Ατομική Διπλωματική Εργασία

“ΥΛΟΠΟΙΗΣΗ ΚΙΝΗΤΗΣ ΕΦΑΡΜΟΓΗΣ ΕΞΥΠΝΗΣ ΣΤΑΘΜΕΥΣΗΣ”

Θεόδωρος Γεωργίου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

"Υλοποίηση Κινητής Εφαρμογής Έξυπνης Στάθμευσης"

Θεόδωρος Γεωργίου

Επιβλέπων Καθηγητής

Βάσος Βασιλείου

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

Μάιος 2016

Περιεχόμενα

Ευχαριστίες

Περίληψη

Κεφάλαιο 1 : Εισαγωγή 1

1.1 Βασικό Υπόβαθρο 1

1.2 Υποκίνηση Εργασίας 3

1.3 Δομή της διπλωματικής εργασίας 4

Κεφάλαιο 2 : Σχετική δουλειά και προτιμήσεις 6

2.1 Διάφορες εφαρμογές και μοναδικά χαρακτηριστικά 6

2.2 Τρόποι βελτίωσης εφαρμογής 12

Κεφάλαιο 3 : Περιγραφή της ιστοσελίδας Spark 16

3.1 Επισκόπηση Αρχιτεκτονικής Spark ιστοσελίδας 16

3.2 Υλικό που χρησιμοποιήθηκε 17

3.3 Λειτουργίες Συστήματος 18

Κεφάλαιο 4 : Περιγραφή της εφαρμογής Spark 22

4.1 Επισκόπηση Αρχιτεκτονικής Spark εφαρμογής 22

4.2 Υλικό που χρησιμοποιήθηκε 23

4.3 Λειτουργίες Εφαρμογής 24

4.4 Wireframe και Flowchart εφαρμογής 35

Κεφάλαιο 5 : Υλοποίηση της εφαρμογής Spark 40

5.1 Σύνδεση στο Server 40

5.2 GUI εφαρμογή Spark 43

5.3 Το λειτουργικό σύστημα Android 46

5.4 Χάρτες Google 47

Κεφάλαιο 6 : Συμπεράσματα 49

6.1 Συμπεράσματα και Μελλοντικές Επεκτάσεις 49

Βιβλιογραφία 50

Παράρτημα Α : Πίνακας με μοναδικά χαρακτηριστικά της κάθε εφαρμογής 52

Παράρτημα Β : Spark app source code 56

Παράρτημα Γ : Server source code 72

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή της παρούσας διπλωματικής εργασίας, κύριο Βάσο Βασιλείου, για την ευκαιρία που μου προσέφερε μιας και μέσα από την καθοδήγησή του αλλά και με τον εξοπλισμό που μου παρείχε κατάφερα να εργαστώ εποικοδομητικά και να φέρω εις πέρας την εν λόγω διπλωματική εργασία.

Επίσης, οφείλω ένα τεράστιο ευχαριστώ στην οικογένειά μου και ιδιαίτερα στους γονείς μου που με στήριξαν ηθικά και οικονομικά καθ' όλη τη διάρκεια των σπουδών μου.

Περίληψη

Κύρια προτεραιότητα της διπλωματικής μου εργασίας είναι η αναφορά των καθημερινών προβλημάτων των οδηγών κατά τη χρήση μηχανοκίνητων οχημάτων αλλά και τη στάθμευση τους. Επίσης θα σας παρουσιάσω μια εφαρμογή που κατά την άποψη μου θα επιλύσει μεγάλο μέρος από αυτά τα προβλήματα που παρουσιάζονται λόγω της κεκτημένης ταχύτητας που επικρατεί σήμερα.

Στις μέρες μας απαιτείται να είμαστε αυτόνομοι, να είμαστε σε θέση να προλάβουμε τις υποχρεώσεις μας εγκαίρως και η πολυτέλεια που ήταν το αυτοκίνητο σε παλαιότερες εποχές σήμερα αποτελεί ανάγκη για όλους μας. Ως αποτέλεσμα αυτού, υπήρξε μειωμένη χρήση μέσων μαζικής μεταφοράς και αυξημένη χρήση αυτοκινήτων στους δρόμους. Ο ολοένα και μεγαλύτερος αριθμός αυτοκινήτων που παρατηρείται καθημερινώς δημιουργεί αρκετά προβλήματα στους οδηγούς κατά τη διακίνηση αλλά και στάθμευση του οχήματος τους. Πιο κάτω θα δούμε μερικά παραδείγματα.

Κάποια από αυτά τα προβλήματα είναι η κυκλοφοριακή συμφόρηση στους δρόμους, είτε λόγω οδικών έργων είτε λόγω κάποιου ατυχήματος. Αυτό προκαλεί δυσανασχέτηση από τους οδηγούς και κατ' επέκταση ένταση λόγω καθυστέρησης υλοποίησης των υποχρεώσεων τους. Ακόμη, η λανθασμένη επιλογή χώρου στάθμευσης από τους οδηγούς οφείλεται στην άγνοιά τους αν πράγματι υπάρχει διαθέσιμος χώρος στάθμευσης προκαλώντας δυσκολία στη διέλευση άλλων οχημάτων αφού σταθμεύουν σε πεζοδρόμια, ακόμη και κατά μήκος του δρόμου. Επιπρόσθετα, ο οδηγός μη γνωρίζοντας εκ των προτέρων τους διαθέσιμους χώρους αναγκάζεται να το σταθμεύσει σε μεγάλο, σίγουρο κατά την γνώμη του, χώρο που όμως απαιτεί καταβολή χρηματικού ποσού. Επομένως, η επιλογή χώρου, είτε μακριά από τον προορισμό του για να εξοικονομήσει χρήματα, είτε κοντά στον προορισμό του αλλά με κόστος, που αρκετές φορές είναι μεγάλο, εναπόκειται στον κάθε οδηγό ο οποίος θα θέσει τα κριτήρια επιλογής του.

Καταλήγοντας θα σας αναφέρω μια κινητή εφαρμογή η οποία θα μας παρέχει απαραίτητες πληροφορίες ανά πάσα στιγμή στον κάθε οδηγό για την εύρεση του καλύτερου χώρου στάθμευσης. Με τη χρήση της εφαρμογής θα μπορεί εισάγοντας αρχικά τον προορισμό του να αναζητήσει τους διαθέσιμους χώρους στάθμευσης μειώνοντας έτσι τις πιθανότητες έκδοσης εξωδίκων καθώς και την παράνομη στάθμευση σε χώρους αναπήρων.

Κεφάλαιο 1

Εισαγωγή

1.1 Βασικό Υπόβαθρο 1

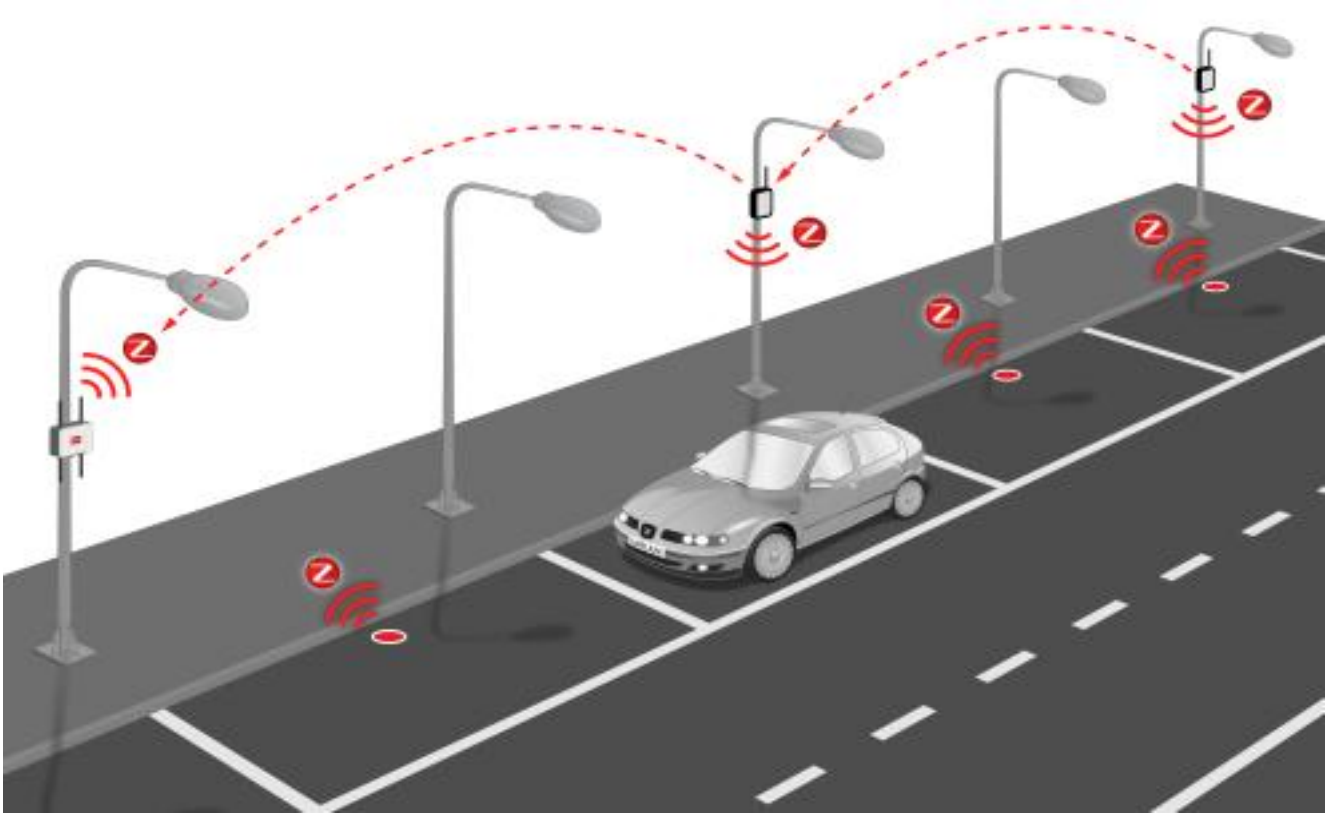
1.2 Υποκίνηση Εργασίας 3

1.4 Δομή της διπλωματικής εργασίας 4

Στο κεφάλαιο αυτό θα αναφερθούμε στο βασικό υπόβαθρο της έξυπνης στάθμευσης εξηγώντας τους λόγους που οδήγησαν στη δημιουργία αυτής της διπλωματικής εργασίας και με ποιο τρόπο θα αντιμετωπίσει ο κάθε οδηγός τα οδικά προβλήματα. Τέλος, θα σχολιάσουμε εν συντομία τι περιέχει το κάθε κεφάλαιο.

1.1 Βασικό Υπόβαθρο

Με τον όρο «έξυπνη στάθμευση» εννοούμε κυρίως ένα σύστημα στάθμευσης αυτοκινήτων που βοηθά τους οδηγούς να βρουν μια κενή θέση στάθμευσης. Αυτό μπορεί να γίνει χρησιμοποιώντας αισθητήρες σε κάθε χώρο στάθμευσης που ανιχνεύουν την παρουσία ή την απουσία ενός οχήματος (σχήμα 1.1.1). Με τον τρόπο αυτό μπορούμε να μετρήσουμε πόσες θέσεις είναι καταλυμένες και να υπολογίσουμε πόσες ακόμη θέσεις παραμένουν ελεύθερες. Σε περιπτώσεις σαν και αυτές, εγκαθίσταται ένας μετρητής έξω από το παρκινγκ για να μπορούν να βλέπουν οι οδηγοί κατά την είσοδό τους την ηλεκτρονική πινακίδα που θα αναγράφει τον αριθμό των κενών θέσεων ώστε να αποφεύγουν τους άσκοπους γύρους στο παρκινγκ ψάχνοντας για ελεύθερο χώρο.



Σχήμα 1.1.1 : Σύστημα στάθμευσης με ασύρματους αισθητήρες [1]

Επίσης, μπορεί ακόμη να είναι μια μηχανική συσκευή που δίνει την δυνατότητα στάθμευσης σε πολλαπλά αυτοκίνητα χρησιμοποιώντας μόνο μια θέση αποθήκευσης (σχήμα 1.1.2). Έτσι, καταφέρνουμε να εξοικονομήσουμε χώρο και προστατεύουμε το αυτοκίνητό μας από διάφορες καιρικές συνθήκες αφού η συσκευή αυτή μπορεί να παρέχει κάλυψη και ασφάλεια στα οχήματα μας.



Σχήμα 1.1.2 : Μηχανική συσκευή αποθήκευσης πολλαπλών αυτοκινήτων [2]

Επιπλέον, «έξυπνη» μπορεί να θεωρηθεί και για τον λόγο ότι μπορεί κανείς να προμελετήσει, να πάρει πληροφορίες από νωρίς και να είναι ενήμερος για τον χώρο που θα παρκάρει το αυτοκίνητό του έτσι ώστε να είναι βέβαιος ότι θα είναι κοντά στον προορισμό του. Όλο αυτό θα γίνει με το ελάχιστο κόστος. Αν και το "Smart Parking" έχει αναπτυχθεί και διεισδύσει στον κόσμο των έξυπνων κινητών συσκευών, η εφαρμογή του στην καθημερινότητά μας ίσως να μην έχει γίνει ακόμη αισθητή λόγο μειωμένης ανάπτυξης στο τομέα αυτό ή είτε και λόγο οικονομικών προβλημάτων που παρουσιάζονται στη χώρα μας.

Η «έξυπνη» στάθμευση είναι ένας από τους τομείς που θα έχει ως στόχο να αναπτύξει μια έξυπνη πόλη. Κάποιοι άλλοι τομείς είναι για παράδειγμα τα κτήρια, η ενέργεια, η ασφάλεια, η γνώση, η υγεία, τα κοινωνικά προγράμματα και πολλοί άλλοι. Λόγω της ραγδαίας εξέλιξης και του πληθωρισμού στις πόλεις η εξέλιξη της έννοιας «έξυπνη πόλη» πρέπει να έρθει στο προσκήνιο. Σήμερα οι άνθρωποι είναι πιο ενεργοί στην τεχνολογία (π.χ. Smartphone) και μπορούν να πάρουμε οποιαδήποτε πληροφορία εύκολα και γρήγορα. Το γεγονός αυτό πρέπει να το εκμεταλλευτούμε και να αναπτύξουμε ιδέες που θα κάνουν τη ζωή μας πιο εύκολη.

1.2 Υποκίνηση Εργασίας

Στόχος λοιπόν της παρούσας διπλωματικής εργασίας είναι να αναπτύξει μια κινητή εφαρμογή, εύκολη στη χρήση που να αντιμετωπίσει το πρόβλημα αναζήτησης χώρου στάθμευσης και να μειώσει την άσκοπη κυκλοφοριακή κίνηση στους δρόμους μας. Καταφέρνοντας αυτό θα μπορέσουμε να μειώσουμε επίσης την ρύπανση του περιβάλλοντος από τα καυσαέρια των αυτοκινήτων, ή ακόμη τα ατυχήματα που συμβαίνουν κατά την αναζήτηση χώρου στάθμευσης.

Η εφαρμογή θα επιτρέπει στον κάθε οδηγό να αναζητήσει τον προορισμό του και να εμφανίζει σε αυτόν τους διαθέσιμους χώρους που υπάρχουν σε συγκεκριμένη ακτίνα από το κέντρο αναζήτησής του.

Η αναζήτηση θα γίνεται με βάση τις προσωπικές προτιμήσεις του κάθε οδηγού, π.χ. με το πόσο κοντά είναι στον τελικό προορισμό, με το λιγότερο κόστος κ.ο.κ.

Ακόμη, θα μπορεί να ψάξει για τα διαθέσιμους χώρους από το σημείο που βρίσκεται εκείνην την στιγμή.

Την κράτηση του επιθυμητού χώρου θα μπορεί κανείς τη διευθετήσει με ηλεκτρονικό τρόπο μέσω της εφαρμογής εξασφαλίζοντας προσφορές στο συνολικό ποσό. Επιπρόσθετα, θα έχει τη δυνατότητα να αποθηκεύει το σημείο που στάθμευσε το αυτοκίνητό του για να μπορεί να το εντοπίσει αργότερα σε περίπτωση που έχει ξεχάσει που το στάθμευσε.

Επιπλέον, οι διαθέσιμοι χώροι που θα εμφανίζονται στον οδηγό θα βρίσκονται στη βάση δεδομένων της εφαρμογής που θα είναι μερικώς προσβάσιμη από τους χρήστες της. Δηλαδή, ο κάθε οδηγός θα έχει μόνο την δυνατότητα να εισάγει (και όχι να διαγράφει η να διαχειρίζεται) ένα διαθέσιμο σημείο στάθμευσης και να ενημερώνει τη βάση σε περίπτωση που δεν υπήρχε μέχρι τώρα το σημείο αυτό ήδη μέσα. Με τον τρόπο αυτό και την συμβολή των χρηστών της εφαρμογής, η βάση θα αυξάνεται σιγά-σιγά με στόχο την βελτίωση της ποιότητάς της και την καλύτερη εξυπηρέτηση των πελατών της.

1.3 Δομή της διπλωματικής εργασίας

Η παρούσα διπλωματική αποτελείται από 6 κεφάλαια:

Στο κεφάλαιο 1 παρουσιάζεται η έννοια της έξυπνης στάθμευσης και μερικά παραδείγματα από τι υπάρχει είδη υλοποιημένο σήμερα σχετικά με το τομέα αυτό. Επίσης , αναφέρεται η προσφορά της παρούσας διπλωματικής εργασίας όπως και μερικές εφαρμογές και μελέτες που έχουν γίνει σχετικά με την έξυπνη στάθμευση.

Στο κεφάλαιο 2 παρουσιάζονται οι διαφορετικές εφαρμογές που υπάρχουν και τα μοναδικά χαρακτηριστικά τους , όπως επίσης και μια σύνοψη με το τι μπορεί να έχει μια εφαρμογή για να είναι ελκυστική στο κοινό.

Στο κεφάλαιο 3 γίνεται επισκόπηση της αρχιτεκτονικής της ιστοσελίδας Spark , το υλικό που χρησιμοποιήθηκε για την δημιουργία της ιστοσελίδας και τις λειτουργίες που παρέχει στο χρήστη.

Στο κεφάλαιο 4 παρουσιάζεται η αρχιτεκτονική της κινητής εφαρμογής Spark . Γίνεται εκτενής αναφορά στις λειτουργίες της εφαρμογής και στο τρόπο χρήσης τους. Ακολουθούν μέθοδοι wireframe και flowcharts που περιγράφουν την λειτουργία της εφαρμογής.

Στο κεφάλαιο 5 περιέχεται η υλοποίηση της κινητής εφαρμογής GUI όπως επίσης αναφορά στο λειτουργικό σύστημα android και χάρτες Google που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής.

Τέλος , στο κεφάλαιο 6 γίνεται μια σύνοψη της δουλειάς που έγινε, παρουσιάζονται τα συμπεράσματα και διάφορες δυσκολίες που αντιμετωπίστηκαν καθώς επίσης και μελλοντικές σκέψεις που αφορούν την συνέχιση αυτής της προσπάθειας.

Κεφάλαιο 2

Σχετική δουλειά και προτιμήσεις

2.1 Διάφορες εφαρμογές και μοναδικά χαρακτηριστικά 6
2.2 Τρόποι βελτίωσης εφαρμογής 12

Στο κεφάλαιο αυτό πρόκειται να μελετήσουμε τις διάφορες εφαρμογές που αναπτύχτηκαν για να εξυπηρετήσουν σκοπούς όπως η έξυπνη στάθμευση , και θα μελετήσουμε πώς αντιμετωπίζει η κάθε εφαρμογή το πρόβλημα και σε ποιο τομέα . Επίσης, θα δούμε τι μπορούμε να κάνουμε για να δημιουργήσουμε μια πρακτικά καλή αλλά και ελκυστική ως προς το κοινό κινητή εφαρμογή.

2.1 Διάφορες εφαρμογές και μοναδικά χαρακτηριστικά

“ Σε μια μελέτη ο οικονομολόγος και πολεοδόμος του πανεπιστημίου UCLA, Donald Shoup, διαπίστωσε ότι σε μια περιοχή 15-μπλοκ, η προσπάθεια για στάθμευση σε μέσο όρο καταναλώνει 950.000 περίσσεια μίλια του ταξιδιού, 47.000 γαλόνια φυσικού αερίου, 100000 ώρες και παράγει 730 τόνους αερίων του θερμοκηπίου διοξειδίου του άνθρακα ετησίως. Επιπλέον, οι πόλεις έχουν συστηματικά κακοδιαχείριση στάθμευσης στο δρόμο λόγο εκτεταμένης αναζήτησης για στάθμευση στο δρόμο. Αυτό το πρότυπο της υπερβολικής προσπάθειας για στάθμευση επαναλαμβάνεται σε κάθε πόλη σε ολόκληρο τον κόσμο ”.[3]

Όπως μπορούμε να συμπεράνουμε, το πρόβλημα στάθμευσης υπάρχει εδώ και καιρό. Αυτό που κάνει τη διαφορά σήμερα που μας δίνει την ελπίδα πως υπάρχει τρόπος για την αντιμετώπισή του, είναι η ενεργή συμμετοχή του κόσμου στο διαδίκτυο και στις γρήγορες εναλλαγές για προσαρμογή του στα νέα δεδομένα.

Έχουν δημιουργηθεί από τότε αρκετές εφαρμογές με κάποιες από αυτές να προσθέτουν κάτι περισσότερο ή κάτι μοναδικό. Μερικές από τις εφαρμογές από τις οποίες αναπτύχτηκαν για αντιμετώπιση αυτών των προβλημάτων και αξίζει να αναφερθούμε είναι:

Οι ακόλουθες εφαρμογές είναι διαθεμένες στο Google Play Store

a. Bring me Back

Με την εφαρμογή αυτή μπορείς να αποθηκεύσεις την τοποθεσία του αυτοκινήτου σου , του ξενοδοχείου σου ή ενός εστιατορίου της επιλογής σου για να την βρεις αργότερα. Επίσης, παρέχει τη δυνατότητα να μοιραστείς με τους φίλους σου τοποθεσίες μέσω email η SMS. Μπορεί να αναδιατάξει την διαδρομή αναζήτησης με μια νέα καλύτερη διαδρομή, όπως επίσης και φωνητική αναζήτηση και διαχείριση για να λύσει τα χέρια του οδηγού.

b. Car Finder AR

Το Car Finder AR (augmented reality) εντοπίζει αυτόματα που είχατε σταθμεύσει . Χρησιμοποιεί εικονική πραγματικότητα, 3D Χάρτες Google και ραντάρ για να βρείτε το σταθμευμένο αυτοκίνητό σας. Επίσης, χρησιμοποιεί ειδοποίηση σε περίπτωση που λήξει ο χρόνος στάθμευσης σας αποφεύγοντας έτσι την έκδοση εξωδίκων για παράνομη στάθμευση.

c. Park Easy

Η εφαρμογή αυτή δίνει την δυνατότητα να μοιράζεστε θέσεις στάθμευσης σας μόνο με άτομα που γνωρίζετε. Για παράδειγμα, τους φίλους και την οικογένεια σας. Επίσης, να βρείτε τα διαθέσιμα και τα φθηνότερα σημεία στάθμευσης όπως επίσης και αποστολή τιμολόγιου / πληρωμής μέσω Paypal.

d. Best Parking

Με την πιο πάνω εφαρμογή μπορείτε να βρείτε τις φθηνότερες και πιο βολικές εγκαταστάσεις στάθμευσης εξοικονομώντας αρκετά χρήματα. Επιπρόσθετα υπολογίζει τις συνολικές αμοιβές στάθμευσης, περιλαμβάνει περιεκτικά χαρακτηριστικά εγκατάστασης παρκινγκ όπως για παράδειγμα ώρες λειτουργίας, αριθμό τηλεφώνου ,

εσωτερική/εξωτερική εγκατάσταση, μετρητά / Πιστωτικές κλπ. Κουπόνια, εκπτώσεις και κρατήσεις για συνεργεία.

e. Parkopedia Parking

Parkopedia είναι ένας συνδυασμός των λέξεων στάθμευσης και εγκυκλοπαίδεια. Μπορείτε να βρείτε χώρους στάθμευσης χρησιμοποιώντας την τρέχουσα τοποθεσία σας ή εισάγοντας μια διεύθυνση. Να δείτε τη διαθεσιμότητα χώρου στάθμευσης σε πραγματικό χρόνο. Επίσης, μπορείτε να βρείτε τις ώρες λειτουργίας των χώρων στάθμευσης, τις τιμές σήμερα, τις μεθόδους πληρωμής. Τέλος, μπορείτε να περιορίσετε τις επιλογές στάθμευσης χρησιμοποιώντας φίλτρα όπως παρκινγκ στο δρόμο μόνο, δωρεάν παρκινγκ, δεκτές πιστωτικές κάρτες, κ.ο.κ. Αποτελεί ίσως από τις πιο ολοκληρωμένες εφαρμογές.

f. Parking Guru

Οπτικοποίηση της στάθμευσης (υπόγεια και μη), καθώς και σχετικές πληροφορίες τους. Το Parking Guru είναι διαθέσιμο σε περιορισμένο αριθμό χωρών.

g. Parkmobile

Παρέχει ασφαλή ηλεκτρονική πρόσβαση σε δεδομένα στάθμευσης σας, έτσι ώστε να μπορείτε να παρακολουθείτε τα έξοδα του παρκινγκ σας, να αλλάξετε τις ρυθμίσεις του λογαριασμού σας ή να εκτυπώσετε μια αναφορά. Μπορείτε επίσης να εξάγετε εύκολα πληροφορίες για επιστροφές εξόδων.

h. Car park Rates

Παρέχει κάμερες κυκλοφορίας για τη μετάδοση κατάσταση της κυκλοφορίας σε πραγματικό χρόνο σε μεγάλους οδικούς άξονες και τα σημεία ελέγχου.

i. Airport Parking

Προσφέρει διαθέσιμους χώρους στάθμευσης σε ένα αριθμό από διαφορετικά αεροδρόμια σε μερικές πόλεις.

j. Waze

Επιτρέπει την ανταλλαγή πληροφορίας σχετικά με την κυκλοφορία στους δρόμους σε πραγματικό χρόνο και οδικές άλλες πληροφορίες για να εξοικονομήσετε χρόνο, χρήματα, όπως και αποφυγή της αστυνομίας από παγίδες. Οι οδηγοί επικοινωνούν μεταξύ τους για ενημέρωση σε περίπτωση ατυχημάτων, κλειστούς δρόμους και άλλα έτσι ώστε αν επιθυμούν να πάρουν διαφορετική διαδρομή για τον προορισμό τους, να το πράξουν έγκαιρα. Με λίγα λόγια το Waze αφορά τη συμβολή και το πόσο επωφελούνται όλοι οι οδηγοί από το κοινό καλό.

k. INRIX XD Traffic Maps & Alerts

Το INRIX χρησιμοποιεί crowd source δεδομένα από περισσότερα από απλά τηλέφωνα. Έχει GPS και δεδομένα χαρτών από τα εκατομμύρια φορτηγά, κάμερες, αισθητήρες δρόμων, και από συστήματα πλοήγησης αυτοκινήτου. Η εφαρμογή δίνει πληροφορίες στους οδηγούς για τις τρέχουσες συνθήκες κυκλοφορίας, καθώς και τις προβλέψεις της κυκλοφορίας για όλους τους αυτοκινητόδρομους, δρόμους σε πόλεις και άλλους δρόμους σε διάφορες πόλεις.

Οι ακόλουθες εφαρμογές είναι διαθεμένες στο Apple IOS

l. HONK

Είναι μια άμεση οδική βοήθεια μέσω κινητής εφαρμογής σε περίπτωση προβλήματος. Προσφέρει άμεση βοήθεια 24 ώρες το 24ωρο γύρο στα 15-30 λεπτά σε περίπτωση ανάγκης. Παρέχει πρόσβαση σε ρυμούλκηση, αλλαγή ελαστικών, εκκίνησης οχήματος, ή σε περίπτωση που κλειδωθήκατε έξω. Τέλος, για να γίνεται μέλος χρειάζεται μόνο $49.

m. Park my Car

Όταν παρκάρετε το αυτοκίνητό σας και λίγο πριν κατεβείτε, ανοίξτε την εφαρμογή και επιλέξτε «Park». Θα βρείτε τη θέση του αυτοκινήτου σας και όταν επιστρέψετε, απλά ανοίξτε την εφαρμογή και πατήστε "Find" θα δείξει το αυτοκίνητό σας σταθμευμένο χρησιμοποιώντας τοποθεσίες στο χάρτη. Για περιορισμένη χρήση αλλά και πολύ απλή εφαρμογή εύκολη στη χρήση. Αυτό βοηθά πολύ σε περιπτώσεις που πάτε βόλτα σε ξένο προορισμό, παρκάρετε και κατεβείτε για εξόρμηση όμως δυσκολεύεστε να βρείτε το όχημα σας μετά επειδή δεν γνωρίζετε την περιοχή.

n. Park Shark

Ίσως από τις πιο απλές εφαρμογές αναζήτησης χώρου στάθμευσης. Τα βήματα που πρέπει να ακολουθήσετε είναι: 1) Επιλέξτε μια θέση όπου θέλετε να παρκάρετε 2) Επιλέξτε την ημερομηνία και την ώρα που θέλετε να ξεκινήσει το παρκόμετρο, 3) Αναφέρεται πόσο χρόνο θα διαρκέσει η στάθμευση σας και τέλος , 4) Επιλέξτε μια μέθοδο πληρωμής. Έξυπνα και γρήγορα !!

o. Gatu Parking Stockholm

Οι λειτουργίες που μπορούν να εμφανίσει στο χάρτη με Gatu parking είναι οι ακόλουθες. Κανονικό παρκινγκ (indoor), κλειστό παρκινγκ (outdoor), χώροι στάθμευσης για άτομα με περιορισμένη κινητικότητα, στάθμευσης MC, στάθμευσης λεωφορείων, σταθμεύσεις φορτηγών και διπλωματικές σταθμεύσεις (diplomatic parking).

p. Trip Buddy

Με την πιο πάνω εφαρμογή μπορείτε να ελέγξτε τα στατιστικά στοιχεία της τοποθεσίας σας, συμπεριλαμβανομένων γεωγραφικό μήκος, πλάτος, ύψος, διανυθείσα απόσταση, την ταχύτητα και τις τρέχουσες και προβλεπόμενες καιρικές συνθήκες για την περιοχή.

Έτσι είστε απόλυτα προγραμματισμένοι για την κάθε σας έξοδο. Επίσης σε περίπτωση που χαθήκατε βρίσκετε την ακριβή σας τοποθεσία ανά πάσα στιγμή.

q. Find my Car Smarter

Χρησιμοποιεί μια νέα τεχνολογία που ονομάζεται "Bluetooth Smart". Όταν ξεκινάτε το αυτοκίνητό σας, η Find My Car Smarter εφαρμογή συνδέεται ασύρματα με τη Find My Car Smarter συσκευή. Αφού έχετε σταθμεύσει το αυτοκίνητό σας, το App συνειδητοποιεί ότι το αυτοκίνητό σας είναι σταθμευμένο και, στη συνέχεια, ενεργοποιεί το GPS για το iPhone σας για να σώσετε την τοποθεσία αυτοκινήτου σας. Αν χάσετε το αυτοκίνητό σας, απλά ξεκινήστε την εφαρμογή και θα σας βοηθήσει να περιηγηθείτε στο δρόμο δείχνοντάς σας την τρέχουσα θέση σας, και τη θέση που έχετε σταθμεύσει τελευταίος το αυτοκίνητό σας.

r. NCP Car Finder

Η πιο πάνω εφαρμογή παρέχει free GPS augmented reality ''find my car/where I parked'' λειτουργία, τηλεφωνική υποστήριξη πελατών 24/7 και UK σύνδεσμο παρκινγκ αεροδρομίου στην online υπηρεσία κράτησης.

s. Parking Pin

Χρησιμοποιεί τους νεότερους iPhones συνεπεξεργαστές σε συνδυασμό με τις Υπηρεσίες τοποθεσίας για να εντοπίσει το σημεία σας όταν εσείς παρκάρετε. Το κάνει αυτό στο παρασκήνιο, ενώ το iPhone σας είναι σε λειτουργία "sleep" ή όταν χρησιμοποιείτε άλλες εφαρμογές. Την επόμενη φορά που θα ανοίξετε την εφαρμογή, η τρέχουσα θέση σας θα αποθηκευτεί μαζί με μια λίστα των πιο πρόσφατων σημείων σας.

t. My Ampera

Παρέχει τις ακόλουθες λειτουργίες: 1) My Ampera Agent, όπου μπορείτε να επικοινωνήστε με τις υπηρεσίες έκτακτης ανάγκης ή το προσωπικό σας Ampera agent απευθείας μέσω τηλεφώνου ή ηλεκτρονικού ταχυδρομείου. 2) Charging stations, που μπορείτε να βρείτε το πλησιέστερο σταθμό φόρτισης των ηλεκτρικών οχημάτων σας και να δείτε την τρέχουσα κατάσταση φόρτισης, παροχέα και τον αριθμό τηλεφώνου του σταθμού. 3) Ampera Info, όπου έχετε πρόσβαση στα κοινωνικά δίκτυα Ampera (Facebook, Twitter και το YouTube), Ampera Hub και θέα 360 μοιρών του οχήματος σας και διάφορες ταπετσαρίες(wallpapers).

u. NWS Weather alert

Παρέχει τις καιρικές συνθήκες, δεδομένα των υδάτων, το κλίμα, τις προβλέψεις και προειδοποιήσεις για την προστασία της ζωής και της περιουσίας. Η εφαρμογή έχει σχεδιαστεί για να επικεντρωθεί φυσικές καταστροφές και καιρικές επιφυλακές στο εσωτερικό της χώρας. Θα λαμβάνετε ειδοποιήσεις σε πραγματικό χρόνο στις iOS συσκευές σας. Παρέχει ειδοποιήσεις για χιόνια, ανέμους, χιονοθύελλα, αμμοθύελλα και πολλά άλλα.

2.2 Τρόποι βελτίωσης εφαρμογής

“Σήμερα οι πελάτες είναι απασχολημένοι και πλημμυρισμένοι με πληροφορίες και επιλογές. Θα πρέπει να κερδίσετε γρήγορα το ενδιαφέρον και την εμπιστοσύνη τους, αν θέλετε να τους οδηγήσετε με επιτυχία να εξετάσουν την αγορά των εφαρμογών σας. Ενώ μεγάλες εφαρμογές έρχονται σε πολλές μορφές και σχήματα, οι ακόλουθες πρακτικές θα σας δώσουν μερικούς από τους πιο κοινούς τρόπους για να ξεχωρίσει η εφαρμογή σας.[4]

Όπως είδαμε μέχρι τώρα έχουν δημιουργηθεί αρκετές εφαρμογές για την εξυπηρέτηση του σκοπού της στάθμευσης. Άλλες αρκετά απλές με περιορισμένη λειτουργικότητα για ευκολία του πελάτη-οδηγού, άλλες λίγο πιο σύνθετες με περισσότερες λειτουργίες με σκοπό να δημιουργήσουν μια πλήρη εφαρμογή. Πολλές από αυτές έχουν πανομοιότυπα

χαρακτηριστικά, κάποιες άλλες όμως εντάσσουν κάτι μοναδικό στο τομέα της στάθμευσης. Κάτι όμως που πολλοί από τους κατασκευαστές ίσως να παραλείπουν είναι το μέρος της εμφάνισης της εφαρμογής. Πως φαίνετε δηλαδή η εφαρμογή από την πρώτη επαφή της με το πελάτη. Όταν ο πελάτης έχει αναζητήσει μια εφαρμογή στο Store από το επίπεδο της περιγραφής (app description) μέχρι και της ανάληψης (download), μια εφαρμογή πρέπει να είναι ελκυστική προς το πελάτη. Αυτά είναι που θα δούμε στο υποκεφάλαιο αυτό αγγίζοντας τα σε διάφορες πτυχές και σε πιο αναλυτικό επίπεδο.

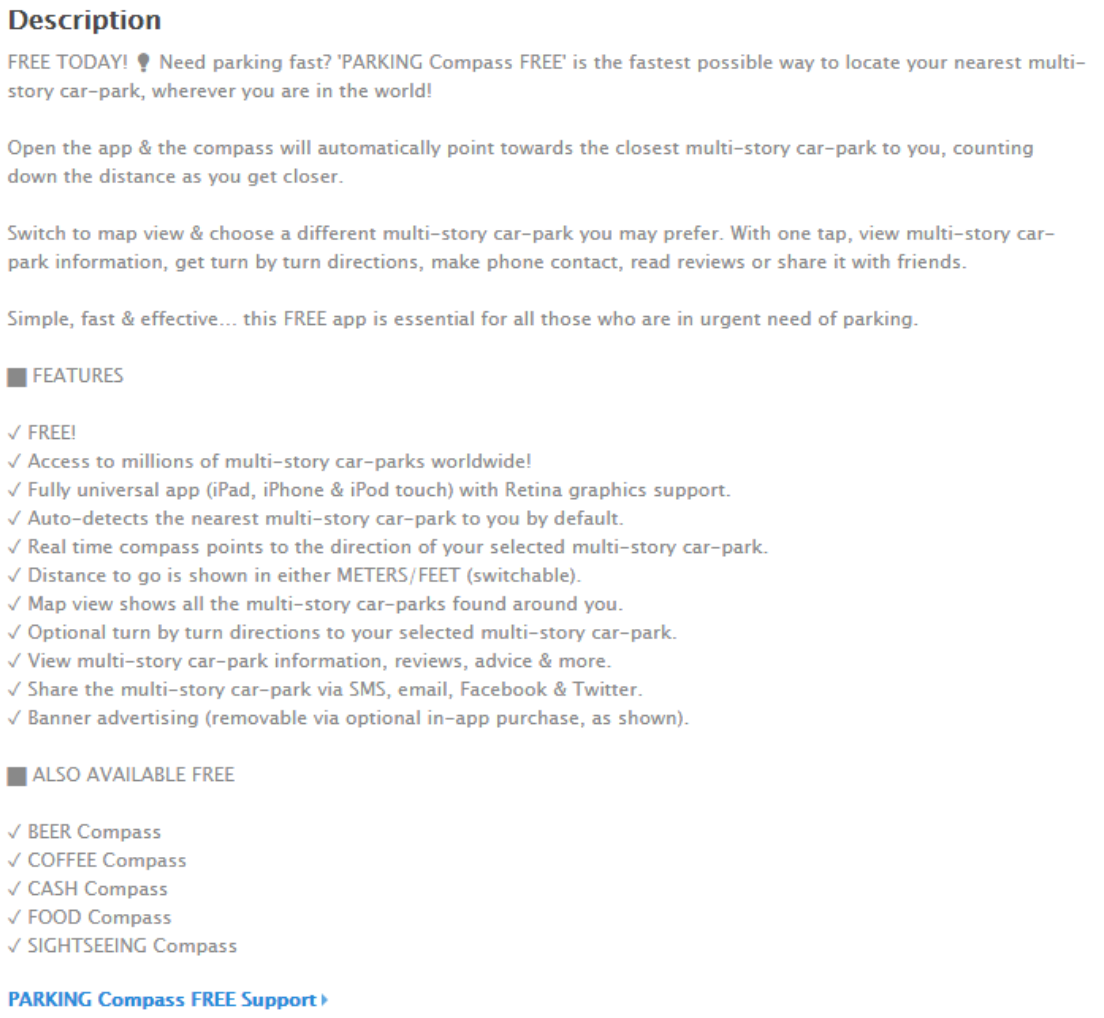
Unique Description for Your Unique App

Αρχικά, κατά την περιγραφή της εφαρμογής στο App Store, πρέπει να έχει ένα ωραίο και αναλυτικό κείμενο περιγραφής. Ένα ικανοποιητικά μεγάλο κείμενο περιεχομένων σχετικά για την εφαρμογή πρέπει να περιέχει ότι παρέχεται από την εφαρμογή με απλά και περιεκτικά λόγια έτσι ώστε ο οποιοσδήποτε, να μπορεί να καταλάβει τι ωφελήματα μπορεί να προσκομίσει κατεβάζοντας τη συγκεκριμένη εφαρμογή. Πρέπει γενικά να μην είναι πολύ μικρό το κείμενο περιγραφής, δηλαδή σε μια παράγραφο για παράδειγμα , γιατί εκεί μέσα ο χρήστης θα πρόκειται να διαβάσει όλη την δουλειά που έχει η εφαρμογή μαζεμένη – με λίγα λόγια, πιθανό με τον τρόπο αυτό να προκαλέσει σύγχυση και ουσιαστικά να μην καταλάβει τι περιέχει η εφαρμογή.

Σε αντίθεση ένα σχετικά μεγάλο κείμενο περιγραφής μπορεί να έχει σε κάθε παράγραφο και κάτι διαφορετικό με οργανωμένο τρόπο. Με τον τρόπο αυτό θα προσελκύσει το ενδιαφέρον του πελάτη. Λόγο αυτών η επεξεργασία του κειμένου περιεχομένων καλό θα ήταν να γίνει από κάποιον ειδικό και όχι από τον ίδιο τον προγραμματιστή όπου θα μπορεί να διατυπώσει αφαιρετικά το περιεχόμενο της εφαρμογής και όχι με σύνθετες λέξεις η όρους που ο πελάτης μπορεί και να μην καταλάβει. Στο κείμενο περιγραφής επίσης είναι καλό να έχει σε λίστα όλα τα features της εφαρμογής αριθμημένα έτσι παρά το κείμενο ο αναγνώστης να μπορεί να δει τι περιέχει η εφαρμογή στα γρήγορα.

Τέλος, αναγκαίο θα ήταν να γραφτούν όποια δικαιώματα (permissions) είναι απαραίτητα για την εφαρμογή. Εκ πρώτης όψης λέμε τα ελαττώματα της εφαρμογής στον αναγνώστη , γενικά όμως είναι καλύτερα έτσι παρά να αποκρύψουμε την αλήθεια.

Ένα παράδειγμα που περιέχει αυτά που μόλις ανάφερα είναι η περιγραφή που φαίνετε στο σχήμα 3 πιο κάτω



Description

FREE TODAY! Need parking fast? 'PARKING Compass FREE' is the fastest possible way to locate your nearest multi-story car-park, wherever you are in the world!

Open the app & the compass will automatically point towards the closest multi-story car-park to you, counting down the distance as you get closer.

Switch to map view & choose a different multi-story car-park you may prefer. With one tap, view multi-story car-park information, get turn by turn directions, make phone contact, read reviews or share it with friends.

Simple, fast & effective... this FREE app is essential for all those who are in urgent need of parking.

■ FEATURES

✓ FREE!
✓ Access to millions of multi-story car-parks worldwide!
✓ Fully universal app (iPad, iPhone & iPod touch) with Retina graphics support.
✓ Auto-detects the nearest multi-story car-park to you by default.
✓ Real time compass points to the direction of your selected multi-story car-park.
✓ Distance to go is shown in either METERS/FEET (switchable).
✓ Map view shows all the multi-story car-parks found around you.
✓ Optional turn by turn directions to your selected multi-story car-park.
✓ View multi-story car-park information, reviews, advice & more.
✓ Share the multi-story car-park via SMS, email, Facebook & Twitter.
✓ Banner advertising (removable via optional in-app purchase, as shown).

■ ALSO AVAILABLE FREE

✓ BEER Compass
✓ COFFEE Compass
✓ CASH Compass
✓ FOOD Compass
✓ SIGHTSEEING Compass

PARKING Compass FREE Support

Σχήμα 3 : Παράδειγμα περιγραφής κινητής εφαρμογής [5]

<u>Create an Informative and Engaging Video</u>

Έπειτα, στην περιγραφή θα ήταν καλό να βάζατε ένα βίντεο demo της εφαρμογής που θα καθοδηγεί τον ενδιαφερόμενο για τον τρόπο λειτουργίας της. Πάντα με ένα βίντεο ο πελάτης καταλαβαίνει καλύτερα και πιο εύκολα τον τρόπο λειτουργία της εφαρμογής. Επίσης, για ενδιαφερόμενους που δεν θέλουν να χάσουν τον χρόνο τους διαβάζοντας της περιγραφή θα προτιμήσουν το βίντεο.

Limited Period Offer Title

“Προσφέροντας applications χωρίς κόστος για τους πιθανούς πελάτες σας εναπόκειται στην απόφασή σας ως developers. Εάν δεν θέλετε την εφαρμογή σας να προσφερθεί δωρεάν, μπορείτε να επισημάνετε την εφαρμογή σας με τίτλο όπως "Περιορισμένη περίοδος προσφοράς, 50% έκπτωση". Έμμεσα, είναι επίσης μια έξυπνη προσέγγιση για να πιάσει την προσοχή των χρηστών και να κάνουν την εφαρμογή σας ελκυστική και δημοφιλές.”[6]

Make your app for all language people

Η γλώσσας γραφής θα ήταν καλό να είναι σε ένα παγκόσμιας διάδοσης γλώσσα όπως για παράδειγμα είναι τα αγγλικά για να μπορεί ένα μεγάλο μέρος των αναγνωστών να το διαβάσουν.

Eye-catching Design

Είναι αλήθεια ότι η λειτουργικότητα και η χρησιμότητα μιας εφαρμογής είναι που πραγματικά έχει σημασία, αλλά αυτά τα πράγματα ο πελάτης θα τα παρατηρήσει τελικά τη στιγμή που θα χρησιμοποιήσει την εφαρμογή σας για λίγες μέρες. Αυτό που συνήθως τραβάει την προσοχή μας πρώτα είναι ο σχεδιασμός. Η φράση «οι πρώτες εντυπώσεις έχουν σημασία" ισχύει και όταν πρόκειται για εφαρμογές. Ποτέ μην υποτιμάτε την ομορφιά της απλότητας καθώς οι περισσότερες από τις δημοφιλείς εφαρμογές είναι απλές αλλά χρήσιμες και αυτό είναι που προσελκύει τους χρήστες.

App tile image

Η εμφάνιση του εικονιδίου της εφαρμογής σας παίζει σημαντικό ρόλο στο πώς η εφαρμογή σας γίνεται αντιληπτή από τους πελάτες. Η δημιουργία ενός ελκυστικού εικονιδίου εφαρμογής είναι ένας από τους καλύτερους τρόπους για να επιστήσει την προσοχή των πελατών στην εφαρμογή σας και να δελεάσει τους ενδιαφερόμενους πελάτες να εξερευνήσουν το προϊόν σας. Κάποια βήματα που μπορούν να βοηθήσουν την εφαρμογή σας να ξεχωρίζει:

- Δημιουργήστε μοναδικό και εντυπωσιακό εικονίδιο.
- Φροντίστε να είναι απλή. αποφεύγοντας τις περιττές ακαταστασίες.
- Βεβαιωθείτε ότι η εικόνα αντιστοιχεί στην εφαρμογή σας.
- Κάντε το κείμενο εύκολο να διαβαστεί.

Κεφάλαιο 3

Περιγραφή της ιστοσελίδας Spark

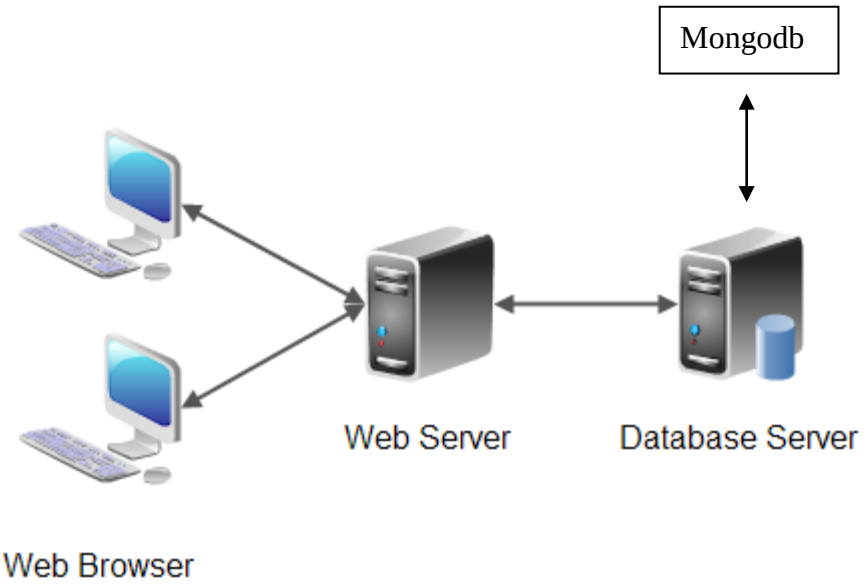
3.1 Επισκόπηση Αρχιτεκτονικής Spark ιστοσελίδας 16

3.2 Υλικό που χρησιμοποιήθηκε 17

3.3 Λειτουργίες Συστήματος 18

Στο κεφάλαιο αυτό θα δούμε αρχικά την ιστοσελίδα Spark. Μια δουλειά και παλαιότερη διπλωματική εργασία που έχει γίνει στο θέμα της έξυπνης στάθμευσης από την οποία εμπνεύστηκε η παρούσα διπλωματική εργασία. Θα δούμε την αρχιτεκτονική της και το υλικό που χρησιμοποιήθηκε για την υλοποίηση της. Τέλος, θα αναφερθούμε στις λειτουργίες της μια προς μια λεπτομερές.

3.1 Επισκόπηση Αρχιτεκτονικής Spark ιστοσελίδας



3.2 Υλικό που χρησιμοποιήθηκε

1. Microsoft Visual Studio με MVC μοτίβο στη C #

   MVC, είναι ένα λογισμικό αρχιτεκτονικό πρότυπο για την υλοποίηση διεπαφών του χρήστη σε υπολογιστές . Χωρίζει μια συγκεκριμένη εφαρμογή λογισμικού σε τρία αλληλοσυνδεόμενα μέρη, έτσι ώστε να διαχωριστούν εσωτερικές αναπαραστάσεις των πληροφοριών από τις πληροφορίες που παρουσιάζονται με ή γίνονται αποδεχτές από τον χρήστη.

2. Fuzzy Logic Framework/Library

   Fuzzy Logic, είναι μια μορφή λογικής που μπορεί να πάρει πολλαπλές τιμές . Οι τιμές αληθείας των μεταβλητών μπορεί να είναι οποιοσδήποτε πραγματικός αριθμός μεταξύ 0 και 1, θεωρείται ότι είναι «ασαφής» .Σε αντίθεση, στην Boolean λογική, οι τιμές αληθείας των μεταβλητών μπορεί να είναι μόνο 0 ή 1 , που συχνά αποκαλείται «crisp» τιμές. Η Ασαφής λογική έχει επεκταθεί για να χειριστεί την έννοια της μερικής αλήθειας, όπου η τιμή αλήθειας μπορεί να κυμαίνεται μεταξύ 0 και 1.

3. Bing Maps

   Είναι μια υπηρεσία web χαρτογράφησης, που παρέχεται από τη Bing “σουίτα” της Microsoft.

4. Google Charts

   Είναι ένας τρόπος για την απεικόνιση των δεδομένων στο διαδίκτυο, χρησιμοποιώντας διάφορες μορφές διαγραμμάτων και γραφημάτων.

5. MongoDB

   Είναι μια cross-platform document-oriented βάση δεδομένων.

6. Windows Server 2012R2

   Για τη φιλοξενία της ιστοσελίδας(hosting), τις διαδικτυακές υπηρεσίες και τη MongoDB, χρησιμοποιήθηκε ο Windows Server 2012R2 64-bit. Σημαντικός

λόγος για την επιλογή του Windows Server είναι η φιλοξενία της ιστοσελίδας και των διαδικτυακών υπηρεσιών.

3.3 Λειτουργίες Συστήματος

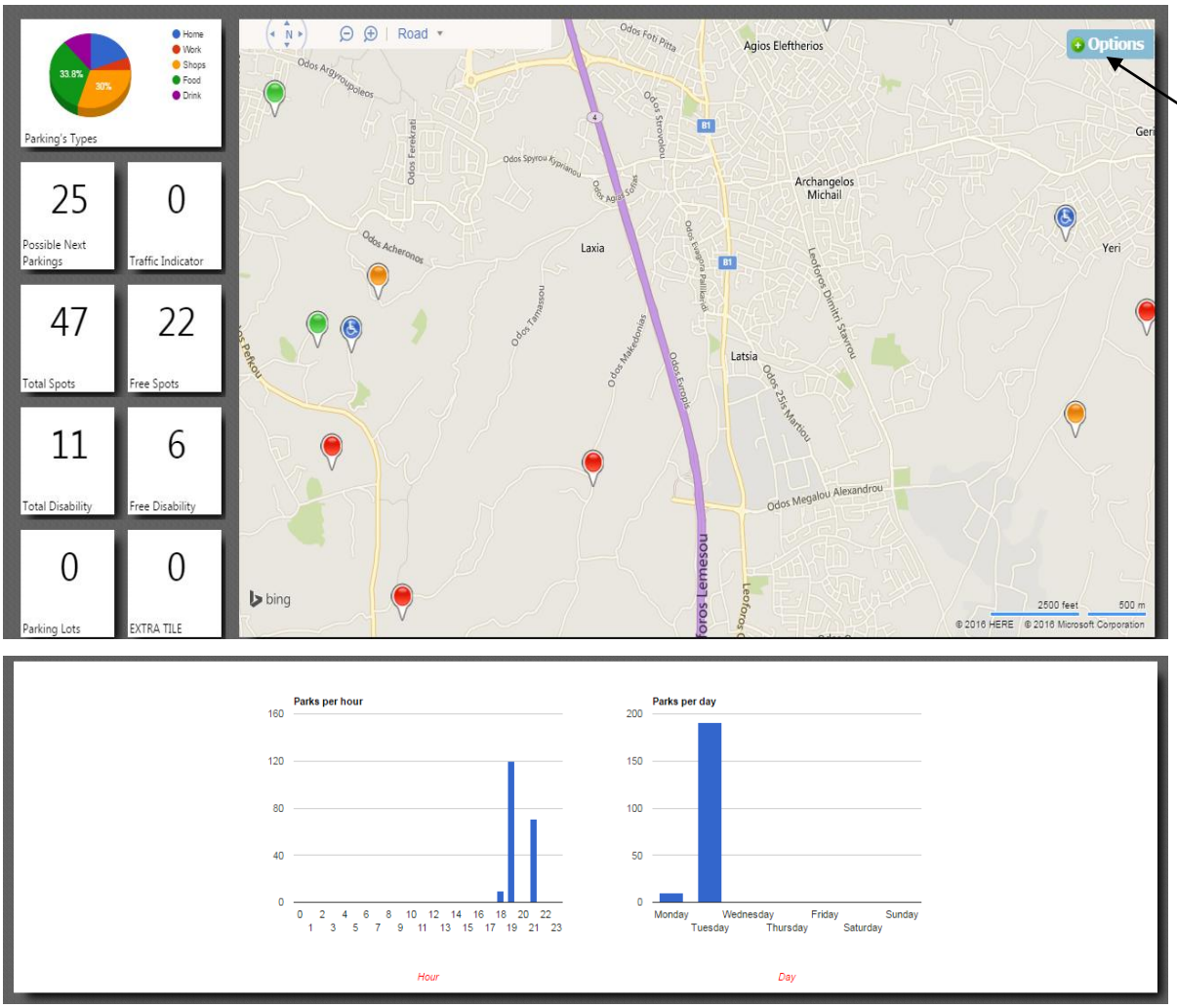


1 Home:

Το κουμπί αυτό παραπέμπει το χρήστη από το σημείο που βρίσκεται στη κυρίως σελίδα της εφαρμογής όπως φαίνετε στην πιο πάνω εικόνα.

2 Map

Εμφανίζει στο χρήστη το χάρτη με τα πλησιέστερα διαθέσιμα παρκινγκ που βρίσκονται κοντά στην τοποθεσία που βρίσκεται. Όπως επίσης και στατιστικά σχετικά με το χώρο αναζήτησης του (park per day + park per hour).



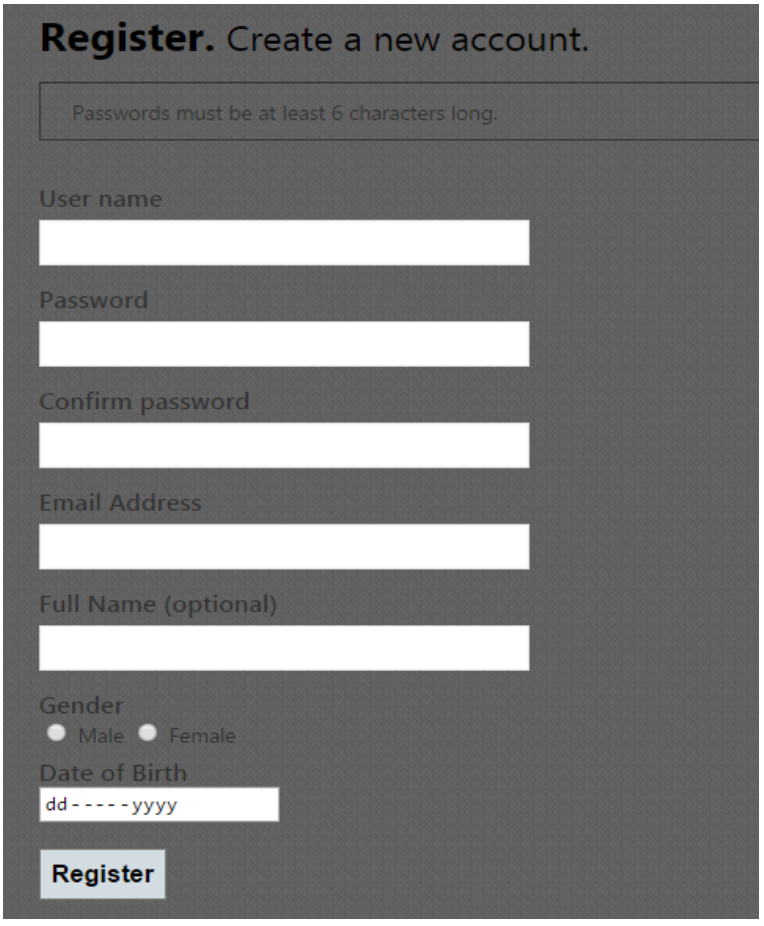
2,1 Options

Η λειτουργία αυτή επιτρέπει στο χρήστη να διαχειριστεί τις επιλογές της αναζήτησης του. Όπως για παράδειγμα ακτίνα αναζήτησης διαθέσιμου χώρου στάθμευσης (range) , χρονικό διάστημα (time interval) , όριο (limit). Επίσης, έχει κάποιες Boolean επιλογές όπως disable (χώροι στάθμευσης για άτομα με ειδικές ανάγκες) , Bookmarks (σελιδοδείκτες) , Parking Lots (χώροι στάθμευσης), Possible Next Free . Τέλος, μπορεί ο χρήστης να κάνει αναζήτηση με βάση όλα τα παρκινγκ – διαθέσιμα και μη (all), διαθέσιμα παρκινγκ (empty), κανένα (none).



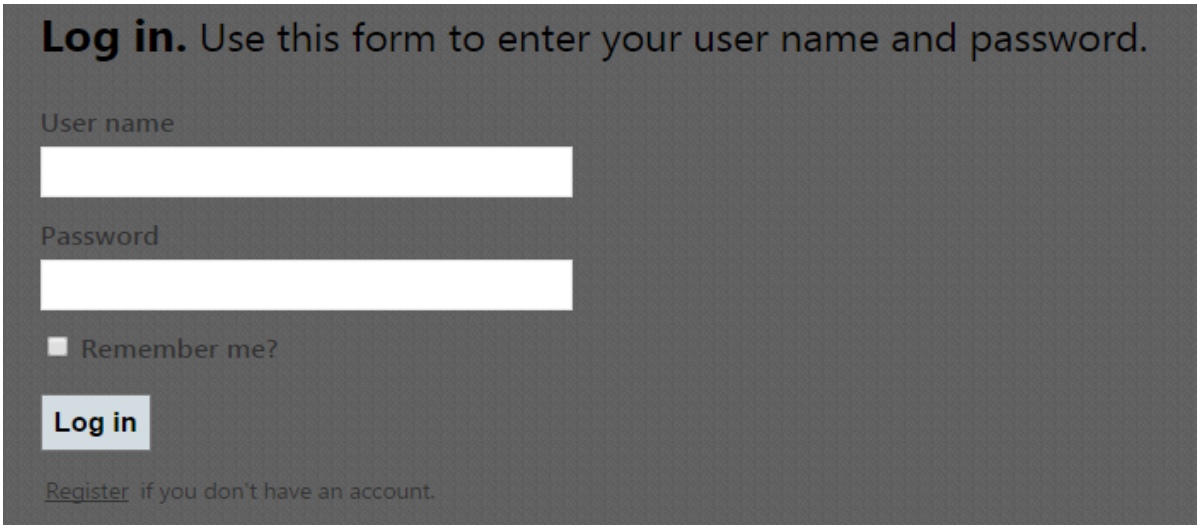
3 Register

Καταχώριση χρήστη. Σε περίπτωση που ο χρήστης δεν είναι μέλος , μπορεί να γίνει κάνοντας “register”. Η διαδικασία αυτή γίνεται μόνο μια φορά , ενεργά μέλη μπορούν αργότερα να κάνουν “log in” στη σελίδα.



4 Log in

Σύνδεση. Αφού έχει προηγηθεί η διαδικασία του “register” για πρώτη φορά , τα ενεργά μέλη τους επιτρέπεται να κάνουν “log in”. Οι χρήστες που έχουν λογαριασμό έχουν κάποια προτερήματα όπως για παράδειγμα να δουν στατιστικά σχετικά με τα requests και τα responses που έγιναν πάνω στη βάση από τους mobile users.



Visit spark at: spark.cs.ucy.ac.cy [7]

Κεφάλαιο 4

Περιγραφή της εφαρμογής Spark

4.1 Επισκόπηση Αρχιτεκτονικής Spark εφαρμογής 22
4.2 Υλικό που χρησιμοποιήθηκε 23
4.3 Λειτουργίες Συστήματος 24
4.4 Wireframe και Flowchart εφαρμογής 35

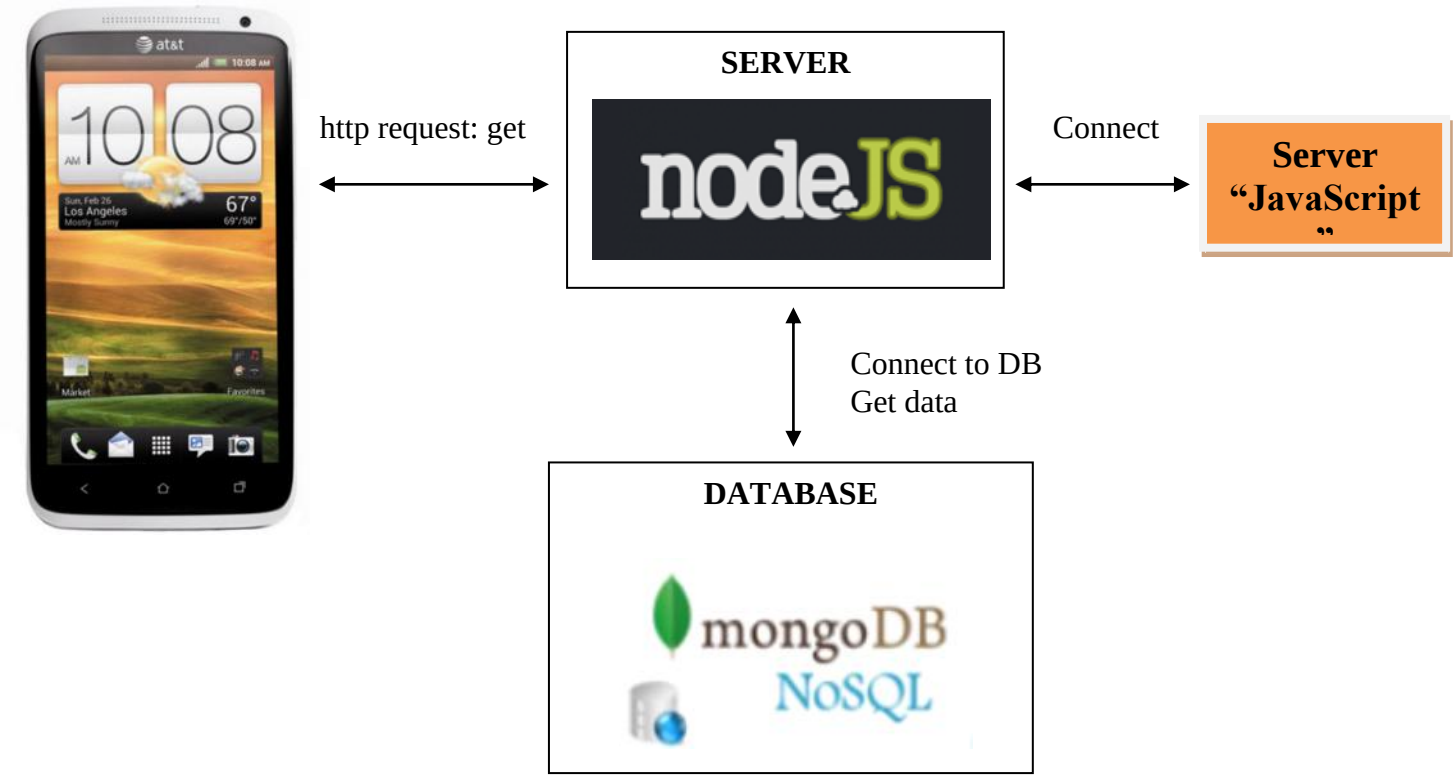
Στο κεφάλαιο αυτό θα δούμε την εφαρμογή Spark. Η εφαρμογή αυτή έχει ως στόχο να συμπληρώσει την πιο πάνω ιστοσελίδα παρέχοντας περισσότερες υπηρεσίες και δημιουργώντας μια πλήρης κινητή εφαρμογή "φιλική" προς τον χρήστη. Θα δούμε αρχικά την αρχιτεκτονική της και το υλικό που χρησιμοποιήθηκε για την υλοποίηση της. Έπειτα, θα αναφερθούμε στις λειτουργίες της μια προς μια. Τέλος, θα παρουσιάσουμε την εφαρμογή Spark μέσο του εργαλείου wireframe και διαγραμματικά μέσω Flowchart.

4.1 Επισκόπηση Αρχιτεκτονικής Spark εφαρμογής

Το Spark απαρτίζεται από μία Android έξυπνη κινητή συσκευή, έναν κεντρικό εξυπηρετητή και μία βάση δεδομένων Mongodb.

Η πιο διαδεδομένη μέθοδος για να γίνει σύνδεση από μία Android συσκευή σε μια απομακρυσμένη βάση δεδομένων Mongodb είναι κάποιο είδος ενδιάμεσης υπηρεσίας. Χρησιμοποιούμε τη γλώσσα JavaScript δεδομένου ότι αλληλεπιδρά με τις βάση δεδομένων, για την ασφαλή ανάκτηση πληροφοριών από τον διακομηστή (server). Για την πραγματοποίηση σύνδεσης χρησιμοποιήσαμε το πρωτόκολλο HTTP από το Android σύστημα. Γίνεται χρήση της αρχιτεκτονικής client - server όπου πελάτης είναι η Android έξυπνη συσκευή ενώ στην πλευρά του διακομιστή

υπάρχει ένας συνδυασμός Java Script με Mongodb. Συνοψίζοντας η γλώσσα JavaScript βρίσκεται στο ενδιάμεσο για την επικοινωνία της Android συσκευής με τη βάση δεδομένων Mongodb (Σχήμα 4.1.1) .



Σχήμα 4.1.1 Επικοινωνίας κινητού με JavaScript και Mongodb

4.2 Υλικό που χρησιμοποιήθηκε

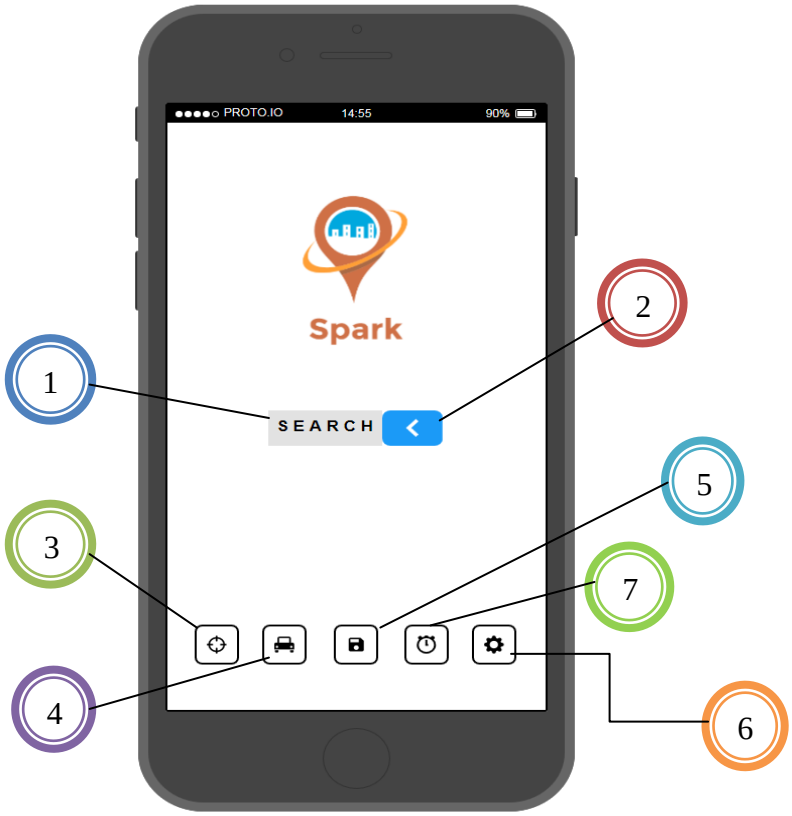
Για την ανάπτυξη του Spark συστήματος χρησιμοποιήθηκε ένας εξυπηρετητής, ο εξυπηρετητής που δημιουργήσαμε μέσω του Xampp ο οποίος είναι ένα δωρεάν και open source cross-platform web server. Μέσω αυτού και χρησιμοποιώντας scripts μπορούσαμε να εκτελούμε και να δοκιμάζουμε την εφαρμογή μας λαμβάνοντας πληροφορίες και προβάλλοντας τις στην εφαρμογή για έλεγχο. Επίσης για την ανάπτυξη της εφαρμογής και για την δοκιμή της χρησιμοποιήθηκε η συσκευή HTC phone one X+ (Σχήμα 4.2.1) , η οποία και παραχωρήθηκε από τον επιβλέπον καθηγητή της εν λόγω ατομικής διπλωματικής εργασίας.

HTC PHONE ONE X+	
	**Storage** Internal phone storage: 64GB RAM: 1GB **CPU Processor** NVIDIA Tegra 3 AP37 quad-core **Platform** Android 4.1.1 **Wi-Fi** IEEE 802.11 b/g/n

Σχήμα 4: HTC PHONE ONE X+ Specs [8]

4.3 Λειτουργίες Εφαρμογής

Κυρίως μενού της εφαρμογής





Search textbox

Στο πεδίο αυτό ο χρήστης γράφει την τοποθεσία που επιθυμεί να αναζητήσει διαθέσιμο χώρο στάθμευσης.



Search button

Αφού ο χρήστης γράψει στο «Search textbox» την τοποθεσία που επιθυμεί να αναζητήσει πατά το συγκεκριμένο κουμπί για να ξεκινήσει η διαδικασία αναζήτησης.



Άμεση αναζήτηση χώρου στάθμευσης

Πατώντας το κουμπί η εφαρμογή θα αναζητήσει τους διαθέσιμους χώρους στάθμευσης στην παρούσα τοποθεσία του χρήστη. Η διαδικασία αναζήτησης και εμφάνισης των αποτελεσμάτων είναι πανομοιότυπη με την πιο πάνω (1).



Αποθήκευση / Αναζήτηση

Το κουμπί αυτό παραπέμπει τον χρήστη στο μενού Save / Find



Save Spot

Πατώντας το κουμπί η εφαρμογή εξετάζει αν η παρούσα τοποθεσία βρίσκεται στη βάση δεδομένων , αν βρίσκεται τυπώνει το ανάλογο μήνυμα , αλλιώς αποθηκεύει την τοποθεσία στη βάση.



Session

Το κουμπί αυτό παραπέμπει τον χρήστη στο μενού Session



Settings

Το κουμπί αυτό παραπέμπει τον χρήστη στο μενού Settings

Αποτελέσματα αναζήτησης χώρου στάθμευσης



1 Pie chart

Η πιο πάνω παράσταση απεικονίζει τα είδη των παρκινγκ που υπάρχουν για την συγκεκριμένη αναζήτηση (πχ Home, Work, Shops e.t.c.)

2 Labels

Οι ετικέτες περιέχουν διάφορες πληροφορίες όπως Possible Next Parkings , Total Spots , Parking Lots e.t.c.

3 Home

Το κουμπί αυτό παίρνει τον χρήστη πίσω στο κυρίως μενού.

4 Map

Ο χάρτης περιέχει τους διαθέσιμους χώρους στάθμευσης που υπάρχουν για στη τοποθεσία αναζήτησης του χρήστη.

5 Statistics

Εδώ απεικονίζονται γραφικές παραστάσεις σχετικές με στατιστικά της αναζήτησης του χρήστη όπως Parks Per hour , Parks Per day. Τις συγκεκριμένες γραφικές ο χρήστης μπορεί να τις χρησιμοποιήσει για να πάρει μια γενική εικόνα για τα παρκινγκ ανά μέρα έτσι ώστε να ξέρει σε ποιες μέρες γεμίζουν και ποιες μέρες είναι συνήθως άδεια για να πάρει τις αποφάσεις του.

6 Back

Το κουμπί αυτό παίρνει τον χρήστη ένα μενού πίσω, στην προκειμένη περίπτωση πίσω στο κυρίως μενού.

Save / Find Location



1 Save / Find Button

Το κουμπί αυτό χρησιμοποιείτε σε περίπτωση που ο χρήστης θέλει να αποθηκεύσει (save) την τοποθεσία του αυτοκινήτου του για να μπορεί αργότερα να την βρει (find). Μόλις πατήσει εμφανίζεται στην οθόνη το μήνυμα car position saved!

2 Home

Το κουμπί αυτό παίρνει τον χρήστη πίσω στο κυρίως μενού.

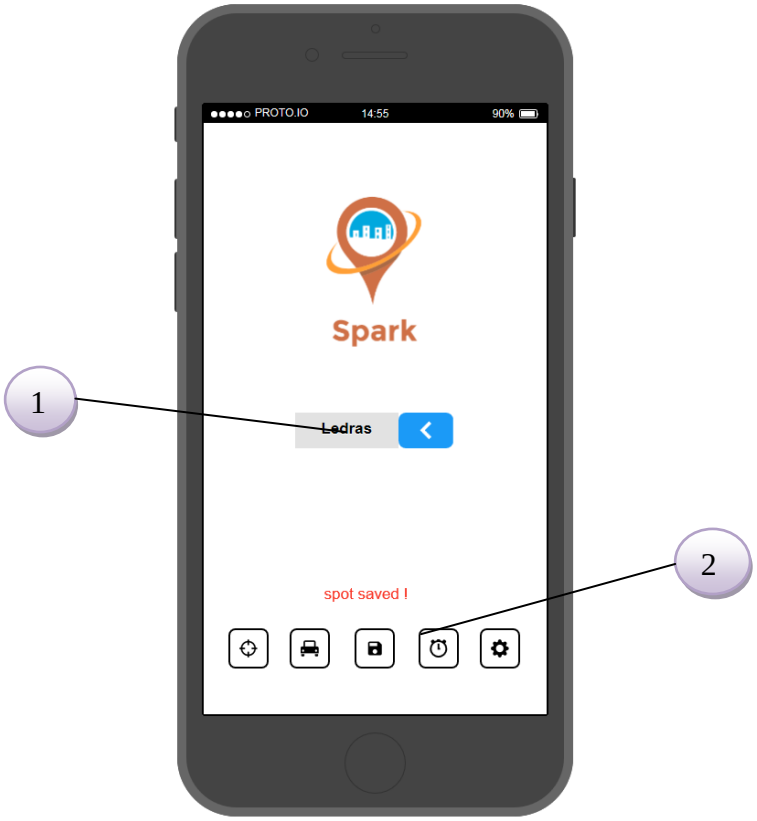
3 Extend Session

Σε περίπτωση που ο χρήστης επιθυμεί να επεκτείνει το χρόνο που είχε βάλει για στάθμευση , η να ορίσει εκείνη την ώρα χρόνο για την στάθμευση του πατώντας το κουμπί τον παραπέμπει στο «extend session» menu όπου μπορεί να το κάνει.

4 Back

Το κουμπί αυτό παίρνει τον χρήστη ένα μενού πίσω, στην προκειμένη περίπτωση πίσω στο κυρίως μενού.

Save park spot



1 Search textbox

Στο πεδίο αυτό ο χρήστης γράφει την τοποθεσία που επιθυμεί να αναζητήσει διαθέσιμο χώρο.

2 Save spot Button

Πατώντας το κουμπί η εφαρμογή εξετάζει αν η παρούσα τοποθεσία βρίσκεται στη βάση δεδομένων , αν βρίσκεται τυπώνει το μήνυμα spot saved , αλλιώς αποθηκεύει την τοποθεσία στη βάση.

Sessions menu



1 Time Remaining

Στο πιο πάνω label αναγράφει στο χρήστη το χρόνο που του απομένει για τη κράτηση που έκανε. Ο χρόνος παρουσιάζεται αναλυτικά σε ώρες, λεπτά, δευτερόλεπτα όπως βλέπουμε πιο πάνω.

2 Session Details

Εδώ παρουσιάζονται στο χρήστη διάφορες πληροφορίες σχετικές με τη κράτηση που έκανε όπως αρ. πινακίδων του αυτοκινήτου που έκανε την κράτηση , πότε άρχισε το παρκόμετρο να μετράει, πότε θα τερματίσει, πόσα θα στοιχίσει συνολικά στο χρήστη κ.τ.λ.

3 Home

Το κουμπί αυτό παίρνει τον χρήστη πίσω στο κυρίως μενού.

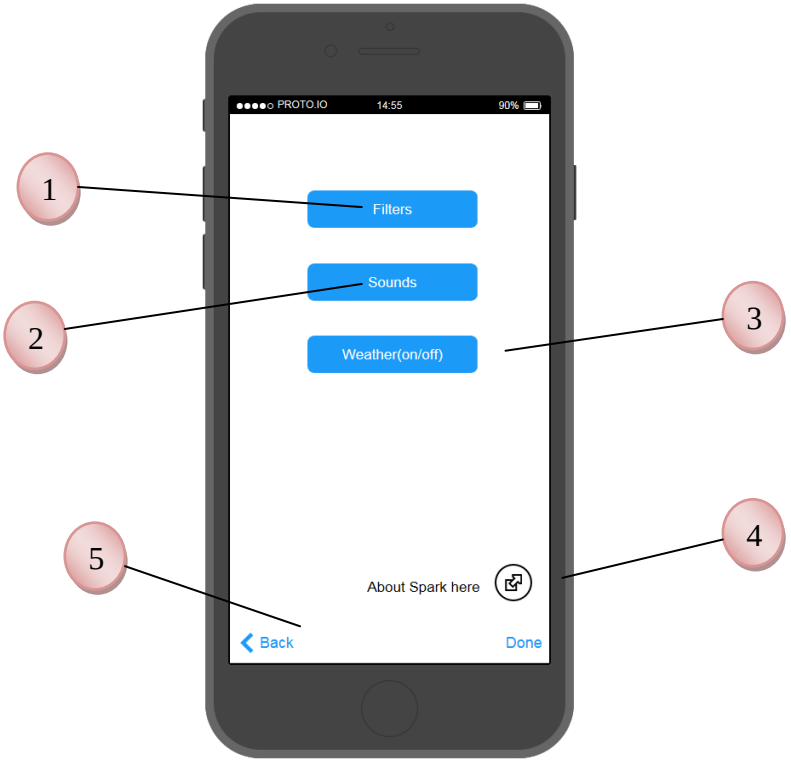
4 Extend Session Button

Σε περίπτωση που ο χρήστης επιθυμεί να επεκτείνει το χρόνο που είχε βάλει για στάθμευση, η να ορίσει εκείνη την ώρα χρόνο για την στάθμευση του πατώντας το κουμπί τον παραπέμπει στο extend session menu όπου μπορεί να το κάνει. Ουσιαστικά δουλεύει ως παρκόμετρο.

5 Back

Το κουμπί αυτό παίρνει τον χρήστη ένα μενού πίσω, στην προκειμένη περίπτωση πίσω στο κυρίως μενού.

Settings menu



1 Filters

Toggle button το οποίο εμφανίζει στο χρήστη τα φίλτρα που παρέχει η εφαρμογή σχετικά με τους χώρους στάθμευσης (βλ. σχήμα 1). Τα φίλτρα ορίζονται σε αύξον αριθμό με 1 το φίλτρο με την μεγαλύτερη προτεραιότητα στις προτιμήσεις του χρήστη και μικρότερο το 3.

2 Sounds

Οδηγεί τον χρήστη στο μενού των ήχων της εφαρμογής (βλ. σχήμα 2) Εκεί, ο χρήστης θα μπορεί να ρυθμίσει τους ήχους της εφαρμογής με βάση τις προτιμήσεις του.

3 Weather

Το toggle button αυτό όταν είναι «on» εμφανίζει στην οθόνη τους χρήστη ένα gadget καιρού. Η εφαρμογή προσφέρει αυτή την υπηρεσία για να μπορεί ο χρήστης να πάρει πληροφορίες σχετικά με τον καιρό και να μπορεί να βάλει τα φίλτρα του καλύτερα , ξέροντας επιπλέων την πληροφορία του καιρού. Για παράδειγμα, αν πρόκειται να βρέξει στα φίλτρα ο χρήστης θα βάλει να του εμφανίσει πρώτα χώρους στάθμευσης που είναι καλυμμένοι και κοντά στο προορισμό του (βλ. σχήμα 3).

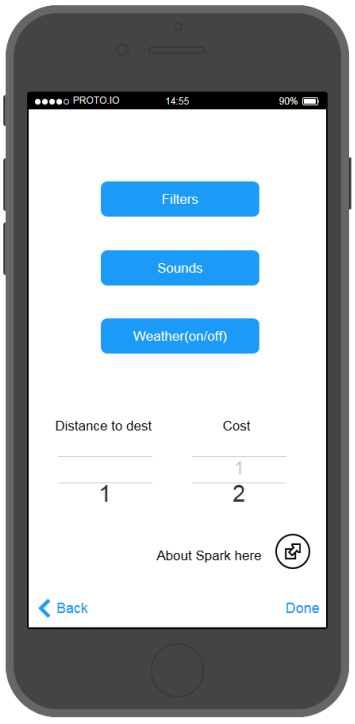
4 About Spark

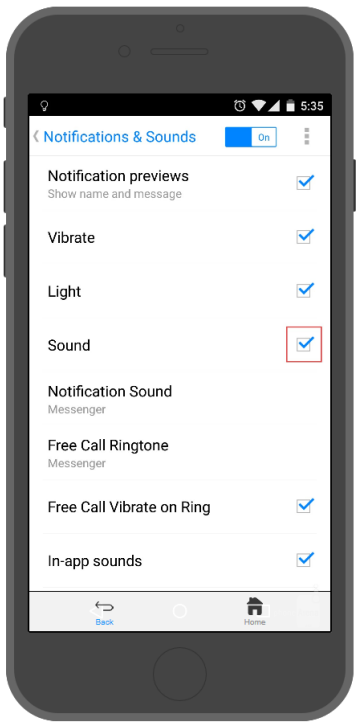
Ο σύνδεσμος αυτός παραπέμπει τον χρήστη στην ιστοσελίδα Spark (βλ. σχήμα 4).

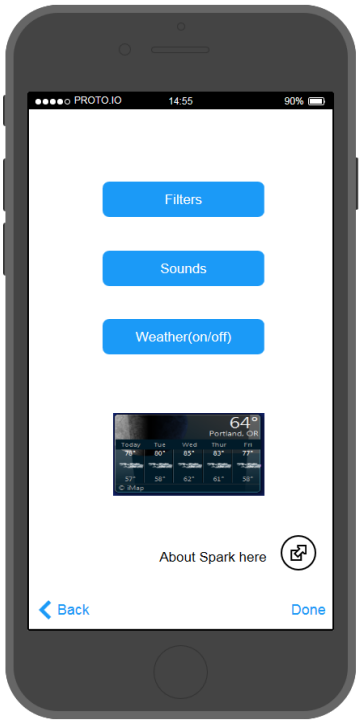
5 Back

Το κουμπί αυτό παίρνει τον χρήστη ένα μενού πίσω, στην προκειμένη περίπτωση πίσω στο κυρίως μενού.

Σχήμα 1 Σχήμα 2 Σχήμα 3 Σχήμα4









Λειτουργίες Εφαρμογής (με λόγια)

1. Αναζήτηση χώρου στάθμευσης “Search”

Στο κυρίως μενού μου ο χρήστης γράφει στο textbox την τοποθεσία που επιθυμεί να αναζητήσει και πατά το κουμπί “enter” που υπάρχει δίπλα για να ξεκινήσει η διαδικασία αναζήτησης.

- Επεξεργασία: Κατά τη διάρκεια της διαδικασία αυτής γίνεται επεξεργασία και αναζήτηση στη βάση δεδομένων αν υπάρχουν διαθέσιμοι χώροι στη τοποθεσία της επιλογής του.

- Αποτελέσματα: Αν η αναζήτηση ολοκληρωθεί με επιτυχία εμφανίζεται στην οθόνη του κινητού το μενού με τους διαθέσιμους χώρους, αλλιώς σε περίπτωση που εισήγαγε χώρο που δεν υπάρχει του εμφανίζει το αντίστοιχο μήνυμα.

2. Άμεση αναζήτηση χώρου στην τοποθεσία του χρήστη

Στο κυρίως μενού ο χρήστης πατώντας το κουμπί  η εφαρμογή θα αναζητήσει τους διαθέσιμους χώρους στάθμευσης στην παρούσα τοποθεσία του χρήστη. Η διαδικασία αναζήτησης και εμφάνισης των αποτελεσμάτων είναι πανομοιότυπη με την πιο πάνω (1).

3. Αποθήκευση / Αναζήτηση αυτοκινήτου

Επιλέγοντας το  του εμφανίζεται το ανάλογο μενού, όπου στο μενού αυτό ο χρήστης έχει τις επιλογές να αποθηκεύσει (save) τον χώρο που στάθμευσε το αυτοκίνητο του για να μπορεί να το αναζητήσει (find) αργότερα σε περίπτωση που έχει ξεχάσει που το έβαλε. Επίσης, μπορεί μέσω του “extend session” button να κάνει κράτηση για το χώρο στάθμευσης που το έχει βάλει.

4. Το κουμπί  που βρίσκεται στο κυρίως μενού χρησιμοποιείται για αποθήκευση του χώρου στάθμευσης που έχει βρει ο χρήστης. Το κουμπί αυτό

χρησιμοποιείται σε περίπτωση που ο χρήστης έχει βρει ένα νέο χώρο που δεν υπήρχε στη βάση δεδομένων της εφαρμογής. Επιλέγοντας το ο χώρος αυτός εισάγεται στη βάση και εμφανίζεται στο χρήστη το μήνυμα “spot saved successfully”.

5. Στο κυρίως μενού υπάρχει επίσης το κουμπί  . Λειτουργία του είναι να εμφανίζει στο χρήστη το μενού κράτησης χώρου στάθμευσης. Όπου θα έχει την δυνατότητα αφού επιλέξει χώρο να κρατήσει τον χώρο πληρώνοντας online. Είναι δηλαδή παρκόμετρο τη διαφορά ότι η πληρωμή γίνεται μέσω visa card. Μέσω της χρήσης του ο χρήστης έχει ως πλεονέκτημα μειωμένες τιμές – εκπτώσεις – στο χώρο στάθμευσης.

6. Τέλος, ο χρήστης έχει την επιλογή να διαχειριστεί τις ρυθμίσεις της εφαρμογής επιλέγοντας  από το κυρίως μενού.

   Στο υπό-μενού των ρυθμίσεων έχει την δυνατότητα να ρυθμίσει:

   - Filters: Να επιλέξει δηλαδή πως θα φιλτράρονται οι χώροι στάθμευσης κατά την αναζήτηση του με βάση τις προτιμήσεις του. Πχ με βάση το κόστος, πόσο κοντά είναι στο προορισμό του κ.ο.κ.

   - Sounds: Να βάλει τις επιλογές του σε θέματα ήχων της εφαρμογής πχ ήχος αναζήτησης, ήχος πλήκτρων, intro sound κ.τ.λ.

   - Weather (on/off): Ο χρήστης με το button αυτό έχει την δυνατότητα να βλέπει τις καιρικές συνθήκες μέσω ενός gadget που θα του εμφανίζεται (αν είναι on), ουτοσώστε να τον βοηθήσει να αποφασίσει που θα σταθμεύσει το αυτοκίνητο του. Πχ σε περίπτωση που ξέρει ότι θα βρέξει στις ρυθμίσεις “filters” να επιλέξει να αναζητήσει χώρο στάθμευσης κοντά στο προορισμό του.

Τέλος, ο χρήστης μέσω του συνδέσμου “visit us here” θα έχει την δυνατότητα να ανεβεί απευθείας μέσω της εφαρμογής στην ιστοσελίδα Spark.

4.4 Wireframe και Flowchart εφαρμογής

Wireframe

Wireframe είναι ένα σημαντικό εργαλείο σχεδιασμού που χρησιμοποιείται στην ανάπτυξη μιας κινητής εφαρμογής. Είναι ένα εργαλείο οπτικοποίησης για την παρουσίαση των λειτουργιών, τη δομή και το περιεχόμενο της εφαρμογής. Με πιο απλά λόγια , είναι ένα εργαλείο που μας παρέχει την δυνατότητα να απεικονίσουμε την εμφάνιση και τη λειτουργία της εφαρμογής χωρίς να παρουσιάζει τον κώδικα που κρύβεται ενδιάμεσα. Με τον τρόπο αυτό μπορούμε να παρουσιάσουμε την εφαρμογή και τις λειτουργίες της με σκοπό να εξετάσουμε αν εξυπηρετεί τις ανάγκες του πελάτη.[9]

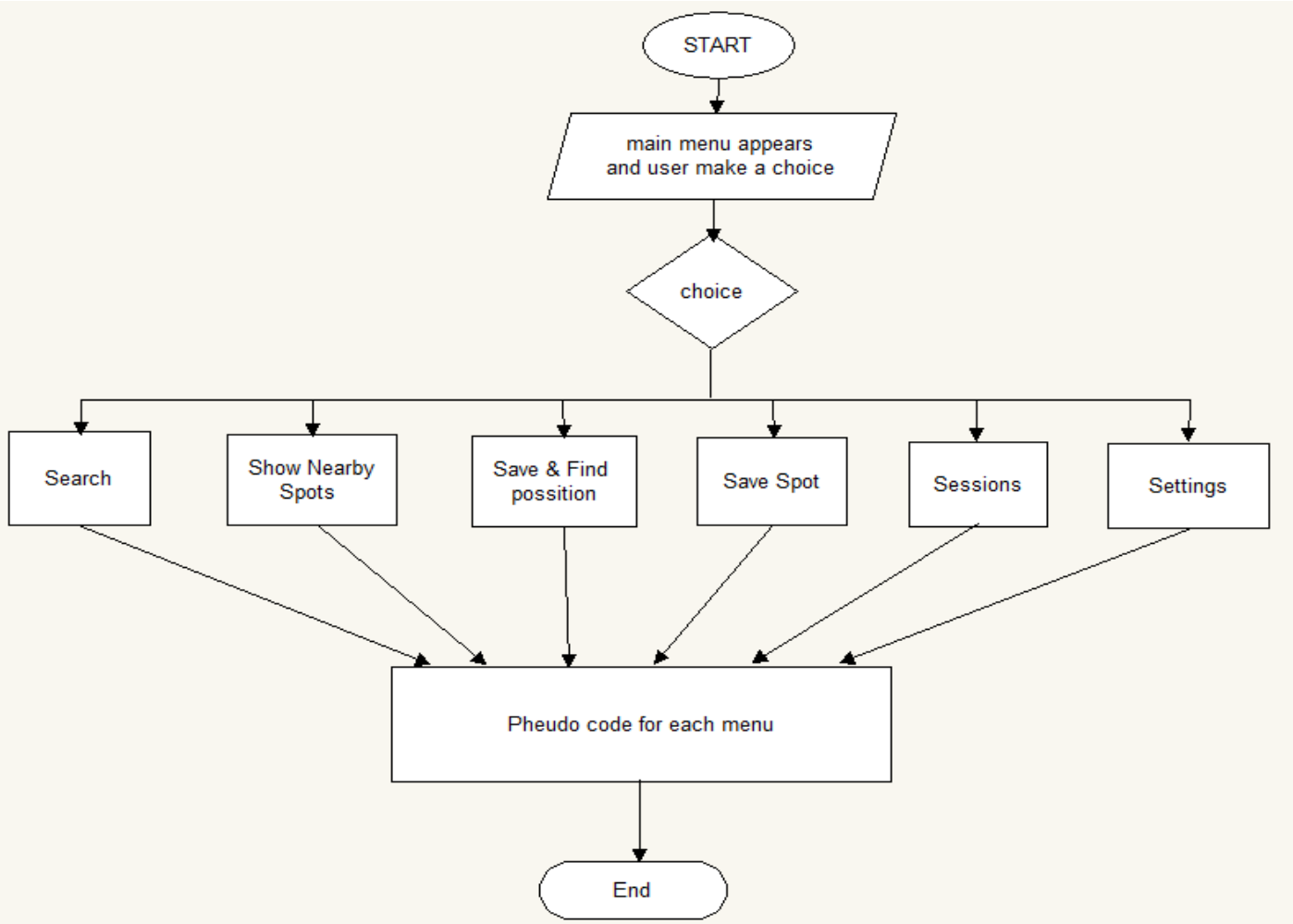
Flowchart

Πιο κάτω θα δούμε το Flowchart της εφαρμογής. Ένα Flowchart είναι ένα είδος διαγράμματος που αντιπροσωπεύει έναν αλγόριθμο, ροή εργασιών ή διαδικασία, δείχνοντας τα βήματα ως κουτιά διαφόρων ειδών, και τη σειρά εκτέλεσης τους μέσω της σύνδεσής τους με βέλη. Στην προκειμένη περίπτωση το flowchart παρουσιάζει τις λειτουργίες της εφαρμογής ως μια ροή από ενέργειες του χρήστη. Αλληλεπίδρασης του χρήστη με την εφαρμογή. Πρόκειται μέσο flowchart να εξετάσουμε στο πρώτο επίπεδο την εφαρμογή. Θα εξετάσουμε δηλαδή τις λειτουργίες της εφαρμογής από το κυρίως μενού. Η μελέτη των λειτουργιών σε βαθύτερα επίπεδα απεικονίζονται μέσω wireframe και με ψευδό-κώδικα σε μεταγενέστερο στάδιο. [10]

Spark Wireframe



Flowchart Spark (level 1)



Pseudo codes for app functions

(Internet access required)

1. Search
    - Connect to server
    - Get searching location
    - If searching location matches to database locations

        Show to screen nearby parkings

    Else

        Show to screen a msg (e.g. No parking spots found at searching location)

2. Show nearby spots
    - Get users location
    - If users location matches to database locations

        Show to screen nearby parkings

    Else

        Show to screen a msg (e.g. No parking spots found at your location)

3. Save and find position

If save / find button pressed once (or pressed odd number times 1, 3, 5)

    Save car location using its longitude and latitude

If save / find button pressed twice (or pressed even number times 2, 4, 6)

- Connect with GPS
- Use longitude and latitude to search the car

  If you find the car

      Show on maps its location

  Else

      Show on screen the msg “car not found”

4. Save spot

If save spot button pressed

- Connect with database
- Connect with GPS
- Get user's longitude and latitude

  If user's longitude and latitude exist on database

  Show on screen a msg "parking spot already exist in database"

  Else

  - Show on screen a msg "spot saved"
  - Add parking spot to database

5. Sessions

If user press session button

Show sessions menu

If user press extend session button (here no session created yet)

- Ask user for vehicle #plate, parking spot space number (if exist), start time (the time user wants for the session to start) , end on time (the time user wants for the session to end) , total cost
- Start session countdown (show the time remaining)

If user press again the session button

Ask if he want to extend session or end session

6. Settings

If "filters" button pressed

Show filters on screen (user number 1 to 3 filters by preferences, 1 most preferred)

If "sound" button pressed

Show sounds and notification menu on screen

User enables the sounds he wants

If button "back" pressed then save changes

If "weather" button pressed

Show weather widget on screen (make it visible)

Κεφάλαιο 5

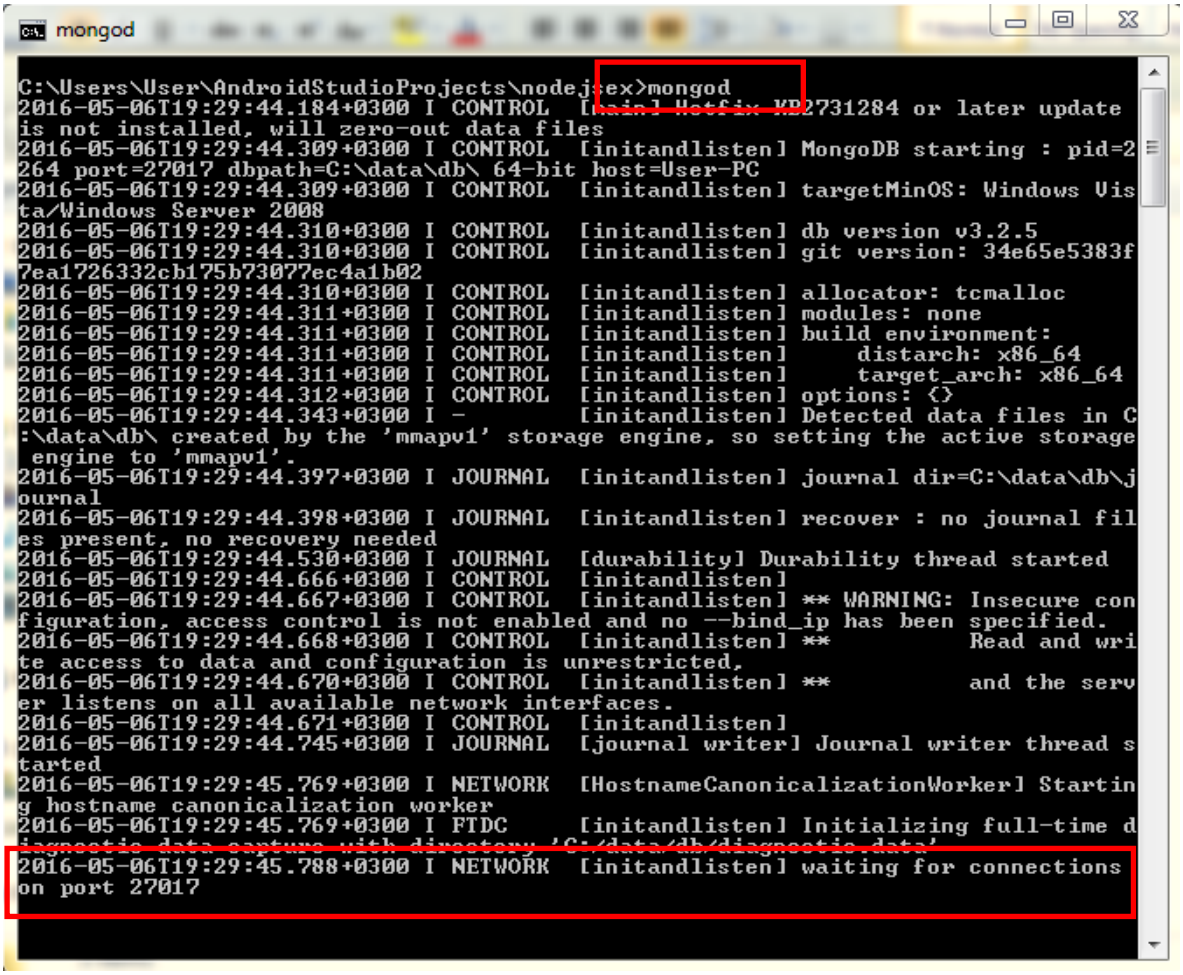
Υλοποίηση της εφαρμογής Spark

5.1	Σύνδεση στο Server	40
5.2	GUI εφαρμογή Spark	43
5.3	Το λειτουργικό σύστημα Android	46
5.4	Χάρτες Google	47

Στο κεφάλαιο αυτό θα δούμε την υλοποίηση της εφαρμογής Spark παρουσιάζοντας τη σύνδεση με τοπικό server και ακολούθως το γραφικό περιβάλλον (GUI) της εφαρμογής. Επίσης θα δούμε το android λειτουργικό σύστημα αναφέροντας τι είχαμε σχετικά με το android περιβάλλον χρησιμοποιήσει για την επιτυχής ανάπτυξη της εφαρμογής. Τέλος, θα μιλήσουμε σχετικά με τους χάρτες της Google, τους οποίους και χρησιμοποιήσαμε για τη παρουσίαση των δεδομένων μας (parking spots) στην εφαρμογή Spark.

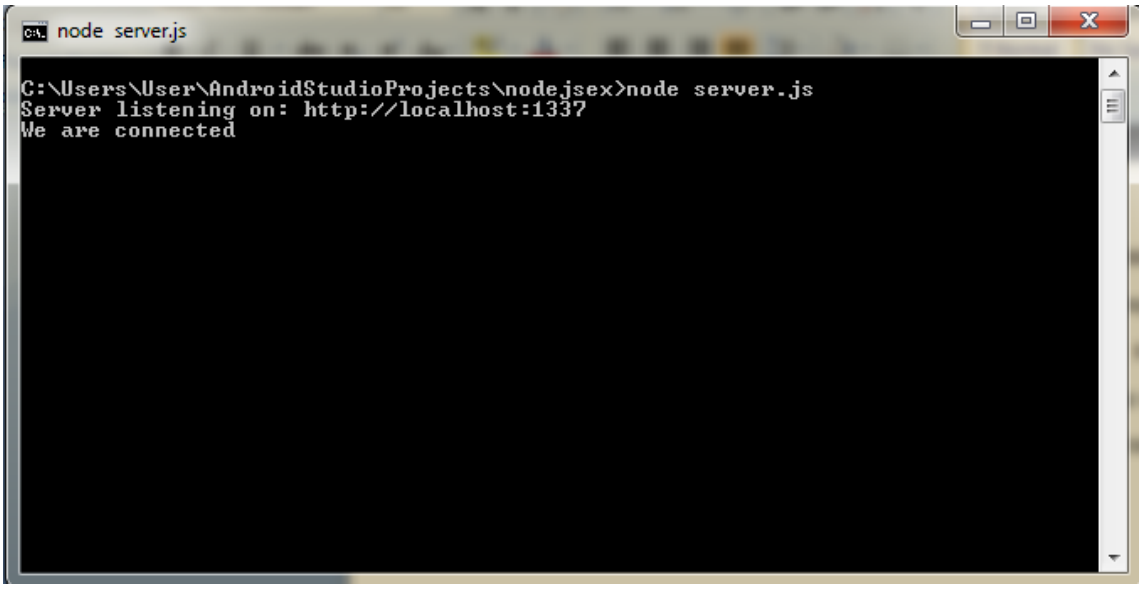
5.1 Σύνδεση στο Server

Αρχικά πριν τρέξουμε την εφαρμογή θα πρέπει να πάμε στο path που είναι αποθηκευμένος ο server.js και να ανοίξουμε ένα terminal στο σημείο αυτό γράφοντας “mongod” . Κάνοντας αυτό θα μας επιτρέψει αργότερα να κάνουμε σύνδεση στη βάση και θα μας δείχνει εκεί τις συνδέσεις που πρόκειται να ανοιχτούν στο port 27017. Στο terminal θα αναφέρει “[init and listen] waiting for connections on port 27017”. Σημείωση: Για να γίνει σωστά η διαδικασία αυτή πρέπει να υπάρχει αρκετός χώρος στο δίσκο μας (C://) όπου και βρίσκεται τοπικά η mongo βάση. Αν η διαδικασία αυτή γίνει επιτυχώς τότε θα εμφανίζεται στην οθόνη μας το terminal όπως πιο κάτω(terminal screen 1).



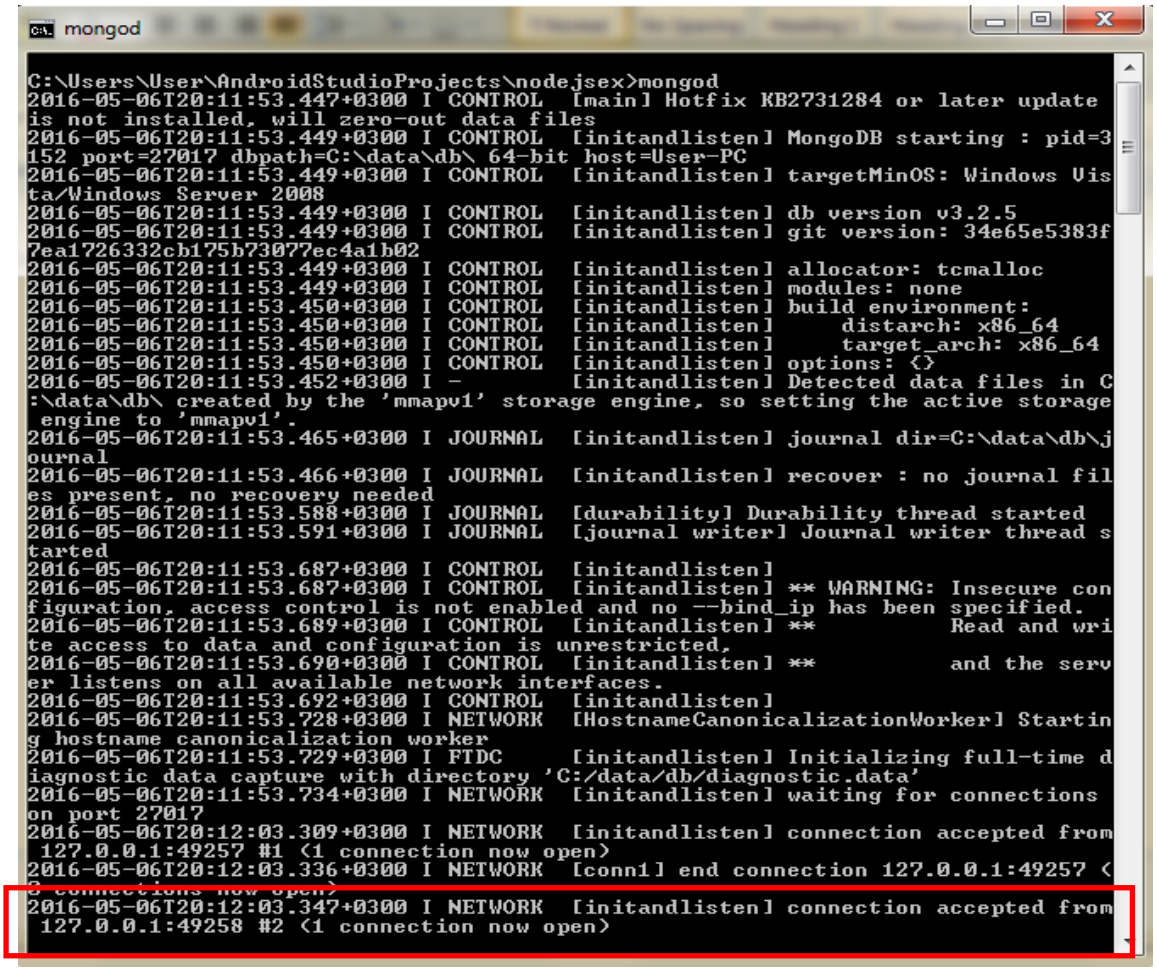
[terminal screen 1]

Ακολούθως, θα πρέπει να ανοίξουμε δεύτερο terminal στο ίδιο σημείο (path) και να γράψουμε “node server.js” . Κάνοντας αυτό τρέχουμε το server μας (χρησιμοποιώντας το node JS) ο οποίος και περιέχει το κώδικα που θα ζητήσει τα δεδομένα που χρειαζόμαστε από τη βάση τα οποία θα χειριστούμε αναλόγως μέσω της εφαρμογής. Αν η διαδικασία αυτή γίνει επιτυχώς τότε θα εμφανίζεται στην οθόνη μας το terminal όπως πιο κάτω “[terminal screen 2]”.



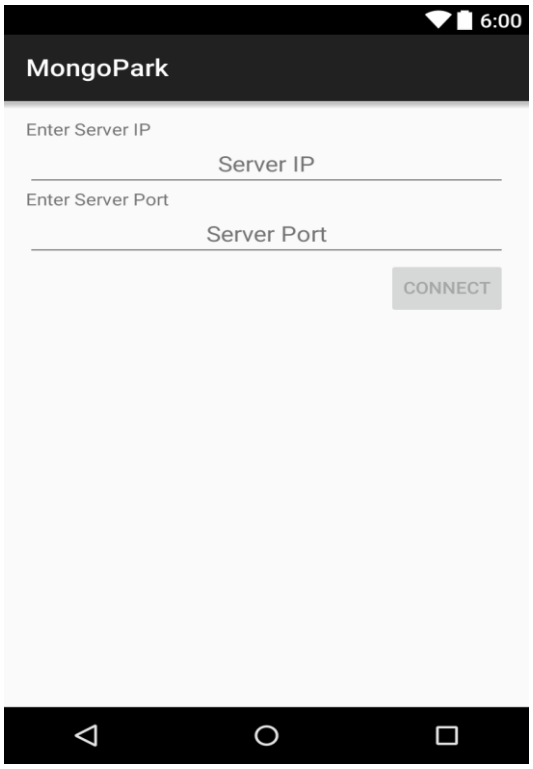
[terminal screen 2]

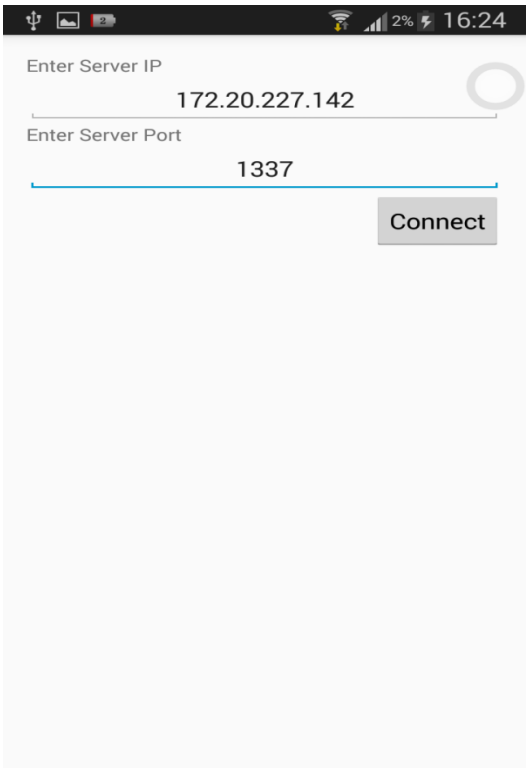
Και στην terminal screen 1 αφού τώρα έχουμε συνδεθεί θα μας εμφανίζει



5.2 GUI εφαρμογή Spark

Main activity

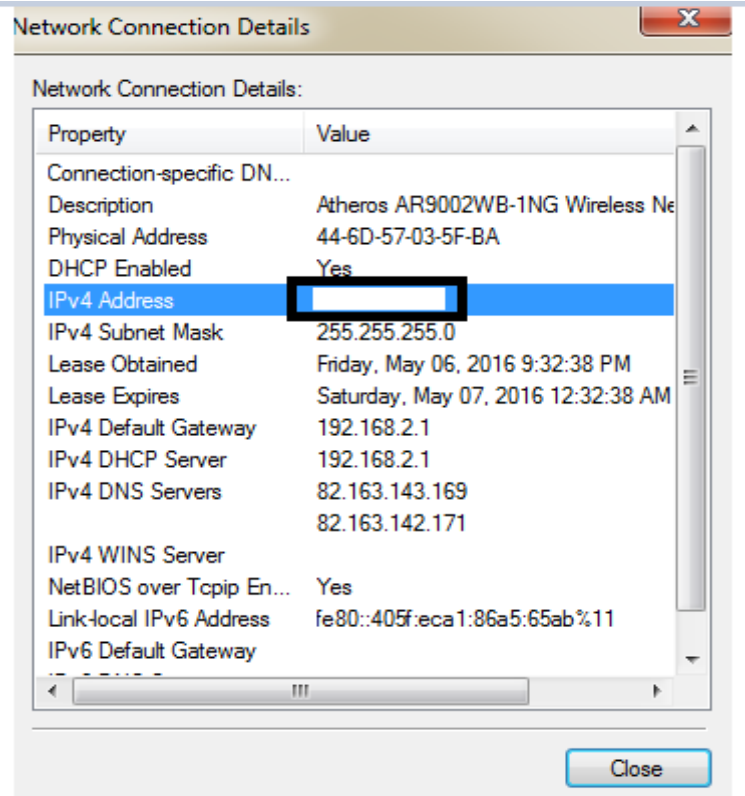




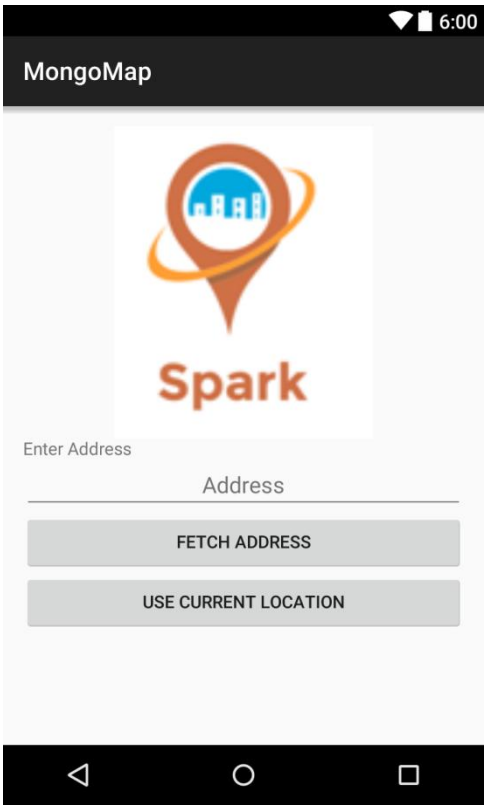
Πιο πάνω βλέπουμε το μενού που εμφανίζει στο χρήστη κατά την είσοδο του στην εφαρμογή. Ο χρήστης στο μενού αυτό πρέπει να εισάγει την IP διεύθυνση του στο πεδίο “Server IP” , ενώ στο πεδίο “Server Port” το port 1337. Εάν τα στοιχεία που έχει βάλει είναι ορθά το “connect” button θα ενεργοποιηθεί και θα δημιουργήσει σύνδεση με τη βάση , διαφορετικά τα στοιχεία που έχει βάλει θα αντικατασταθούν με κενό “ ”.

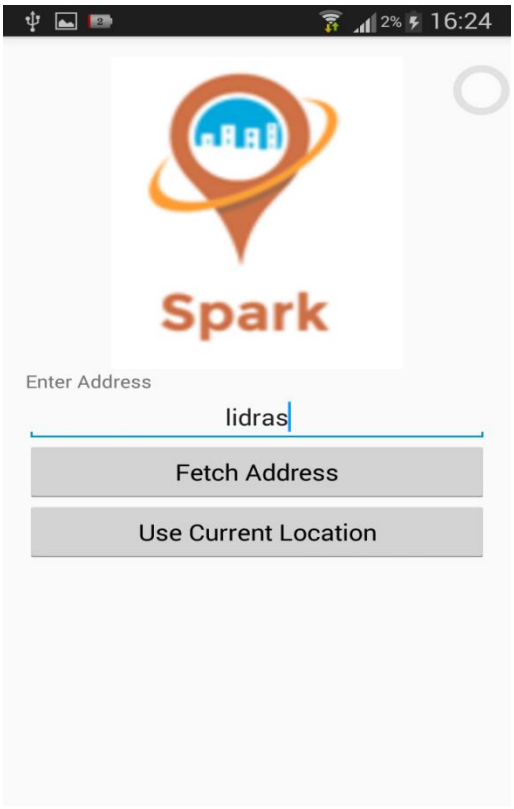
Σημείωση: Για να βρεις το IP address σου:

1. Δεξί κλικ στο εικονίδιο με τα διαθέσιμα networks
2. Open Network and Sharing Center
3. Connections: Wireless Network Connection
4. Details…
5. IPv4 address π.χ. 172.20.X.X

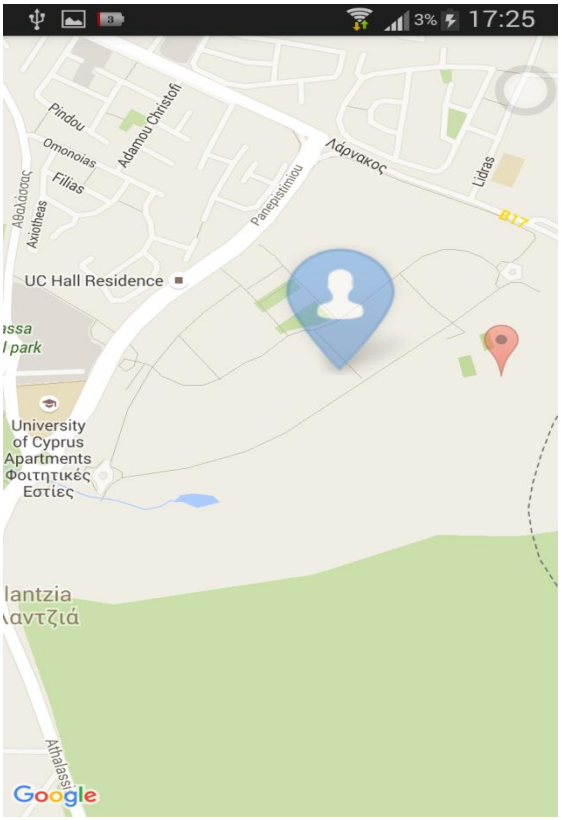


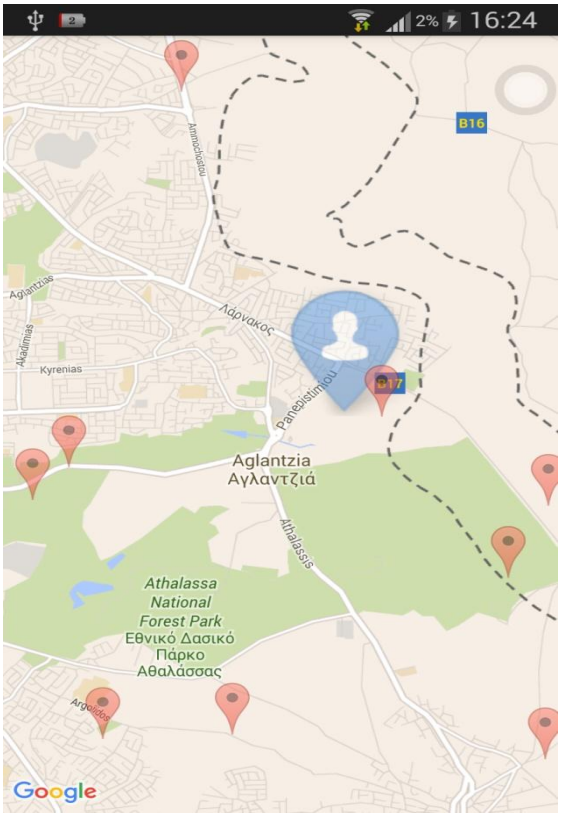
Spark activity:





Πιο πάνω βλέπουμε το κυρίως μενού της εφαρμογής όπου ο χρήστης μπορεί να δει τα διαθέσιμα parking στη περιοχή που βρίσκεται τώρα “Use Current Location”. Το αποτέλεσμα θα εμφανιστεί σε Google Maps όπως πιο κάτω.

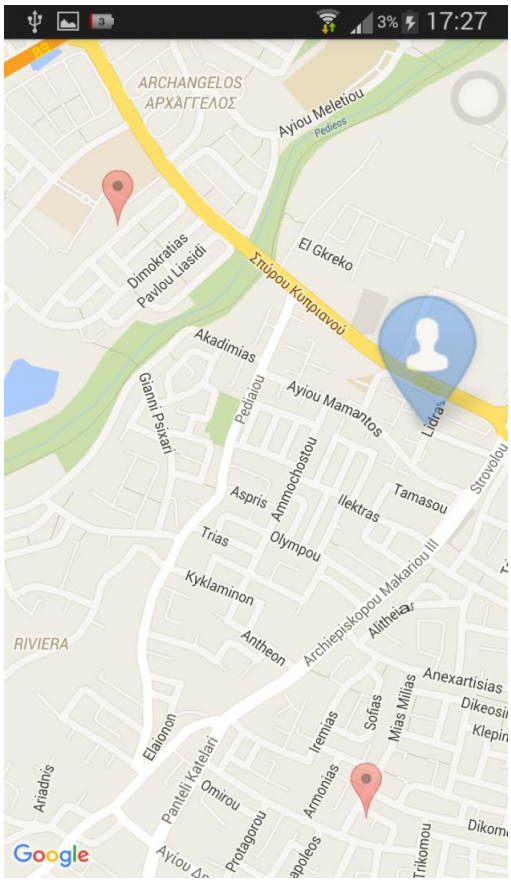




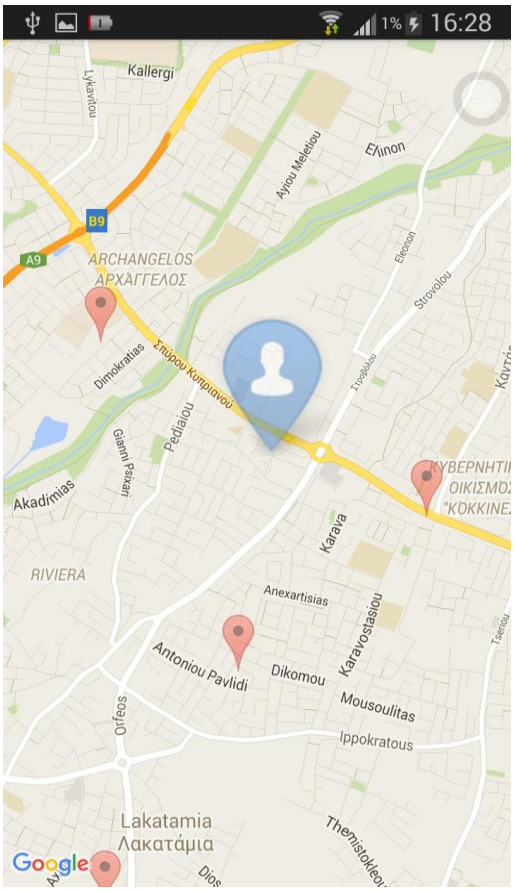
(zoom) (zoom out)

Η τοποθεσία που φαίνεται από τους χάρτες πιο πάνω είναι το Πανεπιστήμιο Κύπρου. Με μπλε χρώμα εμφανίζεται η τοποθεσία του χρήστη ενώ με κόκκινο τα διαθέσιμα παρκινγκ.

Η μπορεί να αναζητήσει διαθέσιμα παρκινγκ μιας περιοχή εισάγοντας το όνομα της π.χ. “Lidras” και επιλέγοντας το “Fetch Address”. Το αποτέλεσμα θα εμφανιστεί σε Google Maps όπως πιο κάτω.



(zoom)



(zoom out)

Με μπλε χρώμα εμφανίζεται η τοποθεσία αναζήτησης του χρήστη «Λήδρας» ενώ με κόκκινο τα διαθέσιμα παρκινγκ.

5.3 Το λειτουργικό σύστημα Android

Η εφαρμογή έχει υλοποιηθεί για χρήση σε συσκευές με λειτουργικό σύστημα Android. Ο προγραμματισμός έγινε σε γλώσσα Java και έπρεπε να χρησιμοποιηθεί ένα πολυάριθμο σύνολο προγραμμάτων τα οποία έπρεπε να συνδυαστούν όλα μεταξύ τους ώστε να λειτουργούν σαν ένα σύνολο. Ένα από τα βασικά εργαλεία που χρησιμοποιήθηκε είναι το Android Studio.

Το Android Studio αποτελεί ένα πολύ-γλωσσικό περιβάλλον ανάπτυξης λογισμικού που αποτελείται από ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) και ένα επεκτάσιμο plug-in σύστημα.

Το Android προσφέρει ένα plug-in το οποίο ονομάζεται Android Developer Tools (ADT). Έχει σχεδιαστεί για να προσφέρει ένα ενοποιημένο περιβάλλον στο οποίο είναι δυνατή η ανάπτυξη εφαρμογών για το Android. Η εγκατάστασή του επιτρέπει την εύκολη ρύθμιση και διαχείριση των εφαρμογών, την δημιουργία γραφικών διεπαφών και τον εντοπισμό σφαλμάτων.

Ακόμη ένα εργαλείο που χρησιμοποιήθηκε είναι το Android Development Kit (ADK). Πρόκειται για ένα σύνολο εργαλείων ανάπτυξης που επιτρέπει στους προγραμματιστές να δημιουργήσουν εφαρμογές για την πλατφόρμα Android. Το Android SDK (Software Development Kit) περιλαμβάνει έναν debugger, βιβλιογραφία, δείγματα κώδικα και τις απαιτούμενες βιβλιοθήκες για την κατασκευή Android εφαρμογών. Με κάθε νέα έκδοση του λειτουργικού συστήματος δημιουργείται και μια νέα έκδοση του SDK. Ένα μέρος της λειτουργικότητας που παρέχει το ADT plug-in οφείλεται στο development kit αυτό.

5.4 Χάρτες Google

Οι χάρτες Google, είναι μια διαδικτυακή εφαρμογή υπηρεσιών χαρτογράφησης. Η τεχνολογία αυτή παρέχεται δωρεάν για προσωπική χρήση από την Google. Ανάμεσα στις υπηρεσίες που προσφέρει, είναι η ιστοσελίδα των χαρτών Google, χάρτες ενσωματωμένους σε ιστοσελίδες μέσω των Google Maps API. Παρέχει οδικούς χάρτες, εφαρμογή δρομολόγησης για ταξίδι με τα πόδια, το αυτοκίνητο ή μέσα μαζικής μεταφοράς και εντοπισμό αστικών επιχειρήσεων για πολλές χώρες σε όλο τον κόσμο.

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

Το Android προσφέρει στους προγραμματιστές την δυνατότητα να ενσωματώσουν κάποιο χάρτη στην εφαρμογή τους. Αυτό μπορεί να γίνει μέσω χρήση ενός στοιχείου που ονομάζεται MapView. Το MapView σε συνδυασμό με την κλάση MapActivity και τις διεπαφές mapping APIs, είναι ένα ισχυρό εργαλείο στα χέρια οποιουδήποτε

προγραμματιστή που θέλει να δημιουργήσει καινοτόμες και πρωτοποριακές εφαρμογές με την χρήση των χαρτών. Με αυτό τον τρόπο ο κατασκευαστής μπορεί να χρησιμοποιήσει υπάρχων χάρτες και να τους τροποποιήσει όπως αυτός θέλει, με την δυνατότητα να χειριστεί τις αλληλεπιδράσεις του χρήστη με τον χάρτη, να τοποθετήσει δικά του δεδομένα πάνω σε αυτόν κλπ.

Το πακέτο για την χρησιμοποίηση των mapping APIs δεν ανήκει στο πλαίσιο του Android, αλλά σε αυτό της Google, γεγονός που πρέπει να δηλωθεί ξεχωριστά. Επιπρόσθετα, για την χρησιμοποίηση των χαρτών η Google απαιτεί την απόκτηση από τον εκάστοτε προγραμματιστή ενός map-api key, κλειδί το οποίο προσφέρει η ίδια. Για την απόκτηση του κλειδιού ο ενδιαφερόμενος πρέπει να δημιουργήσει λογαριασμό στη Google (sign in) και να εκτελέσει τα ακολουθα βήματα:

Get a Key

1. Go to the Google Developers Console.
2. Create or select a project.
3. Click **Continue** to enable the API and any related services.
4. On the **Credentials** page, get a **Browser key** (and set the API Credentials).
   Note: If you have an existing **Browser key**, you may use that key.
5. To prevent quota theft, secure your API key following these best practices.
6. (Optional) Enable billing. See Usage Limits for more information.

Διαδικασία απόκτησης API κλειδιού [11]

Κεφάλαιο 6

Συμπεράσματα

6.1 Συμπεράσματα και Μελλοντικές Επεκτάσεις 49

6.1 Συμπεράσματα και Μελλοντικές Επεκτάσεις

Το smart parking είναι ένα αναδυόμενο πεδίο στο ευρύτερο θέμα των Smart cities και αναμένουμε ότι θα εξελιχθεί ακόμα πιο γρήγορα στο μέλλον. Ο κόσμος θα μάθει να χρησιμοποιεί αυτά τα εύχρηστα προγράμματα για να ευκολύνει τη ζωή του και να μειώσει τον εκνευρισμό και το χάσιμο χρόνου που θα του προσφέρει το συνεχές ψάξιμο για χώρο στάθμευσης του οχήματος του. Θα μπορεί εύκολα και γρήγορα να βρίσκει το καλύτερο σύμφωνα με τα δικά του κριτήρια χώρο στάθμευσης μειώνοντας έτσι την κατανάλωση επιπλέον βενζίνης, μόλυνση της ατμόσφαιρας με τα καυσαέρια που αποβάλλει το όχημα του, επιπλέον χρήματα για πρόστιμο για παράνομη στάθμευση καθώς και αποφυγή κυκλοφοριακής συμφόρησης και δυστυχημάτων.

Τα Smart phones με τις δυνατότητες τους πλέον μπορούν να υποστηρίξουν τις υπάρχουσες εφαρμογές αλλά και πολλές άλλες βελτιστοποιημένες που πρόκειται να αναπτυχθούν για την καλύτερη εξυπηρέτηση του σκοπού της έξυπνης στάθμευσης.

Το μακροπρόθεσμο όραμα μας περιλαμβάνει την επέκταση του συστήματος του Smart parking σε παγκόσμιο επίπεδο ώστε εφαρμογές όπως το Spark να αποτελεί ένα χρήσιμο εργαλείο μέσω του οποίου κάποιος χρήστης θα μπορεί να χρησιμοποιήσει γλιτώνοντας κόπο και χρήματα.

Βιβλιογραφία

[1] Σύστημα στάθμευσης με ασύρματους αισθητήρες
https://www.google.com.cy/search?q=smart+parking+system+security&espv=2&biw=1366&bih=667&source=lnms&tbm=isch&sa=X&ved=0ahUKEwi6w8HLo5_LAhWI7hoKHV_UD00Q_AUIBygB#tbm=isch&q=smart+parking+&imgrc=ZSi3rT0lsWlfyM%3A

[2] Μηχανική συσκευή αποθήκευσης πολλαπλών αυτοκινήτων
https://www.google.com/search?q=smart+parking&hl=en&biw=1366&bih=667&source=lnms&tbm=isch&sa=X&ved=0ahUKEwibmrvboa_MAhWqJsAKHQ9jBk8Q_AUICCgC#imgrc=MXBX_G5BgTeI-M%3A

[3] Smart Parking PI: Robert C. Hampshire Research team: Daniel Jordon, Kats Sasanuma, Numeritics LLC
http://utc.ices.cmu.edu/utc/CMU%20Reports%202013%202/UTC_FinalReport_Smart_Parking_Hampshire.pdf

[4] Make your app more attractive to customers in the Windows Phone Store Read more at https://blogs.windows.com/buildingapps/2013/02/13/make-your-app-more-attractive-to-customers-in-the-windows-phone-store/#lZSHGjMYx37Adrwf.99
https://blogs.windows.com/buildingapps/2013/02/13/make-your-app-more-attractive-to-customers-in-the-windows-phone-store/

[5] App description
https://itunes.apple.com/us/app/parking-compass-free/id550089558?mt=8

[6] Limited Period Offer Title
http://topmobileappsdeveloper.over-blog.com/2015/07/tips-to-make-your-app-attractive-in-app-stores.html

[7] Spark Webpage: http://spark.cs.ucy.ac.cy/

[8] HTC PHONE ONE X+ Specs: https://www.google.com.cy/search?q=HTC+phone+one+x%2B&espv=2&biw=1366&bih=667&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiT1dKBps_LAhXEwBQKHfMBAIIQ_AUIBigB#tbm=isch&q=HTC+phone+one+x%2B+specs&imgrc=Ou1ET4AnxuY5YM%3A

[9] Wireframe: http://www.webopedia.com/TERM/W/wireframe.html

[10] Flowchart: **https://en.wikipedia.org/wiki/Flowchart**

[11] Get API key: https://developers.google.com/maps/documentation/javascript/get-api-key#get-an-api-key

Παράρτημα Α

Πίνακας με τα μοναδικά χαρακτηριστικά κάθε εφαρμογής

Λίστα με μοναδικά χαρακτηριστικά που έχει το κάθε ένα από τα top30topapps

Εφαρμογή	Μοναδικό χαρακτηριστικό – λόγος που το επέλεξα ως ένα από τα top
1. **Bring Me Back**	• it will help you to find your parked car, your hotel or a nice restaurant you have seen during your trip. • If the location information is not available (e.g., because you are in a parking garage), the position will be autmoatically set for a place, as soon as the location information is available again.
2. **ParkingTracker**	to save the parking position you simply press a button. Later when you want to find your car again you press another button in the car locator.
3. **ParkMeRight: Car Locator**	click on the "Find Car" magnifying glass button to find your way back to the car!(niceinterface)
4. **ARCarLocatorFree**	allows you to find your car by using augmented reality
5. **ParkingGuru**	Visualization of parking (underground and relay) as well as information on them (Rennes, France only)
6. **ParkopediaParking**	Users has a unique number and each user gain more coins by using the app. Coins will unlock new features such as “photos added” , “spaces added ” e.t.c. (make user to see the app as a game)
7. **Parker, Find available parking**	• Nice interface • Use filters to only show spots that meet your preferences
8. **Parkmobile**	• provide secure online access to your parking data so you can track your parking expenses, change your account settings or print a report. You can also easily export information for expense reimbursements

	• You'll find Parkmobile in private and public parking lots, on city streets, at airports, stadiums and just about anywhere you park
9. ParkmobileParking	update your personal details, change your payment method, download your invoice and view your parking history.
10. CarparkRates	• Car parking rates at over 700 locations in Singapore. • Traffic cameras to broadcast real-time traffic condition at major roads and checkpoints
11. AirportParking	Find parking at the airport
12. ParkNow	Save favorite parking garages •View and edit your ParkNow account •Pay safely and securely with automatic billing •Receive valet e-ticket straight to phone
13. Waze	You can also actively report accidents, police traps and other hazards you see on the road. Get road alerts along your route and find the cheapest gas prices around you shared by the community. Plus, you can add friends, send locations or keep others posted on your arrival time
14. BeattheTraffic	• Full map view with real-time traffic flow, cameras and incidents • Create a trip, for an at-a-glance personalized traffic report on one screen • Opt in for real-time traffic push notifications or email alerts • Share incidents with other commuters.
15. INRIX Traffic, Maps & Alerts	• See current traffic conditions, incidents, construction, police and road closures. • Traffic Cameras: See what's happening up ahead on your route. • Traffic News: trafficjams
16. HONK	Powerful & Reliable – immediate access to towing, tire change, jump start, fuel, and lock out service Roadsideassistance
17. iParking find my car using augmented reality	• using the GPS and the compass • usingaugmentedreality
18. ParkMyCar	When you park your car and just before get

	down, open the iPhone app and pres 'Park'. It will remember the location of your car parked. When you return, just open the app again and press 'Find'; it will show your car parked location on map.
19. ParkAround	• enables you to find parking spots at special discount rates up to 80% • used in PARKINGS in Greece
20. Parkbud - Parking& Travel Guide	helps you find and navigate to places and parking including parking garages around you while keeping track of your parking, paying for parking through your phone
21. ParkShark	(simple app) All you have to do is: 1) Choose a location where you want to park 2) Select a date and time when you want to start 3) Indicate how long you will park for 4) Choose a payment method
22. GatuparkeringStockholm	Displays: ✓ Regular parkings ✓ Indoor parkings ✓ Parkings for persons with reduced mobility ✓ MC parkings ✓ Bus parkings ✓ Truck parkings ✓ Diplomatic parkings
23. Truck ParkingEurope	Truck Parking Europe is the largest free European platform for truck parking facilities. The constantly growing, community-based parking database contains both officially designated truck and motorway parking places and car parks, as well as other suitable parking places, e.g. in industrial areas
24. SpotHero	• helps drivers find parking deals with garages, lots & valets • Find parking, compare prices & reserve a spot for up to 50% off the drive-up rate • Niceinterface
25. TripBuddy	GPS Weather Find my car
26. Findmycarsmarter	uses a new technology called Bluetooth Smart. When you start your car, the Find My Car Smarter App wirelessly connects to the Find My Car Smarter Device. After you have

	parked your car your engine and the Find My Car Smarter Device, powered by your cigarette lighter, turn off. The App realizes that your car is parked and then activates the GPS on your iPhone momentarily to save your car's location. If you lose your car, simply launch the app and it will help you navigate your way back by showing you your current location and the location of where you last parked your car.
27. NCPCarParkFinder	FREE GPS augmented reality ''find my car/where I parked'' function 24/7 customer telephone support UK airport parking link to online booking service
28. ParkingPin	uses the newest iPhones' Motion Coprocessors in combination with Location Services to pinpoint your spots as you park. It does this in the background, while your iPhone is asleep or as you use other apps. The next time you open the app, your current spot will be saved along with a list of your most recent spots.
29. MyAmpera	MY AMPERA AGENT Contact emergency services or your personal Ampera agent directly via telephone or email. CHARGING STATIONS Find your nearest electric vehicle charging stations and view current charge status, provider and telephone number for the station. AMPERA INFO Access the Ampera social networks (Facebook, Twitter and YouTube), Ampera Hub, 360 degree view of the vehicle and wallpapers.
30. NWSWeatherAlert	Alerts: Blowing Snow , Brisk Wind , Blizzard , Coastal Flood , Dust Storm, Blowing Dust ,Extreme Cold

Παράρτημα Β

SPARK app source code

main activity.java

```
package comp.cyteam.mongopark;

import android.app.Activity;
import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.os.AsyncTask;
import android.os.Bundle;
import android.text.Editable;
import android.text.InputFilter;
import android.text.Spanned;
import android.text.TextWatcher;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.Reader;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.URL;

public class MainActivity extends Activity implements View.OnClickListener {
    private static final String TAG = MainActivity.class.getSimpleName();
    private EditText serverIP, serverPort;
    private Button connect;
    private String mainURL = "";
    private String result = "NOT OK Connect";
    private boolean hasIP, hasPort;

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

        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        setContentView(R.layout.activity_main);

        this.serverIP = (EditText) findViewById(R.id.serverIP);
        this.serverPort = (EditText) findViewById(R.id.serverPort);
        this.connect = (Button) findViewById(R.id.buttonConnect);

        this.serverIP.setFilters(new InputFilter[]{new
InputFilterIPAddress()});
        this.serverPort.setFilters(new InputFilter[]{new InputFilterMinMax(1,
65535)});

        this.hasIP = false;
        this.hasPort = false;

        this.serverIP.addTextChangedListener(new TextWatcher() {
            @Override
```

```
            public void afterTextChanged(Editable arg0) {
                if
(MainActivity.this.serverIP.getText().toString().matches("^\\d{1,3}(\\.(\\d{1,
3}(\\.(\\d{1,3}(\\.(\\d{1,3})?)?)?)?)?)?")) {
                    MainActivity.this.hasIP = true;
                    if (MainActivity.this.hasPort == true) {
                        MainActivity.this.connect.setEnabled(true);
                    }
                }
            }

            @Override
            public void beforeTextChanged(CharSequence s, int start, int
count,
                                          int after) {
            }

            @Override
            public void onTextChanged(CharSequence s, int start, int before,
                                      int count) {
            }
        });

        this.serverPort.addTextChangedListener(new TextWatcher() {
            @Override
            public void afterTextChanged(Editable arg0) {
                if (MainActivity.this.serverPort.getText().toString().length()
> 1) {
                    MainActivity.this.hasPort = true;
                    if (MainActivity.this.hasIP == true) {
                        MainActivity.this.connect.setEnabled(true);
                    }
                }
            }

            @Override
            public void beforeTextChanged(CharSequence s, int start, int
count,
                                          int after) {
            }

            @Override
            public void onTextChanged(CharSequence s, int start, int before,
                                      int count) {
            }
        });

        this.connect.setEnabled(false);
        this.connect.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.buttonConnect) {
            this.connect.setEnabled(false);
            this.mainURL = "http://" + this.serverIP.getText() + ":" +
this.serverPort.getText() + "/";
            Log.i("MONGO URL", mainURL + "connect");
            AsyncTask dt = new ConnectTask().execute(mainURL + "connect");
        }
    }

    public class InputFilterMinMax implements InputFilter {

        private int min, max;

        public InputFilterMinMax(int min, int max) {
```

```
            this.min = min;
            this.max = max;
        }

        @Override
        public CharSequence filter(CharSequence source, int start, int end,
                                   Spanned dest, int dstart, int dend) {
            try {
                int input = Integer.parseInt(dest.toString()
                        + source.toString());
                if (isInRange(min, max, input))
                    return null;
            } catch (NumberFormatException nfe) {
            }
            return "";
        }

        private boolean isInRange(int a, int b, int c) {
            return b > a ? c >= a && c <= b : c >= b && c <= a;
        }
    }

    public class InputFilterIPAddress implements InputFilter {

        @Override
        public CharSequence filter(CharSequence source, int start, int end,
Spanned dest, int dstart, int dend) {
            if (end > start) {
                String destText = dest.toString();
                String resultText = destText.substring(0, dstart) +
source.subSequence(start, end) + destText.substring(dend);
                if (!resultText.matches
("^\\d{1,3}(\\.(\\d{1,3}(\\.(\\d{1,3}(\\.(\\d{1,3})?)?)?)?)?)?")) {
                    return "";
                } else {
                    String[] splits = resultText.split("\\.");
                    for (int i=0; i < splits.length; i++) {
                        if (Integer.valueOf(splits[i]) > 255) {
                            return "";
                        }
                    }
                }
            }
            return null;
        }
    }

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

        @Override
        protected String doInBackground(String... params) {
            //do your request in here so that you don't interrupt the UI
thread
            try {
                return downloadContent(params[0]);
            } catch (IOException e) {
                return "Unable to retrieve data. URL may be invalid.";
            }
        }

        @Override
        protected void onPostExecute(String result) {
            if(result.startsWith("OK")){
                Log.i("MONGO RESULT", result);
                Intent sparkIntent = new Intent(getApplicationContext(),
                        SparkActivity.class);
                sparkIntent.putExtra("serverIP",
```

```
MainActivity.this.serverIP.getText().toString());
                sparkIntent.putExtra("serverPort",
Integer.parseInt(MainActivity.this.serverPort.getText().toString()));
                MainActivity.this.startActivity(sparkIntent);
                MainActivity.this.connect.setEnabled(true);
                //Toast.makeText(MainActivity.this, "OK CONNECTED",
Toast.LENGTH_LONG);
            } else {
                MainActivity.this.serverIP.setText("");
                MainActivity.this.serverPort.setText("");
                MainActivity.this.connect.setEnabled(false);
                MainActivity.this.hasIP = false;
                MainActivity.this.hasPort = false;
            }
        }
    }

    private String downloadContent(String myurl) throws IOException {
        InputStream is = null;
        int length = 500;

        try {
            URL url = new URL(myurl);
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setReadTimeout(10000 /* milliseconds */);
            conn.setConnectTimeout(15000 /* milliseconds */);
            conn.setRequestMethod("GET");
            conn.setDoInput(true);
            conn.connect();
            int response = conn.getResponseCode();
            is = conn.getInputStream();

            // Convert the InputStream into a string
            String contentAsString = convertInputStreamToString(is, length);
            return contentAsString;
        } finally {
            if (is != null) {
                is.close();
            }
        }
    }

    public String convertInputStreamToString(InputStream stream, int length)
throws IOException, UnsupportedEncodingException {
        Reader reader = null;
        reader = new InputStreamReader(stream, "UTF-8");
        char[] buffer = new char[length];
        reader.read(buffer);
        return new String(buffer);
    }
}
```

spark_activity.java

```
package comp.cyteam.mongopark;

import android.app.Activity;
import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.location.Address;
import android.os.Bundle;
import android.os.Handler;
```

```
import android.os.ResultReceiver;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;


public class SparkActivity extends Activity implements View.OnClickListener {
    private static final String TAG = SparkActivity.class.getSimpleName();
    private AddressResultReceiver mResultReceiver;
    private EditText address;
    private Button current, fetch;
    private String serverIP;
    private int serverPort;
    private double lat ;
    private double lng ;
    private int fetchType = Constants.USE_ADDRESS_NAME;

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

        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        setContentView(R.layout.activity_spark);

        this.address = (EditText) findViewById(R.id.addressText);
        this.fetch = (Button) findViewById(R.id.buttonFetch);
        this.current = (Button) findViewById(R.id.buttonCurrent);

        this.serverIP = this.getIntent().getExtras().getString("serverIP");
        this.serverPort = this.getIntent().getExtras().getInt("serverPort");

        this.mResultReceiver = new AddressResultReceiver(null);

        this.fetch.setOnClickListener(this);
        this.current.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.buttonFetch) {
            /* TODO: Fetch Data */
            Intent intent = new Intent(this, GeoAddressIntentService.class);
            intent.putExtra(Constants.RECEIVER, mResultReceiver);
            intent.putExtra(Constants.FETCH_TYPE_EXTRA, fetchType);
            if(fetchType == Constants.USE_ADDRESS_NAME) {
                if(address.getText().length() == 0) {
                    Toast.makeText(this, "Please enter an address name",
Toast.LENGTH_LONG).show();
                    return;
                }
                intent.putExtra(Constants.LOCATION_NAME_DATA_EXTRA,
address.getText().toString());
            }
            Log.e(TAG, "Starting Service");
            startService(intent);
        } else if (v.getId() == R.id.buttonCurrent) {
            Intent mapIntent = new Intent(getApplicationContext(),
                    MapActivity.class);
            mapIntent.putExtra("serverIP", SparkActivity.this.serverIP);
            mapIntent.putExtra("serverPort", SparkActivity.this.serverPort);
            mapIntent.putExtra("location", "current");
            mapIntent.putExtra("latitude", Double.parseDouble("0.0"));
            mapIntent.putExtra("longitude", Double.parseDouble("0.0"));
            SparkActivity.this.startActivity(mapIntent);
        }
```

```
    }

    class AddressResultReceiver extends ResultReceiver {
        public AddressResultReceiver(Handler handler) {
            super(handler);
        }

        @Override
        protected void onReceiveResult(int resultCode, final Bundle
resultData) {
            if (resultCode == Constants.SUCCESS_RESULT) {
                final Address address =
resultData.getParcelable(Constants.RESULT_ADDRESS);
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        lat = address.getLatitude();
                        lng = address.getLongitude();

                        Intent mapIntent = new Intent(getApplicationContext(),
                                MapActivity.class);
                        mapIntent.putExtra("serverIP",
SparkActivity.this.serverIP);
                        mapIntent.putExtra("serverPort",
SparkActivity.this.serverPort);
                        mapIntent.putExtra("location", "useAddress");
                        mapIntent.putExtra("latitude", lat);
                        mapIntent.putExtra("longitude", lng);

                        Log.i("LAT", lat + " " + lng);
                        SparkActivity.this.startActivity(mapIntent);

                    }
                });
            }
        }
    }
}
```

Map activity.java

```
package comp.cyteam.mongopark;

import android.app.Activity;
import android.content.Context;
import android.content.pm.ActivityInfo;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;

import com.google.android.gms.maps.CameraUpdate;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.MapFragment;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;

import java.io.IOException;
```

```
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.Reader;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.ArrayList;
import java.util.List;


public class MapActivity extends Activity {
    private static final String TAG = MapActivity.class.getSimpleName();
    private LocationManager locationManager;
    private LocationListener locationListener;
    private GoogleMap map;
    private List<Marker> markers;
    private LatLng currentLocation;
    private CameraUpdate cameraUpdate;
    private String serverIP;
    private int serverPort;
    private String location;
    private String mainURL;
    private double lat;
    private double lng;
    private double rad = 0.05;

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

        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        setContentView(R.layout.fragment_map);

        this.markers = new ArrayList<Marker>();

        this.serverIP = this.getIntent().getExtras().getString("serverIP");
        this.serverPort = this.getIntent().getExtras().getInt("serverPort");
        this.location = this.getIntent().getExtras().getString("location");
        this.lat = this.getIntent().getExtras().getDouble("latitude");
        this.lng = this.getIntent().getExtras().getDouble("longitude");

        this.locationManager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);
        this.locationListener = new MyLocationListener();

        this.currentLocation = new LatLng(35.173756,    33.379378);

        if(this.location.equals("useAddress")) {
            this.currentLocation = new LatLng(this.lat, this.lng);
            this.mainURL = "http://" + this.serverIP + ":" +
Integer.toString(this.serverPort) + "/select?lat=" + this.lat + "&lng=" +
this.lng + "&rad=" + this.rad;
            Log.i("MONGO URL", mainURL);
            AsyncTask dt = new DownloadTask().execute(mainURL);
        } else {
            this.locationManager.requestLocationUpdates(
                    this.locationManager.getBestProvider(new Criteria(),
true), 0,
                    0, this.locationListener);
        }

        this.map = ((MapFragment) getFragmentManager().findFragmentById(
                R.id.mapFragment)).getMap();

        this.markers.add(this.map.addMarker(new MarkerOptions()
                .position(this.currentLocation).title("Your Location")
```

```
.icon(BitmapDescriptorFactory.fromResource(R.drawable.curr_icon))
                .alpha(0.5f).flat(true)));

        this.cameraUpdate = null;

        this.map.moveCamera(CameraUpdateFactory.newLatLngZoom(
                this.currentLocation, 15));



    }

    private class MyLocationListener implements LocationListener {


        @Override
        public void onLocationChanged(Location loc) {
            MapActivity.this.currentLocation = new LatLng(loc.getLatitude(),
                    loc.getLongitude());

MapActivity.this.markers.get(0).setPosition(MapActivity.this.currentLocation);
            MapActivity.this.map.moveCamera(CameraUpdateFactory.newLatLngZoom(
                    MapActivity.this.currentLocation,
                    MapActivity.this.map.getCameraPosition().zoom));
            MapActivity.this.mainURL = "http://" + MapActivity.this.serverIP +
":" + Integer.toString(MapActivity.this.serverPort) + "/select?lat=" +
MapActivity.this.currentLocation.latitude + "&lng=" +
MapActivity.this.currentLocation.longitude + "&rad=" + MapActivity.this.rad;
            Log.i("MONGO URL", mainURL);
            MapActivity.this.map.clear();
            MapActivity.this.markers.clear();
            MapActivity.this.markers.add(MapActivity.this.map.addMarker(new
MarkerOptions()
                    .position(MapActivity.this.currentLocation).title("Your
Location")

.icon(BitmapDescriptorFactory.fromResource(R.drawable.curr_icon))
                    .alpha(0.5f).flat(true)));
            AsyncTask dt = new DownloadTask().execute(mainURL);
        }

        @Override
        public void onProviderDisabled(String provider) {

        }

        @Override
        public void onProviderEnabled(String provider) {

        }

        @Override
        public void onStatusChanged(String provider, int status, Bundle
extras) {

        }
    }

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

        @Override
        protected String doInBackground(String... params) {
            //do your request in here so that you don't interrupt the UI
thread
            try {
                return downloadContent(params[0]);
            } catch (IOException e) {
```

```
                return "Unable to retrieve data. URL may be invalid.";
            }
        }

        @Override
        protected void onPostExecute(String result) {
            String results[] = result.split("\n");
            Log.i("MONGO DOWNLOAD", result);
            for(int i = 0; i < results.length - 1; i++){
                Log.i("MONGO DOWNLOAD", results[i].split(" ")[0] + "---" +
results[i].split(" ")[1] + "--" + i);

MapActivity.this.markers.add(MapActivity.this.map.addMarker(new
MarkerOptions()
                        .position(new
LatLng(Double.parseDouble(results[i].split(" ")[0]),
Double.parseDouble(results[i].split(" ")[1]))).title("Parking Available")
                        .alpha(0.5f).flat(true)));
            }
        }
    }

    private String downloadContent(String myurl) throws IOException {
        InputStream is = null;
        int length = 500;

        try {
            URL url = new URL(myurl);
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setReadTimeout(10000 /* milliseconds */);
            conn.setConnectTimeout(15000 /* milliseconds */);
            conn.setRequestMethod("GET");
            conn.setDoInput(true);
            conn.connect();
            int response = conn.getResponseCode();
            is = conn.getInputStream();

            // Convert the InputStream into a string
            String contentAsString = convertInputStreamToString(is, length);
            return contentAsString;
        } finally {
            if (is != null) {
                is.close();
            }
        }
    }

    public String convertInputStreamToString(InputStream stream, int length)
throws IOException, UnsupportedEncodingException {
        Reader reader = null;
        reader = new InputStreamReader(stream, "UTF-8");
        char[] buffer = new char[length];
        reader.read(buffer);
        return new String(buffer);
    }
}
```

Constants.java

```
package comp.cyteam.mongopark;


public final class Constants {
    public static final int SUCCESS_RESULT = 0;
```

```
    public static final int FAILURE_RESULT = 1;

    public static final int USE_ADDRESS_NAME = 1;
    public static final int USE_ADDRESS_LOCATION = 2;

    public static final String PACKAGE_NAME =
            "com.sample.foo.simplelocationapp";
    public static final String RECEIVER = PACKAGE_NAME + ".RECEIVER";
    public static final String RESULT_DATA_KEY = PACKAGE_NAME +
".RESULT_DATA_KEY";
    public static final String RESULT_ADDRESS = PACKAGE_NAME +
".RESULT_ADDRESS";
    public static final String LOCATION_LATITUDE_DATA_EXTRA = PACKAGE_NAME +
".LOCATION_LATITUDE_DATA_EXTRA";
    public static final String LOCATION_LONGITUDE_DATA_EXTRA = PACKAGE_NAME +
".LOCATION_LONGITUDE_DATA_EXTRA";
    public static final String LOCATION_NAME_DATA_EXTRA = PACKAGE_NAME +
".LOCATION_NAME_DATA_EXTRA";
    public static final String FETCH_TYPE_EXTRA = PACKAGE_NAME +
".FETCH_TYPE_EXTRA";

}
```

GeoAddressIntentService.java

```
package comp.cyteam.mongopark;

import android.app.IntentService;
import android.content.Intent;
import android.content.Context;
import android.location.Address;
import android.location.Geocoder;
import android.os.Bundle;
import android.os.ResultReceiver;
import android.text.TextUtils;
import android.util.Log;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.Locale;

/**
 * An {@link IntentService} subclass for handling asynchronous task requests
in
 * a service on a separate handler thread.
 * <p/>
 * TODO: Customize class - update intent actions, extra parameters and static
 * helper methods.
 */
public class GeoAddressIntentService extends IntentService {
    private static final String TAG = SparkActivity.class.getSimpleName();

    protected ResultReceiver resultReceiver;

    public GeoAddressIntentService() {
        super(TAG);
    }

    @Override
    protected void onHandleIntent(Intent intent) {
        Log.e(TAG, "onHandleIntent");
        Geocoder geocoder = new Geocoder(this, Locale.getDefault());
        String errorMessage = "";
        List<Address> addresses = null;
```

```
        int fetchType = intent.getIntExtra(Constants.FETCH_TYPE_EXTRA, 0);
        Log.e(TAG, "fetchType == " + fetchType);

        if(fetchType == Constants.USE_ADDRESS_NAME) {
            String name =
intent.getStringExtra(Constants.LOCATION_NAME_DATA_EXTRA);
            try {
                addresses = geocoder.getFromLocationName(name, 1);
            } catch (IOException e) {
                errorMessage = "Service not available";
                Log.e(TAG, errorMessage, e);
            }
        }
        else if(fetchType == Constants.USE_ADDRESS_LOCATION) {
            double latitude =
intent.getDoubleExtra(Constants.LOCATION_LATITUDE_DATA_EXTRA, 0);
            double longitude =
intent.getDoubleExtra(Constants.LOCATION_LONGITUDE_DATA_EXTRA, 0);

            try {
                addresses = geocoder.getFromLocation(latitude, longitude, 1);
            } catch (IOException ioException) {
                errorMessage = "Service Not Available";
                Log.e(TAG, errorMessage, ioException);
            } catch (IllegalArgumentException illegalArgumentException) {
                errorMessage = "Invalid Latitude or Longitude Used";
                Log.e(TAG, errorMessage + ". " +
                        "Latitude = " + latitude + ", Longitude = " +
                        longitude, illegalArgumentException);
            }
        }
        else {
            errorMessage = "Unknown Type";
            Log.e(TAG, errorMessage);
        }

        resultReceiver = intent.getParcelableExtra(Constants.RECEIVER);
        if (addresses == null || addresses.size()  == 0) {
            if (errorMessage.isEmpty()) {
                errorMessage = "Not Found";
                Log.e(TAG, errorMessage);
            }
            deliverResultToReceiver(Constants.FAILURE_RESULT, errorMessage,
null);
        } else {
            for(Address address : addresses) {
                String outputAddress = "";
                for(int i = 0; i < address.getMaxAddressLineIndex(); i++) {
                    outputAddress += " --- " + address.getAddressLine(i);
                }
                Log.e(TAG, outputAddress);
            }
            Address address = addresses.get(0);
            ArrayList<String> addressFragments = new ArrayList<>();

            for(int i = 0; i < address.getMaxAddressLineIndex(); i++) {
                addressFragments.add(address.getAddressLine(i));
            }
            Log.i(TAG, "Address Found");
            deliverResultToReceiver(Constants.SUCCESS_RESULT,
                    TextUtils.join(System.getProperty("line.separator"),
                            addressFragments), address);
        }
    }

    private void deliverResultToReceiver(int resultCode, String message,
```

```
Address address) {
        Bundle bundle = new Bundle();
        bundle.putParcelable(Constants.RESULT_ADDRESS, address);
        bundle.putString(Constants.RESULT_DATA_KEY, message);
        resultReceiver.send(resultCode, bundle);
    }
}
```

IPAddressKeyListenner.java

```
package comp.cyteam.mongopark;

import android.text.InputType;
import android.text.Spanned;
import android.text.method.NumberKeyListener;

public class IPAddressKeyListener extends NumberKeyListener {

    private char[] mAccepted;
    private static IPAddressKeyListener sInstance;

    @Override
    protected char[] getAcceptedChars() {
        return mAccepted;
    }

    private final char[] CHARACTERS =

            new char[] { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '.'
};

    private IPAddressKeyListener() {
        mAccepted = CHARACTERS;
    }

    /**
     * Returns a IPAddressKeyListener that accepts the digits 0 through 9,
     * plus the dot character, subject to IP address rules: the first
     * character has to be a digit, and no more than 3 dots are allowed.
     */
    public static IPAddressKeyListener getInstance() {
        if (sInstance != null)
            return sInstance;

        sInstance = new IPAddressKeyListener();
        return sInstance;
    }

    /**
     * Display a number-only soft keyboard.
     */
    public int getInputType() {
        return InputType.TYPE_CLASS_NUMBER
                | InputType.TYPE_NUMBER_FLAG_DECIMAL;
    }

    /**
     * Filter out unacceptable dot characters.
     */
    @Override
    public CharSequence filter(CharSequence source, int start, int end,
Spanned dest, int dstart, int dend) {
        if (end > start) {
            String destTxt = dest.toString();
```

```
            String resultingTxt = destTxt.substring(0, dstart) +
source.subSequence(start, end) + destTxt.substring(dend);
            if (!resultingTxt.matches
("^\\d{1,3}(\\.(\\d{1,3}(\\.(\\d{1,3}(\\.(\\d{1,3})?)?)?)?)?)?")) {
                return "";
            } else {
                String[] splits = resultingTxt.split("\\.");
                for (int i=0; i<splits.length; i++) {
                    if (Integer.valueOf(splits[i]) > 255) {
                        return "";
                    }
                }
            }
        }
        return null;
    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
    android:layout_height="match_parent"
android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
tools:context=".MainActivity">

    <TextView
        android:id="@+id/serverIPText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/server_ip" />

    <EditText
        android:id="@+id/serverIP"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/serverIPText"
        android:digits="0123456789."
        android:ems="10"
        android:focusable="true"
        android:focusableInTouchMode="true"
        android:gravity="center_horizontal"
        android:inputType="number|numberDecimal"
        android:labelFor="@+id/serverIP"
        android:hint="@string/server_ip_text"
        android:textAlignment="center" />

    <TextView
        android:id="@+id/serverPortText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/serverIP"
        android:text="@string/server_port" />

    <EditText
        android:id="@+id/serverPort"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/serverPortText"
```

```
        android:ems="10"
        android:gravity="center_horizontal"
        android:inputType="number"
        android:labelFor="@+id/ServerPort"
        android:maxLines="1"
        android:hint="@string/server_port_text" />

    <Button
        android:id="@+id/buttonConnect"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_connect"
        android:layout_below="@+id/serverPort"
        android:layout_alignRight="@+id/serverPort"
        android:layout_alignEnd="@+id/serverPort"
        android:enabled="false" />
</RelativeLayout>
```

Activity_spark.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
    android:layout_height="match_parent"
android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
tools:context=".SparkActivity">

    <ImageView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:id="@+id/imageView"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:src="@drawable/splogo"
        android:minHeight="250dp"
        android:maxHeight="250dp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/address"
        android:id="@+id/address"
        android:layout_below="@+id/imageView"
        android:layout_alignLeft="@+id/imageView" />


    <EditText
        android:id="@+id/addressText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/address"
        android:focusable="true"
        android:focusableInTouchMode="true"
        android:gravity="center_horizontal"
        android:labelFor="@+id/address"
        android:hint="@string/address_text"
        android:textAlignment="center" />
```

```
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/button_fetch"
        android:id="@+id/buttonFetch"
        android:layout_below="@+id/addressText"
        android:layout_centerHorizontal="true" />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/button_current"
        android:id="@+id/buttonCurrent"
        android:layout_below="@+id/buttonFetch"
        android:layout_centerHorizontal="true" />
</RelativeLayout>
```

Fragment_map.xml

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

    <fragment
        android:id="@+id/mapFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        class="com.google.android.gms.maps.MapFragment" />

</RelativeLayout>
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="comp.cyteam.mongopark" >

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

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <meta-data
```

```
            android:name="com.google.android.gms.version"
            android:value="@integer/google_play_services_version" />
        <meta-data
            android:name="com.google.android.maps.v2.API_KEY"
            android:value="AIzaSyABJyegFVdm8VbrtY_Kzl66rT5GYZCl6VY" />

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

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".MapActivity"
            android:label="@string/map_screen" >
        </activity>
        <activity
            android:name=".SparkActivity"
            android:label="@string/map_screen" >
        </activity>

        <service
            android:name=".GeoAddressIntentService"
            android:exported="false" >
        </service>
    </application>

</manifest>
```

Παράρτημα Γ

Server source code

```
//Lets require/import the HTTP module
var http = require('http');
var dispatcher = require('httpdispatcher');
var MongoClient = require('mongodb').MongoClient;

//Lets define a port we want to listen to
const PORT=1337;
const URL="mongodb://localhost:27017/parking15";

//Lets use our dispatcher
function handleRequest(request, response){
    try {
        //log the request on console
        console.log("Client Request: " + request.url);
        //Disptach
        dispatcher.dispatch(request, response);
    } catch(err) {
        console.log(err);
    }
}

//For all your static (js/css/images/etc.) set the directory name (relative path).
dispatcher.setStatic('resources');

//A sample GET request
dispatcher.onGet("/select", function(req, res) {
            var lat = parseFloat(req.params.lat);
    var lng = parseFloat(req.params.lng);
    var rad = parseFloat(req.params.rad);
    var result = "";

            if(!isFloat(lat)) {
                    console.log("Variable Error: Value for lat = " + lat + " is incorrect");
        res.writeHead(200, {'Content-Type': 'text/plain'});
        res.end("VARERROR: lat = " + lat + " is incorrect");
        return;
            }

    if(!isFloat(lat)) {
        console.log("Variable Error: Value for lng = " + lng + " is incorrect");
        res.writeHead(200, {'Content-Type': 'text/plain'});
        res.end("VARERROR: lng = " + lng + " is incorrect");
        return;
```

```
    }

    if(!isFloat(rad)) {
        console.log("Variable Error: Value for rad = " + rad + " is incorrect");
        res.writeHead(200, {'Content-Type': 'text/plain'});
        res.end("VARERROR: rad = " + rad + " is incorrect");
        return;
    }


    // Use connect method to connect to the Server
    MongoClient.connect(URL, function (err, db) {
        if (err) {
            console.log('Unable to connect to the mongoDB server. Error:', err);
        } else {
            //HURRAY!! We are connected. :)
            console.log('Connection established to', URL);

            // Get the documents collection
            var collection = db.collection('places');

            collection.find( { empty: true, loc:{$near : [lng, lat], $maxDistance: rad } }
).toArray(function(err, results){
                var i = 0;
                for(i = 0; i < results.length; i++){
                    result = result + results[i].loc[1] + " " + results[i].loc[0] + "\n";
                }
                console.log("Fetched data from Server: \n" + result);
                res.writeHead(200, {'Content-Type': 'text/plain'});
                res.end(result);
                db.close();
            });
        }
    });
});

function isFloat(n){
    return Number(n) === n && n % 1 !== 0;
}

dispatcher.onGet("/connect", function(req, res) {
    res.writeHead(200, {'Content-Type': 'text/plain'});
    res.end("OK Connect");
});

//A sample POST request
dispatcher.onPost("/post1", function(req, res) {
    res.writeHead(200, {'Content-Type': 'text/plain'});
    res.end('Got Post Data');
```

```
});

//Create a server
var server = http.createServer(handleRequest);

//Lets start our server
server.listen(PORT, function(){
   //Callback triggered when server is successfully listening. Hurray!
   console.log("Server listening on: http://localhost:%s", PORT);
   // Retrieve

        // Connect to the db
        MongoClient.connect(URL, function(err, db) {
                if(!err) {
                        console.log("We are connected");
                }
        });
});
```