

Ατομική Διπλωματική Εργασία

**ΑΝΑΠΤΥΞΗ ANDROID APPLICATION ΜΕ ΥΠΗΡΕΣΙΕΣ
ΠΟΥ ΕΞΥΠΗΡΕΤΟΥΝ ΤΟ ΤΟΥΡΙΣΤΙΚΟ ΚΟΙΝΟ
ΤΗΣ ΚΥΠΡΟΥ**

Κωνσταντίνος Φασουλής

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2013

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάπτυξη Android Application Με Υπηρεσίες Που Εξυπηρετούν
Το Τουριστικό Κοινό Της Κύπρου

Κωνσταντίνος Φασουλής

Επιβλέπων Καθηγητής
Δρ. Παρασκευάς Ευριπίδου

Η παρούσα Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2013

Ευχαριστίες

Με το πέρας των προπτυχιακών μου σπουδών αναπολώ συγκινημένος τις στιγμές που πέρασαν από την πρώτη κιόλας μέρα στο Πανεπιστήμιο. Στο μυαλό μου έρχονται τα πρόσωπα που τις συνόδευσαν και στάθηκαν δίπλα μου στην προσπάθεια επίτευξης αυτού του πρώτου μου ακαδημαϊκού στόχου, με αποτέλεσμα να είμαι τώρα εδώ και να γράφω αυτές τις γραμμές.

Καταρχήν, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, Δρ. Παρασκευά Ευριπίδου που μου έδωσε την ευκαιρία να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα.

Επιπρόσθετα, θα ήθελα να εκφράσω την βαθιά μου ευγνωμοσύνη στην Επίκουρη Καθηγήτρια του Τμήματος Δημόσιας Διοίκησης και Διοίκησης Επιχειρήσεων, Δρ. Αλεξία Παναγιώτου για τον καθοριστικό ρόλο που διαδραμάτισε στην ακαδημαϊκή μου πορεία τα τελευταία χρόνια, ωθώντας με να ασχοληθώ με καινούρια θέματα, μεταξύ των οποίων και με την ανάπτυξη της εφαρμογής αυτής.

Επιπλέον, ιδιαίτερες ευχαριστίες οφείλω στο μεταπτυχιακό φοιτητή Κωνσταντίνο Κώστα και τους συμφοιτητές μου Μάριο Μιχαήλ και Κυριακή Γεωργίου για τις πολύτιμες τους συμβουλές κατά τη διάρκεια υλοποίησης της εφαρμογής. Επίσης, ένα μεγάλο ευχαριστώ οφείλω στις φίλες μου Έμιλυ Κάπρου και Πόπη Χριστοπούλου για τη στήριξή τους και στις Άντρη Χριστοδούλου και Λουίζα Αντωνίου για το σχεδιασμό των γραφικών.

Τέλος, θερμές ευχαριστίες οφείλω και στην οικογένεια μου που αποτέλεσε το αποκούμπι μου όλα αυτά τα χρόνια και με στήριξε αμέριστα καθ' όλη τη διάρκεια της ακαδημαϊκής μου πορείας.

Περίληψη

Στην εποχή που διανύουμε, στις ψυχές των ανθρώπων κυριαρχεί η μίζερια και η αβεβαιότητα, λόγω της οικονομικής κρίσης που πλήττει εδώ και μήνες τη χώρα. Οι προβλέψεις για την οικονομία κάθε άλλο παρά αισιόδοξες είναι και τα ποσοστά ύφεσης και ανεργίας ανεβαίνουν μέρα με τη μέρα.

Σε ένα τέτοιο περιβάλλον, η άλλοτε «κότα με τα χρυσά αυγά», ο τουρισμός, φαντάζει ως αχτίδα ελπίδας και οξυγόνο για την οικονομία του τόπου. Η Κύπρος μας είναι ευλογημένη με απaráμιλλη φυσική ομορφιά, πλούσιο πολιτισμό και ιστορία, πράγματα που ελκύουν εκατομμύρια τουρίστες ετησίως. Όλα αυτά, σε συνδυασμό με τη ραγδαία ανάπτυξη που γνωρίζουν τα «έξυπνα» τηλέφωνα και ιδιαίτερα το λειτουργικό σύστημα Android, συνθέτουν ένα ιδανικό σκηνικό για την ανάπτυξη εφαρμογών που έχουν στο επίκεντρό τους τον τουρισμό.

Λαμβάνοντας υπόψη τα πιο πάνω, ανέλαβα, στα πλαίσια της Ατομικής Διπλωματικής μου Εργασίας, να δημιουργήσω μια εφαρμογή σε λειτουργικό σύστημα Android με υπηρεσίες που θα εξυπηρετούν το τουριστικό κοινό της Κύπρου, υπό την ονομασία LoveCyprus.

Κύρια χαρακτηριστικά της εφαρμογής αποτελούν η κατηγοριοποίηση των επιλογών του χρήστη σε μια λίστα που διαχωρίζει τα προτεινόμενα μέρη ανάλογα με τις ανάγκες του, η επισήμανση προτεινόμενων σημείων σε χάρτη ολόκληρης της Κύπρου, ο γεωγραφικός εντοπισμός του χρήστη και η αναφορά εισηγήσεων και οδηγιών για πρόσβαση σε μέρη που βρίσκονται εντός συγκεκριμένης ακτίνας από αυτόν.

Κατα τη διάρκεια της εκπόνησης της εργασίας, μελετήθηκε εις βάθος η δομή του λειτουργικού συστήματος Android, ηλεκτρονικοί τουριστικοί οδηγοί άλλων χωρών και έντυποι τουριστικοί οδηγοί της Κύπρου. Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε το περιβάλλον ανάπτυξης εφαρμογών Eclipse (Eclipse IDE – Integrated Development Environment) με ενσωματωμένο το Android SDK (Software Development Kit) της Google.

Περιεχόμενα

Κεφάλαιο 1	1
1.1 Σκοπός	1
1.2 Δομή Εργασίας	1
Κεφάλαιο 2	3
2.1 «Εξυπνα» τηλέφωνα	3
2.2 Το Λειτουργικό Σύστημα Android	4
2.3 Έντυποι και ηλεκτρονικοί τουριστικοί οδηγοί	6
Κεφάλαιο 3	9
3.1 Γενική Περιγραφή Εφαρμογής	9
3.2 Απαιτήσεις	10
3.3 Προδιαγραφές	13
3.4 Υλοποίηση	14
3.4.1 Κύρια συστατικά εφαρμογής.....	14
3.4.2 Βάση Δεδομένων SQLite	20
3.4.3 Δομή εφαρμογής.....	21
Κεφάλαιο 4	31
4.1 Μέθοδος Αξιολόγησης.....	31
4.2 Αποτελέσματα Αξιολόγησης	32
Κεφάλαιο 5	35
5.1 Σύνοψη.....	35
5.2 Μελλοντική Εργασία	35
5.3 Εμπειρίες που Αποκτήθηκαν	37
Βιβλιογραφία.....	40
Παράρτημα Α.....	A-1
Παράρτημα Β	B-1

Κεφάλαιο 1

Εισαγωγή

1.1 Σκοπός	1
1.2 Δομή	1

1.1 Σκοπός

Η εφαρμογή αυτή στηρίζεται στο λειτουργικό σύστημα Android και αναπτύχθηκε με γνώμονα την ικανοποίηση των μέσων αναγκών των τουριστών που επισκέπτονται το νησί.

Συγκεκριμένα, η εφαρμογή αποσκοπεί στο να προτείνει στους τουρίστες της Κύπρου τα ωραιότερα μέρη που μπορούν να επισκεφθούν. Τα μέρη αυτά χωρίζονται, ανάλογα με την κατηγορία στην οποία ανήκουν, σε αξιοθέατα, εστιατόρια και εμπορικά κέντρα. Ταυτόχρονα, προσφέρονται ιστορικά στοιχεία για το νησί αλλά και πληροφορίες για υπηρεσίες έκτακτης ανάγκης.

Όλα τα πιο πάνω παρουσιάζονται σε χάρτη, είτε σε παγκύπρια βάση, είτε στην περιοχή που βρίσκονται. Η εφαρμογή παρέχει στους χρήστες τη δυνατότητα λήψης πληροφοριών και οδηγιών πρόσβασης σε αυτά.

Με τις υπηρεσίες αυτές η εφαρμογή έχει ως κύριο σκοπό να προσφέρει τη μεγαλύτερη δυνατή ώθηση στον τουρισμό της Κύπρου μέσω των ραγδαία αναπτυσσόμενων «έξυπνων» τηλεφώνων και του λειτουργικού συστήματος Android, με απώτερο στόχο να επιφέρει ευεργετικά αποτελέσματα στην παρακμάζουσα κυπριακή οικονομία.

1.2 Δομή Εργασίας

Το πρώτο κεφάλαιο της Ατομικής Διπλωματικής Εργασίας αποτελεί μια εισαγωγή σχετικά με τον σκοπό και την δομή της παρούσας διπλωματικής εργασίας.

Στο δεύτερο κεφάλαιο γίνεται μια αναφορά στα «έξυπνα» τηλέφωνα και τις δυνατότητες τους, στα είδη λειτουργικών συστημάτων και τέλος στο λειτουργικό σύστημα Android. Επιπρόσθετα, μελετώνται ηλεκτρονικοί τουριστικοί άλλων χωρών για να εξεταστούν διάφοροι τύποι διεπαφών και έντυποι τουριστικοί οδηγοί της Κύπρου για άντληση πληροφοριών για της ανάγκες της εφαρμογής.

Στο τρίτο κεφάλαιο, παρουσιάζεται μια εκτενής περιγραφή της εφαρμογής. Στο κεφάλαιο αυτό περιλαμβάνονται οι απαιτήσεις και οι προδιαγραφές και η υλοποίηση της εφαρμογής.

Στο τέταρτο κεφάλαιο, περιγράφεται η διαδικασία αξιολόγησης της εφαρμογής

Το πέμπτο και τελευταίο κεφάλαιο, περιλαμβάνει μια σύνοψη της διαδικασίας ανάπτυξης που ακολουθήθηκε καθώς και εισηγήσεις για τη μελλοντική επέκταση της εφαρμογής.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

2.1 «Έξυπνα» τηλέφωνα	3
2.2 Το Λειτουργικό Σύστημα Android	4
2.3 Έντυποι και Ηλεκτρονικοί τουριστικοί οδηγοί	6

2.1 «Έξυπνα» τηλέφωνα

Τα έξυπνα κινητά τηλέφωνα (smartphones) προσφέρουν λειτουργίες παρόμοιες με αυτές των ηλεκτρονικών υπολογιστών και βασίζονται σε εξειδικευμένα λειτουργικά συστήματα. Οι λειτουργίες αυτές περιλαμβάνουν, μεταξύ άλλων, αποστολή μηνυμάτων κειμένου, διαχείριση ηλεκτρονικού ταχυδρομείου, GPS, πρόσβαση στον παγκόσμιο ιστό, λήψη φωτογραφιών και βίντεο κ.α. Επιπρόσθετα με τις εφαρμογές που περιλαμβάνονται στο λειτουργικό τους σύστημα, μια πλειάδα άλλων εφαρμογών που είναι διαθέσιμες για εγκατάσταση τα μετατρέπουν σε κινητούς προσωπικούς υπολογιστές [1].

Σήμερα τα έξυπνα κινητά τηλέφωνα έχουν προσελκύσει το ενδιαφέρον όχι μόνο όσων ασχολούνται με την τεχνολογία, αλλά και των απλών χρηστών, αφού μέσα από τις πολυποίκιλες δυνατότητες τους προσφέρουν μια φορητή και πιο εύχρηστη εναλλακτική λύση στη χρήση των ηλεκτρονικών υπολογιστών.

Η επιτυχία που γνωρίζουν στις μέρες μας τα smartphones οφείλεται κατά κύριο λόγο στις δυνατότητες που παρέχονται από τα λειτουργικά τους συστήματα. Δημοφιλέστερο και με συνεχώς αυξανόμενα ποσοστά, το Android κατατάσσεται στην πρώτη θέση μεταξύ των λειτουργικών συστημάτων για «έξυπνα» τηλέφωνα, ενώ ακολουθεί το iOS της Apple, αφήνοντας ως “φτωχούς συγγενείς” τα λειτουργικά Windows Mobile της Microsoft και BlackBerry OS της BlackBerry.

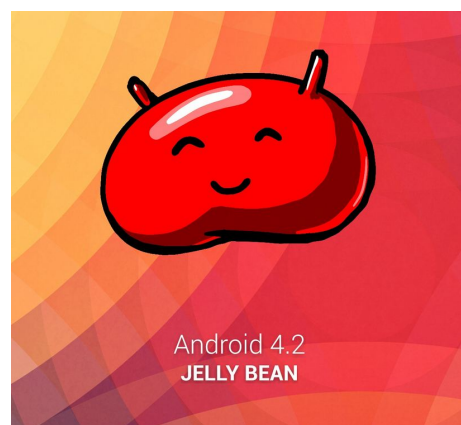
2.2 Το Λειτουργικό Σύστημα Android

Το Λειτουργικό Σύστημα Android είναι ένα λειτουργικό σύστημα βασισμένο στον πυρήνα του λειτουργικού συστήματος Linux και σχεδιάστηκε κυρίως για κινητές συσκευές με οθόνες αφής. Αρχικά, το Android αναπτύχθηκε από την Android, Inc., η οποία εξαγοράστηκε το 2005 από την Google. Τα “αποκαλυπτήρια” του Android έγιναν το 2007 παράλληλα με την ίδρυση της «κοινοπραξίας» εταιρειών λογισμικού, υλικού και κινητής τηλεφωνίας «Open Handset Alliance» με σκοπό την προώθηση ανοιχτών λογισμικών για κινητά τηλέφωνα. [2]

Το Android είναι λογισμικό ανοιχτού κώδικα και επιτρέπει στους προγραμματιστές λογισμικού να δημιουργούν εφαρμογές χρησιμοποιώντας τη γλώσσα προγραμματισμού Java, σε συνδυασμό με της βιβλιοθήκες λογισμικού Android SDK (Software Development Kit) της Google. Οι εφαρμογές σε Android εκτελούνται σε μια εικονική μηχανή Dalvik στις συσκευές Android, αφού προηγουμένως έχουν μετατραπεί από κώδικα Java σε bytecode και αργότερα σε εκτελέσιμα αρχεία Dalvik [3].

Από το 2009 και έπειτα, το Android έρχεται σε διάφορες εκδόσεις, που φέρουν ένα κωδικό όνομα με τις επόμενες εκδόσεις του να ακολουθούν σε αλφαβητική σειρά. Ιστορικά, οι πρώτες εκδόσεις είχαν την ονομασία Android alpha και Android beta ενώ κατά σειρά, οι εμπορικές εκδόσεις του Android ήταν:

- Android 1.5 Cupcake
- Android 1.6 Donut
- Android 2.0 – 2.1 Eclair
- Android 2.2 – 2.2.3 Froyo
- Android 2.3 – 2.3.3 Gingerbread
- Android 3.0 – 3.2 Honeycomb
- Android 4 Ice Cream Sandwich
- Android 4.1 – 4.2 Jelly Bean [4]

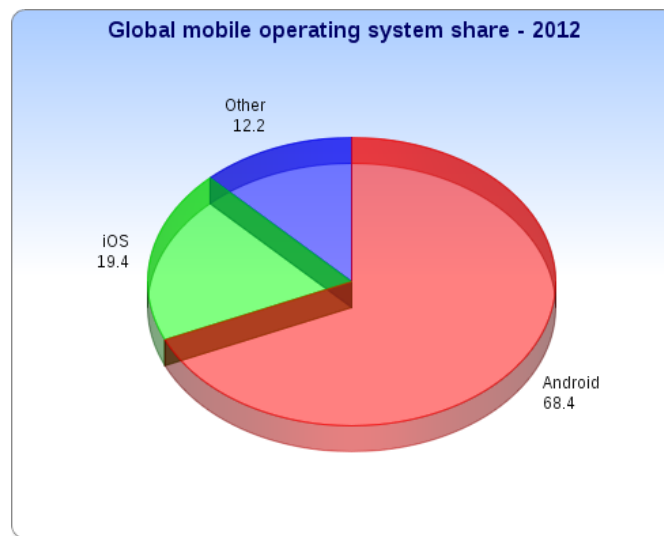


Εικόνα 2.2.1 Η τελευταία έκδοση Android

Η ιλιγγιώδης ανάπτυξη εφαρμογών για το λειτουργικό Android παρουσιάστηκε κυρίως από την έκδοση 4 και έπειτα, με την παράλληλη ενσωμάτωση και

αξιοποίηση διάφορων αισθητήρων στα κινητά τηλέφωνα. Χαρακτηριστικό είναι το γεγονός ότι κάθε μέρα ενεργοποιούνται 1 εκατομμύριο συσκευές Android. Η Google, μέσω των χαρακτηριστικών και της καινοτομίας που ενέπνευσε στις τελευταίες εκδόσεις του λειτουργικού της συστήματος, προσπάθησε και πέτυχε τη μετάβαση αρκετών χρηστών στις τελευταίες εκδόσεις του λειτουργικού της. Παράλληλα, προσέφερε και μια σειρά βιβλιοθηκών υποστήριξης, την Android Support Library, με την οποία οι προγραμματιστές μπορούν να δημιουργούν εφαρμογές που θα ήταν συμβατές με τις τελευταίες αλλά και προηγούμενες εκδόσεις του λειτουργικού συστήματος, προσθέτοντας όμως αρκετές αλλαγές στον τρόπο υλοποίησης των εφαρμογών.

Σύμφωνα με πρόσφατες μελέτες, τη συντριπτική πλειοψηφία «έξυπνων» κινητών συσκευών κατέχουν οι συσκευές με λειτουργικό Android, ενώ ανάμεσα στις διάφορες εκδόσεις του λειτουργικού κερδίζουν έδαφος οι τελευταίες εκδόσεις Ice Cream Sandwich και Jelly Bean [5].



Εικόνα 2.2.2 Μερίδια αγοράς λειτουργικών συστημάτων για smartphones

Λαμβάνοντας υπόψη όλα τα πιο πάνω, την υποστήριξη εφαρμογών ανοικτού κώδικα, την ευκολία χρήσης αλλά και ανάπτυξης εφαρμογών στο εν λόγω λειτουργικό, οδηγήθηκα στην επιλογή της συγκεκριμένης πλατφόρμας για την ανάπτυξη της εφαρμογής που μου ανατέθηκε.

2.3 Έντυποι και ηλεκτρονικοί τουριστικοί οδηγοί

Κατα τη διαδικασία ανάπτυξης της εφαρμογής μελετήθηκαν διάφοροι ηλεκτρονικοί οδηγοί άλλων χωρών για να παρατηρηθούν διάφορες τεχνικές σχεδιασμού της διεπαφής του χρήστη και οι ενσωματωμένες τους λειτουργίες.

Παράλληλα, μελετήθηκαν και έντυποι τουριστικοί οδηγοί της Κύπρου, για άντληση πληροφοριών που θα χρησιμοποιούνταν στην υπό ανάπτυξη εφαρμογή.

Όσον αφορά τους ηλεκτρονικούς οδηγούς, μελετήθηκαν ήδη υπάρχουσες εφαρμογές που περιλαμβάνονται στο Google Play, την πλατφόρμα διανομής εφαρμογών του Android. Χαρακτηριστικά και εικόνες από τις συγκεκριμένες εφαρμογές παρουσιάζονται πιο κάτω.

	Barcelona City Guide (TripAdvisor)	Barcelona City Guide (GuidePal)	Barcelona – Travel Guide (PRS Marketing Services SL)	Tourist Eye – Travel Guide (Tourist Eye)
Free Download	✓	✓	✓	✓
Maps Integration	✓	✓	✓	✓
Other Users Feedback	✓		✓	✓
GPS Based	✓	✓	✓	✓
Application's Social Network			✓	✓
Social Network Integration		✓		

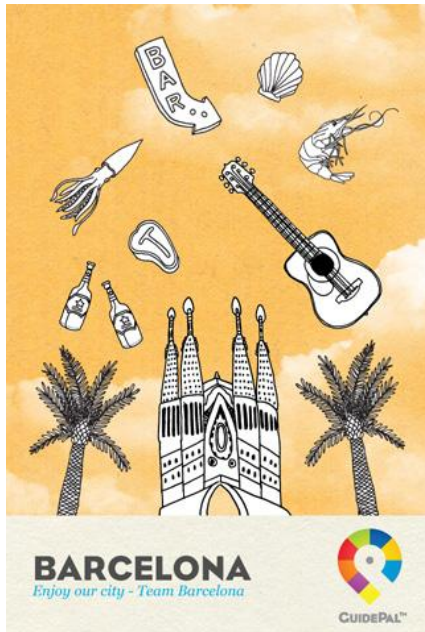
Πίνακας 2.3.1 Χαρακτηριστικά εφαρμογών

Παρόμοια έρευνα έχει γίνει και για ηλεκτρονικούς τουριστικούς οδηγούς που αφορούν την Κύπρο, χωρίς να παρουσιαστεί κάτι το αξιόλογο. Οι μόνες εφαρμογές που εντοπίστηκαν αναφέρονταν κυρίως στα κατεχόμενα από τους Τούρκους εδάφη της Κύπρου, είχαν πολύ χαμηλή βαθμολογία και κακά σχόλια από τους χρήστες τους.

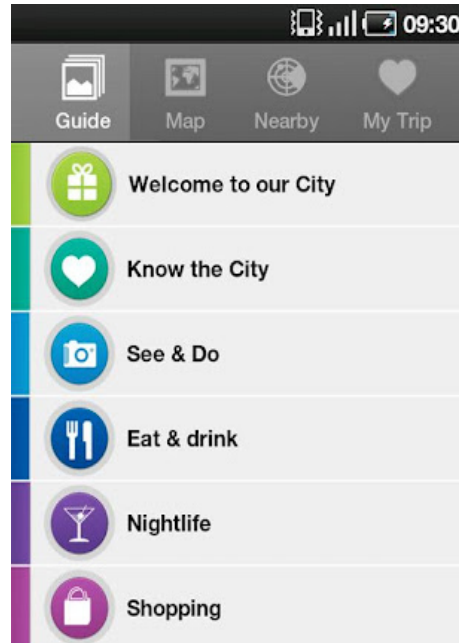
Το γεγονός αυτό ήταν ένα επιπλέον κίνητρο για την ανάπτυξη ενός αξιόλογου τουριστικού οδηγού της Κύπρου, που θα ικανοποιούσε το τουριστικό κοινό της νήσου.

Επεξήγηση σημαντικότερης εφαρμογής

Η εφαρμογή στην οποία βασίστηκε το προσχέδιο της υπό ανάπτυξη εφαρμογής, όσον αφορά τη διεπαφή, είναι η Barcelona City Guide από την εταιρεία GuidePal, εικόνες από την οποία φαίνονται πιο κάτω.



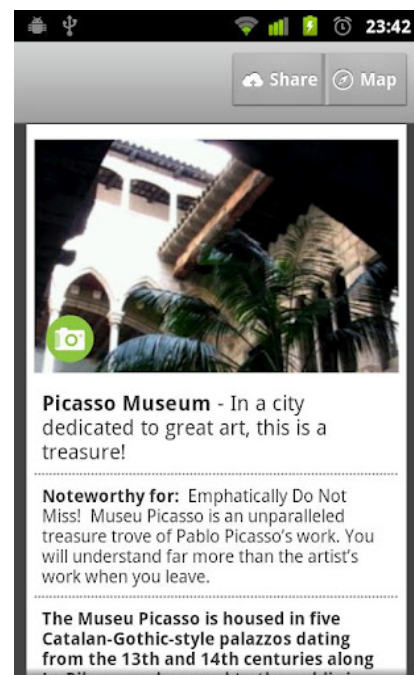
Εικόνα 2.3.1 Εισαγωγική οθόνη



Εικόνα 2.3.2 Κατηγορίες επιλογής



Εικόνα 2.3.3 Χάρτης Πόλης



Εικόνα 2.3.4 Πληροφορίες τοποθεσίας

Στην προσπάθεια άντλησης πληροφοριών για εμπλουτισμό της υπό ανάπτυξη εφαρμογής, μελετήθηκαν οι πιο κάτω περιεκτικοί έντυποι τουριστικοί οδηγοί για την Κύπρο. Στους οδηγούς αυτούς περιλαμβάνονται εισηγήσεις για διάφορα μέρη σε κάθε πόλη ξεχωριστά, αλλά και μονοπάτια που συνδυάζουν διαδρομές και επισκέψεις σε αξιοθέατα διάφορων πόλεων.

Ο οδηγός “Ανακαλύψτε την Κύπρο: Λευκωσία – Λάρνακα – Λεμεσός – Πάφος”, μια ειδική έκδοση της ελληνικής εφημερίδας “Τα ΝΕΑ” αποτέλεσε την κύρια πηγή πληροφοριών για τον εμπλουτισμό της εφαρμογής, αφού αποτελεί πρότυπο τουριστικού οδηγού με λεπτομερείς περιγραφές αξιοθέατων κάθε πόλης και εισηγήσεις διαδρομών που μαγεύουν.

Επιπρόσθετα, ως συμπληρωματικός έντυπος τουριστικός οδηγός χρησιμοποιήθηκε ο “Cyprus Holiday Guide 2012” που διανέμεται από τη Cyta. Στον οδηγό αυτό συμπεριλαμβάνονται, όπως και προηγουμένως, λεπτομερείς περιγραφές αξιοθέατων κάθε πόλης και επιπρόσθετος κατάλογος ανά πόλη με εστιατόρια, καφετέριες, κέντρα αναψυχής, πάρκα, αλλά και εισηγήσεις διαδρομών για αθλητικές δραστηριότητες, όπως ποδηλασία, τρέξιμο, ορειβασία κτλ.



**Εικόνα 2.3.5 Ανακαλύψτε την Κύπρο.
Έντυπος Τουριστικός Οδηγός**



**Εικόνα 2.3.6 Έντυπος Τουριστικός
Οδηγός Cyta**

Κεφάλαιο 3

Περιγραφή Εφαρμογής LoveCyprus

3.1 Γενική Περιγραφή Εφαρμογής	9
3.2 Απαιτήσεις	10
3.3 Προδιαγραφές	13
3.4 Υλοποίηση	14

3.1 Γενική Περιγραφή Εφαρμογής

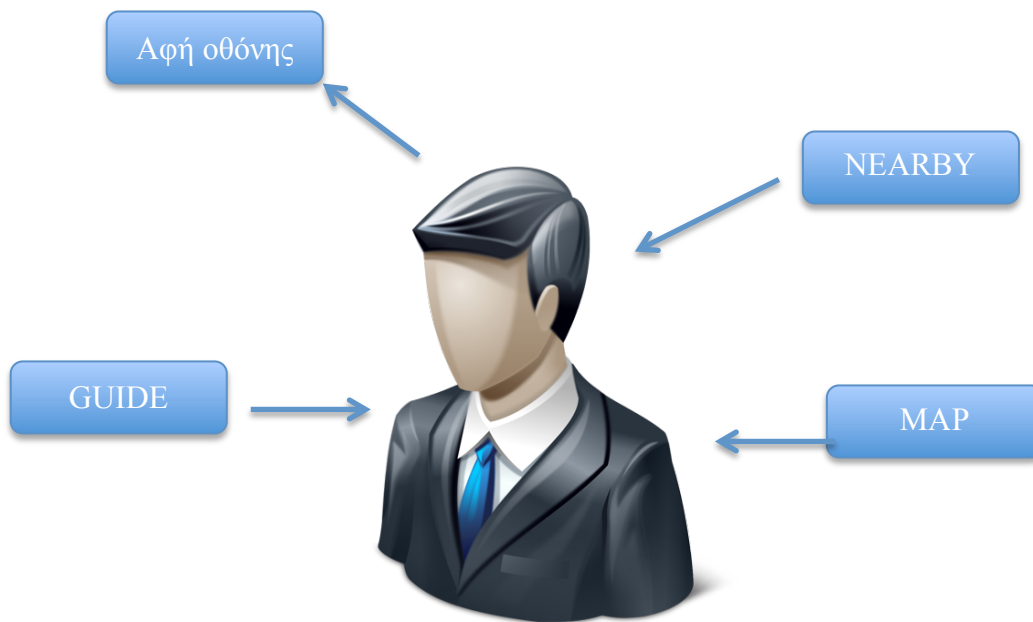
Η εφαρμογή αυτή αναπτύχθηκε με κύριο σκοπό την εξυπηρέτηση των αναγκών των τουριστών που επισκέπτονται την Κύπρο. Με την εγκατάσταση της εφαρμογής στις κινητές συσκευές τους, είτε αυτές είναι «έξυπνα» τηλέφωνα είτε tablets με λειτουργικό Android 4 (Ice Cream Sandwich) και άνω, οι τουρίστες είναι σε θέση να περιηγηθούν στο νησί και να πληροφορηθούν για τα πιο δημοφιλή αξιοθέατα που υπάρχουν.

Όπως στις πλείστες εφαρμογές, έτσι και σε αυτή, στο επίκεντρο βρίσκεται ο χρήστης της εφαρμογής, δηλαδή το τουριστικό κοινό της Κύπρου, αφού είναι το άτομο το οποίο θα αλληλεπιδρά με την εφαρμογή.

Ο χρήστης αλληλεπιδρά με την εφαρμογή μέσω της οθόνης αφής της κινητής του συσκευής, ανάλογα με τις επιλογές που του παρουσιάζονται σε αυτή, από την εφαρμογή.

Με την εκκίνηση της εφαρμογής, παρουσιάζονται στο χρήστη τρεις κύριες επιλογές: Αυτή του τουριστικού οδηγού, του χάρτη της Κύπρου και των πλησιέστερων σε αυτόν σημείων ενδιαφέροντος. Μέσα από αυτές τις επιλογές, ο χρήστης θα μπορεί να περιηγηθεί στην εφαρμογή, είτε ανά κατηγορία ενδιαφέροντος, είτε μέσω των χαρτών της εφαρμογής.

Πιο κάτω φαίνεται η τοποθέτηση του χρήστη στο επίκεντρο της εφαρμογής και η ροή των δεδομένων, ανάλογα με την επιλογή του χρήστη σε μια εκ των κύριων λειτουργιών της εφαρμογής.



3.2 Απαιτήσεις

Η κατανόηση των απαιτήσεων των χρηστών μιας εφαρμογής, ανεξάρτητα από το μέγεθός της, αποτελεί μια από τις κρίσιμότερες φάσεις για την επιτυχή ολοκλήρωση της εφαρμογής. Έχοντας το χρήστη στο επίκεντρο, οι απαιτήσεις καθορίζονται από αυτόν, αλλά συχνά παρουσιάζονται διάφορα προβλήματα από την αυτή την πρώτη κιόλας φάση ανάπτυξης της εφαρμογής.

Αναμφισβήτητα, ο κάθε χρήστης μπορεί να έχει διαφορετικές απαιτήσεις από κάποια εφαρμογή, με αποτέλεσμα οι απαιτήσεις να συγκρούονται. Αυτό μπορεί να συμβαίνει ως αποτέλεσμα του γεγονότος ότι οι χρήστες πολλές φορές δεν ξέρουν τι θέλουν από μια εφαρμογή.

Υπάρχουν αρκετοί τρόποι να περιοριστούν τέτοια προβλήματα, φτάνει να χρησιμοποιηθούν οι κατάλληλες μέθοδοι για κατανόηση και ανάλυση των απαιτήσεων.

Για το λόγο αυτό, επιλέχθηκε η μέθοδος του ερωτηματολογίου, σύμφωνα με την οποία, οι χρήστες παραλαμβάνουν μια σειρά συγκεκριμένων γραπτών ερωτήσεων. Η μέθοδος αυτή παρουσιάζει αρκετά πλεονεκτήματα, αφού είναι πολύ γρήγορη, προσφέρει αποτελέσματα για στατιστική ανάλυση και με τη χρήση της μπορούν να επιτευχθούν ποσοτικά και κάποια ποιοτικά στοιχεία από μεγάλο αριθμό δείγματος. Το ερωτηματολόγιο που δημιουργήθηκε για να αναλυθούν οι απαιτήσεις των χρηστών περιλάμβανε ερωτήσεις «κλειστού» τύπου, δηλαδή ερωτήσεις με σταθερές απαντήσεις, αλλά και «κλειστού» τύπου, δηλαδή ερωτήσεις όπου οι χρήστες ήταν ελεύθεροι να απαντήσουν όπως επιθυμούσαν. Το ερωτηματολόγιο αποτελείται από τα εξής μέρη:

- Βασικές λειτουργίες
- Πληροφόρηση
- Επιπλέον Λειτουργίες

Οι ερωτήσεις «κλειστού» τύπου έδιναν την επιλογή στους ερωτηθέντες να επιλέξουν από μια κλίμακα από το 1 μέχρι το 5, με την πιο κάτω σημασία στην κλίμακα:

- 1: καθόλου
- 2: λίγο
- 3: ούτε λίγο, ούτε πολύ
- 4: πολύ
- 5: πάρα πολύ

Για την εξαγωγή έγκυρων αποτελεσμάτων, μεγάλη σημασία παίζει και το δείγμα χρηστών που θα επιλεγεί. Οπότε, το δείγμα θα πρέπει να είναι αντιπροσωπευτικό. Σε μια προσπάθεια να καλυφθεί ένα ευρύ φάσμα ηλικιών, το ερωτηματολόγιο δόθηκε σε φοιτητές από 18 μέχρι 24 χρονών με εμπειρία του προγράμματος ανταλλαγής φοιτητών Erasmus, και ενήλικες από 25-50 χρονών. Το δείγμα που επιλέχθηκε περιλαμβάνει 30 άτομα, με αναμενόμενες εμπειρίες ταξιδιών, είτε αυτά αφορούσαν εσωτερικό, είτε εξωτερικό τουρισμό.

(Το ερωτηματολόγιο δημιουργήθηκε με τη χρήση Google Docs και παρουσιάζεται μαζί με τα αποτελέσματα του στο Παράρτημα Α)

Συνοψίζοντας τις απαντήσεις που δόθηκαν, μπορούμε να κατηγοριοποιήσουμε τις απαιτήσεις των χρηστών ανάλογα με το βαθμό προτίμησης τους. Στις κατηγορίες αυτές περιλαμβάνονται οι λειτουργίες που προτιμήθηκαν από την πλειοψηφία, αυτές που προτιμήθηκαν από αρκετούς και αυτές που θεωρήθηκε ότι δεν είναι αναγκαίες.

Στην πρώτη κατηγορία ανήκουν τέσσερις λειτουργίες οι οποίες προτιμήθηκαν από την πλειοψηφία. Ήταν αναμενόμενο να επιλεγούν αφού καλύπτουν τις βασικότερες ανάγκες ενός τουρίστα, σε κάποιο μέρος.

Οι λειτουργίες αυτές είναι:

- Διαχωρισμός κατηγοριών ενδιαφέροντος σε λίστα
- Συμπερίληψη κατηγοριών που θεωρούνται βασικές
- Συμπερίληψη όλων των σημείων ενδιαφέροντος σε χάρτη της Κύπρου
- Συμπερίληψη των σημείων ενδιαφέροντος που βρίσκονται σε συγκεκριμένη ακτίνα από το χρήστη

Στη δεύτερη κατηγορία ανήκουν τρεις οι λειτουργίες οι οποίες προτιμήθηκαν από αρκετούς, αλλά δεν ήταν δυνατό να συγκεντρώσουν ικανοποιητικό αριθμό προτιμήσεων. Οι λειτουργίες αυτές έχουν περισσότερο βοηθητικό χαρακτήρα και η χρησιμότητα τους διαφέρει ανάλογα με το κοινό στο οποίο απευθύνονται, δηλαδή σε εσωτερικούς και εξωτερικούς τουρίστες.

Οι λειτουργίες αυτές είναι:

- Συμπερίληψη πληροφοριών κατηγοριών που δεν θεωρούνται άκρως απαραίτητα για την εφαρμογή (π.χ. εστιατόρια, νυχτερινά κέντρα κτλ.)
- Συμπερίληψη πληροφοριών για Μέσα Μαζικής Μεταφοράς
- Δημιουργία κοινωνικού δικτύου εφαρμογής

Τέλος, στην τρίτη κατηγορία ανήκουν οι λειτουργίες που δεν θεωρήθηκαν αναγκαίες, με τους χρήστες να απαντούν ότι η ύπαρξη τους δεν είναι σημαντική.

Οι λειτουργίες αυτές είναι:

- Συμπερίληψη πληροφοριών για TAXI
- Συμπερίληψη πληροφοριών για χώρους διαμονής

3.3 Προδιαγραφές

Με την ολοκλήρωση της διαδικασίας εξεύρεσης και ανάλυσης των απαιτήσεων των χρηστών, λαμβάνοντας υπόψη τα αποτελέσματα που προέκυψαν, καθορίστηκαν οι προδιαγραφές που έπρεπε να πληροί η εφαρμογή.

Καταρχήν, η εφαρμογή είναι συμβατή με κινητές συσκευές που τρέχουν το λειτουργικό σύστημα Android 4.0 (Ice Cream Sandwich) και άνω, εξοπλισμένες με οθόνη αφής, συνδεσιμότητα στο διαδίκτυο, είτε μέσω Wi-Fi είτε μέσω τηλεπικοινωνιακού παροχέα, καθώς και με αισθητήρα GPS.

Σύμφωνα με τις απαιτήσεις που εξάχθηκαν, οι κύριες λειτουργίες της εφαρμογής θα αποτελούνται από:

- Μια λίστα – οδηγό που θα χωρίζει τα σημεία ενδιαφέροντος σε κατηγορίες
- Ένα χάρτη της Κύπρου με ενσωματωμένα όλα τα σημεία ενδιαφέροντος
- Εντοπισμός του χρήστη σε χάρτη της Κύπρου και παρουσίαση σημείων ενδιαφέροντος που βρίσκονται σε συγκεκριμένη ακτίνα γύρω από τον χρήστη.

Οι κύριες λειτουργίες της αυτές προϋποθέτουν κάποιες προδιαγραφές τις οποίες θα πρέπει να πληροί η εφαρμογή οι οποίες καθορίζονται ως εξής:

- Ενσωμάτωση των κατάλληλων εργαλείων για τη διαμόρφωση της διεπαφής και το διαχωρισμό λίστας, και των χαρτών
- Εμφάνιση πληροφοριών σχετικά με τα σημεία ενδιαφέροντος και δυνατότητα επιλογής διαδρομής προς αυτά
- Εμφάνιση τοποθεσίας χρήστη στο χάρτη μέσω GPS
- Χρησιμοποίηση κατάλληλων εργαλείων που προσφέρονται από το Android SDK για τη δημιουργία λίστας
- Ενσωμάτωση Google Maps με την εγκατάσταση των βοηθητικών βιβλιοθηκών Google API και Google Play Services στο Android SDK

3.4 Υλοποίηση

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε η γλώσσα προγραμματισμού Java και το Eclipse IDE (Integrated Development Environment), αφού είχε ενσωματωθεί σε αυτό το Android SDK (Software Development Kit) της Google.

3.4.1 Κύρια συστατικά εφαρμογής

3.4.1.1 Η κλάση Activity

Μια κλάση Activity αποτελεί το κύριο συστατικό σε μια εφαρμογή Android και παρέχει στο χρήστη μια «οθόνη» με την οποία μπορεί να αλληλεπιδράσει για να ολοκληρώσει μια διεργασία (π.χ. να στείλει e-mail, να βγάλει φωτογραφία, να πάρει τηλέφωνο κτλ.). Σε κάθε Activity δίνεται ένα παράθυρο στο οποίο εμφανίζεται η διεπαφή με την οποία αλληλεπιδρά ο χρήστης και συνήθως καταλαμβάνει ολόκληρη την επιφάνεια της οθόνης της κινητής συσκευής, αλλά μπορεί να είναι και μικρότερη και να βρίσκεται πάνω από άλλα παράθυρα.

Μια εφαρμογή μπορεί να αποτελείται από πολλές τέτοιες κλάσεις που δεν είναι συνδεδεμένοι μεταξύ τους απαραίτητα. Συνήθως, μια Activity καθορίζεται ως η κύρια κλάση και ξεκινά με την εκκίνηση της εφαρμογής, και εξακολουθεί να υπάρχει μέχρι τον τερματισμό της εφαρμογής. Κάθε Activity μπορεί να καλέσει με τη σειρά της άλλες Activities και συνήθως κάθε Activity συνδέεται με την ολοκλήρωση ενός σκοπού.

Η κλήση μιας Activity από μια άλλη γίνεται μέσω Intents. Ένα Intent αποτελεί μια περιγραφή της Activity που θέλουμε να ξεκινήσουμε. Μπορούμε να θέσουμε επακριβώς ποια Activity θέλουμε να ξεκινήσει, είτε να θέσουμε κριτήρια ανάλογα με την διεργασία που θέλουμε να εκτελέσουμε και να αφήσουμε το λειτουργικό σύστημα να αποφασίσει ποια Activity θα πρέπει να ξεκινήσει.

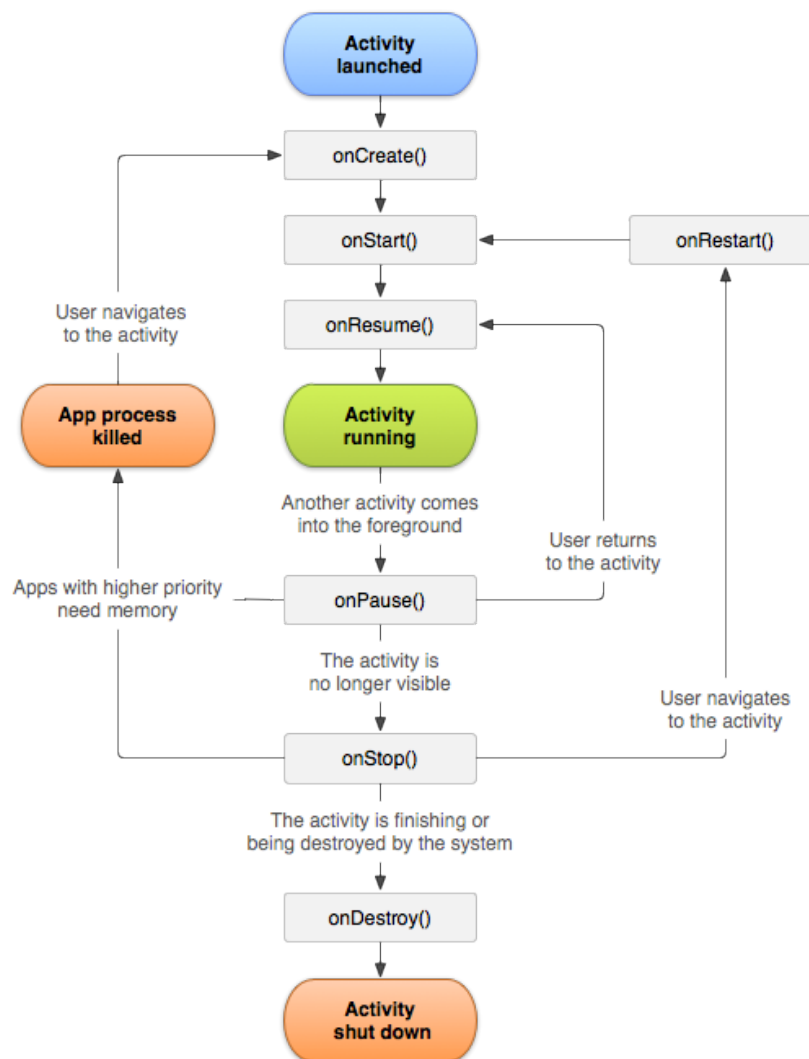
```
Intent intent = new Intent(this,SignInActivity.class);
startActivity(intent);
```

Πιο πάνω καθορίζεται επακριβώς ότι ξεκινά η `SignInActivity.class`

Μια άλλη περίπτωση είναι να θέσουμε τα κριτήρια για την εκκίνηση μιας Activity, και να αφήσουμε το λειτουργικό σύστημα ποια πρέπει να ξεκινήσει. Για παράδειγμα, μπορεί να θέλουμε να στείλουμε ένα e-mail ή ένα SMS οπότεν θέτουμε κριτήρια και το λειτουργικό θα ζητήσει την επιλογή της κατάλληλης προς εκκίνηση εφαρμογής.

```
Intent intent = new Intent(Intent.ACTION_SEND);  
intent.putExtra(Intent.EXTRA_EMAIL, recipientArray);  
startActivity(intent);
```

Κάθε κλάση Activity έχει ένα κύκλο ζωής, που αποτελείται από διάφορες μεθόδους οι οποίες καλούνται την κατάλληλη χρονική στιγμή και ξεκινά με την δημιουργία της κλάσης, και τελειώνει με την καταστροφή της [6]



Εικόνα 3.4.1.1.1 Ο κύκλος ζωής μιας κλάσης Activity

3.4.1.2 Η κλάση Fragment

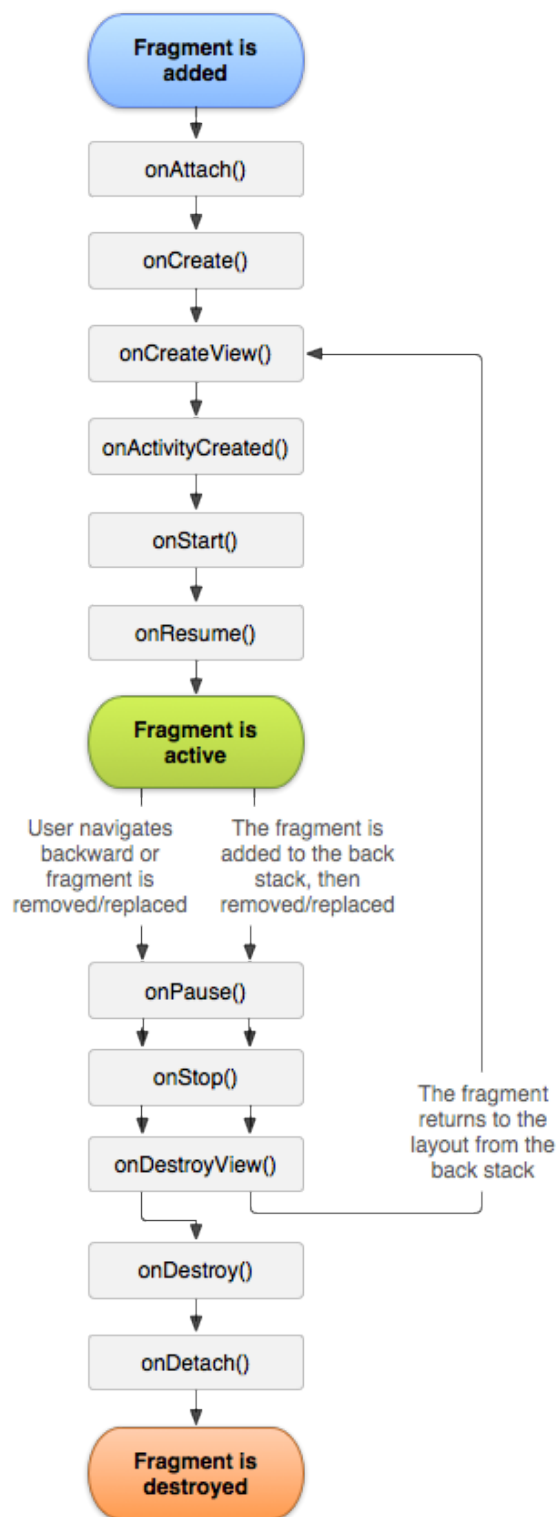
Ένα Fragment αντιπροσωπεύει μια συμπεριφορά ή ένα κομμάτι διεπαφής σε μια Activity. Μπορούμε να συνδυάσουμε την παρουσία πολλαπλών Fragments σε μια Activity, με το κάθε ένα να εκτελεί ένα συγκεκριμένο σκοπό. Τα Fragments είναι σαν επαναχρησιμοποιήσιμα εργαλεία με δικό τους κύκλο ζωής και μπορούν να ενσωματωθούν σε μια Activity καθώς αυτή τρέχει για να εκπληρώσουν μια διεργασία.

Τα Fragments για να τρέξουν πρέπει οπωσδήποτε να ενσωματωθούν σε κάποια Activity και ο κύκλος ζωής τους επηρεάζεται άμεσα από τον κύκλο ζωής της Activity στην οποία ανήκουν. Τα Fragments μπορούν να ενσωματωθούν σε μια Activity, είτε στατικά, μέσω δήλωσης τους στο σε ένα αρχείο XML που περιγράφει την Activity στην οποία θα ανήκουν, είτε δυναμικά, όταν τρέχει η εφαρμογή, μέσω του κώδικά της. Κατά τη διάρκεια ζωής τους, τα Fragments υπόκεινται σε «συναλλαγές», όσον αφορά το ποιό Fragment ορατό κάθε φορά.

```
FragmentManager fragmentManager = getFragmentManager()  
FragmentTransaction fragmentTransaction =  
fragmentManager.beginTransaction();
```

```
ExampleFragment fragment = new ExampleFragment();  
fragmentTransaction.add(R.id.fragment_container,  
fragment);  
fragmentTransaction.commit();
```

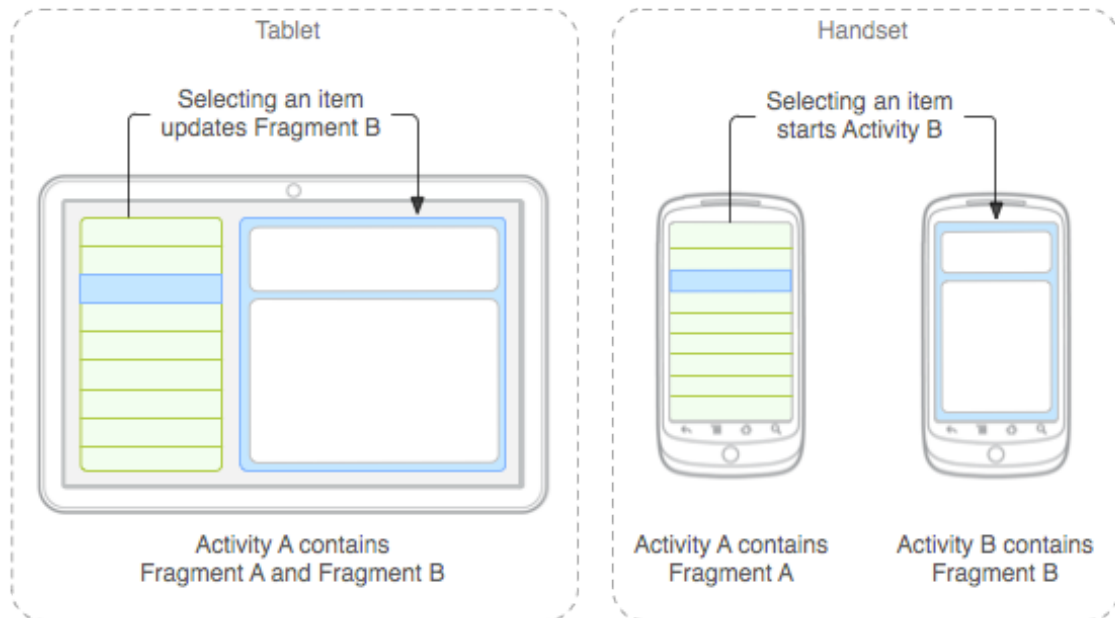
Μέσα από τις συναλλαγές αυτές, γίνονται οι εναλλαγές των Fragments στην κάθε Activity χωρίς να υπάρχει η ανάγκη να ξεκινούν νέες Activities, αν αυτό δεν θεωρείται απολύτως αναγκαίο. Για το λόγο αυτό η Google μετά την έκδοση το Android 3.0 Honeycomb προτείνει τη χρήση των Fragments ενώ θεωρεί παραδοσιακές μεθόδους διαχείρισης των εισόδων του χρήστη ως ξεπερασμένες. Ειδικότερα, όταν η εφαρμογή περιλαμβάνει Tabs επιλογών, η χρήση των Fragments θεωρείται επιβεβλημένη για την ανάπτυξη μιας μοντέρνας εφαρμογής.



Εικόνα 3.4.1.2.1 Ο κύκλος ζωής μιας κλάσης Fragment

Επιπρόσθετα, η προσαρμοστικότητα και η επαναχρησιμοποίηση των Fragments διευκολύνει και την παρουσίαση εφαρμογών σε μια πλειάδα από κινητές συσκευές με

διαφορετικό μέγεθος οθόνης, αφού μπορούν να προσαρμοστούν ανάλογα με το μέγεθος και τον προσανατολισμό της οθόνης.



Εικόνα 3.4.1.2.2 Η προσαρμοστικότητα των Fragments σε κινητές συσκευές με διαφορετικό μέγεθος οθόνης

Για τους πιο πάνω λόγους επιλέχθηκαν τα συγκεκριμένα συστατικά ως τα θεμέλια αυτής της εφαρμογής. [7]

3.4.1.3 AndroidManifest.xml

Κάθε εφαρμογή Android για να μπορεί να τρέξει πρέπει να έχει ένα AndroidManifest.xml αρχείο μέσα στην **root directory**. Το αρχείο αυτό παρουσιάζει στο λειτουργικό σύστημα τις ουσιώδεις πληροφορίες που αφορούν την εφαρμογή, πληροφορίες που είναι αναγκαίες στο λειτουργικό σύστημα για να τρέξει η εφαρμογή. Μεταξύ άλλων, το AndroidManifest.xml είναι υπεύθυνο για τις πιο κάτω λειτουργίες:

- Ονομάζει το Java Package της εφαρμογής, που λειτουργεί ως μοναδικό χαρακτηριστικό της εφαρμογής

- Περιγράφει τα κύρια συστατικά της εφαρμογής, όπως τις κλάσεις Activity και άλλα σημαντικά συστατικά, καθώς και τις περιπτώσεις στις οποίες μπορούν να ενεργοποιηθούν
- Καθορίζει ποιες διεργασίες θα χρησιμοποιηθούν για τα κύρια συστατικά της εφαρμογής
- Δηλώνει τις άδειες (Permissions) που χρειάζεται η εφαρμογή από το λειτουργικό σύστημα, όπως για παράδειγμα πρόσβαση σε δεδομένα του χρήστη ή σύνδεση στο διαδίκτυο
- Καθορίζει την ελάχιστη έκδοση του λειτουργικού με την οποία είναι συμβατή η εφαρμογή
- Δηλώνει τις βιβλιοθήκες που θα χρησιμοποιηθούν [8]

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.thesis.lovecyprus"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-feature
        android:glEsVersion="0x00020000"
        android:required="true"/>

    <permission
        android:name="com.thesis.lovecyprus.permission.MAPS_RECEIVE"
        android:protectionLevel="signature"/>
    <uses-permission android:name="com.thesis.lovecyprus.permission.MAPS_RECEIVE"/>

    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission
        android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>

    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>

    <uses-sdk
        android:minSdkVersion="14"
        android:targetSdkVersion="17" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.thesis.lovecyprus.SplashActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
```

```

<activity
    android:name="com.thesis.lovecyprus.MainActivity"
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
    </intent-filter>
</activity>
<activity
    android:name="com.thesis.lovecyprus.InfoActivity"
    android:label="@string/app_name" >
</activity>

    <meta-data
        android:name="com.google.android.maps.v2.API_KEY"
        android:value="AIzaSyDKLw807v7NrIcVgWBorcTlps08JGMEhMc"/>

</application>

</manifest>

```

Εικόνα 3.4.1.3.1 AndroidManifest.xml για την εφαρμογή LoveCyprus

3.4.2 Βάση Δεδομένων SQLite

Για την μόνιμη αποθήκευση των δεδομένων των σημείων ενδιαφέροντος που θα περιληφθούν στην εφαρμογή, χρησιμοποιήθηκε μια βάση δεδομένων (ΒΔ) SQLite, η οποία ενσωματώνεται στο Android SDK. [9]

Με τη δημιουργία της ΒΔ στην εφαρμογή, έγινε ευκολότερη η αποθήκευση και η ανάκτηση των δεδομένων που είναι αναγκαία για την εφαρμογή. Επιπρόσθετα, οι προοπτικές επέκτασης της εφαρμογής αυξάνονται κατακόρυφα, αφού θα υπάρχει ήδη μια δομή και η επέκταση της δεν είναι ιδιαίτερα δύσκολη.

Για τις ανάγκες της εφαρμογής, εκτός από τα διαδικαστικά που χρειάζονται για την ύπαρξη της βάσης δεδομένων, δημιουργήθηκε ένας πίνακας ο οποίος θα αποθηκεύει τις πληροφορίες για τα σημεία ενδιαφέροντος.

ID	Name	City	Type	Latitude	Longitude	ImageURL	Description	Tel
----	------	------	------	----------	-----------	----------	-------------	-----

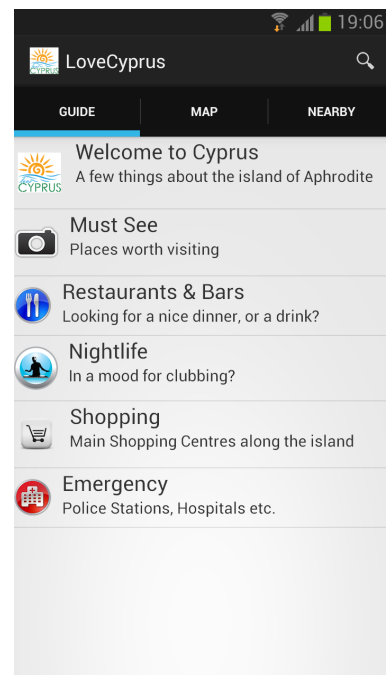
Πίνακας 3.4.2.1 Δομή Πίνακα Places

3.4.3 Δομή εφαρμογής

Κατά την εκκίνηση της εφαρμογής παρουσιάζεται στο χρήστη μια εικόνα με φωτογραφίες από διάφορα μέρη της Κύπρου. Η παρουσίαση αυτή διαρκεί 3 δευτερόλεπτα. Η εικόνα περιέχεται σε μια κλάση, την `SplashScreenActivity.java`, τύπου `Activity`, που ξεκινά με την εκκίνηση της εφαρμογής και τερματίζεται 3 δευτερόλεπτα αργότερα. Ακολούθως φορτώνεται η κλάση `MainActivity.java`, επίσης τύπου `Activity` που τρέχει καθ' όλη τη διάρκεια της ζωής της εφαρμογής. Η κύρια δομή της διεπαφής που εμφανίζεται στην `MainActivity` αποτελείται από `Tabs`. Τα `Tabs`, αν και παλαιότερα εμφανίζονταν ανεξάρτητα σε μια εφαρμογή με δικές τους υλοποιήσεις, στις πρόσφατες εκδόσεις του λειτουργικού `Android` και ειδικότερα μετά την έκδοση `Android 3.0 Honeycomb`, εμφανίζονται ενσωματωμένα στο `ActionBar` της εφαρμογής, δηλαδή στο εργαλείο στο οποίο εμφανίζονται διάφορα εικονίδια για εκπλήρωση διεργασιών.



Εικόνα 3.4.3.1 SplashScreenActivity

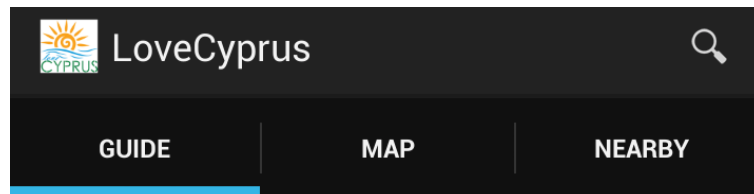


Εικόνα 3.4.3.2 MainActivity

Στις πιο πάνω εικόνες παρουσιάζονται οι δύο πρώτες οθόνες που εμφανίζονται με την εκκίνηση της εφαρμογής. Αρχικά, εμφανίζεται η `SplashScreenActivity` για διάρκεια τριών δευτερολέπτων και στη συνέχεια καταστρέφεται, ενώ εμφανίζεται η `MainActivity` ενόσω τρέχει η εφαρμογή. Εδώ ενεργοποιείται και η βάση δεδομένων, με ελέγχους για τα δεδομένα της, αν αυτά υπάρχουν ή αν χρειάζονται ανανέωση.

```
datasource = new PlaceDataSource(this);
datasource.open();
```

Όπως παρατηρούμε, η MainActivity περιέχει και το μενού επιλογών στο ActionBar της.



Εικόνα 3.4.3.3 ActionBar εφαρμογής

Στο ActionBar παρατηρούμε το λογότυπο της εφαρμογής, και το κουμπί αναζήτησης, καθώς και ενσωματωμένα 3 Tabs που περιέχουν τις κυριότερες λειτουργίες της εφαρμογής.

Ο Action Bar καλείται με την εκκίνηση της MainActivity class και η ενσωμάτωση των tabs γίνεται στη μέθοδο onCreate() της κλάσης.

```
ActionBar actionBar;
// Set up the action bar.
actionBar = getSupportActionBar();

// Specify that tabs should be displayed in the action bar.
actionBar.setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);

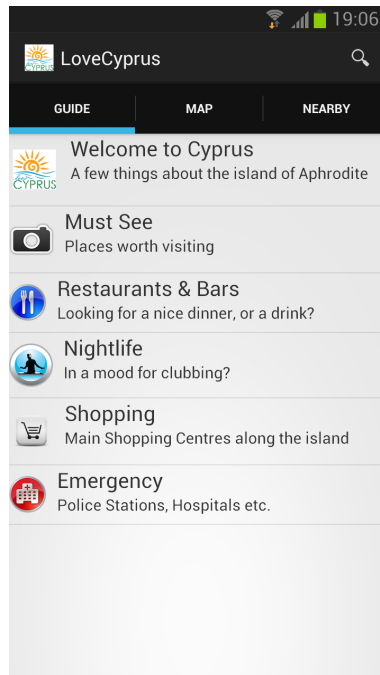
//Add Tabs
for (int i = 0; i < 4; i++) {
    if (i == 0) {
        actionBar.addTab(actionBar.newTab().setText(R.string.title_section1).
            setIcon(R.drawable.ic_launcher).setTabListener(this));
    }
    if (i == 1) {
        actionBar.addTab(actionBar.newTab().setText(R.string.title_section2)
            . setIcon(R.drawable.ic_map).setTabListener(this));
    }

    if (i == 2) {
        actionBar.addTab(actionBar.newTab().setText(R.string.title_section3).
            setIcon(R.drawable.ic_nearby).setTabListener(this));
    }
}
```

Βλέπουμε ότι με την εμφάνιση της MainActivity είναι προεπιλεγμένο το 1^ο Tab με τη λίστα – οδηγό. Αυτό έχει ως αποτέλεσμα την αυτόματη κλήση στη μέθοδο onTabSelected() με όρισμα 0, δηλαδή το δείκτη του 1^{ου} Tab. Έτσι, καλείται το ListFragment.java, το 1^ο Fragment που περιέχει τη λίστα με τις κατηγορίες

ενδιαφέροντος. Η πλοήγηση στην εφαρμογή γίνεται με το άγγιγμα του κάθε Tab. Έχει καταργηθεί η πλοήγηση με την ολίσθηση δακτύλου καθώς θα συγκρουόταν με την πλοήγηση ολίσθησης στους χάρτες που περιέχονται στην εφαρμογή.

```
public void onTabSelected(Tab tab, FragmentTransaction ft) {  
    // GUIDE TAB  
    //empty back stack in case there is a fragment in it  
    fm.popBackStack(null, FragmentManager.POP_BACK_STACK_INCLUSIVE){  
  
    if (tab.getText().equals("Guide")) {  
  
        if (guidefrag == null) {  
  
            guidefrag = new GuideFragment();  
  
            fm.beginTransaction().replace(R.id.fragment_container, guidefrag)  
  
                                .commit();  
  
        }  
    }  
}
```

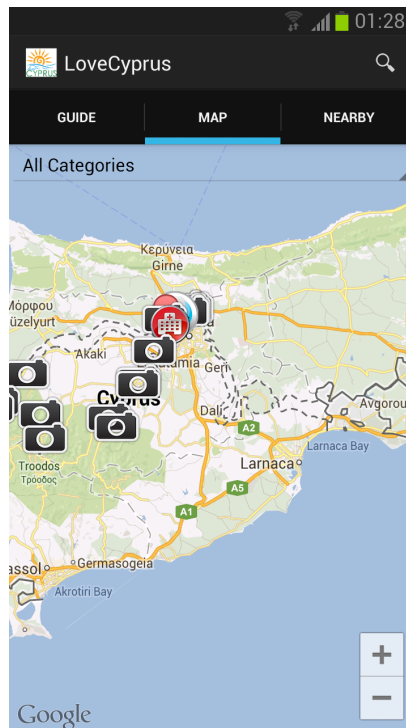


Εικόνα 3.4.3.4 Η MainActivity με ενσωματωμένο το ListFragmer

Ακολούθως, ο χρήστης μπορεί να περιηγηθεί στις διάφορες κατηγορίες της λίστας. Η επιλογή οποιασδήποτε κατηγορίας από τη λίστα θα προκαλέσει την αντικατάσταση του ListFragment με το ανάλογο Fragment που αντιστοιχεί στην κατηγορία που θα εμφανιστεί. Η αντικατάσταση του ListFragment θα προκαλέσει επίσης την τοποθέτηση

του σε στοίβα, την BackStack, που χρησιμεύει στην ομαλή πλοήγηση στα Fragments επαναφορά του ListFragment όταν ο χρήστης επιλέξει το κουμπί back του κινητού του.

Με την επιλογή του Tab “Map”, εμφανίζεται ολόκληρος χάρτης της Κύπρου με σημειωμένα σε διάφορες συντεταγμένες τα σημεία ενδιαφέροντος που περιέχονται στη βάση δεδομένων.

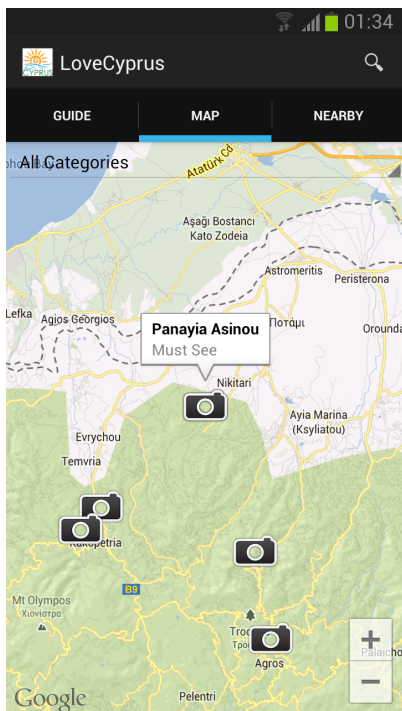


Εικόνα 3.4.3.5 Η MainActivity με ενσωματωμένο το CyprusMap Fragment

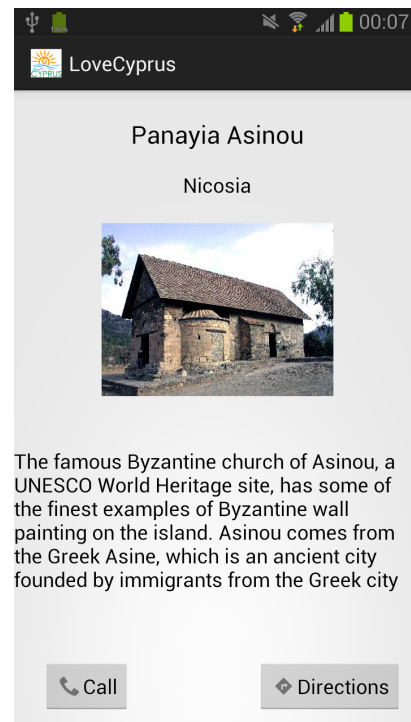
Η εναλλαγή των Fragments γίνεται στην προαναφερθείσα μέθοδο onTabSelected με τον πιο κάτω τρόπο.

```
if (tab.getText().equals("Map")) {  
  
    cyprusmap = new CyprusMap();  
  
    fm.beginTransaction().replace(R.id.fragment_container, cyprusmap).commit();  
}
```

Εδώ ο χρήστης έχει τη δυνατότητα να επιλέξει το σημείο που τον ενδιαφέρει και να πληροφορηθεί γι' αυτό, όπως φαίνεται στις εικόνες πιο κάτω



**Εικόνα 3.4.3.6 Παρουσίαση σημείου
Ενδιαφέροντος**



**Εικόνα 3.4.3.7 Πληροφορίες Σημείου
Ενδιαφέροντος**

Ανάλογα με το σημείο που έχει επιλεγεί, με την επιλογή του εικονιδίου πληροφοριών που περιλαμβάνει το όνομα και την κατηγορία του σημείου ενδιαφέροντος, εκτελείται η μέθοδος `onInfoWindowClicked()` του `Fragment` το οποίο είναι ενεργοποιημένο. Η εφαρμογή στέλνει ένα επερώτημα στη βάση δεδομένων και παίρνει τις πληροφορίες για το συγκεκριμένο σημείο. Έπειτα, δημιουργείται ένα αντικείμενο της κλάσης `InfoActivity` με το όνομα, την πόλη, μια φωτογραφία και την περιγραφή το σημείου ενδιαφέροντος, με δυνατότητα κλήσης για πληροφορίες σχετικά με το συγκεκριμένο σημείο, ή την παροχή οδηγιών πρόσβασης. Για τη μεταφορά των δεδομένων του σημείου από τη μια `Activity` στην άλλη χρησιμοποιείται `Intent` και `Bundle`.

```
public void onInfoWindowClick(Marker marker) {

    Intent intent = new Intent(this.getActivity(), InfoActivity.class);
    Place place = MainActivity.datasource.getPlaceFromMarker(marker);
    Bundle bundle = new Bundle();
    bundle.putLong("id", place.getId());
```

```

        bundle.putString("name",place.getName());
        bundle.putString("city",place.getCity());
        bundle.putString("type", place.getType());
        bundle.putDouble("lat", place.getLatitude());
        bundle.putDouble("long",place.getLongitude());
        bundle.putDouble("currentlat",this.getLatitude() );
        bundle.putDouble("currentlong", this.getLongitude());
        bundle.putString("imageurl", place.getImageurl());
        bundle.putString("description", place.getDescription());
        intent.putExtras(bundle);
        startActivity(intent);
    }

```

Μέσα από την οθόνη κάποιου σημείου εφαρμογής ο χρήστης μπορεί να μάθει χρήσιμες πληροφορίες για αυτό, αλλά και να πάρει οδηγίες πρόσβασης προς αυτό μέσω των Google Maps.



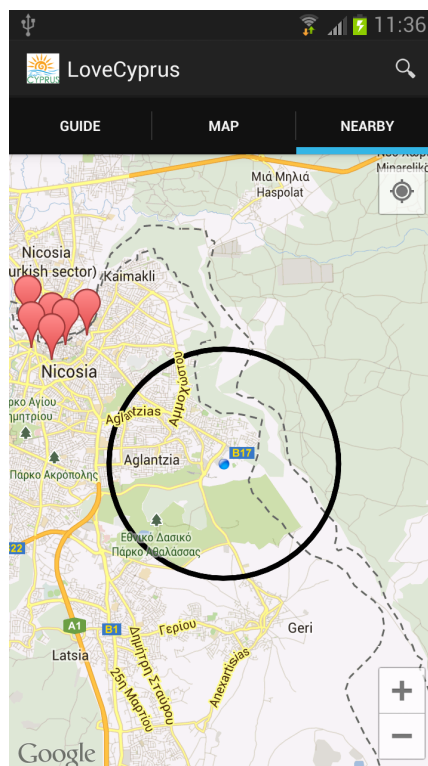
Εικόνα 3.4.3.8 Οδηγίες πρόσβασης σε σημείο ενδιαφέροντος

Για την ολοκλήρωση προσφοράς οδηγιών πρόσβασης σε σημείο ενδιαφέροντος, έγινε συγκέντρωση της τοποθεσίας του χρήστη, των συντεταγμένων του σημείου

ενδιαφέροντος και κλήθηκε η εφαρμογή Google Maps για να εμφανίσει τη διαδρομή, καθώς η Google απαγορεύει τη χρήση οδηγιών πρόσβασης σε άλλες εφαρμογές.

```
button.setOnClickListener(new OnClickListener() {  
    public void onClick(View v)  
    {  
        Intent i = new Intent(android.content.Intent.ACTION_VIEW,  
            Uri.parse("http://maps.google.com/maps?saddr=" +  
currentlat+", "+currentlong  
                                + "&daddr=" +  
latitude+", "+longitude));  
        startActivity(i);  
    }  
});
```

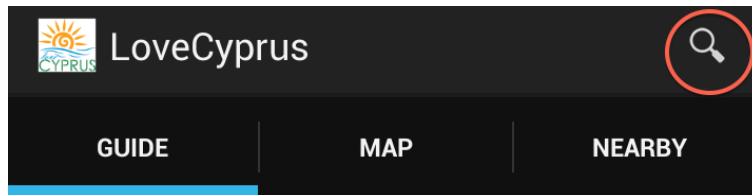
Επιπλέον, ο χρήστης με την επιλογή της τρίτης κύριας λειτουργίας, δηλαδή το tab “Nearby”, προκαλείται η αντικατάσταση του τρέχοντος Fragment με το Nearby Fragment. Σε αυτή τη λειτουργία, παρέχεται στο χρήστη η δυνατότητα να εντοπίσει γεωγραφικά τη θέση του και να επιλέξει να πληροφορηθεί για κοντινά προς αυτόν σημεία ενδιαφέροντος, εντός 3 χιλιομέτρων η οποία φαίνεται με ένα μαύρο κύκλο.



Εικόνα 3.4.3.9 Γεωγραφικός εντοπισμός χρήστη

Μέσω του γεωγραφικού εντοπισμού ο χρήστης μπορεί όπως και προηγουμένως να λάβει οδηγίες πρόσβασης προς τα σημεία που τον ενδιαφέρουν

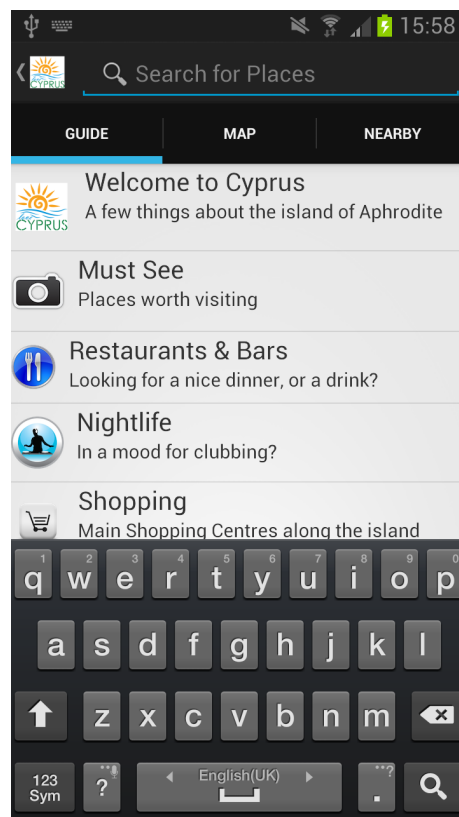
Επιπλέον, η εφαρμογή δίνει στο χρήστη την επιλογή αναζήτησης σημείων ενδιαφέροντος με βάση το όνομά τους, με τη δυνατότητα auto complete, μέσω του πεδίου αναζήτησης που βρίσκεται στο ActionBar της εφαρμογής.



Εικόνα 3.4.3.11 Πεδίο αναζήτησης

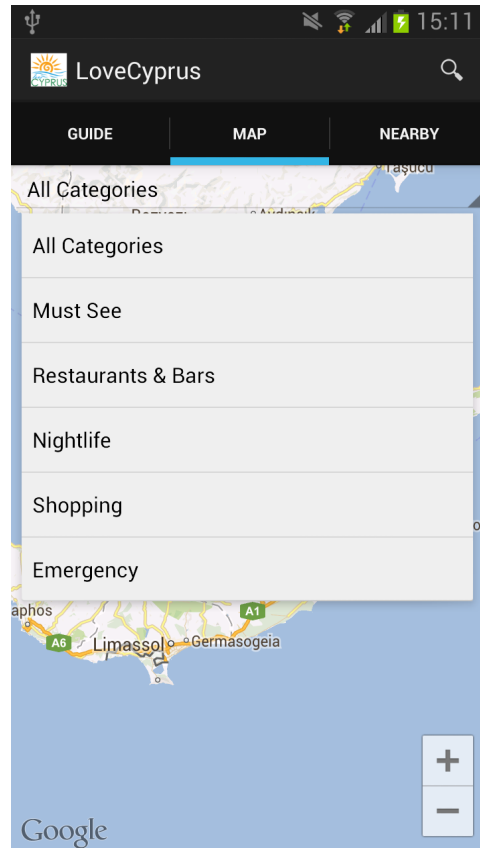
Με τον τρόπο αυτό οι χρήστες έχουν τη δυνατότητα να αναζητήσουν από μόνοι τους τα σημεία που τους ενδιαφέρουν, χωρίς να χρειαστεί να ψάχνουν στο χάρτη, πράγμα που κάνει την πλοήγηση και την ολοκλήρωση διεργασιών εκ μέρους του χρήστη πολύ γρηγορότερη και πιο αποδοτική.

Με την επιλογή του εικονιδίου αναζήτησης, το εικονίδιο επεκτείνεται και ενεργοποιείται το εικονικό πληκτρολόγιο για την διενέργεια αναζήτησης.



Εικόνα 3.4.3.12 Αναζήτηση σημείου ενδιαφέροντος

Τέλος, ο χρήστης μπορεί να φιλτράρει τα αποτελέσματα για τα σημεία ενδιαφέροντος τα οποία εμφανίζονται στους χάρτες .



Εικόνα 3.4.13 Φιλτράρισμα σημείων ενδιαφέροντος στους χάρτες

Η λειτουργία αυτή προσφέρεται και σε συνδυασμό με τη δυνατότητα του γεωγραφικού εντοπισμού του χρήστη και την εισήγηση κοντινών σημείων ενδιαφέροντος, όπως φαίνεται και στις πιο κάτω εικόνες

Κεφάλαιο 4

Αξιολόγηση

4.1 Μέθοδος αξιολόγησης	31
4.2 Αποτελέσματα Αξιολόγησης	32

4.1 Μέθοδος Αξιολόγησης

Για την αξιολόγηση της εφαρμογής χρησιμοποιήθηκαν δύο μέθοδοι αξιολόγησης, η ευρετική αξιολόγηση με ειδικούς ευχρηστίας και η πειραματική αξιολόγηση με χρήστες της εφαρμογής.

Η ευρετική αξιολόγηση αποτελεί μια μέθοδο εξέτασης της εφαρμογής από ειδικούς ευχρηστίας οι οποίοι βασίζονται σε αρχές ευχρηστίας αλλά και μια ομάδα τυπικών κανόνων, γνωστών ως «Οι 10 κανόνες του Nielsen». Οι παρατηρήσεις που καταγράφουν οι ειδικοί ευχρηστίας καταγράφονται και ομαδοποιούνται ανά κατηγορία. Η μέθοδος αξιολόγησης αυτή έχει αποδειχθεί ότι είναι πολύ αποδοτική και αποτελεσματική.

Τυπικοί κανόνες ευρετικής αξιολόγησης	
1	Ορατότητα κατάστασης του συστήματος
2	Συμφωνία του συστήματος με τον έξω κόσμο
3	Έλεγχος και ελευθερία χρήστη
4	Συνοχή και συμμόρφωση έναντι προτύπων
5	Πρόληψη σφαλμάτων
6	Διευκόλυνση χρηστών ως προς τον εντοπισμό, τη διάγνωση και τη διόρθωση σφαλμάτων
7	Αναγνώριση και όχι ανάκληση
8	Προσαρμοστικότητα και αποδοτικότητα χρήσης
9	Βοήθεια και εγχειρίδια
10	Καλαισθητικός σχεδιασμός (κομψότητα, ικανοποίηση)

Πίνακας 4.1.1 Κανόνες ευρετικής αξιολόγησης

Από την άλλη, η πειραματική αξιολόγηση βασίζεται σε δύο σημεία, στη γενική σχεδίαση των οθονών του συστήματος αλλά και τη ροή μηνυμάτων και ενεργειών που απαιτούνται για να ολοκληρωθεί μια συγκεκριμένη διεργασία.

Στην αρχή ο χρήστης πρέπει να είναι ενημερωμένος για τις διεργασίες που θα κληθεί να ολοκληρώσει. Σε αυτή τη διαδικασία ο αξιολογητής είναι παρόν για τυχόν διευκρινίσεις και με τη χρήση του πρωτοκόλλου ομιλούντων υποκειμένων οι ειδικοί ευχρηστίας παροτρύνουν τους χρήστες να εξωτερικεύσουν τις σκέψεις και τα συναισθήματά τους. Η μέθοδος αυτή είναι πολύ αποτελεσματική και απαιτεί ελάχιστους πόρους.

4.2 Αποτελέσματα Αξιολόγησης

Κατα την ευρετική αξιολόγηση, τρεις ειδικοί ευχρηστίας ήρθαν σε επαφή με το σύστημα και κατέγραψαν τις όποιες παρατηρήσεις είχαν. Αυτές, ομαδοποιήθηκαν σε πέντε κατηγορίες, ανάλογα με τη σημαντικότητα των παρατηρήσεων

Αισθητικά λάθη

- Θα μπορούσαν να χρησιμοποιηθούν διαφορετικά χρώματα στις λίστες της εφαρμογής, που να θυμίζουν περισσότερο τις επιλογές του λειτουργικού συστήματος
- Τα εικονίδια των σημείων ενδιαφέροντος δεν έχουν συνοχή ως προς το σχήμα τους

Μικρά λάθη

- Η κατηγορία «Must See» δεν δίνει αρκετές πληροφορίες ως προς το περιεχόμενο της και θα μπορούσε να διασπαστεί σε υποκατηγορίες που να αφορούν πολιτισμό, αθλητισμό, παραλίες κτλ.

Μεσαία λάθη

- Η επιλογή φιλτραρίσματος στους χάρτες με τα σημεία ενδιαφέροντος δεν είναι εύκολα ορατή

Μεγάλα λάθη δεν παρατηρήθηκαν από τους ειδικούς ευχρηστίας. Γενικότερα η δομή της εφαρμογής σχολιάστηκε θετικά, πέραν των παρατηρήσεων που παρατηρήθηκαν, καθώς οι κύριες λειτουργίες της εφαρμογής ήταν ξεκάθαρα χωρισμένες μέσω των Tabs.

Κατα τη διάρκεια της πειραματικής αξιολόγησης, δόθηκε στους χρήστες μια λίστα με τις διεργασίες που έπρεπε να εκτελέσουν χρησιμοποιώντας την εφαρμογή. Καθ' όλη τη διάρκεια της πειραματικής αξιολόγησης, παρότρυνα τους χρήστες να εξωτερικεύουν τις σκέψεις και τα συναισθήματά τους, τα οποία καταγράφονταν.

Χρήστης 1

Ο χρήστης σκεφτόταν για κάποιο χρονικό διάστημα σε ποια κατηγορία ανήκει το αξιοθέατο «Παναγία της Ασίνου», αφού δεν υπήρχε κατηγορία «εκκλησίες» ή «πολιτισμός», αλλά εντασσόταν στην κατηγορία «Must See». Στο χάρτη παρουσίασης όλων των σημείων ενδιαφέροντος ο χρήστης ανέμενε τον εντοπισμό του από την εφαρμογή, ενώ αυτή η λειτουργία εντάσσεται στο χάρτη με τα κοντινά προς το χρήστη σημεία ενδιαφέροντος.

Χρήστης 2

Ο χρήστης, όσον αφορά τη διεργασία εύρεσης πληροφοριών για την «Παναγία της Ασίνου», προσπάθησε να την ολοκληρώσει μέσω της δυνατότητας αναζήτησης, παρά από τη λίστα – οδηγό. Αυτό φανερώνει και τις πολλαπλές επιλογές που προσφέρει η εφαρμογή προς ολοκλήρωση μιας διεργασίας. Επιπρόσθετα, δεν αντιλήφθηκε σε ποια κατηγορία ανήκε το αξιοθέατο «Παναγία της Ασίνου». Τέλος, το εργαλείο φιλτραρίσματος στους χάρτες δεν έγινε αντιληπτό από την αρχή.

Χρήστης 3

Ο χρήστης μετά την εύρεση του αξιοθέατο «Παναγία της Ασίνου», δεν γνώριζε σε ποια κατηγορία βρισκόταν για να περάσει στην επόμενη αναζήτηση και χρησιμοποίησε ενέργειες που δεν υπάρχουν στις πρόσφατες εκδόσεις του λειτουργικού Android. Κατά την εμφάνιση των πληροφοριών από το χάρτη, χρησιμοποίησε διπλό κλικ στο

παράθυρο πληροφοριών. Τέλος, δεν αντιλήφθηκε από την αρχή την ύπαρξη εργαλείου φιλτραρίσματος στους χάρτες της εφαρμογής.

Κεφάλαιο 5

Επίλογος

5.1 Σύνοψη	35
5.2 Μελλοντική Εργασία	35
5.3 Εμπειρίες που Αποκτήθηκαν	37

5.1 Σύνοψη

Στην παρούσα διπλωματική εργασία ασχολήθηκα με τον τομέα ανάπτυξης εφαρμογών σε κινητές συσκευές με λειτουργικό Android, και συγκεκριμένα με την ανάπτυξη μιας Android εφαρμογής με υπηρεσίες που να εξυπηρετούν το τουριστικό κοινό της Κύπρου.

Αρχικά, αναφέρεται μια μικρή εισαγωγή στο τι αναμένεται να υλοποιηθεί στην εφαρμογή, αλλά και η δομή της συγκεκριμένης εργασίας. Έπειτα, ακολουθεί μια σύντομη περιγραφή των εννοιών που θα ακολουθήσουν και ακολούθως η περιγραφή της εφαρμογής με το διαχωρισμό των φάσεων ανάπτυξης της. Στη συνέχεια περιγράφεται η διαδικασία αξιολόγησης.

5.2 Μελλοντική Εργασία

Όπως κάθε εργασία, έτσι και αυτή η εφαρμογή έχει αρκετά περιθώρια βελτίωσης, ειδικά με τη ραγδαία ανάπτυξη τόσο των «έξυπνων» κινητών συσκευών και των δυνατοτήτων που προσφέρουν στις εφαρμογές που υποστηρίζουν, όσο και του λειτουργικού συστήματος Android. Οι συνεχείς αλλαγές με ενημερώσεις της εφαρμογής είναι ένα σύνηθες φαινόμενο και μέσω αυτού του τρόπου, η εφαρμογή μπορεί να ικανοποιήσει νέες πιθανές ανάγκες του τουριστικού κοινού του νησιού. Οι προοπτικές εξέλιξης της εφαρμογής είναι τεράστιες, αν αναλογιστεί κανείς τι μπορεί να κάνει σήμερα με τις εφαρμογές κινητών τηλεφώνων και ειδικότερα αυτές του

λειτουργικού συστήματος Android. Με τον εμπλουτισμό της εφαρμογής με νέα χαρακτηριστικά και λειτουργίες πιθανό να αυξήσει κατακόρυφα τη δημοτικότητά της, μιας και η Κύπρος είναι ένας αρκετά δημοφιλής τουριστικός προορισμός, ειδικότερα το καλοκαίρι.

Καταρχήν θα ήταν ορθό να σημειώσουμε ότι οι νέες ανάγκες των χρηστών μπορούν να γίνουν αντιληπτές μέσα από την επανατροφοδότηση που θα παρέχουν οι χρήστες της εφαρμογής. Η επανατροφοδότηση αυτή παρέχεται από το Google Play, την πλατφόρμα διανομής της Google για τις εφαρμογές Android μέσω στατιστικών στοιχείων, βαθμολογιών αλλά και σχολίων των χρηστών. Λαμβάνοντας υπόψη τα στοιχεία που θα προκύψουν, μπορούμε να εξάγουμε ασφαλή συμπεράσματα για την μελλοντική εξέλιξη της εφαρμογής, με την πιθανότητα μεγαλύτερης επιτυχίας να βρίσκεται σε μεγάλα ποσοστά, αφού οι εισηγήσεις θα προέρχονται από τους ίδιους τους χρήστες.

Επιπρόσθετα, έχοντας εις γνώση τις τάσεις της τεχνολογίας και του κοινού, πιστεύω ότι η ενσωμάτωση λειτουργίας διαμοιρασμού εμπειριών ή τοποθεσιών σε κοινωνικά δίκτυα, μέσω της εφαρμογής, θα ήταν μια πολύ πετυχημένη λειτουργία για την εξέλιξη της. Συγκεκριμένα, θα ήταν πολύ χρήσιμο ο χρήστης να μοιράζεται με τους φίλους του στα κοινωνικά δίκτυα στα οποία ανήκει, τις εμπειρίες του ή τα μέρη που επισκέφθηκε στο νησί. Με αυτό τον τρόπο, θα προσφέρεται μεγαλύτερη διαφήμιση τόσο στην εφαρμογή, όσο και στην Κύπρο γενικότερα.

Μια επιπλέον λειτουργία που θα μπορούσε να θεωρηθεί ως μέρος μελλοντικής εργασίας είναι η υποστήριξη και άλλων γλωσσών στην εφαρμογή, πέραν της αγγλικής. Η χρήση των Fragments στην υλοποίηση της εφαρμογής κάνει τα πράγματα ευκολότερα αφού δεν θα χρειαστεί η επανακατασκευή τους, αλλά κάποιες μικρές προσθήκες που θα μεταφράζουν το κείμενο τους σε κάποια άλλη γλώσσα.

Ακόμα ένα πλεονέκτημα της χρήσης των Fragments, είναι η προσαρμοστικότητα τους στα διάφορα μεγέθη κινητών συσκευών. Έτσι θα είναι εύκολο ως μελλοντική εργασία να δίνεται στην εφαρμογή η δυνατότητα να παρουσιάζεται σωστά σε όλους τους τύπους των κινητών συσκευών, είτε αυτές είναι tablets είτε «έξυπνα» κινητά τηλέφωνα.

Εισηγήσεις για μελλοντικές εργασίες προκύπτουν και από το ερωτηματολόγιο για την ανάλυση των απαιτήσεων. Σύμφωνα με τις απαντήσεις του δείγματος που λήφθηκε, αρκετοί χρήστες εισηγήθηκαν τη δημιουργία κοινωνικού δικτύου της εφαρμογής, μέσω του οποίου οι χρήστες θα κοινοποιούν σε άλλους χρήστες τις εμπειρίες και τα μέρη που επισκέφθηκαν. Αυτό θα βοηθούσε στην καλύτερη ενημέρωση των χρηστών για τις διάφορες υπηρεσίες που παρέχονται σε τουρίστες στο νησί, αλλά και θα συνέτεινε στη βελτίωση των υπηρεσιών αυτών, αφού οι παροχείς θα προσπαθούσαν να προσφέρουν την καλύτερη εξυπηρέτηση προς τους τουρίστες για να αποκτήσουν καλή φήμη μεταξύ των χρηστών.

Μια άλλη λειτουργία που προέκυψε από τις εισηγήσεις των χρηστών στο ερωτηματολόγιο ήταν η ενσωμάτωση φωνητικών οδηγιών στις οδηγίες πρόσβασης προς ένα συγκεκριμένο σημείο ενδιαφέροντος.

Επιπλέον, μια κατά την άποψη μου πολύ χρήσιμη λειτουργία για την εξέλιξη της εφαρμογής θα ήταν η δημιουργία ημερήσιου ταξιδιού, με την επιλογή συγκεκριμένων σημείων ενδιαφέροντος, και τη δημιουργία του συντομότερου μονοπατιού μεταξύ της τοποθεσίας του χρήστη σε συνδυασμό με οδηγίες πρόσβασης.

Μια τέτοια εφαρμογή με τόσες προοπτικές δεν θα μπορούσε να απευθύνεται μόνο σε κατόχους συσκευών που υποστηρίζουν το λειτουργικό σύστημα Android. Εξίσου χρήσιμη μελλοντική εργασία θα ήταν και η επέκταση της εφαρμογής έτσι ώστε να είναι διαθέσιμη και στο επόμενο δημοφιλές λειτουργικό σύστημα, το iOS της Apple.

Τέλος, ο εμπλουτισμός του σχεδιασμού της διεπαφής θα ήταν μια καλή επιπρόσθετη εργασία αφού τα εργαλεία διεπαφής εξελίσσονται συνεχώς και μπορεί να προσφέρουν περαιτέρω δυνατότητες στην λειτουργικότητα της εφαρμογής.

5.3 Εμπειρίες που Αποκτήθηκαν

Η εκπόνηση αυτής της Ατομικής Διπλωματικής Εργασίας με έχει γεμίσει ενθουσιασμό και αυτοπεποίθηση, καθώς η εμπειρίες που αποκόμισα ήταν πολλές και χρήσιμες και

πιστεύω ότι θα αποτελέσουν ορόσημο για το μέλλον μου. Η ιδέα ενασχόλησης μου με τον τομέα ανάπτυξης εφαρμογών για κινητές συσκευές Android πάντα με ενθουσίαζε, όμως μέχρι πριν από την ανάληψη της συγκεκριμένης εργασίας δεν είχα την ευκαιρία να ασχοληθώ με αυτόν.

Μέσω της εργασίας αυτής, που δόθηκε η ευκαιρία να εμβαθύνω κατά πολύ τις γνώσεις μου τόσο σε θέματα προγραμματισμού, όσο και στη χρήση και τις δυνατότητες της γλώσσας προγραμματισμού Java. Ανακάλυψα τις τεράστιες δυνατότητες που παρέχονται μέσω αυτής, αλλά και μέσω των βιβλιοθηκών που προσφέρει η Google για την ανάπτυξη των Android εφαρμογών, πράγμα που με γέμισε με ενθουσιασμό και κίνητρα. Η ευκολία ανάπτυξης εφαρμογών στο περιβάλλον ανάπτυξης εφαρμογών Eclipse και η εξοικείωσή μου με αυτό, ήταν ένα επιπλέον εργαλείο που με βοήθησε στην επιτυχή ολοκλήρωση της εργασίας.

Μέσα από την τριβή μου με αυτό το θέμα και την αρχική μελέτη του λειτουργικού συστήματος Android, ανέπτυξα μια καλή εικόνα τη δομή του αλλά και των κύριων συστατικών του, γεγονός που με βοήθησε στην ευκολότερη υλοποίηση της εφαρμογής. Επιπρόσθετα, μου δόθηκε η ευκαιρία να αντιληφθώ την πλειάδα δυνατοτήτων που προσφέρει το λειτουργικό σύστημα Android στις εφαρμογές που αναπτύσσονται για αυτό, πράγμα που ίσως αποτελέσει κίνητρο για να αναπτύξω και άλλου είδους εφαρμογές στο μέλλον.

Μπορεί να ήρθα αντιμέτωπος με προβληματικές καταστάσεις πρωτόγνωρες για μένα, όμως μέσω της υποστήριξης που προσφέρουν οι βοηθητικές βιβλιοθήκες, αλλά και ολόκληρες κοινότητες από προγραμματιστές εφαρμογών Android, ήμουν σε θέση να επιλύσω τα πολλά προβλήματα που παρουσιάστηκαν.

Πλέον, είμαι σε θέση να φέρω εις πέρας μια εργασία μεγάλης έκτασης χωρίς την προσωπική βοήθεια άλλων, αλλά λαμβάνοντας μόνο υπόψη τις συμβουλές και τις εισηγήσεις τους.

Μέσα από την εκπόνηση αυτής της αντιλήφθηκα το πόσο ραγδαία εξελίσσεται η τεχνολογία, ειδικά όσον αφορά τα «έξυπνα» κινητά τηλέφωνα και έχω εμπεδώσει ότι

δεν πρέπει ποτέ να αρκούμαστε στη στείρα γνώση αλλά να την ανανεώνουμε και να προσαρμόζομαστε στις ανάγκες της εποχής. Κατα τη διάρκεια υλοποίησης της εφαρμογής οι βοηθητικές βιβλιοθήκες και ο τρόπος ανάπτυξης των εφαρμογών στο λειτουργικό σύστημα Android επιδέχονταν αλλαγές, μέσω ανανεώσεων του Android SDK της Google, για να τυγχάνουν υποστήριξης από όσο το δυνατό περισσότερες εκδόσεις του λειτουργικού, αλλά και τη χρήση νεότερων τεχνολογιών κατά την υλοποίηση. Αυτό επέβαλλε την προσαρμογή της εφαρμογής στα νέα δεδομένα και

Πιστεύω ακράδαντα ότι όλα όσα αποκόμισα από αυτή την εμπειρία θα αποτελέσουν εργαλεία που θα με βοηθήσουν τόσο στην μελλοντική μου ακαδημαϊκή πορεία, όσο και στη σταδιοδρομία που θα ακολουθήσω.

Βιβλιογραφία

- [1] PC Magazine. (2013, May) Smartphone Definition from PC Magazine Encyclopedia. [Online].
<http://www.pcmag.com/encyclopedia/term/51537/smartphone>
- [2] Wikipedia. (2013, May) Android (operating system). [Online].
[https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))
- [3] Wikipedia. (2013, May) Dalvik (software). [Online].
[http://en.wikipedia.org/wiki/Dalvik_\(software\)](http://en.wikipedia.org/wiki/Dalvik_(software))
- [4] Wikipedia. (2013, May) Android Version History. [Online].
http://en.wikipedia.org/wiki/Android_version_history
- [5] VentureBeat. (2013, January) Android captured almost 70% global smartphone market share in 2012, Apple just under 20% Read more at <http://venturebeat.com/2013/01/28/android-captured-almost-70-global-smartphone-market-share-in-2012-apple-just-under-20/#b4iPCEz1R3mIEZ3R.99>. [Online].
<http://venturebeat.com/2013/01/28/android-captured-almost-70-global-smartphone-market-share-in-2012-apple-just-under-20/>
- [6] Android Developers. (2013, May) Activities. [Online].
<http://developer.android.com/guide/components/activities.html>
- [7] Android Developers. (2013, May) Fragments. [Online].
<http://developer.android.com/guide/components/fragments.html>
- [8] Android Developers. (2013, May) The AndroidManifest.xml File. [Online].
<http://developer.android.com/guide/topics/manifest/manifest-intro.html>
- [9] Android Developers. (2013, May) Storage Options. [Online].
<http://developer.android.com/guide/topics/data/data-storage.html#db>

Παράρτημα Α

Ακολουθεί το ερωτηματολόγιο που δόθηκε για την έγκυρη εξαγωγή συμπερασμάτων σχετικά με τις απαιτήσεις των χρηστών, καθώς και τα συνοπτικά αποτελέσματα.

Ερωτηματολόγιο Δημιουργίας Τουριστικής Εφαρμογής της Κύπρου σε Android

Στα πλαίσια της Ατομικής Διπλωματικής Εργασίας μου, μου ανατέθηκε η δημιουργία μιας Android εφαρμογής η οποία θα εξυπηρετεί το τουριστικό κοινό της Κύπρου. Θα ήθελα τη βοήθεια σας σχετικά με το τι λειτουργίες πιστεύετε ότι θα πρέπει να συμπεριληφθούν στην εφαρμογή, απαντώντας τις πιο κάτω ερωτήσεις.

***Required**

Με βάση μια κλίμακα από το 1 ως το 5, πόσο σημαντικές είναι για εσάς οι χρεώσεις περιαγωγής από τη χρήση δεδομένων; *

Η ερώτηση αναφέρεται στις χρεώσεις περιαγωγής για σύνδεση στο διαδίκτυο από ένα κυπριακό τηλεπικοινωνιακό παροχέα, οι οποίες καθορίζονται από την ΕΕ και μειώνονται συνεχώς.

1 2 3 4 5

Καθόλου ☐ ☐ ☐ ☐ ☐ Πάρα πολύ

Με βάση μια κλίμακα από το 1 ως το 5, πόσο σημαντική είναι η παρουσίαση όλων των κατηγοριών ενδιαφέροντος σε μια λίστα; *

Η λίστα θα λειτουργεί ως "οδηγός", προτείνοντας σημεία χωρισμένα ανά κατηγορία, όπως "Αξιοθέατα", "Εμπορικά Κέντρα" κτλ

1 2 3 4 5

Καθόλου ☐ ☐ ☐ ☐ ☐ Πάρα πολύ

Με βάση μια κλίμακα από το 1 ως το 5, πόσο σημαντική είναι η συμπερίληψη όλων των σημείων ενδιαφέροντος σε ένα χάρτη της Κύπρου, χωρισμένων ανά κατηγορία; *

1 2 3 4 5

Καθόλου ☐ ☐ ☐ ☐ ☐ Πάρα πολύ

Με βάση μια κλίμακα από το 1 ως το 5, πόσο σημαντική είναι η συμπερίληψη όλων των σημείων ενδιαφέροντος εντός μιας ακτίνας από την περιοχή της Κύπρου στην οποία βρίσκεστε; *

1 2 3 4 5

Καθόλου ☐ ☐ ☐ ☐ ☐ Πάρα πολύ

Με βάση μια κλίμακα από το 1 ως το 5, πόσο σημαντική είναι η συμπερίληψη πληροφοριών για πολιτιστικά μνημεία; *

1 2 3 4 5

Καθόλου ☐ ☐ ☐ ☐ ☐ Πάρα πολύ

Με βάση μια κλίμακα από το 1 ως το 5, πόσο σημαντική είναι η συμπερίληψη χώρων διαμονής στην εφαρμογή; *

Η ερώτηση αφορά κυρίως ξενοδοχεία.

1 2 3 4 5

Καθόλου ☐ ☐ ☐ ☐ ☐ Πάρα πολύ

Με βάση μια κλίμακα από το 1 ως το 5, πόσο σημαντική είναι η συμπερίληψη πληροφοριών για υπηρεσίες έκτακτης ανάγκης; *

1 2 3 4 5

Καθόλου ☐ ☐ ☐ ☐ ☐ Πάρα πολύ

Με βάση μια κλίμακα από το 1 ως το 5, πόσο σημαντική είναι η συμπερίληψη πληροφοριών για εστιατόρια; *

1 2 3 4 5

Καθόλου ☐ ☐ ☐ ☐ ☐ Πάρα πολύ

Με βάση μια κλίμακα από το 1 ως το 5, πόσο σημαντική είναι η συμπερίληψη πληροφοριών για εμπορικά κέντρα; *

1 2 3 4 5

Καθόλου ☐ ☐ ☐ ☐ ☐ Πάρα πολύ

Με βάση μια κλίμακα από το 1 ως το 5, πόσο σημαντική είναι η συμπερίληψη πληροφοριών για Μέσα Μαζικής Μεταφοράς; *

1 2 3 4 5

Καθόλου ☐ ☐ ☐ ☐ ☐ Πάρα πολύ

Με βάση μια κλίμακα από το 1 ως το 5, πόσο σημαντική είναι η συμπερίληψη πληροφοριών για TAXI; *

1 2 3 4 5

Καθόλου ☐ ☐ ☐ ☐ ☐ Πάρα πολύ

Με βάση μια κλίμακα από το 1 ως το 5, πόσο σημαντική είναι η δημιουργία κοινωνικού δικτύου της εφαρμογής; *

Το κοινωνικό δίκτυο της εφαρμογής θα έχει ως μέλη τους χρήστες της εφαρμογής οι οποίοι θα μπορούν να μοιράζονται τις εμπειρίες τους με άλλους χρήστες της εφαρμογής

1 2 3 4 5

Καθόλου ☐ ☐ ☐ ☐ ☐ Πάρα πολύ

Ποιά άλλη λειτουργία θα θεωρούσατε χρήσιμη σε μια εφαρμογή αυτού του είδους;

Never submit passwords through Google Forms.

Powered by



This content is neither created nor endorsed by Google.

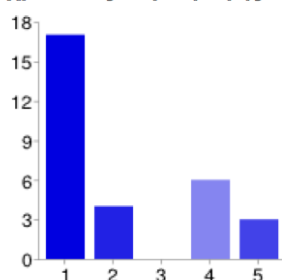
[Report Abuse](#) - [Terms of Service](#) - [Additional Terms](#)

30 responses

[View all responses](#) [Publish analytics](#)

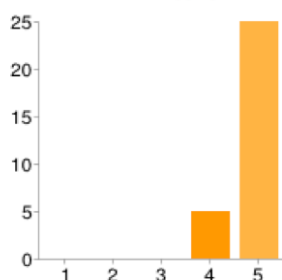
Summary

Με βάση μια κλίμακα από το 1 ως το 5, πόσο σημαντικές είναι για εσάς οι χρεώσεις περιαγωγής από τη χρήση δεδομένων;



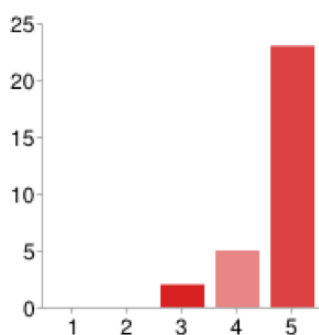
1	17	57%
2	4	13%
3	0	0%
4	6	20%
5	3	10%

Με βάση μια κλίμακα από το 1 ως το 5, πόσο σημαντική είναι η παρουσίαση όλων των κατηγοριών ενδιαφέροντος σε μια λίστα;



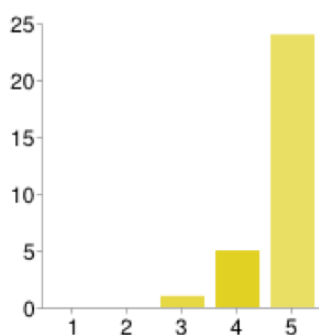
1	0	0%
2	0	0%
3	0	0%
4	5	17%
5	25	83%

Με βάση μια κλίμακα από το 1 ως το 5, πόσο σημαντική είναι η συμπερίληψη όλων των σημείων ενδιαφέροντος σε ένα χάρτη της Κύπρου, χωρισμένων ανά κατηγορία;



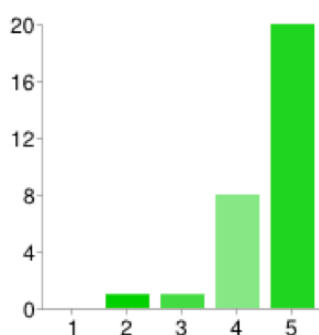
1	0	0%
2	0	0%
3	2	7%
4	5	17%
5	23	77%

Με βάση μια κλίμακα από το 1 ως το 5, πόσο σημαντική είναι η συμπερίληψη όλων των σημείων ενδιαφέροντος εντός μιας ακτίνας από την περιοχή της Κύπρου στην οποία βρίσκεστε;



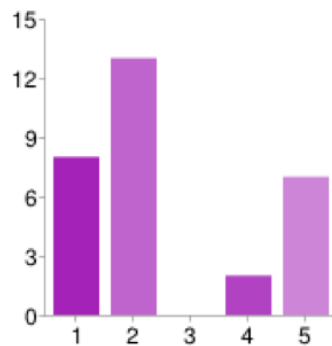
1	0	0%
2	0	0%
3	1	3%
4	5	17%
5	24	80%

Με βάση μια κλίμακα από το 1 ως το 5, πόσο σημαντική είναι η συμπερίληψη πληροφοριών για πολιτιστικά μνημεία;



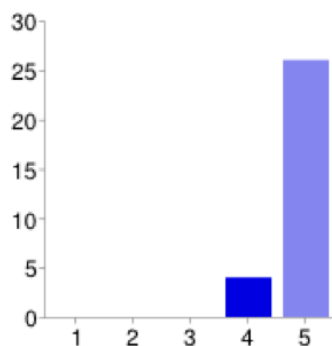
1	0	0%
2	1	3%
3	1	3%
4	8	27%
5	20	67%

Με βάση μια κλίμακα από το 1 ως το 5, πόσο σημαντική είναι η συμπερίληψη χώρων διαμονής στην εφαρμογή;



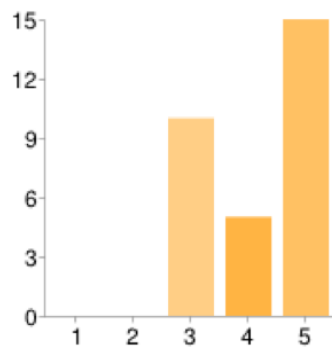
1	8	27%
2	13	43%
3	0	0%
4	2	7%
5	7	23%

Με βάση μια κλίμακα από το 1 ως το 5, πόσο σημαντική είναι η συμπερίληψη πληροφοριών για υπηρεσίες έκτακτης ανάγκης;



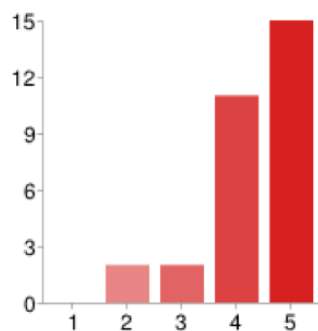
1	0	0%
2	0	0%
3	0	0%
4	4	13%
5	26	87%

Με βάση μια κλίμακα από το 1 ως το 5, πόσο σημαντική είναι η συμπερίληψη πληροφοριών για εστιατόρια;



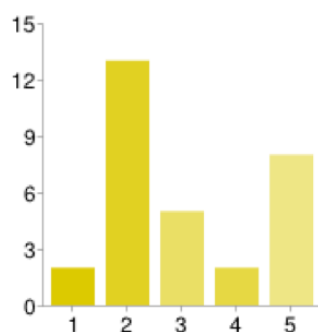
1	0	0%
2	0	0%
3	10	33%
4	5	17%
5	15	50%

Με βάση μια κλίμακα από το 1 ως το 5, πόσο σημαντική είναι η συμπερίληψη πληροφοριών για εμπορικά κέντρα;



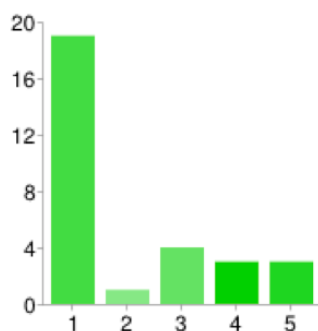
1	0	0%
2	2	7%
3	2	7%
4	11	37%
5	15	50%

Με βάση μια κλίμακα από το 1 ως το 5, πόσο σημαντική είναι η συμπερίληψη πληροφοριών για Μέσα Μαζικής Μεταφοράς;



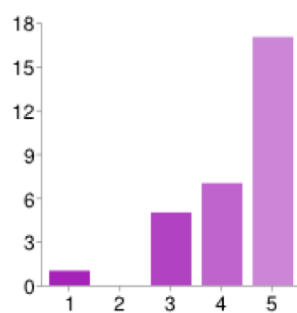
1	2	7%
2	13	43%
3	5	17%
4	2	7%
5	8	27%

Με βάση μια κλίμακα από το 1 ως το 5, πόσο σημαντική είναι η συμπερίληψη πληροφοριών για TAXI;



1	19	63%
2	1	3%
3	4	13%
4	3	10%
5	3	10%

Με βάση μια κλίμακα από το 1 ως το 5, πόσο σημαντική είναι η δημιουργία κοινωνικού δικτύου της εφαρμογής;



1	1	3%
2	0	0%
3	5	17%
4	7	23%
5	17	57%

Ποιά άλλη λειτουργία θα θεωρούσατε χρήσιμη σε μια εφαρμογή αυτού του είδους;

Φωνητικές οδηγίες για να καθοδηγούν τον χρήστη από το ένα σημείο στο άλλο.

Παράρτημα Β

Εδώ παρουσιάζονται οι υλοποιήσεις των κλάσεων που χρησιμοποιήθηκαν για τη δημιουργία της εφαρμογής.

Κλάση CityFragment.java

```
package com.thesis.lovecyprus;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

import android.app.Activity;
import android.app.ListFragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ListView;
import android.widget.SimpleAdapter;

public class CityFragment extends ListFragment {

    OnCityListener mCallback;

    String[] cities = {"Nicosia", "Larnaca", "Limassol", "Paphos", "Famagusta"};

    String[] info = new String[] { "The last divided capital in the world",
        "City of the Palm trees",
        "The second capital",
        "The birthplace of Aphrodite",
        "The Best blue - flagged beaches"};

    int[] icons = new int[] {
        R.drawable.nicosia_logo,
        R.drawable.larnaca_municipality,
        R.drawable.limassol_municipality,
        R.drawable.pafos_municipality,
        R.drawable.famagusta_logo};

    public interface OnCityListener {
        public void OnCitySelected(int position);
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
```

```

        Bundle savedInstanceState) {

            // Each row in the list stores country name, currency and flag
            List<HashMap<String,String>> GuideList = new
            ArrayList<HashMap<String,String>>();

            for(int i=0;i<5;i++){
                HashMap<String, String> hm = new HashMap<String,String>();
                hm.put("txt", cities[i]);
                hm.put("desc", info[i]);
                hm.put("icon", Integer.toString(icons[i]) );
                GuideList.add(hm);
            }

            // Keys used in Hashmap
            String[] from = { "txt","desc","icon" };
            // Ids of views in listview_layout
            int[] to = { R.id.txt,R.id.desc,R.id.icon};

            // Instantiating an adapter to store each items
            // R.layout.listview_layout defines the layout of each item
            SimpleAdapter adapter = new
            SimpleAdapter(this.getActivity().getBaseContext(),
            R.layout.listview_layout, from, to);

            setListAdapter(adapter);

            // TODO Auto-generated method stub
            return super.onCreateView(inflater, container, savedInstanceState);
        }

        @Override
        public void onItemClick(ListView l, View v, int position, long id) {
            // TODO Auto-generated method stub
            mCallback.OnCitySelected(position);
        }

        @Override
        public void onAttach(Activity activity) {
            super.onAttach(activity);

            // This makes sure that the container activity has implemented
            // the callback interface. If not, it throws an exception.
            try {
                mCallback = (OnCityListener) activity;
            } catch (ClassCastException e) {

```

```

        throw new ClassCastException(activity.toString()
            + " must implement OnCityListener");
    }
}
}

```

Κλάση CyprusMap.java

```

package com.thesis.lovecyprus;

import android.app.Activity;
import android.app.Dialog;
import android.app.Fragment;
import android.app.FragmentTransaction;
import android.content.Context;
import android.content.Intent;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Spinner;

import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.GooglePlayServicesUtil;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.GoogleMap.OnInfoWindowClickListener;
import com.google.android.gms.maps.MapFragment;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;

public class CyprusMap extends Fragment implements LocationListener,
    OnInfoWindowClickListener, AdapterView.OnItemClickListener {

    GoogleMap map;
    Spinner spinner;
    private double latitude = 0;
    private double longitude = 0;
}

```

```

public double getLatitude() {
    return latitude;
}

public void setLatitude(double latitude) {
    this.latitude = latitude;
}

public double getLongitude() {
    return longitude;
}

public void setLongitude(double longitude) {
    this.longitude = longitude;
}

public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {

    return inflater.inflate(R.layout.cyprusmap, container, false);
}

@Override
public void onActivityCreated(Bundle savedInstanceState) {

    super.onActivityCreated(savedInstanceState);

    // Getting Google Play availability status
    int status = GooglePlayServicesUtil.isGooglePlayServicesAvailable(this
        .getActivity().getBaseContext());

    // Showing status
    if (status != ConnectionResult.SUCCESS) { // Google Play Services are

        // not available

        int requestCode = 10;
        Dialog dialog = GooglePlayServicesUtil.getErrorDialog(status,
            this.getActivity(), requestCode);
        dialog.show();

    }

    else { // Google Play Services are available

        Spinner spinner = (Spinner)
this.getActivity().findViewById(R.id.categories_spinner);

```

```

        spinner.setOnItemSelectedListener(this);

        map = ((MapFragment) getFragmentManager()
                .findFragmentById(R.id.map)).getMap();
        // map.setMyLocationEnabled(true);
        map.setOnInfoWindowClickListener(this);

        map.moveCamera(CameraUpdateFactory.newLatLngZoom(
                MainActivity.CYPRUS, 8));

        Activity main = getActivity();
        LocationManager locationManager = (LocationManager) main

        .getSystemService(Context.LOCATION_SERVICE);

        // Creating a criteria object to retrieve provider
        Criteria criteria = new Criteria();

        // Getting the name of the best provider
        String provider = locationManager.getBestProvider(criteria,
true);

        // Getting Current Location
        Location location =
locationManager.getLastKnownLocation(provider);

        if (location != null) {
            onLocationChanged(location);
        }

        // Create an ArrayAdapter using the string array and a default
spinner layout
        ArrayAdapter<CharSequence> adapter =
ArrayAdapter.createFromResource(getActivity(),
                R.array.categories_array,
android.R.layout.simple_spinner_item);
        // Specify the layout to use when the list of choices appears

        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdow
n_item);

        // Apply the adapter to the spinner
        spinner.setAdapter(adapter);

    }

```

```

    }

    @Override
    public void onDestroyView() {
        super.onDestroyView();
        MapFragment f = (MapFragment)
getFragmentManager().findFragmentById(R.id.map);

        try {

            if (f != null) {
                FragmentTransaction ft =
this.getActivity().getFragmentManager().beginTransaction();
                Log.d("THE PROB IS", "NEARBYMAP EXCEPTION:
META TO BEGIN");
                ft.remove(f);
                Log.d("THE PROB IS", "NEARBYMAP EXCEPTION:
META TO REMOVE");

                ft.commit();
                Log.d("THE PROB IS", "NEARBYMAP EXCEPTION:
META TO COMMIT");
            }
        } catch (Exception e) {
            Log.d("cyprusmap EXCEPTION: ", e.getMessage());
        }

    }

    @Override
    public void onInfoWindowClick(Marker marker) {

        Intent intent = new Intent(this.getActivity(), InfoActivity.class);

        Place place = MainActivity.datasource.getPlaceFromMarker(marker);

        Bundle bundle = new Bundle();
        bundle.putLong("id", place.getId());
        bundle.putString("name", place.getName());
        bundle.putString("city", place.getCity());
        bundle.putString("type", place.getType());
        bundle.putDouble("lat", place.getLatitude());
        bundle.putDouble("long", place.getLongitude());
        bundle.putString("imageurl", place.getImageurl());
        bundle.putString("description", place.getDescription());
        bundle.putString("telno", place.getTelno());
        bundle.putDouble("currentlat", this.getLatitude());
        bundle.putDouble("currentlong", this.getLongitude());
    }

```

```

        intent.putExtras(bundle);
        startActivity(intent);
    }

    @Override
    public void onLocationChanged(Location location) {
        // TODO Auto-generated method stub
        // Getting latitude of the current location
        double latitude = location.getLatitude();
        this.setLatitude(latitude);

        // Getting longitude of the current location
        double longitude = location.getLongitude();
        this.setLongitude(longitude);
    }

    @Override
    public void onProviderDisabled(String arg0) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onProviderEnabled(String arg0) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onStatusChanged(String arg0, int arg1, Bundle arg2) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onItemSelected(AdapterView<?> parent, View view, int pos,
                                long id) {

        switch (pos){

        case 0:
        {
            map.clear();

            for (int i = 0; i < MainActivity.places.size(); i++) {

```

```

        if
(MainActivity.places.get(i).getType().equals("Nightlife")) {
            map.addMarker(new MarkerOptions()
                .position(
                    new
LatLng(MainActivity.places.get(i)
            .getLatitude(), MainActivity.places
            .get(i).getLongitude()))
                .icon(BitmapDescriptorFactory
                    .fromResource(R.drawable.nightlife))
                .title(MainActivity.places.get(i).getName())
                .snippet(MainActivity.places.get(i).getType()));
        }

        else if (MainActivity.places.get(i).getType()
            .equals("Must See")) {
            map.addMarker(new MarkerOptions()
                .position(
                    new
LatLng(MainActivity.places.get(i)
            .getLatitude(), MainActivity.places
            .get(i).getLongitude()))
                .icon(BitmapDescriptorFactory
                    .fromResource(R.drawable.camera))
                .title(MainActivity.places.get(i).getName())
                .snippet(MainActivity.places.get(i).getType()));
        }

        else if (MainActivity.places.get(i).getType()
            .equals("Restaurants & Bars")) {
            map.addMarker(new MarkerOptions()
                .position(
                    new
LatLng(MainActivity.places.get(i)
            .getLatitude(), MainActivity.places
            .get(i).getLongitude()))
                .icon(BitmapDescriptorFactory

```

```

        .fromResource(R.drawable.restaurant))

        .title(MainActivity.places.get(i).getName())

        .snippet(MainActivity.places.get(i).getType());
    }

    else if (MainActivity.places.get(i).getType()
            .equals("Emergency")) {
        map.addMarker(new MarkerOptions()
            .position(
                new
                LatLng(MainActivity.places.get(i)
                    .getLatitude(), MainActivity.places
                    .get(i).getLongitude()))
            .icon(BitmapDescriptorFactory
                .fromResource(R.drawable.emergency_icon))
            .title(MainActivity.places.get(i).getName())
            .snippet(MainActivity.places.get(i).getType());
        }

    else if (MainActivity.places.get(i).getType()
            .equals("Shopping")) {
        map.addMarker(new MarkerOptions()
            .position(
                new
                LatLng(MainActivity.places.get(i)
                    .getLatitude(), MainActivity.places
                    .get(i).getLongitude()))
            .icon(BitmapDescriptorFactory
                .fromResource(R.drawable.shopping))
            .title(MainActivity.places.get(i).getName())
            .snippet(MainActivity.places.get(i).getType());
        }

    else {

```

```

                                map.addMarker(new MarkerOptions()
                                                .position(
                                new
LatLng(MainActivity.places.get(i)

                                .getLatitude(), MainActivity.places

                                .get(i).getLongitude()))

                                .title(MainActivity.places.get(i).getName())

                                .snippet(MainActivity.places.get(i).getType()));
                                }

                                }
                                break;

                                case 1:
                                {
                                        map.clear();

                                        for (int i = 0; i < MainActivity.places.size(); i++)
                                        {

                                                if (MainActivity.places.get(i).getType()
                                                                .equals("Must See")) {
                                                        map.addMarker(new MarkerOptions()
                                                                .position(
                                                                new
                                                                LatLng(MainActivity.places.get(i)

                                                                .getLatitude(), MainActivity.places

                                                                .get(i).getLongitude()))

                                                                .icon(BitmapDescriptorFactory

                                                                .fromResource(R.drawable.camera))

                                                                .title(MainActivity.places.get(i).getName())

                                                                .snippet(MainActivity.places.get(i).getType()));
                                                                }

                                                                }

                                                                break;
                                }

```

```

        case 2:
        {
            map.clear();

            for (int i = 0; i < MainActivity.places.size(); i++){
                if (MainActivity.places.get(i).getType()
                    .equals("Restaurants & Bars")) {
                    map.addMarker(new MarkerOptions()
                        .position(
                            new
                                LatLng(MainActivity.places.get(i)
                                    .getLatitude(), MainActivity.places
                                        .get(i).getLongitude()))
                            .icon(BitmapDescriptorFactory
                                .fromResource(R.drawable.restaurant))
                            .title(MainActivity.places.get(i).getName())
                            .snippet(MainActivity.places.get(i).getType()));
                }
            }

            break;
        }

        case 3:
        {
            map.clear();

            for (int i = 0; i < MainActivity.places.size(); i++){
                if
                    (MainActivity.places.get(i).getType().equals("Nightlife")) {
                    map.addMarker(new MarkerOptions()
                        .position(
                            new
                                LatLng(MainActivity.places.get(i)
                                    .getLatitude(), MainActivity.places
                                        .get(i).getLongitude()))

```

```

        .icon(BitmapDescriptorFactory
            .fromResource(R.drawable.nightlife))
        .title(MainActivity.places.get(i).getName())
        .snippet(MainActivity.places.get(i).getType()));
    }

    }

    break;
}

case 4:
{
    map.clear();

    for (int i = 0; i < MainActivity.places.size(); i++){

        if (MainActivity.places.get(i).getType()
            .equals("Shopping")) {

            map.addMarker(new MarkerOptions()
                .position(
                    new
LatLng(MainActivity.places.get(i)
            .getLatitude(), MainActivity.places
            .get(i).getLongitude()))
                .icon(BitmapDescriptorFactory
            .fromResource(R.drawable.shopping))
            .title(MainActivity.places.get(i).getName())
            .snippet(MainActivity.places.get(i).getType()));
        }

        break;
    }

case 5:

```

```

        {
            map.clear();

            for (int i = 0; i < MainActivity.places.size(); i++){

                if (MainActivity.places.get(i).getType()
                    .equals("Emergency")) {
                    map.addMarker(new MarkerOptions()
                        .position(
                            new
                                LatLng(MainActivity.places.get(i)
                                    .getLatitude(), MainActivity.places
                                        .get(i).getLongitude()))
                        .icon(BitmapDescriptorFactory
                            .fromResource(R.drawable.emergency_icon))
                        .title(MainActivity.places.get(i).getName())
                        .snippet(MainActivity.places.get(i).getType()));
                }
            }
            break;
        }
    }

    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {
        // TODO Auto-generated method stub
    }
}

```

Κλάση DownloadImageTask.java

```

package com.thesis.lovecyprus;

```

```

import java.io.InputStream;

import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.AsyncTask;
import android.util.Log;
import android.widget.ImageView;

class DownloadImageTask extends AsyncTask<String, Void, Bitmap> {
    ImageView bmImage;

    public DownloadImageTask(ImageView bmImage) {
        this.bmImage = bmImage;
    }

    protected Bitmap doInBackground(String... urls) {
        String urldisplay = urls[0];
        Bitmap mIcon11 = null;
        try {
            InputStream in = new java.net.URL(urldisplay).openStream();
            mIcon11 = BitmapFactory.decodeStream(in);
        } catch (Exception e) {
            Log.e("Error", e.getMessage());
            e.printStackTrace();
        }
        return mIcon11;
    }

    protected void onPostExecute(Bitmap result) {
        bmImage.setImageBitmap(result);
    }
}

```

Κλάση EmergencyFragment.java

```

package com.thesis.lovecyprus;

import java.util.ArrayList;
import java.util.List;

import android.app.Activity;
import android.app.ListFragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ListView;

public class EmergencyFragment extends ListFragment {

```

```

OnEmergencyListener mCallback;

public interface OnEmergencyListener {
    public void OnEmergencySelected(int position);
}

public static List <Place> emergencyplaces = new ArrayList<Place>();
String[] emergencynames;

public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {

    emergencyplaces.clear();

    for (int i=0; i< MainActivity.places.size(); i++)
    {
        if
(MainActivity.places.get(i).getType().equals(GuideFragment.epilogues[MainActivity.Category])
        &&
MainActivity.places.get(i).getCity().equals(MainActivity.cities[MainActivity.CityNumber] ))
        {
            emergencyplaces.add(MainActivity.places.get(i));
        }
    }

    emergencynames = new String[emergencyplaces.size()];

    for (int i=0; i<emergencyplaces.size(); i++)
    {
        emergencynames[i] = emergencyplaces.get(i).getName();
    }

    ArrayAdapter<String> adapter = new
ArrayAdapter<String>(inflater.getContext(),
android.R.layout.simple_list_item_1,emergencynames);
    setListAdapter(adapter);

    return super.onCreateView(inflater, container, savedInstanceState);
}

```

```

public void onActivityCreated (Bundle savedInstanceState)
{
    super.onActivityCreated(savedInstanceState);

}

public void onItemClick(ListView l, View v, int position, long id) {

mCallback.OnEmergencySelected(position);
}

@Override
public void onAttach(Activity activity) {
    super.onAttach(activity);

    // This makes sure that the container activity has implemented
    // the callback interface. If not, it throws an exception.
    try {
        mCallback = (OnEmergencyListener) activity;
    } catch (ClassCastException e) {
        throw new ClassCastException(activity.toString()
            + " must implement OnEmergencyListener");
    }
}

}

```

Κλάση GuideFragment.java

```

package com.thesis.lovecyprus;

```

```

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

import android.app.Activity;
import android.app.ListFragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ListView;

```

```

import android.widget.SimpleAdapter;

public class GuideFragment extends ListFragment {

    OnCategorySelectedListener mCallback;

    public interface OnCategorySelectedListener {
        public void OnCategorySelected(int position);
    }

    public static String[] epilogues = new String[] { "Welcome to Cyprus", "Must
See", "Restaurants & Bars", "Nightlife",
        "Shopping", "Emergency"};

    String[] info = new String[] { "A few things about the island of Aphrodite",
        "Places worth visiting",
        "Looking for a nice dinner, or a drink?",
        "In a mood for clubbing?",
        "Main Shopping Centres along the island",
        "Police Stations, Hospitals etc."};

    int[] icons = new int[] {
        R.drawable.ic_launcher,
        R.drawable.camera,
        R.drawable.restaurant,
        R.drawable.nightlife,
        R.drawable.shopping,
        R.drawable.emergency_icon
    };

    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {

        // Each row in the list stores category, description and icon
        List<HashMap<String,String>> GuideList = new
        ArrayList<HashMap<String,String>>();

        for(int i=0;i<6;i++){
            HashMap<String, String> hm = new HashMap<String,String>();
            hm.put("txt", epilogues[i]);
            hm.put("desc", info[i]);
            hm.put("icon", Integer.toString(icons[i]) );
            GuideList.add(hm);
        }

        // Keys used in Hashmap
        String[] from = { "txt", "desc", "icon" };

```

```

// Ids of views in listview_layout
int[] to = { R.id.txt,R.id.descr,R.id.icon};

// Instantiating an adapter to store each items
// R.layout.listview_layout defines the layout of each item
SimpleAdapter adapter = new SimpleAdapter(this.getActivity().getBaseContext(),
GuideList, R.layout.listview_layout, from, to);

setListAdapter(adapter);

return super.onCreateView(inflater, container, savedInstanceState);

}

public void onItemClick(ListView l, View v, int position, long id) {

    mCallback.OnCategorySelected(position);

}

public void onActivityCreated (Bundle savedInstanceState)
{
    super.onActivityCreated(savedInstanceState);

}

@Override
public void onAttach(Activity activity) {
    super.onAttach(activity);

    // This makes sure that the container activity has implemented
    // the callback interface. If not, it throws an exception.
    try {
        mCallback = (OnCategorySelectedListener) activity;
    } catch (ClassCastException e) {
        throw new ClassCastException(activity.toString()
            + " must implement OnCategorySelectedListener");
    }
}

}

```

Κλάση InfoActivity.java

```
package com.thesis.lovecyprus;

import java.io.InputStream;
import java.net.URL;

import android.app.Activity;
import android.content.Intent;
import android.graphics.drawable.Drawable;
import android.net.Uri;
import android.os.Bundle;
import android.text.method.ScrollingMovementMethod;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;

public class InfoActivity extends Activity {

    double currentlat = 0;
    double currentlong=0;

    double latitude = 0;
    double longitude =0;

    String telno ="";

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.info);

        Bundle b = getIntent().getExtras();

        String name = b.getString("name", "null name");
        String city = b.getString("city", "null city");
        @SuppressWarnings("unused")
        String type = b.getString("type","null type");
        String description = b.getString("description", "null description");
        String imageurl = b.getString("imageurl", "null imageurl");
        telno = b.getString("telno","0");

        final double latitude = b.getDouble("lat");
```

```

final double longitude = b.getDouble("long");

final double currentlat = b.getDouble("currentlat");
final double currentlong = b.getDouble("currentlong");

TextView title = (TextView) findViewById(R.id.info_title);
title.setText(name);

TextView citytitle = (TextView) findViewById(R.id.info_city);
citytitle.setText(city);

TextView infostring = (TextView) findViewById(R.id.info_description);
infostring.setText(description);
infostring.setMovementMethod(new ScrollingMovementMethod());
//infostring.setText("APOEL");

ImageView image = (ImageView) findViewById(R.id.info_image);

new DownloadImageTask(image).execute(imageurl);

Button getDirections = (Button) this.findViewById(R.id.getdirectionsbutton);
getDirections.setOnClickListener(new OnClickListener() {
    public void onClick(View v)
    {
        Intent i = new Intent(android.content.Intent.ACTION_VIEW,
                               Uri.parse("http://maps.google.com/maps?saddr="
+ currentlat+", "+currentlong
                               + "&daddr="
+
latitude+", "+longitude));
        startActivity(i);
    }
});

Button callButton = (Button) this.findViewById(R.id.callbutton);

callButton.setOnClickListener(new OnClickListener() {
    public void onClick(View v)
    {
        Intent callIntent = new Intent(Intent.ACTION_CALL,
Uri.parse("tel:"+telno));
        startActivity(callIntent);
    }
});

}

```

```

        public static Drawable LoadImageFromWeb(String url) {
            try {
                InputStream is = (InputStream) new URL(url).getContent();

                Drawable d = Drawable.createFromStream(is, "src name");
                return d;
            } catch (Exception e) {
                System.out.println("Exception in image loading: " + e );
                return null;
            }
        }
    }
}

```

MustSeeFragment.java

```

package com.thesis.lovecyprus;

import java.util.ArrayList;
import java.util.List;

import android.app.Activity;
import android.app.ListFragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ListView;

public class MustSeeFragment extends ListFragment {

    OnMustSeeListener mCallback;

    public interface OnMustSeeListener {
        public void OnMustSeeSelected(int position);
    }

    public static List <Place> mustseepplaces = new ArrayList<Place>();
    String[] placesnames;

    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {

        mustseepplaces.clear();

        for (int i=0; i< MainActivity.places.size(); i++)

```

```

        {
            if
(MainActivity.places.get(i).getType().equals(GuideFragment.epilogues[MainActivity.Category])
            &&
MainActivity.places.get(i).getCity().equals(MainActivity.cities[MainActivity.CityNumber]
        ))
        {
            mustseeplaces.add(MainActivity.places.get(i));
        }
    }

    placesnames = new String[mustseeplaces.size()];

    for (int i=0; i<mustseeplaces.size(); i++)
    {
        placesnames[i] = mustseeplaces.get(i).getName();
    }

    ArrayAdapter<String> adapter = new
ArrayAdapter<String>(inflater.getContext(),
android.R.layout.simple_list_item_1,placesnames);
    setListAdapter(adapter);

    return super.onCreateView(inflater, container, savedInstanceState);
}

public void onActivityCreated (Bundle savedInstanceState)
{
    super.onActivityCreated(savedInstanceState);

}

public void onItemClick(ListView l, View v, int position, long id) {

mCallback.OnMustSeeSelected(position);
}

@Override
public void onAttach(Activity activity) {
    super.onAttach(activity);
}

```

```

        // This makes sure that the container activity has implemented
        // the callback interface. If not, it throws an exception.
        try {
            mCallback = (OnMustSeeListener) activity;
        } catch (ClassCastException e) {
            throw new ClassCastException(activity.toString()
                + " must implement OnMustSeeListener");
        }
    }
}

```

NearbyMap.java

```

package com.thesis.lovecyprus;

import android.app.Activity;
import android.app.Dialog;
import android.app.Fragment;
import android.content.Context;
import android.content.Intent;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.Adapter;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Spinner;

import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.GooglePlayServicesUtil;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.GoogleMap.OnInfoWindowClickListener;
import com.google.android.gms.maps.MapFragment;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.CircleOptions;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;

```

```

public class NearbyMap extends Fragment implements LocationListener,
OnInfoWindowClickListener,
AdapterView.OnItemClickListener{

    GoogleMap googleMap;
    Spinner snippet;
    private double latitude;
    private double longitude;

    public double getLatitude() {
        return latitude;
    }

    public void setLatitude(double latitude) {
        this.latitude = latitude;
    }

    public double getLongitude() {
        return longitude;
    }

    public void setLongitude(double longitude) {
        this.longitude = longitude;
    }

    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {

        return inflater.inflate(R.layout.nearbymap, container, false);
    }

    @Override
    public void onLocationChanged(Location location) {

        // Getting latitude of the current location
        double latitude = location.getLatitude();
        this.setLatitude(latitude);

        // Getting longitude of the current location
        double longitude = location.getLongitude();
        this.setLongitude(longitude);

        // Creating a LatLng object for the current location
        LatLng latLng = new LatLng(latitude, longitude);

        // Showing the current location in Google Map
        googleMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));
    }
}

```

```

        // Zoom in the Google Map
        googleMap.animateCamera(CameraUpdateFactory.zoomTo(15));

    }

    @Override
    public void onProviderDisabled(String provider) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onProviderEnabled(String provider) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onStatusChanged(String provider, int status, Bundle extras) {
        // TODO Auto-generated method stub
    }

    public void onActivityCreated(Bundle savedInstanceState)
    {
        super.onActivityCreated(savedInstanceState);

        // Getting Google Play availability status
        int status =
GooglePlayServicesUtil.isGooglePlayServicesAvailable(this.getActivity().getBaseContext());

        // Showing status
        if(status!=ConnectionResult.SUCCESS){ // Google Play Services
are not available

            int requestCode = 10;
            Dialog dialog = GooglePlayServicesUtil.getErrorDialog(status,
this.getActivity(), requestCode);
            dialog.show();

        }

        else
        { // Google Play Services are available

            Spinner spinner = (Spinner)
this.getActivity().findViewById(R.id.categories_spinner2);
            spinner.setOnItemSelectedListener(this);

```

```

of activity_main.xml
// Getting reference to the SupportMapFragment
MapFragment fm = (MapFragment)
getFragmentManager().findFragmentById(R.id.map2);

// Getting GoogleMap object from the fragment
googleMap = fm.getMap();

// Enabling MyLocation Layer of Google Map
googleMap.setMyLocationEnabled(true);

googleMap.setOnInfoWindowClickListener(this);

// Getting LocationManager object from System
Service LOCATION_SERVICE
Activity main = getActivity();
LocationManager locationManager =
(LocationManager) main.getSystemService(Context.LOCATION_SERVICE);

// Creating a criteria object to retrieve provider
Criteria criteria = new Criteria();

// Getting the name of the best provider
String provider = locationManager.getBestProvider(criteria,
true);

// Getting Current Location
Location location =
locationManager.getLastKnownLocation(provider);

if(location!=null){
    onLocationChanged(location);
}

locationManager.requestLocationUpdates(provider, 20000,
0, this);

CircleOptions circleOptions = new CircleOptions()
    .center(new
LatLng(locationManager.getLastKnownLocation(provider).getLatitude(),
locationManager.getLastKnownLocation(provider).getLongitude()))
    .radius(3000)// In meters
    //.fillColor(Color.TRANSPARENT);
    ;
// Get back the mutable Circle
googleMap.addCircle(circleOptions);

// Create an ArrayAdapter using the string array and a
default spinner layout

```

```

        ArrayAdapter<CharSequence> adapter =
ArrayAdapter.createFromResource(getActivity(),
        R.array.categories_array,
android.R.layout.simple_spinner_item);
        // Specify the layout to use when the list of choices
appears

        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown
n_item);

        // Apply the adapter to the spinner
spinner.setAdapter(adapter);

    }

}

@Override
public void onDestroyView() {
    super.onDestroyView();
    MapFragment f = (MapFragment)
getFragmentManager().findFragmentById(R.id.map2);

    try{

        if (f != null)
        {

this.getActivity().getFragmentManager().beginTransaction().remove(f).commit();
        //getFragmentManager().executePendingTransactions();
        }
        }
        catch (Exception e)
        {
            Log.d("NEARBYMAP EXCEPTION: ", e.getMessage());
        }
    }

}

@Override
public void onInfoWindowClick(Marker marker) {

    Intent intent = new Intent(this.getActivity(), InfoActivity.class);

    Place place = MainActivity.datasource.getPlaceFromMarker(marker);

    Bundle bundle = new Bundle();

```

```

        bundle.putLong("id", place.getId());
        bundle.putString("name",place.getName());
        bundle.putString("city",place.getCity());
        bundle.putString("type", place.getType());
        bundle.putDouble("lat", place.getLatitude());
        bundle.putDouble("long",place.getLongitude());
        bundle.putDouble("currentlat",this.getLatitude() );
        bundle.putDouble("currentlong", this.getLongitude());
        bundle.putString("imageurl", place.getImageurl());
        bundle.putString("description", place.getDescription());
        bundle.putString("telno", place.getTelno());

        intent.putExtras(bundle);
        startActivity(intent);

    }

    @Override
    public void onItemClick(AdapterView<?> parent, View view, int pos,
        long id) {

        switch (pos){

            case 0:
            {
                googleMap.clear();

                for (int i = 0; i < MainActivity.places.size(); i++) {
                    if
(MainActivity.places.get(i).getType().equals("Nightlife")) {
                        googleMap.addMarker(new
MarkerOptions()
                            .position(
                                new
LatLng(MainActivity.places.get(i)
                            .getLatitude(), MainActivity.places
                            .get(i).getLongitude()))
                            .icon(BitmapDescriptorFactory
                            .fromResource(R.drawable.nightlife))
                            .title(MainActivity.places.get(i).getName())
                            .snippet(MainActivity.places.get(i).getType()));
                    }
                }
            }
        }
    }

```

```

    }

    else if (MainActivity.places.get(i).getType()
            .equals("Must See")) {
        googleMap.addMarker(new
MarkerOptions()
            .position(
                new
LatLng(MainActivity.places.get(i)
            .getLatitude(), MainActivity.places
            .get(i).getLongitude()))
            .icon(BitmapDescriptorFactory
            .fromResource(R.drawable.camera))
            .title(MainActivity.places.get(i).getName())
            .snippet(MainActivity.places.get(i).getType()));
    }

    else if (MainActivity.places.get(i).getType()
            .equals("Restaurants & Bars")) {
        googleMap.addMarker(new
MarkerOptions()
            .position(
                new
LatLng(MainActivity.places.get(i)
            .getLatitude(), MainActivity.places
            .get(i).getLongitude()))
            .icon(BitmapDescriptorFactory
            .fromResource(R.drawable.restaurant))
            .title(MainActivity.places.get(i).getName())
            .snippet(MainActivity.places.get(i).getType()));
    }

    else if (MainActivity.places.get(i).getType()
            .equals("Emergency")) {
        googleMap.addMarker(new
MarkerOptions()
            .position(

```

```

new
LatLng(MainActivity.places.get(i)

    .getLatitude(), MainActivity.places

    .get(i).getLongitude()))

    .icon(BitmapDescriptorFactory

    .fromResource(R.drawable.emergency_icon))

    .title(MainActivity.places.get(i).getName())

    .snippet(MainActivity.places.get(i).getType());
    }

    else if (MainActivity.places.get(i).getType()

    .equals("Shopping")) {

        googleMap.addMarker(new

MarkerOptions()

            .position(

                new

LatLng(MainActivity.places.get(i)

    .getLatitude(), MainActivity.places

    .get(i).getLongitude()))

    .icon(BitmapDescriptorFactory

    .fromResource(R.drawable.shopping))

    .title(MainActivity.places.get(i).getName())

    .snippet(MainActivity.places.get(i).getType());
    }

    else {

        googleMap.addMarker(new

MarkerOptions()

            .position(

                new

LatLng(MainActivity.places.get(i)

    .getLatitude(), MainActivity.places

    .get(i).getLongitude()))

```

```

        .title(MainActivity.places.get(i).getName())
        .snippet(MainActivity.places.get(i).getType());
    }

    }
    break;

    case 1:
    {
        googleMap.clear();

        for (int i = 0; i < MainActivity.places.size(); i++)
        {

            if (MainActivity.places.get(i).getType()
                .equals("Must See")) {
                googleMap.addMarker(new
MarkerOptions()
                    .position(
                        new
LatLng(MainActivity.places.get(i)
                    .getLatitude(), MainActivity.places
                    .get(i).getLongitude()))
                .icon(BitmapDescriptorFactory
                    .fromResource(R.drawable.camera))
                .title(MainActivity.places.get(i).getName())
                .snippet(MainActivity.places.get(i).getType()));
            }

        }
        break;

    case 2:
    {
        googleMap.clear();

```

```

        for (int i = 0; i < MainActivity.places.size(); i++) {
            if (MainActivity.places.get(i).getType()
                .equals("Restaurants & Bars")) {
                googleMap.addMarker(new MarkerOptions()
                    .position(
                        new
LatLng(MainActivity.places.get(i)
                    .getLatitude(), MainActivity.places
                    .get(i).getLongitude()))
                    .icon(BitmapDescriptorFactory
                    .fromResource(R.drawable.restaurant))
                    .title(MainActivity.places.get(i).getName())
                    .snippet(MainActivity.places.get(i).getType()));
            }
        }

        break;
    }

    case 3:
    {
        googleMap.clear();

        for (int i = 0; i < MainActivity.places.size(); i++) {
            if
(MainActivity.places.get(i).getType().equals("Nightlife")) {
                googleMap.addMarker(new
MarkerOptions()
                    .position(
                        new
LatLng(MainActivity.places.get(i)
                    .getLatitude(), MainActivity.places
                    .get(i).getLongitude()))
                    .icon(BitmapDescriptorFactory
                    .fromResource(R.drawable.nightlife))

```

```

        .title(MainActivity.places.get(i).getName())
        .snippet(MainActivity.places.get(i).getType());
    }

    }

    break;
}

case 4:
{
    googleMap.clear();

    for (int i = 0; i < MainActivity.places.size(); i++){

        if (MainActivity.places.get(i).getType()
            .equals("Shopping")) {

            googleMap.addMarker(new
MarkerOptions()
                .position(
                    new
LatLng(MainActivity.places.get(i)
                .getLatitude(), MainActivity.places
                .get(i).getLongitude()))
                .icon(BitmapDescriptorFactory
                .fromResource(R.drawable.shopping))
                .title(MainActivity.places.get(i).getName())
                .snippet(MainActivity.places.get(i).getType()));
        }

    }

    break;
}

case 5:
{

```

```

        googleMap.clear();

        for (int i = 0; i < MainActivity.places.size(); i++){

            if (MainActivity.places.get(i).getType()
                .equals("Emergency")) {
                googleMap.addMarker(new
MarkerOptions()
                    .position(
                        new
LatLng(MainActivity.places.get(i)
                    .getLatitude(), MainActivity.places
                    .get(i).getLongitude()))
                    .icon(BitmapDescriptorFactory
                    .fromResource(R.drawable.emergency_icon))
                    .title(MainActivity.places.get(i).getName())
                    .snippet(MainActivity.places.get(i).getType()));
            }

            break;
        }
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {
        // TODO Auto-generated method stub
    }
}

```

Place.java

```

package com.thesis.lovecyprus;

```

```

public class Place {
    private long id;
    private String name;
    private String city;
    private String type;
    private double latitude;
    private double longitude;
    private String imageUrl;
    private String telno;


    public String getTelno() {
        return telno;
    }
    public void setTelno(String telno) {
        this.telno = telno;
    }
    public String getImageurl() {
        return imageUrl;
    }
    public void setImageurl(String imageUrl) {
        this.imageUrl = imageUrl;
    }


    private String description;


    public String getDescription() {
        return description;
    }
    public void setDescription(String description) {
        this.description = description;
    }
    public long getId() {
        return id;
    }
    public void setId(long id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getCity() {
        return city;
    }

```

```

    }
    public void setCity(String city) {
        this.city = city;
    }
    public String getType() {
        return type;
    }
    public void setType(String type) {
        this.type = type;
    }
    public double getLatitude() {
        return latitude;
    }
    public void setLatitude(double latitude) {
        this.latitude = latitude;
    }
    public double getLongitude() {
        return longitude;
    }
    public void setLongitude(double longitude) {
        this.longitude = longitude;
    }
}

```

```

// Will be used by the ArrayAdapter in the ListView
@Override
public String toString() {
    return name;
}

```

```

}

```

PlaceDataSource.java

```

package com.thesis.lovecyprus;

```

```

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

```

```

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;

```

```

import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteQueryBuilder;

import com.google.android.gms.maps.model.Marker;

public class PlaceDataSource {

    // Database fields
    private SQLiteDatabase database;
    private PlaceDBHelper dbHelper;
    private String[] allColumns = { PlaceDBHelper.COLUMN_ID,

PlaceDBHelper.COLUMN_NAME,

PlaceDBHelper.COLUMN_CITY,

PlaceDBHelper.COLUMN_TYPE,

PlaceDBHelper.COLUMN_LATITUDE,

PlaceDBHelper.COLUMN_LONGITUDE,

PlaceDBHelper.COLUMN_IMAGE,

PlaceDBHelper.COLUMN_DESCRIPTION,

PlaceDBHelper.COLUMN_TELNO };

    private String[] VirtualColumns = { PlaceDBHelper.COLUMN_ID,

PlaceDBHelper.COLUMN_NAME,

PlaceDBHelper.COLUMN_CITY };

    public PlaceDataSource(Context context)
    {
        dbHelper = new PlaceDBHelper(context);
    }

    public void open() throws SQLException
    {
        database = dbHelper.getWritableDatabase();
    }

```

```

        public void close()
        {
            dbHelper.close();
        }

        public Place createPlace(String name, String city, String type, double
latitude, double longitude,
                                String imageUrl, String description, String telno)
        {
            ContentValues values = new ContentValues();
            values.put(PlaceDBHelper.COLUMN_NAME, name);
            values.put(PlaceDBHelper.COLUMN_CITY, city);
            values.put(PlaceDBHelper.COLUMN_TYPE, type);
            values.put(PlaceDBHelper.COLUMN_LATITUDE, latitude);
            values.put(PlaceDBHelper.COLUMN_LONGITUDE,
longitude);
            values.put(PlaceDBHelper.COLUMN_IMAGE, imageUrl);
            values.put(PlaceDBHelper.COLUMN_DESCRIPTION,
description);
            values.put(PlaceDBHelper.COLUMN_TELNO, telno);

            long insertId =
database.insert(PlaceDBHelper.TABLE_PLACES, null, values);

            Cursor cursor =
database.query(PlaceDBHelper.TABLE_PLACES,
                allColumns, PlaceDBHelper.COLUMN_ID + " =
" + insertId, null,
                null, null, null);

            cursor.moveToFirst();

            Place newPlace = cursorToPlace(cursor);

            cursor.close();
            return newPlace;
        }

        public void deletePlace(Place place)
        {
            long id = place.getId();
            database.delete(PlaceDBHelper.TABLE_PLACES,
PlaceDBHelper.COLUMN_ID + " = " + id , null);
        }

```

```

public List<Place> getAllPlaces()
{
    List<Place> places = new ArrayList<Place>();
    Cursor cursor =
database.query(PlaceDBHelper.TABLE_PLACES, allColumns, null, null, null, null,
null);
    cursor.moveToFirst();

    while (!cursor.isAfterLast())
    {
        Place place = cursorToPlace(cursor);
        places.add(place);
        cursor.moveToNext();
    }

    cursor.close();

    return places;
}

```

```

public Place getPlaceFromMarker(Marker marker)
{
    Cursor cursor = database.query
        (PlaceDBHelper.TABLE_PLACES, allColumns,
        PlaceDBHelper.COLUMN_NAME
+ " = '" + marker.getTitle() + "'",
        null, null, null, null);
    cursor.moveToFirst();

    Place place = cursorToPlace(cursor);

    cursor.close();

    return place;
}

```

```

public Place cursorToPlace(Cursor cursor)
{
    Place place = new Place();
    place.setId(cursor.getLong(0));
}

```

```

        place.setName(cursor.getString(1));
        place.setCity(cursor.getString(2));
        place.setType(cursor.getString(3));
        place.setLatitude(cursor.getDouble(4));
        place.setLongitude(cursor.getDouble(5));
        place.setImageurl(cursor.getString(6));
        place.setDescription(cursor.getString(7));
        place.setTelno(cursor.getString(8));
        return place;
    }

    public void loadVirtual() {
        new Thread(new Runnable() {
            public void run() {
                try {
                    fillVirtualTable();
                } catch (IOException e) {
                    throw new RuntimeException(e);
                }
            }
        }).start();
    }

    public void fillVirtualTable() throws IOException {
        Cursor cursor =
        database.query(PlaceDBHelper.TABLE_PLACES, VirtualColumns, null, null, null,
        null, null);
        cursor.moveToFirst();

        while (!cursor.isAfterLast()){
            ContentValues values = new ContentValues();
            values.put(PlaceDBHelper.COLUMN_ID, cursor.getString(0));
            values.put(PlaceDBHelper.COLUMN_NAME,
            cursor.getString(1));
            values.put(PlaceDBHelper.COLUMN_CITY,
            cursor.getString(2));

            database.insert(PlaceDBHelper.FTS_SEARCH_TABLE, null,
            values);

            cursor.moveToNext();
        }

        cursor.close();
    }
}

```

```

        public Cursor getWordMatches(String query, String[] columns) {
            String selection = PlaceDBHelper.COLUMN_NAME + " MATCH ?";
            String[] selectionArgs = new String[] {query+"*"};

            return query(selection, selectionArgs, columns);
        }

        private Cursor query(String selection, String[] selectionArgs, String[]
columns) {
            SQLiteQueryBuilder builder = new SQLiteQueryBuilder();
            builder.setTables(PlaceDBHelper.FTS_SEARCH_TABLE);
            Cursor cursor = builder.query(dbHelper.getReadableDatabase(),
                columns, selection, selectionArgs, null, null, null);

            if (cursor == null) {
                return null;
            } else if (!cursor.moveToFirst()) {
                cursor.close();
                return null;
            }
            return cursor;
        }
    }
}

```

PlaceDBHelper.java

```

package com.thesis.lovecyprus;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

public class PlaceDBHelper extends SQLiteOpenHelper{

    //DATABASE NAME AND VERSION
    private static final int DATABASE_VERSION = 52;

    private static final String DATABASE_NAME = "LoveCyprus.db";

```

```

//TABLE PLACES
public static final String TABLE_PLACES = "places";

//COLUMNS
public static final String COLUMN_ID = "_id";
public static final String COLUMN_NAME = "name";
public static final String COLUMN_CITY = "city";
public static final String COLUMN_TYPE = "type";
public static final String COLUMN_LATITUDE = "latitude";
public static final String COLUMN_LONGITUDE = "longitude";
public static final String COLUMN_IMAGE = "image";
public static final String COLUMN_DESCRIPTION = "description";
public static final String COLUMN_TELNO = "telno";

//VIRTUAL TABLE SEARCH_TABLE
public static final String FTS_SEARCH_TABLE = "searchtable";

private static final String FTS_TABLE_CREATE =
"CREATE VIRTUAL TABLE " + FTS_SEARCH_TABLE +
" USING fts3 (" +
COLUMN_ID + " integer primary key autoincrement, " +
COLUMN_NAME + " text not null unique, " +
COLUMN_CITY + " text not null " + ")";

//DATABASE CREATION STATEMENT
private static final String PLACES_CREATE = "create table " +
TABLE_PLACES +
"(" + COLUMN_ID + " integer primary key autoincrement, " +
COLUMN_NAME + " text not null unique, " +
COLUMN_CITY + " text not null, " +
COLUMN_TYPE + " text not null, " +
COLUMN_LATITUDE + " double not null, " +
COLUMN_LONGITUDE + " double not null, " +
COLUMN_IMAGE + " text, " +
COLUMN_DESCRIPTION + " text, " +
COLUMN_TELNO + " text " + ")";

public PlaceDBHelper(Context context) {
    super(context, DATABASE_NAME, null, DATABASE_VERSION);
}

```

```

@Override
public void onCreate(SQLiteDatabase db) {
    // TODO Auto-generated method stub
    db.execSQL(PLACES_CREATE);
    db.execSQL(FTS_TABLE_CREATE);

}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    // TODO Auto-generated method stub
    Log.w(PleaseDBHelper.class.getName(),
        "Upgrading database from version " + oldVersion + " to "
        + newVersion + ", which will destroy all
old data");
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_PLACES );
    db.execSQL("DROP TABLE IF EXISTS " + FTS_SEARCH_TABLE);
    onCreate(db);

}

public static int getDatabaseVersion() {
    return DATABASE_VERSION;
}

}

```

RestaurantsFragment.java

```

package com.thesis.lovecyprus;

import java.util.ArrayList;
import java.util.List;

import android.app.Activity;
import android.app.ListFragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ListView;

public class RestaurantsFragment extends ListFragment {

```

OnRestaurantListener mCallback;

```
public interface OnRestaurantListener {
    public void OnRestaurantSelected(int position);
}

public static List <Place> restaurants = new ArrayList<Place>();
String[] restaurantnames;

public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {

    restaurants.clear();

    for (int i=0; i< MainActivity.places.size(); i++)
    {
        if
(MainActivity.places.get(i).getType().equals(GuideFragment.epilogues[MainActivity.Category])
        &&
MainActivity.places.get(i).getCity().equals(MainActivity.cities[MainActivity.CityNumber] ))
        {
            restaurants.add(MainActivity.places.get(i));
        }
    }

    restaurantnames = new String[restaurants.size()];

    for (int i=0; i<restaurants.size(); i++)
    {
        restaurantnames[i] = restaurants.get(i).getName();
    }

    ArrayAdapter<String> adapter = new
ArrayAdapter<String>(inflater.getContext(),
android.R.layout.simple_list_item_1,restaurantnames);
    setListAdapter(adapter);

    return super.onCreateView(inflater, container, savedInstanceState);
}

public void onActivityCreated (Bundle savedInstanceState)
```

```

        {
            super.onActivityCreated(savedInstanceState);
        }

        public void onItemClick(ListView l, View v, int position, long id) {

            mCallback.OnRestaurantSelected(position);
        }

        @Override
        public void onAttach(Activity activity) {
            super.onAttach(activity);

            // This makes sure that the container activity has implemented
            // the callback interface. If not, it throws an exception.
            try {
                mCallback = (OnRestaurantListener) activity;
            } catch (ClassCastException e) {
                throw new ClassCastException(activity.toString()
                    + " must implement OnRestaurantListener");
            }
        }
    }
}

```

ShoppingFragment.java

```

package com.thesis.lovecyprus;

import java.util.ArrayList;
import java.util.List;

import android.app.Activity;
import android.app.ListFragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ListView;

public class ShoppingFragment extends ListFragment {

    OnShoppingListener mCallback;

    public interface OnShoppingListener {

```

```

        public void OnShoppingSelected(int position);

    }

    public static List <Place> shops = new ArrayList<Place>();
    String[] shopnames;

    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {

        shops.clear();

        for (int i=0; i< MainActivity.places.size(); i++)
        {
            if
(MainActivity.places.get(i).getType().equals(GuideFragment.epilogues[MainActivity.Category])
&&
MainActivity.places.get(i).getCity().equals(MainActivity.cities[MainActivity.CityNumber] ))
        {
            shops.add(MainActivity.places.get(i));
        }
    }

    shopnames = new String[shops.size()];

    for (int i=0; i<shops.size(); i++)
    {

        shopnames[i] = shops.get(i).getName();
    }

    ArrayAdapter<String> adapter = new
ArrayAdapter<String>(inflater.getContext(),
android.R.layout.simple_list_item_1,shopnames);
    setListAdapter(adapter);

    return super.onCreateView(inflater, container, savedInstanceState);
}

    public void onActivityCreated (Bundle savedInstanceState)
    {
        super.onActivityCreated(savedInstanceState);
    }

```

```

    }

    public void onListItemClick(ListView l, View v, int position, long id) {

        mCallback.OnShoppingSelected(position);
    }

    @Override
    public void onAttach(Activity activity) {
        super.onAttach(activity);

        // This makes sure that the container activity has implemented
        // the callback interface. If not, it throws an exception.
        try {
            mCallback = (OnShoppingListener) activity;
        } catch (ClassCastException e) {
            throw new ClassCastException(activity.toString()
                + " must implement OnShoppingListener");
        }
    }

}

```

SplashActivity.java

```

package com.thesis.lovecyprus;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.Window;
import android.view.WindowManager;

public class SplashActivity extends Activity {

    private static String TAG = SplashActivity.class.getName();
    private static long SLEEP_TIME = 3; // Sleep for some time

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}

```

```
        this.requestWindowFeature(Window.FEATURE_NO_TITLE);    // Removes
title bar
```

```
        this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);    // Removes
notification bar
```

```
        setContentView(R.layout.splash);
```

```
    // Start timer and launch main activity
    IntentLauncher launcher = new IntentLauncher();
    launcher.start();
}
```

```
private class IntentLauncher extends Thread {
```

```
    @Override
    /**
     * Sleep for some time and than start new activity.
     */
```

```
        public void run() {
            try {
                // Sleeping
                Thread.sleep(SLEEP_TIME*1000);
            } catch (Exception e) {
                Log.e(TAG, e.getMessage());
            }
        }
```

```
    // Start main activity
    Intent intent = new Intent(SplashActivity.this, MainActivity.class);
    SplashActivity.this.startActivity(intent);
    SplashActivity.this.finish();
}
```

```
}
```

```
}
```

WelcometocyFragment.java

```
package com.thesis.lovecyprus;
```

```
import java.io.InputStream;
import java.net.URL;
```

```
import android.app.Fragment;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.text.method.ScrollingMovementMethod;
import android.view.LayoutInflater;
```

```

import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

public class WelcometocyFragment extends Fragment{
    final static String ARG_POSITION = "position";

    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {

        return inflater.inflate(R.layout.welcomefrag, container, false);
    }

    public void onActivityCreated (Bundle savedInstanceState)
    {
        super.onActivityCreated(savedInstanceState);
        TextView view = (TextView)
getActivity().findViewById(R.id.CyDescr);
        view.setText("Welcome back to the year-round island. " +
            "Even if you've been here before, there's always a new
world to discover. " +
            "Cyprus lies at the crossroads of three continents, where
East meets West and a " +
            "new experience awaits for you under the sun every day. "
+
            "Where Championship Golf courses, inviting beaches,
and breathtaking mountain trails lie around " +
            "luxurious hotels. Enjoy the island's beauty once more.");
        view.setMovementMethod(new ScrollingMovementMethod());

        String imageUrl =
"http://www.cs.ucy.ac.cy/~cfasou01/CyprusSatellite.jpg";

        ImageView image = (ImageView)
getActivity().findViewById(R.id.welcomeimg);

        new DownloadImageTask(image).execute(imageurl);
    }

    public static Drawable loadImageFromWeb(String url) {
        try {
            InputStream is = (InputStream) new URL(url).getContent();
            // Bitmap b = BitmapFactory.decodeStream(is);

```

```

        Drawable d = Drawable.createFromStream(is, "src name");
        return d;
    } catch (Exception e) {
        System.out.println("Exception in image loading: " + e );
        return null;
    }
}
}

```

MainActivity.java

```

package com.thesis.lovecyprus;

import java.util.ArrayList;
import java.util.List;

import android.app.ActionBar;
import android.app.ActionBar.Tab;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.Dialog;
import android.app.Fragment;
import android.app.FragmentManager;
import android.app.FragmentTransaction;
import android.app.SearchManager;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.net.Uri;
import android.os.Bundle;
import android.view.Menu;
import android.widget.ListView;
import android.widget.SearchView;
import android.widget.SimpleCursorAdapter;

import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.GooglePlayServicesUtil;
import com.google.android.gms.maps.model.LatLng;

public class MainActivity extends Activity implements ActionBar.TabListener,

```

```

        GuideFragment.OnCategorySelectedListener,
        MustSeeFragment.OnMustSeeListener,
        RestaurantsFragment.OnRestaurantListener,
        ShoppingFragment.OnShoppingListener, CityFragment.OnCityListener,
        NightlifeFragment.OnNightlifeListener,
        EmergencyFragment.OnEmergencyListener, LocationListener
        , SearchView.OnQueryTextListener {

    private double latitude;
    private double longitude;

    GuideFragment guidefrag = null;
    CyprusMap cyprusmap = null;
    NearbyMap nearbymap = null;

    public static int CityNumber = -1;
    public static int Category = -1;

    public static PlaceDataSource datasource;
    public static List<Place> places;

    List<String> results;

    private ListView mListView;

    public static final LatLng CYPRUS = new LatLng(35.1667d, 33.3500d);

    public static final String[] cities = {"Nicosia", "Larnaca", "Limassol", "Paphos",
    "Famagusta"};

    FragmentManager fm = getFragmentManager();

    ActionBar actionBar;

    public void onNewIntent(Intent intent) {
        setIntent(intent);
        handleIntent(intent);
    }

    private void handleIntent(Intent intent) {

        if (Intent.ACTION_SEARCH.equals(intent.getAction())) {
            String query = intent.getStringExtra(SearchManager.QUERY);

```

```

Cursor cursor = datasource.getWordMatches(query, null);

if (cursor == null)
{
    new AlertDialog.Builder(this)
        .setTitle("Search Results")
        .setIcon(R.drawable.warning_icon)
        .setMessage("No Results Found!")
        .setCancelable(false)
        .setPositiveButton("OK",null).show();

}

else
{
    int count = cursor.getCount();

    mListView.setTextFilterEnabled(true);

    // Specify the columns we want to display in the result
    String[] from = new String[] {
PlaceDBHelper.COLUMN_NAME,
        PlaceDBHelper.COLUMN_CITY };

    // Specify the corresponding layout elements where we want the
columns to go
    int[] to = new int[] { R.id.PlaceName,
        R.id.PlaceCity };

    // Create a simple cursor adapter for the definitions and apply
them to the ListView
    @SuppressWarnings("deprecation")
        SimpleCursorAdapter adapter = new
SimpleCursorAdapter(this,
        R.layout.result, cursor, from, to);

    mListView.setAdapter(adapter);

    cursor.moveToFirst();

    results = new ArrayList<String>();

    Bundle b = new Bundle();

    int i=0;

```

```

        int j=0;

        for (i =0; !cursor.isAfterLast(); cursor.moveToNext())

        {
            for ( j =0; j<cursor.getColumnCount();j++)
            {
                b.putString("string"+j, cursor.getString(j));
                System.out.println("koko"+j);
            }

        }

        cursor.close();

        b.putInt("i", i);
        b.putInt("j", j);

        // Intent intent2 = new Intent(this,SearchActivity.class);

        //intent2.putExtras(b);

        // startActivity(intent2);
        }

        }
        else if (Intent.ACTION_VIEW.equals(intent.getAction())) {
            // Handle a suggestions click (because the suggestions all use
ACTION_VIEW)
            //Uri data = intent.getData();
            //showResult(data);
        }
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        mListView = (ListView) findViewById(R.id.list);

        handleIntent(getIntent());

        // Set up the action bar.
        actionBar = getActionBar();

```

```

        // Specify that tabs should be displayed in the action bar.

        actionBar.setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);

        for (int i = 0; i < 4; i++) {
            if (i == 0) {
                actionBar.addTab(actionBar.newTab()

                .setText(R.string.title_section1).setTabListener(this));
            }

            if (i == 1) {
                actionBar.addTab(actionBar.newTab()

                .setText(R.string.title_section2).setTabListener(this));
            }

            if (i == 2) {
                actionBar.addTab(actionBar.newTab()

                .setText(R.string.title_section3).setTabListener(this));
            }

        }

        datasource = new PlaceDataSource(this);
        datasource.open();

        places = datasource.getAllPlaces();

        System.out.println("Table lines = "+ places.size());

        if (places.size() == 0)
        {
            insertPlacestoDatabase(datasource, places);
            datasource.loadVirtual();
        }

        // Getting Google Play availability status
        int status =
        GooglePlayServicesUtil.isGooglePlayServicesAvailable(getBaseContext());

        // Showing status

```

```

        if(status!=ConnectionResult.SUCCESS){ // Google Play Services are not available

            int requestCode = 10;
            Dialog dialog = GooglePlayServicesUtil.getErrorDialog(status, this,
requestCode);
            dialog.show();

        }else { // Google Play Services are available

            // Getting LocationManager object from System Service
LOCATION_SERVICE
            LocationManager locationManager = (LocationManager)
getSystemService(LOCATION_SERVICE);

            // Creating a criteria object to retrieve provider
Criteria criteria = new Criteria();

            // Getting the name of the best provider
String provider = locationManager.getBestProvider(criteria, true);

            // Getting Current Location
Location location = locationManager.getLastKnownLocation(provider);

            if(location!=null){
                onLocationChanged(location);
            }
        }
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);

        // Get the SearchView and set the searchable configuration
SearchManager searchManager = (SearchManager)
getSystemService(Context.SEARCH_SERVICE);
SearchView searchView = (SearchView)
menu.findItem(R.id.menu_search).getActionView();
        // Assumes current activity is the searchable activity

searchView.setSearchableInfo(searchManager.getSearchableInfo(getComponentName()
));
        // searchView.setIconifiedByDefault(false); // Do not iconify the
widget; expand it by default

searchView.setQueryRefinementEnabled(false);

```

```

        searchView.setSubmitButtonEnabled(false);

        return true;
    }

    @Override
    public void onTabReselected(Tab tab, FragmentTransaction ft) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onTabSelected(Tab tab, FragmentTransaction ft) {
        // TODO Auto-generated method stub

        // GUIDE TAB

        //empty back stack in case there is a fragment in it
        fm.popBackStack(null,
FragmentManager.POP_BACK_STACK_INCLUSIVE);
        System.out.println("BackStack emptied");

        if (tab.getText().equals("Guide")) {
            if (guidefrag == null) {

                guidefrag = new GuideFragment();
                fm.beginTransaction().replace(R.id.fragment_container,
guidefrag)
                    .commit();
            }

            else {
                Fragment guide = new GuideFragment();

                fm.beginTransaction().replace(R.id.fragment_container,guide).commit();

            }
        }

        // MAP TAB
        if (tab.getText().equals("Map")) {

            cyprusmap = new CyprusMap();

            fm.beginTransaction().replace(R.id.fragment_container,
cyprusmap).commit();

```

```

    }

    // NEARBY TAB
    if (tab.getText().equals("Nearby")) {

        nearbymap = new NearbyMap();

        fm.beginTransaction().replace(R.id.fragment_container, nearbymap).commit();

    }

}

@Override
public void onTabUnselected(Tab tab, FragmentTransaction ft) {
    // TODO Auto-generated method stub

}

public void OnCategorySelected(int position) {
    // TODO Auto-generated method stub

    System.out.println("Category Selected: " + position);

    switch (position) {

        case 0: {

            WelcometocyFragment welcome = new
WelcometocyFragment();

            FragmentTransaction ft = fm.beginTransaction();

            ft.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE);
            ft.replace(R.id.fragment_container, welcome);

            ft.addToBackStack(null);

            ft.commit();

            break;

        }

        default:
        {
            Category = position;

```

```

        CityFragment city = new CityFragment();
        FragmentTransaction ft =
getFragmentManager().beginTransaction();

        ft.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE);
        ft.replace(R.id.fragment_container, city);
        ft.addToBackStack(null);
        ft.commit();
    }

}

private void insertPlacestoDatabase(PlaceDataSource datasource, List<Place>
places ) {

    // TODO Auto-generated method stub

    places.add(datasource.createPlace("Panayia Asinou", "Nicosia",
"Must See", 35.045923,32.973361,ShortClass.asinouurl,ShortClass.asinoudesc,
ShortClass.asinouurl));
    places.add(datasource.createPlace("Ayios Nikolaos Stegis",
"Nicosia", "Must See",
34.977707,32.889419,ShortClass.ayiosnikolaosstegisurl,ShortClass.ayiosnikolaosstegis
descr,ShortClass.ayiosnikolaosstegistel));
    places.add(datasource.createPlace("Panayia tou Araka",
"Nicosia", "Must See",
34.965636,33.006974,ShortClass.arakasurl,ShortClass.arakasdesc,ShortClass.arakastel)
);
    places.add(datasource.createPlace("Old Kakopetria", "Nicosia",
"Must See",
34.990136,32.903152,ShortClass.kakopetriaurl,ShortClass.kakopetriadesc,ShortClass.k
akopetriatel));
    places.add(datasource.createPlace("Fikardou", "Nicosia", "Must
See",
34.95916,33.171329,ShortClass.fikardouurl,ShortClass.fikardoudesc,ShortClass.fikardo
utel));
    places.add(datasource.createPlace("Ancient Tamassos",
"Nicosia", "Must See",
35.031472,33.244114,ShortClass.tamassosurl,ShortClass.tamassosdesc,ShortClass.tama
ssostel));
    places.add(datasource.createPlace("Machairas Monastery",
"Nicosia", "Must See",
34.940777,33.190749,ShortClass.machairasurl,ShortClass.machairasdesc,ShortClass.m
achairastel));
    places.add(datasource.createPlace("Panayia Chrisospilotissa",
"Nicosia", "Must See",

```



```

        places.add(datasource.createPlace("Berlin Wall", "Nicosia",
"Restaurants & Bars", 35.158517, 33.311065, ShortClass.prosilioURL,
ShortClass.prosilioDESCR, ShortClass.prosilioTEL));
        places.add(datasource.createPlace("Souvlatzidiko Taki",
"Restaurants & Bars", "Restaurant", 35.158517, 33.311065, ShortClass.prosilioURL,
ShortClass.prosilioDESCR, ShortClass.prosilioTEL));
        places.add(datasource.createPlace("Andreas", "Nicosia",
"Restaurants & Bars", 35.158517, 33.311065, ShortClass.prosilioURL,
ShortClass.prosilioDESCR, ShortClass.prosilioTEL));
        places.add(datasource.createPlace("Aerikon", "Nicosia",
"Restaurants & Bars", 35.158517, 33.311065, ShortClass.prosilioURL,
ShortClass.prosilioDESCR, ShortClass.prosilioTEL));
        places.add(datasource.createPlace("Steki Loksandras", "Nicosia",
"Restaurants & Bars", 35.158517, 33.311065, ShortClass.prosilioURL,
ShortClass.prosilioDESCR, ShortClass.prosilioTEL));
        places.add(datasource.createPlace("Oktana", "Nicosia",
"Restaurants & Bars", 35.158517, 33.311065, ShortClass.prosilioURL,
ShortClass.prosilioDESCR, ShortClass.prosilioTEL));
        places.add(datasource.createPlace("Erodos", "Nicosia",
"Restaurants & Bars", 35.158517, 33.311065, ShortClass.prosilioURL,
ShortClass.prosilioDESCR, ShortClass.prosilioTEL));
        places.add(datasource.createPlace("Ksefoto", "Nicosia",
"Restaurants & Bars", 35.158517, 33.311065, ShortClass.prosilioURL,
ShortClass.prosilioDESCR, ShortClass.prosilioTEL));
        places.add(datasource.createPlace("Pixida", "Nicosia",
"Restaurants & Bars", 35.158517, 33.311065, ShortClass.prosilioURL,
ShortClass.prosilioDESCR, ShortClass.prosilioTEL));
        places.add(datasource.createPlace("Psarolimano Filippos",
"Nicosia", "Restaurants & Bars", 35.158517, 33.311065, ShortClass.prosilioURL,
ShortClass.prosilioDESCR, ShortClass.prosilioTEL));
        places.add(datasource.createPlace("Domus Lounge Bar &
Restaurant", "Nicosia", "Restaurants & Bars", 35.158517, 33.311065,
ShortClass.prosilioURL, ShortClass.prosilioDESCR, ShortClass.prosilioTEL));

        places.add(datasource.createPlace("Blinkers", "Nicosia",
"Nightlife", 35.163413, 33.332656, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));
        places.add(datasource.createPlace("City Pride", "Nicosia",
"Nightlife", 35.163413, 33.332656, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));
        places.add(datasource.createPlace("Occhio Club", "Nicosia",
"Nightlife", 35.163413, 33.332656, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));
        places.add(datasource.createPlace("Occhio Air", "Nicosia",
"Nightlife", 35.163413, 33.332656, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));
        places.add(datasource.createPlace("Zoo Club", "Nicosia",
"Nightlife", 35.163413, 33.332656, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));

```

```

        places.add(datasource.createPlace("Toy Club", "Nicosia",
"Nightlife", 35.163413, 33.332656, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));
        places.add(datasource.createPlace("Sfinakia Club", "Nicosia",
"Nightlife", 35.163413, 33.332656, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));
        places.add(datasource.createPlace("Mutka Lounge Bar",
"Nicosia", "Nightlife", 35.163413, 33.332656, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));
        places.add(datasource.createPlace("Shiraz Club", "Nicosia",
"Nightlife", 35.163413, 33.332656, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));
        places.add(datasource.createPlace("Mberdema Gold Club",
"Nicosia", "Nightlife", 35.163413, 33.332656, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));
        places.add(datasource.createPlace("Monte Cristo", "Nicosia",
"Nightlife", 35.163413, 33.332656, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));
        places.add(datasource.createPlace("Zoo Lounge", "Nicosia",
"Nightlife", 35.163413, 33.332656, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));
        places.add(datasource.createPlace("Q Lounge", "Nicosia",
"Nightlife", 35.163413, 33.332656, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));

        places.add(datasource.createPlace("The Mall of Cyprus",
"Nicosia", "Shopping", 35.163069, 33.316716, ShortClass.themallofcyprusURL,
ShortClass.themallofcyprusDESCR, ShortClass.themallofcyprusTEL));
        places.add(datasource.createPlace("Madisons Mall", "Nicosia",
"Shopping", 35.163069, 33.316716, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));
        places.add(datasource.createPlace("MyMall Limassol",
"Limassol", "Shopping", 35.163069, 33.316716, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));
        places.add(datasource.createPlace("The Paphos Mall", "Paphos",
"Shopping", 35.163069, 33.316716, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));
        places.add(datasource.createPlace("The City Plaza", "Nicosia",
"Shopping", 35.163069, 33.316716, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));
        places.add(datasource.createPlace("Nicosia City Mall",
"Nicosia", "Shopping", 35.163069, 33.316716, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));
        places.add(datasource.createPlace("The Mall of Engomi",
"Nicosia", "Shopping", 35.163069, 33.316716, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));
        places.add(datasource.createPlace("Orphanides Limassol Mall",
"Nicosia", "Shopping", 35.163069, 33.316716, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));

```

```

        places.add(datasource.createPlace("King's Avenue Mall",
"Paphos", "Shopping", 35.163069, 33.316716, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));

```

```

        places.add(datasource.createPlace("Lakatamia Police Station ",
"Nicosia", "Emergency", 35.148394, 33.320750, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));

```

```

        places.add(datasource.createPlace("Latakia Fire Dept. Station",
"Nicosia", "Emergency", 35.148394, 33.320750, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));

```

```

        places.add(datasource.createPlace("Nicosia General Hospital",
"Nicosia", "Emergency", 35.148394, 33.320750, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));

```

```

        places.add(datasource.createPlace("Apollonion Private Hospital",
"Nicosia", "Emergency", 35.148394, 33.320750, ShortClass.apollonioURL,
ShortClass.prosilioDESCR, "0035722351550"));

```

```

        places.add(datasource.createPlace("Evangelistria Clinic",
"Nicosia", "Emergency", 35.148394, 33.320750, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));

```

```

        places.add(datasource.createPlace("American Heart Institute",
"Nicosia", "Emergency", 35.148394, 33.320750, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));

```

```

        places.add(datasource.createPlace("Ayios Therissos MRI
Centre", "Nicosia", "Emergency", 35.148394, 33.320750, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));

```

```

        places.add(datasource.createPlace("Areteion Private Hospital",
"Nicosia", "Emergency", 35.148394, 33.320750, ShortClass.BlinkersURL,
ShortClass.prosilioDESCR, "0035722351550"));

```

```

    }

```

```

    public void onBackPressed() {

```

```

        if (fm.getBackStackEntryCount() == 0 ) {
            // || fm.getBackStackEntryAt(0) !=
            new AlertDialog.Builder(this)
                .setTitle("Exit LoveCyprus?")
                .setIcon(R.drawable.ic_launcher)
                .setMessage("Are you sure you want to exit?")
                .setCancelable(false)
                .setPositiveButton("Yes",
                    new

```

```

DialogInterface.OnClickListener() {

```

```

                                public                                void
onClick(DialogInterface dialog,

```

```

                                int id) {
                                datasource.close();

                                MainActivity.this.finish();
                                }
                                }).setNegativeButton("No",
null).show();
                                } else {
                                    fm.popBackStack();
                                }
                            }

@Override
public boolean onQueryTextChange(String newText) {
    // TODO Auto-generated method stub

    return false;
}

@Override
public boolean onQueryTextSubmit(String query) {
    // TODO Auto-generated method stub

    return false;
}

@Override
public void onLocationChanged(Location location) {
    // TODO Auto-generated method stub
    double latitude = location.getLatitude();
    this.setLatitude(latitude);

    // Getting longitude of the current location
    double longitude = location.getLongitude();
    this.setLongitude(longitude);
}

public double getLatitude() {
    return latitude;
}

public void setLatitude(double latitude) {
    this.latitude = latitude;
}

```

```

    }

    public double getLongitude() {
        return longitude;
    }

    public void setLongitude(double longitude) {
        this.longitude = longitude;
    }

    @Override
    public void onProviderDisabled(String provider) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onProviderEnabled(String provider) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onStatusChanged(String provider, int status, Bundle extras) {
        // TODO Auto-generated method stub

    }

    @Override
    public void OnMustSeeSelected(int position) {
        // TODO Auto-generated method stub

        Intent intent = new Intent(this, InfoActivity.class);

        Bundle bundle = new Bundle();

        bundle.putLong("id",
MustSeeFragment.mustseeplaces.get(position).getId());

        bundle.putString("name",MustSeeFragment.mustseeplaces.get(position).getName());

        bundle.putString("city",MustSeeFragment.mustseeplaces.get(position).getCity());
        bundle.putString("type",
MustSeeFragment.mustseeplaces.get(position).getType());
        bundle.putDouble("lat",
MustSeeFragment.mustseeplaces.get(position).getLatitude());

```

```

        bundle.putDouble("long",MustSeeFragment.mustseeplaces.get(position).getLongitude()
        );
            bundle.putString("imageurl",
MustSeeFragment.mustseeplaces.get(position).getImageurl());
            bundle.putString("description",
MustSeeFragment.mustseeplaces.get(position).getDescription());
            bundle.putString("telno",
MustSeeFragment.mustseeplaces.get(position).getTelno());
            bundle.putDouble("currentlat",this.getLatitude() );
            bundle.putDouble("currentlong", this.getLongitude());


        intent.putExtras(bundle);
        startActivity(intent);

    }
    @Override

    public void OnRestaurantSelected(int position) {
        // TODO Auto-generated method stub
        Intent intent = new Intent(this, InfoActivity.class);

        Bundle bundle = new Bundle();

        bundle.putLong("id",
RestaurantsFragment.restaurants.get(position).getId());

        bundle.putString("name",RestaurantsFragment.restaurants.get(position).getName());

        bundle.putString("city",RestaurantsFragment.restaurants.get(position).getCity());
        bundle.putString("type",
RestaurantsFragment.restaurants.get(position).getType());
        bundle.putDouble("lat",
RestaurantsFragment.restaurants.get(position).getLatitude());

        bundle.putDouble("long",RestaurantsFragment.restaurants.get(position).getLongitude())
        ;
            bundle.putString("imageurl",
RestaurantsFragment.restaurants.get(position).getImageurl());
            bundle.putString("description",
RestaurantsFragment.restaurants.get(position).getDescription());
            bundle.putString("telno",
RestaurantsFragment.restaurants.get(position).getTelno());
            bundle.putDouble("currentlat",this.getLatitude() );

```

```

        bundle.putDouble("currentlong", this.getLongitude());

        intent.putExtras(bundle);
        startActivity(intent);
    }

    @Override
    public void OnShoppingSelected(int position) {
        // TODO Auto-generated method stub
        Intent intent = new Intent(this, InfoActivity.class);

        Bundle bundle = new Bundle();

        bundle.putLong("id", ShoppingFragment.shops.get(position).getId());

        bundle.putString("name", ShoppingFragment.shops.get(position).getName());

        bundle.putString("city", ShoppingFragment.shops.get(position).getCity());
        bundle.putString("type",
            ShoppingFragment.shops.get(position).getType());
        bundle.putDouble("lat",
            ShoppingFragment.shops.get(position).getLatitude());

        bundle.putDouble("long", ShoppingFragment.shops.get(position).getLongitude());
        bundle.putString("imageurl",
            ShoppingFragment.shops.get(position).getImageurl());
        bundle.putString("description",
            ShoppingFragment.shops.get(position).getDescription());
        bundle.putString("telno",
            ShoppingFragment.shops.get(position).getTelno());
        bundle.putDouble("currentlat", this.getLatitude() );
        bundle.putDouble("currentlong", this.getLongitude());

        intent.putExtras(bundle);
        startActivity(intent);
    }

    @Override
    public void OnNightlifeSelected(int position) {
        // TODO Auto-generated method stub
        Intent intent = new Intent(this, InfoActivity.class);

        Bundle bundle = new Bundle();

```

```

        bundle.putLong("id",
NightlifeFragment.nightplaces.get(position).getId());

bundle.putString("name",NightlifeFragment.nightplaces.get(position).getName());

bundle.putString("city",NightlifeFragment.nightplaces.get(position).getCity());
        bundle.putString("type",
NightlifeFragment.nightplaces.get(position).getType());
        bundle.putDouble("lat",
NightlifeFragment.nightplaces.get(position).getLatitude());

bundle.putDouble("long",NightlifeFragment.nightplaces.get(position).getLongitude());
        bundle.putString("imageurl",
NightlifeFragment.nightplaces.get(position).getImageurl());
        bundle.putString("description",
NightlifeFragment.nightplaces.get(position).getDescription());
        bundle.putString("telno",
NightlifeFragment.nightplaces.get(position).getTelno());
        bundle.putDouble("currentlat",this.getLatitude() );
        bundle.putDouble("currentlong", this.getLongitude());

        intent.putExtras(bundle);
        startActivity(intent);
    }

```

```

@Override
public void OnEmergencySelected(int position) {
    // TODO Auto-generated method stub
    Intent intent = new Intent(this, InfoActivity.class);

    Bundle bundle = new Bundle();

```

```

        bundle.putLong("id",
EmergencyFragment.emergencyplaces.get(position).getId());

bundle.putString("name",EmergencyFragment.emergencyplaces.get(position).getName(
));

bundle.putString("city",EmergencyFragment.emergencyplaces.get(position).getCity());
        bundle.putString("type",
EmergencyFragment.emergencyplaces.get(position).getType());
        bundle.putDouble("lat",
EmergencyFragment.emergencyplaces.get(position).getLatitude());

bundle.putDouble("long",EmergencyFragment.emergencyplaces.get(position).getLongit
ude());

```

```

        bundle.putString("imageurl",
EmergencyFragment.emergencyplaces.get(position).getImageurl());
        bundle.putString("description",
EmergencyFragment.emergencyplaces.get(position).getDescription());
        bundle.putString("telno",
EmergencyFragment.emergencyplaces.get(position).getTelno());
        bundle.putDouble("currentlat",this.getLatitude() );
        bundle.putDouble("currentlong", this.getLongitude());

        intent.putExtras(bundle);
        startActivity(intent);
    }
    @Override
    public void OnCitySelected(int position) {
        // TODO Auto-generated method stub
        CityNumber = position;

        switch (Category) {

            case 1:
            {
                MustSeeFragment must = new MustSeeFragment();
                FragmentTransaction ft =
getFragmentManager().beginTransaction();

                ft.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE);
                ft.replace(R.id.fragment_container, must);
                ft.addToBackStack(null);
                ft.commit();

                break;
            }

            case 2:
            {
                RestaurantsFragment rest = new RestaurantsFragment();
                FragmentTransaction ft =
getFragmentManager().beginTransaction();

                ft.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE);
                ft.replace(R.id.fragment_container, rest);
                ft.addToBackStack(null);
                ft.commit();

```

```

        break;
    }

    case 3:
    {
        NightlifeFragment nightlife = new NightlifeFragment();
        FragmentTransaction ft =
getFragmentManager().beginTransaction();

        ft.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE);
        ft.replace(R.id.fragment_container, nightlife);
        ft.addToBackStack(null);
        ft.commit();

        break;
    }

    case 4:
    {
        ShoppingFragment shopping = new ShoppingFragment();
        FragmentTransaction ft =
getFragmentManager().beginTransaction();

        ft.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE);
        ft.replace(R.id.fragment_container, shopping);
        ft.addToBackStack(null);
        ft.commit();

        break;
    }

    case 5:
    {
        EmergencyFragment emergency = new EmergencyFragment();
        FragmentTransaction ft =
getFragmentManager().beginTransaction();

```

```
ft.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE);
    ft.replace(R.id.fragment_container, emergency);
    ft.addToBackStack(null);
    ft.commit();

    break;
}

}

}

}
```