



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



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Mobile εφαρμογή για εύρεση θέσεως parking
στην Θεσσαλονίκη**



Της φοιτήτριας

Κατσίκη Μάρθας

Αρ. Μητρώου: 093435

Επιβλέπων καθηγητής

Ηλιούδης Χρήστος

Θεσσαλονίκη 2014

ΠΡΟΛΟΓΟΣ

Οι εξελίξεις στην τεχνολογία οδήγησαν στην βελτίωση των κινητών συσκευών, διευρύνοντας τις υπηρεσίες που παρέχουν, έτσι ώστε να εξυπηρετούν περισσότερο τον άνθρωπο. Η χρήση τους δεν περιορίζεται μόνο στην επικοινωνία, αλλά παρέχουν και εφαρμογές που αποσκοπούν στη διευκόλυνση της καθημερινότητας καλύπτοντας διάφορες ανάγκες.

Αναφερόμαστε βέβαια στα αποκαλούμενα smartphones τα οποία μπορεί να τρέχουν σε διαφορετικό λειτουργικό σύστημα με ποιο γνωστά το Android, iOS, Windows Phone κ.ά. Οι προγραμματιστές έχουν την δυνατότητα να υλοποιήσουν υπηρεσίες σύμφωνα με τις ανάγκες των χρηστών.

Σκοπός της παρούσας πτυχιακής εργασίας είναι η δημιουργία μιας υπηρεσίας στο λειτουργικό σύστημα του Android που θα εμφανίζει τα ιδιωτικά parking της Θεσσαλονίκης. Ο χρήστης μπορεί να επιλέξει μια συγκεκριμένη περιοχή για να ψάξει για parking ή να βρει τα πιο κοντινά στην τοποθεσία που βρίσκεται.

ΠΕΡΙΛΗΨΗ

Η πτυχιακή εργασία παρουσιάζει την υλοποίηση μιας Android εφαρμογής που δείχνει στο χάρτη της Θεσσαλονίκης τα ιδιωτικά parking ανάλογα με την αναζήτηση του χρήστη.

Αρχικά αναφέρονται ο ορισμός της καινοτομίας, τα χαρακτηριστικά και τα κριτήρια μιας διαδικτυακής υπηρεσίας.

Κατόπιν αναλύονται οι απαιτήσεις της υπηρεσίας και περιγράφεται αναλυτικά η υλοποίηση και η σχεδίαση. Επιπλέον αναφέρονται οι τεχνολογίες που χρησιμοποιήθηκαν καθώς και το τεχνολογικό περιβάλλον όπου αναπτύχθηκε η εφαρμογή. Στο τέλος αναφέρονται προτάσεις για μερικές επεκτάσεις της εφαρμογής.

ABSTRACT

This Thesis presents the implementation of an Android application that shows in the map all the parking lots/garages of Thessaloniki depending on the user's research.

In the beginning, the definition of innovation, the characteristics and criteria of a web service are mentioned.

Moreover the service's requirements and details of implementation and design are being analyzed. In addition the technologies and the technological environment, that were used in order to develop this application, are also mentioned. In the last chapter there are a few suggestions which are discussed and proposed for the extension of the web service.

ΕΥΧΑΡΙΣΤΙΕΣ

Ένα μεγάλο ευχαριστώ στον επιβλέποντα καθηγητή μου κύριο Ηλιούδη Χρήστο για την καθοδήγηση και την πολύτιμη βοήθεια που μου πρόσφερε κατά την διάρκεια της πτυχιακής, ώστε να επιτύχουμε το καλύτερο δυνατό αποτέλεσμα. Στη συνέχεια θα ήθελα να ευχαριστήσω τους γονείς μου για την στήριξη και την βοήθειά τους, διότι δεν θα κατάφερνα τίποτα χωρίς αυτούς. Τέλος, θα ήθελα να ευχαριστήσω τους φίλους και συναδέλφους Σέργιο Βενετικίδη, Σωτήρη Χατζηαναγνώστου και Μένανδρο Αποστολίδη και όλους όσους με στήριξαν στη συγγραφή της συγκεκριμένης πτυχιακής εργασίας.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ.....	2
ΠΕΡΙΛΗΨΗ.....	3
ABSTRACT.....	4
ΕΥΧΑΡΙΣΤΙΕΣ.....	5
Ευρετήριο σχημάτων	8
ΕΙΣΑΓΩΓΗ.....	10
1Σκοπός της εργασίας	10
2Δομή της εργασίας	10
ΚΕΦΑΛΑΙΟ 1: Καινοτομία Διαδικτυακών υπηρεσιών	13
1.1 Τι είναι καινοτομία;	13
1.2 Χαρακτηριστικά και κριτήρια αξιολόγησης.....	15
1.2.1 Βραχυπρόθεσμες δραστηριότητες.....	15
1.2.2 Ουσιώδη Χαρακτηριστικά Mobile Εφαρμογών	16
1.2.3 Κριτήρια αξιολόγησης εφαρμογών.....	19
1.3 Πρότυπα Αξιοπιστίας των Εφαρμογών	22
1.4 Χαρακτηριστικά των Διαδικτυακών Υπηρεσιών Εξυπηρέτησης Πολιτών.....	25
1.4.1 Διαδικτυακές Υπηρεσίες – Ορισμός.....	25
1.4.2 Πλεονεκτήματα και Μειονεκτήματα των Διαδικτυακών Υπηρεσιών	26
1.4.3 Αρχιτεκτονική Διαδικτυακών Υπηρεσιών	27
1.4.4 Χρήση των Διαδικτυακών Υπηρεσιών στο M-Commerce	29
1.4.5 Mobile Commerce	30
1.4.6 Διαθέσιμα προϊόντα και υπηρεσίες	31
ΚΕΦΑΛΑΙΟ 2: Υφιστάμενες υπηρεσίες: συγκριτική αξιολόγηση	35
2.1 Παροχή παραπλήσιων υπηρεσιών	35

2.2 Αξιολόγηση των εφαρμογών με βάση τα χαρακτηριστικά τους	40
2.3 Αξιολόγηση των εφαρμογών με βάση τα παρακάτω κριτήρια	41
2.4 Επίλογος	41
ΚΕΦΑΛΑΙΟ 3: Ανάλυση Απαιτήσεων	42
3.1 Στόχοι και επιδιώξεις του έργου	42
3.2 Απαιτήσεις ηλεκτρονικής πλατφόρμας (συστήματος).....	43
3.3 Περιορισμοί και ειδικά χαρακτηριστικά του συστήματος	44
3.3.1 Βάση Δεδομένων	45
3.3.2 Διάθεση πλατφόρμας σε διαφορετικές γλώσσες.....	45
3.3.3 Εύρεση τοποθεσίας	45
3.4 Απαιτήσεις χρηστών	46
3.4.1 Ορισμός απαιτήσεων.....	46
3.4.2 Βάση δεδομένων	47
3.5 Ειδικές απαιτήσεις.....	48
3.5.1 Διεπαφές χρηστών εφαρμογής.....	48
3.5.2 Διεπαφές υλικού	48
3.5.3 Διεπαφές λογισμικού	49
3.5.4 Διεπαφές επικοινωνιών	49
3.6 Περιπτώσεις Χρήσης.....	49
3.7 Επίλογος	53
ΚΕΦΑΛΑΙΟ 4: Τεχνολογικό Περιβάλλον Ανάπτυξης.....	54
4.1 Περιβάλλον Eclipse	54
4.1.1 Eclipse	54
4.1.2 Eclipse ADT (Android Development Tools)	55
4.2 Πλατφόρμα Android	56
4.2.1 Λειτουργικό σύστημα Android.....	56
4.2.2 Γιατί επιλέγουμε Android;	57

4.2.3	Ιστορική Αναδρομή	59
4.3	SQLite	65
4.3.1	Τι είναι το SQLite;	65
4.3.2	Το SQLite στο Android	66
4.3.3	Παράδειγμα	66
4.4	Τεχνολογία JSON.....	69
4.4.1	Τι είναι το JSON;.....	69
4.5	Google Play Services Library	70
4.5.1	Εγκατάσταση των Google Play Services στο Eclipse με ADT 71	
4.6	Επίλογος	74
ΚΕΦΑΛΑΙΟ 5:	Η εφαρμογή “Thessaloniki Parking”	75
5.1	Αρχική Διάκριση κλάσεων	75
5.2	Σχεδίαση Εφαρμογής.....	76
5.3	Υλοποίηση - Λειτουργικότητα προγράμματος	88
5.4	Επίλογος	109
ΚΕΦΑΛΑΙΟ 6:	Χαρακτηριστικά Καινοτόμων και Προστιθέμενων που έχει η Υπηρεσία	110
ΚΕΦΑΛΑΙΟ 7:	Μελλοντική Επέκταση – Συμπεράσματα	113
ΒΙΒΛΙΟΓΡΑΦΙΑ ΚΑΙ ΑΝΑΦΟΡΕΣ	115

Ευρετήριο σχημάτων

Εικόνα 1:	Η γενική διαδικασία δημιουργίας της διαδικτυακής υπηρεσίας.....	29
Εικόνα 2:	Easy Taxi App	35
Εικόνα 3:	Taxibeat App.....	37
Εικόνα 4:	Parkmobile App.....	38
Εικόνα 5:	BestParking App	39
Εικόνα 6:	Διάγραμμα περιπτώσεων χρήσης.....	47

Εικόνα 7: Διάγραμμα Προόδου Επιχειρήσεων – Business Program Diagram	53
Εικόνα 8: Λογότυπο του Eclipse	54
Εικόνα 9: Λογότυπο Android.....	56
Εικόνα 10: Η εξέλιξη του Android.....	64
Εικόνα 11: Όλες οι εκδόσεις του Android μέχρι σήμερα.....	64
Εικόνα 12: Λογότυπο SQLite	65
Εικόνα 13: Λόγοτυπου του JSON	69
Εικόνα 14: Google Play Services logo	70
Εικόνα 15: Εγκατάσταση του Android SKD Manager.....	72
Εικόνα 16: Εισαγωγή της βιβλιοθήκης στο workspace του Eclipse.....	73
Εικόνα 17: Προσθήκη της βιβλιοθήκης στην εφαρμογή	74
Εικόνα 18: Οι κλάσεις της εφαρμογής και πώς συνδέονται μεταξύ τους	76
Εικόνα 19: Splash Screen.....	77
Εικόνα 20: MainActivity	79
Εικόνα 21: Επιλογή ελληνικής γλώσσας για την MainActivity	80
Εικόνα 22: Σχετικά με την εφαρμογή.....	81
Εικόνα 23: Το layout της κλάσης BasicMapActivity.....	83
Εικόνα 24: alertDialog του parking lot που περιέχει προσφορά και θέσεις ...	84
Εικόνα 25: alertDialog του parking lot χωρίς προσφορά και θέσεις	85
Εικόνα 26: Παράδειγμα του layout της InfoDetails	86
Εικόνα 27: Παράδειγμα του layout της InfoDetails	87
Εικόνα 28: Το layout της κλάσης DirectionsMap.....	88
Εικόνα 29: Διαδρομή χρήστη - Πράσινο η αφετηρία, Κόκκινο για τον προορισμό	111
Εικόνα 30: Τα markers που έχουν χρώμα μπλε, δείχνουν τα parking με προσφορές	112

Ευρετήριο πινάκων

Πίνακας 1: Αξιολόγηση των εφαρμογών με βάση τα χαρακτηριστικά τους	40
Πίνακας 2: Αξιολόγηση εφαρμογών βάση κριτηρίων	41

ΕΙΣΑΓΩΓΗ

1 Σκοπός της εργασίας

Σκοπός της εργασίας είναι η ανάπτυξη μιας υπηρεσίας/εφαρμογής μέσω της οποίας οι χρήστες να μπορούν με τη χρήση του κινητού τους τηλεφώνου να βρίσκουν θέση parking ανεξάρτητα από το που βρίσκονται την συγκεκριμένη στιγμή – ο χρήστης θα μπορεί είτε να δηλώσει σε ποια περιοχή ψάχνει θέση parking, πριν ακόμα φθάσει, είτε μέσω GPS να βρίσκει η διαδικτυακή υπηρεσία τα ιδιωτικά parking για το μέρος που είναι ο χρήστης εκείνη τη χρονική στιγμή. Μέσω αυτής οι χρήστες θα πληροφορούνται για το ποια ιδιωτικά parking βρίσκονται κοντά τους, αν έχουν άδειες θέσεις, τις ώρες και μέρες λειτουργίας καθώς και την τιμή τους αλλά και τις προσφορές τους (π.χ. ειδικές προσφορές για ανέργους, φοιτητές, κ.ά.).

Η εφαρμογή μπορεί να συμβάλλει στην ανάπτυξη των μικρών αυτών επιχειρήσεων, παρέχοντας μια νέα ψηφιακή δυνατότητα και μια υπηρεσία προστιθέμενης αξίας μέσω smartphones. Η αξιοποίηση καινοτόμων εφαρμογών επί συσκευών κινητών επικοινωνιών μπορεί να ενισχύσει το οικονομικό κέρδος των επιχειρήσεων, στην συγκεκριμένη περίπτωση τα ιδιωτικά parking, με ευκαιρίες εξωστρεφούς ανάπτυξης και βελτίωσης των υπηρεσιών τους. Στις ενισχυόμενες επιχειρήσεις μπορεί να εξασφαλίσει την άμεση πρόσβαση σε εταιρικά δεδομένα (τιμή, προσφορές) καθώς και την αξιοποίηση τους από οπουδήποτε (mobility), επίσης να παρέχει συνθήκες ολοκληρωμένης προσέγγισης νέων πελατών, χρησιμοποιώντας τις δυνατότητες των «έξυπνων» συσκευών κινητών επικοινωνιών για την παροχή προσωποποιημένης –στοχευόμενης πληροφόρησης.

2 Δομή της εργασίας

Κάθε κεφάλαιο έχει διαφορετικό στόχο, το καθένα αποσκοπεί στο να εισαχθούν οι βασικές έννοιες που εξηγούν της υπηρεσίας που υλοποιείται ή να περιγραφτεί η λειτουργία της διαδικτυακής υπηρεσίας καθώς και τις

τεχνολογίες και το περιβάλλον που χρησιμοποιεί ή να παρουσιάζει μελλοντικές επεκτάσεις. ”

Συγκεκριμένα το δεύτερο κεφάλαιο εξηγεί τι σημαίνει "Καινοτομία", ποια είναι τα βασικά χαρακτηριστικά μιας εφαρμογής, με ποια κριτήρια αξιολογείται, ποια είναι τα πρότυπα αξιοπιστίας καθώς και τα χαρακτηριστικά των διαδικτυακών υπηρεσιών που εξυπηρετούν τους πολίτες.

Στο τρίτο κεφάλαιο αναλύονται υφιστάμενες εφαρμογές που παρέχουν παρόμοιες υπηρεσίες και αξιολογούνται με βάση τα κριτήρια που περιγράφηκαν στο προηγούμενο κεφάλαιο.

Στο τέταρτο κεφάλαιο παρουσιάζεται η ανάλυση των απαιτήσεων για την πλατφόρμα, τους χρήστες και αναλύονται οι στόχοι και οι επιδιώξεις του έργου. Επίσης εξηγούνται οι περιορισμοί που υπάρχουν, τα ειδικά χαρακτηριστικά του συστήματος και οι ειδικές απαιτήσεις.

Το πέμπτο κεφάλαιο αναφέρεται στο τεχνολογικό περιβάλλον ανάπτυξης της διαδικτυακής υπηρεσίας. Περιγράφουμε το περιβάλλον του Eclipse, την πλατφόρμα του Android καθώς και τις τεχνολογίες που χρησιμοποιούνται: JSON, SQLite και Google Play Services Library.

Το έκτο κεφάλαιο περιέχει την υλοποίηση της υπηρεσίας. Αρχικά αναφέρονται οι κλάσεις της εφαρμογής, έπειτα η σχεδίασή της και τέλος το λειτουργικότητα της. Επίσης θα εξηγηθεί η χρήση του JSON για τα δεδομένα που θα λαμβάνουμε μέσω ενός server, η χρήση του SQLite για τα δεδομένα που παίρνουμε από την βάση δεδομένων και πώς χρησιμοποιούμε μια βάση δεδομένων στο Android καθώς επίσης και την χρήση της βιβλιοθήκης Google Play Services, διότι καμία υπηρεσία της Google δεν μπορεί να χρησιμοποιηθεί αν μια εφαρμογή δεν περιέχει την βιβλιοθήκη αυτή.

Στο έβδομο κεφάλαιο δίνεται μια σύντομη αναφορά για τα χαρακτηριστικά καινοτόμων που περιέχει η εφαρμογή με στόχο να προσελκύσει περισσότερους χρήστες.

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κασίκη

Το τελευταίο κεφάλαιο θα αποτελέσουν οι μελλοντικές επεκτάσεις της διαδικτυακής υπηρεσίας σε συνδυασμό με τα συμπεράσματα που προκύπτουν από αυτήν.

ΚΕΦΑΛΑΙΟ 1: Καινοτομία Διαδικτυακών υπηρεσιών

Σε αυτό το κεφάλαιο εξηγείται ο ορισμός της καινοτομίας, τα χαρακτηριστικά των διαδικτυακών υπηρεσιών καθώς και τα κριτήρια αξιολόγησης. Επίσης περιγράφονται τα πρότυπα αξιοπιστίας και αναλύεται ο ορισμός των διαδικτυακών υπηρεσιών, αναφέρονται τα πλεονεκτήματα και τα μειονεκτήματά τους καθώς και η χρήση τους στην αγορά των κινητών συσκευών.

1.1 Τι είναι καινοτομία;

“Με τον όρο *καινοτομία* εννοούμε την υλοποίηση μια νέας και πρωτοποριακής ιδέας σε εμπορεύσιμο προϊόν/υπηρεσία ή ακόμα και σε νέα μέθοδο παροχής κοινωνικής υπηρεσίας.”

Με αυτόν τον τρόπο ο όρος αναφέρεται στην διαδικασία. Από την άλλη μεριά, όταν με την λέξη «καινοτομία» υποδηλώνεται ένα νέο και βελτιωμένο προϊόν ή μια υπηρεσία που διαχέεται επιτυχώς στην αγορά, η έμφαση δίνεται στο αποτέλεσμα της διαδικασίας.

Συνδέεται με την έρευνα και την ανάπτυξη, ειδικά στο χώρο των επιχειρήσεων, όπου υπάρχουν και τα αντίστοιχα τμήματα (Research and Development). Στην Ευρωπαϊκή Ένωση, η καινοτομία επιτυγχάνεται μέσω ευρωπαϊκών προγραμμάτων συνεργασίας μεταξύ διακρατικών εταιρών.

Είναι σημαντικό να γίνει διάκριση μεταξύ της τεχνολογικής καινοτομίας προϊόντων και διαδικασιών και της μη τεχνολογικής καινοτομίας (οργάνωσης και εμπορίας). Η δημιουργία μιας απλής ιστοσελίδας με πληροφορίες, χωρίς νέες και πρωτότυπες διαδικτυακές υπηρεσίες δεν αποτελεί καινοτομία.

Μια καινοτομία θεωρείται τεχνολογική μόνο στην περίπτωση που βασίζεται σε νέες τεχνολογικές εφαρμογές, νέων συνδυασμών υπάρχουσών τεχνολογιών ή στη χρησιμοποίηση άλλου είδους γνώσεων που αποκτήθηκαν από την επιχείρηση. Οι μεταβολές φυσικής φύσεως δεν περιλαμβάνονται. Μη

τεχνολογική καινοτομία θεωρείται η εφαρμογή νέων ή βελτιωμένων σχεδίων ή μεθόδων πώλησης που αποσκοπούν στην αύξηση της ελκυστικότητας των αγαθών και των υπηρεσιών ή στην είσοδο σε νέες αγορές.

Όσον αφορά τις τεχνολογικές καινοτομίες, ο τομέας των υπηρεσιών προσφέρει σημαντικές ευκαιρίες (καινοτομίας), οι οποίες μπορούν να αξιοποιηθούν. Η καινοτομία στις υπηρεσίες μπορεί να αφορά νέες τεχνολογίες, διεργασίες, τρόπους αλληλεπίδρασης με τον πελάτη, αλλά και νέα επιχειρηματικά μοντέλα και πρακτικές. [Wikipedia, 2013, Καινοτομία]

Η ανάπτυξη των τεχνολογιών πληροφορικής τα τελευταία χρόνια επιτρέπει την αυτοματοποίηση ορισμένων υπηρεσιών και την ευρεία διάθεσή τους μέσω του διαδικτύου. Με αυτόν τον τρόπο, δίνεται η δυνατότητα ανάπτυξης νέων καινοτόμων υπηρεσιών προστιθέμενης αξίας, οι οποίες αναδεικνύονται ως μια ευκαιρία-κλειδί για ανάπτυξη της αγοράς, της επιχειρηματικότητας και της ανταγωνιστικότητας στην Ευρώπη. Στην Ελλάδα τα τελευταία χρόνια οι κλάδοι υψηλής τεχνολογίας είχαν να παρουσιάσουν μια ταχεία ανάπτυξη. Πάνω από 26.000 επιχειρήσεις δραστηριοποιούνται σε κλάδους υψηλής τεχνολογίας με την πλειονότητα αυτών να έχει ιδρυθεί τα τελευταία δεκαπέντε χρόνια, ενώ και η σχετική απασχόληση παρουσίαζε σημαντική αύξηση (80.000 εργαζόμενοι την περίοδο 2001 – 2006). Οι κλάδοι αυτοί μπορούν να αποτελέσουν σημαντικές πηγές ανάπτυξης συνολικά για την ελληνική οικονομία. Αποτελούν τους πρωτοπόρους κλάδους σε όρους καινοτομίας και επιπλέον συνεισφέρουν στην καινοτομική επίδοση των υπόλοιπων κλάδων της οικονομίας

Οι τεχνολογίες πληροφορικής και επικοινωνιών μπορούν να αποτελέσουν διέξοδο ανάπτυξης των επιχειρήσεων, καθώς οι τελευταίες εξελίξεις σε τεχνολογίες και υπηρεσίες λογισμικού, οι βελτιώσεις στο υλικό και η ανάπτυξη της ευρυζωνικότητας είναι δυνατό να οδηγήσουν στην ανάπτυξη και προσφορά καινοτόμων υπηρεσιών από τις επιχειρήσεις του κλάδου. Έχοντας σαν αρχή αυτές τις καινοτόμες υπηρεσίες, οι επιχειρήσεις θα προωθήσουν την αναπτυξιακή στρατηγική τους, τη βελτίωση των οικονομικών τους αποτελεσμάτων, την αύξηση των παραγωγικών επενδύσεών τους, τη

διατήρηση αλλά και τη δημιουργία νέων θέσεων εργασίας. [PLAN A.E., 2012, Ανάπτυξη Καινοτόμων Υπηρεσιών Προστιθέμενης Αξίας]

1.2 Χαρακτηριστικά και κριτήρια αξιολόγησης

1.2.1 Βραχυπρόθεσμες δραστηριότητες

Οι άνθρωποι τείνουν να χρησιμοποιούν τους επιτραπέζιους υπολογιστές για μεγάλο χρονικό διάστημα. Για τον λόγο αυτό, ο χρόνος έναρξης (startup time) δεν είναι τόσο σημαντικός όσο η παροχή, στους χρήστες, των κατάλληλων χαρακτηριστικών που μπορεί να χρειαστούν ενώ εξερευνούν και δουλεύουν με τις πληροφορίες τους.

Οι φορητοί υπολογιστές τείνουν να μοιράζονται πολλά από τα ίδια μοτίβα χρήσης, όπως οι επιτραπέζιοι υπολογιστές, αλλά είναι πιθανό να δούμε ορισμένα αναδυόμενα χαρακτηριστικά μιας κινητής συσκευής. Ωστόσο, τα πρότυπα χρήσης των φορητών υπολογιστών εξακολουθούν να τείνουν να μοιάζουν πιο πολύ στους επιτραπέζιους υπολογιστές (desktop oriented) από ό,τι σε μια κινητή συσκευή (mobile device oriented) και μέρος αυτού οφείλεται απλώς στο σχετικό φυσικό μέγεθος των συσκευών.

Οι περισσότεροι άνθρωποι κάνουν ένα από τρία πράγματα: την περιήγηση στο Web, τη δουλειά σε έγγραφα (επεξεργασία κειμένου, λογιστικά φύλλα, εικόνες, φωτογραφίες κ.ά.), ή την επικοινωνία (email, instant messaging κ.ά.).

Αυτά τείνουν να είναι μακροπρόθεσμες και διερευνητικές δραστηριότητες και ο χρήστης θα αλλάζει σε τακτά χρονικά διαστήματα μεταξύ αυτών των δραστηριοτήτων. Ένας προγραμματιστής, μπορούμε να προσθέσουμε την ανάπτυξη λογισμικού σε αυτή τη λίστα, το οποίο είναι απολύτως μια δραστηριότητα μεγάλης διάρκειας. Ένας παίκτης ηλεκτρονικών παιχνιδιών, θα ξέρει ότι πολλά από τα πιο ενδιαφέροντα παιχνίδια για επιτραπέζιους υπολογιστές είναι μακροπρόθεσμα και διερευνητικές παιχνίδια που επιτρέπει στους χρήστες να διερευνούν και να αλληλεπιδρούν με σύνθετους εικονικούς κόσμους.

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

Οι άνθρωποι που χρησιμοποιούν και τα δύο, desktop εφαρμογές και εφαρμογές κινητών συσκευών, έχουν την τάση να χρησιμοποιούν αυτά τα δύο σε διαφορετικούς αλλά συμπληρωματικούς τρόπους. Ενώ εργάζεστε με μια κινητή συσκευή, οι δραστηριότητες του χρήστη τείνουν να είναι σύντομη διάρκεια σύνδεσης, interrupt-driven ή interrupt-causing αλληλεπιδράσεις.

Οι mobile εφαρμογές, επίσης, τείνουν να επικεντρώνονται περισσότερο στο να ενεργοποιήσουν μερικά συγκεκριμένα χαρακτηριστικά σε αντίθεση με την προσφορά ενός διερευνητικού περιβάλλοντος γενικού σκοπού σαν αυτό των επιτυχών εφαρμογών desktop. Επειδή οι κινητές συσκευές συχνά λειτουργούν χρησιμοποιώντας μόνο το ένα χέρι ή πατώντας μια μικρή οθόνη με μια γραφίδα, είναι σημαντικό ότι οι χρήστες της συσκευής είναι σε θέση να ανακαλύψουν και να περιηγηθούν γρήγορα στις πληροφορίες και τα χαρακτηριστικά που θέλουν. Η δυνατότητα να περιηγηθούν γρήγορα σε ένα μικρό σύνολο των βασικών χαρακτηριστικών είναι μια σημαντική πτυχή μιας μεγάλης εμπειρίας κινητών συσκευών. [2004, Characteristics of Mobile Applications]

1.2.2 Ουσιώδη Χαρακτηριστικά Mobile Εφαρμογών

Κοιτάζοντας από κοντά τις επιτυχείς mobile εφαρμογές, παρατηρούμε κάποια κύρια χαρακτηριστικά ίσως τα κοινά ουσιώδη χαρακτηριστικά όλων των εφαρμογών. Οι ανάπτυξη εφαρμογών κινητής τηλεφωνίας είναι βεβαίως ένα μεγάλο πεδίο και ακόμη μια πάρα πολύ αναπτυσσόμενη βιομηχανία παγκόσμιας κλίμακας, που έχει δημιουργήσει νέα ευκαιρίες σύγχρονων επιχειρήσεων και εισήγαγε νέες τεχνολογίες. Τα πιο κοινά χαρακτηριστικά βασίζονται σε μεγάλο βαθμό σε υπηρεσία επιπέδου σχεδίασης, οπότε αν χρησιμοποιηθούν σοφά τότε οι εφαρμογές θα είναι πολύ χρήσιμες στην κάλυψη των απαιτήσεων των χρηστών.

Παραθέτουμε τα βασικά χαρακτηριστικά των εφαρμογών:

Συνδεσιμότητα:

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

Η συσκευή είναι συνεχώς συνδεδεμένη στο δίκτυο κινητής τηλεφωνίας καθώς οι εφαρμογές είναι συνεχώς online. Με αυτόν τον τρόπο συγκεκριμένες πληροφορίες του χρήστη και κοινοποιήσεις καταχωρούνται στις εφαρμογές που είναι προσβάσιμες. Αυτό είναι ένα σημαντικό χαρακτηριστικό μια κινητής εφαρμογής· να είναι διαθέσιμη από οπουδήποτε. Αλλιώς η ενεργή κλήση μιας υπηρεσίας για να «αντιδράσει» είναι άχρηστη σε ένα περιβάλλον κινητής τηλεφωνίας. Η ικανότητα της ώθησης γίνεται απαραίτητη στην ανάπτυξη εφαρμογών σε κάθε Smartphone σαν αυτό το χαρακτηριστικό να κρατά την εφαρμογή στο μυαλό του χρήστη.

Εντοπισμός:

Ο εντοπισμός πληροφοριών και η ευκαιρία να γνωστοποιήσουμε τη θέση με βάση πληροφορίες είναι ένα κρίσιμο στοιχείο αλλά και εντυπωσιακό και πρακτικό. Ενσωματώνει την εφαρμογή ανάλογα με το πλαίσιο χρήσης. Το χαρακτηριστικό μπορεί να μην είναι τέλειο για κάθε εφαρμογή αλλά είναι μια καλή σκέψη να συσχετίσουμε την τοποθεσία με ένα σημείωμα ή μια φωτογραφία και τον περιορισμό των πιθανών επιλογών και/ή ταξινομημένες τοποθεσίες. Το χαρακτηριστικό αυτό προσθέτει όμορφα μια καλή εμπειρία για τον χρήστη.

Ευκολία:

Μια απλή (one-handed) και συναισθηματική σχεδίαση εγγυάται μια υψηλή αξία και αποδοχή από τους χρήστες. Μια καλή εφαρμογή μπορεί να κάνει τη δουλειά της σε διαφορετικά περιβάλλοντα και γρήγορα μεταβαλλόμενες καταστάσεις (αλλαγές στο φως και το θόρυβο του περιβάλλοντος, ασταθή κίνηση της συσκευής, κ.λπ.). Η συνολική χρησιμότητα και η δομή πληροφοριών πρέπει να σχεδιαστούν προσεκτικά για να οικοδομήσουν μια χαρούμενη και ταιριαστή ροή αλληλεπίδρασης. Σίγουρα ένα καλό περιεχόμενο είναι ένα επιπλέον θετικό στοιχείο. Η ανάλυση των απαιτήσεων των χρηστών και η δημιουργία χρήσιμων ιδεών από αυτό είναι επίσης απαραίτητη.

Προσέγγιση:

Η ικανότητα προσέγγισης καλύπτει πρόσθετο χαρακτηριστικό που παρέχεται από τη φύση των mobile εφαρμογών. Μια εξαιρετική εφαρμογή μπορεί να

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

χρησιμοποιηθεί οποιαδήποτε στιγμή, σε οποιοδήποτε μέρος η ικανότητα κάλυψης γίνεται διαθέσιμα πλήρως. Αυτό το χαρακτηριστικό είναι υπό την έννοια ενημερωμένων πληροφοριών και χρησιμότητας. Για παράδειγμα, οι επιλογές αναγνώρισης χρόνου και σχετικού περιεχομένου γύρω από το ρολόι.

Εξατομίκευση:

Δημιουργία εξατομικευμένου περιεχομένου που βασίζεται σε (εξατομικευμένο) περιεχόμενο ή στη χρήση άλλου χαρακτηριστικού. Ο καθένας πιστεύει ότι η εφαρμογή του πρέπει να ικανοποιεί τις ανάγκες του και να «συμπεριφέρεται» όπως εκείνος επιθυμεί. Αυτό το συγκεκριμένο χαρακτηριστικό όχι μόνο καλύπτει το εξατομικευμένο περιεχόμενο αλλά επιθυμεί να ελέγχει τα κοινόχρηστα και τα αποθηκευμένα δεδομένα για περισσότερες δράσεις.

Ασφάλεια:

Υπάρχουν πολλές πτυχές της ασφάλειας. Τα δεδομένα που μεταφέρονται μέσω του δικτύου θα πρέπει να είναι κρυπτογραφημένα μέσω του δικτύου μεταφοράς. Κάποιες εφαρμογές συγχρονίζουν δεδομένα με online, διαδικτυακές εφαρμογές, οπότε η αποθήκευση των πληροφοριών στον server πρέπει να είναι ασφαλής. Μια άλλη κρίσιμη παραβίαση ασφάλειας είναι τα δεδομένα στην ίδια την κινητή συσκευή καθώς δεν θέλουμε κανένας να «παίξει» με το τηλέφωνο μας.

Αυτά μπορεί να είναι τα γενικά χαρακτηριστικά και τα χαρακτηριστικά, αλλά παίζουν ζωτικό ρόλο στον τομέα των κινητών εφαρμογών. Σχεδόν κάθε παροχέας υπηρεσιών λογισμικού προσφέρει τις υπηρεσίες ανάπτυξης εφαρμογών, αλλά μόνο λίγες ώριμες εταιρείες ενδιαφέρονται σοβαρά για αυτά τα χαρακτηριστικά και τα χαρακτηριστικά για να δώσουν την έσχατη mobile εφαρμογή για τον τελικό χρήστη. [It Technical Solutions, 2012, Some Essential Characteristics of Mobile Applications]

1.2.3 Κριτήρια αξιολόγησης εφαρμογών

Ενώ οι ίδιες οι εφαρμογές είναι ως επί το πλείστον πολύ διαφορετικές μεταξύ τους –όλες βοηθάνε σε διαφορετικές ανθρώπινες ανάγκες από το να παίζεις παιχνίδια ως τη βοήθεια στις αγορές – μερικές από τις βασικές λειτουργίες τους είναι ίδιες. Αυτό σημαίνει ότι έχουν κάποια κοινά χαρακτηριστικά, τα οποία μπορούμε να τα θέσουμε και σαν κριτήρια αξιολόγησής τους ως προς το ποιες εφαρμογές αποδεικνύονται να έχουν την καλύτερη λειτουργία/ απόδοση. [flamelab.de, Roland Koch, 2011, Six Major Characteristics of Mobile Apps]

1. Κάλυψη ανάγκης αγοράς: Η εφαρμογή επιλύει ένα πρόβλημα που χρειάζεται να λυθεί. Είναι απαραίτητο μια εφαρμογή να αναγνωρίζει μια ιδέα/καινοτομία που να απευθύνεται σε μια ανάγκη/θέμα της αγοράς και ένα κενό της αγοράς. Μια εφαρμογή για να γίνει επιτυχημένη, πρέπει να προσελκύσει μεγάλο πλήθος αγοραστών και αυτό μπορεί να επιτευχθεί όταν ικανοποιεί μια ανάγκη.
2. Αξιοποίηση κρατικού κεφαλαίου. Επιπλέον, θα μπορούσαμε να δείξουμε πως θα μπορούσαμε να αξιοποιήσουμε το κρατικό κεφάλαιο για να βελτιώσουμε την εφαρμογή μας.
3. Οφέλη/Αλλαγές. Οι επιπτώσεις/αλλαγές που θα επιφέρει στην κοινότητα (. Τα απτά οφέλη που θα έχουν οι επιχειρήσεις, οι πολίτες και ο δήμος και την προοπτική να προσθέσουμε μια επιπλέον επένδυση στην πόλη.
4. Ικανότητα προώθησης. Πρέπει να έχουμε την δυνατότητα και την ικανότητα για προωθήσουμε την ιδέα της εφαρμογής.
5. Βελτίωση εφαρμογής. Απόδειξη της ανάγκης για πρόσβαση σε παροχή εξειδικευμένων υπηρεσιών και την συνεργασία μας με τον φορέα παροχή υπηρεσιών να ταιριάζει με την ανάγκη μας να βελτιώσουμε την εφαρμογή μας.

6. Ανταγωνισμός/Πλεονέκτημα. Η προτεινόμενη ιδέα να έχει ένα ανταγωνιστικό πλεονέκτημα στην αγορά. Αν, για παράδειγμα, κάνοντας έρευνα παρατηρήσουμε ότι υπάρχουν ήδη τέτοιες εφαρμογές ή παρόμοιες στην αγορά, πρέπει να βελτιώσουμε/αναπτύξουμε την ιδέα μας ώστε να προσφέρουμε ένα πλεονέκτημα έναντι των άλλων εφαρμογών, κάτι που να μην υπάρχει αυτή την στιγμή στην αγορά και θα κάνει την εφαρμογή «ελκυστική» στους χρήστες για να την προτιμήσουν.
7. User Interface. Το UI μπορεί να προσαρμοστεί, να ελέγχεται και να επικυρώνεται για να εξασφαλίσει αυτό που ταιριάζει ακριβώς σε ό, τι χρειάζονται οι χρήστες της εφαρμογής. Λαμβάνοντας υπόψη την επίπτωση στην χρηστικότητα και την παραγωγικότητα, τη δυνατότητα να βελτιστοποιήσουμε το UI για τους χρήστες θα μπορούσε να δικαιολογήσει την ανάπτυξη μιας προσαρμοσμένης εφαρμογής κινητού.
8. Επεκτασιμότητα. Όσο αυξάνονται οι χρήστες, τόσο αυξάνεται και ο φόρτος της εφαρμογής. Για να μπορεί το σύστημα να ανταπεξέλθει στα πολλά αιτήματα, θα πρέπει να βελτιώνεται και να αναβαθμίζεται συνεχώς.

Εκτός από τα προαναφερθέντα κριτήρια, υπάρχουν και άλλες πτυχές που θα έπρεπε να λάβουμε υπόψη. [Josiah Humphrey, 2012, Appster, The Six Characteristics of the Top 100 Mobile Apps of All Time]

- 1) Η εφαρμογή θα πρέπει να παρέχει εξαιρετικά χρήσιμη υποστήριξη πελατών (Customer Support). Κάποιοι άνθρωποι μπορούν να χειριστούν πολύ τις τεχνολογικές συσκευές και κάποιοι άλλοι όχι. Θα υπάρχουν άνθρωποι που θα χρειαστούν βοήθεια με την εγκατάσταση της εφαρμογής, ίσως να έχουν ερωτήσεις πριν ή μετά την αγορά. Αν δεν έχουν αγοράσει ακόμη την εφαρμογή, η απάντηση στην ερώτησή τους μπορεί να είναι ο μόνος τρόπος να προχωρήσουν στην αγορά

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

της. Ακόμη κι αν ρωτήσουν αφού την αγόρασαν και δεν λάβουν απάντηση, τότε σίγουρα δεν θα την συστήσουν σε άλλους ούτε θα επιδιώξουν μελλοντικές κυκλοφορίες. Επίσης, όσο πιο γρήγορα έρθει η απάντηση τόσο το καλύτερο. Ορισμένες εταιρείες που έχουν δημιουργήσει πολλές εφαρμογές προσλαμβάνουν ωρομίσθιους και τους αναθέτουν την εξυπηρέτηση πελατών, χρησιμοποιώντας ιστοσελίδες όπως τις Elance.com και oDesk.com. Σημείωση: Οι περισσότεροι ιδιοκτήτες εφαρμογών δέχονται συνέχεια τις ίδιες ερωτήσεις, γι' αυτό δημιουργούν ένα τμήμα FAQ (Frequently Asked Questions) που οι πιθανοί αγοραστές μπορούν να διαβάσουν.

- 2) Κριτικές – Reviews. Πριν από την αγορά μιας εφαρμογής πολλοί άνθρωποι ψάχνουν στο διαδίκτυο για κριτικές της εφαρμογής. Σε εκείνες οι εφαρμογές οι οποίες έχουν καταγεγραμμένες πολλές κριτικές (γνήσιες από πραγματικούς χρήστες) θα παρέχουν βοήθεια σε πιθανούς αγοραστές και κατά πάσα πιθανότητα θα τους υποκινήσουν να τις αγοράσουν.
- 3) Η εφαρμογή δεν πρέπει να είναι υπερφορτωμένη με διαφημίσεις. Οι περισσότεροι άνθρωποι βρίσκουν ενοχλητικές τις διαφημίσεις στις ιστοσελίδες και τις εφαρμογές, ειδικά αν καλύπτουν το μισό interface μιας εφαρμογής! Οι καλύτεροι ιδιοκτήτες εφαρμογών το γνωρίζουν πολύ καλά αυτό. Ωστόσο, κάποιες από τις καλύτερες εφαρμογές και με τα περισσότερα downloads περιέχουν διαφημίσεις, χωρίς όμως να είναι επιβλητικές ή ενοχλητικές. Δεν εμφανίζονται στο κέντρο τη εφαρμογής αλλά είναι στρατηγικά τοποθετημένες στο πλάι, στο κάτω ή πάνω μέρος αλλά όχι κατευθείαν στο κέντρο εμποδίζοντας την ικανότητα να δούμε την ίδια την εφαρμογή. Τα έσοδα από τις διαφημίσεις παίζουν σημαντικό ρόλο σε μια τεχνολογική εταιρεία.
- 4) Η διάταξη (layout) του εσωτερικού χώρου της εφαρμογής οφείλει να είναι σαφής, με σταθερή ροή δεδομένων και να μην «κολλάει» (crash). Ένα καθαρό, έξυπνο περιβάλλον είναι αυτό που κάνει μια εφαρμογή να

ξεχωρίζει στους αγοραστές και να γίνει ένα καθοριστικός παράγοντας στην απόφαση τους να την αγοράσουν. Οι ομάδες σχεδίασης εφαρμογών, που είτε ανήκουν στην εταιρεία είτε είναι εξωτερικοί συνεργάτες, δεν έχουν χαμηλό κόστος. Οι εταιρείες των καλύτερων εφαρμογών ξοδεύουν ένα σημαντικό ποσό για να προσλάβουν υψηλής ποιότητας αυτόνομα άτομα να κάνουν το interface της εφαρμογής να δείχνει ωραίο και καλλιτεχνικό. Για να καταφέρει μια εφαρμογή να είναι σαφής και με σταθερή ροή δεδομένων, πρέπει να λάβουμε υπόψη διάφορα θέματα. Πρώτον, δεν μπορούμε να προσθέσουμε πάρα πολύ περιεχόμενο και πληροφορίες στην σχεδίαση ενός interface. Δεν γίνεται να υπερφορτωθεί με κουμπιά καθώς οι οθόνες των κινητών συσκευών είναι μικρές. Οι άνθρωποι δεν θέλουν να αλληθωρίζουν! Τα κουμπιά θα πρέπει να μπορούν να επιλέγονται εύκολα και να είναι ορατά. Το κείμενο επίσης πρέπει να είναι σε ένα σωστό, ευκολοδιάβαστο μέγεθος – ακόμα και για άτομα μεγαλύτερης ηλικίας. Όσον αφορά το θέμα της σωστής λειτουργίας της εφαρμογής και να μην «κολλάει», είναι πολύ σημαντικό. Πολλές εφαρμογές «κολλάνε» συχνά λόγω της φθηνής σχεδίασής τους (κύρια αιτία είναι η υπερφόρτωση), το οποίο αναγκάζει τους χρήστες να κάνουν επανεκκίνηση της κινητής συσκευής κάθε λίγο επειδή παγώνει η οθόνη. Αυτό θα προκαλέσει παράπονα εκ μέρους των χρηστών και θα παρέχει αρνητικές κριτικές σε αυτήν την περίπτωση.

1.3 Πρότυπα Αξιοπιστίας των Εφαρμογών

Μια εφαρμογή για να θεωρηθεί επιτυχημένη, εκτός από την σωστή λειτουργία, πρέπει να θεωρείται και αξιόπιστη. Αυτό σημαίνει ότι είναι αναγκαίο να πάρει τα κατάλληλα μέτρα για την προστασία από απομακρυσμένους αντιπάλους και έχει την κατάλληλη προστασία ενάντια σε ένα ευρύ φάσμα απειλών. Υπάρχουν κάποια πρότυπα/κριτήρια που οφείλει να τηρήσει μια εφαρμογή όσον αφορά την αξιοπιστία. [PLYNT, 2011, Plynt Certification Criteria for Mobile Applications]

- **(Μη) Ασφαλές αποθήκευση ευαίσθητων δεδομένων στην συσκευή.** Η εφαρμογή πρέπει να επιδείξει ότι έχει πάρει επαρκή μέτρα για την προστασία των ευαίσθητων δεδομένων από την πρόσβαση τους σε περίπτωση που η συσκευή χαθεί ή κλαπεί. Η εφαρμογή δεν πρέπει να αποθηκεύει τα ευαίσθητα δεδομένα σε εύκολα προσβάσιμες συσκευές, όπως μια κάρτα μνήμης.
- **Προστασία από διαρροή δεδομένων.** Η εφαρμογή δεν πρέπει να παρουσιάσει διαρροή δεδομένων μέσω καναλιών δικτύου, όπως το cache, logs, προσωρινούς καταλόγους κλπ. όπου τα δεδομένα συνήθως διατηρούνται επ' αόριστον. (side channel attacks)
- **Προστασία από γνωστές επιθέσεις server-side.** Η εφαρμογή πρέπει να αποδείξει, μέσω δοκιμών, ότι δεν είναι ευάλωτη σε γνωστές επιθέσεις τύπου server-side.
- **Υπεράσπιση εναντίον ενός Threat Profile.** Η εφαρμογή πρέπει να αποδείξει ότι μπορεί να υπερασπίζεται τον εαυτό της ενάντια στις απειλές που καθορίζονται σε ένα προφίλ απειλής που έχει αναπτυχθεί ειδικά για την εφαρμογή για κινητά. Το Threat Profile είναι μια λίστα με όλες τις πιθανές απειλές για την εφαρμογή. Αυτές οι απειλές περιλαμβάνουν την παραβίαση των κανόνων των επιχειρήσεων και των κανόνων έγκρισης της εφαρμογής. Αυτό το κριτήριο ισχύει για τις ευπάθειες υψηλού και μέσου επιπέδου που επιτρέπουν τέτοιες απειλές να υλοποιηθούν.
- **Τα ευαίσθητα δεδομένα δεν στέλνονται σε άλλες ιστοσελίδες/εφαρμογές με μη ασφαλή τρόπο.** Η εφαρμογή αυτή δεν πρέπει να στείλει ευαίσθητα δεδομένα σε εξωτερικές σελίδες μέσω tokens, αναφορών κ.ά. Αν αυτή η εφαρμογή χρειάζεται να μεταφέρει δεδομένα μέσω άλλων εφαρμογών στην ίδια συσκευή, τότε τα δεδομένα πρέπει να μοιραστούν με ασφαλή τρόπο.

- **Ασφαλής εκτέλεση εφαρμογών που σχετίζονται με το λειτουργικό σύστημα.** Τα λειτουργικά συστήματα των κινητών τηλεφώνων υλοποιούν λειτουργίες που χρησιμοποιούν χαρακτηριστικά τύπου OS-related. Η εφαρμογή πρέπει να υλοποιήσει αυτά τα χαρακτηριστικά με ασφάλεια, χωρίς να παρουσιάσει ευπάθειες. Δεν περιλαμβάνει ευπάθειες τύπου OS-related που σχετίζονται με χαρακτηριστικά που δεν χρησιμοποιούνται από την εφαρμογή.
- **Απαιτείται επαλήθευση της ταυτοποίησης για τα ευαίσθητα δεδομένα.** Η εφαρμογή πρέπει να επαναλάβει τον έλεγχο ταυτότητας του χρήστη προτού επιτρέψει στο χρήστη να εκτελέσει μια πράξη που περιλαμβάνει ευαίσθητα δεδομένα. Παραδείγματα λειτουργιών που αφορούν τα ευαίσθητα δεδομένα είναι - Αλλαγή κωδικού πρόσβασης, έγκριση κεφαλαίων και συναλλαγών.
- **Υπεράσπιση εναντίων μη εξουσιοδοτημένης χρήσης των πόρων της συσκευής.** Η εφαρμογή πρέπει να διασφαλίσει ότι μια μη αξιόπιστη είσοδος είναι επικυρωμένη προτού χρησιμοποιηθεί από οποιονδήποτε πόρο συσκευής. Η εφαρμογή δεν πρέπει να κάνει κατάχρηση των πόρων της συσκευής. Οι πόροι των κινητών συσκευών περιλαμβάνουν (αν και δεν περιορίζονται μόνο σε αυτά) κλήσεις, κάμερα, μικρόφωνο και SMSs. Αυτό το κριτήριο ελέγχει την χρήση των πόρων των συσκευών, όπως μπορούν να ελεγχθούν από τα αποτελέσματα των δοκιμών παρακολούθησης της συσκευής και των πόρων στην τρέχουσα κινητή συσκευή.
- **Backend υπηρεσίες προστατεύονται από γνωστές ευπάθειες.** Οι Mobile εφαρμογές συνδέονται σε backend υπηρεσίες, προκειμένου να στείλουν και να λάβουν δεδομένα. Οι υπηρεσίες αυτές πρέπει να προστατεύονται έναντι τρωτών σημείων που είναι άμεσα εκμεταλλεύσιμα σε όλη την εφαρμογή. Το κριτήριο αυτό δεν εξετάζει τα εκμεταλλεύσιμα τρωτά σημεία στο backend (μια διαδικτυακή υπηρεσία

ή μια βάση δεδομένων), το οποίο δεν μπορεί να προσεγγιστεί άμεσα από την εφαρμογή για κινητά που δοκιμάζεται.

- **Backend εξυπηρετητής προστατεύεται από γνωστές ευπάθειες.** Ο εξυπηρετητής πρέπει να αναβαθμίζεται και να προστατεύεται ενάντια σε γνωστές ευπάθειες. Η διαδικτυακή υπηρεσία που «τρέχει» στον εξυπηρετητή που φιλοξενεί την εφαρμογή δεν πρέπει να είναι ευάλωτη σε δημοσίως γνωστές εκμεταλλεύσιμες ευπάθειες.
- **Προστασία ευαίσθητων δεδομένων κατά την μεταφορά.** Η εφαρμογή πρέπει να λάβει κατάλληλα μέτρα για την προστασία των ευαίσθητων δεδομένων κατά της κλοπής από το δίκτυο. Θα πρέπει να προστατευτούν τα δεδομένα στον τομέα της μετάδοσης με την υλοποίηση μιας ισχυρής κρυπτογράφησης
- **Δεν πρέπει να περιέχονται ευαίσθητα δεδομένα στον πηγαίο κώδικα.** Ο πηγαίος κώδικας της εφαρμογής θα πρέπει να προστατεύεται από τη διαρροή δεδομένων μέσω αποσυναρμολόγησης του πακέτου εφαρμογών. Η εφαρμογή δεν θα πρέπει να εισάγει απόλυτα και αμετάβλητα (hardcodeS) μυστικά, όπως τους κωδικούς ή τα κλειδιά κρυπτογράφησης στην ίδια την εφαρμογή.

1.4 Χαρακτηριστικά των Διαδικτυακών Υπηρεσιών Εξυπηρέτησης Πολιτών

1.4.1 Διαδικτυακές Υπηρεσίες – Ορισμός

Σύμφωνα με τον W3C (World Wide Web Consortium) ο ορισμός της διαδικτυακής υπηρεσίας είναι ο εξής:

«Μια διαδικτυακή υπηρεσία είναι ένα σύστημα λογισμικού που έχει σχεδιαστεί για να υποστηρίζει την διαλειτουργική αλληλεπίδραση μηχανής-προς-μηχανής μέσω ενός δικτύου. Έχει μια διεπαφή που περιγράφεται σε μια μορφή που μπορεί να επεξεργαστεί από μηχανή (συγκεκριμένα WSDL). Άλλα συστήματα αλληλεπιδρούν με την διαδικτυακή υπηρεσία με έναν τρόπο που προβλέπεται

από την περιγραφή χρησιμοποιώντας μηνύματα τύπου SOAP, τυπικά μεταφέρονται χρησιμοποιώντας HTTP με μια XML serialization σε συνδυασμό με άλλα πρότυπα του παγκόσμιου ιστού.» [W3C, W3C Working Group, 2004, Web Services Architecture]

Η Microsoft μέσα από το MSDN της καταλήγει ότι όλες οι διαδικτυακές υπηρεσίες έχουν τρία κοινά χαρακτηριστικά:

- Εκθέτουν χρήσιμη λειτουργικότητα σε χρήστες του διαδικτύου μέσα από ένα πρότυπο δικτυακό πρωτόκολλο. Στις περισσότερες περιπτώσεις αυτό το πρωτόκολλο είναι το SOAP (Simple Object Access Protocol).
- Παρέχουν ένα τρόπο να περιγράψουν τις διεπαφές τους με αρκετή λεπτομέρεια ώστε να επιτρέψουν στο χρήστη τους να χτίσει μια εφαρμογή πελάτη η οποία να επικοινωνήσει μαζί τους. Η περιγραφή συνήθως παρέχεται σε ένα έγγραφο XML το οποίο ονομάζεται έγγραφο WSDL (Web Services Description Language).
- Καταχωρούνται ώστε οι δυνητικοί χρήστες να μπορούν να τα βρουν εύκολα. Αυτό γίνεται με το UDDI (Universal Discovery Description and Integration). [MSDN, 2001, Roger Wolter, XML Web Services Basics]

1.4.2 Πλεονεκτήματα και Μειονεκτήματα των Διαδικτυακών Υπηρεσιών

Οι διαδικτυακές υπηρεσίες παρέχουν καθιερωμένους τρόπους διαλειτουργικότητας μεταξύ διαφορετικών εφαρμογών λογισμικού, που «τρέχουν» σε μια ποικιλία από πλατφόρμες ή/και πλαίσια.

Πλεονεκτήματα των διαδικτυακών υπηρεσιών:

- **Διαλειτουργικότητα:** Η χαλαρή σύζευξή τους οδηγεί στην ανάπτυξη ευέλικτου λογισμικού στα κατανεμημένα πληροφοριακά περιβάλλοντα.
- **Ενσωμάτωση:** Επιτρέπουν την εύκολη και γρήγορη ενσωμάτωση μειώνοντας σημαντικά το κόστος ανάπτυξης λογισμικού.

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

- **Επεκτασιμότητα:** Έχουν την δυνατότητα εκσυγχρονισμού πληροφοριακών συστημάτων που έχουν δημιουργηθεί με παλαιές τεχνολογίες.
- **Επαναχρησιμοποίηση:** Μπορούν να επαναχρησιμοποιηθούν και να βελτιστοποιηθούν με ευκολία ενσωματώνοντας νέα χαρακτηριστικά.
- **Αναβάθμιση:** Περιέχουν την περιγραφή στο εσωτερικό τους οπότε είναι εύκολο να χρησιμοποιηθούν από πληροφοριακά συστήματα και να τα αναβαθμίσουν.
- Οι υπηρεσίες ιστού χρησιμοποιούν υπάρχοντα **πρότυπα** για την ανάπτυξή τους.
- Απλοποιούν την ενσωμάτωση συστημάτων μεταξύ εταιριών (**Business to Business Integration**).
- Εισάγουν νέα δεδομένα στην δημιουργία εταιρικών μοντέλων.
- Μπορούν να συγκεντρώσουν με χαμηλό κόστος τις επιχειρηματικές διαδικασίες σε ένα κοινό σημείο πρόσβασης.

Μειονεκτήματα των διαδικτυακών υπηρεσιών:

- Βρίσκονται ακόμη σε μεταβατικό στάδιο και δεν έχουν ωριμάσει.
- Χρησιμοποιούν μεγάλη ποσότητα δεδομένων στην ανταλλαγή μηνυμάτων. Τα μηνύματα αποτελούνται από μπλοκ XML και μεταδίδονται σε αυτή τη μορφή. Δε χρησιμοποιείται η δυαδική μέθοδος που είναι αρκετά πιο γρήγορη. [ATEI, 2013, Ηλιούδης Χρήστος, Υπηρεσιοστρεφής Αρχιτεκτονική]

1.4.3 Αρχιτεκτονική Διαδικτυακών Υπηρεσιών

Μια διαδικτυακή υπηρεσία, λοιπόν, είναι μια αόριστη έννοια που πρέπει να υλοποιηθεί από έναν συγκεκριμένο παράγοντα (agent). Ο παράγοντας αυτός είναι ένα συγκεκριμένο κομμάτι του λογισμικού ή του υλικού που στέλνει και λαμβάνει μηνύματα, ενώ η υπηρεσία είναι η πηγή που χαρακτηρίζεται από το

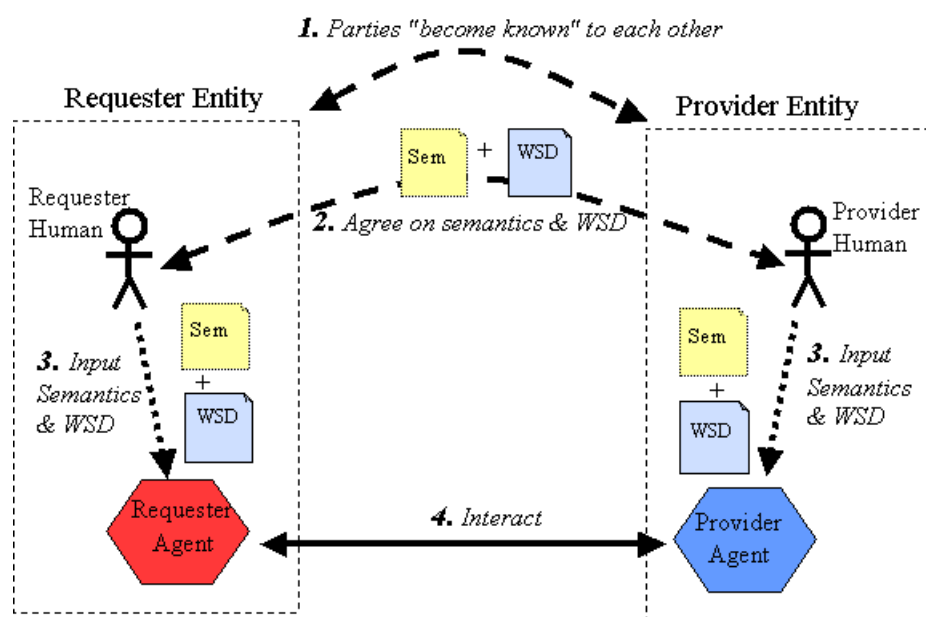
αφηρημένο κομμάτι που παρέχεται. Για να φανεί η διάκριση αυτή, μπορούμε να υλοποιήσουμε μια συγκεκριμένη διαδικτυακή υπηρεσία χρησιμοποιώντας έναν παράγοντα την μια μέρα και έναν διαφορετικό παράγοντα την επόμενη μέρα (γραμμένο επίσης σε άλλη γλώσσα προγραμματισμού) με την ίδια λειτουργικότητα. Παρά το γεγονός ότι ο παράγοντας έχει αλλάξει, η διαδικτυακή υπηρεσία παραμένει ίδια.

Ο σκοπός της διαδικτυακής υπηρεσίας είναι να παρέχει κάποιες λειτουργίες για λογαριασμό του ιδιοκτήτη του - ένα πρόσωπο ή ένας οργανισμός, όπως μια επιχείρηση ή ένα άτομο. Η οντότητα παροχέας είναι το πρόσωπο ή η οργάνωση που παρέχει ένα κατάλληλο μέσο για την υλοποίηση μιας συγκεκριμένης υπηρεσίας. Μια οντότητα αιτών είναι ένα πρόσωπο ή ένας οργανισμός που επιθυμεί να κάνει χρήση της διαδικτυακής υπηρεσίας μιας οντότητας του παροχέα. Θα χρησιμοποιήσει έναν παράγοντα αιτούντα (requester agent) για την ανταλλαγή μηνυμάτων με τον agent του παροχέα της οντότητας του παροχέα.

Για να είναι επιτυχής αυτή η ανταλλαγή μηνυμάτων, η οντότητα του αιτούντα και η οντότητα του παροχέα πρέπει να συμφωνήσουν και οι δύο για τη σημασιολογία και τον μηχανισμό της ανταλλαγής μηνυμάτων. Οι μηχανισμοί ανταλλαγής μηνυμάτων τεκμηριώνονται σε μια περιγραφή της διαδικτυακής υπηρεσίας (WSD).

Το WSD είναι μια προδιαγραφή (που μπορεί να επεξεργαστεί από μηχανή) της διεπαφής της διαδικτυακής υπηρεσίας, γραμμένο σε WSDL. Στην ουσία, η περιγραφή των υπηρεσιών αντιπροσωπεύει μια συμφωνία που διέπει τους μηχανισμούς αλληλεπίδρασης με την εν λόγω υπηρεσία. Η σημασιολογία μιας διαδικτυακής υπηρεσίας είναι η κοινή προσδοκία για τη συμπεριφορά της υπηρεσίας, ιδίως σε απάντηση στα μηνύματα που αποστέλλονται σε αυτό. Στην πραγματικότητα, αυτή είναι η «σύμβαση» μεταξύ της οντότητας αιτούντα και του φορέα παροχέα όσον αφορά το σκοπό και τις συνέπειες της αλληλεπίδρασης. [W3C, W3C Working Group, 2004, Web Services Architecture]

Η παρακάτω εικόνα δείχνει την γενική διαδικασία δέσμευσης μιας διαδικτυακής υπηρεσίας:



Εικόνα 1: Η γενική διαδικασία δημιουργίας της διαδικτυακής υπηρεσίας

1.4.4 Χρήση των Διαδικτυακών Υπηρεσιών στο M-Commerce

Λαμβάνοντας υπόψη τις εξελίξεις στην τεχνολογία, συγκεκριμένα στον τομέα της κινητής τηλεφωνίας, φαίνεται ότι η χρήση των διαδικτυακών υπηρεσιών σε κινητές συσκευές ανοίγει την δυνατότητα για παροχή υπηρεσιών επί τόπου, ανεξάρτητα από το που βρίσκεται ο αιτών ή ο παροχέας των υπηρεσιών συντήρησης. Η δυναμική ανακάλυψη και η επίκληση των υπηρεσιών που η τεχνολογία των διαδικτυακών υπηρεσιών επιτρέπει να συντονίζονται με οράματα για τις mobile εφαρμογές, όπου το περιβάλλον του χρήστη μπορεί να αλλάζει δυναμικά, κάνοντας διάφορες υπηρεσίες ή υλοποιήσεις υπηρεσιών κατάλληλες σε διαφορετικές χρονικές στιγμές.

Κατά την εξέταση του κόσμου των κινητών συσκευών – με δυνατότητα σύνδεσης στο διαδίκτυο, οι τεχνικές προκλήσεις που συνδέονται με την ανάπτυξη του πλαισίου των διαδικτυακών υπηρεσιών χρειάζονται να επεκταθούν ώστε να περιλαμβάνουν θέματα όπως η επίδοση (αποτελεσματική λειτουργία κάτω από περιορισμένες συνθήκες bandwidth), το κόστος και η προσωρινή μη διαθεσιμότητα (λόγω των διακοπών σύνδεσης). Ορισμένα από τα πρωτόκολλα που χρησιμοποιούνται στον κόσμο

των διαδικτυακών υπηρεσιών των σταθερών συσκευών είναι τελείως ακατάλληλα για χρήση στο ακριβό ασύρματο μέσο. Εκτός από το θέμα του εύρους ζώνης, οι ασύρματες συσκευές έχουν επίσης και άλλα μειονεκτήματα όπως χαμηλή CPU και χωρητικότητα μνήμης που πρέπει να ληφθούν υπόψη.

Σημαντικοί εμπορικοί φορείς όπως η Microsoft και η SUN αναγνωρίζουν την σημασία της επέκτασης του προτύπου των διαδικτυακών υπηρεσιών στον ασύρματο κόσμο και να εισάγουν συγκεκριμένα χαρακτηριστικά στα προϊόντα τους. Η Microsoft έχει κυκλοφορήσει το .NET Compact Framework Beta στις αρχές του 2002. Η SUN σχεδιάζει να επεκτείνει την J2ME πλατφόρμα της για να υποστηρίξει πλήρως τις διαδικτυακές υπηρεσίες. Αυτό ως επί το πλείστον επιδιώκεται μέσω του JSR (Java Specification Request) No 172. [Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών, Τ. Πιλιούρα, Α. Τσαλαγατίδου, Σ. Χατζηευθυμιάδης, 2003, Scenarios of using Web Services in M-Commerce]

1.4.5 Mobile Commerce

Η φράση Mobile Commerce επινοήθηκε αρχικά το 1997 από τον Kevin Duffey και σημαίνει «η παράδοση των ικανοτήτων του ηλεκτρονικού εμπορίου απευθείας στο χέρι του καταναλωτή, οπουδήποτε, μέσω της ασύρματης τεχνολογίας». Πολλοί επιλέγουν να σκέφτονται το Mobile Commerce ως «ένα κατάστημα λιανικής πώλησης στην τσέπη του πελάτη». [Wikipedia, 2010-2013, Mobile Commerce]

Ένας άλλος ορισμός για το Mobile Commerce είναι ο εξής:

«Η χρήση ασύρματων φορητών συσκευών, όπως τα κινητά τηλέφωνα και οι φορητοί υπολογιστές για τη διεξαγωγή εμπορικών συναλλαγών σε απευθείας σύνδεση. Οι συναλλαγές στο Mobile Commerce συνεχίζουν να αυξάνονται και ο όρος περιλαμβάνει την αγορά και την πώληση μιας μεγάλης ποικιλίας αγαθών και υπηρεσιών, online banking, πληρωμή λογαριασμών, παροχή υπηρεσιών κ.ά. Επίσης γνωστό σαν M-Commerce.»

Το φάσμα των συσκευών που ενεργοποιούνται για το M-Commerce αυξάνεται, έχοντας επεκταθεί τα τελευταία χρόνια ώστε να συμπεριλαμβάνει τα smartphones και τα tablets. Η αυξανόμενη υιοθέτηση του ηλεκτρονικού

εμπορίου παρέχει ένα ισχυρό θεμέλιο για το M-Commerce, το οποίο βρίσκεται σε μια πολύ ισχυρή τροχιά ανάπτυξης για τα επόμενα χρόνια.

Η ταχεία ανάπτυξη του M-Commerce οδηγείται από μια σειρά θετικών παραγόντων : (1) την ζήτηση για εφαρμογές από μια όλο και μεγαλύτερη κινητικότητα των πελατών και της βάσης των καταναλωτών, (2) την ταχεία υιοθέτηση του ηλεκτρονικού εμπορίου, χάρη στην ανάλυση των θεμάτων ασφάλειας και (3) τις τεχνολογικές προόδους που έχουν δώσει ασύρματες φορητές συσκευές, προηγμένες δυνατότητες και σημαντική υπολογιστική ισχύ. [Investopedia, 2013, Mobile Commerce]

Σύμφωνα με την BI Intelligence τον Ιανουάριο του 2013, το 29 τοις εκατό των χρηστών κινητής τηλεφωνίας έχουν πλέον κάνει μια αγορά με τα τηλέφωνα τους. Η Walmart εκτίμησε ότι το 40 τοις εκατό όλων των επισκέψεων στην ιστοσελίδα τους για διαδικτυακές αγορές τον Δεκέμβριο του 2012 ήταν από μια κινητή συσκευή. Η τράπεζα της Αμερικής προβλέπει ότι 67.1 δισεκατομμύρια δολάρια σε αγορές θα πραγματοποιούνται από τις κινητές συσκευές από ευρωπαίους και αμερικανούς αγοραστές το 2015. Μόνο οι λιανοπωλητές του M-Commerce στο Ηνωμένο Βασίλειο αναμένεται να αυξήσουν τα έσοδα έως και 31 τοις εκατό στο τέλος του 2013-14. [Wikipedia, 2010-2013, Mobile Commerce]

1.4.6 Διαθέσιμα προϊόντα και υπηρεσίες

Μέσω του M-Commerce, οι πολίτες μπορούν να εξυπηρετηθούν χρησιμοποιώντας μια ποικιλία εφαρμογών για να αγοράσουν/πουλήσουν διάφορα προϊόντα αλλά και να έχουν εύκολη πρόσβαση σε διάφορες υπηρεσίες, σπαταλώντας λιγότερο χρόνο και κόπο.

- **Μεταφορά Χρημάτων:** Στην Κένυα η μεταφορά χρημάτων πραγματοποιείται κυρίως μέσω των κινητών τηλεφώνων. Αυτή ήταν μια πρωτοβουλία μιας εταιρείας στην Κένυα.
- **ATM:** Με την εισαγωγή των υπηρεσιών για συναλλαγές στην κινητή τηλεφωνία, οι επιχειρηματίες ψάχνουν τώρα αποτελεσματικούς τρόπους για να αναπτύξουν και να διαχειρίζονται δίκτυα διανομής που

μπορούν να υποστηρίξουν την είσοδο/έξοδο μετρητών. Αντίθετα με τα παραδοσιακά ATM, το sicas Mobile ATM έχει σχεδιαστεί ειδικά για να συνδέεται σε πλατφόρμες συναλλαγών της κινητής τηλεφωνίας και να παρέχει τον βαθμό ποιότητας του ATM μιας τράπεζας. Στην Ουγγαρία, η Vodafone επιτρέπει πληρωμές σε μετρητά ή σε κάρτες τραπέζης για μηνιαίους τηλεφωνικούς λογαριασμούς. Η ουγγρική αγορά είναι εκείνη όπου οι άμεσες χρεώσεις δεν αποτελούν συνήθη πρακτική, έτσι διευκολύνεται το βάρος της αναμονής για το μισό του συμβολαίου της συνδρομητικής βάσης της Vodafone στην Ουγγαρία.

- **Εισιτήρια:** Τα εισιτήρια μπορούν να σταλούν σε κινητά τηλέφωνα χρησιμοποιώντας μια ποικιλία τεχνολογιών. Οι χρήστες τότε μπορούν να χρησιμοποιήσουν τα εισιτήρια αμέσως, παρουσιάζοντας τα κινητά τηλέφωνα στο ταμείο των εισιτηρίων. Η πλειοψηφία των χρηστών τείνει προς αυτήν την τεχνολογία. Το καλύτερο παράδειγμα είναι το IRCTC όπου τα εισιτήρια φθάσουν στην μορφή των SMS στους χρήστες.
- **Απόδειξη πληρωμής, κουπόνια και κάρτες:** Η τεχνολογία για τα εισιτήρια μπορεί να χρησιμοποιηθεί επίσης και για την κατανομή των αποδείξεων, των κουπονιών και των καρτών. Αυτά τα αντικείμενα αντιπροσωπεύονται από ένα εικονικό token που στέλνεται στο κινητό τηλέφωνο. Ο πελάτης παρουσιάζει το κινητό τηλέφωνο με ένα από αυτά τα tokens στο σημείο πώλησης και λαμβάνει τα ίδια οφέλη όπως αν είχε το παραδοσιακό token. Τα μαγαζιά μπορεί να στείλουν κουπόνια σε πελάτες χρησιμοποιώντας υπηρεσίες βασισμένες στην τοποθεσία (location-based services) για να καθορίσουν πότε ο πελάτης είναι σε κοντινή απόσταση.
- **Αγορά και παράδοση περιεχομένου:** Επί του παρόντος, η αγορά κινητής τηλεφωνίας αποτελείται κυρίως από την πώληση ring-tones, wallpapers και παιχνιδιών για κινητές συσκευές. Η σύγκλιση των κινητών τηλεφώνων, των φορητών συσκευών αναπαραγωγής ήχων και βίντεο σε μια ενιαία συσκευή είναι η αύξηση της αγοράς και η παράδοση ολόκληρων μουσικών κομματιών και βίντεο. Οι ταχύτητες που διατίθενται με τα 4G δίκτυα καθιστούν ικανή την αγορά μια ταινίας σε μια κινητή συσκευή μέσα σε δευτερόλεπτα.

- **Υπηρεσίες με βάση την τοποθεσία:** Η τοποθεσία του χρήστη της κινητής συσκευής είναι ένα σημαντικό κομμάτι πληροφορίας που χρησιμοποιείται κατά τις συναλλαγές στο M-Commerce. Η εύρεση της τοποθεσίας των χρηστών επιτρέπει υπηρεσίες όπως: τοπική εκπαιδευτικές προσφορές, τοπικό δελτίο καιρού, εντοπισμός και παρακολούθηση ανθρώπων.
- **Υπηρεσίες πληροφοριών:** Μια μεγάλη ποικιλία υπηρεσιών πληροφοριών μπορεί να παραδοθεί σε χρήστες κινητών τηλεφώνων με τον ίδιο τρόπο όπως παραδίδεται στους υπολογιστές. Αυτές οι υπηρεσίες περιλαμβάνουν: νέα, μετοχές, αποτελέσματα αθλημάτων, οικονομικό ιστορικό, αναφορά κυκλοφορίας.
- **Mobile Banking:** Οι τράπεζες και άλλα χρηματοπιστωτικά ιδρύματα χρησιμοποιούν το M-Commerce για να επιτρέπουν στους πελάτες τους να έχουν πρόσβαση στις πληροφορίες του λογαριασμού τους και να κάνουν συναλλαγές, όπως η αγορά αποθεμάτων ή εμβάσματα.
- **Mobile brokerage:** Υπηρεσίες χρηματιστηρίου που προσφέρονται μέσω κινητών συσκευών έχουν γίνει επίσης πιο δημοφιλείς. Επιτρέπουν στον συνδρομητή να αντιδράσει στις εξελίξεις της αγοράς έγκαιρα και ανεξάρτητα από τη φυσική του θέση.
- **Δημοπρασίες:** Αντίθετα από τις παραδοσιακές δημοπρασίες, η αντίστροφη δημοπρασία (ή δημοπρασία χαμηλής προσφοράς) χρεώνει το τηλέφωνο του καταναλωτή κάθε φορά που τοποθετεί μια προσφορά.
- **Mobile browsing:** Χρησιμοποιώντας ένα φυλλομετρητή κινητής συσκευής – ένας World Wide Web browser σε κινητό τηλέφωνο – οι πελάτες μπορούν να ψωνίσουν σε απευθείας σύνδεση χωρίς να χρειάζονται τον προσωπικό τους υπολογιστή.
- **Mobile purchase:** Οι έμποροι καταλόγων μπορούν να δέχονται εντολές από τους πελάτες ηλεκτρονικά, μέσω της κινητής συσκευής του πελάτη. Σε ορισμένες περιπτώσεις, ο έμπορος μπορεί ακόμα να παραδώσει τον κατάλογο ηλεκτρονικά αντί να ταχυδρομήσει έναν κατάλογο στον πελάτη. Ορισμένοι έμποροι παρέχουν mobile

ιστοσελίδες που είναι προσαρμοσμένες για την μικρότερη οθόνη και με περιορισμένη διεπαφή χρήστη μιας κινητής συσκευής.

- **In-application mobile phone payments:** Οι πληρωμές μπορούν να γίνου απευθείας μέσα από μια εφαρμογή που «τρέχει» σε ένα δημοφιλές λειτουργικό σύστημα ενός Smartphone, όπως το Google Android. Η εταιρεία αναλύσεων Gartner αναμένει ότι οι αγορές των εφαρμογών αυτών θα αποτελούν το 41 τοις εκατό των εσόδων του app store το 2016.
- **Mobile marketing and advertising:** Οι εταιρείες αναφέρουν ότι έχουν δει καλύτερη ανταπόκριση στις mobile καμπάνιες marketing από τις παραδοσιακές καμπάνιες. Ο κύριος λόγος για αυτό είναι η άμεση φύση της λήψης της απόφασης των πελατών που επιτρέπουν οι mobile εφαρμογές και ιστοσελίδες. Ο καταναλωτής μπορεί να λάβει ένα μήνυμα μάρκετινγκ ή ένα εκπτώτικό κουπόνι και μέσα σε λίγα δευτερόλεπτα να πάρει μια απόφαση για να αγοράσει και να συνεχίσει για να ολοκληρωθεί η πώληση· χωρίς να διαταραχθεί η τρέχουσα δραστηριότητά του στον πραγματικό κόσμο. [Wikipedia, 2010-2013, Mobile Commerce]


1.5 Επίλογος

Στο κεφάλαιο αυτό μελετήθηκαν τα χαρακτηριστικά και τα κριτήρια της διαδικτυακής υπηρεσίας. Αναλύθηκαν οι ορισμοί της καινοτομίας και των διαδικτυακών υπηρεσιών καθώς και τα πλεονεκτήματα και τα μειονεκτήματά τους και τέλος η θέση τους στο Mobile Commerce. Στο επόμενο κεφάλαιο θα δούμε παραδείγματα υπηρεσιών που διαθέτουν τα χαρακτηριστικά που αναφέραμε σε αυτό το κεφάλαιο και θα τις αξιολογήσουμε συγκριτικά.

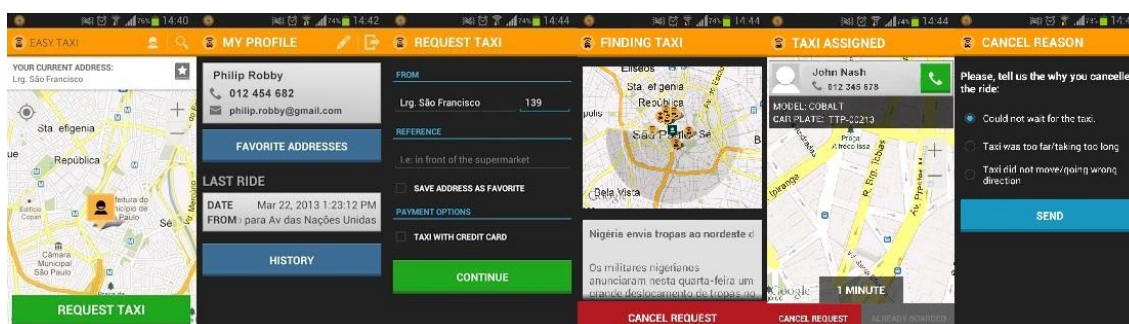
ΚΕΦΑΛΑΙΟ 2: Υφιστάμενες υπηρεσίες: συγκριτική αξιολόγηση

Στο κεφάλαιο αυτό αναλύονται εφαρμογές που έχουν υλοποιηθεί για το λειτουργικό σύστημα του Android και που οι υπηρεσίες είναι παρόμοιες με την διαδικτυακή υπηρεσία που θα αναπτύξουμε. Παρουσιάζεται επίσης μια συγκριτική αξιολόγησή τους.

2.1 Παροχή παραπλήσιων υπηρεσιών

- 1)  Easy taxi – Taxi via mobile (Google – Android app)

Με την εφαρμογή αυτή μπορούμε να καλέσουμε ένα ταξί μέσω ενός Smartphone με την χρήση του GPS και του διαδικτύου κάθε μέρα επί 24ώρου βάσεως. Η εφαρμογή θα εντοπίσει το που βρισκόμαστε την δεδομένη στιγμή (περιοχή και διεύθυνση). Επιλέγουμε να καλέσουμε ταξί (“Request Taxi”), συμπληρώνουμε τα πεδία που μας εμφανίζει και περιμένουμε το ταξί. Είναι real-time εφαρμογή, άρα μπορούμε ακόμη και να εντοπίσουμε το ταξί ή να καλέσουμε τον οδηγό. Επίσης, όλοι οι οδηγοί ταξί είναι εγγεγραμμένοι οπότε μπορούμε να γνωρίζουμε το όνομα τους και την φωτογραφία τους πριν φθάσουν. Δεν υπάρχει επιπλέον χρέωση, το ταξίμετρο ξεκινά όταν επιβιβαστούμε στο ταξί.



Εικόνα 2: Easy Taxi App

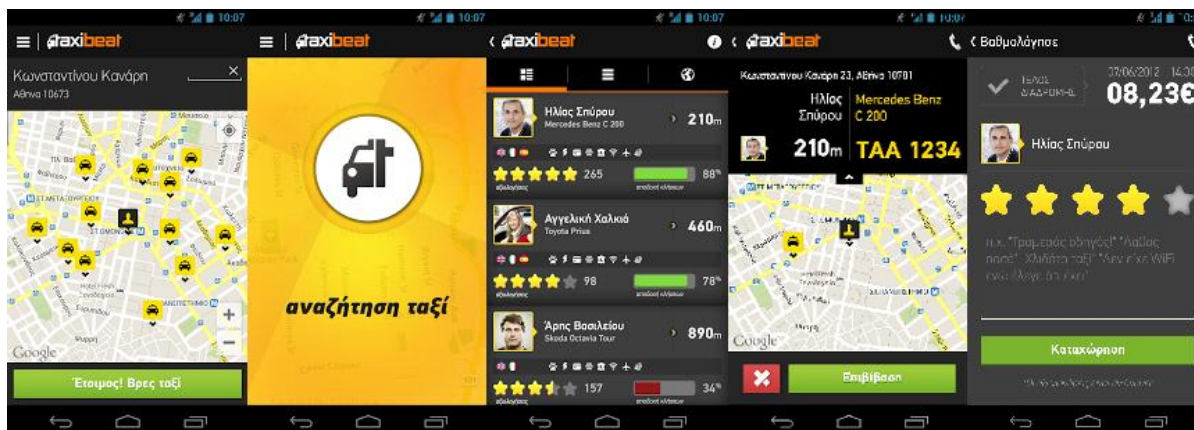
Παρόμοιες εφαρμογές είναι: 99Taxis, WayTaxi, SaferTaxi, TaxiJa, ResolveΑί, Mytekxi, Gett –Taxi & Black Car Service

2) Taxibeat (Google – Android app)

Μια ελληνική εφαρμογή για να καλέσουμε ταξί μέσω κινητού τηλεφώνου είναι το Taxibeat. Μας δίνει τη δυνατότητα να εντοπίσουμε και να καλέσουμε όποιον από τους συνεργαζόμενους οδηγούς ταξί θέλουμε, να μας παραλάβει από όπου είμαστε **χωρίς** καμία επιπλέον **χρέωση**.

Αν βιαζόμαστε, καλούμε τον οδηγό που είναι πιο κοντά μας. Αν επιθυμούμε να σερφάρουμε κατά τη διάρκεια της διαδρομής, τότε καλούμε οδηγό ταξί που προσφέρει δωρεάν WiFi. Κάθε οδηγός είναι βαθμολογημένος από άλλους Taxibeaters. Μπορούμε να είμαστε σίγουροι ότι οι οδηγοί που συνεργάζονται με το Taxibeat θα κάνουν τα πάντα για να μας προσφέρουν την ιδανική διαδρομή, γνωρίζοντας ότι μόνο αν μείνουμε απόλυτα ικανοποιημένοι θα τους βαθμολογήσουμε με 5 αστέρια.

Το Taxibeat εκτός από την Ελλάδα (κυρίως Αθήνα και Θεσσαλονίκη και με περιορισμένη κάλυψη σε Πάτρα, Μυτιλήνη, Χανιά, Ηράκλειο) διατίθεται και στη Βραζιλία, το Μεξικό, τη Γαλλία, τη Νορβηγία και τη Ρουμανία. Για τη διευκόλυνσή σου, το Taxibeat είναι συνδεδεμένο με τα Google Maps και το Foursquare.

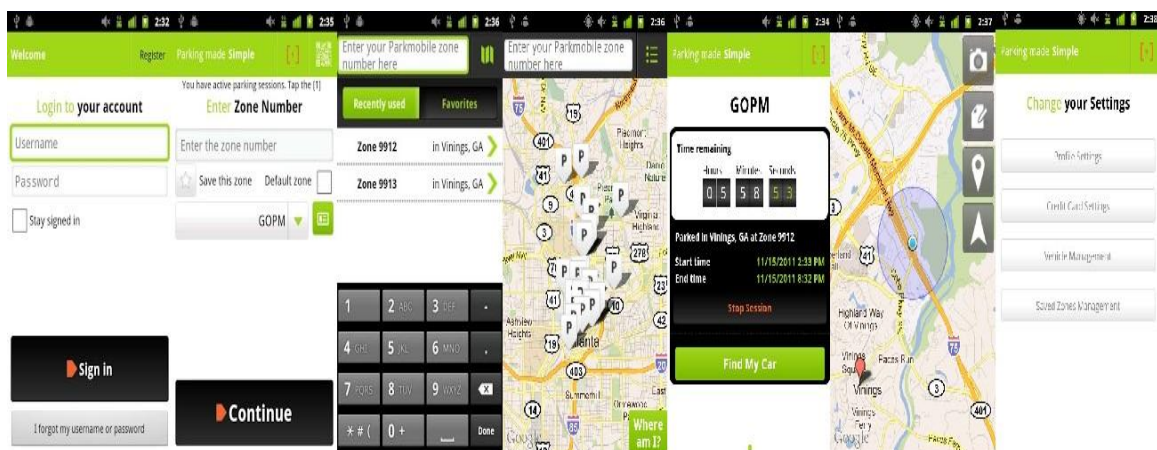


Εικόνα 3: Taxibeat App

Υπάρχει αντίστοιχη εφαρμογή για τους οδηγούς ταξί που λέγεται: Taxibeat Driver

3)  Parkmobile

Η εφαρμογή αυτή παρέχει ένα νέο και καλύτερο τρόπο να πληρώνεις το parking χρησιμοποιώντας το κινητό τηλέφωνο. Η συναλλαγή ολοκληρώνεται πολύ γρήγορα. Άμα οδηγήσουμε διαφορετικό αμάξι μπορούμε να αλλάξουμε τον αριθμό κυκλοφορίας πριν το ενεργοποιήσουμε. Μπορούμε να ζητήσουμε να έρθει ειδοποίηση μέσω μηνύματος στο κινητό δεκαπέντε λεπτά πριν λήξει το χρονικό όριο πληρωμής. Τα δεδομένα προστατεύονται και έχουμε την δυνατότητα να αλλάξουμε τις ρυθμίσεις, να δούμε το ιστορικό, να εκτυπώσουμε μια αναφορά και να εξάγουμε πληροφορίες για επιστροφές εξόδων. Η εφαρμογή χρησιμοποιείται σε ιδιωτικά και δημόσια parking σε οδούς, αεροδρόμια, στάδιο κτλ



Εικόνα 4: Parkmobile App

Παρόμοιες εφαρμογές: Parkmobile Parking, Pocket Market Meter Free

4) BestParking

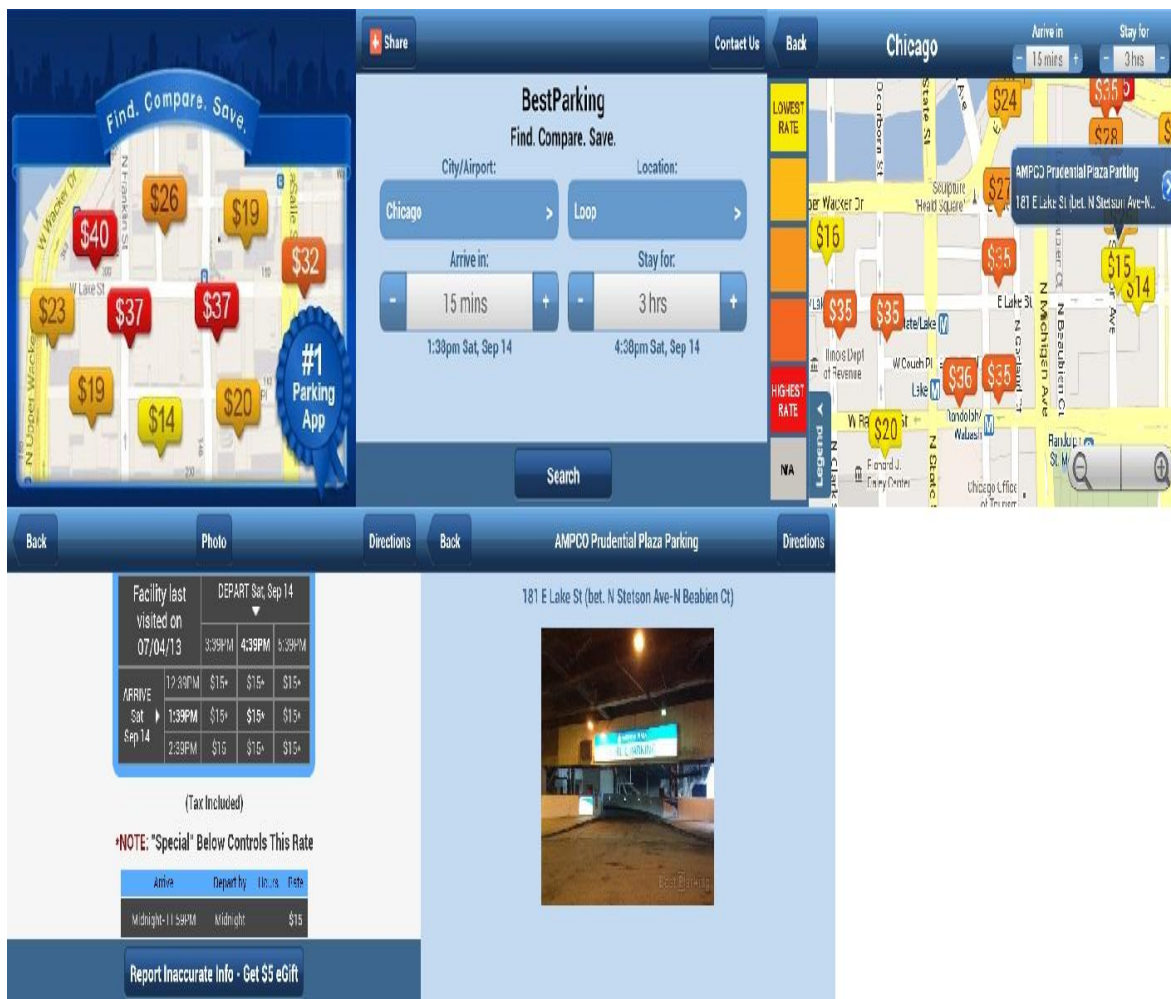
Το BestParking είναι μια μηχανή αναζήτησης στάθμευσης που καθοδηγεί τους οδηγούς προς τους φθηνότερους και πιο βολικούς χώρους στάθμευσης σε 100 πόλεις και 115 αεροδρόμια σε όλη τη Βόρεια Αμερική. Περιλαμβάνει πολλά χαρακτηριστικά των χώρων στάθμευσης (π.χ., Ώρες Λειτουργίας, τηλέφωνο, Indoor / Outdoor, Μετρητά / Πιστωτικές, Self Park / Valet, κ.λπ.).

Το BestParking είναι η μόνη εφαρμογή εύρεσης στάθμευσης που διενεργεί συνεχείς αυτόματες ενημερώσεις όλων των συντελεστών στάθμευσης και περιοχών για να εξασφαλίσει το υψηλότερο επίπεδο ακρίβειας. Σε περίπτωση λανθασμένων οδηγιών δίνει αποζημίωση 5\$.

ΕΠΙΠΛΕΟΝ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ για τους Νεοϋορκέζους:

- Αποκλειστικές εκπτώτικα κουπόνια σε εκατοντάδες χώρους στάθμευσης.
- On-street κανονισμοί για το Μανχάταν και το Downtown Μπρούκλιν. Οι πινακίδες κάθε δρόμου αποκωδικοποιούνται και περιλαμβάνονται στη μηχανή υπολογισμού!

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη



Εικόνα 5: BestParking App

2.2 Αξιολόγηση των εφαρμογών με βάση τα χαρακτηριστικά τους

Ο παρακάτω πίνακας δείχνει τις εφαρμογές που αναλύσαμε νωρίτερα και την αξιολόγηση τους με βάση τα παρακάτω χαρακτηριστικά που αναφέραμε στο κεφάλαιο 1.2.2.

Πίνακας 1: Αξιολόγηση των εφαρμογών με βάση τα χαρακτηριστικά τους

Apps	Ευχρηστία	Platform	Συμβατότητα	Υπέρ	Κατά
<u>Easy Taxi</u>	Εύκολο, Ασφαλές, Βολικό, Δωρεάν	iOS, Android, Windows Phone	iPhone, iPad, iPod, Smartphones	Χρήση GPS, maps, διαθέσιμο 24/7	Bugs, χρειάζεται βελτίωση σταθερότητας και επίδοσης
<u>Taxibeat</u>	Εύκολο, Ασφαλές, Βολικό, Δωρεάν	iOS, Android	iPhone, iPad, iPod, Smartphones	Χρήση GPS, google maps, Foursquare, Βαθμολογία οδηγών ταξί	Bugs, η συνεχής χρήση του GPS μειώνει τη δραματικά την «ζωή» της μπαταρίας
<u>Parkmobil e</u>	Εύκολο, Ασφαλές, Δωρεάν	iOS, Android, Windows Phone	iPhone, iPad, iPod, Smartphones, Blackberry	Standard, Hybrid and Satellite map, μήνυμα υπενθύμισης	Bugs, Crash, μικρή γραμματοσειρά, κακή αντίθεση χρωμάτων – δυσκολία στην ανάγνωση
<u>BestParki ng</u>	Εύκολο, Ασφαλές, Βολικό	iOS, Android	iPhone, iPad, iPod, Smartphones	Αυξημένη ταχύτητα και σταθερότητα, βελτιωμένες οδηγίες για το αναβαθμισμένο υς χρήστες	Bugs, χρειάζεται βελτίωση στις οδηγίες

2.3 Αξιολόγηση των εφαρμογών με βάση τα παρακάτω κριτήρια

Ο παρακάτω πίνακας δείχνει τις εφαρμογές που αναλύσαμε νωρίτερα και την αξιολόγηση τους με βάση τα παρακάτω κριτήρια που αναφέραμε στο κεφάλαιο 2.2.3.

Πίνακας 2: Αξιολόγηση εφαρμογών βάση κριτηρίων

Εφαρμογές Κριτήρια	Easy Taxi	Taxibeat	Parkmobile	BestParking
Κάλυψη ανάγκης αγοράς	5	5	3	4
Αξιοποίηση Κρατικού Κεφαλαίου	1	1	2	1
Οφέλη/Αλλαγές	2	3	4	4
Ικανότητα Προώθησης	5	5	4	5
Βελτίωση Εφαρμογής	4	5	3	4
Ανταγωνισμός/Πλεονέκτημα	3	4	4	5
User Interface	3	4	4	5
Επεκτασιμότητα	4	5	4	5

2.4 Επίλογος

Στο κεφάλαιο αυτό παρουσιάστηκαν παραπλήσιες υπηρεσίες και η περιγραφή των λειτουργιών τους. Επίσης έγινε μια συγκριτική αξιολόγηση με βάση τα κριτήρια που αναλύθηκαν στο προηγούμενο κεφάλαιο. Στο επόμενο θα αναλύσουμε τις απαιτήσεις της δική μας διαδικτυακής υπηρεσίας.

ΚΕΦΑΛΑΙΟ 3: Ανάλυση Απαιτήσεων

Στο κεφάλαιο αυτό θα αναλύσουμε τις περιπτώσει χρήσεις, τις απαιτήσεις του συστήματος, των χρηστών και θα αναφέρουμε τις ειδικές απαιτήσεις και τα ειδικά χαρακτηριστικά καθώς και τους περιορισμούς του συστήματος.

3.1 Στόχοι και επιδιώξεις του έργου

Στόχος κατά την ανάπτυξη της εφαρμογής είναι να έχει τις ιδιότητες του «καλού λογισμικού». Αυτό σημαίνει πως η τελική του μορφή θα πρέπει να προσφέρει λειτουργικότητα και την απόδοση που έχει ζητηθεί από τον πελάτη. Επίσης θα πρέπει να είναι εύκολο στην συντήρηση, να είναι αξιόπιστο και εύχρηστο. Όσον αφορά την συντήρηση θα πρέπει να μπορεί να εξελίσσεται για να καλύπτει τις νέες ανάγκες χρηστών. Σχετικά με την αξιοπιστία οι χρήστες θα πρέπει να το εμπιστεύονται. Η αποδοτικότητα του θα εκφράζεται με την καλύτερη δυνατή χρήση των πόρων του συστήματος. Τέλος βασικότατο χαρακτηριστικό αποτελεί το πόσο ικανοποιημένοι αισθάνονται οι χρήστες όταν δουλεύουν με αυτό.

Για να επιτευχθεί αυτό θα πρέπει το λογισμικό που θα αναπτυχθεί να περάσει από όλες τις απαιτούμενες φάσεις ανάπτυξη λογισμικού. Σε αυτό το κεφάλαιο θα ασχοληθούμε με το πρώτο βήμα που περιλαμβάνει την ανάλυση του προβλήματος. Βασικός στόχος η προκαταρκτική συνεργασία όλων των ενδιαφερόμενων ομάδων και πλευρών που ασχολούνται με την υλοποίηση του έργου. Η συνεργασία μεταξύ τους πρέπει να είναι εποικοδομητική γιατί σε αυτήν στηρίζεται η υλοποίηση της διαδικασίας της μηχανικής των απαιτήσεων. Πρόκειται για την διαδικασία μέσα από την οποία ορίζονται οι υπηρεσίες που θα παρέχονται από το σύστημα και οι περιορισμοί πάνω στους οποίους θα λειτουργήσει βάση των απαιτήσεων του πελάτη. Οι απαιτήσεις βγαίνουν από την περιγραφή των υπηρεσιών που θα παρέχονται και οι περιορισμοί δημιουργούνται κατά την διαδικασία της μηχανικής των απαιτήσεων. Με τον όρο απαίτηση μπορεί να εννοείται η ανάπτυξη μιας υπηρεσίας του συστήματος από τον πελάτη, ένας περιορισμός του συστήματος ή ο τεχνικός προσδιορισμός ενός προβλήματος. Οι απαιτήσεις

διακρίνονται σε λειτουργικές και μη λειτουργικές. Οι λειτουργικές περιγράφουν μια αλληλεπίδραση ανάμεσα στο σύστημα και το περιβάλλον του, τον τρόπο συμπεριφοράς του συστήματος, όταν δέχεται συγκεκριμένα ερεθίσματα, περιλαμβάνουν τον ορισμό των υπηρεσιών που θα πρέπει να υποστηρίζει το σύστημα και πως θα πρέπει να αντιδρά το σύστημα όταν θα δέχεται πληροφορίες. Οι μη λειτουργικές περιγράφουν μια προδιαγραφή του συστήματος που περιορίζει τις επιλογές μας για την κατασκευή μιας λύσης στο πρόβλημα.

Για να θεωρηθεί ολοκληρωμένη η διαδικασία καταγραφής των απαιτήσεων θα πρέπει να έχουν καταγραφεί όλοι οι τύποι απαιτήσεων οι οποίοι περιλαμβάνουν τα εξής:

- Απαιτήσεις χρήστη. Περιλαμβάνουν τις υπηρεσίες που θα προσφέρει το σύστημα και τους περιορισμούς αυτού. Φτιάχνεται για τον πελάτη.
- Απαιτήσεις του συστήματος. Αποτελεί ένα καλά δομημένο έγγραφο το οποίο εξηγεί με μεγάλη λεπτομέρεια τις υπηρεσίες που θα παρέχει το σύστημα. Φτιάχνεται για τον πελάτη και τον ανάδοχο και χρησιμεύει για συμβόλαιο.
- Προσδιορισμός λογισμικού. Αποτελεί τεχνικό έγγραφο με αναλυτική περιγραφή του λογισμικού και χρησιμοποιείται για την σχεδίαση και την ανάπτυξη του συστήματος. [Lawrence Pfleeger, Shari, 2003, Η εξαγωγή των απαιτήσεων, Τεχνολογία Λογισμικού, Γιάννης Στάμελος, Σελίδες 210-296, Αθήνα, Εκδόσεις Κλειδάριθμος]

3.2 Απαιτήσεις ηλεκτρονικής πλατφόρμας (συστήματος)

Έχει ήδη αναφερθεί πως η πλατφόρμα μέσω της οποίας υλοποιείται η πτυχιακή είναι το Android. Εκτενέστερες πληροφορίες για Android δίνονται στο πέμπτο κεφάλαιο καθώς και μια γενική εικόνα για τους λόγους επιλογής του σχετικούς με τα πλεονεκτήματα του Android στην αγορά. Όσον αφορά την εφαρμογή, το Android πληρεί τις προϋποθέσεις που πρέπει να ικανοποιεί μια πλατφόρμα έρευνας parking. Οι απαιτήσεις μπορεί να συνοψιστούν ως εξής:

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

- Να υποστηρίζει σύνδεση με βάση δεδομένων, έτσι ώστε ο χρήστης να μπορεί να κάνει αναζήτηση για parking και να έχει πρόσβαση στις πληροφορίες του.
- Να κάνει έλεγχο για τα δεδομένα (συγκεκριμένα την διεύθυνση) που εισάγει ο χρήστης στο πεδίο αναζήτησης parking - αν είναι έγκυρα ή όχι.
- Να εμφανίζει το χάρτη της Google με markers για όλα τα parking που υπάρχουν στην Θεσσαλονίκη.
- Να υποστηρίζει την λειτουργία εύρεσης της τωρινής τοποθεσίας του χρήστη της Google. Με αυτόν τον τρόπο ο χρήστης μπορεί να βρει τα parking που βρίσκουμε κοντά του.
- Να ελέγχει αν η βάση δεδομένων είναι εγκατεστημένη στην εκάστοτε συσκευή κι αν όχι, να την εγκαθιστά.
- Να δείχνει στον χάρτη της Google οδηγίες για το ποια κατεύθυνση να πάρει ο χρήστης για να πάει από το μέρος που βρίσκεται στο parking που επέλεξε.
- Πρέπει να υπάρχει η δυνατότητα σύνδεσης στο διαδίκτυο εφόσον χρησιμοποιείται η τεχνολογία JSON.
- Το περιβάλλον να είναι προσβάσιμο από όλες τις συσκευές Android και να υποστηρίζεται σε πολλές εκδόσεις του Android όχι μόνο στην πιο πρόσφατη (δες κεφάλαιο 2).
- Να έχει φιλικό περιβάλλον για τον χρήστη.
- Να έχει την δυνατότητα αναβάθμισης της βάσης δεδομένων σε περίπτωση αλλαγής ή προσθήκης νέων δεδομένων.

3.3 Περιορισμοί και ειδικά χαρακτηριστικά του συστήματος

Σε αυτήν την κατηγορία των απαιτήσεων τοποθετούμε εκείνες τις απαιτήσεις που περιγράφουν περιορισμούς του συστήματος οι οποίοι περιορίζουν τις δυνατότητες επιλογών για την υλοποίηση της λύσης του προβλήματος. Οι απαιτήσεις αυτές διαμορφώνονται ως εξής:

3.3.1 Βάση Δεδομένων

Η βάση δεδομένων για την εφαρμογή, όπως αναφέραμε, περιλαμβάνει τις πληροφορίες για κάθε parking. Η αναζήτηση των parking γίνεται με βάση την διεύθυνση που πληκτρολογεί ο χρήστης και για να ψάξει στην βάση δεδομένων θα χρησιμοποιήσει μια εντολή της SQLite. Η SQLite όμως θεωρείται case sensitive, ακόμα κι αν ο χρήστης πληκτρολογήσει σωστά την διεύθυνση μπορεί να μην εμφανίσει τα αποτελέσματα αν η διεύθυνση δεν είναι γραμμένη ακριβώς όπως είναι στην βάση δεδομένων (π.χ. ο χρήστης πληκτρολογεί: κρήτης 18, ενώ στη βάση δεδομένων έχουμε: Κρήτης 18. Δεν θα αναγνωρίσει την διαφορά πεζών - κεφαλαίων, ούτε τον τονισμό) Για αυτό θα πρέπει να λάβουμε υπόψη μας και να τροποποιήσουμε την εντολή της SQLite και/ ή την βάση δεδομένων.

3.3.2 Διάθεση πλατφόρμας σε διαφορετικές γλώσσες

Η αρχική υλοποίηση της εφαρμογής γίνεται συνήθως σε μια ή δυο γλώσσες (στην συγκεκριμένη περίπτωση ελληνικά & αγγλικά) ωστόσο μπορεί να διαθέτει έτοιμες εκδόσεις και άλλων γλωσσών ή να μπορεί να τις υποστηρίξει εύκολα με απλή εισαγωγή των απαραίτητων αρχείων.

3.3.3 Εύρεση τοποθεσίας

Με την χρήση της βιβλιοθήκης Google Play Services Lib μπορούμε να βρίσκουμε την τοποθεσία και να πλοηγηθούμε στους χάρτες ή να δούμε τις οδηγίες πάνω στον χάρτη για την διαδρομή (αναχώρηση-προορισμός). Για να επιτευθεί αυτό πρέπει να είναι εφικτή η σύνδεση με το Google Services.

3.4 Απαιτήσεις χρηστών

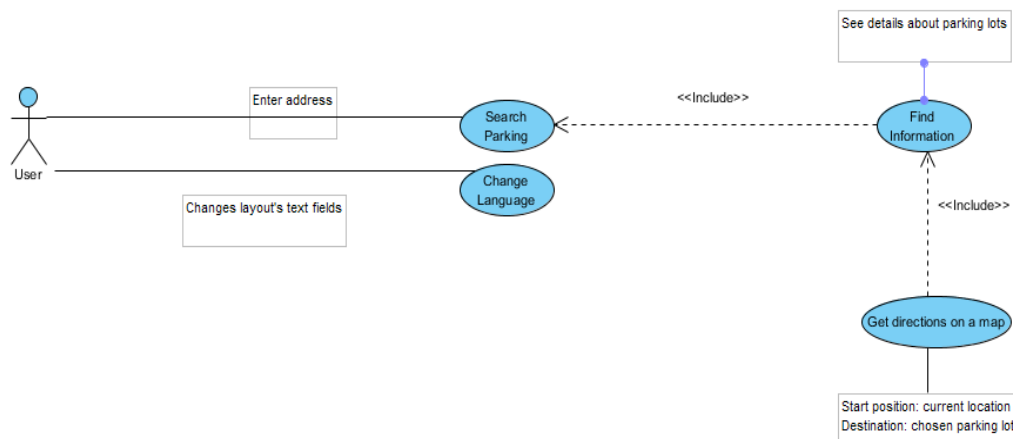
3.4.1 Ορισμός απαιτήσεων

Ο κυριότερος στόχος της εφαρμογής είναι να μπορεί ο χρήστης να εντοπίσει τα parking που υπάρχουν και να διαλέξει ανάλογα με τις πληροφορίες που υπάρχουν για το καθένα αυτό που τον συμφέρει (τοποθεσία, τιμή, ωράριο κλπ). Για να γίνει αυτό πρέπει πρώτα να κάνει αναζήτηση η οποία επιτυγχάνεται με δύο τρόπους: ο χρήστης πληκτρολογεί την διεύθυνση και περιμένει να δει αν υπάρχουν parking στην συγκεκριμένη οδό ή ελέγχει μέσω της Google Play Services Lib για parking στην γύρω περιοχή.

Αν υπάρχουν περισσότερα από ένα αποτελέσματα θα διαλέξει ένα από αυτά που θα τον μεταφέρει στις πληροφορίες υπάρχουν διαθέσιμες γι' αυτό. Αν δεν μείνει ευχαριστημένος επιστρέφει στον χάρτη και βλέπει για το επόμενο. Όταν διαλέξει ένα και δεν ξέρει τι διαδρομή να ακολουθήσει μπορεί να δει τις οδηγίες στον χάρτη.

Επιπρόσθετα, ο κάθε χρήστης διαλέγει την γλώσσα στην οποία θέλει να διαχειρίζεται την εφαρμογή. Η λειτουργία αυτή προστέθηκε με στόχο να συμπεριλάβει στους χρήστες τόσο τους έλληνες πολίτες όσο και τους ξένους που επισκέπτονται την πόλη.

Οι αλληλεπιδράσεις αυτές του χρήστη και της εφαρμογής φαίνονται στο ακόλουθο διάγραμμα χρηστών - στόχων:



Εικόνα 6: Διάγραμμα περιπτώσεων χρήσης

3.4.2 Βάση δεδομένων

Η αναζήτηση και εύρεση parking γίνεται με βάση τα δεδομένα που έχει η εφαρμογή, η οποία πρέπει με κάποιο τρόπο να γνωρίζει τις πληροφορίες αυτές και να τις εμφανίζει κατάλληλα. Γι' αυτό λοιπόν μέσα στον φάκελο assets που αποτελεί έναν από τους φακέλους της εφαρμογής υπάρχει το αρχείο τύπου .sqlite . Το αρχείο αναπαριστά τα parking ως αντικείμενα σε πίνακα. Κάθε parking διαθέτει τα πεδία Region, Name, Address, PhoneNumber, ExtraInfo, Type, HoursPrice, Features, CoordinateA, CoordinateB, Address2, Address3 τα οποία είναι με τη σειρά η περιοχή του parking, το όνομα του parking, η διεύθυνσή του, το τηλέφωνο επικοινωνίας, οι επιπλέον πληροφορίες για το parking, ο τύπος του parking, το ωράριο και οι τιμές του, τα χαρακτηριστικά του, οι συντεταγμένες του και η διεύθυνσή του.

Όπως αναφέραμε νωρίτερα στους περιορισμούς, όταν ένας χρήστης ψάχνει parking με βάση την διεύθυνση πρέπει να την γράφει ακριβώς όπως είναι στην βάση δεδομένων αλλιώς δεν θα εμφανίσει τα αποτελέσματα. Για να λύσουμε αυτό το πρόβλημα προσθέσαμε άλλα δύο πεδία τα οποία περιέχουν την ίδια διεύθυνση γραμμένη διαφορετικά (π.χ. Address2: ΚΡΗΤΗΣ 18, Address3: ΚΡΗΤΗΣ 18) ώστε να αποφευχθεί το μπέρδεμα κεφαλαίων - πεζών χαρακτήρων και ο τονισμός των γραμμάτων. Η βάση δεδομένων

δημιουργήθηκε στο εργαλείο του Firefox το SQLite Manager, η προεπιλεγμένη κωδικοποίηση είναι το UTF-8 οπότε δεν χρειάστηκε περαιτέρω επεξεργασία.

3.5 Ειδικές απαιτήσεις

3.5.1 Διεπαφές χρηστών εφαρμογής

Όταν ο χρήστης χρησιμοποιεί την εφαρμογή έχει την δυνατότητα να:

- 1) Αναζητήσει parking. Ο χρήστης πληκτρολογεί την διεύθυνση στην οποία θέλει να ελέγξει αν υπάρχει parking. Δίνεται η δυνατότητα στον χρήστη να κάνει άμεσα νέα αναζήτηση σε περίπτωση που δεν υπάρχουν αποτελέσματα.
- 2) Αλλάξει τη γλώσσα. Η προεπιλεγμένη γλώσσα είναι τα αγγλικά, ο χρήστης μπορεί να αλλάξει τη γλώσσα σε ελληνικά και αντίστροφα.
- 3) Βρει στον χάρτη την τωρινή τοποθεσία. Έτσι βλέπει ποια parking υπάρχουν κοντά του και να επιλέξει ένα.
- 4) Δει στον χάρτη όλα τα parking της πόλης και επιλέγοντας ένα ανοίγει ένα παράθυρο που δείχνει το όνομα και την διεύθυνση του parking.
- 5) Διαβάσει τις πληροφορίες που υπάρχουν για κάθε parking πατώντας στο παράθυρο που αναφέραμε στο (4). Οι πληροφορίες βρίσκονται στα πεδία της βάσης δεδομένων (κεφάλαιο 4.4.2), παρόλα αυτά δεν υπάρχουν δεδομένα σε όλα τα πεδία για κάθε parking. Οπότε στην οθόνη εμφανίζονται μόνο εκείνα που έχουν δεδομένα.
- 6) Βρει οδηγίες για το πώς να πάει στο parking που θέλει από το σημείο που βρίσκεται. Εφόσον έχει ανοίξει τις πληροφορίες ενός parking έχει την δυνατότητα να επιλέξει να ανοίξει το χάρτη που θα δείχνει την διαδρομή.

3.5.2 Διεπαφές υλικού

Συσκευές εισόδου: οθόνης αφής - εικονικό πληκτρολόγιο

Συσκευές εξόδου: οθόνη

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

3.5.3 Διεπαφές λογισμικού

Απαιτείται χρήση της γλώσσας της Java για την ανάπτυξη του κώδικα και των αντίστοιχων βιβλιοθηκών που χρειάζονται android-support-v4, Google Play Services καθώς και του ανάλογου JDK (Java Development Kit) στο πρόγραμμα υλοποίησης του προϊόντος.

3.5.4 Διεπαφές επικοινωνιών

Η εφαρμογή επικοινωνεί με το Google Services για να μπορέσει να χρησιμοποιήσει τα εργαλεία της Google, π.χ. Google Maps. Χρειάζεται να συνδεθούμε στο Google Play Developer Console για να πάρουμε API key ώστε να μπορέσει η εφαρμογή να συνδεθεί στον χάρτη της Google.

3.6 Περιπτώσεις Χρήσης

Περίπτωση χρήσης:	ΠΧ01
Χειριστής:	Χρήστης
Περιγραφή:	Αναζήτηση parking
Προϋποθέσεις:	Ο χειριστής να γράφει στα ελληνικά
Κύρια ροή γεγονότων:	Ο χειριστής πληκτρολογεί την διεύθυνση στην οποία ψάχνει να βρει parking και πατά το κουμπί Χάρτης
Μετέπειτα κατάσταση:	Ανοίγει ο χάρτης κάνοντας σμίκρυνση στο parking που έψαχνε ο χειριστής
Εναλλακτικές ροές γεγονότων και εξαιρέσεις:	Σε περίπτωση που δεν υπάρχει αποτέλεσμα ή ο χειριστής δεν έγραψε σωστά την διεύθυνση εμφανίζεται αντίστοιχο μήνυμα

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

Περίπτωση χρήσης:	ΠΧ02
Χειριστής:	Χρήστης
Περιγραφή:	Αλλαγή γλώσσας
Προϋποθέσεις:	Ο χειριστής να βρίσκεται στο αρχικό παράθυρο
Κύρια ροή γεγονότων:	Ο χειριστής επιλέγει ένα από τα δύο κουμπιά ανάλογα με το ποια γλώσσα θέλει ελληνικά ή αγγλικά
Μετέπειτα κατάσταση:	Όλα τα κείμενα στο layout αλλάζουν
Εναλλακτικές ροές γεγονότων και εξαιρέσεις:	Τα δεδομένα της βάσης δεδομένων δεν αλλάζουν από ελληνικά σε αγγλικά ή αντίστροφα

Περίπτωση χρήσης:	ΠΧ03
Χειριστής:	Χρήστης
Περιγραφή:	Εύρεση τωρινής τοποθεσίας
Προϋποθέσεις:	Ο χειριστής να έχει ανοίξει τον χάρτη
Κύρια ροή γεγονότων:	Ο χειριστής επιλέγει το κουμπί του Google Maps και εκείνο εμφανίζει στον χάρτη το σημείο που βρίσκεται
Μετέπειτα κατάσταση:	Εμφανίζονται τα parking που υπάρχουν σε κοντινή απόσταση

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

Εναλλακτικές ροές γεγονότων και εξαιρέσεις:	Σε περίπτωση που δεν υπάρχει σύνδεση στο διαδίκτυο δεν εμφανίζει την τοποθεσία
---------------------------------------------	--------------------------------------------------------------------------------

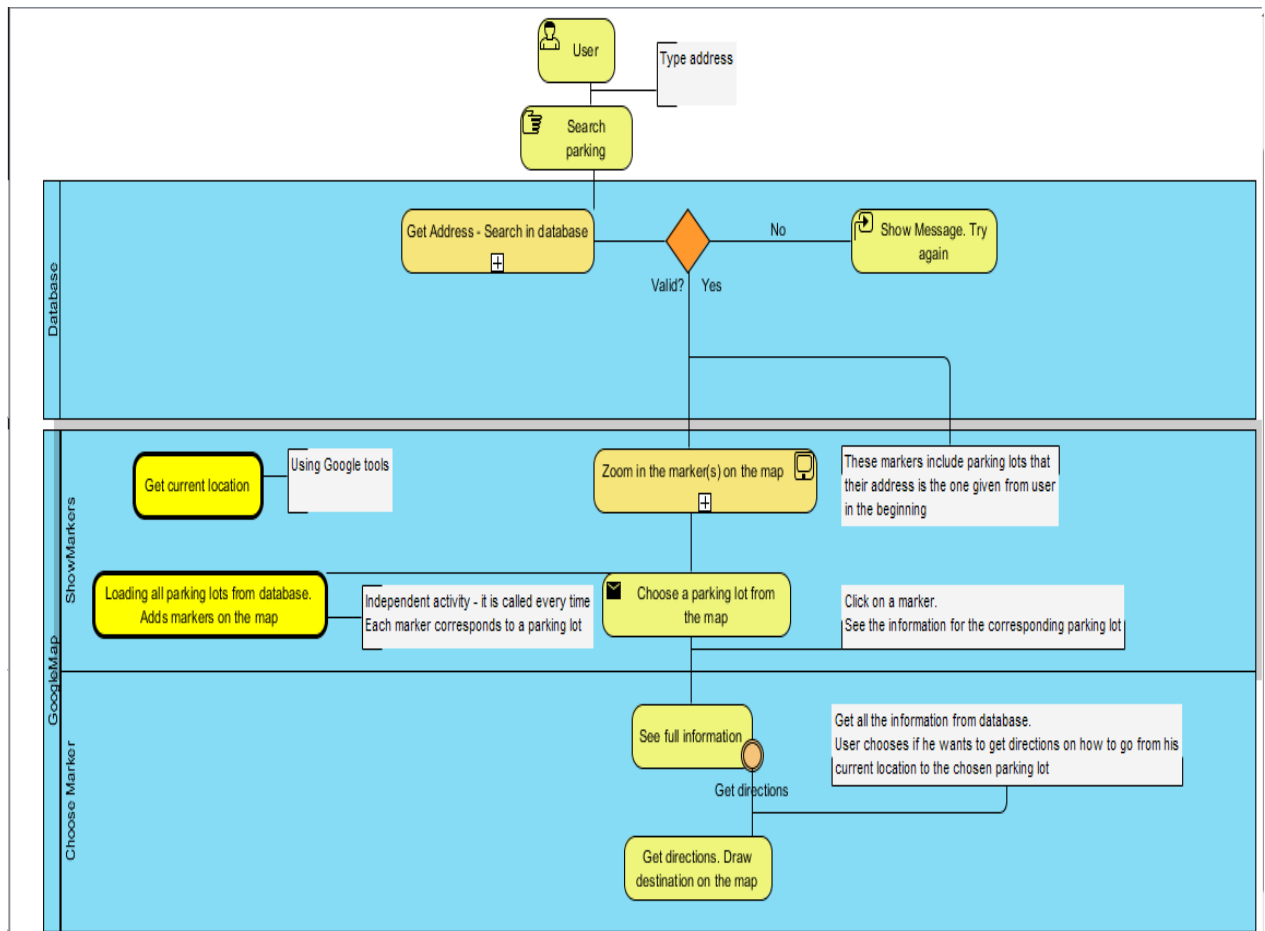
Περίπτωση χρήσης:	ΠΧ04
Χειριστής:	Χρήστης
Περιγραφή:	Επιλογή ενός parking
Προϋποθέσεις:	Ο χειριστής να έχει ανοίξει τον χάρτη
Κύρια ροή γεγονότων:	Ο χειριστής πατάει πάνω σε έναν marker του χάρτη. Κάθε marker αντιστοιχεί σε ένα parking
Μετάπειτα κατάσταση:	Εμφανίζεται ένα pop up παράθυρο με το όνομα και την διεύθυνση του επιλεγμένου parking
Εναλλακτικές ροές γεγονότων και εξαιρέσεις:	Καμία

Περίπτωση χρήσης:	ΠΧ05
Χειριστής:	Χρήστης
Περιγραφή:	Άνοιγμα των πληροφοριών του parking
Προϋποθέσεις:	Ο χειριστής να έχει επιλέξει ένα parking από τον χάρτη

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

Κύρια ροή γεγονότων:	Αφού έχει ανοίξει το pop up παράθυρο ενός marker ο χειριστής πατάει το παράθυρο αυτό και ανοίγει το παράθυρο με τις πληροφορίες για το αντίστοιχο parking
Μετάπειτα κατάσταση:	Εμφανίζονται μόνο τα πεδία που έχουν δεδομένα
Εναλλακτικές ροές γεγονότων και εξαιρέσεις:	Καμία

Περίπτωση χρήσης:	ΠΧ06
Χειριστής:	Χρήστης
Περιγραφή:	Οδηγίες για το πώς να πας σε ένα parking
Προϋποθέσεις:	Ο χειριστής να έχει ανοίξει το παράθυρο των πληροφοριών
Κύρια ροή γεγονότων:	Ο χειριστής πατάει το κουμπί Οδηγίες και ανοίγει ο χάρτης που δείχνει την διαδρομή στο χάρτη
Μετάπειτα κατάσταση:	Δείχνει ένα marker σε πράσινο χρώμα για την τωρινή τοποθεσία και έναν κόκκινο marker για το parking - προορισμό. Η διαδρομή εμφανίζεται στον χάρτη με μια κόκκινη γραμμή.
Εναλλακτικές ροές γεγονότων και εξαιρέσεις:	Σε περίπτωση που δεν υπάρχει σύνδεση στο διαδίκτυο δεν εμφανίζει την τωρινή τοποθεσία του χρήστη



Εικόνα 7: Διάγραμμα Προόδου Επιχειρήσεων – Business Program Diagram

3.7 Επίλογος

Στο κεφάλαιο αυτό αναλύθηκαν οι απαιτήσεις της υπηρεσίας για το σύστημα και τους χρήστες. Έγινε αναφορά στους περιορισμούς και τις ειδικές απαιτήσεις καθώς και τα χαρακτηριστικά, ενώ έγινε πλήρης περιγραφή των περιπτώσεων χρήσης της υπηρεσίας. Στο επόμενο κεφάλαιο θα αναφερθούμε στο περιβάλλον ανάπτυξης και στις τεχνολογίες που χρησιμοποιήθηκαν στην υπηρεσία.

ΚΕΦΑΛΑΙΟ 4: Τεχνολογικό Περιβάλλον Ανάπτυξης

Το κεφάλαιο αυτό εξηγεί λεπτομερώς τις τεχνολογίες που χρησιμοποιήθηκαν στην εφαρμογή. Περιγράφεται το λειτουργικό σύστημα Android και οι λόγοι που το έγινε τόσο δημοφιλές. Έπειτα γίνεται μια ιστορική αναδρομή στις εκδόσεις του. Αναφέρονται επίσης η βιβλιοθήκη Google Play Services η οποία μας βοηθάει να χρησιμοποιούμε όλες τις λειτουργίες της Google και το εργαλείο Eclipse που χρησιμεύει στην ανάπτυξη εφαρμογών Android και όχι μόνο.

4.1 Περιβάλλον Eclipse



Εικόνα 8: Λογότυπο του Eclipse

4.1.1 Eclipse

Το Eclipse είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (integrated development environment - IDE). Περιέχει ένα βασικό χώρο εργασίας (workspace) και ένα επεκτάσιμο plug-in σύστημα για την προσαρμογή του περιβάλλοντος. Γραμμένο κυρίως στην Java, το Eclipse μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών. Με την βοήθεια δοαφόρων plug-ins, η χρήση του Eclipse (στην ανάπτυξη εφαρμογών) μπορεί να επεκταθεί και σε άλλες γλώσσες προγραμματισμού: Ada, ABAP, C, C++, COBOL, Fortran, Haskell, JavaScript, Lasso, Natural, Perl, PHP, Prolog,

Python, R, Ruby (συμπεριλαμβανομένου και του Ruby on Rails framework), Scala, Clojure, Groovy, Scheme, and Erlang. Επίσης υποστηρίζεται από τα παρακάτω λειτουργικά συστήματα: linux, Mac OS X, Solaris, Windows και είναι διαθέσιμο σε πολλές γλώσσες. Τα περιβάλλοντα ανάπτυξης περιέχουν τα Eclipse Java development tools (JDT) για Java και Scala, Eclipse CDT for C/C++ και Eclipse PDT για PHP μεταξύ άλλων. [Wikipedia, 2014, Eclipse]

Η αρχική βάση κώδικα προήλθε από την BM VisualAge. Το Eclipse software development kit (SDK) περιλαμβάνει τα εργαλεία ανάπτυξης για Java και προορίζεται για τους προγραμματιστές της Java. Οι χρήστες μπορούν να γράψουν τα δικά τους plug-ins και να συμβάλλουν στη επέκταση των ικανοτήτων του Eclipse. Κυκλοφόρησε σύμφωνα με τους όρους του Eclipse Public License, το Eclipse SDK είναι ένα ελεύθερο και ανοικτού κώδικα λογισμικό (αν και δεν είναι συμβατό με την GNU General Public License). Ήταν ένα από τα πρώτα IDEs που "έτρεξε" κάτω από GNU Classpath και "τρέχει" χωρίς προβλήματα κάτω από IcedTea. [Wikipedia, 2014, Eclipse]

4.1.2 Eclipse ADT (Android Development Tools)

Το ADT είναι ένα plug-in για το Eclipse IDE που έχει σχεδιαστεί για να μας δώσει ένα ισχυρό, ολοκληρωμένο περιβάλλον στο οποίο θα δημιουργηθούν εφαρμογές Android. Το ADT επεκτείνει τις δυνατότητες του Eclipse για να μπορούμε να ρυθμίσουμε γρήγορα νέες έργα για Android, να δημιουργήσουμε μια UI (user interface) εφαρμογή, να προσθέσουμε πακέτα με βάση το Android Framework API, να διορθώσουμε (debug) τις εφαρμογές χρησιμοποιώντας τα εργαλεία του Android SDK καθώς και να εξαγάγουμε με ή χωρίς υπογραφή αρχεία τύπου .apk προκειμένου να τα διανεμήσουμε στην αγορά. Συνίσταται η ανάπτυξη εφαρμογών με το ADT του Eclipse και είναι ο γρηγορότερος τρόπος για να ξεκινήσει κάποιος. Με την καθοδηγούμενη εγκατάσταση που παρέχεται, καθώς και τα εργαλεία ολοκλήρωσης, τους προσαρμοσμένους XML editors και το παράθυρο εξόδου εντοπισμού σφαλμάτων, το ADT μας προσφέρει μια απίστευτη ώθηση στην ανάπτυξη Android εφαρμογών. [Android Developers, 2014, ADT Plugin]

4.2 Πλατφόρμα Android



Εικόνα 9: Λογότυπο Android

4.2.1 Λειτουργικό σύστημα Android

Το Android είναι το πρώτο δωρεάν, ολοκληρωμένο λειτουργικό σύστημα σχεδιασμό για συσκευές κινητής τηλεφωνίας, όπως smartphones. Αναπτύχθηκε στον πυρήνα του Linux αρχικά από την Android Inc. και αγοράστηκε από την Google το 2005. [Felker, 2011] Οι προγραμματιστές του λογισμικού συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, διαχειρίζονται την συσκευή μέσω βιβλιοθηκών λογισμικού που ανέπτυξε η Google. [Wikipedia,2009, Android]

Η πλατφόρμα του Android δημοσιεύτηκε για πρώτη φορά στις 5 Νοεμβρίου 2007. Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα Android υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού. Το λογότυπο του Android σχεδιάστηκε από την γραφίστρια Irina Block. [Wikipedia,2009, Android]

Η πλατφόρμα παρέχει ένα SDK (Software Development Kit) με χαρακτηριστικά και εργαλεία, έτσι ώστε να υπάρχει μια πλήρης πλατφόρμα για την ανάπτυξη εφαρμογών. [Darcey, Conder, 2012]

Ο άνθρωπος αλληλεπιδρά με το προεπιλεγμένο περιβάλλον του Android με την άμεση χειραγώγηση, χρησιμοποιώντας την οθόνη αφής, που αντιστοιχεί σε πραγματικές αντιδράσεις, όπως το πάτημα ενός κουμπιού, ο σύρσιμο του δακτύλου στην οθόνη, η απομάκρυνση των δακτύλων για μεγέθυνση ή το πλησίασμα για σμίκρυνση καθώς και το εικονικό πληκτρολόγιο.

Το hardware περιέχει αισθητήρες, όπως επιταχυνσιόμετρα, γυροσκόπια και αισθητήρες εγγύτητας, οι οποίοι χρησιμοποιούνται από ορισμένες εφαρμογές για πρόσθετες ενέργειες του χρήστη, για παράδειγμα, την ρύθμιση της οθόνης από κατακόρυφο (portrait) σε οριζόντιο (landscape) προσανατολισμό ανάλογα με τον προσανατολισμό της συσκευής ή επιτρέποντας στο χρήστη να κατευθύνει ένα όχημα σε ένα παιχνίδι αγώνων με την περιστροφή της συσκευής, προσομοιώνοντας έλεγχο ενός τιμονιού.

Οι Android οθόνες των συσκευών αποτελούνται από εικονίδια εφαρμογών και widgets. Τα εικονίδια ξεκινούν τη σχετική εφαρμογή, ενώ τα widgets εμφανίζουν άμεσα αυτόματες ενημερώσεις όπως το δελτίο καιρού ή την ειδοποίηση καινούριου email. Το interface της αρχικής οθόνης είναι σε μεγάλο βαθμό προσαρμόσιμο, επιτρέποντας στο χρήστη να προσαρμόσει την εμφάνιση και την αίσθηση της συσκευής με τις προτιμήσεις τους. [Wikipedia,2009, Android]

4.2.2 Γιατί επιλέγουμε Android;

Το Android είναι το πιο δημοφιλές λειτουργικό σύστημα κινητής τηλεφωνίας. Από το 2013, οι Android συσκευές πούλησαν περισσότερα από τα Windows, iOS και Mac OS συσκευές μαζί. Την ίδια χρονιά, δημοσιεύθηκαν πάνω από 1 εκατομμύριο εφαρμογές Android στο Google Play store και πάνω από 50 δισεκατομμύρια εφαρμογές «κατεβάστηκαν». Μια έρευνα που πραγματοποιήθηκε τον Απρίλιο - Μάιο του 2013 διαπίστωσε ότι το 71 τοις

εκατό των mobile προγραμματιστών αναπτύσσουν Android εφαρμογές. [Wikipedia,2009, Android]

Το Android είναι ένα έτοιμο, χαμηλού κόστους και προσαρμόσιμο λειτουργικό σύστημα για συσκευές υψηλής τεχνολογίας. Διαθέτει το δικό του market, που είναι προεγκατεστημένο σε όλες τις Android συσκευές, διευκολύνοντας τους νέους προγραμματιστές να εισέλθουν στην αγορά εργασίας με το να διαθέσουν τις εφαρμογές τους κατευθείαν στο Google Play store.

Όπως αναφέραμε προηγουμένως το λειτουργικό σύστημα Android είναι ανοιχτή πλατφόρμα, που σημαίνει ότι δεν είναι συνδεδεμένη με έναν κατασκευαστή υλικού και/ή έναν πάροχο. Αυτό του δίνει την δυνατότητα να κερδίσει μερίδιο της αγοράς γρήγορα. Ο πηγαίος κώδικας του Android είναι διαθέσιμος στο <http://source.android.com> για όποιον θέλει να το δει και / ή τροποποιήσει.

Το Android μπορεί να λειτουργήσει σε πολλές συσκευές με διαφορετικά μεγέθη οθόνης και ανάλυση. Πέραν του ότι είναι συμβατό, το Android έρχεται με τα εργαλεία που θα βοηθήσουν έναν προγραμματιστή να αναπτύξει συμβατές εφαρμογές. Η Google επιτρέπει σε εφαρμογές να τρέξουν μόνο σε συμβατές συσκευές. Εάν μια εφαρμογή απαιτεί μια κάμερα στο μπροστινό μέρος, για παράδειγμα, μόνο τηλέφωνα με κάμερα στο μπροστινό μέρος θα είναι σε θέση να δείτε την εφαρμογή στο Android Market. Αυτή η διάταξη είναι γνωστή ως ανίχνευση χαρακτηριστικό (feature detection).

Επίσης, το υλικό των συσκευών και τα APIs της πλατφόρμας προσφέρουν πολλές δυνατότητες. Μια από αυτές είναι ο συνδυασμός δύο ή περισσότερων υπηρεσιών και/ή βιβλιοθηκών για την δημιουργία μιας εφαρμογής. Για παράδειγμα, μπορούμε να χρησιμοποιήσουμε την κάμερα και την υπηρεσία τοποθεσία του Android για να βγάζουμε φωτογραφίες με την ακριβή τοποθεσία στην εικόνα. [Felker, 2011]

4.2.3 Ιστορική Αναδρομή

Η Android Inc. ιδρύθηκε από τον Andy Rubin (συνιδρυτή της Danger), Rich Miner (συνιδρυτή της Wildfire Communications, Inc.), Nick Sears (πρώην αντιπρόεδρο του T-Mobile) και τον Chris White (επικεφαλής σχεδιασμού και ανάπτυξης διεπιφανειών στο WebTV) για να αναπτύξουν, σύμφωνα με τον Rubin, «τις εξυπνότερες κινητές συσκευές που θα είναι πιο ενήμερες για την τοποθεσία του χρήστη και τις προτιμήσεις του». Η Google αγόρασε την εταιρεία στις 17 Αυγούστου του 2005, προσλαμβάνοντας τους Rubin, Miner, White ως υπαλλήλους. Πολλοί πίστεψαν τότε ότι η Google σκοπεύει να εισέλθει στην αγορά της κινητής τηλεφωνίας. Στις 5 Νοεμβρίου του 2007 δημιουργήθηκε η Open Handset Alliance, μια κοινοπραξία 48 τηλεπικοινωνιακών εταιριών (συμπεριλαμβανομένων της Google, HTC, Sony, Samsung), εταιριών λογισμικού καθώς και κατασκευής υλικού, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές κινητής τηλεφωνίας. [Wikipedia, 2009, Android]

Το πρώτο εμπορικά διαθέσιμο smartphone με λειτουργικό Android ήταν το HTC Dream που κυκλοφόρησε στις 22 Οκτωβρίου του 2008. Αρχικά, πολλά χαρακτηριστικά που τώρα είναι αναγκαία δεν υπήρχαν · το εικονικό πληκτρολόγιο, η δυνατότητα πολλαπλών σημείων αφής. Παρόλα αυτά υπάρχουν κάποια χαρακτηριστικά που εξελίχτηκαν και τα συναντάμε μέχρι την σημερινή 4.0 έκδοση.

Αρχικά, ένα αναγνωριστικό είναι η γραμμή ειδοποιήσεων, που περιέχει χρήσιμες πληροφορίες και επεκτείνεται όταν ο χρήστης τη σύρει με το δάχτυλό του προς τα κάτω εμφανίζοντας έτσι όλες τις ενημερώσεις σε μορφή λίστας. Αξίζει να αναφέρουμε ότι η Google πρόσθεσε από την αρχή widgets αλλά με μια σημαντική διαφορά · οι προγραμματιστές δεν μπορούσαν να δημιουργήσουν τα δικά τους.

Η Gmail χρησιμοποιούσε ήδη τα IMAP και POP πρωτοκόλλα για ενσωμάτωση με πελάτες email σε mobile συσκευές. Ωστόσο αυτά τα δυο

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

πρωτόκολλα δεν υποστήριζαν πλήρες τις λειτουργίες του Gmail. Το Android το έφτιαξε αυτό, παρέχοντας την καλύτερη εμπειρία Gmail στην αγορά.

Η Google είχε εγκαταστήσει εξαρχής το Android Market στις συσκευές καθώς και μια φτωχή λίστα εφαρμογών αλλά δεν περιείχε εφαρμογές επί πληρωμή.

Η πρώτη αναβάθμιση, το Android 1.1, έγινε τρεις μήνες αργότερα. Η νέα έκδοση δεν ήταν επαναστατική αλλά διόρθωσε πολλά bugs και κυρίως απέδειξε την ικανότητα του Android στο να κάνει την αναβάθμιση στην εκάστοτε έκδοση εύκολη διαδικασία για τους χρήστες. Ήταν η πρώτη μεγάλη πλατφόρμα που το πέτυχε. [The Verge, 2011]

Η δεύτερη αναβάθμιση, έκδοση Cupcake 1.5, περιείχε σημαντικές βελτιώσεις και πρόσθετα χαρακτηριστικά: αναβάθμιση του gmail, υποστήριξη των youtube και Picasa, ενσωμάτωση του Google Talk σε διάφορα μέρη όπως τις επαφές, τα μηνύματα και το gmail. Έδωσε την δυνατότητα στους προγραμματιστές να προσθέτουν widgets στις εφαρμογές τους, υπήρχε επέκταση η κάμερα για λήψη video, έκανε το πρώτο βήμα για την υλοποίηση του copy - paste και ενσωμάτωσε το εικονικό πληκτρολόγιο στην οθόνη, καταργώντας τα πλήκτρα, εισάγοντας στην αγορά το πρώτο κινητό με οθόνη αφής.

Η έκδοση 1.6 Donut είχε πιο σημαντικές αλλαγές από ότι οι προηγούμενες. Η κυριότερη ήταν η υποστήριξη του CDMA, ακολουθούμενη από την δυνατότητα του λειτουργικού να τρέχει σε συσκευές με διαφορετικές αναλύσεις οθόνης. Επίσης, η 1.6 εισήγαγε την “καθολική αναζήτηση”. Η universal search επιτυγχάνεται μέσα από το Quick Search Box, το οποίο τοποθετήθηκε στην αρχική οθόνη αντικαθιστώντας την διαδικασία αναζήτησης μέσω Chrome στο google.com. Με τις βελτιώσεις του Donut ο χρήστης μπορεί να ψάξει σε μια ποικιλία τοπικού περιεχομένου (εφαρμογές, επαφές κλπ.) αλλά και στο διαδίκτυο ταυτόχρονα από ένα μόνο “κουτί”. Επίσης,

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

αναδιαμορφώνεται και το Android Market όχι μόνο στη σχεδίαση του αλλά προστίθενται οι λίστες με τις top δωρεάν εφαρμογές και τις top εφαρμογές επί πληρωμή. [The Verge, 2011]

Ένα χρόνο αργότερα κυκλοφορεί η νέα έκδοση 2.0/2.1, γνωστή και ως Eclair, αποκλειστικά στο Motorola Droid. Το Google ήταν το μοναδικό που περιείχε υποστήριξη πολλαπλών λογαριασμών Google στην ίδια συσκευή με πρόσβαση σε email και επαφές για τον καθένα και συγχρονισμό με τις εφαρμογές. Επιπλέον, ήταν το μοναδικό κινητό με δυνατότητες πλοήγησης μέσω του Google Maps Navigation, ενώ με την εγκατάσταση του Quick Contact που επιτρέπει στον χρήστη να επικοινωνεί με τις επαφές του με ποικίλους τρόπους (π.χ. email, μήνυμα, κλήση κλπ). Υπήρξαν βελτιώσεις και στο πληκτρολόγιο καθώς άρχισε η λειτουργία του multitouch, εμφανίστηκαν τα πρώτα live wallpapers και ενσωματώθηκε η λειτουργία του speech-to-text ώστε να γράφεται κείμενο που ο χρήστης έχει πει. [The Verge, 2011]

Από την 2.1 έκδοση η Google εισήγαγε την σειρά κινητών Nexus, οπότε όταν το 2010 κυκλοφόρησε η 2.2 Froyo, το Nexus One ήταν το πρώτο που αναβαθμίστηκε. Στο Froyo έχει βελτιωθεί σημαντικά η ταχύτητα και η απόδοση, ενώ οι συσκευές ξεκίνησαν να λειτουργούν και ως hot spots στα οποία συνδέονταν άλλες συσκευές στο διαδίκτυο. Προστέθηκε επίσης και η επιλογή κωδικού στο κλείδωμα της οθόνης εκτός από το μοτίβο. [The Verge, 2011]

Με την έκδοση Gingerbread 2.3 βελτιώθηκαν τα εργαλεία διαχείρισης της μπαταρίας και των εφαρμογών, ενώ στις συσκευές προστέθηκε και μπροστινή κάμερα. Σημαντική προσθήκη ήταν και η δυνατότητα διαχείρισης του ήχου, των γραφικών και της μνήμης της συσκευής - υλοποίηση 3D γραφικών - η βελτίωση του πληκτρολογίου και ο καλύτερος έλεγχος του copy - paste. Για πρώτη φορά υποστηρίζεται το NFC (Near Field Communication) το οποίο

μπορεί να αλλάζει βασικές λειτουργίες κινητού, μέχρι και να χρησιμοποιεί το Google Wallet. [The Verge, 2011]

Η επόμενη αναβάθμιση, το Honeycomb, έκδοση 3.0 - 3.2, δεν αφορούσε αποκλειστικά τα smartphones. Η Google ήθελε να επεκταθεί και σε tablets χωρίς όμως να έχει την αναμενόμενη επιτυχία. Ωστόσο έγιναν κάποιες βελτιώσεις στο λογισμικό. Τα widgets εμφανίζουν τον χώρο που θα καταλαμβάνουν στην οθόνη, βελτιώνεται το multitasking, καταργούνται πλήρως τα φυσικά κουμπιά για τα Back, Home, Menu και Search, ενώ εμφανίζεται και η γραμμή ενεργειών σε κάποιες εφαρμογές. Οι προγραμματιστές που επιθυμούν να ενσωματώσουν την γραμμή ενεργειών μπορούν να χρησιμοποιήσουν το αντίστοιχο API που είναι διαθέσιμο. [The Verge, 2011]

Στην έκδοση Ice Cream Sandwich 4.0, το NFC προεκτείνεται σε Android Bean, το οποίο παρέχει την δυνατότητα ανταλλαγής αρχείων μεταξύ δύο συσκευών απλά ακουμπώντας η μια την άλλη. Επίσης προστίθεται νέα μέθοδος ξεκλειδώματος που γίνεται με την αναγνώριση προσώπου, καινούριο ημερολόγιο και mail εφαρμογές, καθώς και μια εφαρμογή που αναλύει τα δεδομένα χρήσης (Data usage analysis). [The Verge, 2011]

Η έκδοση 4.1 Jelly Bean δίνει έμφαση στα γραφικά. Απαλείφει μια σημαντική καθυστέρηση - που ήταν ορατή και στο scroll - ενισχύοντας τα γραφικά με αποτέλεσμα ο επανασχεδιασμός τους να γίνεται ανά 16 milliseconds. Επιπλέον, προστίθεται η δυνατότητα επέκτασης των ειδοποιήσεων του πίνακα ειδοποιήσεων, τα widgets γίνονται πιο ευέλικτα - αλλάζουν μέγεθος δυναμικά, αλλάζει η εμφάνιση του UI και εισάγεται το "προβλεπόμενο κείμενο" στο πληκτρολόγιο. Επίσης εμφανίζει τις λέξεις που ο χρήστης χρησιμοποιεί περισσότερο. [The Verge, 2011]

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

Στις μετέπειτα εκδόσεις Jelly Bean 4.2 και 4.3 έχουν γίνει αλλαγές στις εφαρμογές και στα widgets ρολογιών, υποστηρίζουν πολλαπλά προφίλ χρηστών, εμφανίζονται τα Photosheres, τα οποία είναι πανοραμικές φωτογραφίες 360 μοιρών. Βελτιώθηκε, επίσης, η προσβασιμότητα στις συσκευές ώστε να μπορούν να χρησιμοποιούνται εύκολα κι από τους ανάπηρους. Για παράδειγμα, η ενσωμάτωση της speech output και Gesture Mode πλοήγησης για τυφλούς χρήστες. Υποστηρίζεται επίσης η μηχανή γραφικών OpenGL ES 3.0. Επιπρόσθετα, ενσωματώθηκε το Swype στο πληκτρολόγιο (επιτρέπει στον χρήστη να γράφει με μια συνεχόμενη κίνηση) και υποστηρίζεται το Miracast το οποίο επιτρέπει στο κινητό να συνδεθεί ασύρματα με άλλη συσκευή. [The Verge, 2011]

Τέλος, η έκδοση 4.4 Kit Kat αναβάθμισε πολλά χαρακτηριστικά εφαρμογών, για παράδειγμα, έδωσε τη δυνατότητα για εφαρμογές πλήρους οθόνης (full screen). Υποστηρίζει το HDR+, βελτίωσε την παραγωγικότητα με την νέα έκδοση του QuickOffice και την ικανότητα εκτύπωσης σε όποιον εκτυπωτή συνδέεται με το Google Cloud Print. Το νέο QuickOffice έδωσε την δυνατότητα να χρησιμοποιούνται εφαρμογές όπως το Dropbox. Ενσωμάτωσε Emoji στο κανονικό πληκτρολόγιο, κυρίως όμως βελτίωσε σημαντικά την αναζήτηση με το εργαλείο **Google Now**. [The Verge, 2011]

Πέρα από τις αναβαθμίσεις του λειτουργικού συστήματος, κάθε εφαρμογή χωριστά αναβαθμίζεται σε τακτά χρονικά διαστήματα είτε αυτόματα είτε ζητώντας άδεια από τον χρήστη. Με αυτόν τον τρόπο αποφεύγεται η καθυστέρηση της χρονοβόρας παραμετροποίησης του λειτουργικού για τις συσκευές τους. [The Verge, 2011]



Εικόνα 10: Η εξέλιξη του Android

Platform Version	API Level	VERSION_CODE	Notes
Android 4.4	19	KITKAT	Platform Highlights
Android 4.3	18	JELLY_BEAN_MR2	Platform Highlights
Android 4.2, 4.2.2	17	JELLY_BEAN_MR1	Platform Highlights
Android 4.1, 4.1.1	16	JELLY_BEAN	Platform Highlights
Android 4.0.3, 4.0.4	15	ICE_CREAM_SANDWICH_MR1	Platform Highlights
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	Platform Highlights
Android 3.0.x	11	HONEYCOMB	Platform Highlights
Android 2.3.4 Android 2.3.3	10	GINGERBREAD_MR1	Platform Highlights
Android 2.3.2 Android 2.3.1 Android 2.3		GINGERBREAD	
Android 2.2.x		8	
Android 2.1.x	7	ECLAIR_MR1	Platform Highlights
Android 2.0.1	6	ECLAIR_0_1	
Android 2.0	5	ECLAIR	
Android 1.6	4	DONUT	Platform Highlights
Android 1.5	3	CUPCAKE	Platform Highlights
Android 1.1	2	BASE_1_1	
Android 1.0	1	BASE	

Εικόνα 11: Όλες οι εκδόσεις του Android μέχρι σήμερα

4.3 SQLite



Εικόνα 12: Λογότυπο SQLite

4.3.1 Τι είναι το SQLite;

Το SQLite είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που περιέχεται σε μια C προγραμματιστική βιβλιοθήκη. Είναι η πιο δημοφιλής επιλογή ως ενσωματωμένη βάση δεδομένων για τοπική αποθήκευση ή αποθήκευση πελάτη σε λογισμικό εφαρμογής όπως ο φυλλομετρητής. Είναι η πιο ευρέως αναπτυσσόμενη βάση δεδομένων και χρησιμοποιείται, συνδέεται με πολλές γλώσσες και η πηγαία γλώσσα της είναι το public domain. [Wikipedia]

Η μηχανή SQLite δεν έχει αυτόνομες διεργασίες · η βιβλιοθήκη της καλείται δυναμικά και είναι συνδεδεμένη ακολουθώντας την βάση δεδομένων Informix SE. Το πρόγραμμα της εφαρμογής καλεί συναρτήσεις μέσα από απλές διεργασίες για να χρησιμοποιήσει το SQLite, μειώνοντας την καθυστέρηση ενώ η λειτουργικότητα είναι πιο αποτελεσματική σε σχέση με την διαδιεργασική επικοινωνία. [Wikipedia]

Το SQLite υλοποιεί το μεγαλύτερο μέρος του προτύπου SQL-92 για το SQL, αλλά του λείπουν κάποια χαρακτηριστικά. Υποστηρίζει όμως την σύνταξη SQL, τις συναλλαγές και τις έτοιμες καταστάσεις. Έχει συνδέσεις για έναν μεγάλο αριθμό γλωσσών προγραμματισμού, που συμπεριλαμβάνουν: Basic, C/C++, C#, Java, JavaScript, Python, Perl, PHP, Ruby κ.ά. Επιπλέον, η βάση δεδομένων απαιτεί περιορισμένη μνήμη κατά το χρόνο εκτέλεσης (περίπου. 250 KByte) που αποδεικνύει ότι το SQLite είναι καλά προσαρμοσμένο σε ενσωματωμένα συστήματα και περιλαμβάνεται επίσης στα: BlackBerry 10,

Windows Phone 8, iOS, Android, Maemo, MeeGo, webOS, NetBSD, OpenBSD, FreeBSD, illumos. [Wikipedia]

4.3.2 Το SQLite στο Android

Το SQLite είναι ενσωματωμένο σε κάθε συσκευή Android γι' αυτό όταν χρησιμοποιείται μια βάση δεδομένων δεν απαιτεί μια διαδικασία εγκατάστασης ή διαχείριση της βάσης δεδομένων. Το μόνο που έχει να κάνει ο διαχειριστής είναι να καθορίσει τις δηλώσεις SQL για την δημιουργία και την ενημέρωση της βάσης δεδομένων. Έπειτα, η βάση δεδομένων διαχειρίζεται αυτόματα από την πλατφόρμα του Android.

Η πρόσβαση σε μια βάση δεδομένων SQLite περιλαμβάνει την πρόσβαση στο σύστημα αρχείων το οποίο μπορεί να έχει καθυστέρηση. Ως εκ τούτου, συνιστάται να εκτελέσει λειτουργίες βάσης δεδομένων ασύγχρονα.

Εάν μια εφαρμογή δημιουργεί μια βάση δεδομένων, αυτή η βάση δεδομένων είναι εξ ορισμού αποθηκεύονται στο DATA κατάλογο / data / APP_NAME / βάσεων δεδομένων / FILENAME όπου DATA είναι το μονοπάτι που επιστρέφει η μέθοδος Environment.getDataDirectory(), το APP_NAME είναι το όνομα της εφαρμογής και FILENAME είναι το όνομα της βάσης δεδομένων που δίνει ο προγραμματιστής στον κώδικα της εφαρμογής. [Vogel, 2014]

4.3.3 Παράδειγμα

Δημιουργία της κλάσης MySQLiteHelper. Αυτή η κλάση είναι υπεύθυνη για την δημιουργία της βάσης δεδομένων. Η μέθοδος onUpgrade() απλά θα σβήσει όλα τα υπάρχοντα δεδομένα και θα ξαναδημιουργήσει των πίνακα. Επίσης ορίζει αρκετές σταθερές για τον πίνακα και τις στήλες του.

```
package de.vogella.android.sqlite.first;

import android.content.Context;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;
```

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

```
import android.util.Log;

public class MySQLiteHelper extends SQLiteOpenHelper {

    public static final String TABLE_COMMENTS = "comments";

    public static final String COLUMN_ID = "_id";

    public static final String COLUMN_COMMENT = "comment";

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

    private static final int DATABASE_VERSION = 1;

    // Database creation sql statement

    private static final String DATABASE_CREATE = "create table "

        + TABLE_COMMENTS + "(" + COLUMN_ID

        + " integer primary key autoincrement, " + COLUMN_COMMENT

        + " text not null);";

    public MySQLiteHelper(Context context) {

        super(context, DATABASE_NAME, null, DATABASE_VERSION);

    }

    @Override

    public void onCreate(SQLiteDatabase database) {

        database.execSQL(DATABASE_CREATE);

    }

}
```

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

```
@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {

    Log.w(MySQLiteHelper.class.getName(),

        "Upgrading database from version " + oldVersion + " to "

        + newVersion + ", which will destroy all old data");

    db.execSQL("DROP TABLE IF EXISTS " + TABLE_COMMENTS);

    onCreate(db);

}

}
```

Η SQLiteDatabase έχει μεθόδους για να δημιουργήσει, να διαγράψει, να εκτελέσει εντολές SQL και να εκτελέσετε άλλες κοινές εργασίες διαχείρισης βάσεων δεδομένων. [Android Developers, 2014]

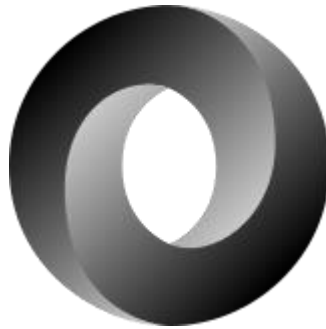
Οι κλήσεις συναρτήσεων (Queries) στην βάση δεδομένων μπορούν να γίνουν με την χρήση των μεθόδων rawQuery() και query(). [Android Developers, 2014]

- Το rawQuery() δέχεται άμεσα ένα SQL select statement ως είσοδο: `Cursor cursor = getReadableDatabase(). rawQuery("select * from todo where _id = ?", new String[] { id });`
- Το και query() παρέχει ένα δομημένο περιβάλλον εργασίας για τον καθορισμό του ερωτήματος SQL: `return database.query(DATABASE_TABLE, new String[] { KEY_ROWID, KEY_CATEGORY, KEY_SUMMARY, KEY_DESCRIPTION }, null, null, null, null, null);`

Ένα query επιστρέφει ένα αντικείμενο Cursor . Ένα Cursor αντιπροσωπεύει το αποτέλεσμα ενός ερωτήματος (query). Με αυτό τον τρόπο το Android μπορούν να κρατήσει μόνο το αποτέλεσμα του ερωτήματος καθώς δεν πρέπει να φορτώσει όλα τα δεδομένα στη μνήμη. [Vogel, 2014]

4.4 Τεχνολογία JSON

4.4.1 Τι είναι το JSON;



Εικόνα 13: Λόγοτυπου του JSON

Το JSON, ή JavaScript Object Notation, είναι μια συμβολική γλώσσα για αναπαράσταση δεδομένων. Βασίζεται σε ένα υποσύνολο της JavaScript το Standard ECMA-262 3rd Edition - December 1999. Χρησιμοποιείται κυρίως για τη μετάδοση δεδομένων μεταξύ server και web εφαρμογής, μοιάζει με την XML (Extensible Markup Language) αλλά είναι πιο κατανοητό από ανθρώπους και μηχανές. Ουσιαστικά, το JSON είναι μια συμβολοσειρά (String), μια μορφή κειμένου, ανεξάρτητη από τις γλώσσες προγραμματισμού αλλά χρησιμοποιεί συμβάσεις με τις οποίες είναι εξοικειωμένοι οι προγραμματιστές των γλωσσών της οικογένειας της C, συμπεριλαμβανομένων των C, C++, C#, Java, JavaScript, Perl, Python και πολλών άλλων. [Wikipedia, 2014] [JSON, 2014]

Αυτές οι ιδιότητες καθιστούν το JSON ιδανική γλώσσα ανταλλαγής δεδομένων. Γι' αυτό υπάρχει κώδικας για την ανάλυση και την παραγωγή δεδομένων JSON που είναι άμεσα διαθέσιμος για μια μεγάλη ποικιλία γλωσσών προγραμματισμού. Πολλές από αυτές έχουν αρχίσει να αναπτύσσουν και βιβλιοθήκες για την ανάγνωση και την γραφή του, ενώ μπορεί να χρησιμοποιηθεί και ως συνδετικός κρίκος των δεδομένων μεταξύ άλλων γλωσσών. [Wikipedia, 2014] [JSON, 2014]

4.5 Google Play Services Library



Εικόνα 14: Google Play Services logo

Οι Google Play Services είναι ένας συνδυασμός ιδιόκτητου SDK (Software Development Kit) και API (Application Programming Interface) για συσκευές Android. Η πρώτη κυκλοφορία ήταν το 2012 και παρείχε APIs που επέτρεπαν στις εφαρμογές να παρέχουν λειτουργικότητα που ενσωματώνεται άμεσα με τις Google Services, όπως ο συγχρονισμός των λογαριασμών, Google+ (κοινοποίηση, δυνατότητα single sign-on), Google Maps, Location APIs, Google Play Games, Cloud Messaging, Android Device Manager κ.ά. Το Google Play Services είναι μια εξάρτηση όλων των εφαρμογών της Google για Android.

Παρά το γεγονός ότι ενεργεί ως μια διαδικασία σε επίπεδο συστήματος, ενημερώνεται αυτόματα μέσω του Google Play Store και εγκαθίσταται επίσης αυτόματα σε κάθε Android συσκευή που τρέχει Android 2.2 ή νεότερη έκδοση. Ως αποτέλεσμα, η Google μπορεί να ενημερώσει ορισμένες πτυχές του Android ανεξάρτητα από το ίδιο το λειτουργικό σύστημα, την καταπολεμώντας τον "κατακερματισμό" (fragmentation) και την ασυνεπή διαθεσιμότητα των νέων εκδόσεων του λειτουργικού συστήματος σε ορισμένες συσκευές, ένα θέμα για το οποίο η πλατφόρμα είχε γίνει διαβόητη.

Οι Google Play Services συχνά αναφέρονται ως εσωτερικά Google Mobile Services (GMS) στην τεκμηρίωση του προγραμματιστή (developer

documentation). Η τελευταία έκδοση, 6.1, κυκλοφόρησε στις 16 Σεπτεμβρίου του 2014. [Wikipedia, 2014, Google Play Services]

Προσφέρουν στις εφαρμογές περισσότερα χαρακτηριστικά για να προσελκύουν χρήστες σε μια ποικιλία συσκευών. Οι βιβλιοθήκες πελατών, που είναι εύκολες στην χρήση, είναι διαθέσιμες για κάθε υπηρεσία που μας επιτρέπει να υλοποιήσουμε την λειτουργικότητα που θέλουμε ευκολότερα και γρηγορότερα. Όλα τα προϊόντα στις Google Play Services μοιράζονται ένα κοινό εξουσιοδοτημένο API που αξιοποιεί τους υπάρχοντες λογαριασμούς της Google στην συσκευή. Οι χρήστες έχουν ένα συνεχή και ασφαλές τρόπο να λάβουν tokens πρόσβασης OAuth2 στις υπηρεσίες της Google. Εφόσον οι Android συσκευές που τρέχουν Android 2.2 ή νεότερη έκδοση λαμβάνουν αυτόματα ενημερώσεις για το Google Play Services, οι προγραμματιστές μπορούν να βελτιώσουν την εφαρμογή τους χωρίς να ανησυχούν για την έκδοση του Android που υπάρχει στις συσκευές των χρηστών. [Android Developers, 2014, Google Play Services]

4.5.1 Εγκατάσταση των Google Play Services στο Eclipse με ADT

Για να υλοποιηθεί μια διαδικτυακή υπηρεσία που να χρησιμοποιεί τα [Google Play services APIs](#), είναι απαραίτητη η ενσωμάτωση στην εφαρμογή του SDK των Google Play Services μέσω του Android SDK Manager. Για να ελέγξουμε ότι λειτουργεί σωστά η εφαρμογή όταν χρησιμοποιεί το SDK, τότε είτε χρησιμοποιούμε Android emulator με ένα AVD που τρέχει την πλατφόρμα του Google API, που βασίζεται στο Android 4.2.2 ή νεότερη έκδοση είτε μια συμβατή συσκευή που τρέχει Android 2.2 ή νεότερη έκδοση και περιλαμβάνει το Google Play Store. Επιπλέον, αφού έχουμε εγκαταστήσει το SDK στο Eclipse θα πρέπει να προσθέσουμε την βιβλιοθήκη των Google Play Services στην εφαρμογή. [Android Developers, 2014, Google Play Services - Set Up]

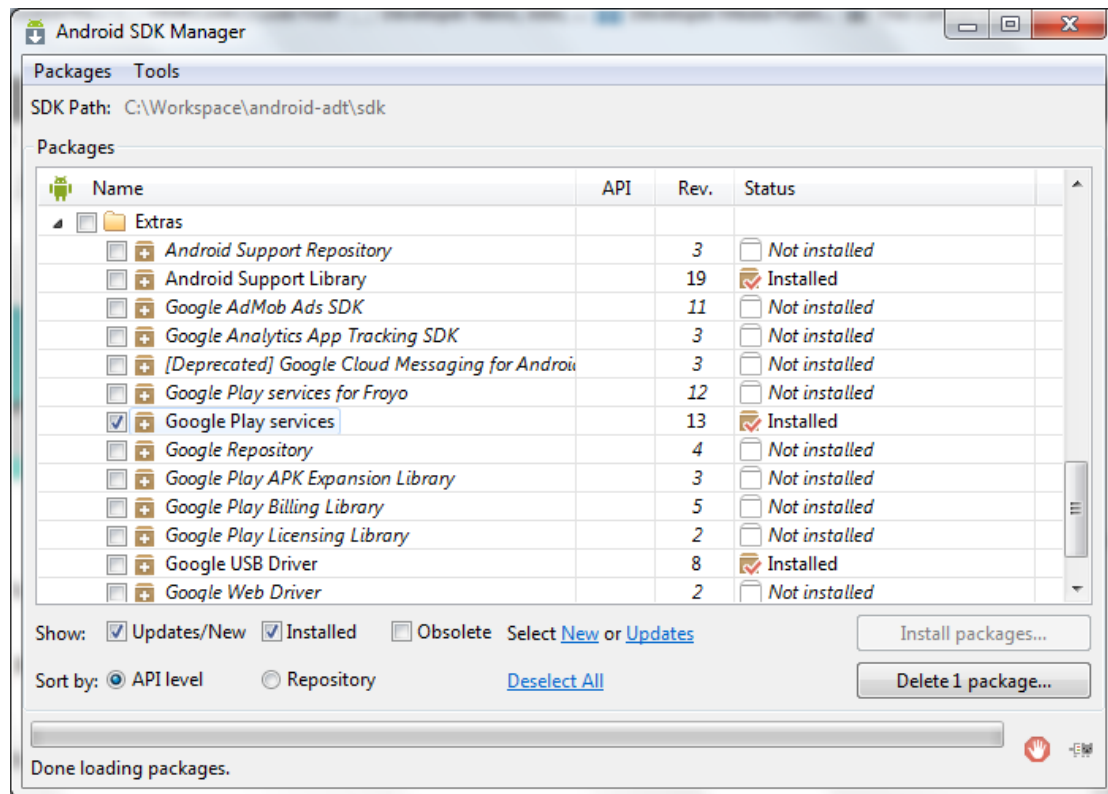
Εισαγωγή της βιβλιοθήκης στο χώρο εργασίας του Eclipse. Κάντε κλικ στο Αρχείο> Εισαγωγή, επιλέξτε το Android> Υφιστάμενος Android κώδικας στο Workspace, και αναζήτηση της βιβλιοθήκης για να εισαχθεί στην εφαρμογή. Αφού ολοκληρωθεί η εισαγωγή, ανοίγουμε το αρχείο AndroidManifest και προσθέτουμε το παρακάτω κομμάτι κώδικα σαν "παιδί" στο στοιχείο <application>: [Android Developers, 2014, Google Play Services - Set Up]

```
<meta-data
```

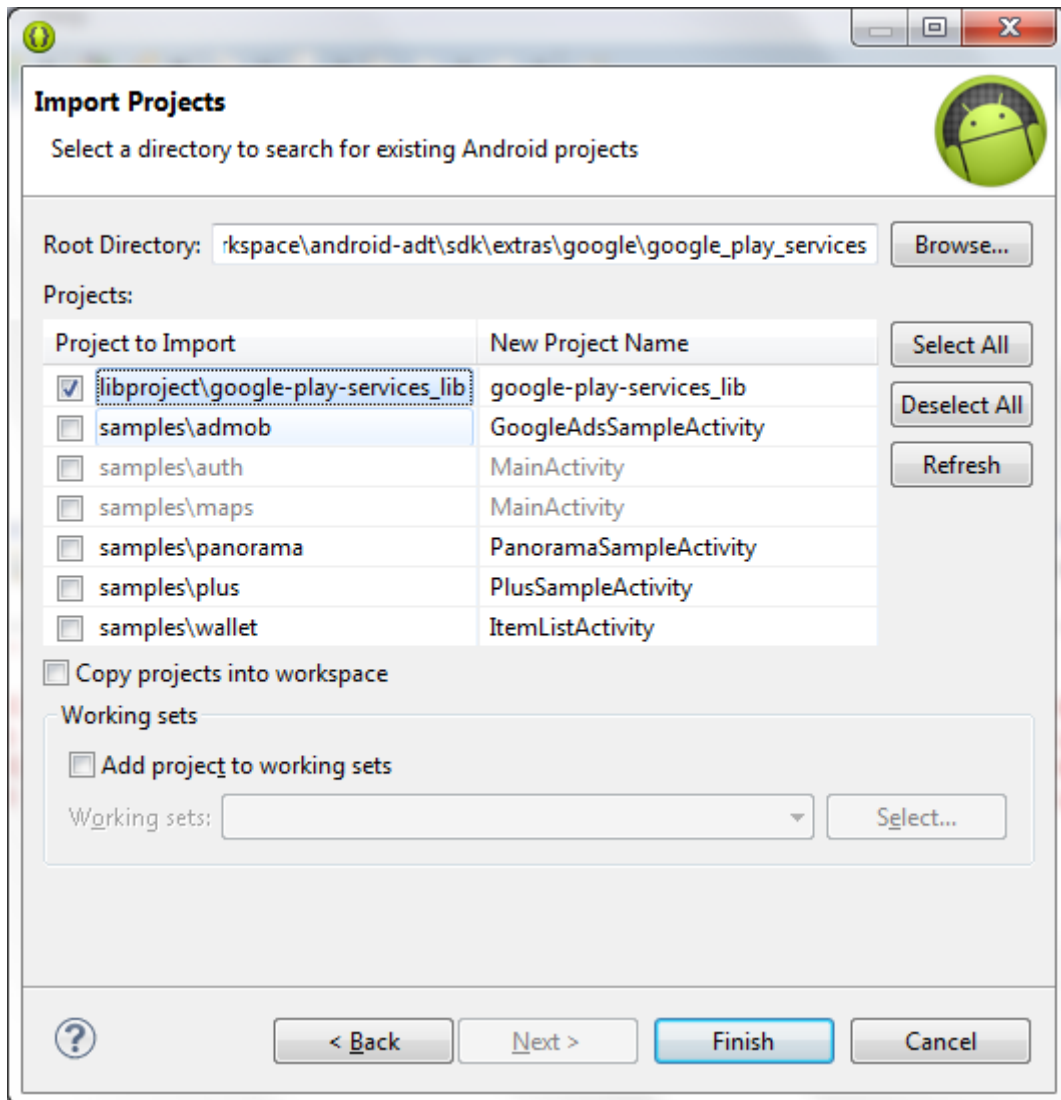
```
android:name="com.google.android.gms.version"
```

```
android:value="@integer/google_play_services_version" />
```

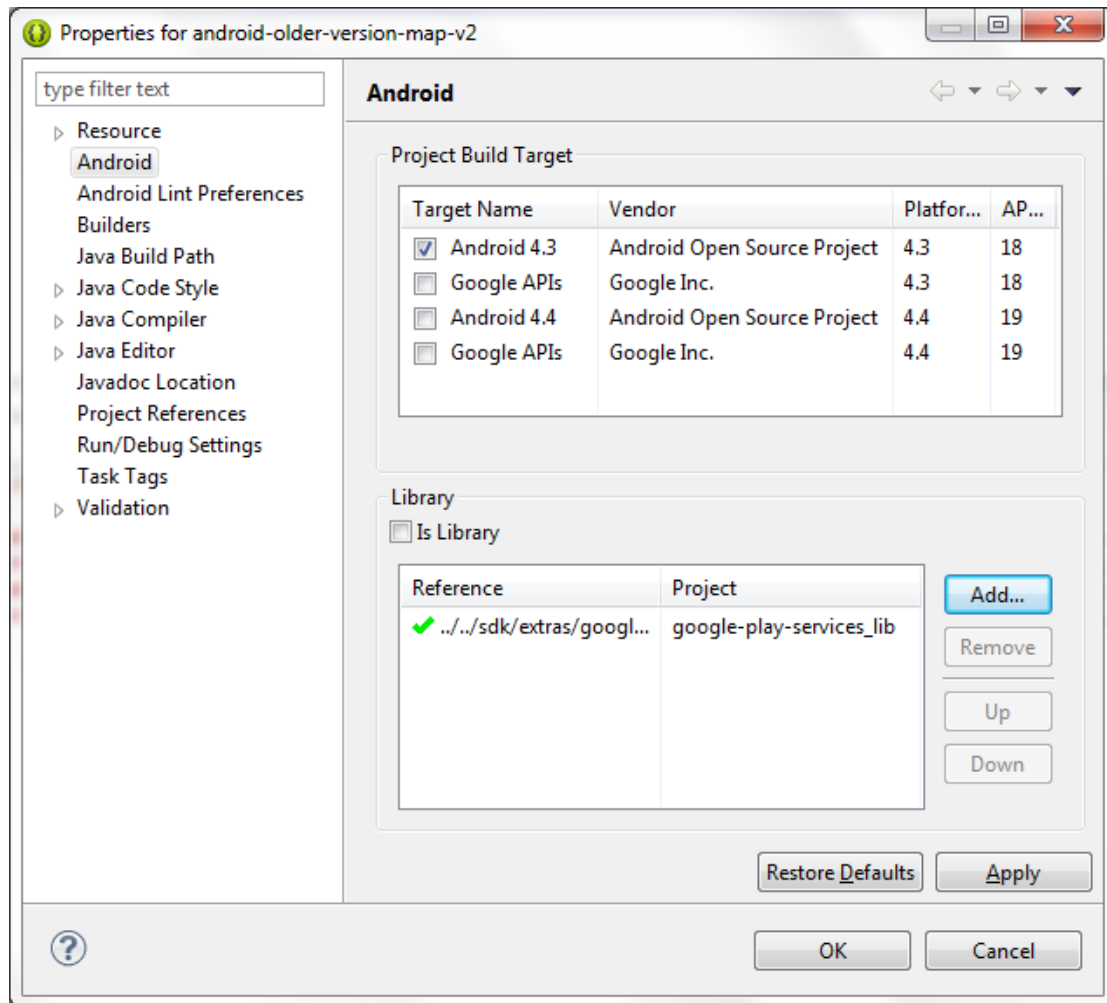
[hmkcode.com, 14/11/2014, Add Google Play Services Library]



Εικόνα 15: Εγκατάσταση του Android SKD Manager



Εικόνα 16: Εισαγωγή της βιβλιοθήκης στο workspace του Eclipse



Εικόνα 17: Προσθήκη της βιβλιοθήκης στην εφαρμογή

4.6 Επίλογος

Στο κεφάλαιο αυτό έγινε ανάλυση της πλατφόρμας Android για την κάθε έκδοσή του και του εργαλείου Eclipse. Επίσης περιγράφηκαν οι τεχνολογίες JSON και SQLite που χρησιμοποιήθηκαν στην υπηρεσία καθώς και η βιβλιοθήκη Google Play Services και το πώς θα ενσωματωθεί στην εφαρμογή. Εφόσον υπάρχει μια πλήρης εικόνα για τις τεχνολογίες που χρησιμοποιήθηκαν και το εργαλείο ανάπτυξης, στο επόμενο κεφάλαιο θα εξηγήσουμε την σχεδίαση και υλοποίηση της εφαρμογής.

ΚΕΦΑΛΑΙΟ 5: Η εφαρμογή “Thessaloniki Parking”

Το κεφάλαιο αυτό περιλαμβάνει τον πλήρη σχεδιασμό της εφαρμογής. Αρχικά παραπέμπει τις κλάσεις και το πως επιδρούν μεταξύ τους. Έπειτα με την βοήθεια εικόνων (screenshots) δείχνει την σχεδίαση (το layout) της εφαρμογής και έπειτα αναλύεται λεπτομερώς η υλοποίηση της εφαρμογής. Για τα διαγράμματα των κλάσεων χρησιμοποιήθηκε το εργαλείο Visual Paradigm.

5.1 Αρχική Διάκριση κλάσεων

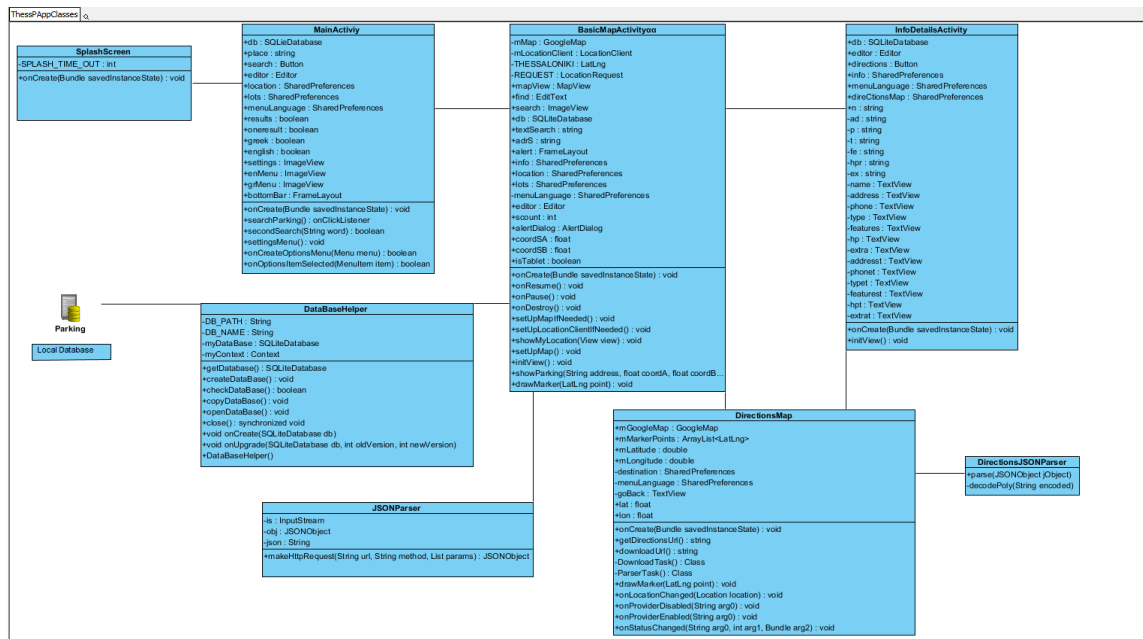
Η εφαρμογή αποτελείται από επτά κλάσεις, δύο όμως είναι αυτές που ξεχωρίζουν · η μια είναι η MainActivity όπου ο χρήστης επιλέγει την γλώσσα της εφαρμογής (αγγλικά ή ελληνικά) και πληκτρολογεί την διεύθυνση όπου επιθυμεί να βρει parking ή επιλέγει να ανοίξει κατευθείαν τον χάρτη. Η δεύτερη είναι η BasicMapActivity η οποία δημιουργεί την βάση δεδομένων στην συσκευή - αφού ελέγξει αν υπάρχει ήδη - καλώντας τις μεθόδους από την κλάση DataBaseHelper και ψάχνει στη βάση δεδομένων για parking στην διεύθυνση που δόθηκε στην κλάση MainActivity ή κοντά στην τωρινή τοποθεσία του χρήστη. Επίσης ο χρήστης μπορεί να κάνει νέα αναζήτηση και να δει ποια parking έχουν προσφορές και ποια ενημερώνουν πόσες άδειες θέσεις έχουν.

Η εφαρμογή ξεκινά με την κλάση SplashScreen η οποία εμφανίζει ένα παράθυρο, καλύπτει όλη την οθόνη, με το λογότυπο και το όνομα της εφαρμογής μέχρι να φορτώσει το κύριο πρόγραμμα. Το παράθυρο αυτό ονομάζεται splash screen και είναι ένα στοιχείο ελέγχου γραφικών. [Wikipedia, 2014]

Έπειτα, ακολουθεί η MainActivity και μετά η BasicMapActivity, που αναφέρθηκαν παραπάνω, για να γίνει η αναζήτηση. Επίσης, η κλάση αυτή συνδέεται με μια άλλη την JSONParser η οποία συνδέεται με έναν server όπου οι ιδιοκτήτες των parking ενημερώνουν για τις διαθέσιμες θέσεις και τις προσφορές τους. Με τις κατάλληλες μεθόδους παίρνουμε τα στοιχεία αυτά από το url με την χρήση του json και τα μετατρέπουμε σε string ώστε να χρησιμοποιηθούν στην εφαρμογή για να ενημερωθεί ο χρήστης. Αν επιλέξει ο

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσική

χρήστης ένα parking lot από τον χάρτη της BasicMapActivity ανοίγει ένα alertDialog παράθυρο με τις εξής πληροφορίες: όνομα parking, διεύθυνση, προσφορές και διαθέσιμες θέσεις. Τα δύο τελευταία εμφανίζονται αν ο ιδιοκτήτης παρέχει αυτές τις πληροφορίες αλλιώς εμφανίζονται μόνο τα δύο πρώτα πεδία. Αν πατήσει (ο χρήστης) στο ανοιχτό παράθυρο τότε θα μεταφερθεί στην κλάση InfoDetails όπου υπάρχουν όλες οι διαθέσιμες πληροφορίες (πχ ωράρια, τιμές κλπ). Από εκεί με το πάτημα του κουμπιού "Οδηγίες" εμφανίζει την κλάση DirectionsMap η οποία συνδέεται με την κλάση DirectionsJSONParser. Η πρώτη κλάση εμφανίζει στο χάρτη δύο pins· πράσινο για την τοποθεσία του χρήστη και κόκκινο για τον προορισμό ενώ η διαδρομή φαίνεται με μια κόκκινη γραμμή στον χάρτη. Η δεύτερη λαμβάνει αντικείμενο τύπου JSON και επιστρέφει μια λίστα με συντεταγμένες.



Εικόνα 18: Οι κλάσεις της εφαρμογής και πώς συνδέονται μεταξύ τους

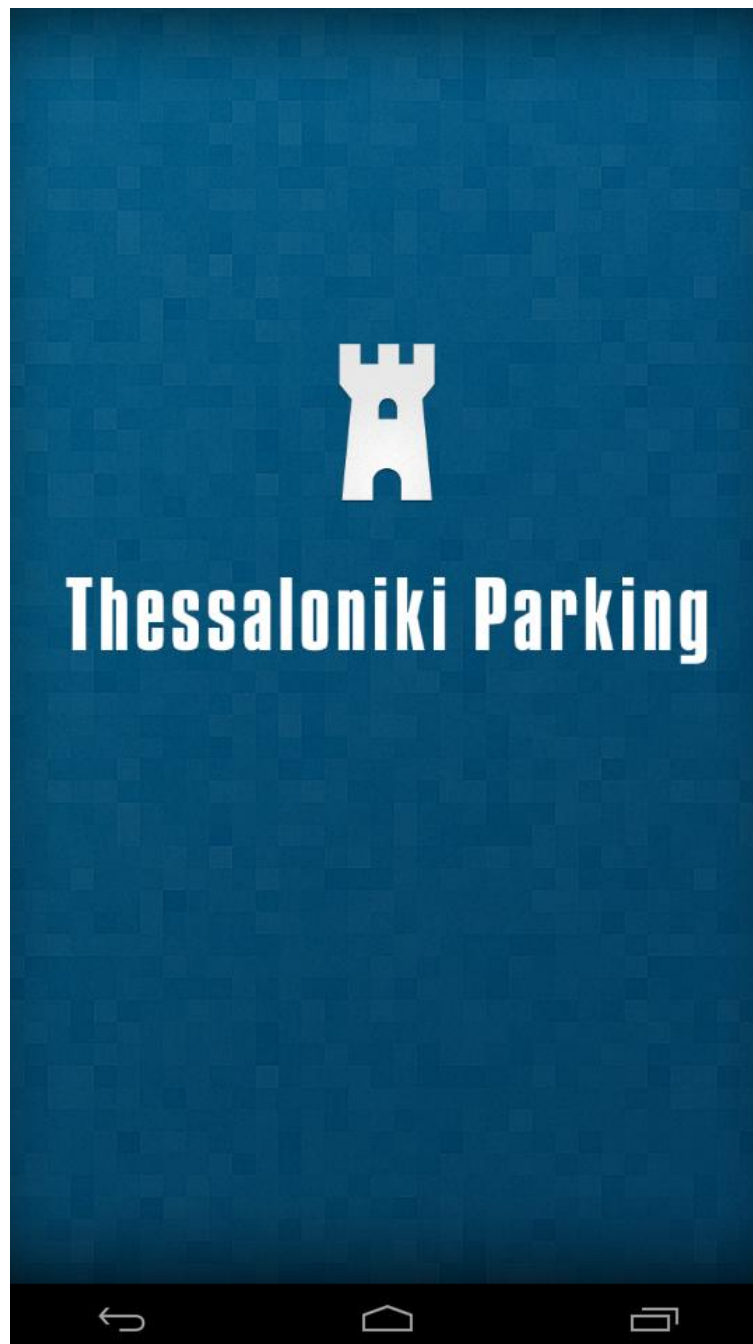
5.2 Σχεδίαση Εφαρμογής

Η καλή σχεδίαση επιτυγχάνεται όταν η εφαρμογή γίνεται όσο το δυνατόν πιο εύχρηστη και απλούστερη. Για τον λόγο αυτό θα προτιμηθούν απλές περιεκτικές οθόνες, που θα περιλαμβάνουν όλες τις απαραίτητες λειτουργίες

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

και πληροφορίες. Για μεγαλύτερη ευκολία και εφ' όσον ο χρήστης έχει ενεργοποιημένο το rotation της συσκευής μπορεί να την γυρίσει για να δει τον χάρτη σε landscape μορφή.

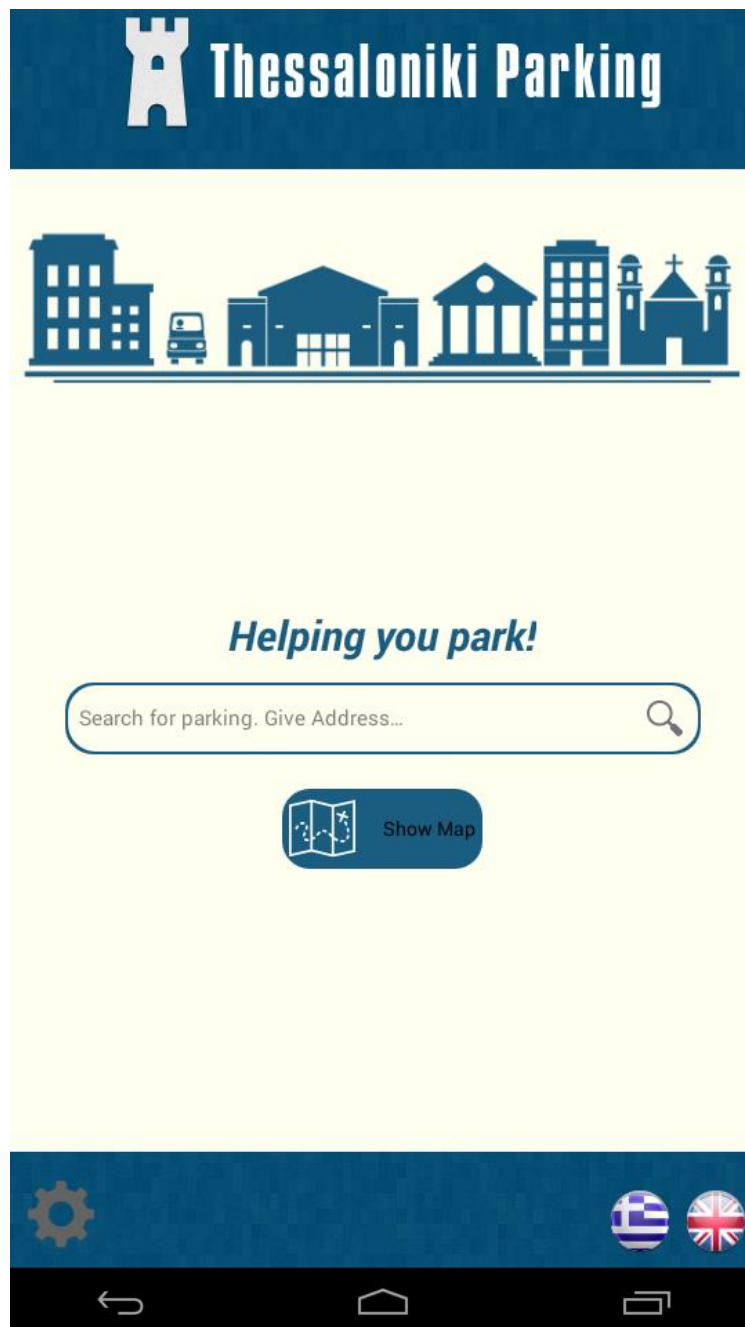
Αρχικά η εφαρμογή εμφανίζει μια splash screen μέχρι να φορτώσει το πρόγραμμα:



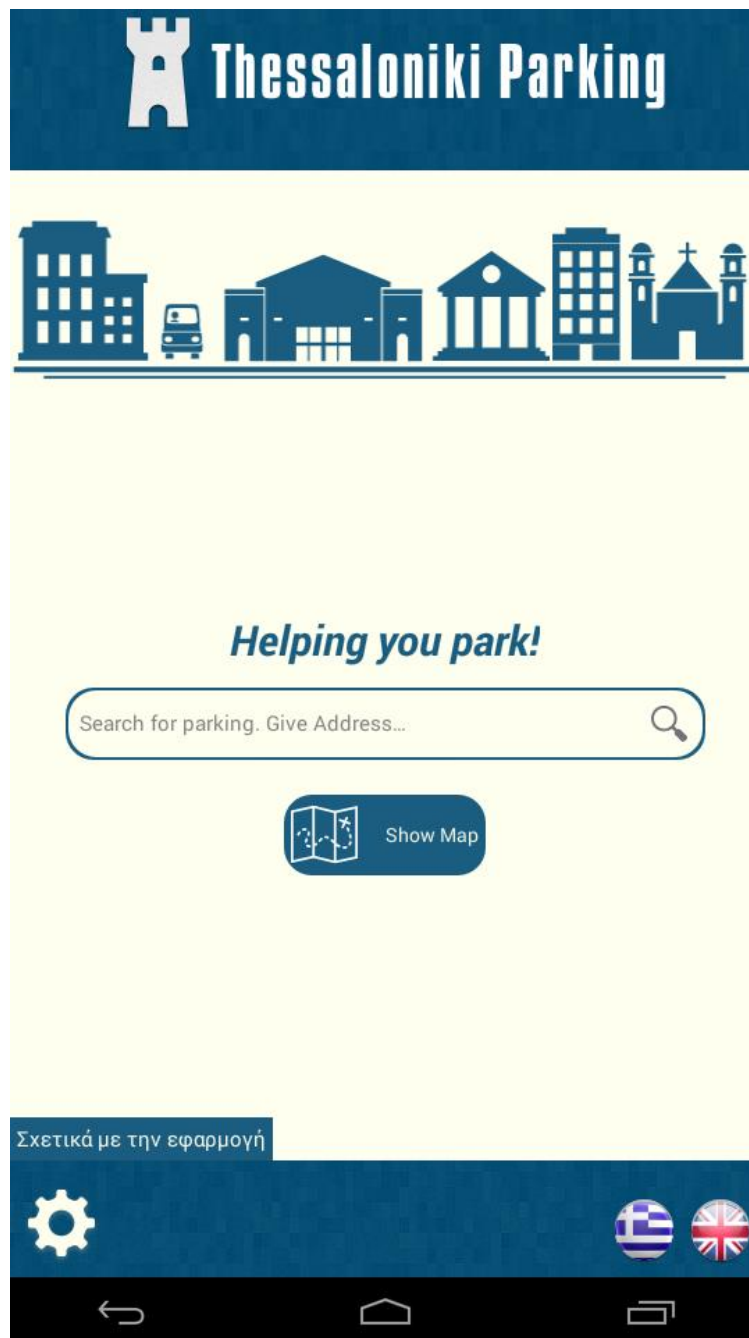
Εικόνα 19: Splash Screen

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

Έπειτα φορτώνει την MainActivity. Το layout της κλάσης περιέχει ένα edit view όπου ο χρήστης πληκτρολογεί την διεύθυνση που θέλει και πατάει το κουμπί "Show Map" αλλιώς πατάει κατευθείαν το κουμπί και ανοίγει εμφανίζεται ο χάρτης με κέντρο την τωρινή τοποθεσία του χρήστη, δείχνοντας τα parking που βρίσκονται σε κοντινή απόσταση. Επίσης, κάτω δεξιά ο χρήστης μπορεί να αλλάξει την γλώσσα της εφαρμογής, πατώντας το αντίστοιχο κουμπί ενώ από την αριστερή μεριά μπορεί να βρει πληροφορίες για την εφαρμογή (και στις δύο γλώσσες).



Εικόνα 20: MainActivity



Εικόνα 21: Επιλογή ελληνικής γλώσσας για την MainActivity



Εικόνα 22: Σχετικά με την εφαρμογή

Μόλις πατηθεί το κουμπί "Show Map/Χάρτης", εμφανίζεται η τρίτη οθόνη (BasicMapActivity) που περιέχει τον χάρτη. Μόλις φορτώσει η βάση δεδομένων εμφανίζονται τα parking lots ως pins/markers στον χάρτη. Αυτά τα οποία έχουν προσφορές ή διαθέσιμες θέσεις είναι μπλε ενώ τα υπόλοιπα είναι κόκκινα. Επίσης πάνω αριστερά υπάρχει ένα edit view έτσι ώστε αν ο χρήστης θέλει να κάνει μια νέα αναζήτηση μπορεί να γράψει εκεί την διεύθυνση χωρίς να χρειαστεί να γυρίσει στην προηγούμενη οθόνη. Η εφαρμογή χρησιμοποιεί το Google maps για να εμφανίζεται ο χάρτης και πρέπει η συσκευή να είναι συνδεδεμένη στο διαδίκτυο γι' αυτό απαιτούνται κάποια permissions στο AndroidManifest.xml καθώς και ένα API key για να μπορεί η εφαρμογή να χρησιμοποιεί το Google maps και να μπορεί να βρίσκει την τοποθεσία (current location) μέσω του διαδικτύου: `<uses-permission android:name="android.permission.INTERNET" />`

```
<uses-permission  
android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

<!-- Allows the API to use WiFi or mobile cell data (or both) to determine the device's location. -->

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

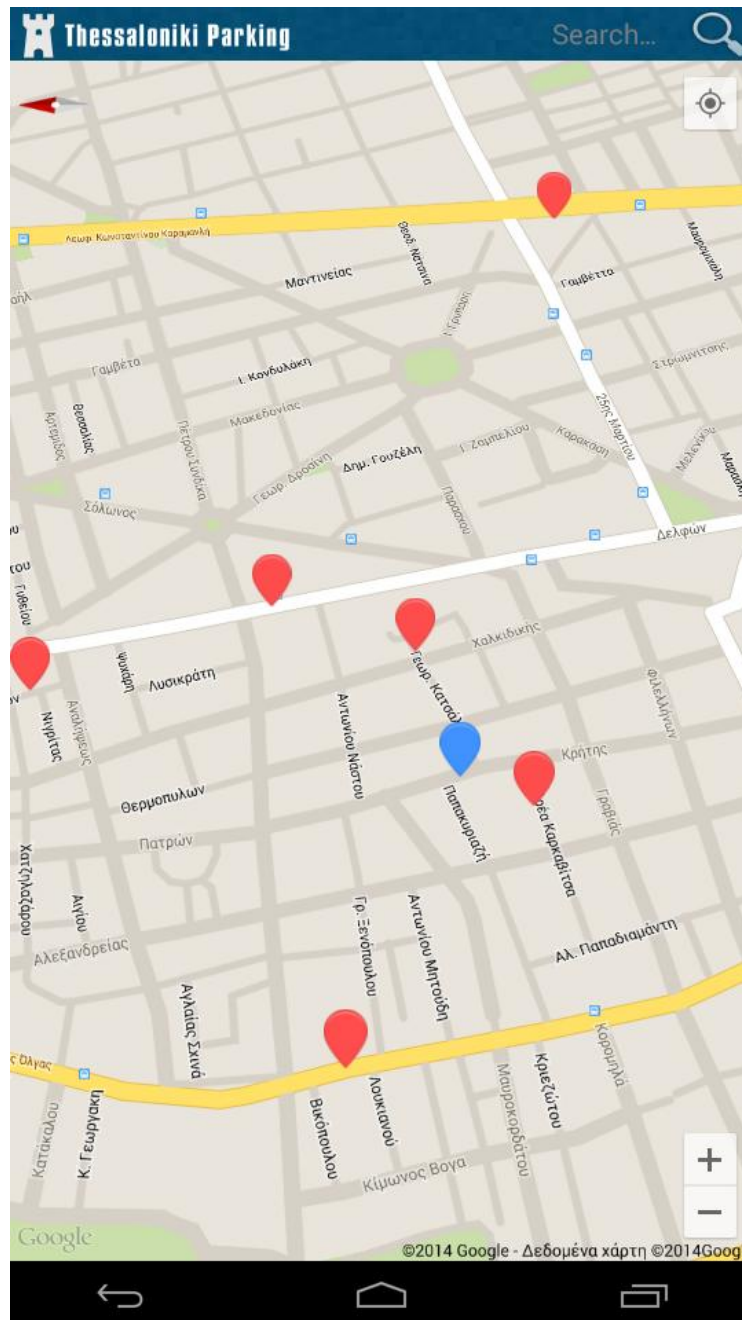
<!-- Allows the API to use the Global Positioning System (GPS) to determine the device's location to within a very small area. -->

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

```
<meta-data android:name="com.google.android.maps.v2.API_KEY"  
android:value="AlzaSyBtjKgA08TXyN4AF6t5jX25LZPvIYwZEUY" />
```

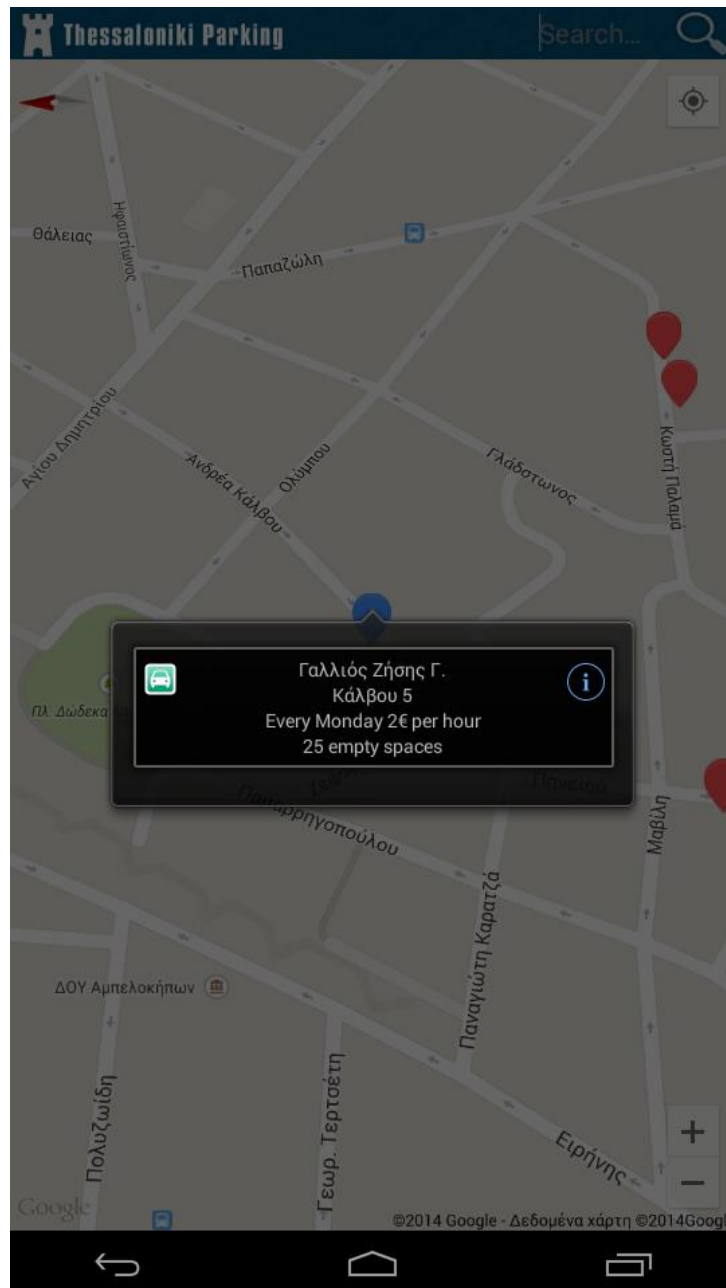
```
<meta-data  
android:name="com.google.android.gms.version"  
android:value="@integer/google_play_services_version" />
```

```
<uses-library android:name="com.google.android.maps" />
```

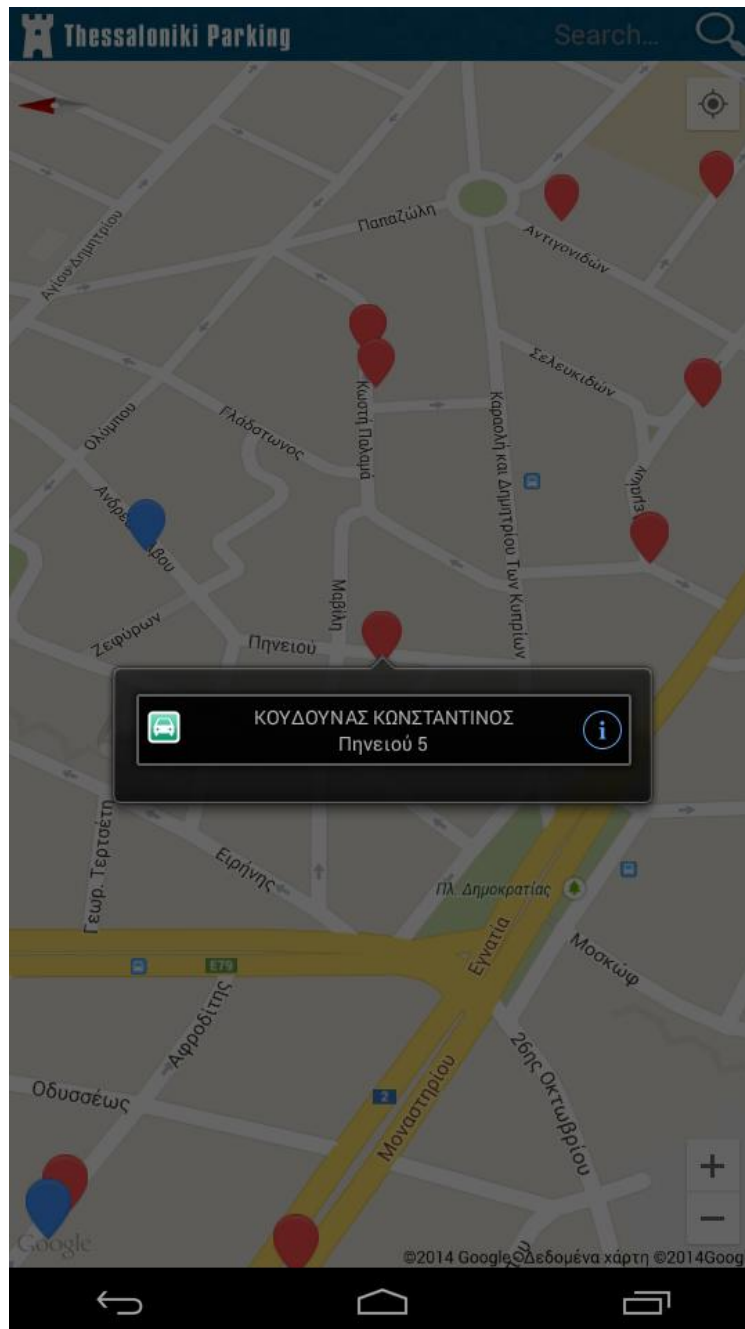


Εικόνα 23: Το layout της κλάσης BasicMapActivity

Αν ο χρήστης επιλέξει ένα από αυτά, για παράδειγμα το μπλε pin στην εικόνα 15, τότε θα ανοίξει ένα dialog με τις πληροφορίες του αντίστοιχου parking και πατώντας το θα εμφανιστεί η κλάση InfoDetails:



Εικόνα 24: alertDialog του parking lot που περιέχει προσφορά και θέσεις



Εικόνα 25: alertDialog του parking lot χωρίς προσφορά και θέσεις

Η InfoDetails κλάση εμφανίζει όλες τις πληροφορίες που είναι διαθέσιμες ανάλογα με το ποιο parking επιλέξει ο χρήστης. Στο layout εμφανίζονται μόνο τα πεδία που έχουν πληροφορίες:

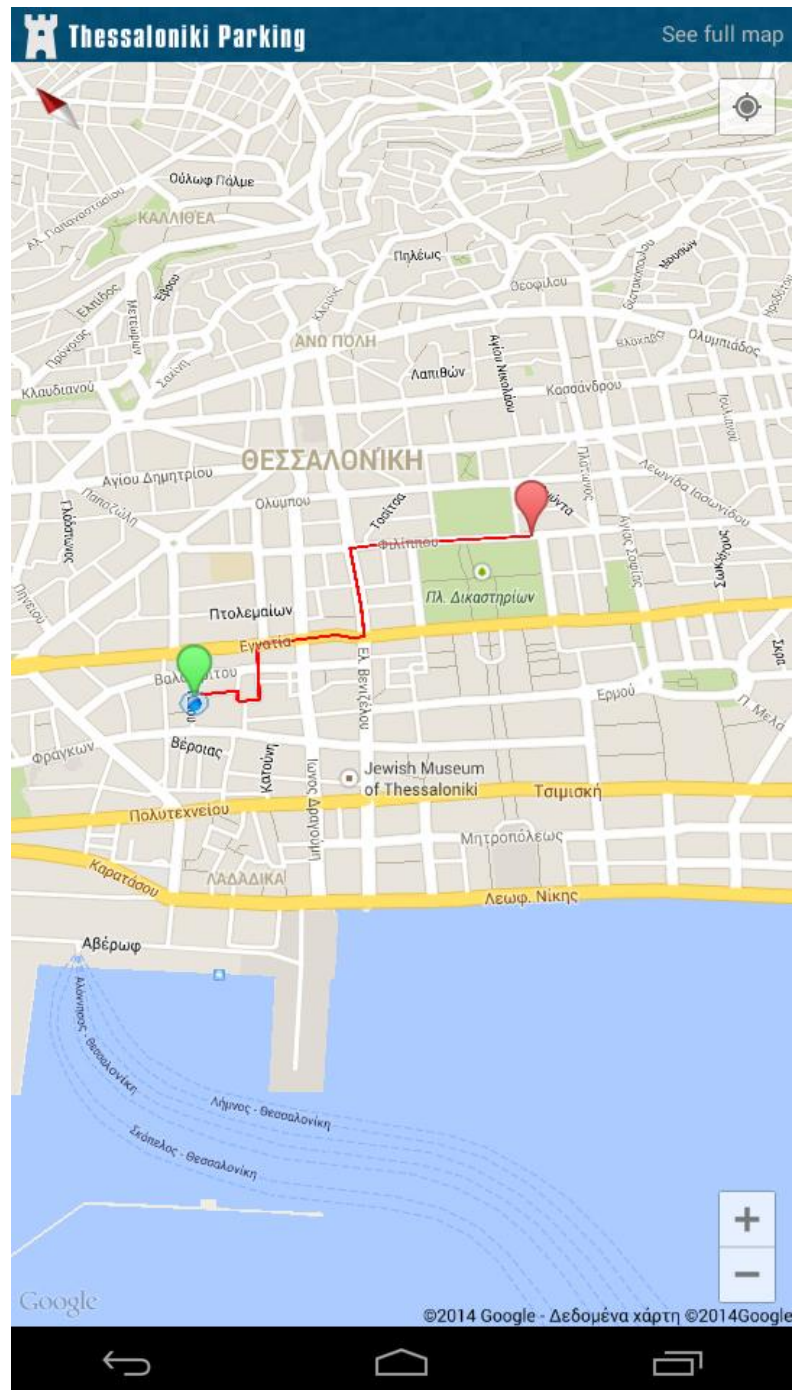


Εικόνα 26: Παράδειγμα του layout της InfoDetails



Εικόνα 27: Παράδειγμα του layout της InfoDetails

Αν ο χρήστης πατήσει το κουμπί "Get directions" θα ανοίξει η νέα οθόνη (DirectionsMap) που δείχνει την διαδρομή από την τωρινή τοποθεσία του χρήστη ως τον προορισμό ακολουθώντας την κατεύθυνση που φαίνεται στο χάρτη κάνοντας "parse" το Google Directions με την μορφή JSON λαμβάνοντας τις συντεταγμένες με την χρήση της κλάσης DirectionsJSONParser που τις επιστρέφει σε μορφή αντικειμένου JSON.



Εικόνα 28: Το layout της κλάσης DirectionsMap

5.3 Υλοποίηση - Λειτουργικότητα προγράμματος

Στην πρώτη εκτέλεση του προγράμματος η MainActivity δημιουργεί την βάση δεδομένων στην συσκευή για να μπορούμε να έχουμε πρόσβαση στα δεδομένα μέσω των query.

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κασίκη

```
DataBaseHelper myDbHelper;

myDbHelper = new DataBaseHelper(this);

try {

    myDbHelper.createDataBase();

} catch (IOException ioe) {

    throw new Error("Unable to create database");

}

try {

    myDbHelper.openDataBase();

} catch (SQLException sqle) {

    throw sqle;

}

db = myDbHelper.getDatabase();

@SuppressWarnings("unused")

Cursor cursor1 = db.rawQuery("SELECT * FROM sqlite_master
WHERE type='table' AND name='LotsGarages'",

    null);

Cursor cursor = db.rawQuery("select * from LotsGarages",
null);// order by PhoneNumber DESC

cursor.moveToFirst();

for (int i = 0; i < cursor.getCount(); i++) {

    cursor.moveToPosition(i);
```

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

```
}
```

Παρατηρούμε στον κώδικα ότι καλεί την μέθοδο `createDataBase()`; της κλάσης `DataBaseHelper`. η μέθοδος αυτή ελέγχει αν η βάση υπάρχει ήδη κι αν όχι τότε την δημιουργεί:

```
/**
 * Creates a empty database on the system and rewrites it with
 your own database.      * */

public void createDataBase () throws IOException {

    boolean dbExist = checkDataBase ();

    if (dbExist) {

        // do nothing - database already exist

        Log.d("createDataBase", "do nothing - database
already exists");

    } else {

        // By calling this method and empty database will
be created into

        // the default system path

        // of your application so we are gonna be able to
overwrite that

        // database with our database.

        this.getReadableDatabase ();

        try {

            copyDataBase ();

            Log.d("createDataBase", "database does not
exist");
```

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κασίκη

```
    } catch (IOException e) {  
  
        throw new Error("Error copying database");  
  
    }  
  
}  
  
}
```

Καλώντας αυτήν την μέθοδο θα δημιουργηθεί μια άδεια βάση δεδομένων στο default σύστημα της εφαρμογής η οποία θα γίνει overwrite από την δική μας.

```
// The Android's default system path of your application database.  
  
private static String DB_PATH =  
"/data/data/com.example.salonikaparking/databases/";  
  
private static String DB_NAME = "parking.sqlite";  
  
private boolean checkDataBase() {  
  
    SQLiteDatabase checkDB = null;  
  
    try {  
  
        String myPath = DB_PATH + DB_NAME;  
  
        checkDB = SQLiteDatabase.openDatabase(myPath, null,  
  
            SQLiteDatabase.OPEN_READONLY);  
  
        Log.d("checkDataBase", "openDatabase");  
  
    } catch (SQLiteException e) {
```

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

```
        // database does't exist yet.
    }

    if (checkDB != null) {
        checkDB.close();
    }

    return checkDB != null ? true : false;
}
}
```

Η μέθοδος αυτή ελέγχει αν η βάση δεδομένων υπάρχει ήδη για να αποφευχθεί να ξαναγραφτεί το αρχείο κάθε φορά που ανοίγουμε την εφαρμογή.

```
private void copyDataBase() throws IOException {

    Log.d("copyDataBase", "copy on device");

    // Open your local db as the input stream
    InputStream myInput =myContext.getAssets().open(DB_NAME);

    // Path to the just created empty db
    String outFileName = DB_PATH + DB_NAME;

    // Open the empty db as the output stream
    OutputStream myOutput =new FileOutputStream(outFileName);
```

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κασίκη

```
// transfer bytes from the inputfile to the outputfile

byte[] buffer = new byte[1024];

int length;

while ((length = myInput.read(buffer)) > 0) {

    myOutput.write(buffer, 0, length);

}

// Close the streams

myOutput.flush();

myOutput.close();

myInput.close();

}
```

Η παραπάνω μέθοδος αντιγράφει την δική μας βάση δεδομένων από τον τοπικό φάκελο `assets` στην άδεια (βάση δεδομένων) που μόλις δημιουργήσαμε στο φάκελο του συστήματος, από όπου μπορεί να είναι προσβάσιμη και διαχειρίσιμη. Αυτό επιτυγχάνεται με την μεταφορά `bytestream`.

```
public void openDataBase() throws SQLException {

    // Open the database

    String myPath = DB_PATH + DB_NAME;

    myDataBase = SQLiteDatabase.openDatabase(myPath, null,

        SQLiteDatabase.OPEN_READONLY);

}

@Override
```

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

```
public synchronized void close() {  
  
    if (myDataBase != null)  
  
        myDataBase.close();  
  
    super.close();}
```

Έπειτα η MainActivity περιέχει την κλάση LoadAllProducts η οποία παίρνει από ένα url τα δεδομένα τα οποία θα χρησιμοποιηθούν για τα πεδία προσφορές και άδειες θέσεις.

```
class LoadAllProducts extends AsyncTask<String, String, String> {  
  
    int success;  
  
    // getting All products from url  
  
    protected String doInBackground(String... args) {  
  
        // Building Parameters  
  
        List<NameValuePair> params = new  
ArrayList<NameValuePair>();  
  
        // getting JSON string from URL  
  
        JSONObject json =  
jParser.makeHttpRequest(url all products, "GET", params);  
  
        // Check your log cat for JSON reponse  
  
        //Log.d("All Products: ", json.toString());  
  
        try {  
  
            // Checking for SUCCESS TAG
```

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

```
try {  
  
    success = json.getInt(TAG_SUCCESS);  
  
} catch (NullPointerException e) {  
  
    System.out.println("success: " + success);  
  
}  
  
if (success == 1) {  
  
    // products found  
  
    // Getting Array of Products  
  
    products = json.getJSONArray(TAG_PRODUCTS);  
  
    // looping through All Products  
  
    for (int i = 0; i < products.length(); i++)  
  
    {  
  
        JSONObject c=products.getJSONObject(i);  
  
        // Storing each json item in variable  
  
        String offers= c.getString(TAG_OFFERS);  
  
        String spaces = c.getString(TAG_FREESPACES);  
  
        String address = c.getString(TAG_ADDRESS);  
  
        offerupdates =  
getSharedPreferences("updates", MODE_PRIVATE);  
  
        editor = offerupdates.edit();  
  
        editor.putString("address", address);  
  
        editor.putString("offers", offers);  
    }  
}
```

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

```
        editor.putString("spaces", spaces);

        editor.commit();

    }

    } else {

        // no products found

    }

    } catch (JSONException e) {

        e.printStackTrace();

    }

    return null;

}
```

Τα δεδομένα τα αποθηκεύουμε στο SharedPreferences για να μπορούμε να έχουμε εύκολη πρόσβαση σε αυτά από κάθε κλάση.

Η εφαρμογή τώρα είναι έτοιμη για να κάνει την αναζήτηση. Όταν πατηθεί το button "Χάρτης" καλείται η μέθοδος onClickListener searchParking στην οποία παίρνουμε το περιεχόμενο του edit view και κάνουμε αναζήτηση στη βάση δεδομένων για να δούμε αν υπάρχει parking στην συγκεκριμένη διεύθυνση :

```
Cursor cursor = db.rawQuery("SELECT CoordinateA, CoordinateB FROM  
LotsGarages WHERE (Address2 LIKE '%" + place.toUpperCase() + "%' OR  
Address3 LIKE '%" + place.toUpperCase() + "%')", null);
```

Αν υπάρχουν αποτελέσματα τότε αποθηκεύει τα στοιχεία στο SharedPreferences και ανοίγει την επόμενη FragmentActivity. Αν έχει δοθεί λάθος διεύθυνση εμφανίζει μήνυμα λάθους, ενώ αν πατήσουμε κατευθείαν το κουμπί ανοίγει ο χάρτης με κέντρο την Θεσσαλονίκη.

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

Στην κλάση BasicMapActivity το πρώτο που πρέπει να κάνουμε είναι να αρχικοποιήσουμε τον χάρτη ως εξής:

```
private void setUpMapIfNeeded() {

    if (mMap == null) {

        //Try to obtain the map from the SupportMapFragment.

        mMap = ((SupportMapFragment)
getSupportFragmentManager()

                .findFragmentById(R.id.map)).getMap();

        //Check if we were successful in obtaining the map.

        if (mMap != null) {

            mMap.setMyLocationEnabled(true);
            mMap.setOnMyLocationButtonClickListener(this);

            setUpMap();

        }

    }

}
```

Ελέγχουμε να δούμε αν ο χάρτης έχει ήδη αρχικοποιηθεί. Αν δεν έχει αρχικοποιηθεί καλούμε την μέθοδο setUpMap() και ορίζουμε την θέση της "κάμερας" πάνω στον χάρτη:

```
private void setUpMap() {

    Log.d("setUpMap", "position the map");

    mMap.setBuildingsEnabled(true);

    CameraPosition cameraPosition = new
    CameraPosition.Builder()

        .target(THESSALONIKI)

        .zoom(16)

        .bearing(90)
```

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

```
        .tilt(30)

        .build();
mMap.animateCamera(CameraUpdateFactory

        .newCameraPosition(cameraPosition));

    initView();
}
```

Στην επόμενη μέθοδο, την `initView()`, παίρνουμε τα δεδομένα από την βάση δεδομένων για να φτιάξουμε τους `markers` στον χάρτη. Στην μέθοδο που καλείται, `drawMarker(new LatLng(lat,lng))`, δίνουμε τις συντεταγμένες του κάθε `parking` ως `argument` και ελέγχουμε για το καθένα αν υπάρχουν προσφορές, έτσι ώστε να δώσουμε στον κάθε `marker` το κατάλληλο χρώμα.

```
public void drawMarker(LatLng point) {

    //Log.d("drawMarker", "draw Markers on the map!");

    // Creating an instance of MarkerOptions

    MarkerOptions markerOptions = new MarkerOptions();

    // Setting latitude and longitude for the marker

    markerOptions.position(point);

    // Adding marker on the Google Map

    if (offer || space) {

                                                //blue pin for
offers
        mMap.addMarker(markerOptions).setIcon(BitmapDescriptorFactory.fromResource(R.drawable.pin_offstreet_open));

    } else {
```

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

```
//else                red                pin
    mMap.addMarker(markerOptions).setIcon(BitmapDescriptorFactory.fromResource(R.drawable.pin_offstreet_high));
}
}
```

Επίσης στην `initView()` ελέγχουμε τα `SharedPreferences` που ορίσαμε στην `MainActivity` για να ορίσουμε εκ νέου το `CameraPosition`. Αν δεν είχαμε δώσει συγκεκριμένη διεύθυνση τότε το `CameraPosition` δείχνει την τωρινή τοποθεσία του χρήστη. Επιπλέον πατώντας σε ένα `marker` καλείται η μέθοδος `onMarkerClick(Marker marker)` η οποία θα εμφανίσει ένα παράθυρο με τις πληροφορίες του συγκεκριμένου `marker` (όνομα, διεύθυνση και αν υπάρχουν : προσφορές, θέσεις). Όσο είναι ανοιχτό το παράθυρο, αν το επιλέξουμε ανοίγει τη νέα `FragmentActivity InfoDetails`. Από το `SharedPreferences` έχουμε διαθέσιμα όλα τα δεδομένα οπότε εμφανίζονται οι πληροφορίες στο `layout`.

Αν θελήσει ο χρήστης να βρει πώς να φθάσει στο συγκεκριμένο `parking`, επιλέγει το `button "Get Directions"` και ανοίγει την κλάση `DirectionsMap`. Αρχικά, ελέγχει για την σύνδεση στο `Google Play`:

```
int status = GooglePlayServicesUtil

    .isGooglePlayServicesAvailable(getBaseContext());
```

και αν `status == ConnectionResult.SUCCESS` κάνουμε την αρχικοποίηση

```
// Initializing

mMarkerPoints = new ArrayList<LatLng>();

// Getting reference to SupportMapFragment of the activity_main

SupportMapFragment fm = (SupportMapFragment)
getSupportFragmentManager()
```

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

```
.findFragmentById(R.id.map);

// Getting Map for the SupportMapFragment

mGoogleMap = fm.getMap();

// Enable MyLocation Button in the Map

mGoogleMap.setMyLocationEnabled(true);

// Getting LocationManager object from System Service

// LOCATION_SERVICE

LocationManager locationManager = (LocationManager)
getSystemService(LOCATION_SERVICE);

// Creating a criteria object to retrieve provider

Criteria criteria = new Criteria();

// Getting the name of the best provider

String provider = locationManager.getBestProvider(criteria,
true);

// Getting Current Location From GPS

Location location = locationManager.getLastKnownLocation(provider);

if (location != null) {

    locationManager.onLocationChanged(location);

    startPoint = new LatLng(location.getLatitude(),
location.getLongitude());
```

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κασίκη

```
}
```

Αν το `location` δεν είναι `null` τότε το `CameraPosition` θα έχει ως `target` (όπως είδαμε πριν στον κώδικα) το `startpoint`. Εφόσον άρουμε τις συντεταγμένες του προορισμού από το `SharedPreferences` καλούμε το `drawMarker(point)`.

```
private void drawMarker(LatLng point) {

    mMarkerPoints.add(point);

    // Creating MarkerOptions

    MarkerOptions options = new MarkerOptions();

    // Setting the position of the marker

    options.position(point);

    /**
     * For the start location, the color of marker is GREEN
     * and for the end
     * location, the color of marker is RED.
     */

    if (mMarkerPoints.size() == 1) {

        options.icon(BitmapDescriptorFactory
            .defaultMarker(BitmapDescriptorFactory.HUE_GREEN));

    } else if (mMarkerPoints.size() == 2) {

        options.icon(BitmapDescriptorFactory
            .defaultMarker(BitmapDescriptorFactory.HUE_RED));

    }
}
```

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κασίκη

```
// Add new marker to the Google Map Android API V2  
mGoogleMap.addMarker(options);  
  
}
```

Έπειτα, ελέγχουμε αν υπάρχουν και τα δύο σημεία για να σχεδιάσουμε την διαδρομή:

```
if (mMarkerPoints.size() >= 2) {  
  
    LatLng origin = mMarkerPoints.get(0);  
    LatLng dest = mMarkerPoints.get(1);  
  
    // Getting URL to the Google Directions API  
    String url = getDirectionsUrl(origin, dest);  
    DownloadTask downloadTask = new DownloadTask();  
  
    // Start downloading json data from Google Directions  
    // API  
    downloadTask.execute(url);  
  
}
```

```
private String getDirectionsUrl(LatLng origin, LatLng dest) {  
  
    // Origin of route  
    String str_origin = "origin=" + origin.latitude + ","  
        + origin.longitude;
```

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

```
// Destination of route

String str_dest = "destination=" + dest.latitude + "," +
dest.longitude;

// Sensor enabled

String sensor = "sensor=false";

// Building the parameters to the web service

String parameters = str_origin + "&" + str_dest + "&" +
sensor;

// Output format

String output = "json";

// Building the url to the web service

String url =
"https://maps.googleapis.com/maps/api/directions/"
+ output + "?" + parameters;

return url;
}

/** A method to download json data from url */
private String downloadUrl(String strUrl) throws IOException {
    Log.d("downloadUrl", " A method to download json data
from url");
```

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

```
String data = "";

InputStream iStream = null;

URLConnection urlConnection = null;

try {

    URL url = new URL(strUrl);

    // Creating an http connection to communicate with url

    urlConnection = (URLConnection)
url.openConnection();

    // Connecting to url

    urlConnection.connect();

    // Reading data from url

    iStream = urlConnection.getInputStream();

    BufferedReader br = new BufferedReader(new
InputStreamReader(iStream));

    StringBuffer sb = new StringBuffer();

    String line = "";

    while ((line = br.readLine()) != null) {

        sb.append(line);

    }

    data = sb.toString();
```


Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

```
        br.close();

    } catch (Exception e) {

    } finally {

        iStream.close();

        urlConnection.disconnect();

    }

    return data;

}

/** A class to download data from Google Directions URL */

private class DownloadTask extends AsyncTask<String, Void,
String> {

    // Downloading data in non-ui thread

    @Override

    protected String doInBackground(String... url) {

        // For storing data from web service

        String data = "";

        try {

            // Fetching the data from web service

            data = downloadUrl(url[0]);

        } catch (Exception e) {

            Log.d("Background Task", e.toString());

        }

    }

}
```

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

```
    }

    return data;
}

// Executes in UI thread, after the execution of
// doInBackground()

@Override

protected void onPostExecute(String result) {

    super.onPostExecute(result);

    Log.d("onPostExecute", " Executes in UI thread,
after the execution of doInBackground()");

    ParserTask parserTask = new ParserTask();

    // Invokes the thread for parsing the JSON data

    parserTask.execute(result);

}

}

/** A class to parse the Google Directions in JSON format */

private class ParserTask extends

        AsyncTask<String, Integer,

List<List<HashMap<String, String>>>> {

    // Parsing the data in non-ui thread

    @Override
```

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κασίκη

```
        protected List<List<HashMap<String, String>>>
doInBackground(
    String... jsonData) {
    JSONObject jObject;
    List<List<HashMap<String, String>>> routes = null;

    try {
        jObject = new JSONObject(jsonData[0]);
        DirectionsJSONParser parser = new
DirectionsJSONParser();

        // Starts parsing data
        routes = parser.parse(jObject);
    } catch (Exception e) {
        e.printStackTrace();
    }

    return routes;
}

// Executes in UI thread, after the parsing process
@Override
protected void onPostExecute(List<List<HashMap<String,
String>>> result) {

    ArrayList<LatLng> points = null;
    PolylineOptions lineOptions = null;
```

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

```
// Traversing through all the routes

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

    points = new ArrayList<LatLng>();

    lineOptions = new PolylineOptions();

    // Fetching i-th route

    List<HashMap<String, String>> path =

result.get(i);

    // Fetching all the points in i-th route

    for (int j = 0; j < path.size(); j++) {

        HashMap<String, String> point =

path.get(j);

        double lat =

Double.parseDouble(point.get("lat"));

        double lng =

Double.parseDouble(point.get("lng"));

        LatLng position = new LatLng(lat, lng);

        points.add(position);

    }

    // Adding all the points in the route to

LineOptions

    lineOptions.addAll(points);

    lineOptions.width(2);

    lineOptions.color(Color.RED);
```

```
    }  
  
    // Drawing polyline in the Google Map for the i-th  
route  
  
    mGoogleMap.addPolyline(lineOptions);  
    }  
}
```

5.4 Επίλογος

Στο κεφάλαιο αυτό κάναμε των κλάσεων της εφαρμογής εξηγώντας τι περιλαμβάνει η κάθε μια και τι λειτουργίες παρέχουν οι μεθόδους της. Στη συνέχεια περιγράψαμε την σχεδίαση της εφαρμογής η οποία παρουσιάζεται και στις εικόνες. Τέλος αναλύσαμε βήμα προς βήμα την υλοποίηση της εφαρμογής με την προσθήκη ενός μέρους του κώδικα που αντιστοιχεί στο κάθε βήμα. Καθώς κάθε εφαρμογή θέλει να προσελκύει περισσότερους χρήστες πρέπει να προσφέρει υπηρεσίες που να την κάνουν να ξεχωρίζει από τις υπόλοιπες. Οι καινοτομίες αυτές που περιλαμβάνει η εφαρμογή θα αναλυθούν στο επόμενο κεφάλαιο συμπεριλαμβανομένου και των χαρακτηριστικών.

ΚΕΦΑΛΑΙΟ 6: Χαρακτηριστικά Καινοτόμων και

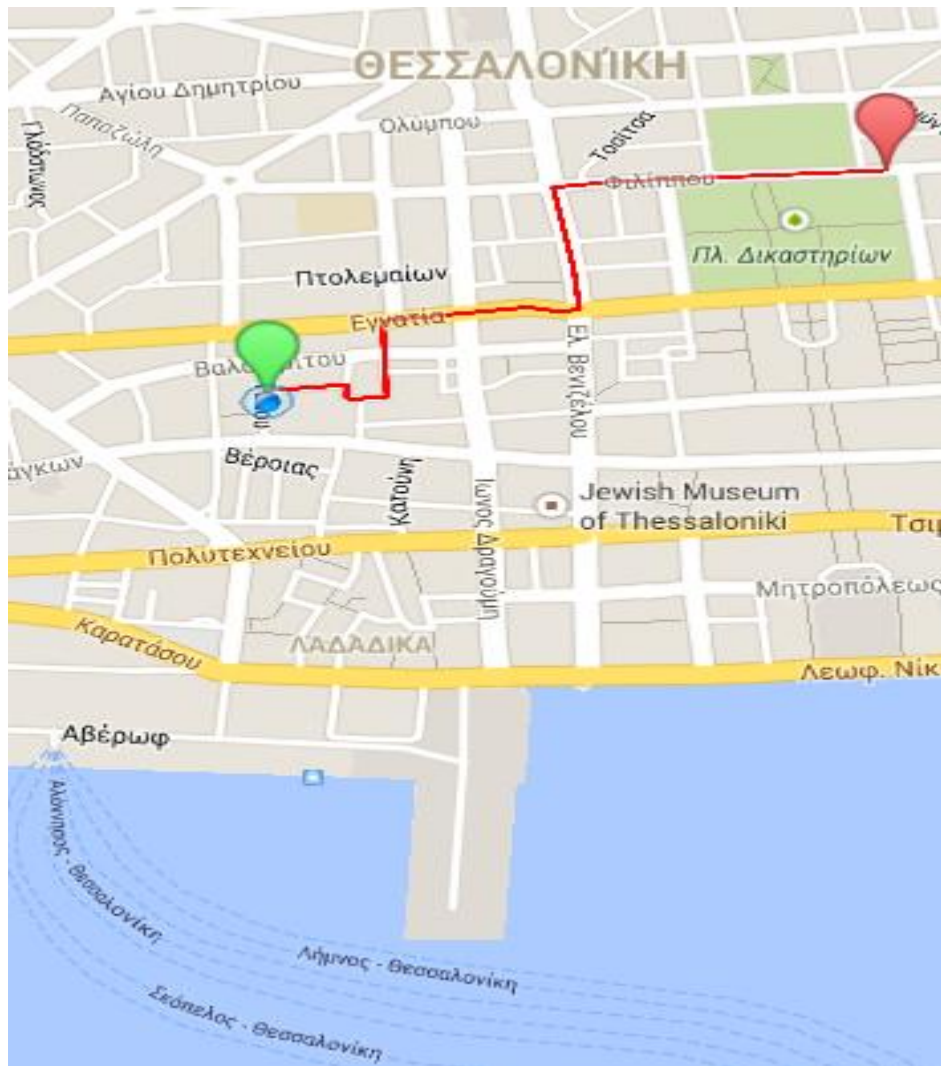
Προστιθέμενων που έχει η Υπηρεσία

Στο τρίτο κεφάλαιο αναλύσαμε εφαρμογές με παραπλήσιες υπηρεσίες και τις αξιολογήσαμε με βάση τα κριτήρια που αναφέραμε αναλυτικά στο δεύτερο κεφάλαιο. Η διαδικτυακή υπηρεσία που αναπτύξαμε βασίστηκε πάνω στα χαρακτηριστικά και στα πρότυπα αξιοπιστίας που αναλύσαμε στο δεύτερο κεφάλαιο με αποτέλεσμα να δημιουργηθεί μια υπηρεσία που να παρέχει διαλειτουργικότητα, ευκολία στην πρόσβαση και διαχείριση με δυνατότητες βελτίωσης και επεκτασιμότητας.

Ο ανταγωνισμός όμως στο τομέα των διαδικτυακών υπηρεσιών έχει αυξηθεί πολύ τα τελευταία χρόνια καθώς ο κόσμος χρησιμοποιεί όλο και περισσότερο το διαδίκτυο καθότι είναι πιο εύκολο μέσο για την ικανοποίηση των διαφόρων αναγκών τους όπως την πληρωμή λογαριασμών. Αυτό έχει ως αποτέλεσμα να δημιουργούνται όλο και περισσότερες υπηρεσίες με σκοπό να καλύψουν τις ανάγκες αυτές. Ως αποτέλεσμα για κάθε υπάρχει μια πληθώρα υπηρεσιών που θα μπορούσαν να χρησιμοποιηθούν για τον ίδιο σκοπό. Ποια όμως θα προσελκύσει τους περισσότερους χρήστες; Πώς θα ξεχωρίσει από τις υπόλοιπες; Η απάντηση είναι απλή · προσφέροντας κάτι που δεν έχει καμία άλλη υπηρεσία στην αγορά: μια καινοτόμος ιδέα.

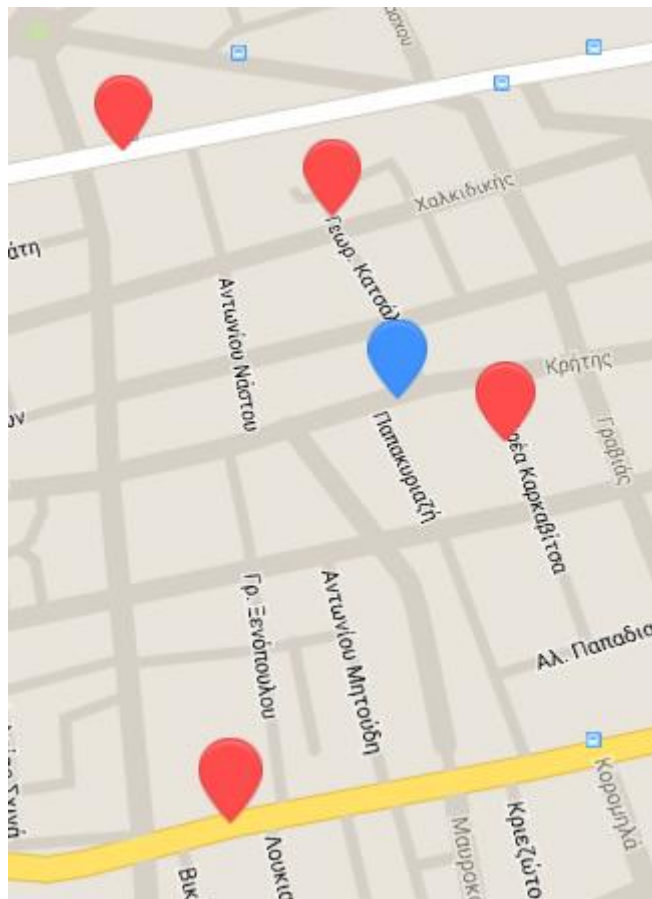
Έχοντας ως στόχο να δημιουργήσουμε μια εφαρμογή που να ικανοποιεί μια ανάγκη των χρηστών αλλά και να ξεχωρίζει από τις υπόλοιπες προσθέσαμε κάποια χαρακτηριστικά που δεν υπήρχαν στις υπάρχουσες εφαρμογές.

Το πρώτο είναι η δυνατότητα που έχουν οι χρήστες να βρουν οδηγίες σχετικά με το τι κατεύθυνση να πάρουν για να φθάσουν στο parking της επιλογής τους. Αφού έχουν επιλέξει ένα parking και άνοιξαν να δουν τις πληροφορίες του, μπορούν να επιλέξουν να δουν την διαδρομή η οποία ξεκινάει από την τοποθεσία που βρίσκονται με προορισμό το parking.



Εικόνα 29: Διαδρομή χρήστη - Πράσινο η αφετηρία, Κόκκινο για τον προορισμό

Το δεύτερο χαρακτηριστικό είναι η γρήγορη και άμεση ενημέρωση της διαδικτυακής υπηρεσίας για τις προσφορές που παρέχει ο κάθε ιδιοκτήτης parking καθώς και τον αριθμό των διαθέσιμων θέσεων κατά την διάρκεια της λειτουργίας ενός parking. Με αυτόν τον τρόπο ο χρήστης μπορεί οποιαδήποτε στιγμή να ψάξει για τα parking που έχουν προσφορές σε κοντινή απόσταση από την περιοχή που βρίσκεται και να ελέγξει επίσης αν υπάρχουν άδειες θέσεις. Στην περίπτωση που δεν υπάρχουν θα μπορούσε έγκαιρα να ψάξει για άλλο parking και να συγκρίνει τιμές και/ή προσφορές, έτσι ώστε να επιλέξει όποιο έχει την πιο προσιτή τιμή.



Εικόνα 30: Τα markers που έχουν χρώμα μπλε, δείχνουν τα parking με προσφορές

Οι εξελίξεις στον τομέα των διαδικτυακών υπηρεσιών δεν σταματούν καθόλου. Ακόμα κι αν αυτή την στιγμή η υπηρεσία παρέχει κάτι διαφορετικό και μοναδικό δεν σημαίνει ότι δεν θα υπάρξει κάποια άλλη που ίσως περιέχει τα ίδια χαρακτηριστικά και πιθανόν ακόμα περισσότερα. Μια διαδικτυακή υπηρεσία για να συνεχίσει να είναι καλή πρέπει να εξελίσσεται συνεχώς. Με το να είναι ανταγωνιστική μπορεί να παραμείνει κερδοφόρα στην αγορά.

ΚΕΦΑΛΑΙΟ 7: Μελλοντική Επέκταση – Συμπεράσματα

Η διαδικτυακή υπηρεσία μπορεί να βελτιωθεί, με την προσθήκη δικτυακής λειτουργίας για τις ισχύουσες προσφορές. Αυτήν την στιγμή η λειτουργία επιτυγχάνεται με την χρήση του Wamp Server το οποίο όμως χρησιμοποιεί το localhost με αποτέλεσμα οι ενημερώσεις να φαίνονται τοπικά, δηλαδή μόνο στις συσκευές που είναι συνδεδεμένες στο ίδιο δίκτυο. Επομένως, με την χρήση ενός server όλοι οι χρήστες, ανεξαρτήτου δικτύου, θα μπορούν να βλέπουν άμεσα τις νέες ενημερώσεις που αφορούν τις προσφορές και τις διαθέσιμες θέσεις σε κάθε parking. Η διαδικασία αυτή έχει περιγραφεί αναλυτικά στο κεφάλαιο 6.3.

Μια ακόμη επέκταση της εφαρμογής θα ήταν αν προσθέταμε τις θέσεις parking που είναι διαθέσιμες στους δρόμους της πόλης και όχι μόνο των ιδιωτικών parking. Θα έπρεπε να έχουμε σύστημα peer-to-peer. Αν ένας χρήστης πρόκειται να μετακινήσει το αυτοκίνητό του τότε ενημερώνει μέσω της εφαρμογής την τοποθεσία που βρίσκεται το αμάξι και την ώρα που θα φύγει (π.χ. οδός Κασσάνδρου 30, 15:00), έτσι ώστε κάποιος χρήστης που ψάχνει θέση για να παρκάρει σε εκείνη την περιοχή, βλέπει το σήμα στον χάρτη που να κατευθυνθεί.

Σύμφωνα με όσα περιγράφηκαν στην εργασία αυτή, συμπεραίνεται ότι το λειτουργικό σύστημα Android είναι εύχρηστο, ευέλικτο και αξιόπιστο για έξυπνες συσκευές. Επιπροσθέτως, είναι ένα λογισμικό ανοιχτού κώδικα και δίνει στους προγραμματιστές την δυνατότητα να υλοποιήσουν εφαρμογές οι οποίες μπορούν να τους αποφέρουν κέρδος εφόσον μπορούν να εισαχθούν εύκολα στην αγορά εργασίας.

Επισημαίνεται ότι στις μεγάλες πόλεις, όπως η Θεσσαλονίκη, είναι αρκετά βολικό να υπάρχουν διαδικτυακές υπηρεσίες που να τους ενημερώνουν για τα μέρη που μπορούν να παρκάρουν καθώς και το κόστος που τους αναλογεί. Η παρούσα διαδικτυακή υπηρεσία θα μπορούσε να χρησιμοποιηθεί σε οποιαδήποτε πόλη στον κόσμο, εφόσον το μόνο στοιχείο που πρέπει να αλλάξει είναι η βάση δεδομένων. Αυτό επιτυγχάνεται με την

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

προσθήκη των δεδομένων που αντιστοιχούν στην εκάστοτε πόλη, περιοχή ή δήμο, πράγμα που την καθιστά ευέλικτη.

Τέλος, μπορούμε να πούμε πως η σχεδίαση και ανάπτυξη μιας εφαρμογής και γενικότερα μιας διαδικτυακής υπηρεσίας αποτελείται από κύκλους βελτιώσεων. Σε κάθε νέα έκδοση μιας εφαρμογής θα μπορούσαν να έχουν διορθωθεί λάθη και προβλήματα που υπήρχαν αλλά δεν ήταν φανερά εξ αρχής, καθώς επίσης και βελτίωση των υπάρχόντων υπηρεσιών/χαρακτηριστικών της εφαρμογής. Επιπλέον με την προσθήκη νέων ιδεών, κατά προτίμηση να μην έχουν χρησιμοποιηθεί από άλλους, η διαδικτυακή υπηρεσία μπαίνει και πάλι στη φάση της ανάπτυξης.

ΒΙΒΛΙΟΓΡΑΦΙΑ ΚΑΙ ΑΝΑΦΟΡΕΣ

- I. Ηλιούδης Χρήστος, 2013, [Υπηρεσιοστρεφής Αρχιτεκτονική](#), ΑΤΕΙ
- II. Τ. Πιλιούρα, Α. Τσαλγατίδου, Σ. Χατζηευθυμιάδης, 2003, [Scenarios of using Web Services in M-Commerce](#), Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
- III. PLAN A.E., 2012, [Ανάπτυξη Καινοτόμων Υπηρεσιών Προστιθέμενης Αξίας](#)
- IV. Wikipedia, 2013, [Καινοτομία](#)
- V. 2004, [Characteristics of Mobile Applications](#)
- VI. Donn Felker, 2011, Android Application Development For Dummies
- VII. Darcey lauren, Conder Shane, 2012, Sams Teach Yourself Android Application Development in 24 hours, Δεύτερη Έκδοση
- VIII. Google Inc., Android Developer, 2014, [ADT Plugin](#)
- IX. Google Inc., Android Developers, 2014, [Google Play Services](#)
- X. Google Inc., Android Developers, 2014, [Google Play Services - Set up](#)
- XI. Google Inc., Android Developers, 2014, [SQLiteDatabase](#)
- XII. hmkcode.com, 14/11/2014, [Add Google Play Services Library](#)
- XIII. Investopedia, 2013, [Mobile Commerce](#)
- XIV. It Technical Solutions, 2012, [Some Essential Characteristics of Mobile Applications](#)
- XV. Josiah Humphrey, 2012, Appster, [The Six Characteristics of the Top 100 Mobile Apps of All Time](#)
- XVI. JSON official website, 2014, [JSON.org](#)
- XVII. Lars Vogel, 2010-2014, [Android SQLite database and content provider](#)
- XVIII. Lauren Darcey, Shane Conder, 2012, Sams Teach Yourself Android Development in 24 hours
- XIX. PLYNT, 2011, [Plynt Certification Criteria for Mobile Applications](#)
- XX. Roger Wolter, 2001, [XML Web Services Basics](#) , MSDN
- XXI. Roland Koch, 2011, [Six Major Characteristics of Mobile Apps](#) , flamelab.de
- XXII. The Verge, 2011, [Android History](#)
- XXIII. Vogel Lars, 2014, [Android SQLite Database](#)

Πτυχιακή εργασία της φοιτήτριας Μάρθας Κατσίκη

- XXIV. W3C, W3C Working Group, 2004, [Web Services Architecture](#)
- XXV. Wikipedia, 2007-2009, [Android](#)
- XXVI. Wikipedia, 2014, [Eclipse](#)
- XXVII. Wikipedia, 2014, [Google Play Services](#)
- XXVIII. Wikipedia, 2014, [JSON](#)
- XXIX. Wikipedia, 2010-2013, [Mobile Commerce](#)
- XXX. Wikipedia, 2009-2014, [SQLite](#)
- XXXI. Wikipedia, 2014, [Splash Screen](#)