

+

# **Υλοποίηση διαδικτυακής εφαρμογής για τη μετακίνηση επιβατών με αστικό λεωφορείο**

**ΒΑΣΙΛΕΙΟΣ ΓΕΩΡΓΟΥΛΑΣ**

**Διπλωματική Εργασία**

Επιβλέπων: Νικόλαος Μαμουλής

Ιωάννινα, Φεβρουάριος , 2023



**ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ**

---

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING  
UNIVERSITY OF IOANNINA**

# Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή μου κ. Νικόλαο Μαμουλή για την ευκαιρία που μου έδωσε, την υπομονή του, τις παρατηρήσεις του όπως επίσης και την επίβλεψη του καθ' όλη την διάρκεια της διπλωματικής μου εργασίας από το ξεκίνημα της ως και το τέλος της. Κλείνοντας θα ήθελα να ευχαριστήσω την οικογένεια και τους φίλους μου για την στήριξη που μου παρείχαν όλα αυτά τα χρόνια.

Φεβρουάριος, 2023

Βασίλειος Γεωργούλας

# Περίληψη

Τα τελευταία χρόνια η δημόσια συγκοινωνία με αστικό λεωφορείο έχει γίνει ένας από τους πιο δημοφιλείς τρόπους μετακίνησης για τους κατοίκους μιας πόλης. Πιο συγκεκριμένα, τα αστικά λεωφορεία χρησιμοποιούνται πιο συχνά λόγω της προσιτής τιμής και της μεγάλης κάλυψης που προσφέρουν. Μερικά από τα πλεονεκτήματα που προσφέρει το αστικό λεωφορείο περιλαμβάνουν έναν γρήγορο, ασφαλή και προσιτό τρόπο για τον πληθυσμό να συνδέεται με την εργασία και τις περισσότερες καθημερινές του δραστηριότητες. Ωστόσο, πολλές φορές παρατηρείται ότι ο χρονοπρογραμματισμός των δρομολογίων που προσφέρει το αστικό ΚΤΕΛ παρουσιάζει αποκλίσεις που αφορούν είτε την ώρα άφιξης είτε την ώρα αναχώρησης των λεωφορείων στις στάσεις. Οι κυριότερες αιτίες που προκαλούν αυτά τα προβλήματα σχετίζονται είτε με την μεγάλη ροή επιβατών και την υπερφόρτωση των αστικών, είτε με την κίνηση που υπάρχει στους δρόμους, είτε με τις καιρικές συνθήκες που επικρατούν, είτε με τυχόν ατυχήματα ή έργα στους δρόμους που μπορεί να είναι σε εξέλιξη και επηρεάζουν τις μετακινήσεις.

Στόχος αυτής της διπλωματικής είναι η ανάπτυξη μιας διαδικτυακής εφαρμογής που θα δώσει στους επιβάτες έναν εύκολο τρόπο να ενημερώνονται για τη διαθεσιμότητα των θέσεων, το ημερήσιο πρόγραμμα των γραμμών, τυχόν καθυστερήσεις και τις αντίστοιχες καταστάσεις των λεωφορείων σε πραγματικό χρόνο. Μια ακόμη λειτουργία που προσφέρει η εφαρμογή είναι η δυνατότητα εύρεσης των κατάλληλων δρομολογίων. Οι επιβάτες θα μπορούν να ορίσουν την έναρξη και τον τερματισμό και το σύστημα θα τους προτείνει τα πιο κατάλληλα δρομολόγια, ώστε να φθάσουν άμεσα στον προορισμό τους με βάση την ώρα άφιξης ή αναχώρησης και τη διάρκεια μετακίνησης. Σκοπός της προτεινόμενης εφαρμογής είναι η καλύτερη δυνατή εξυπηρέτηση των επιβατών που μετακινούνται με τα αστικά λεωφορεία καθώς και η εξαγωγή συμπερασμάτων (στατιστικά) που θα βοηθήσουν στην βελτίωση των ήδη υπαρχουσών λειτουργιών.

**Λέξεις κλειδιά:** δημόσια συγκοινωνία, λεωφορείο, στάση, γραμμή, διαδρομή, δρομολόγιο, χάρτης, πρόγραμμα, συγκοινωνία, OpenTripPlanner, web εφαρμογή, Angular, NestJS, PostgreSQL, TypeORM, Redux

# Abstract

In recent years public transportation by city bus has become one of the most popular modes of transportation for the population. More specifically, city buses are used more often due to the comfort as well as the immediacy they offer. Some of the advantages offered by the city bus include a fast, safe, and affordable way for the population to connect to work and most of their daily activities. However, it is often observed that the scheduling of the routes offered by the city bus shows discrepancies regarding either the arrival and departure time of the buses from the arrival and departure points respectively. The main causes that cause these problems are related either to the large flow of passengers and bus overloading or to the traffic that exists on the roads or to the prevailing weather conditions or to any accidents or works on the roads that may be in progress and affect the movement of the vehicles.

The objective of this thesis is to develop a web application that will give passengers an easy way to be informed about seat availability, the daily schedule of the lines, any delays, and the respective bus statuses in real time. Another function offered by the application is the ability to find suitable routes. Passengers will be able to set the start and the destination and the system will provide them with the most suitable routes to reach their destination immediately based on the time of arrival or departure and the duration of the journey. The purpose of the proposed application is to provide the best possible service to passengers traveling by city buses as well as to draw conclusions (statistics) that will help to improve the existing operations.

Keywords: public transport, bus, stop, line, route, itinerary, map, schedule, transit, OpenTripPlanner, web application, Angular, NestJS, PostgreSQL, TypeORM, Redux

## Πίνακας περιεχομένων

<b>Κεφάλαιο 1. Εισαγωγή.....</b>	<b>1</b>
1.1 Αντικείμενο της διπλωματικής.....	1
1.2 Οργάνωση του τόμου.....	3
<b>Κεφάλαιο 2. Σχετικές εφαρμογές.....</b>	<b>4</b>
2.1 Ηλεκτρονικό εισιτήριο ΚΤΕΛ.....	4
2.2 Εφαρμογή moovit.....	6
2.3 Σύστημα tap-on tap-off.....	7
<b>Κεφάλαιο 3. Τεχνολογίες ανάπτυξης.....</b>	<b>9</b>
3.1 Περιηγητής(web browser).....	9
3.2 HTML.....	10
3.3 TypeScript.....	10
3.4 Angular v.15.....	10
3.5 NGRX.....	11
3.6 NestJS.....	12
3.7 PostgreSQL.....	12
3.8 TypeORM.....	12
3.9 LeafletJS.....	12
3.10 PassportJS.....	13
3.11 Socket.io.....	13
3.12 OpenTripPlanner.....	13
<b>Κεφάλαιο 4. Ανάλυση απαιτήσεων (Use Cases).....</b>	<b>15</b>
<b>Κεφάλαιο 5. Σχεδίαση και υλοποίηση.....</b>	<b>23</b>
5.1 Αρχιτεκτονική τριών επιπέδων.....	23
5.2 Υλοποίηση επιπέδου δεδομένων.....	25
5.2.1 Εξόρυξη δεδομένων.....	25
5.2.2 Σχεδίαση και υλοποίηση βάσης δεδομένων.....	26
5.2.3 Φόρτωση της βάσης.....	32
5.3 Υλοποίηση επιπέδου λογικής.....	33

5.3.1 Προετοιμασία Open Trip Planner.....	33
5.3.2 Υλοποίηση διακομιστή(API).....	36
5.3.2.1 Αποθετήριο-υπηρεσία-ελεγκτής.....	36
5.3.2.2 Ενότητες(Modules).....	36
5.4 Υλοποίηση επιπέδου παρουσίασης .....	41
5.4.1 Ενότητες(Modules).....	41
5.4.2 Ασφάλεια.....	46
5.4.3 Επικοινωνία σε πραγματικό χρόνο.....	46
5.4.4 Testing.....	46
<b>Κεφάλαιο 6. Διεπαφή χρήστη .....</b>	<b>48</b>
6.1 Σχεδιαστικά εργαλεία.....	48
6.2 Περιήγηση στις περιπτώσεις χρήσης .....	49
<b>Κεφάλαιο 7. Πειραματική αξιολόγηση .....</b>	<b>58</b>
<b>Κεφάλαιο 8. Επίλογος.....</b>	<b>62</b>
8.1 Σύνοψη και συμπεράσματα.....	62
8.2 Μελλοντικές επεκτάσεις .....	63

# Κεφάλαιο 1. Εισαγωγή

## 1.1. Αντικείμενο της διπλωματικής

Το σύστημα δημοσίων μεταφορών αποτελεί ένα αναπόσπαστο κομμάτι για τη ζωή των κατοίκων στις πόλεις. Ένα από τα πιο αξιοσημείωτα μέσα που προσφέρει είναι τα αστικά λεωφορεία. Τα αστικά λεωφορεία παρέχουν μια σειρά από πλεονεκτήματα, όπως προσιτή τιμή και είναι γενικά φτηνότερα από άλλες μορφές δημόσιας μεταφοράς όπως τρένα ή μετρό, προσβασιμότητα γιατί μπορούν να φτάσουν σε περιοχές που δεν μπορούν άλλες μορφές δημόσιας μεταφοράς, καθιστώντας τα μια βολική επιλογή για άτομα που ζουν σε λιγότερο πυκνοκατοικημένες περιοχές ή σε γειτονιές που δεν εξυπηρετούνται από άλλες μορφές δημόσιας συγκοινωνίας, μειωμένη κυκλοφοριακή συμφόρηση γιατί συμβάλουν στη μείωση της κίνησης στους δρόμους βγάζοντας τα αυτοκίνητα από το δρόμο, καθώς και αυξημένη κινητικότητα γιατί τα λεωφορεία αποτελούν μια βολική επιλογή για άτομα που δεν έχουν πρόσβαση σε αυτοκίνητο, συμπεριλαμβανομένων μαθητών, φοιτητών, ηλικιωμένων ή ατόμων με αναπηρία, να μετακινηθούν στην πόλη.

Παρά τα πολλά πλεονεκτήματα που προσφέρουν, τα αστικά λεωφορεία μπορεί επίσης να παρουσιάσουν ορισμένα προβλήματα. Κάποια από τα πιο σημαντικά περιλαμβάνουν την κυκλοφοριακή συμφόρηση κατά την οποία τα λεωφορεία μπορεί να κολλήσουν στην κυκλοφορία ή σε περιοχές όπου υπάρχουν έργα σε εξέλιξη, γεγονός που μπορεί να οδηγήσει σε καθυστερήσεις και να κάνει τα λεωφορεία λιγότερο αξιόπιστα, καθώς και την περιορισμένη συχνότητα όπου τα λεωφορεία μπορεί να μην κυκλοφορούν τόσο συχνά δυσκολεύοντας τους επιβάτες να προγραμματίσουν τις μετακινήσεις τους, ειδικά όταν πρόκειται για μετακίνηση σε δημοφιλείς προορισμούς. Ένα άλλο πρόβλημα που είναι αρκετά σημαντικό είναι η περιορισμένη χωρητικότητα. Λόγω του μεγέθους, τα λεωφορεία μπορούν να μεταφέρουν μόνο συγκεκριμένο αριθμό ατόμων τη φορά, κάτι που μπορεί να είναι προβληματικό κατά τις ώρες αιχμής και σε περιοχές με υψηλή πυκνότητα πληθυσμού. Αυτό μπορεί να οδηγήσει σε συνωστισμό και τα λεωφορεία ενδέχεται να γεμίσουν υπερβολικά, δημιουργώντας δυσάρεστες και μερικές φορές επισφαλείς συνθήκες για τους επιβάτες. Όλα αυτά τα προβλήματα γίνονται ιδιαίτερα αισθητά υπό αντίξοες καιρικές συνθήκες.

Το βασικό κίνητρο για την παρούσα διπλωματική εργασία είναι τα πολλαπλά προβλήματα που προκύπτουν κατά τη μετακίνηση των φοιτητών του Πανεπιστημίου Ιωαννίνων με το αστικό λεωφορείο. Οι συνεχείς καθυστερήσεις, η υπερφόρτωση των

οχημάτων και η έλλειψη διαθέσιμων θέσεων είναι κοινά προβλήματα που παρουσιάζονται σε καθημερινή βάση και συχνά οδηγούν σε δύσκολες καταστάσεις τις οποίες αντιμετωπίζουν οι φοιτητές. Ως αποτέλεσμα πολλές φορές οι φοιτητές αργούν ή μπορεί να μην φτάσουν στην ώρα τους για τις διαλέξεις ή τα εργαστήρια. Τα προβλήματα αυτά γίνονται ιδιαίτερα αισθητά κατά τη διάρκεια δύσκολων καιρικών συνθηκών. Αυτά τα προβλήματα μετατρέπουν τον προγραμματισμό των μετακινήσεων των φοιτητών σε μια επίπονη διαδικασία.

Στόχος της παρούσας διπλωματικής εργασίας είναι να δώσει τρόπους αντιμετώπισης αυτών των ζητημάτων μέσω μιας διαδικτυακής εφαρμογής. Αφού μελετήθηκαν τα κύρια προβλήματα καθώς και οι ανάγκες των χρηστών του αστικού ΚΤΕΛ, διαπιστώθηκε ότι η βασική λειτουργία που θα πρέπει να έχει η εφαρμογή είναι η δυνατότητα καταγραφής των επιβατών στα λεωφορεία. Το πρακτορείο του αστικού ΚΤΕΛ θα μπορεί να χρησιμοποιήσει αυτή την πληροφορία για να ελέγξει τη ροή των επιβατών και να κάνει τις απαραίτητες αλλαγές. Η συγκεκριμένη εφαρμογή θα στοχεύσει να λύσει αυτό το ζήτημα δίνοντας την δυνατότητα κράτησης θέσης και λόγω του συνωστισμού που επικρατεί στα λεωφορεία, η δυνατότητα αυτή θα αναθέσει προτεραιότητα μεταξύ των επιβατών. Επίσης είναι πολύ σημαντικό να μπορεί ο χρήστης να ενημερώνεται για τους χρόνους άφιξης και αναχώρησης στις στάσεις, τυχόν καθυστερήσεις καθώς και για το ημερήσιο πρόγραμμα. Συνοπτικά η εφαρμογή έχει σκοπό να προσφέρει τις ακόλουθες λειτουργίες:

- Ο επιβάτης θα μπορεί να βρει τα καλύτερα δρομολόγια με βάση τη διαθεσιμότητα θέσεων και τον χρόνο μετακίνησης, ορίζοντας την έναρξη και τον προορισμό, καθώς και την ώρα της επιθυμητής άφιξης ή αναχώρησης. Εφόσον υπάρχουν διαθέσιμες θέσεις στις προτεινόμενες διαδρομές, το σύστημα θα κάνει την κράτηση και θα ενημερώνει τη βάση δεδομένων, ενημερώνοντας παράλληλα όλους τους χρήστες σε πραγματικό χρόνο. Επίσης, θα δημιουργείται ένας μοναδικός κωδικός QR που θα περιγράφει την κράτηση.
- Ο επιβάτης θα μπορεί να αναζητήσει διαδρομές και για κάθε διαδρομή να δει πληροφορίες για όλες τις στάσεις, το ημερήσιο πρόγραμμα και την κατάσταση των λεωφορείων της γραμμής.
- Ο επιβάτης θα μπορεί να βρει στάσεις με βάση το όνομα ή την τοποθεσία και να δει τις στατικές και σε πραγματικό χρόνο πληροφορίες όπως αφίξεις και αναχωρήσεις.
- Ο επιβάτης θα μπορεί να δημιουργήσει έναν λογαριασμό και να αποθηκεύσει συγκεκριμένες πληροφορίες όπως στάσεις και διαδρομές, για γρήγορη εύρεση.



- Η προτεινόμενη εφαρμογή θα δίνει την δυνατότητα οπτικοποίησης των δρομολογίων με την χρήση του χάρτη. Η πληροφορία που θα περιλαμβάνεται είναι οι διαδρομές, οι στάσεις και οι τοποθεσίες των λεωφορείων.

## 1.2. Οργάνωση του τόμου

Η παρούσα διπλωματική εργασία είναι χωρισμένη σε 8 κεφάλαια. Στο δεύτερο κεφάλαιο θα δοθούν κάποια παραδείγματα εφαρμογών και συστημάτων που αφορούν την μετακίνηση με λεωφορείο. Θα γίνει και μια σύγκριση τους με την εφαρμογή που πραγματεύεται η παρούσα διπλωματική.

Στο κεφάλαιο 3 θα δοθεί το τεχνολογικό υπόβαθρο της εφαρμογής με μια εκτενή περιγραφή.

Στο κεφάλαιο 4 θα παρουσιασθούν οι απαιτήσεις του συστήματος σε μορφή περιπτώσεων χρήσης (use cases).

Στο κεφάλαιο 5 περιγράφονται οι λεπτομέρειες υλοποίησης. Θα δοθεί μια ανάλυση της αρχιτεκτονικής της εφαρμογής καθώς και των επιμέρους κομματιών της, συμπεριλαμβανομένων και των βασικών διαγραμμάτων κλάσεων και ενοτήτων (modules).

Στο κεφάλαιο 6 θα δοθεί μια σύντομη περιγραφή των εργαλείων που χρησιμοποιήθηκαν για την σχεδίαση της διεπαφής χρήστη. Θα δοθεί επίσης και μια περιήγηση στην εφαρμογή.

Στη συνέχεια στο κεφάλαιο 7 θα γίνει ένα πείραμα με σκοπό την πειραματική αξιολόγηση της εφαρμογής και την εξαγωγή κάποιων στατιστικών.

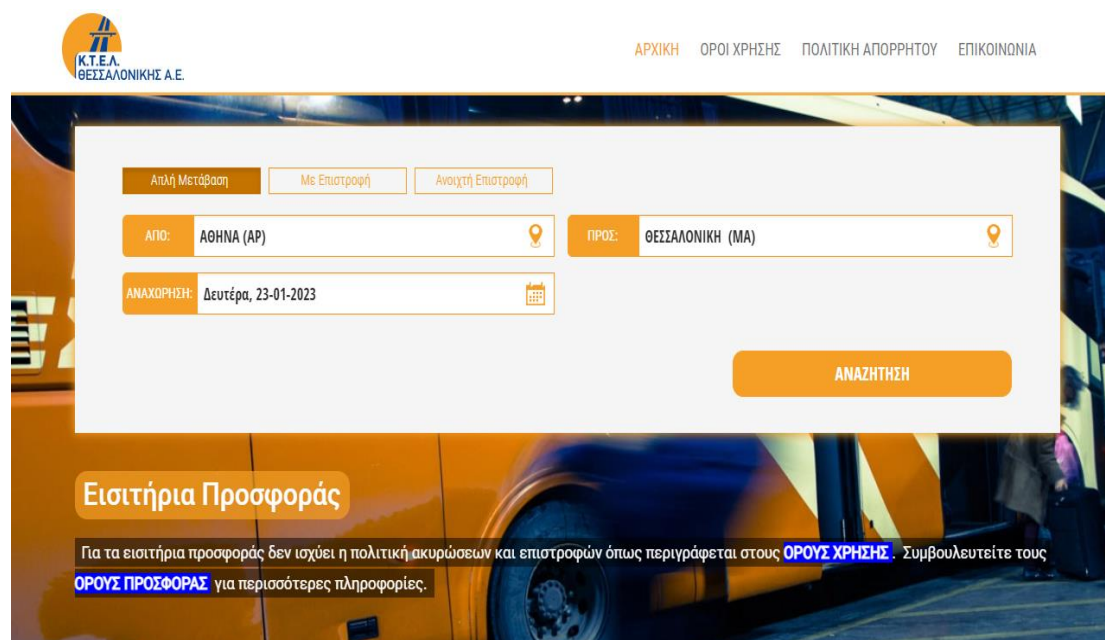
Τέλος, στο κεφάλαιο 8 παραθέτονται κάποιες μελλοντικές επεκτάσεις, τα συμπεράσματα καθώς και ο επίλογος του τόμου.

## Κεφάλαιο 2. Σχετικές εφαρμογές

Ο σκοπός της παρούσας εργασίας είναι η ανάπτυξη μιας εφαρμογής η οποία θα ενημερώνει τους επιβάτες για την κατάσταση των λεωφορείων, τα ωράρια, τις αφίξεις και αναχωρήσεις στις στάσεις και θα δίνει την δυνατότητα κράτησης θέσης. Στο παρελθόν έχουν αναπτυχθεί πολλές εφαρμογές που παρέχουν παρόμοιες λειτουργίες. Ωστόσο, οι περισσότερες από αυτές επικεντρώνονται σε δρομολόγια υπεραστικών λεωφορείων. Στο κεφάλαιο αυτό θα δοθούν κάποια παραδείγματα τέτοιων εφαρμογών.

### 2.1. Ηλεκτρονικό εισιτήριο ΚΤΕΛ([e-ticket](#))

Η πλατφόρμα ηλεκτρονικό εισιτήριο(e-ticket) είναι μια web εφαρμογή η οποία δίνει τη δυνατότητα στους ταξιδιώτες να κλείνουν διαδικτυακά τα εισιτήρια, ένα χρονικό διάστημα πριν την έναρξη του ταξιδιού. Με αυτόν τον τρόπο αν υπάρχουν ελλείψεις θέσεων, η υπηρεσία μπορεί άμεσα να κάνει τις απαραίτητες προσαρμογές όπως την πρόσθεση οχημάτων ή δρομολογίων. Παρακάτω φαίνεται η διεπαφή της εφαρμογής αυτής.



The screenshot shows the KTEL Thessaloniki website interface for e-ticketing. At the top, there is a logo for K.T.E.A. ΘΕΣΣΑΛΟΝΙΚΗΣ Α.Ε. and navigation links: ΑΡΧΙΚΗ, ΟΡΟΙ ΧΡΗΣΗΣ, ΠΟΛΙΤΙΚΗ ΑΠΟΡΡΙΠΤΟΥ, and ΕΠΙΚΟΙΝΩΝΙΑ. The main form has three tabs: Απλή Μετάβαση (selected), Με Επιστροφή, and Ανοιχτή Επιστροφή. Below the tabs, there are input fields for 'ΑΠΟ:' (ATHINA (AP)), 'ΠΡΟΣ:' (ΘΕΣΣΑΛΟΝΙΚΗ (MA)), and 'ΑΝΑΧΟΡΗΣΗ:' (Δευτέρα, 23-01-2023). A large orange button labeled 'ΑΝΑΖΗΤΗΣΗ' is positioned to the right of the date field. Below the form, there is a section titled 'Εισιτήρια Προσφοράς' with a disclaimer: 'Για τα εισιτήρια προσφοράς δεν ισχύει η πολιτική ακυρώσεων και επιστροφών όπως περιγράφεται στους ΟΡΟΥΣ ΧΡΗΣΗΣ. Συμβουλευτείτε τους ΟΡΟΥΣ ΠΡΟΣΦΟΡΑΣ για περισσότερες πληροφορίες.'

Εικόνα 1: Επιλογή διαδρομής στο σύστημα e-ticket, (πηγή: e-ticketing.gr)

1

2

3

4

Ορίστε τον προορισμό σας

Διαλέξτε ώρα δρομολογίου

Επιλέξτε τις θέσεις σας

Ολοκλήρωση αγοράς

### ΔΡΟΜΟΛΟΓΙΑ ΑΝΑΧΩΡΗΣΗΣ

επιλέξτε ένα από τα διαθέσιμα δρομολόγια αναχώρησης

ΑΘΗΝΑ (ΑΡ) - ΘΕΣΣΑΛΟΝΙΚΗ (ΜΑ)		
ΩΡΑ	ΔΙΑΘΕΣΙΜΕΣ ΘΕΣΕΙΣ	ΠΡΟΣΦΟΡΕΣ
⌚ 11:30	41	έκπτωση 10%
⌚ 13:00	0	έκπτωση 10%
⌚ 14:30	24	έκπτωση 10%
⌚ 16:00	0	έκπτωση 10%
⌚ 22:30	39	έκπτωση 10%
⌚ 23:30	35	έκπτωση 10%

ΕΠΟΜΕΝΟ ➡

Εικόνα 2: Επιλογή δρομολογίου στο σύστημα e-ticket, (πηγή: e-ticketing.gr)

### ΛΕΩΦΟΡΕΙΟ ΑΝΑΧΩΡΗΣΗΣ

επιλέξτε τις θέσεις στις οποίες θέλετε να ταξιδέψετε

ΑΘΗΝΑ (ΑΡ) - ΘΕΣΣΑΛΟΝΙΚΗ (ΜΑ)	
Σάββατο, 14-01-2023	16:00
0 ΕΠΙΛΕΓΜΕΝΕΣ ΘΕΣΕΙΣ	

151

147

143

139

135

131

129

125

121

117

113

109

105

152

148

144

140

136

132

130

126

122

118

114

110

106

153

154

149

145

141

137

133

127

123

119

115

111

107

103

155

150

146

142

138

134

128

124

120

116

112

108

104

0

ΔΙΑΘΕΣΙΜΗ

ΕΠΙΛΕΓΜΕΝΗ

ΜΗ ΔΙΑΘΕΣΙΜΗ

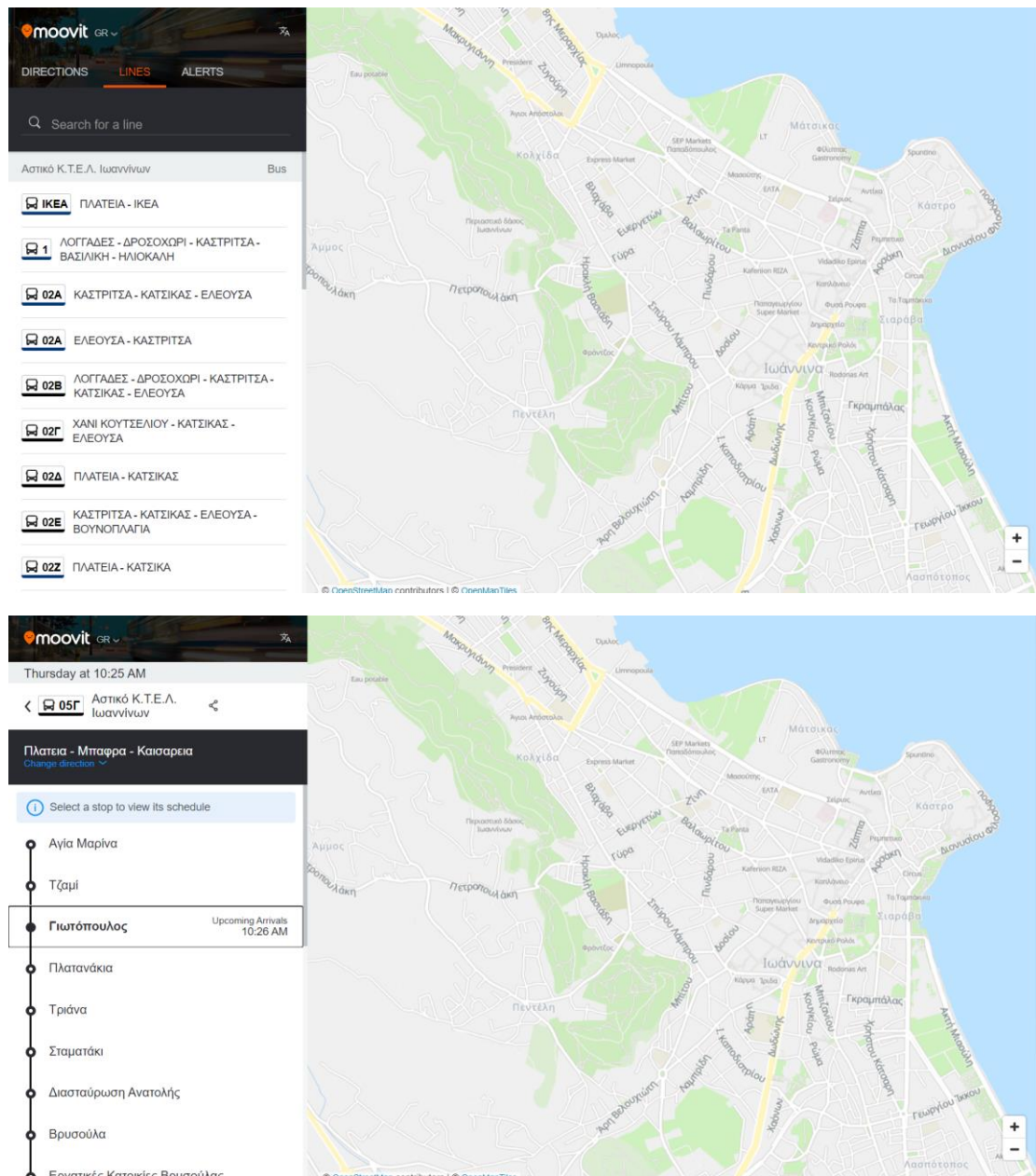
ΟΔΗΓΟΣ

Εικόνα 3: Επιλογή θέσης στο σύστημα e-ticket, (πηγή: e-ticketing.gr)

Με βάση την διεπαφή χρήστη που φαίνεται στις εικόνες 1, 2, και 3 βλέπουμε ότι η εφαρμογή έχει πολλές δυνατότητες. Ο χρήστης αρχικά μπορεί να επιλέξει την αφετηρία και τον προορισμό του ταξιδιού καθώς και την ημερομηνία έναρξης. Έπειτα το σύστημα εμφανίζει όλα τα δρομολόγια της επιλεγμένης ημέρας με τις διαθέσιμες θέσεις κάθε φορά. Αφού γίνει επιλογή του δρομολογίου, δίνεται και η δυνατότητα της επιλογής μιας διαθέσιμης θέσης. Αυτή η λύση περιορίζεται μόνο σε υπεραστικά λεωφορεία όπου εκεί το πλήθος επιβατών καθορίζεται στην στάση έναρξης κατά κανόνα. Επίσης δεν υπάρχει η δυνατότητα οπτικοποίησης των διάδρομων πάνω στον χάρτη ή η ενημέρωση των χρηστών σε πραγματικό χρόνο.

## 2.2. Εφαρμογή moovit([moovit](https://moovitapp.com))

Η εφαρμογή moovit είναι άλλη μια web εφαρμογή η οποία δίνει την δυνατότητα στους χρήστες να ενημερώνονται για τα δρομολόγια των αστικών λεωφορείων. Αυτή η εφαρμογή έχει δεδομένα για πάρα πολλές πόλεις παγκοσμίως και η πόλη των Ιωαννίνων είναι μια από αυτές. Στην εικόνα 4 φαίνεται η διεπαφή χρήστη της εφαρμογής.

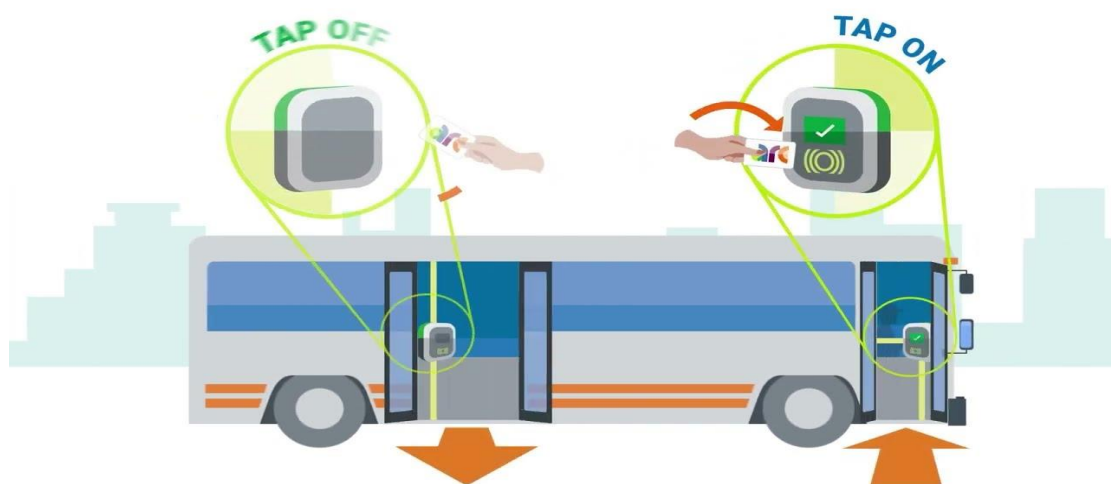


Εικόνα 4: Εφαρμογή moovit (πηγή: [moovitapp.com](https://moovitapp.com))

Στην εικόνα 4 βλέπουμε ότι η εφαρμογή παρέχει την δυνατότητα εύρεσης γραμμών και για κάθε γραμμή εμφανίζει και μια λίστα με τις στάσεις. Για κάθε στάση που επιλέγει ο χρήστης το σύστημα εμφανίζει την επόμενη άφιξη με βάση το στατικό πρόγραμμα της διαδρομής. Επίσης υπάρχει και η δυνατότητα εύρεσης των κατάλληλων δρομολογίων δίνοντας την αρχική και τελική στάση καθώς και την ώρα μετακίνησης. Η εφαρμογή αυτή δεν παρέχει πληροφορίες σε πραγματικό χρόνο όπως αφίξεις, αναχωρήσεις, θέσεις λεωφορείων και την δυνατότητα κράτησης θέσης.

### 2.3. Σύστημα tap-on tap-off.([tap on-tap off](#))

Αξιοσημείωτο είναι το σύστημα που έχει εγκατασταθεί στα αστικά λεωφορεία της Αγγλίας, γνωστό και ως σύστημα tap-on tap-off. Οι επιβάτες πρέπει να σαρώσουν το τηλέφωνό τους μία φορά κατά την επιβίβαση και μία κατά την αποβίβαση. Στη συνέχεια, το σύστημα κατά την αποβίβαση υπολογίζει το κόστος με βάση τις δύο στάσεις. Με αυτό τον τρόπο η υπηρεσία είναι σε θέση να δημιουργήσει ένα ιστορικό και να μελετήσει την ροή των επιβατών. Έτσι είναι δυνατό να γίνουν οι απαραίτητες αλλαγές στα δρομολόγια όπως εισαγωγή νέων δρομολογίων ή οχημάτων και να εξυπηρετηθούν οι ώρες αιχμής ή οι δημοφιλείς προορισμοί πιο αποτελεσματικά. Σε αυτήν την περίπτωση ο επιβάτης δεν χρειάζεται να καθορίσει την εκκίνηση και τον προορισμό. Ένα μειονέκτημα αυτής της προσέγγισης είναι ότι δεν είναι εφικτή η κράτηση θέσης πριν την ανάβαση στο λεωφορείο. Επίσης επειδή είναι ένα σύστημα πληρωμής δεν προσφέρει καμία άλλη δυνατότητα όπως πληροφορίες διάδρομων.



Εικόνα 5: Σύστημα tap-on tap-off, (πηγή: web)

Όπως φαίνεται και στην εικόνα 5 οι επιβάτες ανεβαίνουν από την μπροστινή πόρτα σκανάροντας την κάρτα τους ή το κινητό. Έπειτα κατεβαίνουν από την πίσω επαναλαμβάνοντας τη διαδικασία. Το σύστημα αυτό έχει εφαρμοστεί και σε άλλες χώρες όπως για παράδειγμα στο Hong Kong. Οι επιβάτες μπορούν να προμηθευτούν την ειδική κάρτα octopus την οποία μπορούν να σκανάρουν κατά την είσοδο στο λεωφορείο. Η κάρτα αυτή χρησιμοποιείται και από αλλά μέσα δημόσιας συγκοινωνίας όπως στο μετρό, όπου ο χρήστης θα πρέπει να σκανάρει την κάρτα και κατά την αποβίβαση, όπως και στο σύστημα tap-on tap-off.

Γενικά το πρόβλημα κράτησης θέσης στα αστικά λεωφορεία παρουσιάζει αυξημένες απαιτήσεις. Αυτό γιατί για κάθε δρομολόγιο μπορούμε να έχουμε πολλές υποδιαδρομές. Μια υποδιαδρομή μπορεί να θεωρηθεί ως μια υπακολουθία διαδοχικών στάσεων, πράγμα που σημαίνει ότι το πλήθος των επιβατών είναι μεταβαλλόμενο από στάση σε στάση. Μια κακή και πρόχειρη σχεδίαση θα μπορούσε να επιβαρύνει εκθετικά το σύστημα ή να οδηγήσει σε εσφαλμένη προτεραιότητα, λόγω του μεγάλου όγκου ενεργειών που θα πρέπει να γίνουν κατά τη διάρκεια κράτησης μιας θέσης. Για παράδειγμα θα μπορούσε μια υλοποίηση να είναι παρόμοια με τα υπεραστικά λεωφορεία όπου εκεί ο αριθμός των επιβατών καθορίζεται στην στάση αναχώρησης κατά κανόνα. Θα πρέπει να δοθεί προσοχή στον τρόπο με τον οποίο γίνεται ο έλεγχος διαθεσιμότητας θέσεων και στον τρόπο αποθήκευσης και ανάκτησης των δεδομένων κάθε φορά. Αυτά είναι κάποια από τα προβλήματα που στοχεύει να λύσει αυτή η διπλωματική εργασία, συνδυάζοντας τα 3 συστήματα που προαναφέρθηκαν.



## Κεφάλαιο 3. Τεχνολογίες ανάπτυξης

Η προτεινόμενη εφαρμογή που πραγματεύεται η παρούσα διπλωματική εργασία θα ακολουθήσει μια διαδικτυακή προσέγγιση. Κάποια από τα κύρια πλεονεκτήματα των διαδικτυακών εφαρμογών είναι τα ακόλουθα:

- Επιτρέπουν σε περισσότερους από έναν χρήστες να έχουν πρόσβαση στην ίδια έκδοση της εφαρμογής.
- Δεν απαιτούν εγκατάσταση.
- Είναι προσβάσιμες μέσω μιας ποικιλίας συσκευών, συμπεριλαμβανομένων των επιτραπέζιων υπολογιστών και φορητών συσκευών.
- Είναι ευκολότερες στη διαχείριση, ανάπτυξη και συντήρηση.
- Απαιτούν μόνο σύνδεση στο διαδίκτυο για να λειτουργήσουν.
- Υπάρχει ένα μεγάλο πλήθος εργαλείων και τεχνολογιών για την ανάπτυξη τους.

Τα τελευταία χρόνια οι διαδικτυακές εφαρμογές έχουν γίνει ιδιαίτερα δημοφιλείς και υπάρχουν πολλές προσεγγίσεις και τεχνολογίες για την ανάπτυξή τους. Μετά από μια έρευνα των διαθέσιμων τεχνολογιών και σε συνδυασμό με τις απαιτήσεις της εφαρμογής, έγινε επιλογή των παρακάτω τεχνολογιών και εργαλείων για την υλοποίηση της.

### 3.1. Περιηγητής(web browser)

Το πρόγραμμα περιήγησης ιστού είναι το κέλυφος που φιλοξενεί μια εφαρμογή ιστού και μπορεί να θεωρηθεί ως ένας πελάτης. Ένας πελάτης είναι ένα πρόγραμμα που εστιάζει στην υποβολή αιτημάτων σε έναν διακομιστή, στην εμφάνιση πληροφοριών και επιτρέπει στον χρήστη να αλληλεπιδρά με τις εμφανιζόμενες πληροφορίες. Η επικοινωνία μεταξύ του διακομιστή και ενός πελάτη δημιουργείται χρησιμοποιώντας το πρωτόκολλο HTTP (Hypertext Transfer Protocol) το οποίο βασίζεται στο TCP/IP. Ο περιηγητής ιστού χρησιμεύει επίσης ως εργαλείο ανάπτυξης web εφαρμογών, καθώς προσφέρει πολλές δυνατότητες που μπορούν να βοηθήσουν τη διαδικασία ανάπτυξης.

## 3.2. HTML

Η HTML (Hypertext Markup Language) είναι μια τυπική γλώσσα σήμανσης για τη δημιουργία ιστοσελίδων και εφαρμογών ιστού. Με την HTML θα γραφτεί η διεπαφή χρήστη.

## 3.3. TypeScript

Η TypeScript είναι μια γλώσσα προγραμματισμού που αποτελεί υπερσύνολο της JavaScript και προσθέτει υποστήριξη για έλεγχο τύπων. Αναπτύχθηκε από τη Microsoft και χρησιμοποιείται συνήθως για τη δημιουργία μεγάλων εφαρμογών σε επίπεδο επιχείρησης. Χρησιμοποιώντας τη TypeScript, ο εντοπισμός σφαλμάτων που αφορούν τους τύπους των μεταβλητών συμβαίνει κατά τη μεταγλώττιση. Εισάγει χαρακτηριστικά όπως διεπαφές, αφαιρέσεις, κλάσεις και modules, που μπορούν να βοηθήσουν στην οργάνωση του κώδικα. Ο κώδικας TypeScript πρέπει να μεταφερθεί σε JavaScript για να μπορέσει να εκτελεστεί σε πρόγραμμα περιήγησης ιστού.

## 3.4. Angular v.15

Η Angular είναι ένα πλαίσιο λογισμικού JavaScript για τη δημιουργία εφαρμογών ιστού και κινητών με χρήση TypeScript. Αναπτύχθηκε και συντηρείται από την Google, και εστιάζει κυρίως στον προγραμματισμό με modules, τα οποία είναι σύνολα από μικρά κομμάτια μιας εφαρμογής που εστιάζουν σε συγκεκριμένες λειτουργίες. Το πιο σημαντικό κομμάτι της Angular είναι ο διαχωρισμός του κώδικα σε components. Τα components περιγράφουν και από ένα κομμάτι της διεπαφής χρήστη και μπορούν να φωλιαστούν το ένα μέσα στο άλλο δημιουργώντας έτσι πολύπλοκες διεπαφές χρήστη. Η πλοήγηση μεταξύ των όψεων επιτυγχάνεται με την δυναμική αντικατάσταση των components, δίνοντας έτσι την δυνατότητα της πλοήγησης χωρίς την φόρτωση της σελίδας ή ακόμα και την αλλαγή των επιμέρους κομματιών μιας όψης. Κάθε module έχει ένα σύνολο από components και ένα αρχείο TypeScript το οποίο ορίζει τις διαδρομές(URL) και αντιστοιχίζει κάθε component ή module σε μια διαδρομή. Πρακτικά ένα component αποτελείται από:

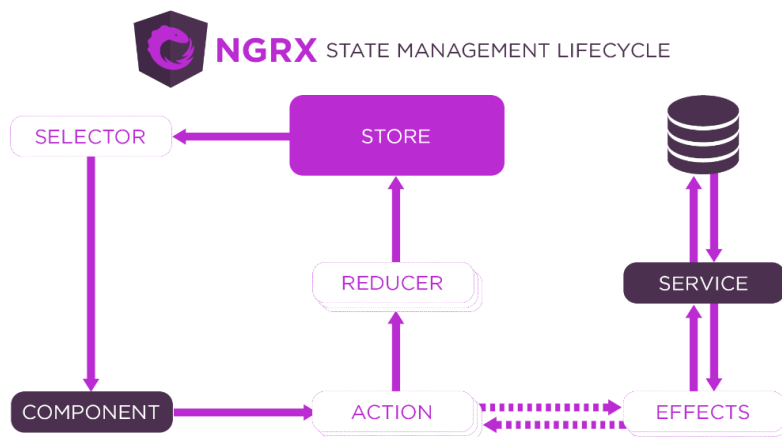
- Ένα αρχείο HTML το οποίο περιγράφει ένα κομμάτι της διεπαφής χρήστη.
- Ένα αρχείο CSS το οποίο δίνει το στυλ στο αρχείο HTML.
- Ένα αρχείο TypeScript το οποίο αποτελεί τον ελεγκτή.

Η Angular θα χρησιμοποιηθεί για την υλοποίηση της διεπαφής χρήστη.



### 3.5. NGRX

Ένα από τα πιο σύνθετα κομμάτια στην υλοποίηση μιας εφαρμογής είναι η διαχείριση κατάστασης καθώς και η επικοινωνία μεταξύ των διάφορων κομματιών της. Η διαχείριση κατάστασης αφορά την αποθήκευση και ανάκτηση δεδομένων, τη διαχείριση της συμπεριφοράς της εφαρμογής και την απόκριση στις αλληλεπιδράσεις των χρηστών. Η βιβλιοθήκη NGRX παρέχει εργαλεία για τη διαχείριση της κατάστασης βασιζόμενη στις αρχές του Redux. Η ιδέα είναι ότι ολόκληρη η κατάσταση της εφαρμογής αποθηκεύεται σε ένα ενιαίο, αμετάβλητο "κατάστημα". Όταν ένας χρήστης αλληλεπιδρά με την εφαρμογή, αποστέλλεται μια "ενέργεια" που περιγράφει την αλλαγή. Στη συνέχεια, χρησιμοποιείται μια συνάρτηση "reducer" για την ενημέρωση της κατάστασης με βάση την ενέργεια. Η ενημέρωση γίνεται με την δημιουργία ενός καινούριου αντικειμένου κατάστασης κάθε φορά. Τα διάφορα κομμάτια της εφαρμογής (observers) παρατηρούν την κατάσταση, και μόλις αλλάξει εμφανίζουν τα νέα δεδομένα. Στην εικόνα 6 φαίνεται ένα διάγραμμα ροής που περιγράφει την υλοποίηση του σχεδιαστικού μοτίβου Redux μέσω αυτής της βιβλιοθήκης.



Εικόνα 6. Σχεδιαστικό μοτίβο Redux(πηγή: [ngrx.io](https://ngrx.io))

Στην εικόνα 6 βλέπουμε τα κομμάτια που απαρτίζουν αυτό το σχεδιαστικό μοτίβο. Παρατηρούμε ότι έχουμε μια υπηρεσία που επικοινωνεί με τον διακομιστή και επιστρέφει μια ροή δεδομένων(observable). Έπειτα έχουμε μια κλάση η οποία ορίζει τα effects. Ένα effect παρατηρεί και δημιουργεί νέες ροές δεδομένων τις οποίες προωθεί σε συνδρομητές(observers) μέσω των actions. Τα actions περιγράφουν μοναδικές ενέργειες που συμβαίνουν και μετατρέπονται σε ροές δεδομένων(observables) με χρήση συναρτήσεων. Στη συνέχεια έχουμε τον reducer ο οποίος 'ακούει' τα actions και με βάση αυτά τροποποιεί την καθολική κατάσταση(Store). Το αντικείμενο Store είναι μια singleton κλάση η οποία διατηρεί τα δεδομένα καθώς και τις μεθόδους για την

τροποποίηση τους. Τέλος έχουμε και τους selectors οι οποίοι επιλέγουν τα δεδομένα της κατάστασης και τα προωθούν στα μέρη της εφαρμογής. Ένας selector ενεργοποιείται κάθε φορά που αλλάζει η κατάσταση. Αυτό το σχεδιαστικό μοτίβο θα αποτελέσει τη βάση για τη σχεδίαση της διεπαφής χρήστη.

### 3.6. NestJS

Εμπνευσμένο από την Angular, το NestJS είναι ένα πλαίσιο λογισμικού για τη δημιουργία εφαρμογών από την πλευρά του διακομιστή σε TypeScript και Node.js και έχει σχεδιαστεί για επεκτασιμότητα και διαχωρισμό του κώδικα σε modules. Παρόμοια με την Angular, χρησιμοποιεί modules για την οργάνωση του κώδικα σε επαναχρησιμοποιούμενα στοιχεία και ένα σύστημα dependency injection για τη διαχείριση των σχέσεων μεταξύ τμημάτων της εφαρμογής. Θα χρησιμοποιήσουμε το nestJS για να υλοποιήσουμε τον διακομιστή (rest API) της εφαρμογής.

### 3.7. PostgreSQL

Η PostgreSQL, επίσης γνωστή ως Postgres, είναι ένα σύστημα διαχείρισης αντικειμενικών-σχεσιακών βάσεων δεδομένων ανοιχτού κώδικα (ORDBMS) που έχει σχεδιαστεί για επεκτασιμότητα και αποτελεσματικότητα. Θα χρησιμοποιήσουμε την Postgres για την υλοποίηση της βάσης δεδομένων.

### 3.8. TypeORM

Το TypeORM είναι ένας αντικειμενοσχεσιακός χάρτης (ORM) που λειτουργεί με TypeScript και JavaScript και μπορεί να χρησιμοποιηθεί με διάφορες βάσεις δεδομένων όπως η PostgreSQL, επιτρέποντας τον ορισμό της δομής της βάσης χρησιμοποιώντας κλάσεις, και παρέχει ένα API για την εκτέλεση κοινών λειτουργιών όπως βασικές ερωτήσεις SQL. Η βιβλιοθήκη αυτή θα μας βοηθήσει στο στήσιμο της βάσης δεδομένων.

### 3.9. LeafletJs

Η Leaflet είναι μια βιβλιοθήκη JavaScript ανοιχτού κώδικα που χρησιμοποιείται για τη δημιουργία διαδραστικών χαρτών σε εφαρμογές ιστού. Είναι ελαφριά και εύκολη στη χρήση και θα χρησιμοποιηθεί για την οπτικοποίηση των δεδομένων.

## 3.10. PassportJs

Το Passport είναι μια βιβλιοθήκη JavaScript για το Node.js που παρέχει έναν απλό και ευέλικτο τρόπο για τον έλεγχο ταυτότητας των χρηστών σε μια εφαρμογή ιστού. Θα χρησιμοποιήσουμε αυτήν την βιβλιοθήκη για να εισάγουμε ασφάλεια στην εφαρμογή.

## 3.11. Socket.io

Το Socket.io είναι μια βιβλιοθήκη που στοχεύει στην απλοποίηση της χρήσης και της ενοποίησης των υποδοχών ιστού (web sockets) και χρησιμοποιείται για επικοινωνία σε πραγματικό χρόνο, από άκρο σε άκρο. Με αυτήν την βιβλιοθήκη θα ικανοποιήσουμε την ανάγκη επικοινωνίας σε πραγματικό χρόνο.

## 3.12. Open Trip Planner

Ο Open Trip Planner (OTP) είναι ένα πολυτροπικό λογισμικό προγραμματισμού ταξιδιού ανοιχτού κώδικα που έχει σχεδιαστεί για να βοηθά τους χρήστες να βρίσκουν την καλύτερη διαδρομή για τη μετακίνηση τους χρησιμοποιώντας μια ποικιλία τρόπων μεταφοράς, όπως δημόσια συγκοινωνία, περπάτημα, ποδηλασία και είναι γραμμένο σε Java. Εκθέτει ένα API endpoint μέσα από το οποίο λαμβάνει και στέλνει δεδομένα. Στον πυρήνα του το OTP δέχεται ένα γράφημα που αντιπροσωπεύει ένα σταθμισμένο και κατευθυνόμενο οδικό δίκτυο και τα δεδομένα συγκοινωνίας από μια υπηρεσία δημόσιων μεταφορών. Στη συνέχεια αντιστοιχίζει τα δεδομένα στο γράφημα. Χρησιμοποιώντας το εκτεταμένο γράφημα, ο OTP χρησιμοποιεί προηγμένους αλγόριθμους γραφημάτων για τον υπολογισμό των βέλτιστων δρομολογίων που παρέχονται με παραμέτρους όπως η έναρξη και το τέλος της διαδρομής, οι χρόνοι άφιξης και οι τρόποι μετακίνησης. Μερικοί από αυτούς τους αλγόριθμους είναι οι εξής: Multi-objective Pareto Shortest Paths και Multi-criteria Range Raptor ο οποίος υπολογίζει τις συντομότερες διαδρομές χρησιμοποιώντας τους χρόνους άφιξης σε ενδιάμεσα σημεία.

Ο OTP δέχεται δύο είδη ροών δεδομένων συγκοινωνίας. Τα αρχεία GTFS και GTFS πραγματικού χρόνου. Τα αρχεία GTFS (General Transit Feed Specification) είναι μια μορφή για την αναπαράσταση πληροφοριών δημόσιας συγκοινωνίας, που ορίζεται από την Google. Χρησιμοποιούνται για να περιγράψουν τα χρονοδιαγράμματα, τις

διαδρομές και τις τοποθεσίες στάσεων ενός συστήματος δημόσιας συγκοινωνίας, όπως ένα σύστημα λεωφορείων ή τρένων, και αποθηκεύονται σε αρχεία κειμένου. Τα GTFS πραγματικού χρόνου μπορούν να χωριστούν σε 3 κατηγορίες: τοποθεσίες λεωφορείων, ενημερώσεις δρομολογίων και ειδοποιήσεις. Ο OTP μπορεί να χρησιμοποιήσει αυτές τις ενημερώσεις για να παρέχει πιο ακριβείς πληροφορίες, όπως ώρες άφιξης και αναχώρησης, την κατάσταση πληρότητας των οχημάτων, πρόσθετες ή ακυρωμένες γραμμές, καθυστερήσεις και αλλαγές στο πρόγραμμα. Σε αντίθεση με τα στατικά GTFS, ο OTP δέχεται τις ενημερώσεις πραγματικού χρόνου ως buffer πρωτοκόλλου (protobuf). Τα πρωτόκολλα buffer είναι μικρά αρχεία που χρησιμοποιούνται για την αποτελεσματική αποθήκευση και μετάδοση δεδομένων μέσω δικτύων και έχουν παρόμοια σύνταξη με τις δομές στη C. Τα Protobufs βασίζονται σε ένα απλό μοντέλο δεδομένων που αποτελείται από μηνύματα με πεδία. Κάθε πεδίο έχει ένα όνομα, έναν τύπο δεδομένων και έναν μοναδικό αριθμό, ο οποίος χρησιμοποιείται για την αναγνώριση του πεδίου κατά τη διαδικασία σειριοποίησης. Το λογισμικό αυτό θα χρησιμοποιηθεί για την εύρεση βέλτιστων δρομολογίων και για την εκτίμηση του χρόνου άφιξης των λεωφορείων στις στάσεις σε πραγματικό χρόνο με βάση τις τοποθεσίες.

## Κεφάλαιο 4. Ανάλυση απαιτήσεων

Στην ενότητα αυτή παρουσιάζονται οι απαιτήσεις της εφαρμογής από την πλευρά του χρήστη σε μορφή use case. Τα use cases περιγράφουν βήμα - βήμα τις περιπτώσεις χρήσης ενός συστήματος. Στην περίπτωση αυτής της εφαρμογής υπάρχουν δύο είδη περιπτώσεων χρήσης. Οι περιπτώσεις που απαιτούν σύνδεση του χρήστη(login) και οι περιπτώσεις που δεν απαιτούν σύνδεση. Αυτό έγινε για ευκολία του χρήστη. Συγκεκριμένα για τις περιπτώσεις χρήσης 3, 4 δεν υπάρχει η απαίτηση σύνδεσης. Παρακάτω παρουσιάζονται τα κύρια use cases της προτεινόμενης εφαρμογής, έτσι όπως προέκυψαν κατά τη σχεδίαση.

### **Περίπτωση χρήσης: Εγγραφή στο σύστημα(UC1)**

#### **Περιγραφή:**

Για αυτήν την περίπτωση χρήσης, ο χρήστης θα πρέπει να έχει τη δυνατότητα να δημιουργήσει λογαριασμό στην εφαρμογή παρέχοντας τα διαπιστευτήριά του. Ο χρήστης πρέπει να παρέχει τα ακόλουθα υποχρεωτικά πεδία:

- Email
- Κωδικό πρόσβασης
- Όνομα χρήστη

#### **Ηθοποιοί:**

Χρήστης

#### **Προϋποθέσεις:**

- Ο χρήστης πρέπει να παρέχει ένα μοναδικό email.
- Ο χρήστης πρέπει να είναι συνδεδεμένος στο διαδίκτυο.
- Το όνομα χρήστη πρέπει να αποτελείται από τουλάχιστον 3 χαρακτήρες.
- Το email πρέπει να έχει τη σωστή μορφή.
- Ο κωδικός πρόσβασης πρέπει να αποτελείται από τουλάχιστον 5 χαρακτήρες.

#### **Κύρια ροή γεγονότων:**

1. Η περίπτωση χρήσης ξεκινά όταν ο χρήστης θέλει να δημιουργήσει έναν λογαριασμό.
2. Ο χρήστης κάνει κλικ στο εικονίδιο που ανοίγει το δευτερεύον μενού πλοήγησης και στη συνέχεια κάνει κλικ στο κουμπί εγγραφής. Έπειτα το σύστημα

εμφανίζει τη σελίδα εγγραφής.

3. Ο χρήστης συμπληρώνει τα απαιτούμενα πεδία.
4. Καθώς ο χρήστης πληκτρολογεί τα πεδία, το σύστημα επικυρώνει τις εισόδους και εμφανίζει μηνύματα ελέγχου.
5. Όταν ο χρήστης συμπληρώσει όλα τα πεδία σωστά και κάνει κλικ στο κουμπί 'Εγγραφή', το σύστημα εκτελεί μια επικύρωση από την πλευρά του διακομιστή και ελέγχει τη βάση δεδομένων εάν το email υπάρχει. Έπειτα συνδέει τον χρήστη και τον μεταφέρει στη αρχική σελίδα σε περίπτωση επιτυχίας.
6. Όταν ο χρήστης είναι συνδεδεμένος, το μενού πλοήγησης εμφανίζει το όνομα χρήστη.

#### **Εναλλακτική ροή:**

1. Ο χρήστης έχει λογαριασμό.
2. Ο χρήστης μεταφέρεται στη σελίδα σύνδεσης μέσω του ρυθμιστικού πλοήγησης.

### **Περίπτωση χρήσης: Είσοδος στο σύστημα (UC2)**

#### **Περιγραφή:**

Για αυτή την περίπτωση χρήσης, ο χρήστης πρέπει να μπορεί να συνδεθεί στην εφαρμογή παρέχοντας τις απαραίτητες πληροφορίες. Ο χρήστης πρέπει να παρέχει τα ακόλουθα υποχρεωτικά πεδία:

- Email
- Κωδικό πρόσβασης

#### **Ηθοποιοί:**

Χρήστης

#### **Προϋποθέσεις:**

- Ο χρήστης πρέπει να έχει αποσυνδεθεί.
- Ο χρήστης πρέπει να έχει λογαριασμό.
- Ο χρήστης πρέπει να είναι συνδεδεμένος στο διαδίκτυο.
- Το email πρέπει να έχει τη σωστή μορφή.

#### **Κύρια ροή γεγονότων:**

1. Η περίπτωση χρήσης ξεκινά όταν ο χρήστης θέλει να συνδεθεί.

2. Ο χρήστης κάνει κλικ στο εικονίδιο που ανοίγει το δευτερεύον μενού πλοήγησης και στη συνέχεια κάνει κλικ στο κουμπί σύνδεσης. Στη συνέχεια, το σύστημα μεταφέρεται στη σελίδα σύνδεσης.
3. Ο χρήστης συμπληρώνει τα απαιτούμενα πεδία.
4. Καθώς ο χρήστης πληκτρολογεί τα πεδία, το σύστημα επικυρώνει τις εισόδους και εμφανίζει μηνύματα.
5. Όταν ο χρήστης συμπληρώσει σωστά όλα τα πεδία και κάνει κλικ στο κουμπί 'Σύνδεση', το σύστημα εκτελεί μια επικύρωση από την πλευρά του διακομιστή και ελέγχει στη βάση δεδομένων εάν υπάρχει ο χρήστης.
  - 5.1. Εάν υπάρχει σφάλμα, το σύστημα εμφανίζει αντίστοιχο μήνυμα.
  - 5.2. Αν περάσουν όλοι οι έλεγχοι, το σύστημα συνδέει τον χρήστη και μεταφέρεται στην αρχική σελίδα.

#### **Εναλλακτική ροή:**

1. Ο χρήστης δεν έχει λογαριασμό.
2. Ο χρήστης μεταφέρεται στη σελίδα εγγραφής μέσω του μενού.

#### **Περίπτωση χρήσης: Εμφάνιση πληροφοριών διαδρομής (UC3)**

##### **Περιγραφή:**

Για αυτήν την περίπτωση χρήσης ο χρήστης πρέπει να μπορεί να αναζητήσει τις διαθέσιμες διαδρομές ανά όνομα ή κωδικό και να δει τις σχετικές πληροφορίες που αποτελούνται από το ημερήσιο πρόγραμμα, τις αφίξεις στάσεων και την κατάσταση των λεωφορείων σε πραγματικό χρόνο.

##### **Ηθοποιοί:**

Χρήστης

##### **Προϋποθέσεις:**

- Ο χρήστης πρέπει να είναι συνδεδεμένος στο διαδίκτυο

##### **Κύρια ροή γεγονότων:**

1. Η περίπτωση χρήσης ξεκινά όταν ο χρήστης θέλει να αναζητήσει πληροφορίες για μια συγκεκριμένη διαδρομή.
2. Ο χρήστης κάνει κλικ στο κουμπί 'Γραμμές' και το σύστημα μεταφέρεται στη λίστα βασικών γραμμών.

3. Ο χρήστης επιλέγει μια διαδρομή μιας βασικής γραμμής και το σύστημα εμφανίζει τις λεπτομέρειες.
4. Για μια επιλεγμένη διαδρομή, το σύστημα εμφανίζει τις στάσεις και την επιλογή των ενεργών λεωφορείων.
  - 4.1. Επιλέγοντας την επιλογή 'Πρόγραμμα', ο χρήστης μπορεί να δει την ταξινομημένη λίστα των στάσεων και να επιλέξει μια ημέρα για να δει τις εκτιμήσεις αναχώρησης για κάθε στάση για τη συγκεκριμένη ημέρα.
  - 4.2. Επιλέγοντας την επιλογή 'Λεωφορεία', ο χρήστης μπορεί να δει τις τοποθεσίες των ενεργών λεωφορείων της διαδρομής, εάν υπάρχουν.
  - 4.3. Ο χρήστης μπορεί επίσης να επιλέξει μια συγκεκριμένη στάση και να δει τις πληροφορίες σε πραγματικό χρόνο για την επόμενη άφιξη και αναχώρηση των εισερχόμενων λεωφορείων της διαδρομής, εάν υπάρχουν.
  - 4.4. Το σύστημα εμφανίζει επίσης και την επιλογή αποθήκευσης της διαδρομής αν ο χρήστης είναι συνδεδεμένος με λογαριασμό.
5. Για κάθε ενέργεια η εφαρμογή εμφανίζει τις αντίστοιχες πληροφορίες στον χάρτη.
  - 5.1. Όταν ο χρήστης επιλέξει μια διαδρομή, το σύστημα δείχνει τη διαδρομή και τις στάσεις στον χάρτη καθώς και τα ενεργά λεωφορεία, αν υπάρχουν.
  - 5.2. Όταν ο χρήστης κάνει κλικ σε ένα ενεργό λεωφορείο, ο χάρτης εστιάζει σε αυτό.
  - 5.3. Όταν ο χρήστης κάνει κλικ σε μια στάση, ο χάρτης εστιάζει στην επιλεγμένη στάση.

### **Περίπτωση χρήσης: Εμφάνιση πληροφοριών στάσης (UC4)**

#### **Περιγραφή:**

Για αυτήν την περίπτωση χρήσης, ο χρήστης πρέπει να μπορεί να αναζητήσει τις στάσεις του αστικού με τη χρήση του ονόματος, του κωδικού ή μέσω του χάρτη. Για μια επιλεγμένη στάση το σύστημα θα εμφανίζει όλα τα δρομολόγια που περνάνε από τη στάση καθώς και τις αφίξεις ανά δρομολόγιο σε πραγματικό και στατικό χρόνο.

#### **Ηθοποιοί:**

Χρήστης

#### **Προϋποθέσεις:**



- Ο χρήστης πρέπει να είναι συνδεδεμένος στο διαδίκτυο.

#### **Κύρια ροή γεγονότων:**

1. Η περίπτωση χρήσης ξεκινά όταν ο χρήστης θέλει να αναζητήσει πληροφορίες για μια συγκεκριμένη στάση.
2. Ο χρήστης κάνει κλικ στο κουμπί 'Στάσεις' και το σύστημα πλοηγείται στη λίστα όλων των στάσεων. Ταυτόχρονα το σύστημα εμφανίζει στον χάρτη όλες τις στάσεις ομαδοποιημένες σε γειτονιές(clusters).
3. Ο χρήστης μπορεί να φιλτράρει τις στάσεις με βάση το όνομα, τον κωδικό ή απλά να ανοίξει τον χάρτη και να επιλέξει μια στάση.
4. Για μια επιλεγμένη στάση, το σύστημα εμφανίζει μια λίστα με τις διαδρομές που περνάνε από αυτή καθώς και μια λίστα με τη πληροφορία πραγματικού χρόνου η οποία περιλαμβάνει τις αφίξεις, τις αναχωρήσεις και τις καθυστερήσεις.
  - 4.1. Επιλέγοντας μια διαδρομή, το σύστημα θα μεταβεί στην οθόνη με τις λεπτομέρειες της διαδρομής.
  - 4.2. Επιλέγοντας μια ζωντανή άφιξη, το σύστημα εστιάζει στο αντίστοιχο λεωφορείο.
5. Για κάθε επιλεγμένη στάση το σύστημα εμφανίζει και την επιλογή αποθήκευσης για γρήγορη εύρεση, αν ο χρήστης είναι συνδεδεμένος με λογαριασμό.

#### **Περίπτωση χρήσης: Κράτηση θέσης (UC5)**

##### **Περιγραφή:**

Για αυτήν την περίπτωση χρήσης, ο χρήστης πρέπει να μπορεί να κάνει κράτηση θέσης σε ένα συγκεκριμένο δρομολόγιο παρέχοντας τις απαραίτητες πληροφορίες. Οι πληροφορίες που θα πρέπει να εισάγει είναι:

- Αρχική στάση
- Τελική στάση
- Ώρα και μέρα μετακίνησης
- Επιθυμία άφιξης στον προορισμό ή αναχώρηση από τη αφετηρία την επιλεγμένη ώρα(προαιρετικά)

##### **Ηθοποιοί:**

Χρήστης

### **Προϋποθέσεις:**

- Ο χρήστης πρέπει να είναι συνδεδεμένος στο διαδίκτυο.
- Ο χρήστης πρέπει να έχει συνδεθεί με λογαριασμό.
- Δεν θα πρέπει να υπάρχουν αλλά δρομολόγια κλεισμένα την ίδια ώρα.
- Ο χρήστης θα πρέπει να κάνει κράτηση τουλάχιστον 30 λεπτά πριν την έναρξη του δρομολογίου από την αφετηρία, ανεξαρτήτως των επιλεγμένων στάσεων.
- Θα πρέπει να υπάρχουν θέσεις στην επιθυμητή διαδρομή.

### **Κύρια ροή γεγονότων:**

1. Η περίπτωση χρήσης ξεκινά όταν ο χρήστης θέλει να κάνει κράτηση.
2. Ο χρήστης κάνει κλικ στο κουμπί 'Κράτηση θέσης' και το σύστημα μεταφέρεται στην οθόνη με τη φόρμα για την εισαγωγή των πεδίων.
3. Αρχικά ο χρήστης επιλέγει αρχικό και τελικό σημείο.
  - 3.1. Ο χρήστης μπορεί να ψάξει στάσεις με βάση τον κωδικό ή το όνομα από τη λίστα.
  - 3.2. Ο χρήστης μπορεί να επιλέξει την τοποθεσία του ως έναρξη.
  - 3.3. Ο χρήστης μπορεί να ανοίξει τον χάρτη και να ορίσει δικά του σημεία με χρήση drag and drop λειτουργίας, όχι απαραίτητα στάσεις.
  - 3.4. Ο χρήστης μπορεί να επιλέξει μια αποθηκευμένη στάση.
4. Στη συνέχεια ο χρήστης μπορεί να συμπληρώσει τα εξής πεδία:
  - 4.1. Ημερομηνία ταξιδιού.
  - 4.2. Ώρα μετακίνησης.
  - 4.3. Επιθυμία αναχώρησης ή άφιξης την επιλεγμένη ώρα.
5. Μετά την επιλογή των απαραίτητων πεδίων, ο χρήστης πατάει το κουμπί εύρεσης.
  - 5.1. Αν κάποιο πεδίο λείπει, το σύστημα εμφανίζει μήνυμα.
  - 5.2. Αν έχουν συμπληρωθεί όλα τα πεδία, το σύστημα μεταφέρεται στην σελίδα με τα προτεινόμενα δρομολόγια.
6. Στη σελίδα με τα δρομολόγια το σύστημα εμφανίζει τις εξής πληροφορίες για κάθε δρομολόγιο:
  - 6.1. Διάρκεια ταξιδιού προσεγγιστικά.
  - 6.2. Αρχική και τελική στάση.
  - 6.3. Ώρα άφιξης και αναχώρησης.
  - 6.4. Πλήθος διαθέσιμων θέσεων μέχρι στιγμής.

7. Πατώντας το κουμπί 'Κράτηση', ο χρήστης μπορεί να κάνει κράτηση στο δρομολόγιο που θέλει.

7.1. Αν δεν παραβιάζονται οι προϋποθέσεις, το σύστημα εμφανίζει μήνυμα επιτυχίας.

7.2. Αν υπάρχει σφάλμα, το σύστημα ενημερώνει κατάλληλα τον χρήστη.

8. Για κάθε δρομολόγιο ο χρήστης έχει τη δυνατότητα να το δει και οπτικά πατώντας στο κουμπί του χάρτη. Επιλέγοντας αυτή την λειτουργία η εφαρμογή:

8.1. Φτιάχνει τη διαδρομή στον χάρτη.

8.2. Εμφανίζει πλήρεις οδηγίες όπως τις ενδιάμεσες στάσεις, χρόνο μεταφοράς από στάση σε στάση προσεγγιστικά και τυχόν αναμονή ή καθυστέρηση πριν την επιβίβαση.

8.3. Αν ένα δρομολόγιο δεν είναι απευθείας με βάση τις επιλογές του χρήστη, το σύστημα θα εμφανίσει οδηγίες για τις στάσεις αναχώρησης και άφιξης.

9. Αν δεν υπάρχουν διαθέσιμα δρομολόγια, το σύστημα εμφανίζει κατάλληλο μήνυμα.

#### **Εναλλακτική ροή:**

1. Ο χρήστης δεν έχει συνδεθεί.

2. Ο χρήστης μεταφέρεται στη σελίδα εισόδου.

#### **Περίπτωση χρήσης: Εμφάνιση πληροφορίας κράτησης (UC6)**

##### **Περιγραφή:**

Για αυτήν την περίπτωση χρήσης, ο χρήστης πρέπει να μπορεί να δει τις κρατήσεις του και τις λεπτομέρειές τους.

##### **Ηθοποιοί:**

Χρήστης

##### **Προϋποθέσεις:**

- Ο χρήστης πρέπει να είναι συνδεδεμένος στο διαδίκτυο.
- Ο χρήστης πρέπει να έχει συνδεθεί με λογαριασμό.

##### **Κύρια ροή γεγονότων:**

1. Η περίπτωση χρήσης ξεκινά όταν ο χρήστης θέλει να εμφανίσει τις κρατήσεις του.
2. Ο χρήστης πατάει στο κουμπί 'Οι κρατήσεις μου' και το σύστημα μεταφέρεται στην οθόνη με τη λίστα των κρατήσεων.

3. Ο χρήστης μπορεί να αναζητήσει κρατήσεις με βάση τις στάσεις, κωδικό δρομολογίου ή την ημερομηνία.
4. Για κάθε κράτηση το σύστημα εμφανίζει τις επιλογές 'Χάρτης', 'QR κωδικός', 'Διαγραφή'.
  - 4.1. Πατώντας την επιλογή του χάρτη, το σύστημα εμφανίζει την διαδρομή με τις στάσεις και τις λεπτομέρειες.
  - 4.2. Πατώντας την επιλογή του κωδικού QR, το σύστημα θα μεταφερθεί στη σελίδα με τον κωδικό QR που περιγράφει την κράτηση.
  - 4.3. Πατώντας την επιλογή διαγραφής, το σύστημα διαγράφει τη κράτηση και έπειτα ενημερώνει τη βάση δεδομένων.
5. Αν δεν υπάρχουν κρατήσεις, το σύστημα εμφανίζει μήνυμα και δίνει τη δυνατότητα μεταφοράς στη σελίδα δημιουργίας κράτησης.

**Εναλλακτική ροή:**

1. Ο χρήστης δεν έχει συνδεθεί.
2. Ο χρήστης μεταφέρεται στη σελίδα εισόδου.

## Κεφάλαιο 5. Σχεδίαση και υλοποίηση

Αυτό το κεφάλαιο έχει ως σκοπό να αναλύσει την αρχιτεκτονική και τις σχεδιαστικές αρχές οι οποίες χρησιμοποιήθηκαν για τη υλοποίηση της εφαρμογής. Έπειτα θα δοθεί μια περιγραφή της οργάνωσης της βάσης δεδομένων, έτσι όπως προέκυψε μετά από την εξόρυξη και ομαδοποίηση των δεδομένων από το ανοιχτό API του αστικού ΚΤΕΛ Ιωάννινων. Θα δοθούν επίσης σχήματα και περιγραφές που αφορούν τις ενότητες (Modules) και τα διαγράμματα των κλάσεων για τη καλύτερη κατανόηση της διάρθρωσης της εφαρμογής.

### 5.1. Αρχιτεκτονική τριών επιπέδων

Η αρχιτεκτονική η οποία θα χρησιμοποιηθεί για την υλοποίηση της εφαρμογής είναι η αρχιτεκτονική τριών επιπέδων. Η αρχιτεκτονική τριών επιπέδων είναι ένα μοτίβο σχεδιασμού λογισμικού που διαχωρίζει μια εφαρμογή σε τρία διακριτά επίπεδα: το επίπεδο παρουσίασης, το επίπεδο λογικής της εφαρμογής και το επίπεδο αποθήκευσης δεδομένων. Τα τρία αυτά επίπεδα λειτουργούν σε διαφορετικές πλατφόρμες και είναι λογικά και φυσικά διαχωρισμένα. Η ανάπτυξη των τριών επιπέδων γίνεται ανεξάρτητα και κάθε επίπεδο μπορεί να επεκταθεί χωρίς να επηρεάζει τις λειτουργίες των άλλων επιπέδων. Στη αρχιτεκτονική τριών επιπέδων όλη η επικοινωνία συμβαίνει μέσω του επιπέδου λογικής. Το επίπεδο παρουσίασης και το επίπεδο δεδομένων δεν μπορούν να επικοινωνήσουν απευθείας. Παρακάτω περιγράφονται αναλυτικότερα τα επίπεδα της αρχιτεκτονικής αυτής.

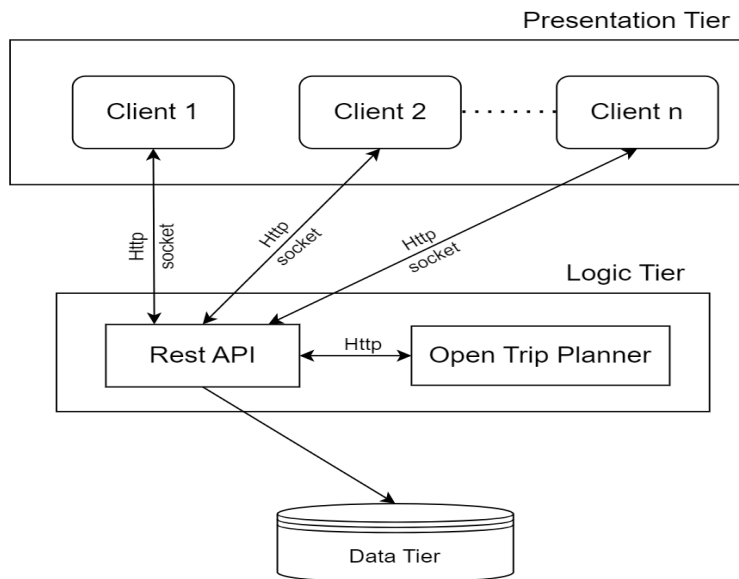
1. Το επίπεδο παρουσίασης είναι υπεύθυνο για την εμφάνιση πληροφοριών στον χρήστη και το χειρισμό των ενεργειών του. Συνήθως περιλαμβάνει μια γραφική διεπαφή χρήστη (GUI) και μπορεί επίσης να περιλαμβάνει ιστοσελίδες, εφαρμογές για κινητά ή διεπαφές γραμμής εντολών.
2. Το επίπεδο λογικής εφαρμογής είναι υπεύθυνο για την υλοποίηση της επιχειρηματικής λογικής της εφαρμογής. Επεξεργάζεται τα δεδομένα των χρηστών και ανακτά ή ενημερώνει δεδομένα από το επίπεδο αποθήκευσης δεδομένων. Χειρίζεται επίσης την επικοινωνία μεταξύ του επιπέδου παρουσίασης και επιπέδου δεδομένων.
3. Το επίπεδο αποθήκευσης δεδομένων είναι υπεύθυνο για την αποθήκευση και ανάκτηση δεδομένων από βάσεις δεδομένων, συστήματα αρχείων ή άλλους

τύπους αποθήκευσης δεδομένων. Αυτό το επίπεδο αφαιρεί την υποκείμενη αποθήκευση δεδομένων από την υπόλοιπη εφαρμογή, επιτρέποντας στο επίπεδο λογικής και στο επίπεδο παρουσίασης να αλλάζουν χωρίς να επηρεάζεται το επίπεδο αποθήκευσης.

Αυτή η αρχιτεκτονική παρέχει μια σειρά από πλεονεκτήματα όπως:

- Βελτιωμένη επεκτασιμότητα γιατί κάθε επίπεδο μπορεί να αναπτυχθεί και να επεκταθεί χωρίς να επηρεάζει τα υπόλοιπα.
- Καλύτερο διαχωρισμό των λειτουργιών γιατί κάθε επίπεδο έχει μια συγκεκριμένη αρμοδιότητα.
- Ασφάλεια γιατί το επίπεδο παρουσίασης δεν μπορεί να επικοινωνήσει απευθείας με το επίπεδο δεδομένων. Αυτό δίνει την δυνατότητα ελέγχου των αιτημάτων πριν την επεξεργασία τους.

Σχηματικά η αρχιτεκτονική της προτεινόμενης εφαρμογής παρουσιάζεται και στην εικόνα 7. Αρχικά παρατηρούμε ότι έχουμε το επίπεδο παρουσίασης το οποίο θα είναι γραμμένο σε Angular. Το επίπεδο αυτό επικοινωνεί άμεσα με το API στο επίπεδο λογικής μέσω αιτημάτων HTTP και σύνδεσης socket. Έπειτα έχουμε το επίπεδο λογικής το οποίο αποτελείται από 2 μέρη. Το API ή αλλιώς ο διακομιστής της εφαρμογής και το λογισμικό Open Trip Planner. Το API θα έχει ως κύρια αρμοδιότητα την διασύνδεση του επιπέδου παρουσίασης με το επίπεδο δεδομένων, και την υλοποίηση της επιχειρηματικής λογικής της εφαρμογής(business logic). Θα επικοινωνεί επίσης με τον OTP μέσω αιτημάτων HTTP. Το λογισμικό OTP θα χρησιμοποιείται για την εκτίμηση αφίξεων σε πραγματικό χρόνο καθώς και για την εύρεση δρομολογίων. Μόλις το API λάβει κάποιο αίτημα που αφορά αυτές τις περιπτώσεις θα καλεί τον OTP και θα λαμβάνει τα αντίστοιχα δεδομένα. Μετά από κάποια επεξεργασία θα τα στέλνει στο επίπεδο παρουσίασης. Τέλος έχουμε και το επίπεδο δεδομένων το οποίο θα υλοποιηθεί με χρήση της PostgreSQL.



Εικόνα 7: Αρχιτεκτονική της εφαρμογής

## 5.2. Υλοποίηση επιπέδου δεδομένων

Έχοντας θέσει τα θεμέλια της εφαρμογής, το επόμενο βήμα είναι η σχεδίαση και ανάπτυξη των κομματιών της. Αρχικά το πρώτο βήμα περιλαμβάνει την υλοποίηση του επιπέδου δεδομένων.

### 5.2.1. Εξόρυξη δεδομένων

Η προτεινόμενη εφαρμογή που πραγματεύεται αυτή η διπλωματική εργασία έχει ως κύριο χαρακτηριστικό την εμφάνιση πληροφοριών στον χρήστη. Αυτό σημαίνει ότι η συλλογή και αποθήκευση τους είναι απαραίτητη. Θα πρέπει να έχουμε δεδομένα οργανωμένα σε μοτίβα τα οποία θα ικανοποιήσουν τις ανάγκες της εύκολης και γρήγορης εύρεσης και παρουσίασης.

Η πρώτη εργασία που έγινε είναι έρευνα των πηγών για την άντληση τους. Η ιστοσελίδα του αστικού ΚΤΕΛ Ιωαννίνων παρέχει μια διεπαφή χρήστη η οποία εμφανίζει βασικά δεδομένα που αφορούν τις γραμμές και το ημερήσιο πρόγραμμα. Με χρήση των εργαλείων που μας παρέχει ο περιηγητής, έγινε έρευνα του πηγαίου κώδικα της ιστοσελίδας και από εκεί εντοπίστηκαν τα βασικά endpoints του διακομιστή.

Στη συνέχεια, με χρήση της εφαρμογής Postman, έγινε κλήση αυτών των endpoints. Η εφαρμογή αυτή χρησιμοποιείται για τον έλεγχο και τεστ ενός API. Παρέχει μια διεπαφή η οποία παρουσιάζει τα αποτελέσματα κάθε φορά, χωρίς να χρειάζεται η γραφή κώδικα. Παρατηρήθηκε ότι αυτά τα endpoints παρέχουν τις ακόλουθες πληροφορίες:

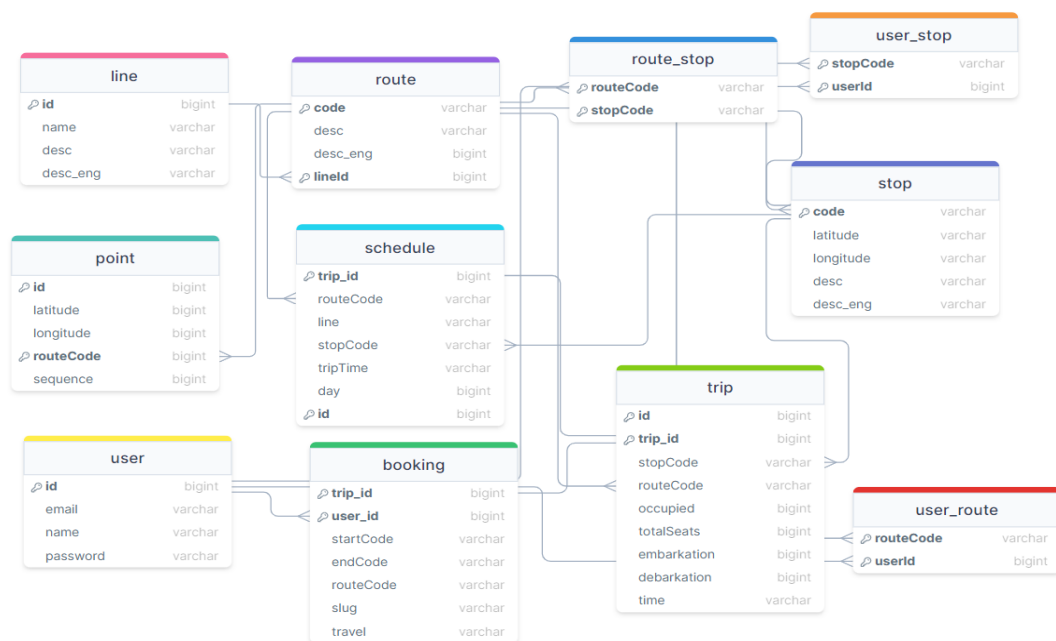
- Μια λίστα των κύριων γραμμών και για κάθε γραμμή μια λίστα με όλες τις διαδρομές.
- Μια λίστα με τις στάσεις.
- Μια διαδρομή με όλα τα σημεία του μονοπατιού της.
- Μια λίστα διαδρομών και για κάθε διαδρομή μια λίστα με τις στάσεις από τις οποίες περνάει.
- Μια λίστα με τους χρόνους αναχώρησης για κάθε στάση ανά μέρα.
- Μια λίστα με τους ζωντανούς χρόνους αναχώρησης για κάποια στάση. Η πληροφορία αυτή δίνει τις συντεταγμένες της στάσης και του αντίστοιχου λεωφορείου.
- Μια λίστα με τις πληροφορίες της θέσης και τους κωδικούς των ενεργών λεωφορείων μιας γραμμής.

Με βάση αυτά τα αποτελέσματα, έγινε ένα φιλτράρισμα για την αποθήκευση μόνο των απαραίτητων πληροφοριών, καθώς παρατηρήθηκε ότι τα δεδομένα είχαν πολλά πεδία τα οποία αφορούσαν αποκλειστικά την σελίδα του αστικού ΚΤΕΛ. Έπειτα ακολούθησε ο διαχωρισμός των δεδομένων και η σχεδίαση της βάσης χρησιμοποιώντας τα αποτελέσματα αυτά, λαμβάνοντας υπόψη και τις υπόλοιπες περιπτώσεις χρήσης της εφαρμογής.

### **5.2.2. Σχεδίαση και υλοποίηση βάσης δεδομένων**

Αρχικά η οργάνωση των δεδομένων θα πρέπει να έχει δυναμική σχεδίαση έτσι ώστε να δέχεται ενημερώσεις χωρίς να υπάρχει δυσκολία κατά την επέκταση. Ιδιαίτερη προσοχή δόθηκε και στη σχεδίαση των συσχετίσεων, έτσι ώστε οι ερωτήσεις στη βάση να είναι οι ελάχιστες δυνατές και να εκτελούνται χωρίς καθυστερήσεις. Στην εικόνα 8 παρουσιάζεται ένα διάγραμμα με τους βασικούς πίνακες της βάσης με τα πεδία και τις συσχετίσεις τους.





Εικόνα 8: Η σχεδίαση της βάσης δεδομένων

Στη συνέχεια δίνεται μια αναλυτικότερη περιγραφή των οντοτήτων της βάσης. Για κάθε οντότητα παρουσιάζεται ένας πίνακας με λεπτομερή περιγραφή των πεδίων και των συσχετίσεων του.

Γραμμή(line)	
Μια γονική διαδρομή η οποία περιλαμβάνει πολλές διαδρομές	
lineId	Αναγνωριστικό γραμμής
name	Κωδικό όνομα γραμμής
desc	Όνομα γραμμής στα ελληνικά
desc_eng	Όνομα γραμμής στα αγγλικά

Πίνακας 1: Γραμμή(line)

Διαδρομή(route)	
Μια ακολουθία από συνεχόμενες στάσεις	
code	Κωδικός διαδρομής
lineId	Ξένο κλειδί προς μια γονική γραμμή
desc	Όνομα διαδρομής στα ελληνικά
desc_eng	Όνομα διαδρομής στα αγγλικά

Πίνακας 2: Διαδρομή(route)

<b>Στάση(stop)</b> Μια στάση από την οποία μπορεί να περνάνε μια ή περισσότερες διαδρομές	
code	Κωδικός στάσης
latitude	Γεωγραφικό πλάτος
longitude	Γεωγραφικό μήκος
desc	Όνομα στάσης στα ελληνικά
desc_eng	Όνομα στάσης στα αγγλικά

Πίνακας 3: Στάση(stop)

<b>Στάση διαδρομής(route_stop)</b> Περιγράφει την 'πολλά προς πολλά' σχέση μεταξύ στάσης και διαδρομής	
routeCode	Ξένο κλειδί προς μια διαδρομή
stopCode	Ξένο κλειδί προς μια στάση

Πίνακας 4: Στάση διαδρομής(line)

<b>Σημείο(point)</b> Ένα σημείο του μονοπατιού μιας διαδρομής	
id	Μοναδικό αναγνωριστικό σημείου
latitude	Γεωγραφικό πλάτος
longitude	Γεωγραφικό μήκος
routeCode	Ξένο κλειδί προς τη διαδρομή
sequence	Αριθμός ακολουθίας

Πίνακας 5: Σημείο(Point)

<b>Χρήστης(user)</b> Στοιχεία λογαριασμού ενός χρήστη	
id	Αναγνωριστικό χρήστη
email	Email χρήστη
name	Όνομα χρήστη
password	Κωδικός εισόδου στην εφαρμογή

Πίνακας 6: Χρήστης(user)

<b>Ημερήσιο πρόγραμμα(schedule)</b> Περιγράφει τα δρομολόγια των διαδρομών μέσα στην βδομάδα	
trip_id	Αναγνωριστικό δρομολογίου
routeCode	Κωδικός διαδρομής
line	Κωδικός γραμμής
stopCode	Κωδικός στάσης της διαδρομής
tripTime	Ώρα αναχώρησης από τη στάση
day	Μέρα εβδομάδας πραγματοποίησης του δρομολογίου

Πίνακας 7: Ημερήσιο πρόγραμμα(schedule)

<b>Διαδρομή χρήστη(user_route)</b> Κρατάει τις αποθηκευμένες διαδρομές των χρηστών	
routeCode	Ξένο κλειδί προς την διαδρομή
userId	Ξένο κλειδί προς τον χρήστη

Πίνακας 8: Διαδρομή χρήστη(user\_route)

<b>Στάση χρήστη(user_stop)</b> Κρατάει τις αποθηκευμένες στάσεις των χρηστών	
stopCode	Ξένο κλειδί προς την στάση
userId	Ξένο κλειδί προς τον χρήστη

Πίνακας 9: Στάση χρήστη(user\_stop)

<b>Κράτηση(booking)</b> Αποθηκεύει τις κρατήσεις των χρηστών	
trip_id	Ξένο κλειδί προς ένα δρομολόγιο
user_id	Ξένο κλειδί προς έναν χρήστη
startCode	Κωδικός στάσης επιβίβασης
endCode	Κωδικός στάσης αποβίβασης
routeCode	Κωδικός διαδρομής δρομολογίου
time	Ώρα αναχώρησης
slug	Αλφαριθμητικό με πληροφορίες για την κατασκευή της διαδρομής

Πίνακας 10: Κράτηση(booking)

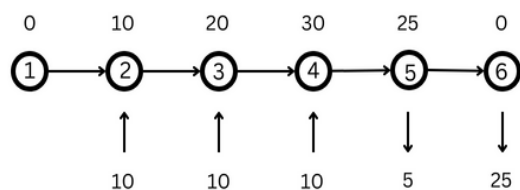
<b>Δρομολόγιο(trip)</b> Αποθηκεύει την κατάσταση πληρότητας των δρομολογίων	
id	Κωδικός εγγραφής
trip_id	Κωδικός δρομολογίου
occupied	Επιβάτες στο λεωφορείο από στάση σε στάση
embarkation	Επιβάτες που θα ανέβουν στην επόμενη στάση
debarkation	Επιβάτες που θα κατεβούν στην επόμενη στάση
totalSeats	Μέγιστη πληρότητα του λεωφορείου
stopCode	Κωδικός επόμενης στάσης
time	Ώρα αναχώρησης από την επόμενη στάση

Πίνακας 11: Δρομολόγιο(trip)

Ο πίνακας trip είναι πολύ σημαντικός γιατί με βάση αυτόν η εφαρμογή θα μπορεί να κάνει τον έλεγχο αν υπάρχουν θέσεις για μια κράτηση. Θα λειτουργήσει και ως ιστορικό με βάση το οποίο θα μπορεί να παρατηρηθεί η ροή των επιβατών και να γίνει ανίχνευση των ωρών αιχμής. Το σύστημα μόλις λάβει μια επιτυχή κράτηση, αυξάνει το πεδίο occupied κατά 1 από την αρχική στάση της υποδιαδρομής ενός δρομολογίου μέχρι και την προτελευταία. Παράλληλα αυξάνει τις τιμές debarkation και embarkation στις στάσεις που επέλεξε ο χρήστης κατά 1. Σημαντικό σε αυτό το σημείο είναι η σύνταξη γρήγορων και μικρού αριθμού ερωτήσεων SQL για την ενημέρωση. Αυτό έγινε με τη δημιουργία λεξικών με χρήση B δένδρων πάνω στα απαραίτητα πεδία.

Σε αντίθεση με τα υπεραστικά λεωφορεία στα αστικά έχουμε μεταβλητό πλήθος επιβατών κατά τη διάρκεια ενός δρομολογίου. Αυτό σημαίνει ότι για να καταλάβουμε αν μια κράτηση μπορεί να πραγματοποιηθεί πρέπει να λάβουμε υπόψη αν σε κάποιο σημείο της επιλεγμένης υποδιαδρομής το λεωφορείο είναι ήδη γεμάτο. Αν ο έλεγχος γίνει με βάση τους επιβάτες που θα βρίσκονται στο λεωφορείο στη στάση επιβίβασης μπορεί να έχουμε συγκρούσεις γιατί μπορεί σε μια από τις επόμενες στάσεις να έχουν ήδη γίνει κρατήσεις οι οποίες θα γεμίζουν το λεωφορείο.

Ο έλεγχος ελεύθερων θέσεων πραγματοποιείται ως εξής: για ένα δρομολόγιο το σύστημα ελέγχει τη μέγιστη τιμή occupied της επιθυμητής υποδιαδρομής. Αν αυτή είναι μικρότερη του πεδίου totalSeats, τότε η κράτηση πραγματοποιείται. Σχηματικά η διαδικασία κράτησης παρουσιάζεται και στην εικόνα 9. Αρχικά για την περίπτωση αυτή ορίζουμε την τιμή totalSeats 30.



Εικόνα 9 Στιγμιότυπο ροής επιβατών

Όπως φαίνεται στην εικόνα 9 στις στάσεις 2, 3, 4 ανεβαίνουν από 10 άτομα. Η τιμή occupied αυξάνεται ανά 10 μέχρι τη στάση 4 όπου το λεωφορείο γεμίζει. Έπειτα 5 αποβιβάζονται στη στάση 5 και οι υπόλοιποι στη στάση 6. Έστω ότι ένας χρήστης θέλει να κάνει κράτηση από τη στάση 2 και να πάει στη στάση 6. Το σύστημα θα ειδοποιήσει τον χρήστη ότι η κράτηση δεν είναι δυνατή γιατί στη στάση 4, η οποία περιλαμβάνεται στη διαδρομή του χρήστη, το λεωφορείο είναι πλήρως κατειλημμένο με βάση τις ήδη υπάρχουσες κρατήσεις. Στη στάση 4 έχουμε τη μέγιστη τιμή occupied. Γενικά το σύστημα θα προτείνει διάφορα δρομολόγια τα οποία είναι αρκετά κοντά στις ανάγκες του χρήστη και για κάθε ένα από αυτά η τιμή occupied ανανεώνεται σε πραγματικό χρόνο, που σημαίνει ότι αν υπάρχει ακύρωση κράτησης ο χρήστης θα δει άμεσα την αλλαγή αυτή. Παρόλου που αυτή η λύση εισάγει την δυνατότητα προτεραιότητας, μπορεί να υπάρξουν διαστήματα που κάποιο λεωφορείο είναι κενό για κάποια ακολουθία στάσεων.

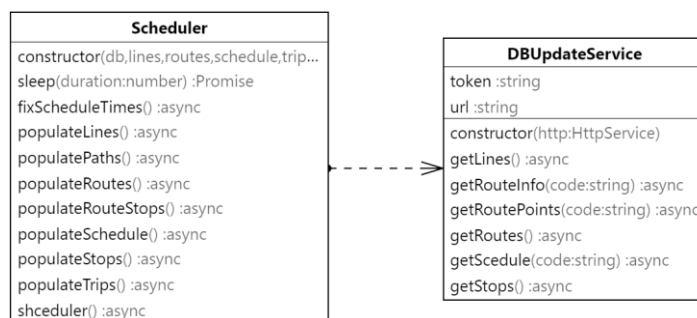
Εκτός από τα στατικά δεδομένα έχουμε και δεδομένα πραγματικού χρόνου τα οποία θέλουμε να τα στέλνουμε στο επίπεδο παρουσίασης. Τα δεδομένα αυτά παρουσιάζουν αυξημένο ενδιαφέρον γιατί περιέχουν πληροφορία που μπορεί να μας βοηθήσει να εξάγουμε συμπεράσματα που αφορούν τις εκτιμήσεις αφίξεων και αναχωρήσεων καθώς και την ροή των αστικών λεωφορείων. Για τον λόγο αυτό δημιουργήθηκε ένας ξεχωριστός πίνακας στη βάση για τη καταγραφή τους. Ο πίνακας αυτός εμπλουτίστηκε και με πληροφορία με τη χρήση του Open Trip Planner. Πιο συγκεκριμένα έγινε ο υπολογισμός της ώρα άφιξης στις στάσεις. Για την συλλογή τους γράφτηκε μια μέθοδος η οποία καλεί το αντίστοιχο endpoint του API του αστικού ΚΤΕΛ ανά μικρά χρονικά διαστήματα και τα αποθηκεύει. Ο πίνακας των δεδομένων πραγματικού χρόνου παρουσιάζεται αναλυτικά παρακάτω.

<b>Δεδομένα πραγματικού χρόνου(live_data)</b> Αποθηκεύει τα δεδομένα πραγματικού χρόνου	
routeCode	Κωδικός διαδρομής
lineCode	Κωδικό όνομα γραμμής
stopCode	Στάση που αφορά η πρόβλεψη
arrivalTime	Χρόνος άφιξης σε λεπτά
departureTime	Χρόνος αναχώρησης σε λεπτά
vehicleCode	Κωδικός οχήματος
stopLatitude	Γεωγραφικό πλάτος στάσης
stopLongitude	Γεωγραφικό μήκος στάσης
vehicleLatitude	Γεωγραφικό πλάτος λεωφορείου
vehicleLongitude	Γεωγραφικό μήκος λεωφορείου
timestamp	Ωρα εισαγωγής εγγραφής στη βάση

Πίνακας. 12. Δεδομένα πραγματικού χρόνου

### 5.2.3. Φόρτωση της βάσης

Μετά την σχεδίαση και υλοποίηση της βάσης έγινε φόρτωση των δεδομένων. Τα δεδομένα φορτωθήκαν με χρήση ερωτήσεων SQL οι οποίες γράφτηκαν με τη βοήθεια του εργαλείου TypeOrm. Με το εργαλείο αυτό έγινε και η δημιουργία των πινάκων, χάρις της δυνατότητας αντιστοιχίας κλάσεων σε πίνακες βάσης δεδομένων. Στη διαδικασία αυτή βοήθησε πολύ και η γραφική διεπαφή που προσφέρει η PostgreSQL με την οποία είναι δυνατή η σύνταξη ερωτήσεων, η τροποποίηση και ο έλεγχος των δεδομένων. Για την φόρτωση των δεδομένων υλοποιήθηκαν οι κλάσεις DBUpdateService και Scheduler οι οποίες φαίνονται στην εικόνα 10.



Εικόνα.10 Κλάσεις για την αναβάθμιση της βάσης δεδομένων.

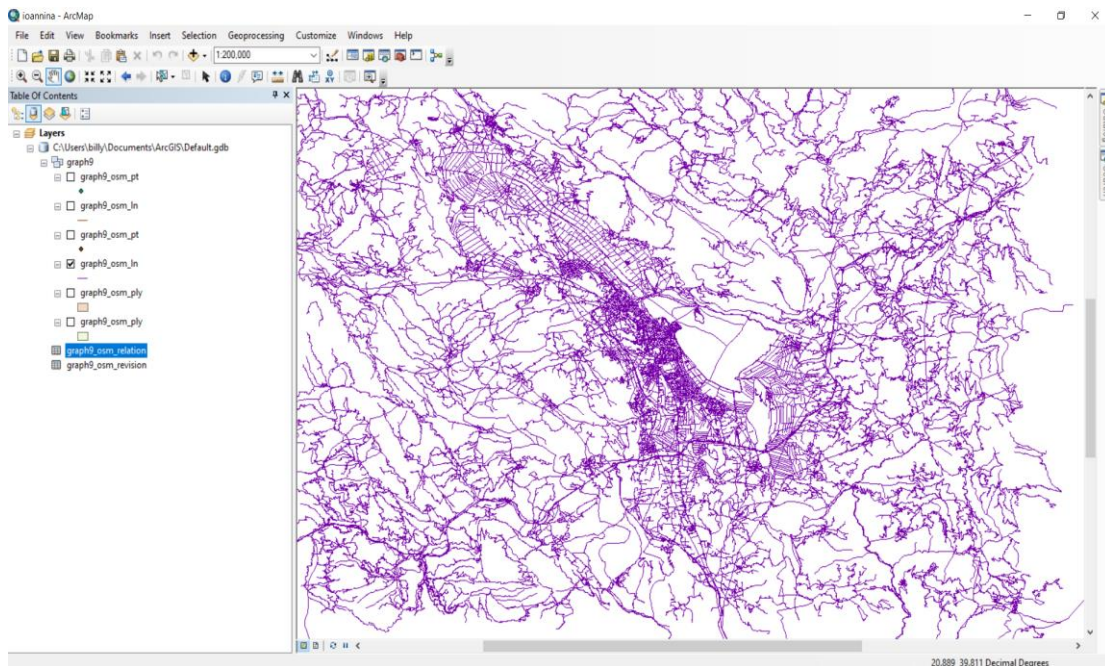
Η κλάση Scheduler έχει μια μέθοδο η οποία ορίζει μια διεργασία που τρέχει σε καθορισμένα χρονικά διαστήματα. Αρμοδιότητα της είναι να αναβαθμίζει τη βάση. Η κλάση αυτή δέχεται τις κλάσεις που υλοποιούν τις ερωτήσεις SQL(repositories) ως πεδία με dependency injection, και ταυτόχρονα καλεί την κλάση DBUpdateService. Η κλάση αυτή καλεί τα endpoints του διακομιστή του αστικού KTEL για την λήψη των δεδομένων. Στην εικόνα 10 βλέπουμε και τις μεθόδους των κλάσεων αυτών.

## 5.3. Υλοποίηση επιπέδου λογικής

Αφού ολοκληρωθεί η σχεδίαση και υλοποίηση του επιπέδου δεδομένων, το επόμενο βήμα είναι η υλοποίηση του επιπέδου της λογικής. Όπως ορίστηκε στην ενότητα 5.1 το επίπεδο λογικής αποτελείται από 2 μέρη. Το λογισμικό Open Trip Planner και τον διακομιστή ή αλλιώς το REST API της εφαρμογής.

### 5.3.1. Προετοιμασία Open Trip Planner

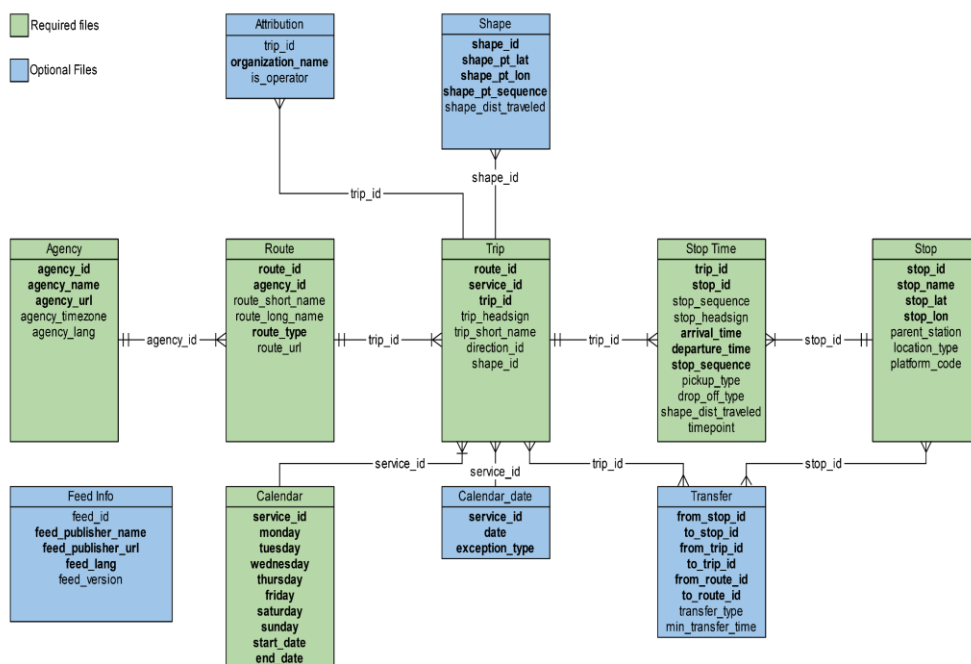
Το πρώτο βήμα της υλοποίησης του επιπέδου λογικής είναι η ρύθμιση του OTP. Όπως αναφέρθηκε, το λογισμικό συνδυάζει ένα οδικό δίκτυο καθώς και τα δεδομένα συγκοινωνίας από μια υπηρεσία δημόσιων μεταφορών. Η εξαγωγή του δικτύου έγινε με χρήση του προγράμματος ArcMap το οποίο είναι ένα πρόγραμμα χαρτογράφησης. Η πηγή του γραφήματος προήλθε από τα δεδομένα του Open Street Map, μια ανοιχτή γεωγραφική βάση δεδομένων. Τα δεδομένα του OSM είναι πάρα πολύ λεπτομερή και ανανεώνονται συνεχώς καθιστώντας τα μια καλή λύση. Περιέχουν πληροφορίες όπως μέσες ταχύτητες στους δρόμους με βάση την κίνηση, τύπους κόμβων κ.α. Η χαρτογράφηση έγινε με βάση το bounding box της πόλης των Ιωάννινων, το οποίο δόθηκε ως είσοδο στο πρόγραμμα ArcMap για εξαγωγή και μετατροπή στο απαιτούμενο format. Η εικόνα 11 απεικονίζει ένα στιγμιότυπο της επεξεργασίας των δεδομένων του OSM.



Εικόνα 11. Το δίκτυο των Ιωάννινων

Έπειτα έγινε η κατασκευή των αρχείων GTFS(general transit feed specification), τα οποία περιγράφουν συνοπτικά τις διαδρομές και τα δρομολόγια του αστικού ΚΤΕΛ. Για τη δημιουργία κατασκευάστηκαν script σε python τα οποία, με είσοδο των δεδομένων της βάσης, φτιάχνουν τα απαραίτητα αρχεία. Ακολουθεί ένα σχήμα των αρχείων αυτών.

GTFS-static model



Εικόνα 12. Το μοντέλο αρχείων GTFS(πηγή: trafiklab.se)



Από τα αρχεία που φαίνονται στην εικόνα 12 θα χρειαστούμε τα ακόλουθα: Shape, Route, Trip, Stop Time, Stop και Calendar. Πιο συγκεκριμένα:

- Το αρχείο Route έχει μια λίστα με όλες τις κύριες γραμμές(master lines) και θα το αντιστοιχίσουμε στον πίνακα Line.
- Το αρχείο Shape έχει όλα τα σημεία(points) μιας διαδρομής και θα το αντιστοιχίσουμε στον πίνακα route. Έτσι με το shape\_id πεδίο θα εννοούμε το πεδίο route\_code.
- Το αρχείο Stop έχει όλες τις στάσεις και αντιστοιχίζεται στον πίνακα stop.
- Το αρχείο Trip έχει μια λίστα με τα δρομολόγια(trips) και για κάθε δρομολόγιο ποια διαδρομή(shape) το εκτελεί, την γραμμή της διαδρομής(route) και την υπηρεσία που το παρέχει(service).
- Το αρχείο Stop Time ακολουθεί την λογική του πίνακα schedule. Έχουμε για κάθε δρομολόγιο και για κάθε στάση τους χρόνους άφιξης και αναχώρησης.
- Το αρχείο Calendar ορίζει τις υπηρεσίες και παίρνει τιμές 0 ή 1 ανάλογα με το αν ένα δρομολόγιο προσφέρεται κάποια μέρα ή όχι. Έτσι με βάση το πεδίο day στον πίνακα schedule κάθε δρομολόγιο(trip) θα έχει και μια υπηρεσία. Επίσης στο αρχείο αυτό ορίζεται και η περίοδος που παρέχεται μια υπηρεσία.

Εκτός από τα στατικά δεδομένα, ο trip planner θα δεχθεί και τα δεδομένα πραγματικού χρόνου τα οποία λαμβάνονται από το API του ΚΤΕΛ ανά μικρά χρονικά διαστήματα και δίνουν μια εκτίμηση αναχώρησης για κάποια στάση, όχι μεγαλύτερη των 20 λεπτών. Για την εργασία αυτή υπάρχουν δεδομένα που περιγράφουν τις αναχωρήσεις και παράλληλα θα γίνει και υπολογισμός των αφίξεων με βάση την τοποθεσία των λεωφορείων. Έτσι ο OTP θα λαμβάνει υπόψιν του όχι μόνο το ημερήσιο πρόγραμμα αλλά και κάποιες προβλέψεις σε πραγματικό χρόνο, τις οποίες θα λαμβάνει ανά μικρά χρονικά διαστήματα κάνοντας ένα αίτημα HTTP στον διακομιστή της εφαρμογής. Έπειτα τα δεδομένα θα λαμβάνονται από το αστικό ΚΤΕΛ και θα μετατρέπονται σε αρχεία protobuf. Με βάση αυτά ο OTP θα προσαρμόζει τα δρομολόγια που προτείνει στους χρήστες. Τα πιο σημαντικά πεδία του αρχείου protobuf είναι:

- Κωδικός στάσης(stop\_id).
- Κωδικός διαδρομής(shape\_id)
- Χρόνος άφιξης(arrival\_time)
- Χρόνος αναχώρησης(departure\_time)
- Κωδικός δρομολογίου(trip\_id).

### 5.3.2. Υλοποίηση διακομιστή(API)

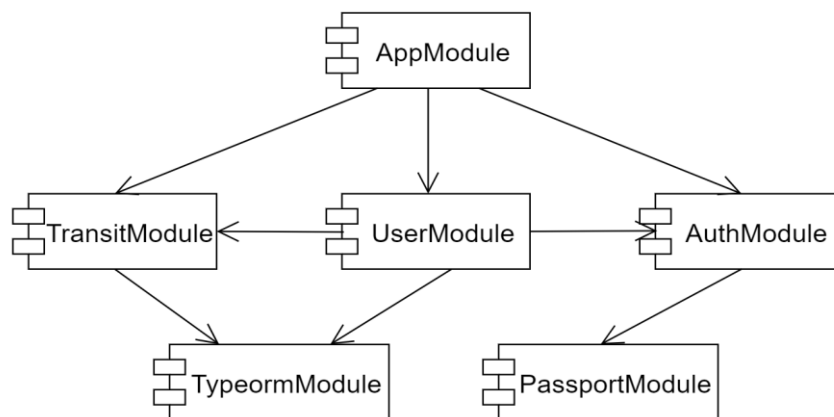
Στο βήμα αυτό θα υλοποιήσουμε τον διακομιστή(API). Αυτό το κομμάτι είναι η καρδιά της εφαρμογής και έχει αρμοδιότητα την διασύνδεση όλων των υπόλοιπων κομματιών της καθώς και τον ορισμό όλων των μεθόδων για την επεξεργασία των δεδομένων.

#### 5.3.2.1. Σχεδιαστικό Μοτίβο Αποθετήριο(Repository Pattern)

Η σχεδιαστική αρχή που χρησιμοποιήθηκε για την υλοποίηση του επιπέδου λογικής είναι το μοτίβο του αποθετηρίου. Το μοτίβο αυτό είναι ένα μοτίβο σχεδιασμού που χρησιμοποιείται στην ανάπτυξη λογισμικού για τον διαχωρισμό των ευθυνών πρόσβασης δεδομένων και λογικής σε μια εφαρμογή λογισμικού. Το μοτίβο περιλαμβάνει τη δημιουργία μιας κλάσης αποθετηρίου που χειρίζεται όλες τις λειτουργίες της βάσης δεδομένων όπως η ανάκτηση και αποθήκευση, μια κλάση υπηρεσίας που χειρίζεται την επιχειρηματική λογική και την επικοινωνία μεταξύ του αποθετηρίου και του ελεγκτή και μια κλάση ελεγκτή που υλοποιεί τη διεπαφή REST της εφαρμογής, συνδέοντας έτσι το επίπεδο παρουσίασης με το επίπεδο λογικής. Αυτός ο διαχωρισμός των λειτουργιών επιτρέπει μια πιο αρθρωτή, επεκτάσιμη και διατηρήσιμη βάση κώδικα. Τυπικά σε μια εφαρμογή nestJS έχουμε μια κλάση υπηρεσίας και μια κλάση αποθετηρίου για κάθε οντότητα της εφαρμογής. Οι οντότητες είναι κλάσεις οι οποίες περιγράφουν τα δεδομένα (domain model) και αντιστοιχίζονται στις οντότητες(πίνακες) της βάσης δεδομένων.

#### 5.3.2.2. Ενότητες(Modules)

Σε αυτή την ενότητα περιγράφονται όλα τα modules του επιπέδου λογικής. Στην εικόνα 13 παρουσιάζεται το αντίστοιχο διάγραμμα με τις συσχετίσεις τους.



Εικόνα 13. Τα modules του επιπέδου λογικής

## App Module

Στο NestJS, το app module αναφέρεται σε μια συγκεκριμένη λειτουργική μονάδα που δημιουργείται όταν δημιουργείται μια εφαρμογή NestJS. Αυτή η μονάδα χρησιμοποιείται συνήθως για την εισαγωγή και την παροχή καθολικών (δηλαδή σε επίπεδο εφαρμογής) στοιχείων, όπως ελεγκτές και παρόχους. Χρησιμοποιείται επίσης για τον καθορισμό οποιασδήποτε διαμόρφωσης σε όλη την εφαρμογή, όπως ενδιάμεσο λογισμικό(middleware) ή καθολικούς φύλακες(guards). Η λειτουργική μονάδα αυτή είναι ένα θεμελιώδες δομικό στοιχείο μιας εφαρμογής NestJS και απαιτείται για τη σωστή λειτουργία.

## Typeorm Module

Κάθε module εκτός από τους ελεγκτές, υπηρεσίες και αποθετήρια περιλαμβάνει και αρχεία TypeScript τα οποία ορίζουν κλάσεις οι οποίες αντιστοιχίζονται στους πίνακες της βάσης. Η αντιστοιχία αυτή γίνεται μέσω του module Typeorm το οποίο δέχεται έναν πίνακα με τα ονόματα αυτών των κλάσεων.

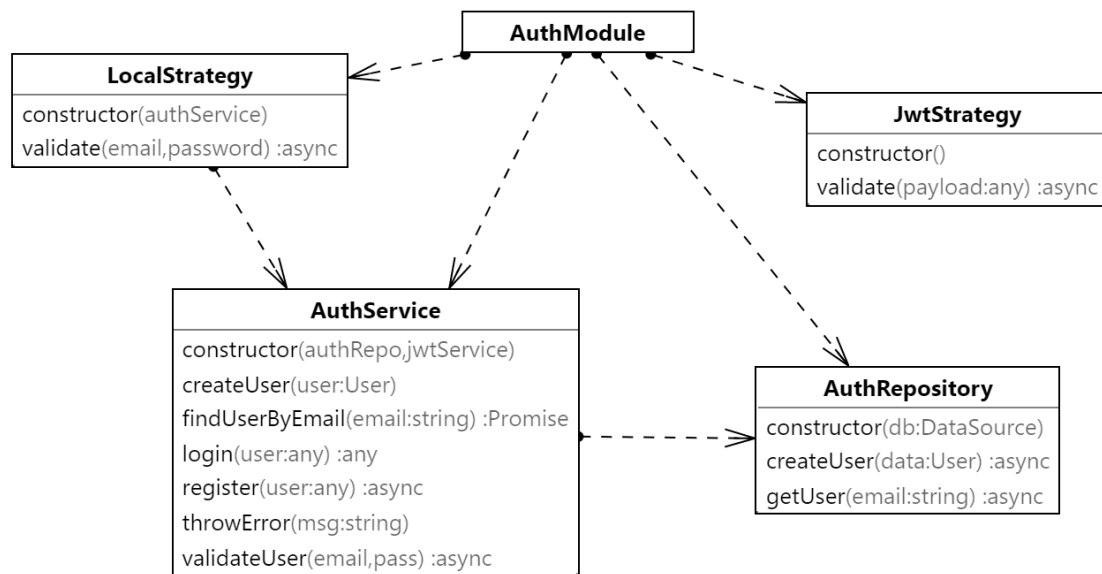
## Passport Module

Το Module αυτό παρέχει διεπαφές για διάφορες στρατηγικές αυθεντικοποίησης ενός χρήστη. Στην περίπτωση αυτή θα χρησιμοποιηθεί η τοπική στρατηγική (Local strategy) η οποία κάνει αυθεντικοποίηση με βάση το email και τον κωδικό. Κατά την επιτυχία επιστρέφει την πληροφορία του χρήστη κωδικοποιημένη ως ένα JWT token(json web token) που θα σταλεί στο επίπεδο παρουσίασης. Το module passport για κάθε νέο token, δημιουργεί και μια μοναδική υπογραφή την οποία αντιστοιχίζει σε έναν χρήστη χρησιμοποιώντας ένα κρυφό κλειδί. Μετά τη δημιουργία, το token θα στέλνεται με κάθε αίτημα του χρήστη στο επίπεδο λογικής και αν είναι έγκυρο, το αίτημα θα πραγματοποιείται. Για την καλύτερη ασφάλεια κάθε token περιλαμβάνει και ένα πεδίο το οποίο ορίζει την ημερομηνία λήξης.

## Auth Module

Στο module αυτό γίνεται η αρχικοποίηση της ενότητας Passport με τα κατάλληλα configurations. Έπειτα έχουμε υλοποίηση των μεθόδων αυθεντικοποίησης επεκτείνοντας τις κλάσεις που παρέχονται από το module Passport. Στην εικόνα 14 παρουσιάζεται ένα διάγραμμα με τις κλάσεις της ενότητας. Οι πιο σημαντικές κλάσεις του module είναι η κλάση JwtStrategy και LocalStrategy. Η κλάση LocalStrategy ορίζει τη μέθοδο αυθεντικοποίησης με βάση τον κωδικό χρήστη και το email κοιτώντας στη

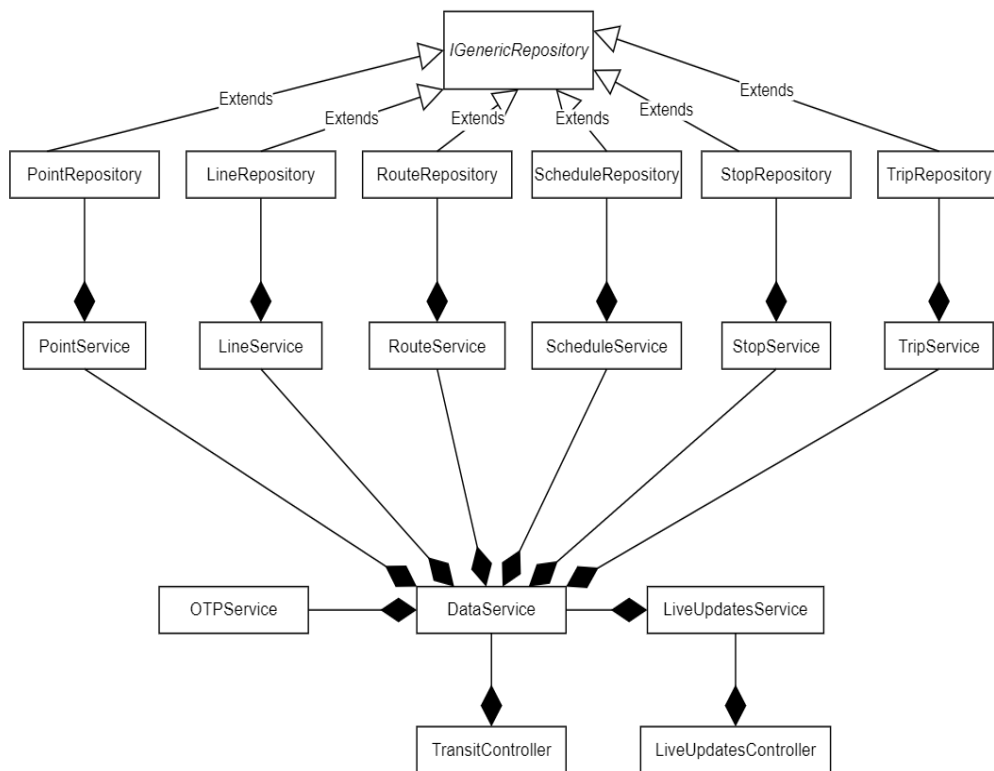
βάση και χρησιμοποιείται μόνο κατά το login. Η κλάση JwtStrategy χρησιμοποιείται για την προστασία του API και κάνει αυθεντικοποίηση με βάση το JWT token. Αυτές οι κλάσεις θα χρησιμοποιηθούν ως middleware από τους ελεγκτές του διακομιστή για κάθε αίτημα. Ένα middleware είναι μια μέθοδος που καλείται πριν την εξυπηρέτηση ενός αιτήματος. Έπειτα έχουμε και την υπηρεσία Auth η οποία επιστρέφει τις πληροφορίες χρήστη ως ένα JWT token με την μέθοδο login ή με τη μέθοδο register αν αυτός είναι έγκυρος. Στο module αυτό γίνεται και η δημιουργία του λογαριασμού χρήστη.



Εικόνα 14. Κλάσεις της ενότητας Auth

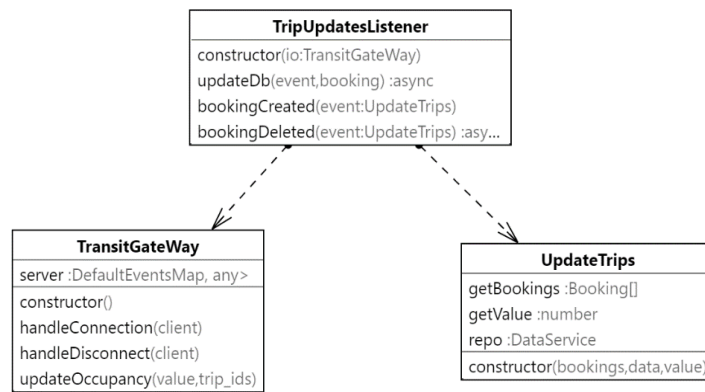
## Transit Module

Σε αυτό το module ορίζονται όλες οι κλάσεις που αφορούν τα δεδομένα του αστικού ΚΤΕΛ. Ποιο συγκεκριμένα οι ελεγκτές της ενότητας θα επιστρέφουν δεδομένα που αφορούν τις οντότητες: Stop, Line, Route, Point, Schedule, Trip καθώς και δεδομένα πραγματικού χρόνου. Η οργάνωση των κλάσεων παρουσιάζεται στην εικόνα 15.



Εικόνα 15. Κλάσεις της ενότητας Transit

Στο διάγραμμα της εικόνας 15 παρατηρούμε αρχικά ότι έχουμε ένα αποθετήριο για τις οντότητες που αφορούν τα δεδομένα του αστικού ΚΤΕΛ και ορίζονται σε αυτό το module. Για την καλύτερη οργάνωση έχει ορισθεί και μια γενική και αφηρημένη κλάση όπου γίνεται η υλοποίηση όλων των κοινών μεθόδων που αφορούν τις λειτουργίες της βάσης δεδομένων και δέχεται ως όρισμα το γενικό αποθετήριο που μας παρέχει το module Typeorm. Έπειτα έχουμε και μια υπηρεσία για κάθε οντότητα, η οποία καλεί τις μεθόδους των αποθετηρίων και χρησιμοποιείται από τους ελεγκτές. Έχουμε επίσης και τις υπηρεσίες OTP και LiveUpdates. Η υπηρεσία OTP επικοινωνεί με τον Open Trip Planner και επιστρέφει τις διαδρομές ή τις εκτιμήσεις αφίξεων τις οποίες υπολογίζει ο OTP. Η υπηρεσία LiveUpdatesService καλεί το endpoint του αστικού ΚΤΕΛ που αφορά τα δεδομένα πραγματικού χρόνου, καθώς και τον OTP μέσω της κλάσης DataService και επιστρέφει την εμπλουτισμένη πληροφορία πραγματικού χρόνου μέσω του ελεγκτή LiveUpdatesController. Ο ελεγκτής αυτός εκθέτει και ένα end point το οποίο στέλνει τα αρχεία protobuf στον OTP. Το module Transit περιλαμβάνει και τις κλάσεις που φαίνονται στην εικόνα 16.

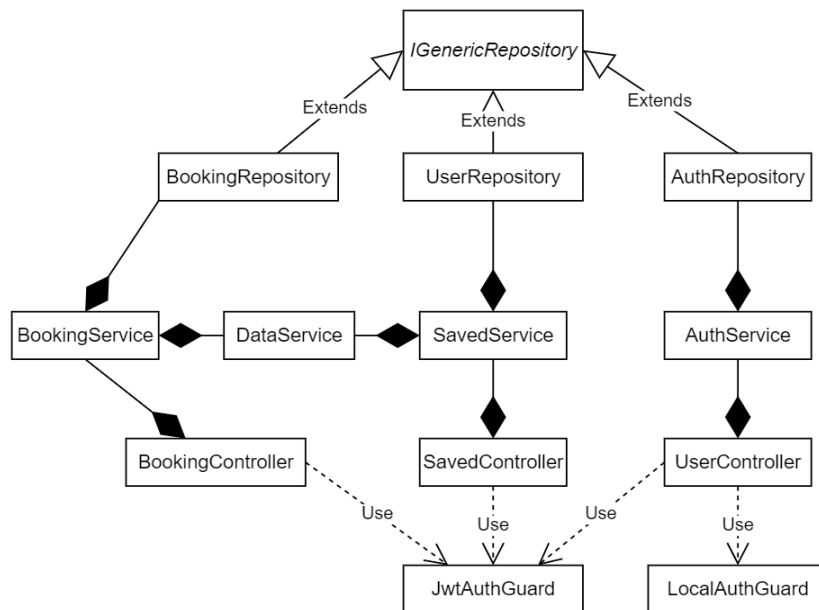


Εικόνα 16. Κλάσεις για την ανανέωση του πελάτη σε πραγματικό χρόνο

Η κλάση `UpdateTrips` ορίζει ένα γεγονός το οποίο περνάει στον ακροατή `TripUpdates`. Ο ακροατής χρησιμοποιεί ένα ξεχωριστό νήμα για να αναβαθμίσει τον πίνακα `trip` της βάσης σε περίπτωση που υπάρχει μια εισαγωγή ή διαγραφή κράτησης. Έπειτα, με χρήση των `web sockets` και μέσω της κλάσης `TransitGateWay`, ενημερώνει τους χρήστες που είναι `online` για την αλλαγή. Το γεγονός αυτό θα συμβαίνει στο `module` του χρήστη κατά την αποθήκευση ή διαγραφή κράτησης. Η διαδικασία αυτή έχει το πλεονέκτημα ότι ο χρήστης λαμβάνει άμεσα την ανταπόκριση από τον διακομιστή χωρίς καθυστερήσεις.

## User Module

Τέλος στο `user module` ορίζονται όλα τα αποθετήρια, ελεγκτές, υπηρεσίες και οντότητες που αφορούν τον χρήστη. Πιο συγκεκριμένα, οι ελεγκτές του `module` αυτού θα χειρίζονται τις πληροφορίες που αφορούν τις οντότητες `Booking`, `User` και την αποθηκευμένη πληροφορία του χρήστη που αφορά τις οντότητες `Route`, `Stop`. Όπως φαίνεται και στην εικόνα 13 το `module` αυτό χρησιμοποιεί το `module Transit` το οποίο εξάγει την κλάση `DataService` και το `module Auth` το οποίο εξάγει την κλάση `AuthService`. Στην εικόνα 17 φαίνεται το διάγραμμα με τις κλάσεις που το απαρτίζουν.



Εικόνα 17. Κλάσεις της ενότητας χρήστη

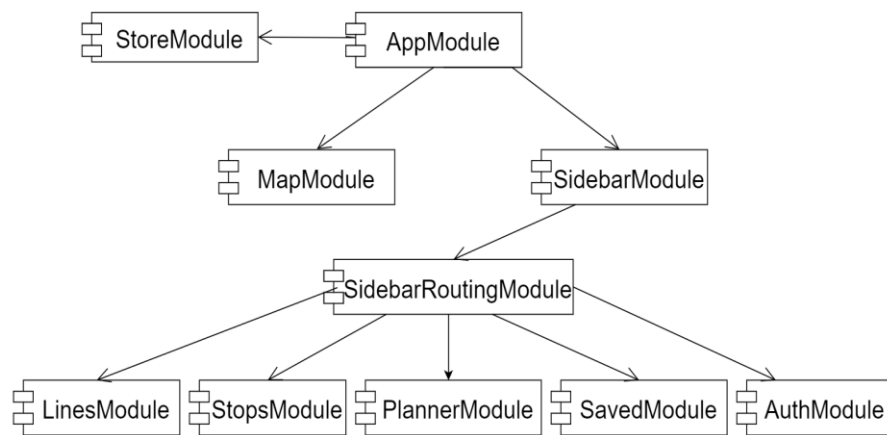
Στο module αυτό έχουμε 2 αποθετήρια. Το αποθετήριο BookingRepository χειρίζεται τις λειτουργίες που αφορούν τις κρατήσεις ενώ το αποθετήριο UserRepository έχει τις ερωτήσεις SQL που αφορούν την αποθηκευμένη πληροφορία του χρήστη. Τα 2 αποθετήρια αυτά επεκτείνουν το γενικό αποθετήριο. Έπειτα έχουμε και από μια υπηρεσία για κάθε αποθετήριο η οποίες τα καλούν. Παρατηρούμε ότι οι υπηρεσίες αυτές χρησιμοποιούν και την κλάση DataService η οποία ορίζεται στο TransitModule. Στη συνέχεια ορίζονται και οι ελεγκτές του module. Παρατηρούμε επίσης, ότι οι ελεγκτές χρησιμοποιούν τις κλάσεις LocalAuthGuard και JwtAuthGuard, οι οποίες επεκτείνουν τις κλάσεις που παρέχονται από την ενότητα Passport. Οι κλάσεις αυτές μέσω του Passport θα καλέσουν τα middleware που ορίσαμε στην ενότητα Auth για την προστασία των ελεγκτών. Ο ελεγκτής UserController χειρίζεται τις λειτουργίες εισόδου και εγγραφής στην εφαρμογή μέσω της υπηρεσίας AuthService.

## 5.4. Υλοποίηση επιπέδου παρουσίασης

Τέλος, μετά την υλοποίηση του επιπέδου λογικής ακολουθεί η σχεδίαση και υλοποίηση του επιπέδου παρουσίασης(front-end).

### 5.4.1. Ενότητες(Modules)

Παρόμοια με το επίπεδο λογικής χωρίζουμε το επίπεδο παρουσίασης σε modules, τα οποία παρουσιάζονται στην εικόνα 18.



Εικόνα 18. Modules του επιπέδου παρουσίασης

## App Module

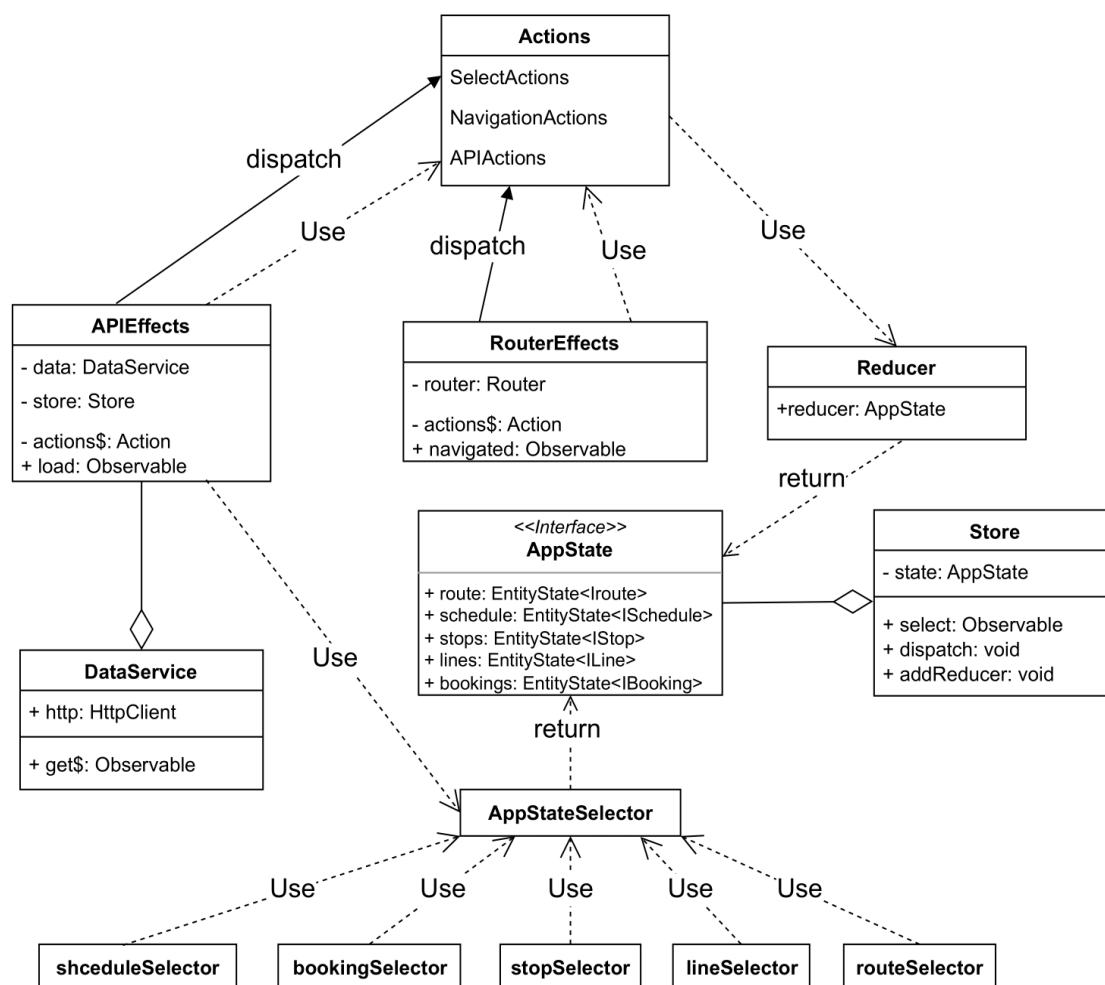
Το App Module είναι το σημείο εκκίνησης της εφαρμογής. Αρχικοποιεί τα 2 κύρια modules που αφορούν τον χάρτη για την οπτικοποίηση των δεδομένων καθώς και ένα μενού πλοήγησης. Αρχικοποιεί επίσης και την κατάσταση(Store) της εφαρμογής και όλες τις κοινές κλάσεις.

## Store Module

Σε αυτό το module ορίζουμε τη καθολική κατάσταση της εφαρμογής, σύμφωνα με το μοτίβο Redux. Η κατάσταση θα ορισθεί ως ένα singleton αντικείμενο το οποίο θα διατηρεί τα δεδομένα και θα μοιραστεί σε όλα τα μέρη του επιπέδου παρουσίασης. Στην εικόνα 19 φαίνεται το διάγραμμα με τις κλάσεις της κατάστασης. Η κλάση Store διατηρεί την κατάσταση με τα δεδομένα καθώς και τις μεθόδους για την τροποποίηση της. Η κατάσταση έχει ορισθεί ως ένα interface με πεδία που έχουν τον τύπο του γενικού interface EntityState. Παρατηρούμε ότι έχουμε έναν κύριο selector ο οποίος επιλέγει την κεντρική κατάσταση και χρησιμοποιείται από τους υπόλοιπους selectors για να εξάγουν τις καταστάσεις των οντοτήτων της εφαρμογής. Στη συνέχεια έχουμε και 2 κλάσεις που ορίζουν τα effects. Βλέπουμε ότι η κλάση APIEffects χρησιμοποιεί τον γενικό selector. Παρόλου που δεν είναι υπεύθυνη για την τροποποίηση της κατάστασης, υπάρχουν περιπτώσεις που πρέπει να γνωρίζει την κατάσταση με βάση την οποία θα ενεργοποιήσει ή όχι τις αντίστοιχες ενέργειες. Η κλάση routerEffects παρατηρεί την κατάσταση πλοήγησης του χρήστη. Όπως βλέπουμε και στην εικόνα 19 έχουμε και έναν reducer ο οποίος 'ακούει' σε actions και επιστρέφει ένα καινούργιο αντικείμενο κατάστασης. Τέλος έχουμε και την υπηρεσία δεδομένων η οποία κάνει requests στον διακομιστή. Πλέον κάθε component του επιπέδου παρουσίασης μπορεί να προμηθευτεί

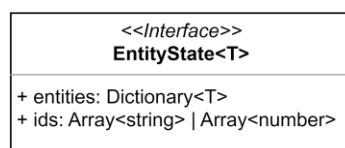


το αντικείμενο Store και να λάβει τα δεδομένα που χρειάζεται μέσω των selectors. Το interface AppState ορίζει και άλλες πληροφορίες που χρησιμοποιούνται για την επικοινωνία των κομματιών της εφαρμογής και παραλείπονται από το διάγραμμα.



Εικόνα 19. Κλάσεις και συναρτήσεις της κατάστασης.

Πολύ σημαντικό είναι να αναφερθούμε και στο πως αποθηκεύουμε τα δεδομένα της κατάστασης. Κάθε πεδίο της κατάστασης έχει τύπο EntityState. Με το interface αυτό οργανώνουμε τα δεδομένα με χρήση ενός λεξικού έτσι ώστε η τροποποίηση και η εύρεση τους να γίνεται γρήγορα. Στην εικόνα 20 βλέπουμε αναλυτικά το interface EntityState.

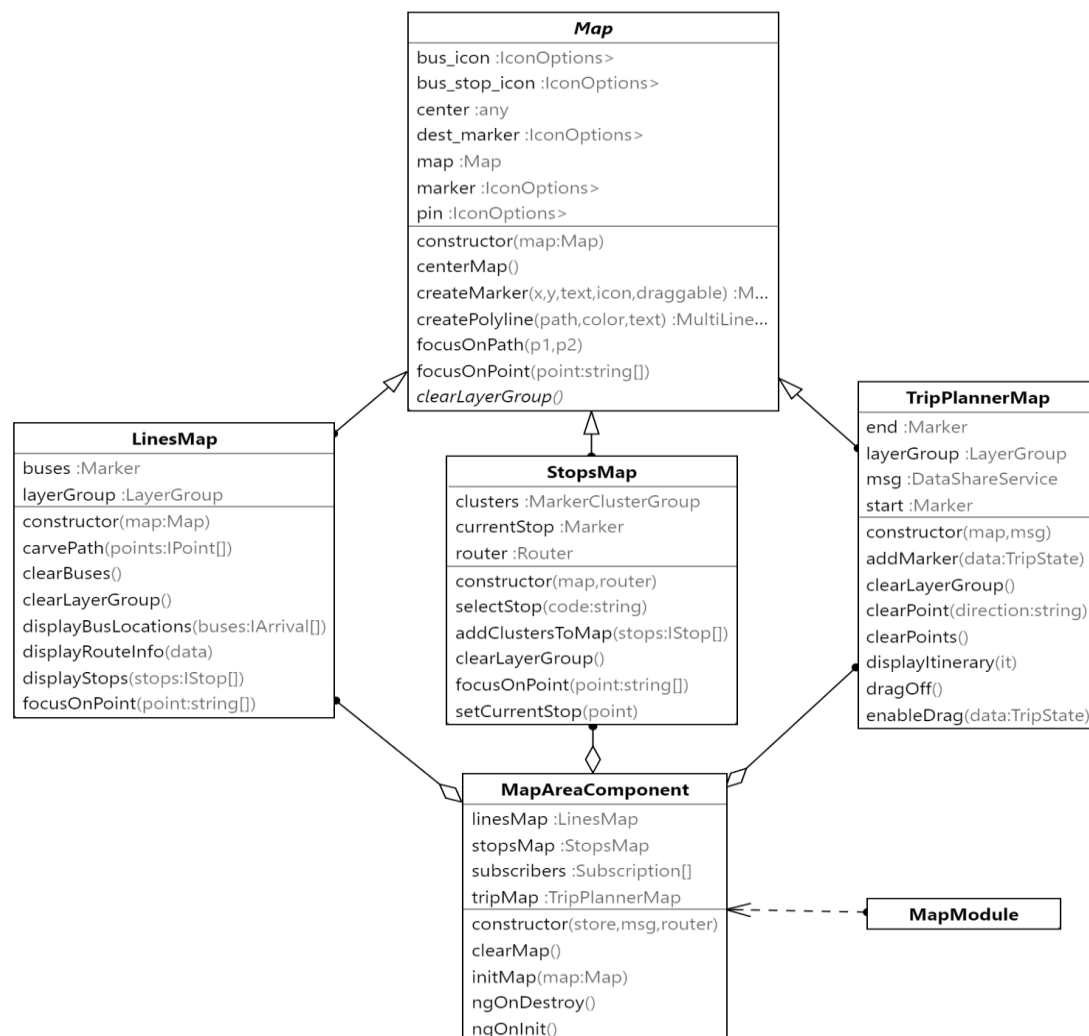


Εικόνα 20. Το γενικό interface EntityState

Όπως βλέπουμε στην εικόνα 20 ορίζουμε ένα πίνακα με τα ids και ένα λεξικό. Το λεξικό αποθηκεύει τα δεδομένα που λαμβάνονται από το επίπεδο λογικής.

## Map Module

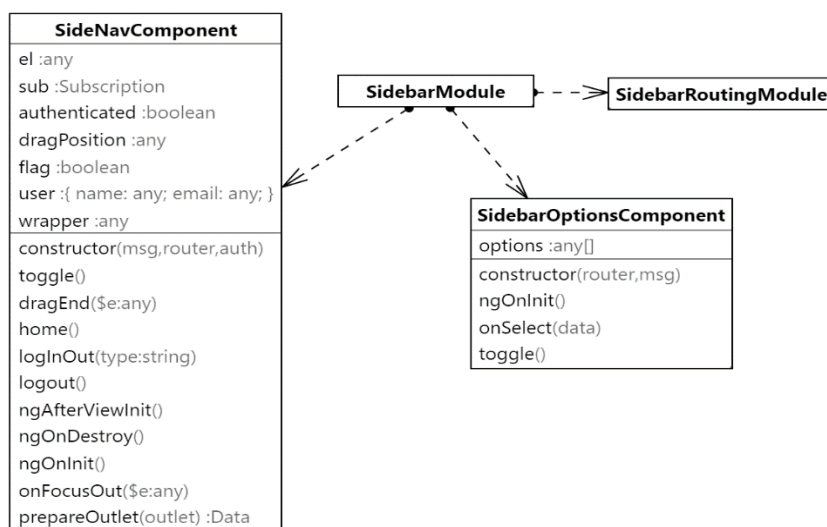
Το Map module περιγράφει το κομμάτι της διεπαφής χρήστη που αφορά τους χάρτες. Επικοινωνεί με τα υπόλοιπα κομμάτια της εφαρμογής μέσω της κλάσης Store με χρήση των κατάλληλων selectors καθώς και των κατάλληλων actions που παράγονται με βάση την αλληλεπίδραση του χρήστη με τον χάρτη. Όπως φαίνεται και στην εικόνα 21 το Map module αποτελείται από έναν component καθώς και τρεις κλάσεις ελεγκτές οι οποίες επεκτείνουν τη γενική και αφηρημένη κλάση Map. Οι ελεγκτές αυτοί υλοποιούν τις μεθόδους για την οπτικοποίηση των δεδομένων πάνω στον χάρτη καλώντας τις μεθόδους της βιβλιοθήκης leaflet.



Εικόνα 21. Κλάσεις του Map Module

## Sidebar Module

Το module sidebar αφορά το βασικό και το δευτερεύον menu πλοήγησης της εφαρμογής. Όπως παρατηρούμε στην εικόνα 18, μέσω του αρχείου πλοήγησης (SidebarRoutingModule) ορίζονται και κάποια ακόμα modules. Αυτά τα modules εφαρμόζουν την τεχνική της αργοπορημένης φόρτωσης(lazy loading) και θα φορτωθούν μόλις ο χρήστης πλοηγηθεί στις λειτουργίες που περιγράφουν. Κάθε ένα από αυτά εστιάζει σε ένα κύριο use case. Το διάγραμμα κλάσεων που απαρτίζουν αυτό το module φαίνεται στην εικόνα 22.



Εικόνα 22. Components του Sidebar Module

Τα υπόλοιπα modules που ορίζονται μέσω του Sidebar module υλοποιούν τις περιπτώσεις χρήσης της εφαρμογής και έχουν ως κύρια αρμοδιότητα την παρουσίαση δεδομένων στον χρήστη. Επικοινωνούν με την υπόλοιπη εφαρμογή μέσω του αντικειμένου Store. Λαμβάνουν τα δεδομένα που χρειάζονται για παρουσίαση και έπειτα δημιουργούν ενέργειες με βάση τις επιλογές του χρήστη. Πιο συγκεκριμένα:

- Lines Module: Υλοποιεί τη περίπτωση χρήσης 3(UC3).
- Stops Module: Υλοποιεί τη περίπτωση χρήσης 4(UC4).
- Planner Module: Υλοποιεί τις περιπτώσεις χρήσης 5, 6(UC5, UC6).
- Auth Module: Υλοποιεί τις περιπτώσεις χρήσης 1, 2(UC1, UC2).

Εκτός από τα κύρια modules έχει ορισθεί και ένα κοινό module(SharedModule) το οποίο ορίζει και εξάγει κάποια κοινά components. Η μόνη αρμοδιότητά τους είναι η παρουσίαση δεδομένων και προώθηση τους, σύμφωνα με τις ενέργειες του χρήστη, προς τους γονικούς components που τους χρησιμοποιούν.

### 5.4.2. Ασφάλεια

Όπως και στο επίπεδο λογικής, χρειαζόμαστε ένα τρόπο ο οποίος θα εισάγει ασφάλεια στην εφαρμογή. Ανάλογα με το αν ο χρήστης είναι συνδεδεμένος ή όχι, θέλουμε να εφαρμόσουμε τους κατάλληλους περιορισμούς. Για τον σκοπό αυτό έχει ορισθεί η singleton υπηρεσία Auth Guard η οποία φαίνεται και στην εικόνα 23.

AuthGuardService
constructor(auth,router)
canActivate():boolean

Εικόνα 23. Υπηρεσία ασφαλείας του επιπέδου παρουσίασης.

Η κλάση αυτή υλοποιεί τη διεπαφή canActivate και χρησιμοποιείται κατά τη διάρκεια πλοήγησης στις σελίδες. Ελέγχει αν ένας χρήστης είναι συνδεδεμένος με βάση το JWT token και αν όχι μεταφέρει τον χρήστη στην σελίδα της εισόδου.

### 5.4.3. Επικοινωνία σε πραγματικό χρόνο

Το επίπεδο παρουσίασης εκτός από τα αιτήματα HTTP λαμβάνει και πληροφορίες μέσω σύνδεσης socket. Συγκεκριμένα θέλουμε να αλλάζουμε τις διαθέσιμες θέσεις άμεσα με κάθε κράτηση. Αυτό επιτυγχάνεται με την βοήθεια της κλάσης που φαίνεται στην εικόνα 24.

SocketIOService
socket:DefaultEventsMap>
constructor(store:Store)

Εικόνα 24. Ακροατής του επιπέδου παρουσίασης.

Η κλάση αυτή αρχικοποιεί έναν ακροατή ο οποίος ακούει σε ένα συγκεκριμένο event. Μόλις αντιληφθεί μια νέα μετάδοση στέλνει ένα action για την αναβάθμιση του αντικειμένου Store.

### 5.4.4. Testing

Εκτός από τα βασικά αρχεία, το command line της Angular παράγει και ένα αρχείο TypeScript(spec.ts) για testing για κάθε component. Αυτά τα τεστ θα χρησιμοποιηθούν για τον έλεγχο της σωστής αρχικοποίησης όλων των εξαρτήσεων των κλάσεων που αφορούν τα components της εφαρμογής φτιάχνοντας ένα mock αντικείμενο για κάθε εξάρτηση. Παρακάτω δίνεται ένα παράδειγμα test καθώς και το αποτέλεσμα όλων των τεστ που έτρεξαν μέσω της βιβλιοθήκης karma.

```

describe('RouteDetailsComponent', () => {
  let component: RouteDetailsComponent;
  let fixture: ComponentFixture<RouteDetailsComponent>;

  beforeEach(async () => {
    const a = setup().default();
    await TestBed.configureTestingModule({
      imports: [
        HttpClientModule,
        RouterTestingModule,
        StoreModule.forRoot({}),
      ],
      declarations: [ RouteDetailsComponent ]
    }).compileComponents();

    fixture = TestBed.createComponent(RouteDetailsComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });

  it('when ngOnInit is called it should', () => {
    // arrange
    const { build } = setup().default();

```

## Karma v 6.4.1 - connected; test: complete;

Chrome 109.0.0.0 (Windows 10) is idle

 **Jasmine** 4.2.0

.....

39 specs, 0 failures, randomized with seed 58286

```

RegisterComponent
  • should create

SidebarOptionsComponent
  • should create

BusEntityComponent
  • should create

AppComponent
  • should have as title 'cityBus'
  • should create the app

MainInputComponent
  • should create

StopSliderComponent
  • should create

ToasterComponent
  • should create

```

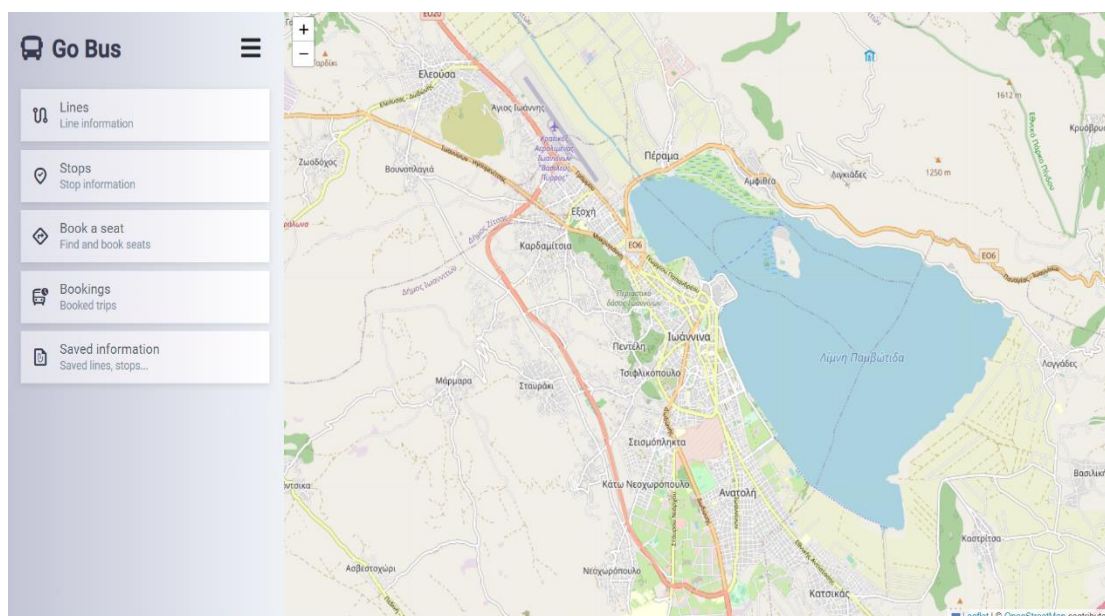
Εικόνα 25. Τεστ του επιπέδου παρουσίασης

## Κεφάλαιο 6. Διεπαφή χρήστη

Σε αυτό το κεφάλαιο θα δοθεί μια σύντομη περιγραφή των βημάτων για τη σχεδίαση της διεπαφής χρήστη καθώς και μια παρουσίαση των περιπτώσεων χρήσης.

### 6.1. Σχεδιαστικά εργαλεία

Η σχεδίαση της διεπαφής χρήστη έγινε με τη χρήση CSS και HTML σε συνδυασμό με τη βιβλιοθήκη Angular Material. Η βιβλιοθήκη αυτή παρέχει ένα μεγάλο σύνολο από components και animations τα οποία στοχεύουν στην αποκρισιμότητα, στη ταχύτητα και στην ασφάλεια. Παράλληλα χρησιμοποιήθηκαν ευρέως τα directives της Angular. Τα directives αυτά επεκτείνουν τα πεδία της HTML με καινούργια σύνταξη με στόχο τη διευκόλυνση της διαχείρισης και τη δημιουργίας των σελίδων. Μεγάλη προσοχή δόθηκε στη αποκρισιμότητα της εφαρμογής. Αυτό σημαίνει ότι θα πρέπει να αποκρίνεται ανάλογα με τη συσκευή στην οποία χρησιμοποιείται. Αυτός ήταν και ο κύριος παράγοντας ο οποίος επηρέασε τον τρόπο σχεδιασμού. Παρακάτω φαίνεται η αρχική όψη.



Εικόνα 26. Η διεπαφή χρήστη

Όπως φαίνεται στην εικόνα 26 η διεπαφή αποτελείται από 2 κύρια μέρη. Το μενού πλοήγησης και τους χάρτες της εφαρμογής. Η ιδέα είναι ότι σε μικρές συσκευές, όπως κινητά, οι χάρτες θα προσαρμοστούν στο μενού πλοήγησης για εξοικονόμηση χώρου.

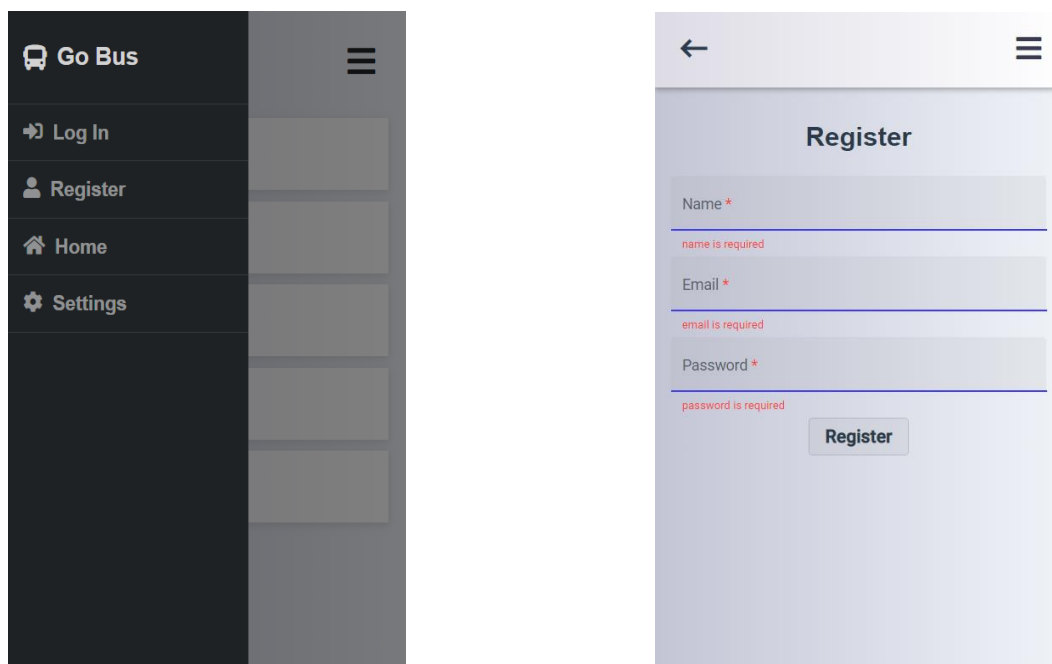
Στη συνέχεια θα δοθεί μια περιήγηση στην εφαρμογή από τη οπτική γωνιά ενός κινητού τηλεφώνου.

## 6.2. Περιήγηση στις περιπτώσεις χρήσης

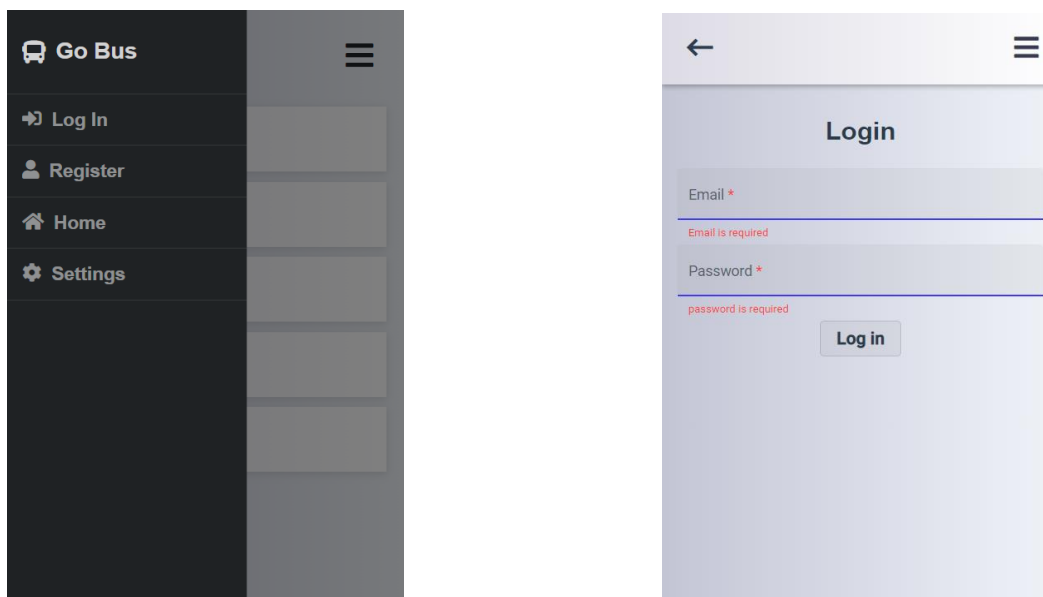
Στην ενότητα αυτή θα δοθούν παραδείγματα χρήσης στην εφαρμογή με βάση τις περιπτώσεις χρήσης.

### Οθόνες εγγραφής και εισόδου στο σύστημα (UC1, UC2)

Στις εικόνες 27, 28 βλέπουμε τη διαδικασία εισόδου και εγγραφής στην εφαρμογή. Ο χρήστης πατώντας στο εικονίδιο που βρίσκεται πάνω δεξιά ανοίγει το δευτερεύον μενού πλοήγησης. Έπειτα, ανάλογα με το αν θέλει να συνδεθεί ή να δημιουργήσει καινούριο λογαριασμό, πατάει στο κουμπί 'Login' ή 'Register'.



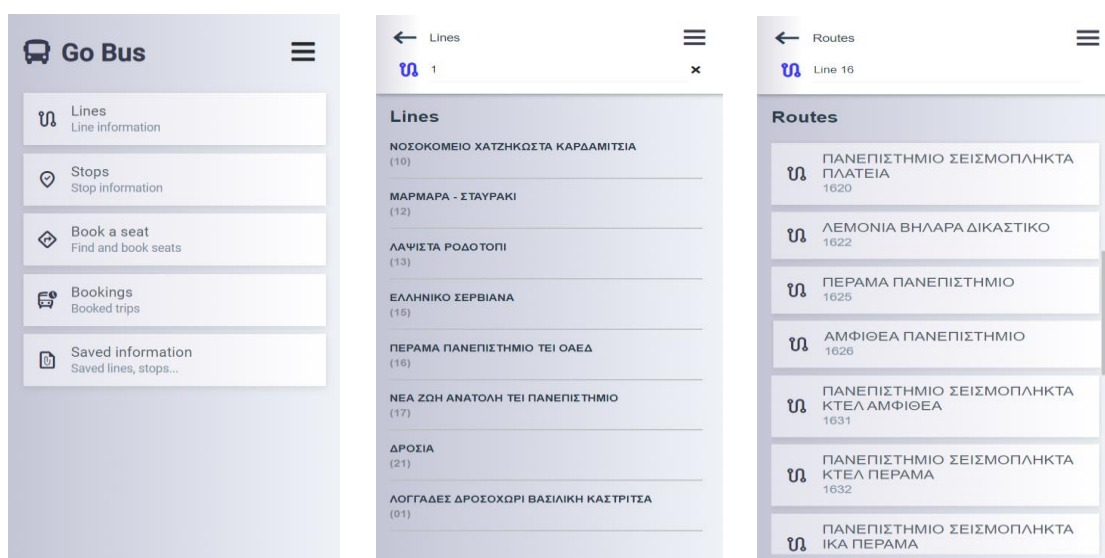
Εικόνα 27. Οθόνη εγγραφής



Εικόνα 28. Οθόνη εισόδου

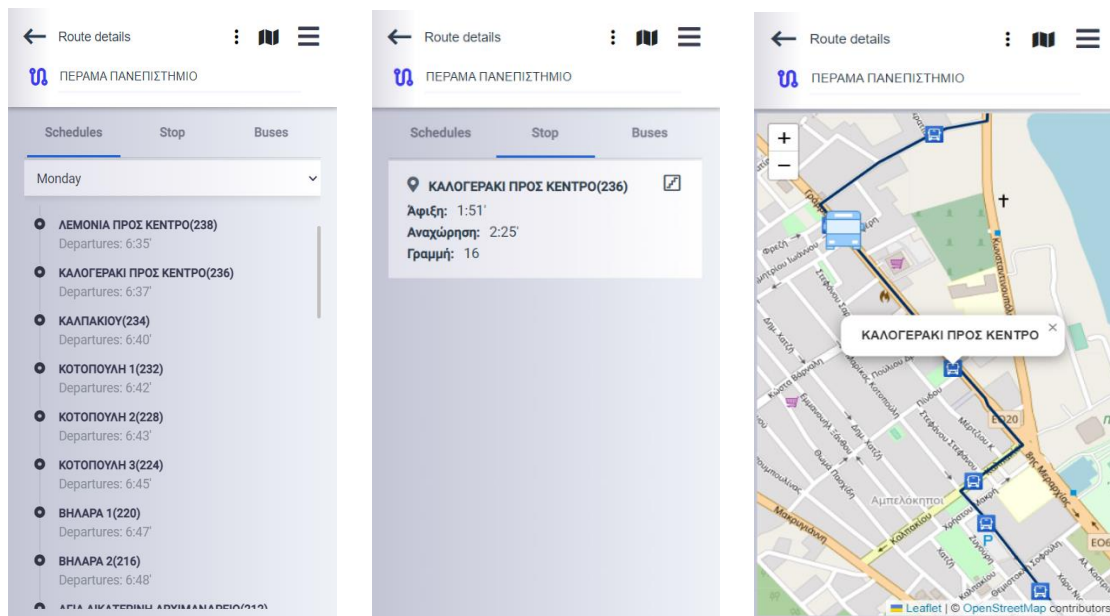
Με την επιλογή του χρήστη το σύστημα μεταφέρεται στην αντίστοιχη οθόνη για την συμπλήρωση των απαραίτητων πεδίων. Όπως βλέπουμε στις εικόνες 27 και 28 έχουμε μηνύματα ελέγχου που ειδοποιούν τον χρήστη σε περίπτωση ύπαρξης σφάλματος. Μόλις γίνει η είσοδος, το δευτερεύον μενού εμφανίζει το όνομα χρήστη πάνω πάνω καθώς και την επιλογή εξόδου.

## Εμφάνιση πληροφοριών διαδρομής (UC3)



Εικόνα 29. Πληροφορίες γραμμών

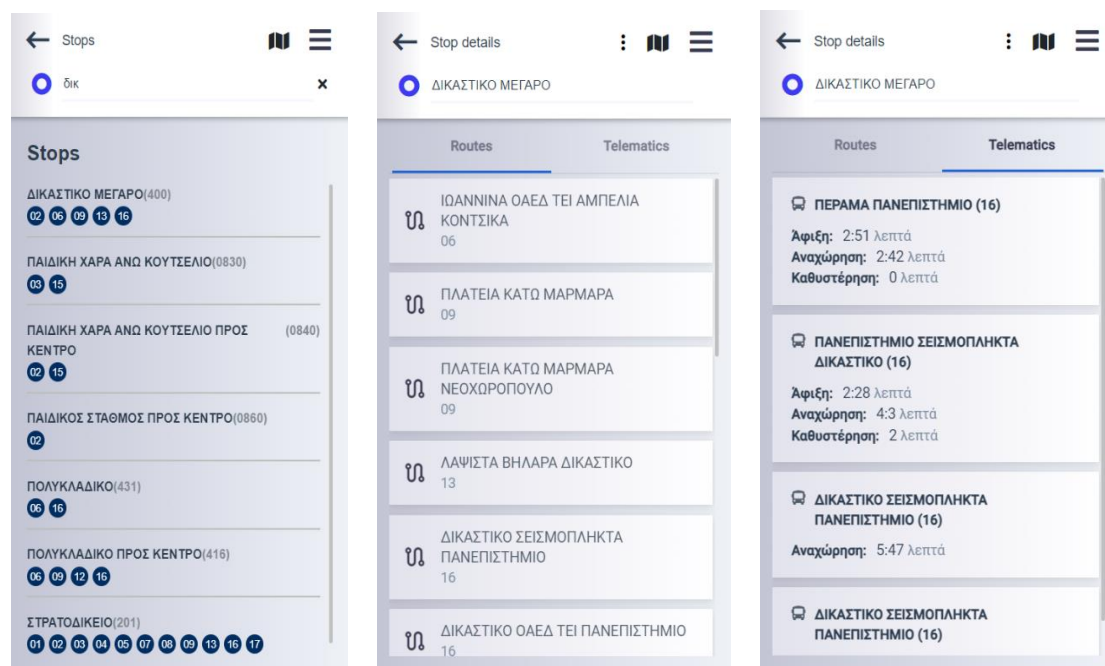




Εικόνα 30. Πληροφορίες διαδρομής

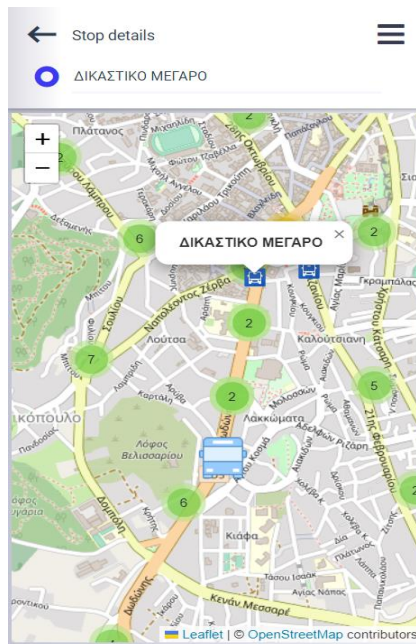
Στις εικόνες 29, 30 βλέπουμε τα βήματα της περίπτωσης. Για την περίπτωση αυτή ο χρήστης πατάει στο κουμπί Lines και μεταφέρεται στην λίστα με τις γραμμές όπου εκεί μπορεί να επιλέξει μια διαδρομή. Έπειτα, επιλέγοντας μια διαδρομή, εμφανίζονται οι επιλογές, 'Schedules', 'Stop' και 'Buses'. Ο χρήστης βρισκόμενος στην επιλογή Schedules βλέπει τις στάσεις με την σειρά. Πατώντας στο μενού με τις μέρες, μπορεί να επιλέξει κάποια μέρα και το σύστημα θα βγάλει προσεγγιστικά τις αφίξεις για κάθε στάση με βάση το πρόγραμμα του αστικού ΚΤΕΛ. Αν δεν υπάρχουν αφίξεις για κάποια μέρα εμφανίζεται κατάλληλο μήνυμα. Πατώντας πάνω σε μια στάση γίνεται αυτόματα μεταφορά στην επιλεγμένη στάση όπου εκεί εμφανίζονται οι εκτιμήσεις αφίξεων και αναχωρήσεων σε πραγματικό χρόνο. Αν δεν υπάρχει επιλεγμένη στάση, το σύστημα εμφανίζει μήνυμα. Πατώντας στα λεωφορεία εμφανίζεται μια λίστα με τα ενεργά λεωφορεία της διαδρομής. Έπειτα πατώντας στο εικονίδιο του χάρτη η εφαρμογή ανοίγει τον χάρτη και εστιάζει στο μονοπάτι της διαδρομής. Αν υπάρχει επιλεγμένη στάση, ο χάρτης θα εστιάσει σε αυτήν όπως φαίνεται και στην εικόνα 30. Όπως βλέπουμε έχουμε και την ακριβή θέση του λεωφορείου η οποία ανανεώνεται ανά μικρά χρονικά διαστήματα.

## Εμφάνιση πληροφοριών στάσης (UC4)



Εικόνα 31. Πληροφορίες στάσης

Ο χρήστης βρισκόμενος στην αρχική σελίδα πατάει στο κουμπί 'Stops' και μεταφέρεται στη λίστα με τις στάσεις. Για κάθε στάση εμφανίζεται και η λίστα των γραμμών που την διασχίζουν. Με την επιλογή της στάσης έχουμε μια λίστα των διαδρομών και μια λίστα με τη πληροφορία πραγματικού χρόνου, καθώς και τις επόμενες αναχωρήσεις με βάση το πρόγραμμα. Όπως βλέπουμε στην εικόνα 31, το σύστημα εμφανίζει και μια εκτίμηση της καθυστέρησης του λεωφορείου η οποία υπολογίζεται με βάση την εκτίμηση αναχώρησης σε πραγματικό χρόνο και την στατική ώρα αναχώρησης που αναγράφεται στο ημερήσιο πρόγραμμα. Στην παρούσα φάση το πεδίο αυτό είναι πειραματικό. Ιδανικά θέλουμε να έχουμε αντιστοιχία του πίνακα live\_data και του πίνακα schedule της βάσης. Για την περίπτωση αυτή γίνεται εύρεση της κοντινότερης στατικής ώρας που έχει το δρομολόγιο για κάποια μέρα σε σχέση με την πρόβλεψη. Ο χρήστης επίσης μπορεί να επιλέξει μια στάση με βάση τον χάρτη. Πατώντας την ένδειξη του χάρτη εμφανίζονται οι στάσεις ομαδοποιημένες σε γειτονιές(clusters). Με την επιλογή μιας στάσης το σύστημα εμφανίζει τις αντίστοιχες πληροφορίες. Η διαδικασία αυτή παρουσιάζεται και στην εικόνα 32. Πατώντας πάνω σε ένα λεωφορείο φαίνεται και η γραμμή που ακολουθεί.



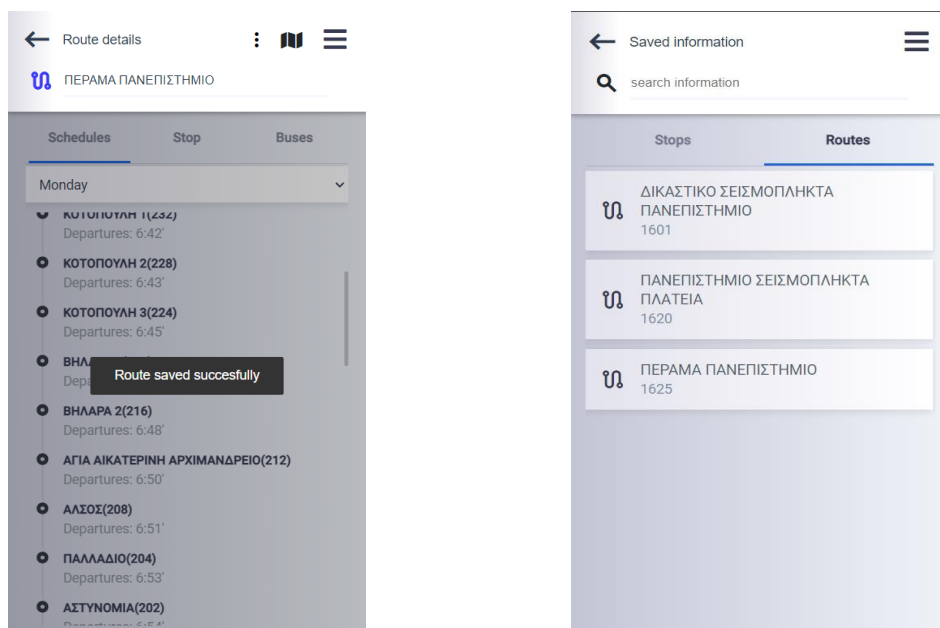
Εικόνα 32. Χάρτης των στάσεων

## Αποθήκευση στάσης και διαδρομής



Εικόνα 33. Επιλογή αποθήκευσης

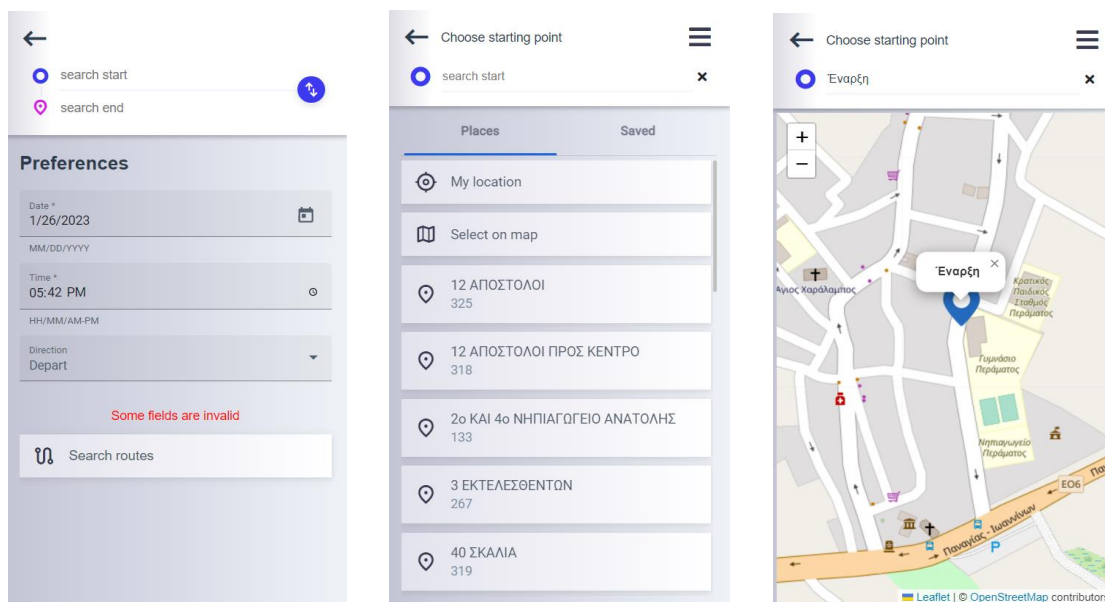
Αυτές οι περιπτώσεις απαιτούν τη σύνδεση του χρήστη. Για την αποθήκευση μιας στάσης ή διαδρομής ο χρήστης πρέπει να πλοηγήθει στις οθόνες με τις πληροφορίες διαδρομών ή στάσεων. Πατώντας στο μικρό μενού πάνω πάνω που φαίνεται και στην εικόνα 33, εμφανίζεται η επιλογή αποθήκευσης ή διαγραφής ανάλογα με την περίπτωση. Μετά από μια επιλογή του χρήστη, εμφανίζεται μήνυμα ολοκλήρωσης. Για τη προβολή των αποθηκευμένων, ο χρήστης πρέπει να ανοίξει την σελίδα με τις αποθηκευμένες πληροφορίες πατώντας στην ένδειξη 'Saved information' από το κεντρικό μενού. Στην εικόνα 34 έχουμε την αποθήκευση μιας διαδρομής καθώς και την εμφάνιση όλων των αποθηκευμένων διαδρομών του χρήστη. Πατώντας πάνω σε μια διαδρομή το σύστημα θα μεταβεί στη σελίδα με τις λεπτομέρειες της. Αν ο χρήστης δεν είναι συνδεδεμένος τότε η ένδειξη αποθήκευσης δεν εμφανίζεται.



Εικόνα 34. Αποθήκευση και εμφάνιση διαδρομής

## Κράτηση θέσης (UC5)

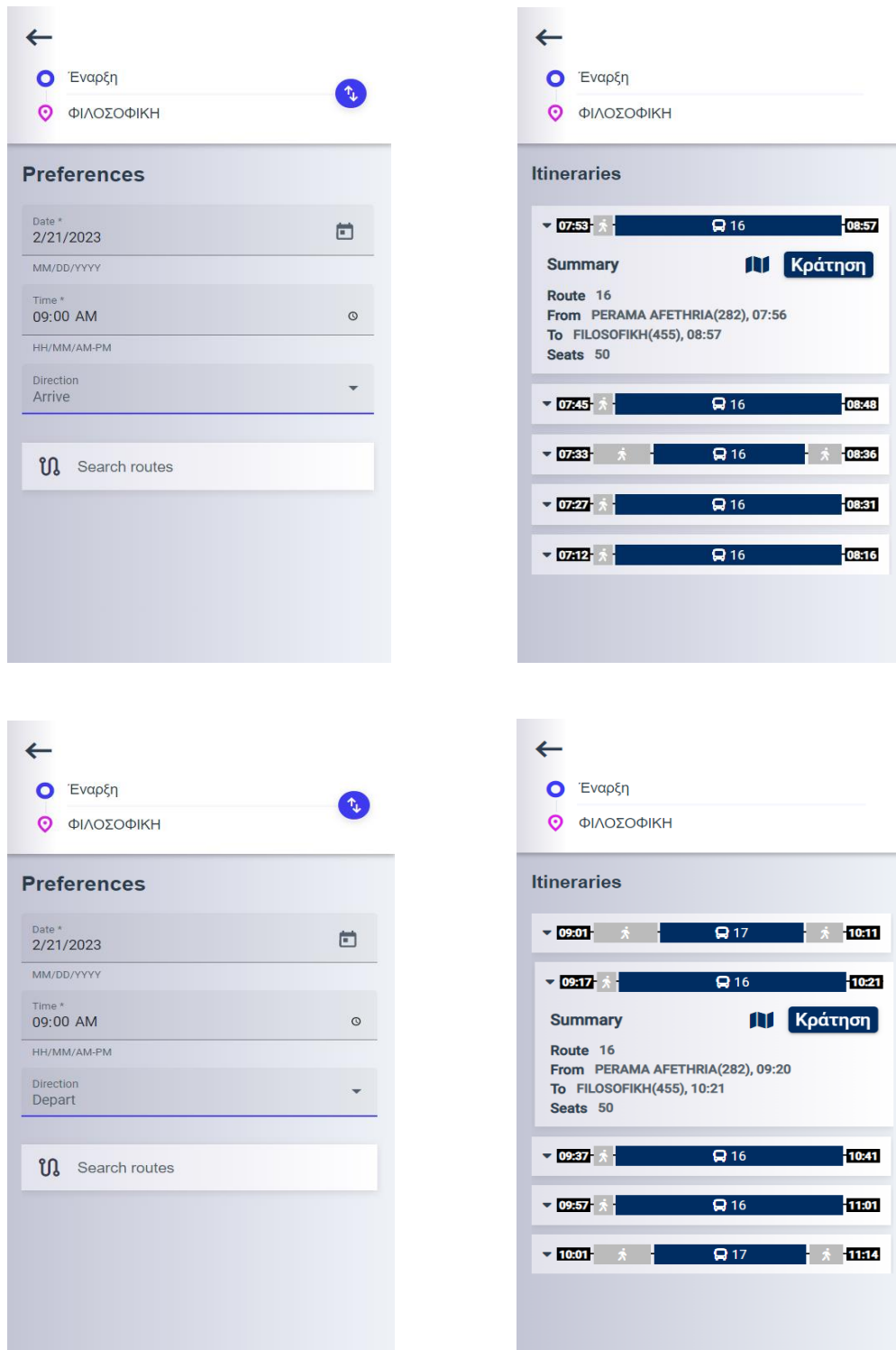
Ξεκινώντας από το κεντρικό μενού ο χρήστης επιλέγει την ένδειξη 'Book a seat' και η εφαρμογή ανοίγει τη φόρμα για την συμπλήρωση των απαιτούμενων πεδίων.



Εικόνα 35. Φόρμα και οθόνη επιλογής σημείου έναρξης

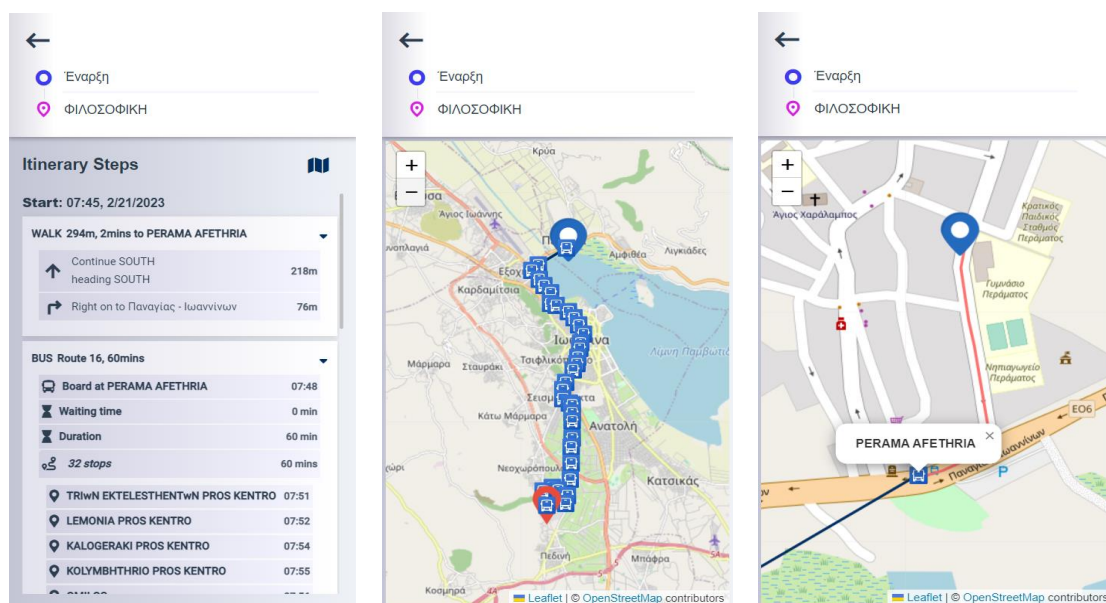
Για την επιλογή σημείων ο χρήστης πρέπει να πατήσει στα πεδία πάνω πάνω. Όπως βλέπουμε στην εικόνα 35 ο χρήστης μπορεί αναζητήσει μια στάση, να επιλέξει μια αποθηκευμένη στάση, να επιλέξει την τοποθεσία του, ή να ανοίξει τον χάρτη και να

μετακινήσει το σημείο στην τοποθεσία που επιθυμεί. Στις δύο τελευταίες περιπτώσεις το σύστημα θα επιλέξει τη καταλληλότερη στάση με βάση τα προτεινόμενα δρομολόγια.



Εικόνα 36. Εύρεση διαδρομών με βάση τον χρόνο άφιξης ή αναχώρησης

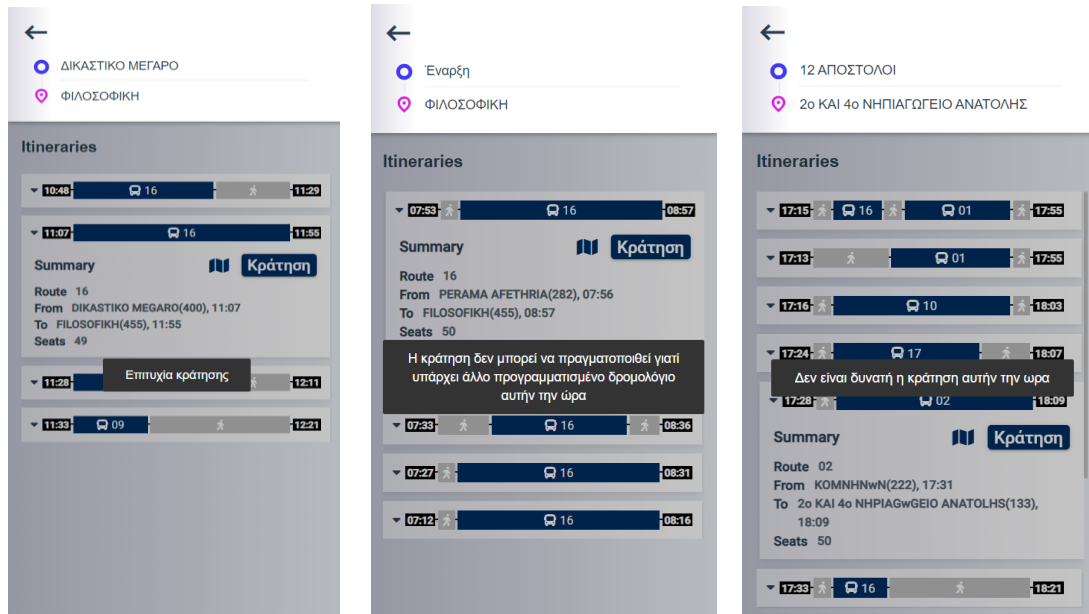
Στην εικόνα 36 παρουσιάζεται η διαδικασία εύρεσης διαδρομών. Ο χρήστης συμπληρώνει την έναρξη, τον τερματισμό και αν θέλει βάζει ώρα και ημερομηνία. Έπειτα υπάρχει η επιλογή προσδιορισμού αν η ώρα θα ορίζει την άφιξη ή την αναχώρηση. Όπως βλέπουμε στην εικόνα 36, έχουμε διαφορετικές προτάσεις δρομολογίων. Μπορεί να μην υπάρχει απευθείας δρομολόγιο με βάση τις εισόδους του χρήστη. Σε κάθε περίπτωση το σύστημα θα βρει τις καλύτερες διαδρομές. Όπως βλέπουμε και στην εικόνα 37 όλες οι διαδρομές περιλαμβάνουν μετακίνηση με τα πόδια. Ο χρήστης μπορεί να κλείσει το δρομολόγιο της διαδρομής. Το σύστημα προβάλλει την αρχική και τελική στάση, τη γραμμή και τις ελεύθερες θέσεις.



Εικόνα 37. Πληροφορίες διαδρομής

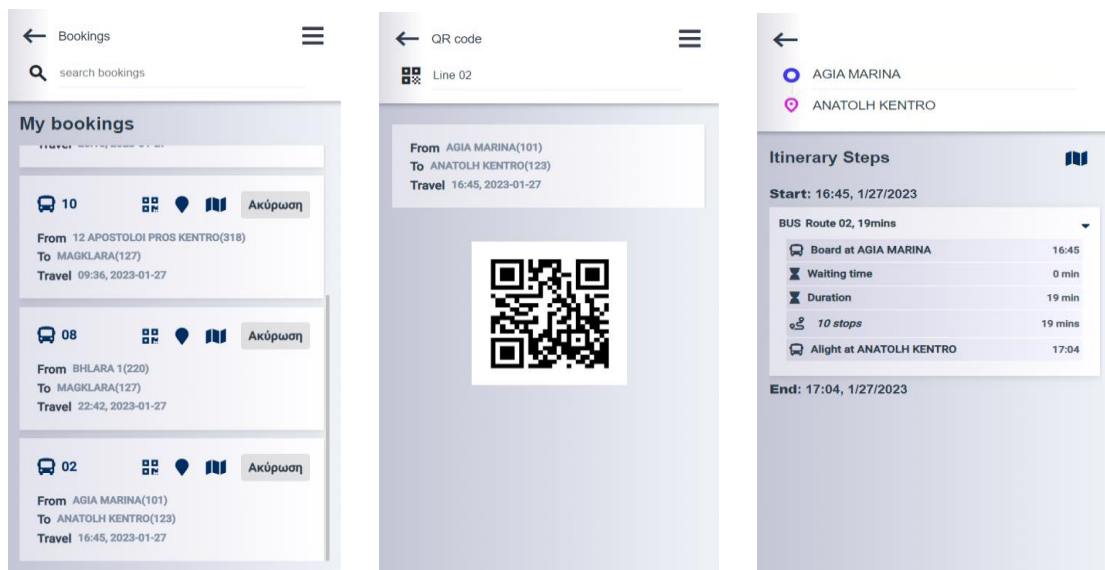
Για κάθε διαδρομή ο χρήστης μπορεί να δει περισσότερες πληροφορίες πατώντας στο εικονίδιο του χάρτη. Όπως βλέπουμε στην εικόνα 37 έχουμε λεπτομερείς πληροφορίες. Αν μια διαδρομή περιλαμβάνει και μετακίνηση με τα πόδια, το σύστημα παρέχει οδηγίες. Για το δρομολόγιο έχουμε τους χρόνους από στάση σε στάση, τη διάρκεια ταξιδιού και έναν χρόνο αναμονής. Οι χρόνοι είναι προσεγγιστικοί και υπολογίστηκαν μέσω του Open Trip Planner. Ο χρήστης έχει την δυνατότητα να δει τη διαδρομή και οπτικά. Πατώντας πάνω σε κάποια οδηγία ο χάρτης θα εστιάσει στον αντίστοιχο δρόμο ή στη στάση της διαδρομής. Στη συνέχεια κάνοντας κράτηση, το σύστημα εμφανίζει τα αντίστοιχα μηνύματα με βάση τις προϋποθέσεις κράτησης του use case και ανανεώνει τις διαθέσιμες θέσεις σε περίπτωση επιτυχίας. Στην εικόνα 38 παρουσιάζονται και οι οθόνες των μηνυμάτων.





Εικόνα 38. Μηνύματα κράτησης

## Εμφάνιση πληροφορίας κράτησης (UC6)



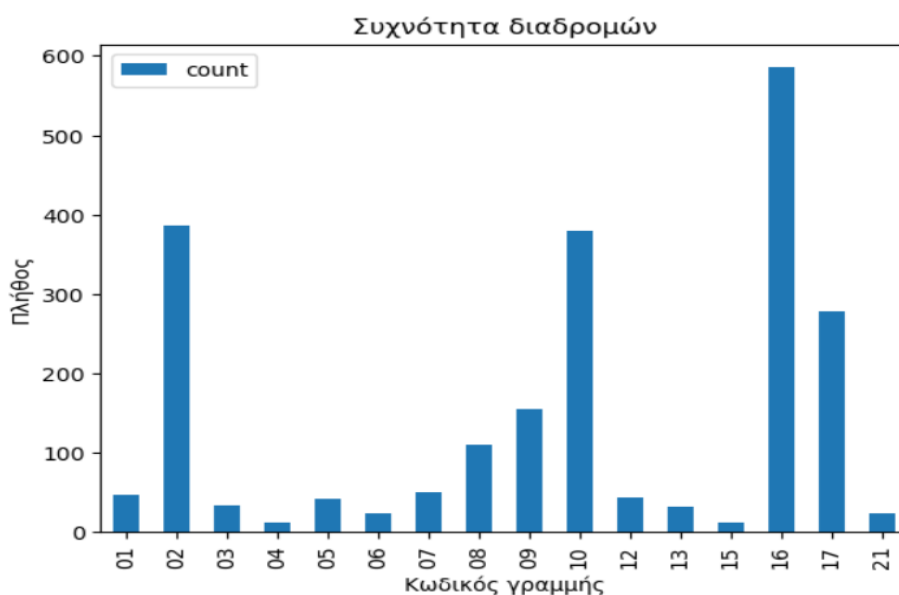
Εικόνα 39. Πληροφορία κράτησης

Για την περίπτωση αυτή ο χρήστης από την αρχική οθόνη επιλέγει το κουμπί 'Bookings'. Το σύστημα εμφανίζει τη λίστα με τις κρατήσεις. Όπως φαίνεται στην εικόνα 39, για κάθε κράτηση το σύστημα δίνει την επιλογή εμφάνισης του κωδικού QR, της πληροφορίας της διαδρομής και την δυνατότητα ακύρωσης πατώντας στα αντίστοιχα εικονίδια. Επίσης για ευκολία του χρήστη, δίνεται η δυνατότητα να δει άμεσα τις πληροφορίες της στάσης έναρξης.

## Κεφάλαιο 7. Πειραματική αξιολόγηση

Στην ενότητα αυτή παρουσιάζεται μια πειραματική αξιολόγηση του συστήματος. Ο στόχος ήταν η χρήση του ιστορικού της βάσης για την εξαγωγή στατιστικών στοιχείων που θα αφορούν την ροή των επιβατών. Η διαδικασία λειτούργησε ως ένα τεστ του συστήματος που το αξιολόγησε με βάση την ταχύτητα εξυπηρέτησης των χρηστών, κάτω από συνθήκες μεγάλου όγκου αιτημάτων.

Το πρώτο βήμα που έγινε ήταν ο εντοπισμός κάποιων δημοφιλών γραμμών. Με βάση το στατικό πρόγραμμα του αστικού ΚΤΕΛ, παρατηρήθηκε ότι οι γραμμές 16, 02, 10, 17 έχουν πολλά δρομολόγια όπως φαίνεται και στην εικόνα 40. Όπως βλέπουμε στην εικόνα 40 έχουμε το πλήθος δρομολογίων μέσα στην εβδομάδα στον άξονα y και το κωδικό όνομα των γραμμών στον άξονα x. Με αφορμή αυτό το γεγονός η αξιολόγηση εστίασε σε αυτά.



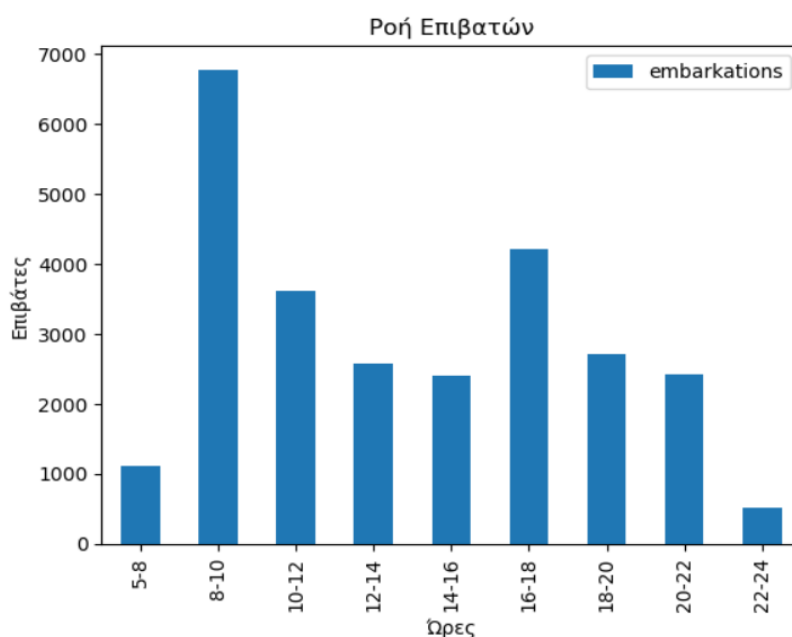
Εικόνα 40. Δρομολόγια γραμμών

Το επόμενο βήμα της προσομοίωσης ήταν η δημιουργία δεδομένων τα οποία προσομοιώνουν ένα σενάριο πραγματικού κόσμου. Για το λόγο αυτό δημιουργήθηκαν 30000 χρήστες οι οποίοι έκαναν κρατήσεις στο πέρας μιας εβδομάδας. Οι κρατήσεις έγιναν τυχαία και με τέτοιο τρόπο ώστε το σύστημα να δώσει μεγαλύτερη πιθανότητα να γίνει μια κράτηση σε δρομολόγιο το οποίο γίνεται κατά τις ώρες αιχμής. Συνήθως οι ώρες αιχμής είναι το πρωί όπου οι άνθρωποι μετακινούνται προς στους χώρους εργασίας ή σε σχολεία και πανεπιστήμια καθώς και το απόγευμα όπου επιστρέφουν.



Καθώς η εφαρμογή λάμβανε τις κρατήσεις ενημέρωνε κατάλληλα και τον πίνακα trip. Επιπρόσθετα η τιμή totalSeats που ορίζει την μέγιστη πληρότητα των λεωφορείων τέθηκε ίση με 50. Αρχικά παρατηρήθηκε ότι το σύστημα είναι ικανό να εξυπηρετήσει ένα μεγάλο όγκο δεδομένων μέσα σε μερικά δευτερόλεπτα. Στη συνέχεια βασιζόμενοι στα δεδομένα της βάσης έγιναν 2 πειράματα.

Στο πρώτο πείραμα παρατηρήθηκε η ροή των επιβατών κατά τις ώρες της ημέρας. Αυτό το πείραμα έγινε με βάση την τιμή embarkation στον πίνακα trip. Στην εικόνα 41 βλέπουμε το διάγραμμα της ροής.

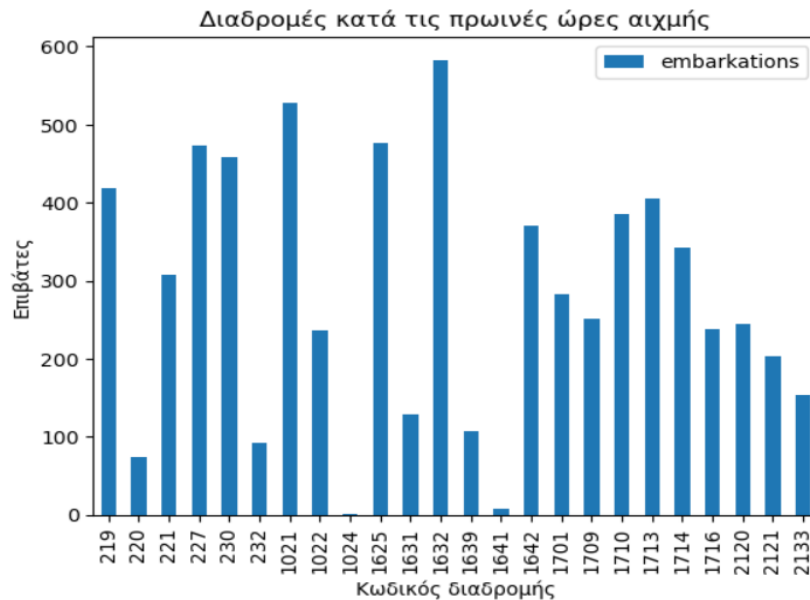


Εικόνα 41. Ροή επιβατών μέσα στην εβδομάδα

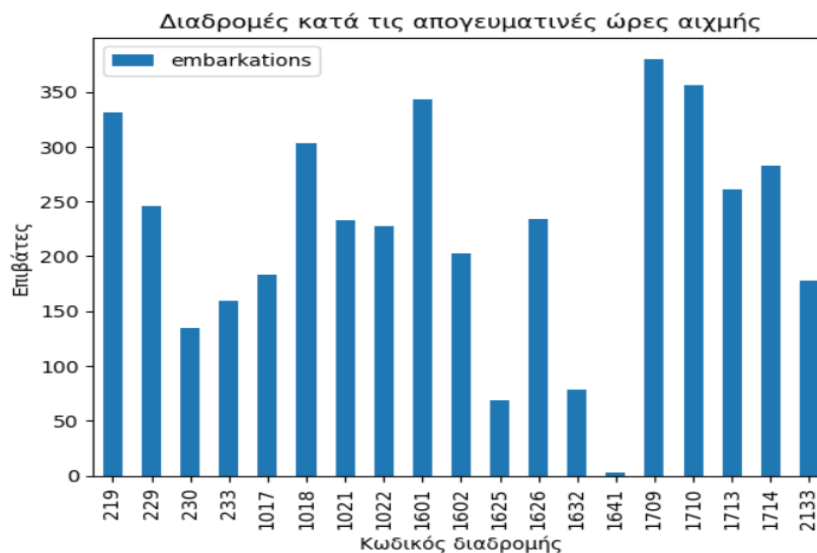
Αρχικά στην εικόνα 41 έχουμε τις ώρες της ημέρας χωρισμένες σε περιόδους στον άξονα x και το συνολικό πλήθος επιβατών σε λεωφορεία στον άξονα y. Όπως παρατηρήθηκε έχουμε ώρα αιχμής στις 8 με 10 το πρωί, έπειτα έχουμε επιβάτες λίγο πιο διάσπαρτα μέχρι και στις 4 η ώρα. Στην περίοδο 4 με 6 πάλι έχουμε μια μικρή άνοδο των επιβατών η οποία σταδιακά μειώνεται. Με βάση το διάγραμμα συμπεράναμε ότι οι ώρες αιχμής πλησιάζουν την κατανομή  $x^2$ . Μέσω του διαγράμματος μπορούμε να πούμε ότι γενικά υπάρχει μεγάλη κινητικότητα τις πρωινές ώρες. Τα αποτελέσματα αυτά συμφώνησαν και με την αρχική υπόθεση με βάση την οποία δημιουργήθηκαν.

Στην συνέχεια εντοπίστηκαν διαδρομές οι οποίες είναι οι πιο δημοφιλείς κατά τις ώρες αιχμής. Για αυτό το πείραμα εστίασαμε στα δρομολόγια που γίνονται στις περιόδους 8 με 10 το πρωί και 4 με 6 το απόγευμα. Στις εικόνες 42, 43 φαίνονται τα διαγράμματα με τις αντίστοιχες διαδρομές. Στον άξονα y έχουμε τους συνολικούς επιβάτες και στον άξονα x τους κωδικούς των διαδρομών. Με βάση το διάγραμμα της εικόνας 42 παρατηρήθηκε ότι οι διαδρομές με κωδικούς 1632, 1021, 1625, 227, 230

έχουν τους περισσότερους επιβάτες ενώ οι διαδρομές 1641 και 1024 τους λιγότερους. Παρόμοια παρατηρώντας το διάγραμμα της εικόνας 43 είδαμε ότι οι διαδρομές 1709, 1710, 1601, 219 έχουν το μεγαλύτερο σύνολο επιβατών. Παρατηρώντας τα διαγράμματα δεν φάνηκε ότι ακολουθούν κάποια γνώστη κατανομή.



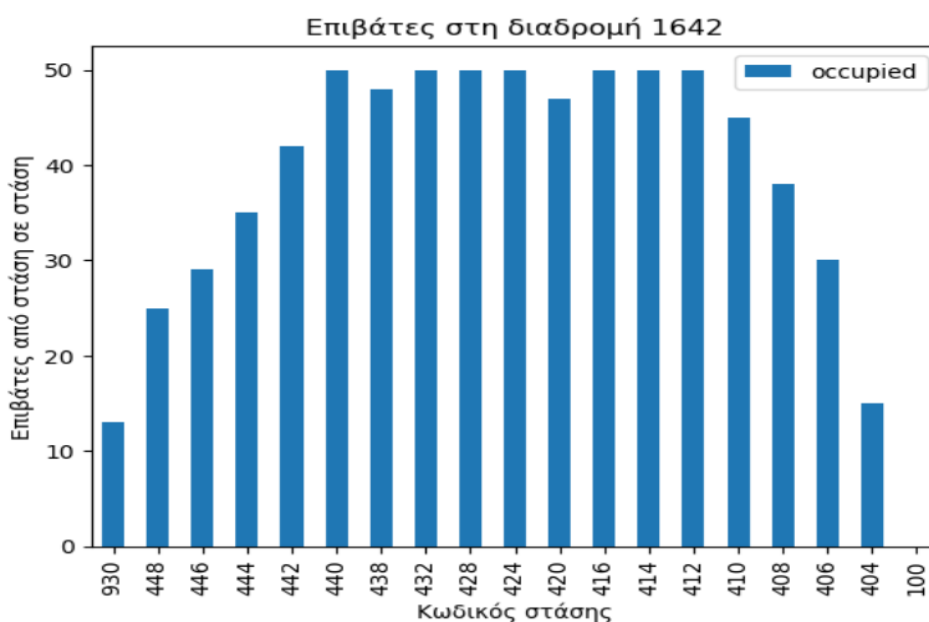
Εικόνα 42. Διαδρομές και αναβάσεις κατά τις πρωινές ώρες αιχμής



Εικόνα 43. Διαδρομές και αναβάσεις κατά τις απογευματινές ώρες αιχμής

Τέλος στην εικόνα 44 δίνεται ένα στιγμιότυπο ροής επιβατών στο δρομολόγιο 223939(trip) με διαδρομή 1642(route). Στον άξονα x έχουμε τις στάσεις της διαδρομής και στον άξονα y τους επιβάτες από στάση σε στάση. Παρατηρήθηκε ότι η χωρητικότητα φθάνει και δεν ξεπερνάει το μέγιστο, που σημαίνει ότι η αναβάθμιση της βάσης έγινε χωρίς σφάλματα. Με βάση την εικόνα 44 φάνηκε επίσης ότι το διάγραμμα

των επιβατών στο λεωφορείο προσεγγίζει μια συμμετρική γενική κανονική κατανομή. Το λεωφορείο αρχίζει να γεμίζει σταδιακά από τη στάση 930 μέχρι τη στάση 440. Έπειτα από τη στάση 440 μέχρι και την στάση 410 διατηρεί το πλήθος επιβατών. Από τη στάση 412 μέχρι και την 100 οι επιβάτες μειώνονται. Όπως φάνηκε έχουμε μια συμμετρία μεταξύ των αναβάσεων και των αποβιβάσεων. Επίσης βγήκε και το συμπέρασμα ότι οι στάσεις από την 930 μέχρι και την 440 αποτελούν τις στάσεις επιβίβασης για τους περισσότερους επιβάτες ενώ οι στάσεις 412 μέχρι και την 100 τις στάσεις αποβίβασης.



Εικόνα 44. Πλήθος επιβατών στη διαδρομή 1642 κατά το δρομολόγιο 223939

Τα πειράματα αυτά έγιναν κυρίως για τον έλεγχο του συστήματος γιατί δεν υπάρχουν πραγματικά δεδομένα που περιγράφουν την ροή των επιβατών στην πόλη των Ιωαννίνων. Με βάση τα πειράματα εντοπίστηκαν και κάποια σφάλματα που υπήρχαν στο σύστημα τα οποία δεν θα ήταν δυνατόν να εντοπιστούν μέσω unit tests. Για παράδειγμα το σύστημα ξεπερνούσε το μέγιστο αριθμό πληρότητας κατά την αναβάθμιση σε κάποιες περιπτώσεις. Επίσης επειδή εφαρμόζεται πολυνηματισμός σε έναν βαθμό κατά την αναβάθμιση της βάσης, παρατηρήθηκε ότι υπό μεγάλο όγκο αιτημάτων μέσα σε πολύ μικρό χρόνο, υπήρχαν deadlocks τα οποία και διορθώθηκαν.

Παρόλου που αυτά τα πειράματα δεν ήταν σε θέση να δώσουν κάποια συμπεράσματα που αφορούν τον πραγματικό κόσμο λειτούργησαν ως μια καλή αξιολόγηση του συστήματος. Στο μέλλον, με την τοποθέτηση των κατάλληλων συστημάτων στα αστικά λεωφορεία, το συγκεκριμένο σύστημα θα είναι σε θέση να ικανοποιήσει τις ανάγκες αποθήκευσης και καταγραφής των δεδομένων.

## Κεφάλαιο 8. Επίλογος

### 8.1. Σύνοψη και συμπεράσματα

Ο στόχος της παρούσας διπλωματικής ήταν η ανάπτυξη μιας εφαρμογής η οποία θα βοηθήσει τους επιβάτες των αστικών λεωφορείων να προγραμματίσουν τις μετακινήσεις τους πιο αποτελεσματικά. Πιο συγκεκριμένα, μέσω του εργαλείου αυτού ο χρήστης έχει την δυνατότητα να ενημερώνεται για την κατάσταση των λεωφορείων, τις αφίξεις και τις αναχωρήσεις στις στάσεις σε πραγματικό χρόνο. Μια άλλη δυνατότητα της εφαρμογής είναι η εύρεση βέλτιστων δρομολογίων, με βάση την διάρκεια και ώρα μετακίνησης. Αυτή η δυνατότητα συνδυάστηκε με την λειτουργία κράτησης θέσης που παρέχει η εφαρμογή. Για κάποιο προτεινόμενο δρομολόγιο ο χρήστης μπορεί να κρατήσει μια θέση αν το λεωφορείο δεν είναι πλήρως κατειλημμένο. Για κάθε κράτηση το σύστημα δημιουργεί ένα μοναδικό κωδικό QR ο οποίος θα την περιγράφει. Σημαντική είναι η αναφορά στις υπηρεσίες χαρτών τις οποίες προσφέρει η εφαρμογή. Ο χρήστης έχει την δυνατότητα να δει όλες τις πληροφορίες των διαδρομών και οπτικά με πολλές λεπτομέρειες.

Η μεγάλη δυσκολία ανάπτυξης αυτής της εφαρμογής προήλθε από την ανεπάρκεια στα δεδομένα. Πιο συγκεκριμένα, η πληροφορία πραγματικού χρόνου που παρέχεται από την υπηρεσία του αστικού ΚΤΕΛ δεν μας δίνει τις εκτιμήσεις αφίξεων στις στάσεις, τον κωδικό της επόμενης στάσης ενός λεωφορείου και τον κωδικό δρομολογίου(trip\_id). Αυτό είχε ως αποτέλεσμα να είναι αδύνατη η αντιστοιχία των πραγματικών δεδομένων με τον πίνακα schedule της βάσης. Αυτό περιόρισε αρκετά τη λειτουργία της κράτησης θέσης. Στην παρούσα φάση ο χρήστης δεν μπορεί να κάνει κράτηση θέσης όταν το λεωφορείο ενός δρομολογίου είναι ενεργό γιατί δεν μπορούμε να ξέρουμε την πληρότητα του. Έπειτα για το πρόβλημα της εκτίμησης αφίξεων, έγινε η σκέψη αποθήκευσης των δεδομένων πραγματικού χρόνου για ένα μεγάλο χρονικό διάστημα. Στόχος ήταν η μελέτη τους και εξαγωγή κάποιων μέσων όρων καθυστέρησης και χρόνου άφιξης με βάση τις συντεταγμένες. Αφού έγινε η συλλογή τους για περίπου 20 μέρες, ακολούθησε μια ανάλυση. Παρατηρήθηκε ότι υπήρχαν μεγάλες αποκλίσεις και ασάφειες με αποτέλεσμα να μην έχουμε καλές ή και καθόλου προβλέψεις για κάποια δρομολόγια. Αυτό το πρόβλημα έχει μεγάλη έκταση γιατί η ώρα άφιξης εξαρτάται κυρίως από τους επιβάτες και τις στάσεις στις οποίες θα ανέβουν ή κατέβουν, την κίνηση στους δρόμους, τις απρόβλεπτες καταστάσεις μέσα σε ένα λεωφορείο και τις καιρικές συνθήκες. Τέτοιου είδους δεδομένα δεν υπήρχαν διαθέσιμα. Όλα αυτά οδήγησαν στην χρήση του εργαλείου Open Trip Planner το οποίο παρέχει δυνατότητα

routing για βαριά οχήματα. Το εργαλείο αυτό συγκρίθηκε και με άλλες υπηρεσίες όπως Google maps και την πολύ δημοφιλή υπηρεσία HERE routing και παρατηρήθηκε ότι έχει αρκετά καλές προβλέψεις, αν και μπορεί να υπάρχουν μικρές αποκλίσεις σε ώρες μεγάλης κίνησης και συνωστισμού. Η προετοιμασία του αποτέλεσε άλλη μια πρόκληση λόγω της ασυμβατότητας των δεδομένων. Τέλος, μεγάλη δυσκολία αποτέλεσε η ανάπτυξη και σχεδίαση της εφαρμογής λόγω των πολύπλοκων τεχνολογιών λογισμικού που επιλέχθηκαν. Δόθηκε μεγάλη προσοχή στη ανάπτυξη με τρόπο ο οποίος να επιτρέπει την επεκτασιμότητα και τον αποτελεσματικό έλεγχο όλων των κομματιών της εφαρμογής.

Κλείνοντας, η ανάπτυξη αυτής της εφαρμογής απαίτησε ένα πολύ μεγάλο χρόνο για έρευνα, σχεδίαση και προγραμματισμό κατά τον οποίο έγινε εμβάθυνση στις υπάρχουσες γνώσεις, και αποδείχθηκε πολύ ενδιαφέρουσα και εκπαιδευτική. Στο μέλλον προβλέπεται ότι η εφαρμογή θα δημοσιευθεί και για χρήση από το κοινό. Τουλάχιστον οι λειτουργίες που δεν αφορούν την κράτηση θέσης, γιατί αυτές απαιτούν την συνεργασία με τον οργανισμό του αστικού ΚΤΕΛ Ιωαννίνων.

## 8.2. Μελλοντικές επεκτάσεις

Πολλές είναι οι επεκτάσεις που θα μπορούσαν να προστεθούν στην εφαρμογή. Ωστόσο οι περισσότερες απαιτούν την συνεργασία του αστικού ΚΤΕΛ. Κάποιες προτάσεις θα αναφερθούν παρακάτω.

- Όπως αναφέρθηκε το λογισμικό Open Trip Planner μπορεί να δεχθεί δεδομένα πραγματικού χρόνου που αφορούν τις αναβαθμίσεις δρομολογίων, τις τοποθεσίες λεωφορείων και ειδοποιήσεις(alerts). Στην παρούσα φάση ο OTP είναι σε πειραματικό στάδιο. Δέχεται αναβαθμίσεις των δρομολογίων μέσω αρχείων protobuf με χρήση δεδομένων ελέγχου καθώς δεν υπάρχουν επαρκείς δεδομένα πραγματικού χρόνου. Συγκεκριμένα απαιτείται να υπάρξει συσχέτιση του πίνακα schedule και του πίνακα live\_data της βάσης. Θα μπορούσαμε να κάνουμε τροποποίηση στα συστήματα του αστικού ΚΤΕΛ ώστε να έχουμε διαθέσιμες αυτές τις πληροφορίες. Ο OTP θα μπορεί να προσαρμόζει τα δρομολόγια που προτείνει και τους χρόνους άφιξης καθώς και να ενημερώνει τον χρήστη για διαδρομές που ακυρώθηκαν, πρόσθετες γραμμές, αλλαγές στο πρόγραμμα και στάσεις οι οποίες θα εξαιρεθούν από κάποιο δρομολόγιο για διάφορους λόγους, όπως ατυχήματα ή έργα.
- Δυνατότητα κράτησης θέσης και όταν το λεωφορείο ενός δρομολογίου είναι ενεργό.
- Δυνατότητα αγοράς εισιτηρίου μέσω της εφαρμογής.

- Δημιουργία ελεγκτή στον διακομιστή για επικύρωση κράτησης όταν ο χρήστης ανεβαίνει στο αντίστοιχο δρομολόγιο ή κατά την ακύρωση, με βάση τον κωδικό QR.
- Υλοποίηση μεθόδου η οποία εξάγει στατιστικά, με βάση τον πίνακα trip της βάσης και καθαρισμό του μετά από την ολοκλήρωση ενός δρομολογίου.
- Επέκταση της διεπαφής χρήστη με ένα module το οποίο θα εμφανίζει αναλυτικά τις στάσεις ενός δρομολογίου και για κάθε στάση πόσοι επιβάτες θα ανέβουν, πόσοι θα κατέβουν καθώς και τον αριθμό επιβατών στο λεωφορείο. Αυτή η πληροφορία θα μπορεί να βοηθήσει τον οδηγό ώστε να πραγματοποιήσει το δρομολόγιο πιο αποτελεσματικά. Για αυτήν την λειτουργία απαιτείται η συνεργασία με τον οργανισμό του ΚΤΕΛ. Θα πρέπει να ξέρουμε ποιο δρομολόγιο εκτελεί κάθε λεωφορείο (δηλαδή ποιο trip\_id) και να υπάρχει η δυνατότητα κράτησης θέσης.

# Βιβλιογραφία

- [1] Διαδικτυακή εφαρμογή: [https://en.wikipedia.org/wiki/Web\\_application](https://en.wikipedia.org/wiki/Web_application)
- [2] Διαδικτυακός διακομιστής: [https://developer.mozilla.org/en-US/docs/Learn/Common\\_questions/What\\_is\\_a\\_web\\_server](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server)
- [3] PostgreSQL: <https://www.postgresql.org/>
- [4] TypeORM: <https://typeorm.io/>
- [5] NestJS: <https://nestjs.com/>
- [6] Angular: <https://angular.io/>
- [7] SocketIO: <https://socket.io/>
- [8] PassportJS: <https://www.passportjs.org/>
- [9] LeafletJS: <https://leafletjs.com/>
- [10] Αρχιτεκτονική τριών επιπέδων: <https://www.ibm.com/topics/three-tier-architecture>
- [11] Redux: <https://redux.js.org/>
- [12] NGRX: <https://ngrx.io/>
- [13] Observable: <https://www.freecodecamp.org/news/an-introduction-to-observables-in-reactive-programming-1cf3e23bb94/>
- [14] Περιγητής ιστού: [https://en.wikipedia.org/wiki/Web\\_browser](https://en.wikipedia.org/wiki/Web_browser)
- [15] Protocol buffers: <https://developers.google.com/protocol-buffers>
- [16] TypeScript: <https://www.typescriptlang.org/>
- [17] VS Code: <https://code.visualstudio.com/>
- [18] General Transit Feed Specification: <https://gtfs.org/>
- [19] Open street map: <https://www.openstreetmap.org/#map=9/39.3619/19.8935>
- [20] ArcMap: <https://en.wikipedia.org/wiki/ArcMap>
- [21] Open Trip Planner: <https://www.opentripplanner.org/>