



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ANDROID ΓΙΑ ΤΗΝ ΑΝΑΖΗΤΗΣΗ ΠΛΗΣΙΕΣΤΕΡΩΝ ΘΕΣΕΩΝ ΕΞΥΠΗΡΕΤΗΣΗΣ-ΕΝΔΙΑΦΕΡΟΝΤΟΣ



Της φοιτήτριας
Σισκάκη Ξένιας
Αρ. Μητρώου: 103633

Επιβλέπων καθηγητής
Κλεφτούρης Δημήτριος

Θεσσαλονίκη 2015

Πρόλογος

Στο παρακάτω κείμενο θα αναφερθώ στην πτυχιακή μου εργασία, κατά την οποία ανέπτυξα μια εφαρμογή android για κινητά τηλέφωνα και tablets. Η εφαρμογή αυτή έχει σκοπό να βοηθήσει τον χρήστη, όπου και αν βρίσκεται αυτός, να βρει τα πλησιέστερα σημεία εξυπηρέτησης-ενδιαφέροντος στην πόλη της Θεσσαλονίκης.

Περίληψη

Η εφαρμογή που δημιούργησα, αναπτύχθηκε με την γλώσσα προγραμματισμού Java for Android και μπορεί να χρησιμοποιηθεί από συσκευές android, δηλαδή συσκευές οι οποίες λειτουργούν με το λειτουργικό σύστημα Android. Τέτοιες συσκευές είναι οι συσκευές νέας τεχνολογίας, οι γνωστές και ως «smart devices».

Η εφαρμογή αυτή αξιοποιεί τη γεωγραφική θέση του χρήστη μέσω του GPS, με σκοπό να προσδιορίσει τα πλησιέστερα σε αυτόν σημεία εξυπηρέτησης-ενδιαφέροντος, όπως για παράδειγμα εστιατόρια, σούπερ-μάρκετ και νοσοκομεία, μεταξύ άλλων.

Στο κείμενο αυτό θα αναφερθώ εκτενέστερα στις διάφορες τεχνολογίες, στα προγράμματα αλλά και τις διάφορες βιβλιοθήκες και APIs που χρησιμοποίησα έτσι ώστε να φέρω εις πέρας την εφαρμογή αυτή.

Εδώ θα παραθέσω εν συντομία μια λίστα αυτών:

1. Eclipse IDE
2. Android SDK
3. Java for Android
4. Βάση δεδομένων MySQL
5. PHP
6. Google Maps API v2
7. Universal Image Loader
8. Google Maps Android Marker Clustering Utility
9. Google Directions Android

Abstract

The application that I developed, was written in Java for Android and can be used by android devices, that is, devices that run on the Android Operating System. Such devices are the new technology devices, also known as “smart devices”.

This application utilizes the user’s geographical location via GPS in order to determine the closest to him points of interest (pois), like for example restaurants, supermarkets and hospitals between others.

In this paper, I will discuss extensively the different technologies, programs, libraries and APIs that I used in order to accomplish the development of this application.

Here is a brief list of the aforementioned:

1. Eclipse IDE
2. Android SDK
3. Java for Android
4. MySQL Database
5. PHP
6. Google Maps API v2
7. Universal Image Loader
8. Google Maps Android Marker Clustering Utility
9. Google Directions Android

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον πατέρα μου, καθώς χωρίς την πολύτιμη βοήθεια που μου πρόσφερε, αυτή η εφαρμογή δεν θα είχε την μορφή που έχει τώρα. Όλα τα εικονίδια της εφαρμογής καθώς και το λογότυπό της σχεδιάστηκαν από αυτόν μετά από πολλές προσπάθειες και αλλαγές. Τον ευχαριστώ ιδιαίτερα για την υπομονή που έδειξε μέχρι το πέρας της εφαρμογής, καθώς η εφαρμογή είχε μεγάλες απαιτήσεις για πολλά, διαφορετικού μεγέθους, εικονίδια.

Ακόμα, θα ήθελα να ευχαριστήσω την συμφοιτήτρια και φίλη μου, Κυριαζίδου Σελήνη, για τις αμέτρητες ώρες συνομιλίας στις οποίες ανταλλάξαμε ιδέες, απόψεις, προβλήματα και λύσεις τόσο για την συγγραφή του κώδικα όσο και για την δημιουργία του γραφικού περιβάλλοντος.

Τέλος, ευχαριστώ όλους τους χρήστες που βρήκαν λίγο χρόνο να τεστάρουν την εφαρμογή και να μου προσφέρουν ανατροφοδότηση έτσι ώστε να γίνει αυτή πιο φιλική προς τον χρήστη.

Περιεχόμενα

Πρόλογος	3
Περίληψη	4
Abstract	5
Ευχαριστίες	6
Περιεχόμενα	7
Ευρετήριο σχημάτων	9
Ευρετήριο πινάκων	10
Ευρετήριο αποσπασμάτων κώδικα	10
Εισαγωγή	11
Κεφάλαιο 1: Η τεχνολογία Android	12
1.1 Ιστορικά γεγονότα	12
1.2 Εκδόσεις και APIs	12
1.3 Χρήση των διαφορετικών εκδόσεων	14
1.4 Ανταγωνισμός	15
Κεφάλαιο 2: Ο server	17
Κεφάλαιο 3: Βάση δεδομένων MySQL	18
3.1 Πίνακες και δεδομένα στην βάση	18
3.2 Το script δημιουργίας της βάσης	20
Κεφάλαιο 4: Η γλώσσα προγραμματισμού PHP	23
4.1 Επικοινωνία με τον server	23
4.1.1 Η μορφοποίηση JSON	23
4.1.2 PHP scripts	24
Κεφάλαιο 5: Ανάπτυξη εφαρμογής	27
5.1 Eclipse	27
5.1.1 Απαιτήσεις για το android	27
5.2 Google Maps	28
5.3 Genymotion	29
Κεφάλαιο 6: Google Maps	30
6.1 Google Maps API v2	30

6.1.1 Προϋποθέσεις χρήσης	30
6.1.2 Το API	31
Κεφάλαιο 7: Εξωτερικές βιβλιοθήκες	33
7.1 Universal Image Loader	33
7.1.1 Το API	33
7.2 Google clustering.....	34
7.2.1 Το API	35
7.3 Google Directions	36
7.3.1 Το API	36
7.4 Google play services	38
Κεφάλαιο 8: Απαιτήσεις εφαρμογής	39
8.1 Internet	39
8.2 GPS.....	39
8.3 Συμβατότητα συσκευών	39
8.3.1 Έκδοση android	39
8.3.2 Οθόνες συσκευών	40
Κεφάλαιο 9: Οδηγός χρήσης λογισμικού	41
9.1 Use case αναζήτησης μιας τράπεζας.....	49
Κεφάλαιο 10: Συσκευές δοκιμής	52
Επίλογος	53
Βιβλιογραφία	54

Ευρετήριο σχημάτων

Εικόνα 1: Τα διάφορα λογότυπα των εκδόσεων του android	14
Εικόνα 2: Τα ποσοστά χρήσης των διάφορων εκδόσεων του android	15
Εικόνα 3: Τα ποσοστά χρήσης των διάφορων κινητών λειτουργικών συστημάτων για το πρώτο εξάμηνο του 2015.....	16
Εικόνα 4: Η γραφική αναπαράσταση των πινάκων της βάσης δεδομένων και των μεταξύ τους σχέσεων.....	20
Εικόνα 5: Παράδειγμα χρήσης του API key στο manifest της εφαρμογής	31
Εικόνα 6: Ένας marker στον χάρτη, όπως αυτό εμφανίζεται στην εφαρμογή.	32
Εικόνα 7: Παράδειγμα εμφάνισης των clusters στον χάρτη, όπως αυτό εμφανίζεται στην εφαρμογή.....	34
Εικόνα 8: Εμφάνιση των λεπτομερειών του σημείου κατά την επιλογή του, όπως αυτό εμφανίζεται στην εφαρμογή.....	35
Εικόνα 9: Εμφάνιση διαδρομής με αυτοκίνητο στο χάρτη, όπως αυτό εμφανίζεται στην εφαρμογή	37
Εικόνα 10: Εμφάνιση διαδρομής με τα πόδια στο χάρτη, όπως αυτό εμφανίζεται στην εφαρμογή	37
Εικόνα 11: Η αρχική οθόνη της εφαρμογής σε κάθετο προσανατολισμό	40
Εικόνα 12: Η αρχική οθόνη της εφαρμογής σε οριζόντιο προσανατολισμό	40
Εικόνα 13: Το λογότυπο της εφαρμογής που εμφανίζεται κατά την έναρξή της	41
Εικόνα 14: Το μενού με τις επιλογές της τοποθεσίας για την αναζήτηση	42
Εικόνα 15: Το μενού των ρυθμίσεων της εφαρμογής	42
Εικόνα 16: Οι ρυθμίσεις της εφαρμογής	42
Εικόνα 17: Το μήνυμα προτροπής προς το χρήστη για την ενεργοποίηση της σύνδεσης στο Διαδίκτυο	43
Εικόνα 18: Το μήνυμα προτροπής προς το χρήστη για την ενεργοποίηση του GPS.....	43
Εικόνα 19: Το μήνυμα που λαμβάνει ο χρήστης όταν δεν έχουν βρεθεί αποτελέσματα για την αναζήτησή του.....	43
Εικόνα 20: Τα αποτελέσματα της αναζήτησης και το μενού ταξινόμησης αυτών	44
Εικόνα 21: Το μενού επιλογής της ακτίνας απόστασης των σημείων από την τοποθεσία του χρήστη	44
Εικόνα 22: Ο χάρτης με τα clustered σημεία και το μενού αλλαγής του εδάφους του χάρτη.....	45
Εικόνα 23: Ο χάρτης με τα clustered σημεία, με δορυφορικό έδαφος.....	45
Εικόνα 24: Ένα επιλεγμένο σημείο στο χάρτη εμφανίζει τις λεπτομέρειές του	46

Εικόνα 25: Η οθόνη με τις λεπτομέρειες του επιλεγμένου σημείου	46
Εικόνα 26: Μήνυμα προειδοποίησης για επιπλέον χρεώσεις κατά την κλήση του τηλεφώνου του σημείου	47
Εικόνα 27: Το ωρολόγιο πρόγραμμα του επιλεγμένου σημείου	47
Εικόνα 28: Ο πίνακας για την αναφορά ανακριβών στοιχείων ενός σημείου	48
Εικόνα 29: Το σημείο στο χάρτη με τις πληροφορίες που παρέχει κατά την επιλογή του marker .	48
Εικόνα 30: Οι επιλογές για την εμφάνιση της διαδρομής προς το επιλεγμένο σημείο στο χάρτη .	49
Εικόνα 31: Η λίστα με τα αγαπημένα	49
Εικόνα 32: Η λίστα με τα αποτελέσματα, με φίλτρο για αναζήτηση συγκεκριμένης τράπεζας.....	50
Εικόνα 33: Οι λεπτομέρειες της τράπεζας	51
Εικόνα 34: διαδρομή με αυτοκίνητο από την τοποθεσία του χρήστη στην τράπεζα	51

Ευρετήριο πινάκων

Πίνακας 1: Οι διάφορες εκδόσεις και τα APIs του λειτουργικού συστήματος android	13
Πίνακας 2: Οι παράμετροι που χρειάζονται για τα διάφορα queries στη βάση δεδομένων και η λειτουργία που εκτελεί το καθένα	25

Ευρετήριο αποσπασμάτων κώδικα

Απόσπασμα κώδικα 1: Η δημιουργία των πινάκων στη βάση.....	21
Απόσπασμα κώδικα 2: Η εισαγωγή των απαραίτητων δεδομένων στους πίνακες.....	22
Απόσπασμα κώδικα 3: Παράδειγμα ενός αντικειμένου JSON.....	24
Απόσπασμα κώδικα 4: Παράδειγμα ανάλυσης(parsing) ενός JSONArray.....	26

Εισαγωγή

Η εφαρμογή αυτή, που ονομάζεται **spotMe, Thessaloniki in spots**, δημιουργήθηκε στο πλαίσιο της πτυχιακής μου εργασίας και έχει ως κύριο σκοπό να βοηθήσει τον χρήστη στην αναζήτηση πλησιέστερων σημείων εξυπηρέτησης-ενδιαφέροντος στην πόλη της Θεσσαλονίκης. Μια τέτοια εφαρμογή αποτελεί χρήσιμο εργαλείο και για τους ίδιους τους κατοίκους της πόλης αυτής, αλλά κυρίως για τους ταξιδιώτες που μπορεί να επισκέπτονται την πόλη για πρώτη φορά. Αποτελεί δηλαδή έναν οδηγό της πόλης στον οποίο ο χρήστης μπορεί να βρει κάθε λογής σημεία ενδιαφέροντος, από κινηματογράφους μέχρι και ξενοδοχεία. Προσφέρει χρήσιμες πληροφορίες για το κάθε σημείο, όπως επίσης και την διαδρομή που πρέπει να ακολουθήσει ο χρήστης από την τωρινή του τοποθεσία μέχρι το σημείο αυτό. Η εφαρμογή είναι διαθέσιμη στα αγγλικά και στα ελληνικά, έτσι ώστε να είναι ευκολόχρηστη και από ξενόγλωσσους χρήστες.

Παρόμοιες εφαρμογές έχουν αναπτυχθεί από την εταιρεία Susuko Maka LLC για τις πόλεις της Αθήνας και Θεσσαλονίκης. Αυτές ονομάζονται AthensBook¹ και ThessBook² αντίστοιχα και ο αναγνώστης μπορεί να τις βρει στο Google Play Store.

¹ <http://www.athensbook.gr/>

² <http://www.thessbook.gr/>

Κεφάλαιο 1: Η τεχνολογία Android

Το Android είναι ένα λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας αλλά και πολλές άλλες συσκευές τελευταίας τεχνολογίας, όπως tablets, τηλεοράσεις και συστήματα πλοήγησης αυτοκινήτων (GPS). Είναι βασισμένο στον πυρήνα του λειτουργικού συστήματος Linux. Είναι σήμερα ένα από τα πιο δημοφιλή, αν όχι το δημοφιλέστερο, λειτουργικά συστήματα «έξυπνων συσκευών», γνωστών και ως «smart devices». Έχει βέβαια αρκετούς ανταγωνιστές στην αγορά, όπως για παράδειγμα το λειτουργικό σύστημα iOS της Apple και το Windows Mobile της Microsoft. Για τον ανταγωνισμό αυτό θα αναφερθώ παρακάτω.

1.1 Ιστορικά γεγονότα

Στα τέλη του 2003 ιδρύθηκε στην California η εταιρεία Android, Inc. από μια μικρή ομάδα τεσσάρων ατόμων. Ο αρχικός σκοπός της εταιρείας ήταν να δημιουργήσει ένα λειτουργικό σύστημα για ψηφιακές φωτογραφικές μηχανές. Παρόλα αυτά, σύντομα συνειδητοποίησαν ότι η αγορά για τέτοιες συσκευές δεν ήταν αρκετά μεγάλη και έτσι έστρεψαν το ενδιαφέρον τους προς την δημιουργία ενός λειτουργικού συστήματος για κινητά τηλέφωνα, το οποίο θα ανταγωνιζόταν τα ήδη υπάρχοντα λειτουργικά συστήματα Symbian και Windows Mobile.

Στα μέσα του 2005, η Google αγόρασε την Android, Inc. για ένα σημαντικά μεγάλο χρηματικό ποσό. Η αγορά αυτή σηματοδότησε την ανάπτυξη του μεγάλου και δημοφιλούς αυτού λειτουργικού συστήματος, το οποίο έκανε την πρώτη του εμφάνιση στις αγορές κινητής τηλεφωνίας τον Οκτώβριο του 2008. Από τότε, πολλές γνωστές εταιρείες κινητής τηλεφωνίας, όπως η Sony, η Samsung, η HTC και η LG, συνεργάζονται με την Google για την κυκλοφορία νέων μοντέλων έξυπνων συσκευών android.

1.2 Εκδόσεις και APIs

Από την εμφάνιση της πρώτης έκδοσης του android μέχρι και σήμερα, έχουν κυκλοφορήσει συνολικά 54 εκδόσεις του γνωστού λειτουργικού συστήματος. Ακόμα, υπάρχουν συνολικά 22 διαφορετικά APIs, καθώς πολλές από τις εκδόσεις εισήγαγαν καινούρια χαρακτηριστικά στο γραφικό περιβάλλον του λειτουργικού συστήματος. Το API -συντομογραφία του αγγλικού όρου **Application Programming Interface**- είναι ένα σύνολο λειτουργιών, μεθόδων και εργαλείων για την δημιουργία εφαρμογών. Εκφράζει δηλαδή, τις λειτουργίες από τις οποίες

αποτελείται ένα συστατικό λογισμικού (software component). Πάνω σε αυτές τις λειτουργίες βασίζεται στην συνέχεια ο προγραμματιστής κατά τη δημιουργία μιας εφαρμογής.³

Ο διαχωρισμός των εκδόσεων, ο αριθμός, η ονομασία και κυκλοφορία έκδοσης, καθώς και τα αντίστοιχα APIs φαίνονται στον Πίνακα 1. Ακόμα, στην Εικόνα 1 φαίνονται τα αστεία αλλά και πρωτότυπα ονόματα και λογότυπα των διαφορετικών εκδόσεων. Να σημειωθεί ότι στον Πίνακα 1 παρατίθενται μόνο οι εκδόσεις από το API 11 (έκδοση 3.0) και πάνω, καθώς μικρότερες εκδόσεις χρησιμοποιούνται πλέον σπάνια από κινητές συσκευές.

Πίνακας 1: Οι διάφορες εκδόσεις και τα APIs του λειτουργικού συστήματος android⁴

Όνομασία	Έκδοση	Ημερομηνία κυκλοφορίας
Android 3.0 Honeycomb (API level 11)	3.0	Φεβρουάριος, 2011
Android 3.1 Honeycomb (API level 12)	3.1	Μάιος, 2011
Android 3.2-3.2.6 Honeycomb (API level 13)	3.2, 3.2.2, 3.2.2, 3.2.3, 3.2.4, 3.2.5, 3.2.6	Ιούλιος, 2011 - Φεβρουάριος, 2012
Android 4.0-4.0.2 Ice Cream Sandwich (API level 14)	4.0, 4.0.1, 4.0.2	Οκτώβριος, 2011 - Νοέμβριος, 2011
Android 4.0.3-4.0.4 Ice Cream Sandwich (API level 15)	4.0.3, 4.0.4	Δεκέμβριος, 2011 - Μάρτιος, 2012
Android 4.1-4.1.2 Jelly Bean (API level 16)	4.1, 4.1.1, 4.1.2	Ιούλιος, 2012 - Οκτώβριος, 2012
Android 4.2-4.2.2 Jelly Bean (API level 17)	4.2, 4.2.1, 4.2.2	Νοέμβριος, 2012 - Φεβρουάριος, 2013

³ https://en.wikipedia.org/wiki/Application_programming_interface

⁴ https://en.wikipedia.org/wiki/Android_version_history

Όνομασία	Έκδοση	Ημερομηνία κυκλοφορίας
Android 4.3-4.3.1 Jelly Bean (API level 18)	4.3, 4.3.1	Ιούλιος, 2013 - Οκτώβριος, 2013
Android 4.4-4.4.4 KitKat (API level 19)	4.4, 4.4.1, 4.4.2, 4.4.4	Οκτώβριος, 2013 - Ιούνιος, 2014
Android 4.4W-4.4W.2 KitKat, with wearable extensions (API level 20)	4.4W, 4.4W.1, 4.4W.2	Ιούνιος, 2014 - Οκτώβριος, 2014
Android 5.0-5.0.2 Lollipop (API level 21)	5.0, 5.0.1, 5.0.2	Νοέμβριος, 2014 - Δεκέμβριος, 2014
Android 5.1-5.1.1 Lollipop (API level 22)	5.1, 5.1.1	Μάρτιος, 2015 - Απρίλιος, 2015



Εικόνα 1: Τα διάφορα λογότυπα των εκδόσεων του android⁵

1.3 Χρήση των διαφορετικών εκδόσεων

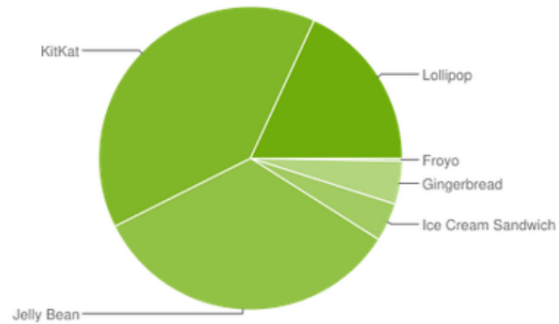
Η αγορά κινητών συσκευών ανανεώνεται συνεχώς και με γρήγορους ρυθμούς. Όπως είναι αναμενόμενο, οι καινούριες συσκευές τείνουν να έχουν εγκατεστημένη την τελευταία διαθέσιμη έκδοση του android και για τον λόγο αυτό οι πρώτες εκδόσεις αρχίζουν να αποσύρονται και να μη χρησιμοποιούνται πλέον.

Η Εικόνα 2 παρουσιάζει τον αριθμό των συσκευών που διαθέτουν μια δεδομένη έκδοση του android, σύμφωνα με έρευνα που έγινε από την Google, τον Αύγουστο

⁵ <http://www.eduardotortosamartin.com/todas-las-versiones-de-android>

του 2015. Τα δεδομένα αυτά πάρθηκαν από το Google Play Store σύμφωνα με την έκδοση των συσκευών που επισκέφτηκε το Store σε ένα διάστημα 7 ημερών. Όπως βλέπουμε, η έκδοση με το μεγαλύτερο ποσοστό χρήσης είναι η 4.4, ενώ εκδόσεις προηγούμενες της 4.1 έχουν μικρό έως και καθόλου ποσοστό χρήσης.

Version	Codename	API	Distribution
2.2	Froyo	8	0.3%
2.3.3 - 2.3.7	Gingerbread	10	4.6%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	4.1%
4.1.x	Jelly Bean	16	13.0%
4.2.x		17	15.9%
4.3		18	4.7%
4.4	KitKat	19	39.3%
5.0	Lollipop	21	15.5%
5.1		22	2.6%



Data collected during a 7-day period ending on August 3, 2015.
Any versions with less than 0.1% distribution are not shown.

Εικόνα 2: Τα ποσοστά χρήσης των διάφορων εκδόσεων του android⁶

1.4 Ανταγωνισμός

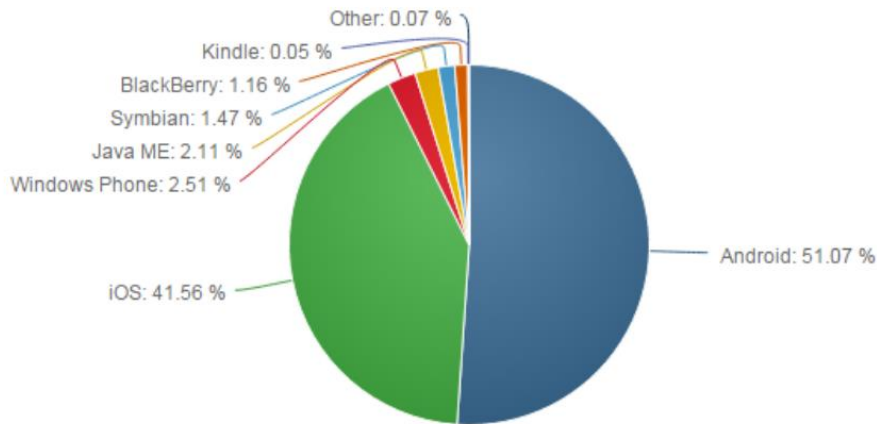
Σήμερα, υπάρχουν στην αγορά περισσότερα από 10 λειτουργικά συστήματα κινητών συσκευών και για τον λόγο αυτό είναι προφανής ο ανταγωνισμός που υπάρχει μεταξύ τους. Αυτός ο ανταγωνισμός είναι πιο εμφανής βέβαια μεταξύ των δημοφιλών λειτουργικών συστημάτων android και iOS.

Το λειτουργικό σύστημα iOS της Apple βγήκε στην αγορά 1 χρόνο πριν την εμφάνιση του android και έτσι είχε στην αρχή μεγαλύτερη απήχηση στην αγορά. Η εμφάνισή του θεωρείται ότι σηματοδότησε την αρχή της μετά-υπολογιστών εποχής. Παρόλα αυτά, με την άφιξη του android, εμφανίστηκαν στην αγορά έξυπνα κινητά τηλέφωνα με κόστος σημαντικά χαμηλότερο από αυτό των κινητών iPhone και έτσι άρχισαν να γίνονται και αυτά δημοφιλή.

Σήμερα, οι προγραμματιστές στοχεύουν στην δημοσίευση εφαρμογών τόσο στο Google Play Store όσο και στο Apple Store, έτσι ώστε να καλύψουν μεγαλύτερο

⁶ <https://developer.android.com/about/dashboards/index.html>

αγοραστικό κοινό. Τα δύο αυτά stores μετρούν το 2015 πάνω από 1,5 εκατομμύριο εφαρμογές, αριθμός που αυξάνεται καθημερινά. Η Εικόνα 3 δείχνει το ποσοστό χρήσης των διάφορων κινητών λειτουργικών συστημάτων για το πρώτο εξάμηνο του 2015, όπως αυτό παρουσιάζεται στην ιστοσελίδα <http://www.netmarketshare.com/>.



Εικόνα 3: Τα ποσοστά χρήσης των διάφορων κινητών λειτουργικών συστημάτων για το πρώτο εξάμηνο του 2015⁷

Μπορούμε να συμπεράνουμε πλέον, ότι το android και το iOS κατέχουν την πρωταρχική θέση στην προτίμηση των αγοραστών.

Ο ανταγωνισμός αυτός έχει θετικά αποτελέσματα για τους χρήστες των λειτουργικών αυτών συστημάτων καθώς οι δύο εταιρείες προσπαθούν να εκλεπτίσουν συνεχώς τα προϊόντα τους, προσθέτοντας νέα χαρακτηριστικά και λειτουργίες και διορθώνοντας σφάλματα στις διάφορες εκδόσεις τους.

⁷ <http://www.netmarketshare.com/>

Κεφάλαιο 2: Ο server

Ο server που χρησιμοποιήθηκε για την εφαρμογή αυτή τρέχει το λειτουργικό σύστημα Linux και χρησιμοποιεί την έκδοση 2.2 του Apache.

Στον server αυτό βρίσκεται εγκατεστημένη η MySQL βάση δεδομένων, για την οποία γίνεται αναφορά στο επόμενο κεφάλαιο. Επίσης, αποθηκεύει όλες τις φωτογραφίες των σημείων που χρειάζεται η εφαρμογή καθώς και τα διάφορα scripts για την επικοινωνία της εφαρμογής με τη βάση δεδομένων.

Κεφάλαιο 3: Βάση δεδομένων MySQL

Η εφαρμογή αυτή βασίζεται εξ ολοκλήρου στην ανάκτηση δεδομένων από μια βάση δεδομένων. Έτσι λοιπόν, για την αποθήκευση των δεδομένων αυτών που απαιτεί η εφαρμογή, χρησιμοποίησα μια βάση δεδομένων MySQL, η οποία είναι εγκατεστημένη στον Linux server που ανέφερα στο προηγούμενο κεφάλαιο. Η έκδοση της βάσης αυτής είναι η 5.5.42. Λεπτομέρειες για την έκδοση αυτή, όπως για παράδειγμα τις διαφορές της με την αμέσως προηγούμενη έκδοση της MySQL, μπορεί να βρει ο αναγνώστης στον παρακάτω σύνδεσμο: <http://dev.mysql.com/doc/relnotes/mysql/5.5/en/news-5-5-42.html>.

3.1 Πίνακες και δεδομένα στην βάση

Η βάση δεδομένων, που έχει αποθηκευμένα τα διάφορα σημεία εξυπηρέτησης-ενδιαφέροντος καθώς και άλλες πληροφορίες που σχετίζονται με αυτά, απαρτίζεται συνολικά από 8 πίνακες, για καθένα από τους οποίους θα κάνω μια σύντομη αναφορά παρακάτω. Σημειώνεται ότι, σε κάθε πίνακα η στήλη id αποτελεί το κύριο κλειδί του πίνακα και είναι ο μοναδικός, αύξων, ακέραιος αριθμός κάθε καταγραφής (record) στον αντίστοιχο πίνακα.

Ο κύριος πίνακας της βάσης ονομάζεται interest_point. Στον πίνακα αυτό αποθηκεύονται τα διάφορα σημεία εξυπηρέτησης-ενδιαφέροντος. Ο πίνακας αυτός απαρτίζεται από τις εξής στήλες: id, name, description και picture_name. Στην στήλη name αποθηκεύεται το όνομα του κάθε σημείου. Η στήλη description εκφράζει με έναν ακέραιο αριθμό τον τύπο του σημείου, για παράδειγμα τράπεζα, και κάνει αναφορά στην στήλη id (description → id) του πίνακα point_type, τον οποίο θα αναλύσω παρακάτω. Τέλος, στην στήλη picture_name αποθηκεύεται το όνομα της φωτογραφίας του κάθε σημείου, όπως αυτό βρίσκεται αποθηκευμένο στον server, για παράδειγμα picture.jpg. Στον πίνακα interest_point μόνο η στήλη picture_name μπορεί να είναι κενή (null).

Στην συνέχεια έχουμε τον πίνακα branch. Ο πίνακας αυτός περιγράφει ένα «υποκατάστημα» ενός σημείου του πίνακα interest_point. Στην συνέχεια της πτυχιακής, αυτό θα αναφέρεται ως σημείο. Απαρτίζεται από τις εξής στήλες: id, point_id, province, street, telephone, latitude και longitude. Η στήλη point_id κάνει αναφορά στο id του σημείου (point_id → id) του πίνακα interest_point. Υποδηλώνει δηλαδή, ότι το συγκεκριμένο σημείο είναι κατάστημα του σημείου με

id ίσο με point_id. Η στήλη province κάνει αναφορά στο id (province → id) του πίνακα province, ο οποίος αναφέρεται παρακάτω και υποδεικνύει την περιοχή του νομού Θεσσαλονίκης στην οποία βρίσκεται το συγκεκριμένο κατάστημα. Η στήλη street περιέχει αποθηκευμένη τη διεύθυνση του καταστήματος, ενώ η στήλη telephone, το τηλέφωνό του. Τέλος, οι στήλες latitude και longitude περιέχουν το γεωγραφικό πλάτος και μήκος του αντίστοιχα.

Ακολουθούν οι πίνακες point_type και province, οι οποίοι είναι πανομοιότυποι σε ό,τι αφορά το σχήμα τους, με την μόνη διαφορά στο όνομα των στηλών. Ο πίνακας point_type περιέχει της στήλες id και type, ενώ ο πίνακας province περιέχει τις στήλες id και province. Η στήλη type περιέχει τον τύπο ενός σημείου ενώ η στήλη name περιέχει έναν δήμο του νομού Θεσσαλονίκης.

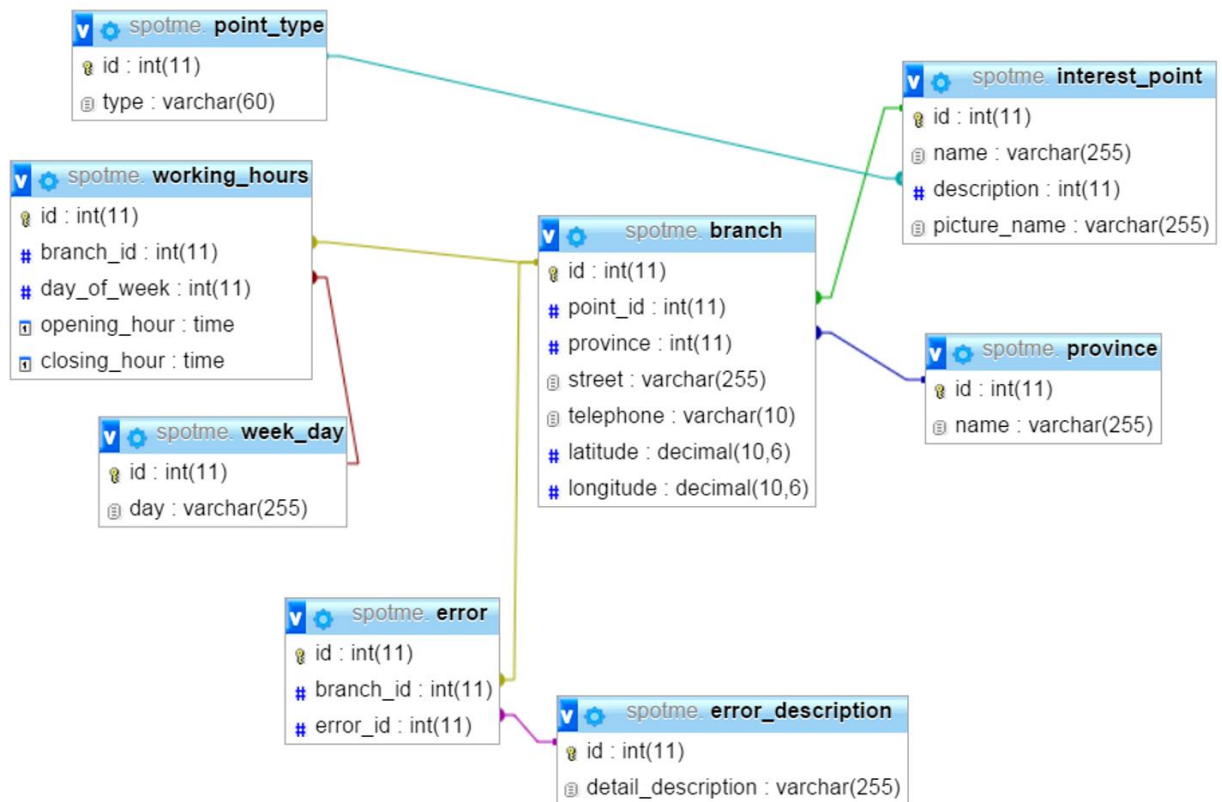
Ο πίνακας week_day περιγράφει τις ημέρες της εβδομάδας και περιέχει 2 στήλες, την στήλη id και την στήλη day. Οι μέρες της εβδομάδες αποθηκεύονται στην στήλη day.

Ο πίνακας working_hours περιγράφει τις ώρες λειτουργίας ενός καταστήματος. Περιλαμβάνει τις εξής στήλες: id, branch_id, day_of_week, opening_hour και closing_hour. Η στήλη branch_id κάνει αναφορά στο id (branch_id → id) του πίνακα branch και υποδηλώνει το κατάστημα στο οποίο αναφέρεται η συγκεκριμένη καταγραφή. Η στήλη day_of_week κάνει αναφορά στο id (day_of_week → id) του πίνακα week_day και περιγράφει την μέρα λειτουργίας του καταστήματος. Τέλος, οι στήλες opening_hour και closing_hour περιέχουν τις ώρες ανοίγματος και κλεισίματος του καταστήματος αντίστοιχα για την συγκεκριμένη μέρα.

Τέλος, έχουμε τους πίνακες error_description και error. Ο πίνακας error_description περιλαμβάνει τις λεπτομέρειες ενός καταστήματος, τις οποίες ο χρήστης μπορεί να αναφέρει ως εσφαλμένες. Περιέχει τις στήλες id και detail_description. Η τελευταία, περιέχει την προαναφέρουσα λεπτομέρεια. Ο πίνακας error περιγράφει μια αναφορά σφάλματος που έχει υποβάλλει κάποιος χρήστης για ένα συγκεκριμένο κατάστημα, όπως επίσης και την λεπτομέρεια του καταστήματος που θεωρεί εσφαλμένη. Περιέχει τις εξής στήλες: id, branch_id και error_id. Η στήλη branch_id κάνει αναφορά στο id (branch_id → id) του

καταστήματος του πίνακα branch και το error_id κάνει αναφορά στο id (error_id → id) του πίνακα error_description.

Αυτοί οι 8 πίνακες που μόλις ανέλυσα, αποτελούν την πηγή δεδομένων της εφαρμογής και παρουσιάζονται γραφικά με τις μεταξύ τους σχέσεις στην Εικόνα 4, όπως αυτή εξάχθηκε από το phpMyAdmin. Ο χρήστης της εφαρμογής δεν μπορεί να τροποποιήσει ήδη υπάρχοντα δεδομένα στην βάση, παρά μόνο να προσθέσει δεδομένα, έμμεσα, μέσω μιας λειτουργίας στην εφαρμογή, στον πίνακα error.



Εικόνα 4: Η γραφική αναπαράσταση των πινάκων της βάσης δεδομένων και των μεταξύ τους σχέσεων

3.2 Το script δημιουργίας της βάσης

Για να δημιουργήσει κάποιος την βάση δεδομένων από την αρχή, θα πρέπει να τρέξει το παρακάτω script στο phpMyAdmin, έτσι ώστε να δημιουργηθούν οι αρχικοί πίνακες που αναλύθηκαν παραπάνω. Ακόμα, στο script αυτό υπάρχουν αρκετά insert statements, ώστε ο χρήστης να μπορεί να τρέξει την εφαρμογή με τον ελάχιστο δυνατό αριθμό δεδομένων.

```
CREATE TABLE IF NOT EXISTS province (  
    id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    name VARCHAR(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 ;  
  
CREATE TABLE IF NOT EXISTS point_type (  
    id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    type VARCHAR(60) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 ;  
  
CREATE TABLE IF NOT EXISTS interest_point (  
    id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    name VARCHAR(255) NOT NULL,  
    description INT NOT NULL,  
    picture_name VARCHAR(255) DEFAULT NULL,  
    FOREIGN KEY (description) REFERENCES point_type (id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 ;  
  
CREATE TABLE IF NOT EXISTS branch (  
    id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    point_id INT NOT NULL,  
    province INT NOT NULL,  
    street VARCHAR(255) NOT NULL,  
    telephone VARCHAR(10),  
    latitude DECIMAL(10,6) NOT NULL,  
    longitude DECIMAL(10,6) NOT NULL,  
    FOREIGN KEY (point_id) REFERENCES interest_point (id),  
    FOREIGN KEY (province) REFERENCES province (id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 ;  
  
CREATE TABLE IF NOT EXISTS week_day (  
    id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    day VARCHAR(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 ;  
  
CREATE TABLE IF NOT EXISTS working_hours (  
    id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    branch_id INT NOT NULL,  
    day_of_week INT NOT NULL,  
    opening_hour TIME,  
    closing_hour TIME,  
    FOREIGN KEY (branch_id) REFERENCES branch (id),  
    FOREIGN KEY (day_of_week) REFERENCES week_day (id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 ;  
  
CREATE TABLE IF NOT EXISTS error_description (  
    id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    detail_description VARCHAR(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 ;  
  
CREATE TABLE IF NOT EXISTS error (  
    id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    branch_id INT NOT NULL,  
    error_id INT NOT NULL,  
    FOREIGN KEY (branch_id) REFERENCES branch (id),  
    FOREIGN KEY (error_id) REFERENCES error_description (id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 ;
```

Απόσπασμα κώδικα 1: Η δημιουργία των πινάκων στη βάση

Πτυχιακή εργασία της φοιτήτριας Σισκάκη Ξένιας

```
INSERT INTO point_type (type) VALUES
('Βενζινάδικο'), ('Εστιατόριο'), ('Καφετέρια'), ('Κινηματογράφος'),
('Μουσείο'), ('Μπαρ-Ποτά'), ('Νοσοκομείο'), ('Ξενοδοχείο'),
('Σουπερμάρκετ'), ('Τράπεζα'), ('Φαρμακείο');

INSERT INTO province (name) VALUES
('Αμπελόκηποι-Μενεμένη'), ('Δήμος Δέλτα'), ('Δήμος Θερμαϊκού'),
('Δήμος Θέρμης'), ('Καλαμαριά'), ('Κορδελιό-Ευόσμος'),
('Δήμος Λαγκαδά'), ('Νεάπολη-Συκιές'), ('Δήμος Παύλου Μελά'),
('Πυλαία-Χορτιάτης'), ('Δήμος Χαλκηδόνας'), ('Ωραιόκαστρο'),
('Δήμος Θεσσαλονίκης');

INSERT INTO week_day (day) VALUES
('SUNDAY'), ('MONDAY'), ('TUESDAY'), ('WEDNESDAY'), ('THURSDAY'),
('FRIDAY'), ('SATURDAY');

INSERT INTO error_description (detail_description) VALUES
('Street'), ('Province'), ('Telephone'), ('Coordinates'),
('Timetable');

INSERT INTO interest_point (name, description, picture_name) VALUES
('Τράπεζα Πειραιώς', 10, 'peiraios.png');

INSERT INTO branch (point_id, province, street, telephone, latitude,
longitude) VALUES
(1, 1, '28ης Οκτωβρίου 131', '2316002430', '40.658233', '22.925208'),
(1, 2, 'Κωνσταντίνου Καραμανλή 30', '2310289500', '40.684169',
'22.859242'),
(1, 3, '25ης Μαρτίου 10', '2392048920', '40.426716', '22.928331'),
(1, 4, 'Φ. Πατίκα 12', '2396026205', '40.479551', '23.135892'),
(1, 5, 'Σμύρνης & Πλ. Ύδρας 3', '2310460631', '40.572162',
'22.957600'),
(1, 6, 'Α. Παπανδρέου & Μ. Αλεξάνδρου 1 & Μπουμπουλίνας 2',
'2310558050', '40.667546', '22.894691'),
(1, 7, 'Παύλου Μελά 17', '2393033050', '40.572484', '23.288366'),
(1, 8, 'Ελ. Βενιζέλου 30 & Νεοχωρίου', '2310988202', '40.649709',
'22.936956'),
(1, 9, 'Σίνα & Μακρυγιάννη 58', '2316002040', '40.673186',
'22.928925'),
(1, 10, 'Κομνηνών 33', '2310331430', '40.591042', '23.037011'),
(1, 11, 'Βασ. Όλγας 281', '2316002170', '40.716345', '22.733297'),
(1, 12, 'Λεωφόρος Δημοκρατίας & Ποσειδώνος', '2310809000',
'40.730761', '22.918526'),
(1, 13, 'Αγ. Δημητρίου 77', '2310254860', '40.639428', '22.946216');
```

Απόσπασμα κώδικα 2: Η εισαγωγή των απαραίτητων δεδομένων στους πίνακες

Κεφάλαιο 4: Η γλώσσα προγραμματισμού PHP

Η γλώσσα προγραμματισμού PHP είναι μια server-side scripting γλώσσα, δηλαδή βοηθάει στην επικοινωνία του server με τον client. Η PHP αναπτύχθηκε το 1994 και μέχρι σήμερα θεωρείται μία από τις πιο πολυχρησιμοποιημένες side-scripting γλώσσες προγραμματισμού, σύμφωνα με μια έρευνα που διεξήχθη το 2013⁸. Στην εφαρμογή αυτή χρησιμοποιείται η έκδοση 5.3.27 της PHP για την επικοινωνία με τον server. Για περισσότερες πληροφορίες, δίνεται ο παρακάτω σύνδεσμος: http://php.net/releases/5_3_27.php.

4.1 Επικοινωνία με τον server

Για την ανάκτηση των δεδομένων, που ανέφερα στο προηγούμενο κεφάλαιο, από την βάση δεδομένων που βρίσκεται εγκατεστημένη στον server, η εφαρμογή εκτελεί αιτήματα HTTP προς τον server. Στην συνέχεια, ο server εκτελεί το αιτούμενο PHP script, το οποίο επιστρέφει τα επιθυμητά αποτελέσματα από την βάση δεδομένων.

4.1.1 Η μορφοποίηση JSON

Τα PHP scripts εμφανίζουν τα αποτελέσματα στην μορφή JSON. Το JSON, ονομασία που προέρχεται από τον αγγλικό όρο **JavaScript Object Notation**, είναι ένας τρόπος απεικόνισης αντικειμένων που αποτελούνται από ζευγάρια κλειδιού-τιμής (key-value pairs) και είναι εύκολα αναγνώσιμος από τον άνθρωπο. Ένα παράδειγμα ενός JSON αντικειμένου αποκονίζεται στο παρακάτω απόσπασμα. Το αντικείμενο αυτό περιγράφει ένα φανταστικό σημείο ενδιαφέροντος-εξυπηρέτησης.

⁸ <http://news.netcraft.com/archives/2013/01/31/php-just-grows-grows.html>

```
{
  "branch_name": "Τράπεζα Πειραιώς",
  "περιοχή": "Θεσσαλονίκη",
  "υποκαταστήματα": [
    {
      "οδός": "Εγνατίας 123",
      "τηλέφωνο": "23920345678",
      "έχει_ATM": true
    }
  ]
}
```

Απόσπασμα κώδικα 3: Παράδειγμα ενός αντικειμένου JSON

Το αντικείμενο JSON που διαβάζει η εφαρμογή μέσω των HTTP requests, αναλύεται στην συνέχεια, γίνεται δηλαδή το λεγόμενο parsing από τον προγραμματιστή, έτσι ώστε ο τελευταίος να εξαγάγει τις επιθυμητές τιμές του αντικειμένου.

4.1.2 PHP scripts

Η εφαρμογή επικοινωνεί με δύο διαφορετικά scripts, τα οποία βρίσκονται αποθηκευμένα στον server, καθ'όλο τον κύκλο εκτέλεσής της. Τα scripts αυτά έχουν μοναδικό σκοπό την σύνδεση με την βάση δεδομένων και την ανάκτηση των επιθυμητών δεδομένων ανάλογα με το αίτημα του χρήστη. Το κάθε αίτημα χαρακτηρίζεται από ένα queryId, το οποίο στέλνεται ως GET παράμετρος στο PHP script και ανάλογα με αυτό, στέλνονται και οι κατάλληλες παράμετροι που χρειάζονται για το εκάστοτε αίτημα στην βάση δεδομένων. Στόν Πίνακα 2 παρουσιάζονται οι διαφορετικές τιμές του queryId, η λειτουργία που εκτελείται ανάλογα με αυτό και οι GET παράμετροι που χρειάζονται ώστε να εκτελεστεί σωστά το αίτημα στην βάση δεδομένων.

Πίνακας 2: Οι παράμετροι που χρειάζονται για τα διάφορα queries στη βάση δεδομένων και η λειτουργία που εκτελεί το καθένα

queryId	Λειτουργία	Παράμετροι
1	Ανάκτηση όλων των καταστημάτων ενός συγκεκριμένου τύπου	pointType
2	Ανάκτηση όλων των καταστημάτων ενός συγκεκριμένου τύπου σε έναν συγκεκριμένο δήμο	provincelId, pointType
3	Ανάκτηση του ωρολόγιου προγράμματος ενός καταστήματος	branchId
4	Εισαγωγή στοιχείων που ο χρήστης θεωρεί λανθασμένα για ένα συγκεκριμένο σημείο	values

Όλες οι πιθανές τιμές που μπορούν να επιστρέψουν τα PHP scripts, είτε αυτές είναι δεδομένα από την βάση, είτε λάθη που μπορούν να συμβούν κατά την εκτέλεση του script, επιστρέφονται σε ένα μοναδικό πίνακα στην μορφή JSON. Αυτό βοηθάει στο να υπάρχει μια ομοιομορφία σε όλα τα HTTP responses που λαμβάνει η εφαρμογή και να μπορεί να τα αναλύσει όλα με τον ίδιο τρόπο. Στην εφαρμογή αυτή, τα κλειδιά στο αντικείμενο JSON είναι οι ονομασίες των στηλών των πινάκων που επιστρέφονται από τα MySQL queries, ενώ οι τιμές των κλειδίων αυτών είναι οι αντίστοιχες τιμές των στηλών στην βάση δεδομένων. Η ανάλυση των απαντήσεων στην πλευρά του χρήστη γίνεται με τον παρακάτω τρόπο. Δημιουργούμε ένα αντικείμενο τύπου JSONArray, το οποίο αρχικοποιούμε με την απάντηση που λάβαμε από τον server, η οποία, όπως είπαμε, είναι ένας μοναδικός πίνακας του τύπου JSONArray. Στην συνέχεια, και μόνο στην περίπτωση που ο πίνακας περιέχει τιμές, δηλαδή έχει μήκος μεγαλύτερο του 0, εκτελούμε έναν βρόγχο επανάληψης για όλες τις τιμές του πίνακα. Κάθε τιμή είναι του τύπου JSONObject. Έτσι, αφού εξάγουμε αυτά τα αντικείμενα, μπορούμε πλέον να διαβάσουμε την τιμή που περιέχει ένα συγκεκριμένο κλειδί του. Αυτό φυσικά σημαίνει ότι ο προγραμματιστής πρέπει να γνωρίζει τα κλειδιά που περιέχει το εκάστοτε αντικείμενο αλλά και τον τύπο αυτών. Στο απόσπασμα 2, παρουσιάζεται ένα απλό παράδειγμα στην Java ανάλυσης ενός αντικειμένου JSON το οποίο είναι παρόμοιο με τα αντικείμενα που λαμβάνει η ίδια η εφαρμογή. Σημειώνεται, ότι τα αντικείμενα στην εφαρμογή δεν επιστρέφονται στην μορφή

αλφαριθμητικού (string), αλλά χρησιμοποιείται η μορφή αυτή για την αρχικοποίηση του JSONArray στο παράδειγμα για να γίνει κατανοητό αυτό προς τον χρήστη.

```
String jsonArrayResponse = "[{\"branchName\": \"Alpha Bank\", \"street\":  
\\\"Βενιζέλου 14\\\", \"province\": 1, \"telephone\": \"2310547892\", \"latitude\":  
40.23456, \"longitude\": 22.12345, \"picture_name\": \"alphaBank.png\"}]\"";  
JSONArray dbReturnedValue = new JSONArray(jsonArrayResponse);  
if (dbReturnedValue.length() > 0) {  
    for (int i = 0; i < dbReturnedValue.length(); i++) {  
        JSONObject object = dbReturnedValue.getJSONObject(i);  
        String branchName = object.getString(\"branchName\");  
        String street = object.getString(\"street\");  
        int province = object.getInt(\"province\");  
        String telephone = object.getString(\"telephone\");  
        double latitude = object.getDouble(\"latitude\");  
        double longitude = object.getDouble(\"longitude\");  
        String picture_name = object.getString(\"picture_name\");  
    }  
}
```

Απόσπασμα κώδικα 4: Παράδειγμα ανάλυσης(parsing) ενός JSONArray

Κεφάλαιο 5: Ανάπτυξη εφαρμογής

Για την ανάπτυξη οποιασδήποτε εφαρμογής, ένας προγραμματιστής πρέπει να διαλέξει τα κατάλληλα για αυτόν εργαλεία έτσι ώστε να παράγει όσο το δυνατόν πιο αποδοτική εφαρμογή γίνεται. Στο κεφάλαιο αυτό, θα αναφέρω συνοπτικά τα εργαλεία που έλαβα υπόψη κατά την ανάπτυξη της εφαρμογής.

5.1 Eclipse

Για την ανάπτυξη της εφαρμογής αυτής, χρησιμοποιήθηκε το ολοκληρωμένο περιβάλλον ανάπτυξης (**I**ntegrated **D**evelopment **E**nvironment, **IDE**) Eclipse, το οποίο μπορεί να βρει ο αναγνώστης δωρεάν στην επίσημη ιστοσελίδα του⁹. Αυτό το περιβάλλον μπορεί να παραμετροποιηθεί ώστε να χρησιμοποιηθεί για την ανάπτυξη προγραμμάτων σε πολλές γλώσσες προγραμματισμού. Η έκδοση που χρησιμοποιήθηκε είναι η 4.4 (Luna).

5.1.1 Απαιτήσεις για το android

Για να χρησιμοποιηθεί το Eclipse για την ανάπτυξη αυτής της εφαρμογής android, χρειάστηκαν κάποια επιπλέον βήματα έπειτα από την εγκατάσταση του προγράμματος, τα οποία παρουσιάζονται συνοπτικά παρακάτω.

Το Android προφέρει ένα plugin για το Eclipse IDE, το οποίο ονομάζεται **A**ndroid **D**evelopment **T**ools (ADT). Αυτό, προσφέρει ένα ολοκληρωμένο περιβάλλον στον προγραμματιστή για την ανάπτυξη εφαρμογών Android. Για την εγκατάσταση του plugin αυτού, πρέπει να πληρούνται οι παρακάτω προϋποθέσεις:

1. Να υπάρχει εγκατεστημένη κάποια έκδοση του Eclipse μεγαλύτερη ή ίση από την 3.7.2 (Indigo), καθώς προηγούμενες εκδόσεις δεν υποστηρίζουν πλέον την τελευταία έκδοση του ADT.
2. Το εγκατεστημένο Eclipse θα πρέπει να υποστηρίζει το JDT (**J**ava **D**evelopment **T**ools) plugin¹⁰, να υπάρχει δηλαδή δυνατότητα προγραμματισμού σε Java.
3. Να υπάρχει εγκατεστημένη η έκδοση 6 του JDK (**J**ava **D**evelopment **K**it)¹¹.

⁹ <http://www.eclipse.org/>

¹⁰ <http://www.eclipse.org/jdt/>

Εκτός από την απαίτηση του ADT, χρειάζεται ένα ακόμα πρόσθετο, το οποίο ονομάζεται SDK, από την αγγλική ορολογία **Software Development Kit**. Το SDK είναι ένα σετ εργαλείων ανάπτυξης android εφαρμογών και περιέχει τα παρακάτω:

- Απαραίτητες βιβλιοθήκες
- Debugger
- Emulators
- Παραδείγματα πηγαίου κώδικα
- Σχετικό υλικό τεκμηρίωσης (documentation) για τα διάφορα APIs
- Οδηγίες χρήσης του android

Για κάθε καινούρια έκδοση του android δημοσιεύεται και η αντίστοιχη καινούρια έκδοση του SDK. Για να μπορούν οι προγραμματιστές να χρησιμοποιούν τα πιο πρόσφατα χαρακτηριστικά του android, θα πρέπει να έχουν εγκατεστημένη και την πιο πρόσφατη έκδοση του SDK. Τέλος, το SDK μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών και μέσω ενός τερματικού αλλά και μέσω ενός IDE, στην συγκεκριμένη περίπτωση του Eclipse, στο οποίο ενσωματώνεται.

Να σημειωθεί, ότι σύμφωνα με ανακοίνωση στο επίσημο blog του Android development¹², θα σταματήσει στο κοντινό μέλλον η υποστήριξη του android για το Eclipse, αλλά κατά την διάρκεια ανάπτυξης της εφαρμογής και συγγραφής αυτής της πτυχιακής, υποστηρίζεται ακόμα.

5.2 Google Maps

Οι χάρτες που προσφέρει η Google αποτελούν κύριο συστατικό της εφαρμογής αυτής, καθώς υποβοηθούν τον χρήστη στην οπτική προβολή των διάφορων σημείων εξυπηρέτησης-ενδιαφέροντος. Στην εφαρμογή αυτή χρησιμοποιήθηκε η έκδοση 2 των Google Maps, για τους οποίους θα κάνω εκτενέστερη αναφορά στο επόμενο κεφάλαιο.

¹¹ <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

¹² <http://android-developers.blogspot.co.at/2015/06/an-update-on-eclipse-android-developer.html>

5.3 Genymotion

Το Genymotion¹³ είναι ένα πρόγραμμα που παρέχει στους χρήστες προσομοιωτές (emulators) android. Ο χρήστης μπορεί να διαλέξει ανάμεσα σε αρκετούς προκαθορισμένους emulators αλλά και να φτιάξει τον δικό του σύμφωνα με τις απαιτήσεις που χρειάζεται η εκάστοτε εφαρμογή. Οι απαιτήσεις αυτές μπορεί να περιλαμβάνουν την ανάλυση και το μέγεθος της οθόνης της συσκευής, τον αριθμό των επεξεργαστών και το μέγεθος της μνήμης και τέλος, την έκδοση του API.

Σήμερα, υπάρχει στην αγορά μεγάλη πληθώρα κινητών συσκευών, οι οποίες έχουν διαφορετικά χαρακτηριστικά μεταξύ τους. Για τον λόγο αυτό, ο χρήστης πρέπει να στοχεύει στην ανάπτυξη ευέλικτων εφαρμογών. Πιο συγκεκριμένα, δεν θα πρέπει οι εφαρμογές να περιορίζονται από χαρακτηριστικά αυτά. Ένα από τα πιο δημοφιλή είναι το μέγεθος της οθόνης, καθώς υπάρχουν κινητά τηλέφωνα με πολύ μικρές οθόνες αλλά και τηλεοράσεις με πολύ μεγάλες οθόνες.

Συνοψίζοντας, οι διάφοροι emulator είναι σημαντικοί κατά την διάρκεια ανάπτυξης μια εφαρμογής για να δοκιμάσει ο προγραμματιστής την εφαρμογή του σε όλες τις πιθανές συσκευές, στις οποίες μπορεί ορισμένα χαρακτηριστικά να μην λειτουργούν. Για τη δοκιμή της εφαρμογής μου χρησιμοποίησα διάφορους emulator του Genymotion, προτιμώντας τους από τους emulators που παρέχει το ίδιο το SDK, γιατί είναι ταχύτεροι και παρέχουν περισσότερες λειτουργίες.

¹³ <https://www.genymotion.com/>

Κεφάλαιο 6: Google Maps

Το Google Maps είναι ένα έργο που αναπτύχθηκε από την Google το 2008 και αποτελεί μια υπηρεσία χαρτογράφησης στο Διαδίκτυο. Προσφέρει στους χρήστες διαδραστικούς χάρτες μέσα από τους οποίους κάποιος μπορεί να βρει εύκολα και γρήγορα διάφορα σημεία σε όλο τον κόσμο. Μία από τις πιο γνωστές λειτουργίες τους, βέβαια, είναι η εμφάνιση οδηγιών (directions) μέσω GPS από ένα σημείο αφητηρίας σε ένα σημείο προορισμού. Στην συνέχεια, θα αναλύσω την χρήση των Google Maps στην εφαρμογή αυτή.

6.1 Google Maps API v2

Η Google προσφέρει ένα API στους προγραμματιστές μέσω του οποίου οι τελευταίοι μπορούν να εισάγουν τους χάρτες στις διάφορες android εφαρμογές τους. Το API αυτό προσφέρει όλες τις λειτουργίες που προσφέρουν και οι χάρτες όταν αυτοί χρησιμοποιούνται από ένα πρόγραμμα περιήγησης (browser). Πιο συγκεκριμένα, υπάρχει δυνατότητα εμφάνισης στον χάρτη ενδείξεων (markers) για την ανάδειξη διάφορων σημείων, εμφάνιση 3D κτηρίων αλλά και εσωτερικών κατόψεων αυτών, εμφάνιση του χάρτη στις διάφορες μορφές του, όπως για παράδειγμα την δορυφορική, και πολλές άλλες λειτουργίες.

6.1.1 Προϋποθέσεις χρήσης

Για την ενσωμάτωση και την χρήση του Google Maps API σε μια εφαρμογή, ο προγραμματιστής πρέπει να ακολουθήσει ορισμένα βήματα.

Αρχικά, η εφαρμογή που προορίζεται να χρησιμοποιήσει το API αυτό, πρέπει να καταχωριθεί στο Google Developers Concole¹⁴. Εκεί, ο προγραμματιστής θα πρέπει να ενεργοποιήσει την κατάλληλη υπηρεσία που επιθυμεί να χρησιμοποιήσει στην εφαρμογή. Στην συγκεκριμένη περίπτωση, αυτή η υπηρεσία είναι η Google Maps Android API v2. Στην συνέχεια, πρέπει να δηλώσει το όνομα του πακέτου της εφαρμογής του και το SHA-1 certificate fingerprint αυτής. Με τα δεδομένα αυτά, δημιουργείται στην συνέχεια ένα κλειδί, το λεγόμενο android API key, το οποίο είναι μια μεγάλη ακολουθία χαρακτήρων. Το κλειδί αυτό είναι απαραίτητο για την επικοινωνία της εφαρμογής με τους εξυπηρετητές των Google Maps και για τον λόγο αυτό, πρέπει να προστεθεί στο manifest της εφαρμογής,

¹⁴ <https://console.developers.google.com/>

όπως αυτό φαίνεται στο παράδειγμα στην Εικόνα 5. Με το `android:name` προσδιορίζουμε την υπηρεσία στην οποία αναφέρεται το συγκεκριμένο κλειδί. Εδώ, αναφερόμαστε στο Google Maps Android API v2.

```
<meta-data
  android:name="com.google.android.maps.v2.API_KEY"
  android:value="AlzaSyBdVI-cTICSwYKrZ95SuvNw7dbMuDt1KGO"/>
```

Εικόνα 5: Παράδειγμα χρήσης του API key στο manifest της εφαρμογής

Ένα API κλειδί μπορεί να αποτελεί κλειδί για πολλές εφαρμογές, με την προϋπόθεση ότι ο προγραμματιστής τις έχει καταχωρήσει με τον παραπάνω τρόπο. Τέλος, για την χρήση του Google Maps API v2, η εφαρμογή χρειάζεται την εξωτερική βιβλιοθήκη του Google Play Services, η οποία παρέχεται στον χρήστη με την εγκατάσταση του SDK. Για τις διάφορες εξωτερικές βιβλιοθήκες, θα γίνει αναφορά στο επόμενο κεφάλαιο.

6.1.2 Το API

Για να ορίσει κανείς έναν χάρτη στο android πρέπει να δημιουργήσει ένα αντικείμενο του τύπου `GoogleMap`. Αυτό το αντικείμενο προσφέρει αρκετές μεθόδους με σκοπό την παραμετροποίηση του χάρτη. Στην συνέχεια, θα αναφέρω μερικές από τις μεθόδους αυτές και τις λειτουργίες τους.

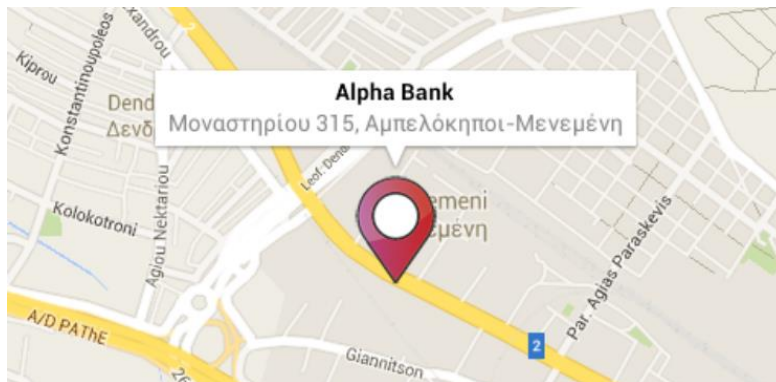
Η μέθοδος `moveCamera(CameraUpdate update)` είναι υπεύθυνη για την εμφάνιση ενός συγκεκριμένου σημείου στον χάρτη και παίρνει ως παράμετρο ένα αντικείμενο τύπου `CameraUpdate`, το οποίο αρχικοποιείται με τις συντεταγμένες του σημείου στο χάρτη και με το ύψος της επιθυμητής μεγένθυσης (`zoom`).

Με την μέθοδο `setMapType(int type)` μπορεί κανείς να ορίσει τον τύπου του εδάφους στον χάρτη μέσω της αέραςιας παραμέτρου `type`. Οι διαθέσιμοι τύποι είναι οι παρακάτω:

- `GoogleMap.MAP_TYPE_NORMAL`
- `GoogleMap.MAP_TYPE_HYBRID`
- `GoogleMap.MAP_TYPE_NONE`
- `GoogleMap.MAP_TYPE_SATELLITE`

- `GoogleMap.MAP_TYPE_TERRAIN`

Μια ακόμα σημαντική μέθοδος είναι η `addMarker(MarkerOptions options)`, η οποία προσθέτει έναν `marker` στον χάρτη στο επιθυμητό σημείο. Η παράμετρος που λαμβάνει είναι ένα αντικείμενο με το οποίο γίνεται η παραμετροποίηση του `marker` αυτού. Πιο συγκεκριμένα, στην παράμετρο `options` μπορεί ο προγραμματιστής να ορίσει τον τίτλο και την περιγραφή του `marker`, ένα εικονίδιο της προτίμησής του και φυσικά της συντεταγμένες του σημείου το οποίο περιγράφει ο `marker` αυτός. Στην Εικόνα 6 φαίνεται ένα παράδειγμα ενός `marker`, το οποίο πάρθηκε από την εφαρμογή.



Εικόνα 6: Ένας `marker` στον χάρτη, όπως αυτό εμφανίζεται στην εφαρμογή.

Τέλος, έχουμε την μέθοδο `addPolyline(PolylineOptions options)`, με την οποία μπορεί ο προγραμματιστής να εμφανίσει διάφορα πολύγγραμμα στον χάρτη, το καθένα από τα οποία ορίζεται από την παράμετρο `options`. Η μέθοδος αυτή συνήθως χρησιμοποιείται για την εμφάνιση οδηγιών στον χάρτη, δηλαδή έχουμε μια γραμμή που ξεκινάει από το σημείο αφετηρίας και φτάνει μέχρι το σημείο προορισμού. Στην παράμετρο `options`, ο προγραμματιστής μπορεί να ορίσει διάφορες επιλογές, όπως για παράδειγμα, το χρώμα και το πάχος της γραμμής καθώς και τα διάφορα σημεία από τα οποία θα περνάει αυτή.

Κεφάλαιο 7: Εξωτερικές βιβλιοθήκες

Αρκετές φορές, ο κώδικας του προγραμματιστή δεν αρκεί για να εκτελέσει η εφαρμογή όλες τις επιθυμητές λειτουργίες. Ενδεχομένως, θα επιθυμούσε να ενσωματώσει στην εφαρμογή μια λειτουργία, ένα API ή και μια άλλη ολόκληρη αυτοτελή εφαρμογή, τα οποία υπάρχουν ήδη υλοποιημένα, είτε από τον ίδιο είτε από κάποιον άλλο προγραμματιστή. Αυτό είναι συνήθως εφικτό με την χρήση εξωτερικών βιβλιοθηκών, από τις οποίες εξαρτάται η εφαρμογή. Η εφαρμογή SpotMe κάνει χρήση τεσσάρων διαφορετικών εξωτερικών βιβλιοθηκών, για τις οποίες γίνεται μια μικρή αναφορά στην συνέχεια.

7.1 Universal Image Loader¹⁵

Η βιβλιοθήκη αυτή αποτελεί ένα δυνατό εργαλείο υποβοήθησης προς τον προγραμματιστή για την ευέλικτη φόρτωση και εμφάνιση φωτογραφιών στο Android. Είναι εύκολα παραμετροποιήσιμο, καθώς παρέχει αρκετές επιλογές για τις διάφορες λειτουργίες. Μερικές από αυτές παρατίθενται στην συνέχεια:

- Multithread φόρτωση εικόνων (σύγχρονα ή ασύγχρονα)
- Δυνατότητα caching των εικόνων στην μνήμη ή την κάρτα μνήμης
- Δυνατότητα παρακολούθησης φόρτωσης της εικόνας

7.1.1 To API

Για να έχει κάποιος πρόσβαση στις διάφορες λειτουργίες της βιβλιοθήκης αυτής, πρέπει να ορίσει και να αρχικοποιήσει ένα αντικείμενο του τύπου `ImageLoader`. Στην συνέχεια, για να παραμετροποιήσει τον τρόπο φόρτωσης των εικόνων, πρέπει να δημιουργήσει ένα αντικείμενο του τύπου `DisplayImageOptions`. Μερικές από τις πιο σημαντικές μεθόδους εξηγούνται στην συνέχεια.

Αρχικά, με την μέθοδο `cacheInMemory(boolean cacheInMemory)` ορίζουμε αν θέλουμε οι φωτογραφίες να γίνονται `cached` στην μνήμη του κινητού.

Τέλος, έχουμε τις μεθόδους `showImageForEmptyUri(int imageRes)`, `showImageOnFail(int imageRes)`, και `showImageOnLoading(int imageRes)` με τις οποίες μπορούμε να ορίσουμε την εικόνα που θα εμφανιστεί σε περίπτωση που ο

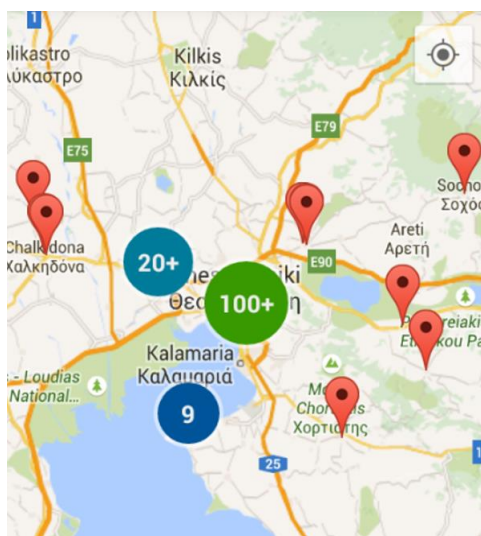
¹⁵ <https://github.com/nostra13/Android-Universal-Image-Loader>

σύνδεσμος της εικόνας είναι άδειος, σε περίπτωση αποτυχίας και κατά την διάρκεια της φόρτωσης αντίστοιχα.

Μετά τον ορισμό των διάφορων επιλογών για την εμφάνιση των εικόνων, πρέπει να καλέσουμε την μέθοδο `displayImage(String uri, ImageView imageView, DisplayImageOptions options)` του αρχικού αντικειμένου `ImageLoader` που δημιουργήσαμε στην αρχή. Η μέθοδος αυτή δέχεται σαν παράμετρο το URI από το οποίο θα φορτωθεί η εικόνα, το `ImageView` στο οποίο θα φορτωθεί η εικόνα και τέλος το αντικείμενο `DisplayImageOptions` που δημιουργήσαμε. Το URI μπορεί να είναι είτε κάποιος σύνδεσμος στο διαδίκτυο, είτε το path της εικόνας στην μνήμη ή την εξωτερική μνήμη του κινητού, είτε μια εικόνα αποθηκευμένη στο android project.

7.2 Google clustering¹⁶

Η βιβλιοθήκη αυτή προσφέρει την δυνατότητα δημιουργίας clusters σε έναν χάρτη ανάλογα με την τοποθεσία των διάφορων markers στον χάρτη. Έτσι, στην περίπτωση που υπάρχουν πάρα πολλά σημεία προς εμφάνιση, με τη βιβλιοθήκη αυτή ο χάρτης γίνεται πιο εύκολα αναγνώσιμος. Ο αριθμός των διάφορων markers που εμφανίζεται σε κάθε cluster, εξαρτάται από το zoom στον χάρτη. Πιο συγκεκριμένα, όσο μεγαλύτερη μεγέθυνση έχουμε, τόσο περισσότερα σημεία εμφανίζονται στον χάρτη. Στην Εικόνα 7 βλέπουμε έναν χάρτη με διάφορα σημεία σε clusters.



Εικόνα 7: Παράδειγμα εμφάνισης των clusters στον χάρτη, όπως αυτό εμφανίζεται στην εφαρμογή

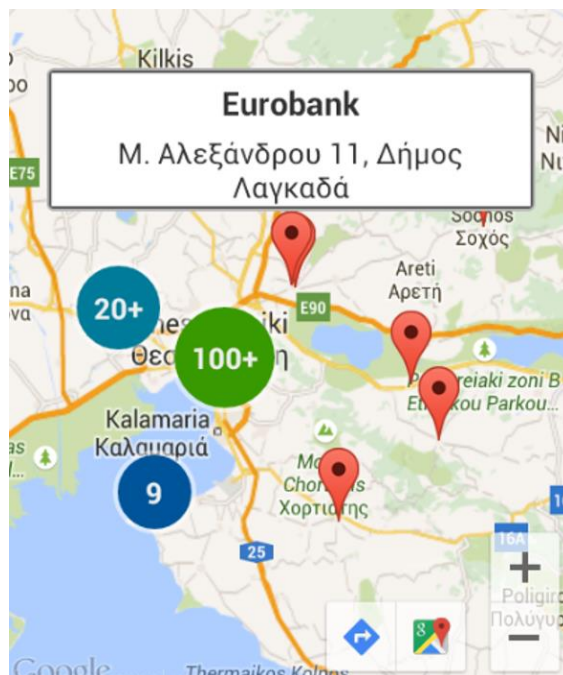
¹⁶ <https://developers.google.com/maps/documentation/android/utility/marker-clustering>

7.2.1 To API

Για να υλοποιήσουμε τα διάφορα clusters στον χάρτη χρειαζόμαστε ένα αντικείμενο τύπου `ClusterManager`, το οποίο είναι υπεύθυνο για την διαδικασία του clustering. Σε αυτό το αντικείμενο προσθέτουμε όλα τα σημεία που θέλουμε να εμφανίσουμε στον χάρτη. Τα σημεία αυτά πρέπει να υλοποιούν τον τύπο `ClusterItem` και να περιέχουν ως υποχρεωτικό πεδίο τις συντεταγμένες στο χάρτη και, επιπλέον, αν επιθυμεί ο προγραμματιστής, διάφορες λεπτομέρειες προς εμφάνιση κατά την επιλογή ενός marker, όπως για παράδειγμα την οδό του σημείου.

Η προσθήκη των διάφορων σημείων στον `ClusterManager` γίνεται μέσω της μεθόδου `addItem(MyItem myItem)`. Η παράμετρος `myItem` αντιπροσωπεύει το κάθε σημείο.

Για να προσθέσουμε λειτουργικότητα κατά την επιλογή ενός marker, πρέπει να υλοποιήσουμε έναν `OnMarkerClickListener()` στον `ClusterManager`. Στην Εικόνα 8, βλέπουμε ένα παράδειγμα εμφάνισης των λεπτομερειών του σημείου κατά την επιλογή του. Ακόμα, υπάρχει η δυνατότητα προσθήκης ενός `OnCameraChangeListener()`, με σκοπό την εκτέλεση διάφορων λειτουργιών όταν αλλάζει θέση η κάμερα στον χάρτη.



Εικόνα 8: Εμφάνιση των λεπτομερειών του σημείου κατά την επιλογή του, όπως αυτό εμφανίζεται στην εφαρμογή

7.3 Google Directions¹⁷

Η βιβλιοθήκη αυτή χρησιμεύει στην εμφάνιση της διαδρομής στο χάρτη από ένα σημείο αφετηρία σε ένα σημείο προορισμού. Η βιβλιοθήκη χρησιμοποιεί το Google Directions API¹⁸.

7.3.1 To API

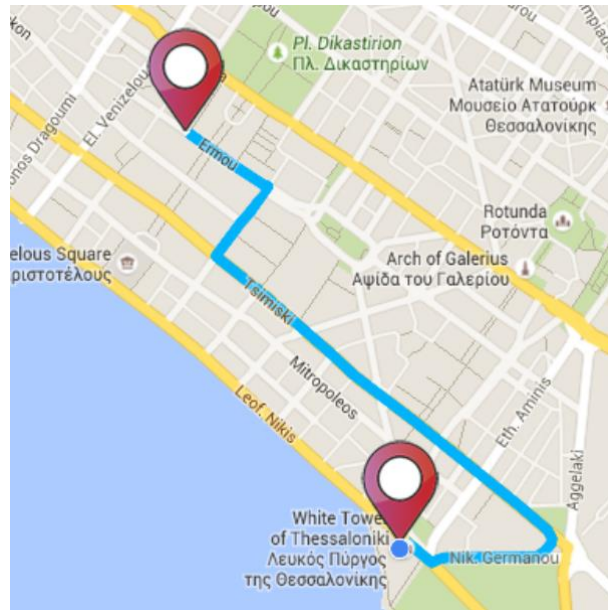
Η διαδρομή ορίζεται από ένα αντικείμενο του τύπου `Routing`, στο οποίο μπορούμε να ορίσουμε όλες τις επιθυμητές επιλογές για την εμφάνιση της διαδρομής. Μία από αυτές τις επιλογές είναι η εμφάνιση διαφορετικών διαδρομών ανάλογα με τον επιθυμητό τύπο, ο οποίος χρησιμοποιείται για την αρχικοποίηση του αντικειμένου. Οι τύποι αυτοί είναι οι εξής:

- `Routing.TravelMode.DRIVING`
- `Routing.TravelMode.BIKING`
- `Routing.TravelMode.WALKING`

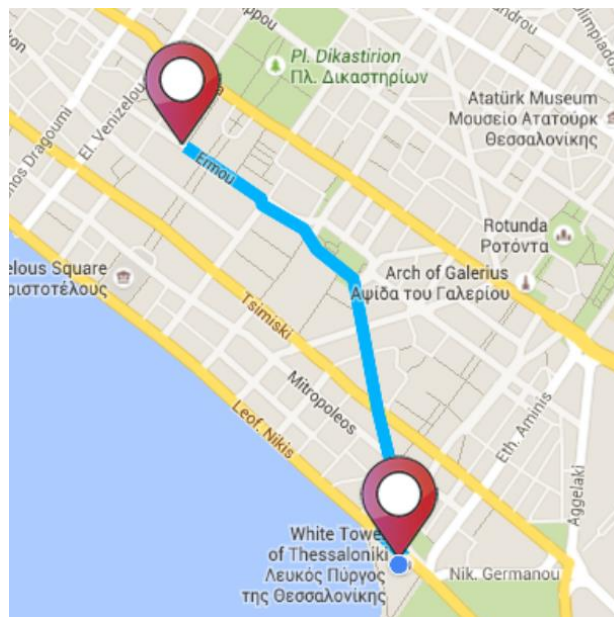
Με την μέθοδο `execute(LatLng... params)` μπορούμε να εμφανίσουμε τη διαδρομή στον χάρτη. Η μέθοδος λαμβάνει σαν παράμετρο ένα και παραπάνω (...) αντικείμενα τύπου `LatLng`, στα οποία ορίζουμε τις διάφορες συντεταγμένες των σημείων από τα οποία θα περνάει η διαδρομή. Στην εφαρμογή *spotMe* χρησιμοποιούμε μόνο 2 σημεία, ένα για την τωρινή τοποθεσία του χρήστη, η οποία αποτελεί την αφετηρία, και ένα για το σημείο στο οποίο θέλει να φτάσει ο χρήστης, δηλαδή τον προορισμό του. Στην Εικόνα 9, βλέπουμε ένα παράδειγμα εμφάνισης μια διαδρομής με αυτοκίνητο στο χάρτη, ενώ στην Εικόνα 10 βλέπουμε την ίδια διαδρομή με τα πόδια. Παρατηρούμε ότι οι δύο διαδρομές είναι διαφορετικές καθώς το API υπολογίζει την κατάλληλη και συντομότερη διαδρομή ανάλογα με τον κάθε τύπο.

¹⁷ <https://github.com/jd-alexander/Google-Directions-Android>

¹⁸ <https://developers.google.com/maps/documentation/directions/intro>



Εικόνα 9: Εμφάνιση διαδρομής με αυτοκίνητο στο χάρτη, όπως αυτό εμφανίζεται στην εφαρμογή



Εικόνα 10: Εμφάνιση διαδρομής με τα πόδια στο χάρτη, όπως αυτό εμφανίζεται στην εφαρμογή

Στο αντικείμενο Routing που δημιουργήσαμε πρέπει να προσθέσουμε έναν Listener, ο οποίος παρέχει τις παρακάτω μεθόδους:

- `onRoutingFailure()`: καλείται στην περίπτωση που η διαδρομή για κάποιον λόγο δεν μπορεί να υπολογιστεί
- `onRoutingStart()`: καλείται κατά την έναρξη υπολογισμού της διαδρομής
- `onRoutingSuccess(PolylineOptions mpPolyOptions, Route route)`: καλείται στην περίπτωση επιτυχίας υπολογισμού της διαδρομής. Η μέθοδος

αυτή παίρνει σαν παράμετρο ένα αντικείμενο τύπου `PolylineOptions`, στο οποίο, όπως ήδη έχει αναφερθεί, μπορούμε να ορίσουμε διάφορες λεπτομέρειες για την γραμμή στο χάρτη, όπως το χρώμα της. Τέλος, σαν 2^η παράμετρο έχουμε ένα αντικείμενο τύπου `Route`, το οποίο κρατάει μια αναφορά στην υπολογισμένη διαδρομή.

7.4 Google play services¹⁹

Με τη χρήση της βιβλιοθήκης αυτής, ο προγραμματιστής επωφελείται από τις πιο πρόσφατες υπηρεσίες που προσφέρει η Google, όπως για παράδειγμα το Google Maps και το Google+. Η βιβλιοθήκη αυτή είναι απαραίτητη για την χρήση των Google Maps και αποτελεί απαραίτητη αναφορά για τις βιβλιοθήκες Google Directions και Google Clustering, καθώς και οι δύο αυτές βιβλιοθήκες κάνουν χρήση του Google Maps API. Επίσης, η εφαρμογή *spotMe* πρέπει να κρατάει μια αναφορά σε αυτή τη βιβλιοθήκη καθώς χρησιμοποιεί και η ίδια τους χάρτες.

¹⁹ <https://developers.google.com/android/guides/overview>

Κεφάλαιο 8: Απαιτήσεις εφαρμογής

Η εφαρμογή έχει κάποιες απαιτήσεις από την πλευρά του χρήστη έτσι ώστε να μπορεί να λειτουργήσει σωστά. Οι απαιτήσεις αυτές αναλύονται στην συνέχεια.

8.1 Internet

Στο μεγαλύτερο μέρος της, η εφαρμογή δεν μπορεί να λειτουργήσει χωρίς να είναι συνδεδεμένη σε κάποιο δίκτυο, είτε αυτό προέρχεται από το 3G δίκτυο της συσκευής ή από το WiFi όταν η συσκευή είναι συνδεδεμένη στο internet μέσω αυτού.

Το internet χρειάζεται κατά κύριο λόγο για να συνδέεται η εφαρμογή με την βάση δεδομένων και να λαμβάνει τα δεδομένα ανάλογα με το εκάστοτε αίτημα του χρήστη. Ακόμα, η εφαρμογή λαμβάνει διάφορες εικόνες για τα σημεία, οι οποίες βρίσκονται αποθηκευμένες στον server και έτσι χρειάζεται το internet και για αυτή την λειτουργία.

Στην περίπτωση που ο χρήστης εκτελεί μια λειτουργία στην εφαρμογή, η οποία χρειάζεται πρόσβαση στο internet, αλλά η συσκευή δεν είναι συνδεδεμένη, θα λάβει κατάλληλο μήνυμα για να τον ενημερώσει.

8.2 GPS

Μια ακόμα απαίτηση αποτελεί το GPS (**G**lobal **P**ositioning **S**ystem), το οποίο όμως δεν αποτελεί απαραίτητο στοιχείο για τη λειτουργία της εφαρμογής. Πιο συγκεκριμένα, η χρήση του εξαρτάται από την λειτουργία που θα εκτελέσει ο χρήστης. Χρειάζεται δηλαδή στην περίπτωση που ο χρήστης θέλει να αναζητήσει σημεία εξυπηρέτησης-ενδιαφέροντος σύμφωνα με την τωρινή τοποθεσία του ή στην περίπτωση που θέλει να δει την διαδρομή ενός σημείου πάλι σύμφωνα με την τοποθεσία του.

8.3 Συμβατότητα συσκευών

Στην συνέχεια θα αναλύσω ποιες συσκευές είναι συμβατές με την εφαρμογή *spotMe*.

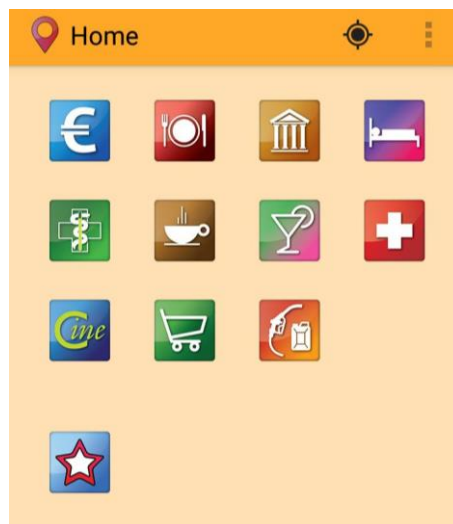
8.3.1 Έκδοση android

Η εφαρμογή ορίζει ότι, για να την χρησιμοποιήσει μια συσκευή, πρέπει η τελευταία να διαθέτει API 13 και πάνω, δηλαδή να υπάρχει εγκατεστημένη η έκδοση 3.2 και

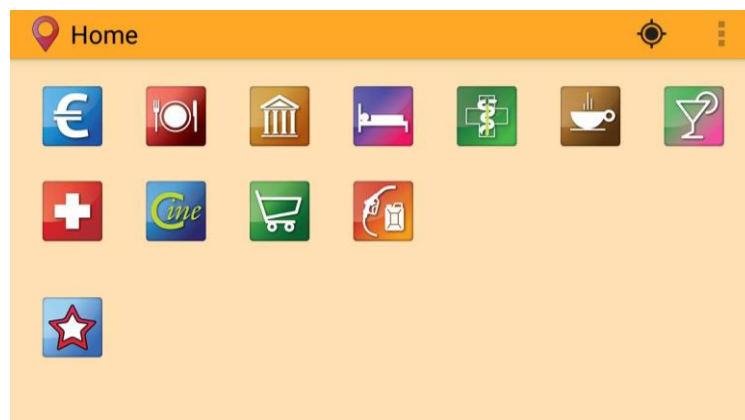
πάνω του android. Η επιλογή της έκδοσης αυτής, έγινε σύμφωνα με τα ποσοστά χρήσης των διάφορων εκδόσεων του android που φαίνονται στην Εικόνα 2 του Κεφαλαίου 1. Μικρότερες εκδόσεις χρησιμοποιούνται πλέον σε πολύ μικρό ποσοστό και για το λόγο αυτό, αποφάσισα να τις εξαιρέσω.

8.3.2 Οθόνες συσκευών

Η εφαρμογή ορίζει ότι υποστηρίζει από μικρές μέχρι και πολύ μεγάλες οθόνες συσκευών, σε ό,τι αφορά το μέγεθος, και οθόνες όλων των πυκνοτήτων pixels. Ακόμα, υποστηρίζονται και οι δύο προσανατολισμοί των συσκευών, κάθετος και οριζόντιος. Στην Εικόνα 10 βλέπουμε την αρχική οθόνη της εφαρμογής στον κάθετο προσανατολισμό, ενώ στην Εικόνα 11 την ίδια οθόνη στον οριζόντιο προσανατολισμό. Οι δύο οθόνες αλλάζουν αυτόματα, καθώς στην εφαρμογή έχουν οριστεί διαφορετικά layouts τόσο για το μέγεθος της οθόνης όσο και για τον προσανατολισμό της.



Εικόνα 11: Η αρχική οθόνη της εφαρμογής σε κάθετο προσανατολισμό



Εικόνα 12: Η αρχική οθόνη της εφαρμογής σε οριζόντιο προσανατολισμό

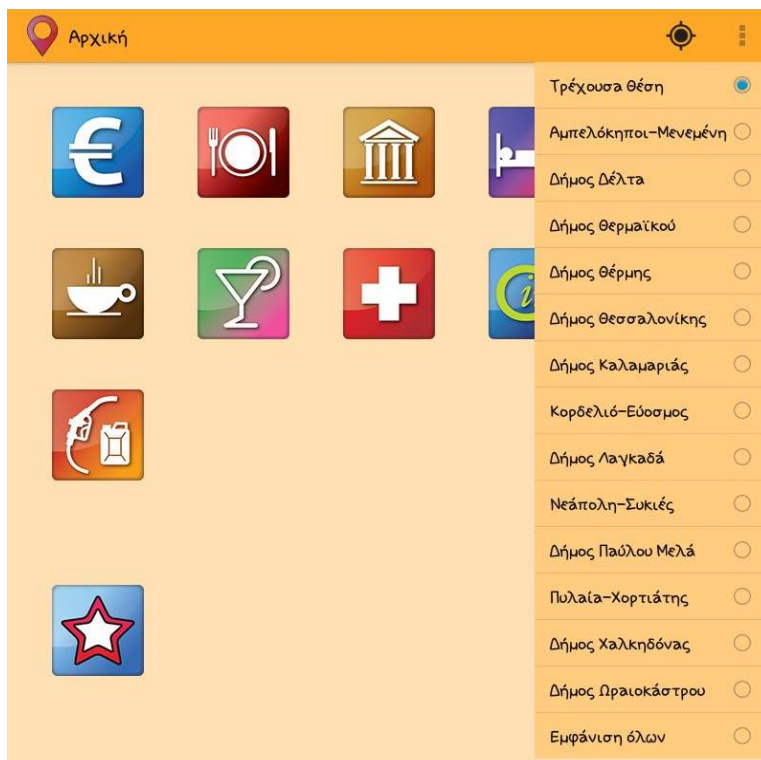
Κεφάλαιο 9: Οδηγός χρήσης λογισμικού

Στο κεφάλαιο αυτό θα γίνει εκτενής αναφορά στον τρόπο χρήσης της εφαρμογής. Θα παραθέσω εικόνες από τις διάφορες οθόνες της εφαρμογής, οι οποίες εμφανίζονται σε κάθετο προσανατολισμό και προέρχονται από tablet 10 ιντσών. Αρχικά, στην Εικόνα 13 βλέπουμε το λογότυπο της εφαρμογής, που εμφανίζεται κάθε φορά που ανοίγει η εφαρμογή και μόνο για 3 δευτερόλεπτα.

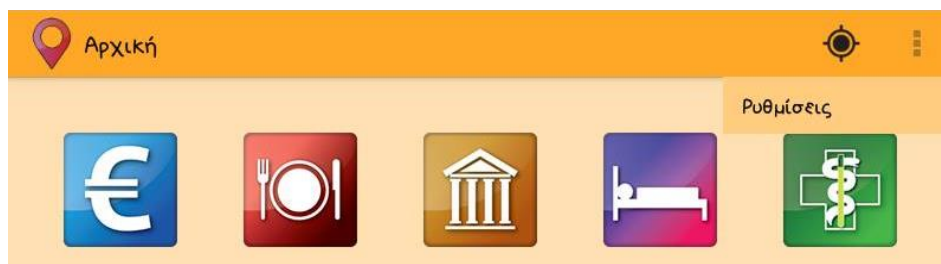


Εικόνα 13: Το λογότυπο της εφαρμογής που εμφανίζεται κατά την έναρξή της

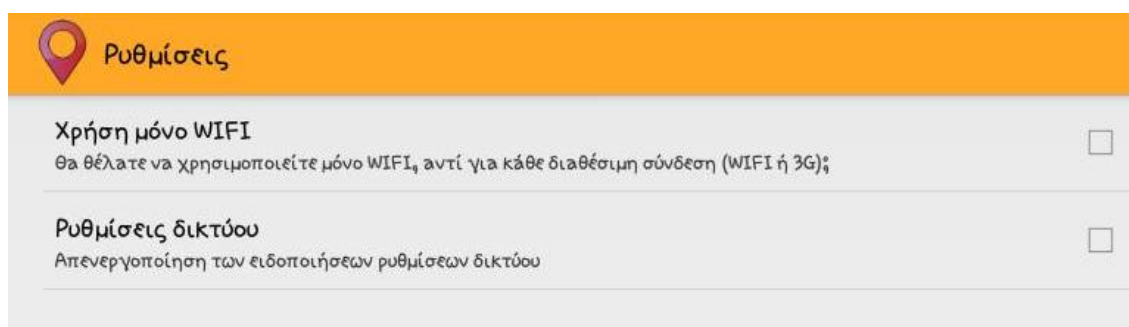
Στην συνέχεια, ο χρήστης οδηγείται στην αρχική οθόνη, που φαίνεται στην Εικόνα 11. Στην οθόνη αυτή, περιέχονται όλα τα σημεία εξυπηρέτησης-ενδιαφέροντος που μπορεί να αναζητήσει ο χρήστης, ένα κουμπί που τον οδηγεί στη λίστα με τα αγαπημένα σημεία που αυτός έχει προσθέσει, όπως επίσης και ένα μενού στη μπάρα (action bar), το οποίο περιέχει τους διάφορους δήμους της Θεσσαλονίκης για να εξιδεικεύσει ο χρήστης την αναζήτησή του, όπως φαίνεται στην Εικόνα 14. Εκτός από τους δήμους υπάρχει η επιλογή της τρέχουσας θέσης και η εμφάνιση όλων των σημείων ανεξάρτητα περιοχής. Τέλος, πάνω στην μπάρα εμφανίζεται το μενού που οδηγεί το χρήστη στις ρυθμίσεις της εφαρμογής, όπως αυτό φαίνεται στην Εικόνα 15. Το μενού αυτό υπάρχει σε όλες τις οθόνες. Οι ρυθμίσεις της εφαρμογής φαίνονται στην Εικόνα 16.



Εικόνα 14: Το μενού με τις επιλογές της τοποθεσίας για την αναζήτηση

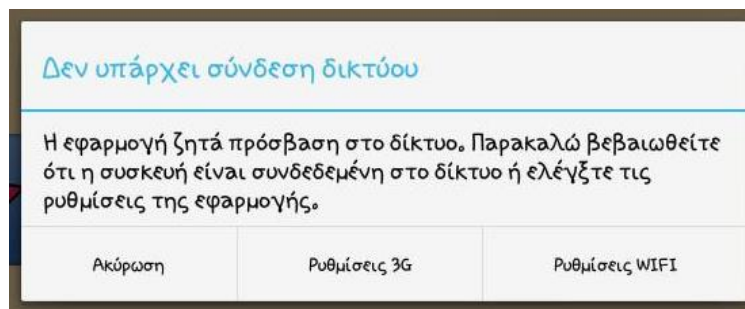


Εικόνα 15: Το μενού των ρυθμίσεων της εφαρμογής



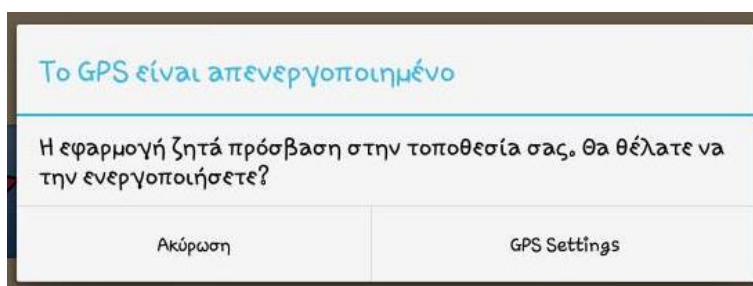
Εικόνα 16: Οι ρυθμίσεις της εφαρμογής

Αν ο χρήστης δεν έχει ενεργοποιήσει την σύνδεση με το internet και προσπαθήσει να αναζητήσει οποιοδήποτε σημείο ενδιαφέροντος, θα λάβει το μήνυμα που φαίνεται στην Εικόνα 17, το οποίο τον προτρέπει να ενεργοποιήσει είτε το WiFi είτε τα 3G δεδομένα.



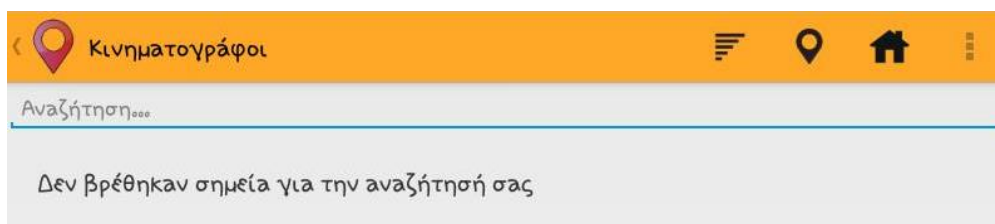
Εικόνα 17: Το μήνυμα προτροπής προς το χρήστη για την ενεργοποίηση της σύνδεσης στο Διαδίκτυο

Αφού υπάρχει μια οποιαδήποτε σύνδεση με τον internet, επιλέγουμε μια περιοχή από το μενού, στην συγκεκριμένη περίπτωση, είναι επιλεγμένη η τρέχουσα θέση και επιλέγουμε ένα σημείο εξυπηρέτησης-ενδιαφέροντος. Στις εικόνες αυτές, τα σημεία θα είναι τράπεζες της Θεσσαλονίκης. Αν η συσκευή του χρήστη δεν έχει ενεργοποιημένη την εύρεση τοποθεσίας του (GPS), τότε θα λάβει το μήνυμα της Εικόνας 18, το οποίο τον προτρέπει να την ενεργοποιήσει.



Εικόνα 18: Το μήνυμα προτροπής προς το χρήστη για την ενεργοποίηση του GPS

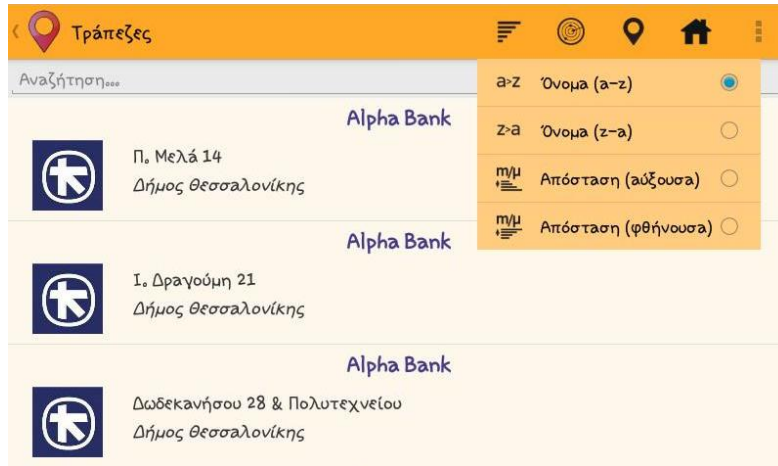
Στην συνέχεια, ο χρήστης θα οδηγηθεί στην οθόνη των αποτελεσμάτων. Σε περίπτωση που δεν βρεθούν αποτελέσματα για την αναζήτησή του θα λάβει το μήνυμα της Εικόνας 19.



Εικόνα 19: Το μήνυμα που λαμβάνει ο χρήστης όταν δεν έχουν βρεθεί αποτελέσματα για την αναζήτησή του

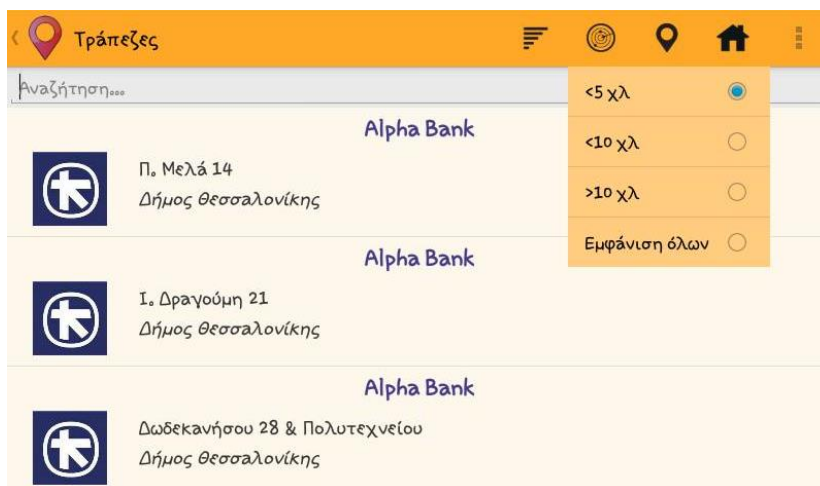
Η λίστα με τα αποτελέσματα της αναζήτησης παρουσιάζεται στην Εικόνα 20. Η λίστα περιέχει και μπάρα αναζήτησης όπου ο χρήστης μπορεί να αναζητήσει συγκεκριμένα σημεία με βάση το όνομά τους. Στην εικόνα παρουσιάζεται επίσης το πρώτο μενού της μπάρας, το οποίο έχει να κάνει με την ταξινόμηση των

σημείων, τόσο με το όνομα, όσο και με την απόστασή τους από την τωρινή θέση του χρήστη. Σημειώνεται ότι σε περίπτωση που η αρχική αναζήτηση δεν γίνει για την τωρινή θέση του χρήστη, οι επιλογές ταξινόμησης με βάση την απόσταση απουσιάζουν. Ακόμα, ως προεπιλογή, τα σημεία εμφανίζονται στην λίστα ταξινομημένα κατά αύξουσα σειρά με βάση το όνομα.



Εικόνα 20: Τα αποτελέσματα της αναζήτησης και το μενού ταξινόμησης αυτών

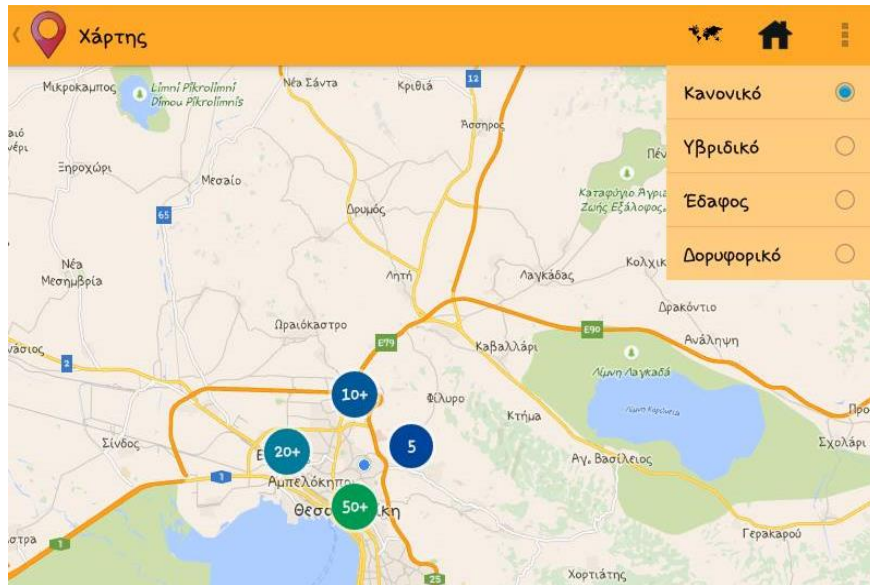
Το δεύτερο μενού της μπάρας, που εμφανίζεται στην Εικόνα 21, περιέχει επιλογές, οι οποίες επιτρέπουν στο χρήστη να αποφασίσει σε ποια ακτίνα απόστασης από τον ίδιο θα βρίσκονται τα σημεία που εμφανίζει η αναζήτηση. Ως προεπιλογή, εμφανίζονται μόνο τα σημεία που βρίσκονται σε ακτίνα μικρότερη των 5 χιλιομέτρων από το χρήστη. Επίσης, έχει την επιλογή να εμφανίσει όλα τα σημεία ανεξάρτητα απόστασης.



Εικόνα 21: Το μενού επιλογής της ακτίνας απόστασης των σημείων από την τοποθεσία του χρήστη

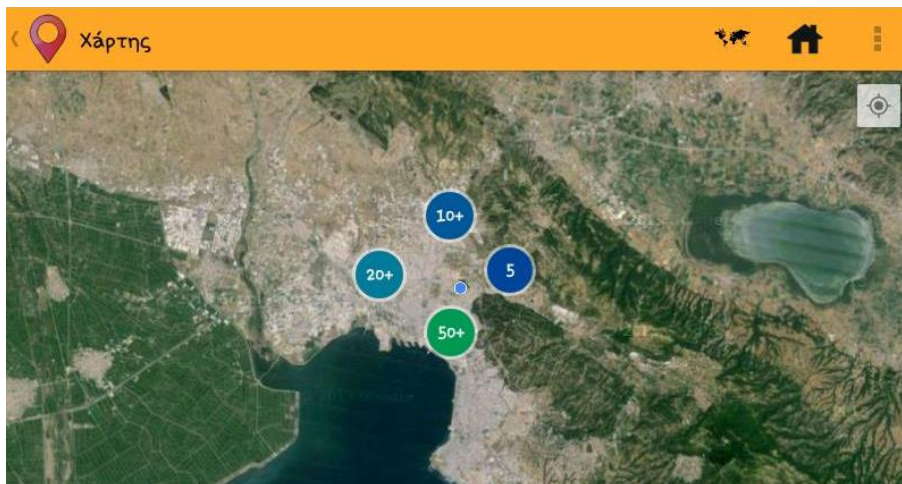
Το τρίτο μενού της οθόνης αυτής, οδηγεί το χρήστη σε μια άλλη οθόνη στην οποία εμφανίζονται όλα τα σημεία που επέστρεψε η αναζήτηση, στο χάρτη σε clusters,

όπως αυτό φαίνεται στην Εικόνα 22. Αν φυσικά η αναζήτηση δεν έχει επιστρέψει αποτελέσματα, ο χρήστης θα λάβει κατάλληλο μήνυμα ότι δεν υπάρχουν σημεία προς εμφάνιση. Το τέταρτο μενού οδηγεί τον χρήστη στην αρχική οθόνη της εφαρμογής και υπάρχει σε όλες τις οθόνες, εκτός βέβαια από την αρχική.



Εικόνα 22: Ο χάρτης με τα clustered σημεία και το μενού αλλαγής του εδάφους του χάρτη

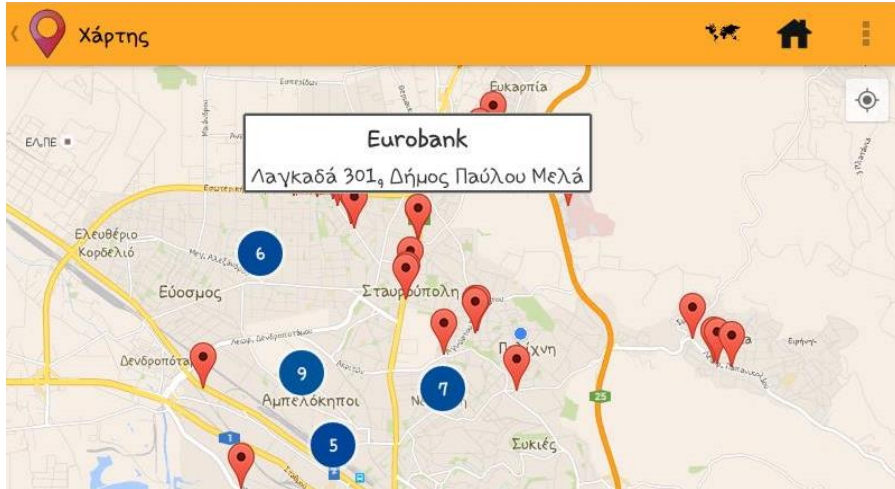
Όπως βλέπουμε στην Εικόνα 22, το πρώτο μενού που περιέχει η μπάρα, επιτρέπει στο χρήστη να αλλάξει τον τρόπο εμφάνισης του εδάφους του χάρτη. Ο ίδιος χάρτης με επιλεγμένη την επιλογή «Δορυφορικό» φαίνεται στην Εικόνα 23.



Εικόνα 23: Ο χάρτης με τα clustered σημεία, με δορυφορικό έδαφος

Τέλος, στο χάρτη αυτό ο χρήστης μπορεί να επιλέξει ένα σημείο έτσι ώστε να δει περισσότερες λεπτομέρειες για αυτό. Στην Εικόνα 24 φαίνεται ένα επιλεγμένο

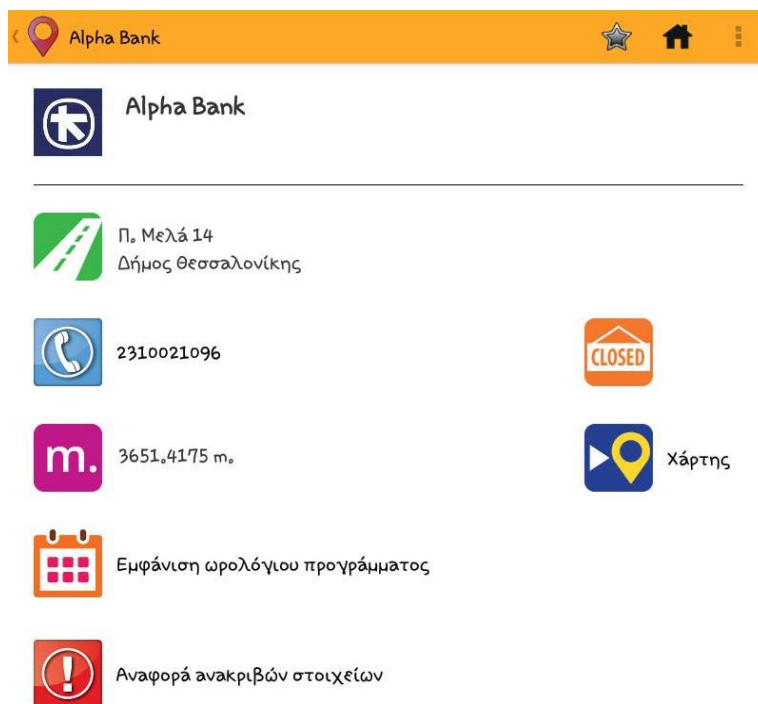
σημείο στον ίδιο χάρτη, ο οποίος απεικονίζει περισσότερα σημεία καθώς έχει μεγαλύτερο zoom.



Εικόνα 24: Ένα επιλεγμένο σημείο στο χάρτη εμφανίζει τις λεπτομέρειές του

Επιστρέφουμε πάλι στην οθόνη με τα αποτελέσματα της αναζήτησης, όπου ο χρήστης μπορεί να επιλέξει ένα από τα σημεία της λίστας για να δει όλες τις λεπτομέρειες του σημείου. Οδηγείται λοιπόν σε μια άλλη οθόνη, η οποία παρουσιάζει τις λεπτομέρειες αυτές, όπως φαίνεται στην Εικόνα 25.

Στη μπάρα της οθόνης αυτής, η πρώτη λειτουργία προσθέτει και αφαιρεί το επιλεγμένο σημείο στη λίστα των αγαπημένων

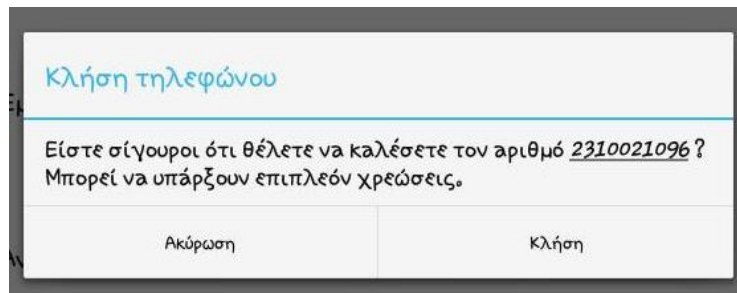


Εικόνα 25: Η οθόνη με τις λεπτομέρειες του επιλεγμένου σημείου

Οι λεπτομέρειες που εμφανίζονται για κάθε σημείο είναι οι εξής:

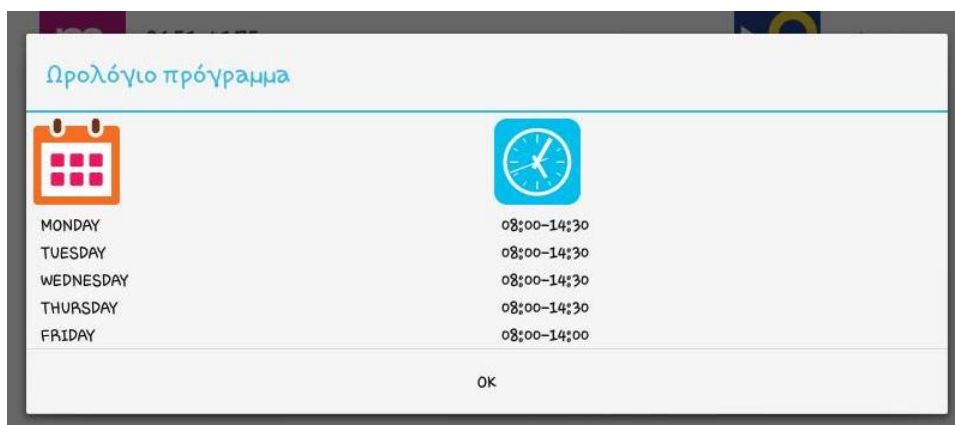
- Οδός και δήμος στον οποίο ανήκει το σημείο
- Τηλέφωνο του σημείου, αν υπάρχει
- Απόσταση του σημείου από το χρήστη, αν είναι ενεργοποιημένο το GPS
- Αν το σημείο είναι ανοιχτό ή κλειστό
- Οι ώρες λειτουργίας του καταστήματος, αν υπάρχουν

Ο χρήστης, αν επιθυμεί, μπορεί να πατήσει πάνω στο τηλέφωνο του σημείου έτσι ώστε να καλέσει τον αριθμό αυτό. Η εφαρμογή όμως, προειδοποιεί το χρήστη ότι μπορεί να προκύψουν επιπλέον χρεώσεις, όπως φαίνεται στην Εικόνα 26.



Εικόνα 26: Μήνυμα προειδοποίησης για επιπλέον χρεώσεις κατά την κλήση του τηλεφώνου του σημείου

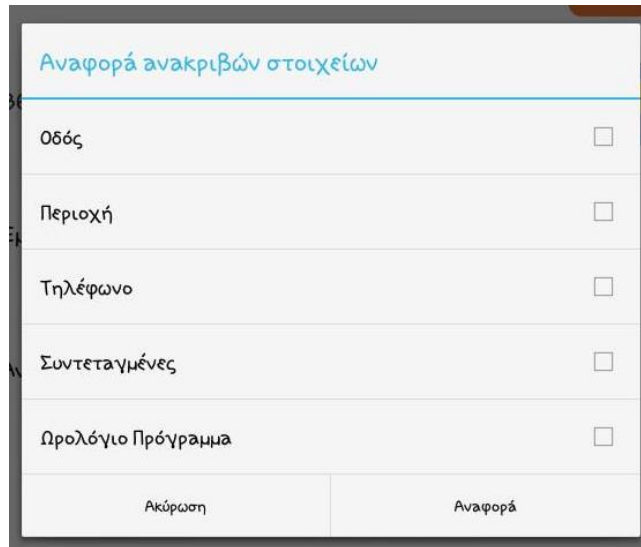
Ακόμα, το ωρολόγιο πρόγραμμα του σημείου εμφανίζεται όταν ο χρήστης πατήσει πάνω στο αντίστοιχο κουμπί, όπως φαίνεται στην Εικόνα 27.



Εικόνα 27: Το ωρολόγιο πρόγραμμα του επιλεγμένου σημείου

Ακόμα, υπάρχουν δύο επιπλέον λειτουργίες. Μία από αυτές είναι η αναφορά ανακριβών στοιχείων του σημείου. Δηλαδή, αν ο χρήστης θεωρεί πως κάποια από τις λεπτομέρειες του σημείου είναι λανθασμένη, μπορεί να την αναφέρει. Το

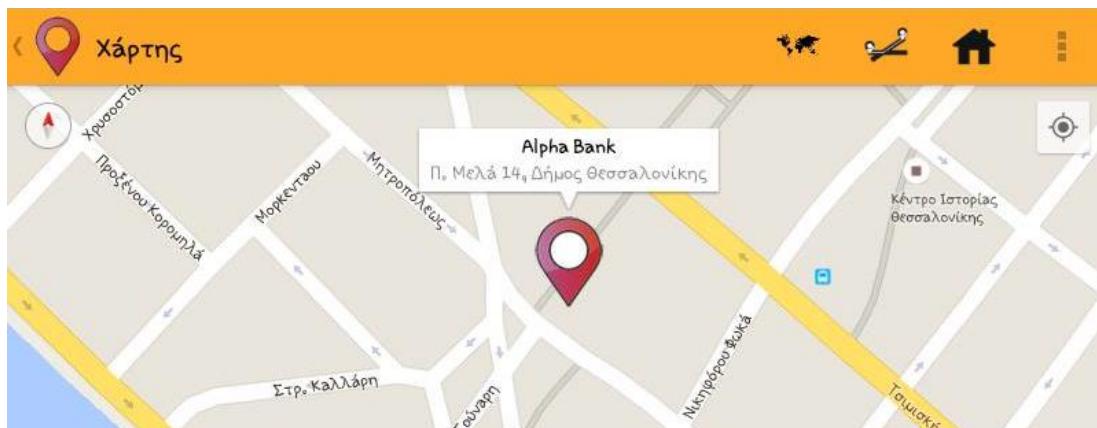
παράθυρο αναφοράς, το οποίο εμφανίζεται με τον ίδιο τρόπο όπως το ωρολόγιο πρόγραμμα, φαίνεται στην Εικόνα 28.



Αναφορά ανακριβών στοιχείων	
Οδός	<input type="checkbox"/>
Περιοχή	<input type="checkbox"/>
Τηλέφωνο	<input type="checkbox"/>
Συντεταγμένες	<input type="checkbox"/>
Ωρολόγιο Πρόγραμμα	<input type="checkbox"/>
Ακύρωση	Αναφορά

Εικόνα 28: Ο πίνακας για την αναφορά ανακριβών στοιχείων ενός σημείου

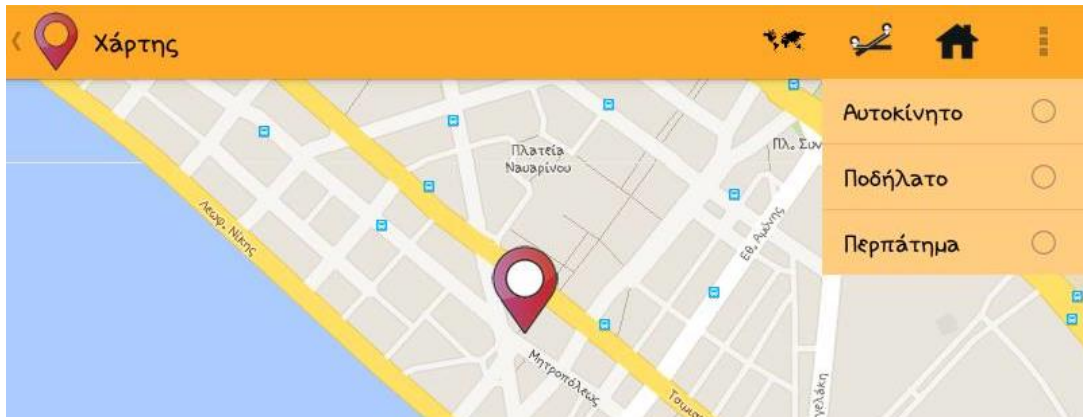
Η τελευταία λειτουργία είναι η εμφάνιση του σημείου στο χάρτη με το πάτημα του αντίστοιχου κουμπιού της οθόνης. Με την ενέργεια αυτή ο χρήστης οδηγείται σε μια νέα οθόνη όπου εμφανίζεται ο χάρτης. Η οθόνη αυτή φαίνεται στην Εικόνα 29.



Εικόνα 29: Το σημείο στο χάρτη με τις πληροφορίες που παρέχει κατά την επιλογή του marker

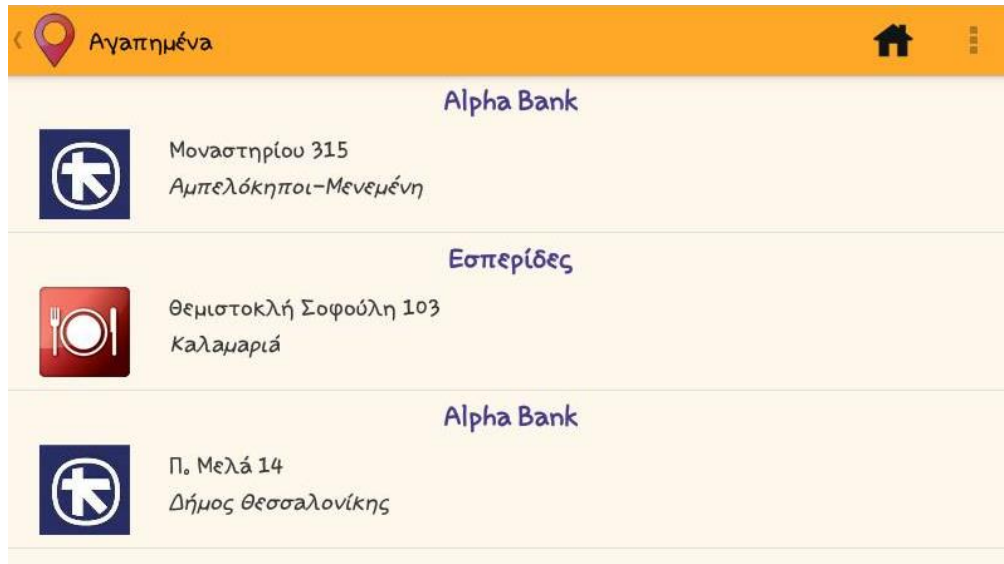
Η μπάρα της οθόνης αυτής περιέχει μια νέα λειτουργία, καθώς η πρώτη λειτουργία είναι ίδια με αυτή που εμφανίζεται στην Εικόνα 22. Η δεύτερη λειτουργία επιτρέπει στο χρήστη να εμφανίσει τη διαδρομή που χρειάζεται να ακολουθήσει από την τωρινή τοποθεσία του μέχρι το επιλεγμένο σημείο, όπως φαίνεται και στις Εικόνες 9 και 10. Η συγκεκριμένη λειτουργία φυσικά, λειτουργεί μόνο στην περίπτωση που ο χρήστης έχει το GPS ενεργοποιημένο. Σε αντίθετη περίπτωση, εμφανίζεται

κατάλληλο μήνυμα. Ο χρήστης μπορεί να επιλέξει μεταξύ τριών επιλογών για την εμφάνιση της διαδρομής, όπως φαίνεται στην Εικόνα 30.



Εικόνα 30: Οι επιλογές για την εμφάνιση της διαδρομής προς το επιλεγμένο σημείο στο χάρτη

Επιστρέφοντας πάλι στην αρχική οθόνη της Εικόνας 14, ο χρήστης μπορεί να επιλέξει να εμφανίσει την λίστα με τα αγαπημένα σημεία. Αν αυτή είναι κενή, θα εμφανιστεί κατάλληλο μήνυμα, αλλιώς θα εμφανιστούν όλα τα σημεία που έχει προσθέσει, όπως φαίνεται στην Εικόνα 31. Με την επιλογή κάποιου σημείου, ο χρήστης θα οδηγηθεί πάλι στην οθόνη της Εικόνας 25.



Εικόνα 31: Η λίστα με τα αγαπημένα

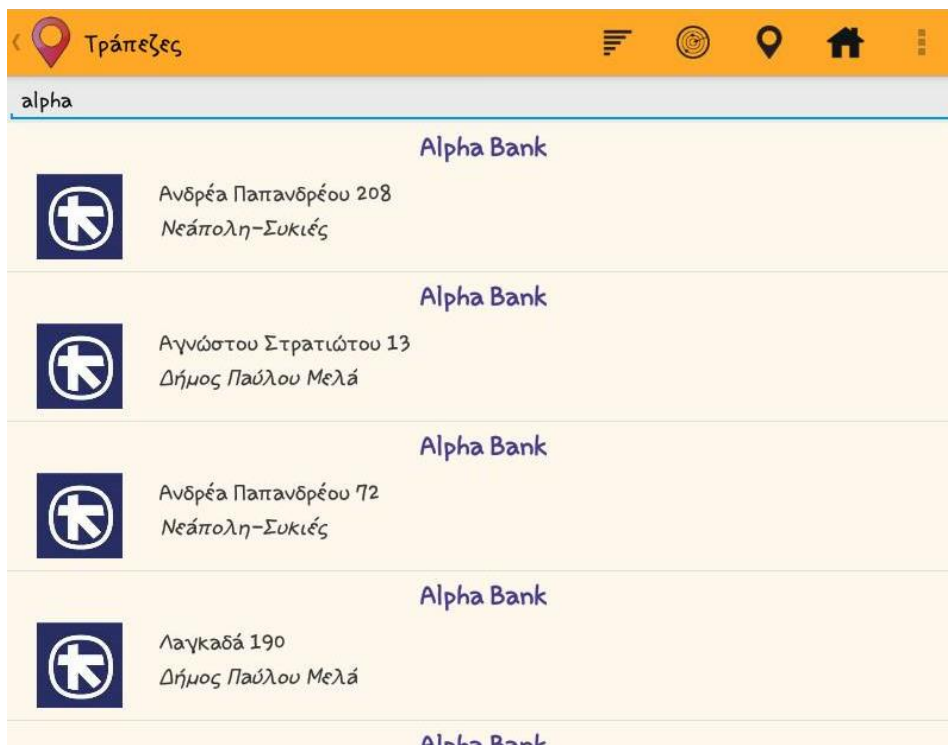
9.1 Use case αναζήτησης μιας τράπεζας

Υποθέτουμε ότι ο χρήστης θέλει να αναζητήσει μια τράπεζα Alpha Bank σύμφωνα με την τωρινή τοποθεσία του και έχει επίσης έχει ενεργοποιημένο το WIFI και το GPS, τα οποία είναι απαραίτητα στην συγκεκριμένη περίπτωση. Στην συνέχεια, θα

παραθέσω τα βήματα που πρέπει να ακολουθήσει έτσι ώστε βρει το επιθυμητό σημείο.

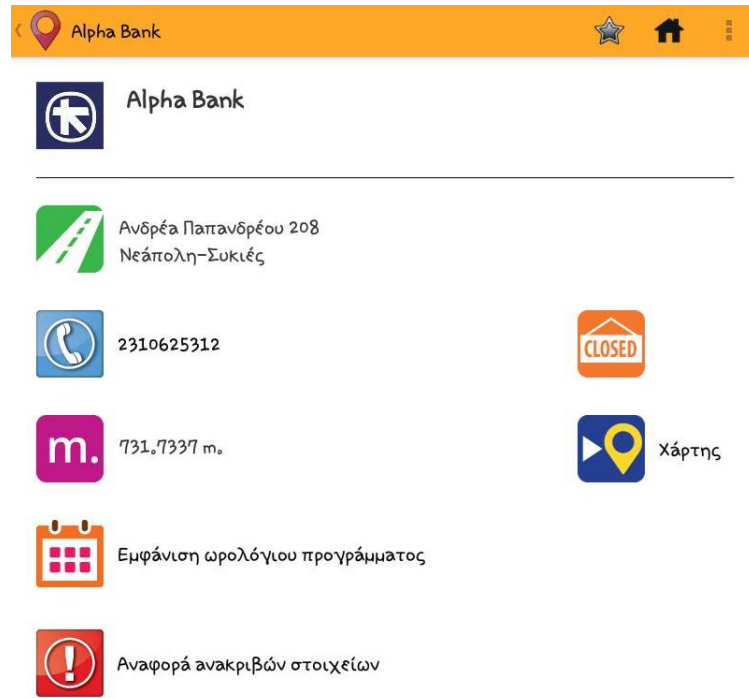
Αρχικά, μετά το άνοιγμα της εφαρμογής, ο χρήστης θα βρεθεί στην αρχική οθόνη που φαίνεται στην Εικόνα 14. Εκεί, από το μενού με τις περιοχές πρέπει να επιλέξει την επιλογή «Τρέχουσα θέση» και στην συνέχεια να πατήσει το πρώτο κουμπί από τα σημεία, που αντιστοιχεί στις τράπεζες. Έπειτα, και αφού φορτώσουν τα αποτελέσματα, θα βλέπει την οθόνη της Εικόνας 20. Υποθέτουμε, ότι υπάρχουν αποτελέσματα σε απόσταση 5 χιλιομέτρων από την τρέχουσα θέση του χρήστη.

Στην συνέχεια, ο χρήστης ταξινομεί τα σημεία με αύξουσα σειρά με βάση την απόσταση έτσι ώστε να βρει την κοντινότερη σε αυτόν τράπεζα και φιλτράρει τα αποτελέσματα ώστε να εμφανίζονται μόνο οι τράπεζες Alpha Bank, όπως φαίνεται στην Εικόνα 32.



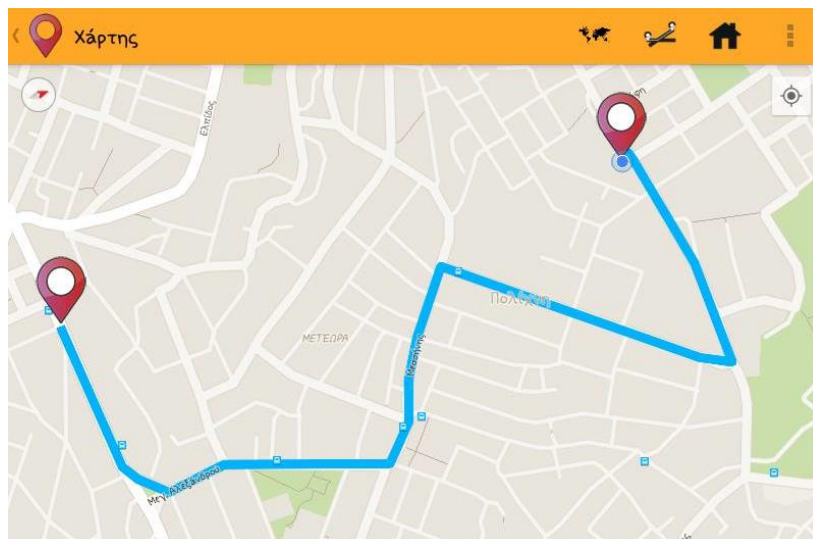
Εικόνα 32: Η λίστα με τα αποτελέσματα, με φίλτρο για αναζήτηση συγκεκριμένης τράπεζας

Επιλέγει το πρώτο σημείο από τη λίστα, που είναι και το κοντινότερο σε αυτόν, και οδηγείται στην οθόνη της Εικόνας 33, όπου βλέπει τις λεπτομέρειες του σημείου.



Εικόνα 33: Οι λεπτομέρειες της τράπεζας

Ο χρήστης επιλέγει να δει τον χάρτη πατώντας το αντίστοιχο κουμπί που βρίσκεται στην οθόνη. Στο χάρτη, εμφανίζεται τη διαδρομή με αυτοκίνητο από την τοποθεσία του μέχρι την τράπεζα, όπως αυτή φαίνεται στην Εικόνα 34. Στην εικόνα αυτή, ο μπλε κύκλος στο δεξιά σημείο υποδεικνύει την τοποθεσία του χρήστη, ενώ το άλλο σημείο υποδεικνύει την τράπεζα.



Εικόνα 34: διαδρομή με αυτοκίνητο από την τοποθεσία του χρήστη στην τράπεζα

Κεφάλαιο 10: Συσκευές δοκιμής

Για τις απαραίτητες δοκιμές της εφαρμογής, χρησιμοποίησα συνολικά 4 διαφορετικές συσκευές, 2 φυσικές και 2 emulators.

Οι φυσικές συσκευές περιλαμβάνουν ένα κινητό smartphone LG G3 s, το οποίο έχει εγκατεστημένη την έκδοση 5.0.2 Lollipop (API 21), με οθόνη 5 ιντσών, και ένα tablet Samsung Galaxy Tab 4, το οποίο έχει εγκατεστημένη την ίδια έκδοση με το κινητό αλλά διαθέτει 10.1 ιντσών. Από τους emulators χρησιμοποίησα ένα κινητό το οποίο είχε εγκατεστημένο το API 16 καθώς ήθελα να δοκιμάσω την εφαρμογή σε μικρότερες εκδόσεις του android αλλά και σε μικρότερη οθόνη, και ένα tablet Sony Xperia Tablet S με το ίδιο API και οθόνη 9.4 ιντσών.

Η δοκιμή με αυτές τις συσκευές βοήθησε στην ανάπτυξη μιας ευέλικτης εφαρμογής. Πιο συγκεκριμένα, κατάφερα να δημιουργήσω τα διάφορα layouts της εφαρμογής για αρκετά μεγέθη οθονών αλλά και την ίδια την εφαρμογή για να υποστηρίζει περισσότερες εκδόσεις του android.

Επίλογος

Η δημιουργία της εφαρμογής αυτής είχε μεγάλο ενδιαφέρον, καθώς ήρθα σε επαφή με την τεχνολογία του Google Maps, με την οποία δεν μου δόθηκε η ευκαιρία να εργαστώ σε προηγούμενες εφαρμογές. Ακόμα, έμαθα να προγραμματίζω με πιο αποδοτικό τρόπο και να βρίσκω λύση στα διάφορα προβλήματα τόσο μέσω της βοήθειας τρίτων, όσο και μέσω των προβλημάτων που ανέφεραν άλλοι χρήστες στο διαδίκτυο.

Τέλος, θεωρώ ότι η εφαρμογή αυτή μπορεί να αποτελέσει ένα χρήσιμο εργαλείο για τους χρήστες στην αναζήτηση των διαφόρων σημείων. Επίσης, στο μέλλον και έξω από τα πλαίσια της πτυχιακής αυτής, σκοπεύω να επεκτείνω τη λειτουργικότητα της εφαρμογής, έτσι ώστε να μπορεί να χρησιμοποιηθεί από κατοίκους διαφορετικών περιοχών.

Βιβλιογραφία

Ηλεκτρονική Βιβλιογραφία

1. Android Developers - <http://developer.android.com/index.html>
2. AndroidHive - <http://www.tutorialspoint.com/index.htm>
3. Eclipse, Android and Java training and support - <http://www.vogella.com/>
4. Github · Where software is built - <https://github.com/>
5. Google - <https://www.google.gr/>
6. Google APIs for Android | Google Developers
<https://developers.google.com/android/>
7. MySQL - <http://dev.mysql.com/>
8. PHP - <https://secure.php.net/>
9. Stack Overflow - <http://stackoverflow.com/>
10. TutorialsPoint - <http://www.tutorialspoint.com/index.htm>
11. Wikipedia, the free encyclopedia - https://en.wikipedia.org/wiki/Main_Page
12. Youtube - <https://www.youtube.com>