ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ - ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

Τομέας: ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
Εργαστήριο: ΔΙΑΔΡΑΣΤΙΚΩΝ ΤΕΧΝΟΛΟΓΙΩΝ

Διπλωματική Εργασία

του φοιτητή του Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών της Πολυτεχνικής Σχολής του Πανεπιστημίου Πατρών

Ευστράτιου Καρδαμπίκη του Θεμιστοκλή

αριθμός μητρώου: 1019506

Θέμα

Σχεδίαση και ανάπτυξη μιας εφαρμογής υποστήριξης κοινότητας χρηστών με χρήση React Native.
Design and development of a user community support application using React Native.

Επιβλέπων

Καθηγητής Αβούρης Νικόλαος
Αριθμός Διπλωματικής Εργασίας:

Πάτρα, Μάιος 2020

ΠΙΣΤΟΠΟΙΗΣΗ

Πιστοποιείται ότι η διπλωματική εργασία με θέμα

Σχεδίαση και ανάπτυξη μιας εφαρμογής υποστήριξης κοινότητας χρηστών με χρήση React Native.
Design and development of a user community support application using React Native.

του φοιτητή του Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

Ευστράτιου Καρδαμπίκη του Θεμιστοκλή

(Α.Μ.: 1019506)

παρουσιάτηκε δημόσια και εξετάστηκε στο τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών στις

__/__/___

Ο Επιβλέπων

Αβούρης Νικόλαος
Καθηγητής

Ο Διευθυντής του Τομέα

Βασίλης Παλιουράς
Καθηγητής

Στοιχεία διπλωματικής εργασίας

Θέμα:

Σχεδίαση και ανάπτυξη μιας εφαρμογής υποστήριξης κοινότητας χρηστών με χρήση React Native.

Design and development of a user community support application using React Native.

Φοιτητής: **Ευστράτιος Καρδαμπίκης του Θεμιστοκλή**

Ομάδα επίβλεψης

Καθηγητής Αβούρης Νικόλαος

Εργαστήρια

ΔΙΑΔΡΑΣΤΙΚΩΝ ΤΕΧΝΟΛΟΓΙΩΝ

ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική εργασία αφορά την δημιουργία και την ανάπτυξη μιας εφαρμογής, χρήσιμης για μια κοινότητα χρηστών. Βασικός στόχος της εφαρμογής είναι η δυνατότητα ενημέρωσης και διευκόλυνσης της καθημερινότητας του χρήστη-ιδιοκτήτη σκύλου.

Παρακάτω αναλύονται όλα τα βήματα που ακολουθήθηκαν κατά την δημιουργία της εφαρμογής. Δηλαδή, η έρευνα που πραγματοποιήθηκε πάνω στην κοινότητα, οι ανάγκες της και τα ήδη υπάρχοντα εργαλεία που διαθέτει το διαδίκτυο για τους ίδιους. Επίσης, περιγράφεται ο σχεδιασμός και η υλοποίηση του συστήματος καθώς και η αξιολόγηση που έγινε από έμπειρους αξιολογητές αλλά και τους ίδιους τους χρήστες.

Καταλήγοντας, γίνεται εκτεταμένη συζήτηση σχετικά με τις τεχνολογίες που χρησιμοποιήθηκαν κατά την υλοποίηση της εφαρμογής, περιγράφεται λεπτομερώς ο κώδικας που αναπτύχθηκε, ενώ γίνεται ιδιαίτερη αναφορά στις εφαρμογές που χρησιμοποιήθηκαν κατά τη συγκεκριμένη αξιολόγηση του συστήματος που αναπτύχθηκε.

Λέξεις-Κλειδιά
Κινητή συσκευή,Εφαρμογή, Ιδιοκτήτες σκυλιών, Εύρεση Τοποθεσιών

Abstract

This dissertation focuses on the creation and development of an application quite useful for a community of users. The main goal of the application is to inform and facilitate the daily life of the user/dog-owner.

The steps taken to create this application are analyzed below including the research, the design, the implementation and the evaluation it received.

Finally, a discussion on the programming languages that were applied in the implementation of the application, the software that was used as well as the corresponding applications that were compared in our evaluation, are all highlighted.

ΕΥΧΑΡΙΣΤΙΕΣ

Αρχικά, Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή μου κύριο Νικόλαο Αβούρη τόσο για την έμπνευση, το ενδιαφέρον και το κίνητρο που μου έδωσε για να ασχοληθώ με το συγκεκριμένο κλάδο όσο και για τη βοήθειά και τις υποδείξεις του κατά την διάρκεια της εκτέλεσης της διπλωματικής εργασίας.

Κλείνοντας, θα ήθελα να εκφράσω την ευγνωμοσύνη μου στην οικογένεια μου για τη στήριξή τους κατά τη διάρκεια της ακαδημαϊκής μου πορείας, όπως επίσης και τα κοντινά μου πρόσωπα για την συμπαράστασή τους όλα αυτά τα χρόνια.

ΠΕΡΙΕΧΟΜΕΝΑ

Κατάλογος σχημάτων **xvii**

Κατάλογος πινάκων **xix**

1 ΕΙΣΑΓΩΓΗ **1**
1.1 Σκοπός Διπλωματικής Εργασίας . . . 1
1.1.1 Περιγραφή Θέματος . . . 1
1.1.2 Στόχοι . . . 2
1.2 Μεθοδολογία . . . 2
1.3 Δομή Διπλωματικής Εργασίας . . . 3

2 ΕΡΕΥΝΑ ΣΧΕΤΙΚΩΝ ΕΦΑΡΜΟΓΩΝ **5**
2.1 Εφαρμογές κινητών συσκευών . . . 5
2.1.1 Dog Health . . . 6
2.1.2 Dog Walk - Track your dogs! . . . 7
2.1.3 BringoFido-Pet Friendly Hotels . . . 8
2.1.4 Doggy Logs-Dog Walk GPS Tracking . . . 9
2.2 Ιστοσελίδες και εφαρμογές για σταθερούς υπολογιστές . . . 10
2.2.1 Doggo . . . 11
2.2.2 Vetstreet . . . 12
2.2.3 Dogster . . . 13
2.2.4 The Dog People by Rover.com . . . 14

3 ΑΝΑΛΥΣΗ **17**
3.1 Συγκέντρωση προδιαγραφών . . . 17
3.1.1 Τεχνολογίες πληροφοριών για ιδιοκτήτες σκύλων . . . 17
3.1.2 Τεχνολογίες ιστού για ιδιοκτήτες σκύλων . . . 18
3.1.3 Δυνατότητες Κινητών Συσκευών για ιδιοκτήτες σκυλιών . . . 19
3.1.4 Ανακεφαλαίωση προδιαγραφών . . . 19
3.2 Ανάλυση χρηστών και περιβάλλοντος εργασίας . . . 20
3.3 Λειτουργικές απαιτήσεις συστήματος . . . 21
3.4 Μη-λειτουργικές απαιτήσεις συστήματος . . . 21

4 ΣΧΕΔΙΑΣΗ **23**
4.1 Χάρτης πλοήγησης . . . 23
4.2 Πρωτότυπο . . . 25
4.2.1 Παράθυρο Login Χρηστών . . . 26
4.2.2 Παράθυρο Εγγραφής Χρήστη . . . 27
4.2.3 Κεντρική σελίδα, Home Page . . . 28
4.2.4 HomePage με επιλογή εμφάνισης άλλων σελίδων . . . 29
4.2.5 Οθόνη AboutPage . . . 29
4.2.6 Οθόνη SettingsPage . . . 31

4.2.7 Προβολή Χάρτη με τοποθεσίες για Πάρκα,Καφετέριες,Κτηνιατρεία και Pet Shops . . . 32
4.2.8 Προβολή Carousel πληροφοριών για τις τοποθεσίες . . . 33
4.2.9 Προβολή σχόλιον για τις τοποθεσίες . . . 34
4.2.10 Προβολή Μεταφοράς στην τοποθεσία . . . 35
4.2.11 Οργάνωση Οθονών . . . 36
4.3 Σενάρια χρήσης εφαρμογής . . . 38
4.3.1 Σενάριο Χρήσης 1 . . . 38
4.3.2 Σενάριο Χρήσης 2 . . . 38
4.3.3 Σενάριο Χρήσης 3 . . . 40
4.3.4 Σενάριο Χρήσης 4 . . . 41
4.3.5 Σενάριο Χρήσης 5 . . . 41
4.3.6 Σενάριο Χρήσης 6 . . . 43
4.3.7 Σενάριο Χρήσης 7 . . . 43
4.3.8 Σενάριο Χρήσης 8 . . . 45
4.3.9 Σενάριο Χρήσης 9 . . . 47

5 ΥΛΟΠΟΙΗΣΗ 49
5.1 Τεχνολογίες Σχεδιασμού Συστημάτων . . . 49
5.2 Είδη τεχνολογιών front end . . . 50
5.3 Είδη τεχνολογιών back end . . . 51
5.3.1 Γλώσσα Προγραμματισμού Java . . . 51
5.3.2 Γλώσσα Προγραμματισμού PHP . . . 51
5.4 Λογισμικό που χρησιμοποιήθηκε . . . 51
5.4.1 Τεχνολογία ανάπτυξης του front end του συστήματος . . . 51
5.4.2 Τεχνολογία Ανάπτυξης του Back End του Συστήματος . . . 55
5.4.3 Λογισμικό Server . . . 57
5.5 Βιβλιοθήκες . . . 57
5.5.1 Vector Icons . . . 57
5.5.2 React Native Router Flux . . . 57
5.5.3 Google Maps API . . . 58
5.5.4 GeoLib . . . 58
5.5.5 React Native Star Rating . . . 58
5.6 Plugins . . . 58
5.6.1 Map View Directions . . . 58
5.6.2 Geolocation . . . 58
5.6.3 Dimensions . . . 58
5.7 ScreenShots Εφαρμογής . . . 58

6 ΑΞΙΟΛΟΓΗΣΗ 67
6.1 Τεχνικές αξιολόγησης . . . 67
6.2 Μεθοδολογία αξιολόγησης . . . 68
6.2.1 Μέθοδος Ευρετικής Αξιολόγησης(Heuristic Evaluation) . . . 68
6.2.2 Πλεονεκτήματα-Μειονεκτήματα . . . 68
6.2.3 Ερωτήματα Αξιολόγησης . . . 69
6.2.4 Ενημέρωση Αξιολογητών . . . 70
6.2.5 Διερεύνηση εφαρμογής . . . 70
6.2.6 Ανάλυση αποτελεσμάτων ευρετικής αξιολόγησης(Heuristic Evaluation) . . . 74
6.2.7 Πρωτόκολλο Ερωτήσεων (Question Asking Protocol) . . . 76
6.2.8 Εκτέλεση Πρωτοκόλλων Ερωτήσεων . . . 76
6.2.9 Ανάλυση αποτελεσμάτων για πρωτόκολλο Ερωτήσεων (Question Asking Protocol) . . . 77

6.2.10 Συμπεράσματα Σχόλια 80

7 ΣΥΜΠΕΡΆΣΜΑΤΑ-ΜΕΛΛΟΝΤΙΚΈΣ ΕΠΕΚΤΆΣΕΙΣ 81
7.1 ΣΥΜΠΕΡΑΣΜΑΤΑ 81
7.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ 81

8 ΠΑΡΑΡΤΗΜΑ 83
8.1 Κώδικας REACT NATIVE 83
8.2 Κώδικας Laravel 94

9 ΒΙΒΛΙΟΓΡΑΦΙΑ 99

Κατάλογος σχημάτων

1.1 Το μοντέλο του καταρράκτη . . . 3

2.1 Η εφαρμογή ”Dog Health” στο Google Play . . . 6
2.2 Η εφαρμογή ”Dog Walk - Track your dogs!” στο Google Play . . . 7
2.3 Η εφαρμογή ”BringoFido-Pet Friendly Hotels” στο Google Play . . . 8
2.4 Η εφαρμογή ”Doggy Logs-Dog Walk GPS Tracking” στο Google Play . . . 9
2.5 Στιγμιότυπο από την ιστοσελίδα του Doggo . . . 11
2.6 Στιγμιότυπο από την ιστοσελίδα του Vetstreet . . . 12
2.7 Στιγμιότυπο από την ιστοσελίδα του Dogster . . . 13
2.8 Στιγμιότυπο από την ιστοσελίδα του The Dog People by Rover.com . . . 14

3.1 Screen Resolutions . . . 22

4.1 Χάρτης Πλοήγησης . . . 24
4.2 Παράθυρο σύνδεσης πρωτοτύπου . . . 26
4.3 Παράθυρο εγγραφής πρωτοτύπου . . . 27
4.4 Παράθυρο HomePage πρωτοτύπου . . . 28
4.5 Παράθυρο HomePage με επιλογή εμφάνισης άλλων σελίδων πρωτοτύπου . . . 29
4.6 Παράθυρο AboutPage πρωτοτύπου . . . 30
4.7 Παράθυρο SettingPage πρωτοτύπου . . . 31
4.8 Παράθυρο παράδειγμα πρωτοτύπου με τις τοποθεσίες για PetShops . . . 32
4.9 Παράθυρο παράδειγμα πρωτοτύπου με το Carousel πληροφοριών για τις τοποθεσίες 33
4.10 Παράθυρο σχόλιον πρωτοτύπου . . . 34
4.11 Παράθυρο μεταφοράς στην τοποθεσία πρωτοτύπου . . . 35
4.12 Τρόποι Οργάνωσης Εφαρμογής . . . 36
4.13 Οργάνωση Οθονών Πρωτοτύπου . . . 37

5.1 Front end vs back end programming . . . 50
5.2 React Native Rendering . . . 55
5.3 Laravel Benefits . . . 56
5.4 Διάγραμμα Apache Web Server . . . 57
5.5 Φόρμα Σύνδεσης . . . 59
5.6 Φόρμα Εγγραφής . . . 59
5.7 Home Page . . . 60
5.8 Παράθυρο εμφάνισης επιλογών . . . 60
5.9 Εμφάνιση άλλων οθονών . . . 61
5.10 About Page . . . 61
5.11 Settings Page . . . 62
5.12 Changing User’s Settings . . . 62
5.13 Επιλογή Dog Parks, είναι ίδιο για κάθε επιλογή . . . 63
5.14 Carousel πληροφοριών για τις τοποθεσίες . . . 63
5.15 Comment Page . . . 64
5.16 Εισαγωγή σχόλιου στο Comment Page . . . 64

5.17 Επιλογή πλοήγησης σε μια τοποθεσία με αμάξι . . . 65
5.18 Επιλογή πλοήγησης σε μια τοποθεσία με τα πόδια . . . 65

6.1 Παράθυρο για βελτίωση Directions . . . 77
6.2 Παράθυρο για βελτίωση Home Page . . . 78
6.3 Παράθυρο εμφάνισης επιλογών . . . 78
6.4 Παράθυρα για βελτίωση About Page, Setting Page . . . 79
6.5 Παράθυρο για βελτίωση carousel . . . 79

8.1 Package.json . . . 83
8.2 App.json . . . 84
8.3 AndroidManifest.xml . . . 84
8.4 Index.json . . . 85
8.5 App.js . . . 85
8.6 Routers.js . . . 86
8.7 Drawer.js . . . 87
8.8 Home.js . . . 88
8.9 Parks.js . . . 88
8.10 fetch εντολή . . . 89
8.11 View Button . . . 89
8.12 Directions Button . . . 90
8.13 Σχήμα Carousel . . . 90
8.14 Συνάρτηση renderCarouselItem 1 . . . 91
8.15 Συνάρτηση renderCarouselItem 2 . . . 91
8.16 Σχήμα Directions . . . 92
8.17 Comments Component 1 . . . 93
8.18 Comments Component 2 . . . 93
8.19 SwipeOut Κώδικας . . . 94
8.20 Api.php . . . 95
8.21 UserController . . . 96
8.22 MainController . . . 96
8.23 commentController . . . 97
8.24 Composer.json . . . 97

Κατάλογος πινάκων

2.1 Πίνακας Σύγκρισης Εφαρμογών 15

4.1 Σενάριο χρήσης 1 38
4.2 Σενάριο χρήσης 2 39
4.3 Σενάριο χρήσης 3 40
4.4 Σενάριο χρήσης 4 41
4.5 Σενάριο χρήσης 5 42
4.6 Σενάριο χρήσης 6 43
4.7 Σενάριο χρήσης 7 44
4.8 Σενάριο χρήσης 8 45
4.9 Σενάριο χρήσης 9 47

6.1 Πίνακας με τους 10 κανόνες των Molich και Nielsen 69
6.2 Κλίμακα Αθροιστικής Βαθμολόγησης – Likert scale 71
6.3 Κλίμακα Αθροιστικής Βαθμολόγησης – Likert scale 72
6.4 Παρουσίαση Προβλημάτων 74

1. ΕΙΣΑΓΩΓΗ

Η ραγδαία εξέλιξη της τεχνολογίας σε συνδυασμό με τη συνεχή βελτίωση των υπολογιστικών συστημάτων και των κινητών συσκευών, οι οποίες έχουν ενσωματωθεί στην καθημερινότητα εκατομμυρίων ανθρώπων, έχει μεγιστοποιήσει την ανάγκη για καλύτερη σχεδίαση, μοντελοποίηση και εξυπηρέτηση των ατόμων που τα χρησιμοποιούν. Εκτός από τη βελτίωση της εξωτερικής εμφάνισης μιας κινητής συσκευής, είναι λογικό να απαιτείται και βελτίωση των δυνατοτήτων στον τομέα της επικοινωνίας, της αποθήκευσης και καταγραφής δεδομένων,αλλά και της ενημέρωσης και εύχρηστης πλοήγησης. Όλα τα προαναφερθέντα αποσκοπούν κυρίως στη διευκόλυνση των χρηστών.

Εκμεταλλευόμενοι λοιπόν αυτή την τεχνολογική δυνατότητα, οδηγηθήκαμε στη δημιουργία της εφαρμογής η οποία εξυπηρετεί σε μεγάλο βαθμό στην εύρεση πληροφοριών σχετικά με τις ανάγκες του ιδιοκτήτη ενός κατοικίδιου ζώου (ενός σκύλου). Στο πλαίσιο αυτής της διπλωματικής θα παρουσιαστούν συνοπτικά οι διαστάσεις της εν λόγω εφαρμογής και θα εξεταστούν οι διαδικασίες προγραμματισμού, ανάπτυξης και αξιολόγησής της.

1.1 Σκοπός Διπλωματικής Εργασίας

Στόχος της διπλωματικής εργασίας είναι η δημιουργία, η ανάπτυξη και ο σχεδιασμός μιας εφαρμογής για κινητές συσκευές, η οποία θα χρησιμοποιηθεί από τους ιδιοκτήτες σκύλων, με σκοπό να ανακαλύψουν πληροφορίες σχετικά με pet shop, κτηνιατρεία, πάρκα για σκυλιά και καφετέριες φιλικές προς τα κατοικίδια, κοντά στην περιοχή τους. Σημαντική λεπτομέρεια αποτελεί το γεγονός ότι θα δίνεται η δυνατότητα στα ίδια τα άτομα να αφήσουν κάποιο σχολιασμό ή απλά βαθμολογία στο μέρος που επισκέφθηκαν μέσω προσωπικών διαδικτυακών σχολίων, βαθμολογιών. Τέλος, καθότι όλες οι πληροφορίες σχετικά με τις τοποθεσίες βρίσκονται πάνω σε χάρτη, καθίσταται δυνατή η πλοήγηση και κατεύθυνση των χρηστών μέσω της εφαρμογής στο συγκεκριμένο σημείο.

1.1.1 Περιγραφή Θέματος

Ένα από τα βασικότερα εργαλεία που πλέον έχουν αντικαταστήσει τα παραδοσιακά -αλλά όχι τόσο εύχρηστα πια- ημερολόγια, τους κλασικούς χάρτες, τις GPS συσκευές και πολλά άλλα αντικείμενα της καθημερινότητας, αποτελούν τα έξυπνα κινητά (smartphones). Τα smartphones χρησιμοποιούνται σήμερα από όλους και παρέχουν δυνατότητες που παλαιότερα δεν είχαμε καν φανταστεί. Άτομα στα οποία είναι αρκετά χρήσιμα τα έξυπνα κινητά είναι και οι ιδιοκτήτες σκύλων. Αξιοποιώντας ένα τέτοιο πολυεργαλείο οι ιδιοκτήτες σκύλων, έχουν την ευκαιρία να βελτιώσουν τη σχέση τους με το κατοικίδιό τους. Η κινητή συσκευή αποτελεί ένα δυνατό μέσο ενημέρωσης και πληροφόρησης για έναν ιδιοκτήτη όσον αφορά ζητήματα υγείας, όπως η άμεση επικοινωνία με τον καλύτερο κτηνίατρο, ζητήματα που σχετίζονται με πρώτες ύλες/ανάγκες του σκύλου (τροφή, αξεσουάρ), αλλά και ζητήματα διασκέδασης που μπορεί να χρειαστεί ένας σκύλος και ο κάτοχος του.

Οι παραπάνω ανάγκες των ιδιοκτητών σκύλων σε συνδυασμό με τις απεριόριστες δυνατότητες της εποχής μας, έδωσαν κίνητρο για τη δημιουργία μίας εφαρμογής που θα παρέχει συγκεντρωμένες όλες αυτές τις επιλογές στους χρήστες (γρήγορη εύρεση τροφής, κτηνιατρεία, πάρκα διασκέδασης κ.λ.π.)

Για να μπορέσει λοιπόν το εργαλείο (εφαρμογή) να είναι χρήσιμο και εύχρηστο προς τους χρήστες, θα χρειαστεί να γίνει η ενδελεχής σχεδίαση και ανάπτυξη. Στο πλαίσιο αυτής της εργασίας, θα

παρουσιαστεί αναλυτικά η υλοποίηση ολόκληρης της εφαρμογής αλλά και σκοπός αυτής, που όπως ήδη προαναφέρθηκε είναι η ενημέρωση ατόμων με κατοικίδια σκυλιά.

1.1.2 Στόχοι

Στόχος της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη και υλοποίηση της εφαρμογής που θα ενημερώνει και θα πληροφορεί τους χρήστες σχετικά με την τοποθεσία που έχει το μέρος που αναζητά κάθε δεδομένη χρονική στιγμή.

Στον χρήστη της εφαρμογής θα δίνεται η δυνατότητα δημιουργίας προσωπικού λογαριασμού με τον οποίο θα συνδέεται κάθε φορά. Αφού εγκριθεί η σύνδεση του, ακολουθούν οι εξής τέσσερις επιλογές-κατηγορίες: α) Πάρκα για σκυλιά, β) Pet shops, γ) Κτηνιατρεία και δ) Pet friendly Café . Η επιλογή του κάθε φορά θα σχετίζεται με την ανάγκη που θα έχει εκείνη την χρονική στιγμή και θα τον μεταφέρει αυτόματα σε έναν ηλεκτρονικό χάρτη (google maps). Στον χάρτη λοιπόν, θα εμφανίζονται οι τοποθεσίες τον αντίστοιχων καταστημάτων/περιοχών σε ακτίνα χιλίων μέτρων από την τοποθεσία του χρήστη.

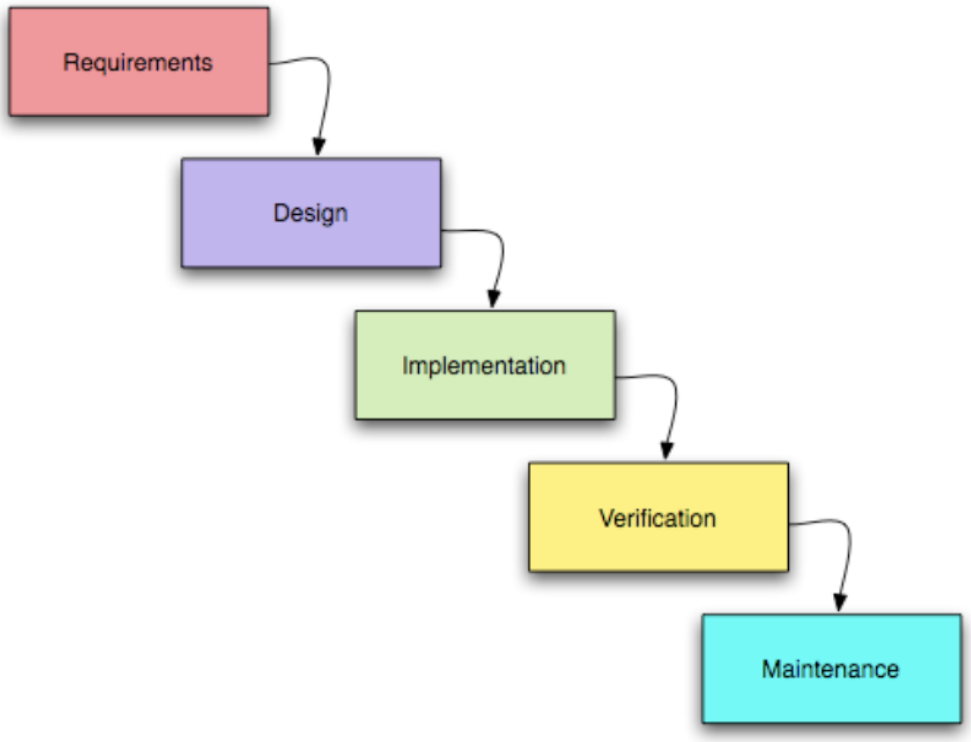
Ουσιαστικά, ο κάθε ιδιοκτήτης θα μπορεί να ενημερώνεται μέσα από τις κριτικές και τα σχόλια για την άποψη που έχουν άλλοι χρήστες για τις τοποθεσίες που ενδιαφέρεται να επισκεφθεί. Θα είναι κατά συνέπεια δυνατό να μεταφερθεί στο σημείο ενδιαφέροντος με την βοήθειά των οδηγιών, είτε με τα πόδια, είτε με κάποιο όχημα. Τέλος, αφού καταλήξει να επισκεφτεί την επιθυμητή τοποθεσία, θα έχει την δυνατότητα να προσθέσει το δικό του σχόλιο και να μοιραστεί την δική του προσωπική εμπειρία, στοχεύοντας στην ενημέρωση όλο και περισσοτέρων χρηστών.

1.2 Μεθοδολογία

Η ανάπτυξη και σχεδίαση του κάθε λογισμικού περνάει από διαφορά στάδια. Ο ρόλος των σταδίων είναι να συγκροτούν τον κύκλο του λογισμικού και καθώς περνούν από διάφορες μεθοδολογίες, να συνδυάζονται για το τελικό αποτέλεσμα. Το λογισμικό της εφαρμογής μας, η οποία μάλιστα φέρει το όνομα Lazy Max, αναπτύχθηκε ακολουθώντας μεθοδολογική ροή παρόμοια με τη ροή του μοντέλου του καταρράκτη (waterfall model). Το προαναφερόμενο μοντέλο αποτελείται από ποικίλες δραστηριότητες κάθε επιπέδου που εκτελούνται σε γραμμική σειρά, ενώ το κάθε επίπεδο εξαρτάται από την κατάσταση του προηγούμενου.

Το μοντέλο αυτό γενικά περιέχει τα παρακάτω στάδια:

1. Ορισμός Προβλήματος
2. Συγκέντρωση προδιαγραφών – απαιτήσεων
3. Σχεδίαση
4. Υλοποίηση
5. Αξιολόγηση και διόρθωση



Σχήμα 1.1: Το μοντέλο του καταρράκτη

Αναλύοντας τα παραπάνω επίπεδα, είναι πρέπων να οριστεί το πρόβλημα και να γίνει μια σαφής περιγραφή του έτσι ώστε στην παρούσα διπλωματική εργασία να επέλθει η λύση του. Συνεχίζοντας, στο δεύτερο επίπεδο υφίσταται η συγκέντρωση των προδιαγραφών-απαιτήσεων που έχει ή χρειάζεται η εφαρμογή καθώς και η μελέτη για υλοποίηση άλλων παρόμοιων εφαρμογών. Αφού ολοκληρωθεί το δεύτερο στάδιο, ακολουθεί το στάδιο του σχεδιασμού της εφαρμογής, το οποίο πραγματοποιείται με τη χρήση εργαλείου σχεδίασης πρωτοτύπων (UXPin). Στη συνέχεια, για να υλοποιηθεί και να κατασκευαστεί τελικά η εφαρμογή θα πρέπει να γίνει ενδελεχής μελέτη του περιβάλλοντος μέσα στο οποίο θα δουλέψουμε. Προχωρώντας, το τελικό στάδιο σχετίζεται με την τελική διόρθωση που μπορεί να χρειαστεί, αλλά και την αξιολόγηση της εφαρμογής.

Το μοντέλο του καταρράκτη έχει ως κύριο μειονέκτημα την αδυναμία λεπτομερούς προδιαγραφής του λογισμικού πριν την υλοποίησή του. Για να αντιμετωπιστεί αυτό το πρόβλημα το πρωτότυπο σχεδιάστηκε με μεγάλη ακρίβεια και υπήρχε τακτική επικοινωνία με τους υπεύθυνους καθηγητές για τυχόν αλλαγές του συστήματος της εφαρμογής. Έτσι, σε κάθε αλλαγή που προέκυπτε, υπήρχε ενημέρωση του πρωτοτύπου ώστε να γίνει ξεκάθαρος ο στόχος της εφαρμογής.

1.3 Δομή Διπλωματικής Εργασίας

Η δομή της συγκεκριμένης διπλωματικής εργασίας ταυτίζεται με τη μεθοδολογία που ακολουθήθηκε κατά την υλοποίηση της εφαρμογής καθώς και ένα κεφάλαιο της εργασίας αφορά την αξιολόγηση της και τη βελτίωση και την αποκατάσταση που μπορεί να υπάρξει στο μέλλον. Στο 1ο κεφάλαιο τοποθετείται το πρόβλημα δηλαδή ο λόγος δημιουργίας της εφαρμογής, οι μέθοδοι που χρησιμοποιούνται για την επίλυσή του (μοντέλο του καταρράκτη) καθώς και οι στόχοι.

Για τον εντοπισμό των κατάλληλων προδιαγραφών, αλλά και των απαιτήσεων της εφαρμογής, πρέπει πρώτα να μελετηθούν οι ήδη υπάρχουσες εφαρμογές που έχουν υλοποιηθεί σε σταθερές και κινητές συσκευές. Με αυτόν τον τρόπο καθίσταται πλέον εφικτό το να επιλέξουμε και να εφαρμόσουμε στην εφαρμογή μας τις πιο χρήσιμες λειτουργίες και τελικά να εξαλειφθούν τυχόν προβλήματα που υπάρχουν σε αυτές με σκοπό να βελτιστοποιήσουμε το αποτέλεσμα μας. Συνεχίζοντας, στο 2ο κεφάλαιο

ακολουθεί μια σύγκριση μεταξύ παρόμοιων προγραμμάτων και εφαρμογών. Τα δεδομένα που θα προκύψουν από την σύγκριση που έγινε σε αυτό το κεφάλαιο σε συνδυασμό με την αξιολόγηση των χρηστών, την ανάλυση των προδιαγραφών και απαιτήσεων θα παρουσιαστούν στο 3ο κεφάλαιο. Έπειτα, στο 4 κεφάλαιο θα αναπτύξουμε και θα αναλύσουμε τις οθόνες του πρωτοτύπου της εφαρμογής και θα παρουσιαστούν κατάλληλα σενάρια χρήσης ώστε να γίνει σαφές πως καλύπτονται οι υπάρχουσες απαιτήσεις.

Στο 5ο κεφάλαιο θα αναλυθεί η υλοποίηση της εφαρμογής, ενώ πρώτα θα αναφερθούμε στα λογισμικά που εφαρμόστηκαν -με τις απαραίτητες πάντα αιτιολογήσεις- και στη συνέχεια θα παρουσιαστούν διάφορα στιγμιότυπα από τη λειτουργία της. Επίσης,το παράρτημα της εργασίας είναι απαραίτητο για την καλύτερη κατανόηση, ακρίβεια και σαφήνεια σε όσα αναφέρονται στην παρούσα διπλωματική. Συνοψίζοντας, στο 6ο κεφάλαιο θα παρουσιαστούν τα συμπεράσματα που προέκυψαν από την αξιολόγηση, καθώς και την ίδια την διαδικασία, ενώ στο 7ο κεφάλαιο θα γίνει αναφορά σε μελλοντικές ιδέες, επεκτάσεις της εφαρμογής και άλλα συμπεράσματα βελτιστοποίησης.

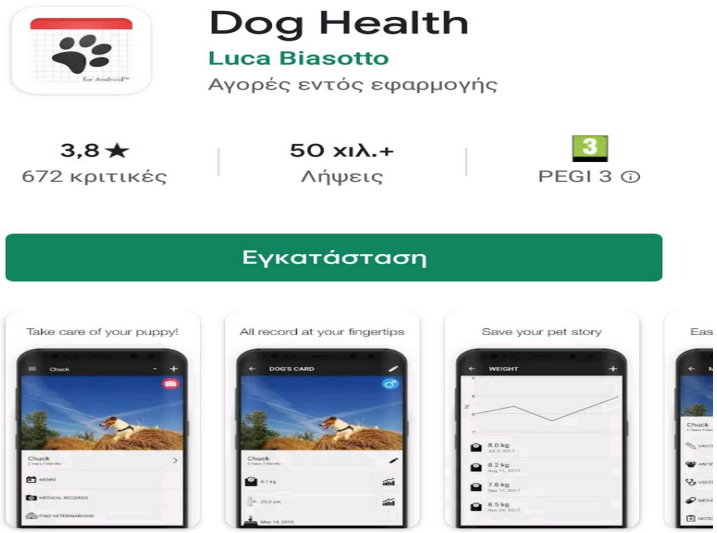
2. ΕΡΕΥΝΑ ΣΧΕΤΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

Σε αυτό το κεφάλαιο θα ξεκινήσει το πρώτο επίπεδο σύμφωνα με το μοντέλο του καταρράκτη, δηλαδή η έρευνα και αναζήτηση σχετικών-παρόμοιων εφαρμογών που έχουν ήδη αναπτυχθεί. Στην έρευνα αυτή θα ληφθούν υπόψη εφαρμογές αναπτυγμένες για κινητές και για σταθερές συσκευές με κύριο και βασικό θέμα την εξυπηρέτηση των ιδιοκτητών των σκύλων. Με αυτή την έρευνα θα γίνει σαφής η αναγκαιότητα ύπαρξης μιας τέτοιας εφαρμογής για τους ιδιοκτήτες και τα κατοικίδια σκυλιά τους. Τέλος, θα ακολουθήσει η παρουσίαση ενός πίνακα (Πίνακας Σύγκρισης Εφαρμογών) ο οποίος θα παρουσιάζει και θα συγκρίνει τις δυνατότητες κάθε εφαρμογών ξεχωριστά αλλά και τις απαιτήσεις του προβλήματος που ήδη τέθηκε.

2.1 Εφαρμογές κινητών συσκευών

Κατά την αναζήτηση εφαρμογών για συσκευές κινητών επικεντρωθήκαμε σε εφαρμογές οι οποίες έχουν αναπτυχθεί και σχεδιαστεί με βάση τα επικρατέστερα λειτουργικά συστήματα, δηλαδή συστήματα iOS και ANDROID. Αρχικά, όσον αφορά λειτουργικά συστήματα Android, θα παρουσιαστούν εφαρμογές από το play store που έχουν περιεχόμενο παρόμοιο με εκείνο της παρούσας διπλωματικής εργασίας αλλά και εφαρμογές γενικού περιεχομένου όσον αφορά Dog Apps και Pet related. Οι εφαρμογές αυτές, βρέθηκαν στο ηλεκτρονικό κατάστημα της Google κατά την αναζήτηση με χρήση λέξεων-κλειδιών όπως “Dog app”, ”Pet Shops”, ”Animal Health” κ.λπ. Παρόμοια αναζήτηση έγινε για το λειτουργικό σύστημα iOS, χρησιμοποιώντας τις ίδιες λέξεις κλειδιά στο αντίστοιχο κατάστημα της Apple, χωρίς όμως να προκύψει κάποιο αποτέλεσμα.

2.1.1 Dog Health



Σχήμα 2.1: Η εφαρμογή ”Dog Health” στο Google Play

Η εφαρμογή “Dog Health” δίνει στο χρήστη την δυνατότητα να εντοπίσει και να επικοινωνήσει με το πλησιέστερο κτηνιατρείο σε περίπτωση ανάγκης ή απλά για ένα κλασικό check up. Ακόμα του παρέχεται η δυνατότητα να παρακολουθεί όλες τις προηγούμενες επισκέψεις του στον κτηνιατρείο, καθώς και να αποθηκεύσει τα προσωπικά δεδομένα του σκύλου του, όπως το όνομα, την ημερομηνία γέννησης άλλα και άλλες βασικές πληροφορίες. Η εφαρμογή, έχει ως κύριο στόχο την υγεία του σκύλου, τη βελτίωση της παρακολούθησης, της αναζήτησης και της ένταξης που θα χρειαστεί ένας νέος ή και παλιός ιδιοκτήτης σκύλου κατά την διάρκεια ανατροφής του.

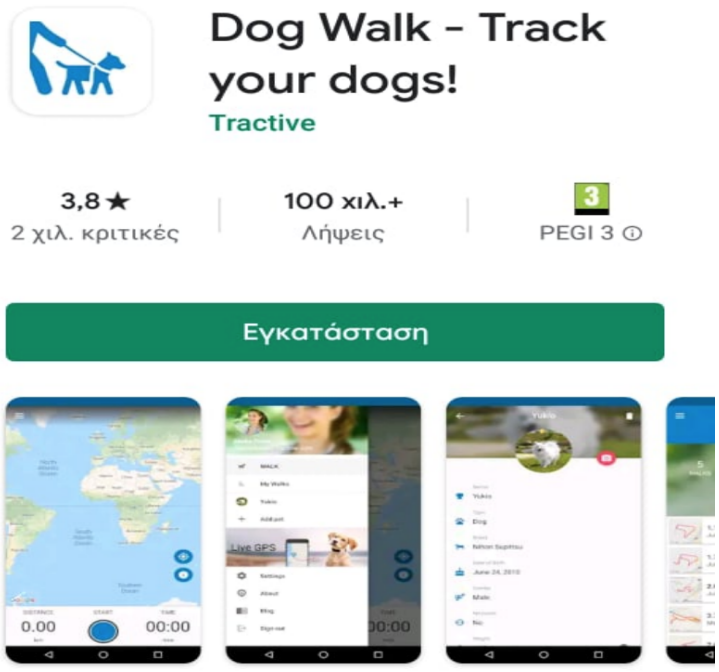
Πρέπει ωστόσο να σημειωθεί ότι η εφαρμογή απαιτεί έκδοση Android 8.0 ή ακόμα και νεότερη, έχει μέγεθος 4.84 Mb και βρίσκεται στην 2.0.12 έκδοση με την τελευταία ενημέρωση στις 27 Ιουλίου 2019. Για τη λειτουργία της, η εφαρμογή απαιτεί πρόσβαση στα κάτωθι στοιχεία της συσκευής:

- Στην τοποθεσία της συσκευής
  - κατά προσέγγιση θέση (βάσει δικτύου)
  - ακριβής θέση (GPS και βάσει δικτύου)
- Στον αποθηκευτικό χώρο
  - Τροποποίηση ή διαγραφή περιεχομένων κάρτας SD
  - ανάγνωση του περιεχομένου της κάρτας SD
- Και σε άλλα
  - έχει πλήρη πρόσβαση στο δίκτυο
  - ελέγχει την δόνηση
  - αποτρέπει το τηλέφωνο να μεταβεί σε κατάσταση αδράνειας

Η εφαρμογή έχει 50.000+ λήψεις και οι κριτικές από χρήστες είναι κατά κύριο λόγο αρκετά καλές. Αναλυτικά παρατηρούμε στον Πίνακα 2.1 (Πίνακας Σύγκρισις Εφαρμογών) τα στοιχεία της που μας ενδιαφέρουν. Το περιβάλλον έχει σχετικά φωτεινά χρώματα και απαιτεί σύνδεση σε WIFI ή 4G. Ταυτόχρονα, στην κεντρική σελίδα υπάρχουν διάφορες επιλογές στις οποίες έχει πρόσβαση ο χρήστης, αφού πρώτα συνδεθεί στην εφαρμογή. Η εφαρμογή κάνει χρήση τεχνολογιών διαδικτύου και αξιοποιεί σε έναν βαθμό τις δυνατότητες των κινητών συσκευών. Ακόμη, πρέπει να ειπωθεί ότι

περιορίζεται μόνο στην υγειά και δεν διευκολύνει σε μεγάλο βαθμό την κοινότητα τον ανθρώπων με κατοικίδια σκυλιά.

2.1.2 Dog Walk - Track your dogs!



Σχήμα 2.2: Η εφαρμογή ”Dog Walk - Track your dogs!” στο Google Play

Η εφαρμογή “Dog Walk – Track your dogs!” δίνει στο χρήστη την δυνατότητα να εντοπίσει, να παρακολουθήσει, να γυμναστεί και ο ίδιος και ακόμα και να καταγράψει την διαδρομή, την απόσταση και την διάρκεια της κάθε βόλτας με τον σκύλο του για να μπορέσει στο τέλος της ημέρας να δει όλη την άσκηση που ακολουθεί ο σκύλος του κάθε μέρα. Επιπρόσθετα, του παρέχεται η δυνατότητα να παρακολουθεί όλες τις προηγούμενες βόλτες και περιπέτειες που έχει ζήσει με τον σκύλο καθώς και να τραβήξει φωτογραφίες από τις βόλτες αυτές και να τις μοιράσει με τους φίλους του και την οικογένεια του. Η εφαρμογή αυτή είναι ένας πολύ εύχρηστος τρόπος για να δημιουργήσει κάποιος αναμνήσεις, να ανακαλύψει καινούργιους τόπους και να έχει έλεγχο και οργάνωση στην αθλητική ζωή του σκύλου του χωρίς να το παραμελεί και να το καθιστά μονότονο.

Σε αυτή την περίπτωση, η εφαρμογή απαιτεί έκδοση Android 6 ακόμα και νεότερη, έχει μέγεθος 12.08 Mb και βρίσκεται στην 1.2.3 έκδοση με την τελευταία ενημέρωση στις 26 Σεπτεμβρίου 2019. Επίσης πρέπει να αναφερθεί ότι για την εύρυθμη λειτουργία της, η εφαρμογή απαιτεί πρόσβαση στα κάτωθι στοιχεία της συσκευής:

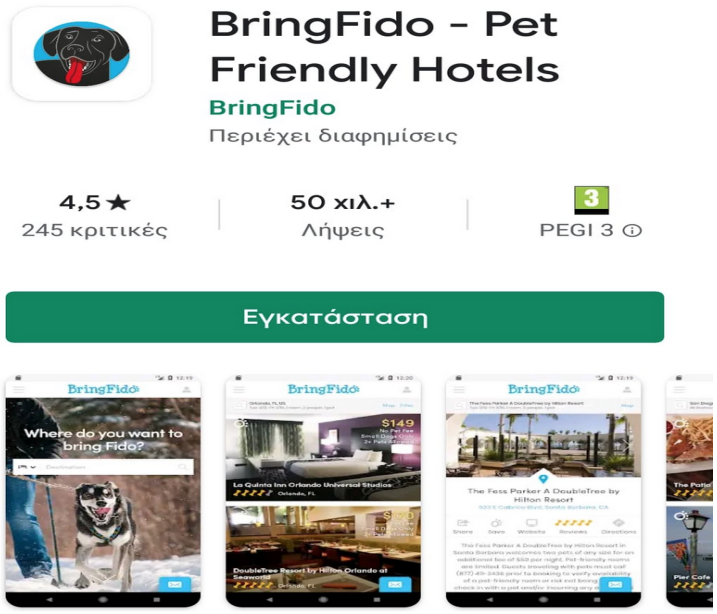
- Στην τοποθεσία της συσκευής
  - κατά προσέγγιση θέση (βάσει δικτύου)
  - ακριβής θέση (GPS και βάσει δικτύου)
- Στον αποθηκευτικό χώρο
  - Τροποποίηση ή διαγραφή περιεχομένων κάρτας SD
  - ανάγνωση του περιεχομένου της κάρτας SD
- Επαφές

  - βρίσκει λογαριασμούς στη συσκευή

- Φωτογραφική μηχανή
  - Κάνει λήψη φωτογραφιών και βίντεο

Η εφαρμογή έχει 100.000+ λήψεις και οι κριτικές από χρήστες είναι κατά κύριο λόγο καλές με βαθμολογία 3.8/5. Αναλυτικά παρατηρούμε στον Πίνακα 2.1 (Πίνακας Σύγκρισης Εφαρμογών) τα στοιχεία της που μας ενδιαφέρουν. Το περιβάλλον χρήσης έχει σχετικά φωτεινά χρώματα και απαιτεί σύνδεση σε WIFI ή 4G. Επίσης, στην κεντρική σελίδα υπάρχει ένας χάρτης με κουμπί εκκίνησης χρονομέτρου αφού ξεκινήσει η βόλτα. Επιπλέον, στην πάνω αριστερά μεριά της οθόνης υπάρχει κουμπί επιλογής λίστας που σε προωθεί σε διάφορες επιλογές και ενέργειες. Σε όλα τα παραπάνω θα έχει πρόσβαση ο χρήστης, αφού όμως πρώτα συνδεθεί στην εφαρμογή, όπως ήδη υπαινίχθηκε. Σε αυτό το σημείο να σημειωθεί πως η εφαρμογή κάνει χρήση και τεχνολογιών διαδικτύου(Maps), αξιοποιεί και σε έναν βαθμό τις δυνατότητες των κινητών συσκευών αλλά περιορίζεται μόνο στην βόλτα του σκύλου και λείπουν και άλλες βασικές πληροφορίες, όπως τοποθεσίες πάρκων για σκυλιά, τα οποία θα διευκόλυναν σε γενικό βαθμό την κοινότητα τον ανθρώπων με κατοικίδια σκυλιά.

2.1.3 BringoFido-Pet Friendly Hotels



Σχήμα 2.3: Η εφαρμογή ”BringoFido-Pet Friendly Hotels” στο Google Play

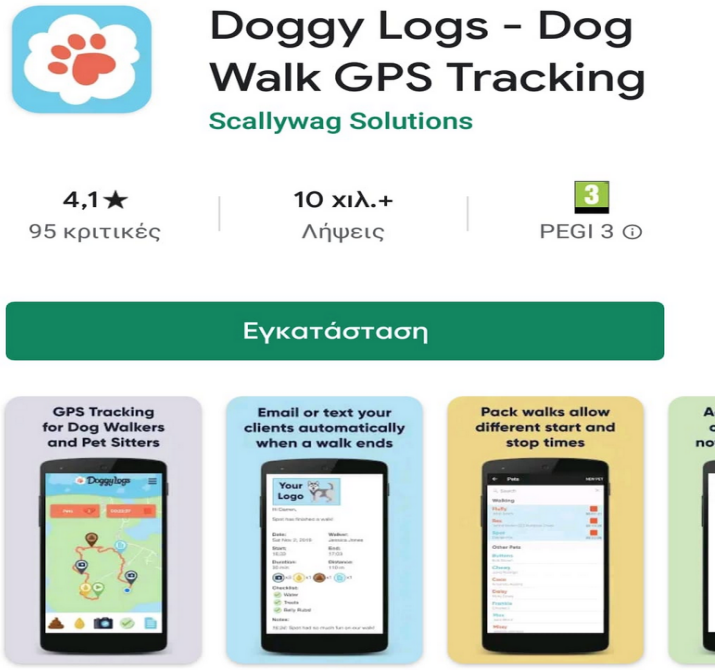
Η εφαρμογή “ BringoFido-Pet Friendly Hotels” δίνει στους χρήστες την δυνατότητα να αναζητήσουν και να κάνουν κράτηση σε ένα φιλικό προς τα κατοικίδια ξενοδοχείο, να εντοπίσουν και να βρουν αξιοθέατα και εστιατόρια που δέχονται κατοικίδια. Ακόμα, τους παρέχεται η δυνατότητα να παρακολουθήσουν κριτικές και σχόλια από άλλους χρήστες με αποτέλεσμα να φιλτράρουν την επιλογή τους και να κατασταλάξουν σε αυτό που τους ταιριάζει περισσότερο. Επίσης,καθίσταται εύκολο για τον χρήστη να εντοπίσει μέσα από τον χάρτη κοντινές τοποθεσίες ξενοδοχείων και εστιατορίων στην περιοχή που βρίσκεται αλλά και να ζήσει μοναδικές εμπειρίες σε προσιτές τιμές από τις εκπτώσεις που θα παρέχει η εφαρμογή. Τέλος, μέσα από την συγκεκριμένη εφαρμογή μπορεί κάποιος να ανακαλύψει καινούργια ή ήδη υπάρχοντα ξενοδοχεία, εστιατόρια και αξιοθέατα τα οποία είναι φιλικά προς τα κατοικίδια κοντά στην περιοχή τους ή και σε απόμακρες περιοχές όπου και θα μπορούν να επισκεφθούν μαζί με τα κατοικίδια τους.

Η εφαρμογή απαιτεί έκδοση Android 7 και νεότερες, έχει μέγεθος 2.24 Mb και βρίσκεται στην 1.1.1 έκδοση με την τελευταία ενημέρωση στις 12 Δεκεμβρίου 2019. Για την λειτουργία της η εφαρμογή απαιτεί πρόσβαση στα κάτωθι στοιχεία της συσκευής:

- Στον αποθηκευτικό χώρο
  - Τροποποίηση ή διαγραφή περιεχομένων κάρτας SD
  - ανάγνωση του περιεχομένου της κάρτας SD
- Και άλλα
  - έχει πλήρη πρόσβαση στο δίκτυο
  - βλέπει τις συνδέσεις δικτύου

Η εφαρμογή έχει 50.000+ λήψεις και οι κριτικές από χρήστες είναι αρκετά καλές με βαθμολογία 4.5/5. Αναλυτικά παρατηρούμε στον Πίνακα 2.1 (Πίνακας Σύγκρισης Εφαρμογών) τα στοιχεία της που μας ενδιαφέρουν. Το περιβάλλον χρήσης έχει αρκετά φωτεινά χρώματα και απαιτεί σύνδεση σε WIFI ή 4G, στην κεντρική σελίδα υπάρχει η δυνατότητα να επιλέξεις το μέρος που θέλεις να επισκεφτείς και πάνω αριστερά έχει ένα κουμπί επιλογής λίστας με διάφορες επιλογές και ενέργειες. Σε όλα τα παραπάνω δεν είναι απαραίτητη η σύνδεση του χρήστη στην εφαρμογή για να έχει πρόσβαση. Η εφαρμογή κάνει χρήση τεχνολογιών διαδικτύου(π.χ. Maps) και αξιοποιεί σε έναν βαθμό τις δυνατότητες των κινητών συσκευών. Ωστόσο μειονέκτημα αποτελεί το γεγονός ότι περιορίζεται μόνο στα ξενοδοχεία, εστιατόρια και αξιοθέατα ενώ θα μπορούσαν να υπάρξουν και άλλες βασικές πληροφορίες, όπως τοποθεσίες για κτηνιατρεία και pet shops, τα οποία μπορεί να χρειαστούν οι χρήστες κατά την διαμονή τους στο ξενοδοχείο και την επίσκεψη τους σε αυτές τις τοποθεσίες.

2.1.4 Doggy Logs-Dog Walk GPS Tracking



Σχήμα 2.4: Η εφαρμογή ”Doggy Logs-Dog Walk GPS Tracking” στο Google Play

Η εφαρμογή “ Doggy Logs-Dog Walk GPS Tracking” δίνει την δυνατότητα στους χρήστες να παρακολουθήσουν την βόλτα του σκύλου τους και να φωτογραφίσουν τις περιπέτειες και τις στιγμές για να τις μοιράσουν με τους φίλους και την οικογένεια τους. Επίσης, τους παρέχεται η επιλογή να

καταγράψουν και να αποθηκεύσουν διαδρομές καθώς και πληροφορίες σχετικά με τις ανάγκες που είχε ο σκύλος κατά την διάρκεια της βόλτας. Μέσα από την εφαρμογή μπορεί κάποιος να ανακαλύψει καινούργιες ή ήδη υπάρχουσες τοποθεσίες πάρκων και επιλογές περιπάτων, τις οποίες πιθανόν δεν έχει ανακαλύψει και κατά συνέπεια επισκεφθεί ενώ παράλληλα είναι εφικτό να γίνει έλεγχος της αθλητικής κατάστασης και της υγείας του σκύλου.

Η εφαρμογή απαιτεί έκδοση Android 6+ και νεότερες, έχει μέγεθος 21.90 Mb και βρίσκεται στην 3.1 έκδοση με την τελευταία ενημέρωση στις 8 Δεκεμβρίου 2019. Για την λειτουργία της η εφαρμογή απαιτεί πρόσβαση στα κάτωθι στοιχεία της συσκευής:

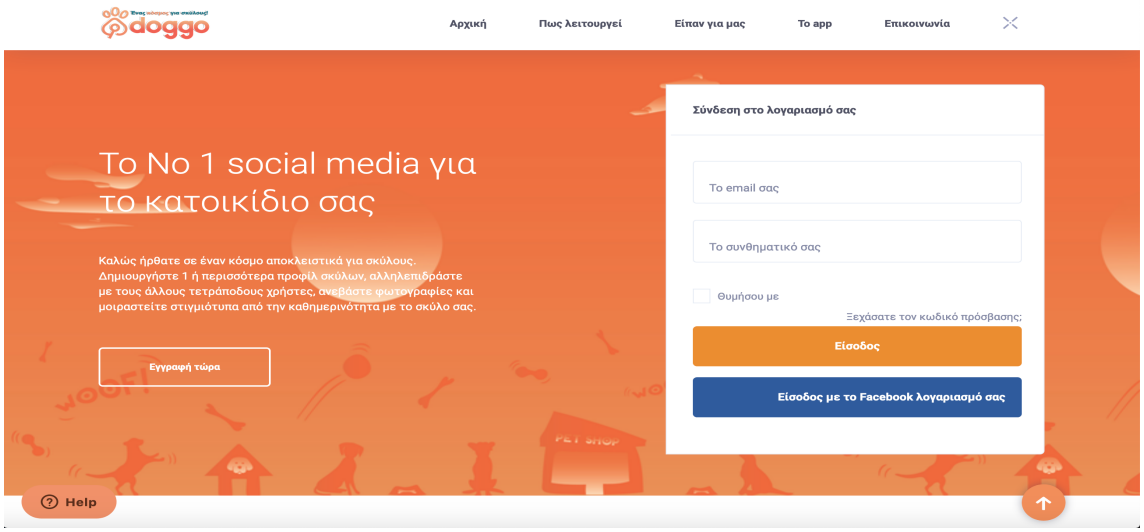
- Στην τοποθεσία της συσκευής
  - κατά προσέγγιση θέση (βάσει δικτύου)
  - ακριβής θέση (GPS και βάσει δικτύου)
- Αποθηκευτικός χώρος
  - Τροποποίηση ή διαγραφή περιεχομένων κάρτας SD
  - ανάγνωση του περιεχομένου της κάρτας SD
- Άλλα
  - έχει πλήρης πρόσβαση στο δίκτυο
  - βλέπει τις συνδέσεις δικτύου
  - ανάγνωση διαμόρφωσης υπηρεσιών Google

Η εφαρμογή έχει 10.000+ λήψεις και οι κριτικές από χρήστες είναι καλές με συνολική βαθμολογία 4.1/5. Αναλυτικά παρατηρούμε στον Πίνακα 2.1 (Πίνακας Σύγκρισης Εφαρμογών) τα στοιχεία της που μας ενδιαφέρουν. Το περιβάλλον χρήσης έχει σχετικά φωτεινά χρώματα και απαιτεί σύνδεση σε WIFI ή 4G, στην κεντρική σελίδα υπάρχει η επιλογή να επιλέξεις μια από τις πέντε ενέργειες και κατά την επιλογή θα μπορείς να μεταφερθείς στην αντίστοιχη σελίδα. Όπως ήδη σημειώθηκε, σε όλα τα παραπάνω είναι απαραίτητη η σύνδεση του χρήστη στην εφαρμογή για να έχει πρόσβαση. Η εφαρμογή κάνει χρήση και τεχνολογιών διαδικτύου (π.χ. Maps), αξιοποιεί σε έναν βαθμό τις δυνατότητες των κινητών συσκευών. Από μειονεκτήματα θα πρέπει να αναφερθεί το γεγονός ότι η εφαρμογή περιορίζεται μόνο στην καταγραφή της βόλτας και της ”ανάγκης” του σκύλου και τη στιγμή που υπάρχουν ελλείψεις σε άλλα σημαντικά ζητήματα, όπως η εμφάνιση τοποθεσιών για πάρκα στα οποία το σκυλί θα είναι ελεύθερο να παίξει και να κινηθεί, πράγμα πολύ σημαντικό για μια κοινότητα ανθρώπων με κατοικίδια σκυλιά.

2.2 Ιστοσελίδες και εφαρμογές για σταθερούς υπολογιστές

Για να μπορέσουμε να δείξουμε τις ευκολίες και τις δυνατότητες που μπορούν να προσφέρουν οι τεχνολογίες διαδικτύου σε μια κοινότητα ανθρώπων με κατοικίδια σκυλιά πρέπει να γίνει έρευνα σε ιστοσελίδες αλλά και παρόμοιες εφαρμογές και σε σταθερούς υπολογιστές. Έτσι, δίνοντας εξίσου έμφαση στις δυνατότητες και υπηρεσίες που θα αναφερθούν σε αυτές τις ιστοσελίδες, θα μπορέσουμε να ενισχύσουμε και να ισχυροποιήσουμε την ανάγκη για ύπαρξη αντίστοιχων εφαρμογών σε κινητές συσκευές.

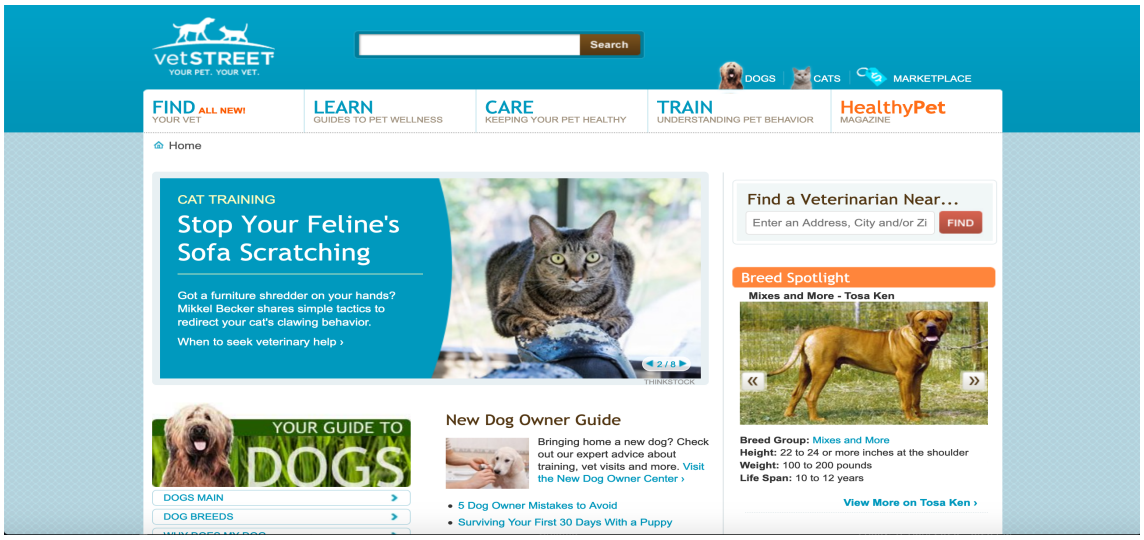
2.2.1 Doggo



Σχήμα 2.5: Στιγμιότυπο από την ιστοσελίδα του Doggo

Το Doggo διαθέτει την ιστοσελίδα www.doggo.gr, το οποίο αποτελεί ένα από τα πιο ενδεικτικά παραδείγματα εφαρμογών αυτής της διπλωματικής εργασίας. Η ιστοσελίδα αυτή παρέχει ένα εύρος δυνατοτήτων, όπως εύρεση τοποθεσιών και αναζήτηση σημείων ενδιαφέροντος σε όλη την Ελλάδα για παράδειγμα πάρκα για σκύλους, Pet Shop, Κτηνιατρεία, Κέντρα Θετικής Εκπαίδευσης, Φιλοζωικά Σωματεία, Ξενοδοχεία Σκύλων, Κομμωτήρια Σκύλων, Ταξί για κατοικίδια και Ζωόφιλα/ Pet Friendly Μαγαζιά. Επίσης, μια άλλη παροχή της εν λόγω ιστοσελίδας είναι η εύρεση και αποθήκευση φίλων/ σκυλιών για το κατοικίδιο, με τα οποία μπορεί να παίζει και να πηγαίνει βόλτα. Επιπρόσθετη λειτουργία του site είναι η εμφάνιση ψυχαγωγικών εκδηλώσεων αποκλειστικά για ιδιοκτήτες σκύλων, με πρωτότυπες δράσεις για σκύλους, με άξονα τη φιλοζωία γεμάτο pet friendly δράσεις, σκυλάκια, αμέτρητους επισκέπτες, φιλοζωικές οργανώσεις και καταξιωμένες επιχειρήσεις από τον κλάδο των ζώων συντροφιάς. Τέλος τα στοιχεία που μας ενδιαφέρουν φαίνονται στον Πίνακα 2.1 (Πίνακας Σύγκρισις Εφαρμογών).

2.2.2 Vetstreet



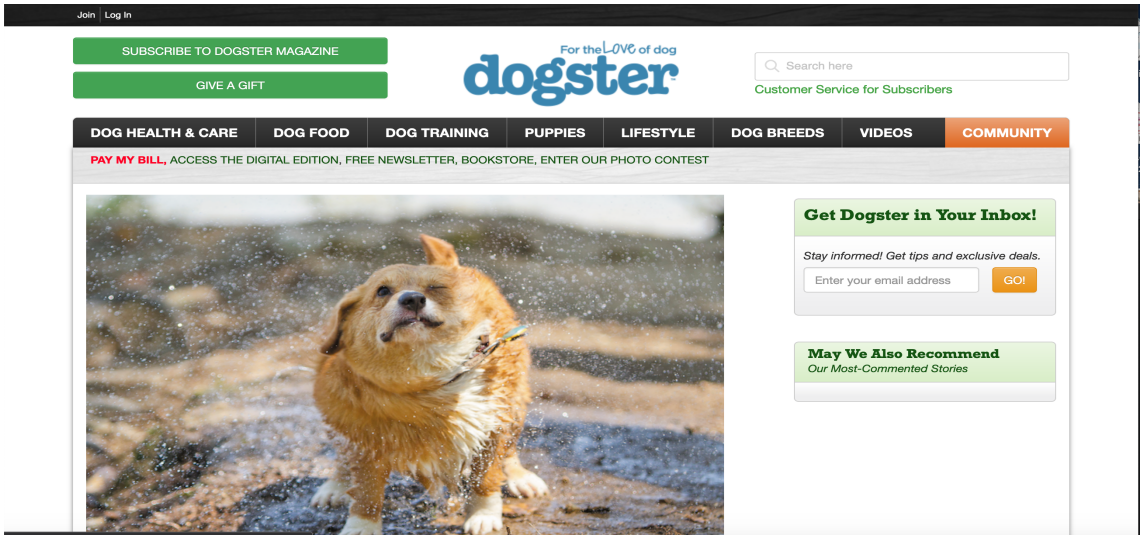
Σχήμα 2.6: Στιγμιότυπο από την ιστοσελίδα του Vetstreet

Το Vetstreet διαθέτει την ιστοσελίδα http://www.vetstreet.com/ η οποία παρέχει όλες τις πληροφορίες που θα χρειαστεί κάποιος για την φροντίδα ενός σκυλού. Η ιστοσελίδα αυτή περιέχει οδηγίες για θέματα υγείας των κατοικίδιων, όπως το πως να παρέχεται στον σκύλο μια υγιέστατη ζωή, άρθρα πάνω σε θέματα εκπαίδευσης και κατανόησης συμπεριφορών σκύλων καθώς και πληροφορίες σχετικές με την σωστή διατροφή που πρέπει να ακολουθεί ο σκύλος. Επίσης, παρατηρείται η επιλογή "Find Your Vet" ή οποία βρίσκει και εντοπίζει τα κοντινότερα στην περιοχή κτηνιατρεία και αντίστοιχα κτηνίατρους.

Αρκετά σημαντικές είναι οι πληροφορίες που μπορεί να συλλέξει κανείς έχοντας περιηγηθεί και μελετήσει άρθρα και σχόλια από αυτήν την ιστοσελίδα. Τέτοια, είναι τα θετικά και αρνητικά, οι διαφορές μεταξύ κάλων και κακών κτηνιατρείων αλλά και οι αξιολογήσεις για τους γιατρούς των σκύλων.

Σε αυτά μπορούν να προστεθούν οι τοποθεσίες και ονομασίες των καταστημάτων και κτηνιάτρων που διαθέτουν συγκεκριμένο εξοπλισμό και την απαραίτητη φροντίδα που ψάχνουν, έτσι ώστε στη συνέχεια να ενημερωθούν για την διαθεσιμότητα σε ραντεβού και να κλείσουν τελικά το δικό τους. Τέλος, οι παραπάνω πληροφορίες αλλά και οι αμέτρητες πληροφορίες που βρίσκονται σε αυτή η ιστοσελίδα, καθιστούν αναγκαία την μεταφορά της σε εφαρμογή κινητής συσκευής για να γίνει η καλύτερη εκμετάλλευση αυτών τον πόρων. Αναλυτικά τα στοιχεία της που μας ενδιαφέρουν φαίνονται στον Πίνακα 2.1 (Πίνακας Σύγκρισης Εφαρμογών).

2.2.3 Dogster

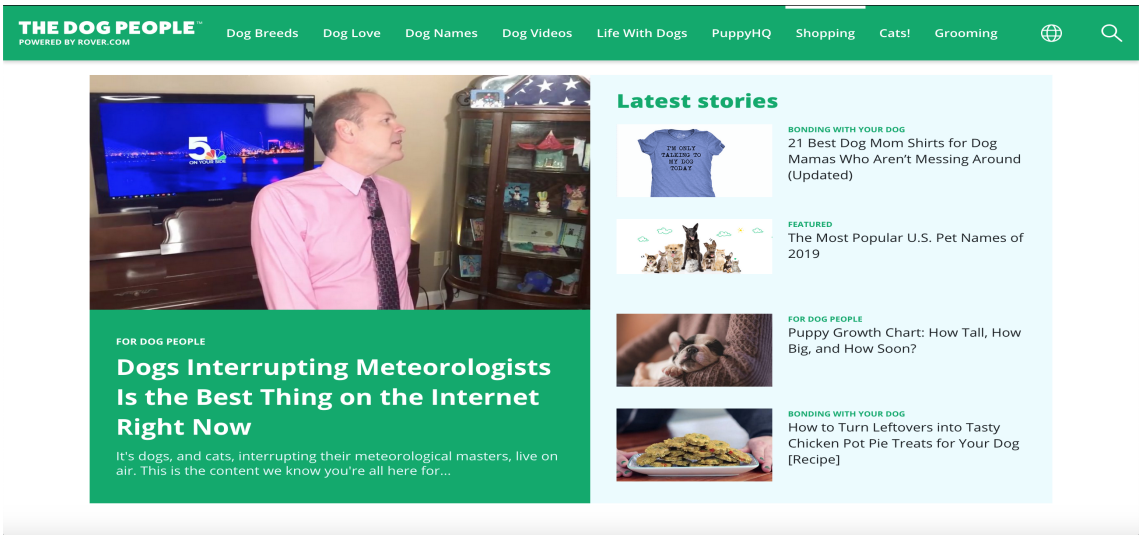


Σχήμα 2.7: Στιγμιότυπο από την ιστοσελίδα του Dogster

Η ιστοσελίδα https://www.dogster.com/ πρόκειται για μια ιστοσελίδα που ενημερώνει και πληροφορεί ιδιοκτήτες με κατοικίδια σκυλιά για ποικίλα θέματα πάνω στην σωστή ανάθρεψη ενός σκύλου. Πιο συγκεκριμένα, δίνεται στον χρήστη η επιλογή να ενημερωθεί και να μελετήσει άρθρα που αφορούν την υγεία του σκυλιού σε συνδυασμό με μέρη μπορεί να επισκεφθεί,έτσι ώστε να δεχτεί τις αντίστοιχες υπηρεσίες υγείας για τον σκύλο του. Στην συνέχεια, ο χρήστης μπορεί να μάθει για τις διάφορες διαθέσιμες σκυλοτροφές και τα είδη φαγητών που υπάρχουν, τα κατάλληλα παιχνίδια και τα απαραίτητα αξεσουάρ για ένα σκυλί.

Όμως, οι σημαντικότερες πληροφορίες που μπορεί να συλλέξει κανείς από αυτήν την ιστοσελίδα είναι τα θετικά και αρνητικά μεταξύ καταστημάτων Pet Shop, κτηνιατρεία και πάρκα για σκυλιά, τις τοποθεσίες και ονομασίες των καταστημάτων που διαθέτουν το συγκεκριμένο προϊόν που ψάχνουν, έτσι ώστε να ενημερωθούν σχετικά με τη διαθεσιμότητα στο αντίστοιχο κατάστημα της περιοχής τους. Τέλος, οι αμέτρητες πληροφορίες που παρέχει αυτή η ιστοσελίδα στην κοινότητα για την οποία συζητάμε, καθιστά αναγκαία την μεταφορά της σε μια εφαρμογή κινητής συσκευής, έτσι ώστε να γίνει καλύτερη εκμετάλλευση αυτών τον πόρων. Αναλυτικά τα στοιχεία της που μας ενδιαφέρουν φαίνονται στον Πίνακα 2.1 (Πίνακας Σύγκρισης Εφαρμογών).

2.2.4 The Dog People by Rover.com



Σχήμα 2.8: Στιγμιότυπο από την ιστοσελίδα του The Dog People by Rover.com

Το The Dog People by Rover.com διαθέτει την ιστοσελίδα https://www.ro-ver.com/blog/ . Πρόκειται για μια ιστοσελίδα που ενημερώνει και πληροφορεί ιδιοκτήτες με κατοικίδια σκυλιά για πολλά θέματα που αφορούν τους κατόχους σκύλων. Πιο συγκεκριμένα, ο χρήστης έχει τη δυνατότητα να μελετήσει άρθρα που αφορούν τις ράτσες σκύλων, ειδικά αξεσουάρ για σκύλους και αξεσουάρ που εξυπηρετούν τους ιδιοκτήτες τις στιγμές που περνούν μαζί με τους σκύλους τους καθώς και άρθρα σχετικά με καταστήματα που παρέχουν τα αντίστοιχα προϊόντα. Πάνω σε αυτό, ο χρήστης μπορεί να διευρύνει τις γνώσεις του για τις ποικιλίες σκυλοτροφών και τα κατάλληλα είδη φαγητών, τα επιθυμητά παιχνίδια και τα είδη πρώτης ανάγκης.

Ακολούθως, σημαντικές πληροφορίες μπορεί να συλλέξει κανείς έχοντας περιηγηθεί και μελετήσει τα άρθρα και τα σχόλια από αυτήν την συγκεκριμένη ιστοσελίδα . Πιο αναλυτικά, θα ενημερωθεί για τα θετικά και αρνητικά καταστημάτων, τοποθεσιών, και πολλά άλλα, αλλά και τις διαφορές μεταξύ των κάλων και κακών καταστημάτων και pet shop. Τέλος, οι παραπάνω πληροφορίες και οι αμέτρητες πληροφορίες που παρέχει αυτή η ιστοσελίδα, ιδιαίτερα στην κοινότητα των ανθρώπων με κατοικίδια σκυλιά, καθιστά αναγκαία τη μεταφορά της σε μια εφαρμογή κινητής συσκευής για να γίνει η καλύτερη εκμετάλλευση αυτών τον πόρων. Αναλυτικά τα στοιχεία της που μας ενδιαφέρουν φαίνονται στον Πίνακα 2.1 (Πίνακας Σύγκρισις Εφαρμογών).

No.	Όνομα	Smartphone	Διαδίκτυο	Σχόλια	Εύρεση Τοποθεσιών	Περιήγηση στην Τοποθεσία	Πληροφορίες για βασικές ανάγκες σκυλιού (π.χ Βόλτα,υγεία,τροφή κ.λπ.)
1	DogHealth	ΝΑΙ	ΟΧΙ	ΟΧΙ	ΝΑΙ	ΟΧΙ	ΜΟΝΟ ΣΤΗΝ ΥΓΕΙΑ
2	DogWalk-Trackyourdogs!	ΝΑΙ	ΟΧΙ	ΝΑΙ	ΝΑΙ	ΟΧΙ	ΟΧΙ
3	BringoFido-PetFriendlyHotels	ΝΑΙ	ΜΕΡΙΚΩΣ (ΥΠΑΡΞΗ ΠΑΡΟΜΟΙΑΣ)	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΜΟΝΟ ΞΕΝΟΔΟΧΕΊΑ,ΕΣΤΙΑΤΌΡΙΑ ΚΑΙ ΑΞΙΟΘΈΑΤΆ
4	DoggyLogs-DogWalkGPSTracking	ΝΑΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ
5	Doggo	ΜΕΡΙΚΩΣ (ΥΠΑΡΞΗ ΠΑΡΟΜΟΙΑΣ)	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΜΕΡΙΚΩΣ
6	Vetstreet	ΟΧΙ	ΝΑΙ	ΝΑΙ	ΟΧΙ	ΟΧΙ	ΜΟΝΟ ΣΤΗΝ ΥΓΕΙΑ
7	Dogster	ΟΧΙ	ΝΑΙ	ΝΑΙ	ΟΧΙ	ΟΧΙ	ΜΟΝΟ ΣΤΗΝ ΥΓΕΙΑ ΚΑΙ ΠΡΟΙΟΝΤΑ/ΑΞΕΣΟΥΑΡ
8	TheDogPeoplebyRover.com	ΟΧΙ	ΝΑΙ	ΝΑΙ	ΟΧΙ	ΟΧΙ	ΜΟΝΟ ΣΤΗΝ ΥΓΕΙΑ ΚΑΙ ΠΡΟΙΟΝΤΑ/ΑΞΕΣΟΥΑΡ

Table 2.1: Πίνακας Σύγκρισης Εφαρμογών

3. ΑΝΑΛΥΣΗ

Στο κεφάλαιο αυτό, χρησιμοποιώντας το μοντέλο του καταρράκτη, θα πραγματοποιηθεί ανάλυση της εφαρμογής . Πιο συγκεκριμένα, σε αυτό το βήμα, υλοποιείται η οργάνωση και η συγκέντρωση των παραπάνω προδιαγραφών της κάθε εφαρμογής στην παρούσα διπλωματική εργασία, ενώ στη συνέχεια θα ακολουθήσει ανάλυση σχετικά με τους χρήστες στους οποίους απευθύνεται η συγκεκριμένη εφαρμογή και κατά βάση μελέτη των αναγκών τους.

3.1 Συγκέντρωση προδιαγραφών

Για να αναλύσουμε σε βάθος την εφαρμογή είναι απαραίτητο πρώτα να γίνει σαφές το εύρος των αναγκών και δυνατοτήτων που θέλουμε να καλύψει. Αρχική και ίσως καθοριστική βοήθεια για την ανακάλυψη των βασικών απαιτήσεων-αναγκών που έχει ένας κάτοχος σκύλου, προήλθε κυρίως από άτομα του κοντινού περίγυρου με κατοικίδια σκυλιά. Αυτά, σε συνδυασμό με τα προηγούμενα κεφάλαια καθόρισαν τις βασικές προδιαγραφές της εφαρμογής που επιλέχθηκαν με κριτήριο τρείς παράγοντες:

- Ανάπτυξη των τεχνολογιών πληροφορίας
- Ανάπτυξη των τεχνολογιών ιστού
- Ανάλυση των δυνατοτήτων των κινητών συσκευών

3.1.1 Τεχνολογίες πληροφοριών για ιδιοκτήτες σκύλων

Κατά τη διάρκεια μίας ημέρας, οι ιδιοκτήτες οφείλουν να είναι ανά πάσα στιγμή σε θέση να αναζητήσουν τοποθεσίες για τα σκυλιά τους, καφετέριες που είναι φιλικές προς αυτά, καταστήματα όπως Pet Shops, ακόμα και κτηνιατρεία ή κτηνίατρους προκειμένου να είναι σε θέση να καλύπτουν τις ανάγκες των σκύλων κάθε χρονική στιγμή. Αφού επισκεφθούν τις τοποθεσίες που έχουν επιλέξει μπορούν να τις σχολιάσουν και να τις βαθμολογήσουν. Οι τοποθεσίες αυτές, που θα συνοδεύονται με τη γεωγραφική θέση και άλλες πληροφορίες (όπως οι κριτικές) είναι ορατές και σε άλλους χρήστες μέσω της προσωπικής τους αναζήτησης στην ίδια περιοχή.

Είναι απαραίτητη επίσης μια βάση δεδομένων για την επίτευξη όλων των προαναφερθέντων, που θα αποθηκεύει πληροφορίες για τα πάρκα, τις pet friendly καφετέριες, τα αντίστοιχα Pet Shops και κτηνιατρεία. Επιπλέον σημαντικό στοιχείο της βάσης αποτελεί η ύπαρξη των πληροφοριών του χρήστη, ώστε ο κάθε ιδιοκτήτης να βρίσκεται σε θέση να αποθηκεύσει και να μπορεί να αλλάζει τα προσωπικά στοιχεία του όπως username, κωδικούς ακόμα και τα σχόλια που επιθυμεί, με σκοπό την ολοένα και καλύτερη ενημέρωση της εφαρμογής και την πλοήγησή των χρηστών στις πιο υψηλά βαθμολογημένος τοποθεσίες.

Ένα σημαντικό και αναπόσπαστο σημείο αποτελεί το γεγονός ότι οι χρήστες θα πρέπει να είναι ειλικρινείς και δίκαιοι κριτές για τις τοποθεσίες που επισκέπτονται, έτσι ώστε να μπορεί όποιος διαβάζει τα σχόλια να έχει μια συλλογική και αληθινή εικόνα για την τοποθεσία που ετοιμάζεται να επισκεφτεί. Για την καλύτερη οργάνωση, οι διαχειριστές θα πρέπει λοιπόν να ελέγχουν τα σχόλια καθώς και τις νέες τοποθεσίες που προστίθονται αρκετά συχνά σε μια περιοχή (αν και οι τοποθεσίες παρέχονται από Google API) ώστε να ενημερώνεται συνέχεια η εφαρμογή με νέες και σωστές πληροφορίες.

3.1.2 Τεχνολογίες ιστού για ιδιοκτήτες σκύλων

Αρχικά, αναγκαίο για την παρούσα εφαρμογή είναι η δημιουργία και ο σχεδιασμός μιας βάσης δεδομένων στην οποία οι διαχειριστές, για την καλύτερη οργάνωση, θα αποθηκεύουν τους χρήστες και τις πληροφορίες/σχόλια μιας τοποθεσίας . Παρόμοιες βάσεις δεδομένων είχαν οι περισσότερες εφαρμογές και τα συστήματα που μελετήθηκαν προηγουμένως με κύριες διαφορές στον τρόπο πρόσβασης και αποθήκευσης σε αυτά. Στα περισσότερα συστήματα και εφαρμογές που μελετήθηκαν υπήρχαν διαχειριστές οι οποίοι πληροφορούσαν την κοινότητα για συγκεκριμένες ανάγκες και τοποθεσίες, όπως για παράδειγμα για θέματα υγείας ή για εύρεση ξενοδοχείων, ενώ στην περίπτωση μας ο χρήστης είχε να επιλέξει μεταξύ τεσσάρων βασικών αναγκών, λόγου χάρη υγεία, αξεσουάρ και ύλες πρώτες ανάγκης, τα κατάλληλα πάρκα για σκυλιά και καφετέριες που δέχονται τετράποδα.

Ωστόσο,τα να μπορούν οι ιδιοκτήτες σκυλιών να αποθηκεύουν προσωπικά δεδομένα και να αναζητούν βασικές πληροφορίες στην βάση αποτελεί την πιο σημαντική προδιαγραφή της εφαρμογής. Η τεχνολογία που θα χρησιμοποιηθεί για την υλοποίηση αυτής της προδιαγραφής είναι η τεχνολογία Ajax (Asynchronous JavaScript And Xml). Η συγκεκριμένη τεχνολογία βρίσκει χρήση τόσο σε αμέτρητες εφαρμογές κινητών συσκευών όσο και σε άλλες ιστοσελίδες και λογισμικά. Αυτά που επιτρέπει η τεχνολογία αυτή είναι:

- Η ανανέωση μιας ιστοσελίδας χωρίς την επαναφόρτιση της
- Η ικανότητα να ζητά και να αποδέχεται δεδομένα από τον server μετά την φόρτωση της σελίδας
- Η αποστολή δεδομένων στο server χωρίς να γίνεται αντιληπτό απ τον χρήστη

Αφού γίνει αντιληπτή η αναγκαιότητα για επικοινωνία της εφαρμογής με το server αξίζει να σημειωθεί η χρήση της τεχνολογίας Json. Πιο συγκεκριμένα, το Json (JavaScript Object Notation) αποτελεί μια από τις πιο γνωστές τεχνολογίες που αφορούν τον τρόπο σύνταξης, αποθήκευσης και ανταλλαγής πληροφοριών μεταξύ client και server με σκοπό την βελτίωση της επικοινωνίας μεταξύ τους. Ο χρήστης δεν επηρεάζεται άμεσα από την τεχνολογία αυτή καθώς δεν την γνωρίζει, επηρεάζεται ωστόσο έμμεσα ολόκληρη η εμπειρία του με την εφαρμογή καθώς υπάρχει σημαντική μείωση στον χρόνο που χρειάζεται να φορτωθούν τα δεδομένα του.

Για την καλύτερη ανάπτυξη και υλοποίηση της εφαρμογής, εκτός από τις τεχνολογίες που αναφέραμε προηγουμένως, είναι απαραίτητη ακόμη η πρόσβαση σε κάποια APIs. Τα API (Application Programming Interface) είναι ένα ενδιάμεσο λογισμικό που επιτρέπει την επικοινωνία μεταξύ δύο εφαρμογών. Με απλά λόγια, είναι ο φορέας που παραδίδει το αίτημα σας στον πάροχο που είστε και στη συνέχεια επιστέφει την απάντηση πίσω σε εσάς. Δηλαδή, είναι ένας βέλτιστος τρόπος, με χρήση συναρτήσεων και προγραμματιστικών διαδικασιών, να έχει κανείς πρόσβαση σε λειτουργίες ή δεδομένα μίας άλλης εφαρμογής, σε βιβλιοθήκες και λειτουργικά συστήματα να κάνει αίτηση και ανταλλαγή δεδομένων. Η ταχύτητα που το API επιτρέπει στους προγραμματιστές να χτίσουν τις εφαρμογές είναι σημαντική για τον σημερινό ρυθμό ανάπτυξης εφαρμογών. Αυτό συμβαίνει για να μπορούν οι διεπαφές να ορίζουν και να διατυπώνουν το σύνολο των λειτουργιών και υπηρεσιών που μπορεί να παρέχει μια βιβλιοθήκη ή ένα λειτουργικό σύστημα σε άλλα προγράμματα, χωρίς να επιτρέπει πρόσβαση στον κώδικα που υλοποιεί αυτές τις υπηρεσίες.

Στη συνέχεια, απαραίτητος για την ανάπτυξη της εφαρμογής είναι ο σχεδιασμός και η προσθήκη ορισμένων plug-ins. Ως plug-in ή add-on, επίσης plugin, ορίζεται ένα σύστημα συστατικών κάποιου λογισμικού που προσθέτει ιδιαίτερες δυνατότητες σε ένα μεγαλύτερο λογισμικό. Βασικός σκοπός μιας εφαρμογής που υποστηρίζει plug-ins είναι η προσαρμογή του χρήστη σε αυτό και η βελτίωση της εμπειρίας του.

Ένα από τα σημαντικότερα χαρακτηριστικά αυτής της εφαρμογής, ίσως και η πιο βασική λειτουργία, είναι η παρουσίαση των δεδομένων στο χάρτη. Αυτό, επιτυγχάνεται με την χρήση του ενός API, πιο συγκεκριμένα το Geo-location API της Google. Κατά την χρήση του συγκεκριμένου API επιτυγχάνεται η παρουσίαση και η επεξεργασία ενός google map σε μια σελίδα ή εφαρμογή όπως και την δικιά μας. Δηλαδή, στην παρούσα διπλωματική απεικονίζονται στον χάρτη σημεία και τοποθεσίες σε ακτίνα 1000 μέτρων από τον χρήστη, καθώς και οδηγίες μεταφοράς από την τοποθεσία του προς το σημείο που επιθυμεί.

3.1.3 Δυνατότητες Κινητών Συσκευών για ιδιοκτήτες σκυλιών

Από τη μελέτη και την ανάλυση που έγινε στα προηγούμενα κεφάλαια, γίνονται αντιληπτές και σαφείς οι δυνατότητες και οι λειτουργίες που παρέχουν οι κινητές συσκευές. Μπορούν να φανούν αρκετά χρήσιμες, μπορούν να διευκολύνουν και να βελτιώσουν σε αρκετά μεγάλο βαθμό την καθημερινή ζωή των ανθρώπων, στην περίπτωσή μας των ιδιοκτητών σκύλων.

Η εφαρμογή, αν σχεδιαστεί και υλοποιηθεί με κατάλληλο τρόπο, θα διευκολύνει στο μέγιστο δυνατό βαθμό την ζωή των κατόχων . Επομένως, για να μπορέσει ένας χρήστης να έχει την βέλτιστη εμπειρία κατά την αλληλεπίδραση με την εφαρμογή, κρίνεται απαραίτητη η ελαχιστοποίηση των ενεργειών του.

Οι πιο βασικές λειτουργίες που διαθέτει μια κινητή συσκευή για τον χρήστη, προκειμένου να βιώσει μια καλύτερη εμπειρία κατά την αλληλεπίδραση, είναι η χρήση GPS, η δυνατότητα εικόνας και σχολίων τα οποία συνδέονται με τις συντεταγμένες του χρήστη και των αντίστοιχων τοποθεσιών. Με αυτόν τον τρόπο, ο χρήστης θα έχει την δυνατότητα, σε αντίθεση με τα λογισμικά σταθερών υπολογιστών, να στείλει δεδομένα της τοποθεσίας του για να λειτουργήσει το GPS από την κινητή συσκευή του και αυτό λοιπόν θα οδηγήσει στην εύρεση τοποθεσιών στην γύρω περιοχή.

3.1.4 Ανακεφαλαίωση προδιαγραφών

Σε αυτό το υποκεφάλαιο θα ακολουθήσει η συνάθροιση προδιαγραφών που χρειάζεται να καλύψει μια εφαρμογή κινητών συσκευών με σκοπό την υποστήριξη και την κάλυψη των αναγκών των ιδιοκτητών στο πλαίσιο της καλύτερης ανάθρεψης των σκυλιών τους. Η μελέτη παρόμοιων εφαρμογών και η σύγκρισή των λειτουργιών τους που παρέχουν οι εφαρμογές και οι ιστοσελίδες αυτές οδήγησαν στην εύρεση των προδιαγραφών της εφαρμογής μας. Η εφαρμογή πρέπει να καλύπτει τις εξής ενέργειες:

1. Βάσεις Δεδομένων: Μια βάση δεδομένων με κύριο ρόλο την ασφαλή καταγραφή και διανομή περιεχομένου μεταξύ των ανθρώπων και κατοικίδιων. Οι ιδιοκτήτες θα μπορούν να πραγματοποιήσουν ασφαλή εγγραφή στην βάση, να αναζητήσουν πληροφορίες για τυχόν τοποθεσίες καθώς και να σχολιάζουν οι ίδιοι.Ταυτόχρονα να έχουν πρόσβαση σε πληροφορίες και κριτικές που έχουν αναρτήσει άλλοι ιδιοκτήτες για τοποθεσίες που πρόκειται να επισκεφθούν καθώς και την επιλογή να αλλάξουν τα προσωπικά τους στοιχεία,παραδείγματος χάριν κωδικούς και username.

2. Σύστημα διαχείρισης περιεχομένου: Αρκετά σημαντικό είναι ο κώδικας να γραφεί με τέτοιο τρόπο, ώστε η ύπαρξη ενός διαχειριστή εντός του συστήματος να μην χρειάζεται, και οι πληροφορίες κάθε χρήστη να αποθηκεύονται αυτόματα.

3. Διαχείριση Σχολίων: Στον χρήστη- ιδιοκτήτη σκυλιού πρέπει να δίνεται η δυνατότητα από το σύστημα να παρακολουθεί τα σχόλια και τις κριτικές των υπόλοιπων χρηστών για τοποθεσίες που πρόκειται να επισκεφτεί, να μπορεί σχολιάσει και να τις αποθηκεύει στη βάση και ταυτόχρονα να διαγράφει σχόλια του κατόπιν επιθυμίας.

4. Διαχείριση Τοποθεσιών: Στον χρήστη- ιδιοκτήτη σκυλιού πρέπει να δίνεται η δυνατότητα να έχει πρόσβαση σε όλα τα πάρκα για σκυλιά, Pet Shops, κτηνιατρεία και καφετέριες φιλικά προς τον σκύλο σε απόσταση χιλίων μέτρων από το σημείο που βρίσκεται, να μπορεί να τα βαθμολογή- σει, να κάνει σχόλια, και φυσικά να μεταφέρεται τελικά σε αυτά.

5. Πλοήγηση: Στον χρήστη- ιδιοκτήτη σκυλιού πρέπει να δίνεται η δυνατότητα να πλοηγηθεί σε όλες τις τοποθεσίες που αναφέρθηκαν στην διαχείριση τοποθεσιών. Πιο συγκεκριμένα, το σύστημα θα πρέπει να του παρέχει τη δυνατότητα άμεσης πλοήγησης προς τα πάρκα για σκυλιά σε περίπτωσης βόλτας, τα Pet shop της περιοχής για αγορά διαφόρων προϊόντων (π.χ. τροφή, αξεσουάρ, σνακ κ.λπ.), τα κτηνιατρεία φυσικά -σε περίπτωση ανάγκης ή για ένα απλό check up- καθώς και τις καφετέριες που δέχονται σκύλους σε περίπτωση που θέλει να απολαύσει έναν καφέ ο χρήστης με τον σκύλο του. Επίσης, εκτός από τις τοποθεσίες τους, είναι απαραίτητο να τους παρέχεται και η δυνατότητα πλοήγησης στις τοποθεσίες είτε με αμάξι είτε με τα πόδια.

3.2 Ανάλυση χρηστών και περιβάλλοντος εργασίας

Ένα από τα πρώτα και βασικότερα βήματα στην δημιουργία και σχεδίαση μιας εφαρμογής, λογισμικού υπολογιστή ή ιστοσελίδας είναι η ανάλυση χρηστών. Δηλαδή, με την κατάλληλη μελέτη γίνεται πιο εύστοχη και βέλτιστη η εφαρμογή καθώς ικανοποιούνται και καλύπτονται περισσότερο οι ανάγκες των χρηστών στους οποίους προωθείται η εφαρμογή. Οπότε, για μεγαλύτερη ευχρηστία της εφαρμογής είναι σημαντικό να απαντήσουμε στα παρακάτω ερωτήματα για τους χρήστες της όπως:

- Ποιοι είναι οι χρήστες, τι γνωρίζουν, τι μπορούν να μάθουν;
- Τι θέλουν οι χρήστες ή τι έχουν ανάγκη να κάνουν;
- Ποιο είναι το γενικό υπόβαθρο των χρηστών;
- Ποιο είναι το πλαίσιο στο οποίο θα εργαστούν;
- Τι κάνει η εφαρμογή και τι ο χρήστης; (Patanothai, 2016)

Το σύστημα για το οποίο γίνεται η μελέτη και η ανάλυση, αφορά το σύνολο των ατόμων που απαρτίζουν μια κοινότητα, στην περίπτωσή μας ατόμων που είναι ιδιοκτήτες σκυλιών. Για την καλύτερη οργάνωση, οι χρήστες μπορούν να ομαδοποιηθούν σε δύο ομάδες:

- Στους πρωτεύοντες χρήστες, δηλαδή εκείνους που αλληλεπιδρούν άμεσα με το σύστημα και πρόκειται να επηρεαστούν από αυτό καθώς και να επηρεάσουν. Σε αυτή την περίπτωση θα ορίζονται ως τέτοιοι οι ιδιοκτήτες σκύλων.
- Στους δευτερεύοντες χρήστες, με άλλα λόγια εκείνους που αλληλεπιδρούν σε μικρότερο βαθμό με το σύστημα και ταυτόχρονα επηρεάζουν την εμπειρία των πρωτεύων χρηστών. Αυτοί θα ορίζονται οι διαχειριστές της Google και των καφετεριών, των Pet Shop, των κτηνιατρείων και των πάρκων για σκυλιά που είναι υπεύθυνοι για τις πληροφορίες των τοποθεσιών και τις ορθές κριτικές από και προς τους χρήστες.
- Στους τριτεύοντες χρήστες, δηλαδή εκείνους που δεν αλληλοεπιδρούν καθόλου με το σύστημα αλλά επηρεάζονται από την εισαγωγή του, οι οποίοι θα ορίζονται ως την κοινότητα ατόμων που δεν απαιτείται η συνεχή επαφή τους με την εφαρμογή και δεν θα επωφεληθούν από το συγκεκριμένο περιεχόμενα μέσω διαδικτύου.

Μετά τον προσδιορισμό των ομάδων των χρηστών κρίνεται αναγκαία η καταγραφή των χαρακτηριστικών τους, τα οποία αποτελούνται από (Αβούρης 2000):

1. Ατομικά χαρακτηριστικά: (ηλικία, φυσικές ικανότητες/ιδιαιτερότητες, μαθησιακή ικανότητα, γνωσιακή ικανότητα, πολιτισμικό υπόβαθρο, φοβίες, προσωπικότητα, εμπειρία-δεξιότητες, κίνητρα και φιλοδοξίες κ.λπ.) στοχεύοντας τους πρωτεύοντες χρήστες, η εφαρμογή αφορά ανθρώπους που ιδιοκτήτες κατοικίδιων σκυλιών, άρα σε άτομα ηλικίας 14-60, τα οποία έχουν όρεξη και υπευθυνότητα να προσέξουν και να αναθρέψουν σωστά τον σκύλο τους και είναι πρόθυμοι να χρησιμοποιήσουν ένα εργαλείο το οποίο θα διευκολύνει την ανατροφή του σκυλιού τους και θα τους ενημερώνει συνέχεια.
2. Χαρακτηριστικά σε σχέση με υπολογιστές: (προηγούμενη εμπειρία σε λογισμικό και λειτουργικά συστήματα, εμπειρία σε χρήστες συσκευών, προδιάθεση έναντι υπολογιστών και πληροφορικής κ.λπ.) οι πρωτεύοντες χρήστες κατά κύριο λόγο αποτελούνται από ανθρώπους με βασική γνώση και εμπειρία στη χρήση κινητών συσκευών, λογισμικών και εφαρμογών. Παρ΄όλα αυτά, είναι απαραίτητο η σχεδίαση της εφαρμογής να είναι όσο το δυνατόν πιο απλή και χρήσιμη για έναν χρήστη.
3. Ομαδικά χαρακτηριστικά: (στόχοι και αποστολή ομάδας, συνοχή και ομοιογένεια μελών της αντίστοιχης ομάδας, αυτονομία, εξάρτηση από άλλες ομάδες, δομή και δυναμικά χαρακτηριστικά, είδος συμμετοχής κ.λπ.). Στη συγκεκριμένη εφαρμογή, αναφερόμαστε σε ανεξάρτητα μέλη. Όμως, μέσα από τον σχολιασμό και τις κριτικές αξιολόγησης τοποθεσιών είναι λογικό να ασκούν επιρροή στις επιλογές και κατά συνέπεια στην εμπειρία άλλων χρηστών.

Τέλος, δεδομένου ότι κατά την χρήση της εφαρμογής ο χρήστης βρίσκεται σε εξωτερικούς χώρους, πιο σπάνια σε εσωτερικούς (σπίτι ή επιθυμητή τοποθεσία), πολλές φορές δημιουργούνται προβλήματα και δυσκολίες που δεν έχουν υπολογισθεί. Ένα παράδειγμα αποτελούν οι καιρικές συνθήκες και λύση σε αυτό αποτελεί η σωστή διοργάνωση και ο έλεγχος.

3.3 Λειτουργικές απαιτήσεις συστήματος

Οι χρήστες-ιδιοκτήτες σκυλιών έχουν την δυνατότητα να συνδεθούν και να έχουν πρόσβαση στην βάση δεδομένων τους μέσα από την εφαρμογή. Αρχικά, ο χρήστης-ιδιοκτήτης θα μπορεί να κάνει εγγραφή στην βάση δεδομένων με τα απαραίτητα στοιχεία του (πχ email,username,password κ.λπ.) και στην συνέχεια θα πραγματοποιεί σύνδεση στην εφαρμογή με τα προσωπικά του δεδομένα τα οποία είχε ήδη δώσει προηγουμένως. Στη συνέχεια, αφού γίνει επιτυχής σύνδεση, μεταφέρεται αυτόματα στην home page οπού θα του δίνεται η δυνατότητα, ανάλογα με την ανάγκη που έχει ο σκύλος του εκείνη την ώρα, να προχωρήσει στην εύρεση των τοποθεσιών που επιθυμεί να επισκεφτεί (πχ βόλτα σε πάρκα για σκυλιά, καφετέριες φιλικές προς τον σκύλο,Pet shops και κτηνιατρεία). Επίσης, οι επιπλέον σελίδες που θα μπορεί να επισκεφτεί είναι η Settings Page, όπου θα μπορεί να αλλάξει και να ρυθμίσει τα προσωπικά δεδομένα που έδωσε κατά την εγγραφή του στην εφαρμογή καθώς και η About Page, όπου θα υπάρχει μια σύντομη αναφορά για το τι εστί η εφαρμογή. Αφού ο χρήστης επιλέξει την εύρεση τοποθεσιών που επιθυμεί στην Home Page, θα μεταφερθεί σε έναν χάρτη όπου θα εμφανίζεται η τοποθεσία του καθώς και όλες οι τοποθεσίες(εάν υπάρχουν βέβαια) της αντίστοιχης αναζήτησης (π.χ. Pet Shops) και θα έχει τις εξής ενέργειες:

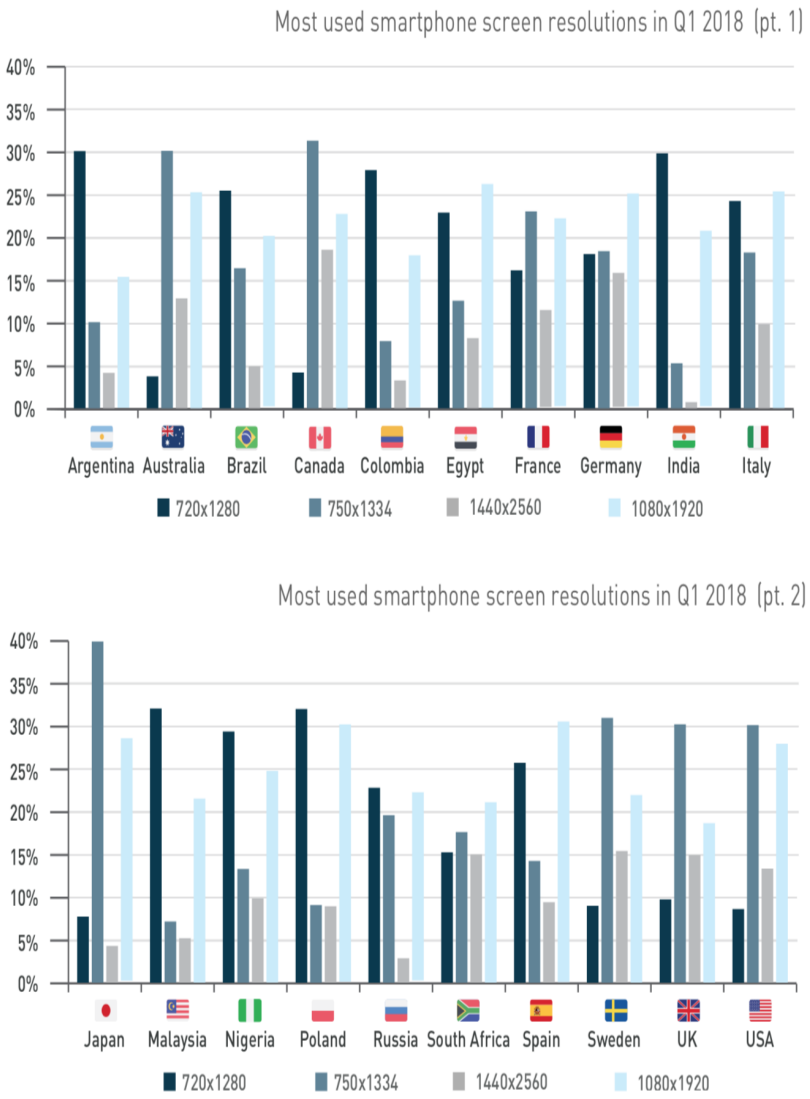
1. Να επιλέξει την επιλογή view οπού θα το εμφανίζεται ένα ”carousel” με τις εικόνες κάθε τοποθεσίας (άμα υπάρχουν στην Google Maps),τη διεύθυνση στην οποία βρίσκονται, την απόσταση που απέχουν από το σημείο του χρήστη, την κριτική βαθμολογία που διατίθεται από το Google Maps, καθώς και ένα κουμπί info, όπου θα μπορεί ο χρήστης να παρακολουθήσει τα σχόλια άλλων χρηστών για την αντίστοιχη τοποθεσία και να μπορεί να σχολιάσει και ο ίδιος.

2. Να επιλέξει την επιλογή ”Directions”, όπου και θα εμφανίζεται ο τρόπος με τον οποίο θα μπορεί να προηγηθεί στην αντίστοιχη τοποθεσία από το σημείο που βρίσκεται, είτε με τα πόδια, είτε με αυτοκίνητο.

3.4 Μη-λειτουργικές απαιτήσεις συστήματος

Το φάσμα κινητών συσκευών για το οποίο καλείται να λειτουργήσει η εφαρμογή της παρούσας διπλωματικής εργασίας είναι ευρύ, ώστε να καλύψει όλες τις ανάγκες όλων των ανθρώπων-ιδιοκτητών σκυλιών. Επομένως, είναι αναγκαίο να είναι κατάλληλα σχεδιασμένη και εφοδιασμένη για ποικιλία μεγεθών και λειτουργικών συστημάτων, πράγμα που έχει ληφθεί υπόψιν όπως θα αναφερθεί στο κεφάλαιο τέσσερα. Ωστόσο, στην παρούσα εφαρμογή δεν ήταν δυνατόν να οριστεί σε κάθε πιθανό μέγεθος και ανάλυση οθόνης με αποτέλεσμα να προσαρμοστεί και να δοκιμαστεί σε ένα android κινητό (Samsung galaxy S7) με ανάλυση 720p την οποία έχουν και χρησιμοποιούν πάρα πολλές συσκευές από το 2017 και ύστερα (εικόνα 3.1) σύμφωνα με έρευνα του DeviceAtlas [12].

Ο χρήστης-ιδιοκτήτης σκύλου, με βάση τις προδιαγραφές της εφαρμογής που προτείνεται διά της παρούσας διπλωματικής εργασίας, για να μπορεί να χρησιμοποιήσει και να εισέλθει στην εφαρμογή, θα χρειαστεί να είναι συνδεμένος με το διαδίκτυο είτε μέσω των δεδομένων κινητής τηλεφωνίας είτε μέσω Wi-Fi. Ακόμη, αναγκαία προϋπόθεση για τη λειτουργία της εφαρμογής είναι οι πληροφορίες της τοποθεσίας του χρήστη έτσι ώστε να χρησιμοποιηθούν για την εύρεση τοποθεσιών κοντά του.



Σχήμα 3.1: Screen Resolutions

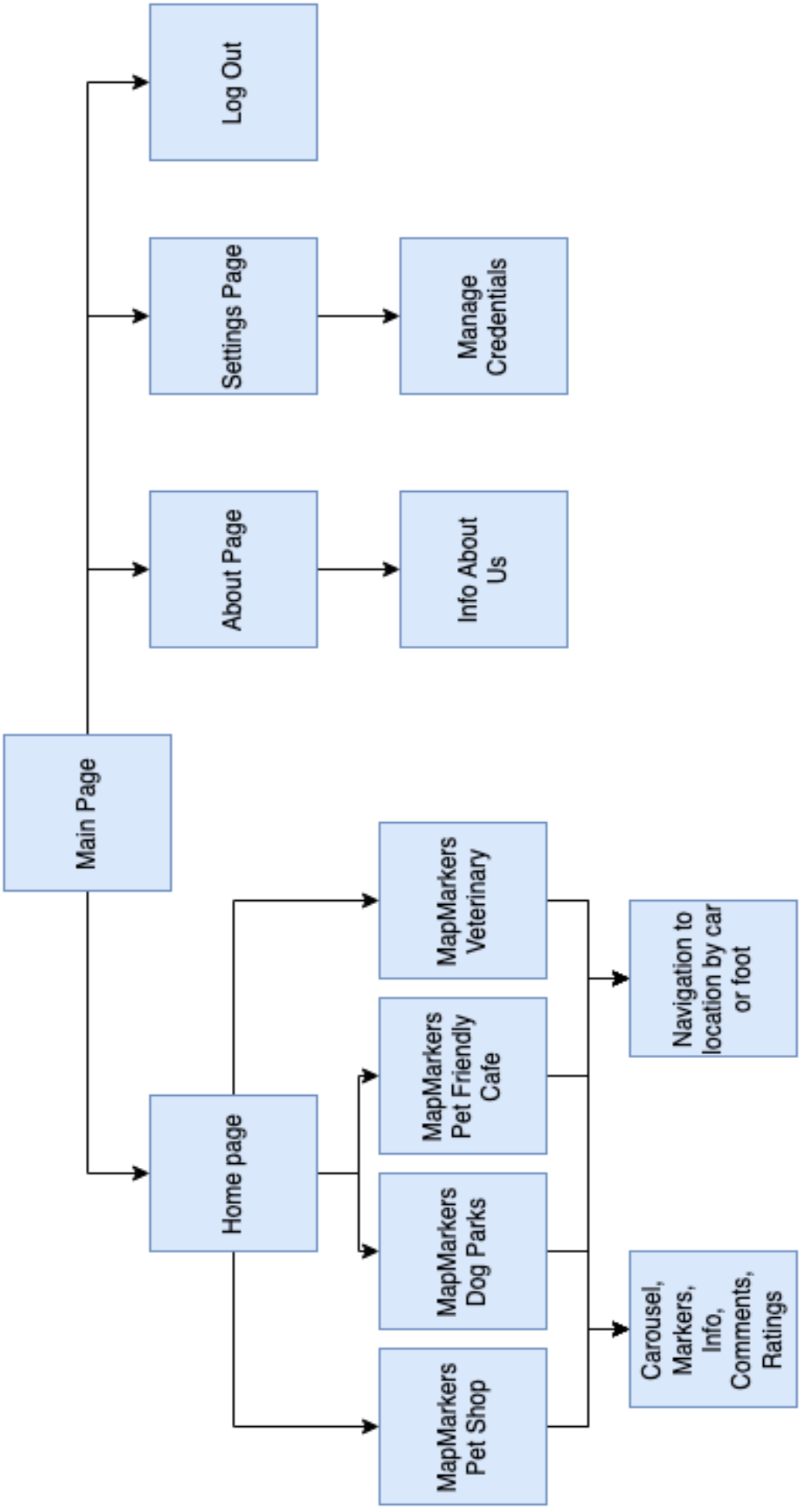
4. ΣΧΕΔΙΑΣΗ

Στο τέταρτο κεφάλαιο της διπλωματικής εργασίας, θα γίνει η ανάλυση και η περιγραφή των μεθόδων που ακολουθήθηκαν αλλά και των ενεργειών που πραγματοποιήθηκαν για την υλοποίηση της εφαρμογής. Σε αρχικό στάδιο παρουσιάζεται ο χάρτης πλοήγησης της εφαρμογής και στην συνέχεια το πρότυπο που σχεδιάστηκε για το τελικό στάδιο της παρούσας εφαρμογής, επί παραδείγματι τα μοντέλα οθονών που σχεδιάστηκαν καθώς και η ανάλυση κάποιων θεωρητικών σεναρίων-παραδειγμάτων.

4.1 Χάρτης πλοήγησης

Αρχικά, λοιπόν θα γίνει μια επιγραμματική επεξήγηση καθώς και παρουσίαση του χάρτη πλοήγησης. Με τον όρο ”χάρτης πλοήγησης” αναφερόμαστε σε ένα διάγραμμα ροής που απεικονίζει με αφαιρετική μορφή την δομή της εφαρμογής. Ο χάρτης πλοήγησης, θα βοηθήσει μελλοντικά να βελτιώσουμε τυχόν προβλήματα της εφαρμογής με σκοπό οι χρήστες να βιώνουν την μέγιστη δυνατή εμπειρία. Στην περίπτωσή μας, η ανάλυση και η σχεδίαση της εφαρμογής επικεντρώθηκε στο να ικανοποιήσει τις ανάγκες των χρηστών όσο το δυνατόν περισσότερο.

Σχήμα 4.1: Χάρτης Πλοήγησης



4.2 Πρωτότυπο

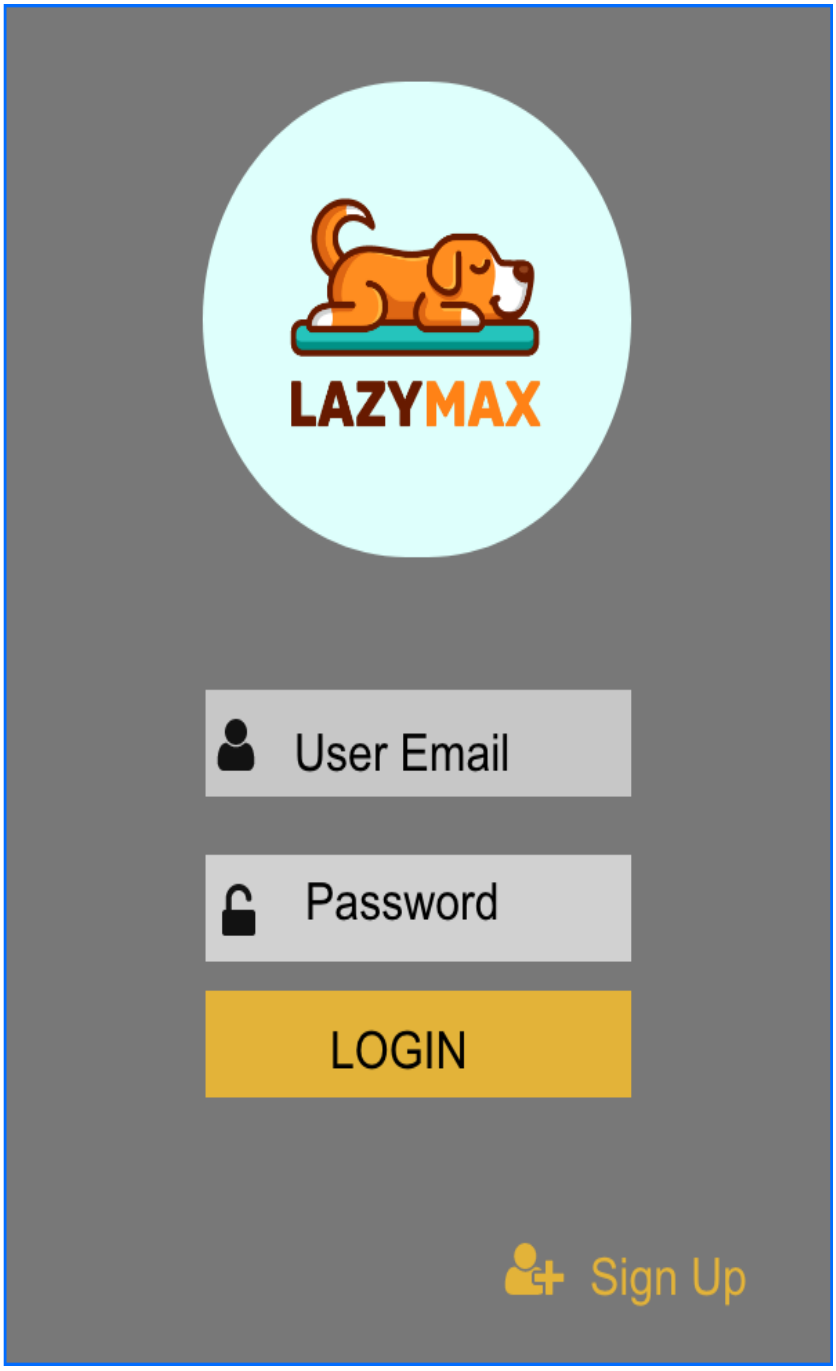
Σε αυτό το υποκεφάλαιο, παρουσιάζεται η υλοποίηση του σχεδιασμού προτύπων της εφαρμογής η οποία αποτελεί και ένα από τα σημαντικότερα στάδια ανάπτυξής της. Για να δημιουργηθεί ο αρχικός ”σκελετός” πάνω στον οποίο θα στηριχτεί η εφαρμογή, απαραίτητη κρίνεται η ύπαρξη ενός μοντέλου ή προτύπου για την αναπαράσταση. Επομένως με βάση το συγκεκριμένο μοντέλο θα γίνουν οι έλεγχοι και η υλοποίηση όλων των προδιαγραφών, ενώ σταδιακά θα προστίθενται περιεχόμενο.

Συνεχίζοντας, χρησιμοποιήθηκε το online πρόγραμμα UXPin (https://www.uxpin.com) για τον σχεδιασμό και την υλοποίηση του προτύπου. Το αναφερόμενο online εργαλείο UXPin πρόκειται για μια πλατφόρμα σχεδιασμού που στοχεύει στην καλύτερη κατανόηση του πρότυπου. Το UXpin επιτρέπει την καταγραφή της συμπεριφοράς των χρηστών, ενώ συγχρονίζεται με αρχεία όπως photoshop και sketch. Παράλληλα δίνεται η δυνατότητα αλλαγής σε διαδραστικά πρωτότυπα κ.λπ. Η απόφαση για τον σχεδιασμό του συγκεκριμένου προτύπου πάρθηκε έπειτα από συζητήσεις με συμφοιτητές στο Πανεπιστήμιο Πατρών. Κατά την υλοποίησή του αναφέρονται όλες οι οθόνες-παράθυ- ρα που εμφανίζονται καθώς και όλες οι λειτουργίες της εφαρμογής. Πιο συγκεκριμένα τα παράθυρα που χρησιμοποιήθηκαν στο πρότυπο είναι τα εξής:

1. Login Χρήστη
2. Sign Up Χρήστη
3. Home Page
4. Home Page με επιλογή για άλλες σελίδες
5. Settings Page
6. About Page
7. Προβολή Χάρτη με τοποθεσίες για Πάρκα, Καφετέριες, Κτηνιατρεία και Pet Shops
8. Προβολή Carousel πληροφοριών για τις τοποθεσίες
9. Προβολή σχόλιον για τις τοποθεσίες
10. Προβολή Μεταφοράς στην τοποθεσία

4.2.1 Παράθυρο Login Χρηστών

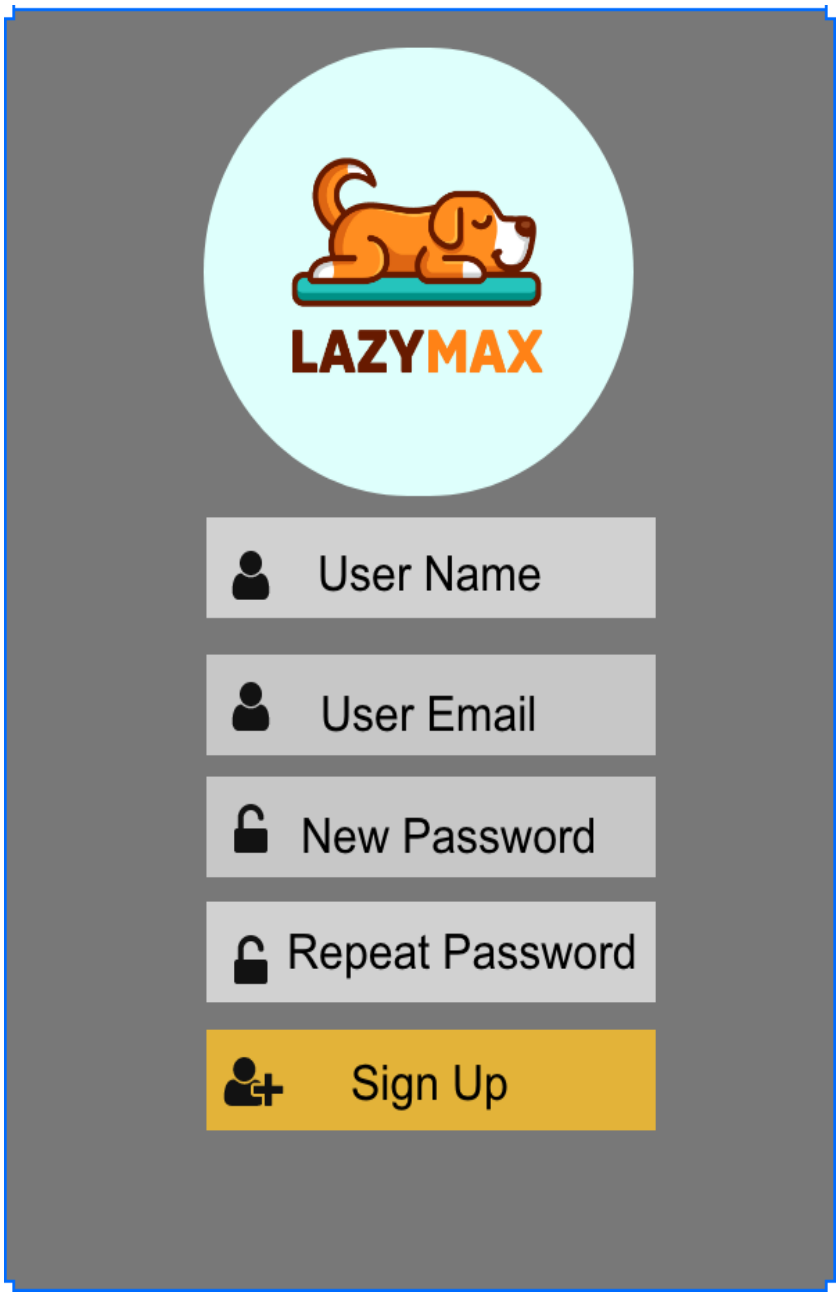
Ο χρήστης-ιδιοκτήτης σκυλιού κατά την είσοδο του στην εφαρμογή, η πρώτη σελίδα που αντικρίζει είναι η οθόνη εισαγωγής στην εφαρμογή(Login Page). Με τον όρο ”Οθόνη Εισαγωγής” αναφερόμαστε σε μια απλή φόρμα στην οποία ο χρήστης πληκτρολογεί τα προσωπικά του δεδομένα για να συνδεθεί (εάν έχει ήδη λογαριασμό), διαφορετικά επιλέγοντας το Sign Up κουμπί πραγματοποιεί την εγγραφή του στην εφαρμογή.



Σχήμα 4.2: Παράθυρο σύνδεσης πρωτοτύπου

4.2.2 Παράθυρο Εγγραφής Χρήστη

Στη συνέχεια, στην περίπτωση που ο χρήστης δεν είναι εγγεγραμμένος και αφού επιλέξει την επιλογή Sign Up), μεταφέρεται στην οθόνη εγγραφής . Εδώ, ο χρήστης καλείται να συμπληρώσει τα προσωπικά του στοιχεία, όπως λόγου χάριν το email,username και κωδικό πρόσβασης. Προχωρώντας, για να μεταφερθεί σε επόμενη σελίδα καλείται να επιλέξει το κίτρινο κουμπί ”Sign Up” ώστε να πραγματοποιηθεί αποθήκευση των στοιχείων του στην βάση δεδομένων. Μετά τις ενέργειες που αναφέρθηκαν, ο χρήστης μεταφέρεται στην επομένη σελίδα, δηλαδή την Home Page.

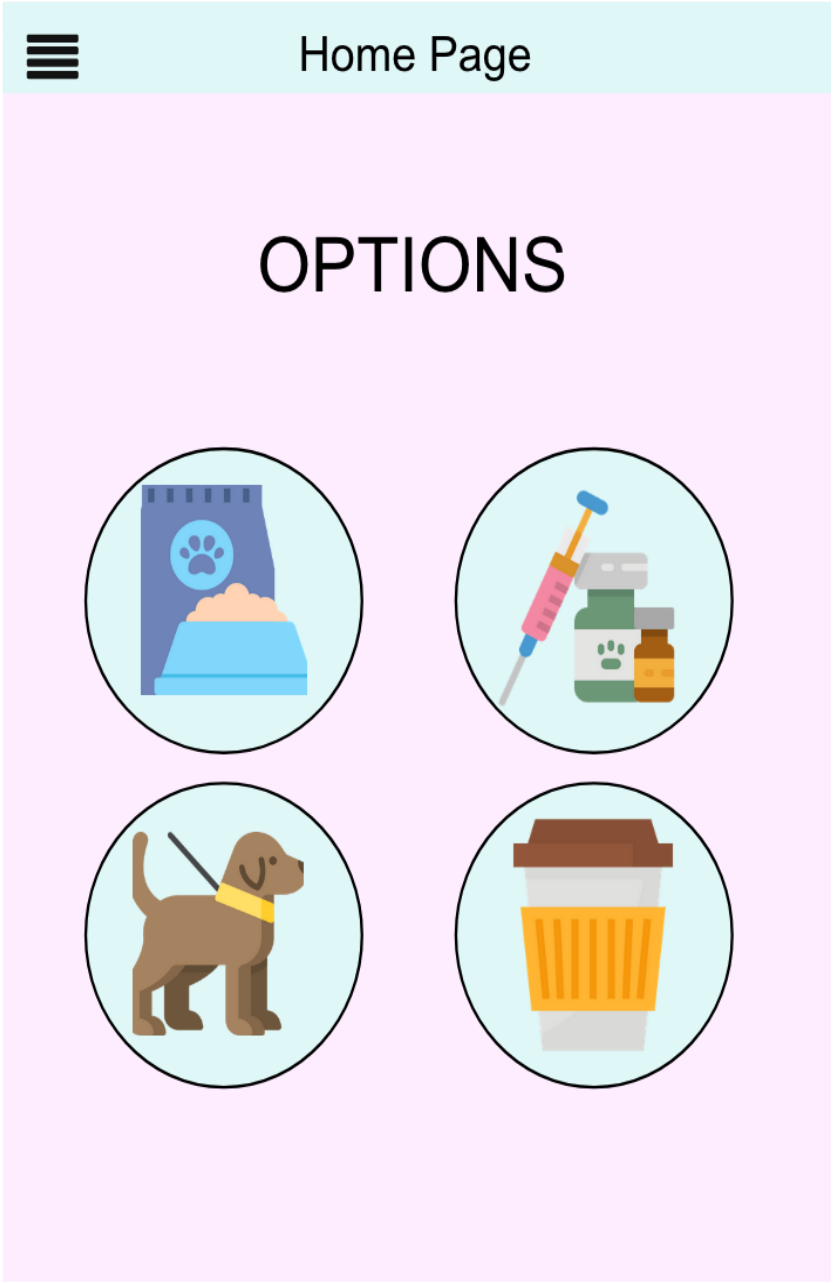


Σχήμα 4.3: Παράθυρο εγγραφής πρωτοτύπου

Στο παράθυρο, γίνεται αντιληπτό ότι δεν υπάρχει τρόπος επιστροφής στην Login In Page σε περίπτωση που ο χρήστης είναι ήδη εγγεγραμμένος. Αυτό το σφάλμα, διορθώνεται, όπως θα καταγραφεί στο κεφάλαιο πέντε της ανά χείρας εργασίας.

4.2.3 Κεντρική σελίδα, Home Page

Η κεντρική οθόνη, ίσως και το πιο σημαντικό παράθυρο, είναι η πρώτη οθόνη που αντικρίζει ο χρήστης αφού συνδεθεί στην εφαρμογή. Πρόκειται για την οθόνη με τις βασικότερες λειτουργίες και ανάγκες σύμφωνα με την οποία φτιάχτηκε αυτή η εφαρμογή. Πιο συγκεκριμένα, στο πρότυπο Home Page παρατηρούνται τέσσερα βασικά κουμπιά ”Επιλογές” και ένα κουμπί ”Άνοιγμα Λίστας” στην επάνω αριστερή γωνία. Τα τέσσερα βασικά κουμπιά αντιστοιχούν στις τέσσερις από τις βασικές ανάγκες ενός σκύλου, δηλαδή, βόλτα σε πάρκα για σκυλιά,pet shops,κτηνιατρεία και καφετέριες φιλικές προς τον σκύλο. Ο χρήστης αφού επιλέξει ένα από τα τέσσερα κουμπιά θα μεταφερθεί σε έναν χάρτη (εικόνα 4.8) όπου θα απεικονίζονται οι τοποθεσίες για την αντίστοιχή αναζήτηση.

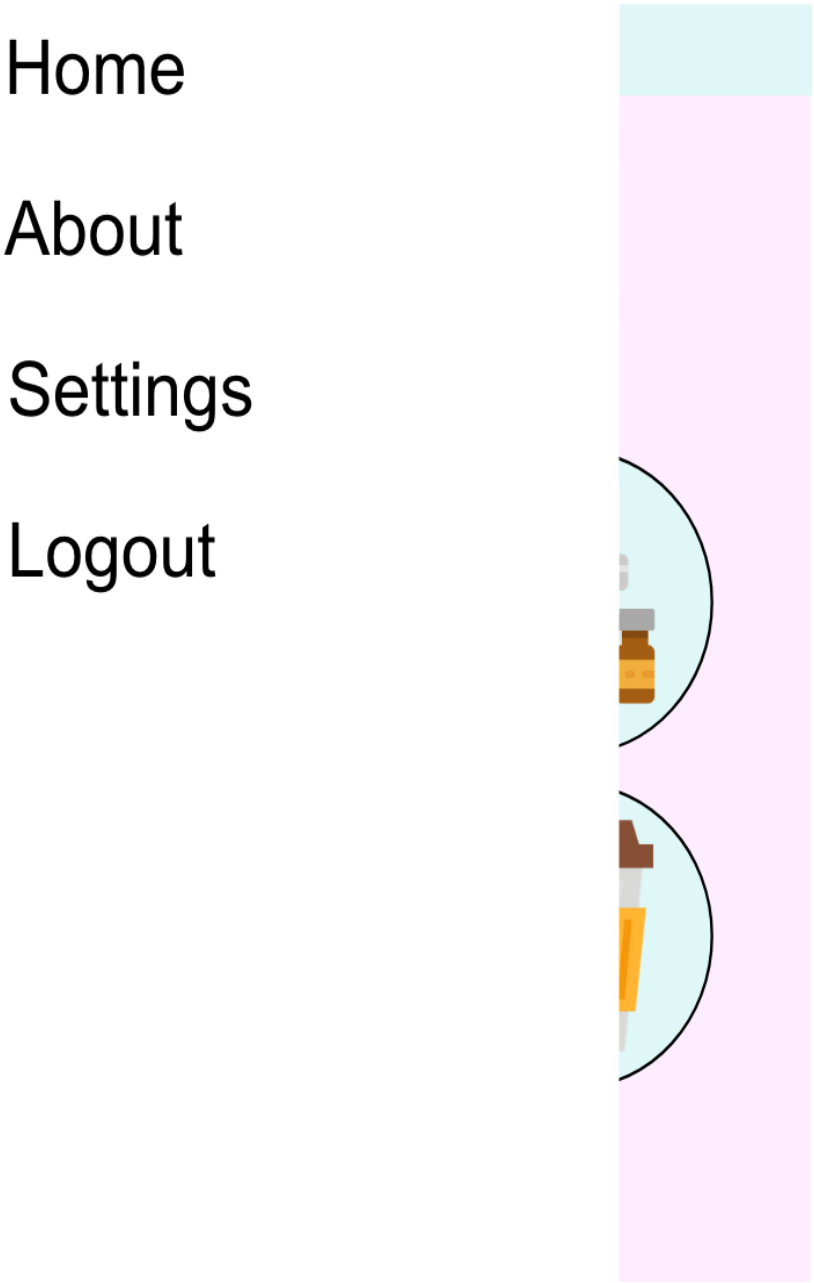


Σχήμα 4.4: Παράθυρο HomePage πρωτοτύπου

Αξίζει να σημειωθεί ότι η αντίστοιχη οθόνη στην τελική εφαρμογή,όπου θα συναντήσουμε στο κεφάλαιο πέντε, έχει διαφορετική αναπαράσταση των κουμπιών.

4.2.4 HomePage με επιλογή εμφάνισης άλλων σελίδων

Στην δεδομένη οθόνη γίνεται αναπαράσταση της επιλογής του πέμπτου κουμπιού που βρίσκεται στην Home Page, στην πάνω αριστερή γωνία. Αυτό εμφανίζει μια λίστα επιλογών με τις υπόλοιπες οθόνες .



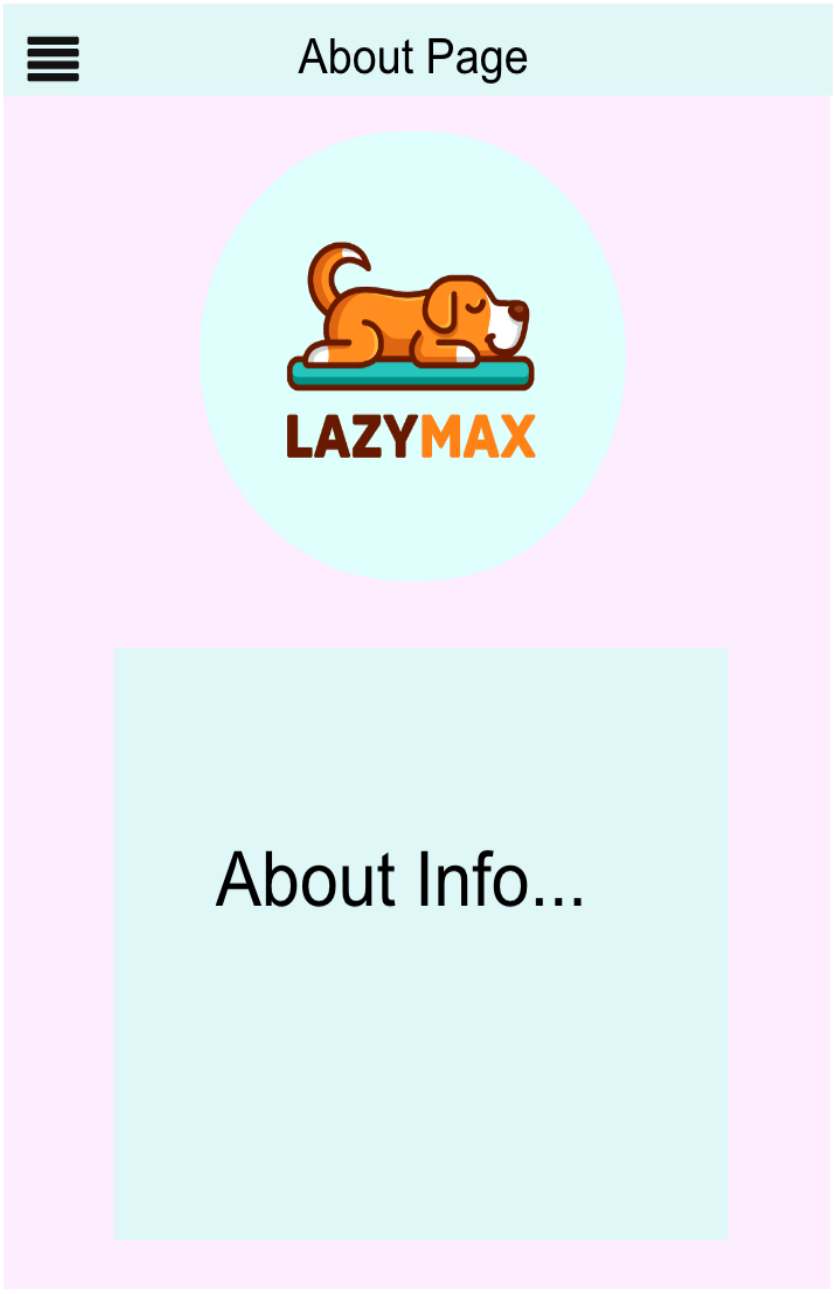
Σχήμα 4.5: Παράθυρο HomePage με επιλογή εμφάνισης άλλων σελίδων πρωτοτύπου

Παρατηρώντας, την επάνω οθόνη συμπεραίνει κανείς ότι θα ήταν πιο σωστό να φαίνεται η επιλογή της σελίδας στην οποία βρίσκεται πιο έντονη από τις υπόλοιπες επιλογές ώστε να γνωρίζει ο χρήστης σε πια σελίδα βρίσκεται κάθε χρονική στιγμή. Αυτό διορθώνεται στην τελική εφαρμογή.

4.2.5 Οθόνη AboutPage

Η οθόνη AboutPage, πρόκειται για μια από τις τρεις βασικές οθόνες τη εφαρμογής που μπορεί να επιλέξει ο χρήστης, αφού συνδεθεί στην εφαρμογή. Η συγκεκριμένη οθόνη έχει ως σκοπό να ”καλωσορίσει” τον χρήστη στην εφαρμογή και να τον ενημερώσει για τις λειτουργίες και τις δυνατότητες που του παρέχονται. Μια τέτοιου είδους εφαρμογή είναι απαραίτητο να διεκπεραιώνει τη βασική και

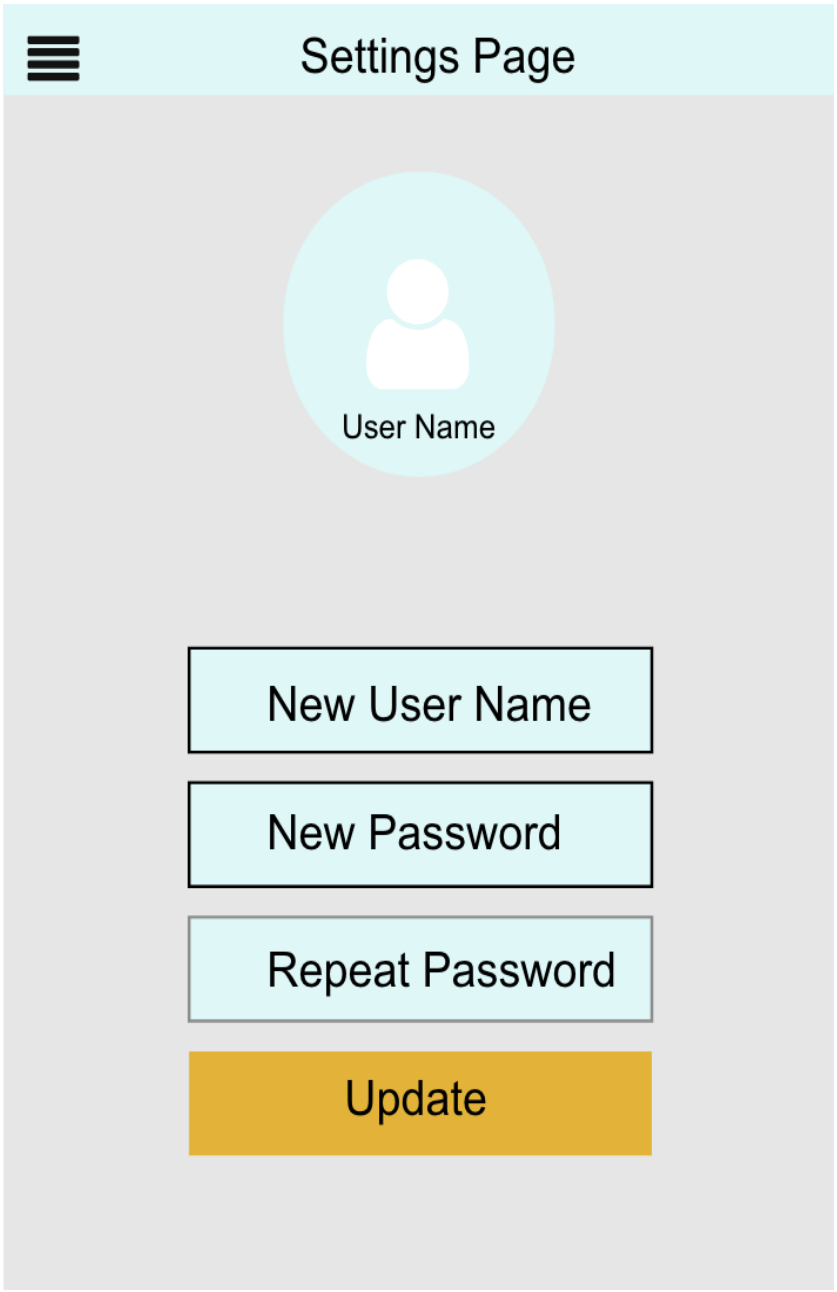
πρωταρχική ιδέα της αλλά και τον κύριο σκοπό της. Με αυτόν τον τρόπο θα μπορεί να βελτιώσει την εμπειρία του χρήστη ενώ την ίδια στιγμή ο χρόνος προσαρμογής του σε αυτήν θα μειώνεται σημαντικά.



Σχήμα 4.6: Παράθυρο AboutPage πρωτοτύπου

4.2.6 Οθόνη SettingsPage

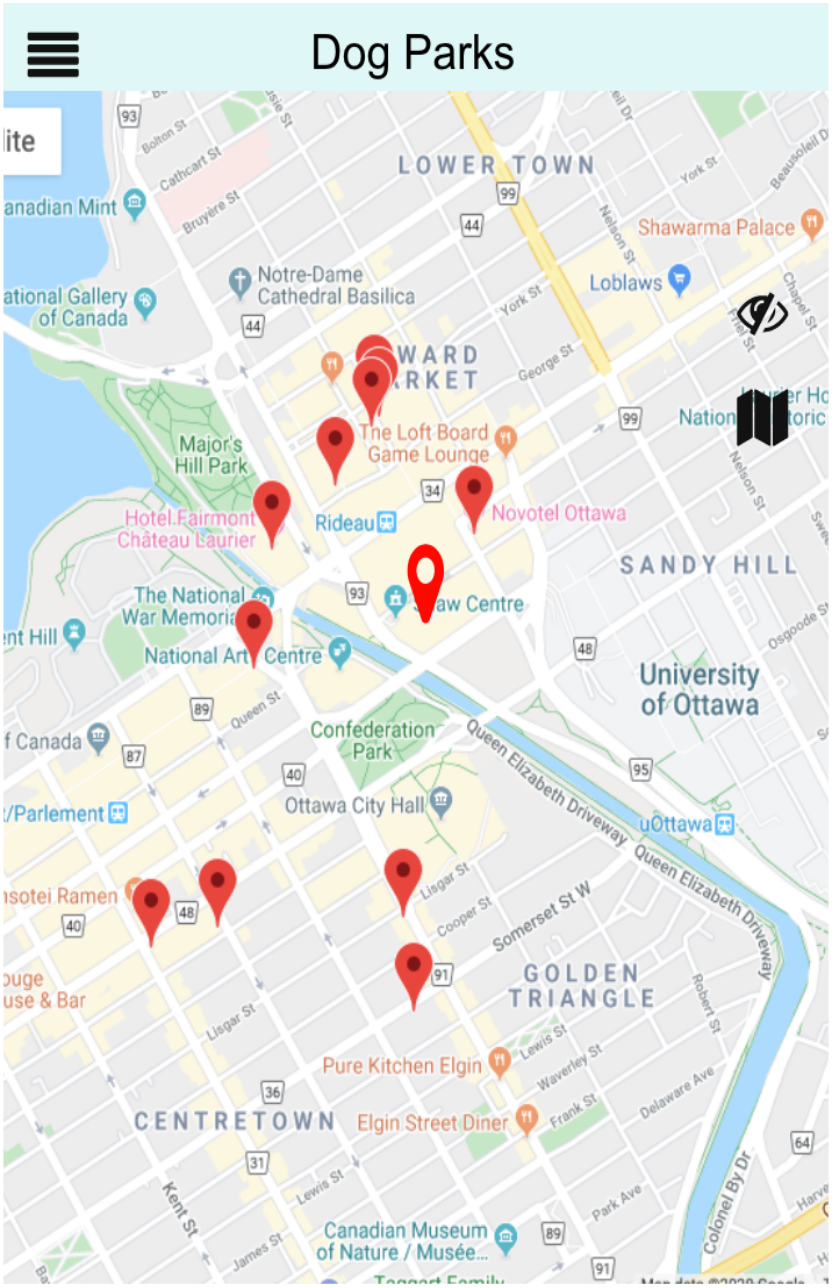
Η οθόνη SettingsPage, ουσιαστικά είναι μια από τις τρεις βασικές οθόνες τη εφαρμογής που μπορεί να επιλέξει ο χρήστης κατά την είσοδο του στην εφαρμογή. Στο παράθυρο αυτό, δίνεται η δυνατότητα στον χρήστη,εφόσον το επιθυμεί, ή ακόμα και για λόγους ασφάλειας, να αλλάξει τα προσωπικά του δεδομένα για παράδειγμα το όνομα χρήστη και τον κωδικό του. Αφού ο χρήστης συμπληρώσει τα κουτιά εισαγωγής στοιχείων και επιλέξει update, η εφαρμογή δίνει αυτόματα εντολή για τη μεταφορά των παραπάνω δεδομένων και έπειτα για την αποθήκευσή τους στη βάση δεδομένων. Τέλος, πραγματοποιείται η ανανέωση στοιχείων για τον συγκεκριμένο χρήστη.



Σχήμα 4.7: Παράθυρο SettingPage πρωτοτύπου

4.2.7 Προβολή Χάρτη με τοποθεσίες για Πάρκα,Καφετέριες,Κτηνιατρεία και Pet Shops

Στη οθόνη αυτού του προτύπου, παρακολουθείται η λειτουργία μιας από τις τέσσερις βασικές επιλογές που υπάρχουν στο παράθυρο HomePage. Στον χάρτη, λήφθηκε μια από τις τέσσερις αναζητήσεις, εκείνη για τα Pet Shop.Το πρότυπο θα είναι ίδιο σε κάθε επιλογή με τη μονή διαφορά ότι οι τοποθεσίες και οι πληροφορίες τους θα είναι διαφορετικές. Παρατηρούνται λοιπόν οι τοποθεσίες όλων των Pet Shop που εμφανίζονται γύρω από την τοποθεσία που βρίσκεται ο χρήστης εκείνη την δεδομένη χρονική στιγμή. Επίσης, στον χάρτη απεικονίζονται δύο κουμπιά, τις λειτουργίες των οποίων θα αναλύσουμε παρακάτω,στα επόμενα παράθυρα προτύπων.

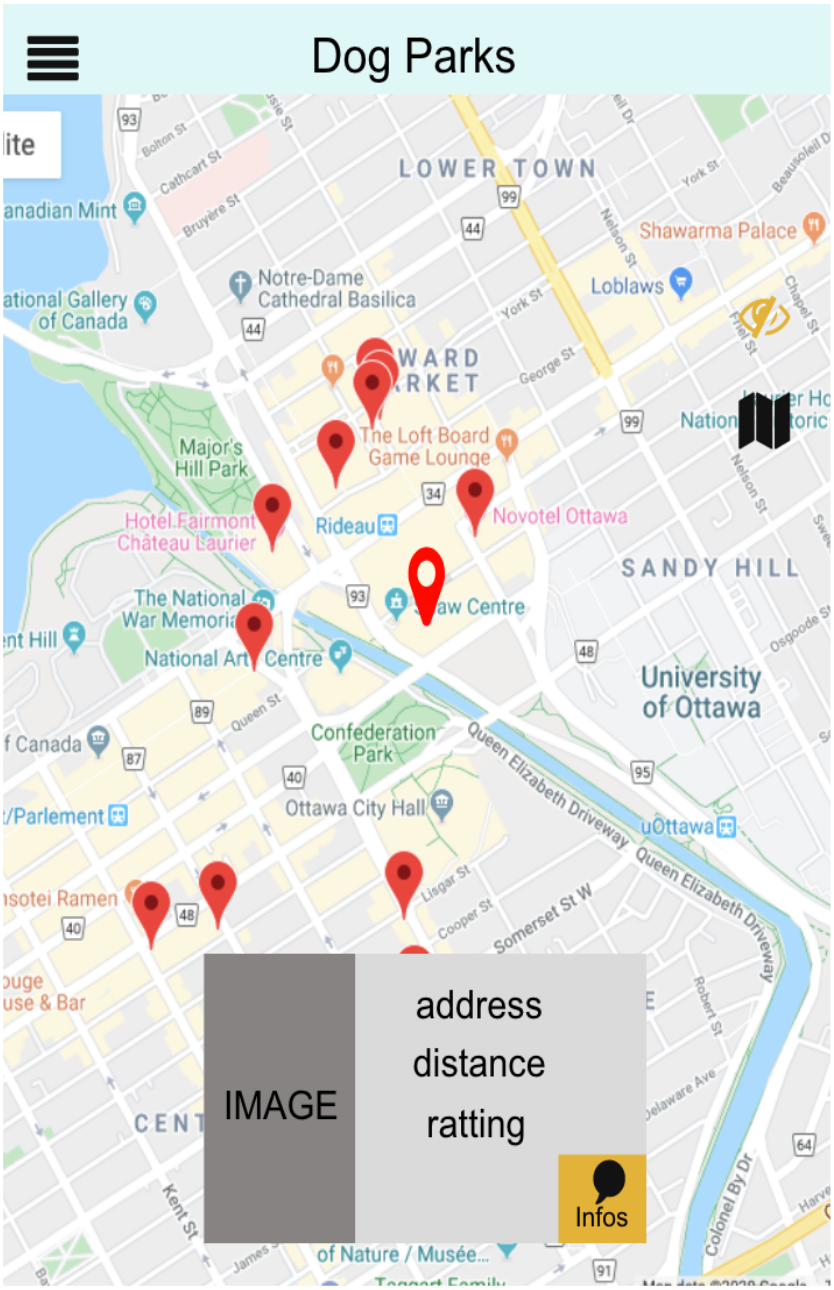


Σχήμα 4.8: Παράθυρο παράδειγμα πρωτοτύπου με τις τοποθεσίες για PetShops

Στο παράθυρο, παρατηρείται ότι υπάρχει το κουμπί εμφάνισης λίστας, ενώ θα έπρεπε να υπάρχει το κουμπί επιστροφής πίσω στην προηγούμενη σελίδα, δηλαδή την HomePage. Αυτό το σφάλμα έχει διορθωθεί στο πέμπτο κεφάλαιο.

4.2.8 Προβολή Carousel πληροφοριών για τις τοποθεσίες

Στο αναφερόμενο παράθυρο, παρατηρούμε τι συμβαίνει όταν ο χρήστης επιλέξει το κουμπί ”View”. Όταν πατήσει λοιπόν το κουμπί, αυτό γίνεται πορτοκαλί για να ενημερώσει τον χρήστη ότι το κουμπί έχει ήδη πατηθεί. Στο κάτω μέρος της οθόνης εμφανίζεται ένα carousel πλαισίων με πληροφορίες για τις τοποθεσίες, με πρώτη επιλογή την πιο κοντινή. Ο χάρτης θα προσανατολίζεται στη συγκεκριμένη τοποθεσία κάθε φορά που ο χρήστης θα κάνει ”Swipe” με το δάχτυλο του στο carousel πλαισίων.



Σχήμα 4.9: Παράθυρο παράδειγμα πρωτοτύπου με το Carousel πληροφοριών για τις τοποθεσίες

Οι πληροφορίες που θα εμφανίζει το carousel για την κάθε τοποθεσία θα είναι οι εξής:

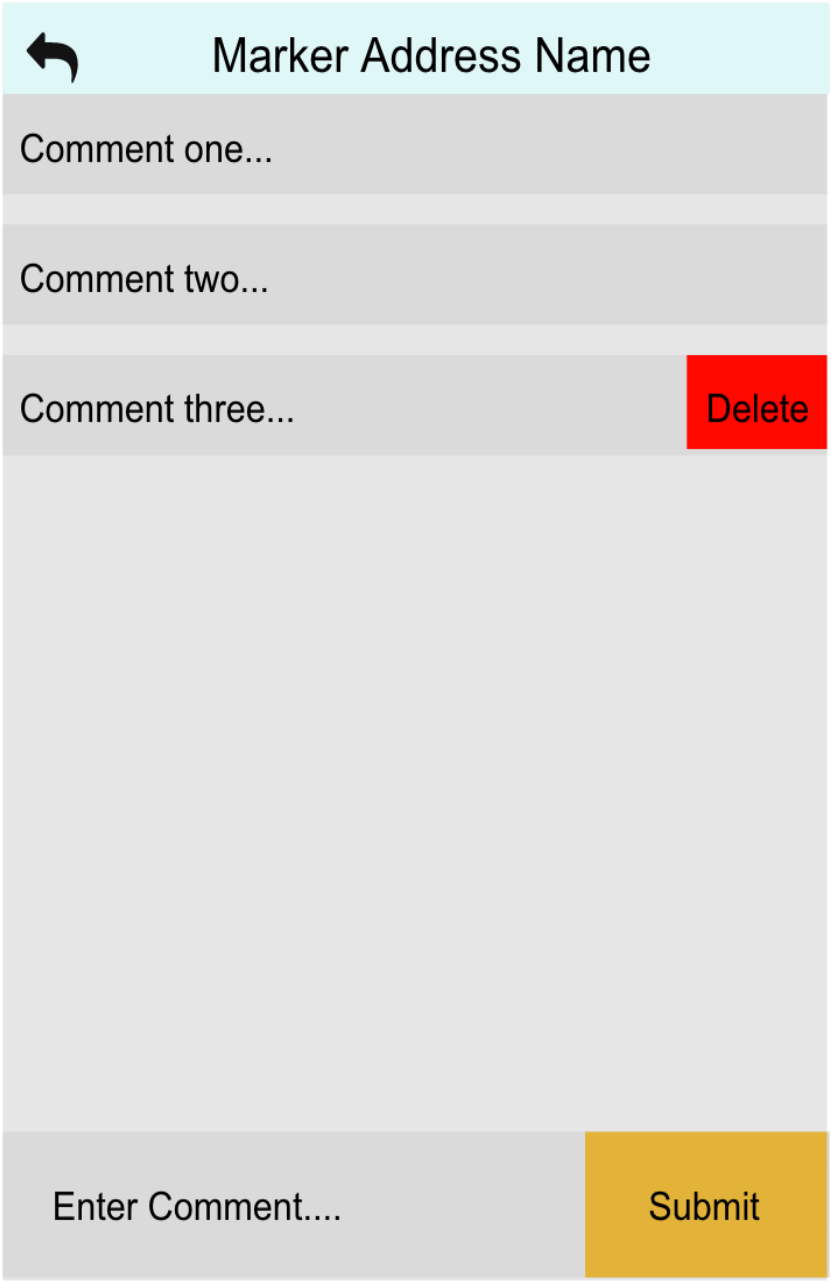
- Είτε μια εικόνα της τοποθεσίας που υπάρχει στο Google Maps API είτε μια default εικόνα.
- Η ονομασία της τοποθεσίας και η διεύθυνσή της.
- Η απόσταση της από το σημείο του χρήστη.
- Η βαθμολογία της στο Google Maps API

- Ένα κουμπί infos το οποίο θα μεταφέρει τους χρήστες στην οθόνη με τα σχόλια της κάθε τοποθεσίας ξεχωριστά όπως θα δούμε παρακάτω (εικόνα 4.10).

Τέλος, πρέπει να σημειωθεί ότι η εικόνα του παραπάνω πλαισίου carousel θα είναι πολύ καλύτερα σχεδιασμένη στο τελικό αποτέλεσμα της εφαρμογής.

4.2.9 Προβολή σχόλιον για τις τοποθεσίες

Όπως αναφέρθηκε προηγουμένως, αφού έχει επιλέξει ο χρήστης το κουμπί "View" και εμφανιστεί το carousel στο κάτω πλαίσιο της οθόνης με τις πληροφορίες της κάθε τοποθεσίας, παρατηρείται ένα κουμπί infos, το οποίο μεταφέρει τον χρήστη στην οθόνη comments όπου προβάλλονται οι κριτικές και τα σχόλια που υπάρχουν ήδη από τους υπόλοιπους χρήστες για τη συγκεκριμένη τοποθεσία.



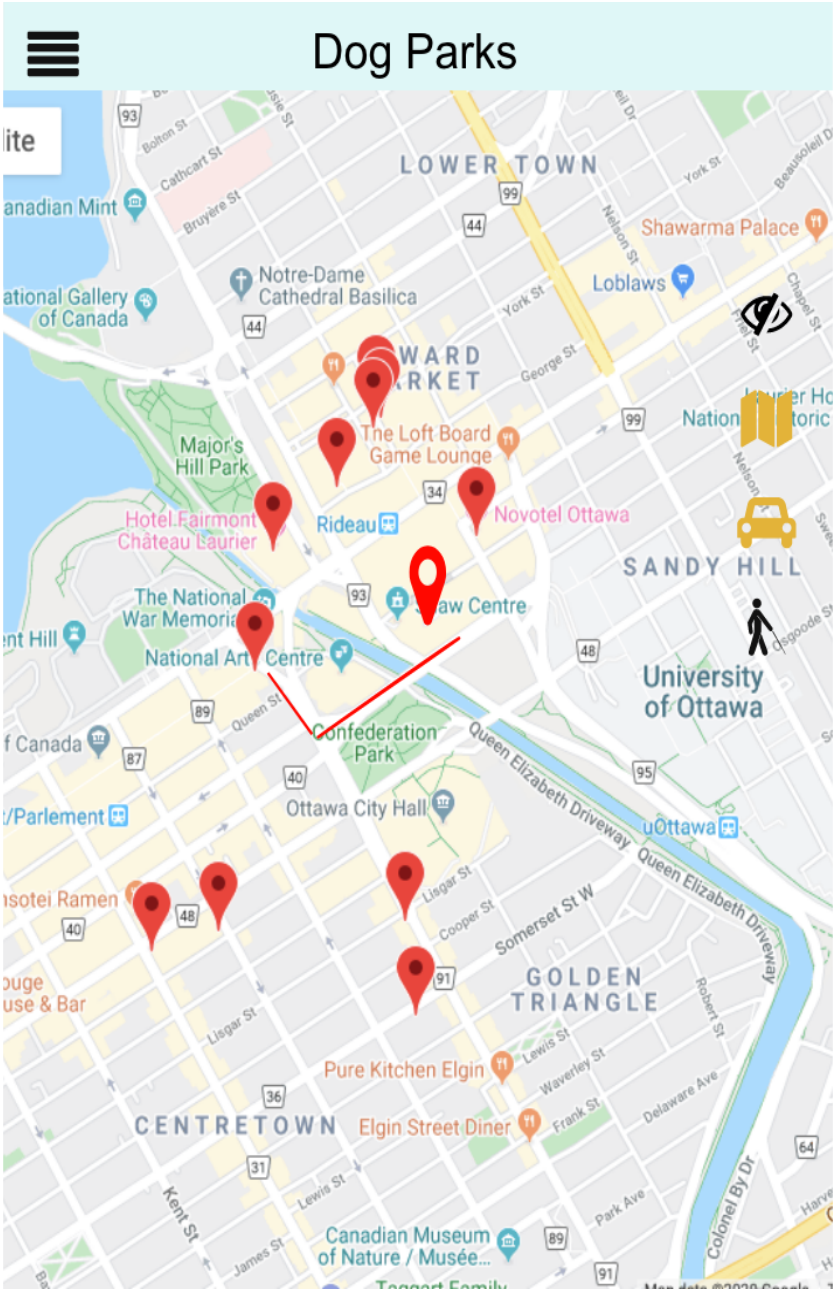
Σχήμα 4.10: Παράθυρο σχόλιον πρωτοτύπου

Στη οθόνη αυτή θα δίνεται η δυνατότητα στον χρήστη να παρακολουθεί τα σχόλια αλλά και τις κριτικές που έχουν κάνει οι υπόλοιποι χρήστες, όσον αφορά την τοποθεσία που έχει επιλέξει να επισκεφτεί (της οποίας το όνομα θα φαίνεται στο επάνω μέρος της σελίδας). Οι ενέργειες στις

οποίες έχει πρόσβαση ο χρήστης στο συγκεκριμένο παράθυρο, είναι η επιλογή να πληκτρολογήσει ένα σχόλιο και να το κάνει submit επιλέγοντας το κουμπί με αποτέλεσμα να στέλνεται στην βάση για να αποθηκευτεί και να εμφανιστεί τελικώς στην οθόνη. Επίσης, σε περίπτωση λάθους του παρέχεται η δυνατότητα διαγραφής του σχολίου χωρίς βέβαια να μπορεί να διαγράψει σχόλια άλλων χρηστών και τέλος θα έχει την επιλογή να γυρίσει πίσω στην προηγούμενη οθόνη.

4.2.10 Προβολή Μεταφοράς στην τοποθεσία

Αφού ο χρήστης τελειώσει με την επιλογή ”View” των τοποθεσιών και τις πληροφορίες σχετικά με αυτές,όπως προαναφέρθηκαν στα από πάνω παράθυρα προτύπου,στον χάρτη υπάρχει και η δυνατότητα επιλογής ”Directions” όπου εμφανίζεται στον χρήστη η διαδρομή που χρειάζεται να ακολουθήσει για να φτάσει στον προορισμό. Η διαδρομή αυτή προσαρμόζεται είτε με αμάξι είτε με τα πόδια.

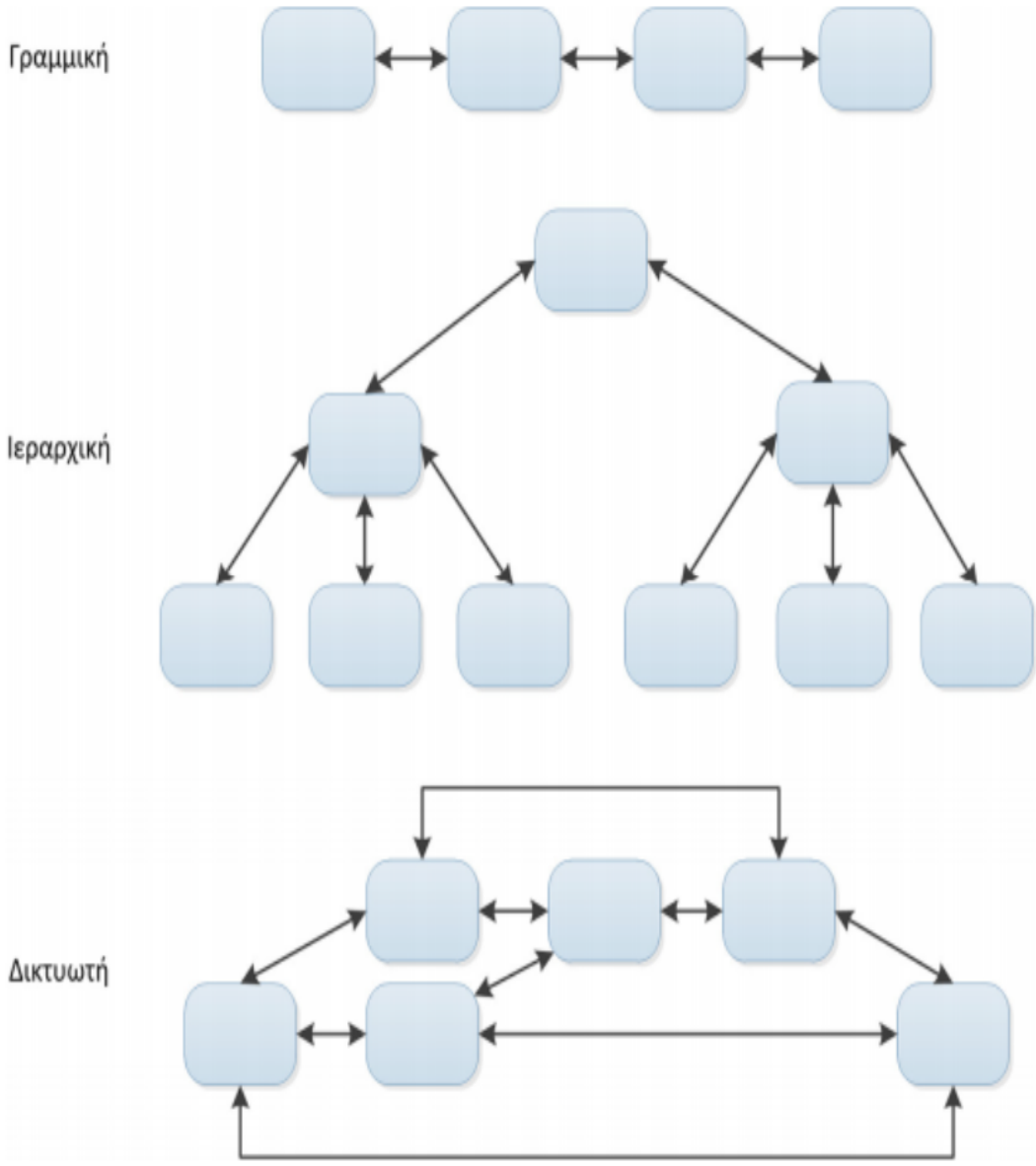


Σχήμα 4.11: Παράθυρο μεταφοράς στην τοποθεσία πρωτοτύπου

4.2.11 Οργάνωση Οθονών

Στην παρούσα διπλωματική εργασία, για την καλύτερη ανάλυση και παρουσίαση της εφαρμογής, ακολουθεί το παρακάτω σχεδιάγραμμα που δείχνει πιο καθαρά τη σύνδεση μεταξύ των παραθύρων του προτύπου στις παραπάνω περιγραφές. Υπάρχουν τρεις τρόποι οργάνωσής των οθονών και είναι οι εξής:

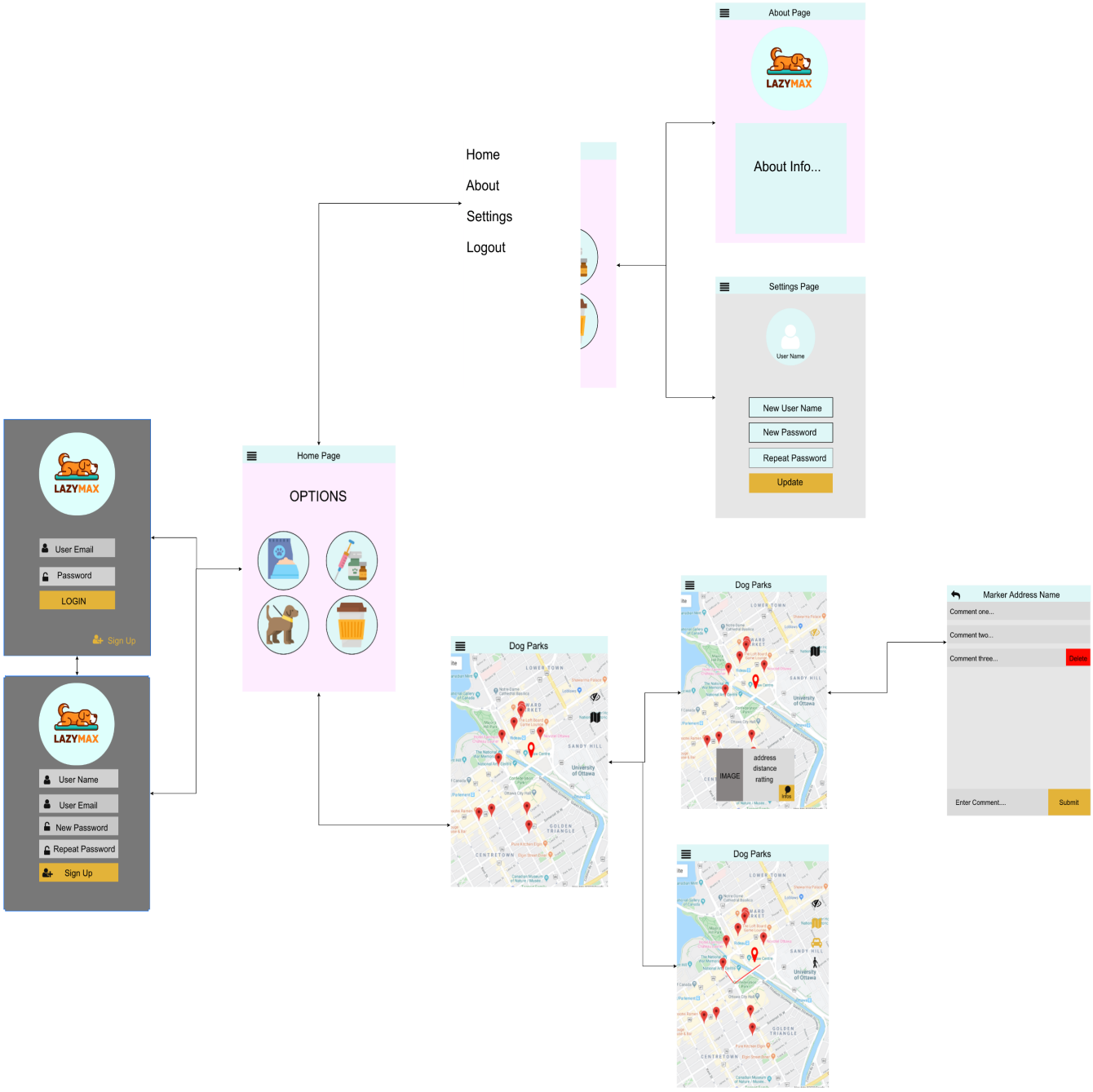
- Γραμμική
- Ιεραρχική
- Δικτυωτή



Σχήμα 4.12: Τρόποι Οργάνωσης Εφαρμογής

Παρατηρώντας και παρακολουθώντας όσα αναφέρθηκαν στα προηγούμενα κεφάλαια γίνεται αντιληπτό ότι ακολουθείται ιεραρχική οργάνωση στη σύνδεση των παραθύρων της εφαρμογής. Η ιεραρχική οργάνωση έχει ως σκοπό να παραχωρεί στον χρήστη της εφαρμογής την ελευθερία να επεξεργάζεται παράλληλα πάνω από ένα παράθυρο. Επομένως, στην κορυφή της πυραμίδας της

ιεραρχικής οργάνωσης βρίσκεται η Home Page στην οποία έφτασε ο χρήστης είτε μέσω εγγραφής είτε μέσω σύνδεσης τους στην εφαρμογή. Από εκεί συνδέονται άμεσα και άλλα κύρια παράθυρα και μέσα από αυτά οδηγείται σε άλλα παράθυρα που προκύπτουν κατεβαίνοντας την πυραμίδα της ιεραρχικής οργάνωσης.



Σχήμα 4.13: Οργάνωση Οθονών Πρωτοτύπου

4.3 Σενάρια χρήσης εφαρμογής

Στο υποκεφάλαιο αυτό, θα παρουσιαστούν ορισμένα πιθανά σενάρια αλληλεπίδρασης του χρήστη με την εφαρμογή. Αυτό, αποσκοπεί στο να κατανοηθεί σε βάθος η εφαρμογή και οι δυνατότητές της. Το συγκεκριμένο αποτελεί ίσως και ένα από τα σημαντικότερα στάδια σχεδιασμού και υλοποίησης της εφαρμογής. Σκοπός της έρευνας αυτής είναι να μπορέσουμε να καλύψουμε όλες τις εκδοχές που μπορεί να προκύψουν κατά τη διάρκεια που ο χρήστης αλληλεπιδρά με την εφαρμογή.

4.3.1 Σενάριο Χρήσης 1

Στόχος: Ο χρήστης επιθυμεί να συνδεθεί στην εφαρμογή.

Πίνακας 4.1: Σενάριο χρήσης 1

Ενέργεια χρήστη	Απόκριση εφαρμογής
Όταν ο χρήστης-ιδιοκτήτης σκυλιών επιλέξει το εικονίδιο της εφαρμογής στην οθόνη του κινητού του.	Ανοίγει η εφαρμογή και εμφανίζεται η πρώτη οθόνη δηλαδή η οθόνη εισαγωγής (Login Page)
Ο χρήστης συμπληρώνει τα στοιχεία σύνδεσης δηλαδή email και κωδικό.	1. Σε περίπτωση εισαγωγής λανθασμένων στοιχείων, εμφανίζεται μήνυμα ”Wrong Credentials” για να ειδοποιήσει τον χρήστη και ύστερα,αφού κλείσει την ειδοποίηση, μπορεί να ξανά προσπαθήσει να εισαγάγει τα σωστά δεδομένα. 2. Σε περίπτωση εισαγωγής σωστών στοιχείων, ο χρήστης οδηγείται στην κεντρική σελίδα δλδ την Home Page.

4.3.2 Σενάριο Χρήσης 2

Στόχος: Ο χρήστης επιθυμεί να εγγραφεί στην εφαρμογή.

Πίνακας 4.2: Σενάριο χρήσης 2

Ενέργεια χρήστη	Απόκριση εφαρμογής
Όταν ο χρήστης-ιδιοκτήτης σκυλιών εισάγεται για πρώτη φορά στην εφαρμογή μεταφέρεται απο την Login Page στην Sign up Page με το πάτημα του κουμπιού signup ώστε να εγγραφεί στην εφαρμογή.	Αφού φορτώσει η οθόνη εγγραφής, στον χρήστη εμφανίζεται η φόρμα συμπλήρωσης με τα στοιχεία username, email, password και repeat password.
Ο χρήστης συμπληρώνει τα στοιχεία εγγραφείς δηλαδή user-name,email,password και repeat password.	1. Σε περίπτωση εισαγωγής ίδιου email στοιχείων, εμφανίζεται μήνυμα ”User email already exist” για να ειδοποιήσει τον χρήστη και ύστερα,αφού κλείσει την ειδοποίηση, μπορεί να ξανά προσπαθήσει να εισαγάγει τα σωστά δεδομένα. 2. Σε περίπτωση εισαγωγής κωδικού διαφορετικού από εκείνων επανάληψης κωδικού εμφανίζεται στον χρήστη το μήνυμα ”Passwords dont match”για να ειδοποιήσει τον χρήστη και ύστερα, αφού κλείσει την ειδοποίηση, μπορεί να ξανά προσπαθήσει να εισαγάγει τα σωστά δεδομένα. 3. Σε περίπτωση που ο χρήστης δεν συμπληρώσει όλα τα πεδία εμφανίζεται μήνυμα ”Please fill up all 4 fields”, αφού κλείσει την ειδοποίηση, μπορεί να ξανά προσπαθήσει να συμπληρώσει όλα τα πεδία. 4. Σε περίπτωση εισαγωγής σωστών στοιχείων, ο χρήστης οδηγείται στην κεντρική σελίδα δηλαδή την Home Page. 5. Σε περίπτωση που βρεθεί σε αυτήν την οθόνη ενώ έχει ήδη λογαριασμό, του δίνεται η δυνατότητα επιστροφής στην προηγούμενη σελίδα (Login Page)

4.3.3 Σενάριο Χρήσης 3

Στόχος: Ο χρήστης επιθυμεί να βρεί τις τοποθεσίες για πάρκα, κτηνιατρεία, Pet Shop ή καφετέριες φιλικές προς τον σκύλο.

Πίνακας 4.3: Σενάριο χρήσης 3

Ενέργεια χρήστη	Απόκριση εφαρμογής
Όταν ο χρήστης εισάγεται με επιτυχία στην εφαρμογή και μεταφέρεται στην κύρια οθόνη στην Home Page.	Αφού φορτώσει η οθόνη Home Page, στον χρήστη εμφανίζονται οι τέσσερις επιλογές (κουμπιά) εύρεσης τοποθεσιών για πάρκα για σκυλιά, κτηνιατρεία, καφετέριες και Pet Shops καθώς και την επιλογή εμφάνισης λίστας των υπόλοιπων οθονών στην εφαρμογή.
Ο χρήστης επιλέγει ένα από τα κουμπιά επιλογής εύρεσης τοποθεσιών για τις παραπάνω επιλογές.	Σε κάθε περίπτωση επιλογής εύρεσης τοποθεσιών ο χρήστης μεταφέρεται σε έναν χάρτη όπου εμφανίζεται η τοποθεσία του και σε ακτίνα χιλίων μέτρων εμφανίζονται όλες οι τοποθεσίες (εάν υπάρχουν) για την αντίστοιχη αναζήτηση.
Ο χρήστης επιλέγει την επιλογή εμφάνισης λίστας των υπόλοιπων οθονών στην εφαρμογή.	Σε αυτή την περίπτωση εμφανίζεται στον χρήστη μια λίστα με τις υπόλοιπες οθόνες που του δίνει δυνατότητα η εφαρμογή να επισκεφτεί.

4.3.4 Σενάριο Χρήσης 4

Στόχος: Ο χρήστης επιθυμεί να επισκεφτεί την οθόνη About Page.

Πίνακας 4.4: Σενάριο χρήσης 4

Ενέργεια χρήστη	Απόκριση εφαρμογής
Όταν ο χρήστης εισάγεται με επιτυχία στην εφαρμογή και μεταφέρεται στην κύρια οθόνη όπου επιλέγει το κουμπί εμφάνισης λίστας υπόλοιπων οθονών και πατήσει την επιλογή About Page.	Αφού ο χρήστης επιλέξει την επιλογή εμφάνισης λίστας των υπόλοιπων οθονών στην εφαρμογή και πατήσει την επιλογή About Page, μεταφέρεται σε αυτήν την σελίδα όπου μπορεί να ενημερωθεί για την ίδια την εφαρμογή, τι προσφέρει, ποιος είναι ο κύριος λόγος ύπαρξής της και τι μπορεί ο χρήστης να εκμεταλλευτεί από την εφαρμογή.
Ο χρήστης βρίσκεται στην οθόνη About Page και διαλέγει την επιλογή εμφάνιση λίστας των υπόλοιπων οθονών στην εφαρμογή.	Σε αυτή την περίπτωση εμφανίζεται στον χρήστη μια λίστα με τις υπόλοιπες οθόνες που περιέχει η εφαρμογή για τον χρήστη να επισκεφτεί.

4.3.5 Σενάριο Χρήσης 5

Στόχος: Ο χρήστης επιθυμεί να επισκεφτεί την οθόνη Settings Page.

Πίνακας 4.5: Σενάριο χρήσης 5

Ενέργεια χρήστη	Απόκριση εφαρμογής
Αφού ο χρήστης εισάγεται με επιτυχία στην εφαρμογή και μεταφέρεται στην κύρια οθόνη όπου επιλέγει το κουμπί εμφάνισης λίστας υπόλοιπων οθονών και πατήσει την επιλογή Settings Page	Αφού ο χρήστης επιλέξει την επιλογή εμφάνισης λίστας των υπόλοιπων οθονών στην εφαρμογή και πατήσει την επιλογή Settings Page, μεταφέρεται σε αυτήν την σελίδα όπου εμφανίζεται μια λίστα εισαγωγής προσωπικών δεδομένων και μπορεί να να αλλάξει, εάν επιθυμεί, τα προσωπικά του δεδομένα εισάγοντας τα καινούρια στοιχεία.
Ο χρήστης συμπληρώνει τα στοιχεία ανανέωσης δηλαδή username,password και repeat password.	1. Σε περίπτωση εισαγωγής κωδικού διαφορετικού από εκείνων στην ”Επανάληψη Κωδικού” εμφανίζεται στον χρήστη το μήνυμα ”Passwords dont match” και ύστερα,αφού κλείσει την ειδοποίηση, μπορεί να ξανά προσπαθήσει να εισαγάγει τα σωστά δεδομένα. 2. Σε περίπτωση που ο χρήστης δεν συμπληρώσει όλα τα πεδία εμφανίζεται μήνυμα ”Please fill up all 3 fields”,αφού κλείσει την ειδοποίηση, μπορεί να ξανά προσπαθήσει να συμπληρώσει όλα τα πεδία. 3. Σε περίπτωση εισαγωγής σωστών στοιχείων, ο χρήστης οδηγείται στην κεντρική σελίδα δηλαδή την Home Page και ειδοποιεί τον χρήστη με μήνυμα ”Update Succeeded”.
Ο χρήστης βρίσκεται στην οθόνη Settings Page και επιλέγει την επιλογή εμφάνισης λίστας των υπόλοιπων οθονών στην εφαρμογή.	Σε αυτή την περίπτωση εμφανίζεται στον χρήστη μια λίστα με τις υπόλοιπες οθόνες που περιέχει η εφαρμογή για τον χρήστη να επισκεφτεί.

4.3.6 Σενάριο Χρήσης 6

Στόχος: Ο χρήστης επιθυμεί να επιλέξει το κουμπί εύρεσης τοποθεσιών για ένα από τις τέσσερις επιλογές(π.χ. καφετέριες φιλικές προς τον σκύλο).

Πίνακας 4.6: Σενάριο χρήσης 6

Ενέργεια χρήστη	Απόκριση εφαρμογής
Όταν ο χρήστης εισάγεται με επιτυχία στην εφαρμογή και μεταφέρεται στην κύρια οθόνη όπου επιλέγει το κουμπί εμφάνισης τοποθεσιών για καφετέριες φιλικές προς τον σκύλο.	Αφού ο χρήστης επιλέξει την εύρεση τοποθεσιών για καφετέριες φιλικές προς τον σκύλο μεταφέρεται σε έναν χάρτη όπου εμφανίζει: • Την τοποθεσία του χρήστη και σε ακτίνα χιλίων μέτρων γύρω από τον χρήστη τις τοποθεσίες όλων των καφετεριών φιλικές προς τον σκύλο. • Κουμπί επιλογής View. • Κουμπί επιλογής directions. • Κουμπί επιστροφής στην κύρια οθόνη Home Page.
Ο χρήστης βρίσκεται στον χάρτη για την εύρεση τοποθεσιών για καφετέριες φιλικές προς τον σκύλο και επιλέγει την επιλογή πίσω.	Σε αυτή την περίπτωση ο χρήστης επιστρέφει πίσω στην προηγούμενη οθόνη δηλαδή την οθόνη Home Page.

4.3.7 Σενάριο Χρήσης 7

Στόχος: Ο χρήστης βρίσκεται στον χάρτη εύρεσης τοποθεσιών για μία από τις τέσσερις επιλογές(π.χ. καφετέριες φιλικές προς τον σκύλο) και επιθυμεί να επιλέξει το κουμπί View.

Πίνακας 4.7: Σενάριο χρήσης 7

Ενέργεια χρήστη	Απόκριση εφαρμογής
Όταν ο χρήστης βρίσκεται στον χάρτη εύρεσης τοποθεσιών για καφετέριες και επιλέξει την επιλογή View.	Σε αυτή την περίπτωση εμφανίζεται στον χρήστη ένα carousel πλαισίων στο κάτω μέρος τις οθόνης με τις πληροφορίες κάθε τοποθεσίας όπως: • Εικόνα της τοποθεσίας (έαν διατίθεται από το Google Maps API). • την διεύθυνση και ονομασία της τοποθεσίας. • Την απόσταση της τοποθεσίας από το σημείο που βρίσκεται ο χρήστης. • Την βαθμολογία της τοποθεσίας με βάση την Google. • Ένα κουμπί infos που σε οδηγεί στην Comment Page. Επίσης ο χάρτης προσανατολίζεται στην τοποθεσία που δείχνει το πλαίσιο,γίνεται πορτοκαλί το κουμπί View για να ενημερώσει τον χρήστη ότι το έχει επιλέξει και ταυτόχρονα απενεργοποιείται το κουμπί Directions.
Ο χρήστης κάνει swipe με το δάχτυλο του προς τα δεξιά ή αριστερά πάνω στο carousel πλαισίων με τις πληροφορίες των τοποθεσιών.	Σε αυτή την περίπτωση το carousel πλαισίων με τις πληροφορίες των τοποθεσιών μεταφέρεται αριστερά ή δεξιά ανάλογα με το swipe που έκανε ο χρήστης με το δάχτυλο του και εμφανίζει το επόμενο ή προηγούμενο πλαίσιο πληροφοριών της συγκεκριμένης τοποθεσίας ενώ παράλληλα ο χάρτης προσανατολίζεται στην συγκεκριμένη τοποθεσία.
Ο χρήστης ξανά πατάει το κουμπί View.	Σε αυτή την περίπτωση κλείνει το carousel πλαισίων με τις πληροφορίες τον τοποθεσιών,από πορτοκαλί το κουμπί γίνεται μαύρο για να δηλώσει στον χρήστη ότι είναι κλειστό και ο χρήστης από εκεί και πέρα επιστρέφει στο σενάριο 6 (πίνακας 4.6).

4.3.8 Σενάριο Χρήσης 8

Στόχος: Ο χρήστης επιθυμεί να επιλέξει το κουμπί infos που βρίσκεται στο πλαίσιο με τις πληροφορίες μιας τοποθεσίας για να μεταφερθεί στην οθόνη Comment Page .

Πίνακας 4.8: Σενάριο χρήσης 8

Ενέργεια χρήστη	Απόκριση εφαρμογής
Όταν ο χρήστης επιλέξει το κουμπί infos που βρίσκεται πάνω στο πλαίσιο πληροφοριών μιας τοποθεσίας	Αφού ο χρήστης επιλέξει το κουμπί infos, μεταφέρεται στην οθόνη Comment Page όπου εμφανίζεται μια λίστα με σχόλια και κριτικές χρηστών,χρονολογικά τοποθετημένες με την πιο πρόσφατη επάνω και την πιο παλιά στο κάτω μέρος, για τη συγκεκριμένη τοποθεσία, της οποίας το όνομα είναι στο επάνω μέρος της οθόνης,καθώς και τις εξής επιλογές: • Ένα πλαίσιο εισαγωγής κειμένου με κουμπί submit για να σχολιάσει ο ίδιος και να εκφράσει κριτική για την τοποθεσία που επισκέφτηκε. • Να διαγράψει κάποιο σχόλιο του. • Να επιλέξει το κουμπί πίσω.
Ο χρήστης επιλέγει να πληκτρολογήσει σχόλιο και να το ανεβάσει.	Σε αυτή την περίπτωση ο χρήστης πρέπει να πληκτρολογήσει το σχόλιο του στο πλαίσιο εισαγωγής κειμένου και να πατήσει submit. Στην περίπτωση που δεν πληκτρολογήσει κάτι εμφανίζεται στον χρήστη το μήνυμα ”Please insert comment” για να τον ενημερώσει.Αφού ολοκληρώσει επιτυχώς τις προϋποθέσεις σχολιασμού ανανεώνεται η οθόνη με το σχόλιο του στο επάνω μέρος.

Ο χρήστης επιλέγει να διαγράψει σχόλιο του,κάνοντας swipe με το δάχτυλο του προς τα αριστερά.	Σε αυτή την περίπτωση με την κίνηση του δάχτυλο του εμφανίζεται δίπλα από το σχόλιο το κουμπί ”Delete” .Στην περίπτωση που το πατήσει και το σχόλιο ανήκει στον ίδιο τότε ανανεώνεται η οθόνη χωρίς να υπάρχει το συγκεκριμένο σχόλιο. Εάν πατήσει ”Delete” και δεν ανήκει το σχόλιο στον ίδιο εμφανίζεται το μήνυμα ”This post doesn’t belong to you” για να ενημερώσει τον χρήστη και η οθόνη παραμένει όπως είναι.
Ο χρήστης επιλεγεί το κουμπί πίσω.	Σε αυτήν την περίπτωση επιστρέφει στην προηγούμενη οθόνη.

4.3.9 Σενάριο Χρήσης 9

Στόχος: Ο χρήστης βρίσκεται στον χάρτη εύρεσης τοποθεσιών για μία από τις τέσσερις επιλογές(π.χ. καφετέριες φιλικές προς τον σκύλο) και επιθυμεί να επιλέξει το κουμπί ”Directions”.

Πίνακας 4.9: Σενάριο χρήσης 9

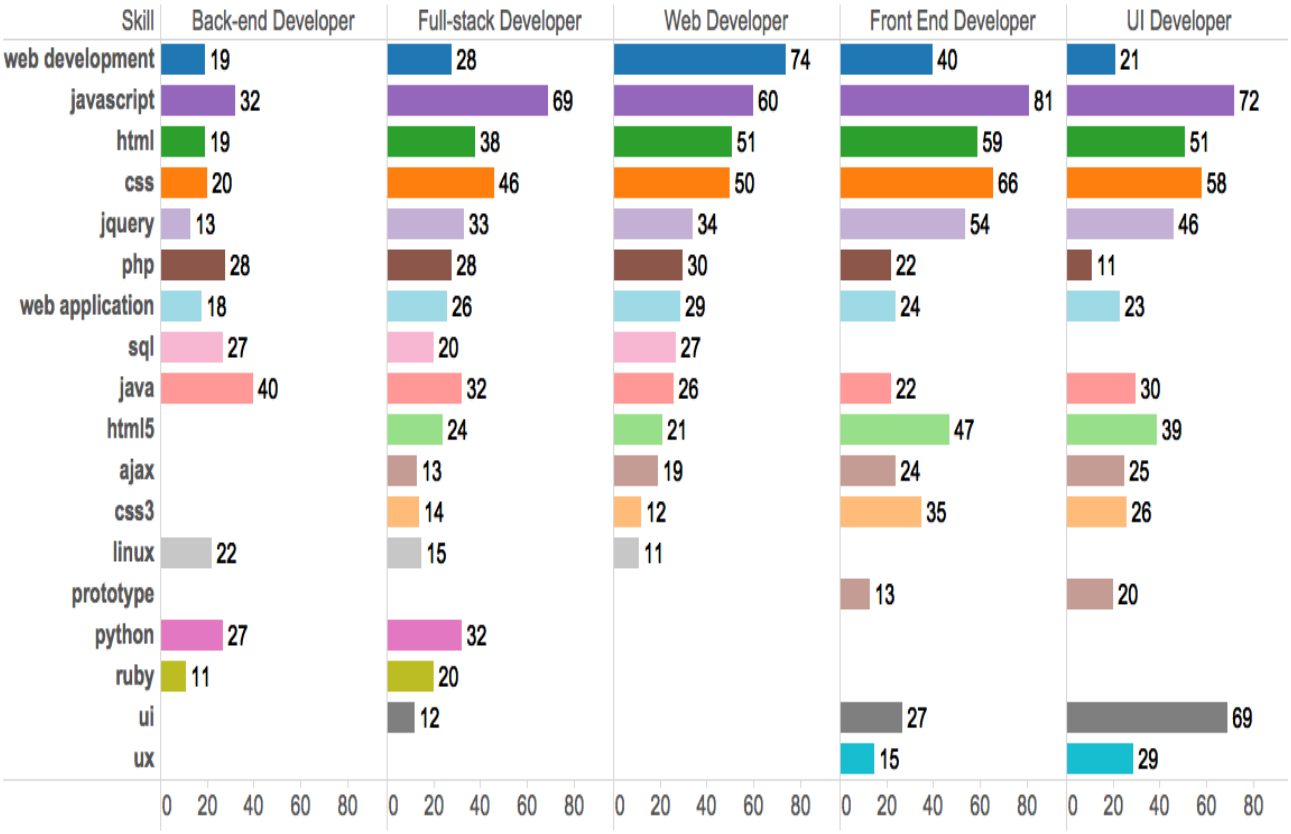
Ενέργεια χρήστη	Απόκριση εφαρμογής
Όταν ο χρήστης βρίσκεται στον χάρτη εύρεσης τοποθεσιών για καφετέριες φιλικές προς τα σκυλιά και επιλέξει την επιλογή ”Directions”.	Σε αυτή την περίπτωση εμφανίζεται στον χρήστη τα εξής κουμπιά: • Κουμπί μετακίνησης με αμάξι. • Κουμπί μετακίνησης με τα πόδια. Επίσης γίνεται πορτοκαλί το κουμπί ”Directions” για να ενημερώσει τον χρήστη ότι το έχει επιλέξει καθώς και ως προεπιλεγμένη επιλογή γίνεται πορτοκαλί το κουμπί μετακίνησης με αμάξι και ταυτόχρονα απενεργοποιείται το κουμπί ”Views”.
Ο χρήστης επιλέγει μια τοποθεσία.	Σε αυτή την περίπτωση δημιουργείται μια κόκκινη διαδρομή από το σημείο του χρήστη μέχρι την τοποθεσία που επέλεξε δείχνοντάς του τον δρόμο πλοήγησης εκεί με αμάξι.
Ο χρήστης επιλέγει το κουμπί πλοήγησης με τα πόδια.	Σε αυτή την περίπτωση δημιουργείται μια κόκκινη διαδρομή από το σημείο του χρήστη μέχρι την τοποθεσία που επέλεξε δείχνοντάς του τον δρόμο πλοήγησης εκεί με τα πόδια ενώ παράλληλα γίνεται πορτοκαλί το κουμπί με τα πόδια.
Ο χρήστης ξανά πατάει το κουμπί ”Directions”.	Σε αυτή την περίπτωση κλείνει το κουμπί ”Directions” και εξαφανίζεται η κόκκινη γραμμή πλοήγησης καθώς και τα κουμπιά με αμάξι και με τα πόδια. Από πορτοκαλί το κουμπί γίνεται μαύρο για να δηλώσει στον χρήστη ότι είναι κλειστό και ο χρήστης από εκεί και πέρα επιστρέφει στο σενάριο 6 (πίνακας 4.6).

5. ΥΛΟΠΟΙΗΣΗ

Στο παρόν κεφάλαιο γίνεται αναφορά στην ανάπτυξη της εφαρμογής. Για την καλύτερη μελέτη της εφαρμογής είναι απαραίτητο να γίνει πρώτα η ανάλυση των βασικών τμημάτων της. Αρχικά θα αναλύσουμε το Front End μέρος της εφαρμογής και τις τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση του. Στη συνέχεια ακολουθεί η μελέτη του Back End μέρους της εφαρμογής και των τεχνολογιών που χρησιμοποιήθηκαν για την υλοποίηση του καθώς και η ανάλυση της βάσης δεδομένων και του διακομιστή (server) του συστήματος. Τέλος, αναλύονται οι τεχνολογίες που εφαρμόστηκαν για την φόρτωση του μεγαλύτερου μέρους της βάσης δεδομένων αλλά και οι βιβλιοθήκες και τα plug-ins που χρησιμοποιήθηκαν.

5.1 Τεχνολογίες Σχεδιασμού Συστημάτων

Η ύπαρξη ενός κατάλληλα σχεδιασμένου και εύχρηστου περιβάλλοντος καθώς και η μέγιστη κάλυψη των υπαρχουσών αναγκών ενός χρήστη αποτελούν τους βασικότερους στόχους στην υλοποίηση μιας εφαρμογής. Επομένως, είναι σημαντικό να δοθεί προσοχή στην κατάλληλη υλοποίηση του back end και στις λειτουργίες του, δηλαδή την ανταλλαγή δεδομένων με την εφαρμογή, τον χρήστη και την βάση δεδομένων. Αντίστοιχα, είναι αναγκαίο να εστιάσουμε στην σχεδίαση και την υλοποίηση του front end τμήματος της εφαρμογής το οποίο αποτελεί το περιβάλλον που θα αλληλεπιδρά ο χρήστης.



Σχήμα 5.1: Front end vs back end programming

5.2 Είδη τεχνολογιών front end

Στην ενότητα αυτή θα μελετήσουμε τα είδη τεχνολογιών του front end. Αρχικά, πρέπει να αναφερθεί, πως στα πλαίσια του διαδικτυακού προγραμματισμού, μία είναι η γλώσσα που είναι η πλέον διαδεδομένη, η Javascript. Η Javascript αποτελεί μία από τις πιο ευέλικτες και αποτελεσματικές γλώσσες υψηλού επιπέδου που μπορεί να χρησιμοποιηθούν από προγραμματιστές διαδικτύου, ενώ παράλληλα τους παρέχει την δυνατότητα ανάπτυξης ιστοσελίδων τόσο σε υπολογιστές όσο και σε κινητές συσκευές.Ο συνδυασμός της με την HTML και CSS αποτελούν τις βασικές τεχνολογίες του World Wide Web.

Τα βασικότερα πλεονεκτήματα της είναι τα εξής:

- Ο κώδικας εκτελείται από τον επεξεργαστή του χρήστη, αντί του εξυπηρετητή ιστού, εξοικονομώντας έτσι εύρος ζώνης.
- Η σύνταξη της είναι αρκετά απλή καθιστώντας την εκμάθησή της ευκολότερη για τους προγραμματιστές.
- Είναι σε μεγάλο βαθμό συμβατή με τις περισσότερες γλώσσες προγραμματισμού σε αντίθεση με άλλες όπως η SSI ή PHP με αποτέλεσμα να χρησιμοποιείται σε περισσότερες εφαρμογές(όπως φαίνεται στην εικόνα 5.1) .

Η Javascript αποτελεί βασικό και αναπόσπαστο στοιχείο για την σχεδίαση ιστοσελίδων, καθώς η HTML και η CSS δεν μπορούν να προσθέσουν την κατάλληλη διαδραστικότητα που χρειάζεται ένας ιστότοπος για να είναι πλήρως λειτουργικός.

Η ύπαρξη τεχνολογιών front end είναι απαραίτητη για καλύτερη εμπειρία ενός χρήστη μιας εφαρμογής σε κινητή συσκευή ή μιας ιστοσελίδας. Για τον λόγο αυτόν υπάρχει μεγάλη εξέλιξη και ανάπτυξη σε τεχνολογίες και γλώσσες προγραμματισμού όπως HTML,CSS και πιο σύνθετες όπως Javascript αλλά και πλήθος βιβλιοθηκών που προκύπτουν από αυτές όπως ReactJS και AngularJS.

5.3 Είδη τεχνολογιών back end

Όπως αναφέρθηκε στην αρχή του παρόντος κεφαλαίου απαραίτητη για την υλοποίηση μιας ολοκληρωμένης εφαρμογής αποτελεί η ύπαρξη τεχνολογιών back end. Η χρήση back end τεχνολογιών καλύπτει σε μεγάλο βαθμό τις κύριες ανάγκες του χρήστη και την αλληλεπίδρασή του με την εφαρμογή.

Ο βασικός ρόλος του back end είναι ο προγραμματισμός ενός διακομιστή μεσολάβησης μεταξύ χρήστη και βάσης δεδομένων. Πιο συγκεκριμένα, όταν η ιστοσελίδα ή η εφαρμογή κάνει αίτημα (POST ή GET) δεδομένων από την βάση το back end λαμβάνει αυτό το αίτημα και το χειρίζεται ανάλογα.

Τέλος, δύο από τις πιο γνωστές και διαδεδομένες γλώσσες προγραμματισμού για υλοποίηση back end είναι η PHP και Java τις οποίες θα αναλύσουμε παρακάτω.

5.3.1 Γλώσσα Προγραμματισμού Java

Η Java αποτελεί μια από τις ισχυρότερες γλώσσες αντικειμενοστραφή προγραμματισμού σήμερα. Προσφέρει ευκολία στην χρήση, αξιοπιστία και συμπεριλαμβάνει ένα υψηλό επίπεδο ασφάλειας. Υλοποιήθηκε ως μια φορητή και cross-platform γλώσσα που δεν ενδιαφέρεται για το λειτουργικό σύστημα, το υλικό ή για τις συσκευές που εγκαθίσταται. Επομένως, μπορεί να προσφέρει πολλά στον προγραμματισμό διαδικτύου και πιο συγκεκριμένα στο back end τμήμα του και τις τεχνολογίες που την αποτελούν.

5.3.2 Γλώσσα Προγραμματισμού PHP

Η PHP (Hypertext Preprocessor) είναι μια γλώσσα προγραμματισμού για την δημιουργία ιστοσελίδων με δυναμικό περιεχόμενο. Η συγκεκριμένη γλώσσα προγραμματισμού είναι εκείνη που χρησιμοποιούμε για το back end μέρος της εφαρμογής μας. Κάποια από τα βασικά πλεονεκτήματα της γλώσσας προγραμματισμού PHP είναι τα εξής :

- Είναι γλώσσα ανοιχτού κώδικα,δηλαδή δεν χρειάζεται να να περιμένουμε για την ανανέωση της έκδοσης της από τον κατασκευαστή.
- Είναι επεκτάσιμη.
- Στηρίζει μεγάλο αριθμό βάσεων δεδομένων.
- Είναι αρκετά απλή, οπότε αποτελεί πολύ καλή επιλογή για αρχάριους, οι οποίοι έχουν αρχίσει μόλις να εμπλέκονται στον κόσμο του προγραμματισμού διαδικτύου.
- Υπάρχουν πολλά frameworks (Laravel, Symfony κ.λπ.) τα οποία έχουν ισχυρή κοινότητα.
- Στηρίζεται από κάθε πάροχο σε σχέση με άλλες γλώσσες προγραμματισμού.

5.4 Λογισμικό που χρησιμοποιήθηκε

5.4.1 Τεχνολογία ανάπτυξης του front end του συστήματος

Για την υλοποίηση και την ανάπτυξη της εφαρμογής στη παρούσα διπλωματική εργασία, υπάρχουν πολλές τεχνολογίες που θα μπορούσαμε να χρησιμοποιήσουμε, κάθε μια από τις οποίες έχει πλεονεκτήματα και μειονεκτήματα. Άρα, για τη βέλτιστη ευχρηστία και την καλύτερη και άνετη εμπειρία του χρήστη είναι αρκετά σημαντικό να χρησιμοποιήσουμε τις καλύτερες τεχνολογίες του διαδικτυακού προγραμματισμού. Όπως αναφέραμε και προηγουμένως, η βέλτιστη γλώσσα προγραμματισμού που χρησιμοποιείται για την σχεδίαση και ανάπτυξη τόσο ιστοσελίδων όσο και εφαρμογών είναι η JavaScript, οπότε γίνεται εύκολα αντιληπτό πως αποτελεί την καλύτερη δυνατή λύση για την υλοποίηση της εφαρμογής μας. Πιο συγκεκριμένα χρησιμοποιούμε μια βιβλιοθήκη της JavaScript, την ReactJS

και ειδικότερα για την σχεδίαση και υλοποίηση της εφαρμογής σε κινητή συσκευή χρησιμοποιήσαμε την React Native .

Η ReactJS αποτελεί βιβλιοθήκη της JavaScript και είναι μια από τις πιο διαδεδομένες βιβλιοθήκες, κατάλληλες για την υλοποίηση και την ανάπτυξη διαδικτυακών εφαρμογών για κινητές συσκευές, ιστοσελίδες καθώς και διεπαφών χρήστη. Κάνει χρήση βοηθητικών βιβλιοθηκών όπως React Router και React Redux, που θα αναλύσουμε παρακάτω, διότι επικεντρώνεται κυρίως στην απόδοση δεδομένων στο μοντέλο αντικείμενου εγγράφου (DOM). Εξίσου σημαντικές, αποτελούν οι οντότητες γνωστές ως Components που μεταφράζονται σε ένα συγκεκριμένο στοιχείο στο DOM με την χρήση της βιβλιοθήκης React DOM. Η React ενημερώνει με αποτελεσματικό τρόπο και επιστρέφει τα Components κατά την ανανέωση δεδομένων κάνοντας πιο εύκολη και ανώδυνη την δημιουργία και υλοποίηση διαδραστικών User Interfaces για την προβολή της δεδομένης κατάστασης. Επιπλέον, κάνουμε χρήση μεθόδων κύκλου ζωής (Lifecycle Methods) ώστε να έχουμε σωστή και βέλτιστη ροή στα Components. Η React χρησιμοποίει τις μεθόδους αυτές με την χρήση ειδικών συναρτήσεων στον κώδικα (π.χ. ComponentDidMount, ComponentWillMount, render κ.λπ.) που θα αναλύσουμε παρακάτω. Τέλος, είναι απαραίτητο να έχουμε γνώση όσον αφορά τις μεθόδους αυτές για την καλύτερη δυνατή διατύπωση κώδικα για εφαρμογές σε React.

Η αποτελεσματικότητα και η ευχρηστία της εφαρμογής μιας κινητής συσκευής οφείλεται στο γεγονός ότι η React χρησιμοποιεί δηλωτικό παράδειγμα (declarative paradigm). Η React ενημερώνει την κατάλληλη στιγμή κάθε κατάσταση (State) στην οποία έχουμε μια σχεδίαση προβολών και επιστρέφει τα κατάλληλα Components με αποτέλεσμα ο κώδικας να διαβάζεται πιο εύκολα σε περίπτωση εντοπισμού σφαλμάτων και να διευκολύνει την διαδικασία debugging καθιστώντας τήν πιο εύκολη και πιο γρήγορη. Πιο συγκεκριμένα, κατά την εισαγωγή δεδομένων η ανάγνωση γίνεται μέσω της χρήσης αναπαράστασης του εικονικού δέντρου DOM,το οποίο με την σειρά του θα προσπαθήσει να βρει τον βέλτιστο τρόπο ενημέρωσης του DOM του προγράμματος περιήγησης. Οι δυνατότητες της React είναι οι εξής:

- Η παροχή Native γραφικού περιβάλλοντος στον χρήστη.
- Υψηλή απόδοση.
- Απόδοση διακομιστή (Server- side rendering).
- Απλό έλεγχο και τεστ κώδικα.
- Framework προσεγγίσιμο από κάθε εφαρμογή.

Οι οντότητες-Components αποτελούν μεγάλο μέρος πάνω στο οποίο βασίζεται η React, δημιουργώντας την ανάγκη να τα μελετήσουμε και να κατανοήσουμε την λειτουργία και την χρησιμότητα τους καθώς και τον λόγο που κάθε εφαρμογή κινητής συσκευής ή ιστοσελίδα που έχει αναπτυχθεί με React αποτελείται από οντότητες-Components. Υπάρχουν δυο βασικά ειδών Components.

1. Λειτουργικές οντότητες (Functional Components): Πρόκειται για συναρτήσεις JavaScript με τις οποίες μπορούμε να δημιουργήσουμε Components τα οποία δέχονται ή και όχι μεταβλητές.
2. Οντότητες-Κλάσεις (Class Components): Πρόκειται για διαφορετικές οντότητες από τις λειτουργικές καθώς μπορούν γνωρίζουν ότι υπάρχουν και άλλες οντότητες μέσα στην εφαρμογή και έχουν την δυνατότητα να επικοινωνήσουν με αυτές αλλά και να ανταλλάξουν δεδομένα.

Η δημιουργία οντοτήτων-Components χωρίζουν την διεπαφή σε πολλά μέρη ,προγραμματίζοντας τα ένα προς ένα και όταν ολοκληρωθεί ο προγραμματισμός τους τα ενώνει για το τελικό αποτέλεσμα της εφαρμογής καθιστώντας την παραπάνω υλοποίηση αρκετά εύκολη για τον χρήστη.

Εκτός από τις οντότητες-Components που αναφέραμε προηγουμένως, είχαμε μιλήσει και για τους μεθόδους κύκλους ζωής της React και των Components. Σε αυτό το σημείο θα αναλυθούν οι μέθοδοι κύκλους ζωής και οι φάσεις που περνάει κάθε Component κατά την λειτουργία τους που είναι τα εξής:

1. Αρχικοποίηση (Initialization): Πρόκειται για την φάση στην οποία δημιουργείται η οντότητα-Component και του παραθέτονται οι προ-επιλεγμένοι παράμετροι. Η φάση αυτή αφορά μόνο τις οντότητες-κλάσεις που αναφέραμε παραπάνω.

2. Τοποθέτηση (Mounting): Πρόκειται για την φάση στην οποία γίνεται rendering του κώδικα ο οποίος έχει επιστρέψει με την μέθοδο render.
3. Ανανέωση (Updating): Πρόκειται για την φάση στην οποία γίνεται η ανανέωση παραμέτρων και καταστάσεων προκειμένου να ξανά πραγματοποιηθεί, η δεύτερη φάση, render της οντότητας.
4. Αφαίρεση (Unmounting): Πρόκειται για την φάση στην οποία γίνεται η αφαίρεση μιας οντότητας από το εικονικό DOM.

Η React σε κάθε μια από τις παραπάνω φάσεις ορίζει και εκτελεί κάποιες συναρτήσεις-διεργασίες, δίνοντας όμως αρκετούς βαθμούς ελευθερίας στον προγραμματιστή να λειτουργήσει και να καθορίσει το τελικό αποτέλεσμα της κατάστασης τους. Όπως:

(a) Αρχικοποίηση (Initialization): Πρόκειται για το στάδιο στο οποίο ο χρήστης κάνει χρήση της συνάρτησης κατασκευαστής(constructor) ώστε να ορίσει παραμέτρους και την αρχική κατάσταση μιας οντότητας.

(b) Τοποθέτηση (Mounting): Πρόκειται για την φάση στην οποία γίνεται η τοποθέτηση της οντότητας στο εικονικό DOM, δηλαδή εμφανίζεται στην ιστοσελίδα, με τη χρήση δυο συναρτήσεων από τον προγραμματιστή που διαθέτει η React:

- Συνάρτηση componentDidMount(): Εκτελείται μετά από την πρώτη εκτέλεση της μεθόδου render.
- Συνάρτηση componentWillMount(): Σε αντίθεση με την προηγουμένη εκτελείται πριν την πρώτη εκτέλεση της μεθόδου render,δηλαδή πριν από την τοποθέτηση της οντότητας στο DOM.

(c) Ανανέωση (Updating): Πρόκειται για την κατάσταση κατά την οποία ο προγραμματιστής χρησιμοποιεί συναρτήσεις για να ανανεώσει κάποια δεδομένα του κατασκευαστή, να πληκτρολογήσει κάτι καινούργιο ή απλά πατήσει κάποιο κουμπί. Οι συναρτήσεις που χρησιμοποιεί είναι οι εξής:

- Συνάρτηση setState(): Δεν αποτελεί ακριβώς μια συνάρτηση,όμως κατά το κάλεσμα της από τον προγραμματιστή πυροδοτείται η ανανέωση οντοτήτων και καταστάσεων.
- Συνάρτηση componentWillUpdate(): Πρόκειται για συνάρτηση που εκτελείται πριν την μέθοδο render και την επανατοποθέτηση της οντότητας στο DOM και πριν γίνει η εκτέλεση του Props() και setState().
- Συνάρτηση componentDidUpdate():Πρόκειται για συνάρτηση που εκτελείται όπως και η componentWillUpdate(),δηλαδή εκτελείται πριν την μέθοδο render και την επανατοποθέτηση της οντότητας στο DOM όμως μετά από την εκτέλεση του Props() και setState().
- Συνάρτηση shouldComponentUpdate(): Πρόκειται για την συνάρτηση όπου ο κύριος ρόλος της είναι να επιβεβαιώνει την React ότι πρόκειται να αλλάξει η έξοδος της οντότητας και καλείται πριν από την επανατοποθέτηση της στο DOM.
- Συνάρτηση componentWillRecieveProps():Πρόκειται για την συνάρτηση που αφορά μόνο τις παραμέτρους μιας οντότητας παρά την κατάσταση της και τίθεται σε λειτουργία όταν μια οντότητα που είναι τοποθετημένη στο DOM ανανεώσει τις παραμέτρους τις ανεξάρτητα αν τις άλλαξε ή όχι.

(d) Αφαίρεση (Unmounting):Πρόκειται για το τελικό στάδιο μεθόδου κύκλου ζωής της οντότητας όπου γίνεται η χρήση μιας συνάρτησης.

- Συνάρτηση componentWillUnmount(): Πρόκειται για την συνάρτηση η οποία μπαίνει σε λειτουργία κατά την αφαίρεση μια οντότητας-Component από το DOM.

Σε αυτό το σημείο,αφού έγινε αναφορά στην React,σχετικά με τις οντότητες και από ποιες συναρτήσεις αποτελείται, θα πρέπει να γίνει λόγος για τη χρήση της native μορφής της React,από την στιγμή που η εφαρμογή μας θα υλοποιηθεί σε οθόνη κινητής συσκευής, δηλαδή την React Native. Κάποια από τα βασικά χαρακτηριστικά της React Native είναι τα εξής:

- Η React Native δεν χειρίζεται το DOM.
- Δεν χρησιμοποιεί HTML ή CSS αλλά κάνει χρήση JavaScript και επιτρέπει την χρήση άλλων γλωσσών όπως Java για Android και Objective-C ή Swift για iOS, κάνοντάς την έτσι ακόμα πιο ευέλικτη.
- Εκτελείται απευθείας από την συσκευή και επικοινωνεί μέσω μιας σειριακής και ασύγχρονης γέφυρας με την εγγενή πλατφόρμα.
- Δίνει την δυνατότητα στην δημιουργία διαδραστικών native εφαρμογών τα οποία δεν βασίζονται στην εμπειρία του χρήστη.
- Για την ευχρηστία του UI χρησιμοποιεί έτοιμα βασικά και αναγκαία native Components όπως View,Text και Image.

Για την καλύτερη κατανόηση μιας εφαρμογής σε React Native πρέπει να γίνουν αντιληπτά τα τέσσερα νήματα(threads) με τα οποία υλοποιείται η εφαρμογή αυτή. Δηλαδή:

(a) UI Thread: Επίσης γνωστό ως κύριο thread έχει χρήση για την απόδοση του εγγενούς χρήστη Android και iOS. Για παράδειγμα, στο android αυτό το thread χρησιμοποιείται για μέτρηση / διάταξη / κλήρωση.

(b) Native Modules Thread: Πρόκειται για το thread με το οποίο η εφαρμογή παίρνει πρόσβαση σε μια API πλατφόρμα.

(c) Render Thread: Προκείται για το thread που χρησιμοποιείται για την σχεδίαση περιβάλλοντος εργασίας του χρήστη μέσω της παραγωγής πραγματικών εντολών OpenGL,μόνο για την περίπτωση λογισμικού Android 5.0.

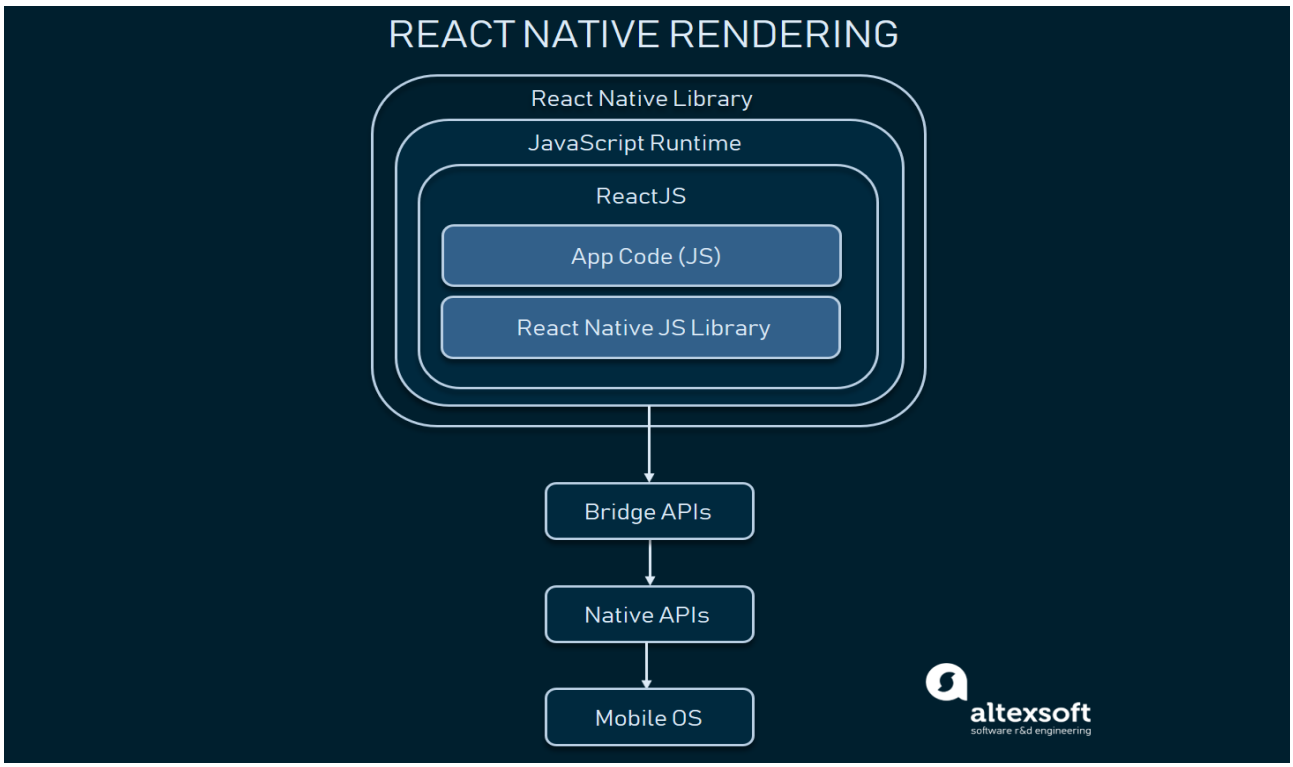
(d) JS Thread: Πρόκειται για το thread που τρέχει την λογική της εφαρμογής. Με άλλα λόγια, στο τέλος κάθε βρόχου συμβάντων αποστέλλονται οι ενημερώσεις στο thread JS.

Κατά τη λειτουργία μιας React Native εφαρμογής περνάμε από διάφορες διαδικασίες όπως:

1. Αρχικά υφίσταται η εκκίνηση της εφαρμογής κατά την οποία το κύριο thread(UI Thread) φορτώνει και εκτελεί τις δέσμες JS (bundles).
2. Αφού έχει φορτωθεί ο κώδικας javascript και γίνει μεταφορά διεργασιών στο JS thread εκτελούνται όλες οι διεργασίες για την καλύτερη εμπειρία του χρήστη.
3. Ύστερα, ξεκινάει να κάνει render η React και αρχίζει η διαδικασία diffing από τον συντονιστή (Reconciler Api) οδηγώντας στην δημιουργία ενός καινούργιου εικονικού DOM σε άλλο νήμα (Thread Shadow).
4. Σε αυτό το στάδιο το Thread Shadow στέλνει τις παραμέτρους που έχει αντίστοιχα υπολογίσει η διάταξη (layout) της αντίστοιχης εφαρμογής.
5. Αφού αποστείλει τις παραμέτρους το Thread Shadow, αποτελεί δουλειά του κύριου thread να τα εμφανίσει στην διεπαφή του χρήστη.

Τέλος, αξίζει να σημειωθούν τα βασικά πλεονεκτήματα της React Native.

- Διαθέτει εύκολη εγκατάσταση μέσω USB καλωδίου από τον υπολογιστή και τρέχει με server του χρήστη αντί να χρειαστεί να ανεβεί η εφαρμογή σε κάποιο App store.
- Βασικό πλεονέκτημα της εφαρμογής αποτελεί το live and hot reloading καθώς επαναφορτίζεται η οθόνη με κάθε αλλαγή στο UI προσφέροντας στον χρήστη μοναδική εμπειρία.
- Υπάρχουν έτοιμα Components για την εύκολη χρήση τους από προγραμματιστές.
- Κάθε κώδικας της εφαρμογής είναι γραμμένο κατά 95% σε Android και IOS αποτελώντας το πλήρες παραγωγικό.



Σχήμα 5.2: React Native Rendering

5.4.2 Τεχνολογία Ανάπτυξης του Back End του Συστήματος

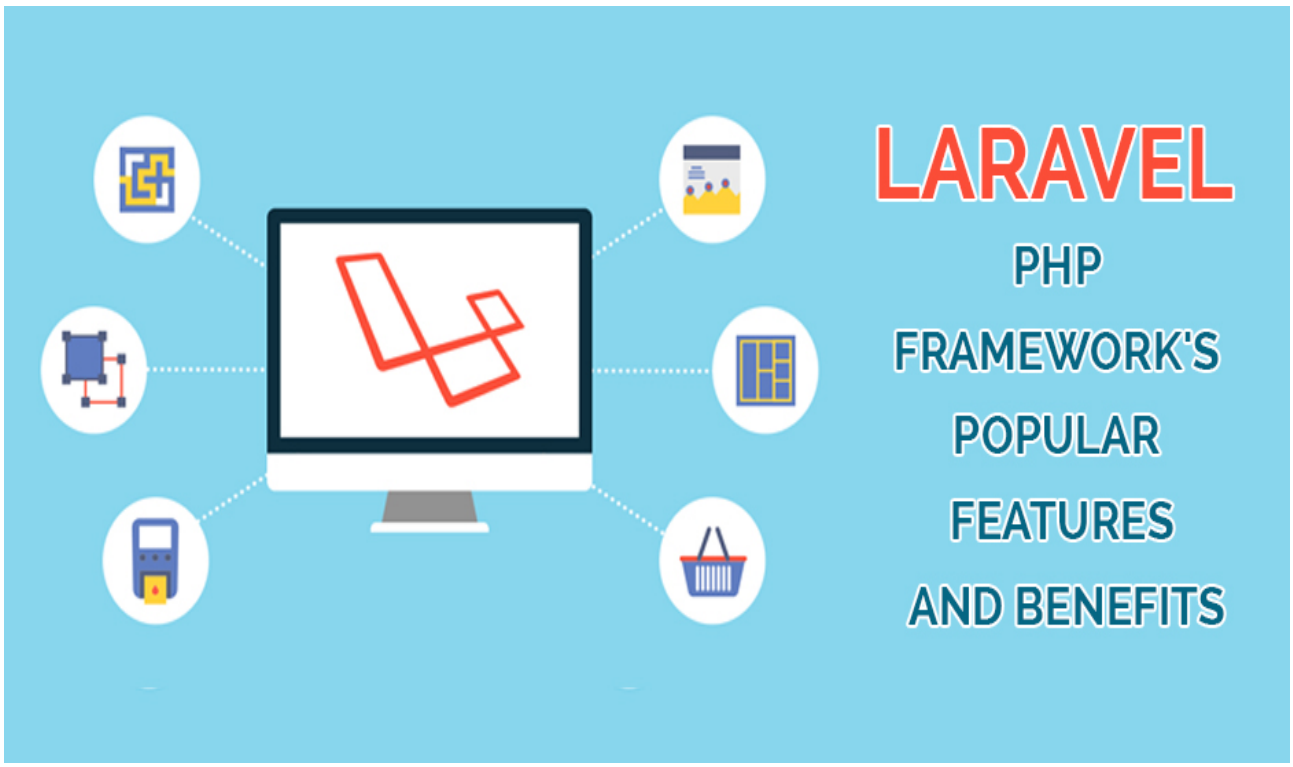
Σε αυτό το υποκεφάλαιο θα μελετηθεί η ανάπτυξη του back end της εφαρμογής μας. Με βάση την μελέτη της εφαρμογής και όπως ήδη προαναφέρθηκε υπάρχουν δύο πιο διαδεδομένες γλώσσες προγραμματισμού η PHP και η JAVA εκ των οποίων επιλέξαμε την PHP για την υλοποίηση της εφαρμογής μας. Πιο συγκεκριμένα, χρησιμοποιήθηκε μια βιβλιοθήκη της PHP, Laravel, η οποία αποτελεί την κορυφαία στον τομέα της. Η Laravel κατέχει την καλύτερη βαθμολογία στο GitHub και διαθέτει τα κατάλληλα εξαρτήματα για την ανάπτυξη και την δημιουργία εφαρμογών και ιστοσελίδων και την συντήρηση τους. Ακόμα, η Laravel προσφέρει:

1. Σύστημα ελέγχου ταυτοποίησης και εξουσιοδότησης. Με αυτόν τον τρόπο η Laravel διαθέτει μεγαλύτερη ασφάλεια και αποτρέπει την είσοδο σε μη εξουσιοδοτημένους χρήστες να εισέλθουν στο σύστημα.

2. Ενσωμάτωση με ειδικά εργαλεία για την μέγιστη απόδοση διαδικτυακών εφαρμογών.

3. Διαμόρφωση σφαλμάτων και χειρισμός εξαιρέσεων οδηγώντας στην καλύτερη εμπειρία που μπορεί να βιώσει ο χρήστης.

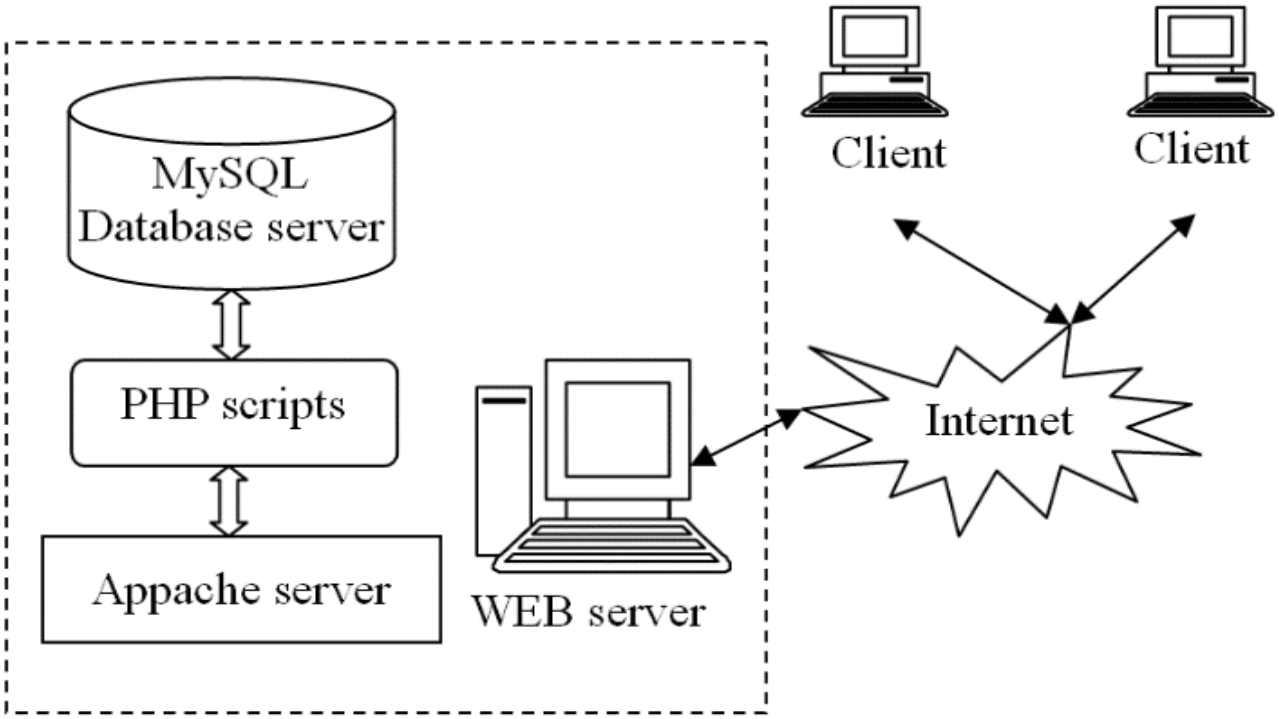
4. Αυτοματοποίηση στον έλεγχο της εφαρμογής. Δηλαδή, η Laravel προσφέρει αυτόματους έλεγχους, προσφέροντας μεγαλύτερη σιγουριά και ασφάλεια σε περίπτωση σφαλμάτων.



Σχήμα 5.3: Laravel Benefits

5.4.3 Λογισμικό Server

Η μεταφορά δεδομένων και η ενημέρωση πληροφοριών μεταξύ της βάσης δεδομένων και των χρηστών, είναι ένα από τα σημαντικότερα μέρη της εφαρμογής και για την εκτέλεση αυτή απαραίτητη προϋπόθεση είναι η ύπαρξη ενός εξυπηρετητή παγκόσμιου ιστού. Πιο συγκεκριμένα, για αυτό το σύστημα χρησιμοποιούμε τον εξυπηρετητή Apache Server μέσω του οποίου γίνονται αιτήματα http.



Σχήμα 5.4: Διάγραμμα Apache Web Server

Στην παρούσα εφαρμογή χρησιμοποιήθηκε η τεχνολογία MySQL για την σχεδίαση της βάσης δεδομένων καθώς και την ανάπτυξη ενός διαγράμματος Οντοτήτων-Συσχετίσεων ενώ η εγκατάσταση πραγματοποιήθηκε μέσω της διεπαφής phpMyAdmin.

5.5 Βιβλιοθήκες

Για την ανάπτυξη της εφαρμογής, εκτός από τις παραπάνω τεχνολογίες χρειάστηκαν επιπλέον και οι απαραίτητες βιβλιοθήκες που εφαρμόστηκαν με τις τεχνολογίες αυτές. Παρακάτω, θα αναφερθούμε σε κάποιες από τις σημαντικές βιβλιοθήκες των οποίων έγινε η χρήση εκτός από αυτών που έχουν αναφερθεί ήδη(π.χ. React Native,Laravel κ.λπ.).

5.5.1 Vector Icons

Πρόκειται για βιβλιοθήκη η οποία δίνει στον χρήστη την δυνατότητα να χρησιμοποιήσει εικονίδια, logo και κουμπιά για να μπορέσει να μορφοποιηθεί καλύτερα το περιβάλλον στο οποίο δουλεύει και βλέπει.

5.5.2 React Native Router Flux

Ουσιαστικά πρόκειται για βιβλιοθήκη που δίνει στον χρήστη την δυνατότητα να πλοηγηθεί μεταξύ διαφόρων οθονών της εφαρμογής. Ως τη βασική βιβλιοθήκη της React Native επιτρέπει στον χρήστη να ορίσει όλες τις διαδρομές-routes σε μια κεντρική οθόνη ώστε να υπάρχει εύκολη και άνετη μεταφορά και αλληλεπίδραση δεδομένων μεταξύ των διάφορων οθονών.

5.5.3 Google Maps API

Πρόκειται για μια από τις σημαντικότερες API βιβλιοθήκες που χρησιμοποιήθηκαν στην εφαρμογής μας καθώς δίνει την δυνατότητα στον χρήστη να έχει πρόσβαση στους χάρτες τις GOOGLE και όλων των δυνατοτήτων που τους προσφέρει όπως οι τοποθεσίες για τα πάρκα για σκυλιά,καφετέριες φιλικά προς τους σκύλους, κτηνιατρεία και PetShops αλλά και όλες τις πληροφορίες για αυτές όπως τα ονόματα, τις τοποθεσίας, φωτογραφίες -εάν υπάρχουν- και την βαθμολογία που τους έχει δοθεί από τους χρήστες της google maps. Επιπλέον, ο χάρτης παρέχει δυνατότητες για διαφορετική προβολή μορφοποίησης εδάφους, street view, μεγέθυνση-σμίκρυνση κ.λ.π.

5.5.4 GeoLib

Πρόκειται για βιβλιοθήκη η οποία δίνει στον χρήστη την δυνατότητα να υπολογίσει την γεωγραφική απόσταση μεταξύ δυο σημείων όπως για παράδειγμα την απόσταση του χρήστη από το συγκεκριμένο σημείο που βρίσκεται.

5.5.5 React Native Star Rating

Πρόκειται για μια βιβλιοθήκη που δίνει στον χρήστη την δυνατότητα βαθμολόγησης και προβολή βαθμολογίας μέσω αστεριών.

5.6 Plugins

5.6.1 Map View Directions

Πρόκειται για τη λειτουργία που παρέχει τη δυνατότητα στον χρήστη να πλοηγηθεί στην συγκεκριμένη τοποθεσία που διάλεξε, καθώς το plugin δέχεται την είσοδο δυο τοποθεσιών και υπολογίζει την βέλτιστη διαδρομή είτε με τα ποδιά είτε με αμάξι.

5.6.2 Geolocation

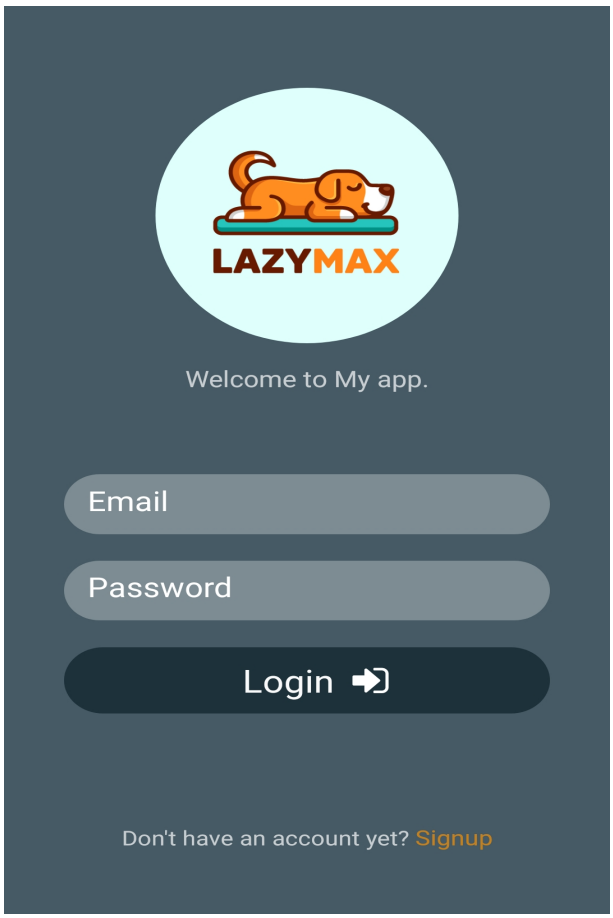
Επίσης απαραίτητο plugin για την εφαρμογή μας, διότι παρέχει την δυνατότητα εύρεσης της τοποθεσίας του χρήστη στον χάρτη ώστε να βρει τις ανάλογες τοποθεσίες που έχει αναζητήσει γύρω από την τοποθεσία του.

5.6.3 Dimensions

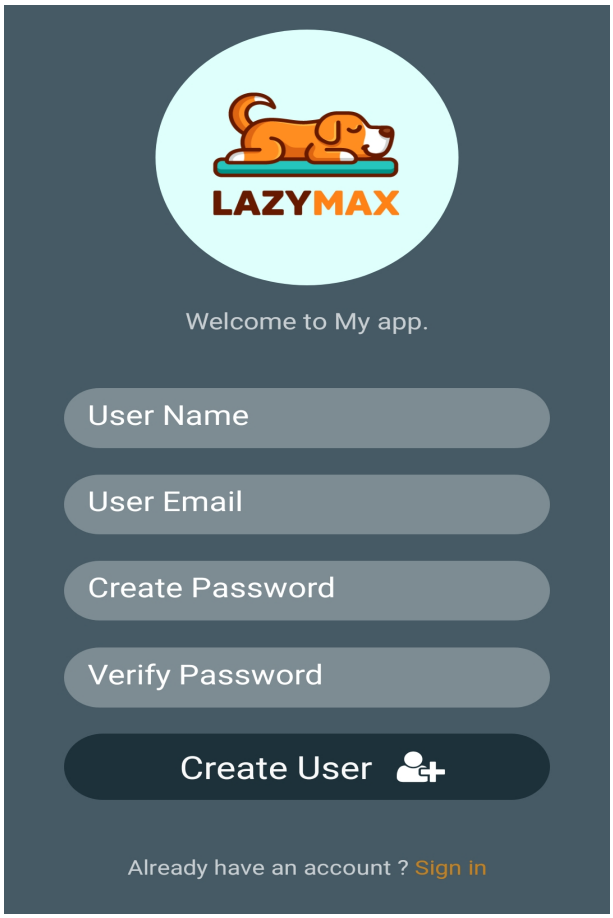
Πρόκειται για το plugin που υπολογίζει τις διαστάσεις της οθόνης της κινητής συσκευής που χρησιμοποιεί ένας χρήστης και προσαρμόζει τις οθόνες της εφαρμογής ανάλογα, πράγμα που καθίσταται αναγκαίο καθώς η εφαρμογή θα χρησιμοποιηθεί από χρήστες με διαφορετικές κινητές συσκευές.

5.7 ScreenShots Εφαρμογής

Τα στιγμιότυπα των οθονών της εφαρμογής παρουσιάζονται παρακάτω.

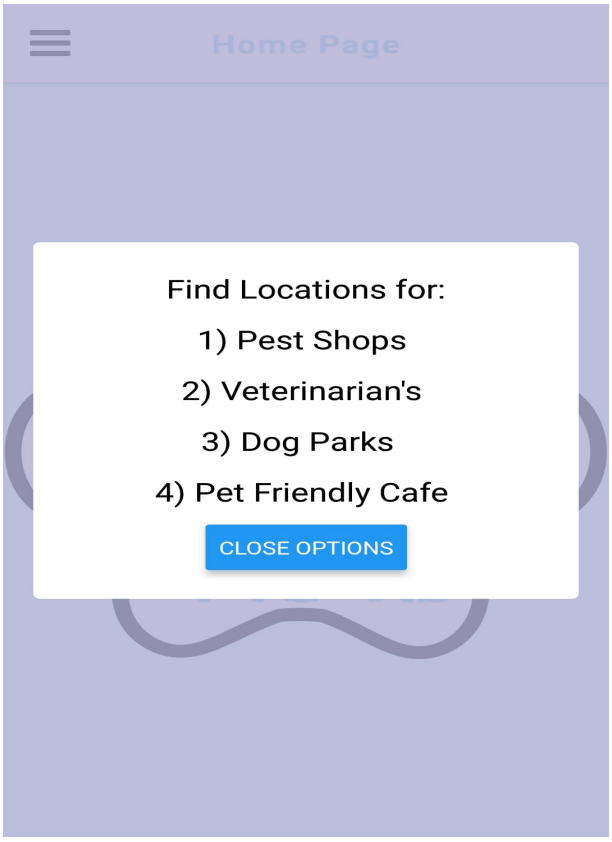


Σχήμα 5.5: Φόρμα Σύνδεσης

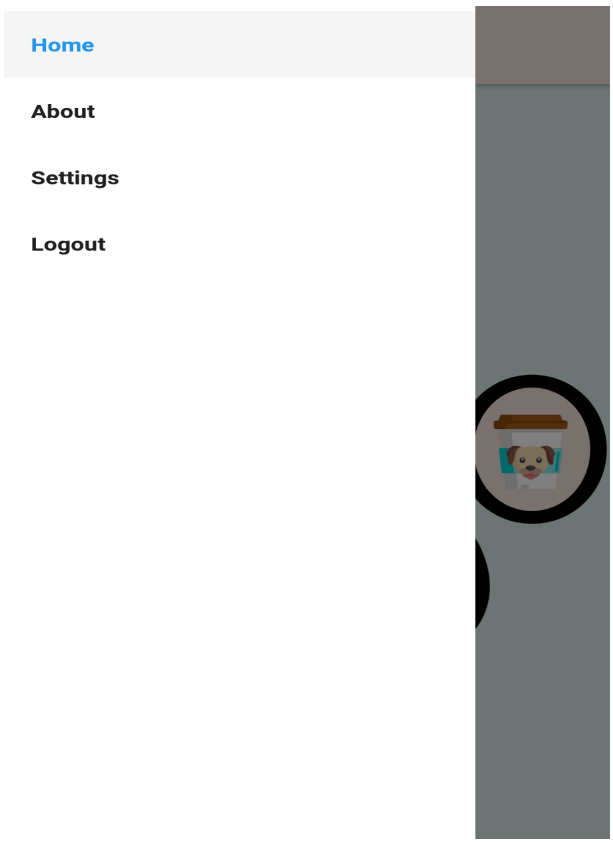


Σχήμα 5.6: Φόρμα Εγγραφής

Σχήμα 5.7: Home Page

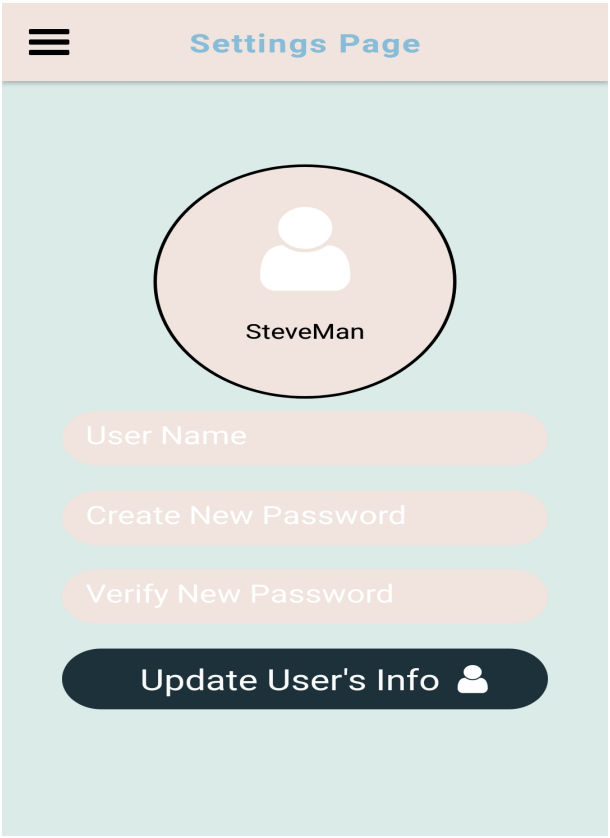


Σχήμα 5.8: Παράθυρο εμφάνισης επιλογών

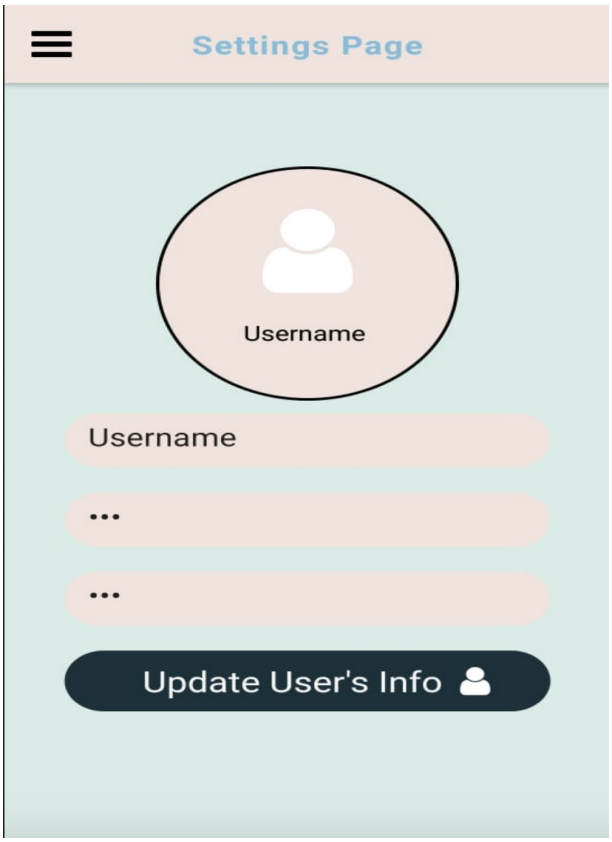


Σχήμα 5.9: Εμφάνιση άλλων οθονών

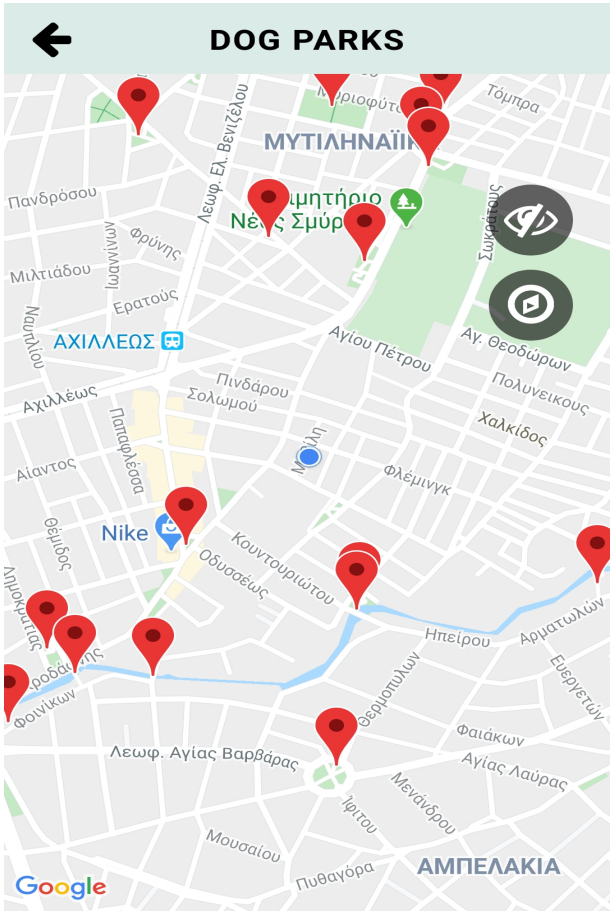
Σχήμα 5.10: About Page



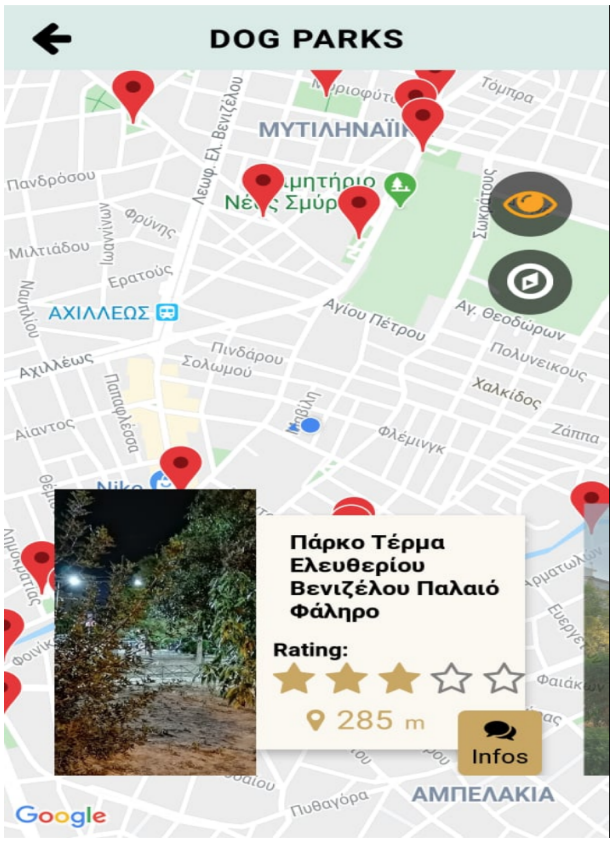
Σχήμα 5.11: Settings Page



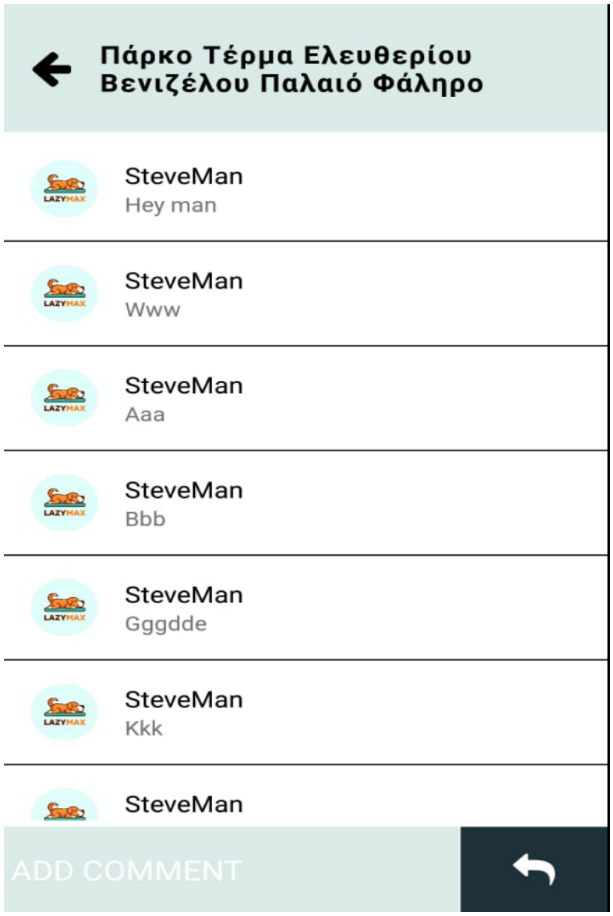
Σχήμα 5.12: Changing User's Settings



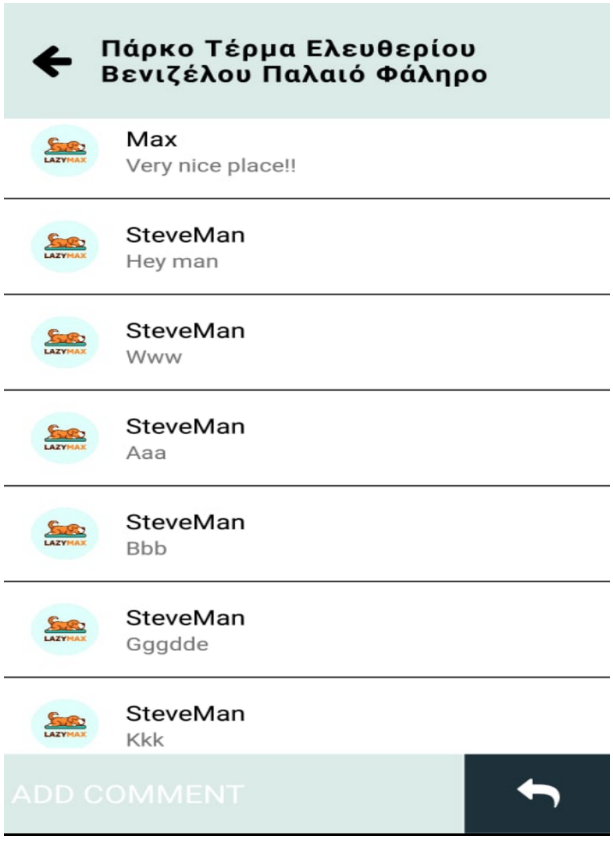
Σχήμα 5.13: Επιλογή Dog Parks, είναι ίδιο για κάθε επιλογή



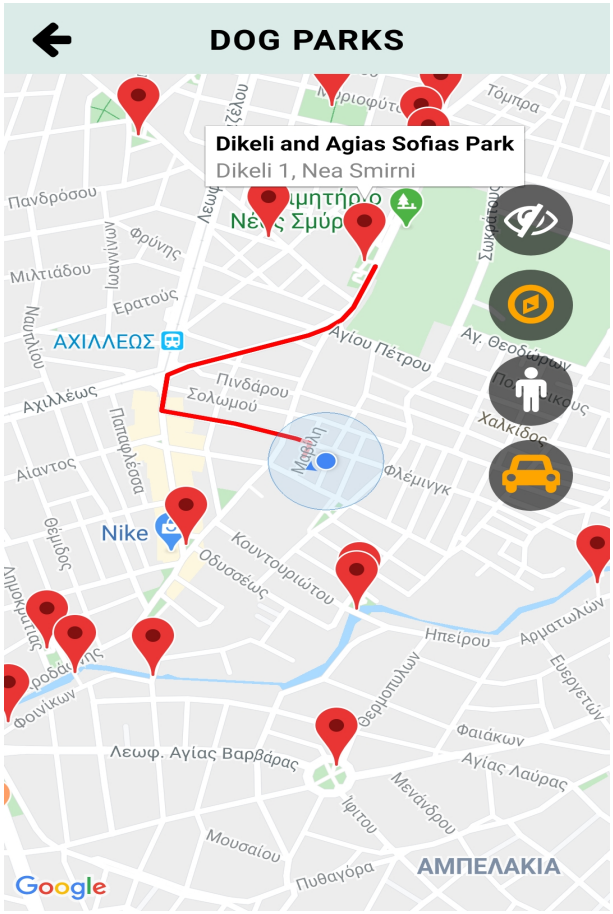
Σχήμα 5.14: Carousel πληροφοριών για τις τοποθεσίες



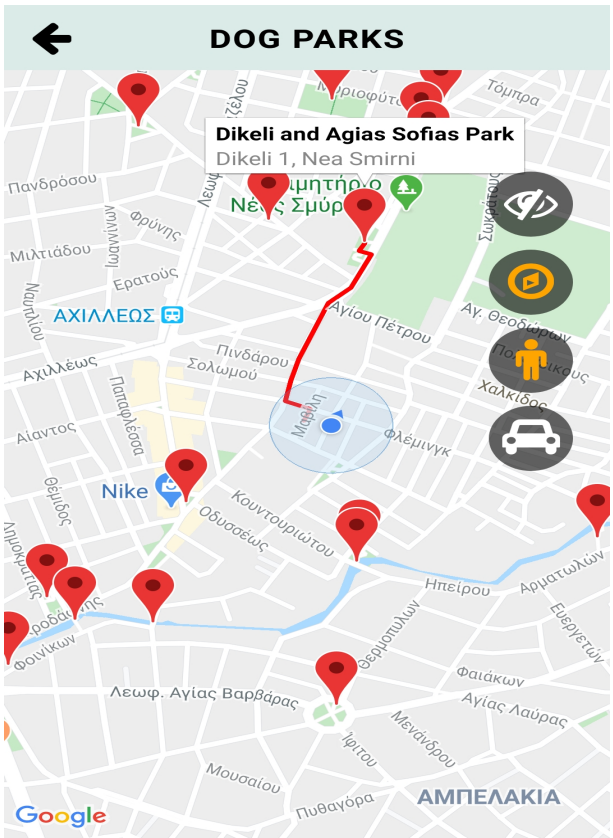
Σχήμα 5.15: Comment Page



Σχήμα 5.16: Εισαγωγή σχόλιου στο Comment Page



Σχήμα 5.17: Επιλογή πλοήγησης σε μια τοποθεσία με αμάξι



Σχήμα 5.18: Επιλογή πλοήγησης σε μια τοποθεσία με τα πόδια

6. ΑΞΙΟΛΟΓΗΣΗ

Σε αυτό το κεφάλαιο γίνεται πλήρης ανάλυση της αξιολόγησης του συστήματος. Ουσιαστικά, σκοπός της αξιολόγησης είναι να να γίνει αντιληπτό το πόσο εύχρηστη είναι η εφαρμογή και να προκύψουν συμπεράσματα, μέσω των οποίων θα μπορέσουμε να κάνουμε τις απαραίτητες τελικές διορθώσεις.

Η ευχρηστία του συστήματος είναι συσχετισμένη με την ευκολία χρήσης και εκμάθησης του, καθώς και με το επίπεδο βοήθειας που παρέχει στους χρήστες κατά την διάρκεια εκτέλεσης του (Dix et al. 2003).

Για να αξιολογήσουμε λοιπόν την ευχρηστία μιας εφαρμογής, μπορούμε να κάνουμε χρήση ποικίλων μεθοδολογιών, εκ των οποίων αρκετά σημαντικές για αναφορά είναι τρεις :

- Usability inspection methods: Πρόκειται για μεθόδους κατά τις οποίες εξειδικευμένα άτομα σε θέματα αλληλεπίδρασης ανθρώπου μηχανής κάνουν επιθεώρηση στο σύστημα-εφαρμογή.
- User testing methods: Αυτή η κατηγορία αναφέρεται σε μεθόδους που πραγματοποιούνται από τους ίδιους τους χρήστες στου οποίους στοχεύει η εφαρμογή.
- Analytic methods: Πρόκειται για τις μεθόδους αναλυτικής αξιολόγησης (Αβούρης, 2000).

6.1 Τεχνικές αξιολόγησης

Σύμφωνα με αυτά που αναφέρθηκαν παραπάνω, τρεις είναι οι βασικές τεχνικές-μέθοδοι αξιολόγησης:

1. Αξιολόγηση από ειδικούς:Βασικό σημείο της κατηγορίας αυτής είναι ότι γίνεται από έμπειρους-ειδικούς αξιολογητές. Με άλλα λόγια είναι άτομα που έχουν τις απαραίτητες γνώσεις πάνω στην αξιολόγηση της ευχρηστίας του συστήματος ή εφαρμογής. Το πλεονέκτημα της συγκεκριμένης αξιολόγησης είναι ότι έχει χαμηλό κόστος και ότι μπορεί να εφαρμοστεί ταυτόχρονα σε όλα τα στάδια σχεδιασμού και ανάπτυξης ενός συστήματος ή μιας εφαρμογής. Κάποιες από τις διαδεδομένες μεθόδους της τεχνικής αυτής είναι το Γνωσιακό Περιδιάβασμα (αξιολόγηση προσομοίωσης ενεργειών ενός χρήστη πάνω σε ένα σύστημα, Lewis) και η Ευρετική Αξιολόγηση (αξιολόγηση βασισμένη στους ευρετικούς κανόνες των Nielsen και Molich [14]).

2. Αξιολόγηση από τελικούς χρήστες:Πρόκειται για τεχνική-μέθοδο αξιολόγησης που στηρίζεται στην αλληλεπίδραση των χρηστών,για του οποίους αναπτύχθηκε και υλοποιήθηκε η εφαρμογή, με το σύστημα. Για την κατηγορία αυτή, μία από τις πιο διαδεδομένες μεθόδους είναι η Μονάδα Μέτρησης Απόδοσης. Κατά τη διαδικασία αυτή, δίνονται στον κάθε χρήστη ένα σύνολο ενεργειών και λειτουργιών με τη βοήθεια των οποίων θα εντοπιστούν τυχόν σφάλματα ενώ ταυτόχρονα θα καταγραφούν οι χρόνοι εκτέλεσης,ο αριθμός ενεργειών κ.λπ. Έπειτα θα πραγματοποιηθεί εξέταση για τη βελτίωση της ευχρηστίας της εφαρμογής. Επιπλέον, σημαντική μέθοδος είναι το Πρωτόκολλο Ομιλούντος Υποκειμένου, κατά την οποία ο χρήστης εξωτερικεύει τις σκέψεις του και τις απόψεις του καθώς αλληλεπιδρά με την εφαρμογή ενώ ταυτόχρονα καταγράφονται. Τέλος, μια ακόμη μέθοδος είναι αυτή των Ερωτηματολογίων. Οι χρήστες της εφαρμογής καλούνται να απαντήσουν τα ερωτηματολόγια ώστε να βελτιωθεί μελλοντικά το σύστημα για τυχόν λάθη.

3. Αναλυτικές μέθοδοι αξιολόγησης: Σε αυτή την κατηγορία γίνεται αναφορά σε μοντέλα-τεχνικές αξιολόγησης που προκύπτουν από Προβλεπτικά Μοντέλα, τα οποία βασίζονται στο μοντέλο

ανθρώπινου επεξεργαστή (Human Information Processing),το μοντέλο πληκτρολογήσεων KLM (Keystroke Level Model) ή στο μοντέλο ανθρώπινου επεξεργαστή (Human Information Processing). Επίσης, υπάρχουν και τα Εμπειρικά Προβλεπτικά Μοντέλα, όπως είναι μεταξύ άλλων ο νόμος του Fitts, τα οποία περιγράφουν και καταγράφουν την συμπεριφορά των ανθρώπων. Τέλος, το πλεονέκτημα αυτών των μεθόδων είναι το χαμηλό τους κόστος και η σημαντική ακρίβεια που προκύπτει από τη μελέτη τους.

6.2 Μεθοδολογία αξιολόγησης

Λόγω της ολοένα και αυξανόμενης αλληλεπίδρασης μεταξύ ανθρώπου και μηχανής, οι διαδραστικές τεχνολογίες καθώς και το διαδίκτυο κατέχουν όλο και πιο σημαντικό ρόλο στη ζωή του ανθρώπου. Γι' αυτό τον λόγο ένας ικανός μηχανικός σχεδιασμού διαδραστικών συστημάτων οφείλει να αξιολογεί συνεχώς αυτά τα συστήματα και να πραγματοποιεί αλλαγές και βελτιώσεις όποτε κρίνεται αναγκαίο. Βασικοί στόχοι των εφαρμογών αξιολόγησης είναι η επίτευξη ευχρηστίας και η δυνατότητας ενός προϊόντος να χρησιμοποιείται από καθορισμένους χρήστες με καθορισμένους στόχους, υπό καθορισμένες συνθήκες χρήσης ώστε να είναι αποτελεσματικό (effectiveness), αποδοτικό (efficiency) και να παρέχει υποκειμενική ικανοποίηση (satisfaction) στους χρήστες του.

Τρεις είναι οι βασικές κατηγορίες σύμφωνα με τις οποίες διαχωρίζονται οι τρόποι αξιολόγησης: οι αναλυτικές, οι πειραματικές και οι διερευνητικές μέθοδοι. Για την πληρέστερη υλοποίηση της αξιολόγησης κρίθηκε απαραίτητο να χρησιμοποιηθεί και να παρουσιαστεί μία μέθοδος αξιολόγησης από τις κατηγορίες αναλυτικές και πειραματικές. Γι' αυτό τον λόγο αποφασίστηκε από την κατηγορία των αναλυτικών μεθόδων να γίνει χρήση της Ευρετικής Αξιολόγησης και από των πειραματικών μεθόδων, το Πρωτόκολλο Ερωτήσεων.

6.2.1 Μέθοδος Ευρετικής Αξιολόγησης(Heuristic Evaluation)

Η Ευρετική Αξιολόγηση αναπτύχθηκε αρχικά από τον Δανό μηχανικό Jakob Nielsen ο οποίος βασίστηκε στην εκτεταμένη εμπειρία του ως καθηγητής και σύμβουλος ευχρηστίας. Αποτελεί μία εξαιρετικά διαδεδομένη αναλυτική μέθοδος αξιολόγησης που βασίζεται στην εφαρμογή ενός μικρού αριθμού γνωστών κανόνων σχεδιασμού διαδραστικών συστημάτων. Θεωρείται ως μία υποκειμενική μέθοδος που διεκπεραιώνεται από πεπειραμένους ειδικούς μηχανικούς ευχρηστίας, οι οποίοι δεν έχουν εμπλακεί στην ανάπτυξη του αξιολογούμενου συστήματος. Ειδικότερα, η μέθοδος αυτή ορίζεται ως «αξιολόγηση ευχρηστίας χαμηλού κόστους» (discount usability engineering) και διεκπεραιώνεται συνήθως από τρεις έως πέντε επιθεωρητές ευχρηστίας που έχουν σαν αποστολή να εξετάσουν ξεχωριστά για μία έως δύο περίπου ώρες το σύστημα, εάν αυτό θεωρείται απλό, και αφού συγκρίνουν τα προβλήματα που βρήκαν, να καταλήξουν σε μία λίστα.

6.2.2 Πλεονεκτήματα-Μειονεκτήματα

Προφανώς, η Ευρετική Αξιολόγηση διαθέτει έναν αριθμό πλεονεκτημάτων και μειονεκτημάτων. Αρχής γενομένης από τα θετικά:

- Διαθέτει λίγες οδηγίες που όμως συλλαμβάνουν πολλά κοινά προβλήματα ευχρηστίας.
- Οι κανόνες είναι εύκολο να απομνημονευτούν και η διαδικασία εφαρμογής τους θεωρείται μέτριας δυσκολίας.
- Δεν απαιτεί εμπλοκή των τελικών χρηστών παρά μόνον των ειδικών.
- Όπως προαναφέρθηκε παραπάνω αποτελεί έναν φθηνό και εύκολο τρόπο για την επιθεώρηση ενός συστήματος.

Όσον αφορά τα αρνητικά:

- Οι κανόνες δεν μπορούν να εξεταστούν ως μία απλή λίστα ελέγχων.
- Υπάρχουν ασάφειες κατά την χρήση της.

6.2.3 Ερωτήματα Αξιολόγησης

Για τη μέθοδο της Ευρετικής Αξιολόγησης με βάση τους κανόνες ευχρηστίας των Molich και Nielsen, δόθηκαν προς μελέτη στους αξιολογητές ώστε να υπάρξει η βέλτιστη αξιολόγηση χωρίς την ύπαρξη παρερμηνείας. Παρακάτω, παρουσιάζεται πίνακας με τους 10 κανόνες των Molich και Nielsen.

Πίνακας 6.1: Πίνακας με τους 10 κανόνες των Molich και Nielsen

Κανόνας	Σύντομη Περιγραφή
1.Ορατότητα της κατάστασης του συστήματος (Visibility of system status).	Το σύστημα θα πρέπει να κρατά πάντα τους χρήστες ενήμερους για το τι συμβαίνει κάθε χρονική στιγμή. Αυτό επιτυγχάνεται με κατάλληλη μορφή ανάδρασης και ανατροφοδότησης μέσα σε εύλογο χρονικό διάστημα.
2.Ταίριασμα μεταξύ συστήματος και πραγματικού κόσμου (Match between system and thereal world).	Το σύστημα θα πρέπει να μιλάει τη γλώσσα των χρηστών, με τις φράσεις, τις έννοιες και τις λέξεις οι οποίες είναι γνώστες και οικίες για τους χρήστες και όχι με όρους συστήματος.
3.Έλεγχος από τον χρήστη και ελευθερία (User control and freedom).	Ουσιαστικά θα πρέπει να μπορεί ο χρήστης να έχει την δυνατότητα ακυρώσεις ενεργειών καθώς την ανάκλαση του ή την επανάληψη τους με ευκολία.
4.Συνέπεια και πρότυπα (Consistency and standards).	Οι χρήστες δεν θα πρέπει να απορούν ή να μπερδεύονται για το εάν διαφορετικές λέξεις, καταστάσεις ή δράσεις σημαίνουν το ίδιο πράγμα.
5.Πρόληψη σφάλματος (Error prevention)	Εκτός από τα μηνύματα ενημέρωσης σε περίπτωση σφάλματος, αρκετά σημαντικό είναι ένα σύστημα να είναι καλά σχεδιασμένο έτσι ώστε να μην προκύψουν σφάλματα και να μην χρειαστούν τα μηνύματα.

6.Αναγνώριση αντί ανάκλησης (Recognition rather than recall).	Οι οδηγίες σχετικά με τη χρήση του συστήματος δεν θα έπρεπε να τις απομνημονεύσει ο χρήστης αλλά θα πρέπει να είναι ανακτήσιμες εύκολα, όποτε ο χρήστης τις χρειάζεται.
7.Ευελιξία και Αποτελεσματικότητα χρήσης (Flexibility and efficiency of use).	Το σύστημα πρέπει να προσαρμόζεται στην εμπειρία των χρηστών είτε είναι έμπειρος είτε είναι άπειρος και να επιτρέπει στους χρήστες να προσαρμόζουν τις ενέργειες ώστε να είναι εύχρηστο για όλους.
8.Αισθητική και μινιμαλιστική σχεδίαση (Aesthetic and minimalist design).	Οποιαδήποτε πρόσθετη πληροφορία μπορεί να οδηγήσει σε λανθασμένη κατανόηση από τους χρήστες ακόμα και να του μπερδέψει.
9.Βοήθεια στους χρήστες για την αναγνώριση, διάγνωση και επαναφορά από σφάλματα (Help users recognize, diagnose and recover from errors).	Σε περίπτωση σφάλματος, το μήνυμα που θα προβάλετε στον χρήστη θα πρέπει να είναι σε κατανοητή γλώσσα να περιγράφεται με ακρίβεια και η λύση να προτείνεται με εποικοδομητικό τρόπο έτσι ώστε να το κατανόηση πλήρης ο χρήστης και να μην υπάρχει περιθώριο αμφισβήτησης.
10.Βοήθεια και τεκμηρίωση (Help and documentation).	Όλες οι πληροφορίες τεκμηρίωσης θα πρέπει όχι μόνο να είναι εύκολο να αναζητηθούν και είναι προσβάσιμες, αλλά και να έχουν ως σκοπό να μην έχει την ανάγκη ο χρήστης να τις αναζητήσει

6.2.4 Ενημέρωση Αξιολογητών

Πριν αρχίσει η διαδικασία της αξιολόγησης, οι αξιολογητές προμηθεύονται για μελέτη τους κανόνες ευχρηστίας. Έπειτα τους αναλύεται η χρήση της εφαρμογής, όπως επίσης και οι απαραίτητες πληροφορίες που θα χρειαστούν. Εφόσον γίνει κατανοητός ο σκοπός της εφαρμογής, οι υπηρεσίες της και οι λειτουργίες που προσφέρει, ακολουθεί μια αναπαράσταση τυπικής χρήσης της εφαρμογής για την καλύτερη κατανόηση. Τέλος, αφού δοθούν στιγμιότυπα του προτύπου της εφαρμογής για να υπάρξει καλύτερη γενική εικόνα στον αναγνώστη της εργασίας, θα συζητηθούν οι μελλοντικές επεκτάσεις τής και οι τρόποι βελτίωσή της καθώς σε αυτή τη φάση της διπλωματικής εργασίας αποτελεί μόνο ένα demo.

6.2.5 Διερεύνηση εφαρμογής

Κατά τη διαδικασία της αξιολόγησης έγινε η χρήση της κινητής συσκευής SAMSUNG GALAXY S7 πάνω στην οποία έτρεχε η εφαρμογή. Στη συνέχεια, ο κάθε αξιολογητής είχε αρκετό χρόνο να περιηγηθεί στις οθόνες της εφαρμογής και στις λειτουργίες της, χωρίς την καθοδήγηση κάποιου ειδικού και την χρήση βοηθημάτων. Επιπρόσθετα, για την καλύτερη κατανόηση της εφαρμογής και την βέλτιστη αξιολόγηση της από τους χρήστες, δόθηκαν 10 σενάρια αντιπροσωπευτικά ως προς

την χρήση της εφαρμογής έτσι ώστε να μπορέσει να πλοηγηθεί καλύτερα και να εξετάσει όλες τις λειτουργίες της ή και να εντοπίσει τυχόν σφάλματα. Τα 10 σενάρια που δόθηκαν στον χρήστη με την σειρά είναι τα εξής :

1. Κάντε εγγραφή στο σύστημα.
2. Συνδεθείτε στο σύστημα με τα προσωπικά στοιχεία που χρησιμοποιήσατε στην εγγραφή.
3. Επιλέξτε οποιοδήποτε κουμπί αναζήτησης τοποθεσιών που επιθυμείτε.
4. Επιλέξτε την επιλογή View και κάντε swipe left στο carousel για να ερευνήσετε τις τοποθεσίες με τις πληροφορίες τους.
5. Επιλέξτε την επιλογή info σε μια τοποθεσία.
6. Εισάγετε κάποιο σχόλιο ή πολλά.
7. Διαγράψτε κάποιο σχόλιο σας.
8. Επιστρέψτε πίσω στον χάρτη και επιλέξτε την επιλογή Directions καθώς και την τοποθεσία που θέλετε να να πάτε είτε με αμάξι είτε με τα πόδια.
9. Γυρίστε πίσω στην Home Page και επιλέξτε την μπάρα επιλογών πάνω αριστερή γωνία.
10. Επιλέξτε την About Page για να ενημερωθείτε για την εφαρμογή.
11. Επιλέξτε μετά την Setting Page.
12. Κάνε update κάποια από τα προσωπικά σου δεδομένα.
13. Κάνε Log Out από την εφαρμογή.

Τέλος, αφού οι αξιολογητές εφαρμόσουν τα παραπάνω σενάρια, θα συνεχίσουν με την αξιολόγηση με βάση τους κανόνες ευχρηστίας των Molich και Nielsen και βαθμολογία κλίμακας Likert [15] από το 1 έως το 7.

Πίνακας 6.2: Κλίμακα Αθροιστικής Βαθμολόγησης – Likert scale

Διαφωνώ απόλυτα	Διαφωνώ αρκετά	Διαφωνώ	Ούτε διαφωνώ Ούτε συμφωνώ	Συμφωνώ	Συμφωνώ αρκετά	Συμφωνώ απόλυτα
1	2	3	4	5	6	7

Πίνακας 6.3: Κλίμακα Αθροιστικής Βαθμολόγησης – Likert scale

Ερωτήματα	Αξιολογητής 1	Αξιολογητής 2	Αξιολογητής 3	Αξιολογητής 4	Αξιολογητής 5	Αποτελέσματα
Το σύστημα ενημερώνει συχνά τον χρήστη για τις σχετικές μεταβολές;	5	4	5	4	5	Θετικό αποτέλεσμα: Διατίθενται συνέχεια ανανεωμένες τοποθεσίες,σχόλια και πληροφορίες για τον χρήστη.
Το σύστημα χρησιμοποιεί κατάλληλη γλώσσα,λέξεις και σύμβολα οικεία στον χρήστη;	6	7	7	6	7	Θετικό αποτέλεσμα: Είναι πλήρως κατανοητό στους χρήστες ορολογίες και τα σύμβολα που χρησιμοποιεί η εφαρμογή.
Το σύστημα παρέχει την δυνατότητα ακύρωσης ενεργειών;	7	6	7	7	7	Θετικό αποτέλεσμα: Για κάθε ενέργεια του χρήστη του δίνεται η επιλογή να την ακυρώσει και να γυρίσει στην προηγούμενη σελίδα.
Το σύστημα παρέχει συνέπεια στην χρήση ορολογίας, σημασιολογία συμβόλων κ.λπ.;	5	6	6	5	5	Θετικό αποτέλεσμα: Οι χρήστες αντιλαμβάνονται την σημασία των συμβόλων στους χάρτες.

Το σύστημα προστατεύει τον χρήστη από πιθανά σφάλματα;	4	3	3	2	3	Αρνητικό αποτέλεσμα: Ο χρήστης δεν ενημερώνεται για την απαραίτητη ύπαρξη διαδικτύου για να μπορέσουν να χρησιμοποιήσουν την εφαρμογή.
Το σύστημα κάνει διάκριση μεταξύ αρχάριων και έμπειρων χρηστών;	5	5	5	4	5	Θετικό αποτέλεσμα: Καθώς οι χρήστες είχαν όλες τι πληροφορίες που είναι απαραίτητες δεν κρίνεται η απαραίτητη διάκριση.
Το σύστημα βοηθάει στην ελαχιστοποίηση του μνημονικού φορτίου του χρήστη;	6	6	7	5	5	Θετικό αποτέλεσμα: Το σύστημα ενημερώνει τον χρήστη σε πια σελίδα βρίσκεται και πιο κουμπί έχει επιλέξει.
Το σύστημα παρέχει καλαισθησία και ευχρηστία από την οργάνωση του περιβάλλοντος;	6	7	7	6	5	Θετικό αποτέλεσμα: Το περιβάλλον της εφαρμογής είναι πλήρως ικανοποιητικό για τους χρήστες.
Το σύστημα διατυπώνει σωστά τα σφάλματα για να τα κατανοήσει ο χρήστης;	5	6	6	5	6	Θετικό αποτέλεσμα: Το σύστημα ενημερώνει τον χρήστη μέσω μηνύματος για την περίπτωση σφάλματος ώστε να το διορθώσει.

Το σύστημα παρέχει αρκετά εγχειρίδια χρήσης εστιασμένα στις ανάγκες του χρήστη;	2	2	1	2	1	Αρνητικό αποτέλεσμα: Δεν παρέχονται στον χρήστη κατάλληλα εγχειρίδια χρήσης για εξάσκηση.

6.2.6 Ανάλυση αποτελεσμάτων ευρετικής αξιολόγησης(Heuristic Evaluation)

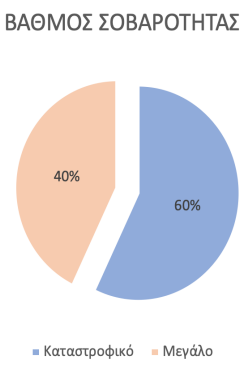
Στην υποενότητα αυτή θα γίνει ανάλυση των αποτελεσμάτων που προέκυψαν από την μέθοδο της Ευρετικής Αξιολόγησης. Κατά την διαδικασία της αξιολόγησης οι χρήστες κατέγραψαν τα σφάλματα και τα προβλήματα που συνάντησαν και τα αντιστοιχούσαν με τον συγκεκριμένο κανόνα που παραβιαζόταν σε κάθε περίπτωση.

Η κλίμακα σοβαρότητας που χρησιμοποιήθηκε ήταν η εξής :

0. Δεν είναι πρόβλημα: δεν συμφωνώ ότι είναι πρόβλημα ευχρηστίας.
1. Δευτερεύων: το διορθώνουμε αν υπάρχει διαθέσιμος χρόνος.
2. Μέτριο: το να το διορθώσουμε έχει χαμηλή προτεραιότητα.
3. Σοβαρό: το να το διορθώσουμε έχει υψηλή προτεραιότητα.
4. Καταστροφικό πρόβλημα: πρέπει να διορθωθεί οπωσδήποτε.

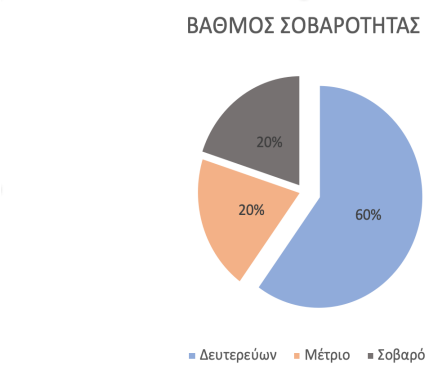
Πίνακας 6.4: Παρουσίαση Προβλημάτων

Πρόληψη σφάλματος (Error prevention).
Δεν υπάρχει η απαραίτητη ενημέρωση των χρηστών για την ύπαρξη δικτύου ώστε να λειτουργήσει η εφαρμογή.



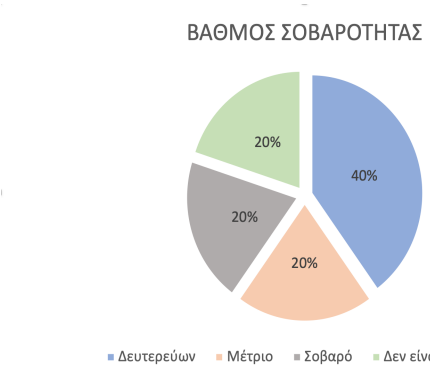
Αποτελεσματικότητα χρήσης και ευελιξία.

Οι χρήστες δεν ενημερώνονται από την εφαρμογή για την ύπαρξη νέων τοποθεσιών παρά μόνο όταν κάνουν οι ίδιοι την αναζήτηση.



Αποτελεσματικότητα χρήσης και ευελιξία.

Οι χρήστες δεν ενημερώνονται από την εφαρμογή για την ύπαρξη άλλων σκυλιών ή της πολυκοσμίας σε ορισμένες τοποθεσίες έτσι ώστε ο χρήστης να επιλέξει ανάλογα άμα θέλει να πάει ή όχι.



Για τα προβλήματα που αναφέρθηκαν παραπάνω συζητήθηκαν τρόποι και λύσεις, όπως για παράδειγμα:

- Πρόληψη σφάλματος: Εμφάνιση κατάλληλων μηνυμάτων στον χρήστη για να τον ενημερώσει ότι δεν είναι συνδεδεμένος στο δίκτυο και είτε να συνδεθεί είτε να ενεργοποίηση τα δεδομένα του.
- Αποτελεσματικότητα χρήσης και ευελιξία:
  1. Ενημέρωση μέσω notification στην οθόνη του κινητού του χρήστη ότι έχει δημιουργηθεί καινούργια τοποθεσία στην περιοχή που βρίσκεται, για μια από τις αναζητήσεις/λειτουργίες που διαθέτει η εφαρμογή.
  2. Να εμφανίζεται στον χάρτη με πράσινες βούλες ο πληθυσμός που βρίσκεται σε μια τοποθεσία ώστε να ενημερώνει τον χρήστη για την πολυκοσμία της περιοχής.

Ένα επιπλέον πρόβλημα που συζητήθηκε ήταν ότι δεν υπήρχε η εναλλαγή γλώσσας από αγγλικά σε ελληνικά καθώς η εφαρμογή απευθύνετε και σε ελληνικό κοινό αλλά και σε κάποιες μεγάλες ηλικίες στους οποίους μπορεί να μην είναι τόσο οικεία τα αγγλικά. Αυτό το πρόβλημα, καταχωρήθηκε ως δευτερεύων από την στιγμή που η αγγλική ορολογία είναι αρκετά απλή και συνήθης αλλά και πολλές από τις τοποθεσίες και τα σχόλια είναι στα ελληνικά ή στην αντίστοιχη γλώσσα της χώρας με αποτέλεσμα να μην μπερδευτεί ο χρήστης.

6.2.7 Πρωτόκολλο Ερωτήσεων (Question Asking Protocol)

Η μέθοδος του πρωτοκόλλου ερωτήσεων ανήκει στην κατηγορία των πειραματικών αξιολογήσεων. Αποστολή της αξιολόγησης αυτής αποτελεί η κατανόηση του νοητικού επιπέδου του χρήστη, διότι αυτός αλληλεπιδρά με το σύστημα. Πρέπει να τονιστεί ότι απαιτείται η συμμετοχή τουλάχιστον τεσσάρων χρηστών ώστε να υπάρχουν έγκυρα αποτελέσματα και για να εντοπιστεί επαρκής αριθμός σφαλμάτων στο σύστημα. Παράλληλα υπάρχει ένας αξιολογητής που βρίσκεται δίπλα του κατά την διάρκεια της αλληλεπίδρασης του χρήστη με το εξεταζόμενο σύστημα, ο οποίος του θέτει ερωτήματα προς απάντηση και ηχογραφεί τη συνομιλία για μελλοντική ανάλυση και εξαγωγή συμπερασμάτων.

Κατά την εκτέλεσή της ο χρήστης χρησιμοποιεί την εφαρμογή ή την ιστοσελίδα για πρώτη φορά, καθοδηγείται από τον εξεταστή και οφείλει να απαντήσει ελεύθερα τις ερωτήσεις που του θέτονται. Επιπλέον μπορεί να εκφράζει την άποψή του για το σύστημα εάν θεωρήσει ότι χρειάζεται, ειδικά εάν διαπιστώσει κάτι που τον δυσαρεστεί ή κάτι που θα ήθελε να είναι διαφορετικό.

Από την πλευρά του ο εξεταστής μπορεί να δώσει την δυνατότητα στον χρήστη να περιηγηθεί ελεύθερα στο σύστημα για να εξοικειωθεί με το περιβάλλον. Ως αποτέλεσμα θα πρέπει να αντεπεξέλθει πιο άνετα στην διαδικασία. Οι ερωτήσεις που θέτει οφείλουν να είναι απλές και κατανοητές χωρίς τη χρήση σύνθετης ορολογίας. Επίσης οφείλει να εξάγει τις σημαντικότερες πληροφορίες και απαντήσεις από τον χρήστη. Τέλος απαιτείται να παρουσιάσει στον χρήστη κάποιους στόχους προς εκτέλεση και να παρακολουθήσει τις ενέργειές του θέτοντας επιπλέον ερωτήματα.

6.2.8 Εκτέλεση Πρωτοκόλλων Ερωτήσεων

Για τη διεξαγωγή της αξιολόγησης αυτής χρησιμοποιήθηκαν δέκα χρήστες-ιδιοκτήτες σκυλιών που είχαν εμπειρία στην ανατροφή σκυλιών αλλά και στη χρήση εφαρμογών, ενώ για την ηχογράφηση χρειάστηκε ένα κινητό τηλέφωνο. Πριν την εκκίνηση της διαδικασίας ρωτήθηκε από τον αξιολογητή εάν ο χρήστης επιθυμεί να ηχογραφηθεί και μόνο μετά την θετική του απάντηση ξεκίνησε η αξιολόγηση. Αφού έγινε κατανοητή η διαδικασία που θα ακολουθούσε, του έγινε γνωστό ότι σε οποιαδήποτε σημείο της διαδικασίας μπορούσε να εκφράσει τις σκέψεις του, τα συναισθήματα και την άποψή του.

Καταρχάς, δόθηκε χρόνος μερικών λεπτών για την ελεύθερη περιήγηση και εξοικείωσή με την εφαρμογή. Στην συνέχεια, υπήρξαν μερικές ερωτήσεις όπως οι παρακάτω:

1. Είναι εύκολο να διαπιστώσει κανείς το είδος της εφαρμογής ;
2. Υπάρχει συνέπεια όσον αφορά την εμφάνιση ή υπάρχουν αλλαγές σε κάποια σημεία και έλλειψη μοτίβου;
3. Είναι κατανοητή η ορολογία;
4. Μπορεί να καταλάβει με σχετική ευκολία σε ποιο σημείο της σελίδας βρίσκεται ανά πάσα στιγμή;
5. Υπάρχουν κάποια περιττές πληροφορίες που προκαλούν σύγχυση ή αποσπάνε την προσοχή;
6. Εάν παρέχονται από την εφαρμογή οδηγίες και βοήθεια για την περιήγηση και για την απάντηση απορiών-ερωτήσεων και εάν ναι μπορούν εντοπιστούν εύκολα;
7. Είναι κατανοητός ο τρόπος με τον οποίο είναι δομημένο το κεντρικό μενού;
8. Είναι κάτι που θα άλλαζαν, τροποποιούσαν ή να πρόσθεταν;

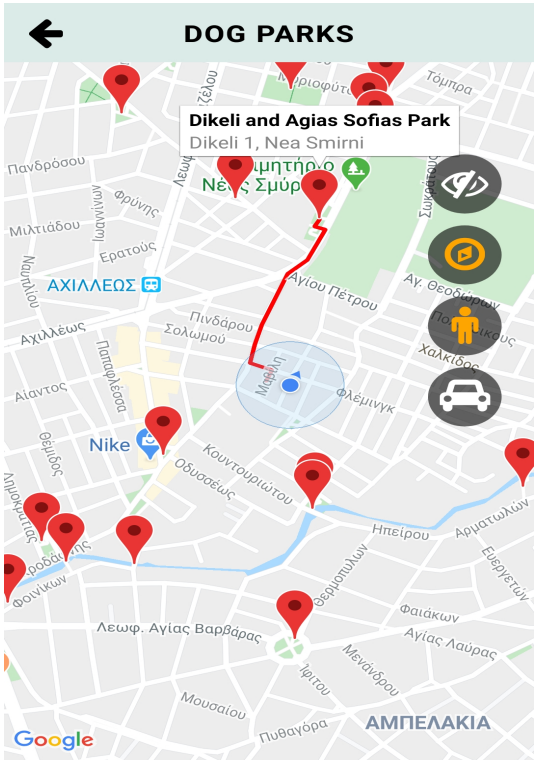
Στην συνέχεια ζητήθηκε η διεκπεραίωση ενός αριθμού τυπικών εργασιών (tasks) όπως εκείνα που εκτελέσαμε στην Ευρετική Αξιολόγηση παραπάνω.

6.2.9 Ανάλυση αποτελεσμάτων για πρωτόκολλο Ερωτήσεων (Question Asking Protocol)

Αφού συλλέχθηκαν οι απαντήσεις, το σύνολο των ξεχωριστών προβλημάτων που εντοπίστηκαν ήταν 7. Κάποια από τα προβλήματα που προέκυψαν ήταν ίδια με εκείνα της Ευρετικής Αξιολόγησης όπως το γεγονός ότι η εφαρμογή δεν διαθέτει εναλλαγή γλώσσας Αγγλικά-Ελληνικά, το οποίο επισημάνθηκε από όλους τους χρήστες, από την στιγμή που απευθύνετε όχι μόνο σε διεθνές αλλά και σε ελληνικό κοινό. Επίσης όπως και με την προηγούμενη αξιολόγηση εντοπίστηκε το πρόβλημα της απουσίας εγχειριδίου βοήθειας ή οδηγιών.

Συνεχίζοντας, εκτός από τις ομοιότητες των προβλημάτων με εκείνες τις Ευρετικής αξιολόγησης αξίζει να αναφερθούν κάποια σημαντικά προβλήματα που προέκυψαν χάρη στην μέθοδο Πρωτόκολλων ερωτήσεων. Όπως:

1. Ένας τυπικός χρήστης ανέφερε την άνεση που θα του δινόταν άμα εμφανιζόταν ο χρόνος της απόστασης που θα χρειαζόταν να διανύσει για να φτάσει στον προορισμό του.

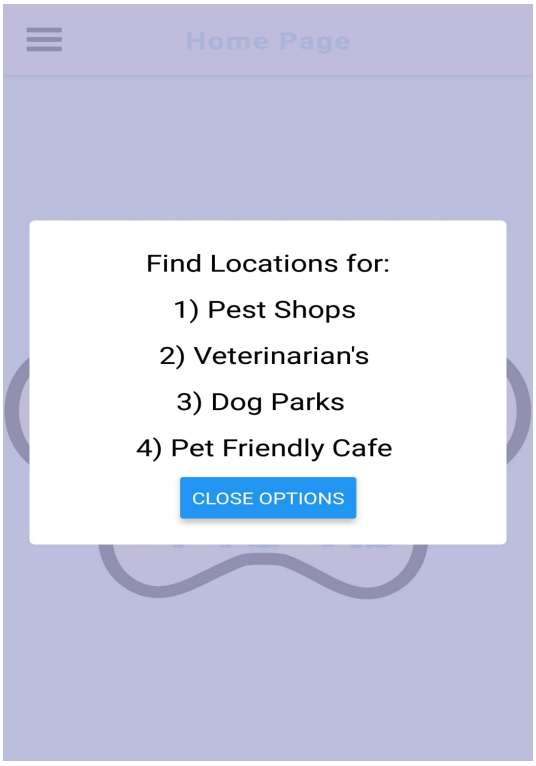


Σχήμα 6.1: Παράθυρο για βελτίωση Directions

2. Ζητήθηκε ένα pop up στο home page όταν επιλέξει το κουμπί options όπου θα εξηγούνται οι επιλογές αναζήτησης πιο αναλυτικά.

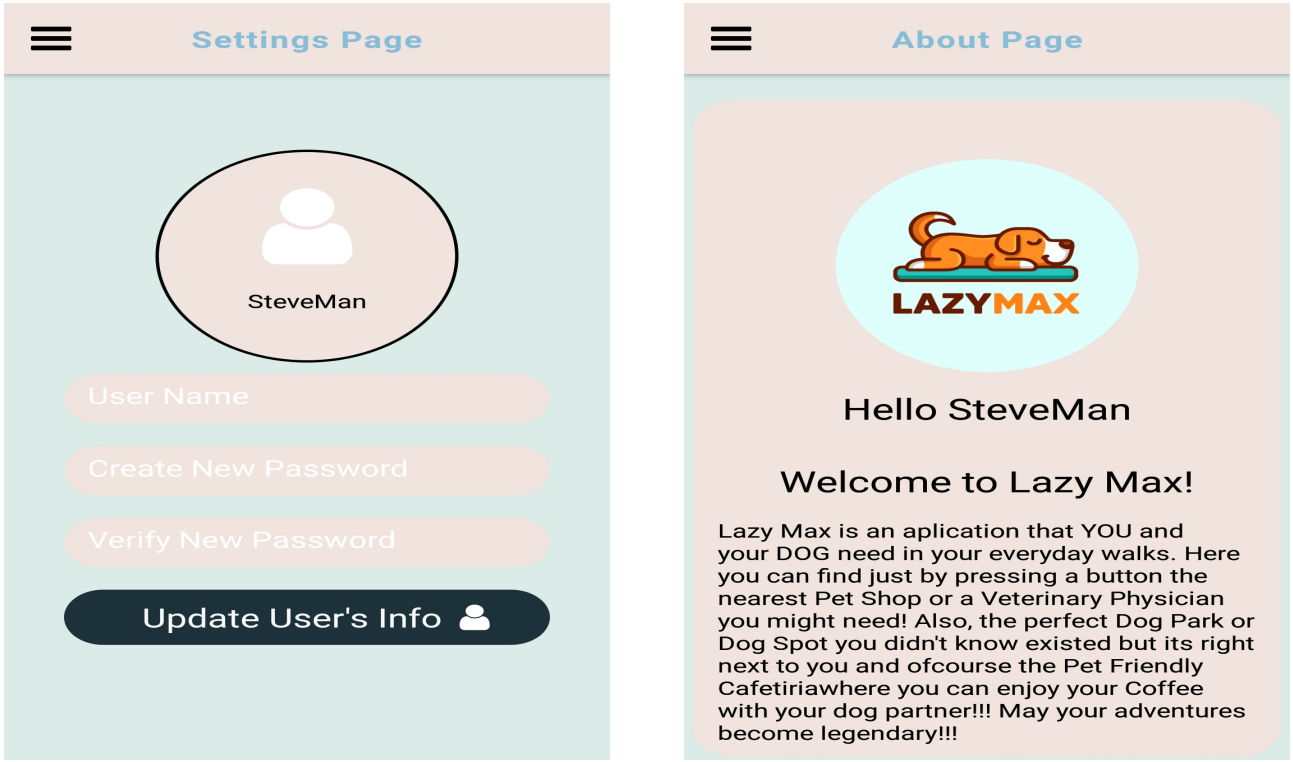
Σχήμα 6.2: Παράθυρο για βελτίωση Home Page

Στο σημείο αυτό αξίζει να αναφερθεί ότι για το παραπάνω πρόβλημα σχεδιάστηκε κατάλληλα το κουμπί ”Options” το οποίο θα ενημερώνει τον χρήστη για τις επιλογές αναζήτησης που έχει κατά την είσοδο του στην εφαρμογή(όπως φαίνεται στην εικόνα 6.3).



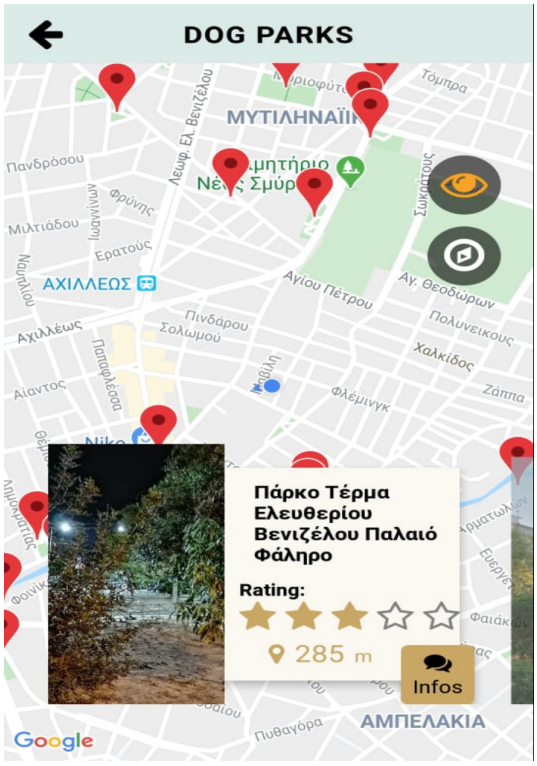
Σχήμα 6.3: Παράθυρο εμφάνισης επιλογών

3. Προχωρώντας, τονίστηκε η καλύτερη οργάνωση πληροφοριών στο Comment page και στο About Page για να περαστεί βέλτιστα το μήνυμα στον χρήστη.



Σχήμα 6.4: Παράθυρα για βελτίωση About Page, Setting Page

4. Τέλος, ένας τυπικός χρήστης ανέφερε ότι θα του άρεσε εάν το carousel πληροφοριών περιείχε ένα link για το κάθε κατάστημα, καφετέρια, πάρκο και κτηνιατρείο που όταν το πατήσει θα το πήγαινε στην ιστοσελίδα της αντίστοιχης τοποθεσίας.



Σχήμα 6.5: Παράθυρο για βελτίωση carousel

6.2.10 Συμπεράσματα Σχόλια

Οι παραπάνω αξιολογήσεις έκαναν ξεκάθαρο το γεγονός ότι η εφαρμογή δεν έχει σοβαρά προβλήματα. Ωστόσο, αυτό φυσικά δεν συνεπάγεται ότι δεν επιδέχεται βελτιώσεις. Με την αντιμετώπιση των προβλημάτων που εντοπίστηκαν -παρόλο που δεν θεωρούνται ζωτικής σημασίας-, καθίσταται δυνατό να οδηγηθούμε σε αναβάθμιση της εφαρμογής, με αισθητή βελτίωση της περιήγησης των χρηστών σε αυτή.

7. ΣΥΜΠΕΡΆΣΜΑΤΑ-ΜΕΛΛΟΝΤΙΚΈΣ ΕΠΕΚΤΆΣΕΙΣ

7.1 ΣΥΜΠΕΡΑΣΜΑΤΑ

Από το προηγούμενο κεφάλαιο μπορεί να διαπιστώσει κανείς την αναγκαιότητα ύπαρξης μιας τέτοιας εφαρμογής που πραγματεύεται η παρούσα διπλωματική εργασία καθώς και τις δυνατότητες που μπορεί να διαθέσει σε μια κοινότητα ανθρώπων που είναι ιδιοκτήτες σκυλιών. Μερικά από τα πλεονεκτήματα αυτής της εφαρμογής είναι ότι δεν απαιτείται καμιά προγραμματιστική γνώση από την πλευρά των χρηστών και ότι ταυτόχρονα καλύπτει και βασικές ανάγκες που έχουν οι ιδιοκτήτες στην ανατροφή των κατοικίδιων τους.

Με βάση τα δεδομένα που προέκυψαν από την αξιολόγηση και από τη σύγκρισή της με άλλες εφαρμογές έγινε φανερή η χρησιμότητα των μεθόδων που εφαρμόστηκαν, καθώς έγινε εντοπισμός κάποιων βασικών προβλημάτων με στόχο την μελλοντική διόρθωση τους.

7.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Στην παρούσα εφαρμογή, δεδομένου του ότι είναι demo, θα περιοριστούμε στην επίδειξη των βασικών λειτουργιών, όπως την εύρεση χώρων (καφετέριες) φιλικά προς τον χρήστη και το κατοικίδιό του, τοποθεσίες για πάρκα σκύλων, κτηνιατρεία και Pet Shops. Ωστόσο, η εφαρμογή έχει αρκετές προοπτικές. Κατόπιν συζητήσεων με χρήστες που συμμετείχαν στην αξιολόγηση της εφαρμογής αλλά και ιδιοκτήτες σκυλιών, επισημάνθηκαν αρκετές δυνατότητες και σημεία στα οποία μπορεί να επεκταθεί η εφαρμογή.

Μία από τις επεκτάσεις που συζητήθηκε αρκετά, ήταν η δημιουργία προσωπικού προφίλ για κάθε χρήστη-σκύλο, το οποίο φυσικά θα χειρίζεται ο αντίστοιχος ιδιοκτήτης. Σε κάθε προφίλ οι χρήστες θα μπορούν να ανεβάζουν φωτογραφίες αλλά και προσωπικές δημοσιεύσεις με τον σκύλο τους, μέρη που έχουν επισκεφθεί καθώς και τοποθεσίες στις οποίες βρίσκονται ή σκοπεύουν να επισκεφθούν. Αυτά, στη συνέχεια θα εμφανίζονται σε μία home page όπου οι φίλοι μεταξύ τους θα μπορούν να παρακολουθούν οτιδήποτε, κάνοντας ένα απλό scroll προς τα κάτω.

Επίσης, μία ακόμη πιθανή εξέλιξη της εφαρμογής είναι η ύπαρξη μίας υποσελίδας, η οποία θα παρουσιάζει όλες τις ράτσες σκύλων μαζί με αναρτήσεις χρηστών οι οποίοι ενδιαφέρονται για ζευγάρωμα των κατοικιδίων τους. Σε περίπτωση που κάποιος ενδιαφέρεται να ζευγαρώσει τον σκύλο του μπορεί εκεί να βρει πολύ εύκολα όλες τις πληροφορίες που χρειάζεται.

Ακόμα, συζητήθηκε αρκετά η ύπαρξη ενός chatroom όπου οι χρήστες θα μπορούν να επικοινωνούν μεταξύ τους, να γνωρίζονται αλλά και να ανταλλάσσουν χρήσιμες για τους σκύλους τους πληροφορίες. Σε αυτή την σκέψη βασίστηκε και η ιδέα των ειδοποιήσεων. Εδώ,οι χρήστες θα μπορούν να στέλνουν την ειδικά σχεδιασμένη ειδοποίηση «Βόλτα» σε έναν φίλο-κατοικίδιο για να κανονίσουν την κοινή τους έξοδο. Ο δέκτης της πρότασης θα ενημερώνεται άμεσα με ειδοποίηση.

Τέλος, μια ακόμη επέκταση της εφαρμογής αποτελεί η δημιουργία ενός Online Καταστήματος εντός της σελίδας. Ο χρήστης θα έχει τη δυνατότητα να αγοράσει προϊόντα απαραίτητα για τον σκύλο, όπως εξοπλισμός για την τροφή, αξεσουάρ, ενδύματα και πολλά άλλα . Ενδιαφέρουσα φαίνεται και η ιδέα να μπορεί να πουλήσει αντίστοιχα αντικείμενα που πλέον δεν χρειάζεται. Η σελίδα θα βρίσκεται σε συνεχή επικοινωνία με φυσικά καταστήματα ώστε να μπορεί να αξιολογεί και να εντοπίζει τα

προϊόντα μέσω της μηχανής αναζήτησης .

8. ΠΑΡΑΡΤΗΜΑ

Σε αυτή την ενότητα θα παρουσιάσουμε πιο αναλυτικά ορισμένα μέρη του κώδικα τα οποία κρίθηκαν ως σημαντικότερα. Συγκεκριμένα, αυτά τα τμήματα αντιπροσωπεύουν τον τρόπο με τον οποίο υλοποιήθηκαν οι βασικότερες λειτουργίες της εφαρμογής, όπως μέρη του front end και back end.

8.1 Κώδικας REACT NATIVE

Για όλα τα τμήματα του κώδικα του front end της εφαρμογής χρησιμοποιήθηκε η βιβλιοθήκη της JavaScript, η React Native. Ξεκινώντας, θα δούμε τις αρχικοποίησεις της εφαρμογής καθώς και τις άδειες που χρειάστηκαν για να έχουμε πρόσβαση στην εφαρμογή όταν αυτή τα ζητάει από την συσκευή του κινητού.

```
{
  "name": "FrontEnd",
  "version": "0.0.1",
  "private": true,
  "scripts": {
    "android": "react-native run-android",
    "ios": "react-native run-ios",
    "start": "react-native start",
    "test": "jest",
    "lint": "eslint ."
  },
  "dependencies": {
    "@react-native-community/geolocation": "^2.0.2",
    "@types/react-native-snap-carousel": "^3.8.1",
    "geolib": "^3.2.1",
    "moment": "^2.24.0",
    "react": "16.11.0",
    "react-native": "0.62.0",
    "react-native-carousel": "^0.12.0",
    "react-native-elements": "^1.2.7",
    "react-native-gesture-handler": "^1.6.1",
    "react-native-maps": "0.27.1",
    "react-native-maps-directions": "^1.7.6",
    "react-native-reanimated": "^1.7.1",
    "react-native-router-flux": "^4.2.0",
    "react-native-screens": "^2.4.0",
    "react-native-searchable-dropdown": "^1.1.1",
    "react-native-snap-carousel": "^3.9.0",
    "react-native-star-rating": "^1.1.0",
    "react-native-styled-text": "^1.0.1",
    "react-native-swipe-gestures": "^1.0.5",
    "react-native-swipeout": "^2.3.6",
    "react-native-vector-icons": "^6.6.0",
    "react-native-webview": "^9.0.2",
    "react-navigation": "^4.3.6",
    "react-navigation-drawer": "^2.4.10"
  }
```

Σχήμα 8.1: Package.json

Το αρχείο Package.json πρόκειται για το αρχείο στο οποίο γίνονται όλες οι αρχικοποίησεις της εφαρμογής όπως για παράδειγμα τα ονόματα,οι βιβλιοθήκες που πρόκειται να χρησιμοποιήσει και να

ενημερώσει αλλά και η έκδοση της React Native. Επίσης, στο αρχείο App.json(εικόνα 8.2) βρίσκουμε το όνομα με το οποίο εμφανίζεται η εφαρμογή στην κινητή συσκευή και μπορούμε να το αλλάξουμε.

```
{
  "name": "FrontEnd",
  "displayName": "FrontEnd"
}
```

Σχήμα 8.2: App.json

Στη συνέχεια παρατηρούμε το αρχείο AndroidManifest.xml (εικόνα 8.3). Το αρχείο αυτό επιτρέπει στην εφαρμογή να έχει πρόσβαση στο δίκτυο και ζητάει άδεια να έχει πρόσβαση στις πληροφορίες της κινητής συσκευής όπως για παράδειγμα την τοποθεσία της, άρα και την τοποθεσία του χρήστη που θα χρειαστούμε αργότερα όπως θα δούμε.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.frontend">

    <uses-permission android:name="android.permission.INTERNET" />
   <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <application
      android:name=".MainApplication"
      android:label="@string/app_name"
      android:icon="@mipmap/ic_launcher"
      android:roundIcon="@mipmap/ic_launcher_round"
      android:allowBackup="false"
      android:theme="@style/AppTheme">
      <activity
        android:name=".MainActivity"
        android:label="@string/app_name"
        android:configChanges="keyboard|keyboardHidden|orientation|screenSize|uiMode"
        android:launchMode="singleTask"
        android:windowSoftInputMode="adjustResize">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
      </activity>
      <activity android:name="com.facebook.react.devsupport.DevSettingsActivity" />
       <meta-data
     android:name="com.google.android.geo.API_KEY"
     android:value="AIzaSyBfn0jiHYaFZbhnO0NuuZyNN0NEjINYfrA"/>

   <!-- You will also only need to add this uses-libray tag -->
   <uses-library android:name="org.apache.http.legacy" android:required="false"/>
    </application>

</manifest>
```

Σχήμα 8.3: AndroidManifest.xml

Όπως κάθε web εφαρμογή έτσι και η δικιά μας κατά την κλήση της,το πρώτο αρχείο που θα διαβαστεί είναι το αρχείο index.json(εικόνα 8.4) και θα ”επιστρέψει” την ίδια την εφαρμογή.

```
/**
 * @format
 */

import {AppRegistry} from 'react-native';
import App from './App';
import {name as appName} from './app.json';
import {gestureHandlerRootHOC} from 'react-native-gesture-handler';
AppRegistry.registerComponent(appName, () => gestureHandlerRootHOC(App));

AppRegistry.registerComponent(appName, () => App);
console.disableYellowBox = true;
```

Σχήμα 8.4: Index.json

Αφού διαβάσει το index.json αρχείο καλείται το πρώτο component της εφαρμογής δηλαδή το App.js(εικόνα 8.5) στο οποίο έχουμε όλες τις οθόνες της εφαρμογής.

```
import React, { Component } from "react";
import { Platform, StyleSheet, Text, View, } from "react-native";
import Icon from 'react-native-vector-icons/MaterialIcons'


import Routes from "./src/routes.js";
import Navigator from './src/drawer'

class App extends React.Component {

  render() {
    return(


      <Routes/>

      )
  }
}
```

Σχήμα 8.5: App.js

Συνεχίζοντας, θα παρακολουθήσουμε τον σχεδιασμό του αρχείου Routes.js (εικόνα 8.6). Πρόκειται για ένα από τα σημαντικότερα αρχεία της εφαρμογής καθώς δίνει την δυνατότητα να οργανώσουμε και να ομαδοποιήσουμε τις οθόνες της εφαρμογής μας, Scenes κάτω από μια κεντρική σελίδα router, καθώς και να τις καλέσουμε με την ονομασία που τους έχουμε δώσει.

```
import React,{Component} from 'react';
import { Router, Stack, Scene,Tabs } from "react-native-router-flux";

import Login from "./pages/Login";
import Signup from "./pages/Signup";
import AfterLogin from './pages/AfterLogin'
import Parks from './pages/Parks'
import PetShops from './pages/PetShop'
import Vet from './pages/Vet'
import PetCafe from './pages/PetCaffe'
import Comments from './pages/Comments'
export default class Routes extends React.Component {
    render(){
        return(
        <Router>
        <Stack key="root" hideNavBar={true} initial>

                <Scene key="login" component={Login} title="Login" initial={true}/>
                <Scene key="signup" component={Signup} title="Register" />
                <Scene  key="afterlogin" component={AfterLogin} title="StudentApp"  hideNavBar={true} />
                <Scene key="parks" component={Parks} title="Dog-Park" />
                <Scene key="petshop" component={PetShops} title="Pet-Shop" />
                <Scene key="vet" component={Vet} title="Vet-Shop" />
                <Scene key="petcafe" component={PetCafe} title="Pet-Cafe" />
                <Scene key="com" component={Comments} title="Comment-Page"/>

          </Stack>
        </Router>

        )
    }
}
```

Σχήμα 8.6: Routers.js

Επιπλέον, αντίστοιχη οθόνη σαν την Router.js είναι και η Drawer.js(εικόνα 8.7), που οργανώνει και ομαδοποιεί τις βασικές οθόνες της εφαρμογής όπως Home Page, Settings Page, About Page και Sign out, με τις οποίες μπορεί να κινηθεί ο χρήστης μέσα στην εφαρμογή.

```
import * as React from 'react';
import { StyleSheet,Button, View } from 'react-native';

import { createDrawerNavigator } from 'react-navigation-drawer';
import { createAppContainer } from 'react-navigation';

import HomeStack from './homeStack';
import AboutStack from './aboutStack';
import SettingsStack from './settingsStack';

import Icon from "react-native-vector-icons/FontAwesome";

import Login from './pages/Login';


const RootDrawerNavigator = createDrawerNavigator({

  Home:{
    screen: HomeStack,
  },
  About:{
    screen: AboutStack,
  },
  Settings:{
  screen: SettingsStack,
  },
  Logout:{
    screen:Login
  },

})

export default createAppContainer(RootDrawerNavigator);
```

Σχήμα 8.7: Drawer.js

Στη συνέχεια, θα παρακολουθήσουμε τους κώδικες για τον σχεδιασμό και την υλοποίηση οθονών. Αρχικά, η οθόνη Home page που μέσα από τον κώδικα της(εικόνα 8.8) επιστρέφει το component με την παλάμη του σκύλου όπου δίνονται όλες οι επιλογές αναζήτησης τοποθεσιών.

```
render(){
  return (

    <View style={styles.container}>



      <Paw/>

          <View style={styles.menuContainer}>
              <DogFood/>
              <Medicine/>
              <DogWalk {...this.state}/>
              <DogCafe/>
            </View>




    </View>
  );
```

Σχήμα 8.8: Home.js

Παρατηρούμε ότι το component αποτελείται από τις τέσσερις επιλογές μέσα στην πατούσα οπότε προχωράμε να δούμε τον κώδικα για μια από της επιλογές η οποία είναι ίδια και για τις υπόλοιπες.

```
return(

 <View style={styles.container}>



   <TouchableOpacity style={styles.header} onPress={()=> this.props.navigation.goBack()}>
            <Icon name="arrow-left" size={28} style={styles.icon}/>
                <View>
                    <Text style={styles.headerText}>DOG PARKS</Text>
                </View>
    </TouchableOpacity>

   <MapView
     provider={PROVIDER_GOOGLE}
     style={styles.map}
     ref={map => this._map = map}
      region={{
          latitude: this.state.latitude,
          longitude:this.state.longitude,
          latitudeDelta: 0.015,
          longitudeDelta: 0.0121,
            }}
     showsUserLocation={true}
     >
     {this.state.markers.map(( marker,index) =>(



       <Marker
       key={marker.id}
       ref={ref => this.state.callouts[index]=ref}
       onPress={()=> this.onMarkerPressed(marker.geometry.location,index)}
       coordinate={{latitude:marker.geometry.location.lat,longitude:marker.geometry.location.lng}}
      title={marker.name}
       description={marker.vicinity}
      >


       </Marker>
```

Σχήμα 8.9: Parks.js

Στην παρούσα εικόνα βλέπουμε την επιστροφή του component με τον χάρτη, την τοποθεσία του χρήστη,τις τοποθεσίες της επιλογής του, τα κουμπιά επιλογής View carousel και Directions καθώς και την επιλογή πίσω. Στο σχήμα 8.9 βλέπουμε το component TouchableOpacity όπου δίνει την επιλογή στον χρήστη να επιστρέψει στην προηγούμενη οθόνη. Επίσης, με την χρήση του MapView εμφανίζεται ο χάρτης καθώς και το location του χρήστη ενώ το Marker εμφανίζει τις τοποθεσίες που παίρνει μέσα από την εντολή fetch από την Google API Maps (σχήμα 8.10).

```
Geolocation.getCurrentPosition(
  position => {
    this.setState({
      latitude: position.coords.latitude,
      longitude: position.coords.longitude,
      latitudeMarker: position.coords.latitude,
      longitudeMarker: position.coords.longitude

    })

                  return fetch('https://maps.googleapis.com/maps/api/place/nearbysearch/json?location='+this.state.latitude+','+this.state.longitude+'&radius=1000&type=pa
                  .then( (response) => response.json() )
                  .then( (responseJson) => {
                    this.setState({

                      markers:responseJson.results

                    })


                  })
    },
      error => this.setState({ error: error.message })


)
    .catch(error => alert("error:", error));
```

Σχήμα 8.10: fetch εντολή

Παρακάτω προτίθενται τα τμήματα κώδικα για τα κουμπιά View carousel και directions.

```
{/* WALKING OR DRIVING BOX */}
        <View style={styles.main}>

            <View style={styles.box}>
              {this.state.showCarousel ? (
                    <TouchableOpacity onPress={() => this.operationCarouseloff()}>

                      <Icon style={styles.iconMan} name="eye" size={32} color="orange" />
                      </TouchableOpacity>
                    ) : null}
                    {this.state.dontshowCarousel ? (
                      <TouchableOpacity onPress={() => this.operationCarouselon()}>
                      <Icon
                        style={styles.iconMan}
                        name="eye-slash"
                        size={32}
                        color="white"
                      />
                      </TouchableOpacity>
                    ) : null}

                </View>

            <View style={styles.secondbox}>
                    {this.state.showDirections ?(
                      <TouchableOpacity onPress={() => this.operationDirectionson()}>

                        <Icon style={styles.iconMan} name="compass" size={32} color="orange" />
                        </TouchableOpacity>
                    ):
                    <TouchableOpacity onPress={() => this.operationDirectionsoff()}>

                        <Icon style={styles.iconMan} name="compass" size={32} color="white" />
                        </TouchableOpacity>
                    }
            </View>
```

Σχήμα 8.11: View Button

```
{this.state.showDirections ? (
        <View style={styles.thirdbox}>
          <TouchableOpacity onPress={() => this.operationWalking()}>
            {this.state.showDriving ? (
              <Icon style={styles.iconMan} name="male" size={32} color="white" />
            ) : null}
            {this.state.showWalking ? (
              <Icon
                style={styles.iconMan}
                name="male"
                size={32}
                color="orange"
              />
            ) : null}
          </TouchableOpacity>
        </View>
        ): null }
{this.state.showDirections ? (
        <View style={styles.fourthbox}>
          <TouchableOpacity onPress={() => this.operationDriving()}>
            {this.state.showWalking ? (
              <Icon style={styles.iconMan} name="car" size={30} color="white" />
            ) : null}
            {this.state.showDriving ? (
              <Icon style={styles.iconMan} name="car" size={30} color="orange" />
            ) : null}
          </TouchableOpacity>
        </View>
        ): null }


</View>
```

Σχήμα 8.12: Directions Button

```
{this.state.showCarousel?(
 <Carousel
   ref={(c) => { this._carousel = c; }}
   containerCustomStyle={styles.carousel}
   data={this.state.markers}
   renderItem={this.renderCarouselItem}
   sliderWidth={Dimensions.get('window').width }
   itemWidth={300 }
   removeClippedSubviews={false}
   onSnapToItem={(index) => this.onCarouselItemaChange(index)}
 />)
 :null
}
```

Σχήμα 8.13: Σχήμα Carousel

Στο σχήμα 8.13 παρουσιάζεται η μέθοδος με τον οποία ορίζουμε ένα carousel με τις τοποθεσίες και τις αντίστοιχες πληροφορίες τους. Για παράδειγμά, ως data δέχεται τα δεδομένα από το Markers (δηλαδή τις τοποθεσίες), το renderItems καλεί την συνάρτηση renderCarouselItems(σχήμα 8.14 και σχήμα8.15) στο οποίο ορίζεται το πλαίσιο με τις αντίστοιχες πληροφορίες της κάθε τοποθεσίας όπως την εικόνα τους που διαθέτονται από την εντολή fetch μέσω Google Maps API, την ονομασία της τοποθεσίας, το starRating που παρέχει η google αλλά και την απόσταση της τοποθεσίας από τον χρήστη που δίνεται από την συνάρτηση getDistance.

```
renderCarouselItem = ({item}) =>
    <View style={styles.cardContainer}>

     {item.photos? (
              <Image  style={styles.cardImage}

                          source= {

                              {uri: 'https://maps.googleapis.com/maps/api/place/photo?maxwidth=9000&photoreference='+item.photos[0].photo_reference+'&key='+ApiKey+'


                              }




                           resizeMode="cover"
                  />) :

                  <Image
                  style={styles.cardImage}
                   source={require("../images/noImageAv.png")}
                   resizeMode='stretch'
                  >

                  </Image>
                  |
                  }
```

Σχήμα 8.14: Συνάρτηση renderCarouselItem 1

```
                          }
<View style={styles.textContainer}>

    <Text style={styles.titleStyle}>{item.name}</Text>

   <View  style={styles.starRating}>
          <Text style={{fontWeight:'bold'}}>Rating:</Text>
        <StarRating

           disabled={false}
           maxStars={5}
           rating={item.rating}
           fullStarColor={'#cba85f'}
           starSize={27}
          // selectedStar={(rating) => this.onStarRatingPress(rating)}
          />
        </View>
  <Text style={styles.locationStyle}>
        <Icon style={styles.iconMarker} name="map-marker" size={19} color="black" />
        <Text>  </Text>
        <Text style={{fontSize:20}}>
        {
          getDistance(
        { latitude: this.state.latitude, longitude:this.state.longitude },
        { latitude:item.geometry.location.lat, longitude:item.geometry.location.lng }
        )   }
        </Text>
        <Text style={{color:'#cba85f'}}>  m</Text>
    </Text>

   </View>

    <TouchableOpacity style={styles.infoBox} onPress={() => Actions.com({email:this.props.email,nametag:item.name})}>
        <Icon style={styles.iconComment} name="comments" size={19} color="black" />
        <Text style={styles.buttonText}>Infos</Text>
     </TouchableOpacity>



        </View>
```

Σχήμα 8.15: Συνάρτηση renderCarouselItem 2

```
}

 { this.state.flag ?(
<MapViewDirections
  origin={{
    latitude:this.state.latitude,
    longitude:this.state.longitude}}
  destination={{
    latitude:this.state.latitudeMarker,
    longitude:this.state.longitudeMarker}}
  apikey={ApiKey}
  strokeWidth={3}
  strokeColor="red"
  mode= {this.state.TravellingMode}
  />)
  :null
  }

</MapView>
```

Σχήμα 8.16: Σχήμα Directions

Στο σχήμα 8.16 παρουσιάζεται ο κώδικας για τον ορισμό του MapViewDirections που δίνει οδηγίες για τη μεταφορά από την θέση που βρίσκεται στην τοποθεσία που επέλεξε. Κάποια από τα βασικά της χαρακτηριστικά είναι οι ορισμοί origin και destination που εισάγονται οι συντεταγμένες του χρήστη και της τοποθεσίας αντίστοιχα. Ύστερα, έχουμε το πάχος και το χρώμα της γραμμής μεταξύ δύο σημείων (strokeWidth και strokeColor) και τέλος έχουμε mode οπού παίρνει ως τιμή ανάλογα με την επιλογή του χρήστη αμάξι ή περπάτημα για να βρεθεί η αντίστοιχη πλοήγηση.

Ένα ακόμα τμήμα του κώδικα ,που είναι άξιο αναφοράς, είναι ο κώδικας για την οθόνη Comments(σχήματα 8.17 και 8.18). Αρχικά, παρατηρούμε ότι υπάρχει ξανά η επιλογή επιστροφής στην προηγούμενη οθόνη μέσω του TouchableOpacity, έχουμε το FlatList στο οποίο δέχεται ως data τα σχόλια που υπάρχουν από κάθε τοποθεσία στην βάση μέσω εντολής fetch και τα κάνει render παρουσιάζοντας τα ως εξής:

- Για κάθε σχόλιο Item ορίζεται μια SwipeOut λειτουργία που κατά την κίνηση του δαχτύλου προς τα αριστερά πάνω στο σχόλιο, εμφανίζει ένα κουμπί delete. Αφού το επιλέξει ο χρήστης του εμφανίζεται μια ερώτηση σχετικά με το εάν είναι σίγουρος ότι θέλει να διαγράψει το σχόλιο του. Αφού συμφωνήσει, γίνεται έλεγχος έλεγχος ταυτοποίησης ,μέσω fetch, ενώ στη συνέχεια ακολουθεί η διαγραφή του (Σχήμα 8.19).
- Για κάθε σχόλιο Item ορίζεται το ItemList το οποίο είναι υπεύθυνο για τον τρόπο που παρουσιάζεται το σχόλιο, συμπεριλαμβάνοντας μαζί και το όνομα του αντίστοιχου χρήστη.

Τέλος, ο κώδικας παρέχει ένα inputBox , στο οποίο ο χρήστης εισάγει το σχόλιό του και πατώντας το κουμπί submit (το οποίο βρίσκεται μέσα στο TouchAbleOpacity) ολοκληρώνεται η διαδικασία εισαγωγής σχολίου.

```
return (

 <View style={styles.container}>
          <TouchableOpacity style={styles.header} onPress={()=> this.props.navigation.goBack()}>
                    <Icon name="arrow-left" size={28} style={styles.icon}/>
                        <View>
                           <Text style={styles.headerText}>{this.props.nametag}</Text>
                        </View>
          </TouchableOpacity>


                    <FlatList
                    style={styles.flatlist}
                    data={this.state.index}
                    ref={flat => this._flat = flat}
                    ItemSeparatorComponent = { this.FlatListItemSeparator }
                   // keyExtractor={item => item.id}
                    renderItem={({item})=>(

                     <Swipeout
                      right={swipeBtns}
                      cancelable='true'
                      autoClose='true'
                      backgroundColor= 'transparent'
                      onOpen={()=>this.setState({comFlag:item.id,comEmail:item.email})}
                      close={this.state.comFlag !== item.id}
                      >


                      <ListItem
                          roundAvatar
                          title={`${item.nameuser}`}
                          subtitle={item.comment}
                          containerStyle={{ borderBottomWidth: 0 }}
                          leftAvatar={{ source: { uri: 'http://'+IpKey+'/thessis/FrontEnd/src/images/Logo.png'} }}
                        //  avatar={{ uri:'http://'+IpKey+'/thessis/FrontEnd/src/images/Logo.png' }}
                        />
```

Σχήμα 8.17: Comments Component 1

```
                      </Swipeout>


                    )}
                    />


          <View style={styles.inputContainer}>
              <TextInput
                 ref={input => { this.textInput = input }}
                style={styles.inputBox}
                underlineColorAndroid="rgba(0,0,0,0)"
                placeholder={"ADD COMMENT"}
                placeholderTextColor="white"
                selectionColor="#fff"
                multiline={true}
                onChangeText={comment =>
                      this.setState({comment})
                    }
              />
        <TouchableOpacity style={styles.button} onPress={() => this.submit()}>
            <Icon style={styles.iconMan} name="reply" size={25} color="white" />

        </TouchableOpacity>
        </View>

    </View>

  );
}
```

Σχήμα 8.18: Comments Component 2

```
let swipeBtns = [{
 text: 'Delete',
 backgroundColor: 'red',
 underlayColor: 'rgba(0, 0, 0, 1, 0.6)',
 onPress: () => {
   Alert.alert(
     'Alert',
     'Are you sure you want to delete ?',
     [
       {text:'No',onPress:()=>console.log('Cancel Pressed'),style:'cancel'},
       {text:'Yes',onPress:()=>{
         if(this.props.email==this.state.comEmail){
         const user = {
           nametag:this.props.nametag,
           comFlag:this.state.comFlag,
           email:this.props.email
         };

         fetch("http://"+IpKey+"/thessis/BackEnd/public/api/comment", {
           method: "POST",
           body: JSON.stringify(user),
           headers: new Headers({
             Accept:'application/json',
             "Content-Type": "application/json"
           })
         })
           .then(res => res.json())
           .then(response => this.setState({

             index:response.message,
             comment:""

           }))
           .catch(error => alert("error:", error));

         }else{
           alert('Sorry,this is not your post')
         }


       }},

     ],
     {cancelable:true}
   );

 }
}];
```

Σχήμα 8.19: SwipeOut Κώδικας

8.2 Κώδικας Laravel

Σε αυτή την υποενότητα θα αναφερθούμε σε μέρη του κώδικα που αφορούν το backend της εφαρμογής. Για την δεδομένη εφαρμογή όλο το backend υλοποιήθηκε σε Laravel. Πιο συγκεκριμένα, θα παρουσιαστούν ορισμένα Controllers με τα οποία γίνονται οι απαραίτητοι έλεγχοι. Όταν τα Controllers καλούνται μέσω των fetch εντολών από το frontend, αποστέλλουν τα απαραίτητα δεδομένα που ζητήθηκαν την δεδομένη στιγμή.

```php
<?php

use Illuminate\Http\Request;

/*
|--------------------------------------------------------------------------
| API Routes
|--------------------------------------------------------------------------
|
| Here is where you can register API routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| is assigned the "api" middleware group. Enjoy building your API!
|
*/

Route::middleware('auth:api')->get('/user', function (Request $request) {
    return $request->user();
});



Route::post('comment','commentController@register');
Route::post('register','UserController@register');
Route::post('store','UserController@store');
Route::post('checkuser','UserController@checkuser');
Route::post('update','mainController@update');
Route::post('updatepassword','mainController@updatepassword');



Route::post('upload','mainController@upload');


Route::resource('fileupload', 'FileuploadController');

Route::middleware('auth:api')->get('/user',function(Request $request){
    return $request->user();
});
```

Σχήμα 8.20: Api.php

Στο αρχείο Api.php (σχήμα 8.20) εμφανίζονται όλα τα αιτήματα που γίνονται στο back end της εφαρμογής από τον χρήστη. Υπάρχουν, δύο τρόποι με τους οποίους μπορεί να πραγματοποιηθεί ένα αίτημα,είτε με την GET είτε με την POST εντολή. Αρχικά, με την GET ο χρήστης κάνει αίτημα στο back end να του επιστρέψει κάποιες πληροφορίες που βρίσκονται στην βάση χωρίς να αποθηκεύσει απολύτως τίποτα. Για παράδειγμα στην εφαρμογή μας, όταν ο χρήστης επιλέξει να δει τα info-σχόλια κάποιας τοποθεσίας κάνει έκκληση στο back end να του επιστρέψει όσα σχόλια υπάρχουν για την συγκεκρι- μένη τοποθεσία. Ενώ, όσον αφορά την μέθοδο POST ο χρήστης στέλνει κάποια δεδομένα στο back end με την προϋπόθεση αυτά να αποθηκευτούν στην βάση δεδομένων. Ένα απλό παράδειγμα είναι εκείνο που ο χρήστης κάνει εγγραφεί στην εφαρμογή και στέλνει τα προσωπικά του δεδομένα όπως username, email και password.

Τέλος, κάθε αίτημα στην Api.php αποτελείται από 2 μεταβλητές. Η πρώτη μεταβλητή, αφορά τον τρόπο με τον οποίο το back end αναγνωρίζει το αίτημα που στάλθηκε και η δεύτερη μεταβλητή αφορά την συνάρτηση και το αντίστοιχο controller μέσα στο οποίο θα εκτελεστεί το αίτημα. Στην συνέχει θα παρακολουθήσουμε τμήματα κώδικα που αφορούν συγκεκριμένα Controller.

Στο σχήμα 8.21 απεικονίζεται το Controller που είναι υπεύθυνο για τον ελέγχο και την εγγραφή των προσωπικών δεδομένων του χρήστη στην Laravel. Πιο συγκεκριμένα, όταν ο χρήστης κάνει εγγραφή στην εφαρμογή στέλνει αίτημα στο back end με POST μέθοδο και αυτό με την σειρά του καλεί την συνάρτηση store του UserController. Η συγκεκριμένη, σε πρώτη φάση ελέγχει εάν υπάρχει εγγραφή κάποιου χρήστη με το ίδιο email. Στην περίπτωση που επιβεβαιωθεί η μοναδικότητά του, αποθηκεύει τα δεδομένα στην βάση. Παρόμοια είναι και η λειτουργία της συνάρτησης checkUser όταν προσπαθεί ο χρήστης να κάνει Log in.

```php
public function store(Request $request)
{
    $data= user::all();
    $email = $request->email;
    $password=$request->password;
    $error_message="DENIED";
    $Success = "Success";


    if (user::where('useremail', '=',$email)->count() > 0)
    {
      $hashed_password= user::where('useremail', '=',$email)->value('password');
      if(Hash::check($password, $hashed_password))
      {
        return response()->json(['message' => $Success]);
      }
      else
      {
        return response()->json(['message' => $error_message]);
      }

    }
    else
    {
      return response()->json(['message' => $error_message]);
    }
}
```

```php
public function checkuser(Request $request)
{
   $data= user::all();
   $email = $request->email;
   $error_message="DENIED";
   $Success = "Success";



   if (user::where('useremail', '=',$email)->count() > 0)
    {
      $user= user::select('username','useremail','password')->where('useremail', '=',$email)->get();
      return response()->json(['message' => $user]);
    }else{
       return response()->json(['message' => $error_message]);
    }

}
```

Σχήμα 8.21: UserController

```php
public function update(Request $request)
{

    $useremail = $request->email;
    $newname = $request->new_name;
    $newpassword = $request->new_password;
    $hash = Hash::make($newpassword);

    $error_message="DENIED";
    $Success = "Success";

    if ($user = User::where('useremail','=' ,$useremail)->first()) {

        $user->username = $newname;
        $user->password = $hash;
        $user->save();


        return response()->json(['message' => $Success]);
    } else {
       return response()->json(['message' => $error_message]);
    }

}
```

Σχήμα 8.22: MainController

Στο σχήμα 8.22 φαίνεται η συνάρτηση update του MainController η οποία είναι υπεύθυνη για την αναβάθμιση των προσωπικών δεδομένων του χρήστη, όπως username και password όταν ο χρήστης κάνει αίτημα μέσα από την Setting Page για αλλαγή των δεδομένων.

```
class commentController extends Controller
{
   public function register(Request $request)
   {   $nametag=$request->nametag;
       $comment=$request->comment;
       $naneuser=$request->naneuser;
       $flag=$request->flag;
       $idFlag=$request->comFlag;
       $error_message="DENIED";
        $Success = "Success";
        $email=$request->email;
        $id=$request->id;


       $name= user::where('useremail', '=',$naneuser)->value('username');
          if($flag=='true'){
           $index= comment::select('id','comment','naneuser','email')->where('nametag', '=',$nametag)->orderBy('id','DESC')->get();

           return response()->json(['message' => $index]);
          }elseif($flag=='false')
          {

          $index =comment::insert(['id'=>$id,'nametag'=>$nametag,'comment'=>$comment,'naneuser'=>$nane,'email'=>$naneuser]);
          $index= comment::select('id','comment','naneuser','email')->where('nametag', '=',$nametag)->orderBy('id','DESC')->get();
          return response()->json(['message' => $index]);
          }else{


              $nonent=comment::where('id', '=',$idFlag)->delete();
              $index= comment::select('id','comment','naneuser','email')->where('nametag', '=',$nametag)->orderBy('id','DESC')->get();
              return response()->json(['message' => $index]);




          }



   }
}
```

Σχήμα 8.23: commentController

Παραπάνω (σχήμα 8.23) εμφανίζεται ένας ακόμα βασικός και απαραίτητος Controller. Το commentController είναι υπεύθυνο για όλα τα σχόλια που αφορούν την κάθε τοποθεσία. Πιο συγκεκριμένα, ως βασικές λειτουργίες έχει την εμφάνιση,εισαγωγή και διαγραφή σχολίων που έχουν κάνει οι χρήστες σε κάθε τοποθεσία και ανάλογα με το αίτημα του χρήστη ενεργοποιείται η αντίστοιχη λειτουργία.

```
{
    "name": "laravel/laravel",
    "type": "project",
    "description": "The Laravel Framework.",
    "keywords": [
        "framework",
        "laravel"
    ],
    "license": "MIT",
    "require": {
        "php": "^7.2.5",
        "fideloper/proxy": "^4.2",
        "fruitcake/laravel-cors": "^1.0",
        "guzzlehttp/guzzle": "^6.3",
        "laravel/framework": "^7.0",
        "laravel/tinker": "^2.0"
    },
    "require-dev": {
        "facade/ignition": "^2.0",
        "fzaninotto/faker": "^1.9.1",
        "laravel/ui": "^2.0",
        "mockery/mockery": "^1.3.1",
        "nunomaduro/collision": "^4.1",
        "phpunit/phpunit": "^8.5"
    },
    "config": {
        "optimize-autoloader": true,
        "preferred-install": "dist",
        "sort-packages": true
    },
    "extra": {
        "laravel": {
            "dont-discover": []
        }
    },
    "autoload": {
        "psr-4": {
            "App\\": "app/"
        },
        "classmap": [
            "database/seeds",
            "database/factories"
        ]
    },
    "autoload-dev": {
        "psr-4": {
            "Tests\\": "tests/"
        }
    }
```

Σχήμα 8.24: Composer.json

Τέλος, στο σχήμα 8.24 Composer.json παρατηρούμε τις αρχικοποιήσεις που χρειάστηκαν να γίνουν στο back end.

Συμπερασματικά, μέσα από τη μελέτη και την υλοποίηση της παρούσας εφαρμογής Lazy Max γίνεται κατανοητό πως υφίσταται μια ανάγκη για τέτοιου είδους εφαρμογές, οι οποίες εξυπηρε- τούν

τις βασικές ανάγκες των ιδιοκτητών κατοικίδιων, εν προκειμένω σκύλων. Στο σημείο αυτό, σημαντικό θα ήτανε να επισημάνουμε το γεγονός ότι τέτοιου είδους εφαρμογές αποτελούν ένα δείγμα για το τι μπορεί να επιτευχθεί μελλοντικά και σε άλλες κοινότητες ατόμων, εφόσον η εφαρμογή αυτή εμπεριέχει την κύρια δομή για την επέκταση σε άλλους τομείς λόγου χάριν εμπορίου, κοινωνικής δικτύωσης και γενικότερα ψυχαγωγίας.

9. ΒΙΒΛΙΟΓΡΑΦΙΑ

(1) Advantages and Disadvantages of Javascript.https://guide.freecodecamp.org/javas-cript/ad- vantages-and-disadvantages-of-javascript/

(2) Αβούρης, N. (2000). Εισαγωγή στην Επικοινωνία Ανθρώπου – Υπολογιστή. Εκδόσεις Δίαυλος.

(3) React Native Benefits.https://medium.com/weblineindia/what-is-reactive-development-advantages-and-disadvantages-e5bf052473a7

(4) How To Create A Todo List Application In React Native With Laravel Backend. https://www.techomoro.com/how-to-create-a-todo-list-application-in-react-native-with-laravel- backend/

(5) The Top 10 Advantages Of Using Laravel PHP Framework. https://belitsoft.com/laravel-develop- ment-services/10-benefits-using-laravel-php-framework

(6) How To Connect React Native Application With Laravel API.https://appdivi-dend.com/2018/03/21/how-to-connect-react-native-application-with-laravel-api/

(7) Create native apps for Android and iOS using React.https://facebook.github.io/react-native/

(8) Laravel Documentation.https://laravel.com/docs/6.x

(9) Google Maps API. https://developers.google.com/maps/documentation

(10) React Native Carousel.https://github.com/archriss/react-native-snap-carousel

(11) React Native Maps. https://github.com/react-native-community/react-native-maps

(12) DeviceAtlas Report.https://techingreek.com/deviceatlas-egine-erevna-kai-t

(13) 10 Usability Heuristics for User Interface Design.https://www.nngroup.com/art-icles/ten-usabil- ity-heuristics/

(14) Nielsen, J. (1994b). Heuristic Evaluation.In Nielsen, J., and Mack, R.L. (Eds.), Usability In- spetion Methods, John Wiley and Sons, New York,NY.

(15) Likert, Rensis (1932). ”A Technique for the Measurement of Attitudes”.Archives of Psychology. 140: 1–55.

Πανεπιστήμιο Πατρών, Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών
Ευστράτιος Καρδαμπίκης του Θεμιστοκλή
© Μάιος 2020 – Με την επιφύλαξη παντός δικαιώματος.