



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών
«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Υλοποίηση εφαρμογής καταγραφής και ανάλυσης διαδρομών (ROUTE ANALYZER) σε πλατφόρμα Android Development of Route Analyzer application (recording, tracking and analyzing routes) on Android platform
Όνοματεπώνυμο Φοιτητή	Εμμανουήλ Παπουτσής
Πατρώνυμο	Γεώργιος
Αριθμός Μητρώου	ΜΠΣΠ/ 13084
Επιβλέπων	Dr. Αλέξης Ευθύμιος, Λέκτορας

Πανεπιστήμιο Πειραιώς

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Όνομα Επώνυμο
Βαθμίδα

Όνομα Επώνυμο
Βαθμίδα

Όνομα Επώνυμο
Βαθμίδα

Περιεχόμενα

Περίληψη	4
1. Παρουσίαση Και Χρήση Της Εφαρμογής.....	5
1.1 Ανάλυση Αναγκών.....	5
1.2 Περιγραφή εφαρμογής.....	6
1.3 Τεχνικές Προδιαγραφές, Απαιτήσεις Συστήματος	6
1.4 Εγχειρίδιο Οδηγιών Χρήσης- Επεξήγηση Γραφικού Περιβάλλοντος.....	7
2. Αρχιτεκτονική Συστήματος.	48
2.1 Επιλογή Των Εργαλείων.	48
2.2 Περιγραφή Ανάπτυξης – Χρήσης Τεχνολογιών.....	48
3. Μελλοντικές Αναβαθμίσεις	74
4. ΑΝΑΦΟΡΕΣ – LINKS	75

Περίληψη

Η παρούσα πτυχιακή εργασία αποτελεί συνέχεια της εργασίας του ΠΟΛΑΚΗ Κωνσταντίνου και περιγράφει τις λεπτομέρειες υλοποίησης της προαναφερθείσας εφαρμογής. Στις σελίδες που ακολουθούν ο αναγνώστης θα γνωρίσει τις ανάγκες που ικανοποιεί η εφαρμογή, τις τεχνικές της προδιαγραφές, καθώς και τον τρόπο λειτουργίας της. Τέλος θα γνωρίσει τα εργαλεία και τις τεχνολογίες που χρησιμοποιήθηκαν κατά τη κατασκευή της.

Abstract

This thesis is the continuation of POLAKIS Konstantinos paper and it describes the technical details of the applications development. In the next chapters, the reader will understand the needs that the application covers, the technical specifications and how it works. Finally, the tools and the technologies that have been used to build this application will be presented and analysed.

1. Παρουσίαση Και Χρήση Της Εφαρμογής

1.1 Ανάλυση Αναγκών.

Είναι μια γενικά αποδεκτή αλήθεια πως μέσω της σύγχρονης τεχνολογίας η καθημερινή ζωή των ανθρώπων αλλάζει ριζικά. Νέες δυνατότητες, εργαλεία που δε μπορούσαμε να φανταστούμε μόλις λίγες δεκάδες χρόνια πριν. Νέες ανάγκες που δημιουργούνται με τα νέα δεδομένα, κίνδυνοι και προβληματισμοί που προκύπτουν από όλη την ανάπτυξη αυτή. Πολλά ζητήματα τα οποία αφορούν τον καταγισμό καινοτομιών που δεχόμαστε και είναι σίγουρα πολύ δύσκολο να μπορέσουμε να διαχειριστούμε όλα τα θέματα αυτά σε τόσο σύντομο χρονικό διάστημα. Παραμερίζοντας όμως τα ηθικά ζητήματα που εγείρονται από την πρόοδο και τις αλλαγές αυτές -ξεφεύγουν του θέματος της παρούσας εργασίας και του αντικειμένου κατάρτισής μας- οφείλουμε να αναγνωρίσουμε πως έχουμε στα χέρια μας νέα εργαλεία που μπορούν να προσφέρουν σημαντικές υπηρεσίες σε επαγγελματίες και ιδιώτες.

Οι υπηρεσίες εντοπισμού θέσης είναι πιο προσιτές από οποτεδήποτε. Υπολογιστικά συστήματα που μέχρι πρότινος κάθονταν πάνω στα γραφεία μας βρίσκονται πια στις τσέπες μας. Αυτό μας παρέχει τεράστιες δυνατότητες διαμόρφωσης και αποστολής της πληροφορίας και σε συνδυασμό με τις ολοένα πιο εξελιγμένες και φθηνές δικτυακές υπηρεσίες καθίστανται εξαιρετικά αποδοτικές. Με την ανάπτυξη των τεχνολογιών αυτών φτάνουμε στο σημείο να μπορούμε να καλύψουμε ανάγκες ελέγχου θέσης αντικειμένων ή προσώπων, να μπορούμε να καταγράψουμε τις διαφορετικές θέσεις από όπου διήλθαν σε βάθος χρόνου και να προχωρήσουμε σε στατιστική ανάλυση των δεδομένων αυτών. Επαγγελματικές ανάγκες παρακολούθησης στόλου οχημάτων, οδηγική συμπεριφορά οδηγών και τήρηση κανόνων του κώδικα οδικής κυκλοφορίας, αποστολή ευαίσθητων πληροφοριών και ευπαθών αντικειμένων, επαγγελματικές ανάγκες καταγραφής της πορείας του χρήστη, ανάγκες καταγραφής πορείας για λόγους αναψυχής και άλλες πολλές περιπτώσεις που χρειάζονται μια λεπτομερή ανάλυση της πορείας, της θέσης και της τάσης κίνησης.

Τέλος η ανάγκη αποδοτικότερης διαχείρισης των ορυκτών καυσίμων κρίνεται αναγκαία στην εποχή μας λόγω του υψηλού κόστους καυσίμων αλλά και της μόλυνσης του περιβάλλοντος. Το γεγονός αυτό καθιστά απαραίτητη και την καταγραφή της κατανάλωσης καυσίμου των οχημάτων. Το εγχείρημα αυτό γίνεται όλο και πιο πραγματοποιήσιμο στις μέρες μας μιας και οι κατασκευαστές οχημάτων εισέρχονται δειλά στις νέες τεχνολογίες της εποχής μας. Σήμερα είναι μικρός ο αριθμός των οχημάτων που εξοπλίζονται με συσκευές που διαθέτουν διασυνδέσεις ώστε να μπορούν να μεταφέρουν δεδομένα από τη λειτουργία του οχήματος, αριθμός ο οποίος αναμένεται όμως να αυξηθεί μιας και λειτουργίες σαν και αυτή έχουν μεγάλη απήχηση στο καταναλωτικό κοινό. Επιπλέον υπάρχουν πολλές εταιρίες που παράγουν συσκευές που αναλαμβάνουν αυτή τη διασύνδεση με το κόστος απόκτησης μιας τέτοιας συσκευής να είναι ιδιαίτερα χαμηλό με αποτέλεσμα να είναι εφικτή η καταγραφή της κατανάλωσης ακόμα και σε παλαιότερα οχήματα.

Τις ανάγκες αυτές η παρούσα εργασία επιχειρεί να καλύψει σε ένα βαθμό. Η εφαρμογή που συνοδεύει την παρούσα εργασία αποτελείται από δύο βασικές λειτουργίες:

- Λειτουργία καταγραφής διανυθείσας διαδρομής, συμπεριλαμβανομένων και των στάσεων, για το χρήστη της εφαρμογής.
- Λειτουργία εμφάνισης προγενέστερης διαδρομής στο χάρτη της Google Maps αλλά και στατιστικών δεδομένων της διαδρομής όπως συνταραγμένες, ταχύτητα και κατανάλωση.

1.2 Περιγραφή εφαρμογής

Η εφαρμογή αυτή θα εκτελείται σε έξυπνα κινητά τηλέφωνα που διαθέτουν λειτουργικό σύστημα τύπου Android, και θα μπορεί να αποθηκεύει πληροφορίες των διαδρομών και των υποδιαδρομών αυτών που πραγματοποιεί ο χρήστης. Πιο συγκεκριμένα, όταν ο χρήστης ξεκινά να κινείται η εφαρμογή θα ξεκινά να καταγράφει τη διαδρομή του. Όταν εκείνος πάψει να κινείται για συγκεκριμένο χρονικό διάστημα η εφαρμογή θα καταγράψει τη χρονική διάρκεια της στάσης. Όταν αρχίσει εκ νέου να κινείται η εφαρμογή θα ξεκινήσει μια νέα υποδιαδρομή. Όταν ο χρήστης ολοκληρώσει τη διαδρομή του η εφαρμογή θα έχει αποθηκεύσει τη διαδρομή μαζί με τις υποδιαδρομές της. Με τον τρόπο αυτό ο χρήστης της εφαρμογής θα είναι σε θέση να γνωρίζει από πού ξεκίνησε, που σταμάτησε και για πόσο χρόνο.

Προκειμένου η εφαρμογή να καταγράψει τη διαδρομή θα δέχεται πληροφορίες της θέσης του χρήστη από τον αισθητήρα GPS της συσκευής και θα τις αποθηκεύει μαζί με πληροφορία για τη χρονική στιγμή που εκείνος βρισκόταν στη θέση αυτή. Παράλληλα με τις πληροφορίες θέσης η εφαρμογή θα αποθηκεύει την ταχύτητα με την οποία κινείται αλλά και την κατανάλωση καυσίμου του οχήματος εκείνης της χρονικής στιγμής. Με τον τρόπο αυτό ο χρήστης της εφαρμογής θα μπορεί να παρακολουθεί στο χάρτη της Google Maps¹ την ακριβή του θέση, να αποθηκεύει τις διαδρομές που πραγματοποιεί μαζί με τις στάσεις που έκανε, τη χρονική διάρκεια των στάσεων αλλά και τη ταχύτητα – κατανάλωση του οχήματος.

Ακολούθως ο χρήστης θα μπορεί να ζητά από την εφαρμογή να του εμφανίσει λεπτομέρειες συγκεκριμένης διαδρομής και των υποδιαδρομών της που έχει αποθηκεύσει. Έτσι θα μπορεί να γνωρίζει από ποιες συντεταγμένες ξεκίνησε αλλά και τη χρονική στιγμή που ξεκίνησε, που σταμάτησε, για πόση ώρα βρισκονταν σε κατάσταση στάσης, πόση απόσταση διένυσε συνολικά, τη ταχύτητα με την οποία κινείτο (μέση, μέγιστη αλλά και ελάχιστη), την κατανάλωση καυσίμου του οχήματος (ομοίως μέση, μέγιστη αλλά και ελάχιστη) καθώς και την κατάσταση της μπαταρίας της συσκευής κατά την αρχή αλλά και κατά την ολοκλήρωση της διαδρομής. Τέλος θα μπορεί να σχεδιάσει τη συγκεκριμένη διαδρομή στο χάρτη της Google Maps με «πινέζες» στα σημεία έναρξης και ολοκλήρωσης των υποδιαδρομών που αποτελείται η συγκεκριμένη διαδρομή. Οι υποδιαδρομές θα εμφανίζονται με διαφορετικό χρώμα ώστε να είναι ευδιάκριτες και πατώντας πάνω στις πινέζες έναρξης ο χρήστης θα μπορεί να βλέπει τη χρονική στιγμή που η υποδιαδρομή ξεκίνησε ενώ πατώντας σε αυτές της ολοκλήρωσης υποδιαδρομής θα βλέπει τη χρονική διάρκεια της στάσης.

Όμοια θα μπορεί να πράξει και για τις υποδιαδρομές των αποθηκευμένων διαδρομών. Πιο συγκεκριμένα, θα μπορεί να βλέπει και σε αυτές τη χρονική στιγμή που ξεκίνησε η υποδιαδρομή, τη χρονική στιγμή που ολοκληρώθηκε, τις συντεταγμένες του σημείου εκκίνησης αλλά και ολοκλήρωσης, της τιμές ταχύτητας αλλά και κατανάλωσης, τη κατάσταση της μπαταρίας καθώς και τη διάρκεια της στάσης. Τέλος θα μπορεί να τις σχεδιάζει και αυτές στο χάρτη της Google Maps τοποθετώντας πινέζες στα σημεία όπου έχουν ληφθεί δεδομένα. Πατώντας πάνω στη πινέζα ο χρήστης θα μπορεί να βλέπει τη ταχύτητα και τη κατανάλωση καυσίμου εκείνης της τοποθεσίας.

1.3 Τεχνικές Προδιαγραφές, Απαιτήσεις Συστήματος

Για τη σωστή λειτουργία της εφαρμογής θα χρειαστεί η συσκευή να είναι εξοπλισμένη με το λειτουργικό Android 2.2 το οποίο υποστηρίζει τη πλατφόρμα Google Api 8.

Η λειτουργία της εφαρμογής μας εξαρτάται κυρίως από τη λειτουργία του GoogleMap API στη συσκευή, επομένως οι ελάχιστες απαιτήσεις σε hardware διαμορφώνονται ως εξής:

1. Οθόνη: QVGA (240x320 pixels) touch screen
2. Πληκτρολόγιο: Virtual keyboard support
3. CPU: Δεν υπάρχουν ελάχιστες απαιτήσεις
4. Μνήμη: 92MB RAM; 150MB χώρος αποθήκευσης

5. Κουμπιά: Home, Menu, και Back
6. Συνδέσεις: Wireless high-speed data ικανά να υποστηρίξουν 200Kbps,
7. Επιπλέον Εξοπλισμός: Compass, GPS receiver, Bluetooth.

Λόγω του ότι η εφαρμογή ανακτά πληροφορίες από τη Google για να σχεδιάσει το χάρτη η συσκευή θα πρέπει να διαθέτει σύνδεση στο Internet είτε μέσω του provider, όπου απαιτείται 3G κατ' ελάχιστο, είτε μέσω Wi-Fi.

Επιπλέον η εφαρμογή επικοινωνεί με το ενσωματωμένο σύστημα διάγνωσης του οχήματος και για το λόγο αυτό απαιτείται ολοκληρωμένο κύκλωμα τύπου ELM327, ή αντίστοιχο, με θύρα επικοινωνίας τύπου Bluetooth. Τέλος το όχημα θα πρέπει να διαθέτει θύρα OBD II για τη διασύνδεση του άνω κυκλώματος. Η θύρα αυτή απαντάται σε όλα τα αυτοκίνητα και τα φορτηγά κατασκευής του 1996 και έπειτα.

1.4 Εγχειρίδιο Οδηγιών Χρήσης- Επεξήγηση Γραφικού Περιβάλλοντος

1.4.1 Σύνδεση του Ολοκληρωμένου Κυκλώματος Στο Όχημα – Συσκευή Χρήστη.

Το ολοκληρωμένο κύκλωμα συνδέεται με το ενσωματωμένο σύστημα διάγνωσης του οχήματος μέσω της θύρας OBD. Η θύρα αυτή βρίσκεται στο εσωτερικό του οχήματος και σε ακτίνα ενός μέτρου από τη θέση του οδηγού. Στα περισσότερα οχήματα βρίσκεται δίπλα στις ασφάλειες του οχήματος η κάτω από το χειρόφρενο. Η θύρα είναι σχεδιασμένη με τέτοιο τρόπο που δεν επιτρέπει τη λανθασμένη σύνδεση. Η σύνδεση θα πρέπει να γίνει με το διακόπτη της μίζας του οχήματος σε θέση κλειστός.

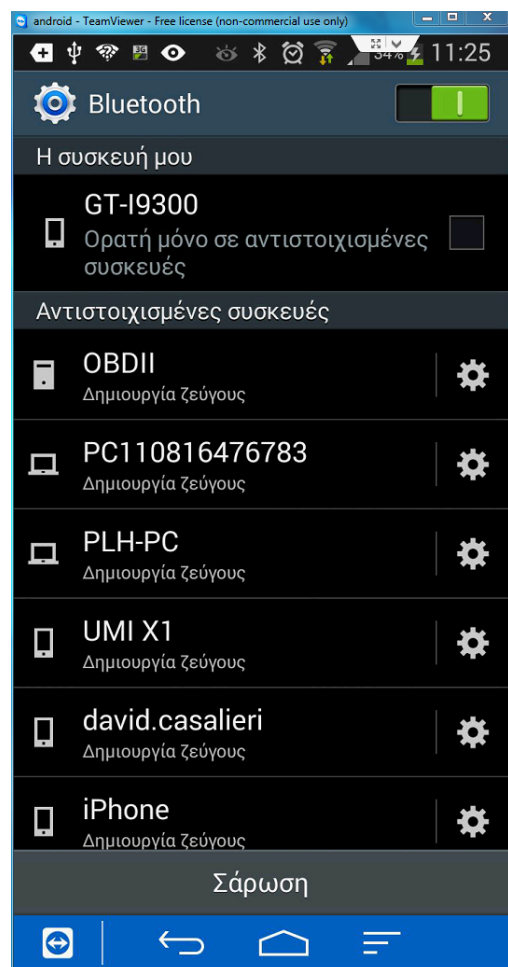


Εικόνα 1: Σύγκριση της θύρας του ολοκληρωμένου με αυτή του οχήματος. Η πάνω πλευρά των θυρών είναι πιο πλατιά από την κάτω προκειμένου να μην είναι εφικτή η λανθασμένη σύνδεση.



Εικόνα 2: Το ολοκληρωμένο σε σύνδεση με τη θύρα OBDII του οχήματος. Διακρίνεται το κόκκινο LED που σηματοδοτεί την ενεργοποίηση της συσκευής.

Αφού ολοκληρώσουμε τη σύνδεση του ολοκληρωμένου με το όχημα θα πρέπει να φέρουμε το διακόπτη της μίζας του οχήματος στη θέση ανοιχτός αλλά να μην θέσουμε τον κινητήρα σε λειτουργία. Ακολούθως αντιστοιχίζουμε το κινητό μας με το ενσωματωμένο, που ονομάζεται OBDII, εισάγοντας τον κωδικό '1234'. Είμαστε πλέον έτοιμοι να επικοινωνήσουμε με το ολοκληρωμένο.

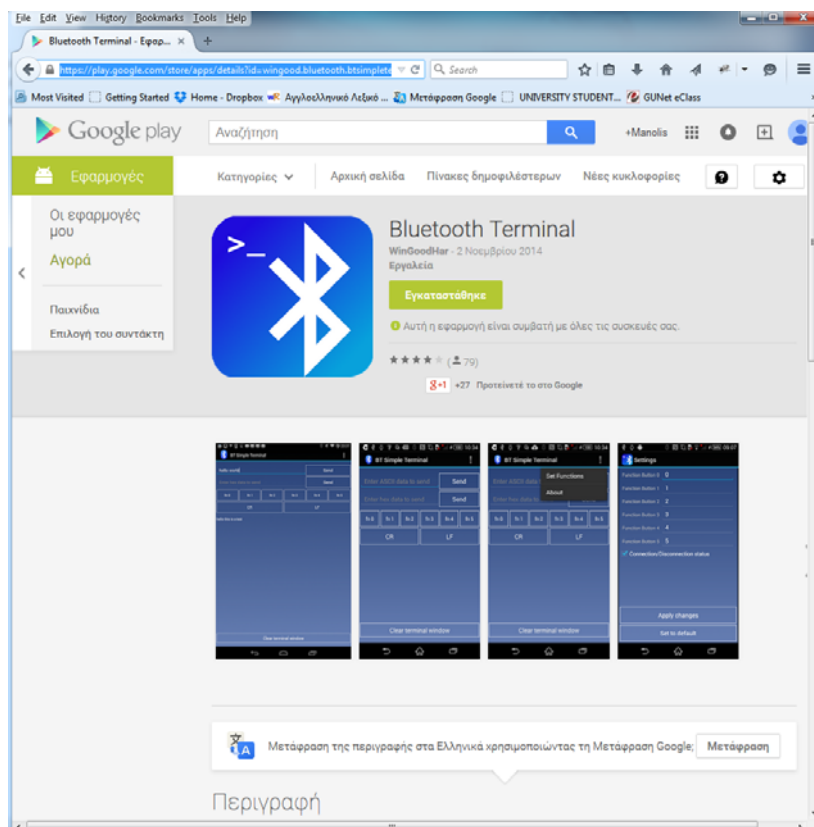


Εικόνα 3: Το ολοκληρωμένο, με όνομα OBDII, αντιστοιχισμένο με το κινητό τηλέφωνο. Ο υπολογιστής με το όνομα PC110816476783 αντιστοιχεί στον υπολογιστή που εκτελεί το λογισμικό εξομοιωτή του OBD II.

1.4.2 Ρύθμιση του Ολοκληρωμένου Κυκλώματος.

Προτού ξεκινήσει κανείς να χειρίζεται την εφαρμογή μας θα πρέπει να ρυθμίζει το ολοκληρωμένο κύκλωμα ώστε να μπορεί να επικοινωνήσει με την εφαρμογή. Στη περίπτωση που το ολοκληρωμένο έχει τις εργοστασιακές ρυθμίσεις του δεν απαιτείτε κάποια ρύθμιση. Όταν όμως έχει προηγηθεί η χρήση του από κάποια άλλη εφαρμογή, όπως η Torque που αναφέραμε εμείς, τότε θα πρέπει να γίνει επαναφορά του ολοκληρωμένου στις εργοστασιακές ρυθμίσεις. Αυτό είναι απαραίτητο μιας και η εφαρμογή Torque ρυθμίζει το ολοκληρωμένο να επικοινωνεί με αυτή. Συγκεκριμένα η εφαρμογή καταργεί το echo με αποτέλεσμα το ολοκληρωμένο να μην ενσωματώνει στην απάντηση που στέλνει τους χαρακτήρες που έλαβε. Επίσης καταργεί και τα κενά μεταξύ των χαρακτήρων. Αποτέλεσμα αυτών των ενεργειών είναι να αλλάζει η μορφή της απάντησης του ολοκληρωμένου και η δική μας εφαρμογή να μην δύναται να λειτουργήσει αφού είναι σχεδιασμένη να διαβάζει τη μορφή των απαντήσεων που έχει το ολοκληρωμένο όταν βρίσκεται στις εργοστασιακές ρυθμίσεις.

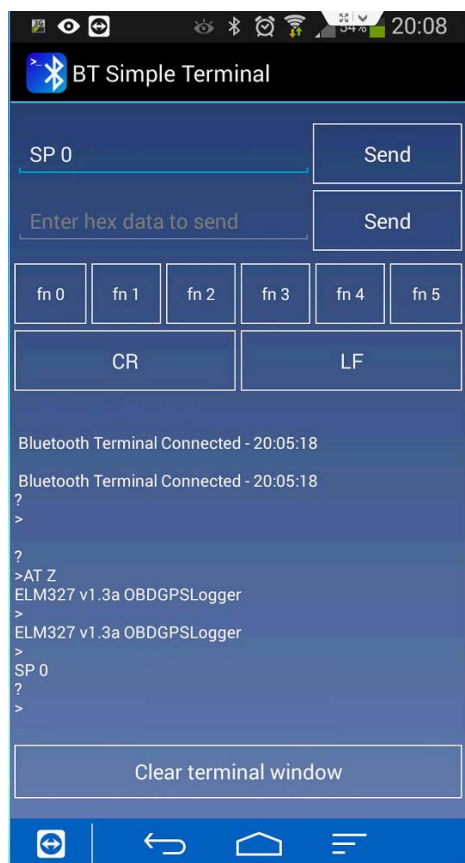
Για το λόγο αυτό κρίνεται σκόπιμο ο χρήστης προτού αποπειραθεί να χρησιμοποιήσει την εφαρμογή μας να εγκαταστήσει στη συσκευή του μια εφαρμογή τερματικού που να υποστηρίζει το Bluetooth πρωτόκολλο. Εφαρμογές αυτού του τύπου υπάρχουν πάρα πολλές στο Play Store και για την απόκτηση τους δεν απαιτείται κάποιο αντίτιμο. Εμείς χρησιμοποιήσαμε την BT Simple Terminal την οποία μπορεί κανείς να την προμηθευτεί από το [ακόλουθο url](https://play.google.com/store/apps/details?id=wingood.bluetooth.btsimpleterminal) :



Εικόνα 4: Η εφαρμογή Bluetooth Terminal στο Play Store.

Μέσω της εφαρμογής αυτής ο χρήστης αφού πρώτα συνδεθεί με το ολοκληρωμένο θα στείλει την εντολή ATZ με την οποία το ολοκληρωμένο θα κάνει μια επαναφορά στις εργοστασιακές ρυθμίσεις. Ακολούθως θα στείλει την εντολή SP 0 ώστε το ολοκληρωμένο να αναζητήσει πιο πρωτόκολλο επικοινωνίας υποστηρίζει το ενσωματωμένο σύστημα διάγνωσης του οχήματος. Μόλις το ολοκληρωμένο εντοπίσει το πρωτόκολλο θα το αποθηκεύσει και αυτό θα χρησιμοποιεί για να επικοινωνεί. Στη περίπτωση που το ολοκληρωμένο συνδεθεί σε διαφορετικό όχημα θα αναζητήσει από μόνο του το νέο πρωτόκολλο επικοινωνίας με το όχημα.

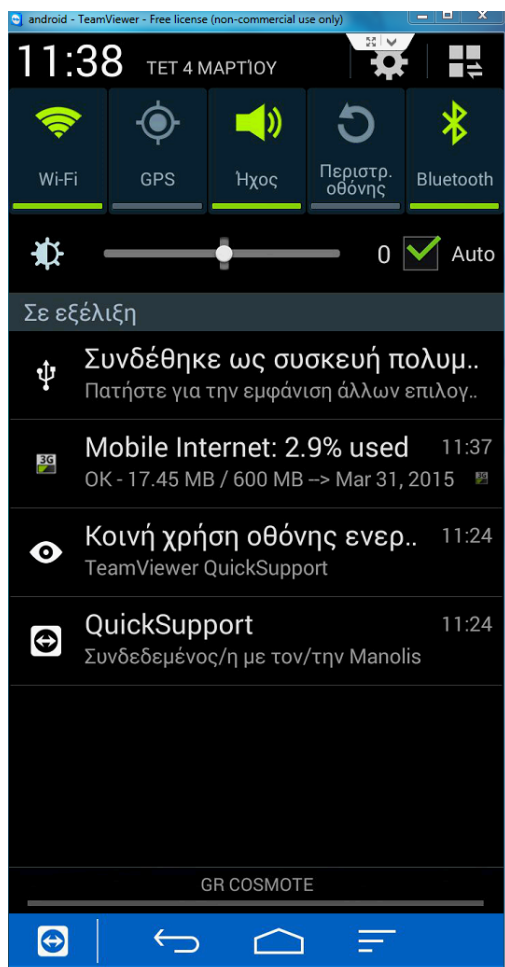
Τις άνω ρυθμίσεις ο χρήστης θα την εκτελέσει μόνο μία φορά. Η συσκευή θα 'θυμάται' τις ρυθμίσεις αυτές για όσο διάστημα χρησιμοποιείται. Μοναδική εξαίρεση όταν ο χρήστης χρησιμοποιήσει άλλη εφαρμογή τύπου Torque η οποία θα διαγράψει τις προηγούμενες ρυθμίσεις και θα εισάγει τις δικές της. Στη περίπτωση αυτή ο χρήστης θα πρέπει να επαναλάβει τις δύο εντολές για να μπορέσει να λειτουργήσει την εφαρμογή μας.



Εικόνα 5: Ρύθμιση του ολοκληρωμένου προκειμένου να συνεργαστεί με την εφαρμογή μας. Η διαδικασία γίνεται στον εξομοιωτή για λόγους ευκολίας. Διακρίνονται οι δύο εντολές που του στέλνουμε και οι απαντήσεις του. Στην εντολή ATZ (Reset) το ολοκληρωμένο απαντά τον τύπο του και κλείνει το μήνυμα με το χαρακτήρα '>' ενώ στην SP 0 που ορίζει το πρωτόκολλο επικοινωνίας απαντά με '?' λόγω του ότι απευθυνόμαστε στον εξομοιωτή ο οποίος δεν γνωρίζει την εντολή. Αντίθετα το ολοκληρωμένο θα απαντούσε 'OK'.

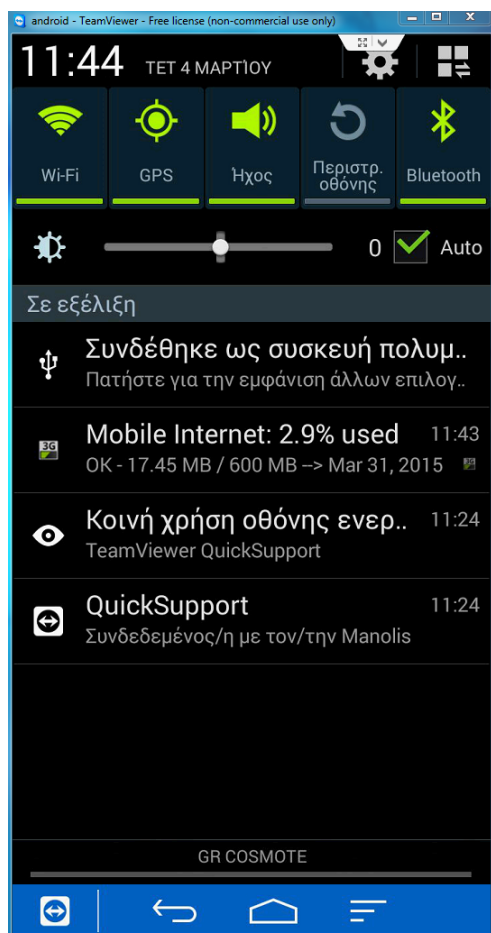
1.4.3 Διαμόρφωση Της Συσσκευής Πριν Τη Χρήση Της Εφαρμογής.

Προτού αρχίσει η εκτέλεση της εφαρμογής θα πρέπει η συσκευή να διαμορφωθεί ώστε να μπορεί να λειτουργήσει απρόσκοπτα η εφαρμογή μας. Σαν πρώτη ενέργεια θα πρέπει η χρήστης να απενεργοποιήσει την περιστροφή της οθόνης. Αυτό κρίνεται απαραίτητο μιας και όταν η οθόνη στο λειτουργικό σύστημα Android περιστρέφεται η εφαρμογή επανακινείται. Η ενέργεια αυτή έχει σαν αποτέλεσμα τη μη κανονική λειτουργία της εφαρμογής και για το λόγο αυτό η λειτουργικότητα αυτή θα πρέπει να απενεργοποιηθεί.



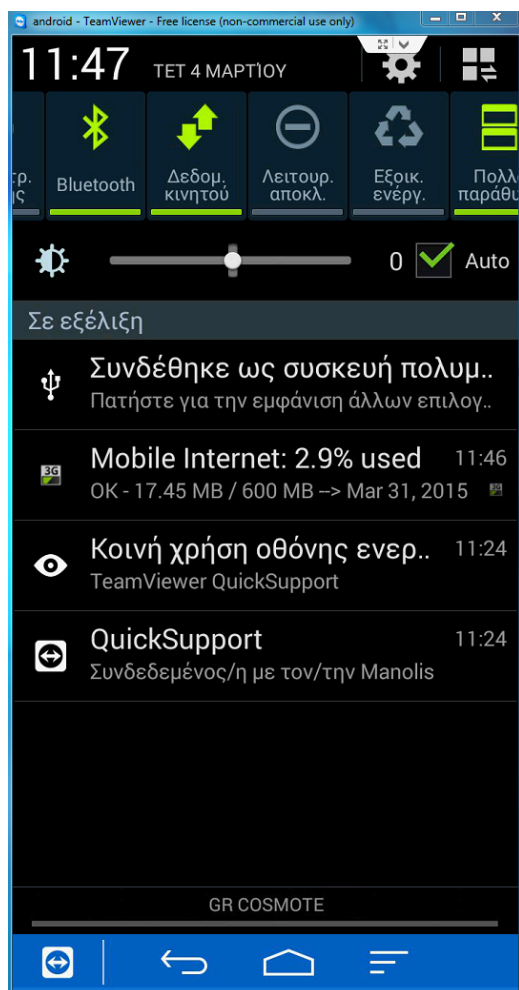
Εικόνα 6: Απενεργοποίηση περιστροφής οθόνης.

Επόμενο βήμα στη διαμόρφωση της συσκευής είναι η ενεργοποίηση του αισθητήρα GPS. Οι συντεταγμένες που παρέχει ο αισθητήρας αυτός είναι απαραίτητες για την καταγραφή της διαδρομής και χωρίς αυτές η εφαρμογή δεν μπορεί να λειτουργήσει.



Εικόνα 7: Ενεργοποίηση του αισθητήρα GPS.

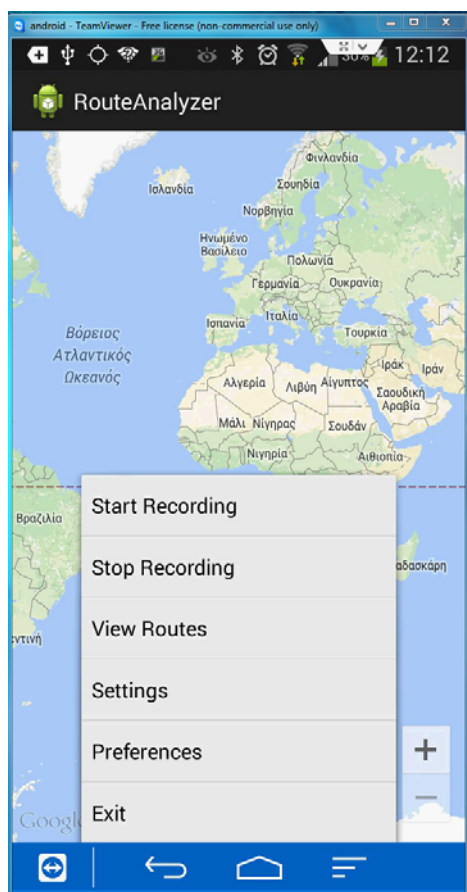
Τέλος, η συσκευή θα πρέπει να διαθέτει σύνδεση με το Internet προκειμένου να μεταφέρει τα δεδομένα του χάρτη της Google. Για το λόγο αυτό ο χρήστης θα πρέπει να ενεργοποιήσει τα δεδομένα στη συσκευή.



Εικόνα 8: Η ενεργοποίηση των δεδομένων κινητού στη συσκευή.

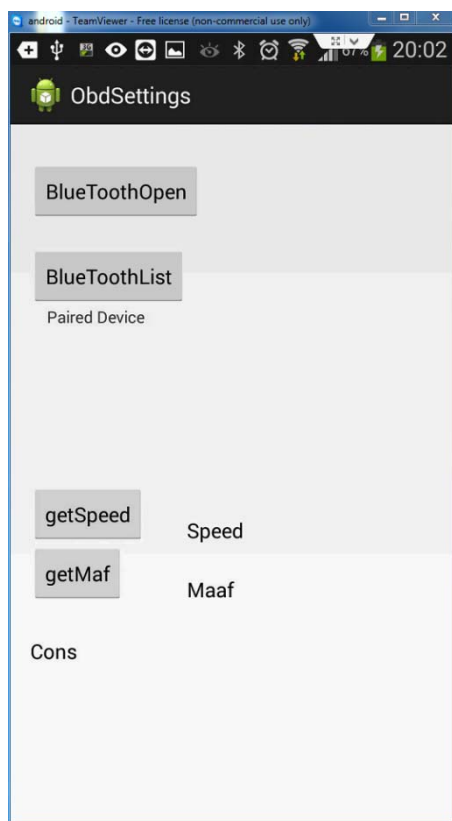
1.4.4 Εγκαθίδρυση Σύνδεσης Εφαρμογής Με Το Ολοκληρωμένο.

Προτού ο χρήστης αρχίσει την καταγραφή της διαδρομής του θα πρέπει να ενεργοποιήσει την επικοινωνία με το ολοκληρωμένο κύκλωμα ώστε η εφαρμογή να έχει πρόσβαση στα δεδομένα ταχύτητας και κατανάλωσης καυσίμου του οχήματος. Για το λόγο αυτό, αφού αγγίξει το εικονίδιο της εφαρμογής προκειμένου να την εκκινήσει, θα πρέπει να επιλέξει από το μενού την επιλογή 'Settings'.

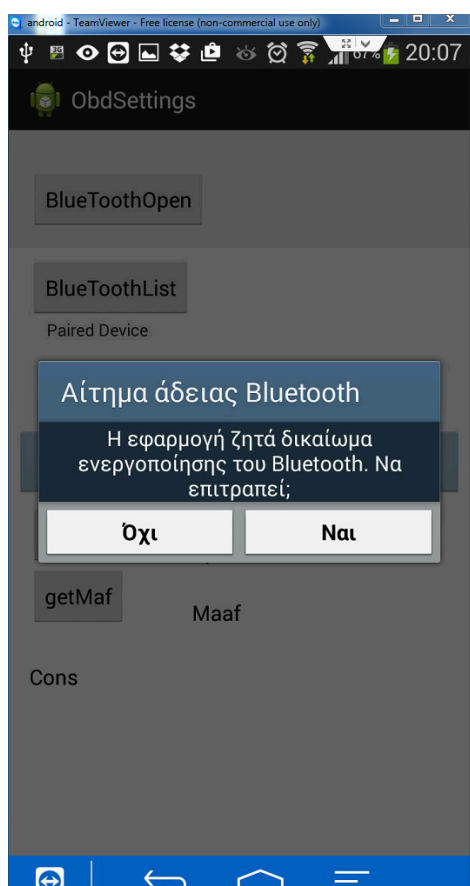


Εικόνα 9: Επιλογή του 'Settings' προκειμένου ο χρήστης να εγκαθιδρύσει τη σύνδεση με το ολοκληρωμένο.

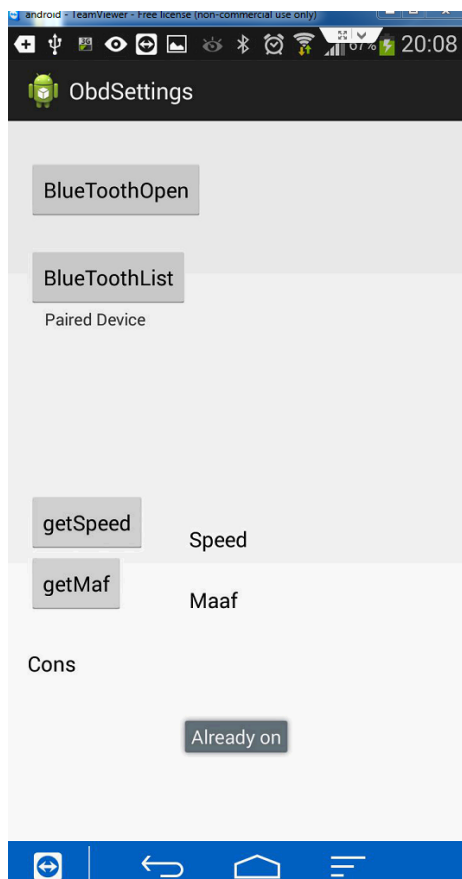
Μόλις ο χρήστης πατήσει την επιλογή η συσκευή εμφανίζει την οθόνη διασύνδεσης με το ολοκληρωμένο. Από την οθόνη αυτή ο χρήστης μπορεί να ανοίξει τον προσαρμογέα Bluetooth πατώντας στο κουμπί BlueToothOpen. Στη περίπτωση που το Bluetooth είναι ανοικτό η εφαρμογή θα εμφανίσει ένα μήνυμα ενημερώνοντας το χρήστη ότι ο προσαρμογέας είναι ανοικτός. Στην περίπτωση που δεν είναι θα τον ενημερώσει ότι η εφαρμογή ζητά να ενεργοποιήσει το προσαρμογέα και θα περιμένει το χρήστη να πατήσει το κουμπί 'Ναι' ώστε να τον ενεργοποιήσει:



Εικόνα 10: Η οθόνη σύνδεσης και εγκατάστασης της επικοινωνίας με το ολοκληρωμένο κύκλωμα.

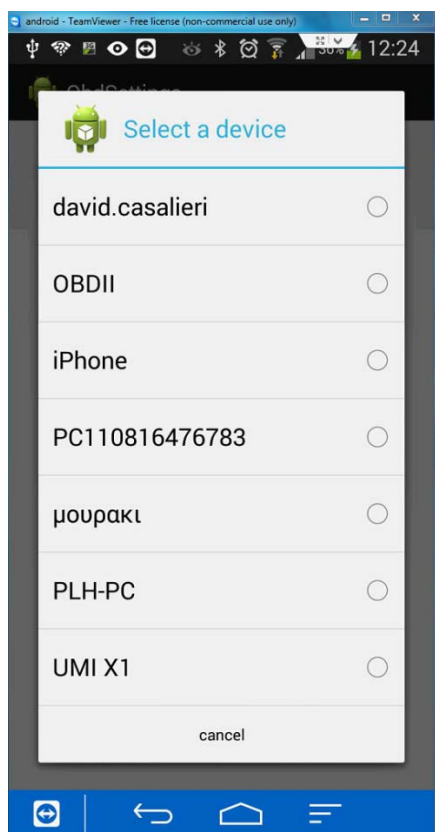


Εικόνα 11: Ερώτηση στο χρήστη αν επιτρέπει στην εφαρμογή να ενεργοποιήσει το Bluetooth.



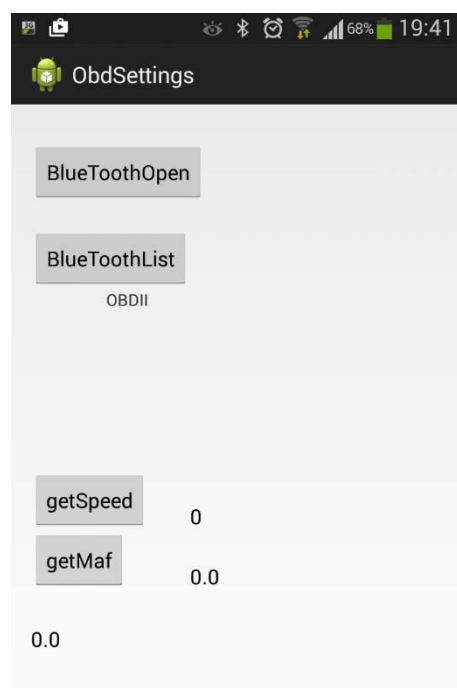
Εικόνα 12: Στη περίπτωση που το Bluetooth είναι ήδη ανοικτό η συσκευή τον ενημερώνει.

Αφού ο χρήστης ενεργοποιήσει τον προσαρμογέα Bluetooth θα πρέπει να πατήσει το κουμπί BlueToothList, ώστε η εφαρμογή να του παρουσιάσει μια λίστα με τις ήδη αντιστοιχισμένες συσκευές ώστε να επιλέξει από αυτές να συνδεθεί με το ολοκληρωμένο κύκλωμα.



Εικόνα 13: Επιλέγοντας τη συσκευή OBDII η εφαρμογή θα συνδεθεί με το ολοκληρωμένο.

Μόλις ο χρήστης επιλέξει τη συσκευή η εφαρμογή θα συνδεθεί και κατόπιν θα εμφανίσει το όνομα της συσκευής με την οποία έχει συνδεθεί στη θέση Paired Devices. Τώρα η συσκευή είναι έτοιμη να επικοινωνήσει με την εφαρμογή μας.



Εικόνα 14: Το ολοκληρωμένο έχει συνδεθεί με τη συσκευή OBDII και η επικοινωνία έχει εγκατασταθεί.

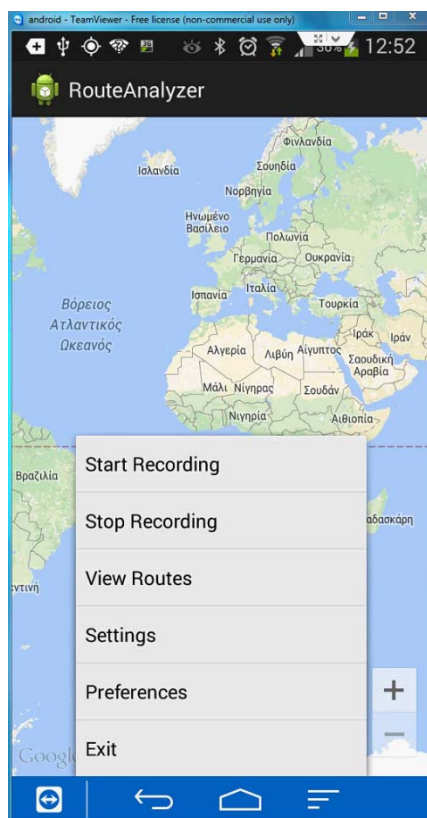
Στο σημείο αυτό ο χρήστης μπορεί να επιβεβαιώσει ότι η εφαρμογή επικοινωνεί σωστά με το ενσωματωμένο σύστημα διάγνωσης του αυτοκινήτου ζητώντας από αυτό τη ταχύτητα του οχήματος. Πατώντας το κουμπί `getSpeed` το κείμενο 'Speed' δεξιά του κουμπιού `getSpeed` θα πρέπει να αντικατασταθεί από τη τρέχουσα ταχύτητα του οχήματος. Στη περίπτωση που αυτό δεν συμβεί και η εφαρμογή εμφανίσει κάποιο μήνυμα αυτό σημαίνει ότι υπάρχει πρόβλημα στην επικοινωνία του ολοκληρωμένου με το όχημα ή στην επικοινωνία του κινητού τηλεφώνου με το ολοκληρωμένο. Για να αντιμετωπίσει το πρόβλημα ο χρήστης θα πρέπει να επαναφέρει το διακόπτη της μίζας στη θέση κλειστός, να αποσυνδέσει το ολοκληρωμένο από το όχημα και έπειτα να το συνδέσει εκ νέου. Ακολουθώντας να επαναφέρει το διακόπτη της μίζας στη θέση ανοικτός χωρίς όμως να θέσει σε λειτουργία το κινητήρα και να επανασυνδέσει την εφαρμογή με το ολοκληρωμένο. Οι ενέργειες αυτές θα επανεκινήσουν το ολοκληρωμένο και τώρα θα είναι σε θέση να επικοινωνήσει σωστά με το όχημα.

Τέλος πατώντας το πλήκτρο της συσκευής 'Πίσω' η εφαρμογή επιστρέφει στην αρχική οθόνη. Από το σημείο αυτό ο χρήστης μπορεί να ξεκινήσει τη καταγραφή της διαδρομής του είτε να εισάγει τις προτιμήσεις του.

1.4.5 Εισαγωγή Προτιμήσεων Του Χρήστη.

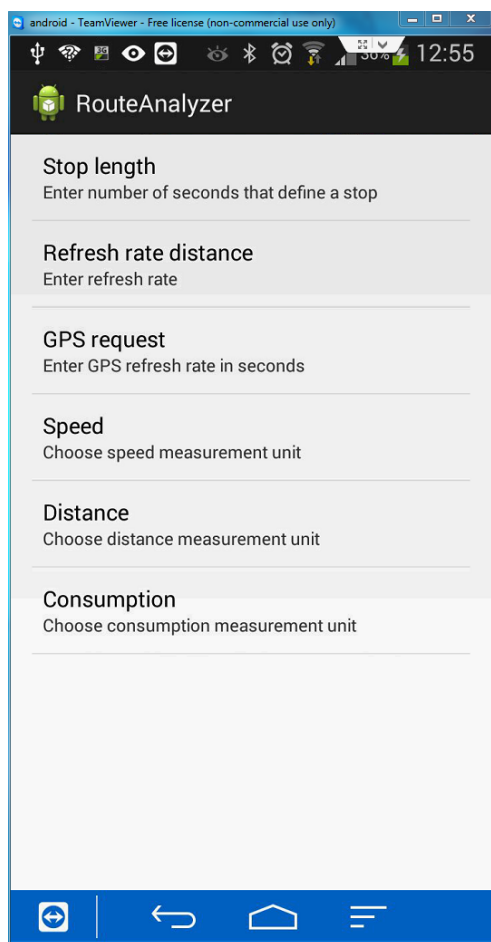
Έχοντας ολοκληρώσει τα άνω βήματα η εφαρμογή είναι έτοιμη να αρχίσει την καταγραφή διαδρομών. Στο σημείο όμως αυτό κρίνεται σκόπιμη η αναφορά στην δυνατότητα του χρήστη να εισάγει τις δικές του προτιμήσεις στην εφαρμογή ώστε να ανταποκρίνεται πλήρως στις απαιτήσεις του και να λειτουργεί σύμφωνα με τις δικές του ανάγκες. Η εφαρμογή μπορεί να λειτουργήσει και χωρίς την εισαγωγή των προτιμήσεων αυτών πλην όμως συνιστάτε να γίνει για καλύτερα αποτελέσματα.

Οι προτιμήσεις που μπορεί να εισάγει ο χρήστης στην εφαρμογή ενεργοποιούνται από το μενού επιλέγοντας την εγγραφή 'Preferences':



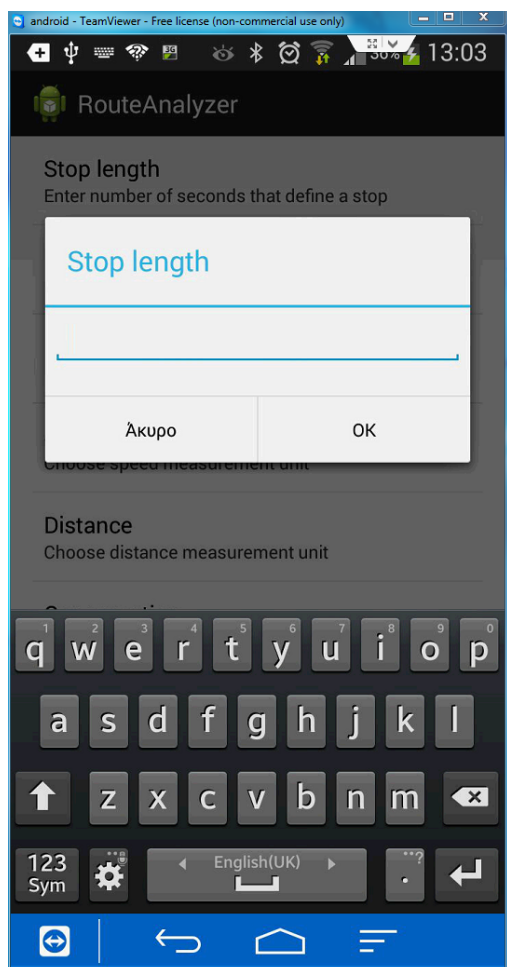
Εικόνα 15: Επιλέγοντας 'Preferences' από το μενού ο χρήστης εισάγει τις προτιμήσεις του.

Όταν ο χρήστης αγγίξει την συγκεκριμένη επιλογή η οθόνη εμφανίζει τις τιμές στις οποίες ο χρήστης καλείται να εισάγει τα επιθυμητά δεδομένα:



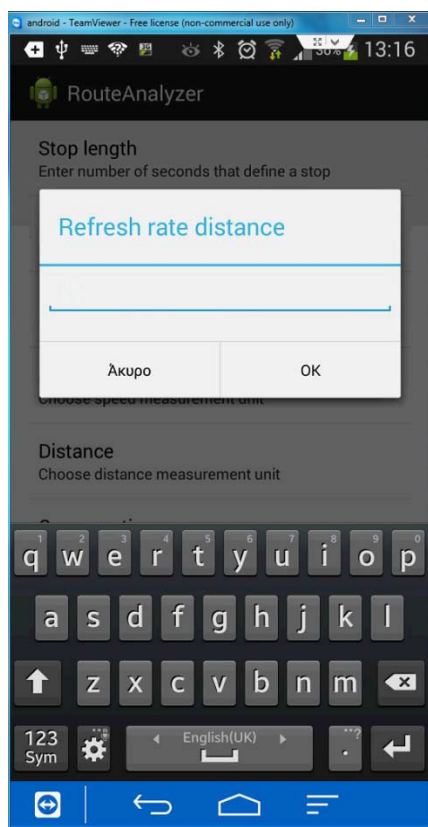
Εικόνα 16: Οθόνη εισαγωγής προτιμήσεων του χρήστη.

Η πρώτη επιλογή που εμφανίζεται στη λίστα είναι ο ορισμός του αριθμού των δευτερολέπτων που θα ορίζουν μια στάση. Πατώντας πάνω στην εγγραφή η συσκευή θα εμφανίσει ένα παράθυρο στο οποίο θα ζητά από το χρήστη να εισάγει την τιμή αυτή. Κατόπιν όταν θα αρχίσει η καταγραφή της διαδρομής η εφαρμογή θα περιμένει το χρήστη να σταματήσει να κινείται. Όταν το χρονικό διάστημα της στάσης υπερβεί την τιμή αυτή η εφαρμογή θα θεωρήσει ότι ο χρήστης έχει ολοκληρώσει μια υποδιαδρομή. Η προ εισαγμένη τιμή έχει ορισθεί στα 50 δευτερόλεπτα.



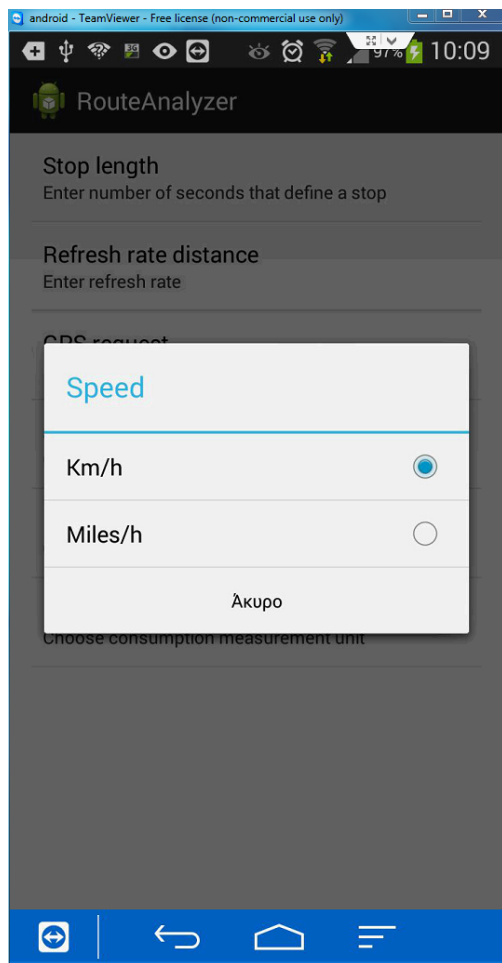
Εικόνα 17: Μόλις ο χρήστης αγγίξει την επιλογή η εφαρμογή θα εμφανίσει το παράθυρο διαλόγου και το πληκτρολόγιο. Μόλις ο χρήστης ολοκληρώσει την εισαγωγή του επιλέγει το κουμπί 'OK'.

Η επόμενη επιλογή αντιστοιχεί στην απόσταση σε μέτρα που θα πρέπει να μετακινηθεί η συσκευή ώστε να θεωρήσει ότι βρίσκεται σε κίνηση. Από την τιμή αυτή εξαρτάται και ο ρυθμός αποθήκευσης δεδομένων στη βάση δεδομένων της συσκευής. Η δυνατότητα αυτή κρίνεται ιδιαίτερα χρήσιμη στη λεπτομέρεια της καταγραφείσας διαδρομής αλλά και στη διαχείριση των πόρων του συστήματος. Για παράδειγμα όταν ο χρήστης κινείται σε εθνική οδό με μεγάλη ταχύτητα θα μπορεί να εισάγει μια σχετικά μεγάλη τιμή, 1000 μέτρα για παράδειγμα αφού η διαδρομή του θα καλύψει μεγάλη χιλιομετρική απόσταση χωρίς να στρίβει σε δρόμους. Έτσι η εφαρμογή εννοώντας τα αραιά σημεία της διαδρομής θα μπορεί να εμφανίσει την ακριβή διαδρομή του. Αντίθετα αυτό δεν μπορεί να γίνει όταν κινείται εντός πόλης με μικρή ταχύτητα αφού η σύνδεση των αραιών σημείων θα έχει σαν συνέπεια την παράληψη τμημάτων της διαδρομής μιας και τα σημεία της διαδρομής συνδέονται με μια ευθεία γραμμή. Η προ εισαγμένη τιμή είναι στα 5 μέτρα.



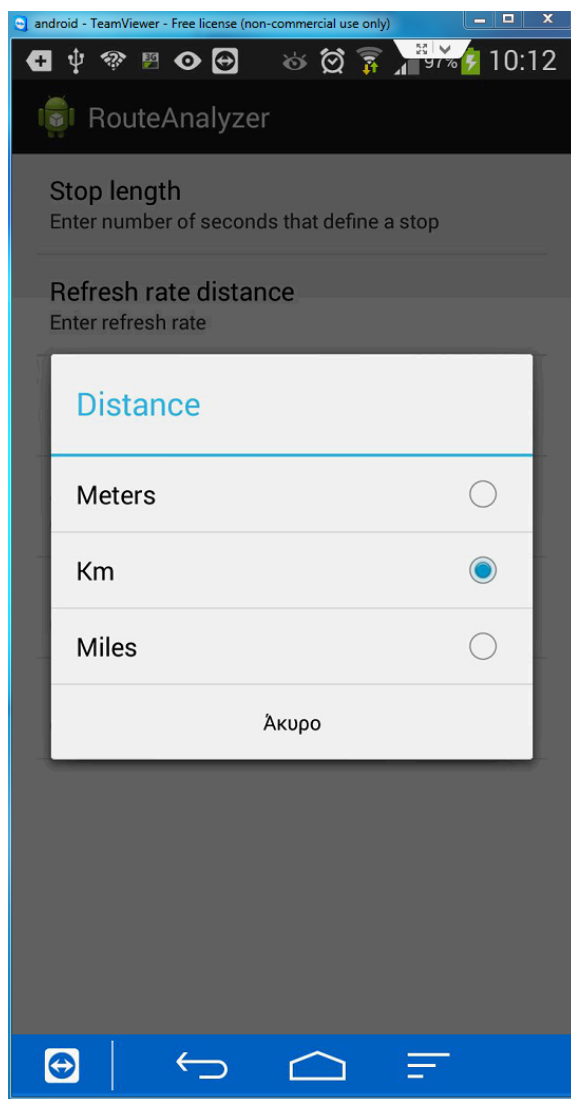
Εικόνα 18: Εισαγωγή της απόστασης σε μέτρα που θα πρέπει να μετακινηθεί η συσκευή ώστε να θεωρήσει ότι βρίσκεται σε κίνηση.

Στις επόμενες τρεις επιλογές ο χρήστης ορίζει τις μονάδες μέτρησης των διάφορων τιμών. Πιο συγκεκριμένα πατώντας στην επιλογή 'Speed' ο χρήστης επιλέγει τη μονάδα μέτρησης της ταχύτητας του οχήματος. Η προ εισαγμένη τιμή είναι χιλιόμετρα ανά ώρα.



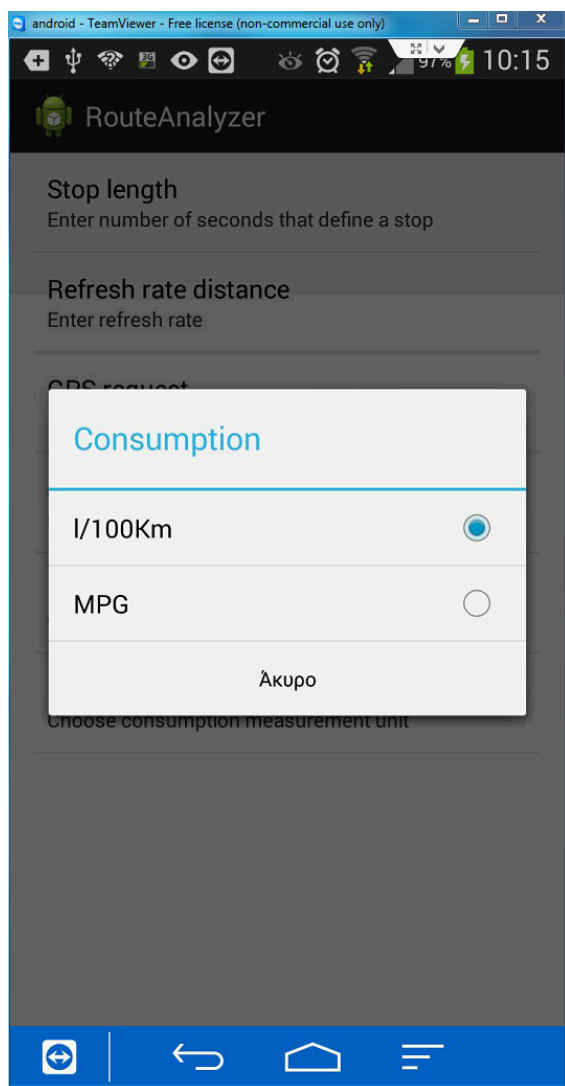
Εικόνα 19: Επιλογή μονάδας μέτρησης ταχύτητας του οχήματος.

Ακολούθως, πατώντας την επιλογή 'Distance' ο χρήστης ορίζει τη μονάδα μέτρησης της απόστασης. Η προ εισαγμένη τιμή είναι χιλιόμετρα.



Εικόνα 20: Εισαγωγή μονάδας μέτρησης της απόστασης.

Τέλος πατώντας την επιλογή 'Consumption' ο χρήστης επιλέγει τη μονάδα μέτρησης κατανάλωσης καυσίμου του οχήματος. Η προ εισαγμένη τιμή είναι λίτρα ανά εκατό χιλιόμετρα.

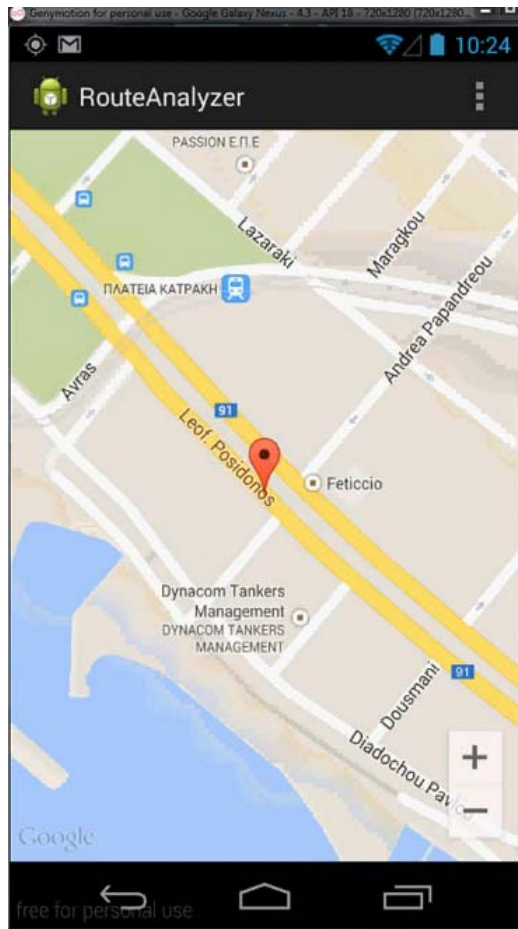


Εικόνα 21: Η εισαγωγή μονάδας μέτρησης κατανάλωσης καυσίμου του οχήματος.

Όταν ο χρήστης ολοκληρώσει την εισαγωγή των προτιμήσεων του αγγίζει το πλήκτρο της συσκευής 'πίσω' και επιστρέφει στην αρχική οθόνη της συσκευής.

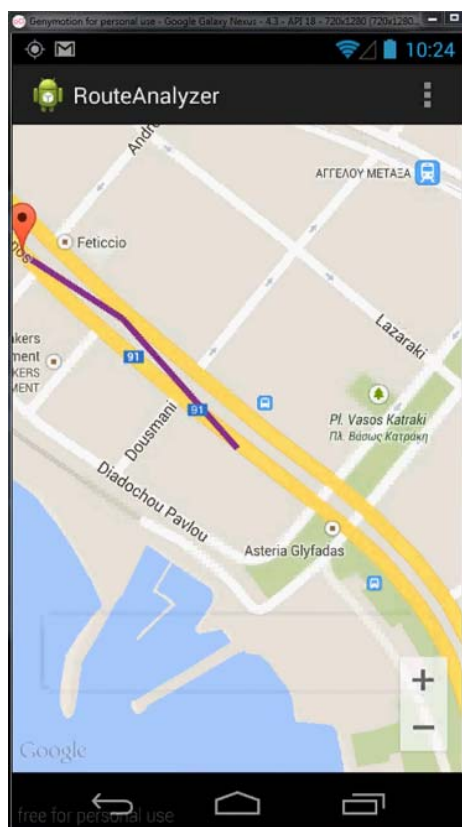
1.4.6 Καταγραφή Της Διαδρομής.

Έχοντας ολοκληρώσει τις ρυθμίσεις του ολοκληρωμένου και της συσκευής ο χρήστης μπορεί πλέον να αρχίσει την καταγραφή διαδρομών. Αυτό επιτυγχάνεται επιλέγοντας από το μενού της εφαρμογής το 'Start Recording'. Αμέσως η εφαρμογή θα εστιάσει στο χάρτη στην τρέχουσα τοποθεσία μας και θα τοποθετήσει μια πινέζα στο σημείο που βρισκόμαστε σηματοδοτώντας έτσι την έναρξη μια διαδρομής.



Εικόνα 22: Αποτέλεσμα της ενεργοποίησης του 'Start Recording' η εφαρμογή σηματοδοτεί την έναρξη καταγραφής διαδρομής.

Ακολούθως και για το χρονικό διάστημα που ο χρήστης κινείται η εφαρμογή σχεδιάζει τη διαδρομή του στο χάρτη.

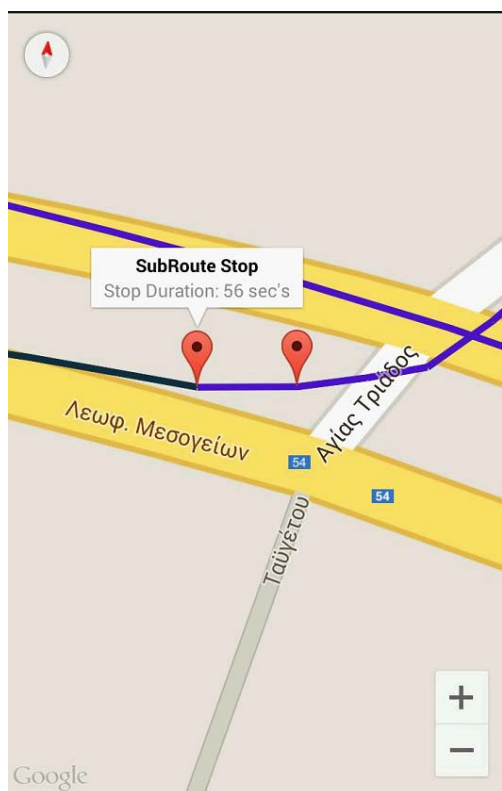


Εικόνα 23: Η αποτύπωση της διαδρομής στο χάρτη.

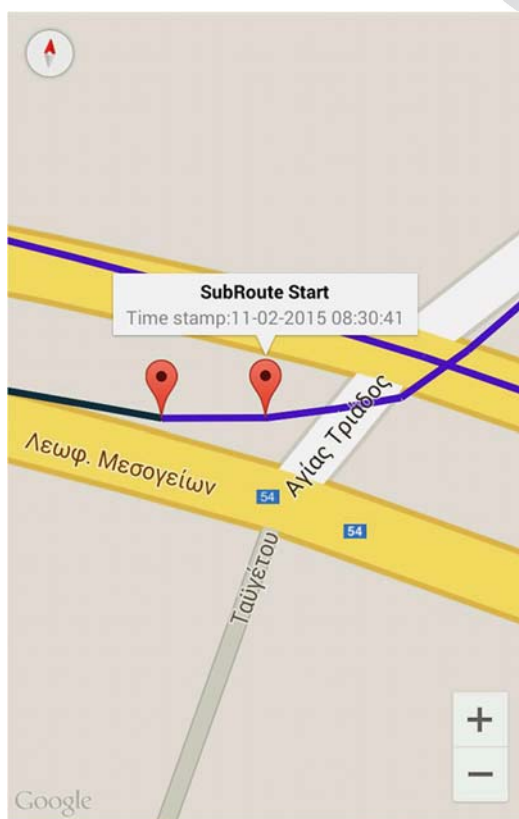
Η ακρίβεια της διαδρομής εξαρτάται από τον ρυθμό λήψης δεδομένων από το GPS καθώς και την ελάχιστη απόσταση που πρέπει να μετακινηθεί η συσκευή ώστε να θεωρήσει ότι βρίσκεται σε κίνηση. Στη συγκεκριμένη διαδρομή οι συντεταγμένες δόθηκαν μέσω λογισμικού εξομοίωσης του αισθητήρα GPS και συγκεκριμένα δόθηκαν τρεις τιμές: Μία στην έναρξη της διαδρομής όπου βρίσκεται η πινέζα, μία στο μέσο της διαδρομής, λίγο πριν την οδό Δουσμάνη, στο σημείο που η διαδρομή κάνει τη 'γωνία' και μία στο τέλος. Δίδοντας στις προτιμήσεις μικρότερη απόσταση η υψηλότερο ρυθμό λήψης δεδομένων από το GPS θα είχε ως αποτέλεσμα περισσότερα σημεία και άρα μεγαλύτερη ακρίβεια στη καταγραφή της διαδρομής.

Η εκτέλεση της εφαρμογής συνεχίζεται με τον ίδιο τρόπο όσο ο χρήστης δεν πάψει να κινείται για χρονικό διάστημα μικρότερο από αυτό που έχει ορίσει σαν στάση. Για παράδειγμα έστω ότι ο χρήστης έχει ορίσει το διάστημα αυτό στα είκοσι δευτερόλεπτα. Για όσο οι στάσεις που κάνει είναι διάρκειας μικρότερης των είκοσι δευτερολέπτων η εκτέλεση της εφαρμογής θα είναι ίδια με την παραπάνω.

Η εκτέλεση διαφοροποιείται όταν πραγματοποιήσει μια στάση χρονικής διάρκειας μεγαλύτερης. Μόλις αρχίσει να κινείται η εφαρμογή θεωρεί ότι έχει ολοκληρώσει μια υποδιαδρομή και ξεκινά μια νέα. Έτσι τοποθετεί μια πινέζα στο τέλος της ολοκληρωμένης διαδρομής στην οποία αναγράφει το χρονικό διάστημα της στάσης. Ακολούθως αρχίζει να σχεδιάζει μια νέα με νέο χρώμα τοποθετώντας στο σημείο έναρξης της μια νέα πινέζα στην οποία αναγράφει ότι αποτελεί νέα διαδρομή συμπληρώνοντας με την χρονική στιγμή έναρξης της υποδιαδρομής.

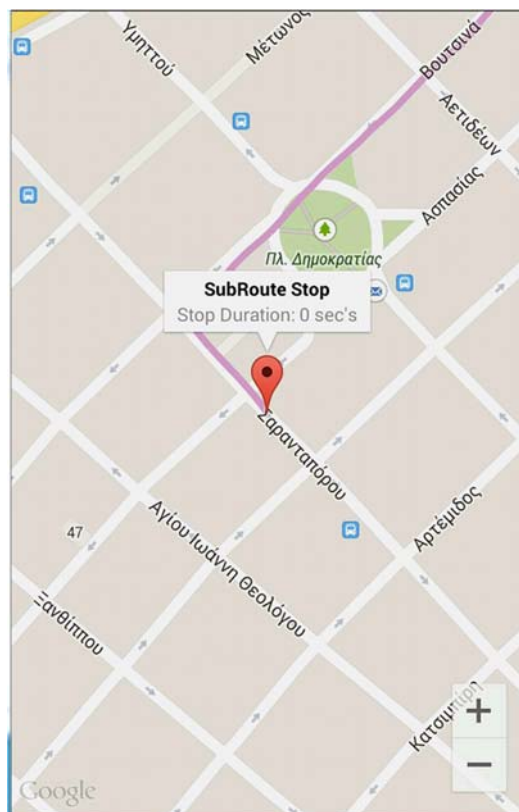


Εικόνα 24: Πινέζα στο σημείο όπου ο χρήστης έπαψε να κινείται για χρονικό διάστημα μεγαλύτερο αυτού της στάσης. Συγκεκριμένα ο χρήστης ήταν σε στάση για 56 δευτερόλεπτα.



Εικόνα 25: Μόλις ο χρήστης αρχίσει να κινείται εκ νέου η εφαρμογή αρχίζει να σχεδιάζει με νέο χρώμα τη νέα διαδρομή τοποθετώντας πινέζα στο επόμενο σημείο συντεταγμένων που λαμβάνει από το GPS μετά τη στάση.

Η εκτέλεση του προγράμματος συνεχίζεται με τον ίδιο τρόπο έως ότου ο χρήστης να ολοκληρώσει την διαδρομή του και να επιλέξει από το μενού το 'Stop Recording'. Τότε θα τοποθετήσει μια πινέζα στο σημείο αναφέροντας και το χρονικό διάστημα που ήταν σε ακινησία στο σημείο αυτό πριν πατήσει το 'Stop Recording'.

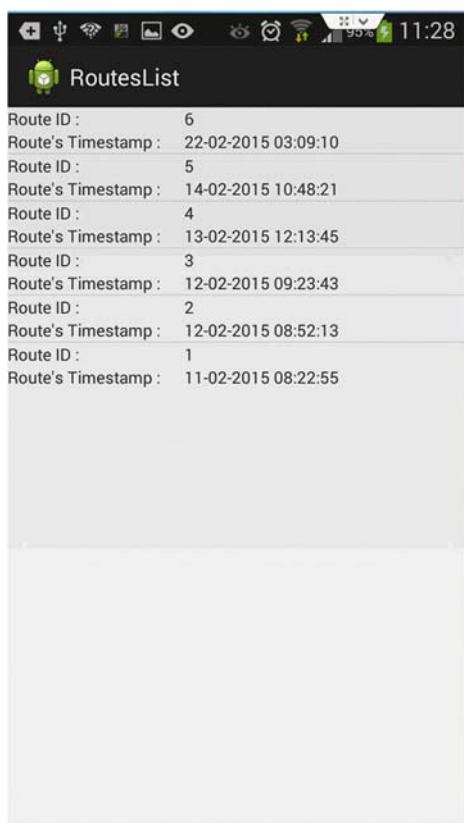


Εικόνα 26: Πατώντας το 'Stop Recording' η εφαρμογή τοποθετεί πινέζα στις τρέχουσες συντεταγμένες.

Από το σημείο αυτό ο χρήστης μπορεί να ξεκινήσει εκ νέου καταγραφή διαδρομής επιλέγοντας το 'Start Recording'. Η εφαρμογή θα λειτουργήσει με τον τρόπο που έχει περιγράψει, σχεδιάζοντας τη διαδρομή στο χάρτη, διατηρώντας ταυτόχρονα και τις παλιές διαδρομές. Αν δεν επιθυμεί την καταγραφή νέα διαδρομής μπορεί να επιλέξει να δει τις διαδρομές που έχει καταγράψει από την επιλογή του μενού 'View Routes'. Τέλος μπορεί να επιλέξει τον τερματισμό της εφαρμογής επιλέγοντας 'Exit' επίσης από το μενού.

1.4.7 Προβολή Καταγεγραμμένων Διαδρομών.

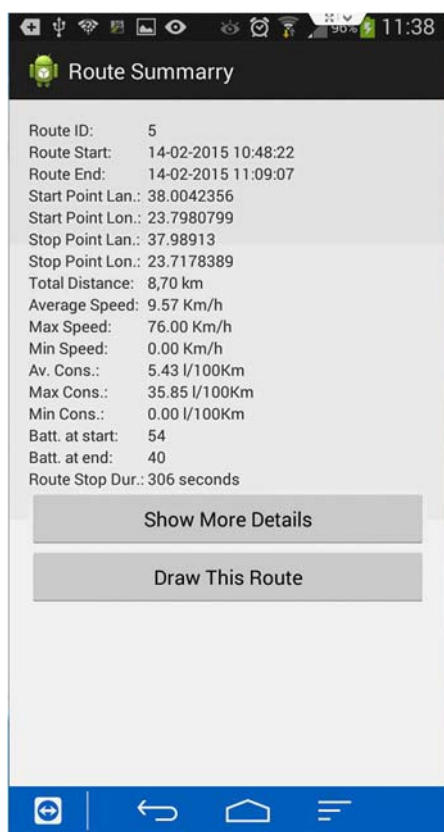
Για να μελετήσει τις καταγεγραμμένες διαδρομές του ο χρήστης επιλέγει από μενού το 'View Routes'. Η λειτουργία αυτή δεν απαιτεί σύνδεση με το ολοκληρωμένο ούτε δεδομένα από το GPS. Έτσι δίνεται η δυνατότητα στο χρήστη να εγκαταλείψει το όχημα και να εισέλθει σε στεγασμένο χώρο όπου δεν διατίθεται σήμα GPS για να επεξεργαστεί τα δεδομένα του είτε να τα παρουσιάσει σε τρίτους.



Εικόνα 27: Επιλέγοντας 'View Routes' η εφαρμογή προβάλλει τις καταγεγραμμένες διαδρομές.

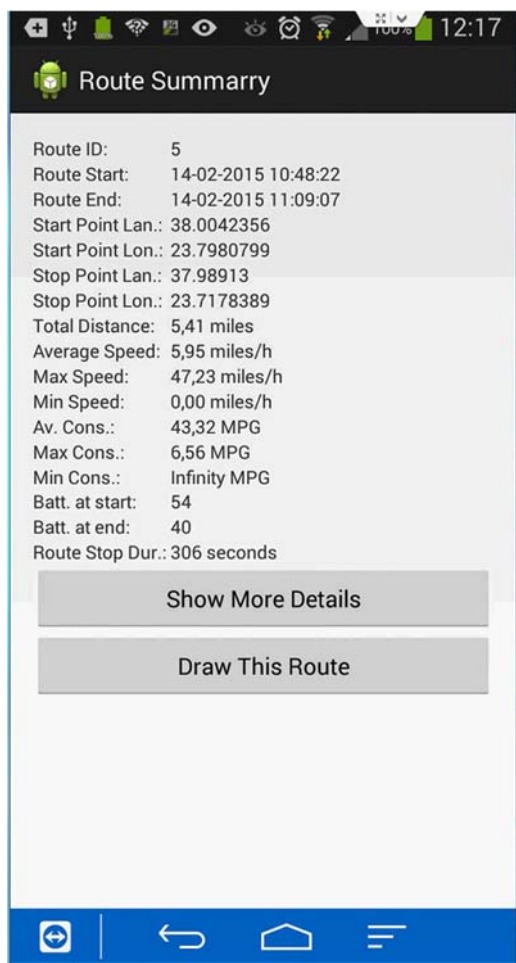
Οι διαδρομές εμφανίζονται ταξινομημένες αντίστροφα ως προς τη σειρά που αποθηκευτήκαν. Έτσι η πιο πρόσφατη διαδρομή εμφανίζεται πρώτη στη λίστα. Αριθμούνται ξεκινώντας από τον αριθμό 1 και ο αριθμός αυτός αποτελεί τον μοναδικό κωδικό διαδρομής, το επωνομαζόμενο Route ID. Το πεδίο Routes Timestamp αναφέρει τη χρονική στιγμή έναρξης της διαδρομής. Συγκεκριμένα η τελευταία διαδρομή με κωδικό 6 ξεκίνησε στις 22 Φεβρουαρίου του 2015 και ώρα 15:09:10.

Αγγίζοντας κάποια διαδρομή η εφαρμογή θα εμφανίσει τις λεπτομέρειες αυτής. Για παράδειγμα επιλέγοντας την διαδρομή με κωδικό 5 η εφαρμογή θα εμφανίσει την ακόλουθη οθόνη στην οποία αναγράφονται οι ακόλουθες τιμές:



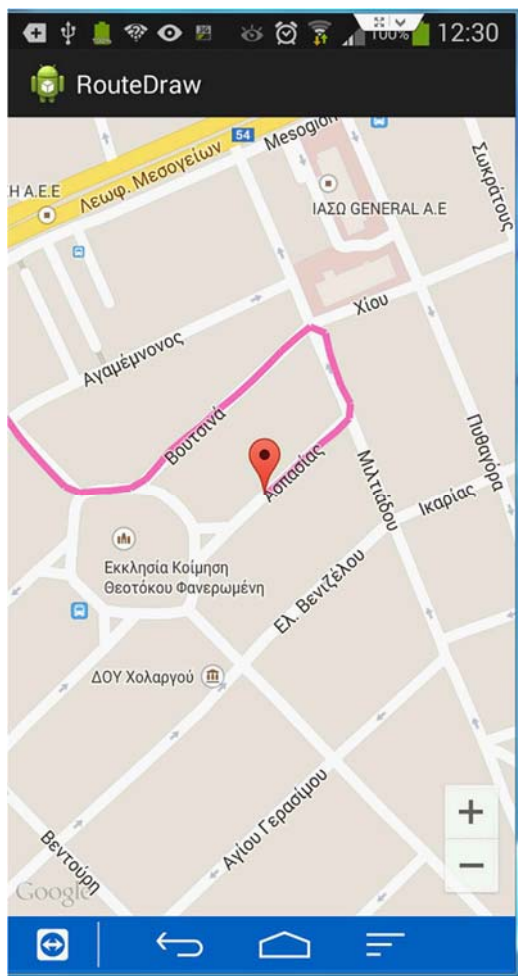
Εικόνα 28: Λεπτομέρειες της διαδρομής με κωδικό 5.

Ξεκινώντας από το πάνω μέρος της οθόνης ο χρήστης βλέπει ότι αυτή είναι η διαδρομή με κωδικό 5 η οποία ξεκίνησε στις 14 Φεβρουαρίου 2015 και ώρα 10:48:22 και ολοκληρώθηκε την ίδια μέρα και ώρα 11:09:07. Ακολούθως βλέπει τις συντεταγμένες του σημείου έναρξης και ακολούθως του σημείου λήξης. Στη συνέχεια αναγράφονται η συνολική απόσταση διαδρομής σε χιλιόμετρα και ο μέσος όρος ταχύτητας του οχήματος σε χιλιόμετρα ανά ώρα. Οι επόμενες δύο εγγραφές αναφέρουν την μέγιστη ταχύτητα που ανέπτυξε το όχημα και ακολούθως την ελάχιστη επίσης σε χιλιόμετρα ανά ώρα. Η μέση κατανάλωση καυσίμου του οχήματος αναγράφεται ακολούθως σε λίτρα ανά 100 χιλιόμετρα και στη συνέχεια η μέγιστη αλλά και η ελάχιστη κατανάλωση. Στη συνέχεια το επίπεδο της μπαταρίας της συσκευής κατά την έναρξη και ακολούθως κατά την ολοκλήρωση της διαδρομής. Τέλος η συνολική διάρκεια των στάσεων που σηματοδοτούν την ολοκλήρωση υποδιαδρομών σε δευτερόλεπτα. Οι τιμές των μονάδων μέτρησης αλλάζουν από την επιλογή Preferences του μενού.



Εικόνα 29: Προβολή λεπτομερειών με διαφορετικές μονάδες μέτρησης. Η τιμή 'Infinity MPG' στο πεδίο ελάχιστη κατανάλωση προκύπτει από την μηδενική κατανάλωση. Αφού το όχημα δεν καταναλώνει καθόλου καύσιμο θεωρητικά μπορεί να διανύσει άπειρα μίλια ανά γαλόνι.

Από την οθόνη αυτή ο χρήστης έχει δύο επιλογές μέσω των κουμπιών στο κάτω μέρος της οθόνης. Μπορεί είτε να σχεδιάσει τη διαδρομή στο χάρτη επιλέγοντας 'Draw This Route' είτε να δει τις λεπτομέρειες των υποδιαδρομών της διαδρομής αυτής επιλέγοντας το κουμπί 'Show More Details'. Στη πρώτη περίπτωση η εφαρμογή θα εμφανίσει το χάρτη εστιάζοντας στο σημείο έναρξης της διαδρομής. Τοποθετεί πινέζα στο σημείο αναγράφοντας ότι πρόκειται για σημείο έναρξης διαδρομής και συμπληρώνει με τη χρονική στιγμή έναρξης της διαδρομής.

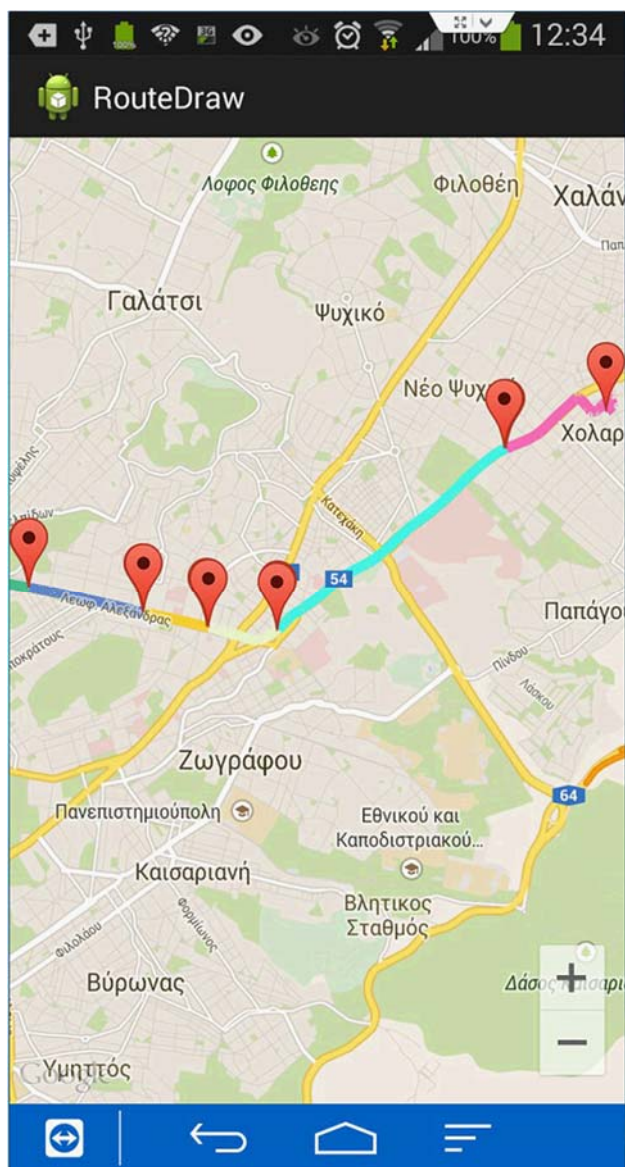


Εικόνα 30: Σχεδιασμός της διαδρομής.

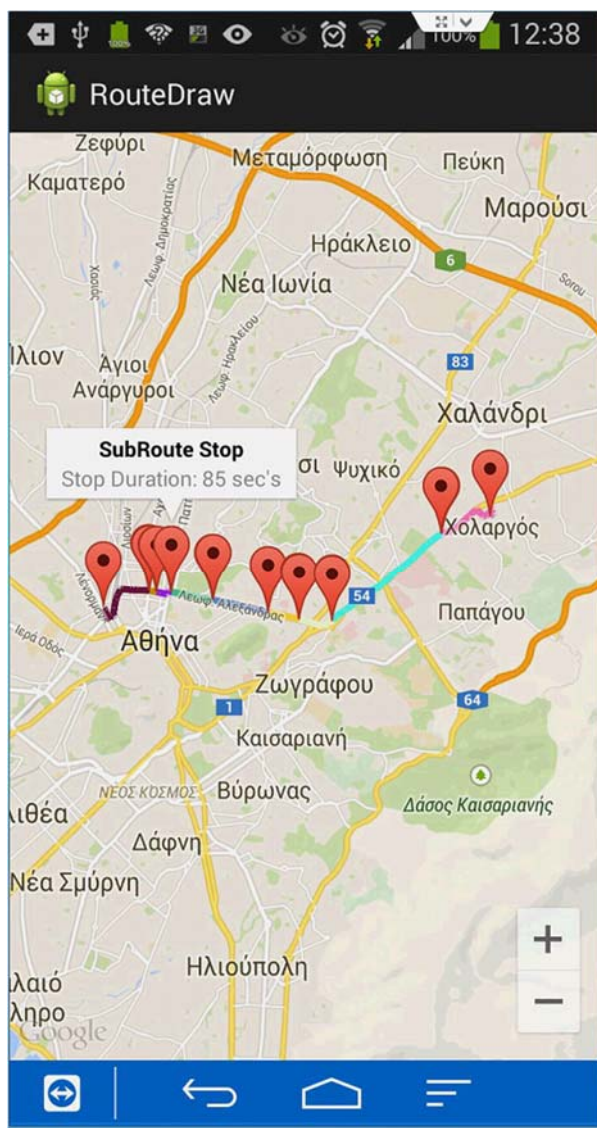


Εικόνα 31: Πατώντας πάνω στη πινέζα ο χρήστης βλέπει τη χρονική στιγμή έναρξης της διαδρομής.

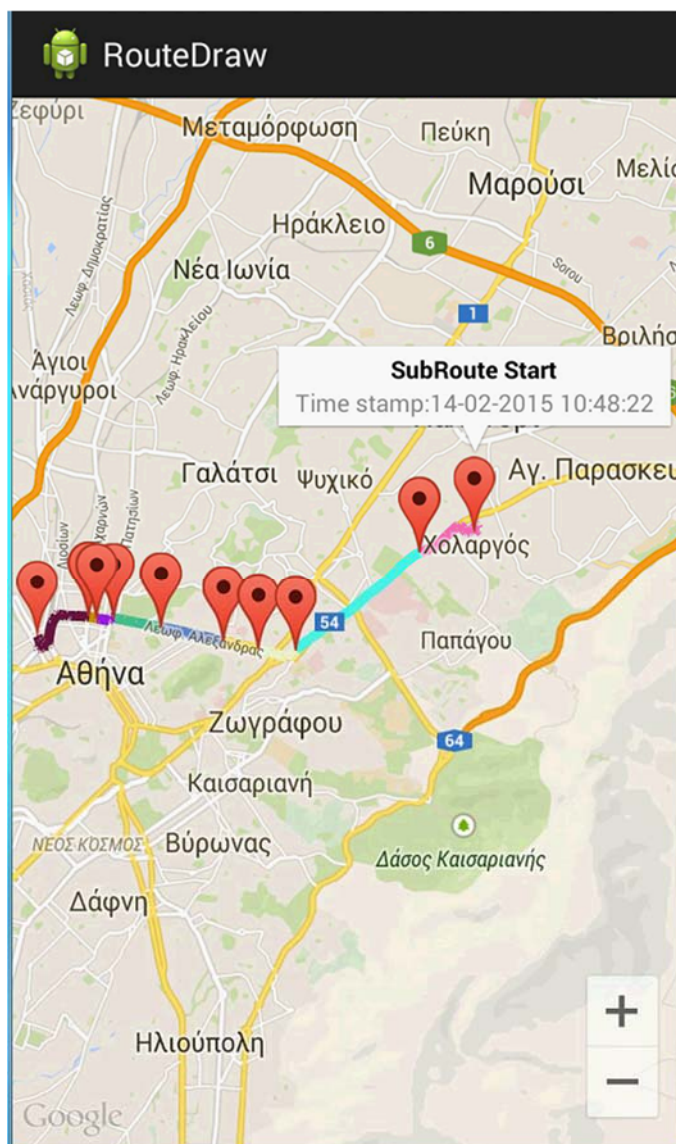
Αλλάζοντας το επίπεδο εστίασης στο χάρτη, πατώντας το μείον για σμίκρυνση ή το συν για μεγέθυνση ο χρήστης μπορεί να βλέπει τα διαφορετικά τμήματα της διαδρομής ή και ολόκληρη.



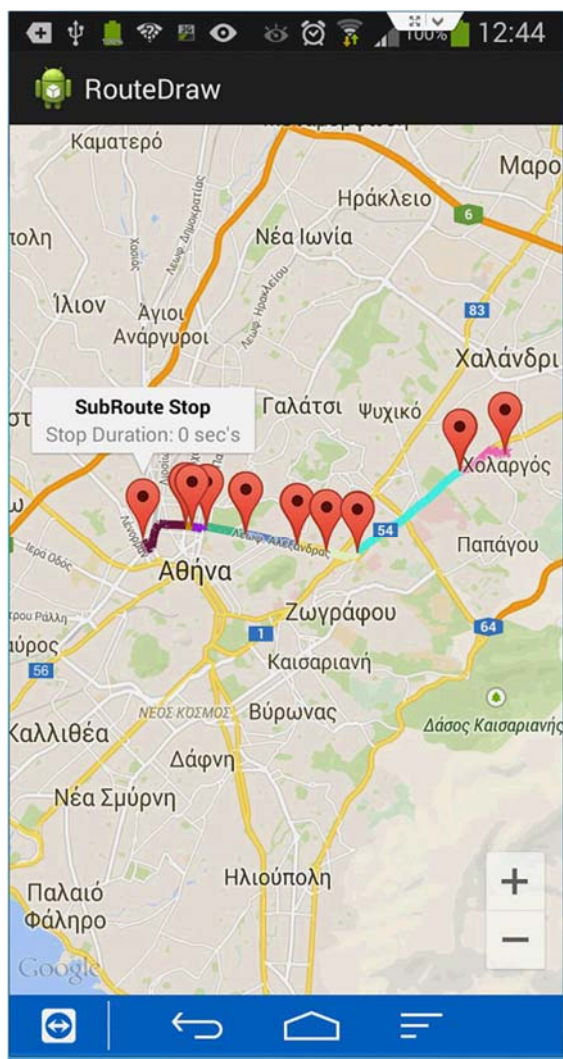
Εικόνα 32: Τμήμα της διαδρομής. Οι υποδιαδρομές εμφανίζονται με διαφορετικό χρώμα ενώ οι πινέζες σηματοδοτούν τα σημεία έναρξης και λήξης υποδιαδρομών.



Εικόνα 33: Το σύνολο της διαδρομής. Η συγκεκριμένη πινέζα απεικονίζει την ολοκλήρωση της υποδιαδρομής και η χρονική διάρκεια της στάσης στο σημείο ήταν 85 δευτερόλεπτα.



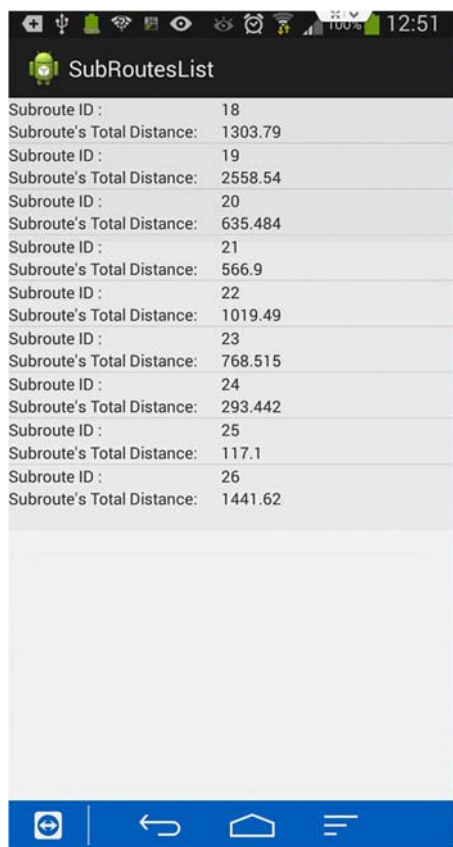
Εικόνα 34: Το σημείο έναρξης της διαδρομής



Εικόνα 35: Το σημείο ολοκλήρωσης της διαδρομής.

Με τον τρόπο αυτό ο χρήστης μπορεί να εστιάζει στα σημεία ενδιαφέροντος και να εντοπίσει την ακριβή διαδρομή του οχήματος, που πραγματοποιήθηκε στάση καθώς και για πόσο χρόνο.

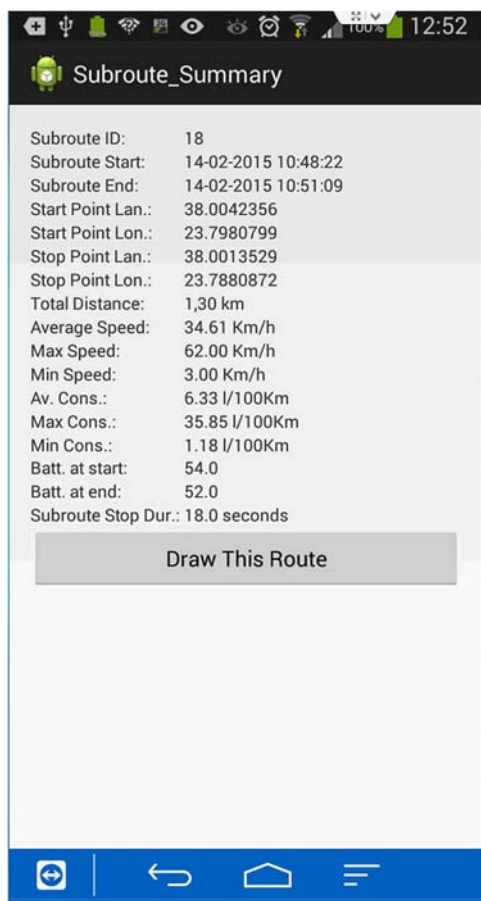
Επιλέγοντας το 'Show More Details' ο χρήστης βλέπει τις υποδιαδρομές της συγκεκριμένης διαδρομής.



Subroute ID	Subroute's Total Distance
18	1303.79
19	2558.54
20	635.484
21	566.9
22	1019.49
23	768.515
24	293.442
25	117.1
26	1441.62

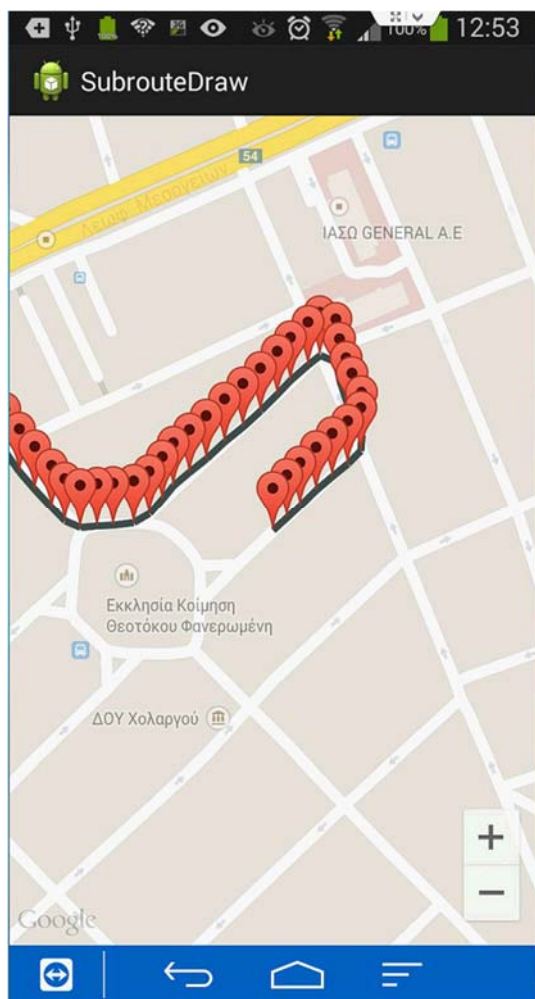
Εικόνα 36: Το σύνολο των υποδιαδρομών της συγκεκριμένης διαδρομής.

Εδώ ο χρήστης βλέπει τις επιμέρους υποδιαδρομές της διαδρομής ταξινομημένες με τη σειρά που πραγματοποιήθηκαν. Οι υποδιαδρομές αριθμούνται στο σύνολο τους με μοναδικό κωδικό υποδιαδρομής, το πεδίο Subroute ID, ενώ το πεδίο Subroute's Total Distance αναγράφει τη συνολική απόστασή της διαδρομής σε μέτρα. Αγγίζοντας μια διαδρομή η εφαρμογή εμφανίζει τις τιμές της υποδιαδρομής όπως και στη περίπτωση της διαδρομής.



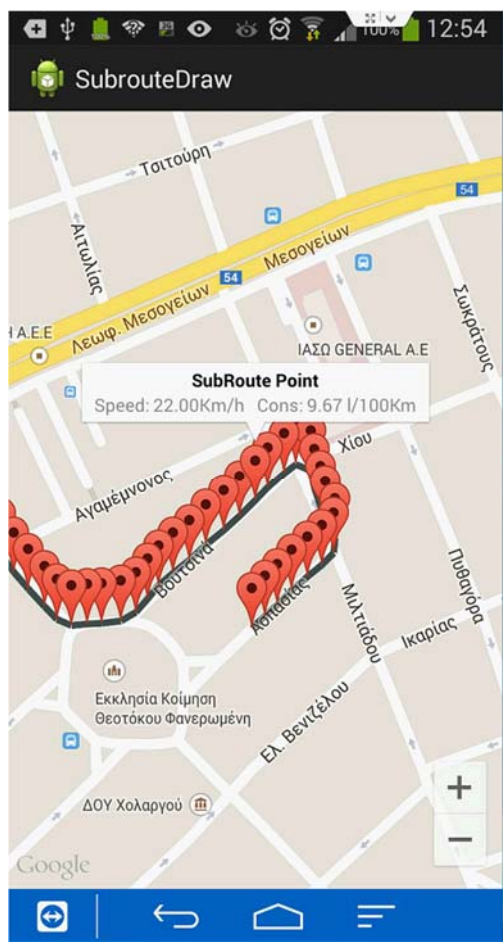
Εικόνα 37: Λεπτομέρειες της υποδιαδρομής.

Όπως και στη περίπτωση της διαδρομής έτσι και εδώ ο χρήστης βλέπει το κωδικό αριθμό της υποδιαδρομής, τη χρονική στιγμή έναρξης και ολοκλήρωσης αυτής, τις συντεταγμένες εκκίνησης και λήξης, τη συνολική απόσταση, τη μέση και ακολούθως τη μέγιστη και ελάχιστη ταχύτητα που ανέπτυξε το όχημα. Ακολούθως τη μέση κατανάλωση καυσίμου του οχήματος και στη συνέχεια τη μέγιστη αλλά και ελάχιστη. Οι επόμενες τιμές αφορούν το επίπεδο της μπαταρίας κατά την έναρξη και ακολούθως κατά την ολοκλήρωση της υποδιαδρομής και τέλος η διάρκεια σε δευτερόλεπτα που το όχημα βρισκόταν σε στάση. Οι τιμές των μονάδων μέτρησης ορίζονται από τα 'Preferences'. Τέλος επιλέγοντας το κουμπί Draw This Route η εφαρμογή ζωγραφίζει τα σημεία της υποδιαδρομής στο χάρτη εστιάζοντας στο σημείο εκκίνησης.

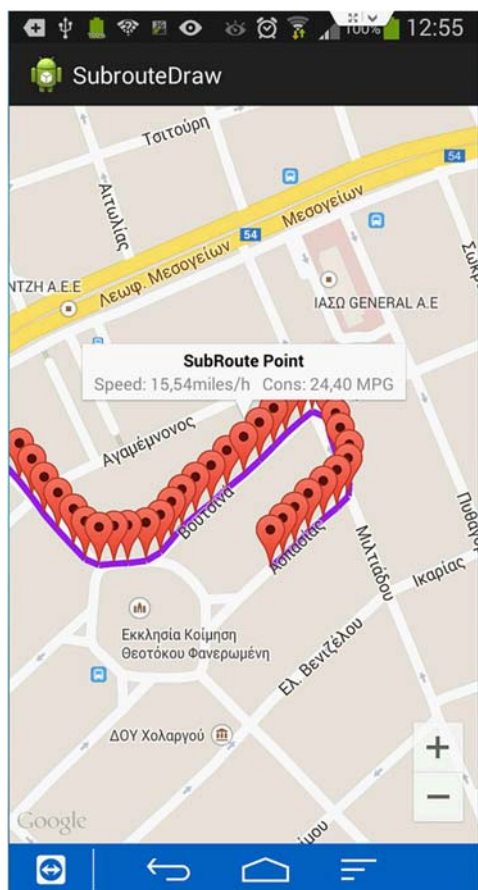


Εικόνα 38: Ο σχεδιασμός της υποδιαδρομής στο χάρτη.

Πατώντας πάνω στις πινέζες ο χρήστης ενημερώνεται για τη ταχύτητα και τη κατανάλωση του οχήματος στο σημείο αυτό. Στο παράδειγμά μας το κατώφλι απόστασης είχε οριστεί στα 10 μέτρα ενώ ο ρυθμός ανάγνωσης του GPS στα 5 δευτερόλεπτα. Για το λόγο αυτό η διαδρομή είναι αρκετά ακριβής με αρκετά σημεία.

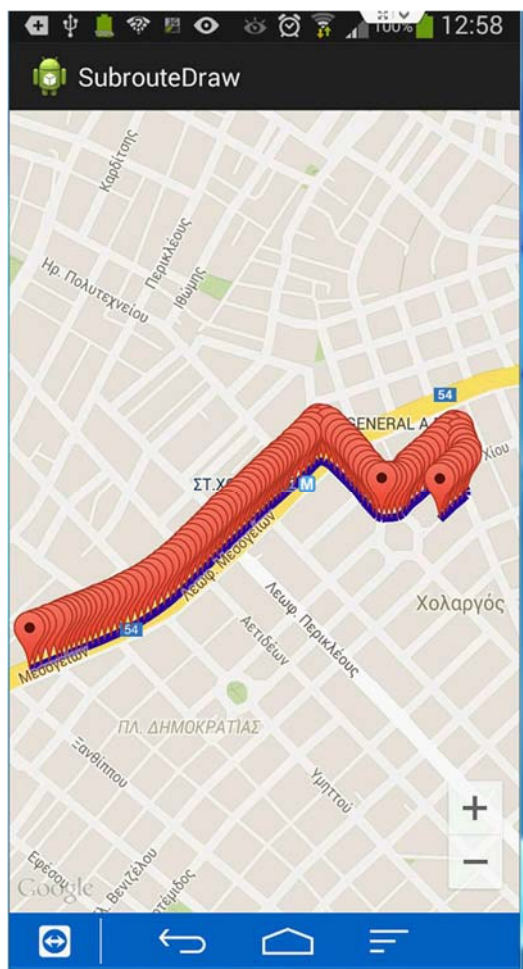


Εικόνα 39: Αγγίζοντας μια πινέζα ο χρήσης βλέπει τις τιμές ταχύτητας και κατανάλωσης του σημείου.

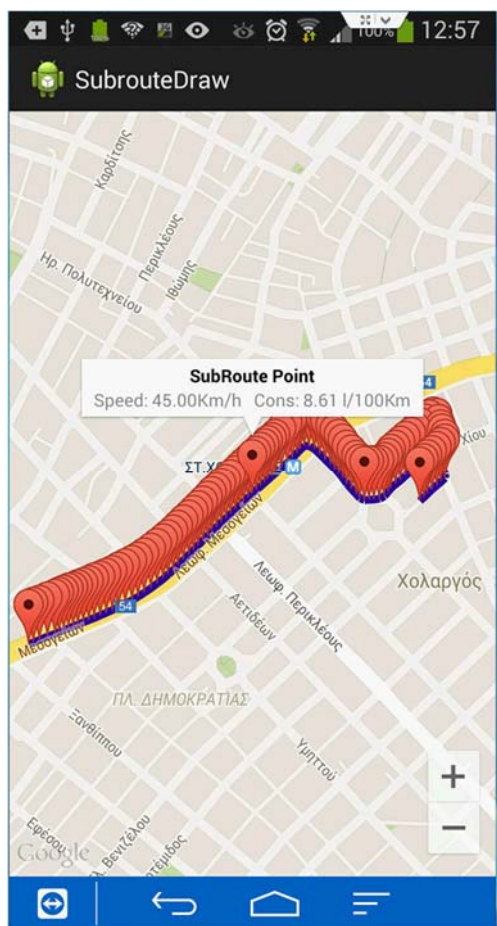


Εικόνα 40: Οι μονάδες μέτρησης ταχύτητας και κατανάλωσης ορίζονται από τα 'Preferences' της εφαρμογής.

Χρησιμοποιώντας τα κουμπιά του χάρτη συν και πλην ο χρήστης αλλάζει το επίπεδο εστίασης του χάρτη ώστε να εστιάσει στα σημεία που τον ενδιαφέρουν.



Εικόνα 41: Το σύνολο της υποδιαδρομής.



Εικόνα 42: Τιμές ταχύτητας και κατανάλωσης συγκεκριμένου σημείου.

2. Αρχιτεκτονική Συστήματος

2.1 Επιλογή Των Εργαλείων.

Το πρώτο βήμα στις ανάπτυξη της εφαρμογής ήταν η επιλογή των εργαλείων. Για το IDE επιλέξαμε το Eclipse όπως αναφέραμε και στην εισαγωγή λόγω της εξοικείωσης που έχουμε με το περιβάλλον. Σε περίπτωση που κάποιος αποφάσιζε να αρχίσει τώρα την ενασχόληση του με τη συγγραφή εφαρμογών για το Android θα του συνιστούσαμε να επιλέξει το Android Studio λόγω του ότι αυτό προτείνει σήμερα η Google.

Προκειμένου να ελέγχουμε τον κώδικα μας αρχικά χρησιμοποιήσαμε τον εξομοιωτή Genymotion και ακολούθως τη συσκευή Samsung Galaxy SIII. Αυτό έγινε διότι η ανάπτυξη της εφαρμογής ακολούθησε δύο στάδια. Το πρώτο περιλάμβανε την καταγραφή της διαδρομής και το δεύτερο την επικοινωνία με το OBD II. Για τη διαδικασία καταγραφής της θέσης οι δοκιμές του κώδικα έγιναν στον εξομοιωτή λόγω του ότι μας παρέχει ένα λειτουργικό γραφικό περιβάλλον για να εισάγουμε συντεταγμένες. Επίσης η βάση δεδομένων της εφαρμογής είναι πρόσβασιμη με τη χρήση του εξομοιωτή με ένα απλό copy – paste. Αντίθετα στη συσκευή η βάση δεδομένων είναι προσβάσιμη μόνο από την εφαρμογή που τη δημιουργεί, πράγμα που δυσχεραίνει το προγραμματιστή.

Το δεύτερο στάδιο της εφαρμογής αναπτύχθηκε πάνω στη συσκευή λόγω του ότι ο εξομοιωτής δεν μπορεί να λειτουργήσει το Bluetooth. Για το λόγο αυτό ότι είχε να κάνει με το OBDII έπρεπε να γίνει πάνω στη συσκευή αφού είχε ολοκληρωθεί πλήρως η λειτουργικότητα της καταγραφής θέσης. Έτσι κατά τη πρώτη φάση η εφαρμογή κατέγραφε τη θέση αποθηκεύοντας για ταχύτητα και κατανάλωση μηδενικές τιμές. Στη συνέχεια και κατά τη δεύτερη δημιουργήθηκαν οι απαραίτητες συναρτήσεις που καλούνταν όταν η εφαρμογή ήθελε να αποθηκεύσει τις άνω τιμές και επέστρεφαν τα δεδομένα του απαντούσε το ολοκληρωμένο κύκλωμα.

Τέλος για το OBDII αρχικά χρησιμοποιήθηκε ο εξομοιωτής OBDSim που αναφέρθηκε σε προηγούμενο κεφάλαιο. Ο εξομοιωτής έτρεχε σε ένα λαπτοπ τύπου HP MINI το οποίο διαθέτει θύρα Bluetooth. Ακολούθως παραλάβαμε από το e-bay τον ELM327 έκδοσης 1.5 προσαρμογέα με κόστος 7€ τον οποίο προσαρμόσαμε σε ένα TOYOTA Yaris 1300cc του 2005.



Εικόνα 43: Ο ELM327 προσαρμογέας που προμηθευτήκαμε.

2.2 Περιγραφή Ανάπτυξης – Χρήσης Τεχνολογιών.

2.2.1 Καταγραφή Της Διαδρομής

Το πρώτο στάδιο ανάπτυξης περιλάμβανε την καταγραφή της διαδρομής του χρήστη. Κρίναμε απαραίτητο η εφαρμογή να εμφανίζει το χάρτη της Google Maps ώστε ο χρήστης να μπορεί να παρακολουθεί τη κίνηση του πάνω στο χάρτη. Για να επιτευχθεί αυτό απαιτείται μεταξύ άλλων το API της Google maps. Από το επίσημο δικτυακό τόπο της εταιρίας πληροφορηθήκαμε τον τρόπο με τον οποίο λειτουργεί το συγκεκριμένο API, τα αντικείμενα που θα πρέπει να δημιουργήσουμε και τις καταχωρήσεις στο αρχείο manifest.

Για να εισαχθεί στην εφαρμογή το συγκεκριμένο API θα πρέπει πρώτιστος να δημιουργηθεί ένα κλειδί και να εισαχθεί στο αρχείο manifest. Οδηγίες για τη διαδικασία παρέχονται από την ιστοσελίδα της εταιρίας. Ακολούθως το αρχείο manifest θα πρέπει να έχει δικαιώματα χρήσης του δικτύου και της κατάστασής του προκειμένου να μεταφέρει δεδομένα, του αισθητήρα GPS για να αντλεί δεδομένα συντεταγμένων και επιπλέον πρόσβαση στο χώρο αποθήκευσης της συσκευής για να αποθηκεύει τα δεδομένα των διαδρομών. Τέλος για την επικοινωνία με το ολοκληρωμένο κύκλωμα μέσω Bluetooth απαιτούνται δικαιώματα χρήσης της εν λόγω συσκευής.

Έχοντας ολοκληρώσει το ευαίσθητο θέμα των δικαιωμάτων της εφαρμογής εισερχόμαστε στην ανάλυση των αντικειμένων της εφαρμογής αρχίζοντας από το βασικότερο που είναι η κλάση MainActivity. Η κλάση αυτή είναι η πρώτη που καλείται όταν εκτελείται η εφαρμογή και συντονίζει το σύνολο των αντικειμένων.

- Η κλάση Main Activity.

Πρώτη ενέργεια της κλάσης είναι να ορίσει τον αισθητήρα GPS και τις διάφορες παραμέτρους του. Στη συνέχεια ελέγχει εάν η υπηρεσία Google Play Services υποστηρίζεται από την συσκευή μιας και η υπηρεσία είναι απαραίτητη προκειμένου να λειτουργήσει το API και ζητά από τον αισθητήρα τις τρέχουσες συντεταγμένες. Ακολούθως αρχικοποιεί το χάρτη και τον εμφανίζει στην οθόνη. Ορίζει ότι η οθόνη θα πρέπει να παραμένει ενεργή όσο ο χρήστης βρίσκεται σε αυτό το Activity ώστε η συσκευή να μην κλείνει την εφαρμογή παρά μόνο όταν ο χρήστης το επιλέξει. Τέλος δημιουργεί τα αντικείμενα DAO που επικοινωνούν με τη βάση δεδομένων.

Έχοντας ολοκληρώσει την αρχικοποίηση αντικειμένων εκτελείται η συνάρτηση `onOptionsItemSelected` και η εφαρμογή περιμένει από το χρήστη να πατήσει το Start Recording από το μενού προκειμένου να αρχίσει η καταγραφή της διαδρομής. Μόλις πατηθεί το συγκεκριμένο κουμπί η πρώτη ενέργεια της συνάρτησης είναι να διαβάσει την κλάση Prefs. Η κλάση Prefs περιέχει τις ρυθμίσεις της εφαρμογής που έχει εισάγει ο χρήστης πατώντας από το μενού το κουμπί Preferences. Στη περίπτωση που ο χρήστης δεν έχει εισάγει δεδομένα για κάποια τιμή η εφαρμογή εισάγει τα προαποφασισμένα δεδομένα.

Ακολούθως η συνάρτηση αρχικοποιεί κάποια flags για να ξεχωρίζει τις διαδρομές. Συγκεκριμένα αν ο χρήστης ολοκληρώσει μια διαδρομή πατώντας Stop Recording από το μενού και ακολούθως πατήσει εκ νέου Start Recording η εφαρμογή θα πρέπει να καταλάβει ότι πρόκειται για νέα διαδρομή και δεν θα ενώσει πάνω στο χάρτη το προηγούμενο σημείο με το νέο.

Στη συνέχεια η εφαρμογή ελέγχει εάν έχει πάρει συντεταγμένες από το προηγούμενο βήμα. Εάν δεν έχει ζητά εκ νέου και ενημερώνει το χρήστη ότι δεν έχει συνταγμένες ώστε αυτός να προβεί στις απαραίτητες ενέργειες. Η συνέχιση της εκτέλεσης της εφαρμογής δεν έχει νόημα χωρίς συντεταγμένες και για το λόγο αυτό η εκτέλεση σταματά μέχρι ο χρήστης να ενεργοποιήσει το GPS και να επιλέξει εκ νέου Start Recording.

Στη περίπτωση που υπάρχουν συντεταγμένες η εφαρμογή ζουμάρει στο χάρτη στο σημείο που βρίσκεται ο χρήστης και τοποθετεί μια πινέζα με τίτλο 'Subroute Start'. Ακολούθως δημιουργεί ένα αντικείμενο route δίνοντας του το τρέχων timestamp. Στη συνέχεια και έχοντας το id του δημιουργεί ένα subroute δίνοντας του το id του route στο οποίο ανήκει ώστε να γνωρίζουμε ότι το συγκεκριμένο subroute σε πιο route ανήκει. Ακολουθεί η δημιουργία του αντικειμένου point το οποίο αποθηκεύει τις τιμές του συγκεκριμένου σημείου. Οι τιμές αυτές είναι το timestamp που δημιουργήθηκε, οι συντεταγμένες του, η απόσταση του σε μέτρα από το προηγούμενο, που στη περίπτωση έναρξης διαδρομής είναι 0, η ταχύτητα με την οποία κινείται ο χρήστης και η τρέχουσα κατανάλωση του οχήματος και τέλος το id του subroute στο οποίο ανήκει το συγκεκριμένο σημείο. Οι δύο συναρτήσεις που επιστρέφουν ταχύτητα και κατανάλωση δημιουργήθηκαν στη δεύτερη φάση της ανάπτυξης της εφαρμογής, που περιλάμβανε την επικοινωνία με το ολοκληρωμένο κύκλωμα. Στη διάρκεια της πρώτης φάσης είχαν αντικατασταθεί από τις τιμές 0 ώστε η εφαρμογή να είναι λειτουργική αποθηκεύοντας απλά 0. Τέλος αποθηκεύεται η στάθμη της μπαταρίας και δημιουργείται ένα χρώμα με τυχαίο τρόπο το οποίο θα χρησιμοποιηθεί στην ένωση των σημείων πάνω στο χάρτη.

Στο σημείο αυτό η εφαρμογή έχει αποθηκεύσει στη βάση τα τρία αντικείμενα. Το route που περιέχει ένα μοναδικό κωδικό id και ένα timestamp της χρονικής στιγμής που άρχισε η καταγραφή της διαδρομής. Το subroute που περιέχει μόνο το id του route στο οποίο ανήκει και το point. Το point περιέχει όπως αναφέραμε timestamp, συντεταγμένες, απόσταση σε

μέτρα από το προηγούμενο point, ταχύτητα και κατανάλωση αλλά και το id του subroute στο οποίο ανήκει. Ακολούθως η εφαρμογή ζητά εκ νέου συντεταγμένες με τη χρήση της συνάρτησης onLocationChanged.

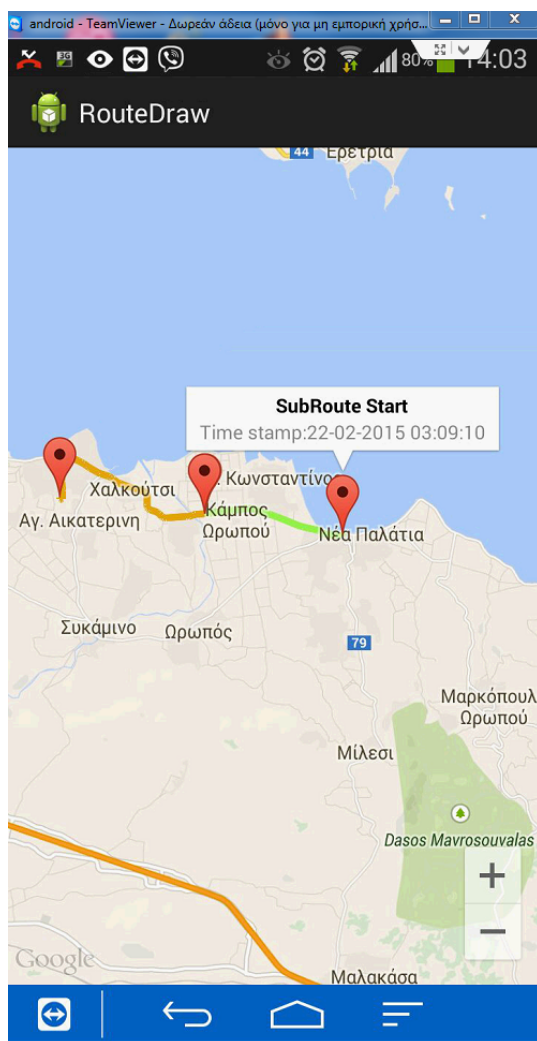
Η συνάρτηση αυτή εκτελεί διαφορετικές λειτουργίες ανάλογα από πιο σημείο της κλάσης καλείται. Για παράδειγμα όταν καλείται κατά την σύνδεση με την υπηρεσία Google Play Services επιστρέφει απλώς τις τρέχουσες συντεταγμένες. Όταν καλείται από την διαδικασία του κουμπιού StartRecording συνεχίζει την καταγραφή της διαδρομής. Συγκεκριμένα πρώτη της ενέργεια είναι να ελέγξει εάν ο χρήστης κινείται. Για να το πετύχει αυτό υπολογίζει την απόσταση σε μέτρα της τρέχουσας τοποθεσίας από την προηγούμενη. Εάν είναι μεγαλύτερη από το κατώφλι που έχει ορίσει ο χρήστης στα Preferences η εφαρμογή θεωρεί ότι ο χρήστης κινείται.

Στη περίπτωση αυτή η εφαρμογή ζουμάρει στη νέα τοποθεσία και δημιουργεί νέο αντικείμενο point το οποίο περιέχει το ίδιο subroute id μιας και ανήκει στην συγκεκριμένη υποδιαδρομή αλλά και στη διαδρομή που δημιουργήθηκαν στο προηγούμενο βήμα. Επίσης τα δύο σημεία ενώνονται στο χάρτη χρησιμοποιώντας σαν χρώμα για τη γραμμή που τα ενώνει αυτό που δημιουργήθηκε στο προηγούμενο βήμα. Η εκτέλεση του προγράμματος συνεχίζεται με τον ίδιο τρόπο όσο ο χρήστης μετακινείται. Η εφαρμογή δημιουργεί νέο αντικείμενο point, το αποθηκεύει στη βάση και ενώνει το νέο point με το προηγούμενο στο χάρτη χρησιμοποιώντας το ίδιο χρώμα.

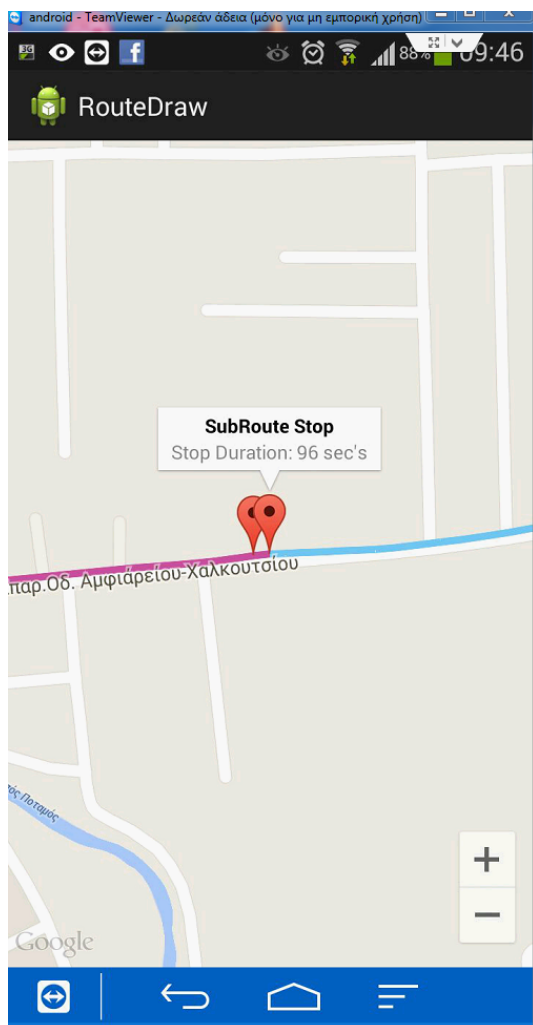
Όταν ο χρήστης πάψει να κινείται η εφαρμογή αποθηκεύει ένα timestamp με τη συνάρτηση elapsedRealTime του αντικειμένου SystemClock. Η συνάρτηση αυτή επιστρέφει το σύνολο των milliseconds που έχουν παρέλθει από την έναρξη της συσκευής. Ακολούθως η εφαρμογή περιμένει να λάβει εκ νέου συντεταγμένες ελέγχοντας το χρονικό διάστημα που έχει περάσει από τη στιγμή που ο χρήστης έπαψε να κινείται. Στη περίπτωση που ο χρήστης κινηθεί και δεν έχει παρέλθει το χρονικό διάστημα που ορίζει τη στάση η εκτέλεση συνεχίζεται όπως έχει περιγραφεί στο προηγούμενο βήμα, δημιουργώντας δηλαδή νέα αντικείμενα point.

Η εκτέλεση της εφαρμογής διαφοροποιείται όταν παρέλθει το χρονικό διάστημα που ορίζει τη στάση. Η εφαρμογή καταλαβαίνει ότι ο χρήστης βρίσκεται σε στάση οπότε έχει ολοκληρώσει μια υποδιαδρομή. Για το λόγο αυτό λαμβάνει εκ νέου τη στάθμη της μπαταρίας και ενημερώνει το αντικείμενο subroute που δημιουργήθηκε στο προηγούμενο βήμα. Γνωρίζοντας το μοναδικό κωδικό του αντικειμένου subroute η εφαρμογή καλεί τη συνάρτηση informSubroute του αντικειμένου υποδιαδρομής subDao δίνοντας της το id του συγκεκριμένου subroute που θα πρέπει να ενημερώσει, την συνολική απόσταση της υποδιαδρομής, το μέσο όρο ταχύτητας, τη μέγιστη αλλά και ελάχιστη ταχύτητα, το μέσο όρο κατανάλωσης καυσίμου του οχήματος, τη μέγιστη αλλά και ελάχιστη κατανάλωση καυσίμου, τα επίπεδα της μπαταρίας κατά την έναρξη και κατά τη λήξη της υποδιαδρομής και τέλος τη χρονική διάρκεια της στάσης. Αμέσως μετά αρχίζει μια νέα υποδιαδρομή, δημιουργώντας ένα νέο αντικείμενο υποδιαδρομής subroute δίνοντας του το κωδικό id του προϋπάρχον route μιας και αυτή η υποδιαδρομή ανήκει στην ίδια διαδρομή. Ακολούθως αρχικοποιεί την τιμή απόσταση, το χρώμα της γραμμής που θα ενώνει τα σημεία της νέας υποδιαδρομής αφού αυτή θα σχεδιάζεται με διαφορετικό χρώμα και αποθηκεύει εκ νέου τη στάθμη της μπαταρίας. Στη συνέχεια δημιουργεί νέο αντικείμενο point δίνοντας του τώρα το id του νέου subroute μαζί με τις υπόλοιπες τιμές που απαιτεί το point. Τέλος τοποθετεί στο χάρτη μια νέα πινέζα με τίτλο Stop Duration : και συμπληρώνει με το χρονικό διάστημα που διήρκεσε η στάση και ενώνει τα δύο points με μια νέα γραμμή με το νέο χρώμα.

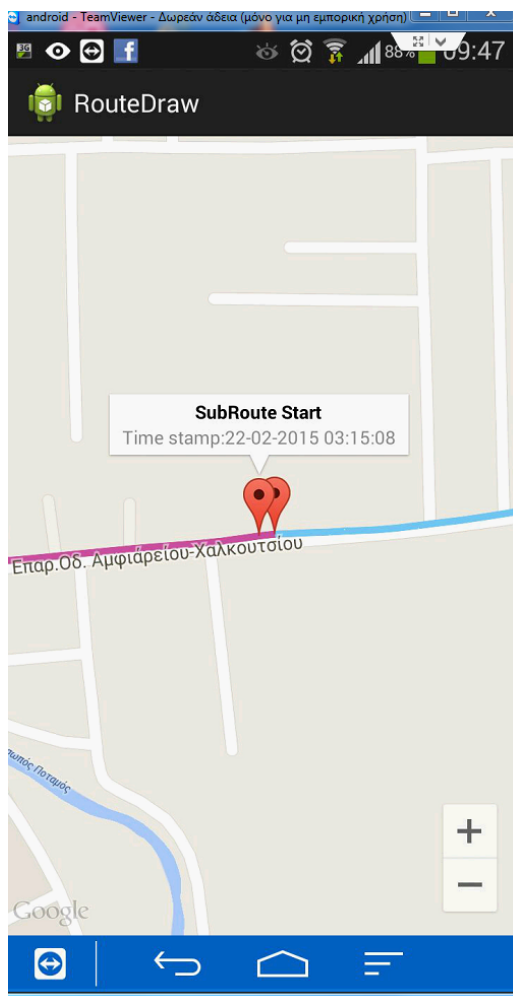
Η εκτέλεση του προγράμματος συνεχίζεται με τον ίδιο τρόπο έως ο χρήστης να πατήσει από το μενού Stop Recording. Δηλαδή η εφαρμογή θα δημιουργεί αντικείμενα point με ρυθμό τη τιμή που έχει ορίσει ο χρήστης στο ρυθμό ανάγνωσης συντεταγμένων από το GPS και θα τα αποθηκεύει στη βάση δεδομένων. Όταν ο χρήστης πάψει να κινείται η εφαρμογή θα το καταλάβει και με το που αρχίσει να κινείται πάλι θα ενημερώσει το αντικείμενο υποδιαδρομή με τις τιμές των αντικειμένων point της συγκεκριμένης διαδρομής και με τη χρονική διάρκεια της στάσης. Θα εμφανίσει στο χάρτη μια πινέζα στο τέλος της υποδιαδρομής και μια στην αρχή της νέας. Έτσι ο χρήστης θα βλέπει στην οθόνη της συσκευής τη διαδρομή που έχει κάνει χωρισμένη σε υποδιαδρομές και πατώντας πάνω στις πινέζες θα βλέπει από πού ξεκίνησε, πού σταμάτησε και πόση ώρα ήταν σε στάση.



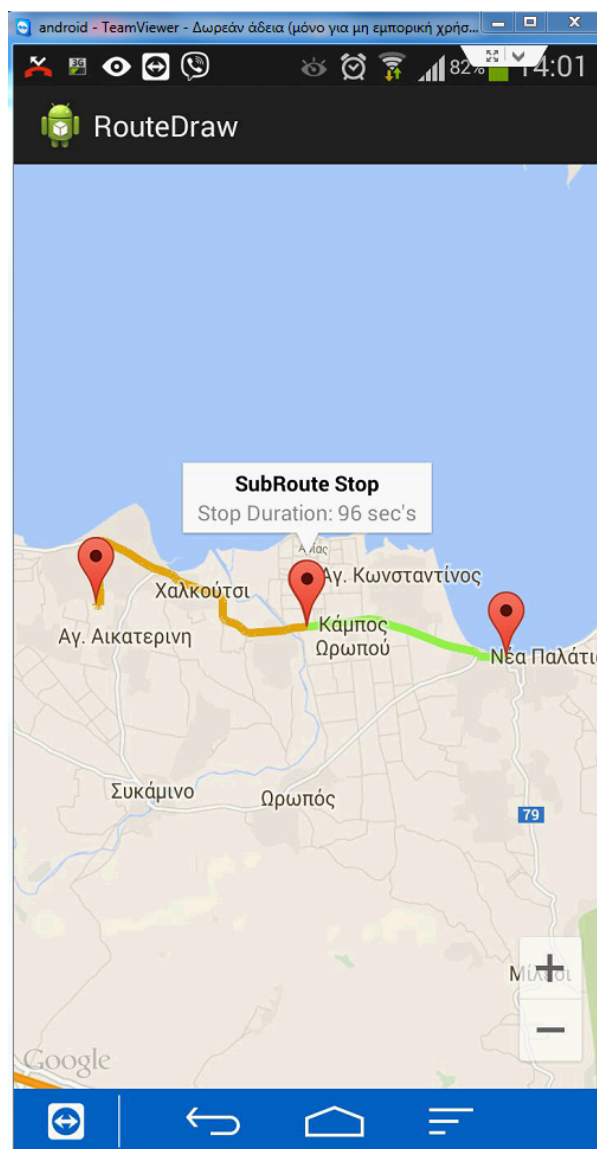
Εικόνα 44: Η διαδρομή που έχει πραγματοποιήσει ο χρήστης με τις υποδιαδρομές της. Οι υποδιαδρομές εμφανίζονται με διαφορετικό χρώμα ενώ στην πινέξες δείχνουν την έναρξη και την ολοκλήρωση της κάθε υποδιαδρομής.



Εικόνα 45: Πινέζα στο τέλος της υποδιαδρομής με αναφορά στη χρονική διάρκεια στάσης



Εικόνα 46: Πινέζα στο σημείο έναρξης της νέα υποδιαδρομής.



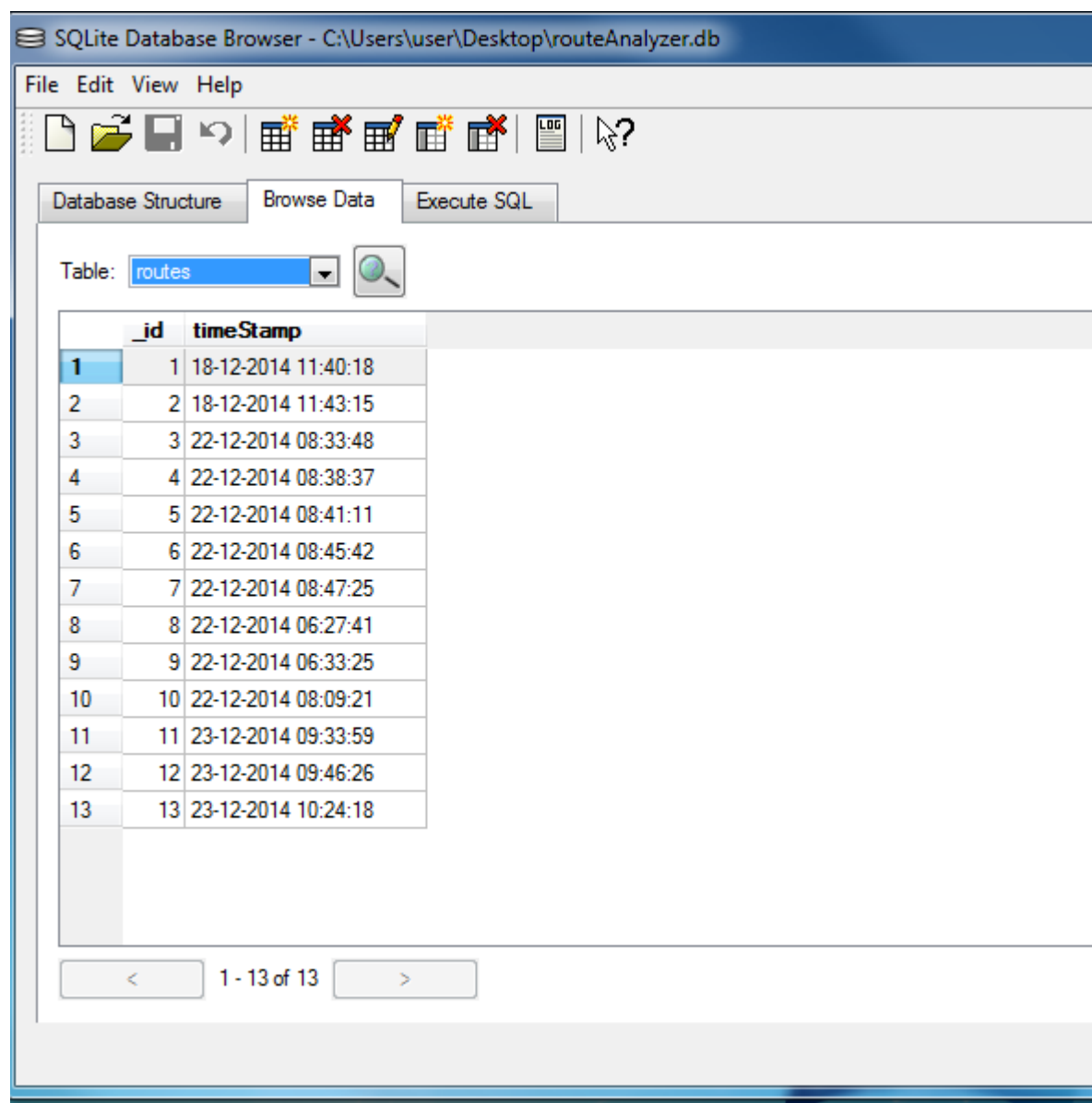
Εικόνα 47: Η εμφάνιση της στάσης μεταξύ δυο υποδιαδρομών μαζί με τη διάρκεια της στάσης. Λόγο του επιπέδου του ζουμ δεν είναι ορατές οι δύο πινέζες στη στάση στο μέσο της διαδρομής.

Όταν ο χρήστης ολοκληρώσει τη διαδρομή του θα επιλέξει από το μενού την επιλογή Stop Recording. Τότε η εφαρμογή θα ελέγξει εκ νέου το επίπεδο της μπαταρίας και θα ενημερώσει το τρέχων αντικείμενο υποδιαδρομής με τις απαιτούμενες τιμές. Θα τοποθετηθεί μια νέα πινέζα στο χάρτη και θα περιμένει το χρήστη να επιλέξει κάποιο στοιχείο από το μενού. Από το σημείο αυτό ο χρήστης μπορεί να ξεκινήσει μια νέα διαδρομή επιλέγοντας Start Recording από το μενού οπότε και η εφαρμογή θα επαναλάβει τα ίδια βήματα σχεδιάζοντας πάνω στο χάρτη τη νέα διαδρομή διατηρώντας ταυτόχρονα και τη παλιά. Επιλέγοντας οποιοδήποτε άλλο στοιχείο του μενού η εφαρμογή εγκαταλείπει την κλάση MainActivity και τη σκυτάλη της εκτέλεσης αναλαμβάνει το αντικείμενο του στοιχείου που έχει επιλέξει ο χρήστης.

- Η Βάση Δεδομένων Της Εφαρμογής Και Τα Αντικείμενα της.

Η βάση δεδομένων της εφαρμογής αποθηκεύει τρία αντικείμενα : Το route, το subroute και το point. Κάθε ένα από αυτά περιέχει τα δικά του δεδομένα και όλα μαζί αποτελούν μια διαδρομή. Για κάθε ένα από αυτά έχει δημιουργηθεί και το αντίστοιχο DAO του που αναλαμβάνει την αντιστοίχιση του αντικειμένου με τον πίνακα της βάσης δεδομένων προκειμένου να απλοποιηθεί η χρήση της βάσης.

Το αντικείμενο route αποτελείται από ένα μοναδικό κωδικό, το id και ένα timestamp που αντιστοιχεί στη χρονική στιγμή έναρξης της διαδρομής.



	_id	timeStamp
1	1	18-12-2014 11:40:18
2	2	18-12-2014 11:43:15
3	3	22-12-2014 08:33:48
4	4	22-12-2014 08:38:37
5	5	22-12-2014 08:41:11
6	6	22-12-2014 08:45:42
7	7	22-12-2014 08:47:25
8	8	22-12-2014 06:27:41
9	9	22-12-2014 06:33:25
10	10	22-12-2014 08:09:21
11	11	23-12-2014 09:33:59
12	12	23-12-2014 09:46:26
13	13	23-12-2014 10:24:18

Εικόνα 48: Στιγμιότυπο του πίνακα Routes της εφαρμογής.

	subr	subtotal_dist	subaver_speed	submax_speed	submin_speed	suba	subn	sub	subb	subl	stop_d	subr
1	1	521.166334152222	25.8665775299072	57.8744163513184	0.0	0.0	0.0	0.0	47.0	47.0	55992.0	1
2	2	231.143280029297	4.99740489323934	6.26879119873047	3.32532072067261	0.0	0.0	0.0	47.0	47.0	51018.0	1
3	3	176.834991455078	2.50767795244853	3.44892716407776	1.81206226348877	0.0	0.0	0.0	47.0	47.0	0.0	1
4	4	160.458000183105	9.51005434989929	17.5340461730957	0.0	0.0	0.0	0.0	47.0	47.0	54971.0	2
5	5	178.379193305969	2.70789906382561	4.91811990737915	0.49472177028656	0.0	0.0	0.0	47.0	47.0	0.0	2
6	6	555.729955673218	7.67737948894501	17.5430393218994	0.0	0.0	0.0	0.0	100.0	00.0	54010.0	3
7	7											4
8	8	468.172180175781	23.8694543838501	34.6452522277832	0.0	0.0	0.0	0.0	100.0	00.0	53986.0	5
9	9											5
10	10	216.450300216675	10.8054565429688	24.2174301147461	0.0	0.0	0.0	0.0	100.0	00.0	54026.0	6
11	11	184.336090087891	3.96863905588786	4.68261671066284	3.49251842498779	0.0	0.0	0.0	100.0	00.0	0.0	6
12	12	270.324807167053	24.2224363327026	70.1562805175781	0.0	0.0	0.0	0.0	100.0	00.0	53011.0	7
13	13	206.158218383789	2.43504765033722	4.20567941665649	1.23959505558014	0.0	0.0	0.0	100.0	00.0	0.0	7
14	14	308.443759918213	5.33031539916992	12.1642818450928	0.0	0.0	0.0	0.0	100.0	00.0	59040.0	8
15	15	250.456540107727	1.61449996232986	2.46782183647156	1.297897756099701	0.0	0.0	0.0	100.0	00.0	56023.0	8
16	16	172.357608795166	1.791786074638367	1.12855660915375	0.54539829492569	0.0	0.0	0.0	100.0	00.0	0.0	8

Εικόνα 49: Στιγμιότυπο του πίνακα Subroutes της εφαρμογής.

Το αντικείμενο Subroute είναι πιο σύνθετο από το Route και περιέχει τα δεδομένα που χαρακτηρίζουν τη συγκεκριμένη υποδιαδρομή. Από αριστερά προς τα δεξιά ο πίνακας περιέχει τα πεδία κωδικός υποδιαδρομής, συνολική απόσταση σε μέτρα, μέσος όρος ταχύτητας, μέγιστη ταχύτητα, ελάχιστη ταχύτητα, μέση κατανάλωση καυσίμου, μέγιστη κατανάλωση και ελάχιστη, επίπεδο της μπαταρίας κατά την έναρξη της διαδρομής και ακολούθως κατά τη λήξη της, χρονική διάρκεια στάσης και τέλος κωδικό αντικειμένου Route στο οποίο ανήκει η διαδρομή. Για παράδειγμα οι πρώτες τρεις εγγραφές αποτελούν την πρώτη υποδιαδρομή της διαδρομής 1 και για το λόγο αυτό έχουν και οι τρεις κωδικό διαδρομής 1 (το τελευταίο πεδίο του πίνακα). Οι τιμές της κατανάλωσης είναι μηδενικές μιας και ο πίνακας αυτός δημιουργήθηκε κατά την πρώτη φάση ανάπτυξης της εφαρμογής όπου η λειτουργικότητα επικοινωνίας με το ενσωματωμένο σύστημα διάγνωσης δεν είχε υλοποιηθεί. Τιμές ταχύτητας υπάρχουν λόγω του ότι η ταχύτητα στη φάση αυτή υπολογιζόταν από το χρονικό διάστημα και την απόσταση σε μέτρα μεταξύ δύο σημείων. Στην επόμενη φάση οι τιμές ταχύτητας αλλά και κατανάλωσης προέρχονται από το ενσωματωμένο σύστημα διάγνωσης του οχήματος.

	point	point_time	point_lat	point_lon	point_dist	point_speed	point_id	point_order
1	1	18-12-2014 11:40:18	65.9346166	-18.5645983	0.0	0.0	0.0	1.0
2	2	18-12-2014 11:40:29	65.93621	-18.5645983	177.674453735352	57.8744163513184	0.0	1.0
3	3	18-12-2014 11:40:30	65.93621	-18.56404	25.4127635955811	7.5601978302002	0.0	1.0
4	4	18-12-2014 11:40:34	65.9374783	-18.5652633	151.990234375	34.0827751159668	0.0	1.0
5	5	18-12-2014 11:40:38	65.9375316	-18.56891	166.088882446289	29.8154983520508	0.0	1.0
6	6	18-12-2014 11:41:36	65.9364816	-18.56891	0.0	5.39810276031494	0.0	2.0
7	7	18-12-2014 11:41:37	65.9364816	-18.5719366	137.763656616211	6.26879119873047	0.0	2.0
8	8	18-12-2014 11:41:59	65.9371116	-18.570585	93.379623413086	3.32532072067261	0.0	2.0
9	9	18-12-2014 11:42:52	65.9365333	-18.5735033	0.0	3.44892716407776	0.0	3.0
10	10	18-12-2014 11:42:53	65.9359033	-18.5742533	78.1049194335938	1.81206226348877	0.0	3.0
11	11	18-12-2014 11:42:55	65.93502	-18.5741033	98.7300720214844	2.26204442977905	0.0	3.0
12	12	18-12-2014 11:43:15	65.93502	-18.5741033	0.0	0.0	0.0	4.0
13	13	18-12-2014 11:43:27	65.9347216	-18.57346	44.3242797851563	12.4808301925659	0.0	4.0
14	14	18-12-2014 11:43:29	65.9341966	-18.5743816	72.0210952758789	17.5340461730957	0.0	4.0
15	15	18-12-2014 11:43:34	65.9345466	-18.5748333	44.1126251220703	8.02534103393555	0.0	4.0
16	16	18-12-2014 11:44:32	65.9340216	-18.5748333	0.0	2.74360466003418	0.0	5.0

Εικόνα 50: Στιγμιότυπο του πίνακα Points της εφαρμογής.

Το τελευταίο αντικείμενο της βάσης δεδομένων είναι το Point το οποίο ουσιαστικά αποτελεί τα σημεία της διαδρομής. Από αριστερά προς δεξιά ο πίνακας περιέχει τα πεδία κωδικός σημείου, το timestamp που πάρθηκε το σημείο, γεωγραφικό πλάτος και μήκος σημείου, η απόσταση του σημείου από το προηγούμενο, ταχύτητα του οχήματος και κατανάλωση καυσίμου και τέλος κωδικός σημείου υποδιαδρομής στην οποία ανήκει το συγκεκριμένο σημείο. Ομοίως οι πρώτες τέσσερις εγγραφές αναφέρονται στα σημεία της πρώτης υποδιαδρομής, εξού και ο κωδικός υποδιαδρομής είναι 1, η οποία υποδιαδρομή ανήκει στη πρώτη διαδρομή. Επίσης το πρώτο σημείο κάθε υποδιαδρομής έχει απόσταση 0 από το προηγούμενο αφού είναι το πρώτο σημείο μιας νέας διαδρομής. Η ταχύτητα είναι 0 όταν ο χρήστης δεν ήταν σε κίνηση κατά την έναρξη της διαδρομής ενώ στις αντίθετες περιπτώσεις περιέχει τιμές.

Έχοντας μια πλήρη εικόνα των αντικειμένων της βάσης μπορούμε να δούμε πώς λειτουργεί η εφαρμογή. Μόλις ο χρήστης επιλέξει από το Menu StartRecording η mainactivity θα δημιουργήσει ένα νέο αντικείμενο Route αποθηκεύοντας σε αυτό το timestamp. Αμέσως μετά θα δημιουργήσει ένα αντικείμενο Subroute δίνοντας του σαν route_id το id του route που δημιούργησε. Με τον τρόπο αυτό γίνεται ο διαχωρισμός των διαφόρων διαδρομών και υποδιαδρομών αυτών. Ακολουθώντας η εφαρμογή δημιουργεί αντικείμενα points δίνοντας σαν timestamp τη χρονική στιγμή λήψης τους, τις τρέχουσες συντεταγμένες από το GPS, την απόσταση του κάθε σημείου από το προηγούμενο, τη ταχύτητα και κατανάλωση του οχήματος και τέλος το κωδικό σημείου της υποδιαδρομής που ανήκει το κάθε point. Με τον τρόπο αυτό γίνεται και ο διαχωρισμός των αντικειμένων point. Με ένα απλό ερώτημα στη βάση τύπου φέρε μου τα points που έχουν subroute_id ίσο με ένα η βάση μας επιστρέφει τα σημεία της πρώτης υποδιαδρομής.

Όσο ο χρήστης βρίσκεται σε κίνηση η εφαρμογή δημιουργεί points με τον ίδιο κωδικό υποδιαδρομής. Η συνάρτηση createPoint του αντικειμένου pointDao κάνει αυτό ακριβώς.

```
Point point = pointDao.createPoint(DateFormat.format("dd-MM-yyyy hh:mm:ss",
new java.util.Date()).toString(), mCurrentLocation.getLatitude(),
```

```
mCurrentLocation.getLongitude(),          dist, readSpeed(),          readCons(),
subr.getId());
```

Η συνάρτηση αρχικά λαμβάνει το timestamp και ακολούθως τις τρέχουσες συντεταγμένες από το αντικείμενο mCurrentLocation. Ακολούθως την απόσταση του αντικειμένου mCurrentLocation από το προηγούμενο, τη ταχύτητα και τη κατανάλωση του οχήματος και τέλος το κωδικό της υποδιαδρομής. Οι συναρτήσεις readSpeed και readCons που επικοινωνούν με το ολοκληρωμένο μπλοκάρουν την εκτέλεση της εφαρμογής έως ότου να έρθει η απάντηση από το ολοκληρωμένο κύκλωμα. Το παρακάτω τμήμα κώδικα αποτελεί την υλοποίηση της δημιουργίας του αντικειμένου point.

```
public Point createPoint(String time, Double Lat, Double Lon, Double Dist,
Double Speed, Double Cons, long subroute_id ) {
    ContentValues values = new ContentValues();
    values.put(DBHelper.COLUMN_POINT_TIME, time);
    values.put(DBHelper.COLUMN_POINT_LAN, Lat);
    values.put(DBHelper.COLUMN_POINT_LON, Lon);
    values.put(DBHelper.COLUMN_POINT_DIST, Dist);
    values.put(DBHelper.COLUMN_POINT_SPEED, Speed);
    values.put(DBHelper.COLUMN_POINT_CONS, Cons);
    values.put(DBHelper.COLUMN_POINT_SUBROUTE_ID, subroute_id);

    long insertId = mDatabase
        .insert(DBHelper.TABLE_POINTS, null, values);

    System.out.println("insertId: "+insertId);

    Cursor cursor = mDatabase.query(DBHelper.TABLE_POINTS,
mAllColumns,
        DBHelper.COLUMN_POINT_ID + " = " + insertId,
null, null,
        null, null);
    cursor.moveToFirst();
    Point newPoint = cursorToPoint(cursor);
    cursor.close();
    return newPoint;
}
```

Μόλις πάψει να κινείται ο χρήστης η εφαρμογή σταματά να δημιουργεί points και περιμένει το χρήστη είτε να αρχίσει εκ νέου να κινείται είτε να παρέλθει το χρονικό διάστημα που ορίζει τη στάση. Αν αρχίσει να κινείται προτού παρέλθει το εν λόγω χρονικό διάστημα η εφαρμογή αρχίζει εκ νέου να δημιουργεί αντικείμενα point με τον ίδιο κωδικό υποδιαδρομής αφού πρόκειται για την συνέχιση της ίδιας υποδιαδρομής. Αν όμως παρέλθει το διάστημα η εφαρμογή εισέρχεται σε κατάσταση στάσης. Αποθηκεύει σε μια μεταβλητή ένα timestamp και περιμένει το χρήστη είτε να αρχίσει να κινείται είτε να πατήσει το Stop Recording.

Μόλις ο χρήστης αρχίσει να κινείται ολοκληρώνεται η τρέχουσα υποδιαδρομή και ξεκινά μια νέα. Η εφαρμογή υπολογίζει το χρονικό διάστημα της στάσης και ενημερώνει την ολοκληρωμένη διαδρομή με το χρονικό διάστημα της στάσης. Ταυτόχρονα και κάνοντας χρήση των συσσωρευτικών συναρτήσεων της SQL υπολογίζει τις απαραίτητες τιμές της συγκεκριμένης διαδρομής. Πιο αναλυτικά για να υπολογίσει τη συνολική απόσταση σε μέτρα της υποδιαδρομής εκτελούμε το ακόλουθο ερώτημα στο πίνακα με τα points:

```
SELECT SUM(point_dist) as point_dist FROM points where point_subroute_id =
+ sub.getId()
```


Το ερώτημα αυτό περιέχεται στη συνάρτηση subDist του αντικείμενου subDao και εκτελείται τη στιγμή ολοκλήρωσης της υποδιαδρομής. Η συνάρτηση λαμβάνει ως όρισμα ένα αντικείμενο subroute και ανακτά το id του μέσω της sub.getId(). Ακολούθως εκτελεί το ερώτημα τα αποτελέσματα του οποίου περιέχονται σε μια δομή δεδομένων της SQL, το αντικείμενο cursor. Τέλος η τιμή μετατρέπεται από κείμενο σε δεκαδικό και επιστρέφεται.

```
public double subDist(Subroute sub){
    double dist=0.0;
    Cursor cursor = mDatabase.rawQuery("SELECT SUM(point_dist) as
point_dist FROM points where point_subroute_id =" + sub.getId()+";",
null);
    if (cursor != null) {
        try {
            if (cursor.moveToFirst()) {
                dist = cursor.getDouble(0);
            }
        } finally {
            cursor.close();
        }
    }

    return dist;
}
```

Με αντίστοιχο τρόπο λαμβάνονται από τα points όλες οι τιμές της υποδιαδρομής. Οι κλήσεις όλων των συναρτήσεων γίνονται από την informSubroute η οποία κλείνει την υποδιαδρομή.

```
public void informSubroute(long subroute_id, double tDist, double avSpeed,
double maxSpeed, double minSpeed, double AverCons, double maxCon, double
minCon, double batStart, double batEnd, double stopDur){
    ContentValues values = new ContentValues();
    values.put(DBHelper.COLUMN_SUBROUTE_TOTAL_DIST, tDist);
    values.put(DBHelper.COLUMN_SUBROUTE_AVER_SPEED, avSpeed);
    values.put(DBHelper.COLUMN_SUBROUTE_MAX_SPEED, maxSpeed);
    values.put(DBHelper.COLUMN_SUBROUTE_MIN_SPEED, minSpeed);
    values.put(DBHelper.COLUMN_SUBROUTE_AVER_CONS, AverCons);
    values.put(DBHelper.COLUMN_SUBROUTE_MAX_CONS, maxCon);
    values.put(DBHelper.COLUMN_SUBROUTE_MIN_CONS, minCon);
    values.put(DBHelper.COLUMN_SUBROUTE_BATT_START, batStart);
    values.put(DBHelper.COLUMN_SUBROUTE_BATT_END, batEnd);
    values.put(DBHelper.COLUMN_SUBROUTE_STOP_DUR, stopDur);

    int insertId =
mDatabase.update(DBHelper.TABLE_SUBROUTES, values, "subroute_id" + "
= " + subroute_id, null);
    if(insertId==1){
        System.out.println("Update Success!!!");
    }
}
```

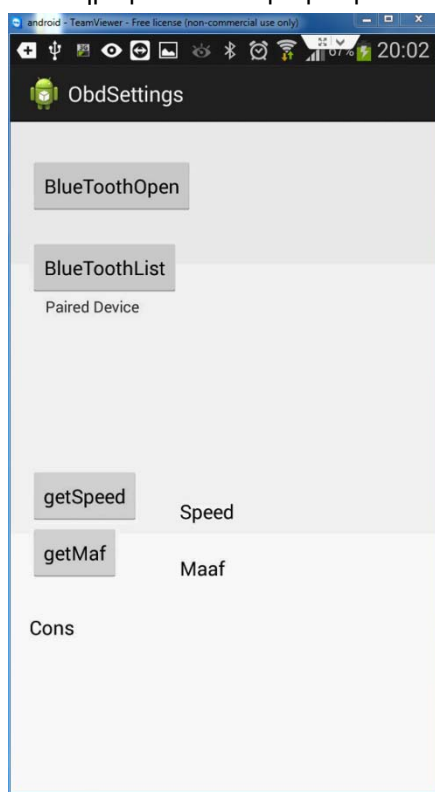
Τώρα η εφαρμογή έχει ολοκληρώσει τη καταγραφή της υποδιαδρομής και ξεκινά μια νέα δημιουργώντας ένα νέο αντικείμενο subDao δίνοντας του το ίδιο route_id. Η λειτουργικότητα αυτή επαναλαμβάνεται έως ότου ο χρήστης επιλέξει από το menu την επιλογή StopRecording.

- Επικοινωνία Με Το ELM327

Όπως προαναφέραμε η εφαρμογή κατά τη δημιουργία των αντικειμένων point επικοινωνεί με το ενσωματωμένο σύστημα διάγνωσης του οχήματος και ανακτά τιμές ταχύτητας και κατανάλωσης καυσίμου. Η επικοινωνία αυτή γίνεται μέσω του Bluetooth πρωτοκόλλου χρησιμοποιώντας sockets.

Από το επίσημο ιστότοπο του Android πληροφορηθήκαμε για τον τρόπο επικοινωνίας του Android με τις συσκευές που υποστηρίζουν το πρωτόκολλο Bluetooth. Αφού ενημερώσαμε το AndroidManifest αρχείο με τα απαραίτητα δικαιώματα δημιουργήσαμε το αντικείμενο ObdSettings το οποίο αναλαμβάνει την επικοινωνία με το ολοκληρωμένο κύκλωμα.

Το αντικείμενο αυτό αποτελεί ένα activity το οποίο ενεργοποιείται όταν ο χρήστης επιλέξει από το menu την επιλογή Settings. Μόλις ο χρήστης 'συνδέσει' τη συσκευή με το ολοκληρωμένο κύκλωμα μπορεί να ελέγξει την επικοινωνία πατώντας το κουμπί getSpeed.



Εικόνα 51: Το Activity obdSettings.

Για να επικοινωνήσει το Activity με το ολοκληρωμένο κύκλωμα δημιουργεί ένα αντικείμενο τύπου ConnectThread που διαχειρίζεται τη σύνδεση και ακολούθως ένα ManageConnectedSocket μέσω του οποίου μεταφέρονται τα μηνύματα. Με τον τρόπο αυτό έχουμε ένα socket στο οποίο το OutputStream γράφουμε το κείμενο που θέλουμε να μεταφέρουμε και ακολούθως στο InputStream διαβάζουμε την απάντηση. Πιο συγκεκριμένα η συνάρτηση write του αντικειμένου ManageConnectedSocket μεταφέρει ένα κείμενο στο OutputStream του το οποίο κείμενο θα διαβάσει το ολοκληρωμένο κύκλωμα:

```
public void write(byte[] bytes) {
    try {
        mmOutputStream.write(bytes);
    } catch (IOException e) { }
}
```

Μέσω της συνάρτησης αυτής το αντικείμενο ObdSettings προκειμένου να διαβάσει για παράδειγμα τη ταχύτητα του οχήματος καλεί την άνω συνάρτηση εισάγοντας ουσιαστικά στο ρεύμα εξόδου τους χαρακτήρες 010D1r. Κατόπιν εκτελεί τη συνάρτηση run του ίδιου αντικειμένου η οποία διαβάζει το ρεύμα εισόδου, το οποίο περιέχει την απάντηση από το

ολοκληρωμένο, έως ότου εντοπίσει το χαρακτήρα '>' ο οποίος σηματοδοτεί το τέλος του μηνύματος. Μόλις ο χαρακτήρας εντοπιστεί η συνάρτηση επιστρέφει:

```
public void run() {
    while (true) {
        try {

            char c = (char) mmInStream.read();
            obdAnswer = obdAnswer + c;
            if(c == '>') {
                System.out.println("Read from the InputStream: " + obdAnswer);
                System.out.println("Message length: " + obdAnswer.length());
                return ;
            }

        } catch (IOException e) {
            break;
        }
    }
}
```

Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει του κουμπί `getSpeed` και όχι από τη `MainActivity` κατά την δημιουργία του αντικειμένου `point`. Αυτό γίνεται για να μπορεί ο χρήστης να βεβαιωθεί ότι η εφαρμογή επικοινωνεί σωστά με το ολοκληρωμένο προτού ξεκινήσει να καταγράφει τη διαδρομή του. Η συνάρτηση αφού λάβει την απάντηση στη μεταβλητή `obdAnswer` παίρνει από αυτή το τμήμα που αναφέρεται στη ταχύτητα μέσω της `substring`, μετατρέπει το δεκαεξαδικό σύστημα σε δεκαδικό και το εμφανίζει στο `TextView`:

```
public void getSpeed (View view){
    String tmp = "";
    if(ManConSo!=null){
        String paramString = "010D1\r";

        byte[] bytes = paramString.getBytes();

        ManConSo.write(bytes);
        ManConSo.run();
        try {
            tmp = ManConSo.obdAnswer.substring(12, 14);
        }
        catch (IndexOutOfBoundsException e){

            Toast.makeText(getApplicationContext(),"substring error: " +
            ManConSo.obdAnswer,Toast.LENGTH_LONG).show();
            speed=0;
        }

        //hex to dec
        try {
            speed = Integer.parseInt(tmp,16);
        }
        catch (NumberFormatException e) {
```

```

    Toast.makeText(getApplicationContext(),"Obd answer at numex : " +
tmp,Toast.LENGTH_LONG).show();
        speed=0;
    }

    //speed to string
    getSpeedT.setText(speed + "");
    ManConSo.obdAnswer = "";
    }
    else

Toast.makeText(getApplicationContext(),"ManConSo is
null",Toast.LENGTH_LONG).show();
    }

```

Αντίθετα η readSpeed καλείται κατά τη δημιουργία του αντικειμένου point. Μόνη της διαφορά είναι ότι αυτή επιστρέφει την ταχύτητα προκειμένου να αποθηκευτεί στη βάση:

```

public Double ReadSpeed (){
    if(ManConSo!=null){
        String paramString = "010D1\r";
        String tmp="";
        byte[] bytes = paramString.getBytes();

        ManConSo.write(bytes);
        ManConSo.run();
        try{
            tmp = ManConSo.obdAnswer.substring(12, 14);
        }
        catch (IndexOutOfBoundsException e){

            Toast.makeText(getApplicationContext(),"substring
error: " + ManConSo.obdAnswer,Toast.LENGTH_LONG).show();

        }
        try {
            speed = Integer.parseInt(tmp,16);
        }
        catch (NumberFormatException e) {

            //Toast.makeText(getApplicationContext(),"Obd answer: " +
ManConSo.obdAnswer,Toast.LENGTH_LONG).show();
            speed=0;
        }
        ManConSo.obdAnswer = "";
        //speed to string
    }
    else{
        Toast.makeText(getApplicationContext(),"ManConSo
is null",Toast.LENGTH_LONG).show();
        speed = 0;
    }
    return Double.valueOf(speed);
}

```

Αντίστοιχη διαδικασία εκτελείται και κατά την ανάγνωση της κατανάλωσης. Έχοντας την ταχύτητα από το προηγούμενο βήμα η εφαρμογή διαβάζει από το ρεύμα εισόδου τη ποσότητα αέρα που εισέρχεται στο θάλαμο καύσης του οχήματος. Ακολούθως εφαρμόζει τη

προαναφερθείσα φόρμουλα ώστε να υπολογίσει τη κατανάλωση σε γαλόνια ανα μίλι. Κατόπιν μετατρέπει τη τιμή αυτή σε χιλιόμετρα ανα λίτρο και είτε την επιστρέφει είτε την εμφανίζει στο TextView μέσω διαφορετικών συναρτήσεων. Στη περίπτωση που η επικοινωνία με το ολοκληρωμένο χαθεί για κάποιο λόγο οι άνω συναρτήσεις θα επιστρέφουν μηδενικές τιμές ενημερώνοντας ταυτόχρονα με toast's το χρήστη ώστε να συνεχίζεται η καταγραφή των διαθέσιμων δεδομένων της διαδρομής. Ο παρακάτω κώδικας αποτελεί τη συνάρτηση readSpeed του αντικειμένου MainActivity.

```
public Double readSpeed(){
    double speed=0;

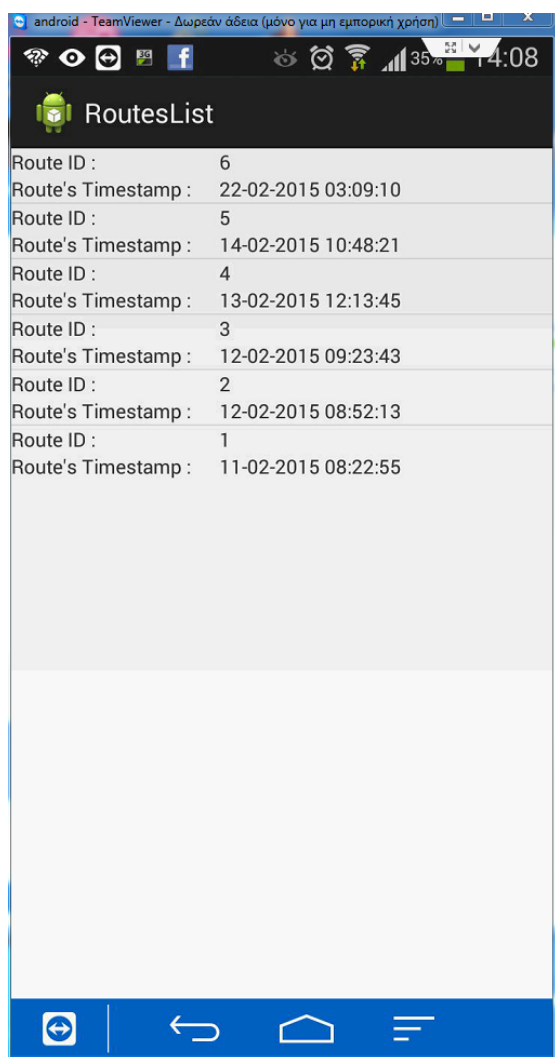
    if(obd!=null) {
        speed = obd.ReadSpeed();
        Toast.makeText(MainActivity.this, "speed from obd: "+
obd.ReadSpeed(), Toast.LENGTH_SHORT).show();
    }

    else {
        Toast.makeText(MainActivity.this, "obd is null",
Toast.LENGTH_SHORT).show();
        speed = 0.0;
    }

    return speed;
}
```

2.2.2 Προβολή Των Καταγεγραμμένων Διαδρομών

Επιλέγοντας από το menu το View Routes η MainActivity ενεργοποιεί το activity RoutesList. Η κλάση αυτή μεταξύ άλλων δημιουργεί το routeDao προκειμένου να ανακτήσει από τη βάση τα αντικείμενα routes και ένα Simple Cursor Adapter ώστε να αντιστοιχίσει τα πεδία του route στο αντικείμενο listView. Ακολούθως διαβάζει από τη βάση όλα τα αντικείμενα routes μέσω της συνάρτησης getAllRoutesToCursor. Αυτή επιστρέφει όλες τις εγγραφές του πίνακα routes στη δομή cursor, η δομή δίδεται ως όρισμα στο Simple Cursor Adapter μαζί με ένα xml το οποίο αποτελεί την κάθε μια εγγραφή του listView και δύο πίνακες που αντιστοιχίζουν πια τιμή θα συνδεθεί με πια. Τέλος κάθε σε κάθε εγγραφή έχει αντιστοιχηθεί onclickListener ώστε πατώντας πάνω σε κάποια εγγραφή να δημιουργείται το activity Route_Summary μέσω ενός intent στο οποίο εισάγουμε και τον κωδικό της διαδρομής στην οποία πάτησε ο χρήστης.



Εικόνα 52: Το Routes List activity. Αποτελείται από ένα ListView και κάθε για κάθε εγγραφή χρησιμοποιεί το activity_route_row.xml. Το αντικείμενο SimpleCursorAdapter αντιστοιχίζει για παράδειγμα το πεδίο Route ID με την τιμή route_id του αντικειμένου cursor.

Το activity Route Summary χρησιμοποιεί το activity_route_summary.xml αρχείο προκειμένου να παρουσιάσει στο χρήστη της λεπτομέρειες της συγκεκριμένης διαδρομής. Ενδεικτικά να αναφέρουμε ότι για να εμφανίσει το μέσο όρο ταχύτητας χρησιμοποιεί τη συνάρτηση routeAverSp του rdao αντικειμένου διαδρομής δίδοντας της ως όρισμα το id της συγκεκριμένης διαδρομής:

```
public String routeAverSp(String routeId){
    double dist=0;
    String txt;
    Cursor cursor = mDatabase.rawQuery("SELECT AVG(subaver_speed)
as averS from subroutes where subroute_route_id =" + routeId +";",null);
    cursor.moveToFirst();
    txt = cursor.getString(0);
    cursor.close();

    if(txt == null)
        txt = "0.0";

    txt = String.format("%.2f", Double.parseDouble(txt));
    txt = txt.replace(",",".");
    if(!this.speedMes.equals("Km/h")){
```

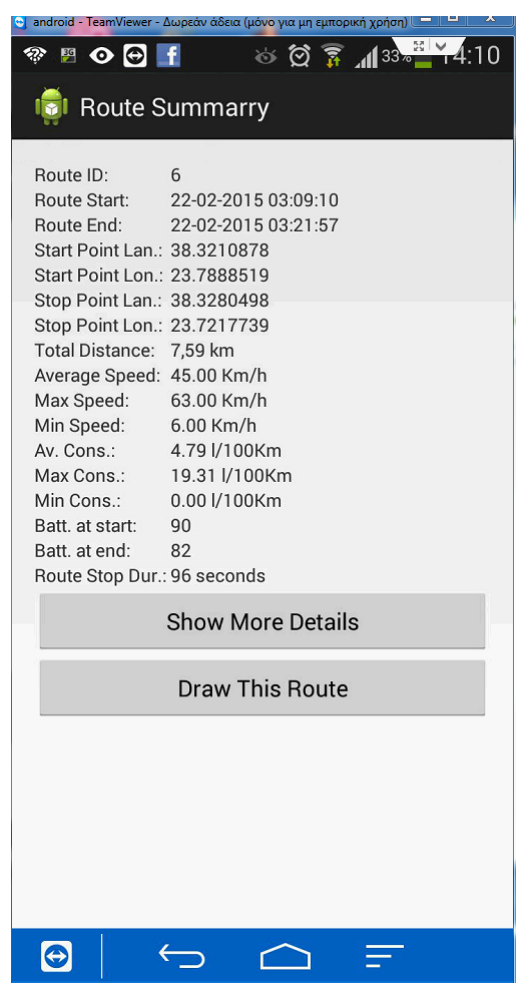
```

        dist = Double.parseDouble(txt) * 0.6214;
        txt = String.format("%.2f",dist);
    }

    return txt + " " + this.speedMes;
}

```

Χρησιμοποιώντας τις συσσωρευτικές συναρτήσεις της SQL παίρνουμε από τον πίνακα των υποδιαδρομών το μέσο όρο ταχύτητας των υποδιαδρομών της συγκεκριμένης διαδρομής. Στη συνέχεια και αφού έχουμε διαβάσει από τη κλάση Preferences την επιθυμητή μονάδα μέτρησης του χρήστη επιστρέφουμε τη τιμή με δύο δεκαδικά στοιχεία προσθέτοντας τη μονάδα μέτρησης ώστε να αναφέρεται στο τέλος. Οι υπόλοιπες τιμές προκύπτουν με παρόμοιους τρόπους.

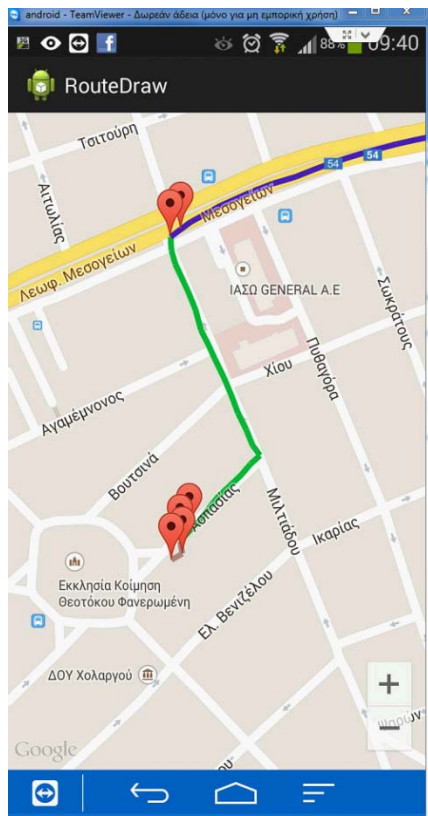


Εικόνα 53: Οι λεπτομέρειες της διαδρομής 6 μέσω του Route Summary activity.

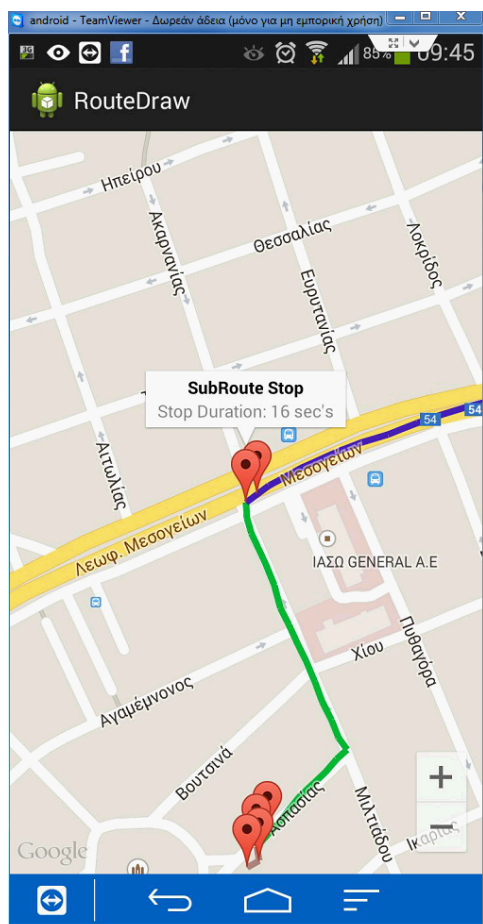
Μέσω αυτού του activity ο χρήστης έχει δύο επιλογές: είτε να ζητήσει από την εφαρμογή να του σχεδιάσει την διαδρομή είτε να δει αντίστοιχες πληροφορίες για τις υποδιαδρομές της διαδρομής. Πατώντας το κουμπί Draw This Route η συνάρτηση onClick του κουμπιού θα δημιουργήσει ένα intent ώστε να κληθεί το activity RouteDraw εισάγοντας στο intent το id της επιλεγμένης διαδρομής ώστε να γνωρίζουμε ποια διαδρομή θα σχεδιάσουμε.

Το activity RouteDraw εμφανίζει το χάρτη της Google Maps και ακολούθως αρχικοποιεί τα DAO προκειμένου να έχει πρόσβαση στα αντικείμενα της βάσης. Ανακτά από

το intent το id της διαδρομής και μέσω της συνάρτησης του DAO της υποδιαδρομής αποκτά μέσα σε λίστες όλα τα αντικείμενα υποδιαδρομών καθώς και τα αντικείμενα των σημείων της διαδρομής αυτής. Ακολουθώντας μέσω βρόχων εκτυπώνουμε τα σημεία της κάθε διαδρομής στο χάρτη και τα ενώνουμε με μια γραμμή διαφορετικού χρώματος για κάθε διαδρομή φροντίζοντας να τοποθετούμε πινέζες στα σημεία έναρξης και λήξης της κάθε υποδιαδρομής.

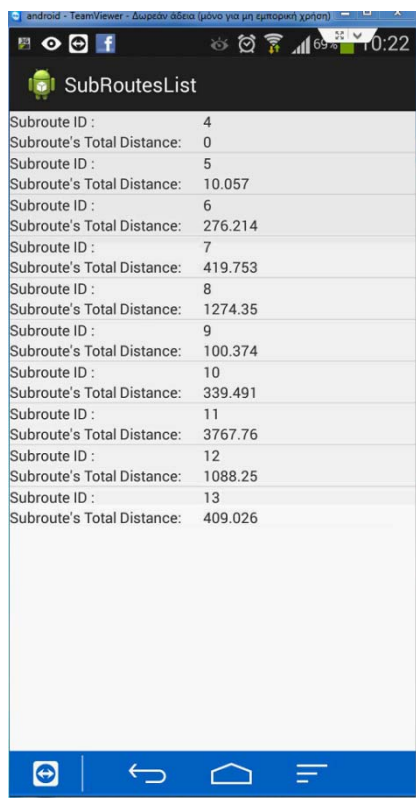


Εικόνα 54: Εμφάνιση της διαδρομής μέσω της κλάσης Route Draw. Η κλάση ανακτά τα αντικείμενα της διαδρομής και τα εκτυπώνει στο χάρτη.



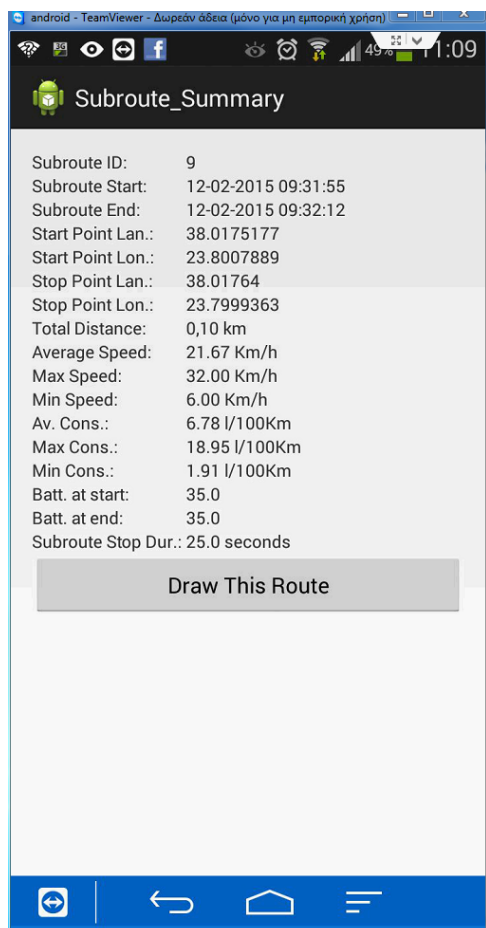
Εικόνα 55: Πατώντας στη πινέζα ολοκλήρωσης της υποδιαδρομής η εφαρμογή μέσω της συνάρτησης `subrouteStopDur` του αντικειμένου DAO της υποδιαδρομής εμφανίζει την χρονική διάρκεια της στάσης.

Πατώντας στο κουμπί Show More Details η Route Summary αντίστοιχα δημιουργεί ένα intent προκειμένου να καλέσει το activity `SubroutesList` εισάγοντας μέσα το id της διαδρομής. Το activity αυτό λειτουργεί με τρόπο αντίστοιχο με το `Routes List` που είδαμε σε προηγούμενο βήμα. Πιο συγκεκριμένα δημιουργεί τα γνωστά μας DAO, ανακτά από τη βάση τα αντικείμενα υποδιαδρομών που ανήκουν στο id διαδρομής που ανακτά από το intent και μέσω του `Simple Cursor Adapter` εμφανίζει τις τιμές τους σε ένα `ListView`. Τέλος στις εγγραφές και αυτού του `ListView` υπάρχει `onClickListener` ο οποίος όταν ενεργοποιείται δημιουργεί το `Subroute_Summary` activity μέσω ενός intent στο οποίο εισάγει το id της υποδιαδρομής που πατήθηκε.



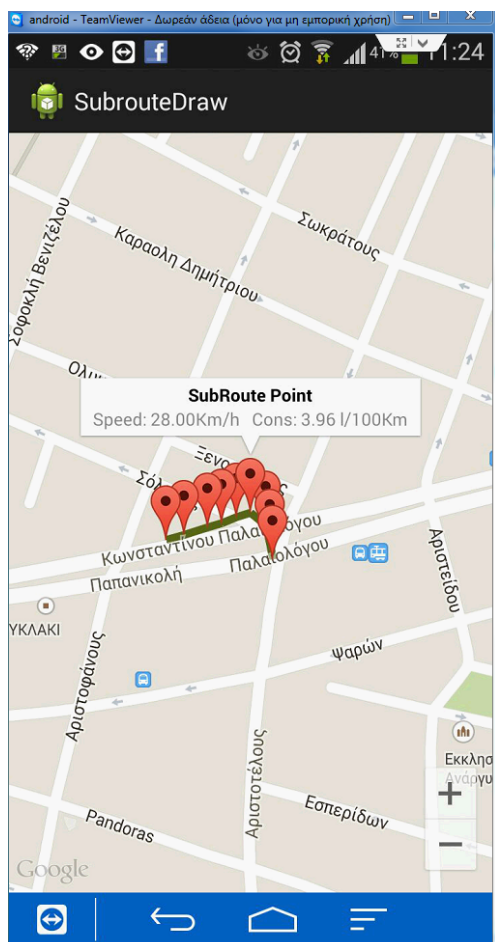
Εικόνα 56: Το SubRoutesList activity. Με τρόπο αντίστοιχο με το RoutesList η εφαρμογή ανακτά τα αντικείμενα υποδιαδρομών της διαδρομής και τα εμφανίζει σε ένα ListView.

Για την εμφάνιση των λεπτομερειών της υποδιαδρομής ακολουθείται παρόμοια λειτουργία με αυτή του Routes Summary. Πατώντας πάνω σε μια εγγραφή δημιουργείται το αντικείμενο Subroute_Summary παρέχοντας μέσα στο intent το id της υποδιαδρομής που πατήθηκε. Κατόπιν μέσω των συναρτήσεων του DAO αντικειμένου υποδιαδρομής ανακτάμε από τη βάση τις τιμές τις και τις εμφανίζουμε στο activity:



Εικόνα 57: Το `Subroute_Summary` activity. Μέσω των συναρτήσεων του DAO της υποδιαδρομής ανακτά τις τιμές της και τις εμφανίζει σε ένα πίνακα.

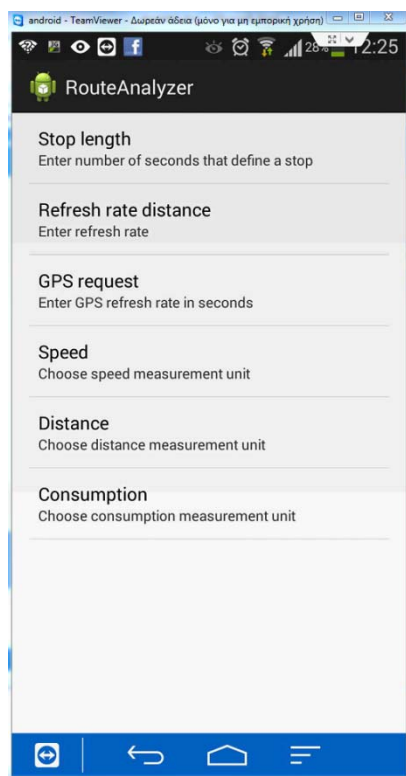
Τέλος πατώντας το `Draw This Route` η `onClick` συνάρτηση καλεί το `SubrouteDraw` activity το οποίο λειτουργεί παρόμοια με το `RouteDraw` που είδαμε σε προηγούμενο βήμα. Βασική διαφορά ότι αυτό δεν δημιουργεί αντικείμενο λίστα με υποδιαδρομές αφού διαχειρίζεται μόνο μία διαδρομή. Δημιουργεί μόνο μια λίστα με `points` τα οποία εμφανίζει πάνω στο χάρτη με πινέζες. Πάνω στις πινέζες με τις συναρτήσεις `getSpeed` και `getCons` του αντικείμενου `point` αναγράφουμε τη ταχύτητα και κατανάλωση του συγκεκριμένου σημείου:



Εικόνα 58: Το SubrouteDraw activity σχεδιάζει πάνω στο χάρτη τα points της υποδιαδρομής.

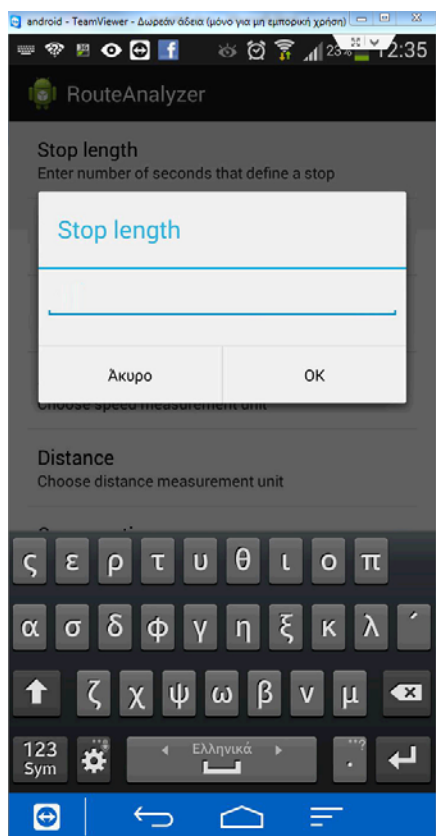
2.2.3 Η Κλάση Prefs.

Το τελευταίο αντικείμενο στο οποίο θα αναφερθούμε είναι η κλάση Prefs η οποία αποθηκεύει τις προτιμήσεις του χρήστη κατά την εκτέλεση του προγράμματος. Η κλάση αυτή καλείται όταν ο χρήστης επιλέξει από το menu το Preferences και αποθηκεύει το χρονικό διάστημα που ορίζει μια στάση, την ελάχιστη απόσταση σε μέτρα που θα πρέπει να μετακινηθεί ο χρήστης ώστε η εφαρμογή να την εκλάβει ως κίνηση, τον ρυθμό ανάγνωσης συντεταγμένων από το GPS αλλά και τις μονάδες μέτρησης της ταχύτητας, διανυθείσας απόστασης αλλά και κατανάλωσης. Οι τιμές αυτές αποθηκεύονται στη συσκευή με αποτέλεσμα η εφαρμογή να τις 'θυμάται' μεταξύ των διαφορετικών εκτελέσεων:

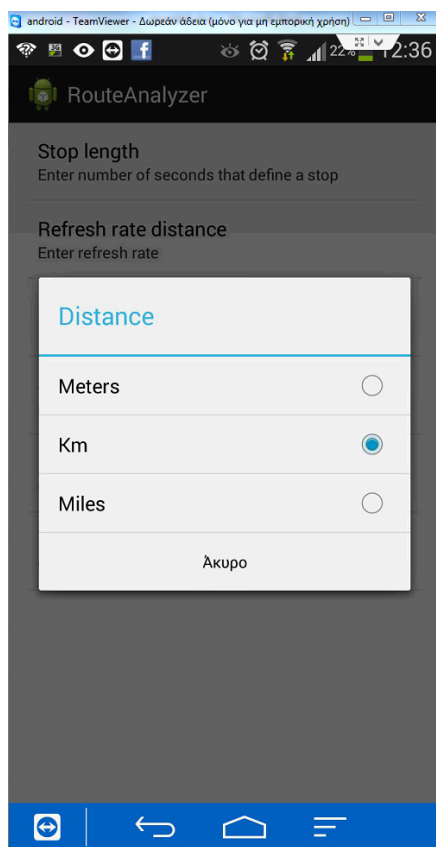


Εικόνα 59: Το activity Prefs

Η χρήση της συγκεκριμένης κλάσης είναι αρκετά τυποποιημένη και οδηγίες για τη χρήση της υπάρχουν στον επίσημο ιστότοπο της Google. Ενδεικτικά θα αναφέρουμε ότι προκειμένου να υλοποιηθεί θα πρέπει κανείς να δημιουργήσει ένα xml αρχείο στο οποίο θα εισάγει αντικείμενα τύπου `editTextPreferences` όπου θέλει να εισάγει τιμές είτε τύπου `ListPreferences`. Ο τελευταίος τύπος αντικειμένου θα εμφανίσει μια λίστα με επιλογές στο χρήστη από τις οποίες εκείνος θα μπορεί να επιλέξει μόνο μία. Οι τιμές των διάφορων λιστών αναγράφονται σε ένα xml αρχείο όπου στη περίπτωση μας είναι το `array.xml`.



Εικόνα 60: Το EditTextPreference αντικείμενο του prefs.xml αρχείου. Ο χρήστης εισάγει τον αριθμό των δευτερολέπτων που αποτελούν μια στάση.



Εικόνα 95: Το ListPreference αντικείμενο του prefs.xml αρχείου. Ο χρήστης εισάγει τη μονάδα μέτρησης απόστασης. Οι αναγραφόμενες τιμές προκύπτουν από το αρχείο array.xml

Προκειμένου τα αντικείμενα της εφαρμογής να έχουν πρόσβαση στα δεδομένα της κλάσης δημιουργούν ένα αντικείμενο τύπου Shared Preferences δίδοντας σαν όρισμα το context της εφαρμογής. Έπειτα μπορούν να εκτελέσουν τη getString συνάρτηση του δίδοντας σαν όρισμα το όνομα της τιμής που θέλουν να διαβάσουν και μια τιμή που θα χρησιμοποιηθεί στη περίπτωση που ο χρήστης δεν έχει εισάγει τιμή για το δεδομένο αυτό. Για παράδειγμα η εντολή :

```
getPrefs.getString("speed", "Km/h")
```

θα επιστρέψει ένα String με την τιμή που υπάρχει στο πεδίο με όνομα speed της κλάσης Prefs. Στη περίπτωση που ο χρήστης δεν έχει εισάγει τιμή και είναι κενή η συνάρτηση θα επιστρέψει την τιμή του δεύτερου ορίσματος και στη περίπτωση του παραδείγματος το String 'Km/H'.

3. Μελλοντικές Αναβαθμίσεις

Στο σημείο αυτό θα αναφερθούμε στις μελλοντικές αναβαθμίσεις που δύναται να υλοποιηθούν ώστε να βελτιώσουν την αξιοπιστία της εφαρμογής, αλλά και να αυξήσουν τη χρηστικότητα του.

Αρχικά, θα πρέπει να βελτιωθεί η επικοινωνία της εφαρμογής με το ολοκληρωμένο ώστε να είναι σε θέση να αντιμετωπίζει τα προβλήματα επικοινωνίας που ενδεχομένως να προκύψουν. Η εφαρμογή, στην τωρινή της έκδοση, αποστέλλει μηνύματα στο ολοκληρωμένο και περιμένει από αυτό να της στείλει απαντήσεις. Οι απαντήσεις, που περιέχονται σε μια ακολουθία χαρακτήρων, θα πρέπει να ακολουθούν συγκεκριμένη μορφή προκειμένου η εφαρμογή να αντλήσει από την ακολουθία τη τιμή που την ενδιαφέρει. Το ολοκληρωμένο όμως θα πρέπει να έχει τις εργοστασιακές ρυθμίσεις προκειμένου η απάντησή του να ακολουθεί τη συγκεκριμένη μορφή. Επιπλέον το ολοκληρωμένο αν δεν μπορεί για κάποιον λόγο να αντλήσει την πληροφορία θα απαντήσει 'NO DATA' ή INITIALIZATION PROTOCOL FAILD' αν δεν γνωρίζει το πρωτόκολλο με το οποίο θα επικοινωνήσει με το όχημα.

Για τους λόγους αυτούς αποτελεί μια σημαντική βελτίωση της εφαρμογής η δυνατότητα να ανιχνεύει αν το ολοκληρωμένο δεν έχει τις εργοστασιακές ρυθμίσεις και να το επαναφέρει. Επιπλέον θα πρέπει να μην περιμένει απλά για τις συγκεκριμένες απαντήσεις αλλά να είναι σε θέση να καταλαβαίνει όλα τα μηνύματα που της αποστέλλει το ολοκληρωμένο και να προβαίνει στις κατάλληλες ενέργειες. Πιο συγκεκριμένα όταν το ολοκληρωμένο της απαντά 'NO DATA' η εφαρμογή θα πρέπει να επανεκκινεί το ολοκληρωμένο ώστε να αρχικοποιεί εκ νέου την επικοινωνία με το όχημα. Επιπλέον όταν η αυτόματη διαδικασία εντοπισμού πρωτοκόλλου επικοινωνίας με το όχημα αποτύχει να έχει τη δυνατότητα να δοκιμάσει τα πρωτόκολλα που χρησιμοποιεί ο κατασκευαστής στο συγκεκριμένο μοντέλο. Η κίνηση αυτή προϋποθέτει την δημιουργία ενός εξυπηρετητή ο οποίος θα διαθέτει μια βάση δεδομένων με όλα τα οχήματα και τα διαθέσιμα πρωτόκολλα τους καθώς και ένα web interface. Στο interface αυτό θα πρέπει να απευθύνεται η εφαρμογή παρέχοντας του τον τύπο του οχήματος ώστε να λάβει σαν απάντηση το συγκεκριμένο πρωτόκολλο.

Σημαντική επίσης βελτίωση στη κατεύθυνση βελτίωσης της χρηστικότητας της εφαρμογής αποτελεί και η δυνατότητα της εφαρμογής να εξάγει τις καταγεγραμμένες διαδρομές. Οι διαδρομές θα πρέπει να αποστέλλονται σε ένα κεντρικό εξυπηρετητή είτε κατά την ολοκλήρωσή τους ή ακόμα καλύτερα 'ζωντανά'. Η δεύτερη τεχνική απογειώνει ακόμα περισσότερο τη χρηστικότητα μιας και οποιοσδήποτε από οπουδήποτε, μέσω μια ιστοσελίδας, θα μπορεί να παρακολουθεί τη κίνηση της συσκευής. Αυτόματα γίνεται κατανοητό στον αναγνώστη πόσο μια τέτοια λειτουργικότητα θα εξυπηρετούσε έναν πελάτη που αναμένει να του παραδοθούν τα προϊόντα της παραγγελίας του. Όμως ακόμα και η απλή καταχώρηση των διαδρομών σε ένα κεντρικό εξυπηρετητή μετά την ολοκλήρωσή τους δεν είναι μικρότερης σημασίας. Θα δύναται η δυνατότητα κεντρικού ελέγχου των διαδρομών, χωρίς να απαιτείται άμεση πρόσβαση στην ίδια τη συσκευή, μέσω της απεικόνισης τους σε μια ιστοσελίδα πάνω στο χάρτη της Google. Έτσι θα μπορούμε για παράδειγμα να εμφανίσουμε περισσότερες από μια διαδρομές της αρεσκείας μας στο χάρτη και να τις συγκρίνουμε σε χρόνο η και σε κόστος διαδρομής.

Κλείνοντας θα αναφέρουμε αναβαθμίσεις που αφορούν τη λειτουργία της εφαρμογής όπως τη δυνατότητα να υπολογίζει την κατανάλωση και για άλλους τύπους καυσίμου πέρα από τη βενζίνη καθώς και να δύναται η δυνατότητα στο χειριστή να αποθηκεύει και έτερες τιμές τις αρεσκείας του πέρα από αυτές τις ταχύτητας και κατανάλωσης. Τέλος θα μπορούσαμε να προσθέσουμε και λειτουργία για την ασφάλεια τόσο του οδηγού αλλά και του ίδιου του οχήματος. Πιο συγκεκριμένα η εφαρμογή θα μπορούσε να ενημερώνει αρμοδίως στη περίπτωση ατυχήματος, αποστέλλοντας τις συντεταγμένες, η στη περίπτωση που παρουσιαστεί πρόβλημα στη λειτουργία του οχήματος αποστέλλοντας τον κωδικό της βλάβης.

4. ΑΝΑΦΟΡΕΣ – LINKS

ⁱ Google Maps, URL: <https://www.google.gr/maps>

Πανεπιστήμιο Πειραιώς