



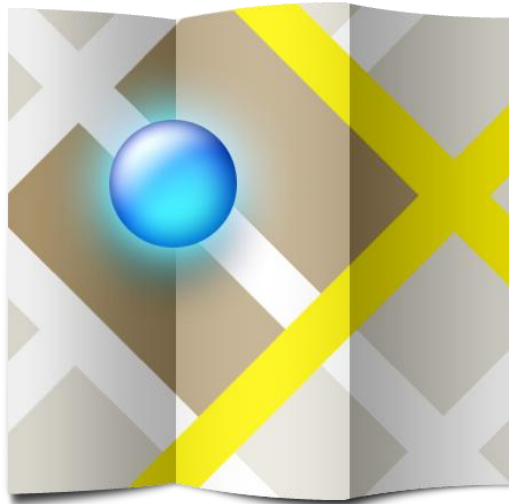
Τμήμα Μηχανικών
Πληροφορικής Τ.Ε



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

Πτυχιακή Εργασία

Ανάπτυξη πλατφόρμας και εφαρμογής κινητού τηλεφώνου για
παιχνίδι τύπου «κρυμμένος θησαυρός» ως μέσο ξενάγησης και
διαφήμισης



Φοιτητής
Γκούτζος Άρης
ΑΜ:06/3036

Επιβλέπων Καθηγητής
Κουρουπέτρογλου Χρήστος

ΘΕΣΣΑΛΟΝΙΚΗ 2013

Γκούτζος Άρης

«Δηλώνω υπεύθυνα ότι το παρόν κείμενο αποτελεί προϊόν προσωπικής μελέτης και εργασίας και πως όλες οι πηγές που χρησιμοποιήθηκαν για τη συγγραφή της δηλώνονται σαφώς είτε στις παραπομπές είτε στη βιβλιογραφία».

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή μου, κ. Κουρουπέτρογλου Χρήστο, κυρίως για την εμπιστοσύνη που μου έδειξε, αναθέτοντας μου αυτήν την πτυχιακή εργασία και αναλαμβάνοντας την επίβλεψη της. Όπως επίσης και για την πολύτιμη βοήθεια και καθοδήγηση που μου προσέφερε για την επίλυση διάφορων θεμάτων.

Θα ήθελα επίσης να απευθύνω τις ευχαριστίες μου στους γονείς μου και την αδερφή μου, στους οποίους κι αφιερώνω αυτήν την πτυχιακή εργασία, για την στήριξη τους όλο αυτό το διάστημα που διήρκησαν οι σπουδές μου, σε αποφάσεις σημαντικές για την σταδιοδρομία μου.

Τέλος, ευχαριστώ τους συμφοιτητές μου και τους φίλους μου, για τις συμβουλές και τις γνώσεις που μου προσέφεραν σε όλους τους τομείς και έπαιξαν καθοριστικό ρόλο για την ολοκλήρωση των σπουδών μου.

Περίληψη

Το παρόν κείμενο παρουσιάζει μια εφαρμογή για το λειτουργικό σύστημα Android με το όνομα CityExplore, που αναπτύχθηκε ως πτυχιακή εργασία για το τμήμα Πληροφορικής του Αλεξάνδρειου ΤΕΙ Θεσσαλονίκης.

Το CityExplore είναι μια εφαρμογή που δίνει τη δυνατότητα στον χρήστη να κερδίζει προσφορές και δώρα σε μορφή κουπονιών ολοκληρώνοντας αποστολές που βρίσκονται σε κοντινή απόσταση από αυτόν, λαμβάνοντας την τρέχουσα τοποθεσία του. Ο εντοπισμός της τρέχουσας τοποθεσίας γίνεται με χρήση GPS σήματος ή υπολογίζεται από την ενεργοποιημένη σύνδεση στο ίντερνετ που έχει ο χρήστης, είτε αυτή πραγματοποιείται μέσω WiFi είτε μέσω του δικτύου κινητής τηλεφωνίας. Η εμφάνιση των κοντινότερων στην τρέχουσα τοποθεσία αποστολών γίνεται με χρήση του Google Maps API και τοποθέτηση σημαδιών στον χάρτη.

Οι αποστολές και τα κουπόνια δημιουργούνται και διαχειρίζονται από τοπικούς φορείς, δημόσιους ή ιδιωτικούς, όπως δήμοι και επιχειρήσεις και η εξαργύρωση τους γίνεται με την παράδοση του κωδικού ή σκανάροντας το QRcode που αντιστοιχούν στην έκαστοτε ολοκληρωμένη αποστολή και εμφανίζονται στην οθόνη του χρήστη.

Δεν έχει γίνει η ανάπτυξη συστήματος για την δημιουργία και την διαχείριση των αποστολών από τους τοπικούς φορείς, καθώς ο στόχος της πτυχιακής εργασίας είναι η χρήση σύγχρονων τεχνολογιών για την ανάπτυξη εφαρμογής Android.

Λέξεις Κλειδιά

- 1.Marker : είναι το “σημάδι” των που εμφανίζεται πάνω στον χάρτη όταν στο συγκεκριμένο σημείο υπάρχει κάτι ενδιαφέρον. Στην παρούσα εργασία οι αποστολές σημαδεύονται με ‘Markers’.
- 2.QRcode : είναι ένα barcode 2 διαστάσεων το οποίο παράγεται όταν ο χρήστης ολοκληρώσει μια αποστολή και αντιστοιχίζεται σε αυτήν και τον χρήστη. Κάθε QRcode είναι μοναδικό.
- 3.HashMap : Στην πληροφορική ένα HashMap είναι μια δομή δεδομένων που χρησιμοποιείται για την υλοποίηση ενός σχετικού πίνακα, που αντιστοιχεί κλειδιά σε τιμές. Χρησιμοποιώντας αυτά τα κλειδιά βρίσκουμε τις τιμές που ψάχνουμε μέσα στον πίνακα.
- 4.Location Based Services (LBS) : Τέτοιες είναι οι εφαρμογές που κάνουν χρήση των υπηρεσιών τοποθεσίας, όπως ο υπολογισμός της τρέχουσας τοποθεσίας, που τους παρέχει η συσκευή τους.

Πίνακας Περιεχομένων

Ευχαριστίες.....	3
Περίληψη.....	4
Λέξεις Κλειδιά.....	5
Πίνακας Περιεχομένων	6
Ευρετήριο Εικόνων	8
ΚΕΦΑΛΑΙΟ 1. Εισαγωγή	10
1.1 Στόχος.....	10
1.2 Location Based Services	10
1.3 Παρόμοιες εφαρμογές.....	11
1.4 Δομή πτυχιακής εργασίας.....	15
ΚΕΦΑΛΑΙΟ 2. Πλατφόρμα ανάπτυξης εφαρμογών Android	17
2.1 Εισαγωγή.....	17
2.2 Τι είναι το Android;.....	18
2.3 Αρχιτεκτονική του Android	18
2.4 Βασικά στατιστικά εφαρμογών Android	22
2.5 Αρχείο AndroidManifest.xml	26
2.6 Γιατί Android;	31
2.7 Επίλογος	32
ΚΕΦΑΛΑΙΟ 3. Google Maps.....	33
3.1 Εισαγωγή.....	33
3.2 Υλοποίηση.....	34
3.3 Google Maps API (Application Programming Interface)	34
3.4 Χάρτες για smartphones και εξέλιξη τους	35
3.5 Google Maps και Android	37
3.6 Επίλογος	38
ΚΕΦΑΛΑΙΟ 4. Απαιτήσεις και ανάλυση της εφαρμογής	39
4.1 Εισαγωγή.....	39
4.2 Εργαλεία και τεχνολογίες που χρησιμοποιήθηκαν.....	40
4.3 Σχεδιασμός βάσης δεδομένων	42
4.4 Επικοινωνία με τη βάση δεδομένων	45
4.5 Καταγραφή απαιτήσεων και σχεδιασμός της εφαρμογής	47

4.5.1	Σύνδεση με τον σέρβερ για πιστοποίηση ή εγγραφή του χρήστη.....	48
4.5.2	Ενσωμάτωση των Google Maps στην εφαρμογή.....	50
4.5.3	Λήψη τοποθεσίας και εμφάνιση της στον χάρτη	54
4.5.4	Εμφάνιση των κοντινότερων στην τρέχουσα τοποθεσία αποστολών.....	55
4.5.5	Επιλογή αποστολής, εμφάνιση λεπτομερειών και ολοκλήρωση της	56
4.5.6	Εμφάνιση ιστορικού ολοκληρωμένων αποστολών και αντίστοιχων βραβείων.....	58
4.6	Επίλογος	60
ΚΕΦΑΛΑΙΟ 5. Χρήση της εφαρμογής		61
5.1	Εισαγωγή.....	61
5.2	Φόρμες της εφαρμογής και ανάλυση τους.....	62
5.2.1	Πιστοποίηση και εγγραφή χρήστη.....	62
5.2.2	Κεντρικό μενού.....	66
5.2.3	Εμφάνιση και εκπλήρωση αποστολών στον χάρτη.....	70
5.2.4	Εμφάνιση ιστορικού και βραβείων	73
5.3	Επίλογος	73
6. Συμπεράσματα και περιθώρια ανάπτυξης		75
Βιβλιογραφία και αναφορές		77

Ευρετήριο Εικόνων

Εικόνα 1 Το περιβάλλον χρήστη της εφαρμογής Foursquare.....	12
Εικόνα 2 Το περιβάλλον χρήστη της εφαρμογής TripAdvisor	13
Εικόνα 3 Το περιβάλλον χρήστη της εφαρμογής CheckIn	15
Εικόνα 4 Το λογότυπο του Android	17
Εικόνα 5 Αρχιτεκτονική του Android	19
Εικόνα 6 Κύκλος ζωής ενός Activity	21
Εικόνα 7 Αντιστοιχία API level και ονομασία έκδοσης του Android	31
Εικόνα 8 Google Maps μέσα από browser	33
Εικόνα 9 Google Maps για Smartphone	35
Εικόνα 10 Λογότυπα γλωσσών προγραμματισμού που χρησιμοποιήθηκαν	40
Εικόνα 11 Σχεδιάγραμμα της βάσης δεδομένων	43
Εικόνα 12 Επικοινωνία Android εφαρμογής με βάση δεδομένων	45
Εικόνα 13 Δεδομένα κωδικοποιημένα σε JSON	47
Εικόνα 14 Πολυνημάτωση	48
Εικόνα 15 Είσοδος χρήστη	49
Εικόνα 16 Εγγραφή χρήστη	49
Εικόνα 17 Λογότυπο Google Maps	50
Εικόνα 18 Εμφάνιση απαραίτητου αποτυπώματος για δήλωση της εφαρμογής	53
Εικόνα 19 Οθόνη ρυθμίσεων Location services	55
Εικόνα 20 Τα σημάδια χάρτη που χρησιμοποιήσαμε	56
Εικόνα 21 Οθόνη εμφάνισης αποστολών	57
Εικόνα 22 Οθόνη εμφάνισης ολοκληρωμένων αποστολών	58
Εικόνα 23 Οθόνη εμφάνισης βραβείου και αντίστοιχου κωδικού	59
Εικόνα 24 Φόρμα εισαγωγής χρήστη	62
Εικόνα 25 Μήνυμα αδυναμίας σύνδεσης με το σύστημα	63

Εικόνα 26 Προειδοποιήσεις για σωστή εισαγωγή στοιχείων.....	64
Εικόνα 27 Φόρμα εγγραφής χρήστη.....	65
Εικόνα 28 Προειδοποίηση αδυναμίας διασταύρωσης κωδικών.....	66
Εικόνα 29 Κεντρικό μενού	67
Εικόνα 30 Μήνυμα επιβεβαίωσης εξόδου από το σύστημα.....	68
Εικόνα 31 Παράθυρο πληροφοριών εφαρμογής	68
Εικόνα 32 Μηνύματα σε περίπτωση μη ενεργοποιημένων υπηρεσιών	69
Εικόνα 33 Παραπομπή χρήστη στις ρυθμίσεις της συσκευής.....	70
Εικόνα 34 Εμφάνιση πληροφοριών επιλεγμένης αποστολής	71
Εικόνα 35 ολοκληρώνοντας μια αποστολή.....	72
Εικόνα 36 Επιτυχία ή αποτυχία ολοκλήρωσης αποστολής.....	72
Εικόνα 37 Λίστα ολοκληρωμένων αποστολών και πληροφορίες βραβείου	73

ΚΕΦΑΛΑΙΟ 1. Εισαγωγή

1.1 Στόχος

Ο κύριος στόχος της παρούσας πτυχιακής εργασίας ήταν η ανάπτυξη μιας εφαρμογής για το λειτουργικό σύστημα Android που θα έχει πρόσβαση στο ίντερνετ, θα πραγματοποιεί σύνδεση και ανταλλαγή στοιχείων με online βάση δεδομένων και θα κάνει χρήση της τελευταίας τεχνολογίας έκδοση του Google Maps API. Ο συνδυασμός τους για την δημιουργία μιας διαδικτυακής εφαρμογής γίνεται με σκοπό την λειτουργικότητα και την ευκολία που μπορούν να προσθέσουν σε αυτό το έργο. Η χρήση αυτών των τεχνολογιών απαιτεί την γνώση συγκεκριμένων πεδίων και για την υλοποίηση αυτής της εφαρμογής χρησιμοποιήθηκαν διαφορετικές γλώσσες προγραμματισμού.

1.2 Location Based Services

Τα Location Based Services (LBS) είναι μία εφαρμογή στις κινητές υπηρεσίες πληροφοριών που αναπτύχθηκαν τα τελευταία χρόνια, χάρη στην ραγδαία εξέλιξη της τεχνολογίας. Οι χρήστες που διαθέτουν συσκευές με δυνατότητα να υπολογίζουν την τοποθεσία τους, μπορούν να αναρωτιούνται και να ενημερώνονται σχετικά με τον περίγυρό τους, όπως για παράδειγμα να μάθουν για το ποιο εστιατόριο βρίσκεται πιο κοντά σε αυτούς, ποιο σουπερ μάρκετ κ.ά. οποιαδήποτε στιγμή και οπουδήποτε. Ενώ το παράδειγμα αυτό δείχνει τη σημασία των LBS, από την άλλη πλευρά, το περιβάλλον των κινητών συσκευών, η χωρική ιδιότητα της θέσης που εξαρτάται από τα δεδομένα και η κινητικότητα των χρηστών δημιουργούν μία μεγάλη πρόκληση στους προγραμματιστές σε αυτό το χώρο. Ένα Location Based Service, είναι μία κινητή εφαρμογή, που εξαρτάται από την τοποθεσία της κινητής συσκευής, όπως το κινητό τηλέφωνο.

Προκειμένου να είναι δυνατή η παροχή υπηρεσιών LBS, κάποια “συστατικά” είναι απαραίτητα, όπως οι κινητές συσκευές, η προσβασιμότητα σε δίκτυο επικοινωνίας, συστατικά τοποθέτησης και διακομιστές υπηρεσιών. Έτσι, λοιπόν, οι κινητές συσκευές είναι τα εργαλεία των χρηστών ώστε να στείλουν αιτήματα και να αποκτήσουν αποτελέσματα των αιτημάτων τους.

Τέτοιες συσκευές μπορεί να είναι φορητές συσκευές πλοήγησης (PND), Personal Data Assistants (PDA's), φορητοί υπολογιστές, κινητά τηλέφωνα κ.ά.

Όσον αφορά τα κινητά τηλέφωνα, οι υπηρεσίες LBS πρέπει να φροντίζουν να μην επιβαρύνουν το τηλέφωνο, να είναι “ελαφριές”, λόγω του περιορισμού μνήμης των κινητών τηλεφώνων, του μικρού μεγέθους οθόνης και του περιορισμού της ισχύς του επεξεργαστή και να εξοικονομούν μπαταρία

1.3 Παρόμοιες εφαρμογές

Παρακάτω παρουσιάζονται κάποιες εφαρμογές παρόμοιες με αυτή που αναπτύχθηκε. Αναφέρονται διαφορές και ομοιότητες, καθώς και κάποια στοιχεία τους τα όποια μας ενέπνευσαν θετικά ώστε να προχωρήσουμε στην υλοποίηση της εφαρμογής μας.

A. Foursquare

Το Foursquare εξελίσσεται με γοργούς ρυθμούς καθημερινά και σιγά σιγά εισβάλλει στην καθημερινότητα μας και στο κινητό μας τηλέφωνο. Παρακάτω θα αναλύσουμε τι είδους υπηρεσία είναι το Foursquare και τις κύριες διαφορές που έχει με την εφαρμογή που αναπτύξαμε.

Το Foursquare πρόκειται για μια από τις μεγαλύτερες δωρεάν υπηρεσίες κινητών τηλεφώνων και Internet που κυριαρχεί αυτή τη στιγμή στις ΗΠΑ και γενικότερα βρίσκεται στις πρώτες θέσεις, όσον αφορά το πλήθος χρηστών, σε διάφορες αγορές ανά τον κόσμο. Ειδικά από το 2011 και μετέπειτα έχει αποκτήσει πολλούς θαυμαστές και στην Ελλάδα. Το Foursquare δεν είναι τίποτα άλλο από ένα location based κοινωνικό δίκτυο που δίνει την δυνατότητα να κάνετε check in στις τοποθεσίες που επισκέπτεστε (από καφετέριες και εστιατόρια έως και μουσεία), να αφήνετε τα θετικά ή τα αρνητικά σας σχόλια, να συγκεντρώνεται πόντους για να ανταγωνιστείτε τους φίλους σας, να βλέπετε τι κάνουν οι φίλοι σας και να ανακαλύπτετε νέες τοποθεσίες.

Ο χρήστης με τα περισσότερα check in σε μια συγκεκριμένη τοποθεσία που επισκέπτεται συνέχεια θα αποτελέσει και τον δήμαρχο της τοποθεσίας (mayor)

Γκούτζος Άρης

και έτσι υπάρχει ένας άτυπος ανταγωνισμός για το ποιος θα γίνει ο ηγέτης του κάθε σημείου.

Στην εφαρμογή υπάρχουν σχεδόν όλα τα σημεία περασμένα για τις μεγάλες πόλεις της Ελλάδος όμως και όσα δεν υπάρχουν μπορεί ο οποιοσδήποτε χρήστης να τα προσθέσει και ταυτόχρονα να εμφανίζονται σε όλους τους χρήστες που είναι συνδεδεμένοι σε εκείνη την περιοχή.



Εικόνα 1 Το περιβάλλον χρήστη της εφαρμογής Foursquare

Ποιο είναι το αντίτιμο όμως ενός δημάρχου και ηγέτη ενός σημείου; Προς το παρόν οι πόντοι που μαζεύονται από τους χρήστες δημιουργούν μόνο έναν ανταγωνισμό μεταξύ φίλων και τα κέρδη είναι κυρίως ηθικά. Στην εφαρμογή μας υπάρχει η δυνατότητα μετά την ολοκλήρωση μιας αποστολής από τον χρήστη, είτε αυτή απαιτεί δήλωση παρουσίας στο σημείο είτε απάντηση σε κάποια ερώτηση, να κερδίσει υπηρεσίες ή υλικά αγαθά. Επίσης τα σημεία θεωρητικά προσθέτονται από τους αντίστοιχους ιδιοκτήτες τους και όχι από οποιονδήποτε χρήστη, ώστε να εξασφαλίζεται στο μέγιστο η αξιοπιστία τους.

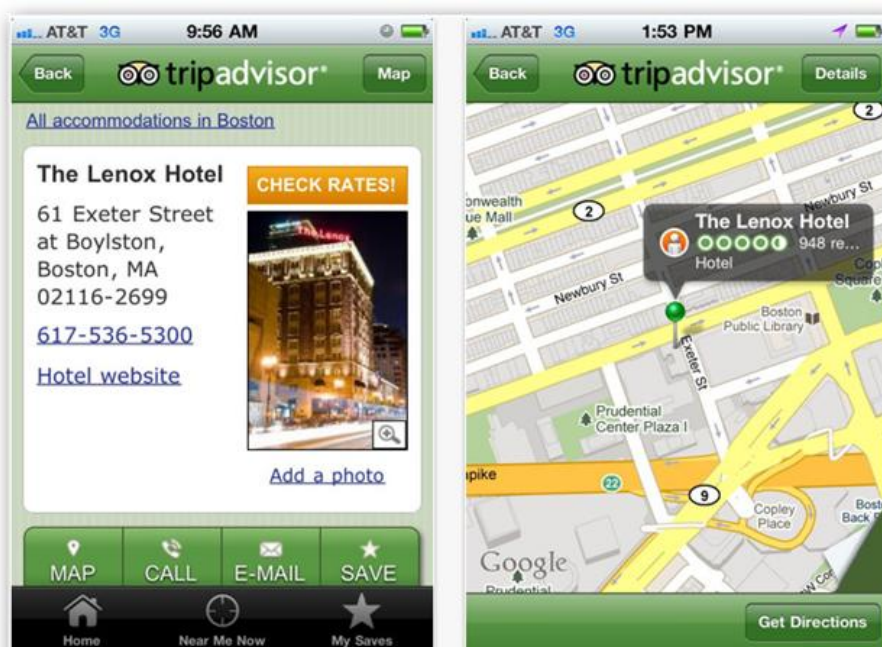
Τέλος, στα κοινά τους σημεία μπορούμε να προσθέσουμε την απαίτηση να διαθέτει ο χρήστης πρόσβαση στο ιντερνέτ και ενεργοποιημένα τα Location

Services ώστε να εμφανιστούν στον χάρτη τα κοντινότερα στην τρέχουσα τοποθεσία του διαθέσιμα σημεία.

B. TripAdvisor

Η συγκεκριμένη υπηρεσία αποτελεί την μεγαλύτερη και δημοφιλέστερη εφαρμογή για φορητές συσκευές στον ταξιδιωτικό τομέα. Όπως φανερώνει και το όνομα της προσφέρει συμβουλές για ταξιδιώτες για τις μεγαλύτερες και πιο τουριστικές πόλεις του κόσμου.

Ο χρήστης κατεβάζει την εφαρμογή που αντιστοιχεί στην πόλη που τον ενδιαφέρει και του δίνεται η δυνατότητα να βλέπει στην οθόνη της συσκευής του όλες τις διαθέσιμες τοποθεσίες που υπάρχουν για την συγκεκριμένη πόλη. Η βάση δεδομένων του TripAdvisor συμπεριλαμβάνει όλα εκείνα τα σημεία που είναι χρήσιμα για τους ταξιδιώτες, από δημόσια κτήρια και υπηρεσίες, όπως αστυνομικά τμήματα και νοσοκομεία, μέχρι ξενοδοχεία, εστιατόρια και νυχτερινά κέντρα, μέχρι και ATM και σταθμούς των μέσων μαζικής μεταφοράς. Υπάρχει η δυνατότητα να αξιολογήσει αυτά τα σημεία και να γράψει κριτικές ή να διαβάσει κριτικές άλλων χρηστών.



Εικόνα 2 Το περιβάλλον χρήστη της εφαρμογής TripAdvisor

Είναι μια βοηθητική υπηρεσία και δεν ανταποδίδει τον χρήστη με κάποιον τρόπο αλλά προσφέρει μια τεράστια βάση πληροφοριών που βοηθάνε τους ταξιδιώτες να προσαρμοστούν και να διασκεδάσουν στην πόλη του προορισμού τους. Του δίνεται η δυνατότητα να βλέπει τι υπάρχει στην γύρω περιοχή του ανάλογα με την τρέχουσα τοποθεσία του, να διαβάσει τις κριτικές που έχουν γράψει άλλοι χρήστες και να επιλέξει το κατάλληλο μέρος για δείπνο, διασκέδαση ή οτιδήποτε άλλο. Θα πρέπει να υπάρχει ενεργή σύνδεση στο ιντερνέτ από την συσκευή του χρήστη ή να έχει φροντίσει προηγουμένως να συγχρονίσει τα δεδομένα της συσκευής με τα δεδομένα της βάσης.

Η κύρια διαφορά με την εφαρμογή μας είναι ότι δεν χρειάζεται συνεχή σύνδεση στο ιντερνέτ αλλά δεν προσφέρει κάποιο βραβείο στον χρήστη παρά μια τεράστια βάση πληροφοριών για την πόλη.

Γ. Checkin

Το Checkin της Cypher Cove είναι μια εφαρμογή επί πληρωμή η οποία μας επιτρέπει με 2 γρήγορες κινήσεις να κάνουμε Check in σε μια τοποθεσία ή να ενεργοποιήσουμε το αυτόματο Check in, αν το επιθυμούμε, ώστε η όλη διαδικασία να γίνεται αυτόματα αν βρισκόμαστε εντός κάποιας ελάχιστης απόστασης από το σημείο. Η απόσταση αυτή είναι ρυθμιζόμενη από τον χρήστη. Συνδέοντας τον λογαριασμό που διατηρεί ένας χρήστης στο Facebook με την εφαρμογή - απαραίτητο αφού η εφαρμογή αναπτύχθηκε με σκοπό την διευκόλυνση των χρηστών του Facebook να "κάνουν" εύκολο και γρήγορο Check in - όλη η δραστηριότητα του χρήστη κοινοποιείται στον λογαριασμό του. Σημαντικό πλεονέκτημα της εφαρμογής είναι ότι δίνεται η δυνατότητα στον χρήστη πολύ γρήγορα να πραγματοποιήσει δήλωση παρουσίας σε μια τοποθεσία, είτε ορίζοντας στα αγαπημένα κάποια σημεία ώστε να γίνεται αυτόματα η διαδικασία αν βρίσκεται μέσα σε συγκεκριμένη απόσταση από αυτό το σημείο, είτε ανοίγοντας ο ίδιος την εφαρμογή και με 2 απλές κινήσεις ολοκληρωθεί χειροκίνητα η όλη διαδικασία. Ο συγκεκριμένος τρόπος λειτουργίας είναι όμοιος με αυτόν που αναπτύχθηκε στην εφαρμογή μας αφού διατίθενται στον χρήστη εκείνα τα σημεία που βρίσκονται εντός κάποια απόστασης από την τοποθεσία του ώστε να μπορέσει να ολοκληρώσει την αποστολή, ανεξαρτήτως το είδος της.



Εικόνα 3 Το περιβάλλον χρήστη της εφαρμογής CheckIn

Οι σημαντικότερες διαφορές, τις οποίες θα σημειώνα και ως μειονεκτήματα, είναι η εξάντληση της μπαταρίας της φορητής συσκευής που έχει εγκατεστημένη την εφαρμογή αν ο χρήστης έχει επιλέξει το αυτόματο Check in. Αυτό συμβαίνει διότι η συσκευή πραγματοποιεί συνεχώς λήψη δεδομένων από το δίκτυο για να ελέγξει αν ο χρήστης βρίσκεται μέσα στην οριζόμενη απόσταση από κάποιο σημείο που έχει επιλέξει στα αγαπημένα. Επίσης, η όλη εφαρμογή δεν προσφέρει κάποιο βραβείο στον χρήστη, ούτε ηθικό αλλά ούτε και υλικό, και παραδόξως είναι επί πληρωμή!

1.4 Δομή πτυχιακής εργασίας

Η δομή της πτυχιακής εργασίας εκτείνεται σε 5 κεφάλαια.

Το κεφάλαιο 1 αποτελείται από τον Στόχο, τις Παρόμοιες εφαρμογές και την Δομή της πτυχιακής εργασίας. Περιγράφει τις διαφορές και τις ομοιότητες της εφαρμογής μας με διάσημες εφαρμογές του εμπορίου, ποιά στοιχεία χρησιμοποιήσαμε από αυτές τις εφαρμογές και πως οδηγηθήκαμε στην δημιουργία της ιδέας.

Το κεφάλαιο 2 προσφέρει βασικές πληροφορίες για το λειτουργικό σύστημα Android. Επίσης αναλύεται η αρχιτεκτονική του, τα συστατικά που περιέχει η δημοφιλής αυτή πλατφόρμα. Τέλος περιγράφονται πως λειτουργούν οι εφαρμογές, οι οποίες είναι και το βασικότερο συστατικό του Android και επιτρέπουν στον χρήστη να αξιοποιήσει πλήρως τις δυνατότητες της συσκευής του.

Το κεφάλαιο 3 περιγράφει το εργαλείο που προσφέρει η Google δωρεάν στους προγραμματιστές που θέλουν να προσθέσουν χάρτη στην εφαρμογή τους ή να χρησιμοποιήσουν τα χαρακτηριστικά της συγκεκριμένης υπηρεσίας, το Google Maps. Επίσης αναφέρονται οι δυνατότητες που μας παρέχουν οι χάρτες αυτοί και παραδείγματα χρήσης τους.

Το κεφάλαιο 4 αναλύει τις απαιτήσεις της εφαρμογής και πώς αυτές υλοποιήθηκαν με τη βοήθεια διάφορων εργαλείων, προγραμμάτων και τεχνολογιών. Περιγράφεται πως η εφαρμογή μας διασπάστηκε σε κάποια αυτόνομα κομμάτια, ώστε να γίνει ευκολότερη η ανάπτυξη της και όσο το δυνατό απλούστερη η χρήση της.

Στο 5^ο κεφάλαιο περιγράφονται όλες οι φόρμες της εφαρμογής, μια προς μια, αναλύοντας τα βήματα που πρέπει να ακολουθηθούν από την αρχή ως το τέλος, από την δημιουργία του απαραίτητου λογαριασμού ως την ολοκλήρωση μιας αποστολής και την παραλαβή του ειδικού κωδικού του βραβείου. Επίσης θα δούμε πως οι σημαντικότερες πληροφορίες μεταφέρονται από την μια φόρμα στην άλλη για την σωστή ροή των δεδομένων και την επιτυχή λειτουργία της εφαρμογής.



Εικόνα 4 Το λογότυπο του Android

ΚΕΦΑΛΑΙΟ 2. Πλατφόρμα ανάπτυξης εφαρμογών Android

2.1 Εισαγωγή

Στα πλαίσια αυτής της πτυχιακής εργασίας, αναπτύχθηκε μια εφαρμογή για το λειτουργικό σύστημα Android. Η εφαρμογή αναπτύχθηκε με χρήση ενός ειδικά διαμορφωμένου πακέτου που μας επιτρέπει πολύ εύκολα να αναπτύσσουμε εφαρμογές, χρησιμοποιώντας σημαντικά εργαλεία και δυνατότητες που μας προσφέρει. Παρακάτω θα δούμε τα βασικά γνωρίσματα του συγκεκριμένου λειτουργικού συστήματος και θα κατανοήσουμε τις δυνατότητες της.

2.2 Τι είναι το Android;

Το Android είναι ένα λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας και ταμπλέτες βασισμένο σε πυρήνα Linux. Δημιουργήθηκε αρχικά από την Android Inc και το 2005 αγοράστηκε από τη Google όπου σε συνεργασία με την OpenHandsetAlliance έγινε η ανάπτυξη της πλατφόρμας. Η πρώτη παρουσίαση έγινε τον Νοέμβρη του 2007 ταυτόχρονα με την ανακοίνωση ίδρυσης του OHA, μιας κοινοπραξίας 48 εταιριών που ασχολούνται με την ανάπτυξη λογισμικού και την κατασκευή υλικού με στόχο την εξέλιξη των προτύπων τηλεπικοινωνίας στη κινητή τηλεφωνία. Έγινε εμπορικά διαθέσιμο το φθινόπωρο του 2008 με τη συσκευή HTC Dream, Linux kernel version 2.6 με πληκτρολόγιο που δεν υποστήριζε την οθόνη αφής παρά μόνο τη χρήση σκληρών κουμπιών της συσκευής. Στη συνέχεια ακολούθησαν και άλλες συσκευές με διαφορετικές δυνατότητες με αποκορύφωμα το Nexus κατασκευασμένο από την LG σε συνεργασία με την Google που κατείχε καινοτόμα χαρακτηριστικά σε σχέση με τους πρόγονους του καθώς και το τελευταίας γενιάς Galaxy S4 της Samsung το οποίο εκμεταλλεύεται στο έπακρο τις πιο σύγχρονες τεχνολογίες υλικού και σε συνδυασμό με το λειτουργικό σύστημα Android παρέχει στον χρήστη μια συσκευή πολυεργαλείο.

2.3 Αρχιτεκτονική του Android

Το Android τρέχει σε πυρήνα Linux και το λειτουργικό του σύστημα στα ανώτερα στρώματα αποτελείται από βιβλιοθήκες, το πλαίσιο εφαρμογής και απλές εφαρμογές γραμμένες από τους κατασκευαστές λογισμικού. Είναι μια στοίβα λογισμικού στην οποία κάθε επίπεδο διαθέτει συγκεκριμένες υπηρεσίες για το επόμενο επίπεδο, ξεκινώντας από τη διαχείριση των διεργασιών στον πυρήνα σε συνεργασία με τις εγγενής βιβλιοθήκες λογισμικού και τις εφαρμογές έως το οπτικό αποτέλεσμα στην οθόνη της συσκευής.



Εικόνα 5 Αρχιτεκτονική του Android

Πυρήνας Linux:

Για την ακρίβεια είναι μια απλουστευμένη έκδοση του Linux στην οποία υπολείπονται βασικά κομμάτια του κορμού. Στις ολοκληρωμένες εκδόσεις του Linux το σύστημα περιέχει δικές του βιβλιοθήκες, shell (κέλυφος) και editors με δικό τους γραφικό περιβάλλον κάτι που δε θα βρούμε αυτούσια στο Android. Επομένως ο πυρήνας ενός Android θα χαρακτηριζόταν ως μια γυμνή εκδοχή ενός συστήματος Linux που σκοπό έχει να χρησιμοποιήσει τα πλεονεκτήματα που του παρέχει. Το βασικότερο είναι η ελευθερία που παρέχει στους προγραμματιστές ως open source δίνοντας τους κίνητρο να πειραματιστούν με αυτό, που έχει ως αποτέλεσμα τη συνεχή ανάπτυξη του συστήματος. Χαρακτηρίζεται για την αξιοπιστία του και την ασφάλεια που διαθέτει στις βελτιωμένες νέες εκδόσεις της πλατφόρμας. Το Android του αναθέτει την διαχείριση της μνήμης, την διαχείριση των διεργασιών και άλλες πρωταρχικές λειτουργίες.

Dalvik runtime:

Σε αυτό το επίπεδο συναντάμε βασικές βιβλιοθήκες της Java και την εικονική μηχανή Dalvik. Η Dalvik είναι η μηχανή διεργασιών που τρέχει τις εφαρμογές στις συσκευές με Android. Οι εφαρμογές συνήθως γράφονται σε java ή C/C++ και μεταγλωττίζονται σε μορφή bytecode. Η εικονική μηχανή μετατρέπει τα αρχεία .class σε αρχεία .dex (DalvikExecutableformat) που είναι ειδικά διαμορφωμένα για συσκευές με περιορισμένη μνήμη και επεξεργαστική ισχύ. Για αυτό τον λόγο όλα τα APIs που χρησιμοποιούν οι εφαρμογές είναι ορισμένα σε κλάσεις, αντικείμενα και μεθόδους της μηχανής Dalvik.

Native libraries:

Αυτό το επίπεδο επιτρέπει στη συσκευή να χειρίζεται διαφορετικούς τύπους δεδομένων. Οι βιβλιοθήκες είναι γραμμένες σε γλώσσα C/C++ και έχουν δημιουργηθεί από τους κατασκευαστές για κάποια συγκεκριμένη συσκευή. Μερικές από αυτές είναι:

- SQLite: Είναι η μηχανή αποθήκευσης δεδομένων που χρησιμοποιεί το Android.
- WebKit: Είναι η μηχανή πλοήγησης που χρησιμοποιείται για την απεικόνιση html εγγράφων.
- OpenGL: Είναι η μηχανή που απεικονίζει τα δισδιάστατα και τρισδιάστατα γραφικά στην οθόνη της συσκευής.

Πλαίσιο εφαρμογής:

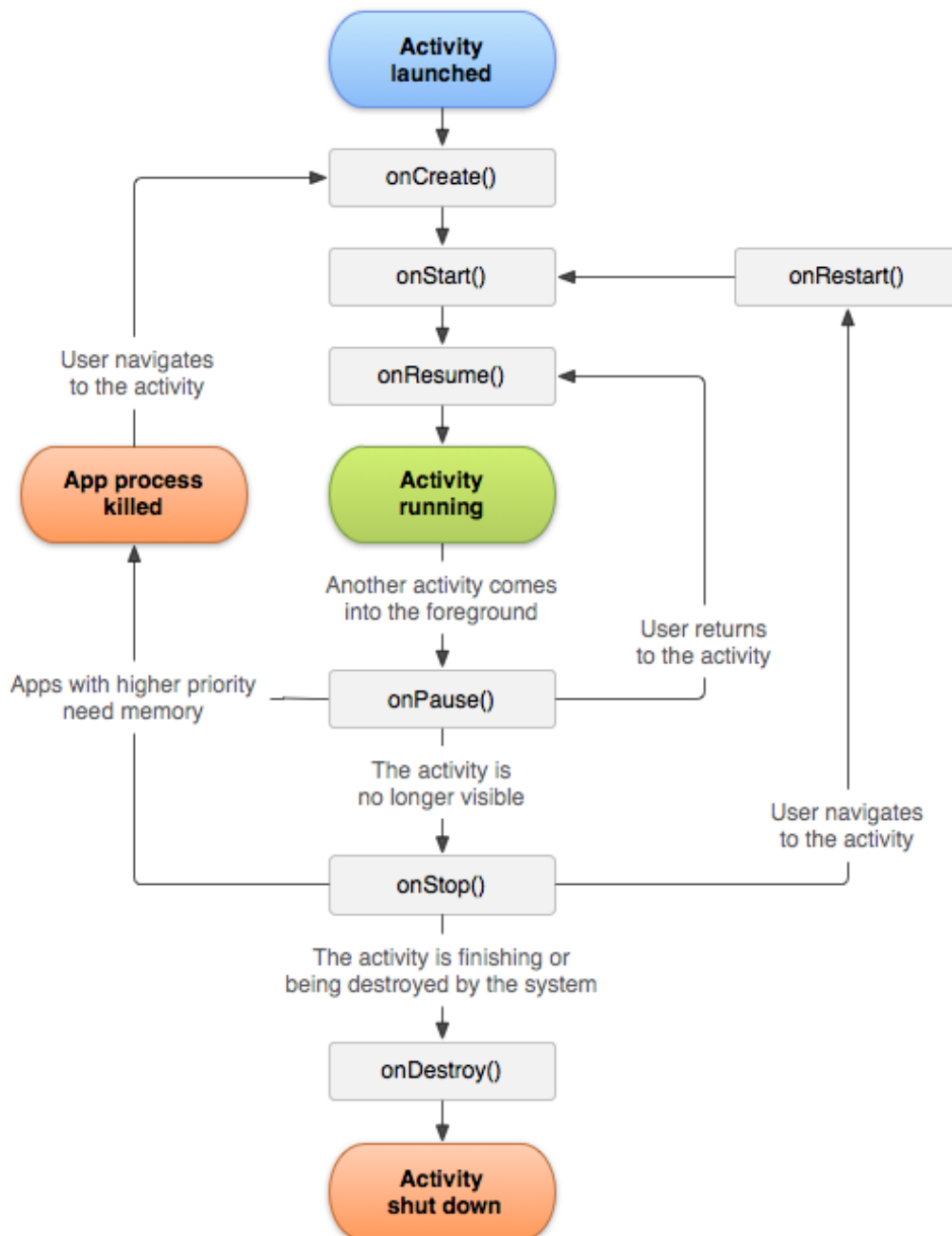
Αυτό είναι το επίπεδο με το οποίο οι εφαρμογές έχουν απευθείας αλληλεπίδραση. Εδώ εκτελούνται οι βασικές λειτουργίες της συσκευής τις οποίες μπορεί να χρησιμοποιήσει ένας προγραμματιστής για την εφαρμογή του.

Τα σημαντικότερα στοιχεία αυτού του επιπέδου είναι:

- Activity Manager – Διαχειριστής Δραστηριοτήτων: Είναι υπεύθυνο για τον έλεγχο του χρόνου ζωής των εφαρμογών.
- Content Providers – Διαχειριστής Περιεχομένου: Είναι υπεύθυνο για τις πληροφορίες που ανταλλάσσουν μεταξύ τους οι εφαρμογές.

Γκούτζος Άρης

- Location Manager – Διαχειριστής Τοποθεσίας: Βρίσκει την τοποθεσία χρησιμοποιώντας το GPS ή το δίκτυο.
- Resource Manager – Διαχειριστής Πόρων: Είναι υπεύθυνο για τα δεδομένα που τρέχουν στις εφαρμογές και δεν είναι κώδικας. Για παράδειγμα μπορεί να είναι σχεδιαγράμματα XML ή κωδικοί χρωμάτων.



Εικόνα 6 Κύκλος ζωής ενός Activity

Εφαρμογές:

Στο τελευταίο επίπεδο βρίσκονται οι εφαρμογές που έχουν δημιουργηθεί για απευθείας αλληλεπίδραση με τον χρήστη. Είναι προγράμματα που είτε έχουν εγκατασταθεί από τους κατασκευαστές λογισμικού είτε έχουν αναπτυχθεί από προγραμματιστές για να διευκολύνουν την καθημερινότητα των χρηστών. Είναι συνήθως γραμμένες σε γλώσσα Java από προγραμματιστές που χρησιμοποιούν το Android Software Development Kit ή σε γλώσσα C/C++ με χρήση του Native Development Kit. Οι προεγκατεστημένες εφαρμογές όπως το ημερολόγιο, οι επαφές και το Gmail συνεργάζονται άμεσα με την εικονική μηχανή Dalvik. Οι περισσότερες είναι opensource εκτός από αυτές που διατίθενται απευθείας από τη Google, όπως το GoogleMaps και το PlayStore. Από το GooglePlay υπάρχει η δυνατότητα αναζήτησης εφαρμογών εκ των οποίων οι περισσότερες διατίθενται δωρεάν. Επίσης κάποιος μπορεί να βρει εφαρμογές από άλλα site χωρίς όμως να φέρουν τον έλεγχο της Google. Τα αρχεία που κατεβάζει ο χρήστης είναι σε μορφή .apk που στην πραγματικότητα είναι ένας συμπιεσμένος .zip φάκελος και επιτρέπουν την εγκατάστασή τους χωρίς να αποσυμπιεστούν.

2.4 Βασικά στατιστικά εφαρμογών Android

Οι εφαρμογές Android αποτελούνται από κάποια συστατικά, που συνδέονται χρησιμοποιώντας ένα XML αρχείο, το AndroidManifest.xml του κάθε project, το οποίο περιγράφει κάθε συστατικό και πώς αλληλεπιδρά με τα άλλα. Υπάρχουν έξι συστατικά που παρέχουν τις δομικές μονάδες για τις εφαρμογές:

- **Δραστηριότητα (Activity)** - Είναι το επίπεδο παρουσίασης (presentation layer) της εφαρμογής. Μια δραστηριότητα είναι μια απλή οθόνη της εφαρμογής. Κάθε δραστηριότητα υλοποιείται σαν μια κλάση που επεκτείνει (extends) την βασική κλάση Δραστηριότητα (Activity base class). Η συγκεκριμένη κλάση προβάλλει μια διεπαφή χρήστη (user interface) αποτελούμενη από Όψεις (Views) και ανταποκρίνεται σε Συμβάντα (Events). Ένα Activity είναι ισοδύναμο με μια Φόρμα (Form) στην ανάπτυξη desktop εφαρμογών.
- **Υπηρεσία (Service)** - Μια Υπηρεσία είναι κώδικας που τρέχει για μεγάλο χρονικό διάστημα και χωρίς διεπαφή χρήστη (UI), ενημερώνοντας τις πηγές δεδομένων και τις ορατές Δραστηριότητες (Activities), ενεργοποιώντας Ειδοποιήσεις (Notifications). Αυτές τρέχουν και επεξεργάζονται δεδομένα ακόμα

και όταν οι Δραστηριότητες της εφαρμογής δεν είναι ενεργές ή ορατές. Ένα καλό παράδειγμα είναι μια εφαρμογή που αναπαράγει μουσική από μια λίστα μουσικών κομματιών (media player). Σε μια τέτοια εφαρμογή, θα υπήρχαν κατά πάσα πιθανότητα μία ή και παραπάνω Δραστηριότητες που επιτρέπουν στον χρήστη να επιλέξει τραγούδια και να τα αναπαράγει. Ωστόσο, η αναπαραγωγή από μόνη της δεν θα έπρεπε να διαχειρίζεται από την δραστηριότητα γιατί ο χρήστης θα περίμενε την μουσική να συνέχιζε να παίζει ακόμη και μετά την πλοήγησή του σε μια νέα οθόνη. Σε αυτή τη περίπτωση, η Δραστηριότητα της αναπαραγωγής μουσικής θα ξεκινούσε μια Υπηρεσία (Service) για να τρέξει στο παρασκήνιο και να συνεχίσει η μουσική να παίζει. Το σύστημα τότε θα κρατά την Υπηρεσία αναπαραγωγής ενεργή μέχρι να τελειώσει το κομμάτι. Όταν πραγματοποιηθεί σύνδεση σε μια Υπηρεσία, μπορεί να υπάρξει επικοινωνία με αυτή μέσω μιας διεπαφής που προσφέρεται από την Υπηρεσία. Για την Υπηρεσία μουσικής, αυτό θα επέτρεπε την παύση ή την επιστροφή πίσω στο κομμάτι (rewind) κλπ.

- **Παροχέας Περιεχομένου (Content Provider)** - Οι εφαρμογές μπορούν να σώζουν τα δεδομένα τους σε αρχεία, σε μια βάση δεδομένων SQLite ή με οποιοδήποτε άλλο μηχανισμό μπορούν. Ένας Παροχέας Περιεχομένου, ωστόσο, είναι χρήσιμος ώστε τα δεδομένα μιας εφαρμογής να είναι διαθέσιμα και σε άλλες εφαρμογές. Ένας Παροχέας Περιεχομένου είναι μια κλάση που υλοποιεί μια συγκεκριμένη ομάδα μεθόδων που επιτρέπουν σε άλλες εφαρμογές να αποθηκεύουν και να επανακτούν δεδομένα του τύπου που διαχειρίζεται ο Παροχέας Περιεχομένου. Οι συσκευές Android περιλαμβάνουν διάφορους εγγενείς Παροχείς Περιεχομένου (Content Providers) που εκθέτουν τις χρήσιμες βάσεις δεδομένων, όπως για παράδειγμα των στοιχείων των επαφών του χρήστη.

- **Προθέσεις (Intents)** - Το Android χρησιμοποιεί μια ειδική κλάση που λέγεται Πρόθεση (Intent) για να κινείται από οθόνη σε οθόνη. Η Πρόθεση περιγράφει τι θέλει η εφαρμογή να γίνει στη συνέχεια. Τα δυο πιο σημαντικά μέρη της δομής δεδομένων της Πρόθεσης είναι η Δράση (Action) και τα δεδομένα βάσει των οποίων αυτή θα εκτελεστεί. Τυπικές τιμές για μια δράση είναι η MAIN (η κεντρική είσοδος της εφαρμογής), VIEW, PICK, EDIT κλπ. Τα δεδομένα εκφράζονται ως URI (Uniform Resource Indicator). Για παράδειγμα, για να εμφανιστεί μια

ιστοσελίδα στον φυλλομετρητή (browser), δημιουργείται μια Πρόθεση με Δράση VIEW και τα δεδομένα ως ένα URL.

```
New Intent (android.content.Intent.VIEW_ACTION;  
ContentURI.create("http://addev.org"));
```

Υπάρχει μια σχετική κλάση που λέγεται Φίλτρο Πρόθεσης (IntentFilter). Ενώ μια Πρόθεση είναι στην ουσία ένα αίτημα για να γίνει κάτι, το Φίλτρο Πρόθεσης είναι μια περιγραφή του τι είναι δυνατόν να διαχειριστεί ένας Δέκτης Πρόθεσης (Intent Receiver). Μια Δραστηριότητα που είναι σε θέση να προβάλει πληροφορίες επικοινωνίας για ένα άτομο, θα ανακοίνωνε με ένα Φίλτρο Πρόθεσης (IntentFilter) ότι γνωρίζει πως να διαχειριστεί την VIEW_ACTION όταν τα δεδομένα αντιπροσωπεύουν ένα άτομο. Οι Δραστηριότητες ανακοινώνουν τα Φίλτρα Πρόθεσης (IntentFilters) στο AndroidManifest.xml αρχείο. Η πλοήγηση από οθόνη σε οθόνη πετυχαίνεται με Προθέσεις. Για να πλοηγηθεί κανείς προς τα μπρος, μια Δραστηριότητα καλεί την startActivity(myIntent). Το σύστημα τότε κοιτά στα Φίλτρα Προθέσεων (Intent Filters) για όλες τις εγκατεστημένες εφαρμογές και διαλέγει την Δραστηριότητα που τα Φίλτρα Πρόθεσης ταιριάζουν καλύτερα με την παράμετρο 'MyIntent' της κλήσης. Η νέα Δραστηριότητα ενημερώνεται για την Πρόθεση και ξεκινά. Η διαδικασία της υλοποίησης των Προθέσεων συμβαίνει κατά τον χρόνο εκτέλεσης της εφαρμογής, όταν δηλαδή καλείται η startActivity, πράγμα που προσφέρει 2 πλεονεκτήματα – κλειδιά:

1. Οι Δραστηριότητες μπορούν να επαναχρησιμοποιούν κάποια λειτουργικότητα από άλλα τμήματα του κώδικα απλά κάνοντας ένα αίτημα υπό την μορφή μιας πρόθεσης.
2. Οι Δραστηριότητες μπορούν να αντικατασταθούν οποιαδήποτε στιγμή από μια νέα Δραστηριότητα με ένα αντίστοιχο Φίλτρο Πρόθεσης

• **Δέκτες Πρόθεσης (Intent Receiver)** - Χρησιμοποιούνται για να εκτελεστεί μια εφαρμογή σε απάντηση ενός εξωτερικού συμβάντος (external event), για παράδειγμα, όταν το τηλέφωνο χτυπά, ή όταν το δίκτυο είναι διαθέσιμο, ή όταν είναι μεσάνυχτα. Οι Δέκτες Πρόθεσης δεν προβάλλουν μια διεπαφή χρήστη (UI), ωστόσο μπορούν να προβάλλουν Ειδοποιήσεις (Notifications) για να ειδοποιήσουν τον χρήστη για κάτι σημαντικό που συνέβη. Οι Δέκτες Πρόθεσης

είναι επίσης καταχωρημένοι στο `AndroidManifest.xml`, αλλά μπορούν επίσης καταχωρηθούν από τον κώδικα χρησιμοποιώντας την `Context.registerReceiver()`. Η εφαρμογή δεν χρειάζεται να τρέχει για να κληθούν οι Δέκτες Πρόθεσης που έχει. Το σύστημα θα κινήσει την εφαρμογή, αν χρειαστεί, όταν ένας Δέκτης Πρόθεσης ενεργοποιηθεί. Οι εφαρμογές μπορούν επίσης να στέλνουν τις δικές τους Ανακοινώσεις Πρόθεσης (Intent Broadcasts) σε άλλους με την `Context.broadcastIntent()`.

• **Δέκτες Μετάδοσης (Broadcast Receivers)** - Βασική Κλάση για τον κώδικα, που θα λαμβάνει τις Προθέσεις (Intents) που στέλνονται από το `sendBroadcast()`. Μπορεί είτε δυναμικά να καταχωρηθεί μια περίπτωση αυτής της κατηγορίας με `Context.registerReceiver()`, είτε να καταχωρηθεί ένα στιγμιότυπο αυτής της κλάσης σε μια εφαρμογή μέσω της ετικέτας `<receiver>` στο `AndroidManifest.xml`. Οι Δέκτες Μετάδοσης αρχίζουν αυτόματα την εφαρμογή αποκρινόμενοι σε μια εισερχόμενη Πρόθεση, που στην ουσία κάνει την εφαρμογή, μια εφαρμογή οδηγούμενη από τα γεγονότα. Υπάρχουν δύο σημαντικές κατηγορίες Μεταδόσεων (Broadcasts) που μπορούν να παραληφθούν:

1. Οι Κανονικές Μεταδόσεις - Normal broadcasts (που στέλνονται με `Context.sendBroadcast()`) είναι απολύτως ασύγχρονες. Όλοι οι Δέκτες της Μετάδοσης οργανώνονται σε μια απροσδιόριστη διαταγή, συχνά συγχρόνως. Αυτό είναι αποδοτικότερο, αλλά σημαίνει ότι οι Δέκτες δεν μπορούν να χρησιμοποιήσουν το αποτέλεσμα ή να αποβάλουν APIs που συμπεριλαμβάνονται εδώ.

2. Οι Μεταδόσεις Διαταγής - Ordered broadcasts (που στέλνονται με `Context.sendOrderedBroadcast()`) παραδίδονται σε έναν Δέκτη τη φορά. Δεδομένου ότι κάθε Δέκτης εκτελείται στη συνέχεια, μπορεί να διαδώσει ένα αποτέλεσμα στον επόμενο Δέκτη, ή μπορεί να αποβάλει τη Μετάδοση έτσι ώστε να μη περάσει σε άλλους δέκτες. Ο χειρισμός της εκτέλεσης των Δεκτών διαταγής μπορεί να γίνει με την ιδιότητα του Φίλτρου Προθέσεων Ταιριάσματος (Matching Intent-Filter). Οι Δέκτες με την ίδια προτεραιότητα τρέχουν με τυχαία σειρά.

• **Ειδοποιήσεις (Notifications)** - Είναι ένα πλαίσιο ειδοποιήσεων χρήστη. Οι Ειδοποιήσεις δίνουν την δυνατότητα ειδοποίησης του χρήστη χωρίς να υπάρξει

παρεμβολή στην τρέχουσα Δραστηριότητα (Activity) και του παρέχουν τις εξής δυνατότητες :

1. Να δημιουργήσει ένα νέο εικονίδιο status bar.
2. Να επιδείξει τις πρόσθετες πληροφορίες (και να προωθήσουν μια πρόθεση) στο εκτεταμένο παράθυρο status bar.
3. Να ενεργοποιήσει τα φώτα/LEDs.
4. Να δονηθεί το τηλέφωνο.
5. Να ενεργοποιήσει διάφορους ήχους (ringtones, ήχοι του τηλεφώνου κ.τ.λ).

Οι Ειδοποιήσεις είναι ο προτιμότερος τρόπος για τα αόρατα τμήματα της εφαρμογής (Δέκτες μετάδοσης, Υπηρεσίες, και ανενεργές Δραστηριότητες) για να προειδοποιήσουν τους χρήστες ότι κάποια γεγονότα που έχουν εμφανιστεί απαιτούν την προσοχή του χρήστη. Παραδείγματος χάριν, όταν λαμβάνει μια συσκευή ένα μήνυμα κειμένων ή μια εισερχόμενη κλήση, προειδοποιεί με τη λάμψη των φώτων, την παραγωγή ήχων, την επίδειξη εικονιδίων ή με την εμφάνιση μηνυμάτων διαλόγου. Ο Χειριστής Ειδοποιήσεων (Notification Manager) είναι μια Υπηρεσία συστήματος που χρησιμοποιείται για να χειριστεί τις ειδοποιήσεις. Καλείται μέσω της μεθόδου `getSystemService`, όπως φαίνεται παρακάτω:

```
String svcName = Context.NOTIFICATION_SERVICE;
```

```
NotificationManager notificationManager;
```

```
NotificationManager = (NotificationManager) getSystemService (svcName);
```

Χρησιμοποιώντας το Χειριστή Ειδοποιήσεων, μπορούν να ενεργοποιηθούν νέες ειδοποιήσεις, να τροποποιηθούν οι υπάρχουσες ή να αφαιρεθούν εκείνες που δεν είναι απαραίτητες ή επιθυμητές.

2.5 Αρχείο AndroidManifest.xml

Κάθε εφαρμογή πρέπει να έχει ένα αρχείο `AndroidManifest.xml` με ακριβώς αυτό το όνομα στον αρχικό (root) κατάλογο του. Στο αρχείο αυτό παρουσιάζονται οι απαραίτητες πληροφορίες σχετικά με την εφαρμογή μας στο σύστημα του

Γκούτζος Άρης

Android, πληροφορίες τις οποίες χρειάζεται το σύστημα πριν μπορέσει να εκτελέσει οποιοδήποτε κώδικα της εφαρμογής. Μεταξύ άλλων το AndroidManifest κάνει και τα ακόλουθα:

- Ονομάζει το Java πακέτο της εφαρμογής. Το όνομα του πακέτου λειτουργεί ως μοναδικό αναγνωριστικό ID για την εφαρμογή.
- Περιγράφει τις συνιστώσες της εφαρμογής. Για παράδειγμα, τις activities, services (υπηρεσίες), content providers (παρόχους περιεχομένου) κτλ.
- Ονομάζει τις κλάσεις που εφαρμόζουν καθεμία από τις συνιστώσες και δημοσιεύει τις ικανότητές τους. Για παράδειγμα, ποιές Intents (προθέσεις) μπορούν να χειριστούν.
- Οι δηλώσεις αυτές επιτρέπουν το σύστημα Android να γνωρίζει ποιές είναι οι συνιστώσες και υπό ποιές συνθήκες μπορούν να ενεργοποιούνται.
- Δηλώνει ποιά δικαιώματα η εφαρμογή πρέπει να έχει προκειμένου να γίνει δυνατή η πρόσβαση σε προστατευμένες περιοχές του API και να μπορεί να αλληλεπιδρά με άλλες εφαρμογές.
- Δηλώνει τις προϋποθέσεις που πρέπει να έχει η συσκευή ώστε να μπορεί να λειτουργήσει σε αυτή η εφαρμογή.
- Δηλώνει το ελάχιστο επίπεδο του Android API που απαιτεί η εφαρμογή.

Πολλά από τα παραπάνω μπορούμε να τα δούμε σε ένα τυπικό αρχείο AndroidManifest.xml που αυτόματα το SDK δημιούργησε μόλις φτιάξαμε ένα νέο Android project, και έχει ως εξής:

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

    <uses-sdk
        android:minSdkVersion="8"
        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.example.test.MainActivity"
```

Γκούτζος Άρης

```
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category>
                android:name="android.intent.category.LAUNCHER"
            </category>
        </intent-filter>
    </activity>
</application>
</manifest>
```

Από τον παραπάνω xml κώδικα θα αναλύσουμε τις ετικέτες που χρησιμοποιούνται και είναι αυτές που απαιτούνται να υπάρχουν ώστε να λειτουργήσει η εφαρμογή μας σε μια Android συσκευή. Επίσης, πρέπει να αναφέρουμε ότι σχεδόν όλα τα γνωρίσματα πρέπει να φέρουν μπροστά τους την λέξη android. Τέλος, ενδεικτικά θα αναφέρουμε ποιές άλλες ετικέτες και γνωρίσματα μπορούμε να χρησιμοποιήσουμε.

- Η ετικέτα `<xml/>` : Με αυτήν την ετικέτα καθορίζουμε ότι στο αρχείο αυτό περιέχεται κώδικας σε μορφή xml. Το `version` μας ενημερώνει για την έκδοση xml που χρησιμοποιούμε και είναι η έκδοση 1 και το `encoding` για την κωδικοποίηση του κειμένου είναι το utf-8.
- Η ετικέτα `<manifest>` αποτελεί την αρχική ετικέτα του αρχείου και πρέπει να περιέχει οπωσδήποτε την ετικέτα `<application>`. Τα γνωρίσματα που περιέχει η ετικέτα `<manifest>` του παραπάνω κώδικα είναι τα εξής:
 - **package**: σε αυτό το γνώρισμα δηλώνεται το πακέτο που χρησιμοποιείται. Στο συγκεκριμένο παράδειγμα είναι το πακέτο `"com.example.test"`.
 - **android:versionCode**: Εδώ δηλώνεται η έκδοση του κώδικα μας και μπορεί να είναι οποιαδήποτε ακέραια τιμή θέλουμε. Οι μεγαλύτεροι αριθμοί δηλώνουν και πιο πρόσφατη έκδοση. Αυτό το γνώρισμα χρησιμοποιείται εσωτερικά από το σύστημα και δεν είναι ορατό από τον χρήστη. Στο παράδειγμα μας είναι η έκδοση 1.
 - **android:versionName**: Εδώ δηλώνουμε τον αριθμό έκδοσης που θέλουμε να εμφανίζεται στον χρήστη. Μπορούμε να δώσουμε οποιοδήποτε όνομα θέλουμε.
- Η ετικέτα `<application>` ορίζει την εφαρμογή μας. Συμπεριλαμβάνεται στην ετικέτα `<manifest>` και μπορεί να περιέχει άλλες ετικέτες όπως `<activity>`,

<service>, <provider> κ.α. Τα γνώρισμα που περιέχει η εν λόγω ετικέτα στον παραπάνω κώδικα είναι τα εξής:

- **android:icon:** Με αυτό το γνώρισμα δηλώνουμε το εικονίδιο της εφαρμογής μας που θα εμφανίζεται στην συσκευή του χρήστη. Το εικονίδιο αυτό θα πρέπει να υπάρχει σε έναν από τους φακέλους /res/drawable.
- **android:label:** Εδώ καθορίζουμε το όνομα της εφαρμογής που θα εμφανίζεται στον χρήστη.
- Η ετικέτα <activity> δηλώνει ένα activity το οποίο αποτελεί εν μέρει και το user interface της εφαρμογής. Το σύστημα δεν μπορεί να αναγνωρίσει ένα activity που δεν είναι δηλωμένο στο αρχείο manifest, οπότε δεν πρόκειται και να το εκτελέσει. Αυτή η ετικέτα συμπεριλαμβάνεται στο στοιχείο <application> και μπορεί να περιέχει στοιχεία <intent-filter> και <meta-data>. Μπορούμε να βρούμε τα εξής γνώρισμα:
 - **android:name:** Εδώ δηλώνουμε το όνομα του activity μας. Πριν το όνομα που δηλώνουμε να πρέπει να υπάρχει μια τελεία (.) που στην ουσία αντικαθιστά το όνομα του πακέτου της εφαρμογής (com.example.test).
 - **android:label:** Σε αυτό το γνώρισμα δηλώνουμε το όνομα που θέλουμε να έχει το activity μας και εμφανίζεται στον τίτλο της όταν εκτελείται.
- Η ετικέτα <intent-filter> καθορίζει τους τύπους των intents που ανταποκρίνεται το activity μας και δηλώνει τις δυνατότητες του τελευταίου καθώς και το τι μπορεί ή όχι να χειριστεί. Αυτή η ετικέτα συμπεριλαμβάνεται σε ένα στοιχείο <activity>, <service> ή <receiver> και πρέπει να περιέχει το στοιχείο <action>. Επιπρόσθετα μπορεί να περιέχει τις <category> και <data>.
 - Η ετικέτα **<action>** πρέπει να συμπεριλαμβάνεται στην <intent-filter> αν θέλουμε να ορίσουμε ποια intents θα περνάνε από το φίλτρο. Αν δεν έχουμε ορίσει κανένα <action> τότε το activity μας δεν λαμβάνει και δεν χειρίζεται κανένα intent. Στο παράδειγμα μας περιέχει το γνώρισμα name με την τιμή "*android.intent.action.MAIN*" που δηλώνει πως το activity μας δεν χρειάζεται δεδομένα ώστε να ξεκινήσει. Σαν να είναι η main μιας Java εφαρμογής.

- Η ετικέτα **<category>** περιέχει επιπρόσθετες πληροφορίες για το είδος της συνιστώσας που πρέπει να χειριστεί το intent. Στο παράδειγμα μας έχουμε το γνώρισμα name με την τιμή android.intent.category.LAUNCHER που δηλώνει ότι το activity μας είναι το πρωταρχικό της εφαρμογής μας και είναι αυτό που θα εκτελεστεί πρώτο όταν φορτώσει η εφαρμογή.

Τα activities που μπορούν να αρχικοποιηθούν, δηλαδή να ξεκινούν εφαρμογές και ταυτόχρονα να τις αντιπροσωπεύουν στον application launcher (προωθητή εφαρμογών), περιέχουν τα στοιχεία, <action> και <category> που περιγράψαμε, με αυτές τις τιμές που περιγράψαμε παραπάνω.

- Η ετικέτα <uses-sdk> μας δίνει τη δυνατότητα να εκφράσουμε τη συμβατότητα της εφαρμογής μας με μία ή περισσότερες εκδόσεις της Android πλατφόρμας, δηλώνοντας τον ακέραιο αριθμό που αντιπροσωπεύει το επίπεδο του API. Συμπεριλαμβάνεται στην ετικέτα <manifest> και μπορεί να περιέχει τα γνώρισμα minSdkVersion, targetSdkVersion και maxSdkVersion. Από το παράδειγμα μας βλέπουμε ότι έχει οριστεί το γνώρισμα minSdkVersion με τιμή 8. Αυτό σημαίνει ότι για να τρέξει η εφαρμογή μας σε μια Android συσκευή πρέπει να έχει έκδοση λειτουργικού συστήματος 2.2 και πάνω.

Platform Version	API Level	VERSION_CODE
Android 4.3	18	JELLY_BEAN_MR2
Android 4.2, 4.2.2	17	JELLY_BEAN_MR1
Android 4.1, 4.1.1	16	JELLY_BEAN
Android 4.0.3, 4.0.4	15	ICE_CREAM_SANDWICH_MR1
Android 4.0, 4.0.1, 4.0.2	14	ICE_CREAM_SANDWICH
Android 3.2	13	HONEYCOMB_MR2
Android 3.1.x	12	HONEYCOMB_MR1
Android 3.0.x	11	HONEYCOMB
Android 2.3.4 Android 2.3.3	10	GINGERBREAD_MR1
Android 2.3.2 Android 2.3.1 Android 2.3	9	GINGERBREAD
Android 2.2.x	8	FROYO
Android 2.1.x	7	ECLAIR_MR1
Android 2.0.1	6	ECLAIR_0_1
Android 2.0	5	ECLAIR
Android 1.6	4	DONUT
Android 1.5	3	CUPCAKE
Android 1.1	2	BASE_1_1
Android 1.0	1	BASE

Εικόνα 7 Αντιστοιχία API level και ονομασία έκδοσης του Android

2.6 Γιατί Android;

Το Android είναι μια μοναδική πλατφόρμα η οποία δίνει την ελευθερία στους προγραμματιστές εφαρμογών να εκμεταλλευτούν πλήρως τις δυνατότητες μιας συσκευής. Παρέχει μια μεγάλη γκάμα από βιβλιοθήκες για την ανάπτυξη λογισμικού και διαθεσιμότητα πολλών εργαλείων που αυξάνει τη παραγωγικότητα και φαντασία των προγραμματιστών για την δημιουργία μιας πρωτότυπης εφαρμογής. Για παράδειγμα, είναι θέμα απλών εντολών για έναν πεπειραμένο προγραμματιστή να συνδέσει λειτουργίες κάμερας, GPS,

φωνητικών εντολών και ίντερνετ σε μια εφαρμογή. Το γεγονός ότι ο πυρήνας του είναι opensource μας διαβεβαιώνει επίσης ότι η ανάπτυξη του θα συνεχίσει και θα υπάρχει συνεχής εξέλιξη καθώς πολλοί προγραμματιστές ασχολούνται με τη δημιουργία και συντήρηση του ελεύθερου λογισμικού.

Από την πλευρά του χρήστη, μια συσκευή Android είναι σε μεγάλο βαθμό διαθέσιμη για παραμετροποιήσεις. Στην οθόνη της συσκευής είναι εύκολο κάποιος να προσθέσει ή να αφαιρέσει εφαρμογές και widgets ανάλογα τη συχνότητα που τα χρησιμοποιεί. Αυτό βέβαια δε σημαίνει ότι οι εφαρμογές που δεν βρίσκονται στην οθόνη δεν θα είναι κανονικά εγκατεστημένες. Οι εφαρμογές που κάποιος χρήστης επιθυμεί να εγκαταστήσει μπορούν να ενσωματώνουν λειτουργίες από τις άλλες βασικές εφαρμογές της συσκευής και να ανταλλάσσουν δεδομένα μεταξύ τους. Οι συσκευές με λογισμικό Android έχουν την δυνατότητα να τρέχουν ταυτόχρονα διαφορετικές εφαρμογές στο background. Αυτή η ιδιότητα τους ονομάζεται multitasking και είναι πολύ σημαντικό εργαλείο για τον χρήστη καθώς μπορεί με τη χρήση ενός κουμπιού να δει και να επιλέξει την εφαρμογή που θέλει να εκτελέσει. Η Google εκτός από κατασκευάστρια του λογισμικού παρέχει και σημαντική υποστήριξη στις υπηρεσίες της. Τα Android συνεργάζονται καλύτερα από όλες τις άλλες πλατφόρμες με σημαντικές εφαρμογές όπως το Gmail, το GoogleCalendar, το GoogleDocs, το GoogleMaps, τη μηχανή αναζήτησης της Google και το YouTube. Τέλος, συσκευές διαφορετικών δυνατοτήτων έχουν δημιουργηθεί με σκοπό να συνεργαστούν με αυτό το λογισμικό. Το Android δημιουργήθηκε για να λειτουργεί σε συσκευές χαμηλού κόστους με μειωμένες δυνατότητες αλλά και σε συσκευές υψηλού κόστους όπου η πλατφόρμα αναδεικνύει τις δυνατότητες του software (ταχύτητα εκτέλεσης διεργασιών) αλλά και του hardware (ταχύτητα επεξεργασίας).

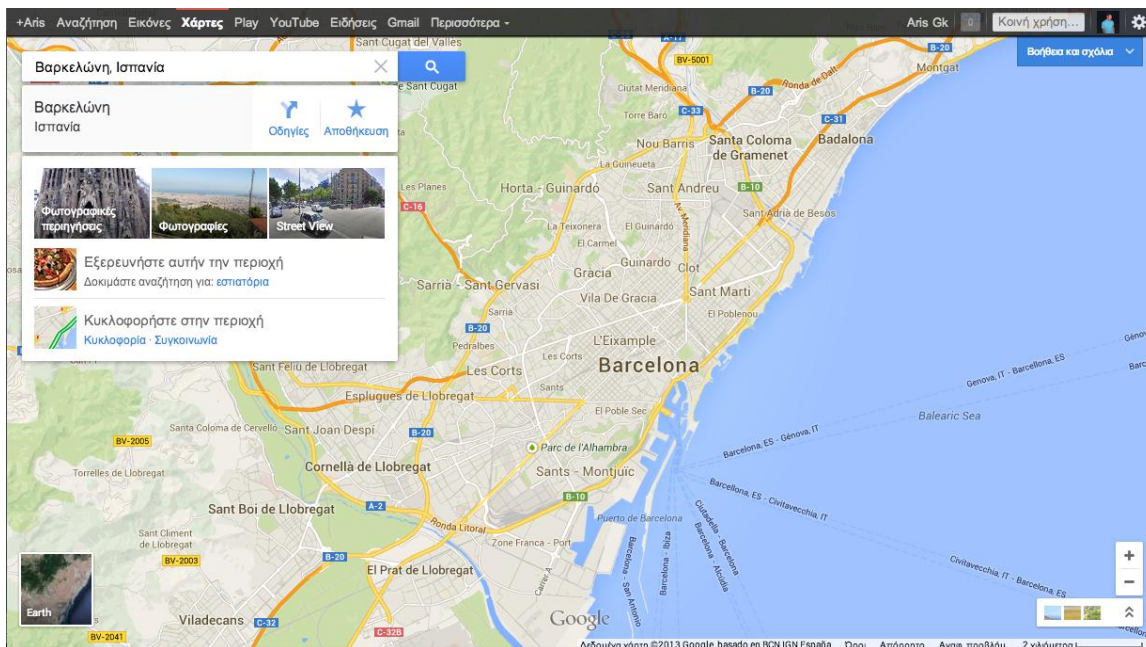
2.7 Επίλογος

Σε αυτό το κεφάλαιο εξηγήσαμε τί είναι το Android, πως εξελίχθηκε ανά τα χρόνια και από ποιά στρώματα αποτελείται. Επίσης είδαμε την χρησιμότητα των εφαρμογών σε συνδυασμό με τα πλεονεκτήματα που παρέχει η πλατφόρμα, τόσο για τους προγραμματιστές όσο και για τους χρήστες.

ΚΕΦΑΛΑΙΟ 3. Google Maps

3.1 Εισαγωγή

Οι χάρτες Google, είναι μια διαδικτυακή εφαρμογή υπηρεσιών χαρτογράφησης. Η τεχνολογία αυτή παρέχεται δωρεάν για προσωπική χρήση από την Google. Ανάμεσα στις υπηρεσίες που μας προσφέρει αυτός ο κολοσσός του τομέα της πληροφορικής, είναι η ιστοσελίδα των Google Maps, η Google Transit και χάρτες ενσωματωμένους σε ιστοσελίδες τρίτων ή εφαρμογές για λειτουργικά συστήματα μέσω των Google Maps API. Παρέχει οδικούς χάρτες, εφαρμογή δρομολόγησης για ταξίδι με τα πόδια, το αυτοκίνητο ή μέσα μαζικής μεταφοράς, ακόμη και ποδήλατο, και εντοπισμό αστικών επιχειρήσεων για πολλές χώρες σε όλο τον κόσμο. Η τεχνολογία των χαρτών Google χρησιμοποιεί μια στενή παραλλαγή της προβολής Mercator (ένας κυλινδρικής προβολής χάρτης του κόσμου), με αποτέλεσμα να μην μπορεί να δείξει περιοχές γύρω από τους πόλους. Ένα σχετικό προϊόν είναι το Google Earth, ένα αυτόνομο πρόγραμμα το οποίο προσφέρει περισσότερα χαρακτηριστικά ως προς την παγκόσμια προβολή, συμπεριλαμβανομένων των πολικών περιοχών.



Εικόνα 8 Google Maps μέσα από browser

3.2 Υλοποίηση

Η τεχνολογία αυτή χρησιμοποιεί εκτενώς την γλώσσα Javascript. Όταν ο χρήστης σύρει τον χάρτη τότε φορτώνονται από τον εξυπηρετητή τα τετράγωνα του πλέγματος, τα οποία ακολούθως εισάγονται στην σελίδα. Όταν ο χρήστης ψάχνει για μια επιχείρηση τα αποτελέσματα φορτώνονται στο παρασκήνιο για εισαγωγή στο πλαίσιο πλαίσιο και τον χάρτη, με αποτέλεσμα να αποφεύγεται η επαναφόρτωση της σελίδας. Οι τοποθεσίες καθορίζονται δυναμικά, με την τοποθέτηση μιας κουκκίδας (pin) πάνω από τις εικόνες του χάρτη.

Επιπρόσθετα, η σελίδα χρησιμοποιεί JSON (τεχνολογία που εισάγαμε και εμείς στην εφαρμογή μας και θα αναλύσουμε παρακάτω) για μεταφορά δεδομένων, αντί XML, για σκοπούς απόδοσης. Το γεγονός ότι οι χάρτες Google είναι τεχνολογία γραμμένη σχεδόν εξολοκλήρου σε Javascript και XML, επιτρέπει σε τελικούς χρήστες να εκτελούν κομμάτια κώδικα στους χάρτες από τον υπολογιστή τους (client-side scripts) και από τη μεριά του διακομιστή (server-side scripts), τα οποία δίνουν την δυνατότητα σε ένα χρήστη ή μια ιστοσελίδα να εισάγουν επεκταμένες ή τροποποιημένες λειτουργίες στην διεπαφή των χαρτών Google.

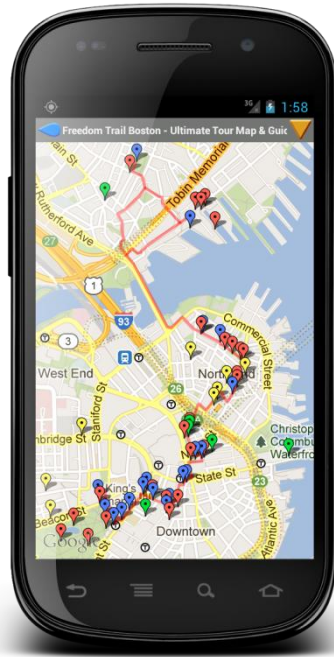
3.3 Google Maps API (Application Programming Interface)

Τον Ιούνιο του 2005, η Google διέθεσε το Google Maps API, με σκοπό να επιτρέψει στους προγραμματιστές να ενσωματώσουν τους Χάρτες Google στις ιστοσελίδες τους. Το API αυτό είναι δωρεάν για εμπορική χρήση, με την προϋπόθεση ότι η ιστοσελίδα για την οποία θα χρησιμοποιηθεί, δεν χρεώνει τους χρήστες της για πρόσβαση σε αυτήν. Ιστοσελίδες που χρεώνουν και θέλουν να χρησιμοποιήσουν το Google Maps API, μπορούν να αγοράσουν το Google Maps Premier. Με την χρήση της διεπαφής, μπορεί κάποιος να ενσωματώσει την ιστοσελίδα Χαρτών Google σε εξωτερική ιστοσελίδα με την δυνατότητα τοποθέτησης δικών του δεδομένων πάνω από τους χάρτες.

Αν και αρχικά το Google Maps API ήταν ένα API για Javascript, μετέπειτα επεκτάθηκε ώστε να συμπεριλάβει ένα API για Adobe Flash και μια υπηρεσία για ανάκτηση στατικών εικόνων χαρτών. Επίσης συμπεριέλαβε δικτυακές υπηρεσίες για την εκτέλεση γεωκωδικοποιήσεων και δημιουργία κατευθύνσεων οδήγησης.

3.4 Χάρτες για smartphones και εξέλιξη τους

Οι Χάρτες Google για κινητά είναι μια εφαρμογή σε Java, η οποία εκδόθηκε από την Google το 2006, με σκοπό να τρέχει σε οποιοδήποτε βασισμένο σε Java κινητό ή φορητή συσκευή. Πολλά από τα χαρακτηριστικά των Χαρτών Google υπάρχουν και σε αυτή την εφαρμογή.



Εικόνα 9 Google Maps για Smartphone

Στις 28 Νοεμβρίου του 2008 κυκλοφόρησε και η δεύτερη έκδοση της εφαρμογής αυτής, οι Χάρτες Google για Κινητά 2.0. Σε αυτήν την καινούργια έκδοση, η Google υλοποίησε μια υπηρεσία παρόμοια με το παγκόσμιο σύστημα εντοπισμού θέσης (GPS-like), η οποία όμως δεν χρειαζόταν δέκτη GPS. Η λειτουργία 'η τοποθεσία μου' (my location), λειτουργεί με χρήση της τοποθεσίας GPS του κινητού, αν αυτή είναι διαθέσιμη. Αυτό γίνεται με τον εξής τρόπο:

Το λογισμικό ψάχνει για το πιο κοντινό ασύρματο δίκτυο ή κύτταρο/κυψέλη. Ακολουθώς ψάχνει για την τοποθεσία του κυττάρου, χρησιμοποιώντας μια βάση δεδομένων με γνωστά ασύρματα δίκτυα και κύτταρα. Ο προσδιορισμός θέσης του κυττάρου γίνεται με μια τριγωνική μέθοδο όπου μελετούνται διαφορετικές εντάσεις του σήματος από διαφορετικούς πομπούς που βρίσκονται στους Σταθμούς Κυττάρων. Ακολουθώς χρησιμοποιώντας την τοποθεσία η οποία

ανακτάται από μια διαδικτυακή βάση δεδομένων, διορθώνεται 'η τοποθεσία μου', καθορίζοντας την τρέχουσα θέση του χρήστη. Η μέθοδος εντοπισμού ασύρματου δικτύου, πραγματοποιείται βρίσκοντας τις πιο κοντινές δυναμικές ζώνες WiFi (WiFi hotspots). Ακολούθως η τοποθεσία αυτών των ζωνών που ανακτάται από αντίστοιχη βάση δεδομένων χρησιμοποιείται για την εξακρίβωση της θέσης του χρήστη. Η εφαρμογή κάνει χρήση υπηρεσιών βασισμένες στο GPS, υπηρεσιών βασισμένες στο WiFi και στο WLAN, και υπηρεσιών βασισμένων στους πομπούς των κυττάρων. Στον χάρτη φαίνονται οι δρόμοι και η εκτιμώμενη θέση του κινητού τηλεφώνου, η οποία δεν είναι απόλυτη καθώς περιβάλλεται από ένα κύκλο που απεικονίζει το εύρος χώρου στον οποίο μπορεί να βρίσκεται το κινητό. Το εκτιμώμενο εύρος χώρου, υπολογίζεται με βάση την ισχύ του σήματος του τηλεφώνου που υποδεικνύει πόσα κοντά αυτό είναι στον Σταθμό του κυττάρου, άρα και στον πομπό. Η υπηρεσία αυτή είναι διαθέσιμη σε πολλές πλατφόρμες όπως για παράδειγμα τις ακόλουθες:

- Android (η σχέση του Android με την υπηρεσία αυτή περιγράφεται παρακάτω).
- iOS (iPhone/iPod Touch/iPad)
- Windows Mobile
- Nokia/Symbian (σε συγκεκριμένα μόνο μοντέλα)
- Symbian OS (UIQ v3)
- BlackBerry
- κ.α

Στις 19 Μαρτίου 2009 το Street View διατέθηκε για το Ηνωμένο Βασίλειο και την Ολλανδία, και αργότερα για τις περισσότερες χώρες της Ευρώπης. Η συγκεκριμένη υπηρεσία επιτρέπει στους χρήστες να δουν πραγματικές φωτογραφίες των δρόμων μιας πόλης, τραβηγμένες από αυτοκίνητο της Google ειδικά διαμορφωμένο με κάμερες που 'σκανάρει' όλους τους δρόμους τραβώντας φωτογραφίες.

Τα τελευταία χρόνια, ανά σύντομα χρονικά διαστήματα, οι χάρτες βελτιώνονται με γοργούς ρυθμούς αφού προστίθενται συνεχώς νέες υπηρεσίες και τεχνολογίες

όπως χαρτογράφηση ποταμών και λιμνών, έλεγχος κυκλοφορίας ενός δρόμου σε πραγματικό χρόνο κ.α.

3.5 Google Maps και Android

Το Android προσφέρει στους προγραμματιστές την δυνατότητα να ενσωματώσουν κάποιο χάρτη στην εφαρμογή τους. Αυτό μπορεί να γίνει με δύο τρόπους. Σύμφωνα με τον πρώτο τρόπο, που είναι και ο πιο απλός, ο προγραμματιστής μπορεί να ενσωματώσει στην οθόνη ένα στοιχείο το οποίο καλείται 'WebView'. Στο 'WebView', ο προγραμματιστής μπορεί να απεικονίσει οποιαδήποτε διαδικτυακή εφαρμογή με αντίστοιχο τρόπο που αυτή απεικονίζεται σε έναν φυλλομετρητή και κατ' επέκταση τους Χάρτες Google.

Η Google όμως θέλοντας να δώσει περισσότερη ελευθερία στους προγραμματιστές, δημιούργησε ένα άλλο στοιχείο διεπιφάνειας χρήστη, το 'MapView' γεγονός που μας οδηγεί στην δεύτερη εναλλακτική. Το MapView σε συνδυασμό με την κλάση MapActivity και τις διεπαφές mapping APIs, είναι ένα ισχυρό εργαλείο στα χέρια οποιουδήποτε προγραμματιστή που θέλει να δημιουργήσει καινοτόμες και πρωτοποριακές εφαρμογές με την χρήση των χαρτών. Με αυτό τον τρόπο ο κατασκευαστής μπορεί να χρησιμοποιήσει υπάρχων χάρτες και να τους τροποποιήσει όπως αυτός θέλει, με την δυνατότητα να χειριστεί τις αλληλεπιδράσεις του χρήστη με τον χάρτη, να τοποθετήσει δικά του δεδομένα πάνω σε αυτόν κ.τ.λ.

Το πακέτο για την χρησιμοποίηση των mapping APIs δεν ανήκει στο πλαίσιο του Android, αλλά σε αυτό της Google (com.google.android.maps), γεγονός που πρέπει να δηλωθεί ξεχωριστά. Επιπρόσθετα για την χρησιμοποίηση των χαρτών η Google απαιτεί την απόκτηση από τον εκάστοτε προγραμματιστή ενός map-api key, κλειδί το οποίο προσφέρει η ίδια (<http://code.google.com/android/maps-api-signup.html>). Για την απόκτηση του κλειδιού ο ενδιαφερόμενος πρέπει να δώσει το αποτύπωμα SHA-1 του ψηφιακού πιστοποιητικού το οποίο θα χρησιμοποιηθεί για την υπογραφή της εφαρμογής και ο τρόπος απόκτησης του αναλύεται στο επόμενο κεφάλαιο.

3.6 Επίλογος

Η υπηρεσία Google Maps προσφέρεται δωρεάν τόσο στους τελικούς χρήστες όσο και στους προγραμματιστές που θέλουν να ενσωματώσουν αυτήν την υπηρεσία στις εφαρμογές τους. Προσφέρει ένα πλήθος δυνατοτήτων οι οποίες αυξάνονται συνεχώς με τον πέρασμα του χρόνου καθώς γίνεται αδιάκοπη προσπάθεια ενσωμάτωσης όλων το πιθανών λειτουργιών που μπορεί κάποιος να φανταστεί σε ένα διαδραστικό χάρτη. Οι χρήστες των Google Maps έχουν την δυνατότητα με ένα κλικ να δουν κάθε γωνιά της γης, να περιηγηθούν σε όλα τα σημεία ενδιαφέροντος, να πλοηγηθούν σε πραγματικό χρόνο και πολλές άλλες δυνατότητες μέσα από την συσκευή τους. Επίσης, οποιοσδήποτε θέλει να χρησιμοποιήσει τους χάρτες για επαγγελματική χρήση του δίνεται το δωρεάν προνόμιο να τροποποιήσει τους χάρτες σύμφωνα με τις ανάγκες του.

ΚΕΦΑΛΑΙΟ 4. Απαιτήσεις και ανάλυση της εφαρμογής

4.1 Εισαγωγή

Το CityExplorer είναι μια εφαρμογή για το λειτουργικό σύστημα Android και αναπτύχθηκε στα πλαίσια της σχολής ως πτυχιακή εργασία. Χρησιμοποιήθηκαν διάφορες σύγχρονες τεχνολογίες ανάπτυξης λογισμικού και γνώσεις που αποκτήθηκαν από διάφορα μαθήματα της σχολής.

Υπάρχουν πολλές παρόμοιες εφαρμογές στο εμπόριο, δωρεάν και επί πληρωμή όπως είδαμε στο 1^ο κεφάλαιο, που δίνουν την δυνατότητα στον χρήστη να χρησιμοποιεί την φορητή συσκευή του για να πραγματοποιήσει κάποιες αποστολές και να δηλώσει παρουσία σε κάποιο σημείο της πόλης. Καμία όμως από αυτές δεν δίνει κάποια ουσιαστική επιβράβευση για αυτές τις ενέργειες του χρήστη, όπως επίσης και η διαχείριση των σημείων αυτών γίνεται αποκλειστικά από τους χρήστες και όχι από τους ιδιοκτήτες ή υπευθύνους των σημείων αυτών με αποτέλεσμα να υπάρχει μια αμφιβολία αξιοπιστίας ως προς αν τα σημεία αυτά είναι υπαρκτά και όχι κάποιο αποτέλεσμα την φαντασίας ενός χρήστη.

Και έτσι γεννήθηκε η ιδέα να δημιουργήσουμε μια εφαρμογή που, προσθέτοντας αυτά τα 2 στοιχεία που λείπουν από τις υπόλοιπες και διατηρώντας κάποια άλλα, να παρέχει μια ολοκληρωμένη λύση στους χρήστες ώστε υπάρχει κάποια επιβράβευση στην προσπάθειά τους και να μην χρησιμοποιείται μόνο ως ενασχόληση αναψυχής.

Παρακάτω θα δούμε τα εργαλεία που χρησιμοποιήθηκαν για να φέρουν εις πέρας το τελικό αποτέλεσμα, πως αυτά συνεργάζονται μεταξύ τους για τη καλύτερη λειτουργία της εφαρμογής και τα κομμάτια που 'σπάσαμε' την εφαρμογή για την ευκολότερη και ουσιαστική της ανάπτυξη χρησιμοποιώντας εν τέλει ανεξάρτητες διαδικασίες ανάπτυξης τους, ώστε να υπάρχουν μηδαμινές ή ελάχιστες επιρροές σε άλλα κομμάτια κώδικα σε περίπτωση εμφάνισης σφάλματος σε κάποια από αυτές. Η καταγραφή των απαιτήσεων βοηθά όχι μόνο στην ταχύτερη και ευκολότερη ανάπτυξη της εφαρμογής αλλά και στην γρήγορη αποσφαλμάτωσή της.

4.2 Εργαλεία και τεχνολογίες που χρησιμοποιήθηκαν

Για τη δημιουργία της εφαρμογής έγινε η εγκατάσταση του προγράμματος Eclipse Indigo και του Android SDK. Το SDK δίνει τη δυνατότητα στους προγραμματιστές να δημιουργήσουν ένα εξομοιωτή (emulator) ενός Android συστήματος μέσω του Android manager. Ένας emulator έχει κάποια κοινά χαρακτηριστικά με ένα simulator (προσομοιωτής) αλλά εκείνο που το κάνει πραγματικά να διαφέρει είναι ότι κατά το compilation του προγράμματος δημιουργείται μία πραγματική μηχανή με όλες τις λειτουργίες ενός τηλεφώνου.

Η μηχανή αυτή δίνει πολλές επιλογές στον προγραμματιστή καθώς μπορεί να περιηγηθεί στο διαδίκτυο μέσω αυτού του εικονικού τηλεφώνου, να κάνει εγκατάσταση εφαρμογών από το GooglePlay ή να δει πως λειτουργεί η εφαρμογή που δημιουργεί. Όλα αυτά γίνονται στην οθόνη του υπολογιστή του προγραμματιστή. Μια εικονική ‘συσκευή’ που με το κλείσιμο του προγράμματος εγγραφής κώδικα συνεχίζει να υπάρχει. Το μόνο μειονέκτημα της είναι ότι δε μπορεί να γίνει πραγματική δοκιμή για λήψη της τοποθεσίας μέσω GPS ή WiFi επομένως για την εφαρμογή μας πραγματοποιήθηκε η σύνδεση του Eclipse με μια Android συσκευή, στην οποία γίνεται η εγκατάσταση της εφαρμογής και στη συνέχεια μπορεί να λειτουργήσει σαν μια κανονική εφαρμογή που έχουμε κατεβάσει από το Google Play . Για να γίνει αυτή η σύνδεση πρέπει αρχικά να εγκατασταθούν οι κατάλληλοι οδηγοί λογισμικού (drivers) για να αναγνωριστεί η συσκευή από τα Windows και να επιτραπεί στο λογισμικό του Android μέσω ρυθμίσεων debugging η εμφάνιση της εφαρμογής στην οθόνη του κινητού.



Εικόνα 10 Λογότυπα γλωσσών προγραμματισμού που χρησιμοποιήθηκαν

Με την ενεργοποίηση της εκκίνησης της εφαρμογής στο Eclipse εμφανίζεται ένα παράθυρο για να επιλέξει ο προγραμματιστής αν θέλει να γίνει εγκατάσταση της εφαρμογής του στον emulator ή στη συσκευή του. Το Eclipse δίνει επίσης τη δυνατότητα πολλών εργαλείων στον χρήστη, στο προγραμματιστικό κομμάτι μέχρι και την εμφάνιση της εφαρμογής στη οθόνη του κινητού.

Για τη λειτουργικότητα της εφαρμογής χρησιμοποιήθηκε η γλώσσα προγραμματισμού Java, όπου το Eclipse διαθέτει έναν κειμενογράφο (editor) με διάφορες αυτόματες ιδιότητες φιλικές στον προγραμματιστή. Για την εμφάνιση της εφαρμογής υπάρχει γραφικό και προγραμματιστικό περιβάλλον στη γλώσσα σήμανσης κειμένου XML. Ο προγραμματιστής μπορεί να κάνει drag and drop διάφορα στοιχεία από τη παλέτα αριστερά και να τα τοποθετήσει με διαφορετική βαρύτητα στην επιφάνεια.

Στο προγραμματιστικό κομμάτι δίνονται κάποιες πληροφορίες αναγνώρισης του κάθε στοιχείου, πληροφορίες στοίχισης κ.α.. με σκοπό να τα χειριστεί ο χρήστης στο πρόγραμμα του με τη χρήση εντολών της Java. Με λίγα λόγια τα στοιχεία της XML και ο κώδικας της Java συνεργάζονται απολύτως.

Στη μεριά του σέρβερ έπρεπε να γίνει η συγγραφή πολλαπλών PHP script και να αποθηκευθούν online. Για τη ανάπτυξη αυτών των προγραμμάτων έγινε χρήση του Komodo Edit 8 για Mac OS. Ιδιαιτερότητες υπάρχουν και σε αυτόν τον κειμενογράφο καθώς τα εργαλεία που χρησιμοποιούνται από τους προγραμματιστές του είναι διαφορετικά από το Eclipse. Τα scripts δεν έχουν γραφικό περιβάλλον καθώς το μόνο που επιστρέφουν συνήθως είναι αποτελέσματα από τη βάση δεδομένων. Ο έλεγχος τους γίνεται μέσω ενός προγράμματος περιήγησης και χρήση απλής HTML. Τα προγράμματα PHP περιέχουν κώδικα σε γλώσσα Javascript, κάτι που υποστηρίζει αυτός ο κειμενογράφος.

Για να γίνει αποθήκευση των script στον σέρβερ έγινε χρήση του Cyberduck και του ενσωματωμένου Terminal που διαθέτει το λειτουργικό σύστημα Mac OS. Το γραφικό περιβάλλον του Cyberduck είναι ιδιαίτερα απλό, δίνοντας τη δυνατότητα στον χρήστη να δει ένα κατάλογο αρχείων στον σκληρό δίσκο του υπολογιστή

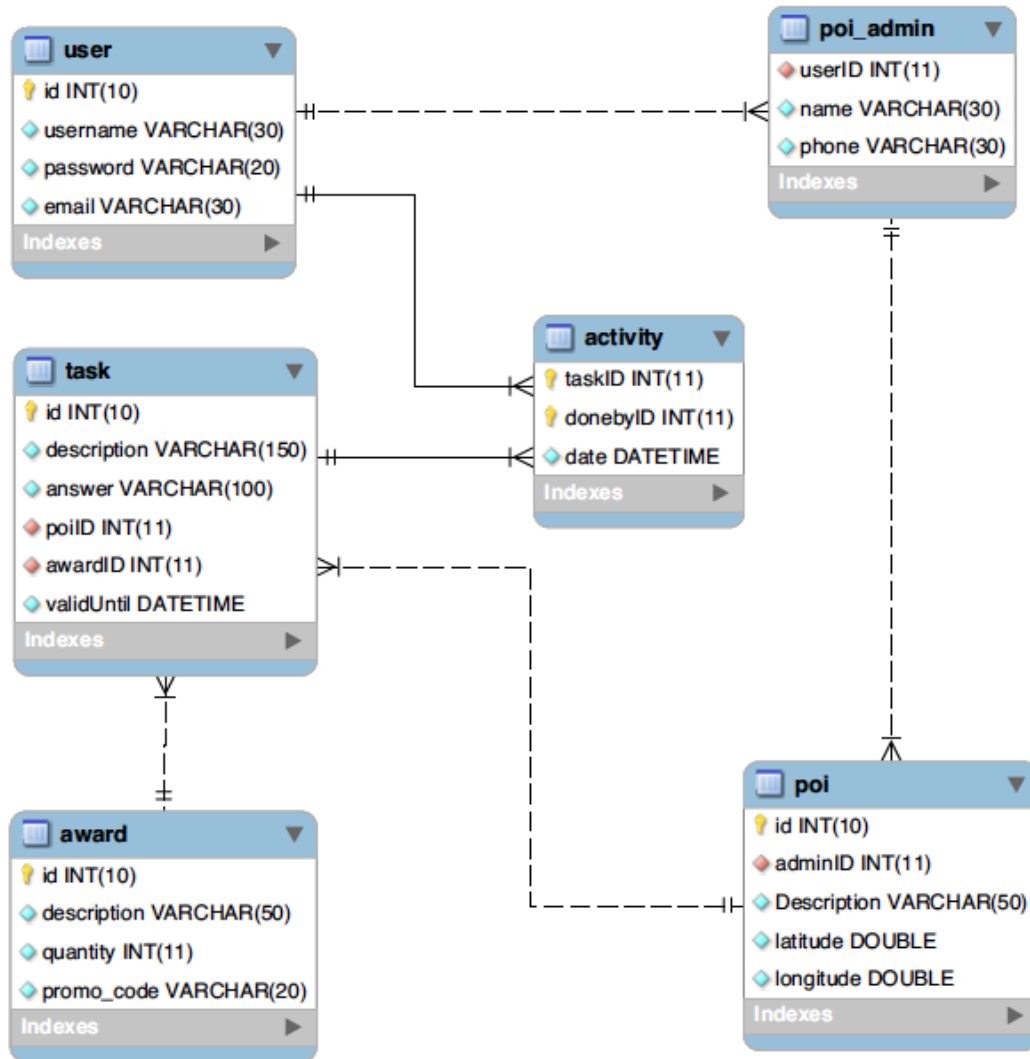
του και ένα κατάλογο αρχείων αποθηκευμένα στον σέρβερ. Με την επιλογή συγκεκριμένων PHP αρχείων εύκολα γίνεται η μεταφορά και αποθήκευση τους στους καταλόγους του σέρβερ. Από τη στιγμή που είναι αποθηκευμένα εκεί, γίνεται εύκολα η κλήση τους τόσο από ένα πρόγραμμα περιήγησης, είτε από το Terminal χρησιμοποιώντας εντολές UNIX είτε μέσω κάποιας διαδικτυακής εφαρμογής που ζητάει κάποια δεδομένα.

Για τη δημιουργία της βάσης έγινε χρήση του προγράμματος phpMyAdmin που φιλοξενείται στον erodio και του εργαλείου MySQLWorkbench για Mac OS. Με την χρήση του phpMyAdmin δημιουργήθηκαν όλοι οι πίνακες με όλα τα απαραίτητα πεδία και κάποια δεδομένα για δοκιμαστική χρήση. Η χρήση της γλώσσας βάσεων δεδομένων MySQL είναι απαραίτητη για έργα μικρού βεληνεκούς. Η εύκολη σύνταξη της βοήθησε στη δημιουργία των κατάλληλων ερωτημάτων για την αναζήτηση δεδομένων από την εφαρμογή ή την εισαγωγή δεδομένων από την εφαρμογή προς την βάση, ερωτήματα τα οποία δοκιμάστηκαν για την αποτελεσματικότητα τους σε αυτό το εργαλείο. Το MySQLWorkbench χρησιμοποιήθηκε κυρίως για την ευκολία που προσφέρει στην εμφάνιση των δεδομένων της βάσης, αφού δεν απαιτεί την συνεχή σύνδεση στην βάση εισάγοντας κάθε φορά το όνομα χρήστη και τον κωδικό πρόσβασης.

Αυτά τα 6 προγράμματα (Eclipse, Komodo Edit 8. phpMyAdmin, MySQLWorkbench, Terminal και Cyberduck) χρησιμοποιήθηκαν κυρίως σε συνδυασμό καθώς οι γλώσσες προγραμματισμού συνεργάζονται σε όλα τα προγράμματα που χρειάστηκαν για την υλοποίηση αυτής της πτυχιακής

4.3 Σχεδιασμός βάσης δεδομένων

Η βάση δεδομένων που χρησιμοποιήθηκε για την αποθήκευση των δεδομένων της εφαρμογής μας, αναπτύχθηκε με το εργαλείο phpMyAdmin το οποίο εγκαταστάθηκε στον διακομιστή του τμήματος Πληροφορικής, Ερωδιό. Η σύνδεση στην βάση για την διαχείριση της πραγματοποιήθηκε είτε απευθείας μέσω του εργαλείου phpMyAdmin σε κάποιον φυλλομετρητή (browser) είτε μέσω του εργαλείου MySQLWorkbench.



Εικόνα 11 Σχεδιάγραμμα της βάσης δεδομένων

Στο παραπάνω σχεδιάγραμμα φαίνονται οι πίνακες της βάσης δεδομένων, τα στοιχεία που περιέχουν, ο τύπος δεδομένων που αυτά υποστηρίζουν, τα κυρίως κλειδιά του κάθε πίνακα καθώς και οι μεταξύ τους συσχετίσεις. Παρακάτω θα αναλύσουμε όλους τους πίνακες, τι δεδομένα περιέχει ο καθένας τους και την χρησιμότητα αυτών.

Στον πίνακα ‘*user*’ γίνεται η αποθήκευση των εγγεγραμμένων χρηστών με τα αντίστοιχα στοιχεία τους, το όνομα χρήστη (username), την ηλεκτρονική τους διεύθυνση (email) και των κωδικό πρόσβασης (password). Το κύριο κλειδί σε αυτόν τον πίνακα είναι το id το οποίο αντιστοιχεί κάθε χρήστη με έναν μοναδικό ακέραιο αριθμό.

Στον πίνακα *'admin'* αποθηκεύονται επιπλέον στοιχεία, όπως το όνομα (name) και το τηλέφωνο (phone), για εκείνους τους χρήστες που έχουν την δυνατότητα να προσθέτουν και να διαχειρίζονται τα σημεία ενδιαφέροντος και τις αποστολές. Ο συγκεκριμένος πίνακας δεν χρησιμοποιείται στην εφαρμογή μας και δημιουργήθηκε για την διευκόλυνση σε περίπτωση περαιτέρω ανάπτυξης της εφαρμογής.

Ο πίνακας *'poi'* περιέχει τις πληροφορίες για κάθε σημείο ενδιαφέροντος. Αυτές περιλαμβάνουν την περιγραφή (description), ποιός δημιούργησε αυτό το σημείο (adminID), την τοποθεσία του σε γεωγραφικό μήκος και πλάτος (Latitude , Longitude) καθώς και το κύριο κλειδί του πίνακα (id).

Ο πίνακας *'award'* περιλαμβάνει τα βραβεία που θα λαμβάνουν οι χρήστες σε περίπτωση ολοκλήρωσης μιας αποστολής. Τα στοιχεία του συγκεκριμένου πίνακα αποτελούν: η περιγραφή του βραβείου (description), η ποσότητα του βραβείου που προσφέρεται στον χρήστη (quantity), ο ειδικός κωδικός του βραβείου (promo code), ο οποίος χρησιμοποιείται για την παραγωγή του ειδικού QRcode και την εξαργύρωση του βραβείου και φυσικά το κύριο κλειδί του πίνακα (id).

Στον πίνακα *'task'* αποθηκεύονται όλες οι αποστολές που είναι διαθέσιμες στους χρήστες και περιλαμβάνει τα εξής στοιχεία: την περιγραφή της διαθέσιμης αποστολής (description), την σωστή απάντηση για να ολοκληρωθεί η αποστολή (answer), τον κωδικό της τοποθεσίας στην οποία προσφέρεται η αποστολή (poiID) και συσχετίζεται με το κύριο κλειδί του πίνακα που περιέχει τα σημεία ενδιαφέροντος (*poi* → id), το κωδικό του βραβείου που κερδίζει ο χρήστης ολοκληρώνοντας την αποστολή (awardID) και συσχετίζεται με το κύριο κλειδί του πίνακα που περιέχει τα βραβεία (*award* → id), την ημερομηνία λήξης της αποστολής (validUntil) μέχρι την οποία θα είναι ενεργή και εμφανίσιμη η αποστολή και όπως οι περισσότεροι πίνακες το κύριο κλειδί του (id).

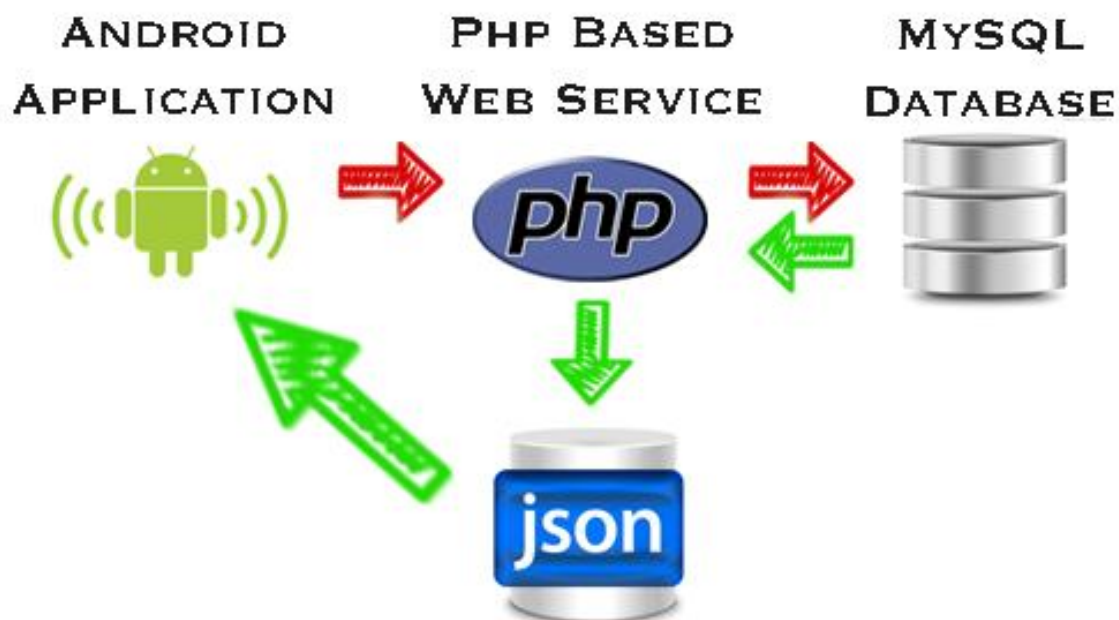
Τέλος, ο πίνακας *'activity'* καταγράφει την δραστηριότητα των χρηστών αποθηκεύοντας την ημερομηνία ολοκλήρωσης της αποστολής (date), ποιός χρήστης την ολοκλήρωσε (donebyID) που συσχετίζεται με το κύριο κλειδί του πίνακα των χρηστών (*user* → id) και φυσικά ποια αποστολή είναι αυτή (taskID) που σχετίζεται με το κύριο κλειδί του πίνακα που περιέχει τις αποστολές

(*task*→*id*). Τα 2 τελευταία στοιχεία αποτελούν και το κύριο κλειδί του πίνακα (*taskID*, *donebyID*).

Οι παραπάνω πίνακες δημιουργήθηκαν ύστερα από μελέτη των απαιτήσεων της εφαρμογής και ο καθένας από αυτούς ή συνδυασμός αυτών χρησιμεύουν σε διαφορετικές λειτουργίες της, όπως θα αναλύσουμε παρακάτω.

4.4 Επικοινωνία με τη βάση δεδομένων

Η επικοινωνία της εφαρμογής με την βάση δεδομένων είναι αμφίδρομη και πραγματοποιείται με την βοήθεια και τον συνδυασμό των γλωσσών προγραμματισμού Java, PHP και MySQL.



Εικόνα 12 Επικοινωνία Android εφαρμογής με βάση δεδομένων

Η επικοινωνία μιας εφαρμογής με μια online βάση δεδομένων, όπως φαίνεται στην παραπάνω εικόνα, χωρίζεται σε δυο μέρη, η αίτηση δεδομένων από την εφαρμογή στην βάση (κόκκινη ροή) και η απάντηση για το αίτημα από την βάση προς την εφαρμογή (πράσινη ροή).

Η εφαρμογή για αποστολή αιτήματος στην βάση ή για επεξεργασία των δεδομένων της βάσης κάνει χρήση της μεθόδου POST, η οποία υποστηρίζεται από το πρωτόκολλο HTTP και είναι σχεδιασμένη να αιτείται ότι ο απομακρυσμένος διακομιστής δέχεται τα δεδομένα, που εμπεριέχονται στο

αίτημα, για αποθήκευση. Σε περίπτωση σφάλματος επιστρέφει το αντίστοιχο μήνυμα και τον κωδικό σφάλματος για τον ευκολότερο εντοπισμό και την διόρθωση του. Τα δεδομένα που αποστέλλει η εφαρμογή επεξεργάζονται από PHP Scripts που στην περίπτωση μας βρίσκονται αποθηκευμένα στον Ερωδιό, σε αποθηκευτικό χώρο που δόθηκε για την ανάπτυξη της συγκεκριμένης εργασίας. Η ηλεκτρονική διεύθυνση των αρχείων PHP ορίζεται στον κώδικα της εφαρμογής, ώστε η εφαρμογή να "γνωρίζει" που να απευθυνθεί για τα το αίτημα της. Τα scripts αυτά είναι διαμορφωμένα να επεξεργάζονται τα δεδομένα που λαμβάνουν από την εφαρμογή και με κατάλληλες μεθόδους επικοινωνούν με την βάση δεδομένων εκτελώντας SQL εντολές.

Για να γίνει επιστροφή δεδομένων στην εφαρμογή, η απάντηση που λαμβάνει το PHP script ύστερα από την εκτέλεση της SQL εντολής κωδικοποιείται σε μορφή JSON. Το JSON είναι ένα ανοιχτό στάνταρντ δομής δεδομένων σχεδιασμένο για ανταλλαγή πληροφοριών μεταξύ των γλωσσών προγραμματισμού σε φιλική προς τον προγραμματιστή μορφή. Η εφαρμογή αποθηκεύει τις δομημένες σε JSON πληροφορίες σε μεταβλητή και χρησιμοποιώντας τις αντίστοιχες ετικέτες που υπάρχουν για να προσδιορίσουν τον τύπο των δεδομένων που περιέχει η μεταβλητή, τα επεξεργάζεται κατάλληλα και τα παρουσιάζει στην οθόνη της συσκευής.

```

{
  "moveString" : "1. Qxf6+ Nxf6 2. Bxf6# 1-0",
  "event" : "White to move.",
  "site" : "?",
  "date" : "1998.??.??",
  "round" : "?",
  "white" : "1001 Brilliant Ways",
  "black" : "to Checkmate",
  "result" : "1-0",
  "annotator" : "Magne, Alf",
  "setup" : "1",
  "fen" : "2r1nr1k/pp1q1p1p/3bpp2/5P2/1P1Q4/P3P3/1B3P1P/R3K1R1 w Q - 0 1",
  "plycount" : "3",
  "eventdate" : "1998.??.??",
  "moves" : {
    1 : {
      "white" : {
        "move" : "Qxf6+"
      }
      ,
      "black" : {
        "move" : "Nxf6"
      }
    }
    ,
    2 : {
      "white" : {
        "move" : "Bxf6#"
      }
    }
  }
}

```

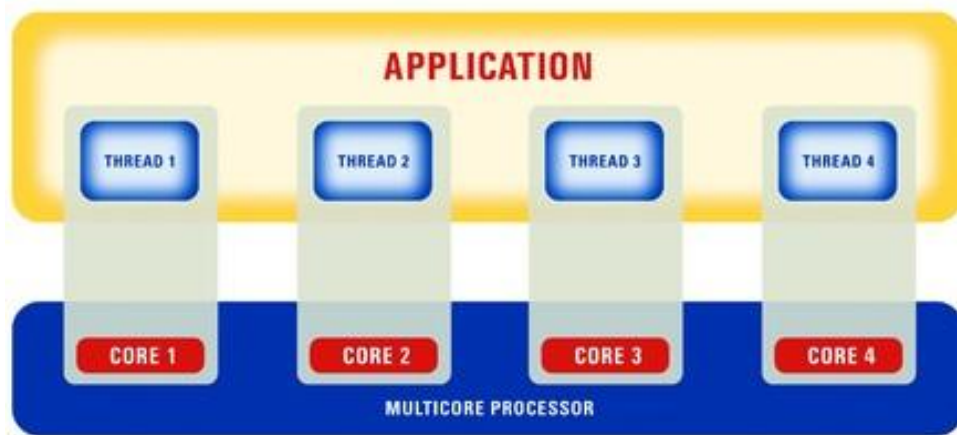
Εικόνα 13 Δεδομένα κωδικοποιημένα σε JSON

4.5 Καταγραφή απαιτήσεων και σχεδιασμός της εφαρμογής

Για να φτάσουμε στο τελικό αποτέλεσμα της εφαρμογής προϋπήρξαν αρκετά στάδια σχεδιασμού. Για να δημιουργήσουμε μια λειτουργική για τον χρήστη εφαρμογή έπρεπε να γίνει καταγραφή των βασικών απαιτήσεων. Έτσι, για να φτιάξουμε την εφαρμογή όσο πιο απλή γίνεται πραγματοποιείται διαχωρισμός της σε 6 κομμάτια που στην συνέχεια συνδυάστηκαν για να καταλήξουμε στο τελικό αποτέλεσμα και στην λειτουργία της εφαρμογής. Ο κεντρικός στόχος είναι η γρήγορη χρήση της εφαρμογής με μικρό αριθμό ενεργειών και ελάχιστη τοποθέτηση δεδομένων από τον χρήστη.

Σε αυτό το σημείο είναι αναγκαίο να επισημάνουμε ότι όλες οι ενέργειες της εφαρμογής που απαιτούν εκτέλεση ερωτημάτων στην βάση δεδομένων ή ανάκτηση πληροφοριών από ιστοσελίδες, όπως η εικόνα του γραμμοκώδικα QRcode, εκτελούνται χρησιμοποιώντας ασύγχρονες εργασίες (Asynchronous tasks). Οι εργασίες αυτές μας επιτρέπουν να τμηματοποιήσουμε τον κώδικα μας

σε νήματα (threads) που εκτελούνται αυτόνομα και μας επιστρέφουν τα αποτελέσματα μετά το πέρας της εκτέλεσής τους.

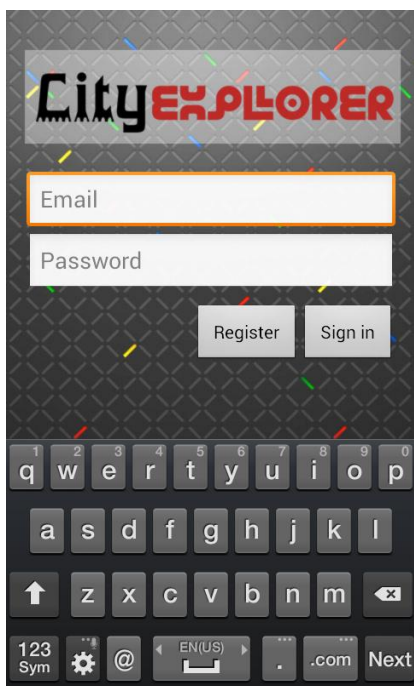


Εικόνα 14 Πολυνημάτωση

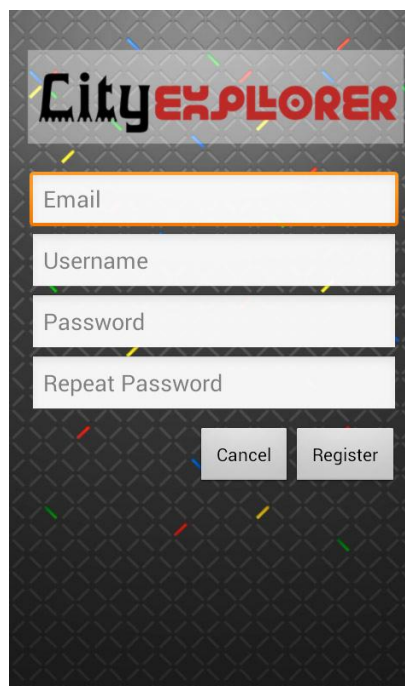
Ένα νήμα εκτέλεσης (thread) είναι η μικρότερη ακολουθία προγραμματισμένων εντολών που μπορεί να διαχειρισθεί ανεξάρτητα, από το λειτουργικό σύστημα. Είναι μια ελαφριά διεργασία και μπορούν να υπάρχουν πολλαπλά νήματα μέσα στην ίδια διεργασία (όπως activity στην περίπτωση μας) τα οποία μπορούν να μοιράζονται πόρους από το σύστημα, όπως μνήμη κάνοντας την εκτέλεση του κώδικα πολύ πιο γρήγορη και αποτελεσματική, ειδικά σε συστήματα που περιλαμβάνουν πολυπύρηνους επεξεργαστές.

4.5.1 Σύνδεση με τον σέρβερ για πιστοποίηση ή εγγραφή του χρήστη

Όπως κάθε διαδικτυακή εφαρμογή που χρησιμοποιείται για ιδιωτική χρήση και χρησιμοποιείται από πολλούς χρήστες, έτσι και στο CityExplorer πρέπει να υπάρχει έλεγχος των προσωπικών στοιχείων ή εγγραφή νέου χρήστη για την ολοκληρωμένη λειτουργία του και την παροχή των υπηρεσιών στον κάθε ένα ξεχωριστά. Ο χρήστης εισάγει τα στοιχεία του σε μια φόρμα και γίνεται ο έλεγχος τους σε σύνδεση με τον σέρβερ δεδομένου ότι η συσκευή έχει πρόσβαση στο διαδίκτυο. Για την περίπτωση που ο χρήστης δεν έχει λογαριασμό θα πρέπει να κάνει εγγραφή στο σύστημα δηλώνοντας κάποια στοιχεία του (username , email , password) ώστε να έχει πρόσβαση στις υπηρεσίες που προσφέρει η εφαρμογή μας.



Εικόνα 15 Είσοδος χρήστη



Εικόνα 16 Εγγραφή χρήστη

Σε αυτό το κομμάτι αρχικά έγιναν δοκιμές για τη σύνδεση της εφαρμογής με τον erodio και με τη βάση δεδομένων που φιλοξενεί. Η εφαρμογή καλεί συγκεκριμένα scripts που είναι αποθηκευμένα στον σέρβερ και εκτελούν τις αντίστοιχες λειτουργίες.

Για τον έλεγχο των στοιχείων email και password και επιτυχή εισαγωγή του χρήστη στο σύστημα καλείτε ένα συγκεκριμένο πρόγραμμα το οποίο παραλαμβάνει τις παραπάνω πληροφορίες από την εφαρμογή και εκτελεί ερωτήματα στη βάση δεδομένων. Στον πίνακα 'user' υπάρχουν τα πεδία 'email' και 'password' όπου και γίνεται η αναζήτηση. Αν βρεθούν τα στοιχεία που έδωσε ο χρήστης τότε η βάση επιστρέφει τα αποτελέσματα στο PHP script και από εκεί αποστέλλονται πίσω στην εφαρμογή. Εκεί γίνεται και δεύτερος έλεγχος ώστε να διαβεβαιωθεί ότι τα στοιχεία είναι τα σωστά. Αν τα στοιχεία που έδωσε ο χρήστης είναι έγκυρα, η εφαρμογή φορτώνει το κεντρικό μενού και δίνει την δυνατότητα στον χρήστη να προχωρήσει ομαλά στην λειτουργία της. Αν βρεθεί κάποιο λάθος τότε εμφανίζεται ένα μήνυμα στην οθόνη του κινητού που ενημερώνει τον χρήστη ότι τα στοιχεία που δόθηκαν είναι λάθος και παραμένει στην φόρμα εισόδου.

Αντίστοιχα γίνεται και η διαδικασία στην περίπτωση εγγραφής, εισαγωγής δηλαδή νέας γραμμής στον πίνακα 'user' της βάσης δεδομένων. Σε αυτήν την

περίπτωση χρειάζονται περισσότεροι έλεγχοι αφού θα πρέπει να ψάξουμε στην βάση αν υπάρχει κάποιος άλλος χρήστης που έχει δηλώσει ήδη κάποιο από τα στοιχεία 'username' ή 'email'. Με την κλήση του αντίστοιχου προγράμματος PHP και εκτέλεση ερωτημάτων στην βάση γίνονται όλοι οι απαραίτητοι έλεγχοι. Αν βρεθούν στην βάση τότε δεν είναι εφικτή η ολοκλήρωση εγγραφής του χρήστη από τη στιγμή που η ταυτότητα του θα πρέπει να είναι μοναδική και όχι ήδη υπάρχουσα στην βάση. Αν τα στοιχεία που έχει δηλώσει ο χρήστης είναι μοναδικά τότε γίνεται επιτυχημένη εισαγωγή των δεδομένων στην βάση και πραγματοποιείται και αυτόματη είσοδος στο σύστημα.

4.5.2 Ενσωμάτωση των Google Maps στην εφαρμογή

Ένα από τα βασικότερα κομμάτια της εφαρμογής CityExplorer είναι οι χάρτες της Google που προσφέρουν την δυνατότητα στην εφαρμογή να εμφανίσει την τρέχουσα τοποθεσία σε πραγματικό χάρτη ή κάποιων γνωστών τοποθεσιών αν γνωρίζουμε το γεωγραφικό μήκος και πλάτος.



Εικόνα 17 Λογότυπο Google Maps

Για να γίνει ενσωμάτωση των χαρτών στην εφαρμογή έπρεπε να κάνουμε μια σειρά από ενέργειες ώστε να γίνει εφικτή η εμφάνιση τους, η σύνδεση με τους διακομιστές της Google για λήψη των χαρτογραφημένων περιοχών και η δυνατότητα να εκτελέσουμε εντολές πάνω στους χάρτες.

Αρχικά έπρεπε να γίνει εισαγωγή των απαραίτητων βιβλιοθηκών στο Eclipse για να έχουμε πλήρη πρόσβαση στις εκτελέσιμες εντολές και δυνατότητες των χαρτών. Αυτό έγινε εισάγοντας τις βιβλιοθήκες Google Play Services SDK, οι

οποίες περιέχουν και το Android Maps API που θέλουμε να χρησιμοποιήσουμε, μέσω του Android SDK Manager στο Eclipse.

Για να αποκτήσουμε πρόσβαση στις διακομιστές του Google Maps με το Maps API, προσθέσαμε ένα Maps API κλειδί στην εφαρμογή μας. Το κλειδί είναι δωρεάν, μπορούμε να το χρησιμοποιήσετε με οποιαδήποτε από τις εφαρμογές μας που καλούν το Maps API ή χρησιμοποιούν χάρτες, και υποστηρίζει απεριόριστο αριθμό χρηστών. Αποκτήσαμε ένα Maps API κλειδί από το Google APIs Console παρέχοντας το πιστοποιητικό υπογραφής της εφαρμογής μας και το όνομα του πακέτου. Μόλις λάβαμε το κλειδί, το εισάγαμε στην εφαρμογή μας, προσθέτοντας ένα στοιχείο στο αρχείο δήλωσης της εφαρμογής AndroidManifest.xml.

Η κατανόηση της διαδικασίας της δήλωσης της εφαρμογής μας για την απόκτηση ενός κλειδιού απαιτεί κάποια γνώση της έκδοσης των εφαρμογών Android στο εμπόριο μέσω του καταστήματος Google Play και των απαιτήσεων για αυτήν την διαδικασία. Εν ολίγοις, όλες οι εφαρμογές Android θα πρέπει να υπογράφονται με ένα ψηφιακό πιστοποιητικό για το οποίο εμείς κρατάμε το κλειδί του. Επειδή τα ψηφιακά πιστοποιητικά είναι μοναδικά, παρέχουν έναν απλό τρόπο μοναδικής ταυτοποίησης των εφαρμογών μας. Αυτό τα καθιστά χρήσιμα για την παρακολούθηση της εφαρμογής σε συστήματα όπως το κατάστημα της Google Play, καθώς και για την παρακολούθηση των πόρων που χρησιμοποιεί η εφαρμογή μας σε συστήματα, όπως το Google Maps servers.

Τα Maps API κλειδιά συνδέονται με συγκεκριμένα ζευγάρια πιστοποιητικών και πακέτων παρά με χρήστες ή εφαρμογές. Χρειάζεται μονάχα ένα κλειδί για κάθε πιστοποιητικό ανεξαρτήτως τον αριθμό των χρηστών που χρησιμοποιούν την εφαρμογή. Εφαρμογές που χρησιμοποιούν το ίδιο πιστοποιητικό χρησιμοποιούν και το ίδιο Maps API κλειδί. Τα κλειδιά αυτά βασίζονται σε μια σύντομη μορφή του ψηφιακού πιστοποιητικού της εφαρμογής μας, γνωστό και ως SHA-1 fingerprint, το οποίο είναι για σειρά από δεκαεξαδικούς χαρακτήρες και παράγεται από τον αλγόριθμο SHA-1 hashing, και χρησιμοποιείται από το Google Maps για ταυτοποίηση των εφαρμογών. Για να εμφανίσουμε το κατάλληλο SHA-1 fingerprint θα πρέπει να αποφασίσουμε τι πιστοποιητικό

θέλουμε. Υπάρχουν 2 ειδών πιστοποιητικά, το Debug Certificate και Release Certificate.

Το Debug Certificate παράγεται αυτόματα όταν τρέχουμε δοκιμαστικά την εφαρμογή μας κατά την διάρκεια ανάπτυξης της. Χρησιμοποιείται για δοκιμαστικό στάδιο και δεν χρειάζεται να δημοσιεύσουμε επίσημα την εφαρμογή μας. Το Release Certificate παράγεται όταν η εφαρμογή μας είναι έτοιμη προς δημοσίευση και διαλέγουμε την αντίστοιχη επιλογή στο Eclipse ώστε να μας 'χτίσει' την εφαρμογή προς δημοσίευση. Στην περίπτωση μας χρησιμοποιήσαμε το Debug Certificate αφού η εφαρμογή αναπτύχθηκε για εκπαιδευτικούς και όχι εμπορικούς λόγους. Παρακάτω θα δούμε τα επόμενα βήματα που ακολουθήσαμε για να μπορέσουμε να δηλώσουμε την εφαρμογή και να έχουμε πρόσβαση στους διακομιστές του Google Maps.

Για να εκδώσουμε το κατάλληλο SHA-1 fingerprint για το Debug Certificate σε Mac OS έπρεπε να εκτελέσουμε την παρακάτω εντολή στο terminal.

```
keytool -list -v -keystore ~/.android/debug.keystore -alias androiddebugkey -storepass android -keypass android
```

Το αποτέλεσμα φαίνεται στην παρακάτω εικόνα. Από τις πληροφορίες που μας δίνει η συγκεκριμένη εντολή χρειαζόμαστε την σειρά δεκαεξαδικών χαρακτήρων που επισημάνεται ως SHA1. (έχουν σβηστεί οι 4οις τελευταίοι χαρακτήρες από κάθε fingerprint για λόγους ασφαλείας)

Γκούτζος Άρης

```
Arís — bash — 80x24
Last login: Sat Sep 14 12:26:38 on console
arismbpro:~ Aris$ keytool -list -v -keystore ~/.android/debug.keystore -alias an
droiddebugkey -storepass android -keypass android

Alias name: androiddebugkey
Creation date: May 3, 2013
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Android Debug, O=Android, C=US
Issuer: CN=Android Debug, O=Android, C=US
Serial number: 5183bebd
Valid from: Fri May 03 15:42:21 CEST 2013 until: Sun Apr 26 15:42:21 CEST 2043
Certificate fingerprints:
    MD5:  DD:8C:C7:8D:4F:99:A7:FE:F9:69:59:6C:70:2E:
    SHA1: F3:18:04:62:02:5F:00:D5:E5:E5:28:EB:8B:02:58:DE:D8:D8:
    Signature algorithm name: SHA1withRSA
    Version: 3
arismbpro:~ Aris$
arismbpro:~ Aris$
```

Εικόνα 18 Εμφάνιση απαραίτητου αποτυπώματος για δήλωση της εφαρμογής

Στην συνέχεια δημιουργούμε ένα project στο Google APIs Console¹, ενεργοποιούμε το Google Maps Android API v2 από τις διαθέσιμες υπηρεσίες που μας προσφέρει η Google (Services) και δημιουργούμε ένα νέο Android κλειδί. Εκεί θα μας ζητηθεί να εισάγουμε το SHA-1 fingerprint, που αποκτήσαμε από τις προηγούμενες διαδικασίες, ακολουθούμενο από ελληνικό ερωτηματικό (;) και το όνομα του πακέτου της εφαρμογής μας. Για παράδειγμα:

```
BB:0D:AC:74:D3:21:E1:43:67:71:9B:62:91:AF:A1:66:6E:44:5D:75;com.example.mapsapp
```

Έπειτα το Google APIs Console μας επιστρέφει το πολυαναζητούμενο Maps API κλειδί που χρειαζόμαστε και έχει μήκος 40 χαρακτήρων, για παράδειγμα:

```
AIzaSyBdVI-cTICSwYKrZ95SuvNw7dbMuDt1KG0
```

Εισάγουμε το παραπάνω κλειδί στην εφαρμογή μας προσθέτοντας τον παρακάτω κώδικα στο αρχείο AndroidManifest.xml και αντικαθιστώντας το API_KEY με το κλειδί που αποκτήσαμε.

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

Τέλος, εισάγουμε το αντίστοιχο MapFragment στοιχείο στην φόρμα μας, ρυθμίζουμε το μέγεθος που θέλουμε να έχει ο χάρτης και είμαστε έτοιμοι!

Αυτή η διαδικασία είναι απαραίτητο να ακολουθηθεί από οποιονδήποτε θέλει να εκμεταλλευτεί τις δωρεάν υπηρεσίες που προσφέρει η Google στους προγραμματιστές Android εφαρμογών.

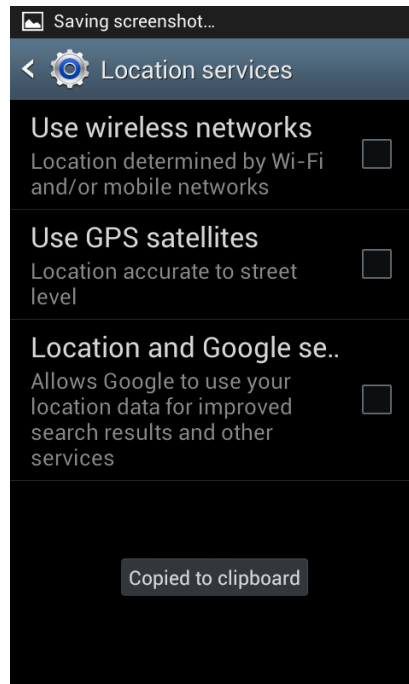
4.5.3 Λήψη τοποθεσίας και εμφάνιση της στον χάρτη

Για να γίνει δυνατή η εμφάνιση των κοντινότερων αποστολών στον χάρτη ανάλογα με την τρέχουσα τοποθεσία του χρήστη όπως είναι φυσικό απαιτείται να υπάρχει κάποια υπηρεσία τοποθεσιών (Location Services) ενεργοποιημένη στην συσκευή.

Υπάρχουν 2 δυνατοί τρόποι να λάβουμε την τρέχουσα τοποθεσία της συσκευής μας, χρησιμοποιώντας την σύνδεση στο διαδίκτυο που ενδεχομένως να υπάρχει στην συσκευή ή χρησιμοποιώντας τον ενσωματωμένο δεκτή GPS που περιέχει η συσκευή μας για να λάβουμε το σήμα από τους αντίστοιχους δορυφόρους. Η πρώτη επιλογή είναι συντριπτικά πιο γρήγορη αφού η τοποθεσία υπολογίζεται από διάφορους αλγόριθμους μέσω της mobile ή WiFi σύνδεσης στο διαδίκτυο και δεν χρειάζεται λήψη σημάτων από δορυφόρους GPS και συγχρονισμός της συσκευής με αυτούς. Αντιστρόφως, ο υπολογισμός της τοποθεσίας μέσω GPS προσφέρει πιο ακριβή αποτελέσματα αφού γίνεται συγχρονισμός της συσκευής με πολλούς δορυφόρους ανά την περίμετρο της γης, έτσι υπάρχει υπερβολική ακρίβεια στην τρέχουσα τοποθεσία που υπολογίζεται.

Στην εφαρμογή, όταν ο χρήστης επιλέξει να εμφανιστούν οι κοντινότερες αποστολές στον χάρτη, γίνεται έλεγχος αν υπάρχει τουλάχιστο μια ενεργοποιημένη υπηρεσία υπολογισμού τοποθεσίας, εκτελώντας την παρακάτω εντολή.

```
( locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER) ||  
locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER) )
```



Εικόνα 19 Οθόνη ρυθμίσεων Location services

Σε αντίθετη περίπτωση γίνεται εμφάνιση αντίστοιχου μηνύματος και οδηγεί τον χρήστη οθόνη ρυθμίσεων της συσκευής, στον τομέα των υπηρεσιών τοποθεσίας, όπως φαίνεται στην παρακάτω εικόνα, ώστε να ενεργοποιήσει κάποια από τις υπηρεσίες και να επιστρέψει στην εφαρμογή.

Παρόμοια διαδικασία ακολουθείται και στην περίπτωση που δεν υπάρχει ενεργή σύνδεση στο διαδίκτυο, όπως θα δούμε στο επόμενο κεφάλαιο.

4.5.4 Εμφάνιση των κοντινότερων στην τρέχουσα τοποθεσία αποστολών

Σε αυτό το στάδιο, αφού γνωρίζουμε πλέον την τρέχουσα τοποθεσία της συσκευής μας, η οποία εκφράζεται σε γεωγραφικό μήκος και πλάτος (latitude, longitude), εκτελούμε ερωτήματα στην βάση δεδομένων στέλνοντας τα συγκεκριμένα στοιχεία και επιπλέον την ταυτότητα του χρήστη. Γίνεται η κλήση κατάλληλων μεθόδων που συγκρίνουν την συγκεκριμένη τοποθεσία με τα πεδία latitude και longitude των αποθηκευμένων τοποθεσιών στον πίνακα **'poi'** της βάσης μας. Η απάντηση είναι ένας πίνακας, κωδικοποιημένος σε JSON όπως είδαμε προηγούμενος, που περιέχει τις διαθέσιμες αποστολές εντός 500μ από την τοποθεσία μας (εύκολα ρυθμιζόμενο, αλλάζοντας την τιμή μιας μεταβλητής)

και τα στοιχεία των σημείων ενδιαφέροντος που προσφέρουν αυτές τις αποστολές.



Εικόνα 20 Τα σημάδια χάρτη που χρησιμοποιήσαμε

Στη συνέχεια γίνεται κατάλληλη επεξεργασία των πληροφοριών αυτών ώστε να διαχωριστούν οι αποστολές σε ολοκληρωμένες και μη ολοκληρωμένες. Στις πρώτες ανατίθεται ένα σημάδι χάρτη (`map marker`) χρώματος πράσινο και στις δεύτερες ένα χρώματος μπλε. Έτσι ο χρήστης θα γνωρίζει ποιές από τις αποστολές που εμφανίζονται στον χάρτη είναι ήδη ολοκληρωμένες και ποιές είναι διαθέσιμες.

Τα σημάδια εισάγονται στον χάρτη εκτελώντας, κατά κύριο λόγο, την παρακάτω εντολή και ορίζοντας σαν ιδιότητες τους το γεωγραφικό μήκος και πλάτος που λάβαμε από την βάση και την ταυτότητα (`id`) της τοποθεσίας.

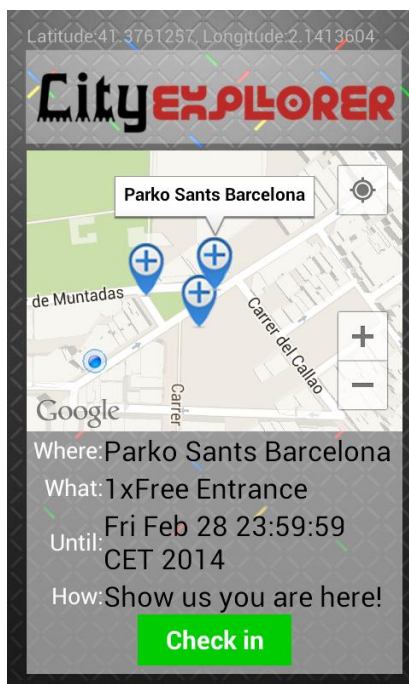
```
Marker marker = googleMap.addMarker();
```

Όπου `googleMap` η μεταβλητή που διαχειρίζεται τον χάρτη.

4.5.5 Επιλογή αποστολής, εμφάνιση λεπτομερειών και ολοκλήρωση της

Ο χρήστης μπορεί να επιλέξει κάποιο από τα σημάδια που εμφανίζονται στον χάρτη και να δει ακριβώς από κάτω τις πληροφορίες που αντιστοιχούν στο συγκεκριμένο σημείο και τις λεπτομέρειες της αποστολής. Αυτό επιτυγχάνεται καθώς με την επιλογή του σημαδιού από τον χρήστη, αποθηκεύεται σε μια μεταβλητή η ταυτότητα (`id`) της τοποθεσίας, την οποία χρησιμοποιούμε για να εκτελέσουμε νέο ερώτημα στην βάση ώστε να μας στείλει τις αντίστοιχες πληροφορίες από τους πίνακες **'poi'**, **'task'** και **'award'**.

Γκούτζος Άρης

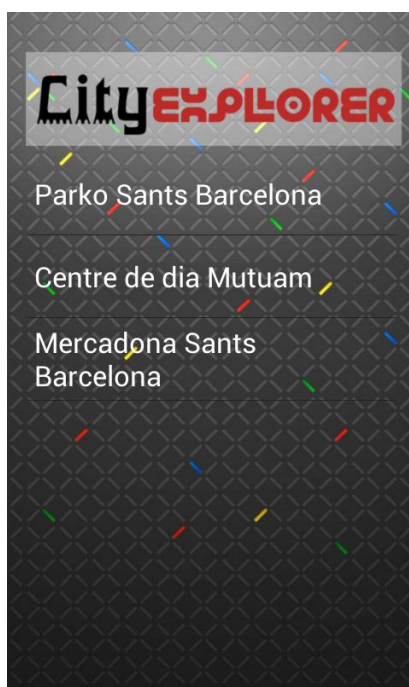


Εικόνα 21 Οθόνη εμφάνισης αποστολών

Όταν η αποστολή έχει πράσινο σημάδι σημαίνει ότι είναι ήδη ολοκληρωμένη, εμφανίζεται το αντίστοιχο μήνυμα στην οθόνη και ο χρήστης μπορεί να δει το βραβείο του στο ιστορικό (keystore), το οποίο θα αναλύσουμε στην επόμενη ενότητα. Όταν η αποστολή έχει μπλε σημάδι τότε είναι μια διαθέσιμη αποστολή και ο χρήστης μπορεί να την ολοκληρώσει ακολουθώντας τις οδηγίες που περιγράφονται (How) στην οθόνη της συσκευής.

Όταν ο χρήστης ολοκληρώσει σωστά μια αποστολή, το οποίο επιβεβαιώνεται από ελέγχους που γίνονται στο πρόγραμμα, με τις κατάλληλες ενέργειες προστίθεται στον πίνακα **'activity'** της βάσης δεδομένων η ταυτότητα του χρήστη στο πεδίο (donebyID), ο κωδικός της αποστολής (taskID) και η ημερομηνία ολοκλήρωσης. Αυτόματα το σημείο που σχετίζεται με την αποστολή αλλάζει χρώμα σε πράσινο αφού πλέον έχει ολοκληρωθεί.

4.5.6 Εμφάνιση ιστορικού ολοκληρωμένων αποστολών και αντίστοιχων βραβείων



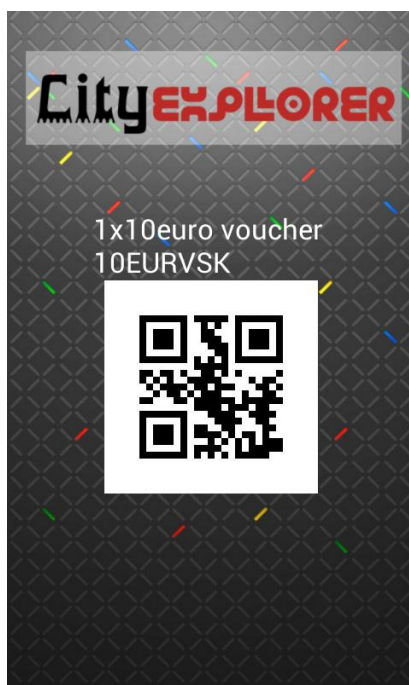
Εικόνα 22 Οθόνη εμφάνισης ολοκληρωμένων αποστολών

Πηγαίνοντας στο myHistory από το κεντρικό μενού εμφανίζεται μια λίστα με τις τοποθεσίες των αποστολών που έχει κατορθώσει να ολοκληρώσει ο χρήστης. Για να ανοίξει η φόρμα αυτή πραγματοποιούνται οι κατάλληλοι έλεγχοι για το αν υπάρχει πρόσβαση στο διαδίκτυο από τη συσκευή, όπως είδαμε παραπάνω. Σε περίπτωση που δεν υπάρχει γίνεται εμφάνιση του αντίστοιχου μηνύματος και προτροπή του χρήστη να ενεργοποιήσει κάποια σύνδεση. Η συγκεκριμένη λίστα φορτώνεται εκτελώντας ερωτήματα στην βάση δεδομένων στέλνοντας το id του χρήστη που είναι εκείνη τη στιγμή συνδεδεμένος στην εφαρμογή. Τα ερωτήματα αυτά βρίσκουν τις ολοκληρωμένες αποστολές του χρήστη, από τον πίνακα **'activity'**, και επιστρέφουν όλες εκείνες τις πληροφορίες που χρειαζόμαστε για να τις εμφανίσουμε στην οθόνη της συσκευής.

Η αποθήκευσή τους γίνεται τοπικά χρησιμοποιώντας μια βοηθητική κλάση που δημιουργήσαμε για αυτό το σκοπό και αν ο χρήστης επιστρέψει στο κύριο μενού, όλες οι πληροφορίες διαγράφονται. Η κλάση αυτή (CompletedTaskList) αποθηκεύει τις απαραίτητες πληροφορίες σε ένα HashMap χρησιμοποιώντας ως

χάρτη το 'roiID' της αποστολής, έτσι ώστε να μπορούμε πολύ εύκολα να εμφανίσουμε, όπως θα δούμε παρακάτω, σε νέα φόρμα το βραβείο που αντιστοιχεί σε κάθε αποστολή που εμφανίζεται στην λίστα.

Στην συνέχεια, όταν ο χρήστης επιλέξει ένα στοιχείο της λίστας ανοίγει νέα φόρμα που περιέχει την περιγραφή του βραβείου που έχει κερδίσει ο χρήστης ολοκληρώνοντας την αποστολή της συγκεκριμένης τοποθεσίας. Επίσης εμφανίζεται και η ποσότητα του βραβείου (quantity), ο ειδικός κωδικός του και ο γραμμωκώδικας QRcode. Όλα αυτά τα στοιχεία τα βρίσκομαι αναζητώντας στο HashMap την ολοκληρωμένη αποστολή (CompletedTask) με αριθμό ταυτότητας (roiID) την ταυτότητα του επιλεγμένου στοιχείου της λίστας.



Εικόνα 23 Οθόνη εμφάνισης βραβείου και αντίστοιχου κωδικού

Το QRcode παράγεται πολύ εύκολα προσθέτοντας στο τέλος της παρακάτω ηλεκτρονικής διεύθυνσης τον ειδικό κωδικό του βραβείου (promo_code) και το όνομα του συνδεδεμένου χρήστη (username).

[http://chart.googleapis.com/chart?chs=250x250&cht=qr&chl="](http://chart.googleapis.com/chart?chs=250x250&cht=qr&chl=)

Για παράδειγμα στην παραπάνω εικόνα το QRcode φορτώνεται από την παρακάτω ηλεκτρονική διεύθυνση.

["http://chart.googleapis.com/chart?chs=250x250&cht=qr&chl=10EURVS_K_user1"](http://chart.googleapis.com/chart?chs=250x250&cht=qr&chl=10EURVS_K_user1)

4.6 Επίλογος

Η δημιουργία μιας εφαρμογής με ενσωμάτωση Google Maps, σύνδεση με βάση δεδομένων, τμηματοποίηση κώδικα και χρησιμοποίηση άλλων μικρότερης δυσκολίας τεχνολογιών δημιουργεί πολλά ερωτήματα για τον τρόπο κατασκευής της. Το γεγονός ότι διαφορετικές τεχνολογίες και γλώσσες προγραμματισμού πρέπει να συνεργαστούν για το τελικό αποτέλεσμα μας οδήγησε στην κατάτμηση του έργου σε μικρά παραδοτέα με αυτοτελή λειτουργία τμήματα. Ο διαχωρισμός τους έγινε με υποκειμενικά κριτήρια κάτω από ώριμη σκέψη με στόχο την απλή χρήση της εφαρμογής από την πλευρά του χρήστη και την εύκολη υλοποίηση τους από την πλευρά του προγραμματιστή. Ο συνδυασμός των διαφορετικών γλωσσών προγραμματισμού δίνει σε κάθε κομμάτι την απαιτούμενη λειτουργικότητα με τον πιο αποδοτικό τρόπο. Με το τέλος της κατασκευής όλων των τμημάτων έγινε η ένωση τους με κάποιες μικρές τροποποιήσεις και παρεμβάσεις, οδηγώντας στην ολοκληρωμένη λειτουργία της εφαρμογής. Όλα τα υποέργα περιέχουν κάποια κοινά και κάποια διαφορετικά στοιχεία προγραμματισμού. Σε αυτό το κεφάλαιο εξηγήσαμε πως υλοποιήθηκαν αυτά τα τμήματα από την πλευρά του προγραμματιστή και στο επόμενο κεφάλαιο θα αναλύσουμε τον τρόπο λειτουργίας της εφαρμογής από την μεριά του χρήστη.

ΚΕΦΑΛΑΙΟ 5. Χρήση της εφαρμογής

5.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα περιγράψουμε τα βήματα αναλυτικά από την αρχή ως το τέλος για να οδηγηθούμε στον τελικό στόχο που είναι η ολοκλήρωση μιας αποστολής και η παραλαβή του ειδικού κωδικού του βραβείου της. Πως μπορεί κάποιος χρήστης που θεωρητικά μόλις έχει εγκαταστήσει την εφαρμογή να πραγματοποιήσει αυτές τις ενέργειες. Επίσης, θα δούμε τις τεχνικές που χρησιμοποιήσαμε για να μεταφέρουμε τα σημαντικότερα δεδομένα από μια φόρμα σε κάποια άλλη.

Ο χρήστης για να μπορέσει να κάνει λειτουργήσει την εφαρμογή θα πρέπει να δημιουργήσει έναν λογαριασμό δηλώνοντας στο σύστημα μια ηλεκτρονική του διεύθυνση, ένα όνομα χρήστη και έναν κωδικό ή να πραγματοποιήσει σύνδεση σε αυτόν χρησιμοποιώντας την ηλεκτρονική του διεύθυνση και ως κωδικό την αλφαριθμητική σειρά χαρακτήρων που έχει δηλώσει. Στη συνέχεια θα μπορεί να πλοηγηθεί ανάμεσα στις αποστολές που βρίσκονται μέσα σε κάποια απόσταση από την τρέχουσα τοποθεσία του και να διαλέξει αυτήν που θέλει να ολοκληρώσει. Τέλος, μπορεί να παραλάβει τον κωδικό του βραβείου του και να το παραδώσει στον υπεύθυνο που δημιούργησε την αποστολή ή να τον αφήσει να σκανάρει το αντίστοιχο QRcode για να εξαργυρώσει το βραβείο του. Για να δούμε ένα πραγματικό σενάριο, ας υποθέσουμε ότι πηγαίνουμε σε μια καφετέρια και υπάρχει μια διαθέσιμη αποστολή η οποία απαιτεί να απαντήσουμε σωστά στο παρακάτω ερώτημα: «Πως ονομάζεται ο σπέσιαλ καφές του καταστήματος μας;» και σαν βραβείο στην σωστή απάντηση αυτού του ερωτήματος δίνεται ένας κωδικός που μας προσφέρει μια κούπα από τον σπέσιαλ καφέ του καταστήματος δωρεάν. Οπότε το μόνο που έχουμε να κάνουμε είναι να απαντήσουμε σωστά και να παραδώσουμε τον κωδικό στον υπεύθυνο του καταστήματος ή στην/στον σερβιτόρα/ο και να απολαύσουμε τον δωρεάν καφέ.

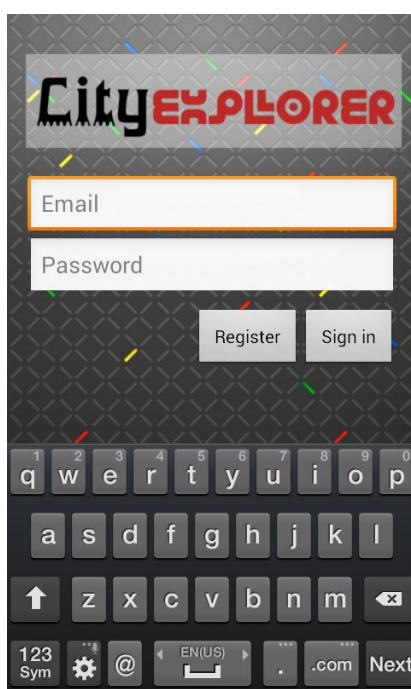
Η εφαρμογή είναι ευέλικτη στην περαιτέρω ανάπτυξη της, προσθέτοντας εύκολα είδη αποστολών και σύνδεση της με μια online πλατφόρμα που θα δίνει την δυνατότητα προσθήκης σημείων καθώς και δημιουργία και διαχείριση των αποστολών.

5.2 Φόρμες της εφαρμογής και ανάλυση τους

Η εφαρμογή CityExplorer μετά από κάποιες απλοποιήσεις για την αποτελεσματικότερη χρήση της κατέληξε να αποτελείται από 5 φόρμες. Κάθε μια από τις φόρμες εκτελεί ξεχωριστή λειτουργία επομένως και ο κώδικας που υπάρχει πίσω από αυτήν αποτελείται από διαφορετικές εντολές. Για το τελικό αποτέλεσμα όλες οι φόρμες συνεργάζονται, καθεμία παρέχοντας στην επόμενη σημαντικά δεδομένα για την ολοκλήρωση των λειτουργιών της. Θα παρουσιάσουμε κάθε φόρμα ξεχωριστά εξηγώντας τις λειτουργίες της, θα αναλύσουμε τη ροή των δεδομένων μέσα στην εφαρμογή καθώς και εκείνα τα κομμάτια κώδικα που παίζουν το σημαντικότερο ρόλο στην σωστή διαχείριση των πληροφοριών και θα περιγράψουμε την αλληλεπίδραση της εφαρμογής με τον τελικό χρήστη μέσω σχεδίασης των διεπαφών για βέλτιστη διασύνδεση χρήστη και ηλεκτρονικού υπολογιστή.

5.2.1 Πιστοποίηση και εγγραφή χρήστη

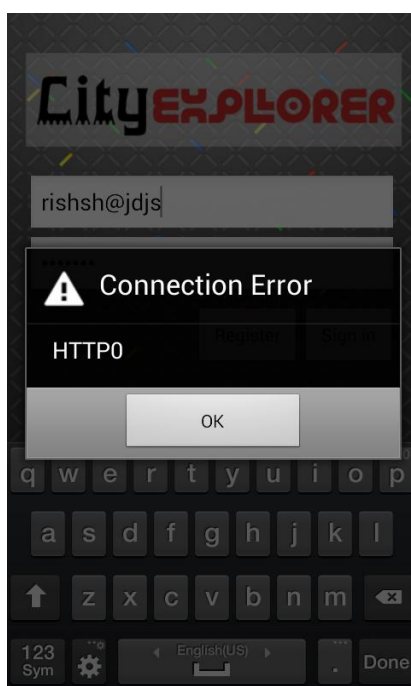
Για να μπορέσει να χρησιμοποιήσει κάποιος την εφαρμογή θα πρέπει είτε να διαθέτει ήδη κάποιον λογαριασμό και χρησιμοποιώντας την ηλεκτρονική του διεύθυνση (email) και τον κωδικό πρόσβασης να εισέλθει στο σύστημα είτε να εγγραφεί στο σύστημα δημιουργώντας έναν νέο λογαριασμό.



Εικόνα 24 Φόρμα εισαγωγής χρήστη

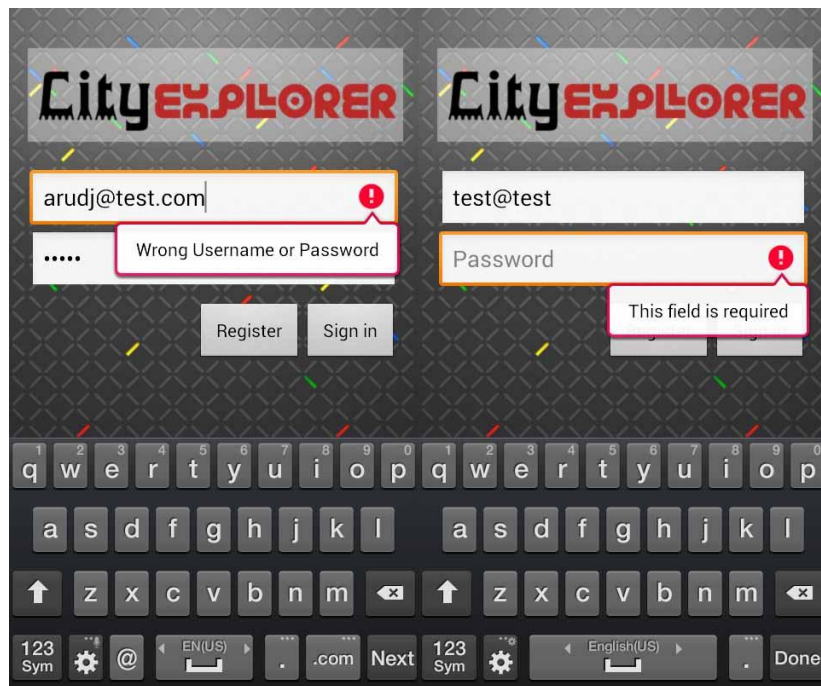
Στην παραπάνω εικόνα βλέπουμε την φόρμα για την είσοδο στον σύστημα. Υπάρχουν 2 πεδία για εισαγωγή στην στοιχεία που δηλώνουν στον χρήστη ότι θα πρέπει να εισάγει το email του και τον κωδικό πρόσβασης του. Ο έλεγχος των στοιχείων πραγματοποιείται με την εκτέλεση κάποιων PHP προγραμμάτων όπως αναφέραμε στο προηγούμενο κεφάλαιο. Οι περιπτώσεις που προκύπτουν ύστερα από το πάτημα του πλήκτρου με την ένδειξη 'Sign in', που ουσιαστικά πυροδοτεί την εκτέλεση του κώδικα για έλεγχο των στοιχείων, είναι: μη δυνατή σύνδεση με τον διακομιστή, μη εισαγωγή κανενός ή κάποιου απαραίτητου στοιχείου, λανθασμένη εισαγωγή κάποιου στοιχείου και επιτυχής σύνδεση στο σύστημα.

Στην περίπτωση που δεν είναι δυνατή η σύνδεση με τον διακομιστή εμφανίζεται το παρακάτω μήνυμα δηλώνοντας μας τον αριθμό σφάλματος που μας επέστρεψε η προσπάθεια σύνδεσης στο σύστημα.



Εικόνα 25 Μήνυμα αδυναμίας σύνδεσης με το σύστημα

Σε περίπτωση λανθασμένης εισαγωγής ή παράλειψης κάποιου από τα στοιχεία έχουμε την εμφάνιση αντίστοιχου μηνύματος όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 26 Προειδοποιήσεις για σωστή εισαγωγή στοιχείων

Αν όλα τα στοιχεία που έχουμε εισάγει είναι σωστά, το όνομα χρήστη και ο μοναδικός αριθμός (id) που τον συνοδεύει στην βάση δεδομένων αποθηκεύονται σε κοινές μεταβλητές εσωτερικά της εφαρμογής χρησιμοποιώντας το παρακάτω κομμάτι κώδικα.

```
//Shared preferences to store the username and id of the logged in user
```

```
public static final String PREFS_NAME = "sharedPrefs";
```

```
//Passing username of the user to shared preferences
```

```
SharedPreferences settings = getSharedPreferences(PREFS_NAME, 0);
```

```
SharedPreferences.Editor editor = settings.edit();
```

```
// set it to false when the user is logged out
```

```
editor.putBoolean("hasLoggedIn", true);
```

```
editor.putString("username", retUsername);
```

```
editor.putInt("userID", retUserID);
```

```
// Commit the edits!
```

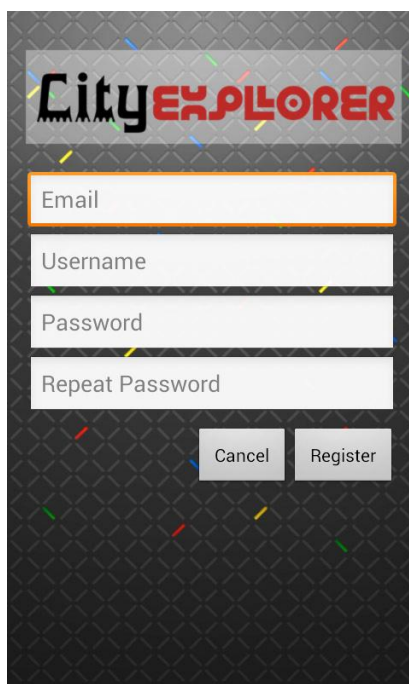
```
editor.commit();
```

Έτσι μας δίνεται η δυνατότητα την επόμενη φορά που ο χρήστης θα τρέξει την εφαρμογή, ελέγχοντας απλά την εφαρμογή `'hasLoggedIn'` να διαπιστώσουμε αν

Γκούτζος Άρης

έχει ήδη πραγματοποιήσει είσοδο στο σύστημα. Σε αυτήν την περίπτωση η εφαρμογή φορτώνει απευθείας το κεντρικό μενού, αποφεύγοντας την εισαγωγή των δεδομένων του χρήστη ξανά για πιστοποίηση. Σε περίπτωση που ο χρήστης έχει πραγματοποιήσει έξοδο από το σύστημα, λειτουργία που θα αναλύσουμε παρακάτω, ή φορτώνεται για πρώτη φορά η εφαρμογή τότε εμφανίζεται η φόρμα εισαγωγής και εγγραφής χρήστη.

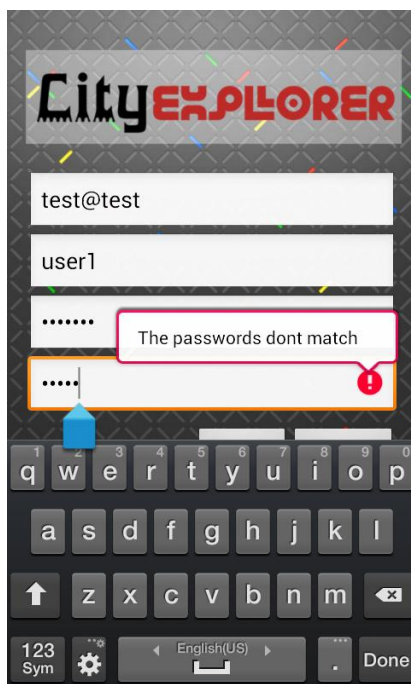
Πατώντας το κουμπί 'Register' προσθέτονται άλλα 2 πεδία εισαγωγής κειμένου 'Username' και 'Repeat Password', κρύβεται το κουμπί 'Sign in' και προσθέτονται ένα ακόμα κουμπί, 'Cancel'.



Εικόνα 27 Φόρμα εγγραφής χρήστη

Ο χρήστης θα πρέπει να συμπληρώσει όλα αυτά τα πεδία που εμφανίζονται στην παραπάνω εικόνα, έτσι ώστε όταν πατήσει το κουμπί 'Register' να προχωρήσει στο επόμενο στάδιο και να αποφύγει τις προειδοποιήσεις. Σε περίπτωση παράλειψης κάποιου πεδίου και απόπειρα εγγραφής η εφαρμογή εμφανίζει αντίστοιχο μήνυμα (προειδοποίηση) επικεντρώνοντας τον κέρσορα στο πεδίο που είναι κενό, όπως είδαμε και παραπάνω στην διαδικασία εισόδου. Εδώ θα πρέπει να σημειώσουμε ότι ο κωδικός πρόσβασης επιβάλλεται να είναι τουλάχιστον 5 χαρακτήρες και το ηλεκτρονικό ταχυδρομείο του χρήστη (email) να περιέχει τον χαρακτήρα '@' για να θεωρηθεί έγκυρος. Τα δεδομένα στα πεδία 'Password' και 'Repeat password' θα πρέπει να είναι πανομοιότυπα, έλεγχος

που πραγματοποιείται προς όφελος του χρήστη για επιβεβαίωση του κωδικού πρόσβασης. Αν τα εισαχθέντα δεδομένα δεν συμφωνούν με τα παραπάνω, γίνεται και εδώ εμφάνιση μηνύματος προειδοποίησης του χρήστη.



Εικόνα 28 Προειδοποίηση αδυναμίας διασταύρωσης κωδικών

Το επόμενο στάδιο, αφού ο χρήστης έχει περάσει επιτυχώς το προηγούμενο, είναι ο έλεγχος των στοιχείων που έχει εισάγει ο χρήστης με αυτά της βάσης δεδομένων. Για να δημιουργηθεί ένας νέος χρήστης δεν θα πρέπει να υπάρχει άλλη εγγραφή στο σύστημα με το ίδιο όνομα χρήστη (username) ή ηλεκτρονικό ταχυδρομείο (email). Κατά το πάτημα του κουμπιού 'Register'

5.2.2 Κεντρικό μενού


Το κεντρικό μενού της εφαρμογής εμφανίζεται ύστερα από επιτυχή σύνδεση του χρήστη στο σύστημα.. Εκεί υπάρχουν 4 ενέργειες που μπορεί κάποιος να πραγματοποιήσει, να επιλέξει το κουμπί 'Explore' ώστε να φορτώσει η φόρμα με τον χάρτη και να εμφανιστούν οι διαθέσιμες αποστολές εντός κάποιας εμβέλειες από την τρέχουσα τοποθεσία της συσκευής.. Επιπλέον, επιλέγοντας το κουμπί 'myHistory', εμφανίζεται η λίστα με τις ολοκληρωμένες αποστολές, όπου ο χρήστης μπορεί να παραλάβει τον ειδικό κωδικό του βραβείου του για κάθε ολοκληρωμένη αποστολή. Βέβαια υπάρχουν κάποιες προϋποθέσεις για να εμφανιστούν οι παραπάνω φόρμες τις οποίες θα αναλύσουμε παρακάτω.

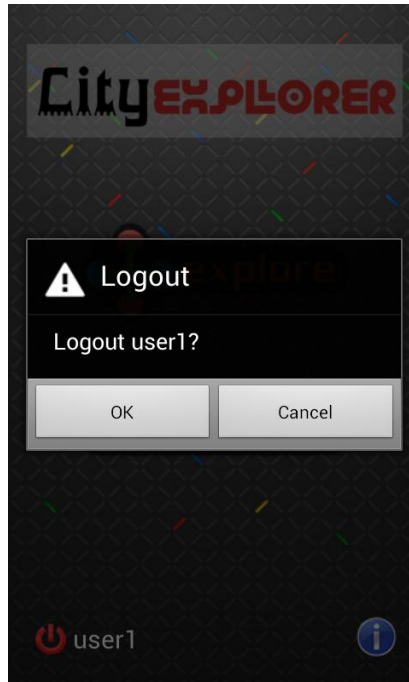


Εικόνα 29 Κεντρικό μενού

Ακόμα σε αυτήν την φόρμα δίνονται άλλες 2 επιλογές για τον χρήστη, να πραγματοποιήσει έξοδο από το σύστημα πατώντας το κουμπί κάτω-αριστερά και να δει κάποιες πληροφορίες για την εφαρμογή πατώντας το κουμπί κάτω-δεξιά.



A) Έξοδος από το σύστημα - Logout

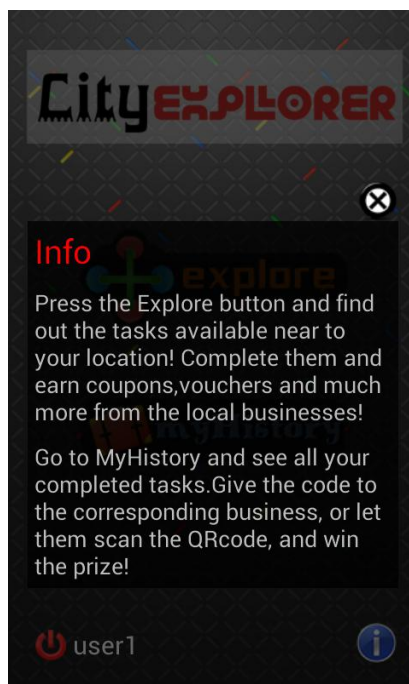
Κάτω αριστερά βλέπουμε ένα κουμπί  και δίπλα του το όνομα του τρέχοντος χρήστη. Επιλέγοντάς το, αποσυνδεόμαστε από το σύστημα και φορτώνεται η φόρμα Εισόδου και Εγγραφής Χρήστη. Πριν οποιαδήποτε ενέργεια η εφαρμογή μας ζητά επιβεβαίωση για το αν θέλουμε όντως να πραγματοποιήσουμε έξοδο από το σύστημα.



Εικόνα 30 Μήνυμα επιβεβαίωσης εξόδου από το σύστημα


B) Πληροφορίες Εφαρμογής - Info

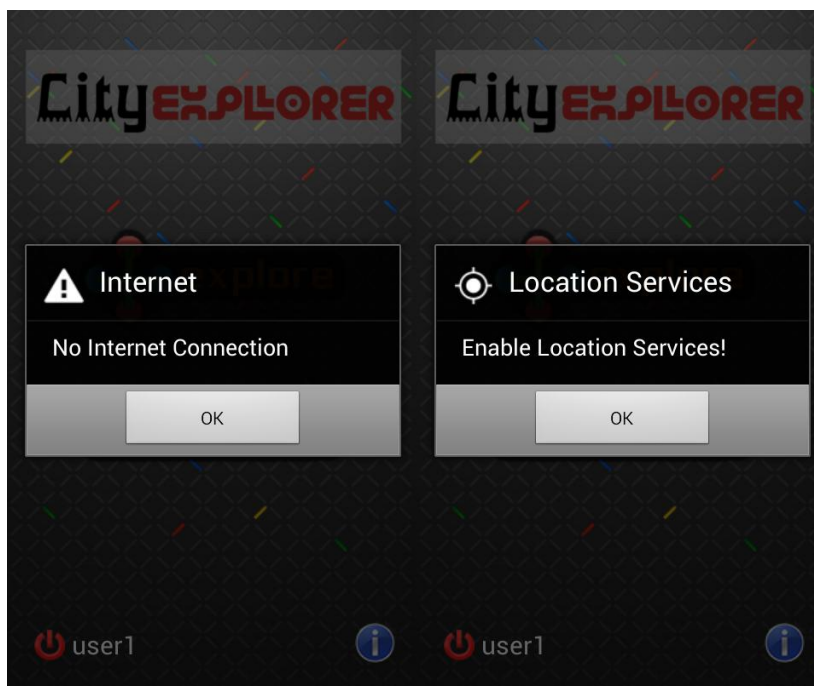
Επιλέγοντας το κουμπί  που βρίσκεται κάτω δεξιά στην οθόνη του κεντρικού μενού, εμφανίζονται κάποιες πληροφορίες για την εφαρμογή και βοηθά τον χρήστη να κατανοήσει τον τρόπο λειτουργίας της. Πατώντας το κουμπί  κλείνει το επιπλέον παράθυρο που άνοιξε και συνεχίζεται κανονικά η ροή της εφαρμογής.



Εικόνα 31 Παράθυρο πληροφοριών εφαρμογής

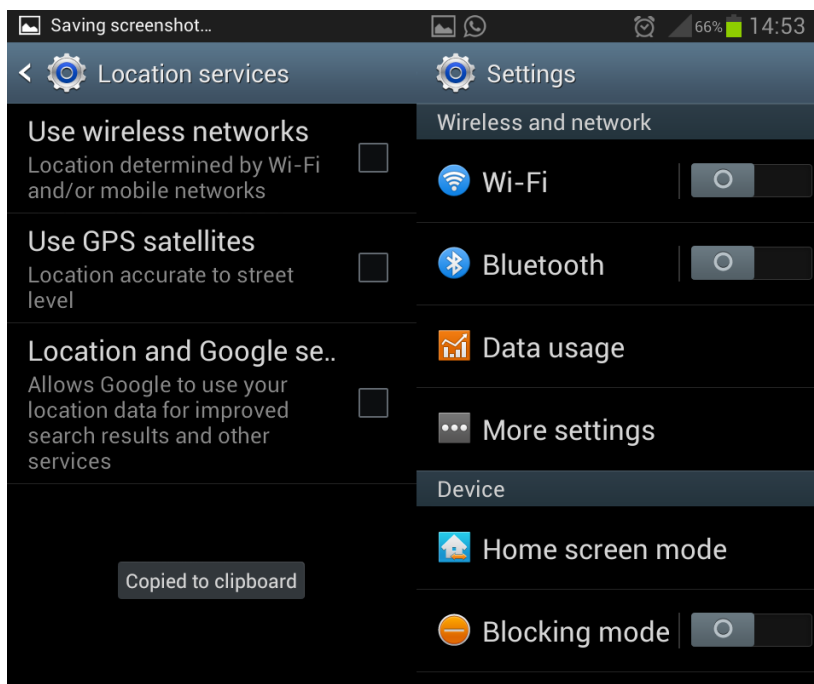
Γ) Explore

Η αναζήτηση των κοντινότερων αποστολών και η εμφάνιση τους στον χάρτη πραγματοποιείται επιλέγοντας το κουμπί  explore. Πατώντας το συγκεκριμένο κουμπί πυροδοτούνται κάποιοι απαραίτητοι έλεγχοι αν οι υπηρεσίες που χρειάζεται η φόρμα με τον χάρτη να λειτουργήσει είναι ενεργοποιημένες. Αυτές περιλαμβάνουν έλεγχο για τρέχουσα ενεργοποιημένη σύνδεση στο διαδίκτυο, έλεγχο για ενεργοποιημένη υπηρεσία υπολογισμού τοποθεσίας και τέλος, έλεγχο για την διαθεσιμότητα των υπηρεσιών της Google, μιας και ο χάρτης για να εμφανιστεί θα πρέπει οι διακομιστές που μας προσφέρουν τις πληροφορίες να λειτουργούν.



Εικόνα 32 Μηνύματα σε περίπτωση μη ενεργοποιημένων υπηρεσιών

Σε διαφορετική περίπτωση εμφανίζονται προειδοποιήσεις στον χρήστη που τον προτρέπει να ενεργοποιήσει τις υπηρεσίες που δε λειτουργούν, και στην συνέχεια να επιστρέψει στην εφαρμογή και να συνεχίσει κανονικά την χρήση της.



Εικόνα 33 Παραπομπή χρήστη στις ρυθμίσεις της συσκευής

Δ) myHistory

Για να εμφανίσουμε τις ολοκληρωμένες αποστολές και τα αντίστοιχα βραβεία τους εκτελούμε πάλι τον έλεγχο για προσβασιμότητα στο διαδίκτυο όπως είδαμε παραπάνω και στην συνέχεια φορτώνεται η λίστα με τις ολοκληρωμένες αποστολές.



Αυτές ήταν οι 4 λειτουργίες την κεντρική φόρμας – μενού, σχεδιασμένες να μετατρέψουν την πρώτη επαφή με την εφαρμογή μια απλή και εύκολη διαδικασία. Παρακάτω θα δούμε τις 2 επιπλέον φόρμες που υπάρχουν και φορτώνονται πατώντας το αντίστοιχο κουμπί από το κεντρικό μενού όπως είδαμε παραπάνω.

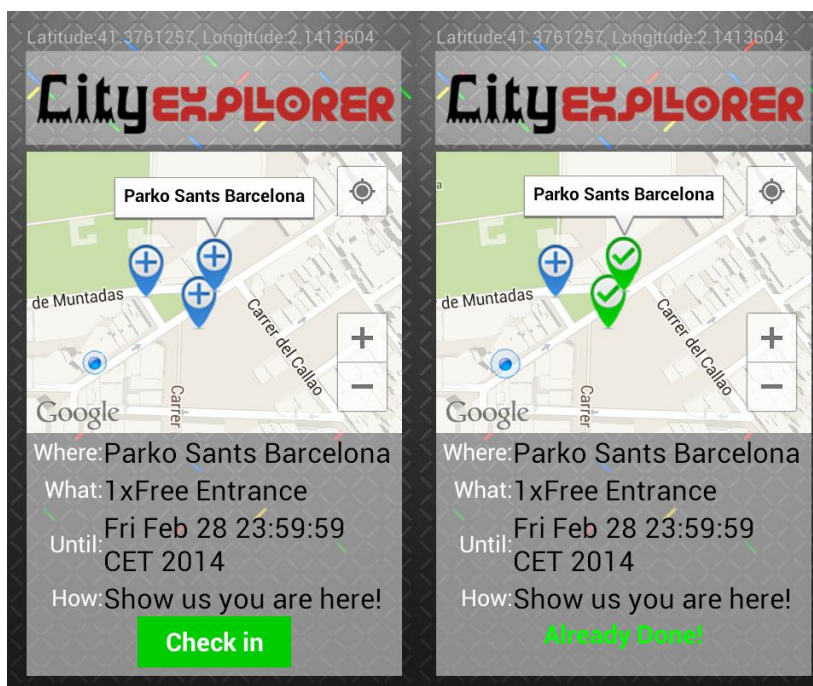
5.2.3 Εμφάνιση και εκπλήρωση αποστολών στον χάρτη

Αν μετά από τους ελέγχους διαπιστωθεί ότι όλες οι απαραίτητες υπηρεσίες είναι ενεργοποιημένες και λειτουργούν κανονικά τότε φορτώνεται η συγκεκριμένη φόρμα. Εδώ, εμφανίζονται στον χάρτη όλες οι αποστολές που υπάρχουν στην τριγύρω περιοχή, αφού πρώτα έχει υπολογιστεί η τρέχουσα τοποθεσία χρησιμοποιώντας τις τεχνικές που αναλύσαμε στο προηγούμενο κεφάλαιο.

Πατώντας ένα από τα σημάδια που βρίσκονται πάνω στον χάρτη, φορτώνονται κάτω από αυτόν οι πληροφορίες της επιλεγμένης αποστολής από τον χρήστη. Αν

Γκούτζος Άρης

η αποστολή επισημαίνεται με το σημάδι  τότε είναι διαθέσιμη για ολοκλήρωση εμφανίζεται ο τρόπος με τον οποίο μπορούμε να την ολοκληρώσουμε. Σε διαφορετική περίπτωση η αποστολή θα είναι ήδη ολοκληρωμένη, θα φέρει το σημάδι , και επιλέγοντας την θα μας εμφανίσει το αντίστοιχο μήνυμα.

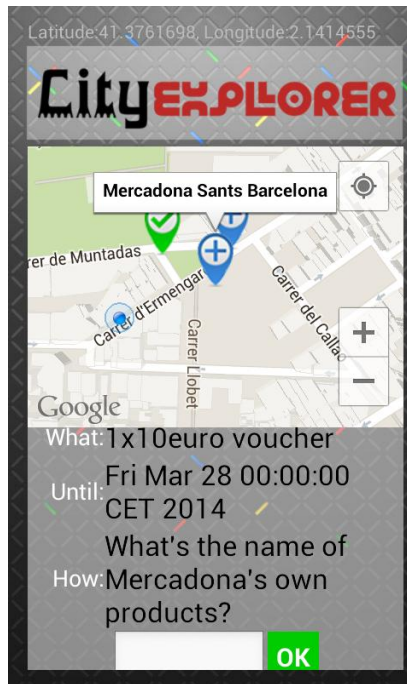


Εικόνα 34 Εμφάνιση πληροφοριών επιλεγμένης αποστολής

Επίσης επιλέγοντας μια αποστολή, εμφανίζονται κάποιες πληροφορίες όπως το όνομα της τοποθεσίας που βρίσκεται, το βραβείο που κερδίζεις ολοκληρώνοντας την, μέχρι τότε είναι ενεργή η αποστολή και με ποιόν τρόπο μπορείς να την ολοκληρώσεις. Στο τελευταίο ερώτημα έχουμε προσθέσει ενδεικτικά 2 τρόπους, την σωστή απάντηση σε κάποια ερώτηση και την δήλωση παρουσίας στην συγκεκριμένη τοποθεσία (CheckIn).

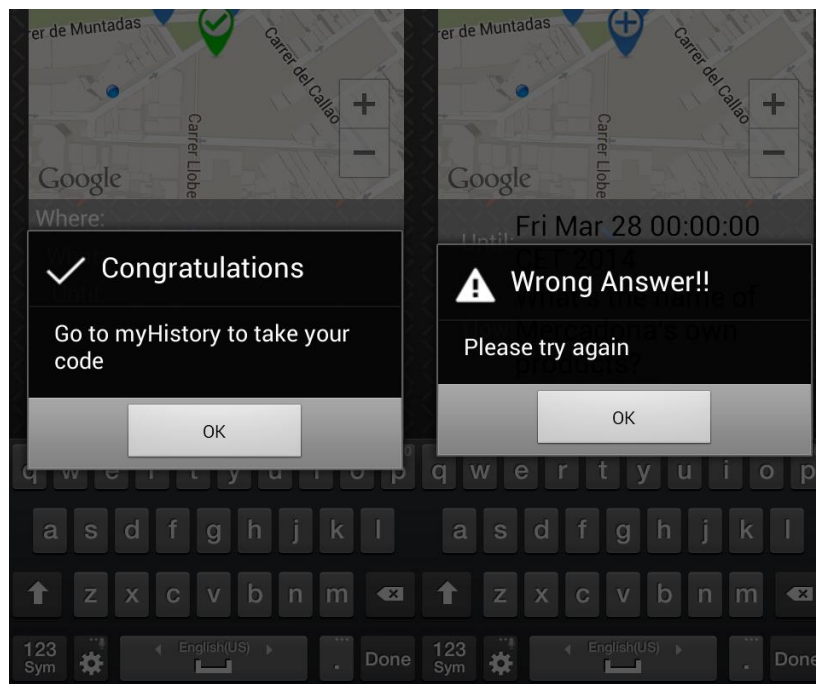
Στην πρώτη περίπτωση όπου ο χρήστης θα πρέπει να απαντήσει σωστά σε κάποιο ερώτημα, διασταυρώνεται η απάντηση που έχει εισάγει στο αντίστοιχο πεδίο που εμφανίζεται κάτω στην οθόνη της συσκευής του αφού πρώτα την έχει υποβάλει πατώντας το κουμπί 'OK' που υπάρχει δίπλα από αυτό το πεδίο.

Γκούτζος Άρης



Εικόνα 35 ολοκληρώνοντας μια αποστολή

Ανάλογα με την απάντηση, αν αυτή είναι σωστή ή όχι, εμφανίζεται το αντίστοιχο μήνυμα. Επιπλέον σε περίπτωση σωστής απάντησης το σημάδι της αποστολής αλλάζει αυτόματα σε πράσινο αφού πλέον η συγκεκριμένη αποστολή για τον χρήστη είναι ολοκληρωμένη και έχει καταγραφεί στο σύστημα.



Εικόνα 36 Επιτυχία ή αποτυχία ολοκλήρωσης αποστολής

5.2.4 Εμφάνιση ιστορικού και βραβείων

Όπως και προηγουμένως, μετά την ολοκλήρωση των απαραίτητων ελέγχων για ενεργοποιημένες υπηρεσίες φορτώνεται η λίστα με τις ολοκληρωμένες αποστολές. Η διαφορά με τους ελέγχους που γίνονται για την φόρμα με τους χάρτες, είναι ότι εδώ χρειαζόμαστε μόνο πρόσβαση στο διαδίκτυο για την ανάκτηση των απαραίτητων πληροφοριών.



Εικόνα 37 Λίστα ολοκληρωμένων αποστολών και πληροφορίες βραβείου

Όπως βλέπουμε στην παραπάνω εικόνα, αριστερά φαίνεται η λίστα με τις ολοκληρωμένες αποστολές του συγκεκριμένου χρήστη που έχει εισέλθει στο σύστημα. Επιλέγοντας κάποια από αυτές, φορτώνει νέα φόρμα που περιγράφει το βραβείο και την ποσότητά του που κερδίζει ο χρήστης εξαργυρώνοντας τον κωδικό που αναγράφεται ακριβώς από κάτω ή επιτρέποντας στον αντίστοιχο υπεύθυνο να σκανάρει το παραγόμενο QRcode που παράγεται όπως περιγράψαμε στο προηγούμενο κεφάλαιο (4.5.6).

5.3 Επίλογος

Σε αυτό το κεφάλαιο δείξαμε το λειτουργικό κομμάτι της εφαρμογής. Αναλύσαμε τον τρόπο που προκύπτουν τα δεδομένα και το πως συνεργάζονται για την τελική έκβαση του αποτελέσματος. Το σκεπτικό της εφαρμογής CityExplorer είναι απλό. Για να το υλοποιήσουμε όμως, απαιτούνται πιο περίπλοκες διαδικασίες που εκτελούνται με άγνοια του χρήστη και έχουν σκοπό την γρήγορη και ποιοτική

εξυπηρέτηση του. Η λειτουργία της είναι επικεντρωμένη στις αδυναμίες του χρήστη και της συσκευής, παρέχοντας πολλούς εξειδικευμένους ελέγχους για πιθανά ατοπήματα και ενημερώνοντάς τον με τα κατάλληλα μηνύματα γίνεται προσπάθεια αποφυγής τους. Επίσης, η εφαρμογή σχεδιάστηκε με κατάλληλο τρόπο έτσι ώστε με απλά βήματα και ελάχιστη αλληλεπίδραση με τον χρήστη να επιτευχθεί το ζητούμενο αποτέλεσμα. Κάθε κάτοχος συσκευής με πλατφόρμα Android μπορεί να κάνει χρήση της εφαρμογής ώστε να εκμεταλλευτεί τα πλεονεκτήματα που προσφέρει.

6. Συμπεράσματα και περιθώρια ανάπτυξης

Το βασικό κομμάτι της πτυχιακής ήταν η δημιουργία μιας εφαρμογής για την πλατφόρμα Android που σκοπό έχει την χρησιμοποίηση διαφορετικών τεχνολογιών και τον συνδυασμό αυτών. Για την ολοκλήρωση αυτού του έργου απαιτούνται γνώσεις που αφορούν την ανάπτυξη εφαρμογών για συσκευές Android, την υλοποίηση προγραμμάτων που εκτελούνται στον σέρβερ και την μετατροπή των δεδομένων που αποθηκεύονται ή ανακτώνται από την βάση δεδομένων. Οι βασικές γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για την εφαρμογή είναι η Java και η XML και για τα προγράμματα στον σέρβερ η PHP, η MySQL και η JSON. Οι περισσότερες από αυτές τις γλώσσες διδάσκονται στο τμήμα Πληροφορικής, όμως στα πλαίσια της δημιουργίας μιας εφαρμογής για την πλατφόρμα Android η μέθοδος που χρησιμοποιείται είναι αρκετά διαφορετική από τον τρόπο που διδαχθήκαμε. Εξαιρετική πηγή πληροφόρησης είναι το διαδίκτυο καθώς κάθε σημαντικό βήμα για την επίλυση αποριών γίνεται με τις κατάλληλες αναζητήσεις σε ιστοτόπους που εξειδικεύονται σε αυτού του είδους τα θέματα. Επίσης τα περιβάλλοντα ανάπτυξης του κώδικα βοήθησαν πολύ καθώς με την χρήση συγκεκριμένων εργαλείων γίνεται ευκολότερη και απλοποιείται η δουλειά του προγραμματιστή.

Η εφαρμογή CityExplorer είναι μια αρχή για την υλοποίηση ενός προγράμματος που εμπεριέχει όλες αυτές τις τεχνολογίες και ο συνδυασμός τους σε αυτό το έργο προσδίδει μεγάλη λειτουργικότητα και ευκολία. Οι περισσότερες συσκευές που κυκλοφορούν στο εμπόριο περιέχουν όλα εκείνα τα στοιχεία που απαιτούνται για την λειτουργία της εφαρμογής, όπως δέκτη GPS και σύνδεση στο διαδίκτυο. Η απόφαση να αναπτυχθεί σε πλατφόρμα Android είναι δικαιολογημένη καθώς τα τελευταία χρόνια αυξάνονται ραγδαία οι χρήστες που χρησιμοποιούν έξυπνα κινητά με αυτό το λειτουργικό σύστημα. Αυτό όμως συνεπάγεται ότι θα υπάρχει και ένας σχετικά μεγάλος αριθμός χρηστών που δε θα μπορεί να κάνει χρήση της, καθώς είναι κάτοχοι συσκευών που λειτουργούν σε διαφορετικές πλατφόρμες (πχ iOS, BlackBerryOS, Windows Phone).

Η προσπάθεια μας αυτή ολοκλήρωσε το κομμάτι της εφαρμογής που χρησιμοποιούν οι τελικοί χρήστες στις φορητές τους συσκευές και τα δεδομένα που χρησιμοποιήθηκαν δημιουργήθηκαν για δοκιμαστικούς σκοπούς. Για να γίνει

πλήρως λειτουργική η συγκεκριμένη ιδέα θα πρέπει να αναπτυχθεί και μια διαδικτυακή εφαρμογή που θα δίνει πρόσβαση στην άλλη κατηγορία χρηστών, τους διαχειριστές των τοποθεσιών και των αποστολών. Η εφαρμογή αυτή θα επιτρέπει όλους εκείνους τους ενδιαφερόμενους, ιδιοκτήτες καταστημάτων, ιδιωτικές επιχειρήσεις, δημόσιους φορείς κ.α να προσφέρουν αποστολές και βραβεία στους καταναλωτές, πολίτες κ.τ.λ. που σκοπό θα έχουν την διαφήμιση τους, την ευχαρίστηση των πελατών τους και την προσφορά υπηρεσιών με μορφή βραβείων προς τους πολίτες.

Η λειτουργία της θα είναι παρόμοια με αυτήν του τελικού χρήστη, με την διαφορά ότι θα είναι μια διαδικτυακή εφαρμογή και θα είναι προσβάσιμη από τους φυλλομετρητές (browsers). Οι χρήστες αυτοί θα πρέπει να δημιουργήσουν έναν λογαριασμό δηλώνοντας κάποια στοιχεία όπως όνομα χρήστη, κωδικό πρόσβασης, γεωγραφικό μήκος και πλάτος της τοποθεσίας που είναι υπεύθυνοι, όνομα αυτής της τοποθεσίας και σε περίπτωση που θελήσουν να προσθέσουν κάποια αποστολή θα πρέπει να εισάγουν όλα εκείνα τα απαραίτητα στοιχεία της αποστολής. Η εισαγωγή της τοποθεσίας θα γίνεται πάλι με χρήση των Google Maps ειδικά τροποποιημένων για την συγκεκριμένη λειτουργία. Έπειτα μετά την επιτυχημένη είσοδο στο σύστημα με την εισαγωγή του ονόματος χρήστη και του κωδικού πρόσβασης θα γίνεται διαχείριση των ολοκληρωμένων αποστολών, δηλαδή θα υπάρχει η δυνατότητα εξαργύρωσης των κουπονιών κάποιων χρηστών, εισάγοντας απλά στο σύστημα το όνομα χρήστη και τον ειδικό κωδικό του βραβείου ή απλά σκανάρωντας το QRcode το οποίο μας δίνει απευθείας αυτές τις 2 πληροφορίες. Η όλη υπηρεσία θα είναι πολύ εύκολα προσβάσιμη από οποιονδήποτε διαθέτει ηλεκτρονικό υπολογιστή ανεξαρτήτου έκδοσης ή τύπου λειτουργικού συστήματος που θα φέρουν.

Για να θεωρηθεί πετυχημένο το εν λόγω εγχείρημα, η ανάπτυξη της εφαρμογής για συσκευές Android, χρησιμοποιήθηκαν οι πιο διάσημες και προσιτές τεχνολογίες της εποχής μας, η πλατφόρμα Android και τα Google Maps. Το αποτέλεσμα ελπίζω να δώσει έναυσμα ώστε να συνεχιστεί η περαιτέρω ανάπτυξη της και με την χρήση περισσότερων τεχνολογιών και γνώσεων να δημιουργηθεί ένα ολοκληρωμένο αποτέλεσμα που θα συνδυάζει τεχνολογίες και υπηρεσίες προσβάσιμες από όλους τους τύπους συσκευών με πρόσβαση στο διαδίκτυο και λειτουργικά συστήματα.

Βιβλιογραφία και αναφορές

1. Πτυχιακή εργασία του Κλιάφα Θεόδωρου
2. Jerome (J.F.) DiMazio 2006, Android:A programmer's guide, <http://iraqandroid.files.wordpress.com/2013/09/android-a-programmers-guide.pdf>
3. https://developers.google.com/maps/faq?hl=el&cs=1#tos_commercial
4. <http://google-latlong.blogspot.com.es/2010/02/introducing-google-maps-labs-your.html>
5. <http://www.android.com/about/>
6. [http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))
7. <https://developers.google.com/maps/>
8. http://en.wikipedia.org/wiki/Google_Maps
9. CheckIn Android Application. <https://play.google.com/store/apps/details?id=com.dkeesto.fbcheckin>
10. Foursquare Android Application. <https://foursquare.com/about>
11. TripAdvisor <http://www.tripadvisor.com.gr/apps-itamobile-candroid>