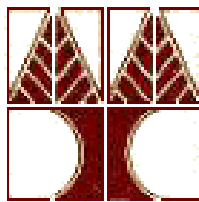


Ατομική Διπλωματική Εργασία

**ΑΝΑΠΤΥΞΗ ANDROID ΕΦΑΡΜΟΓΗΣ ΓΙΑ ΤΗΝ
ΔΡΟΜΟΛΟΓΗΣΗ ΤΩΝ ΛΕΩΦΟΡΕΙΩΝ ΤΗΣ ΚΥΠΡΟΥ**

Αντωνία Μαυρόχαννα

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2014

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάπτυξη Android εφαρμογής για την δρομολόγηση των λεωφορείων της Κύπρου

Αντωνία Μαυρόχαννα

Επιβλέπων Καθηγητής
Παρασκευάς Ευριπίδου

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

Μάιος 2014

Ευχαριστίες

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

Πρώτιστα, ένα μεγάλο ευχαριστώ θα ήθελα να εκφράσω στο επιβλέπων καθηγητή μου Δρ. Παρασκευά Ευριπίδου, που μου έδωσε την ευκαιρία να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα το οποίο μου πρόσφερε αρκετές γνώσεις που θα μου είναι χρήσιμες στο μέλλον. Επίσης, τον ευχαριστώ για τον χρόνο που διέθετε για μένα αφού μέσω των συχνών μας συναντήσεων και των συμβουλών που μου έδινε σε αυτές, κατάφερα να καθοδηγηθώ στην ολοκλήρωση της διπλωματικής μου εργασίας με επιτυχία.

Επιπρόσθετα, θερμές ευχαριστίες θα ήθελα να δώσω στην οικογένεια μου και στους φίλους μου για την συμπαράσταση τους και την βοήθεια τους κατά την διάρκεια φοίτησης μου στο Πανεπιστήμιο Κύπρου. Τέλος, αποτελεί ήσσονος σημασίας για μένα η ηθική και οικονομική υποστήριξη του πατέρα μου καθ' όλη την διάρκεια των σπουδών μου και γι' αυτό ένα ιδιαίτερο ευχαριστώ θα ήθελα να το χαρίσω σε αυτόν.

Περίληψη

Το θέμα της Διπλωματικής μου Εργασίας είναι η ανάπτυξη μίας εφαρμογής για κινητές συσκευές που χρησιμοποιούν λειτουργικό σύστημα Android με σκοπό την δρομολόγηση των λεωφορείων της Κύπρου και την εμφάνιση σχετικών πληροφοριών σε χρήστες της εφαρμογής. Οι χρήστες αυτοί μπορεί να είναι είτε κάτοικοι της Κύπρου, είτε τουρίστες που έρχονται για να επισκεφθούν το ωραίο μας νησί και επιθυμούν να διευκολυνθούν με την μετακίνησή τους στις διάφορες πόλεις και περιοχές μέσω λεωφορείων.

Η εφαρμογή χωρίζεται σε τρία μέρη. Αρχικά, στο πρώτο μέρος ο χρήστης έχει την δυνατότητα να συμπληρώσει σε μία φόρμα τα στοιχεία της διαδρομής που επιθυμεί να πάρει. Πιο συγκεκριμένα, χρειάζεται να προσθέσει την αρχική διεύθυνση στην οποία θα βρίσκεται για να πάρει το λεωφορείο ή να αφήσει την προεπιλεγμένη διεύθυνση που είναι η τοποθεσία στην οποία βρίσκεται αυτήν την στιγμή και να προσθέσει την τελική διεύθυνση στην οποία θέλει να πάει. Έπειτα, μπορεί να επιλέξει την ώρα και ημερομηνία που επιθυμεί για την διαδρομή του μέσω ενός ρολογιού και ενός ημερολογίου αντίστοιχα. Μετά την συμπλήρωση των στοιχείων αυτών, θα μπορεί να προχωρήσει στο δεύτερο μέρος στο οποίο θα μπορεί να δει ξανά τις πληροφορίες που έχει βάλει προηγουμένως και κάποια βασικά στοιχεία της διαδρομής που είναι η διάρκεια και οι στάσεις που ακολουθεί. Τέλος, αφού επιλέξει την διαδρομή οδηγείται στο τρίτο μέρος στο οποίο θα λάβει περισσότερες λεπτομέρειες σχετικά με την διαδρομή αφού θα μπορεί να δει το χρονοδιάγραμμα της και αυτήν σχεδιασμένη στο χάρτη και να λάβει περισσότερες πληροφορίες μέσω της ιστοσελίδας «CyprusByBus» και του τηλεφώνου για γενικές πληροφορίες λεωφορείων που παρέχονται από την εφαρμογή.

Μετά από την ολοκλήρωση της υλοποίησης της, η εφαρμογή δόθηκε για δοκιμή και αξιολόγηση σε πιθανούς χρήστες. Οι χρήστες είχαν εκφράσει μίαν αρκετά θετική αντίδραση σχετικά με την γενική ιδέα και την ευκολία χρήσης της εφαρμογής και μου είχαν αναφέρει ότι θα είναι σίγουρα οι πρώτοι που θα την χρησιμοποιήσουν όταν κοινοποιηθεί. Επίσης, μου είχαν επισημάνει κάποιες μικρές προεκτάσεις που θα ήταν καλό να γίνουν στο μέλλον για βελτίωση της εφαρμογής.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	1
1.1	Σκοπός Διπλωματικής Εργασίας	1
1.2	Εμβέλεια λογισμικού	2
1.3	Κύρια Χαρακτηριστικά Εφαρμογής	3
1.4	Λειτουργικό Σύστημα Android	4
1.5	Οργάνωση κειμένου	7
Κεφάλαιο 2	Ανάλυση και Σχεδίαση Προβλήματος.....	8
2.1	Καταγραφή και Ανάλυση Απαιτήσεων	8
2.1.1	Περιορισμοί και Υποθέσεις εφαρμογής	9
2.1.2	Απαιτήσεις Επίδοσης	10
2.1.3	Δευτερεύουσες απαιτήσεις	10
2.1.4	Χαρακτηριστικά Συστήματος Λογισμικού	10
2.1.4.1	Αξιοπιστία	10
2.1.4.2	Διαθεσιμότητα	11
2.1.4.3	Ασφάλεια	11
2.1.4.4	Συντήρηση	11
2.1.4.5	Μεταφερσιμότητα	12
2.2	Βασικές λειτουργίες Χρήστη	12
2.3	Use Cases	13
2.3.1	Actors	13
2.3.1.1	Actors Descriptions	13
2.3.2	Use Case Diagram	14
2.4	Ανάλυση και Σχεδίαση Βάσης Δεδομένων	15
2.5	Αλγόριθμος Υλοποίησης	17
Κεφάλαιο 3	Υλοποίηση Εφαρμογής.....	21
3.1	Ανατομία μιας εφαρμογής Android	21
3.2	Γλώσσα Προγραμματισμού Java	28
3.3	SDK Manager	30
3.4	Γλώσσα Προγραμματισμού C#	31

Κεφάλαιο 4 Εγχειρίδιο χρήσης Εφαρμογής.....	33
4.1 Αρχική Οθόνη	33
4.1.1 Αλλαγή αρχικής τοποθεσίας	35
4.1.2 Αλλαγή τελικής τοποθεσίας	36
4.1.3 Αλλαγή ημερομηνίας	37
4.1.4 Αλλαγή ώρας	38
4.1.5 Εκτέλεση Αλγορίθμου	39
4.2 Οθόνη Επιλογής Διαδρομής	39
4.3 Οθόνη Λεπτομερειών Συγκεκριμένης Διαδρομής	40
4.3.1 Χρονοδιάγραμμα Διαδρομής	40
4.3.2 Χάρτης Διαδρομής	41
4.3.3 Ενημέρωση για διαδρομή από άλλες πηγές	41
 Κεφάλαιο 5 Δοκιμή και Αξιολόγηση Εφαρμογής	43
5.1 Δοκιμή Εφαρμογής	43
5.2 Δημιουργία Ερωτηματολογίου	44
5.3 Ανάλυση Αποτελεσμάτων	46
 Κεφάλαιο 6 Συμπεράσματα	48
6.1 Συμπεράσματα	48
6.2 Μελλοντικές Προεκτάσεις	50
 Β ι β λ ι ο γ ρ α φ ί α	52

Κεφάλαιο 1

Εισαγωγή

1.1 Σκοπός Διπλωματικής Εργασίας	1
1.2 Εμβέλεια λογισμικού	2
1.3 Κύρια Χαρακτηριστικά Εφαρμογής	3
1.4 Λειτουργικό Σύστημα Android	4
1.3 Οργάνωση Κειμένου	7

Σε αυτό το κεφάλαιο γίνεται μια μικρή εισαγωγή στην παρούσα διπλωματική. Πιο συγκεκριμένα, περιγράφεται ο κύριος στόχος ανάπτυξης του συστήματος αυτού και η εμβέλεια του. Στην συνέχεια, γίνεται μια σύντομη περιγραφή της οργάνωσης του κειμένου.

1.1 Σκοπός Διπλωματικής Εργασίας

Σκοπός της διπλωματικής μου εργασίας, είναι η ανάπτυξη μιας εφαρμογής για κινητές συσκευές που χρησιμοποιούν λειτουργικό σύστημα Android, η οποία θα επιτρέπει στους χρήστες του συστήματος να μπορούν να δουν και να υπολογίσουν μια διαδρομή που επιθυμούν να πραγματοποιήσουν με τα δημόσια και περιφερειακά λεωφορεία της Κύπρου. Κύριος στόχος της εφαρμογής είναι να διευκολύνει τους πολίτες της Κύπρου καθώς επίσης και τους τουρίστες που έρχονται να επισκεφθούν το ωραίο νησί στο να παίρνουν το κατάλληλο λεωφορείο για την μετακίνηση τους στην Κύπρο. Μέσω αυτού του συστήματος, οι χρήστες θα μπορούν να βρίσκουν την κοντινότερη στάση και το γρηγορότερο λεωφορείο που θα τους μεταφέρει στον προορισμό τους.

Μετά από την διεκπεραίωση μερικών συναντήσεων με τον κύριο Ευριπίδου εξακρίβωσα τις ανάγκες που έχουν οι διάφοροι χρήστες του πιο πάνω συστήματος και με βάση τις ανάγκες τους έχει γίνει το έγγραφο καταγραφής απαιτήσεων και επακόλουθα η επιτυχής υλοποίηση του πιο πάνω συστήματος.

1.2 Εμβέλεια λογισμικού

Το λογισμικό το οποίο έχει υλοποιηθεί επιτρέπει στους χρήστες του συστήματος να ενημερώνονται για διαδρομές που επιθυμούν να πραγματοποιήσουν, αφού προσθέσουν τα στοιχεία της διαδρομής με το άνοιγμα της εφαρμογής. Οι χρήστες θα μπορούν να δουν την διαδρομή στον χάρτη και να πάρουν περισσότερες πληροφορίες από το χρονοδιάγραμμα της. Επίσης, θα έχουν την δυνατότητα να πάρουν περισσότερες πληροφορίες από μία ιστοσελίδα και ένα τηλέφωνο γενικών πληροφοριών για λεωφορεία που παρέχονται από την εφαρμογή. Δεν χρειάζεται να κάνουν οποιαδήποτε εγγραφή στο σύστημα και να τους παρέχονται συνθηματικά αφού το σύστημα είναι ανοιχτό για οποιονδήποτε επιθυμεί να το χρησιμοποιήσει.

Η κύρια λειτουργία του συστήματος υποστηρίζεται από μια βάση δεδομένων στο back office του συστήματος και interfaces που θα εφαρμόζονται σε κινητές συσκευές. Το σύστημα αποθήκευσης και διαχείρισης δεδομένων των χρηστών είναι υπεύθυνο για την εισαγωγή όλων των στάσεων της Λευκωσίας, όλων των λεωφορείων και τις μέρες και ώρες που περνά το καθένα και την εισαγωγή όλων των πληροφοριών σχετικά με όλες τις άμεσες διαδρομές μεταξύ δύο κόμβων(ονόματα κόμβων, απόσταση μεταξύ τους και λεωφορείο που την πραγματοποιεί). Σε μεταγενέστερο στάδιο πριν την εγκατάσταση του λογισμικού θα πρέπει να προστεθούν τα πιο πάνω για όλη την Κύπρο. Έτσι ο χρήστης θα μπορεί να λάβει πληροφορίες και για δρομολόγια εκτός Λευκωσίας.

Με την δημιουργία της Βάσης Δεδομένων έχουμε τα ακόλουθα πλεονεκτήματα:

- Γρηγορότερη και ευκολότερη πρόσβαση στα δεδομένα.
- Ευκολότερη διαχείριση δεδομένων.
- Ταυτόχρονη χρήση των δεδομένων του συστήματος από τους χρήστες.

Η διεπαφή του χρήστη θα περιλαμβάνει τις φόρμες εισαγωγής και παρουσίασης δεδομένων της διαδρομής και κάποιων λεπτομερειών που την αφορούν.

1.3 Κύρια Χαρακτηριστικά Εφαρμογής

Αναγκαιότητα της Εφαρμογής:

Σε ένα τέτοιο περιβάλλον, που ζούμε, ο τουρισμός, φαντάζει ως αχτίδα ελπίδας και οξυγόνο για την οικονομία του τόπου. Εκατομμύρια τουρίστες έρχονται κάθε χρόνο στο νησί για να δουν τους μαγευτικούς του χώρους και να κάνουν τις διακοπές τους όσο πιο εύκολα και όμορφα μπορούν. Όλα αυτά, σε συνδυασμό με τη ραγδαία ανάπτυξη που γνωρίζουν τα «έξυπνα» τηλέφωνα και ιδιαίτερα το λειτουργικό σύστημα Android, συνθέτουν ένα ιδανικό σκηνικό για την ανάπτυξη εφαρμογών που έχουν στο επίκεντρό τους τόσο τον τουρισμό, όσο και την προσφορά στους κατοίκους της ίδιας της χώρας για μια πιο απλή και εύκολη ζωή. Έτσι, μια εφαρμογή που θα περιλαμβάνει όλα τα δρομολόγια των λεωφορείων και θα διευκολύνει τους πιο πάνω χρήστες να κινούνται γρήγορα και εύκολα μέσω λεωφορείων στις διάφορες περιοχές της Κύπρου οποιαδήποτε μέρα και ώρα προτιμούν είναι πάντοτε χρήσιμη.

Προσδιορισμός Χρηστών:

Η εφαρμογή απευθύνεται σε οποιονδήποτε επιθυμεί να χρησιμοποιήσει λεωφορεία κατά την διαμονή του στην Κύπρο και ψάχνει ένα γρήγορο, εύκολο και ασφαλές τρόπο να διακινείται με αυτά στις διάφορες περιοχές της.

Που θα χρησιμοποιηθεί η εφαρμογή και τι αναμένεται από αυτή:

Η εφαρμογή θα μπορεί να χρησιμοποιηθεί σε οποιοδήποτε χώρο αρκεί να υπάρχει πρόσβαση στο διαδίκτυο για να μπορεί να βρίσκει μέσα από τους χάρτες της Google την περιοχή που βρίσκεται ο χρήστης και αυτήν που θέλει να πάει και να τις εμφανίζει. Επίσης η εφαρμογή αναμένεται να είναι εύκολη και απλή στην χρήση, με χαμηλή συχνότητα σφαλμάτων. Τέλος, η εφαρμογή θα πρέπει να βρίσκει την διαδρομή γρήγορα γιατί πιθανόν ο χρήστης να μην έχει αρκετό χρόνο στην διάθεση του.

1.4 Λειτουργικό Σύστημα Android

Το Android είναι λειτουργικό σύστημα για κινητές συσκευές με οθόνες αφής όπως κινητά τηλέφωνα και tablets, το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά αναπτύχθηκε από την Android Inc, την οποία στήριζε οικονομικά η Google και αργότερα το 2005 την αγόρασε. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google.

Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance, μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής hardware, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές κινητής τηλεφωνίας. Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού. Το λογότυπο για το λειτουργικό σύστημα Android είναι ένα ρομπότ σε χρώμα πράσινου μήλου, όπως φαίνεται στην πιο κάτω εικόνα.



Εικόνα 1.1 Λογότυπο Android 2

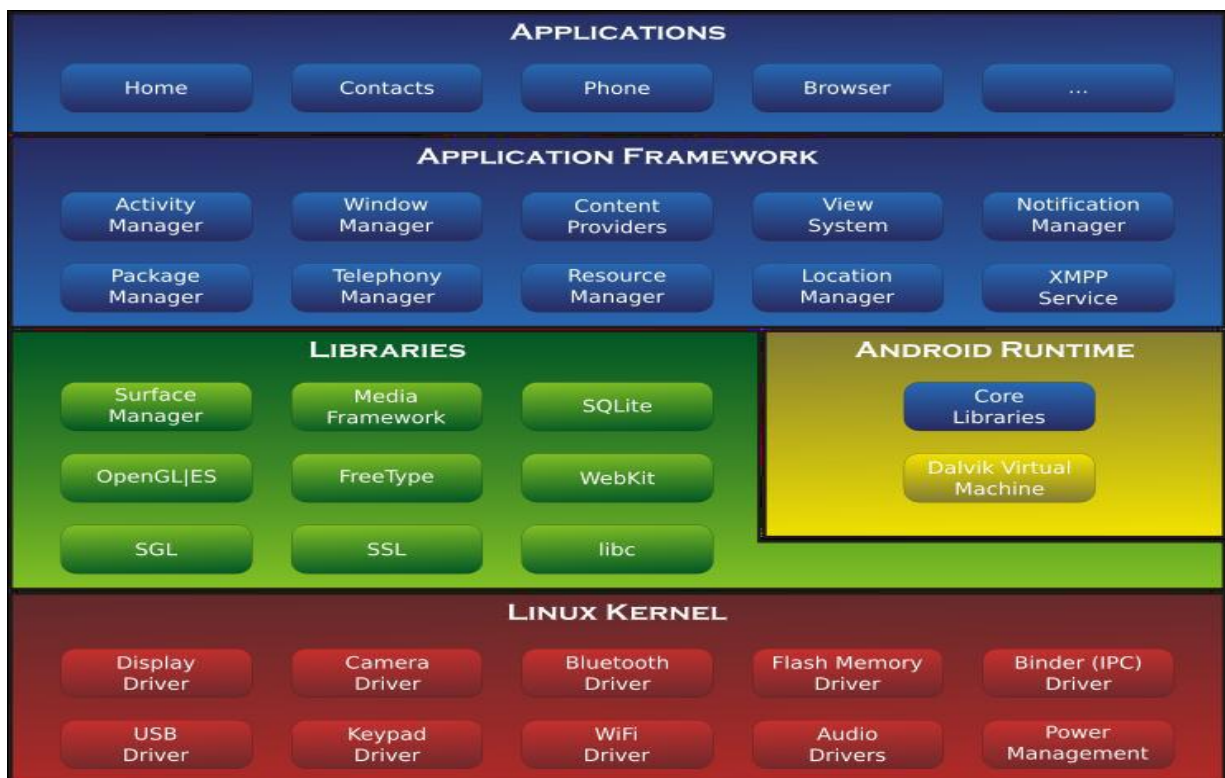
Μέχρι τώρα υπάρχουν περίπου 975,000 εφαρμογές διαθέσιμες στο Google Play, που είναι το βασικότερο κατάστημα για Android εφαρμογές. Υποστηρίζει τεχνολογίες συνδεσιμότητας όπως Wi-Fi, GPS, Bluetooth, GSM κ.α. Σε μια πρόσφατη έρευνα διαπιστώθηκε ότι το 71%

των χρηστών που έχουν Android συσκευή θα ξαναγόραζαν Android συσκευή, ακολουθούμενο από τον μεγαλύτερο ανταγωνιστή την Apple με 57%.

Αρχιτεκτονική Android

Όπως μπορείτε να δείτε στο σχήμα 1.2, στον πυρήνα της πλατφόρμας Android βρίσκεται ένα Linux kernel το οποίο είναι υπεύθυνο για τη διαχείριση των device drivers, τον έλεγχο πρόσβασης στους πόρους του συστήματος, τη διαχείριση μνήμης και τις λοιπές υπηρεσίες που παρέχει ένα λειτουργικό σύστημα. Στους device drivers συγκαταλέγονται αυτοί της οθόνης, του Wi-Fi, της κάμερας, του ήχου κ.α.

Ένα επίπεδο επάνω βρίσκονται οι native βιβλιοθήκες του συστήματος που είναι γραμμένες σε C++ και περιλαμβάνουν το OpenGL, την SQLite, την Media library κ.α. Οι εφαρμογές που τρέχουν στο κινητό μπορούν να έχουν πρόσβαση στις βιβλιοθήκες αυτές μέσω της Dalvik JVM. Όπως έχει ήδη αναφερθεί, οι εφαρμογές Android είναι γραμμένες σε Java και άρα για να τρέξουν χρειάζονται το αντίστοιχο περιβάλλον. Όπως λοιπόν για να εκτελέσουμε μία εφαρμογή σε ένα PC είναι απαραίτητο να είναι εγκατεστημένο το κατάλληλο JRE (Java Runtime Environment), για τις εφαρμογές Android τον ρόλο του JRE παίζει η Dalvik VM.



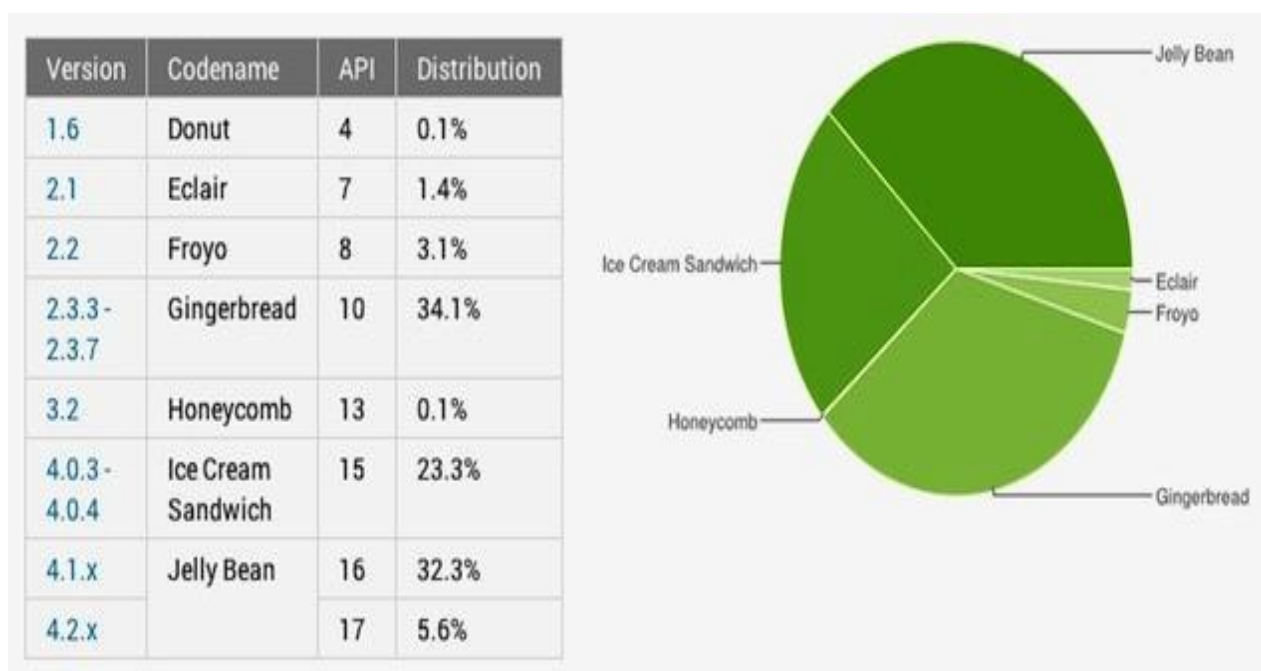
Εικόνα 1.2 Αρχιτεκτονική Android [2]

Εκδόσεις Android

Το Android εξελίσσεται με την πάροδο του χρόνου παρέχοντας νέες δυνατότητες με κάθε έκδοση. Η προτελευταία έκδοση παρουσιάστηκε στις 9 Ιουλίου 2012 και είναι το Jelly Bean και είχε κατακτήσει την πρώτη θέση σε σχέση με τις άλλες εκδόσεις φτάνοντας το 38%.

Ωστόσο τον Νοέμβριο του 2014 δημιουργήθηκε μία νέα έκδοση Android, το **Android 4.4 KitKat** που είναι εμφανώς πιο καθαρό και απλούστερο από την προηγούμενη έκδοση. Το Nexus 5 είναι το πρώτο Smartphone της αγοράς με το νέο λειτουργικό σύστημα Android 4.4 KitKat. Μία ενδιαφέρουσα αλλαγή παρατηρείται στη lock screen όταν ακούει κάποιος μουσική, καθώς θα εμφανίζει το εξώφυλλο του album και έτσι θα μπορεί να αλλάξει τραγούδι χωρίς να ξεκλειδώσει την συσκευή.

Πιο κάτω, φαίνονται καθαρά όλες οι εκδόσεις Android και οι κατανομές τους.



1.5 Οργάνωση Κειμένου

Όπως μπορείτε να δείτε και από τον πίνακα των περιεχομένων, η διπλωματική μου εργασία αναλύεται σε 6 κεφάλαια.

Στο πρώτο κεφάλαιο παρουσιάζονται ο σκοπός και η εμβέλεια του λογισμικού, κάποια κύρια χαρακτηριστικά της εφαρμογής και μια μικρή περιγραφή του λειτουργικού συστήματος android.

Στο δεύτερο κεφάλαιο γίνεται μία ανάλυση του σχεδιασμού της εφαρμογής. Καταγράφονται και αναλύονται οι απαιτήσεις του συστήματος, οι βασικές λειτουργίες χρήστη και οι περιπτώσεις χρήσης του συστήματος. Έπειτα, γίνεται ανάλυση και σχεδίαση της Βάσης Δεδομένων του συστήματος στην οποία θα καταχωρούνται όλα τα δεδομένα που έχουν σχέση με τα λεωφορεία της Κύπρου και τις στάσεις από τις οποίες περνούν. Τέλος, περιγράφεται ο αλγόριθμος εύρεσης της κοντινότερης διαδρομής, των στάσεων από τις οποίες περνά και των λεωφορείων που χρησιμοποιούνται για να πραγματοποιηθεί.

Στο τρίτο κεφάλαιο αναλύονται όλες οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής και με ποιο τρόπο. Οι τεχνολογίες αυτές είναι η γλώσσα προγραμματισμού Java, η γλώσσα σήμανσης XML, η γλώσσα προγραμματισμού C#, ο SDK Manager και ο SQL Server 2008.

Στο τέταρτο κεφάλαιο παρουσιάζεται ο οδηγός χρήσης της εφαρμογής. Η κάθε οθόνη της εφαρμογής περιγράφεται με όλα τα αποτελέσματα.

Στο πέμπτο κεφάλαιο παρουσιάζονται τα αποτελέσματα από τις δοκιμές της εφαρμογής καθώς και η ανάλυση τους και στο έκτο κεφάλαιο καταγράφονται όλα τα συμπεράσματα που εξάχθηκαν από την διπλωματική εργασία και η μελλοντική εργασία που μπορεί να γίνει σε σχέση με αυτήν.

Κεφάλαιο 2

Ανάλυση και Σχεδίαση Προβλήματος

2.1 Καταγραφή και Ανάλυση Απαιτήσεων	8
2.1.1 Περιορισμοί και Υποθέσεις εφαρμογής	9
2.1.2 Απαιτήσεις Επίδοσης	10
2.1.3 Δευτερεύουσες απαιτήσεις	10
2.1.4 Χαρακτηριστικά Συστήματος Λογισμικού	10
2.1.4.1 Αξιοπιστία	10
2.1.4.2 Διαθεσιμότητα	11
2.1.4.3 Ασφάλεια	11
2.1.4.4 Συντήρηση	11
2.1.4.5 Μεταφερσιμότητα	12
2.2 Βασικές λειτουργίες Χρήστη	12
2.3 Use Cases	13
2.3.1 Actors	13
2.3.1.1 Actors Descriptions	13
2.3.2 Use Case Diagram	14
2.4 Ανάλυση και Σχεδίαση Βάσης Δεδομένων	15
2.5 Αλγόριθμος Υλοποίησης	17

Σε αυτό το κεφάλαιο γίνεται ανάλυση και σχεδίαση του προβλήματος με βάση τις απαιτήσεις του πελάτη. Αρχικά, γίνεται μία αναφορά στις απαιτήσεις της εφαρμογής και μετά παρουσιάζονται με λεπτομέρεια οι λειτουργίες του χρήστη που φαίνονται ξεκάθαρα στα διαγράμματα που ακολουθούν. Τέλος, περιγράφεται η Βάση Δεδομένων του συστήματος και παρουσιάζεται ο αλγόριθμος που χρησιμοποιήθηκε για την εύρεση δρομολογίων.

2.1 Καταγραφή και Ανάλυση Απαιτήσεων

Πριν από την υλοποίηση της εφαρμογής, είχα πραγματοποιήσει μία συνάντηση με τον πελάτη για να συζητήσουμε και να καθορίσουμε τις απαιτήσεις της εφαρμογής με βάση τις δικές του ανάγκες και τις ανάγκες κάποιων πιθανών χρηστών που είχα συλλέξει πληροφορίες από αυτούς προηγουμένως. Η κύρια απαίτηση του συστήματος ήταν να μπορεί ο χρήστης δίνοντας κάποια συγκεκριμένα στοιχεία μιας διαδρομής που θα ήθελε να πραγματοποιήσει να του εμφανίζεται (αν υπάρχει) η κοντινότερη διαδρομή που πληρεί αυτά τα στοιχεία. Αν δεν υπάρχει να του εμφανίζεται μήνυμα ότι δεν υπάρχει.

Στην συνέχεια τόσο ο πελάτης όσο και οι μερικοί πιθανοί χρήστες που επέλεξα για να λάβω τις απαιτήσεις τους σχετικά με την εφαρμογή μου είχαν επισημάνει ότι καλό θα ήταν να δίνονται και κάποιες επιπλέον πληροφορίες σχετικά με την διαδρομή που να κάνουν την εφαρμογή πιο εύχρηστη και πιο επιθυμητή σε όλους. Αυτό που ζήτησαν ήταν να εμφανίζεται το χρονοδιάγραμμα της συγκεκριμένης διαδρομής που έχει επιλέξει ο χρήστης το οποίο θα εξηγεί με μία σειρά από βήματα τι πρέπει να κάνει ο χρήστης για να φτάσει στον προορισμό του και επίσης να εμφανίζεται η διαδρομή αυτή στον χάρτη για να κάνει τον χρήστη να κατανοήσει καλύτερα το πού θα βρίσκεται. Τέλος, κύρια επιθυμία του πελάτη ήταν να μπορεί ο χρήστης να τυπώνει την διαδρομή του για να μπορεί να την ξαναδεί και αργότερα χωρίς να χρειάζεται να ενωθεί ξανά στο διαδίκτυο.

Αφού κατέγραψα και μελέτησα καλά τις απαιτήσεις που μου είχαν ζητήσει οι διάφοροι χρήστες και ο πελάτης, δημιούργησα τα σχεδιαγράμματα που ακολουθούν σε επόμενο υποκεφάλαιο και δείχνουν πιο ξεκάθαρα τα πιο πάνω.

2.1.1 Περιορισμοί και Υποθέσεις Εφαρμογής:

Το λογισμικό θα γίνει με σχεσιακή βάση δεδομένων, και θα υλοποιηθεί σε SQL Server. Η γραφική διεπαφή του χρήστη (graphical user interface) θα υλοποιηθεί σε Android. Όλοι οι χρήστες που θα έχουν πρόσβαση στο σύστημα δεν μπορούν να έχουν οποιοδήποτε λογισμικό επιθυμούν στο κινητό τους, γιατί το σύστημά θα τρέχει μόνο σε λειτουργικό σύστημα Android. Για την εκτέλεση των λειτουργιών του συστήματος, απαιτείται σύνδεση με το διαδίκτυο, εφόσον θα είναι μια web-based εφαρμογή.

Η εφαρμογή γενικά πρέπει να είναι απλή και ο τρόπος της διαχείρισης του GUI να είναι συνεπής, όσο το δυνατό να είναι ο ίδιος σε κάθε μέγεθος οθόνης του κινητού τηλεφώνου που χρησιμοποιεί ο καθένας. Γενικά, να είναι μια εύκολη και απλή εφαρμογή χωρίς να δυσκολεύει τους χρήστες της και να τους επιβαρύνει στον χρόνο τους.

Έπειτα, το λογισμικό θα σχεδιαστεί έτσι ώστε να μπορούν εύκολα να προστεθούν μελλοντικές τροποποιήσεις σε περίπτωση που θα υπάρξουν νέες απαιτήσεις, αλλά να είναι εύκολη και η συντήρησή του. Το σύστημα θα σχεδιαστεί χρησιμοποιώντας αρχιτεκτονική η οποία θα επιτρέπει εύκολη υλοποίηση τυχών νέων απαιτήσεων, που ίσως προκύψουν μετά την υλοποίηση του έργου. Το λογισμικό θα είναι σχεδιασμένο με βάση τις πραγματικές ανάγκες του πελάτη.

2.1.2 Απαιτήσεις Επίδοσης

Οι χρήστες του λογισμικού θα είναι όλοι οι επιβάτες λεωφορείων, γι' αυτό είναι απαραίτητη η δυνατότητα ταυτόχρονης χρήσης από πολλούς χρήστες ανά πάσα στιγμή, με μέγιστο αριθμό 100.000 χρήστες.

Το λογισμικό, εφόσον θα βασίζεται σε εφαρμογή για κινητές συσκευές θα πρέπει να ανταποκρίνεται στις εντολές του χρήστη σε χρόνο λιγότερο των 2 δευτερολέπτων με μέγιστη καθυστέρηση 5 δευτερόλεπτα, έτσι ώστε να μην περιμένει μεγάλο χρονικό διάστημα μέχρι να εκτελεστεί η εντολή του.

2.1.3 Δευτερεύουσες απαιτήσεις

Όπως μου έχει αναφέρει και ο πελάτης μου μια μελλοντική απαίτηση του συστήματος θα είναι η χρήση του για όλες τις πόλεις της Κύπρου και η σύνδεση της κάθε πόλης με την άλλη μέσω περιφερειακών λεωφορείων.

Το σύστημα θα μπορούσε να κρατά κατάσταση για τις ενέργειες των χρηστών. Πιο συγκεκριμένα να δίνονται στους χρήστες κωδικοί πρόσβασης και κάθε φορά που εισέρχονται στο σύστημα να αποθηκεύονται οι διαδρομές που επιθυμούν να δουν στην Βάση Δεδομένων ούτως ώστε την επόμενη φορά που θα εισέλθουν στο σύστημα να μπορούν να βλέπουν ποιες διαδρομές είχαν επιλέξει παλιά για να μην χρειαστεί να τις επαναυπολογίσουν.

2.1.4 Χαρακτηριστικά Συστήματος Λογισμικού

2.1.4.1 Αξιοπιστία

Ένα από τα σημαντικότερα χαρακτηριστικά που πρέπει να διαθέτει ένα σύστημα λογισμικού είναι η αξιοπιστία γι' αυτό ήταν και σημαντικό να τεθούν κάποιοι περιορισμοί που θα το κάνουν όσο το δυνατόν πιο αξιόπιστο. Για την εισαγωγή των δεδομένων της τοποθεσίας, ο χρήστης δεν μπορεί να κάνει οποιοδήποτε λάθος, αφού του δίνονται όλες οι τοποθεσίες της Κύπρου σε μια λίστα και επιλέγει μία από αυτές. Έτσι δεν έχει την δυνατότητα να προσθέσει κάτι ο ίδιος και να κάνει λάθος. Το ίδιο συμβαίνει και με την καταχώρηση ώρας και ημερομηνίας αφού του δίνεται μία φόρμα αναπαράστασης ρολογιού και ένα ημερολόγιο αντίστοιχα.

Σχετικά με την Βάση Δεδομένων κανένας χρήστης δεν μπορεί να έχει πρόσβαση σε αυτήν και να της προσθέσει λάθος δεδομένα αφού η εφαρμογή παίρνει όλα τα δεδομένα της Βάσης μόλις ανοίξει και τα αποθηκεύει σε πίνακες της Java και ανάλογα με την επιλογή του χρήστη του εμφανίζονται τα κατάλληλα δεδομένα. Επιπρόσθετα, η Βάση Δεδομένων SQL Server που χρησιμοποιώ παρέχει γρήγορη κατασκευή και λειτουργία του συστήματος, λόγω του ότι αποθηκεύει πολλαπλά αντίγραφα των Βάσεων Δεδομένων (backup) ώστε να μην υπάρχει κίνδυνος απώλειας δεδομένων σε περίπτωση σφάλματος ή κατάρρευσης του συστήματος. Ακόμη, επιτρέπει στους χρήστες την δυνατότητα να κάνουν σχετικά ερωτήματα για αναζήτηση δεδομένων από τη Βάση, ενημέρωση και συγχρονισμό των δεδομένων.

2.1.4.2 Διαθεσιμότητα

Το σύστημα θα είναι πάντα διαθέσιμο στους χρήστες, για να αναζητήσουν οποιαδήποτε διαδρομή επιθυμούν και να δουν τις πληροφορίες της. Χρησιμοποιώντας SQL Server για την αποθήκευση των δεδομένων του συστήματος, η ανάκτηση δεδομένων είναι γρήγορη και εύκολη για την εφαρμογή, χρησιμοποιώντας τα κατάλληλα queries προς τη βάση σχετικά με τα δεδομένα που χρειάζεται.

2.1.4.3 Ασφάλεια

Το σύστημα θα εξασφαλίζει την ακεραιότητα των δεδομένων που είναι καταχωρημένα στη βάση παρ' όλα τα πιθανά σφάλματα, κάτι το οποίο μας παρέχει η Βάση Δεδομένων που θα χρησιμοποιήσουμε, λόγω του ότι δημιουργεί πολλαπλά αντίγραφα της Βάσης, αποκλείοντας έτσι τον κίνδυνο να χάσουμε τα δεδομένα όταν το σύστημα παρουσιάσει κάποιο σφάλμα.

Είναι μια εφαρμογή κατά την οποία ο χρήστης δεν προσθέτει τα προσωπικά του δεδομένα και το μόνο που κάνει είναι να βρίσκει την κοντινότερη διαδρομή μεταξύ δύο τοποθεσιών και να βλέπει τις πληροφορίες της και έτσι δεν γίνονται οποιοδήποτε έλεγχοι για την ασφάλεια του συστήματος, αφού το σύστημα όπως είναι τώρα είναι ασφαλές για τον χρήστη.

2.1.4.4 Συντήρηση

Η συντήρηση ενός συστήματος λογισμικού είναι ιδιαίτερα σημαντική γιατί βοηθά στη διόρθωση λαθών που εμφανίστηκαν μετά τη λειτουργία του συστήματος ή παραλείψεων από την πλευρά του μηχανικού λογισμικού. Επομένως, σε κάθε φάση της ανάπτυξης του συστήματος λογισμικού είναι απαραίτητο να καταγράφεται τεκμηρίωση (σχόλια στον κώδικα) για κάθε ενέργεια που γίνεται καθώς επίσης και διάφορες δυσκολίες και λάθη που αντιμετωπίστηκαν, έτσι ώστε να μπορεί να γίνει εύκολα η συντήρηση του.

2.1.4.5 Μεταφερσιμότητα

Δυστυχώς το παρών σύστημα λογισμικού που έχει αναπτυχθεί δεν μπορεί να εγκατασταθεί σε οποιοδήποτε κινητό τηλέφωνο επιθυμούμε, αφού υποστηρίζει μόνο λειτουργικό σύστημα android για την εμφάνιση της εφαρμογής.

2.2 Βασικές λειτουργίες Χρήστη

Λειτουργία 1: Καταχώρηση αρχικής και τελικής τοποθεσίας της διαδρομής

Ο χρήστης εισάγει την αρχική του τοποθεσία πατώντας στο κουμπί με το εικονίδιο της αναζήτησης δίπλα από την ετικέτα Start, η οποία θα τον μεταφέρει σε ένα άλλο Activity στο οποίο θα πρέπει να συμπληρώσει την διεύθυνση του. Η προκαθορισμένη τοποθεσία του πεδίου αυτού είναι η τοποθεσία που βρίσκεται τώρα. Με τον ίδιο τρόπο εισάγει και την τελική του τοποθεσία για την οποία όμως δεν υπάρχει προκαθορισμένη τιμή.

Λειτουργία 2: Προβολή τοποθεσίας στο χάρτη

Ο χρήστης βλέπει την τοποθεσία που έχει προσθέσει στο χάρτη πατώντας στο εικονίδιο «Refresh» στο πάνω δεξιό μέρος του Activity.

Λειτουργία 3: Καταχώρηση ώρας της διαδρομής

Ο χρήστης εισάγει την ώρα της διαδρομής πατώντας στο κουμπί με το εικονίδιο που βρίσκεται στο δεξιό μέρος της ετικέτας «Time».

Λειτουργία 4: Καταχώρηση ημερομηνίας της διαδρομής

Ο χρήστης εισάγει την ημερομηνία της διαδρομής πατώντας στο κουμπί με το εικονίδιο που βρίσκεται στο δεξιό μέρος της ετικέτας «Date».

Λειτουργία 5: Επιλογή της κοντινότερης διαδρομής

Ο χρήστης επιλέγει την διαδρομή που του εμφανίζεται για να δει τις πληροφορίες της.

Λειτουργία 6: Διάβασμα Χρονοδιαγράμματος

Ο χρήστης διαβάζει το χρονοδιάγραμμα της συγκεκριμένης διαδρομής που έχει επιλέξει όταν πατήσει στην καρτέλα «Time Schedule». Πιο συγκεκριμένα, βλέπει την πιο κοντινή στάση και πόσα λεπτά χρειάζεται να περπατήσει για να φτάσει σε αυτήν. Επίσης, έχει την

δυνατότητα προβολής όλων των στάσεων από τις οποίες περνά η διαδρομή αυτή. Τέλος, μπορεί να δει τον συνολικό χρόνο του ταξιδιού, όπως και τότε το λεωφορείο περνά από την στάση.

Λειτουργία 7: Τύπωμα διαδρομής

Ο χρήστης τυπώνει την συγκεκριμένη διαδρομή που έχει επιλέξει πατώντας στο εικονίδιο της δισκέτας.

Λειτουργία 8: Διάβασμα διαδρομής στον χάρτη

Ο χρήστης διαβάζει την συγκεκριμένη διαδρομή που έχει επιλέξει στον χάρτη, όταν πατήσει στην καρτέλα «Map View».

2.3 Use Cases

2.3.1 Actors

Ένας actor είναι ένα άτομο, σύστημα, συσκευή hardware ή οτιδήποτε άλλο θα αλληλεπιδρά με το σύστημα. Ενεργός actor είναι αυτός που ξεκινά τις αλληλεπιδράσεις με το σύστημα και παθητικός actor είναι αυτός που είναι παραλήπτης αιτημάτων ή ενεργοποιείται από το σύστημα.

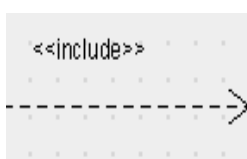
2.3.1.1 Actor Descriptions



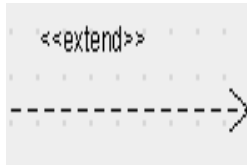
Ένας actor είναι ένα πρόσωπο, σύστημα, κομμάτι του υλικού, που αλληλεπιδρά με το σύστημα. Παρουσιάζεται με το σχήμα στα αριστερά. Διαχωρίζουμε τους ενεργούς actors, που αρχίζουν τις αλληλεπιδράσεις με ένα σύστημα και τους ενεργητικούς actors, οι οποίοι είναι στόχοι των αιτημάτων ή ενεργοποιούνται από το σύστημα.



Ένα use case είναι κομμάτι λειτουργικότητας που θα παρέχεται από το σύστημα.



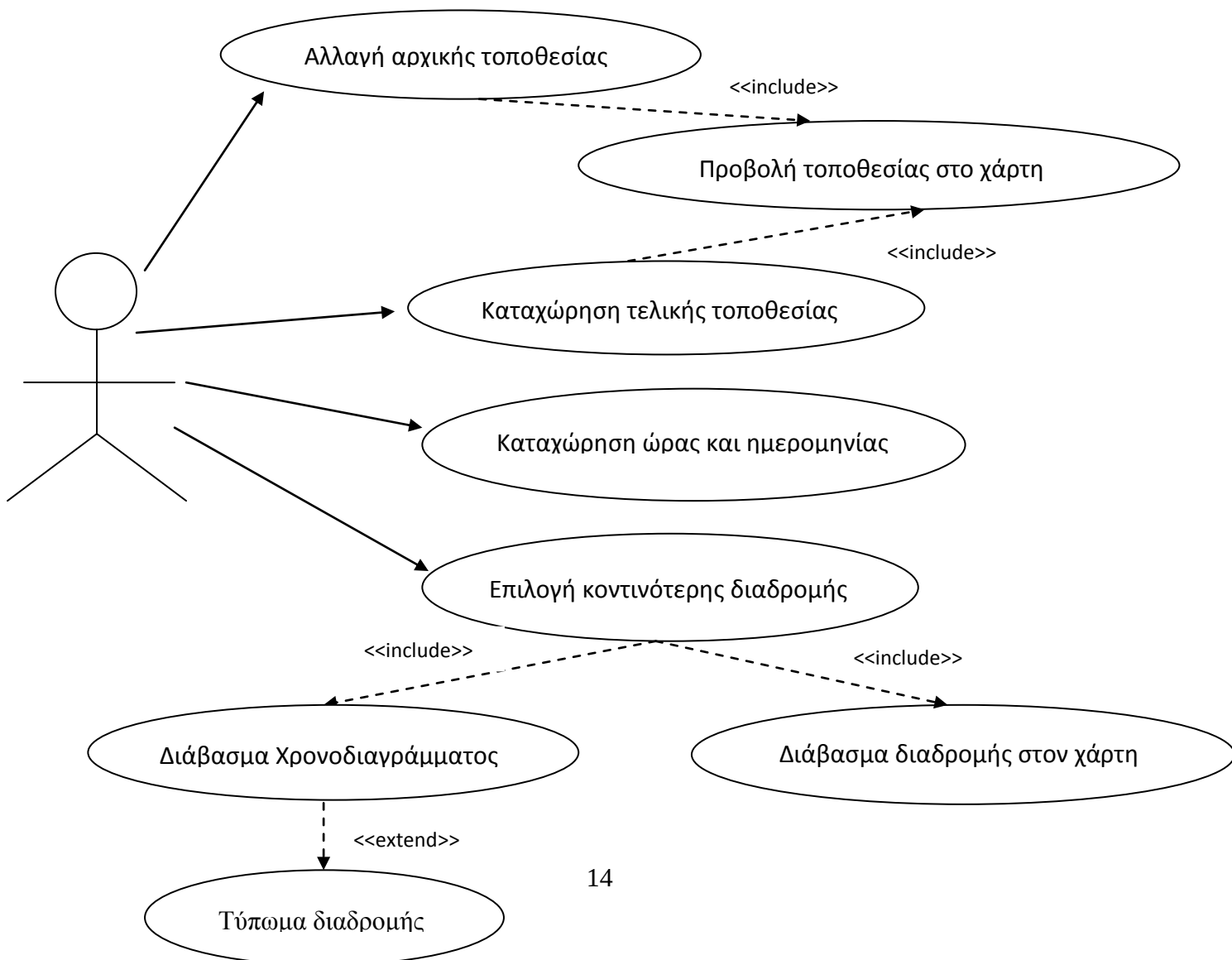
Υποδηλώνει ότι ένα use case πρέπει να συμπεριλαμβάνει ένα άλλο. Για να εκτελεστεί το συγκεκριμένο use case σημαίνει ότι θα εκτελεστεί και το άλλο.



Χρησιμοποιείται όταν ένα use case επιλεκτικά επεκτείνει τη λειτουργικότητα που προσφέρεται από ένα άλλο με ένα extension point. Η εκτέλεση ενός use case σημαίνει ότι μπορεί ή όχι να εκτελεστεί το άλλο.

Περιγραφή επιβάτη λεωφορείου	Ο επιβάτης λεωφορείου μπορεί να εισέρχεται στο σύστημα όποτε θέλει χωρίς να χρειάζεται να συμπληρώσει οποιαδήποτε συνθηματικά και να βρίσκει διαδρομές λεωφορείων που τον ενδιαφέρουν.
Ψευδώνυμο	Δεν υπάρχει
Inherits	Δεν υπάρχει
Τύπος Χρήστη	Ο επιβάτης είναι ενεργό μέλος στο σύστημα μας.

2.3.2 Use Case Diagram

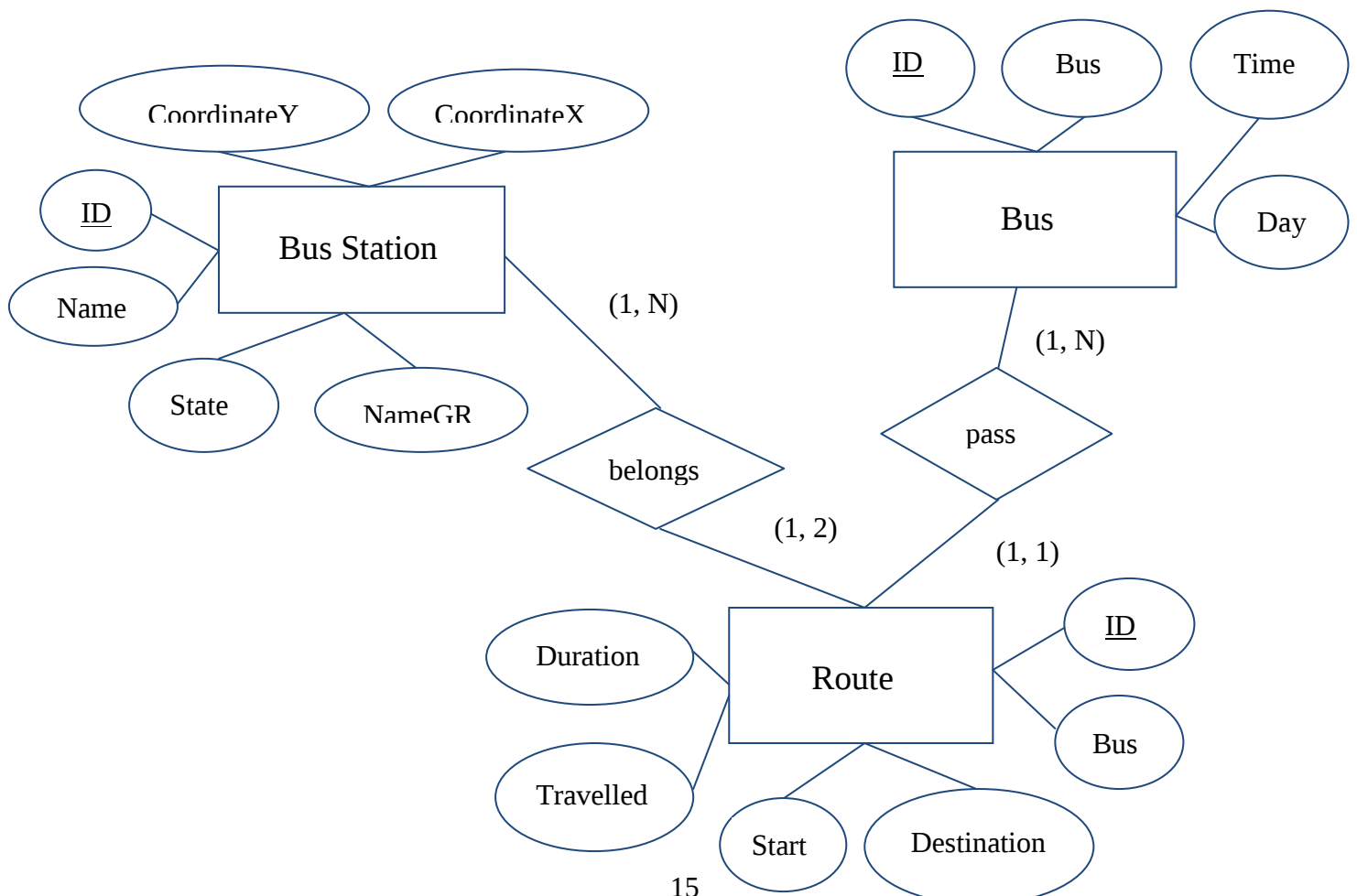


2.4 Ανάλυση και Σχεδίαση Βάσης Δεδομένων

Στην Βάση Δεδομένων του συστήματος αποθηκεύονται όλες οι στάσεις με τα δεδομένα τους και τα ωράρια και τα δρομολόγια των λεωφορείων. Επειδή τα δρομολόγια των λεωφορείων αλλάζουν κάθε περίοδο, η βάση δεδομένων προτιμήθηκε να βρίσκεται στον server του πανεπιστημίου, ούτως ώστε ο χρήστης να μην χρειάζεται να ανανεώνει την εφαρμογή του συνεχώς, για κάποια διαδρομή ή λεωφορείο που μπορεί να προστεθεί στο μέλλον. Για το λόγο αυτό, υλοποιήθηκε μια υπηρεσία διαδικτύου για την επικοινωνία μεταξύ της εφαρμογής και της βάσης.

Για την εφαρμογή δημιουργήθηκαν 3 πίνακες. Ο πίνακας BusStation αποθηκεύει όλες τις στάσεις των λεωφορείων. Ο πίνακας Bus αποθηκεύει όλα τα λεωφορεία. Κάθε λεωφορείο έχει διαφορετικές ώρες για τις εργάσιμες μέρες σε σχέση με το Σάββατο και την Κυριακή. Ο πίνακας Routes αποθηκεύει ποιες στάσεις ενώνονται μεταξύ τους και με ποιο λεωφορείο και πόση είναι η διάρκεια από την μια στην άλλη.

Πιο κάτω βλέπουμε τις σχέσεις που έχουν οι τρεις πίνακες μεταξύ τους μέσω του διαγράμματος συσχετίσεων-οντοτήτων. Σε κάθε σχέση δίνεται και η πληθυντικότητα των συσχετίσεων:



Οι πίνακες που θα αποτελούν τη ΒΔ περιγράφονται πιο κάτω. Αναλυτικά βλέπουμε το κάθε πεδίο και τον τύπο δεδομένων του, το μέγιστο μέγεθος που μπορεί να έχει και την περιγραφή του.

Πίνακας BusStation:

ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ	ΤΥΠΟΣ	ΜΕΓΕΘΟΣ	ΠΕΡΙΓΡΑΦΗ
ID	Integer	4 Bytes	Πρωτεύον κλειδί
Name	Nvarchar(100)	102 Bytes	Όνομα Στάσης
CoordinateX	Float	4 Bytes	Latitude
CoordinateY	Float	4 Bytes	Longitude
State	Nvarchar(10)	12 Bytes	Σε ποια πόλη είναι η στάση
NameGR	Nvarchar(100)	102 Bytes	Ελληνικό όνομα στάσης

Πίνακας Bus:

ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ	ΤΥΠΟΣ	ΜΕΓΕΘΟΣ	ΠΕΡΙΓΡΑΦΗ
ID	Integer	4 Bytes	Πρωτεύον κλειδί
Bus	Integer	4 Bytes	Αριθμός λεωφορείου
Date	Nvarchar(10)	12 Bytes	Ημέρα της Εβδομάδας
Time	Nvarchar(300)	302 Bytes	Ώρες που ξεκινά το λεωφορείο

Πίνακας Route:

ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ	ΤΥΠΟΣ	ΜΕΓΕΘΟΣ	ΠΕΡΙΓΡΑΦΗ
ID	Integer	4 Bytes	Πρωτεύον κλειδί
Bus	Integer	4 Bytes	Αριθμός λεωφορείου
Start	Nvarchar(100)	102 Bytes	Αρχική Στάση
Destination	Nvarchar(100)	102 Bytes	Τελική Στάση
Duration	Float	4 Bytes	Διάρκεια από την αρχική στάση στην τελική στάση.
Travelled	Float	4 Bytes	Χρόνος που διάνυσε το λεωφορείο για να φτάσει στην αρχική στάση.

2.5 Αλγόριθμος Υλοποίησης

Για τον εντοπισμό της κοντινότερης διαδρομής και των λεωφορείων που θα μας πάρουν πιο γρήγορα στον προορισμό μας χρησιμοποιήθηκε ο αλγόριθμος A^* . Ο αλγόριθμος A^* πρόκειται για μία επέκταση του αλγόριθμου Dijkstra, επιτυχαίνει όμως καλύτερη απόδοση στο χρόνο λόγω της χρήσης ευρετικών. Τα ευρετικά υλοποιούνται με μία ευρετική συνάρτηση (Heuristic Function), η οποία επιστρέφει μια τιμή που εκφράζει πόσο κοντά βρίσκεται η τρέχουσα κατάσταση από την τελική κατάσταση που δίνεται αρχικά σαν παράμετρος στο σύστημα.

Ο A^* χρησιμοποιεί best-first search και βρίσκει ένα μονοπάτι ελάχιστου κόστους από ένα δεδομένο αρχικό κόμβο σε ένα κόμβο στόχο. Διασχίζει το γράφημα και ακολουθεί μια πορεία από το χαμηλότερο αναμενόμενο συνολικό κόστος ή απόσταση, κρατώντας μια ταξινομημένη ουρά προτεραιότητας των αναπληρωματικών τμημάτων διαδρομής κατά μήκος του τρόπου που ακολουθεί μέσω του ευρετικού.

Η τιμή που επιστρέφει μέσω του ευρετικού δεν είναι μια αληθινή τιμή απόστασης αλλά μια εκτίμηση. Η ευρετική συνάρτηση του A^* είναι $F(S)=g(S)+h(S)$ όπου:

$g(S)$: είναι η απόσταση από την αρχική κατάσταση, η οποία είναι και η πραγματική τιμή.

$h(S)$: είναι η εκτίμηση της απόστασης S από την τελική κατάσταση μέσω μιας ευρετικής συνάρτησης.

Για την εφαρμογή η ευρετική συνάρτηση υπολογίζεται ως εξής:

- Το $g(S)$ υπολογίζεται από τον χρόνο που χρειάστηκε να διανύσει το λεωφορείο από την αρχική στάση στην στάση που βρίσκεται.
- Το $h(S)$ είναι η εκτίμηση της απόστασης από την συγκεκριμένη στάση μέχρι την τελική στάση. Πιο συγκεκριμένα υπολογίζεται βάσει του χρόνου που χρειάζεται να περιμένει ο χρήστης το λεωφορείο να φτάσει στην στάση που βρίσκεται αυτός, του χρόνου που χρειάζεται το λεωφορείο να φτάσει στην επόμενη στάση και πόσα λεπτά μακριά βρίσκεται η τελική στάση.

Πιο κάτω παραθέτουμε ο ψευδοκώδικας του αλγορίθμου:

```
function A*(start, goal)
  closedset := the empty set      // The set of nodes already evaluated.
  openset := {start}      // The set of tentative nodes to be evaluated, initially
                           containing the start node
  came_from := the empty map      // The map of navigated nodes.

  g_score[start] := 0      // Cost from start along best known path.
  // Estimated total cost from start to goal through y.
  f_score[start] := g_score[start] + heuristic_cost_estimate (start, goal)

  while openset is not empty
    current := the node in openset having the lowest f_score[] value
    if current = goal
      return reconstruct_path(came_from, goal)

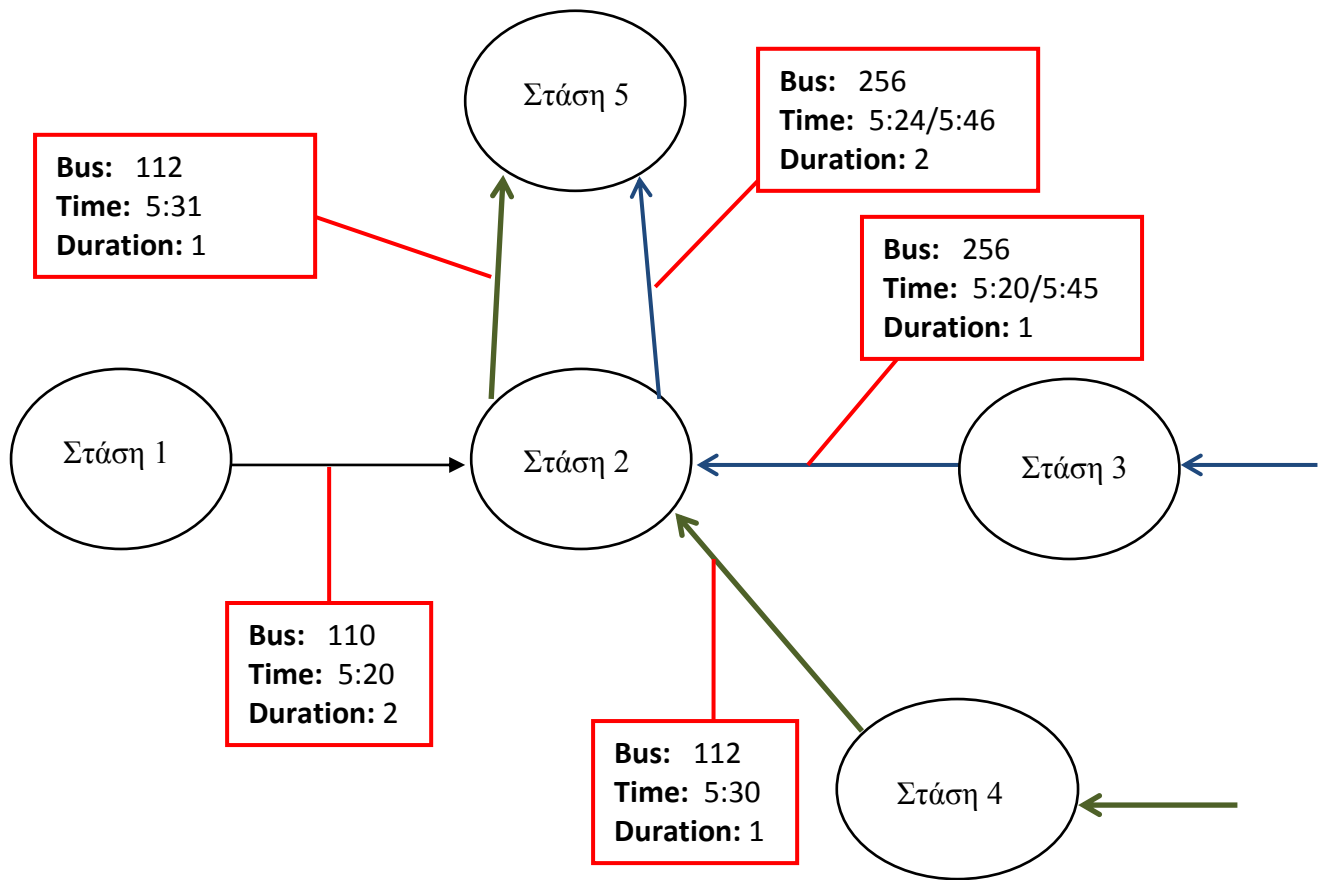
    remove current from openset
    add current to closedset
    for each neighbor in neighbor_nodes(current)
      if neighbor in closedset
        continue
      tentative_g_score := g_score[current] + dist_between(current, neighbor)

      if neighbor not in openset or tentative_g_score < g_score[neighbor]
        came_from[neighbor] := current
        g_score[neighbor] := tentative_g_score
        f_score[neighbor] := g_score[neighbor] +
        heuristic_cost_estimate(neighbor, goal)
        if neighbor not in openset
          add neighbor to openset

    return failure

function reconstruct_path (came_from, current_node)
  if current_node in came_from
    p := reconstruct_path(came_from, came_from[current_node])
    return (p + current_node)
  else
    return current_node
```

Παράδειγμα επιλογής μονοπατιού:



Θέλουμε να πάμε από την ΣΤΑΣΗ 1 στην ΣΤΑΣΗ 5 και έστω τρέχουμε τον αλγόριθμο η ώρα 5:10.

ΣΤΑΣΗ 1: Περιμένουμε 10 λεπτά στην στάση μέχρι να έρθει το πρώτο λεωφορείο.

- $G(1) = 0;$
- $H(1) = 0;$

ΣΤΑΣΗ 1 – ΣΤΑΣΗ 2: Παίρνουμε το λεωφορείο 110 το οποίο έρχεται στις 5:20.

- $G(2) = 2;$
- $H(2) = 10 + 2 + \text{distToDest}();$

ΣΤΑΣΗ 2: Από την στάση 2 περνάνε 2 λεωφορεία το 112 και το 256.

Λεωφορείο 112

Αν πάρουμε το λεωφορείο 112 τότε θα περιμένουμε 9 λεπτά.

Λεωφορείο 256

Αν πάρουμε αρχικά το λεωφορείο 110, χάνουμε το λεωφορείο 256 στις 5:21 γιατί η τρέχουσα ώρα είναι 5:22 και παίρνουμε το επόμενο στις 5:46 άρα θα περιμένουμε 24 λεπτά.

ΣΤΑΣΗ 2 – ΣΤΑΣΗ 5: θα πάρουμε το λεωφορείο 112 γιατί θα περιμένουμε πιο λίγα λεπτά για αυτό.

- $G(3) = 12;$
- $H(3) = 9 + 1 + \text{distToDest}();$

ΣΤΑΣΗ 5: Η στάση 5 είναι η τελική στάση και τερματίζει ο αλγόριθμος.

Το $\text{distToDest}()$ υπολογίζεται με τον haversine τύπο[12] που υπολογίζει την μικρότερη απόσταση μεταξύ δύο σημείων πάνω στην επιφάνεια της Γης αγνοώντας φυσικά οποιαδήποτε βουνά.

Κεφάλαιο 3:

Υλοποίηση Εφαρμογής

3.1 Ανατομία μιας εφαρμογής Android	21
3.2 Γλώσσα Προγραμματισμού Java	28
3.3 SDK Manager	30
3.4 Γλώσσα Προγραμματισμού C#	31
3.5 Microsoft SQL Server 2008	32

Σε αυτό το κεφάλαιο παρουσιάζονται τόσο οι γλώσσες προγραμματισμού όσο και τα εργαλεία ανάπτυξης που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής.

3.1 Ανατομία μιας εφαρμογής Android

Υπάρχουν τέσσερα δομικά blocks από τα οποία αποτελείται μια Android εφαρμογή: Activity, Intent Receiver, Service, Content Provider. Δεν χρειάζεται κάθε εφαρμογή να έχει και τα τέσσερα αυτά συστατικά, αλλά σίγουρα χρειάζεται κάποιον συνδυασμό από αυτά.

Android Manifest

Όταν ο προγραμματιστής αποφασίσει ποια στοιχεία του χρειάζονται για να αναπτύξει την εφαρμογή, αυτά πρέπει να καθοριστούν και να καταγραφούν σε ένα αρχείο που ονομάζεται AndroidManifest.xml. Στο αρχείο αυτό, δηλώνονται τα δομικά blocks που θα χρησιμοποιηθούν και ποιες δυνατότητες και προδιαγραφές θα εξυπηρετούν.

Σε κάθε εφαρμογή πρέπει να υπάρχει το αρχείο AndroidManifest.xml, το οποίο δημιουργείται αυτόματα μετά από την δημιουργία ενός καινούργιου project μιας android εφαρμογής. Το αρχείο αυτό περιέχει βασικές πληροφορίες σχετικά με την εφαρμογή, τις οποίες το λειτουργικό σύστημα πρέπει να γνωρίζει προτού τρέξει οποιοδήποτε άλλο κώδικα.

Οι σημαντικότερες από αυτές τις πληροφορίες περιγράφονται παρακάτω:

- ❖ Η ονομασία του πακέτου της Java της εφαρμογής (Java package).
- ❖ Η έκδοση της εφαρμογής (π.χ. 1.0, 1.4.2, 2.7 κλπ).
- ❖ Η ελάχιστη έκδοση του λειτουργικού συστήματος Android που απαιτεί η εφαρμογή (min Sdk version). Για παράδειγμα αν έχει δηλωθεί min sdk version ο αριθμός 7, που ισοδυναμεί με την έκδοση Android 2.1, τότε η εφαρμογή θα να μπορεί εκτελεστεί σε συσκευές με έκδοση Android μεγαλύτερη ή ίση της 2.1.
- ❖ Το όνομα της εφαρμογής, καθώς και το εικονίδιό της.
- ❖ Οι άδειες που απαιτούνται για να εκτελεστούν ορισμένες λειτουργίες της εφαρμογής, όπως άδεια για σύνδεση στο διαδίκτυο κτλ.

Ιδιαίτερα σημαντικό στοιχείο το οποίο δηλώνεται στο αρχείο Manifest είναι οι άδειες πρόσβασης της εφαρμογής. Στο android οι διάφοροι πόροι της συσκευής (δίκτυο, εξωτερική κάρτα μνήμης, GPS) προστατεύονται και για να μπορεί κάποια εφαρμογή να τους χρησιμοποιήσει πρέπει να έχει δηλωθεί η αντίστοιχη άδεια στο αρχείο αυτό.

Για παράδειγμα, αν η εφαρμογή μας χρησιμοποιεί την κάμερα της συσκευής, θα πρέπει να δηλώνεται και αντίστοιχη άδεια. Αν θέλουμε να έχουμε πρόσβαση στο διαδίκτυο από την εφαρμογή μας, θα πρέπει να δηλώνεται η αντίστοιχη άδεια. Όμοια ισχύουν για το αν θέλουμε να αποθηκεύσουμε αρχεία στην κάρτα SD, αν θέλουμε να στείλουμε μηνύματα SMS, αν θέλουμε να πραγματοποιήσουμε τηλεφωνικές κλήσεις κλπ. Αν δε δηλώσουμε τις άδειες που απαιτούνται για τις διάφορες λειτουργίες της εφαρμογής, τότε θα εμφανίζεται σφάλμα και η εφαρμογή δε θα λειτουργεί. Οι άδειες αυτές περιγράφονται στο χρήστη τη στιγμή που εγκαθιστά την εφαρμογή, και πρέπει να συμφωνήσει με αυτές για να ολοκληρωθεί η εγκατάσταση. Με τον τρόπο αυτό, ο χρήστης αισθάνεται ασφαλής απέναντι σε κακόβουλες εφαρμογές. Για παράδειγμα αν ο χρήστης επιχειρήσει να εγκαταστήσει μία εφαρμογή άσχετη με τηλεφωνικές κλήσεις ή μηνύματα, και δει πριν την εγκατάσταση ότι αυτή ζητάει άδεια για τηλεφωνικές κλήσεις ή αποστολή μηνυμάτων, τότε θα καταλάβει ότι η εφαρμογή αυτή είναι πιθανότατα κακόβουλη και δεν πρέπει να εγκατασταθεί. Αν, βέβαια, οι συγκεκριμένες αυτές άδειες δεν αναγράφονται, τότε ο χρήστης είναι σίγουρος ότι η εφαρμογή δε θα μπορέσει με κανένα τρόπο να στείλει κάποιο γραπτό μήνυμα ή να πραγματοποιήσει κάποια τηλεφωνική κλήση. Ακόμα και αν επιχειρούσε να το κάνει, η εφαρμογή θα εμφάνιζε σφάλμα τη στιγμή που επιχειρούσαμε να την τρέξουμε και δε θα λειτουργούσε.

Στο παρακάτω παράδειγμα, ζητείται άδεια χρήσης του διαδικτύου, της εξωτερικής κάρτας μνήμης και της κάμερας.

```
<uses-permission android: name="android.permission.INTERNET"/>
<uses-permission android: name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<Uses-permission android: name="android.permission.CAMERA"/>
```

Activity

Η Activity είναι το πιο σύνηθες από τα τέσσερα άλλα δομικά στοιχεία. Ένα activity σχετίζεται με μια διαφορετική οθόνη της εφαρμογής. Κάθε activity είναι μια ξεχωριστή κλάση που κληρονομεί από την βασική κλάση Activity του Android. Η κλάση αυτή εμφανίζει στην οθόνη μια διεπαφή χρήστη που συντίθεται από Views και απαντά σε διάφορα γεγονότα (events). Για παράδειγμα, μια εφαρμογή γραπτών μηνυμάτων μπορεί να έχει μια οθόνη που να δείχνει μια λίστα με τις επαφές στις οποίες μπορεί να σταλεί το συγκεκριμένο μήνυμα, μια δεύτερη οθόνη για εγγραφή του μηνύματος στη δεδομένη επαφή και άλλες οθόνες που να κάνουν ρυθμίσεις των μηνυμάτων ή να δείχνουν το ιστορικό αποστολής μηνυμάτων. Κάθε μία από τις οθόνες αυτές θα υλοποιούνται ως μια activity. Η εναλλαγή μεταξύ των διαφορετικών "οθονών" ισοδυναμεί με έναρξη μιας καινούριας activity. Σε μερικές περιπτώσεις μια Activity μπορεί να επιστρέψει μια τιμή σε μια προηγούμενη activity. Όταν μια νέα οθόνη ανοίγει, η προηγούμενη μπαίνει σε αναμονή λειτουργίας και τοποθετείται σε στοίβα με τις πρόσφατα μη ολοκληρωμένες activities. Ο χρήστης μπορεί να πλοηγηθεί προς τα πίσω και να δει τα προηγούμενα ανοιχτά screens. Τα screens αυτά μπορεί επίσης να επιλεγεί να διαγραφούν από τη στοίβα ιστορικού όταν καταλαμβάνουν χώρο και δεν χρειάζεται να μένουν εκεί.

Το λογισμικό Android χρησιμοποιεί μια ειδική κλάση που ονομάζεται Intent για το "πέρασμα" από activity σε activity. Τα δύο πιο σημαντικά στοιχεία της αρχιτεκτονικής λειτουργίας του intent είναι η ενέργεια και τα δεδομένα επί των οποίων θα ενεργήσει. Τυπικές τιμές για μια δράση/ενέργεια είναι οι MAIN, VIEW, PICK, EDIT κ.α. Τα δεδομένα εκφράζονται μέσω ενός URI (Uniform Resource Indicator). Για παράδειγμα, για να δούμε ένα site στον browser , θα πρέπει να δημιουργήσουμε ένα Intent με τη δράση ACTION και με τα δεδομένα να καθορίζονται από το website-uri.

Υπάρχει επίσης μια σχετική κλάση που ονομάζεται IntentFilter η οποία περιγράφει ποιες και πόσες αιτήσεις για να γίνει κάτι (intents) μπορεί να χειριστεί μια activity.

Intent Receiver

Ένας intent receiver χρειάζεται όταν ο προγραμματιστής της εφαρμογής θέλει να χρησιμοποιήσει κώδικα μέσα στην εφαρμογή του, που θα εκτελείται όταν συμβαίνει ένα εξωτερικό γεγονός, για παράδειγμα όταν χτυπά το τηλέφωνο ή όταν ένα ασύρματο δίκτυο γίνεται διαθέσιμο. Οι intent receivers δεν προβάλλουν κάποια διεπαφή χρήστη, αλλά προβάλλουν Notifications για να ειδοποιήσουν τον χρήστη, εάν κάτι σημαντικό λαμβάνει χώρα. Οι Intent Receivers δηλώνονται και αυτοί στο AndroidManifest.xml.

Services

Ένα service είναι τμήμα κώδικα που εκτελείται χωρίς κάποιο interface χρήστη. Ένα καλό παράδειγμα service είναι ο media player που παίζει τραγούδια από μια λίστα. Σε μια εφαρμογή media player, είναι λογικό να υπάρχουν διάφορες οθόνες, άρα και πολλές activities, όπου ο χρήστης θα μπορεί να επιλέξει τραγούδια και να τα ακούσει. Παρόλα αυτά το playback δε θα πρέπει να χειρίζεται από μια activity, γιατί ο χρήστης περιμένει να μπορεί να περιηγείται στη λίστα τραγουδιών του, χωρίς το τραγούδι που ακούει εκείνη τη στιγμή να σταματήσει. Σε αυτή την περίπτωση, η κύρια activity του media player θα ξεκινήσει να εκτελεί ένα service στο background, οπότε ο χρήστης θα μπορεί να κάνει pause, rewind κλπ.

Content Providers

Οι εφαρμογές μπορούν να αποθηκεύσουν τα δεδομένα τους σε αρχεία, στη βάση δεδομένων SQLite, σε preferences ή χρησιμοποιώντας οποιονδήποτε άλλο μηχανισμό που τους παρέχει αυτή τη δυνατότητα. Ένας content provider επίσης είναι χρήσιμος εάν θέλουμε τα δεδομένα μιας εφαρμογής να γίνουν διαθέσιμα και σε άλλες εφαρμογές. Ένας content provider είναι μια κλάση που υλοποιεί ένα standard set από μεθόδους, οι οποίες επιτρέπουν σε άλλες εφαρμογές να αποθηκεύουν και να ανακτούν τον τύπο δεδομένων που χειρίζεται ο content provider.

Resources

Στα resources μιας εφαρμογής ορίζεται το layout των activities, οι διάφορες εικόνες και λεκτικά που χρησιμοποιούνται στα activities. Σε κάθε activity αντιστοιχεί ένα Layout αρχείο, το οποίο περιγράφει τη θέση των διάφορων αντικειμένων στην οθόνη.

Το layout αρχείο είναι ένα αρχείο XML. Στην πράξη το αρχείο αυτό διαμορφώνεται από κατάλληλους γραφικούς editors που προσφέρονται από ολοκληρωμένα περιβάλλοντα ανάπτυξης όπως το Eclipse και από την γλώσσα σήμανσης XML. Σημαντικό να αναφερθεί

ότι στην ανάπτυξη εφαρμογής Android η XML χρησιμοποιείται για την δημιουργία γραφικής διεπαφής, γενικότερα όμως χρησιμοποιείται για αναπαράσταση αυθαίρετων δομών δεδομένων που προκύπτουν για παράδειγμα στις υπηρεσίες ιστού.

Η εντολή που χρησιμοποιείται για να δηλωθεί το αρχείο xml που θα χρησιμοποιηθεί σε μία activity είναι η setContentView(). Όταν θα εκτελεστεί η εφαρμογή μας και συγκεκριμένα η εντολή setContentView() το λειτουργικό σύστημα Android θα ελέγξει τον προσανατολισμό της συσκευής μας και θα επιλέξει το κατάλληλο αρχείο xml.

Στο παράδειγμα που ακολουθεί, ορίζεται ένα Textview με αναγνωριστικό textView1, το οποίο είναι ένα block κειμένου. Δεξιά από αυτό (android: layout_toRightOf="@+id/textView1"), ορίζεται ένα πεδίο στο οποίο μπορεί ο χρήστης να εισάγει κείμενο, με αναγνωριστικό editText1.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/editText1"
        android:layout_alignBottom="@+id/editText1"
        android:layout_alignParentLeft="true"
        android:layout_marginLeft="22dp"
        android:text="Status:" />

    <EditText
        android:id="@+id/editText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/captureFront"
        android:layout_marginBottom="20dp"
        android:layout_marginLeft="14dp"
```

```
        android:layout_toRightOf="@+id/textView1"

        android:ems="10"

        android:inputType="textPersonName" >

        <requestFocus />

    </EditText>

</RelativeLayout>
```

Handlers

Ένας Handler επιτρέπει την αποστολή και επεξεργασία μηνυμάτων και εκτελέσιμων αντικειμένων που σχετίζονται με το Message Queue ενός νήματος. Κάθε αντικείμενο τύπου Handler συνδέεται με ένα νήμα και την ουρά μηνυμάτων αυτού του νήματος.

Υπάρχουν δύο κύριες χρήσεις για ένα Handler:

- ❖ ο χρονοπρογραμματισμός μηνυμάτων και runnables ώστε να εκτελεστούν σε κάποιο σημείο στο μέλλον.
- ❖ η τοποθέτηση στην ουρά μιας ενέργειας που πρέπει να εκτελεστεί σε ένα διαφορετικό νήμα από το τρέχον.

Η δεύτερη περίπτωση έχει ιδιαίτερο ενδιαφέρον, καθώς για λόγους καλής λειτουργίας, χρονοβόρες διαδικασίες που πρέπει να τρέξουν σε κάποιο activity, τοποθετούνται σε διαφορετικό thread. Αν για παράδειγμα με το πάτημα ενός πλήκτρου πρέπει να σταλεί κάτι στο διαδίκτυο και η απάντηση να ενημερώσει κάποιο block στο activity, τότε η διαδικασία γίνεται μέσω ενός νέου νήματος. Προκειμένου όμως το νήμα να μπορεί να ενημερώσει το activity που το δημιούργησε θα πρέπει να χρησιμοποιήσει ένα handler.

Asynchronous Task

Το AsyncTask επιτρέπει την σωστή και εύκολη χρήση του κύριου νήματος που εκτελεί τις λειτουργίες μιας Activity που είναι το UI thread. Η κλάση αυτή επιτρέπει να εκτελεστούν οι λειτουργίες στο παρασκήνιο και να δοθούν τα αποτελέσματα στο νήμα UI χωρίς να χρειάζεται να γίνει διαχείριση οποιωνδήποτε άλλων threads ή handlers.

Το AsyncTask έχει σχεδιαστεί για να είναι μια βοηθητική κλάση γύρω από Threads και Handlers και δεν αποτελεί ένα γενικό πλαίσιο threading. Επίσης, θα πρέπει ιδανικά να χρησιμοποιείται για μικρές λειτουργίες (μερικά δευτερόλεπτα το πολύ). Εάν πρέπει να κρατηθούν νήματα στο παρασκήνιο να τρέχουν για μεγάλο χρονικό διάστημα, συνιστάται να

χρησιμοποιηθούν τα διάφορα APIs που παρέχονται από το πακέτο `java.util.concurrent` όπως είναι ο `Executor`, ο `ThreadPoolExecutor` και το `FutureTask`.

Τέλος, μια ασύγχρονη εργασία ορίζεται από 3 γενικούς τύπους, που ονομάζονται `Params`, `Progress` και `Results`, και 4 συναρτήσεις που τρέχουν με κάποια προκαθορισμένη σειρά για να εκτελεστεί σωστά η ασύγχρονη εργασία. Οι συναρτήσεις αυτές είναι η `onPreExecute`, η `doInBackground`, η `onProgressUpdate` και η `onPostExecute`.

Ένα μικρό παράδειγμα ασύγχρονης εργασίας που κατεβάζει αρχεία από έναν σύνδεσμο όταν ο χρήστης επεξεργάζεται την εφαρμογή:

```
private class DownloadFilesTask extends AsyncTask<URL, Integer, Long> {  
    protected Long doInBackground(URL... urls) {  
        int count = urls.length;  
        long totalSize = 0;  
        for (int i = 0; i < count; i++) {  
            totalSize += Downloader.downloadFile(urls[i]);  
            publishProgress((int) ((i / (float) count) * 100));  
            // Escape early if cancel() is called  
            if (isCancelled()) break;  
        }  
        return totalSize;  
    }  
  
    protected void onProgressUpdate(Integer... progress) {  
        setProgressPercent(progress[0]);  
    }  
  
    protected void onPostExecute(Long result) {  
        showDialog("Downloaded " + result + " bytes");  
    }  
}
```

Στην διπλωματική μου εργασία χρησιμοποίησα τέσσερις φορές ασύγχρονες εργασίες για να κάνω κάποιες λειτουργίες στο παρασκήνιο και να μην μπλοκάρει η διεπαφή του χρήστη. Οι πρώτες τρεις ήταν για να πάρω τα δεδομένα των τριών πινάκων της Βάσης Δεδομένων μέσω της υπηρεσίας διαδικτύου (web service) που είχα δημιουργήσει. Η τελευταία ήταν για να τρέξει ο αλγόριθμος και να υπολογίσει την κοντινότερη διαδρομή με βάση τα δεδομένα που είχε δώσει ο χρήστης στην προηγούμενη οθόνη.

3.2 Γλώσσα Προγραμματισμού Java

Η Java είναι μια αντικειμενοστρεφής γλώσσα προγραμματισμού που σχεδιάστηκε από την εταιρεία πληροφορικής *Sun Microsystems*. Στις αρχές του 1991, η *Sun* αναζητούσε το κατάλληλο εργαλείο για να αποτελέσει την πλατφόρμα ανάπτυξης λογισμικού σε μικροσυσκευές (έξυπνες οικιακές συσκευές έως πολύπλοκα συστήματα παραγωγής γραφικών). Τα εργαλεία της εποχής ήταν γλώσσες όπως η C++ και η C. Μετά από διάφορους πειραματισμούς προέκυψε το συμπέρασμα ότι οι υπάρχουσες γλώσσες δεν μπορούσαν να καλύψουν τις ανάγκες τους. Ο "πατέρας" της Java, James Gosling, που εργαζόταν εκείνη την εποχή για την *Sun*, έκανε ήδη πειραματισμούς πάνω στη C++ και είχε παρουσιάσει κατά καιρούς κάποιες πειραματικές γλώσσες (C++ ++) ως πρότυπα για το νέο εργαλείο που αναζητούσαν στην *Sun*. Τελικά μετά από λίγο καιρό κατέληξαν με μια πρόταση για το επιτελείο της εταιρίας, η οποία ήταν η γλώσσα *Oak*. Η γλώσσα αυτή είχε πολύ πιο έντονο αντικειμενοστρεφή (*object oriented*) χαρακτήρα σε σχέση με την C++ και χαρακτηριζόταν για την απλότητα της. Σύντομα όμως ανακάλυψαν ότι το όνομα *Oak* ήταν ήδη κατοχυρωμένο οπότε αποφάσισαν να μετονομάσουν το νέο τους δημιούργημα σε Java που ήταν το όνομα της αγαπημένης ποικιλίας καφέ για τους δημιουργούς της. Η Java ακολουθεί μία ανοδική πορεία και είναι πλέον μία από τις πιο δημοφιλείς γλώσσες στο χώρο της πληροφορικής. Στις 13 Νοεμβρίου του 2006 η Java έγινε πλέον μια γλώσσα ανοιχτού κώδικα (GPL) όσον αφορά το μεταγλωττιστή (*javac*) και το πακέτο ανάπτυξης (*JDK*, *Java Development Kit*).

Στις 27 Απριλίου 2010, η εταιρία λογισμικού *Oracle Corporation* εξαγόρασε την *Sun Microsystems* και τις τεχνολογίες της. Η συγκεκριμένη συμφωνία θεωρείται σημαντική για το μέλλον της Java και του γενικότερου οικοσυστήματος τεχνολογιών γύρω από αυτή μιας και ο έμμεσος έλεγχος της τεχνολογίας και η εξέλιξη της περνάει σε άλλα χέρια.

Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και της πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε *Windows*, *Linux*, *Unix*, *Macintosh* χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα.

Για την οικοδόμηση εφαρμογών σε λειτουργικό σύστημα *Android* χρειάζεται τόσο ένα περιβάλλον ανάπτυξης επαγγελματικού επιπέδου όπως είναι το *ADT* (*Android developer Tools*) όσο και η γλώσσα προγραμματισμού Java. Κάθε *Layout* είναι συνδεδεμένο με μια *Activity* κλάση. Αυτή η κλάση περιέχει την μέθοδο *onCreate()*, η οποία καλείται από το πρόγραμμα κάθε φορά που καλείται το συγκεκριμένο *Activity*. Επίσης σε αυτή την κλάση υλοποιούνται και οι λειτουργίες *onClick()* που καλούνται όταν πατηθεί κάποιο κουμπί στην

γραφική διεπαφή του χρήστη. Κάθε Activity μπορεί να καλέσει κάποια άλλη Activity και να τερματίσει με την εντολή finish(). Επίσης με την εντολή putExtra() μπορούμε να στείλουμε δεδομένα στην Activity που καλούμε.

Πιο κάτω παρατίθενται δύο παραδείγματα χρήσης της γλώσσας προγραμματισμού Java για την σύνδεση δύο οθονών με το κάλεσμα της μιας από την άλλη και την μεταφορά των δεδομένων που απαιτούνται.

Παράδειγμα καλέσματος μιας Activity από μία άλλη:

```
Intent intent= new Intent(this, MapActivity.class);
Intent.putExtra("Date", date);
Intent.putExtra("Time", time);
this.startActivity(intent);
finish();
```

Παράδειγμα πως παίρνουμε δεδομένα που στάλθηκαν από κάποια άλλη Activity:

```
Intent intent; intent = getIntent();
if (intent != null) {
    if (intent.getStringExtra("Time")!= null){
        time = intent.getStringExtra("Time");
    } }
}
```

Επίσης, για την υλοποίηση του αλγορίθμου που βρίσκει την κοντινότερη διαδρομή χρησιμοποιήθηκε η γλώσσα προγραμματισμού Java, αφού οι λειτουργίες σε μια Android εφαρμογή γράφονται σε αυτήν την γλώσσα.

3.3 Android SDK Manager

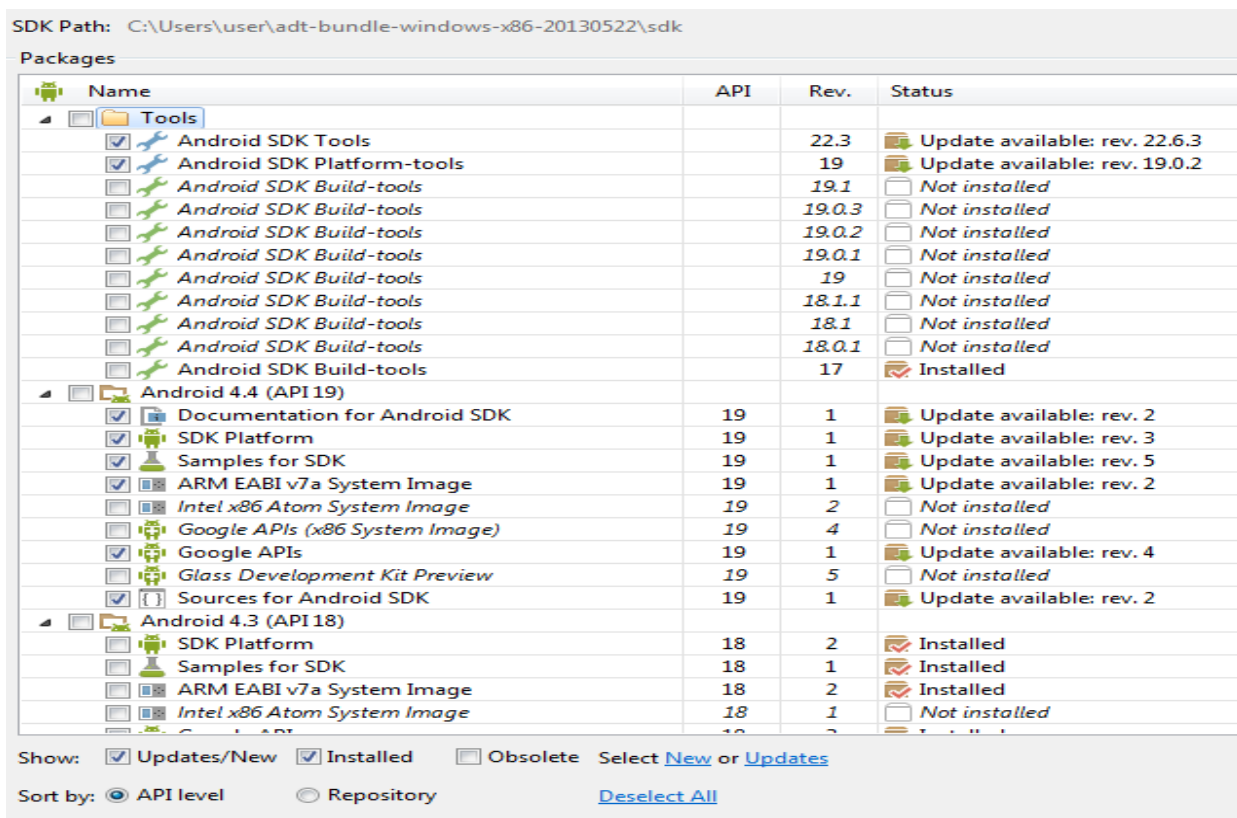
Το Android SDK είναι το επίσημο εργαλείο της Google για αυτούς που θέλουν να δημιουργήσουν εφαρμογές στο Android. Είναι το βασικό εργαλείο για δημιουργία εφαρμογών, κατασκευή custom ROM, μεταγλώττιση πυρήνα για την συσκευή και γενικό hacking.

Μέσω του SDK μπορούν να χρησιμοποιηθούν εργαλεία όπως το ADB, για μεταφορά αρχείων σε χώρους που κανονικά δεν επιτρέπεται και να γίνει χρήση του fastboot για εγκατάσταση custom recovery εικονών και ξεκλείδωμα του bootloader της συσκευής. Κυρίως όμως, μπορούν να κατασκευαστούν προγράμματα για το Android αρκεί να υπάρχουν οι κατάλληλες γνώσεις Java προγραμματισμού για να γίνουν με επιτυχία.

Το Android SDK παρέχει τις βιβλιοθήκες API και τα developer tools που είναι απαραίτητα για την κατασκευή, τη δοκιμή και τον εντοπισμό σφαλμάτων για συστήματα που τρέχουν σε λειτουργικό σύστημα Android.

Εδώ παρατίθεται ο σύνδεσμος στον οποίο μπορείτε να διατρέξετε για να κατεβάσετε τον Android SDK Manager που φαίνεται στην πιο κάτω εικόνα:

<http://developer.android.com/sdk/index.html?hl=sk>



3.4 Γλώσσα Προγραμματισμού C#

Η γλώσσα προγραμματισμού C# αποτελεί βελτίωση της γλώσσας προγραμματισμού C και από την αρχή προοριζόταν να είναι απλή, μοντέρνα, γενικής χρήσης και αντικειμενοστρεφής γλώσσα προγραμματισμού. Η πιο καινούργια έκδοση είναι η C# 5.0 και κυκλοφόρησε στις 15 Αυγούστου 2012 και σχεδιάστηκε από την Microsoft. Η C# μοιάζει στην σύνταξη με την γλώσσα προγραμματισμού Java και διαθέτει αρκετές βιβλιοθήκες που είναι πολύ χρήσιμες για τους προγραμματιστές αφού θα τους προσφέρει καλό, γρήγορο και εύκολο στην χρήση κώδικα για τα συστήματα που αναπτύσσουν.

Το Web Service που υλοποιήθηκε για την επικοινωνία μεταξύ της εφαρμογής και της Βάσης Δεδομένων είναι γραμμένο στην γλώσσα προγραμματισμού C#, χρησιμοποιεί την αρχιτεκτονική REST και είναι σχεδιασμένο με το WCF εργαλείο. Ανάλογα με τα δεδομένα που χρειαζόμαστε από την Βάση στέλνεται ένα query από την εφαρμογή και ελέγχεται από το Web Service, το οποίο αναγνωρίζει τον τύπο του και αναλόγως εκτελεί το query στην Βάση Δεδομένων. Στην συνέχεια, το αποτέλεσμα από το query κωδικοποιείται σε JSON μορφή και ακολούθως επιστρέφεται στην εφαρμογή, όπου γίνεται η αποκωδικοποίηση του, το διάβασμα των δεδομένων και η κατάλληλη εισδοχή τους στους πίνακες της εφαρμογής.

Παράδειγμα της JSON μορφής:

```
[
  {
    "id": 912345678901,
    "text": "How do I read JSON on Android?",
    "geo": null,
    "user": {
      "name": "android_newb",
      "followers_count": 41
    }
  },
  {
    "id": 912345678902,
    "text": "@android_newb just use
android.util.JsonReader!",
    "geo": [50.454722, -104.606667],
    "user": {
      "name": "jesse",
      "followers_count": 2
    }
  }
]
```

3.5 Microsoft SQL Server 2008

Ο **SQL Server** (προφέρεται «σίκουελ σέρβερ») είναι μια σχεσιακή βάση δεδομένων, η οποία αναπτύσσεται από τη Microsoft και βγήκε για πρώτη φορά στην αγορά το 1989 σε συνεργασία με την Sybase. Οι κύριες γλώσσες που χρησιμοποιούνται είναι η T-SQL και η ANSI SQL. Η κύρια μονάδα αποθήκευσης στοιχείων είναι μια βάση δεδομένων, η οποία αποτελείται από μια συλλογή πινάκων και κώδικα.

Ένα από τα σημαντικότερα πλεονεκτήματα του Microsoft SQL Server 2008 είναι ότι περιλαμβάνει ένα module που λέγεται SQL CLR μέσω του οποίου ενσωματώνει το .NET μέσα στον SQL Server. Με το SQL CLR, οι αποθηκευμένες διαδικασίες μπορούν να γραφτούν σε οποιαδήποτε γλώσσα .NET συμπεριλαμβανομένου C# και VB.NET και να δημιουργήσουν μια stored procedure. Αυτό σημαίνει ότι ο SQL Server έχει όλες τις βιβλιοθήκες και πλεονεκτήματα του .NET, αυτόχθονα μέσα στο περιβάλλον του, τα οποία μπορεί να τα καλέσει οποιαδήποτε στιγμή.

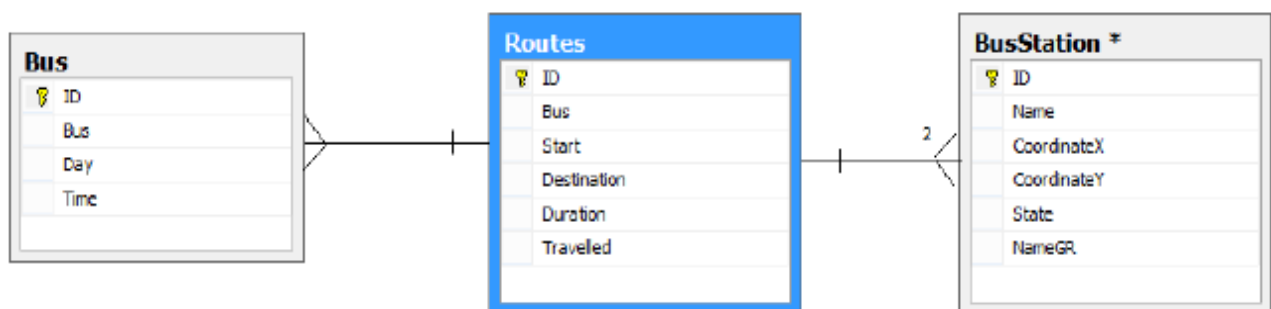
Για την ανάκτηση των δεδομένων από μια βάση δεδομένων, ένας κύριος τρόπος που χρησιμοποιείται είναι η επερώτηση (query).

Παράδειγμα επερώτησης που παίρνει όλα τα δεδομένα που αφορούν τις στάσεις των λεωφορείων:

```
query_station = "SELECT ID, Name, CoordinateX, CoordinateY, State FROM BusStation";  
result = data.execute(query_station).get();
```

Χρησιμοποιώντας τον SQL Server 2008 σχεδιάστηκε η βάση δεδομένων της εφαρμογής. Όπως είχα αναφέρει στην ενότητα 2.4, η βάση δεδομένων αποτελείται από τους πίνακες BusStation, Bus και Routes. Σε κάθε πίνακα το ID που υπάρχει σαν πεδίο αποτελεί το πρωτεύον κλειδί του πίνακα.

Database Diagram



Κεφάλαιο 4

Εγχειρίδιο Χρήσης Εφαρμογής

4.1 Αρχική Οθόνη	33
4.1.1 Αλλαγή αρχικής τοποθεσίας	35
4.1.2 Αλλαγή τελικής τοποθεσίας	36
4.1.3 Αλλαγή ημερομηνίας	37
4.1.4 Αλλαγή ώρας	38
4.1.5 Εκτέλεση Αλγορίθμου	39
4.2 Οθόνη Επιλογής Διαδρομής	39
4.3 Οθόνη Λεπτομερειών Συγκεκριμένης Διαδρομής	40
4.3.1 Χρονοδιάγραμμα Διαδρομής	40
4.3.2 Χάρτης Διαδρομής	41
4.3.3 Ενημέρωση για διαδρομή από άλλες πηγές	41

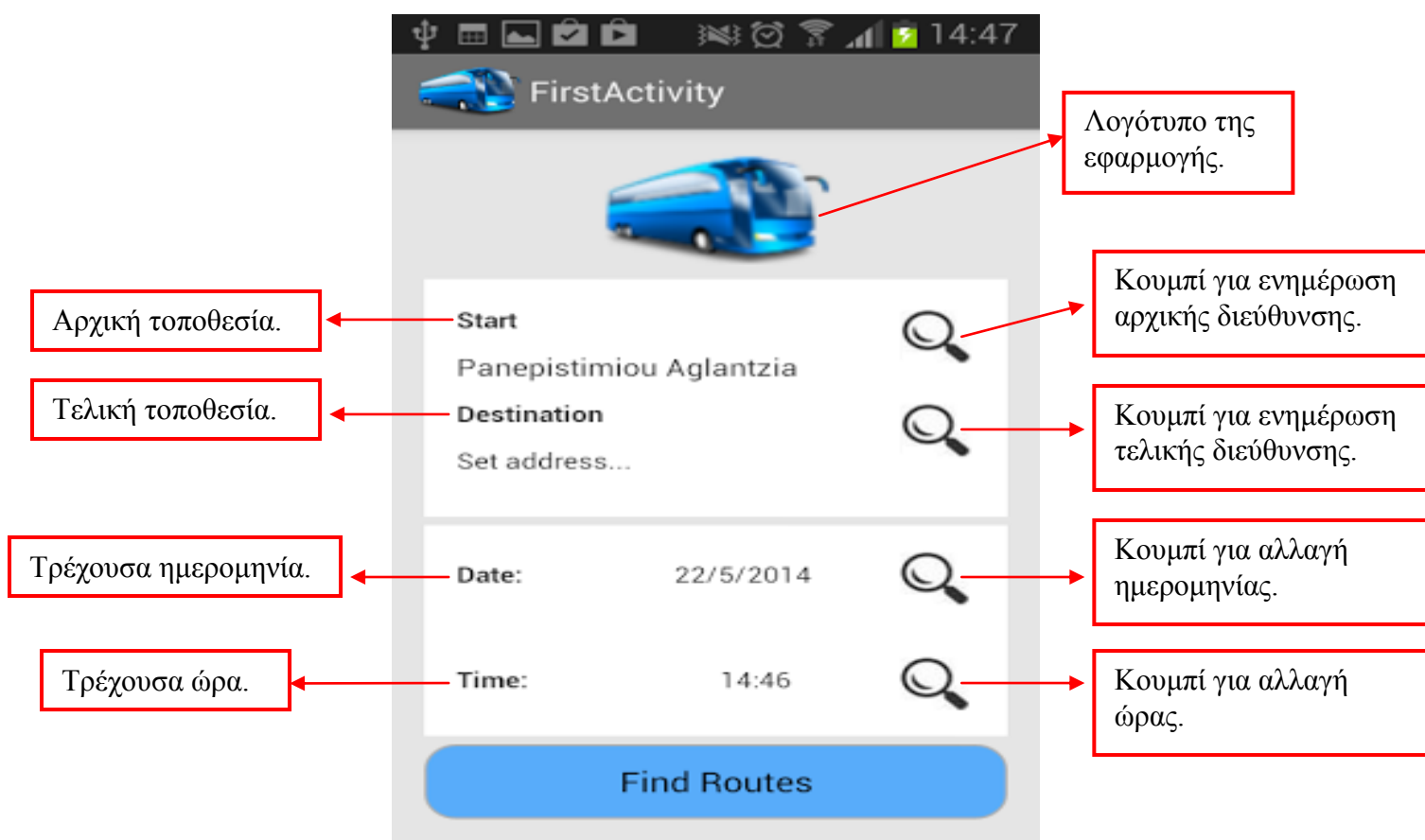
Σε αυτό το κεφάλαιο παρουσιάζεται η μέχρι τώρα μορφή της εφαρμογής «Cyprus Bus Routes» με όλες τις λειτουργίες που υλοποιήθηκαν στην παρούσα διπλωματική εργασία. Η κάθε λειτουργία αναλύεται στην οθόνη που έχει υλοποιηθεί και μπορεί να την εκτελέσει ο χρήστης διατρέχοντας την συγκεκριμένη οθόνη.

4.1 Αρχική οθόνη

Επιλέγοντας την εφαρμογή από την λίστα εφαρμογών που υπάρχουν πάνω στη κινητή συσκευή, εμφανίζεται μετά από μερικά δευτερόλεπτα (μέχρι να ληφθούν και να καταχωρηθούν όλα τα δεδομένα από την βάση δεδομένων και να εισαχθούν στους πίνακες της εφαρμογής) η πιο κάτω οθόνη. Στην οθόνη αυτή καταχωρούνται όλα τα στοιχεία που επιθυμεί ο χρήστης να διαθέτει η διαδρομή αυτή.

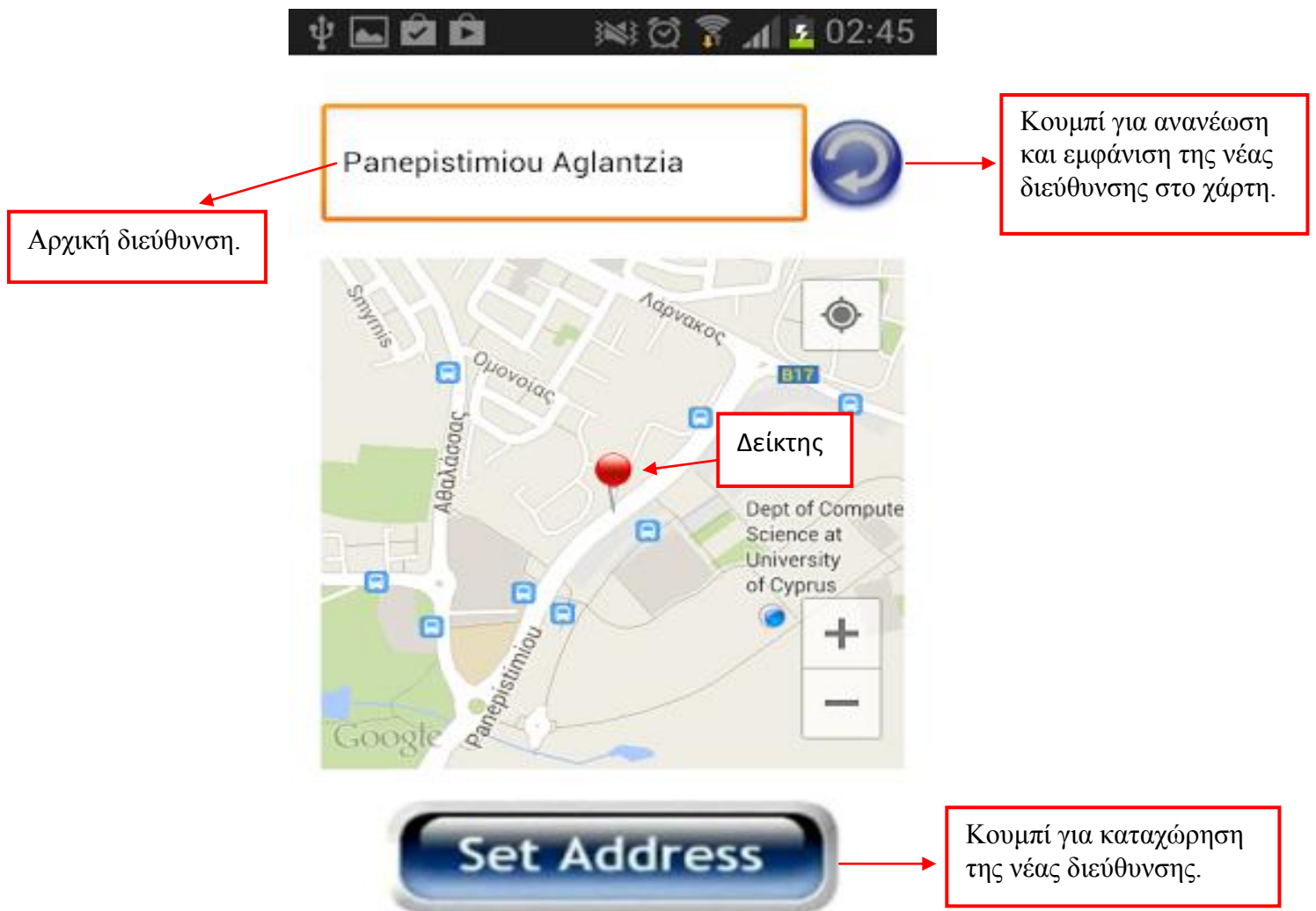
Τα στοιχεία αυτά είναι η αρχική διεύθυνση από την οποία θέλει να πάρει το λεωφορείο, η τελική διεύθυνση στην οποία θέλει να κατεβεί και η ημερομηνία και η ώρα που θέλει να πραγματοποιήσει την διαδρομή.

Τα προκαθορισμένα στοιχεία για την ημερομηνία και την ώρα είναι τα τρέχοντα δεδομένα που λαμβάνονται από την συγκεκριμένη ημερομηνία και ώρα της εφαρμογής. Για την αρχική διεύθυνση, η διεύθυνση που υπάρχει ήδη καταχωρημένη στην αρχική οθόνη προτού ο χρήστης εκτελέσει οποιαδήποτε ενέργεια είναι η τρέχουσα τοποθεσία του χρήστη, η τοποθεσία δηλαδή που βρίσκεται αυτήν την στιγμή. Αυτό ήταν σημαντική απαίτηση όλων των πιθανών χρηστών με τους οποίους είχα συζητήσει αρχικά προτού δημιουργήσω αυτήν την εφαρμογή. Θεωρούσαν απαραίτητο, ιδιαίτερα για τους τουρίστες να φαίνεται η τρέχουσα τοποθεσία του χρήστη για να γνωρίζει ο χρήστης οποιαδήποτε στιγμή που βρίσκεται και πόσο μακριά είναι η διεύθυνση στην οποία επιθυμεί να πάει για να πάρει το λεωφορείο. Για την τελική διεύθυνση δεν υπάρχει οποιαδήποτε προκαθορισμένη τιμή αφού δεν μπορούμε να γνωρίζουμε εκ των προτέρων τον πιθανό προορισμό του χρήστη.

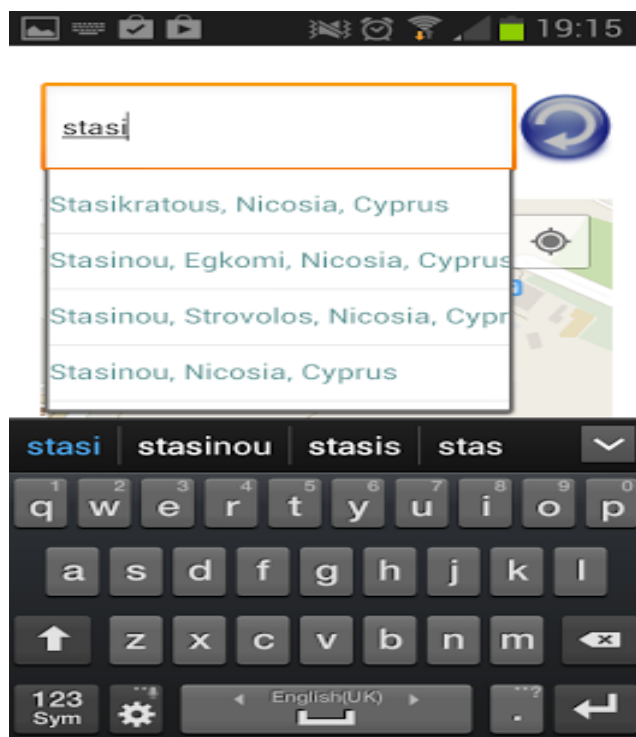


4.1.1 Αλλαγή αρχικής τοποθεσίας

Ο χρήστης πατώντας στο εικονίδιο αναζήτησης που βρίσκεται στο δεξί μέρος της ετικέτας «Start» εισέρχεται στην πιο κάτω οθόνη, στην οποία έχει την δυνατότητα να ανανεώσει και να δει με την βοήθεια ενός δείκτη μια διεύθυνση. Αρχικά, ο χρήστης βλέπει στο χάρτη την προκαθορισμένη τιμή της αρχικής διεύθυνσης που είναι η τρέχουσα του διεύθυνση. Για να την αλλάξει, χρειάζεται να πληκτρολογήσει την νέα διεύθυνση και στην συνέχεια να πατήσει στο εικονίδιο «Refresh» που βρίσκεται στο πάνω δεξιό μέρος για να την δει στο χάρτη. Τέλος, πατώντας στο κουμπί «Set Address» μπορεί να την καταχωρήσει ως αρχική διεύθυνση της διαδρομής.



Κατά την διάρκεια πληκτρολόγησης της διεύθυνσης στο πλαίσιο εισαγωγής τοποθεσίας, φιλτράρονται και εμφανίζονται πιθανές διευθύνσεις που μπορεί να επιθυμεί ο χρήστης. Στην συνέχεια, πατώντας πάνω σε μία από αυτές μπορεί να την καταχωρήσει με επιτυχία στο πλαίσιο. Οι διευθύνσεις αυτές δίνονται στην εφαρμογή μέσω του Google Places Api που χρησιμοποιείται για τον εντοπισμό λεπτομερών πληροφοριών σχετικά με τοποθεσίες ανάμεσα σε ένα ευρύ φάσμα κατηγοριών.



4.1.2 Αλλαγή Τελικής Τοποθεσίας

Με τον ίδιο τρόπο θα χρειαστεί να συμπληρώσει και την τελική του τοποθεσία για την οποία όμως δεν υπάρχει προκαθορισμένη τιμή συμπληρωμένη στο πεδίο. Όταν ο χρήστης ολοκληρώσει την διαδικασία εισαγωγής της τελικής του τοποθεσίας και πατήσει το κουμπί «Set Address», θα οδηγηθεί στην προηγούμενη οθόνη στην οποία η τοποθεσία αυτή θα καταχωρηθεί στο πλαίσιο της τελικής τοποθεσίας. Αν ο χρήστης δεν συμπληρώσει την τελική του τοποθεσία και προσπαθήσει να προχωρήσει παρακάτω η εφαρμογή δεν θα του το επιτρέψει και θα του εμφανίζει ένα μήνυμα που θα τον ενημερώνει ότι χρειάζεται να την συμπληρώσει.

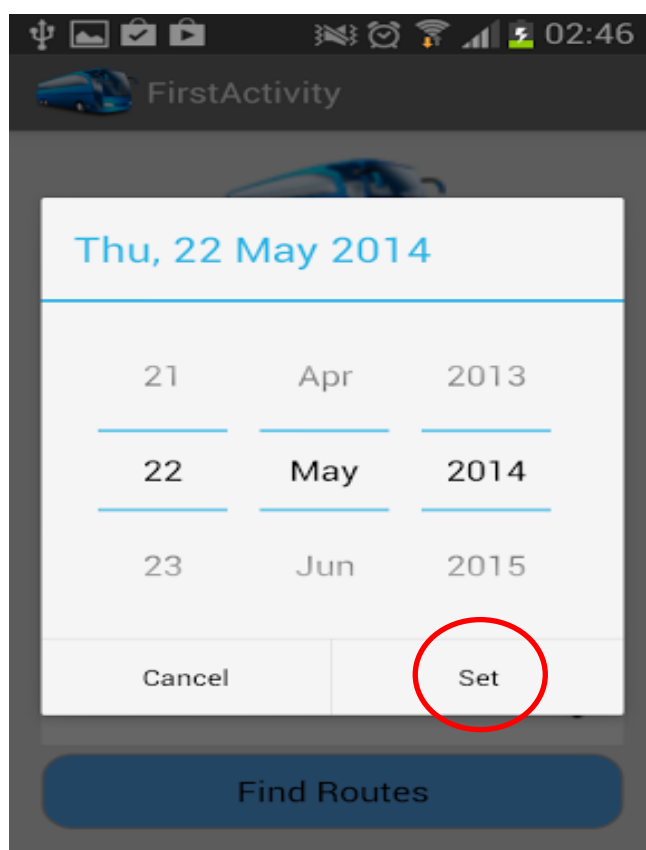


4.1.3 Αλλαγή ημερομηνίας

Ο χρήστης μπορεί να αλλάξει την ημερομηνία της διαδρομής που επιθυμεί να πραγματοποιήσει, πατώντας στο εικονίδιο αναζήτησης που βρίσκεται στο δεξιό μέρος της ετικέτας «Date».

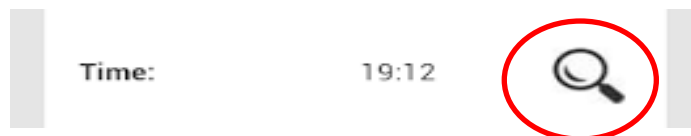


Στην συνέχεια, αφού το κάνει αυτό, θα οδηγηθεί σε μία άλλη οθόνη στην οποία θα μπορεί να καταχωρήσει την ημερομηνία με δύο τρόπους. Ο πρώτος τρόπος είναι πατώντας πάνω στο κάθε πεδίο και αλλάζοντας το από το πληκτρολόγιο ή κάνοντας ολίσθηση προς τα πάνω ή κάτω(scrolling) μέχρι να τοποθετηθεί αυτό που επιθυμεί στο κέντρο. Όταν τελειώσει πατά στο κουμπί «Set» για να καταχωρηθεί η ημερομηνία της διαδρομής με επιτυχία στην προηγούμενη οθόνη.

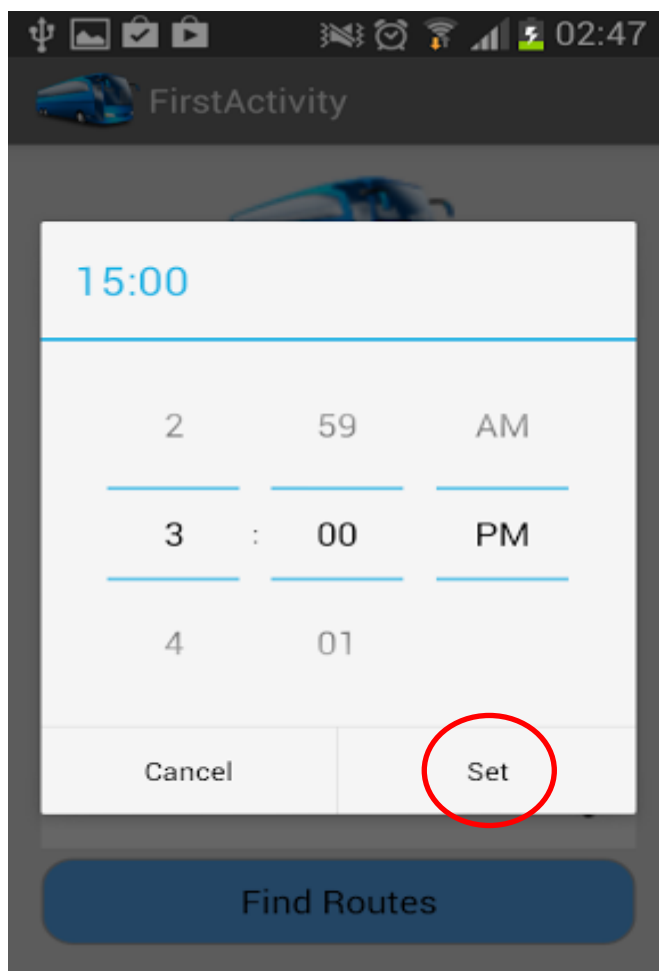


4.1.4 Αλλαγή ώρας

Ο χρήστης μπορεί να αλλάξει την ώρα της διαδρομής που επιθυμεί να πραγματοποιήσει, πατώντας στο εικονίδιο αναζήτησης που βρίσκεται στο δεξιό μέρος της ετικέτας «Time».

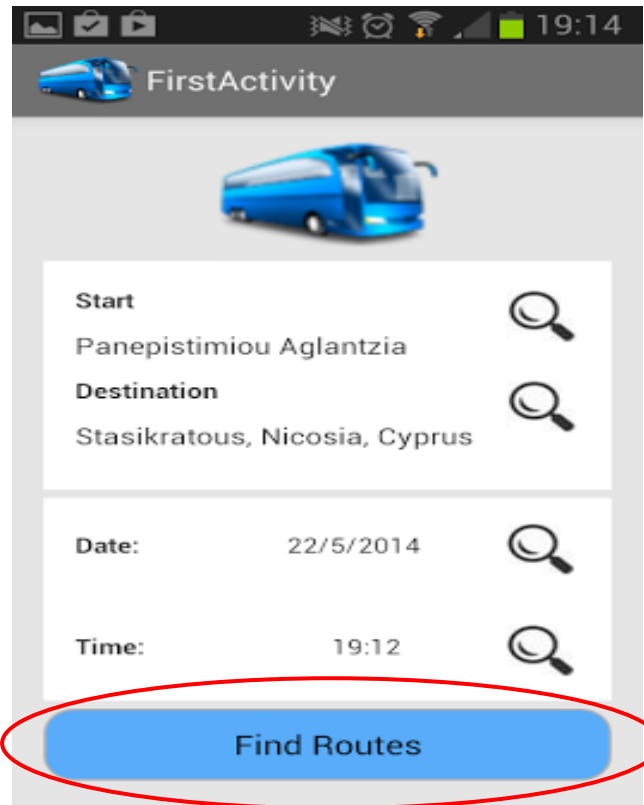


Στην συνέχεια, αφού το κάνει αυτό, θα οδηγηθεί σε μία άλλη οθόνη στην οποία θα μπορεί να καταχωρήσει την ώρα πάλι με δύο τρόπους. Οι δύο αυτοί τρόποι είναι ακριβώς οι ίδιοι με αυτούς που χρησιμοποιούνται για την καταχώρηση της ημερομηνίας. Όταν τελειώσει πατά στο κουμπί «Set» για να καταχωρηθεί η ώρα της διαδρομής με επιτυχία στην προηγούμενη οθόνη.



4.1.5 Εκτέλεση Αλγορίθμου

Αφού όλα τα στοιχεία της διαδρομής, έχουν καταχωρηθεί με επιτυχία, ο χρήστης μπορεί να προχωρήσει στην εκτέλεση του αλγορίθμου πατώντας στο πιο κάτω κουμπί. Όταν πατηθεί, ο χρήστης θα μεταφερθεί στην οθόνη επιλογής διαδρομής.



4.2 Οθόνη επιλογής διαδρομής

Στην οθόνη επιλογής διαδρομής, παρουσιάζεται η κοντινότερη διαδρομή που βρέθηκε από την εκτέλεση του αλγορίθμου και πληρεί τα στοιχεία που είχε καταχωρήσει ο χρήστης στην προηγούμενη οθόνη με κάποιες γενικές πληροφορίες για αυτήν.

Πιο συγκεκριμένα στην οθόνη αυτή παρουσιάζονται τα εξής:

- ✓ Στο πάνω μέρος της οθόνης, παρουσιάζεται τόσο η αρχική και τελική διεύθυνση της διαδρομής, όσο και η ημερομηνία και ώρα που θα πραγματοποιηθεί.
- ✓ Στο κάτω μέρος παρουσιάζεται η κοντινότερη διαδρομή που έχει εντοπιστεί με το χρονικό διάστημα που θέλει για να πραγματοποιηθεί και τον αριθμό των στάσεων από τις οποίες θα περάσει για να καταλήξει το λεωφορείο στον προορισμό του.
- ✓ Για να μπορέσει ο χρήστης να έλθει εις γνώση με περισσότερες πληροφορίες σχετικά με την διαδρομή χρειάζεται απλά να πατήσει επάνω της.



4.3 Οθόνη Λεπτομερειών Συγκεκριμένης Διαδρομής

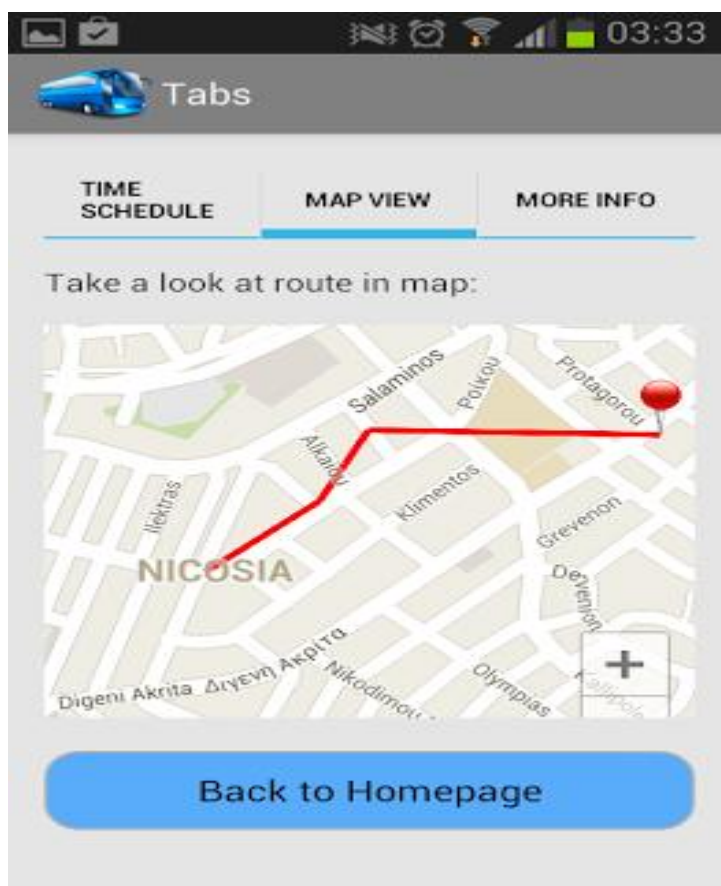
4.3.1 Χρονοδιάγραμμα διαδρομής

Στην οθόνη με το χρονοδιάγραμμα παρουσιάζονται κάποιες χρήσιμες πληροφορίες για τον χρήστη που του εξηγούν με λεπτομέρεια την διαδρομή του.

Πιο συγκεκριμένα, στην οθόνη αυτή ο χρήστης μπορεί να δει τον συνολικό χρόνο της διαδρομής, την αρχική στάση και ώρα που ξεκινά η διαδρομή και πόσο χρόνο χρειάζεται αρχικά για να φτάσει στην στάση αυτή. Επίσης, φαίνονται καθαρά όλες οι στάσεις από τις οποίες θα περάσει το λεωφορείο για να διεκπεραιώσει την συγκεκριμένη διαδρομή και η ώρα που θα φτάσει στον προορισμό του ο χρήστης μαζί με την τελική στάση που θα πρέπει να κατεβεί από αυτό. Τέλος, απαραίτητη λειτουργία που υποστηρίζεται σε αυτήν την οθόνη είναι το τύπωμα της διαδρομής. Επειδή οι λειτουργίες της εφαρμογής είναι διαθέσιμες μόνο με την σύνδεση στο διαδίκτυο, ο χρήστης θα έχει την ευκαιρία να πατά στο εικονίδιο αποθήκευσης και να αποθηκεύει την διαδρομή που επέλεξε στην κινητή του συσκευή.

4.3.2 Χάρτης Διαδρομής

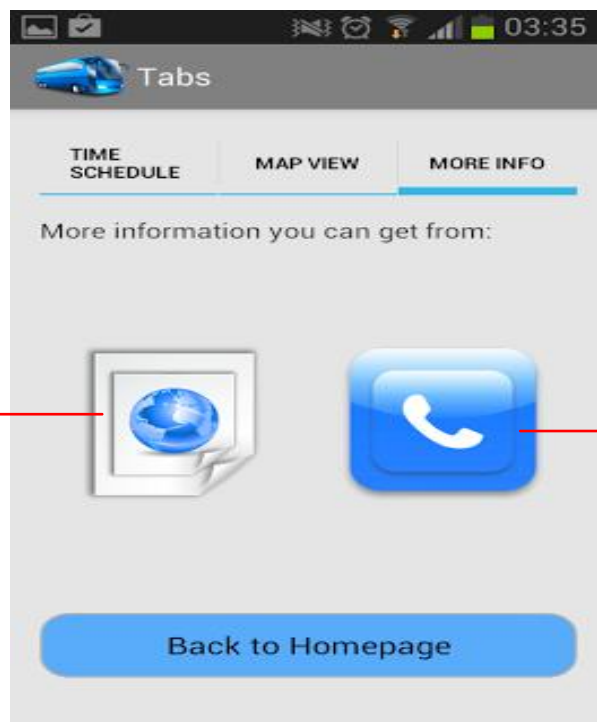
Ο χρήστης πατώντας στον σύνδεσμο «MAP VIEW» έχει την δυνατότητα προβολής της διαδρομής που επέλεξε στο χάρτη, βλέποντας ξεκάθαρα όλες τις στάσεις που θα περάσει το λεωφορείο κάνοντας αυτή την διαδρομή για να φτάσει στον προορισμό του. Επιπρόσθετα, έχει την δυνατότητα να επιστρέψει με ένα γρήγορο και εύκολο τρόπο πίσω στην αρχική οθόνη αν το επιθυμεί και να καταχωρήσει τα στοιχεία άλλης διαδρομής.



4.3.3 Ενημέρωση για διαδρομή από άλλες πηγές

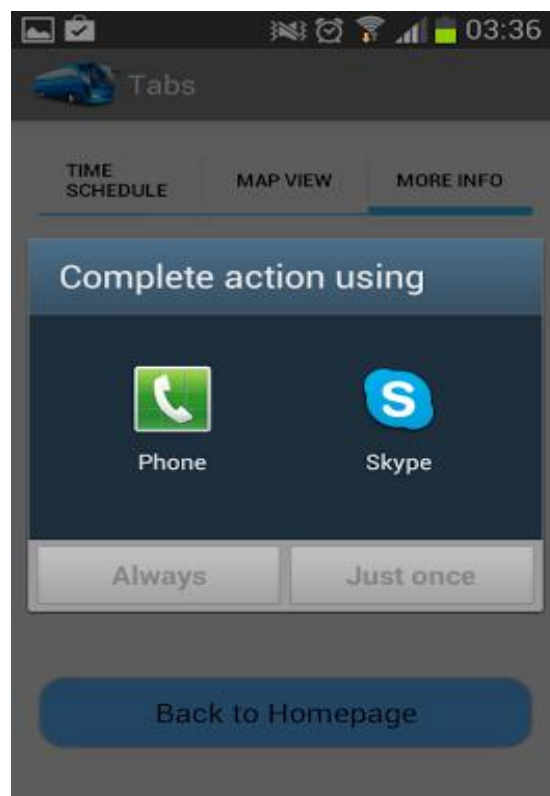
Αυτή η εφαρμογή παρέχει στον χρήστη την δυνατότητα να λάβει περαιτέρω πληροφορίες σχετικά με μια διαδρομή από άλλες πηγές. Αν το επιθυμεί αυτό ο χρήστης μπορεί να πατήσει στον σύνδεσμο «MORE INFO» και να επιλέξει από την οθόνη που θα του εμφανιστεί, την πηγή από την οποία θα ήθελε να λάβει τις πληροφορίες αυτές. Εδώ προσφέρονται δύο πηγές, ένα τηλέφωνο και ένας ιστότοπος.

Κουμπί για εμφάνιση ιστοσελίδας.



Κουμπί για κάλεσμα τηλεφώνου γενικών πληροφοριών.

Εδώ ο χρήστης μπορεί να πατήσει είτε στο πρώτο εικονίδιο και να πλοηγηθεί στην ιστοσελίδα «CyprusByBus» είτε στο δεύτερο εικονίδιο και να καλέσει το τηλέφωνο των γενικών πληροφοριών για λεωφορεία στην Λάρνακα και να μάθει όλα όσα χρειάζεται επιπλέον για μια συγκεκριμένη διαδρομή.



Κεφάλαιο 5

Δοκιμή και Αξιολόγηση Εφαρμογής

5.1 Δοκιμή Εφαρμογής	43
5.2 Δημιουργία Ερωτηματολογίου	44
5.3 Ανάλυση Αποτελεσμάτων	46

Σε αυτό το κεφάλαιο γίνεται αναφορά στην διαδικασία που τηρήθηκε για δοκιμή της εφαρμογής και γίνεται η ανάλυση των αποτελεσμάτων που εξάχθηκαν από αυτή.

5.1 Δοκιμή Εφαρμογής

Μετά την ολοκλήρωση της υλοποίησης, ακολούθησαν εκτενείς δοκιμές για την συμπεριφορά της ολοκληρωμένης εφαρμογής με όλα τα συστατικά μέρη συνενωμένα. Μετά από τις δοκιμές, η εφαρμογή δόθηκε σε επιβάτες λεωφορείων για αξιολόγηση. Αφότου δοκίμασαν την εφαρμογή τους ζητήθηκε να συμπληρώσουν το ακόλουθο ερωτηματολόγιο που περιέχει ερωτήσεις σχετικά με την εμπειρία χρήσης της εφαρμογής.

Ερωτηματολόγιο για την εφαρμογή «Cyprus Bus Routes»

Παρακαλώ όπως υποδείξετε σε ποιο βαθμό συμφωνείτε ή διαφωνείτε με τις επόμενες δηλώσεις (1 : Διαφωνώ πλήρως, 2 : Διαφωνώ, 3 : Ούτε συμφωνώ ούτε διαφωνώ, 4 : Συμφωνώ, 5 : Συμφωνώ πλήρως)

*** Required**

1. Η εφαρμογή είναι εύκολη στην χρήση *

1 2 3 4 5

Διαφωνώ πλήρως ☐ ☐ ☐ ☐ ☐ Συμφωνώ πλήρως

2. Μπορώ να μάθω να χρησιμοποιώ όλες τις λειτουργίες της εφαρμογής εύκολα *

1 2 3 4 5

Διαφωνώ πλήρως ☐ ☐ ☐ ☐ ☐ Συμφωνώ πλήρως

3. Η περιήγηση μέσω των μενού της εφαρμογής είναι εύκολη *

1 2 3 4 5

Διαφωνώ πλήρως ☐ ☐ ☐ ☐ ☐ Συμφωνώ πλήρως

4. Τα μενού και υπομενού της εφαρμογής είναι σωστά δομημένα *

1 2 3 4 5

Διαφωνώ πλήρως ☐ ☐ ☐ ☐ ☐ Συμφωνώ πλήρως

5. Όλα τα αντικείμενα στην εφαρμογή (κουμπιά, textboxes κτλ.) είναι σωστά τοποθετημένα και η λειτουργία τους είναι εύκολα κατανοητή *

1 2 3 4 5

Διαφωνώ πλήρως ☐ ☐ ☐ ☐ ☐ Συμφωνώ πλήρως

6. Η εφαρμογή επιστρέφει κατάλληλα μηνύματα σύμφωνα με τις ενέργειες του χρήστη *

1 2 3 4 5

Διαφωνώ πλήρως ☐ ☐ ☐ ☐ ☐ Συμφωνώ πλήρως

7. Η εφαρμογή είναι αποδοτική και ο χρόνος απόκρισης μετά από τις ενέργειες του χρήστη είναι μικρός *

1 2 3 4 5

Διαφωνώ πλήρως ☐ ☐ ☐ ☐ ☐ Συμφωνώ πλήρως

8. Η εφαρμογή διαθέτει όλες τις λειτουργίες που περίμενα να έχει *

1 2 3 4 5

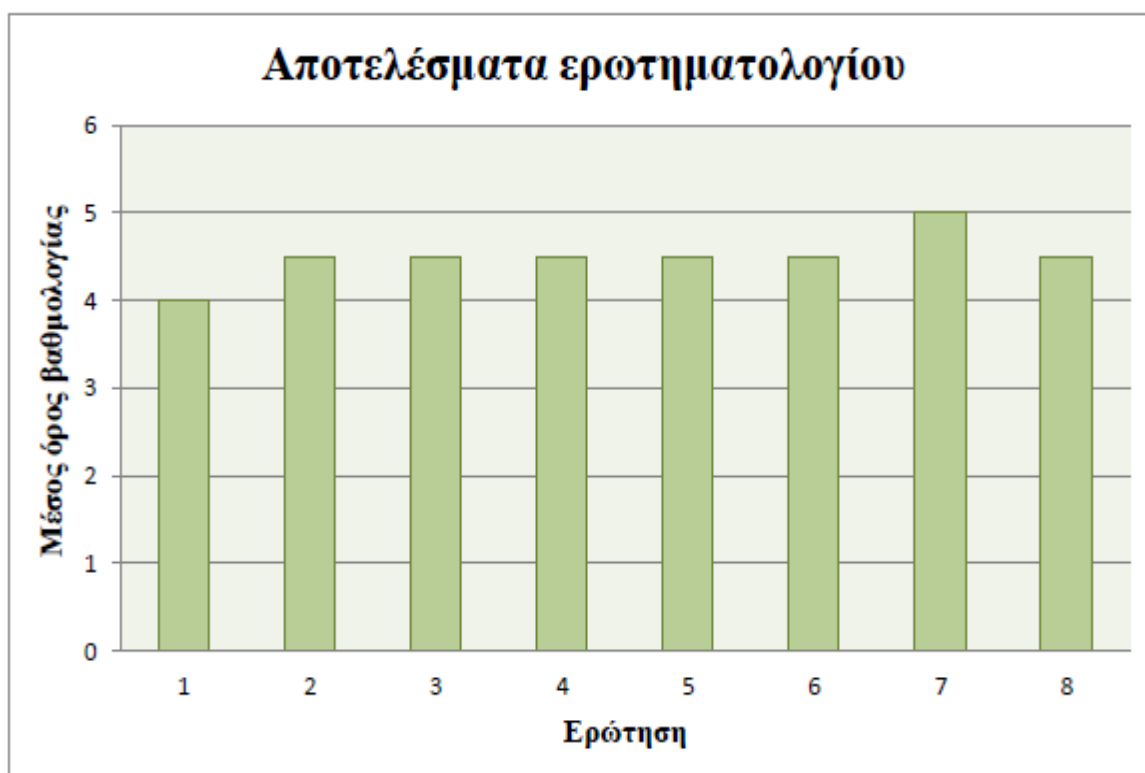
Διαφωνώ πλήρως ☐ ☐ ☐ ☐ ☐ Συμφωνώ πλήρως

9. Υπάρχει κάποια λειτουργία που δεν διαθέτει η εφαρμογή και πιστεύετε πως θα ήταν χρήσιμη;

10. Υπάρχει κάποια λειτουργία που διαθέτει η εφαρμογή και πιστεύετε πως δεν είναι χρήσιμη;

11. Επιπλέον σχόλια σχετικά με την εφαρμογή

5.3 Ανάλυση Αποτελεσμάτων



Οι ερωτήσεις που περιέχονται στο ερωτηματολόγιο καλύπτουν πολλές πτυχές της εφαρμογής, όπως ευχρηστία, αποδοτικότητα, δομή και πληρότητα. Έτσι από τις απαντήσεις στις ερωτήσεις αυτές μπορούν να εξαχθούν χρήσιμα αποτελέσματα για ολόκληρη την εφαρμογή. Τα αποτελέσματα από τις οκτώ ερωτήσεις είναι πολύ ενθαρρυντικά μιας και ο μέσος όρος βαθμολογίας όλων των ερωτήσεων κυμαίνεται μεταξύ 4 και 5. Ψηλότερη βαθμολογία έχει η ερώτηση 7 που σχετίζεται με την αποδοτικότητα της εφαρμογής, ενώ χαμηλότερη βαθμολογία έχει η ερώτηση 1 που αφορά την ευχρηστία της εφαρμογής, δείχνοντας πως ίσως τα μεγαλύτερα περιθώρια βελτίωσης αφορούν εκείνο τον τομέα. Την ευχρηστία καθορίζουν επίσης οι ερωτήσεις 2 και 6 που έχουν και οι δύο μέσο όρο βαθμολογίας 4.5. Οι ερωτήσεις 3, 4 και 5 αφορούν κυρίως την δομή της εφαρμογής. Και οι 3 αυτές ερωτήσεις έχουν μέσο όρο βαθμολογίας 4.5, οπότε μπορούμε να συμπεράνουμε πως οι χρήστες βρήκαν την δομή της εφαρμογής αρκετά καλή. Η ερώτηση 8, καθώς επίσης και οι ερωτήσεις με σχόλια, έχουν να κάνουν με την πληρότητα της εφαρμογής. Ο μέσος όρος βαθμολογίας είναι επίσης 4.5 οπότε και πάλι μπορούμε να συμπεράνουμε πως οι χρήστες βρήκαν αρκετά πλήρη την εφαρμογή. Σύμφωνα με τις απαντήσεις των χρηστών στο ερωτηματολόγιο, φαίνεται πως βρήκαν αρκετά εύχρηστη, καλά δομημένη και πλήρη την

εφαρμογή ενώ έμειναν πολύ ικανοποιημένοι από την απόδοση της. Σίγουρα πάντοτε υπάρχουν περιθώρια βελτίωσης σε όλους τους τομείς..

Επίσης, πέρα από τις βαθμολογίες λήφθηκαν χρήσιμες πληροφορίες και από τις τρεις τελευταίες ερωτήσεις. Έγιναν σχόλια που αφορούσαν περαιτέρω λειτουργίες που θα έπρεπε να διαθέτει η εφαρμογή όπως ή προβολή κόστους εισιτηρίου και η κράτηση αυτού, συμπληρώνοντας προσωπικά στοιχεία και τυπώνοντας τα μαζί με ένα αντίγραφο barcode. Επιπρόσθετα, οι περισσότεροι ανέφεραν την πρόσθεση λεπτομερούς χρονοδιαγράμματος που θα εξηγεί ακριβώς σε ποια στάση θα κατεβεί ο χρήστης για να πάρει το επόμενο λεωφορείο, την ώρα που θα το κάνει αυτό και ποιο είναι το συγκεκριμένο λεωφορείο, τονίζοντας ότι είναι ιδιαίτερα σημαντικό για αυτούς να μην αναζητούν αυτές τις πληροφορίες στην συνέχεια από άλλες πηγές. Τέλος, σχολίασαν ότι δεν υπάρχει κάποια λειτουργία που έχει υλοποιηθεί και υποστηρικτεί από την εφαρμογή και δεν είναι χρήσιμη.

Το μόνο που είχε δεχτεί αρνητικά σχόλια, είναι η καθυστέρηση που προκαλείται στην αρχή μέχρι να ανοίξει η εφαρμογή. Αυτό τους ενόχλησε λίγο αφού φάνηκε μικρή δυσφορία στα πρόσωπα τους κατά την αναμονή.

Κεφάλαιο 6

Συμπεράσματα

6.1 Συμπεράσματα	48
6.2 Μελλοντικές Προεκτάσεις	50

Με αυτό το κεφάλαιο, η διπλωματική μου εργασία φτάνει στο τέλος της. Σε αυτό το σημείο θα ήθελα να σας αναφέρω τα συμπεράσματα που έχουν εξαχθεί από την ολοκλήρωση της και τις μελλοντικές προεκτάσεις που θα ήταν καλό να γίνουν για να μπορεί η εφαρμογή να εισαχθεί επιτυχώς στην αγορά των ηλεκτρονικών εφαρμογών και να χρησιμοποιηθεί από πολλούς χρήστες.

6.1 Συμπεράσματα

Στην παρούσα διπλωματική εργασία αναπτύχθηκε η εφαρμογή «Cyprus Bus Routes» η οποία έχει ως απώτερο σκοπό την διευκόλυνση των επιβατών λεωφορείων στην επιλογή της κοντινότερης διαδρομής με βάση κάποια συγκεκριμένα στοιχεία που επιθυμούν να διαθέτει η διαδρομή αυτή για την διακίνηση τους στο όμορφο νησί.

Η εφαρμογή αυτή έχει σχεδιαστεί και υλοποιηθεί με επιτυχία με βάση κάποιες απαιτήσεις που καθορίστηκαν από την αρχή από τον πελάτη και κάποιους μελλοντικούς χρήστες του συστήματος, για να είναι όσο το δυνατό πιο χρήσιμη και εύχρηστη για αυτούς. Εμφανίζει με επιτυχία την κοντινότερη διαδρομή και δίνει κάποιες λεπτομερείς πληροφορίες σχετικά με αυτήν για να μπορεί ο χρήστης να την κατανοήσει και να είναι έτοιμος να την ακολουθήσει. Επίσης, παρέχει την δυνατότητα στον χρήστη να τυπώνει την διαδρομή του αφού μπορεί να χρειαστεί να την δει και πιο μετά χωρίς να αναγκάζεται να είναι συνδεδεμένος στο διαδίκτυο.

Η διεπιφάνεια του συστήματος άφησε τις καλύτερες εντυπώσεις, αφού χρησιμοποιήθηκαν συστατικά λογισμικού που είναι φιλικά και ευχάριστα για τον χρήστη. Είναι μια αρκετά απλή εφαρμογή και μπορεί εύκολα να την χρησιμοποιήσει οποιοσδήποτε χωρίς να είναι

έμπειρος από πριν. Επιπλέον, όλες οι οθόνες παρέχουν εύκολο τρόπο επιστροφής στην αρχική οθόνη σε περίπτωση που ο χρήστης επιθυμεί να δει άλλη διαδρομή τελικά. Έτσι ο χρήστης μπορεί να διατρέχει όσες φορές θέλει εύκολα και γρήγορα όλη την διεπιφάνεια του συστήματος και να βλέπει διαδρομές.

Σχετικά με τα εργαλεία και τις γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής αυτής, το μόνο που προκάλεσε λίγο πρόβλημα στην εφαρμογή και αγανάκτηση στους χρήστες είναι η χρήση του Web Service που υπήρχε υλοποιημένο από πριν και παίρνει όλα τα δεδομένα από την Βάση μόλις ανοίγει η εφαρμογή. Αυτό προκάλεσε δυσφορία στους χρήστες αφού δεν ήξεραν αν έπρεπε να περιμένουν κάτι από την εφαρμογή να ανοίξει ή αν έχει πάθει κάτι ξαφνικά. Συνολικά, ο χρόνος αναμονής είναι 30 δευτερόλεπτα. Από την αρχή θα έπρεπε να υλοποιηθούν επερωτήσεις στην γλώσσα προγραμματισμού PHP που θα παίρνουν μόνο τα δεδομένα που είναι απαραίτητα για εκείνη την οθόνη και όχι όλα τα δεδομένα μαζί και έτσι η εφαρμογή θα άνοιγε αμέσως και θα ήταν αρκετά πιο γρήγορη.

Αν εξαιρεθεί το πιο πάνω πρόβλημα, η εφαρμογή έχει υλοποιηθεί με αρκετά καλές τεχνολογίες. Αρχικά, με την ραγδαία ανάπτυξη των έξυπνων κινητών συσκευών επιλέχθηκε το λειτουργικό σύστημα Android για την υλοποίηση της εφαρμογής γιατί είναι το πιο διαδεδομένο μέχρι σήμερα, είναι ανοικτού κώδικα λογισμικό και προσφέρει τις περισσότερες δυνατότητες από όλα. Οι κινητές συσκευές με λειτουργικό σύστημα Android παρέχουν πολλές δυνατότητες όπως σύνδεση στο διαδίκτυο, μνήμη, μεγάλη υπολογιστική ισχύ, αισθητήρες, GPS κτλ.

Για να αναπτυχθεί η εφαρμογή χρησιμοποιήθηκαν εργαλεία όπως Eclipse IDE με το Android ADT για την υλοποίηση της εφαρμογής, Microsoft SQL Server 2008 για την υλοποίηση της Βάσης Δεδομένων και Microsoft Visual Studio 2010 για την ανάπτυξη της διαδικτυακής εφαρμογής. Όλα τα εργαλεία ήταν αρκετά εύχρηστα και βοηθητικά και είχαν οδηγίες χρήσης που σε κατεύθυναν εύκολα σε περίπτωση που έκανες κάτι λάθος.

Εν κατακλείδι, όταν η εφαρμογή δόθηκε σε πιθανούς χρήστες είχε αρκετά θετικά αποτελέσματα και καλές εντυπώσεις και μερικοί χρήστες μου έδωσαν κάποιες μελλοντικές ιδέες που θα μπορούσαν να βελτιώσουν ακόμη περισσότερο την εφαρμογή. Οι ιδέες αυτές καταγράφονται στο επόμενο υποκεφάλαιο.

6.2 Μελλοντικές Προεκτάσεις

Με την ολοκλήρωση της διπλωματικής μου εργασίας και επακόλουθα της εφαρμογής «Cyprus Bus Routes» προέκυψαν κάποιες πιθανές βελτιώσεις και ιδέες που θα μπορούσαν να υλοποιηθούν ώστε να βελτιωθεί περισσότερο η εφαρμογή από άποψη ευχρηστίας και χρησιμότητας.

Πρόσθεση περισσότερων γλωσσών

Με την προσθήκη περισσότερων γλωσσών η εφαρμογή θα μπορούσε να έχει πιο πολλούς χρήστες. Αν και η αγγλική γλώσσα είναι η πιο διαδεδομένη γλώσσα, υπάρχουν άνθρωποι που δεν την γνωρίζουν καθόλου και άλλοι που την γνωρίζουν πολύ λίγο. Έτσι γι' αυτούς τους ανθρώπους θα ήταν δύσκολο να χρησιμοποιήσουν την εφαρμογή.

Πρόσθεση κόστους εισιτηρίου

Με την προσθήκη του κόστους εισιτηρίου παρέχεται στον χρήστη η δυνατότητα να γνωρίζει πόσο κοστίζει ένα εισιτήριο λεωφορείου για την διαδρομή που θα διανύσει. Επίσης, θα ήταν καλό να μπορούσε να προκρατήσει ένα εισιτήριο για να είναι σίγουρος ότι θα βρει θέση στο συγκεκριμένο λεωφορείο που θα κάνει την διαδρομή που επιθυμεί.

Πρόσθεση Πόλεων

Οι δημόσιες συγκοινωνίες στην Κύπρο αυξάνονται κάθε χρόνο και ο στόχος του υπουργείου συγκοινωνιών είναι να αυξήσει τη χρήση του λεωφορείου από 2% που ήταν το 2010, σε ποσοστό μεγαλύτερο από 10% μέχρι το 2019. Συνεπώς, η πρόσθεση περισσότερων πόλεων στην εφαρμογή θα εξυπηρετούσε ένα πιο μεγάλο κύκλο ανθρώπων.

Διαθεσιμότητα εφαρμογής και σε άλλα λειτουργικά συστήματα

Μπορεί γενικά οι περισσότερες κινητές συσκευές να έχουν λειτουργικό σύστημα Android αλλά υπάρχει και ένα μεγάλο ποσοστό ανθρώπων που έχουν συσκευές που υποστηρίζουν λειτουργικό σύστημα iOS και Windows phone. Υλοποιώντας την εφαρμογή και σε αυτά τα λειτουργικά συστήματα θα εξυπηρετούνταν σχεδόν όλοι οι άνθρωποι.

Απόκριση σε συχνές ερωτήσεις των χρηστών

Υπάρχουν συχνές ερωτήσεις από τις επιβάτες των λεωφορείων σχετικά με τις διαδρομές και τα ωράρια τις και θα ήταν χρήσιμο η εφαρμογή να μπορεί να παρέχει ανά πάσα στιγμή

τις κατάλληλες απαντήσεις σε τις. Τις από τις ερωτήσεις αυτές είναι «Πού μπορώ να αγοράσω εισιτήρια?», «Επιτρέπεται η μεταφορά ζώων με το λεωφορείο» κτλ.

Εμφάνιση πιο λεπτομερούς χρονοδιαγράμματος

Η εφαρμογή θα μπορούσε να γίνει πιο βοηθητική για τον χρήστη, παρουσιάζοντας του ένα πιο λεπτομερές χρονοδιάγραμμα, που να δείχνει ακριβώς πότε πρέπει να κατέβει από ένα λεωφορείο, ποιο λεωφορείο πρέπει να πιάσει μετά και τι ώρα και πόσην ώρα θα χρειαστεί το λεωφορείο αυτό για να τον πάρει στον προορισμό του.

Γρηγορότερη απόκριση εφαρμογής

Με την χρήση της γλώσσας προγραμματισμού PHP η εφαρμογή θα μπορούσε να λαμβάνει τα δεδομένα από την Βάση Δεδομένων σε μερικά δέκατα του δευτερολέπτου και έτσι αυτή η καθυστέρηση που συμβαίνει τώρα κατά το άνοιγμα της δεν θα φαινόταν καθόλου στον χρήστη.

Βιβλιογραφία

- [1] Android website <http://www.android.com/>
- [2] Wikipedia-Android operating system
[http://en.wikipedia.org/wiki/Android_\(operating_system\)#cite_note-AndroidAnnouncement-11](http://en.wikipedia.org/wiki/Android_(operating_system)#cite_note-AndroidAnnouncement-11)
- [3] Wikipedia-XML <http://el.wikipedia.org/wiki/XML>
- [4] Android versions
<http://gadgets.ndtv.com/mobiles/news/android-jelly-bean-is-now-the-leading-android-version-with-a-distribution-share-of-nearly-38-percent-389696>
- [5] Android developer <http://developer.android.com/index.html>
- [6] Google Maps API v2: <https://developers.google.com/maps/documentation/android/>
- [7] Δρομολόγια Λεωφορείων: <http://www.osel.com.cy/>
- [8] Στάσεις Λεωφορείων: <http://www.cyprusbybus.com/>
- [9] Εικόνες για τους δείκτες: <http://mapicons.nicolasmollet.com/category/markers/>
- [10] Tutorials για το Google map v2: <http://wptrafficanalyzer.in/blog/gps-and-google-map-in-android-applications-series/>
- [11] ListView tutorial: <http://javatechig.com/android/android-listview-tutorial/>
- [12] Διάστημα μεταξύ δύο συντεταγμένων:
<http://www.movable-type.co.uk/scripts/latlong.html>
- [13] Places API: <https://developers.google.com/places/training/>
- [14] Wikipedia: http://en.wikipedia.org/wiki/Main_Page