. ΄Εχει σχεδιαστεί για επικοινωνία μεταξύ συσκευών που βρίσκονται σε μικρή ακτίνα η μία από την άλλη. Η κύρια εφαρμογή του bluetooth ως προς το πεδίο της ανίχνευσης είναι η επικοινωνία με εξωτερικούς αισθητήρες ή συσκευές. Το bluetooth επιτρέπει σε συσκευές να ανιχνεύουν άλλες συσκευές ή ακόμα και πληροφορίες για συσκευές που έχουν bluetooth και είναι σε μικρή απόσταση. Αυτές οι πληροφορίες περιλαμβάνουν τη MAC διεύθυνση της συσκευής, τον τύπο της αλλά και το όνομά της. Η MAC διεύθυνση είναι ένας αριθμός των 48 ψηφίων ο οποίος είναι μοναδικός για κάθε συσκευή. Το όνομα καθορίζεται από τον κάθε χρήστη και ο τύπος είναι 3 ακέραιοι που καθορίζουν το είδος της συσκευής (κινητό τηλέφωνο,

υπολογιστής). Η ικανότητα του bluetooth να ανιχνεύει την παρουσία άλλων συσκευών που είναι σε κοντινή απόσταση έχει χρησιμοποιηθεί σε πολλές εφαρμογές. Το μειονέκτημα του bluetooth είναι η υψηλή κατανάλωση ενέργειας όταν όταν η χρήση του είναι συνεχής.



ΜΙΚΡΟΦΩΝΟ:

Το μικρόφωνο είναι ένας ακουστικός μετατροπέας. Εκτός από τη χρήση του σε φωνητικά τηλεφωνήματα, ερευνητές το έχουν χρησιμοποιήσει για την ανάπτυξη εφαρμογών που στηρίζονται στην ανιχνευτική ικανότητα των μικροφώνων των κινητών τηλεφώνων. ΄Ενα πολύ επιτυχημένο παράδειγμα είναι αυτό των συστημάτων αναγνώρισης ομιλίας [6], τα οποία έχουν ευρέως υλοποιηθεί σε κινητά τηλέφωνα. Αυτά τα συστήματα επιτρέπουν στους χρήστες να χειρίζονται το κινητό τους τηλέφωνο μόνο με φωνητικές εντολές χωρίς τη χρήση πληκτρολογίου. Επίσης, το μικρόφωνο χρησιμοποιείται και για την ανίχνευση του ήχου στο περιβάλλον καθώς έτσι μπορεί να αναγνωριστεί το κοινωνικό περιβάλλον του χρήστη.



ΚΕΦΑΛΑΙΟ 3: ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΕΡΓΑΛΕΙΑ

Η κύρια γλώσσα προγραμματισμού που χρησιμοποιείται για την ανάπτυξη εφαρμογών Android είναι η Java, ενώ τους τελευταίους μήνες γίνονται προσπάθειες για να συμπεριληφθούν και άλλες γλώσσες όπως η C και η C++. Οπότε βασική προϋπόθεση είναι να διαθέτουμε τα αντίστοιχα εργαλεία της γλώσσας προγραμματισμού και συγκεκριμένα το Java Development Kit (JDK). Ακόμη χρειαζόμαστε ένα ολοκληρωμένο περιβάλλον ανάπτυξης (integrated development environment ή IDE) για να μεταγλωττίσουμε και να τρέξουμε τα προγράμματα μας, όπως είναι για παράδειγμα το Eclipse (Εικόνα 13). Το βασικότερο εργαλείο αποτελεί το Android SDK (software development kit) το οποίο ουσιαστικά μας παρέχει τα επιπλέον εργαλεία ώστε να μπορούμε να γράψουμε κώδικα για την κατασκευή μιας εφαρμογής Android. Η σύνδεση του Android SDK με το γραφικό μας περιβάλλον (Eclipse) γίνεται μέσω μιας επέκτασης (Android Development Tools ή ADT Plug-in) που κάνουμε εγκατάσταση στο Eclipse, ώστε να μπορέσουμε να μεταγλωττίσουμε την εφαρμογή μας και έπειτα να την τρέξουμε. Καθώς το λειτουργικό σύστημα Android κυκλοφορεί σε διάφορες εκδόσεις, όπως αναφέραμε παραπάνω, καθίσταται σαφές ότι η κάθε έκδοση θα χρησιμοποιεί και διαφορετικά προγραμματιστικά εργαλεία. Έτσι μέσα από το ADT Plug-in μπορούμε να εγκαταστήσουμε τα εργαλεία καθώς και την τεκμηρίωση (documentation) αλλά και διάφορα παραδείγματα για την υλοποίηση της εφαρμογής σε οποιαδήποτε έκδοση. Για παράδειγμα, αν θέλουμε η εφαρμογή μας να είναι συμβατή και σε παλαιότερες εκδόσεις του Android τότε θα πρέπει να εγκαταστήσουμε και τα αντίστοιχα εργαλεία και να δοκιμάσουμε την εφαρμογή μας σε αυτές. Τέλος, για να δοκιμάσουμε την εφαρμογή μας και να δούμε τα αποτελέσματα της θα χρειαστούμε κάποιον προσομοιωτή κινητού τηλεφώνου, αν δεν διαθέτουμε εμείς οι ίδιοι, στον υπολογιστή μας. Τη λύση μας δίνει μια εικονική συσκευή Android Android Virtual Device ή AVD) η οποία ουσιαστικά αποτελεί προσομοιωτή τόσο software όσο και hardware ενός κινητού τηλεφώνου με λειτουργικό σύστημα Android (Εικόνα 14). Την συσκευή αυτή την εγκαθιστάμε μέσα από το ADT Plug-in, από όπου μπορούμε να ρυθμίσουμε πολλές παραμέτρους της, όπως τι έκδοση Android θα χρησιμοποιεί, το μέγεθος της οθόνης, το μέγεθος της εξωτερικής κάρτας αποθήκευσης και της cache, καθώς και άλλα χαρακτηριστικά που έχουν να κάνουν με την τοποθεσία (GPS), την δυνατότητα λήψης φωτογραφιών, κ.α. Απαιτούνται αρκετοί υπολογιστικοί πόροι και μεγάλη έκταση μνήμης RAM για την εκτέλεση της AVD, πράγμα που καθιστά τις περισσότερες φορές αργή την εκτέλεση της εφαρμογής μας στη συσκευή αυτή. Φυσικά, κάθε φορά μπορούμε να εγκαθιστούμε και να ελέγχουμε τις εφαρμογές μας σε φυσική συσκευή, όπως ένα κινητό τηλέφωνο ή ένα tablet που χρησιμοποιούν λειτουργικό σύστημα Android, συνδέοντας το απλά σε μια θύρα USB του υπολογιστή μας. Στο σημείο αυτό, πρέπει να σημειώσουμε ότι όλα τα παραπάνω προγραμματιστικά εργαλεία διατίθενται ελεύθερα στο διαδίκτυο από τους κατασκευαστές τους τα οποία μπορούμε να κατεβάσουμε και να χρησιμοποιήσουμε χωρίς κανένα κόστος.

3.1 Android Studio

Το Android Studio είναι ένα ολοκληρωμένο προγραμματιστικό περιβάλλον (IDE) για ανάπτυξη εφαρμογών στηνAndroid πλατφόρμα. Ανακοινώθηκε στις 16 Μαΐου 2013 στο συνέδριο Google I/O από την Google Product Manager, Katherine Chou. Το Android Studio είναι διαθέσιμο ελεύθερα με την άδεια Apache License 2.0.[1]

Το Android Studio ήταν διαθέσιμο σε πρώιμο στάδιο για προεπισκόπηση ξεκινώντας από την έκδοση 0.1 τον Μάιο του 2013, έπειτα ξεκίνησε το δοκιμαστικό στάδιο από την έκδοση 0.8 που βγήκε τον Ιούνιο του 2014[2]. Η πρώτη σταθερή έκδοση βγήκε το Δεκέμβριο του 2014, με την έκδοση 1.0.[3]

Βασισμένο στο λογισμικό της JetBrains' IntelliJ IDEA, το Android Studio σχεδιάστηκε αποκλειστικά για Android προγραμματισμό[4]. Είναι διαθέσιμο για Windows, Mac OS X και Linux[2][5], και αντικατέστησε τα Eclipse Android Development Tools (ADT) ως το κύριο IDE της Google για ανάπτυξη Android εφαρμογών.

Οι λειτουργίες που παρέχει στην τωρινή του έκδοση είναι οι παρακάτω:

1. Live Layout: Απόδοση της εμφάνισης της εφαρμογής σε πραγματικό χρόνο, ανάλογα με τις αλλαγές που πραγματοποιεί ο προγραμματιστής

2. Developer Console: Μια κονσόλα προγραμματιστή που προσφέρει συμβουλές βελτιστοποίησης, βοήθεια στη μετάφραση του προγράμματος, παρακολούθηση παραπομπών, μετρήσεις χρήσης.

3. Πρόνοια για διάθεση beta εκδόσεων εφαρμογών

4. Εκτέλεση των εφαρμογών με χρήση του προτύπου Gradle

5. Επανασχεδίαση του κώδικα για καλύτερη αλληλεπίδραση με το Android

6. Εργαλεία για την αποφυγή προβλημάτων σχετικά με την χρηστικότητα, την απόδοση και την συμβατότητα των προγραμμάτων.

7. Υποστήριξη για την δημιουργία εφαρμογών Android Wear.

8. Ενσωματωμένη υποστήριξη του Google Cloud που δίνει τη δυνατότητα υποστήριξης του Google Cloud Messaging

9. Layout editor: Δίνει τη δυνατότητα στους χρήστες να χρησιμοποιούν Dragand-Drop στοιχεία καθώς και επισκόπηση της εμφάνισης της εφαρμογής σε διαφορετικές οθόνες.

10.Πρότυπα για τη δημιουργία σχεδίων και συστατικών για το Android.

11.Δυνατότητα μετακίνησης από άλλα προγράμματα όπως το Eclipse

Η πλατφόρμα είναι διαθέσιμη μέσω της ιστοσελίδας http :// developer. android. com/ sdk/ index. html και περιέχει όλα όσα χρειαζόμαστε για την δημιουργία εφαρμογών για το Android, συμπεριλαμβάνοντας το Android Studio IDE και τα εργαλεία Android Studio SDK. Αφού κατεβάσουμε και τρέξουμε το αρχείο εγκατάστασης του προγράμματος μπορούμε να ξεκινήσουμε την εγκατάσταση. Θα μας ζητηθεί να επιλέξουμε ποια χαρακτηριστικά θέλουμε να εγκατασταθούν καθώς και να επιλέξουμε τον τόπο εγκατάστασης για το πρόγραμμα. Μετά το τέλος της εγκατάστασης έχουμε την επιλογή να εισάγουμε τις ρυθμίσεις μας σε περίπτωση που ήδη χρησιμοποιούσαμε κάποια παλιότερη έκδοση του προγράμματος.

Ένα σημαντικό βήμα πριν να ξεκινήσουμε την ανάπτυξη της εφαρμογής μας είναι να ελέγξουμε ποια πακέτα του Android SDK έχουμε εγκαταστήσει έτσι ώστε να σιγουρευτούμε ότι είμαστε έτοιμοι και δεν μας λείπει κάποιο σημαντικό πακέτο. Για να το κάνουμε αυτό πρέπει να χρησιμοποιήσουμε το εργαλείο Android SDK Manager στο οποίο έχουμε πρόσβαση από την επιλογή Διαμόρφωση – SDK Manager μέσα από την οθόνη καλωσορίσματος του προγράμματος. Μάλιστα το εργαλείο έχει επιλογή να ειδοποιούμαστε για νέες ενημερώσεις των πακέτων που χρησιμοποιούμε έτσι ώστε το πρόγραμμα να συμβαδίζει πάντα με την εξέλιξη του λογισμικού του Android και να μας προσφέρει όσο το δυνατόν γρηγορότερα και πιο εύχρηστα νέες επιλογές για την ανάπτυξη των εφαρμογών μας.

3.2 Parse

Με την αυξημένη χρήση των κινητών συσκευών, είναι σύνηθες για τις εφαρμογές να προσφέρουν χαρακτηριστικά όπως η αποθήκευση αντιγράφων ασφαλείας, συγχρονισμό δεδομένων, ανταλλαγή δεδομένων κλπ .

Χτίζοντας αυτόνομες εφαρμογές που λειτουργούν μόνο και να αποθηκεύσετε τα δεδομένα τους με τη συσκευή που είναι εγκατεστημένες σε αυτό μερικές φορές δεν εφικτό. Ένα σύστημα υποστήριξης είναι συνήθως απαραίτητο σε περίπτωση που τα δεδομένα πρέπει να αποθηκευτούν και να επεξεργαστούν για την εφαρμογή για να παρέχουν την υπηρεσία για την οποία έχουν δημιουργηθεί.

Για την συγκεκριμένη εφαρμογή χρησιμοποιήθηκε η υπηρεσία Parse προκειμένου να παρέχει backend στην android εφαρμογή. Το Parse είναι ένα από τα πιο δημοφιλή Backend . Αυτή η υπηρεσία προσφέρει τρία προϊόντα σε ένα πακέτο. Parse Core,Parse Push και Parse Analytics.

Το Parse Core γενικά χειρίζεται την αποθήκευση των δεδομένων. Το Parse Push χρησιμοποιείται για να στέλνει push notifications. Δίνει την δυνατότητα στον προγραμματιστή να προσαρμόσει, να προγραμματίσει και να στείλει push notifications σε όλους τους εγγεγραμμένους χρήστες ή σε μία επιλεγμένη ομάδα χρηστών. Η Parse Analytics δίνει την δυνατότητα καταγραφής των δεδομένων της εφαρμογής. Μπορούμε να ανιχνεύσουμε τα δεδομένα χρήσης όπως πόσες εγκαταστάσεις , πόσοι ενεργοί χρήστες κτλ.

ΚΕΦΑΛΑΙΟ 4 : Ανάπτυξη της εφαρμογής αναγνώρισης της τρέχουσας κατάστασης του χρήστη (User's Activity Recognition)

4.1 Google Api Client

Ο θεμέλιος λίθος στη δημιουργία της εφαρμογής, όπως επίσης και στη δημιουργία σχεδόν κάθε σύγχρονης εφαρμογής για κινητά τύπου smartphone είναι η επικοινωνία της συσκευής με διάφορα web services τα οποία φροντίζουν να «αποκωδικοποιούν» τα δεδομένα που αποστέλλει η συσκευή και να τα επεξεργάζονται με σκοπό τη δημιουργία χρήσιμης για το χρήστη πληροφορίας.

Τα εκάστοτε web services παρουσιάζουν μια σειρά από πλεονεκτήματα και για το χρήστη αλλά και για το δημιουργό/ούς της εφαρμογής τα οποία θα εξετάσουμε ευθείς αμέσως. Με την πάροδο του χρόνου και κυρίως την εξέλιξη της τεχνολογίας οι συσκευές που κρατάνε στα χέρια τους οι άνθρωποι εσωκλείουν επεξεργαστική ισχύ ανώτερη πολλές φορές από την ειδική ή και «ωμή» ισχύ που παρείχαν επιτραπέζιοι ηλεκτρονικοί υπολογιστές πριν από μία δεκαετία.

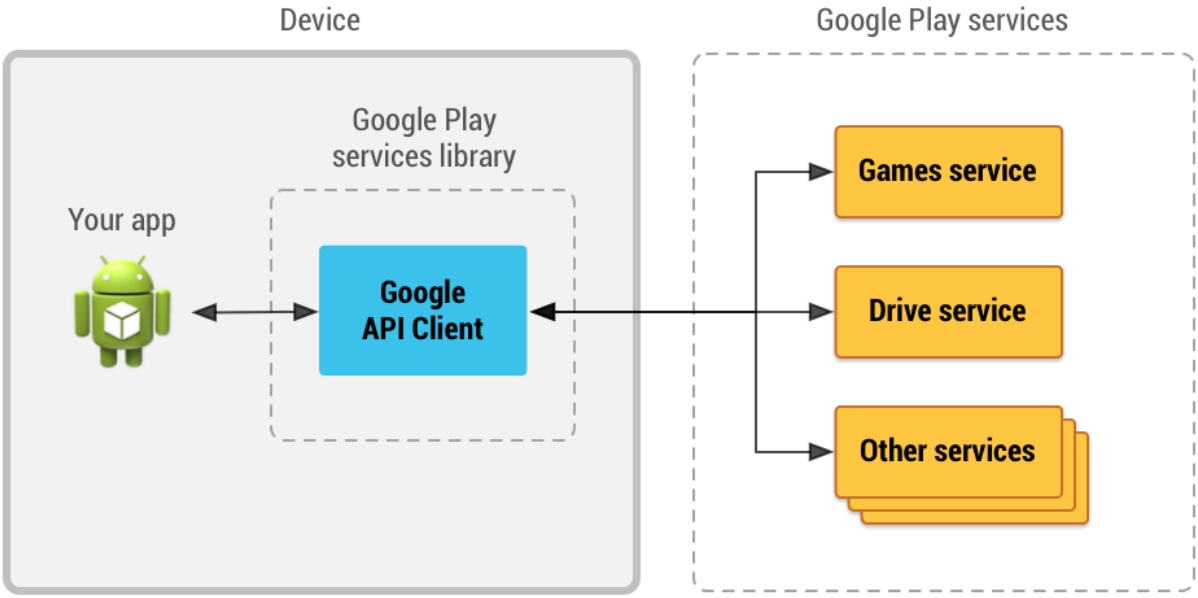
Η επεξεργαστική ισχύς που προσφέρεται πλέον άπλετα έχει όμως ένα μεγάλο αντίκτυπο στη διάρκεια ζωής της μπαταρίας του τηλεφώνου όταν αυτό τη χρησιμοποιείται μέσω εφαρμογών. Εδώ λοιπόν η εισαγωγή των διαφόρων web services έρχεται να δώσει λύση αφού η δαπανηρή σε επεξεργαστικούς πόρους και άρα σε διάρκεια ζωής της μπαταρίας επεξεργασία των δεδομένων δε λαμβάνει χώρα στη συσκευή αλλά στους web servers της υπηρεσίας που προσφέρει το web service.

Κατά δεύτερον, το ίδιο το web service είναι πολύ πιο πιθανό να αναβαθμιστεί και να αναβαθμίσει τα χαρακτηριστικά του αφού υποστηρίζεται από μία πολύ μεγαλύτερη ομάδα δημιουργών software από τις τυπικές ομάδες δημιουργών software mobile εφαρμογών. Έτσι τα δεδομένα υπόκεινται πάντα σε όλο και αναπτυσσόμενες διαδικασίες, ταχύτερες, ικανότερες να παράξουν αξιόπιστα δεδομένα και ακόμη να είναι πάντα στον παλμό της εποχής όσον αφορά τις απαιτήσεις των χρηστών.

Μια διαδικασία αλλαγής ενός τόσου σημαντικού μέρους της εφαρμογής κάθε λίγους μήνες θα είχε σοβαρό κόστος εργατικού δυναμικού και πόρων στα πλαίσια μιας επιχείρησης ή και θα καθιστούνταν αδύνατο στα πλαίσια μιας ομάδας ανεξάρτητων παραγωγών. Έτσι η ομάδα ανάπτυξης της web εφαρμογής έχει το χρόνο και τη δυναμική να εστιάσει σε πιο βασικούς τομείς για τη χρήση αυτής όπως είναι το περιβάλλον διεπαφής.

Επιπλέον, πολλές φορές δυνατότητες χρήσιμες στους προγραμματιστές όπως έτοιμες βιβλιοθήκες λογισμικού είναι διαθέσιμες δωρεάν μέσω των εκάστοτε έτοιμων Application Programming Interface (εφεξής API). Κατά αυτό τον τρόπο είναι δυνατόν ο προγραμματιστής

να βρει μια ήδη υλοποιημένη διαδικασία παρόμοια με αυτή που θέλει να παράξει, γλυτώνοντας έτσι πολύτιμο χρόνο. Ένας από τους πρωτεργάτες σε αυτή τη λύση είναι και η γνωστή εταιρεία Google η οποία προκειμένου να φέρει περισσότερο κόσμο σε επαφή με τα προϊόντα της διαθέτει δωρεάν προς το παρόν APIs τα οποία διευκολύνουν την επεξεργασία δεδομένων και τη χρήση τους από την εφαρμογή του δημιουργού.



Εικόνα: Πως λειτουργεί το Google API Client

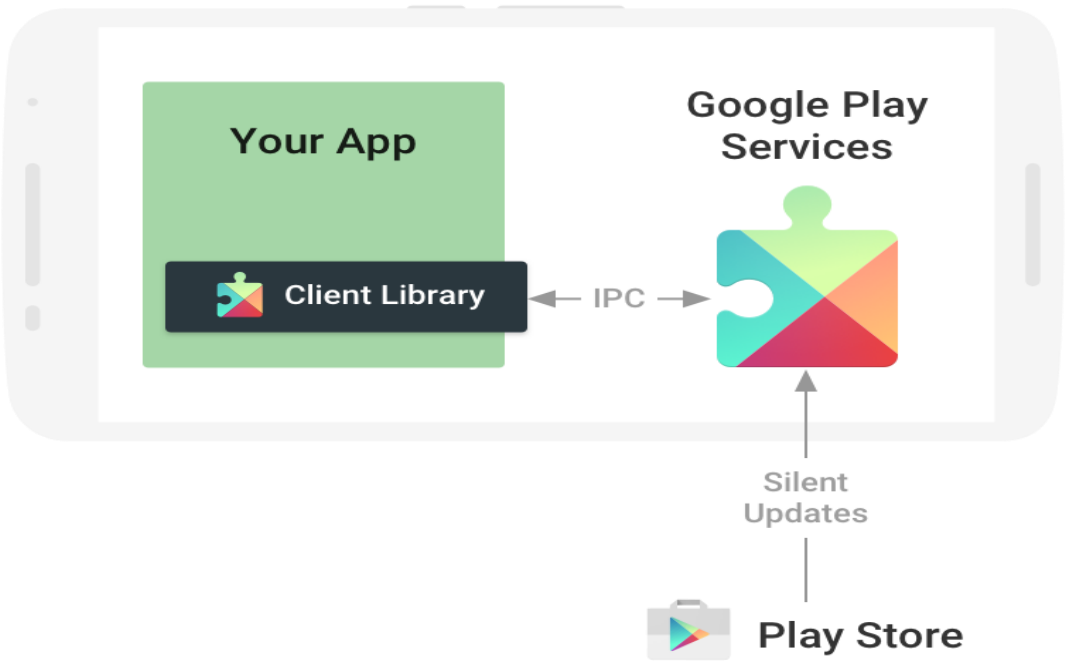
Μερικές από τις δυνατότητες που προσφέρει η βιβλιοθήκη Google API client είναι η αποστολή δεδομένων μέσω του πρωτοκόλλου HTTP, η επεξεργασία JSON, το upload ή download δεδομένων multimedia, η υποστήριξη του πρωτοκόλλου oAUTH 2.0 για την ασφαλή εγγραφή/ σύνδεση με την υπηρεσία, εξελιγμένα μοντέλα XML και JSON.

Η αποστολή δεδομένων από τη συσκευή προς τη web υπηρεσία μπορεί να γίνεται είτε σύγχρονα είτε ασύγχρονα. Αυτό σημαίνει ότι ο χρήστης θα είναι σε θέση να επιλέξει ο ίδιος πότε η εφαρμογή να αποστέλλει τα δεδομένα της είτε η αποστολή και επεξεργασία να γίνεται σε μηδενικό σχεδόν χρόνο την ώρα της επιλεγμένης δραστηριότητας.

Επίσης είναι εφικτό εφόσον το προσφέρει η εφαρμογή ο χρήστης να συνδέεται μέσω του προσωπικού του λογαριασμού Google απλοποιώντας τη διαδικασία εγγραφής ή σύνδεσης αν η υπηρεσία προσφέρει κάποιου είδους συνδρομή σε αυτή.

Ένα ακόμη πλεονέκτημα που προσφέρει ο API client των google services είναι η απλοποιημένη διαχείριση σφαλμάτων συνδέσεων δικτύου μεταξύ τηλεφώνου και υπηρεσίας ενώ και η

εξουσιοδότηση του χρήστη για τη χρήση της υπηρεσίας πραγματοποιείται μόνο μια φορά κατά την εγκατάσταση της εφαρμογής όπου και διερωτάται αν εξουσιοδοτεί την εφαρμογή να χρησιμοποιήσει τυχόν ευαίσθητα δεδομένα όπως π.χ. είναι το ονοματεπώνυμο του χρήστη ή η εξουσιοδότηση να λαμβάνει δεδομένα από το υλικό της συσκευής όπως πχ είναι ο πομπός του Global Positioning System (εφεξής GPS).



Εικόνα: «Κρυφή» ενημέρωση των Google services.

Τέλος είναι επιπλέον δυνατόν οι δυνατότητες που παρέχονται μέσω του API να επεκτείνονται, να απλοποιούνται οι διαδικασίες ή να επιταχύνονται και όλο αυτό χωρίς να χρειάζεται ο δημιουργός της εφαρμογής να καταβάλει κάποια προσπάθεια αφού το μόνο που χρειάζεται είναι η «κρυφή» αναβάθμιση των google services δια μέσω του Google Play Store, της ηλεκτρονικής μορφής καταστήματος εφαρμογών της Google. Μια καλή ιδεά για αυτό μας δίνει η παραπανω εικόνα.

Από τη βιβλιοθήκη του Google API client εμείς χρησιμοποιήσαμε κατά κύριο λόγο το Recognition Activity καθώς και το Detected Activity που είναι υλοποιημένα ως java classes.

4.2 Περιγραφή της λειτουργίας ανίχνευσης κίνησης(Track Movement)

4.2.1 Γενική περιγραφή της εγαρμογης (Track Movement)

Η εφαρμογή που υλοποιήθηκε έχει την δυνατότητα να καταγράφει την τρέχουσα κίνηση του χρήστη. Τα επίπεδα κίνησης που καταγράφονται είναι πέντε.

1. Στασιμότητα

2. Περπάτημα

3. Τρέξιμο

4. Ποδήλατο

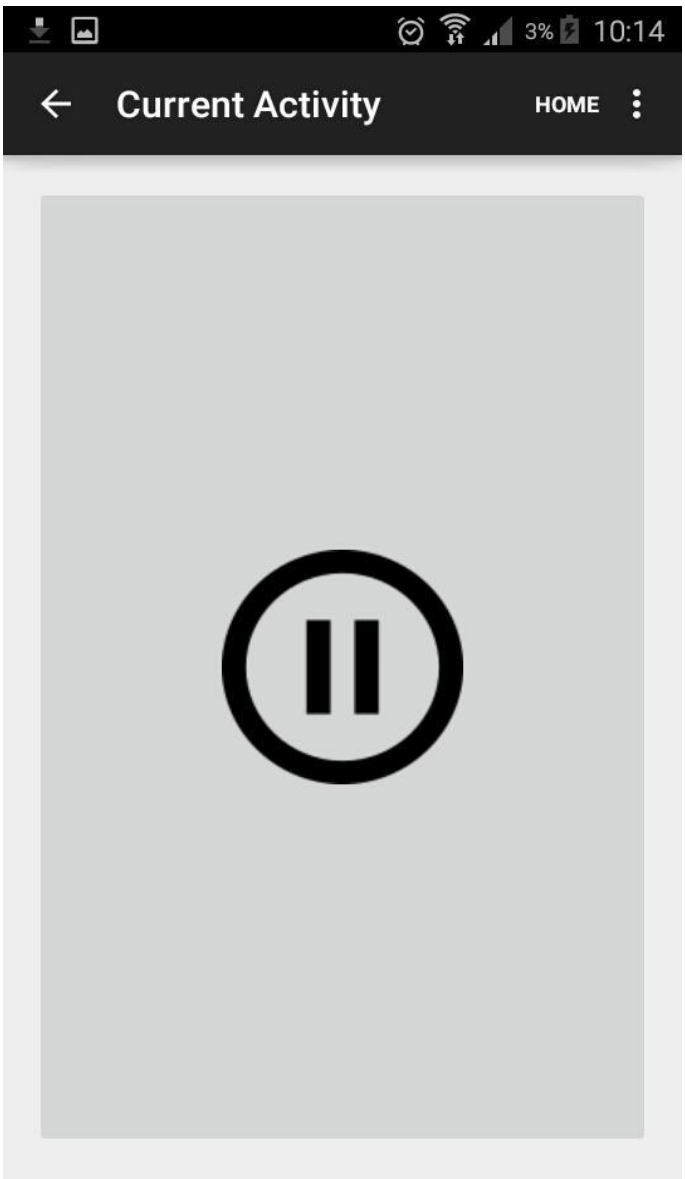
5. Αυτοκίνητο

Ο χρήστης της εφαρμογής μπορεί να έχει την εικόνα της ημερήσιας δραστηριότητας του έχοντας της εφαρμογή να τρέχει στο background ενώ παράλληλα τα δεδομένα όλων των χρηστών συλλέγονται προκειμένου να έχουμε μία γενικότερη εικόνα των δραστηριοτήτων των χρηστών

και να μπορούμε να εξάγουμε κάποια στατιστικά δεδομένα. Πιο ειδική περιγραφή της διαδικασίας που πραγματοποιούνται όλα αυτά γίνεται παρακάτω.

4.2.2 Οδηγίες χρήσης της εφαρμογής.

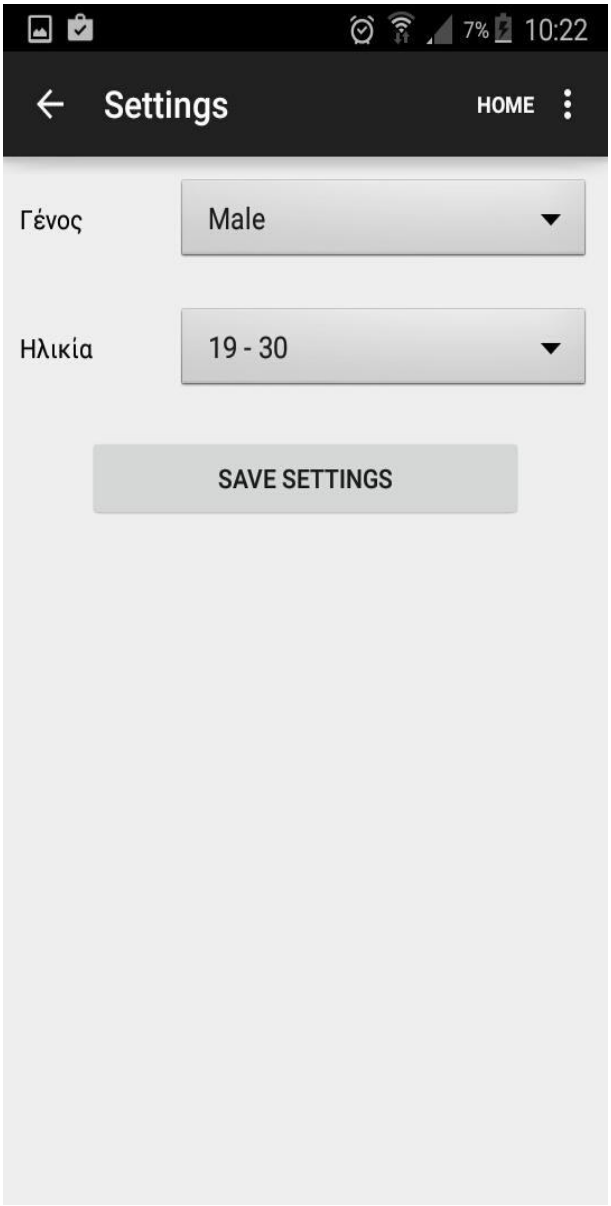
Με το που ξεκινάει η εφαρμογή ο χρήστης βλέπει την οθόνη εισόδου.



Εικόνα: οθόνη εισόδου

Πριν ξεκινήσει την καταγραφή στα settings που μπορεί να μεταβεί είτε πατώντας πάνω στο Home είτε πατώντας πάνω στο βελάκι θα κάνει τις κατάλληλες ρυθμίσεις. Η ρυθμίσεις αυτές είναι χρήσιμες για το share των δεδομένων προκειμένου να έχουμε κάποια στατιστικά αργότερα.

Πατώντας πάνω δεξιά μας εμφανίζονται οι ρυθμίσεις που πατάμε πάνω σε αυτό και μεταβαίνουμε στην οθόνη με τα Settings. Επιλέγουμε το γένος και την ηλικία μας και πατάμε αποθήκευση.

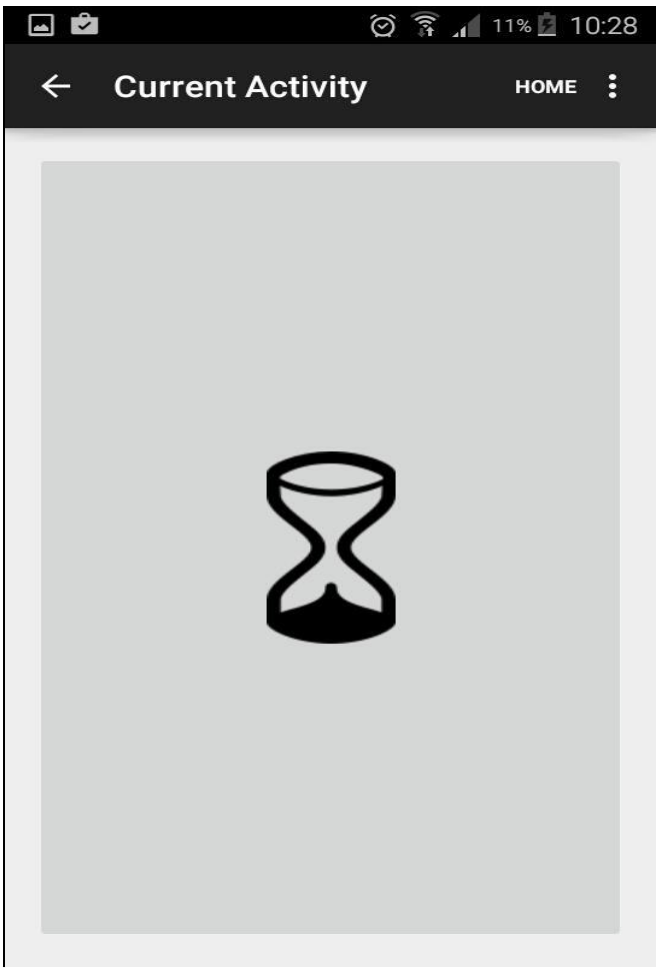


Εικόνα: οθόνη για τα settings

Η λειτουργία της εφαρμογής ξεκινάει με το που επιστρέψουμε στην αρχική οθόνη και πατήσουμε το Pause. Η χρήση της εφαρμογής απαιτεί την χρήση GPS.

Αν δεν το εχουμε ενεργοποιημένο μας πηγαίνει απευθείας στις ρυθμίσεις του κινητού για να το ενεργοποιήσουμε.

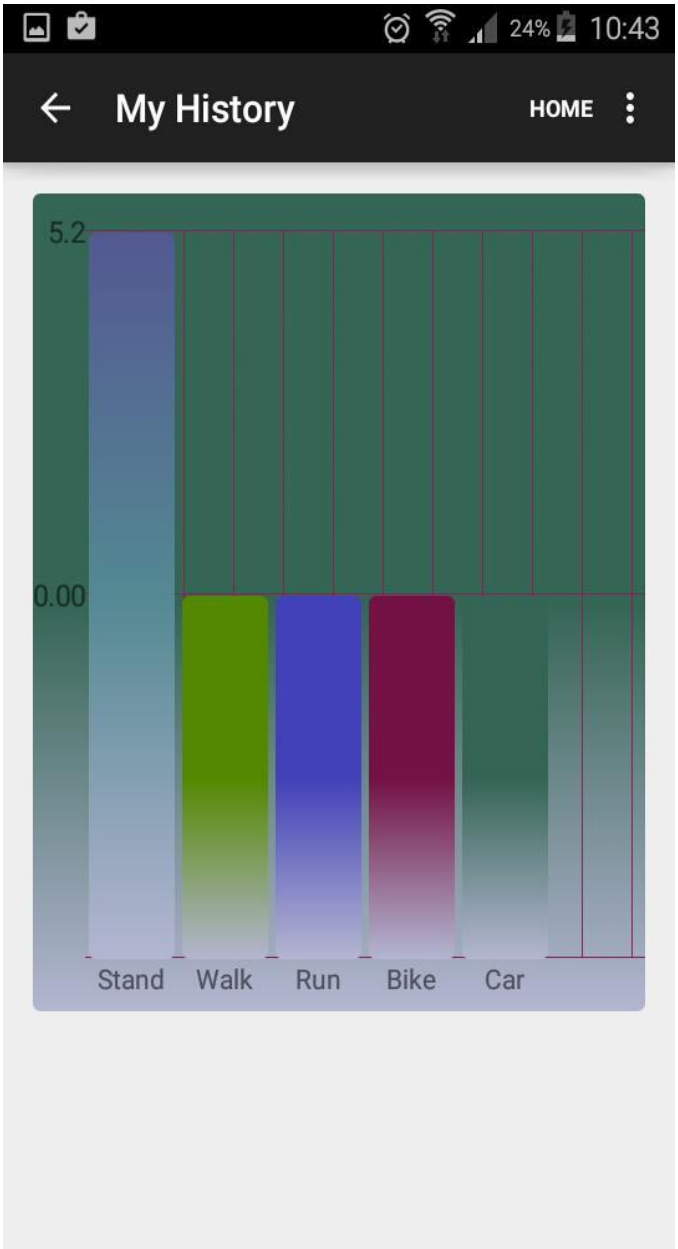
Με το που πατήσουμε το pause ξεκινάει και η εφαρμογή. Εμφανίζεται αρχικά μία κλεψύδρα μέχρι να γίνει η αναγνώριση της δραστηριότας και να μας εμαφανίσει το σχετικό εικονίδιο που υποδηλώνει την κατάσταση του χρήστη. Ενώ κάτω αριστερά εμφανιζεται και η διάρκεια της εκάστοτε κίνησης.



Εικόνα: ένδειξης ακινησίας του χρήστη για 2,2 λεπτά.

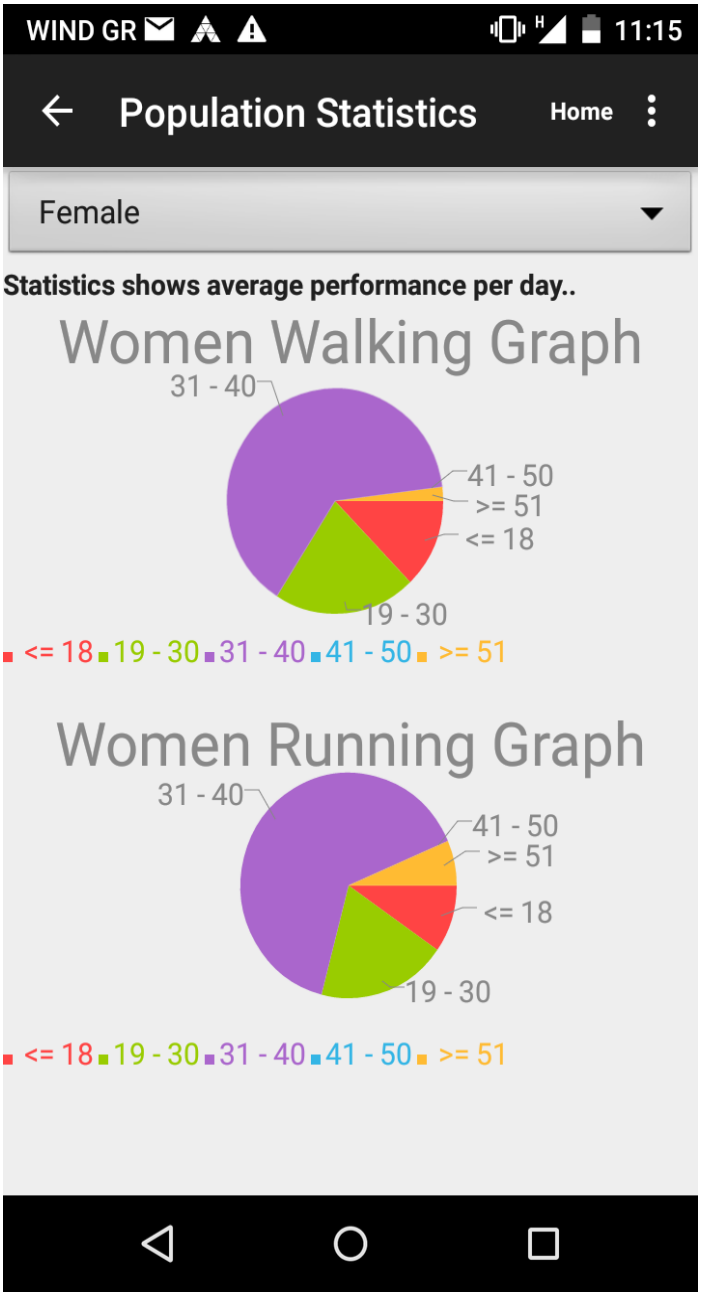
Στην συνέχεια ο χρήστης μέσα από το ειδικό αναδυόμενο μενού και να επιλέξει το My History, όπου μπορεί να δει μέσα από ενα bar chart τους χρόνους που εχει καταγράψει η εφαρμογή για τις δραστηριότητες του μέχρι ώρας.


24% 10:43
Track Movement
Current Activity
My History
Population Statistics
Settings
Menu



Εικόνα:Ατομικά στατιστικά του χρήστη.

Τέλος μέσα από την επιλογή Population Statistics ο χρήστης έχει την δυνατότητα να δει κάποια γενικότερα στατιστικά των χρηστών της εφαρμογής ανά ηλικιακό εύρος και γένος.



Εικόνα: Στατιστικά πληθυσμού

4.2.3 Υλοποίηση της εφαρμογής

Για την υλοποίηση της συγκεκριμένης εφαρμογής, χρησιμοποιήθηκε ο Google Api Client σε περιβάλλον Android. Ο βασικός άξονας μεθόδων, ο οποίος λειτούργησε σε αντιστοιχία με τον Google Api Client είναι ο παρακάτω:

```
new GoogleApiClient.Builder(context)
.addApi(ActivityRecognition.API)        .addConnectionCallbacks(this)
.addOnConnectionFailedListener(this)

.build()
```

Ο παραπάνω κώδικας κάνει αρχικοποίηση του Google Api Client για να χρησιμοποιηθεί το Activity Recognition Api. Το τελευταίο προβλέπει τις κινήσεις του χρήστη μέσα από τους διάφορους αισθητήρες της συσκευής. Μέσω της ConnectionCallbacks όταν πραγματοποιηθεί η σύνδεση με τον Google Api Client γίνεται εφικτό να κληθεί η RequestActivityUpdates. Η RequestAcivityUpdates επιστρέφει τις ενημερώσεις αναφορικά με την κάθε είδους κίνηση. Κατ' αντιστοιχία, η OnConnectionFailedListener χρησιμοποιείται όταν η σύνδεση με τον Google Api Client έχει αποτύχει.

Η RequestActivityUpdates είναι η βασική μέθοδος βάσει της οποίας λαμβάνονται οι ενημερώσεις δραστηριοτήτων του χρήστη.Οι δραστηριότητες ανιχνεύονται με την περιοδική αφύπνιση της εφαρμογής και με την ανάγνωση δεδομένων του σένσορα. Χρησιμοποιούνται μόνο σένσορες χαμηλής ενεργειακής κατανάλωσης. Ένας βασικός κανόνας χρήσης είναι ότι η εφαρμογή πρέπει να έχει τη δυνατότητα καταγραφής των δραστηριοτήτων στο παρασκήνιο και να δραστηριοποείται μόνο όταν μια συγκεκριμένη δραστηριότητα ανιχνεύεται.

Για να πετύχει η συγκεκριμένη διαδικασία έχουμε ορίσει ένα IntentService με όνομα «TrackIntentService» στο οποίο λαμβάνουμε τις ενημερώσεις για τις καταγραφές που μας στέλνει το GoogleApiClient έτσι ώστε να τις κρατήσουμε στην εφαρμογή και να την ενημερώσουμε με updates της κίνησης του χρήστη.

Η βασική μέθοδος καταγραφής των πέντε σταδίων κίνησης που αναφέρθηκαν παραπάνω, είναι η DetectedActivity. Οι παρακάτω σταθερές χρησιμοποιούνται για να δηλώσουν την κάθε κίνηση:

1. IN_VEHICLE
2. ON_BICYCLE
3. ON_FOOT
4. RUNNING
5. STILL
6. TILTING

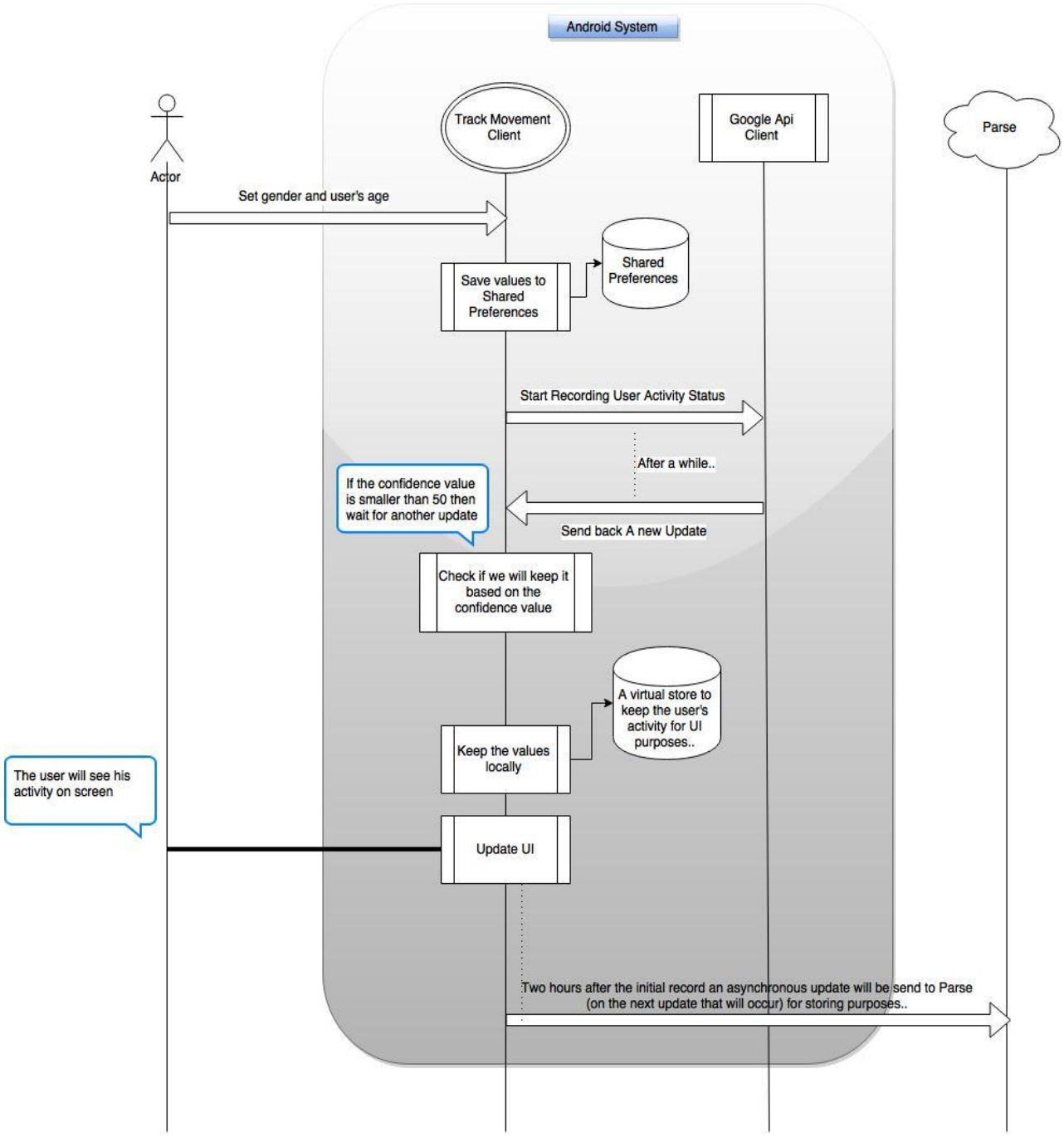
7. UNKNOWN
8. WALKING

Οι καταστάσεις STILL και TILTING είναι πανομοιότυπες στην εφαρμογή, και βάσει της λογικής που έχει υλοποιηθεί η συγκεκριμένη εφαρμογή, υποδηλώνουν την ίδια κατάσταση που είναι η στασιμότητα.

Ένα βασικό στοιχείο της DetectedActivity είναι ότι χρησιμοποιεί πιθανότητες για να καθορίσει αν όντως γίνεται κάποια συγκεκριμένη κίνηση (confidence factor). Η εφαρμογή λαμβάνει σαν σταθερά ότι για να θεωρηθεί έγκυρο ένα επίπεδο κίνησης πρέπει το ποσοστό εμπιστευτικότητας να είναι μεγαλύτερο του 50%. Ένα τέτοιο ποσοστό θεωρούμε πως είναι αρκετό για να κρατήσουμε γενικά στατιστικά της κίνησης του χρήστη.

Διαγράμματα λειτουργίας εφαρμογής

Διάγραμμα καταγραφής κίνησης χρήστη



Τα δεδομένα στην Parse αποθηκεύονται σε έναν εσωτερικό δικό τους εικονικό πίνακα. Ο πίνακας αυτός περιέχει όλες τις πληροφορίες που περνάμε στο Parse και αναφέρονται στην ημερομηνία αναφοράς των τιμών της δραστηριότητας του χρήστη καθώς και των ίδιων των τιμών αναφοράς.

objectId String	devide_id String	createdAt Date	updatedAt Date
ij7HhPCV2W	6bc4d8e3fb665a24	Jul 11, 2015, 07:21	Jul 11, 2015, 07:23
NRzqVe9r1H	6bc4d8e3fb665a24	Jul 10, 2015, 17:01	Jul 13, 2015, 18:40
CLTqXRdyRQ	720c13917d3aea43	Jul 08, 2015, 19:27	Jul 08, 2015, 20:39
BwoQHfSkJL	720c13917d3aea43	Jul 07, 2015, 20:50	Jul 07, 2015, 20:51
wH4ljRduXJ	6bc4d8e3fb665a24	Jul 06, 2015, 21:51	Jul 06, 2015, 21:58
Fc1Pn0lZCn	6bc4d8e3fb665a24	Jun 18, 2015, 05:54	Jul 06, 2015, 19:09

motionless Number	walk Number	run Number	bike Number	car Number	age_range String	gender String	mydate Date
1596.38	0	0	0	0	19 - 30	Male	Jul 10, 2015, 2…
7774.43	25.08	0	0	0	31 - 40	Male	Jul 09, 2015, 2…
40923.86	0	0	0	0	<= 18	Female	Jul 07, 2015, 2…
1949.69	0	0	0	0	19 - 30	Female	Jul 06, 2015, 2…
0.05	0.15	0	0	0	31 - 40	Male	Jul 05, 2015, 2…
0.17	0.08	0.5	0.4	0	19 - 30	Male	Jun 17, 2015, 2…

Ο πίνακας περιέχει τις ακόλουθες στήλες:

1. ObjectId: System πεδίο του Parse.
2. Device_Id: Το unique key της συγκεκριμένης Android Συσκευής
3. CreatedAt: Ημερομηνία δημιουργίας εγγραφής. Εισάγεται αυτόματα από το Parse.
4. UpdatedAt: Ημερομηνία ενημέρωσης εγγραφής. Εισάγεται αυτόματα από το Parse.
5. Motionles: Ο χρόνος παραμονής σε ακινησία σε λεπτά.
6. Walk: Ο χρόνος βάσισης επίσης σε λεπτά.
7. Run: Ο χρόνος τρεξίματος σε λεπτά.
8. Bike: Ο χρόνος ποδηλάτου σε λεπτά.
9. Car: Ο χρόνος κίνησης αυτοκινήτου σε λεπτά.
10. Age- Range: Εύρος ηλικίας του χρήστη, είναι πεδίο string.
11. Gender: Το γένος, πεδίο String.

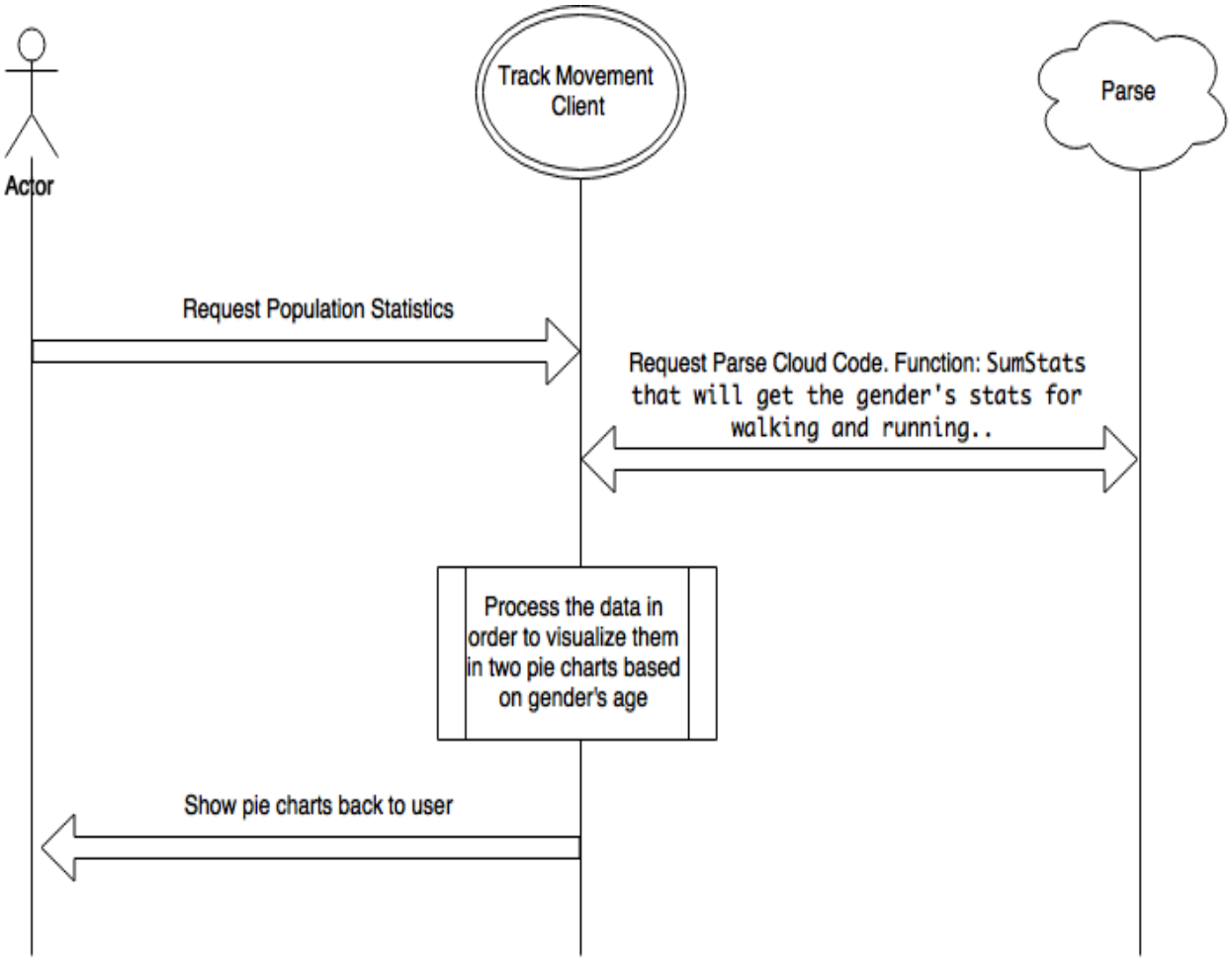
Το παραπάνω use case διάγραμμα περιγράφει τον τρόπο με τον οποίο γίνεται η καταγραφή της κίνησης του χρήστη. Ο χρήστης θέτει την ηλικία και το φύλο του και ξεκινά η διαδικασία καταγραφής της κίνησης. Ο υπεύθυνος για την καταγραφή της κίνησης client, αφενός αποθηκεύει τα δεδομένα τοπικά έτσι ώστε να είναι διαθέσιμα στο χρήστη και αφετέρου τα χρησιμοποιεί για κοινή χρήση (Share των δεδομένων). Σε συνεργασία με το Google Api γίνεται συχνή ενημέρωση των δεδομένων κίνησης. Εδώ, εισάγεται και ο παράγοντας εμπιστοσύνης του συστήματος. Αν ο παράγοντας αυτός είναι μικρότερος του 50, δεν γίνεται άμεση αποθήκευση των δεδομένων απλά το σύστημα αναμένει την επομένη ενημέρωση. Εφόσον ικανοποιείται ο παράγοντας αυτός, το UI της εφαρμογής ενημερώνεται με τα πιο πρόσφατα δεδομένα. Ο χρήστης είναι πλέον σε θέση να δει τα δεδομένα που αφορούν στην κίνησή του. Η εφαρμογή ασύγχρονα αποστέλλει σε διάστημα 5 λεπτών τα δεδομένα στον Parse για αποθήκευση.

Ένα παράδειγμα τιμών που αποθηκεύονται στον πίνακα είναι το παρακάτω (γραμμένο σε json data object).


```
{ "results": [
        {
    "age_range": "19 - 30",
    "bike": 0.4,
    "car": 0,
    "createdAt": "2015-06-18T05:54:48.751Z",
    "devide_id": "6bc4d8e3fb665a24",
    "gender": "Male",
    "motionless": 0.17,
    "mydate": {
      "__type": "Date",
      "iso": "2015-06-17T21:00:00.000Z"
    },
    "objectId": "Fc1Pn0lZCn",
    "run": 0.5,
    "updatedAt": "2015-07-06T19:09:51.349Z",
```

```
    "walk": 0.08
  }] }
```

Διάγραμμα εμφάνισης στατιστικών πληθυσμού.



Το δεύτερο use-case διάγραμμα παρουσιάζει τη δυνατότητα που έχει ο χρήστης της εφαρμογής να βλέπει στατιστικά κίνησης πληθυσμού. Ο χρήστης ζητάει από την εφαρμογή να του

παρουσιάσει τα συγκεκριμένα στατιστικά στοιχεία. Ο client καταγραφής της κίνησης αλληλεπιδρά με τον Parse προκειμένου να ικανοποιήσει το αίτημα του χρήστη. Ο Parse, ο οποίος είναι cloud-based, με τη βοήθεια της συνάρτησης SumStats εντοπίζει τα στοιχεία και τα επιστρέφει στον client, ο οποίος με τη σειρά του είναι υπεύθυνος για την παρουσίασή τους στο χρήστη. Η παρουσίαση γίνεται πλέον με φιλικό προς το χρήστη τρόπο με τη μορφή διαγραμμάτων (pie charts).

Ο λόγος που χρησιμοποιήθηκε η Parse είναι γιατί προσφέρει άμεσα χωρίς τις ανάγκες στησίματος servers και custom web services, τη δυνατότητα να έχουμε άμεσα έναν χώρο αποθήκευσης και έναν εύκολο τρόπο για ερωτήματα προς τον χώρο αυτόν.

Η Parse μέσω του Cloud Code μας δίνει τη δυνατότητα να γράψουμε javascript κώδικα για να δημιουργήσουμε functions οι οποίες θα πραγματοποιούν διάφορους υπολογισμούς για εμάς. Για τις ανάγκες της συγκεκριμένης εφαρμογής φτιάξαμε τη συνάρτηση SumStats στην οποία υπολογίζουμε ανά τύπο δραστηριότητας (Walking + Running), στατιστικά πόσο αθλούνται τα ηλικιακά εύρη που χρησιμοποιούν την εφαρμογή. Ο λόγος που χρησιμοποιήθηκαν μόνο τα συγκεκριμένα δύο activities, είναι γιατί παρουσιάζει μεγαλύτερο ενδιαφέρον και αποτελούν από τα συνηθέστερα φαινόμενα δραστηριότητας.

Η SumStats συνάρτηση λαμβάνει όλα τα δεδομένα των τελευταίων 30 ημερών που έχουν περαστεί στο Parse μέσα από τις καταγραφές που πέφτουν σε κάθε ενημέρωση που πραγματοποιείτε ανά πέντε λεπτά. Τα δεδομένα που λαμβάνει ομαδοποιούνται σε μεταβλητές ανά εύρος ηλικίας.

Συνολικά έχουμε τα ακόλουθα ηλικιακά εύρη όπως ήδη αναφέραμε σε προηγούμενο κεφάλαιο.

1. <= 18
2. 19 – 30
3. 31 – 40
4. 41 – 50
5. >= 51

Ανά ηλικιακό εύρος κρατάμε το σύνολο του χρόνου που δαπανά ο κάθε χρήστης κρατώντας παράλληλα έναν counter για τους χρόνους που συνολικά καταγράφουμε, έτσι ώστε να υπολογίσουμε τελικώς το μέσο όρο που δαπανά ένας χρήστης ανά ημέρα και ανά εύρος ηλικίας. Με αυτόν τον τρόπο στο τέλος καταλήγουμε να έχουμε για κάθε ηλικιακό εύρος το average του χρόνου για τον μήνα μας.

Αθροίζοντας τα averages σε μία μεταβλητή που υπολογίσαμε και υπολογίζοντας στατιστικά πόσο ποσοστιαία έχει κάθε ηλικιακό εύρος δαπανήσει σε χρόνο βάσει του συνόλου, κρατάμε το ποσοστό της πίτας που θα δείξουμε στον χρήστη στο τέλος.

Την παραπάνω διαδικασία την τρέχουμε ξεχωριστά για το Running Activity και το Walking Activity έτσι ώστε να επιστρέψουμε στον Client και τα δύο στατιστικά ανά ηλικιακό εύρος και να τα εμφανίσουμε σε δύο πίτες.

Ο κώδικας της SumStats παρουσιάζεται παρακάτω και αποτελεί όπως αναφέραμε κομμάτι του main.js του Cloud Code της Parse.

main.js

```
// Use Parse.Cloud.define to define as many cloud functions as you want.
// For example:
Parse.Cloud.define("hello", function(request, response) {
  response.success("Hello world!");
});


// Use Parse.Cloud.define to define as many cloud functions as you want.
// For example:
Parse.Cloud.define("SumStats", function(request, response) {

    var today = new Date();
    var dd = today.getDate();
    var mm = today.getMonth()+1; //January is 0!

    var yyyy = today.getFullYear();
    if(dd<10){
        dd='0'+dd
    }
    if(mm<10){
        mm='0'+mm
    }
    var today = yyyy + '-' + mm + '-' + dd;

    var lastMonth = new Date();
    dd = lastMonth.getDate();
    mm = lastMonth.getMonth(); //January is 0!
    yyyy = lastMonth.getFullYear();
    if (mm == 0) { mm = 12; yyyy--; }

    if(dd<10){
        dd='0'+dd
    }
    if(mm<10){
        mm='0'+mm
    }
    var lastMonth = yyyy + '-' + mm + '-' + dd;

    //response.success(request.params.gender + " " + lastMonth + " " + today);
    //return;

  var query = new Parse.Query("TrackUser");
  query.equalTo("gender", request.params.gender);
  query.greaterThanOrEqualTo("mydate", new Date(lastMonth));
  query.lessThanOrEqualTo("mydate", new Date(today));

  query.find({
    success: function(results) {
      var sumWalk18 = 0;
      var sumRun18 = 0;
      var sumWalk1930 = 0;
      var sumRun1930 = 0;
      var sumWalk3140 = 0;
```

```
54          var sumRun3140 = 0;
55          var sumWalk4150 = 0;
56          var sumRun4150 = 0;
57          var sumWalk51 = 0;
58          var sumRun51 = 0;
59          var counter18 = 0;
60          var counter1930 = 0;
61          var counter3140 = 0;
62          var counter4150 = 0;
63          var counter51 = 0;
64          for (var i = 0; i < results.length; i++)
65          {
66            if (results[i].get("age_range") == " <= 18")
67            {
68                var tmp = results[i].get("walk");
69                sumWalk18 += (tmp == "") ? 0 : tmp;
70                tmp = results[i].get("run");
71                sumRun18 += (tmp == "") ? 0 : tmp;
72                counter18++;
73            }
74            else if (results[i].get("age_range") == "19 - 30")
75            {
76                var tmp = results[i].get("walk");
77                sumWalk1930 += (tmp == "") ? 0 : tmp;
78                tmp = results[i].get("run");
79                sumRun1930 += (tmp == "") ? 0 : tmp;
80                counter1930++;
81            }
82            else if (results[i].get("age_range") == "31 - 40")
83            {
84                var tmp = results[i].get("walk");
85                sumWalk3140 += (tmp == "") ? 0 : tmp;
86                tmp = results[i].get("run");
87                sumRun3140 += (tmp == "") ? 0 : tmp;
88                counter3140++;
89            }
90            else if (results[i].get("age_range") == "41 - 50")
91            {
92                var tmp = results[i].get("walk");
93                sumWalk4150 += (tmp == "") ? 0 : tmp;
94                tmp = results[i].get("run");
95                sumRun4150 += (tmp == "") ? 0 : tmp;
96                counter4150++;
97            }
98            else if (results[i].get("age_range") == " >= 51")
99            {
100               var tmp = results[i].get("walk");
101               sumWalk51 += (tmp == "") ? 0 : tmp;
102               tmp = results[i].get("run");
103               sumRun51 += (tmp == "") ? 0 : tmp;
104               counter51++;
105           }
106         }
107
108         var avgWalk18 = counter18 == 0 ? 0 : (sumWalk18 / counter18);
109         var avgWalk1930 = counter1930 == 0 ? 0 : (sumWalk1930 / counter1930);
            var avgWalk3140 = counter3140 == 0 ? 0 : (sumWalk3140 / counter3140);
```

```
110         var avgWalk4150 = counter4150 == 0 ? 0 : (sumWalk4150 / counter4150);
111         var avgWalk51 = counter51 == 0 ? 0 : (sumWalk51 / counter51);
112
113         var avgRun18 = counter18 == 0 ? 0 : (sumRun18 / counter18);
114         var avgRun1930 = counter1930 == 0 ? 0 : (sumRun1930 / counter1930);
115         var avgRun3140 = counter3140 == 0 ? 0 : (sumRun3140 / counter3140);
116         var avgRun4150 = counter4150 == 0 ? 0 : (sumRun4150 / counter4150);
117         var avgRun51 = counter51 == 0 ? 0 : (sumRun51 / counter51);
118
119         var sumAllAvgWalk = avgWalk18 + avgWalk1930 + avgWalk3140 + avgWalk4150 + avgWalk5
120         var sumAllAvgRun = avgRun18 + avgRun1930 + avgRun3140 + avgRun4150 + avgRun51;
121
122         response.success(new Array(["18", [sumAllAvgWalk == 0 ? 0 : ((avgWalk18 / sumAllAv
123 [sumAllAvgWalk == 0 ? 0 : ((avgWalk1930 / sumAllAvgWalk) * 100), sumAllAvgRun == 0 ? 0 :
124 sumAllAvgWalk) * 100), sumAllAvgRun == 0 ? 0 : ((avgRun3140 / sumAllAvgRun) * 100)] ,["4
125 ? 0 : ((avgRun4150 / sumAllAvgRun) * 100)]],["51", [sumAllAvgWalk == 0 ? 0 : ((avgWalk51
126       },
127       error: function(e) {
128         response.error("too bad.. failed" + e.message);
129       }
      });
    });
```

Ο βασικός κώδικας της εφαρμογής

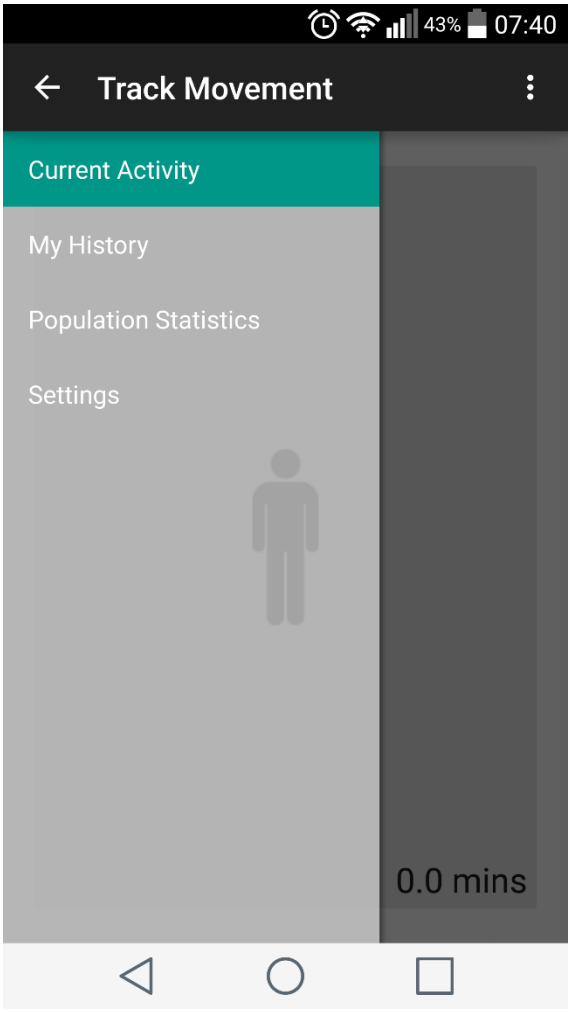
Ο Βασικός κώδικας της εφαρμογής τρέχει μέσα από την κλάση: «TrackerActivity». Όπως κάθε android εφαρμογή έτσι και η «TrackMovement» έχει στο Activity της τη μέθοδο onCreate όπου πραγματοποιούνται οι αρχικοποιήσεις του User Interface της εφαρμογής. Η εφαρμογή μας κληρονομεί από τη «ActionBarActivity» και υλοποιεί τα «NavigationDrawerCallbacks» για να μπορέσει να δημιουργήσει το αριστερό panel που εμφανίζεται πατώντας πάνω αριστερά στην εφαρμογή.

Τα Navigation Fragments που έχουμε δημιουργήσει φορτώνουν τις ακόλουθες σελίδες (Fragments):

1. Current Activity
2. My History
3. Population Statistics
4. Settings

Η αρχικοποίηση αυτών των σελίδων γίνεται μέσα από την NavigationDrawmentFragment, στην οποία υπάρχουν όλα τα events για το αναδυόμενο παράθυρο, τους τίτλους των σελίδων κ.λ.π.. Μόλις η PlaceholderFragment αρχικοποιηθεί θα οριστεί σε αυτή τα views και τα δεδομένα που θα περιέχει η κάθε σελίδα. Επίσης κατά την επιλογή της εκάστοτε σελίδας θα έχουμε τη δυνατότητα να κάνουμε διάφορες κλήσεις για τη λήψη/ενημέρωση δεδομένων για την ορθή

εμφάνισή τους στο χρήστη. Το βασικό κομμάτι κώδικα που δημιουργεί αυτές τις αρχικοποιήσεις βρίσκεται στην function «onCreateView» που είναι υπεύθυνη για το structure της κάθε σελίδας.



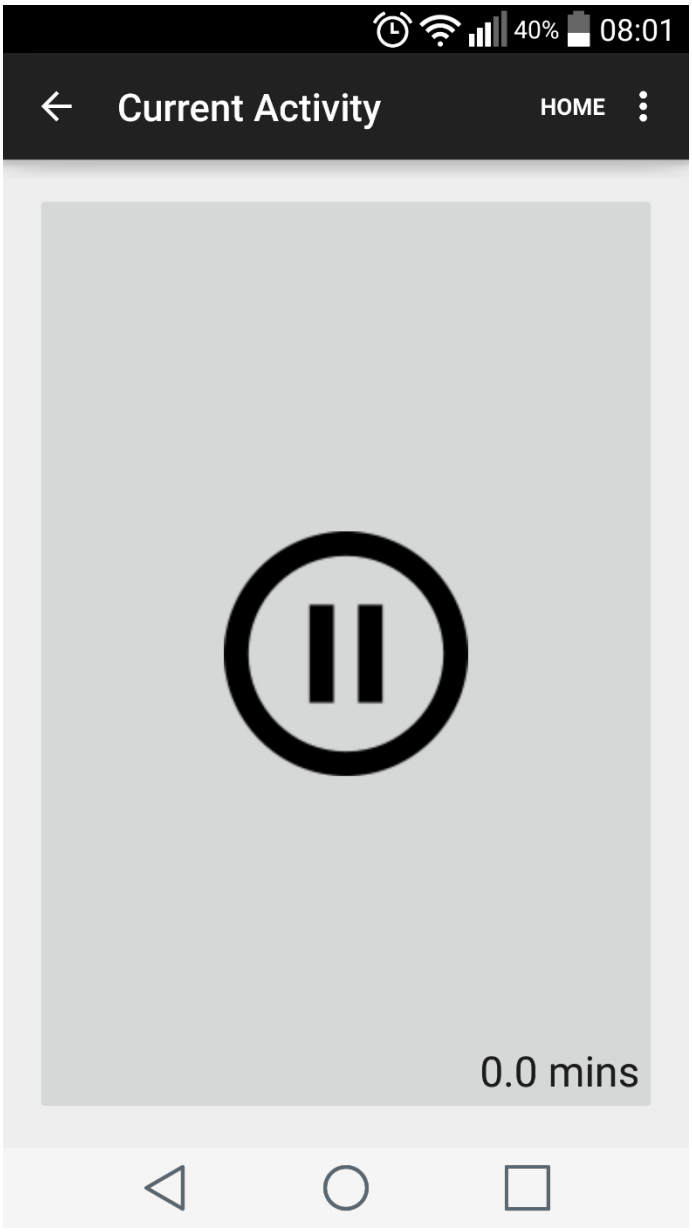
Εικόνα: Εμφάνιση μενού εφαρμογής

Στο κεντρικό μας Activity λοιπόν αρχικοποιούνται τα Navigation Fragments όπου θα στηρίξουν τις διαφορετικές «σελίδες» της εφαρμογής.

Στη συνέχεια κρατάμε στην προσωρινή μνήμη της συσκευής μέσω μίας static μεταβλητής το Android Device Id για να το χρησιμοποιήσουμε μεταγενέστερα ως αναγνωριστικό της συσκευής στην Parse. Εφόσον έχουμε κρατήσει το Id της συσκευής, ξεκινάμε και την αρχικοποίηση της βιβλιοθήκης της Parse με το κλειδί που μας έχει δοθεί από το Site το οποίο είναι σχετιζόμενο με το account που έχουμε δημιουργήσει στην Parse.

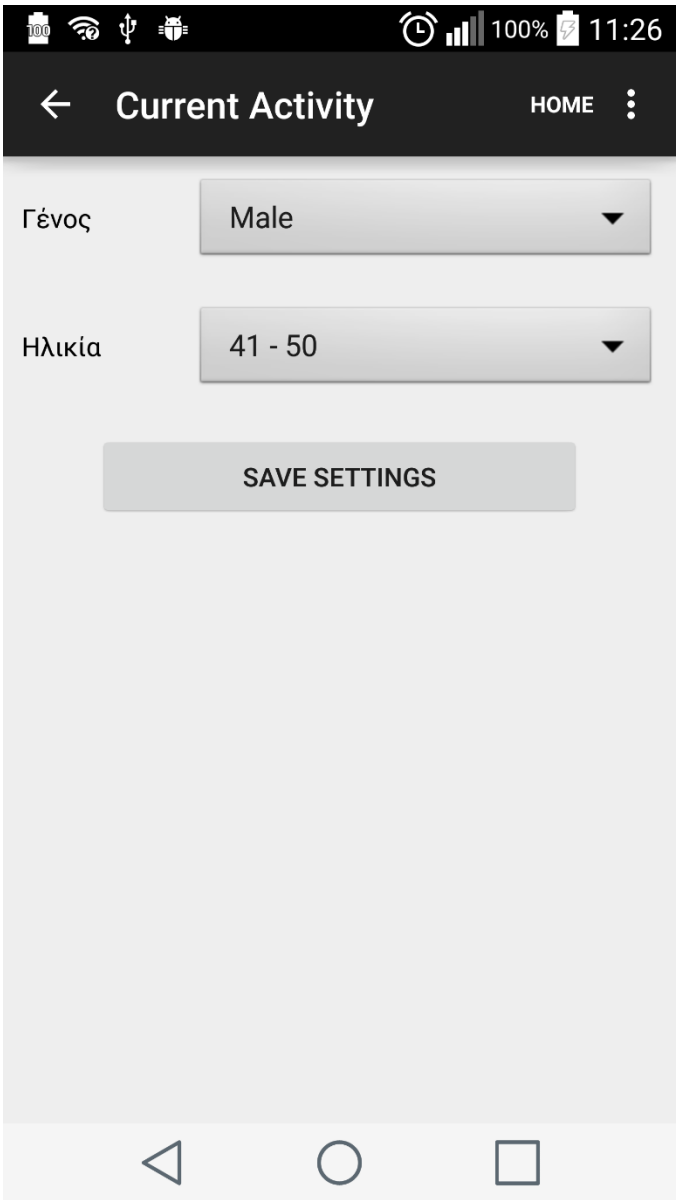
```
Parse.initialize(this,
"qgtvp8EQpF0eu3kFyUqBHrXT8TDqGQMJF6v3RwxO",
"ke93N4GvBuTXQwQnggOsTbuWuJLtxqtpxheKCtP4");
```

Εφόσον έχουν γίνει οι πρώτες αρχικοποιήσεις, η οθόνη του χρήστη εμφανίζει το πρώτο Fragment όπως εμφανίζεται στην Εικόνα 2.



Εικόνα 2: Εμφάνισης αρχικής οθόνης

Την πρώτη φορά που τρέχει η εφαρμογή, ο χρήστης θα πρέπει να ορίσει την ηλικία του καθώς και το γένος του. Αυτό το κάνει μέσα από το μενού Settings (Εικόνα 3). Η TrackerActivity περιέχει τις μεθόδους «LoadSettings» και «SaveSettings» οι οποίες θα διαχειριστούν τα δεδομένα του χρήστη στα SharedPreferences της Android Συσκευής.



Εικόνα 3: Ρυθμίσεις εφαρμογής

Η Βασική διαδικασία έναρξης καταγραφής κίνησης του χρήστη ξεκινάει από την ώρα που ο χρήστης θα πατήσει το Pause (Εικόνα 2). Τότε θα αρχίσει η καταγραφή της δραστηριότητας του χρήστης μέσα από το initialization του Google Api Client. Συγκεκριμένα η «onPauseClick» μέθοδο στην κλάση του Activity θα καλέσει την «startBroadcastListener» μέθοδο η οποία και θα εκκινήσει όλη τη διαδικασία αρχικοποίησης.

Στην πρωτη φάση της αρχικοποίησης θα γίνει έλεγχος μέσω του «LocationManager» αν ο χρήστης έχει ενεργοποιήσει το GPS για τη λήψη τοποθεσίας. Κάτι που είναι από τα βασικά που θα χρησιμοποιήσει ο Google Api Client για τον έλεγχο δραστηριότητας του χρήστη. Αν δεν έχει ενεργοποιηθεί τότε θα δημιουργηθεί ένα νέο intent το οποίο θα μεταφέρει τον χρήστη στις σχετικές ρυθμίσεις.

```
LocationManager service = (LocationManager)
getSystemService(LOCATION_SERVICE);

      boolean enabled =
service.isProviderEnabled(LocationManager.GPS_PROVIDER);


      // check if enabled and if not send user to the GSP settings

      if (!enabled) {

         Intent intent = new
Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);

         startActivity(intent);


         return;

      }
```

Εφόσον ο χρήστης έχει περάσει το στάδιο της ενεργοποίησης του GPS, αρχικοποιούμε τον ActivityGoogleApiTracker ο οποίος υλοποιεί τις μεθόδους του Google Api Client έτσι ώστε να εκκινήσει τη διασύνδεση με αυτό και τη λήψη ενημερώσεων από αυτό.

```
public class ActivityGoogleApiTracker implements
GoogleApiClient.ConnectionCallbacks,
GoogleApiClient.OnConnectionFailedListener
```

Στον Constructor της κλάσης μας ορίζουμε μέσω της μεθόδου addApi πιο Api θέλουμε να χρησιμοποιήσουμε από το GoogleApiClient, όπως φαίνεται παρακάτω.

```
recognitionClient = new GoogleApiClient.Builder(context)
                        .addApi(ActivityRecognition.API)
                        .addApi(LocationServices.API)
                        .addConnectionCallbacks(this)
                       .addOnConnectionFailedListener(this)
                        .build();
```

Επιστρέφοντας στην TrackerActivity το επόμενο στάδιο είναι να κάνουμε register στον LocalBroadcastManager ότι η εφαρμογή δέχεται updates τύπου TrackIntentService.BROADCAST_UPDATE (το οποίο δεν είναι κάτι παραπάνω από απλά ένα διακριτό όνομα για να ξεχωρίζουμε ποια updates θέλουμε να λαμβάνουμε). Η εγγραφή αυτή φαίνεται παρακάτω:

```
LocalBroadcastManager.getInstance(this).registerReceiver(mBroadcastReceiver,
new IntentFilter(TrackIntentService.BROADCAST_UPDATE));
```

Κατά την εκκίνηση της κλάσης «ActivityGoogleApiTracker» μέσω της μεθόδου «start», γίνετια η προσπάθειας σύνδεσης με τον GoogleApiClient. Κατά την προσπάθεια διασύνδεσης γίνεται και η πρώτη καταγραφή μίας fake δραστηριότητας για να γνωρίζουμε το Start Time της δραστηριότητας του χρήστη.

```
//Log the starting/ending time..
```

```
Tracker currentTracker = TrackerUtils.LogTracker(context.MyTracker,
context.LastTrack, new ActivityRecognitionResult(new
DetectedActivity(DetectedActivity.UNKNOWN, TrackerUtils.ConfidenceConst),
1, 1));
```

Έπειτα δημιουργούμε ένα Intent για να ορίσουμε στον GoogleApiClient που θα πρέπει να μας στέλνει τις ενημερώσεις. Στο δικό μας δηλ. IntentService. Συγκεκριμένα ο κώδικας είναι ο:

```
if (recognitionIntent == null)
                {
                        Intent intent = new Intent(context, TrackIntentService.class);
                        recognitionIntent = PendingIntent.getService(context, 0, intent,
PendingIntent.FLAG_CANCEL_CURRENT);//FLAG_UPDATE_CURRENT);
                }
```

Εφόσον ορίσουμε τη λήψη των ενημερώσεων ορίζουμε κάθε πότε θα τις λαμβάνουμε μέσα από το ακόλουθο κομμάτι:

```
PendingResult<Status> status =
ActivityRecognition.ActivityRecognitionApi.requestActivityUpdates(recognitionC
lient, 0, recognitionIntent);
```

Το 0 αντιπροσωπεύει τη γρηγορότερη δυνατή λήψη ενημερώσεων από τον GoogleApiClient.

Κατά τη λήψη ενημερώσεων στην κλάση μας «TrackIntentService», οι ενημερώσεις λαμβάνονται στην παρακάτω μέθοδο:

```
/** Called when a new activity detection update is available.
    *
    */
   @Override
   protected void onHandleIntent(Intent intent){
      try {
         if (ActivityRecognitionResult.hasResult(intent)) {
            Log.d("intentService", "got new update!");


            ActivityRecognitionResult result =
ActivityRecognitionResult.extractResult(intent);


            // notify main activity
            Intent i = new Intent(BROADCAST_UPDATE);
            i.putExtra(RECOGNITION_RESULT, result);
            LocalBroadcastManager manager =
LocalBroadcastManager.getInstance(this);
            manager.sendBroadcast(i);
         }
      }
      catch(Exception ex)
      {
         String message = ex.getMessage();
         message += "d";
      }
   }
```

Με τη λήψη της ενημέρωσης θα πρέπει να διαχειριστούμε τα δεδομένα στο κύριο Thread της εφαρμογής. Για να το κάνουμε αυτό όπως φαίνεται και πιο πάνω καλούμε τον BroadcastManager και περνάμε σε αυτόν την ενημέρωση που θέλουμε. Έτσι λαμβάνουμε τις ενημερώσεις στην TrackerActivity κλάση στο σημείο αρχικοποίησης:

```
private BroadcastReceiver mBroadcastReceiver =  new BroadcastReceiver() {
      @Override
      public void onReceive(Context context, Intent intent) {
         try {
            ActivityRecognitionResult result =
intent.getParcelableExtra(TrackIntentService.RECOGNITION_RESULT);


            KeepResult(result);
         }
         catch(Exception ex){}
      }
   };
```

Τις ενημερώσεις τις διαχειριζόμαστε έπειτα μέσω της KeepResult όπου εκεί θα κρατήσουμε στο log μας τη νέα ενημέρωση. Και μέσω της διαφοράς του enddate και startdate καταλαβαίνουμε τον χρόνο που πέρασε στο διάστημα αυτό. Η TrackerUtils είναι η υπεύθυνη κλάση για τη διαχείριση των Logs και την αποστολή αυτών στην Parse.

Η αποστολή εκτελείτε μέσα από την μέθοδο: «SaveUserStat», όπου στέλνει στον πίνακα στην Parse τα δεδομένα για τη δραστηριότητα του χρήστη. Αυτή η καταγραφή γίνεται προσθετική στο εκάστοτε 24ώρο και αλλάζει σε νέα εγγραφή στον πίνακα σε κάθε επόμενο 24ωρο. Το διάστημα στο οποίο μεσολαβεί μέχρι να στείλουμε την επόμενη ενημέρωση είναι >= του χρόνου σε miliseconds που ορίζονται στην μεταβλητή: «UpdateTicksConst». Με το που περάσει ο χρόνος αυτός, στο επόμενο valid (βάσει του ορίου ConfidenceConst) update θα στείλουμε τις ενημερώσεις στην Parse.

ΚΕΦΑΛΑΙΟ 5 : Χρησιμότητα εφαρμογής

Δυστυχώς η σύγχρονη πραγματικότητα επιβάλει επιμηκυμένα ωράρια εργασίας, κακή και άτακτη διατροφή, μεγάλες περιόδους χωρίς σωματική άσκηση. Πολλές φορές τα άνωθι συνδυάζονται την ίδια χρονική περίοδο επιβαρύνοντας σε επικίνδυνο βαθμό την υγεία.

Με σύμμμαχο το κινητό τηλέφωνο που τελεί ρόλο προσωπικού βοηθού και μας συνοδεύει σε κάθε πτυχή της ζωής μας, η εφαρμογή που δημιουργήσαμε μπορεί να βοηθήσει καταγράφοντας σε πραγματικό χρόνο τις δραστηριότητες του χρήστη.

Στο τέλος της ημέρας ο χρήστης της εφαρμογής έχει καταγεγραμένα και άμεσα προσβάσιμα όσα δεδομένα σχετίζονται με την καθημερινότητά του. Πόση ώρα περπάτησε, πόσο χρόνο αφιέρωσε στην ποδηλασία, πόσο διάστημα διένυσε με το αυτοκίνητο του (χρονικό και χωρικό), πόσο χρόνο ήταν καθηλωμένος στην καρέκλα του γραφείου του (καθαρή υπόθεση).

Εύλογα λοιπόν μπορεί να φθάσει σε συμπεράσματα όσον αφορά τη φυσική του κατάσταση, τους χρόνους που αφιέρωσε σε κάθε δραστηριότητα, τι επίδραση είχε αυτή η δραστηριότητα στον εαυτό του. Αν για παράδειγμα ένας χρήστης παρατηρήσει ότι η καθιστική του κατάσταση υπερβαίνει τις 12 ώρες ημερησίως θα προβεί σε ένα γρήγορο περίπατο μισής ώρας ει δυνατόν και παραπάνω.

Επιπλέον, αν βλέπει πως το γρήγορο περπάτημα δεν αποδίδει αρκετά στον οργανισμό του, θα δει και τις υπόλοιπες δραστηριότητες στις οποίες δεν είναι ενεργός και θα δημιουργηθεί πρόσφορο έδαφος για τη συμμετοχή του σε κάποια από αυτές.

ΚΕΦΑΛΑΙΟ 6 : Μελλοντικές βελτιώσεις

Η εφαρμογή αυτή, όπως και φυσικά κάθε δημιουργία που παίρνει ζωή από τα ανθρώπινα χέρια, έχει περιθώρια περαιτέρω βελτιστοποίησης. Όσον αφορά τις εφαρμογές με τις οποίες ερχόμαστε σε επαφή στα κινητά μας τηλέφωνα οι βελτιώσεις συνήθως διέπονται από τρεις βασικούς πυλώνες:

- Το περιβάλλον διεπαφής με το χρήστη
- Την αποδοτικότητα του κώδικα της εφαρμογής σε ό,τι αφορά:
  A) Το μέγεθος που δημιουργεί η εγκατάστασή της
  B) Το πλήθος των επεξεργαστικών πόρων που απαιτούνται για την εκτέλεσή της
  Γ) Τη μείωση κατανάλωσης της μπαταρίας
- Την προσθήκη επιπλέον στοιχείων στην εφαρμογή

Θα ξεκίνησουμε από το τέλος και θα πούμε το εξής:

Η εφαρμογή ως έχει είναι πολύ χρήσιμη αλλά ένα στοιχείο που όντως θα είχε ουσιαστικό και πρακτικό ενδιαφέρον θα ήταν η παροχή ειδοποιήσεων σε μορφή push notification η οποία θα προέτρεπε το χρήστη σε περιόδους μακράς ακινησίας (όπως πχ κατά την εργασία στο γραφείο, κατά την ξεκούραση στο σπίτι) να σηκωθεί και να κάνει κάποια βήματα, κάποιες διατάσεις ώστε να βελτιωθεί η κυκλοφορία του αίματος και να ενεργοποιήσει το μεταβολισμό του.

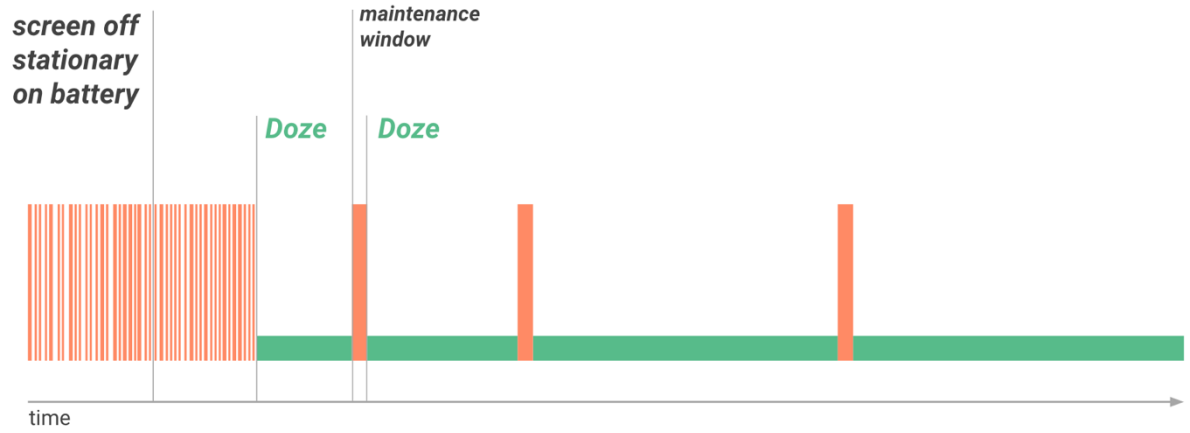
Είναι επίσης εύκολα εφικτό να υπάρχουν «επιβραβεύσεις» από την εφαρμογή ανάλογα με την άσκηση, οι οποίες θα δημιουργούν στο χρήστη κίνητρο να γίνει περισσότερο δραστήριος όπως παραδείγματος χάριν η απόδοση αστεριών όσο περισσότερα χιλιόμετρα τρέχει.

Το τελευταίο παράδειγμα θα μπορούσε κάλλιστα να εφαρμοστεί και στη γενική κατηγορία των στατιστικών όπου με την ύπαρξη μίας λίστας των 100 καλύτερων ανά δραστηριότητα θα μπορούσαν να κερδίσουν ένα ρούχο αθλητικής ένδυσης ή οι 15 πλέον ανερχόμενοι ανά δραστηριότητα να κερδίσουν μία μηνιαία εγγραφή σε ένα γυμναστήριο.

Ακόμη, θα ήταν δυνατή η προσθήκη γεωγραφικών σημείων τα οποία θα αποθηκεύει ο χρήστης. Έτσι όταν η εφαρμογή θα αναγνωρίζει ότι ο χρήστης βρίσκεται για παράδειγμα στο γραφείο του θα μπορούσε να του προτείνει να σηκωθεί και να βηματίσει για λίγη ώρα ή όταν η εφαρμογή θα παρατηρήσει ότι ο χρήστης βρίσκεται στο σπίτι του για πάνω από 12 ώρες θα μπορούσε να τον παρακινεί να βγει από το σπίτι για άθληση.

Σε ό,τι αφορά την κατανάλωση μπαταρίας, ήδη με τη ανακοίνωση της νέας έκδοσης του Android, 6.0 Marshmallow, η οποία ξεκίνησε σιγά σιγά να διατίθεται στις Nexus συσκευές και θα βρει και το δρόμο της για μία πληθώρα τρίτων κατασκευαστών Android smartphone έρχονται δύο νέα χαρακτηριστικά τα οποία θα εξοικονομούν ενέργεια από τη μπαταρία εκ των οποίων το ένα σχετίζεται άμεσα με τις εφαρμογές και το άλλο με τον τρόπο χρήσης.

Το τελευταίο αφορά τη λειτουργία Doze η οποία θα ανιχνεύει τα μεγάλα διαστήματα κατά τα οποία η συσκευή παραμένει ανενεργή και θα υποχρεώνει τη συσκευή να μπει σε μια κατάσταση ύπνου Σε αυτή την κατάσταση, το κινητό θα ξυπνάει κατά διαστήματα επανερχόμενο σε φυσιολογική λειτουργία όπου θα εκτελεί όλες τις λειτουργίες παρασκηνίου όπως ο συγχρονισμός και θα ξαναπέφτει σε λειτουργία ευθύς αμέσως.



Εικόνα: Το Doze παρέχει ένα επαναλαμβανόμενο παράθυρο συντήρησης ώστε οι εφαρμογές να χρησιμοποιούν το διαδίκτυο και να χειρίζονται τυχόν εκκρεμείς δραστηριότητες

Στο τέλος αυτού του χρονικού παραθύρου, το σύστημα επανέρχεται σε κατάσταση ύπνωσης (Doze), διακόπτει τις επικοινωνίες μέσω δικτύου, αναβάλει διεργασίες και συγχρονισμούς. Με την πάροδο του χρόνου το σύστημα ανοίγει το παράθυρο που ενεργοποιεί τις διάφορες δραστηριότητες όλο και πιο αραιά για τη διατήρηση της στάθμης της μπαταρίας. Με το που ο χρήστης κινήσει τη συσκευή, ανοίξει την οθόνη ή συνδέσει το κινητό του με το φορτιστή, η λειτουργία Doze τερματίζεται.

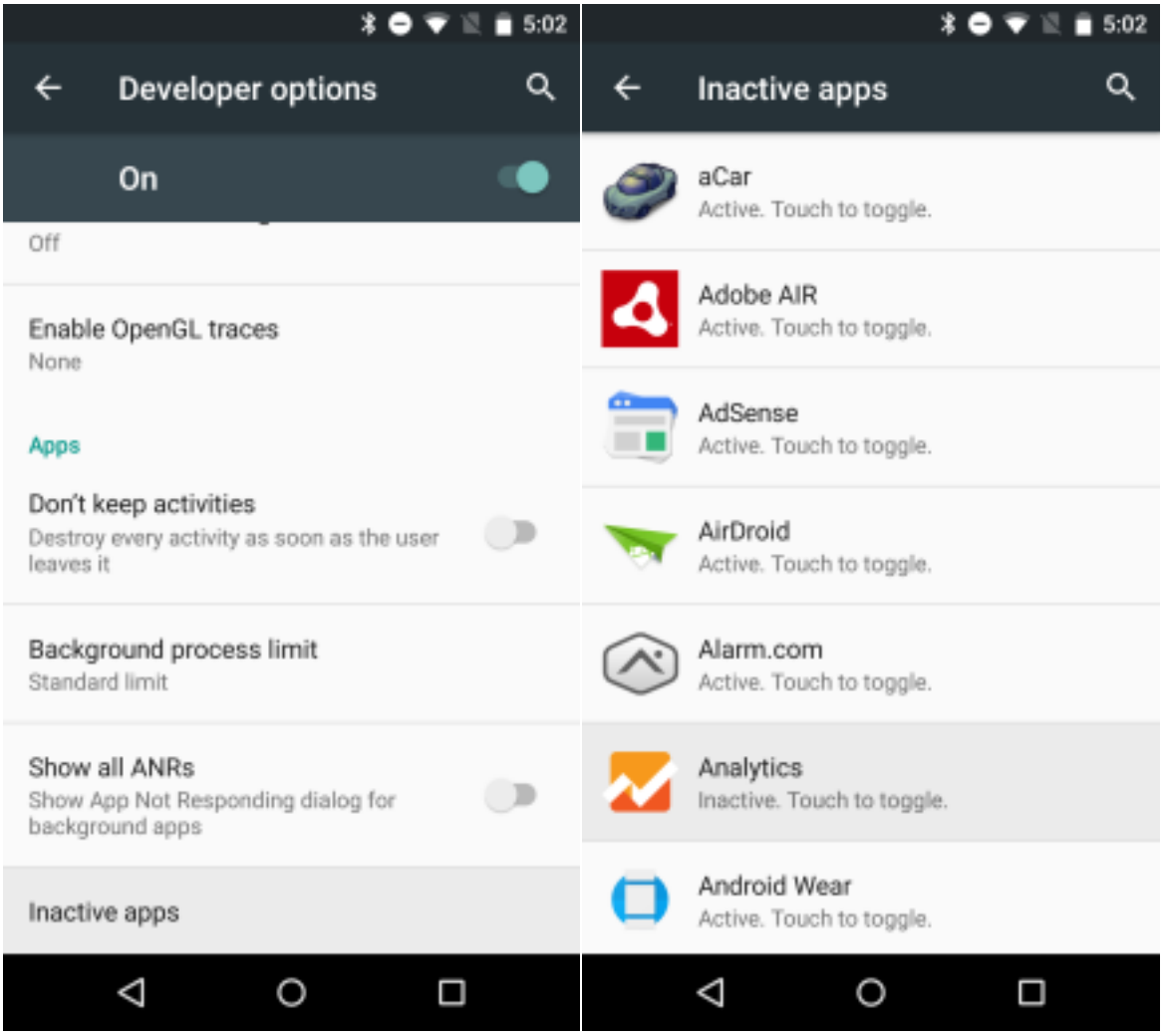
Όπως γίνεται εύκολα κατανοητό, εφόσον η εφαρμογή μας καταγράφει πότε ο χρήστης είναι ανενεργός μπορούμε να θέσουμε ως χρονικό όριο τα 5 λεπτα ακινησίας του χρήστη και η εφαρμογή έκτοτε να υπόκειται στους όρους της λειτουργίας Doze. Με το που ο χρήστης μετακινήσει τη συσκευή του, η λειτουργία θα διακοπεί οπότε μπορούμε να αλλάξουμε κατάσταση κίνησης στην εφαρμογή.

Το πρώτο χαρακτηριστικό όμως που είναι και το πλέον ενδιαφέρον είναι η λειτουργία App Standby. Μέσω αυτής το λειτουργικό σύστημα αντιλαμβάνεται πότε η εφαρμογή έχει αρκετό διάστημα να χρησιμοποιηθεί ενεργά (για παράδειγμα ο χρήστης δεν αλληλεπιδρά μέσω αγγιγμάτων με αυτή) και αναλαμβάνει να διακόψει κατά τα διαστήματα την επικοινωνία αυτής με το διαδίκτυο, αναβάλει τους συγχρονισμούς και η εφαρμογή συμπεριφέρεται σα να βρίσκεται σε αδράνεια.

Για την εφαρμογή αυτής της κατάστασης ο χρήστης δε πρέπει να:

- Ανοίξει ρητά την εφαρμογή
- Η εφαρμοργή να έχει μια διεργασία που να τρέχει στο προσκήνιο
- Η εφαρμογή να παράξει ειδοποιήσεις που θα εμφανιστούν στην οθόνη του χρήστη ή στο συρτάρι ειδοποιήσεων

Όταν ο χρήστης συνδέσει τη συσκευή του με το φορτιστεί τότε η λειτουργία App Standby παύει και όλες οι διεργασίες εκτελούνται κανονικά. Διαφορετικά, για ανενεργές εφαρμογές η λειτουργία επιτρέπει τη σύνδεση με το διαδίκτυο περίπου μία φορά την ημέρα.

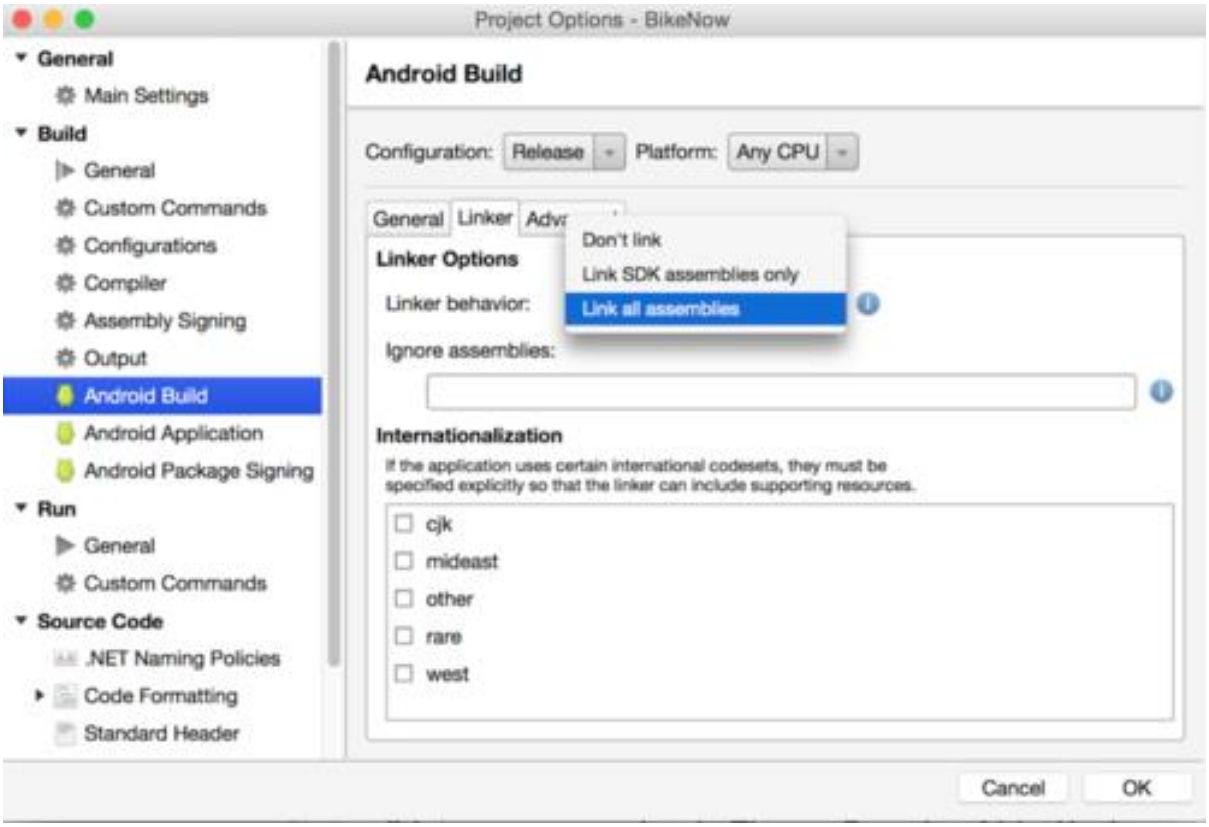


Εικόνα: Εφαρμογές αδρανοποιημένες από τη λειτουργία App Standby

Στην περίπτωση μας, θα μπορούσαμε να βάλουμε κάποια φίλτρα όπως για παράδειγμα αν η ώρα περάσει τις 10 μετά μεσημβρίας η εφαρμογή να τίθεται καθ' ολοκληρίαν στο παρασκήνιο και να επιτρέπει στη λειτουργία App Standby να αναστέλλει την αποστολή δεδομένων στη διαδικτυακή βάση δεδομένων. Τα τελευταία θα αποστέλλονται κατά την επαναφορά της σύνδεσης με το διαδίκτυο. Έτσι η συσκευή δε θα έχει άσκοπα wake locks (διεργασίες που παίρνουν τον έλεγχο παροχής ενέργειας της συσκευής για διάφορα υποσυστήματα όπως η CPU, το WiFi κλπ και οι οποίες δεν επιτρέπουν στη συσκευή να μπει σε κατάσταση αδράνειας) και θα διατηρήσει καλύτερα τα επίπεδα ενέργειας της υπολοιπόμενης μπαταρίας της συσκευής.

Για τη μείωση του μεγέθους της εφαρμογής μπορούμε να χρησιμοποιήσουμε προγράμματα όπως το Xamarin το οποίο λειτουργεί ώς «σύνδεσμος»/ linker. Η λειτουργία του έχει ως εξής: Χρησιμοποιεί μία στατική ανάλυση της εφαρμογής ώστε να διαπιστώσει ποιοι types όντως χρησιμοποιούνται, όπως και ποια assemblies και members. Στη συνέχεια ο «σύνδεσμος»

συμπεριφέρεται σα συλλέκτης «σκουπιδιών» αφού ψάχνει συνεχώς για τέτοια types, members και assemblies και τα εσωκλείει. Όσα δε συμπεριλαμβάνονται σε αυτό το κλειστό γκρουπ, απορρίπτονται ως άχρηστα.



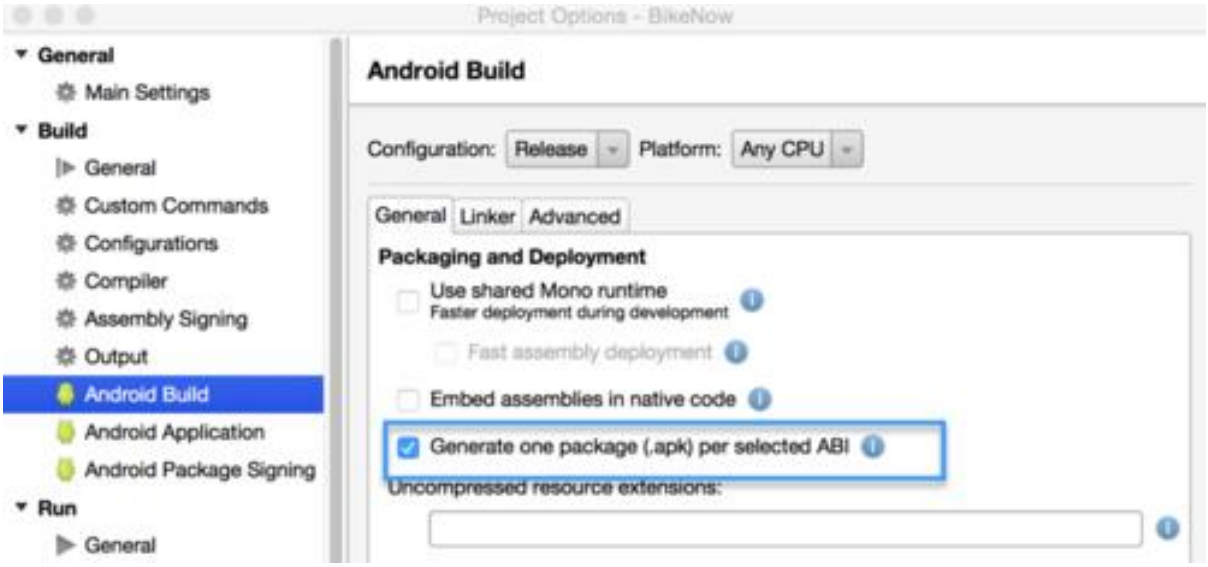
Εικόνα: Περιβάλλον χρήστη του Xamarin

Έτσι μετά τη χρήση του «linker» το μέγεθος ενός Android Application Package (APK) μειώνεται δραστικά σε έως και το 18% του αρχικού μεγέθους όπως φαίνεται στον παρακάτω πίνακα.

Configuration	1st APK	2nd APK
Έκδοση άνευ Linking	14.0 MB	16.0 MB
Έκδοση με Linking	4.2 MB	2.9 MB

Μία άλλη μέθοδος που θα μπορούσαμε να χρησιμοποιήσουμε θα ήταν να κατακερματίσουμε σε διάφορα APKs την εφαρμογή. Στο Android υπάρχουν τα Application Binary Services (ABI) τα οποία υποστηρίζονται με την έκδοση κάθε εφαρμογής. Το πλέον χρησιμοποιούμενο είναι το armeabi-v7a (επί της ουσίας γιατί οι επεξεργαστές με αρχιτεκτονική ARM- Acorn RISC Machine έχουν επικρατήσει στην αγορά). Παρόλα αυτά οι εφαρμογές επιβάλλεται να τρέχουν σε κάθε

είδους εφαρμογή χρειάζεται να υποστηρίζονται και τα ABI για παλαιού τύπου armeabi και x86 τύπου επεξεργαστές. Η υποστήριξη επιπλέον του ενός προτύπου όμως απαιτεί και ξεχωριστά libmonodroid και sgen. Όταν λοιπόν επιλέγουμε να εκδόσουμε ένα APK για την εφαρμογή μας, αυτή συνοδεύεται από υλικό που κατά περίπτωση είναι άχρηστο για κάθε χρήστη (δε χρειάζονται στο χρήστη μιας x86 αρχιτεκτονικής συσκευής η βιβλιοθήκες που απαιτούν οι τύπου ARM αρχιτεκτονικής συσκευές) μεγαλώνοντας κατά πολύ το μέγεθος του APK.



Εικόνα: Δημιουργώντας ένα APK ανα ABI

Η λύση είναι να δημιουργήσουμε ένα APK για κάθε ABI. Έτσι δημιουργούμε τρεις ξεχωριστές εφαρμογές για κάθε τύπο αρχιτεκτονικής οι οποίες είναι σημαντικά μικρότερες από μία μοναδιαία για όλες. Το αποτέλεσμα είναι τρεις εφαρμογές των 10MB για παράδειγμα από μία των 30MB σώζοντας χώρο στη συσκευή του χρήστη.

Τέλος, χρήσιμη μείωση του μεγέθους παρουσιάζει η απομάκρυνση κάθε πληροφορίας σχετικά με την αποσφαλμάτωση (εφεξής debug) της εφαρμογής. Γενικά η εφαρμογή δε βλέπει αυτή την πληροφορία και το λειτουργικό σύστημα Android δε την απαιτεί για να τρέξει η εφαρμογή. Κατ' αυτόν τον τρόπο κάθε πληροφορία σχετιζόμενη με το debug καταλαμβάνει αναίτια λόγο στο σύστημα. Προκειμένου να την απομακρύνουμε θα εκτελέσουμε το εξής:

```
static final debug = false;

if (debug) {
```

```
    Log.v(TAG, "Debug …”);
}
```

Είναι σημαντικό η σημαία (flag) debug να έχει πάρει τιμή compile time (πχ να έχει την τιμή static final) ούτως ώστε ο compiler να καταφέρει να απομακρύνει κάθε σχετική με το debug πληροφορία.

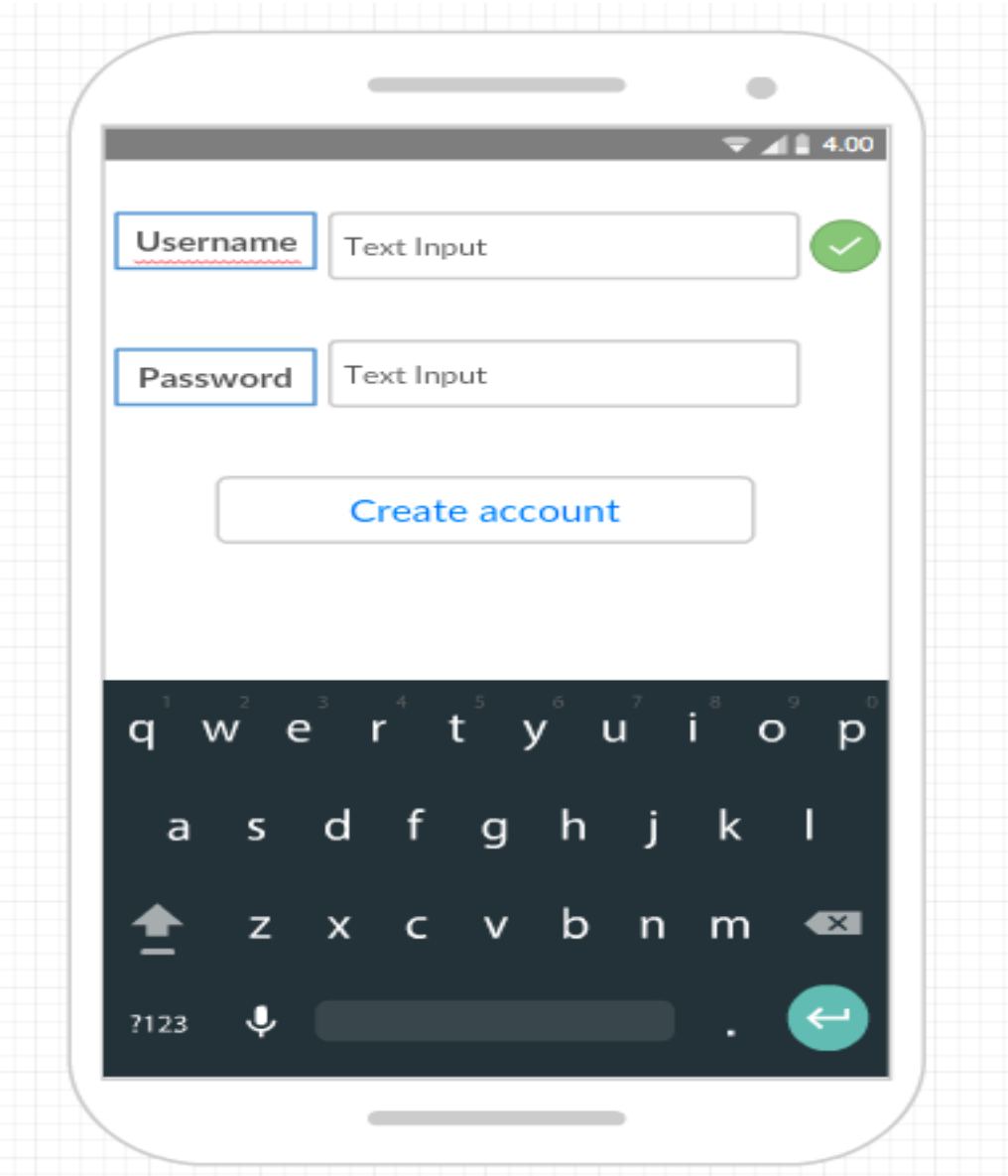
```
public void myDebugPrint() {
    if (Debug) {
        Log.v(TAG, "Debug …”);
    }
}
…
myDebugPrint()
…
```

Επεξήγηση: Κώδικας που αφαιρεί κάθε σύμβολο debug από τις native βιβλιοθήκες

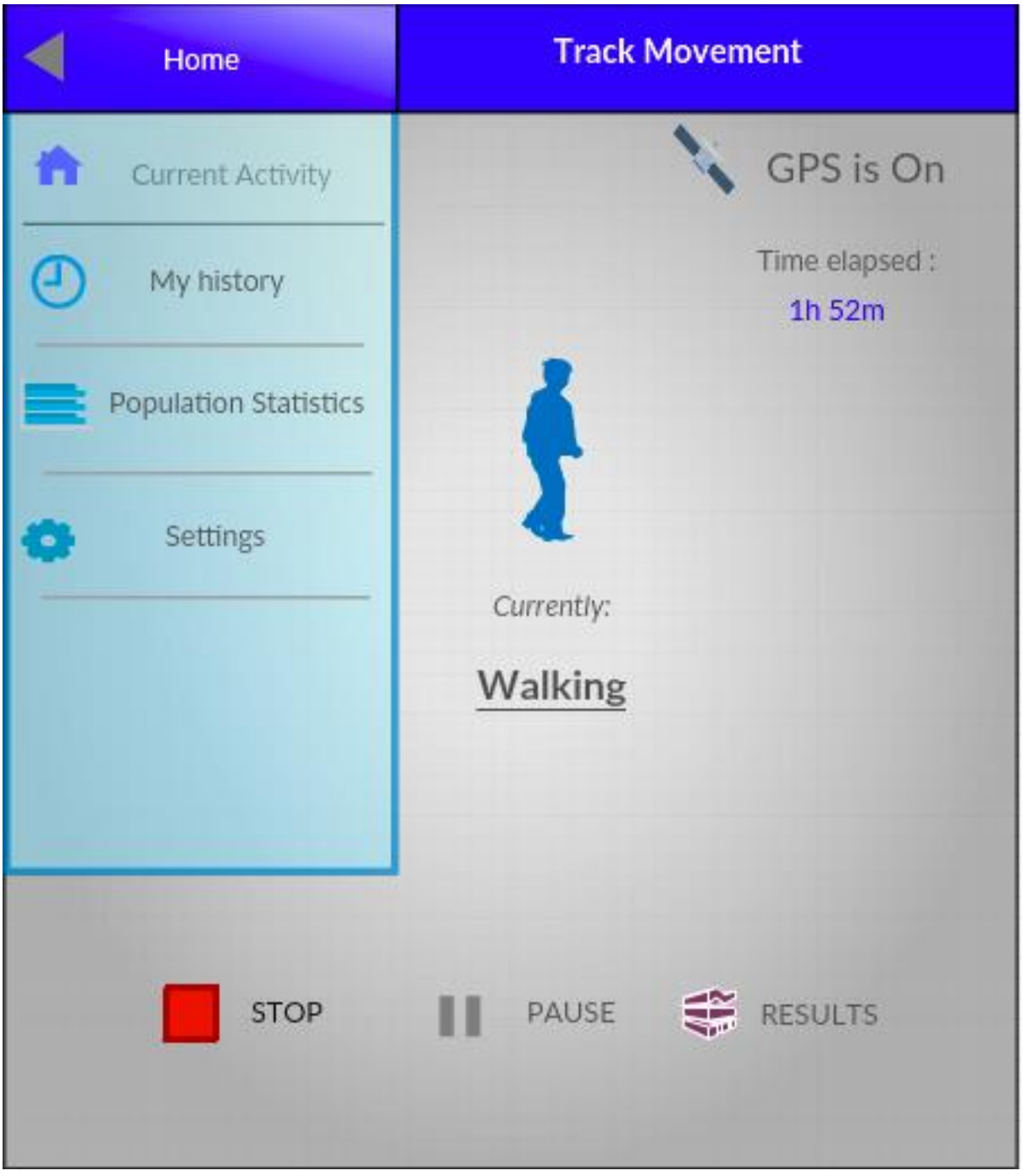
Η χρησιμοποίηση debug symbols έχει νόημα αν η εφαρμογή είναι ακόμη σε φάση ανάπτυξης και απαιτεί τον εντοπισμό σφαλμάτων. Εμείς όμως θέλουμε να εκδόσουμε μια release version της εφαρμογής άρα προτείνεται η απομάκρυνση των debug symbols από τις native βιβλιοθήκες (αρχεία .so) ώστε να εξοικονομήσουμε το χώρο. Αυτό γίνεται χρησιμοποιώντας την εντολή arm-eabi-strip από το Android NDK (Native Development Kit).

Βελτιώνοντας το περιβάλλον χρήσης:

Μία πολύ χρήσιμη προσθήκη στο διαδραστικό περιβάλλον χρήσης θα ήταν η εισαγωγή μίας οθόνης εγγραφής του χρήστη. Κατά συνέπεια αυτό σημαίνει την υιοθέτηση λογαριασμού ανα χρήστη και την εγγραφή αυτού στη βάση δεδομένων της εφαρμογής. Το παράδειγμα δίνεται στην παρακάτω εικόνα:



Μετά την εγγραφή του χρήστη στην εφαρμογή θα είχε ενδιαφέρον μιας νέας σχεδιάσης κεντρική οθόνη όπως φαίνεται παρακάτω η οποία θα είναι πιο φιλική προς το χρήστη.



Σε αυτή λοιπόν την εικόνα έχουμε ένα νέο εικονίδιο για το αναδυόμενο μενού το οποίο πλέον δεν καταλαμβάνει όλο το κατακόρυφο χώρο ώστε να μη δημιουργείτε η αίσθηση στο χρήστη ότι έφυγε από το κύριο παράθυρο.

Οι ενέργειες πλέον στο αναδυόμενο μενού περιλαμβάνουν στα αριστερά και το αντίστοιχο εικονίδιο. Η πρώτη επιλογή είναι το «Current Activity» επί της ουσίας επιστρέφει το χρήστη στην κύρια οθόνη κλείνοντας το αναδυόμενο παράθυρο. Η επιλογή «My history» δείχνει τις προηγούμενες δραστηριότητες του χρήστη. Η επιλογή «Population Statistics» δείχνει τα στατιστικά του γενικότερου πληθυσμού ενώ τέλος η επιλογή «Settings» δείχνει στο χρήστη τις ρυθμίσεις της εφαρμογής.

Στην κύρια οθόνη έχουμε πλέον ένα εικονίδιο για την κατάσταση του Global Positioning System (Ενεργό-On, Ανενεργό-Off). Κάτω από αυτό καταγράφεται η ώρα που έχει διελεύσει από την έναρξη της δραστηριότητας. Η δραστηριότητα πλέον αυτή καθεαυτή αντιστοιχίζεται με ένα ευδιάκριτο εικονίδιο και ακριβώς από κάτω υπογραμμίζεται με τη λέξη της.

Τέλος έχουμε προστεθεί τρία κουμπιά ενεργειών. Με το πρώτο «STOP» σταματάμε την καταγραφή της δραστηριότητας. Με το δεύτερο «PAUSE» αναστέλλουμε προσωρινά την καταγραφή της δραστηριότητας ενώ με το τρίτο δείχνουμε τα αποτελέσματα αυτή της δραστηριότητας.

ΚΕΦΑΛΑΙΟ 7 : Παρόμοιες Εφαρμογές

Στο κεφάλαιο αυτό θα γίνει μία αναφορά σε ήδη υπάρχουσες εφαρμογές οι οποίες βασίζονται πάνω στην αναγνώριση τη δραστηριότητας του χρήστη και είναι διαθέσιμες από το Google Store και Apple Store.

Goole Fit

Ας ξεκινήσουμε με την εφαρμογή Goole Fit η οποία διατίθεται δωρεάν από το Google Play Store. ο Google Fit αποτελεί την απάντηση της εταιρείας στο Apple Health. Χάρη στο Google Fit, δεδομένα που αφορούν στην άσκηση ή στην υγεία που συλλέγονται και από εφαρμογές τρίτων -άλλωστε αποτελεί στην ουσία ένα σύνολο από APIs- συγκεντρώνονται σε ένα μοναδικό σημείο. Εκμεταλλευόμενο τους αισθητήρες του κινητού, το λογισμικό παρακολουθεί το περπάτημα, το τρέξιμο ή άλλες δραστηριότητες άσκησης (π.χ ποδηλασία), επιτρέποντας στον χρήστη να θέσει συγκεκριμένους στόχους και να επωφεληθεί από ορισμένες φιλικές συμβουλές.

Η εφαρμογή συνεργάζεται άριστα και με Android εφαρμογές τρίτων όπως αναφέραμε -Strava, Withings, Runtastic, Runkeeper, Noom Coach κ.ά- καθώς και με συσκευές Android Wear.
Οι χρήστες έχουν πρόσβαση στα δεδομένα της εφαρμογής από κινητό, tablet ή ακόμα και μέσω του web αν χρειαστεί

Για την λειτουργία της εφαρμογής Google Fit, ο χρήστης υποχρεούται να επιτρέψει στο app την πρόσβαση σε ένα σύνολο προσωπικών του στοιχείων όπως βάρος, ύψος, ηλικία, αλλά και δεδομένα όπως η γεωγραφική του θέση, ενώ η εφαρμογή λειτουργεί και σε φορετές συσκευές, όπως 'έξυπνα' ρολόγια, με λειτουργικό Android .

Αρχικά μας ζητείται να καταχωρήσουμε τα φυσικά μας χαρακτηριστικά.(βάρος ύψος και φύλο).. Στην συνέχεια μπορουμε να ορίσουμε ημερήσιους στόχους για τα εξής:

- Πόσα βήματα θέλουμε να κάνουμε
- Πόσο χρόνο θέλουμε να αφιερώσουμε στην προπόνηση
- Την απόσταση που θέλουμε να διανύσουμε με πεζοπορία, τρέξιμο ή ποδηλασία
- Τον αριθμό των θερμίδων που θέλουμε να κάψουμε

  Όταν ανοίξετε το Fit, βλέπουμε πόσο κοντά βρισκόμαστε στην ολοκλήρωση του στόχου μας. Αφού παρακολουθήσουμε τη δραστηριότητά μας για μερικές ημέρες, θα δούμε, επίσης, προτάσεις για μελλοντικούς στόχους. Αυτοί έχουν ως σκοπό να μας βοηθήσει να βελτιώσουμε την καθημερινή φυσική κατάστασή μας.

  Ο προεπιλεγμένος στόχος την πρώτη φορά που θα ξεκινήσουμε με το Fit θα είναι μία ώρα δραστηριότητα την ημέρα..

  Στο Google Fit, μπορούμε να παρακολουθήσουμε τη δραστηριότητα πεζοπορίας, τρεξίματος, ποδηλασίας και περισσότερα. Για παράδειγμα, μπορούμε να δούμε πόσα βήματα κάναμε ή πόσο μεγάλη ήταν η διαδρομή μας με το ποδήλατο, πόση ώρα παίξαμε ποδόσφαιρο ή κάναμε σκι.

  Όταν καταγράφετε μια δραστηριότητα, τις περισσότερες φορές μπορούμε να δούμε εκτιμήσεις για τα εξής:

- Διάρκεια: Η διάρκεια μιας συγκεκριμένης δραστηριότητας.
- Βήματα: Πόσα βήματα κάναμε.
- Θερμίδες: Πόσες θερμίδες κάψαμε.
- Απόσταση: Η απόσταση που διανύσαμε με πεζοπορία ή τρέξιμο. Αυτή η λειτουργία δεν είναι προς το παρόν διαθέσιμη στο στατικό ποδήλατο.
- Βάρος: Η μεταβολή του βάρους μας με την πάροδο του χρόνου. Για να παρακολουθήσουμε το βάρος μας, θα πρέπει να το καταχωρίσουμε μη αυτόματα.

  Σημαντικό: Για να δούμε την απόσταση που έχουμε διανύσει ή τις θερμίδες που έχουμε κάψει, θα πρέπει να συμπληρώσουμε το ύψος, το βάρος και το φύλο μας.

  Οι πληροφορίες ενημερώνονται αυτόματα σε κάθε συσκευή στην οποία χρησιμοποιείτε το Fit, όπως ο υπολογιστής μας, η κινητή μας συσκευή ή το ρολόι Android Wear.

  Μπορούμε οποιαδήποτε στιγμή να επεξεργαστούμε ή να διαγράψουμε οποιαδήποτε άσκηση καταγράφεται από το Fit.

  Όταν ορίσετε το Google Fit να ξεκινήσει την παρακολούθηση της δραστηριότητάς μας, μπορούμε να χρησιμοποιήσουμε την εφαρμογή ή τον ιστότοπο, για να δούμε πόσο κοντά βρίσκόμαστε στην ολοκλήρωση των στόχων μας και να συγκρίνουμε τη δραστηριότητά μας με τις προηγούμενες ημέρες.

  Μπορούμε να προβάλουμε την ημερήσια πρόοδο της φυσικής κατάστασής μας στο Fit, όπως πόσα βήματα έχουμεκάνει, πόσο μεγάλη ήταν η διαδρομή μας με το ποδήλατο και πόσο κοντά βρισκόμαστε στην ολοκλήρωση του ημερήσιου στόχου μας.

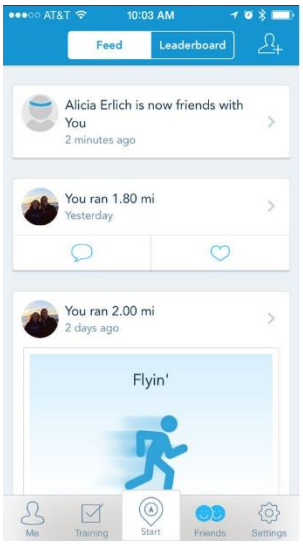
- **Για την τρέχουσα ημέρα:** Μπορούμε να δούμ την παρακολούθηση της φυσικής κατάστασης της τρέχουσας ημέρας στο επάνω μέρος της αρχικής σελίδας στην εφαρμογή ή στον ιστότοπο Fit.

- **Για μια προηγούμενη ημέρα:** Κάνουμε κύλιση προς τα κάτω στην αρχική σελίδα, για να δείτε πληροφορίες σχετικά με τη δραστηριότητα μιας προηγούμενης ημέρας.
  Για να δούμ πιο λεπτομερείς πληροφορίες παρακολούθησης, όπως γραφήματα και διαχωρισμούς δραστηριοτήτων, βρίσκουμε τη δραστηριότητα ή την ημερήσια σύνοψη που θέλουμε στη ροή δραστηριοτήτων μας και κάνουμε κλικ.

Run Keeper

Το RunKeeper μας επιτρέπει να παρακολουθούμε όχι μόνο τα τρεξίματα σας, αλλά ακόμα και τις προπονήσεις που κάνουμε περπατώντας ή κάνοντας ποδήλατο. Μπορούμε να έχουμε τα προσωπικά μας αρχεία και να βάζουμε στόχους, όπως το να χάσουμε βάρος ή να καλύψουμε μια απόσταση.

Μια εφαρμογή αφιερωμένη σε υπαίθριες αθλητικές δραστηριότητες, όπως τρέξιμο, ποδηλασία ή περπάτημα. Χάρη στο Run Keeper, μπορείτε να δημιουργήσετε τις διαδρομές σας ή να καταγράψετε τα χιλιόμετρα που έχετε διανύσει χάρη στο ενσωματωμένο GPS. Μπορείτε επίσης να δείτε μια πλήρη περίληψη της προπόνησης σας (χλμ., καρδιακοί παλμοί, διαδρομές, διανυόμενη απόσταση). Υπάρχει επίσης η δυνατότητα κοινοποίησης των στατιστικών εκγύμνασής σας στα μέσα κοινωνικής δικτύωσης.



Πιο συγκεκριμένα μπορούμε να έχουμε τα παρακάτω χρησιμοποιώντας αυτή την εφαρμογή:

Μπορούμε να υπολογίστε τον ρυθμό τρεξίματος, την ταχύτητα του ποδηλάτου, την απόσταση διαδρομής, ανάδειξη και κάψιμο θερμίδων για οποιαδήποτε δραστηριότητα φυσικής κατάστασης - σε υψηλή ακρίβεια και σε πραγματικό χρόνο!
• Μπορούμε να βρούμε και ακολουθήσουμε προ-σχεδιασμένες διαδρομές και να χαρτογραφήσουμε το τρέξιμό μας ή να περπατήσουμε στο δρόμο,
• Μπορούμε να λαμβάνουμε ενημερώσεις ήχου.

• Μπορούμε να παρακολουθήσουμε τον καρδιακό ρυθμό σας και να μείνουμε σε ζώνες καρδιακού ρυθμού
• το RunKeeper ενσωματώνει αυτόματα με τη μουσική εφαρμογή του τηλεφώνου σας..

Runtastic

Η runtastic PRO είναι ία εφαρμογή που χρησιμοποιεί GPS και παρέχει πληροφορίες όσων αφορά την απόσταση που καλύπτουμε, την ταχύτητα, τις θερμίδες που καίμε κ.α. Πολλά ακόμα εξαιρετικά features συνθέτουν αυτήν την εφαρμογή όπως:

– Voice feedback κατά την διάρκεια της εξάσκησης

– Live tracking: οι φίλοι σας μπορούν να βλέπουν που είστε

– Auto pause: η εφαρμογή σταματάει να μετράει όταν σταματάτε

– ρυθμίσεις για να επιλέξετε το heart rate την άσκησης σας καθώς και να μετράει τους σφυγμούς σας ανά πάσα στιγμή

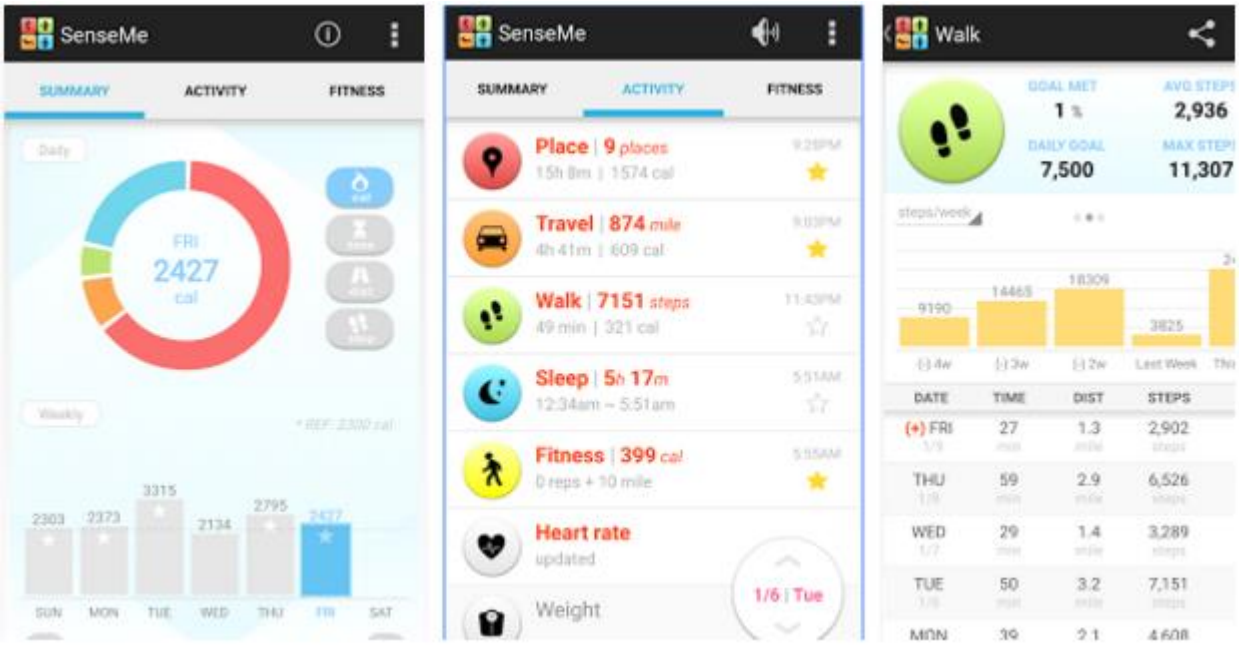
– η εφαρμογή παρέχει επίσης πληροφορίες για τον καιρό, την ανατολή και δύση του ηλίου κ.α.

– μπορείτε να θέσετε στόχους βάση θερμίδων, ταχύτητας κ.α. και να λαμβάνετε voice feedback όταν τους πιάνετε

– Geotagging: βγάλτε φωτογραφίες κατά την διάρκεια της άσκησής σας

– ημερολόγιο για να κρατάτε στοιχεία όσων αφορά την άσκησή σας τα χιλιόμετρα, τις θερμίδες που καίτε κ.α.

Pedometer and Fitness



Η εφαρμογή αυτή προσφέρει τις παρακάτω λειτουργίες:

- Υπολογίζει τον αριθμό βημάτων στο περπάτημα.
- Αναγνωρίζει την κατάσταση ύπνου καθώς και πόσες ώρες κάποιος κοιμάται.
- Αναγνωρίζει τις καθημερινές δραστηριότητες και υπολογίζει τις θερμίδες που κάποιος καίει.
- Υπολογίζει το χάσιμο βάρους με την χρήση BMI calculator.
- Υπολογίζει τις δραστηριότητες του χρήστη και την διάρκεια που κάποιος οδηγά,κάνει ποδήλατο, τρέχει κτλ., ενώ παρουσιάζει και τις διαδρομές στον χάρτη.
- Αναγνωρίζει την κίνηση σε ποικίλες ασκήσεις κατά την διάρκεια αθλημάτων (σχοινάκι,ελλειπτικό,διάδρομο)

ΚΕΦΑΛΑΙΟ 8 : ΒΙΒΛΙΟΓΡΑΦΙΑ

1. https://developers.google.com/android/reference/com/google/android/gms/common/api/GoogleApiClient
2. https://developers.google.com/android/guides/api-client
3. http://developer.android.com/index.html
4. Stefan Br¨ ahler, Analysis of the Android Architecture, Kalsruher Institut fur Technologie
5. http://stackoverflow.com/
6. Wikipedia, the free encyclopedia - http://en.wikipedia.org/wiki/Main_Page
7. Hello, Android, Introducing Google's Mobile Development Platform 3rd Edition – Ed Burnette, The pragmatic programmers
8. Aharony, N. και W. Gardner: Funf Developer Site. http://www. funf.org, 2012
9. Anjum, Alvina και Muhammad Usman Ilyas: Activity recognition using smartphone sensors. Στο Consumer Communications and Networking Conference (CCNC), 2013 IEEE, σελίδες 914–919. IEEE, 2013.
10. Politi, O., I. Mporas, και V. Megalooikonomou: Human motion detection in daily activity tasks using wearable sensors. Στο Eurosipco. IEEE, 2014.
11. Yang, Jun: Toward physical activity diary: motion recognition using simple acceleration features with mobile phones. Στο Proceedings of the 1st international workshop on Interactive multimedia for consumer electronics, σελίδες 1–10. ACM, 2009.
12. http://www.sitepoint.com/creating-cloud-backend-android-app-using-parse/
13. http://www.androidbegin.com/tutorial/android-parse-com-simple-listview-tutorial/

14. Hoseini-Tabatabaei, Seyed Amir, Alexander Gluhak, και Rahim Tafazolli: A survey on smartphone-based systems for opportunistic user context recognition. ACM Computing Surveys (CSUR), 45(3):27, 2013

15. http://www.vogella.com/tutorials/Android/article.html

16. http://developer.android.com/training/location/index.html

17. Activity Recognition using Cell Phone Accelerometers Jennifer R. Kwapisz, Gary M. Weiss, Samuel A. Moore

18. A Survey of Online Activity Recognition Using Mobile Phones Muhammad Shoaib 1,*, Stephan Bosch 1 , Ozlem Durmaz Incel 2 , Hans Scholten 1 and Paul J.M. Havinga 1

19. Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine Davide Anguita1 , Alessandro Ghio, Luca Oneto,