



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

Εφαρμογή συνεπιβίβασης

Κωνσταντίνος Πολύζος

Επιβλέπων: Γεώργιος Θάνος

Σεπτέμβριος 2022



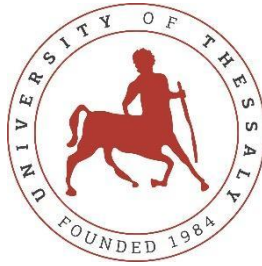
ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

Εφαρμογή συνεπιβίβασης

Κωνσταντίνος Πολύζος

Επιβλέπων: Γεώργιος Θάνος

Σεπτέμβριος 2022



UNIVERSITY OF THESSALY
SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Car pooling application

Konstantinos Polizos

Supervisor: Georgios Thanos

September 2022

Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπων

Θάνος Γεώργιος

Μέλος ΕΔΙΠ, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών
Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος

Τσαλαπάτα Χαρίκλεια

Μέλος ΕΔΙΠ, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών
Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος

Τουσίδου Ελένη

Μέλος ΕΔΙΠ, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών
Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

**ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΑΚΑΔΗΜΑΪΚΗΣ ΔΕΟΝΤΟΛΟΓΙΑΣ ΚΑΙ
ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ**

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα διπλωματική εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελούν αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλουν οποιασδήποτε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχουν έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης. 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. Δηλώνω επίσης ότι τα αποτελέσματα της εργασίας δεν έχουν χρησιμοποιηθεί για την απόκτηση άλλου πτυχίου. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο Δηλών

Κωνσταντίνος Πολύζος

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I also declare that the results of the work have not been used to obtain another degree. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism.

The Declarant

Konstantinos Polizos

Ευχαριστίες ή Σχόλια

Θα ήθελα να εκφράσω τις ευχαριστίες μου προς τον κ. Γεώργιο Θάνο, μέλος ΕΔΙΠ του Πανεπιστημίου Θεσσαλίας, για το ενδιαφέρον και τη βοήθειά του στην ολοκλήρωση της διπλωματικής εργασίας μου.

Εφαρμογή συνεπιβίβασης

Κωνσταντίνος Πολύζος

Περίληψη

Η παρούσα διπλωματική αφορά μία εφαρμογή μεταφοράς χρηστών με αυτοκίνητο. Τέτοιου τύπου εφαρμογές περιμένουν από τον χρήστη μία ένδειξη για τη γεωγραφική θέση του αλλά και τον προορισμό του εντός ή εκτός της πόλης που βρίσκεται. Στη συγκεκριμένη εφαρμογή η λειτουργικότητα διεκπεραιώνεται εντός της πόλης που βρίσκεται ο χρήστης, μέσω σύντομων διαδρομών.

Για την δημιουργία του app χρησιμοποιούμε ένα γνωστό εργαλείο του facebook την React Native, το οποίο είναι ένα framework σε Javascript. Επιλέχθηκε λόγω της ιδιότητας του να τρέχει η εφαρμογή σε οποιοδήποτε λειτουργικό κινητής συσκευής (ios ή android) χωρίς να χρειάζεται να δημιουργήσουμε ξεχωριστές εφαρμογές για κάθε πλατφόρμα. Θα εξηγήσουμε πως καταφέρνουμε να συνδυάσουμε τη γλώσσα javascript και την πλατφόρμα firebase για τη δημιουργία της εφαρμογής.

Η εφαρμογή έχει την δυνατότητα βαθμολόγησης, δημιουργία προσωπικού προφίλ, έλεγχο ειδοποιήσεων, επεξεργασία προφίλ αλλά και την προβολή ενός χάρτη για την πόλη στην οποία βρίσκεται ο χρήστης. Ο χρήστης μπορεί να μετακινηθεί ανά πάσα στιγμή αν και εφόσον υπάρχει διαθέσιμος οδηγός που μπορεί να κάνει κάποια παράκαμψη προς την κατεύθυνση του.

Diploma Thesis

Car pooling application

Konstantinos Polizos

Abstract

This diploma thesis focuses on a car pooling application. These types of applications expect from the end-users an indication of their location on the map and the destination they want to travel to. In this particular application, the functionality is handled within the city where the user is located, through short routes.

To implement the app we use the well-known facebook tool React Native, which is a framework of the Javascript language and was chosen due to its ability to run our application on any mobile operating system (i.e. ios, android) without having to create separate applications for each platform. We will explain how we manage to combine the javascript language and the firebase platform to create the application.

The application provides the ability for users to evaluate, create a personal profile, check notifications, edit their profiles and view a map of the city where the user chooses to be. The user can move at any time if and as long as there is a driver available who can make a detour in their direction.

Πίνακας περιεχομένων

<i>Ευχαριστίες ή Σχόλια</i>	<i>xiii</i>
<i>Περίληψη</i>	<i>xv</i>
<i>Abstract</i>	<i>xvii</i>
<i>Συνοτομογραφίες</i>	<i>xxix</i>
Κεφάλαιο 1 Εισαγωγή	1
1.1 Αντικείμενο της διπλωματικής	1
1.2 Συνεισφορά	2
1.3 Παρόμοιες εφαρμογές	3
1.4 Οργάνωση του τόμου	4
Κεφάλαιο 2 Εργαλεία Εφαρμογής	5
2.1 Εισαγωγή	5
2.2 Javascript	5
2.3 Node.js	6
2.4 NPM	6
2.5 React Native	7
2.6 Firebase	9
Κεφάλαιο 3 Παρουσίαση λειτουργικότητας	10
Κεφάλαιο 4 Αρχιτεκτονική εφαρμογής	19
4.1 Αρχιτεκτονική front - end	19
4.2 Tech Stack και η αρχιτεκτονική front - end και back - end μαζί	22
4.3 Εσωτερική δομή Firebase	23
Κεφάλαιο 5 Πορεία υλοποίησης	28
Κεφάλαιο 6 Πειραματική επικύρωση λειτουργικότητας	30

6.1 Τρόπος διεξαγωγής πειραμάτων	30
6.2 Αποτελέσματα πειραμάτων	34
Κεφάλαιο 7 Συμπεράσματα	36
7.1 Σύνοψη και συμπεράσματα	36
7.2 Μελλοντικές επεκτάσεις	37
Βιβλιογραφία	39

Συντομογραφίες

κ.α. και άλλα

π.χ. παραδείγματος χάριν

κλπ και λοιπά

κ.ο.κ. και ούτω καθεξής

Κεφάλαιο 1 Εισαγωγή

Το πρόβλημα που καλούμαστε να λύσουμε είναι κυρίως πρόβλημα μεταφοράς. Βρισκόμαστε, λοιπόν, σε μια συγκεκριμένη περιοχή στην οποία ο χρήστης πρωτοεμφανίζεται. Πρέπει να βρούμε τον τρόπο με τον οποίο ο χρήστης θα κατευθυνθεί προς μια συγκεκριμένη τοποθεσία, στην περιοχή του, αλλά και το πως θα γίνει αυτή η μεταφορά. Ακόμη παρέχεται η δυνατότητα οι χρήστες να έχουν εναλλασσόμενους ρόλους οδηγού/επιβάτη κάθε φορά που συνδέονται στην εφαρμογή.

Η μεταφορά υλοποιείται από μια εφαρμογή εύρεσης χρηστών, οι οποίοι θέλουν να κατευθυνθούν είτε στον ίδιο είτε σε κοντινό προορισμό. Έτσι, με επικοινωνία μέσω της εφαρμογής οι οδηγοί θα μπορούν να βρίσκουν το κατάλληλο άτομο για να μεταφέρουν και μέσα από διάφορους μηχανισμούς να γίνεται η όλη διαδικασία σχετικά σύντομα. Χρησιμοποιώντας την εφαρμογή σε καθημερινή βάση, ο οδηγός έχει ένα μικρό χρηματικό κέρδος με το οποίο θα επωφελείται για να πληρώνει π.χ. τα καύσιμα του, ένα σημαντικό πρόβλημα που συνεχώς αυξάνεται.

Ένας οδηγός μπορεί να χρησιμοποιεί την εφαρμογή, όχι μόνο για την αποκόμιση κέρδους αλλά να προσφέρει μείωση στην καθημερινή χρήση άλλων αυτοκινήτων. Με την μέθοδο αυτή ο χρήστης μπορεί να μεταφέρει ένα συγκεκριμένο πλήθος ανθρώπων καθημερινά και να αποτρέψει την χρήση οχήματος από όλους. Έτσι, με την πάροδο του χρόνου θα καταφέρουμε να μειώσουμε σε ένα ποσοστό την ατμοσφαιρική ρύπανση της εκάστοτε πόλης.

1.1 Αντικείμενο της διπλωματικής

Στοχεύουμε στο χαμηλό κόστος μεταφοράς, συνεπώς ψάχνουμε τρόπους η συνάντηση μεταξύ οδηγού και επιβάτη να γίνεται σύντομα. Έτσι η απόσταση του ενός από του άλλου δεν πρέπει να είναι αρκετά μεγάλη τόσο για το σημείο άφιξης

όσο και για το σημείο προσέλευσης του καθενός. Οι συμβατικοί τρόποι μεταφοράς χρηστών είναι τα ταξί, τα M.M.M. και η μεταφορά με το δικό του όχημα. Προσφέρουμε έναν εναλλακτικό τρόπο μεταφοράς αξιοποιώντας την εξέλιξη της τεχνολογίας και προσπαθούμε να επιλύσουμε τα παραπάνω προβλήματα.

Η εφαρμογή εμπεριέχει τα βασικά συστατικά μιας σύγχρονης εφαρμογής , το front-end και το back-end. Στο front-end, δηλαδή στο γραφικό περιβάλλον υπάρχει ένας χάρτης με στον οποίο θα εμφανίζονται οι διαδρομές των χρηστών, οι επιβάτες (αν αυτοί υπάρχουν) και τέλος ο ίδιος ο χρήστης. Η ανανέωση του χάρτη γίνεται σε πραγματικό χρόνο συνεπώς η οποιαδήποτε κίνηση προς μια κατεύθυνση του χρήστη θα εμφανίζεται αντίστοιχα στο χάρτη. Στο back-end, δηλαδή το περιβάλλον στο οποίο βρίσκεται η βάση δεδομένων. Ο χρήστης δεν θα έχει πρόσβαση και θα αποθηκεύονται σημαντικές πληροφορίες για αυτόν, όπως η τωρινή τοποθεσία του, στοιχεία προφίλ, ειδοποιήσεις κ.α.. Κάθε χρήστης θα έχει την δυνατότητα να εμφανίζεται στην εφαρμογή είτε ως επιβάτης είτε ως οδηγός ενώ δίνεται στον οδηγό, και μόνο, η δυνατότητα επιλογής επιβάτη αλλά όχι με την μέθοδο της αναζήτησης αλλά της εμφάνισης του επιβάτη στην οθόνη του και επιλογής του.

1.2 Συνεισφορά

Στην παρούσα εργασία καταφέραμε:

1. Η μεταφορά να γίνεται μόνο για κοντινές αποστάσεις.
2. Εμφανίζεται ο κοντινότερος δυνατός επιβάτης σε κάθε οδηγό
3. Εύρεση κατάλληλου σημείου για παράκαμψη από τον οδηγό προς τον πελάτη.
4. Ζωντανή παρακολούθηση τοποθεσίας.
5. Επιλογή επιβάτη μόνο από τον οδηγό.
6. Αλλαγές στον χάρτη σε πραγματικό χρόνο.
7. Εύρεση κατάλληλης τοποθεσίας

8. Σύνδεση με απομακρυσμένη βάση στο Google Cloud.
9. Άμεση επικοινωνία με την βάση χωρίς την υποστήριξη server.
10. Επικοινωνία χρηστών μέσω της βάσης υπό μορφή ειδοποιήσεων.
11. Χρήση περιβάλλοντος ψευδομεταφοράς με χρήση GPS mocking location εφαρμογής για διευκόλυνση σε πειραματικό στάδιο.
12. Διαθέσιμη online στο <https://github.com/KonstantinosPolizos/SoFair>.

1.3 Παρόμοιες εφαρμογές

Υπάρχουν αρκετές εφαρμογές μεταφοράς χρηστών και αρκετή μελέτη πάνω στο συγκεκριμένο αντικείμενο. Τέτοιες εφαρμογές είναι το [uber\[11\]](#), το [lyft\[12\]](#) κ.α. , οι οποίες έχουν αρκετές διαφορές μεταξύ τους (π.χ. πληρωμή, τρόπος διεξαγωγής ταξιδιού). Αυτές οι εφαρμογές στην βάση τους όμως έχουν κάποια στοιχεία που τις χαρακτηρίζουν.

Η κύρια σελίδα όλων είναι ένας χάρτης που μπορεί ο εκάστοτε χρήστης να θέσει έναν τελικό προορισμό, να επιλέξει ένα αυτοκίνητο και τον οδηγό της αρεσκείας του και να μεταφερθεί σε όποιο σημείο ορίσει. Ακόμη υπάρχει η δυνατότητα ζωντανής παρακολούθησης της διαδρομής κυρίως για θέματα ασφάλειας. Εγγυώνται την ασφάλεια του επιβάτη και του οδηγού καθώς παρακολουθείτε συνεχώς το αυτοκίνητο τόσο από την εφαρμογή (μέσω GPS) όσο και από την χρήση εσωτερικής κάμερας.

Ο επιβάτης μπορεί να βλέπει όλη την πορεία του οδηγού που έρχεται να τον παραλάβει αρκεί να υπάρχει η κατάλληλη επικοινωνία μεταξύ τους (π.χ. μέσω ειδοποιήσεων). Οι πληρωμές γίνονται μέσω κάρτας και κοστολογούνται συνήθως την στιγμή που εισέλθεται ο πελάτης στο όχημα καθώς το κόστος προκύπτει μόνο από την διαδρομή που ορίζει η εφαρμογή.

Η εφαρμογή μας παρέχει πολλά από τα παραπάνω στοιχεία συνεπώς ορίζεται ως μια εφαρμογή μεταφοράς. Οι κύριες διαφορές είναι τόσο σε τομέα λειτουργικότητας όσο και σε τομέα πληρωμής. Στην λειτουργικότητα στοχεύουμε κυρίως στον σχεδιασμό της διαδρομής και την εύρεση επιβατών που βρίσκονται σε κοντινή απόσταση αλλά και οδεύουν προς την ίδια κατεύθυνση με τον οδηγό τους. Δίνεται η δυνατότητα επιλογής μόνο από πλευράς οδηγού με την εύρεση κατάλληλου σημείου για την βέλτιστη διαδρομή με στόχο την εξοικονόμηση χρόνου.

Η εφαρμογή δεν στοχεύει στο freelance έτσι ώστε οι χρήστες της να πληρώνονται και να ζούν εξ ολοκλήρου από αυτήν αλλά στην δημιουργία σύντομων διαδρομών που μπορούν να τους οδηγήσουν στον προορισμό τους γρήγορα, με ασφάλεια και με ένα μικρό οικονομικό όφελος.

Άλλες υπηρεσίες που παρέχουν τέτοιου είδους εφαρμογές είναι η μεταφορά δεμάτων αλλά και η δυνατότητα ενοικίασης οχημάτων στους χρήστες - οδηγούς τους. Η εφαρμογή μας δεν παρέχει τέτοιου είδους υπηρεσίες καθώς δεν στοχεύει στην εξ ολοκλήρου απασχόληση των οδηγών και μεταφοράς πακέτων καθώς θα ξέφευγε από την βασική ιδέα των σύντομων διαδρομών.

1.4 Οργάνωση του τόμου

Στο κεφάλαιο 1 γίνεται η εισαγωγή του προβλήματος που λύνει η παρούσα διπλωματική. Ακόμη αναφέρονται κάποιες εφαρμογές ανάλογης κατηγορίας με την δική μας και σε τι διαφέρει από αυτές.

Στο κεφάλαιο 2 γίνεται αναφορά των κύριων εργαλείων που χρησιμοποιήθηκαν και το πώς αυτά μας βοήθησαν στην πρόοδο της εργασίας.

Στο κεφάλαιο 3 παρουσιάζουμε την λειτουργικότητα υπό μορφής screenshot και προσπαθούμε να εξηγήσουμε την βασική ιδέα της εφαρμογής.

Στο κεφάλαιο 4 δείχνουμε υπό μορφή σχημάτων και εικόνων τόσο την αρχιτεκτονική του front - end όσο και το σύνολο της αρχιτεκτονικής όλης της εφαρμογής. Σημαντικό υποκεφάλαιο είναι αυτό για τον τρόπο αποθήκευσης δεδομένων στην βάση αλλά και το υλικό που αποθηκεύεται.

Στο κεφάλαιο 5 εξηγούμε όλη την πορεία υλοποίησης της εφαρμογής και τον τρόπο σκέψης μας πάνω στην επίλυση προβλημάτων που παρουσιάστηκαν.

Στο κεφάλαιο 6 γίνεται αναφορά των εργαλείων που βοήθησαν στην εκτέλεση της εφαρμογής και με ποιον τρόπο έγινε.

Τέλος στο κεφάλαιο 7 παραθέτουμε τα συμπεράσματα τόσο για την εφαρμογή όσο και για τα εργαλεία που αξιοποιήσαμε αλλά και κάποιες μελλοντικές επεκτάσεις για το πως μπορούμε να την εξελίξουμε.

Κεφάλαιο 2 Εργαλεία Εφαρμογής

2.1 Εισαγωγή

Λόγω της ραγδαίας εξέλιξης και ζήτησης στον τομέα των ιστοσελίδων και των εφαρμογών παρέχονται στους προγραμματιστές πολλά εργαλεία τόσο για την δημιουργία του γραφικού περιβάλλοντος όσο και για την δημιουργία προγραμμάτων για την σύνδεση σε βάσεις δεδομένων. Χρησιμοποιήθηκαν σχετικά καινούργια εργαλεία όπως είναι η [React Native\[1\]](#) και η [Firebase\[3\]](#).

2.2 Javascript

Η [Javascript \[8\]\[13\]](#) είναι μια γλώσσα προγραμματισμού που υπάρχει στο χώρο αρκετά χρόνια και έχει ως κύριο σκοπό την δημιουργία ιστοσελίδων. Είναι δυναμική γλώσσα προγραμματισμού καθώς μπορούμε να δίνουμε στην ίδια μεταβλητή διαφορετικές τιμές συχνά διαφορετικού τύπου. Υποστηρίζει αντικειμενοστραφή αλλά και συναρτησιακό προγραμματισμό.

Οι τρεις βασικές γλώσσες για την δημιουργία ιστοσελίδων είναι οι HTML, CSS, Javascript. Η HTML δημιουργεί τον κορμό της ιστοσελίδας δηλαδή δημιουργεί στον κενό καμβά κομμάτια που μπορούμε να γράψουμε π.χ. ένα άρθρο, φόρμες για την συμπλήρωση στοιχείων όπως είναι μια σελίδα εγγραφής κ.α. . Η CSS διαλέγει τα κομμάτια που ορίζουμε στην HTML και τα μορφοποιεί κατάλληλα κυρίως για την δημιουργία μιας καλής εικόνας σε γραφικό επίπεδο για τον χρήστη. Τέλος, η Javascript τοποθετεί στα κομμάτια της HTML λειτουργικότητα και καθιστά την σελίδα δυναμική καθώς μπορεί να προσφέρει μια ιδιαιτερότητα π.χ. εμφάνιση κειμένου με το πάτημα ενός πλήκτρου μέσω του DOM(Document Object Model).

Τέλος είναι μια γλώσσα interpreted καθώς εκτελείται ο κώδικάς της γραμμή-γραμμή σε αντίθεση με την C. Παρόλο που έχει επηρεαστεί από την συγκεκριμένη γλώσσα για τον τρόπο γραφής της διαφέρει σημαντικά σε θέματα απόδοσης.

2.3 Node.js

Το [Node.js](#) [5] είναι ένα Runtime περιβάλλον που χρησιμοποιεί Javascript (εξού και η κατάληξη .js στο όνομά του) και επιτρέπει στην γλώσσα να εκτελείται σε τερματικό χωρίς την χρήση ιστού ή πιο συγκεκριμένα HTML.

Η πραγματική λειτουργία του Node.js όμως έγκειται στις υπηρεσίες server - client καθώς χρησιμοποιεί την μέθοδο μη-μπλοκαρίσματος (non-blocking) για την εξυπηρέτηση αιτημάτων (δηλαδή όταν το πρόγραμμα ζητάει υπηρεσίες εισόδου / εξόδου) από το σύστημά.

Πιο συγκεκριμένα με αυτή την μέθοδο δεν μπλοκάρει την εκτέλεση του προγράμματος αντ' αυτού εκτελείται ασύγχρονα δηλαδή δεν εκτελεί γραμμή-γραμμή τον κώδικα αλλά εκτελεί κομμάτια που είναι έτοιμα προς εκτέλεση. Έτσι, τα μπλοκαρισμένα τμήματα που δεν εκτελέστηκαν μετά από κάποια ώρα εκτελούνται, με αποτέλεσμα την βελτίωση της απόδοσης, η οποία είναι ζωτικής σημασία για τον διακομιστή. Το Node.js χρησιμοποιεί ταυτόχρονη εκτέλεση αιτημάτων, παρά το γεγονός του ενός νήματος που χρησιμοποιεί, λόγω της ασύγχρονης εκτέλεσης.

Ένα ακόμα στοιχείο που κάνει το Node.js δημοφιλές εργαλείο είναι η μεγάλη βιβλιοθήκη πακέτων του που διαχειρίζονται από τον NPM. Είναι μια συλλογή χρήσιμων προγραμμάτων που βοηθούν στην δημιουργία εφαρμογών και ιστοσελίδων.

2.4 NPM

Τα πακέτα είναι ένα εργαλείο για τη δημιουργία εφαρμογών. Με τη χρήση αυτών, μπορούμε να αυξήσουμε τις δυνατότητες μιας εφαρμογής χωρίς να δημιουργούμε νέο κώδικα. Δηλαδή ένα σημαντικό πακέτο για την εφαρμογή ήταν το react-native-maps που παρέχει όλες τις λειτουργίες στον χάρτη. Χωρίς αυτό το πακέτο της React Native καλούμαστε να δημιουργήσουμε έναν δικό μας χάρτη ο

οποίος είναι μια δύσκολη διαδικασία και απαιτεί αρκετό χρόνο αλλά και βελτιστοποίηση.

Ακολούθως για όλα τα πακέτα πρέπει εμείς να κατεβάζουμε, να εγκαθιστούμε και να συνδέουμε το κάθε πακέτο με την εφαρμογή, το οποίο μοιάζει ακατόρθωτο ιδιαίτερα όταν έχουμε να κάνουμε με πολλές δεκάδες πακέτα. Την λύση σε αυτό το πρόβλημα φέρνει ο διαχειριστής πακέτων. Είναι μια συλλογή εργαλείων που αυτοματοποιεί τη διαδικασία εγκατάστασης, αναβάθμισης, προσαρμογής και απεγκατάστασης προγραμμάτων. Ένας γνωστός διαχειριστής πακέτων είναι ο apt του Ubuntu.

Στην παρούσα εργασία χρησιμοποιήθηκε το [Node Package Manager \(NPM\) \[10\]](#), καθώς είναι ο διαχειριστής πακέτων του Node.js. Ο NPM είναι μια συλλογή εργαλείων για τη δημιουργία ιστοσελίδων και εφαρμογών. Με κυριότερα στοιχεία το Command Line Interface(CLI) που είναι η διεπαφή για το τερματικό, καθώς επιτρέπει να εκτελούμε λειτουργίες απευθείας από το τερματικό, και μία ιστοσελίδα (τύπου Github) για την δημοσίευση πακέτων.

Κατά την αρχικοποίηση της εφαρμογής δημιουργείται ένα αρχείο package.json που εμπεριέχει τις εξαρτήσεις και τις πληροφορίες του προγράμματος. Το NPM αξιοποιεί αυτό το αρχείο για την εγκατάσταση τόσο των εξαρτήσεων(Dependencies) όσο και των πακέτων.

Ακολουθούν οι πιο βασικές εντολές NPM:

- npm init: περιλαμβάνει σημαντικές πληροφορίες όταν δημιουργείται μια εφαρμογή.
- npm install: εγκαθιστά ένα πακέτο που περιέχει όλες τις εξαρτήσεις.
- npm uninstall <package-name>: αφαιρεί ένα συγκεκριμένο πακέτο.

2.5 React Native

Για τη δημιουργία της εφαρμογής χρησιμοποιήθηκε η [React Native \[2\]](#). Το JavaScript framework του Facebook το οποίο δημιουργήθηκε για κινητές συσκευές και όχι για προγράμματα περιήγησης (π.χ. Chrome).

Επιπλέον, η React Native διευκολύνει την ταυτόχρονη δημιουργία τόσο για Android όσο και για iOS, επειδή το μεγαλύτερο μέρος του κώδικα που δημιουργείται μπορεί να εκτελεστεί και στις δυο πλατφόρμες. Έτσι, έχουμε την δυνατότητα πλέον να δημιουργούμε εγγενείς εφαρμογές, δηλαδή σαν να έχουν υλοποιηθεί στην γλώσσα προγραμματισμού του λειτουργικού που τρέχουν (π.χ. Kotlin για Android).

Οι εφαρμογές δημιουργούνται χρησιμοποιώντας το JSX (έναν συνδυασμό JavaScript και XML) και της ένωσης μεταξύ του API της React Native με την java-kotlin για Android και με την swift για iOS που εκτελείται στο παρασκήνιο. Αυτό έχει ως αποτέλεσμα να έχει η εφαρμογή μας την ίδια αίσθηση με άλλες εφαρμογές (σε άλλη γλώσσα προγραμματισμού) και να μην είναι απλά μια ιστοσελίδα που προσαρμόζεται στα πλαίσια της οθόνης του κινητού. Ένα ακόμη ζήτημα μεγάλης σημασίας είναι η απόδοση της εφαρμογής. Η λύση που δίνει η React Native είναι στην χρήση πολυνηματισμού δηλαδή χρησιμοποιεί ένα νήμα για το γραφικό περιβάλλον που φαίνεται στον χρήστη και από ένα νήμα για σύνδεση με Android και iOS αντίστοιχα. Έτσι κάθε φορά που αλλάζει η κατάσταση της εφαρμογής γίνεται ενημέρωση και στα δύο νήματα μειώνοντας σημαντικά τους πόρους που απαιτούνται.

Σημαντικό σε επίπεδο δημιουργίας είναι πως με το ίδιο σύνολο γνώσεων που έχουμε στις ιστοσελίδες χρησιμοποιώντας την React, μπορούμε πλέον να το αξιοποιήσουμε και να στραφούμε στον προγραμματισμό εφαρμογών. Σε οικονομικό επίπεδο για μια εταιρία, αφαιρεί την ανάγκη για περισσότερο προσωπικό καθώς χρειάζεται μόνο μια ομάδα που θα γράφει κώδικα σε React Native και όχι ξεχωριστές για κάθε πλατφόρμα και για τον ιστό. Όμως, και για έναν προγραμματιστή που θέλει να δημιουργεί εφαρμογές cross-platform είναι η καλύτερη λύση.

2.6 Firebase

Η [Firebase](#) [4] είναι μια βάση δεδομένων, που ανήκει στην Google και εκτελείται ταυτόχρονα σε πολλά μηχανήματα μέσω του Google Cloud. Παρέχει αρκετά εργαλεία που βοηθούν στην δημιουργία ποιοτικών και γρήγορων εφαρμογών. Είναι NoSQL βάση καθώς αποθηκεύει όλα τα δεδομένα της με μορφή JSON και δεν απαιτεί SQL ερωτήματα για να επικοινωνήσουμε με αυτήν.

Τα δεδομένα της εκάστοτε εφαρμογής που χρησιμοποιεί Firebase για το backend λειτουργούν ως αντίγραφα τοπικά στο κινητό και τα στέλνει στο cloud server διασφαλίζοντας ότι και οι δύο εκδόσεις είναι ακριβείς. Αυτό γίνεται για να βοηθήσει τις εφαρμογές να αξιοποιούν περισσότερο το επεξεργαστή τοπικά χωρίς να χρειάζεται να υπάρχει ανοιχτή μια βάση δεδομένων, όπως η MySQL, που θα τον επιβάρυναν με τις διάφορες διεργασίες της.

Στην παρούσα εφαρμογή χρησιμοποιήθηκε η πραγματικού χρόνου βάση δεδομένων της Firebase (Realtime Database), καθώς παρέχει πραγματικού χρόνου ενημερώσεις και συγχρονισμό μεταξύ των δεδομένων της εφαρμογής. Στην συγκεκριμένη περίπτωση χρειάστηκε όταν ένας χρήστης - επιβάτης αλλάζει τοποθεσία ή δημιουργείται η διαδρομή μεταξύ αυτού και του τελικού προορισμού του ενώ παράλληλα ένας οδηγός κάνει ακριβώς το ίδιο και ενεργοποιεί το κουμπί Active για να εμφανιστεί κάποιος που χρειάζεται μεταφορά.

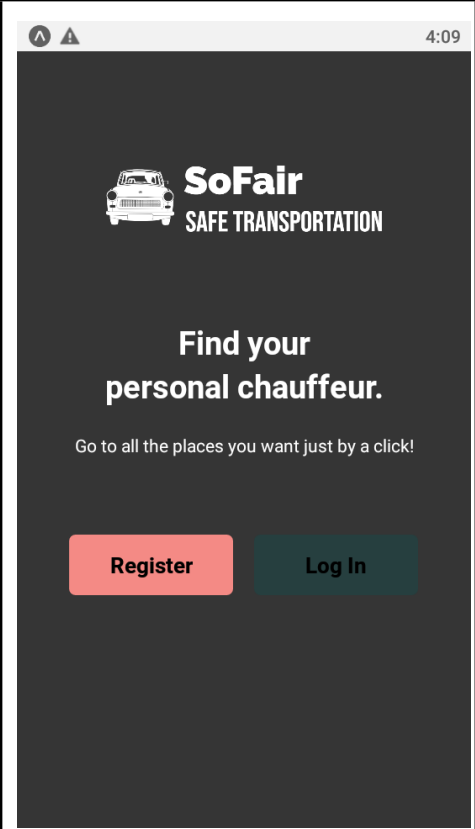
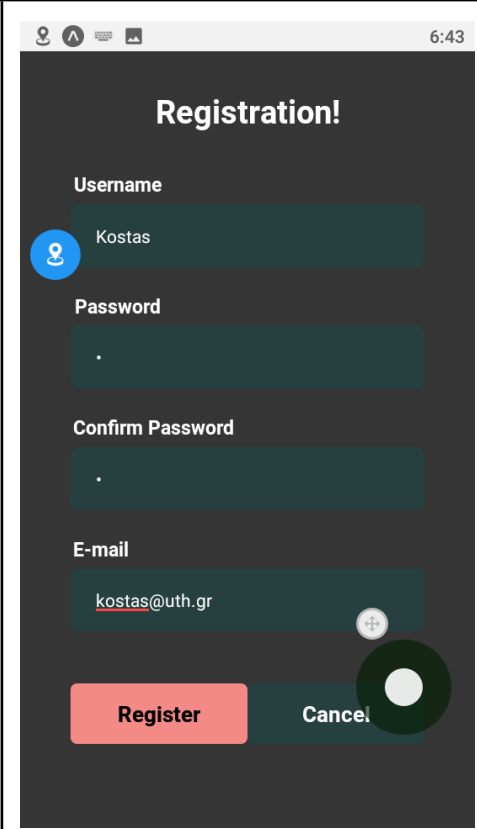
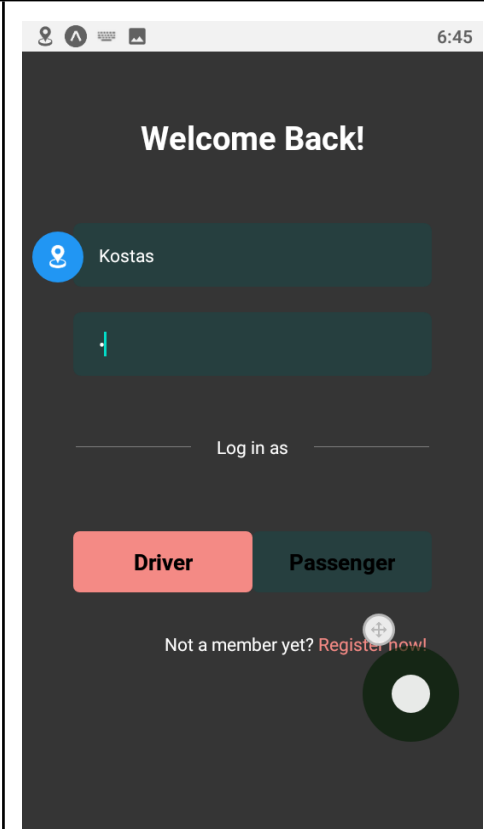
Μία σημαντική λειτουργία που έχει η πραγματικού χρόνου βάση της είναι η υποστήριξη σε offline λειτουργία. Για παράδειγμα, όταν η εφαρμογή σταματήσει για οποιοδήποτε λόγο (π.χ. σύνδεσης στο ίντερνετ) τα δεδομένα δεν χάνονται. Αντ' αυτού χρησιμοποιεί τοπικό caching στην συσκευή και αποθηκεύει τα δεδομένα αλλά και τα αξιοποιεί αν αυτό είναι απαραίτητο μέχρι να ξαναγυρίσουμε σε κατάσταση online και να ενημερώσει - συγχρονίσει την βάση με τα νέα δεδομένα αλλά και μέσω του cloud να ενημερώσει όλες τις συνδεδεμένες συσκευές που τα αξιοποιούν.

Κεφάλαιο 3 Παρουσίαση λειτουργικότητας

Στόχος του κεφαλαίου αυτού είναι να δείξουμε στον αναγνώστη το πως θα χρησιμοποιήσει την εφαρμογή. Απαραίτητη προϋπόθεση είναι να δημιουργήσει λογαριασμό και ακολούθως να αποδεχτεί αιτήματα για ζωντανή παρακολούθηση τοποθεσίας. Μέσα λοιπόν από το παρακάτω παράδειγμα θα είναι σε θέση τόσο να την χρησιμοποιήσει αλλά θα γνωρίζει το τι θα ακολουθεί σε κάθε στάδιο που θα του εμφανίζεται.

Στην εικόνα 1, βλέπουμε την πρώτη σελίδα που θα εμφανιστεί μόλις ανοίξουμε την εφαρμογή. Ο χρήστης καλείται να επιλέξει μια από τις δύο επιλογές που του παρέχονται για να συνδεθεί αν είναι μέλος. Ξεκινώντας λοιπόν (στο παράδειγμα) ως ένα νέο μέλος στην εικόνα 2 βλέπουμε την φόρμα που καλείται να συμπληρώσει κανείς για να δημιουργήσει τον λογαριασμό του.

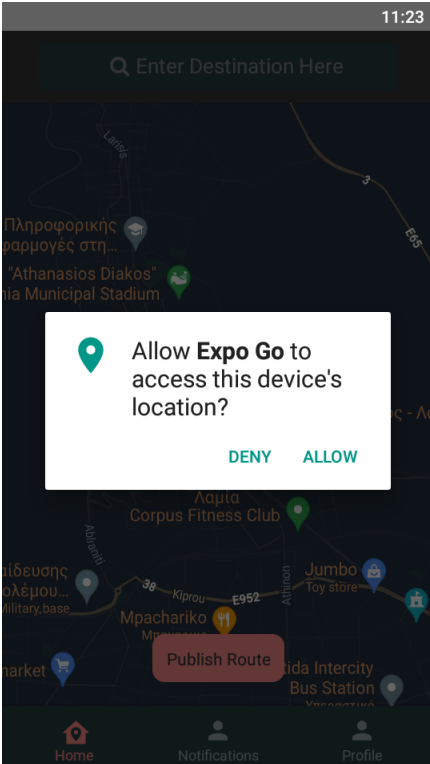
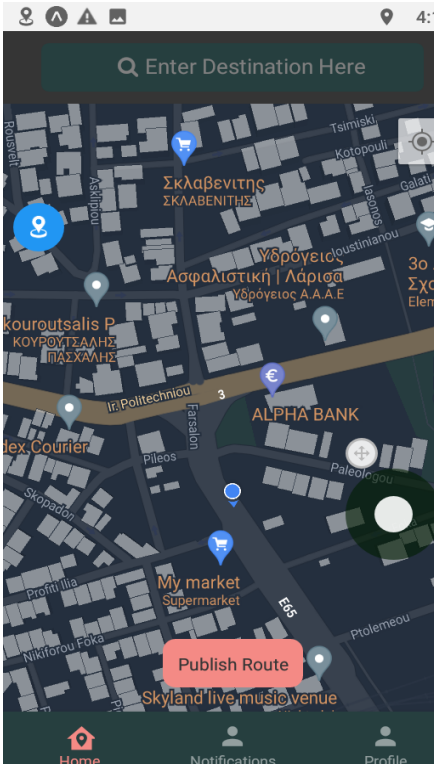
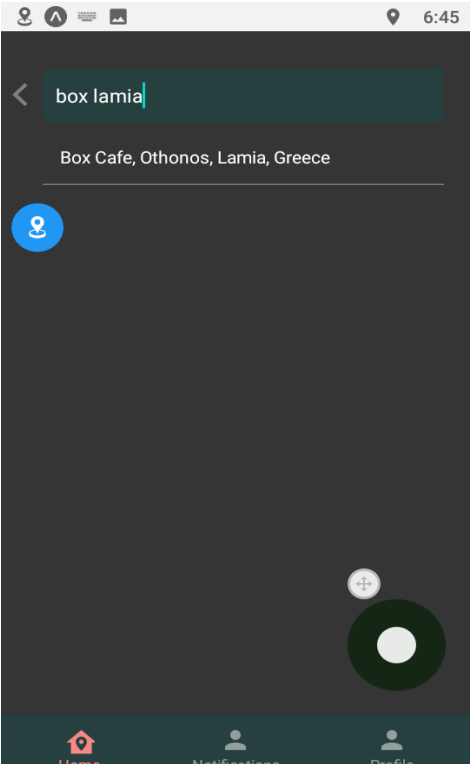
Στην εικόνα 3 βλέπουμε την φόρμα εισόδου, ο χρήστης συμπληρώνει το όνομα και τον κωδικό που έδωσε στην φόρμα εγγραφής και επιλέγει τον τρόπο εισόδου του (οδηγός - επιβάτης). Επίσης μπορεί σε περίπτωση που δεν είναι μέλος να επιλέξει το πλήκτρο Register Now και να ανακατευθυνθεί στην φόρμα εγγραφής που προαναφέραμε.

 The image shows the SoFair app's home screen. At the top, there's a status bar with the time 4:09. The app header features a car icon and the text 'SoFair SAFE TRANSPORTATION'. The main heading is 'Find your personal chauffeur.' followed by the subtext 'Go to all the places you want just by a click!'. At the bottom, there are two buttons: 'Register' (red) and 'Log In' (dark green).	 The image shows the 'Registration!' screen. It has a status bar with the time 6:43. The form includes fields for 'Username' (filled with 'Kostas'), 'Password', 'Confirm Password', and 'E-mail' (filled with 'kostas@uth.gr'). There are 'Register' (red) and 'Cancel' (dark green) buttons at the bottom. A large green circular arrow icon is overlaid on the bottom right.	 The image shows the 'Welcome Back!' screen. It has a status bar with the time 6:45. It displays the user 'Kostas' and a 'Log in as' section. Below are 'Driver' (red) and 'Passenger' (dark green) buttons. At the bottom, it says 'Not a member yet? Register now!' with a plus icon. A large green circular arrow icon is overlaid on the bottom right.
Εικόνα 1 Εικόνα εισόδου	Εικόνα 2 Φόρμα Εγγραφής	Εικόνα 3 Φόρμα εισόδου

Στην εικόνα 4 βλέπουμε τι εμφανίζεται όταν συνδεθούμε για πρώτη φορά στην εφαρμογή είτε ως οδηγός είτε ως επιβάτης και είναι απαραίτητη η επιλογή ALLOW, να ενεργοποιήσουμε το GPS και τα δεδομένα μας. Το συγκεκριμένο μήνυμα δεν ξαναεμφανίζεται κάθε φορά που συνδεόμαστε καθώς αποθηκεύεται αυτόματα στις ρυθμίσεις του κινητού.

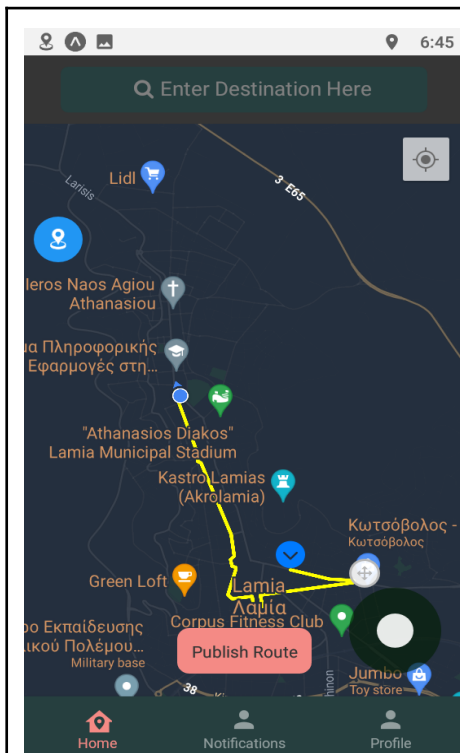
Στην εικόνα 5 βλέπουμε τον χρήστη να εμφανίζεται στο χάρτη βάσει της τοποθεσίας του και εμφανίζεται ως ένας κύκλος με ένα βέλος μπροστά του που δείχνει την κατεύθυνση του. Στην συγκεκριμένη εικόνα ο χρήστης κάνει zoom in για να του εμφανιστούν περισσότερες λεπτομέρειες για την τοποθεσία του (οδοί, καταστήματα κλπ). Ακριβώς πάνω από τον χάρτη είναι η γραμμή αναζήτησης υπό μορφή πλήκτρου έτσι ώστε όταν πατηθεί να μας μεταφέρει στην σελίδα αναζήτησης τοποθεσίας.

Στην εικόνα 6 βλέπουμε την σελίδα αναζήτησης τοποθεσίας και ο χρήστης δίνει τον τελικό προορισμό του πληκτρολογώντας τον και μέσω του autocomplete παίρνει πιθανά σημεία που θα ήθελε. Επιλέγουμε και γίνεται ανακατεύθυνση της σελίδας επιστρέφοντας στον χάρτη.

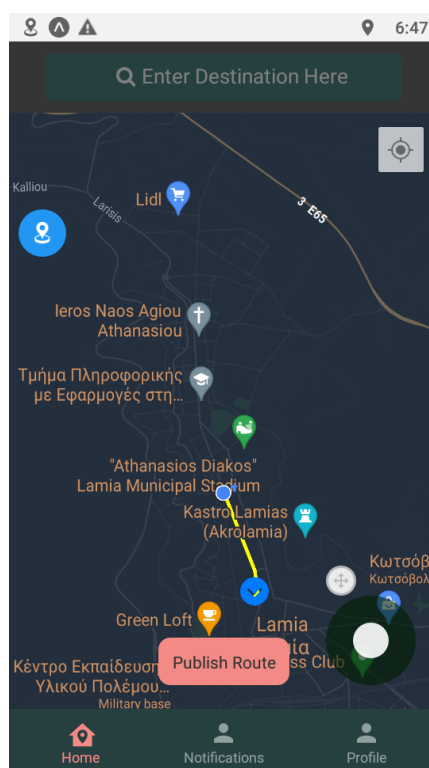
		
Εικόνα 4 Άδεια για GPS	Εικόνα 5 Zoom in/out στο χάρτη	Εικόνα 6 Σελίδα αναζήτησης τοποθεσίας

Στην εικόνα 7, μετά την ανακατεύθυνση, βλέπουμε να χαράζεται η διαδρομή από την αρχική θέση του χρήστη έως την τελική με κίτρινο χρώμα για τον οδηγό. Στην εικόνα 8 βλέπουμε να χαράζεται η διαδρομή του επιβάτη επίσης με κίτρινο και είναι ουσιαστικά η οθόνη του επιβάτη μόνο (δεν δείξαμε την είσοδο και επιλογή διαδρομής όπως κάναμε με τον οδηγό καθώς είναι ακριβώς ίδια διαδικασία). Στην εικόνα 9 βλέπουμε τον οδηγό να πλοηγείται στην σελίδα του προσωπικού προφίλ

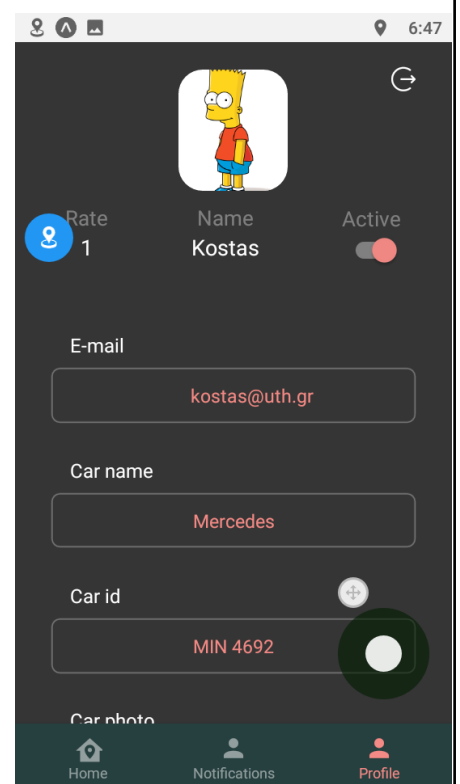
του και να ενεργοποιεί το πλήκτρο κύλισης Active έτσι ώστε να του εμφανιστούν πιθανοί επιβάτες.



Εικόνα 7 Σχεδιασμός διαδρομής με κίτρινο (οδηγού)

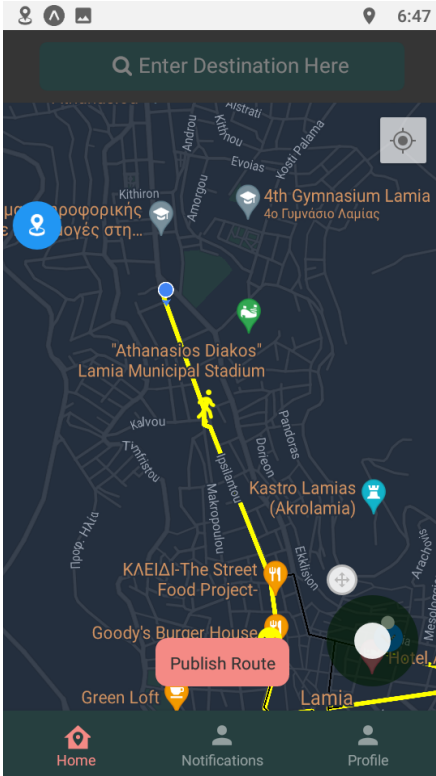
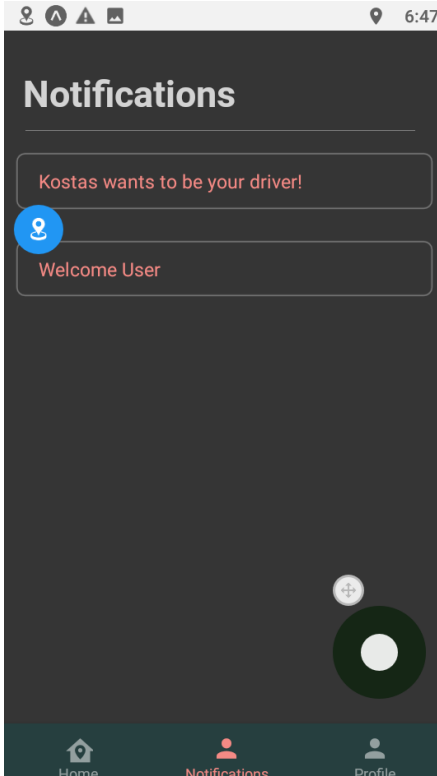
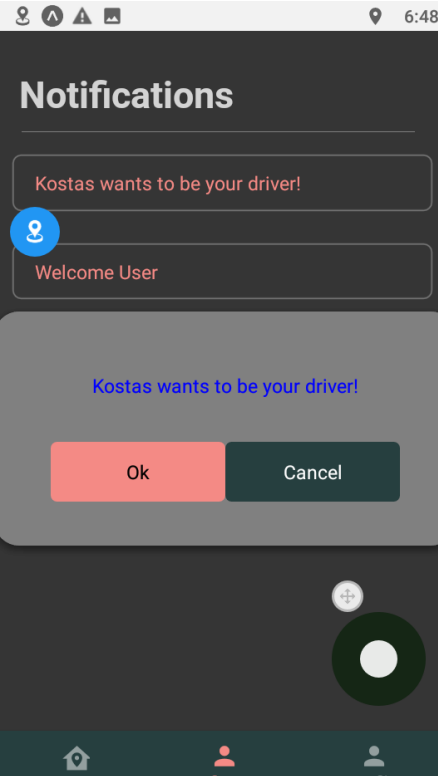


Εικόνα 8 Σχεδιασμός διαδρομής με κίτρινο (επιβάτη)

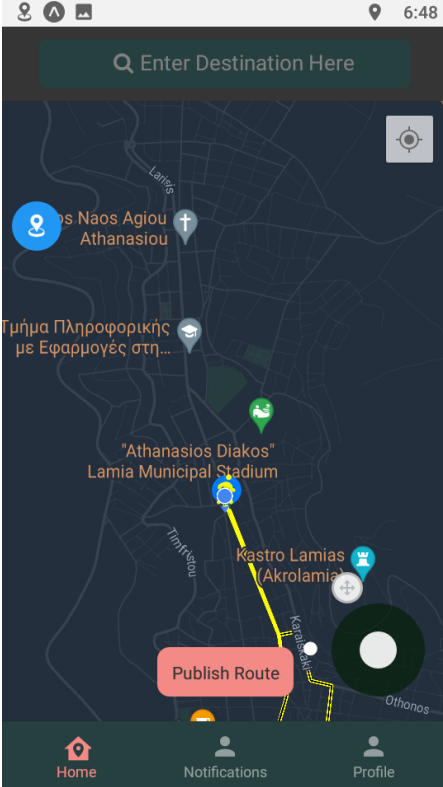
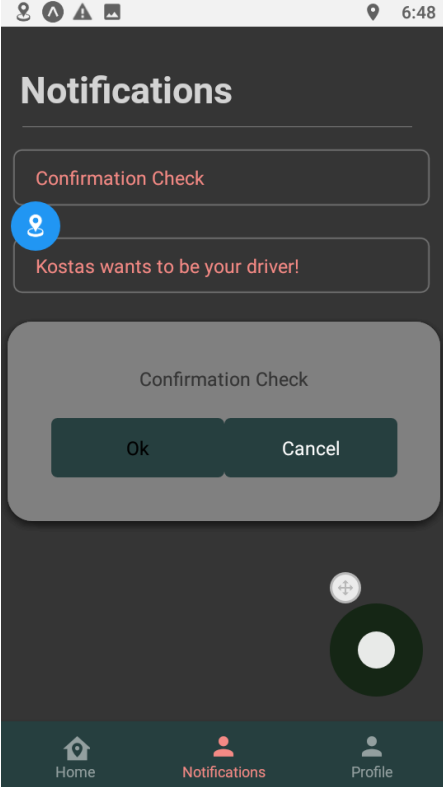
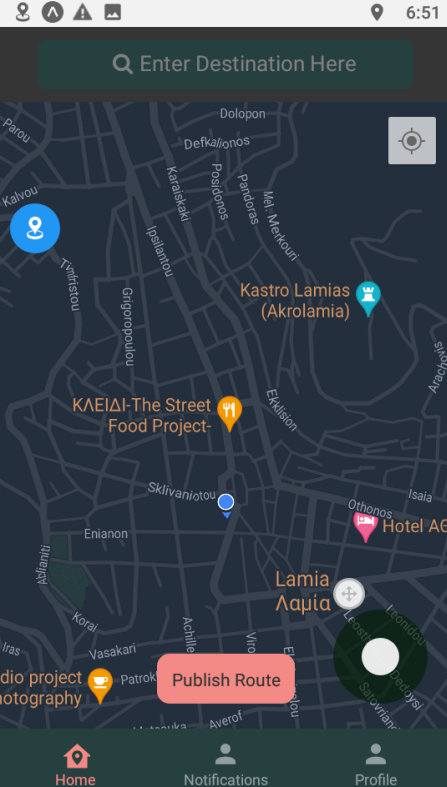


Εικόνα 9 Κουμπί ενεργού οδηγού στο προφίλ του

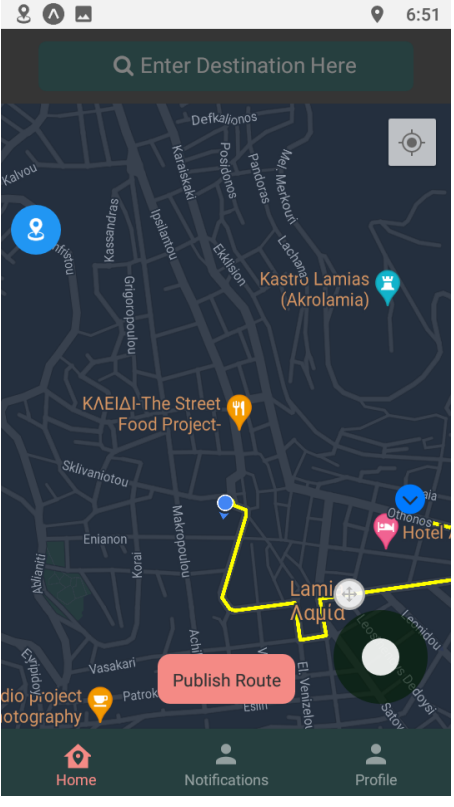
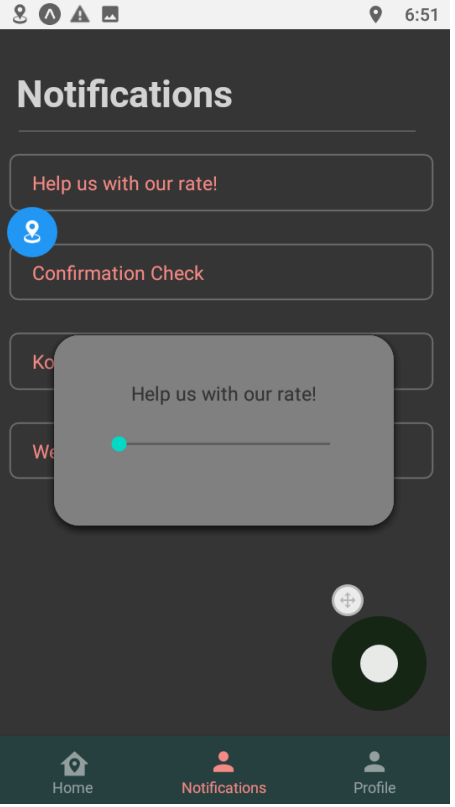
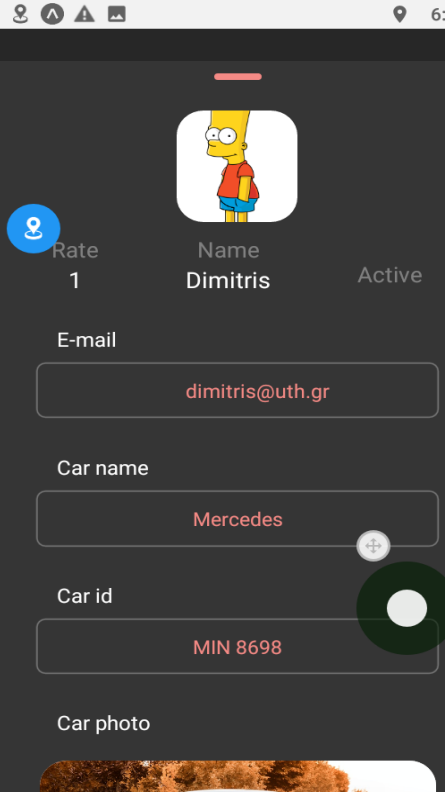
Στην εικόνα 10 βλέπουμε το πως εμφανίζεται ο επιβάτης και η διαδρομή του στην οθόνη του οδηγού. Μετά την επιλογή του επιβάτη από τον οδηγό στην εικόνα 11 βλέπουμε την ειδοποίηση που ήρθε στον επιβάτη και στην εικόνα 12 το άνοιγμά της.

		
Εικόνα 10 Εμφάνιση πελάτη	Εικόνα 11 Ειδοποίηση Επιβάτη για πιθανό οδηγό	Εικόνα 12 Άνοιγμα ειδοποίησης πιθανού οδηγού

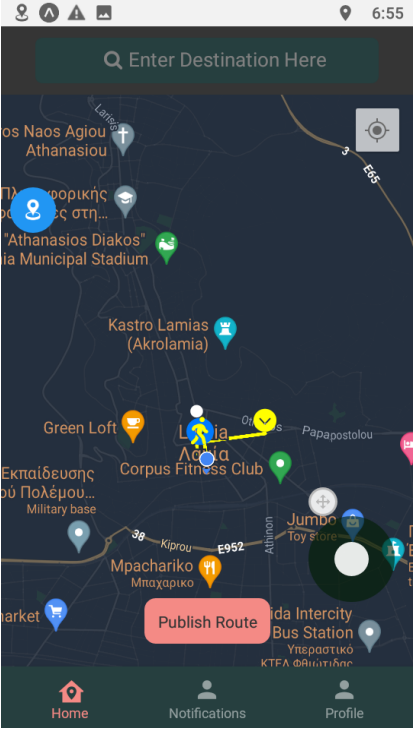
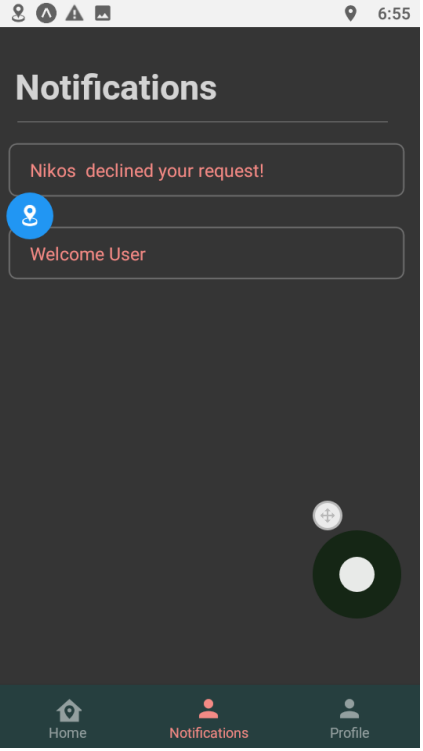
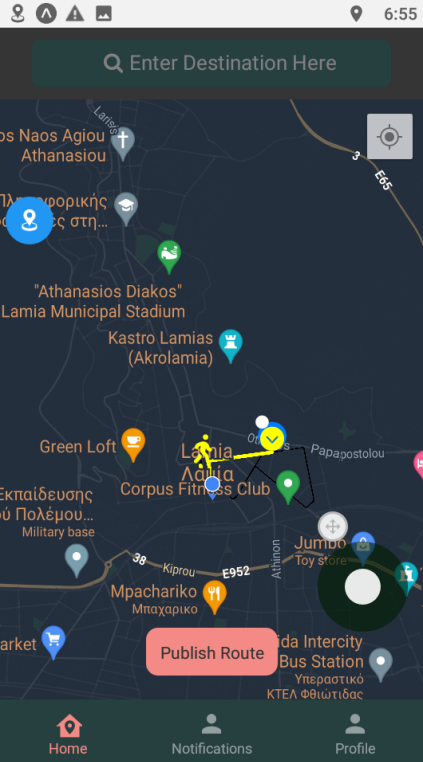
Στην εικόνα 13 βλέπουμε τον οδηγό μας (καθώς βρισκόμαστε στον χάρτη του οδηγού) να καταφθάνει στην αρχική τοποθεσία του επιβάτη. Στην εικόνα 14 αποστέλλεται και στους δύο την στιγμή που γίνεται η συνάντηση ειδοποίηση για το confirmation. Στην εικόνα 15 βλέπουμε την αποβίβαση του πελάτη στον τελικό του προορισμό και την διαγραφή της τελικής τοποθεσίας που είχε ορίσει. Δεν υπάρχει δηλαδή κίτρινη γραμμή (διαδρομής) και τελικός προορισμός (κίτρινος κύκλος με εσωτερικό βέλος).

		
Εικόνα 13 Άφιξη οδηγού και παραλαβής επιβάτη	Εικόνα 14 Ειδοποίηση confirm και για τους δύο	Εικόνα 15 Άφιξη επιβάτη στον τελικό του προορισμό

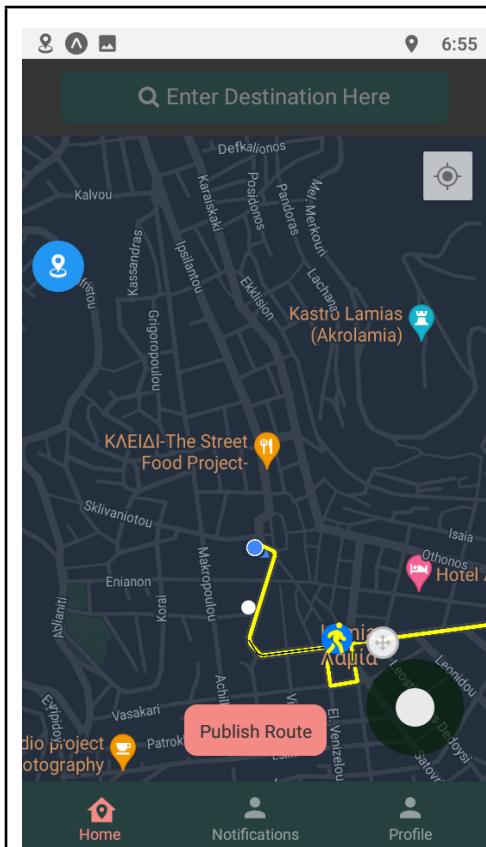
Στην εικόνα 16 βλέπουμε ότι κατά την ολοκλήρωση της διαδρομής του επιβάτη στην οθόνη πλέον του οδηγού δεν υπάρχει ο επιβάτης αλλά μόνο η διαδρομή που πρέπει να διανύσει ο ίδιος για να φτάσει στον τελικό προορισμό του. Στην εικόνα 17 βλέπουμε την ειδοποίηση που καταφθάνει στον πελάτη όταν ολοκληρώνεται η διαδρομή του για να βαθμολογήσει τον οδηγό του. Στην εικόνα 18 βλέπουμε την προσθήκη ενός νέου οδηγού και πως αυτός ενεργοποιεί επίσης το πλήκτρο Active για να δεχτεί κάποιον πελάτη.

		
Εικόνα 16 Επαναφορά τελικού προορισμού οδηγού	Εικόνα 17 Βαθμολόγηση επιβάτη προς οδηγό	Εικόνα 18 Ενεργοποίηση νέου οδηγού

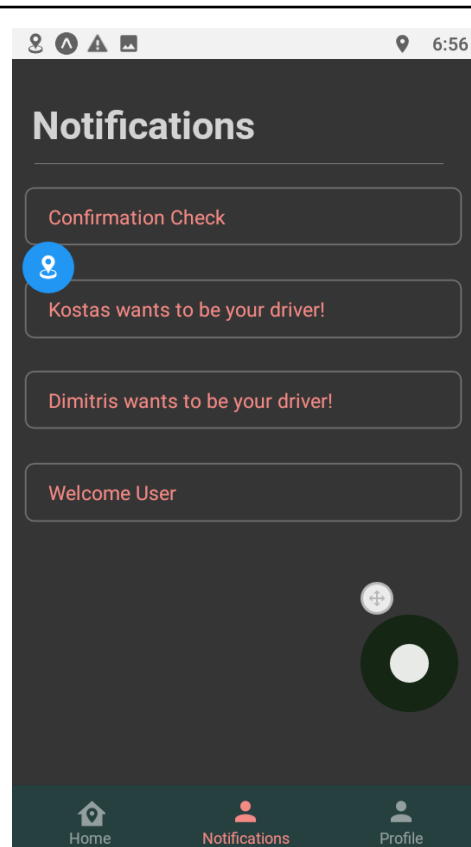
Στην εικόνα 19 βλέπουμε ότι μετά την προσθήκη δύο νέων χρηστών ενός επιβάτη και ενός οδηγού (παράλληλα ο προηγούμενος οδηγός είναι ακόμα ενεργός και κατευθύνεται προς την διαδρομή του) ο νέος οδηγός να επιλέγει τον νέο επιβάτη. Στην εικόνα 20 όμως υπάρχει απόρριψη από πλευράς πελάτη και αποστολής κατάλληλης ειδοποίησης στον οδηγό. Έτσι στην εικόνα 21 βλέπουμε να ξεμαρκάρετε ο επιλεγμένος πελάτης και να είναι πλέον ελεύθερος για επιλογή από άλλον οδηγό.

		
Εικόνα 19 Επιλογή επιβάτη από οδηγό	Εικόνα 20 Ειδοποίηση απόρριψης	Εικόνα 21 Ξεμαρκάρισμα μετά την απόρριψη

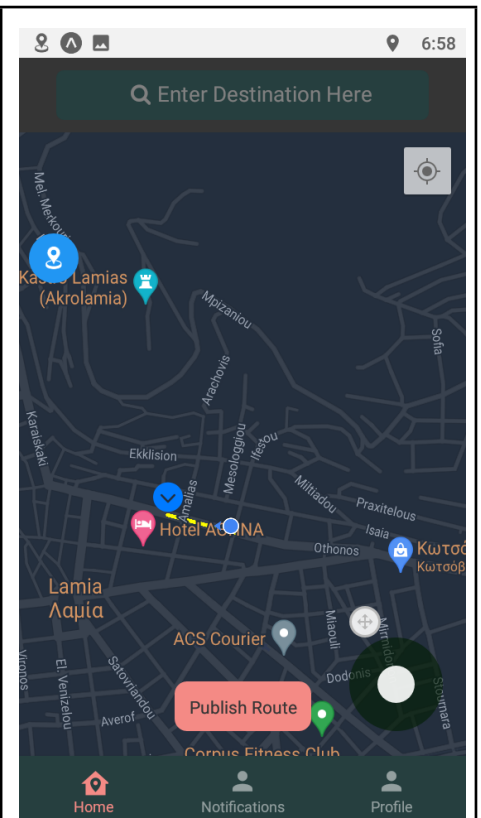
Στην εικόνα 22 βλέπουμε την επιλογή του πελάτη που μόλις απέρριψε τον νέο οδηγό να εμφανίζεται στην οθόνη του παλαιού οδηγού καθώς είναι στην διαδρομή του. Εμφανίζεται επίσης μια λευκή κουκίδα που είναι το σημείο παράκαμψης για την παραλαβή του επιβάτη. Έτσι με κατάλληλη επικοινωνία μέσω ειδοποιήσεων όπως βλέπουμε στην εικόνα 23 ο επιβάτης φτάνει στον προορισμό του. Τέλος στην εικόνα 24 ο οδηγός φτάνει στο δικό του προορισμό.



Εικόνα 22 Επιλογή επιβάτη από οδηγό



Εικόνα 23 Κατάλληλες ειδοποιήσεις επικοινωνίας



Εικόνα 24 Άφιξη οδηγού στον προορισμό του

Κεφάλαιο 4 Αρχιτεκτονική εφαρμογής

4.1 Αρχιτεκτονική front - end

Μεταφερόμαστε στον χάρτη, καθώς είναι το κύριο component της εργασίας. Ο χάρτης αποτελείται από το Google Maps, εργαλείο που μπορούμε να έχουμε μέσω ενός κλειδιού API που παίρνουμε από την Google και μια συνάρτηση της React Native. Έτσι εμείς καλούμαστε να δημιουργήσουμε την επικείμενη λειτουργικότητα σε αυτόν.

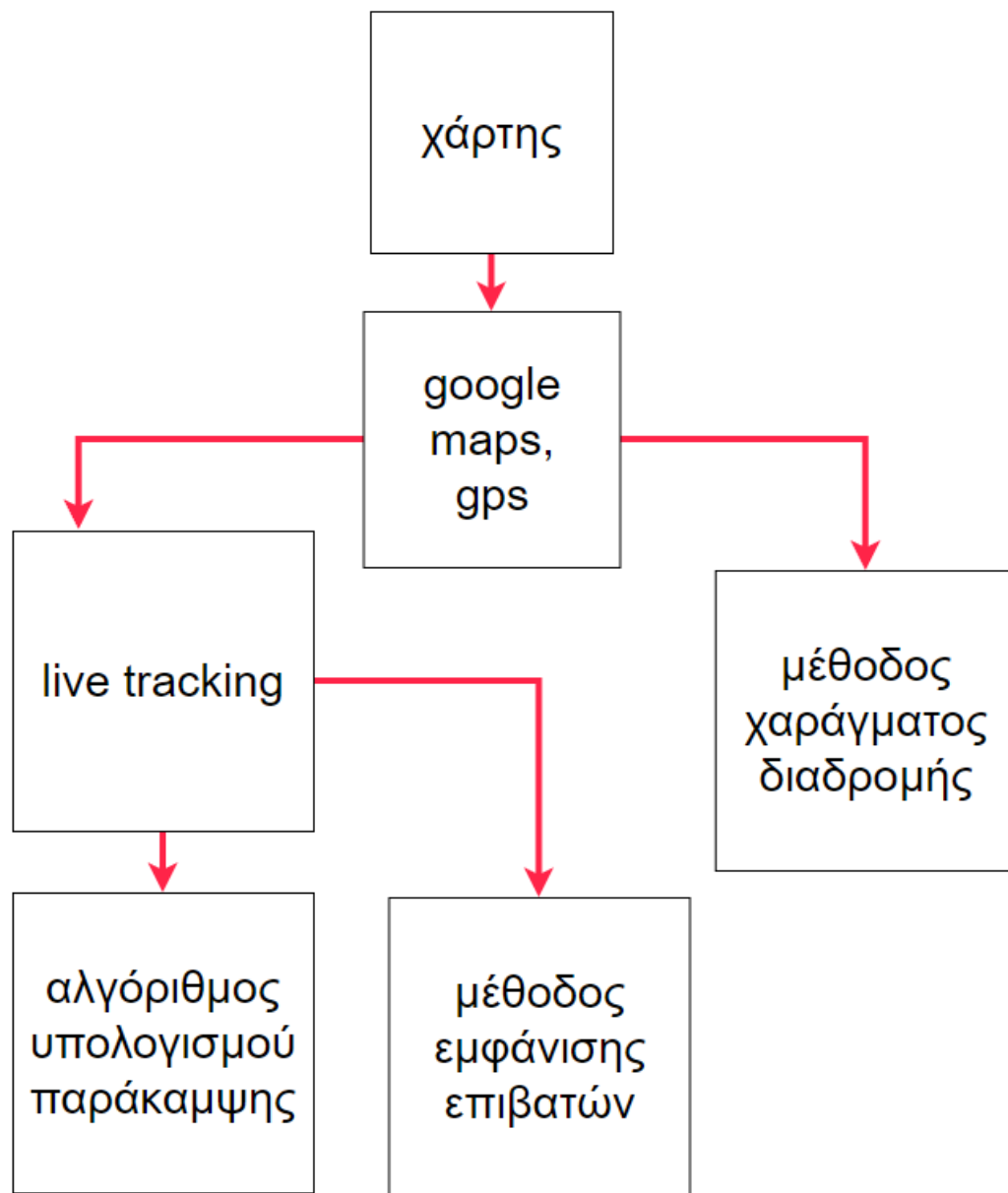
Ο χάρτης έχει δύο κύριες λειτουργίες, την ζωντανή παρακολούθηση της διαδρομής του κάθε χρήστη αλλά και την δημιουργία αυτής της διαδρομής. Η δημιουργία της τελικής διαδρομής αξιοποιεί την συνάρτηση <MapViewDirection/> της React Native η οποία χαράζει και εμφανίζει, στο γραφικό περιβάλλον, μια διαδρομή μεταξύ συντεταγμένων. Σημαντικό είναι επίσης όταν επιλέγεται ένας επιβάτης ο αλγόριθμος θα του αλλάξει προσωρινά την τελική τοποθεσία του οδηγού καθώς πρώτα πρέπει να παραλάβει τον πελάτη του. Κατά την άφιξη του πελάτη του όμως επιστρέφει τη δική του τελική τοποθεσία στον χάρτη που την βρίσκει στην βάση δεδομένων.

Για την δημιουργία της ζωντανή παρακολούθηση έπρεπε να έχουμε πρόσβαση στο GPS του κινητού. Εφόσον, όμως το να αξιοποιούμε τις λειτουργίες του κινητού είναι χαρακτηριστικό της React Native φτάνει να δημιουργήσουμε κάποιες άδειες που ο χρήστης πρέπει να αποδεχθεί, όπως το μήνυμα για πρόσβαση στην θέση της συσκευής. Σκοπός της ζωντανής παρακολούθησης όμως δεν είναι απλά να κάνει την εφαρμογή διαδραστική αλλά να εμφανίσει του επιβάτες στην οθόνη του οδηγού και τα καλύτερα δυνατά σημεία παράκαμψης προς αυτούς.

Ο αλγόριθμος εμφάνισης των επιβατών λειτουργεί όταν ο χρήστης - οδηγός ενεργοποιήσει το πλήκτρο κύλισης Active. Εκείνη την στιγμή δημιουργείται μια λίστα προς αντικείμενο , η οποία περιέχει πληροφορία όπως οι συντεταγμένες προορισμού του επιβάτη. Εφαρμόζοντας κάποιους ελέγχους απόστασης, μεταξύ

αρχικής και τελικής θέσης οδηγού με κάθε επιβάτη στην λίστα, συμπεραίνουμε αν πρέπει να εμφανιστεί ή όχι αυτός ο χρήστης στην οθόνη του οδηγού.

Κατά την εμφάνιση πλέον των επιβατών στον χάρτη του οδηγού, ενεργοποιείται αυτόματα ο αλγόριθμος εύρεσης της καλύτερης δυνατής παράκαμψης. Ο αλγόριθμος ξεκινάει από τυχαίο σημείο στην διαδρομή του οδηγού και προσπαθεί να ορίσει σημείο (υπαρκτό) για κάθε επιβάτη. Πιο συγκεκριμένα κατά την δημιουργία της διαδρομής του οδηγού η συνάρτηση δημιουργίας διαδρομής, που προαναφέραμε, επιστρέφει έναν πίνακα από συντεταγμένες. Αυτές οι συντεταγμένες είναι η διαδρομή που έχει να διανύσει ο οδηγός για να φτάσει στον τελικό του προορισμό. Ο πίνακας είναι μεγάλος συνήθως παρά τις σύντομες διαδρομές που δημιουργούνται. Έτσι, επιλέγουμε, μετά από πειραματικό έλεγχο, 5 σημεία του πίνακα από τα οποία μετράμε αποστάσεις μεταξύ αυτών και της αρχικής θέσης του επιβάτη. Όμως, είναι αδύνατο στο δρόμο να εφαρμόσουμε ευκλείδεια απόσταση καθώς στην πραγματικότητα ποτέ δεν αντιμετωπίζουμε ευθείες. Η λύση στο πρόβλημα είναι πάλι η συνάρτηση δημιουργίας διαδρομής χωρίς όμως την εμφάνισή της στο γραφικό περιβάλλον. Έτσι, δημιουργούμε 5 διαδρομές μεταξύ των τυχαίων σημείων και της αρχικής θέσης του επιβάτη και μας επιστρέφεται πάλι πίνακας από συντεταγμένες. Συγκεκριμένα, πέντε πίνακες αν υπάρχουν και οι πέντε διαδρομές. Επιλέγουμε τον πίνακα με το μικρότερο μέγεθος ($\text{length}(\text{array}[i])$) άρα και το βέλτιστο δυνατό σημείο παράκαμψης, όπως παρατηρούμε στο σχήμα 1 παρακάτω.



Σχήμα 1 Αρχιτεκτονική front - end

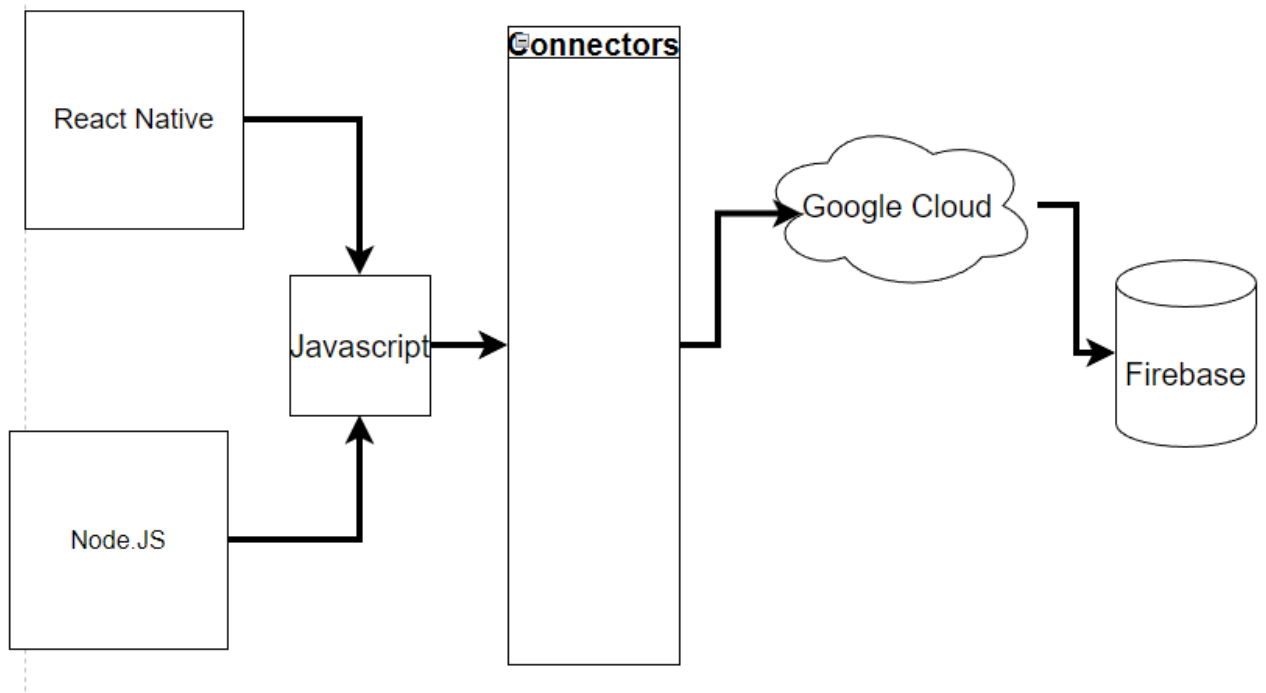
4.2 Tech Stack και η αρχιτεκτονική front - end και back - end μαζί

Η Javascript, όπως έχουμε αναφέρει, ξεκίνησε σαν μια γλώσσα ιστού δηλαδή έδινε στην βασική και στατική ιστοσελίδα λειτουργικότητα. Μέσω της Node.js καταφέρνουμε να ορίσουμε την javascript πλέον ως μια γλώσσα γενικού σκοπού, δηλαδή μια γλώσσα που θα εκτελείται τοπικά στο τερματικό και μπορεί ο προγραμματιστής να δημιουργήσει κι άλλες εφαρμογές όπως παιχνίδια, εφαρμογές για κινητά κλπ.

Μέσα από το runtime που παρέχει η Node.js και την βελτίωση της απόδοσης της γλώσσας επωφελείται η React Native. Δημιουργήθηκε δηλαδή ένα framework που μας παρέχει λειτουργίες και συναρτήσεις πέρα από τον συμβατικό τρόπο των αντικειμένων. Έτσι, το περιβάλλον της React Native είναι κατάλληλο σε αυτή την φάση να δημιουργήσει εφαρμογές κινητών που αξιοποιούν τις δυνατότητες του λειτουργικού που εγκαθίστανται.

Στην εφαρμογή αξιοποιούμε πολλές λειτουργίες του κινητού όπως το GPS, πολυμέσα κλπ. Όμως για την επικοινωνία μεταξύ χρηστών απαιτείται η δημιουργία ενός server. Την λύση μας την δίνει η firebase που καταρρίπτει τους συμβατικούς τρόπους επικοινωνίας με την βάση και αντ' αυτού παρέχει ειδικούς connectors κατευθείαν στην React Native σε javascript. Έτσι, μπορούμε να τους αξιοποιήσουμε για να στείλουμε αίτημα στην βάση είτε για διάβασμα είτε για γράψιμο σε πραγματικό χρόνο και χωρίς τις ανάγκες δημιουργίας server.

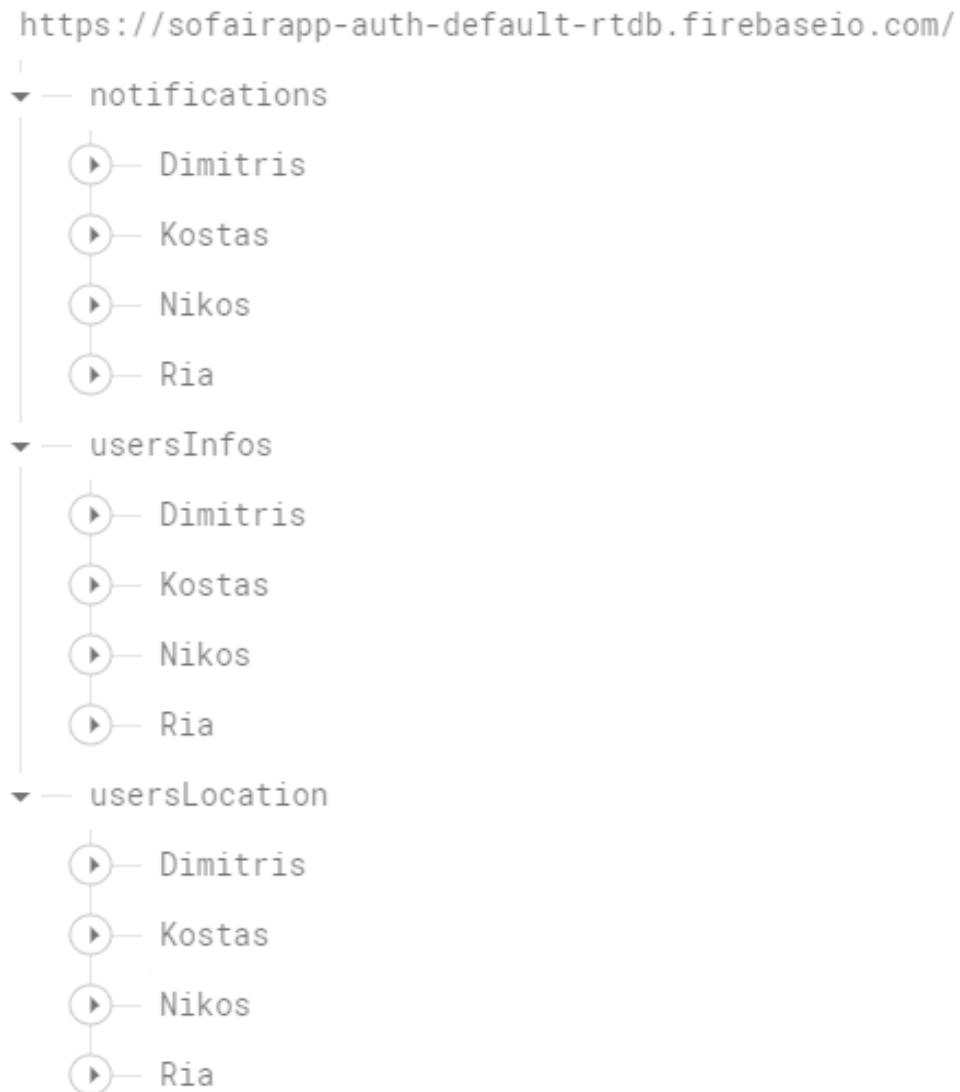
Οι connectors λειτουργούν και προς τις δύο κατευθύνσεις κάνοντας έτσι την επικοινωνία αρκετά εύκολη και με αρκετά μεγάλες ταχύτητες για τις πραγματικού χρόνου αλλαγές. Όπως βλέπουμε στο παρακάτω σχήμα 2.



Σχήμα 2 Αρχιτεκτονική Front - end και Back - end μαζί

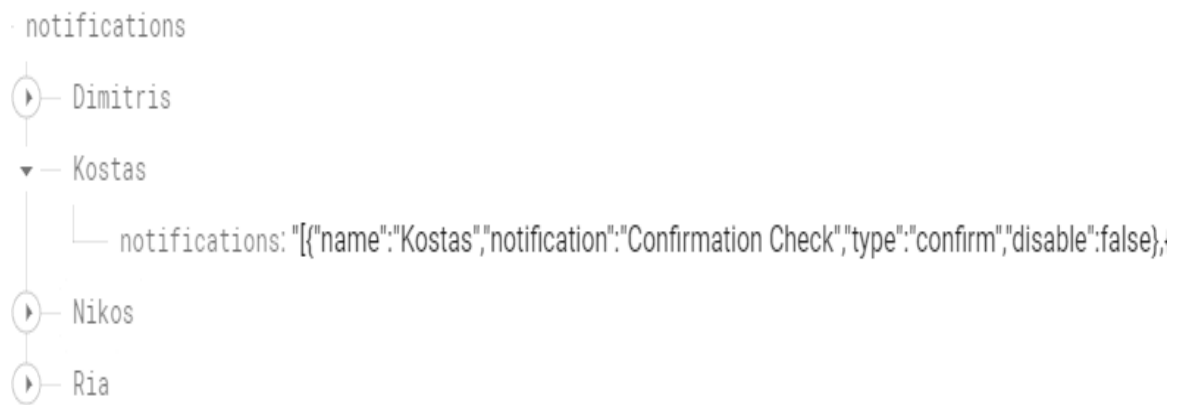
4.3 Εσωτερική δομή firebase

Η [Firebase\[6\]](#) αποθηκεύει εσωτερικά όλα τα απαραίτητα δεδομένα τόσο για την λειτουργικότητα της εφαρμογής όσο και για την επικοινωνία μεταξύ τα χρηστών. Αποτελείται από τρεις JSON δομές που κάθε μία από αυτές σε κάθε γραμμή της εμφανίζει το όνομα στο οποίο ανήκει. Η δομή είναι σαν λεξικό δηλαδή `notifications["Kostas"]`, όπως βλέπουμε στην εικόνα 25.



Εικόνα 25 Κύρια δομή Firebase

Στην εικόνα 26 βλέπουμε την εσωτερική δομή των ειδοποιήσεων. Επιλέγοντας, λοιπόν, τον χρήστη που μας ενδιαφέρει υπάρχει ένας πίνακας από αντικείμενα. Τα αντικείμενα αυτά έχουν τέσσερα πεδία: όνομα χρήστη που ανήκει, τίτλο ειδοποίησης, είδος ειδοποίησης και αν έχει ανοιχτεί. Ο πίνακας αυξάνεται κάθε φορά που υπάρχει επικοινωνία μεταξύ χρηστών για μεταφορά και δεν διαγράφεται καμία ειδοποίηση.



Εικόνα 26 Δομή notifications

Στην εικόνα 27 βλέπουμε την εσωτερική δομή στις πληροφορίες του χρήστη. Εδώ η δομή διαφέρει από την δομή των ειδοποιήσεων. Παρατηρούμε ότι υπάρχουν αρκετές μεταβλητές που κρατάνε μόνο μια τιμή. Είναι σημαντικές τόσο για την δημιουργία του προφίλ, π.χ. `imageUrl` που είναι υπεύθυνο για την φωτογραφία προφίλ, αλλά και για την λειτουργικότητα της εφαρμογής και του χάρτη, π.χ. `isActive` που αποθηκεύει σε boolean μεταβλητή αν ο οδηγός είναι ενεργός για να του εμφανιστούν πελάτες.

Δύο σημαντικές μεταβλητές είναι η `driversName` και η `passengersName`, που αποθηκεύουν τα ονόματα των χρηστών οδηγός - επιβάτη. Σημαντικό είναι ότι κατά την ολοκλήρωση του δρομολογίου και τα δύο ενημερώνονται και ξανά αδειάζουν. Στην παρούσα έκδοση της εφαρμογής ο οδηγός μπορεί να δέχεται μόνο έναν πελάτη στο όχημα του καθώς στην βάση αποθηκεύεται ένα όνομα στο `passengersName`. Σε μελλοντική έκδοση θα μπορούσε να δέχεται παραπάνω άτομα για μεταφορά με την προϋπόθεση της εφαρμογής (σύντομες αποστάσεις) και το μόνο που θα χρειαστεί από την βάση είναι η μεταβλητή αυτή να δέχεται πίνακα από ονόματα επιβατών μεγέθους 4.



Εικόνα 27 Δομή users Infos

Στην εικόνα 28 βλέπουμε την εσωτερική δομή του `usersLocation`, την δομή στην οποία αποθηκεύεται η αρχική και τελική τοποθεσία των χρηστών. Αποτελείται από δύο υποδομές την `startLocation` και την `endLocation`. Σημαντική λειτουργία, που συμβάλλουν και οι δυο υποδομές ταυτόχρονα, είναι η δημιουργία της λίστας των επιβατών όταν ο οδηγός είναι ενεργός. Η πρώτη συμβάλλει στο να βρεθούν επιβάτες στην ίδια πόλη και η δεύτερη στο να σιγουρευτούμε για την ίδια κατεύθυνση της τελικής τοποθεσίας. Ακόμη οι αλλαγές στην βάση γίνονται σε πραγματικό χρόνο καθώς θέλουμε κάθε φορά που παίρνουμε την τοποθεσία του χρήστη να αποθηκεύεται στο `startLocation`.



Εικόνα 28 Δομή users location

Κεφάλαιο 5 Πορεία υλοποίησης

Πριν ξεκινήσουμε να γράφουμε κώδικα για την υλοποίηση της εφαρμογής γνωρίζαμε ότι θα χρησιμοποιήσουμε τους χάρτες. Η Google παρέχει το Google Maps, ένα πολύ καλό εργαλείο τόσο σε επίπεδο προγραμματιστή όσο και σε επίπεδο καθημερινής χρήσης για την εύρεση κάποιου σημείου κλπ. Για να έχουμε όμως πρόσβαση στο συγκεκριμένο εργαλείο είναι υποχρεωτική η ενεργοποίηση ενός κλειδιού API που παρέχεται από την ίδια. Πρώτα επιλέγουμε τι χαρακτηριστικά θέλουμε να έχει το AP, όπως π.χ. σχεδιασμό δρόμων ή χάρτες για Android κλπ, κάνουμε προσθήκη όσων επιθυμούμε και δημιουργούμε το κλειδί. Το κλειδί παρέχεται μέσα από έναν κωδικό και διαρκεί για τρεις μήνες δωρεάν.

Έπειτα, πρέπει να αρχίσουμε να δημιουργούμε το γραφικό περιβάλλον της εφαρμογής, έστω στο πειραματικό του στάδιο, για να καταφέρουμε να τεστάρουμε τον χάρτη. Έτσι, δημιουργήσαμε τις τρεις σελίδες (home, notification, profile) και αρχίσαμε να βάζουμε components που ήταν απαραίτητα. Είναι λογικό στην αρχή της υλοποίησης της εφαρμογής η σελίδα notification να είναι άδεια καθώς δεν υπάρχει επικοινωνία με βάση δεδομένων ακόμα. Το Home και το Profile όμως υλοποιήθηκαν από την αρχή καθώς για το πρώτο έπρεπε να υπάρχει ο κατάλληλος χώρος για την προσθήκη του χάρτη και στο δεύτερο το κουμπί κύλισης Active για την ενεργοποίηση του οδηγού.

Είμαστε έτοιμοι λοιπόν να τοποθετήσουμε τον χάρτη στο Home και να δούμε αν λειτουργεί. Επιτυχώς παρατηρήσαμε ότι με την βοήθεια της React Native, ο χάρτης λειτουργούσε όπως στο Google Maps. Επειδή το χρώμα της εφαρμογής επιλέχθηκε από την αρχή να είναι σκουρόχρωμο αλλάξαμε το style του χάρτη με κατάλληλο τρόπο έτσι ώστε να είναι ευδιάκριτα όλα τα στοιχεία σε αυτόν.

Ακολούθησαν οι κύριες λειτουργίες του χάρτη όπως το να παρακολουθείτε ζωντανά η τοποθεσία του χρήστη και να ανανεώνεται για να φαίνονται οι αλλαγές σε

πραγματικό χρόνο. Παράλληλα με τον χάρτη έπρεπε να γίνει και η σελίδα αναζήτησης τελικού προορισμού με την οποία εμφανιζόταν η διαδρομή μεταξύ χρήστη και σημείου αναζήτησης. Συνεπώς αξιοποιήσαμε το autocomplete και καταφέραμε να κάνουμε την εφαρμογή πιο εύχρηστη και πιο φιλική για τον χρήστη.

Έπειτα, συνδέσαμε την εφαρμογή με την firebase. Τώρα είμαστε σε θέση να δημιουργήσουμε την λίστα με την οποία βρίσκουμε τους επιβάτες και τους εμφανίζουμε στον χάρτη του οδηγού. Έπειτα πρέπει να μπορούμε να επιλέξουμε επιβάτη όπως και γίνεται μέσω του χάρτη. Πριν την επιλογή καλούμαστε να δημιουργήσουμε το κατάλληλο σημείο για την παράκαμψη όπως έχουμε αναφέρει σε προηγούμενο κεφάλαιο. Έτσι, δημιουργούμε και τις ειδοποιήσεις που βοηθούν στην επικοινωνία μεταξύ χρηστών κατά την επιλογή.

Είμαστε έτοιμοι τώρα να ελέγξουμε την λειτουργικότητα όλων μαζί. Χρησιμοποιήθηκε το Bluestacks καθώς μπορούσε να μας παρέχει παράλληλη εκτέλεση κινητών με σχετικά μικρό κόστος πόρων του υπολογιστή. Τέλος, παρατηρήσαμε και επιδιορθώσαμε σημαντικά bugs όπως είναι η παύση της ανανέωσης του χάρτη όταν ο χρήστης είναι ακίνητος, η διαγραφή του πελάτη όταν αυτός φτάνει στον προορισμό του και από την οθόνη του οδηγού και απο την βάση Κ.Ο.Κ. .

Κεφάλαιο 6 Πειραματική επικύρωση λειτουργικότητας

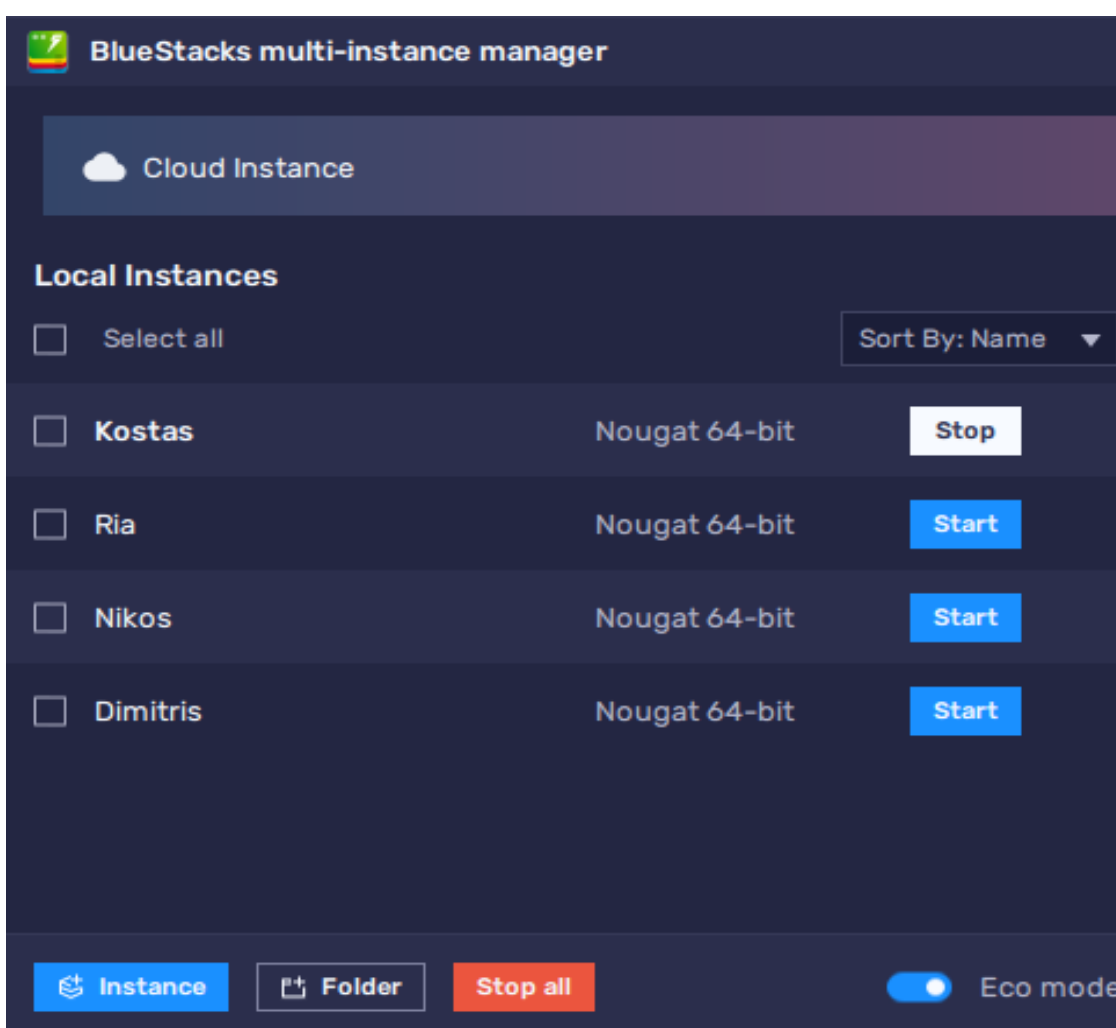
6.1 Τρόπος διεξαγωγής πειραμάτων

Σε αυτό το κεφάλαιο θα δείξουμε το πως καταφέραμε να εκτελέσουμε την εφαρμογή τοπικά στον υπολογιστή χωρίς την χρήση πραγματικού κινητού και χωρίς η εφαρμογή μας να βρίσκεται κτισμένη σε κάποια πλατφόρμα εφαρμογών (Google play, App store). Στην εικόνα 29 βλέπουμε το interface του expo στην γραμμή εντολών. Είναι εργαλείο που μας δίνει την δυνατότητα να εκτελέσουμε την εφαρμογή παρέχοντάς μας ένα URL με το οποίο μπορούμε να την εκτελέσουμε απομακρυσμένα, δηλαδή είτε σε πραγματικό κινητό είτε σε έναν simulator. Είναι απαραίτητο να έχουμε καθ' όλη την διαδικασία το τερματικό ανοικτό καθώς λειτουργεί ως server που κάνει host την εφαρμογή.



Εικόνα 29 Expo CLI

Στην εικόνα 30 βλέπουμε τον πάροχο των προσομοιωτών κινητού που ονομάζεται Bluestacks. Είναι ένα εξαιρετικά γρήγορο και ελαφρύ εργαλείο καθώς παρέχει παράλληλη εκτέλεση προσομοιωτών κινητού με καλές ταχύτητες. Υπάρχει ακόμα η δυνατότητα “οικονομικής εκτέλεσης” δηλαδή κάθε κινητό που θα εκτελείται θα καταναλώνει συγκεκριμένους πόρους από το σύστημά μας με σκοπό της αλληλεπίδραση των κινητών μέσω της εφαρμογής.

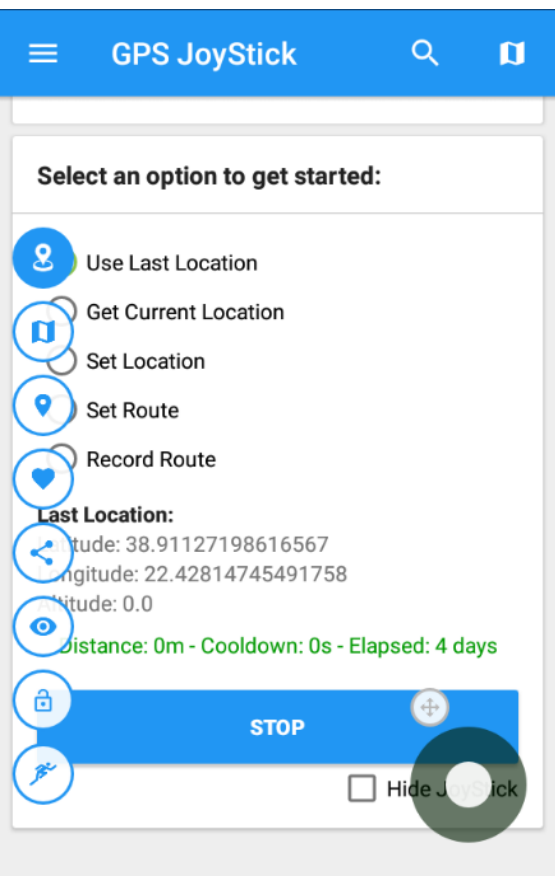
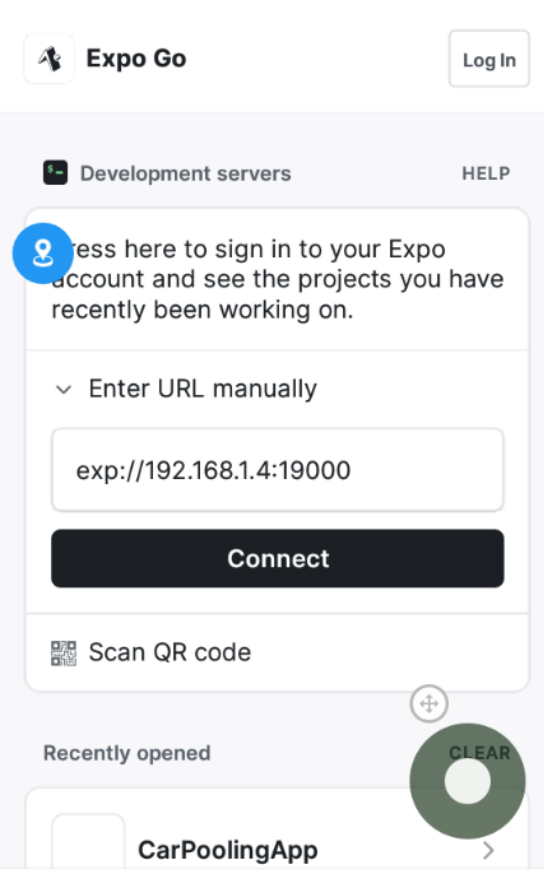


Εικόνα 30 Bluestacks Task Manager

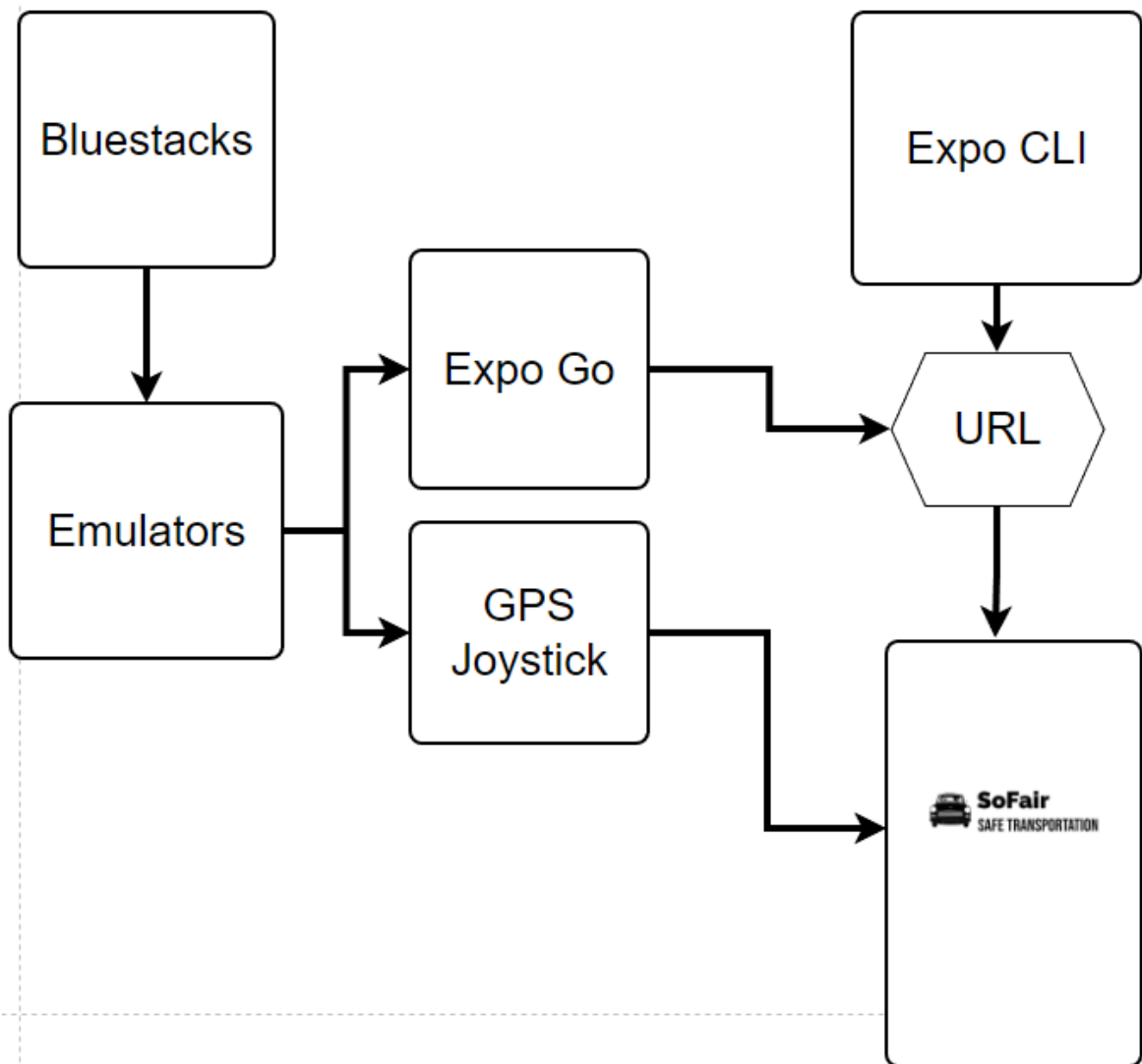
Παρακάτω στην εικόνα 31 εμφανίζεται το εργαλείο που βοήθησε στην προσομοίωση

της κίνησης στον χάρτη. Το GPS, όπως όλοι γνωρίζουμε, κάθε φορά που ανιχνεύει κίνηση παίρνει τις νέες συντεταγμένες και αναγνωρίζει τις αλλαγές στην τοποθεσία. Όμως, όταν τρέχουμε την εφαρμογή σε πειραματικό στάδιο είναι αδύνατη η “κίνηση του υπολογιστή” συνεπώς με την παρακάτω εφαρμογή δίνουμε την αίσθηση στο GPS ότι υπάρχει κίνηση ενώ είμαστε σταθεροί. Αυτό επιτυγχάνεται με έναν μοχλό κίνησης.

Ακόμη στην εικόνα 32 βλέπουμε την εφαρμογή Expo Go που εγκαθίσταται σε όλα τα κινητά η οποία λειτουργεί σαν client και δέχεται το URL από το Expo CLI για να ανοίξει την εφαρμογή απομακρυσμένα.

	
Εικόνα 31 GPS Joystick App	Εικόνα 32 Εφαρμογή Expo Go

Στο παρακάτω σχήμα 3 βλέπουμε την συνολική πορεία που ακολουθείται σε κάθε εκτέλεση και αξιοποιεί τα παραπάνω εργαλεία. Ανοίγοντας τον διαχειριστή προσομοιωτών του Bluestacks εμφανίζονται οι emulators που έχουμε δημιουργήσει. Έτσι ανοίγοντας κάποιους από αυτούς παρουσιάζεται το γραφικό τους περιβάλλον και πράγματι μοιάζουν με πραγματικό κινητό. Καθώς έχουμε εγκαταστήσει τις εφαρμογές Expo Go και GPS Joystick τις ανοίγουμε και κάνουμε τα βήματα που προαναφέραμε. Το URL θα παραχθεί από την γραμμή εντολών και μέσα από τον συνδυασμό Expo Cli και Expo Go η εφαρμογή μας εκτελείται.



Σχήμα 3 Τρόπος διεξαγωγής πειραμάτων

6.2 Αποτελέσματα πειραμάτων

Σε αυτή την ενότητα θα δούμε κάποια από τα πειραματικά σενάρια που μας οδήγησαν στο τελικό αποτέλεσμα. Αρχικά, βασικό ήταν να απορρίπτεται ένας επιβάτης αν η αρχική ή η τελική του θέση διέφερε κατά πολύ από αυτή του οδηγού. Ένα παράδειγμα που εκτελέσαμε ήταν να αναζητήσουμε τελικό προορισμό για τον οδηγό και έναν τελείως αντιδιαμετρικό για τον επιβάτη. Αυτό που περιμέναμε ήταν όταν ο οδηγός γίνει ενεργός να μην υπάρχει στην οθόνη του διαθέσιμος πελάτης. Πράγματι στην οθόνη του οδηγού δεν εμφανίστηκε ο πελάτης αλλά μόλις ξανά αλλάξαμε προορισμό και ορίσαμε έναν νεό σχετικά κοντά στον προορισμό του οδηγού αυτός εμφανίστηκε.

Σημαντική λειτουργία για την εφαρμογή είναι η ζωντανή παρακολούθηση του χρήστη. Ένα παράδειγμα που δοκιμάσαμε ήταν να εμφανίσουμε, σε επίπεδο γραμμής εντολών, τις συντεταγμένες του χρήστη όταν αυτός ήταν ακίνητος. Αυτό που περιμέναμε ήταν οι συντεταγμένες να αλλάζουν λίγο μετά το 5ο ή 6ο δεκαδικό ψηφίο. Πράγματι όταν ο χρήστης έμενε ακίνητος άλλαζαν τα τελευταία ψηφία των συντεταγμένων. Το συγκεκριμένο πείραμα έγινε καθώς υπήρχε πρόβλημα με τον ρυθμό ανανέωσης της σελίδας. Παρατηρήθηκε ότι αν δεν ανανεώνεται η σελίδα όσο ο χρήστης είναι ακίνητος σταματάει να υπάρχει μεγάλος φόρτος μεταφοράς δεδομένων από και προς την βάση.

Για την οργάνωση της λειτουργικότητας έπρεπε όταν επιλέγεται ο επιβάτης από κάποιο οδηγό να αποθηκεύεται στην βάση το όνομα του εκάστοτε χρήστη. Δηλαδή στην βάση του οδηγού να αποθηκεύεται το όνομα του επιλεγμένου πελάτη και αντίστροφα. Αυτό έγινε για να μην συμβεί δύο οδηγοί να επιλέξουν τον ίδιο πελάτη. Και όπως περιμέναμε όταν παρουσιάστηκαν δυο ενεργοί οδηγοί και ένας επιβάτης, οι οθόνες των οδηγών εμφάνισαν και οι δύο τον ίδιο επιβάτη. Όταν όμως ο ένας οδηγός επέλεξε τον επιβάτη και εκείνος αποδέχθηκε το αίτημα αμέσως διαγράφηκε από την οθόνη του άλλου οδηγού. Το συγκεκριμένο πείραμα ήταν σημαντικό καθώς οι επιβάτες είναι δυνατόν να έχουν έναν και μόνο οδηγό και αντίστροφα.

Τέλος, αλλά εξίσου σημαντικό πείραμα, ήταν ότι έπρεπε να διαγράφεται ο επιβάτης από την οθόνη του οδηγού κατά την άφιξή του στον τελικό του προορισμό. Το συγκεκριμένο πείραμα ήταν σημαντικό κυρίως για την απόδοση του κινητού. Καθώς όσο αυξάνονται οι επιβάτες τόσες περισσότερες αιτήσεις για εμφάνιση στον χάρτη έχουμε προς την βάση άρα και μεγαλύτερο φόρτο. Έτσι, η λίστα επιβατών ενημερώνεται συνεχώς και για να αποθηκεύσει κάποιον χρήστη ελέγχει πολλές παραμέτρους μία εκ των οποίων είναι η άφιξη στον τελικό προορισμό.

Κεφάλαιο 7 Συμπεράσματα

7.1 Σύνοψη και συμπεράσματα

Συνοψίζοντας, με την javascript, δηλαδή μια γλώσσα γενικού σκοπού, μπορούμε όχι μόνο να δημιουργήσουμε ιστοσελίδες αλλά σε συνδυασμό με το Node.js να έχουμε ουσιαστικά μια γλώσσα που τρέχει σε τερματικό. Έτσι, με την React Native να είναι χτισμένη πάνω στο συγκεκριμένο εργαλείο μπορούμε να δημιουργήσουμε σταθερές, μεταβλητές και αρκετά κλιμακούμενες εφαρμογές (και με την React ιστοσελίδες) χωρίς να χρειάζεται να γνωρίζουμε γλώσσες προγραμματισμού, όπως java, kotlin, swift κ.ο.κ. , για κάθε λειτουργικό σύστημα κινητών που υπάρχει (android, iOS κ.α.).

Έτσι, για την εφαρμογή επωφεληθήκαμε και χρησιμοποιήσαμε την React Native η οποία παρέχει αρκετές βιβλιοθήκες τόσο για τους χάρτες της εφαρμογής όσο και για τις υπόλοιπες λειτουργίες μέσω των πακέτων του Node. Μερικές ακόμη λειτουργίες είναι η σκιαγράφηση της διαδρομής μεταξύ δύο σταθμών, το Google Autocomplete για την αναζήτηση του τελικού σημείου - προορισμού του εκάστοτε χρήστη (οδηγού - επιβάτη), χωρίς να χρειάζεται να γράψουμε όλο τον προορισμό ακριβώς. Παράλληλα, με τον πυρήνα της εφαρμογής να βασίζεται σε πραγματικού χρόνου αλλαγές στο χάρτη αλλά και στις ειδοποιήσεις δεν θα μπορούσε να λείπει η Firebase (που χρησιμοποιήθηκε η πραγματικού χρόνου NoSQL βάση δεδομένων της) απο την συγκεκριμένη τεχνολογική στοίβα.

Μια εφαρμογή μεταφοράς χρηστών θα μπορούσε να ταιριάζει τέλεια σε μια κοινωνία που οι άνθρωποι μεταφέρονται συνεχώς είτε για εργασία είτε για διασκέδαση. Με τις μικρές αποστάσεις που θα χρειαστεί να διανύσει ο κάθε οδηγός για να μεταφέρει τον πελάτη στον προορισμό του, η μίσθωση θα είναι αρκετά χαμηλή. Συνεπώς ίσως να συμφέρει περισσότερο από ένα ταξί, καθώς ο τρόπος λειτουργίας τους είναι με μια ελάχιστη τιμή και όσο περνάει η ώρα του δρομολογίου να αυξάνει το συνολικό κόστος.

Ακόμη η React Native ή ακόμη το σύνολο εργαλείων που χρησιμοποιήθηκαν στην παρούσα εργασία είναι ικανό να δημιουργήσει εφαρμογές ακόμα μεγαλύτερου φόρτου και να σταθεί αντάξιο δίπλα από πυλώνες στον κόσμο των εφαρμογών και των ιστοσελίδων όπως είναι η PHP. Σίγουρα όμως, επειδή τα εργαλεία είναι σχετικά καινούργια στην προγραμματιστική κοινότητα, χρειάζεται περισσότερη έρευνα και αρκετές δοκιμές από προγραμματιστές είτε frontend είτε backend.

7.2 Μελλοντικές επεκτάσεις

Το συγκεκριμένο θέμα της παρούσας διπλωματικής θα μπορούσε να επεκταθεί σε ένα ανώτερο κοινωνικό δίκτυο. Δηλαδή θα μπορούσε να υπάρχει η δυνατότητα επικοινωνίας μέσω chat, η δυνατότητα προσθήκης στη λίστα φίλων και επικοινωνίας οδηγού - πελάτη αλλά και γενικότερα με οποιονδήποτε τυχαίνει να βρίσκεσαι στο ίδιο αυτοκίνητο. Αυτό απαιτεί επιπλέον αλλαγές τόσο στο front-end δηλαδή στο γραφικό περιβάλλον που θα εμφανίζει την σελίδα του chat όσο και στο backend καθώς η επικοινωνία θα αποθηκεύεται σε μία άλλη βάση δεδομένων που έχει να κάνει με τη σύνδεση των δύο χρηστών.

Έτσι θα μπορεί καθημερινά ο κάθε επιβάτης να έχει τον δικό του προσωπικό οδηγό - φίλο οι οποίοι π.χ μπορούν να πηγαίνουν προς την ίδια κατεύθυνση ή ακόμα και στην ίδια δουλειά και να μην χρειάζεται να περιμένει ο επιβάτης την ενεργοποίηση του οδηγού για να τον βρει ενεργό.

Μία ακόμα μελλοντική επέκταση θα μπορούσαν να είναι τα συστήματα προτάσεων με τεχνικές μηχανικής μάθησης εάν και εφόσον υπήρχε μία μεγάλη βάση δεδομένων από χρήστες. Με αυτόν τον τρόπο θα μπορούσαμε να εκπαιδεύσουμε μοντέλα να λαμβάνουν πληροφορίες από την σελίδα προφίλ του οδηγού και να τον προτείνουν στην σελίδα ειδοποιήσεων του εκάστοτε πελάτη που θέλει να κάνει παρόμοια διαδρομή με αυτόν. Συνεχίζουμε να κρατάμε την έννοια της παράκαμψης μιας και το concept της εφαρμογής έχει χτιστεί γύρω από την εξοικονόμηση τόσο χρόνου όσο και χρήματος.

Ο συνδυασμός των δύο παραπάνω επεκτάσεων θα μπορούσε να οδηγήσει σε μία αρκετά σύγχρονη εφαρμογή και, γιατί όχι, και μια επιχείρηση καθώς στην Ελλάδα στερούμαστε από τέτοιες εφαρμογές και χρησιμοποιούμε ακόμα τους συμβατικούς

τρόπους των ταξί ή των μέσων μεταφοράς. Σε καμία περίπτωση δεν θέλουμε να ακυρώσουμε την σημασία αυτών των μέσων αλλά πιστεύουμε ότι χρειάζεται ένας εναλλακτικός τρόπος μετακίνησης σύμφωνα με τη ραγδαία ανάπτυξη της τεχνολογίας και των εφαρμογών.

Βιβλιογραφία

- [1] React and React Native: A complete hands-on guide to modern web and mobile development with React.js, 3rd Edition by Adam Boduch.
- [2] React Native documentation, <https://reactnative.dev/docs/getting-started>.
- [3] The Road to Firebase: Your journey to master Firebase in JavaScript by Robin Wieruch.
- [4] Firebase main Google page documentation, <https://firebase.google.com/docs>.
- [5] Automating with Node.js by Mr Shaun Michael Stone.
- [6] Node.js version 14.x official documentation, <https://nodejs.org/docs/latest-v14.x/api/>.
- [7] Google Maps: Power Tools for Maximizing the API by Evangelos Petroutsos.
- [8] Coding with Javascript For Dummies, 1st Edition by Chris Minnick.
- [9] Serverless Web Applications with React and Firebase: Develop real-time applications for web and mobile platforms, by Mayur Tanna and Harmeet Singh.
- [10] The package manager NPM documentation, <https://docs.npmjs.com/about-npm>.
- [11] Uber's documentation for developers, <https://developer.uber.com/docs/drivers/introduction>.
- [12] Lyft's documentation for developers, <https://www.lyft.com/developers>.
- [13] Javascript's documentation, <https://devdocs.io/javascript/>.