ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Ανάπτυξη εφαρμογής για κινητά τηλέφωνα
Friends in Range

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΛΑΜΠΡΙΑΝΙΔΗ ΓΕΩΡΓΙΟΥ

Επιβλέπων : Τιμολέων Σελλής
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2008

Η σελίδα αυτή είναι σκόπιμα λευκή.

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Ανάπτυξη εφαρμογής για κινητά τηλέφωνα
Friends in Range

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΛΑΜΠΡΙΑΝΙΔΗ ΓΕΩΡΓΙΟΥ

Επιβλέπων : Τιμολέων Σελλής
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 29η Οκτωβρίου 2008.

.....	.....	.....
Τιμολέων Σελλής	Ιωάννης Βασιλείου	Νεκτάριος Κοζύρης
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Αν. Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2008

.....

ΛΑΜΠΡΙΑΝΙΔΗΣ ΓΕΩΡΓΙΟΣ

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

Περίληψη

Ο σκοπός της διπλωματικής εργασίας ήταν η ανάπτυξη μιας εφαρμογής για κινητά τηλέφωνα ή άλλες φορητές ηλεκτρονικές συσκευές που υποστηρίζουν την πλατφόρμα Java, με σκοπό την πληροφόρηση του χρήστη για την τοποθεσία άλλων χρηστών της εφαρμογής. Το σύστημα που αναπτύχθηκε περιλαμβάνει εκτός από την εφαρμογή για τα κινητά τηλέφωνα και μια εφαρμογή εξυπηρετητή που αναλαμβάνει να συντονίζει και να παραδίδει στις εφαρμογές που τρέχουν στα κινητά τα κατάλληλα κατά περίπτωση μηνύματα. Πρωταρχικό μέλημα του συστήματος είναι η εξασφάλιση της ιδιωτικότητας των χρηστών του, έτσι παρέχει τους κατάλληλους μηχανισμούς που το πετυχαίνουν. Μόλις κάθε χρήστης ορίσει μια λίστα με έμπιστους χρήστες, με τους οποίους επιθυμεί να μοιράζεται την πληροφορία της τρέχουσας θέσης του, το σύστημα του δίνει τη δυνατότητα να εκτελεί διάφορα ερωτήματα. Υπάρχει η δυνατότητα να ενημερώνεται για την απόστασή του από κάποιον άλλο χρήστη, ή να βλέπει μια λίστα με τους πιο κοντινούς σε αυτόν χρήστες, ή τέλος να βλέπει σε ένα χάρτη την τοποθεσία κάποιου χρήστη.

Λέξεις Κλειδιά: κοντινοί φίλοι, κινητά τηλέφωνα, υπηρεσίες βασισμένες σε τοποθεσία, μοντέλο client-server, Java micro edition, GPS, εντοπισμός στίγματος

Η σελίδα αυτή είναι σκόπιμα λευκή.

Abstract

The scope of this thesis was the development of an application for mobile phones, or other electronic portable devices implementing the Java platform, which can inform the user of the location of other users of the application. The system we developed consists of two components: the client application, and the server application that coordinates the client applications by delivering appropriate messages to them. The primary objective of the system is to ensure the privacy of its users, so it provides suitable controls that allow users to achieve this. As soon as each user defines a list of trusted co-users – ie. users he allows to access his location information – the system enables him access to perform certain questions. It is possible to inquire about another user's location, to see a real time list of nearby users, or to view a user's location on a map.

Keywords: friends in range, mobile phones, location based services, client-server model, java micro edition, GPS, positioning

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πίνακας περιεχομένων

Ευρετήριο εικόνων

Ευρετήριο κώδικα

Ευρετήριο πινάκων

1 *Εισαγωγή*

1.1 Διαχείριση δεδομένων σε κινούμενα αντικείμενα - Location based services

Μια ιδιαίτερα αναπτυσσόμενη κατηγορία ηλεκτρονικών υπηρεσιών, είναι αυτή όπου το περιεχόμενο που παραδίδεται στον τελικό χρήστη τροποποιείται ανάλογα με την τοποθεσία που αυτός βρίσκεται. Πρόκειται για τις λεγόμενες location based services (LSB), οι οποίες δέχονται ως παράμετρο και την τοποθεσία του εκάστοτε χρήστη και φιλτράρουν το αποτέλεσμα έτσι ώστε αυτός να λαμβάνει μόνο τις πιο σχετικές πληροφορίες που τον αφορούν.

1.2 Αντικείμενο διπλωματικής

Στην παρούσα διπλωματική αναπτύξαμε μια εφαρμογή που επιτρέπει στους χρήστες της να ενημερώνεται ο ένας για την τοποθεσία του άλλου. Οι χρήστες έχουν τη δυνατότητα να εισάγουν σε μια λίστα με τα άτομα τα οποία επιθυμούν να μαθαίνουν τη θέση τους. Στη συνέχεια υπάρχει δυνατότητα να γίνονται ερωτήματα που ενημερώνουν ένα χρήστη για την τοποθεσία κάποιου άλλου (απόσταση και κατεύθυνση ή εικόνα στο χάρτη) ή που ενημερώνουν έναν χρήστη ποια άτομα βρίσκονται σε μια δεδομένη ακτίνα γύρω του. Στη δεύτερη περίπτωση ο χρήστης δίνει ένα χρονικό διάστημα όπου το ερώτημα αυτό είναι ενεργό και για το διάστημα αυτό υπάρχει άμεση ενημέρωση της δυναμικής λίστας των ατόμων που βρίσκονται μέσα στη δοσμένη ακτίνα. Αυτό σημαίνει ότι νέες αφίξεις εμφανίζονται αμέσως, αλλά και άτομα που βγήκαν εκτός ακτίνας διαγράφονται από τα αποτελέσματα.

1.2.1 Περιγραφή προβλήματος

Η γνώση της θέσης ενός ατόμου είναι μια πολύ σημαντική πληροφορία σε πολλές περιπτώσεις της καθημερινής μας ζωής. Ακολουθούν μερικά παραδείγματα:

- Κάποιος κάθεται σε μια καφετέρια και απολαμβάνει τον καφέ του. Τον ενδιαφέρει αν κάποιος γνωστός του βρεθεί κοντά του, για να τον πάρει τηλέφωνο και να τον συναντήσει στην καφετέρια.
- Κάποιος πάει να δει μια ταινία στον κινηματογράφο. Αν κάποιος γνωστός του τυγχάνει να περάσει κοντά από τον κινηματογράφο, μπορεί να συνεννοηθούν ώστε να δουν την ταινία μαζί.
- Δύο άτομα έχουν κανονίσει να συναντηθούν. Ο πρώτος ενδιαφέρεται να μάθει όταν ο δεύτερος που πηγαίνει να το συναντήσει έχει φτάσει αρκετά κοντά, ώστε να κατευθυνθεί στο σημείο συνάντησης.
- Ο οργανωτής μιας εκδρομής θέλει να γνωρίζει ότι όλοι οι εκδρομείς βρίσκονται κοντά του και δεν απομακρύνονται.

Ακριβώς επειδή η τρέχουσα θέση κάποιου είναι μια πάρα πολύ σημαντική πληροφορία, η εφαρμογή πρέπει να εξασφαλίσει ότι αυτή η πληροφορία είναι προσβάσιμη μόνο από τα άτομα που ο κάθε χρήστης εμπιστεύεται και έχει προσθέσει στη λίστα με τους φίλους του.

1.2.2 Σενάρια επίλυσης

Η εφαρμογή που αναπτύξαμε μπορεί να βοηθήσει σε όλες τις παραπάνω περιπτώσεις. Ο κάθε χρήστης αφού φτιάξει μια λίστα με τα άτομα που επιθυμεί να ενημερώνονται για τη τωρινή του τοποθεσία μπορεί ζητήσει από την εφαρμογή να τον ενημερώνει όταν κάποιος εισέλθει μέσα σε συγκεκριμένη και δοσμένη ακτίνα. Έτσι μπορεί να ενημερώνεται για τους φίλους του που βρίσκονται κοντά (friends in range), καθώς επίσης και να μαθαίνει πόσο μακριά από αυτόν είναι ή να βλέπει το στίγμα τους στο χάρτη. Η λίστα που δημιουργείται αρχικά είναι αυτή που διασφαλίζει ότι μόνο όσα άτομα εμφανίζονται εκεί έχουν δικαίωμα να μαθαίνουν την θέση του χρήστη. Η λίστα είναι δυναμική, δίνεται δηλαδή δυνατότητα στο χρήστη να προσθέτει και αφαιρεί φίλους κατά βούληση.

1.3 Οργάνωση κειμένου

Στο επόμενο κεφάλαιο αναλύουμε τις απαιτήσεις, τόσο σε λογισμικό αλλά και σε υλικό για τη λειτουργία του συστήματος. Δίνεται επίσης και οδηγίες για τη χρήση του. Στο 3^{ο} κεφάλαιο περιγράφουμε τη σχεδίαση του συστήματος αφού διακρίνουμε τα υποσυστήματα που το αποτελούν. Το 4^{ο} κεφάλαιο ασχολείται με θέματα της υλοποίησης, όπως αλγοριθμικά και λειτουργικά ενδιαφέροντα κομμάτια, αλλά και με τα προγραμματιστικά εργαλεία που

χρησιμοποιήθηκαν. Ακολουθεί η μεθοδολογία ελέγχου και αναλυτικά το σενάριο με το οποίο ελέγχθηκε η σωστή λειτουργία της εφαρμογής στο κεφάλαιο 5. Σχετικές εργασίες παρουσιάζονται κατά κατηγορία στο κεφάλαιο 6. Τέλος στο κεφάλαιο 7 δίνονται ενδεικτικά κάποια συμπεράσματα καθώς και ιδέες για μελλοντικές επεκτάσεις της παρούσας διπλωματικής.

Στο τελευταίο κεφάλαιο, δίνονται συνοπτικά και με μορφή οδηγιών, η πορεία που πρέπει να ακολουθήσει κάποιος για να εγκαταστήσει και να εκτελέσει επιτυχώς το σύστημα που αναπτύξαμε σε κάποιο υπολογιστικό σύστημα.

2 *Ανάλυση Απαιτήσεων Συστήματος*

Ακολουθεί η περιγραφή της αρχιτεκτονικής του συστήματος και η ανάλυση των απαιτήσεων για τη λειτουργία του.

2.1 Απαιτήσεις Συστήματος

Το σύστημα που θα υλοποιήσουμε ακολουθεί το μοντέλο client – server. Ένας κεντρικός εξυπηρετητής (server) με τον οποίο επικοινωνούν όλα τα τερματικά (κινητά τηλέφωνα), στα οποία τρέχει η εφαρμογή-πελάτη (client) αναλαμβάνει να συντονίζει αλλά και να παρέχει όλες τις πληροφορίες στους τελικούς χρήστες.

2.1.1 Το ιδανικό σύστημα

Όπως αναφέραμε και προηγουμένως το σύστημα που υλοποιήσαμε χωρίζεται σε δύο μεγάλα υποσυστήματα: τον κεντρικό εξυπηρετητή και την εφαρμογή-πελάτη που τρέχει στα κινητά τηλέφωνα.

Ιδανικά ο εξυπηρετητής είναι πάντα ενεργός ώστε να μπορεί ανά πάσα στιγμή κάποιο κινητό να συνδεθεί στο σύστημα. Θα πρέπει να έχει διαθέσιμη την απαιτούμενη μνήμη, αλλά και το απαιτούμενο εύρος ζώνης που χρειάζεται για να καλύψει πολλές ταυτόχρονες συνδέσεις από τους χρήστες της εφαρμογής-πελάτη. Τα δεδομένα που ανταλλάσσονται μεταξύ server και clients δεν είναι ογκοβόρα, επομένως η έμφαση πρέπει να δοθεί στην υποστήριξη μεγάλου αριθμού ταυτόχρονων συνδέσεων. Ανάλογα με το πλήθος των χρηστών ίσως χρειαστεί κλιμάκωση του συστήματος ώστε να είναι εύκολη και εφικτή η εξυπηρέτηση όλο και περισσότερων χρηστών. Οι διαχειριστές του συστήματος θα πρέπει να είναι σε θέση να επιβλέπουν την κατάσταση του συστήματος και να ενημερώνονται για χρήσιμα στατιστικά, όπως για παράδειγμα ποιες ώρες συνδέονται οι περισσότεροι χρήστες, ή ποιοι χρήστες είναι οι πιο ενεργοί στο σύστημα. Χρήσιμο επιπλέον είναι να υπάρχει δυνατότητα

απομακρυσμένης διαχείρισης μέσω web, καθώς και αυτοματοποιημένη αποστολή κρίσιμων μηνυμάτων προς τους διαχειριστές.

Τα κινητά τηλέφωνα στα οποία τρέχει η εφαρμογή θα πρέπει αρχικά να έχουν την απαιτούμενη μνήμη για να τρέχουν την εφαρμογή καθώς και επαρκή υπολογιστική ισχύ για ομαλή εκτέλεση. Τα περισσότερα μοντέλα που κυκλοφορούν στην αγορά είναι επαρκώς εξοπλισμένα για τις ανάγκες τις εφαρμογής.

Ένα από τα πιο σημαντικά χαρακτηριστικά για τις συσκευές που μπορούν να χρησιμοποιήσουν την υπηρεσία, είναι η δυνατότητα που έχουν να ενημερώνουν την εφαρμογή για την τρέχουσα τοποθεσία του χρήστη. Σε μια ιδανική συσκευή και υλοποίηση, θα πρέπει η εφαρμογή να μπορεί να στέλνει ανά σύντομα και τακτά χρονικά διαστήματα την τοποθεσία του χρήστη στον server χωρίς να υπάρχει παρέμβαση από το χρήστη. Ακόμη για μεγαλύτερη συμβατότητα η εφαρμογή πρέπει να μπορεί να χρησιμοποιεί πολλές διαφορετικές και ανεξάρτητες πηγές για να υπολογίσει τη θέση της στο χάρτη. Για παράδειγμα, μπορεί να χρησιμοποιηθεί κάποιο δορυφορικό σύστημα εντοπισμού θέσης, όπως το GPS, αλλά για συσκευές που αυτό δεν υποστηρίζουν είναι επιθυμητό να μπορεί να γίνει άντληση της πληροφορίας της θέσης από κάποια άλλη πηγή, όπως για παράδειγμα από την τοποθεσία της κεραία κινητής τηλεφωνίας στην οποία είναι συνδεδεμένο το κινητό. Αναλυτικά, εναλλακτικοί τρόποι εντοπισμού της θέσης αναλύονται στο κεφάλαιο 6.3. Φυσικά, συνδυασμός μεθόδων για μεγαλύτερη ακρίβεια είναι δυνατός και μπορεί να δίνεται στο χρήστη σαν επιλογή.

Τέλος, πρέπει να εξετάσουμε τις απαιτήσεις συνδεσιμότητας των κινητών με τον κεντρικό εξυπηρετητή. Η σύνδεση τους γίνεται μέσω του διαδικτύου, έτσι αυτό που έχουμε να εξασφαλίσουμε είναι ότι τα κινητά τηλέφωνα έχουν πρόσβαση στο διαδίκτυο και μπορούν να χρησιμοποιήσουν το πρωτόκολλο socket. Στην έρευνα που έγινε στην ελληνική αγορά, βρέθηκε ότι όλες οι εταιρίες κινητής τηλεφωνίας μπορούν να προσφέρουν μόνο σε πελάτες συμβολαίου το συγκεκριμένο πρόγραμμα για πρόσβαση στο διαδίκτυο. Εναλλακτικά, κινητά τηλέφωνα με ενσωματωμένη κεραία wifi, μπορούν να συνδεθούν στο σύστημα μέσω ενός ασύρματου σημείου πρόσβασης (wifi hotspot), το οποίο όμως έχει περιορισμένη εμβέλεια.

2.1.2 Το σύστημα που υλοποιήθηκε

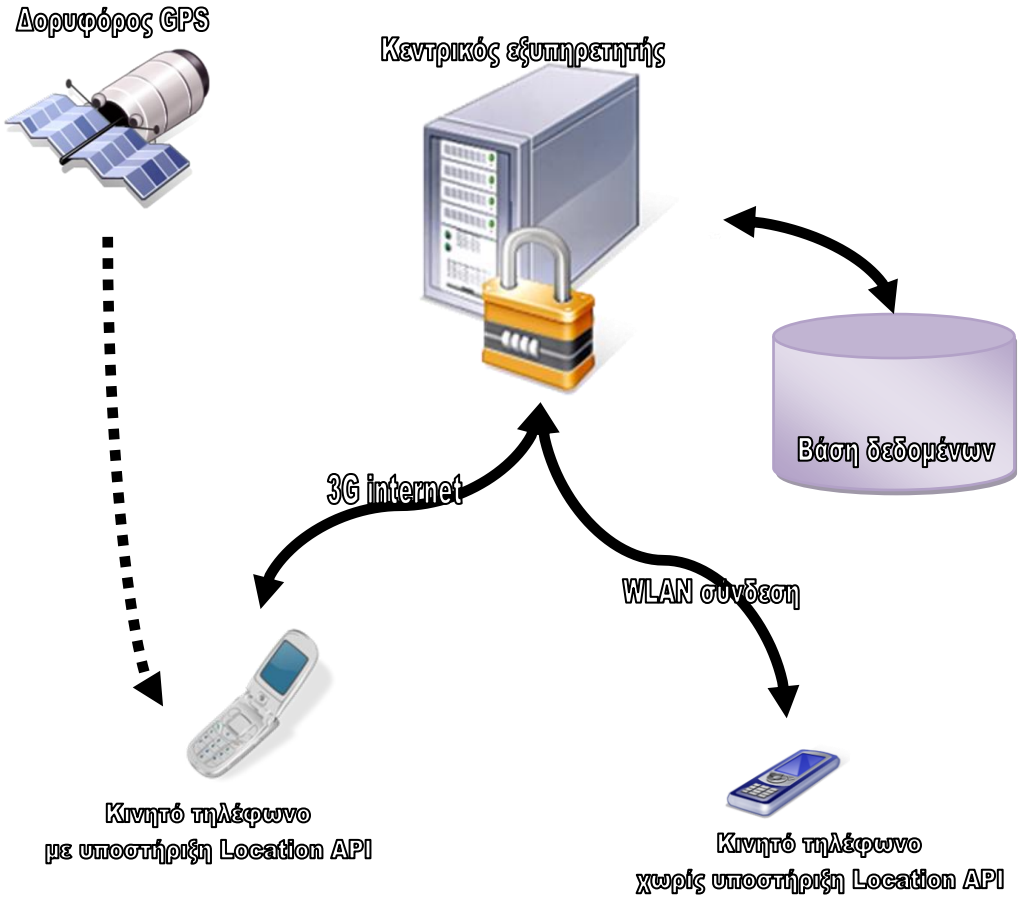
Στο πλαίσιο της συγκεκριμένης διπλωματικής υλοποιήσαμε το παρακάτω σύστημα:

Η εφαρμογή εξυπηρετητή είναι ένα Java 2 SE πρόγραμμα. Στην εφαρμογή του εξυπηρετητή για τους σκοπούς της διπλωματικής δεν ασχοληθήκαμε με το πρόβλημα της ταυτόχρονης σύνδεσης μεγάλου αριθμού χρηστών, αφού για τους σκοπούς ελέγχου είχαμε το πολύ 3 χρήστες ταυτόχρονα συνδεδεμένους. Ακόμη υλοποιήθηκε μια υποτυπώδη web διαχείριση, η οποία εύκολα μπορεί να επεκταθεί για να καλύψει τους σκοπούς που αναλύθηκαν στην

προηγούμενη υποενότητα. Το web interface είναι σελίδες jsp που τρέχουν σε έναν Apache Tomcat web server container. Υποστηρικτικά για λειτουργίες μόνιμης αποθήκευσης λειτουργεί μια βάση δεδομένων σε σύστημα mySQL.

Η εφαρμογή του προγράμματος – πελάτη υλοποιήθηκε σε Java 2 ME (microedition). Η συγκεκριμένη πλατφόρμα προτιμήθηκε για λόγους συμβατότητας με αρκετά μοντέλα της αγοράς. Απαραίτητη προϋπόθεση για να τρέχει η εφαρμογή στα κινητά τηλέφωνα λοιπόν είναι αυτά να υποστηρίζουν την πλατφόρμα Java, και συγκεκριμένα το MIDP 2.0, CLCD 1.1. Επίσης στο θέμα της εύρεσης της τοποθεσίας, η εφαρμογή χρησιμοποιεί το Java Location API, σε όσες συσκευές είναι αυτό διαθέσιμο. Σε διαφορετική περίπτωση δίνεται στο χρήστη η δυνατότητα να επιλέγει από μια λίστα προκαθορισμένων τοποθεσιών (την οποία μπορεί να τροποποιήσει) την τρέχουσα κάθε φορά τοποθεσία του.

Στην Εικόνα 2.1 βλέπουμε τα υποσυστήματα και πως αυτά επικοινωνούν μεταξύ τους:



Εικόνα 2.1. Ανάλυση του συστήματος σε υποσυστήματα και η μεταξύ τους επικοινωνία.

2.2 Περιγραφή Συστημάτων

Περιγράφονται αναλυτικά παρακάτω οι λειτουργίες που εκτελεί κάθε υποσύστημα.

2.2.1 Εφαρμογή client

Η εφαρμογή επιτρέπει στους χρήστες να συνδέονται στο σύστημα και να χρησιμοποιούν την υπηρεσία. Συγκεκριμένα μπορούν:

- Να δημιουργήσουν μια λίστα φίλων. Η λίστα αυτή περιέχει άτομα τα οποία ο χρήστης μπορεί να βλέπει την τοποθεσία τους, αλλά και αυτά έχουν δικαίωμα και δυνατότητα να βλέπουν τη θέση του. Τα άτομα που εμφανίζονται σε αυτή τη λίστα μπορούν να βρίσκονται σε μία από τις ακόλουθες τρεις καταστάσεις: "accepted", "pending", "blocked". Ο χρήστης της εφαρμογής έχει τη δυνατότητα να τροποποιεί τη λίστα και να:
    - προσθέτει νέους φίλους. Είτε από τον τηλεφωνικό κατάλογο της συσκευής, είτε χειροκίνητα, πληκτρολογώντας ο ίδιος όνομα και τηλεφωνικό νούμερο. Αρχικά οι φίλοι που προστίθενται εισάγονται ως “pending”. Όταν ο φίλος αποδεχθεί τη σχέση τότε η κατάστασή τους αλλάζει σε “accepted”.
    - διαγράφει υπάρχοντες φίλους από τη λίστα και έτσι χάνει το δικαίωμα να ενημερώνεται για τη θέση τους αλλά και αυτοί για τη δική του.
    - αποκλείσει προσωρινά κάποιο φίλο ή να άρει τον αποκλεισμό κατά περίπτωση. Ο αποκλεισμός λειτουργεί όπως και η διαγραφή (είναι δηλαδή αμοιβαίος) απλά αφήνει την καταχώρηση στη λίστα, σε κατάσταση “blocked”, έτσι ώστε να μπορέσει ο χρήστης εύκολα να αποκαταστήσει τη σχέση. Η άρση του αποκλεισμού μπορεί να γίνει μόνο από τον χρήστη που τον επέβαλλε αρχικά.
- Να ενημερώνεται για την απόσταση από και κατεύθυνση προς ένα φίλο του.
- Να βλέπει τη θέση ενός φίλου του στο χάρτη (σε κινητό με ενσωματωμένο web browser).
- Να ορίσει μια ακτίνα και να ενημερώνεται για το ποια άτομα από αυτά που βρίσκονται στη λίστα φίλων του βρίσκονται σε απόσταση από αυτόν μικρότερη από την δοσμένη ακτίνα.

Ας δούμε αναλυτικά όλες τις λειτουργίες της εφαρμογής. Τα screenshots που ακολουθούν είναι από το κινητό τηλέφωνο Nokia N80 με λειτουργικό Symbian OS 9.1 3rd edition.

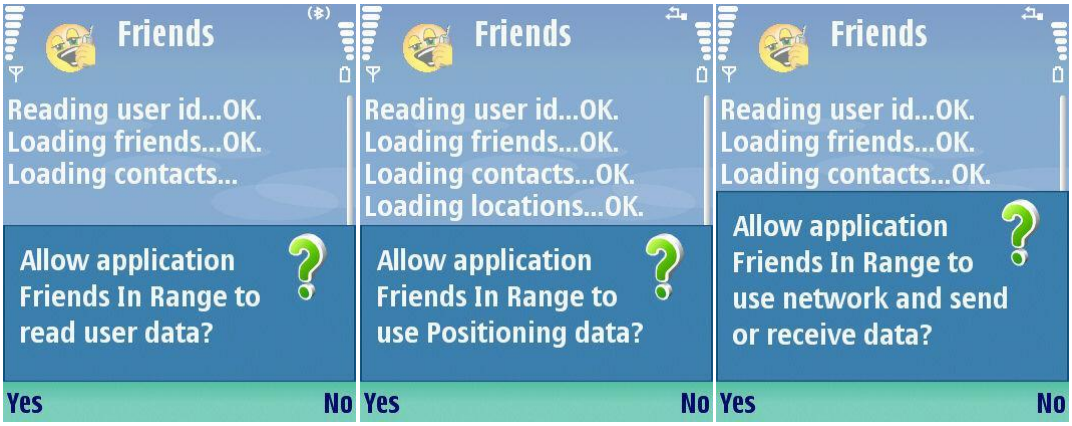
2.2.1.1 Εκκίνηση εφαρμογής.

Όταν αρχικά εκκινήσει η εφαρμογή εμφανίζεται μια οθόνη όπου βλέπουμε πληροφορίες σχετικά με την αρχικοποίησή της. Κατά την πρώτη εκκίνηση, η εφαρμογή ζητάει από το χρήστη να εισάγει τον αριθμό του τηλεφώνου του, μιας και αυτό αποτελεί το μοναδικό χαρακτηριστικό κάθε χρήστη στο σύστημα. Για να επιβεβαιώσει η εφαρμογή ότι ο χρήστης

έδωσε το σωστό αριθμό, επιχειρεί να στείλει ένα γραπτό μήνυμα SMS στον αριθμό αυτό και να λάβει το ίδιο μήνυμα ως επιβεβαίωση ότι όντως ο αριθμός που εισήχθη είναι ο σωστός.

Εικόνα 2.2. Η ακολουθία επικύρωσης του αριθμού τηλεφώνου του χρήστη.
Γίνεται μόνο μια φορά, κατά την πρώτη εκτέλεση της εφαρμογής.

Στη συνέχεια ακολουθεί μια λίστα από μηνύματα αρχικοποίησης, συνοδευόμενα από μια σειρά ερωτήσεων προς το χρήστη, που εξαρτώνται από το επίπεδο ασφαλείας που έχει οριστεί για την πλατφόρμα Java από το λειτουργικό του κινητού τηλεφώνου.



Εικόνα 2.3. Η ακολουθία αρχικοποίησης της εφαρμογής.
Η παραπάνω οθόνη και οι ερωτήσεις εμφανίζονται σε κάθε εκκίνηση της εφαρμογής.

Στην Εικόνα 2.3.α η εφαρμογή ζητάει την άδεια του χρήστη για να διαβάσει τον τηλεφωνικό κατάλογο της συσκευής. Εάν ο χρήστης απαντήσει καταφατικά, τότε γίνεται προσπάθεια ανάγνωσης του τηλεφωνικού καταλόγου, συγκεκριμένα των καταχωρήσεων με αριθμό κινητού τηλεφώνου. Στην συνέχεια όταν ο χρήστης θελήσει να προσθέσει κάποιο γνωστό του στη λίστα φίλων του, μπορεί να επιλέξει από τη λίστα που έχει δημιουργηθεί από τον τηλεφωνικό κατάλογο. Εάν ο χρήστης απαντήσει αρνητικά, τότε μπορεί να προσθέτει νέους φίλους μόνο χειροκίνητα, δηλαδή πληκτρολογώντας ο ίδιος το όνομα και τον αριθμό του φίλου που θέλει να προσθέσει.

Στην Εικόνα 2.3.β. η εφαρμογή ζητάει την άδεια του χρήστη να χρησιμοποιήσει υπηρεσίες εντοπισμού θέσεις. Η συγκεκριμένη ερώτηση εμφανίζεται μόνο σε κινητά τηλέφωνα που

υποστηρίζουν το Java Location API, και μόνο σε αυτά υπάρχει δυνατότητα αυτόματης ενημέρωσης της θέσης του κινητού. Εδώ ο χρήστης μπορεί να απαντήσει θετικά, οπότε το πρόγραμμα θα προσπαθήσει να χρησιμοποιήσει κάποια από τις διαθέσιμες μεθόδους εύρεσης θέσης, για να λάβει την απαιτούμενη πληροφορία. Συνήθως επιχειρείται η σύνδεση με κάποιο ενσωματωμένο ή εξωτερικό δέκτη GPS για λήψη συντεταγμένων. Όταν δεν υπάρχει ενσωματωμένος δέκτης στη συσκευή, ενδέχεται να γίνει προτροπή στο χρήστη για ενεργοποίηση της κεραίας Bluetooth ώστε να γίνει σύνδεση με εξωτερικό δέκτη. Αν ο χρήστης απαντήσει αρνητικά τότε ενεργοποιείται η χειροκίνητη ρύθμιση θέσης. Υπάρχει μια λίστα με προκαθορισμένες θέσεις που ο χρήστης μπορεί να τροποποιήσει. Εκτός από τις τοποθεσίες που μπορεί να προσθέσει ο χρήστης, υπάρχει πάντα και η τοποθεσία με όνομα "Home", η οποία ενεργοποιείται αυτόματα, όταν επιλέξουμε χειροκίνητη ρύθμιση θέσης. Μετά τη λήψη των συντεταγμένων ανεξαρτήτως τρόπου, γίνεται αποστολή αυτών στον κεντρικό εξυπηρετητή.

Η τελευταία ερώτηση (Εικόνα 2.3.γ) ουσιαστικά ζητάει από το χρήστη να επιτρέψει την σύνδεση του προγράμματος στο διαδίκτυο έτσι ώστε να μπορέσει να συνδεθεί με τον κεντρικό εξυπηρετητή. Εδώ χρειάζεται οπωσδήποτε θετική απάντηση για να συνεχίσουμε να εκτελούμε την εφαρμογή, αφού αν δεν επιτευχθεί σύνδεση, η εφαρμογή δεν έχει νόημα να εκτελείται. Το λειτουργικό του τηλεφώνου είναι στη συνέχεια υπεύθυνο να αναζητήσει διαθέσιμους τρόπους σύνδεσης με το διαδίκτυο και να πραγματοποιήσει τη σύνδεση. Αν όλα πάνε καλά, τότε βλέπουμε την παρακάτω οθόνη (Εικόνα 2.4), που είναι και η κεντρική οθόνη της εφαρμογής.

Εικόνα 2.4. Η κεντρική οθόνη της εφαρμογής-πελάτη.

2.2.1.2 Κεντρικό μενού.

Ας δούμε συνοπτικά τις επιλογές που μας δίνει το κεντρικό μενού της εφαρμογής και παρακάτω θα αναλύσουμε με λεπτομέρεια την κάθε επιλογή.

- "**Nearby friends**". Ο χρήστης ορίζει μια ακτίνα, και στη συνέχεια να βλέπει ποιοι από τους φίλους του βρίσκονται σε απόσταση μικρότερη από την δοσμένη ακτίνα.

- "**All friends**". Ο χρήστης διαχειρίζεται τη λίστα φίλων του. Προσθέτει, διαγράφει ή προσωρινά αποκλείει κάποιους από τους φίλους, αλλά ενημερώνεται και για την κατάσταση κάθε φίλου καθώς επίσης και για το πόσο μακριά βρίσκονται.
- "**My location**". Ο χρήστης ορίζει την τοποθεσία που βρίσκεται, είτε από μια λίστα με προεπιλεγμένες τοποθεσίες είτε θέτοντας την επιλογή στην αυτόματη λειτουργία οπότε το πρόγραμμα λαμβάνει την τρέχουσα τοποθεσία από εξωτερική πηγή, όπως για παράδειγμα ένας δέκτης GPS.
- "**Unsubscribe**". Δίνει τη δυνατότητα στο χρήστη να διαγραφεί από το σύστημα.
- "**About**". Πληροφορίες για την έκδοση και άλλα στοιχεία για την εφαρμογή.

2.2.1.3 Διαχείριση λίστας φίλων.

Ας ξεκινήσουμε την αναλυτική περιγραφή ακολουθώντας την 2η επιλογή του κεντρικού μενού "All friends" αφού αυτή θα μας επιτρέψει να δημιουργήσουμε τη λίστα φίλων που είναι απαραίτητη για την περαιτέρω λειτουργικότητα της εφαρμογής, δηλαδή τη δημιουργία ερωτημάτων σχετικά με τις θέσεις των φίλων μας.

Εικόνα 2.5. Ο John είναι accepted, η Helen είναι blocked, ενώ ο Tim είναι pending.

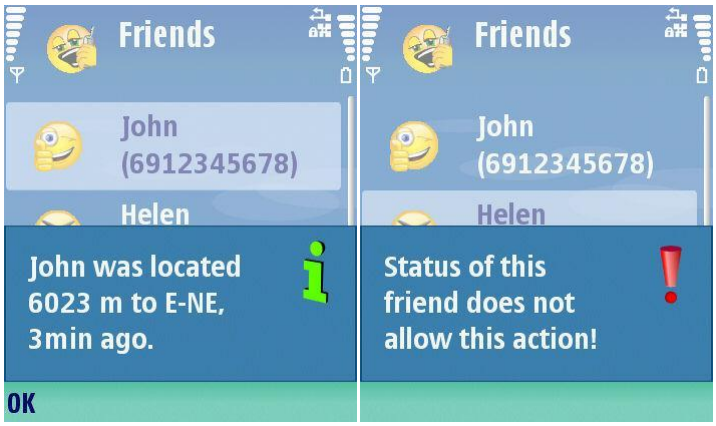
Ακολουθώντας την επιλογή αυτή εμφανίζεται μια λίστα, με τους φίλους που έχει προσθέσει ο χρήστης. Κάθε φίλος αποτελείται από το όνομα και τον αριθμό του τηλεφώνου του σε παρενθέσεις και μπορεί να βρίσκεται σε μία από τις ακόλουθες τρεις καταστάσεις: "accepted", "pending", "blocked". Η κατάσταση ενός φίλου γίνεται αντιληπτή από το εικονίδιο που εμφανίζεται δίπλα στο όνομα του. Η αντιστοιχία φαίνεται στην Εικόνα 2.5.

Είναι δυνατόν να υπάρχουν δύο καταχωρήσεις με το ίδιο όνομα, αλλά όχι με τον ίδιο αριθμό τηλεφώνου. Στο όνομα κάθε φίλου απαγορεύεται να υπάρχουν οι χαρακτήρες των παρενθέσεων.

Οι φίλοι που εμφανίζονται στη λίστα είναι αυτοί για τους οποίους έχουμε εκδηλώσει ενδιαφέρον να μαθαίνουμε τη θέση τους. Αν αυτοί αποδεχτούν το αίτημα, ταυτόχρονα αποκτούν τη δυνατότητα να ενημερώνονται για τη δική μας θέση.

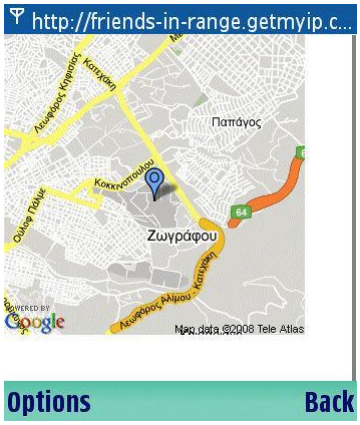
Από το μενού “Options” έχουμε τις εξής επιλογές:

- **Show distance**: εμφανίζει πόσο μακριά από μας βρίσκεται ο επιλεγμένος φίλος και σε ποια κατεύθυνση. Φυσικά, πληροφορίες δίνονται μόνο για τους φίλους που βρίσκονται στην κατάσταση “accepted”. Η πληροφορία εμφανίζεται στη μορφή: “[name] was located [distance]m to [direction], [time]min ago.”, όπου name είναι το όνομα του επιλεγμένου φίλου, distance είναι η απόσταση σε μέτρα μεταξύ του χρήστη και του φίλου του, direction είναι η πορεία που πρέπει να ακολουθήσουμε για να τον συναντήσουμε και time είναι ο χρόνος που έχει περάσει από τη στιγμή που ο φίλος ενημέρωσε τον κεντρικό server για τη θέση του.



Εικόνα 2.6. Επειδή ο John είναι accepted, εμφανίζεται η πληροφορία για τη θέση του, ενώ για την Helen που είναι μπλοκαρισμένη, η πληροφορία είναι μη διαθέσιμη.

- **Show on map**: (λειτουργεί σε κινητά τηλέφωνα με ενσωματωμένο browser). Η λειτουργία αυτή ουσιαστικά ανοίγει ένα συγκεκριμένο URL στον browser της συσκευής και εμφανίζει στο χάρτη τη θέση του φίλου (χρησιμοποιεί τη δωρεάν υπηρεσία Χάρτες Google, και το Google Maps Static API). Ανάλογα με τη συσκευή η δυνατότητα αυτή εμφανίζεται με δύο τρόπους: το λειτουργικό εκκινεί τον browser και εμφανίζει τη συγκεκριμένη σελίδα είτε αμέσως ενώ η εφαρμογή εκτελείται στο παρασκήνιο, είτε μόλις κλείσουμε την εφαρμογή. Στην πρώτη περίπτωση όταν κλείσουμε τον browser η εφαρμογή επανέρχεται στο προσκήνιο. Η υπηρεσία Google Maps για να εμφανίσει ένα συγκεκριμένο σημείο στο χάρτη χρειάζεται τις συντεταγμένες του. Αναλυτικότερα για το πώς λειτουργεί η συγκεκριμένη λειτουργία στην ενότητα που περιγράφει τον server και τον web server.

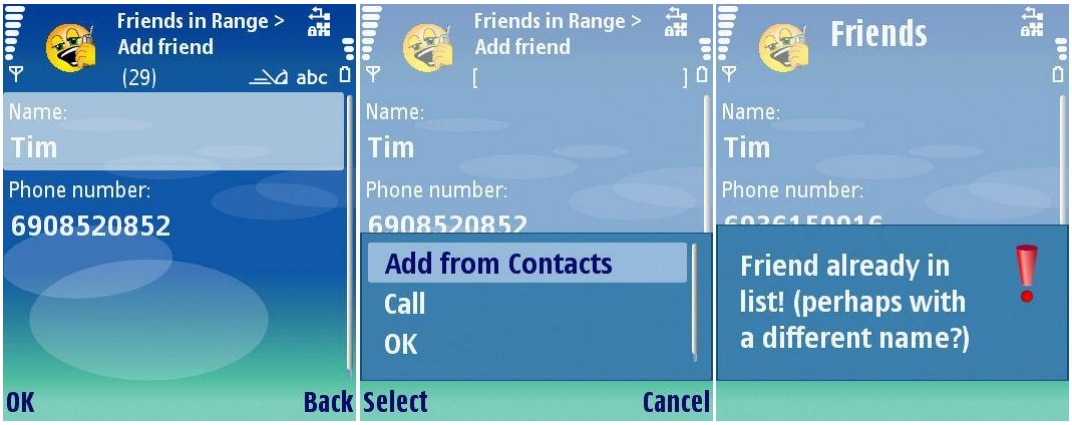


Εικόνα 2.7. Προβολή χάρτη θέσης φίλου στο κινητό.

- **Add from contacts**: Εάν κατά την αρχικοποίηση της εφαρμογής, έχουμε επιτρέψει την ανάγνωση του καταλόγου των επαφών της συσκευής, τότε αυτή η επιλογή εμφανίζει μια λίστα με όλες τις επαφές που έχουν καταχωρημένο αριθμού κινητού τηλεφώνου (πρέπει να υπάρχει τουλάχιστον μία καταχώρηση τύπου mobile phone). Διαφορετικά η οθόνη αυτή είναι μια κενή λίστα. Όπως φαίνεται και στην Εικόνα 2.8, υπάρχει δυνατότητα πολλαπλής επιλογής και στη συνέχεια επιλέγοντας OK, οι επιλεγμένες επαφές εισάγονται ως φίλοι "pending".

Εικόνα 2.8. Στα αριστερά φαίνεται μέρος της λίστας επαφών της συσκευής από όπου έχουμε επιλέξει δύο, και δεξιά η λίστα φίλων όπου οι προηγουμένως επιλεγμένες επαφές έχουν προστεθεί σε κατάσταση "pending".

- **Add**: Εάν κατά την αρχικοποίηση της εφαρμογής δεν έχουμε επιτρέψει την ανάγνωση του καταλόγου των επαφών της συσκευής, ή αν αυτή δεν υποστηρίζεται τότε αυτή η επιλογή μας επιτρέπει να προσθέσουμε φίλους στη λίστα, πληκτρολογώντας το όνομα και το τηλέφωνό τους (Εικόνα 2.9.α). Κατά την εισαγωγή του αριθμού τηλεφώνου, η υλοποίηση δίνει ακόμη μια φορά τη δυνατότητα στο χρήστη να αντιγράψει τον αριθμό από κάποια επαφή αποθηκευμένη στον κατάλογο της συσκευής (Εικόνα 2.9.β). Αν ο χρήστης πληκτρολογήσει αριθμό τηλεφώνου που υπάρχει ήδη στη λίστα φίλων, τότε εμφανίζεται προειδοποιητικό μήνυμα και η εισαγωγή ακυρώνεται (Εικόνα 2.9.γ).



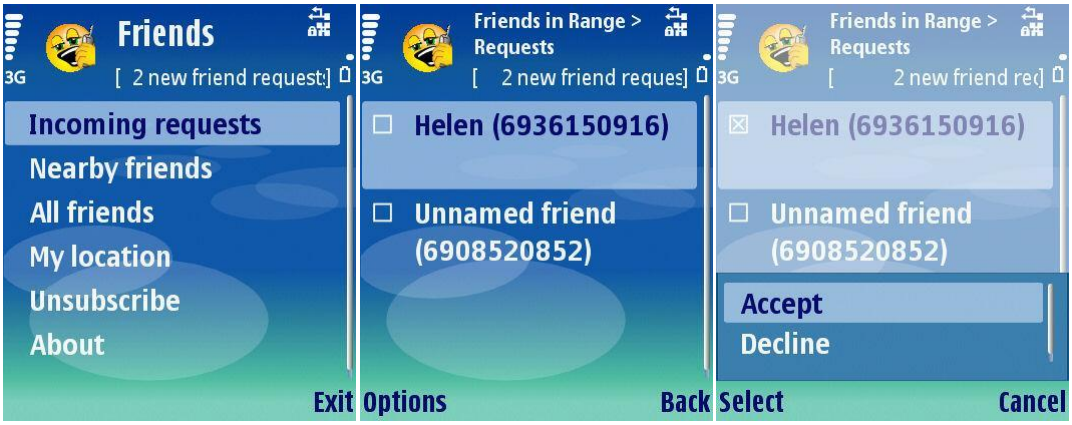
Εικόνα 2.9. Προσθήκη νέου φίλου με πληκτρολόγηση στοιχείων.

- **Remove**: Επιτρέπει την κατάργηση ενός φίλου από τη λίστα. Αν ο χρήστης Α καταργήσει τον χρήστη Β από τη λίστα φίλων του, τότε ο Β στη δικιά του λίστα φίλων βλέπει τον Α σε κατάσταση "pending".
- **Block**: Ο χρήστης αποτρέπει προσωρινά κάποιον από τους φίλους του να ενημερώνεται για τη θέση του, θέτοντας την κατάσταση του φίλου του σε "blocked". Ο αποκλεισμός είναι αμοιβαίος, δηλαδή όταν ο χρήστης Α μπλοκάρει τον Β, τότε κανένας από τους δυο δε μπορεί να λαμβάνει πληροφορίες για τη θέση του άλλου. Και στους δύο χρήστες το εικονίδιο του αντίστοιχου φίλου αλλάζει για να υποδεικνύει τη νέα κατάσταση.
- **Unblock**: Με αυτή την επιλογή γίνεται άρση του αποκλεισμού ενός φίλου και επαναφορά του στην κατάσταση "accepted". Μόνο ο χρήστης που είχε ορίσει αρχικά τον αποκλεισμό έχει τη δυνατότητα να τον άρει. Για παράδειγμα, αν ο Α μπλοκάρει τον Β, τότε μόνο ο Α μπορεί αργότερα να άρει τον αποκλεισμό.
- **Edit**: Δίνει την δυνατότητα στο χρήστη να αλλάζει το εμφανιζόμενο όνομα του φίλου του. Το όνομα ενός φίλου δεν μπορεί να περιέχει τους χαρακτήρες των παρενθέσεων. Υπενθυμίζεται ότι είναι δυνατόν δύο φίλοι να έχουν το ίδιο όνομα, αφού αυτό που τους ξεχωρίζει μοναδικά είναι ο αριθμός του τηλεφώνου τους.
- **Resend request**: Η λειτουργία αυτή χρησιμοποιείται μόνο σε φίλους που βρίσκονται στην κατάσταση "pending". Η λειτουργία του είναι να ξαναστείλει το αίτημα πρόσκληση στον παραλήπτη. Για παράδειγμα αν κάποιος στη λίστα φίλων είναι σε κατάσταση "pending" για μεγάλο χρονικό διάστημα, η επιλογή αυτή ξαναστέλνει την πρόσκληση στον παραλήπτη για να του θυμίσει ότι επιθυμούμε να τον προσθέσουμε στη λίστα φίλων και δεν μας έχει δεχτεί ακόμα.

2.2.1.4 Εισερχόμενα αιτήματα.

Η διαδικασία της δημιουργίας της λίστας φίλων ολοκληρώνεται με αναφορά στις εισερχόμενες αιτήσεις για νέους φίλους. Αυτό είναι επακόλουθο της αποστολής πρόσκλησης

για προσθήκη ενός νέου φίλου. Σε ένα τυπικό σενάριο χρήσης ο χρήστης Α επιθυμεί να προσθέσει τον χρήστη Β στη λίστα φίλων του. Αυτό το επιτυγχάνει σύμφωνα με την παραπάνω διαδικασία επιλέγοντας "Add" ή "Add from Contacts" από το μενού. Ο χρήστης Β άμεσα (ή αν δεν είναι συνδεδεμένος μόλις εισέλθει) λαμβάνει ειδοποίηση για την νέα εισερχόμενη πρόσκληση και έχει τη δυνατότητα να την αποδεχτεί ή να την αρνηθεί. Μόλις ληφθεί ένα νέο αίτημα εμφανίζεται μήνυμα στο χώρο ειδοποιήσεων της εφαρμογής και στο κεντρικό μενού προστίθεται μια καινούρια επιλογή "Incoming requests" (Εικόνα 2.10.α). Από εκεί ο χρήστης μπορεί να διαχειριστεί τις νέες προσκλήσεις που του στέλνονται. Στη λίστα με τις εισερχόμενες αιτήσεις ο χρήστης επιλέγει αυτές που θέλει να αποδεχτεί ή να απορρίψει και ενεργοποιεί την αντίστοιχη εντολή από το μενού (Εικόνα 2.10.γ).



Εικόνα 2.10. Εισερχόμενα αιτήματα για νέους φίλους και διεκπεραίωσή τους.

Εάν ο χρήστης που αποστέλλει την αίτηση υπάρχει στον κατάλογο επαφών της συσκευής (ο αριθμός τηλεφώνου του) τότε στη λίστα εμφανίζεται το καταχωρημένο όνομα δίπλα στον αριθμό. Σε διαφορετική περίπτωση εμφανίζεται το νούμερο του αποστολέα με το όνομα "Unnamed friend" (Εικόνα 2.10.β). Το ίδιο συμβαίνει και όταν κατά την εκκίνηση της εφαρμογής δεν επιτρέψουμε την ανάγνωση του καταλόγου της συσκευής. Τότε όλες οι εισερχόμενες αιτήσεις θα συνοδεύονται από το όνομα "Unnamed friend".

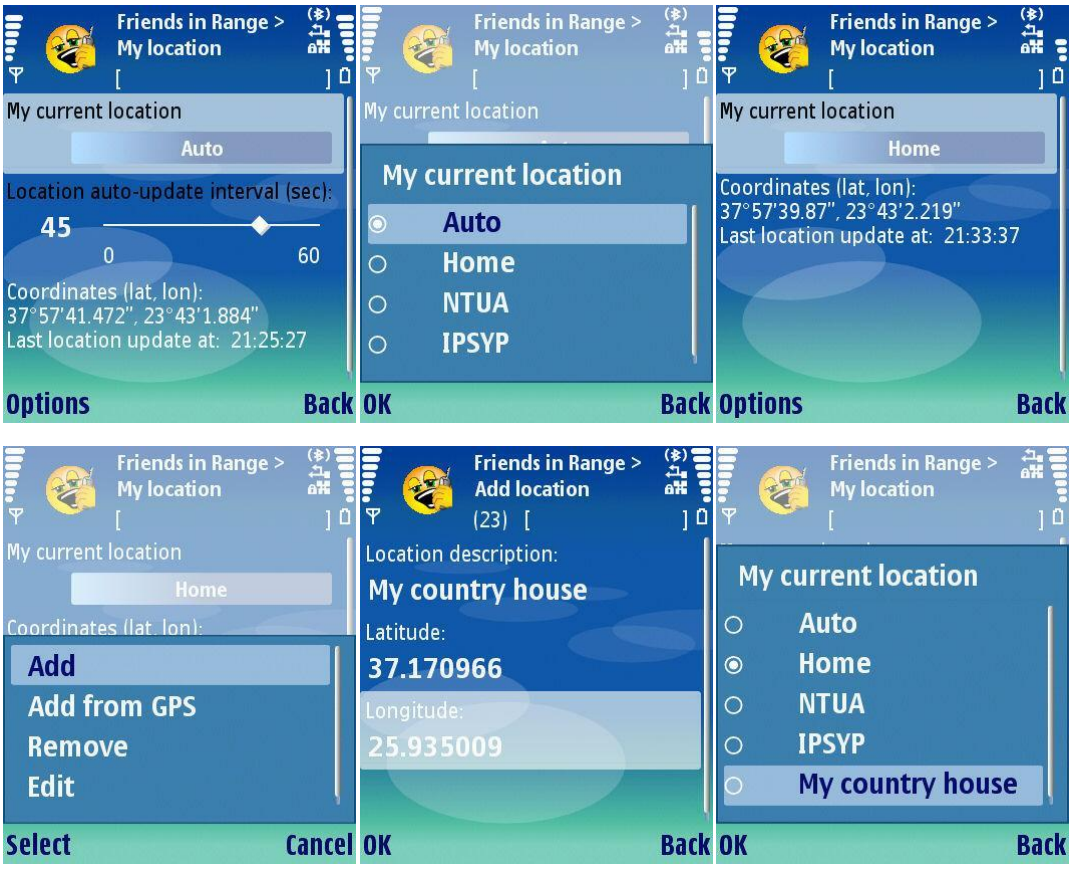
Όταν μια αίτηση γίνεται αποδεκτή, ο αντίστοιχος φίλος απομακρύνεται από τη λίστα εισερχόμενων αιτήσεων και εισάγεται στην λίστα φίλων σε κατάσταση "accepted". Οι αιτήσεις που απορρίπτονται απλά διαγράφονται από τη λίστα εισερχομένων χωρίς κάποια άλλη επεξεργασία. Μόλις όλες οι εισερχόμενες αιτήσεις έχουν διεκπεραιωθεί το μενού "Incoming requests" εξαφανίζεται από το κεντρικό μενού.

2.2.1.5 Ρύθμιση θέσης.

Από την τρίτη επιλογή του κεντρικού μενού (Εικόνα 2.4) "**My location**" γίνεται η ρύθμιση της θέσης του κινητού. Όπως έχουμε αναφέρει πρωτύτερα στην τρέχουσα υλοποίηση υπάρχουν δύο διαφορετική τρόποι ενημέρωση της θέσης του κινητού: είτε αυτόματα

κάνοντας χρήση του Java Location API για όσα κινητά το υποστηρίζουν είτε μη αυτόματα επιλέγοντας κάθε φορά από μια λίστα με προκαθορισμένες τοποθεσίες. Στην πρώτη περίπτωση απαιτείται επιπλέον εξοπλισμός, όπως για παράδειγμα ένας εσωτερικός ή εξωτερικός δέκτης GPS.

Στην οθόνη "My location" από το drop-down μενού "My current location" (Εικόνα 2.11.β) μπορούμε να επιλέγουμε μεταξύ των δύο προαναφερθέντων λειτουργιών. Η επιλογή "Auto" ενεργοποιεί την αυτόματη λειτουργία και το κινητό επιχειρεί να εντοπίσει τη θέση του μέσω του Location API. Κάτω από το μενού "My current location" υπάρχει μια μπάρα την οποία μπορεί να μετακινεί ο χρήστης για ρυθμίσει κάθε πότε το κινητό θα προσπαθεί να εντοπίσει τη θέση του και να τη στείλει στον κεντρικό server. Οι επιτρεπτές τιμές κυμαίνονται από 5 μέχρι 60 δευτερόλεπτα, ενώ η προεπιλεγμένη τιμή είναι 45 δευτερόλεπτα. Ακριβώς από κάτω για πληροφοριακούς λόγους εμφανίζονται οι τρέχουσες συντεταγμένες του κινητού (Εικόνα 2.11.α).



Εικόνα 2.11. Ρυθμίσεις θέσης.

Επιλέγοντας οποιαδήποτε άλλη τοποθεσία από το μενού "My current location" εκτός από το "Auto" απενεργοποιεί την αυτόματη ρύθμιση (εξαφανίζει και τη μπάρα ελέγχου που ρυθμίζει τη χρονική περίοδο αναζήτησης θέσης) και θέτει την τρέχουσα θέση στην επιλεγμένη (Εικόνα 2.11.γ) στέλνοντας ταυτόχρονα τις νέες συντεταγμένες στον κεντρικό εξυπηρετητή. Στις συσκευές όπου το Location API δεν υποστηρίζεται η επιλογή "Auto" δεν εμφανίζεται.

Δίνεται βέβαια η δυνατότητα στο χρήστη μέσα από το μενού Options να τροποποιήσει όπως επιθυμεί τη λίστα με τις διαθέσιμες τοποθεσίες, προσθέτοντας αυτές που εκείνος συχνάζει (Εικόνα 2.11.δ-στ). Η διαφορά μεταξύ των επιλογών “Add” και “Add from GPS” είναι ότι στη μεν πρώτη πληκτρολογούμε όλα τα στοιχεία της νέας τοποθεσίας, δηλαδή περιγραφή και συντεταγμένες, ενώ στη δεύτερη οι συντεταγμένες αντιγράφονται αυτόματα από την ένδειξη του GPS. Ο μόνος περιορισμός είναι ότι πάντα πρέπει να υπάρχει μία τοποθεσία με το όνομα Home.

2.2.1.6 Ερωτήματα θέσης.

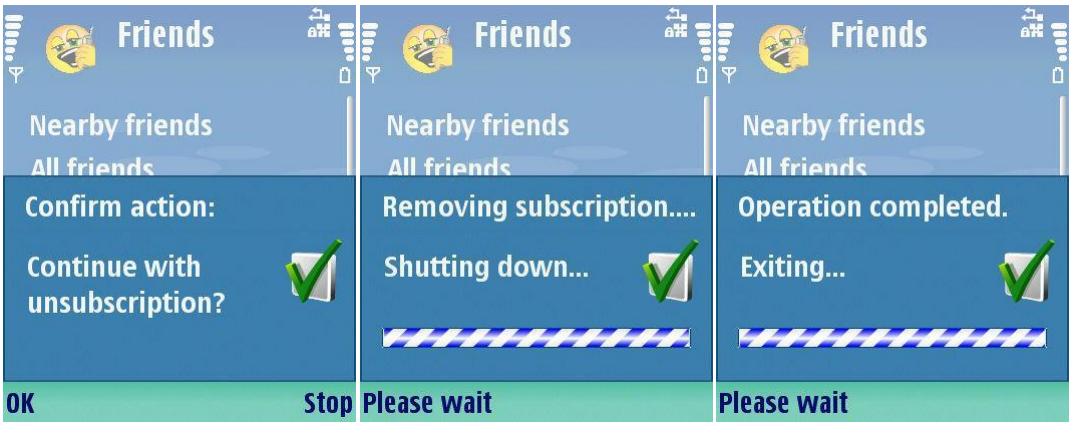
Η κύρια λειτουργία της εφαρμογής είναι να απαντά σε ερωτήματα σχετικά με τη θέση των ατόμων που βρίσκονται στη λίστα φίλων (μόνο όσων είναι accepted). Από την επιλογή “Nearby friends” του κεντρικού μενού μπορούμε να ορίσουμε μια ακτίνα (Εικόνα 2.12.α) και η εφαρμογή να μας ειδοποιεί όταν κάποιος από τη accepted φίλος εισέρχεται σε νοητό κύκλο με κέντρο εμάς και τη δοσμένη ακτίνα. Εκτός από την ακτίνα μπορούμε να επιλέξουμε και ένα χρονικό διάστημα για το οποίο θα είναι ενεργή η οθόνη αποτελεσμάτων των κοντινών φίλων (Εικόνα 2.12.β). Αν επιλέξουμε “never” τότε η εφαρμογή θα μας ενημερώνει για κοντινούς φίλους μέχρι να επιλέξουμε “Stop”. Σε διαφορετική περίπτωση μετά τη λήξη του επιλεγμένου χρόνου η οθόνη των αποτελεσμάτων παύει να είναι διαθέσιμη.

Εικόνα 2.12. α,β: ρυθμίσεις / γ,ε εμφάνιση κοντινών φίλων / δ,στ πληροφορίες απόστασης.

Στην Εικόνα 2.12.γ βλέπουμε μια τυπική οθόνη αποτελεσμάτων. Η ακτίνα έχει οριστεί στα 500 μέτρα (Εικόνα 2.12.α). Πατώντας πάνω σε κάποιον φίλο η εφαρμογή μας ενημερώνει για την απόσταση του (Εικόνα 2.12.δ), εμφανίζοντας ακριβώς τις ίδιες πληροφορίες όπως η επιλογή "Show distance" από την οθόνη "All friends". Όπως βλέπουμε η Helen απέχει λιγότερο από 500 μέρα. Αργότερα από την οθόνη αποτελεσμάτων εξαφανίζεται ο Tim (Εικόνα 2.12.ε). Πηγαίνοντας "Back" και μετά στο μενού "All friends", κάνουμε ερώτηση για τη θέση του και βλέπουμε ότι όντως είναι εκτός ακτίνας (Εικόνα 2.12.στ).

2.2.1.7 Διαγραφή από την υπηρεσία.

Η εγγραφή του χρήστη γίνεται αυτόματα απλά με την εισαγωγή του αριθμού τηλεφώνου του κατά την πρώτη εκκίνηση της εφαρμογής. Αν ένας χρήστης επιθυμεί να διαγραφεί από την υπηρεσία ώστε να μην υπάρχει καμία καταχώρηση στο κεντρικό εξυπηρετητή για τον αριθμό του, εκτελεί την εντολή "Unsubscribe" από το κεντρικό μενού.



Εικόνα 2.13. Διαγραφή από την υπηρεσία.

Η εντολή αυτή διαγράφει

- όλες τις καταχωρήσεις στη λίστα φίλων της εφαρμογής,
- τον αποθηκευμένο αριθμό τηλεφώνου που αποτελεί το μοναδικό χαρακτηριστικό κάθε χρήστη στο σύστημα (σε επόμενη εκκίνηση ζητείται πάλι εισαγωγή και επιβεβαίωση ορθότητας του αριθμού),
- κάθε καταχώρηση του αριθμού του χρήστη από τον κεντρικό εξυπηρετητή,

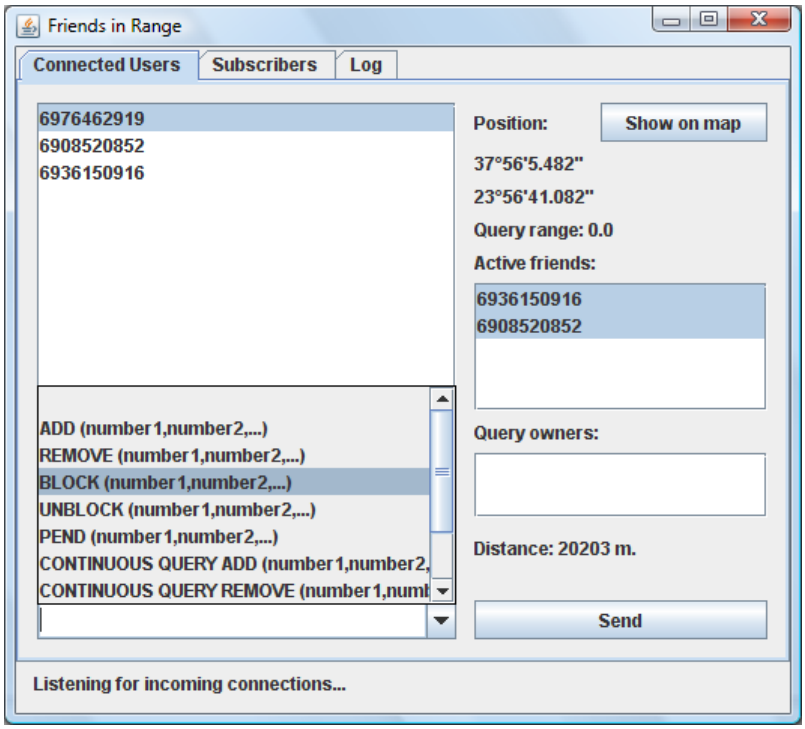
και τέλος κλείνει την εφαρμογή. Όλοι οι φίλοι του χρήστη που διαγράφεται τον βλέπουν πλέον στις λίστες τους ως "pending".

2.2.1.8 Πληροφορίες.

Από την τελευταία επιλογή "About" μπορούμε να δούμε διάφορες πληροφορίες όπως τους δημιουργούς της εφαρμογής.

2.2.2 *Εφαρμογή server*

Η εφαρμογή του κεντρικού εξυπηρετητή, ουσιαστικά συντονίζει και επιτρέπει την επικοινωνία των εφαρμογών στα κινητά τηλέφωνα, παραδίδοντας κατά περίπτωση τα κατάλληλα μηνύματα. Στην Εικόνα 2.14 φαίνεται η κεντρική οθόνη.



Εικόνα 2.14. Η κεντρική οθόνη της εφαρμογής-εξυπηρετητή.

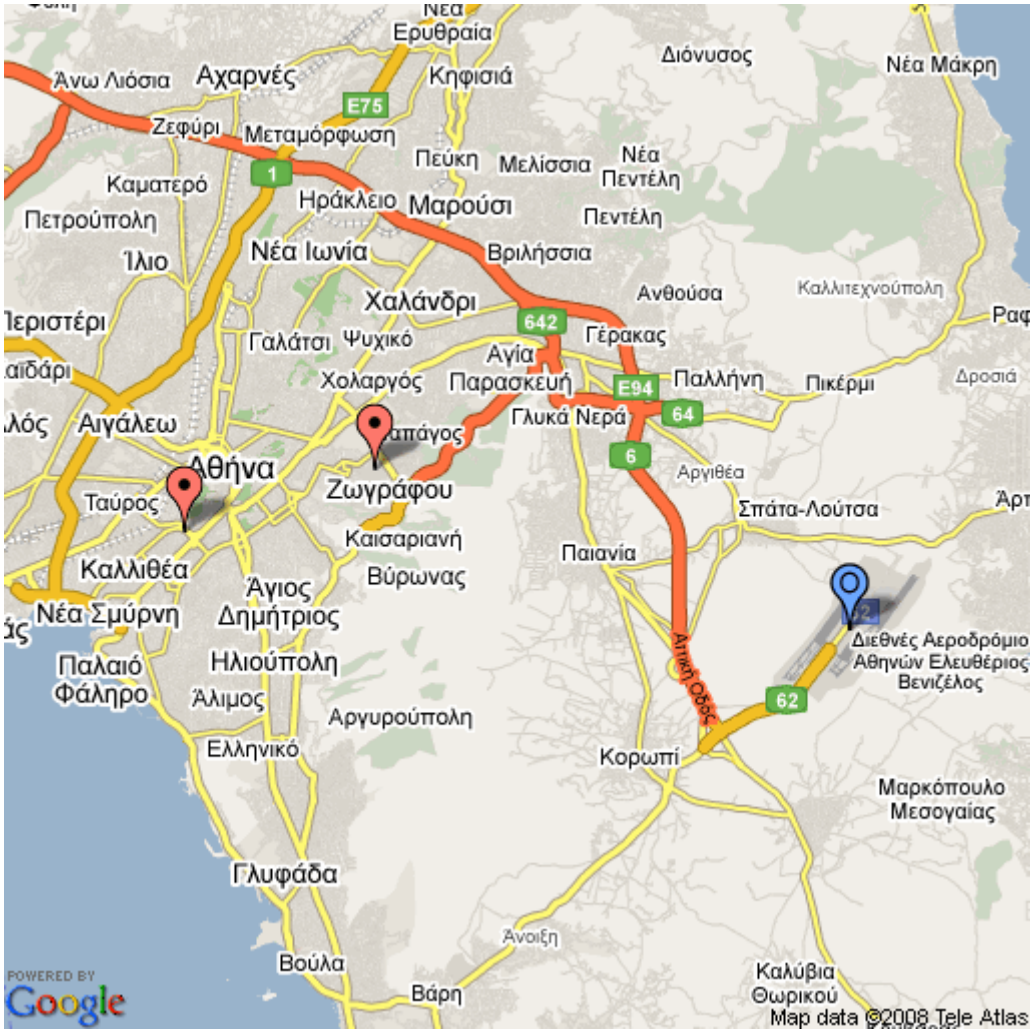
Στην λίστα στα αριστερά του παραθύρου εμφανίζονται οι αριθμοί των συνδεδεμένων χρηστών. Επιλέγοντας έναν από αυτούς έχει ως αποτέλεσμα να συμπληρωθούν οι αντίστοιχες πληροφορίες στα δεξιά του παραθύρου:

- Κάτω από την ετικέτα "Position" εμφανίζεται η τρέχουσα θέση του (συντεταγμένες).
- Δίπλα στην επιλογή "Query range" αναγράφεται η ακτίνα που έχει θέσει ο χρήστης ότι θέλει να ενημερώνεται αν κάποιος τον πλησιάσει. Αν έχει τιμή 0 τότε ο χρήστης δεν έχει ζητήσει να ενημερώνεται για κοντινούς φίλους.
- Στη λίστα "Active friends" αναγράφονται όλοι οι φίλοι του επιλεγμένου χρήστη (όχι μόνο οι συνδεδεμένοι) που βρίσκονται σε κατάσταση accepted.
- Στη λίστα "Query owners" αναγράφονται οι φίλοι του επιλεγμένου χρήστη που έχουν ζητήσει ενημέρωση για κοντινούς φίλους.
- Τέλος δίπλα στην ετικέτα "Distance" αναγράφεται η απόσταση μεταξύ του επιλεγμένου χρήστη και του επιλεγμένου φίλου του από τη λίστα "Active friends".

Οι παραπάνω πληροφορίες ενημερώνονται άμεσα και πάντα στο χρήστη εμφανίζεται έγκυρα δεδομένα.

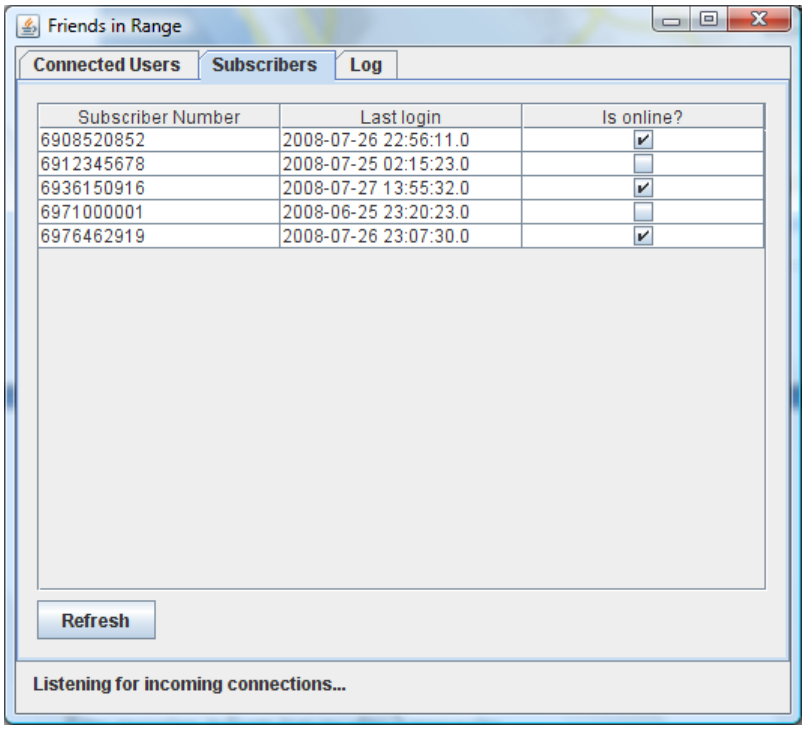
Κάτω από τη λίστα συνδεδεμένων χρηστών υπάρχει ένα drop-down μενού που χρησιμεύει για δοκιμές και παρέχει έναν εύκολο τρόπο για να στείλουμε κάποιο μήνυμα στην επιλεγμένη (στη λίστα συνδεδεμένων χρηστών) εφαρμογή-πελάτη. Επιλέγοντας κάποιο από τα διαθέσιμα μηνύματα, συμπληρώνοντας τις παραμέτρους του (αν αυτό ακολουθείται από παρενθέσεις) και πατώντας το κουμπί "Send" το αντίστοιχο μήνυμα αποστέλλεται στον επιλεγμένο χρήστη. Αναλυτικά για τα μηνύματα που ανταλλάσσονται μεταξύ clients-server στο κεφάλαιο 3.2.4.1.

Τέλος το κουμπί "Show on map" ανοίγει ένα web browser και χρησιμοποιώντας το Google Static Maps API εμφανίζει σε ένα χάρτη την τοποθεσία του επιλεγμένου χρήστη με μια μπλε καρφίτσα. Αν υπάρχουν επιλεγμένοι και φίλοι του συγκεκριμένου χρήστη από τη λίστα "Active friends" τότε στο χάρτη εμφανίζονται όλοι οι επιλεγμένοι χρήστες (οι φίλοι εμφανίζονται με κόκκινες καρφίτσες). Για παράδειγμα με επιλεγμένους τον χρήστη που φαίνονται στην Εικόνα 2.14 και τους δύο φίλους του το σύστημα εμφανίζει τον παρακάτω χάρτη (Εικόνα 2.15). Η εικόνα αυτή είναι στατική και αντιπροσωπεύει τη χρονική στιγμή που πατήθηκε το κουμπί "Show on map". Για ενημέρωση της εικόνας απαιτείται εκ νέου πάτημα του "Show on map" (όχι "refresh" στον browser).



Εικόνα 2.15. Εμφάνιση χάρτη με τις τοποθεσίες του χρήστη και των επιλεγμένων φίλων του.

Στη δεύτερη καρτέλα (Εικόνα 2.16) βλέπουμε έναν πίνακα με όλους τους συνδρομητές της υπηρεσίας, και πληροφορίες για αυτούς όπως η τελευταία φορά που συνδέθηκαν στο σύστημα καθώς και αν είναι συνδεδεμένοι τη συγκεκριμένη στιγμή που κοιτάμε την οθόνη. Πατώντας "Refresh" γίνεται ανανέωση των δεδομένων.



Εικόνα 2.16. Ο πίνακας συνδρομητών της υπηρεσίας.

Τέλος στην τρίτη και τελευταία καρτέλα αναγράφονται όλα τα μηνύματα που έχει ανταλλάξει η εφαρμογή του εξυπηρετητή με τις εφαρμογές στα κινητά τηλέφωνα (Εικόνα 2.18).

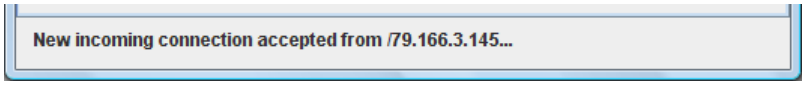
Κάθε εγγραφή ξεκινάει με:

- το αν το μήνυμα αποστέλλεται σε κάποιο κινητό
- from αν το μήνυμα αποστέλλεται από κάποιο κινητό

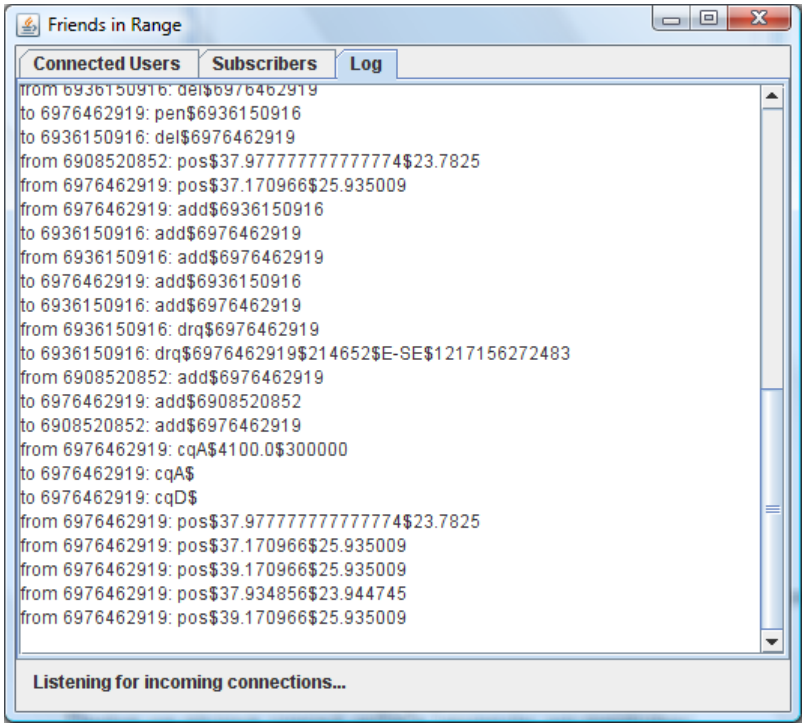
Ακολουθεί ο αριθμός του κινητού με το οποίο έγινε η επικοινωνία και μετά το σώμα του μηνύματος, για τα οποία όπως είπαμε θα αναφερθούμε αναλυτικά στο κεφάλαιο 3.2.4.1.

Εξαίρεση αποτελούν τα μηνύματα νέας σύνδεσης ενός κινητού στο σύστημα τα οποία εμφανίζονται ως εξής: new user : #[αριθμός τηλεφώνου]#.

Στο κάτω μέρος του παραθύρου εμφανίζεται το μήνυμα "Listening for incoming connections" που σημαίνει ότι η εφαρμογή ακούει για εισερχόμενες συνδέσεις σε προκαθορισμένη διεύθυνση και πόρτα. Σε αυτό το χώρο ειδοποιήσεων βλέπουμε πληροφορίες για νέες εισερχόμενες συνδέσεις (Εικόνα 2.17), όταν κάποιο κινητό συνδέεται στον server.



Εικόνα 2.17. Πληροφορία για νέα εισερχόμενη σύνδεση.



Εικόνα 2.18. Η λίστα με όλα τα μηνύματα που έχουν ανταλλαχθεί μεταξύ server-clients.

2.2.2.1 Μοντέλο Οντοτήτων-Σχέσεων Βάσης Δεδομένων

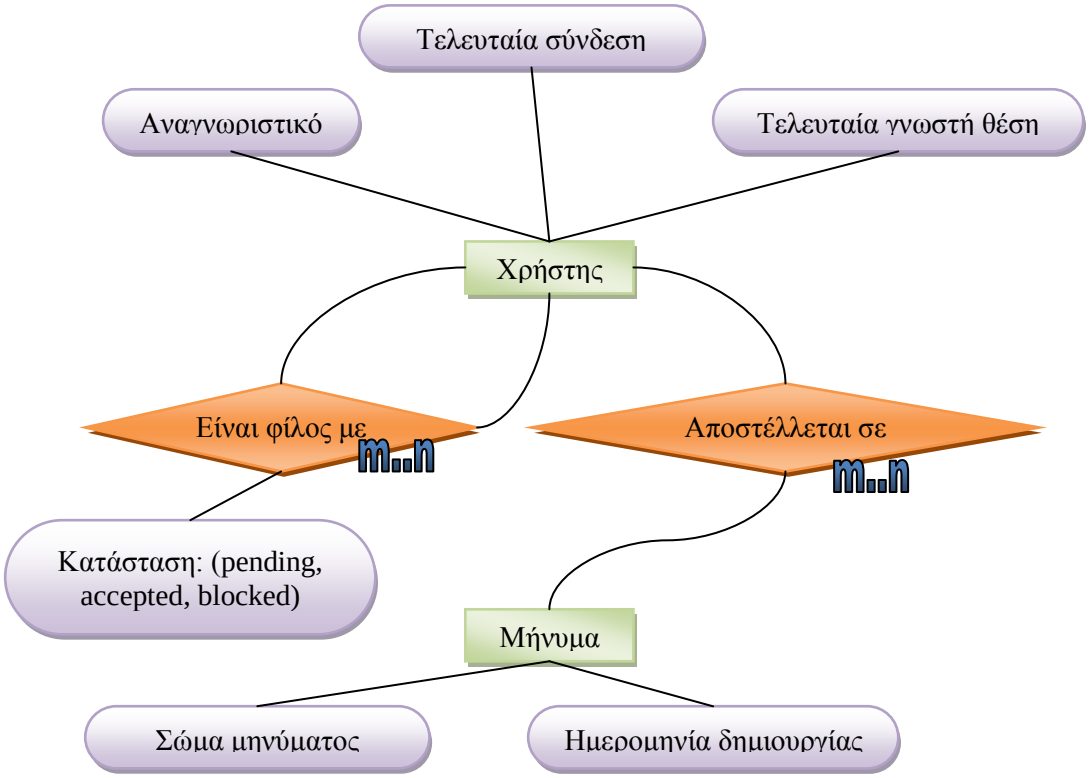
Η εφαρμογή του εξυπηρετητή χρησιμοποιεί μια βάση δεδομένων τη μόνιμη αποθήκευση κρίσιμων δεδομένων. Συγκεκριμένα αποθηκεύονται:

- Οι συνδρομητές της υπηρεσίας.
- Οι συσχετίσεις μεταξύ φίλων.
- Τα μηνύματα που δεν παραδόθηκαν άμεσα γιατί τη στιγμή της δημιουργία τους ο παραλήπτης δεν ήταν συνδεδεμένος.

Σύμφωνα με το μοντέλο οντοτήτων-σχέσεων έχουμε τις ακόλουθες οντότητες και σχέσεις:

- Η οντότητα **χρήστης** που αντιπροσωπεύει ένα χρήστη της υπηρεσίας και έχει τις ακόλουθες ιδιότητες: αναγνωριστικό χρήστη (είναι ο αριθμός του κινητού τηλεφώνου του), ημερομηνία τελευταίας σύνδεσης, συντεταγμένες τελευταίας γνωστής θέσης.
- Η οντότητα **μήνυμα** με τις εξής ιδιότητες: το κείμενο του μηνύματος, και την ημερομηνία που αυτό δημιουργήθηκε.
- Η σχέση **φίλος** που συνδέει χρήστες μεταξύ τους. Η σχέση είναι ένα προς πολλά.
- Η σχέση **αποστέλλεται σε** που συνδέει ένα μήνυμα με τον παραλήπτη του. Η σχέση είναι πολλά προς πολλά.

Ακολουθεί το διάγραμμα οντοτήτων-σχέσεων (Εικόνα 2.19).



Εικόνα 2.19. Το διάγραμμα οντοτήτων – σχέσεων της βάσης δεδομένων που χρησιμοποιείται.

2.2.3 Διαχείριση μέσω web – Λειτουργία υποστήριξης χαρτών

Ένας web server λειτουργεί παράλληλα και επιτρέπει σε διαχειριστές την ανάκτηση διαφόρων πληροφοριών για το σύστημα καθώς επίσης στηρίζει και το κομμάτι της εμφάνισης χαρτών με τις τοποθεσίες των χρηστών.

Η λειτουργία των χαρτών βασίζεται στο Google Static Maps API στο οποίο δίνουμε σαν παράμετρο της συντεταγμένες των χρηστών και μας εμφανίζει τους ανάλογους χάρτες.

Για να λειτουργήσει το Google Static Maps API χρειάζεται να κάνουμε εγγραφή για να αποκτήσουμε ένα δωρεάν κλειδί, το οποίο αντιστοιχεί σε ένα συγκεκριμένο domain. Η υπηρεσία αυτή επιτρέπει την ενσωμάτωση μιας εικόνας χάρτη σε μια ιστοσελίδα χωρίς τη χρήση JavaScript ή οποιασδήποτε τεχνολογίας δυναμικών σελίδων. Η εικόνα δημιουργείται με βάση παραμέτρους που αποστέλλονται με ένα απλό HTTP αίτημα, και μπορεί να εμφανιστεί σε οποιαδήποτε ιστοσελίδα βρίσκεται κάτω από το domain με το οποίο έχει γίνει η απόκτηση του κλειδιού που χρησιμοποιείται (περνιέται και αυτό σαν παράμετρος στο HTTP αίτημα).

Για τους σκοπούς της διπλωματικής, δημιουργήθηκε το domain friends-in-range.getmyip.com για το οποίο κατοχυρώθηκε και το αντίστοιχο υποχρεωτικό κλειδί.

Κάθε URL χάρτη για να θεωρείται έγκυρο και να επιστρέφει σωστές εικόνες πρέπει να έχει την ακόλουθη μορφή: `http://maps.google.com/staticmap?`*`parameters`*

Ορισμένες παράμετροι είναι υποχρεωτικές ενώ άλλες είναι προαιρετικές. Όπως σε όλα τα HTTP αιτήματα οι παράμετροι διαχωρίζονται με τον χαρακτήρα &. Ακολουθεί η λίστα των παραμέτρων και εξηγείται η χρήση τους.

Το Static Maps API[1] δημιουργεί τους χάρτες με βάση τις εξής παραμέτρους:

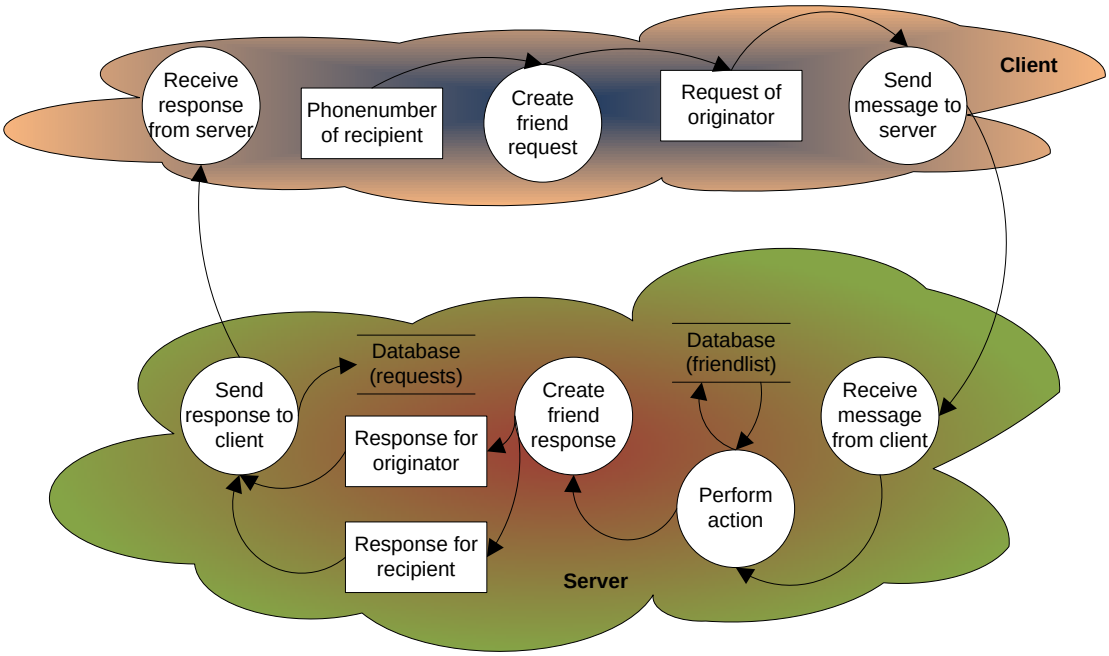
- center (χρειάζεται αν δεν υπάρχει η παράμετρος markers). Ορίζει το κέντρο της εικόνας του χάρτη. Η τιμή της παραμέτρου δίδεται ως ένα χωρισμένο με κόμμα ζεύγος {γεωγραφικό πλάτος, γεωγραφικό μήκος} που προσδιορίζει μονοσήμαντα ένα σημείο στην υδρόγειο.
- zoom (χρειάζεται αν δεν υπάρχει η παράμετρος markers). Ορίζει το επίπεδο εστίασης του χάρτη. Η τιμή της παραμέτρου είναι ένας ακέραιος με επιτρεπτές τιμές από 0 (ολόκληρη η γη σε ένα χάρτη) μέχρι και 19 (η μεγαλύτερη δυνατή λεπτομέρεια). Σημειώνεται ότι δεν είναι διαθέσιμα όλα τα επίπεδα εστίασης σε όλα τα σημεία της υδρόγειου.
- size (υποχρεωτικό). Ορίζει το μέγεθος της εικόνας. Η τιμή της παραμέτρου είναι μια συμβολοσειρά με το εξής format: οριζόντιαΔιάστασηxκατακόρυφηΔιάσταση. Μέγιστο επιτρεπτό μέγεθος είναι 640x640.
- format (προαιρετικό). Ορίζει τον τύπο αρχείου της εικόνας (gif, jpeg, png).
- maptype (προαιρετικό). Ορίζει τον τύπο του χάρτη. Διαθέσιμη τύποι είναι δορυφορικός, εδαφικός, υβριδικός και για κινητό.
- markers (προαιρετικό). Ορίζει σημεία στο χάρτη όπου εμφανίζεται ένα γραφικό σαν καρφίτσα για να υποδηλώσει κάποιο σημείο ενδιαφέροντος. Δίνεται σαν ζεύγη {γεωγραφικού πλάτους, γεωγραφικού μήκος} συνοδευόμενα από το επιθυμητό χρώμα και προαιρετικά έναν χαρακτήρα του αγγλικού αλφάβητου.
- path (προαιρετικό). Δίνει τη δυνατότητα να ζωγραφιστούν πολυγωνικές γραμμές (μονοπάτια) στο χάρτη.
- span (προαιρετικό). Ορίζει ένα ελάχιστον παράθυρο προβολή για το χάρτη σε σχέση με το δοθέν κέντρο.
- frame (προαιρετικό). Περιβάλλει την εικόνα του χάρτη με ένα μπλε περίγραμμα.
- hl (προαιρετικό). Ορίζει τη γλώσσα εμφάνισης των ονομασιών των τοποθεσιών στο χάρτη. Αν η συγκεκριμένη γλώσσα δεν είναι διαθέσιμη, χρησιμοποιείται η προεπιλεγμένη.
- key (υποχρεωτικό). Το κλειδί που έχει αποδοθεί για το συγκεκριμένο domain από το οποίο θα είναι διαθέσιμες οι ιστοσελίδες.

[1] Για ανανεωμένες πληροφορίες σχετικά με το API http://code.google.com/apis/maps/documentation/staticmaps

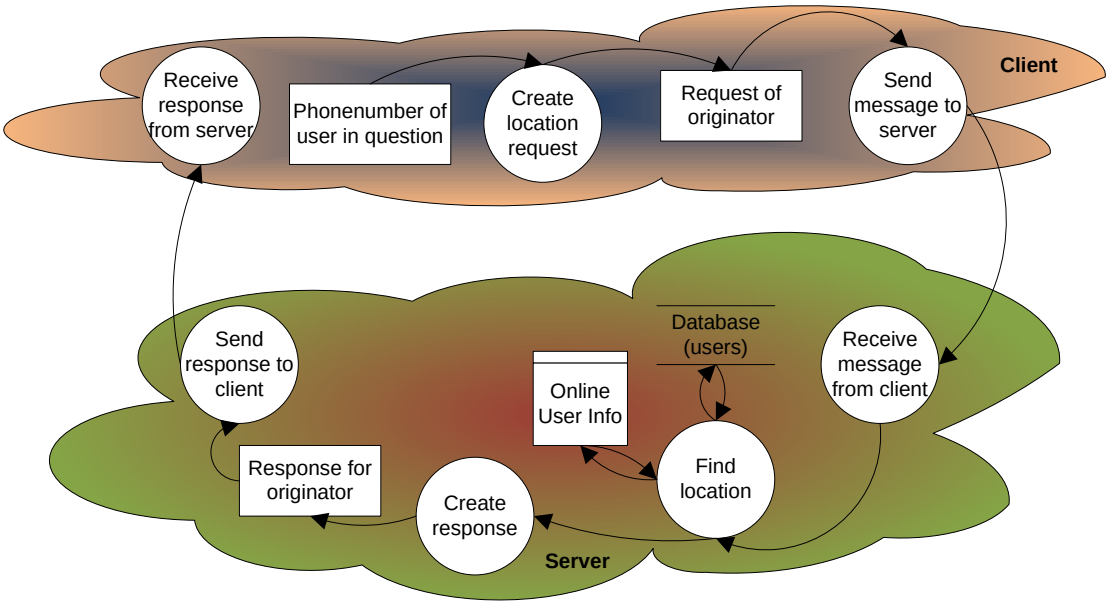
2.2.4 Διαγράμματα ροής δεδομένων

Ακολουθούν διαγράμματα ροής δεδομένων για τις ακόλουθες περιπτώσεις:

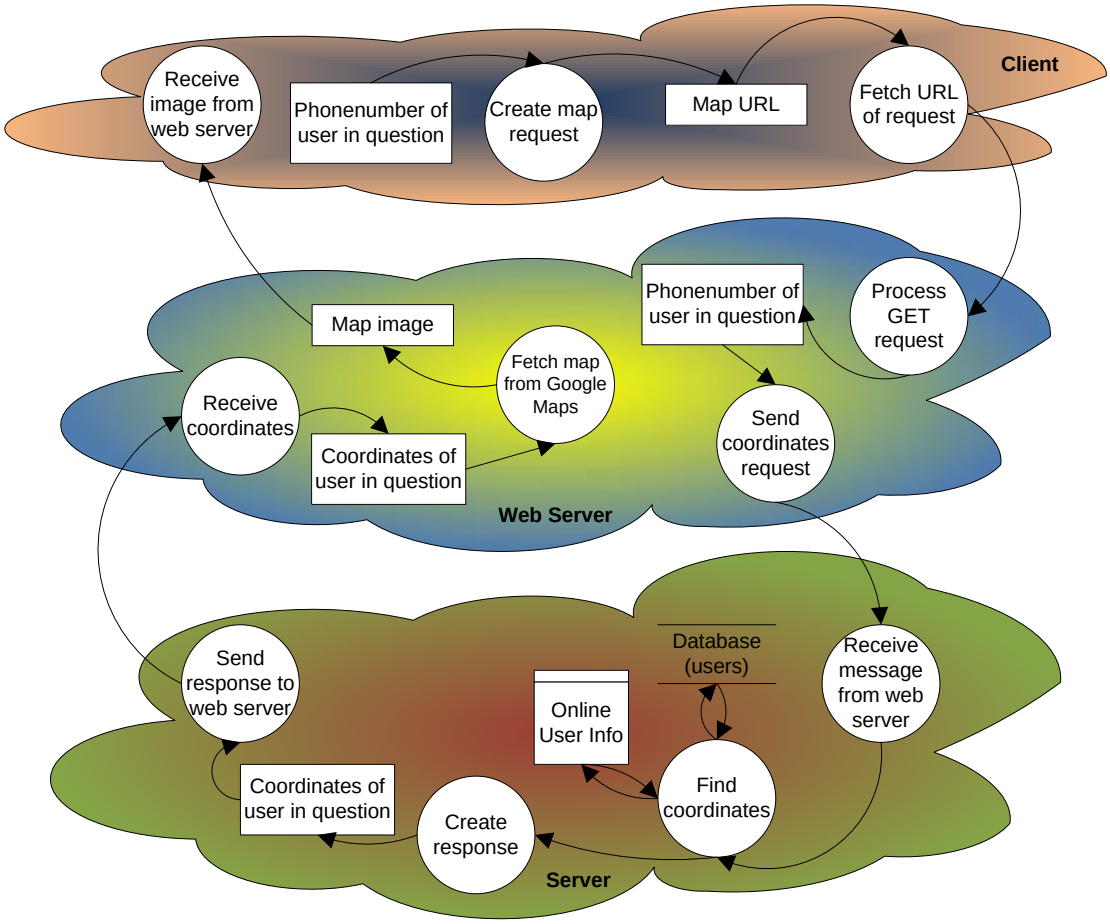
2.2.4.1 Προσθήκη-διαγραφή-αποκλεισμός-άρση αποκλεισμού φίλου.



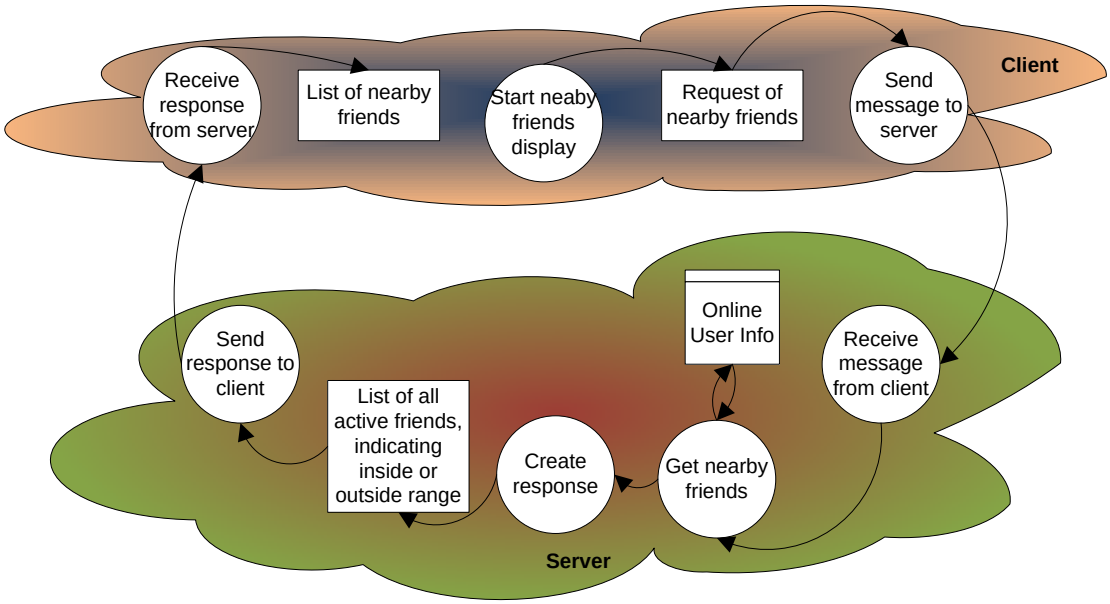
2.2.4.2 Ερώτημα θέσης φίλου.



2.2.4.3 *Προβολή φίλου στο χάρτη.*



2.2.4.4 *Ενεργοποίηση ενημέρωση κοντινών φίλων.*



3 *Σχεδίαση Συστήματος*

Ακολουθεί μια συνοπτική περιγραφή της αρχιτεκτονικής του συστήματος και στη συνέχεια αναλυτική περιγραφή των κλάσεων κάθε υποσυστήματος. Η περιγραφή συμπληρώνεται από ένα γενικό σχήμα που δείχνει τις κλάσεις και πώς αυτές επικοινωνούν μεταξύ τους.

3.1 Αρχιτεκτονική

Το σύστημα ακολουθεί την αρχιτεκτονική 3-tier, όπου τα προγράμματα-πελάτη επικοινωνούν με έναν κεντρικό εξυπηρετητή, ο οποίος για λειτουργίες μόνιμης αποθήκευσης αλλά και ανάκτησης δεδομένων χρειάζεται να επικοινωνήσει με κάποια βάση δεδομένων.

Όπως είναι αναμενόμενο, μεταφέρουμε όλο τον υπολογιστικό φόρτο στην πλευρά του server και σχεδιάζουμε την εφαρμογή πελάτη απλή και εύκολη στη χρήση. Επειδή το πρόγραμμα πελάτη προορίζεται για κινητά τηλέφωνα πρέπει να έχει ένα απλό γραφικό περιβάλλον το οποίο ο χρήστης να μπορεί να οικειοποιηθεί εύκολα και γρήγορα, χωρίς να είναι απαραίτητη η ανάγνωση κάποιου εγχειριδίου χρήσης.

3.2 Περιγραφή Κλάσεων

Παρουσιάζονται παρακάτω συνοπτικά οι λειτουργίες/μέθοδοι/συναρτήσεις των κλάσεων κάθε υποσυστήματος.

3.2.1 Εφαρμογή client

Η εφαρμογή-πελάτη αποτελείται από 16 κλάσεις Java. Αυτές είναι χωρισμένες σε 3 πακέτα:

- πακέτο **visual**: αφορά το γραφικό περιβάλλον της εφαρμογής.
- πακέτο **storage**: ασχολείται με θέματα μόνιμης αποθήκευσης και ανάκτησης πληροφοριών στη συσκευή

- πακέτο **comm**: περιέχει κλάσεις σχετικές με την δικτυακή επικοινωνία του προγράμματος με τον κεντρικό εξυπηρετητή ή με το δίκτυο κινητής τηλεφωνίας

Ακολουθούν αναλυτικά όλες οι κλάσεις της εφαρμογής:

3.2.1.1 visual.UI

Η κεντρική κλάση της εφαρμογής.

Η κλάση αυτή είναι υποκλάση της κλάσης MIDlet που αντιπροσωπεύει μια αυτόνομη εφαρμογή της πλατφόρμας Java ME. Περιέχει κώδικα για τη δημιουργία όλων των οθονών που εμφανίζονται κατά τη διάρκεια εκτέλεσης της εφαρμογής καθώς και των κώδικα για όλους τους ακροατές συμβάντων εισόδου από το χρήστη.

3.2.1.2 visual.Update

Αναλαμβάνει την ενημέρωση του γραφικού περιβάλλοντος. Η κλάση αυτή περιλαμβάνει μεθόδους που ανανεώνουν τις διάφορες οθόνες της εφαρμογής και οι οποίες μπορούν να χωριστούν σε δύο ομάδες: σε αυτές που ανανεώνουν την οθόνη μετά από κάποια ενέργεια του χρήστη (πχ. η προσθήκη ενός νέου φίλου καλεί τη μέθοδο που εισάγει τη νέα καταχώρηση στη λίστα φίλων) και σε αυτές που καλούνται ως απόκριση της εφαρμογής σε κάποιο εισερχόμενο μήνυμα (πχ. ένα εισερχόμενο μήνυμα αλλαγής θέσης ενός φίλου καλεί τη μέθοδο που ενημερώνει τη λίστα κοντινών φίλων).

3.2.1.3 visual.List

Επεκτείνει την οθόνη List.

Η κλάση javax.microedition.lcdui.List είναι υπεύθυνη για τη δημιουργία οθονών που έχουν λειτουργικότητα λίστας, δηλαδή μια σειρά από επιλογές που ο χρήστης μπορεί να προσπελάσει και να επιλέξει αυτή ή αυτές που τον ενδιαφέρουν. Η κλάση visual.List επεκτείνει την προαναφερθείσα κλάση με λειτουργίες αναζήτησης, προσθήκης και διαγραφής καταχωρήσεων.

3.2.1.4 visual.Ticker

Επεκτείνει την οθόνη Ticker.

Η κλάση javax.microedition.lcdui.Ticker είναι υπεύθυνη για τη δημιουργία ενός χώρου προβολής μηνυμάτων της εφαρμογής πάνω στην τρέχουσα οθόνη. Συνήθως αυτά εμφανίζονται ως κυλιόμενα μηνύματα σε ένα χώρο κάτω από τον τίτλο της εφαρμογής. Η κλάση visual.Ticker επεκτείνει τη λειτουργικότητα της προαναφερθείσας κλάσης κάνοντας τα εμφανιζόμενα μηνύματα να συμπεριφέρονται σαν στοίβα. Δηλαδή νέα μηνύματα

εμφανίζονται, διαγράφοντας τα παλιά, ενώ όταν η ειδοποίηση δεν χρειάζεται πλέον, επιστρέφει το προηγούμενα εμφανιζόμενο μήνυμα.

3.2.1.5 storage.Initialization

Αναλαμβάνει την αρχικοποίηση της εφαρμογής. Κατά την πρώτη εκτέλεση της εφαρμογής ζητάει από το χρήστη να εισάγει τον αριθμό του τηλεφώνου του, ενώ σε επόμενες εκτελέσεις διαβάζει τον τηλεφωνικό κατάλογο της συσκευής, ανασύρει τη λίστα φίλων του χρήστη και τις αποθηκευμένες τοποθεσίες από το μόνιμο αποθηκευτικό χώρο της εφαρμογής, ελέγχει για την υποστήριξη αυτόματου εντοπισμού θέσης και εκκινεί την σύνδεση με τον κεντρικό εξυπηρετητή.

3.2.1.6 storage.Load

Περιέχει μεθόδους που εκτελούν λειτουργίες ανάκτησης δεδομένων από το χώρο μόνιμης αποθήκευσης της εφαρμογής. Συγκεκριμένα επιτρέπει την ανάκτηση του καταλόγου των επαφών της συσκευής, του αναγνωριστικού χρήστη, της λίστας φίλων, καθώς και της λίστας αποθηκευμένων τοποθεσιών. Οι μέθοδοι αυτοί καλούνται κατά την αρχικοποίηση της εφαρμογής από την κλάση Initialization.

3.2.1.7 storage.Friend

Η οντότητα φίλος όπως αυτή διαχειρίζεται από την εφαρμογή. Περιέχει πεδία όπως το όνομα, ο αριθμός τηλεφώνου (μοναδικό αναγνωριστικό) και την κατάσταση του φίλου στη λίστα του χρήστη και κατάλληλες μεθόδους για την επεξεργασία αυτών.

3.2.1.8 storage.KnownLocation

Η οντότητα τοποθεσία όπως αυτή διαχειρίζεται από την εφαρμογή. Περιέχει πεδία όπως το όνομα και τις συντεταγμένες της τοποθεσίας καθώς και κατάλληλες μεθόδους για την επεξεργασία αυτών.

3.2.1.9 storage.StringTokenizer

Βοηθητική κλάση που χρησιμοποιείται για το διαχωρισμό μια συμβολοσειράς σε επιμέρους συμβολοσειρές οριοθετούμενες από κάποιον χαρακτήρα ή ακολουθία χαρακτήρων.

3.2.1.10 comm.ServerConnection

Αναλαμβάνει τη σύνδεση στον κεντρικό εξυπηρετητή και τη διατήρησή της. Αφού η σύνδεση επιτευχθεί διατηρεί ανοιχτό το κανάλι επικοινωνίας και παρακολουθεί για εισερχόμενα μηνύματα.

3.2.1.11 comm.Protocol

Υλοποίηση πρωτοκόλλου επικοινωνίας. Παρέχει μεθόδους δημιουργίας όλων των μηνυμάτων που αποστέλλονται κατά τη διάρκεια της εφαρμογής καθώς και αναγνώρισης όλων των εισερχόμενων μηνυμάτων και εκτέλεση της κατάλληλης ενέργειας.

3.2.1.12 comm.Outgoing

Διαχείριση εξερχόμενων μηνυμάτων. Οι μέθοδοι αυτής της κλάσης χρησιμοποιούνται για την αποστολή μηνυμάτων από το κινητό τηλέφωνο στον κεντρικό εξυπηρετητή. Αφού γίνει η απαραίτητη συλλογή πληροφοριών χρησιμοποιούνται οι μέθοδοι της κλάσης Protocol για τη δημιουργία των μηνυμάτων και κατόπιν γίνεται η αποστολή.

3.2.1.13 comm.Validation

Αναλαμβάνει την επικύρωση του αριθμού τηλεφώνου που εισήγαγε ο χρήστης. Όπως έχει ήδη αναφερθεί την πρώτη φορά που εκτελείται η εφαρμογή σε μια συσκευή ζητείται από το χρήστη να εισάγει τον αριθμό τηλεφώνου του. Η κλάση αυτή εκτελεί την επαλήθευση του αριθμού που εισάγει ο χρήστης με το να στέλνει ένα SMS στο νούμερο αυτό και περιμένει τη λήψη από τη συσκευή.

3.2.1.14 comm.BreakConnection

Βοηθητική κλάση για τη διακοπή μιας σύνδεσης. Επειδή οι περισσότερες συνδέσεις που πραγματοποιούνται είναι blocking I/O, δηλαδή το νήμα που τις καλεί σταματάει την εκτέλεση του σε εντολές ανάγνωσης, μέχρις ότου ληφθούν κάποια δεδομένα, χρησιμοποιούμε αυτή την κλάση για να διακόπτουμε αυτές τις συνδέσεις μετά από ένα ορισμένο χρόνο αναμονής. Στην τρέχουσα υλοποίηση αυτό γίνεται μόνο στην αναμονή για εισερχόμενο SMS κατά την αρχική διαδικασία ταυτοποίησης και επικύρωσης του αριθμού τηλεφώνου που εισάγει ο χρήστης.

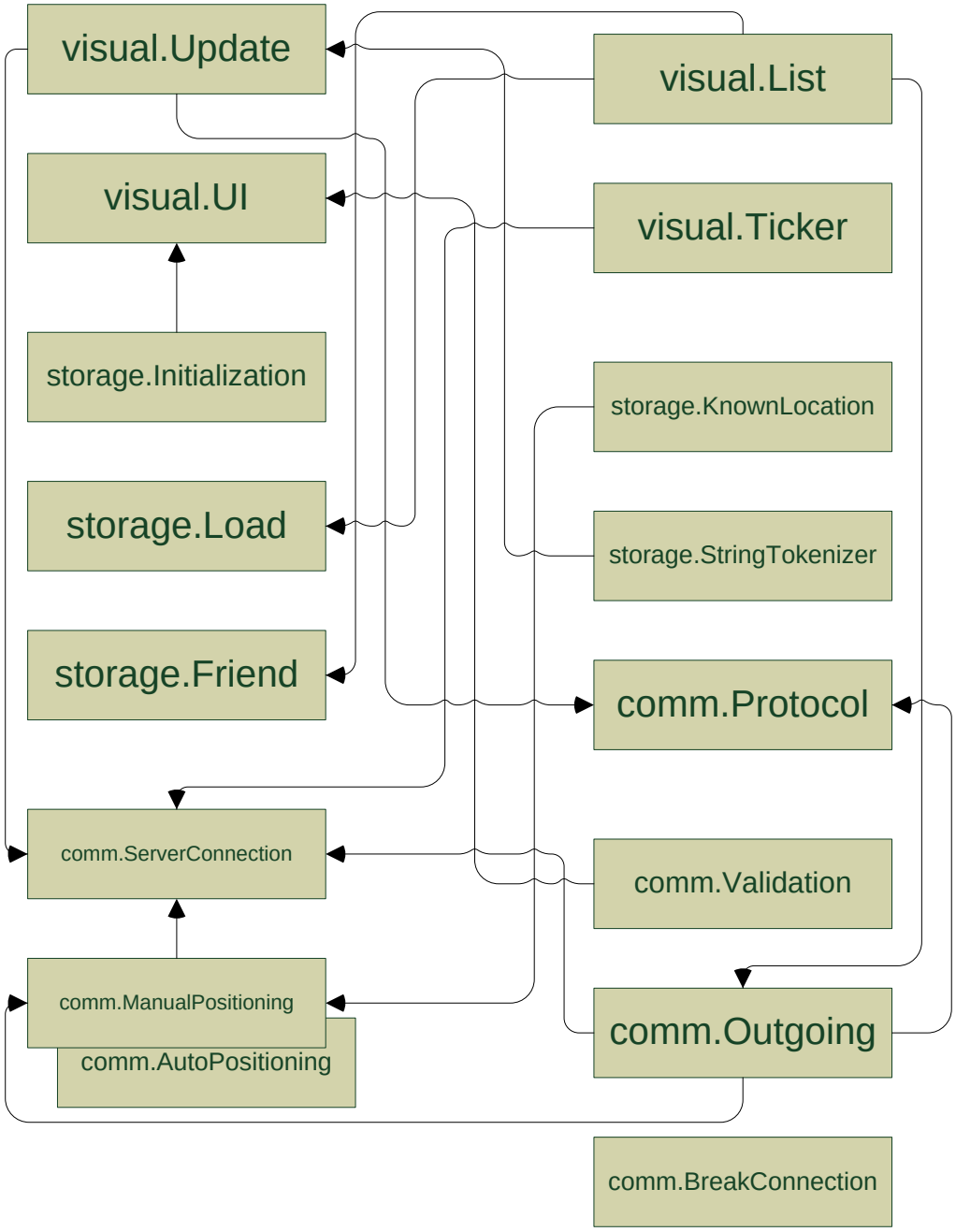
3.2.1.15 comm.ManualPositioning

Λειτουργία μη αυτόματης ρύθμισης θέσης. Εκτελεί αλλαγές στην τρέχουσα θέση μετά από είσοδο από το χρήστη και ενημερώνει κατάλληλα το γραφικό περιβάλλον αλλά και τον κεντρικό εξυπηρετητή.

3.2.1.16 comm.AutoPositioning

Λειτουργία αυτόματης ρύθμισης θέσης. Χρησιμοποιεί το Location API σε όσες συσκευές αυτό υποστηρίζεται για την ανάκτηση της τρέχουσας θέσης του κινητού. Προσφέρει τη

δυνατότητα προγραμματισμού ερώτησης για την θέση του κινητού ανά μεταβλητά χρονικά διαστήματα, ή απενεργοποίησης αυτής.



Εικόνα 3.1. Block διάγραμμα κλάσεων της εφαρμογής-πελάτη.

3.2.2 Εφαρμογή server

Η εφαρμογή-πελάτη αποτελείται από 10 κλάσεις Java. Αυτές είναι χωρισμένες σε 4 πακέτα:

- πακέτο **gui**: αφορά το γραφικό περιβάλλον της εφαρμογής.
- πακέτο **networking**: περιέχει κλάσεις σχετικές με την δικτυακή επικοινωνία του προγράμματος με τα προγράμματα-πελάτη που τρέχουν στα κινητά.

- πακέτο **online**: περιέχει κλάσεις που υλοποιούν οντότητες που κρατούν πληροφορίες σχετικές με τους συνδεδεμένους χρήστες.
- πακέτο **database**: ασχολείται με θέματα μόνιμης αποθήκευσης και ανάκτησης πληροφοριών από τη βάση δεδομένων.

Ακολουθούν αναλυτικά όλες οι κλάσεις της εφαρμογής:

3.2.2.1 gui.MainView

Η κλάση αυτή σχεδιάζει το γραφικό περιβάλλον της εφαρμογής με τη λειτουργικότητα που έχει περιγραφεί στην ενότητα 2.2.2. Περιλαμβάνει επίσης τους ακροατές συμβάντων και εκκινεί τα νήματα που ελέγχουν για νέες εισερχόμενες συνδέσεις.

3.2.2.2 networking.ServerListener

Η κλάση αυτή τρέχει σαν ξεχωριστό νήμα συνεχώς και ελέγχει για νέες εισερχόμενες συνδέσεις από κινητά τηλέφωνα. Μόλις μια νέα σύνδεση έχει επιτευχθεί δημιουργεί ένα νέο νήμα της κλάσης UserListener για τη διαχείριση του συγκεκριμένου χρήστη ενώ συνεχίζει την εκτέλεση της για να αντιμετωπίσει επερχόμενες συνδέσεις.

3.2.2.3 networking.UserListener

Η κλάση αυτή δημιουργεί ένα νήμα το οποίο εκτελεί λειτουργίες που αφορούν αποκλειστικά ένα χρήστη. Υπάρχει ένας ακροατής εισερχόμενων μηνυμάτων από το εκάστοτε κινητό τηλέφωνο του χρήστη που πυροδοτεί στη συνέχεια την εκτέλεσης της κατάλληλης ενέργειας από το server.

3.2.2.4 networking.Protocol

Επιτρέπει τη δημιουργία και αποστολή μηνυμάτων στα κινητά τηλέφωνα καθώς και μεταφράζει τα εισερχόμενα μηνύματα που δέχεται ο εξυπηρετητής εκτελώντας την κατάλληλη κάθε φορά ενέργεια.

3.2.2.5 online.UserInfo

Η κλάση αυτή υλοποιεί τη δομή που αποθηκεύει τις πληροφορίες που αφορούν κάθε χρήστη που είναι συνδεδεμένος στο σύστημα. Οι πληροφορίες αυτές αποτελούνται από στοιχεία ταυτοποίησης του χρήστη (αριθμός τηλεφώνου), στοιχεία σχετικά με τη θέση του χρήστη και τα ερωτήματα που εκτελεί, καθώς και τις συσχετίσεις του με τους άλλους χρήστες της υπηρεσίας. Υπάρχουν τέλος και οι αντίστοιχες μέθοδοι που ενημερώνουν τα παραπάνω δεδομένα, όπως απαιτείται από τη λογική της εφαρμογής.

3.2.2.6 online.Coordinates

Η κλάση αυτή υλοποιεί μια δομή για αποθήκευση συντεταγμένων, δηλαδή ενός μονοσήμαντου σημείου στην υδρόγειο. Οι συντεταγμένες αποθηκεύονται ως ένα ζευγάρι αριθμών (γωνιών), και παρέχονται μέθοδοι για υπολογισμό αποστάσεων μεταξύ δύο σημείων, υπολογισμό κατεύθυνσης, και μετατροπή από δεκαδική μορφή στη μορφή γωνίας και αντίστροφα.

3.2.2.7 database.dbTransaction

Η κλάση αυτή χρησιμεύει για τη σύνδεση με τη βάση δεδομένων και περιέχει μεθόδους για δημιουργία παραμετρικών ερωτημάτων και ενημερώσεων στη βάση.

3.2.2.8 database.Registration

Η κλάση αυτή περιέχει μεθόδους σχετικές με την εγγραφή χρηστών στη βάση και αποθήκευση και ανάκτηση πληροφοριών σχετικών με την τελευταία γνωστή θέση τους και την ημερομηνία τελευταίας σύνδεσης στο συστημα.

3.2.2.9 database.Friend

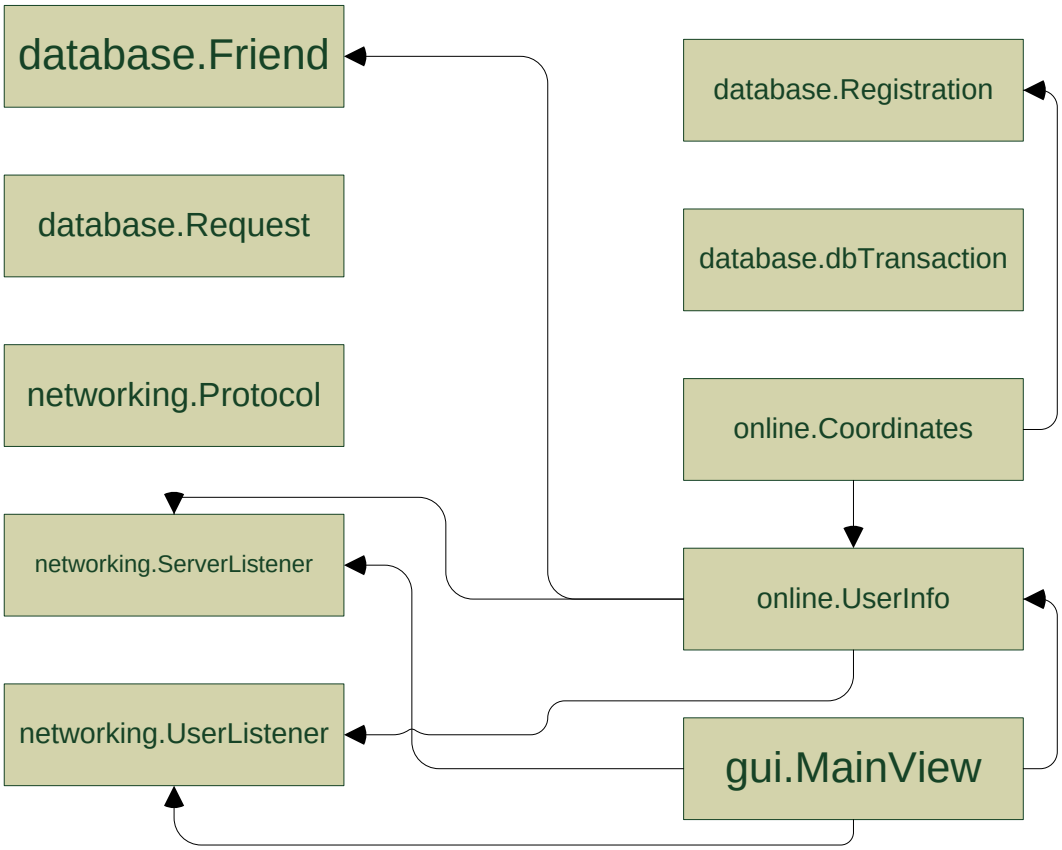
Αναλαμβάνει την ενημέρωση της βάσης δεδομένων για να βρίσκεται σε συνέπεια με τις αλλαγές που κάνουν οι χρήστες στις λίστες φίλων τους. Περιέχει μεθόδους για δημιουργία νέας καταχώρησης, για αλλαγή κατάστασης και για διαγραφή φίλου, μαζί με την αποστολή των κατάλληλων μηνυμάτων προς τους χρήστες που τους αφορά η συγκεκριμένη τροποποίηση.

3.2.2.10 database.Request

Η κλάση αυτή παρέχει μεθόδους για την αποθήκευση στη βάση των μηνυμάτων που αποστέλλονται σε offline χρήστες καθώς επίσης και τις αντίστοιχες μεθόδους ανάκλησης των μηνυμάτων αυτών όταν ο εν λόγω χρήστης συνδεθεί πάλι στο σύστημα.

3.2.3 Βάση Δεδομένων

Η βάση δεδομένων χρησιμοποιείται από την εφαρμογή για λόγους αξιόπιστης και γρήγορης αποθήκευσης μόνιμων πληροφοριών που δεν πρέπει να χαθούν αν για κάποιο λόγο χρειαστεί να τερματιστεί προσωρινά η λειτουργία του εξυπηρετητή (πχ για λόγους συντήρησης ή αναβάθμισης).



Εικόνα 3.2. Block διάγραμμα κλάσεων της εφαρμογής του εξυπηρετητή.

Οι πληροφορίες που αποθηκεύονται είναι:

- Οι αριθμοί των χρηστών που έχουν χρησιμοποιήσει έστω και μια φορά την υπηρεσία, μαζί με την ημερομηνία που πραγματοποιήθηκε η τελευταία σύνδεση και την τελευταία γνωστή τοποθεσία από όπου συνδέθηκαν.
- Οι λίστες φίλων κάθε χρήστη της υπηρεσίας μαζί με την ένδειξη της κατάστασης της σχέσης τους (“accepted”, “blocked”, “pending”).
- Ένας κατάλογος μηνυμάτων που στάλθηκαν σε χρήστες αλλά δεν παραδόθηκαν γιατί οι παραλήπτες ήταν εκτός σύνδεσης. Εδώ αποθηκεύουμε τον αριθμό του παραλήπτη, το κείμενο του μηνύματος καθώς και την ημερομηνία δημιουργίας του μηνύματος. Κατά την επόμενη σύνδεση ενός χρήστη όλα τα μηνύματα που έχουν αυτόν ως παραλήπτη στέλνονται και διαγράφονται από τη βάση.

3.2.4 Web server

Ο web server έχει δύο ρόλους στο σύστημα:

- μπορεί να λειτουργήσει ως ένα διαχειριστικό web interface για το σύστημα δίνοντας σε διαχειριστές δυνατότητα απομακρυσμένης εποπτείας του συστήματος ή να δώσει

σε απλούς χρήστες web πρόσβαση στο λογαριασμό τους. Το κομμάτι αυτό έχει υλοποιηθεί υποτυπωδώς.

- επιτρέπει στους χρήστες και στους διαχειριστές την προβολή της τοποθεσίας άλλων χρηστών σε χάρτη χρησιμοποιώντας της υπηρεσίας Google Maps.

Εδώ θα δούμε τα servlets Map και MobileMap που επιτρέπουν την εμφάνιση χαρτών στους διαχειριστές και στους χρήστες αντίστοιχα.

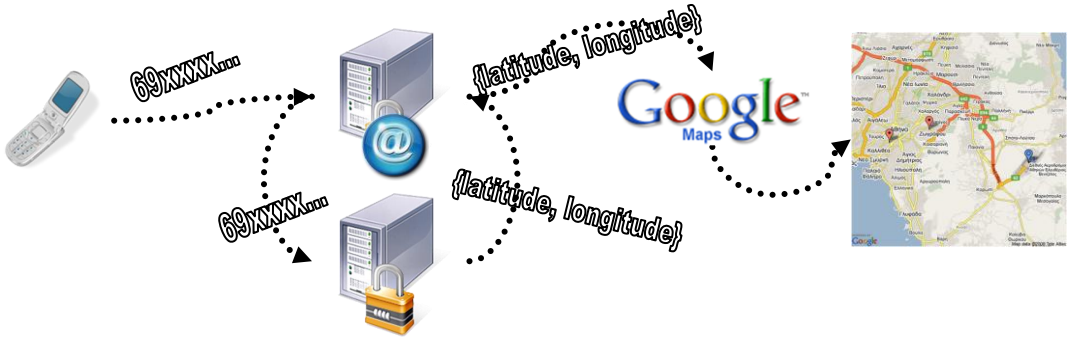
3.2.4.1 Map

Οι διαχειριστές του συστήματος έχουν πρόσβαση στον κεντρικό εξυπηρετητή και άρα γνωρίζουν τις συντεταγμένες κάθε χρήστη του συστήματος. Έτσι για να δουν τη θέση ενός χρήστη στο χάρτη αρκεί να καλέσουν το Google Static Maps API και να δώσουν ως παράμετρο τις συντεταγμένες του.

Εικόνα 3.3. Σχηματική αναπαράσταση δημιουργίας χάρτη από διαχειριστή της υπηρεσίας.

3.2.4.2 MobileMap

Οι χρήστες της υπηρεσίας δεν ξέρουν άμεσα την τοποθεσία άλλων χρηστών. Πρέπει να προηγηθεί ερώτημα στον κεντρικό εξυπηρετητή. Έτσι αυτό το servlet αυτό αντιστοιχεί το νούμερο του τηλεφώνου στις σωστές συντεταγμένες μέσω του κεντρικού εξυπηρετητή και μετά στέλνει την εικόνα του χάρτη στο κινητό τηλέφωνο του χρήστη.



Εικόνα 3.4. Σχηματική αναπαράσταση δημιουργίας χάρτη από χρήστη της υπηρεσίας.

3.3 Πρωτόκολλο επικοινωνίας

Για την επικοινωνία μεταξύ του κεντρικού εξυπηρετητή και των εφαρμογών στα κινητά τηλέφωνα χρησιμοποιείται το παρακάτω πρωτόκολλο. Όλα τα μηνύματα είναι απλά μηνύματα κειμένου και στέλνονται χωρίς κρυπτογράφηση.

Εκτός από το αρχικό μήνυμα που αποστέλλει το κινητό όταν συνδέεται με τον server που έχει τη δομή:

#	ΑΡΙΘΜΟΣ_ΤΗΛΕΦΩΝΟΥ	#

κάθε άλλο μήνυμα έχει την παρακάτω δομή:

ΕΝΤΟΛΗ	$	ΠΑΡΑΜΕΤΡΟΣ1	$	ΠΑΡΑΜΕΤΡΟΣ2	$…

Ας δούμε παρακάτω αναλυτικά τα μηνύματα. Τα μηνύματα χωρίζονται σε δύο κατηγορίες: σε αυτά που στέλνει το κινητό και σε αυτά που στέλνει ο server. Αν και τα περισσότερα είναι κοινά, υπάρχουν εντούτοις κάποια που ανήκουν αποκλειστικά σε κάποια κατηγορία.

Παρακάτω καταγράφονται όλα τα μηνύματα που αποστέλλονται με το κωδικό όνομα που τα χρησιμοποιεί η εφαρμογή και δίπλα την αναπαράστασή τους ως συμβολοσειρά, το κομμάτι δηλαδή που αντικαθιστά την ΕΝΤΟΛΗ στο προηγούμενο σχήμα.

Δίπλα από κάθε μήνυμα υπάρχει ένα σχήμα που δείχνει την κατεύθυνση αποστολής του μηνύματος, δηλαδή αν αυτό αποστέλλεται μόνο από το κινητό, μόνο από τον server ή και από τα δύο υποσυστήματα. Στις περιπτώσεις αμφίδρομου μηνύματος, ο συμβολισμός M → S σημαίνει από το κινητό στον εξυπηρετητή (mobile to server) ενώ το αντίθετο M ← S σημαίνει από τον εξυπηρετητή στο κινητό (server to mobile).

3.3.1.1 *Μήνυμα αρχικής σύνδεσης*

Όταν επιτευχθεί η σύνδεση μεταξύ κινητού και εξυπηρετητή το πρώτο μήνυμα που αποστέλλεται πρέπει να περιέχει τον αριθμό του κινητού όπως φαίνεται στο προηγούμενο σχήμα. Σε διαφορετική περίπτωση ο εξυπηρετητής τερματίζει τη σύνδεση. Ο εξυπηρετητής ενημερώνει τη βάση δεδομένων με τη νέα ημερομηνία τελευταίας εισόδου.

3.3.1.2 *Μήνυμα TERMINATE = "end"*

Δεν δέχεται παραμέτρους. Αποστέλλεται κατά το κλείσιμο της εφαρμογής στο server για να διακόψει την επικοινωνία και να ενημερώσει ότι το κινητό είναι πλέον εκτός σύνδεσης. Ο εξυπηρετητής ενημερώνει τη βάση δεδομένων με τη νέα τελευταία γνωστή θέση του κινητού.

3.3.1.3 Μήνυμα ADD = "add"

Δέχεται πολλές παραμέτρους. Όλες είναι δεκαψήφιοι αριθμοί τηλεφώνου που υποδηλώνουν άλλους χρήστες της εφαρμογής.

M → S. Το κινητό μπορεί να στείλει μήνυμα ADD σε δύο περιπτώσεις. Αρχικά το μήνυμα αυτό αποστέλλεται όταν ο χρήστης αιτηθεί να εισάγει νέους χρήστες στη λίστα φίλων του. Όταν ο χρήστης προσθέσει ένα νέο χρήστη στη λίστα φίλων, αυτός εισέρχεται σε κατάσταση "pending" ενώ ταυτόχρονα η συσκευή στέλνει το μήνυμα ADD$[αριθμός_νέου_χρήστη] στον κεντρικό εξυπηρετητή. Ακόμα όταν ένας χρήστης έχει εισερχόμενο αίτημα για νέο φίλο κατά την αποδοχή αυτού πάλι αποστέλλεται μήνυμα ADD$[αριθμός_χρήστη_αιτήματος].

M ← S. Ο εξυπηρετητής μπορεί να στείλει μήνυμα ADD σε δύο περιπτώσεις. Η πρώτη περίπτωση είναι όταν ο χρήστης Α έχει αιτηθεί να γίνει ο Β φίλος του, ο εξυπηρετητής θα στείλει στον Β μήνυμα ADD$[αριθμόςΑ]. Η εφαρμογή στο κινητό σαν απάντηση στο εισερχόμενο μήνυμα θα πληροφορήσει τον Β ότι έχει μία νέα εισερχόμενη αίτηση. Η δεύτερη περίπτωση (σε συνέχεια του παραπάνω σεναρίου), όταν ο Β αποδεχθεί την αίτηση του Α, και εγκατασταθεί η σχέση ο εξυπηρετητής θα στείλει και στους δύο χρήστες Α και Β, πάλι μήνυμα ADD ώστε να ενημερωθούν και τα κινητά ότι πλέον η σχέση είναι σε ισχύ και να εμφανίσουν τα κατάλληλα μηνύματα στους χρήστες (να αλλάξουν δηλαδή την κατάσταση των φίλων στα κινητά από "pending" σε "accepted").

3.3.1.4 Μήνυμα REMOVE = "del"

Δέχεται πολλές παραμέτρους. Όλες είναι δεκαψήφιοι αριθμοί τηλεφώνου που υποδηλώνουν άλλους χρήστες της εφαρμογής.

M → S. Το μήνυμα αυτό αποστέλλεται όταν ο χρήστης διαγράψει κάποιον φίλο από τη λίστα φίλων του. Η εφαρμογή στο κινητό δεν ενημερώνεται μέχρι να λάβει επιβεβαίωση από τον server ότι η διαγραφή έχει καταχωρηθεί.

M ← S. Το μήνυμα αυτό αποστέλλεται όταν η διαγραφή έχει καταχωρηθεί στη βάση δεδομένων στον χρήστη που έστειλε το αίτημα διαγραφής. Η εφαρμογή στο κινητό ως απάντηση διαγράφει την καταχώρηση από τη λίστα φίλων.

3.3.1.5 Μήνυμα BLOCK = "blk"

Δέχεται πολλές παραμέτρους. Όλες είναι δεκαψήφιοι αριθμοί τηλεφώνου που υποδηλώνουν άλλους χρήστες της εφαρμογής.

M → S. Το μήνυμα αυτό αποστέλλεται όταν ο χρήστης αποκλείσει προσωρινά κάποιον φίλο από τη λίστα φίλων του. Η εφαρμογή στο κινητό δεν ενημερώνεται μέχρι να λάβει επιβεβαίωση από τον server ότι ο αποκλεισμός έχει καταχωρηθεί.

M ← S. Το μήνυμα αυτό αποστέλλεται όταν ο αποκλεισμός έχει καταχωρηθεί στη βάση δεδομένων στον χρήστη που αποκλείστηκε αλλά και στο χρήστη που έστειλε το αίτημα αποκλεισμού. Η εφαρμογή στο κινητό ως απάντηση αλλάζει και στους δύο χρήστες την κατάσταση σε “blocked”.

3.3.1.6 Μήνυμα UNBLOCK = "ubl"

Δέχεται πολλές παραμέτρους. Όλες είναι δεκαψήφιοι αριθμοί τηλεφώνου που υποδηλώνουν

M → S. Το μήνυμα αυτό αποστέλλεται όταν ο χρήστης άρει τον προσωρινό αποκλεισμό σε κάποιον φίλο από τη λίστα φίλων του. Η εφαρμογή στο κινητό δεν ενημερώνεται μέχρι να λάβει επιβεβαίωση από τον server ότι ο αποκλεισμός έχει καταχωρηθεί.

M ← S. Το μήνυμα αυτό αποστέλλεται όταν ο γίνει έλεγχος στη βάση ότι ο χρήστης που άρει τον αποκλεισμό είναι αυτός που τον είχε επιβάλει αρχικά και ότι ο φίλος του δεν έχει επιβάλλει και αυτός δικό του αποκλεισμό. Η αποστολή του μηνύματος γίνεται και στους δύο χρήστες. Η εφαρμογή στο κινητό ως απάντηση αλλάζει και στους δύο χρήστες την κατάσταση σε “accepted”.

3.3.1.7 Μήνυμα PEND = "pen"

Δέχεται πολλές παραμέτρους. Όλες είναι δεκαψήφιοι αριθμοί τηλεφώνου που υποδηλώνουν άλλους χρήστες της εφαρμογής.

Το μήνυμα αυτό υποχρεώνει την εφαρμογή στο κινητό να αλλάξει την κατάσταση των χρηστών που τα αναγνωριστικά τους στέλνονται ως παράμετροι σε “pending”. Αυτό συμβαίνει στο ακόλουθο σενάριο. Ο Α και ο Β είναι ενεργοί φίλοι. Όταν ο Α διαγράψει τον Β από φίλο του τότε στον Β, σαν επακόλουθο του αιτήματος διαγραφής, στέλνεται το μήνυμα PEND και στη συσκευή του Β ο Α εμφανίζεται “pending”.

3.3.1.8 Μήνυμα POSITION = "pos"

Δέχεται δύο παραμέτρους.

Παράμετρος1: τρέχων γεωγραφικό μήκος

Παράμετρος2: τρέχων γεωγραφικό πλάτος

Με το μήνυμα αυτό η εφαρμογή στο κινητό τηλέφωνο ενημερώνει τον κεντρικό εξυπηρετητή για αλλαγή στη θέση του χρήστη. Κάθε φορά που ο εξυπηρετητής λαμβάνει ένα τέτοιο μήνυμα γίνεται έλεγχος στα ενεργά ερωτήματα θέσεων και ενημέρωση όλων των ενδιαφερόμενων χρηστών για τη νέα θέση.

3.3.1.9 Μήνυμα CONT_QUERY_ACTIVATE = "cqA"

M → S. Δέχεται δύο παραμέτρους.

Παράμετρος1: ακτίνα αναζήτησης

Παράμετρος2: ώρα τερματισμού αναζήτησης

Το μήνυμα αυτό αποστέλλεται όταν ο χρήστης ενεργοποιήσει την οθόνη που τον ενημερώνει ποιοι φίλοι του βρίσκονται κοντά του. Η ακτίνα που στέλνεται ως πρώτη παράμετρος είναι η ακτίνα μέσα στην οποία αναζητούνται οι φίλοι του, ενώ στη δεύτερη παράμετρο σημειώνεται για πόσο διάστημα θα είναι ενεργή η συγκεκριμένη οθόνη. Αν η παράμετρος2 είναι μηδενική η οθόνη μένει ενεργή για όση ώρα είναι ανοιχτή η εφαρμογή στο κινητό. Διαφορετικά μετά το πέρας του δοσμένου χρόνου η αναζήτηση διακόπτεται (βλ. παράγραφο 3.3.1.10).

M ← S. Δέχεται πολλές παραμέτρους. Όλες είναι δεκαψήφιοι αριθμοί τηλεφώνου που υποδηλώνουν άλλους χρήστες της εφαρμογής.

Το μήνυμα αυτό αποστέλλεται σαν απάντηση στο κινητό και οι παράμετροι είναι τα αναγνωριστικά των χρηστών που βρέθηκαν εντός ακτίνας όταν ξεκίνησε η αναζήτηση.

3.3.1.10 Μήνυμα CONT_QUERY_DEACTIVATE = "cqD"

Δεν δέχεται παραμέτρους

M → S. Το μήνυμα αυτό αποστέλλεται όταν ο χρήστης σταματήσει την αναζήτηση για κοντινούς φίλους. Η εφαρμογή στο κινητό δεν ενημερώνεται μέχρι να λάβει επιβεβαίωση από τον server ότι η αναζήτηση έχει σταματήσει.

M ← S. Το μήνυμα αυτό αποστέλλεται στο κινητό ως ένδειξη ότι η αναζήτηση κοντινών φίλων έχει σταματήσει είτε από λήξη του χρονικού διαστήματος μιας τρέχουσας αναζήτησης είτε ως απάντηση σε αίτημα διακοπής από το χρήστη.

3.3.1.11 Μήνυμα CONT_QUERY = "ctq"

Δέχεται πολλές παραμέτρους. Όλες είναι προσημασμένοι δεκαψήφιοι αριθμοί τηλεφώνου που υποδηλώνουν άλλους χρήστες της εφαρμογής.

Σε απάντηση των αλλαγών θέσεων κάποιων χρηστών οι ενδιαφερόμενοι γι' αυτούς λαμβάνουν μηνύματα CONT_QUERY όπου: αν ο αριθμός είναι θετικός τότε σημαίνει ότι ο εν λόγω χρήστης είναι εντός ακτίνας και πρέπει να προστεθεί στη λίστα αν δεν υπάρχει ήδη, ενώ αν ο αριθμός είναι αρνητικός σημαίνει ότι ο εν λόγω χρήστης βρίσκεται εκτός ακτίνας και πρέπει να αφαιρεθεί από τη λίστα.

3.3.1.12 Μήνυμα DIRECT_QUERY = "drq"

M → S. Δέχεται μία παράμετρο.

Παράμετρος1: αναγνωριστικό χρήστη

Το μήνυμα αυτό αποστέλλεται όταν κάποιος χρήστης ζητήσει να ενημερωθεί για την απόσταση που πρέπει να διανύσει και την κατεύθυνση που πρέπει να ακολουθήσει για να βρει κάποιον φίλο του ο οποίος προσδιορίζεται από την παράμετρο 1.

M ← S. Δέχεται μία ή τέσσερις παραμέτρους.

Παράμετρος1: αναγνωριστικό χρήστη

Παράμετρος2 (προαιρετική): απόσταση μεταξύ φίλων σε μέτρα

Παράμετρος3 (προαιρετική): κατεύθυνση που πρέπει να ακολουθήσει ο χρήστης που έστειλε το ερώτημα για να μεταβεί στην τοποθεσία που είναι ο φίλος του (N, E, S, W…)

Παράμετρος4 (προαιρετική): ώρα και ημερομηνία όπου ο χρήστης που αντιστοιχεί στην παράμετρο 1 ενημέρωσε για τελευταία φορά το σύστημα για την τοποθεσία του.

Το μήνυμα αυτό αποστέλλεται ως απάντηση σε ληφθέν μήνυμα DIRECT_QUERY και ενημερώνει τον χρήστη για το που βρίσκεται ο φίλος του αν η πληροφορία αυτή είναι διαθέσιμη (όταν οι παράμετροι 2-4 υπάρχουν), ενώ σε διαφορετική περίπτωση εμφανίζει μήνυμα ότι η πληροφορία που ζητήθηκε δεν είναι διαθέσιμη (υπάρχει μόνο η παράμετρος 1).

3.3.1.13 Μήνυμα REQUESTS_DONE = "rqs"

Δεν δέχεται παραμέτρους.

M ← S. Το μήνυμα αυτό αποστέλλεται στο χρήστη όταν υπάρχουν στη βάση δεδομένων μηνύματα που δεν του έχουν παραδοθεί ακόμα. Αμέσως γίνεται αποστολή όλων των σε εκκρεμότητα μηνυμάτων και στο τέλος αποστέλλεται το μήνυμα.

M → S. Αφού το κινητό έχει λάβει όλα τα μηνύματα που στάλθηκαν σε αυτό και τα έχει διεκπεραιώσει, μόλις λάβει το μήνυμα REQUESTS_DONE απαντά στον εξυπηρετητή με ένα ίδιο μήνυμα ώστε τα μηνύματα να σβηστούν από τη βάση και να μην ξανασταλθούν σε επόμενη σύνδεση.

3.3.1.14 Μήνυμα UNSUBSCRIBE = "uns"

Δεν δέχεται παραμέτρους.

M → S. Αποστέλλεται όταν ο χρήστης επιλέξει να διαγραφεί από την υπηρεσία.

M ← S. Αποστέλλεται ως απάντηση στο χρήστη ότι η διαγραφή από τη βάση δεδομένων έχει ολοκληρωθεί.

4 *Υλοποίηση*

Στο κεφάλαιο αυτό θα συζητήσουμε λεπτομερώς θέματα υλοποίησης του συστήματος.

4.1 Ενδιαφέροντα σημεία υλοποίησης

Ακολουθούν μερικά σημαντικά σημεία της υλοποίησης που αξίζουν περαιτέρω προσοχής.

4.1.1 Επαλήθευση αριθμού τηλεφώνου

Το προφίλ MIDP δεν περιλαμβάνει βιβλιοθήκες και κώδικα που να επιτρέπει στο πρόγραμμα να ανακτήσει τον αριθμό του τηλεφώνου της κάρτας SIM της συσκευής. Για να λύσουμε το συγκεκριμένο πρόβλημα χρησιμοποιούμε το Wireless Messaging API (που ανήκει στα προαιρετικά πακέτα), και συγκεκριμένα τη δυνατότητα που δίνει στην εφαρμογή να στέλνει και να λαμβάνει μηνύματα κειμένου (SMS). Τα μηνύματα SMS λαμβάνονται «αθόρυβα» από την εφαρμογή χωρίς να εμφανίζονται στα εισερχόμενα της συσκευής.

Ο τρόπος με τον οποίο ανακτούμε τον αριθμό τηλεφώνου του χρήστη είναι ο εξής: η εφαρμογή μπορεί να ορίσει μία πόρτα στην οποία θα ακούει για εισερχόμενα μηνύματα SMS. Η διεύθυνση αποστολής των μηνυμάτων από την εφαρμογή έχει την εξής μορφή: "sms://[number_in_international_format]:[targer_port]". Αν το κομμάτι target_port παραληφθεί τότε το SMS αποθηκεύεται στα εισερχόμενα της συσκευής και η εφαρμογή δεν έχει κανένα τρόπο να το διαβάσει. Αντίθετα αν το target_port είναι ίδιο με αυτό που έχει ορίσει η εφαρμογή ότι ακούει για εισερχόμενα μηνύματα το SMS οδηγείται στην εφαρμογή, η οποία έχει πρόσβαση στον αποστολέα, στην ώρα λήψης και στο περιεχόμενο.

Έτσι λοιπόν, αρχικά προτρέπουμε το χρήστη να καταχωρήσει τον αριθμό του τηλεφώνου του σε ένα πεδίο κειμένου. Στη συνέχεια γίνεται αποστολή ενός κενού μηνύματος SMS στον αριθμό που πληκτρολόγησε ο χρήστης, ενώ η εφαρμογή ακούει για εισερχόμενα μηνύματα. Αν το SMS παραληφθεί μέσα σε 10 δευτερόλεπτα, τότε ο αριθμός που εισήγαγε ο χρήστης

είναι σωστός και αποθηκεύεται στο μόνιμο αποθηκευτικό χώρο της εφαρμογής. Σε διαφορετική περίπτωση ζητείται από το χρήστη επαναπληκτρολόγηση του σωστού αριθμού (Κώδικας 4.1).

```
διεύθυνσηΑποστολής = "sms://+30" + αριθμόςΧρήστη + ":" + πόρτα + "";

εξερχΣύνδεση = νέα σύνδεσηΜηνύματος (διεύθυνσηΑποστολής);
εισερχΣύνδεση = νέα σύνδεσηΜηνύματος (πόρτα);
διακοπήΣύνδεσης = νέα διακοπή (εισερχΣύνδεση, 10sec);

εξερχΣύνδεση.αποστολή(νέο SMS);
μήνυμα = εισερχΣύνδεση.λήψη();

αν (μήνυμα.αποστολέας == αριθμόςΧρήστη)
{
  έγκυροςΑριθμός = αληθές;
  αποθηκευτικόςΧώροςΑριθμού = νέος αποθηκευτικόςΧώρος ("αριθμός");
  αν (αποθηκευτικόςΧώροςΑριθμού.υπάρχειΚαταχώρηση)
    αποθηκευτικόςΧώροςΑριθμού.διαγραφή(επόμενηΚαταχώρηση);
  αποθηκευτικόςΧώροςΑριθμού.καταχώρησε(αριθμόςΧρήστη);
}
αλλιώς
{
  έγκυροςΑριθμός = ψευδές;
  εμφάνισε("Λάθος αριθμός τηλεφώνου. Επανέλαβε εισαγωγή");
}
```

Κώδικας 4.1. Επικύρωση αριθμού τηλεφώνου χρήστη.

Ο σωστός αριθμός αποθηκεύεται μόνιμα στη συσκευή έτσι η παραπάνω διαδικασία απαιτείται μόνο κατά την πρώτη εκτέλεση της εφαρμογής σε κάθε συσκευή.

4.1.2 Ενημέρωση θέσης / Ερωτήματα θέσης

Οι αλλαγές στα αποτελέσματα των ερωτημάτων θέσης των χρηστών πυροδοτούνται από τις ενημερώσεις νέων θέσεων των κινητών που λαμβάνει ο κεντρικός εξυπηρετητής. Αυτό εξαλείφει την ανάγκη δημιουργίας κάποιας διεργασίας που να ελέγχει ανά τακτά χρονικά διαστήματα και για κάθε χρήστη, αν συνέβη κάποια αλλαγή θέσης στους ενεργούς φίλους του ώστε να τον ενημερώσει κατάλληλα. Επιπλέον η ενημέρωση για τις αλλαγές με αυτόν τον τρόπο είναι αμεσότερη. Υπάρχει βέβαια ανάγκη για αποδοτική αναζήτηση στους φίλους κάθε χρήστη όταν αυτός ενημερώνει τον κεντρικό εξυπηρετητή για μια νέα αλλαγή στην τοποθεσία του.

Όπως έχει αναφερθεί ο εξυπηρετητής κρατάει μια δομή με πληροφορίες για κάθε συνδεδεμένο στο σύστημα χρήστη. Μέσα σ' αυτή τη δομή υπάρχουν δύο λίστες: η πρώτη

περιέχει τους ενεργούς φίλους του και η δεύτερη περιέχει εκείνους τους φίλους του που έχουν ενεργοποιήσει την οθόνη εμφάνισης κοντινών φίλων στη συσκευή τους.

Όταν ένας χρήστης ενεργοποιεί την οθόνη εμφάνισης κοντινών του φίλων τότε το σύστημα διατρέχει τη λίστα των ενεργών φίλων του και τον προσθέτει ως καταχώρηση στη δεύτερη λίστα (τη λίστα των ενδιαφερόμενων για αυτόν) κάθε φίλου του (Κώδικας 4.2).

```
ακτίναΔράσης = δοσμένηΑκτίναΔράσης;
ημερομηνίαΛήξης = δοσμένηΗμερομηνίαΛήξης;

λίσταΕντόςΑκτίνας.διαγραφήΌλων();
για κάθε φίλος στη λίσταΕνεργοίΦίλοι
{
  φίλος.λίσταΕνδιαφερόμενων.πρόσθεσε(τρέχωνΧρήστης);
  απόσταση = φίλος.θέση.απόστασηΩς(τρέχωνΧρήστης.θέση);
  αν (ακτίναΔράσης >= απόσταση) λίσταΕντόςΑκτίνας.πρόσθεσε(φίλος);
}

αν (ημερομηνίαΛήξης != 0)
{
  εργασίαΑποστολήςΛήξης = νέα ΠρογραμματισμένηΕργασία;
  εργασίαΑποστολήςΛήξης.επόμενηΕκτέλεση(ημερομηνίαΛήξης);
}
```

Κώδικας 4.2. Ενεργοποίηση εμφάνισης κοντινών φίλων.

Με αυτό τον τρόπο, είναι εύκολο και γρήγορο όταν το σύστημα λάβει μια αλλαγή θέσης ενός χρήστη, να διατρέξει τη λίστα των ενδιαφερόμενων για το συγκεκριμένο χρήστη (και όχι τη λίστα με όλους τους ενεργούς του φίλους) και να αποστείλει σε αυτούς τα κατάλληλα μηνύματα. Εκεί που είναι απαραίτητο να διατρέξουμε τη λίστα με όλους τους ενεργούς φίλους ενός χρήστη είναι στην περίπτωση που η αλλαγή θέσης προέλθει από χρήστη που έχει ενεργοποιημένη την οθόνη εμφάνισης κοντινών φίλων (Κώδικας 4.3).

Ας δούμε ένας παράδειγμα:

Οι χρήστες Α, Β και Γ είναι φίλοι. Ο χρήστης Α ενεργοποιεί την οθόνη εμφάνισης κοντινών φίλων στη συσκευή του, ενώ οι Β και Γ όχι. Το σύστημα ελέγχει τη λίστα ενεργών φίλων του Α και βρίσκει τους Β και Γ. Στη συνέχεια τοποθετεί τον Α στη λίστα ενδιαφερόμενων του Β και του Γ. Έτσι όταν ο Β ανανεώσει τη θέση του, το σύστημα ψάχνει τη λίστα ενδιαφερόμενων και βρίσκει τον Α οπότε στέλνει σε αυτόν το κατάλληλο μήνυμα για τον Β (αν είναι εντός ή εκτός ακτίνας), χωρίς να εμπλακεί ο Γ καθόλου αφού αυτός δεν λαμβάνει ενημερώσεις για κοντινούς φίλους. Όταν όμως ο χρήστης Α, που έχει ενεργοποιημένη τη οθόνη εμφάνισης κοντινών φίλων, ανανεώσει τη θέση του, τότε το σύστημα είναι υποχρεωμένο να διατρέξει τη λίστα όλων των ενεργών φίλων του Α και να του αποστείλει τα ανάλογα μηνύματα.

```
αν (ακτίναΔρασης > 0) λίστα = λίσταΕνεργώνΦίλων
αλλιώς λίστα = λίσταΕνδιαφερόμενων

για κάθε φίλος σε λίστα
{
  απόσταση = φίλος.θέση.απόστασηΩς(τρέχωνΧρήστης.θέση));
  αν (φίλος.ακτίναΔράσης > 0)
  {
    αν (φίλος.ακτίναΔράσης >= απόσταση)
      φίλος.αποστολή("+" + τρέχωνΧρήστης);
    αλλιώς
      φίλος.αποστολή("-" + τρέχωνΧρήστης);
  }
  αν (ακτίναΔράσης > 0)
  {
    αν (ακτίναΔράσης >= απόσταση)
      λίσταΑποτελεσμάτων.πρόσθεσε("+" + φίλος);
    αλλιώς
      λίσταΑποτελεσμάτων.πρόσθεσε("-" + φίλος);
  }
}
αποστολή(λίσταΑποτελεσμάτων);
```

Κώδικας 4.3. Αλλαγή θέσης χρήστη και ενημέρωση χρηστών.

4.1.3 Υπολογισμός αποστάσεων

Ο υπολογισμός των αποστάσεων μεταξύ δύο χρηστών της εφαρμογής γίνεται με τη βοήθεια του τύπου του Vincenty. Για τον υπολογισμό χρειαζόμαστε μόνο τις συντεταγμένες των δύο σημείων και της παραμέτρους του ελλειψοειδούς μοντέλου της Γης που θα χρησιμοποιηθεί. Η συγκεκριμένη υλοποίηση χρησιμοποιεί το ελλειψοειδές μοντέλο WGS-84 για το οποίο ισχύει:

Μεγάλη ακτίνα έλλειψης:	6.378.137 m (±2 m)
Μικρή ακτίνα έλλειψης:	6.356.752,3142 m
Πεπλάτυνση:	1 / 298.257223563

Ο αλγόριθμος του Vincenty αποτελείται από μια επαναληπτική διαδικασία που σταματά όταν το σφάλμα γίνει όσο μικρό θέλουμε. Το επαναληπτικό κομμάτι του παραπάνω αλγόριθμου μπορεί να μην συγκλίνει για αντίποδα σημεία στη έλλειψη (στην υδρόγειο). Σε αυτή την περίπτωση ένας περιορισμός στον αριθμό των επαναλήψεων λύνει το πρόβλημα (Κώδικας 4.4).

```
a = μεγάλη ακτίνα ελλειψοειδούς;
b = μικρή ακτίνα ελλειψοειδούς;
φ1 = γεωγραφικό πλάτος τοποθεσίας 1;
φ2 = γεωγραφικό πλάτος τοποθεσίας 2;
L = διαφορά γεωγραφικών μήκων;
f = πεπλάτυνση;
U1 = atan((1−f)×tanφ1);
U2 = atan((1−f)×tanφ2);
λ = L; λ′ = 2π;                    /*(π.χ. 0.06mm σφάλμα)*/
όσο ((αριθμόςΕπανάληψης++ < 10) || ((abs(λ−λ′) > 10^-12))
{
   sinσ = √[ (cosU2×sinλ)² + (cosU1×sinU2 − sinU1×cosU2×cosλ)² ];
   cosσ = sinU1×sinU2 + cosU1×cosU2×cosλ;
   σ = atan2(sinσ, cosσ);
   sinα = cosU1×cosU2×sinλ / sinς;
   cos²α = 1 − sin²α;
   cos2σm = cosσ − 2×sinU1×sinU2/cos²α;
   C = f/16×cos²α×[4+f×(4−3×cos²α)];
   λ′ = λ;
   λ = L + (1−C)×f×sinα×{σ+C×sinσ×[cos2σm+C×cosσ×(−1+2×cos²2σm)]};
}                    /*cos2σm=0 για σημεία στον ίδιο παράλληλο*/
u² = cos²α×(a²−b²)/b²;
A = 1+u²/16384×{4096+u²×[−768+u²×(320−175×u²)]};
B = u²/1024×{256+u²×[−128+u²×(74−47×u²)]};
Δσ = B×sinσ×{cos2σm+B/4×[cosσ×(−1+2×cos²2σm) −
             B/6×cos2σm×(−3+4×sin²σ)×(−3+4×cos²2σm)]};
s = b×A×(σ−Δσ); /*η ζητούμενη απόσταση*/
```

Κώδικας 4.4. Αλγόριθμος υπολογισμού απόστασης μεταξύ δύο σημείων στην υδρόγειο.

Ο αλγόριθμος οφείλεται στον Thaddeus Vincenty και δημοσιεύτηκε το 1975.

4.1.4 Αποστολή μηνυμάτων

Τα μηνύματα που αποστέλλονται μεταξύ των κινητών τηλεφώνων και του κεντρικού εξυπηρετητή μπορούν να χωριστούν εννοιολογικά σε τρεις κατηγορίες:

- σε μηνύματα που αφορούν τη διαχείριση της λίστας φίλων ενός χρήστη (ADD, REMOVE, BLOCK, UNBLOCK, PEND)
- σε μηνύματα σχετικά με ερωτήματα θέσης (POSITION, DIRECT_QUERY, CONT_QUERY, CONT_QUERY_ACTIVATE, CONT_QUERY_DEACTIVATE)
- σε διαχειριστικά μηνύματα (μήνυμα σύνδεσης, TERMINATE, REQUESTS_DONE, UNSUBSCRIBE)

Όταν ένας συνδεδεμένος χρήστης στέλνει ένα μήνυμα στον κεντρικό εξυπηρετητή, λαμβάνει πάλι ένα μήνυμα ως απάντηση. Μερικά μηνύματα όμως έχουν ως αποτέλεσμα και την αποστολή μηνυμάτων προς άλλους χρήστες (συνδεδεμένους ή όχι). Μηνύματα σχετικά με ερωτήματα θέσης καθώς και τα μηνύματα σύνδεσης και τερματισμού, μπορεί να πυροδοτήσουν αποστολή μηνυμάτων και προς άλλους συνδεδεμένους στο σύστημα χρήστες. Για παράδειγμα, η αλλαγή θέσης ενός χρήστη ή η αποσύνδεσή του πυροδοτεί αποστολή μηνυμάτων προς όλους τους φίλους του που έχουν ενεργοποιημένη την οθόνη εμφάνισης κοντινών φίλων. Αν κάποιος από τους παραλήπτες, αποσυνδεθεί δε θα λάβει το μήνυμα, αλλά

αυτό δεν είναι μειονέκτημα του συστήματος αφού τα συγκεκριμένα μηνύματα (προσθήκης ή αφαίρεσης από τη λίστα κοντινών φίλων) έχουν νόημα μόνο εκείνη τη στιγμή που στέλνονται. Όταν ο χρήστης που αποσυνδέθηκε επανασυνδεθεί στο σύστημα, ενεργοποιώντας την οθόνη εμφάνισης κοντινών φίλων θα λάβει όλη την απαραίτητη πληροφορία.

Αντίθετα τα μηνύματα που ανήκουν στην πρώτη κατηγορία, πυροδοτούν πάντα μήνυμα που πρέπει να σταλθεί και στον φίλο του χρήστη (αυτόν για τον οποίο έγινε η ενέργεια). Για παράδειγμα προσθήκη του χρήστη Β από τον Α, πυροδοτεί μήνυμα που στέλνεται στον Β για να τον ενημερώσει για το αίτημα του Α. Αυτού του είδους τα μηνύματα δεν πρέπει να χάνονται σε περίπτωση όπου ο παραλήπτης βρίσκεται εκτός σύνδεσης. Για το λόγο αυτό υπάρχει ένας πίνακας στη βάση δεδομένων του εξυπηρετητή όπου τα μηνύματα αυτά αποθηκεύονται μαζί με τον παραλήπτη τους. Έτσι κάθε φορά που ένας χρήστης συνδέεται στο σύστημα, το πρώτο πράγμα που γίνεται είναι η ανάκληση και αποστολή όλων των μηνυμάτων που προορίζονται για αυτόν από τη βάση δεδομένων. Μόλις η εφαρμογή πελάτη επιβεβαιώσει τη λήψη και επεξεργασία τους, αυτά διαγράφονται από τη βάση. Ο μηχανισμός αυτός λειτουργεί με έξυπνο τρόπο φιλτράροντας ίδια αλλά και αντίθετα μηνύματα (Κώδικας 4.5). Για παράδειγμα ένα αίτημα προσθήκης νέου φίλου μπορεί να σταλεί πολλές φορές αλλά θα αποθηκευτεί μία φορά στη βάση. Ακόμη ένα μήνυμα αποκλεισμού αποθηκευμένο στη βάση αλληλοαναιρείται από ένα μήνυμα άρσης αποκλεισμού με τον ίδιο παραλήπτη και αποστολέα.

```
αν (μήνυμα.τύπος == ADD)
  αντίθετοΜήνυμα = νέο μήνυμα(μήνυμα.περιοχόμενο, REMOVE);
αλλιώς αν (μήνυμα.τύπος == BLOCK)
  αντίθετοΜήνυμα = νέο μήνυμα(μήνυμα.περιοχόμενο, UNBLOCK);
αλλιώς αν (μήνυμα.τύπος == UNBLOCK)
  αντίθετοΜήνυμα = νέο μήνυμα(μήνυμα.περιοχόμενο, BLOCK);

αν (βάση.πίνακαςΜυνημάτων.υπάρχειΜήνυμα(αντίθετοΜήνυμα))
  βάση.πίνακαςΜηνυμάτων.διαγραφή(αντίθετοΜήνυμα);

αν (βάση.πίνακαςΜυνημάτων.υπάρχειΜήνυμα(μήνυμα))
  βάση.πίνακαςΜηνυμάτων.ενημέρωση(μήνυμα, τρέχουσαΗμερομηνία);
```

Κώδικας 4.5. Εισαγωγή μηνύματος προς εκτός σύνδεσης χρήστη στη βάση.

Όπως φαίνεται και στον παραπάνω κώδικα αρχικά δημιουργείται αντίθετο μήνυμα και αναζητείται στη βάση. Αν αυτό βρεθεί τότε διαγράφεται από τη βάση έτσι ώστε να μη γίνει καμία ενέργεια με βάση το εισερχόμενο μήνυμα αφού αλληλοαναιρέθηκε από το αντίθετο. Στη συνέχεια γίνεται έλεγχος αν υπάρχει το ίδιο μήνυμα και τότε γίνεται αντικατάσταση.

4.1.5 Διαχείρισης λίστας φίλων

Στην ενότητα αυτή αναλύονται οι μεταβάσεις στην κατάσταση ενός φίλου ως αποτέλεσμα των ενεργειών που κάνει ο χρήστης. Οι μεταβάσεις γίνονται πιο κατανοητές μέσα από το ακόλουθο παράδειγμα.

Αρχικά οι χρήστες Α και Β δεν έχουν κανένα φίλο. Ο χρήστης Α προσθέτει τον χρήστη Β στους φίλους του. Ο χρήστης Β αποδέχεται τον Α. Έτσι και οι δύο έχουν τον φίλο τους σε κατάσταση accepted.

Στη συνέχεια ένας από τους δύο χρήστες εκτελεί μια ενέργεια (διαγραφή, αποκλεισμό, άρση αποκλεισμού) και στον Πίνακας 4.1 φαίνονται οι μεταπτώσεις στις καταστάσεις και για τους δύο χρήστες.

Ενέργεια		Κινητό χρήστη Α	Κινητό χρήστη Β
-	(0)	-	-
(0) ➔ Ο Α ζητάει να γίνει φίλος του Β ➔	(1)	B (pending)	A (incoming request)
(1) ➔ Ο Β αποδέχεται το αίτημα. ➔	(2)	B (accepted)	A (accepted)
(2) ➔ Ο Α διαγράφει τον Β ➔	(3)	-	A (pending)
(2) ➔ Ο Α αποκλείει τον Β ➔	(4)	B (blocked)	A (blocked)
(4) ➔ Ο Β ξε-αποκλείει τον Α ➔	(4)	B (blocked)	A (blocked)
(4) ➔ Ο Α ξε-αποκλείει τον Β ➔	(2)	B (accepted)	A (accepted)
(4) ➔ Ο Α διαγράφει τον Β ➔	(3)	-	A (pending)
(3) ➔ Ο Β διαγράφει τον Α ➔	(0)	-	-

Πίνακας 4.1. Μεταβάσεις κατάστασης φίλου.

Οι αριθμοί μέσα στις παρενθέσεις υποδηλώνουν μία κατάσταση. Στην στήλη «Ενέργεια» αναφέρεται ο αριθμός της τρέχουσας κατάστασης, η ενέργεια που γίνεται και στη διπλανή στήλη η νέα κατάσταση που ισχύει αφού πραγματοποιηθεί η ενέργεια.

4.2 Πλατφόρμες και προγραμματιστικά εργαλεία

Εδώ περιγράφονται τα χαρακτηριστικά της συγκεκριμένης υλοποίησης, όπως η πλατφόρμα ανάπτυξης και εκτέλεσης, τα προγραμματιστικά εργαλεία και οι απαιτήσεις της εφαρμογής για κάθε υποσύστημα.

Λεπτομέρειες για την πλήρη **διαδικασία εγκατάστασης** της διπλωματικής σε υπολογιστή βρίσκονται στο παράρτημα.

4.2.1 Πλατφόρμες ανάπτυξης

Στον Πίνακας 4.2 καταγράφονται οι τεχνολογίες και οι πλατφόρμες που αναπτύχθηκαν τα επιμέρους συστήματα της υπηρεσίας.

Εφαρμογή κεντρικού εξυπηρετητή	**Java SE**
Εφαρμογή κινητού τηλεφώνου	**Java ME**
Εξυπηρετητής Web	**Apache Tomcat**
Βάση δεδομένων	**MySQL**

Πίνακας 4.2. Πλατφόρμες και τεχνολογίες των υποσυστημάτων.

Αναλύονται ξεχωριστά παρακάτω ορισμένα χαρακτηριστικά κάθε πλατφόρμας.

4.2.1.1 Java SE

Η εφαρμογή του εξυπηρετητή έχει χτιστεί στην πλατφόρμα Java SE. Πρόκειται για μια πλατφόρμα ανάπτυξης εφαρμογών γενικού σκοπού, όπου παρέχει μεγάλη φορητότητα κώδικα. Αποτελείται από μια εικονική μηχανή, που χρησιμοποιείται για να εκτελεί τα προγράμματα, μαζί με ένα σύνολο βιβλιοθηκών (ή πακέτων) που χρειάζονται για να επιτρέψουν πχ. πρόσβαση στο σύστημα αρχείων, διαχείριση συνδέσεων με δίκτυα, δημιουργία γραφικού περιβάλλοντος για το χρήστη και πολλές άλλες πρόσθετες δυνατότητες.

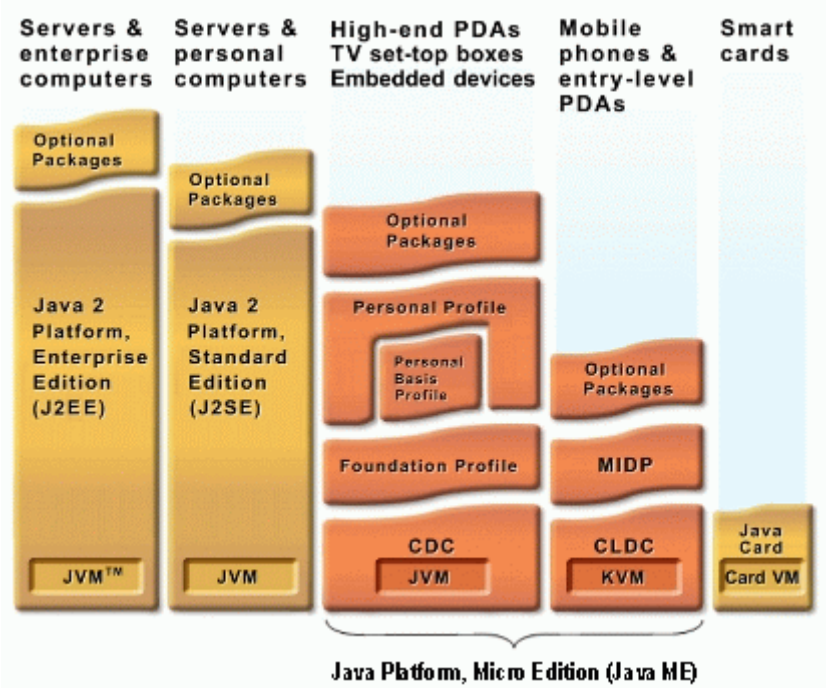
4.2.1.2 Java ME

Η τεχνολογία Java ME που χρησιμοποιήθηκε στην υλοποίηση του προγράμματος-πελάτη αρχικά σχεδιάστηκε για να αντιμετωπίσει τους περιορισμούς που υπάρχουν στην ανάπτυξη εφαρμογών για μικρές συσκευές.

Η πλατφόρμα Java ME είναι ένα σύνολο από τεχνολογίες και τεχνικές προδιαγραφές συνδυασμένες έτσι ώστε να αποτελούν ένα πλήρες περιβάλλον εκτέλεσης εφαρμογών Java, ειδικά σχεδιασμένο να ανταποκρίνεται στις απαιτήσεις κάποιας συσκευής. Οι συσκευές αυτές έχουν περιορισμένους υπολογιστικούς πόρους, όπως μνήμη, υπολογιστική ισχύ και ενεργειακή αυτονομία.

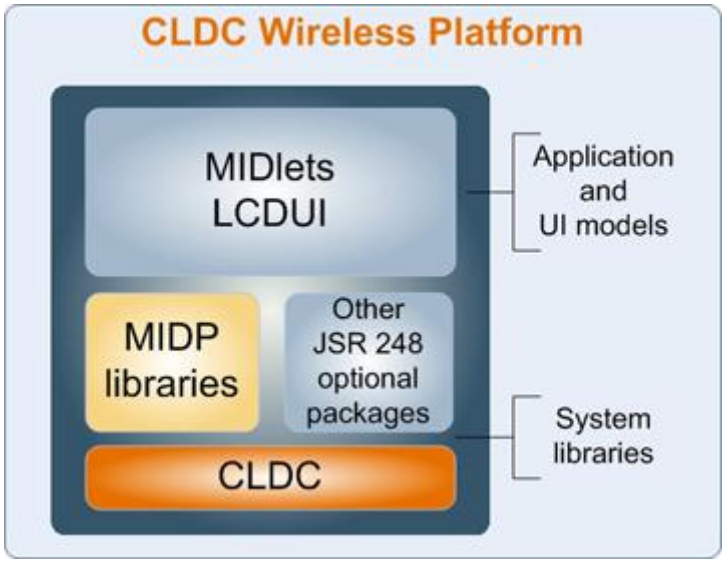
Η τεχνολογία Java ME βασίζεται στα εξής τρία στοιχεία:

- μια ρύθμιση (configuration) παρέχει το πιο βασικό σύνολο βιβλιοθηκών και δυνατοτήτων της εικονικής μηχανής για ένα ευρύ φάσμα συσκευών,
- ένα προφίλ (profile) είναι ένα σύνολο από APIs που υποστηρίζουν ένα πιο περιορισμένο φάσμα συσκευών, και
- προαιρετικά πακέτα (optional packages) που αποτελούνται από αρκετά συγκεκριμένα APIs.



Εικόνα 4.1. Μια γενική εικόνα των συστατικών στοιχείων της τεχνολογίας Java ME και συσχέτιση με άλλες τεχνολογίες Java.

Υπάρχουν δύο ρυθμίσεις: η πρώτη αφορά μικρές συσκευές με περιορισμένους πόρους και ονομάζεται Connected Limited Device Configuration (CLDC) και η δεύτερη αφορά συσκευές με περισσότερες δυνατότητες και ονομάζεται Connected Device Configuration (CDC). Η ανάπτυξη της εφαρμογής μας γίνεται στην πρώτη ρύθμιση, καθώς τα κινητά τηλέφωνα ανήκουν σε αυτήν την κατηγορία.



Εικόνα 4.2. Σχηματική αναπαράσταση των βιβλιοθηκών CLDC – MIDP.

Ένα πολύ ευρέως χρησιμοποιούμενο σχήμα είναι η χρήση του προφίλ Mobile Information Device Profile (MIDP) σε συνδυασμό με το CLDC για τη δημιουργία ενός πλήρες περιβάλλον ανάπτυξης εφαρμογών για κινητά τηλέφωνα και συσκευές με παρόμοια

χαρακτηριστικά και δυνατότητες. Το προφίλ MIDP περιέχει όλες τις απαραίτητες βιβλιοθήκες που χρειάζονται για τη δημιουργία μιας πληθώρας εφαρμογών και παρέχει δυνατότητες όπως δημιουργία γραφικού περιβάλλοντος, διαχείριση πολυμέσων, πρόσβαση στο δίκτυο και λειτουργίες μόνιμης αποθήκευσης. Η αυτόνομη εφαρμογή που δημιουργείται ονομάζεται MIDlet.

Το προαναφερθέν σχήμα έχει χρησιμοποιηθεί και στην ανάπτυξη της εφαρμογής της παρούσας διπλωματικής (CLDC 1.1 + MIDP 2.0).

4.2.1.3 Apache Tomcat

Ο web server τρέχει με τη βοήθεια του web container Tomcat. Ουσιαστικά παρέχει μια υλοποίηση των τεχνολογιών Java Servlet και σελίδων JavaServer. Έτσι μας επιτρέπει να προγραμματίσουμε σε Java τις υπηρεσίες που χρειάζεται να προσφέρει ο web server στους επισκέπτες του.

4.2.1.4 MySQL

Η χρήση της γλώσσας MySQL στην οποία είναι γραμμένη η βάση δεδομένων, μας παρέχει μεγάλη ευελιξία για το λειτουργικό σύστημα στο οποίο θα τρέχει η εφαρμογή καθώς διατίθεται και αυτή για όλα σχεδόν τα λειτουργικά συστήματα που υποστηρίζουν και τη πλατφόρμα Java (υπάρχουν εκδόσεις για πάνω από 20 λειτουργικά συστήματα). Είναι γραμμένη και υλοποιημένη στην τεχνολογία Java, έτσι συνεργάζεται άψογα με τα υπόλοιπα προγράμματα του συστήματος. Έχει ελάχιστες απαιτήσεις σε μνήμη, και παρέχει πολύ γρήγορες απαντήσεις στις εφαρμογές που τη χρησιμοποιούν.

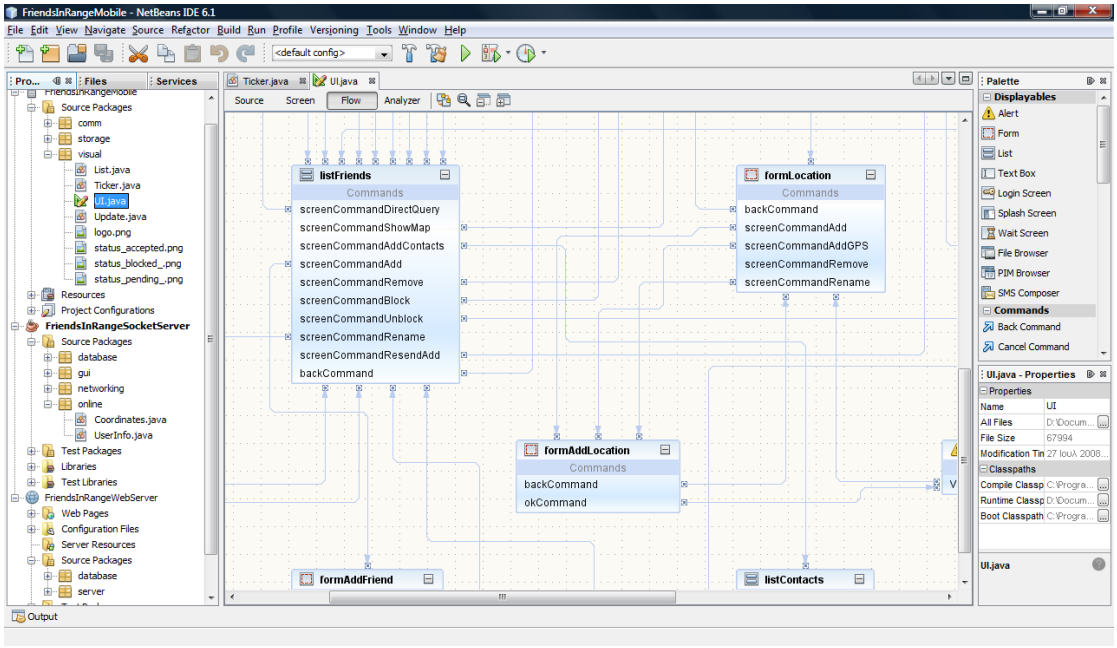
4.2.2 Εργαλεία ανάπτυξης συστήματος

Για την ανάπτυξη του όλου συστήματος χρησιμοποιήθηκε το προγραμματιστικό εργαλείο NetBeans. Πρόκειται για ένα δωρεάν και ανοιχτού κώδικα Ενοποιημένο Περιβάλλον Ανάπτυξης για προγραμματιστές και μηχανικούς λογισμικού. Παρέχει εργαλεία για εύκολη δημιουργία επαγγελματικών εφαρμογών γραφείου, εφαρμογών ιστού αλλά και εφαρμογών για μικρές συσκευές (όπως κινητά τηλέφωνα). Υπάρχει σε εκδόσεις για τα πιο δημοφιλή λειτουργικά συστήματα όπως Windows, Linux, Mac OS X και Solaris.

Το εργαλείο παρέχει:

- ενσωματωμένες λειτουργίες αναζήτησης κώδικα (πχ. αυτόματη επισήμανση κάθε εμφάνισης μιας μεταβλητής ή μιας μεθόδου σε μια κλάση, εμφάνιση όλων των χρήσεων μιας συγκεκριμένης μεθόδου).
- εργαλεία debugging (πχ. εμφάνιση όλων των μεταβλητών στην τρέχουσα εμβέλεια που έχει σταματήσει η εκτέλεση του προγράμματος, multi-threaded debugging).

- εργαλεία σχεδίασης γραφικού περιβάλλοντος (πχ. χρήση drag-n-drop συστατικών για τη δημιουργία του παραθύρου της εφαρμογής, χρήση διαγράμματος ροής ειδικά για τη δημιουργία των mobile εφαρμογών (Εικόνα 4.3)).
- ενσωματωμένη πρόσβαση στους servers (πχ. web servers) και στις βάσεις δεδομένων που χρησιμοποιούν οι εφαρμογές.



Εικόνα 4.3. Το προγραμματιστικό εργαλείο NetBeans.

5 *Έλεγχος*

Ακολουθεί η αξιολόγηση και ο έλεγχος της σωστής λειτουργίας του συστήματος.

5.1 Μεθοδολογία ελέγχου

Ο έλεγχος πραγματοποιήθηκε με τη χρήση του σεναρίου λειτουργίας:

Θεωρούμε δύο χρήστες του συστήματος: το Γιώργο και το Γιάννη. Αρχικά ελέγχουμε τη σωστή λειτουργία του συστήματος προσπαθώντας να τους κάνουμε φίλους. Αφού γίνουν φίλοι μεταξύ τους στην υπηρεσία, εκτελούν ερωτήματα ο ένας για τον άλλο, ενώ μετακινούνται και παράλληλα μεταβάλλουν τις καταστάσεις τους.

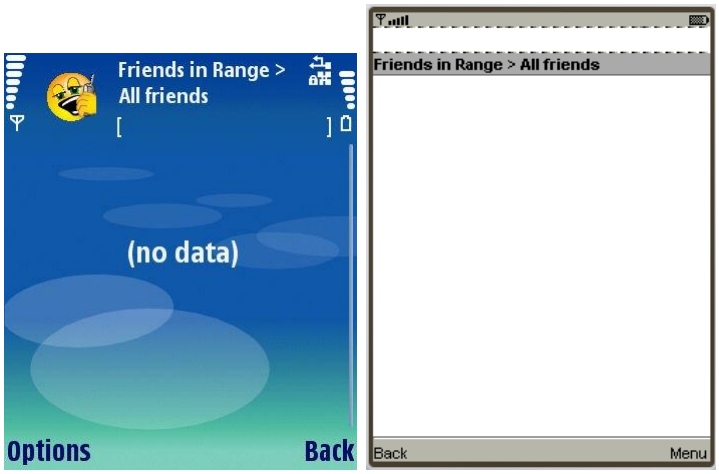
5.1.1 Σενάριο ελέγχου

Αρχικά ο Γιάννης και ο Γιώργος δεν είναι φίλοι. Ο Γιώργος είναι στο σπίτι του, ενώ ο Γιάννης είναι στην Πολυτεχνειούπολη Ζωγράφου. Ο Γιώργος στέλνει στον Γιάννη ένα αίτημα φίλου. Την πρώτη φορά ο Γιάννης απορρίπτει την πρόσκληση. Ο Γιώργος επαναλαμβάνει την αποστολή του αιτήματος. Ο Γιάννης τη δεύτερη φορά αποδέχεται την πρόσκληση. Ο Γιώργος ρωτάει για την απόσταση του Γιάννη, και στη συνέχεια ζητάει την τοποθεσία του στο χάρτη. Ο Γιάννης ενεργοποιεί την εμφάνιση λίστας κοντινών φίλων για μία ακτίνα 3χλμ. Ο Γιώργος ξεκινάει και κατευθύνεται προς την Πολυτεχνειούπολη Ζωγράφου. Λίγο πριν φτάσει στην Πολυτεχνειούπολη Ζωγράφου, ο Γιώργος αποκλείει τον Γιάννη από να λαμβάνει πληροφορίες για τη θέση του, μετά από ένα λεπτό άρει τον αποκλεισμό. Ο Γιώργος φτάνει στην Πολυτεχνειούπολη Ζωγράφου και συναντιέται με το Γιάννη. Αργότερα ο Γιάννης, χωρίς να απενεργοποιήσει την εμφάνιση λίστας κοντινών φίλων ταξιδεύει προς τα ιστορικά κτήρια του Πολυτεχνείου στο κέντρο της Αθήνας. Όταν φτάνει ζητάει να δει την απόστασή του από το Γιώργο.

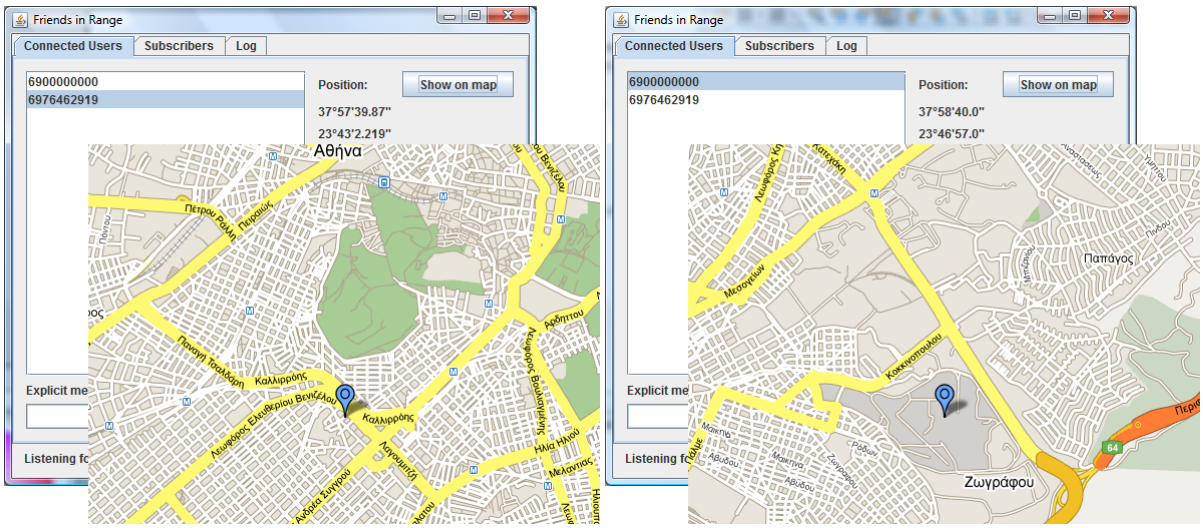
5.2 *Αναλυτική παρουσίαση ελέγχου*

Στην ενότητα αυτή παρουσιάζουμε αναλυτικά τον έλεγχο του συστήματος σύμφωνα με το σενάριο που περιγράφηκε στην προηγούμενη ενότητα. Πολλές από τις λειτουργίες έχουν ήδη περιγραφεί ως παραδείγματα στο κεφάλαιο 2.2.

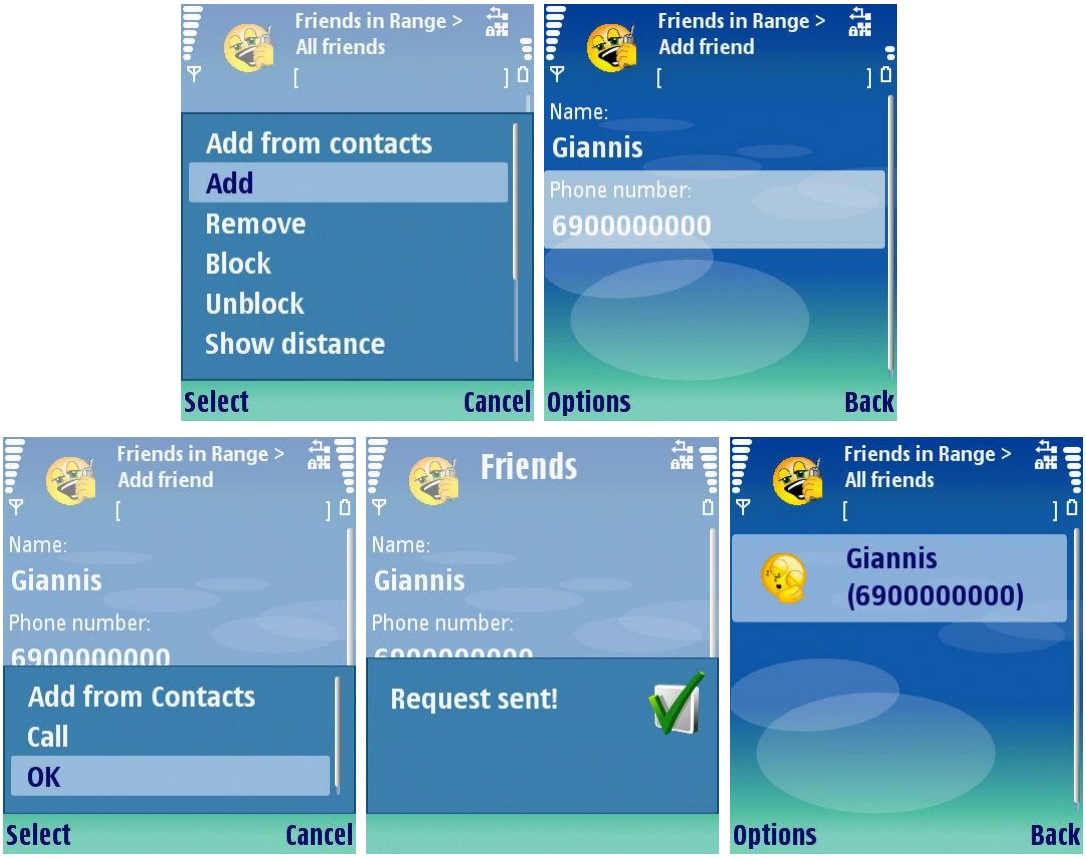
Ακολουθώντας το σενάριο, δείχνουμε σε κάθε βήμα τις οθόνες των κινητών των δύο χρηστών, καθώς και τα μηνύματα που ανταλλάσσονται μέσω του εξυπηρετητή. Οι εικόνες είναι από το κινητό του Γιώργου (6976462919), Nokia N80 (αριστερή Εικόνα 5.1), ενώ από τον εξομοιωτή της Sun (δεξιά Εικόνα 5.1) είναι το κινητό του Γιάννη (6900000000).



Εικόνα 5.1. Αρχικά ο Γιάννης και ο Γιώργος δεν είναι φίλοι.



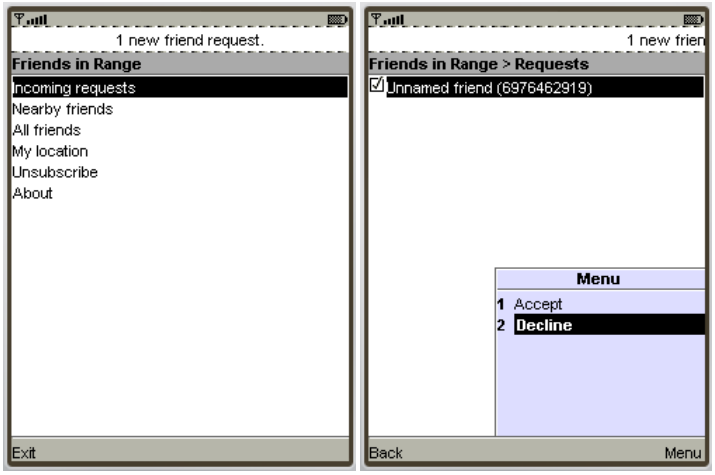
Εικόνα 5.2. Ο Γιώργος είναι στο σπίτι του, ενώ ο Γιάννης είναι στην Πολυτεχνειούπολη Ζωγράφου.




```
from 6976462919: add$6900000000
to 6900000000: add$6976462919
```

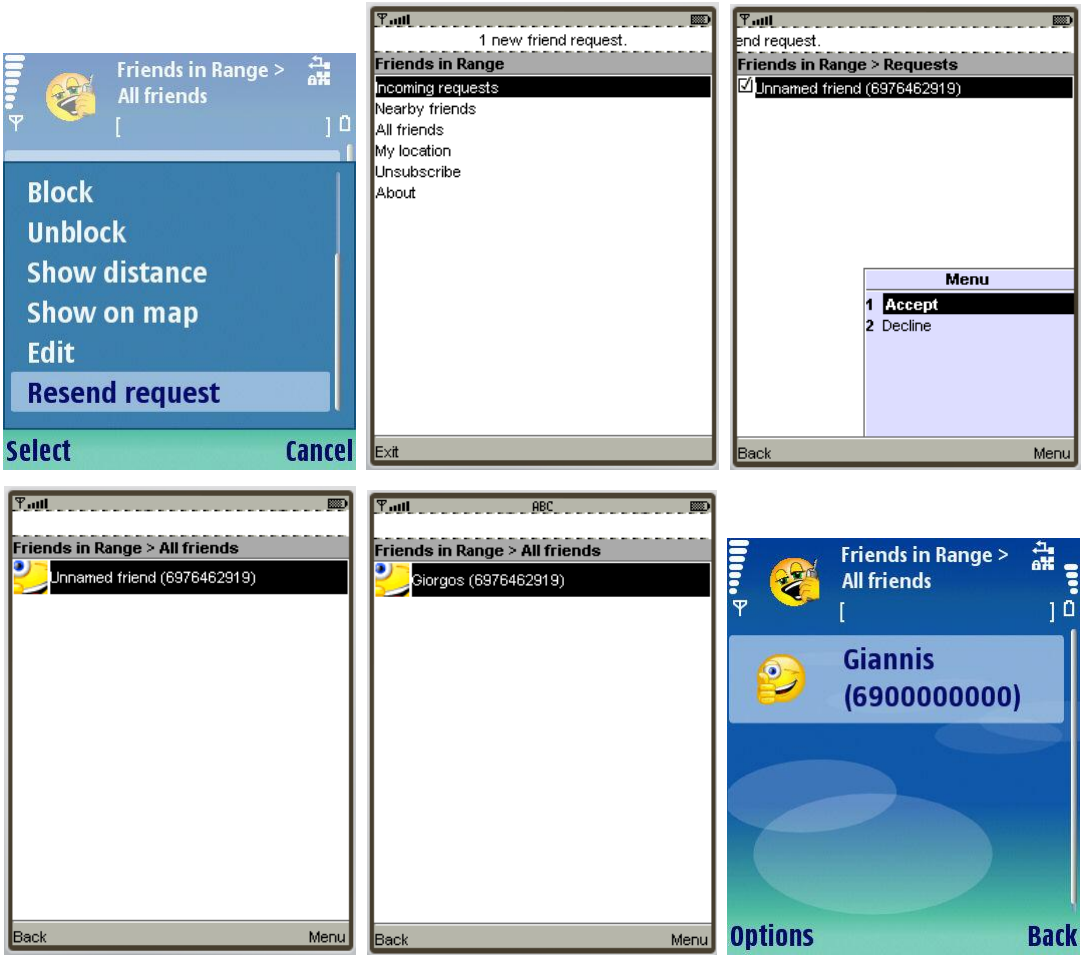
Εικόνα 5.3. Ο Γιώργος στέλνει στον Γιάννη ένα αίτημα φίλου.

Παρατηρούμε ότι τη στιγμή που ο Γιώργος στέλνει αίτημα φίλου, το σύστημα άμεσα προωθεί στο Γιάννη το μήνυμα.



Εικόνα 5.4. Την πρώτη φορά ο Γιάννης απορρίπτει την πρόσκληση.

Παρατηρούμε ότι στην περίπτωση αυτή (απόρριψη πρόσκλησης) δεν ανταλλάσσεται κανένα μήνυμα μεταξύ κινητού και κεντρικού εξυπηρετητή.

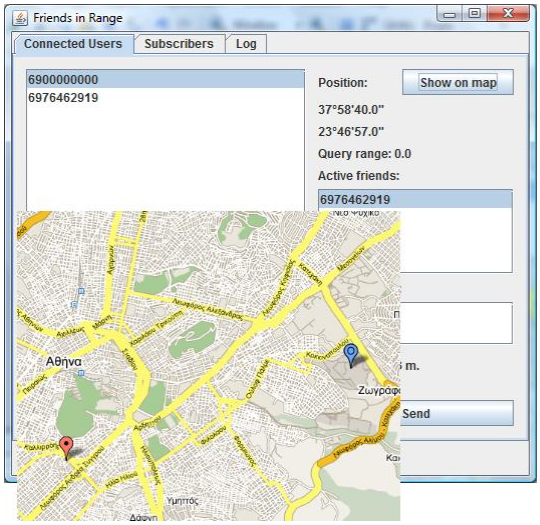



```
from 6976462919: add$6900000000
to 6900000000: add$6976462919
from 6900000000: add$6976462919
to 6976462919: add$6900000000
to 6900000000: add$6976462919
```

Εικόνα 5.5. Ο Γιώργος επαναλαμβάνει την αποστολή του αιτήματος. Ο Γιάννης τη δεύτερη φορά αποδέχεται την πρόσκληση.

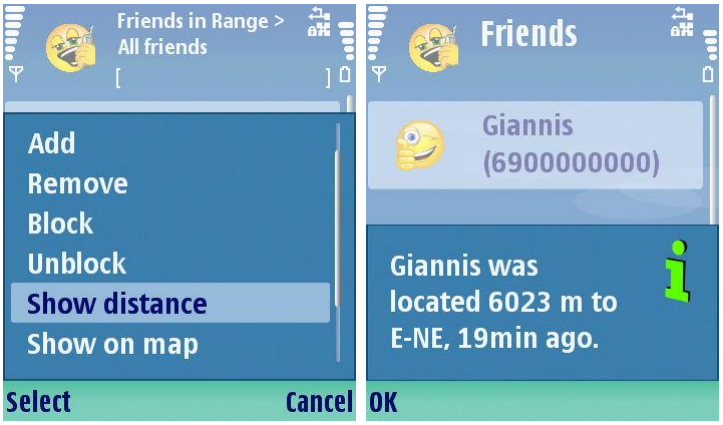
Παρατηρούμε ότι ο Γιάννης δεν είχε αποθηκευμένο το τηλέφωνο του Γιώργου στις επαφές του επομένως η πρόσκληση είχε το προεπιλεγμένο όνομα "Unnamed friend". Στη συνέχεια ο Γιάννης χρησιμοποιώντας την λειτουργία "Edit" αλλάζει το όνομα του φίλου του σε Γιώργος.

Από το log μηνυμάτων του εξυπηρετητή βλέπουμε ότι από τη στιγμή που ο Γιάννης αποδέχεται την πρόσκληση μήνυμα επιβεβαίωσης στέλνεται και στους δύο χρήστες και όχι μόνο στο Γιώργο. Αυτό συμβαίνει γιατί τη στιγμή που ο Γιάννης αποδέχεται την πρόσκληση ουσιαστικά προσθέτει το Γιώργο σε κατάσταση "pending" στους φίλους του. Ο εξυπηρετητής λαμβάνει το μήνυμα και εγκαθιστά τη σχέση στη βάση, στέλνοντας στη συνέχεια το μήνυμα επιβεβαίωσης και στους δύο χρήστες, που με τη σειρά τους μεταβάλλουν την κατάσταση του φίλου τους σε "accepted". Η παραπάνω διαδικασία συμβαίνει αρκετά γρήγορα και συνήθως ο χρήστης που απαντάει στην αίτηση (στην συγκεκριμένη περίπτωση ο Γιάννης) δεν προλαβαίνει να δει όλη αυτή τη διαδικασία, παρά μόνο το τελικό αποτέλεσμα. Παρόλα αυτά η αποστολή επιβεβαίωσης και στους δύο χρήστες αυξάνει την αξιοπιστία του συστήματος.

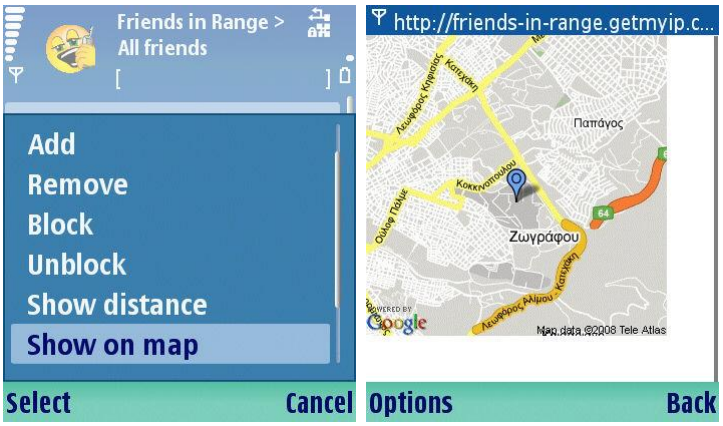


Εικόνα 5.6. Ο διαχειριστής εμφανίζει το χρήστη μαζί με επιλεγμένους φίλους του σε ένα χάρτη.

Τώρα που ο Γιάννης και ο Γιώργος είναι φίλοι, ο διαχειριστής του συστήματος μπορεί να τους βλέπει μαζί σε ένα χάρτη, επιλέγοντας τον πρώτο ως χρήστη (μπλε καρφίτσα) και τον δεύτερο από τη λίστα με τους ενεργούς φίλους (κόκκινες καρφίτσες) (Εικόνα 5.6).

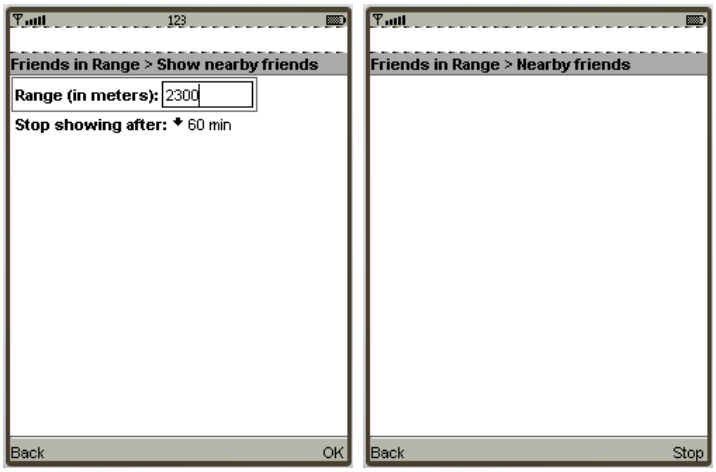



```
from 6976462919: drq$6900000000
to 6976462919: drq$6900000000$6023$E-NE$1221148006475
```



Εικόνα 5.7. Ο Γιώργος ρωτάει την απόσταση του Γιάννη, και στη συνέχεια ζητάει να τον δει στο χάρτη.

Στο κινητό του Γιώργου η εμφάνιση του χάρτη γίνεται άμεσα ενώ η εφαρμογή τρέχει στο παρασκήνιο. Κλείσιμο του browser του κινητού μας επιστρέφει στην εφαρμογή.




```
from 6900000000: cqA$2300.0$3600000
to 6900000000: cqA$
```

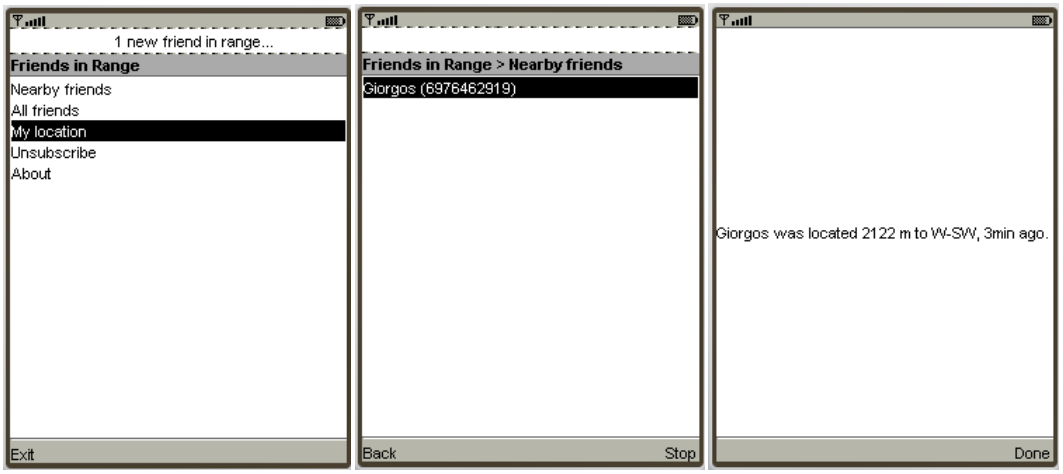
Εικόνα 5.8. Ο Γιάννης ενεργοποιεί την εμφάνιση λίστας κοντινών φίλων για μία ακτίνα 2,3 χλμ.

Αρχικά κανείς δεν εμφανίζεται στη λίστα αφού ο Γιώργος είναι 6023μ μακριά.

```
from 6976462919: pos$37.958509$23.730119
to 6900000000: ctq$-6976462919
from 6976462919: pos$37.964284$23.750183
to 6900000000: ctq$-6976462919
from 6976462919: pos$37.969578$23.754585
to 6900000000: ctq$-6976462919
```

Εικόνα 5.9. Ο Γιώργος ξεκινάει και κατευθύνεται προς την Πολυτεχνειούπολη Ζωγράφου.

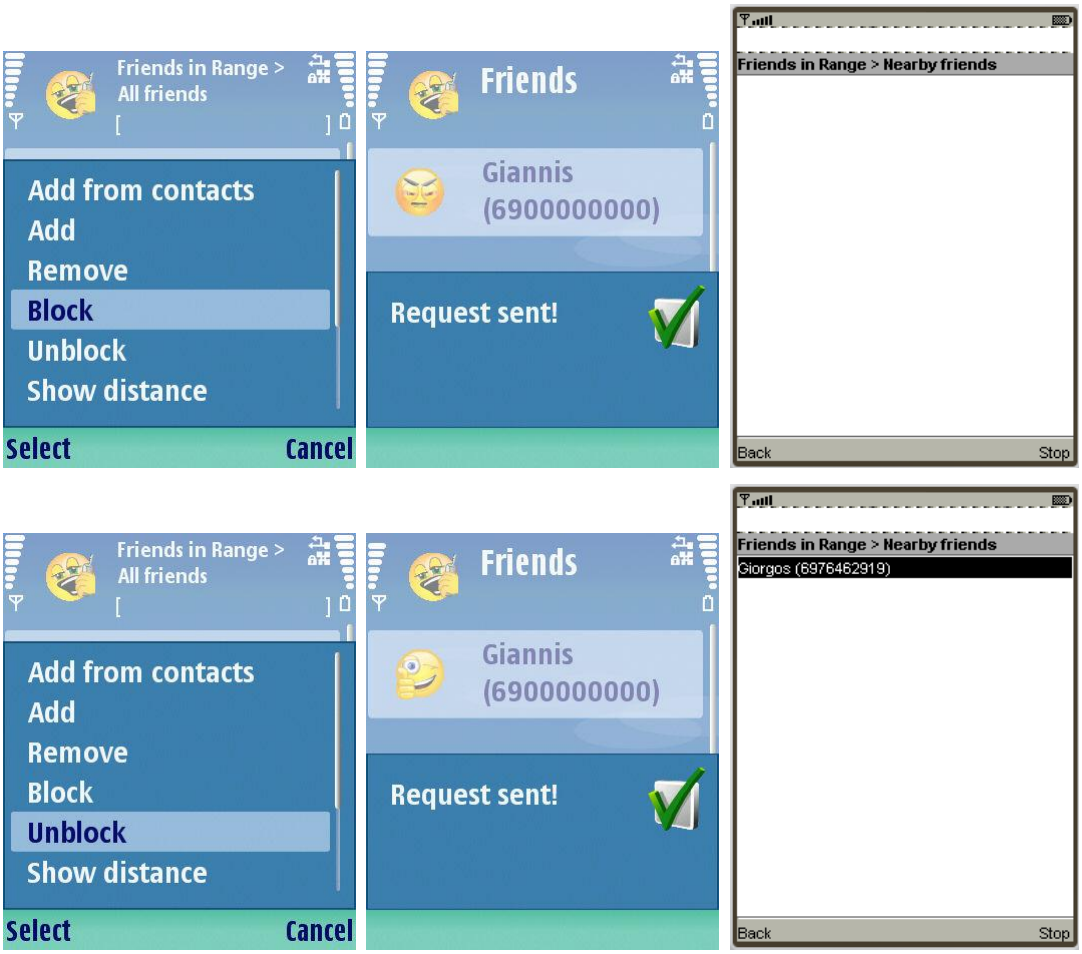
Παρατηρούμε στο log ότι κάθε φορά που ο Γιώργος ενημερώνει τον εξυπηρετητή για τη νέα τοποθεσία του, το σύστημα αποστέλλει στο Γιάννη (που ενδιαφέρεται για τις θέσεις των φίλων του) μήνυμα για το αν ο Γιώργος είναι εντός ή εκτός ακτίνας.




```
from 6976462919: pos$37.972853$23.759169
to 6900000000: ctq$+6976462919
```

Εικόνα 5.10. Όταν ο Γιώργος περνάει έξω από την είσοδο της Πανεπιστημιούπολης Ζωγράφου, είναι πλέον εντός ακτίνας του Γιάννη και εμφανίζεται στην οθόνη του.

Αν ο Γιάννης δεν βρίσκεται στην οθόνη εμφάνισης κοντινών φίλων, τότε εκτός από την ειδοποίηση με δόνηση, εμφανίζεται και αντίστοιχο μήνυμα στο πάνω μέρος της οθόνης "1 new friend in range…", το οποίο παραμένει ενεργό μέχρι ο χρήστης να δει την συγκεκριμένη πληροφορία, ή μέχρι να τερματιστεί αυτόματα η αναζήτηση (λόγω χρονικού ορίου).




```
from 6976462919: blk$6900000000
to 6900000000: ctq$-6976462919
to 6900000000: blk$6976462919
to 6976462919: blk$6900000000
from 6976462919: ubl$6900000000
to 6900000000: ctq$+6976462919
to 6900000000: ubl$6976462919
to 6976462919: ubl$6900000000
```

Εικόνα 5.11. Λίγο αργότερα και πριν φτάσει στην Πολυτεχνειούπολη Ζωγράφου, ο Γιώργος αποκλείει τον Γιάννη από να λαμβάνει πληροφορίες για τη θέση του, μετά από ένα λεπτό άρει τον αποκλεισμό.

Όπως είναι αναμενόμενο όταν ο Γιώργος αποκλείει τον Γιάννη, ο Γιώργος αφαιρείται από την λίστα κοντινών φίλων του Γιάννη, ενώ αντίθετα όταν ο αποκλεισμός αρθεί, ο Γιώργος ξαναπροστίθεται στη λίστα. Ενδιαφέρον παρουσιάζουν και τα μηνύματα που αποστέλλονται μεταξύ του κεντρικού υπολογιστή και των χρηστών. Όταν ο Γιώργος στέλνει μήνυμα και ζητάει τον αποκλεισμό του Γιάννη, ο κεντρικός υπολογιστής στέλνει μήνυμα στο κινητό του Γιάννη για να αφαιρέσει το Γιώργο από τη λίστα κοντινών φίλων, και στη συνέχεια άλλο μήνυμα για να μεταβάλλει την κατάσταση του Γιώργου σε “blocked”. Τέλος στέλνει και μήνυμα επιβεβαίωσης στο κινητό του Γιώργου ώστε και σε αυτό να μεταβληθεί η κατάσταση του Γιάννη σε “blocked”. Ανάλογα μηνύματα ανταλλάσσονται κατά την άρση του αποκλεισμού.

6 *Σχετικές εργασίες*

Θα δούμε τώρα σχετικές με την παρούσα διπλωματική εργασίες, είτε ως παραπλήσιο περιεχόμενο, είτε ως βοηθητικές ή εναλλακτικές μεθόδους που επιλύουν κάποια τεχνικά ζητήματα που αντιμετωπίστηκαν στη διπλωματική.

6.1 Παρεμφερή συστήματα

Ακολουθεί μια σύντομη περιγραφή ορισμένων συστημάτων με παρόμοιο ή παραπλήσιο στόχο, όπως για παράδειγμα το Loopt που είναι ένα παρόμοιο σύστημα εντοπισμού θέσης φίλων για κινητά τηλέφωνα, ή το project Active Campus, ένα σύστημα εντοπισμού φορητών υπολογιστών σε μια πανεπιστημιούπολη.

6.1.1 Loopt

Η υπηρεσία Loopt δείχνει στους χρήστες της που βρίσκονται οι φίλοι τους και τι κάνουν μέσα από λεπτομερείς και διαδραστικούς χάρτες στα κινητά τους τηλέφωνα. Οι χρήστες μπορούν να μοιράζονται ενημερώσεις για την τοποθεσία τους, φωτογραφίες με σήμανση τοποθεσίας λήψης, και σχόλια με τους φίλους τους. Το Loopt είναι διαθέσιμο για αρκετές συσκευές τηλεφώνων, για την νέα πλατφόρμα του iPhone, για κινητά BlackBerry, αλλά και μέσω web. Έχει σχεδιαστεί με κύριο γνώμονα την ασφάλεια και την ιδιωτικότητα των χρηστών του έτσι προσφέρει αποτελεσματική απόκρυψη ευαίσθητων προσωπικών δεδομένων.

6.1.2 Active campus

Το σύστημα αυτό σχεδιάστηκε από το πανεπιστήμιο του San Diego και προσφέρει location-based services για εκπαιδευτικά δίκτυα. Μεταξύ άλλων οι χρήστες του μπορούν να

εντοπίζουν άλλους χρήστες μέσα στην πανεπιστημιούπολη με βάση το σημείο πρόσβασης με από το οποίο συνδέονται.

Υπάρχει ένας χάρτης με όλα τα ασύρματα σημεία πρόσβασης στην πανεπιστημιούπολη και τις τοποθεσίες τους. Όταν ένας χρήστης συνδέεται στο σύστημα, αυτό αναγνωρίζει το σημείο πρόσβαση από το οποίο συνδέθηκε και έτσι προσδιορίζει προσεγγιστικά την τοποθεσία του χρήστη. Το σύστημα δίνει τη δυνατότητα στους χρήστες του να μαθαίνουν την τοποθεσία άλλων χρηστών.

Εικόνα 6.1. Screenshot από την εφαρμογή Active Campus.

6.1.3 Διαχείριση στόλου οχημάτων

Τέτοια συστήματα χρησιμοποιούνται από μεγάλες εταιρείες που κάνουν διανομή προϊόντων σε μεγάλες αποστάσεις. Τα οχήματα εφοδιάζονται με συσκευές εντοπισμού θέσης που στέλνουν το στίγμα τους σε έναν κεντρικό υπολογιστή και εμφανίζονται στο χάρτη. Στη συνέχεια κάποιο λογισμικό μπορεί να αντιπαραβάλει τις εγκεκριμένες διαδρομές των οχημάτων με τις πραγματικές και να ελέγχει για τυχόν αποκλίσεις των οδηγών από τα δρομολόγιά τους.

6.2 Συστήματα στα οποία μπορεί να γίνει επέκταση

Στην ενότητα αυτή μελετάμε συστήματα τα οποία μπορούν να ενσωματωθούν και να επεκτείνουν την παρούσα διπλωματική.

6.2.1 E-communities

Οι ηλεκτρονικές κοινότητες χρηστών είναι πλέον μια πραγματικότητα της καθημερινής μας ζωής. Χρήστες γίνονται μέλη σε αυτές και αρχίζουν να δημιουργούν ηλεκτρονικά δίκτυα φίλων, συναδέλφων, οικογενειών. Έτσι κάθε εφαρμογή που στηρίζεται στο παραπάνω πρότυπο μπορεί να οδηγήσει στη δημιουργία ενός τέτοιου δικτύου ή συνηθέστερα να ενσωματωθεί σε ένα υπάρχον.

Υπάρχοντα social networks όπως το Facebook και το hi5, μπορούν να ενσωματώσουν στις πλατφόρμες τους εφαρμογές τρίτων κατασκευαστών με αποτέλεσμα την αύξηση της ποικιλίας της παρεχόμενης πληροφορίας. Αυτό αυξάνει και τη δημοτικότητα της εφαρμογής αφού προτείνεται από φίλο σε φίλο.

Με αυτό τον τρόπο γίνεται δυνατή η διαχείριση του λογαριασμού του χρήστη στην υπηρεσία, μέσα από το προφίλ του στην κοινότητα.

6.3 Συστήματα εντοπισμού θέσης και εναλλακτικές λύσεις

Παγκόσμια συστήματα εντοπισμού θέσης, όπως το GPS, είναι αρκετά διαδεδομένα αλλά δεν έχουν διεισδύσει αρκετά σε συσκευές όπως τα κινητά τηλέφωνα. Έτσι αναπτύσσονται προγράμματα εντοπισμού θέσης που βασίζονται σε άλλες μεθόδους, κυρίως στις σταθερές και γνωστές τοποθεσίες επίγειων σταθμών πρόσβασης.

6.3.1 Place Lab

Το Place Lab είναι ένα λογισμικό που προσφέρει χαμηλού κόστους και εύκολο στη χρήση εντοπισμό θέσης για υπολογιστικές εφαρμογές που κάνουν χρήση αυτής της πληροφορίας, όπως για παράδειγμα εφαρμογές πλοήγησης. Το σύστημα αυτό προσπαθεί να προσφέρει μια υπηρεσία εντοπισμού θέσης που να λειτουργεί παγκοσμίως, σε εξωτερικούς αλλά και εσωτερικούς χώρους (σε αντίθεση με το GPS που λειτουργεί καλά μόνο σε εξωτερικούς χώρους). Η λήψης πληροφορίας για την τρέχουσα τοποθεσία γίνεται με ιδιοτηκότητα χωρίς να υπάρχει συνεχής επικοινωνία με μια κεντρική υπηρεσία (όπως για παράδειγμα στα κινητά τηλέφωνα όπου η εταιρεία κινητής τηλεφωνίας γνωρίζει τη θέση του τερματικού).

Η προσέγγιση στην επίλυση του προβλήματος βασίζεται στο γεγονός ότι συσκευές όπως φορητοί υπολογιστές, υπολογιστές παλάμης και κινητά τηλέφωνα λαμβάνουν συνεχώς σήματα από επίγειους σταθμούς (όπως wifi hotspots, κεραίες κινητής τηλεφωνίας και συσκευές Bluetooth) που υπάρχουν διασκορπισμένοι στο περιβάλλον. Αυτά τα σήματα έχουν το καθένα κάποιο μοναδικό (ή σχεδόν μοναδικό) αναγνωριστικό που τα ξεχωρίζει από τα υπόλοιπα (για παράδειγμα η διεύθυνση MAC, ή το cell-id). Έτσι λαμβάνοντας σήματα από έναν ή περισσότερους σταθμούς η εφαρμογή είναι σε θέση να εντοπίσει την θέση της

συσκευής στην οποία τρέχει αντιπαραβάλλοντας αυτά τα μοναδικά αναγνωριστικά με κάποια τοπικά αποθηκευμένη λίστα συσχέτισης αναγνωριστικών-τοποθεσιών.

Η παραπάνω προσέγγιση προϋποθέτει ότι αυτή η λίστα που συσχετίζει αναγνωριστικά σταθμών με τοποθεσίες προϋπάρχει. Η εφαρμογή παρέχει αυτόματο τρόπο ενημέρωσης της παραπάνω λίστα με νέους σταθμούς που προστίθενται. Όταν ένα σήμα που ληφθεί προέρχεται από άγνωστο σταθμό η εφαρμογή στέλνει το αναγνωριστικό αυτό μαζί με την εκτίμηση της θέσης του σε έναν κεντρικό υπολογιστή όπου αποθηκεύονται οι λίστες. Η τελική θέση μπορεί να υπολογιστεί από αρκετές τέτοιες ενημερώσεις από διαφορετικούς χρήστες.

7 *Επίλογος*

Ως επίλογο, θα αναφέρουμε κάποια χρήσιμα συμπεράσματα που προέκυψαν, καθώς και κάποιες κατευθύνσεις για μελλοντικές επεκτάσεις της διπλωματικής.

7.1 Σύνοψη και συμπεράσματα

Στην παρούσα διπλωματική ασχοληθήκαμε με την ανάπτυξη ενός συστήματος που παρέχει μια υπηρεσία βασισμένη στη θέση του χρήστη (location based service). Τέτοια συστήματα συναντάμε ολοένα και συχνότερα καθώς ο όγκος της πληροφορίας είναι πλέον τεράστιος, έτσι ο χρήστης επιθυμεί το περιεχόμενο να φιλτράρεται και να του παρουσιάζονται τα πιο σχετικά για αυτόν (και για την τοποθεσία του) αποτελέσματα.

Αναπτύχθηκε τόσο το κομμάτι του κεντρικού υπολογιστή που συντονίζει και ενημερώνει κατάλληλα τους χρήστες, όσο και το κομμάτι της εφαρμογής-πελάτη που τρέχει στα κινητά τηλέφωνα. Τα κινητά τηλέφωνα και άλλες φορητές συσκευές (όπως πχ. τα PDAs) είναι πλέον αναπόσπαστο κομμάτι της καθημερινότητάς μας, επομένως αποτελούν μια πλατφόρμα όπου πρέπει να αναπτυχθούν νέες και συναρπαστικές εφαρμογές. Οι εφαρμογές αυτές αποτελούν πρόκληση καθώς τρέχουν σε συσκευές με περιορισμένες δυνατότητες (όπως υπολογιστική ισχύ και μνήμη), έτσι οι προγραμματιστές που τις αναπτύσσουν πρέπει να προσαρμόσουν ανάλογα τα προγράμματά τους.

Το σύστημα που αναπτύξαμε λύνει το πρόβλημα που αρχικά αναφέραμε στο κεφάλαιο 1.2.1, ενδέχεται όμως βελτιώσεων και επεκτάσεων όπως αναφέρουμε ενδεικτικά παρακάτω.

7.2 Μελλοντικές επεκτάσεις

Παρακάτω αναφέρονται κάποιες ιδέες για επέκταση της παρούσας διπλωματικής.

7.2.1 Εναλλακτικοί τρόποι εύρεσης θέσης

Στην τρέχουσα υλοποίηση στα πλαίσια της διπλωματικής, το κινητό για να εντοπίσει τη θέση του, χρησιμοποιεί είτε το Location API (από τα προαιρετικά πακέτα της Java ME) αν αυτό είναι διαθέσιμο, είτε εναλλακτικά μια λίστα από προεπιλεγμένες τοποθεσίες που επιλέγει κάθε φορά ο χρήστης.

Το Location API στην τρέχουσα έκδοση που δοκιμάστηκε υποστηρίζει εύρεση στίγματος μόνο μέσω GPS. Ακόμη το συγκεκριμένο API, υποστηρίζεται τη στιγμή της συγγραφής από 52 μόνο μοντέλα κινητών τηλεφώνων[2] της αγοράς. Αυτό περιορίζει αρκετά τη λειτουργικότητα της εφαρμογής στα υπόλοιπα μοντέλα. Λύση σε αυτό το πρόβλημα προτείνει ένα λογισμικό σαν το Place Lab (βλ. κεφάλαιο 6.3.1), που θα μπορούσε να ενσωματωθεί στον κώδικα της mobile εφαρμογής και να προσφέρει εναλλακτικές μεθόδους εύρεσης θέσης.

Όσο αφορά τη λειτουργία της μη αυτόματης ρύθμισης θέσης, υπάρχει και εκεί χώρος για βελτιώσεις. Αντί για παράδειγμα ο κάθε χρήστης να είναι αναγκασμένος να εισάγει νέες τοποθεσίες πληκτρολογώντας ονομασία και συντεταγμένες, θα μπορούσε να υπάρχει μια λίστα με τοποθεσίες στον κεντρικό υπολογιστή και οι χρήστες να κατεβάζουν αυτές που τους ενδιαφέρουν. Ακόμη η λίστα του κεντρικού υπολογιστή μπορεί να ενημερώνεται και από τους χρήστες, με το να ανεβάζουν τις δικές τους τοποθεσίες. Τέλος θα δίνεται η δυνατότητα αποστολής τοποθεσιών μεταξύ χρηστών, με τον ίδιο τρόπο που αποστέλλονται αιτήματα για νέους φίλους.

7.2.2 Ανάπτυξη σε διαφορετικές πλατφόρμες

Η Java ME είναι μια αρκετά καλή πλατφόρμα ανάπτυξης της εφαρμογής-πελάτη αφού είναι διαθέσιμη σε ένα μεγάλο εύρος από φορητές συσκευές. Παρόλα αυτά ίσως η γενικότητα που προσφέρει (write once run anywhere) να μην μας δίνει τη δυνατότητα να εκμεταλλευτούμε στο έπακρο όλες τις δυνατότητες που μας προσφέρει μια συσκευή.

Για το λόγο αυτό μια μελλοντική επέκταση θα μπορούσε να είναι η ανάπτυξη της εφαρμογής-πελάτη ειδικά για άλλες πλατφόρμες, όπως για τα λειτουργικό Symbian ή Windows Mobile, τα οποία τρέχουν σε πολλές από τις φορητές συσκευές νέας γενιάς, ή ακόμα και για το νέο iPhone της Apple, ή το PlayStation Portable της Sony.

Μια σημαντική επέκταση είναι η δημιουργία web interface για τους χρήστες ώστε να μπορούν να έχουν πρόσβαση στο λογαριασμό τους στην υπηρεσία και από τον προσωπικό τους υπολογιστή. Η χρησιμότητα αυτής της επέκτασης γίνεται πιο προφανής αν συνδυαστεί με το κεφάλαιο 7.2.3. Το interface αυτό θα προσφέρει στους χρήστες όλες τις δυνατότητες

[2] Περισσότερες και συνεχώς ανανεωμένες πληροφορίες στη διεύθυνση: http://developers.sun.com/mobility/device/device

που προσφέρει και η εφαρμογή-πελάτη που τρέχει στα κινητά τηλέφωνα και πιθανόν μερικές επιπλέον.

7.2.3 Εμπλουτισμός περιεχομένου

Η μόνη πληροφορία που ανταλλάσουν οι χρήστες του συστήματος στην τρέχουσα υλοποίηση είναι αυτή της θέσης τους στο χάρτη. Εμπλουτισμός αυτής της πληροφορίας δημιουργεί ενδιαφέροντες επεκτάσεις για το σύστημα, όπως για παράδειγμα η προσθήκη δραστηριότητας των χρηστών και φυσικά η δυνατότητα αναζήτησης φίλων με βάση τη δραστηριότητά τους.

Για παράδειγμα, σε αναλογία με τα status messages που επιλέγουν οι χρήστες στα πιο δημοφιλή προγράμματα ανταλλαγής άμεσων μηνυμάτων, ένας χρήστης θα μπορούσε να ζητάει να μάθει ποιοι από τους φίλους του έχουν ενεργοποιημένο ένα συγκεκριμένο μήνυμα, ή να ψάχνει λέξεις-κλειδιά στην περιγραφή της δραστηριότητάς τους.

Επιπλέον πληροφορίες θα μπορούσαν να είναι φωτογραφίες των χρηστών που συνοδεύουν τους λογαριασμούς τους στην υπηρεσία, με δυνατότητα επιλογής είτε της φωτογραφίας που έχει επιλέξει ένας χρήστης για τον εαυτό του, είτε αυτής που προτιμάει ο χρήστης που τον προσθέτει ως φίλο του.

Ακόμη, οι χρήστες θα μπορούσαν να αφήνουν σχόλια για τοποθεσίες που επισκέπτονται, ή που επιθυμούν να επισκεφθούν, τα οποία να είναι ορατά στους άλλους χρήστες της υπηρεσίας, ή μόνο στους φίλους τους, ανάλογα με τη ρύθμιση που επιλέγουν όταν τα υποβάλλουν.

Τέλος, ακόμα και οι τρέχουσες λειτουργίες μπορούν να βελτιωθούν με επιπλέον στοιχεία. Για παράδειγμα η οθόνη εμφάνισης κοντινών φίλων μπορεί να τροποποιηθεί και να παρέχει οπτικοακουστική ειδοποίηση στο χρήστη όταν οι κοντινοί φίλοι φτάσουν ένα συγκεκριμένο αριθμό (ή όταν είναι λιγότεροι ή περισσότεροι από ένα συγκεκριμένο αριθμό). Ομάδες φίλων μπορούν να δημιουργηθούν και με ένα παρόμοιο μηχανισμό να υπάρχει ειδοποίηση για μέλη μόνο μερικών ομάδων.

8 *Βιβλιογραφία*

[JAL+04] Programming Java 2 micro edition on symbian OS : a developer's guide to MIDP 2.0 / Martin de Jode ; with Jonathan Allin...[et al.] ; Reviewed by Ivan Litovski...[et al.] ; Managing Editor: Phil Northam ; Assistant Editor: Freddie Gjertsen ; Chichester : John Wiley & Sons, Ltd , c2004

[GIG00] Java 2 micro edition / Eric Giguere ; New York : John Wiley & Sons, Inc , c2000

[CL03] Πλήρες εγχειρίδιο της Java 2 / Laura Lemay, Rogers Cadenhead ; απόδοση Γιάννης Β. Σαμαράς ; Publication Link Αθήνα : Εκδόσεις Μ. Γκιούρδας , c2003

http://www.java-tips.org/java-me-tips/midp

http://wiki.forum.nokia.com/index.php/Java_ME_FAQ

http://code.google.com/apis/maps/documentation/staticmaps

http://www.movable-type.co.uk/scripts/latlong-vincenty.html

http://java.sun.com

http://developers.sun.com/mobility/device/device

http://tomcat.apache.org

http://www.mysql.com

http://www.placelab.org

http://activecampus.ucsd.edu

9 ΠΑΡΑΡΤΗΜΑ

9.1 Εγκατάσταση συστήματος

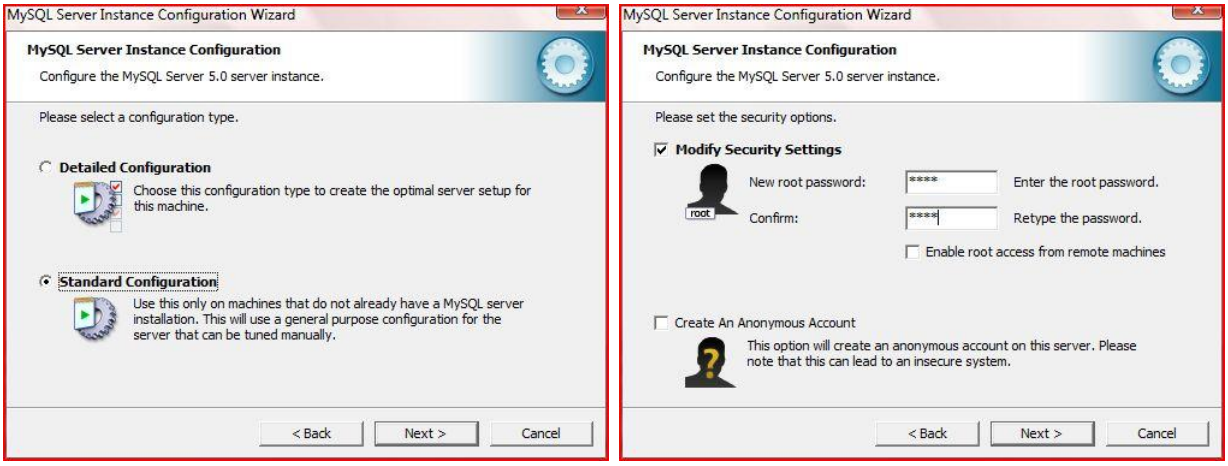
Η εγκατάσταση του συστήματος αποτελείται από δύο μέρη. Από την εγκατάσταση του κεντρικού εξυπηρετητή σε έναν υπολογιστή και από την εγκατάστασης της εφαρμογής-πελάτη στα κινητά τηλέφωνα.

9.1.1 Εγκατάσταση server

Παρακάτω περιγράφονται τα επιμέρους βήματα για την εγκατάσταση των συστατικών στοιχείων της εφαρμογής του εξυπηρετητή.

9.1.1.1 Εγκατάσταση βάσης δεδομένων

Μετάβαση στην τοποθεσία www.mysql.com και λήψη της τελευταίας έκδοσης του MySQL Community Server και εγκατάστασή του χρησιμοποιώντας τις προεπιλεγμένες ρυθμίσεις και επιλέγοντας τυπική εγκατάσταση. Στο τέλος της εγκατάστασης επιλέγουμε να εκτελεστεί η ρύθμιση παραμέτρων (“Configure the MySQL Server now”). Στην αρχική οθόνη ρυθμίσεων επιλέγουμε “Standard configuration” (Εικόνα 9.1.α). Στη συνέχεια επιλέγουμε “Modify security settings” και ως κωδικό του χρήστη “root” πληκτρολογούμε και στα δύο πεδία “toor” (Εικόνα 9.1.β). Στην τελευταία σελίδα με το πάτημα του “Execute” αποθηκεύονται οι ρυθμίσεις μας και ξεκινάει τη λειτουργία του το σύστημα της βάσης δεδομένων.



Εικόνα 9.1. Ρύθμιση παραμέτρων του MySQL Server.

Επόμενο βήμα είναι η δημιουργία της βάσης δεδομένων που χρησιμοποιεί το σύστημα. Για το σκοπό αυτό μπορεί να χρησιμοποιηθεί η κονσόλα γραμμής εντολών της MySQL ή και το αντίστοιχο γραφικό εργαλείο που παρέχεται από την τοποθεσία www.mysql.com. Από την ενότητα Downloads, κάνουμε λήψη του GUI Tools και εγκατάστασή του αφήνοντας τις προεπιλεγμένες ρυθμίσεις και επιλέγοντας πλήρη εγκατάσταση.

Στη συνέχεια εκκινούμε το MySQL Administrator και κάνουμε login στο σύστημα χρησιμοποιώντας το ζεύγος όνομα χρήστη / κωδικού πρόσβασης: "root"/ "toor", που έχουμε ορίσει προηγουμένως.

Επιλέγοντας από την αριστερή στήλη την επιλογή κατάλογοι, κάνουμε δεξί κλικ στο χώρο από κάτω όπου φαίνονται οι ήδη υπάρχουσες βάσεις και επιλέγουμε «Δημιουργία νέου σχήματος» με όνομα "friends". Τώρα μπορούμε να δημιουργήσουμε τους πίνακες τις βάσης πατώντας το κουμπί "Create table" και δημιουργώντας τους πίνακες με τα πεδία που εμφανίζονται στον Πίνακας 9.1.

Όνομα πίνακα	Όνομα πεδίων	Τύπος πεδίων
users	**user_id** last_login last_position_lat last_position_lon	varchar(10) timestamp double double
friendlist	**owner** **friend** ownerStatus friendStatus	varchar(10) varchar(10) enum('on', 'off') enum('on', 'off', 'wait')
requests	recipient message createdOn	varchar(10) text timestamp
admins	**user** pass	varchar(45) varchar(45)

Πίνακας 9.1. Πίνακες βάσης δεδομένων.

Με έντονη γραφή φαίνονται τα πρωτεύοντα κλειδιά κάθε πίνακα. Ο αντίστοιχος κώδικας που δημιουργεί τη βάση και τους πίνακες φαίνεται παρακάτω .

Στον πίνακα με όνομα admins εισάγουμε ένα επιθυμητό ζεύγος username/password. Αυτό το ζευγάρι μας προσφέρει είσοδο στα διαχειριστικά εργαλεία από το web interface της εφαρμογής (κεφάλαιο 9.1.1.2).

Το τελευταίο βήμα είναι η λήψη του αρχείου σύνδεσης της βάσης δεδομένων με το πρόγραμμα του εξυπηρετητή. Από την τοποθεσία www.mysql.com, στην ενότητα Downloads και έπειτα από την επιλογή Connectors, επιλέγουμε και κάνουμε λήψη του Connector/J που είναι ο αντίστοιχος driver για Java. Από το συμπιεσμένο αρχείο που λήφθηκε μας ενδιαφέρει το αρχείο .jar το οποίο είναι ο driver που χρειαζόμαστε για τη σύνδεση. Αποθηκεύουμε αυτό το αρχείο σε έναν υποφάκελο με όνομα lib στο φάκελο όπου είναι αποθηκευμένο και το εκτελέσιμο αρχείο (.jar) της εφαρμογής του εξυπηρετητή.

```
CREATE DATABASE `friends`;
CREATE TABLE `friends`.`users` (
  `user_id` VARCHAR(10) NOT NULL,
  `last_login` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `last_position_lat` DOUBLE NOT NULL DEFAULT 500,
  `last_position_lon` DOUBLE NOT NULL DEFAULT 500,
  PRIMARY KEY (`user_id`)
)
ENGINE = InnoDB;
CREATE TABLE `friends`.`friendlist` (
  `owner` VARCHAR(10) NOT NULL,
  `friend` VARCHAR(10) NOT NULL,
  `ownerStatus` ENUM(`on`, `off`) NOT NULL,
  `friendStatus` ENUM(`on`, `off`, `wait`) NOT NULL,
  PRIMARY KEY (`owner`, `friend`)
)
ENGINE = InnoDB;
CREATE TABLE `friends`.`requests` (
  `recipient` VARCHAR(10) NOT NULL,
  `message` TEXT NOT NULL,
  `createdOn` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
)
ENGINE = InnoDB;
CREATE TABLE `friends`.`admins` (
  `user` VARCHAR(45) NOT NULL,
  `pass` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`user`)
)
ENGINE = InnoDB;
```

Κώδικας 9.1. Κώδικας δημιουργίας βάσης δεδομένων.

Εάν στο μηχάνημα που γίνεται η εγκατάσταση του εξυπηρετητή λειτουργεί τείχος προστασίας (firewall) θα πρέπει η πόρτα 3306 στην οποία γίνονται οι συνδέσεις στη βάση να είναι ανοιχτή.

9.1.1.2 Εγκατάσταση web server

Ο web server που χρησιμοποιεί η εφαρμογή είναι ο Apache Tomcat. Μετάβαση στην τοποθεσία tomcat.apache.org, λήψη της τελευταίας έκδοσης και εγκατάσταση επιλέγοντας

την κανονική εγκατάσταση και όλες τις προεπιλεγμένες ρυθμίσεις. Κατά την προτροπή για κωδικό εισάγουμε "admin" όπως και το όνομα χρήστη.

Στο φάκελο εγκατάστασης του Apache Tomcat υπάρχει ένας φάκελος με όνομα webapps. Εκεί αντιγράφουμε το αρχείο FriendsInRangeWebServer.war το οποίο μόλις ο server εκκινήσει θα επεκταθεί στην αντίστοιχη web εφαρμογή και θα είναι προσβάσιμο από έναν web browser στη διεύθυνση http://localhost:8080/FriendsInRangeWebServer/. Αν εμφανιστεί η σελίδα σημαίνει ότι όλα έχουν εγκατασταθεί σωστά.

9.1.1.3 Ρύθμιση domain και δρομολογητή

Οι εφαρμογές – πελάτη στα κινητά τηλέφωνα είναι ρυθμισμένες έτσι ώστε να συνδέονται σε συγκεκριμένη διεύθυνση με τον κεντρικό εξυπηρετητή. Για το σκοπό αυτό χρησιμοποιήσαμε μια δωρεάν υπηρεσία δυναμικού DNS, την DynDNS, έτσι ώστε σε οποιοδήποτε μηχάνημα να τρέχει ο server αυτό να έχει πάντα την ίδια διεύθυνση. Για τη ρύθμιση αυτή, απαιτείται η λήψη της εφαρμογής DynDNS Updater[3] που αναλαμβάνει ουσιαστικά να αντιστοιχεί την IP διεύθυνση του μηχανήματος που τρέχει με το όνομα domain που έχει κάνει register ο χρήστης μέσω του site www.dyndns.com.

Έτσι αρχικά δημιουργούμε ένα λογαριασμό στο site, και στη συνέχεια κάνουμε register ένα συγκεκριμένο domain. Στη συνέχεια κατεβάζουμε το πρόγραμμα ενημέρωσης της IP και εισάγουμε τις πληροφορίες του λογαριασμού ώστε να συγχρονίζει την IP με το domain της επιλογής μας. Στην τρέχουσα υλοποίηση το domain name που έχει κατοχυρωθεί για τους σκοπούς της εφαρμογής είναι το friends-in-range.getmyip.com. Αν για κάποιο λόγο αλλάξει το domain name θα πρέπει να αλλαχτεί και στον κώδικα των mobile εφαρμογών ώστε να μπορούν να επιχειρούν συνδέσεις στη νέα διεύθυνση.

Μετά την κατοχύρωση του domain, ένα τελευταίο βήμα για τη λειτουργία των χαρτών Google που χρησιμοποιεί η εφαρμογή είναι η λήψη του κλειδιού που αντιστοιχεί στο επιλεγμένο domain. Από την αντίστοιχη σελίδα του Google Code[4], ακολουθώντας τις οδηγίες εισάγουμε το όνομα του domain και πατώντας το κουμπί "Generate API key" εμφανίζουμε τη σελίδα που περιέχει το μοναδικό κωδικό-κλειδί. Αυτό τον κωδικό πρέπει να τον εισάγουμε στα δύο servlets που είναι υπεύθυνα για την εμφάνιση των χαρτών στους χρήστες και στους διαχειριστές της εφαρμογής. Το ισχύον κάθε φορά κλειδί πρέπει να βρίσκεται στο αρχείο key.key στον υφάκελο FriendsInRangeWebServer του φακέλου webapps της εγκατάστασης του Tomcat. Ανοίγουμε το αρχείο με ένα πρόγραμμα επεξεργασίας κειμένου και αντιγράφουμε το νέο κλειδί ως το μοναδικό του περιεχόμενο. Αποθηκεύουμε το αρχείο και επανεκκινούμε τον Tomcat.

[3] http://www.dyndns.com/support/clients

[4] http://code.google.com/apis/maps/signup.html

Τέλος απαιτείται ρύθμιση του μηχανήματος στο οποίο τρέχει η εφαρμογή του server αν αυτό συνδέεται απευθείας στο internet, ή του δρομολογητή στον οποίο είναι συνδεδεμένο το μηχάνημα, αν βρίσκεται πίσω από NAT δίκτυο:

- Αν το μηχάνημα συνδέεται απευθείας στο internet και έχει δημόσια διεύθυνση IP, τότε σε περίπτωση που τρέχει κάποιο firewall, θα πρέπει να ανοιχθούν οι πόρτες 40000 και 40001. Στη συνέχεια πρέπει να ρυθμιστεί η εγκατάσταση του Tomcat ώστε να ακούει για εισερχόμενες συνδέσεις, όχι στην προεπιλεγμένη πόρτα που είναι η 8080 αλλά στην 40001. Για τη ρύθμιση αυτή απαιτείται επεξεργασία του αρχείου server.xml που βρίσκεται στον υποφάκελο conf του κεντρικού φακέλου εγκατάστασης του Tomcat. Στο αρχείο αυτό, χρειάζεται αλλαγή στη γραμμή:

  ```
  <Connector port="8080" protocol="HTTP/1.1" …… />
  ```

  ώστε η παράμετρος port να γίνει 40001, δηλαδή:

  ```
  <Connector port="40001" protocol="HTTP/1.1" …… />
  ```

- Αν το μηχάνημα βρίσκεται πίσω από δίκτυο NAT, έχει δηλαδή ιδιωτική διεύθυνση IP, απαιτείται να γίνει προώθηση θυρών (port forwarding). Για να μπορούν οι εφαρμογές-πελάτη να συνδέονται στον κεντρικό εξυπηρετητή απαιτείται από τη σελίδα ρυθμίσεων του δρομολογητή να γίνουν οι παρακάτω προωθήσεις:
  :40000 → xxx.xxx.xxx.xxx:40000
  :40001 → xxx.xxx.xxx.xxx:8080
  όπου xxx.xxx.xxx.xxx είναι η εσωτερική IP διεύθυνση του μηχανήματος που τρέχει την εφαρμογή server και web server.

9.1.1.4 Εγκατάσταση server

Η εφαρμογή του server αποτελείται από ένα εκτελέσιμο αρχείο Java με όνομα FriendsInRangeSocketServer.jar. Από το φάκελο που βρίσκεται το αρχείο αυτό, και έχοντας ακολουθήσει όλη την προηγούμενη διαδικασία, πληκτρολογούμε στη γραμμή εντολών:

```
java –jar FriendsInRangeSocketServer.jar.
```

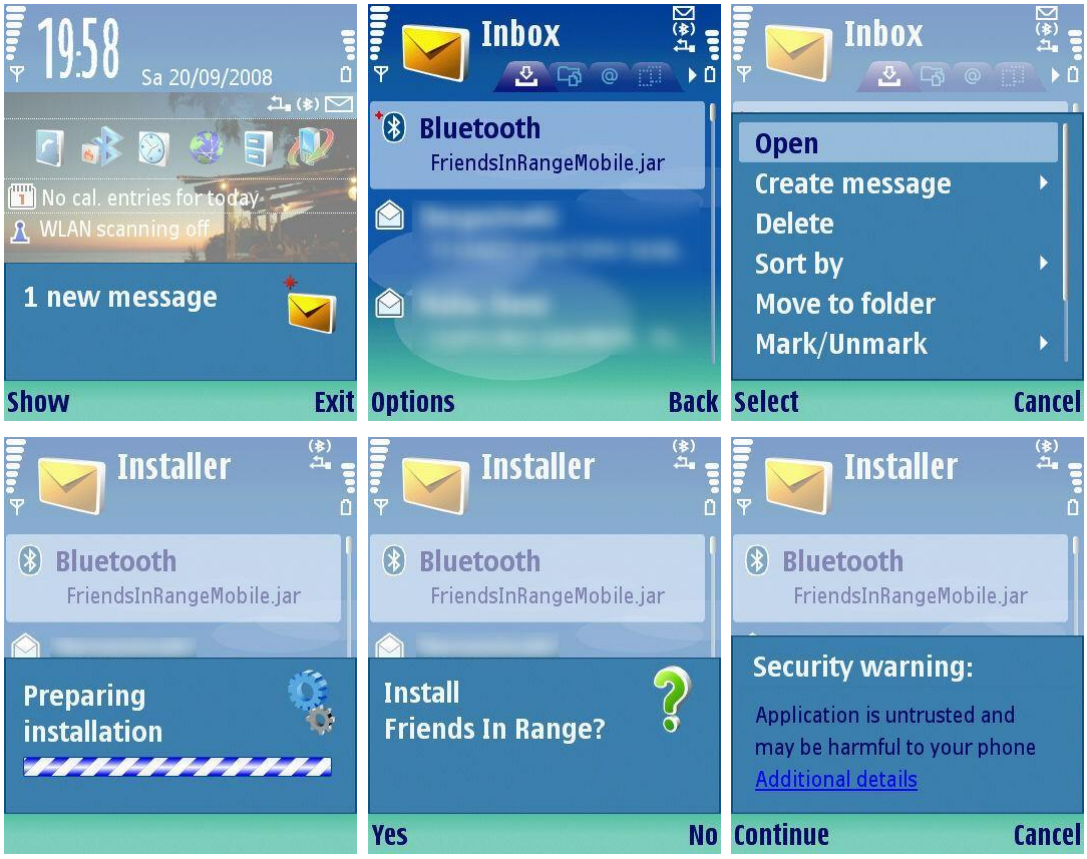
Αν όλα πάνε καλά θα ανοίξει το κύριο παράθυρο της εφαρμογής (Εικόνα 2.14). Αν στο κάτω μέρος της εφαρμογής είναι ορατή η ένδειξη “Listening for incoming connections” τότε η εφαρμογή λειτουργεί κανονικά.

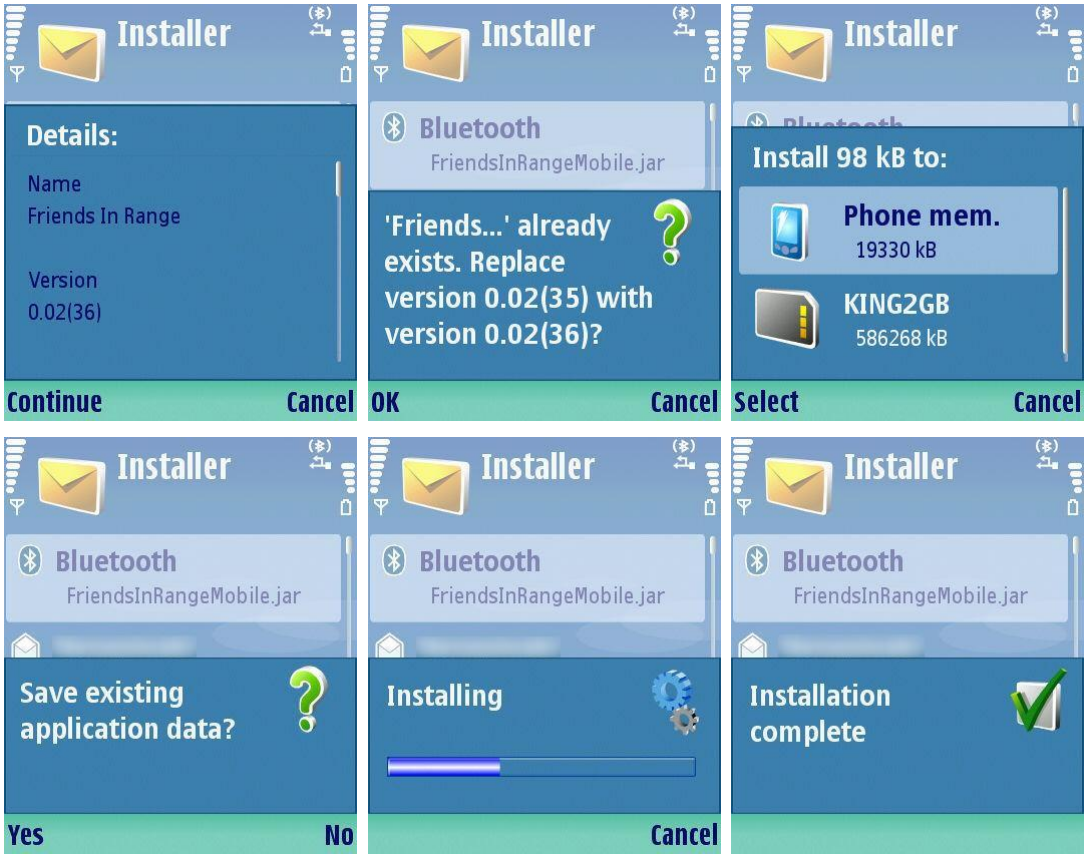
9.1.2 Εγκατάσταση client

Η εφαρμογή πελάτη αποτελείται από ένα .jar αρχείο το οποίο είναι ένα εκτελέσιμο αρχείο Java. Η εγκατάσταση είναι πολύ απλή και απαιτεί την αποστολή του συγκεκριμένου αρχείου

στο κινητό τηλέφωνο είτε με Bluetooth, είτε με υπέρυθρες, είτε με MMS, είτε κατέβασμα του αρχείου απευθείας στο κινητό από το δικτυακό τόπο του προγράμματος.

Αφού το εκτελέσιμο αρχείο ληφθεί στη συσκευή ακολουθούμε τις προτροπές που εμφανίζονται στην οθόνη για την ολοκλήρωση της εγκατάστασης. Απαντάμε καταφατικά σε όλες τις ερωτήσεις. Επειδή η εφαρμογή δεν είναι υπογεγραμμένη ψηφιακά, ίσως εμφανιστεί ένα προειδοποιητικό μήνυμα, που ενημερώνει το χρήστη γι' αυτό. Ο χρήστης πρέπει να επιλέξει να συνεχίσει, διαφορετικά η εγκατάσταση διακόπτεται. Αν υπάρχει ήδη εγκατεστημένη προηγούμενη έκδοση της εφαρμογής, εμφανίζεται ενημερωτικό μήνυμα που ρωτάει αν επιθυμούμε την αντικατάσταση της προηγούμενης έκδοσης από την τρέχουσα. Απαιτείται πάλι καταφατική απάντηση για τη συνέχιση της διαδικασίας. Αν το κινητό έχει πολλαπλούς αποθηκευτικούς χώρους, όπως πχ. εσωτερική μνήμη και αποσπώμενη κάρτα μνήμης, το σύστημα προτρέπει το χρήστη να επιλέξει που επιθυμεί να γίνει η εγκατάσταση. Η επιλογή αυτή είναι καθαρά στην κρίση του χρήστη. Τέλος το σύστημα ρωτάει το χρήστη αν επιθυμεί να κρατήσει ήδη υπάρχοντα δεδομένα της εφαρμογής. Αυτή η ερώτηση εμφανίζεται μόνο όταν γίνεται αντικατάσταση προηγούμενης έκδοσης και πρέπει να απαντηθεί καταφατικά. Σε διαφορετική περίπτωση οι αποθηκευμένες πληροφορίες, όπως ο αριθμός χρήστη και οι λίστες φίλων και τοποθεσιών διαγράφονται (απαιτείται επομένως νέα εγγραφή στην υπηρεσία κατά την πρώτη εκτέλεση της εφαρμογής αφού δεν θα βρεθεί αποθηκευμένος ο αριθμός τηλεφώνου του χρήστη).





Εικόνα 9.2. Εγκατάσταση εφαρμογής στο κινητό τηλέφωνο Nokia N80.

Μετά την εγκατάσταση η εφαρμογή μπορεί να εκκινηθεί από το εικονίδιο που έχει δημιουργηθεί στο φάκελο Εφαρμογές ή στο φάκελο "My own".

9.2 Απαιτήσεις σε υλικό και λογισμικό

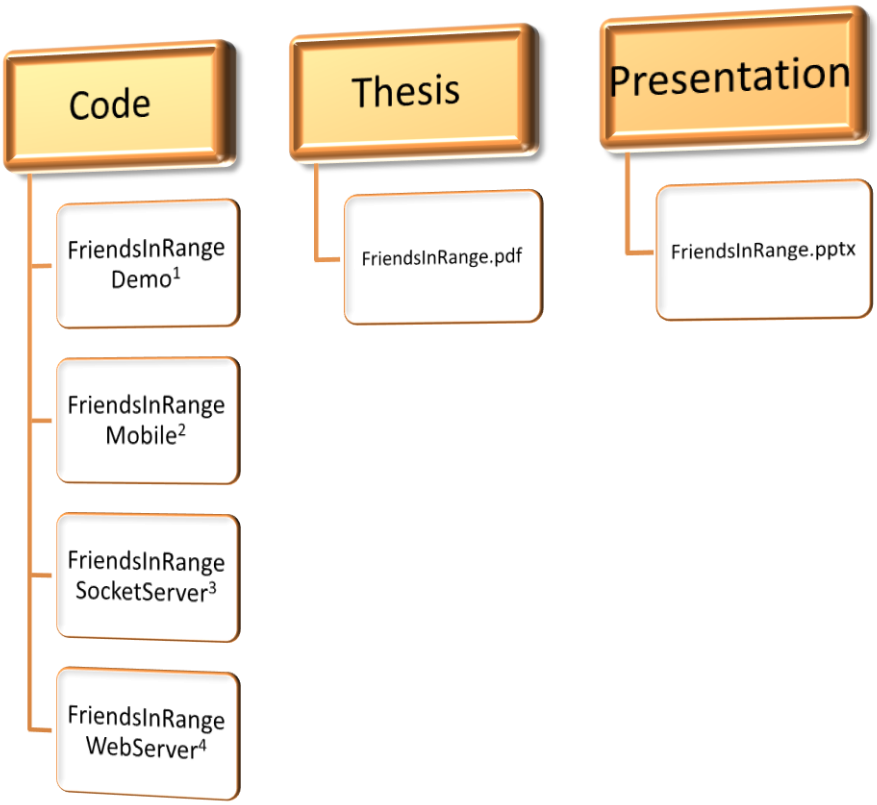
Το μηχάνημα στο οποίο τρέχει ο κεντρικός εξυπηρετητής δεν χρειάζεται να τρέχει κάποιο συγκεκριμένο λειτουργικό καθώς η τεχνολογία Java που χρησιμοποιήθηκε σε όλα τα υποσυστήματα είναι διαθέσιμη για πάνω από 20 διαφορετικά λειτουργικά. Θα πρέπει να έχει όμως συνεχή σύνδεση με το διαδίκτυο και επαρκή μνήμη για να μπορέσει να εξυπηρετήσει όλους τους συνδεδεμένους χρήστες.

Τα κινητά τηλέφωνα στα οποία θα εγκατασταθεί η εφαρμογή θα πρέπει να υποστηρίζουν την πλατφόρμα Java, και συγκεκριμένα το προφίλ MIDP 2.0. Επιπλέον αν υποστηρίζουν το Location API της Java ME, θα μπορούν να χρησιμοποιούν τη λειτουργία αυτόματης ενημέρωσης θέσης σε συνεργασία με έναν εξωτερικό δέκτη GPS.

9.3 *Περιεχόμενα CD*

Ακολουθούν τα περιεχόμενα του CD που συνοδεύει τον παρών τόμο.

Αναλυτικά:



Friends in Range Demo: Η applet εφαρμογή που επιδεικνύει τις βασικές λειτουργίες του συστήματος.

Friends in Range Mobile: Η εφαρμογή Java ME για τα κινητά τηλέφωνα.

Friends in Range Socket Server: Η εφαρμογή Java SE του κεντρικού εξυπηρετητή.

Friends in Range Web Server: Διαχείριση μέσω web interface και χάρτες.

Friends in Range.pdf: Ο παρών τόμος.

Friends in Range.pptx: Η παρουσίαση της διπλωματικής.