

ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

ΚΑΤΕΥΘΥΝΣΗ: 1η
«Υπολογιστικές και Διαδικτυακές Τεχνολογίες και Εφαρμογές»

ΔΙΠΛΩΜΑΤΙΚΗ ΜΕΛΕΤΗ

ΘΕΜΑ
«ΕΞΑΓΩΓΗ ΓΝΩΣΗΣ ΑΠΟ ΔΕΔΟΜΕΝΑ ΚΙΝΗΣΗΣ ΧΡΗΣΤΩΝ»

ΛΑΜΠΟΥ ΣΟΦΙΑ (13404)

Επιβλέποντες: ΒΑΡΛΑΜΗΣ Η., ΜΙΧΑΗΛ Δ., ΔΗΜΗΤΡΑΚΟΠΟΥΛΟΣ Γ.

ΑΘΗΝΑ 2016

Πίνακας Περιεχομένων

Περίληψη	1
Κεφάλαιο 1: Εισαγωγή	2
Κεφάλαιο 1: Σκοπός-Στόχοι	4
Κεφάλαιο 2: Ανασκόπηση Βιβλιογραφίας	5
Κεφάλαιο 3: Τεχνολογίες που χρησιμοποιήθηκαν	11
Κεφάλαιο 4: Περιγραφή υλοποίησης	15
Κεφάλαιο 4.1: Η Βάση Δεδομένων	16
Κεφάλαιο 4.2: Το Back End - API	17
Κεφάλαιο 4.2: Το Back End – Functions (used by API)	18
Κεφάλαιο 4.3: Το Front End – User Interface	19
Κεφάλαιο 4.3: Το Front End – use case login - register	19
Κεφάλαιο 4.3: Το Front End – use case καταγραφή διαδρομής	25
Κεφάλαιο 4.3: Το Front End – use case data analytics	30
Κεφάλαιο 4.3: Δοκιμές – Analytics	33
Κεφάλαιο 5: Συμπεράσματα	38
Βιβλιογραφία	40
Παράρτημα	42

Περίληψη

Η παρούσα εργασία μελετά την υλοποίηση μιας εφαρμογής κινητού η οποία χρησιμοποιεί κυρίως στοιχεία που παρέχει το gps της συσκευής. Η συγκεκριμένη εφαρμογή καταγράφει τις διαδρομές του χρήστη και μετά από υπολογισμούς δίνει χρήσιμες πληροφορίες στο χρήστη της. Οι πληροφορίες αυτές δίνονται μέσα από την ίδια την εφαρμογή με τη μορφή συγκεντρωτικών στοιχείων (data analytics) αλλά και με τη μορφή ειδοποιήσεων.

Τα σύγχρονα αυτοκίνητα παρέχουν όλο και περισσότερες πληροφορίες στον οδηγό διότι είναι ενσωματωμένα με συσκευές πλοήγησης. Υπάρχουν επίσης εξωτερικές συσκευές πλοήγησης αλλά ενδεχομένως μια καλύτερη λύση να αποτελούν οι εφαρμογές κινητών συσκευών, δεδομένου ότι οι χρήστες των smart phones διαρκώς αυξάνονται. Τα smart phones και άλλες κινητές συσκευές διαθέτουν σύστημα εντοπισμού gps το οποίο χρησιμοποιείται από τις περισσότερες εφαρμογές.

Οι χρήσεις της παρούσας πρότυπης εφαρμογής είναι πολλές. Θα μπορούσε να χρησιμοποιηθεί είτε ιδιωτικά ή σε κάποια εταιρία που θα ήθελε να ελέγχει τα οχήματα της όπως για παράδειγμα σε σχολικά λεωφορεία ή εταιρίες ταξί ή ταχυμεταφορές. Πιθανό να ήταν χρήσιμη σε νέους οδηγούς για λόγους ασφάλειας να γνωρίζουν πότε υπερβαίνουν τα όρια ταχύτητας. Ο οδηγός πρέπει να είναι όσο πιο προσεκτικός μπορεί, οπότε δεν είναι δυνατόν κατά τη διαδρομή να προσέχει ταυτόχρονα το κοντέρ ταχυτήτων, τα χιλιόμετρα, την ώρα και άλλες λεπτομέρειες της διαδρομής. Πολλοί οδηγοί, είτε εταιρείες με στόλο οχημάτων θα ήθελαν να γνωρίζουν τις αποστάσεις που διανύουν τα οχήματα τους και σε τι χρονικά διαστήματα. Θα εξυπηρετούσε να ξέρουν ποιες διαδρομές έχουν περισσότερη κίνηση και μάλιστα ποιες μέρες και ώρες ώστε να τις αποφύγουν. Σε κάποιους ίσως θα είχε ενδιαφέρον να γνωρίζουν τις παραβάσεις ταχύτητας ή πόσες φορές χρειάστηκε να σταματήσουν σε κάθε διαδρομή.

Μέσα από την εργασία αυτή επιχειρήθηκε η ανάπτυξη μιας εφαρμογής σε κινητά τηλέφωνα με χρήση εργαλείων ανεξάρτητων πλατφόρμας (cross platform) η οποία θα υποστηρίζεται από μια εφαρμογή server (back-end) και μια βάση δεδομένων για τη συλλογή και διαχείριση δεδομένων με σκοπό την παραγωγή συγκεντρωτικών στοιχείων.

Η εφαρμογή μπορεί να αποτελέσει την αρχή για μια ευρύτερη προσπάθεια να βελτωθεί η κίνηση στους δρόμους μέσα από παρατηρήσεις καθημερινών διαδρομών δίνοντας προτεραιότητα στην οδική ασφάλεια. Να προστατεύσει τον οδηγό από λάθος ενέργειες και να βελτιώσει την οδήγηση και τις διαδρομές του. Να εντοπίσει τα σημεία με κυκλοφοριακή συμφόρηση και παραβάσεις ορίων ταχύτητας.

Κεφάλαιο 1: Εισαγωγή

Στην εποχή μας, το πλήθος των αυτοκινήτων αυξάνεται διαρκώς. Σύμφωνα με στοιχεία της Ελληνικής Στατιστικής Αρχής παρουσιάζεται αύξηση τόσο στο ποσοστό των οχημάτων που κυκλοφορούν για πρώτη φορά από μήνα σε μήνα, όσο και στον συνολικό αριθμό οχημάτων που βρίσκονται σε κυκλοφορία στο τέλος κάθε χρόνου. Αυτό έχει ως αποτέλεσμα την κυκλοφοριακή συμφόρηση στο οδικό δίκτυο και την ρύπανση της ατμόσφαιρας. Η κίνηση στους δρόμους μόνο προβλήματα προκαλεί στην καθημερινότητα μας. Κάθε μέρα ξοδεύουμε όλο και περισσότερο χρόνο στις μετακινήσεις μας και αντιμετωπίζουμε δυσκολία στο να είμαστε συνεπείς και να προγραμματίζουμε. Επιπλέον αυτή η καθυστέρηση δημιουργεί εκνευρισμό στους οδηγούς και τους επιβάτες, προκαλεί φθορά στους κινητήρες των οχημάτων και προκαλεί ρύπους στην ατμόσφαιρα. Οι επιπτώσεις του κυκλοφοριακού προβλήματος είναι σημαντικές στην ποιότητα ζωής εφόσον χάνουμε χρόνο και χρήμα, και δεν βοηθάει καθόλου την ψυχολογία στις δυσκολίες που έχουμε να αντιμετωπίσουμε.

Κατά καιρούς έχουν δοκιμασθεί διάφοροι τρόποι αντιμετώπισης του φαινομένου. Κατασκευάστηκαν δρόμοι και ανισόπεδες γέφυρες. Αυτοματοποιήθηκαν οι σηματοδότες. Δημιουργήθηκαν λεωφορειόδρομοι. Τα αποτελέσματα όμως δεν υπήρξαν ενθαρρυντικά καθώς ο αριθμός των ΙΧ αυξάνεται συνεχώς.

Το οδικό δίκτυο, είναι ένα από τα πιο ευρέως χρησιμοποιούμενα δίκτυα. Τα τελευταία χρόνια έχουν γίνει αρκετές προσπάθειες για την ασφαλή κυκλοφοριακή λειτουργία. Παρόλα αυτά ακόμα υπάρχει ανεπαρκής πληροφόρηση για την πραγματική κίνηση που υπάρχει στους δρόμους. Ακόμη και στην περίπτωση που υπάρχει κάποια υποδομή η οποία ελέγχει την κυκλοφορία, ελέγχει ένα συγκεκριμένο σημείο και δεν παρέχει πληροφορίες, όπως ο χρόνος διαδρομής.

Αν ο οδηγός μπορούσε να γνωρίζει την κίνηση που μπορεί να υπάρχει στην διαδρομή που πρόκειται να ακολουθήσει, θα μπορούσε να την αποφύγει, μειώνοντας έτσι τον χρόνο κάθε διαδρομής. Επίσης με ενημερωμένες πληροφορίες σχετικά με τα σημεία που πάσχουν από κίνηση μπορεί να επιτευχθεί μείωση της κυκλοφοριακής συμφόρησης και αύξηση της οδικής ασφάλειας. Η μείωση του χρόνου διαδρομής θα ελαττώσει την κατανάλωση καυσίμου και κατά συνέπεια θα μειώσει την ποσότητα των εκπομπών διοξειδίου του άνθρακα.

Με τη βοήθεια της τεχνολογίας μπορούν να αντληθούν χρήσιμες πληροφορίες που θα οδηγήσουν σε βελτίωση του προβλήματος. Μπορούν να υλοποιηθούν απλές εφαρμογές πολλών χρηστών που δεν απαιτούν ιδιαίτερο εξοπλισμό για την συγκέντρωση αυτών των πληροφοριών. Ένα μεγάλο ποσοστό των οδηγών διαθέτουν κινητά τηλέφωνα με GPS και internet τα οποία μπορούν να χρησιμοποιήσουν οι εφαρμογές για να συγκεντρώσουν δεδομένα αθόρυβα. Στη συνέχεια το ιδανικό θα ήταν να μελετηθούν αυτά τα στοιχεία από συγκοινωνιολόγους ούτως ώστε να τα λάβουν υπόψη κατά το σχεδιασμό νέων συστημάτων κυκλοφορίας και μεταφορών και ρύθμισης κυκλοφορίας. Με τις κατάλληλες ενέργειες τα μέσα μαζικής μεταφοράς θα καλύψουν τις περιοχές που έχουν πρόβλημα και θα δώσουν κίνητρο στους οδηγούς να τα χρησιμοποιούν πιο συχνά. Η πληροφορική μπορεί να βοηθήσει αν όχι στην εξάλειψη σε μείωση του φαινομένου της κίνησης.

Η εργασία εστίασε στα προβλήματα που υπάρχουν από την έλλειψη γνώσης απλών στοιχείων πλοήγησης και πως αντιμετωπίζονται με τη βοήθεια της πληροφορικής. Ερεύνησε βιβλιογραφικά και στην πράξη, το πόσο εύκολη είναι η συλλογή δεδομένων και η εξαγωγή αναλυτικών στοιχείων από αυτά. Μέσα από την εφαρμογή που αναπτύχθηκε είναι πλέον δυνατό να εκμεταλλευτούμε την καταγραφή πληροφοριών για να βελτιωθεί η καθημερινή κυκλοφορία των οδηγών και των οχημάτων.

Ανάμεσα στα πολλά προβλήματα που εντοπίστηκαν στη σχετική βιβλιογραφία και έχουν να κάνουν με την κίνηση οχημάτων, απουσία τέτοιων εφαρμογών και λύσεων είναι και τα ακόλουθα:

Ο οδηγός παραβιάζει τα όρια ταχύτητας χωρίς να το καταλάβει. Όταν δεν υπάρχει κίνηση στους δρόμους είναι εύκολο να επιταχύνει ασυναίσθητα.

Κάποιος που έχει στην διάθεση του πολλά οχήματα (πχ. Ταξί, courier, λεωφορεία) είναι δύσκολο να τα ελέγξει. Δεν μπορεί να παρατηρήσει ταυτόχρονα την οδική συμπεριφορά όλων των οδηγών ούτε να υπολογίσει πόσα χιλιόμετρα διανύονται.

Συχνά δημιουργείται η ανάγκη να συγκρίνουμε διαδρομές. Κάποιες μέρες και ώρες η ίδια διαδρομή παρουσιάζει διαφορετική κυκλοφοριακή συμφόρηση. Επιπλέον κάποιος χρειάζεται να γνωρίζουν πως διαφοροποιούνται διαδρομές ως προς την απόσταση ή τη διάρκεια.

Δεν πρέπει να αποσπάται η προσοχή του οδηγού σε καμία περίπτωση και ως εκ τούτου δεν είναι δυνατών να αποτυπώνει λεπτομέρειες των διαδρομών στη μνήμη του. Θα βοηθούσε μια εφαρμογή εντοπισμού και καταγραφής κάποιον που έχει ανάγκη να ενημερώνεται διαρκώς για τις διαδρομές που διανύει. Είναι συνηθισμένο να μπερδεύουμε ημέρες και διαδρομές ειδικά αν κυκλοφορούμε καθημερινά. Επίσης συχνό φαινόμενο είναι να αφαιρούνται οι οδηγοί όταν δεν υπάρχει κίνηση και να μην συνειδητοποιούν ότι αυξάνουν υπερβολική ταχύτητα.

Σκοπός-Στόχοι:

Σκοπός της παρούσας εργασίας ήταν κατ' αρχή να μελετήσει αν υπάρχει αντίστοιχο λογισμικό στο εξωτερικό και κατά πόσο αυτό βελτίωσε την ποιότητα των διαδικασιών. Να εντοπίσει τυχόν προβλήματα που συναντήθηκαν και να δει πώς αυτά προσπεράστηκαν. Ταυτόχρονα να διερευνήσει τα πιθανά οφέλη από την εφαρμογή. Για το σκοπό αυτό, στα πλαίσια της εργασίας υλοποιήθηκε μία πρότυπη εφαρμογή κινητού η οποία καταγράφει τις διαδρομές κάθε χρήστη σε μία βάση δεδομένων και σε δεύτερο χρόνο επιτρέπει στο χρήστη να έχει πρόσβαση σε μια παρουσίαση συγκεντρωτικών στοιχείων σχετικά με την ταχύτητα, την απόσταση και τους χρόνους των προσωπικών του διαδρομών που έχουν καταγραφεί από την εφαρμογή.

Η βιβλιογραφική έρευνα που παρουσιάζεται στο κεφάλαιο 2 έχει ως αντικείμενο εφαρμογές κινητών συσκευών εντοπισμού και κυρίως αυτές που ανιχνεύουν και καταγράφουν τις θέσεις της συσκευής. Στα πλαίσια της έρευνας αυτής τέθηκαν οι ακόλουθοι 3 στόχοι.

- **Αντικείμενο-Στόχος 1:** Μελέτη της σχετικής βιβλιογραφίας που υπάρχει στο διαδίκτυο. Ανάλυση παρόμοιων περιπτώσεων και συγκέντρωση των πιο σημαντικών από αυτές.
- **Αντικείμενο-Στόχος 2:** Κριτική αξιολόγηση σε αντίστοιχες εφαρμογές που έχουν αναπτυχθεί τα τελευταία χρόνια στο εξωτερικό. Για το σκοπό αυτό έπρεπε να γνωρίζουμε αν η εφαρμογή ήταν χρήσιμη και αν άξιζε η προσπάθεια να υλοποιηθεί.
- **Αντικείμενο-Στόχος 3:** Ανάλυση των πλεονεκτημάτων και μειονεκτημάτων από τη χρήση ενός τέτοιου λογισμικού. Από την ανάλυση αυτή προέκυψαν χρήσιμα συμπεράσματα.

Το επόμενο στάδιο ήταν η σχεδίαση και υλοποίηση της εφαρμογής. Στα πλαίσια αυτά τέθηκαν οι ακόλουθοι στόχοι:

- **Αντικείμενο-Στόχος 4:** Εντοπισμός των εργαλείων που θα χρησιμοποιηθούν και σχεδίαση της λύσης.
- **Αντικείμενο-Στόχος 5:** Υλοποίηση της εφαρμογής κινητού που ανιχνεύει, καταγράφει διαδρομές και παρέχει αναλυτικά στοιχεία για αυτές.

Δομή Εργασίας:

Η παρούσα εργασία αποτελείται από 5 κεφάλαια

Στο κεφάλαιο 1: Γίνεται μία εισαγωγή στη θεματική περιοχή, τους σκοπούς και το αντικείμενο.

Το κεφάλαιο 2: Παρουσιάζει τη σχετική βιβλιογραφία (κριτική, ανάλυση, συμπεράσματα).

Το κεφάλαιο 3: Περιγράφει τις τεχνολογίες και τα εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής.

Στο κεφάλαιο 4: Παρουσιάζεται η ανάλυση και σχεδιασμός της εφαρμογής.

Στο κεφάλαιο 5: Συμπεράσματα που προέκυψαν κατά την υλοποίηση και τη χρήση της εφαρμογής

Κεφάλαιο 2: Ανασκόπηση Βιβλιογραφίας

Στα πλαίσια της εργασίας αναζητήθηκαν επιστημονικά άρθρα που έχουν δημοσιευτεί στο διαδίκτυο. Διερευνήθηκε αν υπάρχει αντίστοιχη εφαρμογή στο εξωτερικό και τι επιπτώσεις είχε. Αναλυτικότερα προέκυψαν τα παρακάτω:

Η χρήση των κινητών συσκευών έχει γίνει μέρος της καθημερινότητας μας. Οι φορητές συσκευές όπως κινητά τηλέφωνα ή φορητές ψηφιακές οθόνες (PDA) είναι εξοπλισμένα με αισθητήρες παγκόσμιου συστήματος εντοπισμού θέσης (GPS) που επιτρέπουν να πάρουμε τη γεωγραφική θέση της συσκευής, σε πραγματικό χρόνο. Οι Varandas, et al (2010) περιγράφουν την εφαρμογή ενός εργαλείου ανίχνευσης, που ονομάζεται κινητός ανιχνευτής, το οποίο χρησιμοποιεί υπηρεσίες με βάση τη θέση (LBs) όπως το GPS ή το παγκόσμιο σύστημα δικτύου κινητής τηλεφωνίας (GSM) για να ανιχνεύει μια κινητή συσκευή. Μέσα από τη γνωστή γεωγραφική θέση, αυτή η εφαρμογή δίνει τη δυνατότητα στο χρήστη να παρακολουθεί μια κινητή συσκευή και στέλνει ειδοποιήσεις αν είναι σε ακτίνα γύρω από ένα σημείο ενδιαφέροντος, που ορίστηκε προηγουμένως από τον διαχειριστή της εφαρμογής.

Το παγκόσμιο σύστημα εντοπισμού θέσης (GPS) χρησιμοποιείται ευρέως για την ανίχνευση και παρακολούθηση οχημάτων. Πολλά συστήματα έχουν δημιουργηθεί για να παρέχουν τέτοιες υπηρεσίες που τα καθιστούν δημοφιλή και χρειάζονται περισσότερο από ποτέ. Το πανεπιστήμιο της Ιορδανίας προτείνει ένα "GPS σύστημα παρακολούθησης οχήματος". Αυτό το σύστημα είναι χρήσιμο σε φορείς εκμετάλλευσης στόλου για παρακολούθηση της οδικής συμπεριφοράς των εργαζομένων ή σε γονείς για παρακολούθηση εφήβων οδηγών. Επιπλέον, αυτό το σύστημα μπορεί να χρησιμοποιηθεί στην πρόληψη κλοπής ως συσκευή ανάκτησης, εκτός του ότι εργάζεται ως σύστημα ασφαλείας σε συνδυασμό με τους συναγερμούς αυτοκινήτων. Η κύρια συνεισφορά του συστήματος είναι η παροχή δύο τύπων εφαρμογών τελικού χρήστη, μια web εφαρμογή και μια εφαρμογή για κινητά. Με αυτό τον τρόπο το προτεινόμενο σύστημα παρέχει ένα πανταχού παρόν σύστημα εντοπισμού οχημάτων με μέγιστη προσβασιμότητα για τον χρήστη ανά πάσα στιγμή και από οπουδήποτε. Οι υπηρεσίες παρακολούθησης του συστήματος περιλαμβάνουν την απόκτηση της θέσης και την ταχύτητα εδάφους ενός δεδομένου οχήματος κατά την τρέχουσα στιγμή ή σε οποιαδήποτε προγενέστερη ημερομηνία. Παρακολουθεί επίσης το όχημα ρυθμίζοντας όρια ταχύτητας και γεωγραφικά και ως εκ τούτου λαμβάνει ειδοποιήσεις SMS όταν το όχημα υπερβεί αυτά τα προκαθορισμένα όρια. Επιπλέον, όλες οι κινήσεις και οι στάσεις ενός συγκεκριμένου οχήματος μπορούν επίσης να καταγραφούν. Η παρακολούθηση οχημάτων στο σύστημά χρησιμοποιεί ένα ευρύ φάσμα νέων τεχνολογιών και δικτύων επικοινωνίας, συμπεριλαμβανομένων General Packet Radio Service (GPRS), Παγκόσμιο Σύστημα Κινητών Επικοινωνιών (GSM), το Διαδίκτυο ή το World Wide Web και το Παγκόσμιο Σύστημα Εντοπισμού Θέσης (GPS). (Almomani et al, 2011)

Η μελέτη των Behzada et al, (2014) προτείνει, το σχεδιασμό και την ανάπτυξη ενός συστήματος εντοπισμού, στο οποίο τα οχήματα παρακολουθούνται και ελέγχονται με τη χρήση κινητών τεχνολογιών. Το σύστημα περιλαμβάνει ένα δέκτη GPS και ένα μόντεμ GSM διασυνδεδεμένο με έναν μικροελεγκτή. Για να παρακολουθηθεί οποιοδήποτε όχημα, ο ιδιοκτήτης του οχήματος πρέπει να στείλει ένα SMS με το σύστημα παρακολούθησης που έχει εγκατασταθεί στο εσωτερικό του οχήματος. Μετά τη λήψη του SMS, ο μικροελεγκτής λαμβάνει τις συντεταγμένες της τρέχουσας τοποθεσίας από το δέκτη GPS, τις ενσωματώνει

σε ένα SMS και τις στέλνει στον ιδιοκτήτη και σε ένα web server χρησιμοποιώντας μόντεμ GSM. Όταν ο web server λαμβάνει το SMS που περιέχει τις συντεταγμένες των οχημάτων, δείχνει τη θέση του οχήματος στους Χάρτες Google. Για τους χρήστες Android, η θέση εμφανίζεται επίσης σε μια εφαρμογή Android. Σε περίπτωση κλοπής του οχήματος, ο ιδιοκτήτης είναι σε θέση να απενεργοποιήσει τον κεντρικό διακόπτη μίζας, να ελέγξει την κατάσταση και την ταχύτητα του οχήματος απλά με την αποστολή ενός SMS. Το σύστημα είναι επίσης εξοπλισμένο με ένα ειδικό κουμπί ασφαλείας για τα σταθμευμένα οχήματα. Με το πάτημα του κουμπιού στο ON, το σύστημα θα έρθει στην ACTIVE λειτουργία και θα τηρεί ειδικό έλεγχο στη διακίνηση του οχήματος ενώ εκτελεί τα συνήθη καθήκοντα. Αν το σύστημα ανιχνεύσει οποιαδήποτε μετακίνηση του οχήματος κατά την ενεργή λειτουργία, θα γυρίσει τον κύριο διακόπτη μίζας στη θέση OFF και θα ενημερώσει τον ιδιοκτήτη αμέσως στέλνοντας 5 SMS. Η καταγραφή της κίνησης του οχήματος διαχειρίζεται συνεχώς στο web server όπου κάθε ιδιοκτήτης έχει λογαριασμό οχήματος. Έχουν χρησιμοποιηθεί οι τεχνολογίες: Παγκόσμιο Σύστημα Εντοπισμού Θέσης (GPS), Παγκόσμιο Σύστημα Κινητής Επικοινωνίας (GSM) και μικροελεγκτής.

Σύμφωνα με μελέτη των Abbott & Powell που δημοσιεύτηκε το 2002 (Abbott & Powell, 2002) το Global Positioning System (GPS) έχει κάνει τα συστήματα πλοήγησης πρακτικά για έναν αριθμό εφαρμογών πλοήγησης οχημάτων. Τα συστήματα πλοήγησης GPS μπορεί να βρεθούν σε μηχανοκίνητα οχήματα, στη γεωργία και τον εξοπλισμό εξόρυξης, καθώς και μια ποικιλία άλλων χερσαίων οχημάτων. Συγκεκριμένα, αυτή η εργασία παρουσιάζει μια ποσοτική εξέταση των επιπτώσεων που έχουν ξεχωριστοί αισθητήρες πλοήγησης στην απόδοση ενός συστήματος πλοήγησης οχημάτων. Εξετάζονται μια σειρά από επίπεδα απόδοσης αισθητήρα πλοήγησης και η επιρροή τους στην ακρίβεια εντοπισμού του οχήματος. Τα αποτελέσματα δείχνουν ότι, για ένα τυπικό σύστημα πλοήγησης, το σφάλμα τοποθεσίας κυριαρχείται από την ακρίβεια των στιγμάτων θέσης που παρέχονται από το δέκτη GPS. Επιπλέον, τα αποτελέσματα δείχνουν ότι η ακρίβεια των στιγμάτων θέσης GPS έχει σημαντικό αντίκτυπο στις σχετικές συνεισφορές που δίνει κάθε σφάλμα του αισθητήρα πλοήγησης.

Πιο πρόσφατη μελέτη (Amini & Buehrer, 2014) μελετά το πρόβλημα της κινητής παρακολούθησης σε πυκνά περιβάλλοντα. Το Global Positioning System (GPS) είναι η πιο προσπή τεχνική τοποθεσίας. Ωστόσο, το GPS δεν λειτουργεί σωστά σε εσωτερικούς χώρους και πυκνές περιοχές, καθώς ο δέκτης συνήθως δεν έχει πρόσβαση σε επαρκή αριθμό δορυφόρων. Ως εκ τούτου, ο εντοπισμός σε αυτά τα δίκτυα μπορεί να γίνει εναλλακτικά με τη χρήση μετρήσεων που συλλέγονται στο πλαίσιο του δικτύου και χωρίς τη βοήθεια τυχόν εξωτερικών πόρων (π.χ. GPS). Το πρόβλημα εντοπισμού κινητών περιλαμβάνει αρκετούς στατικούς κόμβους αναφοράς του οποίου οι θέσεις είναι σταθερές και γνωστές, και πολλούς κινητούς κόμβους των οποίων οι θέσεις είναι άγνωστες και χρειάζεται να προσδιοριστούν. Το πρόβλημα της κινητής παρακολούθησης μπορεί να λυθεί σε δύο μορφές: κεντρική και κατακεντρωμένη. Ένας κεντρικός αλγόριθμος μπορεί να οδηγήσει σε υψηλή πολυπλοκότητα και λανθάνουσα κατάσταση, ενώ ένας κατακεντρωμένος αλγόριθμος μπορεί να οδηγήσει σε μεγάλα λάθη εκτίμησης. Σε αυτή τη μελέτη, εισάγεται μια νέα τεχνική εντοπισμού η οποία είναι σε θέση να παραδώσει ακρίβεια εντοπισμού, ενώ διατηρεί την καθυστέρηση και την πολυπλοκότητα όσο το δυνατόν χαμηλότερα. Η απόδοση του προτεινόμενου αλγορίθμου συγκρίνεται με εκείνες των άλλων αλγορίθμων όσον αφορά την ακρίβεια εντοπισμού, τη λανθάνουσα κατάσταση, και απαιτείται επικοινωνία δεδομένων μέσω προσομοιώσεων

υπολογιστή. Τα αποτελέσματα της προσομοίωσης δείχνουν την αποτελεσματικότητα του προτεινόμενου αλγορίθμου σε σύγκριση είτε με κεντρικούς ή κατανεμημένους αλγορίθμους. Μια σημαντική εφαρμογή της παρούσας εργασίας είναι ο εντοπισμός οχήματος σε πυκνά περιβάλλοντα όπου τα οχήματα δεν έχουν πρόσβαση στους δορυφόρους GPS και πρέπει να εντοπίζονται από στοιχεία μέσα στο δίκτυο.

Σύμφωνα με τους Chew et al (2006), οι κινητές συσκευές με Internet και τα δίκτυα κινητών επικοινωνιών τρίτης γενιάς έχουν γίνει πραγματικότητα. Οι υπηρεσίες με βάση την τοποθεσία (LBS) αναμένεται να αποτελέσουν σημαντικό τομέα ανάπτυξης. Η παροχή πληροφοριών, περιεχομένου και υπηρεσιών μέσω των τεχνολογιών εντοπισμού θέσης αποτελεί την πλατφόρμα για νέες υπηρεσίες για τους χρήστες και τους προγραμματιστές, καθώς και τη δημιουργία νέων καναλιών εσόδων για τους παρόχους υπηρεσιών. Αυτή η πρόοδος στον τομέα των υπηρεσιών με βάση την τοποθεσία έχει βοηθήσει τον εντοπισμό των ασθενών σε πραγματικό χρόνο σε εφαρμογές προσωπικής υγειονομικής περίθαλψης. Σε αυτή την εργασία, μια υβριδική τεχνική εντοπισμού κινητών που βασίζονται στη χρήση του παγκόσμιου συστήματος εντοπισμού θέσης (GPS) και την κυψελοειδή υποδομή δικτύου κινητής τηλεφωνίας χρησιμοποιείται για να παρέχει τη δυνατότητα εντοπισμού θέσης. Η λειτουργία αυτή θα πρέπει να ενσωματωθεί στο σύστημα εντοπισμού θέσης των ασθενών (PLTS) για να βοηθήσει τους φροντιστές ή μέλη της οικογένειας στον εντοπισμό ασθενών όταν απαιτείται, ιδιαίτερα σε καταστάσεις έκτακτης ανάγκης. Η ικανότητα αυτού του PLTS αποδεικνύεται μέσα από μια σειρά τεστ ανίχνευσης θέσης που διεξάγονται σε διαφορετικές συνθήκες λειτουργίας. Αν και το μοντέλο είναι σε αρχικό στάδιο ανάπτυξης, έχει αποδείξει σχετικά καλή ακρίβεια στον εντοπισμό θέσης και τη δυνατότητα χρήσης ενσωματωμένης ασύρματης τεχνολογίας για να ενισχύσει το υπάρχον σύστημα προσωπικής επικοινωνίας υγειονομικής περίθαλψης μέσω υπηρεσιών που βασίζονται στη θέση.

Η δυνατότητα να παρακολουθούνται τα οχήματα είναι χρήσιμη σε πολλές εφαρμογές συμπεριλαμβανομένης της ασφάλειας των προσωπικών οχημάτων, τα συστήματα μαζικής μεταφοράς, τη διαχείριση στόλου και άλλα. Επιπλέον, ο αριθμός των οχημάτων στο δρόμο σε παγκόσμιο επίπεδο αναμένεται να αυξηθεί ραγδαία. Ως εκ τούτου, η ανάπτυξη συστήματος εντοπισμού οχήματος με χρήση του Παγκόσμιου Συστήματος Εντοπισμού Θέσης (GPS) και Παγκοσμίου Συστήματος Κινητών Επικοινωνιών (GSM) modem γίνεται με σκοπό να επιτρέπει στους χρήστες να εντοπίσουν τα οχήματά τους με ευκολία και ένα βολικό τρόπο. Το σύστημα θα παρέχει στους χρήστες τη δυνατότητα παρακολούθησης του οχήματος από απόσταση μέσω του δικτύου κινητής τηλεφωνίας. Η εργασία των Pham, Driberg & Chi (2013) παρουσιάζει την ανάπτυξη πρωτότυπου υλικού συστήματος εντοπισμού οχήματος. Συγκεκριμένα, το σύστημα χρησιμοποιεί το GPS για να αποκτήσει τις συντεταγμένες του οχήματος και τις μεταδίδει χρησιμοποιώντας μόντεμ GSM στο τηλέφωνο του χρήστη μέσω του δικτύου κινητής τηλεφωνίας. Το σύστημα εντοπισμού οχήματος που αναπτύχθηκε αποδεικνύει τη επίτευξη της σε σχεδόν πραγματικό χρόνο παρακολούθησης οχημάτων και τη βελτιωμένη προσαρμοστικότητα, παγκόσμια λειτουργικότητα και κόστος σε σύγκριση με τις υπάρχουσες λύσεις.

Ο εντοπισμός GPS έχει πολλές χρήσεις στον σημερινό κόσμο. Το σύστημα μπορεί να χρησιμοποιηθεί για παρακολούθηση παιδιών, περιουσιακά στοιχεία, αυτοκίνητα ή οποιοδήποτε εξοπλισμό εντοπισμού και κατασκοπίας. Ο Hind (2014) παρουσιάζει ένα ακριβές

και αξιόπιστο σύστημα εντοπισμού σε πραγματικό χρόνο με χρήση GPS (Global Positioning System) και υπηρεσίες GSM (παγκόσμιο σύστημα κινητής επικοινωνίας), το οποίο σχεδιάστηκε και υλοποιήθηκε με επιτυχία στα εργαστήρια του πανεπιστημίου του Χαρτούμ. Το σύστημα επιτρέπει τον εντοπισμό μιας φορητής μονάδας και διαβιβάζει τη θέση στο κέντρο παρακολούθησης. Το σύστημα εντοπισμού GPS αποτελείται από φορητές εντοπιζόμενες συσκευές που συνδέονται με ένα πρόσωπο, όχημα ή οποιοδήποτε περιουσιακό στοιχείο, και το κέντρο παρακολούθησης, όπου θα πρέπει να παρακολουθούνται οι φορητές συσκευές. Η κινητή συσκευή λαμβάνει τις συντεταγμένες από το GPS και τις στέλνει με SMS μέσω GSM μόντεμ στο κέντρο παρακολούθησης, το οποίο είναι απλά ένας υπολογιστής με πολλά προγράμματα για να εμφανίσει τη θέση στο Google Maps, χρησιμοποιώντας τη δωρεάν έκδοση των Google Maps APIs. Οι δοκιμές δείχνουν ότι το σύστημα πληροί το στόχο του να είναι χαμηλού κόστους, ακριβές, σε πραγματικό χρόνο και να προσαρμόζεται σε διάφορες εφαρμογές.

Ένα αποτελεσματικό σύστημα εντοπισμού οχήματος έχει σχεδιαστεί για την παρακολούθηση κίνησης κάθε εξοπλισμένου οχήματος σε οποιαδήποτε θέση ανά πάσα στιγμή (Lee, Tewolde & Kwon, 2014). Το προτενόμενο σύστημα έκανε καλή χρήση μιας δημοφιλούς τεχνολογίας που συνδυάζει μια εφαρμογή smartphone με ένα μικροελεγκτή. Αυτό θα είναι εύκολο να γίνει και φθινό σε σύγκριση με τα άλλα. Η σχεδιασμένη εντός οχήματος συσκευή λειτουργεί χρησιμοποιώντας Global Positioning System (GPS) και το Παγκόσμιο σύστημα κινητής τηλεφωνίας / General Packet Radio Service (GSM / GPRS), που είναι ένας από τους πιο κοινούς τρόπους για τον εντοπισμό οχήματος. Η συσκευή είναι ενσωματωμένη σε ένα όχημα του οποίου η θέση πρέπει να προσδιορίζεται και να παρακολουθείται σε πραγματικό χρόνο. Ένας μικροελεγκτής χρησιμοποιείται για τον έλεγχο του GPS και των μονάδων GSM / GPRS. Το σύστημα εντοπισμού του οχήματος χρησιμοποιεί τη μονάδα GPS για να πάρει γεωγραφικές συντεταγμένες σε τακτά χρονικά διαστήματα. Η μονάδα GSM / GPRS χρησιμοποιείται για να μεταδώσει και να ενημερώσει μια βάση δεδομένων με τη θέση του οχήματος. Μια εφαρμογή smartphone έχει επίσης αναπτυχθεί για τη συνεχή παρακολούθηση της θέσης του οχήματος. Το API Χαρτών Google χρησιμοποιείται για να εμφανίσει το όχημα στο χάρτη στην εφαρμογή Smartphone. Έτσι, οι χρήστες θα είναι σε θέση να παρακολουθούν συνεχώς ένα κινούμενο όχημα, χρησιμοποιώντας την εφαρμογή Smartphone και να καθορίσουν την αναμενόμενη απόσταση και το χρόνο για να φτάσει το όχημα σε ένα συγκεκριμένο προορισμό. Για να αποδείξει τη σκοπιμότητα και την αποτελεσματικότητα του συστήματος, το σχετικό έγγραφο παρουσιάζει τα πειραματικά αποτελέσματα του συστήματος παρακολούθησης του οχήματος και κάποιες εμπειρίες σχετικά με τις πρακτικές εφαρμογές.

Αρκετές εφαρμογές εντοπισμού θέσης και παρακολούθησης με βάση το GPS πρέπει να είναι πολύ συντηρητικές όσον αφορά την κατανάλωση ενέργειας. Μερικές από αυτές τις εφαρμογές μπορεί ακόμη και να διακινδυνεύσουν την απόδοση σε πραγματικό χρόνο και την ακρίβεια με χαμηλότερη κατανάλωση ενέργειας. Κατά τη διαδικασία της απόκτησης μιας θέσης, η λήψη GPS παίρνει πολύ χρόνο, ειδικά σε κακές συνθήκες σήματος GPS, προκαλώντας σημαντική κατανάλωση ενέργειας στα συστήματα με δέκτη GPS. Αυτός είναι ένας από τους σημαντικότερους λόγους για τους οποίους οι δέκτες GPS έχουν κακές επιδόσεις κάτω από παχιά κάλυψη όπως το φύλλωμα και το μπετό. Το GSM χρησιμοποιείται για να λύσει αυτό το πρόβλημα, όπου η συνδεσιμότητα είναι διαθέσιμη. Ωστόσο, τα

υπάρχοντα συστήματα δεν λειτουργούν σε απομακρυσμένες περιοχές, όπου τα δίκτυα κινητής τηλεφωνίας δεν παρέχουν κάλυψη. Άλλα συστήματα εστιάζονται κυρίως στη βελτίωση της ακρίβειας και της αξιοπιστίας του GPS, αλλά δεν βοηθούν στο να πάρει γρηγορότερα στίγμα το GPS (Patil et al, 2011).

Τα Location Based Services (LBS) θεωρούνται βασικό χαρακτηριστικό πολλών μελλοντικών εφαρμογών κινητών. Το GPS εξυπηρετεί τις περισσότερες εξωτερικές εφαρμογές, ωστόσο, η εξάρτησή του από δορυφόρους το καθιστά αναποτελεσματικό για εσωτερικούς χώρους. Σε διεθνές συνέδριο παρουσιάστηκε μια νέα τεχνική (Chandra, et al, 2011), που στέλνονται συντεταγμένες GPS σε άλλα κινητά με Υπηρεσία σύντομων μηνυμάτων (SMS), με βάση την Global Positioning System (GPS) τεχνολογία. Αυτή η εφαρμογή επιτρέπει επίσης στους χρήστες να πάρουν τις τρέχουσες συντεταγμένες της θέσης τους να τις δουν σε χάρτες Google. Περαιτέρω, η εφαρμογή αυτή επιτρέπει επίσης στο χρήστη να μοιραστεί την τοποθεσία του με τους φίλους τους μέσω ενός web server χρησιμοποιώντας σύνδεση στο internet σε φορητές συσκευές. Το GPS είναι ένα δορυφορικό σύστημα πλοήγησης, που αποτελείται από ένα δίκτυο 24 δορυφόρων τοποθετημένων σε τροχιά από το Υπουργείο Άμυνας των Ηνωμένες Πολιτειών (ΗΠΑ). Το GPS αρχικά προοριζόταν για στρατιωτικές εφαρμογές, αλλά στη δεκαετία του 1980, η κυβέρνηση έκανε το σύστημα διαθέσιμο για χρήση από τους πολίτες. Το GPS μπορεί να δείξει την ακριβή θέση πάνω στη Γη σε οποιοδήποτε καιρικές συνθήκες, οπουδήποτε στον κόσμο, 24 ώρες την ημέρα. Δεν υπάρχουν συνδρομές ή χρεώσεις εγκατάστασης χρήσης του GPS.

Το Αυτόνομο σύστημα ανίχνευσης και εντοπισμού θέσης ενισχύει την ακρίβεια εντοπισμού φίλων με χρήση GPS και πρότυπη τεχνολογία web. Σε άρθρο των Kadibagil & Guruprasad, (2014) παρουσιάζεται σύστημα που περιλαμβάνει ένα κινητό, ένα χώρο αποθήκευσης, ένα πρόγραμμα web και μια υπηρεσία χαρτών. Το κινητό χρησιμοποιείται για να βρεθεί η θέση και να σταλεί ένα Popup SMS στο χρήστη όταν οι φίλοι του έρχονται γύρω από την περιοχή κατεύθυνσης του χρήστη. Αυτές οι πληροφορίες τοποθεσίας μπορούν να σταλούν στο διακομιστή και οι ίδιες πληροφορίες μπορούν να προβληθούν από άλλους χρήστες χρησιμοποιώντας την εφαρμογή web. Ως μελλοντική εργασία, το προτεινόμενο σύστημα μπορεί να ενημερώνει τις λεπτομέρειες θέσης σε ιστοσελίδες κοινωνικής δικτύωσης.

Τα Δημόσια δίκτυα μεταφορών είναι δύσκολο να χρησιμοποιηθούν όταν οι χρήστες δεν είναι εξοικωμένοι με την περιοχή που ταξιδεύουν. Σε αυτές τις περιπτώσεις, κατάλληλες πληροφορίες πλοήγησης διαδρομής μπορεί να διευκολύνουν σημαντικά τη χρήση των δημόσιων μέσων μεταφοράς και οι ταξιδιώτες να τα προτιμούν έναντι των άλλων τρόπων μεταφοράς. Ο εντοπισμός ενός χρήστη, είναι κρίσιμης σημασίας για την παροχή των σχετικών πληροφοριών. Στην ανάλυση των Mane & Khairnar (2014) αξιολογείται ο σχεδιασμός μιας οικονομικής, αποδοτικής υπηρεσίας πλοήγησης φιλική προς το χρήστη που χρησιμοποιεί αισθητήρες θέσης. Βοηθώντας τους ταξιδιώτες να περάσουν από τα οχήματα στα δημόσια συστήματα μεταφορών, οι κοινότητες μπορούν να μειώσουν την κυκλοφοριακή συμφόρηση, καθώς και τις περιβαλλοντικές επιπτώσεις. Γίνονται προσπάθειες ικανοποίησης των σημερινών χρηστών μέσων μαζικής μεταφοράς και παρακίνησης περισσότερων ανθρώπων να τα χρησιμοποιούν. Μπορούν να βοηθηθούν οι υπάρχοντες αναβάτες και να ενθαρρυνθούν νέοι με την ενίσχυση της χρηστικότητας των δημοσίων μεταφορών, μέσω καλών συστημάτων πληροφόρησης ταξιδιωτών. Το κίνητρο για κάθε σύστημα

πληροφόρησης είναι: "Να βοηθήσει με την ακριβή πληροφόρηση, στο σωστό μέρος, σε πραγματικό χρόνο με εξατομικευμένες εγκαταστάσεις και ευαισθησία θέσης». Οι υπηρεσίες που βασίζονται στην τοποθεσία είναι όλο και πιο σημαντικές για τις σύγχρονες φορητές συσκευές όπως τα Smartphone. Ένα σημαντικό χαρακτηριστικό της σύγχρονης κινητής συσκευής είναι ότι μπορεί να λάβει τη θέση της. Όχι μόνο για χρήση στη συσκευή αλλά επίσης και για απομακρυσμένες εφαρμογές που απαιτούν παρακολούθηση της συσκευής. Επιπλέον, η παρακολούθηση θα πρέπει να δώσει δυναμικά ενημερώσεις θέσης όταν έρχεται αντιμέτωπη με μεταβαλλόμενες συνθήκες, όπως καθυστερήσεις λόγω θέσης και επικοινωνίας, και αλλαγή ακρίβειας εντοπισμού θέσης. Το σύστημα παρακολουθεί τους πεζούς που είναι εξοπλισμένοι με συσκευές που έχουν GPS ενεργοποιημένο. Το συγκεκριμένο σύστημα, παρέχει στους αναβάτες εργαλεία σε Smartphone και διεπαφές. Επιδεικνύεται ένα εργαλείο σχεδιασμού διαδρομής πραγματικού χρόνου για κινητό και μια μέθοδος ανίχνευσης σφαλμάτων σε δεδομένα μέσω μαζικής μεταφοράς. Τέλος, παρουσιάζονται αξιολογήσεις που δείχνουν βελτίωση ικανοποίησης με τα μέσα μαζικής μεταφοράς, μείωση στους χρόνους αναμονής, αύξηση στη χρήση μέσω μεταφοράς, ενθάρρυνση για περπάτημα, και βελτίωση της αντίληψης ασφάλειας των αναβατών. Η αστική κινητικότητα είναι ένας βασικός παράγοντας για την ποιότητα ζωής ενός πολίτη, καθώς ένα όλο και μεγαλύτερο ποσοστό του πληθυσμού ζει σε αστικές περιοχές.

Συγγραφέας	Gps/Gsm/Gprs	Web/Mobile Application	Real time tracking	Route Storage	Notify	Στόλος Οχημάτων	Οδική Συμπεριφορά
Varandas, Vaidya & Rodrigues (2010)	Gps/Gsm	Mobile	NAI	OXI	SMS	OXI	OXI
Almomani et al (2011)	Gps/Gsm/Gprs	Web/Mobile	NAI	NAI	SMS	NAI	NAI
Behzada et al (2014)	Gps/Gsm	Web/Mobile	NAI	NAI	SMS	OXI	OXI
Abbott & Powell (2002)	Gps	OXI*	NAI	OXI	OXI	OXI	OXI
Amini, Vaghefi, Garza & Buehrer (2014)	Gps/Gsm	OXI*	NAI	OXI	OXI	OXI	OXI
Chew et al (2006)	Gps/Gsm	Web/Mobile	NAI	OXI	OXI	OXI	OXI
Pham, Driberg & Chi (2013)	Gps/Gsm	Mobile	NAI	OXI	OXI	OXI	OXI
Hind (2014)	Gps/Gsm	Web	NAI	OXI	OXI	OXI	OXI
Lee, Tewolde & Kwon (2014)	Gps/Gsm/Gprs	Mobile	NAI	NAI	OXI	OXI	OXI
Patil B, Patil R & Pittet, (2011)	Gps/Gsm	OXI*	OXI	OXI	OXI	OXI	OXI
Chandra, Jain & Qadeer, (2011)	Gps/Gsm/Gprs	Mobile	NAI	OXI	OXI	OXI	OXI
Kadibagil & Guruprasad, (2014)	Gps/Gsm/Gprs	Web/Mobile	NAI	NAI	SMS	OXI	OXI
Mane & Khairnar, (2014)	Gps	Mobile	NAI	OXI	OXI	OXI	OXI

Πίνακας 1. Συγκριτικός πίνακας των προσεγγίσεων που χρησιμοποιούν κινητά τηλέφωνα

* Αυτά τα άρθρα αναφέρονται σε μελέτες, ποσοτικούς και ποιοτικούς ελέγχους, μετρήσεις.

Όπως είναι εμφανές από τον συγκριτικό πίνακα των άρθρων που μελετήθηκαν οι περισσότερες εφαρμογές έχουν επικεντρωθεί στον εντοπισμό σε πραγματικό χρόνο χωρίς να αποθηκεύουν αυτά τα στοιχεία. Επιπροσθέτως δεν δίνουν βάση στην οδική ασφάλεια ούτε απευθύνονται σε στόλο οχημάτων. Η εφαρμογή που αναπτύχθηκε εστίασε στο να καλύψει τις ελλείψεις των άλλων εφαρμογών με τις πιο σύγχρονες τεχνολογίες.

Κεφάλαιο 3: Τεχνολογίες που χρησιμοποιήθηκαν

Η βάση δεδομένων

Η βάση που χρησιμοποιήθηκε είναι η **PostgreSQL**. Η PostgreSQL είναι ένα ισχυρό, ανοιχτού κώδικα σύστημα βάσης δεδομένων. Έχει περισσότερα από 15 χρόνια ενεργή ανάπτυξη και αποδεδειγμένη αρχιτεκτονική που έχει κερδίσει μια ισχυρή φήμη για την αξιοπιστία, την ακεραιότητα, και την ορθότητα των δεδομένων. Τρέχει σε όλα τα κύρια λειτουργικά συστήματα, συμπεριλαμβανομένων του Linux, UNIX και Windows. Είναι πλήρως συμβατή, έχει την πλήρη υποστήριξη για ξένα κλειδιά, συνδέσεις, όψεις, triggers, και αποθηκευμένες διαδικασίες (σε πολλές γλώσσες). Περιλαμβάνει τους περισσότερους SQL: 2008 τύπους δεδομένων. Επίσης, υποστηρίζει την αποθήκευση των μεγάλων δυαδικών αντικειμένων, συμπεριλαμβανομένων εικόνων, ήχων ή βίντεο. Διαθέτει εγγενείς διεπαφές προγραμματισμού για C / C ++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, και εξαιρετική τεκμηρίωση. (<http://www.postgresql.org/>)

Η διαχείριση της βάσης έγινε με το εργαλείο **pgAdmin**. Το pgAdmin είναι το πιο δημοφιλές και διαθέτει πλατφόρμα διαχείρισης και ανάπτυξης Ανοιχτού Κώδικα για την PostgreSQL, την πιο προηγμένη βάση δεδομένων ανοιχτού κώδικα στον κόσμο. Η εφαρμογή μπορεί να χρησιμοποιηθεί σε οποιαδήποτε πλατφόρμα. Το pgAdmin έχει σχεδιαστεί για να ανταποκριθεί στις ανάγκες όλων των χρηστών, από απλά ερωτήματα SQL μέχρι την ανάπτυξη πολύπλοκων βάσεων δεδομένων. Η γραφική διεπαφή υποστηρίζει όλα τα χαρακτηριστικά της PostgreSQL και κάνει τη διαχείριση εύκολη. Δεν απαιτούνται πρόσθετα προγράμματα οδήγησης για την επικοινωνία με το διακομιστή της βάσης δεδομένων. Το pgAdmin έχει αναπτυχθεί από μια κοινότητα εμπειρογνομόνων PostgreSQL σε όλο τον κόσμο και είναι διαθέσιμο σε περισσότερες από δώδεκα γλώσσες. Είναι Ελεύθερο Λογισμικό, που διατίθεται υπό την Άδεια της PostgreSQL. (<http://www.pgadmin.org/>)

CodeIgniter php framework

Το **CodeIgniter** χρησιμοποιήθηκε για την υλοποίηση του back end. Δηλαδή για τη δημιουργία των api (application programming interface) με τα οποία θα επικοινωνεί η εφαρμογή με τη βάση. Εκτός από το CodeIgniter έγινε χρήση του CodeIgniter Rest Server extension για να επιτευχθεί η επικοινωνία της εφαρμογής με τη βάση μέσω των api.

Το CodeIgniter είναι ένα ισχυρό PHP framework με ένα πολύ μικρό ίχνος, που χτίστηκε για τους προγραμματιστές που χρειάζονται ένα απλό και κομψό πακέτο εργαλείων για να δημιουργήσουν εφαρμογές web με πλήρεις δυνατότητες. (<https://www.codeigniter.com/>)

Το **CodeIgniter Rest Server** είναι μία πλήρως RESTful server υλοποίηση για CodeIgniter χρησιμοποιώντας μια βιβλιοθήκη, ένα αρχείο config και έναν controller (ελεγκτή). (<https://github.com/chriskacerguis/codeigniter-restserver>)

Οι δοκιμές των api έγιναν με τη βοήθεια του εργαλείου **postman** του google chrome. Η εφαρμογή postman σας βοηθά να είστε εξαιρετικά αποτελεσματικοί, ενώ εργάζεστε με API. Το εργαλείο postman επιτρέπει να δημιουργήσει κανείς αιτήματα γρήγορα, να τα αποθηκεύσει για μελλοντική χρήση και να αναλύσει τις απαντήσεις που αποστέλλονται από

το API. Μπορεί να μειώσει δραματικά το χρόνο που απαιτείται για τη δοκιμή και την ανάπτυξη APIs. Προσαρμόζεται εξίσου καλά σε μεμονωμένους προγραμματιστές, μικρές ομάδες ή μεγάλες οργανώσεις. (<https://www.getpostman.com>)

Json (JavaScript Object Notation)

Όλα τα δεδομένα που διαβάζουν τα api από την βάση στέλνονται στην εφαρμογή σε json μορφή. Το JSON είναι μια ελαφριά μορφή ανταλλαγής δεδομένων. Είναι εύκολο να διαβαστούν, να γραφτούν, να αναλυθούν και να δημιουργηθούν. Βασίζεται σε ένα υποσύνολο της γλώσσας προγραμματισμού JavaScript. Το JSON είναι μια μορφή κειμένου που είναι εντελώς ανεξάρτητη από τη γλώσσα, αλλά χρησιμοποιεί συμβάσεις που είναι εξοικειωμένες με τους προγραμματιστές της οικογένειας C γλωσσών, συμπεριλαμβανομένων των C, C ++, C #, Java, JavaScript, Perl, Python, και πολλές άλλες. Αυτές οι ιδιότητες καθιστούν το JSON μια ιδανική γλώσσα για ανταλλαγή δεδομένων. (<http://www.json.org/>)

XMLHttpRequest

Οι κλήσεις των api και η αποστολή των παραμέτρων από την εφαρμογή κινητού έγιναν με ***XMLHttpRequest***. Το XMLHttpRequest είναι ένα API που παρέχει λειτουργικότητα client για τη μεταφορά δεδομένων μεταξύ ενός client και ενός server. Παρέχει έναν εύκολο τρόπο ανάκτησης δεδομένων από μια διεύθυνση URL, χωρίς να χρειάζεται να κάνει πλήρη ανανέωση της σελίδας. Αυτό επιτρέπει σε μια ιστοσελίδα να ενημερώσει μόνο ένα μέρος της χωρίς να διαταράσσεται ο χρήστης. Το XMLHttpRequest χρησιμοποιείται σε μεγάλο βαθμό σε προγραμματισμό AJAX. (<https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest>)

NetBeans

Τόσο το interface όσο και τα web services υλοποιήθηκαν στο εργαλείο ***NetBeans***. Γρήγορη και εύκολη ανάπτυξη desktop, mobile και web εφαρμογές με Java, JavaScript, HTML5, PHP, C / C ++ και άλλα. Το NetBeans IDE είναι δωρεάν, ανοιχτού κώδικα, και έχει μια παγκόσμια κοινότητα χρηστών και προγραμματιστών. Το NetBeans είναι το εργαλείο στο οποίο αναπτύχθηκε κυρίως η εφαρμογή. (<https://netbeans.org/>)

Adobe PhoneGap

Για την υλοποίηση της εφαρμογής επιλέχθηκε η τεχνολογία ***PhoneGap framework*** διότι δίνει την δυνατότητα ανάπτυξης της εφαρμογής για όλες τις πλατφόρμες κινητών συσκευών. Κατασκευή καταπληκτικών εφαρμογών για κινητά με τεχνολογία open web. Επαναχρησιμοποίηση των υφιστάμενων δεξιοτήτων ανάπτυξης ιστοσελίδων για να γίνουν γρήγορα υβριδικές εφαρμογές που έχουν δημιουργηθεί με HTML, CSS και JavaScript. Δημιουργία εφαρμογών για πολλαπλές πλατφόρμες με μια ενιαία βάση κώδικα, ώστε να μην έχει σημασία η συσκευή (android, windows, ios). Από την ομάδα Apache Cordova, το Adobe PhoneGap framework είναι μία ανοιχτού κώδικα διανομή της Cordova - παρέχοντας το πλεονέκτημα της τεχνολογίας που δημιουργήθηκε από μια ποικιλόμορφη ομάδα επαγγελματιών, μαζί με μια ισχυρή κοινότητα των προγραμματιστών - καθώς και πρόσβαση στο σύνολο εργαλείων του PhoneGap. Χρειάστηκαν τόσο το εργαλείο command-line όσο και η desktop εφαρμογή. Η εφαρμογή ***PhoneGap Desktop*** παρέχει μια διεπαφή drag and drop για τη δημιουργία εφαρμογών PhoneGap. Το ***PhoneGap CLI*** παρέχει μια διεπαφή γραμμής

εντολών για τη δημιουργία εφαρμογών PhoneGap. Το PhoneGap CLI έχει κάποια επιπλέον χαρακτηριστικά από την PhoneGap Desktop για την κατασκευή και τη λειτουργία των εφαρμογών σας PhoneGap σε πολλαπλές πλατφόρμες. (<http://phonegap.com/>)

Plugin: Χρειάστηκε να εγκατασταθούν κάποια plugin. Plugin ορίζεται ένα σύστημα συστατικών κάποιου λογισμικού που προσθέτει ιδιαίτερες δυνατότητες σε ένα μεγαλύτερο λογισμικό. Παρακάτω φαίνεται ποια χρησιμοποιήθηκαν. (<https://el.wikipedia.org/wiki/Plug-in>)

cordova-plugin-geolocation: Χρησιμοποιήθηκε κυρίως για να ληφθούν πληροφορίες του gps (θέση, ταχύτητα) είναι το geolocation. Αυτό το plugin παρέχει πληροφορίες σχετικά με τη θέση της συσκευής, όπως γεωγραφικό πλάτος και μήκος. (<https://cordova.apache.org>)

cordovaHTTP: Χρησιμοποιείται για επικοινωνία με http servers. (<https://cordova.apache.org>)

cordova-plugin-device: Αυτό το plugin ορίζει ένα global αντικείμενο συσκευής, το οποίο περιγράφει το υλικό και το λογισμικό της συσκευής. Παρά το γεγονός ότι το αντικείμενο είναι global, δεν είναι διαθέσιμο μέχρι το deviceready event. (<https://cordova.apache.org>)

cordova-plugin-dialog: Παρέχει πρόσβαση σε ορισμένα στοιχεία διαλόγου(alert, beep, vibrate) μέσω ενός global αντικειμένου navigator.notification. (<https://cordova.apache.org>)

cordova-plugin-whitelist: Υλοποιεί μια πολιτική ασφαλείας για πλοήγηση σε webview.

cordova-plugin-file: Αυτό το plugin εφαρμόζει ένα API αρχείου που επιτρέπει την πρόσβαση ανάγνωσης / εγγραφής σε αρχεία της συσκευής. (<https://cordova.apache.org>)

Launch Navigator Cordova/Phonegap Plugin:

Αυτό το plugin Cordova / PhoneGap μπορεί να χρησιμοποιηθεί για πλοήγηση σε έναν προορισμό μέσα από εφαρμογές πλοήγησης του Android. (<https://www.openstreetmap.org>)

OpenStreetMap

Η απεικόνιση της διαδρομής στο χάρτη έγινε σε OpenStreetMap. Το OpenStreetMap είναι χάρτης του κόσμου, που είναι δωρεάν, υπό άδεια ελεύθερης χρήσης. Το OpenStreetMap προσφέρει δεδομένα χάρτη σε χιλιάδες ιστότοπους, εφαρμογές κινητού και συσκευές υλικού. Το OpenStreetMap αναπτύσσεται από μια κοινότητα χαρτογράφων που συνεισφέρουν και διατηρούν δεδομένα σχετικά με δρόμους, μονοπάτια, καφετέριες, σιδηροδρομικούς σταθμούς, και πολλά περισσότερα, σε όλον τον κόσμο. Το OpenStreetMap είναι βάση ανοικτών δεδομένων. (<https://www.openstreetmap.org>)

Javascript-js

Το JavaScript είναι γλώσσα προγραμματισμού HTML και Web. Τα πιο βασικά αρχεία που χρησιμοποιήθηκαν για τις ανάγκες τις εφαρμογής είναι:

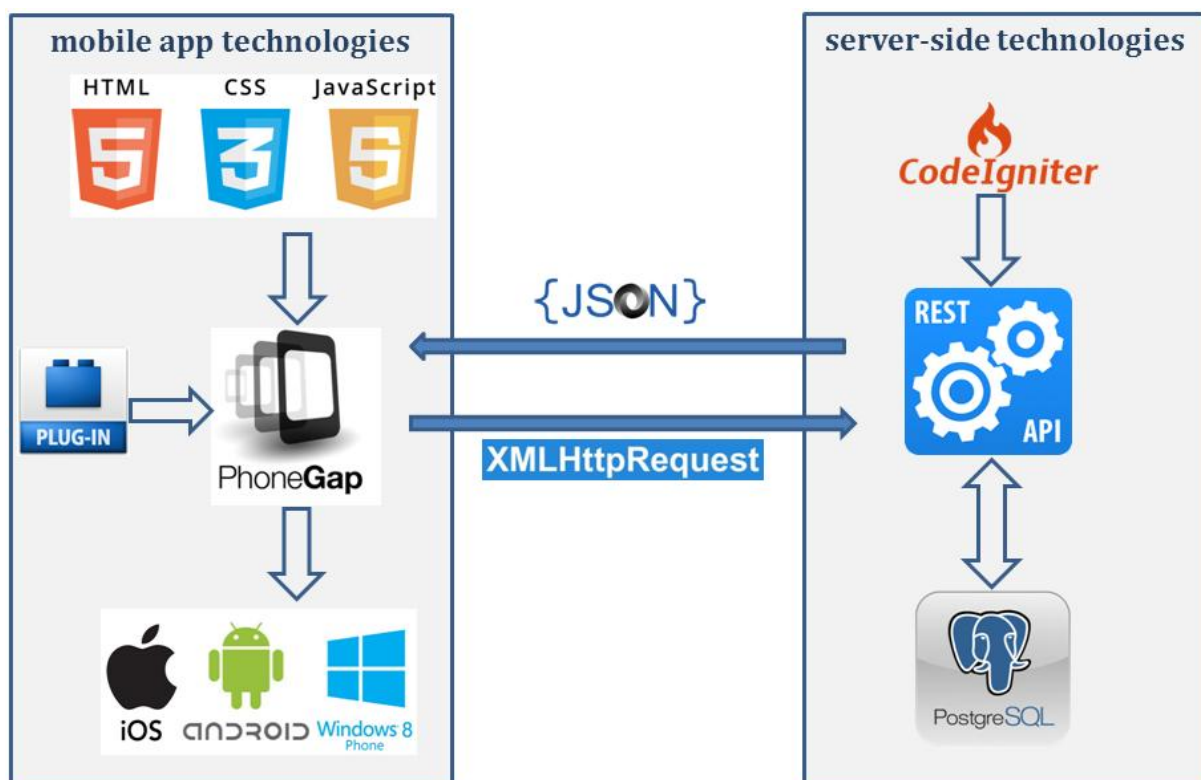
Cordova.js: Ένα ενοποιημένο επίπεδο Javascript για projects Apache Cordova. Το αρχείο δημιουργείτε από το εργαλείο κατά την δημιουργία νέου project. (<https://cordova.apache.org>)

OpenLayers.js: Μία υψηλής απόδοσης, βιβλιοθήκη για όλες τις ανάγκες χαρτογράφησης. (<http://openlayers.org/>)

Moment.js: Αναλύει, επικυρώνει, χειρίζεται και εμφανίζει ημερομηνίες σε JavaScript. (<http://momentjs.com/>)

JavaScript Timing Events: Ο Browser παρέχει έναν ενσωματωμένο χρονοπρογραμματιστή ο οποίος επιτρέπει να ρυθμιστούν κλήσεις function συνεχόμενα κάθε συγκεκριμένη χρονική περίοδο μέχρι να ζητηθεί διακοπή. (<http://javascript.info/tutorial/settimeout-setinterval>)

Στο παρακάτω σχήμα βλέπουμε πως επικοινωνούν οι τεχνολογίες μεταξύ τους. Γίνεται ένας διαχωρισμός στις τεχνολογίες που συμμετέχουν στην εφαρμογή κινητού και σε αυτές που συμμετέχουν από πλευράς server. Τα html, css, javascript μαζί με plugin χρησιμοποιούνται στο phonegap framework για την παραγωγή εφαρμογής κινητού πλατφόρμας ios, android και windows. Το codeIgniter framework χρησιμοποιείται για την δημιουργία των api για επικοινωνία με τη βάση. Η ανταλλαγή δεδομένων γίνεται με XmlHttpRequest και JSON response μεταξύ της εφαρμογής και των api.



Εικόνα 1. Διάγραμμα των τεχνολογιών που χρησιμοποιήθηκαν για την υλοποίηση

Κεφάλαιο 4: Περιγραφή υλοποίησης

Στο παρακάτω διάγραμμα φαίνεται πως λειτουργεί ο εντοπισμός του οχήματος και η πως φτάνει η πληροφορία στον χρήστη. Το κινητό που βρίσκεται μέσα στο όχημα παίρνει το σήμα του gps από το δορυφόρο. Η εφαρμογή στέλνει τις συντεταγμένες και άλλες πληροφορίες μέσω internet στον server όπου αποθηκεύονται στη βάση στη βάση δεδομένων. Στη συνέχεια η εφαρμογή που είναι διαθέσιμη από οποιαδήποτε συσκευή έχει πρόσβαση στο internet (tablet, laptop, smartphone) παρουσιάζει τα data analytics.



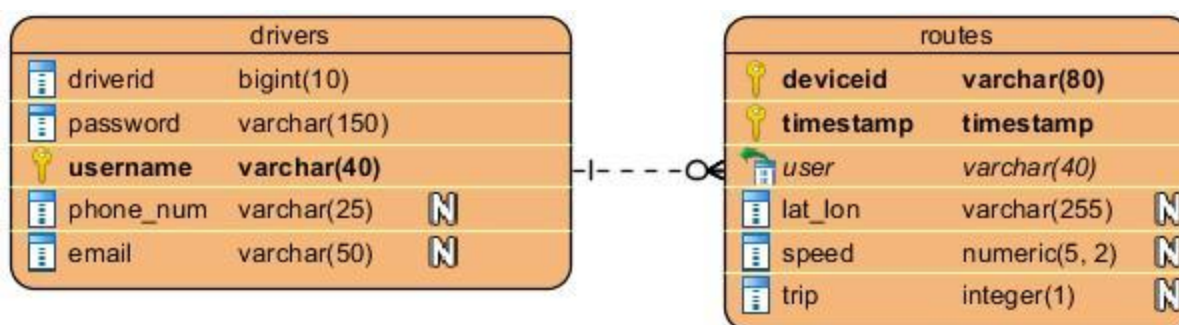
Εικόνα 2. Γενική αποτύπωση της διαδικασίας καταγραφής θέσης

4.1 Βάση δεδομένων

Η βάση δεδομένων αποτελείται από 2 πίνακες (drivers, routes). Ο πίνακας drivers περιλαμβάνει τα στοιχεία των χρηστών της εφαρμογής και ο πίνακας routes τα στοιχεία καταγραφής της κίνησης των οχημάτων. Επίσης υπάρχει η όψη dateroutes που περιέχει πιο αναλυτικά τα στοιχεία που καταγράφονται κατά την κίνηση.

Πίνακας χρηστών drivers (driverid, username, password, phone_num, email). Το driverid παίρνει αυτόματα τιμή από sequence κατά την εισαγωγή του χρήστη. Το username είναι το πρωτεύων κλειδί του πίνακα. Τα username, password είναι υποχρεωτικά πεδία. Τα phone_num, email είναι προαιρετικά πεδία.

Πίνακας διαδρομών routes (deviceid, user, lat_lon, timestamp, speed, trip). Το deviceid είναι το μοναδικό αναγνωριστικό κάθε κινητής συσκευής (Device unique ID). Ο user είναι ξένο κλειδί που αναφέρεται στο username του πίνακα drivers. Το lat_lon είναι τύπου geometry όπου αποθηκεύεται το σημείο που προκύπτει από τις συντεταγμένες latitude και longitude. Το timestamp παίρνει αυτόματα την τιμή της τρέχουσας ημερομηνίας και ώρας κατά την εισαγωγή. Το speed είναι η ταχύτητα του οχήματος και το trip είναι ένας μετρητής που μετράει πόσες διαδρομές κάνει κάθε χρήστης στην ίδια μέρα. Πρωτεύων κλειδί του πίνακα είναι ο συνδυασμός deviceid, timestamp.



Εικόνα 3. Οι δύο βασικοί πίνακες

Η όψη dateroutes (deviceid, user, lat, lon, date, time, speed, trip). Η όψη προέρχεται από τον πίνακα routes. Με απλές εντολές βρίσκει τις συντεταγμένες του σημείου και διασπά το timestamp σε ημερομηνία και ώρα.

dateroutes	
deviceid	varchar(80)
date	date
time	time(8)
lat	numeric(20, 10)
lot	numeric(20, 10)
speed	numeric(5, 2)
user	varchar(40)
trip	integer(1)

Εικόνα 4. Ο πίνακας που καταγράφει τα δεδομένα θέσης

4.2. Back end – api (application programming interface)

Στο CodeIgniter php framework δημιουργήθηκε ο controller User και το model User_model. Ο User κάνει χρήση του Rest Server extension. Τα ***api*** βρίσκονται στο αρχείο ***user.php*** :

Driverlogin:

Το api αυτό καλείται κατά το register του νέου χρήστη. Παίρνει σαν παραμέτρους τα στοιχεία του χρήστη, δηλαδή username, password, email, phonenumber. Κατ αρχήν ελέγχει αν έχουν συμπληρωθεί όλα τα υποχρεωτικά πεδία. Σε περίπτωση που δεν είναι συμπληρωμένα επιστρέφει μήνυμα λάθους «You have not provided complete data» και status false και δεν προχωράει σε εγγραφή το χρήστη. Διαφορετικά ελέγχει αν υπάρχει ο χρήστης καλώντας την function check_driver_id του User_model και αν υπάρχει επιστρέφει μήνυμα «User existed», το username και status true, οπότε δεν πραγματοποιείται εγγραφή διότι ο χρήστης υπάρχει ήδη. Στη συνέχεια αν ο χρήστης δεν υπάρχει καλεί την function add_driver του User_model και προχωράει σε εγγραφή εμφανίζοντας μήνυμα «User added» και status true.

logind:

Για το login του χρήστη καλείται το logind api με παραμέτρους username, password του χρήστη. Ελέγχει αν έχουν συμπληρωθεί και τα δύο πεδία και αν όχι επιστρέφει μήνυμα λάθους «You have not provided complete data» και status false. Αν έχουν συμπληρωθεί και τα δύο ελέγχει αν τα στοιχεία αντιστοιχούν σε χρήστη με κλήση της function login_driver του User_model και επιστρέφει αν ναι επιστρέφει μήνυμα «User existed» και status true. Αν δεν βρει χρήστη με αυτά τα στοιχεία επιστρέφει μήνυμα «User not registered» και false.

getnextTrip:

Το api αυτό καλείται με παράμετρο το χρήστη και επιστρέφει τον αριθμό της επόμενης ημερήσιας διαδρομής του χρήστη σε json format χρησιμοποιώντας την function getnext_trip του user_model.

Senddlocation:

Στο api αυτό στέλνονται οι παράμετροι deviceid, user, lat, lon, speed, trip. Το api ελέγχει αν έχουν έρθει όλα τα δεδομένα και τότε τα στέλνει στην function store_d_loc του user_model για να αποθηκευτούν στη βάση. Αν δεν έχουν έρθει όλα εμφανίζεται μήνυμα λάθους.

retrieveDrivers:

Επιστρέφει τους χρήστες τις εφαρμογής σε json format με κλήση της function retrieve_drivers του user_model. Σε περίπτωση σφάλματος εμφανίζεται λάθος.

retrieveDriverDates:

Με παράμετρο τον χρήστη επιστρέφει τις ημερομηνίες που έχει διαδρομές. Καλεί την retrieve_driver_dates. Σε περίπτωση εξαίρεσης εμφανίζει μήνυμα.

retrieveDriverTracks:

Με παραμέτρους το χρήστη, την ημέρα και τη διαδρομή της ημέρας (πχ. 1^η) επιστρέφει σε json format όλα τα σημεία της διαδρομής με τη βοήθεια της function retrieve_driver_tracks. Αν δεν συμπληρωθούν όλες οι παράμετροι επιστρέφει false και μήνυμα.

Back end – functions called by api

Οι functions που καλούν τα api βρίσκονται στο **user_model.php** και είναι οι παρακάτω:

check driver id:

Στέλνει ένα ερώτημα στη βάση στον πίνακα drivers με παράμετρο τον χρήστη για να δει αν υπάρχει ο συγκεκριμένος χρήστης. Αν υπάρχει επιστρέφει το id του χρήστη αλλιώς false.

add driver:

Κάνει insert μια εγγραφή στον πίνακα drivers τα data που δέχεται ως παράμετρο.

login driver:

Στέλνει ερώτημα στη βάση για να δει αν ο πίνακας drivers έχει χρήστη με συγκεκριμένο username και password. Αν το ερώτημα φέρει ένα αποτέλεσμα επιστρέφει το username σε αντίθετη περίπτωση επιστρέφει false.

getnext trip:

Βρίσκει ποια είναι η τελευταία διαδρομή (trip) του χρήστη για την σημερινή ημερομηνία και προσθέτει ένα για να δώσει τον αριθμό της επόμενης διαδρομής. Στέλνει ένα ερώτημα στη βάση με παράμετρο το χρήστη και την ημερομηνία συστήματος. Αν δεν υπάρχει καμία διαδρομή επιστρέφει 1, δηλαδή αυτή θα είναι η πρώτη διαδρομή. Αν ο χρήστης έχει κάνει διαδρομές στη συγκεκριμένη ημερομηνία βρίσκει τη μέγιστη (τελευταία) και προσθέτει 1.

Store d loc:

Κάνει insert στον πίνακα routes. Μετατρέπει τις συντεταγμένες σε σημείο και επίσης στέλνει το id συσκευής, χρήστη, ταχύτητα και αριθμό ημερήσιας διαδρομής. Ουσιαστικά αποθηκεύει ένα συγκεκριμένο στίγμα του gps με λεπτομέρειες τοποθεσίας, ταχύτητας, συσκευής και χρήστη. Η ημερομηνία και ώρα αποθηκεύεται αυτόματα από default τιμή στη βάση.

retrieve drivers:

Επιλέγει από τον πίνακα drivers στοιχεία των χρηστών (email,username,phone_num).

retrieve driver dates:

Επιλέγει για κάποιο χρήστη της ημερομηνίες και διαδρομές του από την όψη dateroutes με αύξουσα ημερομηνία και διαδρομή. Εκτελεί ένα διακριτό ερώτημα στη βάση με παράμετρο το χρήστη και επιστρέφει ημερομηνία, διαδρομή ταξινομημένα (ascending).

retrieve driver tracks:

Επιστρέφει όλα τα σημεία μιας διαδρομής, την ταχύτητα και την ώρα με αύξουσα χρονική στιγμή. Εκτελεί ένα query στην όψη dateroutes με παραμέτρους το χρήστη, την ημερομηνία και τη διαδρομή και επιστρέφει για κάθε σημείο τις συντεταγμένες, την ταχύτητα και τη χρονική στιγμή με αύξουσα σειρά (χρόνου).

4.3 Front end – user interface

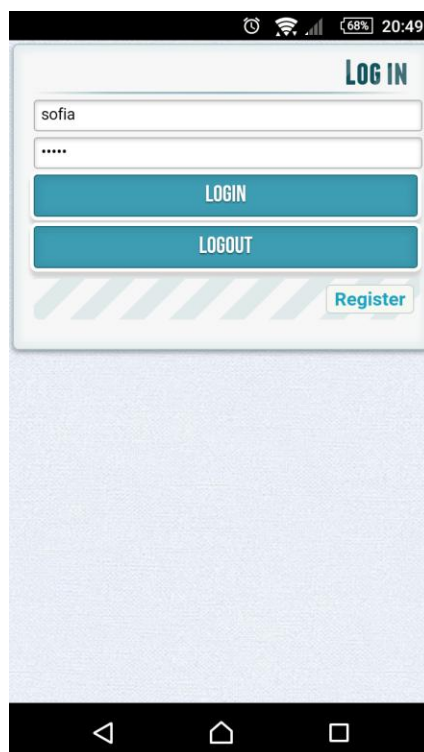
Για την υλοποίηση της εφαρμογής επιλέχθηκε η τεχνολογία PhoneGap framework που δίνει την δυνατότητα ανάπτυξης της εφαρμογής για όλες τις πλατφόρμες κινητών συσκευών. Ωστόσο η εφαρμογή επικεντρώθηκε και δοκιμάστηκε σε πλατφόρμα android. Οι ιστοσελίδες έχουν δημιουργηθεί με κώδικα HTML, CSS και JavaScript. Το project περιλαμβάνει 4 html ιστοσελίδες (index.html, login.html, main.html reports.html).

Οι τρεις βασικές λειτουργίες της εφαρμογής είναι το a) login – register του χρήστη, b) η καταγραφή της διαδρομής και c) τα data analytics. Παρακάτω παρουσιάζονται αναλυτικά:

a) use case login - register

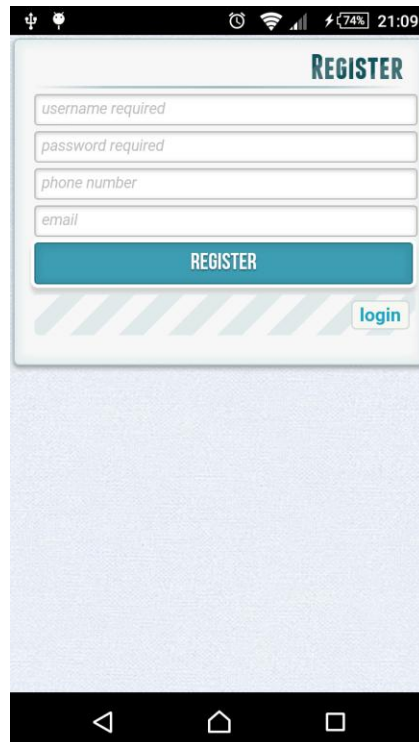
login.html

Η εφαρμογή ξεκινάει με την index.html η οποία κατευθύνει στην login.html. Από αυτή την ιστοσελίδα κάνουν login και register οι χρήστες της εφαρμογής. Χρησιμοποιεί κάποια css και js αρχεία που βρίσκονται στους αντίστοιχους φάκελους. Σε αυτή την ιστοσελίδα βρίσκεται μία animated form που περιλαμβάνει τη φόρμα του **login** και τη φόρμα του **register**. Κατά την έναρξη της εφαρμογής εμφανίζεται η **login** με link στην **register**. Στο onload της login καλείται η function **get_local()**. Η get_local διαβάζει από το localStorage της συσκευής το username και password που είναι αποθηκευμένα και συμπληρώνει με αυτά τα αντίστοιχα πεδία για να μην τα ξαναγράφει ο χρήστης. Παρακάτω φαίνεται η login form:



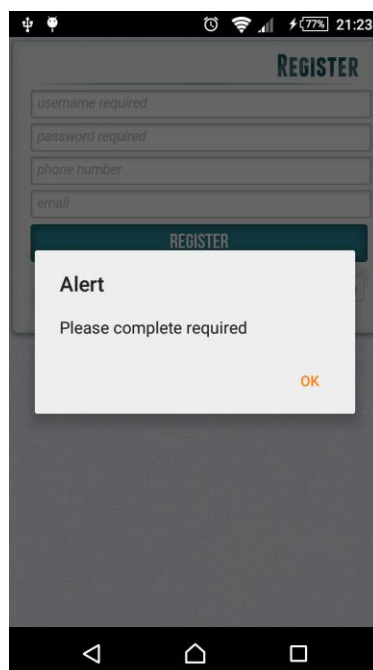
Εικόνα 5. Η εισαγωγική οθόνη (login) της εφαρμογής

Όπως φαίνεται η `get_local` διάβασε τα στοιχεία του χρήστη `sofia` από το `localStorage` της συσκευής και τα εμφάνισε στα αντίστοιχα πεδία. Αν πατήσουμε το `link register` εμφανίζεται η φόρμα εγγραφής του χρήστη. Φαίνεται ποια πεδία πρέπει να συμπληρωθούν υποχρεωτικά (`username` και `password`) και ποια προαιρετικά (`phone` και `email`). Η `register form` παρακάτω:



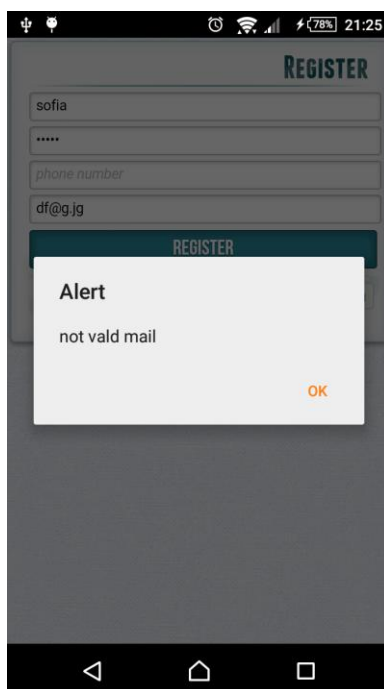
Εικόνα 6. Η οθόνη εγγραφής

Στο event `onclick` του κουμπιού `register` εκτελείται η function `registerDevice()`. Η `registerDevice` ελέγχει πρώτα αν ο χρήστης έχει συμπληρώσει τα υποχρεωτικά πεδία. Αν δεν έχουν συμπληρωθεί τα δύο υποχρεωτικά πεδία εμφανίζει μήνυμα λάθους «Please complete required» όπως φαίνεται στο επόμενο screenshot:



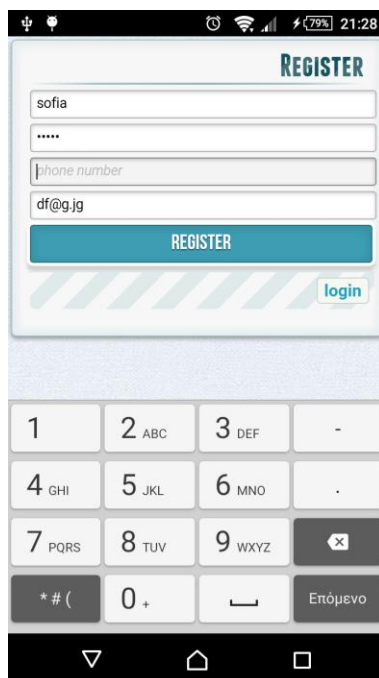
Εικόνα 7. Παράδειγμα ελλιπών στοιχείων

Αν ο χρήστης έχει συμπληρώσει το email ελέγχεται με την function *validateEmail()*. Η function παίρνει παράμετρο ένα string και επιστρέφει true αν είναι έγκυρο. Αν το email δεν έχει το σωστό format (ηχ λείπει το @) επιστρέφει false και το μήνυμα «not valid mail».



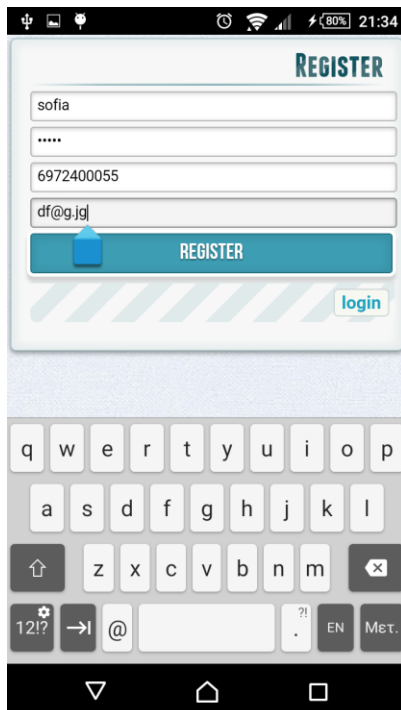
Εικόνα 8. Παράδειγμα εσφαλμένων στοιχείων

Όταν ο χρήστης θελήσει να συμπληρώσει το τηλέφωνο του εμφανίζεται το πληκτρολόγιο μόνο με αριθμούς για να μην πληκτρολογήσει κατά λάθος γράμματα.



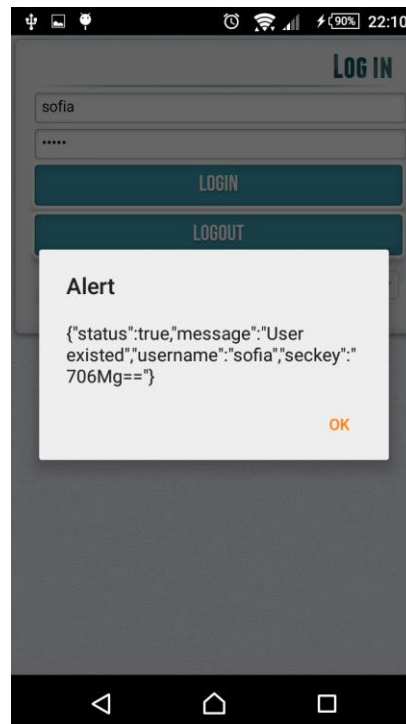
Εικόνα 9. Καταχώρηση στοιχείων

Αν επιλέξει να συμπληρώσει και το email εμφανίζεται το πληκτρολόγιο με το σύμβολο @ για να τον προφυλάξει από λάθος.



Εικόνα 10. Υποστήριξη κατά την καταχώρηση

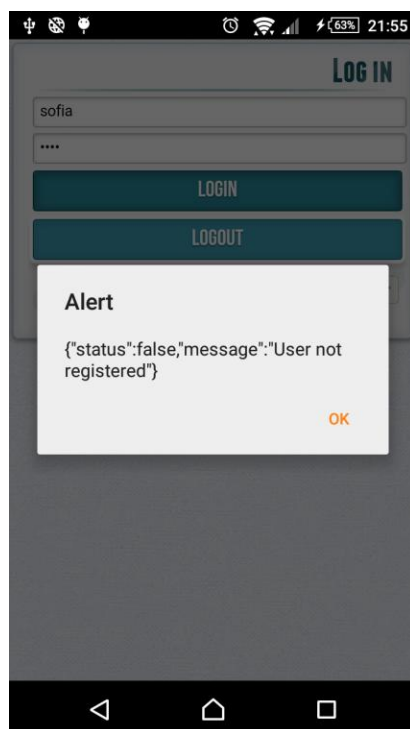
Αν όλα τα υποχρεωτικά πεδία συμπληρωθούν σωστά η `registerDevice` καλεί το api ***driverlogin*** με XMLHttpRequest στέλνοντας τα στοιχεία του χρήστη. Μόλις ολοκληρωθεί η αποστολή αν ήταν επιτυχής εμφανίζεται μήνυμα με το response του api και πηγαίνει στην φόρμα του login αποθηκεύοντας στο `localStorage` τα στοιχεία του χρήστη. Στην εικόνα βλέπουμε ότι ο χρήστης «sofia» που προσπάθησε να κάνει register υπάρχει ήδη. Η `registerDevice` κάλεσε το api `driverlogin` αυτό έλεγξε αν υπάρχει χρήστης με αυτό το όνομα και επέστρεψε μήνυμα «User Existed» και status true. Στην συνέχεια προχώρησε στην οθόνη login και συμπλήρωσε τα πεδία για να είναι έτοιμος ο χρήστης για login.



Εικόνα 11. Επιτυχής σύνδεση

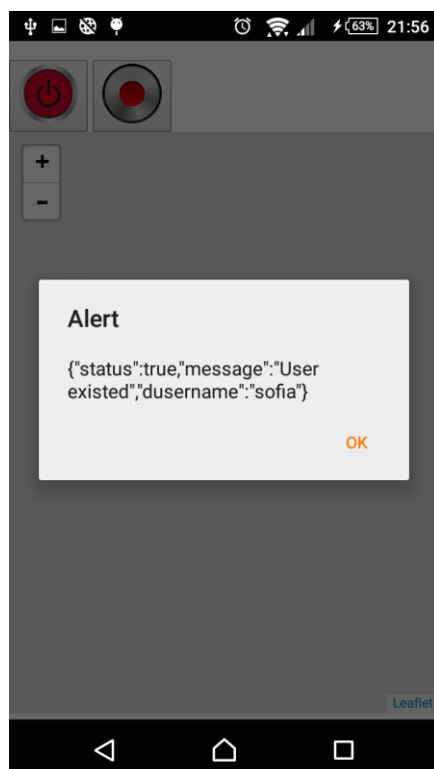
Το κουμπι logout καλεί την function ***clear_local()*** η οποία σβήνει από το `localStorage` τα στοιχεία του χρήστη και τερματίζει την εφαρμογή.

Με παρόμοιο τρόπο με τη register λειτουργεί και η φόρμα του login. Με το πάτημα του κουμπιού καλείται η function ***loginDevice*** η οποία αρχικά ελέγχει αν έχουν συμπληρωθεί όλα τα πεδία (username,password). Αν δεν έχει συμπληρωθεί κάποιο από αυτά εμφανίζεται το μήνυμα "Please complete". Στη συνέχεια καλεί το api `logind` με XMLHttpRequest στέλνοντας τα στοιχεία του χρήστη. Μόλις ολοκληρωθεί ο έλεγχος του χρήστη εμφανίζεται το response και το status του api. Αν πληκτρολογηθούν λάθος στοιχεία επιστρέφει false και μήνυμα «user not registered» και δεν πραγματοποιείται είσοδος στην εφαρμογή. Στην παρακάτω οθόνη βλέπουμε ένα αποτυχημένο login:



Εικόνα 12. Αποτυχία σύνδεσης (δεν υπάρχει ο χρήστης)

Αν ο χρήστης δώσει σωστό όνομα και κωδικό το `api` επιστρέφει `true` και εμφανίζει μήνυμα με το όνομα του χρήστη. Η εφαρμογή πηγαίνει στην οθόνη καταγραφής διαδρομής `main.html` αποθηκεύοντας στο `localStorage` τα στοιχεία του χρήστη με τη function **setlocal**.



Εικόνα 13. Επιτυχής σύνδεση και αποθήκευση στοιχείων

b) use case καταγραφή διαδρομής

main.html

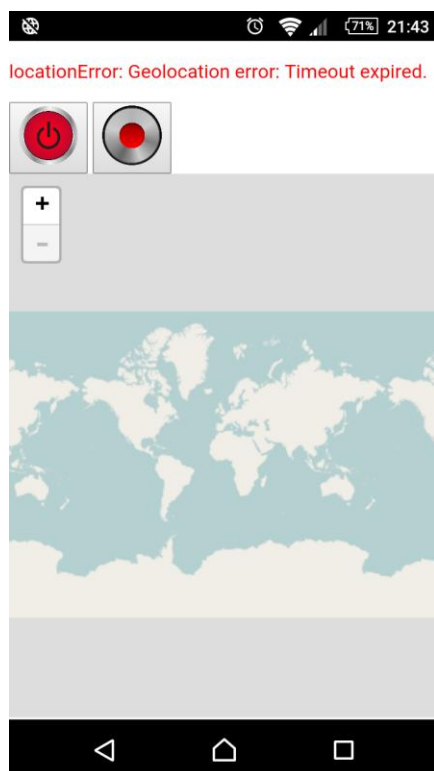
Σε αυτή την ιστοσελίδα εμφανίζεται η θέση και η διαδρομή του χρήστη και πραγματοποιείται η καταγραφή της κατ' επιλογή του. Χρησιμοποιεί κάποια css και js αρχεία που βρίσκονται στους αντίστοιχους φάκελους. Στο onload της login καλείται η function **onLoad()**. Αυτή διαβάζει το όνομα του χρήστη από το localStorage και το δίνει στην global μεταβλητή user. Στη συνέχεια καλεί τις function **setDeviceInfo()** και **init()**, ελέγχει την πλατφόρμα της συσκευής και βάζει ένα event listener στο device ready της συσκευής και τέλος καλεί την function **ondeviceready()**.

Η function **setDeviceInfo** ελέγχει αν η συσκευή έχει σήμα gps και εκτελεί τη μέθοδο **watchposition** του geolocation plugin. Η μέθοδος καλεί την function positionFound κάθε φορά που αλλάζει η θέση της συσκευής και την function onError σε περίπτωση σφάλματος. Στις επιλογές της μεθόδου ενεργοποιήθηκε η υψηλή ακρίβεια gps. Ο χρόνος αναμονής της συσκευής να επιστρέψει θέση ορίστηκε σε 60 δευτερόλεπτα και ο χρόνος προσωρινής αποθήκευσης μνήμης ορίστηκε σε 30 δευτερόλεπτα.

Η function **positionfound()** βρίσκει τις γεωγραφικές συντεταγμένες και την ταχύτητα της συσκευής. Η μέθοδος watchposition σε επιτυχή αλλαγή θέσης επιστρέφει ένα αντικείμενο τύπου position. Το αντικείμενο αυτό έχει τις ιδιότητες latitude, longitude, speed. Η function παίρνει σαν παράμετρο ένα position και επιστρέφει τις τρεις ιδιότητες.

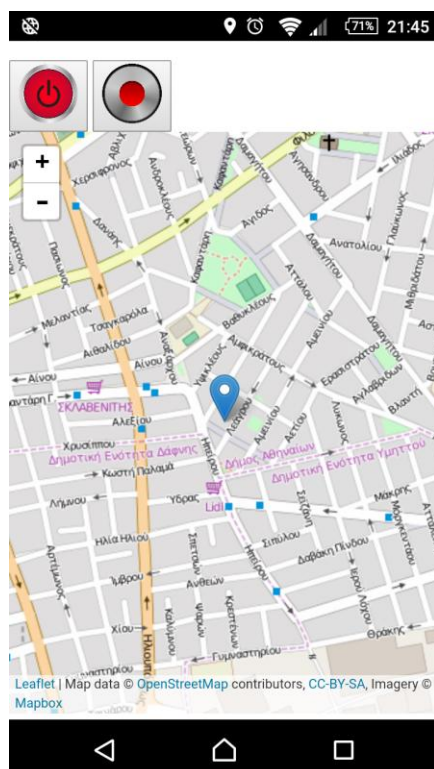
Η function **onDeviceReady()** καλεί επίσης την **setDeviceInfo()** και βάζει ένα listener στο event resume ο οποίος και αυτός καλεί την setDeviceInfo στο resume. Αυτό γίνεται για να εξασφαλίσει ότι η εφαρμογή θα συνεχίσει να ψάχνει σήμα gps σε περίπτωση σφάλματος.

Η function **init()** αρχικοποιεί το χάρτη χρησιμοποιώντας ένα api του OpenStreetMap και ορίζει ένα μέγιστο αριθμό μεγέθυνσης. Σε περίπτωση που δεν υπάρχει σήμα gps εμφανίζεται μήνυμα στην κορυφή της οθόνης και ο παγκόσμιος χάρτης όπως φαίνεται στην εικόνα:



Εικόνα 14. Σφάλμα σήματος GPS

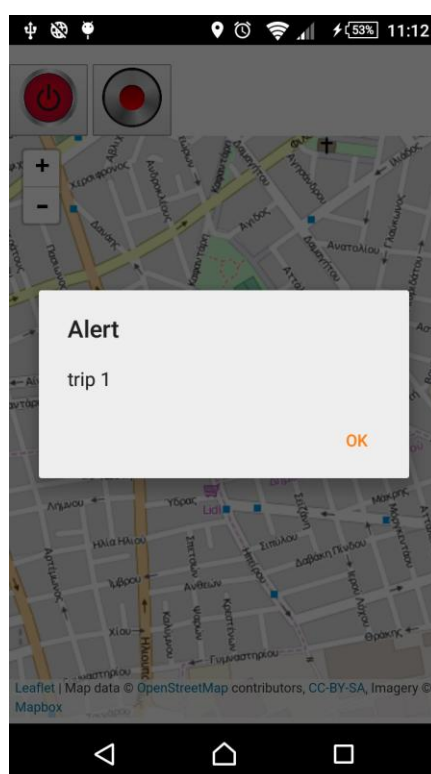
Αν το gps είναι ενεργό και έχει στίγμα εμφανίζει το χάρτη της περιοχής με κέντρο το σημείο που βρίσκεται η συσκευή και τοποθετεί ένα marker στο σημείο αυτό. Ακολουθεί η εικόνα:



Εικόνα 15. Σήμα GPS

Όπως βλέπουμε στα screenshots πάνω από το χάρτη υπάρχουν δύο κουμπιά με τα οποία ξεκινάει και σταματάει η καταγραφή της διαδρομής με τις function **startRec()** και **stopRec()**. Αρχικά το κουμπί της διακοπής «stop» είναι απενεργοποιημένο και το κουμπί της εγγραφής «start» ενεργοποιημένο. Με το κλικ του «start» καλείται η function **startRec** η οποία εκτελεί την function **getTrip()** για να πάρει τον αριθμό της ημερήσιας διαδρομής του χρήστη. Μετά με χρήση των timing events του αντικειμένου window καλεί την function **makepostCall()** κάθε 20 δευτερόλεπτα. Επιπλέον απενεργοποιεί το κουμπί της εγγραφής για να μην το μπορεί να το ξαναπατήσει και ενεργοποιεί το κουμπί της διακοπής.

Η function **getTrip()** καλεί το api getNextTrip με παράμετρο την global μεταβλητή user. Το response του api επιστρέφει τον αριθμό της επόμενης ημερήσιας διαδρομής του χρήστη ο οποίος αποθηκεύεται στην global μεταβλητή trip. Εμφανίζεται ένα notification στο χρήστη με τον αριθμό της διαδρομής όπως βλέπουμε στην παρακάτω εικόνα:

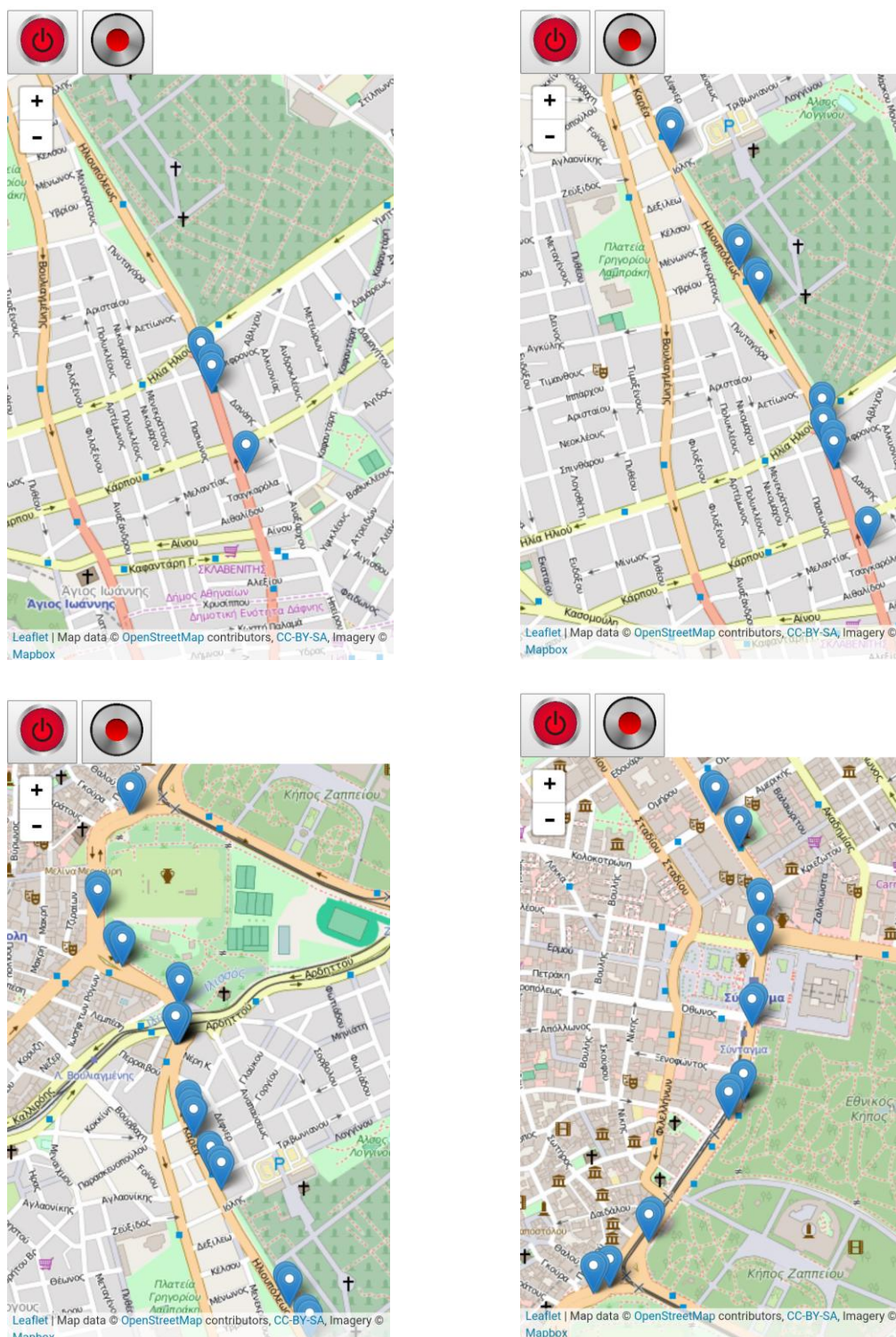


Εικόνα 16. Έναρξη καταγραφής διαδρομής

Η function **makePostCall()** εκτελεί την function **setDeviceInfo()** για να λάβει συντεταγμένες και ταχύτητα από το gps. Εκτελεί την function **redraw** η οποία τοποθετεί ένα marker στο νέο σημείο του χάρτη. Μετατρέπει την ταχύτητα σε km/hour γιατί το gps δίνει meter/sec. Καλεί το api senddlocation με XMLHttpRequest στέλνοντας τα στοιχεία του χρήστη, της συσκευής και της θέσης. Δηλαδή στέλνει τις τιμές που έχουν οι μεταβλητές user και trip, τα latitude, longitude της πιο πρόσφατης θέσης, η ταχύτητα σε km/hour, και η ταυτότητα της συσκευής για αποθήκευση στη βάση. Τέλος καλείται η function **checkSpeed()** η οποία κάνει έλεγχο των ορίων ταχύτητας.

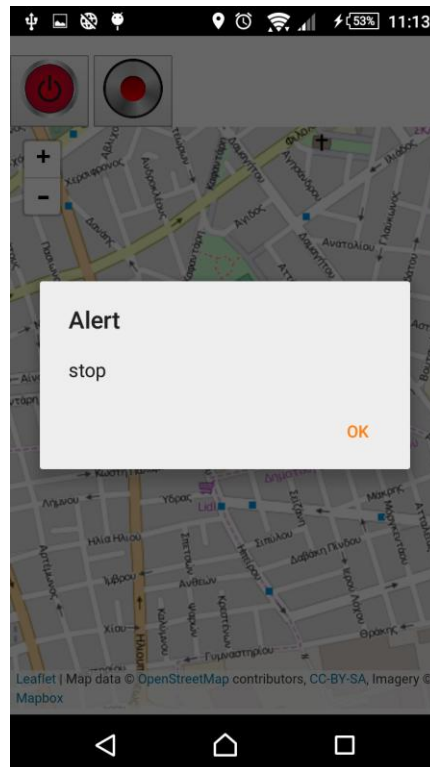
Η function ***checkSpeed()*** παίρνει σαν παράμετρο την ταχύτητα και συγκρίνει αν είναι πάνω από κάποια όρια. Τυχαιά επιλέχτηκε σαν πρώτο όριο τα 80 km/h το οποίο αν ξεπεραστεί το τηλέφωνο κάνει vibration για ένα δευτερόλεπτο. Όταν η ταχύτητα υπερβεί τα 100 km/h το τηλέφωνο κάνει έναν ήχο που είναι ο default που έχει επιλέξει ο χρήστης για notification. Η συγκεκριμένη λειτουργία δεν μπορεί να απεικονιστεί.

Παρακάτω φαίνονται εικόνες από την πλοήγηση μετά την ενεργοποίηση της καταγραφής.



Εικόνα 17. Καταγραφή διαδρομής

Αντίστοιχα στο κλικ του κουμπιού διακοπής «stop» σταματάει η καταγραφή της διαδρομής με την κλήση της function **stopRec**. Η **stopRec()** τερματίζει το timing event που είχε ξεκινήσει η function startRec και καλούσε την makePostCall για αποθήκευση δεδομένων στη βάση κάθε 20 δευτερόλεπτα. Απενεργοποιεί το κουμπί διακοπής «stop» και ενεργοποιεί το κουμπί έναρξης της καταγραφής «start». Εμφανίζει ένα μήνυμα στο χρήστη όπως βλέπουμε:

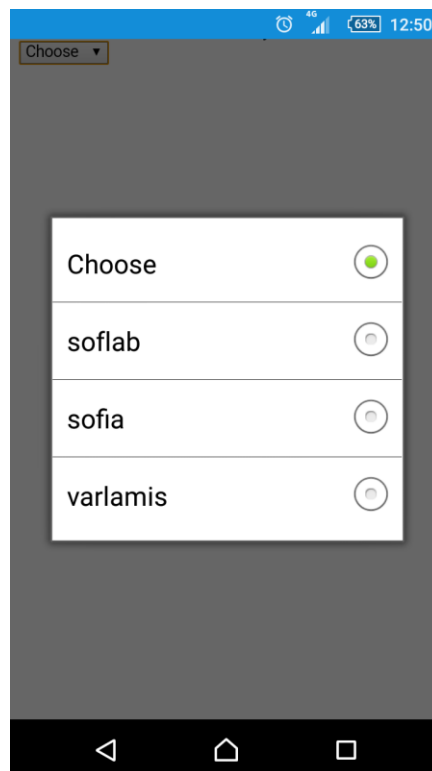


Εικόνα 18. Διακοπή Καταγραφής Διαδρομής

c) use case data analytics

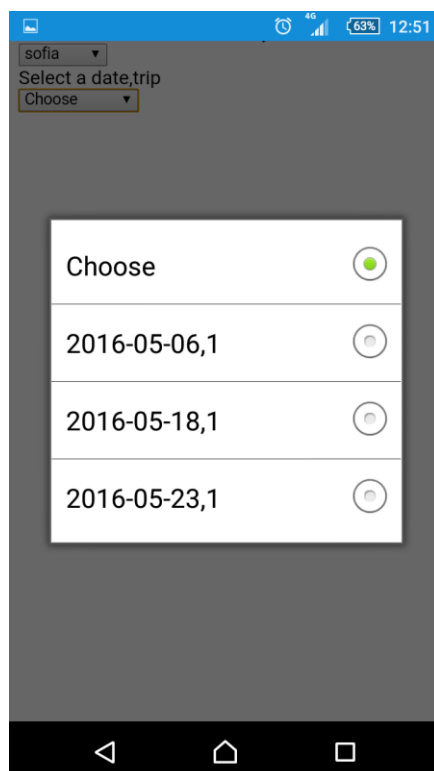
reports.html

Σε αυτή την ιστοσελίδα παρουσιάζονται αναλυτικά στοιχεία των διαδρομών των χρηστών. Η ιστοσελίδα είναι διαθέσιμη και λειτουργεί εξίσου καλά είτε σε web είτε σε mobile εφαρμογή. Στο onload εκτελείται η function **getusers()**. Η getusers καλεί το api retrieveDrivers με XMLHttpRequest και από το json response παίρνει τα usernames και τα εμφανίζει σε ένα μενού επιλογών όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 19. Επιλογή χρήστη

Στο onchange event του μενού επιλογών καλείται η function **getUserDates()** με παράμετρο τον επιλεγμένο χρήστη. Με όμοιο τρόπο η getUserDates καλεί το api retrieveDriverDates και φέρνει σε μενού επιλογών της ημερομηνίες και της διαδρομές του χρήστη. Εμφανίζει τις ημερομηνίες που έχει διαδρομές ο χρήστης με αύξουσα σειρά. Δίπλα από την ημερομηνία εμφανίζεται ο αριθμός της ημερήσιας διαδρομής επίσης με αύξουσα σειρά. Βλέπουμε παράδειγμα στην παρακάτω οθόνη:



Εικόνα 20. Επιλογή διαδρομής

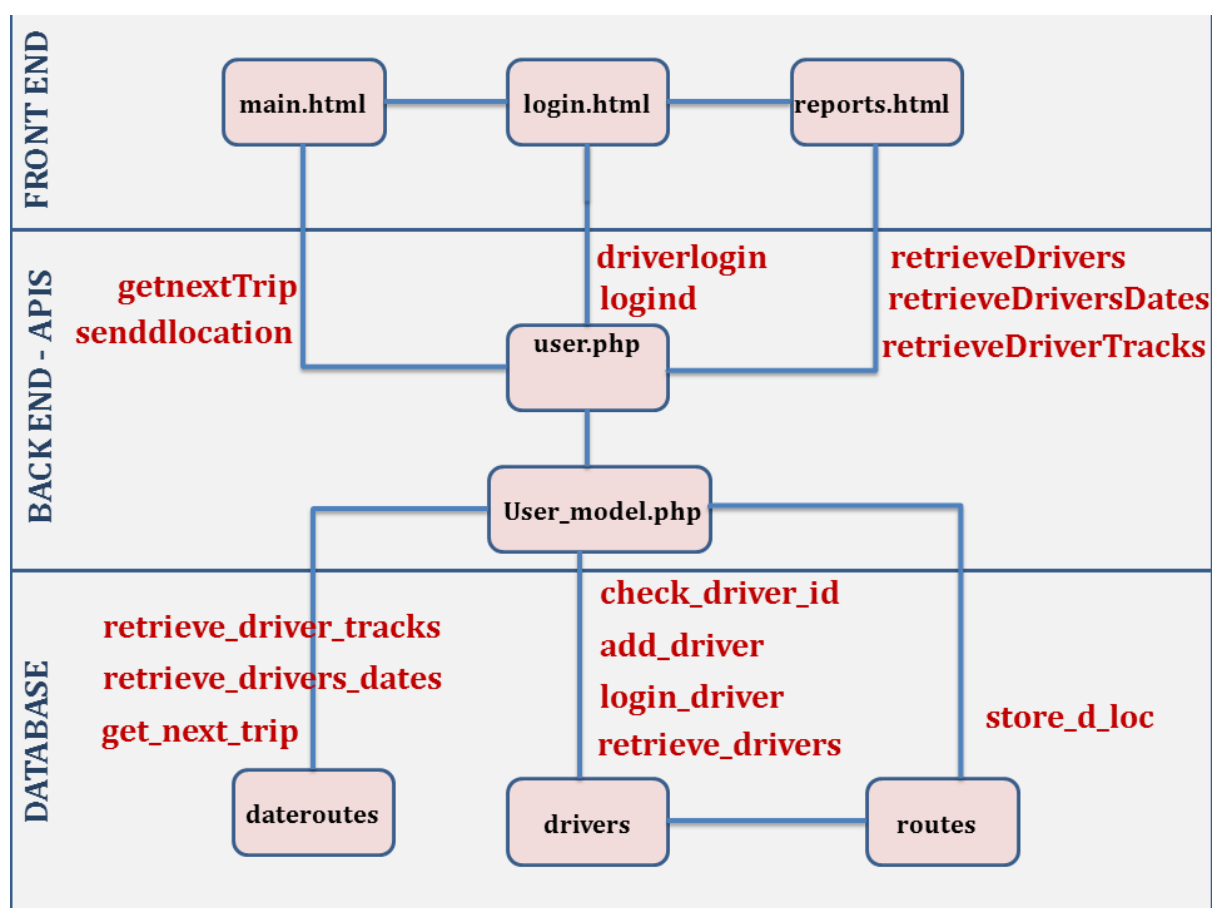
Στο onchange event του μενού επιλογών καλείται η function ***getUserTracks()*** που επιστρέφει την επιλεγμένη διαδρομή του χρήστη. Η function καλεί το api retrieveDriverTracks με παραμέτρους τον επιλεγμένο χρήστη, ημερομηνία και διαδρομή και επιστρέφει όλα τα σημεία σε έναν πίνακα. Για το σύνολο των σημείων της διαδρομής εκτελείται η function ***init()***.

Η function ***init()*** διαχειρίζεται την εμφάνιση της διαδρομής στο χάρτη και τον υπολογισμό των αναλυτικών στοιχείων. Αρχικά κάνει focus το χάρτη στο πρώτο σημείο της διαδρομής και στη συνέχεια διαβάζει όλα τα σημεία και κάνει υπολογισμούς. Κρατάει σε μεταβλητές την ώρα έναρξης και λήξης της διαδρομής. Προσθέτει όλες τις ταχύτητες όλων των σημείων σε μια μεταβλητή για να βρει το μέσο όρο ταχύτητας. Συγκρίνει όλες τις ταχύτητες για να βρει τη μέγιστη. Υπολογίζει την απόσταση δύο διαδοχικών σημείων με τη function distance και την αθροίζει σε ένα σύνολο. Μετράει τις στάσεις της διαδρομής. Υπολογίζει τη διάρκεια της διαδρομής και τη διάρκεια των στάσεων. Επίσης μετράει τις παραβάσεις ταχύτητας. Παράλληλα τοποθετεί ένα marker για κάθε σημείο της διαδρομής. Τα marker είναι διαφορετικού χρώματος ανάλογα με την ταχύτητα. Με μπλε φαίνεται η έναρξη της διαδρομής. Για μικρές ταχύτητες μέχρι 20 χμ/ώρα τοποθετείται κόκκινο marker στο χάρτη. Για ταχύτητα από 20 μέχρι 60 χμ/ώρα μπαίνει πράσινο και για πάνω από 60 πορτοκαλί. Κάτω από το χάρτη εμφανίζονται τα υπολογισμένα στοιχεία: μέση ταχύτητα, μέγιστη ταχύτητα, απόσταση, ώρα έναρξης, ώρα τέλους, διάρκεια, στάσεις, χρόνος στάσης.

Στο παρακάτω διάγραμμα φαίνεται πως χωρίζεται ο κώδικας στα 3 επίπεδα (front end, back end και βάση δεδομένων) και πως επικοινωνούν τα αρχεία μεταξύ τους.

Οι σελίδες χρησιμοποιούν τα apis του controller user και ο user καλεί τις function του model user_model για επικοινωνία με τη βάση. Η login χρησιμοποιεί το api driverlogin για register και το loginid για το login όπου του επιτρέπεται πρόσβαση στην main. Η main βρίσκει την επόμενη διαδρομή του χρήστη με το api getNextTrip και στέλνει τα δεδομένα με το senddlocation για καταγραφή των σημείων της διαδρομής. Η reports καλεί τα api retrieveDrivers, retrieveDriversDates και retrieveDriverTracks για εμφάνιση της διαδρομής.

Ομοίως ο user καλεί τις function check_driver_id, add_driver, login_driver, retrieve_drivers του user_model για έλεγχο, επιλογή και εισαγωγή στοιχείων στον πίνακα drivers. Επίσης χρησιμοποιεί την store_d_bc για καταχώρηση στον πίνακα routes. Οι function get_next_trip, retrieve_driver_dates και retrieve_driver_tracks χρησιμοποιούνται για επιλογή στοιχείων από την όψη dateroutes.



Εικόνα 21. Διάγραμμα κώδικα

ΔΟΚΙΜΕΣ

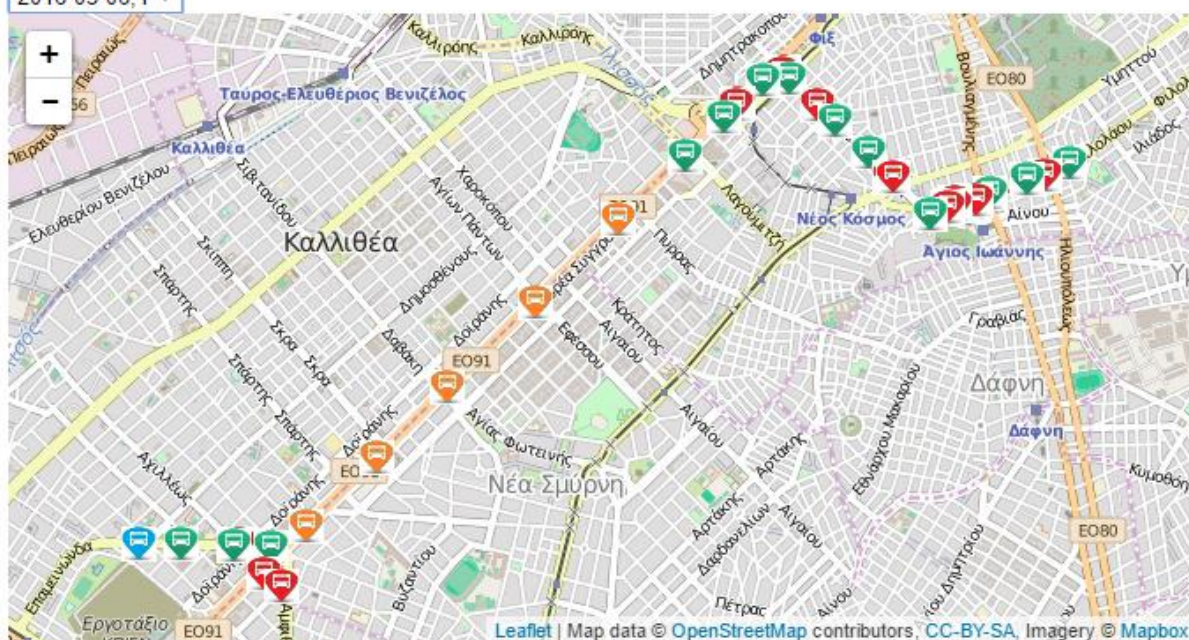
Βλέπουμε παρακάτω μερικές διαδρομές και τα στοιχεία που προκύπτουν από αυτές. Για λόγους καλύτερης ανάλυσης οι επόμενες οθόνες παρουσιάζονται από την web εφαρμογή.

Select a driver to see his daily track

soflab ▼

Select a date, trip

2016-05-06, 1 ▼



Average speed: 30.3 km/h. Max speed: 89.1 km/h. Distance: 5.466 km.

Start: 17:08:33. End: 17:22:39. Duration: 0:14:6 h:m:s. Stops: 6. Stopped: 1:22 m:s.

You had 5 speedlimit violations.

Εικόνα 222. Εμφάνιση Διαδρομής Χρήστη (Δοκιμή 1^η)

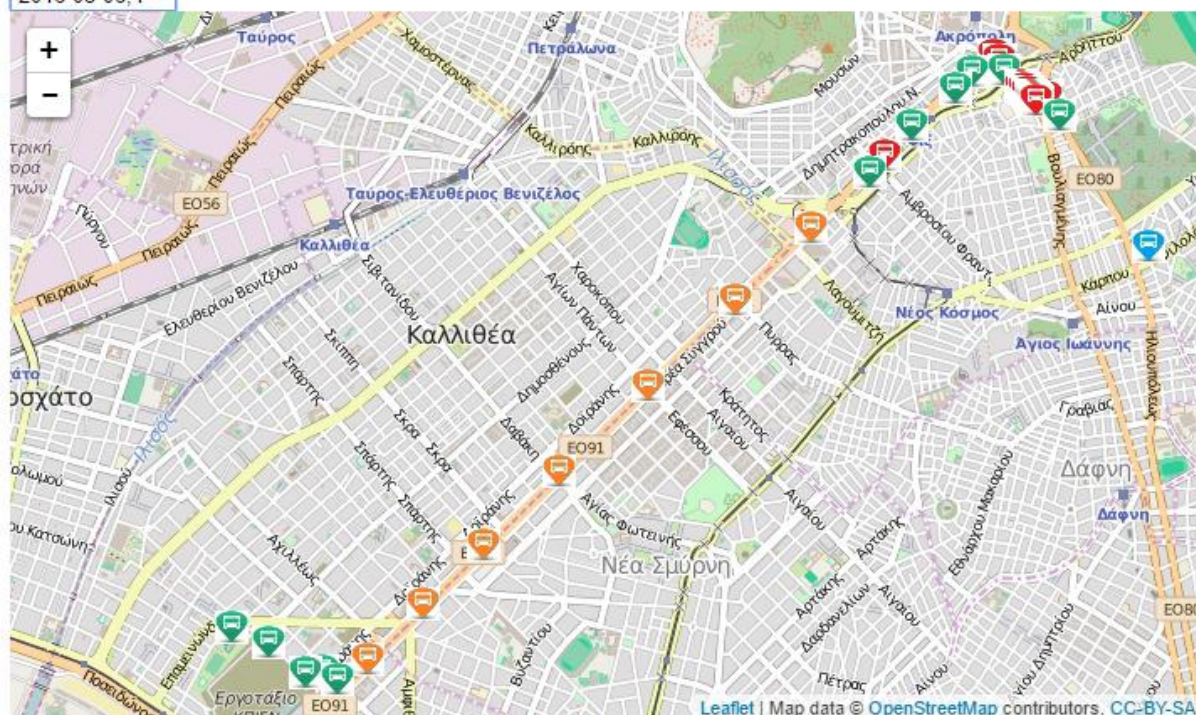
Αυτή η διαδρομή είχε μέση ταχύτητα 30.3 km/h, μέγιστη ταχύτητα 89.1 km/h και συνολική απόσταση 5.466 km. Η διαδρομή ξεκίνησε στις 17:08:33 και τέλεωσε 17:22:39, διάρκεια 14 λεπτά και 6 δευτερόλεπτα. Το όχημα σταμάτησε 6 φορές και παρέμεινε σταματημένο 1 λεπτό και 22 δευτερόλεπτα. Υπήρξαν 5 παραβάσεις ταχύτητας. Το μπλε marker δείχνει την αφετηρία της διαδρομής.

Select a driver to see his daily track

sofia ▼

Select a date, trip

2016-05-06, 1 ▼



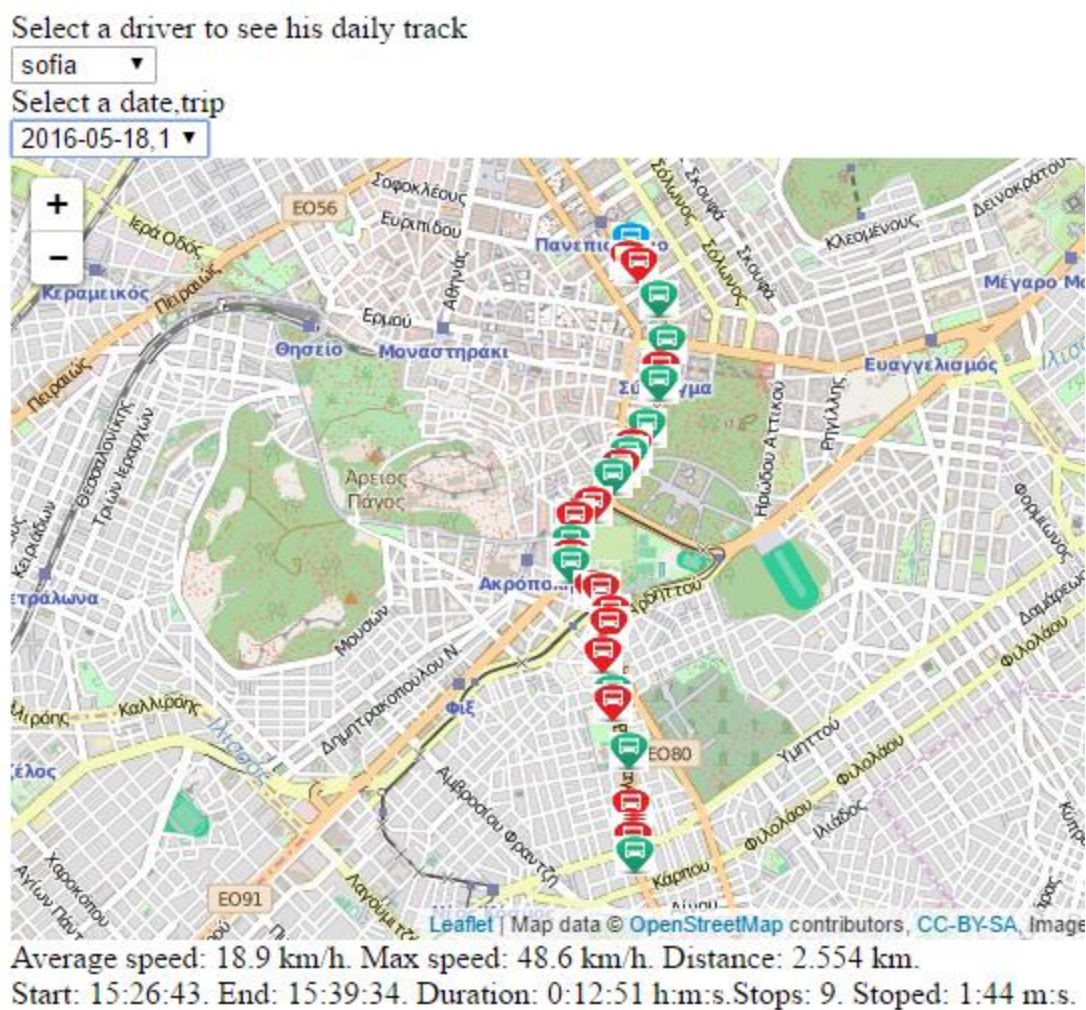
Average speed: 39.1 km/h. Max speed: 102.6 km/h. Distance: 5.772 km.

Start: 15:29:44. End: 15:43:06. Duration: 0:13:22 h:m:s. Stops: 5. Stopped: 2:40 m:s.

You had 7 speedlimit violations.

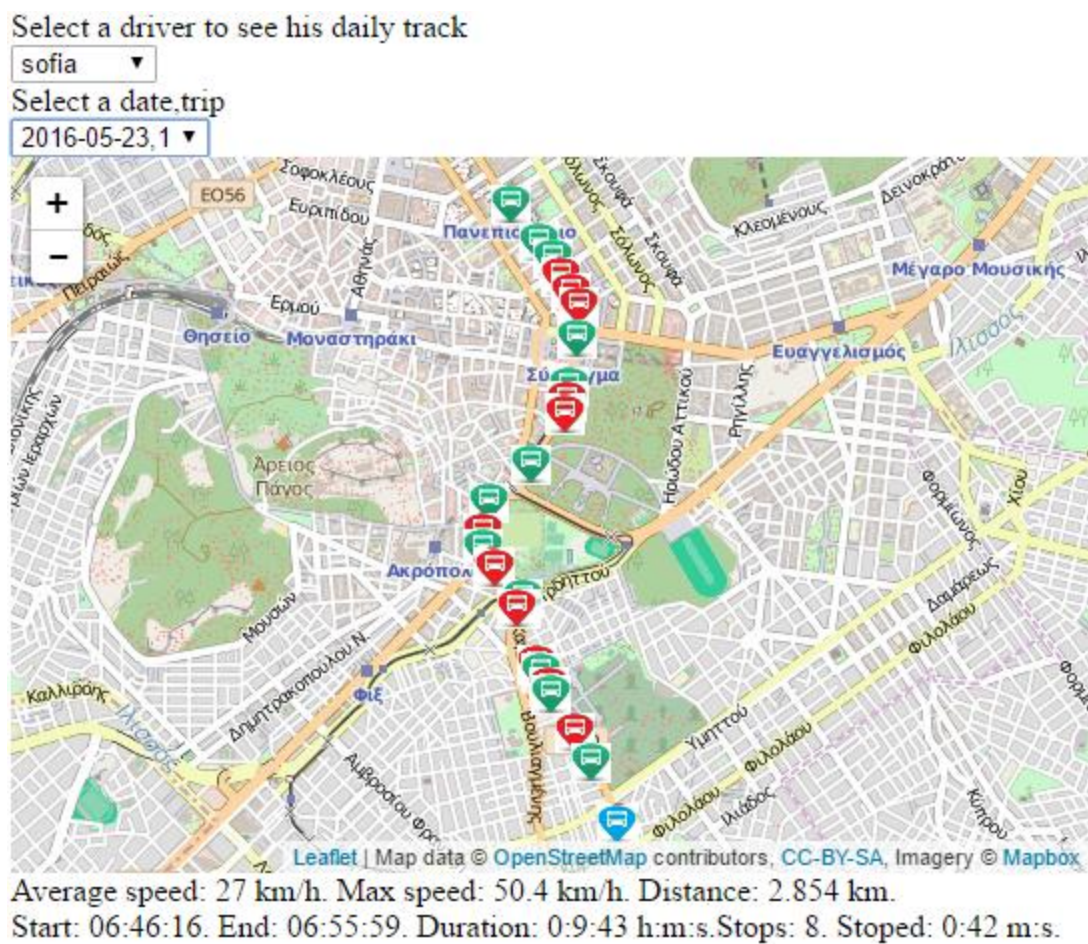
Εικόνα 233. Εμφάνιση Διαδρομής Χρήστη (Δοκιμή 2^η)

Εδώ βλέπουμε την ίδια διαδρομή με αντίθετη κατεύθυνση 2 ώρες νωρίτερα. Αυτή η διαδρομή είχε μέση ταχύτητα 39.1 km/h, μέγιστη ταχύτητα 102.6 km/h και συνολική απόσταση 5.772 km. Η διαδρομή ξεκίνησε στις 15:29:44 και τέλεωσε 15:43:06, διάρκεια 13 λεπτά και 22 δευτερόλεπτα. Το όχημα σταμάτησε 5 φορές και παρέμεινε σταματημένο 2 λεπτά και 40 δευτερόλεπτα. Υπήρξαν 7 παραβάσεις ταχύτητας. Το μπλε marker δείχνει την αφετηρία της διαδρομής. Δεν παρατηρούμε σημαντικές διαφορές στις δύο κατευθύνσεις.



Εικόνα 244. Εμφάνιση Διαδρομής Χρήστη (Δοκιμή 3')

Εδώ φαίνεται μια διαδρομή από το κέντρο της Αθήνας μεσημεριανή ώρα. Αυτή η διαδρομή είχε μέση ταχύτητα 18.9 km/h, μέγιστη ταχύτητα 48.6 km/h και συνολική απόσταση 2.554 km. Η διαδρομή ξεκίνησε στις 15:26:43 και τέλειωσε 15:39:34, διάρκεια 12 λεπτά και 51 δευτερόλεπτα. Το όχημα σταμάτησε 9 φορές και παρέμεινε σταματημένο 1 λεπτό και 44 δευτερόλεπτα. Το μπλε marker δείχνει την αφετηρία της διαδρομής.



Εικόνα 255. Εμφάνιση Διαδρομής Χρήστη (Δοκιμή 4^η)

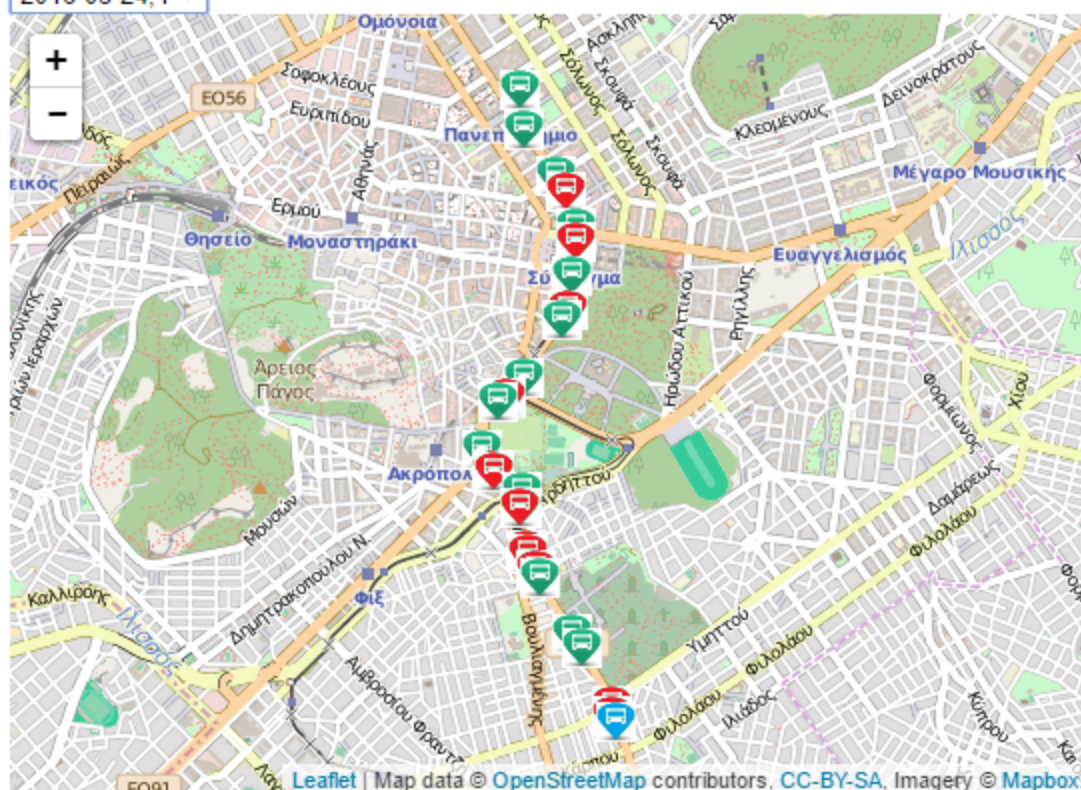
Εδώ βλέπουμε μια διαδρομή προς στο κέντρο της Αθήνας πρωινή ώρα. Αυτή η διαδρομή είχε μέση ταχύτητα 27 km/h, μέγιστη ταχύτητα 50.4 km/h και συνολική απόσταση 2.854 km. Η διαδρομή ξεκίνησε στις 6:46:16 και τέλειωσε 6:55:59, διάρκεια 9 λεπτά και 43 δευτερόλεπτα. Το όχημα σταμάτησε 8 φορές και παρέμεινε σταματημένο 42 δευτερόλεπτα. Το μπλε marker δείχνει την αφετηρία της διαδρομής.

Select a driver to see his daily track

soflab ▾

Select a date, trip

2016-05-24, 1 ▾



Average speed: 22.5 km/h. Max speed: 42.3 km/h. Distance: 2.704 km.

Start: 06:47:07. End: 06:56:44. Duration: 0:9:37 h:m:s. Stops: 5. Stopped: 1:0 m:s.

Εικόνα 266. Εμφάνιση Διαδρομής Χρήστη (Δοκιμή 5^η)

Εδώ βλέπουμε την ίδια διαδρομή προς στο κέντρο της Αθήνας πρωινή ώρα την επόμενη μέρα. Αυτή η διαδρομή είχε μέση ταχύτητα 22.5 km/h, μέγιστη ταχύτητα 42.3 km/h και συνολική απόσταση 2.704 km. Η διαδρομή ξεκίνησε στις 6:47:07 και τέλειωσε 6:56:44, διάρκεια 9 λεπτά και 37 δευτερόλεπτα. Το όχημα σταμάτησε 5 φορές και παρέμεινε σταματημένο 1 λεπτό. Το μπλε marker δείχνει την αφετηρία της διαδρομής.

Παρατηρούμε ότι τις πρωινές ώρες είναι πιο άνετη η κυκλοφορία στο κέντρο της πόλης.

Κεφάλαιο 5: Συμπεράσματα

Συμπεράσματα

Από την έρευνα των επιστημονικών άρθρων και δημοσιεύσεων προέκυψε το συμπέρασμα ότι υπάρχουν αμέτρητες εφαρμογές προσδιορισμού, εντοπισμού και καταγραφής της τοποθεσίας. Ο λόγος είναι ότι δεν απαιτείται ιδιαίτερος εξοπλισμός εφόσον οι σύγχρονες κινητές συσκευές διαθέτουν αισθητήρα gps και οι περισσότεροι έχουν στην κατοχή τους τουλάχιστον μία. Αυτό διαπιστώθηκε και στην πράξη διότι κατά την υλοποίηση της εφαρμογής τα μόνα δεδομένα που εκμεταλλεύτηκαν ήταν αυτά που παρέχει το gps.

Συγκρίσεις

Οι εφαρμογές αυτές έχουν υλοποιηθεί είτε σε native code, δηλαδή για μία πλατφόρμα, είτε σε cross platform εργαλεία που απευθύνονται σε όλες τις πλατφόρμες κινητών συσκευών. Οι Native εφαρμογές συνήθως λειτουργούν και έχουν καλύτερη αίσθηση αλλά είναι συχνά πιο ακριβές στην ανάπτυξη, ιδίως για τις εταιρείες που θέλουν εφαρμογές σε πολλαπλές πλατφόρμες OS. Οι cross platform εφαρμογές προσφέρουν την περισσότερη λειτουργικότητα και προσαρμογή για τον χρήστη αλλά η ανάπτυξη απαιτεί επιπλέον χρόνο και προσπάθεια για να μιμηθεί την Native εμφάνιση και αίσθηση. Δεν υπάρχει απαραίτητα καλύτερη ή χειρότερη προσέγγιση. Η επιλογή εξαρτάται κάθε φορά από τις ανάγκες της εφαρμογής και τις προτεραιότητες του προγραμματιστή. Οι κυριότεροι παράγοντες είναι χρόνος, κόστος, απόδοση και εμφάνιση. Η συγκεκριμένη εφαρμογή που υλοποιήθηκε σε cross platform τεχνολογία ίσως υστέρησε στο ότι δεν είχε πλήρη πρόσβαση στο hardware του κινητού. Ίσως αν γινόταν σε android ή ios να μπορούσαμε να ενεργοποιούμε το gps από την εφαρμογή και να είχε καλύτερη απόδοση όταν η εφαρμογή λειτουργεί στο background.

Χρησιμότητα

Ιδιαίτερα σημαντικό σε κάθε εφαρμογή είναι να μπορεί να έχει κάποια χρησιμότητα. Η συγκεκριμένη εφαρμογή θα μπορούσε να χρησιμοποιηθεί από οποιονδήποτε οδηγό που θέλει να γνωρίζει καθημερινά τι διαδρομές κάνει, πόσα χιλιόμετρα διανύει και πόσο χρόνο είναι καθοδόν. Ακόμα πιο χρήσιμη ίσως ήταν σε κάποιον που έχει στη διάθεση του πολλά οχήματα και θα ήθελε να έχει τον έλεγχο τους. Για παράδειγμα κάποιος που έχει κέντρο ταξί θα μπορούσε να ξέρει καθημερινά τι διαδρομές έκανε κάθε όχημα και ακόμα να κάνει και παρατηρεί αν κάποιος οδηγός ξέφευγε από τη διαδρομή του ή έτρεχε με υπερβολική ταχύτητα. Επίσης θα γνώριζε κατά τη διάρκεια της μέρας ανά πάσα στιγμή αν έχουν φτάσει στον προορισμό τους. Αυτό θα είχε χρησιμότητα και σε άλλες εταιρίες που τους ενδιαφέρει να βρίσκονται τα οχήματα σε συγκεκριμένους προορισμούς και συγκεκριμένη ώρα, όπως πχ τις ταχυμεταφορές ή τα λεωφορεία. Θα μπορούσε να χρησιμοποιηθεί για γονικό έλεγχο στους νέους οδηγούς που δανειζονται τα αυτοκίνητα των γονιών τους. Θα είχε ενδιαφέρον να μπορούσαν να αξιοποιηθούν αυτά τα στατιστικά που προκύπτουν για να βελτιωθεί η κυκλοφορία. Με τον εντοπισμό των σημείων που δημιουργείτε συμφόρηση ενδεχομένως να δινόταν μια λύση. Θα μπορούσε σε αυτά τα σημεία εκείνες τις ώρες να υπάρχει τροχονόμος ή να ρυθμιστούν τα φανάρια σε διαφορετικούς χρόνους ή να δημιουργηθεί μία στάση MMM.

Μελλοντικές εργασίες

Η εφαρμογή θα μπορούσε να επεκταθεί για ακόμα μεγαλύτερη ασφάλεια οδήγησης. Για παράδειγμα θα μπορούσε να δίνεται η δυνατότητα αυτόματης απόρριψης κλήσεων κατά τη χρήση της εφαρμογής. Μια άλλη επέκταση θα ήταν να προστεθούν πάνω στο χάρτη τα σημεία ενδιαφέροντος. Πολύ χρήσιμο θα ήταν να υπάρχουν περισσότερες ειδοποιήσεις από την εφαρμογή για μεγαλύτερη οδική ασφάλεια. Θα προφύλασσε τον οδηγό να υπάρχει ειδοποίηση για ραντάρ και κάμερες ταχύτητας.

Τέλος, υπάρχουν άπειρες εφαρμογές που βασίζονται στο παγκόσμιο σύστημα εντοπισμού θέσης που έχουν υλοποιηθεί σε διάφορες τεχνολογίες και παρέχουν πολλαπλές λειτουργικότητες που πλέον η πιθανότητα να υλοποιηθεί κάτι καινούριο είναι πολύ μικρή.

Βιβλιογραφία

- Abbott E., Powell D.. (2002). Land-vehicle navigation using GPS. *Proceedings of the IEEE*. 87(1), 145 – 162. <http://dx.doi.org/10.1109/5.736347>
- Almomani M. Iman, Alkhalil Y. Nour, Ahmad M. Enas , Jodeh M. Rania. (2011). Ubiquitous GPS vehicle tracking and management system. *Applied Electrical Engineering and Computing Technologies (AEECT)*. 1 – 6. <http://dx.doi.org/10.1109/AEECT.2011.6132526>
- Amini Arghavan, Vaghefi M. Reza, de la Garza M. Jesus, Buehrer R. Michael. (2014). GPS-free cooperative mobile tracking with the application in vehicular networks. *Positioning, Navigation and Communication (WPNC)*. 1 – 6. <http://dx.doi.org/10.1109/WPNC.2014.6843293>
- Behzada M., Sanab A., Khanc M.A., Walayatb Z., Qasimd U., Khane Z.A., Javaid N.. (2014). Design and Development of a Low Cost Ubiquitous Tracking System. *Procedia Computer Science*. 34, 220 –227. <http://dx.doi.org/10.1016/j.procs.2014.07.014>
- Chandra Ankur, Jain Shashank, Qadeer A. Mohammed (2011). GPS Locator: An Application for Location Tracking and Sharing Using GPS for Java Enabled Handhelds. *Computational Intelligence and Communication Networks (CICN)*. 406 – 410. <http://dx.doi.org/10.1109/CICN.2011.85>
- Chew S. H., Chong P. A., Gunawan E., Goh K. W.. (2006). A Hybrid Mobile-based Patient Location Tracking System for Personal Healthcare Applications. *Engineering in Medicine and Biology Society*. 5188 – 5191. <http://dx.doi.org/10.1109/IEMBS.2006.259579>

- Hind Abdalsalam Abdallah Dafallah. (2014). Design and implementation of an accurate real time GPS tracking system. *e-Technologies and Networks for Development (ICeND)*. 183 – 188. <http://dx.doi.org/10.1109/ICeND.2014.6991376>
- Kadibagil Mahesh, Guruprasad H S. (2014). Position Detection and Tracking System. *IRACST - International Journal of Computer Science and Information Technology & Security (IJCSITS)*. 4(3), 67-73
- Lee SeokJu, Tewolde Girma, Kwon Jaerock. (2014). Design and implementation of vehicle tracking system using GPS/GSM/GPRS technology and smartphone application. *Internet of Things (WF-IoT)*. 353 – 358. <http://dx.doi.org/10.1109/WF-IoT.2014.6803187>
- Mane Suresh Pradip, Khairnar Vaishali. (2014). Analysis of Bus Tracking System Using Gps on Smart Phones. *IOSR Journal of Computer Engineering (IOSR-JCE)*. 16(3), 80-82. Retrieved from www.iosrjournals.org
- Patil Bharath, Patil Radhika, Pittet Andre. (2011). Energy saving techniques for GPS based tracking applications. *Integrated Communications, Navigation and Surveillance Conference (ICNS)*. 2155-4943. <http://dx.doi.org/10.1109/ICNSURV.2011.5935335>
- Pham H. Dat, Drieberg Micheal, Nguyen C. Chi. (2013). Development of vehicle tracking system using GPS and GSM modem. *Open Systems (ICOS)*. 89 – 94. <http://dx.doi.org/10.1109/ICOS.2013.6735054>
- Varandas C. M. Luís, Vaidya Binod, Rodrigues J. P. C. Joel. (2010). mTracker: A Mobile Tracking Application for Pervasive Environment. *Advanced Information Networking and Applications Workshops (WAINA)*. 962 – 967. <http://dx.doi.org/10.1109/WAINA.2010.164>

Παράρτημα

Κώδικας Βάσης Δεδομένων

Ο κώδικας για την δημιουργία του πίνακα **drivers** είναι:

```
CREATE TABLE public.drivers
(
    driverid bigint NOT NULL DEFAULT nextval('drivers_driverid_seq'::regclass),
    email character varying(50),
    phone_num character varying(25),
    password character varying(150) NOT NULL,
    username character varying(40) NOT NULL,
    CONSTRAINT drivers_pkey PRIMARY KEY (username)
)
```

Ο κώδικας για την δημιουργία του πίνακα **routes** είναι:

```
CREATE TABLE public.routes
(
    deviceid character varying(80) NOT NULL,
    "user" character varying(40) NOT NULL,
    lat_lon geometry,
    "timestamp" timestamp with time zone NOT NULL DEFAULT now(),
    speed double precision,
    trip integer DEFAULT 0,
    CONSTRAINT routes_pkey PRIMARY KEY (deviceid, "timestamp"),
    CONSTRAINT routes_user_fk FOREIGN KEY ("user")
        REFERENCES public.drivers (username) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

Παρακάτω φαίνεται ο κώδικας δημιουργίας της όψης **dateroutes:**

```
CREATE OR REPLACE VIEW public.dateroutes AS
SELECT routes.deviceid,
       routes."timestamp"::date AS day,
       date_trunc('second'::text, routes."timestamp")::time without time zone AS "time",
       st_x(routes.lat_lon) AS lat,
       st_y(routes.lat_lon) AS lon,
       routes.speed,
       routes."user",
       routes.trip
FROM routes;
```

Κώδικας Back end – api

Ο κώδικας του api **driverlogin:**

```
function driverlogin_post() {
    $username = $this->post('username');
    $email = $this->post('email');
    $phonenum = $this->post('phonenum');
    $password = $this->post('password');
    if ( !$username || !$password ) {
```

```

        $this->response([ 'status' => FALSE,
                        'message' => 'You have not provided complete data'
                        ], REST_Controller::HTTP_BAD_REQUEST);
    } else {
        $driverid = $this->user_model->check_driver_id($username);
        if ($driverid) {
            $sec_userid = rand(100, 999) . base64_encode($driverid);
            $this->response([ 'status' => TRUE,
                            'message' => 'User existed',
                            'username' => $username,
                            'seckey' => $sec_userid
                            ], REST_Controller::HTTP_OK);
        } else {
            $data["username"] = $username;
            $data["email"] = $email;
            $data["phone_num"] = $phonenum;
            $data["password"] = $password;
            $insert_userid = $this->user_model->add_driver($data);
            $sec_userid = rand(100, 999) . base64_encode($insert_userid);
            $this->response([ 'status' => TRUE,
                            'message' => 'User added',
                            'seckey' => $sec_userid
                            ], REST_Controller::HTTP_OK);        }    }    }
    
```

Παρακάτω φαίνεται ο κώδικας του ***logind:***

```

function logind_post() {
    $username = $this->post('username');
    $password = $this->post('password');
    if ( !$username || !$password ) {
        $this->response([ 'status' => FALSE,
                        'message' => 'You have not provided complete data'
                        ], REST_Controller::HTTP_BAD_REQUEST);
    } else {
        $username = $this->user_model->login_driver($username,$password);
        if ($username) {
            $this->response([ 'status' => TRUE,
                            'message' => 'User existed',
                            'username' => $username,
                            ], REST_Controller::HTTP_OK);        }
    } else {
        $this->response([ 'status' => FALSE,
                        'message' => 'User not registered'
                        ], REST_Controller::HTTP_BAD_REQUEST); }    }    }
    
```

Ακολουθεί ο κώδικας του ***getnextTrip:***

```

function getnextTrip_post(){
    $accessCode = $this->post('accessCode');
    $user = $this->post('user');
    if (!$accessCode || !$user || $accessCode <> "1234") {
        $this->response([ 'status' => FALSE,
                        'message' => 'You have not provided valid data'
                        ], REST_Controller::HTTP_BAD_REQUEST);
    }
    }
    
```

```
} else {  
    $trip = $this->user_model->getnext_trip($user);  
    $json = json_encode($trip);  
    $this->response($json);    }    }
```

Εδώ είναι ο κώδικας του **senddlocation**:

```
function senddlocation_post() {  
    $dev_id = $this->post('deviceid');  
    $user = $this->post('user');  
    $gps_lat = $this->post('lat');  
    $gps_lon = $this->post('lon');  
    $gps_speed = $this->post('speed');  
    $trip = $this->post('trip');  
    if (!$dev_id || !$gps_lat || !$gps_lon || !$user ) {  
        $this->response([ 'status' => FALSE,  
            'message' => 'You have not provided complete data'  
        ], REST_Controller::HTTP_BAD_REQUEST);  
    } else {  
        if ($this->user_model->store_d_loc($dev_id,$gps_lat, $gps_lon , $gps_speed,  
            $user, $trip)) {            $this->response([ 'status' => TRUE, 'msg' => 'all done'  
                ], REST_Controller::HTTP_OK);            }    } }
```

Αυτός είναι ο κώδικας του **retrieveDrivers**:

```
function retrieveDrivers_post() {  
    $accessCode = $this->post('accessCode');  
    if (!$accessCode || $accessCode <> "1234") {  
        $this->response([  
            'status' => FALSE,  
            'message' => 'You have not provided a valid access code'  
        ], REST_Controller::HTTP_BAD_REQUEST);  
    } else {  
        $all_users = $this->user_model->retrieve_drivers();  
        $json = json_encode($all_users);  
        $this->response($json);  
    } }
```

Εδώ έχουμε το **retrieveDriverDates**

```
function retrieveDriverDates_post() {  
    $accessCode = $this->post('accessCode');  
    $user = $this->post('user');  
    if (!$accessCode || !$user || $accessCode <> "1234") {  
        $this->response([            'status' => FALSE,  
            'message' => 'You have not provided valid data'  
        ], REST_Controller::HTTP_BAD_REQUEST);  
    } else {  
        $all_user_dates = $this->user_model->retrieve_driver_dates($user);  
        $json = json_encode($all_user_dates);  
        $this->response($json);    } }
```

retrieveDriverTracks

```
function retrieveDriverTracks_post() {  
    $accessCode = $this->post('accessCode');  
    $user = $this->post('user');
```

```

$day = $this->post('day');
$trip = $this->post('trip');
if (!$accessCode || !$user || !$day || !$trip || $accessCode <> "1234") {
    $this->response([
        'status' => FALSE,
        'message' => 'You have not provided valid data'
    ], REST_Controller::HTTP_BAD_REQUEST);
} else {
    $all_user_tracks = $this->user_model->retrieve_driver_tracks($user,
    $day,$trip);
    $json = json_encode($all_user_tracks);
    $this->response($json);    } }

```

functions called by api

check_driver_id

```

public function check_driver_id($username) {
    $this->db->select('driverid');
    $query = $this->db->get_where('drivers', array('username' => $username));
    if ($query->num_rows() == 1) {
        $driverid = $query->row()->driverid;
        return $driverid;
    } else {
        return FALSE;    } }

```

function add_driver

```

public function add_driver($data) {
    $this->db->insert('drivers', $data);
    log_message('info', $this->db->last_query());
    $insert_id = $this->db->insert_id();
    if ($insert_id > 0) {
        return $insert_id;    } }

```

function login_driver

```

public function login_driver($username,$password){
    $this->db->select('username');
    $this->db->from('drivers');
    $this->db->where('username',$username);
    $this->db->where('password',$password);
    $query = $this->db->get();
    if ($query->num_rows() == 1)
    {$deviceid = $query->row()->username;
    return $deviceid;
    } else {    return FALSE;    } }

```

getnext_trip

```

public function getnext_trip($user){
    $this->db->select('COALESCE(max(trip),0)+1 trip');
    $this->db->from('dateroutes');
    $this->db->where('user',$user);
    $this->db->where('day','now()');
    $query = $this->db->get();
    if ($query->num_rows() == 1)

```

```
{ $trip = $query->row()->trip;  
  return $trip;  
} else { return FALSE; } }
```

function store_d_loc

```
public function store_d_loc($dev_id, $lat, $lon, $speed, $user, $trip) {  
  $this->db->set('lat_lon', "ST_GeomFromText('POINT(" . $lat . " " . $lon . ")')",  
  FALSE);  
  $this->db->set('deviceid', $dev_id);  
  $this->db->set('speed', $speed);  
  $this->db->set('user', $user);  
  $this->db->set('trip', $trip);  
  $query = $this->db->insert('routes');  
  log_message('info', 'QUERY'. $this->db->last_query());  
  return TRUE; }
```

retrieve_drivers

```
public function retrieve_drivers(){  
  $this->db->select('email,username,phone_num');  
  $this->db->from('drivers');  
  $query = $this->db->get();  
  if ($query->num_rows() > 0)  
  {  
    return $query->result();  
  }  
  else{ return FALSE; }
```

retrieve_driver_dates

```
public function retrieve_driver_dates($user){  
  $this->db->distinct();  
  $this->db->select('day,trip');  
  $this->db->from('dateroutes');  
  $this->db->where('user',$user);  
  $this->db->order_by("day,trip", "asc");  
  $query = $this->db->get();  
  if ($query->num_rows() > 0)  
  {  
    return $query->result();  
  }  
  else{  
    return FALSE; }
```

retrieve_driver_tracks

```
public function retrieve_driver_tracks($user,$day,$trip){  
  $this->db->select('time,lat,lon,speed');  
  $this->db->from('dateroutes');  
  $this->db->where('user',$user);  
  $this->db->where('day',$day);  
  $this->db->where('trip',$trip);  
  $this->db->order_by("time", "asc");  
  $query = $this->db->get();  
  if ($query->num_rows() > 0)  
  {  
    return $query->result();  
  }
```



```
    }  
    else{  
        return FALSE;    }    }
```

Front end – user interface

Index.html

(window.location='./login.html');

login.html

```
function get_local() {  
    var username = localStorage.getItem("username");  
    if (username) { document.getElementById("username").value = username;  
}  
  
    var password = localStorage.getItem("password");  
    if (password) { document.getElementById("password").value = password; }  
}
```

```
function validateEmail(email){  
    if (email!=""){  
        if (/^(.+)@(.+){2,}\.(.+){2,}$/ .test(email)) {return true;}  
        else {navigator.notification.alert("not vald mail");return false;  
        }  
    }else {return true;}  
}
```

```
function registerDevice() {  
    var user = document.getElementById("user").value;  
    var pass = document.getElementById("pass").value;  
    var phone = document.getElementById("phone").value;  
    var email = document.getElementById("email").value;  
    if (user!=""&& pass!="")  
    { if (validateEmail(email)===true){  
        var urlDev = "http://gianta.xyz/gpstrackback/api/user/driverlogin";  
        var formDevData = new FormData();  
        formDevData.append('username', user);  
        formDevData.append('email', email);  
        formDevData.append('phonenum', phone);  
        formDevData.append('password', pass);  
        var xhrDev = new XMLHttpRequest();  
        xhrDev.open('POST', urlDev, true);  
        xhrDev.onload = function (e) {  
            navigator.notification.alert(this.response);  
            if (xhrDev.status===200){  
                localStorage.setItem("username", user);  
                localStorage.setItem("password", pass);  
                window.location = "login.html";    }    };  
                xhrDev.send(formDevData);  
            }else {navigator.notification.alert("Please complete required");}    }  
}
```

```
function clear_local() {  
    localStorage.setItem("username", "");  
    localStorage.setItem("password", "");  
    navigator.app.exitApp();    }
```

```
function loginDevice( ) {
    var username = document.getElementById("username").value;
    var password = document.getElementById("password").value;
    if (username!=""&& password!="") {
        var urlDev = "http://gianta.xyz/gpstrackback/api/user/logind";
        var formDevData = new FormData();
        formDevData.append('username', username);
        formDevData.append('password', password);
        var xhrDev = new XMLHttpRequest();
        xhrDev.open('POST', urlDev, true);
        xhrDev.onload = function (e) {
            navigator.notification.alert(this.response);
            if (xhrDev.status===200){
                set_local();
                window.location = "main.html";          } };
        xhrDev.send(formDevData);
    } else {navigator.notification.alert("Please complete");} }

function set_local() {
    var username = document.getElementById("username").value;
    if (username) {    localStorage.setItem("username", username);  }
    var password = document.getElementById("password").value;
    if (password) {    localStorage.setItem("password", password);    } }

function onLoad() {
    user= localStorage.getItem("username");
    setDeviceInfo();
    init();
    if (navigator.userAgent.match(/(iPhone|iPod|iPad|Android|BlackBerry|IEMobile)/)) {
        document.addEventListener("deviceready", onDeviceReady, false);
    } else {
        onDeviceReady(); //this is the browser
    }
}

function setDeviceInfo() { if(navigator.geolocation){
    navigator.geolocation.watchPosition(positionFound,onError, { enableHighAccuracy: true ,
    maxAge: 30000, timeout: 60000}); } }

function positionFound(position) {
    latitude=position.coords.latitude;
    longitude=position.coords.longitude;
    speed=position.coords.speed;}

function onDeviceReady() {
    setDeviceInfo();
    document.addEventListener("resume", onResume, false);}

function onResume() { setDeviceInfo();}

function init() { map = new L.Map('map');
```

```
L.tileLayer('http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png?access_token=pk.eyJ1IjoIdmFyYbGFtaXMiLCJhIjoY2lneHJqeWVwMDBkMnZma3JieHRvNGF3cCJ9.DtUXdV-nMLaJdD2qHAMS7Q', {
    maxZoom: 18,
    attribution: 'Map data &copy; <a href="http://openstreetmap.org">OpenStreetMap</a> contributors, ' +
        '<a href="http://creativecommons.org/licenses/by-sa/2.0/">CC-BY-SA</a>, ' +
        'Imagery © <a href="http://mapbox.com">Mapbox</a>',
    id: 'mapbox.streets'}).addTo(map);
function onLocationError(e) {
    notify('locationError: '+e.message);
}
function onLocationFound(e) {
    L.Icon.Default.imagePath = 'images';
    L.marker(e.latlng).addTo(map);
}
map.on('locationfound', onLocationFound);
map.on('locationerror', onLocationError);
map.locate({setView: true, maxZoom: 16}); }
```

```
function startRec() {getTrip();
    myLoop = setInterval(makePostCall,time_interval);
    document.getElementById("RecButton").disabled = true;
    document.getElementById("StopButton").disabled = false; }
```

```
function getTrip() {
    var urlHit="http://gianta.xyz/gpstrackback/api/user/getnextTrip";
    var formData = new FormData();
    formData.append('accessCode','1234');
    formData.append('user', user);
    var xhr = new XMLHttpRequest();
    xhr.open('POST',urlHit, true);
    xhr.onload = function(e) {
        trip =this.response.replace("/g, ").replace(/\\g, "");
        trip=parseInt(trip);
        navigator.notification.alert('trip '+trip);
    };
    xhr.send(formData);}
```

```
function makePostCall() {
    var urlHit="http://gianta.xyz/gpstrackback/api/user/senddlocation";
    setDeviceInfo();
    var speed_kmh=speed*3.6;
    redraw();
    var formData = new FormData();
    formData.append('deviceid', device.uuid);
    formData.append('lat', latitude);
    formData.append('lon', longitude);
    formData.append('speed',speed_kmh);
    formData.append('user', user);
    formData.append('trip', trip);
    var xhr = new XMLHttpRequest();
    xhr.open('POST',urlHit, true);
    xhr.onload = function(e) {
    };
    xhr.send(formData);
    checkSpeed(speed_kmh);}
```

```
function redraw(){
    map.locate({setView: true, maxZoom: 16});
    L.marker([latitude, longitude]).addTo(map); }

function checkSpeed(spd){
    if (spd>80){ navigator.notification.vibrate(1000); }
    if (spd>100){ navigator.notification.beep(1); }}

function stopRec(){navigator.notification.alert('stop');
    clearInterval(myLoop);
    document.getElementById("StopButton").disabled = true;
    document.getElementById("RecButton").disabled = false;}

function getUsers(){
    var urlDev="http://gianta.xyz/gpstrackback/api/user/retrieveDrivers";
    var formDevData = new FormData();
    formDevData.append('accessCode','1234');
    var xhrDev = new XMLHttpRequest();
    xhrDev.open('POST',urlDev,true);
    xhrDev.onload = function(e){
        var json = '{"users":'+this.response.substring(1,this.response.length-1)+'}';
        json = json.replace(/\\/g, "");
        var inner='Select a driver to see his daily track<br>'+
        '<select name="users" id="umenu" onchange="getUserDates(this.value)">'+
        '<option value="Choose" selected="selected">Choose</option>';
        var json_arr=JSON.parse(json);
        var all_users=json_arr.users;
        for (var user in all_users){
            inner=inner+'<option
value="'+all_users[user].username+'">'+all_users[user].username+'</option>';
            inner=inner+'</select>';
            document.getElementById("userdiv").innerHTML=inner;
            xhrDev.send(formDevData); }

function getUserDates(username){
    var urlUserDates="http://gianta.xyz/gpstrackback/api/user/retrieveDriverDates";
    var formUserDates = new FormData();
    formUserDates.append('accessCode','1234');
    formUserDates.append('user',username);
    var xhrUserDates = new XMLHttpRequest();
    xhrUserDates.open('POST',urlUserDates,true);
    xhrUserDates.onload = function(e){
        var json = '{"dates":'+this.response.substring(1,this.response.length-1)+'}';
        json = json.replace(/\\/g, "");
        var usermenu=document.getElementById("umenu");
        var strUser = usermenu.options[usermenu.selectedIndex].value;
        var inner='Select a date,trip<br><select name="dates" '+
        'onchange="getUserTracks(this.value, \''+strUser+'\')"> '+
        '<option value="Choose" selected="selected">Choose</option>';
        var json_arr=JSON.parse(json);
        var all_user_dates=json_arr.dates;
```

```

        for (var dat in all_user_dates){
            inner=inner+'<option
value="'+all_user_dates[dat].day+','+all_user_dates[dat].trip+
            "'>'+all_user_dates[dat].day+','+all_user_dates[dat].trip+'</option>';
        }
        inner=inner+'</select>';
        document.getElementById("userdatesdiv").innerHTML=inner;
    };
    xhrUserDates.send(formUserDates);
}

```

```

function getUserTracks(daytrip,username){
    var urlDev="http://gianta.xyz/gpstrackback/api/user/retrieveDriverTracks";
    var formDevData = new FormData();
    var values =daytrip.split(',');
    var day=values[0];
    var trip=values[1];
    formDevData.append('accessCode','1234');
    formDevData.append('user',username);
    formDevData.append('day',day);
    formDevData.append('trip',trip);
    var xhrDev = new XMLHttpRequest();
    xhrDev.open('POST',urlDev,true);
    xhrDev.onload = function(e){
        var json = '{"tracks":'+this.response.substring(1,this.response.length-1)+'}';
        json = json.replace(/\\/"/g, "");
        var json_arr=JSON.parse(json);
        var all_user_tracks=json_arr.tracks;
        init(all_user_tracks);    };
    xhrDev.onerror = function(e){    alert(e);    };
    xhrDev.send(formDevData);    }

```

```

function init(all_user_tracks) {
    var inner='<leaflet-map fit-to-markers latitude="'+all_user_tracks[0].lat+"
longitude="'+all_user_tracks[0].lon+" zoom="14">';
    var speedsum=0; //sunolo gia mesi taxutita
    var speedcount=0; //metriseis gia meso oro taxutitas
    var speedlimitcount=0; //paravaseis oriou taxutitas
    var stops=0; //staseis
    var starttime; //wra enarxis (i wra tou prwtou simeio)
    var endtime; //wra telous
    var distsum=0; //sunoliki apostasi
    var maxspeed=0; //megisti taxutita
    var stoptime=0;
    for (var ut in all_user_tracks){
        var lati=parseFloat(all_user_tracks[ut].lat); //latitude
        var long=parseFloat(all_user_tracks[ut].lon); //longitude
        var speed=parseFloat(all_user_tracks[ut].speed); //taxutita
        if (ut==0){starttime=all_user_tracks[0].time;}
        endtime=all_user_tracks[ut].time; //sto teleutaio simeio tha xeroume tin wra telous
        speedsum=speedsum+speed; //athroizei oles tis taxutites
        if (speed>maxspeed) //sugkrinei tis taxutites gia na vrei ti megisti

```

```

    {maxspeed=speed;} //an einai megaluteri apo ti megisti tin krataei
    if (ut>0){ //apo to deuthero simeio vriskei apostaseis metaxu 2 simeiwon
        var dist= distance(all_user_tracks[ut-1].lat, all_user_tracks[ut-1].lon,
all_user_tracks[ut].lat, all_user_tracks[ut].lon);
        distsum=distsum+dist;} //prosthetei sti sunoliki apostasi
        if (speed>0) //an exei taxutita auxanei to metriti taxititas
        {speedcount++;}
        else{ //an den exei taxutita kai sto proigoumeno simeio eixe metraei mia stasi
            if (ut>0){
                if (all_user_tracks[ut-1].speed>0) {stops++;} //gia na min metraei parapanw staseis
                if (all_user_tracks[ut-1].speed==0){ // an den eixe taxutita oute sto proigoumeno simeio
                    var send = moment(all_user_tracks[ut].time, 'HH:mm:ss'); //metatropi se moments
                    var sstart = moment(all_user_tracks[ut-1].time, 'HH:mm:ss');
                    var sdur=send.diff(sstart, 'seconds', true); //vriskei ti diafora tous se sec
                    stoptime=stoptime+sdur; //tin prosthetei sto sunoliko xrono stasis
                }
                inner=inner+'<leaflet-marker latitude="'+lati+ '" longitude="'+long;
                if (ut==0){ //sto prwto simeio mple marker gia na xerei pou xekinise
                    inner=inner+' icon=\{"iconSize": [20, 20], "iconUrl":
"images/marker_bus_blue.png"}\';
                } else if (speed>20 && speed<=60){ //apo 20-60 km/h green marker
                    inner=inner+' icon=\{"iconSize": [20, 20], "iconUrl":
"images/marker_bus_green.png"}\';
                } else if (speed>60){ //an upervei ta 60 portokali marker
                    inner=inner+' icon=\{"iconSize": [20, 20], "iconUrl":
"images/marker_bus_orange.png"}\';
                    speedlimitcount++; //metraei paraviassi taxutitas
                } else{ //gia taxutita 0-20 kokkino marker
                    inner=inner+' icon=\{"iconSize": [20, 20], "iconUrl":
"images/marker_bus_red.png"}\';
                    inner=inner+'> </leaflet-marker>'; }
                inner=inner+'</leaflet-map>';
                document.getElementById("mapa").innerHTML=inner;
                distsum=Math.round(distsum)/1000; //metatropi se km
                stoptime=Math.floor(stoptime/60)+':'+stoptime % 60;
                var averagespeed=0; //upologismos mesis taxutitas (sunolo/metriseis)
                if (speedcount>0){ averagespeed=(Math.round(speedsum*10/speedcount)/10);}
                var end = moment(endtime, 'HH:mm:ss'); //metatropi oras enarxis kai lxis se moments
                var start = moment(starttime, 'HH:mm:ss');
                var dur=moment.duration(end.diff(start)); //upologismos diarkeias metaxu 2 moments
                dur=dur.get("hours") + ":"+ dur.get("minutes") + ":"+ dur.get("seconds");
                var summary='Average speed: ' + averagespeed + ' km/h. Max speed: ' + maxspeed + '
km/h. Distance: ' + distsum + ' km.'
                + '</br>Start: ' + starttime + '. End: ' + endtime + '. Duration: ' + dur + ' h:m:s.' ;
                if (stops>0)
                    summary+='Stops: ' + (stops) + '. Stopped: ' + stoptime + ' m:s.';
                if (speedlimitcount>0)
                    summary+= ' <br> <font color="red">You had ' + speedlimitcount + ' speedlimit
violations.</font>';
                document.getElementById("summary").innerHTML=summary; }

```