

**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ  
ΠΟΛΥΤΕΧΝΕΙΟ**  
Σχολή Αγρονόμων & Τοπογράφων  
Μηχανικών  
**ΓΕΩΠΛΗΡΟΦΟΡΙΚΗ  
ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ**



**NATIONAL TECHNICAL  
UNIVERSITY OF ATHENS**  
School of Rural & Surveying  
Engineering  
**GEOINFORMATICS  
POST-GRADUATE PROGRAMME**

Μεταπτυχιακή Εργασία  
Ανάπτυξη εφαρμογής smartphone για την ενημέρωση του χρήστη  
υπεραστικού δικτύου συγκοινωνιών

Ξυλάκης Ι. Εμμανουήλ

Επιβλέπων Καθηγητής  
Κάβουρας Μαρίνος  
Καθηγητής Ε.Μ.Π.

ΑΘΗΝΑ | Μάρτιος 2015



**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ  
ΠΟΛΥΤΕΧΝΕΙΟ**  
Σχολή Αγρονόμων & Τοπογράφων  
Μηχανικών  
**ΓΕΩΠΛΗΡΟΦΟΡΙΚΗ  
ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ**



**NATIONAL TECHNICAL  
UNIVERSITY OF ATHENS**  
School of Rural & Surveying  
Engineering  
**GEOINFORMATICS  
POST-GRADUATE PROGRAMME**

**Master Thesis**  
Smartphone application development for information  
of intercity bus service

**Xylakis I. Emmanouil**

Supervisor  
Kavouras Marinos  
Professor N.T.U.A.

ATHENS | March 2015



**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ  
ΠΟΛΥΤΕΧΝΕΙΟ**  
Σχολή Αγρονόμων & Τοπογράφων  
Μηχανικών  
**ΓΕΩΠΛΗΡΟΦΟΡΙΚΗ  
ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ**



**NATIONAL TECHNICAL  
UNIVERSITY OF ATHENS**  
School of Rural & Surveying  
Engineering  
**GEOINFORMATICS  
POST-GRADUATE PROGRAMME**

Μεταπτυχιακή Εργασία  
Ανάπτυξη εφαρμογής smartphone για την ενημέρωση του χρήστη  
υπεραστικού δικτύου συγκοινωνιών

Ξυλάκης Ι. Εμμανουήλ

Επιβλέπων Καθηγητής  
Κάβουρας Μαρίνος  
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 6<sup>η</sup> Μαρτίου 2015.

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....  
Κάβουρας Μαρίνος  
Καθηγητής Ε.Μ.Π.

.....  
Αργιαλάς Δημήτριος  
Καθηγητής Ε.Μ.Π.

.....  
Κόκλα Μαργαρίτα  
Λέκτορας Ε.Μ.Π.

*(Υπογραφή)*

.....

Ξυλάκης Εμμανουήλ

Διπλωματούχος Αγρονόμος και Τοπογράφος Μηχανικός Ε.Μ.Π.

© 2015 – All rights reserved

## Πρόλογος

Η παρούσα μεταπτυχιακή εργασία ολοκληρώνει το Δ.Π.Μ.Σ. Γεωπληροφορικής της Σχολής Αγρονόμων και Τοπογράφων Μηχανικών του ΕΜΠ. Απώτερος στόχος αυτής είναι η ανάπτυξη μιας εφαρμογής smartphone για την ενημέρωση του χρήστη υπεραστικού δικτύου συγκοινωνιών με την αξιοποίηση των δυνατοτήτων των χωρικών βάσεων δεδομένων.

Θα ήθελα να ευχαριστήσω τη διοίκηση και το αρμόδιο προσωπικό του ΚΤΕΛ Χανίων - Ρεθύμνου για τη διάθεση δεδομένων, ώστε να είναι δυνατή η δοκιμή του συστήματος.

Ακόμα, θα ήθελα να ευχαριστήσω τον κ. Μαρίνο Κάβουρα, επιβλέπων καθηγητή της εργασίας, καθώς και την κ. Μαργαρίτα Κόκλα μέλος της τριμελούς εξεταστικής επιτροπής. Ευχαριστίες οφείλω στον κ. Δημήτριο Αργιαλά, μέλος της τριμελούς εξεταστικής επιτροπής, για τη συνεχή στήριξη του τα χρόνια των σπουδών μου στην Σ.Α.Τ.Μ., καθώς και στον κ. Άγγελο Τζώτσο, για τις πολύτιμες συμβουλές του σχετικά με την υλοποίηση του πειραματικού μέρους της εργασίας.

Τέλος, θερμές ευχαριστίες οφείλω στην οικογένειά μου αλλά και σε όσους ήταν δίπλα μου καθ' όλη τη διάρκεια συγγραφής.





## Περίληψη

Στην σημερινή εποχή σχεδόν όλοι διαθέτουν μια συσκευή smartphone. Συνεπώς, η ανάγκη για εφαρμογές που λειτουργούν σε αυτά αυξάνεται συνεχώς. Οι χρήστες χρειάζονται εφαρμογές, που θα εξυπηρετούν τις καθημερινές τους ανάγκες. Μία από αυτές είναι και η μετακίνηση με τα Μέσα Μαζικής Μεταφοράς (MMM).

Η παρούσα μεταπτυχιακή εργασία στοχεύει στην ανάπτυξη μίας εφαρμογής smartphone για την πληροφόρηση των χρηστών του δικτύου των Υπεραστικών Συγκοινωνιών με αξιοποίηση των δυνατοτήτων των χωρικών βάσεων δεδομένων. Τα δεδομένα προέρχονται από το ΚΤΕΛ Χανίων Ρεθύμνου και η εφαρμογή έχει προσαρμοστεί στις απαιτήσεις του συγκεκριμένου συστήματος. Το σύστημα μπορεί να επεκταθεί και να εφαρμοστεί και σε άλλα δίκτυα παρέχοντας δεδομένα σε κατάλληλη μορφή.

Μέσα από την εφαρμογή που αναπτύχθηκε, ο χρήστης έχει τη δυνατότητα να ενημερωθεί για ένα πλήθος στατικών αλλά και δυναμικών στοιχείων για τη βελτιστοποίηση της οργάνωσης και πραγματοποίησης του ταξιδιού του. Συγκεκριμένα, οι στατικές πληροφορίες που μπορεί να ενημερωθεί αφορούν τα δρομολόγια και τις στάσεις των λεωφορείων, με τους αντίστοιχους χάρτες. Τα παραπάνω στοιχεία διατίθενται και εκτός σύνδεσης, ενώ ανανεώνονται αυτόματα όταν υπάρχει κάποια τροποποίηση σε αυτά. Όσον αφορά τα δυναμικά δεδομένα, τα οποία και είναι διαθέσιμα μόνο όταν υπάρχει σύνδεση στο διαδίκτυο, ο χρήστης μπορεί να ενημερωθεί σχετικά με τις θέσεις και λοιπά στοιχεία των λεωφορείων της εκάστοτε γραμμής. Επίσης, προσφέρονται πληροφορίες για τις κοντινές σε αυτόν στάσεις, και εύρεση του βέλτιστου δυνατού συνδυασμού γραμμών για τη μετάβαση μεταξύ δύο στάσεων του δικτύου.

Καθώς η εφαρμογή απευθύνεται σε ένα ευρύ κοινό και η διάθεσή της απαιτείται να είναι δια-πλατφορμική, η ανάπτυξή της πραγματοποιήθηκε με τεχνολογίες διαδικτύου ως μία «Υβριδική Εφαρμογή» με το πλαίσιο Cordova και με ελάχιστες τροποποιήσεις είναι δυνατό να εγκατασταθεί στις περισσότερες πλατφόρμες smartphone. Η καρδιά του συστήματος είναι ένας απομακρυσμένος εξυπηρετητής, ο οποίος διατηρεί τα δεδομένα σε μια χωρική βάση δεδομένων και είναι υπεύθυνος για όλες τις επεξεργασίες. Τα δεδομένα που μεταβάλλονται στο χρόνο είναι ενδεικτικά, ενώ οι στάσεις και οι γραμμές αντιστοιχούν σε αυτές όπου ήταν διαθέσιμα όλα τα απαραίτητα δεδομένα.

**Λέξεις Κλειδιά** Υβριδική εφαρμογή, Χωρικές Βάσεις Δεδομένων, Cordova, ΚΤΕΛ, Υπεραστικές συγκοινωνίες, Leaflet, jQuery Mobile, Backbonejs, PostgreSQL, PostGIS



## Abstract

Today, almost everyone owns a smartphone device. Therefore the demand for smartphone applications is constantly increasing. Users require applications which can fulfill their daily needs such as the use of public transportation.

The present master thesis aims to the development of a smartphone application which will provide users with information concerning the local intercity bus service, exploiting the potentials of spatial databases. All data is derived from the system of Chania - Rethymno intercity bus service and the application adopts to the requirements of the specific system. The system can be expanded for other regions by providing data at appropriate format.

Throughout the application, users have access to a variety of static and dynamic elements for the organization of their trip. Specifically static elements concern the available routes and bus stops combined with additional maps. They are automatically updated whenever a change occurs and they are available without internet connection. Concerning the dynamic elements (which are available only with internet connection) users have access to information about the bus routes and their exact position. Furthermore, by using the smartphone GPS system, users can detect the bus stops closest to them and the best available route combination in order to reach their destination.

Since the application applies to the public it must be available on most application platforms. For this reason, it has been developed with web technologies as a "hybrid" application with Cordova Framework, which can be installed in most smartphone platforms with minimum modifications. The heart of the application is a remote server which contains all data in a spatial database and is responsible for the management of users requests. All time varying data are indicative whereas the bus stops and routes consist of those for which all necessary data were available.

**Key words** Hybrid Application, Spatial Databases, Cordova, Intercity bus services, Leaflet, jQuery mobile, Backbonejs, PostgreSQL, PostGIS



# Περιεχόμενα

Πρόλογος.....	0
Περίληψη .....	iii
Abstract.....	v
Περιεχόμενα .....	vii
Κατάλογος Εικόνων .....	x
Κατάλογος Πινάκων.....	xi
Κατάλογος Διαγραμμάτων.....	xi
Κατάλογος Κώδικα.....	xi
1. Εισαγωγή.....	1
1.1 Στόχος .....	1
1.2 Επιλογές.....	2
1.3 Διάρθρωση εργασίας.....	4
2.    Ανασκόπηση Βιβλιογραφίας .....	5
2.1 Κατηγορίες Εφαρμογών .....	5
2.1.1 Native εφαρμογές.....	5
2.1.2 Web εφαρμογές.....	6
2.1.3 Υβριδικές εφαρμογές.....	6
2.2 Υβριδικές εφαρμογές.....	8
2.2.1 Πλεονεκτήματα – Μειονεκτήματα Υβριδικής Προσέγγισης.....	8
2.2.2 Apache Cordova .....	10
2.1.3 Αρχιτεκτονική μίας υβριδικής εφαρμογής .....	14
2.3 Διαδικτυακές τεχνολογίες από τη πλευρά του πελάτη (Client Side).....	17
2.3.1 HTML5 .....	17
2.3.2 JavaScript.....	18
2.3.3 CSS .....	18
2.3.4 3 <sup>rd</sup> Party Services.....	19
2.4 Διαδικτυακές τεχνολογίες από τη πλευρά του εξυπηρετητή (Server Side) .....	28
2.4.1 Λειτουργικό σύστημα του εξυπηρετητή.....	28

2.4.2	Γλώσσα ανάπτυξης εξυπηρετητή (PHP).....	28
2.4.3	Το σύστημα διαχείρισης βάσεων δεδομένων (PostgreSQL).....	29
2.4.4	Η γλώσσα SQL .....	32
2.5	Τεχνολογίες επικοινωνίας.....	33
2.5.1	AJAX .....	33
2.5.2	API.....	33
2.5.3	Πρότυπο JSON.....	34
2.6	Σχετικά .....	35
2.6.1	Το πρότυπο GTFS .....	35
2.6.2	Σχετικές εφαρμογές .....	36
3.	Μεθοδολογία.....	41
3.1	Περιγραφή εφαρμογής .....	41
3.1.1	Λειτουργίες.....	41
3.1.2	Αρχιτεκτονική συστήματος.....	42
3.2	Διεπαφή χρήστη .....	44
3.2.1	Σχεδιασμός διεπαφής χρήστη.....	44
3.2.2	Υλοποίηση διεπαφής χρήστη.....	50
3.2.3	Κωδικοποίηση Αρχείων.....	56
3.3	Βάση Δεδομένων .....	57
3.3.1	Σχεδιασμός βάσης Δεδομένων .....	57
3.3.2	Βάση δεδομένων στην πλευρά του εξυπηρετητή .....	58
3.3.4	Βάση δεδομένων στην πλευρά του πελάτη .....	64
3.4	Διαδικασία Λειτουργίας.....	66
3.4.1	Επικοινωνία με την βάση δεδομένων στην πλευρά του πελάτη .....	66
3.4.2	Επικοινωνία με την βάση δεδομένων στην πλευρά του εξυπηρετητή.....	68
3.5	Υλοποιήσεις.....	73
3.5.1	Αρχική οθόνη.....	73
3.5.2	Γραμμές δικτύου .....	74
3.5.3	Στάσεις .....	76
3.5.4	Οργάνωση ταξιδιού .....	77

3.5.5 Κοντινές στάσεις.....	78
3.5.6 Παρακολούθηση Γραμμής Ενδιαφέροντος .....	79
3.5.7 Παρακολούθηση Λεωφορείου Ενδιαφέροντος .....	80
3.5.8 Χρήση εκτός σύνδεσης.....	81
4. Συμπεράσματα και Προοπτικές.....	83
4.1 Συμπεράσματα.....	83
4.2 Προοπτικές.....	84
Βιβλιογραφία.....	87

## Κατάλογος Εικόνων

Εικόνα 2.1: Μέθοδοι ανάπτυξης εφαρμογών smartphones [5].....	7
Εικόνα 2.2: Δημιουργία Υβριδικών Εφαρμογών με χρήση του PhoneGap [6] .....	11
Εικόνα 2.3: Στοιχεία μιας Υβριδικής Εφαρμογής .....	11
Εικόνα 2.4: Αρχιτεκτονική μιας Υβριδικής Εφαρμογής σε πλατφόρμα Android [2] .....	16
Εικόνα 2.5: Τυπική διάταξη jQuery mobile σελίδας.....	20
Εικόνα 2.6: Δημιουργία χάρτη με χρήση της βιβλιοθήκης Leaflet.js.....	26
Εικόνα 2.7: Αναπαράσταση λειτουργίας μιας διεπαφής API [39].....	34
Εικόνα 2.8: Στιγμιότυπο υπηρεσίας δρομολόγησης της Εφαρμογής Google Maps 1 .....	37
Εικόνα 2.9: Στιγμιότυπο υπηρεσίας δρομολόγησης της Εφαρμογής Google Maps 2 .....	37
Εικόνα 2.10: Στιγμιότυπο Εφαρμογής Moovit 1.....	38
Εικόνα 2.11: Στιγμιότυπο Εφαρμογής Moovit 2.....	38
Εικόνα 2.12: Στιγμιότυπο Εφαρμογής Moovit 3.....	38
Εικόνα 2.13: Στιγμιότυπο Εφαρμογής Optitrans 1 .....	39
Εικόνα 2.14: Στιγμιότυπο Εφαρμογής Optitrans 2 .....	39
Εικόνα 2.15: Στιγμιότυπο Εφαρμογής Optitrans 3 .....	39
Εικόνα 3.1: Σχεδιάγραμμα αρχικής οθόνης 1 .....	45
Εικόνα 3.2: Σχεδιάγραμμα αρχικής οθόνης 2.....	45
Εικόνα 3.3: Σχεδιάγραμμα οθόνης γραμμών 1 .....	46
Εικόνα 3.4: Σχεδιάγραμμα οθόνης γραμμών 2.....	46
Εικόνα 3.5: Σχεδιάγραμμα οθόνης γραμμών 3.....	46
Εικόνα 3.6: Σχεδιάγραμμα οθόνης οργάνωσης ταξιδιού.....	47
Εικόνα 3.7: Σχεδιάγραμμα οθόνης επιλογής λεωφορείου .....	48
Εικόνα 3.8: Σχεδιάγραμμα οθόνης επεξεργασίας αγαπημένων .....	49
Εικόνα 3.9: Σχεδιάγραμμα οθόνης χαρτών.....	49
Εικόνα 3.10: Σχεδιασμός Popur στάσης .....	50
Εικόνα 3.11: Σχεδιασμός Popur κοντινής στάσης.....	50
Εικόνα 3.12: Σχεδιασμός Popur λεωφορείου .....	50
Εικόνα 3.13: Δημιουργία λίστας με χρήση της μηχανής templating του Underscore.js..	55
Εικόνα 3.14: Κωδικοποίηση αρχείων .....	56
Εικόνα 3.15: Στιγμιότυπα αρχικής οθόνης .....	73
Εικόνα 3.16: Στιγμιότυπα διαχείρισης αγαπημένων.....	74
Εικόνα 3.17: Στιγμιότυπα υλοποίησης της λειτουργίας των Γραμμών 1.....	75
Εικόνα 3.18: Στιγμιότυπα υλοποίησης της λειτουργίας των Γραμμών 2.....	75
Εικόνα 3.19: Στιγμιότυπα υλοποίησης της λειτουργίας των Γραμμών 3.....	76
Εικόνα 3.20: Στιγμιότυπα υλοποίησης της λειτουργίας των Στάσεων .....	77
Εικόνα 3.21: Στιγμιότυπα υλοποίησης της λειτουργίας Οργάνωσης Ταξιδιού 1.....	78
Εικόνα 3.22: Στιγμιότυπα υλοποίησης της λειτουργίας των Κοντινών Στάσεων .....	79
Εικόνα 3.23: Στιγμιότυπα υλοποίησης της λειτουργίας Παρακολούθησης Γραμμής.....	80



Εικόνα 3.24: Στιγμιότυπα υλοποίησης της λειτουργίας Παρακολούθησης Λεωφ. 1.....	81
Εικόνα 3.25: Στιγμιότυπα υλοποίησης της λειτουργίας Παρακολούθησης Λεωφ. 2.....	81
Εικόνα 3.26: Στιγμιότυπα υλοποίησης της λειτουργίας Χαρτών εκτός σύνδεσης.....	82

### Κατάλογος Πινάκων

Πίνακας 2.1: Τυπικά όρια μιας PostgreSQL βάσης δεδομένων [33].....	30
Πίνακας 2.2: Αρχεία ενός GTFS feed [42].....	36
Πίνακας 3.1: Οντότητες σχετικές με το δίκτυο.....	59
Πίνακας 3.2: Οντότητες σχετικές με την λειτουργία σε πραγματικό χρόνο.....	60
Πίνακας 3.3: Οντότητες σχετικές με την λειτουργία αντίστροφης γεωκωδικοποίησης.....	61
Πίνακας 3.4: Πίνακας Συσχετίσεων.....	62
Πίνακας 3.5: Συσχετίσεις βάσης δεδομένων στην πλευρά του πελάτη.....	64

### Κατάλογος Διαγραμμάτων

Διάγραμμα 2.1: Τυπική μορφή ενός MVC πλαισίου [2].....	21
Διάγραμμα 2.2: Λειτουργία του MV* πλαισίου Backbonejs [21].....	23
Διάγραμμα 2.3: Ασύγχρονη φόρτωση με χρήση του πλαισίου Require.js [27].....	25
Διάγραμμα 2.4: Σειρά ενεργειών κατά την κλήση ενός κώδικα PHP [31].....	29
Διάγραμμα 3.1: Αρχιτεκτονική συστήματος.....	43
Διάγραμμα 3.2: Αναπαράσταση λειτουργίας εφαρμογής.....	44
Διάγραμμα 3.3: Αναπαράσταση ανταλλαγής δεδομένων κατά την πρώτη έναρξη.....	45
Διάγραμμα 3.4: Σχηματική αναπαράσταση λειτουργίας του πλαισίου Backbonejs.....	51
Διάγραμμα 3.5: E-R Model βάσης δεδομένων.....	63
Διάγραμμα 3.6: Σχεσιακό Μοντέλο βάσης δεδομένων.....	63
Διάγραμμα 3.7: E-R model βάσης δεδομένων στην πλευρά του πελάτη.....	65
Διάγραμμα 3.8: Σχεσιακό Μοντέλο βάσης στην πλευρά του πελάτη.....	65

### Κατάλογος Κώδικα

Κώδικας 2.1: Cordova - Geolocation API.....	12
Κώδικας 2.2: Cordova - Network Information API.....	13
Κώδικας 2.3: Cordova - SQLite API.....	14
Κώδικας 2.4: Τυπική διάταξη jQuery mobile σελίδας.....	20
Κώδικας 2.5: Παράδειγμα χρήσης Underscore.js Template Engine.....	24
Κώδικας 2.6: Δημιουργία χάρτη με χρήση της βιβλιοθήκης Leaflet.js.....	26
Κώδικας 2.7 Τυπικό ερώτημα αναζήτησης δεδομένων μίας βάσης σε γλώσσα SQL.....	32
Κώδικας 2.8: Παράδειγμα του πρότυπου JSON.....	35

Κώδικας 3.1: Backbonejs Router.....	52
Κώδικας 3.2: Backbonejs View.....	52
Κώδικας 3.3: HTML template σελίδας jQuery Mobile .....	53
Κώδικας 3.4: Λήψη δεδομένων στις Collections του Backbonejs 1 .....	54
Κώδικας 3.5: Λήψη δεδομένων στις Collections του Backbonejs 2 .....	54
Κώδικας 3.6: Παράδειγμα Backbonejs Model.....	54
Κώδικας 3.7: Δημιουργία λίστας με χρήση της βιβλιοθήκης Underscore.js.....	55
Κώδικας 3.8: Template δημιουργία λίστας με χρήση του Underscore.js.....	55
Κώδικας 3.9: Δεδομένα JSON.....	55
Κώδικας 3.10: Υποβολή ερωτήματος σε τοπική βάση δεδομένων .....	67

# 1. Εισαγωγή

## 1.1 Στόχος

Στην Ελλάδα λειτουργούν 59 ΚΤΕΛ (με τη μορφή ανωνύμων εταιρειών), άλλα με αρκετά μεγάλους στόλους και άλλα με μικρότερους. Το υπεραστικό ΚΤΕΛ Χανίων – Ρεθύμνου αποτελεί ένα από τα μεγαλύτερα της χώρας, με στόλο πάνω από 200 οχήματα και ετήσιο αριθμό εισιτηρίων πάνω από 3 εκατομμύρια. Το έργο του εκτελείται σε πλήρης λειτουργία 12 μήνες το χρόνο με το επιβατικό κοινό του εν λόγω ΚΤΕΛ πέρα από τους καθημερινούς επιβάτες του να εξυπηρετεί ένα μεγάλο αριθμό επισκεπτών του νησιού κατά τους καλοκαιρινούς μήνες, ενώ κατά τη διάρκεια του χειμώνα το κενό των τουριστών συμπληρώνεται από την μετακίνηση των μαθητών.

Στα ελληνικά καταστήματα εφαρμογών θα βρει κάποιος μια πληθώρα εφαρμογών με αντικείμενο την μετακίνηση στο δίκτυο συγκοινωνιών Αθήνας και Θεσσαλονίκης (όπως τα Moovit, Ortitrans, Google Maps). Οι εφαρμογές αυτές πραγματεύονται την απεικόνιση του δικτύου, τη διάθεση των δρομολογίων και την οργάνωση και βελτιστοποίηση του ταξιδιού με χρήση κυρίως, όχι όμως αποκλειστικά, των μέσων μαζικής μεταφοράς της πόλης (δυνατότητα χρήσης και ταξί ή ποδήλατου).

Ωστόσο, στον ελλαδικό χώρο δεν έχει κάνει την εμφάνιση της μια εφαρμογή που να προσδίδει έμφαση στην διάρθρωση του δικτύου και την ζωντανή πληροφόρηση σχετικά με την κίνηση του στόλου των υπεραστικών συγκοινωνιών. Παρόλο που σχεδόν όλα τα υπεραστικά ΚΤΕΛ χρησιμοποιούν ένα σύστημα ζωντανής παρακολούθησης στόλου, δεν έχουν προχωρήσει στην διάθεση αυτής της πληροφορίας στον τελικό χρήστη.

Στο υπό εξέταση υπεραστικό δίκτυο, ένα σημαντικό ποσοστό των γραμμών έχει συχνότητα κατά τους χειμερινούς μήνες μία με δύο φορές την ημέρα ή ακόμα και μερικές φορές την εβδομάδα, πράγμα που σημαίνει ότι ο επιβάτης της συγκεκριμένης γραμμής θα έχει πολύωρη αναμονή ή και αδυναμία μετακίνησης σε περίπτωση που δεν προλάβει την διέλευση του επιθυμητού δρομολογίου. Ακόμα και με την έγκαιρη προσέλευση του χρήστη στη στάση, το μεγάλο πλήθος στάσεων, η μεγάλη διάρκεια δρομολογίου και αρκετοί άλλοι παράγοντες μπορεί να επηρεάσουν την εκτιμώμενη ώρα άφιξης. Επίσης, σε αρκετές άγονες γραμμές, και ιδιαίτερα τους χειμερινούς μήνες το δίκτυο χρησιμοποιείται από ελάχιστους χρήστες, με αποτέλεσμα να περιορίζονται οι πιθανότητες να ενημερωθεί ο επιβάτης που περιμένει στη στάση αν το λεωφορείο έχει διέλθει ή όχι. Όλα τα παραπάνω και αρκετοί ακόμα ειδικότεροι λόγοι του κάθε επιβάτη, καθιστούν την χρήση ενός δυναμικού συστήματος πληροφόρησης σε ένα τέτοιο δίκτυο απαραίτητη για την βελτίωση της εξυπηρέτησης του.

Ειδικότερα στην Κρήτη, τους καλοκαιρινούς μήνες στο δίκτυο των συγκοινωνιών προστίθεται ένας σημαντικός αριθμός επισκεπτών ο οποίος στο μεγαλύτερο του ποσοστό χρησιμοποιεί τα ΜΜΜ για την μετακίνηση του. Κυρίως οι επισκέπτες από το εξωτερικό δεν έχουν γνώση για την διάρθρωση του δικτύου. Οι στάσεις, λόγω του εύρους της περιοχής που εξυπηρετούν (πολλές φορές μια σε κάθε οικισμό) καθιστούν δύσκολη την εύρεση και προσέγγιση της πλησιέστερης από τους χρήστες σε αυτές χωρίς κάποια καθοδήγηση. Η χρήση smartphone και των δυνατοτήτων εντοπισμού τοποθεσίας που ενσωματώνουν, μπορούν να διευκολύνουν την προσέγγιση των χρηστών στην πλησιέστερη σε αυτούς στάση.

Ένα ακόμα σημαντικό στοιχείο σχετικά με τα δίκτυα υπεραστικών συγκοινωνιών είναι ότι εξυπηρετούν μεγάλο αριθμό στάσεων και γραμμών, κάτι που συνεπάγεται μεγάλο όγκο δεδομένων. Οι επισκέπτες από το εξωτερικό, και όχι μόνο, δεν έχουν την ευχέρεια χρήσης απεριόριστων δεδομένων κινητής τηλεφωνίας. Με δεδομένο το παραπάνω και το γεγονός ότι ο χρήστης θα κάνει χρήση της εφαρμογής όταν βρίσκεται «στον δρόμο» (εκτός κάποιου WiFi), είναι σημαντικό το μεγαλύτερο μέρος των απαιτούμενων δεδομένων, αλλά και οι σημαντικότερες λειτουργίες να είναι διαθέσιμες και εκτός σύνδεσης.

## 1.2 Επιλογές

Η ανάπτυξη μίας εφαρμογής για τη διάθεση πληροφοριών στους χρήστες της μπορεί να επιτευχθεί με διάφορους τρόπους από τους προγραμματιστές ανάλογα με τις απαιτήσεις του τελικού συστήματος. Οι χρήστες μπορούν να χρησιμοποιήσουν είτε τον Browser του τηλεφώνου τους και να πάρουν τις απαιτούμενες πληροφορίες από την ιστοσελίδα του παρόχου, είτε να κατεβάσουν και να εγκαταστήσουν μια εφαρμογή από το κατάστημα εφαρμογών της πλατφόρμας. Οι εφαρμογές που βρίσκει ο χρήστης σε κάποιο κατάστημα εφαρμογών διαχωρίζονται σε δύο κατηγορίες ανάλογα με τον τρόπο ανάπτυξης τους, τις Native εφαρμογές (Native Apps) και τις Υβριδικές εφαρμογές (Hybrid Apps). Οι εφαρμογές της πρώτης κατηγορίας είναι γραμμένες σε γλώσσες που εξαρτώνται από το λειτουργικό της συσκευής, πράγμα που σημαίνει ότι είναι διαθέσιμες μόνο στο λειτουργικό που έχουν αναπτυχθεί. Από την άλλη μεριά οι εφαρμογές της δεύτερης κατηγορίας είναι γραμμένες σε κοινές γλώσσες διαδικτύου (HTML, JavaScript και CSS) οι οποίες «συσκευάζονται» σε Native και διατίθενται στο εκάστοτε κατάστημα.

Στην περίπτωση της εν λόγω εφαρμογής η οποία απευθύνεται σε ευρύ κοινό απαιτείται η κάλυψη όσο το δυνατόν περισσότερου αριθμού χρηστών. Το βασικό πλεονέκτημα της υβριδικής προσέγγισης είναι η ευκολία δημιουργίας μίας εφαρμογής και διανομής της σε πολλαπλές πλατφόρμες (Android, iOS, Windows Phone, Blackberry και λοιπά). Ωστόσο, η ανάπτυξη τους είναι απαγορευτική όταν απαιτούνται αρκετοί υπολογιστικοί πόροι από τη πλευρά του πελάτη (client-side) για την επίλυση πολύπλοκων προβλημάτων. Η συγκεκριμένη εφαρμογή αποτελεί μια διεπαφή που επικοινωνεί με τον εξυπηρετητή ο οποίος και επιλύει τα πολύπλοκα προβλήματα, με απλές μόνο διαδικασίες να συμβαίνουν στην πλευρά του πελάτη.

Η ανάπτυξη μιας υβριδικής εφαρμογής κάνει χρήση, όπως προαναφέρθηκε, τεχνολογιών διαδικτύου (HTML, JavaScript, CSS) με χρήση διάφορων βιβλιοθηκών της JavaScript (jQuery Mobile, Backbone.js και άλλες). Η χρήση τέτοιων βιβλιοθηκών προσθέτουν επιπλέον λειτουργικότητα για την δημιουργία μιας φιλικής διεπαφής χρήστη σε κινητές συσκευές, υποστήριξη οθονών αφής, προσομοίωση των λειτουργιών των native εφαρμογών και την καλύτερη διαχείριση των πόρων μίας κινητής συσκευής με περιορισμένες πολλές φορές υπολογιστικές δυνατότητες.

Η αναπαράσταση γεωχωρικής πληροφορίας σε διαδικτυακές εφαρμογές απαιτεί τη χρήση και αντίστοιχης βιβλιοθήκης για την απεικόνιση του χαρτογραφικού υπόβαθρου και των χωρικών δεδομένων σε αυτό. Μεταξύ των διαθέσιμων υπάρχουν και κάποιες που επικεντρώνονται στην δημιουργία διαδικτυακών χαρτών για κινητά τηλέφωνα, όπως η ανοιχτού κώδικα βιβλιοθήκη Leaflet.js, ενώ το χαρτογραφικό υπόβαθρο μπορεί να επιλεγεί μέσα από αρκετές εναλλακτικές Google Maps, Bing Maps, Open Street Maps.

Τέλος, όσον αφορά το client-side μέρος της εφαρμογής, ένα πλαίσιο ανοιχτού κώδικα, το Cordova (open-source έκδοση του δημοφιλούς PhoneGap), είναι υπεύθυνο για την ανάπτυξη δια-πλατφορμικών mobile εφαρμογών με το «πακετάρισμα» μίας εφαρμογής γραμμένης σε γλώσσες διαδικτύου και την διάθεση της στο αντίστοιχο της πλατφόρμας κατάσταση. Το πλαίσιο αυτό δίνει την δυνατότητα αξιοποίησης των λειτουργιών της συσκευής, όπως η κάμερα ή ο αποθηκευτικός χώρος της συσκευής. Το αποτέλεσμα σε επίπεδο διεπαφής χρήστη είναι ουσιαστικά ένας Browser στον οποίο προβάλλεται το HTML περιεχόμενο σε πλήρη οθόνη.

Το σημαντικότερο κομμάτι σε μία εφαρμογή για την πληροφόρηση των χρηστών MMM είναι η συγκέντρωση και διαχείριση μεγάλου όγκου πληροφορίας. Για το σκοπό αυτών έχουν δημιουργηθεί κοινώς αποδεκτά πρότυπα (πρότυπο GTFS – General Transit Feed Specification) με τα οποία οι φορείς διαθέτουν τα δεδομένα τους. Με τη σειρά τους οι προγραμματιστές, μπορούν να αναπτύσσουν συστήματα με δυνατότητα εφαρμογής σε πολλά διαφορετικά δίκτυα (η εφαρμογή Moovit μέσα σε χρονικό διάστημα τριών χρόνων χρησιμοποιείται σε περισσότερες από 500 πόλεις παγκοσμίως). Στην περίπτωση του ΚΤΕΛ που εξετάστηκε τα δεδομένα δεν ακολουθούσαν το συγκεκριμένο πρότυπο ενώ ήταν αναγκαία και η συγκέντρωση στοιχείων για την βελτίωση της εμπειρίας χρήστη στην εφαρμογή.

Όλα τα δεδομένα βρίσκονται αποθηκευμένα σε μία σχεσιακή βάση δεδομένων PostgreSQL, με εγκατεστημένη την χωρική επέκταση PostGIS για την διαχείριση των χωρικών δεδομένων. Οι πίνακες περιλαμβάνουν πληροφορίες σχετικά με τις στάσεις, τις γραμμές, τα λεωφορεία, τις συνδέσεις γραμμών - στάσεων, τα δρομολόγια αλλά και πληροφορίες που εικονικά για τις ανάγκες της εφαρμογής ανανεώνονται σε πραγματικό χρόνο όπως η τοποθεσία και οι υπηρεσίες των λεωφορείων. Επιπλέον, στην βάση δεδομένων υπάρχουν και βοηθητικοί πίνακες όπως οι πόλεις και οι νομοί της χώρας για λειτουργίες Αντίστροφης Γεωκωδικοποίησης (Reverse Geocoding) που προσφέρονται σε κάποιες λειτουργίες της εφαρμογής.

### 1.3 Διάρθρωση εργασίας

Η εργασία χωρίζεται σε τέσσερις ενότητες. Στην δεύτερη ενότητα της ανασκόπησης της βιβλιογραφίας, αρχικά αναλύονται οι τεχνικές ανάπτυξης εφαρμογών για smartphone, και έπειτα παρουσιάζονται τα πλεονεκτήματα και τα μειονεκτήματα της υβριδικής προσέγγισης έναντι των άλλων. Στη συνέχεια, περιγράφεται λεπτομερώς το πλαίσιο ανάπτυξης υβριδικών εφαρμογών PhoneGap, και η αντίστοιχη open-source έκδοση του Cordova, που χρησιμοποιήθηκε στα πλαίσια της εργασίας. Έπειτα, παρουσιάζεται η αρχιτεκτονική μίας υβριδικής εφαρμογής σε πλατφόρμα Android, αλλά και τα δημοφιλέστερα πρόσθετα των δύο πλαισίων που διατίθενται για την γεφύρωση του χάσματος μεταξύ JavaScript και native στοιχείων της πλατφόρμας με στόχο την προσθήκη επιπλέον λειτουργιών στην εφαρμογή.

Στην συνέχεια της ενότητας περιγράφονται οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής, συγκεκριμένα από την πλευρά του πελάτη (client-side), την πλευρά του εξυπηρετητή (server-side) αλλά και οι τεχνολογίες επικοινωνίας χρήστη - εξυπηρετητή. Αναλυτικότερα, από την πλευρά του πελάτη περιγράφονται οι γλώσσες που χρησιμοποιούνται στην ανάπτυξη υβριδικών εφαρμογών, HTML, JavaScript και CSS, και εν συνεχεία αναλύονται οι βιβλιοθήκες JavaScript που χρησιμοποιήθηκαν. Από την πλευρά του εξυπηρετητή γίνεται αναφορά στον εξυπηρετητή NGINX, τη γλώσσα επικοινωνίας του εξυπηρετητή - PHP, το σύστημα βάσεων δεδομένων PostgreSQL με τις χωρικές λειτουργίες του και την γλώσσα SQL. Σχετικά με τις τεχνολογίες επικοινωνίας δίνεται ο ορισμός της τεχνικής ασύγχρονης ανταλλαγής δεδομένων (AJAX) και της διεπαφής προγραμματισμού εφαρμογών - API ενώ περιγράφεται η μορφή του πρότυπου JSON που χρησιμοποιήθηκε για την ανταλλαγή των δεδομένων. Η ενότητα αυτή κλείνει με αναφορά στο πρότυπο GTFS και στις σημαντικότερες εφαρμογές που πραγματεύονται το θέμα των μετακινήσεων με MMM και έχουν βρει εφαρμογή στην χώρα.

Η Τρίτη ενότητα του τόμου είναι εξ' ολοκλήρου αφιερωμένη στη μεθοδολογία ανάπτυξης της εφαρμογής. Συγκεκριμένα, αρχικά παρουσιάζονται οι απαιτήσεις και η αρχιτεκτονική της εφαρμογής, και κατόπιν περιγράφεται ο σχεδιασμός και η υλοποίηση της διεπαφής χρήστη με εκτενή αναφορά στην κάθε λειτουργία. Έπειτα, αναλύεται η καρδιά του συστήματος, και η βάση δεδομένων του. Περιγράφεται, ο σχεδιασμός του σχήματος της βάσης, οι οντότητες και οι συσχετίσεις μεταξύ αυτών. Στην επόμενη παράγραφο επεξηγείται η διαδικασία λειτουργίας και επικοινωνίας μεταξύ των επιμέρους συστημάτων, ενώ η ενότητα κλείνει με παρουσίαση των δυνατοτήτων της εφαρμογής που αναπτύχθηκε.

Στην τελευταία ενότητα της εργασίας, αρχικά παρουσιάζονται τα συμπεράσματα και η αξιολόγηση του συστήματος, και έπειτα οι προοπτικές και οι βελτιώσεις που θα μπορούσε να έχει η συγκεκριμένη εφαρμογή.

## 2. Ανασκόπηση Βιβλιογραφίας

Στην ενότητα της ανασκόπησης βιβλιογραφίας παρουσιάζονται οι βασικές τεχνολογίες για την ανάπτυξη της εφαρμογής, ώστε κάποιος με γενικές γνώσεις στους τομείς των διαδικτυακών τεχνολογιών και των Βάσεων δεδομένων να είναι σε θέση να κατανοήσει τη ροή της εργασίας.

### 2.1 Κατηγορίες Εφαρμογών

Οι διαθέσιμες εφαρμογές στα κυρίαρχα καταστήματα εφαρμογών για το 2014 είναι περίπου 3.2 εκ. [1], ενώ οι λήψεις εκτιμώνται στα 179 δις [1]. Τα τελευταία χρόνια όλο και περισσότερα λειτουργικά για smartphones κάνουν την εμφάνιση τους (Android, iOS, Windows Phone, Firefox OS, Sailfish OS, Tizen, Ubuntu Touch, BlackBerry, Symbian, Windows Mobile, Palm OS, Bada και άλλα). Κάθε ένα από αυτά χρησιμοποιεί την δική του γλώσσα ανάπτυξης, κάτι που απαιτεί επαναλαμβανόμενες διαδικασίες από τους προγραμματιστές για την δημιουργία δια-πλατφορμικών εφαρμογών.

Σε γενικές γραμμές οι εφαρμογές που προορίζονται για χρήση σε πλατφόρμες κινητών συσκευών θα μπορούσαν να ταξινομηθούν σε τρεις διακριτές κατηγορίες, ανάλογα με τον τρόπο ανάπτυξης και λειτουργία τους. Αυτές είναι οι Native Εφαρμογές, οι Web Εφαρμογές και οι Υβριδικές Εφαρμογές όπου αναλύονται στις παραγράφους που ακολουθούν [2].

#### 2.1.1 Native εφαρμογές

Οι Native Εφαρμογές είναι οι πιο κοινές εφαρμογές που διατίθενται σήμερα στα καταστήματα εφαρμογών. Οι εφαρμογές αυτές αναπτύσσονται συνήθως χρησιμοποιώντας τις υψηλότερου επιπέδου γλώσσες προγραμματισμού, όπως η Java για το Android, Objective-C ή Swift για το iOS, ή C# για τα Windows Phone.

Τα APIs της εκάστοτε πλατφόρμας παρέχουν στις native εφαρμογές βέλτιστη πρόσβαση σε δυνατότητες της συσκευής, όπως είναι η κάμερα της συσκευής, το Bluetooth ή το GPS. Επιπλέον, οι χρήστες μπορούν να χρησιμοποιούν αυτές τις εφαρμογές χωρίς σύνδεση στο Internet. Από την άλλη πλευρά, δεδομένου ότι η πλατφόρμα SDK (Software Development Kit) που βασίζονται είναι σε διαφορετικές γλώσσες προγραμματισμού, οι προγραμματιστές πρέπει να συντάξουν πολλαπλές εκδοχές της ίδιας εφαρμογής για τα διαφορετικά καταστήματα εφαρμογών κάτι που καθιστά αρκετά δαπανηρή την ανάπτυξη τους. Η ανάπτυξη Native εφαρμογών ενδείκνυται όταν είναι ιδιαίτερα σημαντική η βέλτιστη της απόδοσης και απαιτείται η επίλυση πολύπλοκων προβλημάτων από την πλευρά του πελάτη.

Για την ανάπτυξη των native εφαρμογών παρέχονται ολοκληρωμένα περιβάλλοντα ανάπτυξης, IDE (Integrated Development Environment), όπως το Xcode από την Apple και το



Android SDK από την Google, με ένα φιλικό UI, εργαλείο αποσφαλμάτωσης (debugger), βιβλιοθήκες εργαλείων και προσομοιωτή συσκευής (Virtual Device). Καθώς η ανάπτυξη εφαρμογών σε γλώσσες όπως η JAVA και η objective C παρουσιάζει αρκετές ιδιαιτερότητες η χρήση των παραπάνω εργαλείων καθίσταται απαραίτητη.

Σε αυτό το σημείο πρέπει να γίνει αναφορά σε μία νέα τάση στην ανάπτυξη δια-πλατφορμικών Native εφαρμογών σε γλώσσα C# για το κοινό κομμάτι του κώδικα, με χρήση του λογισμικού Xamarin [3]. Το Xamarin επιτρέπει την δημιουργία Android, iOS και Windows native εφαρμογών που μοιράζονται τον ίδιο κώδικα στην λογική της εφαρμογής και σε στοιχεία της διεπαφής χρήστη σε ποσοστό που μπορεί να φτάσει το 75%. Επιπλέον, με χρήση του εργαλείου Xamarin.Forms [3] για την δημιουργία της διεπαφής χρήστη το ποσοστό του κοινού κώδικα μπορεί να φτάσει κοντά στο 100%.

### 2.1.2 Web εφαρμογές

Οι διαδικτυακές εφαρμογές (web apps) για κινητά είναι ιστοσελίδες που έχουν σχεδιαστεί για την προβολή τους σε οθόνες κινητών τηλεφώνων. Συνήθως φαίνονται ίδιες σε όλες τις πλατφόρμες και δεν κάνουν χρήση των APIs της πλατφόρμας. Η ανάπτυξη τους είναι πανομοιότυπη με την ανάπτυξη ιστοσελίδων διαδικτύου, με την προσθήκη ιδιαίτερων CSS και βιβλιοθηκών HTML5 Mobile UI, ώστε η χρήση τους να είναι λειτουργικότερη σε μία οθόνη αφής μικρότερων διαστάσεων.

Μια επιμέρους κατηγορία των Web Εφαρμογών είναι και οι Ειδικές web εφαρμογές (Dedicated Web Apps). Είναι εφαρμογές διαδικτύου της παραπάνω κατηγορίας οι οποίες έχουν προσαρμοστεί για μια συγκεκριμένη πλατφόρμα, όπως το Android, το iOS ή το Blackberry, παρέχοντας την αίσθηση των native εφαρμογών στην εκάστοτε πλατφόρμα.

Οι Web Εφαρμογές για κινητά μπορούν να αναπτυχθούν με τη χρήση κοινών τεχνολογιών εξυπηρητητή, όπως NodeJS, PHP και Ruby on Rails. Συνήθως, η πρόσβαση στην εφαρμογή γίνεται πληκτρολογώντας μία διεύθυνση URL στον Browser του κινητού. Τα γραφικά και οι πόροι, συμπεριλαμβανομένων ενδεικτικά αλλά όχι περιοριστικά των πολυμέσων βρίσκονται στον εξυπηρητητή. Ένα μειονέκτημα αυτής της προσέγγισης είναι ότι η λήψη αυτών των στοιχείων στη συσκευή μπορεί όχι μόνο να αυξήσει το κόστος που συνδέεται με τη χρήση δεδομένων, αλλά μπορεί επίσης να επηρεάσει την εμπειρία του χρήστη λόγω αδρανειών που εμπλέκονται σε τέτοια δίκτυα.

### 2.1.3 Υβριδικές εφαρμογές

Οι «Υβριδικές» εφαρμογές (Hybrid Application) αποτελούν μια ειδική κατηγορία διαδικτυακών εφαρμογών που επεκτείνουν τις δυνατότητες τους μέσω της χρήσης των APIs της πλατφόρμας [2]. Είναι εφαρμογές οι οποίες δεν προορίζονται αποκλειστικά για κινητά (Native





## 2.2 Υβριδικές εφαρμογές

Οι υβριδικές εφαρμογές αποτελούν μια ειδική κατηγορία εφαρμογών που θα συναντήσει κάποιος στα καταστήματα εφαρμογών. Ωστόσο η ανάπτυξη τους συνιστάται σε ειδικές περιπτώσεις. Οι δυνατότητες τους είναι περιορισμένες και αρκετές φορές η χρήση τους μπορεί να οδηγήσει σε πολύπλοκες εφαρμογές με αδυναμία λειτουργίας σε συσκευές με περιορισμένη υπολογιστική ισχύς. Για το λόγο αυτό πριν κάποιος ξεκινήσει την ανάπτυξη μιας εφαρμογής με τη συγκεκριμένη προσέγγιση θα πρέπει να έχει προσδιορίσει επακριβώς ποιες θα είναι οι λειτουργίες και οι απαιτήσεις της εφαρμογής. Στο παρών κεφάλαιο αρχικά παρουσιάζονται τα πλεονεκτήματα και μειονεκτήματα της ανάπτυξης υβριδικών εφαρμογών έναντι των άλλων, έπειτα περιγράφεται το πλαίσιο Cordova που είναι υπεύθυνο για την δημιουργία της υβριδικής εφαρμογής και τέλος παρουσιάζεται η αρχιτεκτονική μιας υβριδικής εφαρμογής σε μια πλατφόρμα Android.

### 2.2.1 Πλεονεκτήματα – Μειονεκτήματα Υβριδικής Προσέγγισης

Τα πλεονεκτήματα από την ανάπτυξη υβριδικών εφαρμογών σε σύγκριση με τις native εφαρμογές, θα μπορούσαν να συνοψιστούν στα παρακάτω [2] [4]:

**Ταχύτερη ανάπτυξη και πρόσβαση στα καταστήματα εφαρμογών.** Ο χρόνος που απαιτείται για την ανάπτυξη μιας υβριδικής εφαρμογής είναι συνήθως αρκετά μικρότερος και απαιτεί στο μεγαλύτερο βαθμό επαναχρησιμοποιήσιμα πρότυπα και δεξιότητες σε σύγκριση με τις γλώσσες προγραμματισμού που απαιτούνται στην ανάπτυξη native εφαρμογών.

**Ανέξοδη ανάπτυξη δια-πλατφορμικών εφαρμογών** Η δια-πλατφορμική συμβατότητα, μειώνει την απαίτηση του native κώδικα που απαιτείται, με αποτέλεσμα ένα πιο επαναχρησιμοποιήσιμο κώδικα που μπορεί να μοιραστεί και να αναπτυχθεί σε όλες τις πλατφόρμες με ελάχιστη προσαρμογή. Η ανάπτυξη δια-πλατφορμικών εφαρμογών, επίσης μπορεί να βοηθήσει στην ελαχιστοποίηση του κόστους που συνδέεται με την ανάπτυξη και δοκιμή. Η δυνατότητα επαναχρησιμοποίησης του κώδικα επιτρέπει στους προγραμματιστές να δημιουργήσουν μια κοινή επαναχρησιμοποιήσιμη αρχιτεκτονική. Στις native εφαρμογές, από την άλλη πλευρά, απαιτείται από τους προγραμματιστές να εκτελέσουν πλήρη έλεγχο των δυνατοτήτων σε όλες τις πλατφόρμες στις οποίες αναπτύσσεται η εφαρμογή.

**Ανθρώπινο δυναμικό** Οι υβριδικές εφαρμογές δημιουργούνται με τεχνολογίες διαδικτύου, πράγμα που σημαίνει ότι υπάρχουν πολλοί προγραμματιστές που έχουν τις δεξιότητες για την ανάπτυξη εφαρμογών αυτής της κατηγορίας.

**Κόστος συντήρησης** Το κόστος συντήρησης είναι συνήθως χαμηλότερο καθώς κατά τις αναβαθμίσεις δεν χρειάζεται να γράφει ο κώδικας της εφαρμογής στη native γλώσσα της κάθε πλατφόρμας.

**Διαδικασία έγκρισης** Τα περισσότερα από τα καταστήματα εφαρμογών έχουν μια διαδικασία έγκρισης για την οποία κάθε εφαρμογή πρέπει να πληροί κάποιες προϋποθέσεις ώστε να μπορεί να γίνει διαθέσιμη στο εκάστοτε κατάστημα εφαρμογών. Επειδή μία υβριδική εφαρμογή μπορεί να ενημερωθεί και έξω από τα όρια ενός καταστήματος εφαρμογών, συνήθως δεν χρειάζεται να γίνετε υποβολή της εφαρμογής σε κάθε ανανέωση. Μόλις εγκριθεί μία εφαρμογή, μπορεί να φορτώσει τις μετέπειτα ενημερώσεις ανεξάρτητα, μέσω του εξυπηρετητή.

**Οι υβριδικές εφαρμογές είναι το μέλλον** κοιτάζοντας προς το μέλλον και τις επερχόμενες εξελίξεις στα λειτουργικά κινητών τεχνολογιών, μπορεί κανείς εύκολα να υποστηρίξει ότι η υβριδική προσέγγιση ανάπτυξης εφαρμογών είναι το μέλλον στον τομέα των mobile apps.

Η γλώσσα ανάπτυξης μιας εφαρμογής είναι ένα βασικό κριτήριο για την επιλογή μεθόδου. Η διαδικασία ανάπτυξης μιας native εφαρμογής διαφέρει αρκετά από την ανάπτυξη μιας αντίστοιχης υβριδικής. Από την άλλη μεριά, η ανάπτυξη διαδικτυακών και υβριδικών εφαρμογών είναι πανομοιότυπη. Συνεπώς, εδώ υπάρχουν και άλλα σημαντικότερα κριτήρια στην επιλογή του προγραμματιστή αν θα διαθέσει την εφαρμογή του μέσα από ένα Browser ή μέσω κάποιου καταστήματος εφαρμογών.

Τα πλεονεκτήματα ανάπτυξης υβριδικών εφαρμογών σε σύγκριση με τις διαδικτυακές εφαρμογές [2]:

**Πρόσβαση στις δυνατότητες της συσκευής** Όπως αναφέρθηκε, οι υβριδικές εφαρμογές επεκτείνουν τις δυνατότητες του περιβάλλοντος της JavaScript για πρόσβαση στα APIs της συσκευής τα οποία δεν είναι διαθέσιμα μέσω του Browser.

**Μη διαθέσιμες νέες δυνατότητες** Οι υβριδικές εφαρμογές μπορούν να χρησιμοποιήσουν τα νέα χαρακτηριστικά που διατίθενται σε μεταγενέστερες εκδόσεις του λειτουργικού.

**Διανομή μέσω καταστημάτων** Οι υβριδικές εφαρμογές διανέμονται μέσω των καταστημάτων εφαρμογών, όπως ακριβώς και οι native εφαρμογές. Συνεπώς, υπάρχει η δυνατότητα να μεταφερθούν στη συσκευή και να εγκατασταθούν. Ως εκ τούτου, ο προγραμματιστής, μπορεί να προωθήσει την εφαρμογή μέσα από τα εκάστοτε καταστήματα, ενώ είναι δυνατή και η διάθεση της εφαρμογής έναντι χρηματικού αντίτιμου.

**Offline πρόσβαση και εκτέλεση** Οι υβριδικές όπως και οι native εφαρμογές, μπορούν να εκτελεστούν τοπικά στη συσκευή όταν αυτή είναι εκτός σύνδεσης.

Ωστόσο, η ανάπτυξη υβριδικών εφαρμογών δεν αποτελεί την λύση σε κάθε πρόβλημα. Τα βασικά μειονεκτήματα των υβριδικών εφαρμογών σε σύγκριση με τις native περιλαμβάνουν [2]:

**Απόδοση** Ενδέχεται να αντιμετωπιστούν πιθανά ζητήματα επιδόσεων καθώς η JavaScript είναι μια γλώσσα μονού νήματος (single-threaded), πράγμα που σημαίνει ότι μόνο μία λειτουργία μπορεί να εκτελεστεί τη στιγμή.

**Διαφορές μεταξύ πλατφόρμων** το WebKit δεν είναι το ίδιο σε όλες τις πλατφόρμες κινητών, αυτό σημαίνει ότι μπορεί να υπάρξουν ακαθόριστες διαφορές μεταξύ αποδόσεων και χαρακτηριστικών.

**Δεν είναι διαθέσιμα προηγμένα χαρακτηριστικά** Μπορεί να υπάρχουν προηγμένα χαρακτηριστικά που δεν μπορούν πάντα να υλοποιηθούν εύκολα σε μία υβριδική εφαρμογή.

**Ασυνεπής διεπαφής χρήστη** Η εμφάνιση και η αίσθηση της διεπαφής χρήστη (UI) μίας πλατφόρμας είναι αρκετά δύσκολο να μιμηθεί χρησιμοποιώντας απλά γλώσσες HTML, CSS και JavaScript.

Τέλος, το βασικό μειονέκτημα μιας υβριδικής εφαρμογής σε σύγκριση με μια web εφαρμογή αφορά [2]:

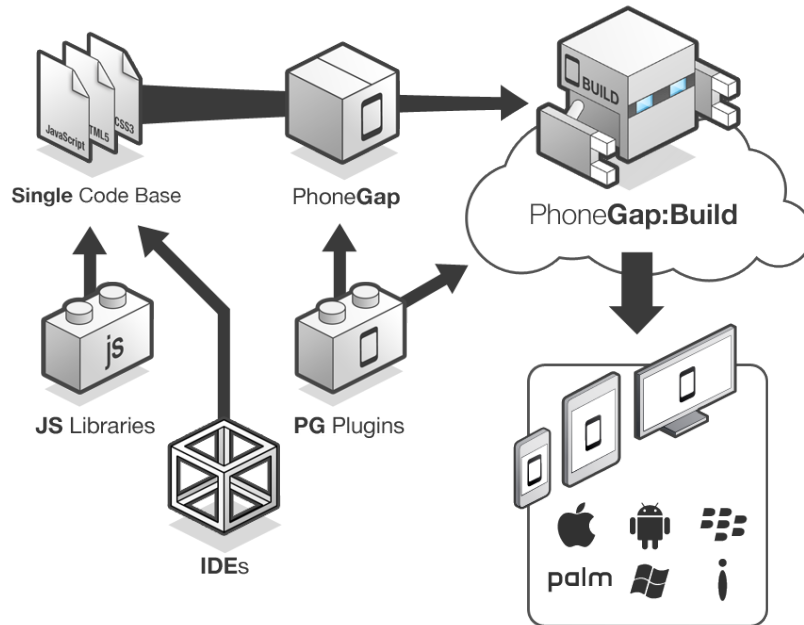
**Μη προσβάσιμη μέσω του Browser** Ο χρήστης καλείται να βρει την εφαρμογή σε ένα κατάστημα εφαρμογών, να την κατεβάσει και να την εγκαταστήσει καθώς δεν μπορεί να έχει πρόσβαση μέσω ενός Browser.

## 2.2.2 Apache Cordova

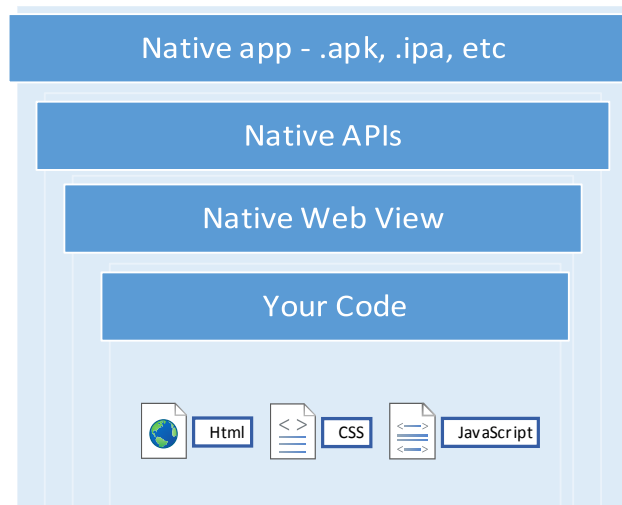
### *Γενικά στοιχεία*

Το 2009 η Nitobi δημιούργησε το PhoneGap, ένα πλαίσιο που επιτρέπει την ανάπτυξη εφαρμογών ανεξαρτήτων πλατφόρμας, χρησιμοποιώντας τεχνολογίες διαδικτύου (JavaScript, HTML5 και CSS3) [6]. Το 2011 η Adobe εξαγόρασε την Nitobi και μαζί με αυτή και το PhoneGap, ενώ ο open-source κώδικας παραχωρήθηκε στην Apache Software Foundation υπό την ονομασία Cordova [7]. Το Cordova δίνει την δυνατότητα σε προγραμματιστές να δημιουργήσουν Υβριδικές εφαρμογές δωρεάν και χωρίς απαραίτητα να έχουν συνδρομή στην Adobe, ενώ τα plugins είναι κοινώς διαθέσιμα μεταξύ των εφαρμογών. Παρόμοια πλαίσια είναι διαθέσιμα όπως το appMobi, Convertigo, Icenium (Telerik), ViziApps, Worklight (IBM).

Μια υβριδική εφαρμογή υλοποιείται ως μια ιστοσελίδα, δηλαδή ως ένα αρχείο index.html, στο οποίο γίνονται αναφορές στα συμπεριλαμβανόμενα αρχεία CSS, JavaScript, εικόνες, αρχεία πολυμέσων ή άλλους πόρους απαραίτητους για να λειτουργήσει. Μια Υβριδική Εφαρμογή αναπτυγμένη με Cordova βασίζεται σε ένα αρχείο config.xml το οποίο περιέχει πληροφορίες σχετικά με την εφαρμογή, όπως τον δημιουργό της, email επικοινωνίας με αυτόν και λοιπά, αλλά και σχετικές παραμέτρους που επηρεάζουν τον τρόπο που αυτή λειτουργεί, παραδείγματος χάρη, πως ανταποκρίνεται σε αλλαγή του προσανατολισμού της συσκευής ή αν επιτρέπεται η λειτουργία πλήρους οθόνης. Συνεπώς, η εφαρμογή εκτελείται ως μια ιστοσελίδα μέσα στο «περιτύλιγμα» μίας native εφαρμογής η οποία διανέμεται στα καταστήματα.



Εικόνα 2.2: Δημιουργία Υβριδικών Εφαρμογών με χρήση του PhoneGap [6]



Εικόνα 2.3: Στοιχεία μιας Υβριδικής Εφαρμογής

### Πρόσθετα

Στις υβριδικές εφαρμογές, όπως προαναφέρθηκε, υπάρχει η δυνατότητα χρήσης των APIs της συσκευής όπως ακριβώς και στις Native εφαρμογές. Μια πληθώρα πρόσθετων γεφυρώνει το χάσμα μεταξύ JavaScript και APIs της συσκευής δίνοντας την δυνατότητα στις υβριδικές εφαρμογές να κάνουν χρήση των βασικών λειτουργιών της όπως για παράδειγμα την κάμερα, πρόσβαση στα αρχεία της συσκευής, στις πληροφορίες δικτύου και μπαταρίας, την

πυξίδα και το GPS. Ορισμένα από τα πρόσθετα προσδίδουν στην εφαρμογή ιδιαίτερα χαρακτηριστικά της εκάστοτε πλατφόρμας όπως είναι το πρόσθετο για native ειδοποιήσεων. Επίσης υπάρχει η δυνατότητα δημιουργίας πρόσθετων και από τους ίδιους προγραμματιστές (Third-party plugins) για πρόσβαση και σε επιπλέον λειτουργίες, όπως η δημιουργία βάσεων δεδομένων SQLite.

Για την πρόσβαση σε πληροφορίες όπως η τοποθεσία, η κατάσταση σύνδεσης, τα αρχεία της συσκευής ο χρήστης ερωτάτε κατά την εγκατάσταση της εφαρμογής. Στη συνέχεια αναλύονται ορισμένα από τα πρόσθετα που κρίνονται απαραίτητα στην ανάπτυξη μιας εφαρμογής ανταλλαγής πληροφοριών με τους χρήστες δικτύου συγκοινωνιών:

### Cordova – Geolocation API

Αυτό το API βασίζεται στις προδιαγραφές του Geolocation API του W3C [8] και παρέχει πληροφορίες σχετικά με τη θέση της συσκευής, όπως τα γεωγραφικό πλάτος και μήκος. Η πληροφορία τοποθεσίας προέρχεται από τα σήματα του δικτύου, όπως η διεύθυνση IP, RFID, WiFi και Bluetooth, MAC διευθύνσεις και κυψέλες του δικτύου GSM / CDMA.

```
// onSuccess Callback
function onSuccess(position) {
    var latitude = position.coords.latitude;
    var longitude = position.coords.longitude;
    var altitude = position.coords.altitude;
    var accuracy = position.coords.accuracy;
    var altitudeAccuracy = position.coords.altitudeAccuracy;
    var heading = position.coords.heading;
    var speed = position.coords.speed;
    var timestamp = position.timestamp;
}
// onError Callback
function onError(error) {
    alert('code: ' + error.code + '\n' +
        'message: ' + error.message + '\n');
}
// Options: throw an error if no update is received every 30
seconds.
var watchID = navigator.geolocation.watchPosition(onSuccess,
onError,
    { timeout: 30000 });
```

Κώδικας 2.1: Cordova - Geolocation API

### Cordova – File API

Το συγκεκριμένο πρόσθετο επιτρέπει κατά βάση την πρόσβαση στα αρχεία της συσκευής. Συγκεκριμένα εκτελεί λειτουργίες όπως είναι η ανταλλαγή αρχείων, η λήψη, αποθήκευση και πρόσβαση αρχείων στη συσκευή.

### Cordova – Network Information API

Το Network Information API παρέχει πρόσβαση στον τύπο της σύνδεσης που χρησιμοποιεί το σύστημα για την πρόσβαση στο διαδίκτυο (δίκτυο κινητής τηλεφωνίας, WiFi, Ethernet και λοιπά) καθώς και ενημέρωση του χρήστη όταν η κατάσταση αλλάξει.

```
function checkConnection() {
  var networkState = navigator.connection.type;
  // network states
  var states = {};
  states [Connection.UNKNOWN] = 'Unknown connection';
  states [Connection.ETHERNET] = 'Ethernet connection';
  states [Connection.WIFI] = 'WiFi connection';
  states [Connection.CELL_2G] = 'Cell 2G connection';
  states [Connection.CELL_3G] = 'Cell 3G connection';
  states [Connection.CELL_4G] = 'Cell 4G connection';
  states [Connection.CELL] = 'Cell generic connection';
  states [Connection.NONE] = 'No network connection';

  alert('Connection type: ' + states [networkState]);
}
checkConnection();
```

Κώδικας 2.2: Cordova - Network Information API

### Cordova - Storage API

Οι διαθέσιμες επιλογές αποθήκευσης που προσφέρονται από το PhoneGap/ Cordova είναι η χρήση του Local Storage και Session Storage της HTML5, καθώς και η δημιουργία βάσεων δεδομένων WebSQL και Indexed DB.

### Cordova – Notification API

Το εκάστοτε πλαίσιο ανάπτυξης διαδικτυακών εφαρμογών για κινητές συσκευές έχει τον δικό του τρόπο προβολής και στυλ ειδοποιήσεων. Ωστόσο, στις native εφαρμογές τα παράθυρα ειδοποιήσεων είναι κοινά και διαμορφώνονται ανάλογα με την έκδοση του λειτουργικού. Με το συγκεκριμένο πρόσθετο οι ειδοποιήσεις έχουν την μορφή που έχουν σε μία native εφαρμογή.

### 3<sup>rd</sup> Party plugin – SQLite Plugin

Το συγκεκριμένο πρόσθετο επιτρέπει την δημιουργία και διαχείριση μιας σχεσιακής βάσης δεδομένων SQLite (προεπιλεγμένο σύστημα που χρησιμοποιείται σε native Android εφαρμογές) [9], [10]. Η χρήση του είναι ιδιαίτερα σημαντική καθώς το W3C έχει εγκαταλείψει το

Project WebSQL [11] που υποστηρίζεται από το επίσημο πρόσθετο Cordova – Storage, ενώ η εναλλακτική που προσφέρει, η Indexed DB, δεν υποστηρίζεται από την πλατφόρμα Android. Επιπλέον, το συγκεκριμένο πρόσθετο δεν έχει περιορισμούς στο μέγεθος της βάσης δεδομένων.

```
var db = window.sqlitePlugin.openDatabase({name: "my.db"});

db.transaction(function(tx) {
  tx.executeSql('DROP TABLE IF EXISTS test_table;');
  tx.executeSql('CREATE TABLE IF NOT EXISTS test_table ('+
    'id integer primary key,'+
    'data text,'+
    'data_num integer);');

  tx.executeSql("INSERT INTO test_table (data, data_num) VALUES (?,?)",
    ["test", 100],
    function (tx, res) {
      db.transaction(function (tx) {
        tx.executeSql("SELECT count(id) AS cnt FROM test_table;",
          function (tx, res) {
            });
      });
    }, function (e) {
      alert("ERROR: " + e.message);
    });
});
```

Κώδικας 2.3: Cordova - SQLite API

Στον Κώδικα 2.3 διακρίνονται οι τρεις παρακάτω μέθοδοι:

- **openDatabase** Άνοιγμα της βάσης δεδομένων. Αν η βάση δεν υπάρχει δημιουργείται εκείνη τη στιγμή.
- **transaction** Αφού ανοίξει η βάση δεδομένων εκτελούνται συναλλαγές σε αυτήν. Το βασικό πλεονέκτημα χρήσης των συναλλαγών είναι η δυνατότητα ακύρωσης τους (rollback) στην περίπτωση που ένα ή περισσότερα από τα επιμέρους ερωτήματα της συναλλαγής αποτύχει. Επίσης, σε μία συναλλαγή διακρίνονται τα *Success* και *Error* callbacks για την διαχείριση των σφαλμάτων.
- **executeSql** Ο SQL κώδικας για την εισαγωγή και ανάκτηση δεδομένων.

### 2.1.3 Αρχιτεκτονική μίας υβριδικής εφαρμογής

Μια υβριδική εφαρμογή συνδυάζει σε ένα επίπεδο τους δύο «κόσμους» της μητρικής γλώσσας της συσκευής και της JavaScript. Ο συνδυασμός αυτός διαφοροποιείται από πλατφόρμα σε πλατφόρμα αλλά δεν υπάρχει καμία συνέπεια σε επίπεδο χρήστη. Στη συνέχεια της παραγράφου περιγράφεται η αρχιτεκτονική μιας υβριδικής εφαρμογής σε πλατφόρμα Android και αναλύονται τα επιμέρους συστατικά της.



Τυπικά σε μια υβριδική εφαρμογή περιλαμβάνονται τα περισσότερα από τα παρακάτω στοιχεία [2]:

**Web View (1)** Μια υβριδική εφαρμογή είναι στην βάση της μια Web εφαρμογή με πρόσβαση στις δυνατότητες της πλατφόρμας. Ως web εφαρμογή συνεπώς απαιτεί κάποιο Browser για την απόδοση του περιεχόμενου και την φιλοξενία της λογικής της εφαρμογής.

**View, model και controller (2)** Καθώς η εφαρμογή είναι κυρίως γραμμένη σε JavaScript, ανάλογα με την βιβλιοθήκη JavaScript που έχει χρησιμοποιηθεί, υπάρχει κάποια μορφή των συστατικών View, Model, και Controller (MVC πλαίσιο).

**JS-Java Bridge (3)** Υπάρχει ένα συνδετικό επίπεδο το οποίο επιτρέπει στα native και web περιβάλλοντα να αλληλοεπιδρούν μεταξύ τους. Η «γέφυρα» μεταξύ αυτών επιτρέπει την εκτέλεση συγχρονισμένων και ασύγχρονων APIs. Αυτό το επίπεδο είναι ένα από τα πιο κρίσιμα μεταξύ άλλων και για την απόδοση, την ευκολία χρήσης και την ασφάλεια σε μία υβριδική εφαρμογή.

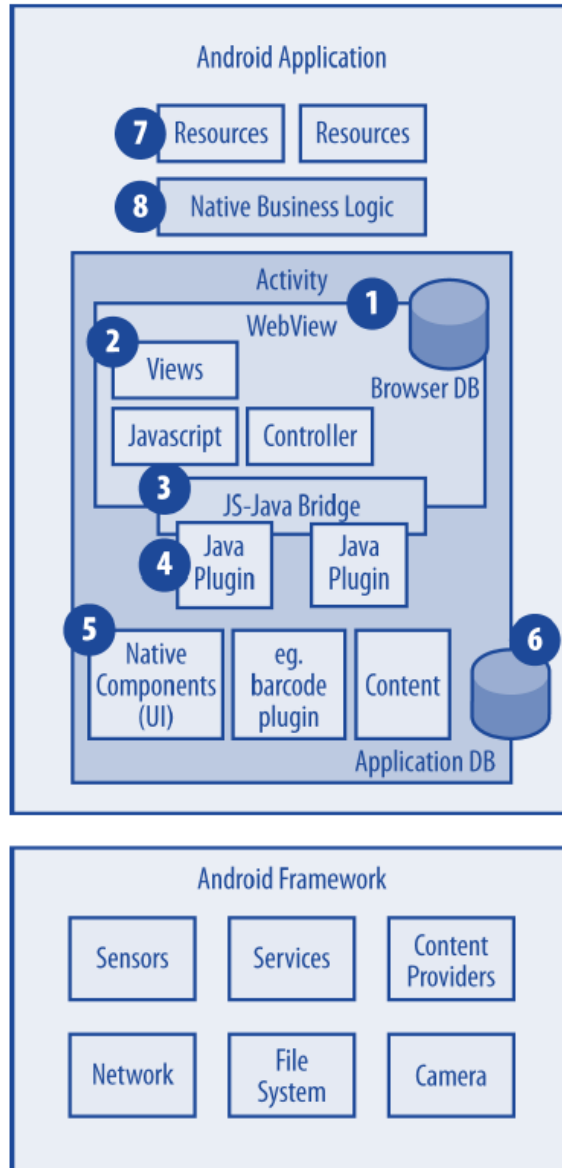
**Java plug-ins (4)** Ένα Java πρόσθετο είναι κάποιο πρόσθετο API το οποίο χρησιμοποιείται στο περιβάλλον της JavaScript.

**Native components (5)** Υπάρχουν συστατικά και υπηρεσίες των native εφαρμογών με δυνατότητα να χρησιμοποιηθούν μέσω διαφόρων πρόσθετων σε μια υβριδική εφαρμογή (όπως είναι η πρόσβαση στην τοποθεσία της συσκευής ή η χρήση του αποθηκευτικού χώρου της συσκευής).

**Application data (6)** Η HTML5 παρέχει διάφορους τρόπους αποθήκευσης των δεδομένων σε μια Web εφαρμογή (local storage, WebSQL), ωστόσο ανάλογα με τις ανάγκες της κάθε εφαρμογής μπορεί να είναι απαραίτητη η αποθήκευση δεδομένων και σε άλλες μορφές. Με την χρήση των ανάλογων πρόσθετων (όπως το Filesystem API) δίνεται η δυνατότητα αποθήκευσης δεδομένων της εφαρμογής στον αποθηκευτικό χώρο της συσκευής.

**Assets and resources (7)** Assets και οι πόροι περιλαμβάνουν όλα τα στατικά δεδομένα που συνοδεύουν την εφαρμογή.

**Native business logic (8)** Ένας πολύ σημαντικός αρχιτεκτονικός διαχωρισμός, κατά το σχεδιασμό υβριδικών εφαρμογών είναι ο διαχωρισμός λογικής της εφαρμογής μεταξύ των native και web συστατικών της. Συχνά μπορεί να είναι απαραίτητη η εκτέλεση μέρους της λογικής της εφαρμογής στο native στρώμα για αρκετούς λόγους, όπως η πρόσβαση σε λειτουργίες της συσκευής.



Εικόνα 2.4: Αρχιτεκτονική μιας Υβριδικής Εφαρμογής σε πλατφόρμα Android [2]

## 2.3 Διαδικτυακές τεχνολογίες από τη πλευρά του πελάτη (Client Side)

Η δημιουργία μιας υβριδικής εφαρμογής, όπως παρουσιάστηκε στην Παράγραφο 2.2, γίνεται με χρήση των βασικών γλωσσών ανάπτυξης ιστοσελίδων διαδικτύου, δηλαδή HTML5, JavaScript και CSS. Τις περισσότερες φορές χρησιμοποιούνται και διάφορες βιβλιοθήκες JavaScript για την προσθήκη επιπλέον λειτουργιών, την απλοποίηση της σύνταξης και την υποστήριξη οθονών αφής. Στη συνέχεια του κεφαλαίου παρουσιάζονται οι γλώσσες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής καθώς και οι διάφορες βιβλιοθήκες της JavaScript.

### 2.3.1 HTML5

Η HTML προέρχεται από το ακρωνύμιο των λέξεων Hyper Text Markup Language και είναι η βασική γλώσσα δημιουργίας σελίδων του διαδικτύου. Η γλώσσα χρησιμοποιεί έναν αριθμό από tags για την μορφοποίηση του κειμένου, την δημιουργία συνδέσμων μετάβασης μεταξύ των σελίδων, την εισαγωγή εικόνων, ήχου και άλλα. Όταν ένας Browser ανοίγει ένα αρχείο HTML τα στοιχεία (tags) μεταφράζονται σε κατάλληλα χαρακτηριστικά με αποτελέσματα στην εμφάνιση και στην λειτουργικότητα της συγκεκριμένης σελίδας.

Η σήμανση της γλώσσας έχει περάσει μέσα από αρκετές αναθεωρήσεις και από τον Οκτώβριο του 2014 η HTML5 αποτελεί την τελική και ολοκληρωμένη έκδοση του πρότυπου HTML του παγκόσμιου ιστού. Η HTML5 αντικαθιστά την προκάτοχο της HTML 4 αλλά και τις XHTML 1.0, και DOM Level 2 HTML. Η νέα γλώσσα εισάγει αρκετά νέα στοιχεία τα σημαντικότερα από αυτά είναι τα παρακάτω [12], [13]:

- Νέα σημασιολογικά στοιχεία όπως <header>, <footer>, <article>, και <section>
- Ελέγχους εισαγόμενων τιμών στις φόρμες όπως αριθμούς, ημερομηνία, ώρα, ημερολόγιο και εύρος
- Νέα γραφικά στοιχεία όπως <svg> (Scallable Vector Graphics) και <canvas>
- Υποστήριξη πολυμέσων όπως ήχο και βίντεο

Πέρα από τα νέα γλωσσικά στοιχεία η HTML5 προσθέτει μία σειρά από νέα APIs τα περισσότερα από τα οποία βελτιώνουν σημαντικά την εμπειρία χρήσης κατά την λειτουργία μια HTML5 σελίδας σε κινητές συσκευές [12], [13]. Μερικά από τα οποία είναι:

- HTML5 Geolocation
- HTML5 Local Storage
- HTML5 Application Cache
- HTML5 Web Workers
- HTML5 SSE

Αξίζει να δοθεί ιδιαίτερη έμφαση στο νέο χαρακτηριστικό HTML5 Local Storage API. Στις παλαιότερες εκδόσεις της γλώσσας η αποθήκευση δεδομένων μίας εφαρμογής απαιτούσε την επικοινωνία της με κάποια βάση δεδομένων σε έναν απομακρυσμένο εξυπηρετητή. Η HTML5

επιτρέπει την αποθήκευση δεδομένων τοπικά στη συσκευή του χρήστη. Τα πλεονεκτήματα με τη χρήση αυτής της τεχνικής είναι πολλά καθώς οι εφαρμογές μπορούν να εκτελεστούν ακόμη και εκτός δικτύου, με χρήση των δεδομένων που αποθηκεύονται τοπικά και συγχρονίζονται όταν υπάρχει σύνδεση στο δίκτυο.

### 2.3.2 JavaScript

Η JavaScript είναι μία αντικειμενοστραφής γλώσσα προγραμματισμού, η οποία ενσωματώνεται στον HTML κώδικα, για την δημιουργία διαδραστικών ιστοσελίδων. Αναπτύχθηκε από την Netscape και η αρχική της έκδοση βασίστηκε στη σύνταξη της γλώσσας προγραμματισμού C.

Ο κώδικας Javascript μιας σελίδας περικλείεται από tag της HTML `<script type="text/javascript">` και `</script>`. Η γλώσσα μπορεί να αλληλοεπιδράσει με τον HTML κώδικα, επιτρέποντας στους προγραμματιστές να εμπλουτίσουν τις ιστοσελίδες τους με δυναμικό περιεχόμενο. Η διαφορά της από την PHP οφείλεται στο ότι ο κώδικας εκτελείται στην πλευρά του πελάτη (client-side) και όχι του εξυπηρετητή (server-side).

Η JavaScript χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων, όπως για παράδειγμα έγγραφα PDF, εξειδικευμένους φυλλομετρητές (site-specific browsers) και οι μικρές εφαρμογές της επιφάνειας εργασίας (desktop widgets).

Με τη διάδοση του Node.js, η JavaScript χρησιμοποιείται και στην πλευρά του εξυπηρετητή. Το Node.js είναι ένα εργαλείο γραμμής εντολών που τρέχει JavaScript σε ένα σύστημα χωρίς να είναι απαραίτητο κάποιο πρόγραμμα περιήγησης. Αυτό επιτυγχάνεται με τη χρήση μιας έκδοσης του Chrome V8 Engine, μια μηχανή της JavaScript που τρέχει στο εσωτερικό του Google Chrome. Πριν το Node.js, οι προγραμματιστές έπρεπε να χρησιμοποιούν διαφορετικές γλώσσες στην πλευρά του πελάτη και του διακοσμητή. Για παράδειγμα, PHP, Java, ASP.Net εκτελούνται στον εξυπηρετητή, και JavaScript στον Browser του πελάτη. Με το Node.js, οι προγραμματιστές μπορούν να χρησιμοποιήσουν την JavaScript τόσο στην πλευρά του εξυπηρετητή, όσο και του πελάτη [14].

### 2.3.3 CSS

Η γλώσσα CSS από το ακρωνύμιο Cascading Style Sheets ανήκει στην κατηγορία των γλωσσών Cascading Style και χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφεί με μια γλώσσα σήμανσης (HTML). Ασχολείται, δηλαδή, αποκλειστικά με την εμφάνιση της ιστοσελίδας. Η CSS αναπτύχθηκε κατά κύριο λόγο για τον διαχωρισμό του περιεχόμενου ενός εγγράφου από την απεικόνιση του, συμπεριλαμβάνοντας στοιχεία όπως η διαρρύθμιση της σελίδας, τα χρώματα και οι γραμματοσειρές [15].

Για κάθε HTML στοιχείο παρέχεται μία λίστα από κανόνες μορφοποίησης. Για παράδειγμα, ένας κανόνας CSS μπορεί να καθορίζει ότι «όλες οι επικεφαλίδες <h1> θα είναι με έντονη γραφή» χωρίς να είναι απαραίτητος ο εγκλεισμός κάθε κεφαλίδας σε ένα <bold> tag. Η χρήση CSS επιτρέπει την απεικόνιση των ίδιων δεδομένων σε πολλές διαφορετικές μορφές, ανάλογα με το μέγεθος της οθόνης, όπως στην οθόνη ενός υπολογιστή ή ενός smartphone. Ένα ακόμα πλεονέκτημα χρήσης της CSS είναι η ευκολία τροποποίησης της εμφάνισης μιας σειράς εγγράφων με την ίδια ευκολία τροποποίησης του ενός.

#### 2.3.4 3<sup>rd</sup> Party Services

Στη συνέχεια του κεφαλαίου, παρουσιάζονται οι βιβλιοθήκες JavaScript που χρησιμοποιήθηκαν για να εμπλουτίσουν και να οργανώσουν τον κώδικα καθώς και να δώσουν επιπλέον λειτουργίες στην εφαρμογή. Οι βιβλιοθήκες αυτές είναι:

- JavaScript Framework - *jQuery*
- HTML5 Mobile UI Framework - *jQuery mobile*
- MV\* Framework - *Backbone.js*
- JavaScript Template Engine - *Underscore.js*
- File and module loader - *Require.js*
- Δημιουργία διαδικτυακών χαρτών - *Leaflet.js*

#### *jQuery και jQuery mobile*

Η jQuery είναι μια βιβλιοθήκη JavaScript, που απλοποιεί αρκετά τον προγραμματισμό σε JavaScript [16]. Έχει τα ίδια αποτελέσματα με την JavaScript αλλά με λιγότερο και πιο εύκολο στη σύνταξη και την ανάγνωση κώδικα. Η βιβλιοθήκη jQuery είναι δωρεάν, ανοιχτού κώδικα υπό διπλή άδεια χρήσης MIT License και GNU General Public License, Version 2. Κυκλοφόρησε το 2006 και χρησιμοποιείται στο 69,6% των 10.000 δημοφιλέστερων ιστοσελίδων [17].

Το jQuery mobile είναι μία JavaScript βιβλιοθήκη, που αναπτύσσεται από την ομάδα του jQuery και προσφέρει σε ιστοσελίδες HTML5 που προορίζονται για συσκευές αφής όπως κινητά και tablets μια φιλική διεπαφή χρήστη. Είναι δομημένο στην βάση του jQuery και του jQuery UI και παρέχει Ajax Navigation (asynchronous JavaScript and XML) με εναλλαγές σελίδων, υποστήριξη σε οθόνες αφής και μία πληθώρα από πρόσθετα που προσομοιώνουν τις λειτουργίες μιας native εφαρμογής. Είναι συμβατό με τις κυρίαρχες πλατφόρμες κινητών συσκευών όπως iOS, Android, Blackberry, WebOS, Symbian, Windows Phone καθώς και με όλους τους βασικούς browser υπολογιστών. Παρέχει το σχεδιασμό θεμάτων μέσα από το site, ενώ ο ίδιος κώδικας προσαρμόζεται αυτόματα ανάλογα με το μέγεθος της οθόνης και ανεξαρτήτως πλατφόρμας [18], [19], [20].

```

<div data-role="page" id="pageone">
  <div data-role="header">
    <h1> Page Title</h1>
  </div>
  <div data-role="main" class="ui-content">
    <p>Content </p>
  </div>
  <div data-role="footer">
    <h1> Footer</h1>
  </div>
</div>

```

Κώδικας 2.4: Τυπική διάταξη jQuery mobile σελίδας



Εικόνα 2.5: Τυπική διάταξη jQuery mobile σελίδας

Η εκμάθηση του jQuery Mobile είναι ιδιαίτερα εύκολη λόγω της ομοιότητας του με την βιβλιοθήκη jQuery, η βιβλιογραφία είναι πολύ πλούσια και διαθέτει και μια αρκετά μεγάλη κοινότητα χρηστών. Εναλλακτικές προτάσεις αποτελούν συστήματα όπως το Sencha Touch, jq Touch, Kendo UI, jq.Mobi, Intel ARK και πολλά ακόμη.

### [Backbone.js](#)

Το jQuery mobile παρέχει μια διεπαφή χρήστη για όλες τις δημοφιλείς πλατφόρμες κινητών συσκευών, αλλά δεν παρέχει μια δομή για την οργάνωση και την αρχιτεκτονική της JavaScript εφαρμογής. Αυτό δεν αποτελεί πρόβλημα για μικρές εφαρμογές μιας σελίδας (Single Page Apps – SPAs), ωστόσο όσο ο κώδικας αυξάνεται, και οι σελίδες γίνονται όλο και περισσότερες, δημιουργούνται προβλήματα στην εκτέλεση του κώδικα από μία κινητή συσκευή περιορισμένης απόδοσης [2]. Για το λόγο αυτό οι σχεδιαστές διαδικτυακών εφαρμογών για κινητές συσκευές στράφηκαν σε MVC πλαίσια και πλαίσια διαχείρισης αρχείων.

### [MVC πλαίσια](#)

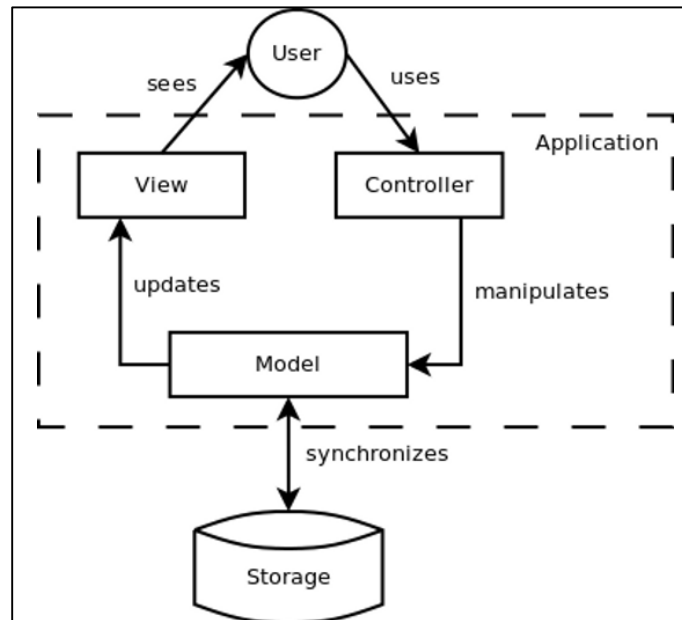
Μια πληθώρα από σύγχρονα πλαίσια JavaScript παρέχουν στους προγραμματιστές έναν εύκολο τρόπο να οργανώσουν τον κώδικα τους χρησιμοποιώντας διάφορα εναλλακτικά πρότυπα γνωστά ως MVC (Model-View-Controller) [2]. Ένα πλαίσιο MVC διαχωρίζει την υπόθεση μιας εφαρμογής σε τρία διακριτά μέρη:

**Models** Τα Models σε ένα MVC πλαίσιο αναπαριστούν την γνώση και τα δεδομένα ενός πεδίου. Συνήθως ένα Model αλληλοεπιδρά με ένα συγκεκριμένο πίνακα μίας βάσης δεδομένων.

**Views** Τις Views όπου αποτελούν την διεπαφή χρήστη (UI) σε μία εφαρμογή (όπως το markup και τα templates). Χρησιμοποιούνται συνήθως, αλλά όχι απαραίτητα, για να παρουσιάσουν τα δεδομένα της εφαρμογής.

**Controllers** Οι Controllers διαχειρίζονται τις εισόδους (όπως τα clicks, ενέργειες του χρήστη) και ανανεώνουν τα αντίστοιχα Models.

Συνοψίζοντας, σε μία MVC εφαρμογή, οι εντολές του χρήστη εκτελούνται από τον Controller ο οποίος στέλνει εντολές προς το Model και ενημερώνει την κατάσταση του, ενώ επικοινωνεί με την View για την αλλαγή του τρόπου παρουσίασης του Model. Ένα Model ειδοποιεί τις σχετικές Views και τον Controller, όταν υπάρχει μια αλλαγή στην κατάσταση του. Η ενημέρωση αυτή επιτρέπει στις Views να ανανεώσουν την διεπαφή χρήστη και οι Controller να αλλάξουν το διαθέσιμο σύνολο εντολών. Μια View ζητά πληροφορίες από το Model που το χρησιμοποιεί για να δημιουργήσει μια αντιπροσωπευτική του έξοδο στο χρήστη.



Διάγραμμα 2.1: Τυπική μορφή ενός MVC πλαισίου [2]

Παρόλο που η τεχνολογία MVC αναπτύχθηκε κυρίως για υπολογιστές, αφομοιώθηκε ευρέως ως μια αρχιτεκτονική για διαδικτυακές εφαρμογές σε σημαντικές γλώσσες προγραμματισμού. Αρκετά πλαίσια διαδικτυακών εφαρμογών έχουν αναπτυχθεί για να ενισχύσουν αυτό το πρότυπο με δημοφιλέστερα το Angular.js από την Google, το Backbone.js, το Ember.js και το Knockout.js.

Τα JavaScript MVC frameworks δεν ακολουθούν πάντα αυστηρά το παραπάνω πρότυπο. Κάποια από αυτά (συμπεριλαμβανομένου του Backbone.js) ενσωματώνουν τη λειτουργία του Controller μέσα στη View, ενώ άλλες προσεγγίσεις προσθέτουν επιπλέον συστατικά. Τέτοια πλαίσια καλούνται MV\* καθώς υπάρχουν Models και Views, αλλά Controller μπορεί να μην υπάρχει ή μπορεί να υπάρχουν και επιπλέον συστατικά.

## [Το MV\\* Πλαίσιο Backbone.js](#)

Ο πιο δημοφιλής συνδυασμός, που προτείνεται και μέσα από το site του jQuery mobile είναι το Backbone.js ως MV\* πλαίσιο και το Require.js για την διαχείριση των εξαρτήσεων, όπου και αναλύεται στη συνέχεια του κεφαλαίου [19].

Το Backbone.js είναι ένα ανοιχτού κώδικα υπό την άδεια MIT License, client-side MV\* πλαίσιο που παρέχει δομή σε JavaScript εφαρμογές [21]. Βοηθά στην οργάνωση του κώδικα σε διαχειρίσιμα τμήματα, δηλαδή Models, Collections και Views. Για την δρομολόγηση χρησιμοποιεί ένα hashtag (#) στην διεύθυνση URL. Χρησιμοποιεί εκτενώς το πρότυπο JSON για τα Models και τις Collections χωρίς να εμπλέκει τα στοιχεία στο DOM.

Τα βασικά συστατικά του Backbone όπως φαίνονται στο Διάγραμμα 2.2 είναι τα παρακάτω [21], [22], [23], [24]:

**Models** Τα Models αποτελούν το βασικό αντικείμενο κάθε JavaScript εφαρμογής, περιλαμβάνοντας όλα τα διαδραστικά δεδομένα, όπως επίσης και ένα μεγάλο μέρος της λογικής που πλασιώνει αυτά.

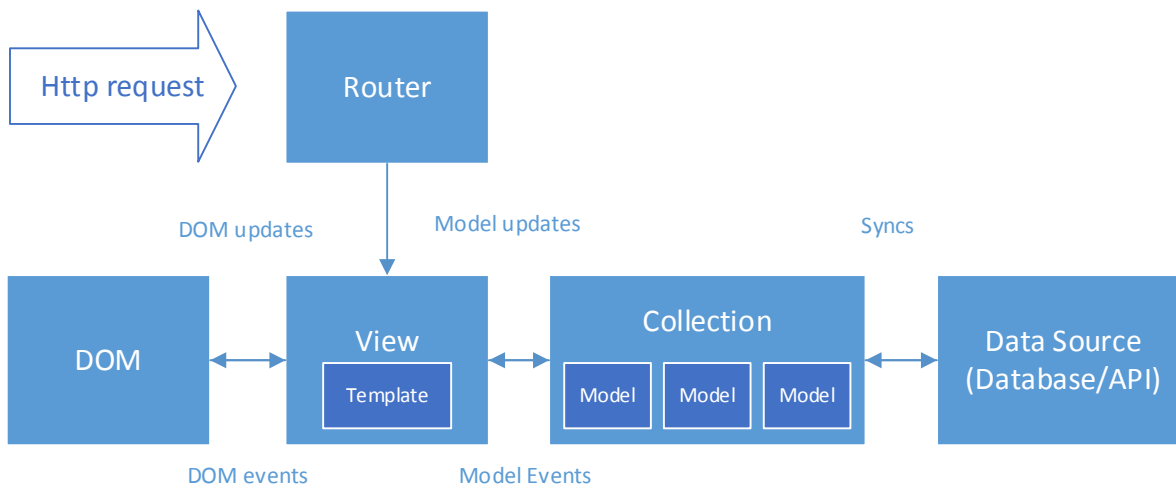
**Collections** Αποτελούν συλλογές αντικειμένων. Οι Collections δίνουν και την δυνατότητα σύνδεσης τους με URLs και την λήψη των δεδομένων τους από τον εξυπηρετητή.

**Views** Οι Views αποτελούν το επίπεδο του γραφικού περιβάλλοντος του Backbone. Αντανακλούν την μορφή των δεδομένων που διατηρούν τα Models. Χρησιμοποιούνται επίσης για την παρακολούθηση των events και αντιδρούν ανάλογα.

**Router** Το Router διαχειρίζεται την κατάσταση της εφαρμογής. Σε μια web εφαρμογή, διαχειρίζεται τις εναλλαγές στην μπάρα διευθύνσεων, ενώ ακόμα και σε μία υβριδική εφαρμογή που δεν είναι εμφανή η μπάρα συνεχίζει να υπάρχει.

**Templates** Τα Templates αποτελούν τμήματα κώδικα HTML τα οποία χρησιμοποιούνται για να αναπαραστήσουν την πληροφορία των Models. Η χρήση των templates γίνεται με χρήση κάποιας JavaScript βιβλιοθήκης templating και στην περίπτωση του Backbonejs την βασική εξάρτηση του Underscore.js, χωρίς όμως να απαγορεύεται η χρήση και κάποιας άλλης βιβλιοθήκης templating.





Διάγραμμα 2.2: Λειτουργία του MV\* πλαισίου Backbonejs [21]

Το Backbone.js είναι γνωστό ως ιδιαίτερα ελαφρύ (μόλις 6kb), ενώ η μόνο εξάρτηση του είναι σε μια βιβλιοθήκη JavaScript, το Underscore.js. Είναι σχεδιασμένο για την ανάπτυξη single page διαδικτυακών εφαρμογών, και για το συγχρονισμό διαφόρων τμημάτων web εφαρμογών με τον εξυπηρετητή.

Ορισμένες εφαρμογές που έχουν αναπτυχθεί με το Backbone.js είναι οι εξής: Foursquare, USA Today, Wordpress.com, Pinterest, Walmart Mobile και Sony Entertainment Network [21].

### Underscore.js

Η δημιουργία μεγάλων τμημάτων HTML κώδικα και ιδιαίτερα σε εφαρμογές μίας σελίδας (SPAs) απαιτεί αρκετούς υπολογιστικούς πόρους, τις περισσότερες φορές κάνει επανάληψη των ίδιων τμημάτων κώδικα και συχνά οδηγεί στο λεγόμενο spaghetti code που είναι δύσκολο να διαβαστεί και να συντηρηθεί [22].

Η χρήση JavaScript πλαισίων που υποστηρίζουν κάποια μορφή MVC/MV\*, συνδυάζεται με κάποια JavaScript templating βιβλιοθήκη για την διαχείριση των όψεων των Views. Βιβλιοθήκες JavaScript templating (όπως Mustache, Handlebars.js, Underscore.js) χρησιμοποιούνται για να καθορίσουν πρότυπα (templates) για τις Views ως HTML κώδικα που περιέχει μεταβλητές templating. Οι μεταβλητές αυτές οροθετούνται από μια ιδιαίτερη σύνταξη, για παράδειγμα `<%= title%>` για την βιβλιοθήκη Underscore.js [2], [25]. Μια δημοφιλής επιλογή βιβλιοθήκης JavaScript templating είναι και το Underscore.js, ένα πλαίσιο ανοικτού κώδικα υπό την άδεια MIT License, που αποτελεί τη βασική εξάρτηση του Backbonejs.

```
<li>
  <a href="#page/<%= id %>" id="<%= id %>">
    <h4><%= name %></h4>
    <p><%= county %></p>
    <p class="ui-li-aside"><strong><% type %></strong></p>
  </a>
</li>
```

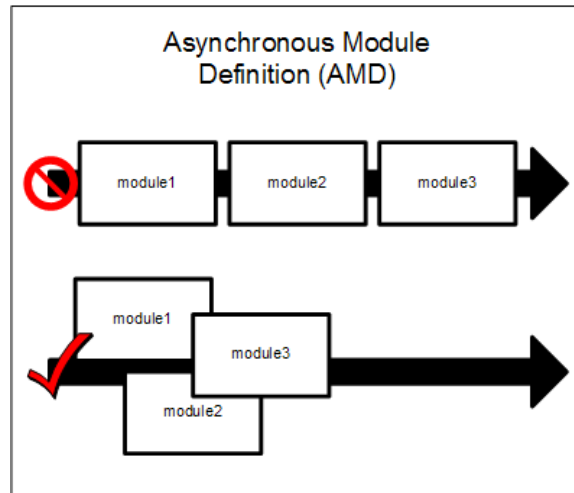
Κώδικας 2.5: Παράδειγμα χρήσης Underscore.js Template Engine

Οι βιβλιοθήκες JavaScript templating συνήθως δέχονται τα δεδομένα σε διάφορες μορφές, όπως JSON. Η συμπλήρωση των templates με τα δεδομένα πραγματοποιείται από το ίδιο το πλαίσιο. Αυτή η τεχνική παρουσιάζει πολλά πλεονεκτήματα, ιδιαίτερα όταν επιλεγεί να αποθηκευτούν οι templates εξωτερικά και φορτώσουν δυναμικά όταν είναι απαραίτητες.

### Require.js

Κάθε φορά που ο Browser φορτώνει ένα αρχείο, στέλνει μια HTTP αίτηση. Οι Browsers περιορίζονται στο πόσες παράλληλες αιτήσεις μπορούν να στείλουν, έτσι συχνά είναι αργή η φόρτωση πολλών αρχείων ταυτόχρονα. Ο αριθμός αυτός εξαρτάται από τις ρυθμίσεις του χρήστη και τον Browser. Καθώς στην ανάπτυξη Backbone εφαρμογών συνίσταται ο διαχωρισμός της εφαρμογής σε πολλαπλά αρχεία JavaScript, αυτό το όριο είναι εύκολο να χτυπήσει γρήγορα [22]. Επιπλέον τα αρχεία φορτώνουν συγχρονισμένα που σημαίνει ότι ο Browser δεν μπορεί να συνεχίσει μέχρι να ολοκληρωθεί η φόρτωση του συνόλου των αρχείων.

Το Require.js είναι ένα πλαίσιο ανοικτού κώδικα υπό την άδεια MIT License για την ασύγχρονη φόρτωση JavaScript αρχείων (AMD – Asynchronous Module Definition) [26]. Η χρήση της τεχνικής AMD φορτώνει τα αρχεία JavaScript μόνο όταν αυτά είναι απαραίτητα επιταχύνοντας έτσι την φόρτωση και την απόδοση των σελίδων. Επιπλέον, επιτρέπεται ο διαχωρισμός του κώδικα σε ενότητες και ο ορισμός εξαρτήσεων που πρέπει να φορτωθούν πριν την εκτέλεση μιας ενότητας. Στο Διάγραμμα 2.3 φαίνεται η διαδικασία φόρτωσης των ενοτήτων (modules) με και χωρίς την χρήση κάποιας βιβλιοθήκης ασύγχρονης φόρτωσης αρχείων.



Διάγραμμα 2.3: Ασύγχρονη φόρτωση με χρήση του πλαισίου *Require.js* [27]

### *Leaflet.js*

Η *Leaflet.js* είναι μια ευρέως γνωστή JavaScript βιβλιοθήκη ανοικτού κώδικα υπό την άδεια BSD License που χρησιμοποιείται για τη δημιουργία διαδικτυακών χαρτών [28]. Η βιβλιοθήκη εστιάζει ιδιαίτερα στην δημιουργία χαρτών που προορίζονται για κινητές συσκευές. Κυκλοφόρησε για πρώτη φορά το 2011 και υποστηρίζεται από τις περισσότερες πλατφόρμες smartphone καθώς και από όλους τους σύγχρονους Browsers. Μαζί με το Open Layers, και το Google Maps API, είναι μία από τις πιο δημοφιλείς βιβλιοθήκες JavaScript του είδους της και χρησιμοποιείται από μεγάλες ιστοσελίδες, όπως το Foursquare, Pinterest και το Flickr.

Η *Leaflet.js* επιτρέπει στους προγραμματιστές, οι οποίοι δεν έχουν GIS υπόβαθρο να εμφανίσουν πολύ εύκολα διαδικτυακούς χάρτες που φιλοξενούνται σε ένα εξυπηρετητή, με αρκετά προαιρετικά επίπεδα σε αυτούς (Κώδικας 2.6, Εικόνα 2.6). Οι προγραμματιστές μπορούν να φορτώσουν τα δεδομένα από αρχεία μορφής GeoJSON, να τα μορφοποιήσουν και να δημιουργήσουν διαδραστικά στρώματα σε αυτούς, όπως markers με αναδυόμενα παράθυρα και διανυσματικά επίπεδα (σημεία, πολυγωνικές και πολύγωνα).

Ο προγραμματιστής έχει την δυνατότητα να επιλέξει για χάρτη υπόβαθρο οποιοδήποτε Web Map Service (WMS), με δημοφιλέστερο αυτό του Open Street Maps [29], ενώ οι δυνατότητες μπορούν να επεκταθούν μέσα από ένα μεγάλο εύρος πρόσθετων όπως για παράδειγμα δημιουργία Marker Clusters, ετικετών στους markers, προσθήκη Shapefiles και συνδυασμό με τα εργαλεία του ArcGIS Online.

```

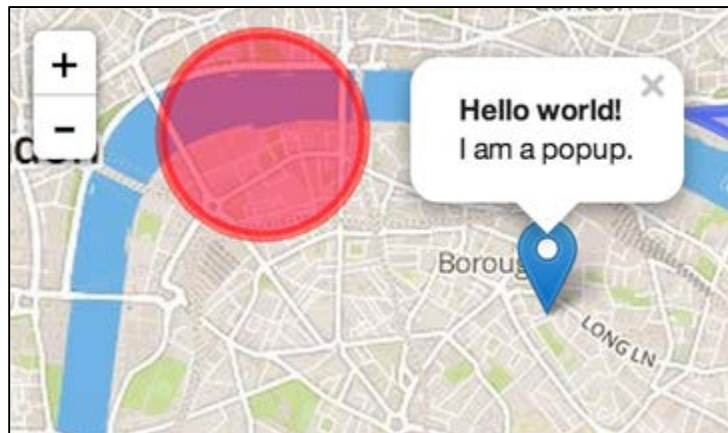
// Create a map in the "map" div, set the view to a given place and zoom
var map = L.map('map').setView( [35.505, 24.012], 13);

// Add an OpenStreetMap tile Layer
L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png',
{attribution: '&copy;
<a href="http://osm.org/copyright">OpenStreetMap</a>'}
)
.addTo(map);

// Add a marker in the given location, attach some popup content to it and
open the popup
L.marker( [35.5, 24,9]).addTo(map)
  .bindPopup('<b>Hello World!</b><br>I am a popup.')
  .openPopup();

```

Κώδικας 2.6: Δημιουργία χάρτη με χρήση της βιβλιοθήκης Leaflet.js



Εικόνα 2.6: Δημιουργία χάρτη με χρήση της βιβλιοθήκης Leaflet.js

Συγκρινόμενη με τα κλειστού κώδικα Google Maps API και Bing Maps API, που επικοινωνούν σε σημαντικό βαθμό με τον εξυπηρετητή για την παροχή υπηρεσιών, η Leaflet.js αποτελεί μια ανοιχτού κώδικα αποκλειστικά client-side βιβλιοθήκη. Δίνει την δυνατότητα να χρησιμοποιηθούν και χάρτες πέρα από τους προεπιλεγμένους των παραπάνω υπηρεσιών, ενώ είναι ευέλικτο και για χρήση χαρτών εκτός σύνδεσης.

Σε σχέση με την open-source βιβλιοθήκη OpenLayers στο σύνολό της είναι πολύ μικρότερη, περίπου 6500 γραμμές κώδικα σε σύγκριση με 21.000 το 2014 (περίπου 33 kb σε σχέση με 950kb). Η βάση του κώδικα είναι νεότερη, και εκμεταλλεύεται πρόσφατες δυνατότητες της

JavaScript. Η βασική βιβλιοθήκη παρέχει τις κύριες λειτουργίες που απαιτούνται σε έναν διαδικτυακό χάρτη αφήνοντας τον προγραμματιστή να επιλέξει μέσα από μεγάλο αριθμό πρόσθετων για την προσθήκη επιπλέον χαρακτηριστικών.

## 2.4 Διαδικτυακές τεχνολογίες από τη πλευρά του εξυπηρετητή (Server Side)

Ως εξυπηρετητής (server) ορίζεται ένα σύστημα που αναλαμβάνει την παροχή διάφορων υπηρεσιών, «εξυπηρετώντας» αιτήσεις άλλων προγραμμάτων, γνωστών ως πελάτες (clients), που μπορούν να τρέχουν στον ίδιο υπολογιστή ή σε σύνδεση με αυτόν μέσω τοπικού δικτύου ή μέσω διαδικτύου [29]. Ο μεγάλος όγκος δεδομένων και οι πολύπλοκες επεξεργασίες που απαιτούνται για την λειτουργία της εφαρμογής που αναπτύχθηκε, καθιστούν απαραίτητη την υποστήριξη της από κάποιο τέτοιου είδους απομακρυσμένο εξυπηρετητή.

### 2.4.1 Λειτουργικό σύστημα του εξυπηρετητή

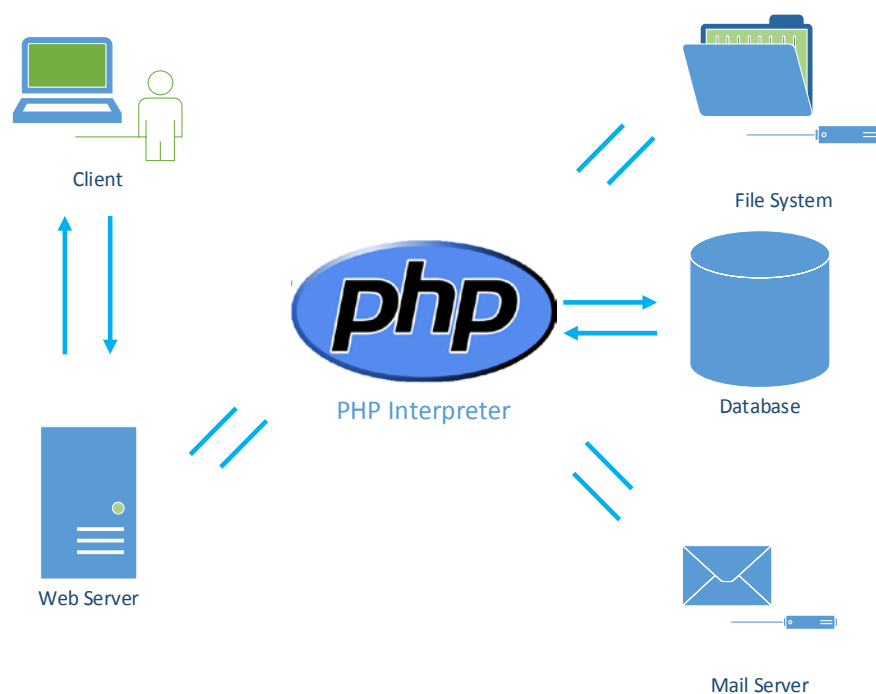
Ο Nginx είναι ένας δωρεάν, ανοιχτού κώδικα, υψηλής απόδοσης HTTP εξυπηρετητής μεσολάβησης και αντίστροφης μεσολάβησης, καθώς και ένας διακομιστής μεσολάβησης IMAP/POP3 [30]. Σήμερα φιλοξενεί περίπου το 12,18% (22,2 εκατομμύρια) των ενεργών sites. Είναι γνωστός για την υψηλή απόδοσή του, τη σταθερότητα, τη πλούσια ποικιλία λειτουργιών, την απλή διαμόρφωση και τη χαμηλή κατανάλωση πόρων.

Είναι ένας από τους ελάχιστους των διακομιστών που έχουν σχεδιαστεί για την αντιμετώπιση του προβλήματος C10K (πρόβλημα στην διαχείριση μεγάλου αριθμού χρηστών ταυτόχρονα). Ακόμα κι αν δεν είναι απαραίτητη η διαχείριση χιλιάδων ταυτόχρονων αιτήσεων, υπάρχουν σημαντικά πλεονεκτήματα από την χρήση του σε θέματα απόδοσης. Ο Nginx χρησιμοποιείται σε αρκετές ιστοσελίδες υψηλής προβολής, όπως το Netflix, Hulu, Pinterest, Cloudflare, Airbnb, WordPress.com, GitHub, SoundCloud, η Zynga, Eventbrite, Zappos [30].

### 2.4.2 Γλώσσα ανάπτυξης εξυπηρετητή (PHP)

Η PHP είναι μια γλώσσα προγραμματισμού που σχεδιάστηκε για τη δημιουργία δυναμικών σελίδων στο διαδίκτυο και είναι επισήμως γνωστή ως: Hyper Text preprocessor. Η σύνταξη της βασίζεται στις γλώσσες προγραμματισμού C, Java και Perl. Είναι μια server-side scripting γλώσσα που γράφεται συνήθως πλαισιωμένη από HTML, για την μορφοποίηση των αποτελεσμάτων. Αντίθετα από μια συνηθισμένη HTML σελίδα, μία σελίδα PHP δεν στέλνεται άμεσα σε έναν πελάτη (client), αλλά πρώτα αναλύεται και έπειτα αποστέλλεται το παραγόμενο αποτέλεσμα. Τα στοιχεία HTML στον πηγαίο κώδικα μένουν ως έχουν, αλλά ο PHP κώδικας ερμηνεύεται και εκτελείται. Γενικά, ο κώδικας που εκτελείται από την πλευρά του πελάτη, μπορεί να τροποποιήσει δυναμικά το περιεχόμενο μιας σελίδας, να απαντήσει σε ερωτήματα χρηστών μέσα από φόρμες HTML και να δώσει πρόσβαση σε δεδομένα και βάσεις δεδομένων [31], [32].

Στην Εικόνα 2.4 παρουσιάζεται μια τυπική σειρά ενεργειών που συμβαίνουν κατά την κλήση ενός κώδικα PHP. Ο χρήστης εισάγει τη διεύθυνση μιας ιστοσελίδας στο Browser αιτούμενος ένα αρχείο PHP. Ο εξυπηρετητής αναγνωρίζοντας ότι το επιλεγμένο αρχείο είναι PHP λόγω της κατάληξης .php τρέχει μέσω του PHP Interpreter τον κώδικα. Ο κώδικας εκτελείται και είτε ο χρήστης παραπέμπεται σε μία άλλη PHP σελίδα για να εκτελεστεί ένας νέος κώδικας, είτε εμφανίζεται η HTML σελίδα που κάλεσε [31].



Διάγραμμα 2.4: Σειρά ενεργειών κατά την κλήση ενός κώδικα PHP [31]

### 2.4.3 Το σύστημα διαχείρισης βάσεων δεδομένων (PostgreSQL)

#### Γενικά στοιχεία

Η PostgreSQL αποτελεί ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων με 20 χρόνια παρουσίας που προσδίδει ιδιαίτερη έμφαση στην επεκτασιμότητα και τα πρότυπα συμμόρφωσης. Η PostgreSQL αναπτύσσεται από την PostgreSQL Global Development Group, μια διευρυμένη ομάδα από πολλές εταιρείες και μεμονωμένους συνεισφέροντες. Είναι δωρεάν και ανοικτού κώδικα λογισμικό, που κυκλοφόρησε υπό τους όρους της Άδειας PostgreSQL (PostgreSQL License), μια ανεκτική άδεια χρήσης ελεύθερου λογισμικού [33].

Η PostgreSQL μπορεί να εγκατασταθεί σε όλα τα βασικά λειτουργικά συστήματα, συμπεριλαμβανομένων Linux, UNIX (AIX, BSD, HP-UX, SGIIRIX, MacOSX, Solaris, Tru64), και Windows. Η διαχείριση γίνεται κυρίως μέσω του PgAdmin III, αλλά και με άλλες εφαρμογές τρίτων όπως PgAccess, PhpPgAdmin, WinSQL. Είναι πλήρως ACID (Atomicity, Consistency, Isolation, Durability) συμβατή, έχει πλήρη υποστήριξη foreign keys, joins, views, triggers και αποθηκευμένων διαδικασιών (σε διάφορες γλώσσες). Περιλαμβάνει το μεγαλύτερο μέρος των τύπων δεδομένων της SQL: 2008, όπως INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL και TIMESTAMP. Επίσης, υποστηρίζει αποθήκευση δυαδικών αντικειμένων, όπως εικόνες, ήχους ή βίντεο. Διαθέτει εγγενείς διεπαφές προγραμματισμού για C / C ++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC. Μερικές γενικές οριακές τιμές μιας PostgreSQL συμπεριλαμβάνονται στον παρακάτω πίνακα:

Όριο	Τιμή
Μέγιστο μέγεθος της βάσης	Απεριόριστο
Μέγιστο μέγεθος πίνακα	32 TB
Μέγιστο μέγεθος πλειάδας	1.6 TB
Μέγιστο μέγεθος πεδίου	1 GB
Μέγιστος αριθμός πλειάδων	Απεριόριστες
Μέγιστος αριθμός από στήλες ανά πίνακα	250 - 1600 εξαρτάται από τον τύπο στης στήλης
Μέγιστος αριθμός ευρετηρίων ανά πίνακα	Απεριόριστα

Πίνακας 2.1: Τυπικά όρια μιας PostgreSQL βάσης δεδομένων [33]

Η PostgreSQL είναι μια δημοφιλής βάση δεδομένων για διαδικτυακά προγράμματα και ιστοσελίδες. Χρησιμοποιείται σε κάποιες από τις πιο διαδεδομένες διαδικτυακές υπηρεσίες, όπως το IMDB.com και το Skype.

### Πρόσθετο PostGIS

Ένα από τα βασικά πρόσθετα της PostgreSQL είναι το PostGIS, το οποίο παρέχει υποστήριξη σε όλους τους τύπους γεωμετρίας και χωρικών λειτουργιών του OGC, επιτρέποντας της να χρησιμοποιηθεί ως μια χωρική βάση δεδομένων για γεωγραφικά συστήματα πληροφοριών (GIS), όπως και το SDE της ESRI ή η Χωρική επέκταση της Oracle [35].

Η υλοποίηση του PostGIS βασίζεται σε «ελαφρύ» γεωμετρικές και ευρετήρια σχεδιασμένα για μείωση των δεδομένων που απαιτείται να μεταφερθούν από το αποθηκευτικό χώρο στην μνήμη RAM για την πραγματοποίηση ερωτημάτων, βελτιώνοντας έτσι αισθητά την απόδοση των ερωτημάτων.

Το πρόσθετο PostGIS εμπλουτίζει τις λειτουργίες μια τυπικής βάσης δεδομένων PostgreSQL με στοιχεία σχετικά με:

- Αποθήκευση τύπων γεωμετρίας για σημεία, πολυγωνικές, πολύγωνα, πολύ-σημειακές, πολύ-πολυγωνικές, πολύ-πολύγωνα και συλλογές-γεωμετριών.
- Χωρικά κατηγορήματα για τον προσδιορισμό των αλληλεπιδράσεων των γεωμετριών
- Χωρικούς τελεστές για τον προσδιορισμό χωρικών μετρήσεων, όπως το εμβαδόν, η απόσταση, το μήκος και η περίμετρο.
- Χωρικούς τελεστές για τον προσδιορισμό χωρικών πράξεων, όπως η ένωση, η διαφορά, η συμμετρική διαφορά και η δημιουργία ζωνών αποκλεισμού
- Γενικευμένο Δέντρο Αναζήτησης (R-tree-over-GiST)- Χωρικά ευρετήρια για υψηλής ταχύτητας χωρικά ερωτήματα.
- Δείκτες επιλεκτικότητας (Index selectivity), για ταχύτερα μικτά χωρικά και μη-χωρικά ερωτήματα.
- Αποθήκευση δεδομένων ράστερ, PostGIS WKT Raster.



## Χωρικές Λειτουργίες

Το PostGIS στην έκδοση 2.1 παρέχει περισσότερες από 700 συναρτήσεις για την διαχείριση χωρικών δεδομένων. Ενδεικτικά αναφέρονται στη συνέχεια ορισμένες από τις διαθέσιμες λειτουργίες και αντίστοιχες SQL (χωρικές) συναρτήσεις που προσφέρονται [36].

### Λειτουργίες διαχείρισης χωρικών δεδομένων

- AddGeometryColumn Προσθήκη στήλης Γεωμετρίας
- UpdateGeometrySRID Αλλαγή του SRID της στήλης

### Λειτουργίες δημιουργίας γεωμετριών

- ST\_GeomFromGeoJSON Δημιουργία γεωμετρίας από αρχείο GeoJSON
- ST\_GeomFromText Δημιουργία γεωμετρίας μέσω WKT
- ST\_MakePoint Δημιουργία σημείου από συντεταγμένες
- ST\_MakeLine Δημιουργία γραμμής από συντεταγμένες

### Λειτουργίες πρόσβασης στις γεωμετρίες

- ST\_Y Πρόσβαση στη X συντεταγμένη του σημείου
- ST\_X Πρόσβαση στη Y συντεταγμένη του σημείου

### Λειτουργίες επεξεργασίας γεωμετριών

- ST\_LineMerge Συγχώνευση πολυγωνικών οντοτήτων για την δημιουργία μιας πολυ-πολυγωνικής
- ST\_Transform Μετατροπή προβολικού συστήματος των οντοτήτων

### Λειτουργίες εξόδου γεωμετριών (Geometry Outputs)

- ST\_AsGeoJSON Εξαγωγή γεωμετρίας σε μορφή GeoJSON
- ST\_AsLatLonText Εξαγωγή γεωμετρίας ως γεωγραφικές συντεταγμένες
- ST\_AsText Εξαγωγή γεωμετρίας ως κείμενο

### Τελεστές χωρικών σχέσεων και μετρήσεων

- ST\_ClosestPoint Εύρεση πλησιέστερης σημειακής οντότητας σε μία γεωμετρία
- ST\_Distance\_Spheroid Προσδιορισμός της απόστασης σε μέτρα σε ένα σύστημα γεωγραφικών συντεταγμένων
- ST\_Length Υπολογισμός του μήκους μιας γραμμικής οντότητας
- ST\_Within Εντοπισμός οντοτήτων που απέχουν μια συγκεκριμένη απόσταση
- ST\_Contains Εύρεση οντοτήτων που περιλαμβάνονται σε μια πολυγωνική οντότητα

## 2.4.4 Η γλώσσα SQL

Η SQL (Structured Query Language) είναι μία γλώσσα για τη διαχείριση δεδομένων, σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS). Η γλώσσα αναπτύχθηκε από την IBM στις αρχές της δεκαετίας του 1970. Βασίζεται στη σχεσιακή άλγεβρα και αποτελείται από δύο μέρη: τη γλώσσα ορισμού δεδομένων DDL (Data Definition Language) με εντολές για τη δημιουργία πινάκων και όψεων σε βάσεις δεδομένων και ευρετηρίων πάνω σε πεδία πινάκων για ταχύτερη προσπέλαση στα δεδομένα και τη γλώσσα χειρισμού δεδομένων DML (Data Manipulation Language) με εντολές για την εισαγωγή, διαγραφή ή τροποποίηση των τιμών των στοιχείων ενός πίνακα και εντολές καταχώρησης ή ακύρωσης των εργασιών που έγιναν στη βάση [34]. Συγκεκριμένα:

### *DDL-SQL*

- CREATE | DROP | ALTER TABLE (base table)
- CREATE | DROP | ALTER VIEW (virtual table)
- CREATE | DROP | ALTER INDEX (index table)

### *DML-SQL*

- SELECT αναζήτηση
- INSERT εισαγωγή εγγραφής
- DELETE διαγραφή εγγραφής
- UPDATE τροποποίηση εγγραφής

Ένα τυπικό ερώτημα της γλώσσας για την αναζήτηση δεδομένων έχει την παρακάτω μορφή:

```
select [ distinct ] target_list
from tuple_variable_list
[ where qualification ]
[ group by grouping_attributes ]
[ having group_condition ]
[ order by target_list_subset ]
```

Κώδικας 2.7 Τυπικό ερώτημα αναζήτησης δεδομένων μίας βάσης σε γλώσσα SQL

## 2.5 Τεχνολογίες επικοινωνίας

Η παρούσα εφαρμογή στοχεύει στην απεικόνιση δυναμικών δεδομένων όπου η επεξεργασία τους πραγματοποιείται σε κάποιον απομακρυσμένο εξυπηρετητή. Σημαντικό ρόλο λοιπόν, παίζουν οι τεχνολογίες επικοινωνίας και ανταλλαγής δεδομένων πελάτη-εξυπηρετητή. Στο παρών κεφάλαιο δίνεται ο ορισμός της τεχνικής AJAX και της προγραμματιστικής διεπαφής API, ενώ στη συνέχεια αναλύεται το πρότυπο JSON που χρησιμοποιείται για την λήψη των δεδομένων από τον server.

### 2.5.1 AJAX

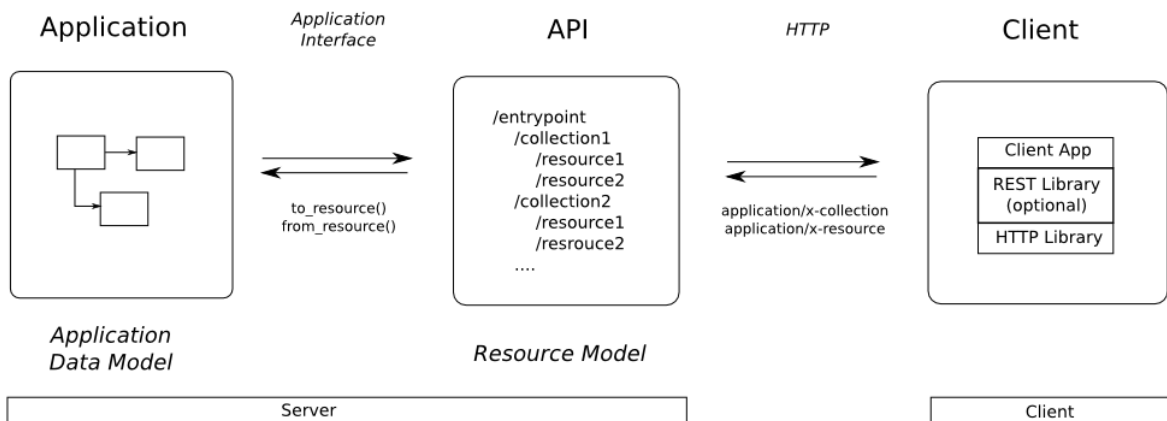
Η τεχνική AJAX (Asynchronous JavaScript and XML) είναι ένα σύνολο από αλληλεξαρτώμενες διαδικτυακές τεχνικές που χρησιμοποιούνται από την πλευρά του πελάτη για την δημιουργία ασύγχρονων διαδικτυακών εφαρμογών. Αυτή η τεχνική επιτρέπει στις διαδικτυακές εφαρμογές την αποστολή και λήψη δεδομένων από ένα server ασύγχρονα (στο υπόβαθρο) χωρίς να επηρεάζεται η εμφάνιση και η συμπεριφορά της υπόλοιπης σελίδας.

Ουσιαστικά η AJAX αποτελεί μια τεχνολογία επικοινωνίας τριών στοιχείων: Του Browser στον οποίο αναπαράγεται το περιεχόμενο και εκτελείται ο κώδικας, ενώ παράλληλα περιέχει και το αντικείμενο XMLHttpRequest, το οποίο επιτρέπει στην JavaScript να επικοινωνήσει με το εξυπηρετητή. Της JavaScript, η οποία στέλνει αίτημα στον εξυπηρετητή, περιμένει την απόκρισή του, την επεξεργάζεται και τη χρησιμοποιεί ανάλογα με το σκοπό για τον οποίο έχει δημιουργηθεί. Τέλος, τον εξυπηρετητή ο οποίος λαμβάνει τα αιτήματα και απαντάει με τις αντίστοιχες πληροφορίες. Παρόλο που γίνεται αναφορά στον τύπο XML είναι δυνατή η χρήση και αρχείων μορφής JSON [37].

### 2.5.2 API

Ως API (Application Programming Interface) ορίζεται μια διεπαφή η οποία λειτουργεί ως γέφυρα μεταξύ δύο χαρακτήρων, ο ένας είναι ο πάροχος των στοιχείων και ο άλλος είναι ο χρήστης αυτών των στοιχείων. Τα web APIs είναι μία τεχνολογία που πρωτοεμφανίστηκε γύρω στο 2000. Τα τελευταία χρόνια αυτός ο τύπος επικοινωνίας χρησιμοποιείται ευρέως στις εφαρμογές κινητών.

Η χρήση κλήσεων API δεν λαμβάνει υπόψη της την αρχιτεκτονική του εξυπηρετητή, αρκεί η πληροφορία που επιστρέφεται να είναι σε κάποιο κοινώς αποδεκτό πρότυπο (όπως τα JSON ή XML). Από την πλευρά του εξυπηρετητή, ένα API είναι μια διεπαφή που ενεργοποιείται σε μια συγκεκριμένη αίτηση, η οποία συνήθως εκφράζεται σε κάποιο κοινώς αποδεκτό πρότυπο (όπως τα JSON ή XML), το οποίο διατίθεται συνήθως μέσω ενός HTTP web server. Στην Εικόνα 2.6 αναπαρίσταται σχηματικά η λειτουργία μιας διεπαφής API [38], [39].



Εικόνα 2.7: Αναπαράσταση λειτουργίας μιας διεπαφής API [39]

### 2.5.3 Πρότυπο JSON

Το JSON ή JavaScript Object Notation, αποτελεί ένα open-source πρότυπο, ανθρωπίνως κατανοητό, για μετάδοση δεδομένων μορφής ζευγών χαρακτηριστικό – τιμή (attribute-value). Χρησιμοποιείται κυρίως για την μετάδοση δεδομένων μεταξύ εξυπηρετητή και web-service ως εναλλακτικό του XML (Extensible Markup Language). Παρόλο που προέρχεται από την JavaScript, είναι ένα πρότυπο ανεξάρτητο γλώσσας που χρησιμοποιείται σε πολλές γλώσσες προγραμματισμού [40].

Οι βασικοί τύποι δεδομένων που χρησιμοποιεί το JSON είναι:

- **Αριθμός** Ένας προσημασμένος δεκαδικός αριθμό που μπορεί να περιέχει κλασματικό μέρος ή να χρησιμοποιεί εκθετικούς αριθμούς.
- **Αλφαριθμητικό** Μια ακολουθία από κανέναν ή περισσότερους χαρακτήρες Unicode.
- **Boolean** Δυαδικές τιμές, true ή false
- **Πίνακας** Μια διατεταγμένη συλλογή τιμών με καμία ή περισσότερες τιμές, καθεμία συλλογή από τις οποίες μπορεί να είναι οποιοδήποτε τύπου.
- **Αντικείμενο** Μια μη ταξινομημένη συλλογή από ζεύγη χαρακτηριστικό-τιμή, όπου τα ονόματα (που ονομάζονται επίσης κλειδιά) είναι συμβολοσειρές.
- **null** — μια κενή τιμή.

```

{
  "book": [
    {
      "id": "01",
      "language": "Java",
      "edition": "third",
      "author": "Herbert Schildt"
    },
    {
      "id": "07",
      "language": "C++",
      "edition": "second",
      "author": "E.Balagurusamy"
    }
  ]
}

```

Κώδικας 2.8: Παράδειγμα του πρότυπου JSON

## 2.6 Σχετικά

### 2.6.1 Το πρότυπο GTFS

Στην εισαγωγή έγινε αναφορά στο πρότυπο GTFS, ένα καθολικό πρότυπο διαθέσεως δεδομένων για τα δρομολόγια των αστικών συγκοινωνιών που ξεκίνησε από μια εταιρεία αστικών συγκοινωνιών στο Portland, του Oregon των ΗΠΑ και αναπτύχθηκε σε συνεργασία με την Google.

Το GTFS (General Transit Feed Specification) ορίζει μια κοινή μορφή διάθεσης των δρομολογίων, του προγράμματος και των σχετικών γεωγραφικών πληροφοριών των μέσων μαζικής μεταφοράς [42]. Οι οργανισμοί δημόσιων συγκοινωνιών δημοσιεύοντας τα στοιχεία μετακίνησης με το παραπάνω πρότυπο επιτρέπουν στους προγραμματιστές να γράψουν εφαρμογές που χρησιμοποιούν αυτά τα δεδομένα με διαλειτουργικό τρόπο. Ένα σετ δεδομένων GTFS αποτελείται από μια σειρά από αρχεία txt που συγκεντρώνονται σε ένα συμπιεσμένο αρχείο. Κάθε σετ περιγράφει πληροφορίες σχετικά με: τις στάσεις, τις γραμμές, τις διαδρομές, καθώς και άλλα δεδομένα του χρονοδιαγράμματος της κάθε υπηρεσίας. Παράλληλα ορίζεται και το GTFS-realtime μια προδιαγραφή η οποία επιτρέπει στους οργανισμούς μεταφορών να διαθέτουν σε πραγματικό χρόνο στοιχεία σχετικά με την κίνηση του στόλου τους στους προγραμματιστές [43].

Στην Ελλάδα οι αστικές συγκοινωνίες Αθηνών και Θεσσαλονίκης διαθέτουν τα στοιχεία τους σε αυτή την μορφή. Στη συνέχεια (Πίνακας 2.3) παρουσιάζονται τα αρχεία που μπορεί να περιλαμβάνει ένα αντίστοιχο σετ δεδομένων.

Όνομασία αρχείου	Τύπος	Περιγραφή
<b>agency.txt</b>	Υποχρεωτικό	Μια ή περισσότερες εταιρείες μεταφορών που παρέχουν τα δεδομένα
<b>stops.txt</b>	Υποχρεωτικό	Οι στάσεις που διέρχονται τα οχήματα του στόλου
<b>routes.txt</b>	Υποχρεωτικό	Διαδρομές. Οι διαδρομές αναφέρονται σε ένα ή περισσότερα δρομολόγια τα οποία παρουσιάζονται στους επιβάτες ως μία υπηρεσία
<b>trips.txt</b>	Υποχρεωτικό	Δρομολόγια της κάθε διαδρομής. Ένα δρομολόγιο αποτελείται από μία ακολουθία δύο ή περισσότερων στάσεων.
<b>stop_times.txt</b>	Υποχρεωτικό	Οι ώρες που ένα όχημα φθάνει και αναχωρεί από μία στάση για κάθε δρομολόγιο
<b>calendar.txt</b>	Υποχρεωτικό	Οι ημερομηνίες καθώς και οι μέρες που εκτελείται η συγκεκριμένη υπηρεσία
<b>calendar_dates.txt</b>	Προαιρετικό	Οι εξαιρέσεις λειτουργίας της συγκεκριμένης υπηρεσίας
<b>fare_attributes.txt</b>	Προαιρετικό	Πληροφορίες για τα ναύλα της διαδρομής
<b>fare_rules.txt</b>	Προαιρετικό	Κανόνες για την πολιτική των ναύλων κάθε διαδρομής
<b>shapes.txt</b>	Προαιρετικό	Κανόνες για την απεικόνιση των διαδρομών
<b>frequencies.txt</b>	Προαιρετικό	Συχνότητα δρομολογίων για διαδρομές με μεταβλητή συχνότητα
<b>transfers.txt</b>	Προαιρετικό	Κανόνες για δημιουργία συνδέσεων μεταξύ των διαδρομών
<b>feed_info.txt</b>	Προαιρετικό	Επιπλέον πληροφορίες σχετικά με το σύνολο δεδομένων

Πίνακας 2.2: Αρχεία ενός GTFS feed [42]

### 2.6.2 Σχετικές εφαρμογές

Αναζητώντας κανείς στο διαδίκτυο και τα καταστήματα εφαρμογών θα βρει αρκετές εφαρμογές που στοχεύουν στην ενημέρωση των χρηστών ΜΜΜ. Η καθιέρωση προτύπων για την διάθεση των δεδομένων επιτρέπει την εφαρμογή και εξάπλωση των συστημάτων που αναπτύσσουν οι προγραμματιστές σε παγκόσμια κλίμακα. Οι εφαρμογές αυτές ποικίλουν ανάλογα με το εύρος των λειτουργιών που διαθέτουν στους χρήστες. Άλλες ασχολούνται με την δρομολόγηση με χρήση συγκοινωνιών και την εύρεση του βέλτιστου συνδυασμού των διαθέσιμων μέσων μεταφοράς, άλλες με την ενημέρωση του επιβατικού κοινού σχετικά με το πρόγραμμα και το δίκτυο ενός φορέα, άλλες με την ζωντανή ενημέρωση του χρήστη για πληροφορίες του δικτύου.

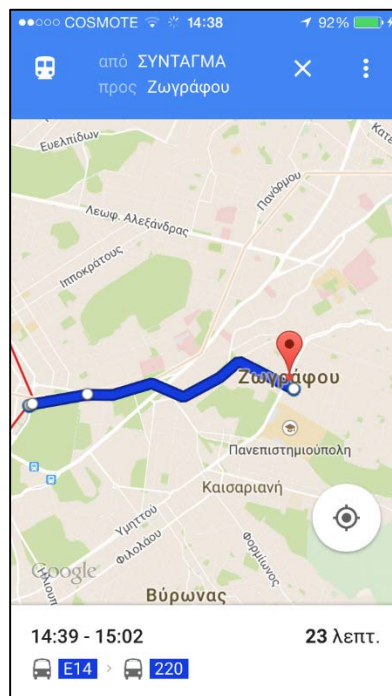
Σήμερα, οι μόνες εφαρμογές που αφορούν το υπεραστικό δίκτυο της χώρας έχουν να κάνουν με την ενημέρωση σχετικά με τις ώρες αναχώρησης των δρομολογίων. Παράλληλα,

υπάρχουν διαθέσιμες εφαρμογές στα ελληνικά καταστήματα που έχουν εφαρμογή στα αστικά δίκτυα συγκοινωνιών Αθήνας και Θεσσαλονίκης, που με ποικίλο τρόπο συνδυάζουν τις διάφορες λειτουργίες που αναφέρθηκαν προηγουμένως κάνοντας χρήση του διαθέσιμου GTFS feed. Σε αυτό το κεφάλαιο γίνεται αναφορά σε ορισμένες από τις σημαντικότερες και πιο διαδεδομένες εφαρμογές που πραγματεύονται το συγκεκριμένο θέμα και έχουν υλοποιηθεί στην χώρα.

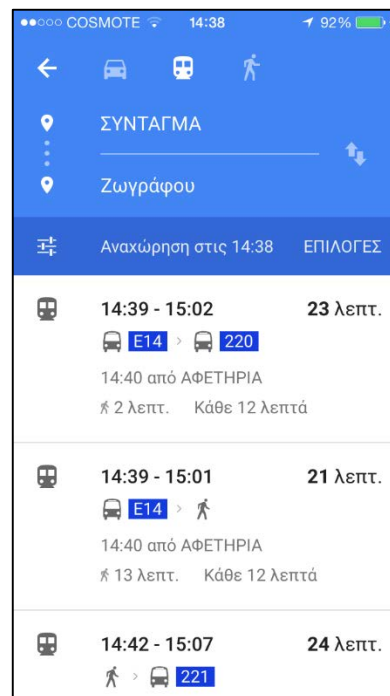
### Google Maps Transit

Οι χάρτες της Google τόσο μέσα από την ιστοσελίδα τους όσο και μέσα από την εφαρμογή στις πλατφόρμες κινητών παρέχουν την υπηρεσία δρομολόγησης Google Transit μεταξύ δύο ή περισσότερων σημείων με χρήση όλων των διαθέσιμων μέσων. Ο χρήστης έχει την δυνατότητα να επιλέξει τα ΜΜΜ και τον επιθυμητό χρόνο έναρξης έτσι ώστε να οργανώσει το ταξίδι του [44].

Από το πρώτο τρίμηνο του 2012 ξεκίνησε και για την Ελλάδα η προσθήκη του δικτύου συγκοινωνιών της χώρας στους χάρτες της Google, με πρώτα τα δρομολόγια του δικτύου της Αθήνας και στη συνέχεια των αστικών συγκοινωνιών της Θεσσαλονίκης (Εικόνες 2.8-2.9).



Εικόνα 2.8: Στιγμιότυπο υπηρεσίας δρομολόγησης της Εφαρμογής Google Maps 1

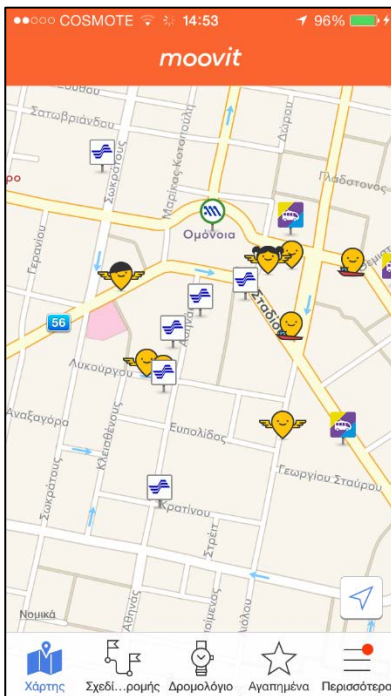


Εικόνα 2.9: Στιγμιότυπο υπηρεσίας δρομολόγησης της Εφαρμογής Google Maps 2

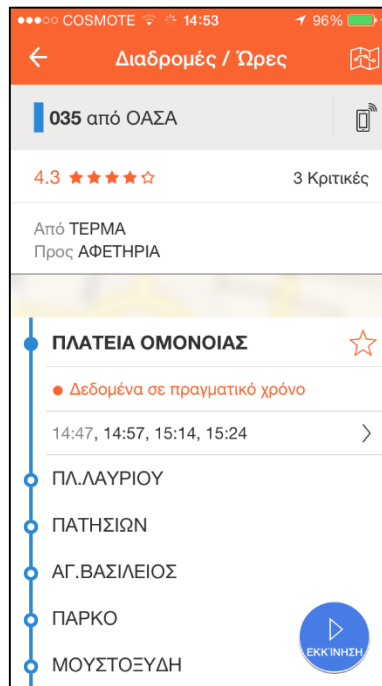


## Moovit

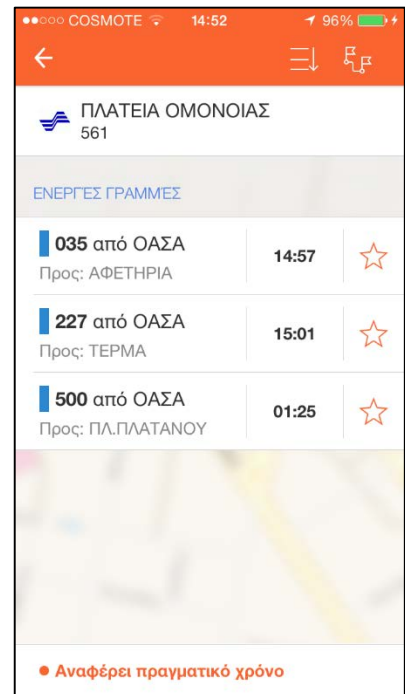
Το Moovit είναι μια δωρεάν εφαρμογή δρομολόγησης με χρήση των MMM [45]. Επίσης παρέχει ζωντανή ενημέρωση σχετικά με τις ώρες άφιξης και αναχώρησης των μέσων από τις στάσεις, χάρτες των στάσεων και γραμμών, ενημέρωση για το πρόγραμμα δρομολογίων και πληροφορίες κατά την διάρκεια του ταξιδιού. Έχει εφαρμογή σε περισσότερες από 500 πόλεις στον κόσμο. Αναπτύχθηκε από την Ισραηλινή εταιρεία Tranzmate Ltd. Η εφαρμογή έκανε διαθέσιμη την πρώτη δοκιμαστική έκδοση το δεύτερο τρίμηνο του 2012, και στο τέταρτο τρίμηνο του 2012 ξεκίνησε η παγκόσμια εφαρμογή της. Στην Ελλάδα έχει βρει εφαρμογή από το 2014 στο δίκτυο της Αθήνας και της Θεσσαλονίκης (Εικόνες 2.10-2.12).



Εικόνα 2.10: Στιγμιότυπο Εφαρμογής Moovit 1



Εικόνα 2.11: Στιγμιότυπο Εφαρμογής Moovit 2



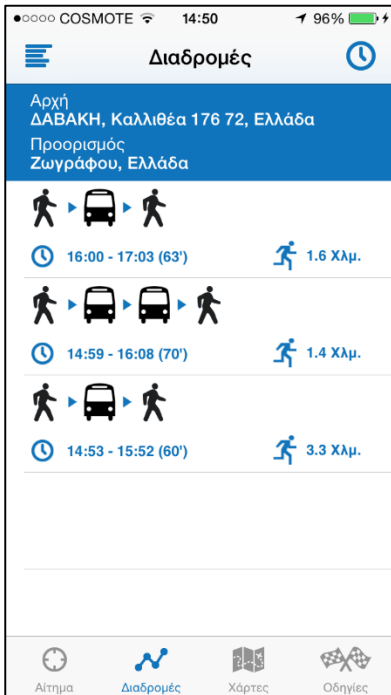
Εικόνα 2.12: Στιγμιότυπο Εφαρμογής Moovit 3

## OptiTrans

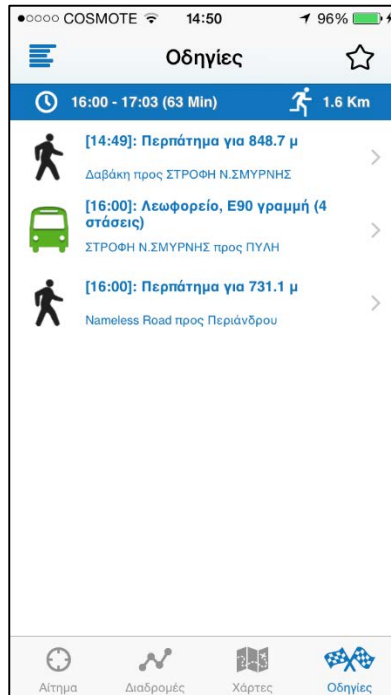
Το OptiTrans είναι ο πρώτος ολοκληρωμένος προγραμματιστής διαδρομών που εφαρμόστηκε στην χώρα και συγκεκριμένα στην Αθήνα το 2011 [46]. Αναπτύχθηκε, στα πλαίσια πανευρωπαϊκών συνεργασιών, από το Εργαστήριο Ολοκληρωμένων Συστημάτων του Ινστιτούτου Πληροφορικής & Τηλεπικοινωνιών του ΕΚΕΦΕ Δημόκριτος. Σήμερα, έχει εφαρμογή σε επτά πόλεις παγκοσμίως και παρέχει πληροφορίες για όλα τα αστικά MMM των πόλεων, όπως μετρό, λεωφορεία, τρόλεϊ, τραμ, προαστιακό ενώ ο χρήστης μπορεί να επιλέξει και ταξί. Το



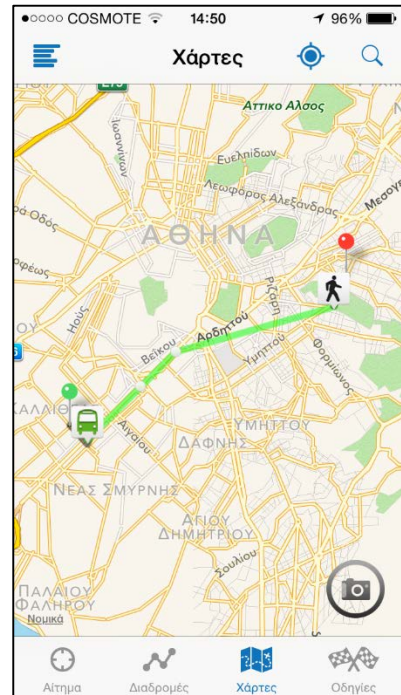
OrtiTrans παρέχει βέλτιστες συνδυαστικές λύσεις για τη χρήση των MMM από ένα σημείο αναχώρησης σε ένα σημείο προορισμού και ενσωματώνει τη χρήση GPS για αυτοματοποιημένη εισαγωγή της αφετηρίας ή προορισμού μιας διαδρομής.



Εικόνα 2.13: Στιγμιότυπο Εφαρμογής Ortitrans 1



Εικόνα 2.14: Στιγμιότυπο Εφαρμογής Ortitrans 2



Εικόνα 2.15: Στιγμιότυπο Εφαρμογής Ortitrans 3



## 3. Μεθοδολογία

Στην προηγούμενη ενότητα αναλύθηκαν οι βασικότερες τεχνολογίες και έννοιες, οι οποίες χρησιμοποιούνται στην εν λόγω εργασία. Στη συνέχεια, περιγράφεται αναλυτικά η διαδικασία σχεδιασμού και υλοποίησης της εφαρμογής.

### 3.1 Περιγραφή εφαρμογής

Στα καταστήματα εφαρμογών υπάρχει πληθώρα εφαρμογών, όπως αναφέρθηκε και στην Παράγραφο 2.6.2, που αντιμετωπίζουν διαφορετικές πτυχές του ίδιου προβλήματος, με έμφαση σε ξεχωριστά στοιχεία η καθεμία. Η εφαρμογή που αναπτύχθηκε, στοχεύει στην καλύτερη οργάνωση και πραγματοποίηση του ταξιδιού, σε συνδυασμό με την ικανοποίηση, την εξυπηρέτηση και την έγκυρη ενημέρωση των επιβατών του υπεραστικού δικτύου συγκοινωνιών.

#### 3.1.1 Λειτουργίες

Ο σχεδιασμός της εφαρμογής επικεντρώθηκε στην επίλυση προβλημάτων που πηγάζουν από τη χρήση των υπεραστικών συγκοινωνιών. Για την επίλυση αυτών, αρχικά, καθορίστηκαν οι επιθυμητές λειτουργίες της εφαρμογής που συνοπτικά παρουσιάζονται στην παρούσα παράγραφο.

Μια απαραίτητη λειτουργία σε οποιαδήποτε εφαρμογή αυτής της κατηγορίας, είναι η πληροφόρηση του χρήστη για στατικά δεδομένα σχετικά με την διάρθρωση και λειτουργία του δικτύου. Στη συγκεκριμένη περίπτωση, τα στοιχεία αυτά περιλαμβάνουν τις γραμμές και τις στάσεις που εξυπηρετεί το δίκτυο, το πρόγραμμα δρομολογίων και τους χάρτες των γραμμών. Ωστόσο, η εφαρμογή επικεντρώνεται και στη διάθεση αυτών των πληροφοριών και εκτός σύνδεσης. Αυτό γίνεται, καταρχάς, διότι ο όγκος των δεδομένων σχετικά με το δίκτυο είναι ιδιαίτερα μεγάλος (χρησιμοποιήθηκαν πάνω από 600 στάσεις και περίπου 100 γραμμές και σε πραγματικές συνθήκες οι αριθμοί αυτοί είναι αρκετά μεγαλύτεροι) για τη λήψη τους μέσω του δικτύου κινητής τηλεφωνίας κατά την λειτουργία της εφαρμογής. Δεύτερον, επιτυγχάνεται η ελαχιστοποίηση της αναμονής κατά την λήψη αυτών και τρίτον, η μείωση του φόρτου του εξυπηρετητή από τις αιτήσεις προς αυτόν. Επίσης, στα πλαίσια της διάθεσης δεδομένων και εκτός σύνδεσης η εφαρμογή συνοδεύεται με μέρος χάρτη που βρίσκεται αποθηκευμένος στον τοπικό αποθηκευτικό χώρο της συσκευής.

Μέσα στις ανάγκες του συγκεκριμένου δικτύου, είναι και ο προσδιορισμός της πλησιέστερης στον χρήστη στάσης με χρήση των δυνατοτήτων εντοπισμού θέσης της συσκευής. Σε αυτή τη λειτουργία υπάρχει και η επιλογή χειροκίνητης εισαγωγής της θέσης, σε περίπτωση αδυναμίας προσδιορισμού της πραγματικής. Επιπλέον, για την ενημέρωση του χρήστη σχετικά με την τοποθεσία του, πραγματοποιείται αντίστροφη γεωκωδικοποίηση αυτής και προσδιορισμός της τοποθεσίας του σε επίπεδο οικισμού και νομού.

Ένας ακόμα από τους στόχους της εφαρμογής, είναι διάθεση δυναμικών στοιχείων σχετικά με την κίνηση του στόλου. Η εξέταση της υποδομής του ΚΤΕΛ επιτρέπει κάτι τέτοιο, ωστόσο, στα πλαίσια της εργασίας χρησιμοποιήθηκαν ενδεικτικά δεδομένα που προσεγγίζουν την πραγματική λειτουργία του δικτύου. Αναφορικά με τις υπηρεσίες, αυτές αντιστοιχούν σε μια ημέρα λειτουργίας του συστήματος, ενώ για την κίνηση των λεωφορείων αντίστοιχο API στον εξυπηρετητή είναι υπεύθυνο για την κίνηση ενός οχήματος κατά μήκος της διαδρομής που εκτελεί.

Στην προηγούμενη λειτουργία συμπεριλαμβάνεται η ενημέρωση του χρήστη για την κίνηση του λεωφορείου που επιβαίνει. Η δυσκολία στον προσδιορισμό του επιθυμητού λεωφορείου επιβάλλει τη διάθεση αρκετών εναλλακτικών επιλογών για αυτή τη διαδικασία (όπως επιλογή πλησιέστερου λεωφορείου, ή λεωφορείου που εκτελεί κάποια γραμμή).

Ο μεγάλος αριθμός στάσεων και γραμμών καθιστά δύσκολη την εύρεση των επιθυμητών στις μεγάλες λίστες που δημιουργούνται. Για το λόγο αυτό, υπάρχει η δυνατότητα αναζήτησης στις λίστες, αλλά και η δημιουργία λίστας αγαπημένων γραμμών και στάσεων στην πλευρική εργαλειοθήκη της οθόνης έναρξης, για γρήγορη πρόσβαση στη ζωντανή παρακολούθηση αυτών.

### 3.1.2 Αρχιτεκτονική συστήματος

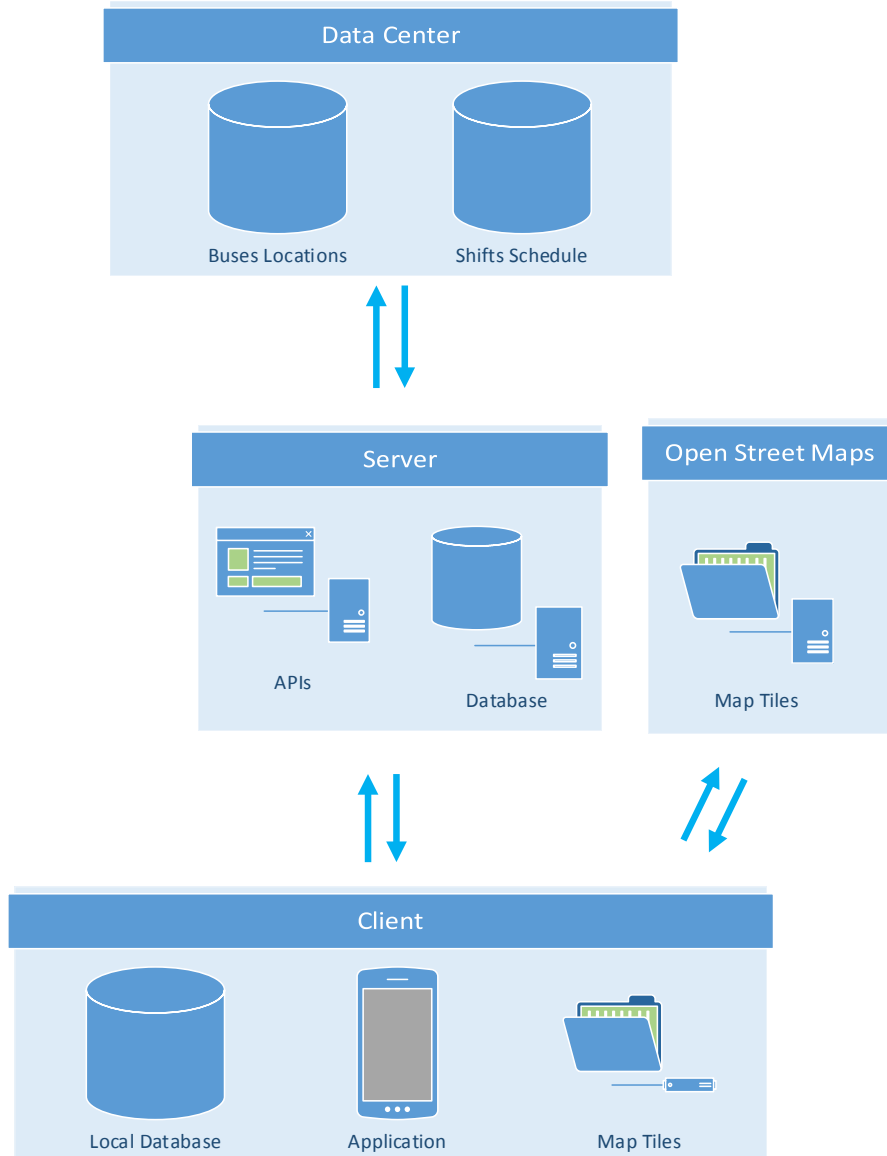
Για την υλοποίηση της εφαρμογής απαιτείται η δημιουργία ενός συστήματος που λειτουργεί σε επίπεδο πελάτη – εξυπηρετητή. Από την πλευρά του εξυπηρετητή, υλοποιήθηκε ένας Web Server (Linux, NGINX, PostgreSQL και PHP), οπου διατηρεί τη βάση δεδομένων με όλα τα δεδομένα του συστήματος. Ο πελάτης υποβάλλει το αίτημά του μέσω της διεπαφής της εφαρμογής, το οποίο μεταβιβάζεται στον εξυπηρετητή. Στον εξυπηρετητή ενεργοποιούνται τα αντίστοιχα APIs τα οποία υποβάλλουν τα ανάλογα ερωτήματα στη βάση δεδομένων και τα αποτελέσματα επιστρέφονται στον πελάτη σε κάποιο κοινώς αποδεκτό πρότυπο.

Στην πρώτη σύνδεση του χρήστη, η εφαρμογή πραγματοποιεί λήψη και αποθήκευση μεγάλου μέρους των δεδομένων στην βάση δεδομένων της συσκευής. Ένα πεδίο με την έκδοση συνοδεύει κάθε σύνολο δεδομένων, ούτως ώστε να πραγματοποιείται η ανανέωση αυτών όταν υπάρχουν μεταβολές. Η εγκατάσταση της εφαρμογής συνοδεύεται με τον χάρτη που εξυπηρετεί το δίκτυο σε περιορισμένο εύρος επιπέδου εστίασης, για τη δυνατότητα προβολής στοιχείων του δικτύου και εκτός σύνδεσης.

Το Data Center (Διάγραμμα 3.1) αντιπροσωπεύει το σύστημα του παρόχου για τη διάθεση στοιχείων σχετικά με τις υπηρεσίες και τις τοποθεσίες των λεωφορείων σε πραγματικό χρόνο. Σε πραγματικές συνθήκες ο εξυπηρετητής επικοινωνεί με το σύστημα του παρόχου και λαμβάνει τις ημερήσιες υπηρεσίες των λεωφορείων ή ενδεχόμενες τροποποιήσεις αυτών, ενώ η λήψη της τοποθεσίας των λεωφορείων πραγματοποιείται με προκαθορισμένη χρονική επανάληψη.

Επίσης, ο χρήστης επικοινωνεί με το API του Open Street Maps για τη λήψη των map tiles που χρησιμοποιούνται στην παρουσίαση των χαρτών. Κατά τη λειτουργία εκτός σύνδεσης, η εφαρμογή, κάνει χρήση των map tiles που είναι αποθηκευμένα στη συσκευή.

Συγκεντρωτικά, στο Διάγραμμα 3.1 φαίνεται η αρχιτεκτονική του συστήματος και οι διάφορες επικοινωνίες μεταξύ των επιμέρους στοιχείων του.



Διάγραμμα 3.1: Αρχιτεκτονική συστήματος

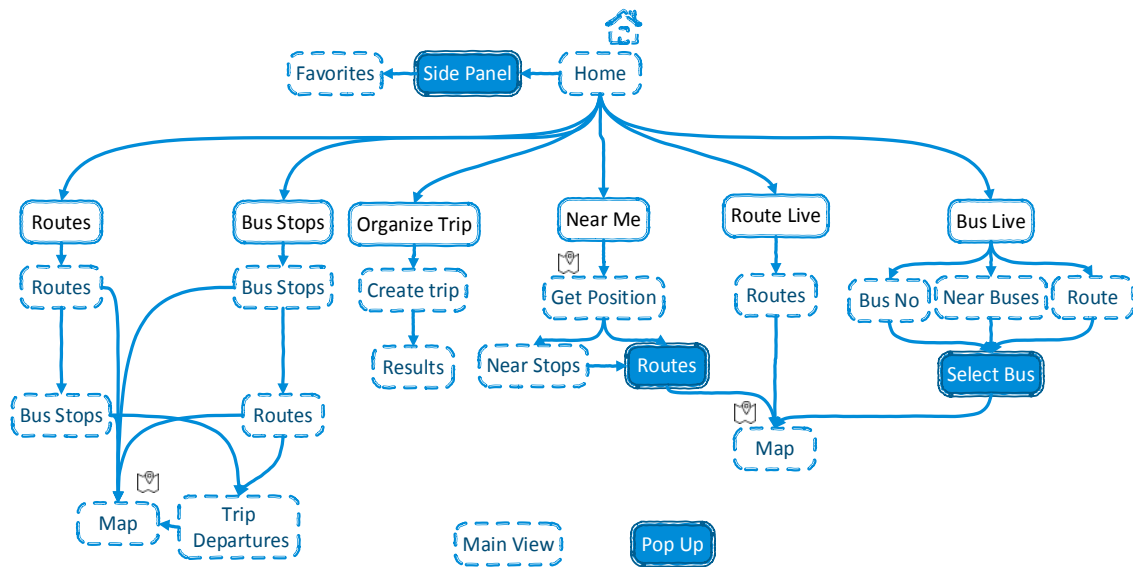
Σε αυτό το σημείο πρέπει να αναφερθεί πως το σύστημα αναπτύχθηκε με χρήση λογισμικών και τεχνολογιών ανοιχτού κώδικα (open-source), δηλαδή εργαλείων που ο καθένας μπορεί ελεύθερα και χωρίς κάποια άδεια να χρησιμοποιεί, να διανέμει, να αντιγράφει και να τροποποιεί ανάλογα με τις ανάγκες του.

### 3.2 Διεπαφή χρήστη

Το παρών κεφάλαιο πραγματεύεται το θέμα του σχεδιασμού και της υλοποίησης της διεπαφής χρήστη. Συγκεκριμένα, αναλύονται οι απαιτήσεις της κάθε λειτουργίας, η δομή του client-side τμήματος του συστήματος και στη συνέχεια παρουσιάζεται η κωδικοποίηση των αρχείων της εφαρμογής.

#### 3.2.1 Σχεδιασμός διεπαφής χρήστη

Η εφαρμογή αποτελείται από 6 κύριες λειτουργίες, που ενεργοποιούνται μέσα από συνολικά 16 όψεις, μία πλευρική εργαλειοθήκη και τρία παράθυρα ειδοποιήσεων. Έπειτα, περιγράφεται η κάθε λειτουργία και η σειρά των ενεργειών που μπορούν να πραγματοποιηθούν εντός αυτών.

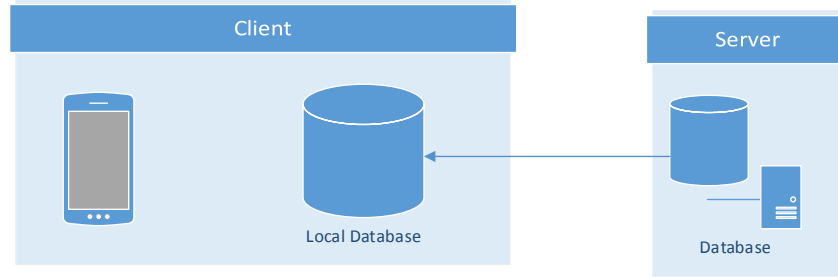


Διάγραμμα 3.2: Αναπαράσταση λειτουργίας εφαρμογής

#### Κατά την εκκίνηση

Σε κάθε εκτέλεση της εφαρμογής εξετάζεται αν έχει ξαναλειτουργήσει στη συγκεκριμένη συσκευή. Κατά την πρώτη εκτέλεση, πραγματοποιείται λήψη των δεδομένων και αποθήκευση αυτών σε μια τοπική βάση δεδομένων (Παράγραφος 2.2.2). Η αποθήκευση των δεδομένων τοπικά εξασφαλίζει την ελαχιστοποίηση της αναμονής και του όγκου των δεδομένων που ανταλλάσσονται με τον εξυπηρετητή, ενώ παράλληλα η χρήση βάσης δεδομένων δίνει την δυνατότητα εύκολης ανανέωσης τους αλλά και πραγματοποίησης ερωτημάτων σε αυτήν.

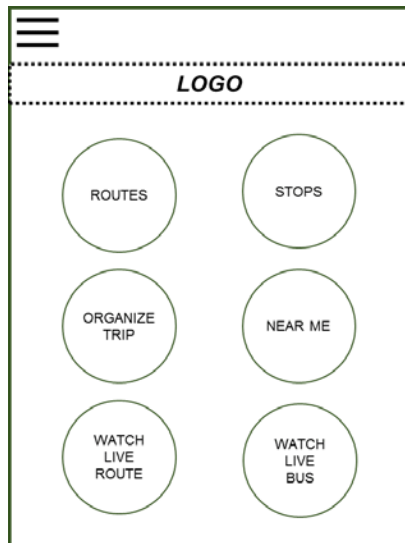
Σε περίπτωση που η εφαρμογή έχει ξαναλειτουργήσει στη συσκευή και είναι διαθέσιμα απαραίτητα δεδομένα, γίνεται έλεγχος της έκδοσης αυτών και ανανέωση τους σε ενδεχόμενες μεταβολές.



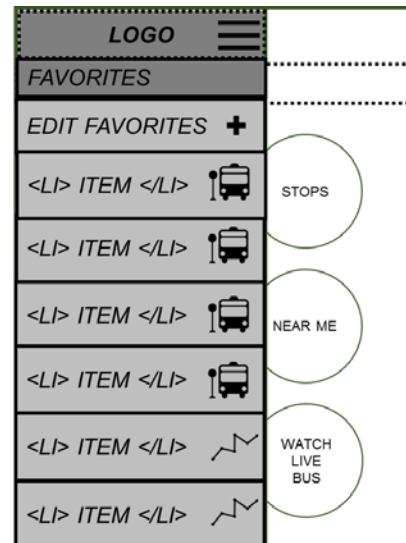
Διάγραμμα 3.3: Αναπαράσταση ανταλλαγής δεδομένων κατά την πρώτη έναρξη

### Αρχική Οθόνη

Η αρχική οθόνη που μεταβαίνει ο χρήστης κατά την έναρξη, περιλαμβάνει όλες τις διαθέσιμες λειτουργίες της εφαρμογής (Εικόνα 3.1). Στην αριστερή πλευρά ενεργοποιείται η πλευρική εργαλειοθήκη με μία λίστα αγαπημένων (Εικόνα 3.2), η οποία μπορεί να επεξεργαστεί ο χρήστης, για γρήγορη μετάβαση στην ζωντανή παρακολούθηση των αγαπημένων του γραμμών και στάσεων.



Εικόνα 3.1: Σχεδιάγραμμα αρχικής οθόνης 1

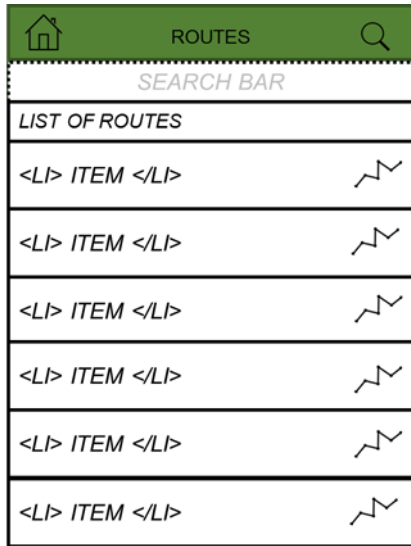


Εικόνα 3.2: Σχεδιάγραμμα αρχικής οθόνης 2

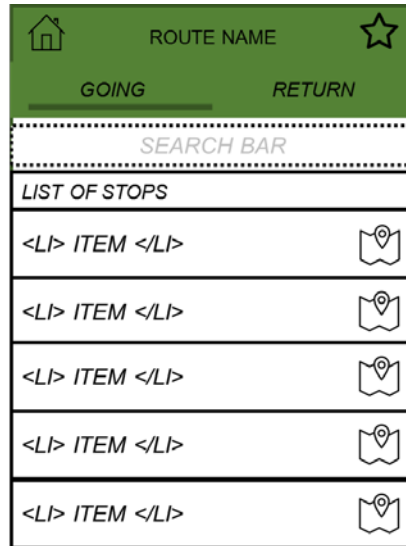
### Γραμμές

Στην οθόνη με τις γραμμές, παρουσιάζονται σε μορφή λίστας όλες οι διαθέσιμες γραμμές του δικτύου (Εικόνα 3.3). Στην πάνω αριστερή πλευρά υπάρχει συντόμευση για ενεργοποίηση της μπάρας αναζήτησης. Επιλέγοντας μια από τις γραμμές παρουσιάζεται λίστα των στάσεων που εξυπηρετεί αυτή, ταξινομημένων με τη σειρά όπου διέρχεται το λεωφορείο και διαχωρισμένων σε μετάβαση και επιστροφή (Εικόνα 3.4). Επίσης, υπάρχει και συντόμευση για την μετάβαση στην προβολή του χάρτη της γραμμής ή την προσθήκη της γραμμής στην λίστα αγαπημένων.

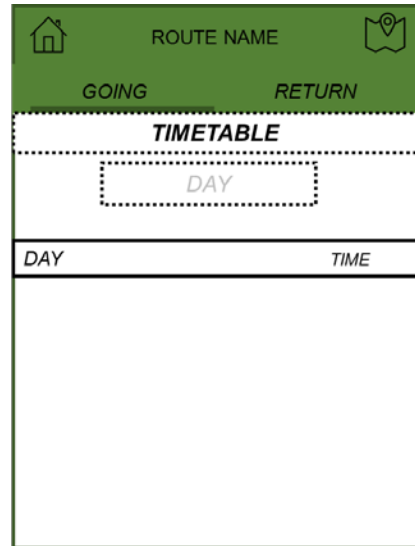
Εάν ο χρήστης επιλέξει να δει τις στάσεις της γραμμής, έχει τη δυνατότητα να ενημερωθεί για πληροφορίες σχετικά με τις ώρες και ημέρες πραγματοποίησης του δρομολογίου (Εικόνα 3.5) ή να μεταβεί ξανά στον χάρτη της γραμμής αυτή τη φορά με επισημασμένη την επιλεγμένη στάση.



Εικόνα 3.3: Σχεδιάγραμμα οθόνης γραμμών 1



Εικόνα 3.4: Σχεδιάγραμμα οθόνης γραμμών 2



Εικόνα 3.5: Σχεδιάγραμμα οθόνης γραμμών 3

### Στάσεις

Η λειτουργία των στάσεων είναι αντίστροφη αυτής των γραμμών. Εδώ ο χρήστης ξεκινάει από την επιλογή της στάσης ενδιαφέροντος και μπορεί να δει τις γραμμές που εξυπηρετούν τη συγκεκριμένη στάση, να προσθέσει τη στάση στις αγαπημένες του, να προβάλει στον χάρτη την τοποθεσία της επιλεγμένης στάσης. Οι στάσεις είναι οργανωμένες σε επίπεδο νομού για την αποφυγή δημιουργίας μεγάλων σε μέγεθος λιστών.




## Οργάνωση Ταξιδιού

Η οργάνωση του ταξιδιού στοχεύει στον βέλτιστο συνδυασμό των διαδρομών του δικτύου για την μετάβαση μεταξύ δύο στάσεων (Εικόνα 3.6). Το αποτέλεσμα της δρομολόγησης βασίζεται στην επιλογή του καλύτερου συνδυασμού διαδρομών για την ελαχιστοποίηση:

- των μετεπιβιβάσεων
- του αριθμού των ενδιάμεσων στάσεων

Αφού υποβληθεί το ερώτημα στον εξυπηρετητή και ληφθεί απάντηση, παρουσιάζονται οι διαθέσιμες επιλογές στην μορφή:

- Στάση εκκίνησης ή μετεπιβίβασης
- Ονομασία διαδρομής
- Αριθμός ενδιάμεσων στάσεων
- Στάση προορισμού ή μετεπιβίβασης.



Εικόνα 3.6: Σχεδιάγραμμα οθόνης οργάνωσης ταξιδιού

## Κοντινές Στάσεις

Η λειτουργία των *Κοντινών Στάσεων* είναι ιδιαίτερα σημαντική στους επισκέπτες που δεν έχουν καμία γνώση της διάρθρωσης του δικτύου. Μεταβαίνοντας στη συγκεκριμένη οθόνη, γίνεται προσπάθεια για πρόσβαση στην τοποθεσία του χρήστη. Σε περίπτωση επιτυχίας, εξετάζεται η ακρίβεια που συνοδεύει το αποτέλεσμα και εάν αυτή είναι ικανοποιητική ξεκινάει η διαδικασία αντίστροφης γεωκωδικοποίησης της τοποθεσίας σε επίπεδο οικισμού και νομού. Σε περίπτωση αδυναμίας προσδιορισμού της τοποθεσίας, υπάρχει η δυνατότητα επιλογής κάποιου σημείου στον χάρτη χειροκίνητα.

Αφότου προσδιοριστεί η τοποθεσία (είτε χειροκίνητα είτε αυτόματα), προβάλλεται μία λίστα με τις κοντινές στην επιλεγμένη τοποθεσία στάσεις. Σε κάθε στάση υπάρχει ένας δείκτης όπου εκφράζει την απόσταση της στάσης από την αιτούμενη τοποθεσία. Εδώ υπάρχουν δύο επιλογές: η προβολή των γραμμών που εξυπηρετούν τη συγκεκριμένη στάση, και η προβολή χάρτη με τις κοντινές στάσεις και την πληροφορία της πρώτης επιλογής να λαμβάνεται μέσω popup παραθύρου.

## Ζωντανή Παρακολούθηση Γραμμών

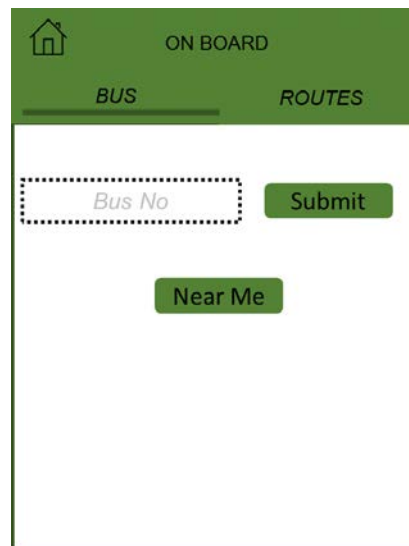
Στην σελίδα *Ζωντανής Παρακολούθησης Γραμμών*, αρχικά επιλέγεται μία γραμμή του δικτύου. Με την επιλογή γραμμής ο χρήστης μεταβαίνει στην προβολή αυτής στον χάρτη με όλα τα λεωφορεία που εκτελούν το συγκεκριμένο δρομολόγιο την δεδομένη χρονική στιγμή. Η τοποθεσία αυτών ανανεώνεται αυτόματα σε προκαθορισμένο χρόνο.

### Παρακολούθηση Λεωφορείου Ενδιαφέροντος

Η επιλογή παρακολούθησης κάποιου λεωφορείου προσφέρει στον χρήστη πληροφορίες για την πορεία του λεωφορείου που είναι επιβιβασμένος, αλλά και σε οποιονδήποτε ενδιαφέρεται για την τοποθεσία του συγκεκριμένου λεωφορείου, όπως για παράδειγμα γονείς μαθητών. Η λειτουργία αυτή κρίνεται ιδιαίτερα σημαντική στους επισκέπτες οι οποίοι δεν έχουν γνώση της σειράς των στάσεων και στο συγκεκριμένο ΚΤΕΛ, δεν υπάρχει κάποιος πίνακας εντός του λεωφορείου να ενημερώνει για την επόμενη στάση. Για την εύρεση του λεωφορείου ενδιαφέροντος διατίθεντο μια σειρά από επιλογές, όπως:

- **Εισαγωγή αριθμού λεωφορείου** Ο χρήστης εισάγει απευθείας τον αριθμό του λεωφορείου και επιβεβαιώνει αυτόν αφού του παρουσιαστεί ο προορισμός του λεωφορείου
- **Εύρεση κοντινών λεωφορείων** Αυτή η επιλογή κάνει χρήση της τοποθεσίας του χρήστη και βρίσκει τα κοντινά σε αυτόν λεωφορεία.
- **Επιλογή γραμμής** Εδώ επιλέγεται μια γραμμή και ο χρήστης βρίσκει το λεωφορείο ενδιαφέροντος μέσα από αυτά που εκτελούν τη συγκεκριμένη γραμμή.

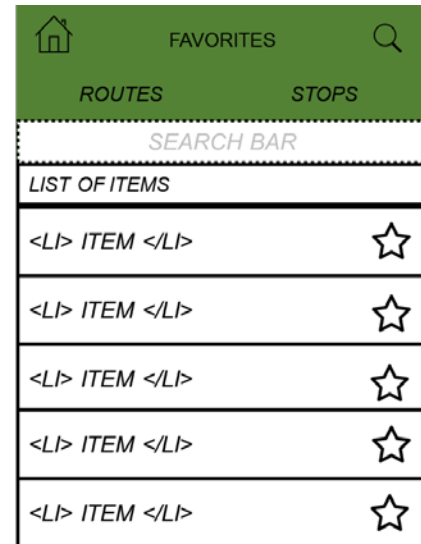
Αφότου ο χρήστης επιλέξει το λεωφορείο ενδιαφέροντος, παρουσιάζεται σε αυτόν ο χάρτης με την επιλεγμένη γραμμή και τη τοποθεσία του λεωφορείου να ανανεώνεται σε προκαθορισμένο χρόνο.



Εικόνα 3.7: Σχεδιάγραμμα οθόνης επιλογής λεωφορείου

### Λίστα Αγαπημένων

Η λίστα αγαπημένων φαίνεται στον πλευρικό πίνακα ελέγχου της αρχικής σελίδας. Ο χρήστης μπορεί να προσθέσει σε αυτήν γραμμές και στάσεις, ώστε να μεταβαίνει γρήγορα στην παρακολούθηση αυτών. Η μεν επιλογή στάσης οδηγεί τον χρήστη στην παρουσίαση των διαδρομών που διέρχονται από αυτή, ενώ η επιλογή γραμμής στην ζωντανή παρακολούθηση αυτής. Υπάρχει ξεχωριστή σελίδα διαχείρισης των στοιχείων της λίστας (Εικόνα 3.8), ενώ η αποθήκευση των στοιχείων γίνεται σε πίνακα της τοπικής βάσης δεδομένων.



Εικόνα 3.8: Σχεδιάγραμμα οθόνης επεξεργασίας αγαπημένων

### Χάρτες

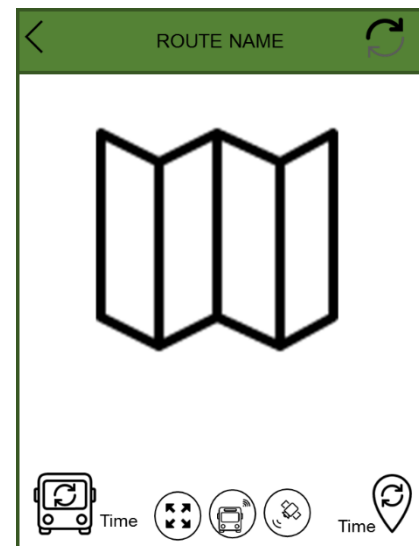
Σημαντικό ρόλο για την λειτουργικότητα της εφαρμογής έχει η διάρθρωση των χαρτών. Σε μια μικρή οθόνη πρέπει να συνοψιστεί ένας μεγάλος όγκος πληροφορίας, ενώ απαιτείται και η ενσωμάτωση πλήκτρων για ειδικές λειτουργίες χειρισμού των χαρτών.

Στις προβολές του χάρτη των γραμμών (Εικόνα 3.9), των στάσεων και της ζωντανής παρακολούθησης αυτών ο χρήστης έχει μια σειρά από επιπλέον επιλογές όπως:

- **Προσαρμογή ορίων του χάρτη** Επέκταση των ορίων του χάρτη για προβολή όλων των στοιχείων που απεικονίζονται σε αυτόν.
- **Λήψη τοποθεσίας** Εύρεση της τοποθεσίας του χρήστη εφόσον αυτός το έχει επιτρέψει (με χρήση του δέκτη GPS ή του δικτύου κινητής τηλεφωνίας ή του σήματος του WiFi που αυτός είναι συνδεδεμένος).
- **Ανανέωση στοιχείων** Προβολή ή ανανέωση της τοποθεσίας των λεωφορείων που αφορά η επιλεγμένη γραμμή.

Οι δύο τελευταίες επιλογές, εφόσον είναι επιτυχής προσθέτουν μια χρονική ένδειξη της ανανέωσης τους στο κάτω μέρος της οθόνης.

Να σημειωθεί ότι η προβολή του χάρτη αναφέρεται σε γραμμή. Για την εναλλαγή μεταξύ μετάβασης και επιστροφής υπάρχει αντίστοιχο εικονίδιο στην επάνω δεξιά πλευρά της οθόνης.



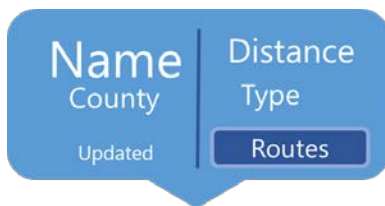
Εικόνα 3.9: Σχεδιάγραμμα οθόνης χαρτών

### PorUp Χαρτών

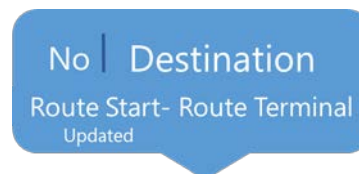
Στην προβολή των χαρτών, επιλέγοντας κάποιων marker λεωφορείου ή στάσης δίνονται επιπλέον πληροφορίες αντίστοιχες της λειτουργίας που βρίσκεται ο χρήστης (Εικόνες 3.10 έως 3.12). Στα δυναμικά δεδομένα υπάρχει επιπλέον η χρονική ένδειξη που συνοδεύει τα αποτελέσματα.



Εικόνα 3.10: Σχεδιασμός Porup στάσης



Εικόνα 3.11: Σχεδιασμός Porup κοντινής στάσης



Εικόνα 3.12: Σχεδιασμός Porup λεωφορείου

### Χάρτες εκτός σύνδεσης

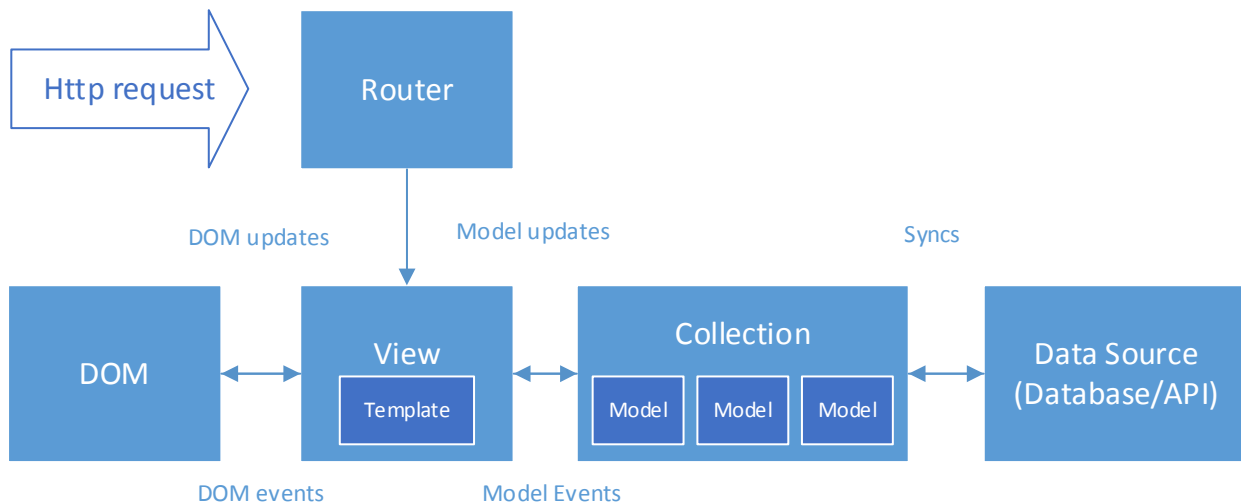
Στα πλαίσια της δυνατότητας χρήσης της εφαρμογής και εκτός σύνδεσης η εφαρμογή συνοδεύεται με ένα μέρος του χάρτη των περιοχών που εξυπηρετεί το δίκτυο (με περιορισμένο εύρος επίπεδου εστίασης). Η επιλογή για διάθεση μονάχα ενός τμήματος οφείλεται στο όγκο των δεδομένων, καθώς το δίκτυο απλώνεται σχεδόν στο σύνολο της χώρας κάτι που αντιστοιχεί σε πάνω από 100mb δεδομένα.

Για την επεξεργασία και λήψη των Map Tiles γίνεται χρήση του λογισμικού Mobile Atlas Creator [50]. Το λογισμικό Mobile Atlas Creator είναι ένα ανοιχτού κώδικα λογισμικό υπό την άδεια GPL για την λήψη και διαχείριση τμημάτων από χάρτες που διανέμονται δωρεάν στο διαδίκτυο σε διάφορες μορφές αποθήκευσης, όπως .png, .jpg, SQLiteDB και άλλες, για χρήση τους σε mobile εφαρμογές. Τα δεδομένα των χαρτών που χρησιμοποιούνται στην παρούσα εφαρμογή προέρχονται από το Open Street Maps, ένα συνεργατικό έργο για την δημιουργία και ελεύθερη διάθεση ενός επεξεργάσιμου χάρτη του κόσμου με βάση την άδεια Open Database License.

### 3.2.2 Υλοποίηση διεπαφής χρήστη

Στην δεύτερη ενότητα παρουσιάστηκαν τα πλεονέκτημα ανάπτυξης υβριδικών εφαρμογών. Με βάση τα πλεονεκτήματα της συγκεκριμένης προσέγγισης και απώτερο στόχο την διάθεση της εφαρμογής σε όσο το δυνατόν περισσότερες πλατφόρμες η εφαρμογή αναπτύχθηκε με τεχνολογίες διαδικτύου. Ωστόσο δεν υπάρχει ένα καθολικά αποδεκτό πρότυπο και συνδυασμός των διαθέσιμων εργαλείων για την ιδανική υβριδική εφαρμογή και ο προγραμματιστής καλείτε να επιλέξει με βάση της ανάγκες του εκάστοτε προβλήματος [2], [48], [49].

Για την δημιουργία μίας HTML5 εφαρμογής με φιλική διεπαφή χρήστη για κινητές συσκευές και την καλύτερη υποστήριξη σε οθόνες αφής απαιτείται μία HTML5 Mobile UI βιβλιοθήκη JavaScript. Στην ανάπτυξη της εφαρμογής έγινε χρήση της δημοφιλούς βιβλιοθήκης jQuery Mobile. Ωστόσο η πολυπλοκότητα της εφαρμογής, ο μεγάλος αριθμός των προβολών και οι απαιτήσεις δρομολόγησης μεταξύ των λειτουργιών καθιστούσαν αναγκαία την χρήση ενός MV\* Framework, του Backbonejs. Το Backbonejs είναι ένα ελαφρύ πλαίσιο, δεν συνδέεται με περιττές δραστηριότητες και επιτρέπει την χρήση οποιασδήποτε βιβλιοθήκης για την διεπαφή χρήστη. Στο Διάγραμμα 3.4 φαίνεται η λειτουργία του πλαισίου και τα βασικά συστατικά που το απαρτίζουν. Στη συνέχεια του κεφαλαίου παραθέτονται ορισμένα γενικά στοιχεία σχετικά με την χρήση των επιμέρους πλαισίων κατά την λειτουργία της εφαρμογής.



Διάγραμμα 3.4: Σχηματική αναπαράσταση λειτουργίας του πλαισίου Backbonejs

### Router

Η δρομολόγηση στο Backbone γίνεται με χρήση του Router (Κώδικας 3.1). Ένα hashtag (#) υποδεικνύει την διαδρομή και καλεί την αντίστοιχη view (για παράδειγμα «*index.html#routes*»). Μέσα από το URL κατά την δρομολόγηση είναι δυνατό να μεταφερθούν και παράμετροι στην νέα View («*index.html#routes/id*»).

```

var Router = Backbone.Router.extend({
  //Routes
  routes: {
    "": "home",
    "routes": "routes",
    "routes/:id": "routeDetails ",
    "*actions": "home"
  },
  home : function () {
    //Call Home Page View
  },
  routes: function () {
    //Call Routes View
  },
  routeDetails: function (id) {
    //Call RouteDetails View
  }
}

```

Κώδικας 3.1: Backbonejs Router

## Views

Κάθε View αποτελεί μια ξεχωριστή jQuery Mobile σελίδα (<div data-role="page">) και συνοδεύεται από μια τουλάχιστον ξεχωριστή HTML Template (Κώδικας 3.2). Η Template, μέσω της βιβλιοθήκης Underscore.js (βασική εξάρτηση του Backbonejs) συμπληρώνεται και απεικονίζει τα δεδομένα των Models.

```

var View_id = Backbone.View.extend({
  role: "page",
  attributes: function() {
    return {
      "data-role" : this.role
    };
  },
  id: 'view_id',
  template: _.template(Template),
  events: {
    "click #button": 'functionButton'
  },
  render:function () {
    $(this.el).html(this.template());
    this.$el.page().enhanceWithin();
    return this;
  },
  functionButton: function(event) {
    //function button
  }
});

```

Κώδικας 3.2: Backbonejs View

```

<div data-role="header" class="ui-header ">
  <h1>Title</h1>
</div>
<div class="ui-content" role="main">
  <h3>Header</h3>
  <p>Content paragraph</p>
</div>
<div data-role="footer" class="ui-footer ">
  <h3>Copyright 2015</h3>
</div>

```

Κώδικας 3.3: HTML template σελίδας jQuery Mobile

Στον Κώδικα 3.2 διακρίνονται ορισμένα σημαντικά στοιχεία όπως:

**el:** Ως στοιχείο «el» ορίζεται το «body»tag για την απεικόνιση το στοιχείων σε αυτό.

**events:** Τα «events»σε μία View διαχειρίζονται τα «click», «dblclick», «mouseover» στα στοιχεία της.

**render:** Ορίζεται η συνάρτηση «render» που είναι υπεύθυνη για την απόδοση των περιεχομένων των Models χρησιμοποιώντας την JavaScript templating engine και την ανανέωση του περιεχομένου της View που αναφέρεται στο στοιχείο «this.el». Η κλήση της γίνεται κατά την αλλαγή της View μέσω του Router

**\_.template:** Χρήση της βιβλιοθήκης Underscore για την απόδοση της Template.

**data-role:** Κάθε View αποτελεί όπως αναφέρθηκε μία σελίδα jQuery Mobile ("data-role"="page").

### Συγχρονισμός

Στις περισσότερες εφαρμογές μίας σελίδας (SPAs), τα Models προέρχονται από κάποιον αποθηκευτικό χώρο που βρίσκεται σε κάποιον απομακρυσμένο εξυπηρετητή. Σε μία Backbone εφαρμογή απλοποιείται σημαντικά ο απαιτούμενος κώδικας για την πραγματοποίηση συγχρονισμού με κάποιον server. Η εντολή `collections.fetch()` λαμβάνει από τον server ένα σύνολο από Models με την μορφή ενός JSON πίνακα, στέλνοντας μία αίτηση HTTP GET στην διεύθυνση URL που ορίζεται στην δήλωση της Collection.

### Collection

Τα Collections (Κώδικας 3.4) περιλαμβάνουν συλλογές από Models (για παράδειγμα η Collection *Stops* στην οποία ανήκουν τα Models *Stop*). Ο συγχρονισμός των Models της Collection πραγματοποιείται είτε κατόπιν αιτήσεως του χρήστη είτε με ορισμό χρονικών διαστημάτων και χρήση της συνάρτησης `fetch` (Κώδικας 3.5). Ο server απαντάει στις `fetch` αιτήσεις επιστρέφοντας έναν JSON πίνακα από Models.

```

var Stops = Backbone.Collection.extend({
  model: Stop,
  url: "http://example.com/stops.php",
  parse: function(response) {
    return response.results;
  }
});

```

Κώδικας 3.4: Λήψη δεδομένων στις Collections του Backbonejs 1

```

collection.fetch({
  reset: true,
  type: 'GET',
  data: {
    "id": passedID
  },
  success: function (data, response, options){
    myFunction(data);
  },
  error: function (data, response, options){
    alert("failed to load data", response);
  }
});

```

Κώδικας 3.5: Λήψη δεδομένων στις Collections του Backbonejs 2

Για κάθε Model (Κώδικας 3.6) ορίζονται μεταξύ άλλων και οι προεπιλεγμένες τιμές (defaults) όπου συνήθως είναι null.

```

var Stop = Backbone.Model.extend({
  defaults: {
    "id": null,
    "name": null,
    "county": null,
    "type": null,
    "lat": null,
    "lon": null
  }
});

```

Κώδικας 3.6: Παράδειγμα Backbonejs Model

Τα περιεχόμενα των Models χρησιμοποιώντας την JavaScript templating engine συμπληρώνουν τις Templates με τα χαρακτηριστικά τους. Η συνάρτηση myFunction καλείται με



την επιτυχή λήψη των δεδομένων από τον εξυπηρετητή με παράμετρο τα Models που επιστράφηκαν. Με τη συνάρτηση each γίνεται πρόσβαση στο κάθε Model το οποίο συμπληρώνει την αντίστοιχη Template και προστίθεται στο τέλος μίας αταξινομητης λίστας <ul>.

```
myFunction: function (stops) {
  _.each(stops.models, this.append, this);
},
append: function(stopModel) {
  $('#ul-list').append(this.template(stopModel.toJSON()));
}
}
```

Κώδικας 3.7: Δημιουργίας λίστας με χρήση της βιβλιοθήκης Underscore.js

```
<li>
  <a href="#nextPage/<%= id %>" id="<%= id %>">
    <h4><%= name %></h4>
    <p><%= county %></p>
    <p class="ui-li-aside"><% type %></p>
  </a>
</li>
```

Κώδικας 3.8: Template δημιουργίας λίστας με χρήση του Underscore.js

```
"data": [
  {
    "id": 23,
    "name": "CHALKIDA",
    "county": "N. EVVIAS",
    "type": "Stop"
  }, {
    "id": 35,
    "name": "CHANIA",
    "county": "N. CHANION",
    "type": "Station"
  }, {
    "id": 56,
    "name": "CHARAKA",
    "county": "N. IRAKLIU",
    "type": "Stop"
  }, {
    "id": 88,
    "name": "CHORA SFAKION",
    "county": "N. CHANION",
    "type": "Stop"
  }, {

```

<b>CHALKIDA</b> N. EVVIAS	Stop
<b>CHANIA</b> N. CHANION	Station
<b>CHARAKA</b> N. IRAKLIU	Stop
<b>CHORA SFAKION</b> N. CHANION	Stop

Εικόνα 3.13: Δημιουργία λίστας με χρήση της μηχανής templating του Underscore.js

Κώδικας 3.9: Δεδομένα JSON

### 3.2.3 Κωδικοποίηση Αρχείων

Η δημιουργία μιας εφαρμογής με χρήση του Apache Cordova απαιτεί μια συγκεκριμένη δομή των αρχείων και των φακέλων αυτής. Οι φάκελοι `hooks`, `platforms` και `plugins` αφορούν την διαδικασία δημιουργίας της υβριδικής εφαρμογής με το πλαίσιο Cordova, ενώ ο φάκελος `www` περιέχει όλα τα αρχεία και τους πόρους της εφαρμογής.

**hooks/:** Κώδικας που χρησιμοποιείται από το Cordova κατά την διαδικασία δημιουργίας της εφαρμογής.

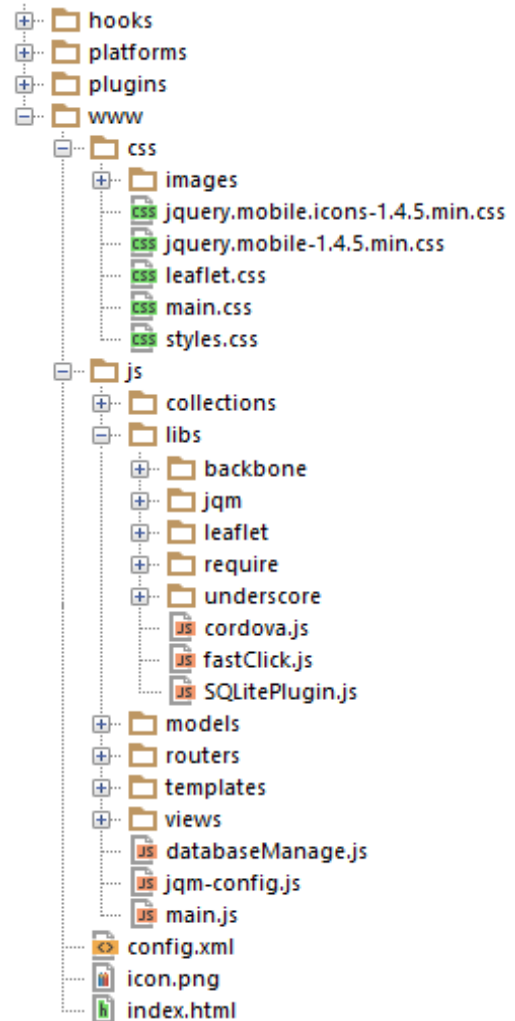
**merges/:** Στο φάκελο `merges` περιλαμβάνονται τα αρχεία HTML, JavaScript και CSS που διαφοροποιούνται μεταξύ των πλατφόρμων. (Στην περίπτωση της εφαρμογής που αναπτύχθηκε δεν υπάρχουν τέτοιου είδους αρχεία αφού η εφαρμογή προορίζεται για λειτουργικό σύστημα android).

**platforms/:** Περιλαμβάνει αρχεία σχετικά με την εκάστοτε πλατφόρμα, που αφορούν την διαδικασία δημιουργίας της εφαρμογής από το Cordova (όπως `Splashscreens` και τα `Εικονίδια`).

**plugins/:** Περιλαμβάνει τα πρόσθετά (`plugin`) του Cordova που έχουν εγκατασταθεί.

**www/:** Στον φάκελο `www` βρίσκονται όλα τα αρχεία της εφαρμογής, όπως HTML, JavaScript, CSS εικόνες και άλλα.

- **css/:** Περιλαμβάνονται τα αρχεία CSS με τις στιλιστικές επιλογές του χρήστη καθώς και όλες οι εικόνες και σύμβολα που συνοδεύουν την εφαρμογή.
- **js/:** Στον φάκελο `js` αντίστοιχα διατηρούνται όλα τα αρχεία JavaScript καθώς και οι HTML templates. Συγκεκριμένα οι φάκελοι `Collections`, `Models`, `Routers`, `Views` αντιστοιχούν στα ομώνυμα συστατικά του Backbone, ενώ στον φάκελο `libs` βρίσκονται οι διάφορες βιβλιοθήκες JavaScript.



Εικόνα 3.14: Κωδικοποίηση αρχείων

### 3.3 Βάση Δεδομένων

Η PostgreSQL έχει εδραιωθεί ως ένα ισχυρό εργαλείο ανοιχτού κώδικα που παρέχει την δυνατότητα διαμόρφωσης σχεσιακών βάσεων δεδομένων. Εφόσον η παρούσα εφαρμογή επιτάσσει την επεξεργασία και ανάλυση χωρικών δεδομένων, αξιοποιούνται και τα εργαλεία της PostGIS (επέκταση της PostgreSQL). Στο παρών κεφάλαιο περιγράφεται ο σχεδιασμός, οι οντότητες και οι συσχετίσεις μεταξύ αυτών τόσο για την βάση δεδομένων που δημιουργήθηκε στον server όσο και για αυτήν στην πλευρά του πελάτη.

#### 3.3.1 Σχεδιασμός βάσης Δεδομένων

##### *Τα Διαθέσιμα δεδομένα*

Η εγκυρότητα και αποτελεσματικότητα ενός συστήματος εξαρτάται από τα διαθέσιμα δεδομένα. Μια από τις δυσκολίες που παρουσιάζει η υλοποίηση μιας εφαρμογής αυτής της κατηγορίας, είναι η συγκέντρωση και οργάνωση μεγάλου όγκου δεδομένων. Η χρήση προτύπων για την διάθεση δεδομένων σχετικών με την διάρθρωση του δικτύου από τους οργανισμούς μετακινήσεων, απλοποιούν αρκετά τις διαδικασίες. Στην περίπτωση που εξετάστηκε, τα δεδομένα είναι διαθέσιμα σε διάφορες μορφές και όχι απαραίτητα συνδεδεμένα μεταξύ τους, αφού εξυπηρετούν διαφορετικούς σκοπούς μέσα στο σύστημα.

Συγκεκριμένα, οι πληροφορίες που απαιτούνται για την υλοποίηση των στατικών λειτουργιών του συστήματος, αφορούν κυρίως τις *Διαδρομές*, τις *Στάσεις* (και τις σημειακές οντότητες αυτών) και τα *Δρομολόγια* του δικτύου. Για τον προσδιορισμό των δυναμικών στοιχείων είναι απαραίτητες πληροφορίες σχετικά με τις *Βάρδιες* των λεωφορείων και η παρακολούθηση της *Τοποθεσίας* αυτών.

Τα δεδομένα που χρησιμοποιούνται στην εφαρμογή αποτελούν ένα μέρος των γραμμών και στάσεων όπου τα στοιχεία τους ήταν τα ελάχιστα απαραίτητα για την χρήση τους στο σύστημα. Οι στάσεις είναι γενικευμένες σε μία ανά οικισμό, ενώ χρησιμοποιούνται οι γραμμές που ήταν διαθέσιμη η αλληλουχία των στάσεων τους. Το πρόγραμμα αναφέρεται σε διαδρομές, ενώ υπάρχει μονάχα η χρονική ένδειξη της έναρξης αυτών.

Η δομή των διαθέσιμων δεδομένων καθώς και η έλλειψη/ αδυναμία προσδιορισμού αρκετών σημαντικών δεδομένων δεν επέτρεπε την δημιουργία και χρήση του πρότυπου GTFS, όπως αυτό αναλύθηκε στην Παράγραφο 2.6.1. Ωστόσο, ο σχεδιασμός της βάσης έχει δανειστεί αρκετά στοιχεία από το παραπάνω πρότυπο σχετικά με την δομή, τις συσχετίσεις και την ονοματολογία των πινάκων και των στοιχείων αυτού έτσι ώστε να είναι δυνατή η μεταφορά της εφαρμογής και σε άλλα συστήματα.

Το λογικό μοντέλο που χρησιμοποιείται από την PostgreSQL είναι το σχεσιακό μοντέλο, το οποίο χρησιμοποιεί πίνακες για την αναπαράσταση των οντοτήτων και κάθε πίνακας απο-

τελείται από ένα σύνολο γραμμών και στηλών. Η κάθε γραμμή αναπαριστά μία εγγραφή δεδομένων και ονομάζεται πλειάδα ενώ οι στήλες ορίζουν τα χαρακτηριστικά ή ιδιότητες της κάθε εγγραφής.

Για κάθε σχέση 1-N, επιλέχθηκε η οντότητα από την πλευρά N, και προστίθεται μια στήλη στη σχέση της, η οποία είναι ξένο κλειδί (FK) προς το πρωτεύον της οντότητας που συμμετέχει με βαθμό πληθικότητας 1 στη σχέση. Για τις σχέσεις M-N, δημιουργήθηκε μια νέα σχέση (πίνακας) και ορίζεται ως πρωτεύον κλειδί, τα επιμέρους πρωτεύοντα κλειδιά των σχέσεων που συμμετέχουν στη σχέση.

### *Χωρικά Δεδομένα*

Στους πίνακες με σημειακή χωρική πληροφορία υπάρχουν τόσο τα πεδία του γεωγραφικού μήκους και πλάτους όσο και το πεδίο με την γεωμετρία της οντότητας. Για τις χωρικές επεξεργασίες γίνεται χρήση όπως είναι φυσικό της γεωμετρίας. Στην αίτηση του χρήστη για στάσεις που αφορούν την δημιουργία χάρτη, επιστρέφονται τα στοιχεία γεωγραφικού μήκους και πλάτους, καθώς κάθε στάση αποτελεί ένα Model του Backbonejs που δεν υποστηρίζει στοιχεία γεωμετρίας.

Το PostGIS υποστηρίζει ένα πλήθος Γεωγραφικών και Προβολικών συστημάτων συντεταγμένων κάθε ένα από τα οποία περιγράφεται από ένα SRID (Spatial Reference Identifier). Στην περίπτωση των πινάκων που δημιουργήθηκαν έχει επιλεγεί το WGS84 όπου είναι το παγκόσμιο σύστημα που χρησιμοποιεί το ελλειψοειδές WGS84 (σύστημα του GPS), σε αυτό αναφέρονται οι όροι «γεωγραφικό μήκος» και «γεωγραφικό πλάτος», ενώ οι μετρήσεις έχουν αποτελέσματα σε μοίρες. Για την μετατροπή τους σε μέτρα χρησιμοποιούνται ανάλογες συναρτήσεις του PostGIS. Το Mercator Spheric είναι μία προβολή επί μίας σφαίρας με περίπου τα ίδια χαρακτηριστικά με το WGS84 (χρησιμοποιεί μία σφαίρα αντί για ελλειψοειδές) και είναι το σύστημα προβολής που χρησιμοποιεί ο μηχανισμός των tiles για τα GoogleMaps/ BingMaps/ OpenStreetMaps.

### *3.3.2 Βάση δεδομένων στην πλευρά του εξυπηρετητή*

Σύμφωνα λοιπόν με τα διαθέσιμα δεδομένα και τις απαιτήσεις του συστήματος ορίζονται οι παρακάτω τύποι οντοτήτων οι οποίες διαχωρίζονται σε αυτές που αφορούν την δομή και λειτουργία του δικτύου, σε αυτές που αφορούν τα δυναμικά στοιχεία του δικτύου και σε αυτές που χρησιμοποιούνται για την διαδικασία αντίστροφης γεωκωδικοποίησης.

### *Οντότητες σχετικές με το δίκτυο*

Εδώ περιέχονται οι πίνακες που αφορούν την διάρθρωση και λειτουργία του δικτύου. Έχει γίνει προσπάθεια οι πίνακες να ακολουθούν την τυποποιημένη μορφή του πρότυπου GTFS ώστε να είναι δυνατή η τροφοδοσία του συστήματος και με δεδομένα από άλλα δίκτυα με ελάχιστες προσαρμογές.

Οντότητα	Περιγραφή
routes	Οι γραμμές που εξυπηρετεί το δίκτυο
trips	Οι επιμέρους διαδρομές των γραμμών
stops	Περιλαμβάνει τις σημειακές οντότητες των στάσεων του δικτύου
trip_stops	Οι σειρά στάσεων της εκάστοτε διαδρομής*
trip_departures	Περιλαμβάνει το πρόγραμμα εκκίνησης των διαδρομών από την αφετηρία
agencies	Περιλαμβάνει στοιχεία για τον φορέα

Πίνακας 3.1: Οντότητες σχετικές με το δίκτυο

\* Είναι αντίστοιχος με τον πίνακα *stop\_times* του Πρότυπου *GTFS* χωρίς όμως να περιέχει πληροφορία για τον χρόνο αφίξεων-αναχωρήσεων

Αναλυτικά τα χαρακτηριστικά της κάθε οντότητας περιλαμβάνουν:

#### **agencies**

- *agency\_id(PK)* : Αναγνωριστικό του φορέα που εκτελεί την γραμμή
- *agency\_name* : Ονομασία Φορέα
- *agency\_url* : Διεύθυνση ιστοσελίδας φορέα
- *agency\_phone* : Τηλεφωνικός αριθμός φορέα

#### **stops**

- *stop\_id(PK)* : Αναγνωριστικό της στάσης
- *stop\_code* : Κωδικός στάσης
- *stop\_name* : Όνομα στάσης
- *stop\_county* : Ο νομός που ανήκει η στάση
- *location\_type* : Ένδειξη για τον αν η στάση αποτελεί σταθμό ή απλή στάση
  - 0 : Απλή στάση
  - 1 : Σταθμός
- *stop\_lat* : Γεωγραφικό πλάτος στάσης
- *stop\_lon* : Γεωγραφικό μήκος στάσης
- *parent\_station* : Η στάση ή ο σταθμός που ανήκει συγκεκριμένη στάση
  - *stop\_id*
- *geom* : Σημειακή οντότητα στάσης

#### **routes**

- *route\_id(PK)* : Αναγνωριστικό της γραμμής
- *route\_short\_name* : Σύντομο όνομα γραμμής
- *route\_long\_name* : Όνομα γραμμής

- route\_type : Τύπος γραμμής
  - 0 : Κυκλική διαδρομή
  - 1 : Διαδρομή διπλής κατεύθυνσης
- agency\_id(FK) : Αναγνωριστικό του φορέα που εκτελεί την γραμμή

### trips

- trip\_id(PK) : Αναγνωριστικό της διαδρομής
- trip\_headsign : Όνομα που είναι γνωστή η διαδρομή
- route\_id(FK) : Αναγνωριστικό της γραμμής που ανήκει η διαδρομή
- direction : Προορισμός διαδρομής
  - 0 : Κίνηση από τον σταθμό αφετηρίας
  - 1 : Κίνηση προς τον σταθμό αφετηρίας

### trip\_stops

- trip\_id (PK)(FK) : Αναγνωριστικό της διαδρομής
- stop\_id (PK) : Αναγνωριστικό της στάσης
- stop\_sequence (PK) : Αύξων αριθμός της στάσης στη συγκεκριμένη διαδρομή

### trip\_departures

- service\_id (PK) : Αναγνωριστικό της υπηρεσίας
- trip\_id (FK) : Αναγνωριστικό της διαδρομής
- trip\_start (FK) : Η στάση εκκίνησης
- time : Η ώρα εκκίνησης
- day : Η ημέρα πραγματοποίησης της υπηρεσίας

### Οντότητες σχετικές με την λειτουργία σε πραγματικό χρόνο

Εδώ περιέχονται οι πίνακες που προσομοιάζουν την πραγματική λειτουργία του δικτύου και είτε μεταβάλλονται σε πραγματικό χρόνο είτε χρησιμοποιούνται για τη σύνδεση μεταξύ των δεδομένων από τα επιμέρους συστήματα. Τα δεδομένα σχετικά με τις υπηρεσίες αφορούν μία ενδεικτική μέρα λειτουργίας, ενώ η τοποθεσία των λεωφορείων μεταβάλλεται με χρήση αντίστοιχου API στον server.

Οντότητα	Περιγραφή
buses	Περιλαμβάνονται τα στοιχεία που αφορούν την θέση των λεωφορείων του στόλου σε πραγματικό χρόνο.
shifts	Περιέχει τις προγραμματισμένες υπηρεσίες και τα αντίστοιχα λεωφορεία που τις εκτελούν για δεδομένη ημερομηνία.
itineraries	Περιέχει τα προγραμματισμένα δρομολόγια των λεωφορείων.

Πίνακας 3.2: Οντότητες σχετικές με την λειτουργία σε πραγματικό χρόνο

Αναλυτικά:

### buses

- bus\_id (PK) : Αναγνωριστικό του λεωφορείου που ταυτίζεται και με τον αριθμό του λεωφορείου
- bus\_lat : Το γεωγραφικό πλάτος της τοποθεσίας του λεωφορείου
- bus\_lon : Το γεωγραφικό μήκος της τοποθεσίας του λεωφορείου
- bus\_timestamp : Η χρονική στιγμή προσδιορισμού της θέσης του λεωφορείου
- geom : Η σημειακή οντότητα του λεωφορείου

### shifts

- shift\_id (PK) : Αναγνωριστικό της υπηρεσίας
- date (PK) : Ημερομηνία που αντιστοιχεί η υπηρεσία
- bus\_id (FK) : Αριθμός λεωφορείου που εκτελεί την υπηρεσία
- shift\_long\_name : Πλήρες όνομα υπηρεσίας

### itineraries

- itinerary\_id (PK) : Αναγνωριστικό του δρομολογίου
- itinerary\_name : Ονομασία δρομολογίου
- itinerary\_start\_time : Ώρα έναρξης δρομολογίου
- itinerary\_finish\_time : Ώρα πέραςτος δρομολογίου

### Οντότητες σχετικές με την λειτουργία αντίστροφης γεωκωδικοποίησης

Εδώ περιλαμβάνονται οι πίνακες που χρησιμοποιούνται στην διαδικασία της αντίστροφης γεωκωδικοποίησης (reverse geocoding) της τοποθεσίας του χρήστη.

Οντότητα	Περιγραφή
counties	Περιλαμβάνει οι πολυγωνικές οντότητες των νομών της χώρας.
towns	Περιλαμβάνει τις σημειακές οντότητες των οικισμών της χώρας.

Πίνακας 3.3: Οντότητες σχετικές με την λειτουργία αντίστροφης γεωκωδικοποίησης

### counties

- county\_id (PK) : Το αναγνωριστικό του νομού
- county\_name\_gr : Το όνομα του νομού στα ελληνικά
- geom : Το πεδίο της γεωμετρίας του νομού

### towns

- town\_id (PK) : Το αναγνωριστικό της πόλης
- town\_name\_gr : Το όνομα του δήμου στα ελληνικά
- geom : Το πεδίο της γεωμετρίας της πόλης

## Συσχετίσεις

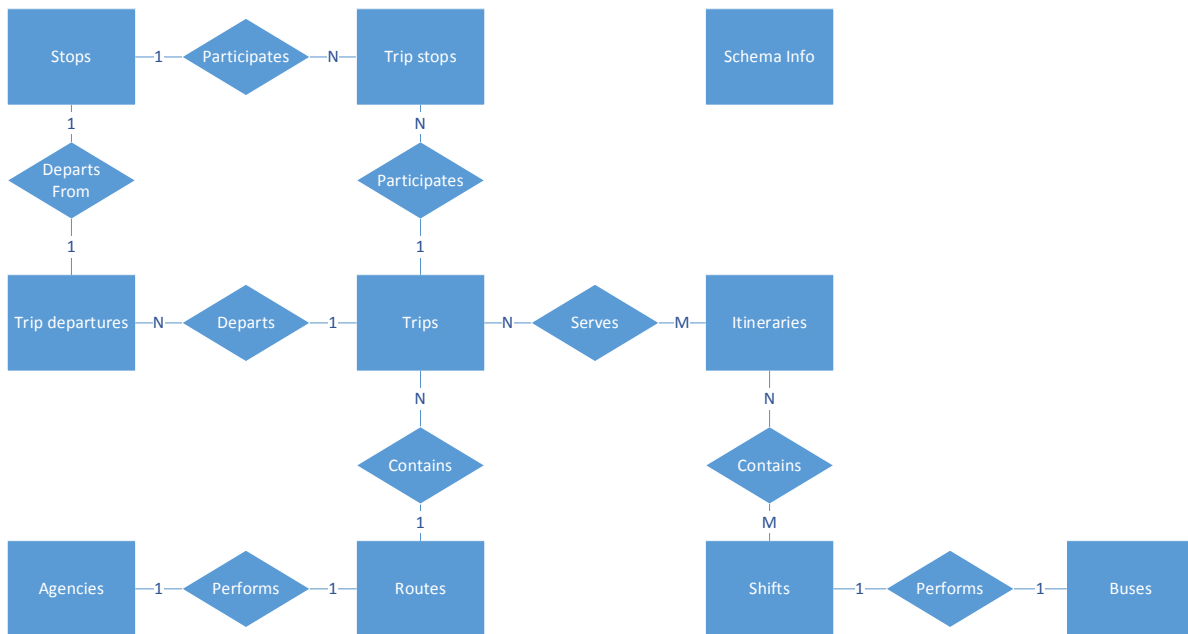
Επίσης, προκύπτουν οι ακόλουθοι τύποι συσχετίσεων (Πίνακας 3.4). Οι συσχετίσεις στο σύνολο τους είναι 2<sup>ου</sup> βαθμού, καθώς αναφέρονται πάντα μεταξύ δύο οντοτήτων.

Συσχέτιση	Περιγραφή
performs	Συνδέει τις οντότητες <i>agencies</i> και <i>routes</i> . Η συσχέτιση έχει λόγο πληθικότητας 1:N καθώς ένας φορέας εκτελεί μία ή περισσότερες γραμμές ενώ μια γραμμή εκτελείται από ένα φορέα.
contains	Συνδέει τις οντότητες <i>trips</i> και <i>routes</i> . Η συσχέτιση έχει λόγο πληθικότητας 1:N, καθώς μια γραμμή μπορεί να περιλαμβάνει μία ή περισσότερες διαδρομές ενώ μία διαδρομή ανήκει σε μία γραμμή.
participates	Συνδέει τις οντότητες των <i>trip_stops</i> και <i>trips</i> . Η συσχέτιση έχει λόγο πληθικότητας 1:N. Η στάση μίας διαδρομής συμμετέχει σε μία διαδρομή, ενώ σε μία διαδρομή συμμετέχουν τουλάχιστον δύο στάσεις.
participates	Αντίστοιχα συνδέει τις οντότητες των <i>stops</i> και <i>trip_stops</i> με λόγο πληθικότητας 1:N. Μία στάση συμμετέχει σε μία ή περισσότερες διαδρομές.
departs	Συνδέει τις οντότητες <i>trips</i> και <i>trip_departures</i> . Η συσχέτιση έχει λόγο πληθικότητας 1:N, καθώς μία διαδρομή έχει μια ή παραπάνω αναχωρήσεις.
departsFrom	Συνδέει τις οντότητες <i>trip_departures</i> και <i>stops</i> . Η συσχέτιση έχει λόγο πληθικότητας 1:1, καθώς η αναχώρηση μίας διαδρομής πραγματοποιείται από μία στάση.
contains	Για τη σύνδεση των οντοτήτων <i>shifts</i> και <i>itineraries</i> , διαμορφώθηκε ένας νέος πίνακας, ο οποίος αντιπροσωπεύει τη σχέση με λόγο πληθικότητας M-N μεταξύ αυτών. Μια υπηρεσία περιέχει ένα ή περισσότερα δρομολόγια, ενώ ένα δρομολόγιο μπορεί να εκτελείται από περισσότερες από μια υπηρεσίες.
serves	Για τη σύνδεση των οντοτήτων <i>itineraries</i> και <i>trips</i> , διαμορφώθηκε ένας νέος πίνακας, ο οποίος αντιπροσωπεύει τη σχέση με λόγο πληθικότητας M-N μεταξύ αυτών. Ένα δρομολόγιο στο πρόγραμμα υπηρεσιών εξυπηρετεί μία ή περισσότερες διαδρομές, ενώ από την άλλη μια διαδρομή εξυπηρετείται από ένα ή περισσότερα δρομολόγια.
performs	Συνδέει τις οντότητες <i>buses</i> και <i>shifts</i> . Η συσχέτιση έχει λόγο πληθικότητας 1:1, καθώς ένα λεωφορείο εκτελεί για δεδομένη ημερομηνία μία υπηρεσία και αντίστροφα.

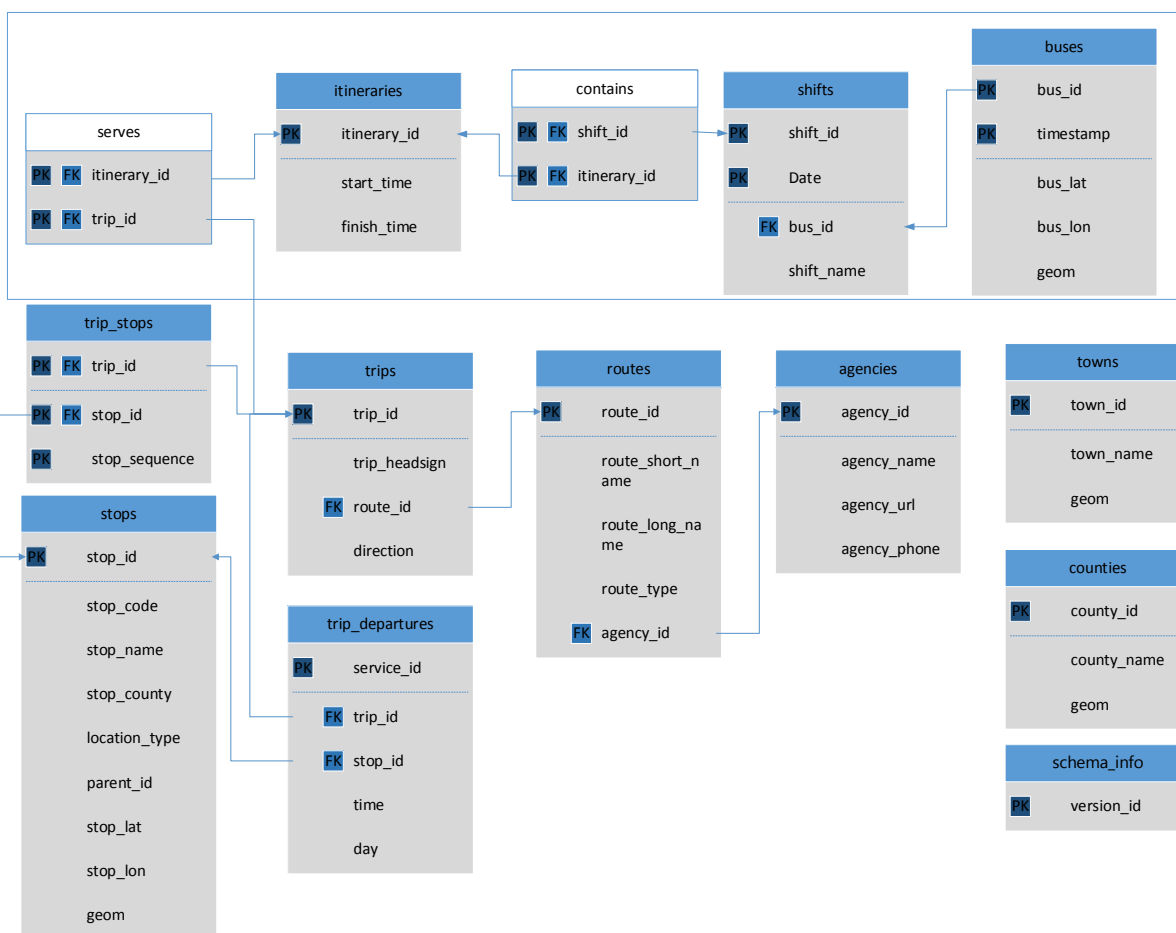
Πίνακας 3.4: Πίνακας Συσχετίσεων

Σύμφωνα με τα παραπάνω δημιουργείται το διάγραμμα Οντοτήτων – Συσχετίσεων που φαίνεται στο Διάγραμμα 3.5 και το Σχεσιακό Μοντέλο Διάγραμμα 3.6.





Διάγραμμα 3.5: E-R Model βάσης δεδομένων



Διάγραμμα 3.6: Σχισιακό Μοντέλο βάσης δεδομένων

### 3.3.4 Βάση δεδομένων στην πλευρά του πελάτη

Πέρα από την βάση δεδομένων στην πλευρά του εξυπηρετητή, υπάρχει και μία δεύτερη βάση δεδομένων στην πλευρά του πελάτη. Το σύστημα που χρησιμοποιείται είναι μια σχεσιακή βάση δεδομένων SQLite. Η αποθήκευση δεδομένων τοπικά εξασφαλίζει την λειτουργία της εφαρμογής και εκτός σύνδεσης ενώ εκμηδενίζει τον χρόνο που απαιτείται για την φόρτωση δεδομένων από τον server. Αναλυτικότερα, η τοπική βάση απαρτίζεται από πίνακες σχετικούς με την διάρθρωση του δικτύου, ενώ περιέχεται και ο πίνακας με τις ώρες των δρομολογίων. Ο τελευταίος πίνακας παρουσιάζει περισσότερες μεταβολές σε σχέση με τους υπόλοιπους, αφού διαμορφώνεται ανάλογα με τις απαιτήσεις της κάθε περιόδου. Η έκδοση των στοιχείων που είναι αποθηκευμένα παρακολουθείται από αντίστοιχο πίνακα και σε ενδεχόμενες μεταβολές ανανεώνονται.

Κατά την έναρξη της εφαρμογής πραγματοποιείται μεταφορά και δημιουργία των παρακάτω πινάκων στην βάση δεδομένων της εφαρμογής:

- **stops**
- **routes**
- **trips**
- **trip\_stops**
- **trip\_departures**

Πέραν των ανωτέρω, στην τοπική βάση δημιουργούνται δύο επιπλέον πίνακες:

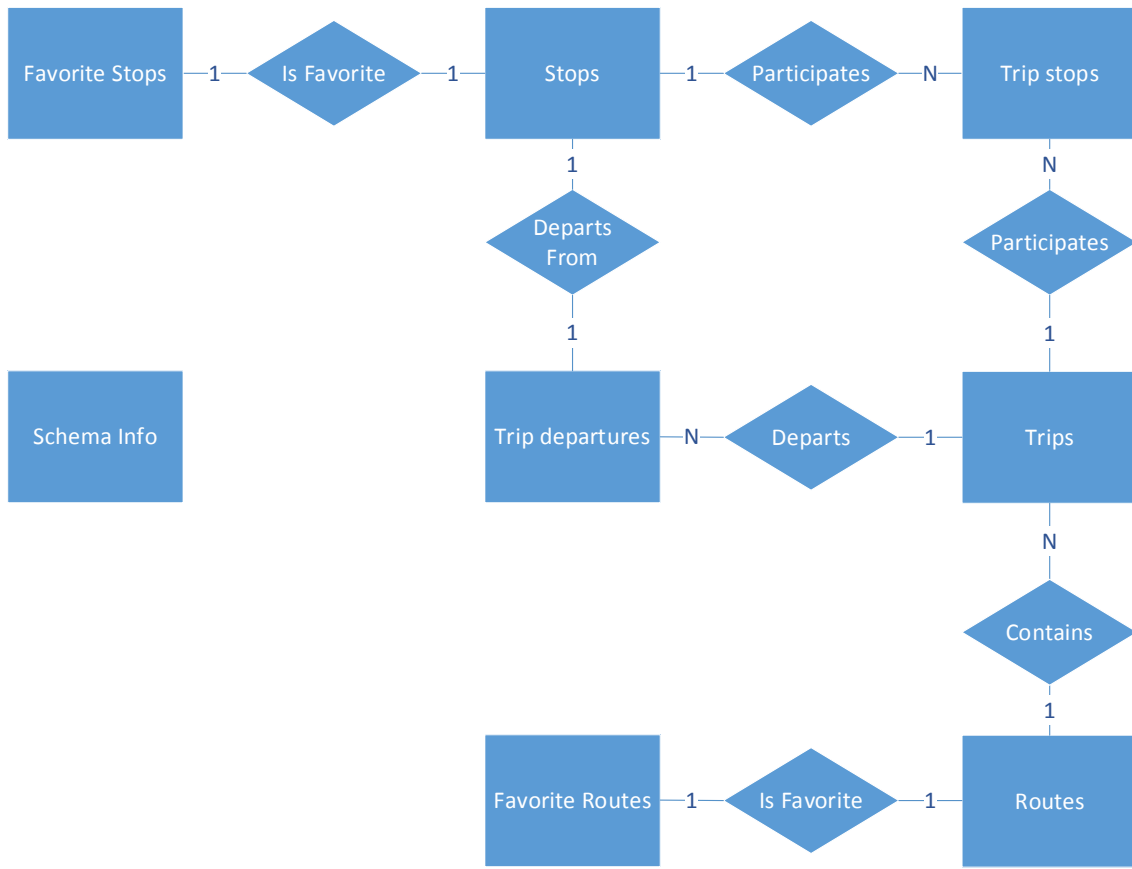
- **schema\_info** Ο πίνακας που περιλαμβάνει ένα μόνο χαρακτηριστικό σχετικά με την έκδοση των δεδομένων
- **version(PK)** : Έκδοση του σχήματος που βρίσκεται στη συσκευή
- **favoritesRoutes** Ο πίνακας με τις αγαπημένες διαδρομές.
- **route\_id(PK)** : Αναγνωριστικό αγαπημένου
- **favoritesStops** Ο πίνακας με τις αγαπημένες στάσεις.
- **stop\_id(PK)** : Αναγνωριστικό αγαπημένου

Επιπλέον, προκύπτουν οι ακόλουθοι τύποι συσχετίσεων 2<sup>ου</sup> βαθμού (Πίνακας 3.5):

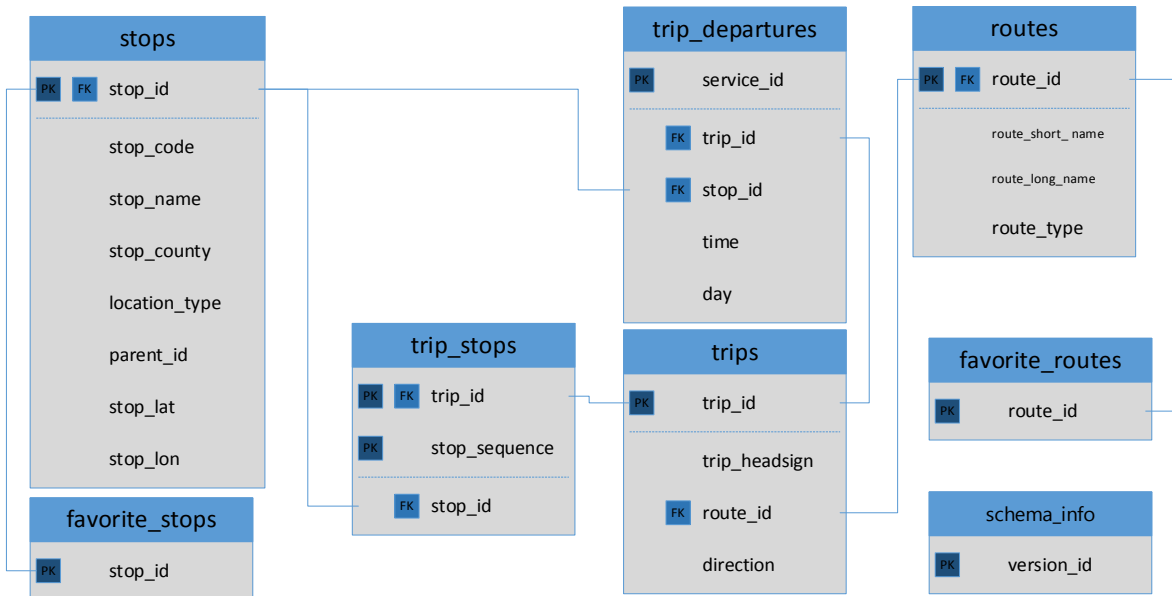
Συσχέτιση	Περιγραφή
is_favorite	Μεταξύ των τύπων οντοτήτων <i>Routes</i> και <i>Favorites Routes</i> με λόγο πληθικότητας 1:1.
is_favorite	Μεταξύ των τύπων οντοτήτων <i>Stops</i> και <i>Favorites Stops</i> με λόγο πληθικότητας 1:1.

Πίνακας 3.5: Συσχετίσεις βάσης δεδομένων στην πλευρά του πελάτη

Στα Διαγράμματα 3.7 και 3.8 φαίνονται το Μοντέλο Οντοτήτων-Συσχετίσεων και το Σχεσιακό Μοντέλο αντίστοιχα.



Διάγραμμα 3.7: E-R model βάσης δεδομένων στην πλευρά του πελάτη



Διάγραμμα 3.8: Σχηματικό Μοντέλο βάσης στην πλευρά του πελάτη

### 3.4 Διαδικασία Λειτουργίας

Όπως παρουσιάστηκε στην περιγραφή της αρχιτεκτονικής του συστήματος, για την λειτουργία της εφαρμογής απαιτείται η επικοινωνία αυτής με κάποια βάση δεδομένων, που είτε βρίσκεται στον τοπικό αποθηκευτικό χώρο της συσκευής, είτε σε κάποιον απομακρυσμένο εξυπηρετητή. Στην παράγραφο που ακολουθεί αναλύεται η επικοινωνία και ανταλλαγή δεδομένων μεταξύ εφαρμογής και των δύο αυτών βάσεων.

#### 3.4.1 Επικοινωνία με την βάση δεδομένων στην πλευρά του πελάτη

Για τη διάθεση μιας σειράς από βασικές πληροφορίες και εκτός σύνδεσης στο διαδίκτυο στον αποθηκευτικό χώρο της συσκευής δημιουργείται και ανανεώνεται μια σχεσιακή βάση δεδομένων (Παράγραφος 3.3.4). Τα ερωτήματα που υποβάλλονται, απαιτούν απλό συνδυασμό των τιμών στους διάφορους πίνακες χωρίς αυξημένους υπολογιστικούς πόρους. Στη συνέχεια, περιγράφεται η πραγματοποίηση των ερωτημάτων αυτών.

##### *Εύρεση γραμμών που διέρχονται από κάποια στάση*

Το συγκεκριμένο ερώτημα επιστρέφει τις γραμμές με παράμετρο κάποια στάση ενδιαφέροντος. Η απάντηση του ερωτήματος βρίσκεται στον πίνακα των στάσεων της διαδρομής *trip\_stops*. Η απάντηση συνοδεύεται και με την κατεύθυνση που διέρχεται η γραμμή από τη συγκεκριμένη στάση, αν δηλαδή διέρχεται κατά την μετάβαση, την επιστροφή ή και στις δύο περιπτώσεις.

##### Παράμετροι ερωτήματος

- **stop\_id** Αναγνωριστικό της στάσης

##### Απάντηση

- **route\_id** Αναγνωριστικό της γραμμής
- **route\_long\_name** Το πλήρες όνομα της γραμμής
- **direction\_inbound** Αν η γραμμή περνάει κατά την μετάβαση
- **direction\_outbound** Αν η γραμμή περνάει κατά την επιστροφή

##### *Εύρεση στάσεων που ανήκουν σε μια διαδρομή*

Το ερώτημα αυτό είναι το αντίστροφο του ανωτέρω με την διαφορά ότι η απάντηση συνοδεύεται και από τον κωδικό που εκφράζει τον αύξοντα αριθμό που διέρχεται η διαδρομή από τη συγκεκριμένη στάση για την ταξινόμηση των αποτελεσμάτων (Κώδικας 3.10). Για την παρουσίαση των αποτελεσμάτων υπάρχει διαχωρισμός μεταξύ μετάβασης και επιστροφής.

##### Παράμετροι ερωτήματος

- **route\_id** Αναγνωριστικό της γραμμής
- **direction** Κατεύθυνση γραμμής

## Απάντηση

- **stop\_id** Αναγνωριστικό της στάσης
- **stop\_name** Το όνομα της στάσης
- **stop\_county** Ο νομός που βρίσκεται η στάση
- **stop\_sequence** Ο αύξων αριθμός της στάσης στην διαδρομή

```
findStops: function (route_id, direction) {
    this.db = window.sqlitePlugin.openDatabase({name: "kteIDB"});
    this.dfdStops = $.Deferred();
    this.stops= [];
    var self=this;
    this.db.transaction(
        function (tx) {
            var sql =
                "SELECT S.stop_id, S.stop_name, S.stop_county, " +
                "S.stop_sequence " +
                "FROM stops AS S " +
                "INNER JOIN trip_stops TS ON TS.stop_id = S.stop_id " +
                "INNER JOIN trips T ON T.trip_id = TS.trip_id " +
                "INNER JOIN routes R ON R.route_id = T.route_id " +
                "WHERE R.route_id='"+ route_id+"' " +
                "AND T.trip_direction='"+ direction+"' " +
                "ORDER BY TS.stop_sequence";
            tx.executeSql(sql, this.txErrorHandler,
                function (tx, results) {
                    var len = results.rows.length;
                    var i = 0;
                    for (; i < len; i = i + 1) {
                        self.stops.push(results.rows.item(i));
                    }
                    self.dfdStops.resolve(self.stops);
                }
            );
        }
    );
    return this.dfdStops.promise(this.stops);
},
txErrorHandler: function (tx) {
    alert(tx.message);
    this.dfdStops.reject("Error", tx.message);
}
```

Κώδικας 3.10: Υποβολή ερωτήματος σε τοπική βάση δεδομένων

## Δρομολόγια γραμμών

Στον πίνακα *trip\_departures* διατηρούνται τα στοιχεία για την ώρα αναχώρησης των λεωφορείων από τον σταθμό εκκίνησης. Η αναζήτηση σε αυτόν γίνεται με παράμετρο την

γραμμή, την κατεύθυνση και την ημέρα πραγματοποίησης του δρομολογίου ενώ τα αποτελέσματα περιέχουν την ώρα αναχώρησης για την μετάβαση και επιστροφή από την αφετηρία.

#### Παράμετροι ερωτήματος

- **route\_id** Αναγνωριστικό της γραμμής
- **direction** Κατεύθυνση γραμμής
- **day** Ημέρα πραγματοποίησης δρομολογίου

#### Απάντηση

- **start\_stop** Η στάση εκκίνησης
- **time** Οι ώρες εκκίνησης

### 3.4.2 Επικοινωνία με την βάση δεδομένων στην πλευρά του εξυπηρετητή

Στο προηγούμενο κεφάλαιο παρουσιάστηκε η διαδικασία λήψης δεδομένων από κάποιον εξυπηρετητή με χρήση της δυνατότητας `collection.fetch()` του `Backbone.js`. Στη συνέχεια, παρουσιάζονται οι αιτήσεις προς τον εξυπηρετητή. Κάθε αίτηση ενεργοποιεί το αντίστοιχο API στον εξυπηρετητή με μια σειρά από ερωτήματα στην βάση δεδομένων και το αποτέλεσμα αυτής επιστρέφεται ως μια απάντηση που ακολουθεί το πρότυπο `JSON`. Η απάντηση, τέλος, συμπληρώνει τα αντίστοιχα `Models`. Παρακάτω για κάθε αίτηση παρουσιάζονται:

- Σύντομη περιγραφή κάθε αίτησης,
- Οι παράμετροι κάθε αίτησης
- Η διαδικασία στον εξυπηρετητή
- Τι επιστρέφεται κάθε φορά από τον εξυπηρετητή

#### Κοντινές στάσεις

Εδώ ο χρήστης αναζητά τις πλησιέστερες σε αυτόν στάσεις σε ακτίνα 10km με παράμετρο το γεωγραφικό μήκος και πλάτος είτε της τοποθεσίας του είτε μιας προσδιορισμένης τοποθεσίας στον χάρτη.

#### Παράμετροι αίτησης

- **lat** Γεωγραφικό μήκος χρήστη
- **lon** Γεωγραφικό πλάτος χρήστη

#### Διαδικασία

Η απόσταση υπολογίζεται βάσει της συνάρτησης του `PostGIS ST_Distance_Spheroid` που υπολογίζει την απόσταση σε μέτρα μεταξύ των σημειακών οντοτήτων, των στάσεων και της τοποθεσίας του χρήστη. Η τιμή της αντιστοιχεί στην απευθείας απόσταση μεταξύ σημείου - στάσης και στην εφαρμογή εκφράζει ένα σχετικό μέτρο σύγκρισης για τον προσδιορισμό των πλησιέστερων στάσεων.

### Απάντηση

- **stop\_id** Αναγνωριστικό της στάσης
- **stop\_name** Η ονομασία της στάσης
- **stop\_county** Ο νομός που ανήκει η επιλεγμένη στάση
- **stop\_lat** Γεωγραφικό μήκος στάσης
- **stop\_lon** Γεωγραφικό πλάτος στάσης
- **location\_type** Ο τύπος της συγκεκριμένης στάσης
- **distance** Η απόσταση της αιτούμενης τοποθεσίας από τη στάση

### *Οργάνωση ταξιδιού*

Σε αυτή την περίπτωση αποστέλλεται στον εξυπηρετητή μια αίτηση με τις απαιτήσεις του χρήστη για την μετάβαση μεταξύ δύο σημείων. Τα σημεία εκκίνησης και προορισμού αποτελούν σημεία του δικτύου.

### Παράμετροι αίτησης

- **start** Ο κωδικός της στάσης εκκίνησης
- **destination** Ο κωδικός της στάσης τερματισμού

### Διαδικασία

Η λειτουργία στοχεύει στον συνδυασμό των διαδρομών του υφιστάμενου δικτύου για την μετακίνηση μεταξύ δύο στάσεων. Αρχικά, με δεδομένη τη στάση εκκίνησης εξετάζεται αν υπάρχει απευθείας σύνδεση με τη στάση προορισμού. Εάν κάτι τέτοιο είναι αληθές γίνεται έλεγχος της σειράς που η διαδρομή διέρχεται από αυτές τις στάσεις. Λόγω της διάρθρωσης του δικτύου η συντομότερη διαδρομή είναι σχεδόν πάντα η απευθείας σύνδεση.

Στην συνέχεια, εξετάζεται αν υπάρχει κοινό σημείο μεταξύ των γραμμών που διέρχονται απευθείας από τις στάσεις ενδιαφέροντος. Αν οι διερχόμενες γραμμές έχουν κάποια κοινή στάση, η στάση αυτή ορίζεται ως ενδιάμεση στάση και η διαδικασία επαναλαμβάνεται για τα δύο επιμέρους τμήματα. Η λογική αυτή μπορεί να επεκταθεί και με χρήση πληροφοριών σχετικά με τον χρόνο απευθείας μετάβασης μεταξύ των διαδοχικών στάσεων ή κάποια άλλη παράμετρο κόστους για την εύρεση της βέλτιστης διαδρομής.

### Απάντηση

- **option\_id** Ο αύξων αριθμός της εναλλακτικής
- **journey\_id** Το αναγνωριστικό της επιμέρους διαδρομής
- **start\_id** Το id της στάσης εκκίνησης της επιμέρους διαδρομής
- **start\_name** Η ονομασία της στάσης εκκίνησης της επιμέρους διαδρομής
- **destination\_id** Το id της στάσης προορισμού της επιμέρους διαδρομής
- **destination\_name** Το όνομα της στάσης προορισμού της επιμέρους διαδρομής

- **trip\_id** Το id της διαδρομής
- **trip\_headsign** Το όνομα της διαδρομής που συνδέει τις δύο στάσης
- **stops\_count** Ο αριθμός των ενδιάμεσων στάσεων

#### *Τοποθεσία λεωφορείων κάποιας γραμμής*

Μια ακόμα ενδιαφέρουσα λειτουργία της εφαρμογής είναι ο προσδιορισμός της θέσης των λεωφορείων κάποιας γραμμής σε πραγματικό χρόνο.

#### Παράμετροι αίτησης

- **route\_id** Αναγνωριστικό της γραμμής
- **direction** Κατεύθυνση διαδρομής (Μετάβαση ή Επιστροφή)

#### Διαδικασία

Για την απάντηση στο συγκεκριμένο ερώτημα απαιτείται ο συνδυασμός πινάκων που αλλάζουν δυναμικά. Αρχικά, εξετάζεται η συσχέτιση διαδρομών-δρομολογίων και έπειτα, με παράμετρο την ώρα εντοπίζονται οι υπηρεσίες και ποια λεωφορεία εκτελούν τη συγκεκριμένη διαδρομή την δεδομένη χρονική στιγμή. Έπειτα από τον πίνακα με τις θέσεις των λεωφορείων λαμβάνεται η πληροφορία για την θέση αυτών που ενδιαφέρουν τον χρήστη. Η απάντηση συνοδεύεται και με την πλησιέστερη στο λεωφορείο πόλη-χωριό για την καλύτερη ενημέρωση του χρήστη.

#### Απάντηση

- **bus\_id** Αναγνωριστικό του λεωφορείου
- **lat** Το γεωγραφικό μήκος της τοποθεσίας του λεωφορείου
- **lng** Το γεωγραφικό πλάτος της τοποθεσίας του λεωφορείου
- **route\_id** Αναγνωριστικό της γραμμής που εκτελεί
- **route\_name** Το όνομα της γραμμής που εκτελεί
- **near** Ο νομός που βρίσκεται το λεωφορείο
- **nearTown** Η περιοχή που βρίσκεται το λεωφορείο

#### *Λεωφορείο Ενδιαφέροντος*

Για την παρακολούθηση ενός λεωφορείου ενδιαφέροντος αρχικά πρέπει να επιλεγεί ο αριθμός του λεωφορείου. Για το σκοπό αυτό είναι διαθέσιμες αρκετές επιλογές λόγω του εύρους των πληροφοριών που είναι δυνατό να γνωρίζει ο επιβάτης.



### Παράμετροι αίτησης

- **type** Ο τύπος της αιτήσεως
  - bus\_id Ο αριθμός του λεωφορείου
  - location Αναζήτηση πλησιέστερου λεωφορείου
  - route\_id Αναζήτηση λεωφορείου που εκτελεί συγκεκριμένη γραμμή
- **id** Η παράμετρος της επιλογής

### Διαδικασία

Οποιαδήποτε από τις εναλλακτικές και αν επιλέξει ο χρήστης η απάντηση είναι κοινή και τα στοιχεία της αποτελούν τις παραμέτρους για την υποβολή της αίτησης παρακολούθησης της πορείας του λεωφορείου. Καθώς μόνο η εισαγωγή του αριθμού του λεωφορείου δίνει μοναδική λύση, ο χρήστης επιλέγει μέσω ενός PopUp παραθύρου το λεωφορείο ενδιαφέροντος με βάση την γραμμή που εκτελεί. Τα εναλλακτικά λεωφορεία της λύσης συνοδεύονται και από την περιοχή την οποία βρίσκονται έτσι ώστε να μπορεί να εκτιμήσει ο χρήστης την απόσταση που απέχει από αυτόν.

### Απάντηση - Επιλογής Λεωφορείου

- **bus\_id** Ο αριθμός του λεωφορείου
- **route\_name** Η γραμμή που εκτελεί
- **direction** Κατεύθυνση γραμμής (Μετάβαση ή Επιστροφή)
- **near** Ο νομός που βρίσκεται το λεωφορείο
- **nearTown** Η περιοχή που βρίσκεται το λεωφορείο

### Απάντηση - Παρακολούθησης Λεωφορείου

- **bus\_id** Ο αριθμός του λεωφορείου
- **bus\_lat** Γεωγραφικό μήκος τοποθεσίας λεωφορείου
- **bus\_lon** Γεωγραφικό πλάτος τοποθεσίας λεωφορείου
- **route\_name** Η γραμμή που εκτελεί
- **direction** Κατεύθυνση γραμμής (Μετάβαση ή Επιστροφή)
- **near** Ο νομός που βρίσκεται το λεωφορείο
- **nearTown** Η περιοχή που βρίσκεται το λεωφορείο
- **updated** Ο χρόνος από τη στιγμή που ανανεώθηκε η τοποθεσία αυτού

### Διαδικασία Αντίστροφης Γεωκωδικοποίησης

Η εφαρμογή σε ορισμένες λειτουργίες κάνει χρήση της τοποθεσίας του χρήστη. Για την επαλήθευση και την παρουσίαση αποτελεσμάτων προηγείται η διαδικασία αντίστροφης γεωκωδικοποίησης της τοποθεσίας σε επίπεδο πόλης και νομού.

### Παράμετροι αίτησης

- **lat** Γεωγραφικό μήκος χρήστη
- **lon** Γεωγραφικό πλάτος χρήστη

### Διαδικασία

Με χρήση χωρικών λειτουργιών του PostGIS προσδιορίζεται η πολυγωνική οντότητα του νομού που βρίσκεται ο χρήστης καθώς και η πλησιέστερη σε αυτόν (εντός ορίων) σημειακή οντότητα πόλης.

### Απάντηση

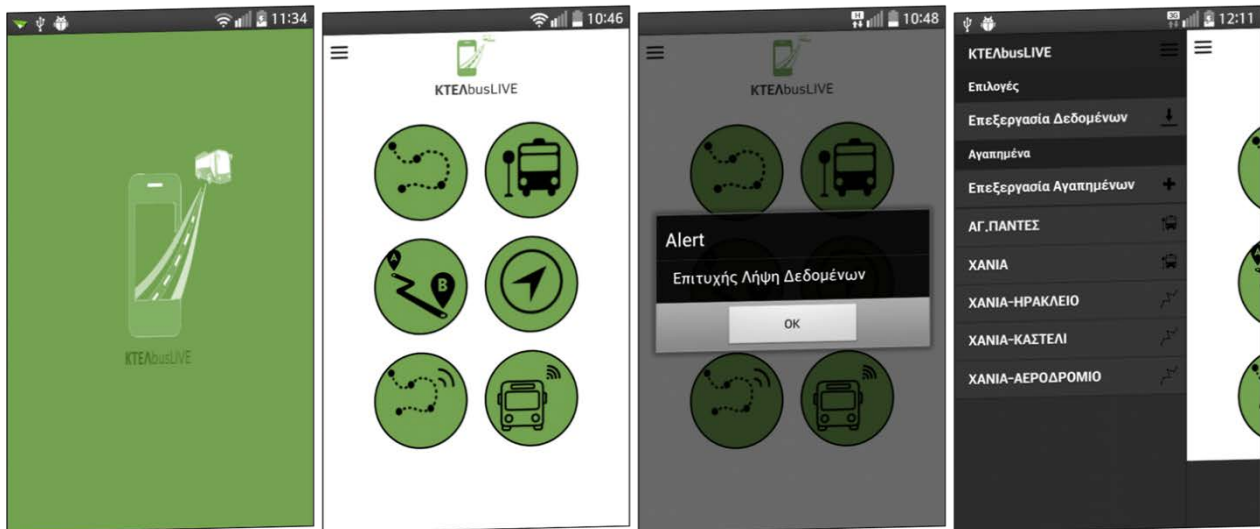
- **county** Το όνομα του νομού
- **town** Το όνομα της πόλης

### 3.5 Υλοποιήσεις

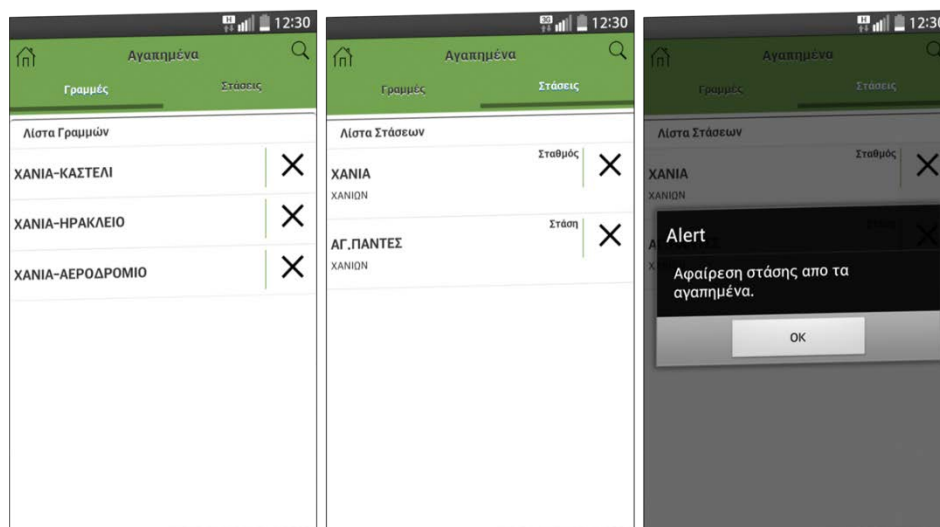
Στην τελευταία παράγραφο της ενότητας γίνεται περιγραφή και εφαρμογή όλων των διαθέσιμων σεναρίων χρήσης του συστήματος, παρουσιάζοντας αντιπροσωπευτικά στιγμιότυπα της κάθε λειτουργίας.

#### 3.5.1 Αρχική οθόνη

Κατά την έναρξη, αν η εφαρμογή δεν έχει ξαναλειτουργήσει στη συγκεκριμένη συσκευή εμφανίζεται αντίστοιχο μήνυμα και γίνεται λήψη των απαιτούμενων δεδομένων. Έπειτα, ο χρήστης μεταβαίνει στην οθόνη με τις επιλογές της εφαρμογής (Εικόνα 3.15). Μια πλευρική εργαλειοθήκη ενεργοποιείται στην αριστερή πλευρά της οθόνης με μια λίστα των αγαπημένων στάσεων και διαδρομών. Μέσα από την λίστα αγαπημένων με την επιλογή γραμμής γίνεται μετάβαση στη σελίδα ζωντανής παρακολούθησης των γραμμών, ενώ με την επιλογή στάσης στην προβολή των διερχόμενων από αυτή γραμμών. Για την διαμόρφωση των αγαπημένων υπάρχει αντίστοιχη οθόνη (Εικόνα 3.16).



Εικόνα 3.15: Στιγμιότυπα αρχικής οθόνης



Εικόνα 3.16: Στιγμιότυπα διαχείρισης αγαπημένων

### 3.5.2 Γραμμές δικτύου

Η λειτουργία αυτή δίνει την δυνατότητα για (Εικόνες 3.17 – 3.19):

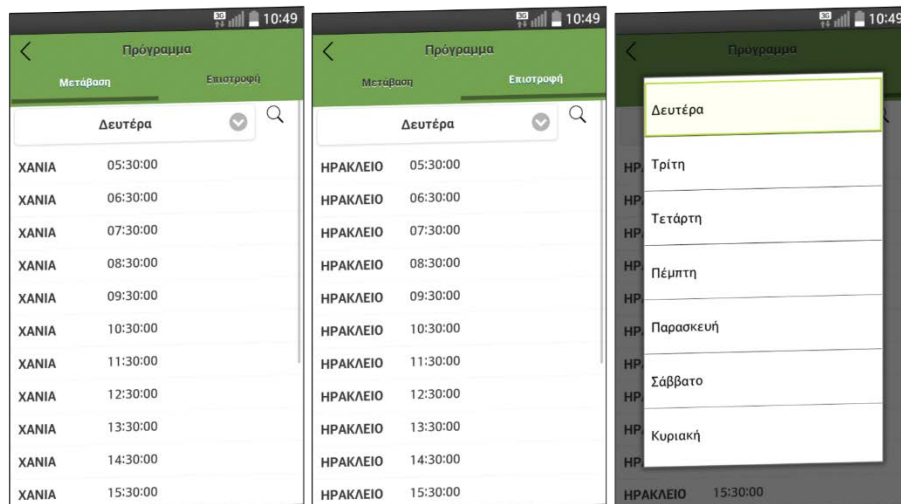
- Αναζήτηση των διαθέσιμων γραμμών
- Προβολή των στάσεων για τις επιμέρους διαδρομές της γραμμής (Μετάβαση-Επιστροφή)
- Προβολή του χάρτη της γραμμής
- Πρόγραμμα εκτέλεσης δρομολογίων
- Προβολή λεωφορείων που εκτελούν την γραμμή σε πραγματικό χρόνο στον χάρτη

Ο χρήστης επιλέγει την γραμμή ενδιαφέροντος μέσα από μία λίστα με δυνατότητα αναζήτησης εισάγοντας μέρος του ονόματος της γραμμής. Οι κυκλικές διαδρομές εμφανίζουν αντίστοιχο σύμβολο για τον διαχωρισμό τους από τις γραμμές διπλής κατεύθυνσης. Με την επιλογή του ονόματος της γραμμής (Εικόνα 3.17-α) μεταβαίνει στην Προβολή στάσεων της γραμμής χωρισμένων σε μετάβαση και επιστροφή, ενώ με επιλογή του σχήματος της γραμμής (Εικόνα 3.17-β) στην προβολή του χάρτη της γραμμής. Στην οθόνη των στάσεων της γραμμής υπάρχει η δυνατότητα για προβολή του προγράμματος της γραμμής (Εικόνα 3.17-γ), μετάβαση στον χάρτη της γραμμής (Εικόνα 3.17-δ) με επισημασμένη την επιλεγμένη στάση αυτή την φορά και προσθήκη της γραμμής στην λίστα αγαπημένων (Εικόνα 3.17-ε).

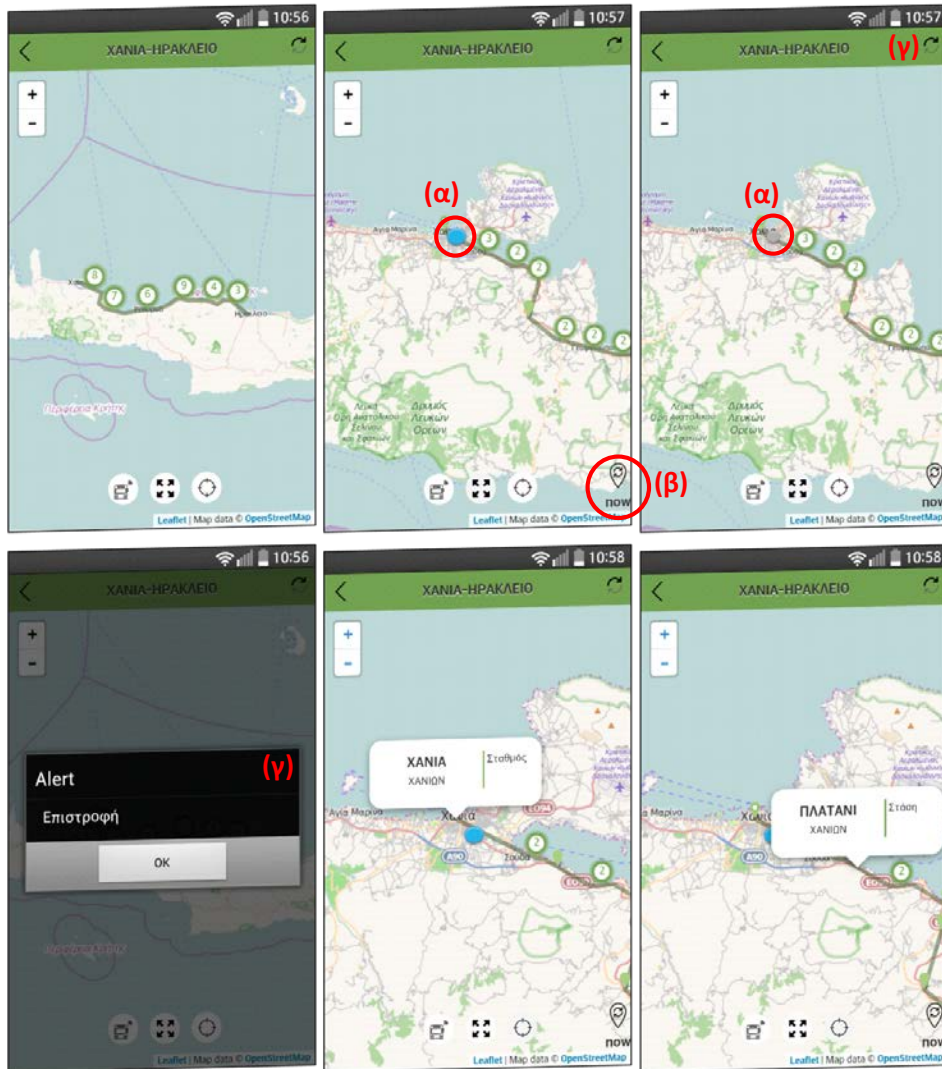


Εικόνα 3.17: Στιγμιότυπα υλοποίησης της λειτουργίας των Γραμμών 1

Αναφορικά με τον χάρτη, ο χρήστης μπορεί να πραγματοποιήσει εντοπισμό της τοποθεσίας του και προβολή αυτής στον χάρτη (Εικόνα 3.19-α). Η τοποθεσία συνοδεύεται με χρονική αναφορά (Εικόνα 3.19-β) και ο marker της αλλάζει αφότου παρέλθει προκαθορισμένος χρόνος. Επίσης, υπάρχουν πλήκτρα για μεγιστοποίηση των ορίων του χάρτη για προβολή όλων των στοιχείων που προβάλλονται σε αυτόν και προβολή των λεωφορείων που εκτελούν τη συγκεκριμένη διαδρομή σε πραγματικό χρόνο (λεπτομέρειες στην Παράγραφο 3.5.6). Εφόσον πρόκειται για γραμμή διπλής κατεύθυνσης, υπάρχει η δυνατότητα εναλλαγής μεταξύ μετάβασης και επιστροφής (Εικόνα 3.19-γ).



Εικόνα 3.18: Στιγμιότυπα υλοποίησης της λειτουργίας των Γραμμών 2

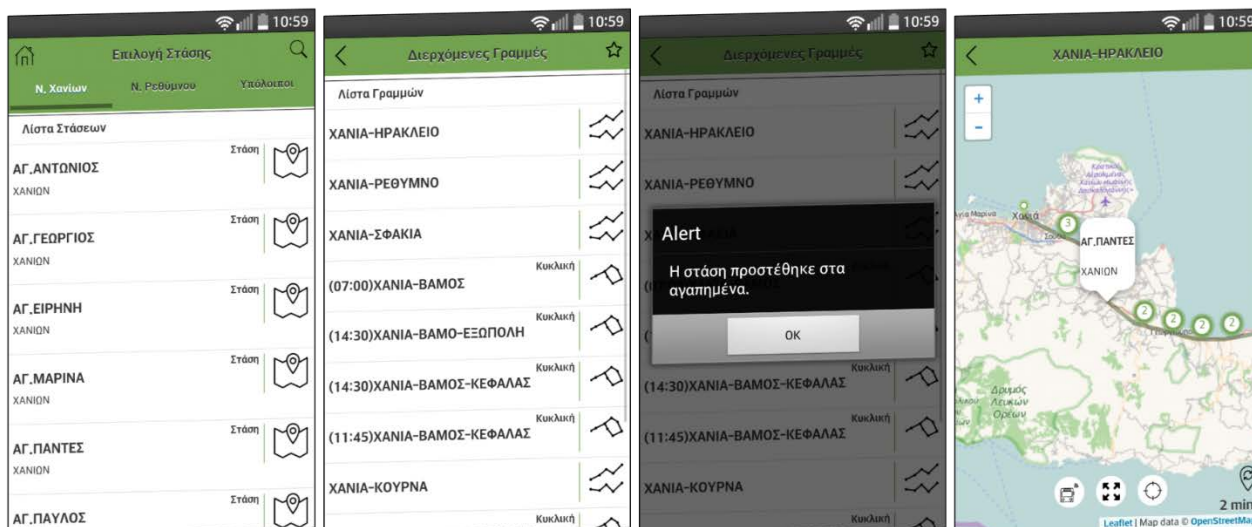


Εικόνα 3.19: Στιγμιότυπα υλοποίησης της λειτουργίας των Γραμμών 3

### 3.5.3 Στάσεις

Η επιλογή προβολής των στάσεων ακολουθεί την λογική της λειτουργίας των γραμμών μόνο που σε αυτή την περίπτωση πραγματοποιείται η αντίστροφη αναζήτηση (Εικόνα 3.20). Συγκεκριμένα, η λειτουργία επιτρέπει την:

- Αναζήτηση μίας στάσης με βάση το νομό ή το όνομά της
- Προβολή της στάσης στον χάρτη
- Λίστα των διερχόμενων γραμμών
- Προβολή της επιλεγμένης διερχόμενης γραμμής στον χάρτη
- Προβολή λεωφορείων που εκτελούν την επιλεγμένη γραμμή στον χάρτη



Εικόνα 3.20: Στιγμιότυπα υλοποίησης της λειτουργίας των Στασεων

### 3.5.4 Οργάνωση ταξιδιού

Η λειτουργία *Οργάνωσης Ταξιδιού* συνδυάζει τις διαθέσιμες διαδρομές για την μετακίνηση μεταξύ δύο στάσεων του δικτύου. Η υλοποίηση της περιλαμβάνει (Εικόνα 3.21):

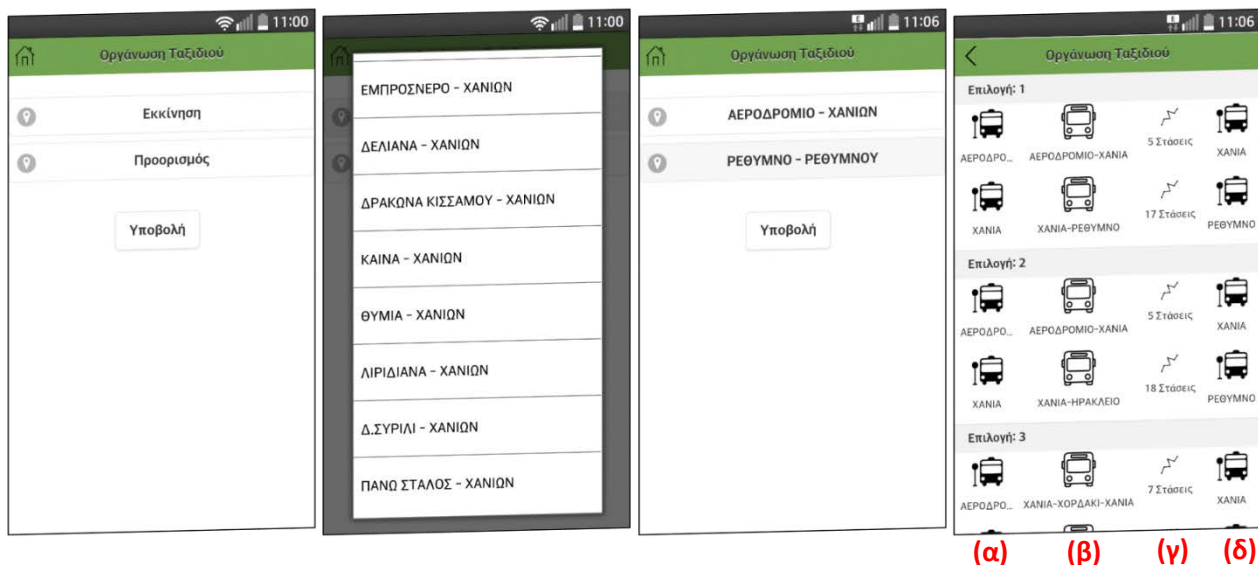
- Επιλογή παραμέτρων ταξιδιού
  - Στάση εκκίνησης
  - Στάση προορισμού
- Προβολή διαθέσιμων εναλλακτικών

Βασικές παράμετροι κόστους στην επιλογή των διαδρομών είναι ο αριθμός των μετεπιβιβάσεων και των ενδιάμεσων στάσεων. Λόγο της διάρθρωσης του δικτύου, συνήθως η συντομότερη διαδρομή είναι και αυτή χωρίς ενδιάμεση στάση.

Για κάθε επιμέρους διαδρομή των διαθέσιμων εναλλακτικών γίνεται αναφορά:

- Στην στάση εκκίνησης ή μετεπιβίβασης (Εικόνα 3.21-α)
- Στο ονομασία της διαδρομής (Εικόνα 3.21-β)
- Στον αριθμός ενδιάμεσων στάσεων (Εικόνα 3.21-γ)
- Στην στάση προορισμού ή μετεπιβίβασης (Εικόνα 3.21-δ)





Εικόνα 3.21: Στιγμιότυπα υλοποίησης της λειτουργίας Οργάνωσης Ταξιδιού 1

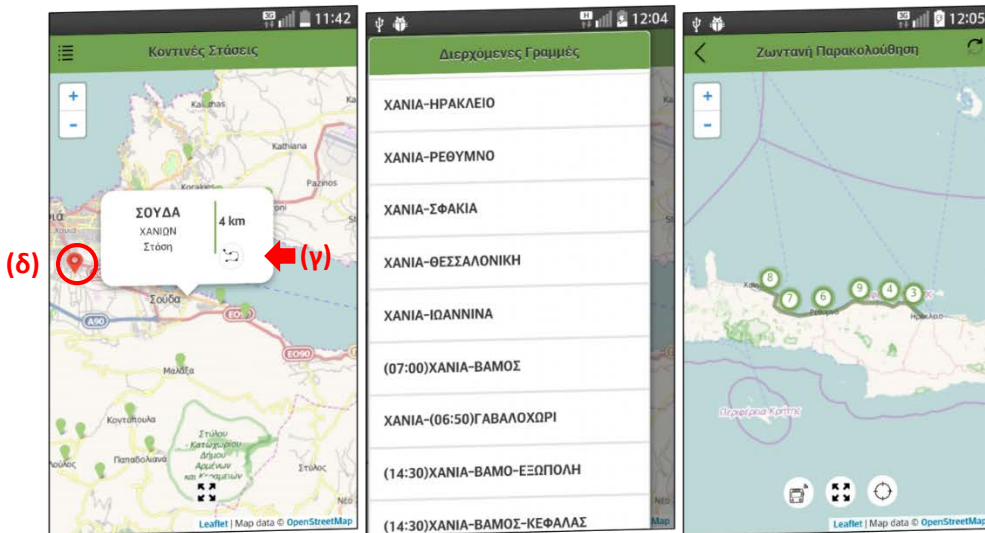
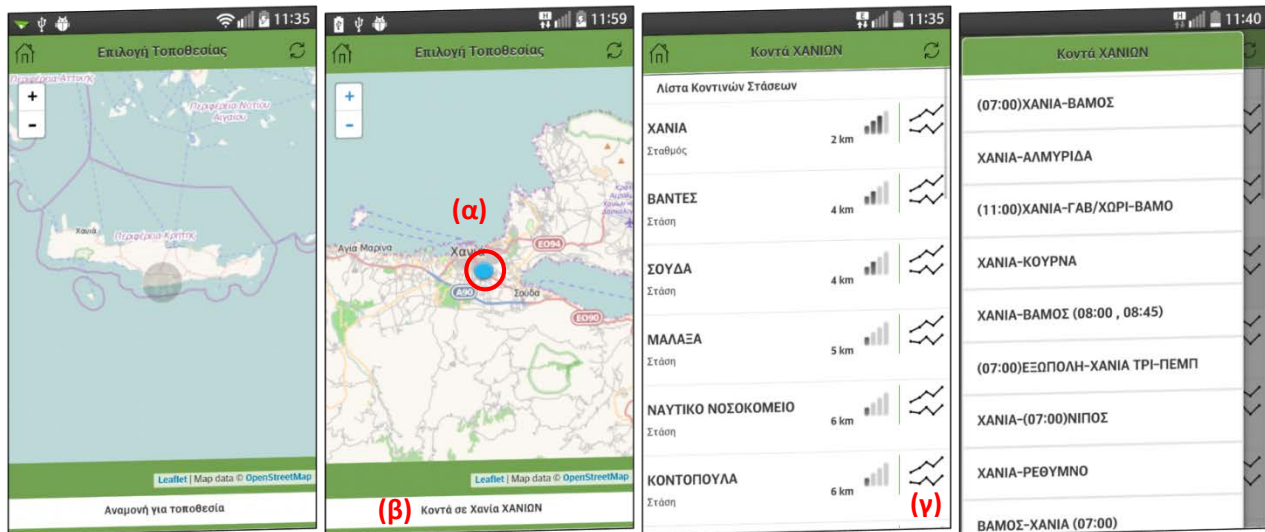
### 3.5.5 Κοντινές στάσεις

Η λειτουργία *Κοντινών στάσεων* κάνει χρήση της τοποθεσίας του χρήστη για τον προσδιορισμό των πλησιέστερων σε αυτόν στάσεων (Εικόνα 3.20). Αναλυτικά:

- Προσδιορισμός της τοποθεσίας του χρήστη είτε αυτόματα είτε χειροκίνητα.
- Εύρεση των πλησιέστερων στάσεων.
- Αναζήτηση γραμμών που διέρχονται από τις επιλεγμένες στάσεις.

Κατά την μετάβαση του χρήστη στη συγκεκριμένη οθόνη, αυτόματα πραγματοποιείται λήψη της τοποθεσίας του χρήστη (Εικόνα 3.22-α). Με τον επιτυχή εντοπισμό της τοποθεσίας, ενεργοποιείται η λειτουργία αντίστροφης γεωκωδικοποίησης για τον προσδιορισμό της πλησιέστερης πόλης (Εικόνα 3.22-β). Στην οθόνη παρουσίασης των αποτελεσμάτων, ο χρήστης έχει την δυνατότητα να ανανεώσει την λίστα με επαναπροσδιορισμό της τρέχουσας τοποθεσίας του, ενώ στο χάρτη παρουσιάζονται οι κοντινές σε αυτόν στάσεις. Μέσω Pop Up παραθύρων στην λίστα και στον χάρτη (Εικόνα 3.22-γ) παρουσιάζονται οι διαθέσιμες γραμμές, με δυνατότητα μετάβασης στην ζωντανή παρακολούθηση αυτών. Αν αποτύχει η διαδικασία εντοπισμού τοποθεσίας, ο χρήστης με παρατεταμένο πάτημα εισάγει χειροκίνητα την τοποθεσία ενδιαφέροντος και η διαδικασία συνεχίζεται (Εικόνα 3.22-δ).



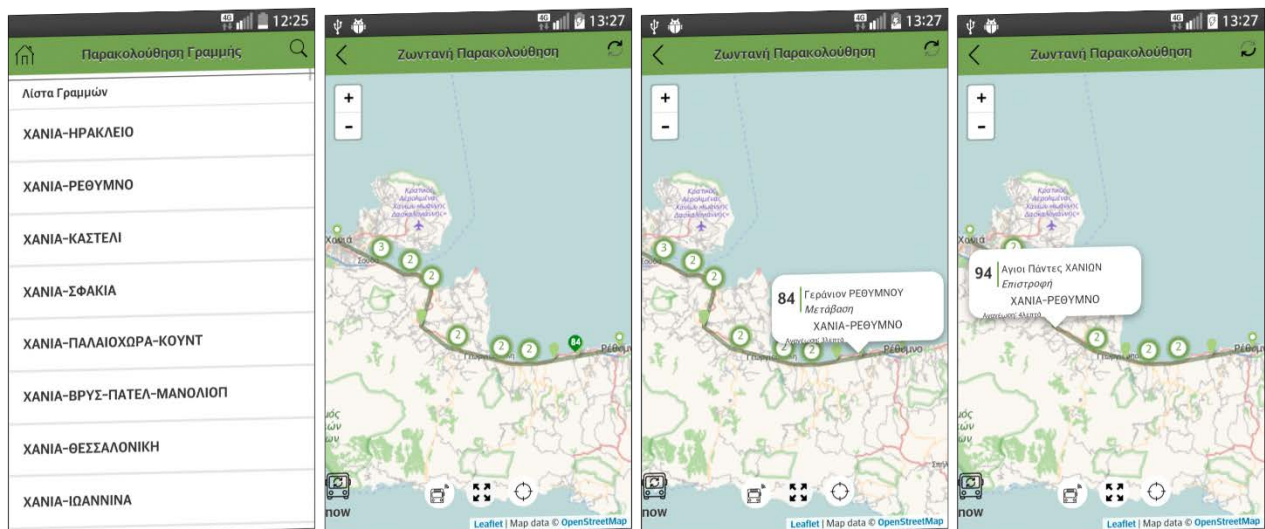


Εικόνα 3.22: Στιγμιότυπα υλοποίησης της λειτουργίας των Κοιτών Στάσεων

### 3.5.6 Παρακολούθηση Γραμμής Ενδιαφέροντος

Η μετάβαση στη συγκεκριμένη λειτουργία πραγματοποιείται σε αρκετά σημεία μέσα από την εφαρμογή και στοχεύει στην αναζήτηση και παρουσίαση όλων των λεωφορείων που εκτελούν μία γραμμή την δεδομένη χρονική στιγμή. Όταν ο χρήστης βρίσκεται εντός της λειτουργίας πραγματοποιούνται αιτήσεις προς τον εξυπηρετητή σε χρόνο ανάλογο με τον ρυθμό ανανέωσης των δεδομένων του συστήματος και παρουσίαση των αποτελεσμάτων στον χάρτη της γραμμής.

Στην προβολή του χάρτη (Εικόνα 3.23), σε περίπτωση που δεν πρόκειται για κυκλική διαδρομή, ο χρήστης μπορεί να εναλλάξει μεταξύ μετάβασης και επιστροφής της γραμμής.



Εικόνα 3.23: Στιγμιότυπα υλοποίησης της λειτουργίας Παρακολούθησης Γραμμής

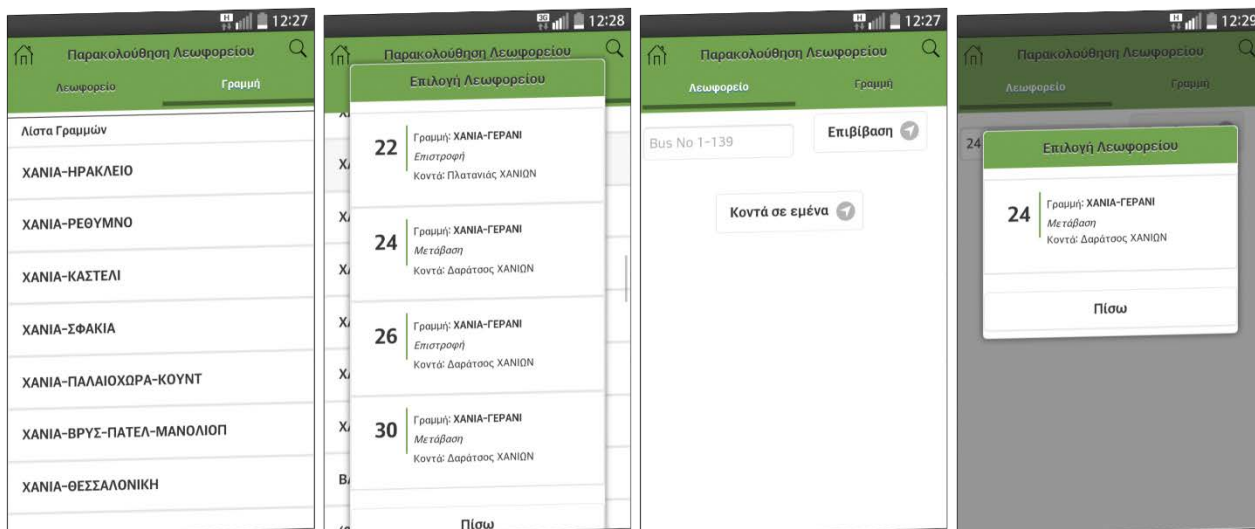
### 3.5.7 Παρακολούθηση Λεωφορείου Ενδιαφέροντος

Η λειτουργία *Παρακολούθησης Λεωφορείου Ενδιαφέροντος* παρέχει πληροφορίες για την κίνηση επιλεγμένου λεωφορείου. Λόγω της αδυναμίας που προκύπτει καμιά φορά στον προσδιορισμό του αριθμού του λεωφορείου δίνονται διάφορες εναλλακτικές (Εικόνα 3.24). Και εδώ ορίζεται χρονική επανάληψη των αιτήσεων προς τον εξυπηρετητή για ανανέωση της τοποθεσίας του λεωφορείου. Αναλυτικά, στη συγκεκριμένη λειτουργία πραγματοποιείται:

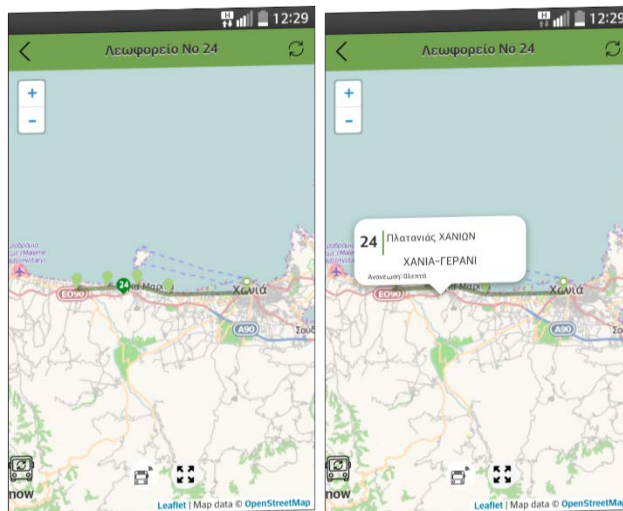
- Επιλογή λεωφορείου ενδιαφέροντος
- Προβολή χάρτη με την κίνηση του λεωφορείου σε πραγματικό χρόνο (Εικόνα 3.25)

Για την επιλογή του λεωφορείου ενδιαφέροντος δίνονται οι παρακάτω εναλλακτικές (Εικόνα 3.24):

- Επιλογή πλησιέστερου λεωφορείου
- Επιλογή λεωφορείου με βάση τον αναγνωριστικό αριθμό του
- Επιλογή λεωφορείου που εκτελεί συγκεκριμένη γραμμή



Εικόνα 3.24: Στιγμιότυπα υλοποίησης της λειτουργίας Παρακολούθησης Λεωφ. 1

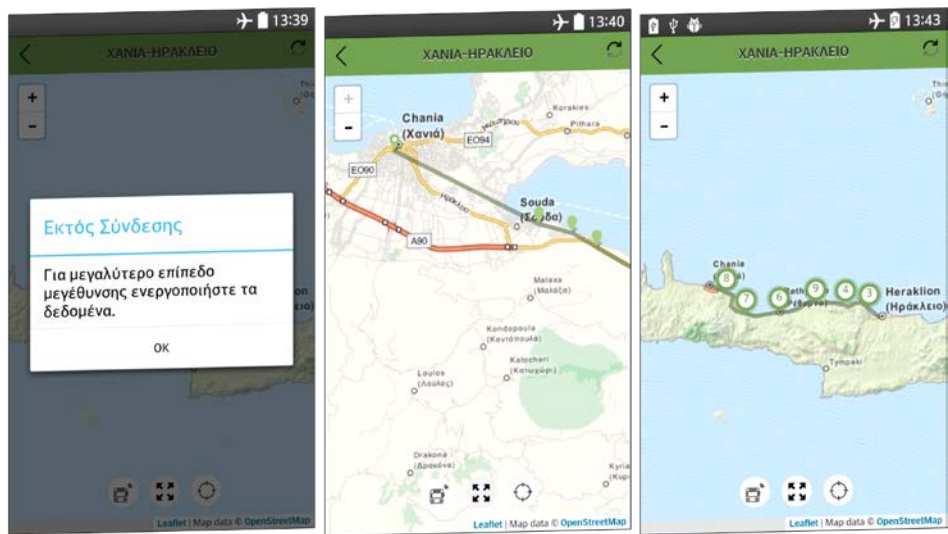


Εικόνα 3.25: Στιγμιότυπα υλοποίησης της λειτουργίας Παρακολούθησης Λεωφ. 2

### 3.5.8 Χρήση εκτός σύνδεσης

Εκτός σύνδεσης ο χρήστης μπορεί να εξετάσει την διάρθρωση του δικτύου, να μεταβεί στο συνηθισμένο πρόγραμμα δρομολογίων και να προβάλει στον χάρτη τις γραμμές και στάσεις ενδιαφέροντος (Εικόνα 3.26). Οι χάρτες που διατίθενται εκτός σύνδεσης, έχουν περιορισμένο εύρος εστίασης, ώστε να μην είναι αρκετά μεγάλο το μέγεθος των αρχείων που συνοδεύει την εγκατάσταση της εφαρμογής. Συγκεκριμένα:

- Εύρος μεγέθυνσης *offline*: 1-12 (Διακρίνεται πλήρως το οδικό δίκτυο οι οικισμοί)
  - Μέγιστη κλίμακα 1:~150 000
- Εύρος μεγέθυνσης *online*: 1-18
  - Μέγιστη κλίμακα 1:~2 000.



Εικόνα 3.26: Στιγμιότυπα υλοποίησης της λειτουργίας Χαρτών εκτός σύνδεσης

## 4. Συμπεράσματα και Προοπτικές

Ολοκληρώνοντας την παρούσα εργασία είναι σημαντικό να γίνει υποβολή των παρατηρήσεων και σπουδαιότερων συμπερασμάτων από την διαδικασία ανάπτυξης της εφαρμογής, καθώς και των ενδεχόμενων προοπτικών για βελτίωση του συστήματος.

### 4.1 Συμπεράσματα

Η δημιουργία υβριδικών εφαρμογών παρουσιάζει ενδιαφέρον λόγω της ευκολίας που έχει ο προγραμματιστής να δημιουργήσει και να διανέμει την εφαρμογή του στα διάφορα κατ'εξοχήματα εφαρμογών. Μια εφαρμογή που σκοπό έχει την εξυπηρέτηση του κοινού μίας επιχείρησης δεν μπορεί να περιοριστεί στην διάθεση της σε μια πλατφόρμα. Παράλληλα, η συντήρηση και αναβάθμιση αυτών δεν παρουσιάζει επαναλαμβανόμενες διαδικασίες. Συνήθως, οι συγκεκριμένες εφαρμογές δεν έχουν ιδιαίτερες υπολογιστικές απαιτήσεις, αφού ο απώτερος σκοπός τους είναι κυρίως η πληροφόρηση του χρήστη σχετικά με δεδομένα που η επεξεργασία τους πραγματοποιείται στην πλευρά του εξυπηρετητή.

Σχετικά με την απόδοση της εφαρμογής που δημιουργήθηκε, σε σύγχρονες συσκευές μπορεί να συγκριθεί με αυτή των native εφαρμογών. Η λειτουργία της εξετάστηκε σε διάφορες συσκευές Android, ακόμα και σε low end συσκευές, με πολύ καλά αποτελέσματα. Καθυστερήσεις εμφανίζονται μόνο στην περίπτωση συσκευών με περιορισμένη υπολογιστική ισχύς κατά την διάρκεια συμπλήρωσης της λίστας με τις στάσεις του δικτύου λόγω του μεγάλου πλήθους στοιχείων που την απαρτίζουν. Η επιλογή για αποθήκευση δεδομένων τοπικά στη συσκευή αύξησε σημαντικά την ταχύτητα λειτουργίας της, καθώς, ο χρήστης αποφεύγει τις καθυστερήσεις από την λήψη των δεδομένων, ενώ παράλληλα μειώνεται σε σημαντικό βαθμό ο φόρτος του εξυπηρετητή.

Αναφορικά με τις λειτουργίες της εφαρμογής, η οργάνωση ταξιδιού που στοχεύει στον βέλτιστο δυνατό συνδυασμό διαδρομών του δικτύου, λόγω της έλλειψης πληροφοριών σχετικά με τον χρόνο που διέρχεται το λεωφορείο από τις στάσεις, δεν επιτρέπει την αξιοποίηση στο μέγιστο των δυνατοτήτων δρομολόγησης. Η δημιουργία λίστας αγαπημένων προσφέρει σημαντική χρηστικότητα στην εφαρμογή, ενώ η δυνατότητα εντοπισμού τοποθεσίας δίνει άμεση ενημέρωση για τις επιλογές μετακίνησης που έχει ο επιβάτης όταν βρίσκεται στον δρόμο. Τέλος, έγινε μια πρόταση για τον τρόπο διάθεσης δυναμικών δεδομένων μέσα από τις διάφορες λειτουργίες της εφαρμογής και την αξιοποίηση που αυτές μπορούν να έχουν στο συγκεκριμένο δίκτυο.

Όσον αφορά τη βάση δεδομένων, αυτή έχει σχεδιαστεί και συμπληρωθεί σύμφωνα με τα διαθέσιμα δεδομένα. Στο βαθμό που ήταν δυνατό, έγινε προσπάθεια σχεδιασμού ενός συστήματος στα πρότυπα του GTFS, ούτως ώστε να είναι δυνατή η μεταφορά του σχεδιασμού και σε άλλα δίκτυα. Παρόλα αυτά, καθώς το υποτιθέμενο σύστημα του παρόχου μεταβάλλεται



στο χρόνο, και έχει απευθείας εξάρτηση από εξωτερικά δεδομένα, ο σχεδιασμός του ήταν αναγκαίο να είναι κοντά στα διάφορα υποσυστήματα του υπό εξέταση ΚΤΕΛ για την ανανέωση των στοιχείων από αυτό.

## 4.2 Προοπτικές

Η ανάπτυξη υβριδικών εφαρμογών βρίσκεται στο επίκεντρο και θα συνεχίσει να βρίσκεται μέχρι κάποια μέρα να καθιερωθεί ένας κοινός τρόπος ανάπτυξης εφαρμογών μεταξύ των πλατφόρμων κινητών συσκευών. Νέες τεχνολογίες διαρκώς έρχονται και φεύγουν από το προσκήνιο. Συνήθως ότι κερδίζει κάποιο νέο πλαίσιο σε ταχύτητα και λειτουργικότητα, το χάνει σε διαθέσιμα πρόσθετα και υποστήριξη από τις κοινότητες χρηστών που μόλις ξεκινάνε να αναπτύσσονται. Το ενδιαφέρον γύρω από την ανάπτυξη πλαισίων HTML5 Mobile UI είναι ιδιαίτερα έντονο με πλαίσια να εστιάζουν κυρίως στην προσομοίωση της εκάστοτε πλατφόρμας για την δημιουργία εφαρμογών που μιμούνται την εμφάνιση των native με σημαντικές βελτιώσεις στην ταχύτητα αυτών. Επιπλέον, αναπτύσσονται όλο και περισσότερα νέα εργαλεία για την δημιουργία εφαρμογών για κινητές συσκευές εισάγοντας νέες μεθόδους. Τα νέα εργαλεία απλοποιούν ακόμα περισσότερο την διαδικασία διάθεσης της εφαρμογής, ενώ ορισμένα κάνουν χρήση ολοκληρωμένων IDE για την δημιουργία δια-πλατφορμικών native εφαρμογών.

Η ανάπτυξη της εφαρμογής επικεντρώθηκε στην πλατφόρμα Android. Το περιβάλλον χρήστη είναι εμπνευσμένο από την πλατφόρμα Android, ώστε να είναι κοντά σε μια native εφαρμογή της πλατφόρμας. Για την εγκατάσταση της εφαρμογής σε συσκευή iOS απαιτείται συνδρομή στο Apple Developer Toolkit και ελάχιστες τροποποιήσεις στον κώδικα και τα γραφικά. Η απόδοση της εφαρμογής σε μια iOS πλατφόρμα εκτιμάται ότι θα είναι σημαντικά καλύτερη.

Τα διαθέσιμα δεδομένα περιορίζαν τις δυνατότητες που θα μπορούσε να έχει η εφαρμογή. Όπως προαναφέρθηκε, οι ώρες αναχωρήσεως αναφέρονταν μόνο στη στάση αφετηρίας, ενώ οι στάσεις σε αρκετές περιπτώσεις είχαν χωρική αναφορά στο κέντρο του οικισμού. Η συγκέντρωση και διάθεση του συνόλου των απαιτούμενων στοιχείων σχετικά με την διάρθρωση και λειτουργία του δικτύου θα έχεις ως αποτέλεσμα μια εφαρμογή που θα μπορεί να προσφέρει ακόμα περισσότερες δυνατότητες ενημέρωσης στον τελικό χρήστη.

Αναφορικά με τα δεδομένα που μεταβάλλονται στο χρόνο, αυτά δεν προέρχονται απευθείας από το σύστημα του ΚΤΕΛ, αλλά αφορούν ενδεικτικά στοιχεία αποθηκευμένα στον εξυπηρετητή. Με τα διαθέσιμα δεδομένα και τεχνολογίες αναπτύχθηκε μια εφαρμογή που φιλοδοξεί να ανταπεξέλθει στις απαιτήσεις του συστήματος. Η λειτουργία της εφαρμογής σε πραγματικές συνθήκες, με δεδομένα που θα ανανεώνονται σε πραγματικό χρόνο δεν θα μπορούσε να λείπει από τις προοπτικές της παρούσας εργασίας.

Για την υλοποίηση του συστήματος δημιουργήθηκε ένας Web Server, ο οποίος φιλοξενεί όλα τα δεδομένα και τα απαιτούμενα APIs της εφαρμογής. Ωστόσο, η χρήση της εφαρμογής

από αρκετούς χρήστες, οι οποίοι θα υποβάλουν ταυτόχρονα αιτήσεις προς τον εξυπηρετητή, απαιτεί ένα ισχυρό υπολογιστικό σύστημα με τις ανάλογες τεχνολογίες για να ανταπεξέλθει.

Τέλος, η λειτουργία οργάνωσης ταξιδιού βασίζεται στην ελαχιστοποίηση των μετεπιβιβάσεων και του αριθμού των ενδιάμεσων στάσεων. Η διάθεση περισσότερων δεδομένων από μεριάς φορέα θα επιτρέψει την ακόμα ακριβέστερη οργάνωση του ταξιδιού και την χρήση αλγόριθμων που λαμβάνουν υπόψη τους και άλλες παραμέτρους (όπως του pgRouting του PostGIS).





## Βιβλιογραφία

- [1] Statista. (2015). *Statista - The Statistics Portal*. Προσπελάστηκε 28 Ιανουαρίου 2015, από <http://www.statista.com/>
- [2] Gok, N., & Khanna, N. (2013). *Building hybrid Android apps with Java and JavaScript*. Sebastopol, CA: O'Reilly Media.
- [3] Xamarin.com. (2015). *Mobile App Development & App Creation Software - Xamarin*. Προσπελάστηκε 15 Ιανουαρίου 2015, from <http://xamarin.com/>
- [4] Saleh, H. (2014). *JavaScript mobile application development*. Birmingham, U.K.: Packt Pub.
- [5] Malavolta, I. (2014). *HTML5: the new frontier of the web*. Presentation, Gran Sasso Science Institute | University of L'Aquila.
- [6] Phonegap.com. (2015). *PhoneGap*. Προσπελάστηκε 21 Φεβρουαρίου 2015, από <http://phonegap.com/>
- [7] Cordova.apache.org. (2015). *Apache Cordova*. Προσπελάστηκε 21 Φεβρουαρίου 2015, από <https://cordova.apache.org/>
- [8] Dev.w3.org. (2015). *Geolocation API Specifications*. Προσπελάστηκε 2 Δεκεμβρίου 2014, από <http://dev.w3.org/geo/api/>
- [9] Sqlite.org. (2015). *SQLite Home Page*. Προσπελάστηκε 8 Δεκεμβρίου 2014, από <https://www.sqlite.org/>
- [10] GitHub. (2015). *Brodysoft - Cordova-SQLitePlugin*. Προσπελάστηκε 18 Δεκεμβρίου 2014, από <https://github.com/brodysoft/Cordova-SQLitePlugin/>
- [11] W3.org. (2015). *Web SQL Database*. Προσπελάστηκε 1 Δεκεμβρίου 2014, από <http://www.w3.org/TR/webdatabase/>
- [12] W3.org. (2015). *HTML5*. Προσπελάστηκε 21 Φεβρουαρίου 2015, από <http://www.w3.org/TR/html5/>
- [13] McLaughlin, B. (2011). *What is HTML5?*. Beijing: O'Reilly.
- [14] Nodejs.org. (2015). *Node.js*. Προσπελάστηκε 18 Φεβρουαρίου, από <https://nodejs.org/>

- [15] W3.org. (2015). *Introduction to CSS3*. Προσπελάστηκε 21 Φεβρουαρίου 2015, από <http://www.w3.org/TR/2001/WD-css3-roadmap-20010523/>
- [16] jQuery.org. (2015). *jQuery Foundation*. Προσπελάστηκε 2 Φεβρουαρίου 2015, από <http://jquery.org/>
- [17] W3techs.com. (2015). *Usage Statistics and Market Share of JQuery for Websites*, March 2015. Προσπελάστηκε 2 Φεβρουαρίου 2015, από <http://w3techs.com/technologies/details/js-jquery/all/all/>
- [18] Firtman, M. (2012). *jQuery mobile*. Sebastopol, CA: O'Reilly Media.
- [19] jQuery Mobile. (2015). *jQuery Mobile*. Προσπελάστηκε 2 Φεβρουαρίου 2015, από <http://jquerymobile.com/>
- [20] Lindley, C. (2010). *JQuery cookbook*. Sebastopol, CA: O'Reilly.
- [21] Backbonejs.org. (2015). *Backbone.js*. Προσπελάστηκε 12 Δεκεμβρίου 2015, από <http://backbonejs.org/>
- [22] Osmani, A. (2012). *Developing Backbone.js applications*. Beijing: O'Reilly.
- [23] Mulder, P. (2014). *Developing Web Applications with Backbone.js*. Sebastopol: O'Reilly & Associates.
- [24] Roemer, R. (2013). *Backbone.js testing*. Birmingham: Packt Pub.
- [25] Underscorejs.org. (2015). *Underscore.js*. Προσπελάστηκε 2 Φεβρουαρίου 2015, από <http://underscorejs.org/>
- [26] Requirejs.org. (2015). *RequireJS*. Προσπελάστηκε 17 Φεβρουαρίου 2015, από <http://requirejs.org/>
- [27] Wikipedia. (2015). *Asynchronous module definition*. Προσπελάστηκε 18 Φεβρουαρίου 2015, από [http://en.wikipedia.org/wiki/Asynchronous\\_module\\_definition/](http://en.wikipedia.org/wiki/Asynchronous_module_definition/)
- [28] Leafletjs.com. (2015). *Leaflet — An open-source JavaScript library for interactive maps*. Προσπελάστηκε 17 Φεβρουαρίου 2015, από <http://leafletjs.com/>
- [29] Wikipedia. (2015). *Server (computing)*. Προσπελάστηκε 2 Φεβρουαρίου 2015, από [http://en.wikipedia.org/wiki/Server\\_\(computing\)/](http://en.wikipedia.org/wiki/Server_(computing)/)

- [30] Wiki.nginx.org. (2015). *Nginx Community*. Προσπελάστηκε 10 Φεβρουαρίου 2015, από <http://wiki.nginx.org/Main/>
- [31] Php.net. (2015). *PHP: Hypertext Preprocessor*. Προσπελάστηκε 8 Φεβρουαρίου 2015, από <http://php.net/>
- [32] Sklar, D., & Trachtenberg, A. (2006). *PHP cookbook*. Sebastopol, Calif: O'Reilly.
- [33] Postgresql.org. (2015). *PostgreSQL: The world's most advanced open source database*. Προσπελάστηκε 27 March 2015, από <http://www.postgresql.org/>
- [34] Www-01.ibm.com. (2015). *IBM Knowledge Center*. Προσπελάστηκε 18 Φεβρουαρίου 2015, από [http://www-01.ibm.com/support/knowledge-center/SSEPGG\\_9.1.0/com.ibm.db2.udb.admin.doc/doc/c0004100.htm](http://www-01.ibm.com/support/knowledge-center/SSEPGG_9.1.0/com.ibm.db2.udb.admin.doc/doc/c0004100.htm)
- [35] Postgis.net. (2015). *PostGIS — Spatial and Geographic Objects for PostgreSQL*. *Postgis.net*. Προσπελάστηκε 8 Φεβρουαρίου 2015, από <http://postgis.net>
- [36] Postgis.net. (2015). *PostGIS 2.1.7dev Manual*. Προσπελάστηκε 1 Φεβρουαρίου 2015, από <http://postgis.net/docs/>
- [37] Holdener, A. (2008). *Ajax*. Beijing: O'Reilly.
- [38] Richardson, L., & Amundsen, M. (2013). *RESTful Web APIs*. Sebastopol, Calif.: O'Reilly.
- [39] Restful-api-design.readthedocs.org. (2015). *The Job of the API Designer*. Προσπελάστηκε 2 Φεβρουαρίου 2015, από <http://restful-api-design.readthedocs.org/en/latest/scope.html>
- [40] Json.org. (2015). *JSON*. Προσπελάστηκε 15 Φεβρουαρίου 2015, από <http://json.org/>
- [41] Geojson.org. (2015). *GeoJSON*. Προσπελάστηκε 3 Φεβρουαρίου 2015, από <http://geojson.org/>
- [42] Google Developers. (2015). *General Transit Feed Specification Reference*. Προσπελάστηκε 15 Φεβρουαρίου 2015, από <https://developers.google.com/transit/gtfs/reference/>
- [43] GTFS-realtime . (2015). *What is GTFS-realtime?*. *Google Developers*. Προσπελάστηκε 2 Φεβρουαρίου 2015, από <https://developers.google.com/transit/gtfs-realtime>
- [44] Google Maps. (2015). *Google Maps - Mobile App*. Προσπελάστηκε 2 Φεβρουαρίου 2015, από <https://play.google.com/store/apps/details?id=com.google.android.apps.maps&hl=el/>

- [45] Moovit. (2015). *Moovit - Mobile App*. Προσπελάστηκε 15 Φεβρουαρίου 2015, από <http://www.moovitapp.com/>
- [46] Optitrans-fp7.eu. (2015). *Opti-Trans - Mobile App*. Προσπελάστηκε 15 Φεβρουαρίου 2015, από <http://www.optitrans-fp7.eu/>
- [47] Wiki.openstreetmap.org. (2015). *YOURS - OpenStreetMap Wiki*. Προσπελάστηκε 7 Φεβρουαρίου 2015, από <http://wiki.openstreetmap.org/wiki/YOURS/>
- [48] Coenraets.org. (2012). *Using Backbone.js with jQuery Mobile | Christophe Coenraets*. Προσπελάστηκε 1 Φεβρουαρίου 2015, από <http://coenraets.org/blog/2012/03/using-backbone-js-with-jquery-mobile/>
- [49] Christie, M. (2014). *Marcus Christie's Blog: A Backbone-JQuery Mobile skeleton project - . Marcus-christie.blogspot.gr*. Προσπελάστηκε 1 Φεβρουαρίου 2015, από <http://marcus-christie.blogspot.gr/2014/06/a-backbone-jquery-mobile-skeleton.html>
- [50] *Mobile Atlas Creator*. (2015). *Mobile Atlas Creator (MOBAC) Site*. Προσπελάστηκε 5 Φεβρουαρίου 2015, από <http://mobac.sourceforge.net/>