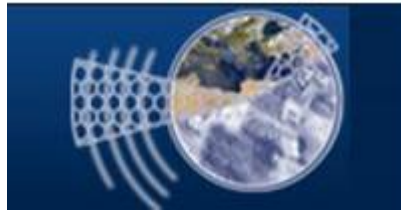




Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης
Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Μηχανικών Πληροφορικής



Πτυχιακή εργασία

Τίτλος: Εφαρμογή αναζήτησης και επιλογής ταξί

Αικατερίνη Παττίχα (ΑΜ: 3705)

Επιβλέπων καθηγητής: Αθανάσιος Μαλάμος

ΗΡΑΚΛΕΙΟ
2017

Abstract

The important role that taxis play in bringing tourist from airport Makedonia to Chalkidiki is often overlooked from the competent public services. In this thesis, we present *Halkidiki Transfer*, an online web application for searching and booking taxi.

First, we present and analyze the available tools that contribute building an online web application and which tools are used for this project.

Secondly, we present the approach of the implementation for the most important parts of the platform. Last but not least we evaluate the performance of the application for the speed and the user experience indicating that the application achieves a significant improvement in both of tourists' transportation and the increasing searching of the taxi.

Σύνοψη

Ο σημαντικός ρόλος που παίζουν τα ταξί στην μεταφορά τουριστών από το αεροδρόμιο Μακεδονία προς τον νομό Χαλκιδικής συχνά παραβλέπεται από τις αρμόδιες δημόσιες υπηρεσίες. Στην συγκεκριμένη πτυχιακή θα παρουσιάσουμε το *Halkidiki Transfer*, μια online εφαρμογή για αναζήτηση ταξί με δυνατότητα προ κράτησης.

Αρχικά παρουσιάζουμε και αναλύουμε τα εργαλεία που είναι διαθέσιμα για την δημιουργία μιας online εφαρμογής καθώς και ποια χρησιμοποιήθηκαν.

Έπειτα θα αναλύσουμε την προσέγγιση που ακολουθήσαμε ως προς την υλοποίηση για τα σημαντικότερα μέρη της πλατφόρμας. Τέλος, αξιολογούμε την απόδοση της εφαρμογής σε επίπεδα ταχύτητας και χρηστικότητας δείχνοντας ότι η εφαρμογή επιτυγχάνει σημαντική βελτίωση τόσο στην μεταφορά τουριστών αλλά και στην αύξηση ζήτησης ταξί.

Περιεχόμενα

ABSTRACT	II
ΣΥΝΟΨΗ	III
1. ΕΙΣΑΓΩΓΗ	1
1.1. ΚΙΝΗΤΡΑ ΓΙΑ ΤΗΝ ΔΙΕΞΑΓΩΓΗ ΤΗΣ ΠΤΥΧΙΑΚΗΣ	1
1.2. ΣΚΟΠΟΣ ΚΑΙ ΣΤΟΧΟΙ ΠΤΥΧΙΑΚΗΣ	1
1.3. ΔΟΜΗ ΠΤΥΧΙΑΚΗΣ.....	1
2. ΤΕΧΝΟΛΟΓΙΚΗ ΑΝΑΛΥΣΗ	2
2.1. HTML	2
2.2. CSS	2
2.3. JAVASCRIPT	3
2.3.1. JQUERY.....	3
2.4. PHP	4
2.5. RUBY ON RAILS	5
2.5.1. RUBY	5
2.5.2. RAILS.....	6
2.5.3. MODEL-VIEW-CONTROLLER (MVC).....	6
2.6. BOOTSTRAP.....	7
2.6.1. RESPONSIVE	7
2.6.2. ΛΟΓΟΙ ΕΠΙΛΟΓΗΣ BOOTSTRAP	8
2.7. ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ	9
2.7.1. POSTGRESQL.....	9
2.8. WEB SERVER	9
2.6.1. APACHE.....	9
2.6.2. NGINX.....	10
2.9. GOOGLE MAPS API	11
3. ΑΝΑΛΥΣΗ ΠΡΟΒΛΗΜΑΤΟΣ	12
3.1. ΣΧΕΔΙΑΣΜΟΣ ΥΛΟΠΟΙΗΣΗΣ	12
3.2. ΑΠΑΙΤΗΣΕΙΣ ΣΥΣΤΗΜΑΤΟΣ	12
3.3. ΥΛΟΠΟΙΗΣΗ.....	13
4. ΕΠΙΛΟΓΟΣ	38
4.1. ΑΠΟΤΕΛΕΣΜΑΤΑ.....	38
4.2. ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ ΚΑΙ ΕΠΕΚΤΑΣΕΙΣ	38
5. ΒΙΒΛΙΟΓΡΑΦΙΑ	39

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

ΕΙΚΟΝΑ 1 LOGO ΡΗΡ ΠΗΓΗ: HTTP://RHP.NET/IMAGES/LOGOS/RHP-MED-TRANS.PNG	4
ΕΙΚΟΝΑ 2 ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ ΑΙΤΗΜΑΤΟΣ ΑΠΟ ΤΟΝ CLIENT ΣΤΟΝ SERVER ΧΡΗΣΙΜΟΠΟΙΩΝΤΑΣ ΡΗΡ, ΠΗΓΗ: HTTPS://I.STACK.IMGUR.COM/LT50Y.GIF	5
ΕΙΚΟΝΑ 3 ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ MVC ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ. ΠΗΓΗ: HTTPS://BETTEREXPLAINED.COM/WP-CONTENT/UPLOADS/RAILS/MVC-RAILS.PNG	7
ΕΙΚΟΝΑ 4 LOGO ΒΟΟΤΣΤΡΑΠ, ΠΗΓΗ: HTTP://V4-ALPHA.GETBOOTSTRAP.COM/	8
ΕΙΚΟΝΑ 5 LOGO ΒΟΟΤΣΤΡΑΠ	8
ΕΙΚΟΝΑ 6 ΑΠΕΙΚΟΝΙΣΗ ΛΕΙΤΟΥΡΓΙΑΣ WEB SERVER, ΠΗΓΗ: HTTPS://RUSLANSPIVAK.COM/LSBAWS-PART1/LSBAWS_HTTP_REQUEST_RESPONSE.PNG	10
ΕΙΚΟΝΑ 7 LOGO NGINX ΠΗΓΗ HTTPS://CDN.WP.NGINX.COM/WP-CONTENT/UPLOADS/2016/08/NGINX-LOGO-SMALLER.JPG	11
ΕΙΚΟΝΑ 8 Ο ΠΙΝΑΚΑΣ USERS	13
ΕΙΚΟΝΑ 9 ΠΙΝΑΚΑΣ BOOKINGS	14
ΕΙΚΟΝΑ 10 ΕΙΚΟΝΑ 10 ΠΙΝΑΚΑΣ LOCATIONS.	15
ΕΙΚΟΝΑ 11 ΦΟΡΜΑ ΕΓΓΡΑΦΗΣ, ΠΗΓΗ HTTP://WWW.HALKIDIKITRANSFER.COM/USERS/SIGN_UP	16
ΕΙΚΟΝΑ 12 Η ΦΟΡΜΑ ΚΡΑΤΗΣΗ ΟΠΩΣ ΕΜΦΑΝΙΖΕΤΑΙ ΣΤΟΥ ΧΡΗΣΤΕΣ, ΠΗΓΗ HTTP://WWW.HALKIDIKITRANSFER.COM/	20
ΕΙΚΟΝΑ 13 ΦΟΡΜΑ ΑΝΑΖΗΤΗΣΗΣ ΤΑΞΙ ΜΕ ΤΑ ΕΠΙΠΛΕΟΝ ΠΕΔΙΑ, ΠΗΓΗ HTTP://WWW.HALKIDIKITRANSFER.COM/	23
ΕΙΚΟΝΑ 14 ΣΤΑΔΙΟ ΕΠΙΒΕΒΑΙΩΣΗΣ ΚΡΑΤΗΣΗΣ, ΠΗΓΗ HTTP://WWW.HALKIDIKITRANSFER.COM/BOOKING/151/CONTINUE	27
ΕΙΚΟΝΑ 15 ΕΜΑΙΛ ΕΠΙΒΕΒΑΙΩΣΗΣ ΠΟΥ ΛΑΜΒΑΝΕΙ Ο ΧΡΗΣΤΗΣ ΜΕΤΑ ΤΗΝ ΟΛΟΚΛΗΡΩΣΗ ΤΗΣ ΚΡΑΤΗΣΗΣ	36
ΕΙΚΟΝΑ 16 ΙΣΤΟΡΙΚΟ ΚΡΑΤΗΣΕΩΝ	37

1. Εισαγωγή

1.1. Κίνητρα για την διεξαγωγή της πτυχιακής

Το βασικό κίνητρο για την ανάληψη της συγκεκριμένης πτυχιακής είναι το γεγονός ότι το πρακτικό κομμάτι της εργασίας έχει ως αποτέλεσμα να δώσει μια εναλλακτική λύση σε ένα πραγματικό πρόβλημα που υπάρχει. Το πρόβλημα κράτησης ταξί από τουρίστες. Επιπλέον σημαντικό κίνητρο αποτέλεσε και η απόκτηση γνώσεων σχετικά με την κατασκευή online εφαρμογών καθώς και η εξοικείωση με νέες τεχνολογίες όπως Ruby on Rails, Bootstrap και Postgresq.

1.2. Σκοπός και στόχοι πτυχιακής

Κύριος σκοπός της εργασίας είναι η σχεδίαση και η υλοποίηση online εφαρμογής για αναζήτηση και κράτηση ταξί. Η εφαρμογή να είναι φιλική προς τον χρήστη αλλά και τον ιδιοκτήτη ταξί. Η σχεδίαση της εφαρμογής αποτελείται από τα εξής:

- online εφαρμογή για τους χρήστες
- Server με βάση δεδομένων.

Οι χρήστες έχουν την δυνατότητα να κάνουν κράτηση μέσω της εφαρμογής με την προσωπική τους συσκευή. Η κράτηση γίνεται με βάση τις διαδρομές και τις ημερομηνίες που επιθυμούν. Εφόσον επιλέξουν διαδρομή, οι χρήστες έχουν την ευκαιρία να δουν την επιλεγμένη διαδρομή στον χάρτη, το χρόνο που απαιτείται και την τιμή για την εκάστοτε διαδρομή.

Τέλος, ο βασικός κορμός της εφαρμογής αποτελείται από τον web server με την βάση δεδομένων. Είναι τα κύρια μέρη τα οποία είναι υπεύθυνα για την διαχείριση την πληροφορίας. Στην βάση αποθηκεύονται όλα τα δεδομένα σχετικά με τις κρατήσεις καθώς και τις επιπλέον πληροφορίες για την ομαλή λειτουργία της εφαρμογής.

1.3. Δομή πτυχιακής

Πρακτικά η εργασία χωρίζεται σε 4 βασικά μέρη. Αρχικά γίνεται περιγραφή των τεχνολογιών που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής, έπειτα αναλύουμε τις απαιτήσεις και παρουσιάζουμε την επίλυση. Τέλος παρουσιάζουμε τα τεστ σενάρια και περιγράφουμε τα συμπεράσματα. Περιγραφικά τα μέρη της πτυχιακής είναι ως εξής:

- Το δεύτερο κεφάλαιο εστιάζει κυρίως στο τεχνολογικό κομμάτι. Αναλύουμε τις γλώσσες προγραμματισμού καθώς και τις νέες τεχνολογίες που βοηθούν στην γρήγορη ανάπτυξη εφαρμογών στον ίντερνετ.
- Το τρίτο κεφάλαιο αποτελεί την υλοποίηση της εφαρμογής. Αναλύουμε το πρόβλημα, τις απαιτήσεις του συστήματος και παρουσιάζουμε την υλοποίηση για τα σημαντικά μέρη της εφαρμογής.
- Στο τέταρτο κεφάλαιο συνοψίζουμε τα συμπεράσματα που δημιουργήθηκαν από την αξιολόγηση και παρουσιάζουμε πληροφορίες για μελλοντικές πιθανές βελτιστοποιήσεις.

2. Τεχνολογική ανάλυση

2.1. HTML

Το ακρόνυμο HTML προέρχεται από τις λέξεις HyperText Markup Language. Συνήα η HTML θεωρείται ως γλώσσα προγραμματισμού αλλά πρόκειται για μια γλώσσα σήμανσης. Πρακτικά η γλώσσα σήμανσης δεν είναι τίποτα παραπάνω από έναν ειδικό τρόπο γραφής και γ αυτόν τον λόγο δεν χρειάζεται μεταγλωττιστή για να εκτελεστεί

Η HTML χρησιμοποιεί ετικέτες για να πληροφορήσει τον browser τι να εμφανίσει. Οι ετικέτες [1] συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα `<h1>` και `</h1>`), με την πρώτη να ονομάζεται ετικέτα έναρξης και τη δεύτερη ετικέτα λήξης (ή σε άλλες περιπτώσεις ετικέτα ανοίγματος και ετικέτα κλεισίματος αντίστοιχα). Ο καθένας μπορεί να γράψει HTML με έναν απλό επεξεργαστή κειμένου και να το αποθηκεύσει με την κατάληξη αρχείου `.html`.

Ο browser παίρνει τις πληροφορίες από τον web server, διαβάζει το html αρχείο και συνθέτει την σελίδα με βάση τις ετικέτες. Οι ετικέτες δεν εμφανίζονται στον χρήστη αλλά όπως αναφερθήκαμε και πιο πάνω χρησιμοποιούνται για να ερμηνεύσει ο browser το περιεχόμενο. Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες, λίστες κλπ.

2.2. CSS

Η CSS [2] προέρχεται από τα ακρωνύμια των Cascading Style Sheets. Εντάσσεται στην κατηγορία γλωσσών σήμανσης και όπως και η html μπορεί να δημιουργηθεί με έναν απλό επεξεργαστή κειμένου με την κατάληξη αρχείου `.css`.

Ως γλώσσα σήμανσης αναπτύχθηκε γύρω στην δεκαετία του 90 για να προσθέσει διάφορα έξτρα χαρακτηριστικά που αφορούν την διαμόρφωση της ιστοσελίδας. Μεταξύ άλλων κάποια από τα χαρακτηριστικά που ορίζονται μέσω της CSS είναι τα χρώματα, η στοίχιση, οι γραμματοσειρές και πολλά άλλα που κάνουν την ιστοσελίδα πιο εύχρηστη και όμορφη.

Για να διαμορφωθεί στιλιστικά ένα html αρχείο θα πρέπει να εμπεριέχει την css. Αυτό γίνεται με 2 τρόπους:

1. Να δημιουργηθεί ξεχωριστό αρχείο και να καλείται από την html.
2. Να εμπεριέχεται στην σελίδα html με την ειδική ετικέτα `<style></style>`

Όπως είναι προφανές ο πρώτος τρόπος είναι πιο αποδοτικός καθώς δίνει την ευελιξία να επαναχρησιμοποιηθεί η στιλιστική διαμόρφωση και σε άλλες σελίδες.

Η διαμόρφωση είναι εξαιρετικά εύκολη καθώς αρκεί μόνο να αναφέρεις στην css ποιο στοιχείο της σελίδας θέλεις να διαμορφώσεις. Ο ορισμός στοιχείου γίνεται είτε με την αναφορά της ετικέτας είτε με επιλεγείς (selectors). Οι selectors είναι δυο, είναι η κλάση και το id. Η κύρια διαφορά τους είναι ότι η κλάση χρησιμοποιείται για να ομαδοποιήσει την μορφοποίηση και γ αυτόν τον λόγο το όνομα μιας κλάσης μπορεί να χρησιμοποιηθεί σε πολλά στοιχεία της HTML. Σε αντίθεση τα ids, αναφέρονται σε ένα μοναδικό στοιχείο. Αξίζει να σημειωθεί ότι ο browser στην στιλιστική διαμόρφωση δίνει μεγαλύτερη βαρύτητα στα Ids. Για παράδειγμα, αν ένα στοιχείο έχει και κλάση και id. Αν η κλάση ορίσει το χρώμα του κειμένου να είναι κόκκινο και το id έχει ορίσει να είναι μαύρο τότε στον browser το κείμενο θα εμφανιστεί μαύρο.

Τέλος, χρησιμοποιώντας μόνο html και css μπορούμε να κατασκευάσουμε απλές στατικές σελίδες η οποίες μπορούν να χρησιμοποιηθούν στον ιστιότοπο.

2.3. JavaScript

Η ιστορία της JavaScript [3] ξεκίνησε από τον δημιουργό της Brendan Eich της εταιρείας Netscape με την αρχική ονομασία Mocha. Αργότερα, Mocha άλλαξε σε LiveScript όπου τελικά λίγο αργότερα κατέληξε στην σημερινή ονομασία JavaScript. Το όνομα της επηρεάστηκε κυρίως από το γεγονός ότι η ανάπτυξη της καθώς και κάποια στοιχεία της γλώσσα επηρεάστηκαν από την γλώσσα προγραμματισμού Java. Λόγο της ονομασίας, συχνά συσχετίζουν την JavaScript με την Java αλλά πρακτικά οι δύο αυτές γλώσσες δεν συσχετίζονται καθόλου και έχουν πολύ διαφορετική σημασία.

Η JavaScript είναι client side γλώσσα δηλαδή η επεξεργασία και η εκτέλεση του κώδικα γίνεται στην μεριά του χρήστη. Στην συγκεκριμένη περίπτωση η εκτέλεση γίνεται στον browser. Το κύριο πλεονέκτημα είναι ότι δεν χρειάζεται να επικοινωνήσει με τον Server για να παράγει το τελικό αποτέλεσμα, αρκεί να τρέξει στον browser. Από την άλλη πλευρά το μειονέκτημα να τρέχει στην πλευρά του χρήστη είναι η ασφάλεια. Για παράδειγμα η ανάθεση πληροφορίας όπως κωδικοί θεωρείται επικίνδυνη καθώς από την στιγμή που εκτελείται και τρέχει στον browser μπορεί ανά πάσα στιγμή ο χρήστης να έχει πρόσβαση σε αυτόν τον κώδικα.

Τέλος, αυτό που έρχεται να προσφέρει η JavaScript στην κατασκευή ιστοσελίδων είναι δυναμικό περιεχόμενο, να ανταλλάσσονται τα δεδομένα ασύγχρονα και να έχει την δυνατότητα ο χρήστης να αλληλοεπιδρά με την ιστοσελίδα.

2.3.1. JQuery

Η jQuery [4] είναι μια βιβλιοθήκη JavaScript, εμφανίστηκε για πρώτη φορά τον Ιανουάριο του 2006 και ο δημιουργός της είναι ο Τζον Ρέριγκ (John Resig). Η βιβλιοθήκη έχει σχεδιαστεί έτσι ώστε να είναι γρήγορη, μικρή και κυρίως να απλοποιήσει την υλοποίηση σεναρίων. Με βάση το Wikipedia φαίνεται πως την βιβλιοθήκη jQuery την χρησιμοποιούν πάνω από το 65% των ιστοσελίδων με την μεγαλύτερη επισκεψιμότητα .

Για την υλοποίηση της εργασίας χρησιμοποιήθηκε η βιβλιοθήκη jQuery. Οι κύριοι λόγοι που συντέλεσαν στην απόφαση αυτή είναι:

- **Οι δυνατότητες που προσφέρει.** Ως βιβλιοθήκη διαθέτει έτοιμες λειτουργικότητες οι οποίες μπορούν να χρησιμοποιηθούν πολύ απλά και γρήγορα στην εφαρμογή.
- **Γρήγορη ανάπτυξη ιστοσελίδας.** Δεδομένου ότι έχει πολλές έτοιμες λειτουργικότητες αρκεί να τις προσαρμόσεις στις ανάγκες της λειτουργίας που υλοποιείς και θα έχεις το επιθυμητό αποτέλεσμα σε μικρό χρονικό διάστημα.

2.4. PHP

"Η ιστορία της PHP [5] ξεκινά από το 1994, όταν ένας φοιτητής, ο Rasmus Lerdorf δημιούργησε χρησιμοποιώντας τη γλώσσα προγραμματισμού C ένα απλό script με όνομα php.cgi, για προσωπική χρήση. Το script αυτό είχε σαν σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το OnLine βιογραφικό του σημείωμα. Αργότερα αυτό το script το διέθεσε και σε φίλους του, οι οποίοι άρχισαν να του ζητούν να προσθέσει περισσότερες δυνατότητες. Η γλώσσα τότε ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter. Το 1997 η PHP/FI έφθασε στην έκδοση 2.0 αριθμώντας περισσότερους από 50.000 ιστότοπους που τη χρησιμοποιούσαν, ενώ αργότερα την ίδια χρονιά οι Andi Gutmans και Zeev Suraski ξαναέγραψαν τη γλώσσα από την αρχή, βασιζόμενοι όμως αρκετά στην PHP/FI 2.0. "

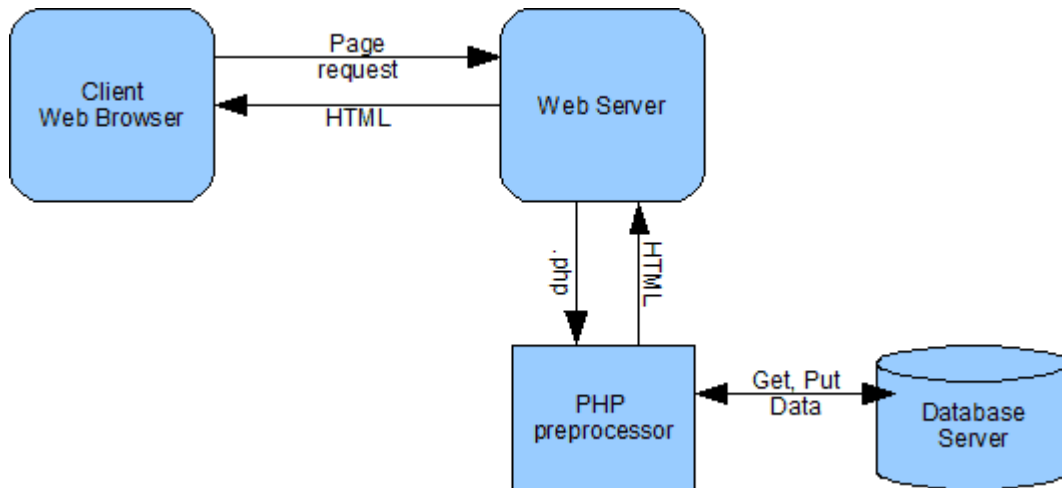
Προέλευση: PHP - <https://en.wikipedia.org>



Εικόνα 1 Logo PHP πηγή: <http://php.net/images/logos/php-med-trans.png>

20 χρόνια αργότερα η PHP αποτελεί μια από τις πιο διαδομένες [6] γλώσσες παγκοσμίως και χρησιμοποιείται στα μεγαλύτερα προτζεκτ. Πλέον η έκδοση της βρίσκεται στο 7.1.1 προσφέροντας νέες λειτουργικότητες με πολλά πλεονεκτήματα.

Η ανάγκη δημιουργίας δυναμικών ιστοσελίδων έκανε την PHP ακόμα πιο διαδεδομένη καθώς προσφέρει αυτήν την δυνατότητα. Κυρίως συνδέεται με web server(διακομιστή) όπως Apache ή Nginx. Όταν γίνει κάποιο αίτημα από την μεριά του client, ο Server δέχεται το αίτημα και ταυτόχρονα τρέχει και επεξεργάζεται το αντίστοιχο php αρχείο. Το αποτέλεσμα στέλνεται πίσω στον web server ως HTML και στην συνέχεια ο web server απαντάει στο αίτημα του client με την ίδια μορφοποίηση HTML.



Εικόνα 2 διάγραμμα ροής αιτήματος από τον client στον Server χρησιμοποιώντας PHP, πηγή: <https://i.stack.imgur.com/Lt5oY.gif>

Όπως φαίνεται και από το παραπάνω διάγραμμα για να εκτελεστεί σωστά το αίτημα θα πρέπει το PHP αρχείο να έχει έγκυρη επέκταση.(π.χ. .php, .php5, .phtml).

Τέλος η PHP είναι μια γλώσσα ανοιχτού κώδικα, αυτό σημαίνει ότι ο πηγαίος κώδικας διατίθεται δωρεάν στους χρήστες δίνοντας τους την δυνατότητα είτε να τον επεξεργαστούν είτε να συμβάλουν στην ανάπτυξη νέων λειτουργιών.

2.5. Ruby on Rails

Το Ruby on Rails [7], συχνά Rails ή RoR, είναι ένα πλαίσιο ανάπτυξης λογισμικού Ιστού ανοιχτού κώδικα για τη γλώσσα προγραμματισμού Ruby. Προορίζεται για χρήση σε συνδυασμό με ευέλικτες μεθοδολογίες ανάπτυξης, οι οποίες χρησιμοποιούνται από τους προγραμματιστές Ιστού για ταχεία ανάπτυξη λογισμικού .

2.5.1. Ruby

Η Ruby ως γλώσσα προγραμματισμού εμφανίστηκε το 1993. Ο δημιουργός της είναι ο Ιάπωνας Yukihiro "Matz" Matsumoto. Ο Matz σχεδίαζε την γλώσσα με τέτοιο τρόπο έτσι ώστε να είναι εύκολη στην χρήση αλλά και στην κατανόηση. Η σύνταξη της και οι εντολές της, που αποτελούνται από λέξεις που χρησιμοποιούμε στην καθημερινότητα, την καθιστούν ευανάγνωστη ακόμα και σε ανθρώπους που δεν γνωρίζουν προγραμματισμό.

Η Ruby έγινε ευρέως γνωστή στην κοινότητα των προγραμματισμών και άρχισε να χρησιμοποιείται πιο συχνά το 2004 όπου δημιουργήθηκε το πλαίσιο ανάπτυξης λογισμικού Rails το οποίο είναι βασισμένο στην Ruby.

2.5.2. Rails

Το Ruby on Rails [7] προήλθε από τη δουλειά του David Heinemeier Hansson στο Basecamp, ένα εργαλείο διαχείρισης Project από την εταιρεία 37signals.

Πρακτικά το Ruby on Rails πρόκειται για ένα πλαίσιο ανάπτυξης λογισμικού για γρήγορη και εύκολη δημιουργία εφαρμογών στον ιστιότοπο. Αυτό επιτυγχάνεται διότι περιλαμβάνει διάφορα εργαλεία που διευκολύνουν προγραμματιστικές εργασίες, όπως η δημιουργία σκελετών προγραμμάτων (scaffolding) που μπορεί να δημιουργήσει αυτόματα κάποια από τα μοντέλα (models) και τις όψεις (views) που χρειάζεται μια βασική σελίδα Ιστού.

Σε ένα πλαίσιο ανάπτυξης λογισμικού, τα επίπεδα που αφορούν την λειτουργικότητας και την εμφάνιση έχουν υλοποιηθεί με τέτοιο τρόπο ώστε να μπορούν να δουλεύουν ταυτόχρονα χωρίς ο προγραμματιστής να επαναλαμβάνει κώδικα. Το Ruby on Rails έχει βασιστεί σε αυτήν την αρχή

«Don't Repeat Yourself (DRY)» ή αλλιώς μην επαναλαμβάνεσαι όπως επίσης και στην αρχή «(Convention over Configuration, CoC)» ή «Σύμβαση έναντι Ρύθμισης».

Η βασική αρχή (DRY) έχει ως αποτέλεσμα η εφαρμογή να έχει λιγότερες γραμμές κώδικα . Αυτό καθιστά την εφαρμογή πιο εύκολη στην κατανόηση, στην διατήρηση αλλά και στην βελτίωση, επίσης παρατηρείται βελτίωση και στην ταχύτητα ανάπτυξης του λογισμικού. Η δεύτερη αρχή (Convention over Configuration, CoC) αποτελεί μια αρχή σχεδίασης, που βοηθάει στο να ελαχιστοποιεί τον αριθμό των αποφάσεων που πρέπει να πάρει ένας προγραμματιστής κερδίζοντας απλότητα αλλά χωρίς απαραίτητα να χάνεται η λειτουργικότητα. [8].

Τέλος, το Rails χρησιμοποιεί αρχιτεκτονική Model-View-Controller (MVC).

2.5.3. Model-View-Controller (MVC)

Η αρχιτεκτονική MVC είναι ένα πρότυπο σχεδίασης που χωρίζει την εφαρμογή σε 3 κατηγορίες. Το μοντέλο(model), τις όψεις(view) και τέλος τον ελεγκτή(Controller). Το κάθε ένα από αυτά είναι ξεχωριστό και αυτόνομο, παρόλα τα επικοινωνεί με τις υπόλοιπες κατηγορίες για την ομαλή λειτουργία. Αναλυτικότερα οι κατηγορίες έχουν ως εξής:

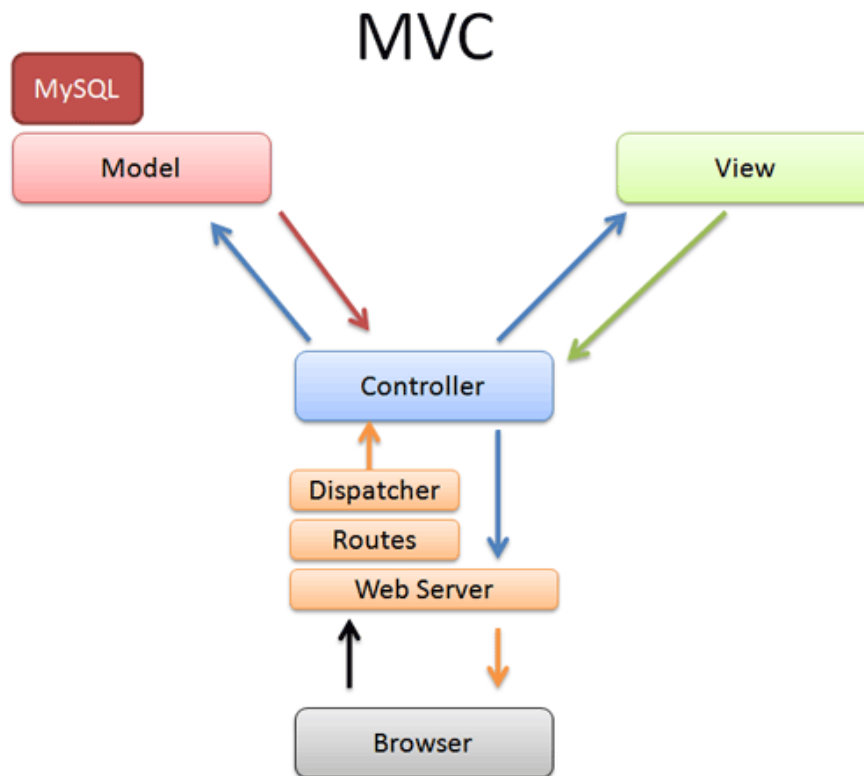
Μοντέλο: Η συγκεκριμένη κατηγορία συσχετίζεται με την αλληλεπίδραση της εφαρμογής με την βάση δεδομένων καθώς και για τις επικυρώσεις που απαιτούνται να πληρούν τα δεδομένα πριν αποθηκευτούν στην βάση δεδομένων. Για παράδειγμα, στο Rails για να διαβάσουμε δεδομένα από την βάση θα χρειαστεί να υλοποιήσουμε την λειτουργικότητα αυτή μέσα στο εκάστοτε μοντέλο. Αντίστοιχα στην περίπτωση που θέλουμε να ελέγχουμε την εγκυρότητα των δεδομένων που αποθηκεύονται στην βάση θα ανατρέξουμε στον κώδικα που βρίσκεται στο μοντέλο.

Όψεις: Η κατηγορία αυτή περιλαμβάνει οτιδήποτε εμφανίζεται στον χρήστη. Με λίγα λόγια για να επεξεργαστούμε την παρουσίαση της εφαρμογής μας στον χρήστη θα τροποποιήσουμε τον κώδικα που αφορούν τις όψεις.

Ελεγκτής: Έχει δύο βασικές λειτουργικότητες. Η πρώτη αφορά την επικοινωνία μοντέλου – όψεων, για να πραγματοποιηθεί μεσολαβεί ο ελεγκτής. Η δεύτερη βασική λειτουργικότητα είναι ότι αποτελεί την λογική της εφαρμογής. Μέσα στον ελεγκτή καθορίζεται τι θα εμφανιστεί και πως θα εμφανιστεί.

Το MVC λειτουργεί κυκλικά και συνήθως η ροή είναι όπως περιγράφεται παρακάτω:

- Ο χρήστης αλληλοεπιδρά με την εφαρμογή, για παράδειγμα να συμπληρώσει μια φόρμα εγγραφής.
- Ο ελεγκτής λαμβάνει τις παραμέτρους που έχει κατοχυρώσει ο χρήστης .
- Έπειτα τις στέλνει στο μοντέλο.
- Το μοντέλο ενημερώνει την βάση και στέλνει πίσω στον ελεγκτή απάντηση.
- Ανάλογα την απάντηση ο ελεγκτής ενημερώνει τις όψεις. Πχ με ένα μήνυμα.



Εικόνα 3 Διάγραμμα ροής mvc αρχιτεκτονικής. Πηγή: <https://betterexplained.com/wp-content/uploads/rails/mvc-rails.png>

2.6. Bootstrap

Το Bootstrap [9] αναπτύχθηκε από τον Mark Otto και τον Jacob Thornton στο Twitter. Κυκλοφόρησε για πρώτη φορά τον Αύγουστο του 2011 ως λογισμικό ανοιχτού κώδικα. Σε λιγότερο από ένα χρόνο, το Φεβρουάριο του 2012 καταφέρνει να γίνει το πιο δημοφιλές έργο ανάπτυξης στην πλατφόρμα συστήματος ελέγχου πηγαίου κώδικα github. Πλέον χρησιμοποιείται από εκατομμύρια ιστοσελίδες και είναι το πιο ευρέως γνωστό πλαίσιο ανάπτυξης λογισμικού για δημιουργία responsive ιστοσελίδων.

2.6.1. Responsive

Με τον όρο responsive, αναφερόμαστε στην δυνατότητα το σχέδιο της ιστοσελίδα να προσαρμόζεται δυναμικά και άμεσα σε κάθε διάσταση της οθόνης. Πρακτικά αυτό σημαίνει ότι όταν ο χρήστης επισκέπτεται την ιστοσελίδα από την κινητή του συσκευή να διαμορφώνεται με τέτοιο τρόπο έτσι ώστε να διευκολύνει την περιήγηση από την συσκευή.

Η ανάγκη για responsive σχεδίαση έχει αυξηθεί τα τελευταία χρόνια καθώς έχει παρατηρηθεί μεγάλη αύξηση στην χρήση κινητών συσκευών με διαφορετικές αναλύσεις. Πάραυτα τα οφέλη του είναι αρκετά. Τα κυριότερα είναι [10]:

- Διευκόλυνση του επισκέπτη της ιστοσελίδας
- Εύκολη ενημέρωση.
- Αύξηση χρηστικότητας και χρόνου παραμονής στην ιστοσελίδα
- Προσαρμοστικότητα
- Κόστος

2.6.2. Λόγοι επιλογής Bootstrap



Εικόνα 4 Logo Bootstrap, πηγή: <http://v4-alpha.getbootstrap.com/>

Η αύξηση των κινητών συσκευών έχει οδηγήσει στην υιοθέτηση νέων τεχνικών, όπως η responsive σχεδίαση. Όμως για να υλοποιηθεί αυτή η τεχνική απαιτεί πολύ χρόνο. Χρησιμοποιώντας ένα έτοιμο πλαίσιο ανάπτυξης λογισμικού όπως το Bootstrap όχι μόνο καταφέρνεις να κερδίσεις χρόνο από την υλοποίηση αλλά να βασιστείς σε σύγχρονα πρότυπα που έχουν ελεγχθεί. Πιο συγκεκριμένα οι λόγοι που οδήγησαν στην επιλογή του Bootstrap είναι οι εξής:

Ταχύτητα ανάπτυξης: Χωρίς αμφιβολία, είναι από τα μεγαλύτερα πλεονεκτήματα που προσφέρει. Αντί να γράφεις από την αρχή μέρη του κώδικα για να τις μορφοποιήσεις, το Bootstrap σου προσφέρει έτοιμα μπλοκ κώδικα που μπορείς να τα χρησιμοποιήσεις.

Responsive: Όπως ήδη αναφέραμε η ανάγκη προσαρμοστικότητας είναι μεγάλη. Την λύση στο συγκεκριμένο θέμα τίθεται να δοθεί από το Bootstrap. Χρησιμοποιώντας συγκεκριμένες κλάσεις μπορείς πολύ γρήγορα και εύκολα να προσαρμόσεις την σελίδα σε διάφορες αναλύσεις.

Προσαρμόζεται στις ανάγκες: Πέρα από το γεγονός ότι το Bootstrap παρέχει πολλά έτοιμα πράγματα, έχει και την δυνατότητα να προσαρμοστεί στις ανάγκες τις ιστοσελίδας. Μπορείς να επιλέξεις ποιες λειτουργίες να χρειάζονται στο προτζεκτ.

2.7. Βάση δεδομένων

Στην επιστήμη της πληροφορικής και στην καθημερινή χρήση των ηλεκτρονικών υπολογιστών, με τον όρο βάσεις δεδομένων [11] αναφερόμαστε σε οργανωμένες, διακριτές συλλογές σχετιζόμενων δεδομένων ηλεκτρονικά και ψηφιακά αποθηκευμένων, στο λογισμικό που χειρίζεται τέτοιες συλλογές (Σύστημα Διαχείρισης Βάσεων Δεδομένων, ή DBMS) και στο γνωστό πεδίο που το μελετά. Υπάρχουν διάφορες βάσεις δεδομένων όπως MySQL, PostgreSQL, Oracle.

2.7.1. PostgreSQL

Για την υλοποίηση την εργασίας χρησιμοποιούμε PostgreSQL. Πρόκειται για σχεσιακή βάση δεδομένων ανοιχτού κώδικα. Με τον όρο σχεσιακή βάση δεδομένων εννοείται μια συλλογή δεδομένων που είναι οργανωμένη σε πίνακες. Οι πίνακες αυτοί συσχετίζονται μεταξύ του και αυτό επιτυγχάνεται προσθέτοντας κοινό πεδίο στους πίνακες. Το χαρακτηριστικό αυτό είναι ιδιαίτερα ισχυρό καθώς κάνει την βάση ευέλικτη.

Πιο συγκεκριμένα η PostgreSQL έκανε την εμφάνιση της γύρω στο 1995 με την ονομασία Postgres95. Λίγο καιρό αργότερα η ονομασία αυτή δεν θα αντέξει και θα οριστικοποιηθεί στην τωρινή ονομασία PostgreSQL έτσι ώστε να αντικατοπτρίσει την σχέση ανάμεσα στην αρχική POSTGRES και τις πιο πρόσφατες εκδόσεις με συμβατότητα SQL. [12].

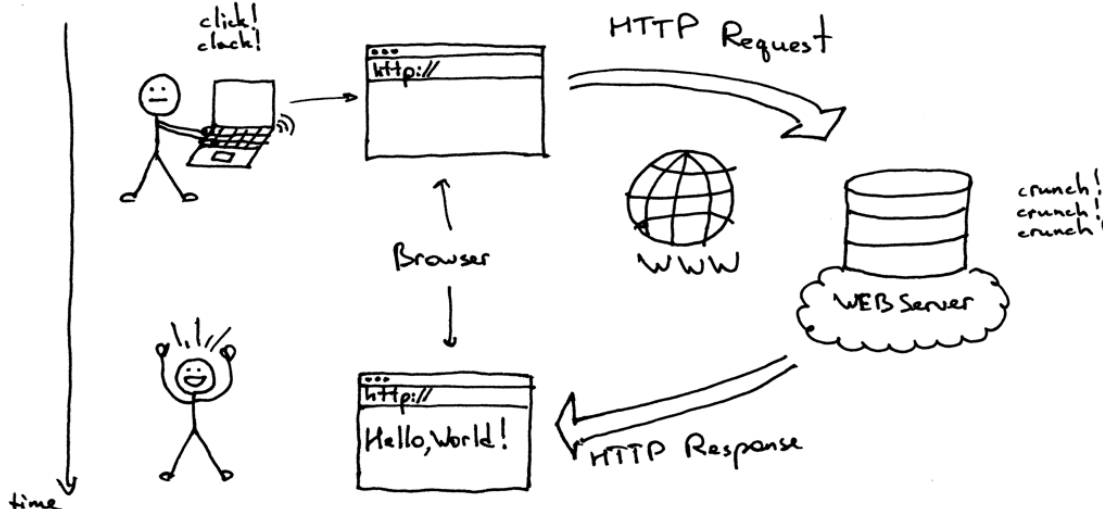
Το γεγονός ότι η αρχιτεκτονική της βασίζεται γύρω από την αξιοπιστία, την σωστή λειτουργία και την ακεραιότητα των δεδομένων συντελεί βασικό λόγο για την επιλογή της στην συγκεκριμένη εργασία. Σημαντικό ρόλος έπαιξε επίσης ότι διαθέτει περιβάλλοντα προγραμματισμού για την γλώσσα Ruby στην οποία βασίστηκε ολόκληρη η υλοποίηση.

2.8. Web Server

Εξυπηρετητής ή Server [13] όπως προσδίδει και το όνομα του είναι ο συνδυασμός software και hardware που κύρια λειτουργία του είναι η εξυπηρέτηση διάφορων αιτημάτων από άλλα προγράμματα. Τα άλλα προγράμματα είναι γνωστά και ως clients, στην περίπτωση του web, ως client μπορούμε να θεωρήσουμε τον browser. Η επικοινωνία τους βασίζεται στο πρωτόκολλο HTTP το οποίο από το 1989 που πρωτοεμφανίστηκε αποτελεί την κύρια συμφωνία κανόνων μεταξύ επικοινωνίας server-client.

2.6.1. Apache

Ο Apache είναι εξυπηρετητής του παγκόσμιου ιστού ή αλλιώς web server. Η πρώτη του έκδοση χρονολογείται το 1995 και έκτοτε αποτελεί έναν από τους δημοφιλέστερους στο είδος του. Ο Apache είναι λογισμικό ανοιχτού κώδικα και λειτουργεί σε διάφορες πλατφόρμες όπως τα Windows, mac, Linux.



Η λειτουργία του είναι βασική καθώς επιστρέφει στον χρήστη την ιστοσελίδα που ζήτησε. Για παράδειγμα όπως εμφανίζεται και στην εικόνα όταν ο χρήστης επισκεφτεί έναν ιστότοπο τότε μέσω του browser δημιουργεί ένα αίτημα. Ο browser επικοινωνεί με τον Apache μέσω του πρωτοκόλλου HTTP και αυτός με την σειρά του παράγει το αίτημα που ζητήθηκε και επιστρέφει την ιστοσελίδα στον browser.

Εικόνα 6 απεικόνιση λειτουργίας web server, πηγή: https://ruslanspivak.com/lsbaws-part1/LSBAWS_HTTP_request_response.png

2.6.2. Nginx



Ο Nginx ανήκει στην κατηγορία των Web Servers και πρωτοεμφανίστηκε το 2004. Ο δημιουργός του είναι ο Igor Sysoev και είχε ξεκινήσει την υλοποίηση του με σκοπό να δημιουργήσει έναν Web Server ο οποίος θα είναι πιο γρήγορος από τον Apache.

Η διαφορά του με τον Apache είναι ότι σερβίρει στατικά αρχεία, χρησιμοποιεί πολύ λιγότερη μνήμη από τον Apache και το σημαντικότερο είναι ότι μπορεί να διαχειριστεί 4 φορές περισσότερα αιτήματα από τους χρήστες ανά δευτερόλεπτο. Αυτό τον καθιστά ιδανικό για εφαρμογές που έχουν πολύ επισκεψιμότητα. Πάραυτα έχει το μειονέκτημα ότι σε σχέση με τον Apache δεν υπάρχει η δυνατότητα να παρακάμψεις ρυθμίσεις του συστήματος όπως η ανακατεύθυνση URL.

Συνοψίζοντας όλα τα παραπάνω καταλήξαμε στο συμπέρασμα ότι και οι δύο Web Servers θα ικανοποιούσαν τις απαιτήσεις της εφαρμογής. Η επιλογή του Nginx βασίστηκε στη μελλοντική χρήση καθώς θα μπορεί να χειριστεί αποδοτικότερα την αυξηθεί της επισκεψιμότητας καθώς και το

γεγονός ότι το μειονέκτημα που έχει για τις ρυθμίσεις του συστήματος δεν επηρεάζουν την υλοποίηση.

Εικόνα 7 logo nginx πηγή <https://cdn.wp.nginx.com/wp-content/uploads/2016/08/NGINX-Logo-smaller.jpg>

2.9. Google Maps API

“Οι Χάρτες Google [14] είναι υπηρεσία χαρτογράφησης στο Διαδίκτυο. Η εφαρμογή και η τεχνολογία της υπηρεσίας παρέχεται από την Google και υποστηρίζει πολλές υπηρεσίες που βασίζονται σε χάρτες, συμπεριλαμβανομένου της ιστοσελίδας "Χάρτες Google". Προσφέρει χάρτες δρόμων και σχεδιαστή διαδρομών για μεταφορές με τα πόδια, αυτοκίνητο, ποδήλατο (beta) ή μέσα μαζικής μεταφοράς. Οι δορυφορικές εικόνες των Χαρτών Google δεν ανανεώνονται σε πραγματικό χρόνο, ωστόσο η Google προσθέτει δεδομένα στη Κύρια Βάση δεδομένων της σε τακτική βάση και οι περισσότερες από τις εικόνες δεν είναι πάνω από τριών ετών.”

Προέλευση: Google Maps - <https://en.wikipedia.org>

Το google maps API δεν πρόκειται για γλώσσα προγραμματισμού ή κάποια πλατφόρμα αλλά πρόκειται για μια ακόμα επιπλέον υπηρεσία που προσφέρει η Google. Με τον όρο API εννοείται η Διεπαφή Προγραμματισμού Εφαρμογών (Application Programming Interface). Με την χρήση API μας επιτρέπει να αντλούμε δεδομένα από τρίτους, σε αυτήν την περίπτωση το google maps , χρησιμοποιώντας ένα σύνολο προκαθορισμένων διαδικασιών. Πρακτικά χρησιμοποιώντας αυτήν την υπηρεσία μας δίνει την δυνατότητα να εμφανίζουμε δυναμικά χάρτες καθώς και διάφορες διαδρομές.

3. Ανάλυση Προβλήματος

Στο κεφάλαιο που ακολουθεί θα παρουσιάσουμε την υλοποίηση της εφαρμογής καθώς και το σχεδιασμό μαζί τις πιθανές απαιτήσεις που εντοπίσαμε ότι έχει ολόκληρη η εφαρμογή. Όπως αναφερθήκαμε και στο πρώτο κεφάλαιο η σχεδίαση της εφαρμογής αποτελείται από την online ιστοσελίδα για τους χρήστες και την online ιστοσελίδα που είναι αποκλειστικά για τους οδηγούς ταξί. Στο πρώτο μέρος παρουσιάζονται οι λειτουργίες που θα πρέπει να πληρεί η εφαρμογή και τα υπόλοιπα μέρη παρουσιάζουν αναλυτικά την υλοποίηση για κάθε μια λειτουργία.

3.1. Σχεδιασμός Υλοποίησης

Πριν προχωρήσουμε στην υλοποίηση είναι απαραίτητο να σχεδιάσουμε και να προκαθορίσουμε τις βασικές λειτουργίες στις οποίες θα βασιστεί η εφαρμογή. Οι βασικές λειτουργίες είναι οι ακόλουθες.

- Δυνατότητα δημιουργίας λογαριασμού στην πλατφόρμα.
 - Ο χρήστης θα πρέπει να επιβεβαιώνει το e-mail του.
 - Ο χρήστης να μπορεί να κάνει ανάκτηση κωδικού.
- Δυνατότητα αναζήτησης ταξί με βάση τις προτιμήσεις του χρήστη.
 - Εμφάνιση διαδρομής στον χάρτη.
 - Υπολογισμός διαδρομής.
 - Υπολογισμός τιμής.
 - Υπολογισμός χρόνου
- Ενημέρωση χρήστη με δυναμικό e-mail με στοιχεία της κράτησης
- Ενημέρωση οδηγού για την κράτηση
- Δυνατότητα προβολής ιστορικού κρατήσεων.
- Δυνατότητα με προεπισκόπηση κράτησης.
- Φόρμα επικοινωνίας.
- Παρουσίαση υπηρεσιών.
- Παρουσίαση προτεινόμενων περιοχών για επίσκεψη.

3.2. Απαιτήσεις Συστήματος

Η απαιτήσεις του συστήματος από πλευράς του χρήστη είναι η σύνδεση στο ίντερνετ καθώς και να έχει εγκατεστημένο κάποιον browser στην προσωπική του συσκευή ή τον υπολογιστή. Όσον αναφορά τις απαιτήσεις του συστήματος από την πλευρά του server απαιτείται να είναι εγκατεστημένη η γλώσσα προγραμματισμού Ruby όπως επίσης και το framework RoR. Τέλος ο server θα πρέπει να υποστηρίζει την PostgreSQL ως βάση δεδομένων.

3.3. Υλοποίηση

Με βάση τις ανάγκες του Project δημιουργούμε τους πίνακες που είναι απαραίτητοι για την υλοποίηση. Οι πίνακες είναι οι εξής:

Ο πίνακας **users** αποθηκεύει όλη την πληροφορία που συσχετίζεται με τους χρήστες της πλατφόρμας, από το όνομα του μέχρι το πόσες φορές έχει συνδεθεί στην πλατφόρμα. Αναλυτικότερα τα πεδία που έχει ο πίνακας **users** φαίνονται στην εικόνα 8.

Column Name	Data Type
id	integer
email	varchar
encrypted_password	varchar
reset_password_token	varchar
reset_password_sent_at	timestamp
remember_created_at	timestamp
sign_in_count	integer
current_sign_in_at	timestamp
last_sign_in_at	timestamp
current_sign_in_ip	inet
last_sign_in_ip	inet
created_at	timestamp
updated_at	timestamp
confirmation_token	varchar
confirmed_at	timestamp
confirmation_sent_at	timestamp
first_name	varchar
last_name	varchar
phone	varchar
country_code	varchar

Εικόνα 8 ο πίνακας users

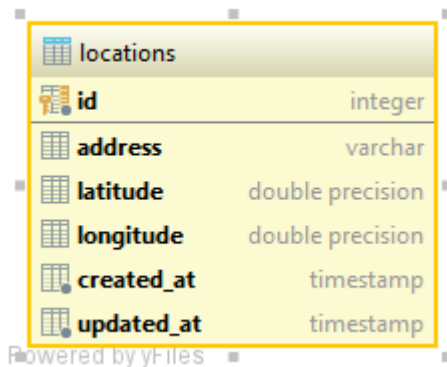
Ο επόμενος βασικός πίνακας είναι ο **bookings** όπως προσδίδει και το όνομα του αφορά τις κρατήσεις που έχουν γίνει στο σύστημα καθώς και την συσχέτιση με τον user που έκανε την κράτηση. Στη εικόνα 9 φαίνεται αναλυτικότερα τα πεδία που ανήκουν στον συγκεκριμένο πίνακα.

Field Name	Data Type
id	integer
name	varchar
last_name	varchar
pick_up	varchar
drop_off	varchar
adults	integer
children_5_11	integer
message	text
flight_no	text
created_at	timestamp
updated_at	timestamp
user_id	integer
email	varchar
pick_up_date	date
pick_up_time	time
return_date	date
return_time	time
return_pick_up	varchar
return_drop_off	varchar
children_2_4	integer
infants	integer
one_way	boolean
confirmation	boolean
confirmation_token	varchar

Εικόνα 9 Πίνακας bookings

Η συσχέτιση με τον user γίνεται με το πεδίο **user_id**. Όταν δημιουργείται μια κράτηση ενημερώνεται το συγκεκριμένο πεδίο με το Id του χρήστη που πραγματοποίησε την κράτηση.

Τέλος ο πίνακας **locations** έχει την πληροφορία σχετικά με τις τοποθεσίες. Στο συγκεκριμένο πίνακα αποθηκεύονται οι συντεταγμένες μια τοποθεσίας έτσι ώστε να διασφαλίσουμε την ακριβή τοποθεσία.



locations	
id	integer
address	varchar
latitude	double precision
longitude	double precision
created_at	timestamp
updated_at	timestamp

Εικόνα 10 Εικόνα 10 πίνακας locations.

Το επόμενο βήμα είναι η σύνδεση της εφαρμογής με την βάση. Στην **rails** αρκεί μόνο να τροποποιήσουμε τους κωδικούς στο αντίστοιχο αρχείο database.yml .

```
default: &default
  adapter: postgresql
  encoding: unicode
  pool: 5
  host: 127.0.0.1
  username: rails
  password: default_pass

development:
  <<: *default
  database: halkidiki_transfer_development
test:
  <<: *default
  database: halkidiki_ransfer_test

production:
  <<: *default
  database: halkidiki__transfer_production
  username: rails
  password: production_pass
  host: 127.0.0.1
```

Έχοντας ρυθμίσει την επικοινωνία της εφαρμογής με την βάση συνεχίζουμε στην ανάπτυξη της εφαρμογής. Τα βήματα που παρουσιάσουμε την υλοποίηση είναι τα εξής.

- Περιγράφουμε την λειτουργικότητα
- Παρουσιάζουμε την οθόνη που βλέπει ο χρήστης σε κινητή συσκευή καθώς και σε οθόνη υπολογιστή.
- Έπειτα αναλύουμε το κώδικα που αφορά τις όψεις (views) δηλαδή αυτό που βλέπει ο χρήστης.
- Αναλύουμε το κώδικα στον controller.
- Τέλος, αναλύουμε την συσχέτιση με το εκάστοτε model.

Δημιουργία λογαριασμού.

Στην συγκεκριμένη λειτουργία καλούμαστε να υλοποιήσουμε μια φόρμα όπου ο χρήστης θα συμπληρώνει τα απαραίτητα στοιχεία για την εγγραφή του στην πλατφόρμα. Στην εικόνα 11 φαίνεται η οθόνη που βλέπει ο χρήστης.

Εικόνα 11 φόρμα εγγραφής, πηγή http://www.halkidikitransfer.com/users/sign_up

Για την υλοποίηση της συγκεκριμένης οθόνης χρησιμοποιήθηκε ο παρακάτω κώδικας

```

1 <div class="registration col-lg-4 col-lg-push-4">
2
3 <h2>Sign up</h2>
4
5 <div class="signup-top-text">
6 <div class="signup-top-text-title">
7   Sign Up to Ride
8 </div>
9 <div class="signup-top-text-subtitle">
10 <p>Welcome to Halkidiki Transfer, the easiest way to get around at the tap of a button.</p>
11 <p>Create your account and get moving in minutes.</p>
12 </div>
13 </div>
14
15 <%= form_for(resource, as: resource_name, url: registration_path(resource_name)) do |f| %>
16 <%= devise_error_messages! %>
17 <h2>Account</h2>
18
19 <div class="field form-group">
20 <%= f.label :email %><br />
21 <%= f.email_field :email, autofocus: true, class: "form-control" %>
22 </div>
23
24 <div class="row">
25
26 <div class="field form-group col-lg-6">

```

```

27 <%= f.label :password %>
28 <% if @minimum_password_length %>
29 <!-- <em><%= @minimum_password_length %> characters minimum</em> -->
30 <% end %><br />
31 <%= f.password_field :password, autocomplete: "off", class: "form-control" %>
32 </div>
33
34 <div class="field form-group col-lg-6">
35 <%= f.label :password_confirmation %><br />
36 <%= f.password_field :password_confirmation, autocomplete: "off", class: "form-control" %>
37 </div>
38 </div>
39
40 <h2>Profile</h2>
41 <div class="row">
42
43 <div class="field form-group col-lg-6">
44 <%= f.label "Name" %><br />
45 <%= f.text_field :first_name, autofocus: true, class: "form-control" %>
46 </div>
47
48 <div class="field form-group col-lg-6">
49 <%= f.label :last_name %><br />
50 <%= f.text_field :last_name, autofocus: true, class: "form-control" %>
51 </div>
52 </div>
53
54 <div class="row">
55 <div class="field form-group col-xs-3">
56 <%= f.label :code %><br />
57 <%= f.text_field :country_code, autofocus: true, class: "form-control" %>
58 </div>
59
60 <div class="field form-group col-xs-9">
61 <%= f.label :phone %><br />
62 <%= f.telephone_field :phone, autofocus: true, class: "form-control" %>
63 </div>
64 </div>
65
66 <div class="actions">
67 <%= f.submit "CREATE ACCOUNT", class: "btn btn-gold cta" %>
68 </div>
69 <% end %>
70
71 <%= render "users/shared/links" %>
72 </div>

```

Όπως μπορείτε να παρατηρήσετε ο παραπάνω κώδικας δεν έχει μόνο στοιχεία από html αλλά και από την rails. Οπότε για να ενσωματώσουμε την rails στην html χρησιμοποιούμε τα σύμβολα `<% %>` όπου ενδιάμεσα προσθέτουμε τις εντολές. Αντίστοιχα προσθέτοντας το σύμβολο `(=) <%= %>` ενημερώνουμε τον browser πως ότι βρίσκεται ανάμεσα θέλουμε να εμφανιστεί στον χρήστη. Τέλος χρησιμοποιείται ο helper `form_for` της rails που παράγει την αντίστοιχη form στην html.

Όταν ο χρήστης πατήσει το create account οι παράμετροι που έχει προσθέσει ο χρήστης θα σταλούν στον controller όπου θα τα διαχειριστεί και θα τα στείλει στο model και θα αποθηκευτούν.

Συγκεκριμένα ο controller που χειρίζεται την εγγραφή των χρηστών είναι ο παρακάτω:

```
1 class Devise::RegistrationsController < DeviseController
2   prepend_before_action :require_no_authentication, only: [:new, :create, :cancel]
3   prepend_before_action :authenticate_scope!, only: [:edit, :update, :destroy]
4   prepend_before_action :set_minimum_password_length, only: [:new, :edit]
5
6   # POST /resource
7   def create
8     build_resource(sign_up_params)
9
10    resource.save
11    yield resource if block_given?
12    if resource.persisted?
13      if resource.active_for_authentication?
14        set_flash_message! :notice, :signed_up
15        sign_up(resource_name, resource)
16        respond_with resource, location: after_sign_up_path_for(resource)
17      else
18        set_flash_message! :notice, :signed_up_but_#{resource.inactive_message}"
19        expire_data_after_sign_in!
20        respond_with resource, location: after_inactive_sign_up_path_for(resource)
21      end
22    else
23      clean_up_passwords resource
24      set_minimum_password_length
25      respond_with resource
26    end
27  end
28
29  protected
30
31  # Build a devise resource passing in the session. Useful to move
32  # temporary session data to the newly created user.
33  def build_resource(hash=nil)
34    self.resource = resource_class.new_with_session(hash || {}, session)
35  end
36
37  # Signs in a user on sign up. You can overwrite this method in your own
38  # RegistrationsController.
39  def sign_up(resource_name, resource)
40    sign_in(resource_name, resource)
41  end
42
43  # The path used after sign up. You need to overwrite this method
44  # in your own RegistrationsController.
45  def after_sign_up_path_for(resource)
46    after_sign_in_path_for(resource)
```



```

47 end
48
49 # The path used after sign up for inactive accounts. You need to overwrite
50 # this method in your own RegistrationsController.
51 def after_inactive_sign_up_path_for(resource)
52   scope = Devise::Mapping.find_scope!(resource)
53   router_name = Devise.mappings[scope].router_name
54   context = router_name ? send(router_name) : self
55   context.respond_to?(:root_path) ? context.root_path : "/"
56 end
57
58 # Authenticates the current scope and gets the current resource from the session.
59 def authenticate_scope!
60   send(:"authenticate_#{resource_name}!", force: true)
61   self.resource = send(:"current_#{resource_name}")
62 end
63
64 def sign_up_params
65   devise_parameter_sanitizer.sanitize(:sign_up)
66 end
67
68 end

```

Περίληπτικά τρέχει η μέθοδος create κάνει τους απαραίτητους ελέγχους και εφόσον πληρούνται όλες οι προϋποθέσεις αποθηκεύει τον χρήστη, τον ανακατευθύνει και του εμφανίζει ενημερωτικό μήνυμα . Σε οποιαδήποτε άλλη περίπτωση του εμφανίζει αντίστοιχο ενημερωτικό μήνυμα για την αποτυχία δημιουργίας λογαριασμού.

Όπως είχαμε αναφέρει, τους ελέγχους που πρέπει να γίνουν πριν αποθηκευτεί οποιαδήποτε πληροφορία στην βάση γίνονται στο μοντέλο. Ο κώδικας που περιλαμβάνει τους ελέγχους και όχι μόνο για τον πίνακα users είναι ο παρακάτω:

```

1 class User < ActiveRecord::Base
2   # Include default devise modules. Others available are:
3   # :confirmable, :lockable, :timeoutable and :omniauthable
4   after_create :send_welcome_email
5   devise :database_authenticatable, :registerable,
6         :recoverable, :rememberable, :trackable, :validatable, :confirmable
7   has_many :bookings
8
9   private
10
11   def send_welcome_email
12     UserMailer.welcome_email(self).deliver
13   end
14 end

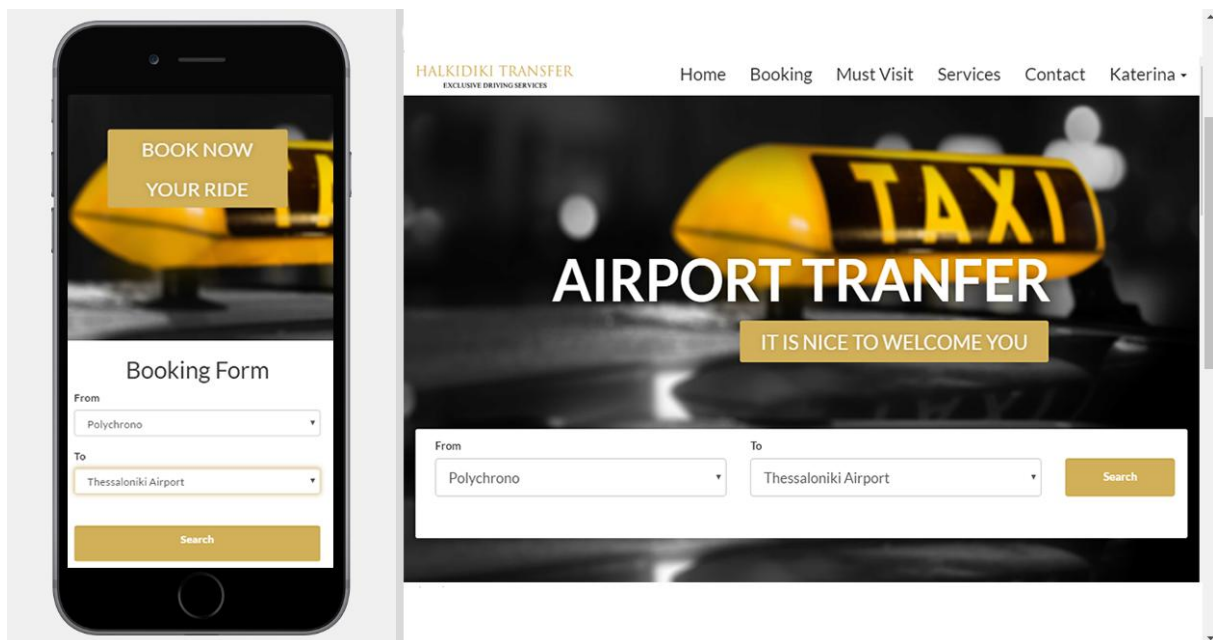
```

Για παράδειγμα το validatable είναι έτοιμο module όπου ελέγχει όχι μόνο αν ο χρήστης έχει προσθέσει το e-mail του στην φόρμα εγγραφής αλλά ελέγχει αν είναι μοναδικό και έχει το σωστό μορμάτ. Τέλος ελέγχει την παρουσία κωδικού, το μέγεθος του και αν ταυτίζεται με το πεδίο όπου ο χρήστης επιβεβαιώνει τον κωδικό. Οπότε αν ένα από τα παραπάνω δεν περάσουν τον έλεγχο τότε θα εμφανιστεί μήνυμα στον χρήστη και δεν θα ολοκληρωθεί η εγγραφή. Αντίστοιχους ελέγχους έχουν και τα υπόλοιπα module.

Στο model δεν πραγματοποιούνται μόνο έλεγχοι αλλά όπως βλέπετε με την μέθοδο **send_welcome_email** στέλνουμε e-mail στον χρήστη. Αξίζει να σημειωθεί ότι η rails μας δίνει την δυνατότητα να χρησιμοποιήσουμε το callback `after_create`. Πρακτικά αυτό που μας προσφέρει είναι να στείλουμε το e-mail **εφόσον** έχει δημιουργηθεί ο λογαριασμός διότι σε οποιαδήποτε άλλη περίπτωση η αποστολή e-mail είναι λανθασμένη. Τέλος η συσχέτιση του χρήστη με την κράτηση γίνεται στην **γραμμή 7**, δεν θα σταθούμε περισσότερο στην συγκεκριμένη γραμμή τώρα αλλά όταν θα αναλύσουμε τον μοντέλο bookings.

Αναζήτηση/κράτηση ταξί

Πρόκειται για την βασική λειτουργία της εφαρμογής. Οι χρήστες να έχουν την δυνατότητα να αναζητήσουν ταξί με βάση τις προτιμήσεις τους καθώς και να προχωρήσουν στην κράτηση εφόσον το επιθυμούν.



Εικόνα 12 η φόρμα κράτηση όπως εμφανίζεται στους χρήστες, πηγή <http://www.halkidikitransfer.com/>

Ο κώδικας για το συγκεκριμένο view είναι:

```
1 <div class="content booking-form">
2
3 <!-- <div class="container-fluid"> -->
4 <div class="device-full-width">
5   <%= form_for @booking, id: "msform", url: booking_index_path do |f| %>
6     <fieldset class="destinations col-md-10 col-xs-12">
7       <h2 class="visible-xs booking_header"> Booking Form </h2>
8
9       <div class="form-group col-md-5 col-sm-5">
10        <%= f.label "From" %>
11        <%= f.select :pick_up, options_for_select(@location.map {|l| [l.cached_locations]}), {}, class: "form-
control" %>
12      </div>
13
14      <div class="form-group col-md-5 col-sm-5">
```

```

15     <%= f.label "To" %>
16     <%= @destination_list %>
17     <%= f.select (:drop_off), options_for_select(@location.map {|l| [l. cached_locations]}),{:required =>
true}, {class => "form-control"}%>
18     </div>
19
20     <div class="form-group col-md-2 col-sm-2 col-xs-12 search">
21     <%= f.label :search, :required => true, class: "sr-only" %>
22     <p class="search form-control btn btn-gold margin_top btn-full"> <span
class="search_centerize">Search</span></p>
23     </div>
24
25     <div class="form-group col-md-2 booking_flight_no">
26     <%= f.label :flight_no %>
27     <%= f.text_field :flight_no, class: "form-control" %>
28     </div>
29 </fieldset>
30
31 <fieldset class="booking_details col-xs-12">
32
33     <div class="form-group col-sm-3 col-xs-6">
34     <%= f.label "Departure Date" %>
35     <div class="input-group date" id="pick_up_date">
36     <%= f.text_field :pick_up_date.to_s, :required => true, class: "form-control" %>
37     <span class="input-group-addon"><span class="glyphicon glyphicon-calendar"></span></span>
38     </div>
39     </div>
40
41     <div class="form-group col-sm-2 col-xs-6">
42     <%= f.label "Departure Time" %>
43     <div class="input-group date" id="pick_up_time">
44     <%= f.text_field :pick_up_time, :required => true, class: "form-control" %>
45     <span class="input-group-addon"><span class="glyphicon glyphicon-time"></span></span></span>
46     </div>
47     </div>
48
49     <!-- return date and time -->
50     <div id="return_datetime">
51     <div class="form-group col-sm-3 col-xs-6">
52     <%= f.label "Return Date" %>
53     <div class="input-group date" id="return_date">
54     <%= f.text_field :return_date.to_s, class: "form-control" %>
55     <span class="input-group-addon"><span class="glyphicon glyphicon-calendar"></span></span></span>
56     </div>
57     </div>
58
59     <div class="form-group col-sm-2 col-xs-6">
60     <%= f.label "Return Time" %>
61     <div class="input-group date" id="return_time">
62     <%= f.text_field :return_time, :required => true, class: "form-control" %>
63     <span class="input-group-addon"><span class="glyphicon glyphicon-time"></span></span></span>
64     </div>
65     </div>
66
67     </div>
68
69     <div class="col-sm-2 pull-right col-xs-12">

```

```

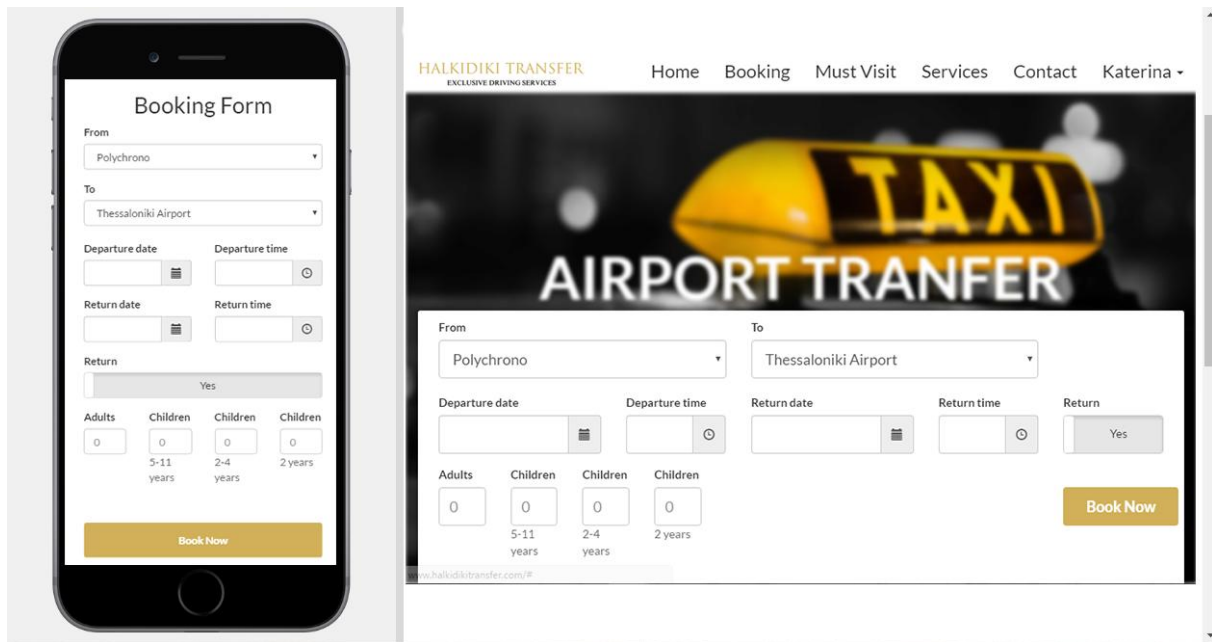
70
71   <div class="form-group">
72     <div class="">
73       <%= f.label "Return", class: "control-label"%>
74       <%= f.check_box :one_way, class:"form-control input-xl", data: {toggle => "toggle", :width => "100%",
:off => "Yes", :on => "No"} %>
75     </div>
76   </div>
77 </div>
78
79 <div class="row">
80   <div class="col-sm-12 content-position">
81     <div class="col-sm-5 row content-position">
82       <div class="form-group col-xs-3">
83         <%= f.label :adults, class: "control-label"%>
84         <%= f.number_field :adults, min: 0, placeholder: 0, :required => true, class: "form-control input-xl col-
sm-8 col-xs-6" %>
85       </div>
86
87       <div class="form-group input-xl col-xs-3">
88         <%= f.label :children, class: "control-label"%>
89         <%= f.number_field :children_5_11, min: 0, placeholder: 0, class: "form-control col-sm-8 col-xs-6" %>
90         <p class="help-block">5-11 years</p>
91
92       </div>
93
94       <div class="form-group input-xl col-xs-3">
95         <%= f.label :children, class: "control-label"%>
96         <%= f.number_field :children_2_4, min: 0, placeholder: 0, class: "form-control col-sm-8 col-xs-6" %>
97         <p class="help-block">2-4 years</p>
98
99       </div>
100
101      <div class="form-group input-xl col-xs-3">
102        <%= f.label :children, class: "control-label"%>
103        <%= f.number_field :infants, min: 0, placeholder: 0, class: "form-control col-sm-8 col-xs-6" %>
104        <p class="help-block">2 years</p>
105
106      </div>
107    </div>
108    <div class="col-sm-2 col-xs-12 pull-right">
109      <%= f.label :book_now, class: "control-label sr-only"%>
110      <%= f.submit t('buttons.book_now'), min: 0, :class => 'btn btn-gold margin_top btn-full' %>
111    </div>
112  </div>
113 </div>
114 </fieldset>
115 <% end %>
116 <!-- End of Form -->
117 </div>
118 </div>

```

Χρησιμοποιείται η ίδια λογική όπως και στην φόρμα εγγραφής. Η κύρια διαφορά είναι η χρήση του κώδικα `@location`. Στην rails χρησιμοποιώντας το σύμβολο `@` μπροστά από μια μεταβλητή μας δίνει την δυνατότητα να έχουμε πρόσβαση σε αυτή την μεταβλητή η οποία έχει οριστεί στον controller. Η

συγκεκριμένη μεταβλητή όπως θα δούμε και παρακάτω δεν είναι τίποτα παραπάνω από τις τοποθεσίες που έχουμε αποθηκευμένες στον αντίστοιχο πίνακα στην βάση μας.

Επίσης στην συγκεκριμένη σελίδα για να επιτύχουμε το εφέ [εικόνα 13] που εμφανίζει επιπλέον πεδία όταν ο χρήστης πατήσει το κουμπί αναζήτηση χρησιμοποιήσαμε JavaScript και πιο συγκεκριμένα το jQuery. Επιπλέον για να κάνουμε φιλικό προς το χρήστη την προσθήκη ημερομηνίας χρησιμοποιήσαμε σε αυτά τα πεδία date pickers. Είναι έτοιμη λειτουργία που μας παρέχει το jQuery όπου αντί ο χρήστης να γράφει την ημερομηνία, εμφανίζουμε σε μορφή ημερολογίου τις επιλογές που έχει.



Εικόνα 13 φόρμα αναζήτησης ταξί με τα επιπλέον πεδία, πηγή <http://www.halkidikitransfer.com/>

```

1 $("#booking_pick_up_date").datetimepicker({
2   format:'Y-m-d',
3   timepicker: false,
4   minDate: 0
5 });
6
7 $("#booking_pick_up_time").datetimepicker({
8   format:'H:i',
9   datepicker: false
10 });
11
12
13 $("#booking_return_date").datetimepicker({
14   format:'Y-m-d',
15   timepicker: false,
16   minDate: 0
17 });
18
19 $("#booking_return_time").datetimepicker({
20   datepicker: false,
21   format:'H:i'
22 });
23
24 $("#booking_pick_up_date").on("change", function () {
25   var minD = $('#booking_pick_up_date').val();
26   $('#booking_return_date').datetimepicker({
27     format:'Y-m-d',

```

```

28   timepicker: false,
29   minDate: minD
30 });
31 });
32
33 $(".booking_flight_no").hide();
34 $(".booking_details").hide();
35 removeSelectedOption();
36
37 $(".search").click(function(e){
38   $(".img-responsive").css('filter', 'blur(2px)');
39
40   $(".destinations").animate({marginTop: -140},
41     800, //time to complete animation
42     function () { // when destination animation completed
43       $(".search").fadeOut(1000);
44       bookingDetails = $(".booking_details");
45       bookingDetails.animate({marginTop: -20});
46       bookingDetails.slideDown();
47       checkReturn();
48       displayFlight();
49     }
50   );
51
52   function displayFlight () {
53
54     var from = $("#booking_pick_up");
55     from.trigger( 'change' );
56
57     from.change(function() {
58       fromVal = from.val();
59       if (fromVal == "Thessaloniki Airport") {
60         $(".booking_flight_no").fadeIn(1000);
61         $("input#booking_flight_no").attr('required', true);
62
63         // $(".input").attr("required", true);
64       } else {
65         $("input#booking_flight_no").attr('required', false);
66         $(".booking_flight_no").fadeOut(1000);
67       }
68     });
69
70   }
71
72   function checkReturn() {
73     $("#booking_one_way").change(function (e){
74       if ( $("#booking_one_way").prop( "checked" ) ) {
75         $("#return_datetime").fadeOut();
76         $("#return_datetime input").attr('required', false);
77       } else {
78         $("#return_datetime").fadeIn();
79         $("#return_datetime input").attr('required', true);
80       }
81     });
82   });
83
84 }
85
86
87 });
88
89 function removeSelectedOption() {
90   $("#booking_pick_up").change(function(){

```

```

91  var selectedOption = $(this).val();
92  $("#booking_drop_off option[value = '"+selectedOption+"']").remove();
93
94  $('#booking_pick_up').change(function(){
95    AddPreviousSelection(selectedOption);
96  });
97
98  });
99  }
100
101  function AddPreviousSelection(selection) {
102    if ($("#booking_drop_off option[value = '" + selection +']").length == 0){
103      $("#booking_drop_off").append("<option></option>").val(selection).html(selection);
104    };
105  }
106  }

```

Εφόσον ο χρήστης έχει συμπληρώσει τα πεδία με βάση τις προτιμήσεις του έχει την δυνατότητα να προχωρήσει στο επόμενο στάδιο. Στο επόμενο στάδιο γίνεται μια προεπισκόπηση των επιλογών του και του παρέχονται επιπλέον πληροφορίες. Όπως η τιμή, η διαδρομή στον χάρτη κτλ.

Την ανακατεύθυνση του χρήστη στο επόμενο στάδιο την αναλαμβάνει ο controller και πιο συγκεκριμένα οι παρακάτω γραμμές κώδικα.

```

1  class BookingController < ApplicationController
2
3  def create
4    user = current_user
5    @booking = Booking.new(booking_params)
6    if (@booking.one_way == false)
7      swap_destinations(@booking)
8    else
9      @booking.return_time = nil
10     @booking.return_date = nil
11   end
12
13   @booking.save
14   redirect_to continue_booking_path(@booking)
15 end
16
17
18 private
19 def booking_params
20   params.require(:booking).permit(
21     :pick_up,
22     :drop_off,
23     :pick_up_time,
24     :pick_up_date,
25     :return_time,
26     :return_date,
27     :adults,
28     :children_5_11,
29     :children_2_4,
30     :infants,

```

```

31     :one_way,
32     :confirmation,
33     :flight_no
34   )
35 end
36
37 private
38
39 def swap_destinations(booking)
40   booking.return_pick_up, booking.return_drop_off = booking.drop_off, booking.pick_up
41 end
42
43 end

```

Η μέθοδος create τρέχει από την στιγμή που ο χρήστης θα πατήσει **Book Now**. Έπειτα στην γραμμή 5 δημιουργείται ένα νέο στιγμιότυπο του μοντέλου bookings. Με την μέθοδο **booking_params** διασφαλίζουμε την ασφάλεια της εφαρμογής καθώς ενημερώνουμε τον controller ότι μόνο αυτοί οι παράμετροι επιτρέπονται να σταλθούν από την φόρμα κράτησης έτσι αποτρέπουμε κακόβουλες προσπάθειες. Τέλος, αφού κάνουμε διάφορους ελέγχους αποθηκεύουμε στην βάση τα στοιχεία της κράτησης και ανακατευθύνουμε το χρήστη στην επόμενη σελίδα. (γραμμή 14.)

Όπως έχουμε αναφέρει αρκετές φορές για να αποθηκευτεί κάτι στην βάση θα πρέπει να πληρεί κάποιους ελέγχους. Οι έλεγχοι που έχουμε ορίσει στο μοντέλο booking είναι οι παρακάτω:

```

1 class Booking < ActiveRecord::Base
2   belongs_to :user
3
4   validates :return_date, :return_time, :return_drop_off, :return_pick_up, absence: true, if: :one_way
5
6
7   def create
8     if valid?
9       @booking.save
10    end
11  end
12
13 end

```

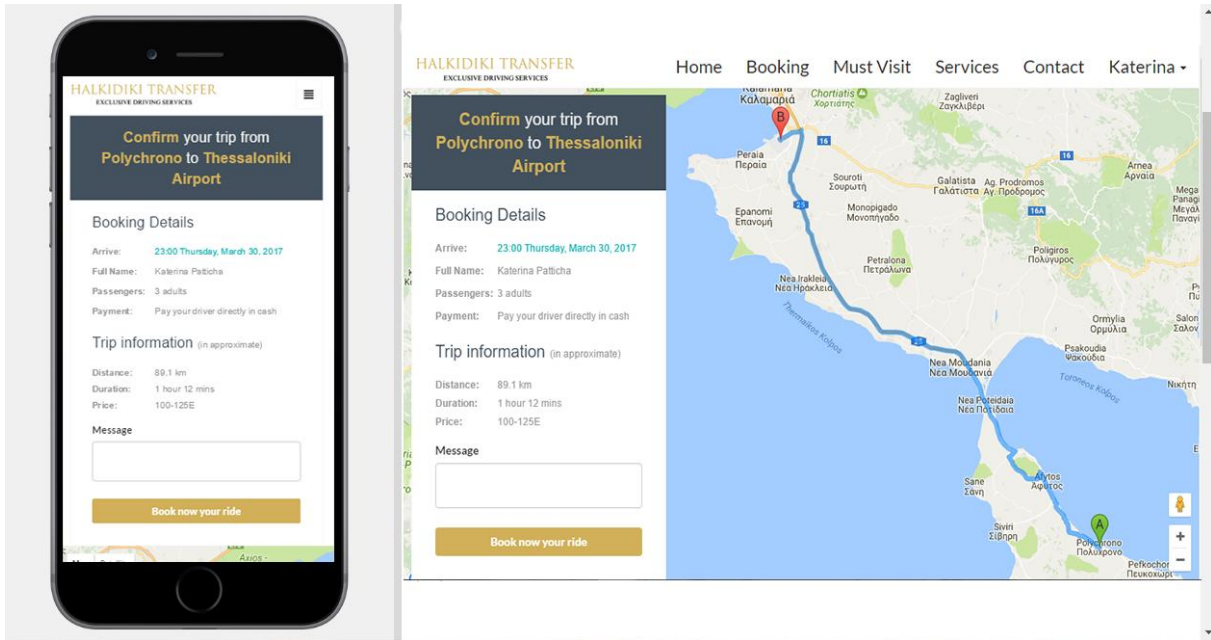
Ο μοναδικός έλεγχος που έχουμε ορίσει είναι ότι τα πεδία που αφορούν την διαδρομή επιστροφής επιτρέπουμε να είναι χωρίς τιμή **μόνο εφόσον** ο χρήστης επιλέξει η διαδρομή να είναι μονή(one way).

Στο σημείο αυτό αξίζει να αναφέρουμε την συσχέτιση των κρατήσεων με τον χρήστη.

Στην γραμμή 2 δηλώνουμε ότι η κράτηση ανήκει σε έναν user ενώ στο μοντέλο του user είχαμε ορίσει *has_many :booking*.

Με αυτό το τρόπο δηλώνουμε την σχέση **ένα-προς-πολλά** όπου ο **χρήστης** μπορεί να **έχει πολλές κρατήσεις** αλλά κάθε **κράτηση** μπορεί να **ανήκει μόνο σε έναν χρήστη**.

Τέλος αφού έχουν δρομολογηθεί όλα τα παραπάνω ο χρήστης βλέπει την παρακάτω οθόνη.



Εικόνα 14 Στάδιο επιβεβαίωση κράτησης, πηγή <http://www.halkidikitransfer.com/booking/151/continue>

Η συγκεκριμένη οθόνη είναι η πιο περίπλοκη και για την υλοποίηση της όπως θα δείτε παρακάτω χρησιμοποιήθηκαν αρκετές γραμμές κώδικα

```

1 <div class="col-lg-4 confirm_booking">
2 <div class="tables">
3 <table border="0" cellpadding="0" cellspacing="0" width="100%">
4 <tr>
5 <td bgcolor="#EBEBEB">
6 <div align="center">
7 <table width="100%" border="0" cellpadding="0" cellspacing="0" class="responsive-table">
8 <!-- LOGO/PREHEADER TEXT -->
9 <tr>
10 <td>
11 <table width="100%" border="0" cellspacing="0" cellpadding="0">
12 <td bgcolor="#43515E" class="pad-header">
13 <table width="100%" border="0" cellspacing="0" cellpadding="0">
14 <tr>
15 <td align="center" class="pad-header-copy">
16 <span class="confirmation_hd_destination">Confirm</span> your trip from <span
class="confirmation_hd_destination"> <%= @booking.pick_up%> </span> to <span
class="confirmation_hd_destination"> <%= @booking.drop_off %></span>
17 </td>
18 </tr>
19 <tr>
20 </tr>
21 </table>
22 </td>
23 </tr>
24 </table>
25 </td>
26 </tr>
27 </table>
28 </div>

```

```

29 </td>
30 </tr>
31 </table>
32
33 <!-- RESERVATION DETAILS -->
34 <table border="0" cellpadding="0" cellspacing="0" width="100%">
35 <tr>
36 <td bgcolor="#EBEBEB" align="center">
37 <table width="100%" border="0" cellpadding="0" cellspacing="0" class="responsive-table">
38 <tr>
39 <td bgcolor="#ffffff">
40 <table width="100%" border="0" cellspacing="0" cellpadding="0">
41 <tr>
42 <td >
43 <table width="100%" border="0" cellspacing="0" cellpadding="0">
44 <tr>
45 <td align="left" >
46 Booking Details
47 </td>
48 </tr>
49 <!-- CHECK-IN -->
50 <tr>
51 <td>
52 <table cellspacing="0" cellpadding="0" border="0" width="100%">
53 <tr>
54 <td valign="top">
55 <!-- LEFT COLUMN -->
56 <table cellpadding="0" cellspacing="0" border="0" width="26%" align="left"
class="responsive-table">
57 <tr>
58 <td align="center" bgcolor="#ffffff" class="flex-p-bold">
59 Arrive:
60 </td>
61 </tr>
62 </table>
63 <!-- RIGHT COLUMN -->
64 <table cellpadding="0" cellspacing="0" border="0" width="70%" align="right"
class="responsive-table">
65 <tr>
66 <td align="center" bgcolor="#ffffff"
class="flex-p">
67 <%= @booking.pick_up_time.strftime('%H:%M') %>
68 <%= @booking.pick_up_date.strftime("%A, %B %d, %Y") %>
69 </td>
70 </tr>
71 </table>
72 </table>
73 </td>
74 </tr>
75 </table>
76 </td>
77 </tr>
78 <!-- return table -->
79 <% if !@booking.one_way %>
80 <tr>
81 <td>
82 <table cellspacing="0" cellpadding="0" border="0" width="100%">
83 <tr>

```

```

84         <td valign="top">
85         <!-- LEFT COLUMN -->
86         <table cellpadding="0" cellspacing="0" border="0" width="26%" align="left"
class="responsive-table">
87         <tr>
88         <td align="center" bgcolor="#ffffff" class="flex-p-bold">
89             Return:
90         </td>
91         </tr>
92         </table>
93         <!-- RIGHT COLUMN -->
94         <table cellpadding="0" cellspacing="0" border="0" width="70%" align="right"
class="responsive-table">
95         <tr>
96         <td align="center" bgcolor="#ffffff"
97             class="flex-p">
98             <%= @booking.return_time.strftime('%H:%M') %>
99             <%= @booking.return_date.strftime("%A, %B %d, %Y") %>
100        </td>
101        </tr>
102        </table>
103        </td>
104        </tr>
105        </table>
106        </td>
107        </tr>
108        <% end %>
109        <!-- # OF return -->
110        <% unless @booking.flight_no.blank? %>
111        <tr>
112        <td>
113            <table cellspacing="0" cellpadding="0" border="0" width="100%">
114            <tr>
115            <td valign="top">
116            <!-- LEFT COLUMN -->
117            <table cellpadding="0" cellspacing="0" border="0" width="26%" align="left"
class="responsive-table">
118            <tr>
119            <td align="center" bgcolor="#ffffff" class="flex-p-bold">
120                Flight:
121            </td>
122            </tr>
123            </table>
124            <!-- RIGHT COLUMN -->
125            <table cellpadding="0" cellspacing="0" border="0" width="70%" align="right"
class="responsive-table">
126            <tr>
127            <td align="center" bgcolor="#ffffff"
128                class="flex-p">
129            <%= @booking.flight_no %>
130            </td>
131            </tr>
132            </table>
133            </td>
134            </tr>
135            </table>
136        </td>

```

```

137         </tr>
138     <% end %>
139     <% if user_signed_in? %>
140         <tr>
141             <td>
142                 <table cellspacing="0" cellpadding="0" border="0" width="100%">
143                     <tr>
144                         <td valign="top">
145                             <!-- LEFT COLUMN -->
146                             <table cellpadding="0" cellspacing="0" border="0" width="26%" align="left"
class="responsive-table">
147                                 <tr>
148                                     <td align="center" bgcolor="#ffffff" class="flex-p-bold">
149                                         Full Name:
150                                     </td>
151                                 </tr>
152                             </table>
153                             <!-- RIGHT COLUMN -->
154                             <table cellpadding="0" cellspacing="0" border="0" width="70%" align="right"
class="responsive-table">
155                                 <tr>
156                                     <td align="center" bgcolor="#ffffff"
class="flex-p">
157                                     <%= current_user.first_name %> <%= current_user.last_name %>
158                                 </td>
159                             </tr>
160                         </table>
161                     </td>
162                 </tr>
163             </table>
164         </td>
165     </tr>
166 <% end %>
167 <tr>
168 <td>
169     <table cellspacing="0" cellpadding="0" border="0" width="100%">
170         <tr>
171             <td valign="top" >
172                 <!-- LEFT COLUMN -->
173                 <table cellpadding="0" cellspacing="0" border="0" width="26%" align="left"
class="responsive-table">
174                     <tr>
175                         <td align="center" bgcolor="#ffffff" class="flex-p-bold">
176                             Passengers:
177                         </td>
178                     </tr>
179                 </table>
180                 <!-- RIGHT COLUMN -->
181                 <table cellpadding="0" cellspacing="0" border="0" width="70%" align="right"
class="responsive-table">
182                     <tr>
183                         <td align="center" bgcolor="#ffffff"
class="flex-p">
184                         <%= @passengers[:adults]? "#{@passengers[:adults]} adults":" " %>
185                         <%= @passengers[:children]? " #{@passengers[:children]} children(s)":" " %>
186                         <%= @passengers[:infants]? "#{@passengers[:infants]} infant":" " %>
187                     </td>
188                 </table>
189             </td>

```

```

190         </tr>
191     </table>
192 </td>
193 </tr>
194 </table>
195 </td>
196 </tr>
197 <tr>
198 <td>
199     <table cellspacing="0" cellpadding="0" border="0" width="100%">
200     <tr>
201     <td valign="top">
202         <!-- LEFT COLUMN -->
203         <table cellpadding="0" cellspacing="0" border="0" width="26%" align="left"
class="responsive-table">
204     <tr>
205     <td align="center" bgcolor="#ffffff" class="flex-p-bold">
206         Payment:
207     </td>
208 </tr>
209 </table>
210     <!-- RIGHT COLUMN -->
211     <table cellpadding="0" cellspacing="0" border="0" width="70%" align="right"
class="responsive-table">
212     <tr>
213     <td align="center" bgcolor="#ffffff"
class="flex-p">
214         Pay your driver directly in cash
215     </td>
216 </tr>
217 </table>
218 </table>
219 </td>
220 </tr>
221 </table>
222 </td>
223 </tr>
224 </table>
225 </td>
226 </tr>
227
228 <tr>
229 <td>
230     <table width="100%" border="0" cellspacing="0" cellpadding="0">
231     <tr>
232     <td align="left">
233         Trip information <span>(in approximate)</span>
234     </td>
235 </tr>
236 <!-- CHECK-IN -->
237 <tr>
238 <td>
239     <table cellspacing="0" cellpadding="0" border="0" width="100%">
240     <tr>
241     <td valign="top">
242         <!-- LEFT COLUMN -->
243         <table cellpadding="0" cellspacing="0" border="0" width="26%" align="left"
class="responsive-table">

```

```

244         <tr>
245         <td align="center" bgcolor="#ffffff" class="flex-p-bold">
246             Distance:
247         </td>
248     </tr>
249
250
251     <tr>
252     <td align="center" bgcolor="#ffffff" class="flex-p-bold">
253         Duration:
254     </td>
255 </tr>
256
257 <tr>
258 <td align="center" bgcolor="#ffffff" class="flex-p-bold">
259     Price:
260 </td>
261 </tr>
262
263 </table>
264 <!-- RIGHT COLUMN -->
265 <table cellpadding="0" cellspacing="0" border="0" width="70%" align="right"
class="responsive-table">
266 <tr>
267 <td id="distance" align="center" bgcolor="#ffffff"
class="flex-p">
268 <!-- <%= @miles_distance %>KM -->
269 </td>
270 </tr>
271
272
273 <tr>
274 <td id="duration" align="center" bgcolor="#ffffff"
class="flex-p">
275 <!-- <%= @duration %> -->
276 <!-- duration is calculated from google maps -->
277 </td>
278 </tr>
279
280
281 <tr>
282 <td align="center" bgcolor="#ffffff"
class="flex-p">
283 <%= @price %>E
284 </td>
285 </tr>
286 </tr>
287 </table>
288 </td>
289 </tr>
290 </table>
291 </td>
292 </tr>
293
294 </table>
295 </td>
296 </tr>
297
298 <%= if user_signed_in? %>
299 <%= form_for @booking, :url => { :action => 'update' } do |f| %>

```

```

300     <% f.check_box :confirmation, {:checked => true } %>
301     <tr width="100%">
302         <td>
303             <div class="form-group">
304                 <%= f.label :message, class: "control-label" %>
305                 <%= f.text_area :message, class: "form-control" %>
306             </div>
307         </td>
308
309     </tr>
310     <tr>
311         <td width="100%" >
312             <%= f.submit "Book now your ride", class: "btn btn-gold cta" %>
313         </td>
314     </tr>
315 <% end %>
316 <% else %>
317     <tr>
318         <td width="100%" >
319             <%= button_to "Sign in to proceed", new_user_session_path, class: "btn btn-gold cta" %>
320         </td>
321     </tr>
322 <% end %>
323
324
325     </table>
326 </td>
327 </tr>
328 </table>
329 </td>
330 </tr>
331 </table>
332 </div>
333
334 </div>
335
336 <div class="col-lg-12 remove_padding_rg">
337 <div id="map" style='width: 100%; height: 600px;'></div>
338 </div>
339

```

Το περιεχόμενο της προεπισκόπησης είναι δυναμικό και κάθε φορά παίρνει τις τιμές από τον controller. Πιο συγκεκριμένα

```

1 class BookingController < ApplicationController
2
3   def continue
4     @booking = Booking.find(params['id'])
5     @cornitates = get_coordinates(@booking)
6
7
8     @passengers = get_passengers(@booking)
9     @price = calculate_price_per_km(distance)
10
11   end

```

```

12
13
14 private
15 def continue_params
16   params.require(:booking).permit(:message)
17 end
18
19 def get_coordinates(booking)
20   pick_up = Location.where(address: booking.pick_up)
21   drop_off = Location.where(address: booking.drop_off)
22   pick_up_cordinates, drop_off_cordinates, cornitates = Hash.new
23   pick_up.each do |p|
24     pick_up_cordinates = {:latitude => p.latitude, :longitude => p.longitude}
25   end
26
27   drop_off.each do |d|
28     drop_off_cordinates = {:latitude => d.latitude, :longitude => d.longitude}
29   end
30   cornitates = {:pick_up => pick_up_cordinates, :drop_off => drop_off_cordinates}
31
32   return cornitates
33 end
34
35
36 def calculate_price_per_km(km)
37   price = (km*1.29 + 1.29).round
38   case price
39   when 0.0..3.0 then return "5"
40   when 4.0..6.0 then return "4-6"
41   when 7.0..9.0 then return "7-9"
42   when 9..11 then return "11-15"
43   when 12..15 then return "15-20"
44   when 15..20 then return "20-25"
45   when 20..30 then return "25-30"
46   when 30..200 then return "100-125"
47
48   end
49 end
50
51
52
53 def get_passengers(booking)
54   passengers = Hash.new
55   children = nil
56   adults = booking.adults
57   if booking.children_2_4 && booking.children_5_11
58     children = booking.children_2_4 + booking.children_5_11
59   end
60   infants = booking.infants
61
62   passengers = {:adults => adults, :children => children, :infants => infants }
63 end
64
65 end

```


Στην γραμμή 4 επιλέγουμε από την βάση την κράτηση με το id που έχουμε στις παραμέτρους και την αναθέτουμε στην μεταβλητή @booking. Πρακτικά αυτή η εντολή δημιουργεί sql query: `SELECT "bookings".* FROM "bookings" WHERE id = params['id']`.

έπειτα δημιουργήσαμε τις διάφορες μεθόδους για να υπολογίσουμε την τιμή, τους συνολικούς επιβάτες και τέλος με βάση τις τοποθεσίες που έχει η κράτηση να βρούμε και να αναθέσουμε τις συντεταγμένες που χρειάζονται για τον υπολογισμό της διαδρομής στον χάρτη. Τέλος, όλες οι μεταβλητές στις οποίες έχουμε αναθέσει τις τιμές έχουν το σύμβολο @ μπροστά έτσι ώστε να μπορούμε να τις χρησιμοποιήσουμε στις όψεις.

Η εμφάνιση του χάρτη όπως και ο υπολογισμός του χρόνου μαζί με την απόσταση μεταξύ των τοποθεσιών υπολογίζονται με εξωτερική υπηρεσία. Η υπηρεσία είναι της google και μας δίνει την δυνατότητα να προσθέσουμε την βιβλιοθήκη τους μέσω JavaScript

```
1 <script src=https://maps.googleapis.com/maps/api/js?key=KEY&language=en&callback=initMap async defer></script>
```

Εφόσον έχουμε φορτώσει την βιβλιοθήκη αναθέτουμε τις συντεταγμένες που υπολογίσαμε στον controller, γίνεται αρχικοποίηση του αντικείμενου Maps και έπειτα περνάμε τις παραμέτρους. Εχοντας ρυθμίσει όλα τα απαραίτητα στοιχεία, το google maps api αναλαμβάνει την εμφάνιση του χάρτη με τις συντεταγμένες που του δώσαμε, όταν ολοκληρώσει τον υπολογισμό το API της google μας επιστρέφει διάφορες τιμές. Οι τιμές που μας ενδιαφέρουν είναι η απόσταση και ο χρόνος. Τις οποίες και εμφανίζουμε στην προεπισκόπηση της κράτησης. Ακολουθεί ο κώδικας που αναλαμβάνει την λειτουργικότητα που περιγράψαμε προ ολίγου .

```
1
2 var pick_up_lat = <%= @cornitates[:pick_up][:latitude] %>,
3   pick_up_lang = <%= @cornitates[:pick_up][:longitude] %>,
4   drop_lat = <%= @cornitates[:drop_off][:latitude] %>,
5   drop_lang = <%= @cornitates[:drop_off][:longitude] %>;
6
7 function initMap() {
8   var pick_up_destination = {lat: pick_up_lat, lng: pick_up_lang};
9   var drop_off_destination = {lat: drop_lat, lng: drop_lang};
10
11
12   var map = new google.maps.Map(document.getElementById('map'), {
13     center: pick_up_destination,
14     scrollwheel: false,
15     zoom: 7
16   });
17
18   var directionsDisplay = new google.maps.DirectionsRenderer({
19     map: map
20   });
21
22   // Set destination, origin and travel mode.
23   var request = {
24     destination: drop_off_destination,
25     origin: pick_up_destination,
26     travelMode: google.maps.TravelMode.DRIVING
27   };
28
29   // Pass the directions request to the directions service.
30   var directionsService = new google.maps.DirectionsService();
31   directionsService.route(request, function(response, status) {
32     if (status == google.maps.DirectionsStatus.OK) {
33       // Display the route on the map.
34       directionsDisplay.setDirections(response);
```

```

35   }
36   });
37
38   var service = new google.maps.DistanceMatrixService;
39   service.getDistanceMatrix({
40     origins: [pick_up_destination],
41     destinations: [drop_off_destination],
42     travelMode: google.maps.TravelMode.DRIVING,
43     unitSystem: google.maps.UnitSystem.METRIC,
44     avoidHighways: false,
45     avoidTolls: false
46   }, function(response, status) {
47     if (status !== google.maps.DistanceMatrixStatus.OK) {
48       alert('Error was: ' + status);
49     } else {
50       var results = response.rows[0].elements,
51           distance = results[0].distance.text,
52           duration = results[0].duration.text;
53       $("#distance").text(distance);
54       $("#duration").text(duration);
55     }
56   });
57 }

```

Τέλος με την επιβεβαίωση του χρήστη η κράτηση ολοκληρώνεται, έπειτα ο χρήστης λαμβάνει συνοπτικό e-mail με τα στοιχεία της κράτησης. Αντίστοιχο e-mail λαμβάνει και ο οδηγός για την ενημέρωση της κράτησης.

HALKIDIKI TRANSFER
EXCLUSIVE DRIVING SERVICES

Online Booking Confirmation

Booking Reference: eae30b

Passengers
Katerina Patticha

Travel itinerary
From Polychrono to Kallithea, Halkidiki - Friday, March 31, 2017

From	To	Car	Duration*
04:00 Polychrono	Kallithea, Halkidiki	Mercedes-Benz E-Class	5-10m

Price Summary

All passengers	1	€15-20
Cars Needed	x1	
TOTAL PRICE*	€ 15-20	

My Booking
Now you can manage your booking online with more flexibility!
[View your booking](#)
Manage my booking

* See notes! Duration and Total Price are calculated in approximate. You are required to pay for each of the unit of the trip.

Εικόνα 15 email επιβεβαίωσης που λαμβάνει ο χρήστης μετά την ολοκλήρωση της κράτησης

Επιπλέον μέσα από την εφαρμογή οι χρήστες έχουν δυνατότητα να δουν το ιστορικό των κρατήσεων τους καθώς και προεπισκόπηση της κράτησης.

Booking History

The list displays overall view of all booking orders

#	Pick Up	Drop Off	Pick Up Date	Return Date	Status	Payment
152	Polychrono	Kallithea, Halkidiki	31/03/2017 04:00	one way	Confirmed	Pay by Cash



About the company

Trusted family company with more than 20 years of experience. We specialize in providing high quality transport services.

Εικόνα 16 ιστορικό κρατήσεων

4. Επίλογος

4.1. Αποτελέσματα

Η υλοποίηση της παρούσας πτυχιακής είχε ως κύριο σκοπό να παρουσιάσει αναλυτικά πως η κατασκευή μιας online εφαρμογής μπορεί να δώσει λύση σε ένα πρόβλημα που παρατηρείται στην καθημερινότητα και να βελτιώσει την περιήγηση του χρήστη υιοθετώντας νέες τεχνικές και εργαλεία όπως το responsive design.

Έχοντας κάνει εκτενή ερευνά πάνω στις γλώσσες και τα εργαλεία που υπάρχουν στην αγορά για την κατασκευή ιστοσελίδων όχι μόνο αποκτήθηκε γνώση αλλά αναπτύχθηκε και η κριτική ικανότητα για την σύγκριση μεταξύ τους. Το σημαντικότερο είναι ότι με τον σχεδιασμό της υπάρχουσας πτυχιακής λήφθηκαν υπόψιν οι πραγματικές ανάγκες ενός προτζεκτ και πόσο σημαντικός μπορεί να είναι συνδυασμός των τεχνολογιών που θα συντελέσουν στην υλοποίηση του.

Τέλος, πολύτιμες γνώσεις αποκτήθηκαν με την ανάπτυξη της εργασίας. Σε επίπεδο γλωσσών προγραμματισμού αλλά και οργάνωσης.

4.2. Μελλοντική εργασία και επεκτάσεις

Ο σκοπός της πτυχιακής ήταν να αναπτυχθεί ένα πρότυπο εφαρμογής όπου θα βασιστούν οι μελλοντικές επεκτάσεις. Μπορεί να αναπτυχθεί ακόμα παραπάνω με πολλές περισσότερες λειτουργικότητες. Μια πιθανή μελλοντική επέκταση θα μπορούσε να είναι η αυτόματη ανάθεση των κρατήσεων σε οδηγούς ταξί που θα συμμετείχαν στην πλατφόρμα. Η αξιολόγηση των υπηρεσιών από τους πελάτες και τέλος να μπορεί να υποστηρίζει πληρωμές μέσα από την εφαρμογή με κάρτα.

5. Βιβλιογραφία

- [1] wikipedia-html, «wikipedia,» wikipedia, [Ηλεκτρονικό]. Available: <https://el.wikipedia.org/wiki/HTML>.
- [2] wikipedia-css, «<https://el.wikipedia.org/wiki/CSS>,» <https://el.wikipedia.org/wiki/CSS>, [Ηλεκτρονικό]. Available: <https://el.wikipedia.org/wiki/CSS>.
- [3] wikipedia-javascript, «<https://el.wikipedia.org/wiki/JavaScript>,» <https://el.wikipedia.org/wiki/JavaScript>, [Ηλεκτρονικό]. Available: <https://el.wikipedia.org/wiki/JavaScript>.
- [4] wikipedia-JQuery, «<https://el.wikipedia.org/wiki/JQuery>,» <https://el.wikipedia.org/wiki/JQuery>, [Ηλεκτρονικό]. Available: <https://el.wikipedia.org/wiki/JQuery>.
- [5] wikipedia, «el.wikipedia.org,» [Ηλεκτρονικό]. Available: <https://el.wikipedia.org/wiki/PHP>.
- [6] builtwith, «https://trends.builtwith.com,» [builtwith.com](https://trends.builtwith.com/), [Ηλεκτρονικό]. Available: <https://trends.builtwith.com/framework/programming-language>.
- [7] wikipedia, «https://el.wikipedia.org,» https://el.wikipedia.org, [Ηλεκτρονικό]. Available: https://el.wikipedia.org/wiki/Ruby_on_Rails.
- [8] dklisiaris, «[dklisiaris](https://dklisiaris.gitbooks.io/thesis/content/about_rails/01-basics.html),» [Ηλεκτρονικό]. Available: https://dklisiaris.gitbooks.io/thesis/content/about_rails/01-basics.html.
- [9] wikipedia, «wikipedia,» wikipedia, [Ηλεκτρονικό]. Available: <https://el.wikipedia.org/wiki/Bootstrap>.
- [10] wikipedia/Responsive_design, «https://el.wikipedia.org,» https://el.wikipedia.org, [Ηλεκτρονικό]. Available: https://el.wikipedia.org/wiki/Responsive_design.
- [11] wikipedia-databases, «https://el.wikipedia.org/wiki/%CE%92%CE%AC%CF%83%CE%B7_%CE%B4%CE%B5%CE%B4%CE%BF%CE%BC%CE%AD%CE%BD%CF%89%CE%BD,» https://el.wikipedia.org/wiki/%CE%92%CE%AC%CF%83%CE%B7_%CE%B4%CE%B5%CE%B4%CE%BF%CE%BC%CE%AD%CE%BD%CF%89%CE%BD, [Ηλεκτρονικό]. Available: https://el.wikipedia.org/wiki/%CE%92%CE%AC%CF%83%CE%B7_%CE%B4%CE%B5%CE%B4%CE%BF%CE%BC%CE%AD%CE%BD%CF%89%CE%BD.
- [12] wikipedia.-PostgreSQL, «<https://el.wikipedia.org/wiki/PostgreSQL>,» [Ηλεκτρονικό]. Available: <https://el.wikipedia.org/wiki/PostgreSQL>.
- [13] wikipedia/server, «wikipedia,» [Ηλεκτρονικό]. Available: <https://el.wikipedia.org/wiki/%CE%95%CE%BE%CF%85%CF%80%CE%B7%CF%81%CE%B5%CF%84%CE%B7%CF%84%CE%AE%CF%82>.
- [14] wikipedia/maps-google, «https://el.wikipedia.org/wiki/%CE%A7%CE%AC%CF%81%CF%84%CE%B5%CF%82_G

oogle.» [Ηλεκτρονικό]. Available:

https://el.wikipedia.org/wiki/%CE%A7%CE%AC%CF%81%CF%84%CE%B5%CF%82_Google.

