

Ατομική Διπλωματική Εργασία

**ΣΥΜΜΕΤΕΧΟΥΣΑ ΑΝΤΙΛΗΨΗ ΜΕ ΑΙΣΘΗΤΗΡΕΣ ΚΙΝΗΤΩΝ
ΣΥΣΚΕΥΩΝ**

Θεόφιλος Φωκάς

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2012

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Συμμετέχουσα Αντίληψη με Αισθητήρες Κινητών Συσκευών

Θεόφιλος Φωκάς

Επιβλέπων Καθηγητής

Γιώργος Πάλλης

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2012

Ευχαριστίες

Θα ήθελα να εκφράσω τις ευχαριστίες μου στον επιβλέποντα καθηγητή μου Δρ. Γιώργο Πάλλη, για την πολύ καλή συνεργασία που είχαμε καθ' όλο το χρονικό διάστημα εκπόνησης της διπλωματικής αυτής εργασίας. Οι συμβουλές του, οδηγίες αλλά και κατευθύνσεις, συνέβαλαν στο μέγιστο, για την επιτυχή κατά την άποψη μου ολοκλήρωση της εργασίας αυτής.

Ιδιαίτερες ευχαριστίες θα ήθελα να εκφράσω στους φοιτητές που εργάζονταν στο εργαστήριο για και ιδιαίτερα τον Νικόλα Λουλλούδη για την πολύτιμη και ανιδιοτελή βοήθεια που μου προσέφερε. Επιπλέον θα ήθελα να εκφράσω ένα μεγάλο ευχαριστώ στον Χαρίτων Ευσταθιάδη για την πολύτιμη υποστήριξη που μου πρόσφερε όλο αυτό το διάστημα.

Τέλος, θα ήθελα να εκφράσω ένα μεγάλο ευχαριστώ στην οικογένεια μου, και ιδιαίτερα στη μητέρα μου, που με στήριξαν ηθικά και ψυχολογικά σε ολόκληρη την πορεία μου κατά τη διάρκεια φοίτησης μου στο Πανεπιστήμιο Κύπρου αλλά και κατά τη δύσκολη αυτή περίοδο ολοκλήρωσης της διπλωματικής μου εργασίας.

Περίληψη

Τα τελευταία χρόνια η ραγδαία ανάπτυξη της τεχνολογίας μας έχει προσφέρει τεράστια υπολογιστική ισχύ, άφθονο αποθηκευτικό χώρο για δεδομένα και δυνατότητες επικοινωνίας που δεν είχαμε στο παρελθόν. Επιπλέον μικροί, φθηνοί αισθητήρες με τεράστιες δυνατότητες έχουν κάνει την παρουσία τους και είναι διαθέσιμοι. Σπουδαιότερο όμως επίτευγμα είναι το γεγονός πως όλες αυτές οι δυνατότητες είναι πλέον ενσωματωμένες στις κινητές συσκευές καθημερινής χρήσης, τα επονομαζόμενα έξυπνα τηλέφωνα, τα οποία έχουν καταλάβει το μεγαλύτερο μερίδιο αγοράς κινητών τηλεφώνων. Αυτό το γεγονός μας έχει δώσει την δυνατότητα να μπορούμε να υλοποιήσουμε ένα τεράστιο αριθμό πρωτοποριακών εφαρμογών στους τομείς της υγείας, κοινωνικής δικτύωσης, ασφάλειας και στην παρακολούθηση του περιβάλλοντος.

Παρόλες τις νέες δυνατότητες υπάρχουν ακόμα αρκετά προβλήματα που χρειάζεται να απαντηθούν. Παρόλη την ανάπτυξη στα τεχνικά χαρακτηριστικά των κινητών συσκευών η ανάπτυξη στην αποθήκευση ενέργειας δεν ήταν η ανάλογη έτσι και οι νέες συσκευές υποφέρουν από περιορισμένες ώρες αυτονομίας. Επιπλέον η εκτεταμένη χρήση αισθητήρων για συλλογή και ανάλυση δεδομένων στην κινητή συσκευή έχει σοβαρή επίπτωση στους πόρους της συσκευής και κατά συνέπεια στην μπαταρία. Η μεταφορά δεδομένων χρησιμοποιώντας δίκτυα όπως 3G, GPRS είναι γρήγορη όμως οι χρήστες έχουν περιορισμένο αριθμό δεδομένων που μπορούν να μεταφέρουν.

Στόχος της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη μιας πλατφόρμας ενδιάμεσου λογισμικού η οποία θα επιτρέπει στους προγραμματιστές να αναπτύξουν εύκολα, γρήγορα και αποδοτικά εφαρμογές αστικής αντίληψης χρησιμοποιώντας τις δυνατότητες που μας προσφέρουν οι σημερινές κινητές συσκευές. Οι εφαρμογές οι οποίες αναπτύσσονται χρησιμοποιώντας την πλατφόρμα μπορούν να έχουν επίγνωση των διαθέσιμων πόρων και ιδιαίτερων χαρακτηριστικών της συσκευής ούτως ώστε να μην αλλοιώνουν την εμπειρία χρήσης των κινητών συσκευών από τους ιδιοκτήτες τους.

Περιεχόμενα

Κεφάλαιο 1	1
Εισαγωγή	1
1.1 Ορισμός Προβλήματος.....	1
1.2 Συμμετέχουσα Αντίληψη.....	3
1.3 Υποκίνηση Εργασίας.....	4
1.4 Στόχοι Εργασίας.....	5
1.5 Περίγραμμα αναφοράς	6
Κεφάλαιο 2	7
Σχετική Βιβλιογραφία	7
2.1 Εισαγωγή.....	7
2.2 The Jigsaw Continuous Sensing Engine for Mobile Phone Applications.....	8
2.2.1 Πρόβλημα	8
2.2.2 Αρχιτεκτονική.....	8
2.3 MyExperience: A system for in situ tracing and capturing of user feedback on mobile phones.....	11
2.3.1 Πρόβλημα.....	11
2.3.2 Αρχιτεκτονική.....	11
2.4 SensLoc: Everyday Places and Paths using Less Energy	13
2.4.1 Πρόβλημα.....	13
2.4.2 Αρχιτεκτονική.....	13
2.5 SociableSense: Exploring the Trade-offs of Adaptive Sampling and Computation Offloading for Social Sensing	14
2.5.1 Πρόβλημα.....	14
2.5.2 Αρχιτεκτονική.....	14
2.6 Funf: Open Sensing Framework.....	16
2.6.1 Πρόβλημα.....	16
2.6.2 Αρχιτεκτονική.....	17
2.7 Σύγκριση συστημάτων	18
Πλατφόρμα ενδιάμεσου λογισμικού Feel The World.....	21
3.1 Πρόβλημα	21
3.2 Τεχνολογίες που χρησιμοποιήθηκαν.....	22
3.3 Αρχιτεκτονική Συστήματος.....	24
3.4 Ανάλυση Αρχιτεκτονικής	25
3.3.1 Εξυπηρετητής.....	25
3.3.2 Κινητή συσκευή	27
3.3.2.1 Data Layer	28
3.3.2.2 Monitoring Service	30
3.3.2.3 File Writer	30

3.3.2.4	Communication Layer.....	31
3.3.2.5	Utilities Services.....	31
3.3.2.6	Αλγόριθμος συμπίεσης	32
Κεφάλαιο 4		35
Υλοποίηση Συστήματος.....		35
4.1	Διάγραμμα κλάσεων.....	36
4.1.1	Resource Manager	36
4.1.2	Data.....	37
4.1.3	Sensing.....	38
4.2	Λεπτομέρειες υλοποίησης.....	39
Κεφάλαιο 5		41
Αξιολόγηση		41
5.1	Εκφραστικότητα πλατφόρμας.....	41
5.2	Εφαρμογή FeelMe.....	42
5.3.1	Γενικά συμπεράσματα	49
5.3	Ρύθμιση για κατανάλωση μπαταρίας	50
Κεφάλαιο 6		54
Συμπεράσματα – Μελλοντική Εργασία		54
6.1	Συμπεράσματα	54
6.2	Μελλοντική Εργασία.....	55
Βιβλιογραφία		57
Παράρτημα Α.....		1
Class BaseBackgroundSensor		1
MonitoringService		5
FileWriterHelper		6
BluetoothManager.....		6
DeflateCompression.....		8

Σχήματα

Εικόνα 1: Αριθμός εφαρμογών.....	2
Εικόνα 2: Τομείς συμμετέχουσας αντίληψης.....	4
Εικόνα 3: Microphone Pipeline - Jigsaw	10
Εικόνα 4: MyExperience Architecture.....	12
Εικόνα 5: Xml Configuration file - MyExperience.....	12
Εικόνα 6: SensLoc Architecture	13
Εικόνα 7: SociableSense architecture	15
Εικόνα 8: SociableSense dynamic monitoring	16
Εικόνα 9: Funf platform.....	17

Εικόνα 10: Smartphone operating System Share.....	23
Εικόνα 11: Feel the World architecture	24
Εικόνα 12: Feel the World architecture – server side	25
Εικόνα 13: Feel the World architecture – client side	27
Εικόνα 14: Feel the World – class diagram.....	36
Εικόνα 15: Feel the World – Data implementation.....	37
Εικόνα 16: Δεδομένα - IData	37
Εικόνα 17: Feel the World – Base Background extend.....	38
Εικόνα 18: BaseSensor	39
Εικόνα 19: FeelMe Setup screen.....	43
Εικόνα 20: FeelMe Setup screen.....	44
Εικόνα 21: FeelMe Display screen	45
Εικόνα 22: FeelMe Map screen	46
Εικόνα 23: FeelMe Resources screen.....	47
Εικόνα 24: FeelMe - robustness.....	49

Κεφάλαιο 1

Εισαγωγή

1.1 Ορισμός Προβλήματος.....	1
1.2 Συμμετέχουσα Αντίληψη.....	3
1.3 Υποκίνηση Εργασίας.....	4
1.4 Στόχοι Εργασίας.....	5
1.5 Περίγραμμα αναφοράς	6

Στο κεφάλαιο αυτό γίνεται μια εισαγωγή στο πεδίο με το οποίο ασχοληθήκαμε στην παρούσα διπλωματική εργασία. Αρχικά γίνεται μια περιγραφή της παρούσας κατάστασης όπως αυτή έχει διαμορφωθεί στις μέρες λόγο της ραγδαίας ανάπτυξης της τεχνολογίας. Ακολούθως παρουσιάζονται τα κίνητρα της διπλωματικής εργασίας και η συνεισφορά της.

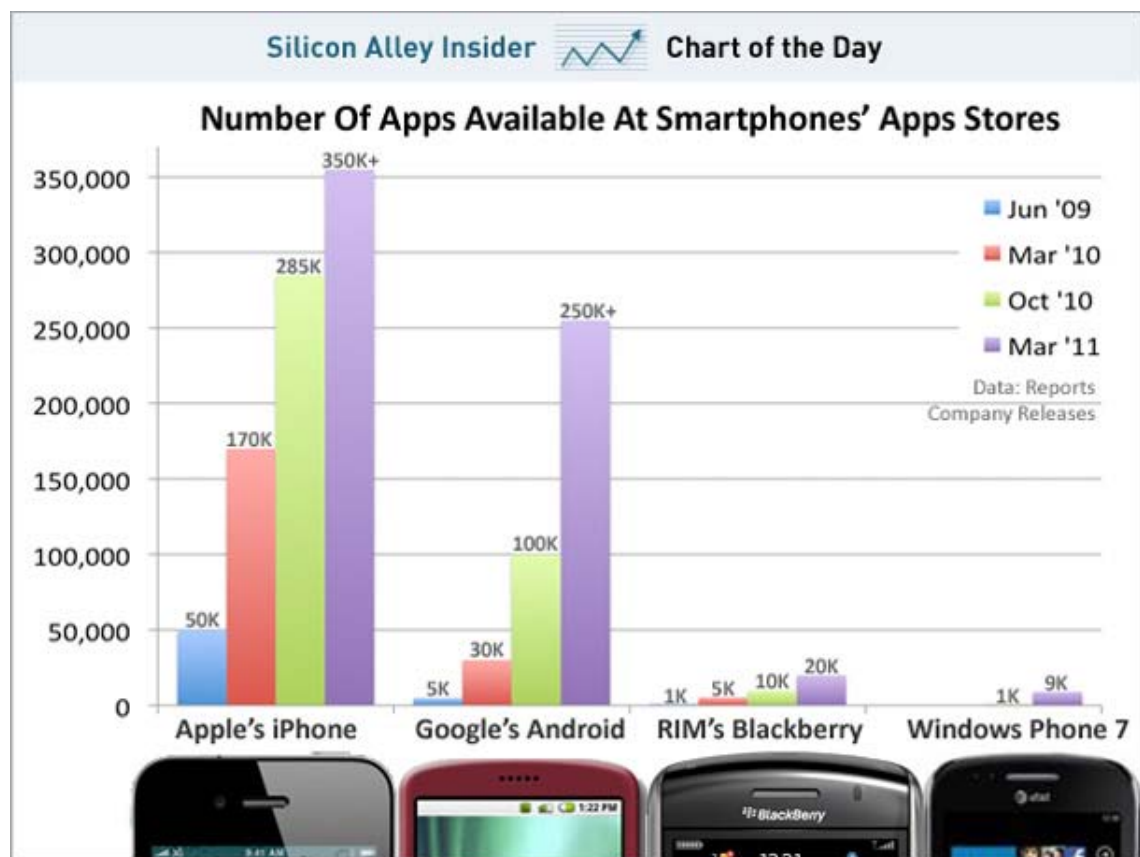
1.1 Ορισμός Προβλήματος

Η ολοκλήρωση της κατασκευής του πρώτου κινητού τηλεφώνου έγινε το 1973 στη Νέα Υόρκη από για λογαριασμό της Motorola. Περίπου 40 χρόνια μετά τα κινητά τηλέφωνα της εποχής μας δεν θυμίζουν σε τίποτα το τηλέφωνο που είχε εμπνευστεί ο Δρ. Μάρτιν Κούπερ το 1973. Τα κινητά είναι οι πιο ισχυρές συσκευές επικοινωνίας. Έρχονται εξοπλισμένες με πολυπύρηνους επεξεργαστές, άφθονη μνήμη και αποθηκευτικό χώρο. Χαρακτηριστικό παράδειγμα είναι το γεγονός πως πρόσφατα έχουν κυκλοφορήσει κινητά τηλέφωνα με τετραπύρηνους επεξεργαστές. Τα έξυπνα κινητά έχουν καταλάβει το μεγαλύτερο μερίδιο της αγοράς και προβλέπεται πως σύντομα οι χρήστες έξυπνων κινητών θα είναι περισσότεροι από αυτούς των συμβατικών.

Επιπλέον της ανάπτυξης των τεχνικών χαρακτηριστικών των τηλεφώνων, όπως επεξεργαστής, μνήμη κλπ, υπήρξε η ανάγκη για ενσωμάτωση αισθητήρων. Αρχικά

ενσωματώθηκε ο δέκτης GPS έτσι ώστε ο χρήστης να μπορέσει ανά πάσα στιγμή να γνωρίζει την θέση του, αλλά και να απολαμβάνει εξειδικευμένες υπηρεσίες όπως πλοήγηση (navigation). Στη συνέχεια ακολούθησε η ενσωμάτωση ενός μεγάλου αριθμού αισθητήρων όπως επιταχυνσιόμετρο, πυξίδα, γυροσκόπιο, φωτοαισθητήρες, αισθητήρες μέτρησης απόστασης κτλ. Σκοπός όλων αυτών των αισθητήρων ήταν η παροχή ευκολίας και μιας μοναδικής εμπειρίας χρήσης στους ιδιοκτήτες κινητών συσκευών.

Η παρουσία όλων αυτών των τεχνικών χαρακτηριστικών αλλά και των αισθητήρων σε συσκευές που οι άνθρωποι χρησιμοποιούν στην καθημερινότητά τους και που έχουν πάντοτε μαζί τους, έδωσαν την δυνατότητα να αναπτυχθούν ένας τεράστιος αριθμός καινοτόμων εφαρμογών στους τομείς της υγείας, κοινωνικής δικτύωσης, ασφάλειας και στην παρακολούθηση του περιβάλλοντος.



Εικόνα 1: Αριθμός εφαρμογών

Μια από τις σημαντικότερες παραμέτρους είναι πως χρησιμοποιώντας τις τεχνολογίες που παρέχουν οι έξυπνες συσκευές μπορούμε να συλλέγουμε δεδομένα που αφορούν την καθημερινότητα των χρηστών. Για παράδειγμα πώς χρησιμοποιούν

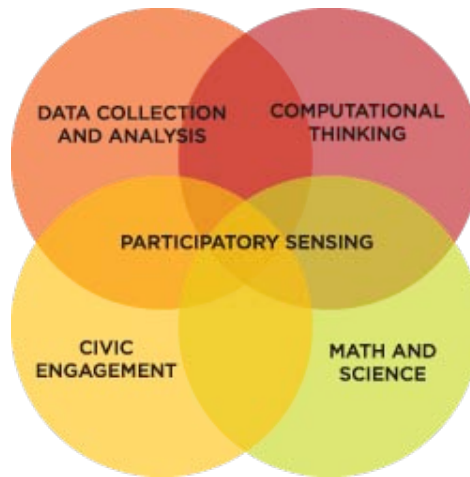
τα τηλέφωνα τους οι άνθρωποι όταν βρίσκονται σε τραίνο. Τα δεδομένα ήταν και είναι απαραίτητα στην προώθηση της έρευνας και στην εξαγωγή απαραίτητων συμπερασμάτων με σκοπό την βελτίωση της ποιότητας ζωής και των παρεχόμενων υπηρεσιών. Η συλλογή τους πάντοτε στοίχιζε σε χρόνο και σε χρήμα. Στις μέρες μας είναι εφικτή η συλλογή τεράστιου όγκου δεδομένων μέσα από την καθημερινότητα των χρηστών με πολύ χαμηλό αντλώντας δεδομένα από τις κινητές συσκευές. Ο κλάδος αυτός ονομάζεται Participatory Sensing.

1.2 Συμμετέχουσα Αντίληψη

Με τον όρο συμμετέχουσα αντίληψη (participatory sensing) αναφερόμαστε σε μία προσέγγιση συλλογής δεδομένων όπου ο χρήστης συνειδητά επιλέγει να μοιραστεί κάποια δεδομένα του χωρίς να έχει οποιοδήποτε προσωπικό ή οικονομικό συμφέρον. Άτομα, μεμονωμένα είτε σε ομάδες, συνειδητά πραγματοποιούν την συλλογή των δεδομένων χρησιμοποιώντας τις κινητές τους συσκευές.

Τα συστήματα αστικής αντίληψης που υιοθετούν αυτή την προσέγγιση πρέπει να υποστηρίζουν διάφορα στάδια αποφάσεων. Ο χρήστης ενός τέτοιου συστήματος πρέπει να είναι σε θέση να επιλέγει τα δεδομένα τα οποία επιθυμεί να μοιράζεται καθώς και το επίπεδο ιδιωτικότητας που επιθυμεί. Η σχεδίαση ενός τέτοιου συστήματος χρησιμοποιεί εργαλεία που βοηθούν τους χρήστες να δημοσιεύσουν, να αναζητήσουν, να ερμηνεύσουν και να επαληθεύσουν τις πληροφορίες που συλλέχθηκαν χρησιμοποιώντας κοινωνικά ανεπτυγμένες τεχνικές ενθάρρυνσης της συμμετοχής του κοινού.

Η υπολογιστική σκέψη που απαιτείται για να πλαισιώσει το πρόβλημα, τον σχεδιασμό του πειράματος, την ανάπτυξη του λογισμικού και την ανάλυση των αποτελεσμάτων παρέχει πολύ πλούσιες δυνατότητες μάθησης στο πλαίσιο των μαθηματικών ή της επιστήμης της πληροφορικής.



Εικόνα 2: Τομείς συμμετέχουσας αντίληψης

1.3 Υποκίνηση Εργασίας

Τα τελευταία χρόνια έγινε μια μεγάλη προσπάθεια για δημιουργία εφαρμογών που αφορούν την συλλογή δεδομένων. Η δημιουργία τέτοιων εφαρμογών χαρακτηρίζεται από αρκετή πολυπλοκότητα. Οι προγραμματιστές εκτός από την λογική που θα ενσωματώσουν στην εφαρμογή τους είναι επιφορτισμένοι και με τους περιορισμούς που έχουν οι κινητές συσκευές σε θέματα ενέργειας αλλά και υπολογιστικής ισχύς. Το γεγονός πως οι εφαρμογές αυτές απευθύνονται σε μεγάλο αριθμό διαφορετικών συσκευών με διαφορετικά τεχνικά χαρακτηριστικά κάνει το πρόβλημα ακόμη δυσκολότερο. Τέλος, οι εφαρμογές που υλοποιούνται είναι συνήθως μιας χρήσης αφού είναι πιο εύκολη η δημιουργία καινούργιας εφαρμογής για να καλύψει τις ανάγκες ενός νέου αντικειμένου αίσθησης, παρά να γίνει τροποποίηση της παλιάς. Αυτό οδηγεί στην ανάγκη να γίνει ξανά υλοποίηση κώδικα και ιδεών, με αποτέλεσμα τη σπατάλη χρόνου κατά τη δημιουργία εφαρμογών από την αρχή.

Λύση σε αυτά τα προβλήματα μπορούν να παρέχουν οι πλατφόρμες ενδιάμεσου λογισμικού (middleware). Οι πλατφόρμες ενδιάμεσου λογισμικού είναι προγράμματα που παρέχουν υπηρεσίες επιπλέον από αυτές του λειτουργικού συστήματος. Βοηθούν στην ανάπτυξη εφαρμογών παρέχοντας εξειδικευμένες συναρτήσεις που διαφορετικά ο προγραμματιστής θα χρειαζόταν να υλοποιήσει ο ίδιος. Προκειμένου να αντιμετωπιστούν τα προβλήματα που έχουμε προαναφέρει θα

ήταν ιδανική η δημιουργία μίας επαναχρησιμοποιήσιμης πλατφόρμας ανοικτού λογισμικού που θα παρέχει εύκολη πρόσβαση στους αισθητήρες που βρίσκονται ενσωματωμένοι στις κινητές συσκευές αλλά και σε εξωτερικούς αισθητήρες που επικοινωνούν με αυτές μέσω κάποιου τύπου σύνδεσης. Το ενδιαμέσο αυτό λογισμικό πρέπει να παρέχει εύκολη διαχείριση των αισθητήρων, εύκολη πρόσβαση στους πόρους του τηλεφώνου και γνώση των ιδιαίτερων χαρακτηριστικών της κινητής συσκευής. Επιπλέον πρέπει να παρέχει ευκολίες εξοικονόμησης πόρων. Τέλος, η πλατφόρμα πρέπει να μπορεί να χρησιμοποιηθεί από χρήστες που δεν έχουν μεγάλη προγραμματιστική πείρα δίνοντας τους την δυνατότητα να αναπτύξουν τις δικές τους εφαρμογές.

1.4 Στόχοι Εργασίας

Στόχος της διπλωματικής εργασίας είναι η εκπόνηση μίας καινοτόμας πλατφόρμας λογισμικού η οποία θα ευκολύνει την ανάπτυξη εφαρμογών που αφορούν την συμμετέχουσα αντίληψη. Πρόκληση στην υλοποίηση της πλατφόρμας είναι να αντιμετωπίσουμε το πρόβλημα της συνεχούς αίσθησης (continuous sensing). Πιο συγκεκριμένα οι εφαρμογές που αναπτύσσονται χρησιμοποιώντας την πλατφόρμα πρέπει να μπορούν συνεχώς να συλλέγουν δεδομένα χωρίς να επηρεάζεται αρνητικά η εμπειρία χρήσης με την κινητή συσκευή.

Ο σχεδιαστής της εφαρμογής πρέπει να μπορεί να ρυθμίσει την συλλογή δεδομένων ως προς τον ρυθμό συλλογής και την διάρκεια της κάθε συλλογής. Επιπλέον η πλατφόρμα πρέπει να παρέχει πληροφορίες που αφορούν τους διαθέσιμους πόρους της συσκευής. Γνωρίζοντας τους διαθέσιμους πόρους παρέχουμε την δυνατότητα της δυναμικής ρύθμισης της συλλογής δεδομένων αναλόγως των πόρων.

Η πλατφόρμα πρέπει να παρέχει ευκολίες στην αποστολή των δεδομένων σε κάποιο εξυπηρετητή προκειμένου να τύχουν επεξεργασίας. Το τμήμα της πλατφόρμας που αναλαμβάνει την αποστολή των δεδομένων πρέπει να λαμβάνει υπόψη του κατά πόσο υπάρχει διαθέσιμο Wi-Fi δίκτυο όπως επίσης και την προτεραιότητα των δεδομένων που έχει επιλέξει ο σχεδιαστής της εφαρμογής προκειμένου να αποφασίσει

πότε θα σταλούν τα δεδομένα. Στόχοι είναι να μην επιβαρύνουμε τον χρήστη με επιπλέον κόστος.

Η πλατφόρμα ενδιάμεσου λογισμικού πρέπει να παρέχει μία αφαιρετική αναπαράσταση των δεδομένων για να μπορεί να χρησιμοποιηθεί για μεγάλο εύρος εφαρμογών. Επιπλέον πρέπει να παρέχεται αφαιρετική αναπαράσταση των αισθητήρων ούτως ώστε η πλατφόρμα να είναι εύκολα επεκτάσιμη με αισθητήρες τους οποίους θα δημιουργήσουν οι προγραμματιστές των εφαρμογών για να καλύψουν τις ιδιαίτερες τους ανάγκες τους. Τέλος, η πλατφόρμα πρέπει να είναι σχεδιασμένη με τρόπο που να ευκολύνει την ανάπτυξη εφαρμογής και από άτομα χωρίς ιδιαίτερη προγραμματιστική εμπειρία.

1.5 Περίγραμμα αναφοράς

Στο επόμενο κεφάλαιο της αναφοράς μας θα περιγράψουμε ανταγωνιστικές πλατφόρμες που έχουν μελετηθεί. Θα αναφερθούμε στα ανοικτά ζητήματα που απευθύνεται η κάθε πλατφόρμα και τον τρόπο με τον οποίο τα προσεγγίζει. Στην συνέχεια ακολουθεί παρουσίαση της πλατφόρμας Feel The World, αναλυτική περιγραφή της αρχιτεκτονικής της και των συστατικών της μελών. Στο τέταρτο κεφάλαιο αναφερόμαστε στον τρόπο υλοποίησης της πλατφόρμας καθώς και στους κύριους στόχους που είχαμε θέσει. Ιδιαίτερη έμφαση δίνεται στην αφαιρετικότητα στα δεδομένα και στους αισθητήρες. Στο πέμπτο κεφάλαιο γίνεται αξιολόγηση της πλατφόρμας. Η αξιολόγηση αφορά την εκφραστικότητα που παρέχει η πλατφόρμα και τις δυνατότητες στην ανάπτυξη εφαρμογών μέσω μίας demo εφαρμογής. Στο τελευταίο κεφάλαιο παρουσιάζουμε τα συμπεράσματα που προέκυψαν από την έρευνα καθώς και τις προτάσεις μας για ανάπτυξη και βελτίωση της πλατφόρμας.

Κεφάλαιο 2

Σχετική Βιβλιογραφία

-
- 2.1 Εισαγωγή.....**Error! Bookmark not defined.**
 - 2.2 The Jigsaw Continuous Sensing Engine for Mobile Phone Applications**Error! Bookmark not defined.**
 - 2.3 MyExperience: A system for in situ tracing and capturing of user feedback on mobile phones.....**Error! Bookmark not defined.**
 - 2.4 SensLoc: Everyday Places and Paths using Less Energy**Error! Bookmark not defined.**
 - 2.5 SociableSense: Exploring the Trade-offs of Adaptive Sampling and Computation Offloading for Social Sensing**Error! Bookmark not defined.**
 - 2.6 Funf: Open Sensing Framework.....**Error! Bookmark not defined.**
 - 2.7 Feel the world: A Social Mobile Framework for People-centric Sensing **Error! Bookmark not defined.**
 - 2.8 Σύγκριση συστημάτων**Error! Bookmark not defined.**
-

Στο κεφάλαιο αυτό παρουσιάζεται η σχετική βιβλιογραφία που ερευνήθηκε στα πλαίσια της διπλωματικής εργασίας. Παρουσιάζονται συστήματα τα οποία αφορούν το πρόβλημα της αστικής αντίληψης καθώς και ο τρόπος με τον οποίο προσεγγίζει το πρόβλημα το κάθε σύστημα.

2.1 Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζεται η σχετική βιβλιογραφία που ερευνήθηκε στα πλαίσια της διπλωματικής εργασίας. Παρουσιάζονται συστήματα τα οποία αφορούν το πρόβλημα της αστικής αντίληψης καθώς και ο τρόπος με τον οποίο προσεγγίζει το πρόβλημα το κάθε σύστημα.

Σε αυτό το κεφάλαιο θα παρουσιαστεί η έρευνα που έχει γίνει στον τομέα των συστημάτων αστικής αντίληψης καθώς και στις διάφορες τεχνολογίες που τα

υποστηρίζουν. Συγκεκριμένα θα παρουσιάσουμε τα συστήματα και το πρόβλημα στο οποίο απευθύνονται. Επιπλέον θα αναφερθούμε στις καινοτομίες του κάθε συστήματος, τα πλεονεκτήματα και μειονεκτήματά του. Στο τέλος αυτού του κεφαλαίου θα παρουσιάσουμε ένα συνοπτικό πίνακα με όλα τα συστήματα και τα ιδιαίτερα χαρακτηριστικά τους.

2.2 The Jigsaw Continuous Sensing Engine for Mobile Phone Applications

2.2.1 Πρόβλημα

Η πλατφόρμα Jigsaw [4] απευθύνεται στο πρόβλημα της συνεχούς συλλογής δεδομένων για δημιουργία εφαρμογών που χρειάζονται συνεχή ενημέρωση για τις δραστηριότητες των χρηστών (continuous sensing). Αυτό συνεπάγεται πως όλη η προσπάθεια στην συγκεκριμένη πλατφόρμα ήταν να μειωθεί η επιβάρυνση στους πόρους της κινητής συσκευής είτε αυτό αφορά ενεργειακή κατανάλωση, είτε επιβάρυνση στον επεξεργαστή.

Πιο συγκεκριμένα απευθύνεται στα ακόλουθα ανοικτά ζητήματα

- Αντλληψη περιβάλλοντος του χρήστη
- Ισοζύγιο μεταξύ ακρίβειας και απαιτήσεις σε πόρους για συνεχή συλλογή δεδομένων
- Παροχή μιας επαναχρησιμοποιήσιμης πλατφόρμας που μπορεί να χρησιμοποιηθεί για μεγάλο αριθμό εφαρμογών

2.2.2 Αρχιτεκτονική

Η αρχιτεκτονική του συστήματος χωρίζεται σε τρία βασικά υποσυστήματα (classification pipelines):

- Accelerometer pipeline
- Microphone pipeline
- GPS pipeline

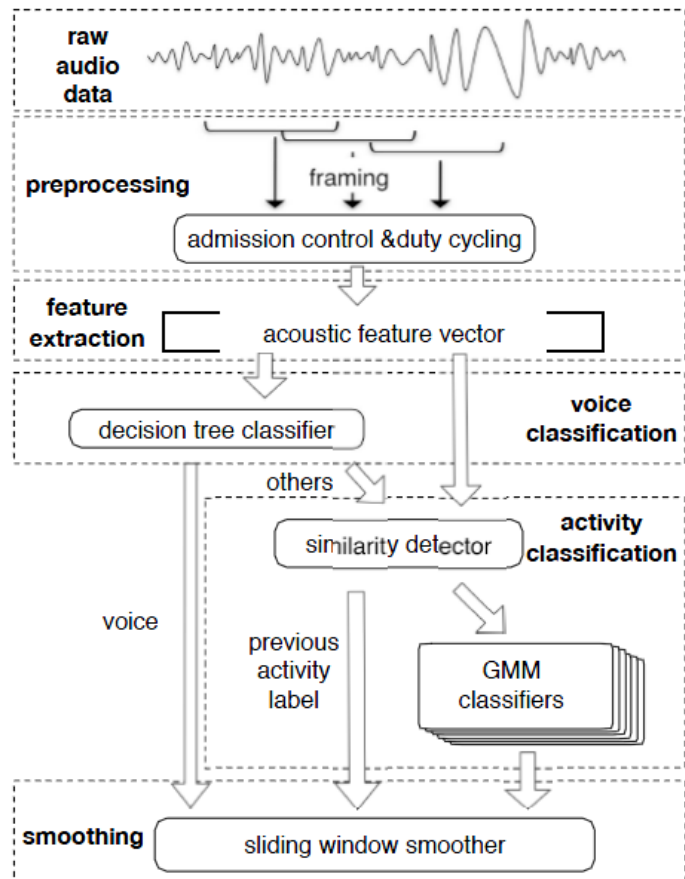
Accelerometer Pipeline

Το ενεργειακό κόστος της χρήσης του accelerometer δεν είναι απαγορευτικό έτσι η προσοχή δόθηκε στην ευρωστία της ανίχνευσης. Δηλαδή στο κατά πόσο η δραστηριότητα η οποία ανιχνεύεται είναι η ορθή.

Microphone Pipeline

Το χαρακτηριστικό της χρήσης του μικροφώνου είναι πως επιβαρύνει ιδιαίτερα την υπολογιστική μονάδα κατά την διάρκεια της ανάλυσης των αποτελεσμάτων. Για την αντιμετώπιση αυτού του προβλήματος ακολουθήθηκαν δύο τεχνικές:

- Χωρίζουν τους ήχους σε εύκολα αναγνωρίσιμους και όχι εύκολα αναγνωρίσιμους περνώντας τους από ένα decision tree classifier, όπως φαίνεται στο σχήμα. Αν ο ήχος είναι εύκολα αναγνωρίσιμος τότε δεν τυγχάνει ιδιαίτερης επεξεργασίας.
- Απαλύφουν γειτονικά πλαίσια ήχου (frames) που στην ουσία αποτελούν πλεονασμό, με σκοπό να μειώσουν το κόστος επεξεργασίας



Jigsaw Microphone Pipeline

Εικόνα 3: Microphone Pipeline - Jigsaw

GPS Pipeline

Το χαρακτηριστικό της χρήσης του GPS δέκτη είναι πως η χρήση του έχει μεγάλη επίπτωση στην μπαταρία. Όσο μεγαλύτερος είναι ο ρυθμός λήψης δεδομένων, τόσο μεγαλύτερο είναι και το ενεργειακό κόστος. Η προσέγγιση που ακολούθησαν στην συγκεκριμένη πλατφόρμα είναι ο ρυθμός λήψης δεδομένων να αλλάζει δυναμικά αναλόγως της ταχύτητας του χρήστη. Με αυτό τον τρόπο υπάρχει αρκετά υψηλή ακρίβεια στα δεδομένα που συλλέγονται, ενώ ταυτόχρονα μειώνεται το ενεργειακό κόστος.

2.3 MyExperience: A system for in situ tracing and capturing of user feedback on mobile phones

2.3.1 Πρόβλημα

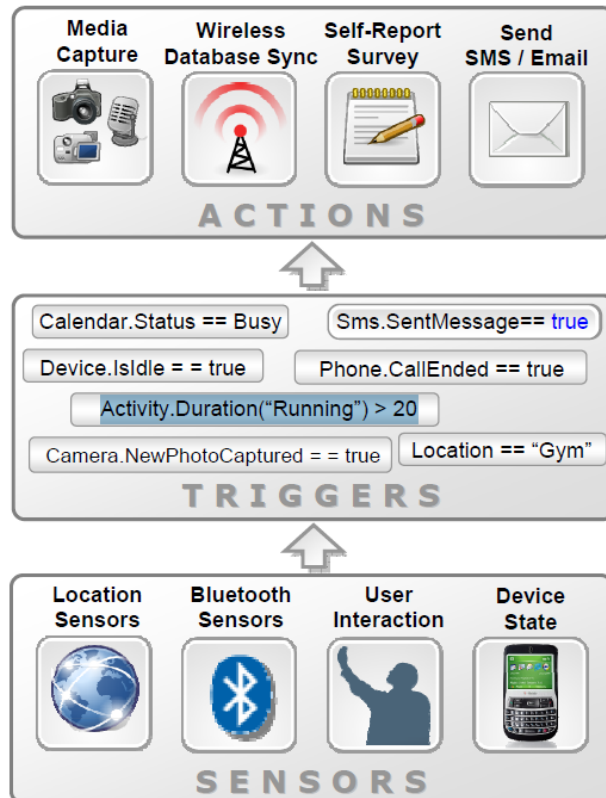
Είναι πλατφόρμα [8] η οποία χρησιμοποιείται για συλλογή τόσο αντικειμενικών δεδομένων (πραγματικές μετρήσεις από αισθητήρες) αλλά και υποκειμενικών (π.χ. γνώμη του χρήστη για ένα συμβάν). Ανώτερος στόχος της συγκεκριμένης πλατφόρμας είναι να αντληφθούν όχι μόνο τις ενέργειες των χρηστών αλλά και τον λόγο ορισμένων ενεργειών. Τα ανοικτά προβλήματα στα οποία απευθύνεται η συγκεκριμένη πλατφόρμα συνοψίζονται πιο κάτω:

- Πλατφόρμα ανοικτού λογισμικού για συλλογή ποσοτικών και ποιοτικών δεδομένων που αφορούν τις ενέργειες των χρηστών.
- Ευκολία στην χρήση της πλατφόρμας και από άτομα που δεν έχουν εμπειρία προγραμματισμού χρησιμοποιώντας XML-based αρχείο ρυθμίσεων.

2.3.2 Αρχιτεκτονική

Η αρχιτεκτονική του συστήματος αποτελείται από τρία κύρια μέρη:

- Αισθητήρες (Sensors)
- Σκανδάλες (Triggers)
- Ενέργειες (Actions)



Εικόνα 4: MyExperience Architecture

Οι αισθητήρες συλλέγουν συνεχώς δεδομένα. Όταν κάποιοι προκαθορισμένοι, από τον σχεδιαστή της εφαρμογής, κανόνες ικανοποιηθούν τότε πυροδοτείται μία σκανδάλη η οποία ενημερώνει τον χρήστη για την συμπλήρωση μίας έρευνας. Με αυτό τον τρόπο ο ερευνητής γνωρίζοντας από τα δεδομένα των αισθητήρων την ενέργεια του χρήστη, ζητώντας του να συμπληρώσει το ερωτηματολόγιο μπορεί να συλλέξει ποιοτικά δεδομένα. Τα δεδομένα μαζί με το συμπληρωμένο ερωτηματολόγιο αποστέλλονται ανώνυμα μέσω HTTPS σύνδεσης στον server. Σημαντικό χαρακτηριστικό της πλατφόρμας είναι πως χρησιμοποιείται ευκαιριακός συγχρονισμός με τον server (opportunistic synchronization) όταν υπάρχει Wi-Fi έτσι ώστε να μην επιβαρύνεται ο χρήστης.

```
<!--Activate phone call & cell related sensors-->
<sensor name="PhoneCall" type="PhoneCallSensor"/>
<sensor name="CellStrength"
  type="CellSignalStrengthSensor"/>
<sensor name="CellId" type="CellIdSensor"/>
```

Defining sensors in XML

Εικόνα 5: Xml Configuration file - MyExperience

2.4 SensLoc: Everyday Places and Paths using Less Energy

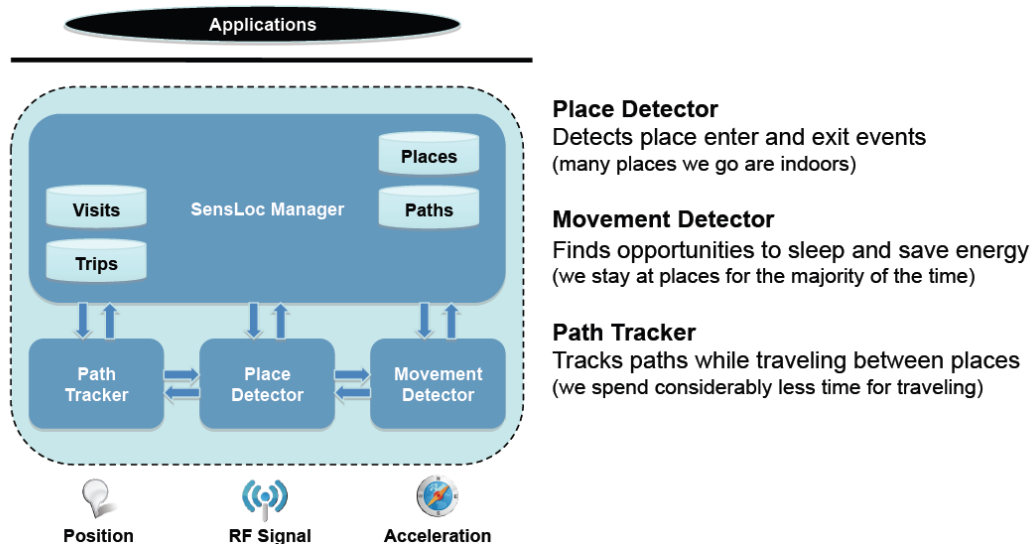
2.4.1 Πρόβλημα

Το SensLoc [2] είναι ένα σύστημα το οποίο περιλαμβάνει ένα εύρωστο αλγόριθμο αναγνώρισης τοποθεσίας, ένα ευαίσθητο ανιχνευτή κίνησης και επίσης μπορεί να σχηματίσει το μονοπάτι που ακολουθήθηκε από τον χρήστη. Ο βασικότερος στόχος του είναι να αντιλαμβάνεται τον χώρο στον οποίο βρίσκεται ο χρήστης με αποδοτικό τρόπο, χρησιμοποιώντας λιγότερη ενέργεια.

2.4.2 Αρχιτεκτονική

Η αρχιτεκτονική του συστήματος αποτελείται από τρία βασικά μέρη:

- Ανιχνευτής χώρου-Wi-Fi (place detector)
- Ανιχνευτής κίνησης-Accelerometer (movement detector)
- Ανιχνευτής μονοπατιού-GPS (path tracker)



Εικόνα 6: SensLoc Architecture

Τα τρία αυτά μέρη έχουν ως απώτερο στόχο την αντίληψη του χώρου στον οποίο βρίσκεται ο χρήστης με αποδοτικό τρόπο. Για να το πετύχουν αυτό επικοινωνούν με τον εξής τρόπο:

- Όταν ο χρήστης κινείται με χαμηλή ταχύτητα – λογικά περπατά – ενεργοποιείται ο ανιχνευτής χώρου ο οποίος χρησιμοποιεί Wi-Fi αισθητήρα και είναι χαμηλής ενεργειακής κατανάλωσης.
- Όταν ο ανιχνευτής χώρου εντοπίσει είσοδο σε κάποιο κτίριο –από σήματα Wi-Fi- τότε ενεργοποιεί τον ανιχνευτή κίνησης και απενεργοποιεί τον ανιχνευτή μονοπατιού.
- Όταν ο ανιχνευτής κίνησης ανιχνεύσει ακινησία τότε απενεργοποιεί τον ανιχνευτή χώρου.

2.5 SociableSense: Exploring the Trade-offs of Adaptive Sampling and Computation Offloading for Social Sensing

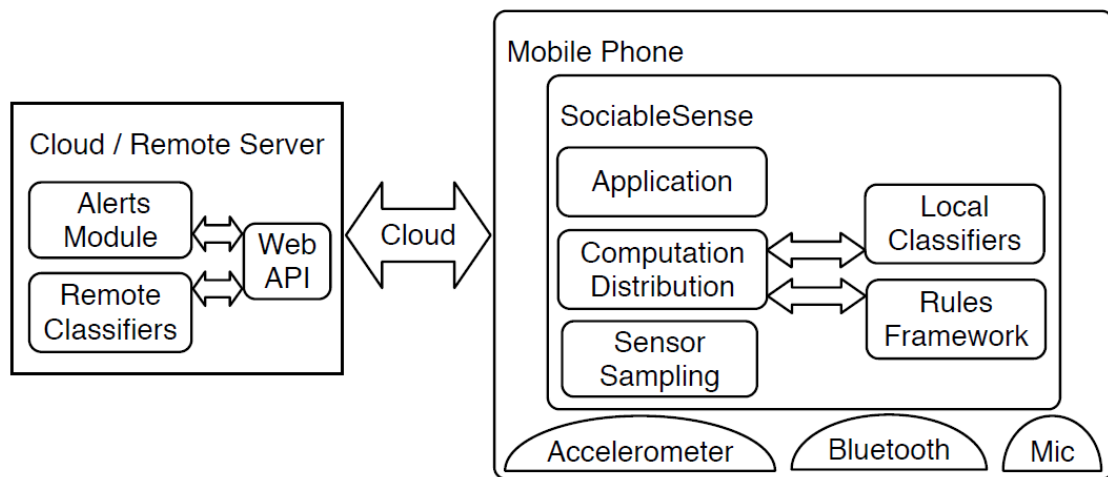
2.5.1 Πρόβλημα

Το sociableSense [1] είναι μία πλατφόρμα η οποία έχει ως στόχο την συλλογή δεδομένων από το εργασιακό περιβάλλον των χρηστών. Πιο συγκεκριμένα προτείνει λύσεις στα πιο κάτω προβλήματα:

- Ενεργειακή κατανάλωση (Energy Consumption)
- Προσαρμοστικός ρυθμός συλλογής δεδομένων (Adaptive Sampling)
- Απόφαση κατά πόσο η επεξεργασία θα γίνει τοπικά (στην συσκευή) είτε σε απομακρυσμένο εξυπηρετητή (server) (Computation Scheme)

2.5.2 Αρχιτεκτονική

Η αρχιτεκτονική του SociableSense αποτελείται από δύο κύρια μέρη. Υπάρχει το κομμάτι που αφορά τον εξυπηρετητή (server) και το κομμάτι που αφορά την κινητή συσκευή.



Εικόνα 7: SociableSense architecture

Θα αναφερθούμε στην κομμάτι της κινητής συσκευής αφού αυτό συγκεντρώνει το μεγαλύτερο ενδιαφέρον. Ο ρυθμός συλλογής δεδομένων καθορίζεται ως εξής. Τα δεδομένα συλλέγονται από τους αισθητήρες και προωθούνται στον ανιχνευτή δραστηριότητας (activity recognition classifier). Αν από τα δεδομένα ανιχνεύεται μία δραστηριότητα, τότε αυτό θεωρείται ως ένα σημαντικό γεγονός (unmissable event), διαφορετικά θεωρούμε πως τα δεδομένα δεν είχαν πρακτική σημασία (missable event). Στην περίπτωση όπου το γεγονός ήταν σημαντικό ο ρυθμός συλλογής δεδομένων αυξάνεται με τον ακόλουθο τρόπο:

- success - unmissable event

$p_i = p_i + \alpha(1 - p_i)$, where $0 < \alpha < 1$, i = accelerometer, Bluetooth, microphone

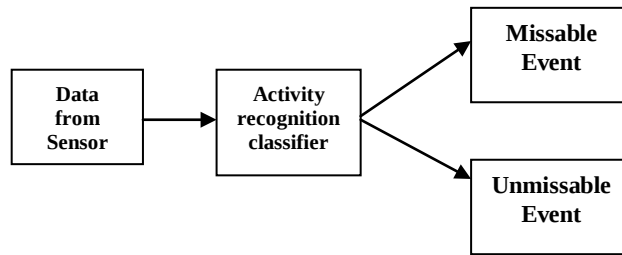
α_i = be the sensing action on a sensor s_i

$0.1 < p_i < 0.9$

Διαφορετικά μειώνεται:

- failure - missable event

$p_i = p_i - \alpha p_i$



Εικόνα 8: SociableSense dynamic monitoring

Η λογική πίσω από αυτή την πρακτική είναι πως αν όταν ανιχνευθεί μία δραστηριότητα, τότε πολύ πιθανόν ο χρήστης βρίσκεται σε κίνηση όποτε ακολουθούν επιπλέον σημαντικές δραστηριότητες που πρέπει να ανιχνευθούν. Διαφορετικά, ο χρήστης βρίσκεται σε κάποια αδρανή κατάσταση και πολύ πιθανόν να παραμείνει σε αυτή την κατάσταση για κάποιο χρονικό διάστημα.

Παρατηρούμε από την αρχιτεκτονική πως το κομμάτι που ανιχνεύει τις δραστηριότητες βρίσκεται τόσο στον εξυπηρετητή όσο και στην κινητή συσκευή. Αυτό συμβαίνει λόγω του ότι η πλατφόρμα λαμβάνει υπόψη ένα μεγάλο αριθμό παραμέτρων (ενέργεια, καθυστέρηση δικτύου(latency), αριθμό δεδομένων που θα αποσταλούν και κατά πόσο υπάρχει διαθέσιμο δίκτυο), αποφασίζει κατά πόσο η επεξεργασία των δεδομένων θα γίνει τοπικά στην συσκευή ή τα δεδομένα θα σταλούν στον εξυπηρετητή για να γίνει εκεί η επεξεργασία.

2.6 Funf: Open Sensing Framework

2.6.1 Πρόβλημα

Το Funf [19] είναι ένα σύστημα ανοικτού κώδικα, επαναχρησιμοποιήσιμο και επεκτάσιμο το οποίο επιτρέπει την συλλογή δεδομένων και την αποστολή τους σε ένα απομακρυσμένο εξυπηρετητή. Βασικότερος στόχος το υ είναι η συνεχής συλλογή δεδομένων από την κινητή συσκευή τόσο από αισθητήρες όσο και από δεδομένα του χρήστη όπως επαφές, μηνύματα κλπ.

2.6.2 Αρχιτεκτονική



Εικόνα 9: Funf platform

Το κύριο χαρακτηριστικό της πλατφόρμας Funf είναι η αφαιρετική αναπαράσταση των αισθητήρων και των δεδομένων. Υποστηρίζονται 32 διαφορετικοί τύποι αίσθησης από 6 διαφορετικές κατηγορίες:

- Τοποθεσία (Positioning data)
- Κοινωνικοποίηση (Social data)
- Κίνηση (Motion data)
- Περιβαλλοντικά (Environmental data)
- Πληροφορίες συσκευής (Device Info)
- Αλληλεπίδραση με συσκευή (Device Interaction)

Οι αισθητήρες μπορούν να ρυθμιστούν ως προς τον ρυθμό συλλογής δεδομένων (PERIOD), την διάρκεια (DURATION), την ώρα έναρξης (START TIME) και την ώρα παύσης τους (END TIME).

Επιπλέον το σύστημα παρέχει λειτουργίες εύκολης αποθήκευσης δεδομένων χρησιμοποιώντας SQLite βάση δεδομένων και επιπλέον ευκολία αποστολής των δεδομένων.

2.7 Σύγκριση συστημάτων

Τα συστήματα που μελετήθηκαν προσεγγίζουν το πρόβλημα της αστικής αίσθησης το κάθε ένα από την δική του σκοπιά. Ακολουθεί συγκριτικός πίνακας μεταξύ των συστημάτων που έχουμε προαναφέρει:

Πίνακας 1

	Jigsaw	MyExperience	SensLoc	SociableSense	Funf	FeelTheWorld
Αισθητήρες Κινητής Συσκευής	Μικρόφωνο Επιταχυνσιόμετρο GPS	GPS Bluetooth	WiFi Επιταχυνσιόμετρο GPS	Μικρόφωνο Επιταχυνσιόμετρο Bluetooth	32	18
Εξωτερικοί Αισθητήρες	OXI	OXI	OXI	OXI	OXI	NAI
Ανίχνευση Δραστηριότητας	NAI	NAI	NAI	NAI	OXI	OXI
Αποστολή δεδομένων μέσω διαθέσιμου δικτύου	OXI	NAI	OXI	NAI	NAI	NAI
Προσαρμογή ρυθμού λήψης δεδομένων	OXI	OXI	OXI	NAI	OXI	NAI
Μονάδα Απόφασης για Αποστολή Δεδομένων μέσω διαδικτύου	OXI	OXI	OXI	NAI	OXI	NAI

Παρατηρούμε πως οι πλατφόρμες Jigsaw, MyExperience, SensLoc και SociableSense χρησιμοποιούν περιορισμένο αριθμό αισθητήρων. Αντίθετα το Funf και η πλατφόρμα Feel The World χρησιμοποιούν όλους τους αισθητήρες. Επιπλέον αντίθετα με την πλατφόρμα Feel The World καμία άλλη πλατφόρμα δεν έχει πρόνοια για σύνδεση με εξωτερικούς αισθητήρες.

Οι πλατφόρμες Jigsaw, MyExperience, SensLoc και SociableSense περιέχουν ενσωματωμένους ανιχνευτές δραστηριότητας (classifiers), αφού ένα από τα προβλήματα που απευθύνονται είναι η ανίχνευση της δραστηριότητας του χρήστη με αποδοτικό τρόπο. Αυτό συνεπάγεται πως αυτές οι πλατφόρμες απευθύνονται σε συγκεκριμένες κατηγορίες εφαρμογών. Αντίθετα οι πλατφόρμες Funf και Feel The World παρέχουν μια πιο αφαιρετική δομή και είναι πιο γενικής χρήσης.

Τέλος, παρατηρούμε πως μόνο η πλατφόρμα SociableSense και Feel The World περιέχουν μονάδα απόφασης για αποστολή δεδομένων μέσω διαδικτύου. Μονάδα δηλαδή που απαντά στο ερώτημα «Πότε;» πρέπει να σταλούν τα δεδομένα.

Κεφάλαιο 3

Πλατφόρμα ενδιάμεσου λογισμικού Feel The World

3.1 Πρόβλημα	21
3.2 Τεχνολογίες που χρησιμοποιήθηκαν.....	22
3.3 Αρχιτεκτονική Συστήματος.....	24
3.4 Ανάλυση Αρχιτεκτονικής.....	25
3.3.1 Εξυπηρετητής.....	25
3.3.2 Κινητή συσκευή	27

Στην ενότητα αυτή γίνεται παρουσίαση της πλατφόρμας που έχει αναπτυχθεί στα πλαίσια της διπλωματικής εργασίας. Αρχικά γίνεται αναφορά στο πρόβλημα στο οποίο απευθύνεται και τους τρόπους με τους οποίους προσεγγίζουμε το πρόβλημα. Ακολούθως αναφερόμαστε στις τεχνολογίες που χρησιμοποιήθηκαν. Τέλος, παρουσιάζουμε την αρχιτεκτονική της πλατφόρμας την οποία και αναλύουμε.

3.1 Πρόβλημα

Η πλατφόρμα Feel the World αναφέρεται στο ερευνητικό τομέα της συνεχούς αίσθησης (continuous sensing). Το σημαντικότερο από τα προβλήματα που αντιμετωπίζει ο συγκεκριμένος τομέας είναι οι περιορισμένοι πόροι που παρέχουν οι κινητές συσκευές και ιδιαίτερα η χαμηλή διάρκεια ζωής της μπαταρίας. Μία ικανοποιητική εφαρμογή που συλλέγει δεδομένα πρέπει να μπορεί να συλλέγει δεδομένα καθόλη την διάρκεια της μέρας, χωρίς να έχει αρνητικό αντίκτυπο στην εμπειρία χρήσης του ιδιοκτήτη της συσκευής.

Η πλατφόρμα η οποία προτείνουμε στα πλαίσια αυτής της διπλωματικής διαφοροποιείται σε μεγάλο βαθμό από τις πλατφόρμες που παρουσιάστηκαν στο κεφάλαιο 2 αφού παρέχει ένα αφαιρετικό τρόπο χειρισμού όλων των αισθητήρων και

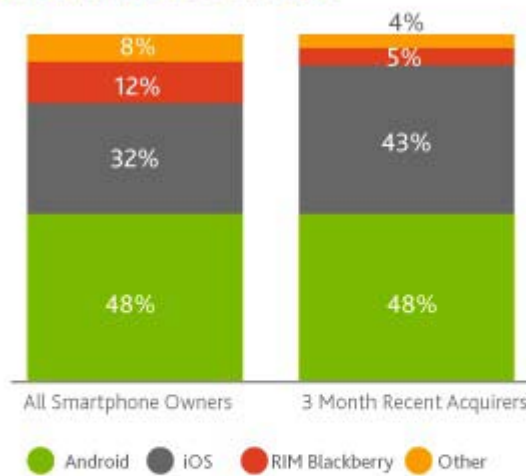
δυνατότητα ενσωμάτωσης εξωτερικών αισθητήρων κάτω από ένα ενιαίο πλαίσιο. Ο σχεδιαστής της εφαρμογής μπορεί να ρυθμίσει την συλλογή δεδομένων ως προς τον ρυθμό συλλογής και την διάρκεια της κάθε συλλογής. Επιπλέον κατά διαστήματα συλλέγονται οι διαθέσιμοι πόροι της συσκευής. Γνωρίζοντας τους διαθέσιμους πόρους παρέχουμε την δυνατότητα της δυναμικής ρύθμισης της συλλογής δεδομένων αναλόγως των πόρων. Η πλατφόρμα αναλαμβάνει την αποστολή των δεδομένων σε κάποιο εξυπηρετητή προκειμένου να τύχουν επεξεργασίας. Το τμήμα της πλατφόρμας που αναλαμβάνει την αποστολή των δεδομένων λαμβάνει υπόψη του κατά πόσο υπάρχει διαθέσιμο Wi-Fi δίκτυο όπως επίσης και την προτεραιότητα των δεδομένων που έχει επιλέξει ο σχεδιαστής της εφαρμογής προκειμένου να αποφασίσει πότε θα σταλούν τα δεδομένα. Τέλος, η πλατφόρμα είναι σχεδιασμένη με τρόπο που να ευκολύνει την ανάπτυξη εφαρμογής και από άτομα χωρίς ιδιαίτερη προγραμματιστική εμπειρία.

3.2 Τεχνολογίες που χρησιμοποιήθηκαν

Για την ανάπτυξη της πλατφόρμας Feel the World επιλέχθηκε το λογισμικό Google Android. Οι εναλλακτικές επιλογές ήταν το iOS της Apple και το Windows Phone 7 της Microsoft. Ο κυριότερος λόγος που επιλέξαμε το λογισμικό Android της Google είναι το γεγονός πως σε σχέση με τα ανταγωνιστικά λογισμικά παρέχει μεγάλη ελευθερία στον προγραμματιστή αλλά και πρόσβαση σε πόρους του κινητού που τα υπόλοιπα λογισμικά δεν επιτρέπουν. Για παράδειγμα το λογισμικό Windows Phone 7 δεν δίνει την δυνατότητα υλοποίησης διεργασίας που τρέχει στο παρασκήνιο με ρυθμό καθορισμένο από τον προγραμματιστή. Ο περιορισμός αυτός και μόνο είναι απαγορευτικός στην υλοποίηση ενός πλαισίου συνεχούς συλλογής δεδομένων από αισθητήρες. Από την άλλη το iOS δεν επιτρέπει την σύνδεση με τους εξωτερικούς αισθητήρες της Sensaris άρα και αυτή η επιλογή εγκαταλείφθηκε άμεσα. Επιπλέον όμως θετικό της επιλογής του λειτουργικού Android της Google είναι πως το συγκεκριμένο λειτουργικό κατέχει το μεγαλύτερο μερίδιο της αγοράς έχοντας επίσης και τον μεγαλύτερο ρυθμό αύξησης. Αυτό συνεπάγεται πως η πλατφόρμα που υλοποιούμε θα έχει χρήση σε μεγαλύτερο αριθμό ατόμων.

Smartphone Operating System Share

February 2012, Nielsen Mobile Insights



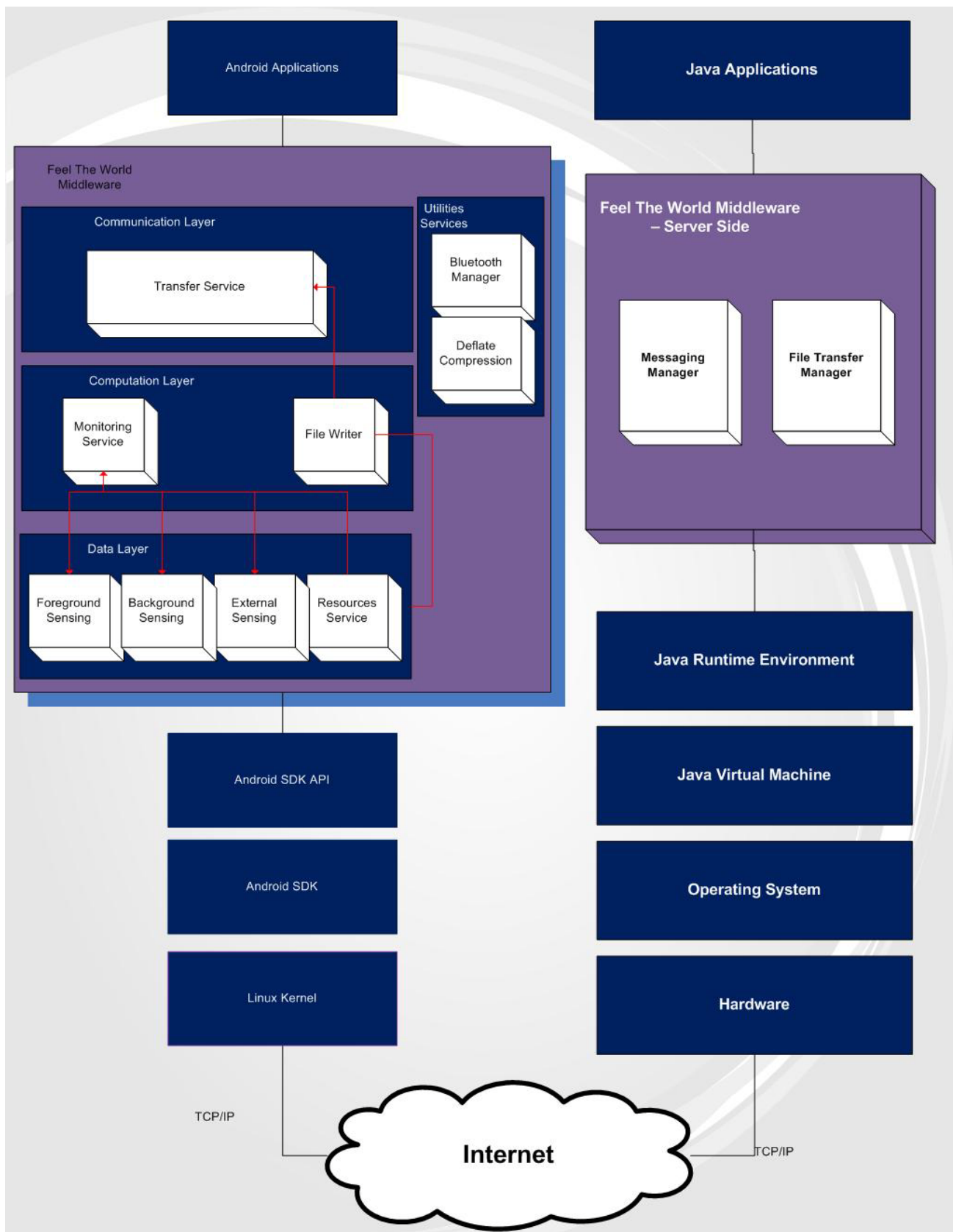
Read As: During February 2012, 48 percent of smartphone owners had a device that runs on the Android operating system

Source: Nielsen

nielsen

Εικόνα 10: Smartphone operating System Share

3.3 Αρχιτεκτονική Συστήματος

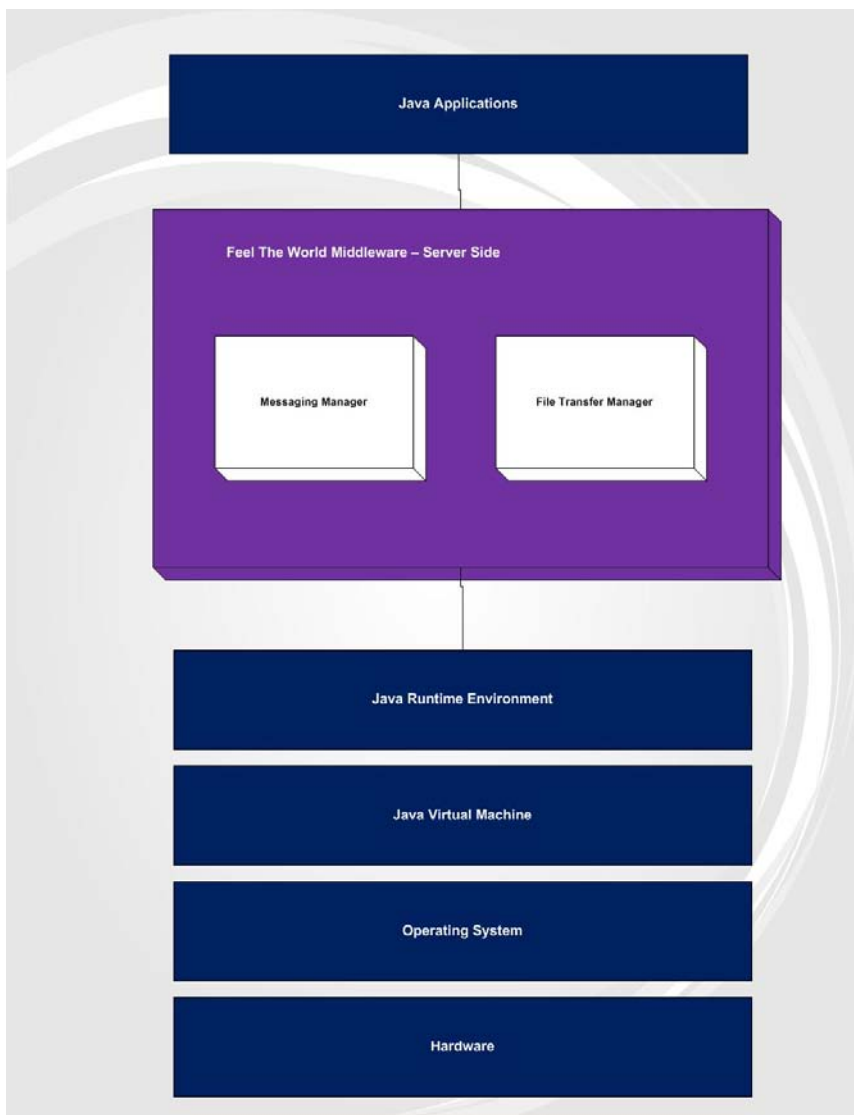


Εικόνα 11: Feel the World architecture

3.4 Ανάλυση Αρχιτεκτονικής

Η αρχιτεκτονική της πλατφόρμας αποτελείται από δύο μέρη. Υπάρχει το μέρος το οποίο αφορά την κινητή συσκευή που τρέχει Android λειτουργικό σύστημα, και το μέρος που αφορά τον εξυπηρετητή.

3.3.1 Εξυπηρετητής

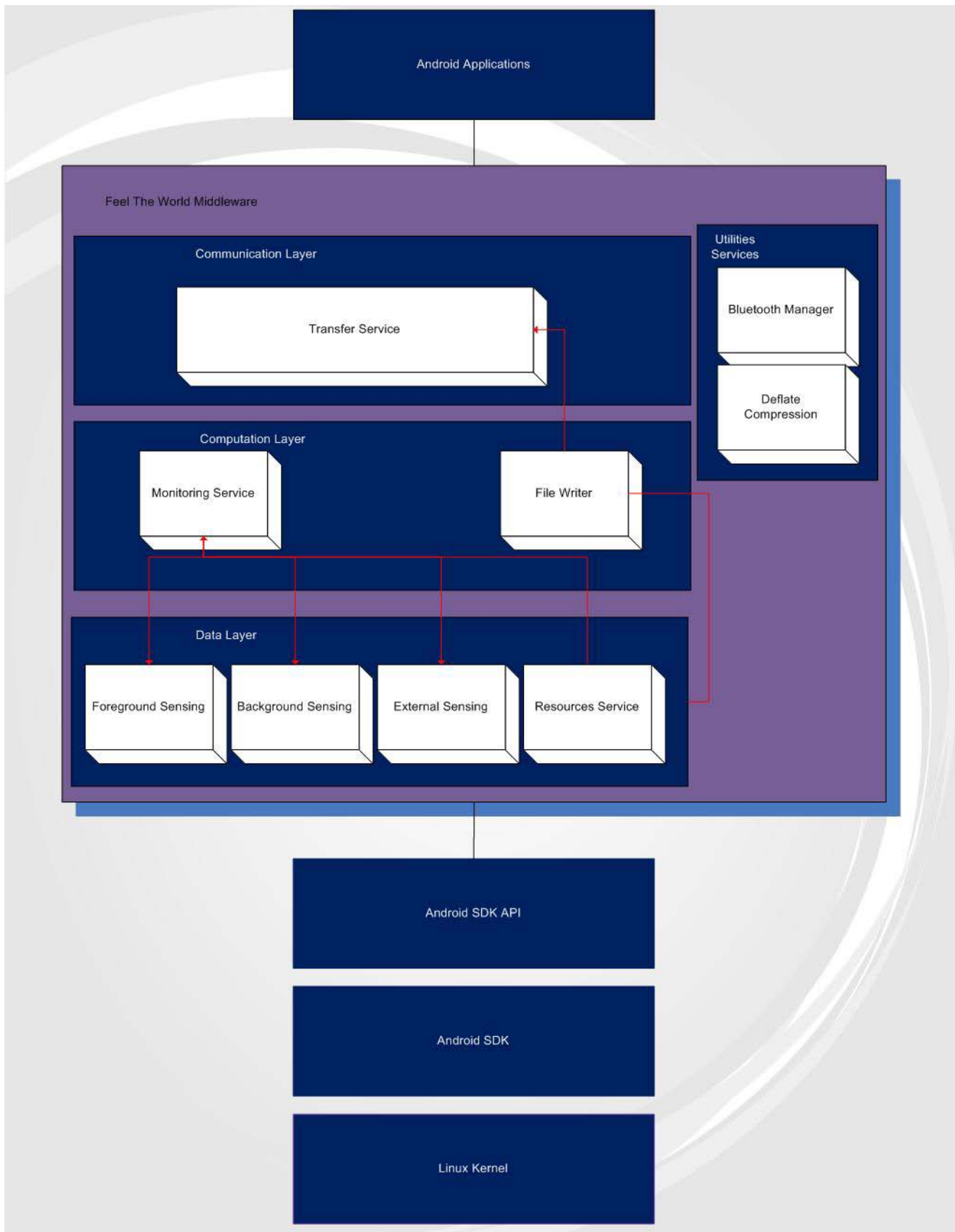


Εικόνα 12: Feel the World architecture – server side

Οι υπηρεσίες που παρέχονται στον εξυπηρετητή αναπτύχθηκαν πάνω από το περιβάλλον της Java και την Java Virtual Machine. Οι βιβλιοθήκες της Java που χρησιμοποιούνται, παρέχονται δωρεάν, και προσφέρουν πολλές και ποικίλες κλάσεις στους προγραμματιστές.

Στον εξυπηρετητή υπάρχουν οι μονάδες (messaging manager, file transfer manager) που είναι υπεύθυνες για την λήψη των δεδομένων από τις κινητές συσκευές και την φύλαξή τους στον εξυπηρετητή για ανάλυση. Οι μονάδες αυτές παρέχονται για διευκόλυνση του σχεδιαστή της εφαρμογής αφού εύκολα και με λίγες γραμμές κώδικα μπορεί να υλοποιήσει ένα πλήρως λειτουργικό εξυπηρετητή.

3.3.2 Κινητή συσκευή



Εικόνα 13: Feel the World architecture – client side

Στην κινητή συσκευή συγκεντρώνεται τον ενδιαφέρον μέρος της πλατφόρμας. Παρατηρούμε αρχικά πως η αρχιτεκτονική χωρίζεται σε 5 διαφορετικά υποσυστήματα:

- ❖ Data Layer
- ❖ Monitoring Service
- ❖ File Writer
- ❖ Communication Layer
- ❖ Utilities Services

3.3.2.1 Data Layer

Το συγκεκριμένο επίπεδο είναι αυτό που αλληλεπιδρά άμεσα με το Android SDK προκειμένου να συλλέξει τα απαραίτητα δεδομένα. Εδώ υλοποιούνται οι αφαιρετικές δομές των αισθητήρων που είναι υπεύθυνες για την συλλογή δεδομένων. Οι αισθητήρες χωρίζονται σε τρεις κατηγορίες:

- ❖ Background Sensors
- ❖ Foreground Sensors
- ❖ External Sensors

Η κατηγορία των **Background Sensors** περιλαμβάνει τους ενσωματωμένους αισθητήρες της συσκευής και παρέχει την δυνατότητα για υλοποίηση μιας υπηρεσίας που τρέχει στο παρασκήνιο και συλλέγει δεδομένα από τους αισθητήρες της συσκευής. Η συλλογή δεδομένων θα συνεχίσει και μετά το κλείσιμο της εφαρμογής

Η κατηγορία των **Foreground Sensors** αφορά και αυτή τους ενσωματωμένους αισθητήρες της συσκευής. Διαφέρει από την κατηγορία των Background Sensors στο γεγονός πως αφορά την συλλογή δεδομένων στο προσκήνιο. Δεν υλοποιείται σαν υπηρεσία παρασκήνιου και κλείνοντας την εφαρμογή σταματά και η λήψη των δεδομένων. Η υλοποίηση αυτή έχει λιγότερο κόστος από την υλοποίηση των background sensors.

Η κατηγορία των **External Sensors** αφορά τους εξωτερικούς αισθητήρες οι οποίοι μπορεί να είναι ενωμένοι στην κινητή συσκευή. Η δομή αυτή παρέχει ένα εύκολο τρόπο συλλογής δεδομένων από εξωτερικούς αισθητήρες και ενσωμάτωσής τους στο πλαίσιο της πλατφόρμας.

Η πλατφόρμα μπορεί αυτή την στιγμή να συλλέξει δεδομένα από 17 διαφορετικούς τύπους αισθητήρων που βρίσκονται ενσωματωμένοι στην κινητή συσκευή, εφόσον βέβαια υποστηρίζονται από την κινητή συσκευή και επιπλέον από εξωτερικούς αισθητήρες. Συγκεκριμένα μπορούμε να συλλέξουμε δεδομένα από τους ακόλουθους αισθητήρες:

- ❖ Επιταχυνσιόμετρο
- ❖ Θερμοκρασία περιβάλλοντος
- ❖ Αισθητήρας βαρύτητας
- ❖ Γυροσκόπιο
- ❖ Αισθητήρας φωτός
- ❖ Αισθητήρας γραμμικής επιτάχυνσης
- ❖ Αισθητήρας μαγνητικού πεδίου
- ❖ Αισθητήρας προσανατολισμού
- ❖ Αισθητήρα πίεσης
- ❖ Αισθητήρας προσέγγισης
- ❖ Αισθητήρας υγρασίας
- ❖ Αισθητήρας διανύσματος περιστροφής
- ❖ Αισθητήρας θερμοκρασίας
- ❖ Τοποθεσία – από το δίκτυο
- ❖ Τοποθεσία – από GPS
- ❖ Μικρόφωνο
- ❖ Εξωτερικός αισθητήρας

Ο σχεδιαστής της εφαρμογής μπορεί εύκολα να καθορίσει τον ρυθμό συλλογής δεδομένων, την διάρκεια της κάθε συλλογής (όπου αυτό έχει σημασία) καθώς και την προτεραιότητα των δεδομένων επιλέγοντας από τρεις προκαθορισμένες προτεραιότητες (high, normal, low).

Επιπλέον στο επίπεδο των δεδομένων (data layer) υπάρχει το **Resources Service** το οποίο είναι υπηρεσία που τρέχει στο παρασκήνιο και ανά τακτά χρονικά διαστήματα ενημερώνεται για τους διαθέσιμους πόρους της κινητής συσκευής. Οι πόροι της κινητής συσκευής χωρίζονται σε δύο κατηγορίες. Το α πόρο α που αλλάζουν δυναμικά όπως διαθέσιμη κύρια μνήμη, μπαταρία και αριθμός εφαρμογών που τρέχουν και τους πόρους που δεν μεταβάλλονται όπως την συχνότητα του επεξεργαστή. Συγκεκριμένα οι πληροφορίες που συλλέγονται είναι οι ακόλουθες:

- ❖ Συχνότητα επεξεργαστή
- ❖ Διαθέσιμη μπαταρία
- ❖ Αριθμός υπηρεσιών που τρέχουν αυτή την στιγμή
- ❖ Αριθμός εφαρμογών που τρέχουν αυτή την στιγμή
- ❖ Διαθέσιμη κύρια μνήμη

3.3.2.2 *Monitoring Service*

Η υπηρεσία αυτή είναι υπεύθυνη για τον συντονισμό της συλλογής δεδομένων στο data layer. Υλοποιείται σαν background service και τρέχει συνεχώς στο παρασκήνιο. Ανά τακτά χρονικά διαστήματα ενημερώνεται από το Resources service για τους πόρους της κινητής συσκευής και στέλνει τα κατάλληλα μηνύματα στους αισθητήρες. Χρησιμοποιώντας αυτή την υπηρεσία ο προγραμματιστής μπορεί να θέσει απλούς κανόνες προκειμένου να κάνει την εφαρμογή του πιο φιλική προς τον χρήστη. Για παράδειγμα ένας κανόνας μπορεί να είναι σε περίπτωση που το επίπεδο της μπαταρίας είναι κάτω από 20% τότε να σταματήσει η συλλογή δεδομένων. Αυτό μπορεί να γίνει για να μην σπαταλήσουμε την μπαταρία του χρήστη αλλά να του αφήσουμε κάποιο περιθώριο μέχρι να φορτίσει την κινητή του συσκευή. Στην συνέχεια αν το επίπεδο της μπαταρίας ανέβει τότε μπορεί η συλλογή δεδομένων να επαναρχίσει. Η υπηρεσία Monitoring Service τρέχει με προκαθορισμένο ρυθμό τον οποίο μπορεί να επιλέξει ο προγραμματιστής. Οι αισθητήρες που επιθυμούμε να λαμβάνουν ενημερώσεις «δένονται» (bind) με αυτή την υπηρεσία προκειμένου να λαμβάνουν μηνύματα τα οποία χειρίζονται.

3.3.2.3 *File Writer*

Αποτελεί μια σειρά από βοηθητικές συναρτήσεις τις οποίες χρησιμοποιούν οι συλλογείς δεδομένων για να αποθηκεύσουν τα δεδομένα τους σε αρχεία. Τα δεδομένα φυλάσσονται σε αρχεία txt (όλα εκτός του μικροφώνου), σε μορφή CSV(comma separated values).

Η μονάδα αυτή είναι εύκολα επεκτάσιμη. Στην περίπτωση που ο σχεδιαστής της εφαρμογής επιθυμεί να χρησιμοποιήσει τον δικό του τρόπο αποθήκευσης δεδομένων μπορεί εύκολα να αντικαταστήσει το συγκεκριμένο μέρος της πλατφόρμας.

3.3.2.4 *Communication Layer*

Υλοποιείται σαν υπηρεσία παρασκηνίου και αναλαμβάνει την αποστολή των δεδομένων στον εξυπηρετητή. Για την αποστολή των δεδομένων λαμβάνονται υπόψη διάφοροι παράμετροι. Αρχικά ελέγχουμε κατά πόσο υπάρχει διαθέσιμο Wi-Fi δίκτυο. Ο λόγος είναι πως δεν θέλουμε να στέλνουμε πολλά δεδομένα μέσω 3G δικτύου επειδή συνήθως οι χρήστες δικαιούνται περιορισμένο μέγεθος δεδομένων κάθε μήνα. Αναλόγως λοιπόν της προτεραιότητας των δεδομένων και του διαθέσιμου δικτύου παίρνεται η απόφαση κατά πόσο τα δεδομένα θα σταλούν στον εξυπηρετητή.

Η μονάδα αυτή μπορεί να έχει ακόμα μία χρήση. Γνωρίζοντας τους διαθέσιμους πόρους και κατά πόσο υπάρχει διαθέσιμο δίκτυο, ο προγραμματιστής της εφαρμογής μπορεί να τοποθετήσει την λογική της εφαρμογής του που θα αφορά την αποστολή δεδομένων. Δηλαδή μπορεί να αποφασίζει κατά πόσο θα γίνει η επεξεργασία των δεδομένων τοπικά στην συσκευή ή αν είναι προτιμότερο τα δεδομένα να σταλούν στον εξυπηρετητή και να τύχουν επεξεργασίας εκεί.

3.3.2.5 *Utilities Services*

Η κατηγορία αυτή περιέχει υλοποίηση που θα βοηθήσει τον προγραμματιστή στην ανάπτυξη της εφαρμογής του. Στην μονάδα **Deflate Compression** βρίσκεται υλοποιημένος ο αλγόριθμος συμπίεσης Deflate ο οποίος μπορεί να χρησιμοποιηθεί είτε

κατά την αποθήκευση των δεδομένων στην συσκευή, είτε πριν την αποστολή των δεδομένων για να μειώσουμε το μέγεθος των δεδομένων που αποστέλλονται.

Η μονάδα **Bluetooth Manager** μπορεί να χρησιμοποιηθεί για την ανίχνευση των γειτονικών συσκευών και την παρουσίαση τους. Χρησιμοποιώντας την συγκεκριμένη υλοποίηση μπορούμε πολύ εύκολα με λίγες γραμμές κώδικα να συνδέσουμε την κινητή μας συσκευή με ένα εξωτερικό αισθητήρα και να αρχίσουμε να συλλέγουμε δεδομένα.

3.3.2.6 Αλγόριθμος συμπίεσης

Για την συμπίεση των δεδομένων χρησιμοποιείται ο αλγόριθμος συμπίεσης Deflate. Η συμπίεση δεδομένων μειώνει τον χρόνο διακίνησης των δεδομένων στο δίκτυο αφού μειώνει το μέγεθος τους μειώνοντας αυτόματα το transmission time. Επιπλέον μειώνει το κόστος της αποστολής στις περιπτώσεις όπου τα δεδομένα στέλλονται μέσω των υπηρεσιών 3G και GPRS.

Η συμπίεση έχει ένα μειονέκτημα, το γεγονός πως απαιτείται επεξεργασία τόσο κατά την συμπίεση όσο και κατά την αποσυμπίεση των δεδομένων. Το γεγονός αυτό μας απασχολεί ιδιαίτερα στην πλευρά του κινητού όπου οι πόροι είναι περιορισμένοι και εκτενής χρήση του επεξεργαστή έχει άμεσο αντίκτυπο στην μπαταρία. Σε μερικούς αλγορίθμους συμπίεσης το κόστος επεξεργασίας είναι πολύ μεγάλο σε βαθμό που είναι απαγορευτική η χρήση τους σε κινητές συσκευές.

Η έρευνα για την εύρεση του καταλληλότερου αλγορίθμου έγινε από τον Χάρη Ευσταθιάδη στα πλαίσια της δικής του διπλωματικής εργασίας. Έγιναν δοκιμές κάποιων αλγορίθμων ούτως ώστε να αποφασιστεί ποιος είναι ο επικρατέστερος. Οι δοκιμές έγιναν στους εξής τύπους κωδικοποίησης:

- Αλγόριθμος Huffman[2]
- Συνδυασμός αλγορίθμων Huffman και Run-Length coding
- Συμπίεση ZIP
- Συμπίεση Deflate (συνδυασμός αλγορίθμων LZ[12] και Huffman)

Παρουσιάζονται τα αποτελέσματα συμπίεσης 63 και 504 bytes.

Πίνακας 2

Αλγόριθμος	Μήκος Εισόδου (Bytes)	Μήκος Εξόδου (Bytes)	Ποσοστό Συμπίεσης
Huffman	63	200	-217.46 %
Huffman - Runlength	63	115	-82.53 %
ZIP	63	168	-166.66 %
Deflate	63	50	20.63 %

Πίνακας 3

Αλγόριθμος	Μήκος Εισόδου (Bytes)	Μήκος Εξόδου (Bytes)	Ποσοστό Συμπίεσης
Huffman	504	1600	-217.4 %
Huffman - Runlength	504	899	-78.37 %
ZIP	504	177	64.88 %
Deflate	504	59	88.29 %

Όπως παρατηρούμε από τα πειράματα που έγιναν ο αλγόριθμος Deflate έχει τα καλύτερα αποτελέσματα με συμπίεση κοντά στο 88% όταν τα δεδομένα είναι περίπου 500bytes. Όσον αφορά το πρόβλημα της αύξησης της επεξεργασίας που πρέπει να γίνει στα δύο άκρα, καθώς πριν τη διαδικασία αποστολής πρέπει να γίνει συμπίεση των δεδομένων και αποσυμπίεση τους κατά την παραλαβή, λαμβάνοντας υπόψη μας τα

αποτελέσματα της έρευνας που έγινε στο Σουηδικό Ινστιτούτο Επιστήμης Υπολογιστών ο τύπος συμπίεσης Deflate προσφέρει 67% περισσότερη ταχύτητα καθώς επίσης και 69% εξοικονόμηση ενέργειας στην αποστολή δεδομένων μέσα σε ένα δίκτυο. Για τους λόγους αυτούς ο αλγόριθμος Deflate επιλέχτηκε για συμπίεση των δεδομένων.

Κεφάλαιο 4

Υλοποίηση Συστήματος

4.1 Διάγραμμα κλάσεων.....	36
4.1.1 Resource Manager	36
4.1.2 Data.....	37
4.1.3 Sensing	38
4.2 Λεπτομέρειες υλοποίησης.....	39

Σε αυτό το κεφάλαιο αναφέρουμε τα σημαντικά στοιχεία της υλοποίησης του συστήματος. Αρχικά παρουσιάζεται ένα διάγραμμα κλάσεων του συστήματος και στην συνέχεια αναφερόμαστε στον τρόπο με τον οποίο το σύστημα έχει υλοποιηθεί χρησιμοποιώντας το λογισμικό Android.

4.1 Διάγραμμα κλάσεων



Εικόνα 14: Feel the World – class diagram

4.1.1 Resource Manager

Παρατηρούμε το διάγραμμα κλάσεων και την ιεραρχική δομή της πλατφόρμας. Στο κάτω μέρος του διαγράμματος παρατηρούμε το Resource Manager, το οποίο περιέχει τις πληροφορίες που αφορούν τους πόρους της συσκευής. Οι πληροφορίες που συλλέγονται είναι οι ακόλουθες:

- ❖ Συχνότητα επεξεργαστή
- ❖ Διαθέσιμη μπαταρία
- ❖ Αριθμός υπηρεσιών που τρέχουν αυτή την στιγμή
- ❖ Αριθμός εφαρμογών που τρέχουν αυτή την στιγμή

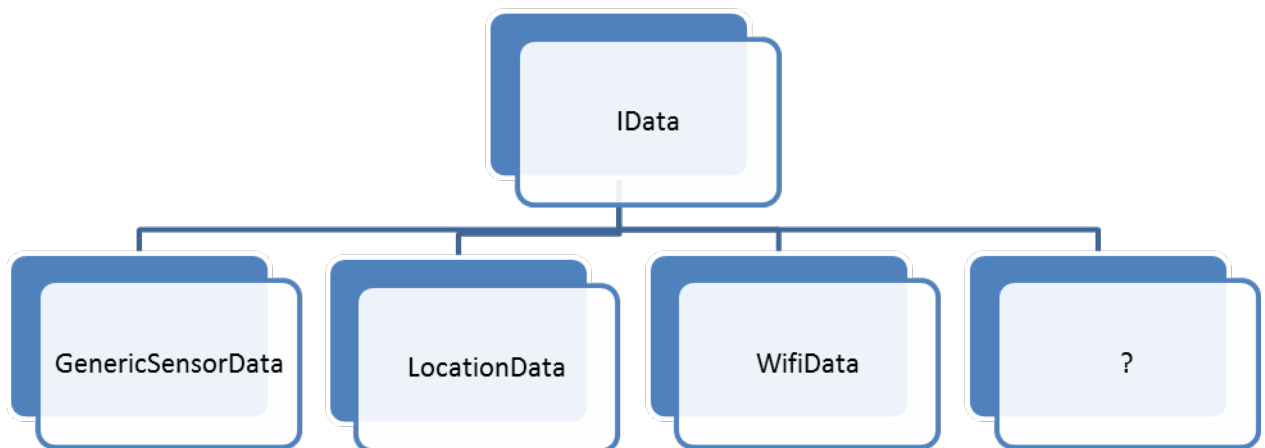
❖ Διαθέσιμη κύρια μνήμη



Εικόνα 15: Feel the World – Data implementation

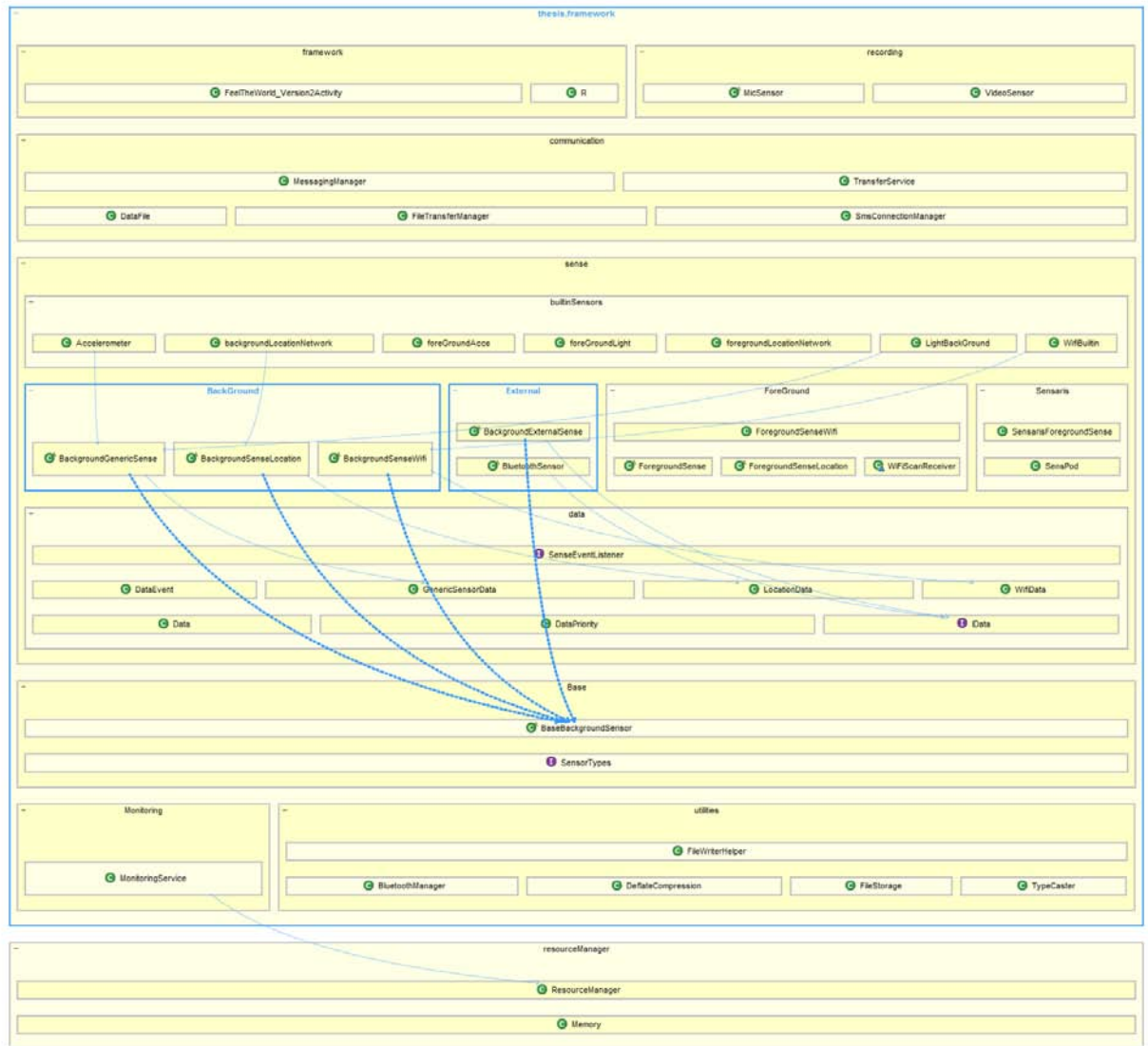
4.1.2 Data

Στο διάγραμμα φαίνονται οι κλάσεις των δεδομένων και οι εξαρτήσεις τους. Κάθε κλάση που αναπαριστά δεδομένα υλοποιεί την διαπροσωπεία IData προκειμένου να υπάρχει ένα ενιαίο πλαίσιο στην αναπαράσταση των δεδομένων και ταυτόχρονα να μπορούμε να εξειδικεύουμε τα δεδομένα αναλόγως του τι συλλέγεται.



Εικόνα 16: Δεδομένα - IData

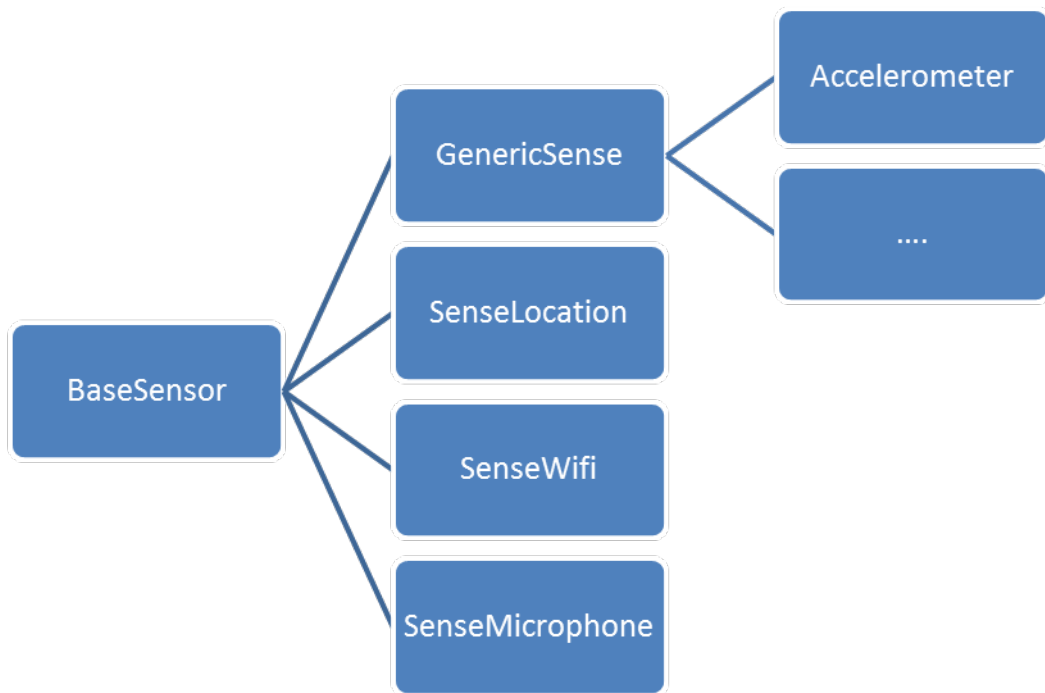
Όπως παρατηρούμε οι τύποι GenericSensorData, LocationData, WifiData υλοποιούν την διαπροσωπεία IData. Αυτό δίνει μία ενιαία αναπαράσταση και όμοιο χειρισμό στα δεδομένα ενώ ταυτόχρονα μπορούμε να εξειδικεύσουμε τον τύπο όπως εμείς επιθυμούμε. Επιπλέον παρέχεται μεγάλη ευκολία επέκτασης της πλατφόρμας.



Εικόνα 17: Feel the World – Base Background extend

4.1.3 Sensing

Στο πακέτο Sense βρίσκονται υλοποιημένοι οι διάφορες κλάσεις αισθητήρων που αφορούν την συλλογή δεδομένων. Ο κάθε αισθητήρας κληρονομεί από την κλάση BaseBackgroundSensor. Η συγκεκριμένη κλάση παρέχει τον επαναχρησιμοποιήσιμο κώδικα και εξειδικευμένες λειτουργίες που αφορούν την αλληλεπίδραση των αισθητήρων με τα υπόλοιπα τμήματα της πλατφόρμας.



Εικόνα 18: BaseSensor

Η κλάση BaseBackgroundSensor παρέχει την απαραίτητη αφαιρετικότητα που απαιτεί μια τέτοια πλατφόρμα. Κληρονομώντας την συγκεκριμένη κλάση μπορούμε να δημιουργήσουμε τους δικούς μας αισθητήρες παρέχοντας την λειτουργικότητα που εμείς θέλουμε να έχουν και ενσωματώνοντας τους έτσι στην υπόλοιπη πλατφόρμα.

4.2 Λεπτομέρειες υλοποίησης

Οι υπηρεσίες της πλατφόρμας (monitoring service, resource manager, sensors) υλοποιήθηκαν κληρονομώντας από την κλάση Service την οποία παρέχει το Android SDK. Κληρονομώντας από την συγκεκριμένη κλάση μας δίνεται η δυνατότητα να παρέχουμε μια λειτουργία η οποία θα τρέχει στο παρασκήνιο ακόμα και όταν η εφαρμογή είναι κλειστή. Η λειτουργικότητα δηλαδή που θέλουμε να έχουμε στις δικές μας υπηρεσίες αφού επιθυμούμε η συλλογή δεδομένων να συνεχίζεται διαρκώς.

Για την επικοινωνία και την ανταλλαγή δεδομένων μεταξύ των υπηρεσιών χρησιμοποιούμε message-based communication χρησιμοποιώντας την κλάση Messenger του Android SDK. Οι υπηρεσίες που συλλέγουν δεδομένα μπορούν να δεθούν (bind) με το Monitoring Service λαμβάνοντας έτσι ρυθμίσεις που αφορούν την

λειτουργία τους. Με την υλοποίηση ενός message-based communication διασφαλίζουμε πως οι διάφορες κλάσεις δεν έχουν απευθείας πρόσβαση σε μεταβλητές άλλων κλάσεων. Επιπλέον είναι πιο εύκολη η επέκταση της πλατφόρμας με επιπλέον κλάσεις που θα δημιουργήσουν οι προγραμματιστές.

Κεφάλαιο 5

Αξιολόγηση

5.1 Εκφραστικότητα πλατφόρμας.....	41
5.2 Εφαρμογή FeelMe.....	42
5.3.1 Γενικά συμπεράσματα	49
5.3 Ρύθμιση για κατανάλωση μπαταρίας	50

Σε αυτό το κεφάλαιο επιχειρούμε μία πρώτη αξιολόγηση της πλατφόρμας. Οι μετρικές που χρησιμοποιούνται είναι τόσο ποιοτικές όσο και ποσοτικές. Μέσα από την αξιολόγηση επιχειρούμε να αναδείξουμε τις δυνατότητες που παρέχονται από την πλατφόρμα.

5.1 Εκφραστικότητα πλατφόρμας

Με τον όρο εκφραστικότητα αναφερόμαστε στην δυνατότητα που παρέχει η πλατφόρμα για χρήση όσο το δυνατόν μεγαλύτερου αριθμού αισθητήρων και την ελευθερία ρύθμισής τους όπως ο χρήστης επιθυμεί. Για να δείξουμε την εκφραστικότητα της πλατφόρμας έχουμε δημιουργήσει ένα μεγάλο αριθμό διαφορετικών αισθητήρων στους οποίους δώσαμε διαφορετικά χαρακτηριστικά. Με αυτό τον τρόπο επιχειρούμε να αναδείξουμε τις τεράστιες δυνατότητες που παρέχει η πλατφόρμα στην ανάπτυξη εφαρμογών αστικής αίσθησης λαμβάνοντας υπόψη τις ιδιαίτερες ανάγκες της κάθε εφαρμογής.

Πίνακας 4

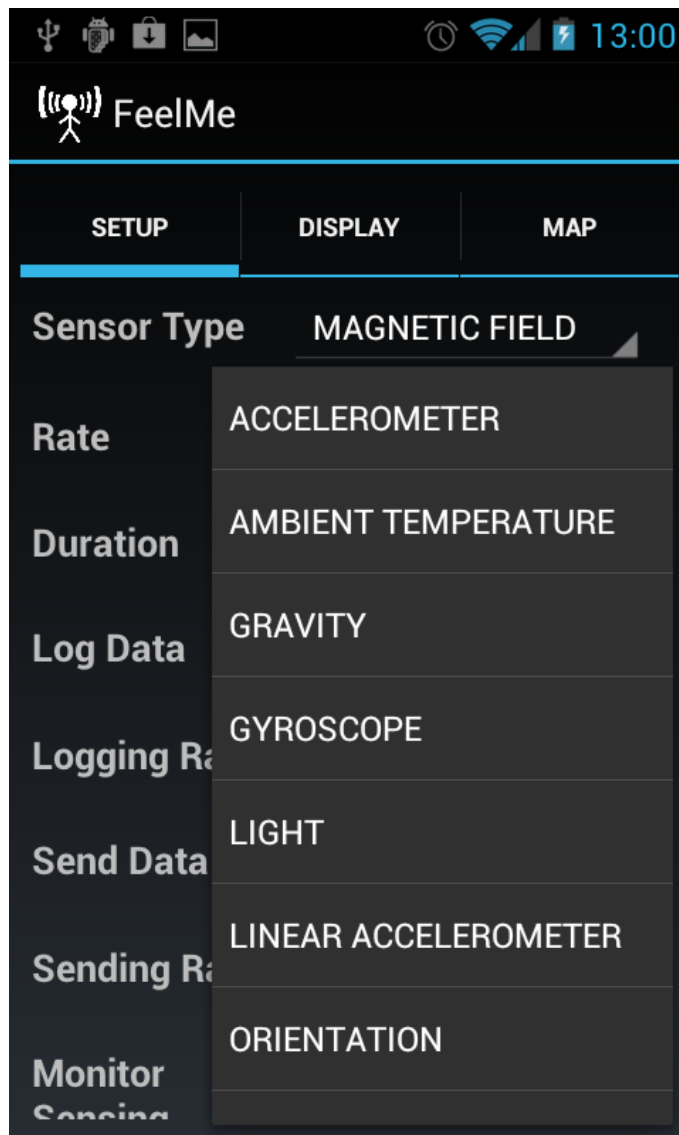
<i>Αισθητήρας</i>	<i>Τρέχει στο παρασκήνιο</i>	<i>Ρυθμός(sec.)</i>	<i>Διάρκεια(sec)</i>	<i>Αυτόματη Ρύθμιση</i>	<i>Καταγραφή σε αρχείο</i>	<i>Αποστολή δεδομένων σε εξυπηρετητή</i>
Επιταχυνσιόμετρο	✓	600	60	✓	✓	✓
Επιταχυνσιόμετρο	✗	100	30	✗	✗	✗
Light Sensor	✓	500	60	✓	✓	✓
Μικρόφωνο	✓	500	60	✗	✓	✓
Wi-Fi	✓	1000	-	✗	✗	✗
Location - GPS	✓	2000	-	✗	✗	✗
Location – Network	✓	2000	-	✓	✗	✗

Παρατηρούμε πως μπορούμε να δημιουργήσουμε μια πληθώρα από διαφορετικούς συλλογείς δεδομένων με διαφορετικά χαρακτηριστικά ο κάθε ένας. Οι μονάδες αυτές τρέχουν παράλληλα μεταξύ τους συλλέγοντας δεδομένα.

5.2 Εφαρμογή FeelMe

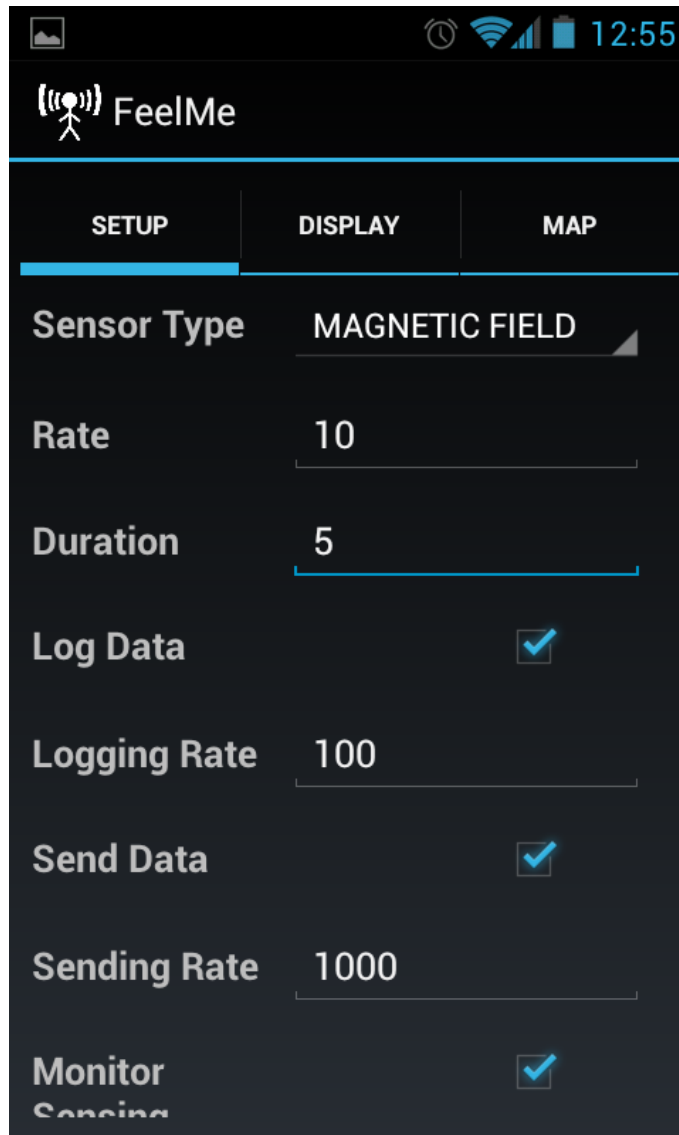
Για την καλύτερη αξιολόγηση της πλατφόρμας προχωρήσαμε στην δημιουργία μίας DEMO εφαρμογής με στόχο να παρουσιάσουμε κάποιες από τις δυνατότητες που παρέχονται από την πλατφόρμα.

Η εφαρμογή FeelMe είναι μια εφαρμογή συμμετέχουσας αντίληψης που έχει ως στόχο την συλλογή δεδομένων από την κινητή συσκευή. Μπορεί να χρησιμοποιηθεί από ερευνητές ή οποιοδήποτε άτομο ενδιαφέρεται για την συλλογή δεδομένων που αφορούν καθημερινές του δραστηριότητες, το περιβάλλον του ή ακόμα την τοποθεσία του.



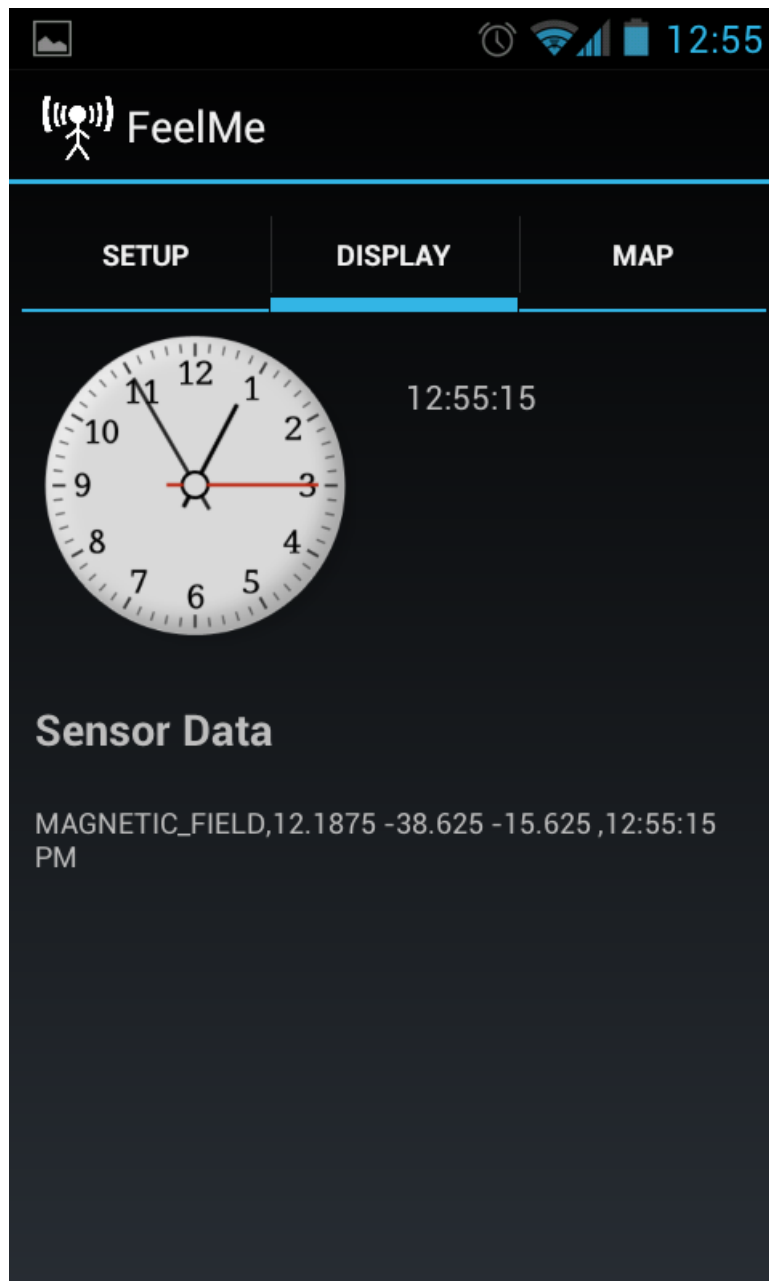
Εικόνα 19: FeelMe Setup screen

Στην πρώτη οθόνη ο χρήστης μπορεί να επιλέξει τον αισθητήρα από τον οποίο επιθυμεί να αντλεί δεδομένα. Οι αισθητήρες αφορούν τους ενσωματωμένους αισθητήρες της συσκευής(ανεξάρτητα αν υποστηρίζονται από την συσκευή).



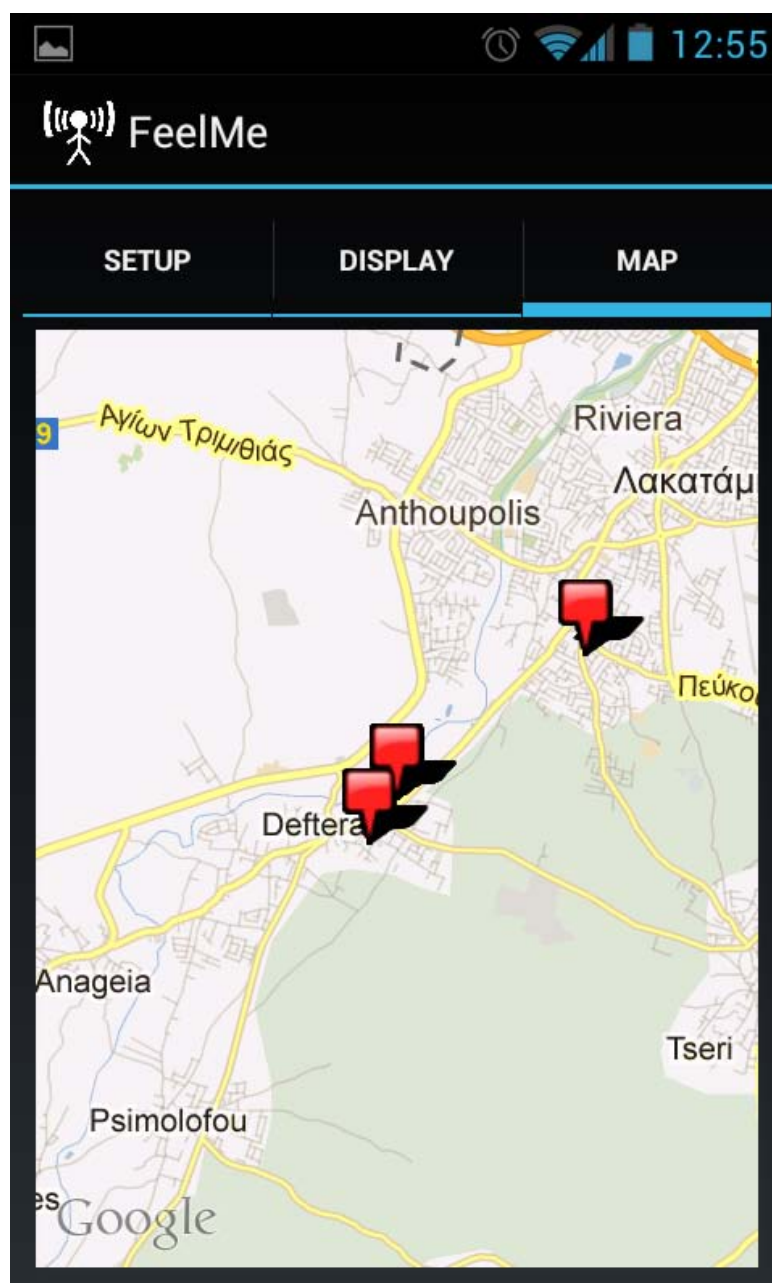
Εικόνα 20: FeelMe Setup screen

Ο χρήστης έχει την δυνατότητα να ρυθμίσει την λήψη δεδομένων ανάλογα με τις επιθυμίες του. Μπορεί να ρυθμίσει τον ρυθμό λήψης δεδομένων, την διάρκεια, κατά πόσο τα δεδομένα θα καταγράφονται σε αρχείο, τον ρυθμό καταγραφής των δεδομένων στο αρχείο, αν τα δεδομένα θα αποστέλλονται σε εξυπηρετητή, τον ρυθμό αποστολής των δεδομένων και τέλος, κατά πόσο ο αισθητήρας θα ρυθμίζεται αυτόματα. Στα πλαίσια της συγκεκριμένης εφαρμογής ο κανόνας αυτόματης ρύθμισης που έχουμε θέσει είναι όταν η μπαταρία πέσει κάτω από 50% να σταματά κάθε λήψη δεδομένων.



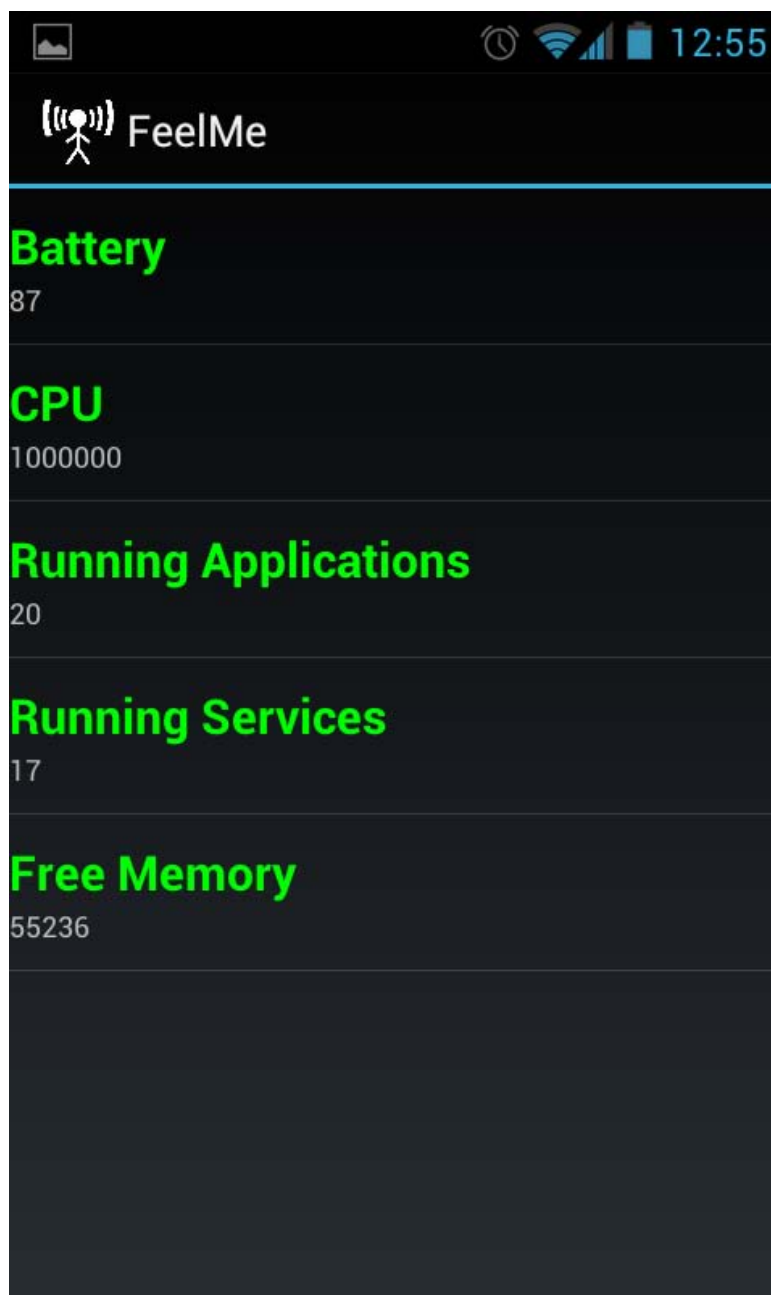
Εικόνα 21: FeelMe Display screen

Στην δεύτερη οθόνη παρουσιάζονται ζωντανά τα δεδομένα που αντλούνται από τον αισθητήρα. Για σκοπούς επιβεβαίωσης πως η λήψη γίνεται αναλόγως του ρυθμού που επέλεξε ο χρήστης εμφανίζουμε ρολόι με δευτερόλεπτα. Επιπλέον με τον τρόπο αυτό προβάλλουμε την δυνατότητα που δίνει η πλατφόρμα για ανάπτυξη εφαρμογών που χειρίζονται και προβάλλουν άμεσα τα δεδομένα.



Εικόνα 22: FeelMe Map screen

Επιπλέον του αισθητήρα που επιλέγει ο χρήστης για να συλλέξει δεδομένα, έχουμε ενεργοποιημένη την λήψη της γεωγραφικής τοποθεσίας του χρήστη χρησιμοποιώντας δεδομένα από το δίκτυο και προβάλλοντάς τα στον χάρτη που φαίνεται πιο πάνω. Ταυτόχρονα τα δεδομένα που συλλέγονται αποθηκεύονται σε αρχείο. Ο συγκεκριμένος αισθητήρας δεν τυγχάνει αυτόματης ρύθμισης έτσι συνεχίζει μέχρι ο χρήστης να τον τερματίσει ρητά.



Εικόνα 23: FeelMe Resources screen

Τέλος, παρουσιάζουμε στην οθόνη Resources τους πόρους της συσκευής, για να προβάλλουμε την δυνατότητα της πλατφόρμας για συνεχή συλλογή αυτών των δεδομένων. Οι πόροι ανανεώνονται κάθε δευτερόλεπτο.

Κατά την ανάπτυξη της εφαρμογής FeelMe ήταν η πρώτη φορά που ήρθαμε σε επαφή με την πλατφόρμα ενδιάμεσου λογισμικού Feel The World για την ανάπτυξη μιας ολοκληρωμένης εφαρμογής, αξιολογώντας την έτσι με τον καλύτερο τρόπο. Αξίζει

να σημειώσουμε πως οι συλλογείς δεδομένων συνεχίζουν την λήψη δεδομένων και αφότου τερματίσουμε την κυρίως εφαρμογή. Επανεκκινώντας την κυρίως εφαρμογή , συνδέεται ξανά με τους συλλογείς δεδομένων και η διαπροσωπεία της εφαρμογής ενημερώνεται κανονικά.

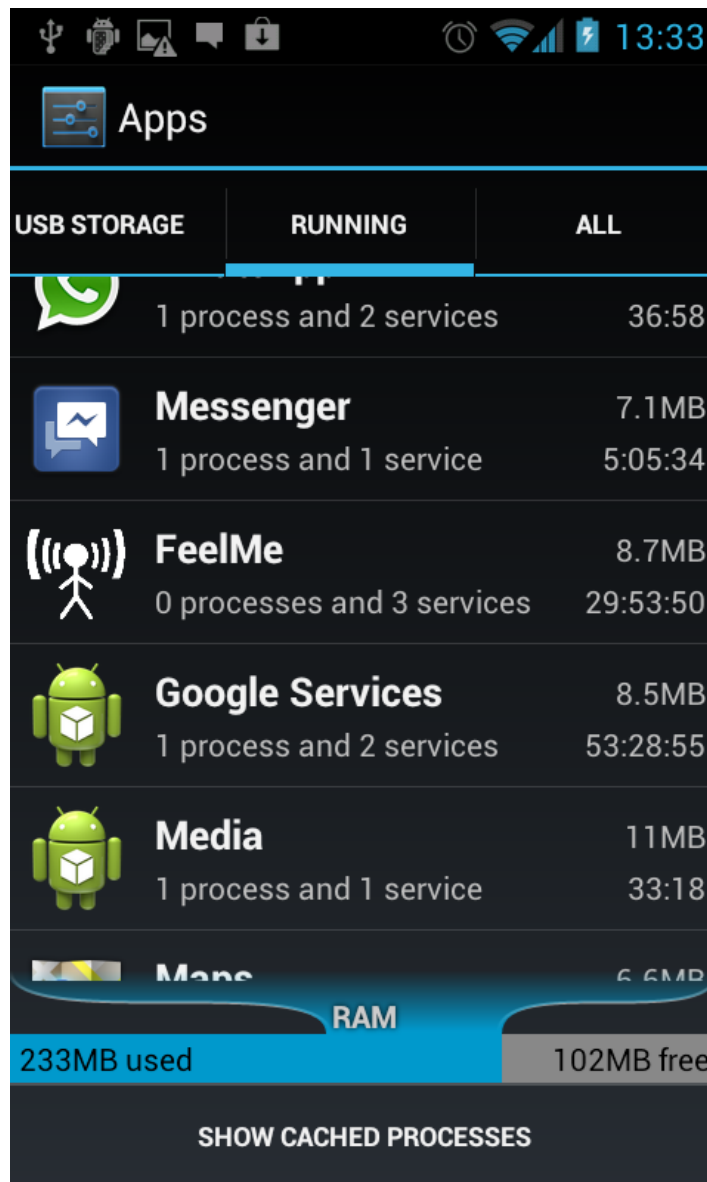
Δεδομένα

ACCELEROMETER,2.585738 8.00621 4.9224787 ,12:09:52 PM
ACCELEROMETER,2.585738 8.00621 4.9224787 ,12:09:53 PM
ACCELEROMETER,2.6048915 7.9487495 4.8841715 ,12:09:53 PM
ACCELEROMETER,2.6048915 7.9487495 4.8841715 ,12:09:53 PM
ACCELEROMETER,2.585738 8.00621 4.9224787 ,12:09:53 PM
ACCELEROMETER,2.585738 8.00621 4.9224787 ,12:09:53 PM
ACCELEROMETER,2.6048915 7.9487495 4.8841715 ,12:09:53 PM
ACCELEROMETER,2.6048915 7.9487495 4.8841715 ,12:09:53 PM
ACCELEROMETER,2.585738 8.00621 4.9224787 ,12:09:53 PM
ACCELEROMETER,2.585738 8.00621 4.9224787 ,12:09:53 PM
ACCELEROMETER,2.6048915 7.9487495 4.8841715 ,12:09:53 PM
ACCELEROMETER,2.6048915 7.9487495 4.8841715 ,12:09:53 PM
ACCELEROMETER,2.585738 8.00621 4.9224787 ,12:09:53 PM
ACCELEROMETER,2.585738 8.00621 4.9224787 ,12:09:54 PM
ACCELEROMETER,2.585738 8.00621 4.9224787 ,12:09:54 PM
ACCELEROMETER,2.585738 8.00621 4.9224787 ,12:09:54 PM
ACCELEROMETER,2.6048915 7.9487495 4.8841715 ,12:09:54 PM
ACCELEROMETER,2.6048915 7.9487495 4.8841715 ,12:09:54 PM
ACCELEROMETER,2.585738 8.00621 4.9224787 ,12:09:54 PM
ACCELEROMETER,2.585738 8.00621 4.9224787 ,12:09:54 PM
ACCELEROMETER,2.6048915 7.9487495 4.8841715 ,12:09:54 PM
ACCELEROMETER,2.6048915 7.9487495 4.8841715 ,12:09:54 PM

LIGHT,400.8624 0.0 0.0 ,9:49:16 AM
LIGHT,0.3310833 -0.1810193 -0.3258915 ,9:49:16 AM
LIGHT,400.8624 0.0 0.0 ,9:49:16 AM
LIGHT,0.3310833 -0.1810193 -0.3258915 ,9:49:16 AM
LIGHT,400.8624 0.0 0.0 ,9:49:16 AM
LIGHT,400.8624 0.0 0.0 ,9:49:16 AM
LIGHT,0.3310833 -0.1810193 -0.3258915 ,9:49:16 AM
LIGHT,400.8624 0.0 0.0 ,9:49:16 AM
LIGHT,400.8624 0.0 0.0 ,9:49:16 AM
LIGHT,400.8624 0.0 0.0 ,9:49:16 AM
LIGHT,0.3310833 -0.1810193 -0.3258915 ,9:49:16 AM
LIGHT,400.8624 0.0 0.0 ,9:49:16 AM
LIGHT,0.3310833 -0.1810193 -0.3258915 ,9:49:16 AM

LOCATION_NETWORK,lat=35.0887299,Long=33.2845346,Altitude=0.0,Accuracy=151.0,1:34:12 PM
LOCATION_NETWORK,lat=35.1077723,Long=33.3042828,Altitude=0.0,Accuracy=1638.0,1:34:17 PM

Πιο πάνω παρουσιάζονται κάποια ενδεικτικά δεδομένα που έχουν συλλεγεί χρησιμοποιώντας την εφαρμογή FeelMe. Παρατηρούμε πως τα δεδομένα αποθηκεύονται σαν CSV(comma separated values) ακολουθώντας το ίδιο μοτίβο. Αρχικά έχουμε τον τύπο του αισθητήρα ακολουθούμενο από τις τιμές και στο τέλος έχουμε την ώρα συλλογής.



Εικόνα 24: FeelMe - robustness

5.3.1 Γενικά συμπεράσματα

Ευρωστία

Μέσα από την συγκεκριμένη εφαρμογή είχαμε την ευκαιρία να αξιολογήσουμε μία σειρά από παραμέτρους της πλατφόρμας. Μία από αυτές είναι η ευρωστία που παρέχει. Όπως παρατηρούμε από την πιο πάνω εικόνα η εφαρμογή FeelMe τρέχει για σχεδόν 30 ώρες συνεχόμενες. Αυτό συνεπάγεται πως η πλατφόρμα βρίσκεται σε αρκετά σταθερό επίπεδο και επιτρέπει όντως την συλλογή δεδομένων για μεγάλο χρονικό διάστημα.

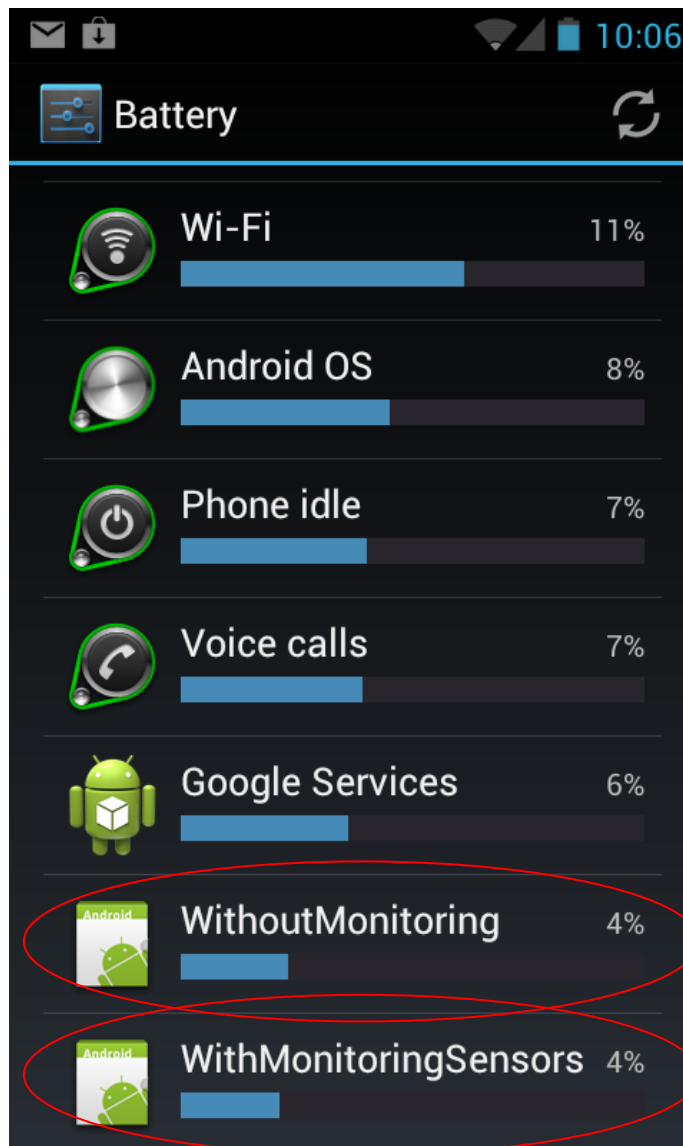
Γραμμές Κώδικα

Ένας από τους στόχους που θέσαμε εξ αρχής ήταν η ανάπτυξη εφαρμογών να γίνεται εύκολα. Ένα από τα κριτήρια ευκολίας ανάπτυξης εφαρμογών είναι οι γραμμές κώδικα που απαιτούνται. Για την εφαρμογή FeelMe οι γραμμές κώδικα που χρειάστηκαν για την συλλογή δεδομένων από την πλατφόρμα, την έναρξη αισθητήρων και γενικά της λογικής του προγράμματος ήταν περίπου 320. Αριθμός πολύ χαμηλός. Από αυτές οι πιο πολλές αφορούν την επικοινωνία της εφαρμογής με την πλατφόρμα. Οι υπόλοιπες γραμμές της εφαρμογής αφορούν την προβολή των δεδομένων και την διαπροσωπεία της εφαρμογής.

5.3 Ρύθμιση για κατανάλωση μπαταρίας

Για την αξιολόγηση της χρήσης του monitoring service και του τρόπου λειτουργίας του προχωρήσαμε στην δημιουργία δύο πειραματικών εφαρμογών οι οποίες συλλέγουν δεδομένα από το επιταχυνσιόμετρο με σταθερό ρυθμό. Η πρώτη εφαρμογή δεν χρησιμοποιεί monitoring service, δηλαδή δυναμική ρύθμιση των αισθητήρων. Η δεύτερη εφαρμογή ρυθμίζει τον ρυθμό συλλογής ως εξής. Κάθε φορά το sampling rate ρυθμίζεται ως $(\%used\ Battery + 1) * sampling\ rate$. Για παράδειγμα έστω ότι υπάρχει 80 % μπαταρία διαθέσιμη. Τότε η εφαρμογή ένα θα συλλέγει δεδομένα με ρυθμό sampling rate. Ενώ η εφαρμογή δύο με ρυθμό $1.2 * sampling\ rate$.

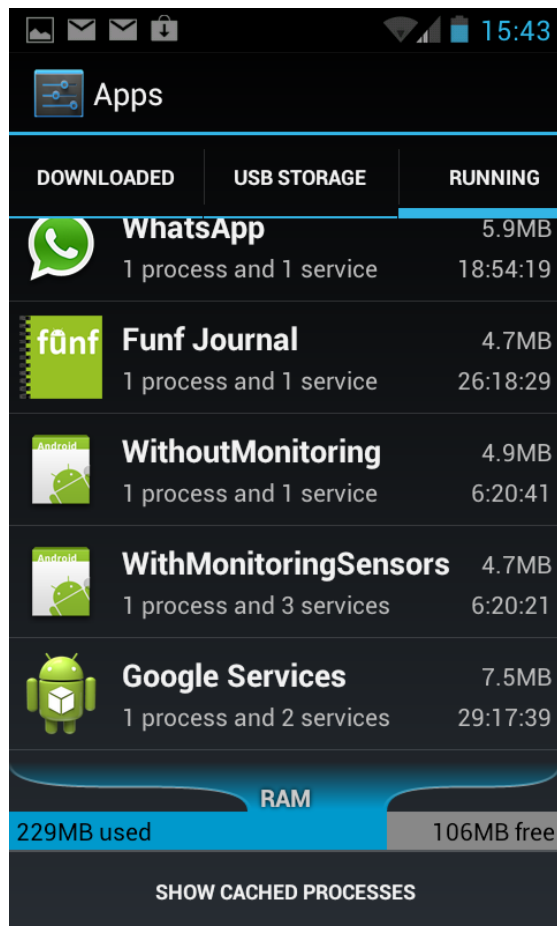
Παρατηρήσαμε πως για όσο η κινητή συσκευή φορτιζόταν και η μπαταρία διατηρείται σε ψηλά επίπεδα οι δύο εφαρμογές είχαν την ίδια κατανάλωση.



Εικόνα 25: Battery usage

Αυτό είναι λογικό και φυσιολογικό αφού και οι δύο εφαρμογές συλλέγουν δεδομένα με τον ίδιο ρυθμό

Στην συνέχεια αποσυνδέσαμε την κινητή συσκευή από τον παροχέα άρα η μπαταρία είχε αρχίσει να εξαντλείται. Παρατηρήσαμε πως σταδιακά η εφαρμογή που δεν χρησιμοποιούσε monitoring service σπαταλούσε περισσότερη ενέργεια από την εφαρμογή που χρησιμοποιούσε. Μετά από 6 περίπου ώρες συνεχούς λειτουργίας των δύο εφαρμογών έχουμε την ακόλουθη εικόνα

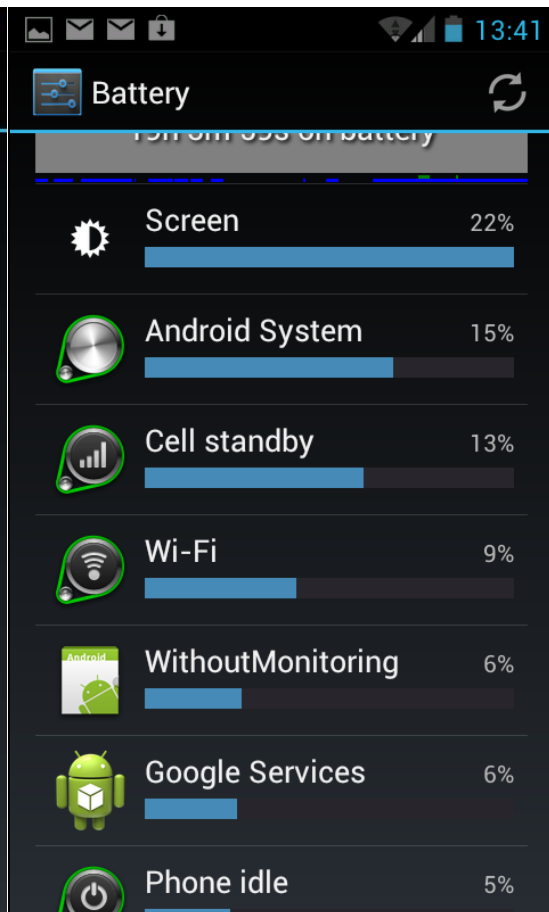


Εικόνα 26: Running Applications

Παρατηρούμε πως οι εφαρμογές τρέχουν για 6 ώρες και 20 λεπτά, έχοντας εξαντλήσει μέρος της μπαταρίας.



Εικόνα 27: Battery usage 2



Εικόνα 28: Battery usage 2

Η εφαρμογή που δεν χρησιμοποιεί monitoring service έχει σπαταλήσει 6% από την μπαταρία. Αντίθετα η εφαρμογή η οποία χρησιμοποιεί έχει ανεβάσει τον ρυθμό συλλογής της αυτόματα έχοντας σπαταλήσει λιγότερο από 4% από την μπαταρία.

Από αυτές τις δύο πειραματικές εφαρμογές προκύπτει η χρήση του monitoring service. Για όσο υπάρχουν οι κατάλληλες συνθήκες ο ρυθμός συλλογής δεδομένων συνεχίζεται κανονικά. Όταν τα επίπεδα μπαταρίας αρχίζουν να πέφτουν, τότε και ο ρυθμός συλλογής μειώνεται για να μην σπαταλήσουμε την μπαταρία του χρήστη.

Κεφάλαιο 6

Συμπεράσματα – Μελλοντική Εργασία

6.1 Συμπεράσματα	54
6.2 Μελλοντική Εργασία.....	55

Στο κεφάλαιο αυτό παρουσιάζουμε τα συμπεράσματα που προέκυψαν κατά την διάρκεια της συγκεκριμένης έρευνας καθώς και κάποιες κατευθύνσεις σχετικά με την μελλοντική εργασία που μπορεί να γίνει.

6.1 Συμπεράσματα

Σε αυτή την εργασία είχαμε την ευκαιρία να ασχοληθούμε εκτενώς με τον τομέα της αστικής αντίληψης. Η ανάπτυξη των συστημάτων αστικής αντίληψης είναι ένας τομέας έρευνας με συνεχή μελέτη και υπόκειται σε διαρκή βελτίωση. Το χαρακτηριστικό που κάνει τον τομέα αυτό να εξελίσσεται με ραγδαίους ρυθμούς είναι το γεγονός πως βασίζεται και υποστηρίζεται από συσκευές που χρησιμοποιούν στην καθημερινότητά τους οι άνθρωποι, τα κινητά τηλέφωνα. Επιπλέον σημαντικό είναι το γεγονός πως οι συμμετέχοντες στην αστική αντίληψη, τα άτομα δηλαδή που παρέχουν δεδομένα, νιώθουν το αίσθημα πως συμβάλλουν με τον τρόπο τους στην πρόοδό της επιστήμης ικανοποιώντας το αίσθημα αλτρουισμού που έχουν έμφυτο όλοι άνθρωποι σε κάποιο βαθμό. Από την πλευρά μας η επιστημονική κοινότητα οφείλουμε να επενδύσουμε σε αυτή την κατεύθυνση δημιουργώντας εφαρμογές που θα συλλέγουν τα απαραίτητα δεδομένα επιβαρύνοντας τους χρήστες όσο το δυνατόν λιγότερο.

Η εργασία μας στον συγκεκριμένο τομέα ξεκίνησε πριν από περίπου ένα χρόνο. Αρχικά η προσπάθεια που καταβλήθηκε αφορούσε την κατανόηση του τομέα της αστικής αίσθησης, των ιδιαίτερων χαρακτηριστικών του και των προβλημάτων που υπάρχουν. Στη συνέχεια διεξήχθη εκτενής μελέτη των εφαρμογών και των πλατφόρμων

ενδιάμεσου λογισμικού που υπάρχουν στο ερευνητικό πεδίο. Μελετήθηκαν σε μεγάλο βάθος οι υπηρεσίες που προσφέρουν, η αρχιτεκτονική και ο τρόπος λειτουργίας τους. Η πλατφόρμα Funf αξίζει ιδιαίτερης αναφοράς επειδή ήταν ανοικτού λογισμικού, κατά την γνώμη μας ήταν η πληρέστερη από τις σχετικές εργασίες και αποτέλεσε πηγή έμπνευσης για την δική μας πλατφόρμα.

Μέσα από την έρευνά μας διαπιστώσαμε πως αρκετά ζητήματα του τομέα ακόμα μένουν ανοικτά ή οι απαντήσεις που δόθηκαν δεν ήταν ικανοποιητικές. Απώτερος μας στόχος ήταν να δημιουργήσουμε μία νέα πλατφόρμα η οποία να αντιμετωπίζει άμεσα τα προβλήματα που εμείς θεωρούμε πιο σημαντικά. Η πλατφόρμα λογισμικού που αναπτύχθηκε έχει ως κύριο στόχο την συλλογή δεδομένων σεβόμενοι πάντα τον χρήστη. Χρησιμοποιώντας την πλατφόρμα μας ερευνητές χωρίς μεγάλη προγραμματιστική εμπειρία έχουν την δυνατότητα να αναπτύξουν εφαρμογές αστικής αίσθησης που εξοικονομούν ενέργεια και πόρους. Τέλος, μέσα από αυτή την εργασία επιχειρήσαμε να αντιμετωπίσουμε το πρόβλημα από μία νέα οπτική γωνία. Παρουσιάζουμε την συλλογή δεδομένων μέσα από μία αυτοματοποιημένη διαδικασία (συλλογή δεδομένων, αποθήκευση, αποστολή, συντονισμός) χρησιμοποιώντας μία αφαιρετική και επεκτάσιμη αναπαράσταση των δεδομένων. Πιστεύουμε αυτή είναι και η μεγαλύτερη συνεισφορά της εργασίας μας.

6.2 Μελλοντική Εργασία

Η πλατφόρμα που παρουσιάζουμε έχει ως κύριο χαρακτηριστικό την επεκτασιμότητα. Αν και καταφέραμε να φέρουμε την πλατφόρμα σε ένα καλό και λειτουργικό στάδιο, υπάρχουν ακόμα μερικοί τομείς που χρίζουν βελτίωσης.

Ως μελλοντική εργασία θεωρούμε βασικό να αναπτυχθεί η πλατφόρμα λαμβάνοντας υπόψη τον παράγοντα χρόνου στα δεδομένα. Δηλαδή να δίνει την δυνατότητα στον προγραμματιστή να καθορίζει χρονική προτεραιότητα στα δεδομένα που συλλέγονται. Να υπάρχουν δεδομένα που πρέπει να σταλούν άμεσα και δεδομένα που να είναι ελαστικά στον χρόνο αποστολής τους. Αυτό απαιτεί μία μορφή χρονοδρομολόγησης των δεδομένων και φυσικά πρόνοια για αποφυγή δυσάρεστων

καταστάσεων όπως παρατεταμένη στέρηση. Η συγκεκριμένη βελτίωση χρειάζεται κάποια προσπάθεια για να υλοποιηθεί όμως πιστεύουμε είναι σημαντική να γίνει.

Επιπλέον, είναι σημαντικό να υπάρχει η δυνατότητα για κρυπτογράφηση των δεδομένων πριν την αποστολή το φ. Παρόλο που μέσω της πλατφόρμας δεν συλλέγουμε δεδομένα που περιέχουν ευαίσθητες προσωπικές πληροφορίες των χρηστών ή πληροφορίες με τις οποίες θα μπορούσαμε να ταυτοποιήσουμε κάποιο χρήστη, είναι σημαντικό να υπάρχει κρυπτογράφηση των δεδομένων. Ο λόγος είναι πως ενισχύει το αίσθημα ασφάλειας των χρηστών ιδίως αυτών που δεν θεωρούνται έμπειροι χρήστες. Η βελτίωση αυτή δεν χρειάζεται ιδιαίτερη προσπάθεια για να υλοποιηθεί.

Τέλος, θα μπορούσαμε να βελτιώσουμε τον τρόπο αποθήκευσης των δεδομένων. Αντί να αποθηκεύουμε τα δεδομένα σε αρχεία σαν CSS values θα μπορούσαμε να χρησιμοποιήσουμε βάση δεδομένων. Η πλατφόρμα android παρέχει την δυνατότητα να χρησιμοποιήσουμε την βάση δεδομένων SQLite. Η χρήση μιας βάσης δεδομένων θα βοηθούσε στον καλύτερο έλεγχο των δεδομένων και ευκολότερη ανάλυσή τους στην συνέχεια.

Βιβλιογραφία

- [1] Kiran K. Rachuri, Cecilia Mascolo, Mirco Musolesi, and Peter J. Rentfrow, “Sociable Sence”, *Proceedings of the 17th annual international conference on Mobile computing and networking - MobiCom '11*, New York, New York, USA: 2011, p. 73.
- [2] Donnie H. Kim, Younghun Kim, Deborah Estrin and Mani B. Srivastava “SensLoc”, *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems – SenSys '10*, Zurich, Switzerland: 2010, p. 43.
- [3] R. Ballagas, F. Memon, R. Reiners, and J. Borchers, “iStuff mobile”, *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '07*, San Jose, California, USA: 2007, p. 1107.
- [4] H. Lu, W. Pan, N.D. Lane, T. Choudhury, and A.T. Campbell, “SoundSense”, *Proceedings of the 7th international conference on Mobile systems, applications, and services - Mobisys '09*, Wroclaw, Poland: 2009, p. 165.
- [5] H. Lu, J. Yang, Z. Liu, N.D. Lane, T. Choudhury, and A.T. Campbell, “The Jigsaw continuous sensing engine for mobile phone applications”, *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems - SenSys '10*, Zurich, Switzerland: 2010, p. 71.
- [6] M. Mun, P. Boda, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, and R. West, “PEIR, the personal environmental impact report, as a platform for participatory sensing systems research”, *Proceedings of the 7th international conference on Mobile systems, applications, and services - Mobisys '09*, Wroclaw, Poland: 2009, p. 55.
- [7] S. Smaldone, B. Gilbert, N. Bila, L. Iftode, E. de Lara, and M. Satyanarayanan, “Leveraging smart phones to reduce mobility footprints”, *Proceedings of the 7th*

international conference on Mobile systems, applications, and services - Mobisys '09, Wroclaw, Poland: 2009, p. 109.

- [8] Jon Froehlich, Mike Y. Chen, Sunny Consolvo, Beverly Harrison, James A. Landay, "MyExperience: a system for in situ tracing and capturing of user feedback on mobile phones", *Proceedings of the 5th international conference on Mobile systems, applications, and services - Mobisys '07*, New York, USA: 2007, pp. 55-70.
- [9] Y. Zheng, Y. Chen, Q. Li, X. Xie, and W.-Y. Ma, "Understanding transportation modes based on GPS data for web applications", *ACM Transactions on the Web*, vol. 4, Jan. 2010, pp. 1-36.
- [10] Android SDK, <http://developer.android.com/sdk/index.html>
- [11] API, http://en.wikipedia.org/wiki/Application_programming_interface
- [12] Global Positioning System, http://en.wikipedia.org/wiki/Global_Positioning_System
- [13] Google Android, [http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))
- [14] Google Maps API, <http://code.google.com/intl/el-GR/apis/maps/index.html>
- [15] HTC Desire, www.htc.com/.../desire/overview.html
- [16] Java Virtual Machine, http://en.wikipedia.org/wiki/Java_Virtual_Machine
- [17] MotorolaDynaTAC, http://en.wikipedia.org/wiki/Motorola_DynaTAC
- [18] Sensaris, <http://www.sensaris.com/>
- [19] Funf, <http://www.funf.com/>

Παράρτημα Α

Στο παράρτημα αυτό επισυνάπτεται το Application Programming Interface της πλατφόρμας λογισμικού Feel The World. Μέσω αυτού του API ο προγραμματιστής μπορεί να χρησιμοποιήσει την πλατφόρμα αυτή.

IData

Data interface class

Method Summary	
int	getType()
java.lang.String	toString()

Class BaseBackgroundSensor

java.lang.Object

└ Service

└ thesis.framework.Base.BaseBackgroundSensor

All Implemented Interfaces:

[SensorTypes](#)

Direct Known Subclasses:

[BackgroundExternalSense](#), [BackgroundGenericSense](#), [BackgroundSenseLocation](#), [BackgroundSenseWifi](#), [MicSensor](#)

```
public abstract class BaseBackgroundSensor
extends Service
implements SensorTypes
```

Field Summary

protected static int	CAPACITY capacity of list of storing recent data
protected java.util.ArrayList< IData >	collectedData list for storing recent data
protected static boolean	DEFAULT_DO_SEND Default option for sending to server
protected static boolean	DEFAULT_IS_MONITORED Default option for binding to

	monitoring service
protected static DataPriority	DEFAULT_PRIORITY Default priority value - normal
protected static int	DEFAULT_RATE Default rate if developer hasn't given one
protected static int	DEFAULT_WRITE_RATE Default duration if developer hasn't given one
protected static boolean	DEFAULT_WRITE_TO_FILE Default option for writing in file
protected Handler	handler Handler for performing rates
protected boolean	isMonitored if sensor is monitored by monitoring Service
protected DataPriority	priority priority of data
protected int	rate Sampling Rate
protected boolean	running controls the thread
protected boolean	sendToServer if collected data will be send to server
protected int	sensorType type of the sensor that is observed
protected static java.lang.String	TAG Tags for Log cat debugging
protected static java.lang.String	TAG_D
protected thesis.framework.Base.BaseBackgroundSensor.WriteData	writer timer task for

		writing data to file
protected	java.util.Timer	writesTimer timer for scheduling writes
protected	boolean	writeToFile write data to file
protected	int	writeToFileRate Rate of writing to file

Fields inherited from interface [thesis.framework.Base.SensorTypes](#)

[ACCELEROMETER](#), [AMBIENT_TEMPERATURE](#), [CAM_SENSOR](#), [EXTERNAL_SENSOR](#), [GRAVITY](#), [GYROSCOPE](#), [LIGHT](#), [LINEAR_ACCELEROMETER](#), [LOCATION_GPS](#), [LOCATION_NETWORK](#), [MAGNETIC_FIELD](#), [MIC_SENSOR](#), [ORIENTATION](#), [PRESSURE](#), [PROXIMITY](#), [RELATIVE_HUMIDITY](#), [ROTATION_VECTOR](#), [TEMPERATURE](#), [WIFI_RECEIVER](#)

Constructor Summary

[BaseBackgroundSensor](#)()

Method Summary

protected	void	CheckIfServiceIsRunning () if monitoring service is running binds to it
protected	void	doBindService () perform binding with monitoring service
abstract	void	start (Context callerContext) Example Intent intent = new Intent(callerContext,Your.class); intent.putExtra("rate", 3000); intent.putExtra("duration", 1000); intent.putExtra("writeToFile",false); callerContext.startService(new Intent(callerContext, Your.class));
abstract	void	stop (Context callerContext)

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

ResourceManager

Method Summary

static int	<u>get_battery()</u> Get current battery level
static int	<u>getCPU_Freq()</u> Get CPU frequency
static boolean	<u>isRunning()</u>
IBinder	<u>onBind()</u> (Intent intent)
void	<u>onCreate()</u> On Service Create
void	<u>onDestroy()</u> On Service Destroy
void	<u>onStart()</u> (Intent intent, int startid) On Service Start

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

TransferService

Method Summary

static void	<u>getCompressedFileFromServer()</u> (java.lang.String serverIP, int serverPort, java.lang.String path) Method use to get a compressed file from the server
static void	<u>getFileFromServer()</u> (java.lang.String serverIP, int serverPort, java.lang.String path) Method use to get a file from the server
static void	<u>sendCompressedFileToServer()</u> (java.lang.String serverIP, int serverPort, java.lang.String filePath, int type) Method use to send a compressed file from the device to the server
static void	<u>sendFileAndTextToServer()</u> (java.lang.String serverIP, int serverPort, java.lang.String filePath, int type, java.lang.String text) Method use to send a file from the device to the server
static void	<u>sendFileToServer()</u> (java.lang.String serverIP, int serverPort, java.lang.String filePath) Method use to send a file from the device to the server

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

MonitoringService

java.lang.Object

└ Service

└ thesis.framework.Monitoring.MonitoringService

```
public class MonitoringService
extends Service
```

Class responsible for monitoring sensor clients Registered clients get messages in order to set their rate, duration e.t.c.

Author:

Theo

Field Summary

static int	MSG_REGISTER_CLIENT Protocol Values for handling messages
static int	MSG_SET_DURATION
static int	MSG_SET_MINDISTANCE
static int	MSG_SET_RATE
static int	MSG_STOP
static int	MSG_UNREGISTER_CLIENT

Constructor Summary

MonitoringService()	
-------------------------------------	--

Method Summary

static boolean	isRunning()
IBinder	onBind (Intent intent)
void	onCreate ()
void	onDestroy ()
int	onStartCommand (Intent intent, int flags, int startId)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

FileWriterHelper

java.lang.Object

└─ thesis.framework.utilities.FileWriterHelper

```
public class FileWriterHelper
extends java.lang.Object
```

Utility class that offers the functionalities of writing the data to files

Author:

Theo

Field Summary

static java.lang.String	app_folder application's storage folder on external storage root directory
-------------------------	---

Constructor Summary

FileWriterHelper ()	
-------------------------------------	--

Method Summary

static java.lang.String	writeToFile (java.util.ArrayList< IData > dataList, int sensor_type) Write data to file
-------------------------	--

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

BluetoothManager

java.lang.Object

└─ Activity

└─ thesis.framework.utilities.BluetoothManager

```
public class BluetoothManager
extends Activity
```

Class used to get a list of the bluetooth devices that are in range example of use: Intent myIntent = new Intent(getApplicationContext(), BluetoothScanner.class); startActivityResult(myIntent, 0); BluetoothScanner.getDevices();

Author:

harisefstathiades - Theo

Constructor Summary

[BluetoothManager\(\)](#)

Method Summary

static BluetoothSocket	connectSelectedDevice (BluetoothDevice device) First run: Intent myIntent = new Intent(getApplicationContext(), BluetoothScanner.class); startActivityResult(myIntent, 0); and select a device Connect to a bluetooth device
static void	disconnectSocket () disconnect the connected bluetooth Socket
static java.util.ArrayList<BluetoothDevice>	getDevices ()
static BluetoothDevice	getSelectedDevice () Get selected device
protected void	onActivityResult (int requestCode, int resultCode, Intent data)
void	onCreate (Bundle savedInstanceState)
boolean	onCreateOptionsMenu (Menu menu)
protected void	onDestroy ()
boolean	onOptionsItemSelected (MenuItem item)
void	showListWithDevices () show pop up list with devices founded

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

DeflateCompression

java.lang.Object

└ thesis.framework.utilities.DeflateCompression

```
public class DeflateCompression
extends java.lang.Object
```

Class that contains the methods for deflate compression and decompression

Author:

harisefstathiades

Constructor Summary

DeflateCompression()	
--------------------------------------	--

Method Summary

static int	compressData (byte[] data, byte[] output, int len) Method for deflate compression
static int	compressString (java.lang.String data, byte[] output, int len) Method for deflate compression
static byte[]	decompressData (byte[] input, int len) Method for deflate decompression
static java.lang.String	decompressString (byte[] input, int len) Method for deflate decompression

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
--