



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

**Ανάπτυξη Πλατφόρμας Δυναμικής Παροχής Υπηρεσιών
Βασισμένων στη Θέση των Χρηστών**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Θεόδωρος Η. Μπούρχας

Επιβλέπων : **Μιχαήλ Θεολόγου**
Καθηγητής

Αθήνα, Σεπτέμβριος 2011



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάπτυξη Πλατφόρμας Δυναμικής Παροχής Υπηρεσιών Βασισμένων στη Θέση των Χρηστών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Θεόδωρος Η. Μπούρχας

Επιβλέπων : **Μιχαήλ Θεολόγου**

Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 9^η Σεπτεμβρίου 2011

.....
Μιχαήλ Θεολόγου
Καθηγητής Ε.Μ.Π

.....
Ευστάθιος Συκάς
Καθηγητής Ε.Μ.Π

.....
Στασινόπουλος Γεώργιος
Καθηγητής Ε.Μ.Π

Αθήνα, Σεπτέμβριος 2011

.....
Θεόδωρος Η. Μπούρχας

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Θεόδωρος Η. Μπούρχας, 2011.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Αφιερώνεται στην οικογένειά μου

Μπούρχας Θεόδωρος

Περιεχόμενα

Περίληψη.....	10
Summary.....	11
Κεφάλαιο 1: Υπηρεσίες Εντοπισμού Θέσης.....	13
1.1 Εισαγωγή στις Υπηρεσίες Εντοπισμού Θέσης.....	14
1.2 Τι είναι οι Υπηρεσίες Εντοπισμού Θέσης.....	14
1.3 Κύρια χαρακτηριστικά των LBS.....	16
1.4 Χρησιμότητα των Location-Based Services στις κύριες ενέργειες των χρηστών.....	16
1.5 Κατηγοριοποίηση των Location Based Services.....	18
1.6 Τι ιδιαίτερο έχουν οι Υπηρεσίες που βασίζονται στη Θέση του χρήστη(LBS);.....	20
1.7 Δομικά Μέρη των Υπηρεσιών που βασίζονται στη θέση του χρήστη(LBS).....	22
1.8 Αρχιτεκτονική Location Based Services και εξέλιξη αυτής.....	23
1.9 Πώς λειτουργούν τα συστήματα εντοπισμού θέσης;.....	27
1.10 Τεχνολογίες Προσδιορισμού Θέσης Χρήστη.....	29
1.11 Μέθοδοι βασιζόμενες στο δίκτυο (Network-based positioning).....	31
1.11.1 Μέθοδος αναγνώρισης ταυτότητας κυψέλης (Cell-Id).....	31
1.11.2 Μέθοδος του χρόνου άφιξης (Time of Arrival-TOA).....	32
1.11.3 Μέθοδος των διαφορών των χρόνων άφιξης(Time Difference of Arrival TDOA).....	33
1.11.4 Μέθοδος γωνίας άφιξης(Angle of Arrival-AOA).....	34
1.12 Μέθοδοι βασιζόμενες στο κινητό τερματικό(terminal-based positioning).....	35
1.12.1 Δορυφορικός Προσδιορισμός Θέσης(GPS).....	35
1.13 Υβριδικές Μέθοδοι προσδιορισμού της θέσης (hybrid technologies).....	36
1.13.1 Υποβοηθούμενο GPS (Assisted GPS –AGPS).....	36
1.14 Άλλες τεχνολογίες εντοπισμού θέσης.....	38
1.15 Συγκεντρωτική παρουσίαση των μεθόδων προσδιορισμού θέσης.....	40
1.16 Υπηρεσίες προσδιορισμού θέση(Location based Services) και Ιδιωτική ζωή των Mobile χρηστών.....	42
1.16.1 Πρόσφατες εξελίξεις στον τομέα της προστασίας της ιδιωτικής ζωής των κινητών χρηστών	42

1.16.2 Κυριότερα ζητήματα που απασχολούν τους χρήστες των LBS σχετικά με τα smartphones τους.....	44
1.16.3 Νομοθετικές ρυθμίσεις περί προστασίας των προσωπικών δεδομένων....	46
1.16.4 Τεχνικές προσέγγισης από τους developers για την παροχή ασφάλειας στη χρήση των LBS.....	48
1.16.5 Μοντέλα προστασίας των προσωπικών δεδομένων (privacy models).....	50
1.16.5.1 Location k-anonymity προσέγγιση.....	52
1.16.5.2 Transformation-based matching τεχνική.....	53
Κεφάλαιο 2 : Λογισμικό Android.....	54
2.1 Εξελίξεις στην αγορά των smartphones- Εμφάνιση του λογισμικού Android.....	55
2.2 Αρχιτεκτονική Android.....	56
2.3 Ανατομία μιας Android εφαρμογής.....	59
2.4 Android και Location Based Services.....	62
Κεφάλαιο 3 : Περιγραφή-Υλοποίηση Εφαρμογής.....	65
3.1 Αντικείμενο Διπλωματικής.....	66
3.2 Η εφαρμογή Viewer.....	66
3.3 Υλοποίηση Εφαρμογής.....	79
3.4 Ιστοσελίδα για την εισαγωγή προσφορών.....	89
3.5 Διάρθρωση του κώδικα της εφαρμογής Viewer.....	92
Παράρτημα: Πηγαίος Κώδικας της Εφαρμογής.....	93
Viewer.....	93
Server.....	117
Βιβλιογραφία.....	124

Περίληψη

Σε μια εποχή όπου τα κινητά τηλέφωνα παύουν να χρησιμοποιούνται μόνο για ομιλία και απλή αποστολή γραπτών μηνυμάτων και όπου παρατηρείται έξαρση της διακίνησης της πληροφορίας, οι άνθρωποι επιζητούν την πρόσβαση σε πηγές δεδομένων/ενημέρωσης οποιαδήποτε στιγμή και σε οποιοδήποτε τόπο και αν βρίσκονται. Για να ανταποκριθεί στις νέες απαιτήσεις η βιομηχανία της κινητής τηλεφωνίας προσπαθεί να συνδυάσει τις τεχνολογίες κινητής τηλεφωνίας και των υπολογιστικών συστημάτων και έχει καταφέρει να ενσωματώσει στην κινητή συσκευή λειτουργίες πλοήγησης στο διαδίκτυο και εφαρμογές που εκμεταλλεύονται την τρέχουσα θέση του χρήστη. Με αυτό τον τρόπο, η κινητή επικοινωνία αλλάζει προφίλ μεταβαίνοντας από την εποχή της απλής φωνητικής επικοινωνίας στην εποχή της πολύπλοκης ανταλλαγής δεδομένων.

Στα πλαίσια αυτής της μετάβασης, οι υπηρεσίες που βασίζονται στη θέση του χρήστη (LBS-Location Based Services) έχουν αρχίσει να έχουν ολοένα μεγαλύτερη απήχηση στο αγοραστικό κοινό και αυξανόμενο εμπορικό ενδιαφέρον για τις εταιρείες κινητών επικοινωνιών. Από υπηρεσίες παροχής πληροφοριών σε τουρίστες και δυνατότητες πλοήγησης σε μια περιοχή ή παρακολούθησης της κίνησης κάποιων ατόμων μέχρι υπηρεσίες έκτακτης ανάγκης και παροχής υπηρεσιών υγείας και παιχνίδια βασισμένα στη θέση του χρήστη/παίκτη, ο κλάδος των εφαρμογών που λειτουργούν στο επίπεδο ενός κινητού τερματικού και βασίζονται στην τρέχουσα θέση του χρήστη αναπτύσσεται διαρκώς.

Σημαντικός παράγοντας στην ανάπτυξη αυτή έπαιξε φυσικά η εμφάνιση του Android λογισμικού. Το Android είναι λογισμικό ανοικτού κώδικα, όπου οποιοσδήποτε χρήστης έχει τη δυνατότητα να σχεδιάσει τη δική του εφαρμογή, ανάλογα με τις ανάγκες του, προγραμματίζοντας ο ίδιος όποιο κομμάτι λογισμικού της κινητής συσκευής επιθυμεί. Με τη χρήση μάλιστα ειδικών βιβλιοθηκών που το Android έχει ενσωματωμένες στην πλατφόρμα του, η ανάπτυξη καινοτόμων εφαρμογών που βασίζονται στη θέση του χρήστη γίνεται εύκολη υπόθεση.

Έτσι, στα πλαίσια της συγκεκριμένης διπλωματικής εργασίας αναπτύχθηκε μια πλατφόρμα παροχής υπηρεσιών σε κινητούς χρήστες βάσει της θέσης στην οποία βρίσκονται, καθώς και των ενδιαφερόντων ή επιλογών τους. Η πλατφόρμα αυτή επιτρέπει σε παρόχους υπηρεσιών να συνδέονται και να τοποθετούν προσφορές για προϊόντα και υπηρεσίες, παρεχόμενα σε συγκεκριμένες θέσεις και για συγκεκριμένο χρονικό διάστημα. Η πλατφόρμα θα προτείνει και θα υποδεικνύει στους τελικούς χρήστες/καταναλωτές εκείνα τα προϊόντα ή υπηρεσίες που βρίσκονται πλησιέστερα στη τρέχουσα θέση τους και συμπίπτουν με τα ενδιαφέροντά τους.

Λέξεις Κλειδιά

Κινητά τηλέφωνα, χρήστες, υπηρεσίες βασισμένες στη θέση του χρήστη (LBS-location based services), Android λογισμικό, πλατφόρμα παροχής υπηρεσιών, προσφορές, προϊόντα, τρέχουσα θέση.

Abstract

In an era where mobile phones are not only used for voice communication and message sending, an increasing need of information flow is noticed. More and more people demand access on data anywhere and anytime. To meet this new requirements, mobile communications industry is forced to combine new mobile technologies with computing and has succeeded in implementing on the mobile device many new functionalities , such as web browsing and GPS applications. In this way, mobile communication gradually changes its trends, from simple voice communications to complex data communications.

In the frame of this change, Location Based Services(LBS) have an increasing influence on mobile users and a vast impact on mobile communications industry. From tourist-oriented information services, navigation, routing and tracking-enabled applications to emergency, safety and medical/health services and context-aware games, location-based services are developing quickly and at a large scale.

Android platform is a contributive factor on this enormous development of LBS. Android is the first open source mobile application platform, where any user, no matter what his development background is, can design its own application according to his needs and choices , by programming every hardware unit of the mobile terminal. With its powerful APIs and libraries, excellent documentation and a thriving developer community, the development of innovative location-based services becomes easier and easier.

The scope of this diploma thesis is the development of a platform which can provide mobile users with services based on their current location and on their interests and their choices. The platform will allow the service providers to connect and put special offers for products and services, on specific locations ad for specific time intervals. The platform will also suggest to users those products/services which are closer to their current location and are totally shared with their preferences and their requirements.

Key Words

Mobile devices, users, location-based services, Android platform, service-provided platform, offers, products/services, current location, preferences.

Κεφάλαιο 1

Υπηρεσίες Εντοπισμού Θέσης (Location Based Services)

1.1 Εισαγωγή στις Υπηρεσίες Εντοπισμού Θέσης

Τα κινητά τηλέφωνα και το Internet έχουν επιφέρει επανάσταση στις επικοινωνίες και μαζί στον τρόπο ζωής των ανθρώπων. Ένας ολοένα αυξανόμενος αριθμός κινητών τηλεφώνων, smart phones και Personal Digital Assistants(PDA) επιτρέπουν στους χρήστες την πρόσβαση στο Internet από όπου και αν βρίσκονται και όποτε το θελήσουν. Μέσω του Internet μπορούν να αποκτήσουν πληροφορίες σχετικά με διάφορα γεγονότα(συναυλίες, πάρτι, συγκεντρώσεις) καθώς και για τοποθεσίες (χάρτες πόλεων, εστιατόρια, μουσεία, νοσοκομεία).

Η ραγδαία εξέλιξη των κινητών τηλεπικοινωνιών έχει μετατοπίσει ωστόσο και το είδος των υπηρεσιών που προσφέρονται στους χρήστες, από υπηρεσίες μετάδοσης φωνής σε ποικίλες εφαρμογές μετάδοσης πληροφοριών μέσω πολυμέσων. Πλέον τα κινητά δεν αντιμετωπίζονται σαν απλές συσκευές τηλεφωνικής συνδιάλεξης ή αποστολής μηνυμάτων, αλλά ως εργαλεία παροχής πιο εξειδικευμένων υπηρεσιών εστιασμένων στις προσωπικές ανάγκες και στο προφίλ του κάθε χρήστη. Τέτοιες υπηρεσίες που χρησιμοποιούν τη γνώση της θέσης του χρήστη ονομάζονται Υπηρεσίες Εντοπισμού Θέσης(Location Based Services) και βρίσκονται ολοένα στο επίκεντρο του ενδιαφέροντος των παρόχων κινητών επικοινωνιών, οι οποίοι τις θεωρούν ως μέσον για να αναβαθμίσουν τις προσφερόμενες υπηρεσίες τους και να αυξήσουν τα έσοδά τους.

Η συζήτηση σχετικά με τη δυναμική και την προοπτική ενσωμάτωσης των LBS στα κινητά τηλέφωνα γίνεται εδώ και αρκετό καιρό, αλλά πλέον λαμβάνοντας υπόψη μας την αυξανόμενη στροφή στις εφαρμογές κινητών τηλεφώνων και τη μεγάλη διείσδυση των GPS κινητών συσκευών , παρατηρούμε πως η αγορά είναι πιο ώριμη από ποτέ για τέτοιες υπηρεσίες. Τα τελευταία χρόνια, τα τεχνολογικά επιτεύγματα έχουν αλλάξει το τοπίο και η βιομηχανία έχει αρχίσει να επενδύει σε αυτό το αναπτυσσόμενο περιβάλλον. Η πλήρης εμπορευματοποίηση των LBS αναμένεται να επέλθει σε σύντομο χρονικό διάστημα, αφού όλο και περισσότερες εμπορικές υπηρεσίες και εφαρμογές συμπεριλαμβανόμενων αυτών για έκτακτες ανάγκες, πλοήγηση, ενημέρωση και πληροφορίες με βάση την τοποθεσία, θα γίνονται διαθέσιμες στους χρήστες.

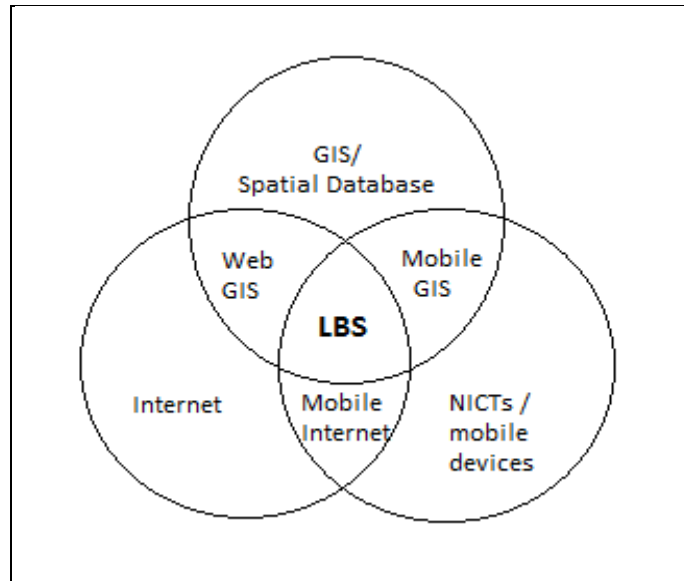
1.2 Τι είναι οι Υπηρεσίες Εντοπισμού Θέσης

Διάφοροι ορισμοί έχουν προταθεί για να εξηγήσουν το τι είναι Υπηρεσίες Εντοπισμού Θέσης(Location Based Services).

Ορισμός 1: Οι Location Based Services, είναι υπηρεσίες πληροφοριών(information services) που παρέχονται στους χρήστες των δικτύων κινητών και προσωπικών επικοινωνιών και χρησιμοποιούν/εκμεταλλεύονται τη γνώση της θέσης του κινητού τερματικού.

Ορισμός 2: Είναι wireless-IP υπηρεσίες που χρησιμοποιούν τις γεωγραφικές συντεταγμένες του χρήστη για να προσθέσουν αξία και για να βελτιώσουν τη σχετικότητα και το περιεχόμενο των πληροφοριών που θα του παρέχουν.

Από τους παραπάνω ορισμούς γίνεται σαφές πως οι LBS είναι η σύγκλιση τριών τεχνολογιών(σχήμα 1). Προέρχονται από την «τομή» των Νέων Τεχνολογιών Πληροφόρησης και Επικοινωνιών(New Information and Communication Technologies-NICTs), όπως το σύστημα των κινητών τηλεπικοινωνιών και οι συσκευές κινητής τηλεφωνίας, του Internet και των Συστημάτων Γεωγραφικής Πληροφόρησης(Geographic Information Systems-GIS) σε συνδυασμό με βάσεις χωρικών δεδομένων(spatial databases).



Σχήμα 1. LBS ως συνδυασμός τριών τεχνολογιών

Από ιστορικής πλευράς, πληροφορίες βασισμένες στη θέση των χρηστών δεν εμφανίζονται για πρώτη φορά με την εφεύρεση των κινητών τηλεφώνων. Πληροφορίες σχετικές με τη θέση μεταφέρονται εύκολα στην καθημερινή επικοινωνία μεταξύ των ανθρώπων με γκράφιτι ή post-it σημειώματα. Επίσης, οι αφίσες ή οι διαφημιστικές ταμπέλες στο δρόμο είναι τρόποι μαζικής πληροφόρησης όσων περνούν από την περιοχή εκείνη. Όμως, η επικοινωνία αυτή είναι μονόδρομη.

Οι LBS δίνουν τη δυνατότητα μιας αμφίδρομης επικοινωνίας και αλληλεπίδρασης με το χρήστη. Αυτό συμβαίνει γιατί ο χρήστης ουσιαστικά ενημερώνει τον πάροχο της υπηρεσίας για το πραγματικό περιεχόμενο που τον ενδιαφέρει, δηλαδή το είδος των πληροφοριών που χρειάζεται, τις προτιμήσεις του και τη θέση του. Αυτό βοηθά τον πάροχο τέτοιων υπηρεσιών, βασισμένων στην τοποθεσία, να διανέμει τις πληροφορίες με τέτοιο τρόπο ώστε να ταιριάζουν ακριβώς στις ανάγκες του χρήστη και να περιορίζονται στο χώρο που τον περιβάλλει.

Ας φανταστούμε πόσο χρήσιμο θα ήταν να καλέσουμε ένα ταξί από το αεροδρόμιο χωρίς να ξέρουμε τον τοπικό αριθμό και κωδικό της περιοχής ή να καλέσουμε ένα ασθενοφόρο αν γινόμασταν μάρτυρες ενός ατυχήματος σε ένα δρόμο που δεν ξέρουμε με το πάτημα μόνο ενός κουμπιού στο κινητό μας τηλέφωνο. Επίσης, πόσο εξυπηρετικό θα ήταν να μπορούσαμε να βρίσκαμε κάποια γρήγορη διαδρομή χωρίς κίνηση εν μέσω ενός μποτιλιαρίσματος ή να βρίσκαμε το πλησιέστερο χώρο για παρκάρισμα χρησιμοποιώντας το κινητό μας τηλέφωνο και την ανάλογη location-based υπηρεσία. Οι LBS μπορούν να βγάλουν από το αδιέξοδο χρήστες που αναζητούν κάποιο εφημερεύον φαρμακείο, ή ένα πρατήριο βενζίνης ή κάποιο εστιατόριο στην περιοχή γύρω από την τρέχουσα θέση τους και μάλιστα να δώσουν και σαφείς οδηγίες για το πώς θα φτάσει εκεί. Αυτές είναι μόνο μερικές από τις πολυάριθμες υπηρεσίες τέτοιου είδους που διαφοροποιούνται από τις κοινές ασύρματες υπηρεσίες στο ότι χρησιμοποιούν όλες τη γνώση της τοποθεσίας (location-awareness) και διανέμουν πληροφορίες στο χρήστη σύμφωνες με τις ανάγκες του και τις προτιμήσεις του.

Για να έχει πρόσβαση ο χρήστης στις υπηρεσίες αυτές , αρκεί να είναι συνδεδεμένος στο διαδίκτυο μέσω του δικτύου του παρόχου κινητών τηλεπικοινωνιών ή μέσω των διαθέσιμων ασύρματων υποδομών που υπάρχουν στο περιβάλλον του. Ακόμη, σημαντικό χαρακτηριστικό των LBS είναι ότι οι χρήστες δεν χρειάζεται να υποβάλλουν τα στοιχεία τους ούτε οποιαδήποτε αναγνωριστικά θέσης όταν περιπλανώνται κατά τη διάρκεια χρήσης των υπηρεσιών αυτών , καθώς το σύστημα αντιλαμβάνεται αυτόματα που βρίσκεται ο χρήστης.

1.3 Κύρια χαρακτηριστικά των LBS

Οι υπηρεσίες εντοπισμού θέσης(LBS) χειρίζονται δεδομένα σχετικά με την τοποθεσία , χρησιμοποιούν συναρτήσεις ανάλυσης χωρικών δεδομένων και δίνουν απαντήσεις σε ερωτήματα, όπως:

- α)Πού είμαι;
- β)Τι βρίσκεται τριγύρω μου;
- γ)Πώς θα μεταβώ σε μια συγκεκριμένη περιοχή;

Είναι σχετικά πρόσφατες τεχνολογίες και απευθύνονται σε ένα ευρύ πεδίο χρηστών, χωρίς εμπειρία σε τέτοια συστήματα και χωρίς να χρησιμοποιούν τις υπηρεσίες αυτές για καθαρά επαγγελματικούς σκοπούς.

Επίσης, πρέπει να σημειωθεί πως οι LBS είναι υπηρεσίες που αναπτύσσονται και λειτουργούν υπό τους περιορισμούς που τίθενται από το υπολογιστικό περιβάλλον και το λειτουργικό σύστημα της κινητής συσκευής. Δηλαδή, για να είναι μια εφαρμογή λειτουργική και κατ' επέκταση εκμεταλλεύσιμη από το αγοραστικό κοινό πρέπει να προσαρμοστεί στα παρακάτω:

- α) Χαμηλή υπολογιστική ισχύς (όσο επιτρέπει η κινητή συσκευή), οπότε πρέπει να γίνει προσπάθεια για δέσμευση όσο λιγότερων υπολογιστικών πόρων της συσκευής.
- β)Λειτουργία σε μικρή οθόνη , οπότε πρέπει να ληφθεί σοβαρά η μικρή επιφάνεια των κινητών τηλεφώνων για προβολή και αποτύπωση των δεδομένων της εφαρμογής.
- γ)Λειτουργία εξαρτημένη από το χρόνο ζωής της μπαταρίας του τηλεφώνου.

1.4 Χρησιμότητα των Location-Based Services στις κύριες ενέργειες των χρηστών

Η χρησιμότητα των LBS έγκειται στο ότι σχεδιάζονται ακριβώς για να ικανοποιήσουν τις ανάγκες των χρηστών. Δεδομένου ότι οι χρήστες, όταν βρεθούν σε ένα περιβάλλον άγνωστο για αυτούς, συμπεριφέρονται με παρόμοιο τρόπο και έχουν προβλέψιμες ανάγκες, κάνει τις LBS να εστιάζουν στην ικανοποίηση συγκεκριμένων ενεργειών και στο να παρέχουν πληροφορίες που ανταποκρίνονται στις απαιτήσεις των ενδιαφερομένων.

Έτσι, είναι χρήσιμο να ομαδοποιήσουμε τις τυπικές ενέργειες και δραστηριότητες των χρηστών, ώστε να έχουμε μια γενική εικόνα των κυριότερων επιθυμιών τους βασισμένων πάντα στην τρέχουσα θέση τους, καθώς και να δείξουμε πώς οι LBS ανταποκρίνονται στα αιτήματά τους.






Η πιο προφανής ερώτηση των χρηστών κατά τη διάρκεια της κίνησής τους είναι να μάθουν που βρίσκονται σε σχέση με κάποιον ή σε σχέση με κάτι(εντοπισμός-*locating*).

Επίσης, μπορεί να ψάχνουν για πρόσωπα, αντικείμενα ή γεγονότα (αναζήτηση-**searching**) και να ρωτούν πληροφορίες για το πώς θα πάνε εκεί(πλοήγηση-**navigating**).

Άλλοι χρήστες μπορεί να αναζητούν πληροφορίες για μια περιοχή(**identifying**) ή να τσεκάρουν για γεγονότα που διαδραματίζονται σε μια τοποθεσία ή κοντά σε μια περιοχή(**checking**).

Σαν μια γενική παρατήρηση, αξίζει να σημειωθεί πως για τις ενέργειες του locating και του navigating απαιτούνται γεωγραφικές πληροφορίες με βάση τη θέση του χρήστη και μόνο τέτοιες, ενώ οι τρεις άλλες κατηγορίες χρειάζονται επιπρόσθετα με τις πληροφορίες τοποθεσίας και άλλου είδους πληροφορίες, όπως στατικές πληροφορίες(static information), τοπικές πληροφορίες(topical information) που μπορεί να αλλάζουν κατά τη διάρκεια της κίνησης του χρήστη καθώς και πληροφορίες ασφαλείας(safety information).

Στον παρακάτω πίνακα παρουσιάζονται συνοπτικά οι κύριες ενέργειες των χρηστών, τα ερωτήματα που προσπαθούν να δώσουν απάντηση και οι λειτουργίες που αναπτύσσονται από τις LBS ώστε να απαντήσουν στα ερωτήματα αυτά.

	Ενέργεια Χρήστη	Ερωτήματα Χρήστη	Λειτουργίες
	Προσανατολισμός και εντοπισμός θέσης (orientation and locating)	Πού είμαι; Πού βρίσκεται τι; (πρόσωπο/αντικείμενο)	Προσδιορισμός Θέσης, Εύρεση συντεταγμένων με χρήση άλλων γεωγραφικών στοιχείων, όπως ονόματα δρόμων κτλ (geocoding) και αντίστροφα (degeocoding).
	Πλοήγηση(Navigation)	Πώς φτάνω σε μία τοποθεσία όταν γνωρίζω το όνομά της ή τη διεύθυνση ή τις συντεταγμένες της.	Προσδιορισμός Θέσης, Geocoding / Degeocoding, Δρομολόγηση.
	Αναζήτηση (ανθρώπων ή αντικειμένων)(Search)	Πού είναι το πλησιέστερο ή το πιο σχετικό με τις ανάγκες και επιθυμίες μου πρόσωπο/αντικείμενο;	Προσδιορισμός Θέσης, Υπολογισμός απόστασης και ζώνης, Εύρεση σχέσεων μεταξύ ανθρώπων.
	Αναγνώριση ατόμων και αντικειμένων (Identification)	Τι/Ποιος /Πόσο είναι εδώ/εκεί;	
	Τσεκάρισμα γεγονότων (Event Checking)	Τι συμβαίνει εδώ/εκεί;	

Πίνακας 1.Κύριες Ενέργειες χρηστών

1.5 Κατηγοριοποίηση των Location Based Services

Διάφορες προσεγγίσεις έχουν προταθεί σχετικά με την ταξινόμηση των LBS εφαρμογών σε κατηγορίες. Γενικά, μπορούν να διαφοροποιηθούν σε υπηρεσίες προσανατολισμένες ως προς το χρήστη (*person-oriented LBS*) και σε υπηρεσίες που εστιάζουν στο κινητό τερματικό (*device-oriented LBS*).

Person-oriented Services: Τέτοιου είδους εφαρμογές εστιάζουν στο να προσδιορίσουν τη θέση του χρήστη ή να χρησιμοποιήσουν την τρέχουσα θέση του χρήστη ώστε να ενεργοποιήσουν μια συγκεκριμένη υπηρεσία. Συνήθως, ο χρήστης μπορεί να ελέγξει την ενεργοποίηση ή μη της εφαρμογής.

Device-Oriented Services: Είναι εφαρμογές εξωτερικές στο χρήστη. Έτσι παρόλο που μπορούν να εστιάσουν στη θέση του χρήστη, εντούτοις δεν το χρειάζονται. Αντί για ένα πρόσωπο ή ένα αντικείμενο, τέτοιου είδους εφαρμογές μπορούν να εντοπίσουν τη θέση ενός συνόλου ανθρώπων. Συνήθως, ο χρήστης δεν μπορεί να ελέγξει την εφαρμογή αυτή από το κινητό του τερματικό.

Μια δεύτερη κατηγοριοποίηση των LBS ,ανάλογα με το αν η πληροφορία που διατίθεται στο χρήστη γίνεται μετά από δική του αλληλεπίδραση ή όχι, χωρίζει τις LBS σε *υπηρεσίες Pull* και σε *υπηρεσίες Push*.

Pull Services: Στην κατηγορία των υπηρεσιών pull, ο χρήστης του κινητού αιτείται την εξυπηρέτησή του από κατάλληλη υπηρεσία σχετική με τη θέση του, κινώντας τη διαδικασία προσδιορισμού της θέσης του. Αυτό είναι όμοιο με το άνοιγμα μιας ιστοσελίδας στο Διαδίκτυο όπου πληκτρολογούμε αυτοβούλως το url της ιστοσελίδας που θέλουμε να επισκεφτούμε στο πεδίο του web-browser. Δηλαδή , ο χρήστης ενεργά ζητά την ενεργοποίηση και χρήση μιας εφαρμογής, με την έννοια ότι τραβά πληροφορίες από το δίκτυο εκούσια. Μπορούμε να διαχωρίσουμε και περαιτέρω τις pull services σε “*λειτουργικές*” *υπηρεσίες(functional services)*, όπως η κλήση ενός ταξί ή ενός ασθενοφόρου με το πάτημα μόνο ενός κουμπιού στο κινητό τερματικό και σε *υπηρεσίες πληροφόρησης(information services)*, όπως η αναζήτηση του πλησιέστερου εστιατορίου, ξενοδοχείου κτλ.

Push Services: Στην κατηγορία των υπηρεσιών αυτών, ο χρήστης λαμβάνει πληροφορίες σχετικές με το περιβάλλον γύρω από την τρέχουσα θέση του, χωρίς να τις έχει ζητήσει ενεργά. Οι πληροφορίες αυτές μπορεί να έχουν σταλεί στο χρήστη είτε με προηγούμενη συγκατάθεσή του, δηλαδή ο ίδιος να είναι προεγγεγραμμένος σε επιλεγμένες υπηρεσίες/εφαρμογές , είτε χωρίς την έγκρισή του(όπως πχ ένα διαφημιστικό μήνυμα που έρχεται στο κινητό του μόλις εισέλθει σε μια καινούρια περιοχή). Και στις δύο περιπτώσεις ο εξυπηρετητής της συγκεκριμένης εφαρμογής εκτελεί αναζήτηση στο σύστημα προσδιορισμού θέσης που εξυπηρετεί το συγκεκριμένο δίκτυο για χρήστες που πληρούν ορισμένα κριτήρια, γεωγραφικά και άλλα.

Στον παρακάτω πίνακα παρουσιάζονται συνοπτικά κάποια παραδείγματα εφαρμογών για τις κατηγορίες των LBS που ορίσαμε παραπάνω.

	Push Services	Pull Services
Person-Oriented Services		
Επικοινωνία	<ul style="list-style-type: none"> • Ειδοποίηση από μια υπηρεσία εύρεσης φίλων ότι κάποιος φίλος του χρήστη έχει μόλις εισέλθει στην κοντινή περιοχή του. • Ένα μήνυμα προωθείται στο χρήστη ρωτώντας τον αν επιτρέπει σε ένα φίλο του να τον εντοπίσει. 	Ο χρήστης ζητά από μια εφαρμογή εύρεσης φίλων να μάθει ποιοι φίλοι του βρίσκονται κοντά του.
Πληροφόρηση	Ο χρήστης ειδοποιείται για μια τρομοκρατική απειλή που λαμβάνει χώρα στην πόλη που μόλις εισήλθε.	Ο χρήστης αναζητά το πλησιέστερο σινεμά στην περιοχή που βρίσκεται και οδηγίες πλοήγησης για να μεταβεί εκεί.
Διασκέδαση		Ο χρήστης παίζει ένα location-based παιχνίδι και ψάχνει για κάποιον άλλο παίκτη ώστε να του «επιτεθεί».
Εμπόριο και Διαφημίσεις	Ένας κατάλογος με προσφορές ηλεκτρονικά κουπόνια και εκπτώσεις στέλνεται στον χρήστη από ένα εστιατόριο μόλις προσεγγίσει εκείνη την περιοχή.	Ο χρήστης ψάχνει για εκδηλώσεις, συναυλίες και άλλα γεγονότα που λαμβάνουν χώρα στην περιοχή που βρίσκεται.
Device-Oriented Services		
Εύρεση Ίχνους(Tracking)	Ο χρήστης παίρνει μια ειδοποίηση πως το παιδί του έχει φύγει από το σχολείο.	Ο χρήστης ζητά πληροφορίες για το που βρίσκεται το όχημά του.

Πίνακας 2. Κατηγορίες και παραδείγματα LBS εφαρμογών.

1.6 Τι ιδιαίτερο έχουν οι Υπηρεσίες που βασίζονται στη Θέση του χρήστη(LBS);

Οι υπηρεσίες που βασίζονται στη θέση του χρήστη διαφέρουν από τα συμβατικά φυλλάδια/εφημερίδες και τα ιντερνετικά μέσα ενημέρωσης και πληροφόρησης επειδή έχουν γνώση του γενικού πλαισίου (*context*) στο οποίο πρέπει να κινηθούν και να ανταποκριθούν και έτσι μπορούν να προσαρμόσουν το περιεχόμενό τους, το τι θα παρουσιάσουν και πως θα το προβάλλουν ανάλογα. Οι LBS μπορούν και εστιάζουν σε διάφορους τύπους περιεχομένου που ο χρήστης έμμεσα ή άμεσα γνωστοποιεί στους παρόχους των υπηρεσιών αυτών. Έτσι, οι LBS για να παρέχουν στους χρήστες υπηρεσίες και πληροφορίες άμεσα συνδεδεμένες με τις απαιτήσεις τους και τις προτιμήσεις του, πρέπει να λάβουν υπόψη μια σειρά από διαφορετικές πληροφορίες που λαμβάνουν από τους χρήστες και το περιβάλλον γύρω τους. Παρακάτω παρουσιάζονται τα διαφορετικά «πλαίσια» περιεχομένου που επιδρούν και διαμορφώνουν μια location-based service.



Σχήμα 2. Διαφορετικοί τύποι περιεχομένου που επηρεάζουν το γενικό πλαίσιο που θα κινηθούν οι LBS

Χαρτογράφηση του κινητού χρήστη: Η ταυτότητα του χρήστη είναι σημαντική γιατί επιτρέπει στην υπηρεσία να γνωρίζει:

- Την ηλικία και το φύλο των χρηστών , οπότε για παράδειγμα να ξέρει πως σε χρήστες-παιδιά δεν χρειάζεται να προβάλλει pubs ή μπαρ.
- Τις προσωπικές προτιμήσεις τους, όπως για παράδειγμα τη γλώσσα που θέλουν να έχει η εφαρμογή/ υπηρεσία όταν προβάλλεται στο κινητό τερματικό.
- Ποιοι είναι οι φίλοι ή οι συνάδελφοί τους, εάν θελήσουν να βγουν ή να φικά συνεργαστούν μαζί τους.

Θέση χρήστη: Η τοποθεσία/τρέχουσα θέση του χρήστη είναι το πιο βασικό στοιχείο που πρέπει να λάβει υπόψη της μια LBS, γιατί με αυτό τον τρόπο οι πληροφορίες και οι υπηρεσίες θα παρέχονται με γνώμονα γεωγραφικά κριτήρια. Η θέση του χρήστη μπορεί να είναι απόλυτη, δηλαδή να περιγραφεί μέσω γεωγραφικών συντεταγμένων ή σχετική, όπως για παράδειγμα ένα δωμάτιο μέσα σε ένα κτίριο.

Χρόνος: Ο χρόνος μπορεί να αναφέρεται είτε σε μια συγκεκριμένη χρονική στιγμή της μέρας είτε σε ένα χρονικό διάστημα, όπως ένα πρωινό, ένα απόγευμα, μια μέρα της εβδομάδας κτλ. Γενικά, στις υπηρεσίες διασκέδασης ο χρόνος μπορεί να χρησιμοποιηθεί για να καθοριστεί αν ένα γεγονός που πρόκειται να προβληθεί στο χρήστη είναι ακόμα έγκυρο ή όχι.

Προσανατολισμός: Ο προσανατολισμός του χρήστη είναι σημαντικός για μια LBS για να καθοριστεί η κατεύθυνση στην οποία κινείται ο χρήστης και επομένως τι βρίσκεται μπροστά του ή και γύρω του. Για παράδειγμα, η γνώση αυτής της πληροφορίας θα ήταν χρήσιμη για μια υπηρεσία για τουρίστες που θέλουν να μάθουν ποιο ιστορικό μνημείο βρίσκεται μπροστά τους ή κοντά τους, καθώς επίσης και για μια υπηρεσία πλοήγησης που ελέγχει αν ο χρήστης πηγαίνει στη σωστή κατεύθυνση ή όχι.

Ιστορικό Πλοήγησης: Το ιστορικό πλοήγησης είναι μια πληροφορία που μπορεί να αξιοποιηθεί από τις LBS ώστε να δώσουν στους χρήστες τη δυνατότητα να δουν σε ποια μέρη έχουν βρεθεί, τι έχουν δει και τι έχουν κάνει. Η πληροφορία αυτή μπορεί να είναι χρήσιμη σε περιπτώσεις που ο χρήστης έχει χαθεί και θέλει να επιστρέψει εκεί που βρισκόταν. Τέλος, το ιστορικό πλοήγησης βοηθά τους παρόχους των LBS να σχηματίσουν ένα προφίλ για τον κάθε χρήστη και τις προτιμήσεις τους και έτσι να διανέμουν πιο σχετικές πληροφορίες και εστιασμένες στις πραγματικές τους ανάγκες.

Πρόθεση χρήσης της LBS: Το πώς προτίθεται να χρησιμοποιήσει ο χρήστης μια συγκεκριμένη υπηρεσία/εφαρμογή καθορίζεται από τους στόχους του, τις δραστηριότητές του, τις επιθυμίες του κτλ. Διαφορετικοί τρόποι χρήσης απαιτούν διαφορετικές πληροφορίες, διαφορετικούς τρόπους αλληλεπίδρασης και τέλος διαφορετικούς τρόπους παρουσίασης, για παράδειγμα είτε προβολή σε χάρτη, είτε παρουσίαση με κείμενο, είτε με φωνητική ειδοποίηση.

Κοινωνικό & Πολιτιστικό Περιβάλλον: Το κοινωνικό προφίλ του χρήστη χαρακτηρίζεται από το πόσο κοντά είναι με τους άλλους, από τις κοινωνικές του σχέσεις και από τις σχέσεις συνεργασίας του με τους άλλους. Τέτοιες πληροφορίες μπορούν να χρησιμοποιηθούν κατάλληλα από location-based εφαρμογές.

Για παράδειγμα, ένας χρήστης μπορεί να θέλει να ακολουθήσει το πλήθος για να βρει γνωστά μέρη για βραδινή έξοδο ή να αποφύγει κάποιους ή απλά να θέλει να βρει τους φίλους του.

Φυσικό Περιβάλλον: Το φυσικό περιβάλλον επιδρά κυρίως στον τρόπο που παρουσιάζεται και λειτουργεί η εφαρμογή στο κινητό τερματικό. Έτσι, παράγοντες όπως το επίπεδο φωτισμού και ο θόρυβος που επικρατεί στη γύρω περιοχή πρέπει να λαμβάνονται υπόψη από την εφαρμογή ή από την ίδια τη συσκευή και να γίνεται η ανάλογη προσαρμογή (πχ προσαρμογή του φωτισμού και του contrast της οθόνης ώστε ο χρήστης να μπορεί να διαβάσει το περιεχόμενο της σε οποιεσδήποτε καιρικές συνθήκες).

Δυνατότητες Συστήματος: Ο τύπος της συσκευής που χρησιμοποιεί ο χρήστης και οι λειτουργικές της δυνατότητες(οθόνη αφής, έγχρωμη ή ασπρόμαυρη, ανάλυση οθόνης, επεξεργαστής, μνήμη), η δυνατότητα συνεχούς πρόσβασης στο Internet ή όχι, η ταχύτητα της σύνδεσης και η ποιότητα προσδιορισμού της τρέχουσας θέσης που έχει άμεση σχέση με το GPS δέκτη του κινητού τερματικού είναι πληροφορίες που θα πρέπει να λαμβάνει υπόψη του ο πάροχος των LBS , ώστε να προσφέρει την υπηρεσία στο χρήστη με το βέλτιστο τρόπο.

Η γνώση όλων των παραπάνω πληροφοριών ουσιαστικά δημιουργεί ένα γενικότερο πλαίσιο στο οποίο προσαρμόζονται οι LBS δυναμικά τόσο σε επίπεδο παρεχόμενων πληροφοριών, όσο και σε επίπεδο παρουσίασης στο χρήστη καθώς και σε τεχνικό επίπεδο ώστε να προσαρμόζονται στα χαρακτηριστικά της δεδομένης συσκευής.

1.7 Δομικά Μέρη των Υπηρεσιών που βασίζονται στη θέση του χρήστη(LBS)

Τα δομικά στοιχεία που αποτελούν μια υπηρεσία βασισμένη στη θέση του χρήστη είναι πέντε και η σύνδεσή τους φαίνεται παρακάτω:

Κινητό τερματικό(Mobile Devices):

Το εργαλείο-συσκευή που χρησιμοποιεί ο χρήστης για να ζητήσει την επιθυμητή πληροφορία. Τα αποτελέσματα επιστρέφονται είτε σε μορφή ηχητικής εγγραφής είτε γίνεται χρήση εικόνων και κειμένου.

Τηλεπικοινωνιακό Δίκτυο(Communication Network):

Το δεύτερο δομικό στοιχείο είναι το δίκτυο κινητών και προσωπικών επικοινωνιών που από τη μία μεταφέρει τα δεδομένα του χρήστη και την αίτηση του για χρήση της location-based υπηρεσίας στον πάροχο και στη συνέχεια επιστρέφει τις ζητούμενες πληροφορίες πίσω στο κινητό τερματικό.

Σύστημα Εύρεσης Θέσης (Positioning System):

Για τη παροχή μια location-based υπηρεσίας πρέπει να καθορίζεται κάθε στιγμή η θέση του κινητού τερματικού. Αυτό γίνεται δυνατό είτε μέσω του δικτύου κινητών τηλεπικοινωνιών είτε χρησιμοποιώντας GPS(Global Positioning System). Ο προσδιορισμός της τρέχουσας τοποθεσίας μπορεί να γίνει και μέσω του ασύρματου δικτύου το οποίο παρέχει πληροφορίες θέσης με τη βοήθεια των σταθμών βάσης(WLAN stations). Η τελευταία μέθοδος μπορεί να χρησιμοποιηθεί αποτελεσματικότερα για την πλοήγηση σε εσωτερικούς χώρους, όπως σε ένα μουσείο. Τέλος, αν η θέση δεν έχει προσδιοριστεί αυτόματα με τους παραπάνω τρόπους, τότε ο χρήστης μπορεί να την ορίσει χειροκίνητα(αν φυσικά τη γνωρίζει).

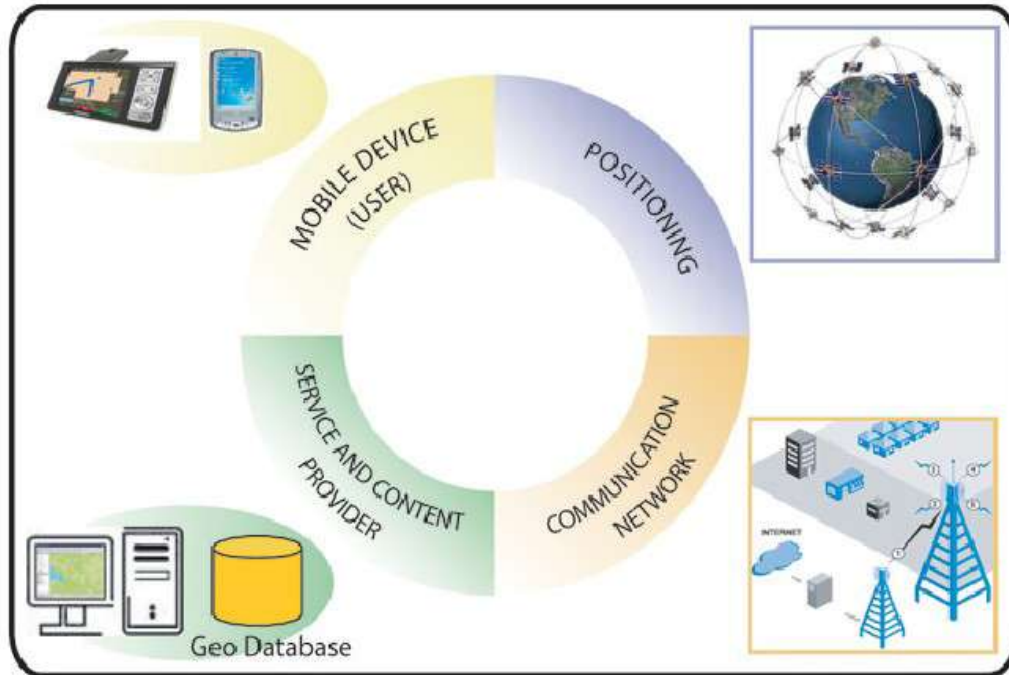
Πάροχοι Υπηρεσιών και Εφαρμογών (Service and Application Provider):

Ο πάροχος υπηρεσιών προσφέρει ένα μεγάλο αριθμό LBS στο χρήστη και είναι υπεύθυνος για την επεξεργασία των αιτήσεων του χρήστη. Τέτοιες υπηρεσίες απαιτούν τον υπολογισμό της θέσης του χρήστη, την εύρεση μιας διαδρομής, την αναζήτηση συγκεκριμένων πληροφοριών για αντικείμενα που άπτονται των ενδιαφερόντων του χρήστη κοκ.

Πάροχοι Περιεχομένου και Δεδομένων (Data and Content Provider):

Οι πάροχοι υπηρεσιών συνήθως δεν αναλαμβάνουν την αποθήκευση και τη «συντήρηση» όλων των πληροφοριών που μπορούν να ζητηθούν από τους χρήστες. Γι' αυτό, τα γεωγραφικά δεδομένα και οι πληροφορίες για τοποθεσίες που αιτούνται μέσω κάποιας LBS, συνήθως παραλαμβάνονται από διαχειριστικές αρχές (όπως εταιρείες χαρτών) ή άλλους εταιρικούς συνεργάτες των παρόχων υπηρεσιών (όπως πχ εταιρείες που ελέγχουν την κίνηση σε δρόμους) .

Τα παραπάνω πέντε βασικά συστατικά των LBS φαίνονται στο παρακάτω σχήμα.

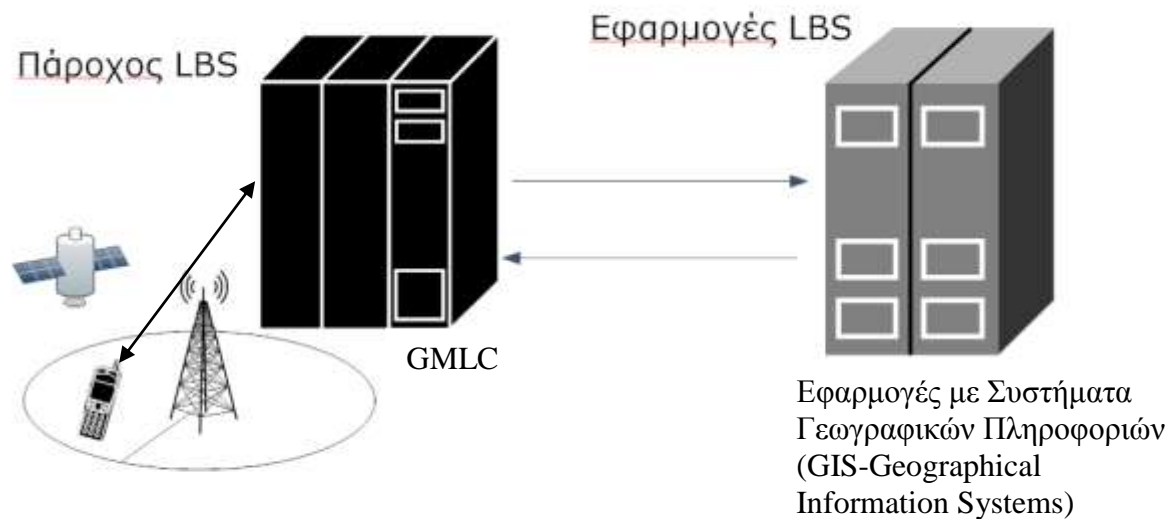


Σχήμα 3. Τα βασικά συστατικά μιας location-based υπηρεσίας: Χρήστης, Δίκτυο Επικοινωνιών, Σύστημα Εύρεσης Θέσης, Πάροχος Υπηρεσιών και Πάροχος Περιεχομένου

1.8 Αρχιτεκτονική Location Based Services και εξέλιξη αυτής

Κατά τη διάρκεια των πρώτων χρόνων σχεδίασης των LBS (1997-2001) , οι πάροχοι στόχευαν στην υλοποίηση ενός κεντρικού δικτυακού κόμβου που θα ανήκε στο δίκτυο σηματοδότησης SS7 και θα είναι ικανός να αποσπά από το δίκτυο πληροφορίες σχετικά με τη γεωγραφική θέση των κινητών τερματικών. Αυτός ο κόμβος, στα δίκτυα κινητών επικοινωνιών GSM ονομάζεται *διαβιβαστικό κέντρο προσδιορισμού θέσης των κινητών τερματικών* (Gateway Mobile Location Center-GMLC). Οι πρώτες εταιρείες που προμηθεύτηκαν τέτοιους κόμβους ήταν οι Ericsson, Nokia, SignalSoft (τώρα Openwave) και η TCS (telecommunication Systems). Οι υπόλοιποι που ακολούθησαν τα βήματά τους ήταν οι παραδοσιακοί κατασκευαστές ασύρματης υποδομής, όπως οι εταιρείες Alcatel, Motorola, Siemens, Nortel, Intrado, Lucent κ.α.

Στην πρώτη αρχιτεκτονική, οι περισσότεροι πάροχοι επέλεξαν την προσέγγιση με την οποία ανοίγεται η διεπαφή προς τον κόμβο GMMLC και απλά επιτρέπεται η πρόσβαση σε όλες τις εφαρμογές που φιλοξενούνται στο IP δίκτυο.



Σχήμα 4. Πρώτη Αρχιτεκτονική LBS (1997-2001)

Σύμφωνα με το παραπάνω μοντέλο, οι πάροχοι θεωρούσαν ότι οι προγραμματιστές των εφαρμογών ήταν υπεύθυνοι για να αποκτήσουν τις γεωγραφικές πληροφορίες που χρειαζόνταν για μια συγκεκριμένη εφαρμογή. Κατά τη διάρκεια αυτής της περιόδου, δεν υπήρχε προτυποποίηση για τη διεπαφή με το GMMLC, ούτε για τις διεπαφές με τους servers GIS. Οι περισσότεροι εξυπηρετητές GIS είχαν τα δικά τους ανοικτά πρότυπα XML αλλά περιείχαν και ειδικά κομμάτια λογισμικού.

Κατά τη διάρκεια της χρήσης της συγκεκριμένης αρχιτεκτονικής, οι συμβατικοί προμηθευτές υπηρεσιών GIS περιορίζονταν στο να προωθούν και να πωλούν χαρτογραφημένες πληροφορίες στους προγραμματιστές των εφαρμογών, οι οποίοι με τη σειρά τους έχτιζαν εφαρμογές και τις προωθούσαν με δικές τους εμπορικές προσφορές στους παρόχους κινητών τηλεπικοινωνιών. Το μη κεντρικό αυτό μοντέλο αρχιτεκτονικής δούλεψε καλά για τις εταιρείες GIS και είχε αρκετά έσοδα για αυτές, αλλά αποδείχθηκε πως δεν ήταν καλή λύση για τους παρόχους που θα ήθελαν να υλοποιούν πολλές LBS εφαρμογές ταυτόχρονα, γρήγορα και χωρίς να αντιμετωπίζουν προβλήματα.

Τα κυριότερα προβλήματα ήταν η ενσωμάτωση των νέων υπηρεσιών με τα υπάρχοντα συστήματα και η διαλειτουργικότητα μεταξύ των συστημάτων. Έτσι, κάθε φορά που ένας πάροχος ήθελε να ενεργοποιήσει μια καινούρια εφαρμογή LBS και να τη διανείμει στους εγγεγραμμένους ή μη χρήστες έπρεπε να την προσαρμόσει έτσι ώστε να μπορέσει αυτή να συνεργαστεί με τα υπάρχοντα συστήματα, πράγμα που απαιτούσε χρόνο και αρκετά έξοδα. Η κάθε εφαρμογή συνήθως χρησιμοποιούσε διαφορετικά GIS και ειδικά τμήματα λογισμικού, οπότε και διαφορετικά APIs(application interfaces). Επομένως, κάθε καινούρια εφαρμογή δεν μπορούσε εύκολα να προσαρμοστεί με τα υπάρχοντα εγκατεστημένα συστήματα του παρόχου. Επιπροσθέτως, υπήρχαν και προβλήματα χρέωσης των υπηρεσιών(billing issues), αφού ήταν δύσκολο για τους παρόχους να εισάγουν νέες εφαρμογές, όταν δεν υπήρχαν κατάλληλοι μηχανισμοί για να κοστολογήσουν μια δραστηριότητα και να τη χρεώσουν ανάλογα.

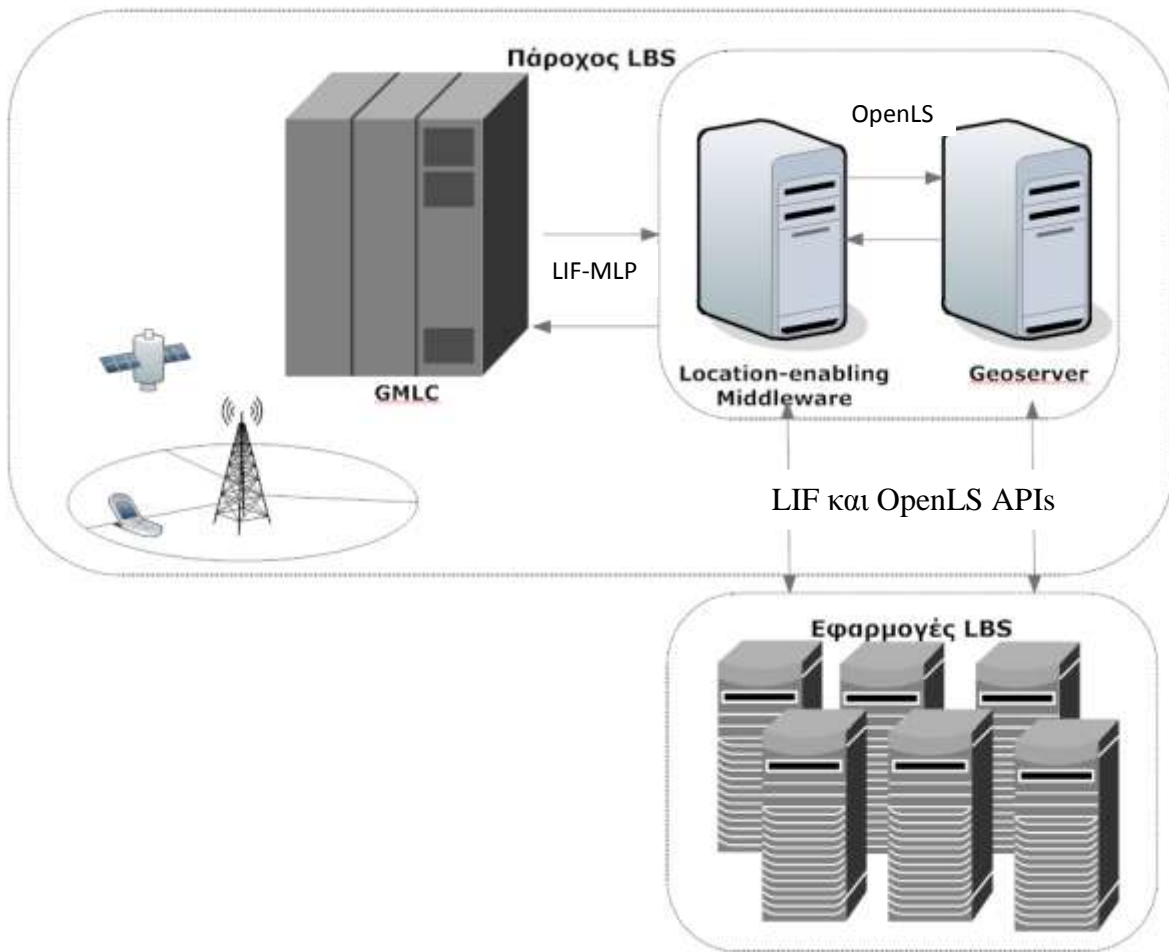
Συμπερασματικά, το πρώτο αυτό μοντέλο αρχιτεκτονικής LBS δεν δούλεψε αποδοτικά, εκτός από περιπτώσεις μη-χρεωστικών, κλειστών εφαρμογών (όπως η υπηρεσία έκτακτης ανάγκης E911) που εξυπηρετούσαν μεμονωμένες λειτουργίες, οπότε και σύντομα εγκαταλείφθηκε. Η βιομηχανία των LBS «αναγκάστηκε» ουσιαστικά να προσδιορίσει ένα καινούριο μοντέλο αρχιτεκτονικής των LBS με σκοπό να διευρύνει τις δυνατότητες των παρόχων για ανάπτυξη περισσότερων LBS εφαρμογών.

Στο δεύτερο μισό του 2000 και στην αρχή του 2001, επήλθε μια σημαντική αλλαγή στον τρόπο με τον οποίο οι πάροχοι επέλεξαν να υλοποιήσουν τα συστήματά τους. Η αλλαγή αυτή ήρθε κυρίως εξαιτίας παλαιότερων αποτυχιών, λαθών και πολλών δοκιμών παρά από κάποια καινοτομία ή κάποια δημιουργική ιδέα. Πλήθος παρόχων που είχαν αντιμετωπίσει τα σοβαρά προβλήματα της πρώτης αρχιτεκτονικής στην υλοποίηση των υπηρεσιών, αναγνώρισαν την αναγκαιότητα για δημιουργία μιας πιο κεντρικής αρχιτεκτονικής. Ήθελαν ένα σύστημα με το οποίο θα μπορούσαν να έχουν τον απόλυτο έλεγχο των LBS και ιδιαίτερα του τρόπου που οι εφαρμογές τους συνδέονταν με το δίκτυο. Το πιο σημαντικό βέβαια ήταν ότι ήθελαν να επιταχύνουν τη διαδικασία εισαγωγής νέων LBS εφαρμογών, τις οποίες θα μπορούσαν να προσφέρουν σε όλους τους συνδρομητές τους.

Η λύση στο πρόβλημα ήταν μια αρχιτεκτονική που βασίζεται στις αρχές των κεντρικών εφαρμογών Web και στο περιβάλλον του one-stop shopping για τους προγραμματιστές. Στο μοντέλο για τις Web εφαρμογές, οι σχεδιαστές εφαρμογών χτίζουν τις εφαρμογές LBS χρησιμοποιώντας μια μεγάλη ποικιλία από κατάλληλες τεχνολογίες και API (application interfaces). Στη συνέχεια, διαθέτουν τις εφαρμογές τους στους παρόχους και αυτοί με τη σειρά τους τις πωλούν στους συνδρομητές τους. Η μετατόπιση αυτή στην προσέγγιση της αρχιτεκτονικής ανάγκασε τη βιομηχανία των LBS να εισάγει δύο νέα λειτουργικά στοιχεία δικτύου στο αρχικό μοντέλο κορμού των LBS, το *Location-enabling Middleware* και τον *Geoserver*, όπως φαίνονται στο σχήμα 5.

Το στοιχείο δικτύου *Location-enabling Middleware* διαχειρίζεται πολλές λειτουργικές και κρίσιμες ερωτήσεις από τα συστήματα των παρόχων και η τεχνολογία του θεωρείται νευραλγική για το σύστημα LBS. Σε αντίθεση με το αρχικό μοντέλο αρχιτεκτονικής που παρουσιάστηκε παραπάνω, στη δεύτερη αυτή αρχιτεκτονική αναπτύχθηκαν διεπαφές λογισμικού προς το GMMLC, με αποτέλεσμα να ελαχιστοποιηθεί η προσπάθεια που απαιτείται για τη διασύνδεση και την ενεργοποίηση νέων εφαρμογών καθώς και ο χρόνος που χρειάζεται ο κάθε πάροχος για μια καινούρια εφαρμογή. Επίσης, το στοιχείο *Location-enabling Middleware* διαχειρίζεται τα προσωπικά και ευαίσθητα δεδομένα που χρησιμοποιούνται από τις εφαρμογές. Το ενδιάμεσο αυτό στοιχείο δικτύου συνδέεται με το σύστημα GIS που στην παρούσα αρχιτεκτονική το ονομάζουμε *Geoserver*.

Γενικά, στο δεύτερο αυτό αρχιτεκτονικό μοντέλο οι πάροχοι των εφαρμογών έχουν τον απόλυτο έλεγχο και έχουν μειώσει κατά πολύ τα κόστη της υλοποίησης των LBS. Επίσης, η ύπαρξη πλέον προτυποποιημένων διεπαφών παίζει σημαντικό ρόλο για το μέλλον των υπηρεσιών LBS, αφού αφενός υπόσχονται εύκολη ανάπτυξη και υλοποίηση νέων υπηρεσιών και αφετέρου θα προσελκύσουν προγραμματιστές οι οποίοι εστιάζουν σε εφαρμογές που αφορούν τη μαζική αγορά, τις επιχειρήσεις κ.α. Τα προτεινόμενα πρότυπα για τις διεπαφές, όπως φαίνονται και στο σχήμα 5, είναι το *Location Interoperability Forum – Mobile Location Protocol (LIF-MLP) API* για τη γεωγραφική θέση και το *OpenLS API* για τη χωρική επεξεργασία. Το LIF-MLP και OpenLS API έλυσαν την πλειονότητα των προβλημάτων για την ολοκλήρωση της διασύνδεσης, τα οποία καθυστέρησαν την ανάπτυξη της αγοράς των LBS.



Σχήμα 5. Δεύτερη αρχιτεκτονική LBS(2002-τώρα)

Συμπερασματικά, λοιπόν, το μέλλον του δεύτερου αυτού αρχιτεκτονικού μοντέλου υπόσχεται να παρέχει ένα one-stop-shop API για τους προμηθευτές των εφαρμογών. Δε θα παίζει κανένα ρόλο για τους παρόχους των εφαρμογών, εάν εστιάζουν στις εφαρμογές μεταξύ επιχειρήσεων και πελατών (Business to Customers-B2C) ή στις εφαρμογές μεταξύ επιχειρήσεων (Business to Business-B2B). Έχοντας πλέον όλα τα τεχνικά θέματα λυμένα, η βιομηχανία των LBS πρέπει να εστιάσει στα επιχειρηματικά μοντέλα και στο πώς θα ελκύσουν τους χρήστες των κινητών επικοινωνιών να χρησιμοποιούν γεωγραφικές πληροφορίες και GIS(Geographical Information Systems) σε καθημερινή βάση.

1.9 Πώς λειτουργούν τα συστήματα εντοπισμού θέσης:

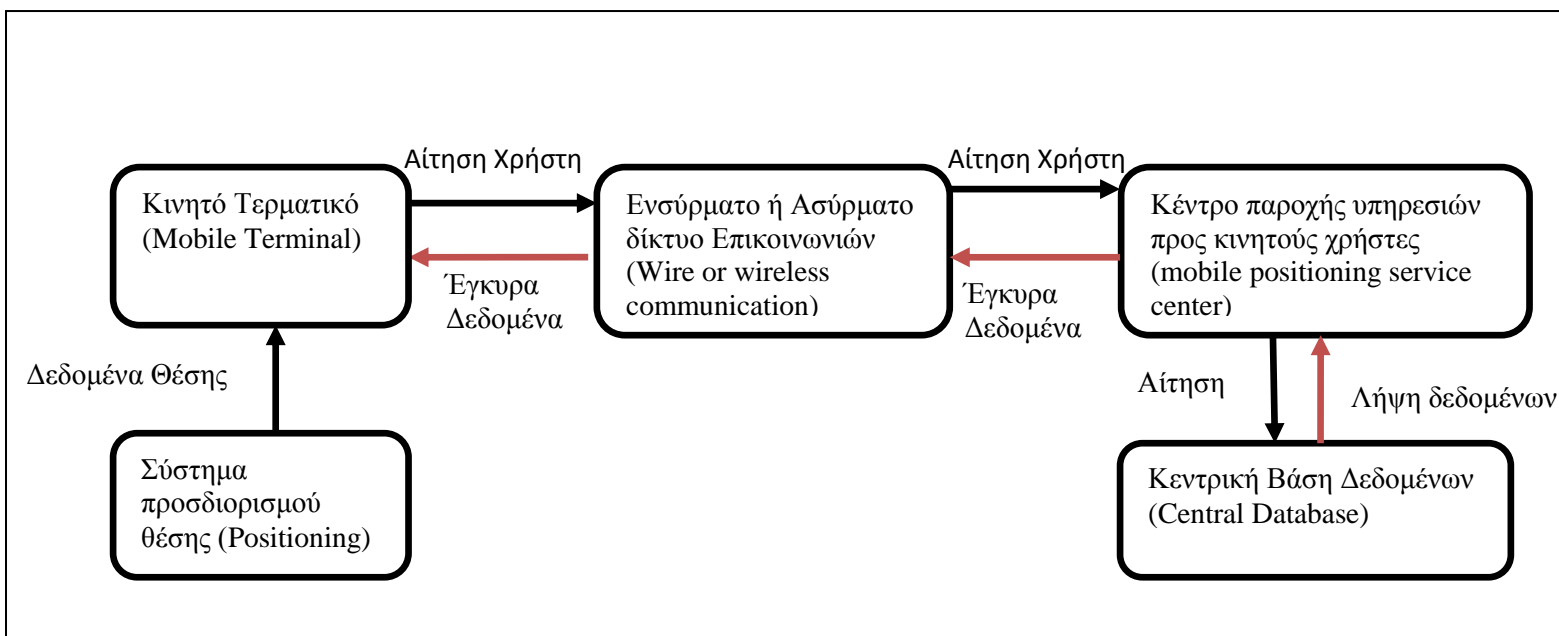
Οι LBS είναι προσωποκεντρικές υπηρεσίες που βασίζονται στην τρέχουσα θέση του χρήστη και επομένως χρησιμοποιούν την τεχνολογία GPS και το δίκτυο κινητών επικοινωνιών για να λάβουν τις πληροφορίες τις σχετικές με την τοποθεσία και να τις μεταδώσουν στην πλευρά του server για περαιτέρω επεξεργασία.

Γενικά, όποτε και οπουδήποτε ένας κινητός χρήστης χρειάζεται πληροφορίες ή κάποια υπηρεσία βασισμένη στην τρέχουσα θέση του ή το σύστημα διαχείρισης LBS χρειάζεται να τον εντοπίσει, τότε το GPS ή το σύστημα εντοπισμού θέσης βασισμένο στο δίκτυο (network positioning system) παρέχει τις πληροφορίες θέσης και τις μεταδίδει στο κέντρο. Οι απαιτήσεις του χρήστη αναλύονται από το κέντρο παροχής υπηρεσιών (service provider) και αφού διαμορφωθεί η απάντηση με τη χρήση της κεντρικής βάσης δεδομένων, οι ζητούμενες πληροφορίες στέλνονται πίσω στο κινητό τερματικό.

Συμπερασματικά, οι υπηρεσίες που βασίζονται στη θέση του χρήστη περιλαμβάνουν δύο κύριες διαδικασίες:

- α) «Συλλογή» της πληροφορίας της θέσης/τοποθεσίας του κινητού τερματικού.
- β) Παροχή της υπηρεσίας σύμφωνα με τις προδιαγραφές που θέτει ο ίδιος ο χρήστης.

Το απλοποιημένο λειτουργικό διάγραμμα των LBS φαίνεται στο παρακάτω σχήμα:



Σχήμα 4. Μπλοκ διάγραμμα λειτουργίας των LBS

Ένα παράδειγμα που δείχνει πως ακριβώς λειτουργεί μια location-based service παρατίθεται παρακάτω. Στο σύστημα αυτό, θεωρούμε πως ένας χρήστης ψάχνει για ένα εστιατόριο και περιγράφουμε βήμα-βήμα την αλυσίδα μετάβασης των πληροφοριών από την αίτηση που κάνει ο χρήστης μέχρι την απάντηση που λαμβάνει. Η πληροφορία που ο χρήστης ουσιαστικά ζητά είναι μια διαδρομή προς το κοντινότερο εστιατόριο. Επομένως, ο χρήστης «εκφράζει» την ανάγκη του, επιλέγοντας στο κινητό τερματικό του την κατάλληλη λειτουργία, πχ menu /search/restaurants

Η ακολουθία των βημάτων που θα λάβουν χώρα είναι η παρακάτω:

1. Εάν η λειτουργία ενεργοποιήθηκε, η πραγματική θέση της κινητής συσκευής λαμβάνεται μέσω του συστήματος προσδιορισμού θέσης (positioning service). Αυτό μπορεί να γίνει, όπως έχει αναφερθεί και παραπάνω είτε από την ίδια τη συσκευή, χρησιμοποιώντας GPS είτε μέσω μιας δικτυακής υπηρεσίας. Στη συνέχεια, ο πελάτης στέλνει τις πληροφορίες που θέλει να λάβει, οι οποίες περιέχουν τον ακριβή στόχο του, δηλαδή το τι θέλει να μάθει και τι ψάχνει πραγματικά και φυσικά στέλνει την τρέχουσα θέση του μέσω του τηλεπικοινωνιακού δικτύου στο GMLC.

2. Το GMLC, όπως ήδη έχει αναφερθεί παραπάνω, ανταλλάσσει μηνύματα μεταξύ του δικτύου κινητών τηλεπικοινωνιών και του server των LBS. Επίσης, γνωρίζει τις διευθύνσεις των servers που είναι υπεύθυνοι για τις LBS εφαρμογές, οπότε και κατευθύνει την αίτηση του χρήστη προς τον κατάλληλο κάθε φορά server. Με τη διασύνδεση του GMLC με το στοιχείο δικτύου Location-enabling middleware και με τον Geoserver, επιτυγχάνεται ακόμη η αποθήκευση πληροφοριών σχετικά με τη συσκευή και το χρήστη που έκανε την αίτηση.

3. Ο server της εφαρμογής που ζητήθηκε από το χρήστη, «διαβάζει» την αίτηση και τις παραμέτρους που έχει ορίσει ο χρήστης και ενεργοποιεί την ανάλογη/αντίστοιχη υπηρεσία-στην περίπτωση του παραδείγματός μας καλεί μια υπηρεσία χωρικής αναζήτησης εστιατορίων.

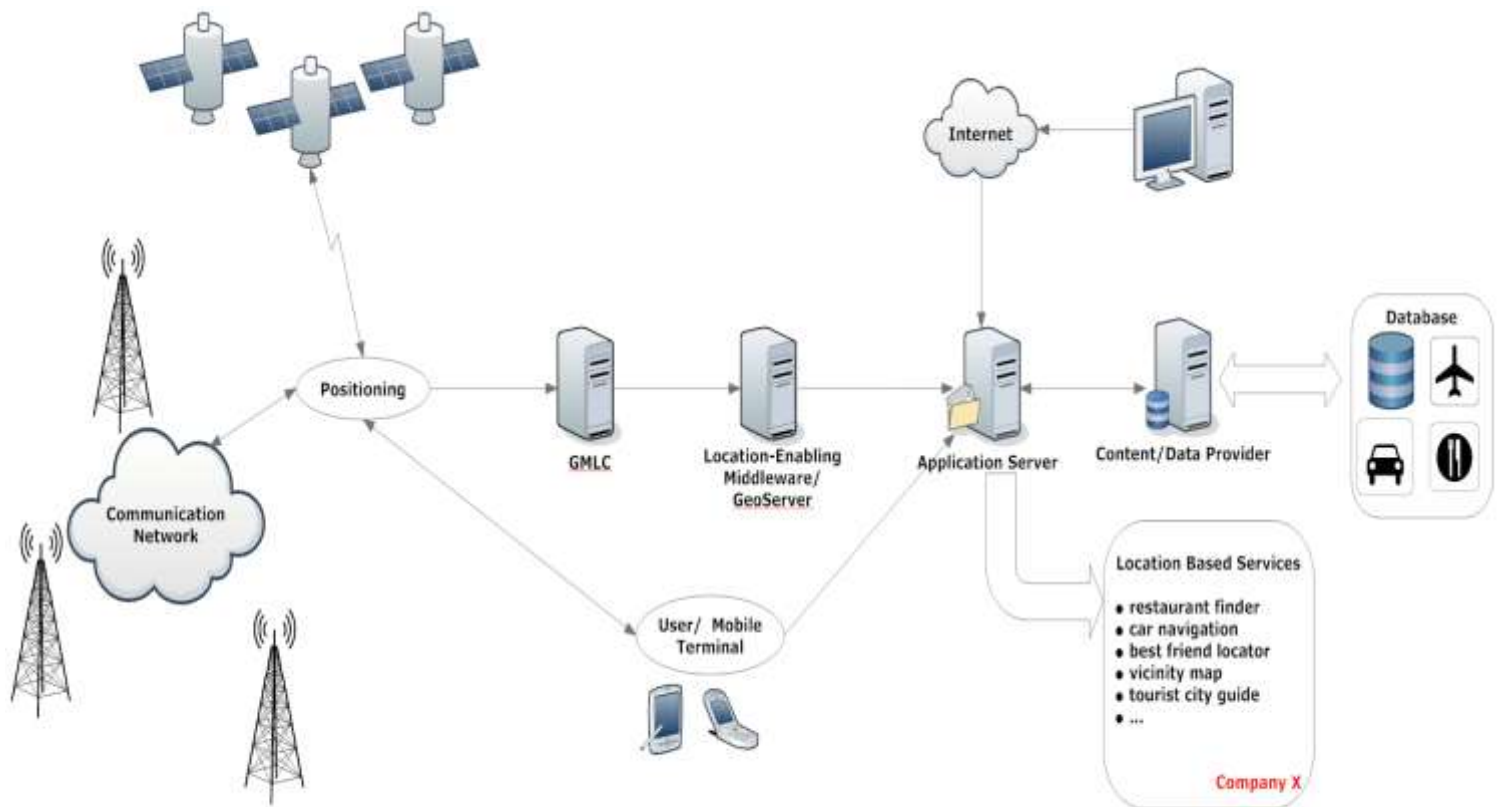
4. Η υπηρεσία αναλύει το μήνυμα και ελέγχει τα κριτήρια αναζήτησης που έχει θέσει ο χρήστης καθώς και τη θέση του, ώστε να διαμορφώσει την απάντηση στην αίτησή του. Στην περίπτωση μας, η υπηρεσία θα ρωτήσει τον πάροχο δεδομένων για τα εστιατόρια που βρίσκονται γύρω από την τρέχουσα θέση του χρήστη.

5. Στη συνέχεια, η υπηρεσία θα βρει πληροφορίες για το αν τα συγκεκριμένα εστιατόρια είναι προσβάσιμα από την θέση του χρήστη και με ποιο μέσο(πόδια, αυτοκίνητο, λεωφορείο), ρωτώντας κάποιον άλλο πάροχο που έχει γνώση του χάρτη της πόλης και των διαφόρων διαδρομών.

6. Τέλος, αφού συλλεχθούν όλες τις παραπάνω πληροφορίες, μια λίστα με τα εστιατόρια που βρίσκονται κοντά στο χρήστη και με τις οδηγίες πρόσβασης προς αυτά αποστέλλεται πίσω στο χρήστη μέσω του Internet, του GMLC και του δικτύου κινητών επικοινωνιών.

7. Τα εστιατόρια θα παρουσιαστούν στο χρήστη είτε με τη μορφή λίστας(ταξινομημένης αλφαβητικά ή ανάλογα με την απόσταση του κάθε εστιατορίου από το χρήστη) είτε θα προβληθούν με pins πάνω σε χάρτη. Στη συνέχεια, ο χρήστης μπορεί να ρωτήσει για περισσότερες πληροφορίες για τα συγκεκριμένα εστιατόρια, όπως πχ τιμές, μενού ή να αξιολογήσει το συγκεκριμένο εστιατόριο αφού το επισκεφτεί ή τέλος να ζητήσει τη διαδρομή προς αυτό.

Στο παρακάτω σχήμα φαίνονται τα διάφορα στάδια που περνά η πληροφορία από τη στιγμή που χρήστης στέλνει την αίτηση του, μέχρι που παίρνει την απάντηση.

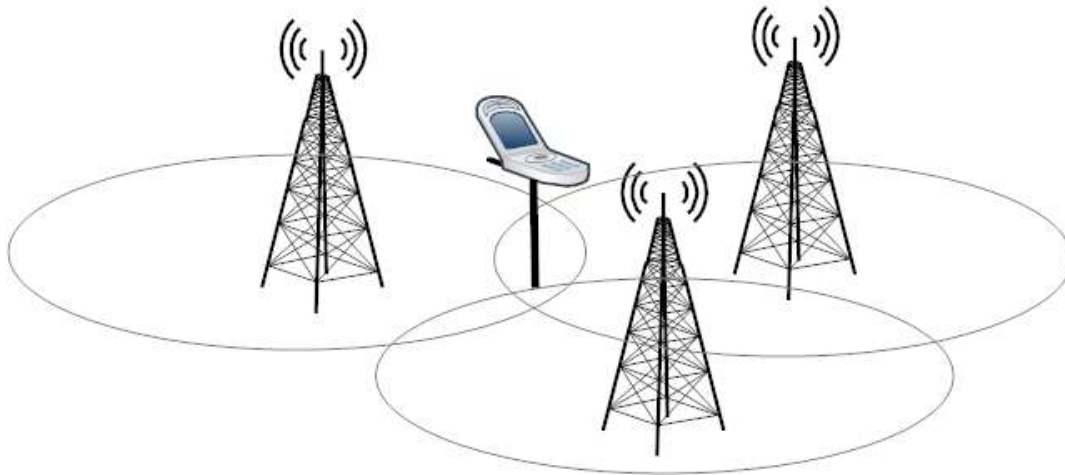


Σχήμα 6. Δομικά στοιχεία των LBS και ροή πληροφοριών

1.10 Τεχνολογίες Προσδιορισμού Θέσης Χρήστη

Βασικό και δομικό τμήμα των LBS, όπως έχει γίνει σαφές και στις παραπάνω ενότητες, είναι ο προσδιορισμός της θέσης των κινητών τερματικών. Οι τεχνολογίες εντοπισμού της θέσης ενός χρήστη έχουν μελετηθεί διεξοδικά τα τελευταία χρόνια καθότι αφενός επιτρέπουν την παροχή Υπηρεσιών Θέσης και αφετέρου είναι εξαιρετικά σημαντικές για την παροχή υπηρεσιών έκτακτης ανάγκης. Μέχρι σήμερα έχουν αναπτυχθεί και προταθεί αρκετές μέθοδοι για την εκτίμηση της θέσης των κινητών τερματικών, οι οποίες αν εξαιρέσουμε την περίπτωση που ο χρήστης χειροκίνητα προσδιορίζει την τρέχουσα θέση του, τότε μπορούμε να τις ταξινομήσουμε σε δύο κατηγορίες: Στην πρώτη κατηγορία βρίσκονται οι μέθοδοι βασιζόμενες στο δίκτυο (network-based positioning), όπου η εκτίμηση της θέσης του χρήστη γίνεται από το δίκτυο των σταθμών βάσης και κάποιο υπεύθυνο κέντρο υπολογισμού θέσης υπολογίζει τη γεωγραφική θέση. Στην κατηγορία αυτή τα κινητά τερματικά στέλνουν κάποιο σήμα στους σταθμούς βάσης ή αναγνωρίζονται αυτόματα από το δίκτυο.

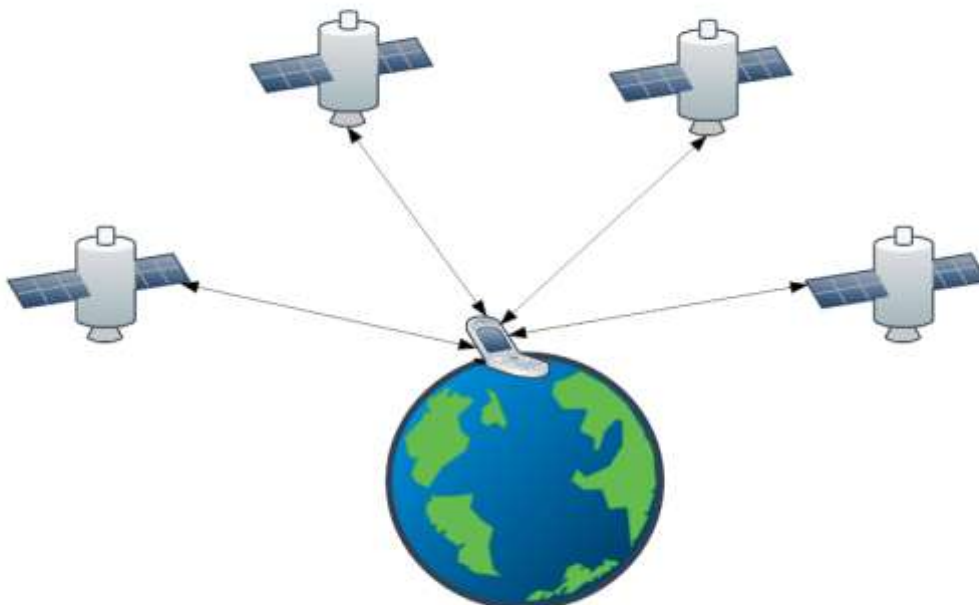
Network Positioning



Σχήμα 7. Προσδιορισμός Θέσης βασιζόμενος στο δίκτυο

Στη δεύτερη κατηγορία βρίσκονται οι μέθοδοι που βασίζονται στο κινητό τερματικό (terminal-based positioning), όπου το ίδιο το κινητό τερματικό λαμβάνει μετρήσεις-σήματα από τους σταθμούς βάσης και υπολογίζει μόνο του τη γεωγραφική του θέση. Το πιο χαρακτηριστικό παράδειγμα μεθόδου που βασίζεται στο κινητό τερματικό είναι η χρήση του GPS(Global Positioning System), όπου οι σταθμοί βάσης είναι οι GPS-δορυφόροι.

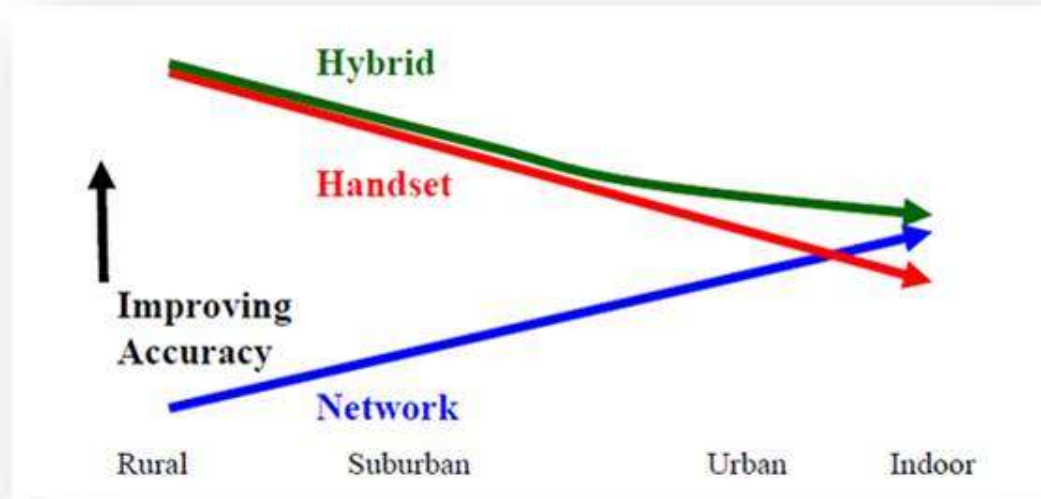
Device/Terminal Positioning



Σχήμα 8. Προσδιορισμός Θέσης βασιζόμενος στο κινητό τερματικό

Τέλος, μια τρίτη κατηγορία μεθόδων προσδιορισμού θέσης είναι οι υβριδικές μέθοδοι-hybrid techniques for positioning που προέρχονται από το συνδυασμό και τη συνεργασία των μεθόδων που βασίζονται στο κινητό τερματικό και σε αυτές που στηρίζονται στο δίκτυο, δηλαδή είναι είτε network-assisted είτε terminal-assisted τεχνικές .

Σαν μια γενική παρατήρηση μπορούμε να αναφέρουμε πως οι network-based τεχνικές για τον προσδιορισμό της θέσης παρουσιάζουν χαμηλότερη ακρίβεια από ότι οι terminal-based τεχνικές, όπου η τρέχουσα θέση βασίζεται στο GPS. Η μεγαλύτερη ακρίβεια εμφανίζεται προφανώς στις υβριδικές μεθόδους, όπως φαίνεται και στο παρακάτω σχήμα.



Σχήμα 9. Ακρίβεια των τριών κατηγοριών προσδιορισμού θέσης

Παρακάτω θα δώσουμε μια σύντομη περιγραφή των μεθόδων που χρησιμοποιούνται για τον εντοπισμό της θέσης του χρήστη.

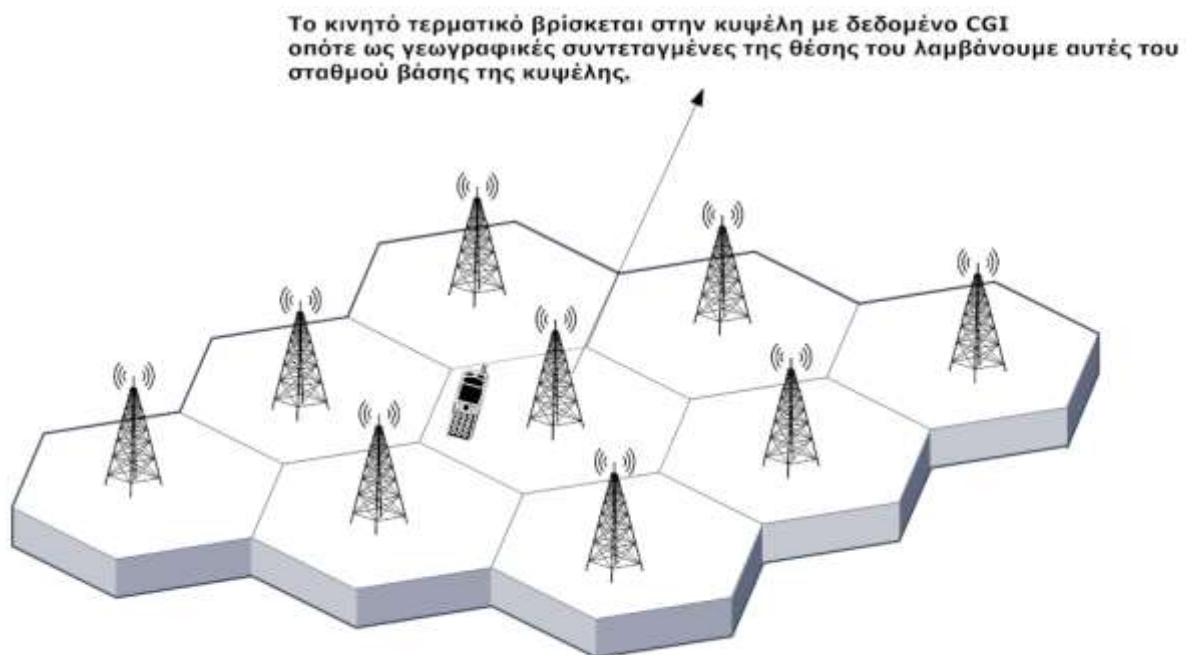
1.11 Μέθοδοι βασιζόμενες στο δίκτυο (Network-based positioning)

Θα αναφερθούμε σε τέσσερις network-based τεχνολογίες εντοπισμού και θα δώσουμε συνοπτικά τα βασικά χαρακτηριστικά λειτουργίας αυτών.

1.11.1 Μέθοδος αναγνώρισης ταυτότητας κυψέλης (Cell-Id)

Η μέθοδος αυτή είναι η πιο απλή και η λιγότερο ακριβή μέθοδος εντοπισμού θέσης και βασίζεται στην αναγνώριση της ταυτότητας της κυψέλης (cell-id) στην οποία βρίσκεται το κινητό τερματικό και η οποία ταυτότητα είναι ήδη διαθέσιμη τόσο στο δίκτυο όσο και στο κινητό τερματικό. Η πληροφορία που χρησιμοποιείται είναι η *παγκόσμια ταυτότητα της κυψέλης (CGI-cell global identity)*, η οποία είναι μοναδική παγκοσμίως και χαρακτηρίζει κάθε κυψέλη. Συνοπτικά, με τη μέθοδο αυτή καθορίζεται η κυψέλη που εξυπηρετεί το κινητό τερματικό και χρησιμοποιούνται οι γεωγραφικές συντεταγμένες του σταθμού βάσης ή του κέντρου της περιοχής κάλυψης της συγκεκριμένης κυψέλης ως μία εκτίμηση της θέσης του κινητού. Οι πληροφορίες για την αντιστοίχιση του CGI με το γεωγραφικό πλάτος και μήκος του σταθμού βάσης βρίσκονται αποθηκευμένες στη βάση δεδομένων του δικτύου. Από τη

μία πλευρά, η εφαρμογή της μεθόδου αυτής απαιτεί ελάχιστες αλλαγές στα υπάρχοντα συστήματα , ενώ παράλληλα ο χρόνος που απαιτείται για την εξαγωγή των αποτελεσμάτων είναι ελάχιστος. Από την άλλη όμως, η ακρίβεια της μεθόδου αυτής εξαρτάται από την ακτίνα της κυψέλης η οποία κυμαίνεται από 50m περίπου για εσωτερικούς χώρους έως 30km για αγροτικές περιοχές , οπότε ένα σφάλμα στο προσδιορισμό της θέσης κατά κάποια km δεν θα ήταν αποδεκτό. Ακόμη, η ακρίβεια της τεχνικής αυτής επηρεάζεται και από τα φαινόμενα διάδοσης και έτσι η κυψέλη που εξυπηρετεί το κινητό μπορεί να μην είναι και η πιο κοντινή, οπότε και πάλι έχουμε λανθασμένο προσδιορισμό της θέσης. Βέβαια, η ακρίβεια μπορεί να βελτιστοποιηθεί με τη χρήση κατευθυντικών κεραιών , με την αύξηση της πυκνότητας των σταθμών βάσης, με την αλλαγή της γεωμετρίας των κυψελών καθώς και με τη μείωση τους μεγέθους τους(microcells, picocells).



Σχήμα 10. Μέθοδος Cell-Id

Η μέθοδος Cell-Id αναβαθμίστηκε στην *εξελιγμένη μέθοδο αναγνώρισης ταυτότητας κυψέλης (Enhanced Cell-Id) ή CGI++*, η οποία χρησιμοποιεί επιπρόσθετα στοιχεία για τον προσδιορισμό της θέσης του κινητού τερματικού. Τέτοια στοιχεία είναι αφενός οι μετρήσεις ισχύος του σήματος, οι οποίες στέλνονται περιοδικά από τα κινητά προς το δίκτυο και αφετέρου οι χάρτες ραδιοκάλυψης που είναι καταχωρημένοι ως ψηφιακές εγγραφές σε βάσεις δεδομένων του συστήματος.

1.11.2 Μέθοδος του χρόνου άφιξης (Time of Arrival-TOA)

Γνωρίζοντας πως τα ηλεκτρομαγνητικά κύματα διαδίδονται με την ταχύτητα του φωτός , στη μέθοδο αυτή μετρούμε το χρόνο άφιξης ενός γνωστού σήματος από το κινητό τερματικό στο σταθμό βάσης ή και αντίστροφα και υπολογίζουμε την απόσταση βάση της σχέσης

$$d_i = c * t_i + c * t_{off,i}$$

Όπου d_i = απόσταση κινητού τερματικού από i -οστό σταθμό βάσης

t_i = χρόνος διάδοσης του σήματος από το κινητό στο σταθμό ή αντίστροφα

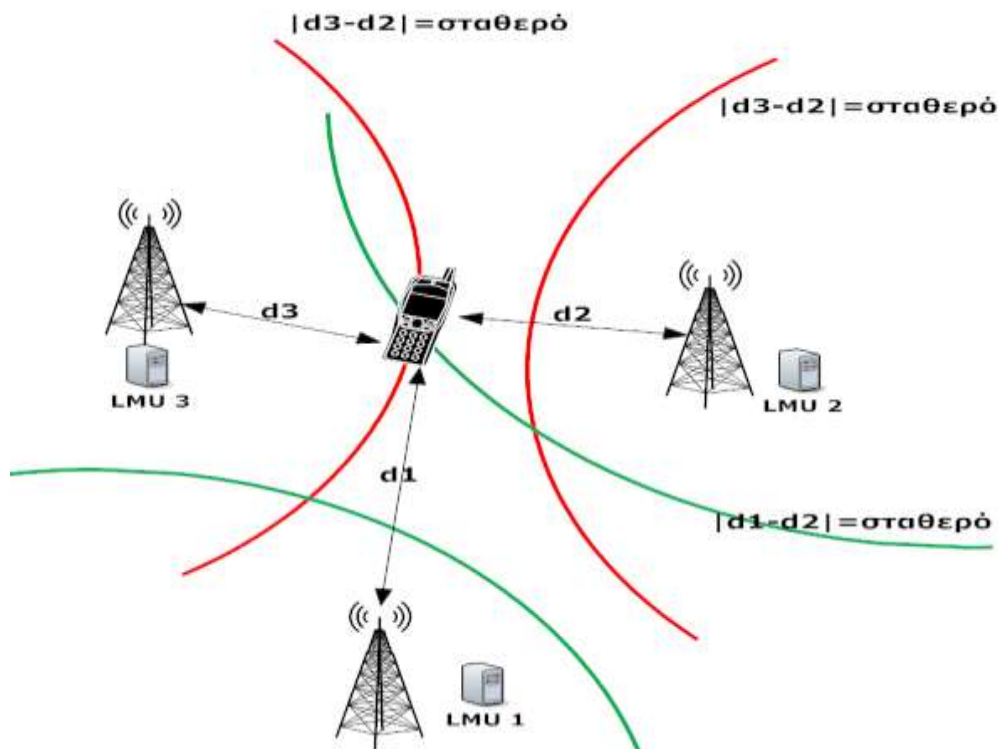
$t_{off,i}$ = διαφορά χρονισμού στα ρολόγια του κινητού και του i -οστού σταθμού βάσης

Η μέτρηση των TOA από τρεις σταθμούς βάσης ορίζει τρεις αποστάσεις, οι οποίες με τη σειρά τους ορίζουν μονοσήμαντα ένα σημείο στο χώρο.

Το κύριο πρόβλημα της μεθόδου αυτής είναι το σφάλμα που υπεισέρχεται λόγω της διαφοράς χρονισμού μεταξύ των ρολογιών σταθμού βάσης και κινητού τερματικού, οπότε και έχει αντικατασταθεί από την επόμενη μέθοδο.

1.11.3 Μέθοδος των διαφορών των χρόνων άφιξης (Time Difference of Arrival -TDOA)

Η μέθοδος αυτή βασίζεται στις διαφορές των χρόνων άφιξης, σε αντίθεση με τη μέθοδο TOA όπου έχουμε απόλυτους χρόνους. Έτσι, καταφέρνουμε να απαλείψουμε τη διαφορά χρονισμού t_{off} που προκαλούσε σφάλματα στην προηγούμενη μέθοδο. Πραγματοποιούμε μετρήσεις του χρόνου άφιξης του σήματος που εκπέμπει το κινητό από τρεις ή περισσότερους σταθμούς βάσης. Το μόνο που απαιτείται στη μέθοδο αυτή είναι η εγκατάσταση σε συγκεκριμένους σταθμούς βάσης κάποιων μονάδων υπολογισμού θέσης του τερματικού, που ονομάζονται LMU (Location Measurement Units). Οι διαφορές των χρόνων άφιξης μεταφράζονται σε διαφορές αποστάσεων, οπότε για κάθε ζευγάρι σταθμών βάσης προκύπτει μια υπερβολή επάνω στην οποία πρέπει να βρίσκεται το κινητό τερματικό και εστίες της υπερβολής είναι οι υπόψη δύο σταθμοί βάσης (σχήμα 11).



Σχήμα 11. Μέθοδος TDOA

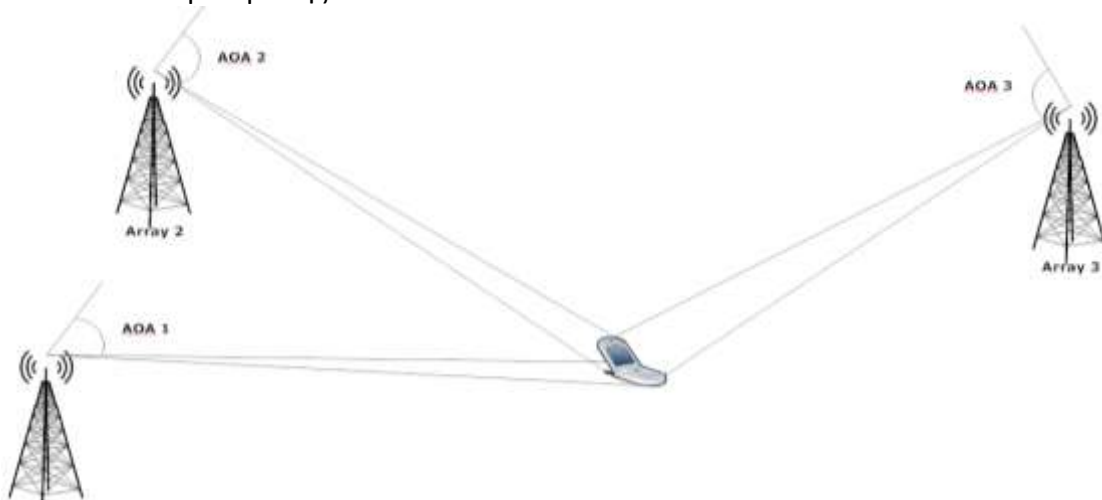
Η τομή δύο υπερβολών προσδιορίζει τη θέση του κινητού τερματικού. Ωστόσο, όπως φαίνεται και στο σχήμα 11, όταν χρησιμοποιούμε τρεις σταθμούς βάσης, τότε οι δύο ανεξάρτητες υπερβολές θα τέμνονται σε περισσότερα από ένα σημεία (συνήθως δύο). Η απόρριψη του ενός σημείου γίνεται είτε με τρίτη υπερβολή, είτε χρησιμοποιώντας κάποια αρχική πληροφορία για τη θέση του κινητού.

Η μέθοδος TDOA παρουσιάζει πολύ καλή ακρίβεια, και ο χρόνος που απαιτείται για την εξαγωγή αποτελεσμάτων είναι αρκετά ανεκτός. Από την άλλη απαιτείται η εισαγωγή νέων μονάδων στο δίκτυο και την προσαρμογή του λογισμικού των κινητών τερματικών. Επίσης, θα πρέπει να λάβουμε υπόψη μας και το πιθανό σφάλμα που εισάγεται λόγω της πολυδιαδρομικής διάδοσης του σήματος από το κινητό τερματικό στους διάφορους σταθμούς βάσης.

Η TDOA εξελίχθηκε στην E-OTD (Enhanced Observed Time Difference) ή εξελιγμένη μέθοδο διαφορών χρόνων άφιξης, όπου εδώ το ίδιο το κινητό τερματικό μετρά τις χρονικές διαφορές μεταξύ σημάτων από διάφορους σταθμούς βάσης και όχι οι σταθμοί βάσης, όπως συμβαίνει στην TDOA. Η E-OTD επιτυγχάνει μεγαλύτερη ακρίβεια και παράλληλα ο χρόνος που απαιτείται για την εξαγωγή των αποτελεσμάτων είναι σε ανεκτά επίπεδα. Παρόλα αυτά όμως, η υλοποίησή της είναι αρκετά ακριβή λόγω της ανάγκης εισαγωγής νέων μονάδων τόσο στο δίκτυο όσο και στο τερματικό.

1.11.4 Μέθοδος γωνίας άφιξης (Angle of Arrival-AOA)

Η μέθοδος αυτή απαιτεί την εγκατάσταση πολύ κατευθυντικών κεραιών στους σταθμούς βάσης και με δυνατότητα στροφής του λοβού ακτινοβολίας ανάλογα με τη θέση του κινητού, ώστε να μπορούν να εντοπίσουν τη γωνία άφιξης του σήματος που εκπέμπεται από το τερματικό. Με λίγα λόγια η μέθοδος αυτή απαιτεί τη χρήση «έξυπνων» κεραιών smart antennas, οι οποίες εντοπίζουν το κινητό τερματικό μέσω της γωνίας ημίσειας ισχύος $\Delta_{3\text{dB}}$ (γωνία που σχηματίζουν οι διευθύνσεις εκατέρωθεν της διεύθυνσης μεγίστου και για τις οποίες η ένταση ακτινοβολίας είναι η μισή της μέγιστης τιμής). Όπως φαίνεται και στο σχήμα 12, οι μετρήσεις για τον προσδιορισμό της θέσης του κινητού πρέπει να γίνουν από δύο τουλάχιστον σταθμούς βάσης, οπότε και η περιοχή που βρίσκεται το κινητό τερματικό περικλείεται μέσα στον κλειστό χώρο που ορίζουν οι γωνίες μισής ισχύος των κύριων λοβών του κάθε σταθμού βάσης.



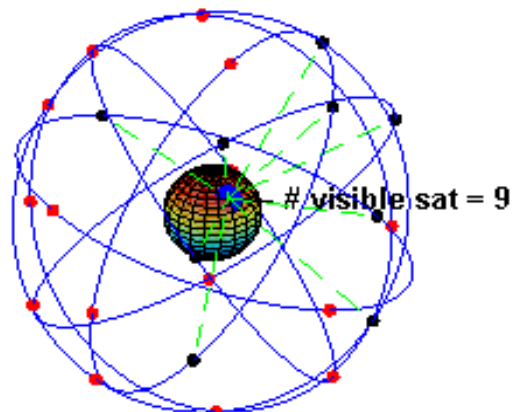
Σχήμα 12. Μέθοδος Angle Of Arrival (AOA)

Η ακρίβεια της μεθόδου στηρίζεται στην απόσταση μεταξύ του κινητού τερματικού και της κεραίας καθώς και στη διαδρομή που ακολουθεί το σήμα καθώς διαδίδεται. Η μέθοδος AOA απαιτεί συνθήκες LOS (Line-of-Sight) διάδοσης για να δώσει σωστά αποτελέσματα, ενώ η λειτουργία της δεν είναι εγγυημένη σε NLOS περιπτώσεις. Σε ένα πυκνό πολυδιαδρομικό περιβάλλον, η θέση του κινητού τερματικού προσδιορίστηκε με ακρίβεια 300m όταν δύο σταθμοί βάσης ήταν ορατοί και με ακρίβεια 200m όταν η μέτρηση έγινε από τρεις σταθμούς. Στις αγροτικές και ημιαστικές περιοχές η ακρίβεια είναι ακόμα υψηλότερη και έχει αποδειχθεί πως η χρήση δύο κύριων σταθμών βάσης είναι αρκετή. Βέβαια, το κύριο εμπόδιο της μεθόδου AOA είναι η υλοποίηση και η εγκατάσταση των διατάξεων με τις έξυπνες κεραίες, μια διαδικασία που είναι πολύπλοκη, υψηλού κόστους και η οποία δεν περιλαμβάνεται στα κοινά δίκτυα κινητών και προσωπικών επικοινωνιών.

1.12 Μέθοδοι βασιζόμενες στο κινητό τερματικό (terminal-based positioning)

1.12.1 Δορυφορικός Προσδιορισμός Θέσης (GPS)

Το σύστημα GPS-Global Positioning System σχεδιάστηκε αρχικά για το υπουργείο Άμυνας των ΗΠΑ, αλλά τα τελευταία χρόνια χρησιμοποιείται ευρέως σε όλο τον κόσμο. Το δίκτυο GPS αποτελείται από 24 δορυφόρους που κινούνται σε έξι 12ωρες ελλειπτικές τροχιές (4 δορυφόροι ανά τροχιά) σε απόσταση 20,2 km από την επιφάνεια της γης. Το σύστημα είναι έτσι σχεδιασμένο ώστε όλοι οι δορυφόροι να βρίσκονται σε κλίση 55° ως προς το επίπεδο του ισημερινού, σε 60° μεταξύ τους ο ένας από τον άλλο στο επίπεδο της κάθε τροχιάς και σε κάθε χρονική στιγμή από οποιοδήποτε σημείο της γης να είναι ορατοί τουλάχιστον έξι από αυτούς. Οι δορυφόροι αυτοί «σκεπάζουν» ομοιόμορφα με το σήμα τους ολόκληρο τον πλανήτη, γεγονός που αποδεικνύει τη φιλοσοφία που κρύβεται πίσω από τη λειτουργία του συστήματος GPS, δηλαδή τη διαθεσιμότητά του σε κάθε σημείο της Γης, ώστε να μην υπάρχει κίνδυνος να αποπροσανατολιστεί κανείς ποτέ και πουθενά.



Σχήμα 13. Αναπαράσταση του συστήματος GPS, με 24 δορυφόρους GPS (4 δορυφόροι σε καθεμία από τις 6 τροχιές). Σε μια τυχαία χρονική στιγμή φαίνεται ο αριθμός των ορατών δορυφόρων (9) από ένα σημείο (45° Βόρεια) που δείχνει το βέλος.

Εκτός από το διαστημικό του τμήμα, το σύστημα GPS απαρτίζεται και από επίγειο τμήμα ελέγχου με ένα επανδρωμένο τμήμα επίγειου ελέγχου και τέσσερα μη επανδρωμένα κέντρα σε ισάριθμες περιοχές του πλανήτη. Τα κέντρα αυτά πραγματοποιούν ελέγχους όσον αφορά την ταχύτητα, το υψόμετρο και την επάρκεια των δορυφόρων σε ηλεκτρική ενέργεια. Παράλληλα, γίνονται και οι όποιες διορθωτικές ενέργειες απαιτούνται και αφορούν το σύστημα χρονομέτρησης των δορυφόρων ώστε να αποτρέπεται η παροχή λανθασμένων πληροφοριών στους χρήστες του συστήματος.

Ο κυριότερος σταθμός βάσης είναι αυτός στο Colorado Springs των ΗΠΑ, ο οποίος είναι μάλιστα και ο μοναδικός που βρίσκεται στην ξηρά. Αναλαμβάνει τον έλεγχο της σωστής λειτουργίας των εναπομεινάντων τεσσάρων σταθμών, καθώς και τον συντονισμό τους. Σημειώνοντας τη θέση των σταθμών αυτών πάνω σε έναν παγκόσμιο χάρτη, παρατηρεί κανείς ότι η διάταξή τους δεν είναι τυχαία, αλλά ακολουθούν μια γραμμή παράλληλη με τα γεωγραφικά μήκη της Γης.

Η τεχνολογία του GPS απαιτεί για τον εντοπισμό της θέσης του κινητού τερματικού την ενσωμάτωση ενός δέκτη GPS στο κινητό του χρήστη. Ο δέκτης αυτός μπορεί να υπολογίσει τη θέση του χρήστη βασιζόμενος σε μετρήσεις του χρόνου διάδοσης των σημάτων από τους δορυφόρους. Γενικά, οι δορυφόροι μεταδίδουν συνεχώς σήματα που περιλαμβάνουν τον ακριβή χρόνο μετάδοσής τους και πληροφορίες της τροχιάς του δορυφόρου. Ο δέκτης με τη σειρά του μετρά το χρόνο άφιξης του κάθε μηνύματος (TOA-time of arrival) και υπολογίζει την απόσταση από κάθε δορυφόρο. Έχει αποδειχθεί πως 4 ή περισσότεροι δορυφόροι που βρίσκονται σε οπτική επαφή με το κινητό τερματικό επαρκούν για να προσδιορίσουν τις γεωγραφικές συντεταγμένες του και μάλιστα με πάρα πολύ ακρίβεια, της τάξης των 3-50m για το 95% του χρόνου.

Παρόλα αυτά όμως, το σύστημα GPS για να λειτουργεί σωστά πρέπει να έχει καθαρή επαφή με το κινητό τερματικό, οπότε παρουσιάζονται προβλήματα σε περιπτώσεις εντοπισμού τερματικών σε εσωτερικούς χώρους. Εκεί, το σήμα που φτάνει από τους δορυφόρους είναι αρκετά ασθενές λόγω των κτιρίων που παρεμβάλλονται με αποτέλεσμα η πληροφορία του μηνύματος πλοήγησης να μη λαμβάνεται σωστά ή να χάνεται εντελώς, πράγμα που ουσιαστικά «παραλύει» τη λειτουργικότητα του συστήματος GPS.

Η λύση στο πρόβλημα της κάλυψης εσωτερικών χώρων βρίσκεται στην ανάπτυξη του A-GPS (Assisted GPS), όπου η πληροφορία που μεταδίδεται από το δορυφόρο περνά μέσα από το δίκτυο. Η μέθοδος εντοπισμού A-GPS θα αναλυθεί στην επόμενη παράγραφο.

1.13 Υβριδικές μέθοδοι προσδιορισμού της θέσης (hybrid technologies)

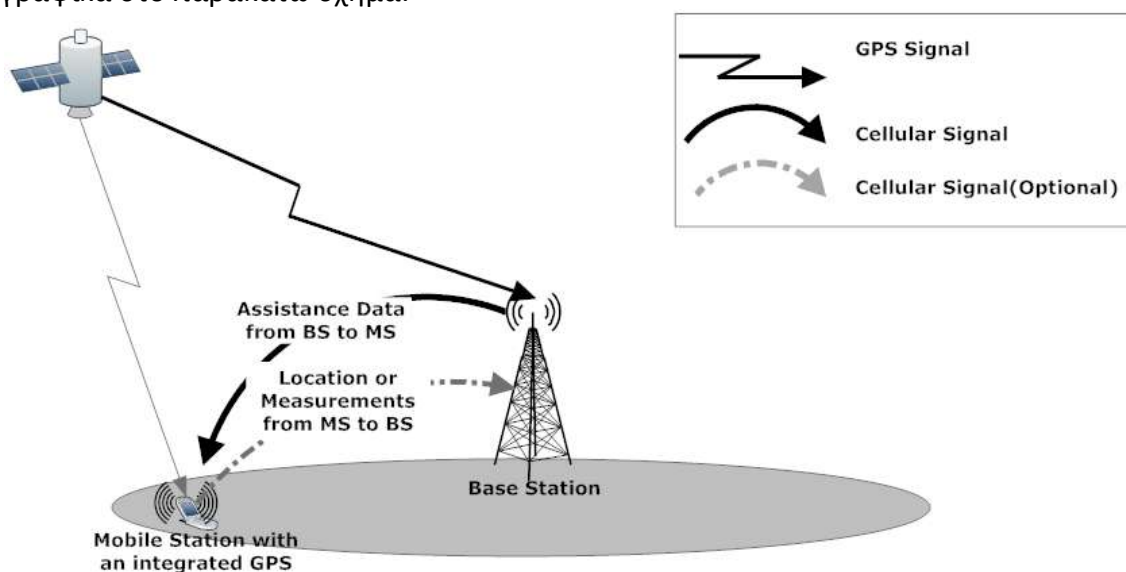
1.13.1 Υποβοηθούμενο GPS- Assisted GPS(A-GPS)

Το υποβοηθούμενο GPS (Assisted GPS, A-GPS) είναι μία μέθοδος που βελτιώνει κατά πολύ την επίδοση του GPS και επιφέρει σημαντική αύξηση της διαθεσιμότητας της πληροφορίας θέσης στους κινητούς χρήστες. Η τεχνική αυτή βασίζεται στο γεγονός πως για τον υπολογισμό της θέσης του κινητού τερματικού συμβάλλει και το κινητό στο οποίο είναι ενσωματωμένος ο δέκτης GPS. Ο τελικός υπολογισμός της θέσης γίνεται είτε στο κινητό τερματικό είτε στο MLC (Mobile Location Center) του δικτύου. Η πιο συνήθης υλοποίηση της μεθόδου περιλαμβάνει την εγκατάσταση ηλεκτρονικών διατάξεων GPS στο δέκτη, με ταυτόχρονη εγκατάσταση κατάλληλης υποδομής στο δίκτυο.

Όπως αναφέρθηκε στην προηγούμενη παράγραφο, ο προσδιορισμός της θέσης μέσω GPS γίνεται δύσκολος σε περίπτωση που χαθεί η οπτική επαφή του δορυφόρου με το κινητό τερματικό. Επίσης, ο χρόνος που χρειάζεται για την πρώτη σύνδεση- στην περίπτωση του GPS- μπορεί να είναι σχετικά μεγάλος(min-30sec) ειδικά σε εφαρμογές έκτακτης ανάγκης(emergency applications). Έτσι, για την επίλυση των παραπάνω προβλημάτων, καθώς επίσης για τη μείωση της ισχύος που καταναλώνει η συσκευή όταν χρησιμοποιεί το GPS, για την αύξηση της ευαισθησίας του GPS δέκτη και για την αύξηση της ακρίβειας στον προσδιορισμό της θέσης, αναπτύχθηκε το A-GPS.

Η βασική ιδέα του A-GPS είναι πως για τον υπολογισμό της θέσης του κινητού τερματικού απαιτείται να δημιουργηθεί ένα δίκτυο αναφοράς GPS συνδεδεμένο με το δίκτυο κινητών επικοινωνιών, του οποίου οι δέκτες να έχουν καθαρή οπτική επαφή(LOS-Line of Sight) με τους δορυφόρους και να μπορούν να λειτουργούν συνεχώς. Οι στατικοί αυτοί δέκτες εγκαθίστανται ομοιόμορφα στην περιοχή γεωγραφικής κάλυψης του δικτύου επικοινωνιών και συλλέγουν δεδομένα από τους δορυφόρους που αφορούν την πλοήγηση, την ακριβή ώρα, Doppler συχνότητα κ.α. Όταν το κινητό τερματικό, λοιπόν, ζητά πληροφορία θέσης, τότε οι δέκτες αυτοί στέλνουν βοηθητικά δεδομένα και βοηθούν το κινητό τερματικό να «κλειδώσει» γρηγορότερα τους δορυφόρους που απαιτούνται για να προσδιορίσει τη θέση του, με αποτέλεσμα τη σημαντική μείωση του χρόνου απόκρισης(<30 sec). Επίσης, καθώς οι διάφορες λειτουργίες λαμβάνουν χώρα σε server-MLC εκτός του κινητού τερματικού, ο οποίος έχει πάντα επαφή με τους δορυφόρους και μεγάλη υπολογιστική ισχύ, είναι επόμενο να μειώνεται σημαντικά η ελάχιστη στάθμη ισχύος λήψης των κινητών τερματικών, οπότε μπορούν να εντοπιστούν και σε συνθήκες NLOS.

Η τεχνική αυτή που υποβοηθείται από το κινητό τερματικό(mobile-assisted) ουσιαστικά μετατοπίζει τις λειτουργίες του παραδοσιακού δέκτη GPS προς το δίκτυο. Το μόνο που απαιτείται εκτός του ενσωματωμένου στο κινητό τερματικό δέκτη είναι κάποιες αλλαγές στο δίκτυο, όπως εισαγωγή κεραίας, συστήματος RF και ψηφιακού επεξεργαστή στους επίγειους δέκτες. Το δίκτυο εκπέμπει το βοηθητικό μήνυμα που συνθέτει από τα σήματα πληροφορίας των δορυφόρων(satellite messages) και βοηθά το δέκτη GPS των κινητών τερματικών να μειώσει αισθητά το χρόνο ανάκτησης της πληροφορίας θέσης. Η διαδικασία αυτή φαίνεται γραφικά στο παρακάτω σχήμα.



Σχήμα 14. Σύστημα Εντοπισμού Θέσης A-GPS

Εκτός από το mobile-assisted AGPS , η λύση που βασίζεται στο κινητό τερματικό (mobile-based) περιλαμβάνει έναν πλήρως λειτουργικό δέκτη GPS ενσωματωμένο στο κινητό τερματικό. Στην τεχνική αυτή η τελική θέση του κινητού υπολογίζεται από το ίδιο το τερματικό. Το μόνο που απαιτείται είναι να παρασχεθεί ένας σημαντικός αριθμός δεδομένων στο τερματικό σε σχέση με τη mobile-assisted λύση, όπως χρόνος, θέση αναφοράς, πληροφορία για συγχρονισμό των ρολογιών κ.α.

Η τεχνική του A-GPS παρέχει ακόμα μεγαλύτερη ακρίβεια αν συνδυαστεί με τεχνικές διαφορικού GPS (DGPS), όπου χρησιμοποιείται ένας σταθερός δέκτης αναφοράς που μπορεί και διορθώνει σφάλματα πόλωσης και κοινού ρυθμού σε κινητούς δέκτες. Γενικά, σε εξωτερικό περιβάλλον και αγροτικές περιοχές η εκτιμώμενη απόδοση της μεθόδου είναι 10m , σε ημιαστικές περιοχές 10-20m , σε αστικές περιοχές όπου η πολυδιαδρομική διάδοση σήματος είναι πιο έντονη (multipath propagation) 10-100m, ενώ τέλος σε εσωτερικούς χώρους όπου το σύννεφες GPS έχει ασθενή κάλυψη 30m για το 67% των μετρήσεων που έγιναν.

Η αξιοπιστία του AGPS σαν μέθοδος προσδιορισμού θέσης είναι αρκετά υψηλή καθώς παρέχει πολύ μεγαλύτερη διαθεσιμότητα της πληροφορίας θέσης κινητών τερματικών σε σχέση με την απλή τεχνική του GPS. Επίσης, σημαντικός παράγοντας στην εφαρμογή της μεθόδου αυτής παίζει το γεγονός ότι οι ενσωματωμένοι δέκτες GPS στο κινητό γίνονται ολοένα και πιο μικροί και προσαρμόζονται πιο εύκολα και πιο αποδοτικά με αποτέλεσμα ο δέκτης GPS να λειτουργεί με χαμηλή κατανάλωση ισχύος.

1.14 Άλλες τεχνολογίες εντοπισμού θέσης

Μια κατηγορία τεχνολογιών θέσης που χρησιμοποιείται για την παροχή κυρίως κοντινών σε απόσταση LBS είναι το Bluetooth, WLAN και οι υπέρυθρες. Οι τεχνολογίες αυτές χρησιμοποιούνται για να συνδεθούν τα κινητά τερματικά σε τοπικές υπηρεσίες εξυπηρετητές και αυτοί με τη σειρά τους να παρέχουν στο χρήστη την τοποθεσία στην οποία βρίσκεται και ενδεχομένως και άλλες πληροφορίες και δεδομένα σχετικά με τη γύρω περιοχή.

Οι τεχνικές αυτές είναι ιδιαίτερα χρήσιμες για το προσδιορισμό θέσης σε εσωτερικούς χώρους. Στην περίπτωση αυτή, τεχνικές όπως το E-OTD μπορούν και παρέχουν συνήθως μόνο πληροφορία για το κτίριο ή το συγκρότημα κτιρίων που βρίσκεται το κινητό τερματικό. Για την παροχή πιο λεπτομερών και μεγαλύτερης ακρίβειας πληροφοριών απαιτείται ένα δίκτυο εσωτερικών (εντός του κτιρίου) σταθμών βάσης. Ακόμα όμως και σε αυτή την περίπτωση, μέθοδοι που βασίζονται στη μέτρηση των χρονικών καθυστερήσεων του σήματος υποφέρουν από περιορισμένη ακρίβεια που εξαρτάται σε μεγάλο βαθμό από τις διαστάσεις του κτιρίου. Έτσι, τεχνικές που βασίζονται στη μέτρηση του επίπεδου του ληφθέντος σήματος είναι πιο αποτελεσματικές, ειδικά στην περίπτωση των εσωτερικών χώρων.

Στη λογική αυτή, αυξάνοντας την πυκνότητα των πηγών σήματος που βρίσκονται μέσα στο κτίριο ενισχύουμε την ακρίβεια προσδιορισμού της θέσης του κινητού τερματικού. Τα WLANs-Wireless Local Area Networks συνήθως αποτελούνται από ένα πυκνό δίκτυο από Access Points (APs), οπότε προσφέρουν πολύ καλές πιθανότητες για προσδιορισμό θέσης βάση της έντασης του σήματος.

Μια εξίσου αποτελεσματική τεχνική εντοπισμού θέσης σε εσωτερικούς χώρους είναι η χρήση δικτύου με πομπούς Bluetooth. Η τεχνολογία Bluetooth βρίσκεται σε όλες τις κινητές συσκευές, δεν απαιτεί μεγάλη κατανάλωση μπαταρίας και υλοποιεί ταυτόχρονα πολλές υπηρεσίες δικτύου. Κάθε συσκευή Bluetooth έχει ένα μοναδικό αναγνωριστικό και μπορεί να ανιχνευθεί εύκολα από οποιαδήποτε άλλη συσκευή. Αν και αρχικά το Bluetooth δεν αναπτύχθηκε για εύρεση θέσης και LBS, αλλά για επικοινωνία χρηστών σε προσωπικά ανοικτά δικτυακά περιβάλλοντα, εντούτοις υπάρχουν δύο προσεγγίσεις για την εύρεση θέσης με τη συγκεκριμένη τεχνολογία:

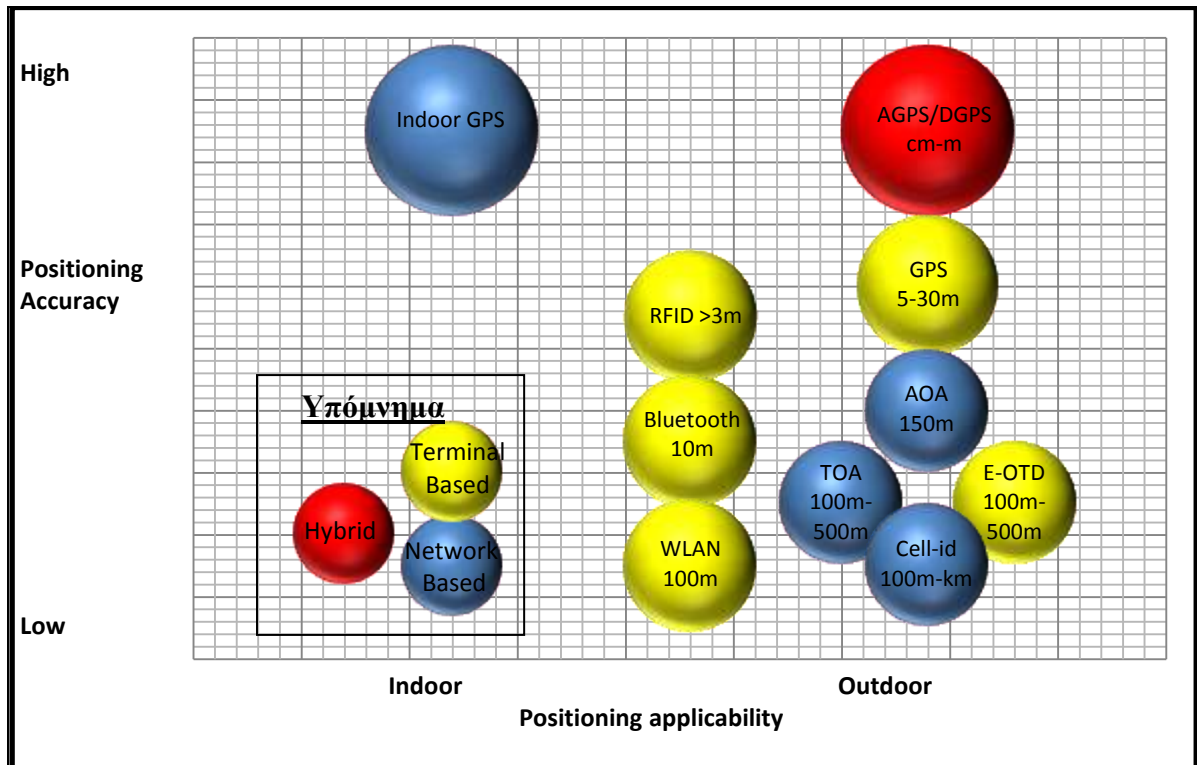
α) Binary Location: Η προσέγγιση αυτή απαιτεί την εγκατάσταση ενός AP-Access Point σε κάθε δωμάτιο του κτιρίου, στο οποίο το κινητό θα συνδέεται και έτσι θα μπορεί να προσδιορίσει τον χώρο στον οποίο βρίσκεται.

β) Analog Location: Στην προσέγγιση αυτή τα APs είναι εγκατεστημένα σε πιο αραιά σημεία στο χώρο του κτιρίου, η απόσταση από το κάθε AP μετράται και το σύστημα με βάση τη μέτρηση τριών τέτοιων αποστάσεων και τη διαδικασία του triangulation βρίσκει κατά προσέγγιση που βρίσκεται ο χρήστης.

1.15 Συγκεντρωτική παρουσίαση των μεθόδων προσδιορισμού θέσης

Όπως είδαμε παραπάνω, είναι πολύ σημαντικό για τη σωστή λειτουργία και την εφαρμογή των Location Based Services η επιλογή της καταλληλότερης τεχνολογίας εντοπισμού θέσης του κινητού τερματικού, εκείνης δηλαδή που θα δίνει σε κάθε περίπτωση την καλύτερη ακρίβεια. Κάθε τεχνολογία έχει πλεονεκτήματα και μειονεκτήματα και ο μόνος τρόπος να διακρίνουμε την καταλληλότερη είναι βάση συγκεκριμένων κριτηρίων, όπως η επίδραση στο δίκτυο και στο κινητό τερματικό, η ακρίβεια της πληροφορίας θέσης, ο χρόνος υπολογισμού, η πολυπλοκότητα κ.α.

Στο παρακάτω γράφημα, φαίνονται οι κυριότερες μέθοδοι εντοπισμού θέσης, η ακρίβεια που παρέχουν και η καταλληλότερη εφαρμογή τους σε εσωτερικούς και εξωτερικούς χώρους.



Σχήμα 15. Μέθοδοι προσδιορισμού θέσης, ακρίβεια και εφαρμογή

Από το σχήμα γίνεται σαφές ότι οι μέθοδοι που βασίζονται στο δίκτυο μπορούν να χρησιμοποιηθούν για LBSs σε περιπτώσεις που η ακρίβεια του προσδιορισμού δεν είναι κρίσιμος παράγοντας. Από την άλλη οι τεχνικές που βασίζονται στο τερματικό συνιστώνται για εφαρμογές όπου η ακρίβεια είναι σημαντική. Τέλος, η υβριδική τεχνολογία του AGPS φαίνεται πως υπερέχει καθώς συνδυάζει τα πλεονεκτήματα και καλύπτει τις αδυναμίες των δύο άλλων κατηγοριών.

Στον παρακάτω πίνακα παρουσιάζονται επιγραμματικά οι προαναφερθείσες τεχνικές προσδιορισμού θέσης με τα κρίσιμα χαρακτηριστικά τους.

Μέθοδος	Πολυπλοκότητα	Ακρίβεια	Παράγοντες που επηρεάζουν την ακρίβεια	Αλλαγές στο δίκτυο	Αλλαγές στο τερματικό
Cell-Id	Ελάχιστη –απαιτείται μόνο ένας σταθμός βάσης	500m-20km	Μέγεθος Κυψέλης-multipath propagation	Καμία- Η πληροφορία είναι ήδη διαθέσιμη	Καμία
TOA	Μικρή	200m-1km	1)Multipath propagation 2)Άγνωστο toff 3)Συνθήκες NLOS	Μέτρια-Επιπλέον μονάδες συγχρονισμού	Καμία
TDOA	Μικρή/Μέτρια	40m-500m	1)Multipath 2)Συνθήκες NLOS	Μέτρια-Επιπλέον μονάδες συγχρονισμού	Καμία
AOA	Μέτρια	50m-150m	1)Συνθήκες NLOS 2)Σφάλματα πόλωσης	Μεγάλη-Απαιτούνται πολύ ακριβές διατάξεις στοιχειοκεραιών	Καμία
GPS	Μικρή	5m-100m	1)Συνθήκες NLOS 2)Σφάλματα πόλωσης	Καμία	Πολύ μεγάλη-Ενσωμάτωση δέκτη GPS
A-GPS	Μικρή/Μέτρια	3m-50m	Ελάχιστοι	Μικρή-Μετάδοση βοηθητικών δεδομένων	Πολύ μεγάλη-Ενσωμάτωση δέκτη GPS

1.16 Υπηρεσίες προσδιορισμού θέσης(Location based Services) και Ιδιωτική Ζωή των Mobile χρηστών

Τα τελευταία χρόνια η αγορά των LBS διευρύνεται με εντυπωσιακό τρόπο και γίνονται ολοένα και περισσότερες προσπάθειες για καλύτερο σχεδιασμό της δομής τους όσον αφορά το περιβάλλον χρήστη, την πιο εύκολη ενσωμάτωσή τους στις κινητές συσκευές και τη βελτίωση των δικτυακών υποδομών που θα μπορούν να τις υποστηρίξουν. Η ολοένα και αυξανόμενη χρήση τους, που βασίζεται κυρίως στην εντυπωσιακή εξέλιξη των τεχνολογιών εντοπισμού θέσης-όπως αυτές περιγράφηκαν παραπάνω-έχει μεγάλη επιρροή τόσο σε κοινωνικό όσο και σε επαγγελματικό επίπεδο. Από τη μια πλευρά, ο εντοπισμός της θέσης ενός ατόμου μπορεί να είναι χρήσιμος σε περιπτώσεις παρακολούθησης υπόπτων από την αστυνομία ή η παρακολούθηση των οχημάτων μεταφοράς και η διαχείριση στόλου να είναι αρκετά συχνά ζωτικής σημασίας για τις επιχειρήσεις και τη σωστή λειτουργία τους(fleet management) . Από την άλλη , όμως, όσο το σύστημα των LBS έχει να κάνει με εμπιστευτικά προσωπικά δεδομένα και πληροφορίες, όπως η τρέχουσα θέση ενός ατόμου , ο προσωπικός αριθμός τηλεφώνου του, η διεύθυνσή του, το ιστορικό των κινήσεων του κ.α., τότε εγείρονται αναπόφευκτα θέματα προσωπικής ασφάλειας και καταστρατήγησης της ιδιωτικής ζωής του καθενός.

Ακόμη, με την έκρηξη του mobile social networking , του instant messaging και των sites κοινωνικής δικτύωσης(Facebook, MySpace) και την ενσωμάτωσή τους στις εφαρμογές των κινητών τηλεφώνων, η πιθανότητα κατάχρησης των προσωπικών δεδομένων και κατ' επέκταση της ατομικής ελευθερίας γίνεται ολοένα και μεγαλύτερη , με αποτέλεσμα να αυξάνεται ταυτόχρονα και ο σκεπτικισμός γύρω από την ασφάλεια χρήσης των LBS.

Έτσι, είναι κρίσιμο πλέον, τόσο για τους παρόχους των υπηρεσιών αυτών, όσο και για τις επιχειρήσεις που προτίθενται να επενδύσουν στον τομέα την περαιτέρω ανάπτυξης των LBS , να σχεδιάσουν τη δομή ασφαλείας του συστήματος με γνώμονα πάντα την προστασία των προσωπικών δεδομένων των χρηστών.

1.16.1 Πρόσφατες εξελίξεις στον τομέα της προστασίας της ιδιωτικής ζωής των κινητών χρηστών

Τα πρόσφατο εύρημα ότι τα I-phones και τα android κινητά τηλέφωνα αποθηκεύουν τα δεδομένα θέσης των χρηστών έχει προκαλέσει την προσοχή των ανθρώπων που ασχολούνται με θέματα ασφάλειας προσωπικών δεδομένων και προστασίας της ιδιωτικής ζωής του καθενός. Συγκεκριμένα, εντοπίστηκε στα I-Phones και στα I-Pads αρχείο που αποθήκευε το ιστορικό των τοποθεσιών που βρέθηκε ο χρήστης και μάλιστα το αρχείο αυτό ήταν «εκτεθειμένο» χωρίς καμία κρυπτογράφηση, προστασία ή κάποιο όριο μεγέθους, ενώ παράλληλα μπορούσε να αναγνωρισθεί από οποιονδήποτε υπολογιστή που είχε τη δυνατότητα συγχρονισμού με τις I-OS συσκευές. Επίσης, το συγκεκριμένο αρχείο μπορούσε να γίνει προσβάσιμο και μέσω της ίδιας συσκευής αν αυτή έπεφτε σε λάθος χέρια.

Στη συνέχεια ακολούθησαν αναφορές πως και η Google αποθήκευε επίσης δεδομένα θέσης των χρηστών της στις Android συσκευές.

Η απάντηση από την Apple στην ανησυχία και το σκεπτικισμό που είχε δημιουργηθεί ήρθε μια εβδομάδα αργότερα, οπότε και δόθηκαν οι απαραίτητες διαβεβαιώσεις πως η παροχή LBS εφαρμογών ήταν απολύτως ασφαλής και πως τα προσωπικά δεδομένα των χρηστών φυλάσσονταν από τους διαχειριστές και ουδέποτε χρησιμοποιήθηκαν.

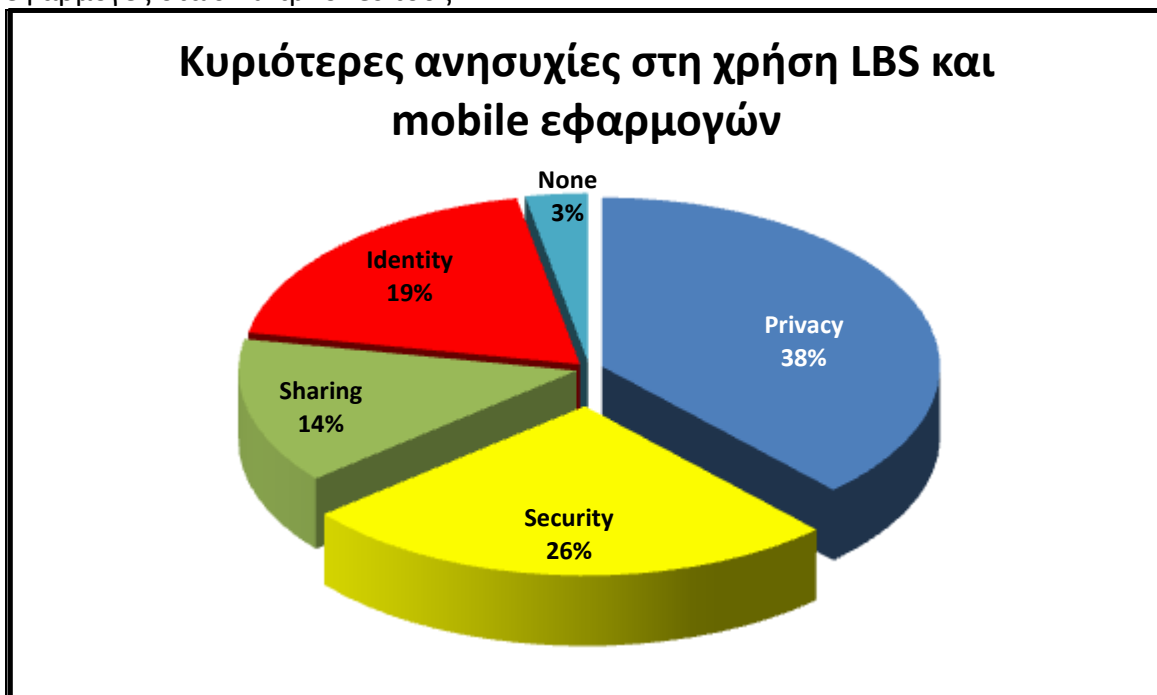
Στη συνέχεια η Microsoft παραδέχτηκε και αυτή, πως αποθήκευε δεδομένα θέσης αλλά με έναν πιο ασφαλή τρόπο, από ότι οι άλλες δύο εταιρείες. Ο θόρυβος και η ανησυχία στο καταναλωτικό κοινό που προκλήθηκε κάποια στιγμή είναι βέβαιο πως θα κοπάσει, αλλά το μόνο σίγουρο είναι πως μέσω των smartphones οι χρήστες παρακολουθούνται ανά πάσα στιγμή, όπου και αν βρίσκονται.

Τα παραπάνω ευρήματα ήταν απλά η αφορμή για να δοθεί περισσότερη έμφαση και προσοχή σε ήδη υπάρχουσες αναφορές για κατάχρηση των πληροφοριών που χρησιμοποιούνταν από τις LBS και για να τροφοδοτηθεί μια σειρά ερωτημάτων από τους χρήστες των εφαρμογών αυτών.

- Δημιουργούνται πράγματι απειλές για την προσωπική ζωή των χρηστών από τη χρήση των τεχνολογιών εντοπισμού θέσης;
- Χρειάζεται οι χρήστες να μπορούν να καταλαβαίνουν τι είδους πληροφορία συλλέγεται και πως οι κινήσεις τους παρακολουθούνται;
- Υπάρχει ένα υγιές trade-off ανάμεσα στην απώλεια της ελευθερίας της προσωπικής ζωής και στην παροχή υπηρεσιών προσανατολισμένων στο προφίλ του κάθε χρήστη;
- Ποιός συλλέγει, διαχειρίζεται και επεξεργάζεται τα δεδομένα θέσης και τι δικαιοδοσία έχει λάβει από το άτομο που παρακολουθείται;

Όλες οι παραπάνω ερωτήσεις πρέπει να λάβουν σύντομα απάντηση και να λαμβάνονται συνεχώς υπόψη από τους παρόχους και τους developers LBS εφαρμογών, καθώς οι χρήστες γίνονται όλο και πιο ευαισθητοποιημένοι σε θέματα που αφορούν τα προσωπικά τους δεδομένα και τον τρόπο που αυτά χειρίζονται.

Στο επόμενο σχήμα παρατίθεται μια πρόσφατη έρευνα που διεξήχθη από την εταιρεία Harris Interactive υπό την επίβλεψη της TRUSTe και δείχνει κάποια ενδιαφέροντα αποτελέσματα σχετικά με τις ανησυχίες και τη συμπεριφορά των χρηστών όταν αφορά τις location-based εφαρμογές στα smartphones τους.



Σχήμα 16. Έρευνα των Harris Interactive και TRUSTe πάνω στην προστασία της ιδιωτικής ζωής των κινητών χρηστών-Φεβρουάριος 2011

1.16.2 Κυριότερα ζητήματα που απασχολούν τους χρήστες των LBS σχετικά με τα smartphones τους

Προστασία της ιδιωτικής ζωής(Privacy) και Ασφάλεια (Security)

Όπως γίνεται σαφές από τα αποτελέσματα της έρευνας , οι καταναλωτές νιώθουν πως η προστασία της ιδιωτικής τους ζωής(privacy) είναι η πρώτη τους προτεραιότητα και μάλιστα το 38% των συμμετεχόντων στην έρευνα μοιράζεται αυτή την άποψη. Επίσης, όπως αναμενόταν άλλωστε, οι φόβοι για κατάχρηση των προσωπικών δεδομένων και εισβολή σε πληροφορίες της ιδιωτικής ζωής των ατόμων, είναι πιο έντονοι σε χρήστες μεγαλύτερης ηλικίας από ότι στους πιο νέους. Γενικότερα, η έρευνα έδειξε πως οι χρήστες δεν εμπιστεύονται όσο θα περίμενε κανείς το application store και μόνο το 25% θεωρεί πως τους παρέχει εφαρμογές οι οποίες εγγυώνται για τα προσωπικά τους στοιχεία.

Εξίσου ζωτικής σημασίας για τους mobile χρήστες είναι η ασφάλεια(security) με 26% των ερωτηθέντων να την θεωρεί ως πρωτεύουσας σημασίας. Η ασφάλεια ήταν πάντα πρωταρχική ανησυχία για τους καταναλωτές και απλά η έρευνα ήρθε για να το επιβεβαιώσει.

Το παράξενο όμως είναι πως μόνο το ένα τρίτο των χρηστών που θεωρεί αναγκαίο χαρακτηριστικό για κάθε mobile ή location-based εφαρμογή την προστασία της ιδιωτικής ζωής και την ασφάλεια, νιώθει πως έχει τον πλήρη έλεγχο της κινητής του συσκευής.

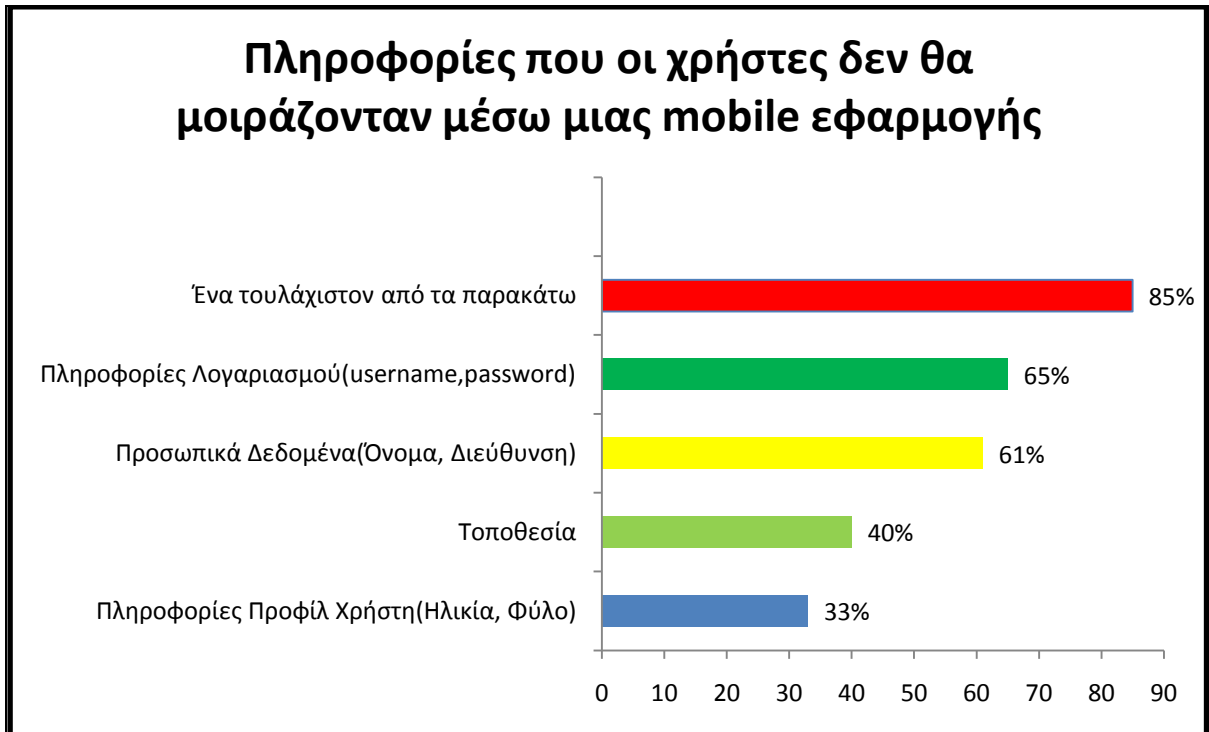
Παρακολούθηση της θέσης και των δραστηριοτήτων (Location and Activity Tracking)

Ένα ακόμη αποτέλεσμα που είχε ο πρόσφατος θόρυβος που έχει προκληθεί από τις ανακαλύψεις των αρχείων με τα δεδομένα θέσης που υπάρχουν στα κινητά, είναι ότι οι χρήστες δεν θέλουν σε καμία περίπτωση να παρακολουθούνται. Η επιβεβαίωση αυτής της αίσθησης έρχεται από αποτελέσματα που δείχνουν πως η πλειονότητα των καταναλωτών δε θέλουν να μοιράζονται δεδομένα θέσης με τους παρόχους των εφαρμογών. Μάλιστα, το 77% των χρηστών δεν θέλει να μοιράζεται το πού βρίσκεται με κανέναν και μόλις το 2% συμφωνεί να μοιράζεται πληροφορίες θέσης με κοντινούς του ανθρώπους παρά με τρίτους. Επίσης, οι καταναλωτές που παίρνουν μηνύματα-ενημερώσεις (push location based services) είναι πιο πιθανό να επιτρέπουν το location sharing , θεωρώντας πάντα πως όταν είναι αναγκαίο για έναν developer μιας εφαρμογής να έχει πρόσβαση στις πληροφορίες θέσης του χρήστη, τότε μπορεί να δίνεται η δυνατότητα στο χρήστη με ένα απλό pop-up παράθυρο να επιβεβαιώνει ή όχι την «εκμετάλλευση» των δεδομένων θέσης του. Μ' αυτό τον τρόπο υπάρχει μια αίσθηση εμπιστοσύνης μεταξύ των δύο πλευρών.

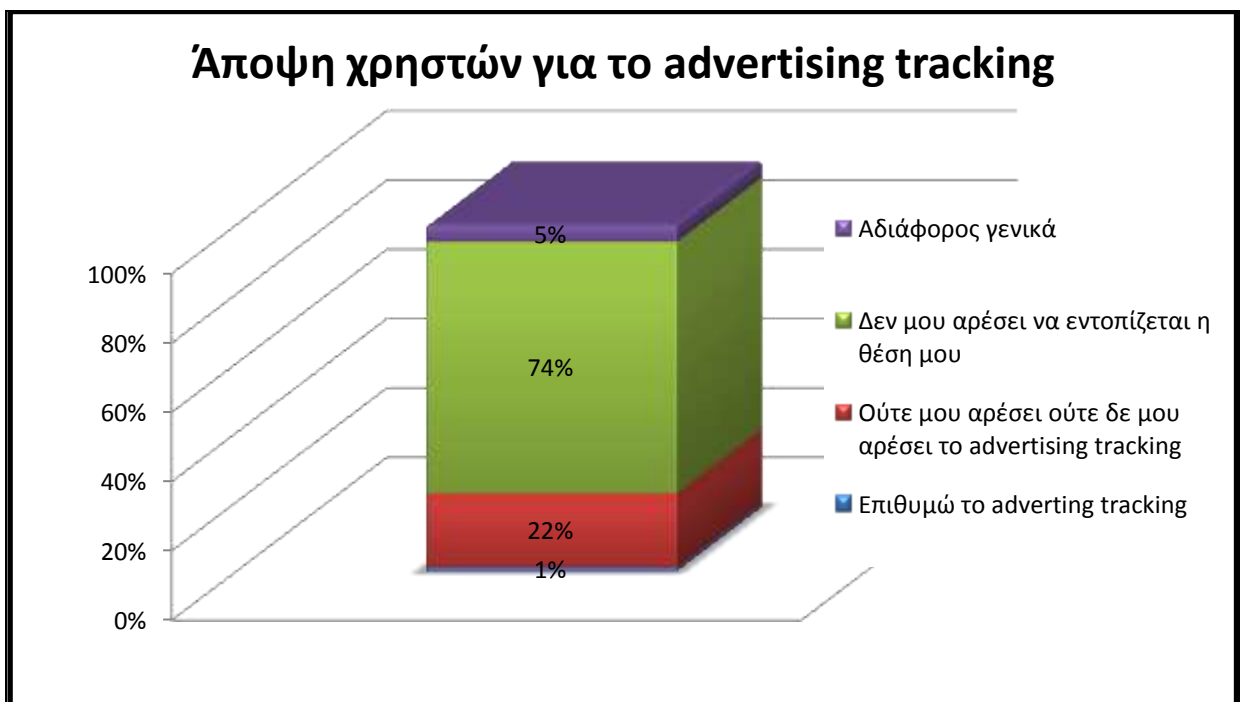
Εντοπισμός Θέσης και Διαφήμιση(Advertising tracking)

Οι διαφημίσεις είναι μια μεγάλη υπόθεση στον κόσμο των location-based εφαρμογών και κανείς δεν μπορεί να παραγνωρίσει το γεγονός πως τα κύρια έσοδα των επιχειρήσεων και των παρόχων υπηρεσιών που βασίζονται στη θέση των χρηστών θα προέρχονται από εκεί. Ο εντοπισμός της θέσης των χρηστών για προβολή διαφημίσεων στα κινητά τους τερματικά γίνεται ολοένα και πιο συχνός και μάλιστα το 85% των χρηστών θέλουν να μη γίνονται δέκτες στοχευμένων διαφημίσεων, ενώ το 74% νιώθει άβολα με την ιδέα ότι τα δεδομένα θέσης τους θα χρησιμοποιούνται για καταιγισμό διαφημιστικών μηνυμάτων. Εντυπωσιακό είναι επίσης το γεγονός πως το 68% των χρηστών δηλώνουν πως εντοπίζονται και δέχονται τέτοιου είδους διαφημίσεις.

Στα επόμενα γραφήματα παρουσιάζονται κάποιες σημαντικές πληροφορίες που συλλέχθηκαν από την έρευνα των TRUSTe και Harris Interactive και δίνουν με σαφή τρόπο το προφίλ και τη συμπεριφορά των καταναλωτών-χρηστών στα κυριότερα ζητήματα που τους ανησυχούν γύρω από τα smartphones τους.



Σχήμα 17. Πληροφορίες που οι χρήστες δε θα μοιράζονταν μέσω των LBS (Έρευνα των Harris Interactive και TRUSTe-Φεβρουάριος 2011)



Σχήμα 18. Αισθήματα καταναλωτών σχετικά με το advertising tracking

Από τα παραπάνω γραφήματα γίνεται σαφές πως οι περισσότεροι χρήστες δε θέλουν να εντοπίζονται και γενικά να μη διοχετεύονται τα δεδομένα θέσης τους από τους διαχειριστές των LBS και ειδικά για εφαρμογές advertising tracking(74% από τους ερωτηθέντες δεν επιθυμεί να λαμβάνει ειδοποιήσεις)

Τέλος, η έρευνα αυτή καταλήγει στο ότι μόνο ένα μικρό ποσοστό των χρηστών νιώθουν πως έχουν επιλογή και μπορούν να ελέγξουν τη συλλογή και την επεξεργασία των δεδομένων θέσης που χρησιμοποιούνται από μια δεδομένη εφαρμογή. Από αυτούς οι χρήστες I-phone (45%) και οι χρήστες Android συσκευών(41%) θεωρούν πως έχουν μια μεγαλύτερη επιλογή και ευελιξία στο συγκεκριμένο ζήτημα σε σχέση με τους χρήστες συσκευών Windows Phone(38%) και BlackBerry(20%).

Συμπερασματικά, ο μέσος χρήστης των LBS εφαρμογών, όπως προκύπτει από τη έρευνα θέλει:

1. **Έλεγχο επί των δεδομένων που μοιράζεται**(Control Over Sharing) : Οι χρήστες ζητούν περισσότερο έλεγχο επί των προσωπικών τους δεδομένων και 98% από αυτούς απαιτούν εύκολη πρόσβαση μέσω των κινητών τους τερματικών στο να επιτρέπουν ή να μπλοκάρουν τη συλλογή και χρήση των δεδομένων θέσης τους από τους διαχειριστές/ παρόχους των LBS. Ακόμη και στις οικονομικές συναλλαγές, εννιά στους δέκα καταναλωτές εκφράζουν την πρόθεσή τους να μη μοιράζονται τις επαφές τους, τις φωτογραφίες τους και τις συνήθειες τακτικές που ακολουθούν στην πλοήγηση στο web.
2. **Ευκολία και Άνεση όταν χρησιμοποιούν λογαριασμούς για την ενεργοποίηση location-based εφαρμογών σε συνδυασμό με ασφάλεια και προστασία των ιδιωτικών τους δεδομένων** : Περίπου μισοί από τους χρήστες που ρωτήθηκαν θεωρούν πως δεν νιώθουν άνετα ή το βρίσκουν αρκετά πολύπλοκο και χρονοβόρο να χρησιμοποιούν λογαριασμούς(sign-on accounts) για τα εισέλθουν στις εφαρμογές. Τα αποτελέσματα αυτά ποικίλλουν ανάλογα με την ηλικία με τους πιο νέους να επιλέγουν την ευκολία χρήσης των εφαρμογών, ενώ τους μεγαλύτερους σε ηλικία να προτιμούν περισσότερο έλεγχο και μεγαλύτερη ασφάλεια.
3. **Διαφάνεια στα δεδομένα που διαμοιράζονται**(Transparency of shared data): Περίπου το 45% των χρηστών υποστηρίζουν πως η διαφάνεια, όσον αφορά το τι δεδομένα συλλέγονται είναι εξαιρετικής σημασίας, ώστε να έχουν μια ιδέα για το ποιος κάνει τι. Οι καταναλωτές μάλιστα απάντησαν πως είναι θετικοί στην ύπαρξη εφαρμογών, οι οποίες θα παρέχουν πληροφορίες για τα δεδομένα που συλλέγουν.

1.16.3 Νομοθετικές ρυθμίσεις περί προστασίας των προσωπικών δεδομένων

Στα πλαίσια της προστασίας της ιδιωτικής ζωής των χρηστών και της ασφάλειας των ηλεκτρονικών επικοινωνιών έχουν υπογραφεί νομοθετικές διατάξεις που καλύπτουν τα κυριότερα ζητήματα ασφαλείας όσον αφορά τη χρήση των location-based υπηρεσιών.

Τα κυριότερα αυτά ζητήματα είναι:

- a) Η ακούσια και χωρίς συγκατάθεση από το χρήστη αποστολή μηνυμάτων (spamming)
- b) Η συλλογή, κατάχρηση και εκμετάλλευση των προσωπικών δεδομένων και η διείσδυση στην ιδιωτική ζωή των χρηστών, χωρίς να το ξέρουν και να το επιθυμούν.

Έτσι, το 2005 με ψήφιση νόμου στις ΗΠΑ , γίνεται παράνομη η αποστολή μηνυμάτων στον τελικό χρήστη χωρίς ο χρήστης να το έχει εγκρίνει πιο πριν.

Στην Ευρώπη, παρέχεται ειδικό νομοθετικό πλαίσιο για την προστασία των προσωπικών δεδομένων που πρόκειται να χρησιμοποιηθούν στις LBS και πιο συγκεκριμένα μια σειρά από οδηγίες που αφορούν

- a) Προσωπικά δεδομένα (Οδηγία 95/46/EC)
- b) Προσωπικά δεδομένα και Ηλεκτρονικές Επικοινωνίες (Οδηγία 2002/58/EC)
- c) Κράτηση δεδομένων από παρόχους/διαχειριστές (Οδηγία 2006/24/EC)

Περίληπτικά, το νομοθετικό πλαίσιο έτσι όπως έχει καθοριστεί, αναφέρει πως τα push μηνύματα που ανήκουν στην κατηγορία των push location-based services και στέλνονται στο χρήστη ανάλογα με την τοποθεσία στην οποία βρίσκεται, υπάγονται σε ειδική νομοθετική ρύθμιση τα κυριότερα σημεία της οποίας είναι:

- a) Η αυτόματη κλήση επιτρέπεται μόνο σε εγγεγραμμένους στην υπηρεσία χρήστες που έχουν δώσει εκ των προτέρων τη συγκατάθεσή τους.
- b) Μόνο ο οργανισμός/ εταιρεία όπου ο χρήστης έχει υπογράψει συμβόλαιο/συμφωνία μπορεί να κατέχει τα στοιχεία επικοινωνίας με το χρήστη και να του παρέχει άμεσα διαφημιστικά μηνύματα ή μηνύματα προώθησης προϊόντων όταν εισέρχεται σε μια περιοχή.
- c) Όταν ο πάροχος μιας συγκεκριμένης υπηρεσίας θέλει να κάνει άμεσο marketing, τότε θα πρέπει να δίνεται στο χρήστη η δυνατότητα να αρνηθεί να αποστείλει τα στοιχεία του και τα δεδομένα θέσης του και μάλιστα χωρίς χρέωση και με εύκολο τρόπο. Αυτή η δυνατότητα θα πρέπει να δίνεται σε κάθε μήνυμα.
- d) Ηλεκτρονικά μηνύματα που κρύβουν την ταυτότητα του αποστολέα ή δεν έχουν έγκυρη διεύθυνση για απάντηση πρέπει να απαγορεύονται.

Επίσης, η χρήση δεδομένων που αφορούν την τοποθεσία του χρήστη καθορίζεται και αυτή νομοθετικά:

- Τα location δεδομένα πρέπει να επεξεργάζονται μόνο όταν είναι ανώνυμα ή όταν συλλέγονται με τη συγκατάθεση του χρήστη και μόνο κατά τη διάρκεια που απαιτείται για την παροχή της location-based υπηρεσίας
- Η location-oriented υπηρεσία πρέπει να ενημερώνει τον χρήστη, να παίρνει τη συγκατάθεσή του για τον τύπο των δεδομένων θέσης που θα επεξεργαστεί, για την πρόθεση χρήσης αυτών, τη διάρκεια της επεξεργασίας τους και για το αν τα δεδομένα αυτά θα γίνουν γνωστά σε τρίτους και σε ποιους συγκεκριμένα.
- Πρέπει να δίνεται η δυνατότητα στους χρήστες να μπορούν να αποσύρουν οποιαδήποτε στιγμή το επιθυμούν τη συγκατάθεσή τους για χρήση των προσωπικών δεδομένων θέσης τους και να διακόπτουν την όποια επικοινωνία με τον server της εφαρμογής που τρέχουν εκείνη τη στιγμή.

1.16.4 Τεχνικές προσέγγισης από τους developers για την παροχή ασφάλειας στη χρήση των LBS

Πίσω από το νομοθετικό πλαίσιο, οι ίδιοι οι πάροχοι των υπηρεσιών αυτών θα πρέπει να αναπτύξουν το κατάλληλο πλαίσιο ασφαλείας το οποίο θα εγγυάται προστασία της ιδιωτικότητας των χρηστών που «τρέχουν» την εφαρμογή. Τέτοιες τεχνολογίες που θα βρίσκονται ενσωματωμένες μέσα στην εφαρμογή ονομάζονται *privacy-enhancing technologies* (PETs) και μπορεί να είναι αρκετά απλοϊκές, όπως on/off buttons για ενεργοποίηση/απενεργοποίηση της αποστολής των δεδομένων θέσης αλλά μπορεί να είναι και αρκετά πολύπλοκες μέθοδοι που χρησιμοποιούν τεχνικές ανωνυμίες, όπως k-anonymity. Το σίγουρο είναι πως για τους παρόχους και τους developers των LBS, είναι πλέον επιτακτική η ανάγκη να καθορίσουν το πώς θα προστατεύουν τα δεδομένα θέσης των χρηστών, ανάλογα με:

1. Τη συγκεκριμένη εφαρμογή
2. Την αγορά πελατών στην οποία στοχεύουν

Από τη μία, προειδοποιήσεις ασφαλείας που είναι πολύ ελαστικές μπορεί να μη λαμβάνονται υπόψη από τους πελάτες ή άνευ σημασίας.

Από την άλλη, πολύ αυστηροί περιορισμοί θα προκαλέσουν εκνευρισμό των χρηστών, ειδικά αν συνοδεύονται από ένα πολύπλοκο και δύσχρηστο περιβάλλον, μεγαλύτερα κόστη υλοποίησης κατά τη διάρκεια ανάπτυξης της εφαρμογής και γενικότερα μπορεί να αποθαρρύνουν τους χρήστες από το να εγγραφούν και να ενεργοποιήσουν την εφαρμογή στα κινητά τους τηλέφωνα. Το τελευταίο, όπως γίνεται σαφές, επηρεάζει σε μεγάλο βαθμό τους παρόχους και τις εταιρείες που θέλουν να επενδύσουν στις LBS και ουσιαστικά οι περιορισμοί αυτοί μπορεί να αποτελέσουν τροχοπέδη στη διαδικασία προώθησης των υπηρεσιών τους.

Γενικότερα, οι απαιτήσεις για *privacy* πρέπει να βασίζονται στη φύση της εφαρμογής και στο πόσο πιθανό είναι μέσω της χρήσης της συγκεκριμένης LBS, τα προσωπικά δεδομένα του χρήστη να εκτεθούν. Οι developers θα πρέπει να καθορίσουν την προστασία των προσωπικών δεδομένων του χρήστη ανάλογα με τον τύπο της αλληλεπίδρασης της κάθε εφαρμογής. Πιο συγκεκριμένα,

- Εφαρμογές που απευθύνονται σε ένα χρήστη (*single-user applications*) σε ένα κλειστό περιβάλλον με άμεση σύνδεση σε πηγές δεδομένων είναι περιπτώσεις όπου απαιτείται φιλτράρισμα του άσκοπου *spamming* και γενικότερα βελτιστοποίηση της δομής ασφαλείας, ώστε να προλαμβάνεται η κατάχρηση των προσωπικών πληροφοριών των χρηστών.
Είναι ακριβώς αυτή η περίπτωση που οι LBS αρχίζουν να παρέχουν πρόσωπο-με-πρόσωπο αλληλεπίδραση σε συνδυασμό με πληροφορίες θέσης, οπότε και τα ζητήματα ασφαλείας γίνονται πιο επιτακτικά.
- Εφαρμογές που προσανατολίζονται σε αλληλεπίδραση μελών μιας οικογένειας, ή γενικότερα μιας κλειστής ομάδας, χρειάζονται γενικά ένα σχετικά χαμηλότερο επίπεδο προστασίας, αφού τα μέλη δίνουν εκ των προτέρων τη συγκατάθεσή τους για αλληλεπίδραση και γνωρίζονται μεταξύ τους.

- Εφαρμογές με εντελώς ξένους απαιτούν το υψηλότερο επίπεδο προστασίας και ασφάλειας

Παρακάτω φαίνεται γραφικά η πυραμίδα-πλαίσιο που θα πρέπει να ακολουθούν οι developers για να ελέγχουν τι είδους προδιαγραφές ασφαλείας θα πρέπει να υιοθετούν και να ενσωματώνουν σε συγκεκριμένες εφαρμογές.



Σχήμα 19. Πυραμίδα με την προτεραιότητα των privacy requirements των LBS

Είναι σαφές πως όλες οι location-based εφαρμογές χρειάζονται ένα βασικό set προδιαγραφών ασφαλείας κορμού, που προφανώς θα περιλαμβάνει υιοθέτηση των standards που διέπουν τις ασύρματες επικοινωνίες, αλλά ειδικότερα θα ικανοποιούν απαιτήσεις εμπιστευτικότητας των πληροφοριών των χρηστών.

Όπως παρατηρούμε στην πυραμίδα, καθώς κινούμαστε προς το κάτω μέρος της, οπότε και η πληροφορία για την τοποθεσία του χρήστη γίνεται διαθέσιμη σε περισσότερους ανθρώπους, επιπρόσθετες δυνατότητες για privacy χρειάζεται να προστεθούν. Πρόσθετα στοιχεία στις εφαρμογές, όπως πλήκτρα busy/not busy ή work/not work είναι σημαντικά γιατί παρέχουν πληροφορίες «παρουσίας» ή μη του χρήστη και του εξασφαλίζουν την απαιτούμενη χωρίς διακοπές από τρίτους χρήση της υπηρεσίας. Οι business applications που μπορεί να χρησιμοποιούνται για διαχείριση στόλου ή η παρακολούθηση της εργασίας των τεχνικών μιας εταιρείας θα πρέπει να είναι εφοδιασμένες με δυνατότητα απενεργοποίησης του εντοπισμού θέσης σε ώρες εκτός ωραρίου εργασίας. Οι εφαρμογές του mobile social networking και του mobile gaming χρειάζονται επίσης ισχυρές δικλίδες προστασίας των προσωπικών δεδομένων. Εφαρμογές που περιλαμβάνουν online φίλους που δεν έχουν συναντηθεί ποτέ πιο πριν, χρειάζεται να παρέχουν προστασία των προφίλ των χρηστών όσον αφορά σχολιασμό τους από τρίτους.

Το mobile gaming που χρησιμοποιεί δεδομένα θέσης των παικτών χρειάζεται να παρέχει πρωτοποριακές δυνατότητες ασφαλείας ώστε να επιτρέπει μεν τον ανταγωνισμό αλλά παράλληλα να προστατεύει τους παίκτες από το να εντοπίζονται από αγνώστους. Αυτό επιτυγχάνεται είτε αλλάζοντας την αναφερθείσα τοποθεσία του χρήστη είτε αναφέροντας μια ευρύτερη περιοχή στην οποία βρίσκεται.

Μερικές από αυτές τις ιδέες είναι τελείως καινούριες στη φιλοσοφία των LBS, και πολλές από αυτές προέρχονται από τον online κόσμο και τα sites κοινωνικής δικτύωσης, όπως το μπλοκάρισμα ή όχι της άμεσης αποστολής μηνυμάτων(instant messaging), οι τροποποιήσεις της κατάστασης του χρήστη(status adjustments όπως available, busy , offline) , οι λίστες ομάδων ατόμων που θέλουμε να αγνοήσουμε(ignore lists), οι φόρμες εγγραφής στα sites του eBay κ.α.

Συνοπτικά, οι developers δε θα πρέπει να αφήνουν τα ζητήματα της ασφάλειας των δεδομένων των χρηστών στο τέλος της υλοποίησης της εφαρμογής ή να τα αντιμετωπίζουν σαν ένα ανεξάρτητο κομμάτι της λειτουργίας της. Αντιθέτως, θα πρέπει η διαδικασία αυτή να ενσωματωθεί στις κύριες προδιαγραφές ανάπτυξης έτσι ώστε να εξασφαλιστούν τα απαραίτητα επίπεδα προστασίας και η εφαρμογή να γίνεται όσο πιο εύκολη και άνετη στη χρήση.

Κάνοντας αυτά οι πάροχοι των LBS θα προσφέρουν ηρεμία στον καταναλωτή σε θέματα ασφαλείας, θα κάνουν τις εφαρμογές τους πιο ελκυστικές σε μεγαλύτερο μερίδιο αγοράς και κυρίως θα προλάβουν πιθανά περιστατικά που θα μπορούσαν να είναι καταστροφικά για την επένδυση στις location-based υπηρεσίες.

1.16.5 Μοντέλα προστασίας των προσωπικών δεδομένων (privacy models)

Για την επίτευξη της ανωνυμίας στη μεταφορά των προσωπικών δεδομένων των χρηστών που χρησιμοποιούν τις LBS έχουν προταθεί αρκετές λύσεις.

Οι πιο απλές και λιγότερο πολύπλοκες ως προς την υλοποίησή τους τεχνικές αναπτύχθηκαν σχετικά νωρίς και περιλαμβάνουν:

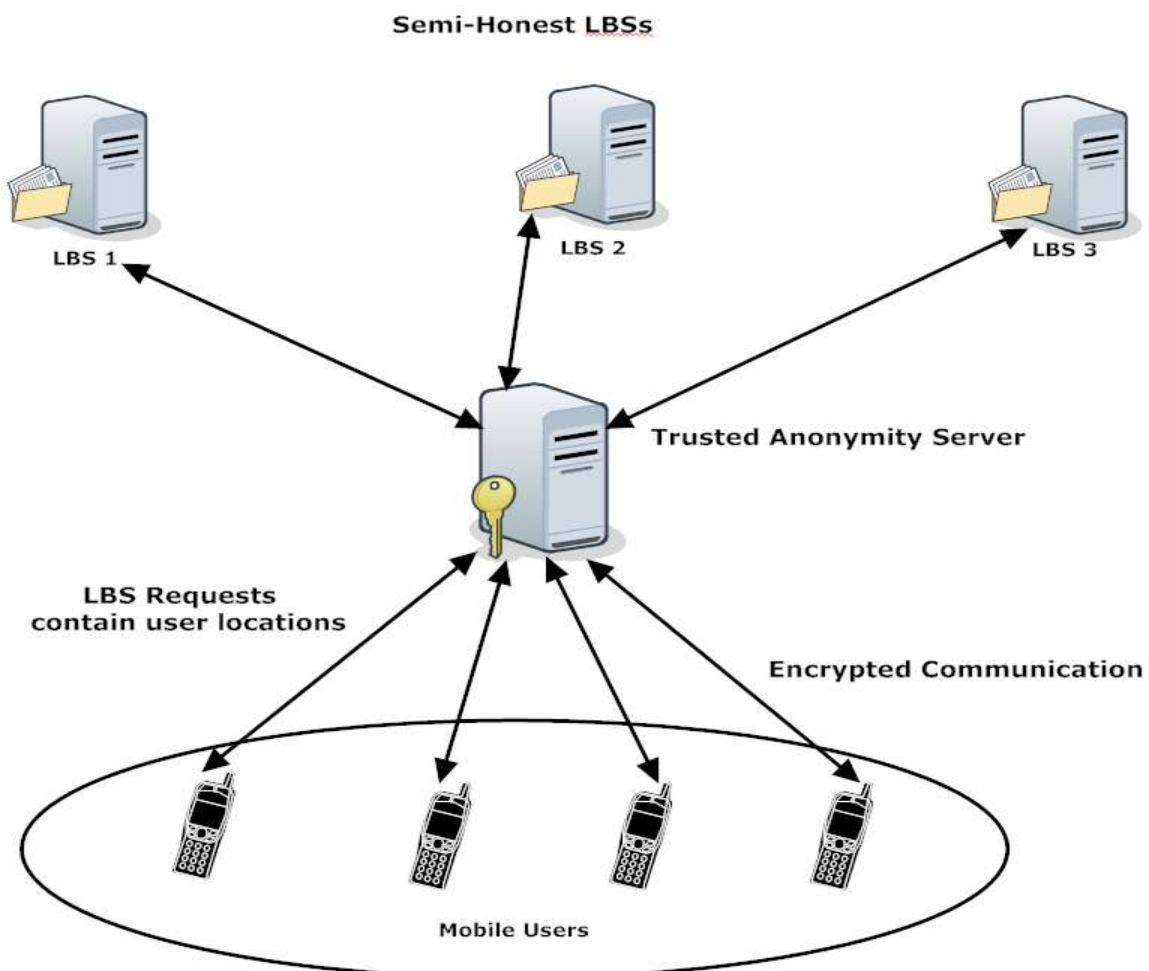
- a) Είτε απευθείας ερώτημα στον server των LBS από τον client, οπότε ο server μπορεί να απαντήσει με λίγα αποτελέσματα, εστιασμένα στις απαιτήσεις του χρήστη, αλλά από την άλλη ο server γνωρίζει άμεσα την τοποθεσία του client(που δεν το θέλουμε στα πλαίσια της ανώνυμης επικοινωνίας να συμβαίνει)
- b) Είτε με απευθείας μεταφορά όλων των δεδομένων από τον server προς τον χρήστη, όπου το καλό είναι πως η τοποθεσία του χρήστη παραμένει άγνωστη αλλά έχουμε μεγάλο κόστος στη μεταφορά τόσων πολλών δεδομένων
- c) Είτε με αποστολή από τον server στον client ενός συνόλου σημείων ενδιαφέροντος και όχι όλων , οπότε από τη μία μειώνεται το κόστος σε σχέση με την προηγούμενη λύση, ο server δεν γνωρίζει και πάλι τη θέση του χρήστη αλλά είναι προφανές πως το τελικό αποτέλεσμα δεν θα είναι καθόλου ακριβές.

Γίνεται σαφές πως οι παραπάνω baseline solutions δεν εξασφάλιζαν καμία αποδοτικότητα στη λειτουργία των LBS. Έτσι, άρχισαν να προτείνονται πιο πολύπλοκοι αλγόριθμοι που από τη μία θα εξασφάλιζαν την προστασία των προσωπικών δεδομένων των χρηστών και την ανωνυμία τους ενώ ταυτόχρονα θα εγγυώνται αποδοτική λειτουργία και ακριβείς απαντήσεις στις εκάστοτε απαιτήσεις του χρήστη.

Στα πλαίσια της χρήσης των LBS έχουν προταθεί δύο προσεγγίσεις για την προστασία της location privacy των χρηστών: οι policy-based προσεγγίσεις, όπου οι χρήστες των κινητών προδιαγράφουν τις προτιμήσεις τους και τις επιλογές τους ως προς την ασφάλεια των προσωπικών τους δεδομένων και ουσιαστικά εμπιστεύονται τους παρόχους των LBS(είναι σαν να υπογράφουν μια συμφωνία, όπου θέτονται οι κανόνες χρήσης των υπηρεσιών αυτών) και οι anonymity-based πολιτικές όπου οι πάροχοι δεν έχουν την πλήρη εμπιστοσύνη των χρηστών.

Μια τέτοια anonymity-based λύση είναι η k-anonymity προσέγγιση και η transformation-based matching προσέγγιση που θα περιγράψουμε επιγραμματικά παρακάτω.

Και στις δύο αυτές περιπτώσεις, λόγω της έλλειψης απόλυτης εμπιστοσύνης στον server προσδιορίζεται μια περίμετρος ασφαλείας γύρω από τη θέση του κινητού χρήστη/πελάτη. Η περιοχή αυτή ασφαλείας στην περίπτωση των anonymity-based προσεγγίσεων περιλαμβάνει έναν trusted anonymity server και ένα κανάλι ασφαλείας για την επικοινωνία των δύο οντοτήτων, του client και του server, η οποία γίνεται με χρήση κρυπτογραφίας.

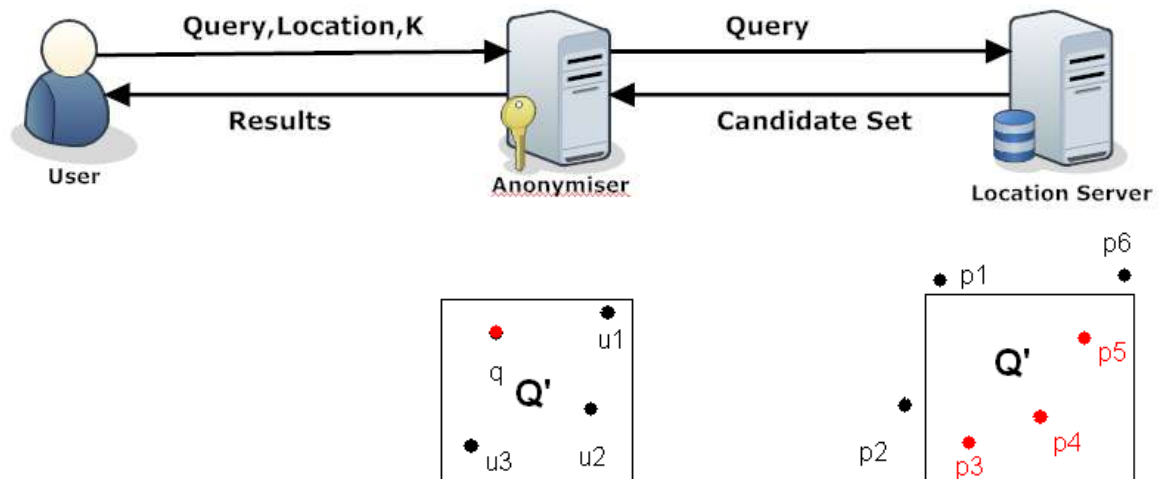


Όταν μιλάμε για όχι απόλυτα έμπιστο server εννοούμε πως οι LBS πάροχοι φυσικά και μπορούν να επεξεργαστούν και να απαντήσουν στα αιτήματα των χρηστών, αλλά από την άλλη μπορεί να θέλουν να προσδιορίσουν και την ταυτότητα του πελάτη. Ακόμα και αν δεν λάβουμε υπόψη μας την περίπτωση της εκούσιας εκμετάλλευσης των προσωπικών δεδομένων θέσης και της κατάχρησης αυτών, η γνώση και μόνο ότι οι server μιας location-based εφαρμογής μπορούν ανά πάσα στιγμή να εντοπίσουν έναν χρήστη, δεν είναι το επιθυμητό.

1.16.5.1 Location k-anonymity προσέγγιση

Η ιδέα της προσέγγισης αυτής βρίσκεται στο ότι ο χρήστης-client βρίσκεται σε ένα σημείο, έστω q και στέλνει ένα αίτημα για κάποιο ή κάποια σημεία ενδιαφέροντος (points of interest-POI) στον anonymity server. Στη συνέχεια, ο «έμπιστος» αυτός server υπολογίζει μια k -anonymous περιοχή Q' γύρω από τη θέση εντοπισμού q . Η περιοχή αυτή περιέχει την τοποθεσία q του χρήστη και τουλάχιστον άλλες $k-1$ διακριτές θέσεις(συνολικά λοιπόν k διακριτές τοποθεσίες). Στη συνέχεια, ο anonymity server στέλνει την περιοχή Q' στον server της location-based υπηρεσίας. Ο location server υπολογίζει ένα set αποτελεσμάτων που περιλαμβάνει όλα τα πιθανά σημεία ενδιαφέροντος γύρω από όλες τις k διακριτές θέσεις(βλέπε στο σχήμα $p_1, p_2, p_3, p_4, p_5, p_6$) και τα στέλνει πίσω στον anonymity server. Εκείνος, τέλος, υπολογίζει από το set αυτών των αποτελεσμάτων το ζητούμενο για τον χρήστη και το στέλνει σ' αυτόν.

Γραφικά, η ροή των μηνυμάτων που ανταλλάσσονται φαίνεται παρακάτω:



Το ζήτημα στην προσέγγιση αυτή είναι ο υπολογισμός της k -anonymous περιοχής Q' .

Το πλεονέκτημα αυτής της μεθόδου είναι ότι παρέχει ισχυρή προστασία και εγγυάται ασφάλεια της πληροφορίας θέσης του χρήστη ακόμα και αν ο «εισβολέας» που θέλει να επιτεθεί γνωρίζει τις θέσεις όλων των χρηστών.

Τα κύρια μειονεκτήματα της privacy-enhancing αυτής τεχνικής είναι η χρήση ενός trusted anonymous server, ο οποίος είναι αυτός που συνδέει client και server και πιθανή κατάρρευσή του θα οδηγήσει σε αδιέξοδο όλο το σύστημα παροχής υπηρεσιών. Επίσης, ερωτηματικό παραμένει και το κατά πόσον είναι ο server αυτός πιο έμπιστος από τον location server.

Ένα δεύτερο πρόβλημα είναι πως οι χρήστες ακόμα και αν δεν έχουν να κάνουν κάποια αίτηση προς τον server, πρέπει να τον ενημερώνουν συνεχώς για την τρέχουσα θέση τους (location update) , πράγμα που ισοδυναμεί με μεγάλο κόστος επεξεργασίας και μεγάλο όγκο αποστολής δεδομένων μέσω του καναλιού επικοινωνίας.

1.16.5.2 Transformation-based matching τεχνική

Η λύση αυτή αποφεύγει τα προβλήματα που μπορεί να δημιουργούνται από τη χρήση του “έμπιστου” anonymous server και βασίζεται στο μοντέλο της τυπικής αρχιτεκτονικής client-server. Στη μέθοδο αυτή τα 2D σημεία που αντιστοιχούν στις θέσεις των χρηστών μετασχηματίζονται σε μονοδιάστατες τιμές οι οποίες δεν έχουν κάποιο νόημα για οποιονδήποτε άλλο εκτός από τους παρόχους των LBS και τους χρήστες που στέλνουν το αίτημα.

Για παράδειγμα, η τοποθεσία με συντεταγμένες (2,5) → Τιμή 18; Location (4,6) → Value 13.

Η προσέγγιση αυτή λειτουργεί ως εξής:

Ο πελάτης/client που βρίσκεται στη θέση q στέλνει στον server την μετασχηματισμένη κατά Hilbert τιμή $H(q)$. Ο server έχει ήδη μετασχηματίσει κάθε τοποθεσία του σημείου ενδιαφέροντος σε τιμές κατά Hilbert, δηλαδή για τα σημεία ενδιαφέροντος p_1, p_2, p_3 κτλ υπάρχουν οι τιμές $H(p_1), H(p_2), H(p_3)$. Όταν ο server δέχεται την τιμή $H(q)$ βρίσκει την πιο κοντινή στην $H(q)$ τιμή Hilbert και την στέλνει στον client. Ο χρήστης στη συνέχεια αποκωδικοποιεί την τιμή αυτή στη δεδομένη τοποθεσία.

Η μόνη πολυπλοκότητα που παρουσιάζει η μέθοδος αυτή είναι ότι ο server καλείται να υπολογίσει διαφορές ανάμεσα στις τιμές Hilbert που αντικατοπτρίζουν ουσιαστικά τις πραγματικές αποστάσεις των σημείων ενδιαφέροντος από την τρέχουσα θέση του χρήστη.

Η μέθοδος αυτή παρουσιάζει πλήρη ανωνυμία (full domain anonymity) ειδικά στην περίπτωση που η συνάρτηση μετασχηματισμού είναι πλήρως μη αναστρέψιμη από τον “εισβολέα”, οπότε δεν μπορεί να γίνει διακριτό τι αντιπροσωπεύουν οι τιμές που βλέπει και ποια η σύνδεσή τους με τα πραγματικά δεδομένα θέσης.

Συνοπτικά, η μέθοδος αυτή μπορεί να παρέχει μια κάποια ασφάλεια , χωρίς να χρησιμοποιεί ενδιάμεσο server και σε πρώτη φάση να μην ανταλλάσσονται άμεσα οι πραγματικές τοποθεσίες του χρήστη. Από την άλλη όμως, η ακρίβεια και η ορθότητα των αποτελεσμάτων που επιστρέφονται δεν είναι η καλύτερη δυνατή και δεν υπάρχει καμιά εγγύηση γι αυτό. Επίσης, η όλη διαδικασία απαιτεί ένα προκαταρκτικό στάδιο, αφού ο server θα πρέπει να έχει υπολογίσει τις τιμές Hilbert για όλα τα σημεία ενδιαφέροντος. Τέλος, ο όποιος θέλει να εισβάλλει και να υποκλέψει τα δεδομένα θέσης μπορεί μετά από κάποιο διάστημα αν οι τιμές Hilbert παραμένουν οι ίδιες , να βρει μια προσέγγιση της συνάρτησης Hilbert και να αποκρυπτογραφήσει τις πληροφορίες με τις πραγματικές συντεταγμένες θέσης.

Περισσότερες πληροφορίες σχετικές με Anonymous Mobile Community Services(AMCs) , την αρχιτεκτονική αυτών και τη λειτουργία τους καθώς και μια πιο λεπτομερή περιγραφή της location k-anonymity προσέγγισης βρίσκονται στα papers [16], [17].

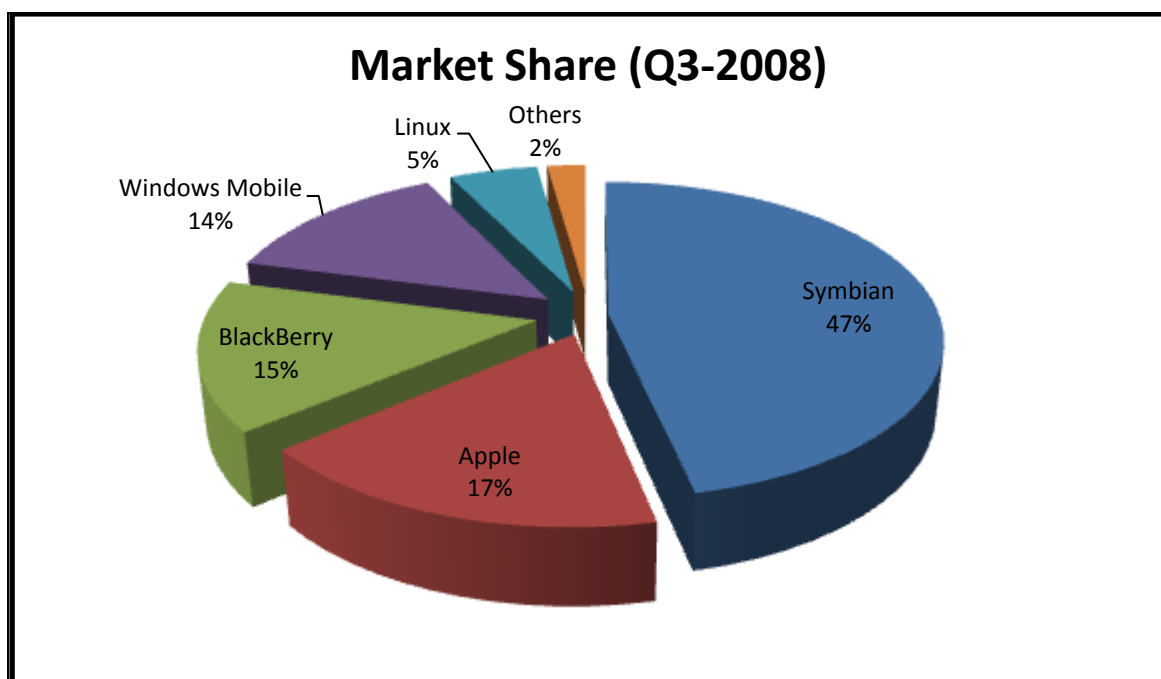
Κεφάλαιο 2

Λογισμικό Android

2.1 Εξελίξεις στην αγορά των smartphones- Εμφάνιση του λογισμικού Android

Πριν το μέσο του 2007, οι διαθέσιμες πλατφόρμες λογισμικού για κινητά τηλέφωνα ήταν οι J2ME, Symbian της Nokia , Windows Mobile , RIM και Palm OS , ενώ τα κινητά τερματικά στα οποία “έτρεχαν” τα παραπάνω λογισμικά ήταν ογκώδεις συσκευές με περιορισμένες δυνατότητες και μικρό μέγεθος οθόνης. Παράλληλα, οι ενσωματωμένοι browsers είχαν αρκετά προβλήματα λειτουργίας, το mobile internet ήταν σε πρώιμο στάδιο και η μεταφορά δεδομένων ήταν αρκετά ακριβή.

Το μεγάλο βήμα στον τομέα του mobile computing έγινε το 2008, όταν η Apple εισήγαγε στην αγορά το iPhone σαν ένα ολοκληρωμένο πακέτο. Το iPhone “τάραξε τα νερά” προσφέροντας στους καταναλωτές μια λύση κομψή και ελκυστική, με πλήρως εξοπλισμένο browser και εύκολη μεταφορά δεδομένων σε μια λεπτή συσκευή με μεγάλο σχετικά μέγεθος οθόνης αφής και χωρίς πλήκτρα. Αν και η δυνατότητα για ανάπτυξη εφαρμογών ήταν εύκολη, η Apple είχε πλήρη και αυστηρό έλεγχο του τι είδους εφαρμογή θα “ανέβει” στο App Store και θα “τρέξει” στο iPhone. Φιλοσοφία της Apple ήταν πως το iPhone δεν ήταν απλά ένα κινητό τηλέφωνο αλλά ένα νέο brand το οποίο προσπαθούσε να προωθήσει. Έτσι, μέχρι το τρίτο τρίμηνο του 2008 , το μερίδιο αγοράς στο χώρο του λογισμικού για κινητά διαμορφώθηκε ως εξής:



Παρατηρούμε πως το λογισμικό Android δεν εμφανίζεται πουθενά στο μερίδιο αγοράς μέχρι το τρίτο τρίμηνο του 2008.

Με τον καιρό, όμως, έγινε μεγαλύτερη η ανάγκη για μια νέα mobile πλατφόρμα λογισμικού που θα διατίθεται ελεύθερα στους developers, θα είναι ανεξάρτητη από το διαχειριστή των εφαρμογών και οι εφαρμογές της οποίας δεν θα υπόκεινται σε τόσο αυστηρό έλεγχο (όπως οι εφαρμογές της Apple για το iPhone).

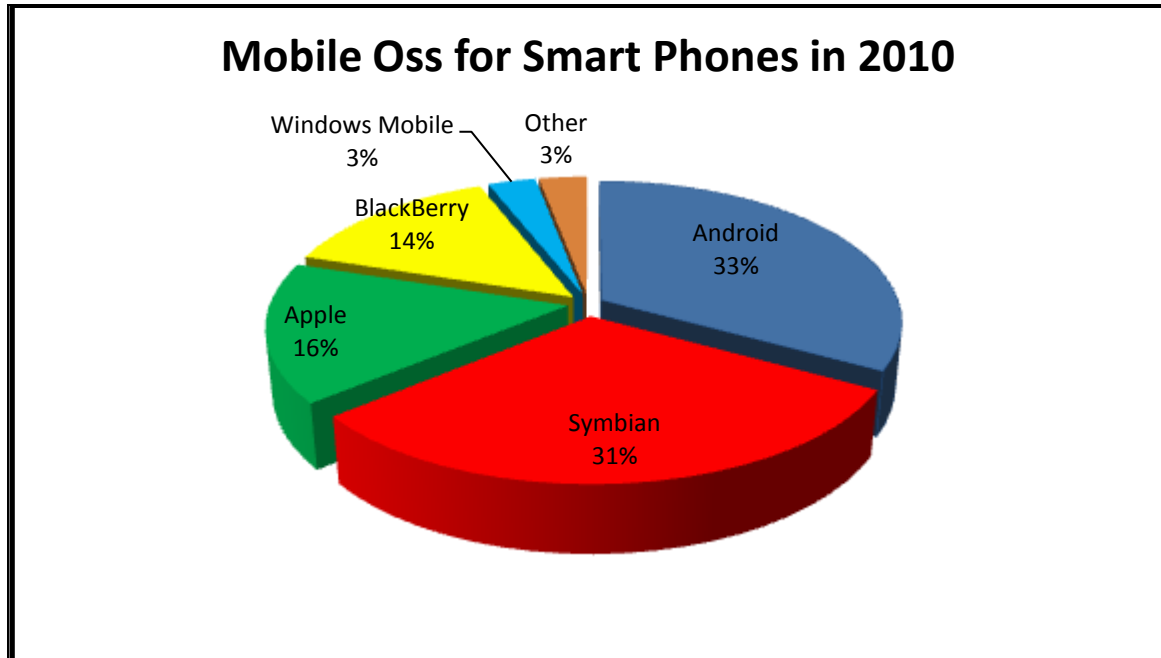
Τέλος, δημιουργήθηκε η ανάγκη οι προσπάθειες για βελτιστοποίηση των εφαρμογών και των δυνατοτήτων της πλατφόρμας να μοιράζονται και οι developers να συνεργάζονται πάνω σε μια κοινή βάση, ώστε να παρέχουν καλύτερα τελικά προϊόντα στον καταναλωτή-χρήστη.

Έτσι η Google το Νοέμβριο του 2007 ανακοινώνει μια platform-based λύση , το Android λογισμικό. 48 τηλεπικοινωνιακοί κολοσσοί μεταξύ των οποίων κατασκευαστές κινητών τερματικών, πάροχοι τηλεπικοινωνιακών υπηρεσιών και σχεδιαστές mobile platforms και εφαρμογών, όπως η Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, NVIDIA συμφωνούν και σχηματίζουν την Open Handset Alliance με στόχο:

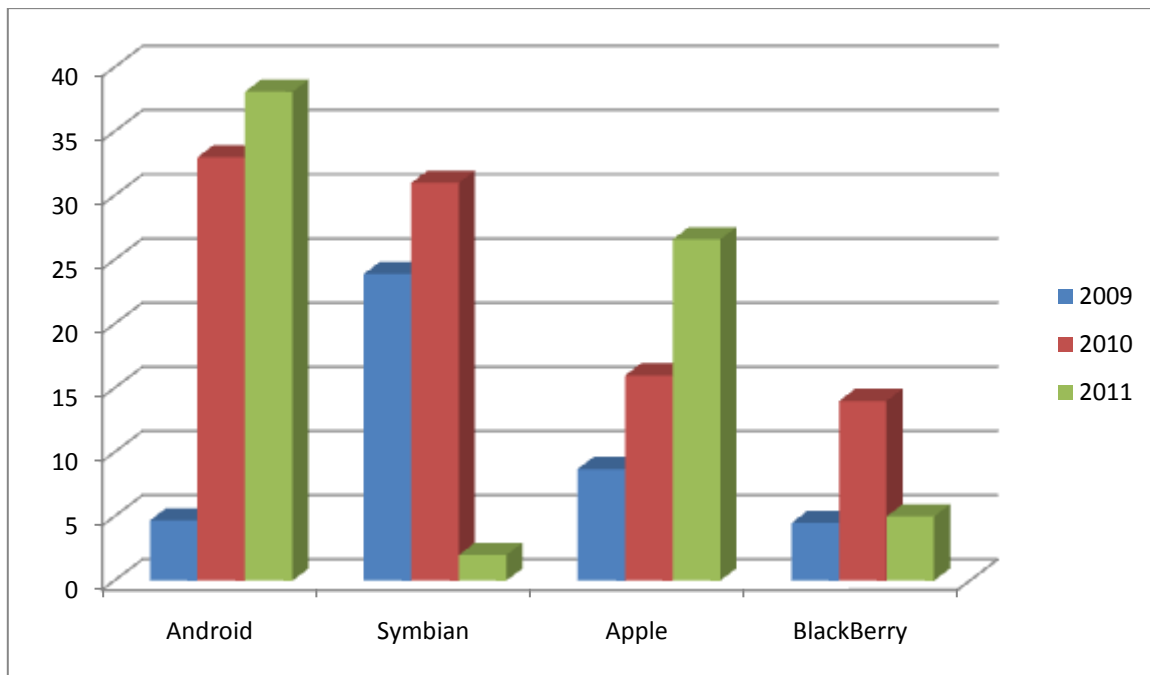
- a) Τη δημιουργία λειτουργικού συστήματος ανοικτού κώδικα(open-source) που θα επιτρέπει την ανάπτυξη πιο φτηνών και απαιτητικών smartphones.
- b) Την ανάπτυξη λογισμικού που θα επιτρέπει σε οποιονδήποτε developer από τον πιο αρχάριο μέχρι τον πιο έμπειρο να αναπτύξει καινοτόμες εφαρμογές και να τις “ανεβάσει” στο Android Store χωρίς καμία παρέμβαση, έλεγχο ή εποπτεία από τους διαχειριστές.
- c) Τη χρήση του Android λογισμικού σε οποιαδήποτε συσκευή και εκτός κινητών τηλεφώνων.
- d) Γενικά την ανάπτυξη ανοικτών προτύπων (open standards) για τις κινητές συσκευές.

Σύμφωνα λοιπόν με πρόσφατες έρευνες το Μάρτιο του 2010 παρατηρήθηκε πως το μερίδιο αγοράς των smartphones με λογισμικό Android έχει αυξηθεί κατακόρυφα, γεγονός που καταδεικνύει τη στροφή των χρηστών καταναλωτών σε κινητά με την τεχνολογία αυτή.

Σύμφωνα με στατιστικές μελέτες το 2010 καταγράφηκαν τα παρακάτω μερίδια αγοράς για τα mobile λειτουργικά συστήματα



Στο παρακάτω διάγραμμα παρουσιάζεται μια εξέλιξη των πωλήσεων των κινητών με διαφορετικά λειτουργικά συστήματα στα έτη 2009-2011.



2.2 Αρχιτεκτονική Android

Το Android είναι ένα πακέτο λογισμικού για κινητές συσκευές που περιλαμβάνει ένα λειτουργικό σύστημα (operating system), middleware και εφαρμογές κορμού (core applications). Το Android SDK παρέχει εργαλεία και APIs (application interfaces), τα οποία είναι απαραίτητα για την ανάπτυξη των android εφαρμογών στην Android πλατφόρμα χρησιμοποιώντας τη γλώσσα προγραμματισμού Java.

Η πλατφόρμα Android είναι μια ανοικτού συστήματος αρχιτεκτονική, με ευέλικτο περιβάλλον ανάπτυξης ενώ παράλληλα υποστηρίζει την επεκτασιμότητα της εμπειρίας χρήστη, με τα βελτιστοποιημένα γραφικά συστήματα που διαθέτει, την πλούσια υποστήριξη πολυμέσων και τον ισχυρό browser που έχει ενσωματωμένο. Επιτρέπει τη χρήση και την αντικατάσταση συστατικών του και υποστήριξη μιας αποτελεσματικής βάσης δεδομένων, ενώ ταυτόχρονα υποστηρίζει ποικίλα μέσα ασύρματων επικοινωνιών. Χρησιμοποιεί επίσης μια εικονική συσκευή, την Dalvik Virtual Machine, που είναι άριστα βελτιστοποιημένη για τις κινητές συσκευές.

Ακόμη, το Android υποστηρίζει λειτουργίες GPS, Videocamera, πυξίδας, τρισδιάστατο επιταχυνσιόμετρο και παρέχει πλούσια APIs για λειτουργίες με χρήση χαρτών και για εφαρμογές βάση θέσης. Οι χρήστες μπορούν εύκολα να έχουν πρόσβαση, να ελέγξουν και να επεξεργαστούν χωρίς χρέωση τους χάρτες Google και να εγκαταστήσουν location based υπηρεσίες στο κινητό τους τηλέφωνο σε πολύ χαμηλό κόστος.

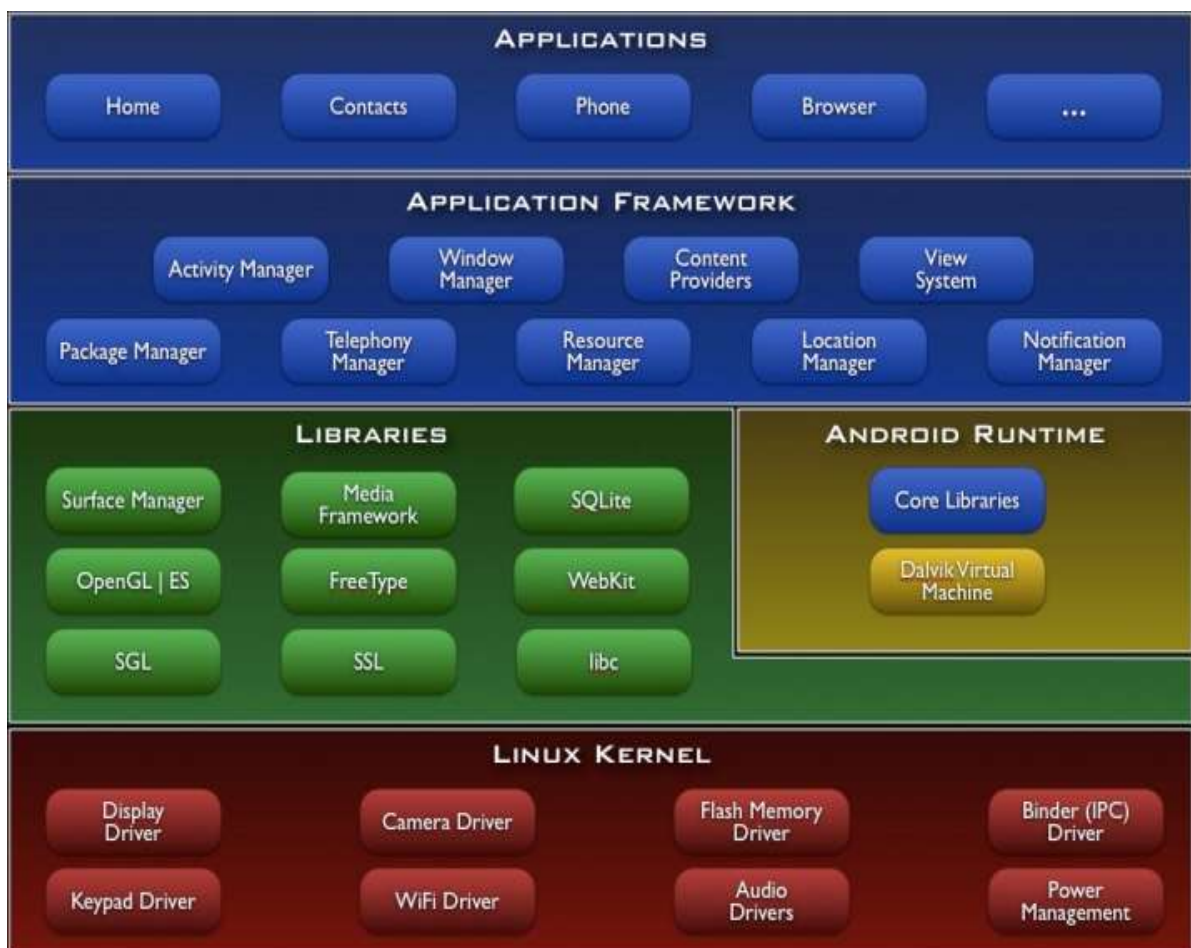
Μέσω του Android λογισμικού και των χαρακτηριστικών του, η Google δεν παρέχει απλά μια νέα τεχνολογία αλλά βοηθά στη μείωση του κόστους ανάπτυξης εφαρμογών και επιτρέπει στους developers να σχεδιάσουν τα mobile συστήματά τους με τα δικά τους μοναδικά χαρακτηριστικά.

Η Google με την εμφάνιση του λογισμικού Android προωθεί και τη βασική της φιλοσοφία και τον πρωταρχικό στόχο δημιουργίας της συγκεκριμένης πλατφόρμας, που δεν είναι άλλη από τη δημιουργία μιας ευρείας βάσης developers που θα συνεργάζονται πάνω στην ίδια πλατφόρμα, χρησιμοποιώντας ανοικτό κώδικα με αποδεδειγμένα λειτουργικές γλώσσες προγραμματισμού, όπως Java, XML και φυσικά θα αναπτύσσουν εφαρμογές ελεύθερα, δωρεάν χωρίς την εποπτεία κάποιου διαχειριστή.

Γενικά, μπορούμε να πούμε πως το Android έγινε γρήγορα ελκυστικό σε τρεις διαφορετικές κατηγορίες ανθρώπων:

- a) Στους developers των εφαρμογών, λόγω των καινοτόμων δυνατοτήτων που τους παρέχει και λόγω της ελευθερίας ανάπτυξης ποικίλων εφαρμογών.
- b) Στους κατασκευαστές, επειδή είναι μια μοναδική πλατφόρμα και δεν χρειάζεται συνεχείς προσαρμογές και μετατροπές
- c) Στους χρήστες-καταναλωτές των κινητών τηλεφώνων, εξαιτίας των πλούσιων χαρακτηριστικών που διαθέτει, της εξελιγμένης τεχνολογίας στην οποία βασίζεται, στο χαμηλότερο κόστος της και στις πιο φιλικές ως προς το χρήστη εφαρμογές της.

Η αρχιτεκτονική του Android και τα κύρια συστατικά της φαίνονται στο παρακάτω σχήμα:



Σχήμα 20. Android architecture

Linux Kernel

Το λογισμικό Android βασίζεται στα Linux (Kernel Version 2.6.x) για τις κύριες λειτουργίες πυρήνα (core system services), όπως διαχείριση μνήμης, επεξεργασία δεδομένων, ασφάλεια, λειτουργίες δικτύου και επιλογή του μοντέλου των χρησιμοποιούμενων οδηγών (drivers). Το Linux Kernel λειτουργεί σαν ένα στρώμα μεταξύ του hardware και της υπόλοιπης λίστας του software.

Android Runtime

Το Android περιλαμβάνει ένα set από βιβλιοθήκες πυρήνα (core libraries) που παρέχουν όλες εκείνες τις λειτουργίες που είναι διαθέσιμες στη γλώσσα προγραμματισμού Java. Τέτοιες βιβλιοθήκες είναι: java.io*, java.util*, java.net*, java.math* κ.α. Κάθε εφαρμογή τρέχει τη δική της ανεξάρτητη διαδικασία/λειτουργία και κατέχει τη δική της θέση στην Dalvik Virtual Machine. Η Dalvik έχει δημιουργηθεί με τέτοιο τρόπο ώστε να μπορούν να λειτουργήσουν αποδοτικά αρκετές εικονικές συσκευές ταυτόχρονα. Η Dalvik Virtual Machine εκτελεί αρχεία τύπου .dex (Dalvik Executable), μια μορφή η οποία παρέχει βελτιστοποίηση στη χρήση της μνήμης και της cpu του κινητού. Η Virtual Machine απαρτίζεται από καταχωρητές και τρέχει κλάσεις οι οποίες μεταγλωττίζονται από Java compiler. Το Linux Kernel μπορεί να εκτελέσει πολλαπλές instances της Dalvik VM, ενώ παράλληλα παρέχει λειτουργικότητα και για δευτερεύουσες εφαρμογές, όπως τα threads και χαμηλού επιπέδου διαχείριση μνήμης.

Libraries

Οι βιβλιοθήκες του Android περιλαμβάνουν ένα set από C/C++ βιβλιοθήκες που χρησιμοποιούνται από διάφορα δομικά στοιχεία του συστήματος. Αυτές διατίθενται στους προγραμματιστές/developers μέσω του Android application framework. Μερικές από αυτές είναι οι παρακάτω:

- a) System C Libraries
- b) Media Libraries: Οι βιβλιοθήκες αυτές υποστηρίζουν δυνατότητες εγγραφής και playback πολλών γνωστών τύπων ήχου, εικόνας και video, όπως MPEG4, H.264, MP3, AAC, AMR, JPEG, PNG.
- c) Surface Manager: Βοηθά στην πρόσβαση στα subsystems της εφαρμογής
- d) WebKit: Μηχανή για μοντέρνο web browser
- e) SGL: Μηχανή για 2D γραφικά
- f) 3D Libraries: Βιβλιοθήκες βασισμένες στα OpenGL ES 1.0 APIs
- g) SQLite: μια ευέλικτη και ισχυρή βάση δεδομένων διαθέσιμη σε όλες τις εφαρμογές.

Application Framework

Οι developers έχουν πρόσβαση στο ίδιο framework από APIs που χρησιμοποιείται και από τις εφαρμογές πυρήνα. Η αρχιτεκτονική των εφαρμογών έχει σχεδιαστεί με τέτοιο τρόπο ώστε να απλοποιεί την επαναχρησιμοποίηση όλων των δομικών συστατικών του Android λογισμικού. Αυτό επίσης επιτρέπει σε κάθε κομμάτι να αντικαθίσταται από τον χρήστη.

Όλες οι εφαρμογές αποτελούνται από ένα set από services και συστήματα, που περιλαμβάνουν μια πλούσια ομάδα από:

- ActivityViews που χρησιμοποιούνται για να “χτίσουν” μια εφαρμογή, όπως grids, lists, textViews, editIntroductionTexts, Spinners, Buttons, έναν ενσωματωμένο web browser ακόμα και MapView που μπορεί να χρησιμοποιηθεί σε κάθε εφαρμογή απλά με λίγες γραμμές κώδικα.
- Content Providers που χρησιμοποιούνται για να επιτρέπουν στις εφαρμογές να έχουν πρόσβαση σε δεδομένα άλλων εφαρμογών (όπως πχ στις επαφές του κινητού τηλεφώνου) ή να μοιράζονται με άλλες εφαρμογές τα δικά τους δεδομένα.
- Έναν αυτόματο Resource Manager που χρειάζεται για να κάνει προσβάσιμες resources εκτός κώδικα μέσα στον κώδικα.
- Notification Manager που επιτρέπει στις εφαρμογές να εμφανίζουν ειδοποιήσεις (alerts) στο πάνω μέρος της οθόνης, πχ notification για την εύρεση δικτύου wi-fi κατά την εκτέλεση της εφαρμογής ή κατά τη λήψη ενός μηνύματος sms.
- Activity Manager που διαχειρίζεται τον κύκλο ζωής των εφαρμογών παρέχοντας χρήσιμες δυνατότητες πλοήγησης σε αυτές.

Applications

Ένα set από εφαρμογές βρίσκονται στο υψηλότερο επίπεδο της αρχιτεκτονικής του Android λογισμικού και περιλαμβάνει email client, SMS/MMS εφαρμογή, ημερολόγιο, έναν web browser, χάρτες και εφαρμογές επί αυτών, επαφές(contacts) κ.α. Όλες οι εφαρμογές είναι γραμμένες σε γλώσσα προγραμματισμού Java.

2.3 Ανατομία μιας Android Εφαρμογής

Υπάρχουν τέσσερα δομικά blocks από τα αποτελείται μια Android εφαρμογή: Activity, Intent Receiver, Service, Content Provider. Δεν χρειάζεται κάθε εφαρμογή να έχει και τα τέσσερα αυτά συστατικά , αλλά σίγουρα χρειάζεται κάποιον συνδυασμό από αυτά. Όταν ο χρήστης/προγραμματιστής αποφασίσει ποια στοιχεία του χρειάζονται για να αναπτύξει την εφαρμογή , αυτά πρέπει να καθοριστούν και να καταγραφούν σε ένα αρχείο που ονομάζεται AndroidManifest.xml. Στο αρχείο αυτό, δηλώνονται τα δομικά blocks που θα χρησιμοποιηθούν και ποιες δυνατότητες και προδιαγραφές θα εξυπηρετούν.

Activity

Η Activity είναι το πιο σύνηθες από τα τέσσερα άλλα δομικά στοιχεία. Μια activity είναι συνήθως το τι εμφανίζεται στην οθόνη όταν τρέχει μια εφαρμογή. Κάθε activity είναι μια ξεχωριστή κλάση που κάνει extend την βασική κλάση Activity του Android. Η κλάση αυτή εμφανίζει στην οθόνη ένα interface χρήστη που συντίθεται από Views και απαντά σε διάφορα γεγονότα-events. Για παράδειγμα, μια εφαρμογή γραπτών μηνυμάτων μπορεί να έχει μια οθόνη που να δείχνει μια λίστα με τις επαφές στις οποίες μπορεί να σταλεί το συγκεκριμένο μήνυμα, μια δεύτερη οθόνη για εγγραφή του μηνύματος στη δεδομένη επαφή και άλλες οθόνες που να κάνουν ρυθμίσεις των μηνυμάτων ή να δείχνουν το ιστορικό αποστολής μηνυμάτων. Κάθε μία από τις οθόνες αυτές θα υλοποιούνταν ως μια activity. Η εναλλαγή μεταξύ των διαφορετικών “οθονών” ισοδυναμεί με έναρξη μιας καινούριας

activity. Σε μερικές περιπτώσεις μια Activity μπορεί να επιστρέψει μια τιμή σε μια προηγούμενη activity.

Όταν μια νέα οθόνη ανοίγει, η προηγούμενη μπαίνει σε αναμονή λειτουργίας και τοποθετείται σε στοίβα με τις πρόσφατα ανολοκλήρωτες activities. Ο χρήστης μπορεί να πλοηγηθεί προς τα πίσω και να δει τα προηγούμενα ανοιχτά screens. Τα screens αυτά μπορεί επίσης να επιλεγεί να διαγραφούν από τη στοίβα ιστορικού όταν καταλαμβάνουν χώρο και δεν χρειάζεται να μένουν εκεί.

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

Υπάρχει επίσης μια σχετική κλάση που ονομάζεται IntentFilter. Αν το intent είναι μία αίτηση για να γίνει κάτι, το intent filter περιγράφει ποια και πόσα τέτοια intents μπορεί να χειριστεί μια activity

Intent Receiver

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

Services

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

Content Providers

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

2.4 Android και Location Based Services

Η συμβολή του λογισμικού Android στην ανάπτυξη των LBS είναι πολύ σημαντική καθώς μέχρι πριν, η mobile χαρτογράφηση και η ανάπτυξη εφαρμογών βάση θέσης ήταν πολύπλοκη, δύσκολη και απαιτούσε συχνά πληρωμή αρκετά υψηλών συνδρομών στους παρόχους. Με το Android λογισμικό –όπως θα δείξουμε και παρακάτω- οι δυνατότητες ανάκτησης της θέσης είναι αρκετές , καθώς εκμεταλλεύεται όλες τις τεχνολογίες εντοπισμού θέσης, ενώ παράλληλα παρέχει πακέτα-βιβλιοθήκες στους developers για την ανάπτυξη τέτοιου είδους εφαρμογών.

Συγκεκριμένα, το λογισμικό Android παρέχει στους προγραμματιστές τη βιβλιοθήκη/package `com.google.android.maps` η οποία δίνει τη δυνατότητα σε κάθε εφαρμογή να δείχνει στο γραφικό της περιβάλλον-οθόνη, έναν χάρτη με μία ευρεία γκάμα εστίασης. Το πλεονέκτημα αυτού του πακέτου είναι το γεγονός πως ο χρήστης δε χρειάζεται να ενεργοποιήσει τον browser ή κάποιο άλλο πρόγραμμα του κινητού προκειμένου να έχει πρόσβαση στους χάρτες, αλλά απευθείας μπορεί να δει στην εφαρμογή του το χάρτη αρκεί βέβαια να έχει πρόσβαση στο διαδίκτυο για να μπορέσει να κατεβάσει το τμήμα του χάρτη που θέλει. Επίσης, μπορεί να μετακινήσει το χάρτη προς οποιαδήποτε κατεύθυνση ή να αλλάξει επίπεδο εστίασης προκειμένου να έχει μια πιο εποπτική εικόνα.

Επίσης, το Android προσφέρει το package `android.location` το οποίο δίνει στον προγραμματιστή τη δυνατότητα να χρησιμοποιήσει στις εφαρμογές του το δέκτη GPS , το σύστημα του παρόχου για απόκτηση πληροφοριών θέσης από τις κεραίες κινητής τηλεφωνίας και από το ασύρματο δίκτυο στο οποίο είναι συνδεδεμένο το κινητό με τη βοήθεια της Google για την απόκτηση μιας περιοχής σύγκλισης για το που βρίσκεται ο χρήστης. Με το πακέτο αυτό ο προγραμματιστής έχει τη δυνατότητα να ενημερώνει την εφαρμογή του για το που βρίσκεται και αν απαιτείται, να προβάλλει τη θέση του με το `com.google.android.maps` πακέτο όπως αναφέρθηκε.

Οι ευκολίες και οι ποικίλες δυνατότητες που παρέχει το λογισμικό Android στην ανάπτυξη location based εφαρμογών έχουν κάνει τον τομέα των υπηρεσιών αυτών τη νέα μόδα στο χώρο του social networking. Μέχρι τώρα, οι υλοποιήσεις LBS καλύπτουν την ανάγκη για πλοήγηση από ένα σημείο σε ένα άλλο , οδηγώντας ουσιαστικά το χρήστη να ακολουθήσει τη διαδρομή που του προτείνουν. Μια άλλη κατηγορία είναι οι εφαρμογές εκείνες που στόχο έχουν την ενημέρωση του χρήστη για τον καιρό ή πρόβλεψη για την αυριανή μέρα , ανάλογα με την τοποθεσία που βρίσκεται.

Τέλος, μια άλλη κατηγορία εφαρμογών είναι αυτές που ενημερώνουν τον χρήστη για πιθανά σημεία ενδιαφέροντος (POIs-points of interest) κοντά στην περιοχή που βρίσκεται, όπως μουσεία, εστιατόρια, βενζινάδικα κτλ.

Παρακάτω κάνουμε μια αναφορά στις πέντε top location-based services:

a) Foursquare



Το Foursquare είναι μια κινητή location-based πλατφόρμα που κάνει τις πόλεις πιο εύκολες και πιο ενδιαφέρουσες σε κάποιον για να τις εξερευνήσει. Με το “check-in” από το smartphone, οι χρήστες

μοιράζονται την πληροφορία της θέσης τους με φίλους ενώ παράλληλα συλλέγουν πόντους ή εικονικά χρήματα. Η εφαρμογή αυτή καθοδηγεί και προσανατολίζει τους χρήστες, επιτρέποντάς τους να συλλέγουν και να ταξινομούν πληροφορίες σχετικά με διάφορους τόπους συναντήσεων που θέλουν να επισκεφτούν και να βρίσκουν σχετικές προτάσεις για κοντινές τοποθεσίες. Από την πλευρά τους οι έμποροι και οι εταιρείες χρησιμοποιούν την πλατφόρμα foursquare για να πληροφορήσουν το αγοραστικό κοινό για τις υπηρεσίες που προσφέρουν και να προσελκύσουν όσο το δυνατόν περισσότερους πελάτες.

Η απήχηση της συγκεκριμένης εφαρμογής είναι τόσο μεγάλη που απαριθμεί πάνω από δέκα εκατομμύρια χρήστες παγκοσμίως, με 3 εκατομμύρια check-ins καθημερινώς και 750 εκατομμύρια check-ins συνολικά, ενώ πάνω από 500,000 επιχειρήσεις χρησιμοποιούν την Merchant πλατφόρμα ώστε να διαφημιστούν και να προβάλλουν τις διαθέσιμες υπηρεσίες/προϊόντα τους.

b) Facebook Places

Το facebook ανακοίνωσε την εφαρμογή αυτή τον Αύγουστο 2008. Είναι μια εφαρμογή που επιτρέπει στους χρήστες να κάνουν check-in στο facebook, χρησιμοποιώντας το κινητό τους τηλέφωνο. Με αυτό τον τρόπο, μπορούν να ενημερώσουν τους φίλους στο facebook πού βρίσκονται οποιαδήποτε στιγμή της ημέρας. Σαν μια επέκταση-δυνατότητα του ισχυρότερου site κοινωνικής δικτύωσης, κάνει το facebook μια από τις πιο ισχυρές υπηρεσίες Location Sharing.



c) Loopt



Το Loopt είναι ένας από τους καλύτερους τρόπους για να μπορεί ένας χρήστης να έχει πρόσβαση σε reviews και να μοιραστεί τις δικές του απόψεις για μια τοποθεσία. Προσφέρει απαντήσεις πραγματικού χρόνου σε κοινές ερωτήσεις, όπως: Είναι μακριά το συγκεκριμένο

κατάστημα; Έχει η συγκεκριμένη περιοχή Wi-Fi; Ποιο είναι το καλύτερο special hour μενού στην περιοχή; Το Loopt βελτιώνει την εμπειρία χρήστη στον πραγματικό κόσμο. Ειδοποιεί τους χρήστες, αν υπάρχουν συμφέρουσες προσφορές στην περιοχή που βρίσκεται ώστε να εξοικονομήσει χρήματα ή να κάτι προσφέρεται δωρεάν.

Οι συνεργαζόμενοι πάροχοι με το Loopt έχουν προσφέρει κατά καιρούς δωρεάν εισιτήρια για συναυλίες, αεροπορικά ταξίδια, δωρεάν γεύματα σε εστιατόρια κ.α.

Επίσης, αν ο χρήστης δεν βρίσκει τις προσφορές που θέλει το Loopt της υπηρεσίας u-Deals, του δίνει τη δυνατότητα να ζητήσει τι θέλει και η αίτηση του να καταγραφεί. Ακόμη δίνει τη δυνατότητα στους mobile χρήστες να βρίσκουν τους φίλους τους, να ειδοποιούν τους συνεργάτες τους σε περίπτωση που αργούν στο ραντεβού τους προβάλλοντάς τους την κατάστασή τους(σε πέντε λεπτά είμαι εκεί, κόλλησα στην κίνηση) μαζί με την τρέχουσα θέση τους.

d) SCVNGR

Είναι μια location-based πλατφόρμα παιχνιδιού με περιεχόμενο κοινωνικής δικτύωσης. Το SCVNGR είναι ένα παιχνίδι. Ο χρήστης πηγαίνει σε διάφορες περιοχές, ανακαλύπτει νέες, μοιράζεται τέτοιου είδους εμπειρίες, συμμετέχει σε “αποστολές”, μαζεύει πόντους, ξεκλειδώνει δώρα είτε εικονικά είτε και πραγματικά. Η εφαρμογή αυτή είναι κάτι παραπάνω από ένα απλό παιχνίδι, είναι ένα κοινωνικό mobile παιχνίδι που διαμορφώνεται και εξελίσσεται στην καθημερινή ζωή των χρηστών. Οποιοσδήποτε μπορεί να συμμετέχει και να χτίσει την εφαρμογή αυτή περισσότερο, από μεμονωμένους χρήστες μέχρι τοπικά καταστήματα, μουσεία, εστιατόρια, ιστορικά μνημεία. Με λίγα λόγια, η SCVNGR είναι μια εφαρμογή που δημιουργεί ένα game layer πάνω από την κοινωνία.



e) Yelp



Η εφαρμογή Yelp είναι συντομογραφία των **Yellow Pages**. Παρέχει online τοπικές δυνατότητες αναζήτησης επιχειρήσεων για τους επισκέπτες της περιοχής. Μια τυπική αναζήτηση περιλαμβάνει το τι ψάχνει ο χρήστης της εφαρμογής και την τοποθεσία στην οποία θέλει να γίνει αυτή η αναζήτηση. Η επιθυμητή τοποθεσία εισάγεται με τη μορφή δεδομένης διεύθυνσης ή μιας ευρύτερης περιοχής γύρω από ένα συγκεκριμένο σημείο. Κάθε αποτέλεσμα αναζήτησης επιστρέφεται με δυνατότητα αξιολόγησης πέντε σημείων από τους επισκέπτες καθώς και με κριτικές, σχόλια αυτών και περισσότερες πληροφορίες σχετικά με τις προσφερόμενες υπηρεσίες. Οι επισκέπτες του site μπορούν να το κρατούν συνεχώς ενημερωμένο, με την έγκριση φυσικά του διαχειριστή, ενώ ταυτόχρονα οι ιδιοκτήτες των επιχειρήσεων μπορούν να ενημερώνουν συνεχώς το προφίλ των επιχειρήσεών τους.

Κεφάλαιο 3

Παρουσίαση Εφαρμογής

3.1 Αντικείμενο της Διπλωματικής

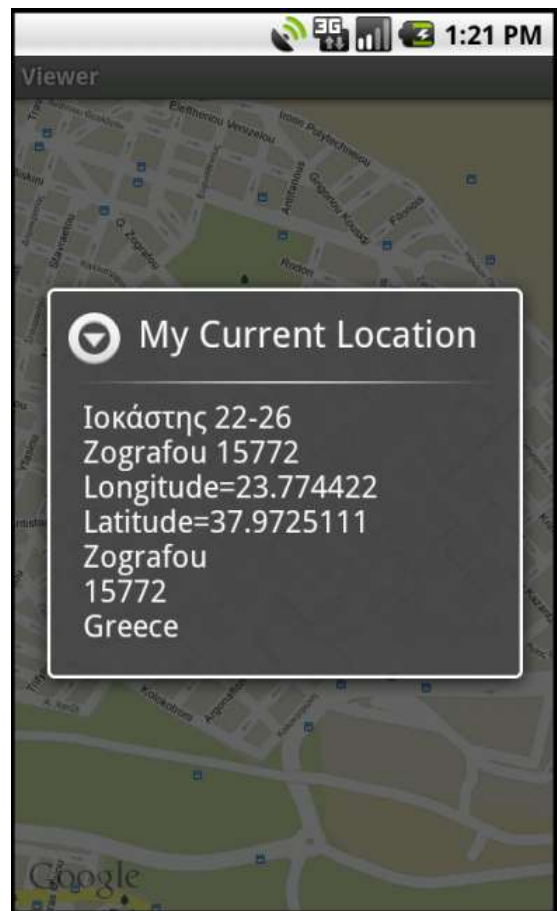
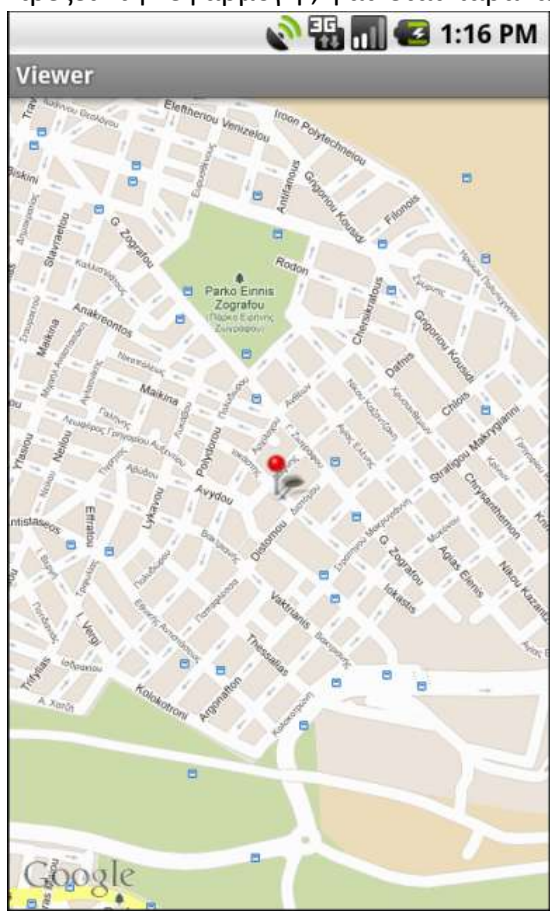
Στην παρούσα διπλωματική αναπτύχθηκε μια εφαρμογή που επιτρέπει στους χρήστες κινητών τερματικών με λειτουργικό Android , ανάλογα με την θέση που βρίσκονται να έχουν πρόσβαση σε προσφορές καταστημάτων διαφόρων τύπων, να παίρνουν πληροφορίες γι' αυτές-για τη διάρκεια των προσφορών, το είδος τους, την αξιολόγηση τους από τους μέχρι εκείνη τη στιγμή χρήστες(ranking),για τη θέση των καταστημάτων κ.α.- και τελικά να αξιολογούν και οι ίδιοι την ποιότητα της προσφερόμενης υπηρεσίας.

Στην πλευρά, λοιπόν, του client-χρήστη δημιουργήσαμε ένα περιβάλλον αλληλεπίδρασης στο κινητό που του δίνει τη δυνατότητα να επιλέξει παραμέτρους αναζήτησης διαφορετικού τύπου προσφορών που ενδέχεται να έχει ανάγκη, να αποφασίσει για το πόσο μακριά θέλει να βρίσκονται τα καταστήματα που προσφέρουν τις προσφορές αυτές, να αναζητήσει προσφορές γύρω από την τρέχουσα θέση του ή γύρω από μια περιοχή που πρόκειται να επισκεφτεί και τέλος να περιηγηθεί σ' αυτές είτε πάνω στον χάρτη , είτε στην προβαλλόμενη λίστα, ανάλογα πάντα με τις προτιμήσεις του.

Στην πλευρά του server, δημιουργήσαμε ένα website όπου οι επιχειρηματίες μπορούν να καταχωρούν τις προσφορές τους σε μια βάση δεδομένων η οποία ενημερώνει αυτόματα τους χρήστες των κινητών.

3.2 Η εφαρμογή VIEWER

Σε αυτή την ενότητα θα παρουσιαστεί η εφαρμογή Viewer και θα αναφερθούν οι επιμέρους δυνατότητές της. Αρχικά, η κεντρική οθόνη της εφαρμογής που βλέπει ο χρήστης, όταν “τρέξει” την εφαρμογή , φαίνεται παρακάτω.



Στην πρώτη οθόνη φαίνεται η τρέχουσα θέση του χρήστη, όπως λαμβάνεται από τον καλύτερο location provider που βρίσκει η συσκευή και ανταποκρίνεται στα κριτήρια αναζήτησης που έχουν τεθεί από τον developer της εφαρμογής.

Συγκεκριμένα, ανάλογα με το κινητό τερματικό, υπάρχουν διάφορες τεχνολογίες που χρησιμοποιεί το Android για να προσδιορίσει την τρέχουσα θέση του χρήστη. Κάθε τεχνολογία ή κάθε Location Provider, προσφέρει διαφορετικές δυνατότητες όσον αφορά την κατανάλωση ενέργειας που χρειάζεται για να λειτουργήσει, την ακρίβεια και την ταχύτητα προσδιορισμού της θέσης και το κόστος σε υπολογιστικούς πόρους που δεσμεύει(resources).

Μέσα στον κώδικα, λοιπόν, η επιλογή του καλύτερου δυνατού Location Provider βάση και των κριτηρίων που θέλουμε να ικανοποιεί και η εύρεση της τρέχουσας θέσης του κινητού τερματικού υλοποιείται ως εξής:

```
LocationManager locationManager;  
String context = Context.LOCATION_SERVICE;  
locationManager = (LocationManager) getSystemService(context);  
  
Location location=locationManager.getLastKnownLocation(provider);  
  
Criteria criteria = new Criteria();  
criteria.setAccuracy(Criteria.ACCURACY_FINE);  
criteria.setAltitudeRequired(false);  
criteria.setBearingRequired(false);  
criteria.setCostAllowed(true);  
criteria.setPowerRequirement(Criteria.POWER_LOW);  
  
String provider = locationManager.getBestProvider(criteria, true);
```

Παρατηρούμε πως τίθενται κριτήρια που αφορούν το είδος της ακρίβειας των αποτελεσμάτων που θα επιστραφούν(fine ή coarse accuracy), την ενέργεια που θέλουμε να καταναλωθεί(low, medium, high) , το οικονομικό κόστος(cost allowed) και τη δυνατότητα επιστροφής τιμών για το υψόμετρο, ταχύτητα και bearing.

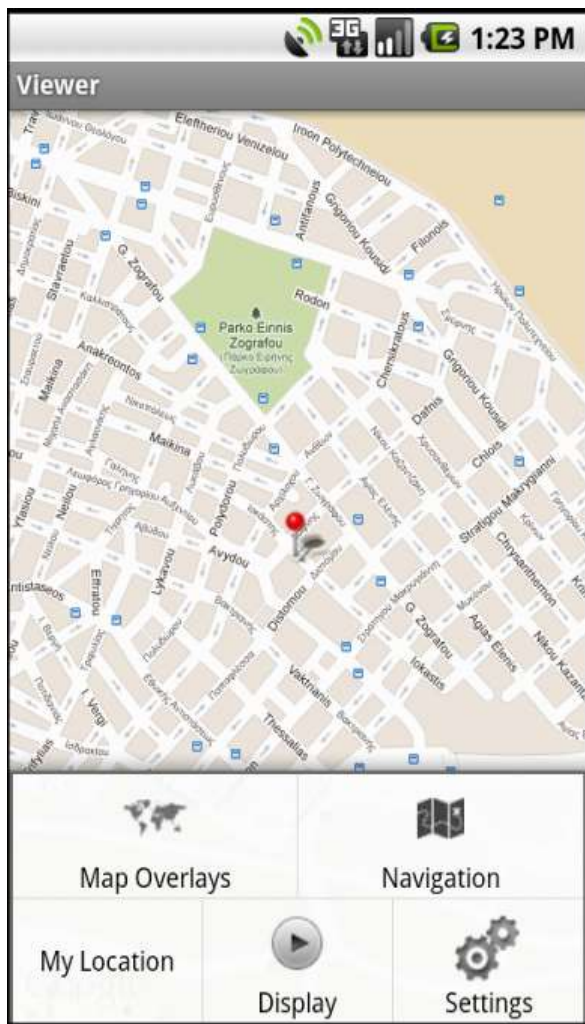
Τέλος μετά τον καθορισμό των προδιαγραφών-κριτηρίων, χρησιμοποιούμε τη μέθοδο `getBestProvider`, η οποία επιστρέφει τον Location Provider εκείνο που ταιριάζει καλύτερα σ' αυτά που ζητούμε. Εάν περισσότεροι από ένας providers είναι διαθέσιμοι, τότε αυτός με την μεγαλύτερη ακρίβεια επιστρέφεται. Εάν δεν υπάρχει provider που να ανταποκρίνεται στις προδιαγραφές, τότε τα κριτήρια γίνονται ελαστικά, με την ακόλουθη σειρά μέχρι να βρεθεί κάποιος provider:

1. Power use
2. Accuracy
3. Ability to return bearing, altitude and speed

Επίσης, αξίζει να σημειώσουμε πως η πρόσβαση στις location-based υπηρεσίες χειρίζεται από το Location Manager System service. Για να έχουμε πρόσβαση στον Location Manager, ζητούμε στην αρχή του κώδικα μια instance της LOCATION_SERVICE χρησιμοποιώντας τη μέθοδο `getSystemService`.

Στη δεύτερη οθόνη φαίνονται κάποια στοιχεία της τρέχουσας θέσης, όπως μία κεντρική διεύθυνση , γεωγραφικό πλάτος και μήκος, ταχυδρομικός κώδικας της περιοχής, το όνομα της περιοχής , η χώρα κτλ , τα οποία γίνονται διαθέσιμα όταν ο χρήστης πατήσει πάνω στο κόκκινο pin.

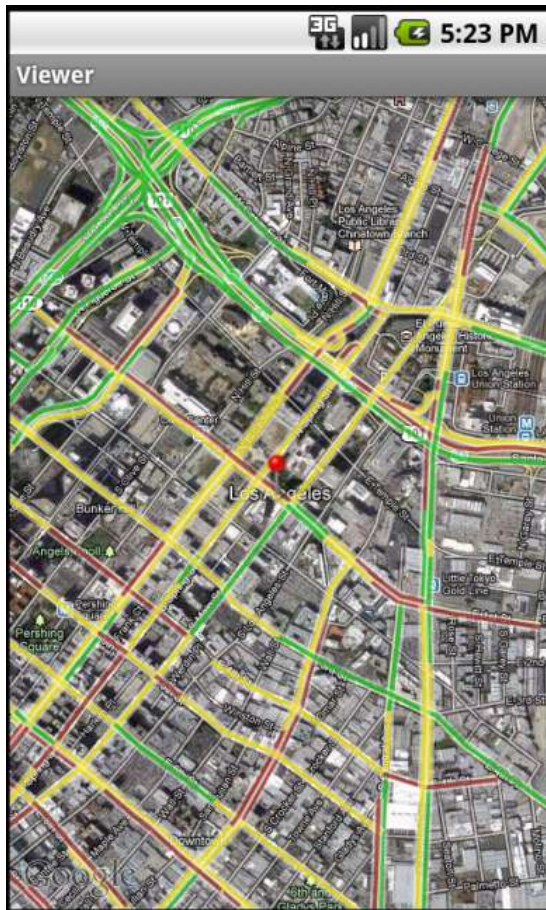
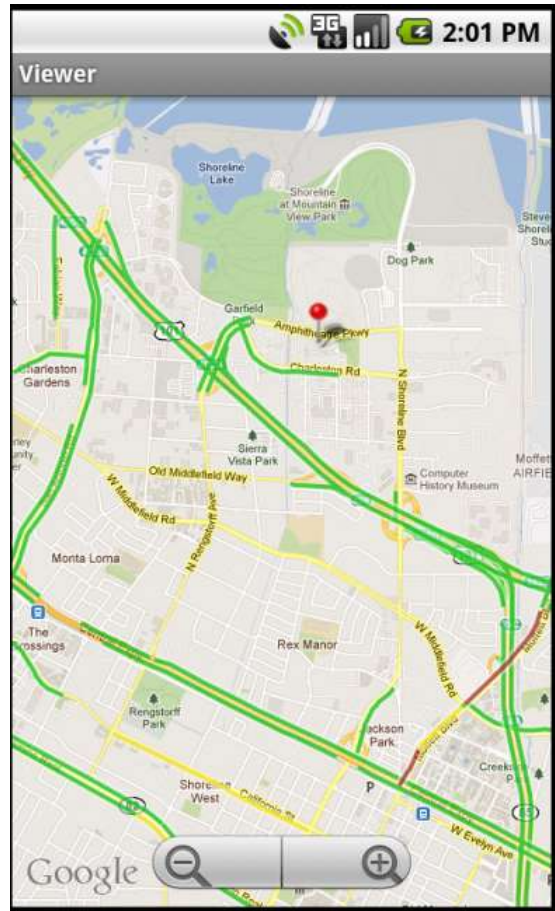
Μόλις ο χρήστης πατήσει το πλήκτρο “menu” του κινητού, εμφανίζεται το κεντρικό μενού της εφαρμογής, όπως φαίνεται παρακάτω.



Πατώντας το πλήκτρο Map Overlays, ο χρήστης έχει τη δυνατότητα να επιλέξει διαφορετικούς τρόπους εμφάνισης του χάρτη. Οι τρεις διαθέσιμοι τρόποι είναι σε μορφή χάρτη (default εμφάνιση), σε μορφή satellite-όπου έχουμε την εικόνα γεωγραφικού ανάγλυφου και σε μορφή traffic όπου επάνω στο χάρτη δίνεται μια εικόνα της κίνησης στους δρόμους (με διαφορετικά χρώματα φαίνεται που έχει περισσότερη κίνηση και που λιγότερη) σε όποιες βέβαια περιοχές υπάρχει η διαθέσιμη αυτή πληροφορία.

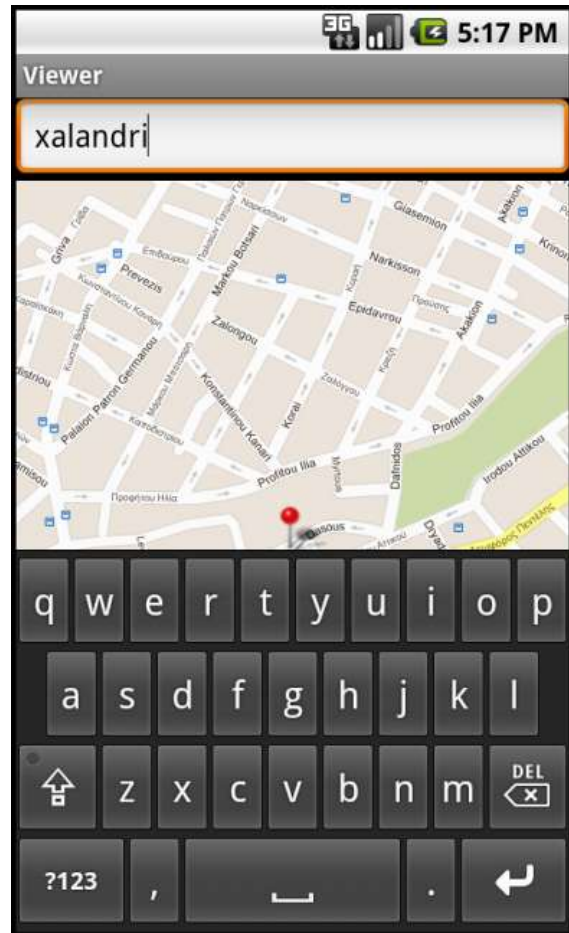
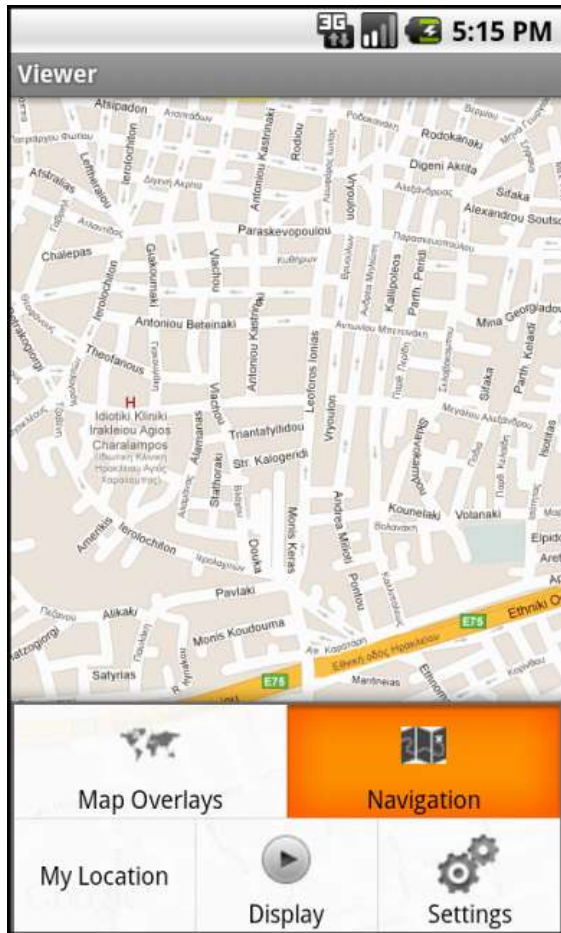
Επιλέγει είτε satellite είτε traffic είτε και τα δύο και πατώντας apply έχει την εικόνα του χάρτη που επιθυμεί.

Στα παρακάτω screenshots φαίνονται τα διαφορετικά map overlays που μπορεί να χρησιμοποιήσει ο χρήστης.

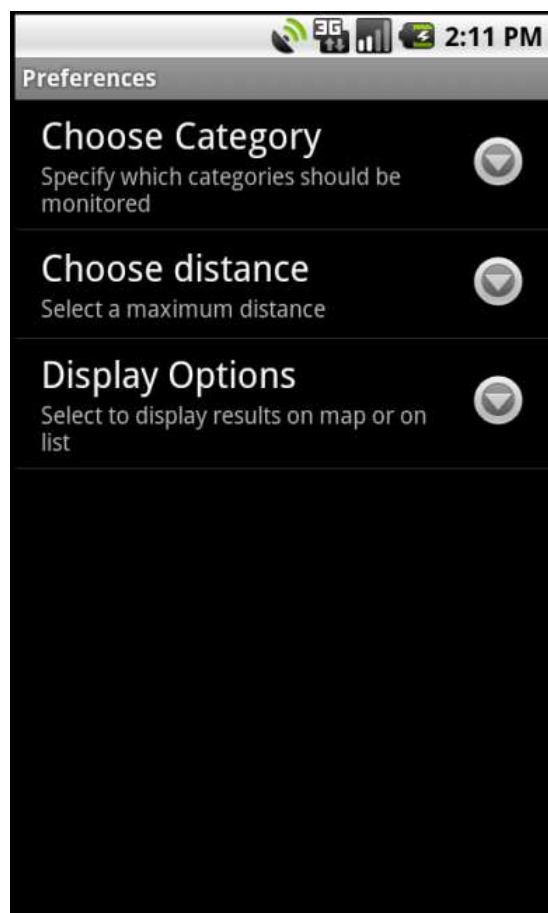
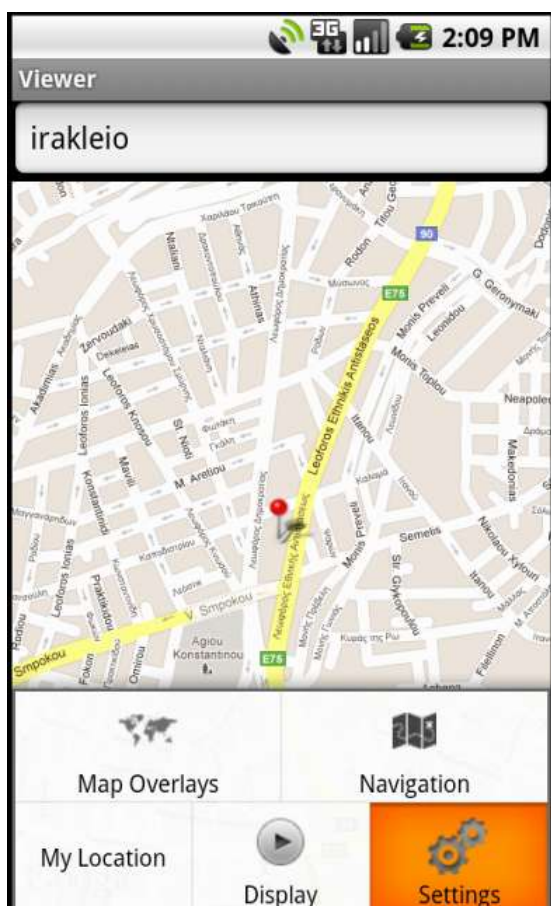


Πατώντας ο χρήστης το πλήκτρο navigation μπορεί να μεταβεί σε οποιαδήποτε περιοχή επιθυμεί γράφοντας στο edit box το όνομα της τοποθεσίας που τον ενδιαφέρει. Το edit box λειτουργεί ακριβώς όπως το αντίστοιχο πεδίο του Google maps και επιστρέφει ανάλογα αποτελέσματα.

Στις παρακάτω εικόνες φαίνεται η υλοποίηση της παραπάνω δυνατότητας της εφαρμογής.



Επιλέγοντας το πλήκτρο Settings , ο χρήστης μεταβαίνει στις ρυθμίσεις όπου καθορίζει τις επιλογές/προτιμήσεις αναζήτησης που θέλει.



Συγκεκριμένα, υπάρχουν τρεις επιλογές και για κάθε μια από αυτές υπάρχει ένδειξη για το τι ακριβώς ρυθμίζει. Διακρίνονται σε τρεις κατηγορίες:

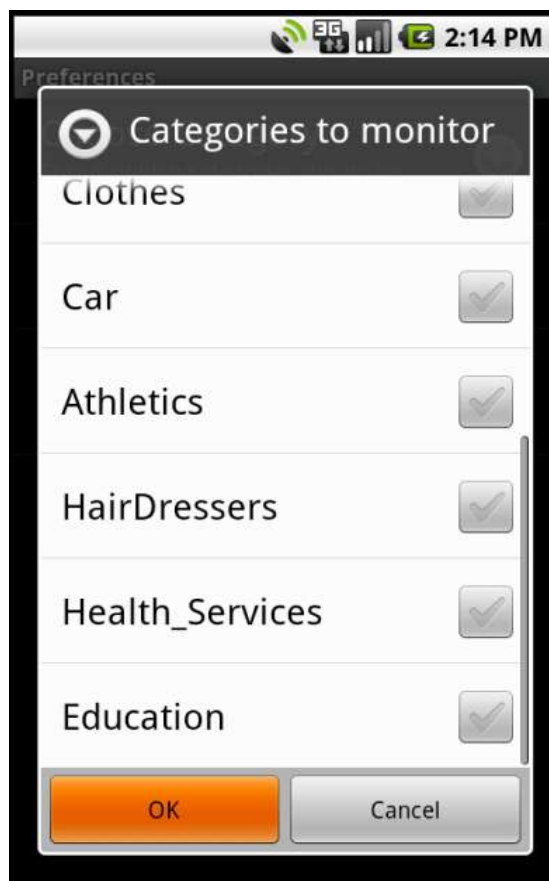
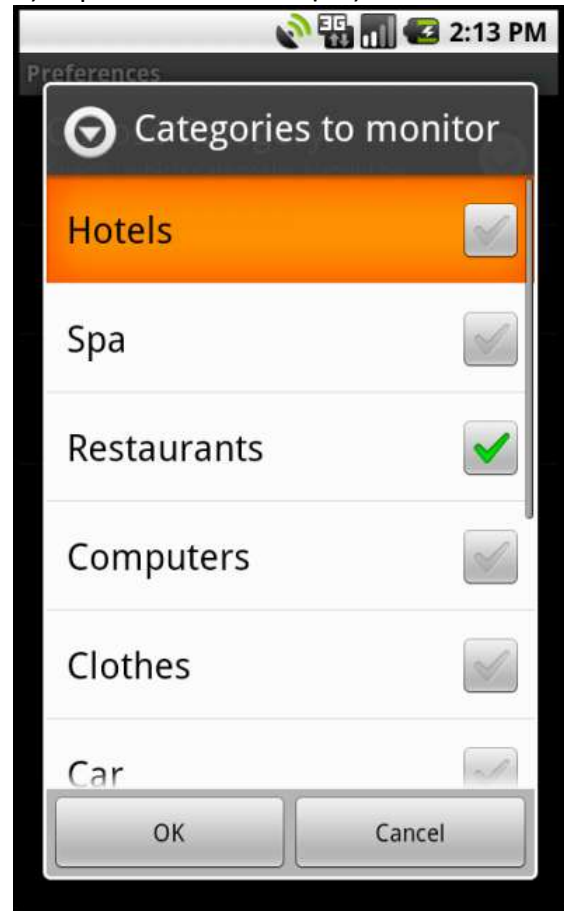
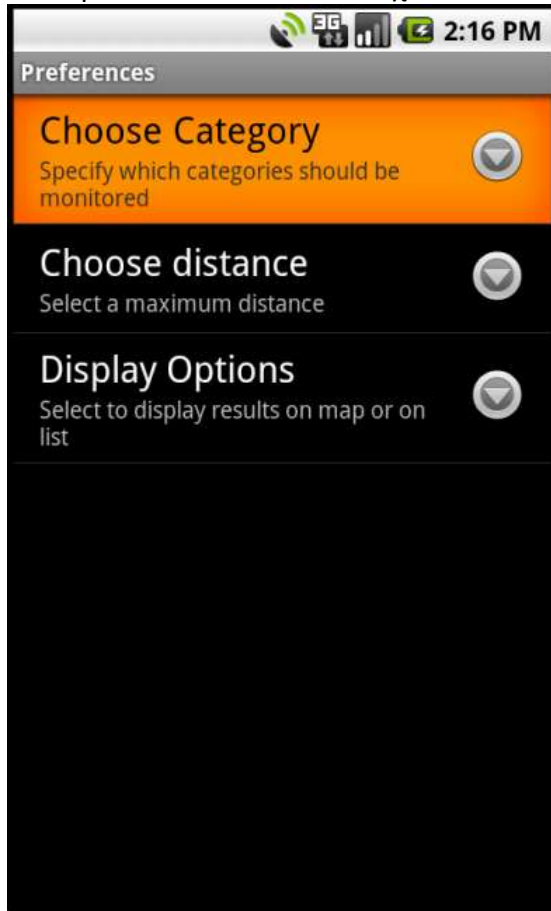
Choose Category: Ο χρήστης επιλέγει από μια λίστα με checkboxes τις κατηγορίες εκείνες των καταστημάτων που τον ενδιαφέρουν (restaurants, hotels, spa, computers, clothes, car, education, athletics, hairdressers, health services) και τις προσφορές που διατίθενται.

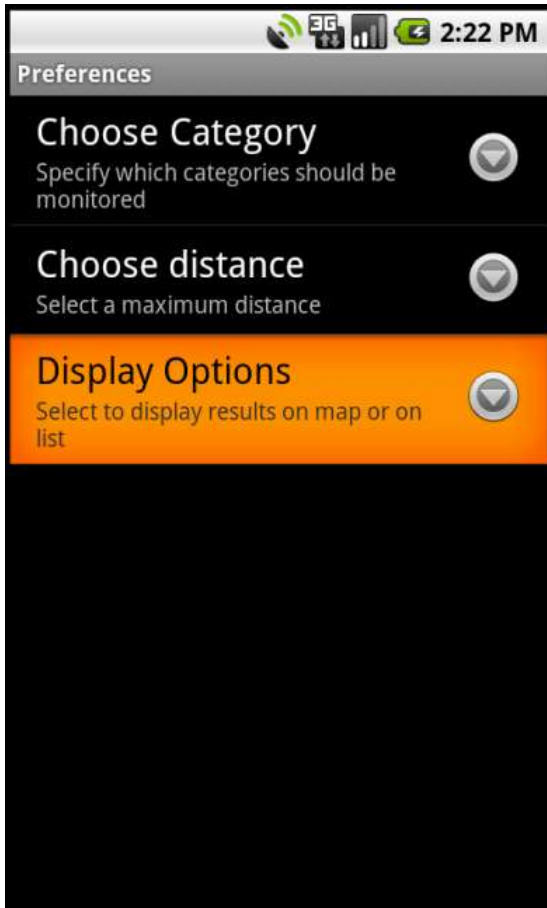
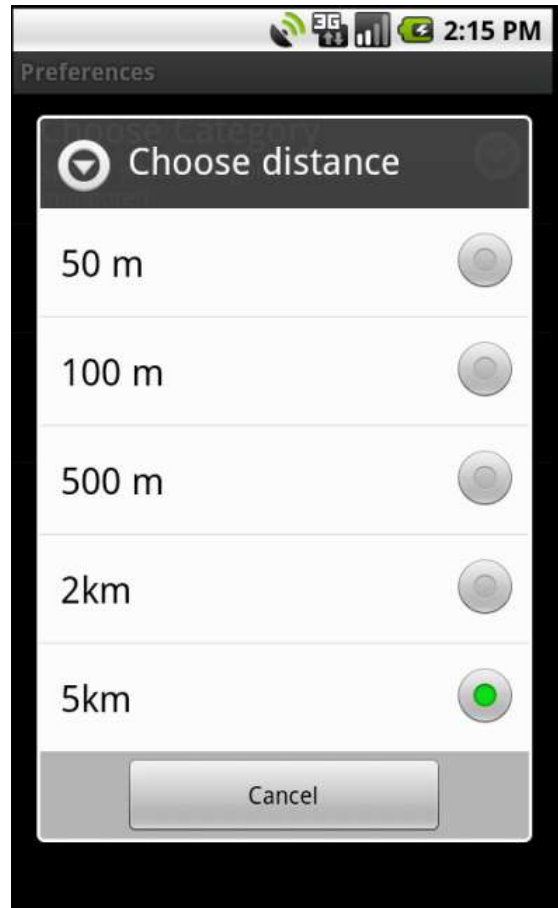
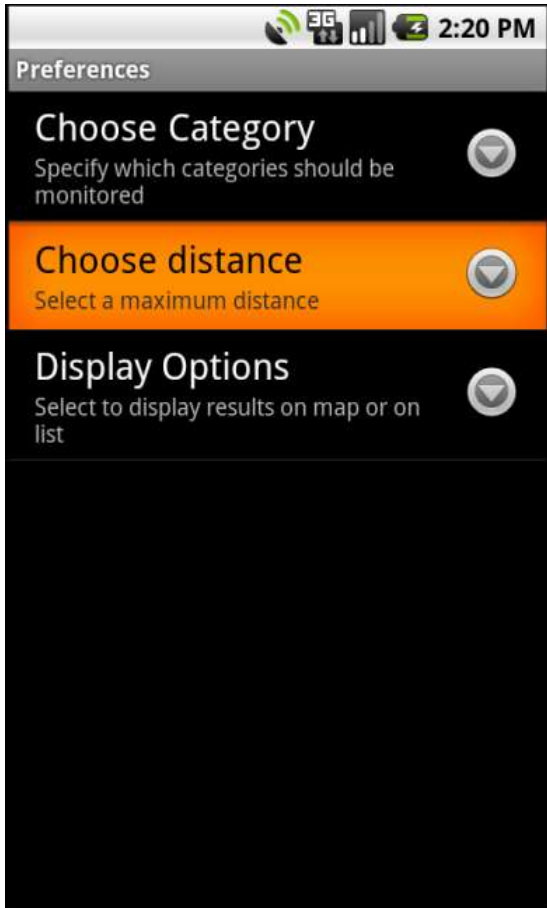
Choose Distance: Ο χρήστης μπορεί να καθορίσει την απόσταση στην οποία επιθυμεί να γίνει η αναζήτηση για τις προσφορές των παραπάνω καταστημάτων με τη χρήση των radiobuttons (50m, 100m, 500m, 2km, 5km). Η απόσταση αυτή δηλώνει πως η αναζήτηση που θα γίνει από την εφαρμογή θα εμφανίσει αποτελέσματα εντός μιας περιοχής με κέντρο την τρέχουσα θέση του χρήστη ή το σημείο που έχει μεταβεί με χρήση του navigation, και ακτίνα την απόσταση που έχει δηλώσει.

Display Options: Ο χρήστης έχει τη δυνατότητα να επιλέξει τον τρόπο που θέλει να του προβληθούν τα αποτελέσματα της αναζήτησης. Οι επιλογές του είναι δύο:

- a) Σε λίστα
- β) Σε χάρτη

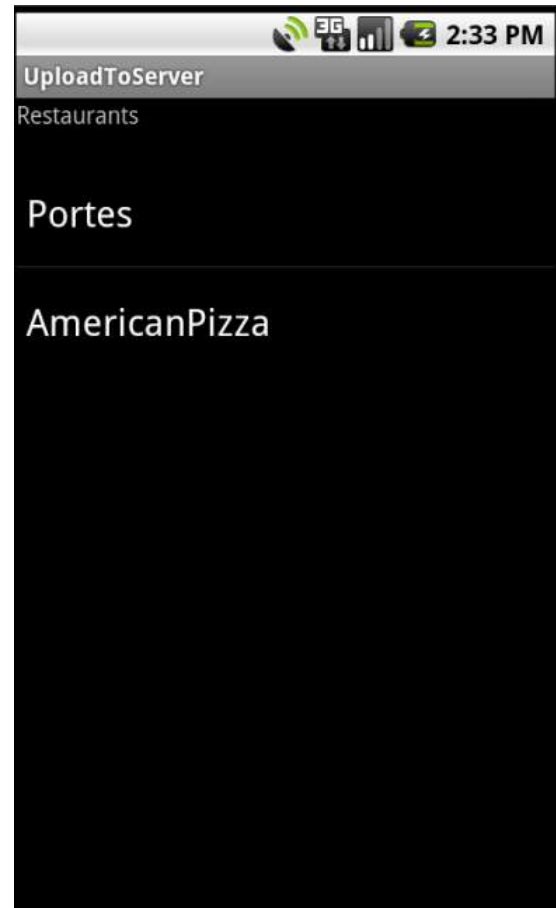
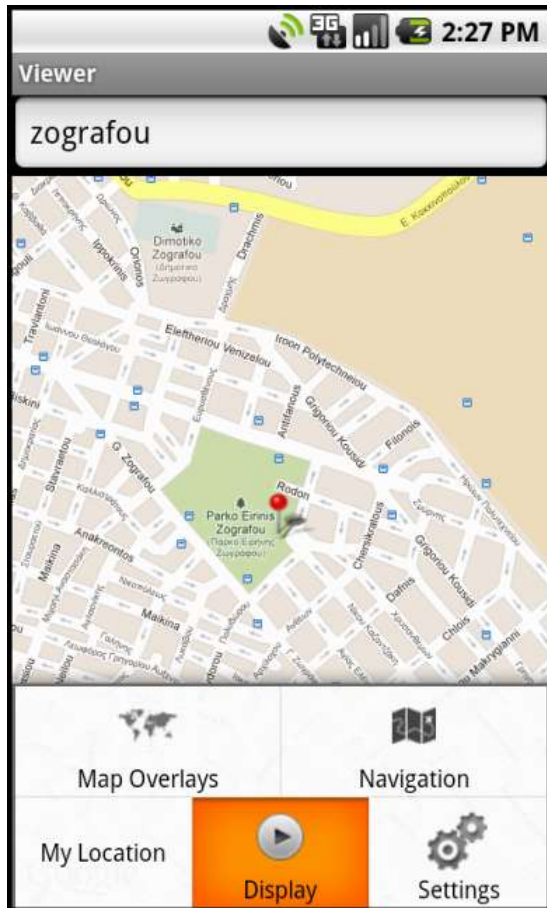
Τα παρακάτω screenshots δείχνουν αναλυτικά τις παραπάνω δυνατότητες.





Έχοντας ρυθμίσει κατάλληλα μέσω του menu_item “settings” τα παραπάνω, ο χρήστης έχει δηλώσει τις απαιτήσεις του και πλέον θέλει να τις δει να προβάλλονται.

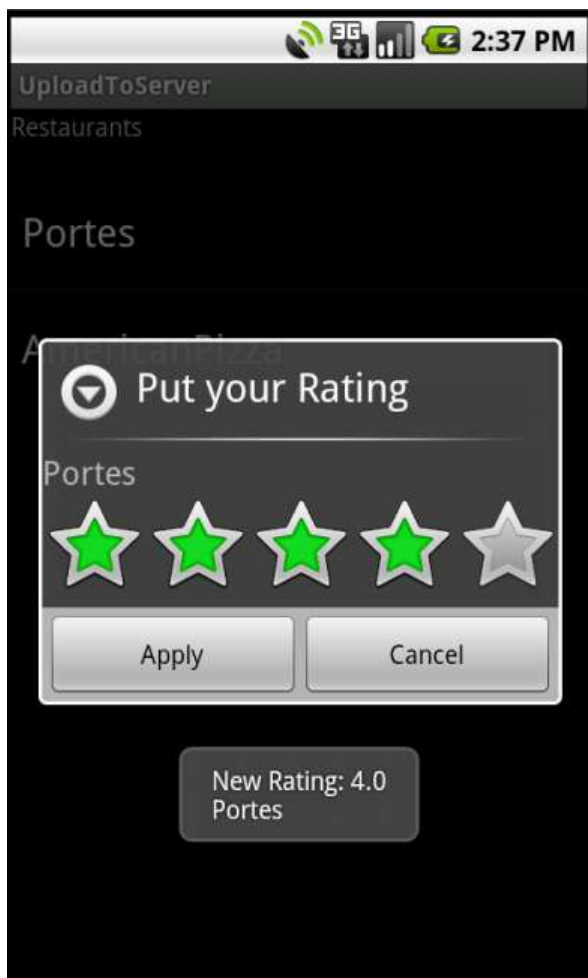
Έτσι για παράδειγμα, έστω ότι είμαστε στην περιοχή Ζωγράφου και επιλέγουμε να μας εμφανιστούν όλες οι προσφορές των εστιατορίων που βρίσκονται σε απόσταση 5km από την τρέχουσα θέση μας και μάλιστα σε μορφή λίστας. Τότε, ο χρήστης πατά το πλήκτρο display στο menu και παίρνει τα παρακάτω αποτελέσματα:



Ο χρήστης έχοντας μπροστά του τη λίστα με τα εστιατόρια που πληρούν τις απαιτήσεις του, μπορεί να πληροφορηθεί για τις προσφορές τους πατώντας πάνω σε κάθε ένα από αυτά.

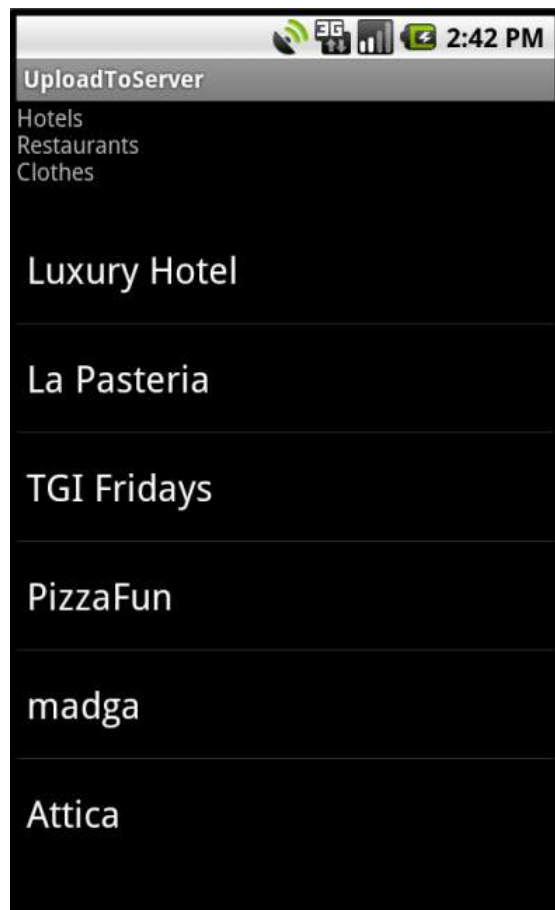
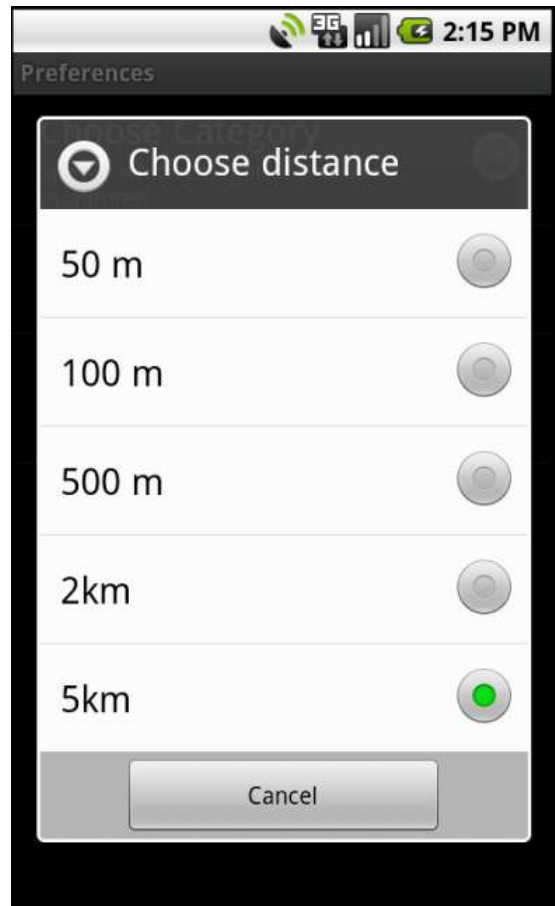
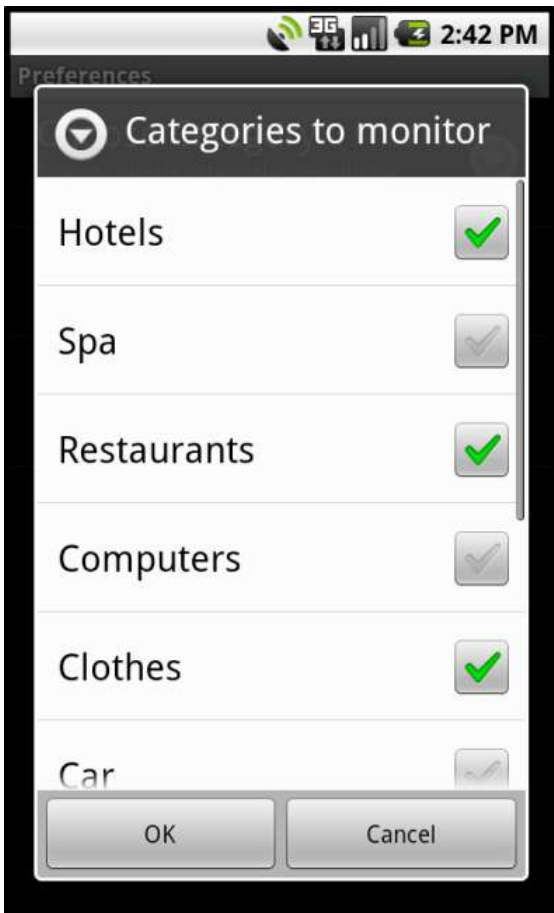
Η πληροφόρηση που έχει για τις προσφορές αφορά:

- a) Τη κατηγορία της προσφοράς
- b) Τις γεωγραφικές συντεταγμένες του καταστήματος που διαθέτει την προσφορά αυτή
- c) Συνοπτική περιγραφή της προσφοράς
- d) Ημερομηνία έναρξης και ημερομηνία λήξης της προσφοράς
- e) Το link του καταστήματος ώστε ο χρήστης να μπορεί να περιηγηθεί στο site, να μάθει περισσότερες πληροφορίες για το κατάστημα
- f) Αξιολόγηση της προσφοράς από άλλους χρήστες



Παράλληλα, ο χρήστης μπορεί να αξιολογήσει και αυτός με τη σειρά του τη συγκεκριμένη προσφορά τοποθετώντας τη βαθμολογία του από ένα μέχρι πέντε αστέρια, ανάλογα με το πόσο ικανοποιημένος ή πόσο κοντά στις επιθυμίες του ήταν η συγκεκριμένη προσφορά. Επίσης, το rating για την κάθε προσφορά που εμφανίζεται προκύπτει από το μέσο όρο των βαθμολογιών των χρηστών που έχουν αξιολογήσει μέχρι εκείνη τη στιγμή, η διαδικασία υπολογισμού του γίνεται από τον server και η τιμή αυτή αποθηκεύεται στην database στο πεδίο της συγκεκριμένης προσφοράς. Όποτε, λοιπόν, ο χρήστης ζητήσει πληροφορία για την προσφορά αυτή, “κατεβάζει” ουσιαστικά και το rating αυτής.

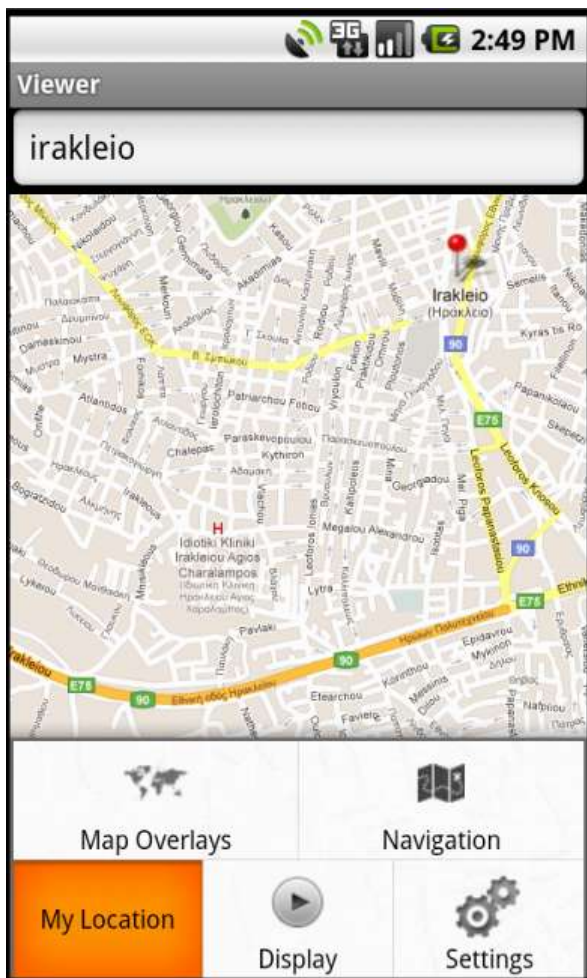
Εκτός από προβολή σε λίστα, ο χρήστης μπορεί να επιλέξει να του εμφανιστούν τα διαθέσιμα αποτελέσματα και πάνω στο χάρτη, ώστε να έχει μια εικόνα και των αποστάσεων των καταστημάτων από την τρέχουσα θέση του. Έτσι, για παράδειγμα, αν ο χρήστης μεταβεί στην περιοχή του Ηρακλείου και επιθυμεί να του εμφανιστούν οι προσφορές που σχετίζονται με εστιατόρια, ξενοδοχεία και καταστήματα ρουχισμού, σε απόσταση 5km από τη θέση του τότε θα έχουμε τα εξής αποτελέσματα:





Πατώντας πάνω στα διαφορετικά εικονίδια-rins που αντιστοιχούν στις διαφορετικές κατηγορίες, ο χρήστης μπορεί να πάρει ξανά τις πληροφορίες που χρειάζεται για τις προσφορές που του είναι διαθέσιμες.

Τέλος, με το εικονίδιο “my location” ο χρήστης αν έχει μεταβεί σε κάποια άλλη περιοχή μπορεί να επιστρέψει και να προβάλλει μπροστά του την τρέχουσα θέση του.



3.3 Υλοποίηση Εφαρμογής

Client

Ο client/χρήστης μέσω της εφαρμογής Viewer έχει τη δυνατότητα να καθορίσει τις προτιμήσεις αναζήτησης προσφορών. Αυτό επιτυγχάνεται μέσα από το menu_item “Settings”. Εκεί έχουμε δημιουργήσει ένα Preference Screen Layout , όπου οι επιλογές του χρήστη αποθηκεύονται στη μορφή Shared Preferences.

Οι Shared Preferences είναι ένας απλός και ελαφρύς ,από θέμα κατανάλωσης πόρων συστήματος, μηχανισμός αποθήκευσης δεδομένων της εφαρμογής. Επιτρέπουν στον χρήστη να αποθηκεύει ζεύγη key/value δεδομένων σαν named preferences και να τα χρησιμοποιεί σε κάθε κλάση της εφαρμογής του. Το preference key είναι απλά ένα string που ταυτοποιεί μοναδικά τη συγκεκριμένη preference και η preference value είναι η τιμή της preference αυτής.

Οι τύποι δεδομένων που υποστηρίζονται από την SharedPreferences class , μέσω της οποίας μπορούμε να δημιουργήσουμε named preferences , είναι: Boolean, float, integer, long, string. Λόγω της δυνατότητάς τους αυτής, χρησιμοποιούνται για να αποθηκεύουν γρήγορα default τιμές, class instance variables, την τρέχουσα κατάσταση του user interface και φυσικά τις επιλογές χρήστη. Επίσης, χρησιμοποιούνται συχνά για να κρατούν μόνιμα δεδομένα κατά τη διάρκεια των user sessions και για να κάνουν διαθέσιμες τις επιλογές αυτές σε κάθε component της εφαρμογής.

Μέσω, λοιπόν, του Preference Screen Layout που δημιουργήσαμε στην κλάση ShowPreferences.java , καθορίσαμε τρία τέτοια ζεύγη Shared Preferences.

- a. Ένα με key = “CATEGORY_PREF” , με value κάποια ή κάποιες από τις επιλογές κατηγορίας προσφορών.
- b. Ένα με key= “PROXIMITY_PREF”, με value την απόσταση που επιλέγει ο χρήστης για να κάνει η εφαρμογή την αναζήτηση προσφορών
- c. Ένα με key= “DISPLAY_PREF”, με value είτε “List” είτε “Map” ανάλογα με τον τρόπο που επιθυμεί να του παρουσιασθούν τα αποτελέσματα.

Στην περίπτωση των SharedPreferences (b) και (c), η value που μπορεί να πάρει η καθεμία από αυτές είναι μοναδική και καθορίζεται από radiobuttons , όπως είδαμε παραπάνω.

Αντίθετα, στην περίπτωση που ο χρήστης επιλέγει κατηγορία προσφορών, υπάρχει η δυνατότητα να επιλέξει ταυτόχρονα περισσότερες από μια κατηγορίες, οπότε και έχουν νόημα τα checkboxes . Εκεί ουσιαστικά χρησιμοποιούμε μια άλλη κλάση την *ListPreferenceMultiSelect.java*. Η κλάση αυτή, μέσω της μεθόδου *parseStoredValue* δέχεται την value της preference με key =“CATEGORY_PREF” σαν ενιαίο string, στο οποίο τα ονόματα των επιμέρους κατηγοριών χωρίζονται με ένα string SEPARATOR. Η μέθοδος αυτή χωρίζει ,λοιπόν, το ενιαίο string και αποθηκεύει σε πίνακα τα ονόματα των κατηγοριών που ο χρήστης επέλεξε.

Για να μπορέσουμε να πάρουμε και να επεξεργαστούμε τις preferences από οποιαδήποτε κλάση της εφαρμογής Viewer ,καλούμε τη μέθοδο `getDefaultSharedPreferences()` με παράμετρο το context της εφαρμογής.

```
SharedPreferences showprefs;  
showprefs = PreferenceManager.getDefaultSharedPreferences(this);  
String preference_string2= showprefs.getString("DISPLAY_PREF", "");  
String preference_string3= showprefs.getString("PROXIMITY_PREF", "");
```

Στην περίπτωση της preference "CATEGORY_PREF", έχουμε:

```
String preference_string1= showprefs.getString("CATEGORY_PREF", "");  
String[] selected=ListPreferenceMultiSelect.parseStoredValue(preference_string1);
```

οπότε στον πίνακα `selected[]` υπάρχουν τώρα αποθηκευμένα τα ονόματα των διαφορετικών κατηγοριών που ο χρήστης έχει επιλέξει.

Στη συνέχεια, αφού έχουμε αποθηκεύσει τα παραπάνω, ακολουθεί η αποστολή των απαιτήσεων του χρήστη στον server ώστε να μπορεί να επεξεργαστεί τα ληφθέντα στοιχεία και να απαντήσει κατάλληλα στο χρήστη.

Τα στοιχεία που χρειάζεται να αποσταλούν είναι:

- a) Οι κατηγορίες προσφορών που ενδιαφέρουν το χρήστη
- b) Το πλήθος αυτών, όπως καταγράφονται στον πίνακα `selected[]`. Η πληροφορία αυτή θα χρειαστεί στον server ο οποίος θα αναλάβει να "σπάσει" ο ίδιος το ενιαίο string με τους separators ενδιάμεσα.
- c) Η απόσταση από την τρέχουσα θέση , μέχρι την οποία ο χρήστης θέλει να γίνει η αναζήτηση.
- d) Η τρέχουσα θέση του χρήστη , με τη μορφή των γεωγραφικών του συντεταγμένων (latitude & longitude)

Η αποστολή των παραπάνω πληροφοριών γίνεται με μήνυμα POST στον server που εξυπηρετεί την εφαρμογή με τη μορφή NameValuePairs , ώστε να μπορούν να επεξεργαστούν ανεξάρτητα από τον server.

Συγκεκριμένα:

```
HttpClient client = new DefaultHttpClient();  
HttpPost post= new HttpPost("http://147.102.7.162:80/prosfores/first.php");  
  
String preference_string= prefs.getString("CATEGORY_PREF", "");  
String[] selected=ListPreferenceMultiSelect.parseStoredValue(preference_string);  
  
List <NameValuePair> pairs = new ArrayList<NameValuePair>();  
pairs.add(new BasicNameValuePair("number", Integer.toString(selected.length));  
pairs.add(new BasicNameValuePair("key", prefs.getString("CATEGORY_PREF", "")));  
pairs.add(new BasicNameValuePair("prox", prefs.getString("PROXIMITY_PREF", "")));  
pairs.add(new BasicNameValuePair("lat", Double.toString(lat));  
pairs.add(new BasicNameValuePair("lng", Double.toString(lng));  
  
post.setEntity( new UrlEncodedFormEntity(pairs));
```

Τα NameValuePairs είναι ουσιαστικά ζεύγη attribute name-attribute value. Τα ονόματα αυτά μπορούν να χρησιμοποιηθούν στη συνέχεια μέσα στο αρχείο first.php που “χτυπά” η κλήση του client και οι τιμές τους να διαβαστούν και να αλλαχθούν εκεί. Η ανάγνωση τους στο αρχείο first.php του server γίνεται ως εξής:

```
$key= $_POST["key"];
$number= $_POST["number"];
$parts = explode("OV=I=XseparatorX=I=VO", $key);
$prox=floatval($_POST["prox"]);
$lat=deg2rad(floatval($_POST["lat"]));
$lng=deg2rad(floatval($_POST["lng"]));
```

Παρατηρούμε πως με τη 3^η εντολή το ενιαίο string που περιλαμβάνει η attribute value με όνομα “key” σπάει και στη μεταβλητή πίνακα parts αποθηκεύονται οι κατηγορίες προσφορών που επέλεξε ο χρήστης.

Στις επόμενες τρεις εντολές τα attribute values που αντιστοιχούν στα attribute names prox,lat, lng όπως στάλθηκαν από τον client μετατρέπονται σε πραγματικές τιμές ,float και σε ακτίνια(rad) αντίστοιχα.

Έχοντας γνώση πλέον των απαιτήσεων του χρήστη, ο server συνδέεται με τη βάση δεδομένων MySQL, στην οποία βρίσκονται αποθηκευμένες οι προσφορές.

Σε πρώτη φάση επιχειρείται η σύνδεση με τη database με όνομα “prosfores” από τον server που ακούει στην IP 147.102.7.162 στην πόρτα 80.

```
$user_name="root";
$password="19654";
$database="prosfores";
$server="localhost";
```

```
mysql_connect($server, $user_name, $password);
$db_found=mysql_select_db($database);
```

Εάν η σύνδεση είναι επιτυχής τότε γίνεται ερώτημα στον πίνακα με όνομα “discount_table” της database να αναζητήσει στη στήλη του με όνομα “Category” το όνομα που είναι αποθηκευμένο στη μεταβλητή “key”. Το αποτέλεσμα θα αποθηκευτεί στη μεταβλητή “result”. Όταν βρεθεί όντως μια εγγραφή με περιεχόμενο στη στήλη της κατηγορίας το ζητούμενο, τότε η εγγραφή αυτή ελέγχεται για το αν ικανοποιεί τις απαιτήσεις που ο χρήστης έχει θέσει σε απόσταση.

Αν ικανοποιεί τις προϋποθέσεις, τότε το αποτέλεσμα εισάγεται σαν string στη μεταβλητή πίνακα “output”. Κάθε στοιχείο του πίνακα output Η διαδικασία αυτή επαναλαμβάνεται για όλα τα ονόματα κατηγοριών που υπάρχουν στον πίνακα parts, όπως δημιουργήθηκε παραπάνω.

Η παραπάνω διαδικασία υλοποιείται στο παρακάτω κομμάτι κώδικα:

```
if ($db_found) {

$SQL="SELECT*FROM discount_table WHERE Category LIKE '%$key%'";

$result=mysql_query($SQL);

while ($db_field = mysql_fetch_assoc($result)) {
```

```

        $dblalt=deg2rad($db_field['Latitude']);
        $dblng=deg2rad($db_field['Longitude']);
        $d=acos(sin($lat)*sin($dblalt)+cos($lat)*cos($dblalt)*cos($lng-
        $dblng))*$R;

        if ($d<$prox){
array_push($output,$db_field);
}

}

```

Ο client με τη σειρά του λαμβάνει την απάντηση από τον server και την αποθηκεύει στη μεταβλητή τύπου string και με όνομα result.

```

HttpResponse response = client.execute(post);
HttpEntity resEntity= response.getEntity();
is = resEntity.getContent();

BufferedReader reader = new BufferedReader(new
InputStreamReader(is,"iso-8859-1"),8);
sb = new StringBuilder();
sb.append(reader.readLine() + "\n");
String line = null;
while ((line = reader.readLine()) != null) {
        sb.append(line + "\n");
}

is.close();

result=sb.toString();

```

Με τις παραπάνω εντολές ο χρήστης παίρνει την απάντηση του server και την σπάει σε γραμμές ανάλογα με τα στοιχεία του πίνακα output. Δηλαδή ενώ ο πίνακας output είχε ως στοιχεία του τα περιεχόμενα των εγγραφών του πίνακα της database που αντιστοιχούσαν στις επιλογές του χρήστη, στη μεταβλητή result τα μετατρέπουμε όλα σε ενιαίο string, όπου τα αποτελέσματα των διαφορετικών εγγραφών βρίσκονται σε διαφορετικές γραμμές.

Στη συνέχεια, για να διαχειριστούμε το περιεχόμενο της μεταβλητής result δημιουργούμε ένα πίνακα JSONArray με JSON objects. Ένα JSONObject είναι ένα αταξινόμητο σετ από ζευγάρια name/values. Η εξωτερική του μορφή του JSONObject είναι ένα string μέσα σε {} με "," ανάμεσα στα διαφορετικά ζευγάρια names και τα values και ":" μεταξύ του name και της value του κάθε ζευγαριού.

Παράδειγμα της μορφής που εμφανίζεται ένα JSONObject είναι το παρακάτω:

```
[{"DiscountID":"14","Shop":"CompTech","Category":"Computers","Latitude":"35.31876","Longitude":"25.13467","Description":"New Laptops...","StartDate":"2009-9-9","EndDate":"2009-9-10","Link":"www.tech.gr","Ranking":"2.96"}]
```

Έτσι, λοιπόν, διαμορφώνουμε το ενιαίο string σε διαδοχικά JSONObjects και για να πάρουμε τις values που χρειαζόμαστε, χρησιμοποιούμε τις μεθόδους *getString*, *getDouble* ανάλογα κιάλας με τον τύπο της κάθε value.

Τα παραπάνω υλοποιούνται μέσα από τον κώδικα:

```
jArray = new JSONArray(reply);
JSONObject json_data=null;

for(int i=0;i<jArray.length();i++){
    json_data = jArray.getJSONObject(i);
    shop_name=json_data.getString("Shop");
    category_name=json_data.getString("Category");
    lat_value=json_data.getDouble("Latitude");
    long_value=json_data.getDouble("Longitude");
    details=json_data.getString("Description");
    startdate=json_data.getString("StartDate");
    enddate=json_data.getString("EndDate");
    link=json_data.getString("Link");
    ranking=json_data.getDouble("Ranking");
}
```

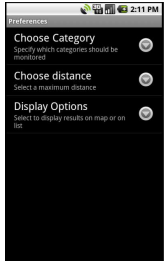
Εφόσον, έχουμε αποθηκεύσει τις τιμές των values σε αντίστοιχες μεταβλητές στη συνέχεια μπορούμε να τις επεξεργαστούμε, όπως θέλουμε.

Το διάγραμμα ροής των δεδομένων μεταξύ client, server και database για την εξαγωγή των ζητούμενων πληροφοριών από τη βάση δεδομένων φαίνεται παρακάτω:

Client (Viewer Application)



1. Ο χρήστης καθορίζει μέσα από το Preference Screen Layout τις προτιμήσεις του.



2. Δημιουργεί ζευγάρια name/value pairs, που αντιστοιχούν στα δεδομένα που θέλει να στείλει στον server για να ξεκινήσει η αναζήτηση, τα οποία είναι:
α) Κατηγορίες προσφορών
β) Απόσταση από την τρέχουσα θέση που επιθυμεί να μεταβεί.
γ) Συντεταγμένες τρέχουσας θέσης

3. Ο χρήστης στέλνει ένα POST μήνυμα στον server

`first.php`

4. Ο server υποδέχεται το μήνυμα POST, λαμβάνει τις επιλογές του χρήστη και εκτελεί ερώτημα στη βάση δεδομένων.

5. Η database αναζητά στον πίνακα με όνομα `discount_table` τις εγγραφές εκείνες που αντιστοιχούν στις απαιτήσεις του χρήστη και επιστρέφει τα περιεχόμενα αυτών.

6. Ο server υποδέχεται το μήνυμα-απάντηση της SQL database και το προωθεί στον client

7. Ο χρήστης υποδέχεται την Απάντηση ως ενιαίο string, τη μετατρέπει σε JSON Object και την επεξεργάζεται καταλλήλως.

Server (IP:147.102.7.162)



Database



Παρόμοια είναι και η ανταλλαγή μηνυμάτων όταν ο client στέλνει την αξιολόγησή του με τη μορφή βαθμολογίας από το 1-5 στον server. Εκεί η εφαρμογή παίρνει την πληροφορία έτσι όπως καταχωρείται από τον χρήστη στο rating bar, και μαζί με το όνομα του καταστήματος /προσφοράς την οποία βαθμολογεί ο χρήστης στέλνονται στον server και αυτός τα προωθεί στη database, όπου και εισάγεται η βαθμολογία στο πεδίο του πίνακα με όνομα Ranking.

Συνοπτικά, η ανταλλαγή μηνυμάτων φαίνεται στο ακόλουθο διάγραμμα:

Client (Viewer Application)



Server (IP:147.102.7.162)



Database



1. Μέσα από το Dialog Box, ο χρήστης βαθμολογεί την προσφορά και η επιλογή του αποθηκεύεται στη μεταβλητή rank.



2. Στη συνέχεια το ranking του χρήστη και το όνομα του καταστήματος που προσφέρει την προσφορά τοποθετούνται, όπως και πριν σε ζεύγη name/value pairs και στέλνονται στον server.

3. Ο χρήστης αποστέλλει μήνυμα POST στον server.

rating.php

4. Ο server δέχεται το μήνυμα και το επεξεργάζεται με το αρχείο rating.php.



5. Γίνεται σύνδεση με τη database και εισάγεται η τιμή στο πεδίο Ranking σε εκείνη την εγγραφή με όνομα, αυτό που στάλθηκε.

Στο σημείο αυτό, θα εξηγήσουμε τη σύνδεση και τα ερωτήματα που πραγματοποιούνται μεταξύ server και database.

Σε κάθε περίπτωση, η σύνδεση server και database γίνεται με τις παρακάτω εντολές:

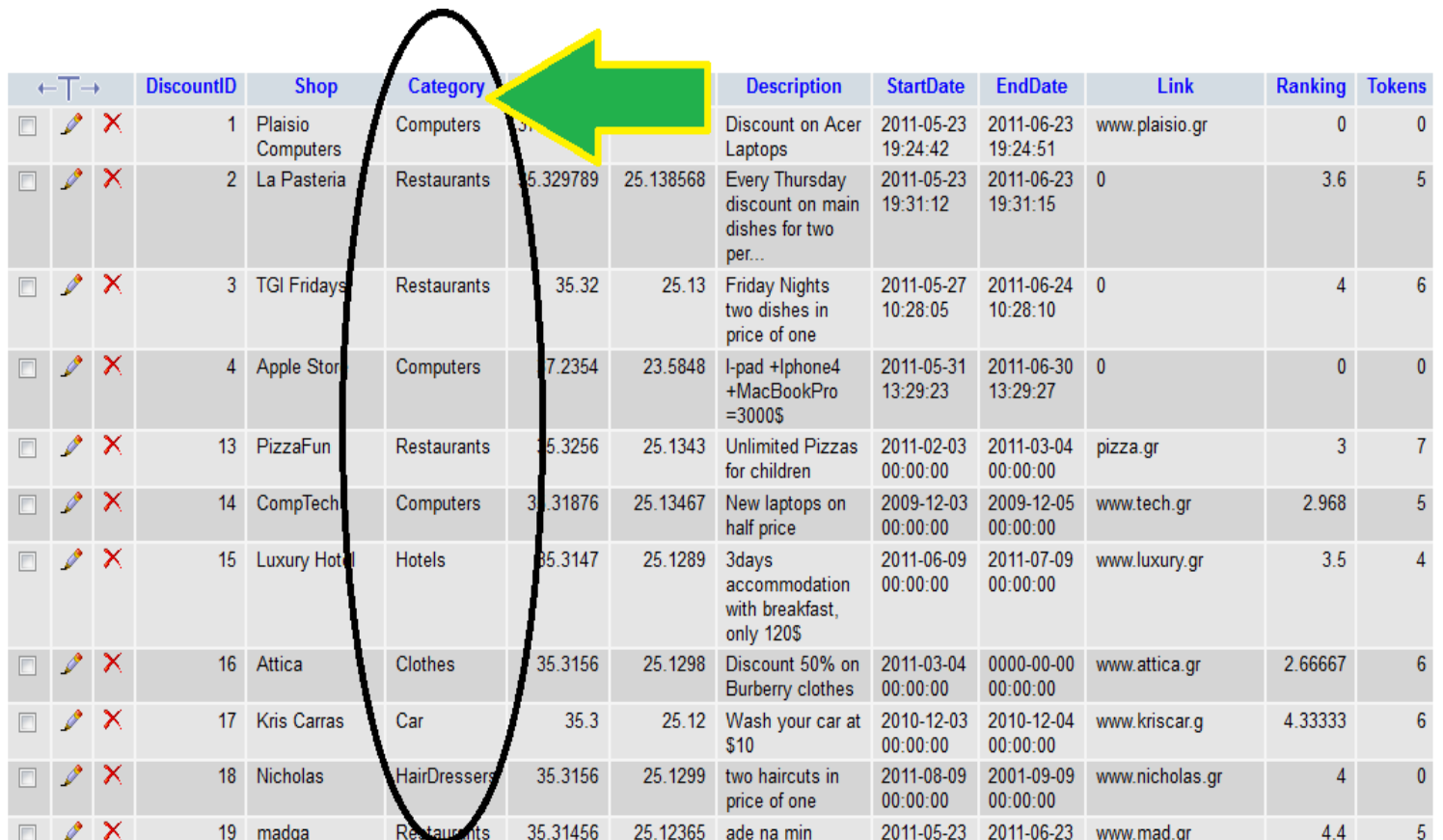
```
$user_name="admin";  
$password="admin";  
$database="prosfores";  
$server="147.102.7.162";
```

```
$db= mysql_connect($server, $user_name, $password) or die ('Error:'.mysql_error());  
mysql_select_db($database,$db);
```

Στην πρώτη περίπτωση που ο χρήστης στέλνει τα κριτήρια αναζήτησης που επιθυμεί, γίνεται το εξής ερώτημα στη database:

```
SELECT*FROM discount_table WHERE Category LIKE '%$key%'";
```

Όπου γίνεται αναζήτηση στον πίνακα με όνομα *discount_table* που φαίνεται παρακάτω για να βρεθεί αντιστοιχία εγγραφής που στο πεδίο πίνακα με όνομα *Category* θα έχει τιμή ίδια με αυτή που αναγράφεται στη μεταβλητή *key*.



	DiscountID	Shop	Category	Description	StartDate	EndDate	Link	Ranking	Tokens
<input type="checkbox"/>	1	Plaisio Computers	Computers	Discount on Acer Laptops	2011-05-23 19:24:42	2011-06-23 19:24:51	www.plaisio.gr	0	0
<input type="checkbox"/>	2	La Pasteria	Restaurants	Every Thursday discount on main dishes for two per...	2011-05-23 19:31:12	2011-06-23 19:31:15	0	3.6	5
<input type="checkbox"/>	3	TGI Fridays	Restaurants	Friday Nights two dishes in price of one	2011-05-27 10:28:05	2011-06-24 10:28:10	0	4	6
<input type="checkbox"/>	4	Apple Stor	Computers	I-pad +Iphone4 +MacBookPro =3000\$	2011-05-31 13:29:23	2011-06-30 13:29:27	0	0	0
<input type="checkbox"/>	13	PizzaFun	Restaurants	Unlimited Pizzas for children	2011-02-03 00:00:00	2011-03-04 00:00:00	pizza.gr	3	7
<input type="checkbox"/>	14	CompTech	Computers	New laptops on half price	2009-12-03 00:00:00	2009-12-05 00:00:00	www.tech.gr	2.968	5
<input type="checkbox"/>	15	Luxury Hotel	Hotels	3days accommodation with breakfast, only 120\$	2011-06-09 00:00:00	2011-07-09 00:00:00	www.luxury.gr	3.5	4
<input type="checkbox"/>	16	Attica	Clothes	Discount 50% on Burberry clothes	2011-03-04 00:00:00	0000-00-00 00:00:00	www.attica.gr	2.66667	6
<input type="checkbox"/>	17	Kris Carras	Car	Wash your car at \$10	2010-12-03 00:00:00	2010-12-04 00:00:00	www.kriscar.g	4.33333	6
<input type="checkbox"/>	18	Nicholas	HairDressers	two haircuts in price of one	2011-08-09 00:00:00	2001-09-09 00:00:00	www.nicholas.gr	4	0
<input type="checkbox"/>	19	madqa	Restaurants	ade na min	2011-05-23	2011-06-23	www.mad.gr	4.4	5

Στην περίπτωση που ο χρήστης στέλνει τη βαθμολογία του για μια προσφορά, ο server συνδέεται με τη database με την ίδια ακριβώς εντολή και η εισαγωγή γίνεται ως εξής:

```
$query = "SELECT Ranking ,Tokens FROM discount_table WHERE Shop LIKE '%$shop%'";
$result = mysql_query($query);
```

```
$ranking=$row['Ranking'];
```

```
$tokens=$row['Tokens'];
```

```
$rank= ($tokens * $ranking +$rank)/($tokens+1);
```

```
$tokens1=$tokens+1;
```

```
mysql_query ("UPDATE discount_table SET Ranking='$rank', Tokens= '$tokens1' WHERE Shop='$shop'");
```

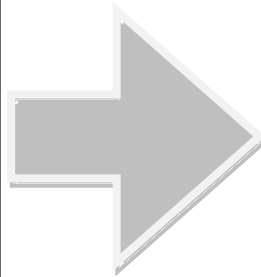
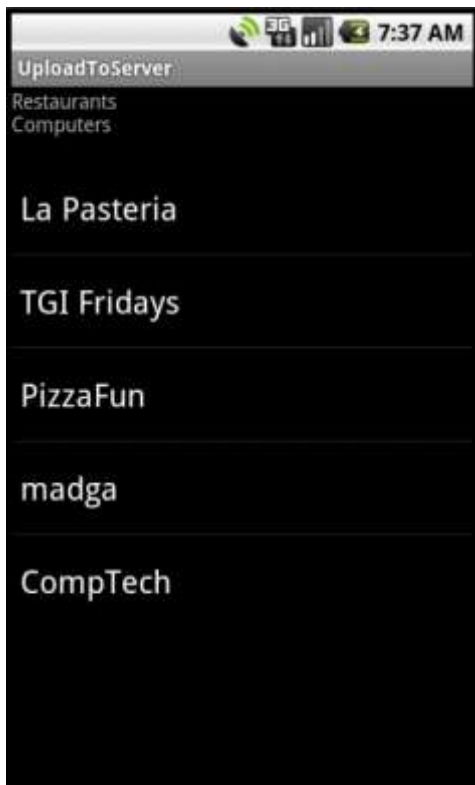
Βάση των παραπάνω εντολών, αποθηκεύεται στις μεταβλητές *ranking* και *tokens* το περιεχόμενο των πεδίων με όνομα **Ranking** και **Tokens** της εγγραφής εκείνης, που έχει όνομα στο πεδίο *Shop* που αντιστοιχεί στην τιμή της μεταβλητής *shop*.

Πιο συγκεκριμένα, για κάθε νέο χρήστη που βαθμολογεί μια προσφορά, η μεταβλητή *tokens* αυξάνεται κατά μία μονάδα, ενώ αυτόματα υπολογίζεται ο μέσος όρος βαθμολογίας της συγκεκριμένης προσφοράς που προκύπτει από τις μέχρι τότε βαθμολογίες και το μέχρι τότε πλήθος χρηστών και την καινούρια βαθμολογία.

Τέλος, γίνεται εισαγωγή των νέων στοιχείων στα παραπάνω πεδία με την εντολή UPDATE και SET.

Για παράδειγμα, έχουμε την εγγραφή PizzaFun, όπου δεν έχει εισάγει κανένας χρήστης μέχρι εκείνη τη στιγμή κάποια βαθμολογία.

DiscountID	Shop	Category	Latitude	Longitude	Description	StartDate	EndDate	Link	Ranking	Tokens
1	Plaisio Computers	Computers	37.234521	23.567898	Discount on Acer Laptops	2011-05-23 19:24:42	2011-06-23 19:24:51	www.plaisio.gr	0	0
2	La Pasteria	Restaurants	35.329789	25.138568	Every Thursday discount on main dishes for two per...	2011-05-23 19:31:12	2011-06-23 19:31:15	0	4.33	3
3	TGI Fridays	Restaurants	35.32	25.13	Friday Nights two dishes in price of one	2011-05-27 10:28:05	2011-06-24 10:28:10	0		
4	Apple Store	Computers	37.2354	23.5848	Lead MacBook +MacBookPro =3000\$	2011-05-31 13:29:23	2011-06-30 13:29:27	0		0
13	PizzaFun	Restaurants	35.3256	25.1343	Unlimited Pizzas for children	2011-02-03 00:00:00	2011-03-04 00:00:00	pizza.gr	0	0
14	CompTech	Computers	35.31876	25.13467	New laptops on half price	2009-12-03 00:00:00	2009-12-05 00:00:00	www.tech.gr	4.66667	6
15	Luxury Hotel	Hotels	35.3147	25.1289	3days accommodation with breakfast, only 120\$	2011-06-09 00:00:00	2011-07-09 00:00:00	www.luxury.gr	3.5	4
16	Attica	Clothes	35.3156	25.1298	Discount 50% on Burberry clothes	2011-03-04 00:00:00	0000-00-00 00:00:00	www.attica.gr	2.66667	6
17	Kris Carras	Car	35.3	25.12	Wash your car at \$10	2010-12-03 00:00:00	2010-12-04 00:00:00	www.kriscar.g	4.33333	6
18	Nicholas	HairDressers	35.3156	25.1299	two haircuts in price of one	2011-08-09 00:00:00	2001-09-09 00:00:00	www.nicholas.gr	4	0
19	madga	Restaurants	35.31456	25.12365	ade na min trelathw	2011-05-23 00:00:00	2011-06-23 00:00:00	www.mad.gr	4.4	5



Μετά την εισαγωγή της βαθμολογίας από τον χρήστη, τα πεδία Ranking και Tokens αλλάζουν και ο πίνακας της database ανανεώνεται με τις νέες τιμές.

Sort by key: None

+ Options

	DiscountID	Shop	Category	Latitude	Longitude	Description	StartDate	EndDate	Link	Ranking	Tokens
<input type="checkbox"/>	1	Plaisio Computers	Computers	37.234521	23.567898	Discount on Acer Laptops	2011-05-23 19:24:42	2011-06-23 19:24:51	www.plaisio.gr	0	0
<input type="checkbox"/>	2	La Pasteria	Restaurants	35.329789	25.138568	Every Thursday discount on main dishes for two per...	2011-05-23 19:31:12	2011-06-23 19:31:15	0	4.33333	3
<input type="checkbox"/>	3	TGI Fridays	Restaurants	35.32	25.13	Friday Nights two dishes in price of one	2011-05-27 10:28:05	2011-06-24 10:28:10	0		10
<input type="checkbox"/>	4	Apple Store	Computers	37.2354	23.5848	Load iPhone4 +MacBookPro =3000\$	2011-05-31 13:29:23	2011-06-30 13:29:27	0		0
<input type="checkbox"/>	13	PizzaFun	Restaurants	35.3256	25.1343	Unlimited Pizzas for children	2011-02-03 00:00:00	2011-03-04 00:00:00	pizza.gr	5	1
<input type="checkbox"/>	14	CompTech	Computers	35.31876	25.13467	New laptops on half price	2009-12-03 00:00:00	2009-12-05 00:00:00	www.tech.gr	4.06667	6
<input type="checkbox"/>	15	Luxury Hotel	Hotels	35.3147	25.1289	3days accommodation with breakfast, only 120\$	2011-06-09 00:00:00	2011-07-09 00:00:00	www.luxury.gr	3.5	4
<input type="checkbox"/>	16	Attica	Clothes	35.3156	25.1298	Discount 50% on Burberry clothes	2011-03-04 00:00:00	0000-00-00 00:00:00	www.attica.gr	2.66667	6
<input type="checkbox"/>	17	Kris Carras	Car	35.3	25.12	Wash your car at \$10	2010-12-03 00:00:00	2010-12-04 00:00:00	www.kriscar.g	4.33333	6
<input type="checkbox"/>	18	Nicholas	HairDressers	35.3156	25.1299	two haircuts in price of one	2011-08-09 00:00:00	2001-09-09 00:00:00	www.nicholas.gr	4	0

3.4 Ιστοσελίδα για την εισαγωγή προσφορών

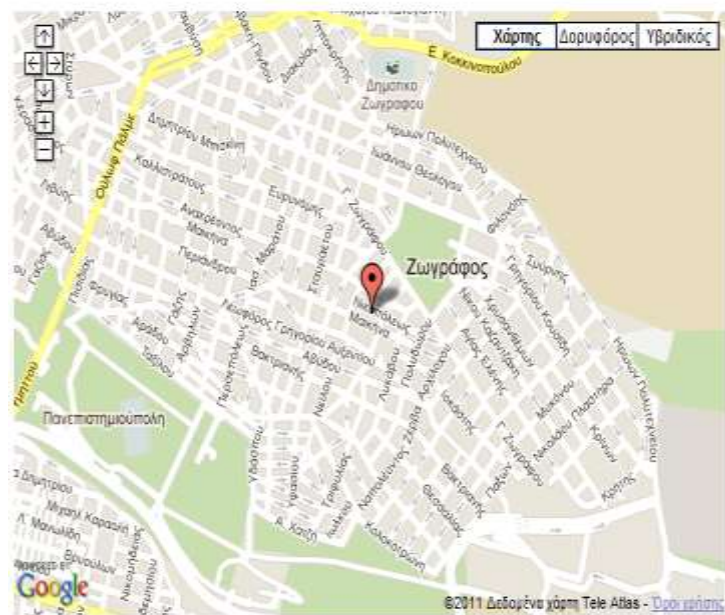
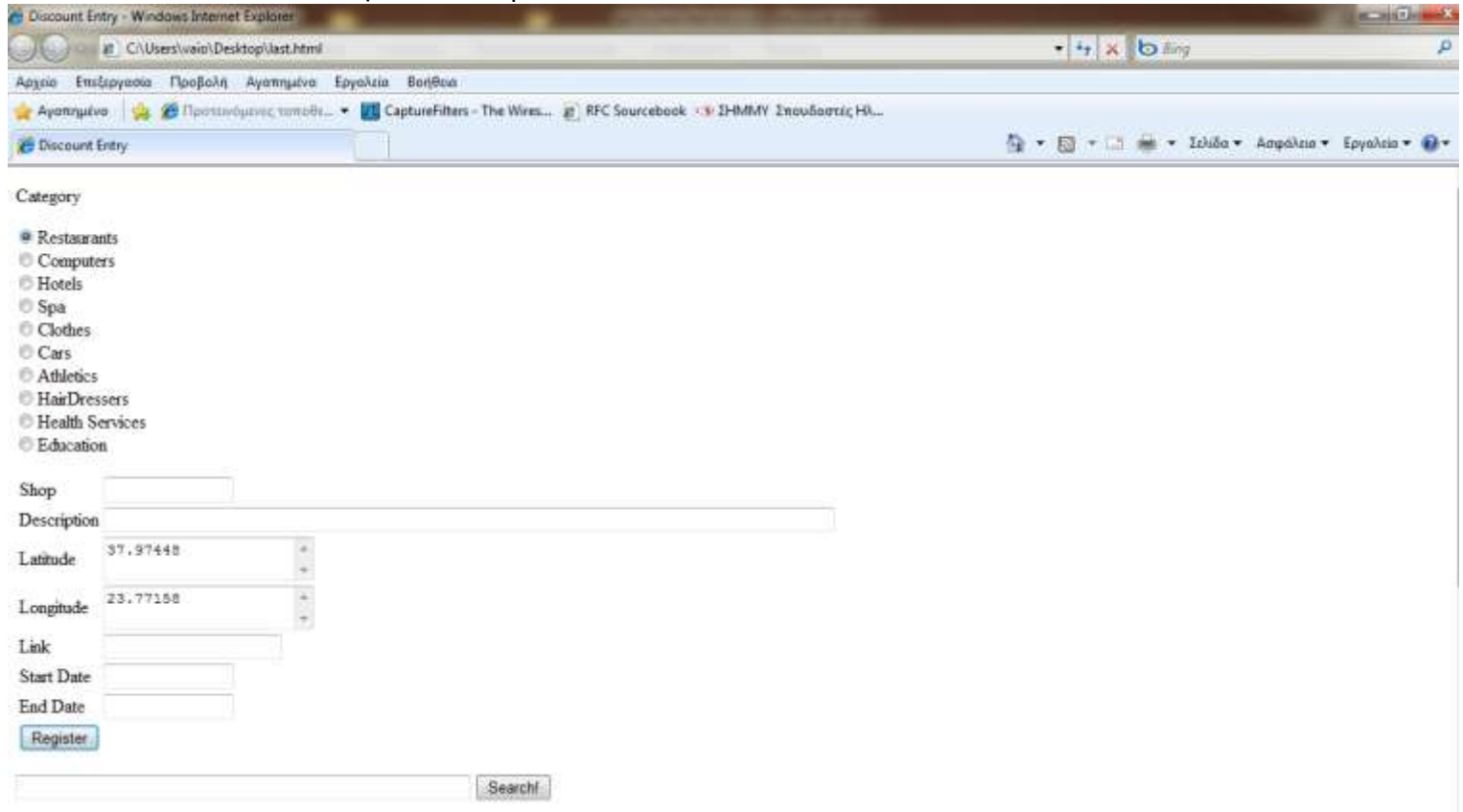
Στην πλευρά του server έχουμε δημιουργήσει επίσης μια ιστοσελίδα για την εισαγωγή προσφορών από την πλευρά των καταστηματαρχών στη βάση δεδομένων.

Συγκεκριμένα, όποιος θέλει να εισάγει μια προσφορά έχει τη δυνατότητα:

- a) Να επιλέξει την κατηγορία της προσφοράς που επιθυμεί (τις ίδιες κατηγορίες μπορεί να επιλέξει και ο χρήστης μέσω της εφαρμογής Viewer που τρέχει στο κινητό τερματικό)
- b) Να καταγράψει προσωπικά του στοιχεία, όπως το όνομα του καταστήματος, μια περιγραφή της προσφοράς που διατίθεται στους πελάτες, ένα link με κάποιο site του καταστήματος, τη διάρκεια της προσφοράς(ημερομηνία έναρξης-λήξης)
- c) Στο site προσφέρεται η δυνατότητα να προσδιοριστεί η τοποθεσία του καταστήματος με δύο τρόπους:
 1. Αν ο χρήστης ξέρει τη διεύθυνση του καταστήματός του, τότε τη γράφει στο πεδίο , πατάει Search και αυτόματα οι συντεταγμένες latitude, longitude αλλάζουν.
 2. Αν πάλι ο χρήστης επιθυμεί να προσδιορίσει τη θέση του πάνω στο χάρτη, τότε μπορεί να μετακινήσει το pin στην επιθυμητή θέση και τότε και πάλι οι συντεταγμένες θα αλλάξουν αυτόματα.

Μετά την ολοκλήρωση των παραπάνω, πατώντας το Register η προσφορά καταχωρείται στη βάση δεδομένων και οι πληροφορίες εισάγονται στα ανάλογα πεδία του πίνακα discount_table της database "prosfores".

Η ιστοσελίδα φαίνεται παρακάτω:



Παρακάτω φαίνεται η εισαγωγή μιας προσφοράς στη βάση δεδομένων:

Category

- Restaurants
- Computers
- Hotels
- Spa
- Clothes
- Cars
- Athletics
- HairDressers
- Health Services
- Education

Shop

Description

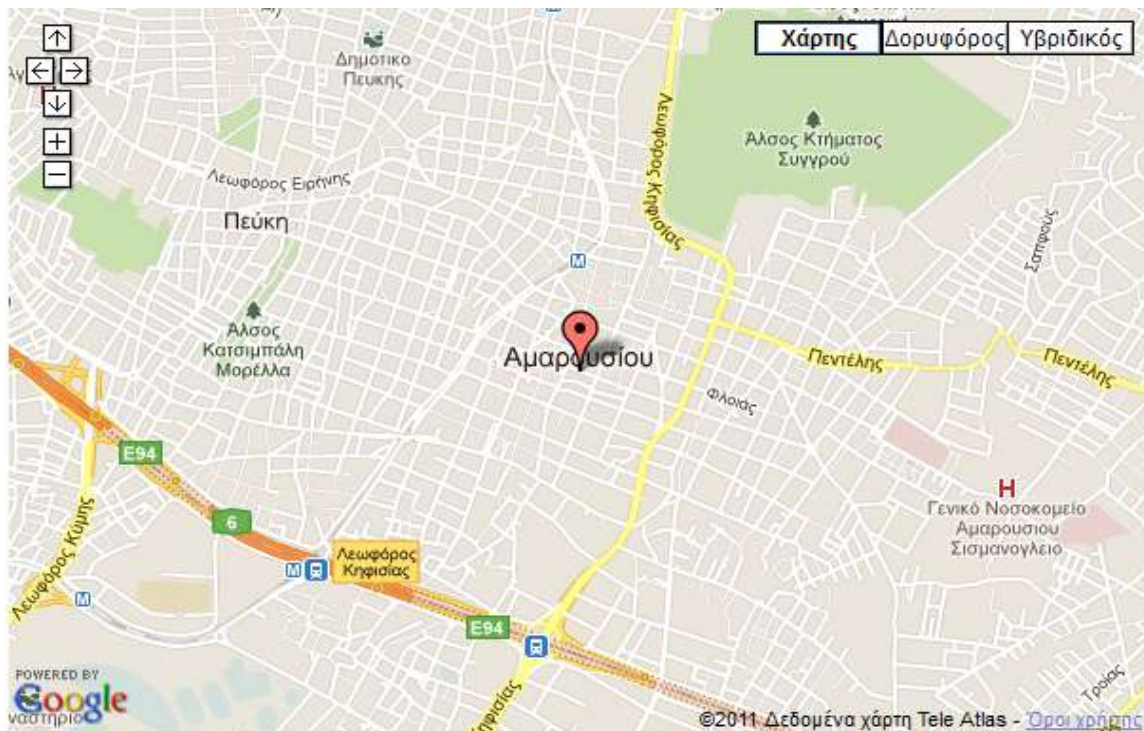
Latitude

Longitude

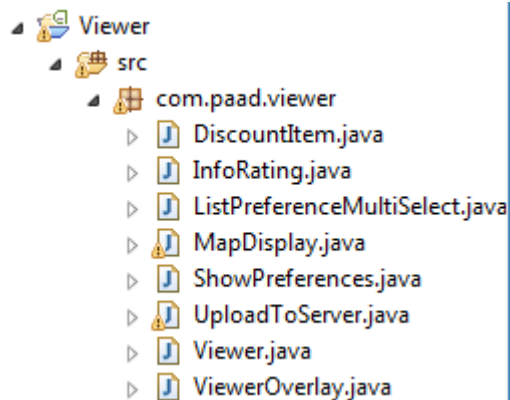
Link

Start Date

End Date



3.5 Διάρθρωση του κώδικα της εφαρμογής Viewer



Οι κλάσεις από τις οποίες αποτελείται η εφαρμογή φαίνονται στο διπλανό σχήμα. Η καθεμιά από αυτές υλοποιεί μια διαφορετική λειτουργία.

Η κύρια κλάση **Viewer.java** υλοποιεί την activity που “τρέχει”, όταν ανοίγει η εφαρμογή. Συγκεκριμένα, εμφανίζει την τρέχουσα θέση του χρήστη πάνω σε χάρτη, ενώ μαζί με τη **ViewerOverlay.java** πετυχαίνει τη γραφική αναπαράσταση πάνω στο χάρτη της θέσης του χρήστη με ένα pin.

Μέσα στη Viewer υλοποιείται το μενού της εφαρμογής με τα πέντε menu_items, όπως αυτό εμφανίζεται πατώντας το πλήκτρο “Menu” στο κινητό.

Επίσης, κατασκευάζεται και το dialog_box που εμφανίζεται με το πάτημα του πλήκτρου Map_Overlays, όπου ο χρήστης έχει τη δυνατότητα να επιλέξει τρόπο εμφάνισης του χάρτη.

Τέλος, πατώντας τα υπόλοιπα πλήκτρα του Menu, ξεκινούν άλλες Activities.

Με την κλάση **ShowPreferences.java** μεταβαίνουμε σε ένα Preference Screen Layout που καθορίζεται στο xml αρχείο res/xml/preference_screen1, όπου ο χρήστης επιλέγει από τρία preference controls τύπου Spinner(επιλέγοντας το συγκεκριμένο preference control, εμφανίζεται ένα dialog box με μια λίστα τιμών από τις οποίες ο χρήστης θα επιλέξει την επιθυμητή).

Η κλάση **DiscountItem.java** χρειάζεται για την επεξεργασία των αποτελεσμάτων που επιστρέφονται από τη database. Συγκεκριμένα, περιγράφει τα χαρακτηριστικά κάθε προσφοράς όταν ο χρήστης επιλέγει να προβάλλει το σύνολο των προσφορών σε λίστα.

Η κλάση **InfoRating.java** υλοποιεί το dialog box που επιτρέπει στον χρήστη να αξιολογήσει μια προσφορά χρησιμοποιώντας μια rating bar. Η υλοποίηση του rating bar γίνεται μέσω xml αρχείου res/layout/inforating.xml

Η κλάση **ListPreferenceMultiSelect.java** χρησιμοποιείται για τον διαχωρισμό του ενιαίου string με τις επιλεχθείσες κατηγορίες προσφορών σε διαφορετικά μεταξύ τους string, ώστε να γίνει πιο εύκολη η χρησιμοποίησή τους.

Η κλάση **MapDisplay.java** χρησιμοποιείται για την εμφάνιση των διαθέσιμων προσφορών πάνω στο χάρτη με διαφορετικά pins ανά κατηγορία.

Η κλάση **UploadToServer.java** στέλνει τις επιλογές του χρήστη όσον αφορά την αξιολόγηση κάποιας προσφοράς στον server και υποδέχεται την απάντηση που στέλνει ο server και την εμφανίζει σε μορφή λίστας.

Παράρτημα

Πηγαίος Κώδικας Εφαρμογής

DiscountItem.java

```
package com.paad.viewer;

public class DiscountItem {
    private String shop;
    private String category;
    private Double latitude;
    private Double longitude;
    private String description;
    private String startdate;
    private String enddate;
    private String link;
    private Double rank;

    public String getStartdate() {
        return startdate;
    }

    public String getEnddate() {
        return enddate;
    }

    public String getLink() {
        return link;
    }

    public Double getRank() {
        return rank;
    }

    public String getShop() {
        return shop;
    }

    public String getCategory() {
        return category;
    }

    public Double getLatitude() {
        return latitude;
    }

    public Double getLongitude() {
        return longitude;
    }

    public String getDescription() {
        return description;
    }
}

public DiscountItem(String shop, String category, Double latitude,
    Double longitude, String description, String startdate, String enddate,
    String link, Double rank) {
    super();
    this.shop = shop;
    this.category = category;
    this.latitude = latitude;
    this.longitude = longitude;
    this.description = description;
    this.startdate=startdate;
    this.enddate=enddate;
    this.link=link;
    this.rank=rank;
}
```

```

@Override
    public String toString() {
        return shop ;
    }

}

```

InfoRating.java

```

package com.paad.viewer;

import android.app.Activity;
import android.os.Bundle;

public class InfoRating extends Activity{

    @Override
    public void onCreate(Bundle icle) {
        super.onCreate(icle);

        setContentView(R.layout.inforating);
    }
}

```

ListPreferenceMultiSelect.java

```

package com.paad.viewer;

import android.app.AlertDialog.Builder;
import android.content.Context;
import android.content.DialogInterface;
import android.preference.ListPreference;
import android.util.AttributeSet;

public class ListPreferenceMultiSelect extends ListPreference{
    private static final String SEPARATOR = "OV=I=XseparatorX=I=VO";
    private boolean[] mClickedDialogEntryIndices;

    public ListPreferenceMultiSelect(Context context, AttributeSet attrs) {
        super(context, attrs);
        // TODO Auto-generated constructor stub

        mClickedDialogEntryIndices = new boolean[getEntries().length];
    }

    @Override
    public void setEntries(CharSequence[] entries) {
        super.setEntries(entries);
        mClickedDialogEntryIndices = new boolean[entries.length];
    }

    public ListPreferenceMultiSelect(Context context) {
        this(context, null);
    }
}

```

```

@Override
protected void onPrepareDialogBuilder(Builder builder) {
    CharSequence[] entries = getEntries();
    CharSequence[] entryValues = getEntryValues();

    if (entries == null || entryValues == null || entries.length !=
        entryValues.length ) {
        throw new IllegalStateException("ListPreference requires an entries
            array and an entryValues array which are both the same length");
    }

    restoreCheckedEntries();
    builder.setMultiChoiceItems(entries, mClickedDialogEntryIndices,
        new DialogInterface.OnMultiChoiceClickListener() {
            public void onClick(DialogInterface dialog, int which, boolean
val) {
                mClickedDialogEntryIndices[which] = val;
            }
        });
    }

    public static String[] parseStoredValue(CharSequence val) {
        if ( "".equals(val) )
            return null;
        else
            return ((String)val).split(SEPARATOR);
    }

    private void restoreCheckedEntries() {
        CharSequence[] entryValues = getEntryValues();

        String[] vals = parseStoredValue(getValue());
        if ( vals != null ) {
            for ( int j=0; j<vals.length; j++ ) {
                String val = vals[j].trim();
                for ( int i=0; i<entryValues.length; i++ ) {
                    CharSequence entry = entryValues[i];
                    if ( entry.equals(val) ) {
                        mClickedDialogEntryIndices[i] = true;
                        break;
                    }
                }
            }
        }
    }

@Override
// protected void onDialogClosed(boolean positiveResult) {
    super.onDialogClosed(positiveResult);

    CharSequence[] entryValues = getEntryValues();
    if (positiveResult && entryValues != null) {
        StringBuffer value = new StringBuffer();
        for ( int i=0; i<entryValues.length; i++ ) {
            if ( mClickedDialogEntryIndices[i] ) {
                value.append(entryValues[i]).append(SEPARATOR);
            }
        }

        if (callChangeListener(value)) {
            String val = value.toString();
            if ( val.length() > 0 )

```

```

        val = val.substring(0, val.length()-SEPARATOR.length());
        setValue(val);
    }
}
}
}

```

MapDisplay.java

```

package com.paad.viewer;

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

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import com.google.android.maps.GeoPoint;
import com.google.android.maps.MapActivity;
import com.google.android.maps.MapController;
import com.google.android.maps.MapView;
import com.google.android.maps.Overlay;

import android.app.Activity;
import android.content.SharedPreferences;
import android.graphics.drawable.Drawable;
import android.location.Address;
import android.location.Geocoder;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.util.Log;
import android.widget.EditText;
import android.widget.TextView;

public class MapDisplay extends MapActivity {

    SharedPreferences prefs;
    String reply;
    MapController mapcontroller;
    ViewerOverlay markers,markers1,markers2,markers3, markers4,
    markers5, markers6, markers7, markers8;
    double latitude,longitude;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.mapdisplay);
        Bundle bundle= this.getIntent().getExtras();
        reply=bundle.getString("res");
        latitude = bundle.getDouble("latitude");
        longitude=bundle.getDouble("longitude");

        MapView myMapView=(MapView) findViewById(R.id.myMapView1);
        myMapView.setBuiltInZoomControls(true);
        myMapView.setClickable(true);
        myMapView.setStreetView(false);
        mapcontroller = myMapView.getController();
        mapcontroller.setZoom(15);
    }
}

```

```

        List<Overlay> overlays =myMapView.getOverlays();
        Drawable
drawable=this.getResources().getDrawable(R.drawable.pinrest);
        Drawable
drawable1=this.getResources().getDrawable(R.drawable.ping);
        Drawable
drawable2=this.getResources().getDrawable(R.drawable.hotels);
        Drawable
        drawable3=this.getResources().getDrawable(R.drawable.computer);
        Drawable
drawable4=this.getResources().getDrawable(R.drawable.cars);
        Drawable
drawable5=this.getResources().getDrawable(R.drawable.apps);
        Drawable
drawable6=this.getResources().getDrawable(R.drawable.gym);
        Drawable drawable7=this.getResources().getDrawable(R.drawable.flagred);
        Drawable
drawable8=this.getResources().getDrawable(R.drawable.clothes);

markers =new ViewerOverlay(drawable,this);
markers1= new ViewerOverlay(drawable1,this);
markers2=new ViewerOverlay(drawable2,this);
markers3=new ViewerOverlay(drawable3,this);
markers4=new ViewerOverlay(drawable4,this);
markers5=new ViewerOverlay(drawable5,this);
markers6=new ViewerOverlay(drawable6,this);
markers7=new ViewerOverlay(drawable7,this);
markers8=new ViewerOverlay(drawable8,this);

overlays.add(markers);
overlays.add(markers1);
overlays.add(markers2);
overlays.add(markers3);
overlays.add(markers4);
overlays.add(markers5);
overlays.add(markers6);
overlays.add(markers7);
overlays.add(markers8);
try {
    addnewflag(latitude,longitude,"My Current Location","","");
} catch (IOException e1) {
    // TODO Auto-generated catch block
    e1.printStackTrace();
}

try{

JSONArray jArray = new JSONArray(reply);
JSONObject json_data=null;
String shop_name,details,category_name,startdate,enddate,link;

double lat_value=0.0, long_value=0.0, rating;
String addressString=null;
for(int i=0;i<jArray.length();i++){

        json_data = jArray.getJSONObject(i);
        shop_name=json_data.getString("Shop");
        category_name=json_data.getString("Category");

```



```

        lat_value=json_data.getDouble("Latitude");
        long_value=json_data.getDouble("Longitude");
        details=json_data.getString("Description");
        startdate=json_data.getString("StartDate");
        enddate=json_data.getString("EndDate");
        link=json_data.getString("Link");
        rating=json_data.getDouble("Ranking");
        String title=shop_name;

addressString="Category:"+category_name+"\n"+"Description:"+details+"\n"
+"StartDate: "+startdate+"\n" + "EndDate: "+enddate + "\n"+"Link:
"+link+"\n" +"Ranking: "+rating;

addnewflag(lat_value,long_value, title,addressString,category_name);

    }

    //txt.setText("coord"+lat_value + long_value);

} catch (JSONException e) {
    Log.e("log_tag", "Error parsing data "+e.toString());
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

}

private void addnewflag(double lat_value, double long_value, String title,
String info, String category) throws IOException {
    // TODO Auto-generated method stub

    GeoPoint newpoint = new
GeoPoint((int)(lat_value*1E6),(int)(long_value*1E6));
    mapcontroller.animateTo(newpoint);
    mapcontroller.setZoom(19);
    Geocoder geo =new Geocoder(this, Locale.getDefault());
    List<Address> addresses = geo.getFromLocation(lat_value,
long_value, 1);
    StringBuilder sb = new StringBuilder();
    if (addresses.size() > 0) {
        Address address = addresses.get(0);

        for (int i = 0; i < address.getMaxAddressLineIndex(); i++)
        sb.append(address.getAddressLine(i)).append("\n");
        sb.append("Longitude=");
        sb.append(address.getLongitude()).append("\n");
        sb.append("Latitude=");
        sb.append(address.getLatitude()).append("\n");
        sb.append(address.getLocality()).append("\n");
        sb.append(address.getPostalCode()).append("\n");
        sb.append(address.getCountryName());
    }
    if (info==""){
        info = info+"\n"+sb.toString();
        markersl.addItem(newpoint,title,info);
    }
}

```

```

    }
    else{
        info = info+"\n"+sb.toString();
        if(category.equals("Restaurants")){
markers.addNewItem(newpoint, title, info);
        }
        else if(category.equals("Hotels")){
markers2.addNewItem(newpoint, title, info);
        }
        else if (category.equals("Computers")){
markers3.addNewItem(newpoint, title, info);
        }
        else if (category.equals("Car")){
            markers4.addNewItem(newpoint, title, info);
        }
        else if (category.equals("Education")){
            markers5.addNewItem(newpoint, title, info);
        }
        else if (category.equals("Athletics")){
            markers6.addNewItem(newpoint, title, info);
        }
        else if (category.equals("Clothes")){
            markers8.addNewItem(newpoint, title, info);
        }
        else {
            markers7.addNewItem(newpoint, title, info);
        }
    }
}

@Override
protected boolean isRouteDisplayed() {
    // TODO Auto-generated method stub
    return false;
}
}

```

ShowPreferences.java

```

package com.paad.viewer;

import android.content.SharedPreferences;
import android.os.Bundle;
import android.preference.PreferenceActivity;

public class ShowPreferences extends PreferenceActivity {
    public static final String CATEGORY_PREF = "CATEGORY_PREF";
    public static final String PROXIMITY_PREF = "PROXIMITY_PREF";
    public static final String DISPLAY_PREF = "DISPLAY_PREF";

    SharedPreferences prefs;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        addPreferencesFromResource(R.xml.preference_screen1);
    }
}

```

```
}
```

UploadToServer.java

```
package com.paad.viewer;

import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import java.util.List;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.util.ByteArrayBuffer;
import org.apache.http.util.EntityUtils;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.Dialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.DialogInterface.OnClickListener;
import android.content.SharedPreferences.Editor;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.RatingBar;
import android.widget.RatingBar.OnRatingBarChangeListener;
```

```

import android.widget.TextView;
import android.widget.Toast;

public class UploadToServer extends Activity {
    static final private int DISCOUNT_DIALOG=1;
    static final private int STAR_RATING=2;
    static final private int INFOS=3;
    SharedPreferences showprefs;
    ListView discountListView;
    ArrayAdapter<DiscountItem> aa;
    TextView txt, texting;
    ArrayList<DiscountItem> discounts;
    DiscountItem disc;
    Double latitude;
    Double longitude;
    JSONArray jArray;
    String result, reply, text="";
    private static final String SEPARATOR = "OV=I=XseparatorX=I=VO";
    RatingBar ratingbar;
    Float rank;

    @Override
    public void onCreate(Bundle icle){
        super.onCreate(icle);

        Bundle bundle= this.getIntent().getExtras();
        latitude=bundle.getDouble("latitude");
        longitude=bundle.getDouble("longitude");
        reply=bundle.getString("res");
        showprefs =
PreferenceManager.getDefaultSharedPreferences(this);

        setContentView(R.layout.listview);
        txt =(TextView) this.findViewById(R.id.textview);

        String preference_string= showprefs.getString("CATEGORY_PREF",
        "");
        String[] selected =
        ListPreferenceMultiSelect.parseStoredValue(preference_string);

        for(int i=0; i<selected.length; i++){
            text= text+ (selected[i]+"\\n");
        }

        txt.setText(text);
        discountListView=
(ListView) this.findViewById(R.id.discountListView);

        discountListView.setOnItemClickListener(new
OnItemClickListener() {

            @Override
            public void onItemClick(AdapterView arg0, View arg1, int
arg2,

                long arg3) {
                // TODO Auto-generated method stub
                disc= discounts.get(arg2);

                showDialog(DISCOUNT_DIALOG);

```

```

        }

    });
    discounts =new ArrayList<DiscountItem>();
    int layoutID =android.R.layout.simple_list_item_1;
    aa = new ArrayAdapter<DiscountItem>(this,layoutID,discounts);
    discountListView.setAdapter(aa);

    try{

        JSONArray = new JSONArray(reply);
        JSONObject json_data=null;
        String shop_name,details,category_name, startdate, enddate,
link;
        Double lat_value, long_value, ranking;

        for(int i=0;i<jArray.length();i++){
            json_data = jArray.getJSONObject(i);
            shop_name=json_data.getString("Shop");
            category_name=json_data.getString("Category");
            lat_value=json_data.getDouble("Latitude");
            long_value=json_data.getDouble("Longitude");
            details=json_data.getString("Description");
            startdate=json_data.getString("StartDate");
            enddate=json_data.getString("EndDate");
            link=json_data.getString("Link");
            ranking=json_data.getDouble("Ranking");

            DiscountItem disc_item= new
DiscountItem(shop_name,category_name,lat_value,long_value, details,
startdate,enddate,link,ranking);

            addNewItem(disc_item);

        }

    }catch(JSONException e){
        Log.e("log_tag", "Error parsing data "+e.toString());
    }

}

private void addNewItem(DiscountItem disc_item) {
    // TODO Auto-generated method stub
    discounts.add(disc_item);
    aa.notifyDataSetChanged();
}

@Override
public Dialog onCreateDialog(int id){

```

```

switch(id){
case(DISCOUNT_DIALOG):{

LayoutInflater li =LayoutInflater.from(this);

View discountDetailsView =
li.inflate(R.layout.discounts_details, null);
AlertDialog.Builder discountDialog = new
AlertDialog.Builder(this);

discountDialog.setView(discountDetailsView);
discountDialog.setTitle("Shop");

discountDialog.setPositiveButton(R.string.rating,new
OnClickListener() {

public void onClick(DialogInterface dialog, int which) {
// TODO Auto-generated method stub
//Toast.makeText (UpshowDialog (INFOS);

showDialog (STAR_RATING);

}

});

discountDialog.setNegativeButton(R.string.cancel,new
OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {
// TODO Auto-generated method stub

}

});

return discountDialog.create();
}
case (INFOS):{
LayoutInflater li =LayoutInflater.from(this);
View discountDetailsView =
li.inflate(R.layout.discounts_details, null);
AlertDialog.Builder infoDialog = new
AlertDialog.Builder(this);

infoDialog.setView(discountDetailsView);
infoDialog.setTitle("Shop");

return infoDialog.create();

}
case (STAR_RATING):{
LayoutInflater li =LayoutInflater.from(this);
View view =li.inflate(R.layout.inforating, null);
AlertDialog.Builder builder =new
AlertDialog.Builder(this);

ratingbar = (RatingBar)
view.findViewById(R.id.ratingbar);

```

```

        builder.setTitle("Put your Rating");

        ratingbar.setOnRatingBarChangeListener(new
        OnRatingBarChangeListener() {
            public void onRatingChanged(RatingBar ratingBar, float
rating, boolean fromUser) {

                //Toast.makeText(UploadToServer.this, "New Rating:
" + rating, Toast.LENGTH_SHORT).show();

                rank=rating;
                sendRating(rank);

            }
        });

        builder.setPositiveButton(R.string.apply,new OnClickListener()
{

            public void onClick(DialogInterface dialog, int which) {
                // TODO Auto-generated method stub
                //Toast.makeText(UploadToServer.this, "New
Rating: " + rank, Toast.LENGTH_SHORT).show();
                sendRating(rank);

            }

        });

        builder.setNegativeButton(R.string.cancel,new OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {
                // TODO Auto-generated method stub

            }

        });

        builder.setCancelable(true);
        builder.setView(view);
        return builder.create();

    }

    return null;

}

@Override
public void onPrepareDialog(int id,Dialog dialog){
    switch(id){
        case(DISCOUNT_DIALOG):{
            AlertDialog discountDialog = (AlertDialog) dialog;
            discountDialog.setTitle(disc.getShop());
        }
        case(INFOS):{
            String discount_text= "Category="+disc.getCategory()+
"\n"+

```



```

        }
    }
}

```

Viewer.java

```

package com.paad.viewer;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.List;
import java.util.Locale;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;

import com.google.android.maps.GeoPoint;
import com.google.android.maps.MapActivity;
import com.google.android.maps.MapController;
import com.google.android.maps.MapView;
import com.google.android.maps.Overlay;

import android.app.AlertDialog;

import android.app.Dialog;

import android.content.Context;

import android.content.DialogInterface;
import android.content.DialogInterface.OnClickListener;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.SharedPreferences.Editor;

import android.graphics.drawable.Drawable;
import android.location.Address;
import android.location.Criteria;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.preference.PreferenceManager;

```

```

import android.util.Log;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.CompoundButton.OnCheckedChangeListener;
import android.widget.EditText;
import android.widget.Toast;

public class Viewer extends MapActivity {
    public static final String SATELLITE = "SATELLITE";
    public static final String TRAFFIC="TRAFFIC";

    MapController mapController;
    ViewerOverlay markers;
    SharedPreferences prefs;
    MapView myMapView;
    private CheckBox satelliteBox;
    private CheckBox trafficBox;
    private EditText myEditText;
    Double lat ;
    Double lng ;
    String result;
    Intent intent;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        Context context1=getApplicationContext();
        prefs=PreferenceManager.getDefaultSharedPreferences(context1);

        myEditText=(EditText) findViewById(R.id.myEditText);

        refreshPreferences();
        showCurrentLocation();
    }

    private void showCurrentLocation() {
        // TODO Auto-generated method stub
        LocationManager locationManager;
        String context = Context.LOCATION_SERVICE;
        locationManager = (LocationManager) getSystemService(context);

        Criteria criteria = new Criteria();
        criteria.setAccuracy(Criteria.ACCURACY_FINE);
        criteria.setAccuracy(Criteria.ACCURACY_COARSE);
        criteria.setAltitudeRequired(false);
        criteria.setBearingRequired(false);
        criteria.setCostAllowed(true);
        criteria.setPowerRequirement(Criteria.POWER_LOW);
        String provider = locationManager.getBestProvider(criteria, true);

        Location location =

```

```

        locationManager.getLastKnownLocation(provider);

        updateWithNewLocation(location);

        locationManager.requestLocationUpdates(provider, 2000, 10,
                                                locationListener);
    }

    @Override
    public void onResume(){
        super.onResume();
    }

    @Override
    public void onPause(){
        super.onPause();
    }

    @Override
    public void onDestroy(){
        super.onDestroy();
    }

    private final LocationListener locationListener = new
LocationListener() {
        public void onLocationChanged(Location location) {
            updateWithNewLocation(location);
        }

        public void onProviderDisabled(String provider){
            updateWithNewLocation(null);
        }

        public void onProviderEnabled(String provider){ }
        public void onStatusChanged(String provider, int status,
                                    Bundle extras){ }
    };

    private void updateWithNewLocation(Location location) {

        if (location != null) {
            // Update the map location.
            Double geoLat = location.getLatitude()*1E6;
            Double geoLng = location.getLongitude()*1E6;
            GeoPoint point = new GeoPoint(geoLat.intValue(),
                                           geoLng.intValue());

            mapController.animateTo(point);

            lat=location.getLatitude();
            lng=location.getLongitude();

            Geocoder gc = new Geocoder(this, Locale.getDefault());
            downloadLocationInfo(point, gc, lat, lng);
        }
    }
}

```

```

public void downloadLocationInfo(GeoPoint point, Geocoder gc,
    double latitude, double longitude) {
    String addressString = "No address found";
    // TODO Auto-generated method stub
    try {
        List<Address> addresses = gc.getFromLocation(latitude,
longitude, 1);
        StringBuilder sb = new StringBuilder();
        if (addresses.size() > 0) {
            Address address = addresses.get(0);

            for (int i = 0; i < address.getMaxAddressLineIndex(); i++)
                sb.append(address.getAddressLine(i)).append("\n");
            sb.append("Longitude=");
            sb.append(address.getLongitude()).append("\n");
            sb.append("Latitude=");
            sb.append(address.getLatitude()).append("\n");
            sb.append(address.getLocality()).append("\n");
            sb.append(address.getPostalCode()).append("\n");
            sb.append(address.getCountryName());
        }
        addressString = sb.toString();
        markers.addNewItem(point, "My Current
Location", addressString);
    } catch (IOException e) {}
    }
}

```

```

@Override
protected boolean isRouteDisplayed() {
    // TODO Auto-generated method stub
    return false;
}

static final private int MENU_LAYERS=Menu.FIRST;
static final private int NAVIGATION=Menu.FIRST+1;
static final private int MYLOCATION=Menu.FIRST+2;
static final private int SERVER=Menu.FIRST+3;
static final private int SETTINGS=Menu.FIRST+4;

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);

    menu.add(0, MENU_LAYERS, Menu.NONE, R.string.menulayers).setIcon(R.drawable.icon_world_map);

    menu.add(0, NAVIGATION, Menu.NONE, R.string.navigation).setIcon(R.drawable.images2);
    menu.add(0, MYLOCATION, Menu.NONE, R.string.mylocation);

    menu.add(0, SERVER, Menu.NONE, R.string.displaying).setIcon(R.drawable.play);

    menu.add(0, SETTINGS, Menu.NONE, R.string.settings).setIcon(R.drawable.settings);

    return true;
}

```

```

}

private static final int DIALOG_LAYERS = 1;

public boolean onOptionsItemSelected(MenuItem item){
    super.onOptionsItemSelected(item);
    prefs=PreferenceManager.getDefaultSharedPreferences(this);
    switch(item.getItemId()){
        case(MENU_LAYERS):
            myEditText.setVisibility(View.GONE);
            showDialog(DIALOG_LAYERS);
            return true;
        case(NAVIGATION):
            refreshPreferences();
            searchForLocation();

            return true;

        case(MYLOCATION):
            myEditText.setVisibility(View.GONE);

            refreshPreferences();
            showCurrentLocation();
            return true;

        case(SERVER):

            postData();
            if (prefs.getString("DISPLAY_PREF", "").equals("List")){
                intent = new Intent(this,UploadToServer.class);
            }
            else{
                intent = new Intent(this,MapDisplay.class);
            }
            intent.putExtra("latitude", lat);
            intent.putExtra("longitude", lng);
            intent.putExtra("res", result);
            startActivityForResult(intent, 0);

            return true;

        case (SETTINGS):

            Intent i = new Intent(this,ShowPreferences.class);
            startActivityForResult(i,0);
            return true;

    }

    return false;
}

```

```

public void searchForLocation() {
    myEditText.setVisibility(View.VISIBLE);
    myEditText.requestFocus();
    myEditText.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            String maparea=myEditText.getText().toString();
            changearea(maparea);

        }

    });
}

public void changearea(String area) {
    // TODO Auto-generated method stub
    Geocoder gc =new Geocoder(this, Locale.getDefault());
    List<Address> locations =null;
    GeoPoint newLocation = null;
    lat =0.0 ;
    lng = 0.0;

    try
    {
        locations=gc.getFromLocationName(area, 1);
        if(locations.size()>0){
            Toast.makeText(this, "country: " +
String.valueOf(locations.get(0).getCountryName()),
Toast.LENGTH_SHORT).show();
            lat= locations.get(0).getLatitude();
            lng=locations.get(0).getLongitude();
        }
        else{
            Toast.makeText(this, "record not found",
Toast.LENGTH_SHORT).show();
            return;
        }
    }
    catch(IOException io)
    {
        Toast.makeText(this, "Connection Error",
Toast.LENGTH_SHORT).show();
    }
    newLocation = new GeoPoint( (int) (lat * 1E6), (int)
(lng * 1E6));
    mapController.animateTo(newLocation);
    mapController.setZoom(17);
    downloadLocationInfo(newLocation,gc,lat,lng);
}
@Override

```

```

    public Dialog onCreateDialog(int id){
        switch(id){
            case(DIALOG_LAYERS):
                LayoutInflater li =LayoutInflater.from(this);
                View view =li.inflate(R.layout.overlay_dialog, null);
                AlertDialog.Builder builder =new
AlertDialog.Builder(this);
                satelliteBox = (CheckBox)
view.findViewById(R.id.satellite_layer);
                satelliteBox.setOnCheckedChangeListener( listener);
                trafficBox=(CheckBox)
view.findViewById(R.id.traffic_layer);
                trafficBox.setOnCheckedChangeListener(listener);
                builder.setTitle("Choose Map Display Option");
                builder.setPositiveButton(R.string.apply,new
OnClickListener() {

                    public void onClick(DialogInterface dialog, int
which) {

                        // TODO Auto-generated method stub
                        refreshPreferences();

                    }

                });
                builder.setNegativeButton(R.string.cancel,new
OnClickListener() {

                    @Override
                    public void onClick(DialogInterface dialog, int
which) {

                        // TODO Auto-generated method stub

                    }

                });
                builder.setIcon(R.drawable.map);
                builder.setCancelable(true);
                builder.setView(view);
                return builder.create();
            }
            return null;
        }

        @Override
        public void onPrepareDialog(int id,Dialog dialog){
            switch(id){
                case(DIALOG_LAYERS):
                    satelliteBox.setChecked(prefs.getBoolean(SATELLITE,
false));
            }

            super.onPrepareDialog(id, dialog);
        }
        private void refreshPreferences(){

            MapView myMapView=(MapView) findViewById(R.id.myMapView);
            myMapView.setBuiltInZoomControls(true);
            myMapView.setClickable(true);
            myMapView.setStreetView(false);

```

```

myMapView.setSatellite(prefs.getBoolean(SATELLITE, false));
myMapView.setTraffic(prefs.getBoolean(TRAFFIC, false));
mapController = myMapView.getController();
mapController.setZoom(17);

List<Overlay> overlays =myMapView.getOverlays();
Drawable drawable =
this.getResources().getDrawable(R.drawable.pinred);
markers =new ViewerOverlay(drawable,this);
overlays.add(markers);

}

private final OnCheckedChangeListener listener = new
OnCheckedChangeListener() {

    public void onCheckedChanged(CompoundButton buttonView,
boolean isChecked) {
        // TODO Auto-generated method stub
        int which=buttonView.getId();
        switch(which) {
            case R.id.satellite_layer:
                setSatelliteLayer(isChecked);
                break;
            case R.id.traffic_layer:
                setTrafficLayer(isChecked);
                break;
        }
    }

};

private void setSatelliteLayer(boolean isChecked) {

    // TODO Auto-generated method stub
    Editor editor =prefs.edit();
    editor.putBoolean(SATELLITE, isChecked);
    editor.commit();
}
private void setTrafficLayer(boolean isChecked) {
    // TODO Auto-generated method stub
    Editor editor =prefs.edit();
    editor.putBoolean(TRAFFIC, isChecked);
    editor.commit();
}
public String setTexty(){
    String text="teo";
    return text;
}

public void postData() {
    // TODO Auto-generated method stub

    HttpClient client = new DefaultHttpClient();
    //HttpPost post = new
HttpPost("http://10.0.2.2:80/first.php");
    HttpPost post = new
HttpPost("http://147.102.7.162:80/prosfores/first.php");

```



```

        InputStream is=null;
        StringBuilder sb=null;

        String preference_string=
prefs.getString("CATEGORY_PREF", "");
        String[] selected =
ListPreferenceMultiSelect.parseStoredValue(preference_string);

        try{
            List <NameValuePair> pairs = new
ArrayList<NameValuePair>();
            pairs.add(new
BasicNameValuePair("number", Integer.toString(selected.length)));

            pairs.add(new
BasicNameValuePair("key",prefs.getString("CATEGORY_PREF", "")));

            pairs.add(new
BasicNameValuePair("prox",prefs.getString("PROXIMITY_PREF", "")));
            pairs.add(new
BasicNameValuePair("lat",Double.toString(lat)));
            pairs.add(new
BasicNameValuePair("lng",Double.toString(lng)));

            post.setEntity( new UrlEncodedFormEntity(pairs));

            HttpResponse response = client.execute(post);
            HttpEntity resEntity= response.getEntity();
            is = resEntity.getContent();
        }
        catch(Exception e){
            Log.e("UploadToServer", "Error in http connection
"+e.toString());
        }

        //convert response to string
        try{

            BufferedReader reader = new BufferedReader(new
InputStreamReader(is,"iso-8859-1"),8);
            sb = new StringBuilder();
            sb.append(reader.readLine() + "\n");
            String line = null;
            while ((line = reader.readLine()) != null) {
                sb.append(line + "\n");
            }
            is.close();

            result=sb.toString();

        }catch(Exception e){
            Log.e("UploadToServer", "Error converting result
"+e.toString());
        }
    }
}

```

```
    }  
}
```

ViewerOverlay.java

```
package com.paad.viewer;  
  
import java.util.ArrayList;  
  
import android.app.AlertDialog;  
import android.content.Context;  
import android.graphics.drawable.Drawable;  
  
import com.google.android.maps.GeoPoint;  
import com.google.android.maps.ItemizedOverlay;  
import com.google.android.maps.OverlayItem;  
  
public class ViewerOverlay extends ItemizedOverlay<OverlayItem> {  
    private ArrayList<OverlayItem> items;  
    Context mContext;  
    public ViewerOverlay(Drawable defaultMarker, Context context) {  
        super(boundCenterBottom(defaultMarker));  
        items =new ArrayList<OverlayItem>();  
        mContext = context;  
        populate();  
        // TODO Auto-generated constructor stub  
    }  
    public void addNewItem(GeoPoint point,String markertext, String  
snippet){  
        items.add(new OverlayItem(point,markertext,snippet));  
        populate();  
    }  
    public void removeItem(int index){  
        items.remove(index);  
        populate();  
    }  
    @Override  
    protected OverlayItem createItem(int i) {  
        // TODO Auto-generated method stub  
        return items.get(i);  
    }  
    @Override  
    public int size() {  
        // TODO Auto-generated method stub  
        return items.size();  
    }  
  
    @Override  
    protected boolean onTap(int index){  
        OverlayItem item = items.get(index);  
        AlertDialog.Builder dialog = new  
AlertDialog.Builder(mContext);  
        dialog.setTitle(item.getTitle());  
        dialog.setMessage(item.getSnippet());  
        dialog.show();  
        return true;  
    }  
}
```

Server files

first.php

```
<?php

$user_name="root";
$password="19654";
$database="prosfores";
$server="localhost";

$number= $_POST["number"];

$output=array();
mysql_connect($server, $user_name, $password);
$db_found=mysql_select_db($database);

$key= $_POST["key"];
$parts = explode("OV=I=XseparatorX=I=VO",$key);

$i=0;
while ($i<$number) {

$key=$parts[$i];

$prox=floatval($_POST["prox"]);
$lat=deg2rad(floatval($_POST["lat"]));
$lng=deg2rad(floatval($_POST["lng"]));

$R=6371;

if($db_found){

$SQL="SELECT*FROM discount_table WHERE Category LIKE '%$key%'";

$result=mysql_query($SQL);

while ($db_field = mysql_fetch_assoc($result)) {
    $dblats=deg2rad($db_field['Latitude']);
    $dblng=deg2rad($db_field['Longitude']);
    $d=acos(sin($lat)*sin($dblats)+cos($lat)*cos($dblats)*cos($lng-$dblng))*$R;

    if ($d<$prox){
array_push($output,$db_field);
    // $output[]= $db_field;
}

}

}

else {
print "Database NOT Found ";
mysql_close();
}
$i++;
}
mysql_close();

print(json_encode($output));
```

?>

rating.php

<?php

```
$user_name="root";
$password="19654";
$databse="prosfores";
$server="localhost";

$rank=$_POST['rank'];
$shop=$_POST['shop'];

$rank=floatval($rank);

echo "Working".$rank;

$db= mysql_connect($server, $user_name, $password) or die
('Error:'.mysql_error());
mysql_select_db($databse,$db);

$query = "SELECT Ranking,Tokens FROM discount_table WHERE Shop LIKE
'%"$shop%"";
$result = mysql_query($query);
while($row = mysql_fetch_assoc($result))
{
$ranking=$row['Ranking'];
$tokens=$row['Tokens'];
$rank= ($tokens * $ranking +$rank)/($tokens+1);
$tokens1=$tokens+1;
mysql_query("UPDATE discount_table SET Ranking='$rank',Tokens= '$tokens1'
WHERE Shop='$shop'");

mysql_close($db);

}
?>
```

last.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
    <title>Discount Entry</title>
    <script
src="http://maps.google.com/maps?file=api&v=2&key=ABQIAAAAgrj58PbX
r2YriiRDqbnLlRSqrCjdkglBijPNIiYrqkVvDlR4QxRl47Yh2D_0C1l5KXQJGrbkSDvXFA"
type="text/javascript"></script>
    <script type="text/javascript">

function load() {
  if (GBrowserIsCompatible()) {
    var map = new GMap2(document.getElementById("map"));
    map.addControl(new GSmallMapControl());
    map.addControl(new GMapTypeControl());
    var center = new GLatLng(37.97245, 23.77435);
    map.setCenter(center, 15);
    geocoder = new GClientGeocoder();
    var marker = new GMarker(center, {draggable: true});
```

```

        map.addOverlay(marker);
        document.getElementById("lat").innerHTML =
center.lat().toFixed(5);
        document.getElementById("lng").innerHTML =
center.lng().toFixed(5);

        GEvent.addListener(marker, "dragend", function() {
var point = marker.getPosition();
            map.panTo(point);
document.getElementById("lat").innerHTML = point.lat().toFixed(5);
document.getElementById("lng").innerHTML = point.lng().toFixed(5);

        });

        GEvent.addListener(map, "moveend", function() {
            map.clearOverlays();
var center = map.getCenter();
            var marker = new GMarker(center, {draggable: true});
            map.addOverlay(marker);
            document.getElementById("lat").innerHTML =
center.lat().toFixed(5);
            document.getElementById("lng").innerHTML =
center.lng().toFixed(5);

            GEvent.addListener(marker, "dragend", function() {
var point =marker.getPosition();
                map.panTo(point);
document.getElementById("lat").innerHTML = point.lat().toFixed(5);
                document.getElementById("lng").innerHTML =
point.lng().toFixed(5);

            });

        });

    }
}

function showAddress(address) {
    var map = new GMap2(document.getElementById("map"));
    map.addControl(new GSmallMapControl());
    map.addControl(new GMapTypeControl());
    if (geocoder) {
        geocoder.getLatLng(
            address,
            function(point) {
                if (!point) {
                    alert(address + " not found");
                } else {
                    document.getElementById("lat").innerHTML =
point.lat().toFixed(5);
                    document.getElementById("lng").innerHTML =
point.lng().toFixed(5);
                    map.clearOverlays()
                    map.setCenter(point, 14);
var marker = new GMarker(point, {draggable: true});
                    map.addOverlay(marker);

                    GEvent.addListener(marker, "dragend", function() {
var pt = marker.getPosition();

```

```

        map.panTo(pt);
        document.getElementById("lat").innerHTML = pt.lat().toFixed(5);
        document.getElementById("lng").innerHTML =
pt.lng().toFixed(5);
    });

    GEvent.addListener(map, "moveend", function() {
        map.clearOverlays();
        var center = map.getCenter();
        var marker = new GMarker(center, {draggable: true});
        map.addOverlay(marker);
        document.getElementById("lat").innerHTML =
center.lat().toFixed(5);
        document.getElementById("lng").innerHTML =
center.lng().toFixed(5);

        GEvent.addListener(marker, "dragend", function() {
            var pt = marker.getPosition();
            map.panTo(pt);
            document.getElementById("lat").innerHTML = pt.lat().toFixed(5);
            document.getElementById("lng").innerHTML = pt.lng().toFixed(5);
        });
    });
}
}
);
}
}
</script>
</head>

```

```
<body onload="load()" onunload="GUnload()" >
```

```

<form name="myform" action="discount_entry.php" method="post">

    <p>
        <p> Category </p>
        <input type="radio" name="category"
value="Restaurants"
        CHECKED>Restaurants<br>

                <input type="radio" name="category"
value="Computers">Computers<br>
                <input type="radio" name="category"
value="Hotels">Hotels<br>
                <input type="radio" name="category"
value="Spa">Spa<br>
                <input type="radio" name="category"
value="Clothes">Clothes<br>
                <input type="radio" name="category"
value="Car">Cars<br>
                <input type="radio" name="category"
value="Athletics">Athletics<br>
                <input type="radio" name="category"
value="HairDressers">HairDressers<br>

```

```

                <input type="radio" name="category"
value="Health_Services">Health Services<br>
                <input type="radio" name="category"
value="Education">Education<br>
            </p>
            <table border="0">
            <tr>
                <td> Shop </td>
                <td><input type="text" name="shop" maxlength="13"
size="13"></td>
            </tr>
            <tr>
                <td> Description </td>
                <td><input type="text" name="description"
maxlength="100" size="100"></td>
            </tr>
            <tr>
                <td> Latitude </td>
                <td ><textarea id="lat" name="lat"></textarea></td>
            </tr>
            <tr>
                <td> Longitude </td>
                <td ><textarea id="lng" name="lng"></textarea></td>
            </tr>
            <tr>
                <td> Link </td>
                <td><input type="text" name="link" maxlength="20"
size="20"></td>
            </tr>
            <tr>
                <td> Start Date </td>
                <td><input type="date" name="start" maxlength="13"
size="13"></td>
            </tr>
            <tr>
                <td> End Date </td>
                <td><input type="date" name="finish" maxlength="13"
size="13"></td>
            </tr>
            <tr>
                <td colspan="2"><input type="submit"
value="Register"></td>
            </tr>
        </table>
    </p>
</form>

<form action="#" onsubmit="showAddress(this.address.value); return
false">
    <p>
        <input type="text" size="60" name="address" />
        <input type="submit" value="Search!" />
    </p>
</form>

```

```

<p align="left">

</p>
<p>
<div align="center" id="map" style="width: 600px; height:
400px"><br/></div>
</p>
</div>
</body>

</html>

```

discount_entry.php

```

<?php
    $category=$_POST['category'];
    $shop=$_POST['shop'];
    $description= $_POST['description'];
    $latitude=$_POST['latitude'];
    $longitude=$_POST['longitude'];
    $link=$_POST['link'];
    $start=$_POST['start'];
    $finish=$_POST['finish'];

if(!$category || !$shop || !$description || !$latitude || !$longitude ||
!$link || !$start || !$finish)
{
    echo 'You have not entered all the required details.<br/>'
    .'Please go back and try again.';
    exit;
}

$category= addslashes($category);
$shop= addslashes($shop);
$description=addslashes($description);
$latitude=doubleval($latitude);
$longitude=doubleval($longitude);
$link=addslashes($link);
$start=addslashes($start);
$finish=addslashes($finish);

$user_name="root";
$password="19654";
$databse="prosfores";
$server="localhost";

$db= mysql_connect($server, $user_name, $password) or die
('Error:'.mysql_error());;

```



```
mysql_select_db($database,$db);

$query = "INSERT INTO discount_table(Shop, Category,
Latitude,Longitude,Description,StartDate,EndDate,Link) VALUES ('$shop',
'$category',
'$latitude','$longitude','$description','$start','$finish','$link')";

@mysql_query($query) or die('The error was: ' . mysql_error());

echo "Database Updated with:". $shop ;

?>
```

Βιβλιογραφία

- [1] Μ.Ε.Θεολόγου , «Δίκτυα Κινητών και Προσωπικών Επικοινωνιών», Εκδόσεις Τζιόλα, 2010.
- [2] Reto Meier , «Professional Android 2 Application Development», Wiley Publishing, 2010.
- [3] Joshen Schiller, Agnes Voisard, «Location Based Services», Morgan Kaufmann,Elsevier,2004.
- [4] Stefan Steiniger, Moritz Neun, Alister Edwardes , «Foundations of Location Based Services- Lesson 1», Cartouche 2008.
- [5] ESRI, «A Brief History of LBS and How OpenLS fits into the new value chain», LBS-Hist2003, Industry Solutions 2003.
- [6] V.Liutkauskas, D.Matulis, R.Plestys, «Location Based Services», ISSN 1392-1215 ELECTRONIKA IR ELEKTROTECHNIKA.2004.Nr.3(52)
- [7] Chen Xiuwan, Zhang Feizhou, Sun Min and Luo Yuanhua , «System Architecture of LBS Based on Spatial Information Integration», 2004,IEEE.
- [8] Narushige Shiode, Chao Li, Michael Batty. Paul Longley and David Maguire, «The Impact and Penetration of Location-Based Services». , Paper 50, 2002.
- [9] ITU- «Mobility-The shape of things to come»
<http://www.itu.int/itu-news/issue/2001/07/mobility.html>
- [10] ITU-«Mobile Broadband»
<http://www.itu.int/net/itu-news/issues/2010/03/09.aspx>
- [11] Jacub Marek Borkowski, «Performance of Cell-Id +RTT Hybrid Positioning Method for UMTS»,
Master of Science Thesis , 2003.
- [12] Xianhua Shu, Zhenjun Du, Rong Chen, «Research on Mobile Location Service Design Based on Android», 2009, IEEE.
- [13] Jerome E. Dobson and Peter F. Fisher, «GEOSLAVERY»,
Spring 2003, IEEE.
- [14] Guanling Chen and Faruq Rahman, «Analyzing Privacy Designs of Mobile Social Networking Applications», 2008, IEEE.

- [15] Bugra Gedik, Member, IEEE, and Ling Liu, Senior Member, IEEE, « Protecting Location Privacy with Personalized k-Anonymity: Architecture and Algorithms», 2008,IEEE.
- [16] Konstantinos P. Demestichas, Evgenia F. Adamopoulou, John G. Markoulidakis, and Michael E.Theologou, «Towards Anonymous Mobile Community Services», 2007.
- [17] Ken, Mang Lung Yiu, «Location Privacy in LBS-Part 2», Department of Computer Science, Aalborg
- [18] Daniele Quercia, Neal Lathia, Francesco Calabrese, Giusy Di Lorenzo, Jon Crowcroft, «Recommending Social Events from Mobile Phone Location Data».
- [19] Tongyu Zhu, Chen Wang, Guannan Jia, Jian Huang, « Toward Context-Aware Location Based Services», ICEIE 2010.
- [20] TelecomCircle -«Introduction to Location Based Services»,
<http://www.telecomcircle.com/2009/06/introduction-to-lbs>.
- [21] Kathy Sierra, Bert Bates, «Head First Java», O'REILLY,2005.
- [22] Luke Welling, Laura Thomson, «PHP and MySQL Web development», Sams Publishing,2003.
- [23] Android Developers- <http://developer.android.com/guide/index.html>
- [24] Α. Ανδρονικάκης, «Σύστημα Προσδιορισμού Θέσης Σταθμών Βάσης σε Δίκτυα Κινητών Επικοινωνιών», 2010.
- [25] Αλέξανδρος Σ. Ντάκας, «Σύστημα Καταγραφής της Εμπειρίας Κινητών Χρηστών σε Δίκτυα Ασυρμάτων Επικοινωνιών»,2011.