



**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**  
**ΣΧΟΛΗ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ**  
**ΤΟΜΕΑΣ ΒΙΟΜΗΧΑΝΙΚΗΣ ΔΙΟΙΚΗΣΗΣ &**  
**ΕΠΙΧΕΙΡΗΣΙΑΚΗΣ ΕΡΕΥΝΑΣ**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

---

**Περιρρέουσα Νοημοσύνη (Ambient Intelligence)**  
**Ανάπτυξη εφαρμογής : Ανίχνευση ανθρωπίνων**  
**δραστηριοτήτων**

---

**Σεκλιζιώτης Δ. Γεώργιος**

Επιβλέπων:

**Ναθαναήλ Δημήτριος, Λέκτορας Ε.Μ.Π.**

Αθήνα, Οκτώβριος 2014

Υπεύθυνη δήλωση για τη λογοκλοπή:

*Γνωρίζω και έχω κατανοήσει τους κανόνες για τη λογοκλοπή και τον τρόπο σωστής αναφοράς των πηγών που πρέπει να τηρούνται κατά τη συγγραφή Διπλωματικών Εργασιών. Δηλώνω ότι, από όσα γνωρίζω, το περιεχόμενο της παρούσας Διπλωματικής εργασίας είναι προϊόν δικής μου δουλειάς και υπάρχουν αναφορές σε όλες τις πηγές που χρησιμοποίησα.*

**Σεκλιζιώτης Γεώργιος**

## Ευχαριστίες

Η παρούσα διπλωματική εργασία αποτέλεσε μια πολύ καλή αφορμή να ασχοληθώ με ένα καινούριο για μένα γνωστικό πεδίο τις εφαρμογές έξυπνων κινητών τηλεφώνων. Στο σημείο αυτό θα ήθελα να εκφράσω τις ευχαριστίες μου στον επιβλέποντα της διπλωματικής μου εργασίας κύριο Δημήτριο Ναθαναήλ Λέκτορα στη Σχολή Μηχανολόγων Μηχανικών του Εθνικού Μετσόβιου Πολυτεχνείου στον Τομέα Βιομηχανικής Διοίκησης & Επιχειρησιακής Έρευνας , για την ευκαιρία που μου προσέφερε να ασχοληθώ με ένα τόσο επίκαιρο θέμα, για την εμπιστοσύνη που μου έδειξε αναθέτοντάς μου αυτή την ενδιαφέρουσα διπλωματική, για τη συνεχή καθοδήγησή και βοήθειά του κατά την εκπόνηση της διπλωματικής μου εργασίας, για τις χρήσιμες συμβουλές του καθώς και για την πολύ καλή συνεργασία που είχαμε καθ' όλη τη διάρκεια της εργασίας μου.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου για τη στήριξη που μου παρέχει διαρκώς, καθώς και τους φίλους μου στο Ε.Μ.Π. για αυτά τα υπέροχα χρόνια που περάσαμε μαζί.



## Περίληψη

Στα πλαίσια της διπλωματικής εργασίας γίνεται ανάλυση των διαδικασιών ανίχνευσης συγκεκριμένων ανθρωπίνων κινήσεων μέσα από επεξεργασία δεδομένων που λαμβάνονται από τους αισθητήρες των έξυπνων κινητών και σύγκρισης τους με ασαφή μοντέλα των ιδίων κινήσεων. Σε αυτό το πλαίσιο αναπτύχθηκε μια εφαρμογή η οποία αναγνωρίζει την πορεία ενός χρήστη έξυπνου κινητού τηλεφώνου και την αποθηκεύει σε μία απομακρυσμένη βάση δεδομένων. Επιπροσθέτως μέσα από μοντελοποίηση διαδικασιών όπως το περπάτημα και η οδήγηση χρησιμοποιώντας αισθητήρες γίνεται ανίχνευση της θέσης στάθμευσης του χρήστη η οποία αποθηκεύεται στην απομακρυσμένη βάση. Αυτές οι διαδικασίες γίνονται χωρίς να χρειάζεται να υπάρχει διάδραση του χρήστη με την εφαρμογή αφού μπορεί να λειτουργεί στο υπόβαθρο χωρίς να αντιλαμβάνεται ο χρήστης ότι λειτουργεί. Ο χρήστης μπορεί να χρησιμοποιεί την εφαρμογή μονάχα για να ενημερώνεται για την θέση στάθμευσης είτε με προβολή των αποθηκευμένων πληροφοριών (γεωγραφικών συντεταγμένων) της βάσης είτε με αυτόματη αναπαράσταση (με στίγμα) των συντεταγμένων σε οδικό χάρτη όπου επιθυμεί αυτός μέσα από τα αντίστοιχα κουμπιά που βρίσκονται στην διεπαφή χρήστη (user interface) της εφαρμογής. Πέρα από αυτές τις διαδικασίες ο χρήστης μπορεί μονάχα να ενεργοποιεί και να απενεργοποιεί την υπηρεσία της εφαρμογής που τρέχει στο υπόβαθρο μέσα από την διεπαφή χρήστη της εφαρμογής.

Η εφαρμογή αναπτύχθηκε στα πλαίσια της περιρρέουσας νοημοσύνης. Για ευκολότερη κατανόηση γίνεται αναφορά στα στοιχεία της και στα πεδία εφαρμογής της. Στην συνέχεια γίνεται σε βάθος ανάλυση της επίτευξης της περιρρέουσας νοημοσύνης μέσω αισθητήρων και πιο συγκεκριμένα μέσω των αισθητήρων των έξυπνων κινητών τηλεφώνων αφού έχουν πολλές δυνατότητες και η χρήση τους έχει εξαπλωθεί ραγδαία. Στα συμπεράσματα αναδεικνύονται οι μεγάλες δυνατότητες επέκτασης της εφαρμογής αφού η εφαρμογή είναι μονάχα ένα παράδειγμα παρουσίασης των δυνατοτήτων που έχουν οι έξυπνες συσκευές. Μέσα από ανάλυση πολλών δεδομένων (big data) που μπορεί να κάνει ένας απομακρυσμένος υπολογιστής με μεγάλη υπολογιστική ισχύ και των δυνατοτήτων των συσκευών που χρησιμοποιούμε καθημερινά είναι εφικτό ένα μεγάλο βήμα για να πλησιάσουμε το όραμα της περιρρέουσας νοημοσύνης έτσι ώστε οι συσκευές γύρω μας να μας εξυπηρετούν χωρίς να είναι απαραίτητη οποιοδήποτε είδους διάδραση με αυτές και χωρίς να αντιλαμβανόμαστε την παρουσία τους.

# Summary

This diploma thesis employs the concept of ambient intelligence. There is in depth analysis of the detection of specific human movements by processing data obtained from the sensors of smart phones and comparing them with fuzzy models representing those movements.

In this context there was an application developed that recognizes the course of a smart phone user and stores it into a remote database. Additionally through modeling specific processes such as walking and driving by using sensors , the application is able to detect when and where the user parks his car and automatically stores those data to the remote database. These procedures do not require user to interact with the application's interface since they run as application's services on the background. The user may interact with the app in order to be informed, through roadmap, about the last parking location that was stored. Roadmap is implemented using googlemaps' libraries. The user may also open the remote database's table to check the past coordinates followed by a parking boolean variable of whether those are parking or course coordinates. The ability to view the table was provided for scientific and app's future expansion purposes.

# Περιεχόμενα

|  |    |
|--|----|
| Ευχαριστίες .....  | 3  |
| Περίληψη .....   | 5  |
| Summary .....  | 6  |
| Περιεχόμενα.....   | 7  |
| Κατάλογος εικόνων .....  | 11 |
| 1 Εισαγωγή.....  | 13 |
| 1.1 Αντικείμενο της εργασίας .....   | 13 |
| 1.2 Δομή.....  | 14 |
| 2 Περιρρέουσα Νοημοσύνη (Ambient Intelligence).....  | 15 |
| 2.1 Ορισμός.....   | 15 |
| 2.2 Διάχυτη πληροφορική (Pervasive Computing) και διάχυτος υπολογισμός (Ubiquitous computing).....                           | 17 |
| 2.3 Παραδείγματα πεδίων εφαρμογής διάχυτης πληροφορικής και διάχυτης υπολογιστικής (pervasive and ubiquitous computing)..... | 18 |
| 2.4 Ανθρωποκεντρικός σχεδιασμός (Human centered computing).....  | 20 |
| 2.5 Τομείς εφαρμογής της περιρρέουσας νοημοσύνης.....  | 20 |
| 2.6 Απαιτούμενες Τεχνολογίες .....   | 23 |
| 3 Έξυπνα κινητά τηλέφωνα: Αισθητήρες.....  | 25 |
| 3.1 Αναφορά στους αισθητήρες .....   | 26 |
| 3.2 Χρήση αισθητήρων για μοντελοποίηση ενεργειών και συνδυασμός τους για μεγαλύτερη αξιοπιστία .....                         | 27 |
| 3.2.1 Μοντελοποίηση ανθρωπίνων συμπεριφορών .....  | 28 |
| 3.2.2 Ασαφής Λογική (Fuzzy Logic).....   | 29 |
| 3.2.3 Τεχνική δημιουργίας προφίλ (profiling) και αναγνώρισης προτύπων (pattern recognition).....                             | 30 |
| 3.3 Εφαρμογές.....   | 33 |
| 4 Η αναπτυχθείσα εφαρμογή.....   | 35 |

|        |  |    |
|--------|--|----|
| 4.1    | Η διαδικασία .....   | 35 |
| 4.1.1  | Επιλογή λειτουργικού συστήματος.....   | 36 |
| 4.1.2  | Android: χαρακτηριστικά, προδιαγραφές, συστατικά στοιχεία.....   | 37 |
| 4.1.3  | Επιλογή περιβάλλοντος προγραμματισμού (eclipse) .....  | 42 |
| 4.1.4  | Γλώσσα προγραμματισμού – Java .....  | 44 |
| 4.1.5  | Η γλώσσα XML (Extensible Markup Language) Εκτεταμένη Γλώσσα Σήμανσης.....  | 45 |
| 4.1.6  | Αρχείο MANIFEST.xml.....   | 46 |
| 4.1.7  | Layouts : Διατάξεις – χωροταξία των αντικειμένων.....  | 48 |
| 4.1.8  | Οι εξυπηρετητές Ιστού (Web Servers) .....  | 53 |
| 4.1.9  | Επικοινωνία εξυπηρετητή ιστού με την αναπτυχθείσα εφαρμογή (PHP και JSON Parser) .....   | 54 |
| 4.1.10 | Χρήση πολλαπλών αισθητήρων στην εφαρμογή .....   | 56 |
| 4.1.11 | Ανίχνευση ανθρώπινου βαδίσματος μέσω προτύπων αναγνώρισης χρησιμοποιώντας το επιταχυνσιόμετρο του έξυπνου τηλεφώνου (Detect human walk through pattern recognition using smartphone accelerometer) ..... | 58 |
| 4.1.12 | Αναγνώριση οδήγησης (Driving recognition) .....  | 60 |
| 5      | Επιμέρους παρουσίαση τεχνικών λύσεων που χρησιμοποιήθηκαν.....   | 61 |
| 5.1    | Δήλωση των συστατικών στοιχείων της εφαρμογής .....  | 63 |
| 5.2    | Περιβάλλον διεπαφής χρήστη (user interface) και αρχεία xml .....   | 64 |
| 5.2.1  | Αρχεία xml .....   | 64 |
| 5.2.2  | Εισαγωγή των GOOGLE MAPS APIs (application programming interface) .....  | 67 |
| 5.3    | Ο διερμηνέας JSONParser.java που χρησιμοποιήθηκε καθώς και τα PHP αρχεία με τα οποία επικοινωνεί .....   | 70 |
| 5.3.1  | JSON PARSER.....   | 70 |
| 5.3.2  | PHP .....  | 72 |
| 6      | Παρουσίαση και επικύρωση της αναπτυχθείσας εφαρμογής.....  | 77 |
| 6.1    | Παρουσίαση εφαρμογής.....  | 77 |
| 6.2    | Επικύρωση εφαρμογής .....  | 82 |
| 7      | Συμπεράσματα .....   | 87 |
|        | Βιβλιογραφία .....   | 89 |



|                          |    |
|--------------------------|----|
| Παράρτημα 1.....         | 91 |
| Main_activity.java ..... | 91 |
| Παράρτημα 2.....         | 96 |
| maps.java .....          | 96 |
| Παράρτημα 3.....         | 99 |
| Testservice.java .....   | 99 |



## Κατάλογος εικόνων

|   |    |
|---|----|
| Εικόνα 1 : Περιρρέουσα νοημοσύνη .....  | 15 |
| Εικόνα 2 : Οι τρεις συνιστώσες της περιρρέουσας νοημοσύνης.....                                     | 16 |
| Εικόνα 3 : Πεδίο εφαρμογής έξυπνης σκόνης (smart dust).....   | 18 |
| Εικόνα 5 : Αισθητήρες έξυπνων κινητών τηλεφώνων .....   | 26 |
| Εικόνα 6 : Η εφαρμογή Ambient .....   | 33 |
| Εικόνα 7 : Η εφαρμογή Runkeeper.....  | 34 |
| Εικόνα 8 : Η εφαρμογή Placeme .....   | 34 |
| Εικόνα 9 : Η αρχιτεκτονική του Android.....   | 37 |
| Εικόνα 10 : Ο κύκλος ζωής μιας Activity.....  | 39 |
| Εικόνα 12 : Android Virtual Device μέσα από το SDK του Eclipse .....                                | 41 |
| Εικόνα 13 : : Διαχείριση διαθέσιμων αναβαθμίσεων του Eclipse .....                                  | 43 |
| Εικόνα 14 : Το διαχειριστικό περιβάλλον του Eclipse .....   | 43 |
| Εικόνα 15 : Η δομή των αρχείων μιας εφαρμογής Android .....   | 52 |
| Εικόνα 16 : Η διαδικασία σε οπτική μορφή.....   | 54 |
| Εικόνα 17 : Οπτική αναπαράσταση της διαδικασίας ανίχνευσης ανθρώπινου βαδίσματος με αισθητήρες..... | 58 |
| Εικόνα 18 : Εφαρμογή δημιουργίας γραφημάτων με επιταχυνσιόμετρο .....                               | 59 |
| Εικόνα 19 : ο φάκελος της εφαρμογής μέσα από το περιβάλλον του eclipse.....                         | 61 |
| Εικόνα 20 : Ανάλυση των αρχείων που περιέχονται στον φάκελο της εφαρμογής .....                     | 62 |
| Εικόνα 21 : Λήψη google play services.....  | 67 |
| Εικόνα 22 : Εισαγωγή google play services στο workspace του eclipse.....                            | 68 |
| Εικόνα 23 : Προσθήκη googleplay services ως βιβλιοθήκη .....  | 68 |
| Εικόνα 24 : Δήλωση Sha-1 fingerprint στο google api console .....                                   | 69 |
| Εικόνα 25 : Το στίγμα της τελευταίας θέσης στάθμευσης του αυτοκινήτου στο googlemaps.<br>.....      | 69 |
| Εικόνα 26 : Η οθόνη της εφαρμογής στο κινητό .....  | 77 |

|  |    |
|--|----|
| Εικόνα 27 : Εκτέλεση της εφαρμογής.....                                      | 78 |
| Εικόνα 28 : Η οθόνη της επιλογής ΧΑΡΤΗΣ.....                                 | 79 |
| Εικόνα 29 : Η οθόνη του rhrmyadmin στον απομακρυσμένο εξυπηρετητή ιστού..... | 80 |
| Εικόνα 30 : Τα δεδομένα του πίνακα mobile2_data της βάσης android2.....      | 81 |
| Εικόνα 31 : Στίγμα θέσης στάθμευσης του αυτοκινήτου .....                    | 82 |
| Εικόνα 32 : Στίγμα θέσης στάθμευσης του αυτοκινήτου .....                    | 83 |
| Εικόνα 33 : Στίγμα θέσης στάθμευσης του αυτοκινήτου .....                    | 84 |
| Εικόνα 34 : Τα δεδομένα του πίνακα της βάσης μας .....                       | 85 |

# 1 Εισαγωγή

## 1.1 Αντικείμενο της εργασίας

Σύμφωνα με τον Mark Weiser, τον πατέρα της πανταχού παρούσας υπολογιστικής, οι πιο ισχυρές τεχνολογίες είναι αυτές που δεν γίνονται αντιληπτές. Υφαίνονται μέσα στον ιστό της καθημερινής μας ζωής σε σημείο που δεν μπορούμε πλέον να τις ξεχωρίσουμε από αυτήν. Αυτά τα λόγια αποτέλεσαν πηγή έμπνευσης για μια γενιά νέων ερευνητών στον τομέα της πανταχού παρούσας υπολογιστικής και οδήγησαν στο όραμα της Περιρρέουσας Νοημοσύνης όπου οι τεχνολογίες πληροφορικής και επικοινωνιών έχουν ενσωματωθεί τόσο τέλεια στο φυσικό περιβάλλον που δεν γίνονται αντιληπτές από τον άνθρωπο .

Στη σύγχρονη εποχή, οι πρωτοποριακές, ανθρωποκεντρικές τεχνολογίες της Περιρρέουσας Νοημοσύνης έχουν τη δυνατότητα να δημιουργήσουν «έξυπνα» περιβάλλοντα, που να είναι ευπροσάρμοστα για τον άνθρωπο, έτσι ώστε να εξυπηρετήσουν τις ανάγκες και τους στόχους του, σε διάφορες δραστηριότητες της καθημερινής του ζωής. Αυτά τα «έξυπνα» ψηφιακά περιβάλλοντα, έχουν την ικανότητα να διακρίνουν την ανθρώπινη παρουσία και την προσωπικότητα, ίσως και την ψυχολογική διάθεση, να προσαρμόζονται στις στιγμιαίες και διαρκείς ανάγκες του χρήστη, να ανταποκρίνονται έξυπνα στις χειρονομίες ή τις φωνητικές εντολές του και να είναι ευχάριστα και χαλαρωτικά, χωρίς να κουράζουν με πολύπλοκες διαδικασίες εκμάθησης της λειτουργίας τους. Στα περιβάλλοντα αυτά θεωρείται δεδομένη η δυνατότητα πρόσβασης σε πηγές πληροφοριών και η ανταλλαγή τους οπότε η επικοινωνία ανάγεται σε αγαθό, η διάθεση του οποίου πρέπει να υποστηρίζεται σε διαρκή βάση.

Οι τεχνολογίες περιρρέουσας νοημοσύνης ενσωματώνουν δυνατότητες ανίχνευσης, επεξεργαστική ισχύ, μηχανισμούς συλλογισμού, εγκαταστάσεις δικτύων, εφαρμογές και υπηρεσίες ψηφιακού περιεχομένου κ.α.

Στόχος της περιρρέουσας νοημοσύνης είναι είτε να κρύψει εξ ολοκλήρου από τους χρήστες την παρουσία της ευρείας ποικιλίας των διαφορετικών τεχνολογιών που εμπλέκονται ή να τις εντάξει ομαλά ως αντικείμενα του βελτιωμένου περιβάλλοντος, και όχι ως τεχνολογικά gadgets. Βασικό της όραμα είναι να βελτιώσει την ποιότητα της ζωής των ανθρώπων μέσα από τη δημιουργία και την παροχή ασφαλών, αποδοτικών και φιλικών προς το χρήστη τεχνολογιών, οι οποίες αναμένεται ότι θα καλύψουν τις ανάγκες του με έναν ενιαίο, διακριτικό και αόρατο τρόπο.

Η παρούσα διπλωματική εργασία ασχολείται με το πεδίο της Περιρρέουσας Νοημοσύνης με σκοπό να αναδείξει τη χρησιμότητά της στην διευκόλυνση του ανθρώπου στην καθημερινή του ζωή και στη δημιουργία προτύπων ανθρώπινης συμπεριφοράς. Υλοποιήθηκε δε μια εφαρμογή συλλογής δεδομένων για τη διαδρομή που ακολουθεί κάποιος χρήστης έξυπνου κινητού τηλεφώνου από τη στιγμή που φεύγει από το σπίτι του μέχρι την ώρα που θα σταθμεύσει χρησιμοποιώντας μόνο τα στοιχεία που προέρχονται από αισθητήρες του κινητού του. Σκοπός και στόχος αυτής της εφαρμογής είναι να γίνει πιο

κατανοητή η έννοια της Περιρρέουσας Νοημοσύνης και της τεχνολογίας που χρησιμοποιείται χάριν αυτής, προκειμένου να παρέχει σημαντικές πληροφορίες οι οποίες θα βοηθήσουν σε διάφορους τομείς.

Η σκέψη επάνω στην οποία στηρίχτηκε και υλοποιήθηκε η συγκεκριμένη εφαρμογή είναι η δημιουργία μιας μεγάλης βάσης δεδομένων όπου θα αποθηκεύονται διαρκώς τα δεδομένα από τους αισθητήρες του κινητού. Η αξιοποίηση αυτής της βάσης δεδομένων, θα βοηθήσει στη λύση προβλημάτων εύρεσης χώρων στάθμευσης, στις συγκοινωνίες, στην κυκλοφορία των αυτοκινήτων και άλλα.

## 1.2 Δομή

Συνοπτικά, η δομή της παρούσας διπλωματικής εργασίας είναι η εξής:

- **Κεφάλαιο 1: Εισαγωγή.** Εισαγωγή του αναγνώστη στο θέμα που πραγματεύεται η διπλωματική αυτή εργασία και παρουσίαση της δομής αυτής.
- **Κεφάλαιο 2: Περιρρέουσα Νοημοσύνη.** Αφού γίνει αντιληπτή από τον αναγνώστη η σημασία της περιρρέουσας νοημοσύνης στο σύγχρονο κόσμο και δοθούν κάποιοι απαραίτητοι για την κατανόησή της αρχικοί ορισμοί, παρουσιάζονται τα περιβάλλοντα και η τεχνολογία υλοποίησης της και τα πεδία εφαρμογής της .
- **Κεφάλαιο 3: Έξυπνα κινητά τηλέφωνα : Αισθητήρες.** Αναφορά της τεχνολογίας των αισθητήρων και της χρήσης τους στην μοντελοποίηση ανθρωπίνων συμπεριφορών και στην τεχνική δημιουργίας προφίλ. Παραδείγματα εφαρμογών.
- **Κεφάλαιο 4: Η αναπτυχθείσα εφαρμογή.** Περιγραφή της αναπτυχθείσας εφαρμογής, των απαραίτητων στοιχείων και της διαδικασίας που ακολουθήθηκε κατά την υλοποίηση της.
- **Κεφάλαιο 5: Επιμέρους παρουσίαση των τεχνικών λύσεων που χρησιμοποιήθηκαν.** Αναφορά των χαρακτηριστικών στοιχείων των προδιαγραφών και των αρχείων της εφαρμογής.
- **Κεφάλαιο 6: Παρουσίαση και επικύρωση της αναπτυχθείσας εφαρμογής.** Παρουσίαση των οθονών της εφαρμογής και των αποτελεσμάτων λειτουργίας της.
- **Κεφάλαιο 7: Συμπεράσματα.** Τα συμπεράσματα που αντλούνται από αυτήν την εφαρμογή.

# 2 Περιρρέουσα Νοημοσύνη (Ambient Intelligence)



Εικόνα 1 : Περιρρέουσα νοημοσύνη

## 2.1 Ορισμός

Η έννοια της περιρρέουσας νοημοσύνης αναφέρεται στο όραμα, της δεκαετίας του 90, της εξέλιξης των τεχνολογιών της κοινωνίας της πληροφορίας με στόχο την βελτίωση της ποιότητας της ζωής των ανθρώπων. Η ολοένα αυξανόμενη ζήτηση για συνεχή και απανταχού διαθέσιμη πρόσβαση σε πληροφορίες και ηλεκτρονικές υπηρεσίες δημιούργησε την ανάγκη σύλληψης νέου τρόπου αλληλεπίδρασης ανθρώπου – υπολογιστή. Αυτό έχει σαν αποτέλεσμα αντί να προσαρμόζεται ο άνθρωπος στο περιβάλλον του να προσαρμόζεται το περιβάλλον στις ανάγκες του ανθρώπου.

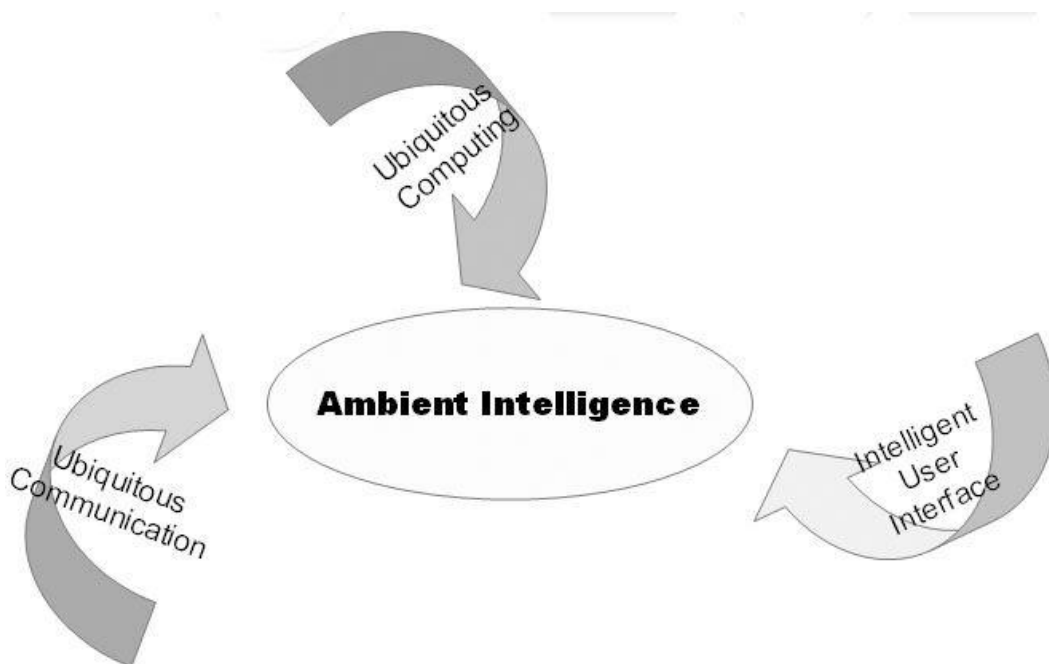
Σήμερα οι άνθρωποι περιβάλλονται από φυσικές ευφυείς διεπαφές χρήσης (intelligent intuitive interfaces), που είναι ενσωματωμένες σε όλα τα καθημερινά αντικείμενα. Το περιβάλλον αυτό είναι σε θέση να αναγνωρίζει την παρουσία κάθε ανθρώπου, να προσαρμόζεται σ' αυτήν και το πιο σημαντικό είναι ότι οι άνθρωποι δε συνειδητοποιούν ότι βρίσκονται σε ένα υπολογιστικό περιβάλλον επειδή είναι συνήθως «αόρατο». Δηλαδή βασικός στόχος είναι η απόκρυψη της τεχνολογίας και όχι η ανάδειξή της μέχρι τελικά να

«εξαφανιστεί». Δίνεται έμφαση δηλαδή στην ευκολία χρήσης και στην υποστήριξη και ενδυνάμωση της αυτόματης αλληλεπίδρασης των ανθρώπων με το περιβάλλον.

Ένα περιβάλλον περιρρέουσας νοημοσύνης πρέπει να περιέχει τα παρακάτω βασικά χαρακτηριστικά :

- **Εμφύτευση (Embedded)** : Πολλές συσκευές είναι διασκορπισμένες στο περιβάλλον χωρίς να φαίνονται.
- **Επίγνωση Περιεχομένου (Context aware)** : Το περιβάλλον έχει τη δυνατότητα να αναγνωρίζει το εκάστοτε πρόσωπο και την κατάστασή του.
- **Διάχυση (Pervasive)** : Η αλληλεπίδραση με το σύστημα δεν γίνεται μόνον από έναν σταθμό εργασίας αλλά και από επιμέρους συσκευές.
- **Εξατομίκευση (Personalized)** : Το περιβάλλον μπορεί να συμπεριφέρεται ανάλογα με το ποιός χρήστης εμφανίζεται κάθε φορά.
- **Προσαρμοστικότητα (Adaptive)** : Το περιβάλλον αλλάζει ανάλογα με τις ανάγκες του κάθε χρήστη.
- **Προνοητικότητα (Anticipatory)** : Το περιβάλλον έχει τη δυνατότητα να προνοεί σχετικά με τις επιθυμίες του κάθε χρήστη.

Οι εφαρμογές των smartphones και των tablets είναι ένα κομμάτι από τα θεμέλια επίτευξης αυτού του οράματος μέσω της διάχυτης υπολογιστικής (π.χ. εφαρμογές πλοήγησης ή φωνητικής αναγνώρισης). Η περιρρέουσα νοημοσύνη δομείται πάνω στη σύγκλιση τριών τεχνολογιών: αυτή του πανταχού παρόντα υπολογιστή (Ubiquitous Computing), της παν-επικοινωνίας (Ubiquitous Communication), και της ανάπτυξης έξυπνων και φιλικών διεπαφών χρήστη (Intelligent User Friendly Interfaces).



**Εικόνα 2 : Οι τρεις συνιστώσες της περιρρέουσας νοημοσύνης.**



## 2.2 Διάχυτη πληροφορική (Pervasive Computing) και διάχυτος υπολογισμός (Ubiquitous computing)

Οι έννοιες ubiquitous και pervasive computing συγχέονται και στα ελληνικά αναφέρονται και οι δύο συχνά ως διάχυτη υπολογιστική, πληροφορική ή υπολογισμός ενώ για ακρίβεια pervasive σημαίνει διεισδυτικός, διαπεραστικός και ubiquitous σημαίνει πανταχού παρόν.

Στο περιβάλλον διάχυτης πληροφορικής (Pervasive Computing), μια κατάλληλα διαμορφωμένη συσκευή μπορεί να χρησιμοποιηθεί εύκολα ως πύλη για τον χρήστη της για το πέρασμα σε έναν χώρο με χρήσιμες πληροφορίες και εφαρμογές και όχι ως μια «αποθήκη» λογισμικού την οποία διαχειρίζεται ο χρήστης. Όσον αφορά την εφαρμογή, αυτή είναι το μέσο με το οποίο ο χρήστης διεκπεραιώνει μια εργασία και όχι κάποιο είδος λογισμικού, χωρίς το οποίο δεν μπορούν να αξιοποιηθούν οι δυνατότητες μιας συσκευής.

Υπάρχουν συσκευές με την ικανότητα να καταγράφουν τα χαρακτηριστικά του περιβάλλοντος στο οποίο βρίσκεται ο χρήστης και να παρέχουν στα “ubiquitous και pervasive” συστήματα που είναι συνδεδεμένες, τις σχετικές πληροφορίες. Με τη σειρά τους, τα συστήματα αυτά τις αξιοποιούν με σκοπό την ομαλή αλληλεπίδραση με τους χρήστες αλλά και τις υπόλοιπες συσκευές που είναι συνδεδεμένες.

Μέσω του περιβάλλοντος της **διάχυτης πληροφορικής (Pervasive Computing)**, οι χρήστες μπορούν να βρίσκονται πλέον σε κίνηση, οι συσκευές πληροφοριών αποκτούν περισσότερες λειτουργίες, και η επικοινωνία γίνεται ευκολότερη, μεταξύ ανθρώπων και αντικειμένων, καθώς και αντικειμένων μεταξύ τους. (Ark and Selker 1999)

Η έννοια της **διάχυτης υπολογιστικής (ubiquitous computing)** αναφέρεται στη σύγκλιση και σύνθεση των τεχνολογιών πληροφορικής και επικοινωνιών (ΤΠΕ) σε ένα ενιαίο περιβάλλον που αλλάζει την καθημερινότητά μας. Οι άνθρωποι περιβάλλονται από αντικείμενα, συσκευές κλπ, που διαθέτουν ενσωματωμένη υπολογιστική ικανότητα, συνθέτοντας το όραμα για “περιρρέουσα νοημοσύνη” (ambient intelligence). Η πρόσβαση στο περιβάλλον αυτό πραγματοποιείται μέσω φυσικών διεπαφών χρήστη (intuitive user interfaces), συχνά χωρίς ο χρήστης να συνειδητοποιεί ότι αλληλεπιδρά με μια υπολογιστική εφαρμογή.

Στην υλοποίηση των συστημάτων διάχυτου υπολογισμού θα πρέπει να λαμβάνονται οι απαραίτητες ενέργειες για την προστασία προσωπικών δεδομένων στον βέλτιστο εφικτό βαθμό εφόσον είναι επιθυμητή η ύπαρξη ιδιωτικότητας αλλιώς θα καταργηθεί. Αυτό είναι πιθανό επειδή μειώνεται η αποτελεσματικότητα των φυσικών, ψυχολογικών και κοινωνικών μηχανισμών που συνήθως χρησιμοποιούμε για να ρυθμίσουμε την “ιδιωτικότητά” μας (Palen, 2003).

Η αξιοπιστία των συστημάτων διάχυτου υπολογισμού είναι ένα μείζον ζήτημα, απαραίτητο για την υλοποίηση του οράματος της περιρρέουσας νοημοσύνης. Η αξιοπιστία αυτή είναι ένα πεδίο συνεχώς εξελισσόμενο, το οποίο περιλαμβάνει ποσοστά ακρίβειας του συστήματος. Για να είναι ικανοποιητικά αυτά τα ποσοστά είναι απαραίτητη η συνεχής και αυτόνομη λειτουργία, η συνεχής πρόσβαση σε δεδομένα, η σωστή ταυτοποίηση κ.α.

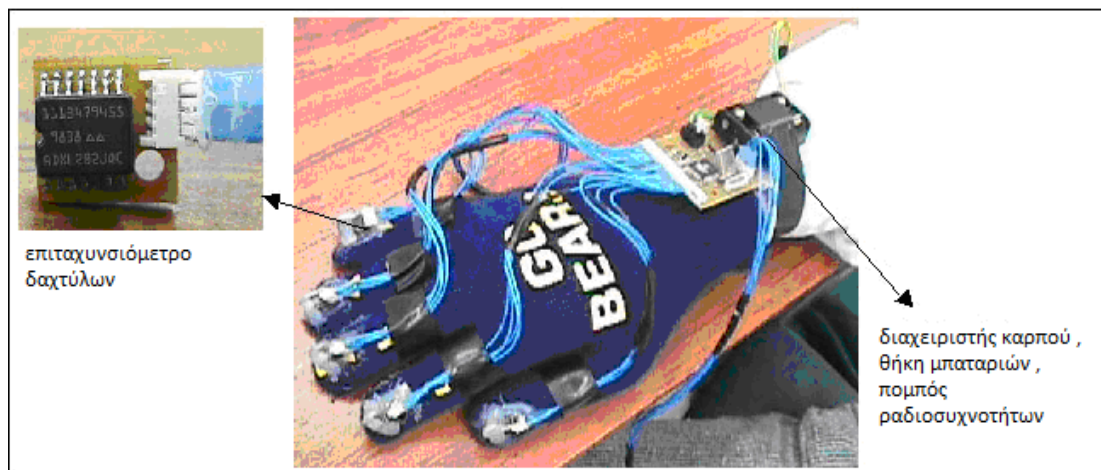
Η ολοκληρωμένη υλοποίηση του περιβάλλοντος της περιρρέουσας νοημοσύνης απαιτεί χρήση μεσολογισμικού που βοηθά μια εφαρμογή να επικοινωνήσει με άλλες εφαρμογές, δίκτυα, υλικό και λειτουργικά συστήματα και αναμένεται να αλλάξει ριζικά τον τρόπο με τον οποίο επικοινωνούμε, εκπαιδευόμαστε, εργαζόμαστε, ψυχαγωγούμαστε, κλπ. (Weiser and Brown 1996; Weiser 2001).

## 2.3 Παραδείγματα πεδίων εφαρμογής διάχυτης πληροφορικής και διάχυτης υπολογιστικής (pervasive and ubiquitous computing)

Τρία χαρακτηριστικά πεδία εφαρμογής της έρευνας που διεξάγεται στα συστήματα διάχυτου υπολογισμού και πληροφορικής είναι η σχεδόν αόρατη "έξυπνη σκόνη" (smart dust), οι πληροφορικές συσκευές (information appliances) κατά τον D. Norman (Norman, 1999) που αντικαθιστούν σταδιακά τα σημερινά αντικείμενα του χώρου γύρω μας, και οι φορητοί υπολογιστές (wearable computers).

Ως έξυπνη σκόνη (smart dust) χαρακτηρίζεται ένα σύστημα διάχυτης υπολογιστικής που αποτελείται από ένα τεράστιο αριθμό πάρα πολύ μικρών, εντελώς αυτόνομων συσκευών, με δυνατότητες αίσθησης, υπολογισμού και επικοινωνίας, οι οποίες έχουν την ικανότητα να αυτό-οργανώνονται και να συνεργάζονται για να επιτελέσουν ένα έργο αναγνώρισης του περιβάλλοντος.

Παραδείγματα εφαρμογής της τεχνολογίας αυτής είναι η χαρτογράφηση περιοχής, η καταγραφή σύνθεσης εδάφους, αέρα, κλπ, άγνωστης ή επικίνδυνης περιοχής, η πυροπροστασία, ο εντοπισμός ζωής (π.χ. στα ερείπια ενός σεισμού) και εντοπισμός διαρροών σε άγνωστα, αφιλόξενα, ανεξερεύνητα ή δυναμικά μεταβαλλόμενα περιβάλλοντα.



Εικόνα 3 : Πεδίο εφαρμογής έξυπνης σκόνης (smart dust)

Σύμφωνα με τον Gordon Moore, συνιδρυτή της Intel και γνωστό για τον 'νόμο' του( ο αριθμός των κρυσταλλοτριόδων (transistors) σε ένα πυκνό ολοκληρωμένο κύκλωμα διπλασιάζεται κάθε περίπου δύο χρόνια), **πληροφορικές συσκευές** (information appliances) είναι μια ομάδα συσκευών καθεμία από τις οποίες είναι σχεδιασμένη για να υποστηρίξει συγκεκριμένες εφαρμογές του τελικού χρήστη μέσω ενσωμάτωσης υπολογιστικής ισχύος σε καθημερινά αντικείμενα και δημιουργίας "έξυπνων τεχνουργημάτων" (smart artifacts).

Στην πραγματικότητα, χρησιμοποιούμε ήδη πολλές πληροφορικές συσκευές, χωρίς ίσως να το γνωρίζουμε: κατάλογοι διευθύνσεων και ψηφιακά ημερολόγια, ψηφιακοί βοηθοί, συστήματα πλοήγησης στα αυτοκίνητα, ψηφιακές κάμερες και κινητά τηλέφωνα, ψηφιακές ιατρικές συσκευές, πλατφόρμες δικτυακών παιχνιδιών, αλλά και ηλεκτρικές συσκευές, όπως τηλεοράσεις, ψυγεία και φούρνοι. Οι περισσότερες από τις συσκευές αυτές προς το παρόν ενσωματώνουν τη δυνατότητα επεξεργασίας. Λίγες έχουν δυνατότητες δικτύωσης, και ακόμη λιγότερες αντιλαμβάνονται το περιβάλλον λειτουργίας ώστε να μπορούν να προσαρμοστούν σε αυτό. Η διάδραση με το χρήστη είναι δύσκαμπτη, η προστασία ανύπαρκτη και η αξιοπιστία μικρή.



Ένα χαρακτηριστικό παράδειγμα της έννοιας της διάχυτης υπολογιστικής είναι η έννοια των **φορετών υπολογιστών** (wearable computers), οι οποίοι είναι συνήθως ενσωματωμένοι στα ρούχα μας, ή μπορεί να τους φέρουμε μαζί μας όπως ένα ρολόι, ή να είναι ενσωματωμένοι σε αντίστοιχα αντικείμενα, όπως κινητό τηλέφωνο, κλπ.

**Εικόνα 4 : Φορετός υπολογιστής καρπού**

Οι εφαρμογές των φορετών υπολογιστών είναι επίσης αμέτρητες: ο φορετός υπολογιστής είναι πάντα διαθέσιμος (always on, always accessible). Καταρχήν, η χρήση τους είναι ιδανική για περιπτώσεις όπου ο χρήστης δεν έχει ελεύθερα τα χέρια του, τα μάτια του, κλπ. Για παράδειγμα, ένας μηχανικός μπορεί να έχει πρόσβαση σε πληροφορίες την ώρα που επισκευάζει ένα μηχάνημα. Ένας στρατιωτικός μπορεί επίσης να λαμβάνει χρήσιμες πληροφορίες την ώρα που έχει στραμμένη την προσοχή του αλλού. Πολύ σημαντικές είναι οι εφαρμογές σε άτομα με ειδικές ανάγκες, όπως για παράδειγμα η καθοδήγηση ατόμων με προβλήματα όρασης, καθώς και η συνεχής ιατρική παρακολούθηση από απόσταση.

## 2.4 Ανθρωποκεντρικός σχεδιασμός (Human centered computing)

Ανθρωποκεντρικός σχεδιασμός είναι ένα σύνολο μεθοδολογιών που εφαρμόζονται σε κάθε κλάδο την πληροφορικής όπου οι άνθρωποι επικοινωνούν άμεσα με συσκευές και συστήματα πληροφορικής. Η έρευνα στο πεδίο αυτό έχει διάφορους στόχους. Ένας από αυτούς είναι η κατανόηση του ανθρώπου σαν ξεχωριστή οντότητα αλλά και σαν μέρος ενός κοινωνικού συνόλου, εστιάζοντας στο πώς ο άνθρωπος προσαρμόζει και οργανώνει την ζωή του γύρω από τις υπολογιστικές τεχνολογίες. Άλλος στόχος είναι η ανάπτυξη στρατηγικών σχεδιασμού υπολογιστικών εφαρμογών. Ο ανθρωποκεντρικός σχεδιασμός ασχολείται με προβλήματα όπως η μέτρηση της παραγωγικότητας και της αποδοτικότητας τα οποία οι παραδοσιακές τεχνικές σχεδιασμού δεν εξετάζουν.

Η δημιουργία αποτελεσματικών τεχνικών επικοινωνίας ανθρώπου υπολογιστή κάθε στιγμή μέσα από την επεξεργασία πολλών δεδομένων με χρήση υπολογιστικής ισχύος είναι άμεσος στόχος του ανθρωποκεντρικού σχεδιασμού. Συνεπώς μέσω αυτού ερχόμαστε ένα βήμα πιο κοντά στο όραμα της περιρρέουσας νοημοσύνης.

## 2.5 Τομείς εφαρμογής της περιρρέουσας νοημοσύνης

### Υγεία

Η Περιρρέουσα Νοημοσύνη στο πλαίσιο της υγειονομικής περίθαλψης αποσκοπεί στη συγχώνευση της πανταχού παρούσα υπολογιστικής, τις κοινωνικές διεπαφές χρηστών και υπηρεσίες e-υγείας, προκειμένου να παρέχει μια ευέλικτη, δυναμική, έξυπνη τεχνολογική υποδομή για την πανταχού παρούσα διαχείριση της κατάστασης της υγείας. Ο κύριος στόχος είναι να συνδυάσει τις τεχνολογίες της πληροφορίας και της επικοινωνίας, προκειμένου να συντονίσει και να υποστηρίξει τις ασφαλείς και υψηλής ποιότητας υπηρεσίες φροντίδας υγείας σε ένα πλούσιο σε πληροφορίες, εξυπηρετικό και εύκολα προσβάσιμο νοσοκομειακό περιβάλλον, αλλά και στο σπίτι.

Για παράδειγμα, αισθητήρες ενσωματωμένοι σε υφάσματα (π.χ. εσώρουχα) μπορούν να καταγράφουν σημαντικά βιολογικά στοιχεία υγείας, όπως καρδιακός παλμός, αρτηριακή πίεση και θερμοκρασία σώματος. Όταν τα στοιχεία αυτά ξεπερνούν κάποια ασφαλή όρια, τότε οι αισθητήρες και οι έξυπνες συσκευές αντιδρούν ανάλογα με κάποιον καθορισμένο τρόπο, π.χ. αποστολή μηνύματος ασύρματα. Τέτοια συστήματα παρακολούθησης είναι πολύ χρήσιμα σε νοσοκομεία όπου η συνεχής παρακολούθηση της υγείας του ασθενούς και η προειδοποίηση για οποιαδήποτε παρέκκλιση από καθορισμένες βιολογικές παραμέτρους, είναι πρωταρχικής σημασίας.

## **Σπίτι**

Στο περιβάλλον του σπιτιού, οι τεχνολογίες Περιρρέουσας Νοημοσύνης επιτρέπουν την αυτοματοποίηση ή υποστηρίζουν δραστηριότητες της καθημερινής ζωής για τους κατοίκους, καθώς και την ασφάλεια, την υγεία, την επικοινωνία, κ.α. Οι ερευνητικές προσπάθειες που πραγματοποιούνται στον τομέα αυτό αφορούν τη διευκόλυνση της διαχείρισης των οικιακών συσκευών μέσω διεπαφών προσαρμοσμένων στον χρήστη που μπορεί να είναι πανταχού παρόν μέσα στο περιβάλλον, καθώς και εφαρμογών και υπηρεσιών για «σπίτια μακριά από το σπίτι».

Για παράδειγμα, σε ένα Έξυπνο Σπίτι όταν οι αισθητήρες κίνησης και σώματος που είναι ενσωματωμένοι στους τοίχους και στο πάτωμα αντιληφθούν ότι κάποιιο άτομο δεν περπατά ούτε στέκει στα πόδια του αλλά είναι ξαπλωμένο στο πάτωμα, τότε εκπέμπεται σήμα κινδύνου. Τέτοια συστήματα μπορούν να χρησιμοποιηθούν σε χώρους όπου ζουν άτομα με προδιάθεση να χάνουν την ισορροπία τους, όπως οι ηλικιωμένοι.

## **Τέχνες & Πολιτισμός**

Στον τομέα του πολιτισμού, και σε χώρους όπως εκθέσεις και μουσεία, η Περιρρέουσα Νοημοσύνη προσφέρει στους χρήστες μια νέα αλληλεπιδραστική εμπειρία που συνδυάζει την πληροφόρηση και τη μάθηση με το παιχνίδι. Τα Διαδραστικά συστήματα που χρησιμοποιούνται είναι μικρές ή μεγάλες κατασκευές με ενσωματωμένους υπολογιστές που δίνουν τη δυνατότητα στους επισκέπτες των Μουσείων να εξερευνήσουν ψηφιακές αναπαραστάσεις αρχαίων αριστουργημάτων, όπου η αλληλεπίδραση γίνεται με απλό, «φυσικό» και «αυθόρμητο» τρόπο, χωρίς χρήση πληκτρολογίου ή ποντικιού.

Στον τομέα της τέχνης, διερευνά νέα μέσα για τη δημιουργία τέχνης μέσω της ενσωμάτωσης των έξυπνων διαδραστικών τεχνολογιών στην πραγματικότητα και δημιουργεί νέες αναζωογονητικές εμπειρίες καλλιτεχνικής δημιουργίας.

## **Εμπόριο & Μάρκετινγκ**

Η Περιρρέουσα Νοημοσύνη παρέχει ένα νέο μέσο προώθησης και διαφήμισης πέρα από τις συμβατικές μεθόδους. Το κοινό δεν είναι πια ο παθητικός αποδέκτης ενός μηνύματος μάρκετινγκ. Τα Περιβάλλοντα Περιρρέουσας Νοημοσύνης καλούν τον χρήστη να συμμετέχει ενεργά σε νέες διαδραστικές εμπειρίες. Οι χρήστες εξουσιοδοτούνται έτσι να κάνουν τις επιλογές τους και να αποκτήσουν το περιεχόμενο που είναι εξατομικευμένο ανάλογα με τις ανάγκες τους.

## **Μάθηση και Εκπαίδευση**

Οι τεχνολογίες περιρρέουσας νοημοσύνης μπορούν να διαδραματίσουν σημαντικό ρόλο στον εμπλουτισμό της εμπειρίας της εκπαίδευσης. Οι τεχνολογίες αυτές προσφέρουν στους σπουδαστές την αυξημένη πρόσβαση σε πληροφορίες μέσα σε ένα διδακτικό περιβάλλον που ενθαρρύνει την ενεργό μάθηση και τη συνεργασία, ενισχύοντας τα κίνητρά τους για να μάθουν. Το ερευνητικό έργο στον τομέα αυτό περιλαμβάνει τον μαθητοκεντρικό σχεδιασμό και την εφαρμογή των τεχνολογιών υποδομής, πρωτότυπα ευφυή συστήματα και εφαρμογές, έξυπνα αντικείμενα μάθησης και σοβαρά παιχνίδια.

## **Ελεύθερος χρόνος & Διασκέδαση**

Τα παιχνίδια και άλλες εφαρμογές διασκέδασης στα περιβάλλοντα της περιρρέουσας νοημοσύνης είναι διασκεδαστικά και ελκυστικά για τους ανθρώπους όλων των ηλικιών, προσφέροντας αξέχαστες εμπειρίες για τους παίκτες, και επεκτείνουν το σκοπό τους σε πολλούς άλλους τομείς, όπως η περιήγηση πληροφοριών, τη μάθηση, τη διαφήμιση, κ.λπ. Μια ποικιλία από παιχνίδια του περιβάλλοντος και τα συστήματα ψυχαγωγίας έχουν αναπτυχθεί και αναπτύσσονται σε συνθήκες πραγματικής ζωής, με πρωτοφανή επιτυχία από τους παίκτες και τους επισκέπτες.

## **Εργασία**

Η περιρρέουσα νοημοσύνη έχει διεισδύσει στον χώρο εργασίας, καθώς επιτρέπει τη βελτίωση της ποιότητας της εργασίας και μια καλύτερη ισορροπία εργασίας - ζωής μέσω εύκολης πρόσβασης στις Τεχνολογίες Πληροφορίας και Επικοινωνίας, καινοτόμων πρακτικών προσαρμογής των χώρων εργασίας καθώς και των δεξιοτήτων στις Τεχνολογίες Πληροφορίας και Επικοινωνίας. Επιπλέον, έξυπνα περιβάλλοντα παρέχουν απομακρυσμένη πρόσβαση και την αύξηση της συνεργασίας. Η έρευνα σε αυτόν τον τομέα εφαρμόζει καινοτόμες τεχνικές αλληλεπίδρασης για να υποστηρίξει φυσικά και με πολλούς τρόπους αλληλεπίδραση στο περιβάλλον του γραφείου, βελτιώνοντας ριζικά τη αποδοτικότητα και τη συνολική παραγωγικότητα του χρήστη.

## **Τεχνολογία**

Τα περιβάλλοντα Διάχυτης Νοημοσύνης χαρακτηρίζονται, μεταξύ άλλων, από την συμβολή των διαφορετικών υπολογιστικών πλατφορμών, τη διακριτική παρακολούθηση και αίσθηση (για παράδειγμα, μέσω της όρασης υπολογιστών), και από τις οπτικές αναπαραστάσεις των πληροφοριών. Οι υπηρεσίες αυτές αναπτύσσονται ως τεχνολογικά στοιχεία που υποστηρίζουν την ανάπτυξη των εφαρμογών της περιρρέουσας νοημοσύνης, υπηρεσιών και περιβαλλόντων.

## **Γεωργία - Κτηνοτροφία**

Η εφαρμογή της περιρρέουσας νοημοσύνης σε καλλιεργήσιμες εκτάσεις με τοποθέτηση αισθητήρων μέτρησης υγρασίας και ανάλυσης της σύστασης του εδάφους επιτρέπει την ακριβέστερη και αποδοτικότερη λίπανση και άρδευση των εκτάσεων.

Επίσης, η εκτροφή ζώων μπορεί να ωφεληθεί τοποθετώντας αισθητήρες στα ζώα που ελέγχουν την κατάσταση της υγείας τους.

## **Συγκοινωνία**

Στα ευφυή οδικά συστήματα αισθητήρες τοποθετούνται στους δρόμους, ακόμα και στα κράσπεδα των δρόμων οι οποίοι συλλέγουν πληροφορίες για την κίνηση και την κατάσταση του οδικού δικτύου γενικότερα και επικοινωνούν με τους οδηγούς δίνοντάς τους χρήσιμες πληροφορίες.

## Ένοπλες Δυνάμεις

Κάποιες από τις πιθανές στρατιωτικές εφαρμογές της περιρρέουσας νοημοσύνης είναι η παρακολούθηση της κατάστασης των εξοπλισμών και των πολεμοφοδίων, η στενή παρακολούθηση του πεδίου της μάχης, η αναγνώριση των εχθρικών δυνάμεων, η εκτίμηση των καταστροφών μετά από μάχη καθώς και ο εντοπισμός και η αναγνώριση χημικής, ατομικής ή βιολογικής επίθεσης.

## 2.6 Απαιτούμενες Τεχνολογίες

Για να γίνει πραγματικότητα ένα περιβάλλον περιρρέουσας νοημοσύνης, απαραίτητες είναι οι εξής τεχνολογίες :

- Τεχνολογίες πρόσβασης (κινητά τηλέφωνα, Laptops, προσωπικοί ψηφιακοί βοηθοί P.D.As κ.α.)
- Τεχνολογίες πρόσληψης και αντίληψης του φυσικού περιβάλλοντος (αισθητήρες, smart dust, τα συστήματα αυτόματου προσδιορισμού A.I.S, τα συστήματα ταυτοποίησης μέσω ραδιοσυχνοτήτων R.F.IDs κ.α.)
- Τεχνολογίες επικοινωνίας (ασύρματα δίκτυα κ.α.)
- Τεχνολογίες ασφάλειας δικτύων (firewalls, ψηφιακές υπογραφές κ.α.)





# 3

## Έξυπνα κινητά τηλέφωνα: Αισθητήρες

Τα έξυπνα κινητά τηλέφωνα (smartphones) ανήκουν στην κατηγορία των φορητών υπολογιστών και είναι συσκευές που μέσω φυσικών διεπαφών χρήστη (intuitive user interfaces) και της δυνατότητας εκμετάλλευσης της προαναφερθείσας διάχυτης υπολογιστικής μας φέρνουν ένα βήμα πιο κοντά με την απτή αλλά και την περιρρέουσα νοημοσύνη αφού έχουν παραπάνω δυνατότητες από τα υπόλοιπα κινητά και ξεχωρίζουν από αυτά διότι έχουν ένα από τα παρακάτω λειτουργικά συστήματα:

1. Windows Mobile
2. iPhone OS
3. Google's Android
4. Symbian OS
5. RIM's BlackBerry
6. Palm's WebOS
7. Linux
8. Bada

Οι δυνατότητες χρήσης των έξυπνων κινητών τηλεφώνων είναι πολλές αφού συνοδεύονται με:

- εφαρμογές που μπορεί κάποιος να εγκαταστήσει ανά πάσα στιγμή
- πρόσβαση στο διαδίκτυο
- πληκτρολόγιο 'qwerty' δηλαδή με σειρά γραμμάτων όπως του υπολογιστή
- δυνατότητα συγχρονισμού των διάφορων εφαρμογών που χρειάζονται διαδίκτυο , των επαφών , των e-mails, κλπ.
- Διάφορους αισθητήρες που χρησιμοποιούνται κυρίως για την επέκταση και ανάπτυξη εφαρμογών

Αφού το πεδίο ανάπτυξης εφαρμογών είναι άπειρο , το διαδίκτυο πολύ μεγάλο και οι αισθητήρες αρκετοί για την αναγνώριση των περισσότερων κινήσεων, ο συνδυασμός αυτών εξελίσσει και πληθαίνει τις δυνατότητες των έξυπνων κινητών τηλεφώνων συνεχώς. Για την καλύτερη κατανόηση θα πρέπει να γίνει αναφορά στην επίτευξη της περιρρέουσας νοημοσύνης μέσω αισθητήρων που είναι από τα πιο απαραίτητα αλλά και παραμελημένα μέσα δυναμικής συλλογής δεδομένων (αφού τα δεδομένα αλλάζουν διαρκώς).

### 3.1 Αναφορά στους αισθητήρες

| Αισθητήρες έξυπνων κινητών τηλεφώνων |   |
|--------------------------------------|---|
| αισθητήρας                           | μέγεθος μέτρησης                          |
| τριαξονικό γυροσκόπιο                | περιστροφή στον χώρο                      |
| τριαξονικό μαγνητόμετρο              | κατεύθυνση (πυξίδα)                       |
| επιταχυνσιόμετρο                     | επιτάχυνση σε x,ψ,ζ άξονες                |
| αισθητήρας ατμοσφαιρικού φωτός       | φωτεινότητα                               |
| κάμερα                               | εικόνες, βίντεο                           |
| παγκόσμιο σύστημα θεσιθεσίας (GPS)   | γεωγραφικές συντεταγμένες                 |
| αισθητήρας μέτρησης υγρασίας         | υγρασία                                   |
| μικρόφωνο                            | ήχος                                      |
| πιεσόμετρο                           | πίεση (χρησιμεύει για εύρεση υψομέτρου)   |
| αισθητήρας εγγύτητας                 | απόσταση συσκευής από κοντινά αντικείμενα |
| μετρητής θερμοκρασίας                | θερμοκρασία                               |

**Εικόνα 5 : Αισθητήρες έξυπνων κινητών τηλεφώνων**

Οι **βασικοί αισθητήρες** που χρησιμοποιούνται για τις περισσότερες εφαρμογές είναι το γυροσκόπιο (gyroscope) τα τελευταία 2 χρόνια, το επιταχυνσιόμετρο (accelerometer), ο αισθητήρας εγγύτητας (proximity), το gps, το μικρόφωνο και η κάμερα ενώ οι υπόλοιποι συναντούνται πιο σπάνια και η ακρίβεια των τιμών τους εξαρτάται σχεδόν πάντα από την συσκευή.

Το **επιταχυνσιόμετρο (accelerometer)** χρησιμοποιείται για να ανιχνεύει τις αλλαγές στον προσανατολισμό της συσκευής που χρησιμοποιείς (π.χ. όταν γέρνεις το κινητό στο πλάι, αλλάζει ο web browser που χρησιμοποιείς σε Landscape για να γίνεται πιο ευρεία η εικόνα). Παίρνει τιμές και στους τρεις άξονες και οι τιμές δείχνουν την φαινομενική επιτάχυνση που δέχεται στους άξονες από τις δυνάμεις που του ασκούνται κάθε microsecond, π.χ.  $9.81 \text{ m/s}^2$  στον άξονα Ψ αν το κρατάς κατακόρυφα λόγω βαρύτητας. Είναι αξιόπιστο αλλά σε απότομες αλλαγές προσανατολισμού 'κάνει παράσιτα' και αποκλίνει στον υπολογισμό ταχυτήτων (μέσω της επιτάχυνσης) και ακόμα περισσότερο στον υπολογισμό αποστάσεων.

Το **γυροσκόπιο (gyroscope)** χρησιμοποιείται για τη μέτρηση, εύρεση ή τη διατήρηση προσανατολισμού της συσκευής και βασίζεται στις αρχές της στροφορμής (π.χ. σε

συστήματα πλοήγησης και σε συστήματα αναγνώρισης κινήσεων-χειρονομιών). Στην ουσία υπολογίζει την ταχύτητα περιστροφής (rad/s) γύρω από τους άξονες.

Το **παγκόσμιο σύστημα θεσιθεσίας (Global Positioning System - GPS)** είναι ένα αρκετά ενεργοβόρο σύστημα πλοήγησης με σκοπό να προσδιορίσει τη θέση της συσκευής στη Γη ακρίβεια έως περίπου 10 με 50 μέτρα αναλόγως την συσκευή. Τα GPS παρέχουν γεωγραφικό μήκος και πλάτος (σε μοίρες και υποδιαϊρέσεις αυτών). Στην αρχή GPS συσκευές χρησιμοποιούνταν μόνο από το στρατό, από τους πιλότους των αεροσκαφών και από τους ναυτικούς αλλά στη δεκαετία του 1980, έγινε διαθέσιμο για χρήση το σύστημα και για πολίτες.

**Ο αισθητήρας περιβάλλοντος φωτός(ambient light sensor)** χρησιμοποιείται για την ανίχνευση της φωτεινότητας (lux ή  $\mu\text{W}/\text{cm}^2$ ) του περιβάλλοντος της συσκευής και συνήθως χρησιμοποιείται για την προσαρμογή της φωτεινότητας της οθόνης της συσκευής ώστε να φαίνεται η οθόνη καλά από γυμνό μάτι αλλά ταυτόχρονα να μην είναι πολύ φωτεινή με στόχο την εξοικονόμηση ενέργειας και αποτέλεσμα να αντέχει περισσότερο η μπαταρία της συσκευής.

**Ο αισθητήρας εγγύτητας (Proximity sensor)** ανιχνεύει πόσο κοντά βρίσκεται κάτι από την οθόνη του κινητού, χρησιμοποιείται κυρίως για εξοικονόμηση ενέργειας, ενεργοποιείται σε πολύ συγκεκριμένες ενέργειες, και μέσω αυτού του αισθητήρα μπορεί να καταλαβαίνει το έξυπνο κινητό τηλέφωνο (smartphone) αν βρίσκεται το κινητό κοντά στο αυτί του χρήστη και τότε συνήθως σβήνει η οθόνη για εξοικονόμηση μπαταρίας και για να αποφευχθεί η τυχαία επαφή του χρήστη με την οθόνη αφής. Στην ουσία όταν δεχόμαστε ένα τηλεφώνημα και φέρουμε το κινητό κοντά στο πρόσωπο κλείνει αυτόματα οποιαδήποτε λειτουργία κάναμε προηγουμένως και συνεχίζει αυτόματα αφού το απομακρύνουμε. Στις περισσότερες συσκευές android ο αισθητήρας μπορεί να επιστρέφει δυο τιμές, κοντά ή μακριά.

**Η Πυξίδα** είναι αισθητήρας που χρησιμοποιείται για να εμφανίζεται ο προσανατολισμός της συσκευής σε σχέση με το μαγνητικό πεδίο της γης μέσω του φαινομένου hall effect χωρίς να χρησιμοποιούνται μαγνητικά κυκλώματα στην συσκευή αλλά με το ρεύμα που διέρχεται σε αυτήν καθώς βρίσκεται στο μαγνητικό πεδίο της γης και μέσω αυτού η συσκευή μπορεί να μετρήσει το συνολικό διάνυσμα του μαγνητικού πεδίου.

## 3.2 Χρήση αισθητήρων για μοντελοποίηση ενεργειών και συνδυασμός τους για μεγαλύτερη αξιοπιστία

Μέσω της κατανόησης του τρόπου λειτουργίας των αισθητήρων είναι εφικτή η υλοποίηση του οράματος της περιρρέουσας νοημοσύνης. Αυτό συμβαίνει αφού με την σωστή χρήση συνδυασμού των αισθητήρων μπορεί να γίνει με μεγάλη ακρίβεια ανίχνευση ή και “χαρτογράφηση” (συνήθως μέσω GPS, ALS, humidity sensor, πιεσόμετρου, θερμόμετρου, μαγνητόμετρου, κάμερας ) του περιβάλλοντος ενός έξυπνου κινητού τηλεφώνου και

συνεπώς του χρήστη του (π.χ. μέσω του μικροφώνου, του επιταχυνσιόμετρου και του GPS μπορεί να ανιχνεύσει αν χρησιμοποιεί μετρό ή λεωφορείο) ενώ παράλληλα γίνεται εφικτή η μοντελοποίηση για αρχή και η ανίχνευση στην συνέχεια (συνήθως μέσω επιταχυνσιόμετρου, γυροσκοπίου, GPS, Proximity meter, μικροφώνου) ανθρωπίνων συμπεριφορών (π.χ. περπάτημα).

Συνεπώς μέσω μιας εφαρμογής έξυπνου κινητού τηλεφώνου μπορούμε να αναγνωρίσουμε και τον συνδυασμό μιας ανθρώπινης συμπεριφοράς σε ένα συγκεκριμένο περιβάλλον (π.χ. πόσες φορές πήγα από το σπίτι στην δουλειά σήμερα χρησιμοποιώντας το GPS και αν περπάτησα για να πάω χρησιμοποιώντας επιταχυνσιόμετρο).

Για να γίνει με μεγάλη ακρίβεια η ανίχνευση της ύπαρξης μίας διαδικασίας και της διαδικασίας της ίδιας, χρησιμοποιούνται πολλαπλές πηγές σημάτων και ενδείξεων διότι οι επιπλέον πληροφορίες από τις υπόλοιπες πηγές (π.χ. αισθητήρες) μπορούν να βοηθήσουν πάνω στην βελτίωση των συμπερασμάτων που προκύπτουν. Για παράδειγμα, **ένα γυροσκόπιο με ένα επιταχυνσιόμετρο** επιτρέπει στη συσκευή να ανιχνεύει την κίνηση σε έξι κατευθύνσεις - αριστερά, δεξιά, πάνω, κάτω, εμπρός και πίσω, καθώς και περιστροφές γύρω από τους άξονες x,y,z - επιτρέποντας μεγαλύτερη ακρίβεια ανίχνευσης των κινήσεων συγκρίσιμη με έναν χειριστήριο παιχνιδιών όπως το Wii – mote που έχει φτιαχτεί για να επιτελεί μόνο αυτόν τον σκοπό. Συνδυάζοντας στην συνέχεια τα δεδομένα αυτά με το GPS μπορεί να υπάρξει επαλήθευση για την στιγμιαία ταχύτητα της συσκευής και συνεπώς ένα πιο σίγουρο συμπέρασμα.

### 3.2.1 Μοντελοποίηση ανθρωπίνων συμπεριφορών

Η ανάλυση και μοντελοποίηση μίας ανθρώπινης κίνησης ή ενέργειας είναι αυξανόμενου ενδιαφέροντος για την κοινωνία της πληροφορικής αφού χρησιμοποιείται σε πολλά πεδία πέρα από εφαρμογές όπως ταινίες, κινούμενα σχέδια, άθληση καθώς και σε αναπηρίες. Κοινωνικές εφαρμογές που βοηθούν στην καθημερινή ζωή έχουν αρχίσει να προσελκύουν τα βλέμματα πολλών επιστημόνων.

Για την ανίχνευση μίας διαδικασίας από το έξυπνο κινητό τηλέφωνο (smartphone ) απαιτείται κάποιου είδους μοντελοποίηση της και στην συνέχεια έλεγχος σε τι ποσοστό ακολουθείται το μοντέλο. Υπάρχουν πολλές θεωρίες και τεχνικές για την μοντελοποίηση της κάθε διαδικασίας ανθρώπινης ή μη. Για την ύπαρξη ενός μοντέλου της διαδικασίας απαιτείται ακριβής ανάλυση της και συνήθως γνώσεις μαθηματικών και προγραμματισμού λογικής σκέψης . Το ίδιο συμβαίνει σε όλους τους τομείς αφού και η δημιουργία ενός επιχειρηματικού μοντέλου δεν απέχει σε λογική.

Η λογική με την οποία μοντελοποιήθηκαν οι ενέργειες στην αναπτυχθείσα εφαρμογή δεν ακολουθεί κάποιο ακριβές σχέδιο αλλά πλησιάζει την **ασαφή λογική (fuzzy logic)** που χρησιμοποιείται για εξαγωγή συμπερασμάτων από μη γραμμικό και μεγάλο πλήθος συχνά μη ακριβή δεδομένων και έχει να κάνει με προσέγγιση μιας διαδικασίας κριτικών σκέψεων που δεν είναι συγκεντρωμένη και ορισμένη.

### 3.2.2 Ασαφής Λογική (Fuzzy Logic)

Η ασαφής λογική (fuzzy logic) είναι μια επέκταση της κλασσικής αριστοτέλειας λογικής. Μια πρόταση μπορεί να είναι αληθής "με κάποιο βαθμό αληθείας", και όχι απλά αληθής ή ψευδής. Με απλά λόγια, η ασαφής λογική λέει ότι τα πράγματα συχνά δεν είναι «άσπρο-μαύρο» αλλά «αποχρώσεις του γκρι». Η ιδέα αυτή απετέλεσε επανάσταση στη θεωρία της λογικής, γιατί ξέφυγε από το μοντέλο που κυριαρχούσε εδώ και 2500 χρόνια, δηλαδή το μοντέλο του «0-1», «αληθές-ψευδές». Με την ασαφή λογική θα θέλαμε ξεκινώντας από τη δίτιμη κλασσική λογική (0 ή 1) να την επεκτείνουμε εισάγοντας, ασάφεια, αοριστία, αβεβαιότητα κτλ, έτσι ώστε αφενός να προσεγγίζει την εκφραστική δύναμη και απλότητα της φυσικής γλώσσας, αλλά και αφετέρου να περισώζει όσον το δυνατόν περισσότερο τη γνωστή μαθηματική δομή της κλασσικής λογικής.

Η ασαφής λογική είναι η προσπάθεια των επιστημόνων και κυρίως αυτών που ασχολούνται με την **“τεχνητή νοημοσύνη”** να μελετήσουν και να μαθηματικοποιήσουν τη δομή της φυσικής γλώσσας του ανθρώπου και τη φυσική ασάφεια που διέπει την ανθρώπινη νοημοσύνη - συμπεριφορά, καθώς και η αναγκαιότητα για μια πιο ρεαλιστική θεώρηση της έννοιας της αβεβαιότητας στην πράξη. Έτσι, στην ασαφή λογική εκτός των δύο τιμών αληθείας 0 και 1, έχουμε ως τιμή αλήθειας και κάθε αριθμό στο κλειστό απειροδιάστημα [0,1].

Η θεωρία των ασαφών συνόλων είναι ένας τρόπος να προσδιορίζουμε πόσο καλά ένα αντικείμενο ικανοποιεί μια αόριστη περιγραφή. Αυτός ο προσδιορισμός είναι υποκειμενικός, γιατί εξαρτάται από το πώς αντιλαμβάνεται ο καθένας τους λεκτικούς προσδιορισμούς ποσοτικών μεγεθών.

Εμπνευστής της θεωρίας των Ασαφών Συνόλων είναι ο Καθηγητής του Πανεπιστημίου Berkeley της Καλιφόρνιας *Lotfi Zadeh* όπου με την εργασία του με τίτλο *“Fuzzy Sets-Ασαφή Σύνολα”*, διατύπωσε εισήγαγε για πρώτη φορά την έννοια του Ασαφούς Συνόλου με πεδίο αληθοτιμών το απειροδιάστημα [0,1] και τον όρο *fuzzy* στη διεθνή βιβλιογραφία, ανοίγοντας παράλληλα το δρόμο και για τις πρακτικές εφαρμογές της. Ο *Zadeh* διατύπωσε την περίφημη **Αρχή της Ασυμβατότητας** : «Καθώς η πολυπλοκότητα ενός συστήματος αυξάνεται, η ικανότητα για ακριβείς και ταυτόχρονα σημαντικές δηλώσεις που αφορούν τη συμπεριφορά του μειώνεται, και πέρα από ένα σημείο η ακρίβεια και η σημαντικότητα αποτελούν σχεδόν αμοιβαία αποκλειόμενα χαρακτηριστικά». Μετά την αρχική δυσπιστία και τις πρώτες φυσιολογικές αμφισβητήσεις της νέας Θεωρίας, σύντομα ακολούθησαν και οι τεχνολογικές εφαρμογές της Ασαφούς Λογικής, κυρίως στα δυναμικά μη-γραμμικά συστήματα ελέγχου (non-linear control systems), όπου συνήθως τα συμβατικά μαθηματικά μοντέλα δεν ισχύουν πια. Σήμερα είναι πολυάριθμες σε όλο τον κόσμο οι Τεχνολογικές και Θεωρητικές εφαρμογές της Ασαφούς Λογικής. Η αγορά έχει κυριολεκτικά πλημμυρίσει με συσκευές που ενσωματώνουν ασαφείς ελεγκτές (πλυντήρια, ψυγεία, κλιματιστικά, κιβώτια ταχυτήτων, ABS, κλπ), όπως και ασαφείς ελεγκτές μη-επανδρωμένων οχημάτων-αεροσκαφών, λειτουργίας τσιμεντοβιομηχανιών, βιντεοκαμερών, ρομπότ, ιατρικής διάγνωσης, αναισθησίας, χειρουργικής, κ.τ.λ.

Τέλος, δεν υπάρχει ίσως επιστημονικός κλάδος σήμερα που να μην επεκτείνεται ραγδαία σε έννοιες και εφαρμογές της Ασαφούς Λογικής, όπως: Μαθηματικά (ασαφής Άλγεβρα ή Τοπολογία, ασαφείς Πιθανότητες), Οικονομία, Στατιστική, Φιλοσοφία, Σεισμολογία, Ιατρική, Ρομποτική, Βιολογία, Ψυχολογία, Οικολογία, Διαστημική, Κοινωνιολογία, Πυρηνική, Γενετική, και άλλοι.

Εφαρμογές της Ασαφούς Λογικής εμφανίζονται σε πεδία :

- **Μοντελοποίησης (Modeling),**
- **Πρόβλεψης (Prediction),**
- **Ελέγχου (control),**
- **Κατάταξης δειγμάτων σε κατηγορίες (Classification),**
- **Συστημάτων υποστήριξης αποφάσεων (Decision Support Systems),**
- **Θεωρίας δυνατοτήτων (Possibility Theory),**
- **Βελτιστοποίησης (Optimization),**
- **Δημιουργίας προφίλ (profiling),**
- **Αναγνώρισης προτύπων (Pattern Recognition)**

και άλλα.

### **3.2.3 Τεχνική δημιουργίας προφίλ (profiling) και αναγνώρισης προτύπων (pattern recognition)**

Επειδή η εφαρμογή είναι ανθρωποκεντρική συχνά αυτή η ασαφής λογική μοντελοποίησης αναφέρεται ως τεχνική δημιουργίας προφίλ (profiling practices) το οποίο για την επιστήμη της πληροφορικής είναι η διαδικασία δημιουργίας και εφαρμογής προφίλ που προκύπτουν από υπολογιστική ανάλυση δεδομένων.

Η τεχνική δημιουργίας προφίλ συνήθως περιλαμβάνει χρήση αλγορίθμων και μαθηματικών τεχνικών που επιτρέπουν την εύρεση μοτίβων (patterns) ή συσχετίσεων σε μεγάλο πλήθος δεδομένων που συνήθως βρίσκονται σε βάσεις ή πίνακες. Όταν αυτά τα μοτίβα ή συσχετίσεις χρησιμοποιούνται για την αναγνώριση ή εκπροσώπηση κάποιου ανθρώπου τότε λέγονται προφίλ. Η τεχνική δημιουργίας προφίλ πέρα από τα παραπάνω περιέχει και την εφαρμογή μίας ομάδας προφίλ σε μεμονωμένα άτομα για την ευκολότερη εξατομίκευση.

Στην συνέχεια παρουσιάζεται η γενίκευση της λογικής που χρησιμοποιήθηκε στην αναπτυχθείσα εφαρμογή για τη μοντελοποίηση οποιασδήποτε ενέργειας μέσω έξυπνων κινητών τηλεφώνων.

**Έστω ότι πρέπει να μοντελοποιηθεί και να ανιχνευθεί μια γνωστή φυσική διαδικασία - ενέργεια Δ.**

- i. Για αρχή πρέπει να βρεθεί αν επηρεάζεται η συσκευή κατά την διάρκεια της Δ, πριν την Δ (έστω Δ1 η διαδικασία πριν) ή και μετά την Δ (έστω Δ2 η διαδικασία μετά).
- ii. Βρίσκονται οι κύριοι αισθητήρες που επηρεάζονται από την Δ1 ή και Δ2 ή και Δ, έστω ο α1,α2,α3,.. (όπου α1,α2,α3,.. οποιοιδήποτε αισθητήρες)
- iii. Στην συνέχεια πρέπει να διαπιστωθεί αν οι αισθητήρες επηρεάζονται διαφορετικά ανάλογα με την περιοχή (π.χ. υψόμετρο, πόλη ή χωριό) και τον άνθρωπο (π.χ. ύψος, βάρος, ηλικία).
- iv. Λαμβάνονται τιμές Δ1(T1[ ], T2[ ], T3[ ],...), Δ2(T1[ ], T2[ ], T3[ ],...), Δ(T1[ ], T2[ ], T3[ ],...) από τους α1, α2, α3,.. για τις Δ1,Δ2,Δ ( για την Δ1 ο α1 θα πάρει το σύνολο τιμών Δ1(T1[.]) ) και αποτυπώνονται αναλυτικά δημιουργώντας μία βάση δεδομένων ή έναν πίνακα μεταβολής των τιμών τους για όσο δυνατόν περισσότερες περιοχές και ανθρώπους γίνεται εφόσον επηρεάζεται από αυτό και δεν μπορεί να βρεθεί κάποια συγκεκριμένη ή απλούστερη λύση για αυτήν την διαφορά που υπάρχει σε κάθε άνθρωπο ή περιοχή.
- v. Για καθεμία από τις Δ1,Δ2,Δ Βρίσκονται ακολουθίες προφίλ f1(Δ1), γ1(Δ1),.. , f2(Δ2), γ2(Δ2),.. , f(Δ), γ(Δ),.. που ακολουθούν οι ληφθέντες των α1,α2,.. που θα χαρακτήριζαν μόνο αυτές τις διαδικασίες, εφόσον γίνεται, έτσι ώστε κατά την Δ1,Δ2,Δ οι τιμές των α1, α2, α3,.. θα παίρνουν πάντα τιμές τέτοιες ώστε να ικανοποιούν τις f(Δ(T1[ ])),γ(Δ(T2[ ])),.. f1(Δ1(T1[ ])),γ1(Δ1(T2[ ])),.. f2(Δ2(T1[ ])),γ2(Δ2(T2[ ])),.. . Αυτές οι f(),γ(),.. θα πρέπει να είναι γενικευμένες για όλους τους ανθρώπους και τις περιοχές αλλιώς θα πρέπει να βρεθούν f1(), f2(), f(),.. , γ1(), γ2(), γ(),.. , ... για τις διάφορες περιοχές και ανθρώπους.
- vi. Γίνεται έλεγχος κάθε φορά αν οι τιμές που λαμβάνουν οι α1, α2, α3,.. ικανοποιούν τις { f(Δ(T1[ ])),γ(Δ(T2[ ])),.. } ή { f1(Δ1(T1[ ])),γ1(Δ1(T2[ ])),.. } ή {f2(Δ2(T1[ ])),γ2(Δ2(T2[ ])),.. } οπότε αναγνωρίζεται ποια διαδικασία συμβαίνει. (συχνά δεν γίνεται να βρεθεί η Δ αλλά μερικές φορές γίνεται να την καταλάβουμε αφού θα έχει συμβεί η Δ1 και η Δ2 - θα σημαίνει πως ενδιάμεσα έγινε η Δ)

Για παράδειγμα έστω ότι πρέπει να αναγνωρίζεται εάν κάποιος παραλαμβάνει ένα αντικείμενο (Δ) το οποίο γνωρίζουμε ότι βρίσκεται σε ένα σκοτεινό χώρο Α και θέλει να το μεταφέρει σε έναν χώρο Β χωρίς να είναι απαραίτητο να ακολουθήσει κάποια συγκεκριμένη διαδρομή τότε:

- 1) (Βήμα i, ii) Κοιτάμε αν επηρεάζονται οι αισθητήρες κατά την μεταφορά. Έστω ότι επηρεάζονται το GPS από την αλλαγή της τοποθεσίας όπου  $\Delta 1$  είναι η άφιξη στον χώρο A και  $\Delta 2$  η άφιξη στον χώρο B, ο αισθητήρας φωτός – ALS (διότι στο A είναι σκοτεινά) από την εναλλαγή στην ένταση του φωτός και το επιταχυνσιόμετρο - accelerometer (λόγο συγκεκριμένης κίνησης για παραλαβή αντικειμένου) από την παραλαβή του αντικειμένου.
- 2) (Βήμα iii) Παρατηρούμε ότι δεν επηρεάζεται διαφορετικά κάποιος αισθητήρας για τα διάφορα άτομα που μπορούν να κάνουν την μεταφορά και οι χώροι βρίσκονται σε συγκεκριμένες τοποθεσίες άλλα λόγω μη εγκυρότητας του GPS μπορεί να υπάρχει σφάλμα έως 10 – 15 μέτρα.
- 3) (Βήμα iv) Βλέπουμε τι τιμές παίρνει το GPS στους χώρους A και B (A. γεωγραφικό μήκος, πλάτος και B. γεωγραφικό μήκος, πλάτος) και τι τιμές παίρνει το επιταχυνσιόμετρο και το ALS κατά την παραλαβή (έστω πίνακας επιταχυνσιόμετρου [τιμές κατά την παραλαβή] και φωτεινότητα ALS στον A )
- 4) (Βήμα v) Για να δεχτούμε πως ο χρήστης θα βρίσκεται στο σημείο A σημαίνει ότι το GPS θα πρέπει να έχει ένδειξη A. γεωγραφικό μήκος, πλάτος  $\pm 15$  μέτρα σε μοίρες (έλεγχος  $f1(\Delta 1[A. \text{γεωγραφικό μήκος, πλάτος}])$  ) και πως έχει περίπου την ανάλογη φωτεινότητα με φωτεινότητα ALS στον A (έλεγχος  $\gamma 1(\Delta 1(\text{φωτεινότητα ALS στον A}))$ ). Επίσης ο πίνακας επιταχυνσιόμετρου [τιμές κατά την παραλαβή έχει σχεδόν συγκεκριμένες τιμές κάθε φορά οπότε βρίσκεται συνάρτηση (ακολουθία προφίλ) f που μόνο κάθε φορά που θα παραλαμβάνεται το αντικείμενο αυτές θα επαληθεύονται, δηλαδή  $f(\Delta(\text{πίνακας επιταχυνσιόμετρου [τιμές κατά την παραλαβή]})$ ) Για να δεχτούμε πως ο χρήστης θα βρίσκεται στο σημείο B σημαίνει ότι το GPS θα πρέπει να έχει ένδειξη B. γεωγραφικό μήκος, πλάτος  $\pm 15$  μέτρα σε μοίρες (έλεγχος  $f2(\Delta 2[B. \text{γεωγραφικό μήκος, πλάτος}])$  ).
- 5) (Βήμα vi) Στην συνέχεια γίνονται οι έλεγχοι, εάν ικανοποιούνται οι έλεγχοι  $f1, \gamma 1$  ο χρήστης βρίσκεται στον χώρο A, τότε εάν ικανοποιηθεί ο έλεγχος f το αντικείμενο παραλήφθηκε και τέλος εάν ικανοποιηθεί ο  $f2$  το αντικείμενο μεταφέρθηκε στον B.



### 3.3 Εφαρμογές

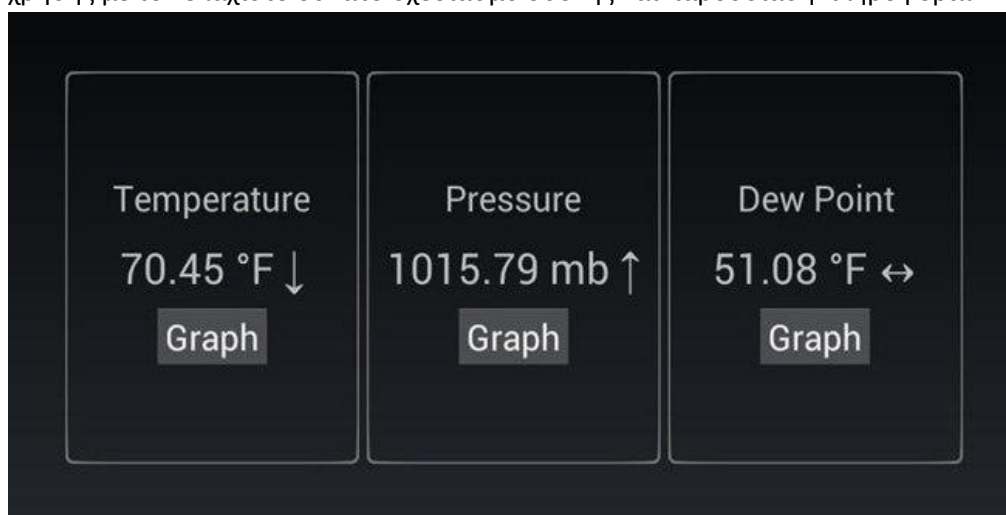
Υπάρχουν πάρα πολλοί λόγοι για την χρήση και συνεπώς ανάπτυξη εφαρμογών για έξυπνα κινητά τηλέφωνα (smartphone). Όπου και να χρησιμοποιεί κανείς το τηλέφωνό του, υπάρχουν εφαρμογές που μπορεί να ταιριάζουν με τις ανάγκες του και αφορούν διασκέδαση και παιχνίδια, εκδόσεις ( π.χ. e-books, ηλεκτρονικά βιβλία, περιοδικά), ψυχαγωγία, εκπαίδευση, επικοινωνία, μουσική, ταξίδια, υγεία θρησκεία, ειδήσεις, αθλήματα, οικονομικά και πολλά άλλα.. Ποιές εφαρμογές είναι διαθέσιμες για συγκεκριμένο έξυπνο κινητό τηλέφωνο (smartphone) εξαρτάται από τη μάρκα και το μοντέλο του. Μερικές εφαρμογές είναι καθολικές, ενώ άλλες είναι ειδικά για ορισμένες μάρκες (για συγκεκριμένες εκδόσεις συγκεκριμένων λειτουργικών συστημάτων που ίσως να έχουν συγκεκριμένες απαιτήσεις hardware).

Η ανάπτυξη εφαρμογών που βοηθούν στην επίτευξη μιας ενιαίας κοινωνίας ανθρώπων και υπολογιστών που θα επικοινωνούν με τον ελάχιστο δυνατό κόπο έχει φέρει επανάσταση στον χώρο των έξυπνων κινητών τηλεφώνων αν και ακόμα βρίσκεται στα πρώτα της στάδια. Προς το παρόν ένα μεγάλο κύμα εφαρμογών ξεχύθηκε με έμμεσο στόχο την συλλογή πληθώρας δεδομένων με την πρόφαση ότι η γνώση είναι δύναμη για την δημιουργία προφίλ χρηστών και την πώλησή τους σε ενδιαφερόμενες εταιρίες για εξατομικευμένες προωθητικές ενέργειες των προϊόντων τους ενώ πολλές εφαρμογές χρησιμοποιούνται καθημερινά από εκατομμύρια ανθρώπους χωρίς να το γνωρίζουν οι ίδιοι για την βελτίωση της καθημερινότητάς τους στον βαθμό που τους επιτρέπεται.

Εφαρμογές οι οποίες χρησιμοποιούν τους αισθητήρες για την ενημέρωση του χρήστη για κάποιο γεγονός ή για γενικά στοιχεία υπάρχουν πλέον πολλές και συνεχώς αυξάνονται καθώς είναι ένας γρήγορα και συνεχώς αναπτυσσόμενος τομέας.

Για παράδειγμα:

- **Ambient** : υπολογίζει την θερμοκρασία την υγρασία και την πίεση με την χρήση των αισθητήρων της συσκευής , έχει ένα απλό αλλά εύκολα κατανοητό περιβάλλον χρήσης με τον ελάχιστο δυνατό σχεδιασμό οθόνης και παρουσίαση πληροφοριών.



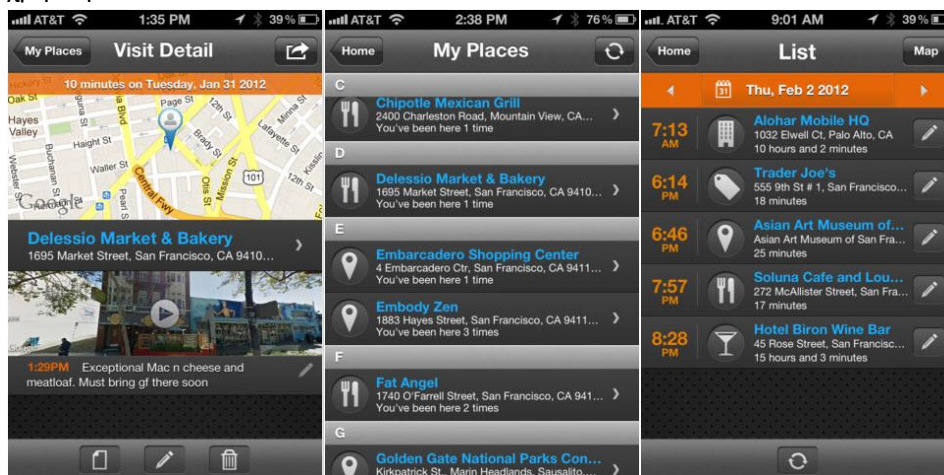
Εικόνα 6 : Η εφαρμογή Ambient

- **Runkeeper** : Εφαρμογή αγαπημένη για τους δρομείς , χρησιμοποιεί το GPS του τηλεφώνου για να παρακολουθεί τον χρήστη στο τρέξιμο, στην ποδηλασία , στην πεζοπορία και σε άλλες δραστηριότητες που ο χρήστης διανύει απόσταση . Η εφαρμογή καταγράφει το ρυθμό τρεξίματος , την διανυόμενη απόσταση , τον συνολικό χρόνο άσκησης , τις θερμίδες που καίγονται και άλλες χρήσιμες μετρήσεις, επιτρέποντας στον χρήστη να τραβήξει φωτογραφίες ενώ μπορεί να τις ανεβάσει σε κάποια ιστοσελίδα εν κινήσει . Οι χρήστες μπορούν να το χρησιμοποιούν για δικές τους διαδρομές , ή να κατεβάσουν από το διαδίκτυο και να ακολουθούν δημοφιλή δρομολόγια λειτουργίας . Δίνει την δυνατότητα παρακολούθησης της προόδου κατά την διαδρομή καθώς και κρατάει τα ιστορικά της προπόνησης, ενώ βρίσκεται σε συγχρονισμό με πολλές άλλες εφαρμογές .



*Εικόνα 7 : Η εφαρμογή Runkeeper*

- **Placeme** : Η εφαρμογή αυτή κρατάει αυτόματα στην μνήμη όλα τα μέρη που επισκέπτεται ο χρήστης και την διάρκεια επίσκεψης με έξυπνη χρήση των αισθητήρων ώστε να μην χρησιμοποιείται συνέχεια το GPS και καταναλώνει την μπαταρία της συσκευής και χωρίς την εισαγωγή κάποιας πληροφορίας από τον χρήστη.



*Εικόνα 8 : Η εφαρμογή Placeme*

# 4

## Η αναπτυχθείσα εφαρμογή

Σκοπός της εφαρμογής είναι να επικοινωνεί μία συσκευή τηλεφώνου android με έναν απομακρυσμένο εξυπηρετητή (server) και να αποθηκεύει σε μία βάση δεδομένων στον εξυπηρετητή (server) τις γεωγραφικές συντεταγμένες της διαδρομής που ακολούθησε ο χρήστης της συσκευής. Επίσης η συσκευή θα πρέπει να καταλαβαίνει μέσα από μερικές συναρτήσεις πότε ο χρήστης στάθμευσε το όχημα του και να εμφανίζει εφόσον του ζητηθεί έναν πίνακα ή τα σημεία σε χάρτη google maps με τις γεωγραφικές συντεταγμένες όπου θα είναι οι θέσεις στις όποιες έχει σταθμεύσει με σειρά παλαιότητας.

### 4.1 Η διαδικασία

Στην αρχή έγινε επιλογή λειτουργικού συστήματος προορισμού εφαρμογής, γλώσσας προγραμματισμού και περιβάλλοντος της. Επιλέχθηκε λειτουργικό σύστημα android, eclipse για περιβάλλον ανάπτυξης της εφαρμογής, συνεπώς γλώσσα προγραμματισμού java για την ανάπτυξη και xml για την διάταξη των στοιχείων της εφαρμογής (layouts). Στην συνέχεια επιλέχθηκε η γλώσσα php (χρησιμοποιώντας το πρόγραμμα ανάπτυξης κώδικα php «codelobster php edition»), βάση δεδομένων η mySQL . Στήθηκε τοπικά ένας εξυπηρετητής ιστού (web server) χρησιμοποιώντας το ελεύθερο λογισμικό XAMPP το οποίο κατεβάσαμε δωρεάν από τον ιστότοπο (<http://www.apachefriends.org>). Το λογισμικό XAMPP με ελάχιστες ρυθμίσεις και χωρίς εγκατάσταση, παρέχει ένα πλήρες σύστημα εξυπηρετητή ιστού Apache, την γλώσσα PHP, το σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων Mysql μαζί με το λογισμικό phpMyAdmin. Αφού επετεύχθη η επικοινωνία του τοπικού εξυπηρετητή ιστού (web server) με μια android συσκευή (μετάδοση δεδομένων από android σε server και αντιστρόφως) μετά έγινε το ίδιο διαδικτυακά με διαχειριστικά εργαλεία το cpanel και το phpmyadmin. Δηλαδή για τις ανάγκες της εφαρμογής αναρτήσαμε μια ιστοσελίδα στο διαδίκτυο σε συγκεκριμένο πάροχο. Αξιοποιήσαμε τη mysql και το λογισμικό phpmyadmin που μας παρέχει η εταιρεία φιλοξενίας ιστοσελίδων και δημιουργήσαμε μια βάση δεδομένων την android2 και τον πίνακα mobile2\_data με πεδία pid, timestamp, x, y, z, t.

Κατόπιν αφού έγινε εφικτή η επικοινωνία του απομακρυσμένου εξυπηρετητή ιστού με την android συσκευή μας έγινε προσπάθεια να γίνεται in real time (την δεδομένη χρονική στιγμή) με δεδομένα τα οποία έπαιρνε κάθε στιγμή με την μέγιστη απόδοση από τους αισθητήρες. Αυτό δυστυχώς δεν ήταν εφικτό διότι μέχρι να σταλθεί μία μέτρηση στον server (0.5 έως 1.5 δευτερόλεπτα περίπου) ο αισθητήρας είχε λάβει περισσότερες από μία τιμές (ορίζοντας τον σε χαμηλή συχνότητα sensor\_delay\_normal) το οποίο σημαίνει ότι δεν προλάβαινε να τις στείλει όλες τις μετρήσεις αλλά αρκεστήκαμε στο να ρίξουμε την απόδοση και να στέλνουμε τιμές από τους αισθητήρες κάθε 7 δευτερόλεπτα για να μην

υπάρχει περίπτωση συγκέντρωσης περισσότερων τιμών σε αυτό το διάστημα (ίσως λόγω αργής σύνδεσης) γεγονός που θα μπορούσε να μπλοκάρει την εφαρμογή.

Από την στιγμή που είχαμε μία αξιόπιστη μεταφορά δεδομένων και από τις δύο μεριές (android – server) έπρεπε να γίνει η αναγνώριση της διαδικασίας της στάθμευσης του οχήματος. Το να πατάει ο χρήστης ένα πλήκτρο στο κινητό του για να στέλνει την τοποθεσία που πάركαρε (όπως έχει ήδη αναπτυχθεί) απέχει πολύ από τον στόχο της εφαρμογής και έχει μεγάλο περιθώριο εξέλιξης ως προς την περιρρέουσα νοημοσύνη (ambient intelligence). Αυτή η αναγνώριση στάθμευσης είναι αδύνατο να γίνεται άμεσα εκείνη την στιγμή διότι εκείνη η στιγμή δεν λαμβάνει κάποιες ιδιαίτερες ενδείξεις από τους αισθητήρες αλλά η διαδικασία έχει κάποια προαπαιτούμενη σειρά ενεργειών και κάποια σειρά ενεργειών ύστερα από τη στάθμευση, σχεδόν μοναδικές για τη στάθμευση και μόνο, οι οποίες γίνεται να αναλυθούν, να μοντελοποιηθούν και να ανιχνευθούν από μία συσκευή android έτσι ώστε να γίνει εφικτή η γνώση των γεωγραφικών συντεταγμένων στάθμευσης. Αυτές είναι το περπάτημα του χρήστη του κινητού (προς το αυτοκίνητο), η οδήγηση (μέσα στο αυτοκίνητο) και το περπάτημα μετά την έξοδο από το αυτοκίνητο.

Αφού έγιναν όλα τα παραπάνω είχαμε ως αποτέλεσμα μία εφαρμογή που στέλνει τα απαραίτητα δεδομένα (συντεταγμένες γεωγραφικού μήκους και πλάτους, δυαδική μεταβλητή για το αν είναι θέση parking και ημερομηνία και ώρα) στην βάση δεδομένων ενός εξυπηρετητή ιστού (web server).

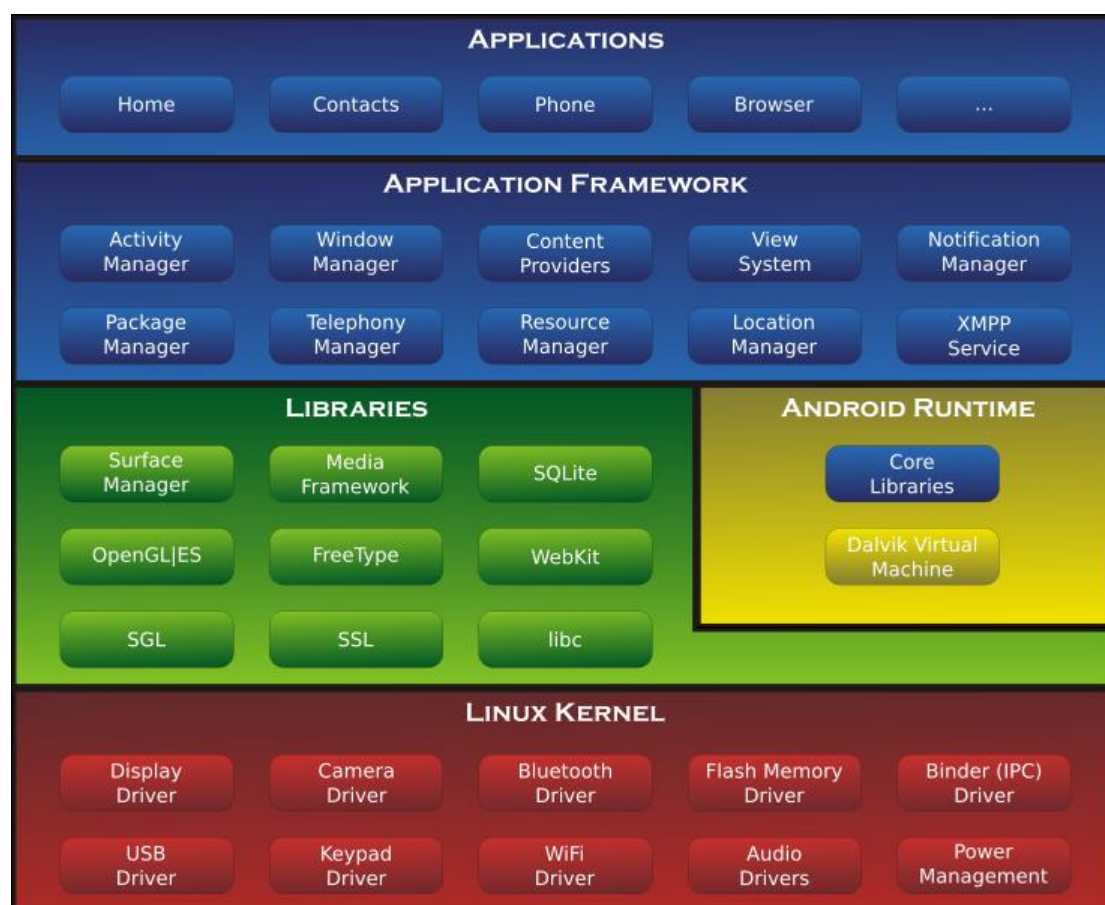
### 4.1.1 Επιλογή λειτουργικού συστήματος

Στην συγκεκριμένη μελέτη επιλέχθηκε η χρήση λειτουργικού συστήματος android γιατί είναι το πιο διαδεδομένο λειτουργικό σύστημα για έξυπνες συσκευές αφού εξυπηρετεί τις περισσότερες συσκευές χωρίς να είναι απαραίτητο συγκεκριμένο υλικό (hardware) όπως στα iPhone, επίσης η πλατφόρμα ανάπτυξης παρέχεται δωρεάν από την Google με πληθώρα πληροφοριών πάνω στον προγραμματισμό σε java και όλες τις απαραίτητες πληροφορίες για ανάπτυξη εφαρμογών από αρχάριους έως και προχωρημένους. (<http://developer.android.com/index.html>).

Δεν επιλέχτηκε λειτουργικό σύστημα iOS για λόγους ευκολίας αφού δεν υπήρχε μεγάλη εξοικείωση με iPhone. Το περιβάλλον ανάπτυξης για iOS παρέχεται δωρεάν σε κάθε υπολογιστή mac και μπορεί κάποιος να κάνει εγγραφή για ανάπτυξη για iOS δωρεάν αλλά η εγγραφή για να μπορέσει να δοκιμάσει την κάθε εφαρμογή σε μια συσκευή κοστίζει 99\$ τον χρόνο. Επίσης για να αναπτύξει κάποιος εφαρμογές για iOS χρησιμοποιώντας υπολογιστή με λειτουργικό σύστημα windows (που χρησιμοποιώ εγώ) χρειάζεται πολύ μεγαλύτερη διαδικασία (ενώ είναι πολύ πιο εύκολο για android). Τέλος ακόμα και αν η εφαρμογή που θα αναπτύξει κάποιος δεν πρόκειται να πωληθεί στο App Store το κόστος εγγραφής είναι 299 δολάρια ανά έτος και υπάρχουν κάποια επιπλέον έξοδα, κυρίως για τον έλεγχο του πηγαίου κώδικα.

## 4.1.2 Android: χαρακτηριστικά, προδιαγραφές, συστατικά στοιχεία

Το android λειτουργικό σύστημα είναι ένα multi-user (με περισσότερους από έναν χρήστες) σύστημα σε πυρήνα Linux στο οποίο κάθε πρόγραμμα – εφαρμογή (app) είναι και από ένας διαφορετικός χρήστης. Προκαθορίζει εξαρχής το σύστημα μία ταυτότητα (ID) για το κάθε app (εφαρμογή) και ορίζει τις άδειες που έχει η κάθε εφαρμογή οι οποίες αντιστοιχίζονται με την ταυτότητα της (ID). Κάθε εφαρμογή έχει την δική της εικονική μηχανή (virtual machine) οπότε και ο κώδικας της τρέχει ξεχωριστά από τις υπόλοιπες εφαρμογές. Κάθε εφαρμογή τρέχει μέσα στην δική της linux διαδικασία. Η διαδικασία της εφαρμογής ξεκινάει αν οποιοδήποτε κομμάτι της εφαρμογής πρέπει να εκτελεστεί και σταματάει όταν δεν είναι πλέον απαραίτητη ή πρέπει να γίνει ανάκτηση μνήμης για να ξεκινήσει μία νέα εφαρμογή.



**Εικόνα 9 : Η αρχιτεκτονική του Android**

Κάθε εφαρμογή έχει ελάχιστα δικαιώματα για θέματα ασφαλείας αλλά πάντα υπάρχουν τρόποι να μεγαλώσουν και να πληθύνουν τα δικαιώματα της. Μπορούν να συγχωνευτούν εφαρμογές χρησιμοποιώντας την ίδια ταυτότητα (ID) για να έχουν δικαίωμα η μία στα αρχεία της άλλης ή μέχρι και πρόσβαση στα SMS, στις επαφές, στην κάρτα SD και σε όλους τους αισθητήρες.

Για ευκολότερη κατανόηση πιο τεχνικών κομματιών θα πρέπει στην συνέχεια να γίνει αναφορά τουλάχιστον σε ότι έχει χρησιμοποιηθεί από τα παρακάτω:

- Στα συστατικά στοιχεία μίας εφαρμογής
- Στο αρχείο manifest στο οποίο ορίζονται τα διάφορα στοιχεία και άδειες λειτουργίας καθώς και τα απαιτούμενα χαρακτηριστικά συσκευής για την λειτουργία της εφαρμογής.
- Στις πηγές πέρα από τον κώδικα που βοηθάν μια εφαρμογή να βελτιώνουν την συμπεριφορά της για το πλήθος των συσκευών στις οποίες μπορεί να χρησιμοποιηθεί.

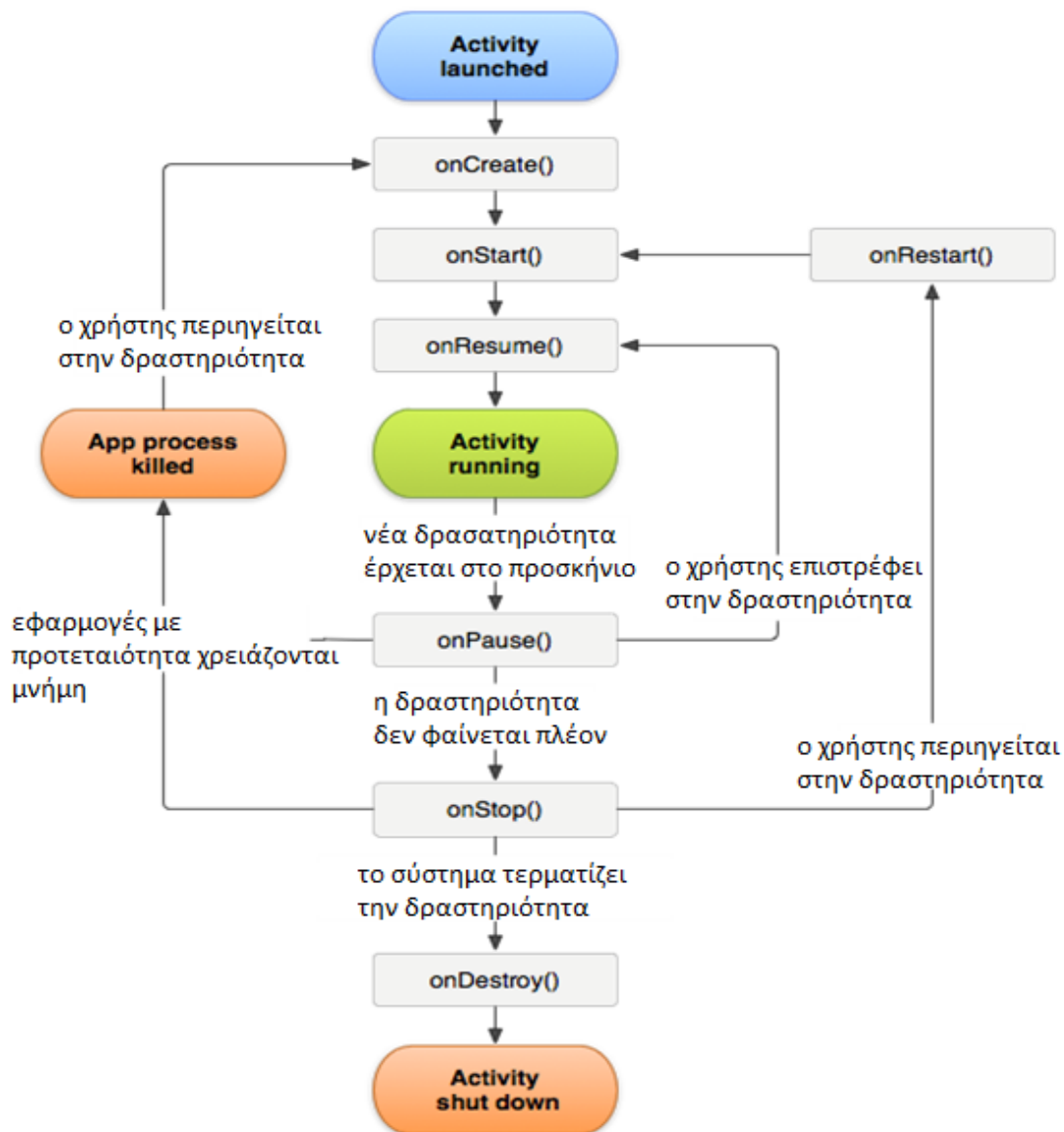
Με τον όρο συστατικά στοιχεία μίας εφαρμογής εννοούμε τα απαραίτητα δομικά στοιχεία μιας εφαρμογής Android. Το κάθε στοιχείο στην ουσία είναι ένας τρόπος πρόσβασης του λειτουργικού συστήματος στην εφαρμογή μας. Μπορούμε να τα διακρίνουμε σε τέσσερα βασικά στοιχεία : **τις δραστηριότητες (activities), τις υπηρεσίες (services), τους παρόχους περιεχομένου (content providers), και τους καθολικούς παραλήπτες μηνυμάτων (broadcast receivers).**

#### ➤ **Δραστηριότητες - activities**

Δραστηριότητα (Activity) είναι ένα συστατικό της εφαρμογής το οποίο παρέχει τη διεπαφή με την οποία αλληλεπιδρά ο χρήστης ώστε να κάνει κάποιες ενέργειες, όπως για παράδειγμα κλήση μιας επαφής, αποστολή ενός mail, τράβηγμα φωτογραφίας κλπ. Κάθε οθόνη που βλέπει και αλληλεπιδρά ο χρήστης είναι μια κλάση που κληρονομεί την κλάση activity.

Μια εφαρμογή συνήθως αποτελείται από πολλές δραστηριότητες που συνδέονται μεταξύ τους. Το πιο κοινό σενάριο είναι να υπάρχει μια κεντρική δραστηριότητα, η οποία παρουσιάζεται στο χρήστη κατά την εκτέλεση της εφαρμογής. Κάθε δραστηριότητα μπορεί να ξεκινήσει μια άλλη ώστε να εκτελέσει διαφορετικές ενέργειες. Κάθε φορά που μια νέα δραστηριότητα ξεκινά, η προηγούμενη σταματάει, αλλά το σύστημα την βάζει σε μια στοίβα. Έτσι όταν ο χρήστης ολοκληρώσει τις ενέργειες του σε αυτή την δραστηριότητα και πατήσει το πλήκτρο επιστροφής, η προηγούμενη δραστηριότητα θα επαναφερθεί από τη στοίβα και θα συνεχίσει την εκτέλεσή της.

Κάθε δραστηριότητα έχει ένα συγκεκριμένο κύκλο ζωής και κάποιες καταστάσεις, όπως για παράδειγμα Resumed, Paused και Stopped. Όταν μια δραστηριότητα περνάει σε μια κατάσταση αυτό αναγνωρίζεται από κάποιες μεθόδους επιστροφής (callback methods). Μια τέτοια μέθοδος επιστροφής είναι η onCreate την οποία πρέπει να υλοποιήσει ο προγραμματιστής και εκεί να ορίσει με την συνάρτηση setContentView() τη γραφική διεπαφή που θα έχει το Activity.



Εικόνα 10 : Ο κύκλος ζωής μιας Activity

### ➤ Υπηρεσίες -services

Μια υπηρεσία είναι ένα συστατικό στοιχείο που τρέχει στο παρασκήνιο (background) για την εκτέλεση λειτουργιών μακράς διάρκειας ή για να εκτελεί εργασίες για απομακρυσμένες διαδικασίες. Η υπηρεσία δεν παρέχει περιβάλλον εργασίας για τον χρήστη (δεν έχει layout). Για παράδειγμα, μια υπηρεσία θα μπορούσε να παίζει μουσική στο παρασκήνιο ενώ ο χρήστης βρίσκεται σε μία διαφορετική εφαρμογή ή θα μπορούσε τραβάει δεδομένα μέσω δικτύου χωρίς να εμποδίζει την αλληλεπίδραση του χρήστη με μια δραστηριότητα. Ένα άλλο στοιχείο, όπως μια δραστηριότητα, μπορεί να ξεκινήσει την υπηρεσία και να την αφήσει να τρέχει ή να συνδέεται με αυτή, όταν πρέπει να αλληλεπιδρούν. Τότε υπηρεσία υλοποιείται ως υποκατηγορία της δραστηριότητας όπως θα έχει γίνει και στην εφαρμογή που αναπτύχθηκε.

### ➤ **Οι πάροχοι περιεχομένου – content providers**

Οι πάροχοι περιεχομένου (content provides) διαχειρίζονται αποθηκευτικούς χώρους για δεδομένα, οι οποίοι είναι προσπελάσιμοι από οποιαδήποτε εφαρμογή. Αποτελούν τον μοναδικό τρόπο για αποθήκευση δεδομένων στην συσκευή, στα οποία έχουμε πρόσβαση και από άλλες εφαρμογές. Κλασικό παράδειγμα είναι η εφαρμογή για την αναπαραγωγή μουσικής η οποία αποθηκεύει τα αρχεία μας σε συγκεκριμένο χώρο ώστε αυτά να είναι προσβάσιμα από οποιαδήποτε άλλη εφαρμογή. Σε αυτό το πλαίσιο υπάρχει και η ενσωματωμένη βάση δεδομένων στο λειτουργικό σύστημα Android, η SQLite Database (είναι σαν την SQL αλλά λειτουργεί μόνο τοπικά), στην οποία μπορεί να αποθηκεύσουν και να διαβάσουν δεδομένα οι πάροχοι περιεχομένου και να τα περάσουν στην εφαρμογή.

Οι πάροχοι περιεχομένου είναι επίσης χρήσιμοι για την ανάγνωση και εγγραφή ιδιωτικών δεδομένων που δεν μοιράζεται η εφαρμογή με άλλες. Για παράδειγμα, μία εφαρμογή σημειωματάριο (notepad) χρησιμοποιεί μια υπηρεσία content provider για να αποθηκεύει τις σημειώσεις.

### ➤ **Αποδέκτες σήματος – Broadcast Receivers**

Αποδέκτης σήματος (Broadcast receiver) είναι ένα στοιχείο που ανταποκρίνεται σε ολόκληρο το σύστημα μετάδοσης “ανακοινώσεων” μιας android συσκευής. Πολλές εκπομπές σήματος προέρχονται από το σύστημα, για παράδειγμα γίνεται εκπομπή όταν η οθόνη έχει απενεργοποιηθεί, όταν η μπαταρία είναι χαμηλή, ή βγει μία φωτογραφία. Οι εφαρμογές μπορούν επίσης να εκπέμπουν σήμα, για παράδειγμα μία εφαρμογή μπορεί να στέλνει σήμα σε άλλες εφαρμογές ότι κάποια δεδομένα έχουν κατέβει στη συσκευή και είναι διαθέσιμα για χρήση. Κατά την μετάδοση σήματος δεν υπάρχει κάποια επικοινωνία με τον χρήστη μέσω κάποιας διεπαφής ώστε να το βλέπει ο ίδιος. Συνήθως ένας δέκτης σήματος είναι απλά μια "πύλη" για άλλα στοιχεία και για έναρξη δραστηριοτήτων και συνήθως δεν καταναλώνει υπολογιστική ισχύ και ενέργεια. Για παράδειγμα, θα μπορούσε να στέλνει σήμα με το άνοιγμα του κινητού σε μια υπηρεσία - service ώστε να αρχίσει να εκτελείται.

Μια μοναδική πτυχή στον σχεδιασμό του συστήματος του android είναι ότι κάθε εφαρμογή μπορεί να καλέσει ένα συστατικό στοιχείο από μία άλλη εφαρμογή. Για παράδειγμα, εάν θέλουμε σε μία αναπτυσσόμενη εφαρμογή να επεξεργαζόμαστε φωτογραφίες που τραβάμε με την κάμερα της συσκευής, υπάρχει λογικά μία άλλη εφαρμογή που μπορούμε να χρησιμοποιήσουμε τις δραστηριότητες της αντί να αναπτύξουμε μια νέα δραστηριότητα για την χρήση της κάμερας. Στην συνέχεια αρκεί να συνδέσουμε τις εφαρμογές έτσι ώστε να περνάμε τις φωτογραφίες από άλλη εφαρμογή σε αυτή και να επεξεργαζόμαστε τις φωτογραφίες.

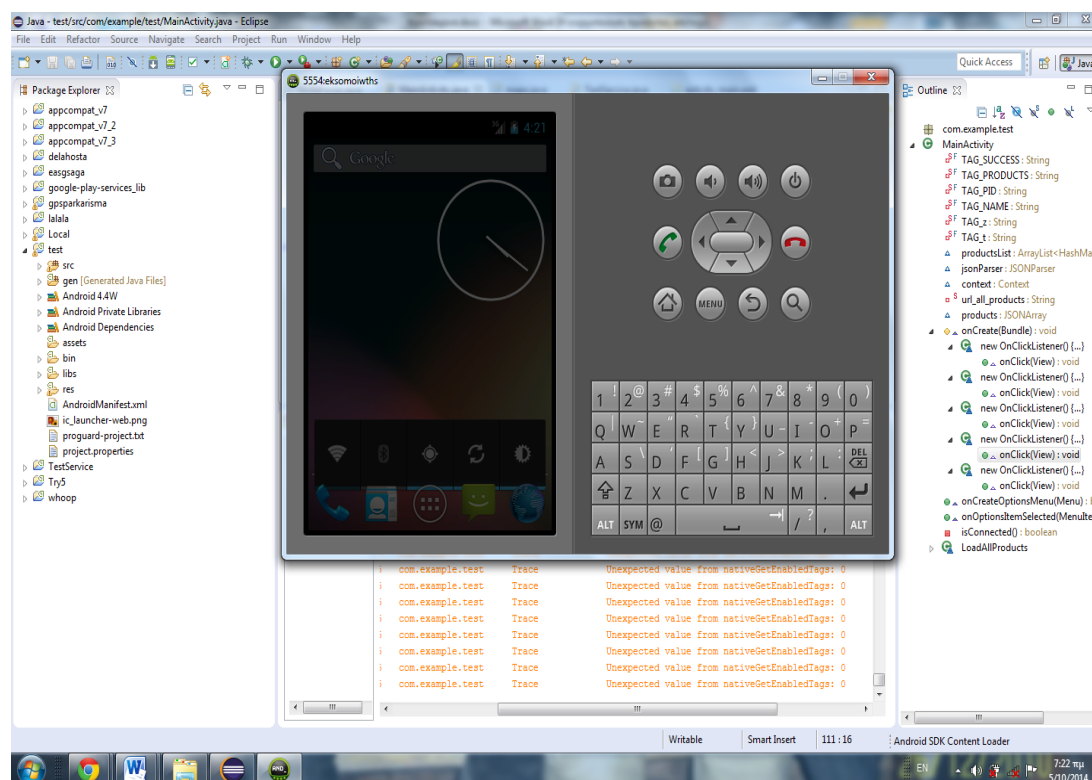
Επειδή το σύστημα εκτελεί κάθε εφαρμογή σαν μια ξεχωριστή διαδικασία με συγκεκριμένες άδειες που περιορίζουν την πρόσβαση σε άλλες εφαρμογές, οι εφαρμογές δεν μπορούν να ενεργοποιήσουν άμεσα ένα συστατικό στοιχείο από μία άλλη εφαρμογή, το σύστημα όμως μπορεί. Οπότε για να ενεργοποιηθεί ένα συστατικό μίας άλλης εφαρμογής θα πρέπει να παραδοθεί ένα μήνυμα προς το σύστημα που θα δείχνει την πρόθεσή της τρέχουσας εφαρμογής να ξεκινήσει ένα συγκεκριμένο συστατικό στοιχείο της άλλης έτσι ώστε το σύστημα να ενεργοποιήσει το συγκεκριμένο στοιχείο.



## Android SDK

Το πρώτο βήμα στην πορεία ανάπτυξης της εφαρμογής είναι η εγκατάσταση και ρύθμιση του Android SDK. Το Android SDK παρέχει τα εργαλεία που χρειάζονται για την ανάπτυξη εφαρμογών στην πλατφόρμα Android. Περιλαμβάνει παραδείγματα εφαρμογών με τον πηγαίο τους κώδικα, εργαλεία ανάπτυξης, ένα εξομοιωτή και τις απαιτούμενες βιβλιοθήκες για την ανάπτυξη των εφαρμογών στη γλώσσα προγραμματισμού Java. Ακόμη αναλαμβάνει τη μεταγλώττιση του πηγαίου κώδικα ώστε να τρέχει στην εικονική μηχανή Dalvik.

Οι εξομοιωτές του android (Android Virtual Device – AVD) επιτρέπουν τον ορισμό διαφόρων πτυχών του υλικού του εικονικού τηλεφώνου και επιτρέπουν τη δημιουργία πολλών συνθέσεων ώστε να υπάρχει ποικιλία παραλλαγών υλικού διασφαλίζοντας την ομαλή λειτουργία της εφαρμογής σε πολλούς τύπους συσκευών Android. Όταν η εφαρμογή λειτουργεί με τον εξομοιωτή, μπορεί να χρησιμοποιήσει τις υπηρεσίες της πλατφόρμας Android για να επικαλεστεί άλλες εφαρμογές, να έχει πρόσβαση στο δίκτυο, να αναπαράγει ήχο και βίντεο, να αποθηκεύει και να ανακτά δεδομένα, να ενημερώνει το χρήστη και άλλα. Ο εξομοιωτής περιλαμβάνει επίσης μια ποικιλία από δυνατότητες εντοπισμού σφαλμάτων, όπως μια κονσόλα στην οποία καταγράφονται τα μηνύματα εξόδου του πυρήνα, προσομοίωση τυχόν διακοπών της εφαρμογής (όπως όταν φθάνουν μηνύματα SMS ή τηλεφωνικές κλήσεις), καθώς και απορρίψεις στο δίκτυο δεδομένων.



**Εικόνα 11 : Android Virtual Device μέσα από το SDK του Eclipse**

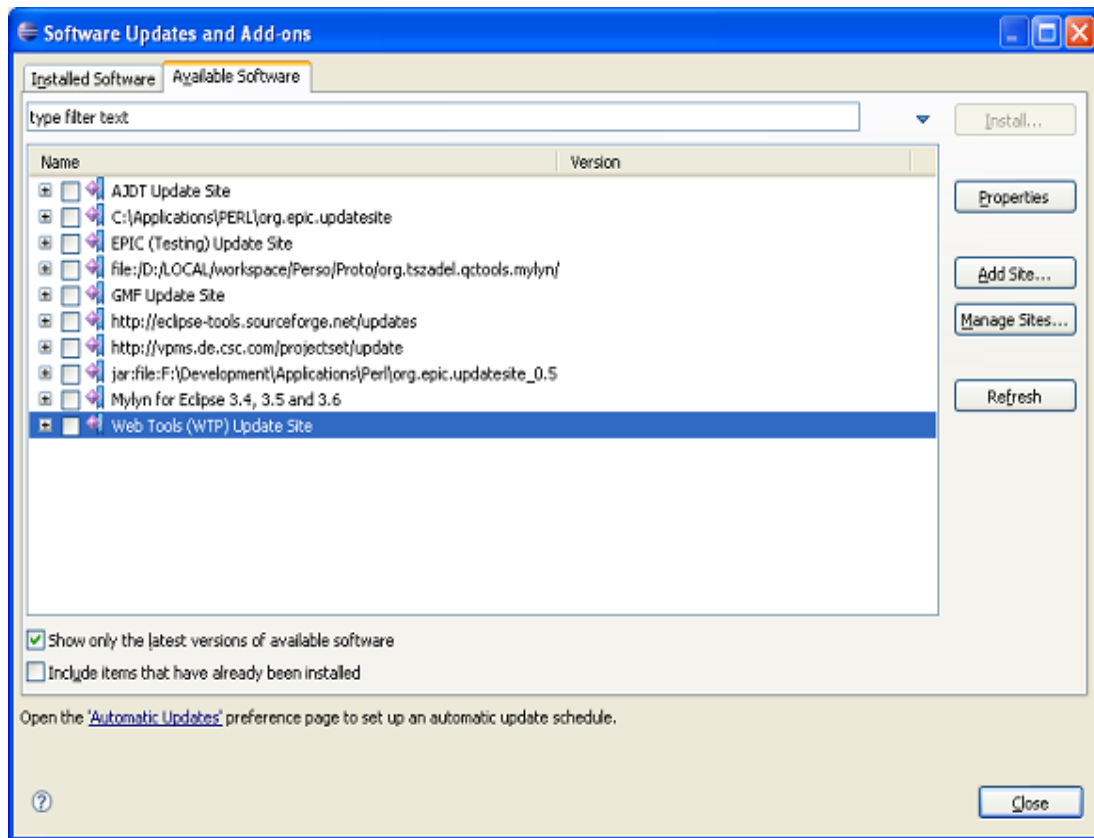
### 4.1.3 Επιλογή περιβάλλοντος προγραμματισμού (eclipse)

Οι περισσότερες εφαρμογές στο Android βασίζονται στην γλώσσα προγραμματισμού Java. Έτσι, ο κάθε προγραμματιστής μπορεί να χρησιμοποιήσει έναν οποιονδήποτε text editor για να γράψει τον κώδικα και μετέπειτα να μεταγλωττίσει τα αρχεία μέσω γραμμής εντολών. Ο συγκεκριμένος τρόπος ανάπτυξης δεν είναι ιδιαίτερα φιλικός προς το χρήστη γι' αυτό συνίσταται η χρήση ενός ολοκληρωμένου περιβάλλοντος ανάπτυξης - IDE (Integrated Development Environment) που να υποστηρίζει Java, όπως το Eclipse.

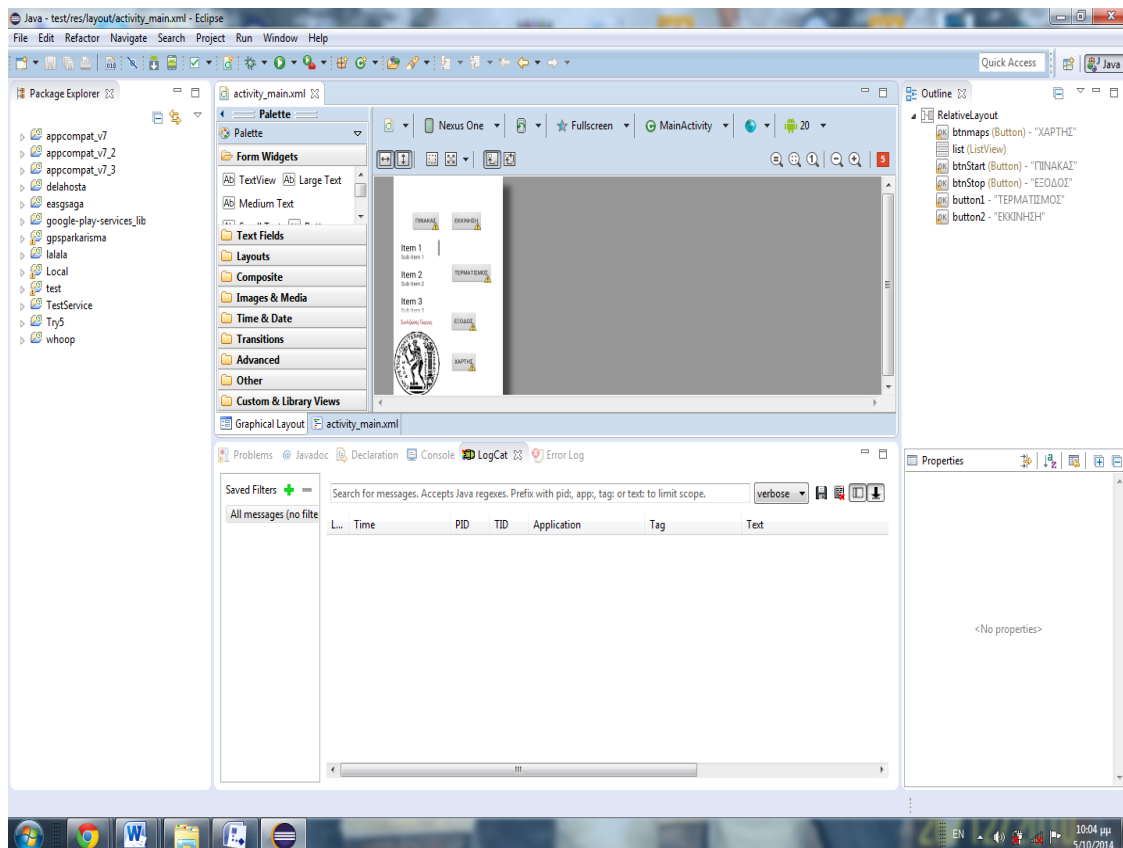
Το Eclipse είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης λογισμικού με δυνατότητα εύκολης επεκτασιμότητας χάρη στο σύστημα εργαλείων που διαθέτει. Είναι γραμμένο κυρίως στη γλώσσα προγραμματισμού Java και μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών σε Java και μέσω των διαφόρων εργαλείων μπορεί να υποστηρίξει και επιπλέον γλώσσες προγραμματισμού όπως C, C++, Perl, PHP, Python, Ruby και άλλες. Το Eclipse είναι μια εφαρμογή Ελεύθερου Λογισμικού καθώς διατίθεται κάτω από την άδεια Eclipse Public License. Αναπτύσσεται ραγδαία από την παγκόσμια κοινότητα του Ελεύθερου Λογισμικού ενώ χρηματοδοτείται από το ίδρυμα Eclipse και από διάφορες δωρεές.

Το ολοκληρωμένο περιβάλλον ανάπτυξης (integrated development environment) που χρησιμοποιήθηκε για την εφαρμογή ήταν το eclipse και θεωρήθηκε το καλύτερο στην συγκεκριμένη περίπτωση (χωρίς να περιλαμβάνεται το android studio σε αυτή την σύγκριση διότι δεν είχε βγει μέχρι πρόσφατα). Παρόλο που δέχεται αρκετές αρνητικές κριτικές η ύπαρξη ενός συστήματος διαχείρισης ανανέωσης λογισμικού, θεωρώ πως είναι απαραίτητη. Μόνο η ικανότητα προσθήκης μιας ιστοσελίδας για αναβάθμιση (για λήψη καινούργιων εκδόσεων) της πλατφόρμας και των επεκτάσεων της διευκολύνει την διαδικασία ανάπτυξης της εφαρμογής.

Το eclipse έχει επίσης κάποια από τα καλύτερα εργαλεία διαχείρισης τόσο για τα Java projects όσο και για το γραφικό περιβάλλον τους (user interface). Οι επεξεργαστές αρχείων Java και xml είναι εξαιρετικά λειτουργικοί. Περιλαμβάνει επίσης το πρόσθετο λογισμικό ADT (Android Developer Tools), το οποίο συνδέει το Eclipse με το Android SDK (ένα ολοκληρωμένο σετ ανάπτυξης και επιδιόρθωσης εφαρμογών android) και όλες τις δυνατότητες του.



**Εικόνα 12 : : Διαχείριση διαθέσιμων αναβαθμίσεων του Eclipse**



**Εικόνα 13 : Το διαχειριστικό περιβάλλον του Eclipse**

## 4.1.4 Γλώσσα προγραμματισμού – Java

Επιλέχθηκε Java ως γλώσσα ανάπτυξης αντί για c++ διότι ήταν η καταλληλότερη για τον σκοπό της εφαρμογής αφού παρέχει μεγάλες διευκολύνσεις και είχε επιλεγεί από την google που αγόρασε την android inc (δημιουργό του λειτουργικού) ως το επίσημο εργαλείο ανάπτυξης παρόλο που το μεγαλύτερο μέρος του λειτουργικού συστήματος android είχε γραφτεί σε C.

Η Java είναι μια γλώσσα προγραμματισμού που μπορεί να κάνει παράλληλους υπολογισμούς, στηρίζεται σε κλάσεις, είναι αντικειμενοστραφής, και ειδικά σχεδιασμένη για να έχει όσο το δυνατόν λιγότερες εξαρτήσεις γίνεται. Σκοπός της είναι να επιτρέπει στους προγραμματιστές "write once, run anywhere" (WORA), πράγμα που σημαίνει ότι άμα ο κώδικας τρέχει σε μια πλατφόρμα δεν χρειάζεται να επανασυντεθεί για να τρέξει σε άλλη. Οι εφαρμογές Java συνήθως μεταφράζονται σε bytecode που μπορεί να τρέξει σε οποιαδήποτε εικονική μηχανή Java (Java Virtual Machine), ανεξάρτητα από την αρχιτεκτονική του υπολογιστή. Η Java είναι, από το 2014, μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού που χρησιμοποιούνται, ιδιαίτερα για διαδικτυακές εφαρμογές τύπου εξυπηρετητή – πελάτη (client-server), με περίπου 9 εκατομμύρια χρήστες.

Όσον αφορά την ιστορία της, η Java αναπτύχθηκε αρχικά από τον James Gosling της εταιρείας Sun Microsystems (η οποία έκτοτε έχει συγχωνευθεί με την Oracle Corporation) και κυκλοφόρησε το 1995 ως βασική συνιστώσα της πλατφόρμας Java της Sun Microsystems. Η σύνταξη της γλώσσας αυτής έχει πάρει ένα μεγάλο μέρος των κανόνων της σύνταξης των γλωσσών προγραμματισμού C και C ++, αλλά έχει λιγότερα στοιχεία χαμηλού επιπέδου (π.χ. γλώσσας μηχανής) από αυτές.

Οι αρχικοί μεταγλωττιστές Java, οι εικονικές μηχανές (VM), και οι βιβλιοθήκες κλάσης αναπτύχθηκαν από την Sun το 1991 και κυκλοφόρησαν για πρώτη φορά το 1995. Από το Μάιο του 2007, σύμφωνα με τις προδιαγραφές του Java Community Process (όπου ανήκουν ενδιαφερόμενα μέλη για την ανάπτυξη τεχνικών προδιαγραφών για την Java), η Sun ανέκτησε το μεγαλύτερο κομμάτι των τεχνολογιών της Java υπό τους όρους του GNU General Public License (για την ελεύθερη χρήση από όλους). Στην συνέχεια αναπτύχθηκαν από άλλους εναλλακτικές εφαρμογές των τεχνολογιών αυτών της Sun, όπως το GNU Compiler για Java (bytecode compiler), το GNU Classpath (βιβλιοθήκες για κλάσεις), και το IcedTea-Web (Java web browser plugin).

Παράδειγμα κλάσης java με την μέθοδο onCreate():

```
//στην αρχή μπαίνει το όνομα του φακέλου της εφαρμογής μας
package com.example.myapplication;
//μετά μπαίνουν τα απαραίτητα εξωτερικά στοιχεία που θα χρησιμοποιηθούν στην κλάση
//και είναι απαραίτητα για να τρέξει η εφαρμογή
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.support.v4.app.NavUtils;
import android.view.MenuItem;
```

```
//δήλωση της κλάσης και της ιδιότητας της
public class My_activity extends Activity {

    //στην συνέχεια ορίζουμε τι θα γίνεται κατά την έναρξη της κλάσης
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //ορισμός της θέσης της οθόνης που θα βλέπει ο χρήστης έτσι όπως
        //βρίσκεται στον φάκελο με τα υπόλοιπα layouts
        setContentView(R.layout.activity_display_message);
        //ανάπτυξη των διαδικασιών που θα γίνονται
        //.....
    }
}
```

#### 4.1.5 Η γλώσσα XML (Extensible Markup Language) Εκτεταμένη Γλώσσα Σήμανσης

Σε ένα κόσμο όπου οι πληροφορίες παρέχονται μέσω του παγκόσμιου διαδικτύου, τα έγγραφα πρέπει να είναι εύκολα προσβάσιμα, μεταφέρσιμα και ευέλικτα. Πρέπει επίσης να είναι ανεξάρτητα οποιουδήποτε συστήματος και περιεχομένου. Οι γενικευμένες γλώσσες έχουν τέτοια χαρακτηριστικά, παρέχοντας στα έγγραφα αυτά μια δυνατότητα η οποία δεν υπάρχει σε άλλες γλώσσες περιγραφής εγγράφων. Η XML δεν είναι μία σημειακή γλώσσα όπως η γλώσσα δημιουργίας ιστοσελίδων HTML, είναι μία γλώσσα που χρησιμοποιείται για την περιγραφή μίας σημειακής γλώσσας. Ο τεχνικός όρος μιας τέτοιας γλώσσας είναι μετα-γλώσσα. Η HTML είναι προβληματική και περιοριστική γλώσσα. Η XML έλυσε πολλά από τα προβλήματα που αντιμετώπισαν οι σχεδιαστές του διαδικτύου και είναι υπεύθυνη για την XHTML, μια ανασχεδιασμένη HTML. Η XML συμπληρώνει και δεν αντικαθιστά την HTML. Ενώ η HTML χρησιμοποιείται στη διατύπωση και την εμφάνιση των δεδομένων η XML αναπαριστά τη συναφή έννοια των δεδομένων.

Η XML είναι, δηλαδή, ένα σύνολο κανόνων (ή διαφορετικά ένα πακέτο κατευθυντήριων γραμμών ή συμβάσεων) για το σχεδιασμό μορφών κειμένου οι οποίες διευκολύνουν τη δόμηση των δεδομένων. Η XML δεν είναι γλώσσα προγραμματισμού. Δεν χρειάζεται, επομένως, να είναι κανείς προγραμματιστής για να την χρησιμοποιήσει ή να τη μάθει. Η XML διευκολύνει τον υπολογιστή να παράγει δεδομένα, να διαβάζει δεδομένα και να εξασφαλίζει τη σαφήνεια της δομής των δεδομένων. Η XML αποφεύγει τις συνήθεις παγίδες του σχεδιασμού γλωσσών: είναι επεκτάσιμη, ανεξάρτητη συστήματος υλικού και μπορεί να υποστηρίξει διεθνείς και τοπικές προσαρμογές. Η XML είναι πλήρως συμβατή με Unicode. Στην HTML οι ετικέτες (tags) είναι προκαθορισμένες ενώ η XML παρέχει τη δυνατότητα να καθορίζουν οι χρήστες τις ετικέτες και τις δομημένες μεταξύ τους σχέσεις.

Θα χρησιμοποιείται για πολλά χρόνια επειδή προσφέρει αποτελεσματικές και δυναμικές πολυμεσικές λύσεις.

Όλα τα XML έγγραφα πρέπει να έχουν τα εξής χαρακτηριστικά :

- Να ορίζουν τα δεδομένα με μια δομημένη μορφή, σύμφωνα με το εξής στυλ : `<tag>δεδομένα </tag>`.
- Να έχουν επέκταση αρχείου `.xml`.
- Να μπορούν να δημιουργηθούν μ' έναν κειμενογράφο ή μ' έναν XML επεξεργαστή.
- Να μεταφέρονται από τον εξυπηρετητή ιστού (Web server ) στον πελάτη ιστού (Web client) μέσω του πρωτοκόλλου μεταφοράς υπερκειμένου (HTTP).
- Να αποτελούν απαραίτητο στοιχείο μιας εφαρμογής, η οποία περιλαμβάνει επιπλέον ένα προαιρετικό ορισμό τύπου εγγράφου(DTD), έναν XML αναλυτή (parser) και μια λογική παρουσίαση/επεξεργασία.

## 4.1.6 Αρχείο MANIFEST.xml

Πριν ξεκινήσει μία εφαρμογή, το σύστημα πρέπει να γνωρίζει ότι υπάρχει στην εφαρμογή το αρχείο `AndroidManifest.xml`, το πιο βασικό αρχείο της εφαρμογής. Σε κάθε εφαρμογή πρέπει να δηλώνονται όλα τα συστατικά στοιχεία της σε αυτό το αρχείο , το οποίο θα πρέπει να βρίσκεται στον αρχικό φάκελο της εφαρμογής .

Το αρχείο αυτό πέρα από τη δήλωση των συστατικών στοιχείων της εφαρμογής χρησιμοποιείται για:

- Τον προσδιορισμό τυχόν δικαιωμάτων χρήστη που απαιτεί η εφαρμογή , όπως πρόσβαση στο Διαδίκτυο ή πρόσβαση στις επαφές του χρήστη.
- Την δήλωση του ελάχιστου επίπεδου συστήματος της android συσκευής που απαιτείται για χρήση της εφαρμογής.
- Την δήλωση των χαρακτηριστικών λογισμικού και υλικού που απαιτούνται από την εφαρμογή, όπως κάμερα , bluetooth, οθόνη πολυαφής κλπ.
- Αναφορά των βιβλιοθηκών με τις οποίες η εφαρμογή θα πρέπει να συνδέεται όπως η βιβλιοθήκη του Google Maps .

και πολλά άλλα.

Η γενική δομή του αρχείου αυτού και τα στοιχεία που μπορεί να περιέχει φαίνονται στην συνέχεια. Κάθε στοιχείο μαζί με όλα του τα υπόλοιπα γνωρίσματα αποθηκεύεται σε ξεχωριστό αρχείο.

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

```
<manifest>
```

```

<uses-permission />
<permission />
<permission-tree />
<permission-group />
<instrumentation />
<uses-sdk />
<uses-configuration />
<uses-feature />
<supports-screens />
<compatible-screens />
<supports-gl-texture />

<application>

  <activity>
    <intent-filter>
      <action />
      <category />
      <data />
    </intent-filter>
    <meta-data />
  </activity>
  <activity-alias>
    <intent-filter> . . . </intent-filter>
    <meta-data />
  </activity-alias>

  <service>
    <intent-filter> . . . </intent-filter>
    <meta-data />
  </service>

  <receiver>
    <intent-filter> . . . </intent-filter>
    <meta-data />
  </receiver>

  <provider>
    <grant-uri-permission />
    <meta-data />
    <path-permission />
  </provider>
  <uses-library />
</application>

</manifest>

```

Όλα τα στοιχεία που μπορούν να υπάρξουν στο αρχείο manifest παρατίθενται με αλφαβητική σειρά:

```

<action>
<activity>
<activity-alias>
<application>
<category>

```

```
<data>
<grant-uri-permission>
<instrumentation>
<intent-filter>
<manifest>
<meta-data>
<permission>
<permission-group>
<permission-tree>
<provider>
<receiver>
<service>
<supports-screens>
<uses-configuration>
<uses-feature>
<uses-library>
<uses-permission>
<uses-sdk>
```

## 4.1.7 Layouts : Διατάξεις – χωροταξία των αντικειμένων

Τα layouts είναι στην ουσία η δομή της διεπαφής (User Interface) της εφαρμογής δηλαδή το περιβάλλον χρήσης της. Αυτό είναι διαχειρίσιμο είτε ορίζοντας τα στοιχεία της εφαρμογής σε XML (Extensible Markup Language) ώστε να ανεβάζει κανείς αρχεία στο διαδίκτυο χρησιμοποιώντας το δικό του λεξιλόγιο για την επικοινωνία με τις κλάσεις είτε φτιάχνοντας τα στοιχεία του περιβάλλοντος αυτού προγραμματιστικά μέσα στην κλάση από την εκκίνηση της εφαρμογής . Κλάση είναι το υπερσύνολο κοινών αντικειμένων π.χ. αν ο αντικείμενο είναι ρόδα από ποδήλατο τότε το ποδήλατο είναι μια κλάση, το ίδιο συμβαίνει και προγραμματιστικά μία κλάση μπορεί να περιλαμβάνει μία σειρά αντικειμένων ορισμένων σε αυτή που εξυπηρετούν τον σκοπό της. Χρησιμοποιήθηκαν και οι δύο τρόποι για λόγους διευκόλυνσης ανάλογα με το κάθε αντικείμενο. Τα αντικείμενα τα οποία ήθελαν ξεχωριστή μεταχείριση ορίστηκαν στο XML για να έχουν την δικιά τους οντότητα μαζί με το σύνολο της διεπαφής (User interface) για να γίνει ευκολότερη επεξεργασία τους για διάφορα μεγέθη και είδη οθονών, ενώ τα υπόλοιπα στη JAVA.

Στα layouts το κάθε αντικείμενο τους ορίζεται στο XML αρχείο με τη δική του ταυτότητα (ID), έχει τα δικά του ορίσματα και καλείται από την κλάση για την οποία αναπτύχθηκε το layout.

Για παράδειγμα στη γλώσσα xml, αντικείμενο μπορεί να είναι ένα κουμπί με τα δικά του χαρακτηριστικά στοιχεία:



```
<Button android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello, I am a Button" />
```

Για την χωροταξία των αντικειμένων στην οθόνη υπάρχουν τρία βασικά layouts.

### **LINEAR LAYOUT**



Οργανώνει τα αντικείμενα του σε οριζόντιες και κάθετες σειρές. Και δημιουργεί αυτόματα μια κυλιόμενη μπάρα (scrolling bar) σε περίπτωση που τα αντικείμενα δεν χωράν όλα στην οθόνη.

### **RELATIVE LAYOUT**



Επιτρέπει την τοποθέτηση των αντικειμένων σχεσιακά μεταξύ τους (το αντικείμενο A δεξιά από το αντικείμενο B) ή σε σχέση με τα όρια της οθόνης (τοποθέτηση του A στην κορυφή της οθόνης ή σε συγκεκριμένη απόσταση από αυτή).

## WEB VIEW



Για εμφάνιση διαδικτυακών σελίδων.

Οπότε αν θέλαμε να βάλουμε το κουμπί μέσα σε ένα linear layout στο xml θα ήταν:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>
```

Τα layouts καλούνται από τις διαδικασίες (activities) όταν πρέπει να ενεργοποιούνται, δηλαδή αν θέλαμε με την εκκίνηση μίας διαδικασίας να ενεργοποιείται το παραπάνω layout θα αρκούσε να το καλέσουμε με το όνομα με το οποίο το αποθηκεύσαμε (έστω main\_layout) μέσα στην ίδια διαδικασία (διαδικασία.java)

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main_layout);
}
```

και αν θέλω να χρησιμοποιήσω το αντικείμενο μου (κουμπί στην προκείμενη) να κάνει κάτι τότε πρέπει να το ορίσω και αυτό πρώτα μέσα στην διαδικασία (διαδικασία.java)

```
Button myButton = (Button) findViewById(R.id.my_button);
```

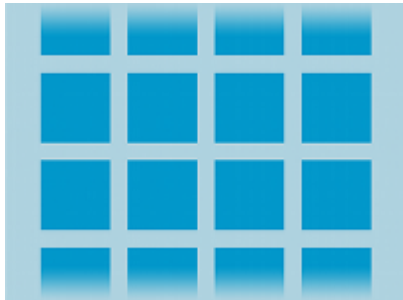
Τα αντικείμενα μέσα σε ένα layout πολλές φορές δεν είναι συγκεκριμένα όπως στην προκείμενη εφαρμογή τα δεδομένα του πίνακα της βάσης. Τότε συνήθως χρησιμοποιείται ένας άλλος τρόπος για να προσαρμόζονται τα αντικείμενα (adapter) ανάλογα με τον τρόπο που θες αυτά να εμφανίζονται.

### ***LIST VIEW***



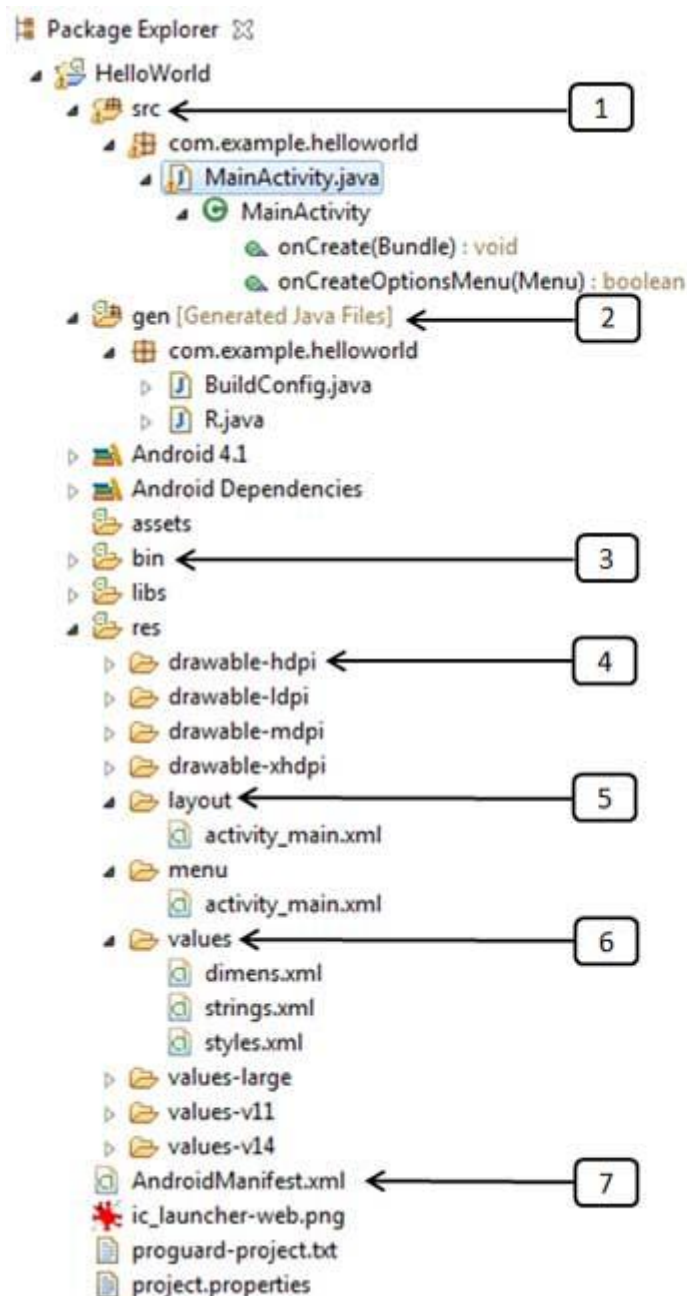
Εμφανίζει μία λίστα αντικειμένων με αυτόματη μπάρα κύλισης

### ***GRID VIEW***



Εμφανίζει έναν πίνακα αντικειμένων.

Συνοψίζοντας η δομή των αρχείων μιας εφαρμογής Android είναι:



**Εικόνα 14 : Η δομή των αρχείων μιας εφαρμογής Android**

| A/A | Φάκελος, αρχείο και περιγραφή   |
|-----|---|
| 1   | <b>src</b><br>περιλαμβάνει των πηγαία αρχεία <b>.java</b> . Εξ αρχής δημιουργείται μία κύρια κλάση Main_activity η οποία θα ενεργοποιείται με την εκκίνηση της εφαρμογής.         |
| 2   | <b>gen</b><br>περιέχει το αρχείο <b>.R</b> που παράγεται αυτόματα από τον compiler και αναφέρεται σε όλους τους πόρους της εφαρμογής. Δεν πρέπει να τροποποιείται από τον χρήστη. |

|   |   |
|---|---|
| 3 | <b>bin</b><br>ο φάκελος αυτός περιέχει το αρχείο <b>.apk</b> που δημιουργείται από τα ADT (android developer tools) και ότι άλλο χρειάζεται για να τρέξει η εφαρμογή. |
| 4 | <b>res/drawable-hdpi</b><br>φάκελος για εικόνες με μεγάλη ανάλυση που θα χρησιμοποιηθούν.   |
| 5 | <b>res/layout</b><br>εδώ καθορίζεται το user interface  |
| 6 | <b>res/values</b><br>φάκελος για άλλα αρχεία XML που περιλαμβάνουν μια συλλογή πόρων όπως καθορισμοί χρωμάτων ή λέξεων με strings.                                    |
| 7 | <b>AndroidManifest.xml</b><br>ο φάκελος με τα συστατικά στοιχεία της εφαρμογής.   |

## 4.1.8 Οι εξυπηρετητές Ιστού (Web Servers)

Για αρχή ήταν επιθυμητό να υπάρχει επικοινωνία μεταξύ android smartphone με μία απομακρυσμένη βάση που θα συγκεντρώνει δεδομένα από τους αισθητήρες. Η εφαρμογή δοκιμάστηκε τοπικά, στέλνοντας τα δεδομένα σε τοπικό υπολογιστή για αρχή, που ορίστηκε να λειτουργεί ως τοπικός εξυπηρετητής ή διακομιστής ιστού (web server) ο apache web server. Ο apache web server έπρεπε να αποθηκεύει την MySQL βάση και τα web services (αρχεία PHP απαραίτητα για απομακρυσμένη επικοινωνία).

Για αυτό εγκαταστάθηκε το xampp, ένα πακέτο προγραμμάτων ελεύθερου λογισμικού, λογισμικού ανοικτού κώδικα και ανεξαρτήτου πλατφόρμας που συμπεριλαμβάνει τον εξυπηρετητή ιστού http Apache, την βάση δεδομένων MySQL και έναν διερμηνέα για κώδικα γραμμένο σε γλώσσα προγραμματισμού PHP. Στον τοπικό υπολογιστή φτιάχτηκε μία βάση (android2) με έναν πίνακα (mobile2\_data) με έξι πεδία, και αποθηκεύθηκαν τα απαραίτητα PHP αρχεία για την επικοινωνία (db\_config, db\_connect) και τα PHP αρχεία που περιλάμβαναν τα βασικά ερωτήματα (queries) για τις CRUD λειτουργίες (create, read, update, delete) για εγγραφή, εμφάνιση, επεξεργασία και διαγραφή μίας ή περισσότερων πληροφοριών στον πίνακα της βάσης.

Αφού ήταν σίγουρο ότι όλα λειτουργούν σωστά έγινε το ίδιο με μία βάση MySQL στο διαδίκτυο σε απομακρυσμένο εξυπηρετητή ιστού (web server) όπου και έγινε η ίδια διαδικασία και δεν χρειάστηκε καμία αλλαγή στον πηγαίο κώδικα (source code) του προγράμματος στη java προφανώς εκτός της διεύθυνσης (URL) την οποία καλούσε (από <http://192.168.2.1/Z:/...> Σε [http://www.eretria24.gr/mobile2/create\\_acceldata.php](http://www.eretria24.gr/mobile2/create_acceldata.php)). Στην συνέχεια θα εξηγήσουμε γιατί είναι απαραίτητα τα PHP και γιατί μία εφαρμογή συσκευής android καλεί αυτά και δεν μπορεί να κάνει από μόνη της τις απαραίτητες crud(create, read, update, delete) λειτουργίες.

## 4.1.9 Επικοινωνία εξυπηρετητή ιστού με την αναπτυχθείσα εφαρμογή (PHP και JSON Parser)

Πρέπει να σημειωθεί πως μια εφαρμογή android και μια απομακρυσμένη βάση δεδομένων MySQL δεν μπορούν να επικοινωνούν μεταξύ τους απευθείας (χωρίς να είναι μια απίστευτα κουραστική διαδικασία). Θα ήταν σαν να προσπαθούν να επικοινωνήσουν δύο άτομα που μιλάνε διαφορετική γλώσσα. Δεν θέλουμε να παίξουν παντομίμα έτσι αντ' αυτού, υπάρχει ένας «διερμηνέας». Αυτός ο «διερμηνέας» που λειτουργεί σαν μέσο επικοινωνίας του κινητού και της βάσης δεδομένων είναι η γλώσσα προγραμματισμού php για την επεξεργασία ή δημιουργία ιστοσελίδων με δυναμικό περιεχόμενο. Ο php κώδικας μεταφράζεται στον εξυπηρετητή ή διακομιστή ιστού (web server) και δημιουργεί HTML ή άλλη έξοδο που θα δει ο επισκέπτης. Για παράδειγμα όταν ο χρήστης συμπληρώνει κάποιες πληροφορίες σε μία εφαρμογή και πατάει να τις υποβάλει για να ανέβουν σε μία βάση, θα περάσει τις πληροφορίες στην υπηρεσία web PHP πρώτα . Η διαδικτυακή αυτή υπηρεσία θα συνδεθεί με τη βάση δεδομένων MySQL , και θα βρει τον κατάλληλο πίνακα για να καταχωρηθούν οι πληροφορίες στη βάση δεδομένων MySQL σύμφωνα με τις πληροφορίες που στάλθηκαν από την Android συσκευή μας. Μόλις καταχωρηθούν , το webservice (στην PHP) θα μας στείλει κάποια δεδομένα για να μας πει αν καταχωρήθηκαν τα δεδομένα με επιτυχία ή όχι. Η Android συσκευή μας θα αναλύσει αυτά τα δεδομένα JSON (javascript object notation) μέσω ενός αποκωδικοποιητή JSON parser (του οποίου τον κώδικα περιλαμβάνουμε στην εφαρμογή) ώστε να μπορεί να εμφανιστεί ένα μήνυμα "επιτυχίας" , ή "αποτυχίας" της εγγραφής στην οθόνη της συσκευής μας.

Η μετάδοση όμως πληροφοριών από την android συσκευή προς την βάση mysql μέσω αρχείων PHP και αντιστρόφως χρειάζεται έναν αποκωδικοποιητή δεδομένων. Ο αποκωδικοποιητής JSON Parser στην ουσία μετατρέπει σε javascript objects notations τις ενδείξεις επιτυχίας ή αποτυχίας που μας επιστρέφει ο server όταν στέλνουμε πληροφορίες σε αυτόν αφού τις (ενδείξεις) έχει αναλύσει και το ίδιο κάνει για τα δεδομένα που τραβάμε από την βάση μέσω της PHP. Ουσιαστικά περιλαμβάνει μία βασική μέθοδο, την makeHttpRequest όπου απλά καλεί τα ανάλογα PHP αρχεία. Αυτή περιλαμβάνει δύο άλλες μεθόδους, την POST με την οποία όταν στέλνουμε πληροφορίες στον server λαμβάνουμε μήνυμα για το αν το καταφέραμε επιτυχώς και τη GET με την οποία λαμβάνουμε πληροφορίες από τον server και την βάση όπου και πάλι λαμβάνουμε αντίστοιχο μήνυμα.



Εικόνα 15 : Η διαδικασία σε οπτική μορφή

Παρόλα αυτά πρέπει να σημειωθεί πως για να επικοινωνήσει η εφαρμογή μας με έναν εξυπηρετητή ιστού web server δεν είναι απαραίτητη η χρήση 'διερμηνέα' αρκεί να μιλάει την ίδια γλώσσα με την βάση δεδομένων. Δηλαδή θα αρκούσε μία NoSQL (Not only SQL) μη σχεσιακή βάση δεδομένων που καταλάβαινε JSON όπως η mongodb, η couchdb και η elasticsearch (που έχουν την δυνατότητα αποθήκευσης εγγράφων JSON-style) αλλά θεωρήθηκε σωστό να διαφαινεται η ανεξαρτησία της εφαρμογής (stand alone) ανεξαρτήτως με το είδος της βάσης για να γίνει κατανοητό ότι μπορεί να γίνει χρήση υπαρχόντων βάσεων και επεξεργασίας των δεδομένων τους από νέες εφαρμογές δηλαδή κάθε είδους βάση με πληροφορίες από αισθητήρες θα ήταν εύκολα αναλύσιμη και επεξεργάσιμη από μία συσκευή android για την εξαγωγή επιθυμητών συμπερασμάτων.

### Παραδείγματα αρχείων php.

Δημιουργία βάσης δεδομένων με php.

Η CREATE DATABASE δήλωση χρησιμοποιείται για να δημιουργήσει μια βάση δεδομένων σε MySQL. Θα πρέπει να προσθέσετε την CREATE DATABASE δήλωση στην λειτουργία mysqli\_query () για να εκτελέσει την εντολή.

Το ακόλουθο παράδειγμα δημιουργεί μια βάση δεδομένων που ονομάζεται "my\_db":

```
<?php
$con=mysqli_connect("example.com","peter","abc123");
// Έλεγχος σύνδεσης με τον εξυπηρετητή
if (mysqli_connect_errno()) {
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
}

// Δημιουργία της βάσης my_db
$sql="CREATE DATABASE my_db";
if (mysqli_query($con,$sql)) {
    echo "Database my_db created successfully";
} else {
    echo "Error creating database: " . mysqli_error($con);
}
?>
```

### **Δημιουργία πίνακα με php.**

Η CREATE TABLE δήλωση χρησιμοποιείται για να δημιουργήσετε έναν πίνακα σε MySQL. Πρέπει να προσθέσετε την CREATE TABLE δήλωση στην λειτουργία mysqli\_query () για να εκτελέσει την εντολή.

Το ακόλουθο παράδειγμα δημιουργεί έναν πίνακα που ονομάζεται " Persons", με τρεις στήλες: "FirstName", " LastName " και " Age ":

```

<?php
$con=mysqli_connect("example.com","peter","abc123","my_db");
// Check connection
if (mysqli_connect_errno()) {
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
}

// Δημιουργία πίνακα
$sql="CREATE TABLE Persons(FirstName CHAR(30),LastName CHAR(30),Age INT)";

// Execute query
if (mysqli_query($con,$sql)) {
    echo "Table persons created successfully";
} else {
    echo "Error creating table: " . mysqli_error($con);
}
?>

```

#### 4.1.10 Χρήση πολλαπλών αισθητήρων στην εφαρμογή

Αφού έγινε εφικτή η επικοινωνία μεταξύ android συσκευής και server για αρχή έγινε προσπάθεια ανίχνευσης και μεταφοράς στον server κάποιον ενεργειών. Ορίστηκε το επιταχυνσιόμετρο ως κύριος άξονας και έγινε εφικτή η αναγνώριση βασικών κινήσεων όπως όταν ο χρήστης αναποδογυρίζει το κινητό. Ένας μετρητής (μεταβλητή) αυξανόταν κατά ένα κάθε φορά που το επιταχυνσιόμετρο λάμβανε αρνητικές τιμές στον άξονα z.

Θέλαμε όμως να βεβαιωθούμε ότι μπορεί να χρησιμοποιηθεί και το gps ταυτόχρονα προτού προβούμε στην μοντελοποίηση της διαδικασίας στάθμευσης. Όποτε δοκιμάστηκε επιτυχώς να στέλνονται οι γεωγραφικές συντεταγμένες κάθε φορά που ο μετρητής αναποδογυρίσματος του κινητού έφτανε το δέκα.

Αφού παρατηρήθηκε ότι είναι εφικτή η χρήση πολλών αισθητήρων και αφού ήταν γνωστό ότι θα ήταν απαραίτητο στην συνέχεια, προβήκαμε στην διαδικασία υπολογισμού ταχύτητας της συσκευής. Αυτή είναι υπολογίσιμη με δύο προσεγγιστικούς τρόπους.

- Ο ένας είναι με το gps ως πηλίκο μεταξύ της διαφοράς των αποστάσεων δύο διαδοχικών γεωγραφικών σημείων προς την διαφορά των χρόνων που λήφθηκαν τα

$$\frac{[(d)_1 - d_2]}{(t_1 - t_2)}$$

αντίστοιχα σημεία, δηλαδή  $u = \frac{[(d)_1 - d_2]}{(t_1 - t_2)}$  (d τα σημεία και t οι χρόνοι) όπου οι χρόνοι ήταν εύκολα μετρήσιμοι αφού κάθε φορά που το gps παίρνει μία τιμή αποθηκεύουμε και την χρονική εκείνη στιγμή σε μία μεταβλητή. Από την άλλη η διαφορά αποστάσεων d1-d2 είναι εύκολα υπολογίσιμη μόνο σε μοίρες αφού από το gps λαμβάνουμε μοίρες γεωγραφικού πλάτους και μήκους οπότε είναι αν x και y

το γεωγραφικό πλάτος και μήκος:  $[(d)_1 - d_2]^o = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$   
 όμως η απόσταση μοιρών διαφέρει ανάλογα με το γεωγραφικό μήκος της περιοχής



οπότε έπρεπε να γίνει ανάλυση για εύρεση μίας συνάρτησης για τον υπολογισμό της απόστασης σε μέτρα που αντιστοιχεί η απόσταση μίας μοίρας το οποία ήταν αρκετά μεγάλη διαδικασία οπότε πήραμε συγκεκριμένα για την Αθήνα την συσχέτιση μοίρας με μέτρα που είναι  $111\mu = 1^\circ$  στο γεωγραφικό πλάτος και  $87\mu = 1^\circ$  στο γεωγραφικό μήκος . Σύμφωνα με τις δοκιμές που έγιναν , η ακρίβεια του συγκεκριμένου τρόπου προσέγγισης της ταχύτητας είναι αρκετά καλή με ένα σφάλμα που φτάνει μέγιστο τα 18 km/h στο 99% των περιπτώσεων.

- Ο άλλος τρόπος είναι με υπολογισμό διαφοράς ταχύτητας με το επιταχυνσιόμετρο. Θεωρώντας  $a_x$   $a_y$   $a_z$  τις επιταχύνσεις στους τρεις άξονες που λαμβάνουμε και  $t$  τις χρονικές στιγμές τότε

$$du = \sqrt{[(a]_{x_1}(t_1 - t_2))^2 + [(a]_{y_1}(t_1 - t_2))^2 + [(a]_{z_1}(t_1 - t_2))^2}$$

λαμβάνοντας την παραδοχή ότι η επιτάχυνση είναι σταθερή ή αμελητέα μεταβαλλόμενη για το μικρό χρονικό διάστημα  $(t_1 - t_2)$  αφού αναφερόμαστε σε χρονικές διαφορές microsecond. Άρα  $u = u_o + du$  όπου  $u_o$  την στιγμή  $t_o = 0$  όπου θεωρούμε ότι οι τιμές  $a_x$   $a_y$   $a_z$  έχουν πολύ μικρή διακύμανση από την μια χρονική στιγμή στην άλλη έτσι ώστε να προσεγγίζουν τις τιμές που θα είχαν στην απόλυτη ακινησία της συσκευής. Έτσι μπορούμε με μεγάλη ασφάλεια να υποστηρίξουμε ότι η αρχική ταχύτητα θα ληφθεί μία στιγμή που θα υπάρξει απόλυτη ακινησία και θα ισχύει  $u_o = 0$ . Σύμφωνα με τις δοκιμές που έγιναν , η ακρίβεια του συγκεκριμένου τρόπου προσέγγισης της ταχύτητας περιλαμβάνει ένα συνεχώς αυξανόμενο σφάλμα που για να μηδενίζεται, κάθε φορά που θεωρείται το σώμα ακίνητο σύμφωνα με τα παραπάνω κριτήρια θέτουμε πάλι  $t_o = 0$  και  $u_o = 0$  . Η προσέγγιση αυτή μπορεί να βγάξει σφάλμα έως περίπου 25km/h μέσα σε μία ώρα.

Γενικώς η χρήση πολλαπλών πηγών δεδομένων με διαφορετικά ποσοστά ακρίβειας η κάθε μία , για την εξαγωγή ενός συμπεράσματος , είναι πολύπλοκη διαδικασία αν επιθυμούμε να γίνει η καλύτερη δυνατή ανάλυση τους όσων αφορά το συμπέρασμα. Για την εξαγωγή ενός συμπεράσματος ( όπως μία τιμή ταχύτητας στην προκειμένη) μέσα από μη ακριβείς διαδικασίες όπου δεν μπορεί κανείς να βγάλει ένα 100% ακριβές συμπέρασμα μπορεί να φανεί χρήσιμη η ποσοτικοποίηση της ακρίβειας του συμπεράσματος που προκύπτει από την κάθε πηγή δεδομένων ώστε να ελεγχθεί η συνολική ακρίβεια.

Δηλαδή στην προκειμένη αν θεωρήσουμε πως ο μέσος άνθρωπος έχει μια μέση ταχύτητα 7km/h τότε τα σφάλματα είναι αρκετά μεγάλα για να γίνει εξαγωγή συμπεράσματος. Αν όμως χωρίσουμε τα σφάλματα για την μέση ταχύτητα ακινησίας (0km/h) , περπατήματος (5km/h) και οδήγησης 60 (km/h) τότε:

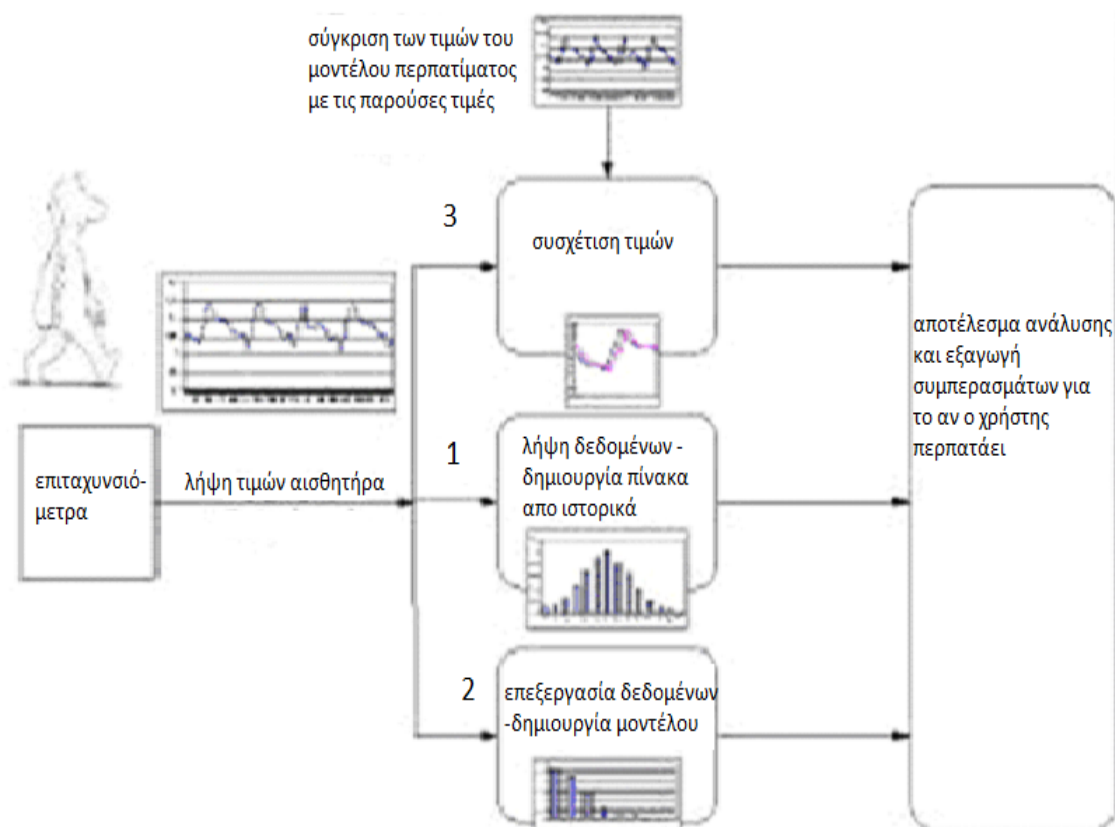
- Στην ακινησία ο δεύτερος τρόπος υπολογισμού ταχύτητας είναι πολύ ακριβής και αναγνωρίζεται περίπου στο 99% των περιπτώσεων.
- Στο περπάτημα η προσέγγιση της ταχύτητας δεν μπορεί να βρεθεί με μεγάλη ακρίβεια. Το GPS αν έχει σφάλμα 5 μ κάποια στιγμή τότε από το ένα δευτερόλεπτο στο άλλο η ταχύτητα θα μεταβληθεί κατά 5μ/s από την πραγματική. Το επιταχυνσιόμετρο επειδή δέχεται συνεχώς αυξομειώσεις στις τιμές που παίρνει σε

όλους τους άξονες το σφάλμα αυξάνεται πολύ πιο γρήγορα. Παρόλα αυτά επειδή το σφάλμα σύμφωνα με το GPS θα είναι στιγμιαίο προτιμάται ο πρώτος τρόπος προσέγγισης αφού θα έχει καλύτερο μέσο όρο ακρίβειας.

- Στην οδήγηση χρησιμοποιώντας το GPS προκύπτει η καλύτερη προσέγγιση της πραγματικής ταχύτητας αφού μία απότομα εσφαλμένη τιμή GPS θα εμφανίσει διαφορά  $5\mu/s = 18\text{km/h}$  από την πραγματική ταχύτητα αλλά η διαφορά αυτή είναι μικρότερη σε σχέση με την συνολική ταχύτητα και το σφάλμα θα είναι στιγμιαίο μέχρι να πάρει άλλες δύο τιμές το GPS (μερικά millisecond).

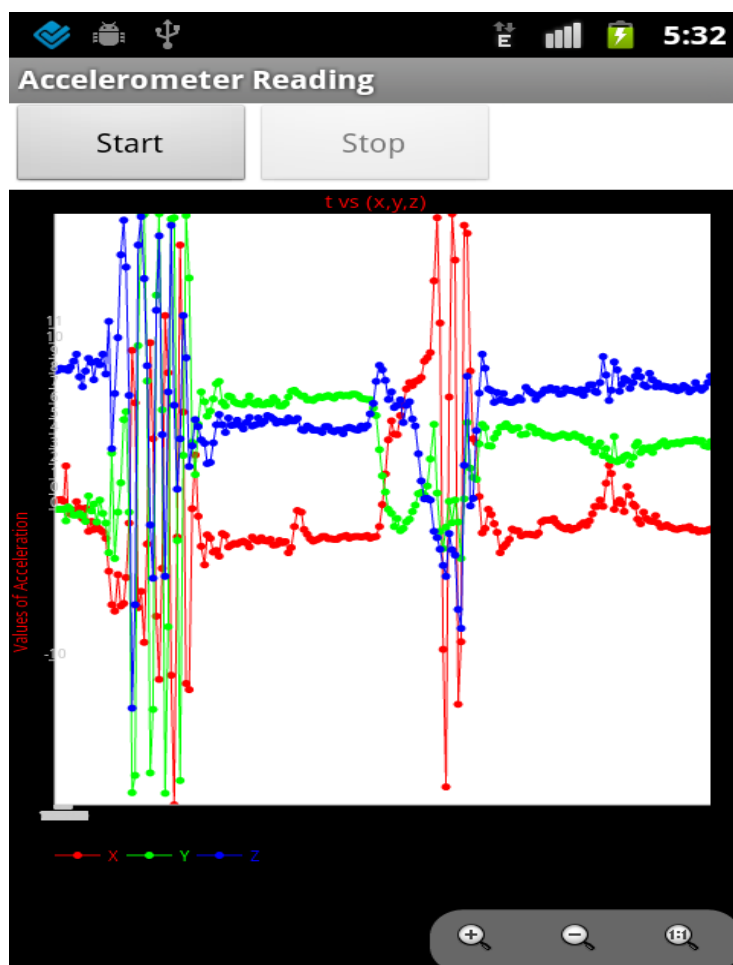
#### 4.1.11 Ανίχνευση ανθρώπινου βαδίσματος μέσω προτύπων αναγνώρισης χρησιμοποιώντας το επιταχυνσιόμετρο του έξυπνου τηλεφώνου (Detect human walk through pattern recognition using smartphone accelerometer)

Για την δημιουργία της εφαρμογής θεωρήθηκε απαραίτητο η ανίχνευση για το πότε ο χρήστης περπατάει.



Εικόνα 16 : Οπτική αναπαράσταση της διαδικασίας ανίχνευσης ανθρώπινου βαδίσματος με αισθητήρες

Για την αναγνώριση της διαδικασίας περπατήματος φτιάχτηκε μια βοηθητική εφαρμογή για να αποθηκεύονται τα δεδομένα του επιταχυνσιόμετρου και στους τρεις άξονες κάθε millisecond και την δημιουργία ενός γραφήματος με άξονες την επιτάχυνση που δέχεται η συσκευή και τον χρόνο.



**Εικόνα 17 : Εφαρμογή δημιουργίας γραφημάτων με επιταχυνσιόμετρο**

Έτσι αφού χρησιμοποιήθηκε αυτή η βοηθητική εφαρμογή από αρκετούς ανθρώπους ώστε να βρεθούν τα κοινά στοιχεία στο περπάτημα αρκούσαν μερικοί έλεγχοι πάνω σε αυτά τα κοινά στοιχεία που εμπεριέχει το περπάτημα όλων μας για να αναγνωρίζεται από την κύρια εφαρμογή μας αν και τότε ο χρήστης περπατάει η όχι.

Πιο συγκεκριμένα παρατηρήθηκε πως σε κάθε πάτημα ενός ποδιού, λόγω αδράνειας το κινητό έκανε μια μικρή ταλάντωση που ξεπερνούσε τα  $2.5 \text{ m/s}^2$  (κατώτατο όριο μέχρι και στο πιο ήρεμο περπάτημα) σε καθένα από τους τρεις άξονες και μετά ξαναέπεφταν οι ενδείξεις κοντά στο 0 μέχρι το πάτημα του άλλου ποδιού και η διαδικασία επαναλαμβανόταν.

Οπότε στην εφαρμογή γίνεται έλεγχος συνέχεια (κάθε microsecond) στα δεδομένα του επιταχυνσιόμετρου και για κάθε φορά που σε έναν άξονα περάσει η τιμή το  $2.5 \text{ m/s}^2$  (ενώ το προηγούμενο microsecond ήταν κάτω από 2.5) τότε υπάρχει ένας μετρητής που αυξάνει κατά ένα. Οπότε σε κάθε βήμα οι μετρητές των αξόνων αυξάνονται όλοι κατά ένα σύμφωνα

με την προαναφερθείσα παρατήρηση. Θεωρήθηκε επίσης πως για να καλύψω τα περισσότερα 'είδη' περπατήματος κάθε βήμα γίνεται σε λιγότερο από 1.2 δευτερόλεπτο οπότε κάθε 6 δευτερόλεπτα περπατήματος οι τιμές των μετρητών στους τρεις άξονες θα είναι από 5 και πάνω. Για αυτό κάθε 6 δευτερόλεπτα ελέγχεται αν οι τιμές των μετρητών και στους τρεις άξονες έχουν ξεπεράσει το 5, και αν ισχύει τότε θεωρούμε πως ο χρήστης περπατάει και υπάρχει ένας μετρητής που αυξάνει κάθε φορά που ο χρήστης μεταβαίνει από κατάσταση που δεν περπατάει σε κατάσταση που περπατάει και πάει συν ένα (θα χρησιμοποιηθεί αργότερα). Αν δεν ισχύει (ότι οι μετρητές πέρασαν το 5) οι τιμές των μετρητών μηδενίζονται και ξαναγίνεται έλεγχος για τα επόμενα 6 δευτερόλεπτα.

Επιλέχθηκε σαν χρόνος 6 δεύτερα γιατί είναι αρκετά σύντομος αλλά ταυτόχρονα αρκετός για 5 επαναλήψεις, ώστε να παραλειφθούν τυχαία λάθη όπως να πάρουν κάποια φορά οι μετρητές συν ένα από κάποια άλλη κίνηση.

#### **4.1.12 Αναγνώριση οδήγησης (Driving recognition)**

Στην συνέχεια θεωρήθηκε απαραίτητο να ανιχνεύεται εάν ο χρήστης βρίσκεται σε αυτοκίνητο. Δυστυχώς όμως δεν κάνει κάποια ιδιαίτερη κίνηση αφού κάθετα σχεδόν ακίνητος οπότε μια καλή προσέγγιση γίνεται μέσω της εύρεσης της ταχύτητας της συσκευής που είναι ίδια με αυτήν του χρήστη. Έτσι όταν ξεπερνάει ο χρήστης ένα όριο ταχύτητας θεωρούμε πως βρίσκεται σε αυτοκίνητο, την οποία ταχύτητα της συσκευής υπολογίζουμε με τους δύο τρόπους που προαναφέραμε στην χρήση πολλών αισθητήρων χρησιμοποιώντας το GPS όταν το αυτοκίνητο βρίσκεται εν κινήσει και το επιταχυνσιόμετρο όταν σταματάει (μπορεί να είναι και ταξί λεωφορείο ή κάποια γραμμή μετρό όπου πιάνει σήμα GPS από τους δορυφόρους). Τέλος όταν η ταχύτητα ξεπερνάει τα 15 km/h θεωρούμε πως ο χρήστης βρίσκεται στο αυτοκίνητο εν κινήσει και κάθε φορά που κάνει στάση και η ταχύτητα πέφτει κάτω από 3 km/h αποθηκεύουμε προσωρινά τις μεταβλητές θέσης (γεωγραφικό μήκος, πλάτος) μέχρι να ξεκινήσει ξανά να κινείται (για παράδειγμα επειδή μπορεί να σταματάει για φανάρι) και όταν ο χρήστης περπατήσει (άρα βρίσκεται έξω από το αυτοκίνητο) αυτές οι προσωρινά αποθηκευμένες μεταβλητές στέλνονται στον απομακρυσμένο εξυπηρετητή ιστού.

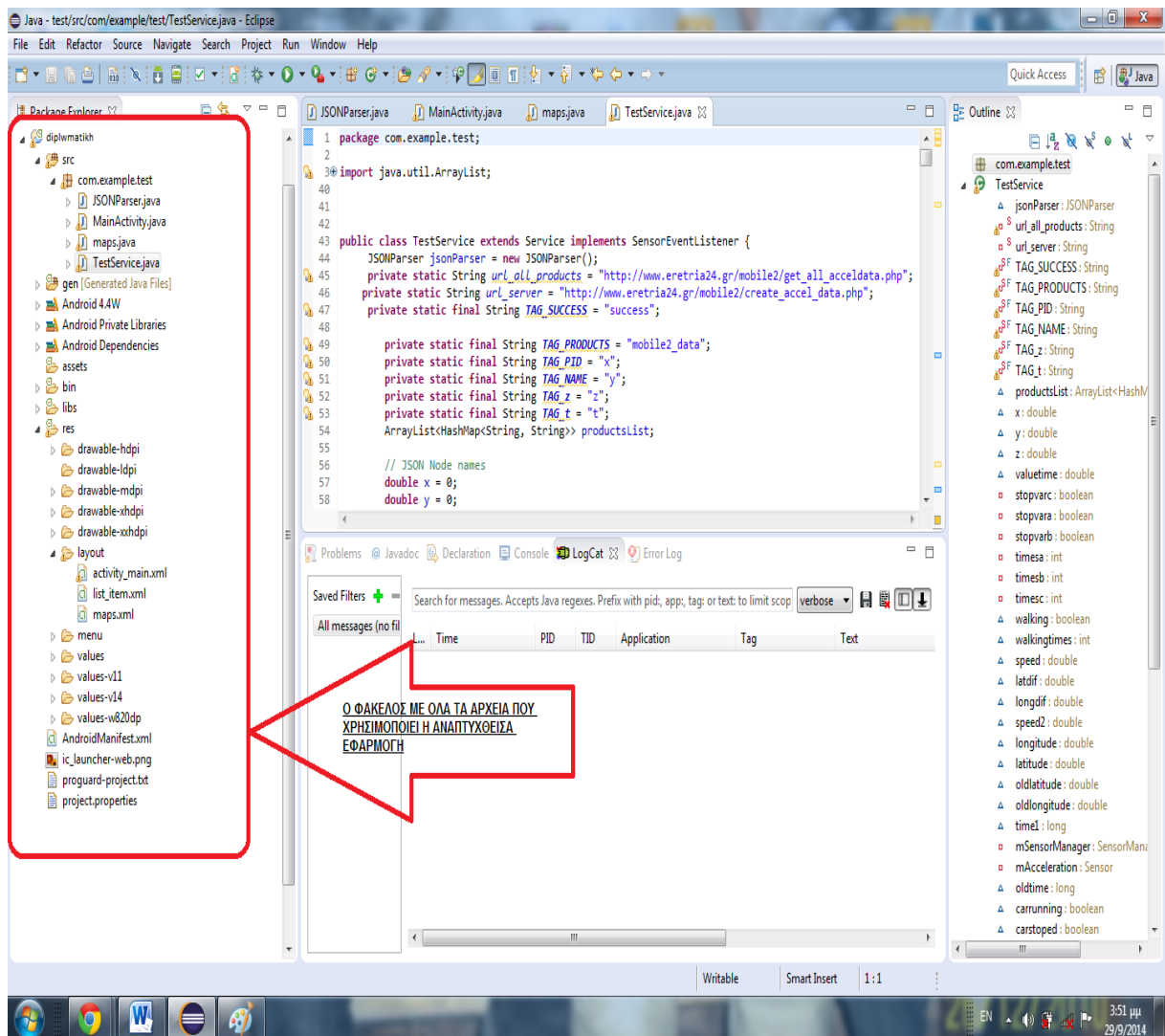
Για μικρές αποστάσεις που μπορεί να γίνουν με πολύ μικρή ταχύτητα (κάτω από 15km/h) δεν γίνεται κάποια αναγνώριση στην εφαρμογή διότι κρίθηκε πολύ σπάνιο φαινόμενο και θα είχε λιγότερα ποσοστά επιτυχίας (θα μπορούσε να είναι και κάποιος που τρέχει) αλλά σίγουρα είναι εφικτό να γίνει και θα μπορούσε να καταλαβαίνει άμα ο χρήστης τρέχει με το επιταχυνσιόμετρο (από τις δονήσεις της συσκευής) και να μην το υπολογίζει σαν να βρίσκεται σε αυτοκίνητο επειδή ξεπέρασε μία ορισμένη ταχύτητα.

# 5

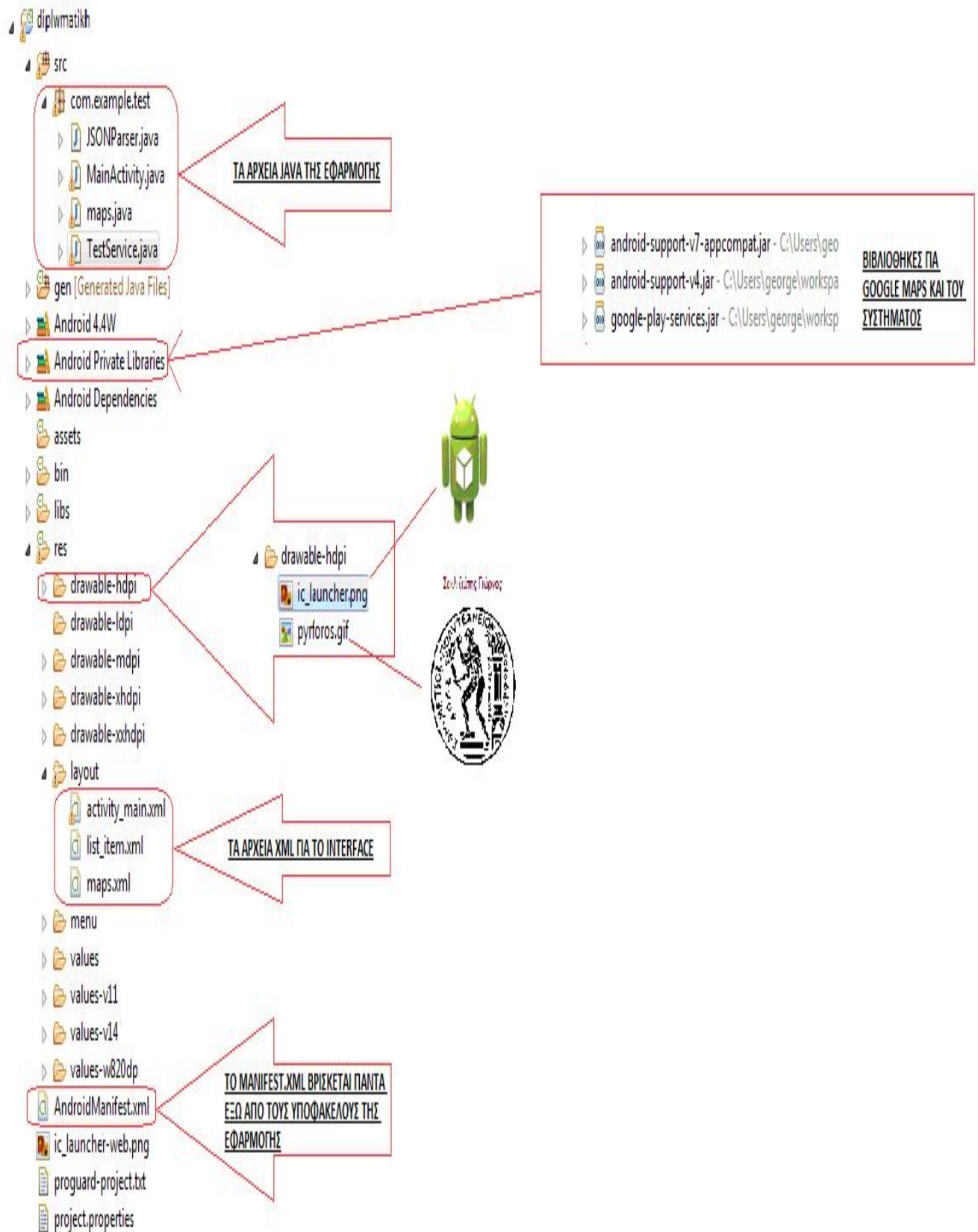
## Επιμέρους παρουσίαση τεχνικών λύσεων που χρησιμοποιήθηκαν

Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκε το ολοκληρωμένο περιβάλλον διαχείρισης eclipse. Τα βασικά αρχεία της εφαρμογής για την επικοινωνία της με τον απομακρυσμένο εξυπηρετητή, για τον ορισμό των προδιαγραφών της και για τις διάφορες οθόνες που θα εμφανίζονται θα αναλυθούν στην συνέχεια του κεφαλαίου.

Τα αρχεία της εφαρμογής όπως φαίνονται μέσα από το eclipse:



Εικόνα 18 : ο φάκελος της εφαρμογής μέσα από το περιβάλλον του eclipse



Εικόνα 19 : Ανάλυση των αρχείων που περιέχονται στον φάκελο της εφαρμογής

## 5.1 Δήλωση των συστατικών στοιχείων της εφαρμογής

Το αρχείο manifest της εφαρμογής (manifest.xml) όπου δηλώνονται τα συστατικά στοιχεία της εφαρμογής και οι απαραίτητες άδειες που πρέπει να έχει από το λειτουργικό σύστημα.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.diplwmatikh"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk
        android:minSdkVersion="12"
        android:targetSdkVersion="21" />
    <permission
        android:name="com.example.googlemaps.permission.MAPS_RECEIVE"
        android:protectionLevel="signature" />
    <uses-feature
        android:glEsVersion="0x00020000"
        android:required="true" />
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <service android:name=".TestService"
            android:enabled="true"/>
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name="com.example.test.maps"
            android:label="goglemap" >
        </activity>
        <meta-data
            android:name="com.google.android.maps.v2.API_KEY"
            android:value="AIzaSyC2XiX1Mt8K59LQ09wVKujE4zYunj5cZgU" />
        <meta-data
            android:name="com.google.android.gms.version"
            android:value="@integer/google_play_services_version" />
    </application>
    <uses-permission
        android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="com.example.test.permission.MAPS_RECEIVE" />
    <uses-permission
        android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
</manifest>
```

## 5.2 Περιβάλλον διεπαφής χρήστη (user interface) και αρχεία xml

### 5.2.1 Αρχεία xml

Η διεπαφή (user interface) της κεντρικής οθόνης (το αρχείο Main.xml )

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/pyrforos"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.test.MainActivity" >
```

Κουμπί για την εκκίνηση της εφαρμογής και της υπηρεσίας που στέλνει τα δεδομένα στον server:

```
<Button
    android:id="@+id/btnStart"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textView1"
    android:layout_below="@+id/textView1"
    android:layout_marginLeft="35dp"
    android:layout_marginTop="56dp"
    android:text="ΠΙΝΑΚΑΣ" />
```

Κουμπί για την έξοδο από την εφαρμογή αλλά συνέχεια της υπηρεσίας:

```
<Button
    android:id="@+id/btnStop"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/btnStart"
    android:layout_alignBottom="@+id/btnStart"
    android:layout_alignParentRight="true"
    android:layout_marginRight="30dp"
    android:text="ΕΞΟΔΟΣ" />
```

Κουμπί για την εμφάνιση σημείου στάθμευσης σε googlemaps:

```
<Button
    android:id="@+id/btnmaps"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignRight="@+id/btnStop"
    android:layout_marginBottom="42dp"
    android:text="ΧΑΡΤΗΣ" />
```

Εμφάνιση των δεδομένων του server που θέλουμε να εμφανίζονται σε μορφή λίστας (για επαλήθευση):



```
<ListView
  android:id="@android:id/list"
  android:layout_width="250dp"
  android:layout_height="180dp"
  android:layout_alignLeft="@+id/btnStart"
  android:layout_below="@+id/btnStart"
  android:layout_marginTop="32dp" >
</ListView>
```

Κουμπί για τον τερματισμό της εφαρμογής και της υπηρεσίας που στέλνει στον server:

```
<Button
  android:id="@+id/button1"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_alignLeft="@+id/btnmaps"
  android:layout_alignTop="@android:id/list"
  android:layout_marginTop="60dp"
  android:text="ΤΕΡΜΑΤΙΣΜΟΣ" />
```

Κουμπί για την έναρξη της υπηρεσίας που στέλνει στον server:

```
<Button
  android:id="@+id/button2"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_alignBaseline="@+id/btnStart"
  android:layout_alignBottom="@+id/btnStart"
  android:layout_alignLeft="@+id/button1"
  android:text="ΕΚΚΙΝΗΣΗ" />
```

```
</RelativeLayout>
```

Η λίστα εμφάνισης των συντεταγμένων που αποθηκεύονται στον server έτσι όπως θέλουμε να φαίνονται στην κεντρική οθόνη της εφαρμογής από την οποία καλούνται (List\_item.xml).

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical" >
    <TextView
        android:id="@+id/name"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:paddingLeft="6dip"
        android:paddingTop="6dip"
        android:textSize="17dip"
        android:textStyle="bold"
        android:textColor="#FFA500" />
    <TextView
        android:id="@+id/z"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:paddingTop="6dip"
        android:paddingLeft="6dip"
        android:textSize="17dip"
        android:textStyle="bold" />
    <TextView
        android:id="@+id/t"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:paddingTop="6dip"
        android:paddingLeft="6dip"
        android:textSize="17dip"
        android:textStyle="bold" />
</LinearLayout>

```

Ο χάρτης googlemaps που καλούμε μέσω της κεντρικής οθόνης (main\_activity) από τις βιβλιοθήκες του google play services (το αρχείο Maps.xml της εφαρμογής).

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

    <fragment
        android:id="@+id/map"
        android:name="com.google.android.gms.maps.MapFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

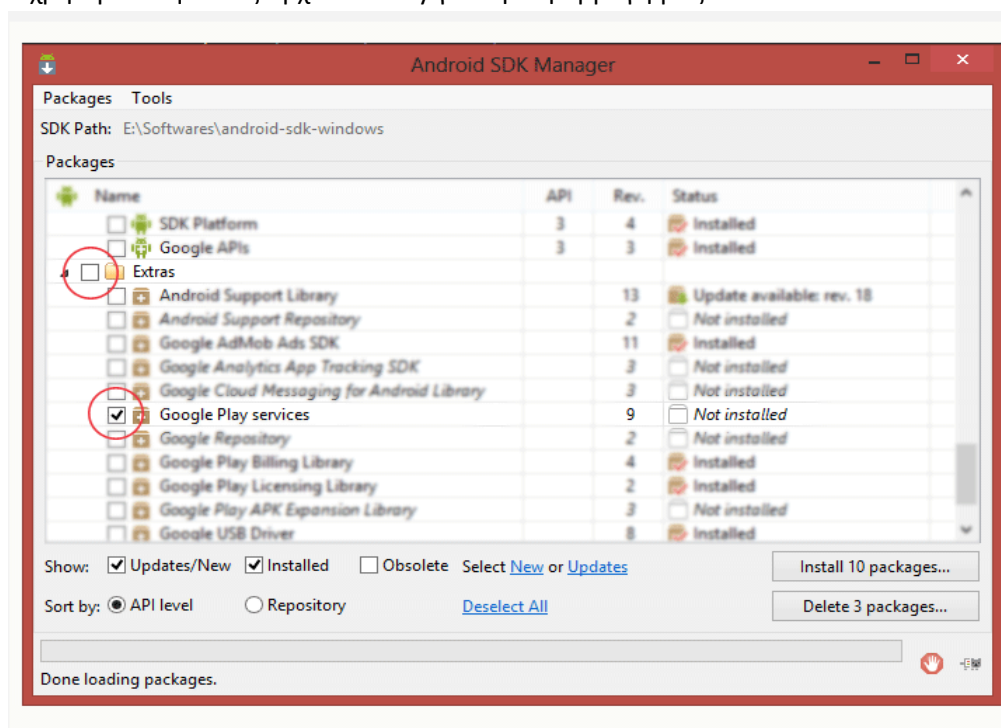
</RelativeLayout>

```

## 5.2.2 Εισαγωγή των GOOGLE MAPS APIs (application programming interface)

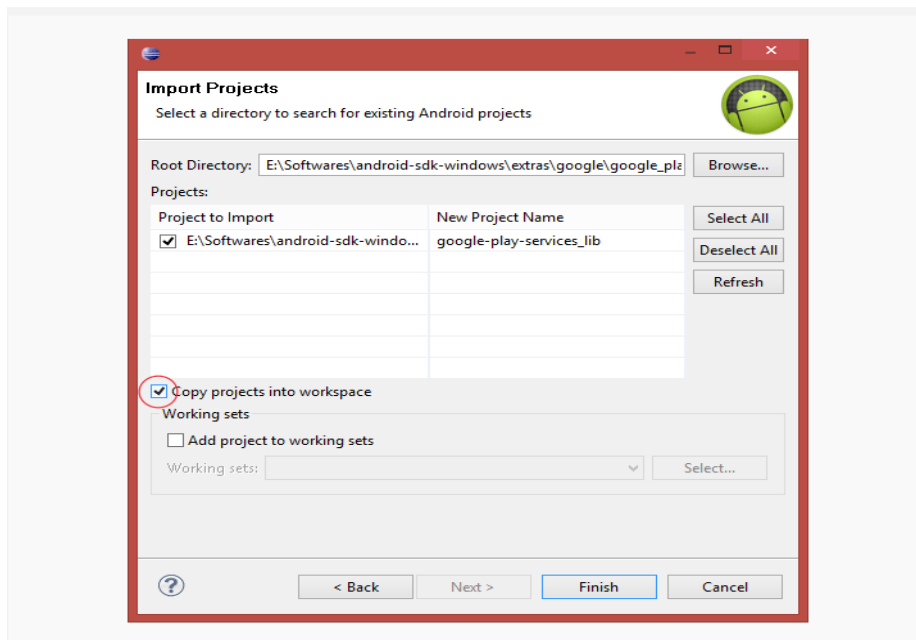
Θεωρήθηκε σωστό να προβάλλεται το σημείο στάθμευσης σε google maps το οποίο σημαίνει ότι έπρεπε να γίνει χρήση του google maps android API version2 διότι η πρώτη έκδοση καταργήθηκε και πλέον χρησιμοποιείται αυτή ( version 2). Η διαδικασία είναι η εξής:

1. Από το SDK manager του eclipse (αφού έχει ενσωματωμένα τα SDK tools) έπρεπε να κατέβει και να εγκατασταθεί η τελευταία έκδοση των Google play services διότι θα χρησιμοποιηθεί ως αρχείο library για την εφαρμογή μας.



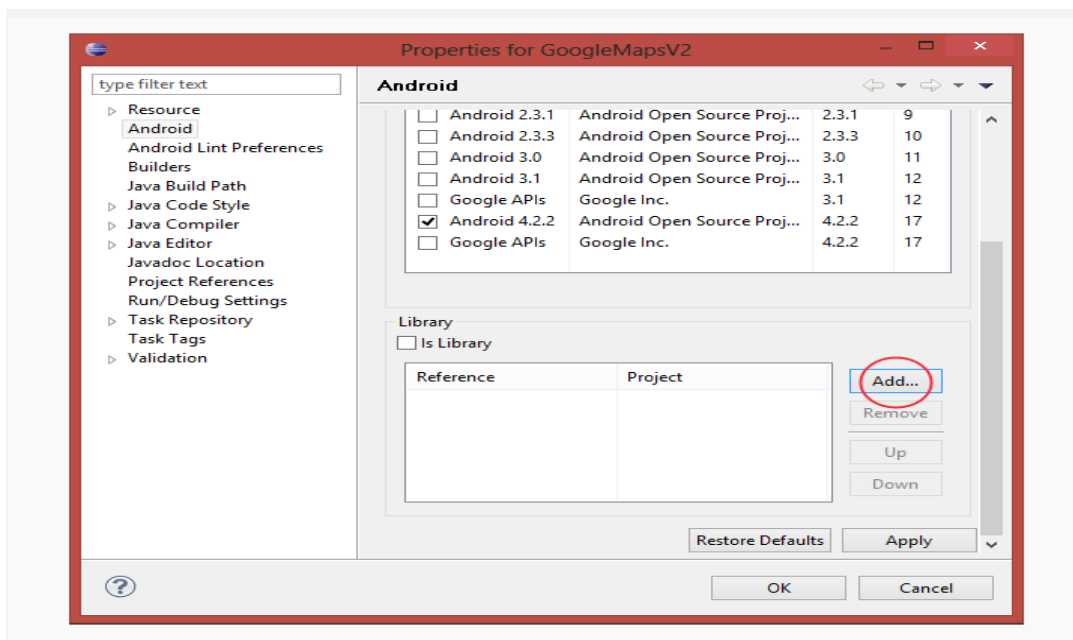
**Εικόνα 20 : Λήψη google play services**

2. Στην συνέχεια τα Google play services έπρεπε να αντιγραφούν στο workspace του eclipse και να εισαχθούν (import) στο IDE (integrated development environment) του eclipse για να αναγνωρίζονται ως ξεχωριστό πρόγραμμα (project).



**Εικόνα 21 : Εισαγωγή google play services στο workspace του eclipse**

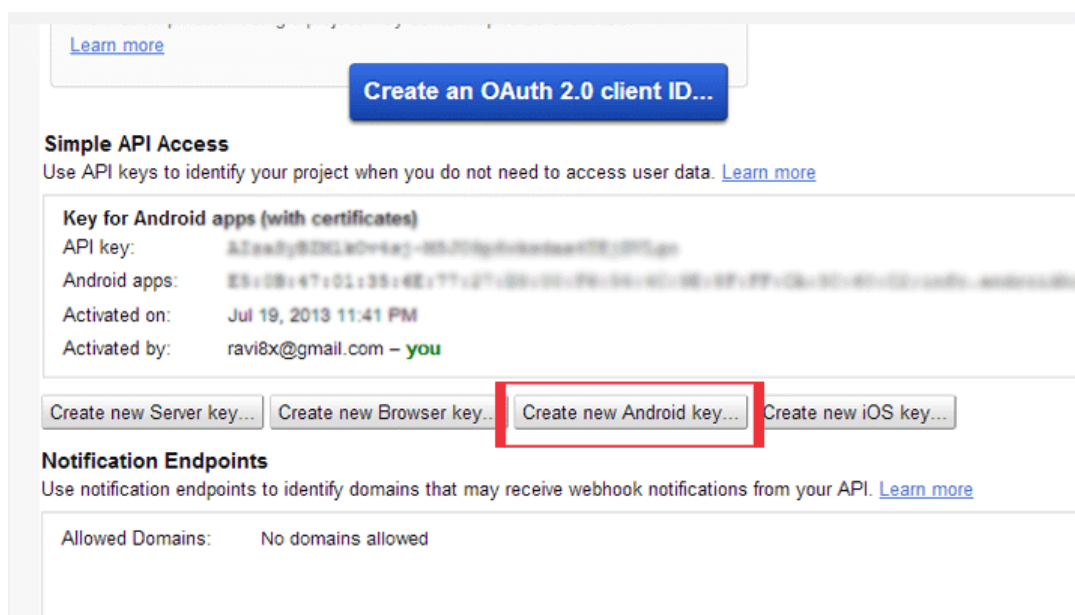
3. Τα Google play services έπρεπε να εισαχθούν στο δικό μας πρόγραμμα ως βιβλιοθήκη.



**Εικόνα 22 : Προσθήκη googleplay services ως βιβλιοθήκη**

4. Για να γίνει χρήση του google maps api έπρεπε να γίνει δεκτό ένα μοναδικό Key της εφαρμογής μας, το SHA-1 fingerprint, από το Google API Console (<https://code.google.com/apis/console>) για να παραχθεί ένα νέο android key από το Google API Console , το οποίο θα χρησιμοποιηθεί στην συνέχεια ως κωδικός χρήσης

του google maps από Google play services library ο οποίος θα οριστεί στο Manifest.xml της εφαρμογής μας.



**Εικόνα 23 : Δήλωση Sha-1 fingerprint στο google api console**

5. Το μόνο που μένει για την εμφάνιση του χάρτη είναι η διαδικασία συμπλήρωσης των αδειών χρήσης (uses permission) και του google android maps v2 api key στο αρχείο manifest.xml καθώς και η δημιουργία ενός σχεδίου (layout) όπου θα καλεί τον χάρτη (map fragment) από τις υπάρχουσες βιβλιοθήκες του google play και θα καλείται με το πάτημα ενός κουμπιού της κεντρικής διαδικασίας.
6. Στο τέλος έγινε η προσθήκη ενός σημαδιού (marker) πάνω στον χάρτη για το που στάθμευσε τελευταία φορά ο χρήστης του κινητού λαμβάνοντας από την βάση διαδικτυακά το γεωγραφικό μήκος και πλάτος και τοποθετώντας ένα σημάδι στο αντίστοιχο σημείο του χάρτη με κατάλληλη ενημέρωση.



**Εικόνα 24 : Το στίγμα της τελευταίας θέσης στάθμευσης του αυτοκινήτου στο googlemaps.**

## 5.3 Ο διερμηνέας JSONParser.java που χρησιμοποιήθηκε καθώς και τα PHP αρχεία με τα οποία επικοινωνεί

### 5.3.1 JSON PARSER

Όπως αναφέραμε χρησιμοποιήσαμε μία κλάση (αρχείο java) για να καταλαβαίνει ο server τα αρχεία JavaScript Object Notation που στέλνουμε ( JSONParser.java ) δηλαδή έναν διερμηνέα :

```
JSONParser
//Δήλωση της εφαρμογής στην οποία αναφέρεται
package com.example.diplwmatikh;
//Δηλώσεις των στοιχείων android τα οποία χρησιμοποιεί και πρέπει να
//εισαχθούν
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.util.List;
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.client.utils.URLEncodedUtils;
import org.apache.http.impl.client.DefaultHttpClient;
import org.json.JSONException;
import org.json.JSONObject;
import android.util.Log;
//Ορισμός της κλάσης
public class JSONParser {
    static InputStream is = null;
    static JSONObject jsonObj = null;
    static String json = "";
    public JSONParser() {
    }
//Η διαδικασία μετατροπής των δεδομένων από JSON σε URL και αντιστρόφως
//μέσω της μεθόδου GET ή POST της HTTP
    public JSONObject makeHttpRequest(String url, String method,
        List<NameValuePair> params) {
    Δημιουργεία λήψης της μεθόδου που απαιτείται
    try {
    Έλεγχος αν είναι POST ή GET η επιθυμητή μέθοδος και κάλεσμα των απαραίτητων
    PHP αρχείων που είναι αποθηκευμένα στον server
        if(method == "POST"){
            Αν είναι POST έχει το ανάλογο "HTTP request"
            DefaultHttpClient httpClient = new DefaultHttpClient();
            HttpPost httpPost = new HttpPost(url);
            httpPost.setEntity(new UrlEncodedFormEntity(params));
            HttpResponse httpResponse = httpClient.execute(httpPost);
            HttpEntity httpEntity = httpResponse.getEntity();
            is = httpEntity.getContent();
        }
    }
}
```

```

    }else if(method == "GET"){
        Av είναι GET έχει το ανάλογο "HTTP request"
        DefaultHttpClient httpClient = new DefaultHttpClient();
        String paramString = URLEncodedUtils.format(params, "utf-
8");

        url += "?" + paramString;
        HttpGet httpGet = new HttpGet(url);
        HttpResponse httpResponse = httpClient.execute(httpGet);
        HttpEntity httpEntity = httpResponse.getEntity();
        is = httpEntity.getContent();
    }
} catch (UnsupportedEncodingException e) {
    e.printStackTrace();
} catch (ClientProtocolException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
}
try {
    BufferedReader reader = new BufferedReader(new
InputStreamReader(
        is, "iso-8859-1"), 8);
    StringBuilder sb = new StringBuilder();
    String line = null;
    while ((line = reader.readLine()) != null) {
        sb.append(line + "\n");
    }
    is.close();
    json = sb.toString();
} catch (Exception e) {
    Log.e("Buffer Error", "Error converting result " +
e.toString());
}
try {
    jsonObj = new JSONObject(json);
} catch (JSONException e) {
    Log.e("JSON Parser", "Error parsing data " + e.toString());
}
return jsonObj;
}
}
}

```

## 5.3.2 PHP

- Το αρχείο `db_config_accel.php` από το οποίο παίρνει ο `JSONParser` τις πληροφορίες πρόσβασης στον εξυπηρετητή (`server`).

```
<?php
/*
 * οι μεταβλητές για την σύνδεση με την βάση
 */
define('DB_USER', "..."); // χρήστης
define('DB_PASSWORD', "..."); // κωδικός βάσης
define('DB_DATABASE', "eretriag_android2"); // όνομα βάσης
define('DB_SERVER', "localhost"); // ο server
?>
```

- Το αρχείο `db_connect_accel.php` το οποίο καλεί ο `JSONParser` για σύνδεση με την βάση.

```
<?php
/**
 * ένα αρχείο – κλάση για σύνδεση με την βάση
 */
class DB_CONNECT {

    // ορισμός συνάρτησης σύνδεσης
    function __construct() {
        // σύνδεση με την βάση
        $this->connect();
    }

    // ορισμός συνάρτησης αποσύνδεσης
    function __destruct() {
        // αποσύνδεση από την βάση
    }
}
```



```

        $this->close();
    }
    /**
     * κατασκευή της συνάρτησης σύνδεσης
     */
    function connect() {
        // εισαγωγή μεταβλητών σύνδεσης με την βάση
        require_once __DIR__ . '/db_config_accel.php';
        // σύνδεση
        $con = mysql_connect(DB_SERVER, DB_USER, DB_PASSWORD) or die(mysql_error());
        // επιλογή βάσης
        $db = mysql_select_db(DB_DATABASE) or die(mysql_error()) or die(mysql_error());
        // επιστροφή σημείου σύνδεσης
        return $con;
    }
    /**
     * κατασκευή συνάρτησης για αποσύνδεση
     */
    function close() {
        // αποσύνδεση
        mysql_close();
    }
}
?>

```

- Το αρχείο `create_accel_data.php` το οποίο καλείται για εγγραφή δεδομένων στον πίνακα της βάσης μας.

```
<?php
/*
 * δημιουργία νέας εγγραφής
 * Οι πληροφορίες της εγγραφής έρχονται μέσω HTTP POST request
 */
// πίνακας για ανταπόκριση με JSONParser της εφαρμογής
$response = array();
// έλεγχος για τα απαραίτητα πεδία
if (isset($_POST['x']) && isset($_POST['y']) && isset($_POST['z']) && isset($_POST['t'])) {
    $x = $_POST['x'];
    $y = $_POST['y'];
    $z = $_POST['z'];
    $t = $_POST['t'];
    // εισαγωγή της κλάσης - αρχείου σύνδεσης
    require_once __DIR__ . '/db_connect_accel.php';
    // σύνδεση με την βάση
    $db = new DB_CONNECT();
    // εισαγωγή σειράς σε mysql
    $result = mysql_query("INSERT INTO mobile2_data(x, y, z, t) VALUES('$x', '$y', '$z', '$t')");
    // έλεγχος για επιτυχή εισαγωγή της σειράς
    if ($result) {
        // επιτυχής εισαγωγή σειράς
        $response["success"] = 1;
    } else {
        // αποτυχία εισαγωγής
        $response["success"] = 0;
    }
}
```

```

    }
} else {
    //όταν λείπει κάποιο απαραίτητο πεδίο από την σειρά
    $response["success"] = 0;
}
?>

```

➤ **Το αρχείο `get_all_acceldata.php` που καλείται για την λήψη των τιμών του πίνακα της βάσης.**

```

<?php
/*
 * Ο παρακάτω κώδικας εμφανίζει μια λίστα όλων των εγγραφών που υπάρχουν στη βάση
 */
// πίνακας για ανταπόκριση στοιχείων JSON
$response = array();
// περιλαμβάνει την κλάση για την σύνδεση με την βάση
require_once __DIR__ . '/db_connect_accel.php';
// σύνδεση με την βάση
$db = new DB_CONNECT();
// επιλογή όλων των εγγραφών από τον πίνακα Mobile2_data
$result = mysql_query("SELECT * FROM mobile2_data order by timestamp asc") or
die(mysql_error());
// έλεγχος για αποτέλεσμα
if (mysql_num_rows($result) > 0) {
    $response["mobile2_data"] = array();
    while ($row = mysql_fetch_array($result)) {
        $product = array();
        $product["timestamp"] = $row["timestamp"];
        $product["x"] = $row["x"];
    }
}

```

```
$product["y"] = $row["y"];
$product["z"] = $row["z"];
$product["t"] = $row["t"];
    array_push($response["mobile2_data"], $product);
}
// επιτυχές
$response["success"] = 1;
echo json_encode($response);
} else {
    // αλλιώς δεν βρέθηκαν αποτελέσματα.
    $response["success"] = 0;
    $response["message"] = "Δεν βρέθηκαν δεδομένα";
    // echo no users JSON
    echo json_encode($response);
}
?>
```

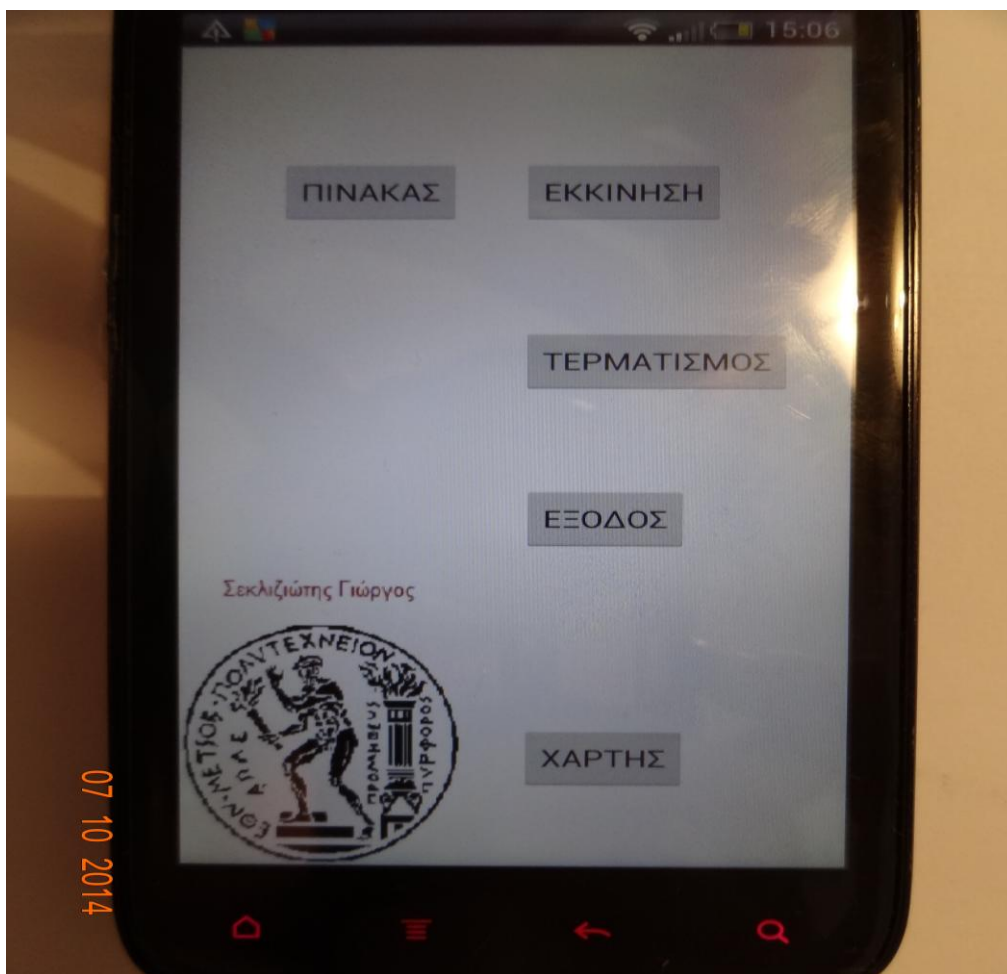
# 6

## Παρουσίαση και επικύρωση της αναπτυχθείσας εφαρμογής

Σε αυτό το κεφάλαιο θα γίνει παρουσίαση των διάφορων οθονών της εφαρμογής καθώς και επεξήγηση των κουμπιών της. Στην συνέχεια θα γίνει αναφορά στην χρήση της εφαρμογής που έγινε κατά την ολοκλήρωση της για την εξαγωγή συμπερασμάτων ως προς την αποτελεσματικότητά της.

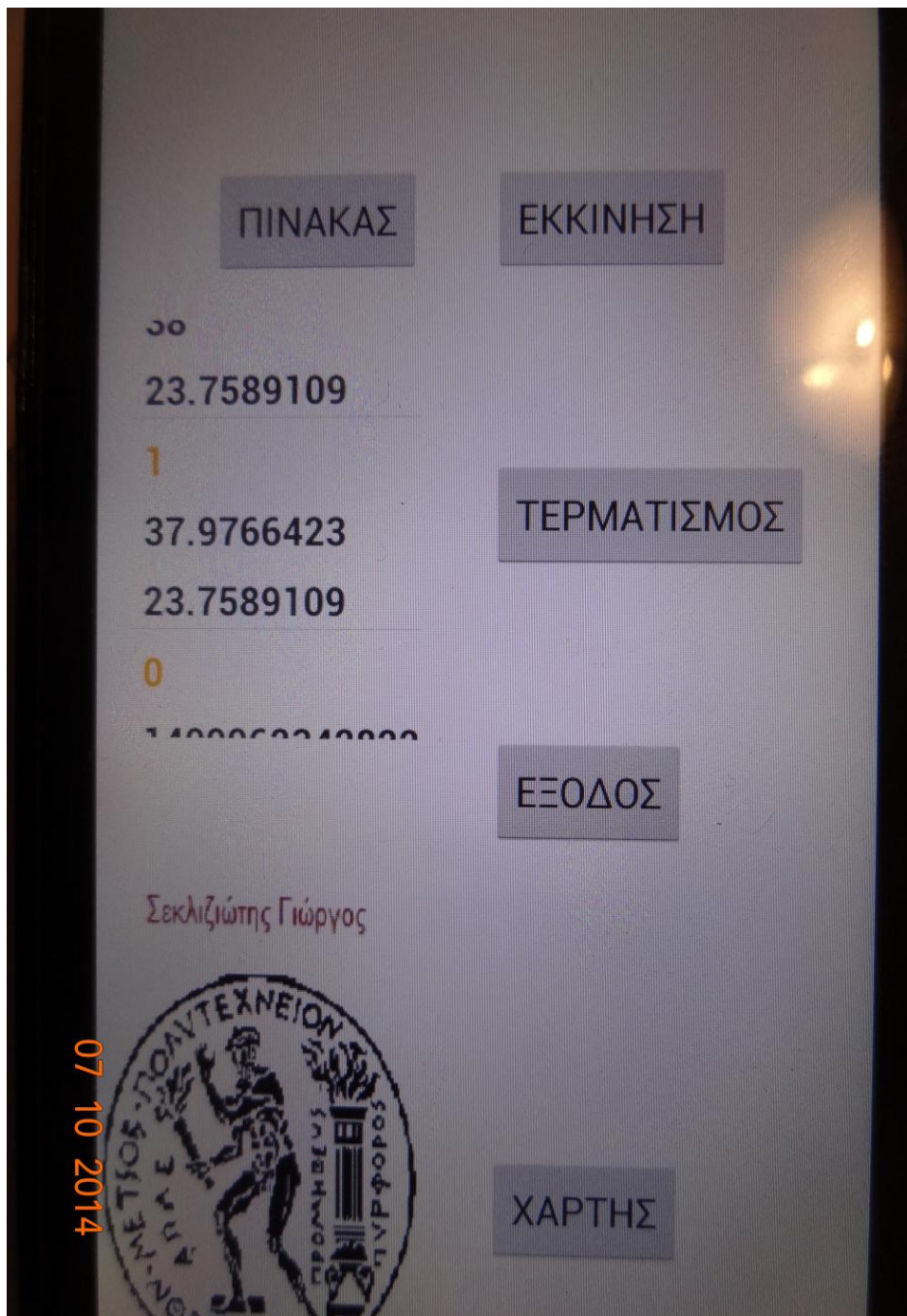
### 6.1 Παρουσίαση εφαρμογής

Η αναπτυχθείσα εφαρμογή παρουσιάζει στο κινητό μία οθόνη αποτελούμενη από πέντε κουμπιά και το σήμα του πολυτεχνείου με το όνομα μου. Τα κουμπιά είναι: ΠΙΝΑΚΑΣ, ΕΚΚΙΝΗΣΗ, ΤΕΡΜΑΤΙΣΜΟΣ, ΕΞΟΔΟΣ, ΧΑΡΤΗΣ.



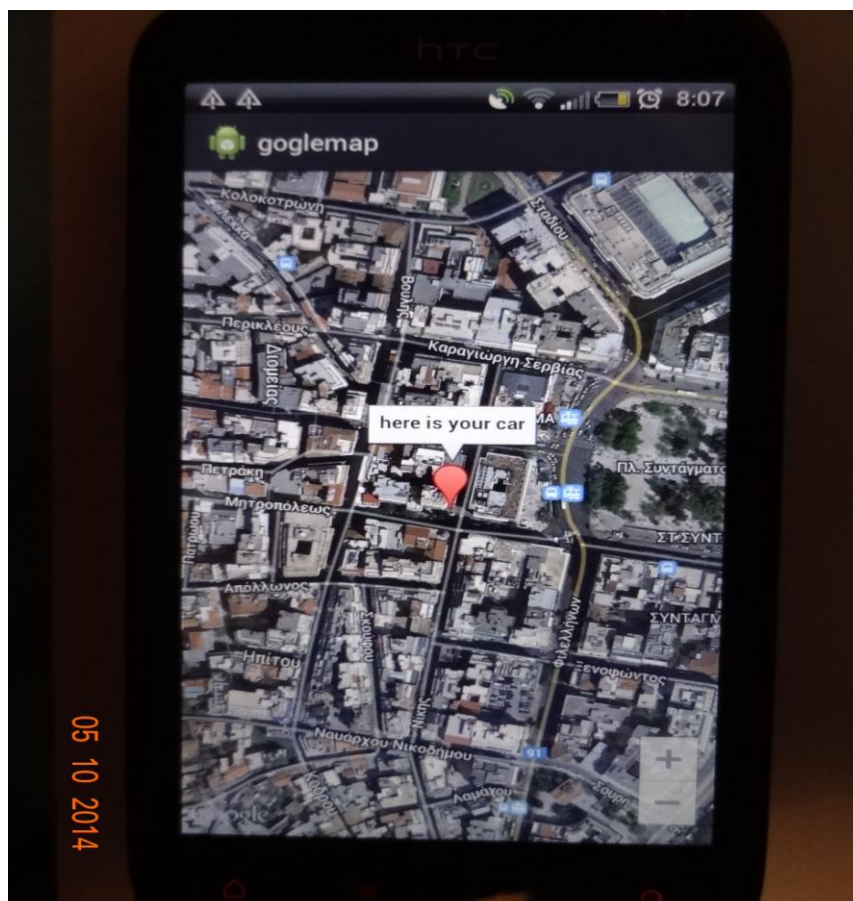
Εικόνα 25 : Η οθόνη της εφαρμογής στο κινητό

1. Πατώντας το κουμπί “πίνακας” αν υπάρχει σύνδεση στο διαδίκτυο εμφανίζεται από τον πίνακα της βάσης μία στήλη με μπάρα κύλισης που περιλαμβάνει σε λίστα τις τιμές γεωγραφικού μήκους και πλάτους και με πορτοκαλί χρώμα την μεταβλητή 0 ή 1 ανάλογα με το αν ο χρήστης έχει σταθμεύσει σε εκείνες τις θέσεις. Αν δεν υπάρχει σύνδεση στο διαδίκτυο παρουσιάζεται ανάλογο μήνυμα.



Εικόνα 26 : Εκτέλεση της εφαρμογής

2. Με το κουμπί “εκκίνηση” ενεργοποιείται μια υπηρεσία (service) η οποία τρέχει μόνιμα στο υπόβαθρο (background ) και δεν σταματάει κατά την έξοδο από την εφαρμογή. Η υπηρεσία όποτε έχει πρόσβαση στο διαδίκτυο στέλνει κάθε μερικά δευτερόλεπτα την τοποθεσία της συσκευής μαζί με μία μεταβλητή με τιμή 0. Επίσης συνεχώς ελέγχει την είσοδο σε αυτοκίνητο και εφόσον υπάρξει είσοδος σε αυτοκίνητο και κίνηση του, μετά από ανάπτυξη ταχύτητας ελέγχει αν σταματήσει και κατέβει ο χρήστης. Μόλις ο χρήστης του έξυπνου κινητού τηλεφώνου κατέβει και περπατήσει , η εφαρμογή στέλνει την τοποθεσία στάθμευσης μαζί με μία μεταβλητή (οπού στην περίπτωση στάθμευσης είναι ένα) για να μπορεί διαχωρίζει αυτός που κάνει ανάλυση των δεδομένων της βάσης τις συντεταγμένες κίνησης από αυτές στάθμευσης και να μπορεί να ερμηνεύει καλύτερα την συμπεριφορά του χρήστη της συσκευής.
3. Με το κουμπί “τερματισμός” τερματίζεται η υπηρεσία (service) που στέλνει τα δεδομένα.
4. Με το κουμπί “έξοδος” βγαίνουμε από την εφαρμογή χωρίς όμως να σταματάει την υπηρεσία.
5. Με το κουμπί “χάρτης” εμφανίζεται η τελευταία τοποθεσία στάθμευσης σε χάρτη πόλης με ένα σημάδι και ανάλογο μήνυμα. Σε περίπτωση που δεν υπάρχει διαδικτυακή σύνδεση πάλι υπάρχει ανάλογο μήνυμα ενημέρωσης.



**Εικόνα 27 : Η οθόνη της επιλογής ΧΑΡΤΗΣ**

Ο πίνακας στον οποίο στέλνονται τα δεδομένα φαίνεται από κάτω μέσα από το διαχειριστικό περιβάλλον cpanel και το εργαλείο phpMyAdmin:

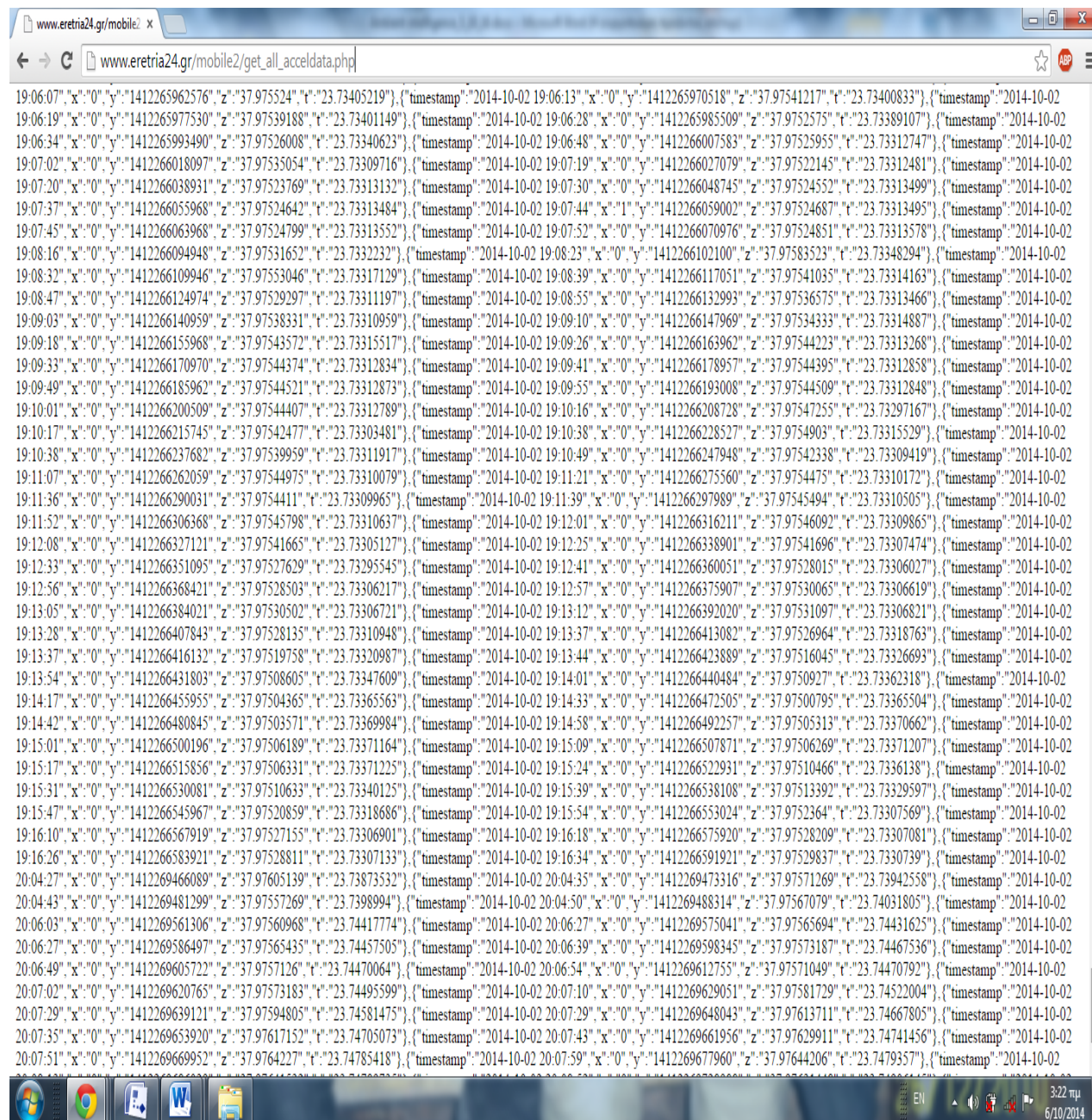
The screenshot shows the phpMyAdmin interface for the 'mobile2\_data' table. A green message at the top indicates that 25 records were displayed out of 1125 total records. The table below contains the following data:

|                          | pid | timestamp           | x | y             | z           | t           |
|--------------------------|-----|---------------------|---|---------------|-------------|-------------|
| <input type="checkbox"/> | 26  | 2014-09-06 03:14:44 | 0 | 1409962479254 | 37.9763267  | 23.75887525 |
| <input type="checkbox"/> | 27  | 2014-09-06 03:14:53 | 0 | 1409962484299 | 37.97632308 | 23.75899638 |
| <input type="checkbox"/> | 28  | 2014-09-06 03:15:13 | 0 | 1409962506273 | 37.97663251 | 23.75907928 |
| <input type="checkbox"/> | 29  | 2014-09-06 03:15:21 | 0 | 1409962515272 | 37.97718043 | 23.75934046 |
| <input type="checkbox"/> | 30  | 2014-09-06 03:15:38 | 0 | 1409962521287 | 37.97714228 | 23.75943803 |
| <input type="checkbox"/> | 31  | 2014-09-06 03:15:41 | 0 | 1409962537263 | 37.97619677 | 23.75929256 |
| <input type="checkbox"/> | 32  | 2014-09-06 03:15:42 | 0 | 1409962539330 | 37.97618963 | 23.75926809 |
| <input type="checkbox"/> | 33  | 2014-09-06 03:15:44 | 0 | 1409962540258 | 37.97617885 | 23.75926318 |
| <input type="checkbox"/> | 34  | 2014-09-06 03:15:45 | 0 | 1409962542277 | 37.9761725  | 23.7593168  |
| <input type="checkbox"/> | 35  | 2014-09-06 03:15:47 | 0 | 1409962543277 | 37.97616868 | 23.75931927 |
| <input type="checkbox"/> | 36  | 2014-09-06 03:15:53 | 0 | 1409962549272 | 37.97618351 | 23.75938853 |
| <input type="checkbox"/> | 37  | 2014-09-06 03:16:01 | 0 | 1409962556289 | 37.97619295 | 23.75941023 |
| <input type="checkbox"/> | 38  | 2014-09-06 03:16:08 | 0 | 1409962564281 | 37.97618492 | 23.75941706 |

**Εικόνα 28 : Η οθόνη του phpmyadmin στον απομακρυσμένο εξυπηρετητή ιστού**



Ο πίνακας όπως εμφανίζεται καλώντας το get\_all\_accel\_data.php αρχείο διαδικτυακά :

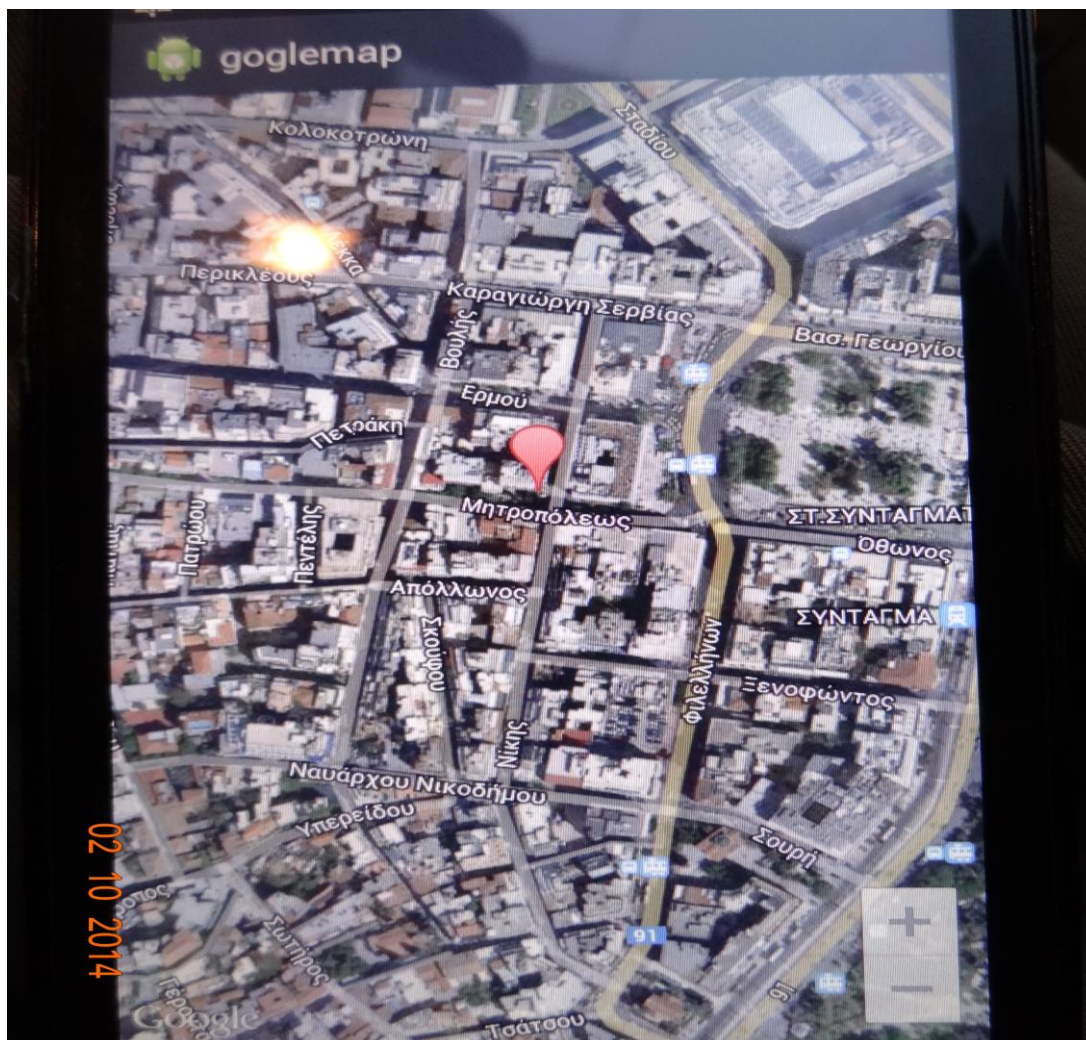


Εικόνα 29 : Τα δεδομένα του πίνακα mobile2\_data της βάσης android2

## 6.2 Επικύρωση εφαρμογής

Το σφάλμα των τιμών του πίνακα είναι σχεδόν πάντα κάτω από είκοσι μέτρα. Όσον αφορά τις τοποθεσίες στάθμευσης συγκεκριμένα το σφάλμα παρατηρήθηκε μέχρι περίπου 15 μέτρα μέσα σε πάνω από εκατό δοκιμές αλλά με μία συχνή απόκλιση από 0 έως 8 μέτρα. Εξετάστηκε πάνω πενήντα φορές και θα γίνει αναφορά σε τρεις αντιπροσωπευτικές.

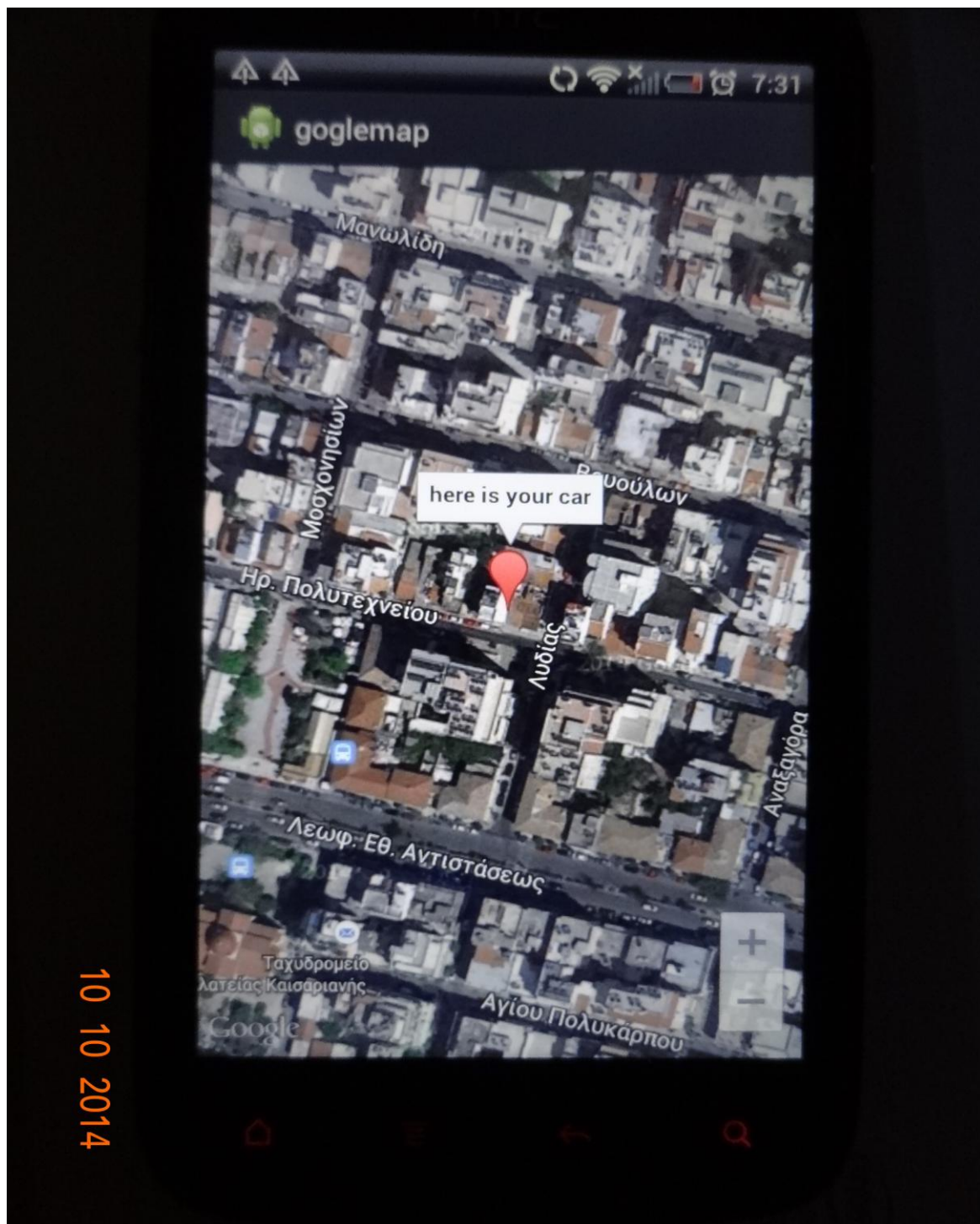
Για παράδειγμα στην περίπτωση που εμφανίζει στον χάρτη:



**Εικόνα 30 : Στίγμα θέσης στάθμευσης του αυτοκινήτου**

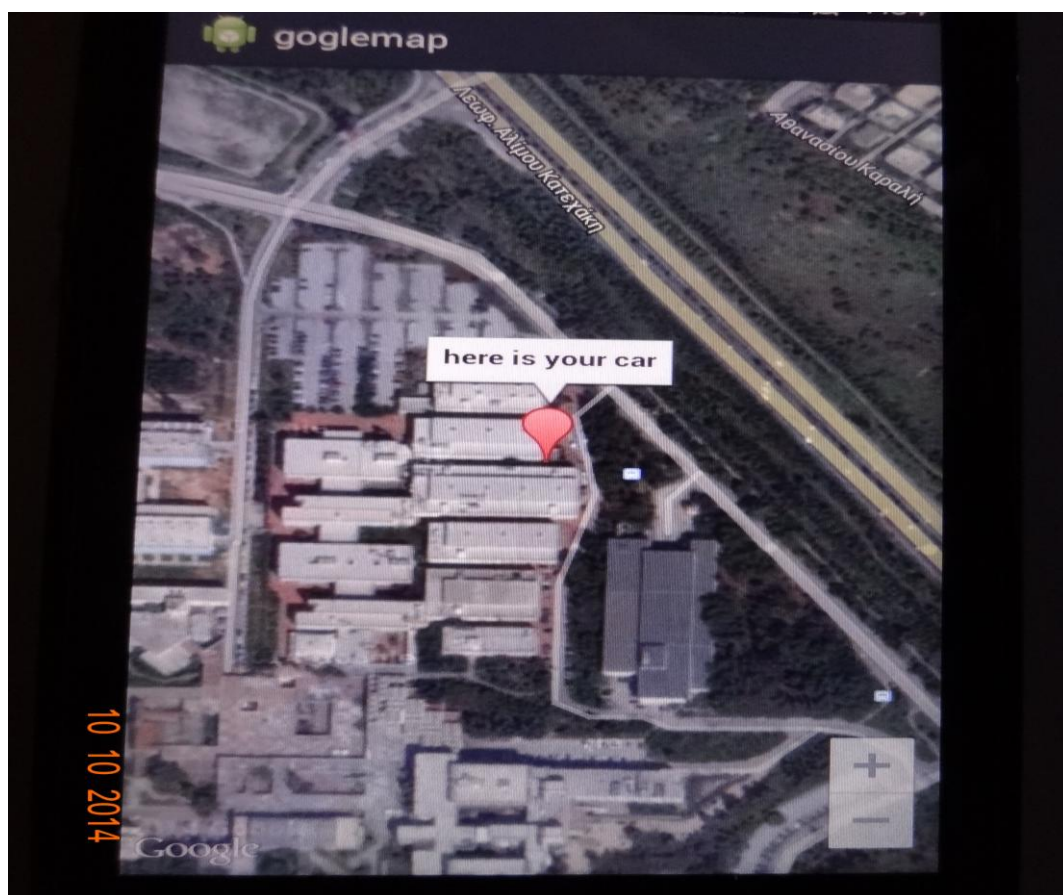
το αυτοκίνητο είναι γύρω στα πέντε μέτρα μακριά από εκεί που δείχνει ο χάρτης αφού είναι πάνω στην οδό Μητροπόλεως δίπλα στην γωνία με την οδό Νίκης

Στην παρακάτω περίπτωση το αυτοκίνητο βρίσκεται πάνω στην Ηρ. Πολυτεχνείου απέχοντας κάτω από πέντε μέτρα από το σημάδι



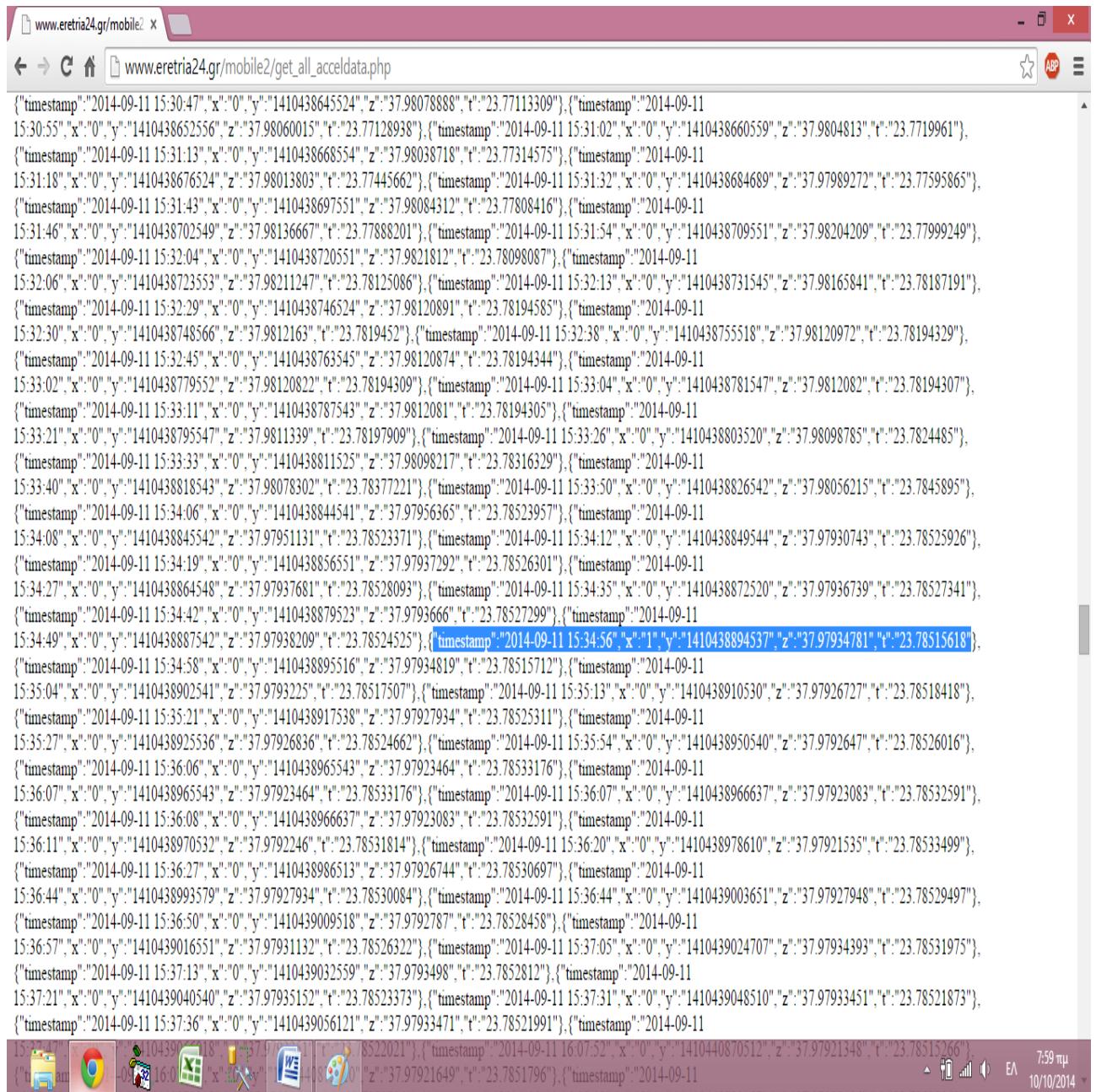
Εικόνα 31 : Στίγμα θέσης στάθμευσης του αυτοκινήτου

Ενώ στην περίπτωση που δείχνει το αυτοκίνητο πάνω στα κτήρια της σχολής μας το αυτοκίνητο απέχει περίπου δεκαπέντε μέτρα από το σημάδι.



**Εικόνα 32 : Στίγμα θέσης στάθμευσης του αυτοκινήτου**

Για καλύτερη κατανόηση των τιμών που έχει ο πίνακας στην βάση μας , παρατίθεται στην συνέχεια ακατέργαστη εικόνα του με επιλεγμένες τις τιμές για την τελευταία θέση στάθμευσης που είδαμε παραπάνω.



**Εικόνα 33 : Τα δεδομένα του πίνακα της βάσης μας**

`{\"timestamp\": \"2014-09-11 15:34:56\", \"x\": \"1\", \"y\": \"1410438894537\", \"z\": \"37.97934781\", \"t\": \"23.78515618\"}`

Όπου οι δύο τελευταίοι αριθμοί που αντιστοιχούν στις μεταβλητές  $z$  ,  $t$  είναι το γεωγραφικό μήκος και πλάτος αντίστοιχα, ενώ το  $x$  είναι ένα (1) επειδή είναι θέση στάθμευσης (θα ήταν 0 αν ήταν σημείο πορείας). Το  $y$  είναι μετρητής ώρας του κινητού σε millisecond που οποίο χρησιμοποιήθηκε για να παρατηρηθεί η διαφορά ώρας που κάνει για να σταθεί μια γραμμή στον πίνακα από την επόμενη. Το timestamp είναι η ώρα που καταχωρήθηκε επιτυχώς η γραμμή στον πίνακα της βάσης.

Επίσης παρατηρήθηκε πως για τα δεδομένα που στέλνονται στον πίνακα για την αναγνώριση της πορείας του χρήστη η χρονική διάρκεια μεταξύ δύο συνεχόμενων γραμμών συχνά έφταναν μέχρι και τα τριάντα δευτερόλεπτα ενώ ήταν προγραμματισμένο να γίνεται κάθε επτά δευτερόλεπτα. Αυτό οφείλεται στην αργή σύνδεση με το διαδίκτυο που είχε το συγκεκριμένο κινητό.

Τα σημεία πορείας που φαίνονται στον πίνακα παρατηρήθηκε πως είναι πολύ κοντά στα πραγματικά (με σφάλμα κάτω από τρία μέτρα) σε ποσοστό 95%.

Γενικώς το αποτέλεσμα ήταν ικανοποιητικό και με μία πιο σύγχρονη συσκευή android και μία αξιοπρεπή σύνδεση στο διαδίκτυο τα αποτελέσματα πολύ πιθανό να πλησιάζουν την τελειότητα.

# 7 Συμπεράσματα

Στην παρούσα διπλωματική εργασία αναπτύχθηκαν έννοιες και αναφέρθηκαν στοιχεία και εφαρμογές που αφορούν την περιρρέουσα νοημοσύνη, τα περιβάλλοντά της, τη δομή τους και τα βασικά τους χαρακτηριστικά.

Στο πρακτικό μέρος της διπλωματικής εργασίας υλοποιήθηκε εφαρμογή που αξιοποίησε συγκεκριμένους αισθητήρες των έξυπνων κινητών τηλεφώνων. Η υλοποίηση της συγκεκριμένης αναπτυχθείσας εφαρμογής έδωσε τη δυνατότητα να μελετηθεί το λειτουργικό σύστημα Android και οι αισθητήρες των έξυπνων κινητών τηλεφώνων. Το επιταχυνσιόμετρο και το GPS αποδείχθηκε ότι είναι χρήσιμα εργαλεία για τον προσδιορισμό των δραστηριοτήτων που βασίζονται σε κινήσεις του έξυπνου κινητού τηλεφώνου.

Η εφαρμογή που αναπτύχθηκε στέλνει στοιχεία σε μια βάση δεδομένων που είναι αναρτημένη σε έναν εξυπηρετητή στο διαδίκτυο. Συγκεκριμένα στέλνονται πληροφορίες για τις διάφορες μετακινήσεις καθώς και για τη θέση στάθμευσης του αυτοκινήτου ενός χρήστη έξυπνου κινητού τηλεφώνου. Βασικός στόχος της εφαρμογής ήταν να ανιχνεύσουμε τις ανθρώπινες δραστηριότητες και να εντοπίσουμε τη θέση της στάθμευσης του αυτοκινήτου ενός χρήστη έξυπνου κινητού τηλεφώνου με μικρή απόκλιση από την πραγματική. Οι μετρήσεις που διενεργήθηκαν χάριν της παρούσας διπλωματικής εργασίας έδειξαν ότι υπάρχουν μικρές ατέλειες που οφείλονται στις δυνατότητες του κινητού τηλεφώνου και στην σύνδεση του με το διαδίκτυο οπότε κάνοντας χρήση τελευταίας τεχνολογίας κινητό τηλέφωνο μπορεί να περιοριστεί ή και να εξαλειφθεί η περίπτωση σφάλματος. Επίσης η διαρκής αναβάθμιση του λειτουργικού συστήματος που χρησιμοποιήθηκε (android) θα βοηθήσει στη βελτίωση της εφαρμογής. Στην περίπτωση δε που δεν είναι εφικτή η διαρκής αποστολή δεδομένων στη βάση λόγω μη σύνδεσης του κινητού με το διαδίκτυο αυτά τα δεδομένα δύνανται να φυλάσσονται σε προσωρινά αρχεία στη μνήμη του κινητού και να αποστέλλονται μαζικά μόλις ανιχνευθεί διαθέσιμο ασύρματο δίκτυο.

Εφικτός στόχος της εφαρμογής είναι να εμπλουτιστεί με πάρα πολλά δεδομένα η απομακρυσμένη βάση ώστε να αποτελέσει μια βάση big data που θα αξιοποιηθεί δίνοντας χρήσιμες πληροφορίες σε επιστημονικές ομάδες, επιχειρήσεις και τελικούς ενδιαφερόμενους. Βέβαια η έρευνα δείχνει ότι η ανάλυση, ο συσχετισμός, η μεταφορά, η ανταλλαγή κλπ. των big data της δημιουργηθείσας βάσης θα μπορούσαν να αποτελέσουν μια παραβίαση της ιδιωτικής ζωής των χρηστών θέμα που θα κληθούν να λύσουν οι επιστήμονες και οι ανεξάρτητες αρχές που ασχολούνται με την προστασία .προσωπικών δεδομένων.

Ως γνωστόν, τα τελευταία χρόνια το τοπίο στις βάσεις δεδομένων αλλάζει με γοργούς ρυθμούς με αποτέλεσμα ολοένα και περισσότερες εταιρείες να αποφασίζουν πως η καλύτερη λύση για τη χρήση μεγάλων όγκων δεδομένων είναι η μετάβαση από σχεσιακά σε

μη-σχεσιακά συστήματα. Επομένως, θα ήταν εξαιρετικά ενδιαφέρον το θέμα του επανασχεδιασμού της βάσης της εφαρμογής σύμφωνα με τη NoSQL φιλοσοφία. Ανεξάρτητα από το ποιο σύστημα NoSQL θα χρησιμοποιηθεί, είναι αντιληπτό πως η υιοθέτηση ενός μη-σχεσιακού μοντέλου και το ξαναχτίσιμο της βάσης γύρω από αυτό θα ήταν ένα πρωτοποριακό βήμα.

Ο απώτερος στόχος παρόμοιων επιστημονικών ερευνών θα είναι προς μια κατεύθυνση όπου οι έξυπνες συσκευές του περιβάλλοντος του ανθρώπου θα του παρέχουν εξατομικευμένες πληροφορίες χωρίς να είναι απαραίτητη οποιαδήποτε είδους διάδραση με αυτές και χωρίς να αντιλαμβάνεται την ύπαρξή τους..



# Βιβλιογραφία

- Aarts, E., Marzano, S. (2003). *The New Everyday : Views on Ambient Intelligence* . Rotterdam: 010 Publishers
- Adelstein , F., Gupta , K.S., S., Richard III , G., Schwiebert , L. (2004). *Fundamentals of Mobile and Pervasive Computing*. United States: McGraw-Hill
- Ahola, J. (2001) *Ambient Intelligence*. ERCIM NEWS, 47 October 2001, 8.
- Ark , W. S. , Selker, T. (1999). *A look at human interaction with pervasive computers*. California : IBM SYSTEMS JOURNAL, 38(4)
- Bieliková, M., Krajcovic, T. (2001). *Ambient Intelligence within a Home Environment* . ERCIM NEWS, 47 October 2001, 12-13.
- Dix, A., Finlay, J., Abowd, G., Beale, R. (2004). *Human – Computer Interaction*. Prentice Hall
- Lee, W. M. (2011). *Beginning Android Application Development*. Indianapolis : Wiley Publishing
- Norman, A. D. (1999). *The Invisible Computer: Why Good Products Can Fail, the Personal Computer Is So Complex, and Information Appliances Are the Solution*. USA : MIT Press
- Meier, R. (2010). *Professional Android 2 Application Development*. Indianapolis : Wiley Publishing
- Murphy, M. (2010). *Android Programming Tutorials, 3rd Edition*. USA : CommonsWare
- Palen, L., Dourish, P. (2003) *Unpacking “Privacy” for a Networked World*. CHI 2003, April 5–10, 2003, Ft. Lauderdale, Florida, USA.
- Satyanarayanan, M. (2001). *Pervasive Computing: Vision and Challenges*. IEEE Personal Communications
- Stachowicz, M., S., Kochanska, M., E. (1987). *Fuzzy modeling of the process*. Proc. of Second International Fuzzy Systems Association Congress, Tokyo, 1987.
- Stefanidis, C. (2001). *Ambient Intelligence in the Context of Universal Access*. ERCIM NEWS, 47, October 2001, 10-11.
- Weber, W., Rabaey, J. M., Aarts, E. (2005). *Ambient Intelligence*. Berlin, Heidelberg, New York: Springer
- Weiser, M., Brown, S. J. (1996). *Designing Calm Technology*. PowerGrid Journal, 1.01, July 1996
- Weiser, M. (1994). *The world is not a desktop*. Interactions, 1(1), January 1994, 7-8.

Weiser, M. (1993). *Hot Topics: Ubiquitous Computing*. IEEE Computer, 26(3), October 1993, 71-72.

Weiser, M. (1991). The Computer for the Twenty-First Century. Scientific American, 265(3), September 1991, 94-104.

Zadeh A. L. (1965). *Fuzzy Sets*. Information and Control 8 , 338-353

Zimmermann , H.-J. (2001) , *Fuzzy Set Theory -- and Its Applications*. Boston, Dordrecht, London : Kluwer Academic Publishers

## Ηλεκτρονικές αναφορές

<http://developer.android.com/index.html>

# Παράρτημα 1

## Main\_activity.java

```
package com.example.diplwmatikh;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import android.widget.ListAdapter;
import android.widget.SimpleAdapter;
import android.widget.Toast;
import org.apache.http.NameValuePair;
import org.json.JSONException;
import org.json.JSONObject;
import org.json.JSONArray;
import com.example.test.TestService.server;
import android.annotation.SuppressLint;
import android.app.Activity;
import android.app.ListActivity;
import android.support.v7.app.ActionBarActivity;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.Window;
import android.widget.Button;

public class MainActivity extends ListActivity {
    private static final String TAG_SUCCESS = "success";
    private static final String TAG_PRODUCTS = "mobile2_data";
    private static final String TAG_PID = "x";
    private static final String TAG_NAME = "y";
    private static final String TAG_z = "z";
    private static final String TAG_t = "t";

    ArrayList<HashMap<String, String>> productsList;
    JSONParser jsonParser = new JSONParser();
    Context context;
```

```

private static String url_all_products =
"http://www.eretria24.gr/mobile2/get_all_acceldata.php";
JSONArray products = null;
@Override
protected void onCreate(Bundle savedInstanceState) {
    requestWindowFeature(Window.FEATURE_NO_TITLE);
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Button btnStart = (Button) findViewById(R.id.btnStart);
    Button btnStop = (Button) findViewById(R.id.btnStop);
    Button btnmaps = (Button) findViewById(R.id.btnmaps);
    Button button1 = (Button) findViewById(R.id.button1);
    Button button2 = (Button) findViewById(R.id.button2);
    productList = new ArrayList<HashMap<String, String>>();

    btnStart.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if (isConnected()){
                new LoadAllProducts().execute();
                Toast toast4 = Toast.makeText(getApplicationContext(), "ο πίνακας
φορτώνει", Toast.LENGTH_LONG );
                toast4.show();
            }
            else{
                Toast toast3 = Toast.makeText(getApplicationContext(), "δεν έχετε
σύνδεση", Toast.LENGTH_LONG );
                toast3.show();
            }
        }
    });
    button2.setOnClickListener(new View.OnClickListener() {

        @Override

        public void onClick(View view) {
            Intent i = new Intent(MainActivity.this, TestService.class);
            MainActivity.this.startService(i);
            Toast toast5 = Toast.makeText(getApplicationContext(), "η
υπηρεσία ξεκίνησε", Toast.LENGTH_LONG );
            toast5.show();
        }
    });

    btnStop.setOnClickListener(new View.OnClickListener() {
        @SuppressWarnings("ShowToast")
        @Override
        public void onClick(View view) {
            Toast toast1 = Toast.makeText(getApplicationContext(), "γίνεται
έξοδος", Toast.LENGTH_LONG );
            toast1.show();
            finish();
        }
    });
}

```

```

button1.setOnClickListener(new View.OnClickListener() {
    @SuppressWarnings("ShowToast")
    @Override
    public void onClick(View view) {
        Toast toast2 = Toast.makeText(getApplicationContext(), "η υπηρεσία
τερματίστηκε, γίνεται έξοδος", Toast.LENGTH_LONG );
        toast2.show();
        Intent i = new Intent(MainActivity.this, TestService.class);
        MainActivity.this.stopService(i);
        finish();
    }
});
btnmaps.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        if (isConnected()){
            Toast toast6 = Toast.makeText(getApplicationContext(), "ο χάρτης
φορτώνει", Toast.LENGTH_LONG );
            toast6.show();
            Intent i = new Intent(getApplicationContext(), maps.class);
            startActivity(i);
        }else{
            Toast toast7 = Toast.makeText(getApplicationContext(),
"δεν έχετε σύνδεση", Toast.LENGTH_LONG );
            toast7.show();

        }
    }
});

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.action_settings) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}

```

```

private boolean isConnected() {
    ConnectivityManager connMgr = (ConnectivityManager)
        getSystemService(Activity.CONNECTIVITY_SERVICE);
    NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();
    if (networkInfo != null && networkInfo.isConnected())
        return true;
    else
        return false;
}

class LoadAllProducts extends AsyncTask<String, String, String> {
    protected String doInBackground(String... args) {
        List<NameValuePair> params = new ArrayList<NameValuePair>();
        JSONObject json = jsonParser.makeHttpRequest(url_all_products, "GET",
params);
        Log.d("All Products: ", json.toString());
        try {
            int success = json.getInt(TAG_SUCCESS);
            if (success == 1) {
                products = json.getJSONArray(TAG_PRODUCTS);
                for (int i = 0; i < products.length(); i++) {
                    JSONObject c = products.getJSONObject(i);
                    String pid = c.getString(TAG_PID);
                    String name = c.getString(TAG_NAME);
                    String z = c.getString(TAG_z);
                    String t = c.getString(TAG_t);

                    HashMap<String, String> map = new HashMap<String, String>();

                    map.put(TAG_PID, pid);
                    map.put(TAG_NAME, name);
                    map.put(TAG_z, z);
                    map.put(TAG_t, t);

                    productsList.add(map);
                }
            } else {

                Intent i = new Intent(getApplicationContext(),
                    server.class);

                i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                startActivity(i);
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }

        return null;
    }
}

```

```

protected void onPostExecute(String file_url) {

    runOnUiThread(new Runnable() {
        public void run() {
            ListAdapter adapter = new SimpleAdapter(
                MainActivity.this, productsList, R.layout.list_item, new String[]
{TAG_PID, TAG_NAME, TAG_z, TAG_t},
                new int[] { R.id.name , R.id.z , R.id.t      });
            setListAdapter(adapter);
        }
    });
}
}
}

```

## Παράρτημα 2

### maps.java

```
package com.example.diplwmatikh;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import org.apache.http.NameValuePair;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.ListAdapter;
import android.widget.SimpleAdapter;
import com.example.test.TestService.server;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.MapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v4.app.FragmentActivity;
import android.util.Log;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.widget.Toast;

public class maps extends FragmentActivity {
    private static final String TAG_SUCCESS = "success";
    private static final String TAG_PRODUCTS = "mobile2_data";
    private static final String TAG_PID = "x";
    private static final String TAG_NAME = "y";
    private static final String TAG_z = "z";
    private static final String TAG_t = "t";
    public double lo = 20;
    public double la = 20;

    LatLng parking = new LatLng (lo ,la);
    static final LatLng TutorialsPoint = new LatLng(21 , 57);
    private GoogleMap googleMap;

    JSONParser jsonParser = new JSONParser();
    Context context;
```



```

        private static String url_all_products =
"http://www.eretria24.gr/mobile2/get_all_acceldata.php";
        JSONArray products = null;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.maps);
    try {
        if (googleMap == null) {
            googleMap = ((MapFragment) getFragmentManager().
                findFragmentById(R.id.map)).getMap();
        }
        googleMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);
    } catch (Exception e) {
        e.printStackTrace();
    }
    new LoadAllProducts().execute();
}

@Override
public boolean onCreateOptionsMenu(Menu menu) { is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.action_settings) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}
class LoadAllProducts extends AsyncTask<String, String, String> {

protected String doInBackground(String... args) {
    List<NameValuePair> params = new ArrayList<NameValuePair>();
    JSONObject json = jsonParser.makeHttpRequest(url_all_products, "GET", params);
    Log.d("All Products: ", json.toString());
    try {
        int success = json.getInt(TAG_SUCCESS);
        if (success == 1) {
            products = json.getJSONArray(TAG_PRODUCTS);
            for (int i = 0; i < products.length(); i++) {

                JSONObject c = products.getJSONObject(i);
                String pid = c.getString(TAG_PID);
                String name = c.getString(TAG_NAME);
                String z = c.getString(TAG_z);
                String t = c.getString(TAG_t);
                HashMap<String, String> map = new HashMap<String, String>();

```

```

        map.put(TAG_PID, pid);
        map.put(TAG_NAME, name);
        map.put(TAG_z, z);
        map.put(TAG_t, t);
        double doublepid = Double.parseDouble(pid);
        if (doublepid== 1){
            parking = new LatLng (Double.parseDouble(z) ,Double.parseDouble(t));
        };
    }
} else {

    Intent i = new Intent(getApplicationContext(),
        server.class);
    i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    startActivity(i);
}
} catch (JSONException e) {
    e.printStackTrace();
}

return null;
}

protected void onPostExecute(String result) {

    Marker TP = googleMap.addMarker(new MarkerOptions().
        position(TutorialsPoint).title("TutorialsPoint"));
    Marker parkedmark = googleMap.addMarker(new MarkerOptions().
        position(parking).title("here is your car"));

googleMap.moveCamera(CameraUpdateFactory.newLatLngZoom(parking,15));
    // Zoom in
    googleMap.animateCamera(CameraUpdateFactory.zoomIn());
    // Zoom out
googleMap.animateCamera(CameraUpdateFactory.zoomTo(15), 2000, null);
}

}

}

```

# Παράρτημα 3

## Testservice.java

```
package com.example.diplwmatikh;

import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.List;
import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONArray;
import android.app.Activity;
import android.app.ListActivity;
import org.json.JSONException;
import org.json.JSONObject;
import android.app.Service;
import android.content.ComponentName;
import android.content.Context;
import android.content.Intent;
import android.content.ServiceConnection;
import android.graphics.Color;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.location.GpsStatus;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.IBinder;
import android.util.Log;
import android.widget.ListAdapter;
import android.widget.SimpleAdapter;
import android.widget.Toast;

public class TestService extends Service implements SensorEventListener {
    JSONParser jsonParser = new JSONParser();
    private static String url_all_products =
"http://www.eretria24.gr/mobile2/get_all_acceldata.php";
    private static String url_server =
"http://www.eretria24.gr/mobile2/create_accel_data.php";
    private static final String TAG_SUCCESS = "success";

    private static final String TAG_PRODUCTS = "mobile2_data";
    private static final String TAG_PID = "x";
```

```

private static final String TAG_NAME = "y";
private static final String TAG_z = "z";
private static final String TAG_t = "t";
ArrayList<HashMap<String, String>> productsList;

    // JSON Node names
double x = 0;
    double y = 0;
    double z = 0;
    double valuetime = System.currentTimeMillis();
    private boolean stopvarc = true;
    private boolean stopvara = true;
    private boolean stopvarb = true;
    private int timesa = 0;
    private int timesb = 0;
    private int timesc = 0;
    boolean walking = false;
    int walkingtimes=0;
double speed =0;
    double latdif = 0;
    double longdif = 0;
    double speed2 = 0;
    double longitude = 0;
    double latitude = 0;
    double oldlatitude = 0;
    double oldlongitude = 0;
    long time1=System.currentTimeMillis();
    private SensorManager mSensorManager;
    private Sensor mAcceleration;
    long oldtime = System.currentTimeMillis();
    boolean carrunning = false;
    boolean carstoped = false;
    double laststoplat = 0 ;
    double laststoplong = 0 ;
    double parkedboolean = 0;
    long speedtime = 0;
long dt= 0;
    Context context;
    long time = 1;
    JSONArray products = null;
    long timestamp = System.currentTimeMillis();
double flag = 0;

```

```

@Override
public void onCreate() {

    mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
    mAcceleration =
    mSensorManager.getDefaultSensor(Sensor.TYPE_LINEAR_ACCELERATION);
    mSensorManager.registerListener(this, mAcceleration ,
    SensorManager.SENSOR_DELAY_FASTEST);

```

```

        locationManager = (LocationManager) this
            .getSystemService(Context.LOcation_SERVICE);
    }
    @Override
    public void onDestroy() {
        super.onDestroy();
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        mSensorManager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);
        mAcceleration =
mSensorManager.getDefaultSensor(Sensor.TYPE_LINEAR_ACCELERATION);
        mSensorManager.registerListener(this, mAcceleration ,
SensorManager.SENSOR_DELAY_FASTEST);

        locationManager = (LocationManager) this
            .getSystemService(Context.LOcation_SERVICE);

        LocationListener locationListener = new LocationListener() {
            public void onLocationChanged(Location location) {
                time = System.currentTimeMillis();
                latitude = location.getLatitude();
                longitude = location.getLongitude();
                latdif = latitude - oldlatitude;

                longdif = longitude - oldlongitude;
                speed2 = 3600*1000*(Math.sqrt (( latdif * 111 ) * ( latdif * 111 ) + ( longdif *
87 ) * ( longdif * 87)))/(time - oldtime);

                oldlatitude = latitude;
                oldlongitude = longitude;
                oldtime = System.currentTimeMillis();

                if (speed2>8 ){
                    carrunning = true;
                    carstoped = false;
                }
                else {
                    if (carrunning && speed<5 && speed2<5){
                        carstoped = true;
                        carrunning = false;
                        laststoplat = latitude;
                        laststoplong = longitude;
                    }
                }

                if (time1 < System.currentTimeMillis() - 7000 &&
isConnected() && location != null){

                    parkedboolean = 0;
                    new server().execute();
                }
            }
        };
    }

```

```

        time1=System.currentTimeMillis();

    }

    }

    public void onStatusChanged(String provider, int status,
        Bundle extras) {

    }

    public void onProviderEnabled(String provider) {

    }

    public void onProviderDisabled(String provider) {

    }

};

        locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0,
locationListener);

    return START_STICKY;
}

    private boolean isConnected() {
        ConnectivityManager connMgr = (ConnectivityManager)
getSystemService(Activity.CONNECTIVITY_SERVICE);
        NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();
        if (networkInfo != null && networkInfo.isConnected())
            return true;
        else
            return false;
    }

    @Override
    public IBinder onBind(Intent intent) {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public void onSensorChanged(SensorEvent event) {

        if (event.sensor.getType() == Sensor.TYPE_LINEAR_ACCELERATION) {
            //ypologismos V me accel
            float a = event.values[0];
            float b = event.values[1];
            float c = event.values[2];
            parkedboolean = 0 ;
            dt = System.currentTimeMillis() – speedtime;
            speedtime = System.currentTimeMillis();

```

```

speed = speed + Math.sqrt((a*dt)*(a*dt)+ (b*dt)*(b*dt)+ (c*dt)*(c*dt));

    if ( stopvarc == true && c < 2.5 ) {
timesc = timesc + 1;
stopvarc = false;
}

if (stopvarc == false && c > 2.5) {
    stopvarc = true;
}

if ( stopvara == true && a < 2.5 ) {
    timesa = timesa + 1;
    stopvara = false;
}

if (stopvara == false && a > 2.5) {
    stopvara = true;
}

if ( stopvarb == true && b < 2.5 ) {
    timesb = timesb + 1;
    stopvarb = false;
}

if (stopvarb == false && b > 2.5) {
    stopvarb = true;
}

if ( System.currentTimeMillis() - valuetime >6000){
    valuetime = System.currentTimeMillis();
    if (timesa >5 && timesb>5 && timesc>5) {

        walkingtimes = walkingtimes +1;
        timesb=0;
        timesc=0;
        walking = true;
    }else{
        walking = false;
        timesa = 0;
        timesb = 0;
        timesc = 0;
    }

}

}

if (carstoped && walking && isConnected()){
    carrunning = false;
    carstoped = false;
}

```

```

        walking=false;
        parkedboolean = 1;
        new server().execute();
    }
}

```

```

class server extends AsyncTask<String, String, String> {

```

```

    @Override
    protected void onPreExecute() {
    }

    protected String doInBackground(String... args) {

        String a = Double.toString(parkedboolean);
        String b = String.valueOf(time);
        String c = Double.toString(latitude);
        String d = Double.toString(longitude);
        List<NameValuePair> params = new ArrayList<NameValuePair>();
        params.add(new BasicNameValuePair("x", a));
        params.add(new BasicNameValuePair("y", b));
        params.add(new BasicNameValuePair("z", c));
        params.add(new BasicNameValuePair("t", d));
        JSONObject json = jsonParser.makeHttpRequest(url_server,
            "POST", params);

        return null ;
    }

    protected void onPostExecute(String file_url) {
        // dismiss the dialog once done
    }

}

@Override
public void onAccuracyChanged(Sensor sensor, int accuracy) {
    // TODO Auto-generated method stub
}

protected void onResume() {

    mSensorManager.registerListener(this, mAcceleration,
    SensorManager.SENSOR_DELAY_FASTEST);
}

protected void onPause() {
    mSensorManager.unregisterListener(this);
}
}

```