

ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ  
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ  
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ΚΙΝΗΤΗΣ ΣΥΣΚΕΥΗΣ ΚΑΙ ΠΛΗΡΟΦΟΡΙΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ ΓΙΑ ΤΗΝ ΔΙΑΧΕΙΡΙΣΗ ΣΤΟΛΟΥ ΟΧΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

---

**ΓΚΑΓΚΑΡΑΤΖΑΣ ΠΑΝΑΓΙΩΤΗΣ**

**Επιβλέπων:** Λούτα Μαλαματή

Αναπληρώτρια Καθηγήτρια

ΚΟΖΑΝΗ / ΙΟΥΝΙΟΣ / 2023





HELLENIC DEMOCRACY  
UNIVERSITY OF WESTERN MACEDONIA  
SCHOOL OF ENGINEERING  
DEPARTMENT OF ELECTRICAL  
& COMPUTER ENGINEERING

# MOBILE APPLICATION AND INFORMATION SYSTEM FOR VEHICLES MANAGEMENT

THESIS

---

**PANAGIOTIS GKAGKARATZAS**

**SUPERVISOR:** Louta Malamati

Associate Professor

KOZANI / JUNE / 2023





ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ  
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ  
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

## ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα Διπλωματική Εργασία με τίτλο “Ανάπτυξη εφαρμογής κινητής και πληροφοριακού συστήματος για την διαχείριση στόλου οχημάτων” καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους του Τμήματος κ. Μαλαματή Λούτα αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και μόνο.

Copyright (C) Γκαγκαράτζας Παναγιώτης, Μαλαματή Λούτα, 2023, Κοζάνη

Υπογραφή Φοιτητή:



# Περίληψη

Η παρούσα πτυχιακή εργασία αναφέρεται στην δημιουργία ενός **πλοηγού** για smartphones με λειτουργικό σύστημα Android. Εκτός από την εφαρμογή, έχει υλοποιηθεί και ένα **πληροφοριακό σύστημα**. Το σύστημα αυτό αποτελείται από μια non relational βάση δεδομένων MongoDB, λογισμικό πακέτο για την επεξεργασία, ανάλυση και διαχείριση των πληροφοριών, λογισμικό υπηρεσιών ιστού (web services) μέσω του οποίου θα πραγματοποιείται η διασύνδεση της εφαρμογής του smartphone με τη βάση δεδομένων, μια εφαρμογή ιστού (web application) και ενός δικτυακού τόπου (website) για την οπτικοποίηση της πληροφορίας. Η εφαρμογή θα λαμβάνει από τον χρήστη όλα τα σημεία (στάσεις) από τα οποία θέλει ο ίδιος να περάσει με το όχημά του. Στη συνέχεια θα στέλνει στο πληροφοριακό σύστημα τα δεδομένα που έχει συλλέξει όπως την ταχύτητα, τις συντεταγμένες, την κατεύθυνση του οχήματος και τις στάσεις που όρισε ο χρήστης. Το πληροφοριακό σύστημα θα αξιολογεί τις πληροφορίες που έλαβε από την εφαρμογή και θα ορίζει την βέλτιστη διαδρομή. Επιπλέον ο χρήστης μέσα από την εφαρμογή μπορεί να αποκτήσει πρόσβαση στις προσωπικές του πληροφορίες, στις πληροφορίες του οχήματός του καθώς και στο ιστορικό των διαδρομών που έχει ήδη πραγματοποιήσει. Πέρα από την εφαρμογή του κινητού, μέσω του website, έχει την δυνατότητα να αλλάξει τα στοιχεία του προφίλ του ή των οχημάτων που έχει στη διάθεσή του και να δει σε πραγματικό χρόνο αν βρίσκεται εν κινήσει κάποιο δρομολόγιο. Ακόμη μπορεί να δει την ώρα αναχώρησης και άφιξης στον προορισμό του, να ανοίξει σε χάρτη την διαδρομή με όλες τις στάσεις από τις οποίες πέρασε το όχημα. Στόχος της εργασίας είναι ανάδειξη της πολύ μεγάλης εξέλιξης των πληροφοριακών συστημάτων αλλά και της χρησιμότητας των πλοηγών στα smartphones. Μέσα από αυτή την εργασία θα ήθελα να παρουσιάσω μια εφαρμογή που θα μπορούσε να εξυπηρετήσει ένα μεγάλο μέρος των χρηστών που χρειάζονται και χρησιμοποιούν καθημερινά navigators για διαδοχικούς προορισμούς χωρίς να χρειάζεται να ολοκληρωθεί μια διαδρομή για να εμφανίσει την επόμενη.

**Λέξεις Κλειδιά :** android, διακομιστής, πλοηγός, βάση δεδομένων

---





# Abstract

This thesis refers to the creation of a navigator for smartphones with the Android operating system. In addition to the application, an information system has also been implemented. This system consists of a non-relational MongoDB database, a software package for the processing, analysis and management of information, web services through which the smartphone application will be connected to the database, a web application and a website for information visualization. The application will receive from the user all the points (stops) through which he wants to pass with his vehicle. It will then send to the information system the data it has collected such as the speed, coordinates, direction of the vehicle and the stops defined by the user. The information system will evaluate the information received from the application and determine the optimal route. In addition, the user through the application can gain access to his personal information, his vehicle information as well as the history of the routes he has already taken. In addition to the mobile application, through the website, he has the possibility to change the details of his profile or the vehicles available to him, to see in real time if a route is in motion. He can also see the time of departure and arrival at his destination, open the route on a map with all the stops the vehicle passed through. The aim of the work is to highlight the very big evolution of information systems and the usefulness of navigators on smartphones. Through this work I would like to present an application that could serve a large part of users who need and use daily navigators for consecutive destinations without having to complete a route to display the next one.

**Keywords :** android, web server, navigator, database

---



# Ευχαριστίες

Θα ήθελα να ευχαριστήσω την Δρ Λούτα Μαλαματή και τον κ. Δημόκα Νίκο για την δυνατότητα που μου έδωσαν να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα αλλά και την πολύτιμη βοήθεια που μου προσέφεραν καθ' όλη τη διάρκεια εκπόνησης της πτυχιακής εργασίας.

Επίσης, θα ήθελα να ευχαριστήσω τους φίλους μου για τις αμέτρητες στιγμές χαράς που ζήσαμε και για την έμπρακτη υποστήριξη τους όποτε χρειάστηκε.

Τέλος δεν θα μπορούσα να παραλείψω την οικογένειά μου που μου στάθηκαν όλα αυτά τα χρόνια και να τους ευχαριστήσω μέσα από την καρδιά μου για την τεράστια προσφορά τους.



# Περιεχόμενα

---

<b>Περίληψη</b>	<b>1</b>
<b>Abstract</b>	<b>3</b>
<b>Ευχαριστίες</b>	<b>5</b>
<b>Περιεχόμενα</b>	<b>7</b>
<b>Κατάλογος εικόνων</b>	<b>9</b>
<b>Κεφάλαιο 1: Εισαγωγή</b>	<b>12</b>
1.1 Πρόλογος	12
1.2 Δομή εργασίας	12
<b>Κεφάλαιο 2: Βασικές έννοιες</b>	<b>15</b>
2.1 Web Server	15
2.1.1 Τι είναι ο web server	15
2.1.2 Πως δουλεύει ο web server	16
2.1.3 Χαρακτηριστικά ενός web server	17
2.1.4 Κατηγορίες web server	17
2.2 Βάση δεδομένων	17
2.2.1 Τι είναι η βάση δεδομένων	18
2.2.2 Από τι αποτελείται μια βάση δεδομένων	18
2.2.3 Κατηγορίες βάσεων δεδομένων	19
2.3 Λειτουργικό σύστημα Android	20
2.3.1 Η έννοια του λειτουργικού συστήματος	20
2.3.2 Τι είναι το Android	21
2.3.3 Συσκευές με λειτουργικό σύστημα Android	23
2.3.4 Χαρακτηριστικά του Android	23
2.4 Παγκόσμιο Σύστημα Εντοπισμού - GPS	25
2.4.1 Η έννοια του παγκόσμιου συστήματος εντοπισμού	25
2.4.2 Εντοπισμός τηλεφώνου μέσω GPS	26
2.4.3 Εντοπισμός τηλεφώνου μέσω δικτύου κινητής τηλεφωνίας	27
<b>Κεφάλαιο 3: Παρόμοιες εφαρμογές</b>	<b>29</b>
3.1 Google Maps	29
3.2 Map Quest	31
3.3 Circuit Route Planner	32
3.4 Upper Route Planner	33
<b>Κεφάλαιο 4: Υλοποίηση εφαρμογής</b>	<b>36</b>
4.1 Εισαγωγή	36
4.1.1 API	36
4.1.2 Web Service	36
4.1.3 REST	37

4.1.4 JSON	39
4.1.5 JQuery	41
4.1.6 AJAX	41
4.1.7 Retrofit	41
4.1.8 NodeJS	42
4.1.9 MongoDB	42
4.2 Υλοποίηση Web server	42
4.3 Υλοποίηση Android εφαρμογής	48
4.4 Υλοποίηση Web εφαρμογής	54
<b>Κεφάλαιο 5: Οδηγός Χρήσης του Android app</b>	<b>63</b>
5.1 Εισαγωγή	63
5.2 Είσοδος χρήστη	63
5.3 Εγγραφή χρήστη	71
5.3.1 Δημιουργία προφίλ οδηγού	71
5.3.2 Δημιουργία προφίλ οχήματος	73
5.4 Αρχική σελίδα	74
5.4.1 Πληροφορίες λογαριασμού	80
5.4.2 Πληροφορίες οχήματος	82
5.4.3 Πληροφορίες διαδρομών	84
5.5 Πλοήγηση	85
<b>Κεφάλαιο 6: Οδηγός Χρήσης του Web app</b>	<b>92</b>
6.1 Εισαγωγή	92
6.2 Είσοδος χρήστη	92
6.3 Εγγραφή χρήστη	94
6.3.1 Δημιουργία προφίλ οδηγού	95
6.3.2 Δημιουργία προφίλ οχήματος	96
6.4 Αρχική σελίδα διαχειριστή	98
6.4.1 Πληροφορίες λογαριασμού	99
6.4.2 Πληροφορίες οχήματος	103
6.4.3 Πληροφορίες διαδρομών	108
6.4.4 Πληροφορίες συνόλου χρηστών	110
6.4.5 Πληροφορίες συνόλου οχημάτων	113
6.4.6 Πληροφορίες συνόλου διαδρομών	116
6.4 Σελίδα χρήστη	118
6.5 Χάρτης διαδρομών	120
<b>Κεφάλαιο 7: Επίλογος</b>	<b>123</b>
7.1 Συμπεράσματα	123
7.2 Μελλοντικές Επεκτάσεις	123

# Κατάλογος εικόνων

---

Εικόνα 1: Μοντέλο Πελάτη-Εξυπηρετητή	16
Εικόνα 2: Ιεραρχικά επίπεδα ενός λειτουργικού συστήματος	20
Εικόνα 3: Το πρώτο smartphone με λειτουργικό σύστημα android	21
Εικόνα 4: Μέθοδος της τριπλοποίησης (trilateration)	26
Εικόνα 5: Google Maps	29
Εικόνα 6: Map Quest	30
Εικόνα 7: Circuit Route Planner	31
Εικόνα 8: Upper Route Planner	33
Εικόνα 9: Είσοδος χρήστη στο Android App	64
Εικόνα 10: Μήνυμα σφάλματος μετά την εισαγωγή λανθασμένων στοιχείων σύνδεσης για 4η φορά στο Android app	66
Εικόνα 11: Αντίστροφη μέτρηση για την επανενεργοποίηση του κουμπιού εισόδου στο Android App	67
Εικόνα 12: Απόκρυψη κωδικού πρόσβασης κατά την είσοδο στο Android App	68
Εικόνα 13: Εμφάνιση κωδικού πρόσβασης κατά την είσοδο στο Android App	68
Εικόνα 14: Ανάκτηση κωδικού πρόσβασης στο Android App	69
Εικόνα 15: Αγγλική έκδοση του Android App	70
Εικόνα 16: Εγγραφή χρήστη στο Android app	72
Εικόνα 17: Εγγραφή οχήματος στο Android app	74
Εικόνα 18: Επιλογή οχήματος στο Android app	75
Εικόνα 19: Αρχική σελίδα του Android app	76
Εικόνα 20: Ενεργοποίηση εντοπισμού συσκευής στο Android App	77
Εικόνα 21: Σφάλμα ανενεργού διακόπτη εντοπισμού στο Android App	78
Εικόνα 22: Μήνυμα αναμονής για τον εντοπισμό της συσκευής στο Android App	78
Εικόνα 23: Dropdown menu της αρχικής οθόνης στο Android app	79
Εικόνα 24: Σελίδα στοιχείων χρήστη στο Android app	80
Εικόνα 25: Αλλαγή κωδικού πρόσβασης από την σελίδα χρήστη στο Android app	81
Εικόνα 26: Σελίδα στοιχείων οχήματος του χρήστη στο Android app	83
Εικόνα 27: Αλλαγή οχήματος δρομολογίων από την σελίδα οχήματος στο Android app	84
Εικόνα 28: Σελίδα δρομολογίων του χρήστη στο Android app	85
Εικόνα 29: Μήνυμα εισόδου στην σελίδα του χάρτη στο Android App	86
Εικόνα 30: Σελίδα του χάρτη στο Android app	87
Εικόνα 31: Αναζήτηση οδού στο Android app	88
Εικόνα 32: Σχεδίαση διαδρομής στο Android app	88
Εικόνα 33: Διαγραφή μιας διεύθυνσης από την λίστα στάσεων στο Android app	89
Εικόνα 34: Πλοήγηση στο Android app	90
Εικόνα 35: Είσοδος χρήστη στο Web app	93
Εικόνα 36: Ανάκτηση κωδικού πρόσβασης στο Web app	94
Εικόνα 37: Εγγραφή χρήστη στο Web app	96
Εικόνα 38: Εγγραφή οχήματος στο Web app	97
Εικόνα 39: Αρχική σελίδα διαχειριστή στο Web app	98
Εικόνα 40: Dropdown menu της αρχικής οθόνης διαχειριστή στο Web app	99
Εικόνα 41: Σελίδα στοιχείων χρήστη στο Web app	100
Εικόνα 42: Dropdown menu της σελίδας χρήστη στο Web app	101
Εικόνα 43: Επεξεργασία στοιχείων χρήστη στο Web app	102
Εικόνα 44: Αλλαγή κωδικού πρόσβασης από την σελίδα χρήστη στο Web app	103
Εικόνα 45: Σελίδα στοιχείων οχημάτων ενός χρήστη στο Web App	104
Εικόνα 46: Επεξεργασία στοιχείων οχήματος στο Web app	105
Εικόνα 47: Αναζήτηση ενός οχήματος στο Web App	106

Εικόνα 48: Προσθήκη νέου οχήματος στο Web App	107
Εικόνα 49: Σελίδα δρομολογίων ενός χρήστη στο Web app	109
Εικόνα 50: Εύρεση δρομολογίου ενός χρήστη στο Web app	110
Εικόνα 51: Σελίδα χρηστών του συστήματος στο Web app	111
Εικόνα 52: Επεξεργασία στοιχείων χρήστη από την σελίδα του διαχειριστή στο Web app	112
Εικόνα 53: Αναζήτηση ενός χρήστη από την σελίδα διαχειριστή στο Web app	113
Εικόνα 54: Σελίδα συνόλου οχημάτων του συστήματος στο Web app	114
Εικόνα 55: Επεξεργασία στοιχείων οχήματος από την σελίδα διαχειριστή στο Web app	115
Εικόνα 56: Αναζήτηση ενός οχήματος από την σελίδα διαχειριστή στο Web app	116
Εικόνα 57: Σελίδα συνόλου δρομολογίων του συστήματος στο Web app	117
Εικόνα 58: Αναζήτηση ενός δρομολογίου από την σελίδα διαχειριστή στο Web App	118
Εικόνα 59: Αρχική σελίδα χρήστη στο Web app	119
Εικόνα 60: Dropdown menu της αρχικής οθόνης χρήστη στο Web app	120
Εικόνα 61: Χάρτης διαδρομής στο Web app	121





# Κεφάλαιο 1: Εισαγωγή

---

## 1.1 Πρόλογος

Η ραγδαία εξέλιξη της τεχνολογίας στον χώρο των κινητών τηλεφώνων (smartphones) έχει αλλάξει την ζωή μας σε πολλούς τομείς. Ένας τομέας που έχει βοηθήσει ιδιαίτερα ανθρώπους που κάνουν ταξίδια, είτε ως χόμπι είτε ως δουλειά, είναι οι εφαρμογές πλοήγησης (navigators). Με την εισαγωγή μιας διεύθυνσης, το smartphone χρησιμοποιώντας υπηρεσίες τοποθεσίας (GPS) και δεδομένα κινητής τηλεφωνίας μπορεί να εντοπίσει την τρέχουσα τοποθεσία του χρήστη και να ορίσει μια διαδρομή για την τοποθεσία που ο ίδιος έχει επιλέξει απαλλάσσοντας τον από τον κόπο εύρεσης της διαδρομής με παραδοσιακές μεθόδους (χάρτες). Έτσι λοιπόν το κέρδος είναι διπλό καθώς ο πλοηγός μας γλιτώνει από περιττή σπατάλη χρόνου και χρημάτων, σε περίπτωση εργασίας, καθώς η διαδρομή που θα μας υποδείξει είναι έγκυρη με μικρές πιθανότητες λάθους.

## 1.2 Δομή εργασίας

Η παρούσα εργασία διαχωρίζεται σε 4 κεφάλαια.

- Στο **δεύτερο** κεφάλαιο γίνεται μια επεξήγηση των τεχνολογιών που συμμετέχουν στην εργασία όπως ο web server, η βάση δεδομένων, το λειτουργικό σύστημα android και το GPS. Συγκεκριμένα θα αναλύσουμε τι είναι το καθένα από αυτά, τα χαρακτηριστικά τους και πως δουλεύουν.
- Στο **τρίτο** κεφάλαιο γίνεται μια αναφορά σε σχετικές υλοποιήσεις και περιληπτική περιγραφή για το πως δουλεύει η κάθε μια από αυτές. Μην ξεχνάμε ότι η πρώτη συσκευή κινητής τηλεφωνίας με ενσωματωμένο GPS κατασκευάστηκε το 1999 (Benefon), το 2001 ξεκίνησε η εισαγωγή τους σε πλοηγούς αυτοκινήτων ενώ η Qualcomm που γνωρίζουμε μέχρι και σήμερα ανέπτυξε το δικό της σύστημα GPS το 2004. Όπως καταλαβαίνουμε, πρόκειται για έναν τομέα με ιστορικό τουλάχιστον 20 ετών συνεπώς είναι λογικό να υπάρχουν ήδη παρόμοιες εφαρμογές τόσο σε Android κινητά όσο και σε iOS.
- Στο **τέταρτο** κεφάλαιο γίνεται αφορά στο προγραμματιστικό κομμάτι της εφαρμογής. Πιο αναλυτικά, περιγράφεται η δομή του web server, στα web services μέσω των οποίων επικοινωνεί η εφαρμογή με την βάση δεδομένων και το σχήμα της βάσης δεδομένων. Από την πλευρά του Android, αναλύονται οι μέθοδοι για την ανταλλαγή δεδομένων με τον server, τι μορφή έχουν τα δεδομένα που δέχεται και λαμβάνει και γενικά πως έχει υλοποιηθεί. Μια αντίστοιχη περιγραφή πραγματοποιείται και για την υλοποίηση του Web app.
- Στο **πέμπτο** κεφάλαιο γίνεται αναλυτική παρουσίαση της Android εφαρμογής. Συγκεκριμένα γίνεται μια αναφορά στο λογισμικό που επιλέχθηκε για την ανάπτυξη της εφαρμογής, τις γλώσσες προγραμματισμού για την πραγματοποίηση τόσο των μεθόδων για την επικοινωνία με τον server όσο και του κομματιού της εμφάνισης

(user interface). Τέλος, για την κατανόηση χρήσης της εφαρμογής έχει υλοποιηθεί σχετικό εγχειρίδιο.

- Στο **έκτο** κεφάλαιο γίνεται αναλυτική παρουσίαση του web application. Με άλλα λόγια περιγράφεται και εδώ το λογισμικό ανάπτυξης του app, οι γλώσσες προγραμματισμού για το front-end και το back-end. Στη συνέχεια ακολουθεί το αναλυτικό εγχειρίδιο χρήσης της εφαρμογής.



# Κεφάλαιο 2: Βασικές έννοιες

---

## 2.1 Web Server

### 2.1.1 Τι είναι ο web server

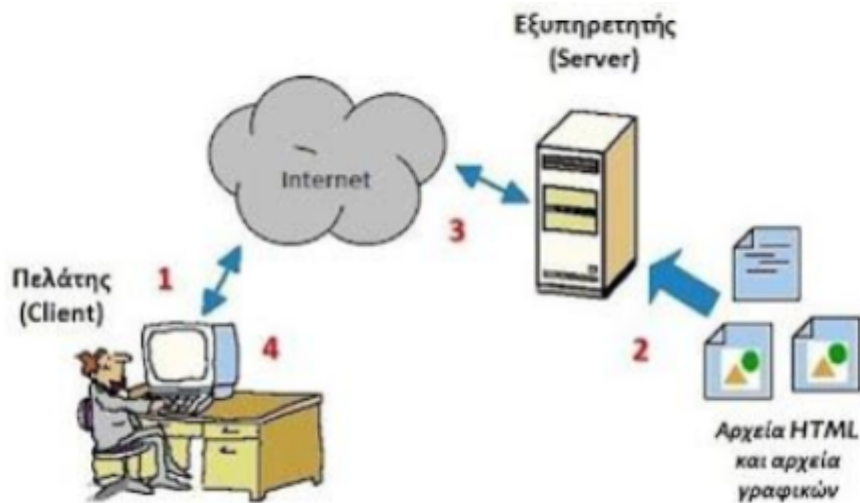
Λίγο πολύ η έννοια του web server είναι γνωστή στις μέρες μας λόγω της τεράστιας προσφοράς στους στον χώρο του διαδικτύου. Με απλά λόγια ο web server είναι ένας υπολογιστής στον οποίο φιλοξενούνται οι ιστοσελίδες. Βασικές εργασίες ενός web server είναι να αποθηκεύει, να επεξεργάζεται και να ανταποκρίνεται στα αιτήματα (requests) των χρηστών που επιθυμούν να αποκτήσουν πρόσβαση σε μια σελίδα [1]. Αν θέλουμε να είμαστε πιο συγκεκριμένοι, ο web server αποτελείται από το **hardware** κομμάτι και το **software** κομμάτι. [2]

- Όταν λέμε hardware εννοούμε έναν υπολογιστή που έχει πρόσβαση στο internet για να μπορέσουν οι υπόλοιποι υπολογιστές να συνδεθούν σε αυτόν και στον οποίο είναι αποθηκευμένα όλα τα αρχεία (πχ HTML, CSS, JavaScript, αρχεία εικόνων) που χρειάζεται μια ιστοσελίδα για να είναι λειτουργική.
- Όταν λέμε software εννοούμε το πως οι χρήστες θα μπορέσουν να αποκτήσουν πρόσβαση στα αρχεία του server. Την δουλειά αυτή την αναλαμβάνει ο HTTP (*Hypertext Transfer Protocol*) server ο οποίος αναγνωρίζει HTTP requests και URLs (διευθύνσεις ιστοσελίδων).

### 2.1.2 Πως δουλεύει ο web server

Για να κατανοήσουμε την χρησιμότητα του web server ας δούμε πως ακριβώς δουλεύει. Γενικά ένας web server λειτουργεί με βάση το μοντέλο πελάτη-εξυπηρετητή (**client-server**).

1. Όταν ένας πελάτης εισάγει στον browser το URL μιας σελίδας, ο browser μέσω ενός μηχανισμού (Domain Name System – DNS) μεταφράζει το URL στην αντίστοιχη διεύθυνση IP με σκοπό να επικοινωνήσει με τον server της σελίδας.
2. Ο browser στέλνει στον server ένα HTTP request με τα αρχεία που επιθυμεί να λάβει από εκείνον για να αποκτήσει πρόσβαση στην σελίδα.
3. Μέσω του HTTP server που αναφέραμε πιο πάνω, ο web server λαμβάνει το request και το επεξεργάζεται.
4. Τελικά ο browser λαμβάνει από τον server τα αρχεία που ζήτησε με σκοπό την εμφάνιση της σελίδας.



Εικόνα 1: Μοντέλο Πελάτη-Εξυπηρετητή

(Πηγή: [https://sites.google.com/site/yperesiesdiadiktyou123/\\_/rsrc/1483740585579/to-montelo-tou-pelate--exyperetete/client-server.jpg?height=220&width=400](https://sites.google.com/site/yperesiesdiadiktyou123/_/rsrc/1483740585579/to-montelo-tou-pelate--exyperetete/client-server.jpg?height=220&width=400))

Σε περίπτωση που ο HTTP server δεν μπορεί να επεξεργαστεί το request (πχ δεν μπορεί να βρει το αρχείο που ζήτησε ο πελάτης) επιστρέφει στον browser μήνυμα σφάλματος.

### 2.1.3 Χαρακτηριστικά ενός web server

Στο σημείο αυτό αξίζει να αναφέρουμε μερικά από τα βασικά χαρακτηριστικά ενός web server.

- **Γλώσσα προγραμματισμού:** Αναφέρεται στην γλώσσα που αναπτύχθηκαν τα προγράμματα τα οποία θα εκτελεστούν από τον server. Μερικές από τις πιο γνωστές γλώσσες προγραμματισμού αυτής της κατηγορίας είναι η Python, η PHP και η NodeJS που επιλέχθηκε για την ανάπτυξη του server στην εργασία μας.
- **Έλεγχος ταυτότητας:** Ένα από τα πιο βασικά χαρακτηριστικά ενός web server είναι η δυνατότητα να ορίζει ποιος χρήστης έχει το δικαίωμα πρόσβασης στα αρχεία του αποτρέποντας μη εξουσιοδοτημένους χρήστες από την παράνομη είσοδο και κλοπή δεδομένων. Η πιο γνωστή μέθοδος αυθεντικοποίησης είναι η καταχώρηση ονόματος χρήστη (username) και κωδικού πρόσβασης (password).
- **Αποθηκευτικός χώρος:** Ορίζεται ως το μέγεθος του χώρου στον οποίο μπορούν να αποθηκευτούν τα αρχεία του server.
- **Καταγραφή δραστηριότητας:** Σε κάθε server υπάρχουν αρχεία τα οποία χωρίζονται σε κατηγορίες και καταγράφουν με βάση την κατηγορία τους την εκάστοτε δραστηριότητα. Για παράδειγμα σε περίπτωση που προκύψει

κάποιο σφάλμα στον server, το συμβάν θα καταγραφεί στο αρχείο σφαλμάτων.

## 2.1.4 Κατηγορίες web server

- Τοπικός web server

Ένας τοπικός (*local*) server είναι ένα σύστημα το οποίο δουλεύει σε ένα τοπικό δίκτυο και οι σελίδες που φιλοξενεί είναι προσβάσιμες μόνο από τους χρήστες που είναι συνδεδεμένοι στο δίκτυο αυτό. Συνήθως χρησιμοποιείται από τους προγραμματιστές που αναπτύσσουν ιστοσελίδες για να δοκιμάσουν την λειτουργικότητα τους.

- Απομακρυσμένος web server

Ένας απομακρυσμένος (*remote*) server είναι ένα σύστημα με πολύ καλύτερα τεχνικά χαρακτηριστικά από έναν τοπικό server διότι έχει την δυνατότητα μέσω του internet να εξυπηρετήσει πολύ μεγαλύτερο αριθμό χρηστών. Ανεξάρτητα από την τοποθεσία του, με την προϋπόθεση ότι είναι συνδεδεμένος στο internet, μπορεί να εξυπηρετήσει χρήστες από ολόκληρο τον κόσμο.

## 2.2 Βάση δεδομένων

### 2.2.1 Τι είναι η βάση δεδομένων

Όπως αναφέραμε στο κεφάλαιο του web server, μια από τις βασικές λειτουργίες του είναι να αποθηκεύει δεδομένα. Αυτή είναι και η χρησιμότητα της βάσης δεδομένων. Γενικά ως βάση δεδομένων ορίζουμε ένα σύνολο από δεδομένα που αποθηκεύονται με συγκεκριμένο τρόπο σε ένα υπολογιστικό σύστημα με σκοπό την εύκολη πρόσβαση και διαχείρισή τους. Συνήθως τα δεδομένα που φτάνουν σε μια βάση δεδομένων αποθηκεύονται σε πίνακες που αποτελούνται από *γραμμές* και *στήλες*. Στο κομμάτι του software, η βάση δεδομένων χρειάζεται ένα πρόγραμμα το οποίο θα διαχειρίζεται τα δεδομένα. Την εργασία αυτή αναλαμβάνει το **σύστημα διαχείρισης βάσης δεδομένων** (Database Management System – DBMS) [3]. Μερικές από τις πιο χαρακτηριστικές αρμοδιότητες αυτού του συστήματος είναι: [4]

- Η αποθήκευση, η ενημέρωση και η διαγραφή δεδομένων.
- Η δημιουργία καινούργιων βάσεων δεδομένων.
- Η δημιουργία καινούργιων πινάκων.
- Η ανάπτυξη αντιγράφων ασφαλείας των δεδομένων.
- Η καταγραφή των δραστηριοτήτων της βάσης δεδομένων.
- Η παράδοση στον χρήστη των δεδομένων που ζήτησε.

- Ο καθορισμός των δικαιωμάτων που θα έχει ο κάθε χρήστης στην βάση δεδομένων.

## 2.2.2 Από τι αποτελείται μια βάση δεδομένων

Σε αυτή την υποενότητα θα επισημάνουμε από τι αποτελείται μια βάση δεδομένων. [5]

- **Δεδομένα:** Οποιαδήποτε πληροφορία στον τομέα των υπολογιστών ορίζεται ως δεδομένο. Τα δεδομένα μπορεί να έχουν μορφή κειμένου, ήχου, εικόνας κλπ.
- **Γλώσσα προγραμματισμού:** Όταν θέλουμε να επικοινωνήσουμε με την βάση δεδομένων, στέλνουμε τα λεγόμενα ερωτήματα (*queries*) με σκοπό να λάβουμε ή να στείλουμε δεδομένα.
- **Λογισμικό:** Το λειτουργικό σύστημα, η εφαρμογή διαχείρισης της βάσης δεδομένων και το πρόγραμμα με το οποίο συνδέεται η βάση δεδομένων στο internet αποτελούν μέρος του πακέτου προγραμμάτων που διαχειρίζονται την βάση.
- **Υπολογιστικό σύστημα:** Εκτός από τις εφαρμογές, το hardware της βάσης δεδομένων απαρτίζεται από τον υπολογιστή που εκτελείται η ίδια, τις συσκευές αποθήκευσης κλπ.

## 2.2.3 Κατηγορίες βάσεων δεδομένων

Ανάλογα με την δομή και τις λειτουργίες τους χωρίζονται στις εξής κατηγορίες:

### ◆ Βάσεις δεδομένων cloud (Cloud databases).

Είναι βάσεις δεδομένων που φιλοξενούνται σε κάποια cloud πλατφόρμα. Το πλεονέκτημα των βάσεων αυτής της κατηγορίας είναι ότι μπορεί με το αντίστοιχο χρηματικό αντίτιμο να μεταβληθεί το μέγεθός τους και η ταχύτητα μεταφοράς δεδομένων. Επιπρόσθετα δεν υπάρχει η ανάγκη ύπαρξης του hardware της βάσης δεδομένων από τον προγραμματιστή καθώς αυτό παρέχεται ήδη από την cloud πλατφόρμα.

### ◆ Κατανεμημένες βάσεις δεδομένων (Distributed databases)

Στις κατανεμημένες βάσεις δεδομένων, η πληροφορία αποθηκεύεται τοπικά στα υπολογιστικά συστήματα. Τα δεδομένα δεν αποθηκεύονται σε ένα μέρος και διανέμονται σε διάφορες σελίδες. Το γεγονός ότι οι κατανεμημένες βάσεις δεδομένων επικοινωνούν μεταξύ τους έχει ως αποτέλεσμα η πληροφορία να εμφανίζεται μια φορά στον χρήστη.



#### ◆ Βάσεις δεδομένων εγγράφων (Document databases)

Τα δεδομένα σε αυτές τις βάσεις δεδομένων αποθηκεύονται σε μορφή εγγράφου, συνήθως σε XML, JSON ή BSON μορφή.

#### ◆ Βάσεις δεδομένων γραφημάτων (Graph databases)

Στις βάσεις δεδομένων γραφημάτων η πληροφορία αποθηκεύεται σε μορφή διαγραμμάτων. Ένα παράδειγμα διαγράμματος είναι οι οντότητες και οι σχέσεις μεταξύ τους.

#### ◆ Βάσεις δεδομένων πολλαπλών μοντέλων (Multi model databases)

Οι βάσεις δεδομένων πολλαπλών μοντέλων έχουν την δυνατότητα να αποθηκεύουν πολλαπλά μοντέλα δεδομένων ακόμα και όταν τα δεδομένα αυτά δεν είναι όλα του ίδιου τύπου.

#### ◆ Βάσεις δεδομένων NoSQL (NoSQL databases)

Γενικά, οι μη σχεσιακές βάσεις δεδομένων επιλέγονται για μεγάλα σύνολα δεδομένων τα οποία συνήθως δεν έχουν δομή. Οι NoSQL βάσεις δεδομένων αν και είναι πολύ αποτελεσματικές στην ανάλυση δεδομένων μεγάλου μεγέθους αντιμετωπίζουν προβλήματα απόδοσης λόγω του μεγέθους των δεδομένων.

#### ◆ Αντικειμενοστραφείς βάσεις δεδομένων (Object-oriented databases)

Στις αντικειμενοστραφείς βάσεις, τα δεδομένα αποθηκεύονται με την μορφή αντικειμένων όπως γίνεται και στις αντικειμενοστραφείς γλώσσες προγραμματισμού. Μια αντικειμενοστραφής βάση δεδομένων αποτελείται από:

- Αντικείμενα
- Κλάσεις
- Μεθόδους
- Δείκτες

#### ◆ Σχεσιακές βάσεις δεδομένων (Relational databases)

Στις σχεσιακές βάσεις δεδομένων η πληροφορία αποθηκεύεται σε πίνακες. Ο πίνακας αποτελείται από σειρές και στήλες προσφέροντας της δυνατότητα ταξινόμησης των δεδομένων.

#### ◆ Αυτόνομες βάσεις δεδομένων (Self-driving databases)

Αποτελεί τον πιο σύγχρονο τύπο βάσης δεδομένων. Χρησιμοποιούν εξελιγμένες τεχνολογίες όπως μηχανική μάθηση (*machine learning*) για την αυτοματοποίηση βασικών λειτουργιών της βάσης σε θέματα ενημερώσεων, ασφάλειας, αντιγράφων ασφαλείας κλπ.

## 2.3 Λειτουργικό σύστημα Android

### 2.3.1 Η έννοια του λειτουργικού συστήματος

Οποιαδήποτε ηλεκτρονική συσκευή κυκλοφορεί αυτή την στιγμή θα ήταν αδύνατον να λειτουργήσει χωρίς την εγκατάσταση σε αυτή ενός λειτουργικού συστήματος. Θα μπορούσαμε να πούμε ότι το λειτουργικό σύστημα είναι ένα λογισμικό το οποίο με βάση το εκάστοτε hardware του συστήματος που έχει εγκατασταθεί συμβάλλει στην σωστή λειτουργία του. Ένα λειτουργικό σύστημα αποτελείται από 3 επίπεδα όπου κάθε ένα αναλαμβάνει να εκτελέσει συγκεκριμένες εργασίες [6]:

- **Πυρήνας (kernel)**

Όταν ανοίγουμε τον υπολογιστή μας, ο πυρήνας φορτώνεται πρώτος στην κύρια μνήμη και εκτελείται όσο έχουμε το σύστημα μας σε λειτουργία. Ο πυρήνας αποτελεί τον διαμεσολαβητή μεταξύ λογισμικού και υλικού.

- **Σύστημα Αρχείων (File System)**

Διαχειρίζεται τα αρχεία του συστήματος με σκοπό την παράδοσή τους στον χρήστη.

- **Διερμηνευτής εντολών (Command Interpreter)**

Στο επίπεδο αυτό δημιουργείται το περιβάλλον επικοινωνίας ανάμεσα στον χρήστη και τον υπολογιστή με το γνωστό σε όλους γραφικό περιβάλλον (Graphical User Interface – GUI) ή μέσω περιβάλλοντος γραμμής εντολών (command mode). Σκοπός του είναι να στέλνει στο σύστημα τις εντολές που έδωσε ο χρήστης και να μεταφέρει στον χρήστη τα μηνύματα από το σύστημα.



Εικόνα 2: Ιεραρχικά επίπεδα ενός λειτουργικού συστήματος

(Πηγή: [http://ebooks.edu.gr/ebooks/v/html/8547/2716/Pliroforiki\\_B-Lykeiou\\_html-empl/image/3\\_1\\_fmt.png](http://ebooks.edu.gr/ebooks/v/html/8547/2716/Pliroforiki_B-Lykeiou_html-empl/image/3_1_fmt.png))

Αξίζει να αναφέρουμε μερικά από τα πιο γνωστά λειτουργικά συστήματα.

- Microsoft Windows
- Linux
- Microsoft DOS
- Mac OS
- Android OS

### 2.3.2 Τι είναι το Android

Το Android αποτελεί μέλος της οικογένειας των λειτουργικών συστημάτων για κινητές συσκευές που αναπτύχθηκε από την Google. Είναι ένα λογισμικό ανοιχτού κώδικα (*open source*) που σημαίνει ότι ο κάθε προγραμματιστής έχει την δυνατότητα να αποκτήσει πρόσβαση στον πηγαίο κώδικα (*source code*) και να τον τροποποιήσει σύμφωνα με τις δικές του απαιτήσεις [7]. Η ανάπτυξη του android ξεκίνησε το 2003 από τον Andy Rubin και την ομάδα του. Βασική ιδέα ήταν η ανάπτυξη ενός λειτουργικού συστήματος για ψηφιακές κάμερες ωστόσο στην συνέχεια και αφού αντιλήφθηκαν ότι ο χώρος των καμερών δεν θα ήταν τόσο κερδοφόρος ξεκίνησαν την ενσωμάτωσή τους σε κινητά τηλέφωνα. Στις 23 Σεπτεμβρίου του 2008 η T-Mobile ανακοίνωσε το πρώτο smartphone με λειτουργικό σύστημα android. [8]



Εικόνα 3: Το πρώτο smartphone με λειτουργικό σύστημα android  
(Πηγή: [https://upload.wikimedia.org/wikipedia/commons/thumb/1/18/T-Mobile\\_G1\\_launch\\_event\\_2.jpg/220px-T-Mobile\\_G1\\_launch\\_event\\_2.jpg](https://upload.wikimedia.org/wikipedia/commons/thumb/1/18/T-Mobile_G1_launch_event_2.jpg/220px-T-Mobile_G1_launch_event_2.jpg))

Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε για την ανάπτυξη του android είναι η Java και επιλέγεται ακόμα και σήμερα για την υλοποίηση των εφαρμογών. Οι προγραμματιστές, με σκοπό την ανάπτυξη των εφαρμογών για android χρησιμοποιούν ένα “πακέτο” εργαλείων που ονομάζεται SDK (*Software Development Kit*) [9]. Πιο αναλυτικά, το SDK είναι ένα λογισμικό που χρησιμοποιείται για την ανάπτυξη εφαρμογών για συγκεκριμένη πλατφόρμα. Βασικά εργαλεία που περιλαμβάνει είναι:

- **Οδηγίες χρήσης (Documentation):** Περιλαμβάνει οδηγίες για το πως ακριβώς πρέπει να χρησιμοποιηθεί ο κώδικας αλλά και κομμάτια κώδικα με επεξήγηση για το αποτέλεσμα εκτέλεσής του.
- **Βιβλιοθήκες (Libraries):** Αποτελούνται από αρχεία κώδικα που παίρνουν οι προγραμματιστές και τα ενσωματώνουν στις εφαρμογές που αναπτύσσουν με σκοπό την ορθή εκτέλεσή τους στην ανάλογη πλατφόρμα.
- **Περιβάλλον ανάπτυξης (IDE):** Είναι το περιβάλλον που δουλεύει ο developer την εφαρμογή του. Απαρτίζεται από τον πηγαίο κώδικα, τον debugger ο οποίος εντοπίζει τυχόν λάθη στον κώδικα που θα μπορούσαν να προκαλέσουν σφάλμα στην λειτουργία του λογισμικού και τον compiler που παίρνει τον κώδικα και τον “μετατρέπει” από την γλώσσα προγραμματισμού που υλοποιήθηκε σε γλώσσα μηχανής με σκοπό την εκτέλεσή του.
- **Διεπαφές εφαρμογών (API):** Ορίζεται ως μηχανισμός που ενσωματώνει ο προγραμματιστής στην εφαρμογή του με σκοπό την επικοινωνία της με άλλες εφαρμογές.
- **Δείγματα κώδικα (Code samples):** Είναι συνήθως απλές εφαρμογές που δείχνουν στην πράξη πως δουλεύει ο κώδικας επιτρέποντας τον προγραμματιστή να έχει μια πιο σαφή εικόνα.

### 2.3.3 Συσκευές με λειτουργικό σύστημα Android

Λόγω της ραγδαίας εξέλιξής του, το Android αποτελεί κύριο λειτουργικό σύστημα όχι μόνο των κινητών τηλεφώνων. Το γεγονός ότι είναι λογισμικό ανοιχτού κώδικα επιτρέπει τους προγραμματιστές να τροποποιούν ορισμένες παραμέτρους για τις ανάγκες της συσκευής που επιθυμούν να το ενσωματώσουν. Επιπρόσθετα, η ικανότητα των android συσκευών να επικοινωνούν εύκολα όχι μόνο μεταξύ τους αλλά και με άλλες συσκευές με διαφορετικό λειτουργικό σύστημα τις καθιστά κυρίαρχες στον τομέα της συνδεσιμότητας. Αξίζει λοιπόν να αναφέρουμε μερικές από τις πιο γνωστές συσκευές που “τρέχουν” κάποια android έκδοση. [10]

- Tablet**
- Έξυπνο ρολόι (Smartwatch)**
- Έξυπνη τηλεόραση (Smart TV)**
- Κονσόλες (Gaming consoles)**
- Αυτοκίνητο (Android Auto)**

- Οικιακές συσκευές
- Έξυπνα γυαλιά

### 2.3.4 Χαρακτηριστικά του Android

Ένας από τους βασικούς λόγους που έγινε τόσο δημοφιλής το android αποτελεί το γεγονός ότι είναι πολύ εύκολο στο χειρισμό από τον μέσο χρήστη. Επιπλέον υποστηρίζει έναν μεγάλο αριθμό εφαρμογών προσφέροντας στον κάτοχο της συσκευής την δυνατότητα να διαχειριστεί πιο εύκολα τις καθημερινές εργασίες του χωρίς να καταβάλλει μεγάλο χρηματικό ποσό για να αποκτήσει μια τέτοια συσκευή. Σε αυτή την υποενότητα θα αναφέρουμε τα βασικά χαρακτηριστικά μιας συσκευής με λειτουργικό σύστημα android.[11]

#### □ Επικοινωνία μέσω μηνυμάτων (Messaging)

Κάθε smartphone με android λειτουργικό σύστημα επιτρέπει την επικοινωνία μέσω SMS (Short Message Service) ή MMS (Multimedia Messaging Service).

#### □ Επικοινωνία μέσω βίντεο (Video calling)

Οι συσκευές που έχουν πρόσβαση στο internet είτε μέσω Wi-Fi είτε μέσω δεδομένων κινητής τηλεφωνίας έχουν την ευκαιρία να πραγματοποιούν βιντεοκλήσεις. Μερικές από τις πιο γνωστές εφαρμογές είναι το Google Talk, το Hangout, το Skype, το Messenger, το Instagram κλπ.

#### □ Καταγραφή οθόνης (Screen Capture)

Σε συσκευές με έκδοση android προγενέστερη της 4.0 η καταγραφή της οθόνης ήταν εφικτή μόνο με την χρήση τρίτων εφαρμογών. Πλέον στις σύγχρονες εκδόσεις του λειτουργικού συστήματος προσφέρεται η επιλογή λήψης στιγμιότυπου οθόνης, καταγραφής της οθόνης και καταγραφής του μικροφώνου με εργοστασιακές εφαρμογές.

#### □ Υποστήριξη πολλαπλών γλωσσών (Multiple language support)

Παρέχεται η δυνατότητα υποστήριξης πολλαπλών γλωσσών ανάλογα με τον χρήστη. Με αυτό τον τρόπο οι εφαρμογές “διαβάζουν” την γλώσσα που έχει επιλέξει ο χρήστης και εμφανίζουν ανάλογα το περιεχόμενό τους.

#### □ Φωνητικές λειτουργίες (Voice-based features)

Η Google μέσω της υπηρεσίας Voice Access παρέχει την επιλογή στον χρήστη να δώσει μια φωνητική εντολή στο σύστημα (πχ να πραγματοποιήσει μια κλήση, να στείλει ένα μήνυμα κλπ).

#### □ Δυνατότητα πολλαπλών παράλληλων εργασιών (Multitasking)

Ένα από τα πιο σημαντικά χαρακτηριστικά των smartphones είναι η ικανότητά τους να εκτελούν ταυτόχρονα πολλές εφαρμογές. Με την σωστή διαχείριση των πόρων του συστήματος από το android, ο χρήστης έχει την ευχέρεια να εκτελεί παραπάνω από μια εφαρμογές χωρίς να χρειαστεί να τερματίσει την προηγούμενη.

□ **Κοντινό πεδίο επικοινωνίας (Near Field Communication – NFC)**

Αποτελεί μια ασύρματη τεχνολογία που επιτρέπει την επικοινωνία μεταξύ συσκευών που βρίσκονται σε κοντινές αποστάσεις. Δυο smartphones που ενσωματώνουν αυτή την τεχνολογία μπορούν να τοποθετηθούν σε απόσταση μερικών εκατοστών μεταξύ τους και να ανταλλάξουν αρχεία. Ακόμη, το NFC μπορεί να χρησιμοποιηθεί και για την πραγματοποίηση πληρωμών. [12]

□ **Πολλαπλά πληκτρολόγια (Multiple Keyboards)**

Το android παρέχει την δυνατότητα στους χρήστες του να κατεβάζουν και να χρησιμοποιούν διαφορετικά στυλ πληκτρολογίου από το εργοστασιακό όπως το Swift-Key, το Skype-app σε αντίθεση με άλλα λειτουργικά συστήματα που δεν επιτρέπουν την αλλαγή πληκτρολογίου.

□ **Γραφικά στοιχεία (Widgets)**

Βασικός στόχος των widgets είναι να απαλλάξουν τον χρήστη από τον περιττό χρόνο να ανοίξει μια εφαρμοστέι για να δει αυτό που θέλει. Με τα widgets μπορούμε να ορίσουμε εμείς τι θέλουμε να εμφανίζεται από μια εφαρμογή στην αρχική οθόνη της συσκευής χωρίς να επιβαρύνουμε άλλο το σύστημα προκειμένου να εισέλθουμε σε αυτήν.

□ **Έλεγχος χωρίς αφή (No Touch Control - Gestures)**

Ένας από τους λόγους που έχουμε βαφτίσει τα κινητά τηλέφωνα *έξυπνα* είναι διότι είμαστε σε θέση να αλληλεπιδράσουμε με αυτά χωρίς να έρθουμε άμεσα σε επαφή. Το smartphone μέσω μηχανισμών ανίχνευσης διάφορων χειρονομιών μπορεί να λάβει εντολές από εμάς χωρίς να χρησιμοποιήσουμε την αφή. Με απλά λόγια, μπορούμε να αλλάξουμε ένα τραγούδι που παίζει στη συσκευή, να απορρίψουμε μια εισερχόμενη κλήση ή να τραβήξουμε μια φωτογραφία με μια μόνο χειρονομία. [13]

□ **Αυτοματοποίηση (Automation)**

Σε αυτή την κατηγορία ανήκουν οι εργασίες που ορίζει μια φορά ο χρήστης και εκτελούνται με βάση το πλάνο που έχει ορίσει ο ίδιος. Η ενεργοποίηση του Wi-Fi όταν μπαίνει στο σπίτι, η εφαρμογή του προφίλ “Μην Ενοχλείτε” συγκεκριμένες ώρες της ημέρας, η αυτόματη δημιουργία αντιγράφων ασφαλείας των δεδομένων αποτελούν μερικές αυτοματοποιημένες εργασίες. [14]

□ **Τροποποιημένες εκδόσεις λειτουργικού συστήματος (Custom ROMs)**

Όπως αναφέραμε και στην υποενότητα 2.3.2, το android είναι ένα λογισμικό ανοιχτού κώδικα που επιτρέπει στους προγραμματιστές να το τροποποιούν και να αναπτύσσουν δικές τους εκδόσεις. Οι αλλαγές αυτές μπορεί να αφορούν μόνο το κομμάτι της εμφάνισης ενώ άλλες να εισάγουν καινούργια χαρακτηριστικά καθιστώντας τις εκδόσεις αυτές πιο ελκυστικές από άλλα λειτουργικά συστήματα.

## 2.4 Παγκόσμιο Σύστημα Εντοπισμού - GPS

### 2.4.1 Η έννοια του παγκόσμιου συστήματος εντοπισμού

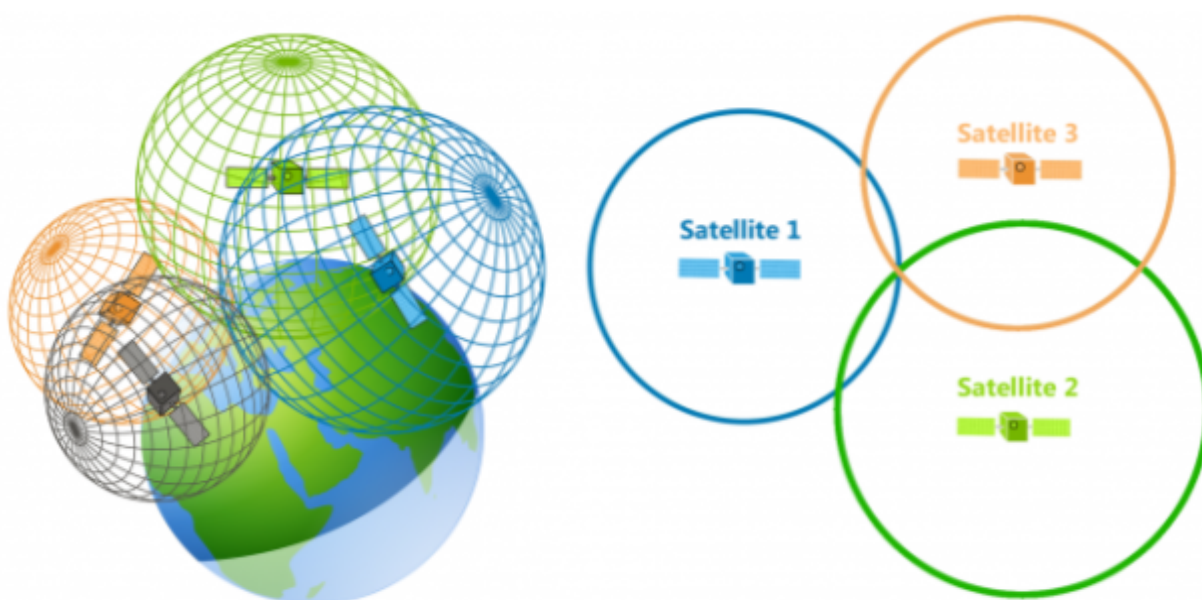
Το Παγκόσμιο Σύστημα Εντοπισμού (*Global Positioning System – GPS*) είναι μια τεχνολογία που αναπτύχθηκε στις ΗΠΑ το 1973. Στόχος του είναι να παρέχει στις συσκευές πληροφορίες που αφορούν ώρα και τοποθεσία. Αρχικά, το GPS εξυπηρετούσε αποκλειστικά το αμερικανικό ναυτικό και αεροπορία. Η ελεύθερη πρόσβαση του από το κοινό ξεκίνησε από την δεκαετία του 1980 [15]. Κύρια αιτία αποτέλεσε το τραγικό περιστατικό της 1ης Σεπτεμβρίου του 1983 όπου ένα επιβατικό αεροσκάφος Boeing 747 της Κορεάτικης αεροπορίας ύστερα από λανθασμένες εκτιμήσεις του πληρώματος σχετικά με την κατεύθυνση του σκάφους βρέθηκε πάνω από εναέριο χώρο όπου η Σοβιετική Ένωση είχε θέσει ως απαγορευμένο. Το αποτέλεσμα ήταν η κατάρριψη του αεροσκάφους από ένα Sukhoi Su-15 της πολεμικής αεροπορίας των Σοβιετικών [16]. Μέχρι το 1995 η αμυντική βιομηχανία των ΗΠΑ είχε αναπτύξει ένα σύστημα που χρησιμοποιούσε 24 δορυφόρους. Έκτος από την Αμερική, συστήματα εντοπισμού αναπτύχθηκαν και σε άλλες χώρες. Η Ρωσία ανέπτυξε το δικό της σύστημα (*GLONASS*) το 1982 [17], η Ιαπωνία (*Quasi-Zenith*) το 2010 [18], η Ευρώπη (*Galileo*) το 2011 [19] και η Ινδία (*IRNSS*) το 2013 [20]. Πλέον τουλάχιστον 24 δορυφόροι σε τροχιά γύρω από την γη λειτουργούν σε ύψος το λιγότερο 20.000 χιλιομέτρων για να καλύψουν τις ανάγκες των κρατών ανά τον πλανήτη. Το GPS λειτουργεί κάτω από οποιοσδήποτε καιρικές συνθήκες με βασική προϋπόθεση να υπάρχει ανεμπόδιση οπτική επαφή με τουλάχιστον 4 δορυφόρους [21]. Οι 3 δορυφόροι δημιουργούν ένα τρίγωνο για να εντοπίσουν με όσο το δυνατόν μεγαλύτερη ακρίβεια την τοποθεσία και ο τέταρτος για τον προσδιορισμό του υψομέτρου.

Το GPS προκειμένου να εντοπίσει μια τοποθεσία χρησιμοποιεί ραδιοκύματα μεταξύ των δορυφόρων και της συσκευής. Οι συσκευές δεν αποστέλλουν δεδομένα στον δορυφόρο παρά μόνο το αντίθετο. Ένα από τα κυριότερα μειονεκτήματα του ειδικά στα κινητά τηλέφωνα είναι το γεγονός ότι καταναλώνουν αρκετή ισχύ. Οι χρήστες συνήθως αιτούνται την τοποθεσία τους όταν είναι στον δρόμο, συνεπώς δημιουργήθηκε η ανάγκη διαμόρφωσης ενός συστήματος εντοπισμού που θα καταναλώνει λιγότερη ενέργεια. Ιδιαίτερα όταν ανάμεσα στην συσκευή και τους δορυφόρους υπάρχουν εμπόδια, ο εντοπισμός του τηλεφώνου γίνεται ακόμα πιο δύσκολος και ενεργοβόρος. Για τον λόγο αυτό αναπτύχθηκε το Υποβοηθούμενο Παγκόσμιο Σύστημα Εντοπισμού (*Assisted Global Positioning System – AGPS*). Το AGPS εκμεταλλεύεται το δίκτυο κινητής τηλεφωνίας της συσκευής για τον γρήγορο εντοπισμό της. Αυτό προκύπτει λόγω της ικανότητας των κεραιών κινητής τηλεφωνίας να εντοπίζουν την τοποθεσία του κινητού τηλεφώνου. [22]

### 2.4.2 Εντοπισμός τηλεφώνου μέσω GPS

Ο εντοπισμός μιας συσκευής μέσω GPS αποτελεί μια απλή και ξεκάθαρη διαδικασία. Όπως προαναφέραμε, τα κινητά τηλέφωνα δεν στέλνουν δεδομένα στον δορυφόρο αλλά μόνο λαμβάνουν από αυτόν. Ο κάθε δορυφόρος έχει το δικό του ρολόι και μεταδίδει πληροφορίες με την μορφή ραδιοκυμάτων ανά τακτικά χρονικά διαστήματα σταθερής περιόδου σε συγκεκριμένες συχνότητες. Αυτές οι

πληροφορίες περιέχουν την θέση του δορυφόρου την συγκεκριμένη χρονική στιγμή και την ώρα μετάδοσης. Για τον επιτυχή προσδιορισμό της τοποθεσίας, η συσκευή θα πρέπει να λάβει πληροφορίες από τουλάχιστον 3 δορυφόρους. Ο χρόνος που απαιτείται για την εύρεσή τους ονομάζεται **Ώρα για Πρώτη Διόρθωση (Time to First Fix – TTFF)** και επηρεάζεται άμεσα από την ύπαρξη εμποδίων που μπορούν να καθυστερήσουν την επικοινωνία με κάποιο δορυφόρο [23]. Στην συνέχεια για να εντοπίσει την απόστασή του από τους δορυφόρους ακολουθεί μια συγκεκριμένη διεργασία. Λαμβάνει τον χρόνο μετάδοσης της πληροφορίας από τον δορυφόρο, τον χρόνο λήψης της από την συσκευή και βρίσκει την διαφορά τους. Μόλις ολοκληρώσει αυτή την διεργασία για τουλάχιστον 3 δορυφόρους περνάει στην δεύτερη φάση που ονομάζεται **τριπλοποίηση (trilateration)**. Σε αυτό το στάδιο, η συσκευή αφού έχει υπολογίσει την απόσταση από τους δορυφόρους δημιουργεί μια νοητή σφαίρα με κέντρο τον πρώτο δορυφόρο. Ακολουθεί την ίδια διαδικασία και για τους υπόλοιπους. Το σημείο τομής των σφαιρών αποτελεί την ακριβή τοποθεσία της συσκευής. [21]



Εικόνα 4: Μέθοδος της τριπλοποίησης (trilateration)  
(Πηγή: <https://gisgeography.com/wp-content/uploads/2016/11/GPS-Trilateration-Feature.png>)

### 2.4.3 Εντοπισμός τηλεφώνου μέσω δικτύου κινητής τηλεφωνίας

Η μεθοδολογία που ακολουθείται για τον εντοπισμό μιας συσκευής μέσω του δικτύου κινητής τηλεφωνίας δεν διαφέρει πολύ από εκείνη για τον εντοπισμό μέσω GPS. Αρχικά θα πρέπει να αντιληφθούμε πως δουλεύει ένα δίκτυο κινητής τηλεφωνίας. Όλα τα δίκτυα κινητής τηλεφωνίας αποτελούνται από **σταθμούς βάσης (base stations)**. Ο σταθμός βάσης είναι οι γνωστές κεραιές που βλέπουμε συνήθως στις κορυφές των βουνών και αποτελούν το κύριο σημείο επικοινωνίας για μια ή περισσότερες συσκευές κινητής τηλεφωνίας. Η επικοινωνία προς και από τον σταθμό βάσης πραγματοποιείται μέσω ραδιοκυμάτων. Η περιοχή κάλυψης ενός σταθμού βάσης ονομάζεται **κυσέλη** και η εμβέλειά της διαφέρει ανάλογα με την



μορφολογία του εδάφους. Τα κινητά τηλέφωνα διαθέτουν πομπούς οι οποίοι επικοινωνούν συνεχώς με τον πλησιέστερο σταθμό βάσης [24]. Το σήμα του τηλεφώνου εξασθενεί όσο πλησιάζει στα όρια μιας κυψέλης και ενισχύεται στο κέντρο της. Όταν απομακρύνεται η συσκευή από τον σταθμό βάσης που επικοινωνεί συντονίζεται με τον επόμενο πλησιέστερο σταθμό με σκοπό να αποφύγει την απώλεια σύνδεσης. Συνεπώς ο εντοπισμός της συσκευής πραγματοποιείται λαμβάνοντας υπ'όψιν την ισχύ του σήματος μεταξύ των σταθμών βάσεων που επικοινωνεί [25].



# Κεφάλαιο 3: Παρόμοιες εφαρμογές

---

Το γεγονός ότι το θέμα της εργασίας αποτελεί αντικείμενο έρευνας και ενασχόλησης αρκετών χιλιάδων προγραμματιστών περίπου 15 χρόνια τώρα μας οδηγεί στο ασφαλές συμπέρασμα ότι έχουν υλοποιηθεί παρόμοιες εφαρμογές. Σε αυτό λοιπόν το κεφάλαιο θα κάνουμε μια σύντομη αναφορά στον τρόπο λειτουργίας αυτών, στα πλεονεκτήματα και τα μειονεκτήματα τους έτσι ώστε να γίνει κατανοητό σε ποιους τομείς υπερτερεί και που υστερεί η συγκεκριμένη εφαρμογή. Η επιλογή των παρακάτω εφαρμογών βασίζεται στις ομοιότητες που εμφανίζουν με την εφαρμογή μας.

## 3.1 Google Maps



*Εικόνα 5: Google Maps*

(Πηγή: <https://1000logos.net/wp-content/uploads/2021/05/Google-Maps-logo.png>)

Αποτελεί ίσως την πιο δημοφιλή εφαρμογή πλοήγησης για κινητά τηλέφωνα. Είναι συμβατή τόσο με Android όσο και με iOS συσκευές. Υπάρχει και έκδοση για υπολογιστή. Το Google Maps αρχικά είχε αναπτυχθεί σε C++ από τα αδέρφια Lars and Jens Rasmussen ενώ το 2004 η εταιρία που εργάζονταν εξαγοράστηκε από την Google. Το 2005 κυκλοφόρησε η πρώτη έκδοση του Google Maps ενώ το 2008 η πρώτη έκδοση για Android και iOS [26]. Πλέον με το Google Maps, ο χρήστης μπορεί να αποκτήσει πρόσβαση σε οδικούς χάρτες, δορυφορικές εικόνες, αεροφωτογραφίες, να εντοπίσει σε πραγματικό χρόνο δρόμους με έντονη κυκλοφοριακή συμφόρηση, να λάβει εικόνες 360 μοιρών από δρόμους (Street View). Επιπλέον έχει την ευκαιρία να λάβει οδηγίες για μια συγκεκριμένη κατεύθυνση που έχει ορίσει ο ίδιος με βασική προϋπόθεση τη σύνδεση στο internet και επικοινωνία μέσω GPS με δορυφόρο.

Ένα από τα πιο χρήσιμα εργαλεία του Google Maps αποτελεί το Route Planner. Με το route planner δίνεται η δυνατότητα στον χρήστη να ορίσει το σημείο αναχώρησης, το σημείο τερματισμού και μια λίστα από ενδιάμεσες στάσεις. Στη συνέχεια η εφαρμογή ορίζει

μια διαδρομή που περιλαμβάνει όλες τις στάσεις λαμβάνοντας υπ' όψιν την κυκλοφοριακή συμφόρηση, πιθανές κλειστές οδούς, αυτοκινητόδρομους, διόδους, πορθμεία και ότι άλλα κριτήρια ορίσει ο χρήστης. Επιπλέον σε περίπτωση που η συσκευή κατευθύνεται σε περιοχή που δεν υπάρχει πρόσβαση στο internet, η εφαρμογή μπορεί να πραγματοποιήσει λήψη της περιοχής με σκοπό την ακάθεκτη λειτουργία της πλοήγησης σε αυτήν ακόμη και χωρίς σύνδεση στο διαδίκτυο [27].

Αξίζει να σημειωθεί ότι το Google Maps παρέχει οδηγίες πλοήγησης όχι μόνο για αυτοκίνητα αλλά και για ποδήλατα, μέσα μαζικής μεταφοράς, για περπάτημα, μοτοσικλέτα, και για αεροσκάφος (σε beta μορφή). Τέλος, δεν θα μπορούσαμε να παραλείψουμε την λειτουργία Street View. Μέσω αυτής, ο χρήστης μπορεί να “μπει” μέσα στον χάρτη και συγκεκριμένα μέσα στον δρόμο που επιθυμεί να επισκεφθεί έχοντας ακριβής εικόνα του προορισμού του.

Το Google Maps έρχεται με κάποιους περιορισμούς οι οποίοι το καθιστούν μη αποδοτικό για χρήση από ορισμένους χρήστες. Ένα βασικό μειονέκτημά του είναι ο αριθμός των ενδιάμεσων στάσεων που μπορεί να λάβει. Το ανώτερο όριο στάσεων ανά δρομολόγιο ανέρχεται στις 10. Συνεπώς δεν μπορεί να χρησιμοποιηθεί από ανθρώπους που πραγματοποιούν μεγάλο αριθμό στάσεων μέσα στην μέρα (πχ διανομείς). Επιπρόσθετα σε περίπτωση που ο χρήστης ορίσει ενδιάμεσες στάσεις, η εφαρμογή θα παρουσιάσει μια διαδρομή με βάση την σειρά που έλαβε τις στάσεις. Δεν θα ληφθεί υπόψιν αν επιθυμεί να φτάσει σε έναν προορισμό πιο γρήγορα από κάποιον άλλο ούτε η απόσταση του ίδιου από την κάθε στάση [28]. Συνεπώς σε περίπτωση που κάποια στάση από τις τελευταίες στην λίστα είναι πιο κοντά στον χρήστη σε σχέση με κάποια που βρίσκεται ψηλά στην λίστα του, ο χρόνος που χρειάζεται για να ολοκληρωθεί το δρομολόγιο είναι αρκετά μεγαλύτερος γεγονός που μπορεί να προκαλέσει σοβαρά προβλήματα στον χώρο εργασίας του χρήστη.

## 3.2 Map Quest



Εικόνα 6: Map Quest

(Πηγή:[https://static.wikia.nocookie.net/logopedia/images/0/08/Mapquest\\_logo\\_%28since\\_2010%29.svg/revision/latest?cb=20180723093534](https://static.wikia.nocookie.net/logopedia/images/0/08/Mapquest_logo_%28since_2010%29.svg/revision/latest?cb=20180723093534))

Το Map Quest συνιστά το πρώτο λογισμικό πλοήγησης και σχεδίασης διαδρομής. Ξεκίνησε το 1967 από την Cartographic Services, τμήμα της R.R. Donnelley, μιας εταιρεία εμπορικών εκτυπώσεων. Στόχος της ήταν η εκτύπωση χαρτών του οδικού δικτύου με σκοπό την διανομή τους σε πρατήρια βενζίνης. Το 1994 η Cartographic Services έγινε ανεξάρτητη εταιρεία αλλάζοντας το όνομά της σε GeoSystems Global Corporation ενώ 2 χρόνια αργότερα ανέπτυξε την πρώτη ιστοσελίδα χαρτογράφησης, το MapQuest.com. Το 1999 η εταιρεία πάλι αλλάζει το όνομά της σε MapQuest [29]. Το 2004, παρουσίασε την εφαρμογή Map Quest Find Me για την λειτουργία της εφαρμογής στα κινητά τηλέφωνα που διέθεταν

GPS ενώ το 2007 έγινε διαθέσιμη στο iPhone. Πλέον υπάρχει έκδοση για υπολογιστές, συσκευές με Android και iOS. Παρέχει στον χρήστη χρήσιμες λειτουργίες όπως φωνητικές οδηγίες πλοήγησης, αναζήτηση σημείων ενδιαφέροντος (Point Of Interest – POI), Street View, εντοπισμό οδών με έντονη κυκλοφοριακή κίνηση [30].

Ένα από τα χαρακτηριστικά της εφαρμογής που είναι άξιο αναφοράς είναι το γεγονός ότι παρέχει στον χρήστη, με βάση την διαδρομή και τις στάσεις που έχει ορίσει, την βέλτιστη διαδρομή έχοντας εντοπίσει και υπολογίσει ναρτίτερα τις τιμές από τα πρατήρια βενζίνης από τα οποία θα περάσει το όχημα [28]. Με βάση αυτή την προεργασία προτείνει στον οδηγό την διαδρομή με το μικρότερο κόστος ανεφοδιασμού καυσίμου. Ακόμα, παρέχει μέχρι 26 στάσεις ανά δρομολόγιο στην δωρεάν έκδοση που σημαίνει ότι μπορεί να εξυπηρετήσει ένα μεγάλο μέρος των χρηστών. Ένα επιπλέον θετικό στοιχείο της εφαρμογής είναι τα φίλτρα που παρέχει για την διαμόρφωση μιας διαδρομής. Η αποφυγή διοδίων, πορθμείων, δρόμων στα σύνορα της χώρας, ο εντοπισμός κυκλοφοριακής κίνησης, εστιατορίων, βενζινάδικων, ξενοδοχείων, καταστημάτων περιλαμβάνονται στα φίλτρα.

Από τη άλλη πλευρά, θα πρέπει να αναφέρουμε και τα μειονεκτήματα αυτής της εφαρμογής. Αρχικά, όπως και στο Google Maps, δεν ορίζει μια διαδρομή με βάση την απόστασή της κάθε στάσης από τον ίδιο αλλά με βάση την σειρά των στάσεων που έχει θέσει ο χρήστης. Επίσης δεν παρέχεται το offline mode. Αυτό σημαίνει ότι σε περίπτωση που χαθεί η σύνδεση με το internet δεν μπορεί να συνεχιστεί η πλοήγηση. Επιπλέον, όπως και στις περισσότερες δωρεάν εφαρμογές, εμφανίζονται αναδυόμενες διαφημίσεις κατά την χρήση της. Ιδιαίτερα κατά την διάρκεια της πλοήγησης οι διαφημίσεις αποσπούν την προσοχή του οδηγού και τις καθιστούν αρκετά ενοχλητικές. Τέλος, ορισμένοι οδηγοί έχουν αναφέρει λανθασμένες οδηγίες για τις στάσεις που έχουν θέσει ή για οδηγίες πλοήγησης που έλαβαν από την εφαρμογή.

### 3.3 Circuit Route Planner



Εικόνα 7: Circuit Route Planner

(Πηγή: [https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSJjXY9SVE\\_O6Skc9rEZPllPbNs2W-6z8fDBKA-Z37gt-Q0RwvRjxfJ5VvJY7g\\_BAZdHĀo&usqp=CĀU](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSJjXY9SVE_O6Skc9rEZPllPbNs2W-6z8fDBKA-Z37gt-Q0RwvRjxfJ5VvJY7g_BAZdHĀo&usqp=CĀU))

Το Circuit Route Planner είναι μια εφαρμογή για Android και iOS συσκευές που απευθύνεται τόσο σε επιχειρήσεις όσο και σε ιδιώτες οδηγούς στον τομέα της διανομής. Βασικός στόχος της εφαρμογής είναι η εύρεση της βέλτιστης διαδρομής ύστερα από τον καθορισμό της αφετηρίας, των ενδιάμεσων στάσεων και του σημείου τερματισμού. Χωρίζεται σε δύο εκδόσεις ανάλογα με τα χαρακτηριστικά που προσφέρει η κάθε μια: στην δωρεάν έκδοση που συνήθως είναι αρκετά ικανοποιητική για τις ανάγκες ενός ιδιώτη και στην έκδοση επί πληρωμή για τις επιχειρήσεις όπου οι ανάγκες είναι μεγαλύτερες. Στην έκδοση επί πληρωμή, παρέχεται η δυνατότητα χρήσης της εφαρμογής για τουλάχιστον έναν οδηγό ενώ η δωρεάν έκδοση απευθύνεται αποκλειστικά σε έναν [27]. Μια ακόμη διαφορά των δυο εκδόσεων είναι ο αριθμός των στάσεων για κάθε δρομολόγιο. Στην δωρεάν έκδοση ο οδηγός έχει την δυνατότητα να πραγματοποιήσει όσα δρομολόγια επιθυμεί με μόλις 10 στάσεις για κάθε ένα από αυτά. Στην premium έκδοση ο αριθμός των δρομολογίων είναι επίσης απεριόριστος και ο αριθμός των στάσεων ανέρχεται στις 500 ανά δρομολόγιο. Μερικά ακόμη πλεονεκτήματα της premium έκδοσης είναι η εισαγωγή στάσεων μέσω XLS αρχείου από Android συσκευές και ο εντοπισμός σε πραγματικό χρόνο του οδηγού κατά την διάρκεια ενός δρομολογίου [31].

Ένα από τα μεγαλύτερα πλεονεκτήματα που προσφέρει η συγκεκριμένη εφαρμογή είναι η βελτιστοποίηση διαδρομής. Με βάση τις στάσεις που έχουν οριστεί, ο χρήστης μπορεί να θέσει προτεραιότητες. Αυτό σημαίνει ότι δεν έχει σημασία η σειρά με την οποία έχει εισάγει τις στάσεις αλλά η σειρά με την οποία θέλει να φτάσει σε αυτές. Ένα χαρακτηριστικό παράδειγμα είναι η παράδοση ενός πακέτου σε έναν πελάτη μια συγκεκριμένη χρονική περίοδο. Η εφαρμογή ανεξάρτητα από τις υπόλοιπες στάσεις θα σχεδιάσει μια διαδρομή με αυτή την στάση να προηγείται των υπολοίπων. Επίσης για κάθε στάση, εμφανίζεται στην οθόνη ο χρόνος που απαιτείται για την άφιξη του οδηγού σε αυτήν. Η λειτουργία αυτή βασίζεται στο ιστορικό του οδηγού. Για κάθε δρομολόγιο που πραγματοποιείται από έναν οδηγό, καταγράφεται ο χρόνος που χρειάστηκε για την ολοκλήρωση του. Με βάση λοιπόν αυτή την λειτουργία, ο οδηγός έχει μια ξεκάθαρη εικόνα του δρομολογίου που έχει αναλάβει. Επιπλέον, και σε αυτή την εφαρμογή υπάρχει ο εντοπισμός κυκλοφοριακής συμφόρησης. Σε περίπτωση που η εφαρμογή εντοπίσει μεγάλη κίνηση σε μια οδό βελτιστοποιεί αυτόματα την διαδρομή γλιτώνοντας τον οδηγό από χάσιμο χρόνου για την ολοκλήρωση ενός δρομολογίου.

Πέρα από τα θετικά χαρακτηριστικά της εφαρμογής, καλό θα είναι να τονιστούν και τα αρνητικά της. Πρώτα απ' όλα δεν υπάρχει έκδοση της εφαρμογής για υπολογιστές. Όπως αναφέραμε και στην αρχή της ενότητας, το λογισμικό είναι συμβατό μόνο με Android και iOS συσκευές συνεπώς είναι δύσκολο ειδικά σε επιχειρήσεις. Επιπλέον έχουν γίνει αναφορές από χρήστες για ξαφνική διακοπή (crash) της εφαρμογής κατά την διάρκεια πλοήγησης προκαλώντας σημαντικά προβλήματα στην ολοκλήρωση δρομολογίων. Ένα ακόμη μειονέκτημα αποτελεί η αδυναμία ακύρωσης ενός δρομολογίου σε περίπτωση που οριστεί. Όταν ολοκληρωθεί η εισαγωγή μιας διαδρομής, τα δεδομένα για κάθε στάση αποστέλλονται στον αντίστοιχο παραλήπτη-πελάτη. Σε περίπτωση που το δρομολόγιο ακυρωθεί πριν την ολοκλήρωση του, οι πελάτες δεν θα λάβουν σχετική ειδοποίηση δημιουργώντας σύγχυση και προβλήματα στην φήμη της εταιρείας. Τέλος αξίζει να σημειωθεί το γεγονός ότι ακόμη και στην έκδοση επί πληρωμή παρέχονται λειτουργίες που μπορούμε να τις βρούμε σε κάποια άλλη εφαρμογή εντελώς δωρεάν.

### 3.4 Upper Route Planner



*Εικόνα 8: Upper Route Planner*

(Πηγή:<https://www.upperinc.com/wp-content/uploads/2021/10/Upper-Route-Planner-Logo.png>)

Η τελευταία εφαρμογή που θα αναφέρουμε σε αυτό το κεφάλαιο ονομάζεται Upper Route Planner. Πρόκειται για μια ακόμη εφαρμογή πλοήγησης πολλαπλών διαδρομών με δυνατότητα πραγματοποίησης πολλών στάσεων. Απευθύνεται κυρίως σε διανομείς. Υπάρχει έκδοση για υπολογιστή (web application), για Android συσκευές και για iOS συσκευές. Απευθύνεται κυρίως σε επιχειρήσεις διότι πρόκειται για μια εφαρμογή επί πληρωμή. Υπάρχει φυσικά και δωρεάν δοκιμαστική έκδοση 7 ημερών όπου οι χρήστες μπορούν να κατεβάσουν την εφαρμογή, να την δοκιμάσουν στην πράξη και να αποφασίσουν αν θα προβούν σε αγορά.

Στα πλεονεκτήματα του συγκεκριμένου λογισμικού καταλογίζεται η δυνατότητα εισαγωγής στάσεων μέσω αρχείου Excel. Η εισαγωγή μπορεί να γίνει από το web app και το ανώτατο όριο των στάσεων που μπορεί να λάβει είναι 500 ανά δρομολόγιο. Αξίζει να σημειωθεί ότι σε περίπτωση που το αρχείο περιέχει λάθος πληροφορίες για την διεύθυνση μιας στάσης, η εφαρμογή με την χρήση προηγμένων αλγόριθμων μπορεί να εντοπίσει τις λανθασμένες πληροφορίες και να τις διορθώσει άμεσα για την αποφυγή προβλημάτων. Επίσης για δρομολόγια που βρίσκονται σε εξέλιξη, ο κάθε πελάτης λαμβάνει SMS στο κινητό του με την τοποθεσία του οχήματος που μεταφέρει το πακέτο του και τον χρόνο παράδοσης. Ακόμη, το γεγονός ότι απευθύνεται σε επιχειρήσεις, σημαίνει ότι οι οδηγοί παράδοσης δεμάτων θα είναι σίγουρα τουλάχιστον 2. Για τον λόγο αυτό η εφαρμογή μπορεί να ορίσει δρομολόγια για κάθε οδηγό ξεχωριστά και να τους ενημερώνει σε πραγματικό χρόνο για τυχόν αλλαγές. Εννοείται πως όπως και στις προαναφερθείσες εφαρμογές έτσι και σε αυτήν, ο ορισμός σημείων αποφυγής (π.χ διόδια, αυτοκινητόδρομοι) είναι διαθέσιμος για τους χρήστες. Τέλος, δίνεται η επιλογή διαμόρφωσης ενός δρομολογίου σύμφωνα με τον τύπο του οχήματος που έχει επιλέξει ο εκάστοτε οδηγός.

Όπως αναφέραμε και παραπάνω, η εφαρμογή είναι επί πληρωμή με δυνατότητα ελεύθερης χρήσης της έως 7 ημέρες. Επομένως μετά το πέρας της δοκιμαστικής περιόδου ο χρήστης είναι υποχρεωμένος να καταβάλλει το αντίστοιχο αντίτιμο που ορίζει η εταιρεία για την χρήση της. Το κόστος χρήσης της εφαρμογής λοιπόν συγκαταλέγεται στα αρνητικά της. Επιπλέον, το λογισμικό δεν λειτουργεί σαν πλοηγός. Πιο συγκεκριμένα, σκοπός του είναι να ορίζει διαδρομές σύμφωνα με τις στάσεις που επιθυμεί ο χρήστης και όχι να τον καθοδηγεί για το πως θα φτάσει σε αυτές. Κλείνοντας θα πρέπει να τονίσουμε την αδυναμία επικοινωνίας των οδηγών της εταιρείας απευθείας μέσω της εφαρμογής.





# Κεφάλαιο 4: Υλοποίηση εφαρμογής

---

## 4.1 Εισαγωγή

Ο web server αποτέλεσε τον πρώτο στόχο για την πραγματοποίηση της εφαρμογής. Δεν μπορεί να υπάρξει εφαρμογή που να μην επικοινωνεί με κάποιο server. Το γεγονός ότι πρόκειται για εφαρμογή στα πλαίσια διπλωματικής εργασίας, οδήγησε στην απόφαση για υλοποίηση ενός local server. Παρακάτω γίνεται μια περιληπτική αναφορά στις τεχνολογίες μέσα από τις οποίες τελικά ολοκληρώθηκε η συγκεκριμένη εργασία.

### 4.1.1 API

Το **API** ορίζεται ως Διασύνδεση Προγραμματισμού Εφαρμογών (**Application Programming Interface**). Με απλά λόγια, είναι το μέσο ανταλλαγής δεδομένων μεταξύ εφαρμογών. Όταν ένας χρήστης εισάγει δεδομένα σε μια εφαρμογή, μέσω του API η ίδια στέλνει τα δεδομένα στον server και στην συνέχεια λαμβάνει από αυτόν τα αποτελέσματα και τα εμφανίζει στον χρήστη [33]. Ένα από τα πλεονεκτήματα που προσφέρει το API είναι ότι οι προγραμματιστές μπορούν να αποκτήσουν πρόσβαση στα δεδομένα μιας άλλης εφαρμογής χωρίς να χρειάζεται να αποκτήσουν πρόσβαση και στον κώδικά της ή να γνωρίζουν πως έχει υλοποιηθεί.

Ένα παράδειγμα μέσα από την συγκεκριμένη εφαρμογή για να γίνει αντιληπτή η χρησιμότητα του API είναι η χρήση των χαρτών. Η εφαρμογή μας, προκειμένου ο χρήστης να αποκτήσει πρόσβαση στην σελίδα χαρτών, αποστέλλει στην Mapbox ένα API για να παραχωρήσει σε εμάς τα δεδομένα που απαιτούνται για την εμφάνιση του χάρτη. Στην συνέχεια, μέσω του API λαμβάνονται τα δεδομένα και τελικά εμφανίζεται ο χάρτης με τις τοποθεσίες που ζήτησε ο χρήστης.

### 4.1.2 Web Service

Η υπηρεσία ιστού (web service) είναι μια διεπαφή λογισμικού που καθορίζει μια ομάδα λειτουργιών στις οποίες μπορεί να προσπελαστεί μέσω του δικτύου χρησιμοποιώντας μηνύματα XML. Για να καθορίσει μια δραστηριότητα που θα εκτελεστεί και τα δεδομένα που θα μοιραστούν με άλλο πρόγραμμα, χρησιμοποιεί πρότυπα που έχουν δημιουργηθεί στη γλώσσα XML. Η διεπαφή κρύβει τις λεπτομέρειες υλοποίησης, επιτρέποντας στους χρήστες να έχουν πρόσβαση σε αυτήν ανεξάρτητα από την πλατφόρμα υλικού ή λογισμικού ή τη γλώσσα προγραμματισμού που χρησιμοποιείται για την ανάπτυξή της. Για να ολοκληρωθεί μια εμπορική συναλλαγή, οι υπηρεσίες ιστού μπορούν να χρησιμοποιηθούν ανεξάρτητα ή σε συνδυασμό με άλλες υπηρεσίες ιστού.

### 4.1.3 REST

Το Representational State Transfer (**REST**) ορίζεται ως αρχιτεκτονική λογισμικού που θέτει περιορισμούς στον τρόπο λειτουργίας ενός API. Στις αρχές, το REST αναπτύχθηκε ως ένα σύνολο κανόνων για τον έλεγχο της επικοινωνίας σε ένα περίπλοκο δίκτυο όπως το διαδίκτυο. Οι προγραμματιστές που ασχολούνται με τα APIs μπορούν να δημιουργήσουν API χρησιμοποιώντας μια ποικιλία αρχιτεκτονιών. Τα REST APIs είναι APIs που συμμορφώνονται με τον αρχιτεκτονικό σχεδιασμό REST. Οι υπηρεσίες ιστού RESTful (RESTful web services) είναι διαδικτυακές υπηρεσίες που χρησιμοποιούν την αρχιτεκτονική REST [34].

Μόλις λίγα χρόνια μετά την κυκλοφορία του, το REST έγινε το εξέχον μοντέλο σχεδίασης υπηρεσιών ιστού με βάση το ποσοστό αυτών που το χρησιμοποιούν. Έχει αντικαταστήσει κυρίως την αρχιτεκτονική διεπαφής που βασίζεται σε SOAP και WSDL λόγω του απλούστερου σχεδιασμού του. Μεγάλο πλεονέκτημα του REST αποτελεί η επικοινωνία μεταξύ εφαρμογών ιστού που έχουν δημιουργηθεί με διαφορετικές γλώσσες προγραμματισμού. Οι εφαρμογές μπορούν επίσης να υπάρχουν σε διάφορα συστήματα, όπως το Linux ή τα Windows. Επιπλέον ακόμα και σε εφαρμογές για κινητές συσκευές η επικοινωνία με εφαρμογές ιστού μπορεί να πραγματοποιηθεί πολύ πιο εύκολα σε αντίθεση με άλλες αρχιτεκτονικές [35].

Οι υλοποιήσεις του client και του server στο REST μπορούν να πραγματοποιηθούν ανεξάρτητα η μία από την άλλη χωρίς το ένα μέρος να γνωρίζει το άλλο. Αυτό σημαίνει ότι οι αλλαγές στον κώδικα από την πλευρά του server μπορούν να γίνουν χωρίς να επηρεάσουν τη λειτουργικότητα του client και το αντίστροφο. Μπορούν να διατηρηθούν ανεξάρτητα, εφόσον το ένα μέρος γνωρίζει την κατάλληλη μορφή μηνύματος για παράδοση στο άλλο [36].

Βασικές αρχές της αρχιτεκτονικής REST είναι:

- **Client-server**

Αυτός ο περιορισμός ουσιαστικά δηλώνει ότι οι εφαρμογές πελάτη και διακομιστή πρέπει να επιτρέπεται να αναπτύσσονται ανεξάρτητα η μία από την άλλη. Ένας client θα πρέπει να γνωρίζει μόνο τα URI πόρων. Για παράδειγμα, ο client δεν σχετίζεται με την αποθήκευση δεδομένων που είναι θέμα server και οι servers δεν σχετίζονται με τη διεπαφή client ή την κατάσταση του.

- **Uniform interface**

Αυτή είναι η θεμελιώδης μέθοδος μέσω της οποίας πρέπει να λειτουργούν τα RESTful web services. Το RESTful χρησιμοποιεί κυρίως το επίπεδο ιστού HTTP με τις ακόλουθες βασικές μεθόδους για να αλληλεπιδράσει με τους πόρους του διακομιστή[37]:

1. **POST**. Μέσω αυτής της μεθόδου, ο client μπορεί να στείλει δεδομένα στον server για την αποθήκευσή τους στην βάση δεδομένων. Ο server λαμβάνει τα δεδομένα στο request body και αφού πραγματοποιήσει έλεγχο για την εγκυρότητα τους εκτελεί τις αντίστοιχες εντολές και απαντάει στον client στέλλοντας την πληροφορία στο response body.
2. **GET**. Με την μέθοδο αυτή, ο client λαμβάνει δεδομένα από τον server. Ο server δεν αποθηκεύει δεδομένα. Ο ρόλος του είναι η ανάκτηση των δεδομένων από την βάση δεδομένων και η αποστολή τους στον client.
3. **PUT / PATCH**. Η patch μέθοδος ενημερώνει συγκεκριμένα δεδομένα από την βάση δεδομένων χωρίς να τροποποιήσει τα υπόλοιπα. Από την άλλη πλευρά, η put μέθοδος ενημερώνει ολόκληρο τον πόρο. Ο τρόπος λειτουργίας και των 2 είναι ο εξής: ο client στέλνει ένα API όπου περιλαμβάνει εκτός από τα δεδομένα που θέλει να εισάγει στην βάση και το πρωτεύον κλειδί (id) του πόρου που επιθυμεί να τροποποιήσει. Ο server, αφού το λάβει, επικοινωνεί με την βάση δεδομένων για την εκτέλεση της εντολής.
4. **DELETE**. Η μέθοδος αυτή διαγράφει μια πληροφορία από την βάση δεδομένων. Όπως και στις μεθόδους put και patch έτσι και σε αυτή την μέθοδο ο client στέλνει ένα API με το πρωτεύον κλειδί (id) του πόρου που επιθυμεί να διαγράψει. Ο server λαμβάνει το API και επικοινωνεί με την βάση δεδομένων για την διαγραφή του πόρου.

Από την πλευρά του server, κάθε απάντηση που στέλνει στον client συνδέεται με ένα συγκεκριμένο HTTP status code:

- 200 Το request ολοκληρώθηκε με επιτυχία.
- 201 Επιτυχής καταχώρηση δεδομένων στην βάση.
- 204 Το request ολοκληρώθηκε με επιτυχία αλλά δεν επιστρέφεται κάτι ως απάντηση στον client.
- 400 Το request δεν μπορεί να ολοκληρωθεί λόγω κακής σύνταξης ή κάποιου άλλου σφάλματος από την πλευρά του client.
- 403 Ο client δεν έχει δικαίωμα πρόσβασης στον πόρο που ζήτησε.
- 404 Ο πόρος δεν βρέθηκε.
- 500 Ο server αντιμετωπίζει κάποιο σφάλμα και δεν ξέρει πως να το χειριστεί.

- **Stateless**

Ένας client μπορεί να στείλει στον διακομιστή πολλά αιτήματα, αλλά το καθένα πρέπει να είναι ανεξάρτητο και να περιέχει όλες τις πληροφορίες που απαιτούνται για να το κατανοήσει ο server και να το χειριστεί σωστά. Από την άλλη, ο server δεν χρειάζεται να αποθηκεύσει καμία πληροφορία σχετικά με την κατάσταση του client.

Οποιαδήποτε κατάσταση πληροφοριών, όπως οι συνεδρίες, πρέπει να παραμείνει στον client.

- **Cacheable**

Η προσωρινή αποθήκευση, η οποία περιλαμβάνει την αποθήκευση ορισμένων responses στον client για τη μείωση των χρόνων απόκρισης του server, υποστηρίζεται από τα RESTful web services. Η τεχνική αυτή αποτρέπει την άσκοπη επικοινωνία του client με τον server βελτιώνοντας την απόδοση του συστήματος.

- **Layered system**

Η ιδέα πίσω από ένα πολυεπίπεδο σύστημα είναι ότι οποιοδήποτε επιπλέον επίπεδο, όπως ένα επίπεδο ενδιάμεσου λογισμικού, μπορεί να προστεθεί μεταξύ του client και του server που πραγματικά φιλοξενεί την υπηρεσία ιστού RESTful. Πριν πραγματοποιηθεί μια κλήση στον web server, ο client ενδέχεται να αλληλεπιδράσει με αυτό το πρόσθετο service που έχει δημιουργηθεί. Ωστόσο, η ανάπτυξη αυτού του επιπέδου πρέπει να είναι απρόσκοπτη για να αποφευχθεί η παρεμβολή στην επικοινωνία client-server. [38]

- **Code on demand**

Αυτή η ανάγκη επιτρέπει στον client να εκτελέσει συγκεκριμένο κώδικα όπως χρειάζεται. Ως αποτέλεσμα, ακόμη και όταν χρησιμοποιούν τα ίδια ακριβώς web services, διάφοροι χρήστες μπορεί να συμπεριφέρονται διαφορετικά. Αυτό το στοιχείο είναι προαιρετικό επειδή δεν αποτελεί μέρος της ίδιας της αρχιτεκτονικής. Μπορεί να χρησιμοποιηθεί για την εκτέλεση διαφόρων λειτουργιών από την πλευρά του client που είναι ταχύτερες ή πιο αποτελεσματικές.

#### 4.1.4 JSON

Τα δεδομένα που στέλνει η εφαρμογή στον server μέσω REST API και αυτά που στέλνει ο server στην εφαρμογή είναι σε μορφή JSON. Γενικά, JSON (JavaScript Object Notation) ονομάζονται δομημένα δεδομένα που παρουσιάζονται σε μια τυποποιημένη μορφή. Αν και το JSON εξελίχθηκε από τη JavaScript, είναι πλέον ένας κοινός τρόπος για τα συστήματα να αποστέλλουν και να λαμβάνουν δεδομένα. Λόγω της απλότητας και της ευανάγνωστης δομής του JSON, τα περισσότερα από τα τρέχοντα API χρησιμοποιούν ερωτήματα (queries) JSON και επιστρέφουν απαντήσεις JSON. Σε αντίθεση με την XML που χρησιμοποιείται τώρα, επιτρέπει ταχύτερη προσβασιμότητα, βελτιστοποίηση μνήμης, είναι πιο απλή και στερείται πολύπλοκης σύνταξης. Τα ονόματα αρχείων JSON χρησιμοποιούν την επέκταση .json [39].

ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ JSON	
String	<p>Οποιαδήποτε ακολουθία χαρακτήρων</p> <pre>{   "type": "car" }</pre>
Number	<p>Οποιοσδήποτε πραγματικός αριθμός</p> <pre>{   "password": 1234 }</pre>
Boolean	<p>True ή False</p> <pre>{   "active": true }</pre>
Array	<p>Σύνολα τιμών</p> <pre>[   {     "manufacturer": "Audi",     "model": "A4",     "color": "black"   },   {     "manufacturer": "Mazda",     "model": "RX3",     "color": "green"   },   {     "manufacturer": "Toyota",     "model": "Yaris",     "color": "red"   } ]</pre>
Object	<p>Μη ταξινομημένα σύνολα από ζεύγη ονομάτων/τιμών</p> <pre>{   "manufacturer": "Audi",   "model": "A4",   "color": "black" }</pre>
Null	<p>Χωρίς τιμή</p> <pre>{   "manufacturer": null }</pre>

Όπως φαίνεται και από τον πίνακα, ένα έγγραφο JSON ακολουθεί ορισμένους κανόνες συντακτικού:

- “ : ” Διαχωρίζει το όνομα από την τιμή του
- “ , “ Διαχωρίζει τα ζεύγη
- “ { “ Ανοίγει ένα αντικείμενο
- “ } “ Κλείνει ένα αντικείμενο
- “ [ “ Ανοίγει έναν πίνακα
- “ ] “ Κλείνει έναν πίνακα

#### 4.1.5 JQuery

Το jQuery είναι μια βιβλιοθήκη της JavaScript που δημιουργήθηκε για να διευκολύνει τη σύνταξη JavaScript και HTML. Το jQuery χρησιμοποιείται από το 77% των κορυφαίων 10 εκατομμυρίων ιστότοπων από τον Αύγουστο του 2022. Η μελέτη δείχνει ότι είναι μακράν η πιο ευρέως χρησιμοποιούμενη βιβλιοθήκη JavaScript, που χρησιμοποιείται τουλάχιστον τρεις έως τέσσερις φορές πιο συχνά από οποιαδήποτε άλλη βιβλιοθήκη JavaScript. Υποστηρίζει πολλά προγράμματα περιήγησης και είναι cross-platform. Ένα άλλο όνομα για αυτήν είναι το «γράψε λιγότερο, κάνε περισσότερα». επειδή μετατρέπει έναν μεγάλο αριθμό λειτουργιών ρουτίνας που χρειάζονται πολλές γραμμές κώδικα JavaScript για να ολοκληρωθούν σε μεθόδους που μπορούν να κληθούν όποτε είναι απαραίτητο. Ο αρθρωτός σχεδιασμός του jQuery επιτρέπει τη δημιουργία δυναμικών ιστοσελίδων και εφαρμογών ιστού. Είναι επίσης εξαιρετικά χρήσιμο για την απλοποίηση πολλών από τις πολύπλοκες λειτουργίες της JavaScript όπως τον χειρισμό συμβάντων, την κινούμενη εικόνα CSS και τα αιτήματα AJAX [40].

#### 4.1.6 AJAX

Το AJAX, κοινώς γνωστό ως "Asynchronous JavaScript και XML", είναι μια συλλογή μεθόδων ανάπτυξης ιστού για τη δημιουργία ασύγχρονων διαδικτυακών εφαρμογών χρησιμοποιώντας μια ποικιλία τεχνολογιών web από την πλευρά του client. Το Ajax επιτρέπει στις εφαρμογές ιστού να μεταδίδουν και να λαμβάνουν δεδομένα ασύγχρονα (στο παρασκήνιο) από έναν server χωρίς να αλλάζουν την εμφάνιση ή τη συμπεριφορά της σελίδας. Στην ουσία επιτρέπει σε ιστοσελίδες και κατ' επέκταση, σε εφαρμογές ιστού να αλλάζουν περιεχόμενο δυναμικά χωρίς να χρειάζεται να ανανεώσουν ολόκληρη τη σελίδα [41].

#### 4.1.7 Retrofit

Το Retrofit αποτελεί μια πολύ καλή βιβλιοθήκη για σύνδεση από μια εφαρμογή Android σε υπηρεσίες API που βασίζονται σε HTTP. Χρησιμοποιεί τη

θεμελιώδη λειτουργικότητα της βιβλιοθήκης OkHttp και προσθέτει έναν αριθμό δυνατοτήτων, ενώ παράλληλα απαλλάσσει τον προγραμματιστή από τον κώδικα που θα έπρεπε να γράψει αν θέλει απλώς να χρησιμοποιήσει τη βιβλιοθήκη OkHttp και καμία από τις δυνατότητες που προσφέρει το Retrofit. Ακόμη, το Retrofit έχει μια σειρά από σημαντικά χαρακτηριστικά, συμπεριλαμβανομένης της δυνατότητας γρήγορης ακύρωσης αιτημάτων, του καθορισμού κεφαλίδων αιτημάτων (π.χ μετάδοσης της κεφαλίδας εξουσιοδότησης μέσω σχολιασμών ή υποκλοπής αιτημάτων) και χρήσης διαφορετικών τύπων μετατροπών για τη μορφοποίηση των αντικειμένων αιτήματος/απάντησής του client σύμφωνα με τις προτιμήσεις του [42].

#### 4.1.8 NodeJS

Το NodeJS είναι μια Javascript πλατφόρμα που τρέχει την μηχανή Javascript V8 του Google Chrome εκτός προγράμματος περιήγησης. Περιέχει έναν μεγάλο αριθμό από λειτουργικές μονάδες (modules) με σκοπό την εύκολη και γρήγορη ανάπτυξη web εφαρμογών. Το Node.js χρησιμοποιεί το μοντέλο εισόδου/εξόδου που βασίζεται σε συμβάντα χρηστών για εκτέλεση. Επίσης ο server δεν χρειάζεται να περιμένει από ένα API να επιστρέφει δεδομένα. Προχωράει στο επόμενο API και ένας μηχανισμός ειδοποίησης το NodeJS ειδοποιεί τον server για επερχόμενη απάντηση από το προηγούμενο API [43].

#### 4.1.9 MongoDB

Η MongoDB είναι μια βάση δεδομένων που αναπτύχθηκε από την MongoDB Inc γύρω στα μέσα της δεκαετίας του 2000. Αποτελεί μέρος της οικογένειας των συστημάτων βάσεων δεδομένων NoSQL. Αυτό σημαίνει ότι τα δεδομένα δεν αποθηκεύονται σε πίνακες όπως συμβαίνει στις σχεσιακές βάσεις δεδομένων αλλά ως έγγραφα τύπου JSON. Η επιλογή μη σχεσιακής βάσης δεδομένων προκύπτει από το γεγονός ότι στις βάσεις δεδομένων αυτού του τύπου είναι δυνατή η αποθήκευση και οργάνωση δεδομένων διαφόρων τύπων μαζί, ένα χαρακτηριστικό που τις καθιστά αρκετά ευέλικτες σε αντίθεση με τις σχεσιακές βάσεις δεδομένων.

Για την διαχείριση της βάσης δεδομένων και την οπτικοποίηση των πληροφοριών χρησιμοποιήθηκε το λογισμικό MongoDB Compass. Ειδικότερα, μέσω του συγκεκριμένου λογισμικού μπορούμε να εισάγουμε νέα δεδομένα στην βάση μας μέσω CSV ή JSON αρχείων, να λάβουμε, να στείλουμε ή να διαγράψουμε δεδομένα από αυτήν μέσω ερωτημάτων (queries) ή να διαχειριστούμε την βάση δεδομένων μέσω του τερματικού που προσφέρει (MongoDB Shell).

## 4.2 Υλοποίηση Web server

Σε αυτή την ενότητα γίνεται αναφορά στα web services μέσω των οποίων επικοινωνεί η εφαρμογή με την βάση δεδομένων καθώς και τι μορφή έχει η βάση δεδομένων. Ακόμη εστιάζουμε στα δεδομένα που ανταλλάσσουν εφαρμογή και βάση, τι



μορφή έχουν και τι παραμέτρους απαιτεί κάθε μέθοδος για αποστολή και λήψη των πληροφοριών.

Αρχικά, το backend κομμάτι υλοποιήθηκε στο Visual Studio Code. Ακολουθούν τα αρχεία που απαρτίζουν το backend του server με μια μικρή επεξήγηση για την λειτουργία τους.

- **server.js**

Σε αυτό το αρχείο πραγματοποιείται το “στήσιμο” του server όπως ο ορισμός των parsers για την μετατροπή των δεδομένων που λαμβάνονται από το web και android app σε κατάλληλη μορφή, τα modules που χρειάζεται, η επικοινωνία με την βάση δεδομένων, ο καθορισμός των routes έτσι ώστε ο server να γνωρίζει ποιος κώδικας να εκτελεστεί όταν ζητείται μια συγκεκριμένη διεύθυνση URL καθώς και η θύρα (port) που θα ακούει.

- **package.json**

Το αρχείο αυτό περιλαμβάνει το όνομα του έργου, πληροφορίες σχετικά με την έκδοση του και όλα τα εργαλεία (dependencies) που έχουν εγκατασταθεί και είναι απαραίτητα για την ορθή λειτουργία του server.

- **models/Mission.js**

Στο αρχείο αυτό ορίζουμε την δομή του εγγράφου που θα αποθηκευτεί στην βάση δεδομένων και για κάθε μεταβλητή το είδος της. Στην MongoDB τα έγγραφα αποθηκεύονται μέσα σε συλλογές (collections). Στο αρχείο Mission.js δημιουργούμε ένα collection με όνομα missions όπου μέσα σε αυτό θα αποθηκεύονται όλα τα έγγραφα με τις πληροφορίες των δρομολογίων. Οι μεταβλητές του κάθε εγγράφου Mission είναι όλες αλφαριθμητικά (User\_id, Vehicle\_plate, Start\_time, End\_time, Start, Stops, Coordinates) εκτός από την μεταβλητή Active που είναι boolean ενώ για την επιτυχή καταχώρηση των δεδομένων στην βάση απαιτούνται όλες οι μεταβλητές (required: true).

- **routes/missions.js**

Αυτό το αρχείο περιλαμβάνει τις μεθόδους για την αποστολή δεδομένων στον client που αφορούν τα δρομολόγια των οδηγών. Ακόμη περιέχει μεθόδους για λήψη δεδομένων δρομολογίων από τον client και αποθήκευση τους στην βάση δεδομένων.

Αρχικά υπάρχει μια μέθοδος HTTP POST για την αποθήκευση των δεδομένων των δρομολογίων στην βάση δεδομένων. Με την χρήση κατάλληλων συναρτήσεων λαμβάνουμε την τρέχουσα ημερομηνία και ώρα και τις αποθηκεύουμε στις μεταβλητές current\_Date και current\_Time αντίστοιχα. Στη συνέχεια δημιουργούμε μια μεταβλητή mission που θα εκχωρηθεί σε ένα καινούργιο Mission από το μοντέλο που είδαμε στο αρχείο Mission.js. Στις μεταβλητές User\_id, Vehicle\_plate,

End\_time, Start, Stops, Coordinates και Active αποθηκεύονται τα δεδομένα από το request του client. Λόγω του γεγονότος ότι τα δεδομένα αυτά βρίσκονται στο body του request, χρησιμοποιούμε έναν body parser για το πέρασμα τους στον server. Μετά την δημιουργία της μεταβλητής mission, κάνουμε αναζήτηση στην βάση δεδομένων και συγκεκριμένα στην συλλογή missions για έγγραφο με αριθμό πινακίδας αυτόν που έχουμε λάβει από τον client ΚΑΙ την μεταβλητή Active να είναι true. Αν ισχύει αυτή η συνθήκη τότε ενημερώνει το ήδη υπάρχον αρχείο με τις καινούργιες πληροφορίες. Σε αντίθετη περίπτωση λαμβάνει το JSON αντικείμενο, πραγματοποιεί μια καινούργια καταχώρηση στην βάση δεδομένων δημιουργώντας ένα νέο έγγραφο και αποθηκεύοντας μέσα σε αυτό τα δεδομένα. Σε κάθε περίπτωση εάν η αποθήκευση πληροφορίας στην βάση δεδομένων αποτύχει επιστρέφει 200 HTTP response σε HTML μορφή με το μήνυμα σφάλματος στον client.

Εκτός από την POST υπάρχουν και GET μέθοδοι για την αποστολή των δεδομένων στον client. Αρχικά υπάρχει μια HTTP GET μέθοδος για την ανάκτηση όλων των εγγράφων από την συλλογή missions. Τα δεδομένα μέσα στα έγγραφα είναι σε μορφή JSON. Η μέθοδος αυτή επιστρέφει τον αριθμό πινακίδας, την ημερομηνία έναρξης του δρομολογίου, την αφετηρία, τις ενδιάμεσες στάσεις, την ημερομηνία ολοκλήρωσης και την τρέχουσα κατάσταση του. Επιστρέφει τα αρχεία σε μορφή JSON σε περίπτωση επιτυχίας (200 response) αλλιώς επιστρέφει HTTP 400 response σε HTML μορφή με την αιτία του σφάλματος.

Ακόμη υπάρχει μια GET μέθοδος που επιστρέφει στον client συγκεκριμένο έγγραφο με βάση το id του που λαμβάνεται από το URL του route. Εξάγει το User\_id από το route και πραγματοποιεί αναζήτηση στο collection missions της βάσης δεδομένων για έγγραφο που περιέχουν αυτό. Επιστρέφει τα αρχεία σε μορφή JSON ή μήνυμα λάθους σε HTML μορφή ότι δεν υπάρχει τέτοιο έγγραφο σε περίπτωση επιτυχίας αλλιώς επιστρέφει HTTP 400 response με την αιτία του σφάλματος.

Σε άλλη μια μέθοδο GET επιστρέφεται στον client συγκεκριμένο έγγραφο που περιέχει την ημερομηνία έναρξης, την αφετηρία και όλες τις στάσεις του δρομολογίου με βάση τον αριθμό πινακίδας του οχήματος του που περιλαμβάνεται στο URL του route. Και σε αυτή την περίπτωση επιστρέφει τα αρχεία σε μορφή JSON ή μήνυμα ότι δεν υπάρχει τέτοιο έγγραφο σε HTML μορφή σε περίπτωση επιτυχίας αλλιώς επιστρέφει HTTP 400 response με την αιτία του σφάλματος.

Επιπλέον η μέθοδος DELETE διαγράφει ένα έγγραφο από την συλλογή missions με βάση το id που δέχεται ως παράμετρο στο route. Αν υπάρχει σφάλμα επιστρέφει HTTP 400 response σε HTML μορφή με την αιτία του σφάλματος ενώ αν έχουμε 200 response επιστρέφει απλό κείμενο ενημερώνοντας τον client αν έχει διαγραφεί με επιτυχία ή όχι.

Τέλος με βάση το route, υπάρχουν GET μέθοδοι όπου φορτώνεται στο πρόγραμμα περιήγησης ο αντίστοιχος HTML κώδικας για προβολή της σελίδας με την χρήση της συνάρτησης render().

- **models/Vehicle.js**

Στο αρχείο αυτό δημιουργούμε ένα collection με όνομα vehicles όπου μέσα σε αυτό θα αποθηκεύονται όλα τα έγγραφα με τις πληροφορίες των οχημάτων κάθε οδηγού.

Οι μεταβλητές του κάθε εγγράφου VEHICLE είναι όλες αλφαριθμητικά (User\_id, Vehicle\_plate, Manufacturer, Model, Type, Color, CC, HP, Date\_of\_construction) ενώ για την επιτυχή καταχώρηση των δεδομένων στην βάση απαιτούνται όλες οι μεταβλητές (required: true).

- **routes/vehicle.js**

Αυτό το αρχείο περιλαμβάνει τις μεθόδους για την αποστολή δεδομένων στον client που αφορούν τα οχήματα των οδηγών. Ακόμη περιέχει μεθόδους για λήψη δεδομένων σχετικά με τα οχήματα από τον client και αποθήκευση τους στην βάση δεδομένων.

Αρχικά υπάρχει μια μέθοδος HTTP POST για την αποθήκευση των δεδομένων για κάθε όχημα στην βάση δεδομένων. Τα δεδομένα που υπάρχουν στο body του request είναι σε μορφή JSON και περιλαμβάνουν το id του χρήστη, τον κατασκευαστή του οχήματος, το μοντέλο του, τον αριθμό πινακίδας του, τον τύπο οχήματος, το χρώμα του, τα κυβικά του, την ιπποδύναμή του και την ημερομηνία κατασκευής του. Πριν αποθηκευτούν στην βάση ελέγχονται ως προς την εγκυρότητα τους. Αν υπάρξει σφάλμα επιστρέφει στον client HTTP 200 response σε HTML μορφή ενημερώνοντας τον για το σφάλμα μέσω μηνύματος σε HTML μορφή. Σε περίπτωση που δεν εντοπιστεί στο collection missions της βάσης δεδομένων άλλο όχημα με τον ίδιο αριθμό πινακίδας δημιουργείται μια μεταβλητή newVehicle που θα εκχωρηθεί σε ένα καινούργιο Vehicle. Η μεταβλητή newVehicle περιλαμβάνει τα δεδομένα που έλαβε από τον client και πραγματοποιεί μια καινούργια καταχώρηση στην βάση δεδομένων δημιουργώντας ένα νέο έγγραφο. Αν και σε αυτή την περίπτωση υπάρξει σφάλμα ενημερώνει τον client για το σφάλμα μέσω κειμένου.

Το συγκεκριμένο αρχείο περιλαμβάνει 5 GET μεθόδους. Στην πρώτη, ανακτώνται όλα τα έγγραφα από την συλλογή vehicles. Κάθε έγγραφο περιλαμβάνει το id του χρήστη, τον κατασκευαστή του οχήματος, το μοντέλο του, τον αριθμό πινακίδας του, τον τύπο οχήματος, το χρώμα του, τα κυβικά του, την ιπποδύναμή του και την ημερομηνία κατασκευής του. Σε περίπτωση σφάλματος επιστρέφει HTTP 400 response με την αιτία του σφάλματος ενώ σε περίπτωση επιτυχούς απόκρισης ο client λαμβάνει τα αρχεία σε μορφή JSON.

Η επόμενη GET μέθοδος επιστρέφει τον αριθμό πινακίδας ενός οχήματος με βάση το User\_id που περιλαμβάνεται στο URL του route. Το ερώτημα (query) που στέλνεται στο collection vehicles της βάσης δεδομένων περιλαμβάνει το id του χρήστη που εξάγεται από το route και επιστρέφει τον αριθμό πινακίδας του οχήματος (Vehicle\_plate) σε μορφή JSON ή ένα text που ενημερώνει τον client ότι δεν υπάρχει τέτοιο όχημα καταχωρημένο στην βάση. Αν υπάρχει σφάλμα επιστρέφει HTTP 400 response με την αιτία του σφάλματος.

Μια ακόμη GET μέθοδος επιστρέφει μόνο τα έγγραφα που το User\_id τους είναι αυτό που λαμβάνεται από το URL του route. Αν υπάρχει σφάλμα ακολουθείται η ίδια διαδικασία με τις προηγούμενες μεθόδους.

Επιπλέον μια GET μέθοδος επιστρέφει τα έγγραφα με το Vehicle\_plate που λαμβάνεται από το URL του route.

Η τελευταία GET μέθοδος στέλνει ένα query στο collection vehicles της βάσης δεδομένων για να λάβει τον αριθμό των εγγράφων της συλλογής vehicles για έναν συγκεκριμένο χρήστη. Το id το λαμβάνει από το URL του route και επιστρέφει σε μορφή JSON το response.

Η UPDATE μέθοδος λαμβάνει από το body του request τον κατασκευαστή του οχήματος, το μοντέλο του, τον αριθμό πινακίδας του, τον τύπο οχήματος, το χρώμα του, τα κυβικά του, την ιπποδύναμή του και την ημερομηνία κατασκευής του. Αν υπάρξει σφάλμα επιστρέφει HTTP 200 response ενημερώνοντας τον client για το σφάλμα μέσω μηνύματος σε HTML μορφή. Στην συνέχεια ελέγχει τον αριθμό πινακίδας που έλαβε. Αν είναι ίδιος με τον τρέχων αριθμό πινακίδας του οχήματος ενημερώνει αυτόματα το έγγραφο με αυτή την πινακίδα. Αν δεν είναι, πραγματοποιεί αναζήτηση σε ολόκληρο το collection για οχήματα με αυτή την πινακίδα. Σε περίπτωση που βρει όχημα με την ίδια πινακίδα επιστρέφει στον client σχετικό μήνυμα. Σε κάθε άλλη περίπτωση κάνει αντικατάσταση των δεδομένων του εγγράφου της βάσης με τα δεδομένα από το body.

Τέλος η μέθοδος DELETE διαγράφει ένα έγγραφο από την συλλογή vehicles με βάση το Vehicle\_plate που δέχεται από τον client στο body του request. Στέλνει ένα query στην βάση όπου ζητάει την διαγραφή του εγγράφου που έχει για αριθμό πινακίδας αυτό το Vehicle\_plate. Σε κάθε περίπτωση επιστρέφει στον client ένα response για την κατάσταση της διαδικασίας διαγραφής.

Για την εμφάνιση των σελίδων που αφορούν τα οχήματα, με βάση το route, υπάρχουν GET μέθοδοι όπου φορτώνεται στο πρόγραμμα περιήγησης ο αντίστοιχος HTML κώδικας με την βοήθεια της συνάρτησης render().

- **models/User.js**

Στο αρχείο αυτό δημιουργούμε ένα collection με όνομα users όπου μέσα σε αυτό θα αποθηκεύονται όλα τα έγγραφα με τις πληροφορίες των χρηστών του συστήματος. Οι μεταβλητές του κάθε εγγράφου USER είναι όλες αλφαριθμητικά (FirstName, LastName, Account\_Type, Email, Password, Recovery\_PIN, Register\_time) ενώ για την επιτυχή καταχώρηση των δεδομένων στην βάση απαιτούνται όλες οι μεταβλητές (required: true).

- **routes/users.js**

Αυτό το αρχείο περιλαμβάνει τις μεθόδους για την αποστολή δεδομένων στον client που αφορούν τον ίδιο τον οδηγό. Ακόμη περιέχει μεθόδους για λήψη δεδομένων σχετικά με τον χρήστη από τον client και αποθήκευση τους στην βάση δεδομένων.

Αρχικά περιλαμβάνει 3 POST μεθόδους. Η πρώτη μέθοδος καταχωρεί τις πληροφορίες ενός νέου χρήστη. Το request body είναι σε JSON μορφή και περιλαμβάνει το όνομα, το επώνυμο, το είδος λογαριασμού, το email, τον κωδικό πρόσβασης, την επιβεβαίωση κωδικού πρόσβασης και το PIN επαναφοράς που θα γνωρίσουμε αναλυτικά στο επόμενο κεφάλαιο. Αρχικά, ελέγχει αν τα δεδομένα που στέλνει ο client είναι έγκυρα. Αν δεν είναι, επιστρέφει HTTP response (200 OK) στον client που τον ενημερώνει για το σφάλμα σε HTML μορφή. Αν είναι έγκυρα,

πραγματοποιείται έλεγχος στο collection users της βάσης δεδομένων για υπάρχον χρήστη με αυτό το email. Σε περίπτωση που υπάρχει χρήστης, πάλι επιστρέφει HTTP response στον client. Σε αντίθετη περίπτωση δημιουργείται μια μεταβλητή newUser που θα εκχωρηθεί σε έναν καινούργιο User αποθηκεύοντας τα δεδομένα από το request του client και επιστρέφοντας σε αυτόν ένα HTTP response (200 OK) για την ενημέρωση της επιτυχημένης καταχώρησης.

Η επόμενη POST μέθοδος αφορά την είσοδο του χρήστη στο σύστημα από την android εφαρμογή. Ο client στέλνει σε μορφή JSON το email και τον κωδικό πρόσβασης και αφού πραγματοποιηθεί έλεγχος εγκυρότητας των δεδομένων, αποστέλλεται στην βάση δεδομένων query για την εύρεση στο collection users ενός εγγράφου με email και κωδικό πρόσβασης αυτό που λήφθηκε από το request body. Σε περίπτωση που βρεθεί τέτοιο έγγραφο, επιστρέφεται στον client ένα HTTP response (200 OK) με το id του χρήστη σε μορφή JSON. Αν δεν βρεθεί το έγγραφο επιστρέφει πάλι 200 HTTP response που ενημερώνει τον χρήστη με μήνυμα σε HTML μορφή.

Η είσοδος του χρήστη από την web εφαρμογή πραγματοποιείται μέσω άλλης μεθόδου POST η οποία χρησιμοποιεί το passport-local, ένα πολύ γνωστό middleware όπου το μόνο που χρειάζεται είναι το email και τον κωδικό πρόσβασης για να πιστοποιήσει έναν χρήστη. Λαμβάνει τα δεδομένα από τα αντίστοιχα πεδία που έχει εισάγει ο χρήστης, επικοινωνεί με την βάση δεδομένων για την εγκυρότητα των στοιχείων και αν είναι σωστά τα στοιχεία τον ανακατευθύνει στην αρχική σελίδα. Σε αντίθετη περίπτωση του εμφανίζει μήνυμα λάθους στην οθόνη με την χρήση του εργαλείου connect-flash.

Σε συνέχεια των POST μεθόδων, ακολουθεί αυτή που έχει ως ρόλο την επαναφορά κωδικού πρόσβασης του χρήστη από την αρχική σελίδα. Αφού λάβει τα δεδομένα τα οποία έχουν JSON μορφή (email, νέος κωδικός πρόσβασης, επιβεβαίωση κωδικού, PIN επαναφοράς) και ελέγξει ότι είναι έγκυρα, αναζητά στην συλλογή users της βάσης δεδομένων για το email που δόθηκε. Αν υπάρχει επιστρέφει 200 HTTP response που ενημερώνει τον χρήστη με σχετικό μήνυμα σε HTML μορφή. Σε άλλη περίπτωση, αν ο χρήστης έχει εισάγει το σωστό PIN επαναφοράς τότε ενημερώνεται ο κωδικός πρόσβασης με τον νέο κωδικό που καταχώρησε. Με λάθος PIN επαναφοράς ειδοποιείται ο client μέσω response σε HTML μορφή.

Για την αλλαγή του κωδικού πρόσβασης μέσα από την σελίδα του χρήστη υλοποιήθηκε ακόμη μια POST μέθοδος. Η λειτουργία της με την προηγούμενη μέθοδο είναι σχεδόν ίδια με την μοναδική διαφορά ότι σε αυτή την μέθοδο δεν απαιτείται η γνώση του PIN επαναφοράς αλλά του τρέχον κωδικού πρόσβασης.

Εκτός από τις POST μεθόδους, για την αποστολή δεδομένων στον client υπάρχουν 5 GET μέθοδοι. Η πρώτη μέθοδος επιστρέφει στον client όλα τα έγγραφα από τον collection users σε μορφή JSON.

Η επόμενη μέθοδος, επιστρέφει μόνο το έγγραφο από το collection users που έχει ως id το id που περιέχεται στο URL του route. Και σε αυτή την περίπτωση το response είναι σε JSON μορφή.

Η τρίτη μέθοδος λαμβάνει ως παράμετρο το email, στέλνει ένα query στην βάση δεδομένων για εύρεση στο collection users του συγκεκριμένου εγγράφου που το περιέχει ζητώντας το είδος λογαριασμού και στην συνέχεια επιστρέφει στον client ένα JSON response με αυτό.

Μια ακόμη GET μέθοδος επιστρέφει το id του χρήστη με βάση το email του. Το response που θα λάβει ο client είναι σε JSON μορφή.

Η τελευταία GET μέθοδος αφού λάβει ως παράμετρο το id του χρήστη, στέλνει ένα query στην βάση δεδομένων για να λάβει το όνομα και το επώνυμο του και τα επιστρέφει στον client ως JSON.

Η UPDATE μέθοδος λαμβάνει από τον client το όνομα, το επώνυμο και το email και ελέγχει αν είναι έγκυρα. Στην περίπτωση που είναι, αν ο χρήστης κατά την επεξεργασία των στοιχείων του δεν άλλαξε το email του τότε ένα query αποστέλλεται στην βάση δεδομένων για την εύρεση στο collection ενός εγγράφου με το email αυτό. Αν βρεθεί τέτοιο έγγραφο τότε πραγματοποιείται η αντικατάσταση των δεδομένων. Αν δεν βρεθεί, επιστρέφει HTTP response που ενημερώνει τον χρήστη. Για την περίπτωση που ο χρήστης αλλάξει το email του πραγματοποιείται εκ νέου αναζήτηση στην συλλογή users της βάσης δεδομένων για την ύπαρξη κάποιου χρήστη που κατέχει το καινούριο email ενώ αν δεν βρεθεί χρήστης τότε πραγματοποιείται η καταχώρηση του νέου email και των υπόλοιπων δεδομένων που πρέπει να αλλάξουν.

Κλείνοντας, υπάρχουν 2 DELETE μέθοδοι. Η μία μέθοδος λαμβάνει από το body του request το id του χρήστη και στέλνοντας ένα query στην συλλογή users της βάσης δεδομένων ζητάει την διαγραφή του εγγράφου με αυτό το id.

Η δεύτερη DELETE μέθοδος διαγράφει οριστικά κάθε πληροφορία του χρήστη από το σύστημα. Πιο συγκεκριμένα, με βάση το id που θα λάβει από τον client, στέλνει 3 queries για κάθε collection της βάσης δεδομένων (missions, vehicles, users) ζητώντας την διαγραφή των εγγράφων που έχουν αυτό το id.

Στο συγκεκριμένο αρχείο έχουμε ορίσει και ένα cookie το οποίο δημιουργείται κάθε φορά που πραγματοποιεί επιτυχή είσοδο ένας χρήστης στην web εφαρμογή. Το cookie παράγει ένα δικό του μοναδικό id για κάθε χρήστη. Το πρόγραμμα περιήγησης λαμβάνει αυτό το cookie κατά την είσοδο του χρήστη, το αποθηκεύει και κάθε φορά που θα κάνει κάποιο request, ο server θα γνωρίζει με βάση το id ποιος είναι, θα τον πιστοποιεί και θα του δίνει πρόσβαση σε κάθε πόρο που ζητάει. Το cookie διαγράφεται με την αποσύνδεση του χρήστη από το σύστημα και την ανακατεύθυνση του στην σελίδα εισόδου.

Για την εμφάνιση των σελίδων που αφορούν τους χρήστες, με βάση το route, υπάρχουν GET μέθοδοι όπου φορτώνεται στο πρόγραμμα περιήγησης ο αντίστοιχος HTML κώδικας με την βοήθεια της συνάρτησης render().

### 4.3 Υλοποίηση Android εφαρμογής

Η ενότητα αυτή αφορά τον τρόπο υλοποίησης της android εφαρμογής. Πιο συγκεκριμένα, γίνεται αναφορά σε κάθε activity, πως επικοινωνεί με τον server, τι δεδομένα στέλνει και λαμβάνει από αυτόν καθώς και τον ρόλο των APIs.

- **AndroidManifest.xml**

Προκειμένου η εφαρμογή να έχει δικαίωμα χρήσης της τοποθεσίας και πρόσβασης στο internet, θα πρέπει κάπου μέσα στον κώδικα να ζητάει από το android σχετική άδεια. Η άδεια αυτή ορίζεται στο manifest αρχείο. Γενικά στο αρχείο αυτό ορίζεται το package name της εφαρμογής και το εικονίδιο της. Ακόμη δηλώνονται όλα τα activities της εφαρμογής, τα ονόματά τους και οι ιδιότητές τους.

- **Api.java**

Σε αυτό το αρχείο ορίζονται όλα τα API endpoints του Retrofit δηλαδή το URL στο οποίο θα απευθυνθεί το API για την αποστολή ή λήψη δεδομένων. Η αποστολή δεδομένων στον server και η λήψη δεδομένων από αυτόν γίνεται με την χρήση της βιβλιοθήκης okhttp ενώ μέσω των μηχανισμών okhttp interceptors παρέχεται η δυνατότητα παρακολούθησης της εξέλιξης ενός request. Τα δεδομένα που λαμβάνει η εφαρμογή από τον server είναι σε μορφή JSON. Η χρήση του Gson, ενός converter, συμβάλλει στην μετατροπή των δεδομένων που είναι σε μορφή JSON σε αντικείμενα java (java objects).

- **RetrofitInterface.java**

Αφού έχουν οριστεί τα API endpoints, θα πρέπει να καθοριστούν και οι HTTP μέθοδοι έτσι ώστε το Retrofit API να γνωρίζει τι request θα στείλει στον server. Πιο συγκεκριμένα υπάρχουν 6 POST methods, 6 GET methods και 2 DELETE methods. Στις POST μεθόδους, τα δεδομένα αποστέλλονται στο body του request σε μορφή JSON, στις GET μεθόδους η κάθε μεταβλητή αποστέλλεται ως παράμετρος στο URL του API ενώ στις DELETE μεθόδους στο body του request όπως στις POST.

**Όλες οι μέθοδοι θα μελετηθούν αναλυτικά μέσα στα αρχεία που καλούνται.**

Από τις POST μεθόδους, η πρώτη αφορά την αποστολή του email και κωδικού πρόσβασης για την είσοδο στην εφαρμογή.

Η δεύτερη POST μέθοδος αναφέρεται στην εγγραφή ενός χρήστη στο σύστημα. Στο body του request περιλαμβάνεται το όνομα του, το επώνυμο του, είδος του λογαριασμού, το email του, ο κωδικός πρόσβασής του, η επιβεβαίωση κωδικού και ο αριθμός επαναφοράς (PIN).

Η τρίτη μέθοδος σχετίζεται με την εγγραφή ενός οχήματος στο σύστημα. Τα δεδομένα που αποστέλλονται είναι το id του χρήστη, ο κατασκευαστής του οχήματος, το μοντέλο του, ο αριθμός πινακίδας του, ο τύπος οχήματος, το χρώμα του, τα κυβικά του, η ιπποδύναμή του και η ημερομηνία κατασκευής του.

Η τέταρτη μέθοδος αφορά την αποστολή των δεδομένων που αφορούν ένα δρομολόγιο. Πιο αναλυτικά το id του χρήστη, τον αριθμό πινακίδας του οχήματος που πραγματοποιεί το δρομολόγιο, την αφετηρία, όλες τις στάσεις, τις συντεταγμένες των στάσεων και της αφετηρίας, την κατάσταση του δρομολογίου και την ημερομηνία και ώρα λήξης του δρομολογίου (αν έχει ολοκληρωθεί).

Η πέμπτη μέθοδος έχει σχέση με την ανάκτηση κωδικού πρόσβασης ενός χρήστη. Ο client στέλνει το email, τον αριθμό επαναφοράς, τον νέο κωδικό πρόσβασης και την επιβεβαίωση κωδικού.

Η έκτη μέθοδος σχετίζεται με την καταχώρηση νέου κωδικού πρόσβασης στέλνοντας στο server το email, τον τρέχον κωδικό πρόσβασης, τον καινούργιο κωδικό πρόσβασης και την επιβεβαίωση κωδικού.

Από τις GET μεθόδους, η πρώτη επιστρέφει το id του χρήστη με βάση το email που έλαβε.

Η δεύτερη μέθοδος επιστρέφει τον αριθμό πινακίδας του οχήματος με βάση το id του χρήστη.

Η τρίτη μέθοδος επιστρέφει το όνομα και το επίθετο του χρήστη αφού λάβει στο id.

Η τέταρτη μέθοδος αφού λάβει το id του χρήστη επιστρέφει το όνομα του, το επώνυμο του, τον αριθμό πινακίδας του οχήματός του, το email του και την ημερομηνία εγγραφής του στο σύστημα.

Η πέμπτη μέθοδος δέχεται ως παράμετρο τον αριθμό πινακίδας ενός οχήματος και επιστρέφει το id του χρήστη, τον κατασκευαστή του οχήματος, το μοντέλο του, τον τύπο οχήματος, το χρώμα του, τα κυβικά του, την ιπποδύναμή του και την ημερομηνία κατασκευής του.

Η τελευταία GET μέθοδος επιστρέφει τις πληροφορίες ενός δρομολογίου με βάση τον αριθμό πινακίδας που θα λάβει από τον client. Οι πληροφορίες αυτές είναι το id του δρομολογίου, η ημερομηνία έναρξής του, η αφετηρία και το σημείο τερματισμού.

Από τις DELETE μεθόδους η πρώτη διαγράφει μόνο το έγγραφο από την συλλογή users της βάσης δεδομένων που έχει ως id χρήστη αυτό που έλαβε από τον client.

Η επόμενη DELETE μέθοδος διαγράφει κάθε έγγραφο από κάθε συλλογή (users, vehicles, missions) με βάση το id του χρήστη.

- **SignIn.java**

Το αρχείο αυτό παράγει την σελίδα εισόδου της android εφαρμογής. Αρχικά περιλαμβάνει τα πεδία του email και του κωδικού πρόσβασης για την είσοδο του χρήστη στο σύστημα. Όπως παρουσιάστηκε και στο αρχείο RetrofitInterface.java, η POST μέθοδος στέλνει ένα HTTP request στον server όπου στο body του request αποθηκεύονται σε μορφή JSON το όνομα και επώνυμο του. Η μετατροπή των δεδομένων σε JSON μορφή γίνεται μέσω του Gson converter με την χρήση μιας POJO (Plain Old Java Object) κλάσης (PostLogin) που σειριοποιεί σε JSON μορφή τις μεταβλητές. Σε περίπτωση επιτυχούς response (200 HTTP) λαμβάνει από τον server είτε το id του σε μορφή JSON είτε το μήνυμα σφάλματος σε HTML μορφή. Αφού λάβει το id του χρήστη, στέλνει ένα GET request στον server για την



ανάκτηση του αριθμού πινακίδας του σύμφωνα με το id. Το id το “περνάει” ως παράμετρο στο URL του API. Λόγω του γεγονότος ότι το response του server είναι σε μορφή JSON χρησιμοποιείται ο Gson converter για την μετατροπή του σε java object. Η μετατροπή αυτή πραγματοποιείται μέσω της POJO κλάσης GetVehiclePlate στην οποία κάθε παράμετρος από το response του server ανατίθενται σε μια αντίστοιχη μεταβλητή.

Αν ο χρήστης ξεχάσει τον κωδικό πρόσβασης του, μπορεί να εισάγει το email του, τον αριθμό επαναφοράς, τον νέο κωδικό πρόσβασης και την επιβεβαίωση κωδικού πρόσβασης στα αντίστοιχα πεδία και μέσω του API να αποσταλούν στον server σε μορφή JSON με την βοήθεια της POJO κλάσης PostRecoverPassword. Από τον sever επιστρέφεται HTTP 200 response σε HTML μορφή που τον ενημερώνει για την επιτυχή αλλαγή ή μήνυμα σφάλματος. Σε κάθε άλλη περίπτωση επιστρέφει HTTP 400 με μήνυμα σφάλματος σε HTML μορφή.

- **User.java**

Αυτό το αρχείο αποτελεί την αρχική σελίδα του χρήστη. Η εφαρμογή προκειμένου να μπορέσει να αποκτήσει την τοποθεσία της συσκευής αρχικά χρειάζεται το API του service της Google που κάνει αυτή την δουλειά. Στην συνέχεια γίνεται έλεγχος στα permissions της εφαρμογής για την χρήση του GPS. Για τον έλεγχο της κατάστασης του GPS (ενεργό/ανενεργό) έχει δημιουργηθεί ένα thread που το ελέγχει ανά 5 δευτερόλεπτα ενημερώνοντας κάθε φορά την εφαρμογή.

Λόγω του γεγονότος ότι στην αρχική οθόνη εμφανίζεται το ονοματεπώνυμο του χρήστη που έχει εισέλθει, υπάρχει μια GET μέθοδος που δέχεται ως παράμετρο το id του και επιστρέφει από το collection users της βάσης δεδομένων το όνομα και το επώνυμο σε μορφή JSON μέσω της POJO κλάσης GetUsername σε περίπτωση επιτυχούς request (HTTP 200). Σε κάθε άλλη περίπτωση επιστρέφει HTTP 400 με μήνυμα σφάλματος σε HTML μορφή. Σε ένα άλλο thread (**ThreadWork.java**), αφού η εφαρμογή έχει λάβει τις συντεταγμένες του χρήστη μετατρέπει το γεωγραφικό μήκος και πλάτος που έλαβε σε ονόματα οδών. Η διαδικασία αυτή πραγματοποιείται μέσω της κλάσης Geocoder. Ο Geocoder χρησιμοποιείται για την αναζήτηση τοποθεσιών από το API της Google, την μετατροπή των συντεταγμένων σε διευθύνσεις και το αντίστροφο (reverse geocoding). Αφού ληφθούν οι πληροφορίες τοποθεσίας της συσκευής, εμφανίζονται στην οθόνη.

Τέλος για την διαγραφή του χρήστη από το σύστημα έχει υλοποιηθεί αντίστοιχο API που στέλνει στον server μια DELETE μέθοδο με το id του σε μορφή JSON με την βοήθεια την κλάσης DeleteUser που αναλαμβάνει την μετατροπή των δεδομένων στη σωστή μορφή.

- **MyProfile.java**

Μέσω του αρχείου αυτού, εμφανίζονται στην οθόνη οι πληροφορίες που αφορούν τον χρήστη-οδηγό. Για την λήψη των πληροφοριών αυτών, χρησιμοποιείται ένα API που στέλνει ένα HTTP GET request στον server. Το response body που επιστρέφει περιλαμβάνει το όνομα χρήστη, το επώνυμο του, ο email του και την ημερομηνία

που έκανε την εγγραφή του στην εφαρμογή. Τα δεδομένα αυτά έχουν JSON μορφή και μετατρέπονται σε αντικείμενα java μέσω της κλάσης GetAccountInfo.

Κάθε χρήστης έχει δικαίωμα μέσα από την σελίδα αυτή να αλλάξει τον κωδικό πρόσβασής του. Θα πρέπει να εισάγει το email του, τον τωρινό κωδικό πρόσβασης του, τον νέο κωδικό πρόσβασης και την επιβεβαίωση κωδικού πρόσβασης στα αντίστοιχα πεδία και μέσω του API να αποσταλεί ένα HTTP POST request στον server σε μορφή JSON με την βοήθεια της POJO κλάσης PostChangePass. Από τον sever επιστρέφεται HTTP 200 response σε HTML μορφή που τον ενημερώνει για την επιτυχή αλλαγή ή μήνυμα σφάλματος. Σε κάθε άλλη περίπτωση επιστρέφει HTTP 400 με μήνυμα σφάλματος σε HTML μορφή.

- **MyVehicle.java**

Αυτή η σελίδα εμφανίζει τις πληροφορίες του οχήματος που έχει επιλέξει ο χρήστης. Αρχικά μέσω API αποστέλλεται ένα HTTP GET request για την λήψη των πληροφοριών του οχήματος με βάση τον αριθμό πινακίδας που υπάρχει ως παράμετρος στο URL του API. Ο server επιστρέφει τον κατασκευαστή του οχήματος, το μοντέλο του, τον αριθμό πινακίδας του, τον τύπο οχήματος, το χρώμα του, τα κυβικά του, την ιπποδύναμή του και την ημερομηνία κατασκευής του σε μορφή JSON αν υπάρξει επιτυχής response (HTTP 200) με την κλάση GetVehicle να αναλαμβάνει το parsing των δεδομένων που ελήφθησαν.

Για τους χρήστες που έχουν στην κατοχή τους περισσότερα από ένα οχήματα, δίνεται η δυνατότητα πρόσβασης στις πληροφορίες και των υπόλοιπων οχημάτων. Μπορούν να επιλέξουν ανάμεσα σε μια λίστα οχημάτων που διαθέτουν, αυτό που επιθυμούν να προβάλλουν τις πληροφορίες του. Η λίστα αυτή προκύπτει ύστερα από ένα GET request που στέλνει ο client για την λήψη από τον server όλων των οχημάτων που έχουν το id του χρήστη. Το id περιλαμβάνεται στο URL του API. Το response body είναι σε JSON μορφή και η κλάση GetVehiclePlate το μετατρέπει σε java object.

Αφού ληφθεί με επιτυχία η λίστα μετά η εφαρμογή στέλνει εκ νέου GET request για να πάρει τις πληροφορίες του επιλεγμένου οχήματος.

- **MyMissions.java**

Σε αυτή την σελίδα εμφανίζονται όλα τα δρομολόγια που έχει πραγματοποιήσει ο οδηγός με το επιλεγμένο όχημα. Μέσω ενός ακόμη API η εφαρμογή στέλνει ένα HTTP GET request για να λάβει την ημερομηνία έναρξης, την αφετηρία, και όλες τις στάσεις της διαδρομής. Στην συνέχεια αφού λάβει τα JSON δεδομένα και τα μετατρέψει στην κατάλληλη μορφή με την κλάση GetMissions, εμφανίζει στον χρήστη μια λίστα με την ημερομηνία και ώρα έναρξης κάθε δρομολογίου, το σημείο εκκίνησης και το σημείο τερματισμού.

- **Navigation.java**

Η σελίδα αυτή αποτελεί και το κυρίως θέμα της διπλωματικής εργασίας, την πλοήγηση. Εδώ ο χρήστης εισάγει διευθύνσεις και η εφαρμογή παράγει την αντίστοιχη διαδρομή. Πρώτα απ' όλα για να δουλέψει σωστά αυτό το activity απαιτείται η επικοινωνία με την Mapbox η οποία πραγματοποιείται μέσω API. Το API απαιτεί ένα access token για την πιστοποίηση του χρήστη που χρησιμοποιεί τα services της εταιρείας. Με την είσοδο του χρήστη σε αυτή την σελίδα-activity πραγματοποιείται έλεγχος σχετικά με τις άδειες που έχει παραχωρήσει το android στην εφαρμογή. Για τον εντοπισμό της συσκευής απαιτείται παραχώρηση άδειας στις υπηρεσίες τοποθεσίας της. Όταν ο χρήστης παραχωρήσει το δικαίωμα αυτό και η εφαρμογή πραγματοποιήσει επιτυχή επικοινωνία με την Mapbox για λήψη γεωγραφικών δεδομένων εμφανίζεται ο χάρτης με την τρέχουσα τοποθεσία του χρήστη. Η εύρεση της τοποθεσίας του χρήστη γίνεται σε ξεχωριστό thread για να μην επιβαρύνεται επιπλέον το main thread.

Η εισαγωγή μιας διεύθυνσης μπορεί να πραγματοποιηθεί με δυο τρόπους. Ο ένας είναι πληκτρολογώντας την διεύθυνση στο πεδίο της αναζήτησης και ο άλλος είναι πατώντας με το δάχτυλο στην οθόνη της συσκευής τον συγκεκριμένο δρόμο. Στην πρώτη περίπτωση αποστέλλεται ένα API στην Mapbox με το access token και την διεύθυνση για την εύρεσή της στον χάρτη και τον σχεδιασμό διαδρομής προς αυτήν ενώ στην δεύτερη περίπτωση με βάση το γεωγραφικό μήκος και πλάτος της τοποθεσίας υπολογίζεται η διεύθυνση και στην συνέχεια παράγεται η διαδρομή. Τα δεδομένα από το Mapbox Geocoding API response αποθηκεύονται στην κλάση CarmenFeature όπου γενικά ρυθμίζει τα markers (στάσεις) κάθε τοποθεσίας, το ύψος της κάμερας που δείχνει στο marker και την μορφή/σχήμα που θα έχει κάθε marker.

Αφού ολοκληρωθεί η εισαγωγή των διευθύνσεων, αποθηκεύονται σε μια λίστα οι συντεταγμένες όλων των στάσεων. Έπειτα, ένα ακόμη thread αναλαμβάνει την σχεδίαση της τελικής διαδρομής σύμφωνα με την λίστα. Πιο αναλυτικά, δέχεται ως παράμετρο τις συντεταγμένες του σημείου αφετηρίας και την λίστα, στέλνει στην Mapbox ένα API με αυτές τις παραμέτρους και το access token και ως response λαμβάνει την διαδρομή.

Τέλος, για την έναρξη λήψης οδηγιών της διαδρομής αποστέλλεται ένα API στο αντίστοιχο service της Mapbox με παραμέτρους τις συντεταγμένες της αφετηρίας και κάθε στάσης. Για την αποθήκευση της αφετηρίας και των ενδιάμεσων στάσεων στην βάση δεδομένων ακολουθήθηκε η εξής διαδικασία: Για κάθε σημείο, με την χρήση του Geocoder, εξάγεται η διεύθυνση του. Οι διευθύνσεις αυτές αποθηκεύονται σε μια λίστα ενώ σε μια επιπλέον λίστα αποθηκεύονται οι συντεταγμένες τους (γεωγραφικό μήκος, γεωγραφικό πλάτος). Με την έναρξη της πλοήγησης από τον χρήστη, μέσω ενός HTTP POST request στέλνεται στον server ένα API για την αποθήκευση στο collection missions της βάσης δεδομένων του id του χρήστη, του αριθμού πινακίδας του οχήματος που πραγματοποιεί το δρομολόγιο, της διεύθυνσης του σημείου αφετηρίας και των συντεταγμένων του σημείου αφετηρίας (τα υπόλοιπα πεδία αποστέλλονται με null τιμές). Συνεχίζοντας και μέχρι το τέλος της πλοήγησης, κάθε φορά που φτάνει σε μια στάση η εφαρμογή στέλνει μέσω API ένα HTTP POST request στον server με το body του request να αποτελείται από το id του χρήστη, τον αριθμού πινακίδας του οχήματος, την διεύθυνση του σημείου αφετηρίας, την διεύθυνση της στάσης, τις συντεταγμένες της στάσης, την τρέχουσα κατάσταση του δρομολογίου (αν ολοκληρώθηκε ή όχι) και την ημερομηνία και ώρα ολοκλήρωσης της διαδρομής (αν έχει ολοκληρωθεί). Τα δεδομένα αποστέλλονται σε μορφή JSON με την βοήθεια της POJO κλάσης

PostMission. Από τον sever επιστρέφεται HTTP 200 response σε HTML μορφή που τον ενημερώνει για πιθανό σφάλμα κατά την αποθήκευση των πληροφοριών.

- **Register.java**

Το αρχείο αυτό περιλαμβάνει την σελίδα εγγραφής ενός χρήστη. Αφού συμπληρώσει ο ίδιος το όνομα του, το επώνυμο του, το email του, τον κωδικό πρόσβασης στην εφαρμογή, την επιβεβαίωση κωδικού πρόσβασης και τον αριθμό επαναφοράς οι πληροφορίες αυτές αποστέλλονται σε μορφή JSON μέσω της κλάσης PostRegisterUser στον server μέσω HTTP POST request χάρη στο αντίστοιχο API. Ο server επιστρέφει ένα HTTP 200 response σε HTML μορφή που ενημερώνει τον client για την επιτυχή καταχώρηση των δεδομένων στο users collection ή για πιθανό σφάλμα κατά την αποθήκευση των πληροφοριών.

- **RegisterVehicle.java**

Η εγγραφή ενός οχήματος στο σύστημα πραγματοποιείται μέσα από αυτό το αρχείο. Ο χρήσης αφού ολοκληρώσει την καταχώρηση των προσωπικών του πληροφοριών οφείλει να εισάγει και τα δεδομένα του οχήματος του προκειμένου να ολοκληρώσει επιτυχώς την διαδικασία εγγραφής. Σε αυτό το σημείο στέλνει ένα HTTP GET request στον server προκειμένου να λάβει το id του χρήστη που θα καταχωρηθεί στα στοιχεία του οχήματος. Η GET μέθοδος δέχεται ως παράμετρο το email που είχε εισάγει ο χρήστης στην σελίδα εγγραφής του. Στη συνέχεια εισάγει τον κατασκευαστή του οχήματος, το μοντέλο του, τον αριθμό πινακίδας του, τον τύπο οχήματος, το χρώμα του, τα κυβικά του, την ιπποδύναμή του και την ημερομηνία κατασκευής του. Έπειτα υλοποιείται μια HTTP POST μέθοδος για την αποστολή των δεδομένων στον server. Τα δεδομένα μετατρέπονται σε JSON μορφή μέσω την κλάσης PostRegisterVehicle. Ο server επιστρέφει ένα HTTP 200 response σε HTML μορφή που ενημερώνει τον client για την επιτυχή καταχώρηση των δεδομένων στο vehilces collection ή για πιθανό σφάλμα κατά την αποθήκευση των πληροφοριών. Σε περίπτωση σφάλματος, για την αποτροπή δημιουργίας προφίλ χρήστη χωρίς όχημα αποστέλλεται μια HTTP DELETE μέθοδος που διαγράφει από το users collection τον χρήστη που δημιουργήθηκε στην προηγούμενη σελίδα.

## 4.4 Υλοποίηση Web εφαρμογής

Η ενότητα αυτή αφορά τον τρόπο υλοποίησης την web εφαρμογής. Πιο συγκεκριμένα, εξηγείται ο τρόπος ανάπτυξης κάθε σελίδας, τον ρόλο της βιβλιοθήκης jQuery για την αλληλεπίδραση με τον server και η χρησιμότητα της μεθόδου Ajax για αποστολή δεδομένων στον server και λήψη δεδομένων από αυτόν χωρίς να χρειάζεται ανανέωση η σελίδα για την προβολή των νέων δεδομένων.

- **views/login\_fields.ejs**

Το αρχείο αυτό αναπαράγει τον HTML/Javascript κώδικα της σελίδας εισόδου της εφαρμογής. Με την συμπλήρωση του email και κωδικού πρόσβασης από τον χρήστη, μια HTML form POST μέθοδος αποστέλλει στον server τα δεδομένα σε μορφή x-www-form-urlencoded για την πιστοποίησή του. Με την πιστοποίηση του client, ο server αποστέλλει ένα cookie στον client το οποίο αποθηκεύεται στο πρόγραμμα περιήγησης και χρησιμοποιείται για την αναγνώριση του χρήστη σε κάθε request που πραγματοποιεί.

Εκτός από την σύνδεση στην εφαρμογή, ο χρήστης μπορεί να πραγματοποιήσει επαναφορά κωδικού πρόσβασης σε περίπτωση που έχει ξεχάσει τον τρέχον. Βασική προϋπόθεση αποτελεί η γνώση του email και του κωδικού επαναφοράς. Πιο αναλυτικά, ο χρήστης εισάγει το email του, τον αριθμό επαναφοράς, τον νέο κωδικό πρόσβασης και την επιβεβαίωση κωδικού πρόσβασης στα αντίστοιχα πεδία και μέσω της jQuery Ajax μεθόδου έχει υλοποιηθεί μια HTTP POST μέθοδος η οποία στέλνει ένα request μέσω του API στον server με αυτά τα δεδομένα σε x-www-form-urlencoded μορφή λαμβάνοντας ένα HTTP response (200 OK) που ενημερώνει τον client με μήνυμα για την επιτυχή αλλαγή του κωδικού σε μορφή HTML.

- **views/redirect\_progress.ejs**

Μια μεγάλη διαφορά της web με την android εφαρμογή είναι ότι στην πρώτη υπάρχει η επιλογή εισόδου για απλό χρήστη ή διαχειριστή. Μετά την επιτυχή είσοδο του client, αποστέλλεται ένα ακόμη HTTP GET request μέσω Ajax για την λήψη της κατηγορίας λογαριασμού (χρήστης/διαχειριστής) με βάση το email. Το email περιλαμβάνεται στο URL του API και για HTTP 200 response επιστρέφεται σε JSON μορφή η κατηγορία λογαριασμού.

- **views/admin\_main\_work.ejs**

Αυτό το αρχείο δημιουργεί την αρχική σελίδα του διαχειριστή. Αρχικά, με την είσοδο του client στην εφαρμογή, μια HTTP GET μέθοδος αποστέλλει ένα request στον server και λαμβάνει σε JSON μορφή το id του στέλνοντας ως παράμετρο στο URL του API το email του. Σε περίπτωση που επιστραφεί HTTP 400 ενημερώνεται ο χρήστης μέσω popup μηνύματος με την χρήση της Javascript βιβλιοθήκης SweetAlert. Η διαδικασία αυτή πραγματοποιείται για την λήψη του ονόματος και του επωνύμου του χρήστη που απαιτεί την γνώση του id του.

Αφού ληφθεί το id, μια δεύτερη Ajax GET μέθοδος αποστέλλει το id με τον ίδιο τρόπο με την πρώτη μέθοδο και λαμβάνει ένα HTTP 200 response με το ονοματεπώνυμο του χρήστη σε μορφή JSON. Ακόμη, με την βοήθεια της βιβλιοθήκης jQuery και συγκεκριμένα της συνάρτησης load, σε ένα μέρος της σελίδας (element) φορτώνονται από τον server σελίδες που έχει επιλέξει ο χρήστης χωρίς να επηρεάζονται τα υπόλοιπα στοιχεία της σελίδας. Οι σελίδες που μπορεί να φορτώσει η συνάρτηση load είναι:

- Σελίδα προφίλ οδηγού

- Σελίδα προφίλ οχημάτων οδηγού
- Σελίδα δρομολογίων οδηγού
- Σελίδα συνόλου χρηστών
- Σελίδα συνόλου οχημάτων
- Σελίδα συνόλου δρομολογίων

- **views/user\_main\_work.ejs**

Η σελίδα που προκύπτει από το αρχείο αυτό είναι ακριβώς η ίδια με την σελίδα διαχειριστή με μια σημαντική διαφορά. Η συνάρτηση load της βιβλιοθήκης jQuery φορτώνει μόνο τις σελίδες που έχει δικαίωμα εισόδου ο απλός χρήστης. Πιο αναλυτικά, οι σελίδες αυτές είναι:

- Σελίδα προφίλ οδηγού
- Σελίδα προφίλ οχημάτων οδηγού
- Σελίδα δρομολογίων οδηγού

- **views/getAccount.ejs**

Η σελίδα αυτή αποτελεί την σελίδα προφίλ οδηγού. Εδώ ο χρήστης μπορεί να δει τις προσωπικές του πληροφορίες, να τις επεξεργαστεί, να αλλάξει κωδικό πρόσβασης ή να διαγράψει τον λογαριασμό του.

Αρχικά η λήψη των προσωπικών του πληροφοριών επιτυγχάνεται με μια GET μέθοδος που παίρνει ως παράμετρο στο URL του API το id του χρήστη και λαμβάνει από τον server ένα HTTP 200 response με το body του να περιλαμβάνει το email, το όνομα, το επώνυμο και την ημερομηνία εγγραφή του σε μορφή JSON. Αν υπάρξει σφάλμα στην λήψη των δεδομένων (400 response) εμφανίζεται popup μήνυμα.

Για την επεξεργασία των δεδομένων, ο χρήστης αφού εισάγει στα αντίστοιχα πεδία τις πληροφορίες που επιθυμεί να αλλάξουν (όνομα, επώνυμο, email) στη συνέχεια αποστέλλεται στον server ένα Ajax POST request σε μορφή x-www-form-urlencoded. Αν το request πραγματοποιηθεί με επιτυχία (HTTP 200) εμφανίζεται popup μήνυμα με το response που θα λάβει από το server.

Παρόμοια διαδικασία ακολουθείται και για την αλλαγή του κωδικού πρόσβασης. Μια Ajax POST μέθοδος που στο body του request περιλαμβάνεται το email του χρήστη, τον τρέχον κωδικό πρόσβασης, τον νέο κωδικό πρόσβασης και την επιβεβαίωση κωδικού αποστέλλεται στον server σε μορφή x-www-form-urlencoded. Ως response επιστρέφεται σε HTML μορφή ένα μήνυμα για την ενημέρωση του χρήστη σχετικά με την επιτυχία ή αποτυχία του request.

Τέλος, για την διαγραφή του λογαριασμού, έχει υλοποιηθεί μια Ajax DELETE μέθοδος που στέλνει μέσω API στον server ένα request το οποίο περιέχει στο body του το id του χρήστη. Αν το request πραγματοποιηθεί με επιτυχία (HTTP 200) εμφανίζεται popup μήνυμα με το response που θα λάβει από το server.

- **views/getVehicle.ejs**

Η σελίδα προφίλ των οχημάτων του οδηγού δημιουργείται από αυτό το αρχείο. Σε αυτή την σελίδα ο χρήστης μπορεί να δει τις πληροφορίες των οχημάτων του, να τις επεξεργαστεί, να προσθέσει ένα καινούργιο όχημα, να αναζητήσει ένα συγκεκριμένο όχημα από την λίστα του συνόλου και να διαγράψει όποιο από αυτά επιθυμεί.

Η λήψη των πληροφοριών των οχημάτων επιτυγχάνεται με μια GET μέθοδος που παίρνει ως παράμετρο στο URL του API το id του χρήστη και λαμβάνει από τον server ένα HTTP 200 response με το body του να περιλαμβάνει τον κατασκευαστή του κάθε οχήματος, το μοντέλο του, τον αριθμό πινακίδας του, τον τύπο οχήματος, το χρώμα του, τα κυβικά του, την ιπποδύναμή του και την ημερομηνία κατασκευής του σε μορφή JSON. Τα δεδομένα αποθηκεύονται και προβάλλονται μέσα σε πίνακα. Αν υπάρξει σφάλμα στην λήψη των δεδομένων (400 response) εμφανίζεται popup μήνυμα.

Για την επεξεργασία ενός οχήματος, ο χρήστης αφού εισάγει στα αντίστοιχα πεδία τις πληροφορίες που επιθυμεί να αλλάξουν (αριθμό πινακίδας, κατασκευαστή, μοντέλο, χρώμα, κυβικά, ιπποδύναμη) στη συνέχεια αποστέλλεται στον server ένα Ajax POST request σε μορφή x-www-form-urlencoded. Αν το request πραγματοποιηθεί με επιτυχία (HTTP 200) εμφανίζεται popup μήνυμα με το response που θα λάβει από το server. Για την εμφάνιση των αλλαγών πραγματοποιείται αυτόματη ανανέωση της σελίδας.

Η προσθήκη ενός νέου οχήματος απαιτεί την συμπλήρωση από τον χρήστη των πεδίων του κατασκευαστή του οχήματος, του μοντέλου του, του αριθμού πινακίδας του, του τύπου οχήματος, το χρώματος του, των κυβικών του, της ιπποδύναμή του και της ημερομηνία κατασκευής του. Έπειτα, μια POST μέθοδος αποστέλλεται μέσω API στον server με τα δεδομένα που αναφέρθηκαν να περιλαμβάνονται στο body του request για την αποθήκευση των πληροφοριών στην βάση δεδομένων. Ως response επιστρέφεται σε HTML μορφή ένα μήνυμα για την ενημέρωση του χρήστη σχετικά με την επιτυχία ή αποτυχία του request.

Επιπρόσθετα, για την διαγραφή ενός οχήματος, έχει υλοποιηθεί μια Ajax DELETE μέθοδος που στέλνει μέσω API στον server ένα request το οποίο περιέχει στο body του τον αριθμό πινακίδας του οχήματος προς διαγραφή. Ως response είτε επιτυχής (200 OK) είτε ανεπιτυχής επιστρέφεται σε HTML μορφή ένα μήνυμα για την ενημέρωση του χρήστη.

Το γεγονός ότι δεν μπορεί να υπάρξει οδηγός χωρίς όχημα καθιστά απαραίτητο τον έλεγχο του αριθμού των οχημάτων που αντιστοιχούν σε κάθε έναν. Για τον λόγο αυτό, πριν από την διαγραφή ενός οχήματος αποστέλλεται ένα HTTP GET request για να ληφθεί ο αριθμός των οχημάτων που έχει στην κατοχή του ένας χρήστης. Για να γίνει αυτό απαιτείται το id του χρήστη το οποίο “φορτώνεται” στο URL του API. Το response που επιστρέφει από τον server σε περίπτωση επιτυχίας (200 OK) είναι σε JSON μορφή.

- **views/getMission.ejs**

Σε αυτό το αρχείο έχει υλοποιηθεί ο HTML/Javascript κώδικας της σελίδας δρομολογίων του οδηγού. Εδώ χρήστης έχει την δυνατότητα να δει τις πληροφορίες των δρομολογίων που έχει ολοκληρώσει ή βρίσκονται σε εξέλιξη, να προβάλλει στον χάρτη την πορεία ενός δρομολογίου, να αναζητήσει ένα συγκεκριμένο δρομολόγιο από την λίστα του συνόλου και να διαγράψει όποιο από αυτά επιθυμεί.

Αρχικά, για την εμφάνιση όλων των δρομολογίων, μια HTTP GET μέθοδος αποστέλλει ένα request στον server περνώντας ως παράμετρο στο URL του API το id του χρήστη/οδηγού. Το response (200 OK) που λαμβάνει σε JSON μορφή περιέχει για κάθε δρομολόγιο τον αριθμό πινακίδας του οχήματος, την ημερομηνία και ώρα έναρξης, την διεύθυνση του σημείου αφετηρίας, την διεύθυνση του σημείου τερματισμού, την ημερομηνία και ώρα ολοκλήρωσης της διαδρομής (αν έχει ολοκληρωθεί) και την τρέχουσα κατάσταση του (αν ολοκληρώθηκε ή όχι). Τα δεδομένα αυτά εμφανίζονται στην σελίδα μέσα σε ένα πίνακα. Αν επιστραφεί HTTP 400 response, το body περιέχει μήνυμα για το σφάλμα σε HTML μορφή που ενημερώνει τον client τι πήγε στραβά.

Προκειμένου ο χρήστης να ανοίξει την σελίδα του χάρτη, μια Ajax GET μέθοδος παίρνει ως παράμετρο στο URL του API το id του δρομολογίου και λαμβάνει από τον server ένα HTTP 200 response με το body του να περιλαμβάνει τις συντεταγμένες του σημείου αφετηρίας και όλων των στάσεων του δρομολογίου που έχει επιλέξει να προβάλλει σε μορφή JSON.

Τέλος, για την διαγραφή ενός δρομολογίου, έχει υλοποιηθεί μια Ajax DELETE μέθοδος που στέλνει μέσω API στον server ένα request το οποίο περιέχει στο body του το id του δρομολογίου. Αν το request πραγματοποιηθεί με επιτυχία (HTTP 200) εμφανίζεται popup μήνυμα με το response που θα λάβει από το server.

- **views/getAllUsers.ejs**

Το συγκεκριμένο αρχείο καθώς και τα επόμενα δύο είναι προσβάσιμα αποκλειστικά και μόνο από τους διαχειριστές της εφαρμογής. Οι σελίδες που υλοποιούν αυτά τα αρχεία προβάλλουν δεδομένα για κάθε χρήστη που είναι εγγεγραμμένος στο σύστημα. Συνεπώς ο διαχειριστής μπορεί να επέμβει και να αλλάξει ή να διαγράψει τις πληροφορίες ενός οδηγού, ενός οχήματος ή ενός δρομολογίου.

Σε αυτή την σελίδα (σελίδα συνόλου χρηστών) εμφανίζεται ένας πίνακας με όλους τους χρήστες-οδηγούς της εφαρμογής. Η λίστα αυτή προκύπτει ύστερα από μια Ajax GET μέθοδος που αποστέλλει μέσω API ένα request για την λήψη σε περίπτωση επιτυχίας ( HTTP 200) του αντίστοιχου response σε μορφή JSON όλων των εγγράφων από την συλλογή users.

Η επεξεργασία των πληροφοριών ενός οδηγού ακολουθεί ακριβώς την ίδια διαδικασία με εκείνη του αρχείου getAccount.ejs. Συγκεκριμένα, ο διαχειριστής εισάγει στα αντίστοιχα πεδία τις πληροφορίες που επιθυμεί να αλλάξουν (όνομα, επώνυμο, email) και στη συνέχεια αποστέλλεται στον server ένα Ajax POST request σε μορφή x-www-form-urlencoded. Αν το request πραγματοποιηθεί με επιτυχία (HTTP 200) εμφανίζεται popup μήνυμα με το response που θα λάβει από το server.



Η διαγραφή ενός οδηγού από την σελίδα του διαχειριστή διαφέρει λίγο με την διαγραφή του από την σελίδα του απλού χρήστη. Ο λόγος είναι ότι για να διαγραφεί ένας οδηγός απαιτείται η γνώση του id του, μια πληροφορία που δεν υπάρχει στην λίστα (το περιεχόμενο της λίστας αποτελείται από το όνομα κάθε χρήστη, το επώνυμο του, το email του και την ημερομηνία εγγραφής του). Συνεπώς, ο διαχειριστής αφού επιλέξει τον χρήστη που επιθυμεί να διαγράψει, αποστέλλει ένα Ajax GET request στον server με το email του να περιέχεται στο URL του API. Σε HTTP 200 response, επιστρέφεται στον client το id του υποψήφιου προς διαγραφή χρήστη σε JSON μορφή. Αν υπάρξει σφάλμα στην λήψη των δεδομένων (400 response) εμφανίζεται popup μήνυμα.

Αφού ληφθεί το id του χρήστη που θα διαγραφεί από το σύστημα, έχει υλοποιηθεί μια Ajax DELETE μέθοδος που στέλνει μέσω API στον server ένα request το οποίο περιέχει στο body του το συγκεκριμένο id. Αν το request πραγματοποιηθεί με επιτυχία (HTTP 200) εμφανίζεται popup μήνυμα με το response που θα λάβει από το server. Αν υπάρξει σφάλμα (400 response) εμφανίζεται popup μήνυμα.

- **views/getAllVehicles.ejs**

Η σελίδα συνόλου οχημάτων εμφανίζει μια λίστα με όλα τα οχήματα που είναι εγγεγραμμένα στο σύστημα. Για την προβολή των οχημάτων αποστέλλεται ένα HTTP GET request μέσω API. Σε περίπτωση HTTP 200 response επιστρέφονται όλα τα έγγραφα από την συλλογή vehicles σε μορφή JSON. Οι πληροφορίες που περιλαμβάνει το body του response είναι για κάθε όχημα ο κατασκευαστής, το μοντέλο του, ο αριθμός πινακίδας του, ο τύπος του κάθε οχήματος, το χρώμα του, τα κυβικά του, η ιπποδύναμή του και η ημερομηνία κατασκευής του. Αν επιστραφεί HTTP 400 response, το body περιέχει μήνυμα για το σφάλμα σε HTML μορφή που ενημερώνει τον client μέσω popup μηνύματος τι πήγε στραβά.

Για την επεξεργασία των πληροφοριών ενός οχήματος, ο διαχειριστής αφού εισάγει στα αντίστοιχα πεδία τις πληροφορίες που επιθυμεί να αλλάξουν (αριθμό πινακίδας, κατασκευαστή, μοντέλο, χρώμα, κυβικά, ιπποδύναμή) στη συνέχεια αποστέλλεται στον server ένα Ajax POST request σε μορφή x-www-form-urlencoded. Αν το request πραγματοποιηθεί με επιτυχία (HTTP 200) εμφανίζεται popup μήνυμα με το response που θα λάβει από το server. Για την εμφάνιση των αλλαγών πραγματοποιείται αυτόματη ανανέωση της σελίδας.

Η διαγραφή ενός οχήματος χωρίζεται σε δύο στάδια. Στο πρώτο στάδιο πραγματοποιείται έλεγχος αν ο οδηγός που κατέχει το συγκεκριμένο όχημα διαθέτει τουλάχιστον ένα ακόμη (δεν μπορεί να υπάρξει οδηγός χωρίς όχημα). Λόγω αυτής της συνθήκης, αποστέλλεται ένα HTTP GET request για να ληφθεί ο αριθμός των οχημάτων που διαθέτει ο κάτοχος του οχήματος. Για να γίνει αυτό απαιτείται το id του οδηγού το οποίο “φορτώνεται” στο URL του API. Το response που επιστρέφει από τον server σε περίπτωση επιτυχίας (200 OK) είναι σε JSON μορφή.

Στο δεύτερο στάδιο, αφού επιβεβαιωθεί ότι ο οδηγός κατέχει τουλάχιστον ένα όχημα, έχει υλοποιηθεί μια Ajax DELETE μέθοδος που στέλνει μέσω API στον server ένα request το οποίο περιέχει στο body του τον αριθμό πινακίδας του οχήματος προς διαγραφή. Ως response είτε επιτυχής (200 OK) είτε ανεπιτυχής επιστρέφεται σε HTML μορφή ένα μήνυμα για την ενημέρωση του χρήστη.

- **views/getAllMissions.ejs**

Η σελίδα συνόλου των δρομολογίων δημιουργείται από αυτό το αρχείο. Η μοναδική διαφορά της με την σελίδα δρομολογίων του χρήστη είναι ότι η λίστα δρομολογίων περιέχει τα δρομολόγια όλων των οχημάτων. Επιπλέον ο διαχειριστής μπορεί να διαγράψει τις πληροφορίες ενός δρομολογίου που πραγματοποίησε άλλος χρήστης ή διαχειριστής.

Η λίστα δρομολογίων προκύπτει μετά από μια Ajax GET μέθοδος που αποστέλλει μέσω API ένα request για την λήψη όλων των εγγράφων από την συλλογή missions σε μορφή JSON (HTTP 200 response). Στο body του response περιέχεται για κάθε δρομολόγιο ο αριθμός πινακίδας του οχήματος, η ημερομηνία και ώρα έναρξης, η διεύθυνση του σημείου αφετηρίας, η διεύθυνση του σημείου τερματισμού, η ημερομηνία και ώρα ολοκλήρωσης της διαδρομής (αν έχει ολοκληρωθεί) και η τρέχουσα κατάσταση του (αν ολοκληρώθηκε ή όχι). Σε περίπτωση HTTP 400 response, το body περιέχει μήνυμα για το σφάλμα σε HTML μορφή που ενημερώνει τον client μέσω popup μηνύματος τι πήγε στραβά.

Οι υπόλοιπες μέθοδοι είναι ακριβώς ίδιες με αυτές της σελίδας δρομολογίων του χρήστη. Αν ο διαχειριστής επιθυμεί να ανοίξει την σελίδα του χάρτη για ένα συγκεκριμένο δρομολόγιο, μια Ajax GET μέθοδος παίρνει ως παράμετρο στο URL του API το id του δρομολογίου και λαμβάνει από τον server ένα HTTP 200 response με το body του να περιλαμβάνει τις συντεταγμένες του σημείου αφετηρίας και όλων των στάσεων του δρομολογίου που έχει επιλέξει να προβάλλει σε μορφή JSON.

Τέλος, για την διαγραφή ενός δρομολογίου, έχει υλοποιηθεί μια Ajax DELETE μέθοδος που στέλνει μέσω API στον server ένα request το οποίο περιέχει στο body του το id του δρομολογίου. Αν το request πραγματοποιηθεί με επιτυχία (HTTP 200) εμφανίζεται popup μήνυμα με το response που θα λάβει από το server.

- **views/getMap.ejs**

Όπως αναφέρθηκε και στις σελίδες δρομολογίων, υπάρχει η επιλογή εμφάνισης ενός δρομολογίου στον χάρτη. Το αρχείο αυτό υλοποιεί την σελίδα του χάρτη. Πρώτα απ' όλα για την εμφάνιση του χάρτη και της διαδρομής απαιτείται η επικοινωνία με την Mapbox η οποία πραγματοποιείται μέσω API. Το API απαιτεί ένα access token για την πιστοποίηση του χρήστη που χρησιμοποιεί τα services της εταιρείας. Έχοντας ληφθεί οι συντεταγμένες ενός δρομολογίου, αποστέλλεται ένα request στην Mapbox με το access token και τις συντεταγμένες κάθε στάσης για την εμφάνιση των στάσεων στον χάρτη και την αναπαράσταση της διαδρομής.

- **views/register\_user.ejs**

Σε περίπτωση που ένας οδηγός δεν έχει λογαριασμό μπορεί να δημιουργήσει έναν μέσα από την σελίδα που υλοποιείται από αυτό το αρχείο. Πιο αναλυτικά, ο χρήστης εισάγει το όνομα του, το επώνυμο του, την κατηγορία λογαριασμού, το email του, τον κωδικό πρόσβασης στην εφαρμογή, την επιβεβαίωση κωδικού πρόσβασης και τον αριθμό επαναφοράς στα αντίστοιχα πεδία και μέσω της Ajax μεθόδου έχει υλοποιηθεί μια HTTP POST μέθοδος η οποία στέλνει ένα request μέσω του API

στον server με αυτά τα δεδομένα σε x-www-form-urlencoded μορφή λαμβάνοντας ένα HTTP response που ενημερώνει τον client με μήνυμα σε μορφή HTML για την επιτυχή ή ανεπιτυχή εγγραφή των πληροφοριών χρήστη.

- **views/register\_vehicle.ejs**

Ύστερα από την επιτυχή αποθήκευση των πληροφοριών του χρήστη στην βάση δεδομένων, σειρά έχει η εγγραφή ενός οχήματος για αυτό τον χρήστη. Για την εγγραφή ενός οχήματος απαιτείται το όνομα του κατασκευαστή, το μοντέλο του, ο αριθμός πινακίδας του, ο τύπος του, το χρώμα του, τα κυβικά του, η ιπποδύναμή του και η ημερομηνία κατασκευής του. Αφού συμπληρωθούν αυτά τα πεδία, αποστέλλεται ένα HTTP POST request σε μορφή x-www-form-urlencoded στον server για την αποθήκευση των πληροφοριών στην βάση δεδομένων. Αν το request πραγματοποιηθεί με επιτυχία (HTTP 200) εμφανίζεται popup μήνυμα με το response που θα λάβει από το server ανακατευθύνοντας τον χρήστη στην σελίδα εισόδου.



# Κεφάλαιο 5: Οδηγός Χρήσης του Android app

---

## 5.1 Εισαγωγή

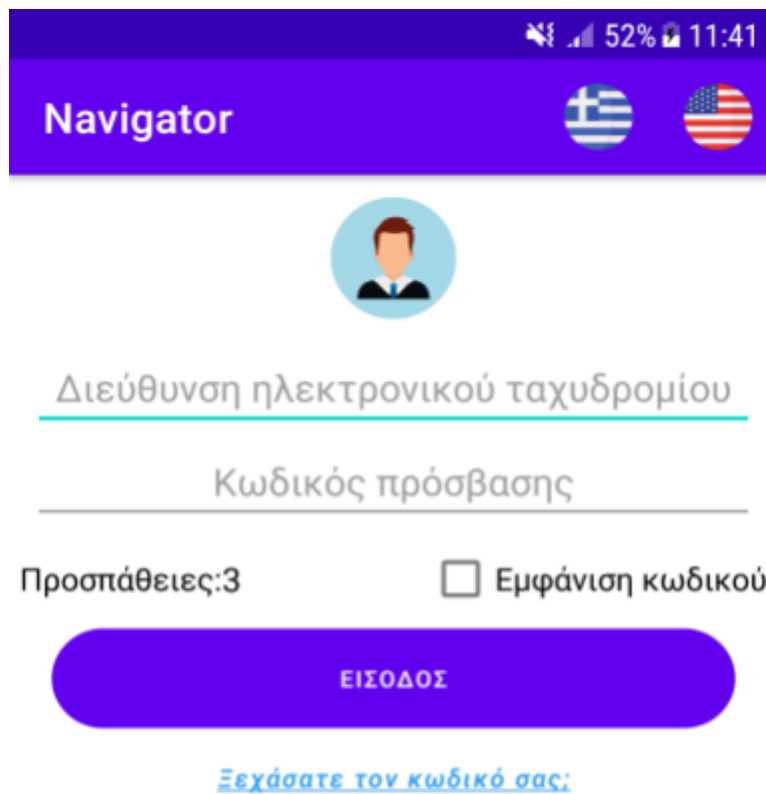
Η συγκεκριμένη εφαρμογή που υλοποιήθηκε στα πλαίσια την διπλωματικής εργασίας απευθύνεται σε μικρομεσαίες επιχειρήσεις στον τομέα της διανομής. Δίνει την δυνατότητα στον κάθε οδηγό να δημιουργήσει την δική του διαδρομή χωρίς περιορισμό στον αριθμό των ενδιάμεσων στάσεων. Η εφαρμογή για κάθε δρομολόγιο που έχει οριστεί παρέχει στον οδηγό χρήσιμες πληροφορίες όπως την συνολική απόσταση που έχει να διανύσει σε χιλιόμετρα και μια μέση εκτίμηση του χρόνου που θα χρειαστεί για να ολοκληρωθεί το δρομολόγιο. Αξίζει να σημειώσουμε ότι ο κάθε οδηγός έχει το δικαίωμα κατοχής περισσότερων από ένα οχημάτων. Γενικά, ο χρήστης μπορεί να αποκτήσει πρόσβαση στην εφαρμογή είτε κάνοντας λήψη του αρχείου εγκατάστασης στην Android συσκευή του είτε μπαίνοντας στην ιστοσελίδα από τον υπολογιστή του. Βασική προϋπόθεση για την ορθή λειτουργία της αποτελεί η ύπαρξη σύνδεσης στο διαδίκτυο. Ιδιαίτερα όσον αφορά την Android εφαρμογή, ο συνδυασμός σύνδεσης στο internet και λειτουργίας του GPS είναι μονόδρομος για την επικοινωνία με τον server και τον προσδιορισμό της τοποθεσίας της συσκευής, της ταχύτητας με την οποία κινείται το όχημα και της κατεύθυνσης του.

Αναπτύχθηκε σε Java και το λογισμικό που χρησιμοποιήθηκε για την ανάπτυξη της είναι το Android Studio, το πιο δημοφιλή πρόγραμμα ανάπτυξης εφαρμογών για Android συσκευές. Πιο συγκεκριμένα το Android Studio αποτελεί ένα ολοκληρωμένο περιβάλλον ανάπτυξης (*IDE*) για εφαρμογές Android [32]. Για την υποστήριξη της ανάπτυξης των εφαρμογών το συγκεκριμένο λογισμικό χρησιμοποιεί τέσσερα βασικά στοιχεία:

- Πρότυπα κώδικα
- Εξομοιωτή (emulator)
- Υποστήριξη και ενσωμάτωση του GitHub
- Σύστημα κατασκευής που βασίζεται στο εργαλείο gradle

## 5.2 Είσοδος χρήστη

Στην σελίδα εισόδου όπως βλέπουμε και στην εικόνα 9, αρχικά εμφανίζονται τα πεδία του email και του κωδικού πρόσβασης όπου κάθε χρήστης θα πρέπει να γνωρίζει για την επιτυχή είσοδό του στο σύστημα. Μετά την εισαγωγή των στοιχείων του θα πρέπει να πατήσει το κουμπί “ΕΙΣΟΔΟΣ” για τη εισαγωγή του στο σύστημα. Πέρα από την δυνατότητα εισόδου, ο χρήστης μπορεί σε περίπτωση που δεν έχει λογαριασμό να πατήσει στο κουμπί “Εγγραφή τώρα!” που φαίνεται στο κάτω μέρος της οθόνης όπου θα τον ανακατευθύνει σε άλλη σελίδα για την δημιουργία καινούργιου λογαριασμού.



Δεν έχετε λογαριασμό; [Εγγραφή τώρα!](#)

Εικόνα 9: Είσοδος χρήστη στο Android App

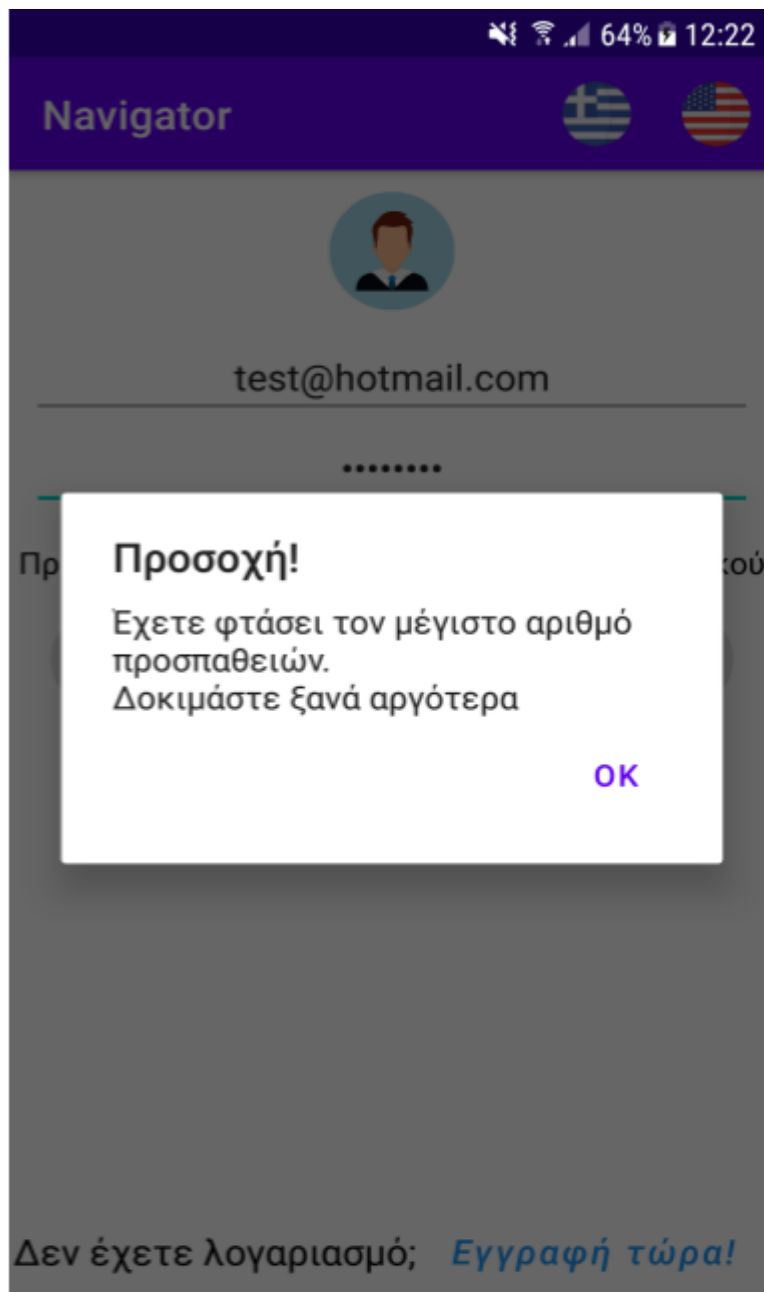
Ο κάθε χρήστης έχει το δικαίωμα να κάνει το πολύ 3 φορές λάθος στην εισαγωγή των στοιχείων του. Οι περιπτώσεις που το σύστημα ορίζει ως λάθος στην εισαγωγή των στοιχείων είναι οι εξής:

- **Κενά πεδία**
- **Λανθασμένη μορφή email.** Το email θα πρέπει να έχει την σωστή μορφή που απαρτίζεται από 4 βασικά κομμάτια:
  - Όνομα χρήστη (Username)
  - Το σύμβολο @

- Όνομα τομέα (Domain name)
- Το σύμβολο .
- **Λανθασμένη μορφή κωδικού πρόσβασης.** Ο κωδικός πρόσβασης θα πρέπει να ξεκινάει με χαρακτήρα και να αποτελείται από τουλάχιστον 7 και το πολύ 14 χαρακτήρες.
- **Λανθασμένος κωδικός πρόσβασης.** Το email είναι σωστό αλλά ο κωδικός πρόσβασης λάθος.
- **Λανθασμένο email και κωδικός πρόσβασης.** Το email και ο κωδικός πρόσβασης δεν αντιστοιχούν σε κάποιο λογαριασμό.

Ο χρήστης εισάγει τα δεδομένα, η εφαρμογή επικοινωνεί με τον server στον οποίο πραγματοποιείται ο έλεγχος εγκυρότητας των στοιχείων και στην περίπτωση που τα στοιχεία είναι έγκυρα, ο server αποστέλλει τα δεδομένα στην βάση δεδομένων με σκοπό την επιτυχή αναγνώριση και είσοδο του στο σύστημα. Στις περιπτώσεις λανθασμένου κωδικού πρόσβασης ή λανθασμένου email και κωδικού πρόσβασης, ο server αποστέλλει τα δεδομένα στην βάση και στην συνέχεια αφού δεν ταιριάζουν με κάποιον λογαριασμό επιστρέφουν στην εφαρμογή σχετικό μήνυμα.

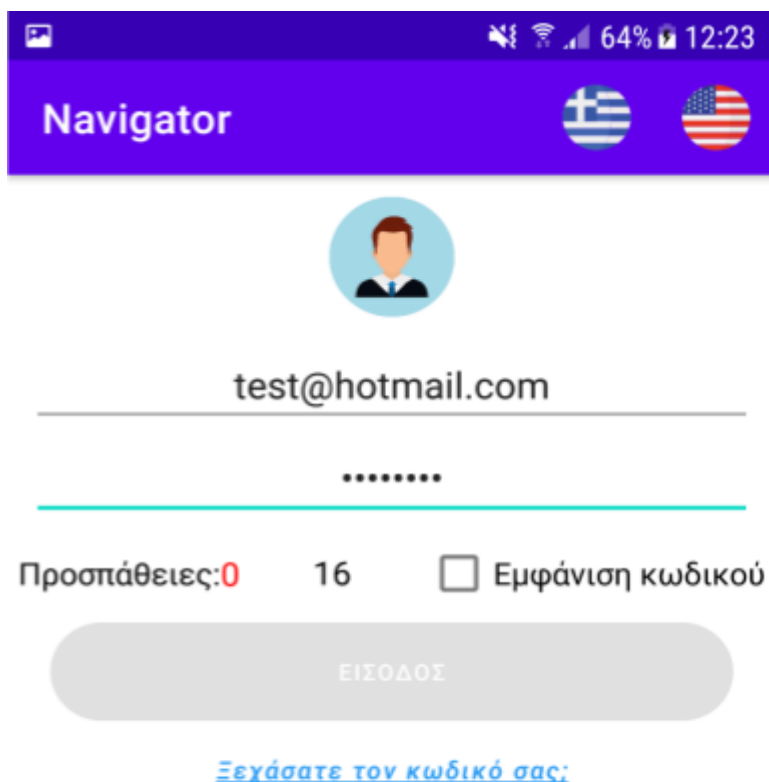
Σε περίπτωση λάθους για 4η φορά εφαρμόζεται μια προσωρινή απενεργοποίηση του κουμπιού εισόδου στο σύστημα για 30 δευτερόλεπτα όπως φαίνεται και στην εικόνα 10. Δεν επηρεάζεται το προφίλ του χρήστη ή τα στοιχεία του που είναι αποθηκευμένα στην βάση δεδομένων.



*Εικόνα 10: Μήνυμα σφάλματος μετά την εισαγωγή λανθασμένων στοιχείων σύνδεσης για 4η φορά στο Android app*

Στην εικόνα 11 βλέπουμε ένα στιγμιότυπο μετά την λανθασμένη εισαγωγή στοιχείων για 4η φορά όπου ο αριθμός των προσπαθειών που απέμειναν είναι 0 και ο χρόνος που απομένει για την ενεργοποίηση της επιλογής εισόδου στην εφαρμογή είναι 16 δευτερόλεπτα. Μετά το πέρας των 30 δευτερολέπτων, ο χρήστης έχει το δικαίωμα να προσπαθήσει ξανά ακόμη 3 φορές μέχρι να του εμφανιστεί πάλι το ίδιο μήνυμα.

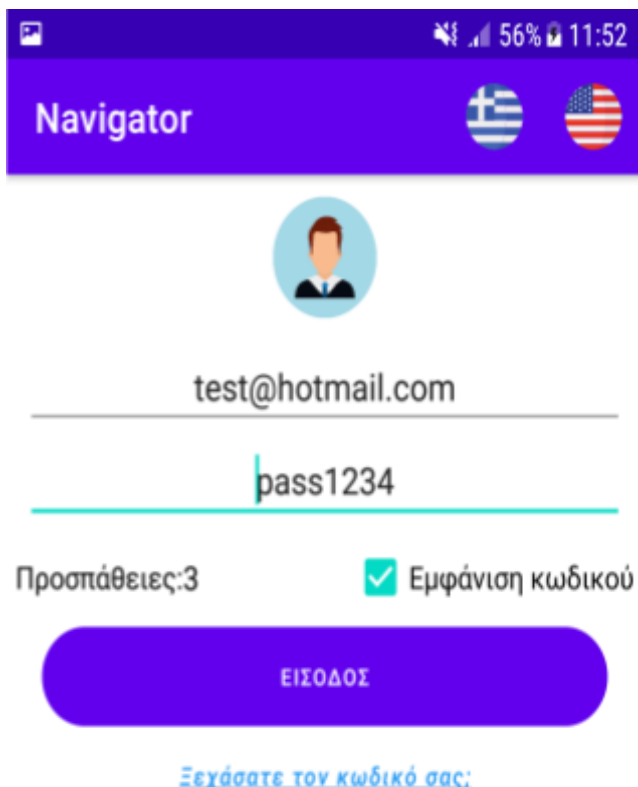
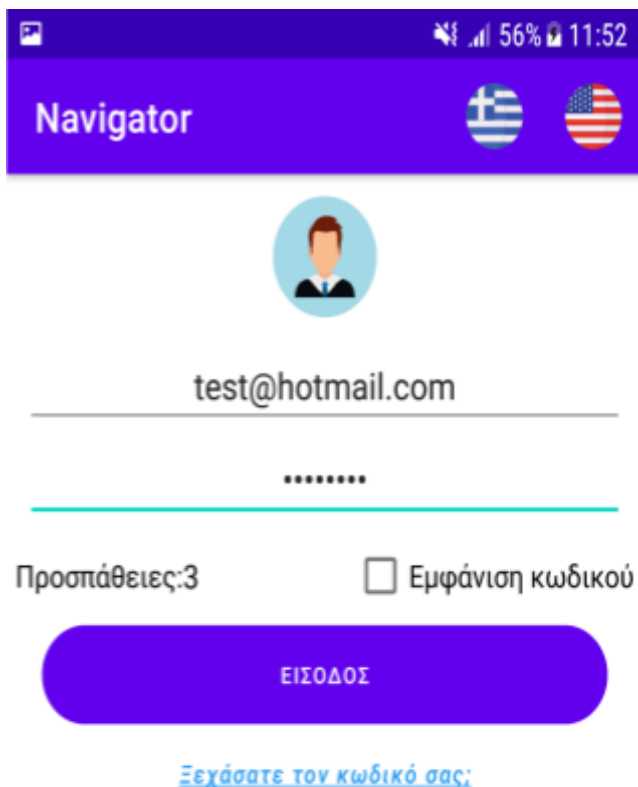




**Δεν έχετε λογαριασμό; [Εγγραφή τώρα!](#)**

*Εικόνα 11: Αντίστροφη μέτρηση για την επανενεργοποίηση του κουμπιού εισόδου στο Android App*

Όπως μπορούμε να παρατηρήσουμε και από την εικόνα 11, ο κωδικός πρόσβασης που πληκτρολόγησε ο χρήστης δεν εμφανίζεται όπως ακριβώς τον έχει εισάγει ο ίδιος. Αντιθέτως εμφανίζεται με την μορφή κουκίδων. Αυτό αποτελεί ένα ελάχιστο μέτρο ασφάλειας έτσι ώστε να μην είναι ορατός ο κωδικός είτε σε κάποιον που καταγράφει την οθόνη της συσκευής είτε σε αδιάκριτα μάτια τρίτων στο περιβάλλον που βρίσκεται ο χρήστης. Δεν πρέπει να ξεχνάμε ότι για την ασφάλειά μας απαγορεύεται αυστηρά η δημόσια έκθεση τόσο σημαντικών δεδομένων όπως είναι το email μας και οι κωδικοί πρόσβασης. Ωστόσο, η εφαρμογή παρέχει την δυνατότητα εμφάνισης του κωδικού πρόσβασης στον χρήστη πατώντας στο κουτάκι “Εμφάνιση κωδικού”. Παρακάτω εμφανίζονται 2 εικόνες όπου στην πρώτη ο χρήστης δεν επιθυμεί την εμφάνιση του κωδικού που μόλις πληκτρολόγησε ενώ στην δεύτερη ο κωδικός εμφανίζεται.



Δεν έχετε λογαριασμό; [Εγγραφή τώρα!](#)

Δεν έχετε λογαριασμό; [Εγγραφή τώρα!](#)

Εικόνα 12: Απόκρυψη κωδικού πρόσβασης κατά την είσοδο στο Android App

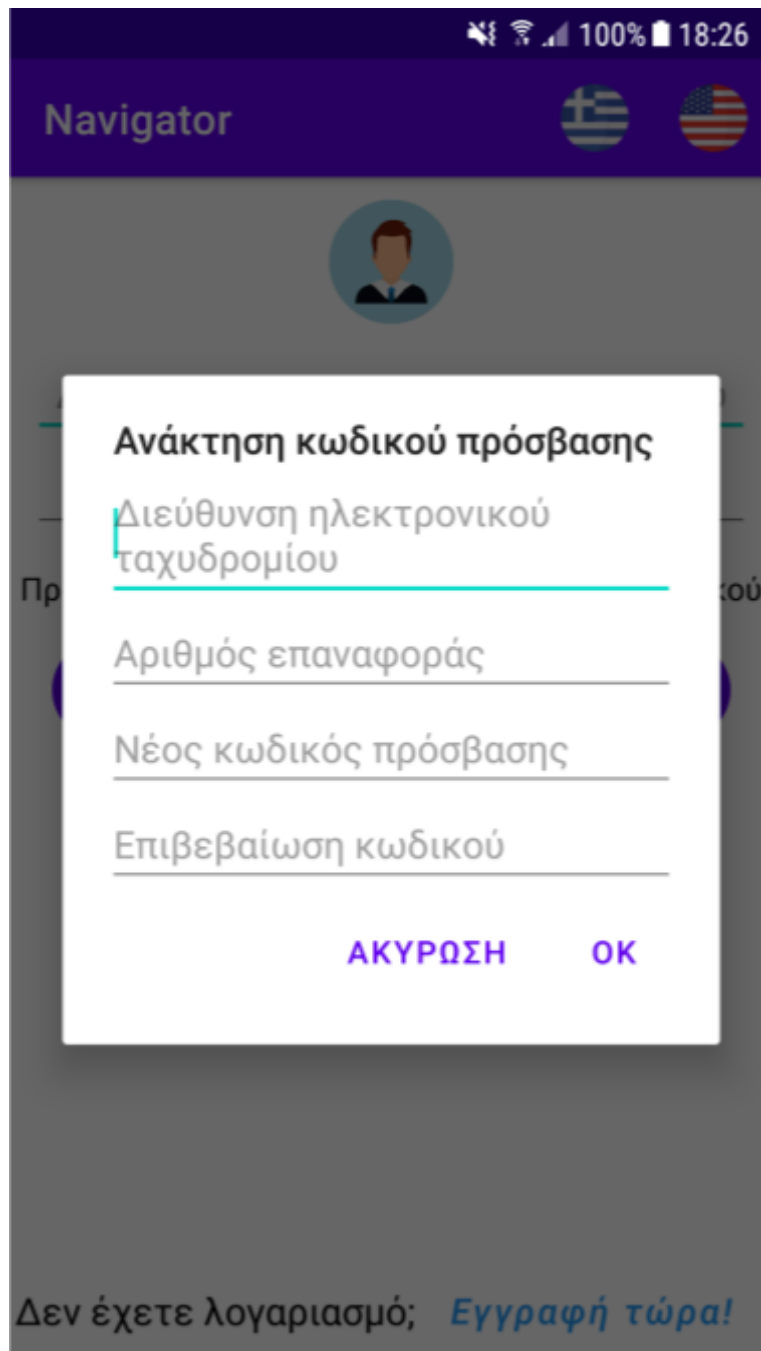
Εικόνα 13: Εμφάνιση κωδικού πρόσβασης κατά την είσοδο στο Android App

Σε περίπτωση που ο χρήστης ξεχάσει τον κωδικό πρόσβασης του έχει την δυνατότητα να ορίσει έναν νέο κωδικό μέσω μιας πολύ απλής διαδικασίας. Το μόνο που απαιτείται είναι η γνώση του email του και του αριθμού επαναφοράς (PIN), ενός τετραψήφιου αριθμού που ορίζει ο ίδιος κατά την δημιουργία του λογαριασμού του. Βασικές προϋποθέσεις για την επιτυχή αλλαγή του κωδικού πρόσβασης είναι:

- Εισαγωγή email σε σωστή μορφή.
- Εισαγωγή αριθμού επαναφοράς που αντιστοιχεί στο σωστό email.

- Εισαγωγή κωδικού πρόσβασης που να ξεκινάει με χαρακτήρα και να αποτελείται από τουλάχιστον 7 και το πολύ 14 χαρακτήρες.
- Τα πεδία των κωδικών πρόσβασης να περιέχουν τον ίδιο κωδικό.

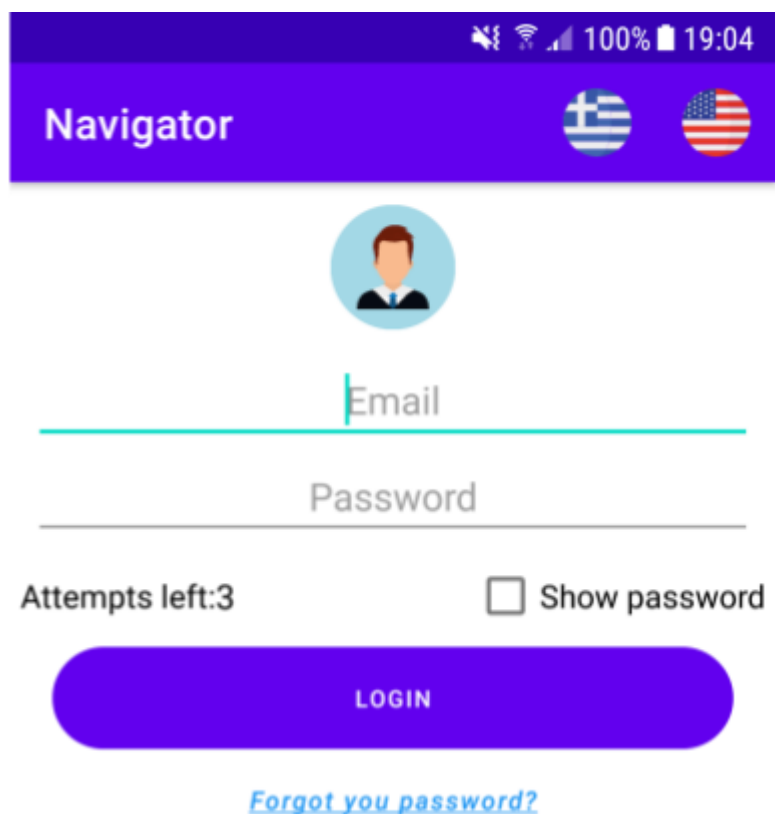
Στην Εικόνα 14 βλέπουμε το στιγμιότυπο από την οθόνη ενός χρήστη την στιγμή που πάτησε το κουμπί “Ξεχάσατε τον κωδικό σας;”.



Εικόνα 14: Ανάκτηση κωδικού πρόσβασης στο Android App

Τέλος δεν θα μπορούσαμε να παραλείψουμε το γεγονός ότι η εφαρμογή έχει πλήρη υποστήριξη τόσο της Ελληνικής όσο και της Αγγλικής γλώσσας. Η αναγκαιότητα υποστήριξης της Αγγλικής γλώσσας σε οποιαδήποτε εφαρμογή κυκλοφορεί αυτή την στιγμή

στο διαδίκτυο μας υποχρεώνει να την συμπεριλάβουμε στην εφαρμογή μας. Από την άλλη πλευρά, επεκτείνει το εύρος χρηστών αφού μια εφαρμογή που υποστηρίζει την Αγγλική γλώσσα μπορεί να χρησιμοποιηθεί όχι μόνο από γνώστες της Ελληνικής γλώσσας αλλά και από ανθρώπους ανά τον πλανήτη που γνωρίζουν Αγγλικά. Στην παρακάτω εικόνα παρουσιάζουμε την σελίδα εισόδου χρηστών στην αγγλική έκδοση. Οι χρήστες μπορούν να αλλάξουν την γλώσσα της εφαρμογής πατώντας την αντίστοιχη σημαία που υπάρχει στο πάνω δεξιά μέρος της οθόνης.



Navigator

Email

Password

Attempts left:3  Show password

LOGIN

[Forgot your password?](#)

Not a member? [Sign up now!](#)

*Εικόνα 15: Αγγλική έκδοση του Android App*

## 5.3 Εγγραφή χρήστη

Η διαδικασία εγγραφής ενός χρήστη αποτελείται από 2 τμήματα. Το πρώτο μέρος αφορά τα στοιχεία του οδηγού ενώ το δεύτερο αφορά τις πληροφορίες του οχήματος.

### 5.3.1 Δημιουργία προφίλ οδηγού

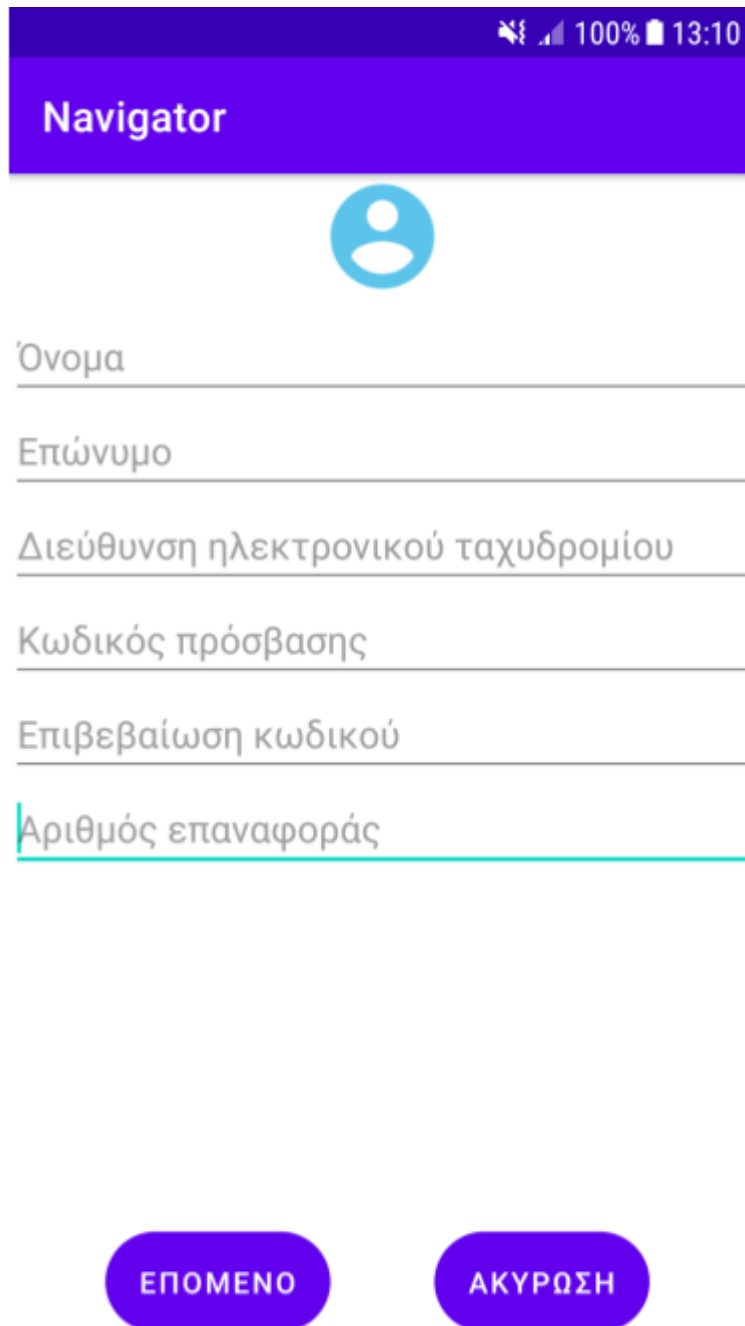
Όπως αναφέραμε και παραπάνω, από την σελίδα εισόδου πατώντας το κουμπί “Εγγραφή τώρα” ο χρήστης μεταφέρεται στην σελίδα εγγραφής. Εκεί αρχικά θα πρέπει να εισάγει τις προσωπικές του πληροφορίες. Πιο αναλυτικά τα στοιχεία που είναι υποχρεωμένος να εισάγει είναι τα εξής:

- Όνομα χρήστη
- Επώνυμο χρήστη
- Email
- Κωδικός πρόσβασης
- Επιβεβαίωση κωδικού πρόσβασης
- Αριθμός επαναφοράς

Ο αριθμός επαναφοράς είναι ένα τετραψήφιο νούμερο που εισάγεται κατά την δημιουργία του λογαριασμού ενός οδηγού και χρησιμοποιείται μόνο σε περίπτωση που έχει ξεχάσει τον κωδικό πρόσβασης του. Πιο συγκεκριμένα, γνωρίζοντας το email και τον αριθμό επαναφοράς μπορεί να αλλάξει τον κωδικό πρόσβασης του. Για κάθε ένα από τα παραπάνω πεδία που αφορούν τον χρήστη υπάρχουν ορισμένοι κανόνες που οφείλει να τηρήσει ο ίδιος:

1. Δεν μπορούν να υπάρχουν κενά πεδία.
2. Το όνομα χρήστη περιέχει μόνο χαρακτήρες.
3. Το επώνυμο χρήστη περιέχει μόνο χαρακτήρες.
4. Το email να είναι στην σωστή μορφή.
5. Δεν μπορούν να υπάρχουν εγγεγραμμένοι χρήστες με το ίδιο email.
6. Ο κωδικός πρόσβασης να ξεκινάει με χαρακτήρα και να αποτελείται από τουλάχιστον 7 και το πολύ 14 χαρακτήρες.
7. Τα πεδία των κωδικών πρόσβασης να περιέχουν τον ίδιο κωδικό.
8. Ο αριθμός επαναφοράς να αποτελείται μόνο από τέσσερις ακέραιους αριθμούς.

Στην εικόνα 16 βλέπουμε την σελίδα εγγραφής ενός χρήστη. Μπορεί οποιαδήποτε στιγμή να ακυρώσει την εισαγωγή των στοιχείων του πατώντας το κουμπί “ΑΚΥΡΩΣΗ” ή να προχωρήσει στο επόμενο στάδιο της εγγραφής πατώντας το κουμπί “ΕΠΟΜΕΝΟ”.



Navigator

Όνομα

Επώνυμο

Διεύθυνση ηλεκτρονικού ταχυδρομίου

Κωδικός πρόσβασης

Επιβεβαίωση κωδικού

Αριθμός επαναφοράς

ΕΠΟΜΕΝΟ

ΑΚΥΡΩΣΗ

Εικόνα 16: Εγγραφή χρήστη στο Android app

### 5.3.2 Δημιουργία προφίλ οχήματος

Μετά την επιτυχή εισαγωγή των προσωπικών πληροφοριών του ο χρήστης μεταβαίνει στην σελίδα εισαγωγής των πληροφοριών του οχήματος που έχει στην κατοχή του για την εκτέλεση των δρομολογίων. Οι πληροφορίες αυτές είναι:

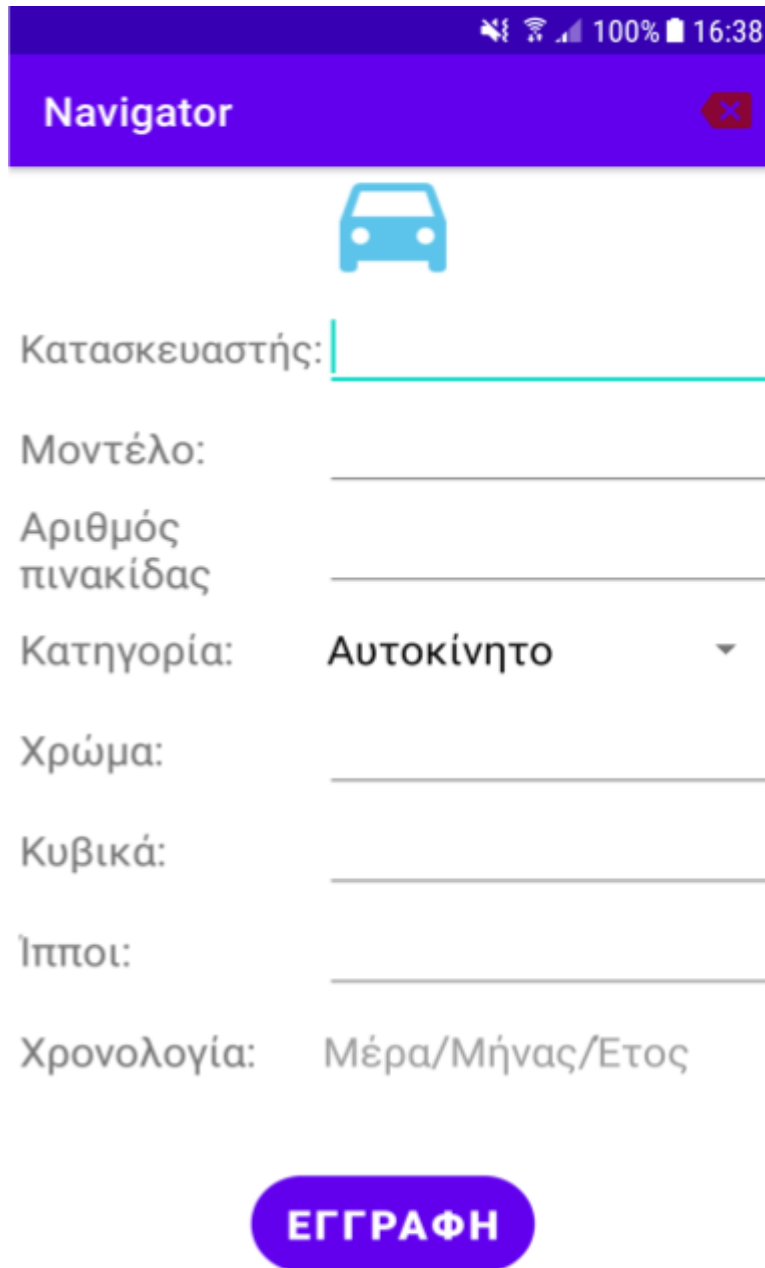
- Ο κατασκευαστής του οχήματος
- Το μοντέλο του οχήματος
- Ο αριθμός πινακίδας του
- Η κατηγορία του οχήματος
- Το χρώμα του
- Τα κυβικά του
- Οι ίπποι του
- Η ημερομηνία κατασκευής του

Και σε αυτή την διαδικασία, για την επιτυχή εγγραφή του οχήματος ο χρήστης θα πρέπει να είναι ιδιαίτερα προσεκτικός σε ορισμένα σημεία κατά την εισαγωγή των δεδομένων. Συγκεκριμένα:

1. Δεν μπορούν να υπάρχουν κενά πεδία.
2. Το όνομα του κατασκευαστή περιέχει μόνο χαρακτήρες.
3. Το μοντέλο του οχήματος περιέχει μόνο χαρακτήρες και αριθμούς.
4. Ο αριθμός πινακίδας του οχήματος αποτελείται μόνο από τέσσερις λατινικούς χαρακτήρες που υπάρχουν και στο ελληνικό αλφάβητο, το σύμβολο – και από 3 ψηφία.
5. Δεν μπορούν να υπάρχουν στο σύστημα δύο ή παραπάνω οχήματα με τον ίδιο αριθμό πινακίδας.
6. Το πεδίο του χρώματος του οχήματος περιέχει μόνο χαρακτήρες.
7. Το πεδίο των κυβικών του αποτελείται μόνο από αριθμούς.
8. Το πεδίο των ίππων του αποτελείται μόνο από αριθμούς.

Στην εικόνα 17 βλέπουμε την σελίδα εγγραφής ενός οχήματος. Μπορεί οποιαδήποτε στιγμή να ακυρώσει την εισαγωγή των στοιχείων του πατώντας το κόκκινο κουμπί που φαίνεται πάνω δεξιά. Με το πάτημα του κουμπιού, ο χρήστης επιστρέφει στην προηγούμενη καρτέλα και διαγράφονται αυτόματα τα στοιχεία οδηγού που εισήγαγε πριν λίγο. Τέλος, με το πάτημα του κουμπιού “ΕΓΓΡΑΦΗ”

καταχωρούνται στην βάση δεδομένων τα στοιχεία του, επιστρέφει στην αρχική σελίδα όπου πλέον μπορεί να κάνει είσοδο με τα στοιχεία που μόλις δημιούργησε.



The screenshot shows the registration screen of the Navigator app. At the top, there is a status bar with icons for signal, Wi-Fi, and battery (100%), and the time 16:38. Below the status bar is a purple header with the text "Navigator" and a close button. A blue car icon is centered below the header. The registration form consists of several fields: "Κατασκευαστής:" with a text input field, "Μοντέλο:" with a text input field, "Αριθμός πινακίδας" with a text input field, "Κατηγορία:" with a dropdown menu showing "Αυτοκίνητο", "Χρώμα:" with a text input field, "Κυβικά:" with a text input field, "Ίπποι:" with a text input field, and "Χρονολογία:" with a dropdown menu showing "Μέρα/Μήνας/Ετος". At the bottom, there is a large purple button with the text "ΕΓΓΡΑΦΗ".

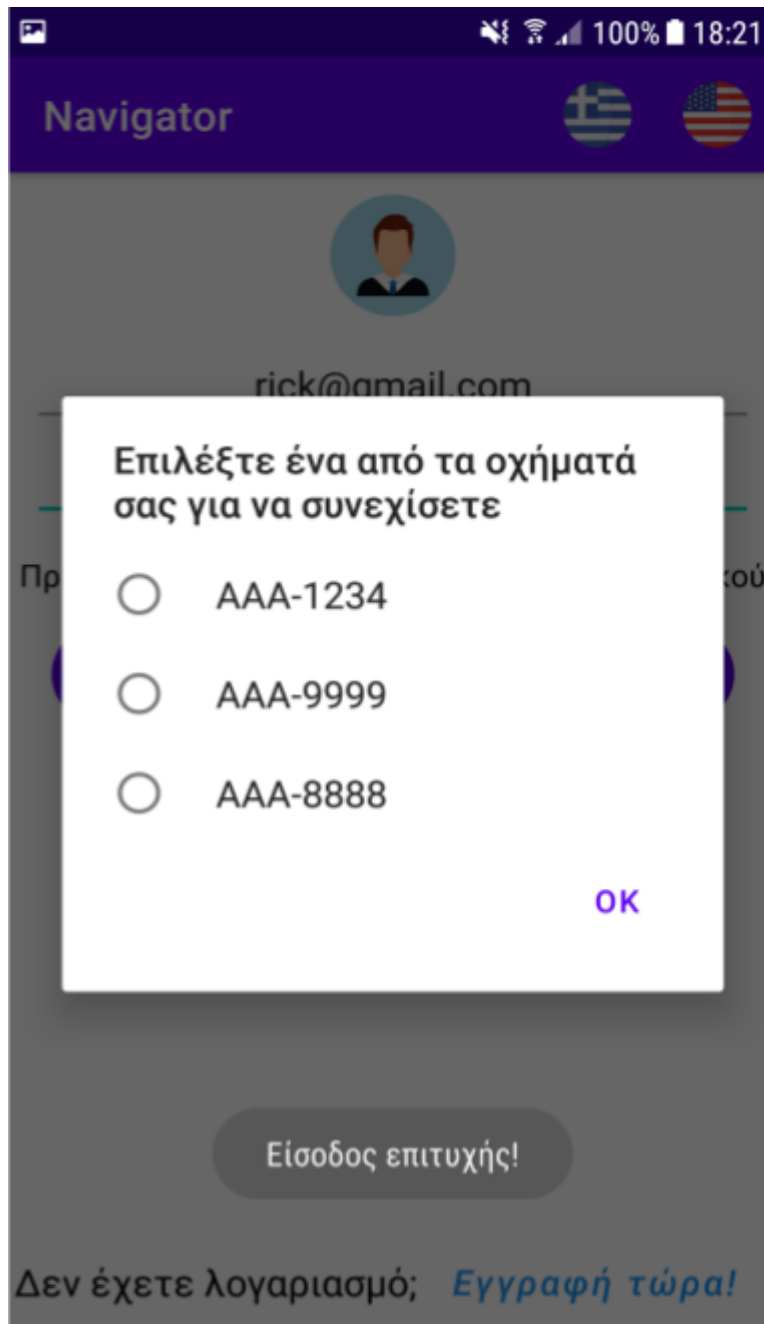
Εικόνα 17: Εγγραφή οχήματος στο Android app

## 5.4 Αρχική σελίδα

Μετά την επιτυχή εισαγωγή από τον χρήστη του email και του κωδικού πρόσβασης του, η εφαρμογή ελέγχει στην βάση δεδομένων τον αριθμό των οχημάτων που έχει ο ίδιος στην κατοχή του. Όπως έχουμε αναφέρει και στην εισαγωγή του συγκεκριμένου κεφαλαίου, ο κάθε οδηγός μπορεί να κατέχει περισσότερα από ένα οχήματα. Σε περίπτωση που η



εφαρμογή εντοπίσει ότι ο χρήστης διαθέτει τουλάχιστον ένα όχημα εμφανίζει, όπως βλέπουμε και στην εικόνα 18, σχετικό μήνυμα για να επιλέξει ο ίδιος με ποιο όχημα θέλει να πραγματοποιήσει το δρομολόγιο του. Σε αντίθετη περίπτωση μεταβαίνει απευθείας στην αρχική σελίδα με την εφαρμογή να επιλέξει απευθείας το μοναδικό όχημα του. Θα πρέπει να τονίσουμε ότι δεν μπορεί να υπάρξει ενεργός χρήστης χωρίς να διαθέτει τουλάχιστον ένα όχημα.

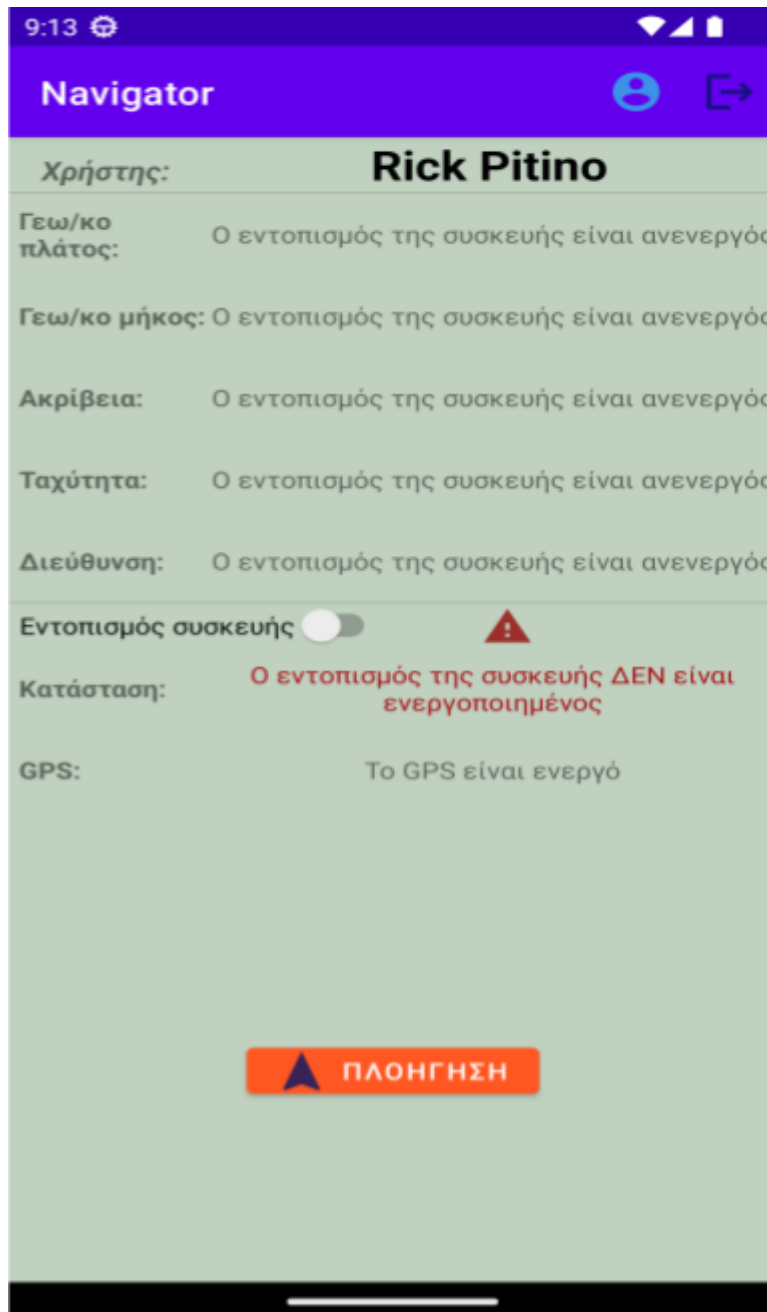


Εικόνα 18: Επιλογή οχήματος στο Android app

Μπαίνοντας στην αρχική σελίδα, οι πληροφορίες που μπορεί να αντλήσει ο χρήστης είναι οι εξής:

- Ονοματεπώνυμο χρήστη

- Συντεταγμένες οχήματος
- Ακρίβεια εντοπισμού
- Ταχύτητα οχήματος
- Διεύθυνση οχήματος

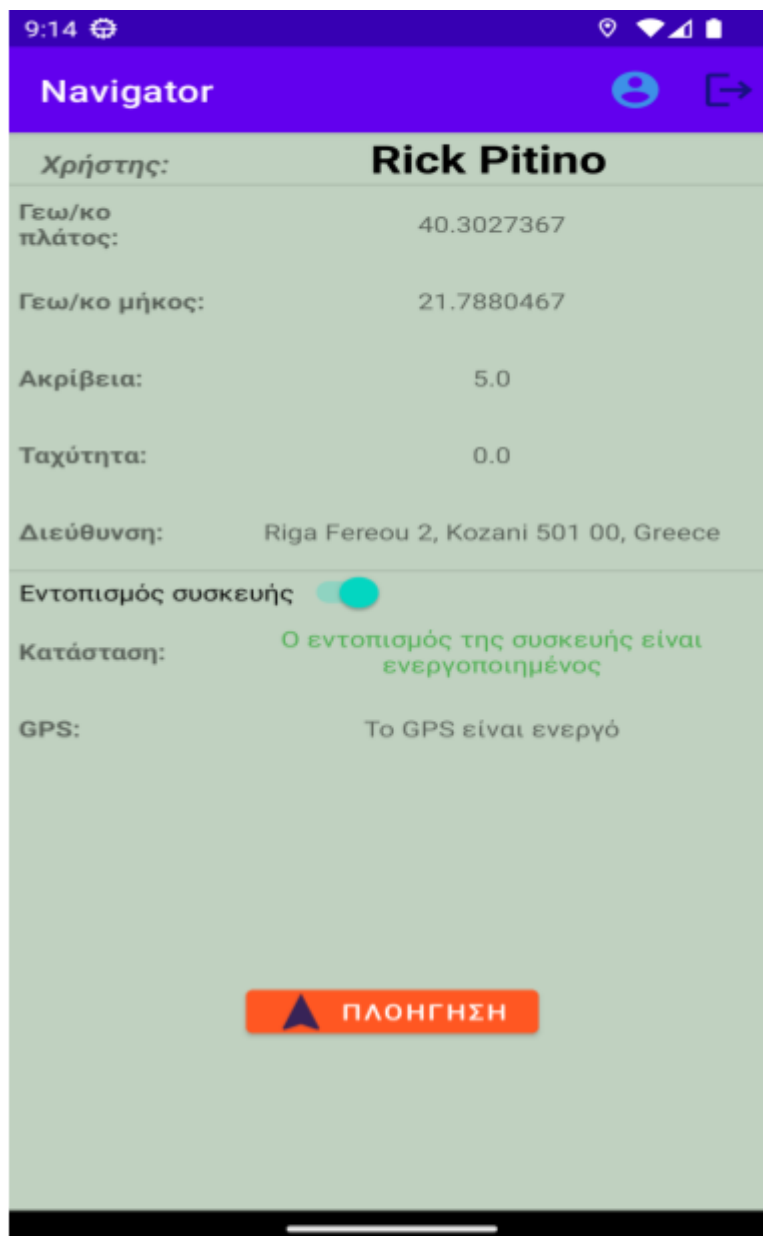


Εικόνα 19: Αρχική σελίδα του Android app

Η ακρίβεια εντοπισμού είναι ένας αριθμός που αντιπροσωπεύει την απόκλιση της τοποθεσίας που εντόπισε η συσκευή με την πραγματική τοποθεσία της. Συνεπώς καταλαβαίνουμε ότι όσο πιο μικρός είναι αυτός ο αριθμός τόσο πιο ακριβής είναι η τοποθεσία της συσκευής.

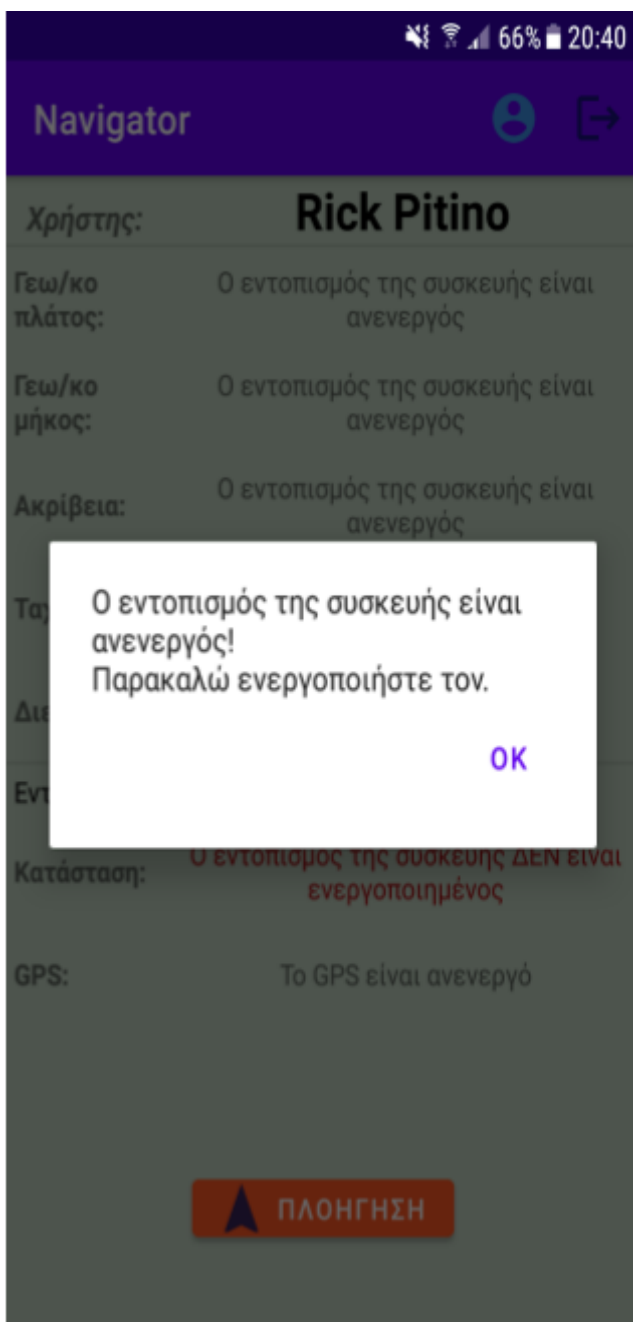
Με την βοήθεια του GPS και των δεδομένων κινητής τηλεφωνίας, η συσκευή λαμβάνει τις συντεταγμένες. Στην συνέχεια μέσω μιας διαδικασίας που ονομάζεται Geocoding, μετατρέπουμε τις συντεταγμένες που λάβαμε σε ονόματα οδών.

Πέρα από τις παραμέτρους που ήδη αναφέραμε, στην αρχική οθόνη βλέπουμε έναν διακόπτη που ονομάζεται “Εντοπισμός συσκευής”. Το γεγονός ότι οι υπηρεσίες εντοπισμού καταναλώνουν ένα σεβαστό ποσοστό της μπαταρίας δεν μας δίνει την πολυτέλεια να τις χρησιμοποιούμε συνέχεια. Για τον λόγο αυτό, όσο είναι ανενεργός ο διακόπτης αυτός η συσκευή δεν θα προσπαθεί να εντοπίσει την τοποθεσία της. Αντιθέτως μόλις ο χρήστης ενεργοποιήσει τον διακόπτη, η συσκευή με την χρήση του GPS και της σύνδεσης στο διαδίκτυο θα ξεκινήσει την διαδικασία εντοπισμού. Προκειμένου ο χρήστης να έχει μια πλήρη εικόνα της κατάστασης σχετικά με την διαδικασία εύρεσης της συσκευής έχουμε ορίσει το πεδίο “Κατάσταση”. Όσο ο διακόπτης εντοπισμού είναι ανενεργός, η αναφορά της Κατάστασης θα είναι αυτή που βλέπουμε στην εικόνα 19 ενώ στην εικόνα 20 βλέπουμε ένα παράδειγμα από την ενεργοποίηση του διακόπτη εντοπισμού.

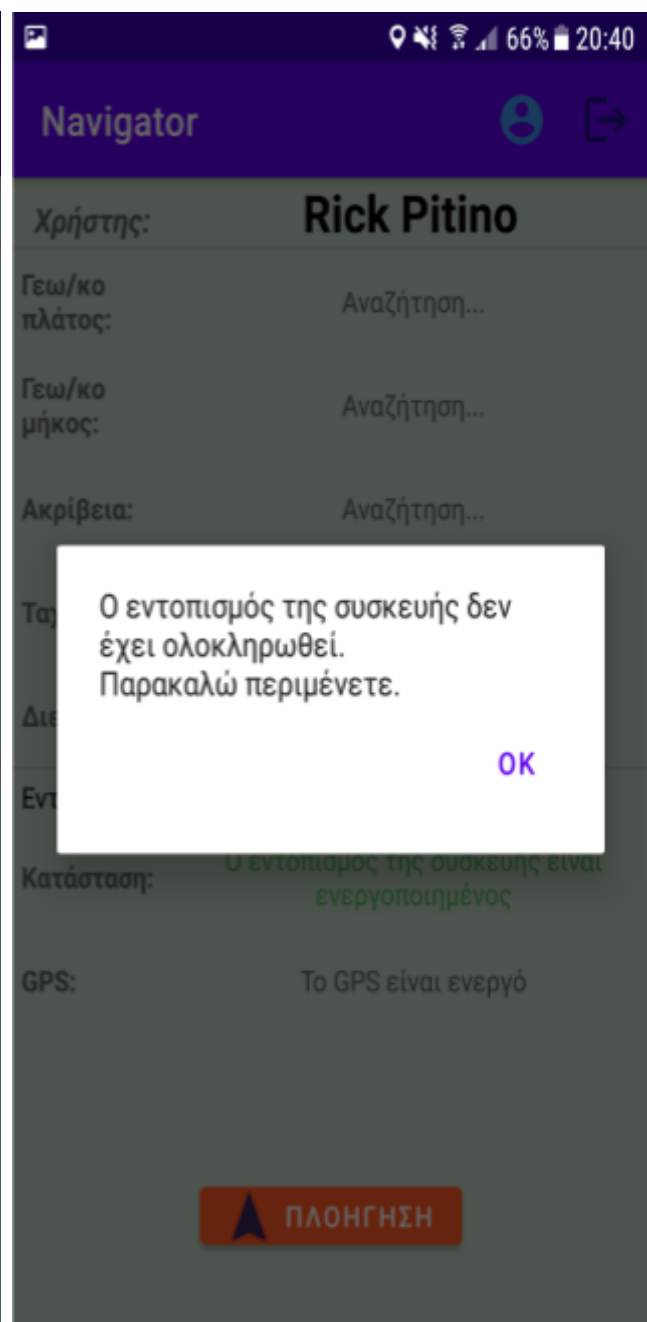


Εικόνα 20: Ενεργοποίηση εντοπισμού συσκευής στο Android App

Η τελευταία πληροφορία που μπορούμε να λάβουμε από την αρχική οθόνη είναι η κατάσταση των υπηρεσιών εντοπισμού της συσκευής. Για την έγκαιρη ενημέρωση των χρηστών η εφαρμογή ελέγχει ανά τακτά χρονικά διαστήματα εάν είναι ενεργό το GPS ή όχι χωρίς να χρειάζεται να μεταβαίνουν οι ίδιοι στις ρυθμίσεις ή στην γραμμή κατάστασης της συσκευής για να ελέγξουν. Τέλος υπάρχει το κουμπί “ΠΛΟΗΓΗΣΗ” όπου μεταφέρει τον χρήστη στην σελίδα του χάρτη. Θα πρέπει να σημειώσουμε ότι για την μετάβαση στην σελίδα των χαρτών είναι απαραίτητη η ενεργοποίηση του διακόπτη εντοπισμού και η ολοκλήρωση της διαδικασίας εντοπισμού. Σε αντίθετη περίπτωση εμφανίζονται μηνύματα σφάλματος όπως φαίνονται στις εικόνες 21 και 22 αντίστοιχα.





Εικόνα 21: Σφάλμα ανενεργού διακόπτη εντοπισμού στο Android App



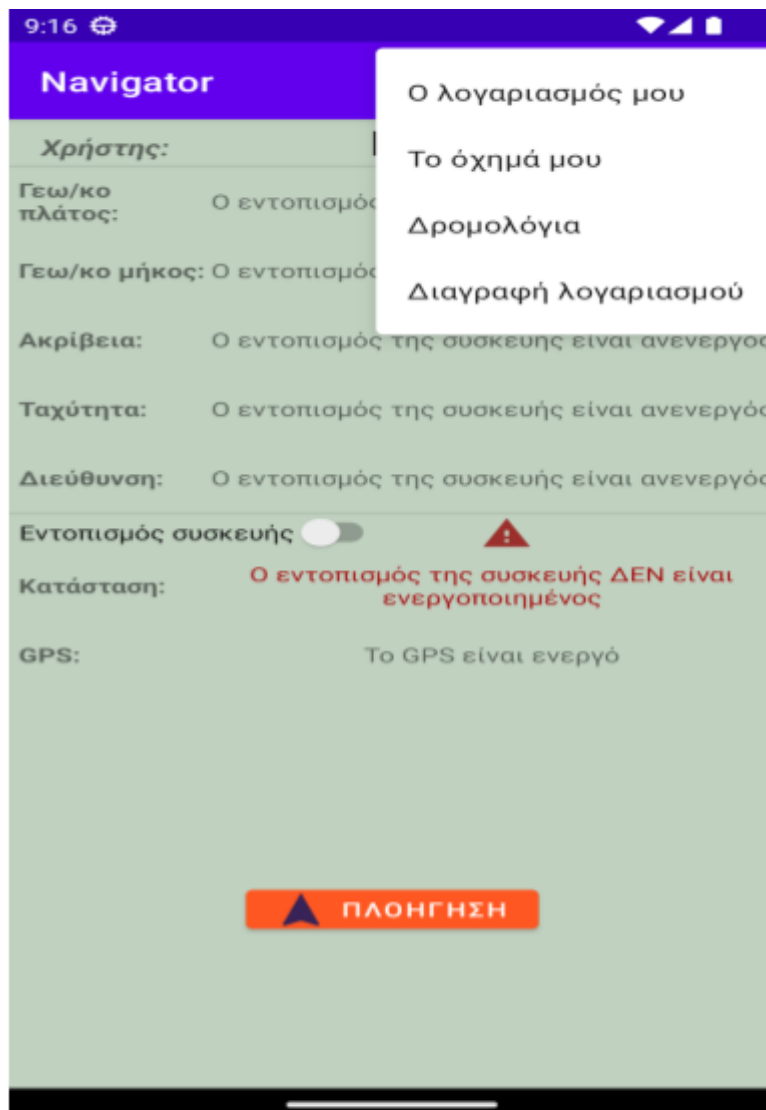
Εικόνα 22: Μήνυμα αναμονής για τον εντοπισμό της συσκευής στο Android App

Κλείνοντας την ενότητα της αρχικής σελίδας, μπορούμε να διακρίνουμε στην δεξιά πλευρά της γραμμής κατάστασης 2 εικονίδια.

Πατώντας ο χρήστης το εικονίδιο  πραγματοποιεί αποσύνδεση από τον λογαριασμό του και μεταφέρεται στην σελίδα εισόδου.

Πατώντας το εικονίδιο  εμφανίζεται, όπως φαίνεται στην εικόνα 23, ένα μενού με 4 επιλογές:

1. **Ο Λογαριασμός μου** όπου μπορεί να αποκτήσει πρόσβαση στις προσωπικές του πληροφορίες. Ακόμη δίνεται η δυνατότητα αλλαγής του κωδικού πρόσβασης.
2. **Το όχημά μου** όπου λαμβάνει πληροφορίες σχετικά με το όχημά του. Επιπλέον σε περίπτωση που έχει στην κατοχή του περισσότερα από ένα οχήματα, έχει την ευκαιρία να επιλέξει με ποιο όχημα επιθυμεί να πραγματοποιήσει το δρομολόγιό του.
3. **Δρομολόγια** εμφανίζοντας μια λίστα με όλα τα δρομολόγια που έχει ολοκληρώσει.
4. **Διαγραφή λογαριασμού** για την οριστική διαγραφή του χρήστη από το σύστημα.



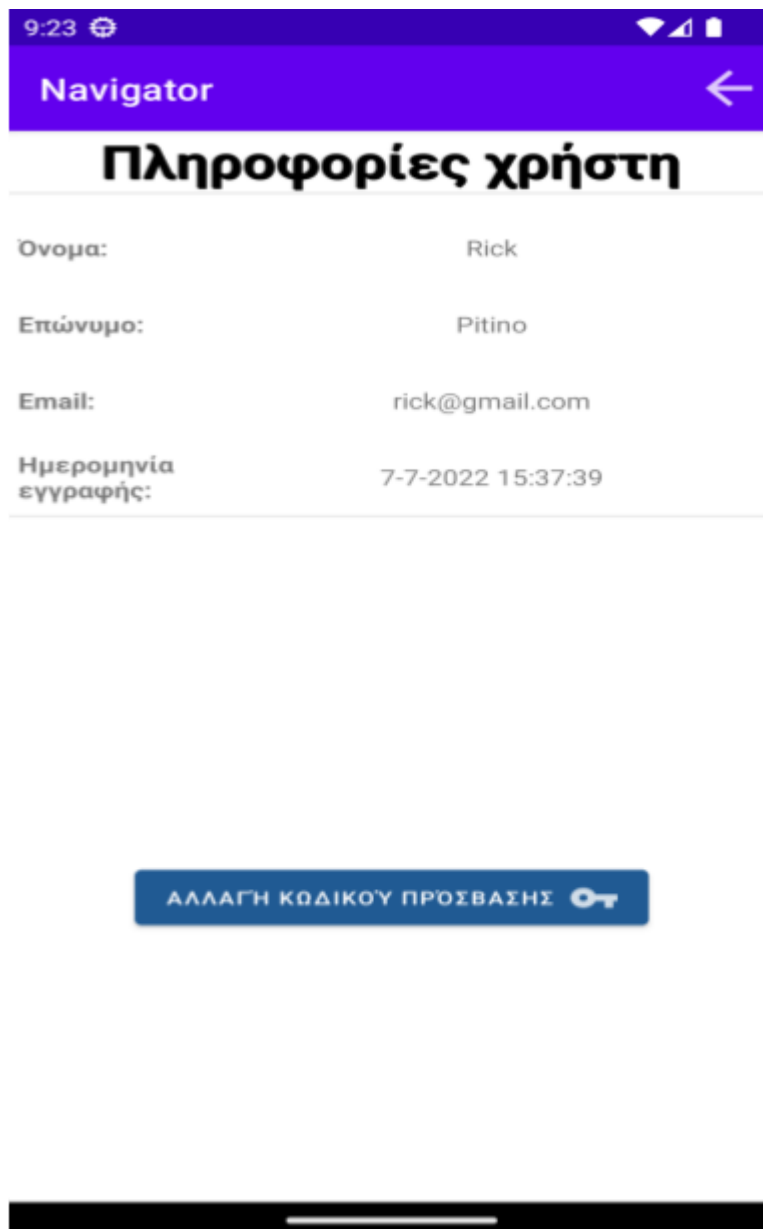
Εικόνα 23: Dropdown menu της αρχικής οθόνης στο Android app

### 5.4.1 Πληροφορίες λογαριασμού

Σε αυτή την υποενότητα βλέπουμε την σελίδα που θα εμφανιστεί στον χρήστη όταν πατήσει την επιλογή “Ο Λογαριασμός μου” από το μενού της αρχικής σελίδας. Η σελίδα αυτή περιέχει πληροφορίες που αφορούν αποκλειστικά τον ίδιο. Πιο συγκεκριμένα, οι πληροφορίες αυτές είναι:

- Το όνομα του χρήστη
- Το επώνυμο του χρήστη
- Το email του
- Η ημερομηνία που πραγματοποίησε την εγγραφή του

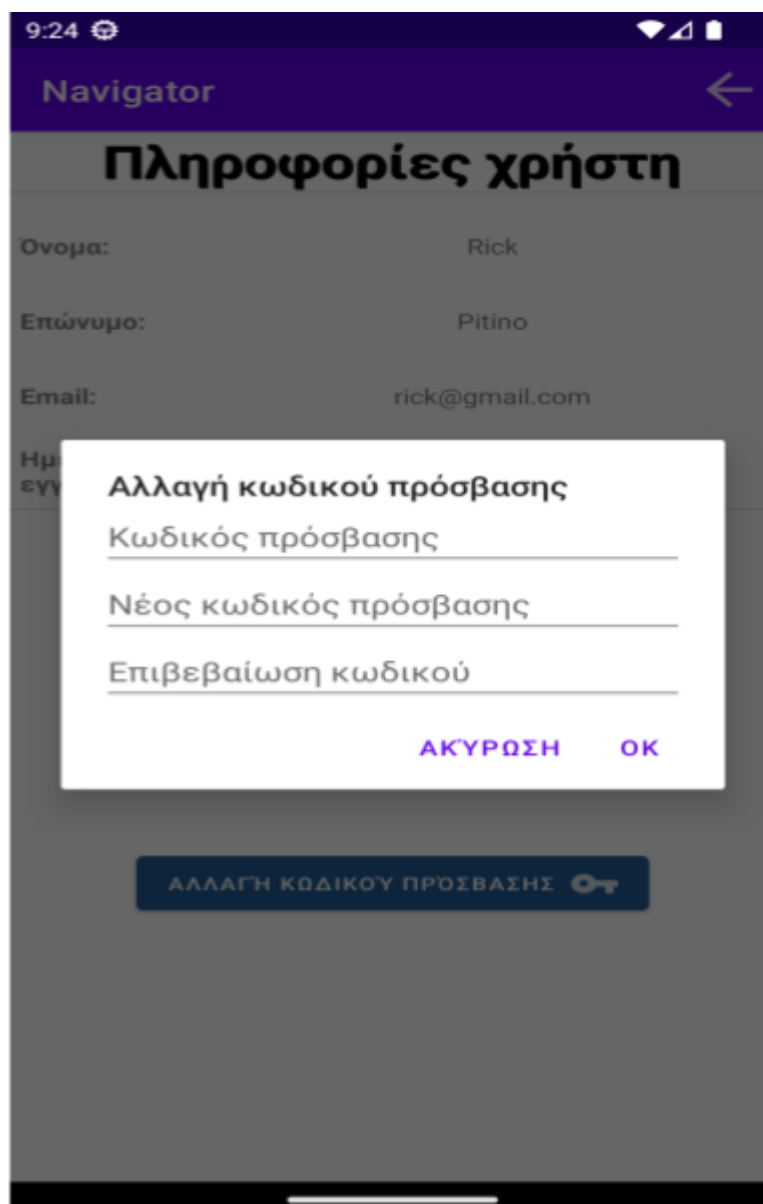
Στην εικόνα 24 βλέπουμε ένα στιγμιότυπο από την σελίδα του χρήστη Rick Pitino με το email του να είναι “[rick@gmail.com](mailto:rick@gmail.com)” ενώ η ημερομηνία εγγραφή του είναι 7 Ιουλίου 2022 και ώρα 3:37 το μεσημέρι.



Εικόνα 24: Σελίδα στοιχείων χρήστη στο Android app

Ακόμη, παρατηρούμε το κουμπί επιστροφής που βρίσκεται στην δεξιά πλευρά της γραμμής κατάστασης και δίνει την δυνατότητα στον χρήστη να επιστρέψει στην αρχική σελίδα με ένα πάτημα ενώ στο κέντρο της σελίδας υπάρχει το κουμπί “ΑΛΛΑΓΗ ΚΩΔΙΚΟΥ ΠΡΟΣΒΑΣΗΣ” για την εύκολη και άμεση αλλαγή του κωδικού πρόσβασης.

Η αλλαγή κωδικού πρόσβασης αποτελεί μια εξαιρετικά απλή διαδικασία. Όπως έχουμε παρατηρήσει σχεδόν σε όλες τις γνωστές εφαρμογές που υπάρχουν αυτή την στιγμή στο internet, όταν ένας χρήστης επιθυμεί να αλλάξει τον κωδικό πρόσβασης του λογαριασμού του το μόνο που χρειάζεται να γνωρίζει είναι ο τρέχων κωδικός του. Σε αυτή την λογική βασίζεται και η συγκεκριμένη εφαρμογή. Σε αυτό το σημείο είμαστε υποχρεωμένοι να τονίσουμε ότι δεν απαιτείται η γνώση του αριθμού επαναφοράς (PIN) που γνωρίσαμε στην σελίδα εισόδου διότι ο χρήστης έχει πραγματοποιήσει επιτυχή είσοδο στον λογαριασμό του. Ο αριθμός επαναφοράς χρησιμοποιείται μόνο σε περίπτωση που δεν μπορεί να εισέλθει στον λογαριασμό του και θέλει να αλλάξει τον κωδικό πρόσβασης από την σελίδα εισόδου. Στην εικόνα 25 φαίνεται το στιγμιότυπο στην στιγμή που ο χρήστης Rick Pitino πάτησε το κουμπί “ΑΛΛΑΓΗ ΚΩΔΙΚΟΥ ΠΡΟΣΒΑΣΗΣ”.



Εικόνα 25: Αλλαγή κωδικού πρόσβασης από την σελίδα χρήστη στο Android app

## 5.4.2 Πληροφορίες οχήματος

Σε αυτή την υποενότητα βλέπουμε την σελίδα που θα εμφανιστεί στον χρήστη όταν πατήσει την επιλογή “Το όχημά μου” από το μενού της αρχικής σελίδας. Η σελίδα αυτή περιέχει πληροφορίες που αφορούν το όχημα του. Πιο συγκεκριμένα, οι πληροφορίες αυτές είναι:

- Ο αριθμός πινακίδας του οχήματος
- Ο κατασκευαστής του οχήματος
- Το μοντέλο του
- Η κατηγορία στην οποία ανήκει το όχημα
- Το χρώμα του
- Τα κυβικά του
- Οι ίπποι του
- Η ημερομηνία κατασκευής του

Το κουμπί επιστροφής βρίσκεται στην δεξιά πλευρά της γραμμής κατάστασης και δίνει την δυνατότητα επιστροφής στην αρχική σελίδα με ένα πάτημα. Στην εικόνα 26 βλέπουμε τα στοιχεία ενός εκ των οχημάτων που ανήκουν στον Rick Pitino. Περνώντας σε λεπτομέρειες, ο χρήστης αυτός έχει στην κατοχή του ένα Tesla Model 3 με αριθμό πινακίδας AAA-9999. Το συγκεκριμένο όχημα είναι πράσινου χρώματος, η ιπποδύναμη του ανέρχεται στα 200 άλογα ενώ τα κυβικά του στα 2000. Τέλος, πρόκειται για ένα καινούργιο αυτοκίνητο καθώς κατασκευάστηκε στις 10 Σεπτεμβρίου του 2020. Η συγκεκριμένη σελίδα εμφανίζει τις πληροφορίες του οχήματος που έχει επιλέξει ο χρήστης κατά την είσοδο του στην εφαρμογή. Δεν λαμβάνονται υπ όψιν τα υπόλοιπα οχήματα που μπορεί να έχει ένας οδηγός. Στο σημείο αυτό να τονίσουμε ότι οι πληροφορίες που παρουσιάζουμε για τα οχήματα είναι πλασματικές και δεν αντικατοπτρίζουν σε καμία περίπτωση την πραγματικότητα. Τα δεδομένα έχουν καταχωρηθεί με σκοπό την υλοποίηση και παρουσίαση της τρέχουσας εργασίας. Οποιαδήποτε ομοιότητα στα χαρακτηριστικά ενός οχήματος που παρουσιάζεται σε αυτή την εργασία με κάποιο πραγματικό όχημα είναι εντελώς συμπτωματική.

Το τελευταίο κουμπί που φαίνεται στην κάτω πλευρά της οθόνης με όνομα “ΑΛΛΑΓΗ ΟΧΗΜΑΤΟΣ” προσφέρει στον χρήστη την επιλογή να αλλάξει το τρέχων όχημα του με κάποιο άλλο που διαθέτει για να εκτελέσει το δρομολόγιό του.

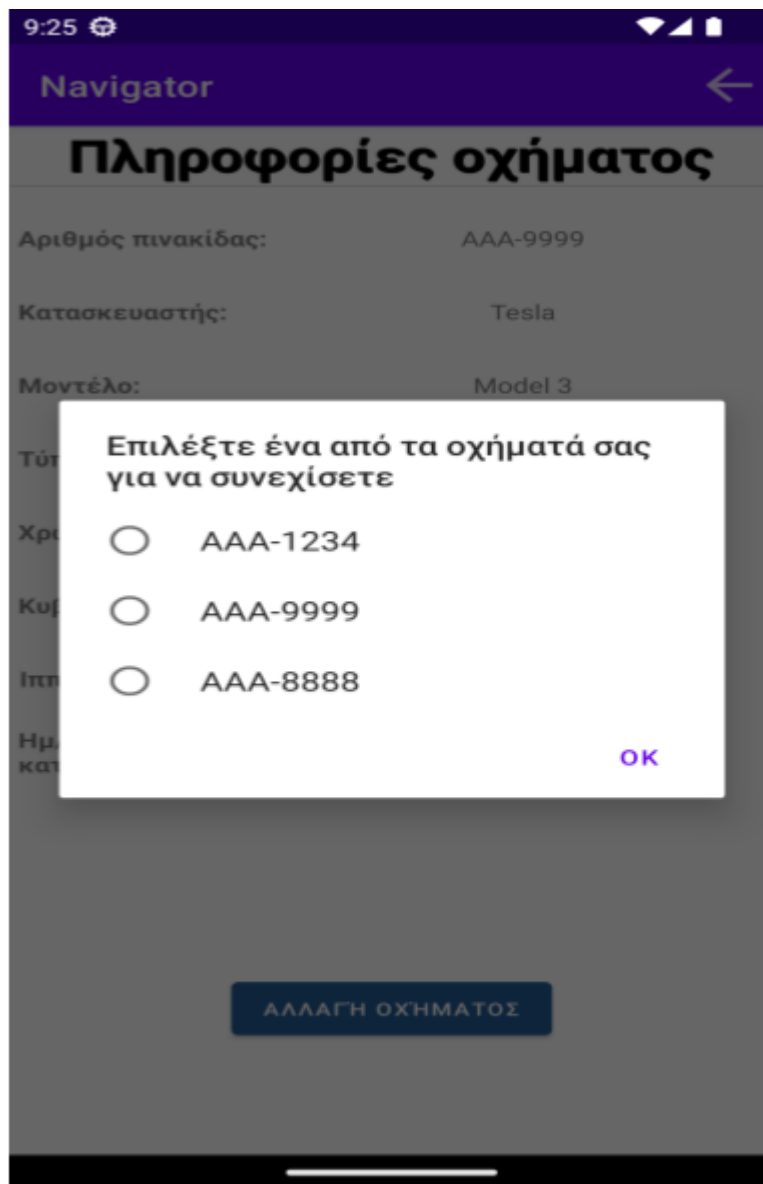




Εικόνα 26: Σελίδα στοιχείων οχήματος του χρήστη στο Android app

Με σκοπό την διευκόλυνση των χρηστών που έχουν στην κατοχή τους τουλάχιστον ένα όχημα, μέσα από την σελίδα πληροφοριών του οχήματος παρέχεται η δυνατότητα επιλογής ενός από αυτά δίχως την υποχρέωση να κάνουν αποσύνδεση και να εισέλθουν στην εφαρμογή για να επιλέξουν διαφορετικό όχημα. Στην εικόνα 27 παρατηρούμε ότι ο Rick Pitino διαθέτει 3 οχήματα με πινακίδες:

- AAA-1234
- AAA-9999
- AAA-8888



Εικόνα 27: Αλλαγή οχήματος δρομολογίων από την σελίδα οχήματος στο Android app

### 5.4.3 Πληροφορίες διαδρομών

Κλείνοντας το κεφάλαιο της αρχικής σελίδας θα πρέπει να αναφέρουμε την σελίδα που θα εμφανιστεί στον χρήστη όταν πατήσει την επιλογή “Δρομολόγια” από το μενού της αρχικής σελίδας. Η σελίδα αυτή περιέχει πληροφορίες που αφορούν τα δρομολόγια που έχει ολοκληρώσει ο οδηγός. Πιο συγκεκριμένα, οι πληροφορίες αυτές είναι:

- Η ημερομηνία και ώρα έναρξης του δρομολογίου
- Το σημείο έναρξης
- Το σημείο προορισμού

Και σε αυτή την σελίδα υπάρχει το κουμπί επιστροφής στην γραμμή κατάστασης για την επιστροφή του χρήστη στην αρχική σελίδα. Η εικόνα 28 αποτελεί ένα παράδειγμα με τα δρομολόγια που έχει πραγματοποιήσει ο Rick Pitino.

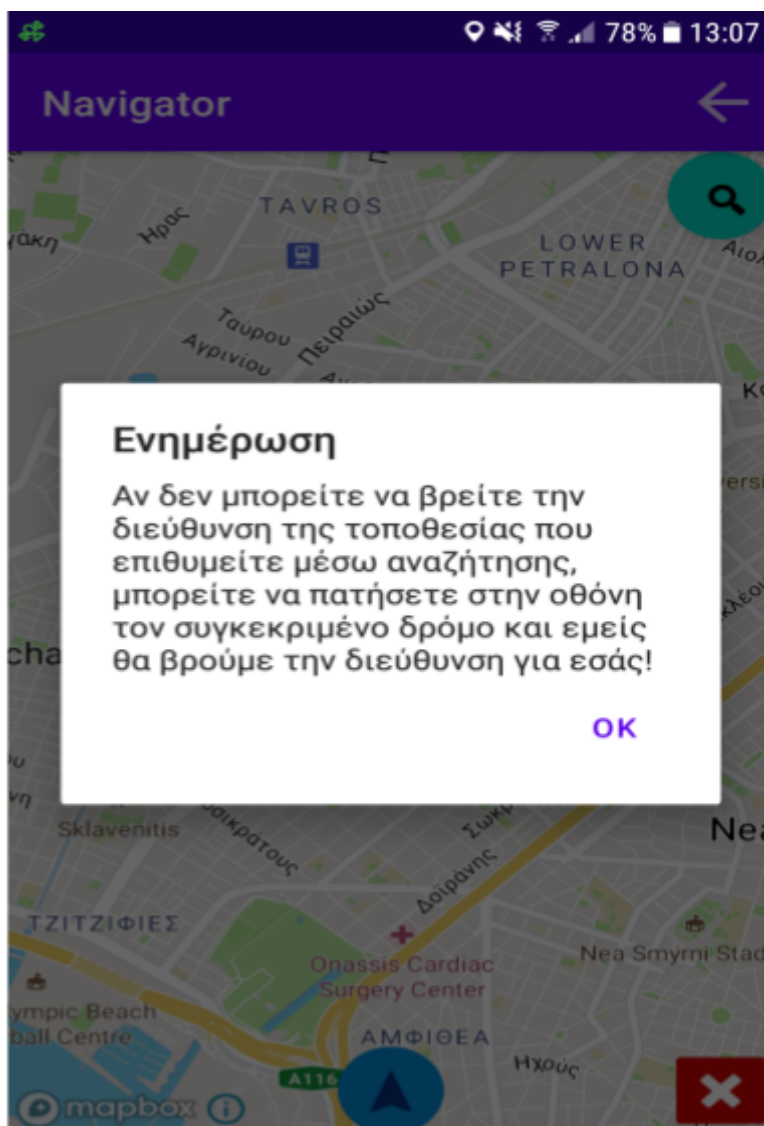
Εναρξη	Αφετηρία	Προορισμός
6-10-2022 16:20:42	Πλ. Ομόνοιας 2 Αθήνα 105 52 Ελλάδα	Σαλαμίνας 23 Αθήνα 104 35 Ελλάδα
6-10-2022 16:52:21	Πλ. Ομόνοιας 2 Αθήνα 105 52 Ελλάδα	Βεΐκου 31 Αθήνα 117 42 Ελλάδα
6-10-2022 16:53:30	Πλ. Ομόνοιας 2 Αθήνα 105 52 Ελλάδα	Εμμανουήλ Μπενάκη 134 Αθήνα 114 73 Ελλάδα
6-10-2022 18:26:40	Πλ. Ομόνοιας 2 Αθήνα 105 52 Ελλάδα	Αρκαδίου 74 Δάφνη Αττικής 172 34 Ελλάδα
6-10-2022 18:27:48	Πλ. Ομόνοιας 2 Αθήνα 105 52 Ελλάδα	Ηρακλειδών 5 Αθήνα 118 51 Ελλάδα
6-10-2022 18:28:50	Πλ. Ομόνοιας 2 Αθήνα 105 52 Ελλάδα	Παπαρρηγοπούλου 2 Περιστερί 121 32 Ελλάδα
6-10-2022 18:30:24	Πλ. Ομόνοιας 2 Αθήνα 105 52 Ελλάδα	Χαμοστέρνας 64 Αθήνα 118 53 Ελλάδα
6-10-2022 18:31:19	Πλ. Ομόνοιας 2 Αθήνα 105 52 Ελλάδα	Αγίας Παρασκευής 26 Περιστερί 121 32 Ελλάδα
6-10-2022 18:32:52	Πλ. Ομόνοιας 2 Αθήνα 105 52 Ελλάδα	Σχολείου 5 Αθήνα 105 58 Ελλάδα
6-10-2022 18:33:55	Πλ. Ομόνοιας 2 Αθήνα 105 52 Ελλάδα	Πλ. Αγίου Νικολάου 15 Αθήνα 104 46 Ελλάδα

Εικόνα 28: Σελίδα δρομολογίων του χρήστη στο Android app

## 5.5 Πλοήγηση

Στην ενότητα αυτή παρουσιάζεται το κομμάτι της πλοήγησης που αποτελεί και το βασικό αντικείμενο της διπλωματικής εργασίας. Στην παρουσίαση της αρχικής σελίδας, είδαμε ένα πορτοκαλί κουμπί στο κάτω μέρος της οθόνης με όνομα “ΠΛΟΗΓΗΣΗ”. Πατώντας ο οδηγός το κουμπί αυτό μεταβαίνει στην σελίδα του χάρτη. Με την είσοδό του στην σελίδα εμφανίζεται ένα μήνυμα: “Αν δεν μπορείτε να βρείτε την διεύθυνση της τοποθεσίας που επιθυμείτε μέσω αναζήτησης, μπορείτε να πατήσετε στην οθόνη τον συγκεκριμένο δρόμο και εμείς θα βρούμε την διεύθυνση για εσάς!”. Πολλές φορές βλέπουμε

σε έναν χάρτη μια συγκεκριμένη τοποθεσία (πχ κατάσταση) ή μια οδό αλλά δεν γνωρίζουμε την διεύθυνσή της. Η εφαρμογή μας προσφέρει την δυνατότητα να εντοπίσουμε στον χάρτη την τοποθεσία αυτή και με ένα απλό πάτημα της οθόνης στο σημείο που θέλουμε να βρει την διεύθυνση για εμάς και να σχεδιάσει την διαδρομή από την τρέχουσα τοποθεσία μας προς εκείνη. Στην εικόνα 29 βλέπουμε το χαρακτηριστικό μήνυμα εισόδου.

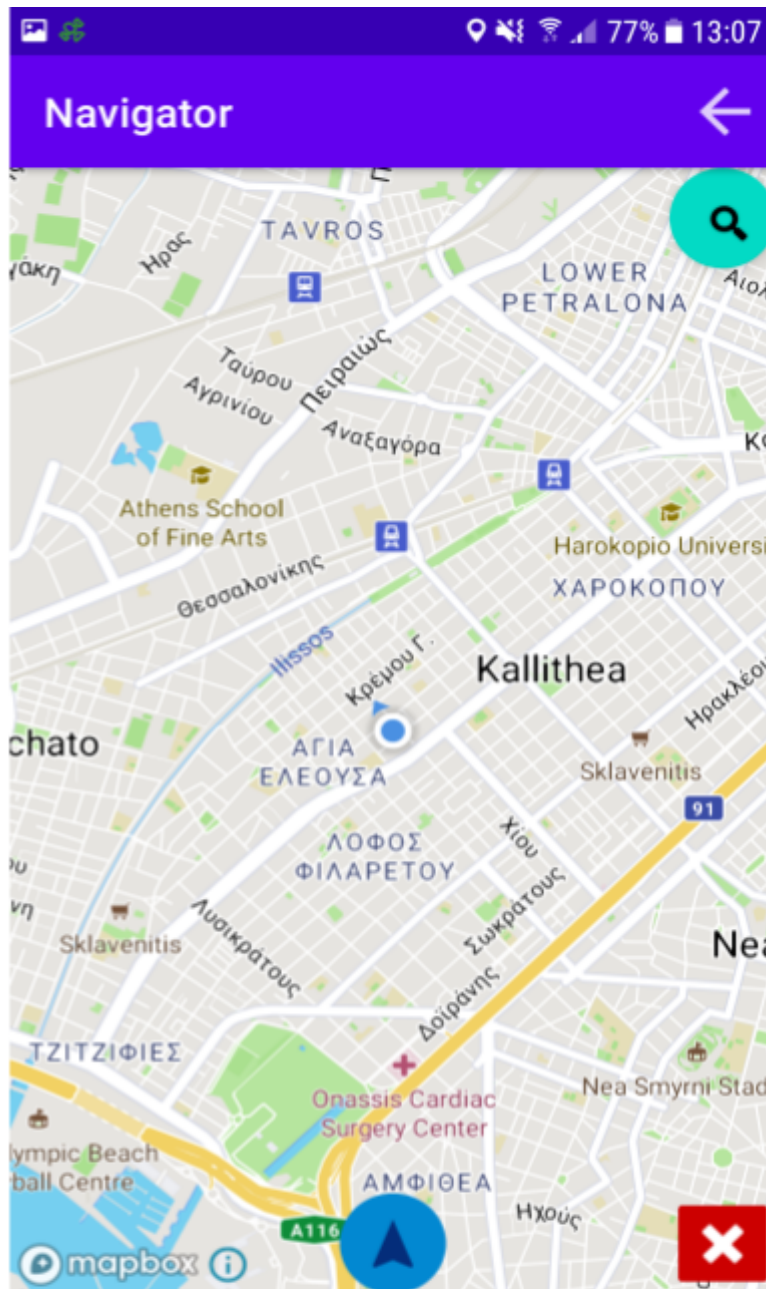


Εικόνα 29: Μήνυμα εισόδου στην σελίδα του χάρτη στο Android App

Η εφαρμογή, με την είσοδο του χρήστη στην σελίδα του χάρτη, εμφανίζει την τοποθεσία του. Στη συνέχεια οι επιλογές που του προσφέρει είναι:

- Επιστροφή στην αρχική σελίδα
- Αναζήτηση μιας διεύθυνσης


- Διαγραφή μιας διεύθυνσης
- Πλοήγηση

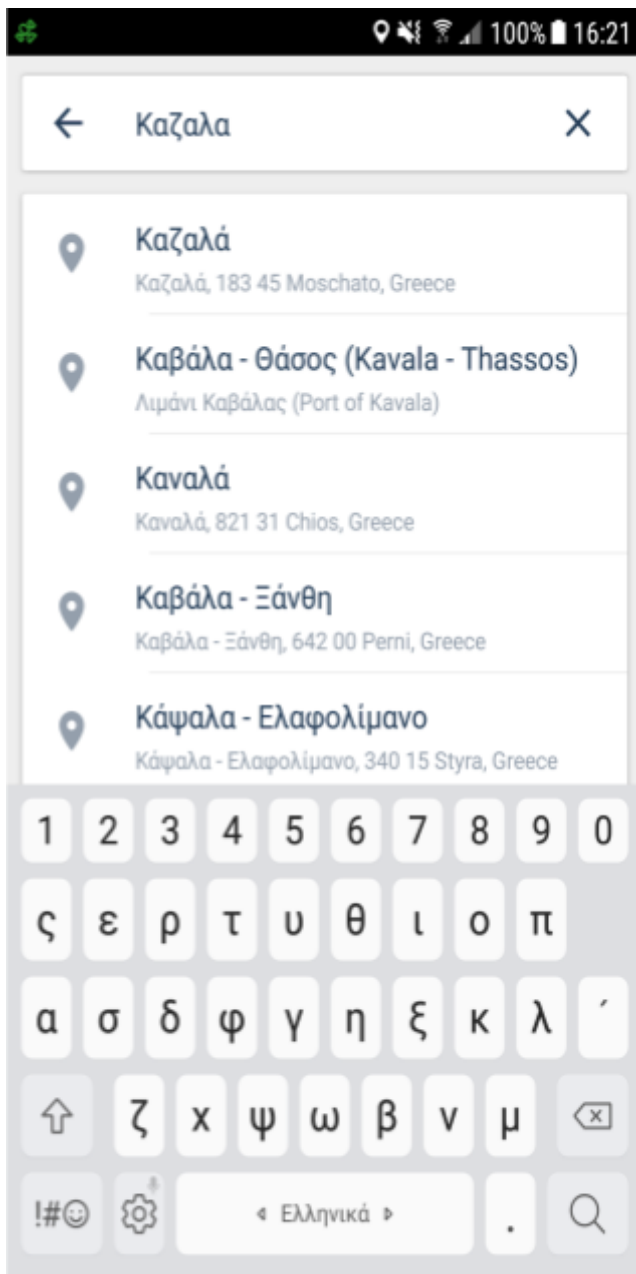


Εικόνα 30: Σελίδα του χάρτη στο Android app

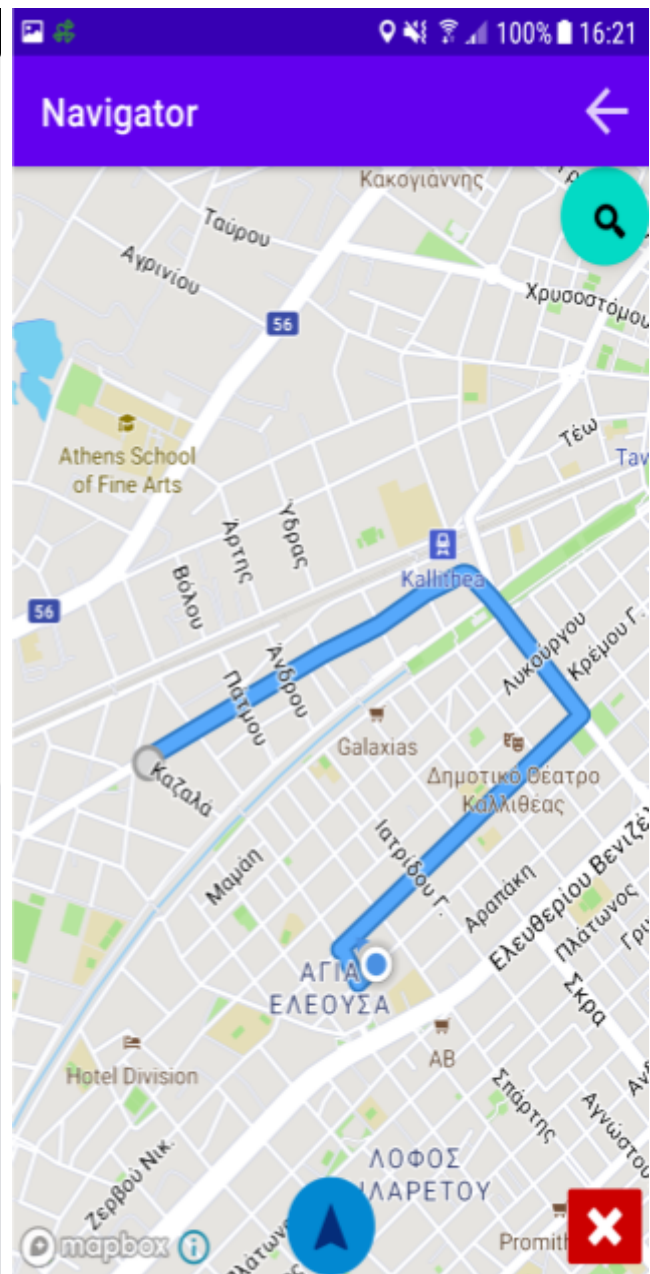
Στην εικόνα 30, η εφαρμογή εντόπισε την τοποθεσία της συσκευής στην Αγία Ελεούσα στην περιοχή της Καλλιθέας. Επίκεντρο της σελίδας είναι πάντα η τρέχουσα τοποθεσία της συσκευής. Σε αυτό το σημείο ο οδηγός έχει την δυνατότητα να εισάγει τουλάχιστον μια τοποθεσία με σκοπό η εφαρμογή να διαμορφώσει μια διαδρομή και να ξεκινήσει το δρομολόγιό του.

Η αναζήτηση μιας διεύθυνσης στον χάρτη μπορεί να πραγματοποιηθεί όχι μόνο με το πάτημα στην οθόνη που αναφέραμε παραπάνω. Στην περίπτωση που η διεύθυνση μιας

στάσης μάς είναι γνωστή μπορούμε πατώντας το κουμπί  που βρίσκεται πάνω δεξιά, να πληκτρολογήσουμε την διεύθυνση και η εφαρμογή θα εντοπίσει στον χάρτη την ακριβή τοποθεσία και θα σχεδιάσει απευθείας μια διαδρομή με προορισμό αυτήν. Στην εικόνα 31, ο χρήστης έκανε αναζήτηση για την οδό Καζαλά και όπως φαίνεται στην εικόνα 32 η εφαρμογή έχει διαμορφώσει την διαδρομή του για την συγκεκριμένη οδό.




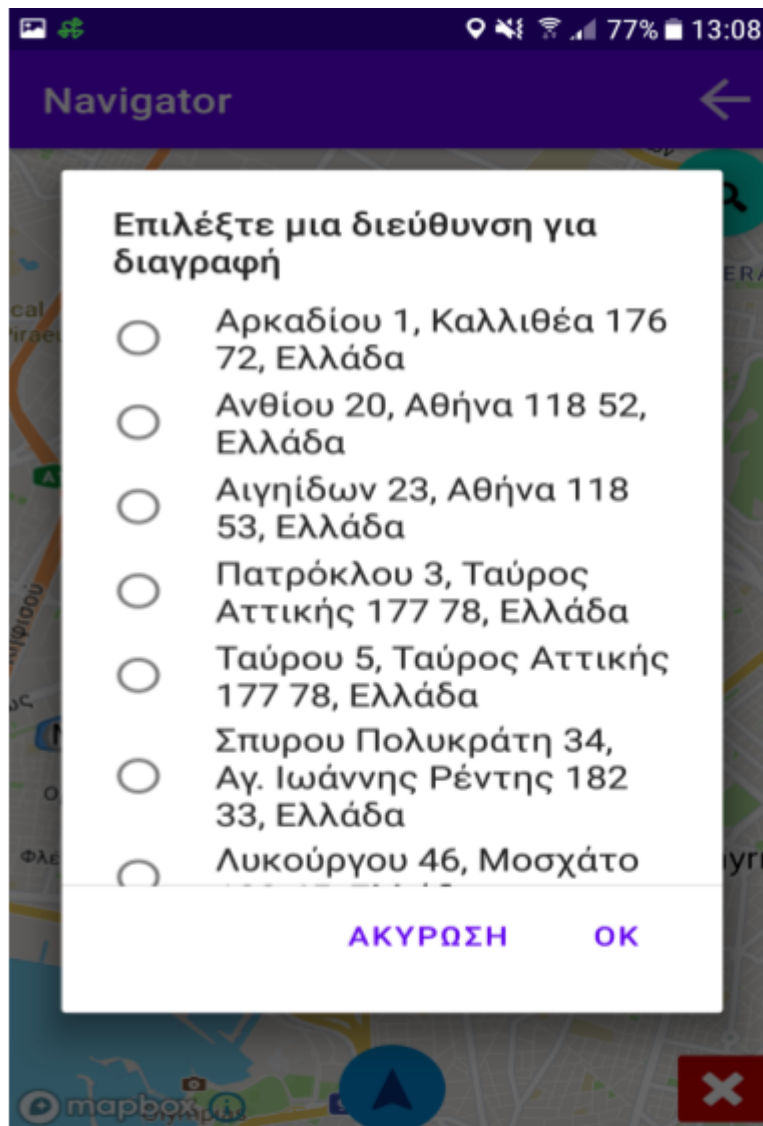
Εικόνα 31: Αναζήτηση οδού στο Android app




Εικόνα 32: Σχεδίαση διαδρομής στο Android app

Σε περίπτωση που ο οδηγός εισάγει μια διεύθυνση όπου τελικά δεν επιθυμεί να μεταβεί μπορεί να την σβήσει από την εφαρμογή μέσα από μια πολύ απλή διαδικασία.

Στο κάτω δεξιά μέρος της οθόνης υπάρχει ένα κόκκινο κουμπί . Πατώντας το κουμπί αυτό εμφανίζεται μια λίστα με όλες τις διευθύνσεις που έχει εισάγει. Επιλέγοντας μια διεύθυνση και πατώντας το OK, η διεύθυνση αφαιρείται από τον χάρτη και δημιουργείται εκ νέου η διαδρομή.

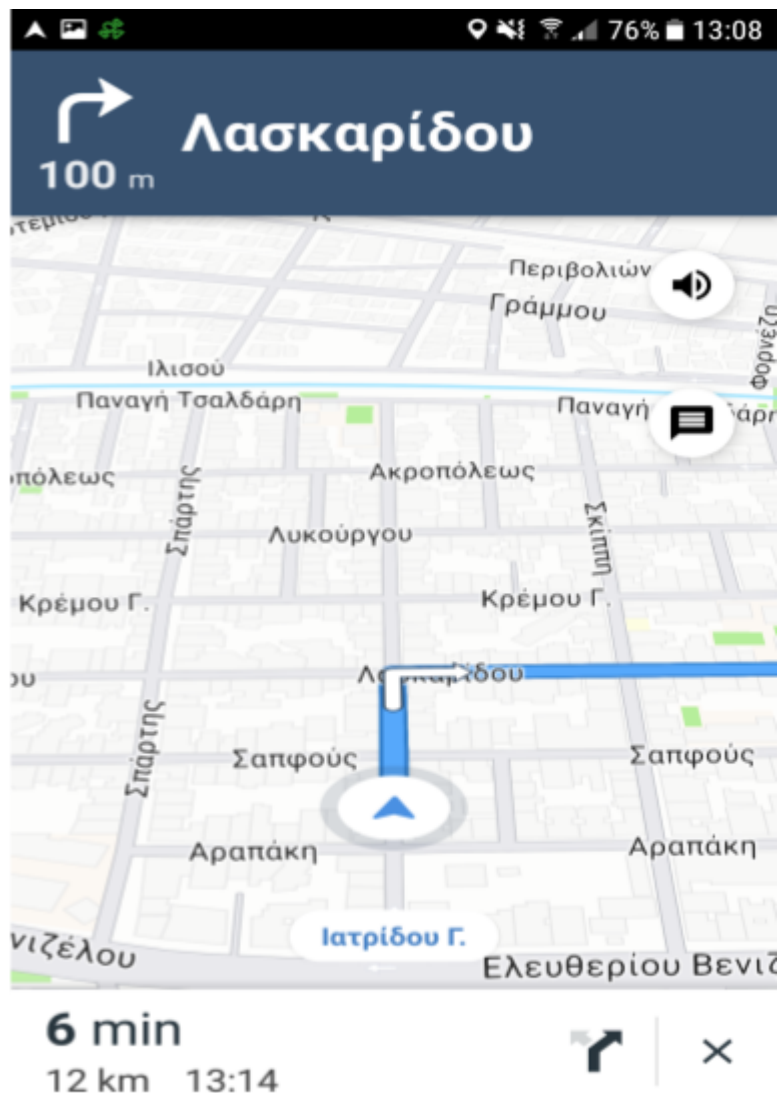


Εικόνα 33: Διαγραφή μιας διεύθυνσης από την λίστα στάσεων στο Android app

Μετά τον επιτυχή καθορισμό των στάσεων ο οδηγός είναι έτοιμος να ξεκινήσει την διαδρομή του. Το μόνο που χρειάζεται να κάνει είναι να πατήσει το κουμπί  της πλοήγησης και η εφαρμογή θα ξεκινήσει την διαδικασία. Κατά την διάρκεια της πλοήγησης, παρέχονται στον οδηγό πολύ χρήσιμες πληροφορίες με σκοπό την καλύτερη εμπειρία χρήσης της εφαρμογής. Οι πληροφορίες που παρέχει είναι:

- Απόσταση από την στροφή που πρέπει να ακολουθήσει και όνομα της οδού που θα εισέλθει.
- Συνολικά χιλιόμετρα διαδρομής
- Εκτιμώμενος χρόνος άφιξης στην επόμενη στάση
- Εκτιμώμενη ώρα άφιξης στην επόμενη στάση

Ακόμη, υπάρχει η επιλογή φωνητικών εντολών (στην αγγλική γλώσσα), η αναφορά μιας οδού ως κλειστή, ως μη αποδοτική ή ως μη επιτρεπτή για προσπέλαση. Η εικόνα 34 αποτελεί ένα στιγμιότυπο πλοήγησης όπου ο οδηγός θα πρέπει να στρίψει δεξιά σε 100 μέτρα στην οδό Λασκαρίδου, έχοντας να διανύσει συνολικά 12 χιλιόμετρα ενώ από την πρώτη στάση απέχει μόλις 6 λεπτά με εκτιμώμενη ώρα άφιξης σε αυτήν στις 1:14 το μεσημέρι.



Εικόνα 34: Πλοήγηση στο Android app





# Κεφάλαιο 6: Οδηγός Χρήσης του Web app

---

## 6.1 Εισαγωγή

Για την εύκολη πρόσβαση στο σύστημα, έχει αναπτυχθεί και μια web εφαρμογή προσφέροντας ευελιξία στον χρήστη. Υλοποιήθηκε στο Visual Studio Code, ένα λογισμικό που αναπτύχθηκε από την Microsoft και χρησιμοποιείται ευρέως για την ανάπτυξη εφαρμογών υποστηρίζοντας μια μεγάλη γκάμα γλωσσών προγραμματισμού (πχ Java, Javascript, C, C++, HTML CSS κλπ). Το backend κομμάτι της εφαρμογής είναι γραμμένο σε Javascript ενώ το frontend σε HTML/CSS.

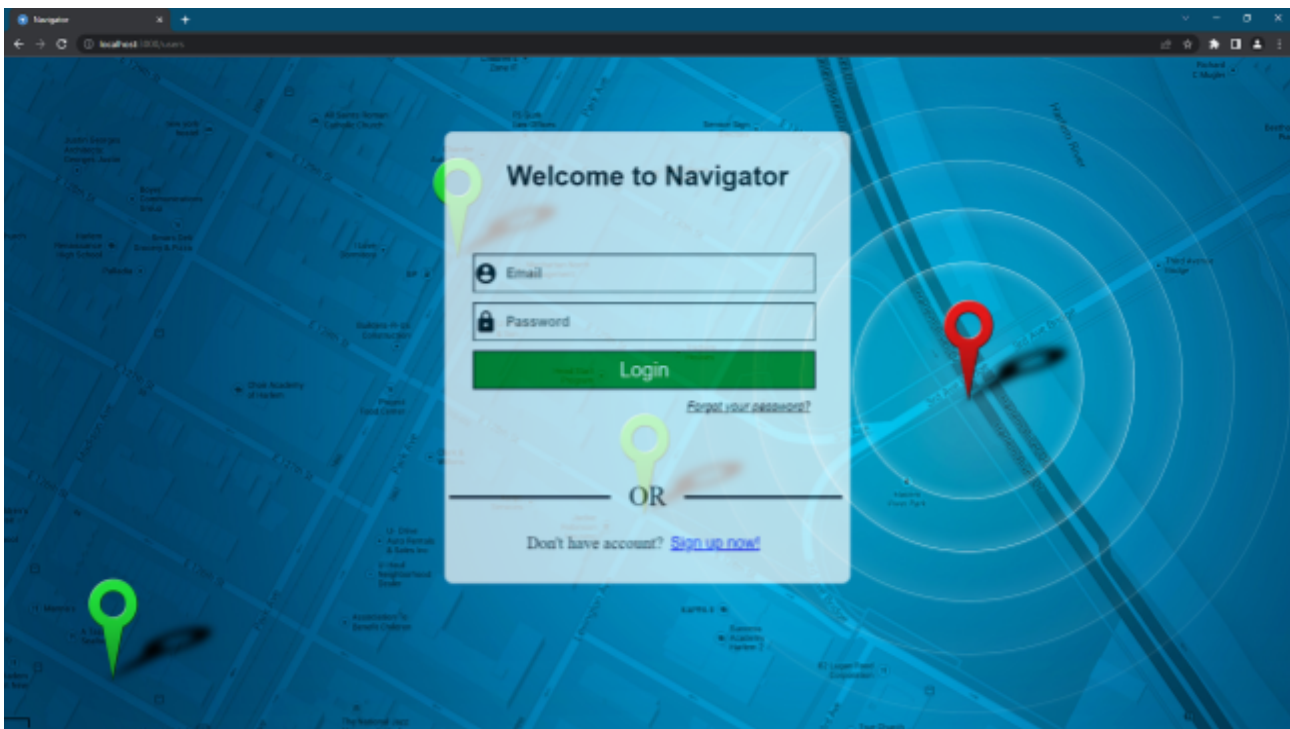
Σκοπός της web εφαρμογής είναι η δυνατότητα παροχής πρόσβασης από τον χρήστη των πληροφοριών που συλλέγει η android εφαρμογή. Μια βασική διαφορά της web με την android εφαρμογή είναι τα δικαιώματα που έχει ένας επισκέπτης. Στην android εφαρμογή ο κάθε οδηγός συνδέεται ως απλός χρήστης. Αντίθετα, στην web εφαρμογή υπάρχει η επιλογή εγγραφής ενός χρήστη ως διαχειριστή. Ένας διαχειριστής έχει πρόσβαση όχι μόνο στις πληροφορίες του λογαριασμού του, των οχημάτων του και των δρομολογίων που έχει ολοκληρώσει αλλά και στις αντίστοιχες πληροφορίες όλων των υπόλοιπων εγγεγραμμένων χρηστών. Επίσης ένας διαχειριστής μπορεί να διαγράψει όποια πληροφορία επιθυμεί από οποιονδήποτε χρήστη. Επιπλέον, έχει την δυνατότητα να επεξεργαστεί τα στοιχεία από το προφίλ κάποιου χρήστη ή κάποιου οχήματος. Τέλος εφόσον έχει πρόσβαση σε όλες τις πληροφορίες των χρηστών, έχει το δικαίωμα διαγραφής κάποιου οχήματος, χρήστη ή δρομολογίου. Από την άλλη πλευρά, ένας χρήστης μπορεί να εισέλθει στην web εφαρμογή για να αλλάξει τα προσωπικά του στοιχεία, τον κωδικό πρόσβασης του ή να διαγράψει τον λογαριασμό του. Ακόμη, έχει την επιλογή να επεξεργαστεί τις πληροφορίες των οχημάτων του, να προσθέσει ένα καινούργιο ή να διαγράψει κάποιο από αυτά. Τέλος, επιτρέπεται να δει ή να διαγράψει ένα δρομολόγιο που έχει ολοκληρώσει.

Ένα μεγάλο πλεονέκτημα της συγκεκριμένης εφαρμογής είναι η εμφάνιση στον χάρτη, για κάθε δρομολόγιο που πραγματοποιήθηκε, των στάσεων της διαδρομής. Ο χρήστης μπορεί να επιλέξει ένα δρομολόγιο από την λίστα του και με το πάτημα ενός κουμπιού να ανοίξει μια νέα καρτέλα στο πρόγραμμα περιήγησης του παρουσιάζοντας ένα στιγμιότυπο χάρτη όπου δείχνει το σημείο αναχώρησης και όλες τις στάσεις που περιλάμβανε η συγκεκριμένη διαδρομή. Θα πρέπει να σημειώσουμε ότι για κάθε δρομολόγιο καταγράφουμε την τοποθεσία αναχώρησης, την ημερομηνία και ώρα αναχώρησης, την τοποθεσία του προορισμού και την ημερομηνία και ώρα άφιξης στον προορισμό.

## 6.2 Είσοδος χρήστη

Η σελίδα εισόδου μας καλωσορίζει στη εφαρμογή. Όπως βλέπουμε στην εικόνα 35, αρχικά εμφανίζονται τα πεδία του email και του κωδικού πρόσβασης με τα αντίστοιχα εικονίδια με σκοπό την εύκολη και ευχάριστη χρήση της εφαρμογής. Οι περιπτώσεις που το σύστημα ορίζει ως λάθος στην εισαγωγή των στοιχείων είναι οι εξής:

- **Κενά πεδία**
- **Λανθασμένη μορφή email.** Το email θα πρέπει να έχει την σωστή μορφή που απαρτίζεται από 4 βασικά κομμάτια:
  - Όνομα χρήστη (Username)
  - Το σύμβολο @
  - Όνομα τομέα (Domain name)
  - Το σύμβολο .
- **Λανθασμένη μορφή κωδικού πρόσβασης.** Ο κωδικός πρόσβασης θα πρέπει να ξεκινάει με χαρακτήρα και να αποτελείται από τουλάχιστον 7 και το πολύ 14 χαρακτήρες.
- **Λανθασμένος κωδικός πρόσβασης.** Το email είναι σωστό αλλά ο κωδικός πρόσβασης λάθος.
- **Λανθασμένο email και κωδικός πρόσβασης.** Το email και ο κωδικός πρόσβασης δεν αντιστοιχούν σε κάποιο λογαριασμό.



*Εικόνα 35: Είσοδος χρήστη στο Web app*

Μετά την εισαγωγή των στοιχείων, ο χρήστης θα πρέπει να πατήσει το πράσινο κουμπί “LOGIN” για τη είσοδό του στο σύστημα. Σε αντίθεση με την Android εφαρμογή, στην Web εφαρμογή δεν υπάρχει περιορισμός όσον αφορά τις προσπάθειες εισόδου στο σύστημα. Δεν λαμβάνονται υπόψιν οι αποτυχημένες απόπειρες σύνδεσης και δεν εφαρμόζεται προσωρινό time out όπως είδαμε στην mobile έκδοση. Σε περίπτωση που κάποιος χρήστης ξεχάσει τον κωδικό πρόσβασης του έχει την δυνατότητα να ορίσει έναν νέο κωδικό μέσω μιας πολύ απλής διαδικασίας. Το μόνο που απαιτείται είναι η γνώση του

email του και τον αριθμό επαναφοράς (PIN). Βασικές προϋποθέσεις για την επιτυχή αλλαγή του κωδικού πρόσβασης είναι:

- Εισαγωγή email σε σωστή μορφή
- Εισαγωγή αριθμού επαναφοράς που αντιστοιχεί στο σωστό email
- Εισαγωγή κωδικού πρόσβασης που να ξεκινάει με χαρακτήρα και να αποτελείται από τουλάχιστον 7 και το πολύ 14 χαρακτήρες.
- Τα πεδία των κωδικών πρόσβασης να περιέχουν τον ίδιο κωδικό

Στην Εικόνα 36 βλέπουμε το στιγμιότυπο από την οθόνη ενός χρήστη την στιγμή που πάτησε το κουμπί “Forgot your password?”.



Εικόνα 36: Ανάκτηση κωδικού πρόσβασης στο Web app

### 6.3 Εγγραφή χρήστη

Πέρα από την δυνατότητα εισόδου, ο επισκέπτης μπορεί σε περίπτωση που δεν έχει λογαριασμό να πατήσει το κουμπί “Sign up now!” που φαίνεται στο κάτω μέρος της οθόνης όπου θα τον ανακατευθύνει σε άλλη σελίδα για την δημιουργία καινούργιου λογαριασμού. Η δημιουργία ενός καινούργιου λογαριασμού πραγματοποιείται σε 2 φάσεις. Η **πρώτη** φάση, όπως και στην Android εφαρμογή, αφορά την εισαγωγή πληροφοριών που σχετίζονται τον οδηγό ενώ η **δεύτερη** φάση αφορά τις πληροφορίες του οχήματος.

### 6.3.1 Δημιουργία προφίλ οδηγού

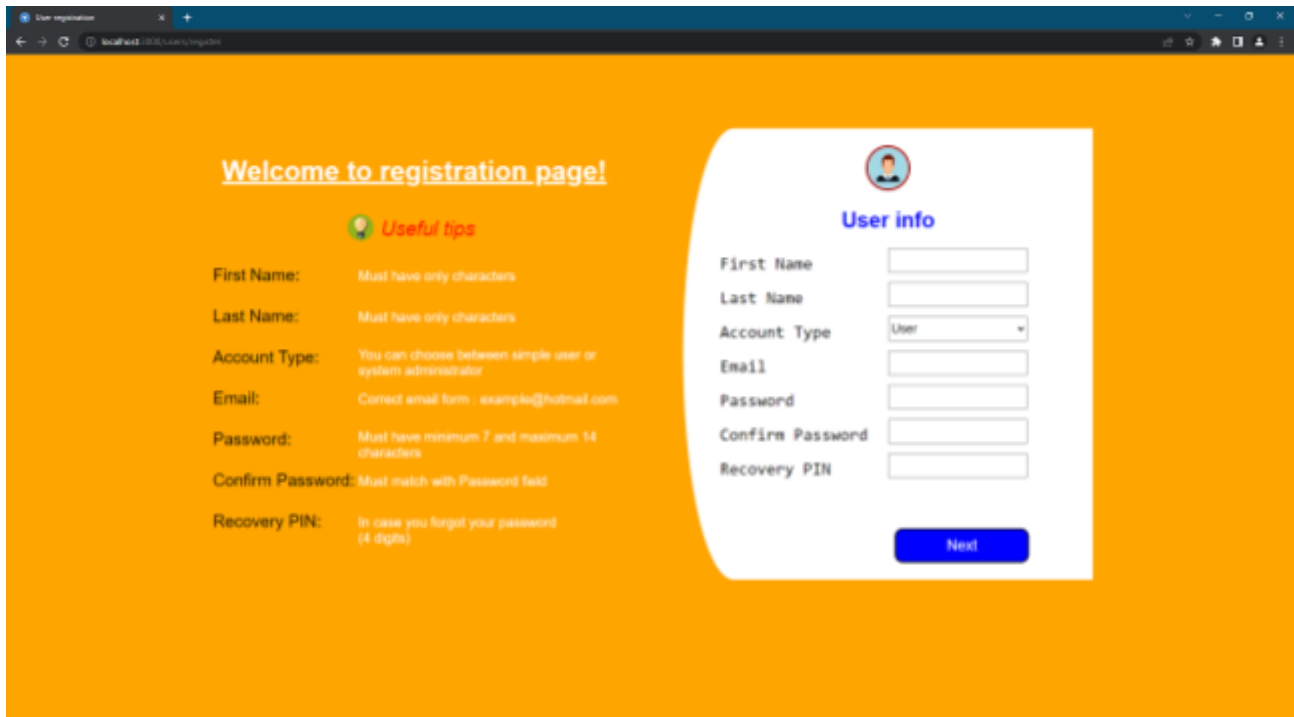
Σε αυτή την υποενότητα θα αναφερθούμε στην πρώτη φάση εγγραφής. Αρχικά ο οδηγός θα πρέπει να εισάγει τις προσωπικές του πληροφορίες. Πιο αναλυτικά τα στοιχεία που είναι υποχρεωμένος να εισάγει είναι τα εξής:

- Όνομα χρήστη
- Επώνυμο χρήστη
- Είδος λογαριασμού
- Email
- Κωδικός πρόσβασης
- Επιβεβαίωση κωδικού πρόσβασης
- Αριθμός επαναφοράς

Το είδος λογαριασμού αναφέρεται στα δικαιώματα που έχει ο χρήστης στην εφαρμογή. Μπορεί να επιλέξει ανάμεσα σε απλό χρήστη και διαχειριστή. Ο αριθμός επαναφοράς είναι ένα τετραψήφιο νούμερο που εισάγει ο χρήστης κατά την δημιουργία του λογαριασμού του και χρησιμοποιείται μόνο σε περίπτωση που έχει ξεχάσει τον κωδικό πρόσβασης του. Πιο συγκεκριμένα , γνωρίζοντας το email και τον αριθμό επαναφοράς μπορεί να αλλάξει τον κωδικό πρόσβασης του. Για κάθε ένα από τα παραπάνω πεδία που αφορούν τον οδηγό υπάρχουν ορισμένοι κανόνες που οφείλει να τηρήσει ο ίδιος:

1. Δεν μπορούν να υπάρχουν κενά πεδία.
2. Το όνομα χρήστη περιέχει μόνο χαρακτήρες.
3. Το επώνυμο χρήστη περιέχει μόνο χαρακτήρες.
4. Το email να είναι στην σωστή μορφή.
5. Δεν μπορούν να υπάρχουν εγγεγραμμένοι χρήστες με το ίδιο email.
6. Ο κωδικός πρόσβασης να ξεκινάει με χαρακτήρα και να αποτελείται από τουλάχιστον 7 και το πολύ 14 χαρακτήρες.
7. Τα πεδία των κωδικών πρόσβασης να περιέχουν τον ίδιο κωδικό.
8. Ο αριθμός επαναφοράς να αποτελείται μόνο από τέσσερις ακέραιους αριθμούς.

Στην εικόνα 37 φαίνεται η σελίδα εγγραφής ενός οδηγού. Ολοκληρώνοντας την εισαγωγή των στοιχείων του και πατώντας το κουμπί “Next” μεταβαίνει στην σελίδα εγγραφής οχήματος.



Εικόνα 37: Εγγραφή χρήστη στο Web app

### 6.3.2 Δημιουργία προφίλ οχήματος

Μετά την επιτυχή εισαγωγή των προσωπικών πληροφοριών του ο χρήστης μεταβαίνει στην σελίδα εισαγωγής των πληροφοριών του οχήματος που έχει στην κατοχή του για την εκτέλεση των δρομολογίων. Οι πληροφορίες αυτές είναι:

- Ο κατασκευαστής του οχήματος
- Το μοντέλο του οχήματος
- Ο αριθμός πινακίδας του
- Η κατηγορία του οχήματος
- Το χρώμα του
- Τα κυβικά του
- Οι ίπποι του
- Η ημερομηνία κατασκευής του

Και σε αυτή την διαδικασία, για την επιτυχή εγγραφή του οχήματος ο χρήστης θα πρέπει να είναι ιδιαίτερα προσεκτικός σε ορισμένα σημεία κατά την εισαγωγή των δεδομένων. Συγκεκριμένα:

1. Δεν μπορούν να υπάρχουν κενά πεδία.

2. Το όνομα του κατασκευαστή περιέχει μόνο χαρακτήρες.
3. Το μοντέλο του οχήματος περιέχει μόνο χαρακτήρες και αριθμούς.
4. Ο αριθμός πινακίδας του οχήματος αποτελείται μόνο από τέσσερις λατινικούς χαρακτήρες που υπάρχουν και στο ελληνικό αλφάβητο, το σύμβολο – και από 3 ψηφία.
5. Δεν μπορούν να υπάρχουν στο σύστημα δύο ή παραπάνω οχήματα με τον ίδιο αριθμό πινακίδας.
6. Το πεδίο του χρώματος του οχήματος περιέχει μόνο χαρακτήρες.
7. Το πεδίο των κυβικών του αποτελείται μόνο από αριθμούς.
8. Το πεδίο των ίππων του αποτελείται μόνο από αριθμούς.
9. Στο πεδίο της ημερομηνίας, ο αριθμός που αντιστοιχεί στην ημέρα δεν μπορεί να είναι μεγαλύτερος του 31 και ο αριθμός που αντιστοιχεί στον μήνα δεν μπορεί να είναι μεγαλύτερος του 12.

Εικόνα 38: Εγγραφή οχήματος στο Web app

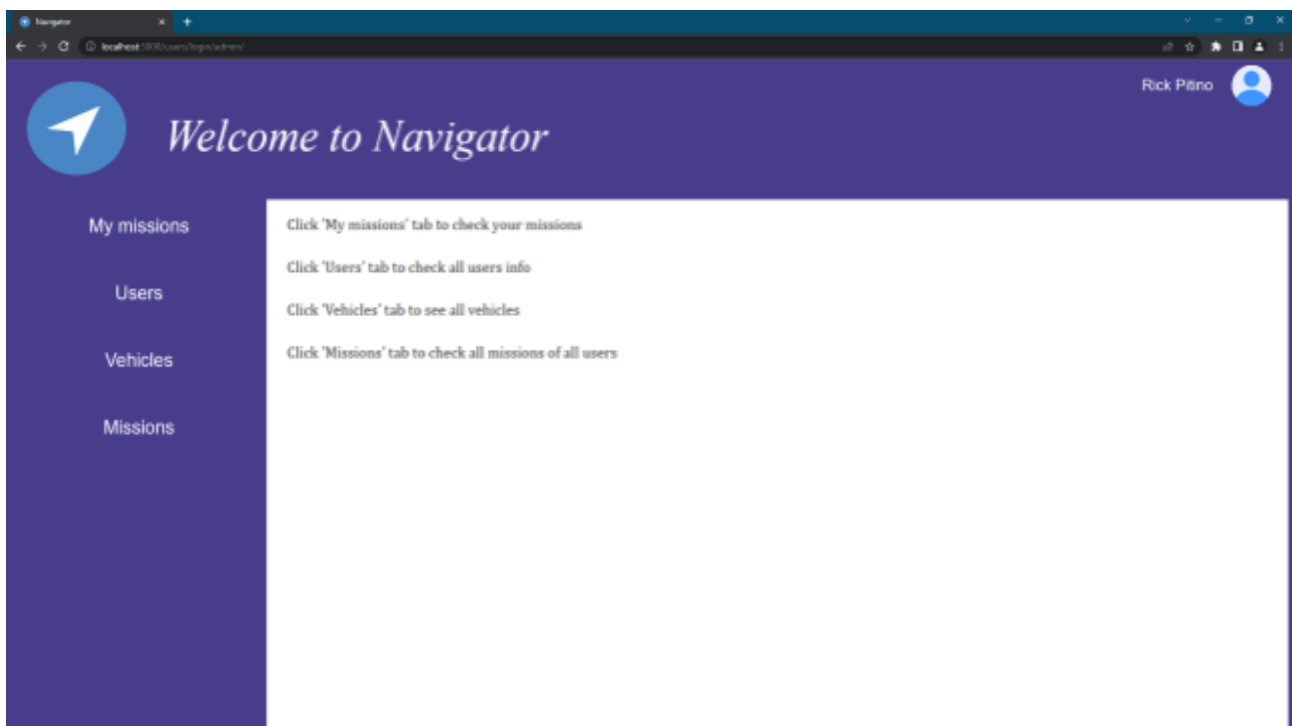
Η εγγραφή ολοκληρώνεται επιτυχώς με το πάτημα του κουμπιού “Register” όπου καταχωρούνται στην βάση δεδομένων τα στοιχεία του χρήστη. Στη συνέχεια η εφαρμογή τον ανακατευθύνει στην αρχική σελίδα όπου πλέον μπορεί να κάνει είσοδο με τα στοιχεία που μόλις δημιούργησε.

## 6.4 Αρχική σελίδα διαχειριστή

Μετά την εισαγωγή του email και του κωδικού πρόσβασης από τον χρήστη, η εφαρμογή ελέγχει την ιδιότητα του (απλός χρήστης / διαχειριστής). Ο διαχειριστής έχει το δικαίωμα πρόσβασης:

- Στο προφίλ του
- Στο προφίλ των οχημάτων του
- Στην λίστα των δρομολογίων του
- Στο προφίλ οποιουδήποτε οδηγού
- Στο προφίλ οποιουδήποτε οχήματος
- Στην λίστα όλων των δρομολογίων

Η εικόνα 39 αποτελεί ένα στιγμιότυπο από την είσοδο του Rick Pitino στην εφαρμογή. Όπως μπορούμε να διακρίνουμε, ο συγκεκριμένος χρήστης είναι διαχειριστής αφού μπορεί να αποκτήσει πρόσβαση σε όποια πληροφορία επιθυμεί. Στην αρχική σελίδα υπάρχει ένα άσπρο πλαίσιο μέσα στο οποίο εξηγείται η λειτουργία των πλαϊνών κουμπιών ενώ πάνω δεξιά εμφανίζεται πάντα ένα κουμπί που περιλαμβάνει το ονοματεπώνυμο του οδηγού που εισέρχεται. Τέλος το μεγάλο εικονίδιο της εφαρμογής που βρίσκεται στα αριστερά, αποτελεί ταυτόχρονα και κουμπί αρχικής οθόνης επιστρέφοντας τον χρήστη σε αυτή ακριβώς την σελίδα κάθε φορά που το πατάει.

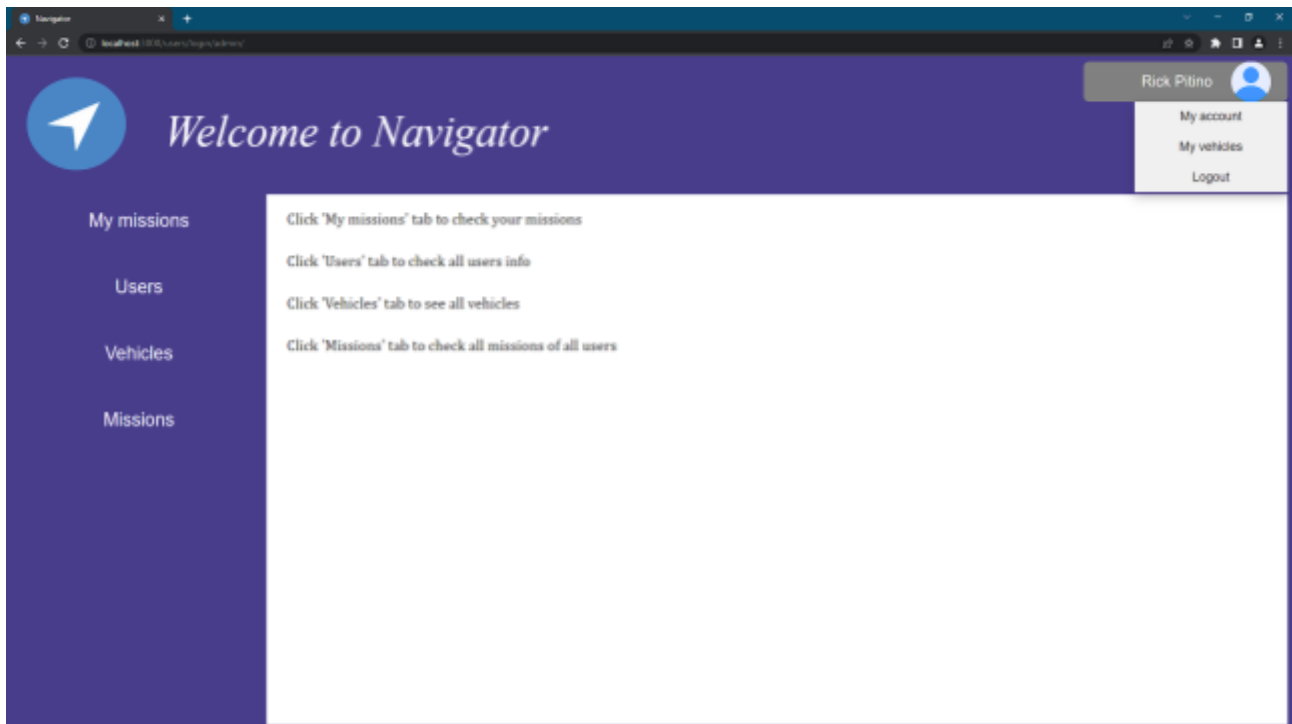


Εικόνα 39: Αρχική σελίδα διαχειριστή στο Web app



Όπως αναφέραμε και πιο πάνω, στην αρχική σελίδα υπάρχει ένα κουμπί στο πάνω δεξιά μέρος της οθόνης που περιέχει το ονοματεπώνυμο του οδηγού. Πατώντας το κουμπί αυτό εμφανίζεται ένα μενού με 3 επιλογές:

1. **My account** για την μετάβαση στην σελίδα που περιέχει τις προσωπικές του πληροφορίες.
2. **My vehicles** για μετάβαση στην σελίδα που περιέχει τα οχήματά του.
3. **Logout** για αποσύνδεση από τον λογαριασμό και επιστροφή στην σελίδα εισόδου.

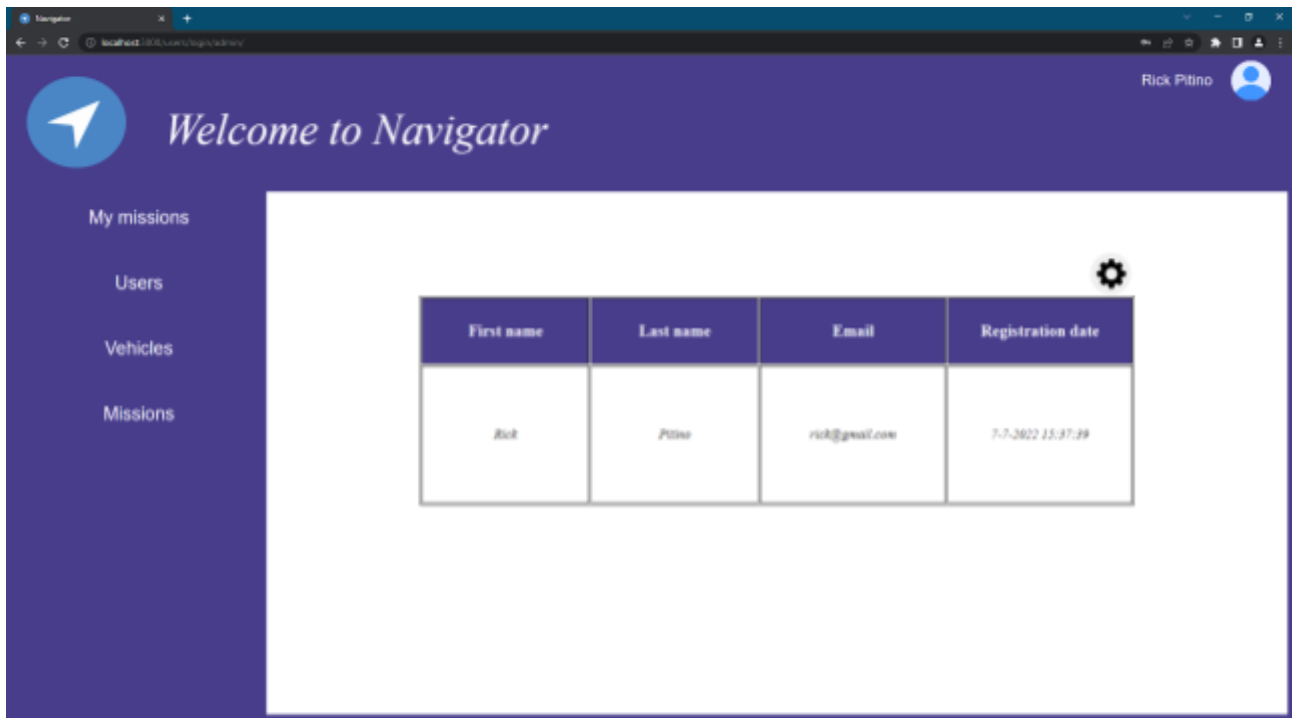


Εικόνα 40: Dropdown menu της αρχικής οθόνης στο Web app

#### 6.4.1 Πληροφορίες λογαριασμού


Σε αυτή την υποενότητα βλέπουμε την σελίδα που θα εμφανιστεί στον χρήστη όταν πατήσει την επιλογή “Ο Λογαριασμός μου” από το μενού της αρχικής σελίδας. Η σελίδα αυτή περιέχει πληροφορίες που αφορούν αποκλειστικά τον ίδιο. Πιο συγκεκριμένα, οι πληροφορίες αυτές είναι:

- Το όνομα του χρήστη
- Το επώνυμο του χρήστη
- Το email του
- Η ημερομηνία που πραγματοποίησε την εγγραφή του



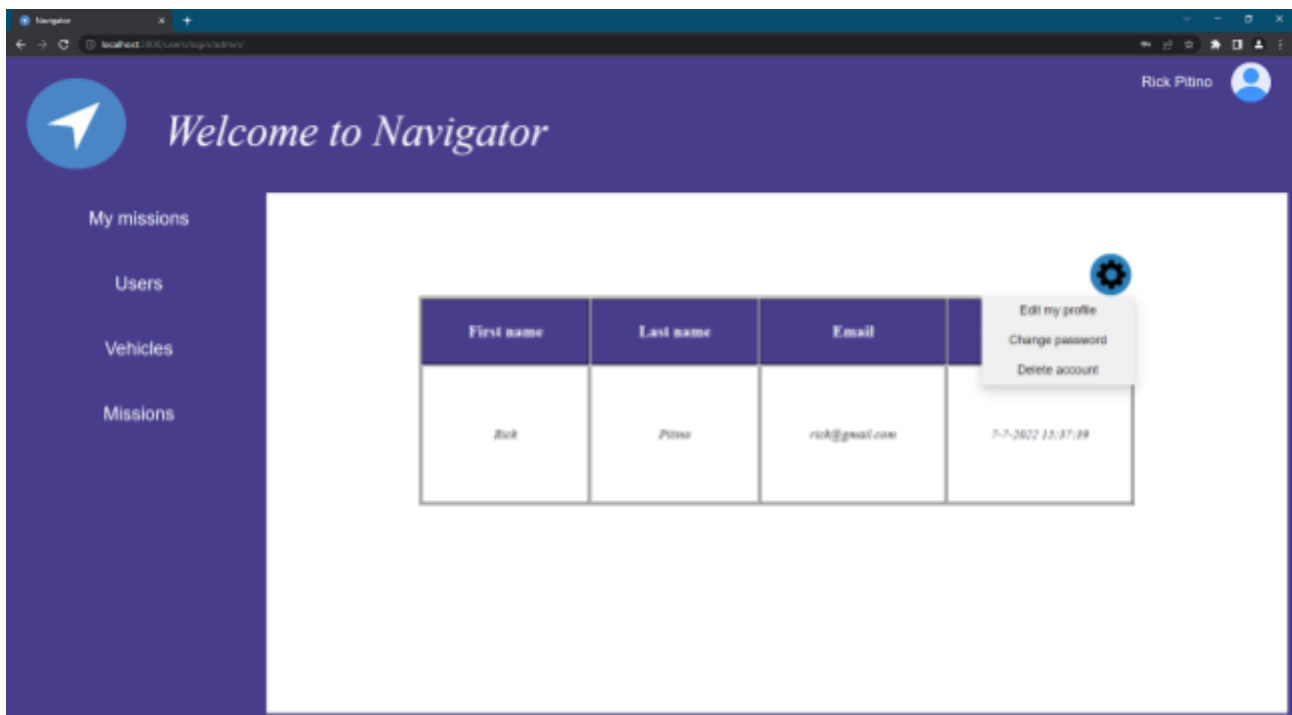
Εικόνα 41: Σελίδα στοιχείων χρήστη στο Web app

Στην εικόνα 41 βλέπουμε ένα στιγμιότυπο από την σελίδα του Rick Pitino με το email του να είναι “[rick@gmail.com](mailto:rick@gmail.com)” ενώ η ημερομηνία εγγραφής του είναι 7 Ιουλίου 2022 και ώρα 3:37 το μεσημέρι.

Στην συγκεκριμένη εικόνα μπορούμε να διακρίνουμε ένα κουμπί  που βρίσκεται ακριβώς πάνω από τον πίνακα με τα στοιχεία του οδηγού. Το κουμπί αυτό αποτελεί συντόμευση ενός μενού. Με το πάτημα από τον χρήστη του κουμπιού εμφανίζονται 3 επιλογές:

1. **Edit my profile** Για την επεξεργασία των προσωπικών του στοιχείων.
2. **Change password** για την αλλαγή του κωδικού πρόσβασης.
3. **Delete account** για την οριστική διαγραφή των στοιχείων του οδηγού από την βάση δεδομένων.

Στην εικόνα 42 ο χρήστης Rick Pitino πάτησε το κουμπί που περιγράψαμε παραπάνω και του εμφάνισε το μενού.



Εικόνα 42: Dropdown menu της σελίδας χρήστη στο Web app

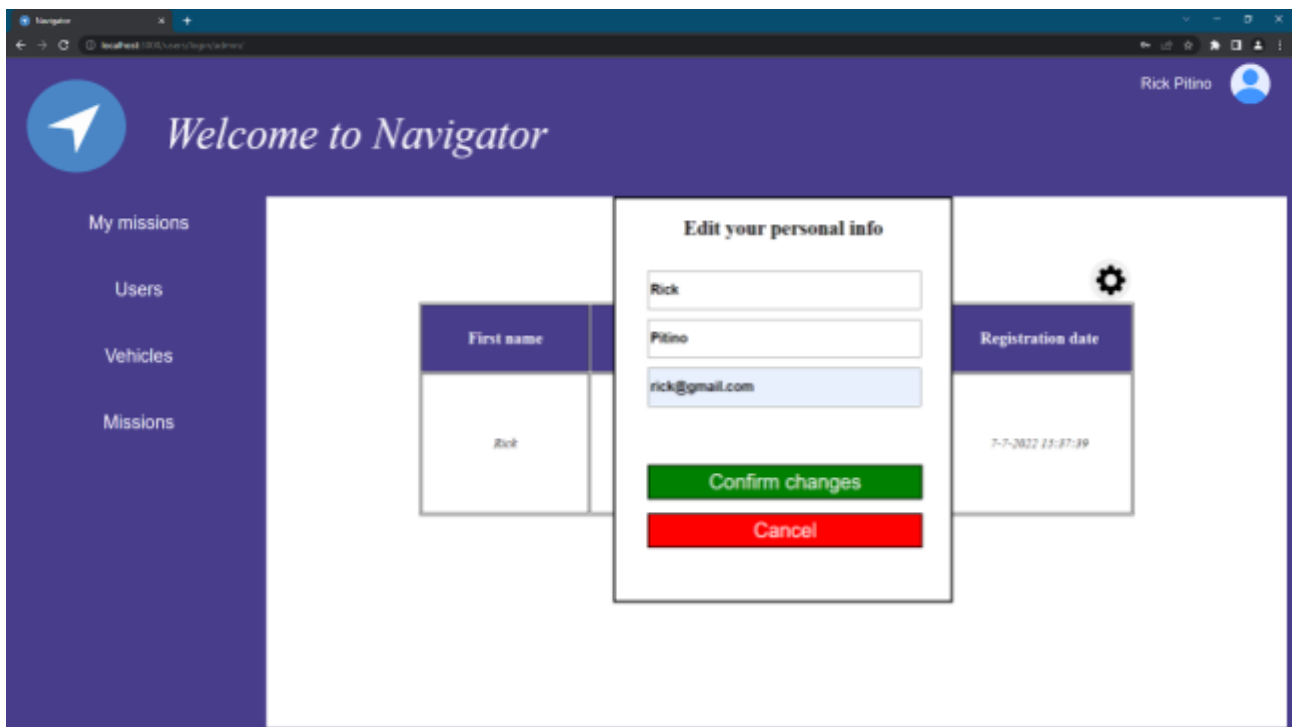
Μια μεγάλη διαφορά της Android με την Web εφαρμογή είναι ότι στην πρώτη ο οδηγός δεν έχει την δυνατότητα να αλλάξει τα προσωπικά του στοιχεία. Το κενό αυτό έρχεται να αναπληρώσει η δεύτερη με ένα πλήρες μενού δίνοντας την επιλογή στον κάθε οδηγό είτε είναι απλός χρήστης είτε διαχειριστής να αλλάξει κάθε είδους πληροφορία που σχετίζεται με τον ίδιο και τα οχήματά του. Πατώντας από το μενού την επιλογή “Edit my profile” ανοίγει ένας πίνακας επιτρέποντας τον να αλλάξει:

- Το όνομα του
- Το επώνυμο του
- Το email του

Για κάθε ένα από τα παραπάνω πεδία που αφορούν τον οδηγό υπάρχουν ορισμένοι κανόνες που οφείλει να τηρήσει ο ίδιος:

1. Δεν μπορούν να υπάρχουν κενά πεδία.
2. Το όνομα χρήστη περιέχει μόνο χαρακτήρες.
3. Το επώνυμο χρήστη περιέχει μόνο χαρακτήρες.
4. Το email να είναι στην σωστή μορφή.
5. Δεν μπορούν να υπάρχουν εγγεγραμμένοι χρήστες με το ίδιο email.

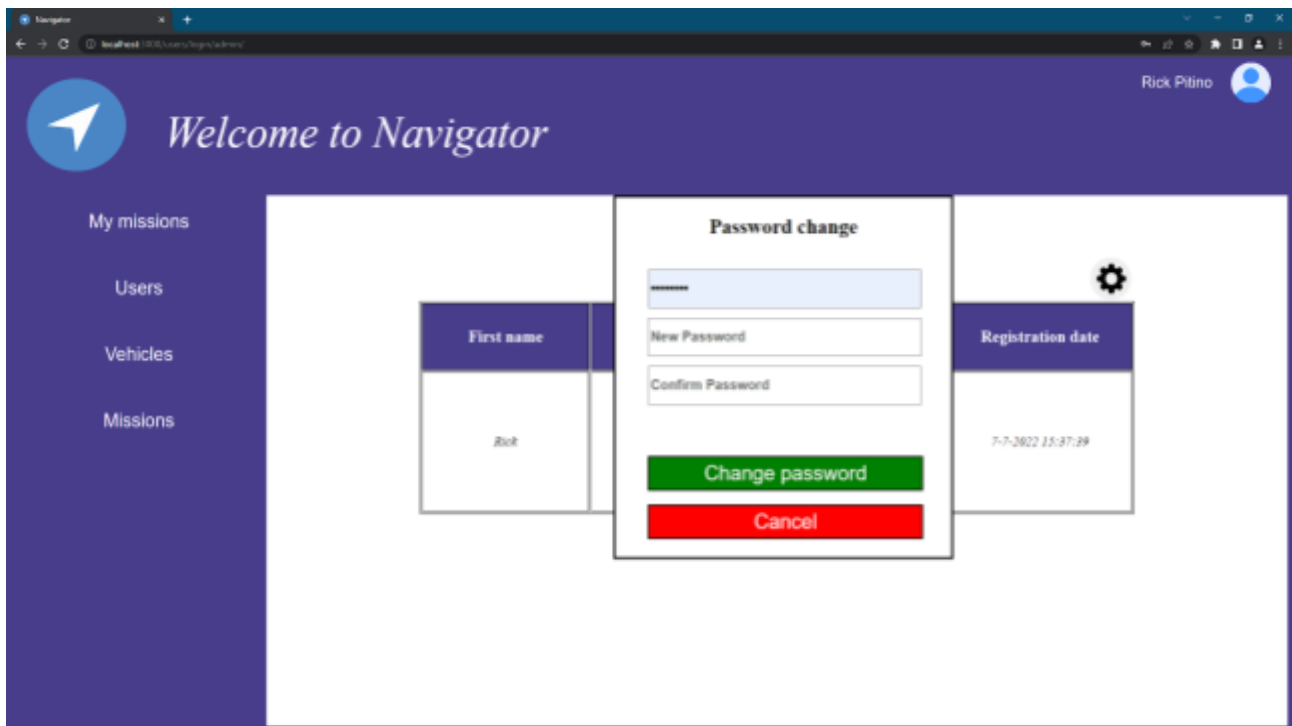
Η εφαρμογή από προεπιλογή συμπληρώνει αυτόματα στο κάθε πεδίο τις αντίστοιχες πληροφορίες που υπάρχουν ήδη στην βάση δεδομένων για τον χρήστη. Στην εικόνα 43, ο διαχειριστής Rick Pitino πάτησε την επιλογή “Edit my profile” και του εμφανίστηκε ο πίνακας επεξεργασίας των στοιχείων. Βλέπουμε ότι η εφαρμογή έλαβε από την βάση δεδομένων το όνομα, το επώνυμο και το email του και τα τοποθέτησε στα κατάλληλα πεδία. Η αυτοματοποιημένη αυτή διαδικασία γίνεται διότι ο χρήστης ίσως να μην θέλει να αλλάξει όλες τις πληροφορίες του καθώς σε αυτή την περίπτωση είναι υποχρεωμένος να πληκτρολογήσει στοιχεία που δεν επιθυμεί να αλλάξουν. Με αυτή την μέθοδο το μόνο που πρέπει να κάνει είναι να σβήσει και να συμπληρώσει τα πεδία που θέλουν αλλαγή με τα υπόλοιπα στοιχεία να παραμένουν αναλλοίωτα. Η διαδικασία ολοκληρώνεται με το πάτημα του πράσινου κουμπιού “Confirm changes” όπου τα δεδομένα στέλνονται στην βάση δεδομένων για αποθήκευση. Φυσικά υπάρχει και η δυνατότητα ακύρωσης της επεξεργασίας με το πάτημα του κόκκινου κουμπιού “Cancel”.



Εικόνα 43: Επεξεργασία στοιχείων χρήστη στο Web app

Η δεύτερη επιλογή από το μενού στην σελίδα του χρήστη είναι η “Change password”. Η αλλαγή κωδικού πρόσβασης αποτελεί μια εξαιρετικά απλή διαδικασία. Όπως έχουμε παρατηρήσει σχεδόν σε όλες τις γνωστές εφαρμογές που υπάρχουν αυτή την στιγμή στο internet, όταν ένας χρήστης επιθυμεί να αλλάξει τον κωδικό πρόσβασης του λογαριασμού του το μόνο που χρειάζεται να γνωρίζει είναι ο τρέχων κωδικός του. Σε αυτή την λογική βασίζεται και η συγκεκριμένη εφαρμογή. Σε αυτό το σημείο είμαστε υποχρεωμένοι να τονίσουμε ότι δεν απαιτείται η γνώση του αριθμού επαναφοράς (PIN) που γνωρίσαμε στην σελίδα εισόδου διότι το άτομο έχει πραγματοποιήσει επιτυχή είσοδο στον λογαριασμό του. Ο αριθμός επαναφοράς χρησιμοποιείται μόνο σε περίπτωση που δεν μπορεί να εισέλθει στον λογαριασμό του και θέλει να αλλάξει τον κωδικό πρόσβασης από την σελίδα εισόδου. Στην

εικόνα 44 φαίνεται το στιγμιότυπο στην στιγμή που ο Rick Pitino πάτησε το κουμπί “Change password”.



Εικόνα 44: Αλλαγή κωδικού πρόσβασης από την σελίδα χρήστη στο Web app

Να διευκρινίσουμε ότι στην εικόνα 44 το πρώτο πεδίο που αντιστοιχεί στον τρέχον κωδικό πρόσβασης του χρήστη είναι ήδη συμπληρωμένο. Αυτό συμβαίνει διότι το άτομο κατά την είσοδο του στην εφαρμογή επέλεξε να αποθηκευτεί ο κωδικός πρόσβασης από το πρόγραμμα περιήγησης. Δεν είναι αποτέλεσμα της εφαρμογής μας αλλά του συγκεκριμένου προγράμματος που επέλεξε ο χρήστης για να εισέλθει σε αυτήν.

#### 6.4.2 Πληροφορίες οχήματος

Σε αυτή την υποενότητα βλέπουμε την σελίδα που θα εμφανιστεί στον χρήστη όταν πατήσει την επιλογή “My vehicles” από το μενού της αρχικής σελίδας. Η σελίδα αυτή περιέχει πληροφορίες που αφορούν τα οχήματα που έχει στην κατοχή του. Πιο συγκεκριμένα, οι πληροφορίες αυτές είναι:

- Ο κατασκευαστής του οχήματος
- Το μοντέλο του οχήματος
- Ο αριθμός πινακίδας του
- Η κατηγορία του οχήματος
- Το χρώμα του

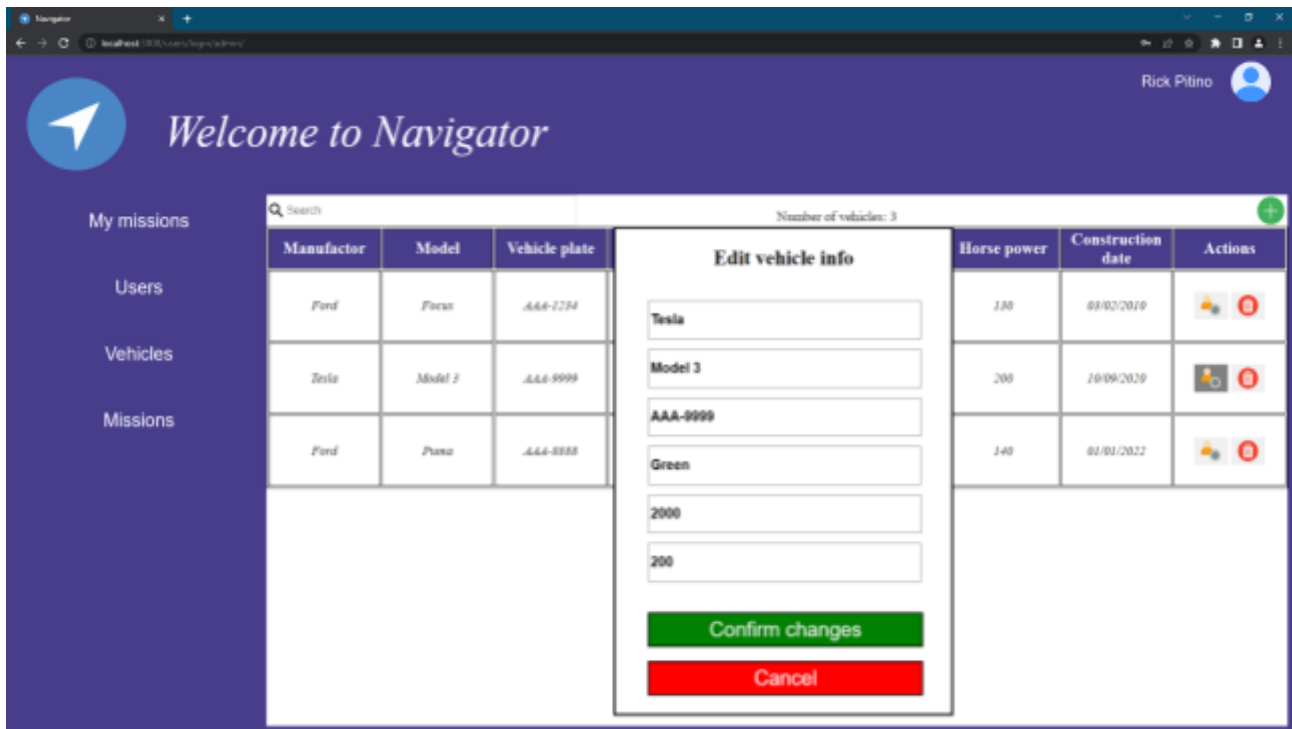
- Τα κυβικά του
- Οι ίπποι του
- Η ημερομηνία κατασκευής του

Manufacturer	Model	Vehicle plate	Type	Color	CC	Horse power	Construction date	Actions
Ford	Focus	AAA-1234	Car	Black	1900	130	03/02/2010	
Tesla	Model 3	AAA-9999	Car	Green	2000	200	10/09/2020	
Ford	Focus	AAA-8888	Car	Red	1400	140	01/01/2022	

Εικόνα 45: Σελίδα στοιχείων οχημάτων ενός χρήστη στο Web App

Στην εικόνα 45 μπορούμε να παρατηρήσουμε μια ακόμη στήλη στον πίνακα των οχημάτων με όνομα Actions. Η στήλη αυτή δίνει την επιλογή στον χρήστη να επεξεργαστεί τα στοιχεία ενός οχήματος ή να διαγράψει κάποιο από αυτά. Βασική προϋπόθεση για την διαγραφή ενός οχήματος είναι η ύπαρξη τουλάχιστον ενός ακόμη στην κατοχή του. Δεν μπορεί να υπάρξει χρήστης χωρίς να κατέχει όχημα.

Πιο συγκεκριμένα, με το πάτημα από τον χρήστη του κουμπιού διαγράφεται από το σύστημα το όχημα της αντίστοιχης σειράς ενώ πατώντας το κουμπί εμφανίζεται ένας πίνακας για την επεξεργασία των πληροφοριών του οχήματος. Όπως είδαμε και στην επεξεργασία των δεδομένων ενός χρήστη, η εφαρμογή συλλέγει αυτόματα τα υπάρχοντα δεδομένα και τα συμπληρώνει στα αντίστοιχα πεδία. Στην εικόνα 46 ο χρήστης πάτησε να επεξεργαστεί τις πληροφορίες ενός Tesla Model 3 με αριθμό πινακίδας AAA-9999 χρώματος πράσινου. Τα κυβικά του ανέρχονται στα 2000 με 200 ίππους. Η διαδικασία ολοκληρώνεται με το πάτημα του πράσινου κουμπιού “Confirm changes” όπου τα δεδομένα στέλνονται στην βάση δεδομένων για αποθήκευση. Φυσικά υπάρχει και η δυνατότητα ακύρωσης της επεξεργασίας με το πάτημα του κόκκινου κουμπιού “Cancel”.

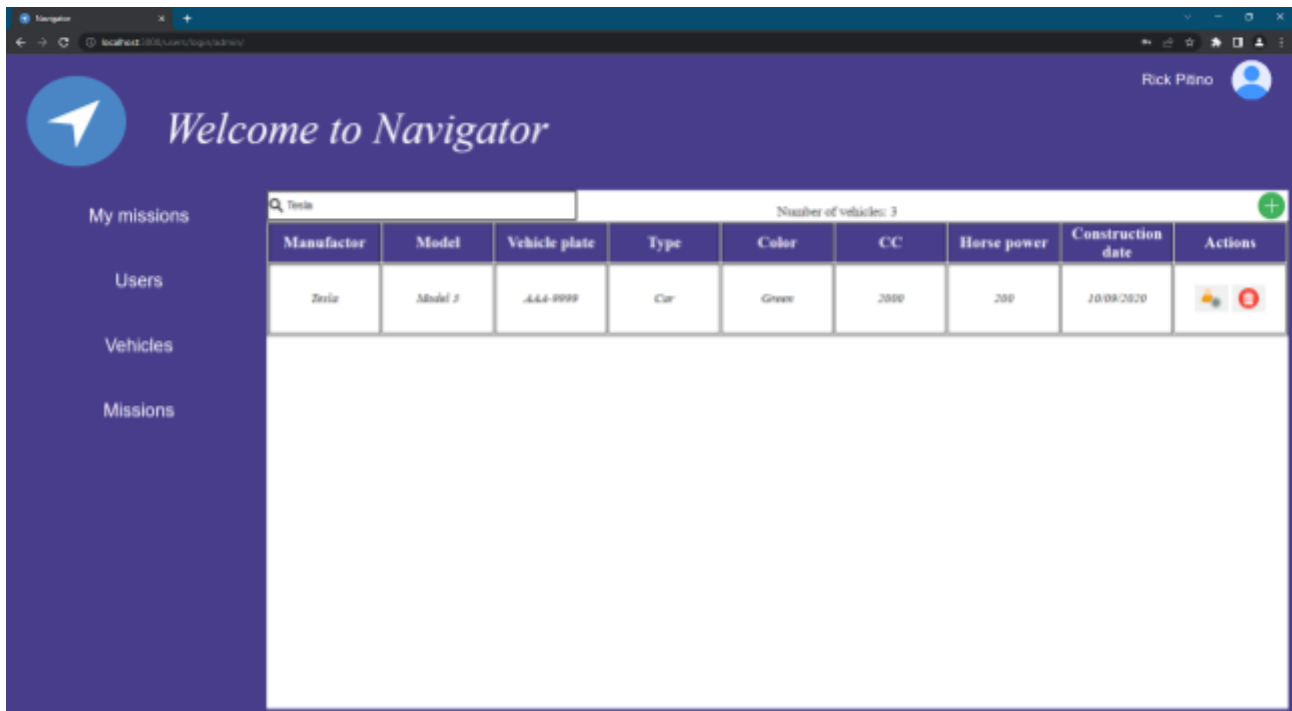


Εικόνα 46: Επεξεργασία στοιχείων οχήματος στο Web app

Πάνω από τον πίνακα των οχημάτων υπάρχει μια μπάρα με τα εξής στοιχεία:


- Αναζήτηση ενός οχήματος
- Μετρητής για τον αριθμό των οχημάτων
- Προσθήκη νέου οχήματος

Όταν λέμε αναζήτηση ενός οχήματος αναφερόμαστε στην δυνατότητα που έχει το κάθε άτομο να βρει ένα όχημα μέσα στην λίστα του συνόλου των οχημάτων πληκτρολογώντας στο πεδίο αναζήτησης οποιαδήποτε πληροφορία σχετίζεται με αυτό. Στο παρακάτω στιγμιότυπο της εικόνας 47, ο χρήστης αναζητά το Tesla όχημα του χωρίς να γνωρίζει κάποια άλλη πληροφορία. Το μόνο που έκανε είναι να πληκτρολογήσει την λέξη “Tesla” και η εφαρμογή πραγματοποίησε αναζήτηση στην λίστα για οχήματα Tesla. Η λειτουργία αναζήτησης δεν δουλεύει μόνο εισάγοντας το όνομα του κατασκευαστή ενός οχήματος. Η εφαρμογή πραγματοποιεί αναζήτηση σε ολόκληρο τον πίνακα. Συνεπώς σε περίπτωση που ένας χρήστης δεν γνωρίζει το όνομα του κατασκευαστή αλλά το μοντέλο του ή κάποια άλλη από τις πληροφορίες ενός οχήματος μπορεί απλά να το εισάγει στο πεδίο της αναζήτησης για την εύρεσή του.



Εικόνα 47: Αναζήτηση ενός οχήματος στο Web App

Στην μέση της μπάρας υπάρχει η εξής πληροφορία: “Number of vehicles: 3”. Για την εύκολη διαχείριση του συνόλου των οχημάτων παρέχεται στον οδηγό ο ακριβής αριθμός των οχημάτων που έχει στην κατοχή του.

Τέλος στην δεξιά πλευρά της μπάρας υπάρχει το κουμπί . Με το πάτημα αυτού του κουμπιού δίνεται η επιλογή προσθήκης ενός νέου οχήματος για τον τρέχον χρήστη. Οι πληροφορίες που οφείλει να εισάγει είναι:

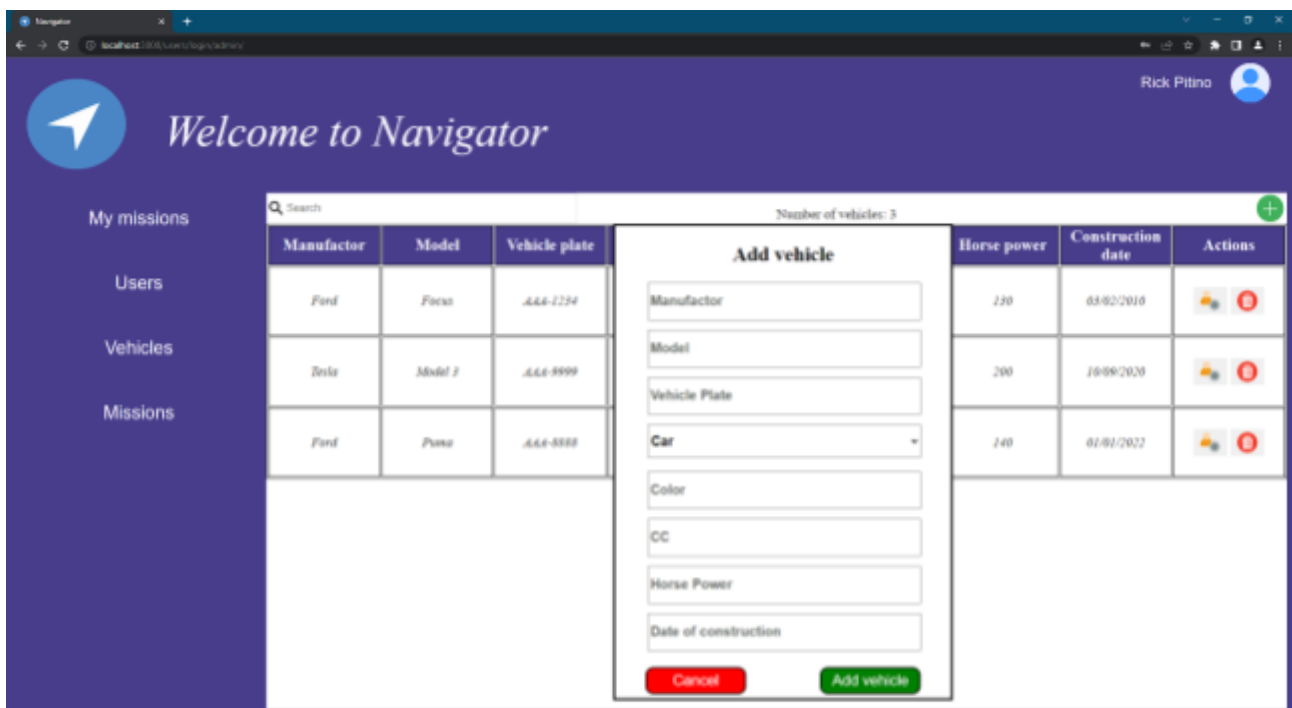
- Ο κατασκευαστής του οχήματος
- Το μοντέλο του οχήματος
- Ο αριθμός πινακίδας του
- Η κατηγορία του οχήματος
- Το χρώμα του
- Τα κυβικά του
- Οι ίπποι του
- Η ημερομηνία κατασκευής του

Για την επιτυχή εγγραφή του οχήματος ο χρήστης θα πρέπει να είναι ιδιαίτερα προσεκτικός σε ορισμένα σημεία κατά την εισαγωγή των δεδομένων. Συγκεκριμένα:

1. Δεν μπορούν να υπάρχουν κενά πεδία.



2. Το όνομα του κατασκευαστή περιέχει μόνο χαρακτήρες.
3. Το μοντέλο του οχήματος περιέχει μόνο χαρακτήρες και αριθμούς.
4. Ο αριθμός πινακίδας του οχήματος αποτελείται μόνο από τέσσερις λατινικούς χαρακτήρες που υπάρχουν και στο ελληνικό αλφάβητο, το σύμβολο – και από 3 ψηφία.
5. Δεν μπορούν να υπάρχουν στο σύστημα δύο ή παραπάνω οχήματα με τον ίδιο αριθμό πινακίδας.
6. Το πεδίο του χρώματος του οχήματος περιέχει μόνο χαρακτήρες.
7. Το πεδίο των κυβικών του αποτελείται μόνο από αριθμούς.
8. Το πεδίο των ίππων του αποτελείται μόνο από αριθμούς.
9. Στο πεδίο της ημερομηνίας ο αριθμός που αντιστοιχεί στην ημέρα δεν μπορεί να είναι μεγαλύτερος του 31 και ο αριθμός που αντιστοιχεί στον μήνα δεν μπορεί να είναι μεγαλύτερος του 12.



Εικόνα 48: Προσθήκη νέου οχήματος στο Web App

Η διαδικασία ολοκληρώνεται με το πάτημα του πράσινου κουμπιού “Add vehicle” όπου τα δεδομένα στέλνονται στην βάση δεδομένων για αποθήκευση. Φυσικά υπάρχει και η δυνατότητα ακύρωσης της επεξεργασίας με το πάτημα του κόκκινου κουμπιού “Cancel”.

### 6.4.3 Πληροφορίες διαδρομών

Στην αριστερή πλευρά της αρχικής σελίδας υπάρχει μια στήλη με τέσσερα κουμπιά το ένα κάτω από το άλλο.


- My missions
- Users
- Vehicles
- Missions


Οι επιλογές Users, Vehicles και Missions είναι προσβάσιμες μόνο από τους διαχειριστές του συστήματος καθώς οι πληροφορίες που μπορούν να αντλήσουν αφορούν όλους τους χρήστες της εφαρμογής.

Σε αυτή την υποενότητα θα δούμε την σελίδα που εμφανίζεται στον χρήστη όταν πατήσει την επιλογή “My missions”. Η σελίδα αυτή περιέχει πληροφορίες που αφορούν τα δρομολόγια που βρίσκονται σε εξέλιξη ή έχει ολοκληρώσει ο οδηγός. Πιο συγκεκριμένα, οι πληροφορίες αυτές είναι:

- Ο αριθμός πινακίδας του οχήματος που πραγματοποιήθηκε το δρομολόγιο
- Η ημερομηνία και ώρα αναχώρησης
- Η τοποθεσία αναχώρησης
- Η τοποθεσία προορισμού
- Η ημερομηνία και ώρα άφιξης στον προορισμό
- Η τρέχουσα κατάσταση ενός δρομολογίου (αν έχει ολοκληρωθεί ή όχι)

Η τελευταία στήλη με όνομα Actions αποτελείται από δύο κουμπιά για κάθε διαδρομή.

Το κουμπί  όπως θα δούμε και στην ενότητα 6.5, ανοίγει μια νέα καρτέλα στο πρόγραμμα περιήγησης παρουσιάζοντας στον χάρτη την διαδρομή με το σημείο αφετηρίας, όλες τις ενδιάμεσες στάσεις και το σημείο προορισμού.

Το κουμπί  διαγράφει το συγκεκριμένο δρομολόγιο και όλες τις πληροφορίες του από την βάση δεδομένων. Από την στιγμή που θα διαγραφεί το δρομολόγιο είναι αδύνατη η επαναφορά του.

Vehicle Plate	Start Time	Departure	Destination	End Time	Active	Actions
AAA-1234	6-10-2022 16:20:42	Πλ. Ουδέσσας 2 Αθήνα 103 52	Σολομών 29 Αθήνα 104 55	06-10-2022 16:20:59	false	
AAA-1234	6-10-2022 16:52:23	Πλ. Ουδέσσας 2 Αθήνα 103 52	Βίλλια 31 Αθήνα 117 42	06-10-2022 16:52:48	false	
AAA-1234	6-10-2022 16:53:30	Πλ. Ουδέσσας 2 Αθήνα 103 52	Εμπορική Μπενίκης 134 Αθήνα 114 73	06-10-2022 16:53:37	false	
AAA-9999	6-10-2022 17:0:58	Κοινοστάσιον 43 Κολοκόλλι 176 73	Αθηνάς 37 Κολοκόλλι 176 73	06-10-2022 17:01:10	false	
AAA-9999	6-10-2022 17:2:3	Κοινοστάσιον 43 Κολοκόλλι 176 73	Κοινοστάσιον 6 Νέα Σηλαση 173 29	06-10-2022 17:02:45	false	
AAA-9999	6-10-2022 17:4:3	Κοινοστάσιον 43 Κολοκόλλι 176 73	Εμπορική 31 Κολοκόλλι 176 74	06-10-2022 17:04:20	false	
AAA-8888	6-10-2022 17:7:10	Βετίσιον 4 Παρατός 185 42	Σολομών 27 Παρατός 185 45	06-10-2022 17:07:27	false	
AAA-8888	6-10-2022 17:10:0	Βετίσιον 4 Παρατός 185 42	Τη: Μαρτίου 1944 14 Νέσσον 184 50	06-10-2022 17:10:07	false	
AAA-1234	6-10-2022 18:26:40	Πλ. Ουδέσσας 2 Αθήνα 103 52	Αρακίτιον 74 Αθήνα Αρακίτιος 172 94	06-10-2022 18:26:52	false	
AAA-1234	6-10-2022 18:27:48	Πλ. Ουδέσσας 2 Αθήνα 103 52	Ηρακλείδων 5 Αθήνα 118 54	06-10-2022 18:27:55	false	
AAA-1234	6-10-2022 18:28:30	Πλ. Ουδέσσας 2 Αθήνα 103 52	Παρακοινοστάσιον 2 Παρατός 122 82	06-10-2022 18:28:02	false	
AAA-1234	6-10-2022 18:30:24	Πλ. Ουδέσσας 2 Αθήνα 103 52	Κοινοστάσιον 64 Αθήνα 118 55	06-10-2022 18:30:40	false	
AAA-1234	6-10-2022 18:31:19	Πλ. Ουδέσσας 2 Αθήνα 103 52	Αγιος Παρασκευάς 26 Παρατός 122 82	06-10-2022 18:31:31	false	
AAA-1234	6-10-2022 18:32:32	Πλ. Ουδέσσας 2 Αθήνα 103 52	Συρούσιον 3 Αθήνα 103 58	06-10-2022 18:32:59	false	
AAA-1234	6-10-2022 18:33:35	Πλ. Ουδέσσας 2 Αθήνα 103 52	Πλ. Αγίου Νικολάου 15 Αθήνα 104 46	06-10-2022 18:34:02	false	

Εικόνα 49: Σελίδα δρομολογίων ενός χρήστη στο Web app

Όπως φαίνεται και από την εικόνα 49, πάνω από τον πίνακα των δρομολογίων βρίσκεται μια μπάρα όπου στα δεξιά υπάρχει η πληροφορία του ακριβή αριθμού δρομολογίων της λίστας “Number of missions: 19”. Στην αριστερή πλευρά υπάρχει το πεδίο της αναζήτησης για την εύρεση ενός δρομολογίου που επιθυμεί ο χρήστης. Η αναζήτηση πραγματοποιείται σε ολόκληρο τον πίνακα συνεπώς η εύρεση του δρομολογίου πραγματοποιείται πολύ εύκολα απλά εισάγοντας μια οποιαδήποτε πληροφορία. Στην εικόνα 50 ο οδηγός θέλει να βρει μια διαδρομή που έχει σχέση με την οδό Ευριπίδου.

Vehicle Plate	Start Time	Departure	Destination	End Time	Active	Actions
.AAA-9999	6-10-2022 17:4:3	Κιλίκουρη 43 Κιλίκουρη 176 73	Κιλίκουρη 32 Κιλίκουρη 176 74	06-10-2022 17:04:20	false	

Εικόνα 50: Εύρεση δρομολογίου ενός χρήστη στο Web app

#### 6.4.4 Πληροφορίες συνόλου χρηστών

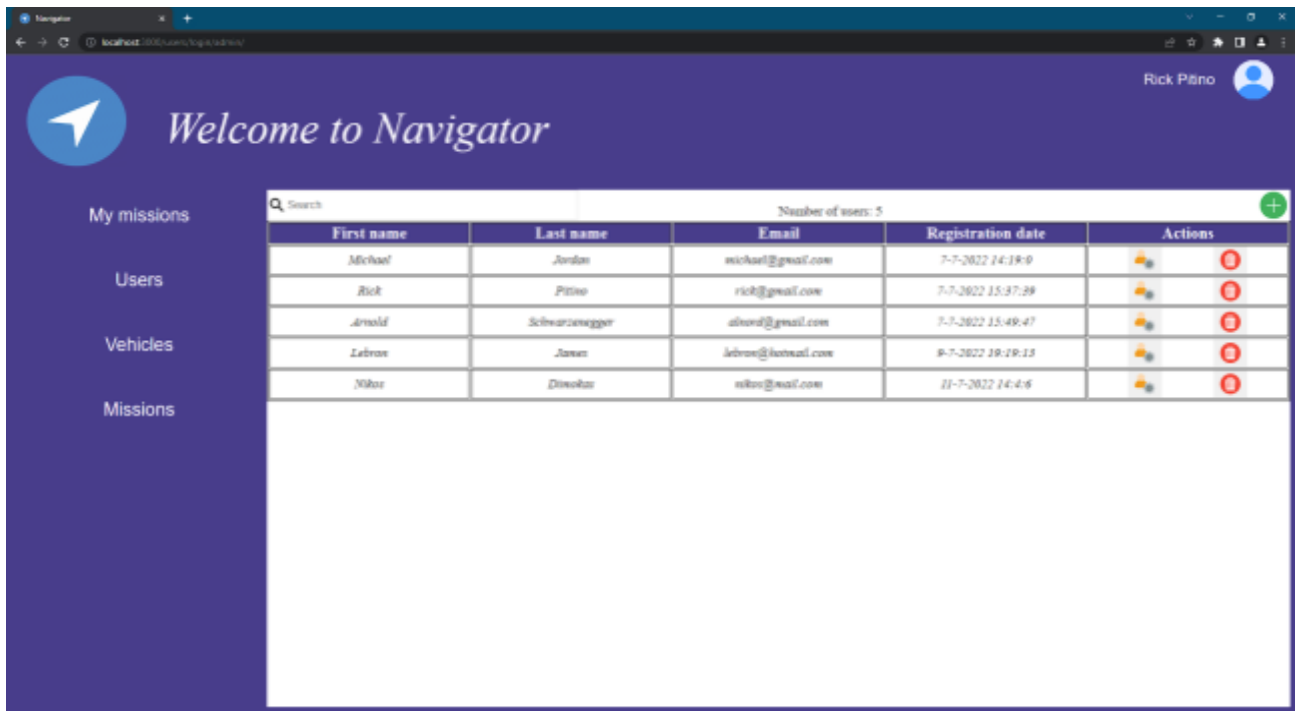
Η υποενότητα αυτή, όπως και οι επόμενες 2, αναφέρονται σε σελίδες που είναι προσβάσιμες μόνο από τους διαχειριστές του συστήματος. Με το πάτημα του κουμπιού “Users” εμφανίζεται μια σελίδα που περιέχει τον πίνακα με όλους τους εγγεγραμμένους χρήστες της εφαρμογής. Πιο αναλυτικά, οι πληροφορίες που περιλαμβάνονται για κάθε χρήστη είναι:

- Το όνομά του
- Το επώνυμό του
- Το email του
- Η ημερομηνία και ώρα εγγραφής του στο σύστημα.

Η τελευταία στήλη με όνομα Actions αποτελείται από δύο κουμπιά για κάθε χρήστη.

Το κουμπί εμφανίζει ένα πίνακα για την επεξεργασία των πληροφοριών του χρήστη.

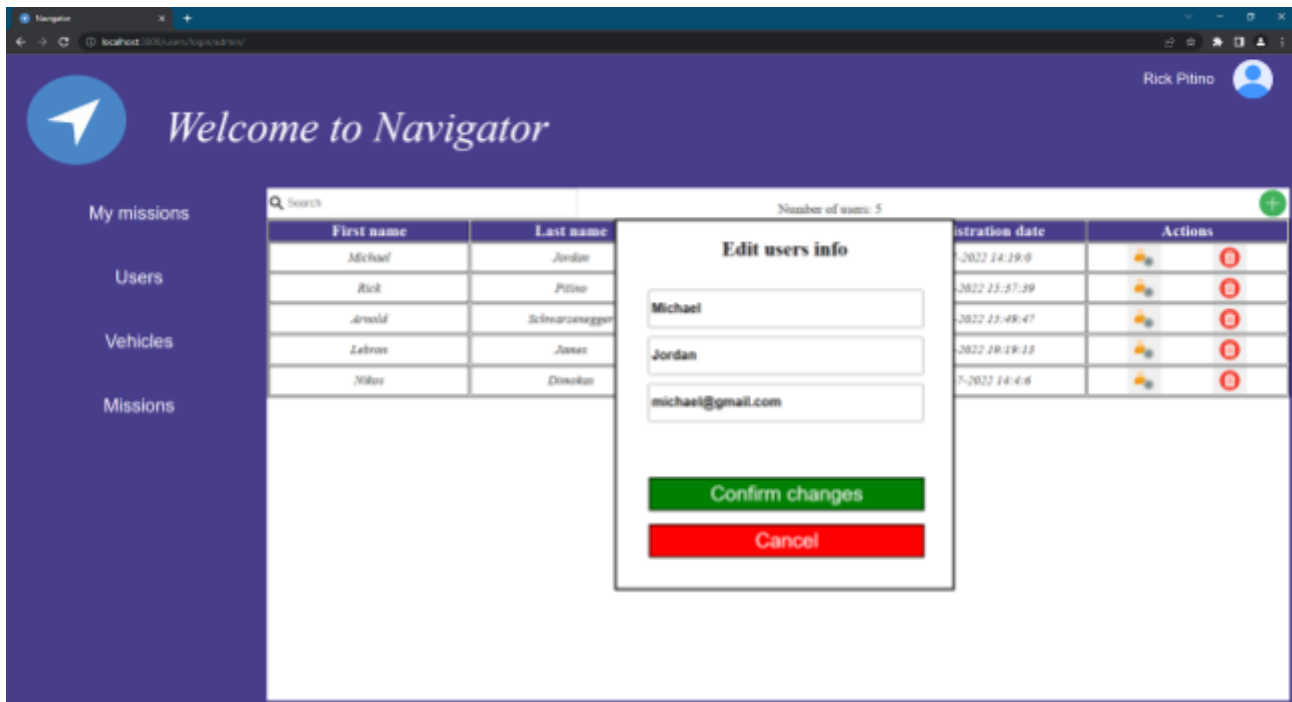
Το κουμπί διαγράφει τον συγκεκριμένο χρήστη και όλες τις πληροφορίες του από την βάση δεδομένων.



Εικόνα 51: Σελίδα χρηστών του συστήματος στο Web app


Οι διαχειριστές έχουν το δικαίωμα να τροποποιήσουν τις πληροφορίες από το προφίλ άλλων χρηστών.

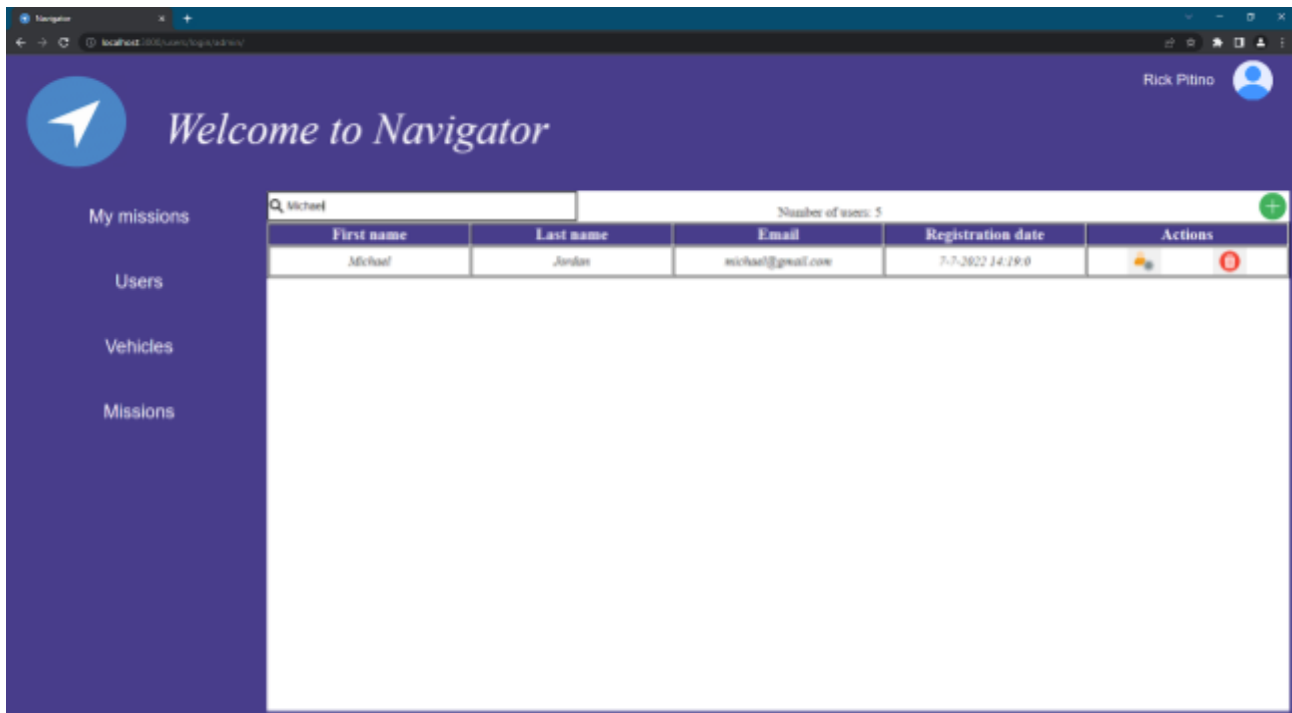
Ο πίνακας επεξεργασίας εμφανίζεται πατώντας κουμπί από την στήλη Actions. Κάθε κουμπί αντιστοιχεί και στον χρήστη της συγκεκριμένης σειράς. Στην εικόνα 52 ο διαχειριστής Rick Pitino επιχειρεί να αλλάξει τα στοιχεία ενός άλλου χρήστη, του Michael Jordan.



Εικόνα 52: Επεξεργασία στοιχείων χρήστη από την σελίδα του διαχειριστή στο Web app

Η διαδικασία ολοκληρώνεται με το πάτημα του πράσινου κουμπιού “Confirm changes” όπου τα δεδομένα στέλνονται στην βάση δεδομένων για αποθήκευση. Φυσικά υπάρχει και η δυνατότητα ακύρωσης της επεξεργασίας με το πάτημα του κόκκινου κουμπιού “Cancel”.

Όπως φαίνεται και από την εικόνα 52, πάνω από τον πίνακα χρηστών βρίσκεται μια μπάρα όπου στα δεξιά υπάρχει το κουμπί . Με το πάτημα αυτού του κουμπιού δίνεται η επιλογή προσθήκης ενός νέου χρήστη ανακατευθύνοντας τον τρέχον διαχειριστή στην σελίδα εγγραφής που είδαμε στην ενότητα 6.3. Στην μέση της μπάρας υπάρχει η εξής πληροφορία: “Number of users: 5” όπου δηλώνει τον αριθμό των εγγεγραμμένων χρηστών στην εφαρμογή. Τέλος στην αριστερή πλευρά της μπάρας υπάρχει το πεδίο της αναζήτησης για την εύρεση ενός οδηγού από το σύνολο. Στην εικόνα 53 πραγματοποιήθηκε αναζήτηση του Michael Jordan με βάση το όνομά του. Κάθε χρήστης μπορεί να εντοπιστεί εισάγοντας το όνομα ή επώνυμο ή email του.



Εικόνα 53: Αναζήτηση ενός χρήστη από την σελίδα διαχειριστή στο Web app

#### 6.4.5 Πληροφορίες συνόλου οχημάτων

Ένας διαχειριστής προκειμένου να αποκτήσει πρόσβαση στις πληροφορίες όλων των οχημάτων του συστήματος, αρκεί να πατήσει το κουμπί “Vehicles” όπως φαίνεται στην εικόνα 54. Πιο συγκεκριμένα, οι πληροφορίες για κάθε όχημα είναι:

- Ο κατασκευαστής του οχήματος
- Το μοντέλο του οχήματος
- Ο αριθμός πινακίδας του
- Η κατηγορία του οχήματος
- Το χρώμα του
- Τα κυβικά του
- Οι ίπποι του
- Η ημερομηνία κατασκευής του

Welcome to Navigator

Rick Páino

My missions

Users

Vehicles

Missions

Search

Number of vehicles: 8

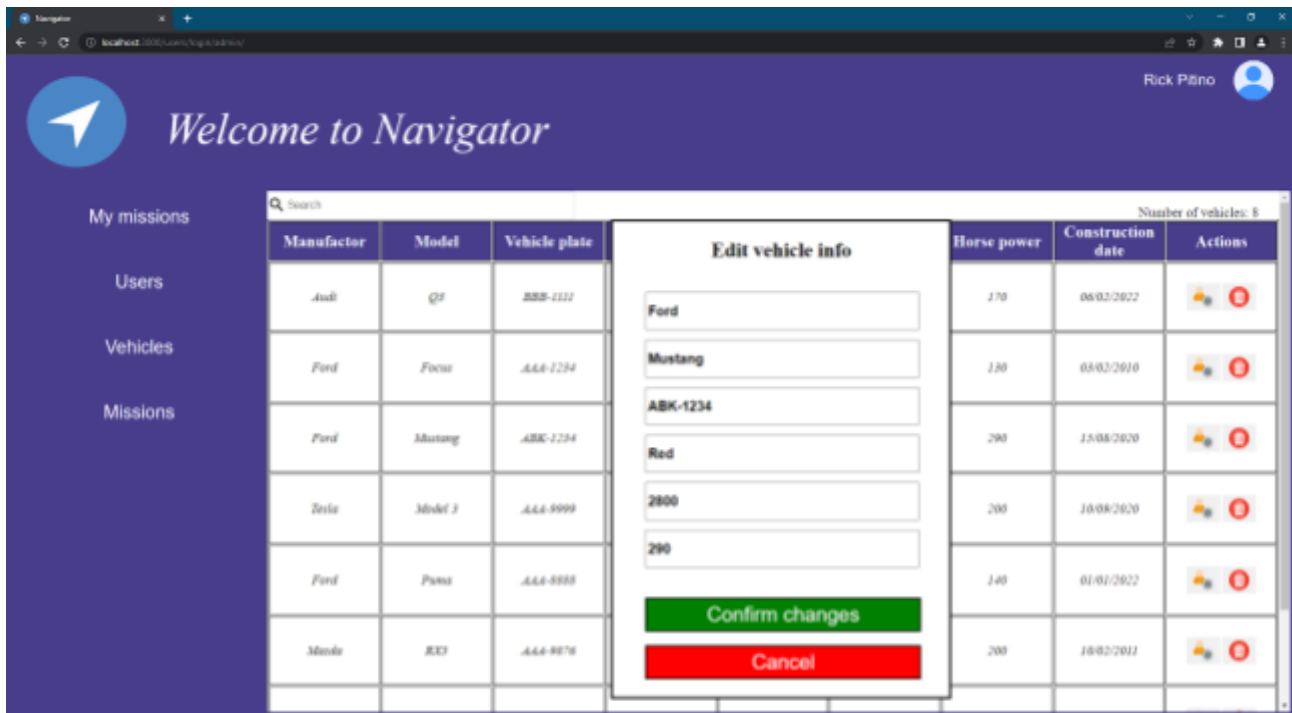
Manufacturer	Model	Vehicle plate	Type	Color	CC	Horse power	Construction date	Actions
Judi	Q5	BBB-1111	Car	Black	1800	170	06/02/2022	
Ford	Focus	AAA-1234	Car	Black	1000	130	03/02/2010	
Ford	Mustang	ABK-1234	Car	Red	2800	290	15/08/2020	
Tesla	Model 3	AAA-9999	Car	Green	2000	200	10/09/2020	
Ford	Puma	AAA-5555	Car	Red	1400	140	01/01/2022	
Mitsubishi	R2D	AAA-9876	Car	Yellow	2100	200	18/02/2011	

Εικόνα 54: Σελίδα συνόλου οχημάτων του συστήματος στο Web app

Οι διαχειριστές εκτός από το δικαίωμα αλλαγής των στοιχείων που αφορούν τους χρήστες μπορούν να παρέμβουν και τις πληροφορίες των οχημάτων. Στην εικόνα 54 μπορούμε να παρατηρήσουμε μια ακόμη στήλη στον πίνακα των οχημάτων με όνομα Actions. Η στήλη αυτή δίνει την επιλογή στον διαχειριστή να επεξεργαστεί τα στοιχεία ενός οχήματος ή να διαγράψει κάποιο από αυτά. Βασική προϋπόθεση για την διαγραφή ενός οχήματος είναι η ύπαρξη τουλάχιστον ενός ακόμη στην κατοχή του χρήστη. Δεν μπορεί να υπάρξει χρήστης χωρίς να κατέχει όχημα.

Πιο συγκεκριμένα, με το πάτημα από τον διαχειριστή του κουμπιού διαγράφεται από το σύστημα το όχημα της αντίστοιχης σειράς ενώ πατώντας το κουμπί εμφανίζεται ένας πίνακας για την επεξεργασία των πληροφοριών του οχήματος. Όπως είδαμε και στην επεξεργασία των δεδομένων ενός οδηγού, η εφαρμογή συλλέγει αυτόματα τα υπάρχοντα δεδομένα και τα συμπληρώνει στα αντίστοιχα πεδία. Στην εικόνα 55 ο χρήστης πάτησε να επεξεργαστεί τις πληροφορίες ενός Ford Mustang με αριθμό πινακίδας ABK-1234 χρώματος κόκκινου. Τα κυβικά του ανέρχονται στα 2800 με 290 ίππους. Η διαδικασία ολοκληρώνεται με το πάτημα του πράσινου κουμπιού “Confirm changes” όπου τα δεδομένα στέλνονται στην βάση δεδομένων για αποθήκευση. Φυσικά υπάρχει και η δυνατότητα ακύρωσης της επεξεργασίας με το πάτημα του κόκκινου κουμπιού “Cancel”.



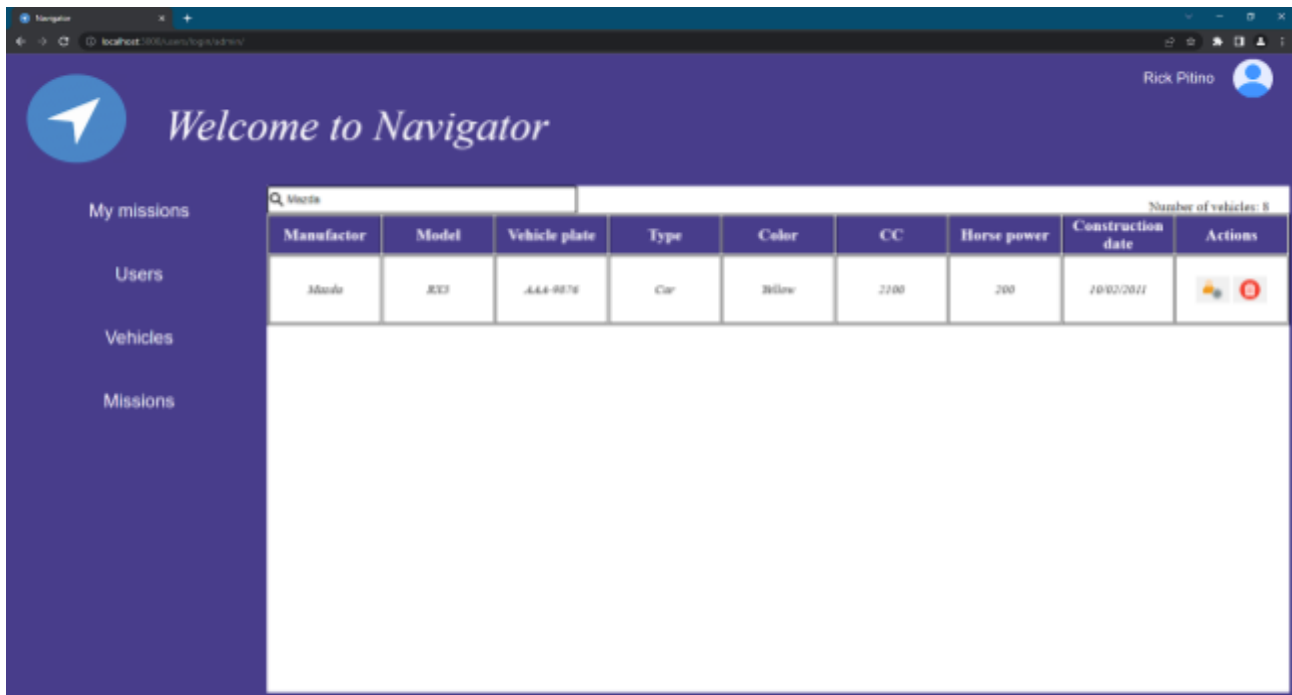


Εικόνα 55: Επεξεργασία στοιχείων οχήματος από την σελίδα διαχειριστή στο Web app

Πάνω από τον πίνακα των οχημάτων υπάρχει μια μπάρα με τα εξής στοιχεία:

- Αναζήτηση ενός οχήματος
- Μετρητής για τον αριθμό των οχημάτων

Όταν λέμε αναζήτηση ενός οχήματος αναφερόμαστε στην δυνατότητα που έχει ο χρήστης να βρει ένα όχημα μέσα στην λίστα του συνόλου των οχημάτων πληκτρολογώντας στο πεδίο αναζήτησης οποιαδήποτε πληροφορία σχετίζεται με αυτό. Στο παρακάτω στιγμιότυπο της εικόνας 56, ο Rick Pitino αναζητά από το σύνολο των οχημάτων ένα Mazda όχημα χωρίς να γνωρίζει κάποια άλλη πληροφορία. Το μόνο που έκανε είναι να πληκτρολογήσει την λέξη “Mazda” και η εφαρμογή πραγματοποίησε αναζήτηση στην λίστα για οχήματα Mazda. Η λειτουργία αναζήτησης δεν δουλεύει μόνο εισάγοντας το όνομα του κατασκευαστή ενός οχήματος. Η εφαρμογή πραγματοποιεί αναζήτηση σε ολόκληρο τον πίνακα. Συνεπώς σε περίπτωση που ένας χρήστης δεν γνωρίζει το όνομα του κατασκευαστή αλλά το μοντέλο του ή κάποια άλλη από τις πληροφορίες ενός οχήματος μπορεί απλά να το εισάγει στο πεδίο της αναζήτησης για την εύρεσή του.



Εικόνα 56: Αναζήτηση ενός οχήματος από την σελίδα διαχειριστή στο Web app


Στην δεξιά πλευρά της μπάρας υπάρχει η εξής πληροφορία: “Number of vehicles: 8”. Για την εύκολη διαχείριση του συνόλου των οχημάτων παρέχεται στον διαχειριστή ο ακριβής αριθμός όλων των οχημάτων που είναι εγγεγραμμένα στο σύστημα.


#### 6.4.6 Πληροφορίες συνόλου διαδρομών

Σε αυτή την υποενότητα θα δούμε την σελίδα που εμφανίζεται στον διαχειριστή όταν πατήσει το κουμπί “Missions”. Η σελίδα αυτή περιέχει όλα τα δρομολόγια όλων των οδηγών. Με άλλα λόγια, σε αυτή την σελίδα παρουσιάζονται οι πληροφορίες που αφορούν τα δρομολόγια που βρίσκονται σε εξέλιξη ή έχει ολοκληρώσει ο κάθε οδηγός. Πιο συγκεκριμένα, οι πληροφορίες αυτές είναι:




- Ο αριθμός πινακίδας του οχήματος που πραγματοποιεί το δρομολόγιο
- Η ημερομηνία και ώρα αναχώρησης
- Η τοποθεσία αναχώρησης
- Η τοποθεσία προορισμού
- Η ημερομηνία και ώρα άφιξης στον προορισμό
- Η τρέχουσα κατάσταση ενός δρομολογίου (αν έχει ολοκληρωθεί ή όχι)

Η τελευταία στήλη με όνομα Actions αποτελείται από δύο κουμπιά για κάθε διαδρομή.

Το κουμπί  όπως θα δούμε και στην ενότητα 6.5, ανοίγει μια νέα καρτέλα στο πρόγραμμα περιήγησης παρουσιάζοντας στον χάρτη την διαδρομή με το σημείο αφητηρίας, όλες τις ενδιάμεσες στάσεις και το σημείο προορισμού.

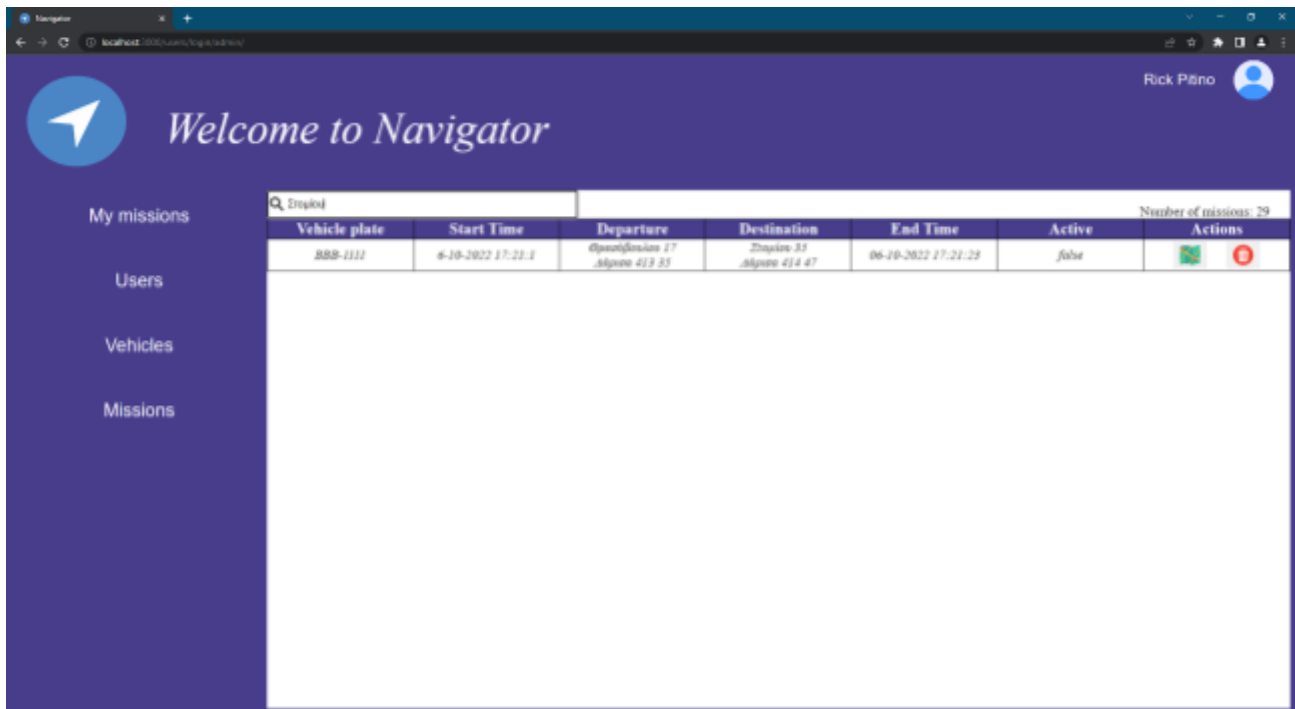
Το κουμπί  διαγράφει το συγκεκριμένο δρομολόγιο και όλες τις πληροφορίες του από την βάση δεδομένων. Από την στιγμή που θα διαγραφεί το δρομολόγιο είναι αδύνατη η επαναφορά του.



Vehicle plate	Start Time	Departure	Destination	End Time	Active	Actions
AAA-1234	6-10-2022 16:20:42	Πλ. Ομόνοιας 2 Αθήνα 183 32	Συγγρούς 23 Αθήνα 184 33	06-10-2022 16:20:59	false	 
AAA-1234	6-10-2022 16:32:21	Πλ. Ομόνοιας 2 Αθήνα 183 32	Βέροια 31 Αθήνα 217 42	06-10-2022 16:32:48	false	 
AAA-1234	6-10-2022 16:53:30	Πλ. Ομόνοιας 2 Αθήνα 183 32	Συγγρούς 23 Αθήνα 214 73	06-10-2022 16:53:37	false	 
AAA-9999	6-10-2022 17:0:18	Κολοκοιτώνη 43 Κηφισία 176 73	Αθήνας 57 Κηφισία 176 73	06-10-2022 17:01:10	false	 
AAA-9999	6-10-2022 17:2:3	Κολοκοιτώνη 43 Κηφισία 176 73	Κολοκοιτώνη 6 Νέα Σηράμη 173 29	06-10-2022 17:02:43	false	 
AAA-9999	6-10-2022 17:4:3	Κολοκοιτώνη 43 Κηφισία 176 73	Συγγρούς 31 Κηφισία 176 74	06-10-2022 17:04:20	false	 
AAA-5555	6-10-2022 17:7:10	Βασιλέως 4 Πειραιάς 183 42	Συγγρούς 27 Πειραιάς 183 43	06-10-2022 17:07:27	false	 
AAA-5555	6-10-2022 17:10:0	Βασιλέως 4 Πειραιάς 183 42	7ης Μαγιάλας 1944 14 Νέοι 184 30	06-10-2022 17:10:07	false	 
ABK-1234	6-10-2022 17:13:29	Βιαρθέου 73 Βίλιας 282 22	Γκαλιάνου 102 Βίλιας 282 22	06-10-2022 17:13:01	false	 
ABK-1234	6-10-2022 17:15:31	Βιαρθέου 73 Βίλιας 282 22	Κωνσταντίνου 88 Βίλιας 283 33	06-10-2022 17:15:58	false	 
ABK-1234	6-10-2022 17:16:46	Βιαρθέου 73 Βίλιας 282 22	Αρκαδίας 72 Βίλιας 285 00	06-10-2022 17:16:53	false	 
BBB-1111	6-10-2022 17:18:33	Θρακοπούλου 17 Αίγινα 413 33	Γαλακτοπία 7 Αίγινα 412 22	06-10-2022 17:18:43	false	 
BBB-1111	6-10-2022 17:19:34	Θρακοπούλου 17 Αίγινα 413 33	Μη. Πολιτιστικού 203 Αίγινα 412 22	06-10-2022 17:19:51	false	 
BBB-1111	6-10-2022 17:21:1	Θρακοπούλου 17 Αίγινα 413 33	Σπυριών 23 Αίγινα 414 47	06-10-2022 17:21:23	false	 
AAA-9976	6-10-2022 17:23:19	Συγγρούς 14 Κηφισία 173 41	Συγγρούς 11 Κηφισία 173 41	06-10-2022 17:23:51	false	 

Εικόνα 57: Σελίδα συνόλου δρομολογίων του συστήματος στο Web app

Όπως φαίνεται και από την εικόνα 57, πάνω από τον πίνακα των δρομολογίων βρίσκεται μια μπάρα όπου στα δεξιά υπάρχει η πληροφορία του ακριβή αριθμού δρομολογίων της λίστας “Number of missions: 29”. Στην αριστερή πλευρά υπάρχει το πεδίο της αναζήτησης για την εύρεση ενός δρομολογίου που επιθυμεί ο διαχειριστής. Η αναζήτηση πραγματοποιείται σε ολόκληρο τον πίνακα συνεπώς η εύρεση του δρομολογίου πραγματοποιείται πολύ εύκολα απλά εισάγοντας μια οποιαδήποτε πληροφορία. Στην εικόνα 58 ο διαχειριστής θέλει να βρει μια διαδρομή που έχει σχέση με την οδό Στομίου.

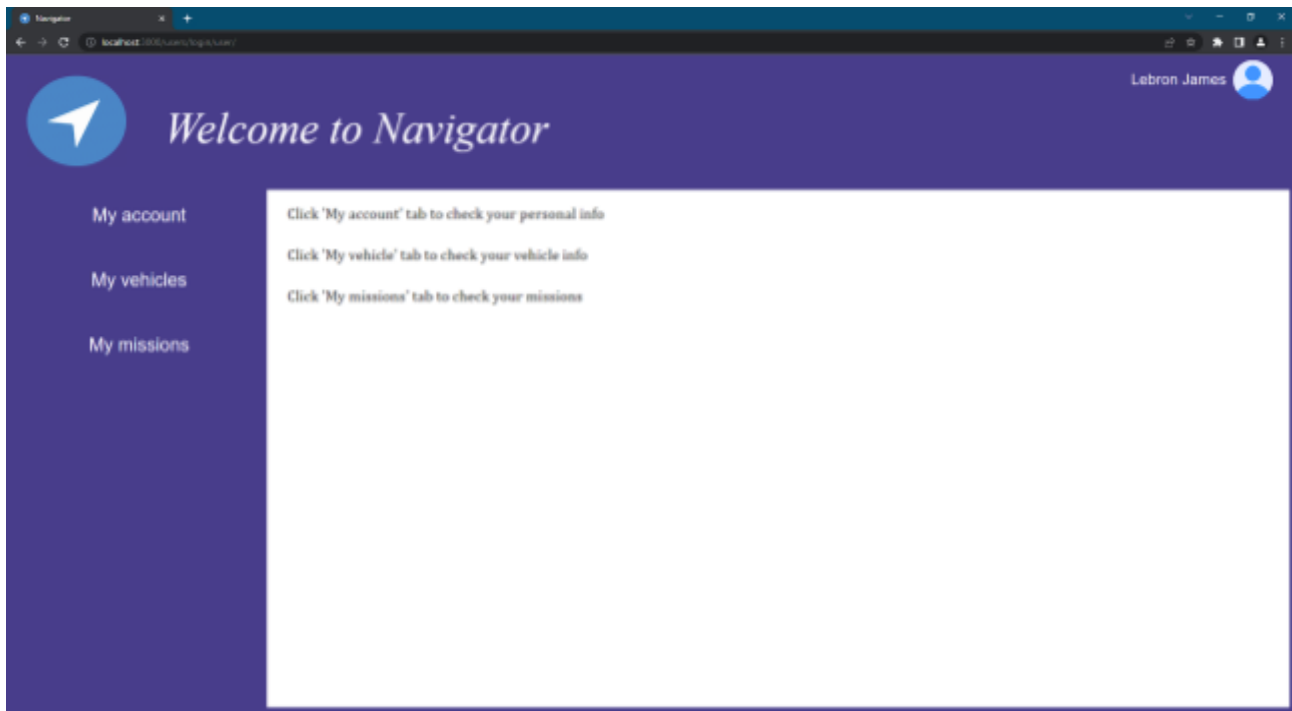


Εικόνα 58: Αναζήτηση ενός δρομολογίου από την σελίδα διαχειριστή στο Web App

## 6.4 Σελίδα χρήστη

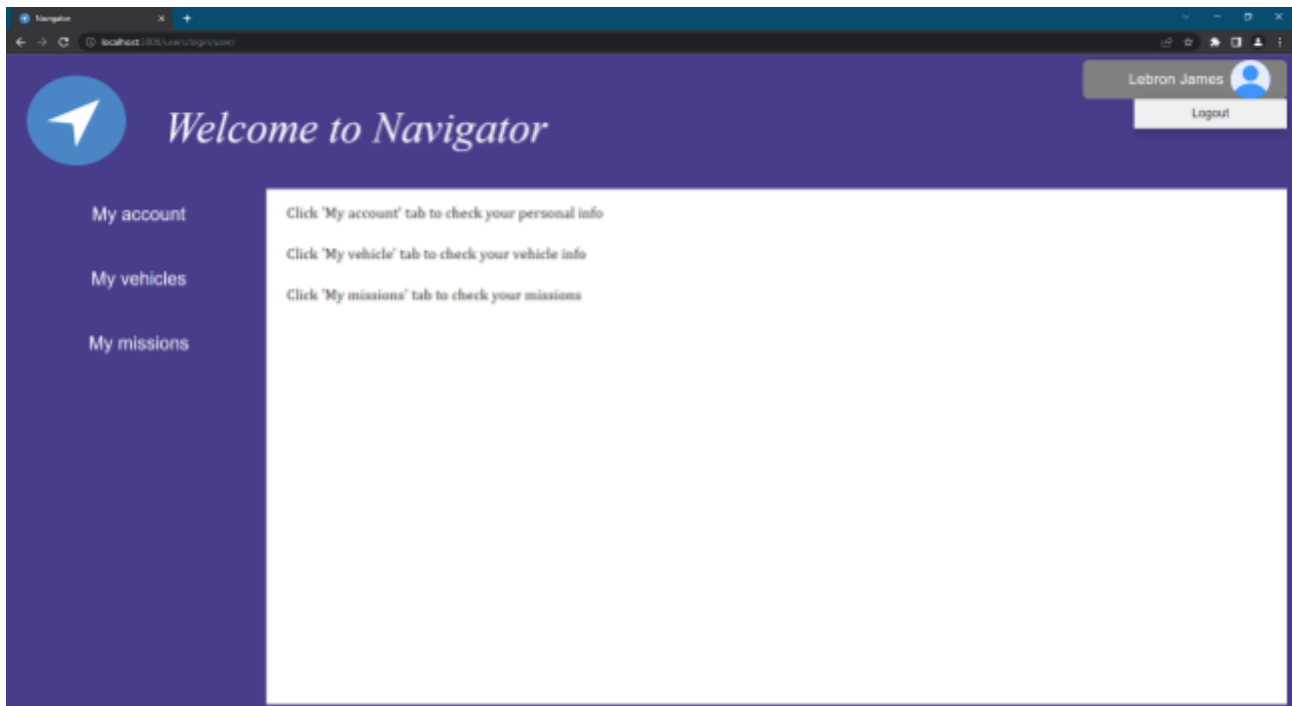
Η προηγούμενη ενότητα αφορούσε την παρουσίαση της εφαρμογής από την πλευρά ενός διαχειριστή του συστήματος. Σε αυτή την ενότητα θα γνωρίσουμε την εφαρμογή από την πλευρά του απλού χρήστη. Εισάγοντας ο χρήστης το email και τον κωδικό πρόσβασής του, εισέρχεται στην εφαρμογή όπου έχει το δικαίωμα πρόσβασης:

- Στο προφίλ του
- Στο προφίλ των οχημάτων του
- Στην λίστα των δρομολογίων του




*Εικόνα 59: Αρχική σελίδα χρήστη στο Web app*

Οι σελίδες του προφίλ, των οχημάτων και των δρομολογίων είναι ακριβώς οι ίδιες που εξετάσαμε στις υποενότητες 6.4.1 , 6.4.2 και 6.4.3 αντίστοιχα. Αξίζει να σημειώσουμε ότι στην σελίδα χρήστη, όπως και στην αντίστοιχη του διαχειριστή, υπάρχει ένα κουμπί στο πάνω δεξιό μέρος της οθόνης που περιέχει το ονοματεπώνυμο του. Πατώντας το κουμπί αυτό εμφανίζεται μια επιλογή: **Logout** για αποσύνδεση από τον λογαριασμό και επιστροφή στην σελίδα εισόδου. Στην εικόνα 60 φαίνεται η επιλογή της αποσύνδεσης ύστερα από το πάτημα του κουμπιού.



Εικόνα 60: Dropdown menu της αρχικής οθόνης χρήστη στο Web app

## 6.5 Χάρτης διαδρομών

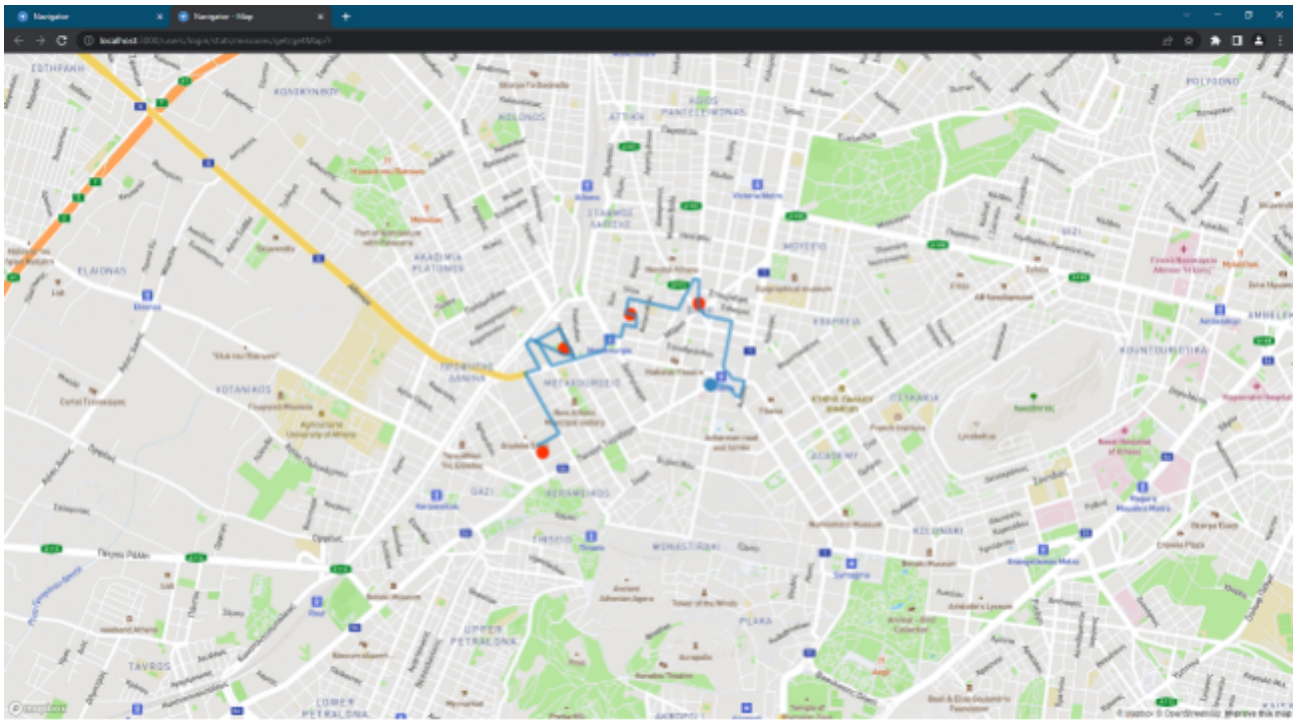
Όπως παρατηρήσαμε στα προηγούμενα κεφάλαια, στις σελίδες των δρομολογίων υπήρχε ένα κουμπί . Το κουμπί αυτό ανοίγει μια καινούργια καρτέλα στο πρόγραμμα περιήγησης παρουσιάζοντας στον χάρτη την διαδρομή με το σημείο αφετηρίας, όλες τις ενδιάμεσες στάσεις και το σημείο προορισμού. Ο μπλε κύκλος αποτελεί το σημείο έναρξης της διαδρομής ενώ οι υπόλοιποι κόκκινοι κύκλοι είναι οι στάσεις. Στην εικόνα 61 βλέπουμε ένα πλήρες δρομολόγιο.

Αφετηρία

**Πλατεία Ομόνοιας 2, Αθήνα**

Στάσεις

1. **Αχαρνών 2, Αθήνα**
2. **Βίκτωρος Ουγκώ 32 & Φαβιέρου 33, Αθήνα**
3. **Βιργινίας Μπενάκη 8-10, Αθήνα**
4. **Σαλαμίνος 23, Αθήνα**



*Εικόνα 61: Χάρτης διαδρομής στο Web app*





# Κεφάλαιο 7: Επίλογος

---

Σε αυτό το κεφάλαιο παρουσιάζονται τα συμπεράσματα που αντλήθηκαν κατά της διάρκεια εκπόνησης της διπλωματικής εργασίας. Ακόμη καταγράφονται οι μελλοντικοί στόχοι για την βελτίωση της εφαρμογής με σκοπό την χρήση της από το ευρύ κοινό.

## 7.1 Συμπεράσματα

Τα smartphones αποτελούν ίσως την πιο διαδεδομένη συσκευή του 21ου αιώνα. Μπορούν να εξυπηρετήσουν τους χρήστες σε αμέτρητες ανάγκες τους, να αποτελέσουν αντικείμενο ψυχαγωγίας ή να επικοινωνήσουν με άλλους. Προφανώς και η χρήση τους δεν περιορίζεται μόνο σε αυτούς τους τομείς. Το γεγονός ότι πρόκειται για φορητές συσκευές, τις καθιστά πολύ χρήσιμα εργαλεία ακόμη και για δουλειά όπως φάνηκε από το αντικείμενο της συγκεκριμένης εργασίας. Με το smartphone μπορούμε οποιαδήποτε στιγμή να βρούμε την τοποθεσία μας στον χάρτη και να λάβουμε οδηγίες προκειμένου να φτάσουμε στον προορισμό μας. Σε αυτό τον τομέα έχουν συμβάλει τα μέγιστα και τα δίκτυα κινητής τηλεφωνίας. Οι περισσότεροι πάροχοι κινητής τηλεφωνίας προσφέρουν ικανοποιητική γεωγραφική κάλυψη με αποτέλεσμα τα smartphones να μπορούν να λειτουργούν σχεδόν παντού και κάτω από οποιεσδήποτε συνθήκες. Όταν λοιπόν οι υπηρεσίες εντοπισμού συνεργάζονται με το δίκτυο κινητής τηλεφωνίας το αποτέλεσμα είναι ο εντοπισμός της συσκευής με πολύ μεγάλο ποσοστό επιτυχίας.

Στην παρούσα εφαρμογή που υλοποιήθηκε στα πλαίσια της διπλωματικής εργασίας παρουσιάστηκε πως λειτουργεί μια σύγχρονη εφαρμογή πλοήγησης. Μια εφαρμογή που απαιτεί γνώσεις σε αρκετούς τομείς για την επίτευξη της. Βασικός στόχος της είναι η παροχή οδηγιών κατεύθυνσης σε ανθρώπους που πραγματοποιούν πολλά δρομολόγια κατά την διάρκεια της ημέρας με αρκετές στάσεις. Δόθηκε ιδιαίτερη προσοχή στην εμφάνιση της εφαρμογής με σκοπό την ευχάριστη χρήση της. Ακόμη, έγινε αντιληπτό πως δουλεύει ένας server, πως μπορεί να επικοινωνήσει μια εφαρμογή με αυτόν, τι μορφή έχουν τα δεδομένα που ανταλλάσσουν και με ποιο τρόπο αποθηκεύει πληροφορίες σε μια βάση δεδομένων. Η εξέλιξη των τεχνολογιών στον χώρο του διαδικτύου απέδειξε ότι μπορεί να διευκολύνει σε μεγάλο βαθμό μια εφαρμογή να επικοινωνήσει με μια άλλη και να ανταλλάξουν δεδομένα.

## 7.2 Μελλοντικές Επεκτάσεις

Η διπλωματική εργασία έφτασε στο τέλος της. Δεν συμβαίνει το ίδιο και με την εφαρμογή. Όπως κάθε εφαρμογή που κυκλοφορεί αυτή την στιγμή στο διαδίκτυο έτσι και η συγκεκριμένη δέχεται ορισμένες βελτιώσεις. Θα ήταν χρήσιμο λοιπόν να αναφερθούν οι αλλαγές που θα βελτιώναν την λειτουργικότητα της εφαρμογής και θα την καθιστούσαν πιο ανταγωνιστική έναντι άλλων αντίστοιχων εφαρμογών.

- Ο server μέσω του οποίου δούλεψε η web και android εφαρμογή για την επικοινωνία με την βάση δεδομένων ήταν τοπικός. Αυτό σημαίνει ότι το σύστημα δεν μπορεί να δουλέψει εκτός του τοπικού δικτύου. Η χρήση ενός remote server θα πολλαπλασίαζε την χρησιμότητα της εφαρμογής για την αξιοποίησή της από χρήστες εκτός τοπικού δικτύου.
- Η android εφαρμογή δεν περιλαμβάνει live tracking. Η παρακολούθηση σε πραγματικό χρόνο της τοποθεσίας ενός οχήματος αποτελεί πολύ μεγάλο πλεονέκτημα ιδιαίτερα όταν πρόκειται για εταιρεία. Με αυτό τον τρόπο ο κάθε πελάτης θα ήταν σε θέση να γνωρίζει που βρίσκεται το πακέτο του και πόσο χρόνο θα χρειαστεί να περιμένει για να το παραλάβει.
- Τόσο στην android όσο και στην web εφαρμογή, κατά την δημιουργία ενός λογαριασμού ο χρήστης δημιουργούσε έναν τετραψήφιο αριθμό επαναφοράς (PIN) για την περίπτωση που ξεχνούσε τον κωδικό πρόσβασης. Η δημιουργία ενός μηχανισμού που θα επικοινωνεί με τον χρήστη μέσω email για την επαναφορά του κωδικού του όπως γίνεται πλέον σε όλες τις εφαρμογές αποτελεί στόχο για την εφαρμογή.

# Βιβλιογραφία

- [1] “Web server” [What is a Web Server and How Does it Work? \(techtarget.com\)](#) (Πρόσβαση 30 Αυγούστου 2022)
- [2] “What Is a Web Server? How It Works and More” [What Is a Web Server? Everything You Need to Know \(hostinger.in\)](#) (Πρόσβαση 30 Αυγούστου 2022)
- [3] “What Is a Database?” [What Is a Database | Oracle](#) (Πρόσβαση 1 Σεπτεμβρίου 2022)
- [4] “What Is a Database?” [What is a Database?](#) (Πρόσβαση 1 Σεπτεμβρίου 2022)
- [5] “What is a Database? Definition, Meaning, Types with Example” [What is a Database? Definition, Meaning, Types with Example \(guru99.com\)](#) (Πρόσβαση 1 Σεπτεμβρίου 2022)
- [6] “Λειτουργικά Συστήματα” [3.1 Λειτουργικά Συστήματα \(ebooks.edu.gr\)](#) (Πρόσβαση 3 Σεπτεμβρίου 2022)
- [7] “Android” [Android Definition \(techterms.com\)](#) (Πρόσβαση 3 Σεπτεμβρίου 2022)
- [8] “Android (operating system)” [Android \(operating system\) – Wikipedia](#) (Πρόσβαση 4 Σεπτεμβρίου 2022)
- [9] “What Is an SDK?” [What Is an SDK? \(auth0.com\)](#) (Πρόσβαση 4 Σεπτεμβρίου 2022)
- [10] “Android Everywhere: 10 Types of Devices That Android Is Making Better” [Android Everywhere: 10 Types of Devices That Android Is Making Better \(androidauthority.com\)](#) (Πρόσβαση 4 Σεπτεμβρίου 2022)
- [11] “List of features in Android” [List of features in Android – Wikipedia](#) (Πρόσβαση 5 Σεπτεμβρίου 2022)
- [12] “Features of Android Operating System OS” [Features of Android Operating System OS 7.4.3: - W7cloud](#) (Πρόσβαση 5 Σεπτεμβρίου 2022)
- [13] “What Is Android? 7 Unique Features Of Android Operating System.” [What Is Android? 7 Unique Features Of Android Operating System. - FITA Academy](#) (Πρόσβαση 5 Σεπτεμβρίου 2022)
- [14] “Set it and forget it: 5 things you should always automate on your phone” [Android Automation - 5 simple automation tasks that will make life easier \(androidauthority.com\)](#) (Πρόσβαση 6 Σεπτεμβρίου 2022)
- [15] “Global Positioning System” [Global Positioning System - Wikipedia](#) (Πρόσβαση 6 Σεπτεμβρίου 2022)
- [16] “Korean Air Lines Flight 007” [Korean Air Lines Flight 007 - Wikipedia](#) (Πρόσβαση 12 Σεπτεμβρίου 2022)
- [17] “GLONASS” [GLONASS - Wikipedia](#) (Πρόσβαση 12 Σεπτεμβρίου 2022)
- [18] “Quasi-Zenith Satellite System” [Quasi-Zenith Satellite System - Wikipedia](#) (Πρόσβαση 13 Σεπτεμβρίου 2022)
- [19] “Galileo” [Galileo \(satellite navigation\) - Wikipedia](#) (Πρόσβαση 13 Σεπτεμβρίου 2022)

- [20] “Indian Regional Navigation Satellite System” [Indian Regional Navigation Satellite System - Wikipedia](#) (Πρόσβαση 14 Σεπτεμβρίου 2022)
- [21] “How GPS works?” [How GPS works? - GeeksforGeeks](#) (Πρόσβαση 16 Σεπτεμβρίου 2022)
- [22] “How does GPS work on my phone?” [How does GPS work on my phone? | Android Central](#) (Πρόσβαση 16 Σεπτεμβρίου 2022)
- [23] “Understanding AGPS (Assisted GPS)” [Understanding AGPS \(Assisted GPS\) | by abbas aniefa | IVYMobility TechBytes | Medium](#) (Πρόσβαση 17 Σεπτεμβρίου 2022)
- [24] “How GPS Phones Work” [How GPS Phones Work | HowStuffWorks](#) (Πρόσβαση 22 Σεπτεμβρίου 2022)
- [25] “How GPS Actually Works” [How GPS Actually Works \(howtogeek.com\)](#) (Πρόσβαση 22 Σεπτεμβρίου 2022)
- [26] “Google Maps” [Google Maps - Wikipedia](#) (Πρόσβαση 22 Σεπτεμβρίου 2022)
- [27] “The 11 Best Free Route Planner With Unlimited Stops (Unique Features and Comparison)” [The 11 Best Free Route Planners with Unlimited Stops \[Comparison\] - Upper Route Planner \(upperinc.com\)](#) (Πρόσβαση 26 Σεπτεμβρίου 2022)
- [28] “The 11 Best Free Route Planners with Unlimited Stops” [The 11 Best Free Route Planners with Unlimited Stops 2022 | Maptive](#) (Πρόσβαση 26 Σεπτεμβρίου 2022)
- [29] “MapQuest” [MapQuest | American company | Britannica](#) (Πρόσβαση 26 Σεπτεμβρίου 2022)
- [30] “MapQuest” [MapQuest - Wikipedia](#) (Πρόσβαση 26 Σεπτεμβρίου 2022)
- [31] “Circuit Route Planner Reviews – Does It Make Your Delivery Process Efficient?” [Circuit Route Planner Review \[In-depth & Updated Jan '22\] - Upper Route Planner \(upperinc.com\)](#) (Πρόσβαση 27 Σεπτεμβρίου 2022)
- [32] “What is Android Studio? Things You Should Know” [What is Android Studio? Things You Should Know - JoyofAndroid.com](#) (Πρόσβαση 1 Οκτωβρίου 2022)
- [33] “Τι Είναι το API και Πώς Λειτουργεί; (Οδηγός 2022)” [Τι Είναι το API και Πώς Λειτουργεί; \(Οδηγός 2022\) \(bigblue.academy\)](#) (Πρόσβαση 18 Οκτωβρίου 2022)
- [34] “What is RESTful API?” [What is RESTful API? - RESTful API Beginner's Guide - AWS \(amazon.com\)](#) (Πρόσβαση 18 Οκτωβρίου 2022)
- [35] “RESTful Web Services” [RESTful Web Services - GeeksforGeeks](#) (Πρόσβαση 18 Οκτωβρίου 2022)
- [36] “What is REST?” [What is REST? | Codecademy](#) (Πρόσβαση 18 Οκτωβρίου 2022)
- [37] “HTTP request methods” [HTTP request methods - HTTP | MDN \(mozilla.org\)](#) (Πρόσβαση 18 Οκτωβρίου 2022)
- [38] “What is REST? REST Architecture and REST Constraints” [REST Architectural Constraints \(restfulapi.net\)](#) (Πρόσβαση 18 Οκτωβρίου 2022)

- [39] “A Beginner's Guide to JSON with Examples” A Beginner's Guide to JSON with Examples (attacomsian.com) (Πρόσβαση 19 Οκτωβρίου 2022)
- [40] “jQuery” jQuery - Wikipedia (Πρόσβαση 19 Οκτωβρίου 2022)
- [41] “Ajax (programming)” Ajax (programming) - Wikipedia (Πρόσβαση 19 Οκτωβρίου 2022)
- [42] “Retrofit Tutorial in Android – Part 1: Introduction” Retrofit Tutorial in Android – Part 1: Introduction - CodeProject (Πρόσβαση 18 Οκτωβρίου 2022)
- [43] “Node.js – Introduction” Node.js - Introduction (tutorialspoint.com) (Πρόσβαση 18 Οκτωβρίου 2022)