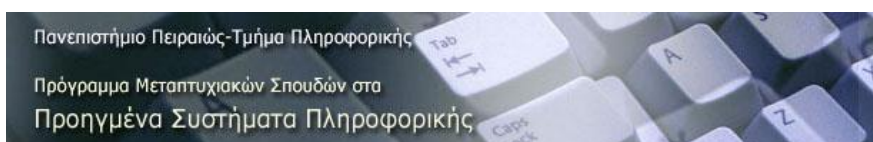




Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Διαδικτυακό Πληροφοριακό Σύστημα Διαχείρισης Έκτακτων Περιστατικών με Δυνατότητα Πρόσβασης μέσω Κινητών Συσκευών με Λειτουργικό Android
Αγγλικός Τίτλος διατριβής	Online Information System of managing Medical Emergencies accessible via Mobile Devices with Android Operating System
Όνοματεπώνυμο Φοιτητή	Χρήστος Μήτσος
Πατρώνυμο	Ευάγγελος
Αριθμός Μητρώου	ΜΠΣΠ/09006
Κατεύθυνση	Δικτυοκεντρικά Πληροφοριακά Συστήματα
Επιβλέπων	Δουληγέρης Χρήστος, Καθηγητής
Καθοδήγηση εργασίας	Δρ. Σαράντης Μητρόπουλος



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Δουληγέρης Χρήστος

Αποστόλου Δημήτρης

Κοτζανικολάου
Παναγιώτης

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τους καθηγητές μου κ. Δουληγέρη Χρήστο για την ανάθεση της μεταπτυχιακής διατριβής, καθώς και τον κ. Μητρόπουλο Σαράντη για την επίβλεψη, την επιστημονικά και τεχνικά άρτια καθοδήγηση στην ανάλυση, σχεδίαση και ανάπτυξη του συστήματος, καθώς και την όλη συνεργασία που είχαμε κατά την διάρκεια της εκπόνησης της μεταπτυχιακής διατριβής.

Θα ήθελα επίσης να ευχαριστήσω την οικογένειά μου για την πολύτιμη υλική, ηθική και πνευματική στήριξή τους καθώς και οποιονδήποτε άλλο συνέβαλε με οποιοδήποτε τρόπο στην δημιουργία της παρούσης εργασίας.

ΠΕΡΙΛΗΨΗ

Η παρούσα μεταπτυχιακή διατριβή ασχολείται με την ανάπτυξη ενός πιλοτικού πληροφοριακού συστήματος που προσομοιώνει την κεντρική υπηρεσία του Εθνικού Κέντρου Άμεσης Βοήθειας (Ε.Κ.Α.Β.). Το όνομα αυτού είναι eEKAB. Η ανάπτυξη του eEKAB είχε ως πρωταρχικό στόχο την βελτιστοποίηση των υπηρεσιών του ΕΚΑΒ και σαν επιμέρους στόχο τη χρήση καινούριων και εξελισσόμενων τεχνολογιών όπως το λειτουργικό Android. Η βασική ιδέα είναι ότι ασθενοφόρα και γιατροί διαθέτουν κινητά Android συνδεδεμένα με έναν κεντρικό server ο οποίος είναι συνδεδεμένος σε μία κεντρική βάση δεδομένων. Όταν ένα περιστατικό φτάσει στο τηλεφωνικό κέντρο του ΕΚΑΒ καταγράφονται τα στοιχεία του περιστατικού και το περιστατικό προωθείται στα κοντινότερα ασθενοφόρα και γιατρούς, μέσω της εφαρμογής που τρέχει στον server, για να σπεύσουν στο περιστατικό άμεσα. Η εφαρμογή των κινητών, με τη σειρά της, είναι υπεύθυνη για την ενημέρωση της κεντρικής βάσης, μέσω του εξυπηρετητή, με την εκάστοτε τοποθεσία των ασθενοφόρων και των γιατρών. Παράλληλα με την ανάπτυξη της εφαρμογής εξετάζονται και οι υπάρχουσες τεχνικές ανά τον κόσμο στον τομέα της διαχείρισης έκτακτων αναγκών όπως επίσης αναλύονται και οι τεχνολογίες που χρησιμοποιήθηκαν στο παρόν πληροφοριακό σύστημα. Τέλος, γίνεται αξιολόγηση του συστήματος ούτως ώστε να γίνει φανερό το κατά πόσο μπορεί ένα τέτοιο σύστημα να βελτιώσει τις υπάρχουσες υπηρεσίες.

ΛΕΞΕΙΣ – ΚΛΕΙΔΙΑ: ΕΚΑΒ, πληροφοριακά συστήματα, android, client-server, GIS, mobile network, πληροφοριακό σύστημα υγείας, αξιολόγηση

ABSTRACT

This thesis deals with the development of a pilot information system that simulates the main services offered by the National Instant Aid Centre (EKAB). Its name is eEKAB. The overriding aim of eEKAB's development was to optimize the ambulance service and the specific objective was the use of new and evolving technologies such as the operating system Android. The basic idea is that ambulances and doctors have Android phones connected to a central server which is connected to a central database. When an incident reaches the EKAV's call center the details of the incident are recorded and the incident is promoted to the nearest ambulances and doctors, through the application running on the server, in order to rush to the incident immediately. The mobile application, in turn, is responsible for updating the central database via the server, with the current location of ambulances and doctors. Along with the development of the application, existing techniques across the world in the field of emergency management are examined. Technologies used in the present information system are also analyzed. Finally, the system is evaluated in order to become apparent whether such a system can improve the existing services.

KEY WORDS: EKAV, Information Systems, Android, Client-Server, GIS, mobile network, medical information system, evaluation

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ.....	5
ABSTRACT.....	6
ΠΕΡΙΕΧΟΜΕΝΑ.....	7
ΛΙΣΤΑ ΕΙΚΟΝΩΝ.....	10
ΛΙΣΤΑ ΠΙΝΑΚΩΝ.....	11
1. ΕΙΣΑΓΩΓΗ.....	12
1.1. Γενικά.....	12
1.2. Σκοπός της μεταπτυχιακής διατριβής.....	12
1.3. Δομή μεταπτυχιακής διατριβής.....	13
2. ΥΠΗΡΕΣΙΕΣ ΕΠΕΙΓΟΥΣΑΣ ΙΑΤΡΙΚΗΣ ΦΡΟΝΤΙΔΑΣ.....	14
2.1. Εισαγωγή.....	14
2.2. Διαχείριση επειγόντων περιστατικών.....	15
2.2.1. Περιγραφή Διαδικασιών.....	15
2.2.2. Καθήκοντα μονάδων προνοσοκομειακής ιατρικής φροντίδας.....	18
2.2.3. Προνοσοκομειακή ιατρική φροντίδα στην Ελλάδα.....	18
2.3. Μοντέλα προνοσοκομειακής ιατρικής φροντίδας.....	19
2.4. Επίπεδα προνοσοκομειακής ιατρικής φροντίδας.....	20
2.5. Υπάρχουσες Τεχνολογίες στα Συστήματα Έκτακτης Ιατρικής Φροντίδας.....	21
2.5.1. Ατομικό Ηλεκτρονικό Αρχείο Υγείας (Electronic Personal Health Record).....	22
2.5.2. E-911.....	22
2.5.3. Αυτόματος Εντοπισμός Ατυχημάτων (Automatic Crash Notification).....	23
2.5.4. Ανάθεση με τη βοήθεια υπολογιστή (Computer Aided Dispatch).....	23
2.5.5. Σύστημα Διαχείρισης Κυκλοφορίας (Traffic Management System).....	24
2.5.6. Σύστημα Γεωγραφικών Πληροφοριών/Παγκόσμιο Σύστημα Εντοπισμού (Geographic Information System/ Global Positioning System).....	24
2.5.7. Τερματικό Κινητών Δεδομένων (Mobile Data Terminal).....	24

3. ΥΠΟΒΑΘΡΟ ΤΕΧΝΟΛΟΓΙΩΝ	26
3.1. Android	26
3.1.1. Τι είναι το Android	26
3.1.2. Πλεονεκτήματα του Android	26
3.1.3. Αρχιτεκτονική του Android	27
3.1.4. Αρχιτεκτονική των εφαρμογών	29
3.2. Μοντέλο πελάτη/εξυπηρετητή σε κινητό περιβάλλον	31
3.2.1. Λεπτή αρχιτεκτονική (thin architecture)	32
3.2.2. Ολοκληρωμένη αρχιτεκτονική (full architecture)	32
3.2.3. Εναλλακτικές αρχιτεκτονικές	33
4. ΕΦΑΡΜΟΓΗ eΕΚΑΒ	35
4.1. Ανάλυση απαιτήσεων	35
4.2. Σχεδιασμός εφαρμογής	36
4.3. Υλοποίηση εφαρμογής	40
4.3.1. Client	40
4.3.2. Server	40
4.4. Περιγραφή της λειτουργίας του προγράμματος	41
4.4.1. Server	41
4.4.2. Client	52
5. ΑΞΙΟΛΟΓΗΣΗ ΕΦΑΡΜΟΓΗΣ eΕΚΑΒ	57
5.1. Περιγραφή μοντέλου αξιολόγησης	57
5.2. Κριτήρια αξιολόγησης	58
5.2.1. Απόδοση	58
5.2.2. Χρησιμότητα	61
5.2.3. Αξιοπιστία	62
5.2.4. Αποτελεσματικότητα	63
5.3. Αποτελέσματα αξιολόγησης	63
5.4. Συμπεράσματα	63
6. ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΤΑΣΕΙΣ	65
7. ΒΙΒΛΙΟΓΡΑΦΙΑ	66
ΠΑΡΑΡΤΗΜΑ.....	69
Α. Επεξήγηση κώδικα	69

A.1. Κώδικας προγράμματος κεντρικής εφαρμογής	69
A.2. Κώδικας προγράμματος εφαρμογής κινητού.....	79
B. Σχεδίαση Βάσης Δεδομένων	85
Γ. Οδηγίες εγκατάστασης.....	86
Γ.1. Κεντρική εφαρμογή.....	86
Γ.2. Mobile εφαρμογή	87
Δ. Διαδικασία εκκίνησης της εφαρμογής e-EKAB.....	90
Δ.1. Web Server	90
Δ.2. Κινητό Android.....	91

ΛΙΣΤΑ ΕΙΚΟΝΩΝ

1. Αστέρι της ζωής, σύμβολο των υπηρεσιών έκτακτης ιατρικής βοήθειας
2. Διαδικασίες κατά τη διαχείριση έκτακτων περιστατικών
3. Παράδειγμα κατανομής ασθενοφόρων και κινητών ιατρικών μονάδων του ΕΚΑΒ σε ένα κομμάτι της Αττικής
4. Λογότυπο Android
5. Αρχιτεκτονική του Android
6. Κύκλος ζωής μίας δραστηριότητας
7. Αρχιτεκτονική της εφαρμογής eEKAB
8. Αρχική σελίδα της εφαρμογής
9. Εμφάνιση λαθών στην περίπτωση που κάποιο πεδίο είναι κενό
10. Εμφάνιση λαθών στην περίπτωση λανθασμένης εισαγωγής
11. Η λίστα του πεδίου «Ειδικότητα Ιατρού»
12. Η αυτόματη λίστα (auto complete) του πεδίου «Διεύθυνση»
13. Μετά την υποβολή του περιστατικού εμφανίζεται η περιοχή «Αναζήτηση Ιατρών και Οχημάτων»
14. Αν πατήσουμε σε κάποιο ιατρό η ασθενοφόρο εμφανίζεται η ακριβής διεύθυνση που βρίσκεται εκείνη τη στιγμή
15. Ο χάρτης μετά την απόκρυψη των γιατρών
16. Ο χάρτης μετά την εμφάνιση των γιατρών
17. Συντομότερη διαδρομή από έναν γιατρό έως το περιστατικό
18. Εμφάνιση εκτιμώμενου χρόνου της διαδρομής
19. Αρχική σελίδα καρτέλας «Οχήματα»
20. Επιλογή μίας εγγραφής στον πίνακα της καρτέλας «Οχήματα»
21. Η φόρμα της εισαγωγής ασθενοφόρου – οδηγού
22. Μηνύματα λάθους στην περίπτωση κενού πεδίου
23. Μηνύματα λάθους στην περίπτωση λανθασμένων τιμών στα πεδία
24. Μήνυμα επιβεβαίωσης ότι η εισαγωγή ήταν επιτυχής
25. Η φόρμα της τροποποίησης ασθενοφόρου – οδηγού
26. Παράθυρο διαλόγου για επιβεβαίωση διαγραφής
27. Φόρμα εισαγωγής ιατρού
28. Κύρια οθόνη e-EKAB στο κινητό
29. Μήνυμα λάθους όταν ο αριθμός δεν είναι σωστός
30. Μήνυμα λάθους όταν ο κωδικός δεν είναι σωστός
31. Μήνυμα επιβεβαίωσης σύνδεσης
32. Ειδοποιητικό μήνυμα για τις λειτουργίες της εφαρμογής
33. Εικονίδιο στην πάνω αριστερή μεριά του κινητού που μας πληροφορεί ότι η εφαρμογή τρέχει στο παρασκήνιο
34. Έξοδος από την εφαρμογή
35. Ειδοποίηση άφιξης καινούριου περιστατικού
36. Διαδρομή από το σημείο που βρίσκεται το κινητό εκείνη τη στιγμή (γαλάζια κουκίδα) μέχρι το περιστατικό
37. Επιλογή του μενού για έξοδο
38. Παράθυρο διαλόγου επιβεβαίωσης εξόδου από την εφαρμογή

39. Διεργασίες προνοσοκομειακής ιατρικής φροντίδας πριν την ενσωμάτωση του ΠΣ eEKAB
40. Διεργασίες προνοσοκομειακής ιατρικής φροντίδας μετά την ενσωμάτωση του ΠΣ eEKAB
41. Διαδικασία καταγραφής στοιχείων περιστατικού πριν την ενσωμάτωση του ΠΣ eEKAB
42. Διαδικασία ανάθεσης περιστατικού σε κοντινότερο ασθενοφόρο/γιατρό πριν το eEKAB
43. Σχηματική αναπαράσταση της βάσης δεδομένων της εφαρμογής
44. Παράθυρο διαλόγου της εγκατάστασης του XAMPP για επιλογή συντομεύσεων και εκτέλεσης του Apache και της MySQL σαν υπηρεσιών
45. Scan από το Wizard της εγκατάστασης του Android SDK για εύρεση του JDK
46. Τέλος εγκατάστασης του Android SDK Starter Package
47. Επιλογή εγκατάστασης api από τον Android SDK Manager
48. Εγκατάσταση του ADT plugin
49. Εγκατάσταση του ADT plugin (2)
50. Πίνακας ελέγχου του XAMPP
51. Android Virtual Device Manager
52. Δημιουργία νέας εικονικής συσκευής Android

ΛΙΣΤΑ ΠΙΝΑΚΩΝ

1. Γενικά χαρακτηριστικά προνοσοκομειακής ιατρικής φροντίδας ορισμένων χωρών
2. Υπάρχουσες τεχνολογίες ανά διαδικασία EMS
3. Διάγραμμα ακολουθίας eEKAB

1. ΕΙΣΑΓΩΓΗ

1.1 Γενικά

Όσο αναπτύσσεται ο κλάδος της πληροφορικής τόσο και πιο πολύ προσπαθεί να διεισδύσει στα μονοπάτια άλλων επιστημών όπως η ιατρική, η αστρονομία, η βιολογία και πάρα πολλές άλλες. Στην εποχή που διανύουμε, την εποχή της πληροφορίας, δε νοείται επιστήμη που δε χρησιμοποιεί έστω και σε μικρό ποσοστό τις δυνατότητες που μπορεί να της παρέχει ένα υπολογιστικό σύστημα. Έτσι και η επιστήμη της ιατρικής έχει εμπιστευτεί πλέον πολλές από τις λεπτές διαδικασίες και υπηρεσίες της στην επιστήμη της πληροφορικής και όχι άδικα, αφού πλέον οι ασθενείς απολαμβάνουν αποτελεσματικότερες και ποιοτικότερες υπηρεσίες.

Μεγάλο ρόλο στη βελτιστοποίηση των υπηρεσιών υγείας έχει παίξει και η εξελισσόμενη ανάπτυξη των τηλεπικοινωνιών. Είναι γεγονός ότι στη σημερινή εποχή μέσω ενός κινητού τηλεφώνου μπορούμε να έχουμε πρόσβαση στο Internet, να επικοινωνούμε με τον οποιοδήποτε, να γνωρίζουμε πού βρισκόμαστε ανά πάσα στιγμή και να μπορούμε να επιλέξουμε ποια διαδρομή θα ακολουθήσουμε για να φτάσουμε στον επιθυμητό προορισμό. Αυτές είναι μόνο λίγες δυνατότητες που μας προσφέρει ένα κινητό τηλέφωνο.

Οπότε είναι φανερό ότι ο συνδυασμός του κλάδου της πληροφορικής και αυτού των τηλεπικοινωνιών θα μπορούσε να μας δώσει καλύτερες υπηρεσίες στον τομέα της υγείας. Ένα πληροφοριακό σύστημα υγείας το οποίο θα μπορούσε να εκμεταλλευτεί τις δυνατότητες που προσφέρει ο κλάδος των τηλεπικοινωνιών θα ήταν ασυναγώνιστο. Και αυτό γιατί οι υπηρεσίες θα ήταν πιο γρήγορες κάτι το οποίο είναι και το πιο σημαντικό όταν έχουμε να κάνουμε με υπηρεσίες υγείας που και το ένα δευτερόλεπτο είναι σημαντικό.

Το πληροφοριακό σύστημα που υλοποιήθηκε στην παρούσα μεταπτυχιακή διατριβή αποτελεί ένα πληροφοριακό σύστημα άμεσης επέμβασης το οποίο χρησιμοποιεί στο έπακρο τις δυνατότητες των τηλεπικοινωνιών για να προσφέρει γρήγορες υπηρεσίες σε έκτακτα περιστατικά τα οποία χωρίς τη συμβολή ενός πληροφοριακού συστήματος θα είχαν πιο αργή αντιμετώπιση. Ένα πληροφοριακό σύστημα το οποίο διαθέτει μία βάση δεδομένων συνδεδεμένη με κινητά – δέκτες (ασθενοφόρα, γιατροί) που ενημερώνει ανάλογα με το εκάστοτε περιστατικό τα διαθέσιμα ασθενοφόρα και γιατρούς που βρίσκονται πιο κοντά στο περιστατικό με τη χρήση google maps είναι ένα πολύ αποτελεσματικό εργαλείο για τον κλάδο της υγείας και είναι αυτό το οποίο θα αναλύσουμε στην παρούσα μεταπτυχιακή διατριβή.

1.2 Σκοπός της μεταπτυχιακής διατριβής

Αντικείμενο της παρούσας μεταπτυχιακής διατριβής αποτελεί η ανάπτυξη της eEKAB, μίας πελάτη - εξυπηρετητή εφαρμογής για τη διαχείριση έκτακτων ιατρικών περιστατικών. Η εφαρμογή eEKAB αναπτύχθηκε με στόχο να καλύψει ένα μεγάλο φάσμα τεχνολογιών όπως αυτό του GIS, της ανάπτυξης εφαρμογών σε Android και το μοντέλο πελάτη - εξυπηρετητή σε κινητά περιβάλλοντα. Ουσιαστικά είναι μία προσπάθεια βελτίωσης των υπηρεσιών που προσφέρει το Εθνικό Κέντρο Άμεσης Βοήθειας με τη βοήθεια της σύγχρονης τεχνολογίας.

Η βασική ιδέα της εφαρμογής είναι ασθενοφόρα και γιατροί να διαθέτουν όλοι από ένα κινητό με λειτουργικό σύστημα Android. Τα κινητά αυτά θα συνδέονται με έναν κεντρικό server στο κέντρο ελέγχου. Όταν ένα περιστατικό θα φτάνει στο κέντρο ελέγχου θα προωθείται αμέσως στα κοντινότερα ασθενοφόρα/γιατρούς. Με αυτό τον τρόπο θα έχουμε πολύ γρήγορη εξυπηρέτηση των περιστατικών.

Στόχος επίσης είναι και η αξιολόγηση του πληροφοριακού συστήματος με βάση το υπάρχον σύστημα στην Ελλάδα και η εξαγωγή συμπερασμάτων για το κατά πόσο ένα σύστημα θα ήταν αποτελεσματικό σε ένα κέντρο άμεσης βοήθειας.

1.3 Δομή μεταπτυχιακής διατριβής

Στην παρούσα μεταπτυχιακή διατριβή, αρχικά, κάνουμε μία εισαγωγή στις υπηρεσίες επείγουσας ιατρικής βοήθειας, περιγράφοντας τον τρόπο με τον οποίο λειτουργούν ανά τον κόσμο αλλά και στην Ελλάδα. Παρουσιάζονται, επίσης, τα μοντέλα και τα επίπεδα προνοσοκομειακής ιατρικής φροντίδας και γίνεται μία σύγκριση του επιπέδου της Ελλάδας με άλλες χώρες στον τομέα αυτό. Στη συνέχεια γίνεται μία σύντομη περιγραφή των τεχνολογιών που χρησιμοποιούνται στην διαχείριση επειγόντων περιστατικών και αναλύονται οι κυριότερες από αυτές.

Σειρά στο κείμενο έχει το υπόβαθρο τεχνολογιών της εφαρμογής που υλοποιήθηκε στην παρούσα μεταπτυχιακή διατριβή, το eEKAB. Συγκεκριμένα γίνεται μία εισαγωγή στο λειτουργικό Android και στη συνέχεια αναλύεται η αρχιτεκτονική του λειτουργικού και η αρχιτεκτονική των εφαρμογών. Τέλος, παρουσιάζεται το μοντέλο πελάτη – εξυπηρετητή και περιγράφονται οι προκλήσεις που υπεισέρχονται στο μοντέλο αυτό στο κινητό περιβάλλον καθώς και οι προτεινόμενες λύσεις της βιβλιογραφίας.

Έχοντας παρουσιάσει αναλυτικά το θεωρητικό και τεχνολογικό υπόβαθρο του τομέα της διαχείρισης έκτακτων περιστατικών στη συνέχεια προβάλλουμε και τη δική μας εφαρμογή το eEKAB. Έτσι λοιπόν, ξεκινάμε με την ανάλυση απαιτήσεων, την περιγραφή του σχεδιασμού εφαρμογής, τον τρόπο υλοποίησής της και, τέλος, την περιγραφή λειτουργίας με αρκετά screenshots για την καλύτερη κατανόηση της εφαρμογής.

Μετά την παρουσίαση της εφαρμογής ακολουθεί η αξιολόγηση του εγχειρήματός μας. Στο κεφάλαιο της αξιολόγησης επιλέγουμε το μοντέλο αξιολόγησης που προτείνουν οι Olegas Vasilecas, Algis Saulis, Saulius Dereškevičius στο άρθρο τους «EVALUATION OF INFORMATION SYSTEMS PROCUREMENT: GOAL AND TASK-DRIVEN APPROACHES» με το οποίο εξάγουμε την ποιοτική απόδοση του συστήματός μας με βάση τις αρχικές απαιτήσεις που είχαμε θέσει.

Στο τέλος της μεταπτυχιακής διατριβής παρατίθενται τα συμπεράσματα στα οποία καταλήγουμε κατόπιν της αξιολόγησης του πληροφοριακού μας συστήματος. Τα αποτελέσματα αυτά σχολιάζονται και προτείνονται λύσεις και βελτιώσεις που θα μπορούσαν να γίνουν ώστε να γίνει το σύστημα πιο αποδοτικό. Ακολουθεί η παράθεση της βιβλιογραφίας που χρησιμοποιήθηκε.

Στο τέλος της μεταπτυχιακής διατριβής βρίσκεται ένα παράρτημα στο οποίο αρχικά παρουσιάζονται τα κυριότερα σημεία του κώδικα με εκτενή επεξηγηματικά σχόλια και περιγράφεται η σχεδίαση της βάσης δεδομένων. Το παράρτημα καθώς και η μεταπτυχιακή διατριβή κλείνει με τις οδηγίες εγκατάστασης της εφαρμογής και τη διαδικασία εκκίνησης της εφαρμογής.

2. ΥΠΗΡΕΣΙΕΣ ΕΠΕΙΓΟΥΣΑΣ ΙΑΤΡΙΚΗΣ ΦΡΟΝΤΙΔΑΣ

2.1 Εισαγωγή

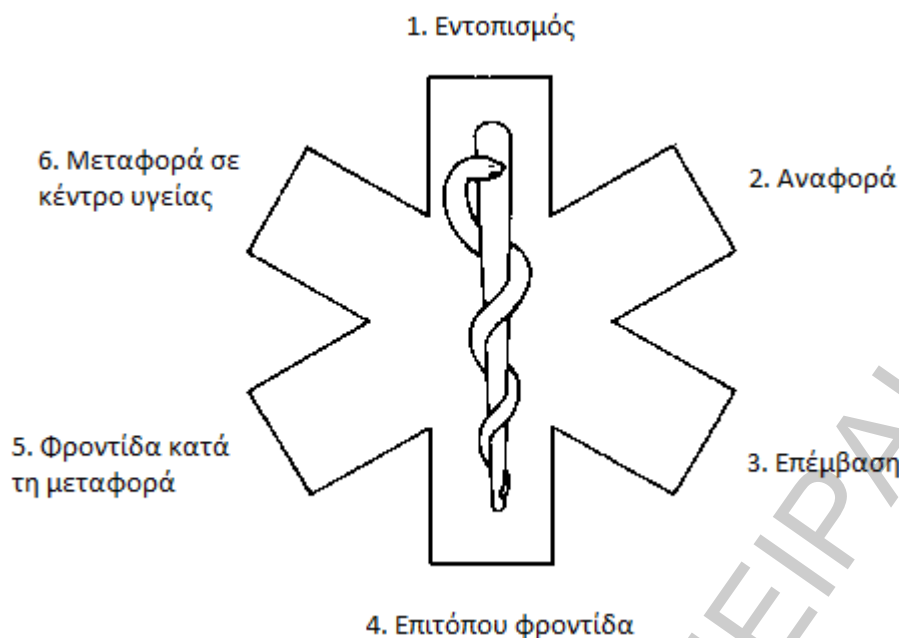
Μία υπηρεσία επείγουσας ιατρικής φροντίδας (EMS – Emergency Medical Service) μπορεί να οριστεί ως «Ένα ολοκληρωμένο σύστημα το οποίο παρέχει τις ρυθμίσεις του προσωπικού, των εγκαταστάσεων και του εξοπλισμού για την αποτελεσματική, συντονισμένη και έγκαιρη προσφορά υπηρεσιών υγείας και ασφάλειας στα θύματα μίας ξαφνικής ασθένειας ή τραυματισμού». Ο στόχος αυτού του συστήματος εστιάζεται στην παροχή έγκαιρης περίθαλψης σε θύματα ξαφνικών και απειλητικών για τη ζωή τραυματισμών ή καταστάσεων έκτακτης ανάγκης προκειμένου να αποφευχθεί η θνησιμότητα ή η μακροχρόνια νοσηρότητα. Η λειτουργία ενός EMS μπορεί να απλοποιηθεί σε τέσσερα κύρια συστατικά: πρόσβαση σε πρώτες βοήθειες, περίθαλψη στην κοινωνία, περίθαλψη καθ' οδόν και περίθαλψη κατά την άφιξη στο κέντρο υγείας για να λάβει ο ασθενής φροντίδα [1].

Το EMS αποτελεί ένα υποσύστημα του συστήματος υγείας κάθε ανεπτυγμένης χώρας. Τα σημερινά παγκόσμια EMS έχουν προοδεύσει τόσο πολύ ώστε να συμβάλλουν ευρέως στη συνολική λειτουργία των συστημάτων υγειονομικής περίθαλψης. Ο Παγκόσμιος Οργανισμός Υγείας αντιμετωπίζει τα EMS σαν ένα αναπόσπαστο κομμάτι οποιουδήποτε αποτελεσματικού και λειτουργικού υγειονομικού συστήματος. Το EMS είναι το πρώτο σημείο επαφής με τις υπηρεσίες υγείας, για την πλειοψηφία των ανθρώπων, σε περίπτωση έκτακτης ανάγκης και σοβαρών ατυχημάτων και λειτουργεί σαν «πύλη» για την πρόσβαση σε δευτερογενείς και τριτογενείς υπηρεσίες [1].

Διεθνώς, τα EMS παρουσιάζουν μεγάλη ανομοιογένεια. Διαφέρουν σημαντικά ανάλογα με τη γεωγραφική θέση, την τοπογραφία, τους διαθέσιμους πόρους καθώς και τη νομοθεσία κάθε χώρας. Αυτό οφείλεται στο γεγονός ότι η προσπάθεια που γίνεται για εκσυγχρονισμό των συστημάτων διαχείρισης των επειγόντων περιστατικών γίνεται με βάση τα διαφορετικά υγειονομικά, πολιτισμικά, πολιτικά και οικονομικά χαρακτηριστικά κάθε χώρας [2].

Οι κυριότεροι στόχοι ενός EMS προκύπτουν ουσιαστικά από το σύμβολο «αστέρι της ζωής», που φαίνεται και στην εικόνα 1, το οποίο διεθνώς αποτελεί το σήμα των Συστημάτων Έκτακτης Ιατρικής Φροντίδας. Οι στόχοι πιο αναλυτικά είναι οι εξής [3, 23]:

1. *Έγκαιρη Αντίληψη του Περιστατικού*
Οι πρώτοι άνθρωποι που θα φτάσουν στο περιστατικό, οι οποίοι μπορεί να είναι περαστικοί ή να εμπλέκονται στο περιστατικό, παρατηρούν το περιστατικό, κατανοούν την κρισιμότητα της κατάστασης, αναγνωρίζουν πιθανούς κινδύνους για τους ίδιους και τους άλλους και λαμβάνουν τα απαραίτητα μέτρα για να εξασφαλίσουν την ασφάλεια της περιοχής. (ραδιενέργεια, χημικά, ηλεκτρισμός, κτλ)
2. *Έγκαιρη αναφορά του περιστατικού*
Οι πρώτοι άνθρωποι που θα φτάσουν στο περιστατικό πραγματοποιούν μία κλήση στο Εθνικό Κέντρο Άμεσης Βοήθειας (ΕΚΑΒ) και δίνουν όλα τα στοιχεία που θα τους ζητηθούν.
3. *Έγκαιρη επέμβαση στο περιστατικό*
Οι πρώτοι διασώστες της υπηρεσίας επείγουσας ιατρικής φροντίδας καταφθάνουν στο περιστατικό όσο το δυνατόν πιο γρήγορα.
4. *Επιτόπου φροντίδα*
Η υπηρεσία επείγουσας ιατρικής φροντίδας παρέχει τις πρώτες βοήθειες στον τόπο του περιστατικού
5. *Ιατρική φροντίδα κατά τη μεταφορά*
Η υπηρεσία φορτώνει τον ασθενή στο κατάλληλο μεταφορικό μέσο (ασθενοφόρο, ελικόπτερο κτλ) και συνεχίζει να του παρέχει ιατρική φροντίδα σε όλη τη διάρκεια του ταξιδιού
6. *Μεταφορά σε κέντρο υγείας*
Ο ασθενής μεταφέρεται σε κάποιο κέντρο υγείας (νοσοκομείο, ιδιωτικό κέντρο κτλ) όπου και αφήνεται πλέον στα έμπειρα χέρια των γιατρών.



Εικόνα 1: Αστέρι της ζωής, σύμβολο των υπηρεσιών έκτακτης ιατρικής βοήθειας [3]

Από τα παραπάνω καθίσταται σαφές ότι η αποτελεσματικότητα ενός EMS εξαρτάται από διάφορα κριτήρια όπως είναι η ποσότητα και η ποιότητα των ασθενοφόρων, ο εξοπλισμός των ασθενοφόρων, ο αριθμός και το επίπεδο κατάρτισης του νοσηλευτικού και ιατρικού προσωπικού καθώς και το πληροφοριακό σύστημα που θα συνδέει και θα συντονίζει τις μονάδες. Σε διεθνές επίπεδο έχουν γίνει πολλές προσπάθειες για την ανάπτυξη και τη χρήση πληροφοριακών συστημάτων επείγουσας ιατρικής φροντίδας με μικρότερη ή μεγαλύτερη επιτυχία. Ένα από τα σημαντικότερα μειονεκτήματα αυτών των πληροφοριακών συστημάτων είναι η ανεπαρκής σύνδεση των μονάδων που απαρτίζουν ένα EMS [2, 24, 25, 26].

Στην παρούσα μεταπτυχιακή διατριβή γίνεται μία προσπάθεια για την υλοποίηση ενός πληροφοριακού συστήματος που θα ενώνει αυτές τις μονάδες με αποτελεσματικό τρόπο.

Στη συνέχεια θα περιγραφούν οι διαδικασίες που εκτελούνται κατά τη διαχείριση επειγόντων περιστατικών και στη συνέχεια θα περιγραφεί η δομή και η λειτουργία του ελληνικού συστήματος παροχής έκτακτης προνοσοκομειακής ιατρικής φροντίδας (ΕΚΑΒ).

2.2 Διαχείριση επειγόντων περιστατικών

2.2.1 Περιγραφή Διαδικασιών

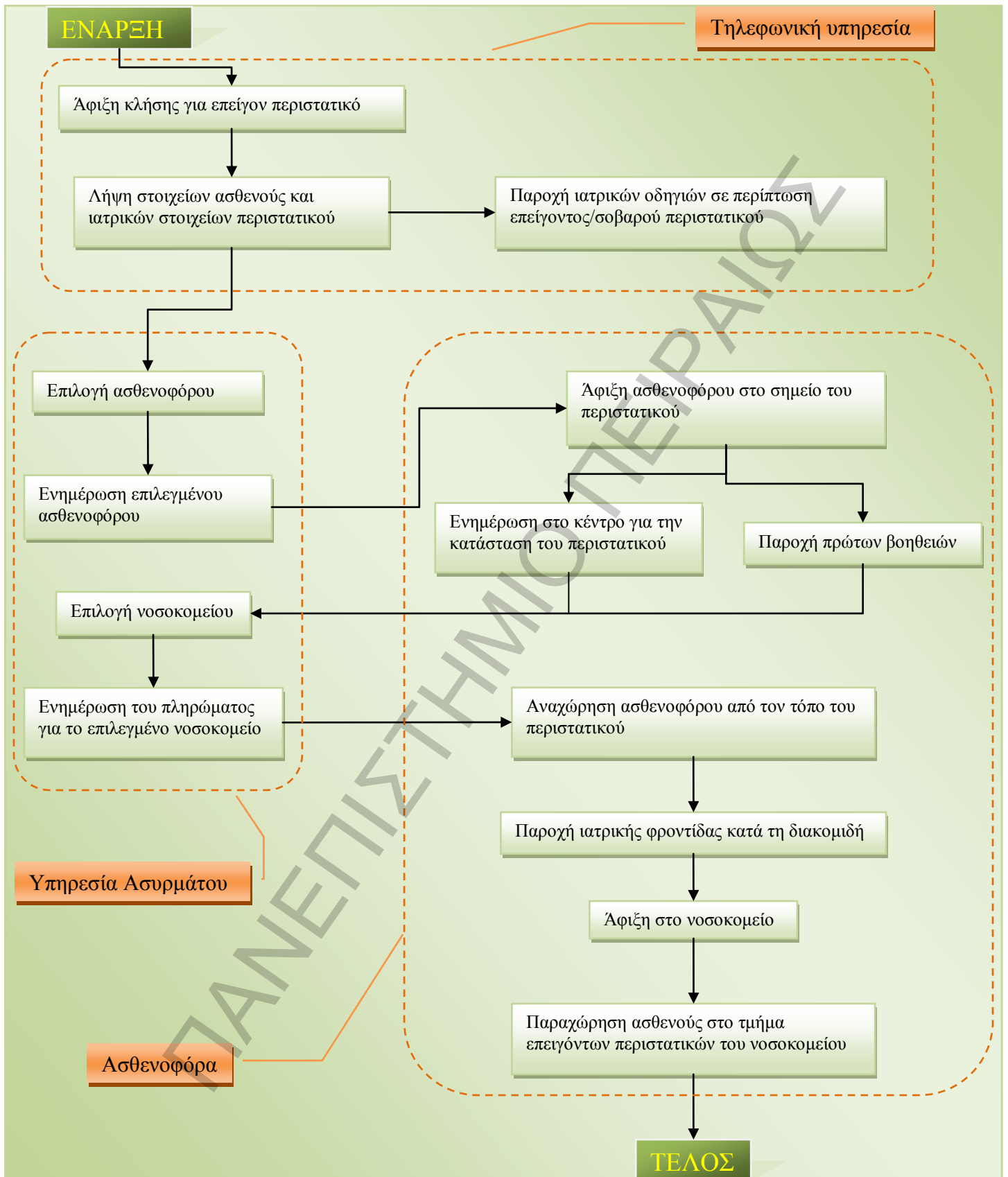
Η διαχείριση επειγόντων περιστατικών αναφέρεται στις διαδικασίες εκείνες οι οποίες εκτελούνται από τη στιγμή που ο ασθενής ή κάποιος ο οποίος βρίσκεται στο περιστατικό καλεί το Εθνικό Κέντρο Άμεσης Βοήθειας (ΕΚΑΒ) μέχρι τη στιγμή που ο ασθενής παίρνει εξιτήριο από το νοσοκομείο στο οποίο νοσηλεύεται. Στην ενότητα αυτή περιγράφουμε μόνο τις διαδικασίες της προνοσοκομειακής ιατρικής φροντίδας και όχι της ενδονοσοκομειακής αφού δεν αποτελούν αντικείμενο της παρούσας μεταπτυχιακής διατριβής. Η προνοσοκομειακή έκτακτη φροντίδα παρουσιάζει μικρές διαφορές από χώρα σε χώρα.[2] Εδώ γίνεται μία γενική περιγραφή της προνοσοκομειακής ιατρικής φροντίδας [1, 2, 4, 5, 27].

Η διαχείριση έκτακτων περιστατικών, όπως είπαμε, ξεκινά με την κλήση στο ΕΚΑΒ. Στο κέντρο η τηλεφωνική υπηρεσία καταγράφει τα στοιχεία του ασθενούς καθώς και την ακριβή διεύθυνση στη οποία αυτός βρίσκεται. Εάν είναι αναγκαίο καταγράφει και ιατρικές πληροφορίες. Στη συνέχεια εάν το περιστατικό κριθεί αρκετά επείγον η κλήση προωθείται στην ιατρική υπηρεσία του κέντρου για να δοθούν οι απαραίτητες ιατρικές οδηγίες προς αυτόν που ανέφερε το περιστατικό. Ταυτόχρονα, τα στοιχεία του ασθενούς δίνονται στην υπηρεσία ασυρμάτου η

οποία με τη σειρά της εντοπίζει το καταλληλότερο ασθενοφόρο και το ενημερώνει για το περιστατικό και τη τοποθεσία του έτσι ώστε να ξεκινήσει για την παραλαβή του ασθενούς [1, 2, 4, 5, 27].

Μόλις το ασθενοφόρο φτάσει στο περιστατικό το πλήρωμά του αφού εκτιμήσει πρώτα την κατάσταση ενημερώνει το κέντρο. Στη συνέχεια δίνονται οι πρώτες βοήθειες στον ασθενή από το πλήρωμα του ασθενοφόρου με ταυτόχρονη επικοινωνία με την ιατρική υπηρεσία του ΕΚΑΒ σε περίπτωση που δεν υπάρχει κάποιος ιατρός μέσα στο ασθενοφόρο. Στη συνέχεια, η υπηρεσία ασυρμάτου επιλέγει το καταλληλότερο νοσοκομείο για τη διακομιδή του ασθενούς και ενημερώνει το πλήρωμα του ασθενοφόρου για το νοσοκομείο που επιλέχθηκε και για την ταχύτερη διαδρομή την οποία θα πρέπει να ακολουθήσει [1, 2, 4, 5, 27].

Καθ' όλη τη διάρκεια του περιστατικού δίνεται ιατρική φροντίδα στον ασθενή υπό τις οδηγίες της ιατρικής υπηρεσίας εάν χρειάζεται. Τέλος το ασθενοφόρο μόλις φτάσει στο νοσοκομείο παραδίδει τον ασθενή στους γιατρούς των επειγόντων περιστατικών για να αναλάβουν αυτοί πλέον την περαιτέρω ιατρική φροντίδα και ενημερώνει το κέντρο για την περάτωση της αποστολής του [1, 2, 4, 5, 27]. Όλες οι διαδικασίες φαίνονται σχηματικά στην εικόνα 2.



Εικόνα 2: Διαδικασίες κατά τη διαχείριση έκτακτων περιστατικών

2.2.2 Καθήκοντα μονάδων προνοσοκομειακής ιατρικής φροντίδας

Όπως φαίνεται και από την εικόνα 2 οι μονάδες που εμπλέκονται στην προνοσοκομειακή ιατρική φροντίδα είναι 4, η τηλεφωνική υπηρεσία, η υπηρεσία ασυρμάτου η ιατρική υπηρεσία και το πλήρωμα των ασθενοφόρων. Στη συνέχεια περιγράφουμε πιο αναλυτικά τις δραστηριότητες / καθήκοντα κάθε υπηρεσίας / μονάδας [1, 2, 4, 5, 27].

Τηλεφωνική υπηρεσία

- λαμβάνει τις κλήσεις των πολιτών που χρειάζονται βοήθεια σχετικά με ένα έκτακτο περιστατικό
- καταγράφει τα στοιχεία του ασθενούς καθώς και τα ιατρικά στοιχεία του περιστατικού
- εκτιμάει την κρισιμότητα του περιστατικού και ανάλογα προωθεί την κλήση στην ιατρική υπηρεσία
- προωθεί τα στοιχεία του ασθενούς και τα ιατρικά στοιχεία στην υπηρεσία ασυρμάτου.

Ιατρική υπηρεσία

- παρέχει ιατρικές οδηγίες μέσω τηλεφώνου απευθείας στον αναφέρων του περιστατικού
- παρέχει ιατρικές οδηγίες στο πλήρωμα του ασθενοφόρου, ούτως ώστε η ιατρική βοήθεια στον ασθενή να είναι όσο το δυνατόν πιο αποτελεσματική.

Υπηρεσία ασυρμάτου

- επιλέγει το καταλληλότερο ασθενοφόρο που θα αναλάβει την αποστολή της προνοσοκομειακής φροντίδας
- ενημερώνει το πλήρωμα του επιλεγμένου ασθενοφόρου για τα ιατρικά στοιχεία του ασθενούς καθώς και για τον τόπο του περιστατικού
- επιλέγει το κατάλληλο νοσοκομείο στο οποίο θα διακομιστεί ο ασθενής
- ενημερώνει το πλήρωμα του ασθενοφόρου για το επιλεγμένο νοσοκομείο
- ενημερώνει το επιλεγμένο νοσοκομείο για την αναμενόμενη άφιξη του ασθενοφόρου και για τα ιατρικά στοιχεία του ασθενούς που προέκυψαν από τη διαδικασία.

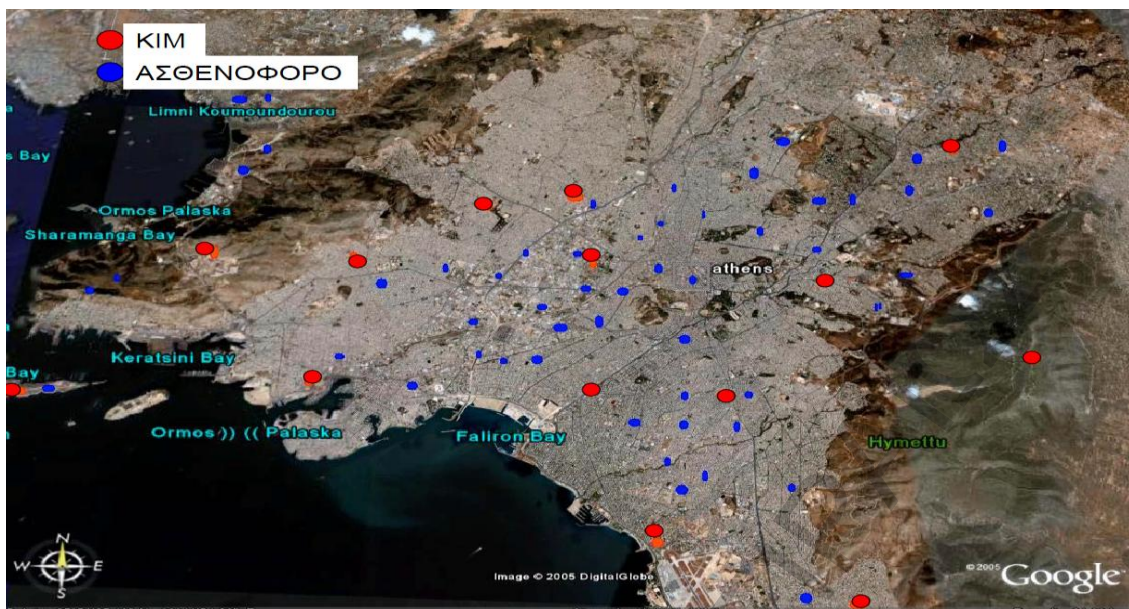
Πληρώματα ασθενοφόρων

- ενημερώνουν την υπηρεσία ασυρμάτου για την πορεία και την εξέλιξη του περιστατικού
- ενημερώνουν την ιατρική υπηρεσία για τις πρώτες βοήθειες που προσφέρουν στο περιστατικό.

2.2.3 Προνοσοκομειακή ιατρική φροντίδα στην Ελλάδα

Στην Ελλάδα τις υπηρεσίες έκτακτης ιατρικής φροντίδας προσφέρει το Ε.Κ.Α.Β. (Εθνικό Κέντρο Άμεσης Βοήθειας). Μάλιστα είναι ο μοναδικός επίσημος φορέας της πολιτείας παροχής υπηρεσιών Επείγουσας Προνοσοκομειακής Ιατρικής και Νοσηλευτικής Φροντίδας. Σκοπός του είναι η «άμεση ιατρική και νοσηλευτική φροντίδα σε όλους τους πολίτες οπουδήποτε και οποτεδήποτε χρειαστεί, καθώς και η ασφαλής και ταχεία μεταφορά τους σε κατάλληλες μονάδες παροχής υπηρεσιών υγείας» [5,6].

Διαθέτει 735 ασθενοφόρα (350 ασθενοφόρα έχουν παραχωρηθεί από το ΕΚΑΒ και χρησιμοποιούνται από νοσοκομεία και κέντρα υγείας), 105 ειδικές κινητές μονάδες επείγουσας προνοσοκομειακής ιατρικής, 25 μοτοσικλέτες, 4 μικρά οχήματα ταχείας πρόσβασης, 21 λοιπά οχήματα, 3 ελικόπτερα, 2 αυτοκινούμενα συντονιστικά κέντρα και 2 οχήματα αντιμετώπισης καταστροφών. Ένα παράδειγμα τοπογραφικής κατανομής του ΕΚΑΒ φαίνεται στην εικόνα 3.



Εικόνα 3: Παράδειγμα κατανομής ασθενοφόρων και κινητών ιατρικών μονάδων του ΕΚΑΒ σε ένα κομμάτι της Αττικής [5]

Επίσης αποτελείται από περισσότερους από 2900 διασώστες, 100 γιατρούς (όλοι ειδικευμένοι και εκπαιδευμένοι στην επείγουσα προνοσοκομειακή ιατρική) οι οποίοι ασκούν την επείγουσα ιατρική σε προνοσοκομειακή επίπεδο.

Το Επιχειρησιακό Κέντρο Επικοινωνίας και Συντονισμού λειτουργεί 24 ώρες το 24ωρο και είναι υπεύθυνο για τα παρακάτω [5,6]:

- δέχεται κλήσεις για επείγοντα περιστατικά, καταγράφει τα στοιχεία τους και τα κατηγοριοποιεί με βάση το βαθμό του επείγοντος
- επιλέγει το κοντινότερο ασθενοφόρο ή κινητή μονάδα παροχής επείγουσας προνοσοκομειακής ιατρικής και τους προωθεί το περιστατικό
- παρακολουθεί και καθοδηγεί τις κινητές ιατρικές μονάδες που σπεύδουν προ του τόπου του συμβάντος
- ενεργοποιεί την Ομάδα Διαχείρισης Κρίσης (ΟΔΙΚ) του ΕΚΑΒ η οποία είναι υπεύθυνη σε περιπτώσεις ομαδικών ατυχημάτων ή μαζικών καταστροφών.
- επικοινωνεί με τα τμήματα των κέντρων υγείας που θα υποδεχτούν τα έκτακτα περιστατικά και τα ενημερώνει για τους ασθενείς των περιστατικών
- καταγράφει όλες τις πληροφορίες για τα διακινούμενα περιστατικά.

Επίσης διαθέτει [5,6]:

- τηλεφωνικό κέντρο με έναν εθνικό αριθμό κλήσης (166)
- σύγχρονα ψηφιακά δίκτυα ασύρματης επικοινωνίας για τη συνεχή επαφή με ασθενοφόρα και άλλα μέσα καθώς και με άλλους εμπλεκόμενους φορείς (ΕΛ.ΑΣ, Π.Σ., Γ.Γ.Π.Π.)
- πειραματικό ασύρματο ψηφιακό σύστημα εντοπισμού και καθοδήγησης όλων των οχημάτων του ΕΚΑΒ Αθηνών
- δίκτυο επικοινωνίας με τα τμήματα υποδοχής επειγόντων περιστατικών των κέντρων υγείας
- πληροφοριακά συστήματα για καταγραφή και επεξεργασία πληροφοριών των περιστατικών,
- σύγχρονα μέσα μετάδοσης ιατρικών πληροφοριών.

2.3 Μοντέλα προνοσοκομειακής ιατρικής φροντίδας

Σε διεθνές επίπεδο έχουν κυριαρχήσει δύο μοντέλα προνοσοκομειακής ιατρικής φροντίδας που διαφοροποιούνται στη φιλοσοφία που ακολουθούν για την παραλαβή και μετάβαση του Διαδικτυακό Πληροφοριακό Σύστημα Διαχείρισης Έκτακτων Περιστατικών με Δυνατότητα Πρόσβασης μέσω Κινητών Συσκευών με Λειτουργικό Android

ασθενούς στο κέντρο υγείας. Το πρώτο μοντέλο είναι το φρανκο - γερμανικό και το άλλο είναι το αγγλο – αμερικάνικο.

Το φρανκο - γερμανικό μοντέλο προνοσοκομειακής ιατρικής φροντίδας βασίζεται στη φιλοσοφία “stay and stabilize”. Σύμφωνα με τη φιλοσοφία αυτή οι διασώστες που καταφθάνουν στο περιστατικό δίνουν περισσότερη έμφαση στη προσφορά πρώτων βοηθειών με σκοπό τη σταθεροποίηση της υγείας του ασθενούς πριν τον μεταφέρουν σε ένα κέντρο υγείας για περαιτέρω περίθαλψη. Αυτό το μοντέλο εφαρμόζεται κυρίως από γιατρούς που υπάρχουν στα ασθενοφόρα οι οποίοι διαθέτουν πολύ προηγμένη τεχνολογία. Οι γιατροί έχουν την δικαιοδοσία να πραγματοποιούν πολύπλοκες ιατρικές εκτιμήσεις και να θεραπεύουν τους ασθενείς στο χώρο του συμβάντος. Αυτό έχει σαν αποτέλεσμα οι περισσότεροι ασθενείς να θεραπεύονται επί τόπου παρά να μεταφέρονται σε νοσοκομεία για θεραπεία. Οι λιγιστοί ασθενείς που μεταφέρονται με κάποιο ασθενοφόρο διακομίζονται κατευθείαν στις πτέρυγες των κέντρων υγείας από τους ιατρούς που βρίσκονταν στο περιστατικό παρακάμπτοντας το τμήμα επείγοντων περιστατικών. Αυτό το μοντέλο είναι αρκετά διαδεδομένο στην Ευρώπη και το έχουν υιοθετήσει χώρες όπως η Γερμανία, η Αυστρία, η Γαλλία, η Μάλτα και η Ελλάδα [1, 2, 7, 27].

Το αγγλο – αμερικάνικο μοντέλο από την άλλη βασίζεται στη φιλοσοφία «scoop and run». Ο σκοπός αυτής της φιλοσοφίας είναι να μεταφέρει τους ασθενείς όσο το δυνατόν πιο γρήγορα στο κέντρο υγείας με λιγότερη προνοσοκομειακή ιατρική φροντίδα. Σε χώρες που ακολουθούν αυτό το μοντέλο η προνοσοκομειακή ιατρική υπηρεσία συνεργάζεται συνήθως με τις υπηρεσίες της αστυνομίας και της πυροσβεστικής αντί με τις υπηρεσίες υγείας και τα νοσοκομεία. Σε χώρες που ακολουθούν αυτό το μοντέλο, η επείγουσα ιατρική είναι πολύ ανεπτυγμένη και αναγνωρίζεται σαν ξεχωριστή ιατρική ειδικότητα. Σχεδόν όλοι ασθενείς σε αυτό το μοντέλο μεταφέρονται σε ανεπτυγμένα ειδικά κέντρα για έκτακτα περιστατικά παρά σε πτέρυγες νοσοκομείων. Μερικές από τις χώρες οι οποίες ακολουθούν αυτό το μοντέλο είναι οι ΗΠΑ, ο Καναδάς, η Νέα Ζηλανδία και η Αυστραλία [1, 2, 7, 27].

Στο ερώτημα ποιο μοντέλο είναι καλύτερο δεν υπάρχει μία ξεκάθαρη απάντηση. Πολλές έρευνες προσπάθησαν να συγκρίνουν αυτά τα δύο μοντέλα με όρους κόστους-αποτελεσματικότητας και αποτελέσματος αλλά τελικά φαίνεται ότι αυτά τα δύο μοντέλα στην πραγματικότητα δεν μπορούν να συγκριθούν και αυτό γιατί λειτουργούν σε διαφορετικά πλαίσια και έχουν διαφορετικούς τύπους απαιτήσεων να εκπληρώσουν. Επίσης, η έλλειψη ενιαίων προτύπων μεταξύ των δύο μοντέλων καθιστά τη σύγκριση αδύνατη. Οπότε προς το παρόν δεν υπάρχουν αποδείξεις ότι κάποιο μοντέλο είναι καλύτερο από το άλλο και οι έρευνες συνεχίζουν να δίνουν συγκρουόμενα συμπεράσματα [1].

2.4 Επίπεδα προνοσοκομειακής ιατρικής φροντίδας

Σε μία διεθνή κλίμακα, μπορούμε να πούμε ότι η προνοσοκομειακή φροντίδα χωρίζεται σε 4 κατηγορίες / επίπεδα. Αυτές οι κατηγορίες είναι η «ανοργάνωτη προνοσοκομειακή φροντίδα», η «Βασική Υποστήριξη της Ζωής» (Basic Life support – BLS), η «Προηγμένη Υποστήριξη της Ζωής» (Advanced Life Support – ALS) και η «Προηγμένη Υποστήριξη της Ζωής από ιατρικό προσωπικό» (DOC – ALS systems) [1, 2, 28].

Στην ανοργάνωτη προνοσοκομειακή φροντίδα, η οποία εμφανίζεται περισσότερο στις αναπτυσσόμενες χώρες, δεν υπάρχει ένα επίσημο σύστημα που να παρέχει προνοσοκομειακή φροντίδα στο σημείο του περιστατικού και οι ασθενείς μεταφέρονται σε εγκαταστάσεις ιατρικής περίθαλψης από το προσωπικό επιβολής του νόμου, με ιδιωτικά ή δημόσια οχήματα, με μοτοσυκλέτες, καροτσάκια και άλλα μεταφορικά μέσα [1, 2, 28].

Στο επίπεδο BLS, τεχνικοί επείγουσας ιατρικής Emergency Medical Technicians (EMTs) παρέχουν μη επεμβατική βασική φροντίδα στους ασθενείς με εξωτερικά τραύματα. Ο κύριος στόχος των EMTs σε αυτό το επίπεδο είναι η ταχεία μεταφορά των ασθενών σε μια ιατρική μονάδα. Πολλές μικρές πόλεις και αγροτικές περιοχές, σε χώρες υψηλού και μεσαίου εισοδήματος, εξυπηρετούνται από αυτό το είδος συστήματος [2].

Το επίπεδο ALS παρέχει πιο εξελιγμένη και επεμβατική θεραπεία, όπως ενδοφλέβια θεραπεία και ενδοτραχειακή διασωλήνωση [2].

Τέλος, στα Doc – ALS συστήματα, οι γιατροί επεμβαίνουν στον τόπο του περιστατικού και είναι υπεύθυνοι να παρέχουν προηγμένη ιατρική φροντίδα στους ασθενείς με εξωτερικά

τραύματα και στον τόπο του συμβάντος αλλά και κατά τη διακομιδή του ασθενούς στο κέντρο υγείας [2].

Η Ελλάδα έχει ανεπτυγμένες υπηρεσίες προνοσοκομειακής ιατρικής φροντίδας καθώς οι υπηρεσίες που προσφέρει ανήκουν στην κατηγορία Doc – ALS. Το ένα άτομο από τα πληρώματα των ασθενοφόρων και των κινητών μονάδων εντατικής θεραπείας είναι πάντα γιατρός. Οι κινητές μονάδες εντατικής θεραπείας παρέχουν προνοσοκομειακή φροντίδα σε πιο σοβαρά τραυματισμένους ασθενείς. Το κέντρο ελέγχου του ΕΚΑΒ διαβαθμίζει το αναφερόμενο περιστατικό με βάση την κρισιμότητά του και ανάλογα στέλνει ασθενοφόρο ή κινητή μονάδα εντατικής θεραπείας [1, 2, 4, 5].

Στον παρακάτω πίνακα φαίνεται το επίπεδο της προνοσοκομειακής ιατρικής φροντίδας σε διάφορες χώρες του κόσμου (όπου EMT, Emergency Medical Technician είναι το εκπαιδευμένο προσωπικό για παροχή βοήθειας έκτακτης ανάγκης [2]).

Χώρα	Περιοχές	Τύπος EMS	Αριθμός προσωπικού ανά ασθενοφόρο
Ιράν	Τεχεράνη	BLS	2 EMT - βασικής φροντίδας
Μεξικό	Νουέβο Λεόν	BLS	2 EMT - βασικής φροντίδας
Αυστραλία	Βικτώρια	ALS	2 EMT - προηγμένης φροντίδας
Νέα Ζηλανδία	Ώκλαντ	ALS	2 EMT - προηγμένης φροντίδας
Ολλανδία	---	ALS	1 EMT - βασικής φροντίδας, 1 EMT - προηγμένης φροντίδας
Ηνωμένο Βασίλειο	Μάντσεστερ, Γιόρκσαϊρ, Μέρσεϋ και Τρεντ	ALS	2 EMT - προηγμένης φροντίδας
ΗΠΑ	Κίνγκ Κάουντι	ALS	2 EMT - προηγμένης φροντίδας
Αυστρία	---	Doc-ALS	1 EMT - βασικής φροντίδας, 1 γιατρός
Καναδάς	Μόντρεαλ	Doc-ALS	1 EMT - βασικής φροντίδας, 1 γιατρός
Γερμανία	Εθνικά δεδομένα	Doc-ALS	1 EMT - βασικής φροντίδας, 1 γιατρός
Ελλάδα	Αθήνα, Βόλος, Κέρκυρα	Doc-ALS	1 EMT - βασικής φροντίδας, 1 γιατρός
Ηνωμένο Βασίλειο	Λονδίνο	Doc-ALS	1 EMT - βασικής φροντίδας, 1 γιατρός

Πίνακας 1: Γενικά χαρακτηριστικά προνοσοκομειακής ιατρικής φροντίδας ορισμένων χωρών [2]

2.5 Υπάρχουσες Τεχνολογίες στα Συστήματα Έκτακτης Ιατρικής Φροντίδας

Διεθνώς χρησιμοποιούνται πολλές διαφορετικές τεχνολογίες σε κάθε μία από τις διαδικασίες των EMS. Επιγραμματικά αυτές οι τεχνολογίες φαίνονται στον πίνακα 2 [8]. Στη συνέχεια παρουσιάζονται οι κυριότερες τεχνολογίες πιο αναλυτικά.

Χρονικά διαστήματα διαδικασιών	Παραδείγματα τεχνολογιών που χρησιμοποιούνται
Προετοιμασία πριν το περιστατικό	Ατομικό ηλεκτρονικό αρχείο υγείας για έκτακτα περιστατικά (PHR - Electronical Personal Health Record)
Από τη στιγμή του συμβάντος μέχρι τη στιγμή της ειδοποίησης	911, E-911, Αυτόματος εντοπισμός ατυχημάτων (ACN - Automatic Crash Notification)
Από τη στιγμή της ειδοποίησης μέχρι τη στιγμή της ανάθεσης του περιστατικού	Ανάθεση με τη βοήθεια του υπολογιστή (CAD - Computer Aided Dispatch), Συστήματα διαχείρισης της κυκλοφορίας (TMS - Traffic Management Systems), Σύστημα Γεωγραφικών Πληροφοριών/Παγκόσμιο Σύστημα Εντοπισμού (GIS/GPS - Geographical Information System/Global Positioning System), Τερματικά Κινητών Δεδομένων (MDT's - Mobile Data Terminals), εργαλεία υποστήριξης αποφάσεων, ασύρματοι πομποδέκτες, συσκευές τηλεειδοποίησης (βομβητές), κινητά τηλέφωνα
Από τη στιγμή της ανάθεσης μέχρι τη στιγμή άφιξης στον τόπο του συμβάντος (επιτόπου φροντίδα)	Ανάθεση με τη βοήθεια του υπολογιστή, Συστήματα διατήρησης αρχείου φροντίδας των ασθενών (PCR - Patient Care Record systems), Συστήματα διαχείρισης της κυκλοφορίας, Σύστημα Γεωγραφικών Πληροφοριών/Παγκόσμιο Σύστημα Εντοπισμού, Τερματικά Κινητών Δεδομένων, εργαλεία υποστήριξης αποφάσεων, ασύρματοι πομποδέκτες, συσκευές τηλεειδοποίησης (βομβητές), κινητά τηλέφωνα, συστήματα πλοήγησης
Από τη στιγμή της άφιξης στον τόπο του συμβάντος μέχρι τη στιγμή της αναχώρησης προς το νοσοκομείο/κέντρο υγείας (επιτόπου φροντίδα και μεταφορά)	Συστήματα διατήρησης αρχείου φροντίδας των ασθενών, εργαλεία υποστήριξης αποφάσεων, εφαρμογές τηλειατρικής (φροντίδα από απόσταση), ασύρματη επικοινωνία δεδομένων, συστήματα εύρεσης διαθεσιμότητας νοσοκομείων
Από τη στιγμή της αναχώρησης προς το νοσοκομείο/κέντρο υγείας μέχρι τη στιγμή της άφιξης στο νοσοκομείο/κέντρο υγείας (μεταφορά και μεταπομπή στο τμήμα επειγόντων περιστατικών του νοσοκομείου)	Συστήματα διατήρησης αρχείου φροντίδας των ασθενών, Συστήματα διαχείρισης της κυκλοφορίας, Σύστημα Γεωγραφικών Πληροφοριών/Παγκόσμιο Σύστημα Εντοπισμού, συστήματα πλοήγησης, συστήματα εύρεσης διαθεσιμότητας νοσοκομείων
Από τη στιγμή της εισαγωγής του ασθενούς στο νοσοκομείο μέχρι τη στιγμή της αναχώρησης του από αυτό	Διατήρηση αρχείου εισαγωγής ασθενών από τα τμήματα επειγόντων περιστατικών των νοσοκομείων, διατήρηση αρχείου πάθησης, διατήρηση ηλεκτρονικού ιατρικού αρχείου, κλινικά πληροφοριακά συστήματα, ηλεκτρονικά εργαστήρια/ακτινολογικά συστήματα, εργαλεία υποστήριξης ιατρικών αποφάσεων
Αξιολόγηση μετά το περιστατικό	αποθήκες δεδομένων, επιχειρηματική ευφυΐα, συστήματα ανάλυσης των περιστατικών

Πίνακας 2: Υπάρχουσες τεχνολογίες ανά διαδικασία EMS [8]

2.5.1 Ατομικό Ηλεκτρονικό Αρχείο Υγείας (Electronic Personal Health Record)

Το Ατομικό Ηλεκτρονικό Αρχείο Υγείας, ή εν συντομία PHR, είναι ένα αρχείο το οποίο περιέχει προσωπικά στοιχεία που αφορούν την υγεία ενός ασθενή καθώς και πληροφορίες σχετικές με την περίθαλψή του. Το αρχείο αυτό διατηρείται και ενημερώνεται από τον ίδιο τον ασθενή. Ο σκοπός του PHR είναι να παρέχει μία πλήρη και ακριβή περίληψη του ιατρικού ιστορικού ενός ατόμου που θα είναι διαθέσιμη online [9].

Τα δεδομένα που περιέχουν τα PHR μπορεί να περιλαμβάνουν χρόνιες ασθένειες, αποτελέσματα εξετάσεων, ακτινογραφίες, οικογενειακό ιστορικό, αλλεργίες, αλλεργικές αντιδράσεις σε κάποιο φάρμακο, πιθανές εγχειρήσεις κ.α. Αυτά τα δεδομένα εισάγονται από τον χρήστη ηλεκτρονικά, συνήθως με τη συμπλήρωση μίας φόρμας [9].

Νοσοκομεία και κέντρα υγείας δίνουν τη δυνατότητα στους ασθενείς να αποστείλουν τα PHR τους έτσι ώστε τα αρχεία αυτά να μπορούν σε έκτακτες καταστάσεις να χρησιμοποιηθούν ούτως ώστε η επέμβαση του EMS να είναι πιο ακριβής και αποτελεσματική [9].

2.5.2 E-911

Στη Βόρεια Αμερική υπάρχει το σύστημα E - 911 (Enhanced 911) το οποίο είναι το αντίστοιχο 166 της Ελλάδας. Αυτό το σύστημα ουσιαστικά συνδέει αυτόν που καλεί στο 911 κατευθείαν με το κέντρο αποστολής το οποίο αποστέλλει τους κατάλληλους διασώστες στον τόπο του περιστατικού. Το E-911 αυτόματα βρίσκει την τοποθεσία του καλούντος και τη δίνει στο κέντρο αποστολής σε αντίθεση με το 911 που δεν έχει αυτή τη δυνατότητα. Αυτό είναι πολύ σημαντικό

σε ορισμένες περιπτώσεις όπου το περιστατικό είναι πολύ σοβαρό και ο καλών δεν μπορεί να δώσει πληροφορίες για την τοποθεσία του [10].

2.5.3 Αυτόματος Εντοπισμός Ατυχημάτων (Automatic Crash Notification)

Η τεχνολογία Automatic Crash Notification (ACN) χρησιμοποιείται σε περιστατικά ατυχημάτων στα οποία εμπλέκονται οχήματα. Είναι μία τεχνολογία η οποία είναι κομμάτι των τηλεματικών συστημάτων των οχημάτων [11]. Η τεχνολογία αυτή μπορεί να είναι ιδιαίτερα σημαντική, δεδομένου ότι επιτρέπει στα οχήματα να ειδοποιούν αυτόματα ένα τηλεφωνικό κέντρο σε περίπτωση σύγκρουσης του οχήματος. Ένα σύστημα ACN αποτελείται από τα τέσσερα ακόλουθα τμήματα: τους αισθητήρες (που βρίσκονται συνήθως στην μπροστινή και πλαϊνή μεριά του οχήματος ή ακόμα και στους αερόσακους), το Διαγνωστικό Εργαλείο Αισθητήρων (Sensing Diagnostic Module – SDM, το οποίο περιλαμβάνει έναν καταγραφέα γεγονότων και ένα μετρητή επιτάχυνσης), το Εργαλείο Επικοινωνίας και Διεπαφής Οχήματος (Vehicle Communication and Interface Module - VCIM) και μία κυψελοειδή κεραία [11,12].

Όταν ένα αυτοκίνητο εμπλακεί σε ένα αυτοκινητιστικό ατύχημα, οι αισθητήρες του μεταδίδουν πληροφορίες στο SDM. Το SDM στη συνέχεια στέλνει πληροφορίες σχετικά με τη σοβαρότητα του ατυχήματος στο VCIM, το οποίο χρησιμοποιεί την κυψελοειδή κεραία για να στείλει ένα μήνυμα στο τηλεφωνικό κέντρο. Όταν ένας χειριστής του τηλεφωνικού κέντρου λάβει το μήνυμα χρησιμοποιεί την τεχνολογία GPS για να αναγνωρίσει την τοποθεσία του οχήματος και προσπαθεί να έρθει σε επαφή με τον οδηγό. Εάν δεν υπάρξει απάντηση, ο χειριστής επικοινωνεί με το ΕΚΑΒ για να αναλάβει το περιστατικό [12].

Συστήματα ACN έχουν αναπτυχθεί από διάφορες αυτοκινητοβιομηχανίες όπως η BMW (BMW Assist), η Mercedes (Mercedes – Benz Tele Aid) και η Ford (Ford's VEMS/RESCU system).

2.5.4 Ανάθεση με τη βοήθεια υπολογιστή (Computer Aided Dispatch)

Η τεχνολογία Computer Aided Dispatch (CAD) είναι ένας συνδυασμός υλικού και λογισμικού που παρέχει είσοδο δεδομένων, προτείνει πόρους/οχήματα/μεταφορικά μέσα, ειδοποιεί, εντοπίζει και παρακολουθεί την πορεία αυτών των πόρων πριν, κατά τη διάρκεια και μετά το συναγερμό και διατηρεί αρχείο αυτών των συναγερωμένων και των αλλαγών κατάστασης για μετέπειτα ανάλυση [13].

Στα πιο μοντέρνα συστήματα CAD, οι χειριστές των τηλεφωνικών κέντρων λαμβάνουν κλήσεις για έκτακτα περιστατικά μέσω του συστήματος E911 που περιγράφηκε παραπάνω. Αυτό το σύστημα είναι συνδεδεμένο σε υπολογιστή και διαθέτει αυτόματη αναγνώριση αριθμού και αυτόματο εντοπισμό θέσης. Το σύστημα αυτό αυτόματα εισάγει τον τηλεφωνικό αριθμό και τη διεύθυνση του καλούντος στο σύστημα CAD, μειώνοντας το χρόνο που απαιτείται για την επεξεργασία του αιτήματος για βοήθεια. Ότι πληροφορίες σχετικά με το περιστατικό εξάγει ο χειριστής από την κλήση τις εισάγει και αυτές στο CAD. Όταν το σύστημα CAD διαπιστώσει ότι έχει αρκετές πληροφορίες ώστε να μπορεί να προτείνει ένα πρότυπο απάντησης θα το πράξει δίνοντας αυτή την πρόταση στο χειριστή για έγκριση. Δεν υπάρχει καμία ανάγκη για τον χειριστή να συμβουλευτεί χάρτες, να ελέγξει τη διαθεσιμότητα των μονάδων, να καθορίσει την κατάλληλη απάντηση, να ειδοποιήσει με ασύρματο την μονάδα και να ανακοινώσει το περιστατικό. Εάν ο χειριστής συμφωνεί με την πρόταση του CAD τότε με το πάτημα ενός κουμπιού ο συναγερμός προωθείται στην κατάλληλη μονάδα. Στα περισσότερα συστήματα αυτό μπορεί να επιτευχθεί μέσα σε 30 με 45 δευτερόλεπτα [14,15,16].

Τα μοντέρνα συστήματα CAD είναι ιδιαίτερα περίπλοκα. Δεν περιορίζονται στην επεξεργασία του συναγερωμού αλλά εκτελούν και πολλές άλλες λειτουργίες. Τα συστήματα CAD μπορούν να εμφανίσουν στον χειριστή το ιατρικό ιστορικό των ασθενών το οποίο μπορεί να το προωθήσει και στα πληρώματα των μονάδων για αποτελεσματικότερη επέμβαση στο περιστατικό. Μπορούν να προτείνουν αλλαγές στην τοποθεσία μίας μονάδας για να μειώσουν το χρόνο σε μία πιθανή αποστολή της σε περιστατικό. Μπορούν, επίσης, να στέλνουν στα πληρώματα χάρτες με τη συντομότερη διαδρομή, να καταγράφουν όλες τις πληροφορίες της εξέλιξης του περιστατικού σε ένα αρχείο, να δημιουργούν ιστορικό περιστατικών και πολλές άλλες λειτουργίες [14,15,16].

2.5.5 Σύστημα Διαχείρισης Κυκλοφορίας (Traffic Management System)

Το Σύστημα Διαχείρισης Κυκλοφορίας (TMS) αποτελεί υποκατηγορία του γενικότερου τομέα των Συστημάτων Μεταφορών. Ουσιαστικά είναι ένα σύστημα διαχείρισης που ενσωματώνει την τεχνολογία κυρίως για τη βελτίωση της κυκλοφορίας των οχημάτων και τη βελτίωση της ασφάλειας στους δρόμους.

Πραγματικού χρόνου δεδομένα από κάμερες που ελέγχουν την κυκλοφορία, από αισθητήρες ταχύτητας κ.τ.λ. κατευθύνονται σε ένα Κέντρο Διαχείρισης της Κυκλοφορίας όπου ενσωματώνονται στο σύστημα και επεξεργάζονται για τυχόν εντοπισμό ατυχημάτων ούτως ώστε να ληφθούν οι κατάλληλες αποφάσεις για έγκαιρη αποστολή βοήθειας [17].

2.5.6 Σύστημα Γεωγραφικών Πληροφοριών/Παγκόσμιο Σύστημα Εντοπισμού (Geographic Information System/ Global Positioning System)

Υπάρχουν πολλοί διαφορετικοί ορισμοί στη βιβλιογραφία για το τι είναι το Geographic Information System (GIS). Ένας γενικός ορισμός είναι ο εξής: Το GIS είναι μία ειδικού σκοπού ψηφιακή βάση δεδομένων στην οποία ένα κοινό χωρικό σύστημα συντεταγμένων είναι το κύριο μέσο αναφοράς. Ένα ολοκληρωμένο σύστημα GIS απαιτεί να υπάρχει ένα μέσο για:

- εισαγωγή δεδομένων από χάρτες, αεροφωτογραφίες, δορυφόρους, έρευνες και άλλες πηγές
- αποθήκευση και ανάκτηση δεδομένων καθώς και επερωτήσεις πάνω στα δεδομένα
- μετασχηματισμό, ανάλυση και μοντελοποίηση των δεδομένων, συμπεριλαμβανομένων των χωρικών στατιστικών, και
- δεδομένα αναφοράς όπως χάρτες, αναφορές καθώς και σχέδια [18].

Η τεχνολογία του GIS βελτιστοποιεί τα συστήματα Προνοσοκομειακής Ιατρικής Φροντίδας παρέχοντας σημαντικές πληροφορίες στο ΕΚΑΒ. Αυτές οι πληροφορίες βοηθούν στην ανάλυση ιστορικών περιστατικών με σκοπό τον προσδιορισμό των αναγκών του προσωπικού και των πόρων, καθώς και στον καθορισμό της βέλτιστης τοποθέτησης ενός ασθενοφόρου μέσω ανάλυσης των δεδομένων της βάσης GIS. Επίσης, τα συστήματα GIS παρέχουν στους χειριστές του κέντρου αποστολής μία ολοκληρωμένη εικόνα των ασθενοφόρων και των λοιπών μονάδων καθώς και την ικανότητα αντίδρασης τους σε περιπτώσεις συμβάντων [19].

Το Global Positioning System (GPS) είναι ουσιαστικά ένα παγκόσμιο σύστημα εντοπισμού που αναπτύχθηκε για στρατιωτικούς σκοπούς αρχικά από τις ΗΠΑ αλλά αργότερα έγινε δημόσιο έτσι ώστε και απλοί πολίτες να μπορούν να απολαμβάνουν τις υπηρεσίες που μπορεί να προσφέρει. Το σύστημα αυτό αποτελείται από 27 δορυφόρους που περιστρέφονται γύρω από τη γη. Οι τροχιές τους είναι έτσι σχεδιασμένες έτσι ώστε κάθε χρονική στιγμή, σε οποιοδήποτε μέρος του κόσμου να είναι ορατοί τουλάχιστον 4 δορυφόροι. Χρησιμοποιώντας τα συνεχόμενα σήματα που στέλνουν αυτοί οι δορυφόροι και κάνοντας κάποιους πολύπλοκους μαθηματικούς υπολογισμούς (trilateration), ένας δέκτης GPS μπορεί ανά πάσα στιγμή να γνωρίζει την τοποθεσία του [20].

Το GIS έχει ενσωματωθεί με την τεχνολογία GPS, με σκοπό να παρέχει ένα εξαιρετικά σημαντικό πλεονέκτημα στις CAD λειτουργίες. Η ικανότητα εντοπισμού των θέσεων των ασθενοφόρων καθώς και των έκτακτων περιστατικών εξασφαλίζει ότι θα σταλεί στο περιστατικό το πλησιέστερο διαθέσιμο ασθενοφόρο. Αυτή η τεχνολογία παρέχει και ένα άλλο εργαλείο για τη μείωση του χρόνου που απαιτείται από τη στιγμή της ειδοποίησης για ένα περιστατικό μέχρι τη στιγμή της άφιξης ενός ασθενοφόρου στο συμβάν. Δεδομένα του GIS (τα οποία με κάποια διαδικασία αναπαριστώνται πάνω σε ένα χάρτη) μπορούν να σταλούν στα επιλεγμένα ασθενοφόρα παρέχοντάς τους τη δυνατότητα, βλέποντας αυτά τα δεδομένα σε ένα χάρτη, να φθάσουν πιο γρήγορα στο περιστατικό. Επίσης τα συστήματα GIS ενημερώνουν τα ασθενοφόρα σε ποια σημεία του χάρτη υπάρχουν πορείες, έργα σε δρόμους κτλ [21].

2.5.7 Τερματικό Κινητών Δεδομένων (Mobile Data Terminal)

Τα Τερματικά Κινητών Δεδομένων (MDTs) είναι συσκευές που τοποθετούνται συνήθως σε οχήματα και τα οποία λαμβάνουν μηνύματα από κάποιο κέντρο ελέγχου παρουσιάζοντας όλες Διαδικτυακό Πληροφοριακό Σύστημα Διαχείρισης Έκτακτων Περιστατικών με Δυνατότητα Πρόσβασης μέσω Κινητών Συσκευών με Λειτουργικό Android 24

τις απαραίτητες πληροφορίες στην οθόνη τους. Αντικαθιστούν τη ραδιοεπικοινωνία μεταξύ του οδηγού και του χειριστή στο κέντρο ελέγχου. Αυτόματα στέλνουν την τοποθεσία του οχήματος, των αριθμό των επιβατών, την κατάσταση της μηχανής και άλλες πληροφορίες.[22]

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

3. ΥΠΟΒΑΘΡΟ ΤΕΧΝΟΛΟΓΙΩΝ

3.1 Android

Μία από τις κυριότερες τεχνολογίες που χρησιμοποιήθηκαν στην παρούσα μεταπτυχιακή διατριβή είναι η τεχνολογία της Google, το γνωστό και διαδεδομένο πλέον λειτουργικό σύστημα για κινητές συσκευές, το Android. Στην παρακάτω ενότητα θα παρουσιάσουμε το λειτουργικό σύστημα Android, τις κυριότερες λειτουργίες του, τα πλεονεκτήματά του σε σχέση με τους ανταγωνιστές του, την αρχιτεκτονική του συστήματος καθώς και των εφαρμογών του.

3.1.1 Τι είναι το Android

Το Android είναι ένα λειτουργικό σύστημα για κινητές συσκευές το οποίο είναι βασισμένο σε μία τροποποιημένη έκδοση του Linux. Αρχικά αναπτύχθηκε από την Android, Inc, μία εταιρεία η οποία ιδρύθηκε το 2003 στο Palo Alto της Καλιφόρνια των ΗΠΑ από τους Andy Rubin, Rich Miner, Nick Sears και Chris White. Το 2005 η Google θέλοντας να διεισδύσει στο χώρο της κινητής τηλεφωνίας αγόρασε την Android και παράλληλα την προγραμματιστική της έρευνα και εργασία [29, 30].

Το Android παρουσιάστηκε για πρώτη φορά το 2007 από μια ομάδα εταιρειών που είναι γνωστές ως Open Handset Alliance (OHA), με επικεφαλής την Google. Σήμερα, πολλές εταιρείες, και τα αρχικά μέλη της OHA και άλλες, έχουν επενδύσει σε μεγάλο βαθμό στο Android, κυρίως με τη μορφή της διάθεσης σημαντικών τεχνικών πόρων με σκοπό τη βελτίωση του Android και την προώθηση των συσκευών Android στην αγορά [31].

Ο απώτερος σκοπός του Android είναι να υπάρχει πάντα μια ανοιχτή πλατφόρμα διαθέσιμη την οποία θα χρησιμοποιούν οι κατασκευαστές πρωτότυπων συσκευών (OEM's), οι πάροχοι υπηρεσιών κινητής τηλεφωνίας καθώς και οι προγραμματιστές για να κάνουν τις καινοτόμες ιδέες τους πραγματικότητα. Γι' αυτό ο περισσότερος πηγαίος κώδικας του Android δημιουργήθηκε με τη βοήθεια της άδειας ανοιχτού κώδικα της Apache, που σημαίνει ότι οποιοσδήποτε που θέλει να χρησιμοποιήσει το Android μπορεί να το κάνει κατεβάζοντας ολόκληρο τον πηγαίο κώδικα του Android [30, 31].

Με τη δημιουργία του Android επιδιώχθηκε να διασφαλιστεί ότι δεν θα υπήρχε κεντρικό σημείο αποτυχίας, όπου μία εταιρία θα μπορούσε να περιορίσει ή να ελέγξει τις καινοτομίες του οποιοδήποτε άλλου. Γι' αυτό άλλωστε και το Android δημιουργήθηκε σαν μία ανοιχτού κώδικα πλατφόρμα. Ο γενικότερος στόχος του Android ήταν να δημιουργήσει ένα επιτυχημένο ρεαλιστικό προϊόν το οποίο θα βελτίωνε την κινητή εμπειρία των τελικών χρηστών [30].

3.1.2 Πλεονεκτήματα του Android

Υπάρχουν πάρα πολλές κινητές πλατφόρμες στην αγορά σήμερα, όπως το Symbian, το iPhone, το Windows Mobile, το BlackBerry και πολλά άλλα. Τι είναι αυτό λοιπόν αυτό που κάνει το λειτουργικό Android ξεχωριστό; Σε αυτό το ερώτημα θα απαντήσουμε παρακάτω [32].

Το Android, όπως προαναφέρθηκε, είναι μία ανοιχτή ελεύθερη πλατφόρμα βασισμένη στο Linux και ανοιχτού κώδικα. Αυτό είναι κάτι που αρέσει στους κατασκευαστές κινητών συσκευών επειδή μπορούν να χρησιμοποιήσουν και να προσαρμόσουν την πλατφόρμα χωρίς να χρειάζεται να πληρώσουν δικαιώματα σε κάποια εταιρεία [31, 32].

Επίσης, αποτελείται από μία αρχιτεκτονική η οποία βασίζεται σε components. Αυτό πρακτικά έχει σαν αποτέλεσμα κομμάτια μίας εφαρμογής να χρησιμοποιούνται (άθικτα ή τροποποιημένα από τον προγραμματιστή) σε μία άλλη εφαρμογή με τρόπο που δεν είχε Διαδικτυακό Πληροφοριακό Σύστημα Διαχείρισης Έκτακτων Περιστατικών με Δυνατότητα Πρόσβασης μέσω Κινητών Συσκευών με Λειτουργικό Android



Εικόνα 4: Λογότυπο Android [43]

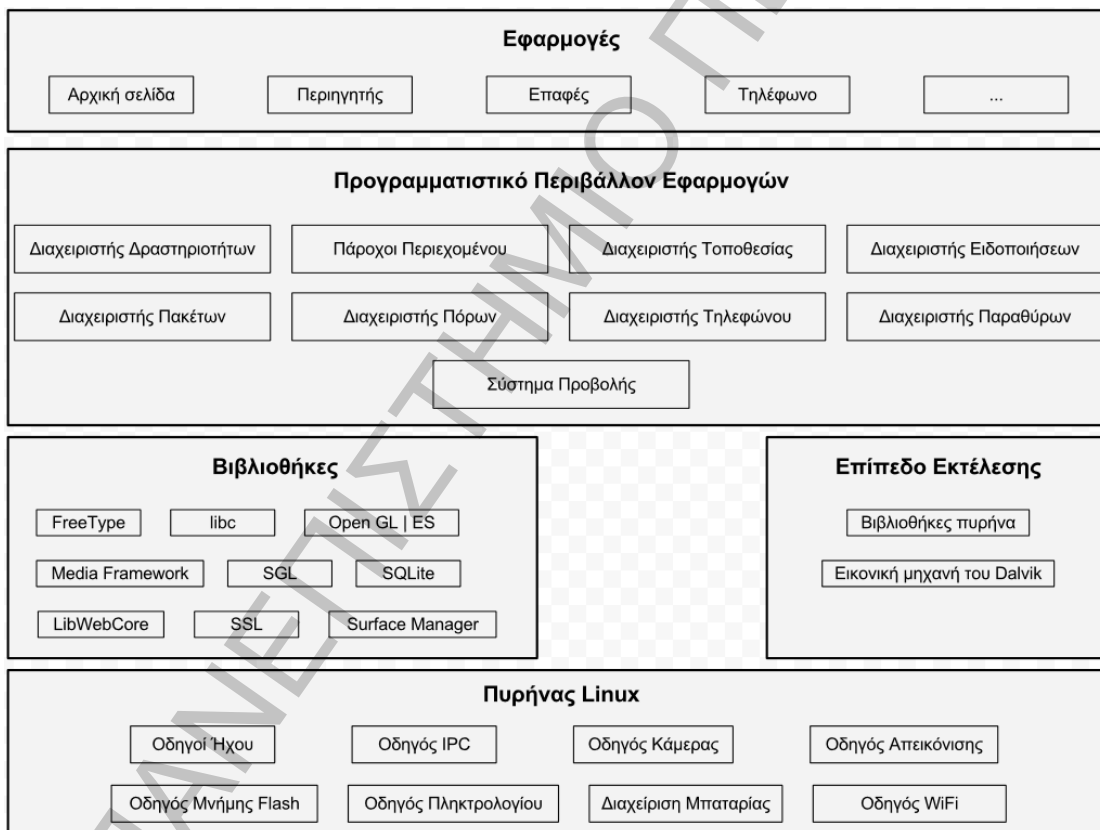
φανταστεί αρχικά ο προγραμματιστής γεγονός που ενισχύει την δημιουργικότητα στο χώρο των κινητών συσκευών [31, 32].

Το Android επίσης διαθέτει πάρα πολλές ενσωματωμένες υπηρεσίες. Μερικές από αυτές είναι οι υπηρεσίες που ασχολούνται με τη γεωγραφική θέση της κινητής συσκευής χρησιμοποιώντας τεχνολογίες GPS και triangulation του σήματος από τις κυψέλες. Επίσης διαθέτει μία SQL βάση δεδομένων η οποία μπορεί να χρησιμοποιηθεί για τοπική αποθήκευση δεδομένων. Αυτές και άλλες δυνατότητες του Android βελτιώνουν την λειτουργία των κινητών συσκευών και μειώνουν το κόστος ανάπτυξης μίας εφαρμογής [31, 32].

Επιπλέον διαθέτει αυτόματη διαχείριση του κύκλου ζωής των εφαρμογών, που σημαίνει ότι ο χρήστης δεν χρειάζεται πλέον να ανησυχεί για το ποιες εφαρμογές είναι ενεργές ή να κλείνει προγράμματα για να μπορέσει να τρέξει άλλα. Τέλος διαθέτει υψηλής ποιότητας γραφικά και ήχο καθώς και φορητότητα σε ένα ευρύ φάσμα τρέχοντος και μελλοντικού υλικού [31, 32].

3.1.3 Αρχιτεκτονική του Android

Η αρχιτεκτονική του Android αποτελείται από τέσσερα επίπεδα όπου το κάθε ένα με τη σειρά του αποτελείται από διάφορα στοιχεία. Κάθε επίπεδο, επίσης, χρησιμοποιεί τις υπηρεσίες που παρέχονται από το επίπεδο που βρίσκεται από κάτω του. Η αρχιτεκτονική του Android φαίνεται σχηματικά στην εικόνα 5 [44].



Εικόνα 5: Αρχιτεκτονική του Android [44]

Το Android είναι κατασκευασμένο πάνω στον πυρήνα του Linux. Ο πυρήνας του Linux είναι το πιο χαμηλό επίπεδο της αρχιτεκτονικής και αυτό το οποίο αλληλεπιδρά άμεσα με το υλικό της συσκευής. Το επίπεδο αυτό είναι επιφορτισμένο με λειτουργίες όπως η διαχείριση της μνήμης, των διεργασιών και της ενέργειας, η ασφάλεια κ.τ.λ. Το επίπεδο αυτό δεν είναι ορατό από τους τελικούς χρήστες ούτε και από τους προγραμματιστές. Γενικά, μόνο οι κατασκευαστές συσκευών και υλικού ασχολούνται με αυτό το επίπεδο κυρίως προσαρμόζοντας το στο εκάστοτε υλικό της συσκευής [32, 36, 37].

Διαδίκτυο Πληροφοριακό Σύστημα Διαχείρισης Έκτακτων Περιστατικών με Δυνατότητα Πρόσβασης μέσω Κινητών Συσκευών με Λειτουργικό Android

Το επίπεδο πάνω από τον πυρήνα του Linux περιλαμβάνει τις ενσωματωμένες βιβλιοθήκες του Android καθώς και κομμάτι της αρχιτεκτονικής μέσα στο οποίο γίνεται η εκτέλεση όλων των εντολών. Οι βιβλιοθήκες λειτουργούν σαν ένα μεταφραστικό επίπεδο μεταξύ του πυρήνα και του πλαισίου εφαρμογών. Οι βιβλιοθήκες αυτές είναι γραμμένες σε C/C++ αλλά παρέχονται στους προγραμματιστές μέσω ενός Java API. Οι προγραμματιστές μπορούν να χρησιμοποιήσουν το πλαίσιο εφαρμογών της Java για να αποκτήσουν πρόσβαση σε αυτές τις βιβλιοθήκες. Μερικές από τις σημαντικότερες βιβλιοθήκες περιλαμβάνουν τα ακόλουθα [32, 36, 37]:

- Διαχειριστής οθόνης (Surface Manager): Αυτή η βιβλιοθήκη έχει πρόσβαση στο υποσύστημα αναπαράστασης γραφικών στην οθόνη όπου και συνθέτει διδιάστατα και τρισδιάστατα επίπεδα γραφικών από πολλαπλές εφαρμογές
- SQL βάση δεδομένων: Το Android περιλαμβάνει την ελαφριά έκδοση της SQL, την SQLite βάση δεδομένων, η οποία χρησιμεύει για την προσωρινή αποθήκευση δεδομένων των εφαρμογών
- LibWebCore: Επιτρέπει πρόσβαση στον φυλλομετρητή ιστού
- Media Framework: Επιτρέπει πρόσβαση σε συναρτήσεις ήχου και βίντεο
- OpenGL | ES – SGL: Επιτρέπει την πρόσβαση στις μηχανές σχεδιασμού 2D και 3D γραφικών.

Το επίπεδο εκτέλεσης βρίσκεται και αυτό πάνω από το επίπεδο του πυρήνα και μοιράζεται τον ίδιο χώρο με το επίπεδο των βιβλιοθηκών. Το επίπεδο αυτό περιλαμβάνει την εικονική μηχανή του Dalvik (Dalvik Virtual Machine – Dalvik VM) καθώς και τις βιβλιοθήκες πυρήνα. Η VM του Dalvik είναι ένα πολύ σημαντικό στοιχείο για το λειτουργικό Android [32, 34, 36, 37].

Η VM του Dalvik είναι μία εικονική μηχανή σχεδιασμένη και ανεπτυγμένη από τον Dan Bornstein της Google. Η μηχανή αυτή αρχικά σχεδιάστηκε για να επιτρέψει την εκτέλεση εφαρμογών σε συσκευές με πολύ περιορισμένους πόρους. Γενικά μία εικονική μηχανή είναι ένα απομονωμένο λειτουργικό σύστημα το οποίο τρέχει μέσα σε ένα άλλο λειτουργικό σύστημα το οποίο το φιλοξενεί. Το πλεονέκτημα των VM είναι ότι οι εφαρμογές θα εκτελεστούν σαν να εκτελούνταν σε μία φυσική μηχανή. Αυτό έχει σαν αποτέλεσμα ο κώδικας των εφαρμογών να είναι ανεξάρτητος του υλικού και να τρέχει σε οποιαδήποτε συσκευή διαθέτει μία συμβατή VM. Στο Android τώρα ο κώδικας των εφαρμογών μεταγλωττίζεται σε ένα σύνολο εντολών ανεξάρτητων του υλικού που ονομάζονται bytecodes, οι οποίες εκτελούνται από την VM του Dalvik [32, 36, 37].

Το παραπάνω επίπεδο είναι το πλαίσιο εφαρμογών. Το πλαίσιο εφαρμογών περιέχει όλα εκείνα τα στοιχεία τα οποία χρειάζεται ο προγραμματιστής έτσι ώστε να μπορεί να αναπτύσσει εφαρμογές. Το framework διατίθεται προεγκατεστημένο με το Android αλλά ένας προγραμματιστής μπορεί να το επεκτείνει και με δικά του στοιχεία. Τα πιο σημαντικά στοιχεία του framework είναι τα ακόλουθα [32, 36, 37]:

- Διαχειριστής Δραστηριοτήτων (Activity Manager): Αυτό το στοιχείο ελέγχει τον κύκλο ζωής των εφαρμογών και διατηρεί μία στοίβα των δραστηριοτήτων για περιήγηση στις εφαρμογές από τους χρήστες
- Πάροχοι Περιεχομένου (Content Providers): Αυτό το στοιχείο ενσωματώνει δεδομένα τα οποία μπορούν να μοιραστούν μεταξύ των εφαρμογών
- Διαχειριστής Πόρων (Resource Manager): Αυτό το στοιχείο επιτρέπει σε μία εφαρμογή να αποκτήσει πρόσβαση στους πόρους της (όπως εικόνες, xml αρχεία κτλ)
- Διαχειριστής ειδοποιήσεων (Notification Manager): Αυτό το στοιχείο επιτρέπει σε μία εφαρμογή να ειδοποιήσει το χρήστη για ένα σημαντικό γεγονός (ζυπνητήριο, ραντεβού, άφιξη sms κτλ) χωρίς να διακόπτει την τρέχουσα εφαρμογή που τρέχει ο χρήστης στη συσκευή του
- Διαχειριστής τοποθεσίας (Location Manager): Αυτό το στοιχείο επιτρέπει στο κινητό να γνωρίζει πού ακριβώς βρίσκεται.

Το τελευταίο επίπεδο είναι εκεί που βρίσκονται οι εφαρμογές. Αυτό το επίπεδο είναι ορατό σε όλους και κυρίως στους τελικούς χρήστες [32, 36, 37].

3.1.4 Αρχιτεκτονική των εφαρμογών

Βασικά χαρακτηριστικά των εφαρμογών

Στο Android υπάρχει μία εφαρμογή στο προσκήνιο η οποία τυπικά καταλαμβάνει όλη την οθόνη της κινητής συσκευής εκτός από τη γραμμή κατάστασης. Η πρώτη εφαρμογή που εκτελείται όταν ο χρήστης ανοίγει το κινητό του είναι η εφαρμογή Home. Στη συνέχεια ο χρήστης μπορεί να ανοίξει και άλλες εφαρμογές τις οποίες το λειτουργικό σύστημα τις ανοίγει και τις φέρνει στο προσκήνιο. Αυτές οι εφαρμογές αποθηκεύονται σε μία στοίβα από τον Activity Manager και πάνω - πάνω στη στοίβα βρίσκεται εφαρμογή που βρίσκεται στο προσκήνιο εκείνη τη χρονική στιγμή. Ο χρήστης μπορεί με το πάτημα ενός κουμπιού να επιστρέψει σε προηγούμενες εφαρμογές (οι οποίες διαγράφονται από τη στοίβα) με τον ίδιο τρόπο που θα έκανε περιήγηση σε έναν web browser [32].

Κάθε εφαρμογή σε μία συσκευή Android αποτελεί ουσιαστικά ένα διαφορετικό χρήστη σε ένα πολλαπλών-χρηστών σύστημα Linux. Το σύστημα αντιστοιχεί κάθε εφαρμογή με ένα user ID το οποίο το γνωρίζει μόνο το σύστημα και όχι η εφαρμογή. Στη συνέχεια το λειτουργικό δίνει δικαιώματα σε όλα τα αρχεία της εφαρμογής, τέτοια ώστε μόνο το user ID που αντιστοιχεί σε αυτή την εφαρμογή να μπορεί να τα επεξεργαστεί. Επίσης, κάθε εφαρμογή εκτελείται όπως προαναφέρθηκε στη δική της VM, με αποτέλεσμα να τρέχει απομονωμένη από τις άλλες εφαρμογές, καθώς και στη δικιά της Linux διεργασία. Το android είναι υπεύθυνο για να ξεκινήσει τη διεργασία όταν κάποιο από τα στοιχεία της εφαρμογής χρειαστεί να εκτελεστεί και είναι επίσης υπεύθυνο να την «σκοτώσει» όταν δε χρειάζεται άλλο ή όταν το σύστημα πρέπει να ανακτήσει μνήμη για μία άλλη εφαρμογή. Με αυτό τον τρόπο δημιουργείται ένα πολύ ασφαλές περιβάλλον καθώς μία εφαρμογή εκτελείται απομονωμένη και δεν μπορεί να έχει πρόσβαση σε τμήματα του συστήματος για τα οποία δεν έχει δικαιώματα [38].

Συστατικά των εφαρμογών

Κάθε εφαρμογή αποτελείται από τέσσερα στοιχεία (activities, services, content providers, broadcast receivers) τα οποία μοιράζονται ένα κοινό σύστημα αρχείων, μία βάση δεδομένων, πόρους και φυσικά τη διεργασία Linux. Για την υλοποίηση μίας εφαρμογής πρέπει να χρησιμοποιηθεί τουλάχιστον ένα από τα τέσσερα αυτά στοιχεία. Δεν είναι απαραίτητο να χρησιμοποιηθούν και τα τέσσερα [32, 37, 39, 40].

Δραστηριότητες (Activities)

Μία δραστηριότητα είναι ουσιαστικά μία οθόνη διεπαφής χρήστη. Πιο απλά είναι το στοιχείο εκείνο που παρέχει μία διεπαφή χρήστη έτσι ώστε ο χρήστης να μπορεί να αλληλεπιδράσει με την εφαρμογή. Μία εφαρμογή μπορεί να αποτελείται από μία ή περισσότερες δραστηριότητες κάθε μία από τις οποίες αναλαμβάνει και ένα κομμάτι της εφαρμογής. Οποιαδήποτε εφαρμογή μπορεί να ξεκινήσει μία δραστηριότητα μίας άλλης εφαρμογής εάν αυτή της το επιτρέπει. Μπορούμε να φανταστούμε την δραστηριότητα σαν μία ιστοσελίδα και την εφαρμογή στην οποία ανήκει σαν ένα ιστόχωρο. Ο χρήστης μόλις ξεκινήσει την εφαρμογή βλέπει την αρχική δραστηριότητα της εφαρμογής και μέσω της αρχικής κάνει περιήγηση και στις άλλες δραστηριότητες της εφαρμογής. Στη συνέχεια, όπως και σε ένα φυλλομετρητή ιστού, μπορεί να πηγαίνει πίσω και εμπρός στις δραστηριότητες της εφαρμογής [32, 37, 38, 39, 40].

Υπηρεσίες (Services)

Μία υπηρεσία είναι ένα στοιχείο το οποίο τρέχει στο παρασκήνιο και δεν είναι ορατό στην οθόνη έτσι ώστε να μπορεί ο χρήστης να αλληλεπιδράσει μαζί του. Για παράδειγμα, μία υπηρεσία μπορεί να παίζει μουσική στο παρασκήνιο ενώ ο χρήστης αλληλεπιδρά με μία εφαρμογή. Ένα άλλο στοιχείο, όπως μία δραστηριότητα, μπορεί να ξεκινήσει την υπηρεσία και να την αφήσει να τρέξει [32, 37, 38, 39, 40].

Πάροχοι Περιεχομένου (Content Providers)

Διαδικτυακό Πληροφοριακό Σύστημα Διαχείρισης Έκτακτων Περιστατικών με Δυνατότητα Πρόσβασης μέσω Κινητών Συσκευών με Λειτουργικό Android

Ένας πάροχος περιεχομένου είναι ένα στοιχείο το οποίο θέτει ένα σύνολο δεδομένων μίας εφαρμογής στη διάθεση άλλων εφαρμογών. Είναι ο καλύτερος τρόπος για να μοιραστείς δεδομένα με άλλες εφαρμογές. Τα δεδομένα αποθηκεύονται στο σύστημα αρχείων ή και στην SQL βάση. Ένα παράδειγμα είναι ο content provider που παρέχει η google για τις επαφές. Όλες οι πληροφορίες που υπάρχουν εκεί (ονόματα, τηλέφωνα, διευθύνσεις κ.τ.λ.) είναι προσβάσιμες από όποια εφαρμογή θέλει να τις χρησιμοποιήσει [32, 37, 38, 39, 40].

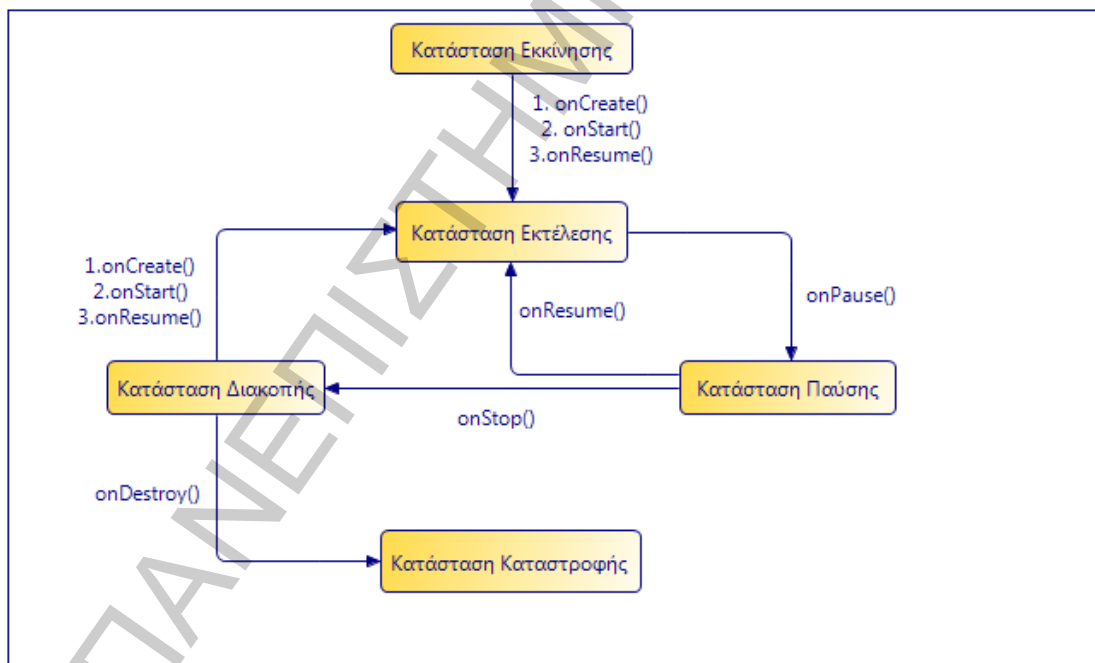
Δέκτης μηνυμάτων Broadcast (Broadcast Receiver)

Ο broadcast receiver είναι το στοιχείο εκείνο που λαμβάνει και αντιδρά στα μηνύματα broadcast (τα μηνύματα δηλαδή που στέλνονται σε όλες τις εφαρμογές). Παραδείγματα broadcast μηνυμάτων είναι μία ανακοίνωση προς όλες τις εφαρμογές ότι η ενέργεια της μπαταρίας είναι χαμηλή ή ότι κάποια δεδομένα έχουν κατέβει από το διαδίκτυο και είναι διαθέσιμα προς όλες τις εφαρμογές [37, 38, 39]

Κύκλος ζωής των δραστηριοτήτων (lifecycle of activities)

Όταν μία δραστηριότητα ξεκινάει για πρώτη φορά, έρχεται στο προσκήνιο της οθόνης έτσι ο χρήστης να μπορεί να αλληλεπιδράσει με την εφαρμογή. Κατά τη διάρκεια αυτής της διαδικασίας, το σύστημα του Android καλεί μία σειρά από callback μεθόδους στις οποίες ο προγραμματιστής μπορεί να δηλώσει πώς επιθυμεί η δραστηριότητα να συμπεριφέρεται όταν ο χρήστης φεύγει από τη συγκεκριμένη δραστηριότητα και επανέρχεται αργότερα.

Κατά τη διάρκεια της ζωής της μία δραστηριότητα μπορεί να περάσει από πολλά στάδια όπως φαίνεται από την εικόνα 6. Ο προγραμματιστής δεν έχει τον έλεγχο επιλογής κατάστασης της δραστηριότητας. Αυτό ελέγχεται από το σύστημα. Παρ' όλα αυτά ο προγραμματιστής ειδοποιείται όταν η κατάσταση πρόκειται να αλλάξει μέσω των μεθόδων callback [32, 37, 39, 41, 42].



Εικόνα 6: Κύκλος ζωής μίας δραστηριότητας [32]

Κατάσταση Εκκίνησης (Starting State)

Όταν μία δραστηριότητα δεν υπάρχει ακόμα στη μνήμη τότε βρίσκεται στην κατάσταση εκκίνησης. Μόλις ξεκινήσει, θα εκτελεστούν κάποιες μέθοδοι callback μέχρι τελικά να φτάσει στην ενεργή κατάσταση. Αυτή η διαδικασία είναι μία από τις πιο «βαριές» διαδικασίες και γι' αυτό οι δραστηριότητες δεν καταστρέφονται αμέσως μόλις φύγουν από το προσκήνιο αλλά Διαδικτυακό Πληροφοριακό Σύστημα Διαχείρισης Έκτακτων Περιστατικών με Δυνατότητα Πρόσβασης μέσω Κινητών Συσκευών με Λειτουργικό Android 30

αποθηκεύονται σε μία στοίβα από τον activity manager σε περίπτωση που ο χρήστης ξαναγυρίσει σε αυτές [39, 41].

Κατάσταση Εκτέλεσης (Running State)

Η δραστηριότητα που βρίσκεται σε κατάσταση εκτέλεσης είναι η δραστηριότητα την οποία βλέπει τη δεδομένη χρονική στιγμή ο χρήστης και με την οποία μπορεί να αλληλεπιδράσει. Μόνο μία δραστηριότητα μπορεί να βρίσκεται σε κατάσταση εκτέλεσης μία δεδομένη χρονική στιγμή. Όταν μία δραστηριότητα βρίσκεται σε αυτή την κατάσταση έχει προτεραιότητα από όλες τις άλλες στους πόρους του συστήματος έτσι ώστε να παρουσιάζεται στο χρήστη όσο το δυνατόν πιο γρήγορη [39, 41].

Κατάσταση Παύσης (Paused State)

Όταν μία δραστηριότητα δεν είναι εστιασμένη αλλά είναι ορατή στην οθόνη, τότε βρίσκεται σε κατάσταση παύσης. Αυτό συμβαίνει συνήθως με τα παράθυρα διαλόγου τα οποία εμφανίζονται μπροστά από τη δραστηριότητα και την αναγκάζουν να μπει σε κατάσταση παύσης. Παρ' όλα αυτά οι δραστηριότητες που βρίσκονται σε αυτή την κατάσταση έχουν ακόμα προτεραιότητα στους πόρους του συστήματος καθώς είναι ορατές στην οθόνη [39].

Κατάσταση Διακοπής (Stopped State)

Μία δραστηριότητα βρίσκεται σε αυτή την κατάσταση όταν δεν είναι ορατή στην οθόνη αλλά παραμένει στη μνήμη του συστήματος. Αυτές οι δραστηριότητες μπορούν είτε να επανέλθουν στην οθόνη και να γίνουν ενεργές ή να καταστραφούν και να αφαιρεθούν από τη μνήμη. Ο λόγος που αυτές οι δραστηριότητες βρίσκονται στη μνήμη είναι επειδή υπάρχει μεγάλη πιθανότητα ο χρήστης να θελήσει να ξαναγυρίσει σε αυτές τις δραστηριότητες και η διαδικασία της επαναφοράς μίας δραστηριότητας στην οθόνη της συσκευής είναι μία διαδικασία υπολογιστικά φθηνότερη από το ξεκίνημά της από την αρχή. Είναι κάτι αντίστοιχο με την cache ενός web browser η οποία κρατάει στη μνήμη του browser ιστοσελίδες έτσι ώστε να γίνεται πιο γρήγορη η περιήγηση [39].

Κατάσταση Καταστροφής (Destroyed State)

Μία δραστηριότητα σε αυτή την κατάσταση δεν βρίσκεται άλλο πια στη μνήμη του συστήματος. Ο διαχειριστής δραστηριοτήτων αποφάσισε ότι αυτή η δραστηριότητα δε χρειάζεται άλλο πια και την αφαίρεσε από τη μνήμη [39].

3.2 Μοντέλο πελάτη/εξυπηρετητή σε κινητό περιβάλλον

Ένα από τα πιο διαδεδομένα μοντέλα δικτύου που χρησιμοποιείται κατά κόρον στο χώρο του Διαδικτύου είναι το μοντέλο πελάτη - εξυπηρετητή ή αλλιώς client/server. Στο client/server μοντέλο ο υπολογιστής ενός δικτύου παίζει το ρόλο ή του client ή του server. Ο server είναι συνήθως ένας υπολογιστής ή ένα σύστημα πολλαπλών υπολογιστών με μεγάλη υπολογιστική ισχύ ο οποίος προσφέρει ένα συγκεκριμένο είδος υπηρεσιών στους clients. Ανάλογα με το είδος των υπηρεσιών που προσφέρει ένας server μπορεί να διακριθεί σε database server, email server, application server, web server κτλ.

Η πρόοδος που έχει σημειωθεί στην τεχνολογία ασύρματης δικτύωσης και των κινητών συσκευών έχει δημιουργήσει ένα νέο τομέα στο χώρο της πληροφορικής, εκείνου της κινητής πληροφορικής (mobile computing), στον οποίο οι χρήστες που χρησιμοποιούν κινητές συσκευές έχουν πρόσβαση σε διάφορες υπηρεσίες μέσα σε ένα κοινό δίκτυο, ανεξάρτητα από τη φυσική τους θέση ή τον τρόπο που κινούνται στο χώρο. Ένα τέτοιο νέο περιβάλλον εισάγει νέες τεχνικές προκλήσεις στον τομέα της πρόσβασης σε υπηρεσίες. Το μοντέλο client/server καθώς και οι εφαρμογές σε ένα κινητό περιβάλλον θα πρέπει να προσαρμοστούν στις ανάγκες που απορρέουν από την κινητικότητα των χρηστών. Πιο συγκεκριμένα, η προσαρμογή τους θα πρέπει να εστιαστεί στους περιορισμένους πόρους μίας κινητής συσκευής καθώς και στην αναξιопιστία ενός ασύρματου δικτύου σε σχέση με ένα ενσύρματο. Η Διαδικτυακό Πληροφοριακό Σύστημα Διαχείρισης Έκτακτων Περιστατικών με Δυνατότητα Πρόσβασης μέσω Κινητών Συσκευών με Λειτουργικό Android 31

αναξιοπιστία των ασύρματων δικτύων προκύπτει από το γεγονός των διακοπτόμενων συνδέσεων και του χαμηλού εύρους ζώνης. Οι περιορισμένοι πόροι μίας κινητής συσκευής, όπως η μπαταρία, η μικρή υπολογιστική ισχύς, ο μικρός αποθηκευτικός χώρος κτλ, δημιουργούν την ανάγκη στις εφαρμογές και στις αρχιτεκτονικές δικτύων να αναπτύξουν μηχανισμούς για την αντιμετώπισή τους [45].

Στην ενότητα αυτή θα παρουσιάσουμε αυτούς τους μηχανισμούς και θα δούμε ποιои από αυτούς και πώς μπορούν να υλοποιηθούν στο μοντέλο του client/server.

Η λειτουργικότητα του κώδικα μίας εφαρμογής δεν είναι απαραίτητα ομοιόμορφος σε όλη την εφαρμογή. Συνήθως ο κώδικας χωρίζεται σε τμήματα ή αλλιώς layers. Κάθε τμήμα αναλαμβάνει και μία διαφορετική λειτουργία της εφαρμογής. Γενικά χρησιμοποιούνται μέχρι 3 τμήματα εκ των οποίων ένα ασχολείται με την διεπαφή του χρήστη και την παρουσίαση της εφαρμογής στην οθόνη, ένα το οποίο περιέχει την υπολογιστική λογική της εφαρμογής και ένα το οποίο συνδέει την εφαρμογή με την βάση δεδομένων. Όταν η εφαρμογή client δεν χωρίζεται σε layers τότε λέμε ότι έχουμε thin clients ενώ αν χωρίζεται σε 1 έως 3 layers τότε λέμε ότι έχουμε fat clients. Γενικότερα όταν η υπολογιστική λογική της εφαρμογής βρίσκεται κατά κύριο λόγο στο server λέμε ότι έχουμε thin αρχιτεκτονική ενώ όταν η υπολογιστική λογική της εφαρμογής βρίσκεται στον client λέμε ότι έχουμε full αρχιτεκτονική. Και οι δύο περιπτώσεις έχουν πλεονεκτήματα και μειονεκτήματα [45, 46].

3.2.1 Λεπτή αρχιτεκτονική (thin architecture)

Η περίπτωση της thin αρχιτεκτονικής αντιμετωπίζει το πρόβλημα των περιορισμένων πόρων μίας κινητής συσκευής. Οι περιορισμοί των πόρων των κινητών συσκευών μπορεί να καταστήσουν αναγκαίο το γεγονός ορισμένες λειτουργίες οι οποίες κανονικά θα εκτελούνταν στους κινητούς clients να χρειαστεί να εκτελεστούν σε πλούσιους σε φυσικούς πόρους servers. Ένα παράδειγμα είναι τα web services, εφαρμογές/υπηρεσίες δηλαδή οι οποίες εκτελούνται σε έναν απομακρυσμένο υπολογιστή/server και στέλνουν τα αποτελέσματα πίσω στον client. Για παράδειγμα, όταν θέλουμε να βρούμε ένα σημείο που βρίσκεται στο χάρτη χρησιμοποιώντας google maps όλες οι διεργασίες της εφαρμογής εκτελούνται σε έναν server της google και στον client, σε εμάς δηλαδή, επιστρέφεται μόνο μία εικόνα που απεικονίζει ένα χάρτη και το σημείο πάνω σε αυτή. Οπότε όλη η «βαριά» δουλειά γίνεται στο server και στον client με τους περιορισμένους πόρους εκτελείται μόνο η παρουσίαση της εφαρμογής. Από την άλλη αυτή η αρχιτεκτονική έχει το μειονέκτημα ότι επειδή η περισσότερη λειτουργικότητα φιλοξενείται στο server σε διακοπτόμενες συνδέσεις, όπως στις ασύρματες, η εφαρμογή θα πρέπει να περιμένει μία επανασύνδεση με το server για να ολοκληρωθεί, κάτι αρκετά χρονοβόρο [45, 46].

3.2.2 Ολοκληρωμένη αρχιτεκτονική (full architecture)

Αντιθέτως η full αρχιτεκτονική αντιμετωπίζει ακριβώς αυτό το πρόβλημα της μικρής αξιοπιστίας των ασύρματων συνδέσεων. Οι χρήστες κινητών συσκευών θα πρέπει να είναι σε θέση να χρησιμοποιούν το ασύρματο δίκτυο με κάποια δυσάρεστα χαρακτηριστικά: διακοπτόμενη σύνδεση, χαμηλό εύρος ζώνης, υψηλή καθυστέρηση, ή ακόμα και υψηλό κόστος. Οποιαδήποτε σύνδεση με ένα από τα παραπάνω χαρακτηριστικά ονομάζεται ασθενής σύνδεση. Σε ακραίες περιπτώσεις, οι χρήστες είναι αναγκασμένοι να εργαστούν σε κατάσταση offline. Σε ορισμένες περιπτώσεις η offline σύνδεση μπορεί να είναι και πλεονέκτημα αφού μπορεί να προσφέρει χαμηλή κατανάλωση στη μπαταρία της κινητής συσκευής σε περίπτωση που ο όγκος των δεδομένων προς μετάδοση και παραλαβή είναι μεγάλος καθώς επίσης μπορεί να μειώσει και το κόστος στις περιπτώσεις που η ασύρματη σύνδεση έχει υψηλό κόστος για το χρήστη και δεν παρέχεται δωρεάν (πχ μέσω ελεύθερου wifi). Η full αρχιτεκτονική χρησιμοποιείται αποτελεσματικά, οπότε, για clients που χρησιμοποιούν δίκτυα με ασθενείς συνδέσεις ή για clients που αναγκάζονται να εργαστούν offline. Από την άλλη όμως όταν οι περισσότερες διεργασίες μίας εφαρμογής λαμβάνουν χώρα στην κινητή συσκευή επιβαρύνονται αρκετά οι μειωμένοι πόροι της συσκευής. Σε ορισμένες περιπτώσεις οι πόροι δεν είναι καν αρκετοί για να πραγματοποιηθούν δύσκολες και «βαριές» διεργασίες [45, 46].

3.2.3 Εναλλακτικές αρχιτεκτονικές

Πράκτορες (agents)

Οι πράκτορες είναι προγραμματιστικές οντότητες που βρίσκονται σε κάποιο περιβάλλον και έχουν τη δυνατότητα να πραγματοποιούν αυτόνομες ενέργειες στο περιβάλλον αυτό ώστε να επιτύχουν το στόχο που τους έχει ανατεθεί [49].

Μία καλή αρχιτεκτονική για μη αξιόπιστα ασύρματα δίκτυα θα ήταν να προσθέσουμε έναν agent στη μεριά του server. Σε αυτή την περίπτωση ο agent γίνεται ο ενδιάμεσος των κινητών συσκευών και του server. Με αυτό τον τρόπο ο server μπορεί να επικοινωνήσει με τον agent ακόμα και αν η κινητή συσκευή είναι αποσυνδεδεμένη από το δίκτυο και ο agent θα επικοινωνήσει με τον client όταν η ασύρματη σύνδεση ανακτηθεί [47, 49].

Παρόμοια με την προηγούμενη αρχιτεκτονική θα μπορούσαμε να προσθέσουμε δύο agents, έναν στη μεριά του server όπως και προηγουμένως και έναν στη μεριά του client. Ο στόχος είναι να αποκρύψουμε την πολυπλοκότητα των διεργασιών της ασύρματης επικοινωνίας στον client, όπως ακριβώς κάνει ο server agent στο server. Οι client και server πράκτορες αλληλεπιδρούν με σκοπό να μειώσουν τις ασύρματες επικοινωνίες και έχουν τη δυνατότητα να χωρίσουν τις απαιτούμενες εργασίες μεταξύ τους ανάλογα με τις εκάστοτε συνθήκες [47, 49].

Κινητοί πράκτορες (mobile agents)

Μία εναλλακτική αρχιτεκτονική η οποία αποφεύγει τις ακραίες περιπτώσεις των thin και full αρχιτεκτονικών όπου οι λογικοί υπολογισμοί της εφαρμογής γίνονται είτε στον server είτε στον client είναι αυτή των κινητών πρακτόρων ή αλλιώς mobile agents. Οι mobile agents είναι ουσιαστικά προγραμματιστικές οντότητες οι οποίες έχουν την ικανότητα να κινούνται αυτόνομα από υπολογιστή σε υπολογιστή με σκοπό να εκπληρώσουν τις ενέργειες που τους έχουν ανατεθεί. Ένας mobile agent αποφασίζει πότε και πού να κινηθεί μέσα στο δίκτυο έτσι ώστε να εκπληρώσει το σκοπό του. Έχει τη δυνατότητα να αποθηκεύει την κατάσταση εκτέλεσης του πριν από ένα «ταξίδι» και την ικανότητα να επανέρχεται στην κατάσταση πριν από το «ταξίδι» του ανακτώντας την αποθηκευμένη κατάσταση εκτέλεσής του. Οι κινητοί πράκτορες επιτρέπουν στις διεργασίες του client να εκτελεστούν όχι μόνο σε έναν κινητό κόμβο του δικτύου αλλά και σε ένα σταθερό. Με τον ίδιο τρόπο επιτρέπουν στους clients να κατεβάσουν κώδικα από τον server και να τον εκτελέσουν στην κινητή συσκευή [45, 46, 47].

Το πλεονέκτημα των mobile agents συνίσταται στη δυνατότητα του πελάτη, εφόσον δεν είναι συνδεδεμένος, να αναπτύξει έναν mobile agent με συγκεκριμένο στόχο (πχ, αίτηση για μια υπηρεσία που παρέχεται από κάποιον εξυπηρετητή), να τον στείλει στον προορισμό κατά την διάρκεια μιας συνόδου σύνδεσης και στη συνέχεια να αποσυνδεθεί. Ο πελάτης μπορεί να λάβει την απάντηση, σε μία επόμενη σύνοδο σύνδεσης. Επίσης, ο mobile agent μπορεί να αναλάβει την ανάκτηση των πληροφοριών, το φιλτράρισμά τους και την επιστροφή του αποτελέσματος στον πελάτη. Έτσι, αφενός οι πληροφορίες που μεταφέρονται μέσω του δικτύου μειώνονται, αφετέρου η κινητή συσκευή απελευθερώνεται από την διαδικασία επεξεργασίας μεγάλου όγκου δεδομένων [48].

Διακομιστής μεσολάβησης (proxy)

Μία άλλη εναλλακτική λύση είναι η χρήση ενός ενδιάμεσου υπολογιστή που θα βοηθάει στην επικοινωνία μεταξύ client και server όπως ένας agent με τη διαφορά ότι θα τρέχει σε κάποιο άλλο υπολογιστή εκτός του client και του server. Ο σκοπός, όπως και στην περίπτωση των πρακτόρων, είναι να λειτουργήσει ο proxy σαν ενδιάμεσος ανάμεσα στον client και στο server, με τα ίδια πλεονεκτήματα που μπορούν να προσφέρουν και οι πράκτορες, με τη διαφορά ότι οι εκτελέσεις επίπονων υπολογισμών θα γίνονται σε ξεχωριστό υπολογιστή για μέγιστη απόδοση της επικοινωνίας. Από τη μεριά του client, ο proxy είναι απλά ένας server με τον οποίο επικοινωνεί.

Στην περίπτωση του Διαδικτύου τον proxy τον ονομάζουμε web proxy. Ένας web proxy δίνει τη δυνατότητα στις εφαρμογές περιήγησης στο Διαδίκτυο να λειτουργούν μέσω ασύρματων

συνδέσεων χωρίς την επιβολή αλλαγών στους browsers και στους servers. Ο web proxy μπορεί να χρησιμοποιηθεί για να αποθηκεύσει τις ιστοσελίδες στην cache της κινητής συσκευής [45]. Όταν ένας χρήστης του Διαδικτύου πληκτρολογήσει ένα url στον web browser που χρησιμοποιεί ουσιαστικά στέλνει μία αίτηση στον server που φιλοξενεί το site του οποίου η διεύθυνση είναι η διεύθυνση που πληκτρολόγησε ο χρήστης. Εάν το site είναι δυναμικό τότε ο server κάνει κάποιους υπολογισμούς οι οποίοι μπορεί να είναι απλοί ή πολύπλοκοι ανάλογα με τις υπηρεσίες του site και δημιουργεί την ιστοσελίδα που απαιτεί ο χρήστης και στη συνέχεια τη στέλνει. Με τη βοήθεια του proxy οι σελίδες που επισκέπτεται πιο συχνά ο χρήστης ή σελίδες που επισκέφτηκε πιο πρόσφατα αποθηκεύονται είτε στην cache της κινητής συσκευής έτσι ώστε να μπορεί να κάνει περιήγηση ακόμα και όταν είναι αποσυνδεδεμένος είτε στην cache του proxy έτσι ώστε η περιήγηση να γίνεται πιο γρήγορα αφού δεν θα χρειαστεί να γίνουν εκτελέσεις διεργασιών για τη δημιουργία της ιστοσελίδας από την αρχή.

Επίσης, ο web proxy έχει τη δυνατότητα να συμπιέζει και να μετατρέπει τις εικόνες έτσι ώστε μικρότερος όγκος δεδομένων να μεταφέρεται μέσω ενός διαύλου μικρού εύρους ζώνης, όπως συμβαίνει με τις ασύρματες συνδέσεις. Μπορεί ακόμα να υποστηρίξει ασύγχρονες λειτουργίες περιήγησης ή λειτουργίες χωρίς πρόσβαση στο Διαδίκτυο. Αυτό μπορεί να γίνει για παράδειγμα αποθηκεύοντας τα δεδομένα που στέλνει ο server στον client ύστερα από μία αίτηση σε μία προσωρινή βάση δεδομένων του proxy και όταν αποκατασταθεί η ασύρματη σύνδεση να στέλνει τα δεδομένα πίσω στον client [45].

4. ΕΦΑΡΜΟΓΗ eEKAB

Στα προηγούμενα κεφάλαια είδαμε τις υπάρχουσες υπηρεσίες επειγόντων περιστατικών, τις τεχνολογίες που χρησιμοποιούν και επίσης παρουσιάσαμε τις τεχνολογίες που χρησιμοποιήθηκαν στην εφαρμογή που υλοποιήθηκε στην παρούσα μεταπτυχιακή διατριβή, το eEKAB. Στο κεφάλαιο αυτό θα παρουσιάσουμε την ανάλυση απαιτήσεων, θα περιγράψουμε το σχεδιασμό της εφαρμογής και τον τρόπο υλοποίησής της και, τέλος, θα κάνουμε μία πλήρη περιγραφή της λειτουργίας της τόσο στη μεριά του εξυπηρετητή όσο και σε αυτή του πελάτη.

4.1 Ανάλυση απαιτήσεων

Στην ενότητα αυτή περιγράφουμε την ανάλυση απαιτήσεων ενός πληροφοριακού συστήματος στον τομέα της προνοσοκομειακής ιατρικής φροντίδας. Πιο συγκεκριμένα παρουσιάζονται τα χαρακτηριστικά που απαιτείται να έχει το ΠΣ eEKAB που υλοποιήθηκε στην παρούσα μεταπτυχιακή διατριβή.

Γενικότερα οι απαιτήσεις είναι:

- Συντονισμένες ενέργειες μεταξύ των τμημάτων: Σε ένα σύστημα επείγουσας ιατρικής φροντίδας απαιτείται όλα τα άτομα που εμπλέκονται και έχουν διαφορετικά καθήκοντα και ρόλους κατά τις διαδικασίες προνοσοκομειακής φροντίδας να συνεργάζονται αρμονικά και να συντονίζουν τις ενέργειές τους για το καλύτερο αποτέλεσμα της προνοσοκομειακής φροντίδας. Πιο συγκεκριμένα, το τηλεφωνικό κέντρο του EKAB, η ασυρματική υπηρεσία, η ιατρική υπηρεσία καθώς και το πλήρωμα των ασθενοφόρων θα πρέπει να συντονίσουν τις ενέργειες τους με σκοπό να παρέχουν συντονισμένες ιατρικές υπηρεσίες στα επείγοντα περιστατικά.
- Αυτοματοποίηση δραστηριοτήτων και διαδικασιών
- Μηχανογραφημένη υποστήριξη διαδικασιών που εκτελούνται κατά την προνοσοκομειακή ιατρική φροντίδα
- Μείωση συνολικού χρόνου εκτέλεσης των διεργασιών των τμημάτων του EKAB μέσω των αυτοματοποιημένων διαδικασιών
- Προϋπολογισμός ΠΣ σε λογικά πλαίσια: Ο προϋπολογισμός του ΠΣ θα πρέπει να βρίσκεται μέσα σε κάποιο πλαίσιο αποδεκτό για ένα τέτοιο εγχείρημα
- Μείωση πόρων συστήματος και ανακατανομή ανθρώπινου δυναμικού: Θα πρέπει εκτός των άλλων το ΠΣ να στοχεύει και στην κατά το δυνατόν μείωση των πόρων του συστήματος και ανακατανομής του ανθρώπινου δυναμικού με σκοπό την καλύτερη λειτουργία του συστήματος.
- Εύκολο στη χρήση
- Παρέχει πλήθος λειτουργιών στο χρήστη
- Παράλληλη λειτουργία του από πολλούς χρήστες
- Ασφάλεια του δικτύου: Η ασφάλεια του δικτύου είναι πολύ σημαντική καθώς οποιοδήποτε κενό ασφαλείας μπορεί να στοιχίσει ειδικά σε έναν τόσο σοβαρό τομέα όπως αυτός της υγείας

Ειδικότερα οι απαιτήσεις ενός ΠΣ επείγουσας ιατρικής φροντίδας είναι:

- Γνώση της διαθεσιμότητας των ασθενοφόρων/γιατρών: Θα πρέπει το κέντρο ελέγχου να γνωρίζει ποιοι γιατροί και ασθενοφόρα είναι διαθέσιμα ανά πάσα στιγμή
- Γνώση της τοποθεσίας των ασθενοφόρων/γιατρών: Το κέντρο ελέγχου είναι αναγκαίο να γνωρίζει την τοποθεσία των ασθενοφόρων/γιατρών ανά πάσα στιγμή έτσι ώστε η επιλογή ασθενοφόρου και γιατρού για αποστολή σε ένα έκτακτο περιστατικό να είναι η καλύτερη δυνατή
- Οπτική απεικόνιση της διαδρομής μέχρι τον τόπο του συμβάντος σε ασθενοφόρα/γιατρούς: Με κάποιο τρόπο τα ασθενοφόρα/γιατροί χρειάζεται να διαθέτουν ένα ψηφιακό χάρτη με την τοποθεσία τους και την τοποθεσία του περιστατικού και οδηγίες για το πώς θα φτάσουν στο περιστατικό με σκοπό την γρηγορότερη άφιξη τους στον τόπο του συμβάντος
- Ειδοποίηση ασθενοφόρων/γιατρών προς το κέντρο ελέγχου για λήψη των συντεταγμένων του περιστατικού: Το κέντρο ελέγχου θα πρέπει να λαμβάνει μία

αναφορά επιβεβαίωσης από τα ασθενοφόρα/γιατρούς όταν οι τελευταίοι λάβουν τις συντεταγμένες του περιστατικού από το κέντρο ελέγχου.

- Εύκολη εξαγωγή συμπερασμάτων από το πλήθος των διαφορετικών περιστατικών: Θα πρέπει τα στοιχεία των περιστατικών να καταγράφονται με τέτοιο τρόπο ώστε να γίνονται αναλύσεις των δεδομένων εύκολα.

Αναλυτικότερα απαιτείται:

- Ύπαρξη μίας κεντρικής βάσης δεδομένων: Απαιτείται μία κεντρική βάση δεδομένων η οποία θα περιέχει όλους τους γιατρούς και τα ασθενοφόρα στην οποία θα μπορούν να συνδέονται οι τελευταίοι και θα μπορούν να την ανανεώνουν με την εκάστοτε τοποθεσία τους ασύγχρονα. Επίσης σε αυτήν θα μπορούν να καταγράφονται τα στοιχεία των περιστατικών για μετέπειτα ανάλυση και εξαγωγή συμπερασμάτων.
- Εφοδιασμός ασθενοφόρων και γιατρών με κινητές συσκευές: οι γιατροί και τα ασθενοφόρα θα πρέπει να έχουν μία κινητή συσκευή η οποία θα μπορεί να συνδέεται στην βάση και η οποία να μπορεί να εντοπίζει από μόνη της τη θέση της
- Λειτουργίες εισαγωγής, τροποποίησης και διαγραφής πάνω στη βάση: Το κέντρο ελέγχου θα πρέπει να μπορεί να εκτελεί λειτουργίες πάνω στη βάση όπως το να εισάγει, τροποποιεί ή και να διαγράφει ασθενοφόρα και γιατρούς ανάλογα με τις εκάστοτε συνθήκες
- Έλεγχος κατά τη συμπλήρωση των στοιχείων του περιστατικού
- Έλεγχος για την αν η τοποθεσία του περιστατικού υπάρχει και είναι έγκυρη

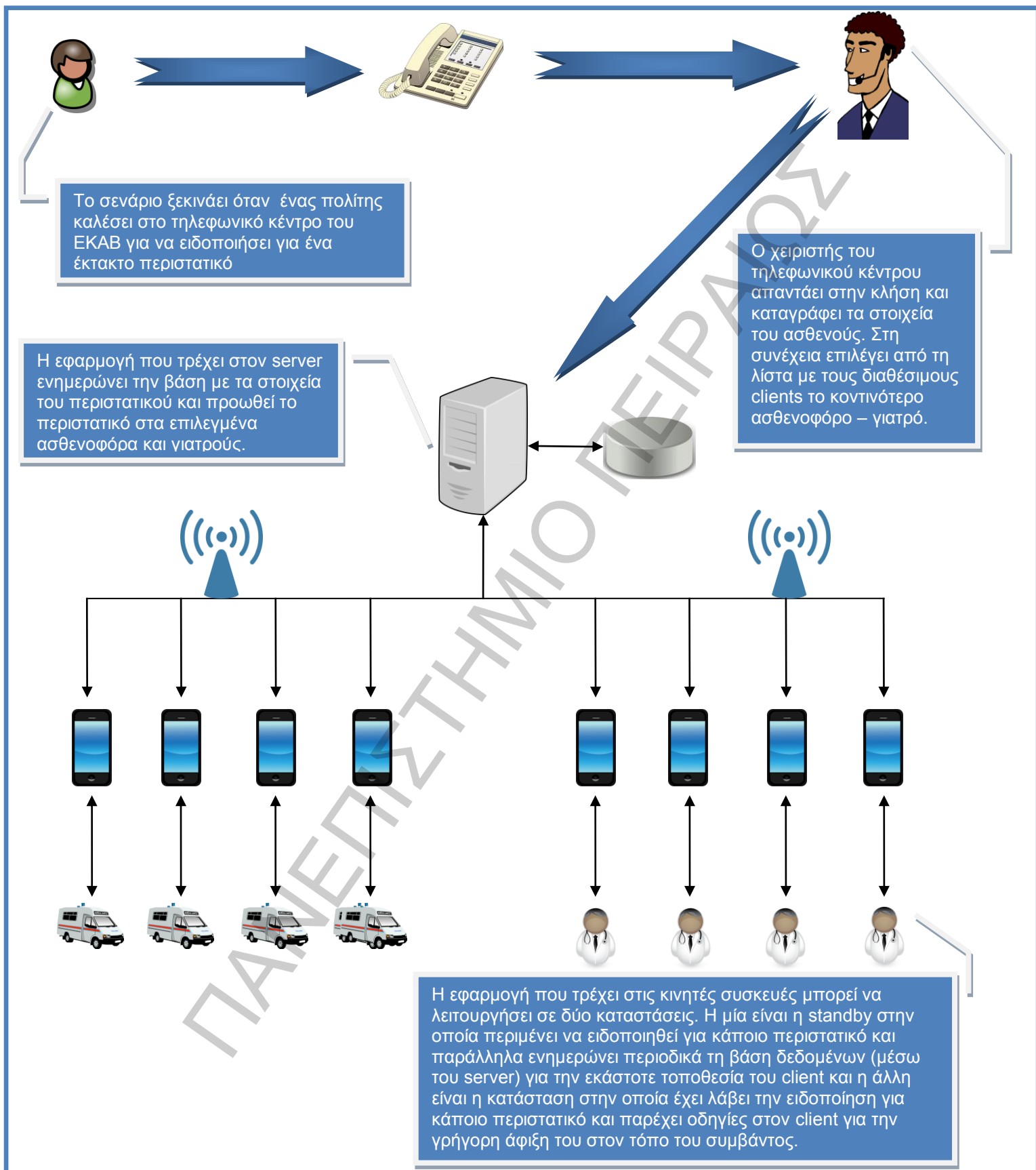
4.2 Σχεδιασμός εφαρμογής

Η εφαρμογή eEKAB αναπτύχθηκε με σκοπό να καλύψει τις περισσότερες από τις απαιτήσεις που παρουσιάστηκαν στην προηγούμενη ενότητα. Το eEKAB σχεδιάστηκε για να βοηθήσει στο συντονισμό και τη συνεργασία μεταξύ των τμημάτων του eEKAB και συγκεκριμένα του κέντρου ελέγχου με το πλήρωμα των ασθενοφόρων και των γιατρών. Ο απώτερος σκοπός της σχεδίασης του ήταν η βελτιστοποίηση των υπηρεσιών επείγουσας ιατρικής φροντίδας μέσω της αυτοματοποίησης των διαδικασιών και η μείωση του χρόνου των διαδικασιών της προνοσοκομειακής φροντίδας.

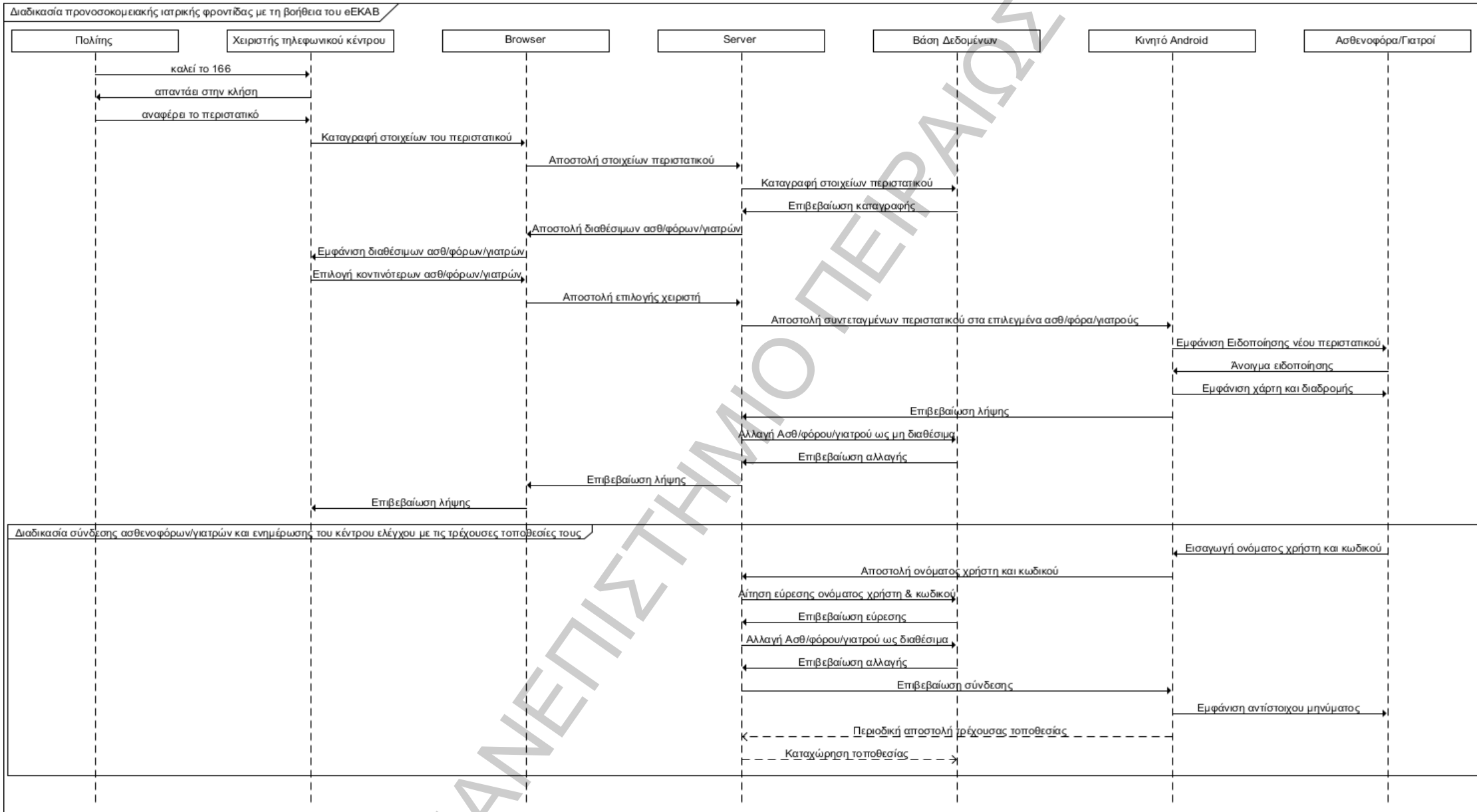
Για το σχεδιασμό της εφαρμογής χρησιμοποιήσαμε την αρχιτεκτονική client – server την οποία την προσαρμόσαμε στο κινητό περιβάλλον της προνοσοκομειακής διαδικασίας. Πιο συγκεκριμένα, το eEKAB αποτελείται από έναν web server ο οποίος είναι συνδεδεμένος με μία βάση δεδομένων και ένα πλήθος κινητών clients οι οποίοι συνδέονται με τον server και αλληλεπιδρούν με τη βάση δεδομένων. Η εφαρμογή αποτελείται από δύο επιμέρους εφαρμογές, η μία είναι αναπτυγμένη για να τρέχει στον web server και η άλλη είναι σχεδιασμένη για να τρέχει στους κινητούς clients. Στην περίπτωση του eEKAB τη server εφαρμογή την χειρίζονται οι υπεύθυνοι του κέντρου ελέγχου ενώ την κινητή εφαρμογή την χειρίζονται τα ασθενοφόρα και οι γιατροί οι οποίοι παίζουν το ρόλο των κινητών clients. Οι κινητοί clients είναι ουσιαστικά κινητά τηλέφωνα τα οποία διαθέτουν οι γιατροί και το πλήρωμα των ασθενοφόρων για να συνδέονται στον web server και να αλληλεπιδρούν με τη βάση. Συγκεκριμένα και οι δύο εφαρμογές χρησιμοποιούν δύο agents έναν στη μεριά του server και έναν στη μεριά του client οπότε η αρχιτεκτονική είναι client - agent – agent - server. Οι agents αυτοί είναι ουσιαστικά προγραμματιστικά τμήματα των εφαρμογών τα οποία αναλαμβάνουν την επικοινωνία μεταξύ server και clients ασύγχρονα έτσι ώστε το κέντρο ελέγχου να μην καθυστερεί με τις διαδικασίες που απαιτούνται για την εγκαθίδρυση των συνδέσεων μεταξύ του server και του client και την μεταξύ τους επικοινωνία.

Το σενάριο που ακολουθεί η εφαρμογή eEKAB είναι η ακόλουθη: Όταν μία κλήση για ένα περιστατικό έκτακτης ανάγκης φτάσει στο τηλεφωνικό κέντρο του EKAB τότε αυτός που βρίσκεται πίσω από το ακουστικό αρχίζει και καταγράφει τα στοιχεία του ασθενούς στην φόρμα της server εφαρμογής. Στη συνέχεια επιλέγει τον κοντινότερο γιατρό και ασθενοφόρο από τη λίστα με τους διαθέσιμους clients. Το πλήρωμα των ασθενοφόρων και οι γιατροί λαμβάνουν στην κινητή συσκευή τους ειδοποίηση ότι ένα περιστατικό είναι σε εξέλιξη. Όταν ανοίξουν την ειδοποίηση τότε αυτόματα εμφανίζεται στην οθόνη της κινητής τους συσκευής η τοποθεσία του περιστατικού και οδηγίες για το πώς θα φτάσουν ακολουθώντας τον πιο σύντομο δρόμο. Παράλληλα με το άνοιγμα της ειδοποίησης ενημερώνεται και η βάση δεδομένων με τη βοήθεια

των agents ότι το συγκεκριμένο ασθενοφόρο/γιατρός ανέλαβε το περιστατικό. Τέλος, όταν το ασθενοφόρο/γιατρός προσφέρει επιτυχώς τις υπηρεσίες του στο περιστατικό κλείνει την ειδοποίηση και ενημερώνει τη βάση ότι είναι ξανά διαθέσιμο/ς. Παράλληλα η κινητή εφαρμογή επανέρχεται σε κατάσταση standby στην οποία περιμένει για κάποια ειδοποίηση ενώ παράλληλα ενημερώνει την βάση ανά τακτά χρονικά διαστήματα για την τρέχουσα τοποθεσία της κινητής συσκευής. Ο σχεδιασμός της εφαρμογής eEKAB φαίνεται στην εικόνα 7. Στον πίνακα 3 παρουσιάζεται επίσης το διάγραμμα ακολουθίας της εφαρμογής.



Εικόνα 7: Αρχιτεκτονική της εφαρμογής eEKAB



Πίνακας 3: Διάγραμμα ακολουθίας eEKAB

4.3 Υλοποίηση εφαρμογής

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκαν σύγχρονες τεχνολογίες. Στην ενότητα αυτή περιγράφουμε πώς χρησιμοποιήθηκαν αυτές οι τεχνολογίες τόσο στη μεριά του client όσο και στη μεριά του server στην υλοποίηση της εφαρμογής eEKAB.

4.3.1 Client

Το λειτουργικό σύστημα που χρησιμοποιήθηκε στις κινητές συσκευές των clients δηλαδή των ασθενοφόρων και των ιατρών είναι το android της google. Το λειτουργικό android παρέχει πολλές δυνατότητες οι οποίες χρησιμοποιήθηκαν κατάλληλα για την υλοποίηση της εφαρμογής στους clients. Μία από τις δυνατότητές του ήταν ότι μπορεί να υποστηρίξει προγραμματισμό sockets τις οποίες χρησιμοποιήσαμε για την επικοινωνία του client με τον server. Επίσης το android είναι λειτουργικό το οποίο υποστηρίζει ενσωμάτωση των google maps μέσω του google map api για κινητά κάτι το οποίο ήταν πολύ χρήσιμο για την παρουσίαση της συντομότερης διαδρομής προς το περιστατικό στους clients. Τέλος το android δίνει τη δυνατότητα εντοπισμού της θέσης της κινητής συσκευής μέσω του καταλληλότερου location provider όπως το gps, wifi και cellID και επίσης δίνει τη δυνατότητα προγραμματισμού αισθητήρων κίνησης της συσκευής με βάση διάφορα κριτήρια. Στην εφαρμογή eEKAB χρησιμοποιώντας αυτές τις δυνατότητες του android οι clients ενημερώνουν την κεντρική βάση δεδομένων με την εκάστοτε τοποθεσία τους σε δύο περιπτώσεις: όταν ο client μετακινηθεί παραπάνω από ένα χιλιόμετρο ή όταν περάσει ένα λεπτό από την προηγούμενη ενημέρωση. Όποιο από τα δύο κριτήρια εκπληρωθεί πρώτο ενημερώνεται η κεντρική βάση με την καινούρια τοποθεσία.

Ένα άλλο εργαλείο που χρησιμοποιήθηκε είναι το google maps api για κινητά. Το api αυτό ενσωματώνεται πλήρως στο λειτουργικό android και παρέχει πολλά χρήσιμα εργαλεία τα οποία χρησιμοποιήθηκαν στην εφαρμογή eEKAB. Το api αυτό χρησιμοποιήθηκε για να υλοποιηθεί η διαδικασία παρουσίασης ενός χάρτη στους clients στον οποίο θα φαίνεται η τοποθεσία τους, η τοποθεσία του περιστατικού και η συντομότερη διαδρομή για να φτάσουν στον προορισμό τους.

4.3.2 Server

Η εφαρμογή του server υλοποιήθηκε με τη χρήση της γλώσσας προγραμματισμού php η οποία είναι αρκετά δημοφιλής για web εφαρμογές. Με την εκκίνηση της εφαρμογής εκτελείται παράλληλα και ένα άλλο πρόγραμμα που παίζει το ρόλο του agent και αναλαμβάνει το κομμάτι της επικοινωνίας με τον client με τη βοήθεια των sockets.

Επίσης χρησιμοποιήθηκε η jquery η οποία είναι μία βιβλιοθήκη της javascript για ευκολότερη ανάπτυξη της html σελίδας. Η jquery είναι υπεύθυνη για τον έλεγχο όλων των events (πχ τις ενέργειες που ακολουθούν όταν ένα κουμπί πατηθεί), τον έλεγχο όλων των φορμών για λάθος εισαγωγές δεδομένων, την παρουσίαση των αρχικών καρτελών της εφαρμογής, την παρουσίαση των παραθύρων διαλόγων καθώς επίσης και την παρουσίαση των φορμών για εισαγωγή/τροποποίηση ασθενοφόρων και γιατρών.

Επιπλέον χρησιμοποιήθηκε η τεχνική ajax η οποία συνέβαλε στην εκτέλεση κάποιων ενεργειών ασύγχρονα. Συγκεκριμένα χρησιμοποιείται στο πεδίο της διεύθυνσης του περιστατικού για να εμφανίσει μία λίστα με πιθανές διευθύνσεις (οι οποίες είναι οι έγκυρες διευθύνσεις του google map) ανάλογα με τη διεύθυνση που πληκτρολογείται, έτσι ώστε να διασφαλίζεται ότι η διεύθυνση είναι πάντα έγκυρη. Ακόμα, κατά την υποβολή της φόρμας με τα στοιχεία του περιστατικού εμφανίζεται ασύγχρονα η λίστα με τα διαθέσιμα ασθενοφόρα και γιατρούς.

Το Google Map API βοήθησε αρκετά στην υλοποίηση της εφαρμογής. Μέσω του api αυτού το κέντρο ελέγχου θα μπορεί να έχει οπτική απεικόνιση πάνω σε χάρτες της τοποθεσίας τόσο των ασθενοφόρων όσο και των γιατρών και του περιστατικού. Επίσης, με τα web service της google μπορέσαμε και υπολογίσαμε τις αποστάσεις και τον εκτιμώμενο χρόνο ενός

ασθενοφόρου ή ενός γιατρού έτσι ώστε να επιλεχθούν οι πιο κοντινοί για να αναλάβουν το περιστατικό.

Τέλος για την μορφοποίηση της εφαρμογής χρησιμοποιήθηκε CSS.

4.4 Περιγραφή της λειτουργίας του προγράμματος

4.4.1 Server

Η αρχική σελίδα της εφαρμογής e-EKAB φαίνεται στην εικόνα 8. Άμεσα παρατηρούμε ότι αποτελείται από 3 καρτέλες. Η πρώτη που αποτελεί και την αρχική σελίδα είναι η καρτέλα με τίτλο «Έκτακτα Περιστατικά». Αυτή η καρτέλα ασχολείται με την εισαγωγή ιατρικών περιστατικών. Στα αριστερά της καρτέλας αυτής παρατηρούμε τη φόρμα εισαγωγής των περιστατικών και στα δεξιά παρατηρούμε ένα χάρτη (προερχόμενο φυσικά από τους χάρτες της Google) στον οποίο θα αποτυπώνονται το περιστατικό, οι διαθέσιμοι γιατροί και τα διαθέσιμα ασθενοφόρα.

Εικόνα 8: Αρχική σελίδα της εφαρμογής

Καρτέλα έκτακτα περιστατικά

Η φόρμα διαθέτει αυτόματο έλεγχο λαθών των πεδίων ο οποίος ενεργοποιείται με δύο τρόπους. Ή πατώντας το κουμπί υποβολή έχοντας λανθασμένες τιμές στα πεδία ή προσπαθώντας να «φύγουμε» από ένα πεδίο όταν η τιμή που έχουμε εισάγει είναι λανθασμένη. Στην εικόνα 9 παρατηρούμε τα μηνύματα λάθους που εμφανίζονται όταν δεν έχουμε εισαγάγει κάποια τιμή στα πεδία ενώ στην εικόνα 10 παρατηρούμε τα μηνύματα λάθους όταν έχουμε εισαγάγει λανθασμένες τιμές.

Τα πεδία της φόρμας είναι 7 από τα οποία μόνο τα 4 είναι διαθέσιμα για εισαγωγή τιμών. Αυτά είναι τα πεδία «Όνομα Ασθενούς», «Επίθετο Ασθενούς», «Ειδικότητα Ιατρού» και «Διεύθυνση».

Στα πεδία «Όνομα Ασθενούς» και «Επίθετο Ασθενούς» εισάγουμε το όνομα και το επίθετο του ασθενούς αντίστοιχα. Για να είναι έγκυρες οι τιμές θα πρέπει οι χαρακτήρες να είναι ελληνικοί και το πρώτο γράμμα να είναι κεφαλαίο. Σε διαφορετική περίπτωση θα εμφανιστεί μήνυμα λάθους (εικόνα 10).

Κύρια Στοιχεία

Όνομα Ασθενούς
Το πεδίο είναι υποχρεωτικό

Επίθετο Ασθενούς
Το πεδίο είναι υποχρεωτικό

Ωρα κλήσης

Ειδικότητα Ιατρού
Επιλέξτε ειδικότητα ιατρού

Διεύθυνση
Συμπληρώστε μία διεύθυνση

Περιοχή Τ.Κ.

Εικόνα 9: Εμφάνιση λαθών στην περίπτωση που κάποιο πεδίο είναι κενό

Κύρια Στοιχεία

Όνομα Ασθενούς
Μη έγκυρα δεδομένα

Επίθετο Ασθενούς
Μη έγκυρα δεδομένα

Ωρα κλήσης

Ειδικότητα Ιατρού

Διεύθυνση
Η διεύθυνση δεν είναι έγκυρη

Περιοχή Τ.Κ.

Εικόνα 10: Εμφάνιση λαθών στην περίπτωση λανθασμένης εισαγωγής

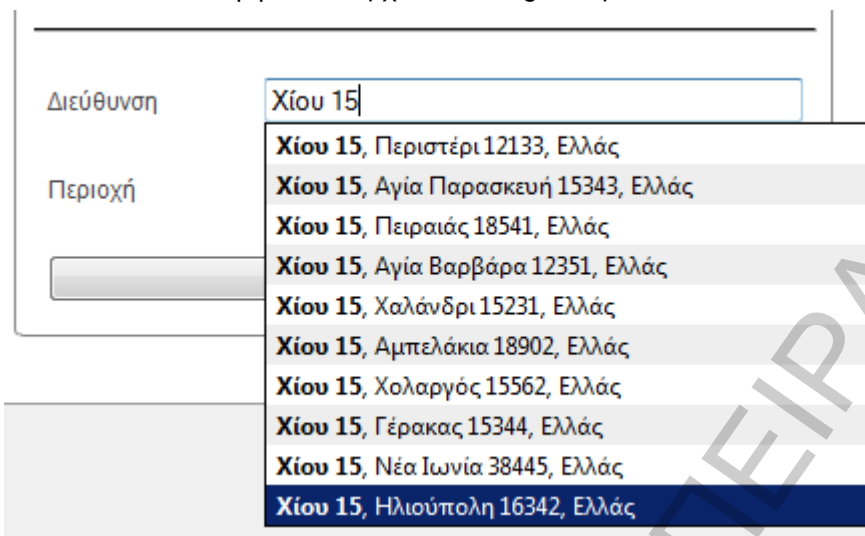
Στο πεδίο «Ειδικότητα Ιατρού» επιλέγουμε από τη λίστα μία από τις διαθέσιμες επιλογές για ειδικότητα Ιατρού (εικόνα 11). Αν δεν επιλέξουμε κάποια κατηγορία θα εμφανιστεί μήνυμα λάθους (εικόνα 9).

Ειδικότητα Ιατρού

Εικόνα 11: Η λίστα του πεδίου «Ειδικότητα Ιατρού»

Τέλος στο πεδίο «Διεύθυνση» εισάγουμε το όνομα μίας διεύθυνσης (οδός και αριθμός). Κατά τη διάρκεια της πληκτρολόγησης η εφαρμογή χρησιμοποιώντας ένα web service της Google αυτόματα μας εμφανίζει μία λίστα από τις διαθέσιμες διευθύνσεις που ταιριάζουν στους χαρακτήρες που έχουμε πληκτρολογήσει μέχρι εκείνη τη στιγμή (εικόνα 12). Σε περίπτωση που δεν εμφανιστεί η διεύθυνση που θέλουμε προσπαθούμε ξανά πληκτρολογώντας και την περιοχή π.χ. Χίου 15, Ηλιούπολη. Επιλέγουμε την διεύθυνση που επιθυμούμε από τη λίστα και

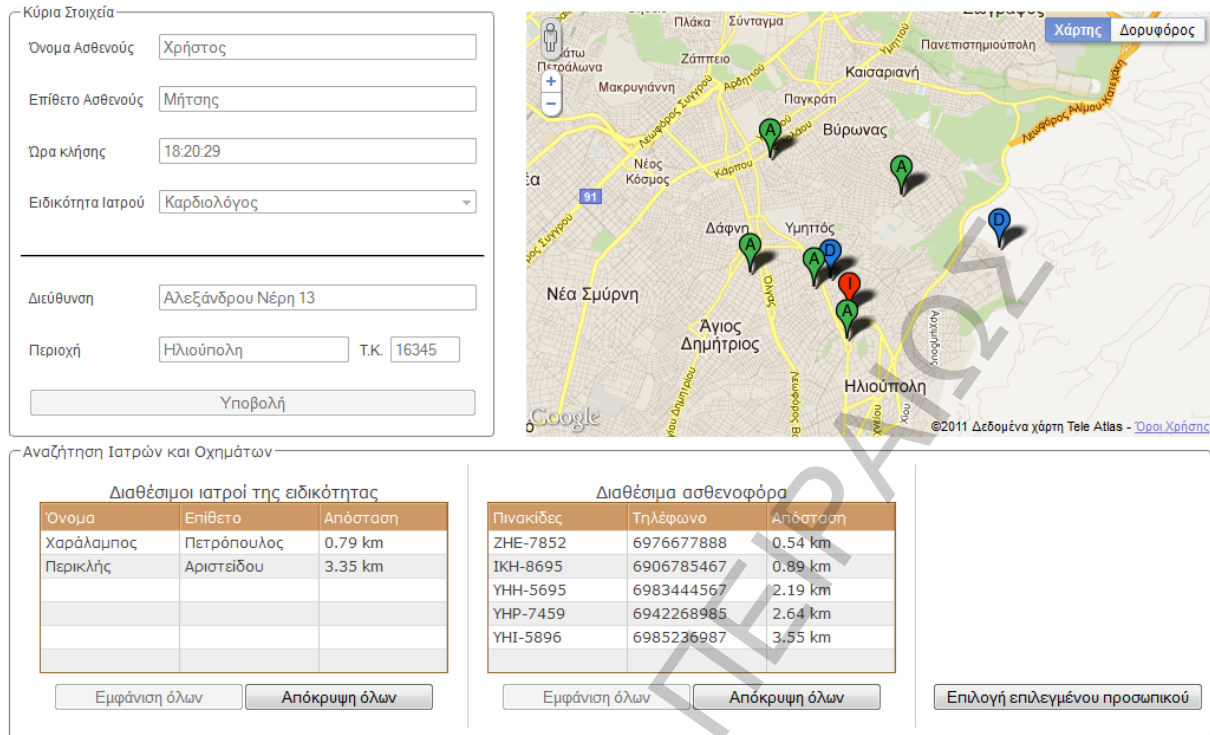
αυτόματα τα πεδία «Περιοχή» και «Τ.Κ.» συμπληρώνονται με τις τιμές της διεύθυνσης που επιλέξαμε. Επίσης στο χάρτη εμφανίζεται αυτόματα η διεύθυνση που επιλέξαμε. Σε περίπτωση που δεν επιλέξουμε από τη λίστα που εμφανίζεται αυτόματα εμφανίζεται μήνυμα λάθους όπως φαίνεται από την εικόνα 10. Με αυτό τον τρόπο αποφεύγεται η περίπτωση επιλογής διεύθυνσης που δεν είναι σωστή ή δεν υπάρχει στα Google maps.



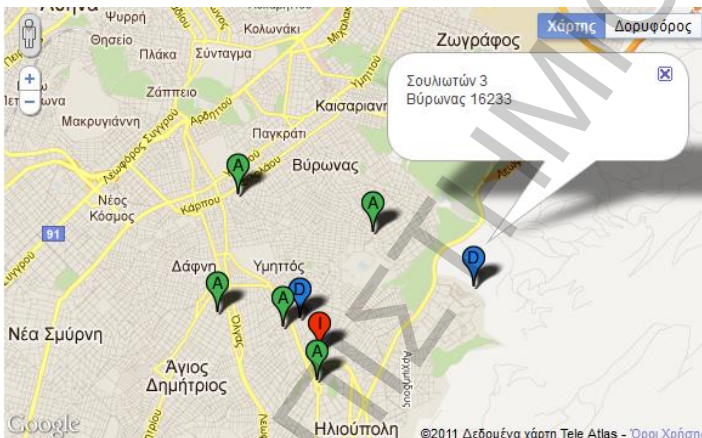
Εικόνα 12: Η αυτόματη λίστα (auto complete) του πεδίου «Διεύθυνση»

Τέλος, το πεδίο «Ωρα κλήσης» συμπληρώνεται αυτόματα με την τρέχουσα ώρα του server κάθε φορά που εισάγουμε και ένα καινούριο περιστατικό.

Μόλις εισάγουμε σωστές τιμές σε όλα τα πεδία και πατήσουμε το κουμπί «Υποβολή» αυτόματα γίνεται εισαγωγή του περιστατικού στη βάση και εμφανίζεται στην κάτω μεριά της οθόνης μία καινούρια περιοχή η οποία ονομάζεται «Αναζήτηση Ιατρών και Οχημάτων» (εικόνα 13). Παράλληλα απενεργοποιείται η περιοχή «Κύρια Στοιχεία» προς αποφυγή αλλαγών. Στον χάρτη εμφανίζονται, επίσης, όλοι οι διαθέσιμοι γιατροί της ειδικότητας που επιλέξαμε και όλα τα διαθέσιμα ασθενοφόρα εκείνη τη στιγμή. Το περιστατικό εμφανίζεται με κόκκινο χρώμα και το γράμμα I (Incident), οι γιατροί με μπλε χρώμα και το γράμμα D (Doctor) και τα ασθενοφόρα με πράσινο χρώμα και το γράμμα A (Ambulance). Εάν πατήσουμε πάνω σε οποιοδήποτε γιατρό - ασθενοφόρο πάνω στο χάρτη μας δίνεται η δυνατότητα να δούμε την ακριβή διεύθυνση που βρίσκεται ο αντίστοιχος γιατρός – ασθενοφόρο (εικόνα 14).



Εικόνα 13: Εμφάνιση της περιοχής «Αναζήτηση Ιατρών και Οχημάτων» μετά την υποβολή του περιστατικού

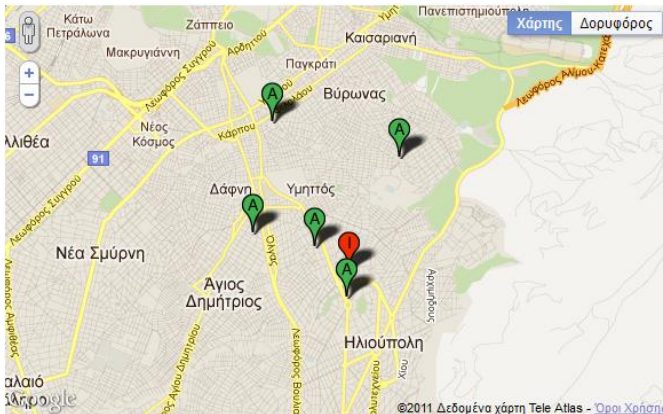


Εικόνα 14: Εμφάνιση της ακριβούς διεύθυνσης που βρίσκεται εκείνη τη στιγμή αν επιλέξουμε σε κάποιον ιατρό ή ασθενοφόρο

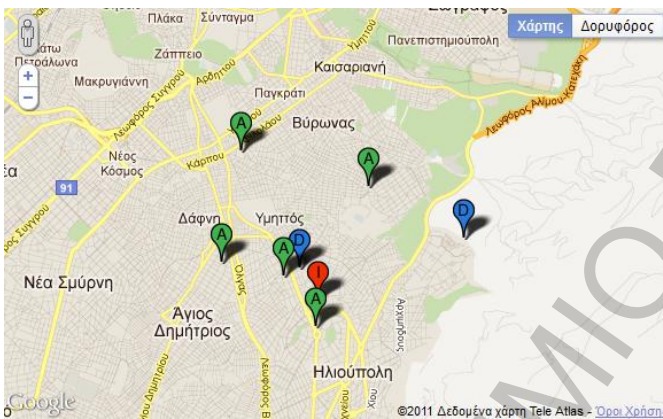
Η καινούρια περιοχή χωρίζεται σε τρεις υποπεριοχές.

Η πρώτη περιλαμβάνει τον πίνακα «Διαθέσιμοι ιατροί της ειδικότητας» όπου και εμφανίζονται οι διαθέσιμοι γιατροί της ειδικότητας εκείνη τη στιγμή. Συγκεκριμένα εμφανίζεται το όνομα του ιατρού, το επίθετο του καθώς και η χιλιομετρική απόσταση (κατά προσέγγιση) του ιατρού από το περιστατικό. Οι πίνακες είναι ταξινομημένοι με βάση την απόσταση έτσι ώστε να δίνεται προτεραιότητα στους γιατρούς που βρίσκονται πιο κοντά στο περιστατικό. Καθώς περνάμε τον δείκτη του ποντικιού πάνω από κάθε γιατρό η καταχώριση του συγκεκριμένου γιατρού τόσο στον πίνακα όσο και στο χάρτη αλλάζει χρώμα (πορτοκαλί και κίτρινο χρώμα αντίστοιχα) για να είναι ευδιάκριτο ποια καταχώριση θα επιλέξουμε πριν την επιλέξουμε καθώς και για να είναι ευδιάκριτη η αντιστοιχία μεταξύ πίνακα και χάρτη. Επίσης είναι διαθέσιμες δύο επιλογές: η «Απόκρυψη όλων» και η «Εμφάνιση όλων». Η πρώτη «κρύβει» τους γιατρούς από το χάρτη έτσι ώστε να μπορεί ο χρήστης να παρατηρήσει το χάρτη μόνο με το περιστατικό και τα διαθέσιμα ασθενοφόρα (εικόνα 15). Η επιλογή «Εμφάνιση όλων» ενεργοποιείται μόνο όταν

έχει επιλεγθεί η επιλογή «Απόκρυψη όλων» και το μόνο που κάνει είναι να εμφανίζει ξανά στο χάρτη τους γιατρούς (εικόνα 16).



Εικόνα 15: Ο χάρτης μετά την απόκρυψη των γιατρών

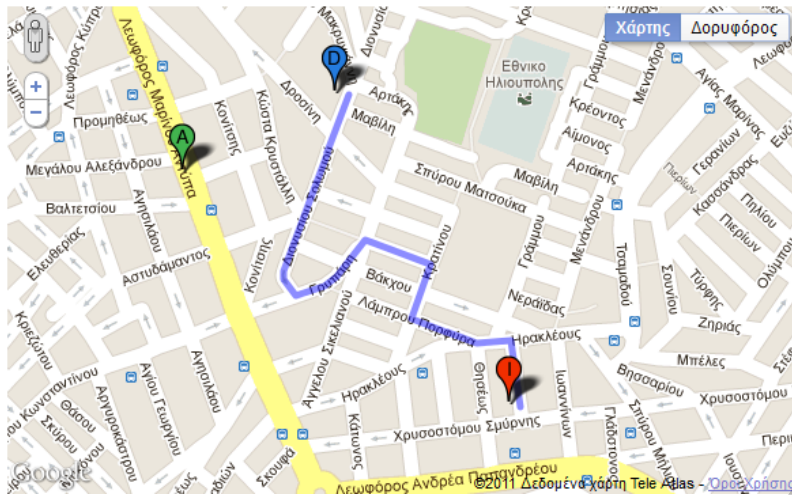


Εικόνα 16: Ο χάρτης μετά την εμφάνιση των γιατρών

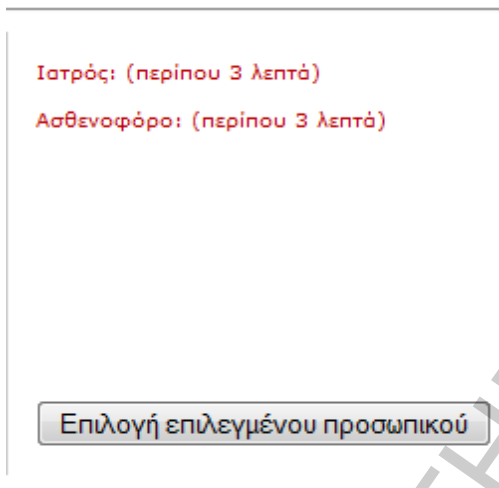
Η δεύτερη υποπεριοχή περιλαμβάνει τον πίνακα «Διαθέσιμα ασθενοφόρα» όπου και εμφανίζονται τα διαθέσιμα ασθενοφόρα εκείνη τη στιγμή. Ισχύει ό,τι και για την πρώτη περιοχή με τη διαφορά ότι το όνομα και το επίθετο έχουν αντικατασταθεί με τη στήλη «Πινακίδες» και τη στήλη «Τηλέφωνο» όπου και εμφανίζονται η πινακίδα του ασθενοφόρου και το τηλέφωνο του οδηγού.

Στην τρίτη υποπεριοχή βρίσκουμε την επιλογή «Επιλογή επιλεγμένου προσωπικού» και μία κενή περιοχή ακριβώς από πάνω από την επιλογή αυτή. Με την επιλογή αυτή στέλνουμε στο γιατρό και στο ασθενοφόρο, που προηγουμένως επιλέξαμε από τους αντίστοιχους πίνακες «Διαθέσιμοι Ιατροί της Ειδικότητας» και «Διαθέσιμα Ασθενοφόρα», τις συντεταγμένες του περιστατικού έτσι ώστε να μεταβούν έγκαιρα στο σημείο. Παράλληλα η εφαρμογή επανέρχεται στην αρχική σελίδα (εικόνα 8) για την εισαγωγή καινούριου περιστατικού. Η κενή περιοχή χρησιμοποιεί για την εμφάνιση μηνυμάτων που θα αναφερθούν στη συνέχεια.

Τι γίνεται όμως όταν επιλέγουμε έναν γιατρό ή ένα ασθενοφόρο από τους αντίστοιχους πίνακες; Καταρχάς η καταχώριση χρωματίζεται για να γνωρίζουμε την επιλογή μας. Παράλληλα εμφανίζεται στον χάρτη η αντίστοιχη συντομότερη διαδρομή από το γιατρό ή το ασθενοφόρο μέχρι το περιστατικό (εικόνα 17). Τέλος εμφανίζεται στην κενή περιοχή πάνω από την επιλογή «Επιλογή επιλεγμένου Προσωπικού» ο εκτιμώμενος χρόνος της διαδρομής με βάση τους χάρτες της Google (εικόνα 18).



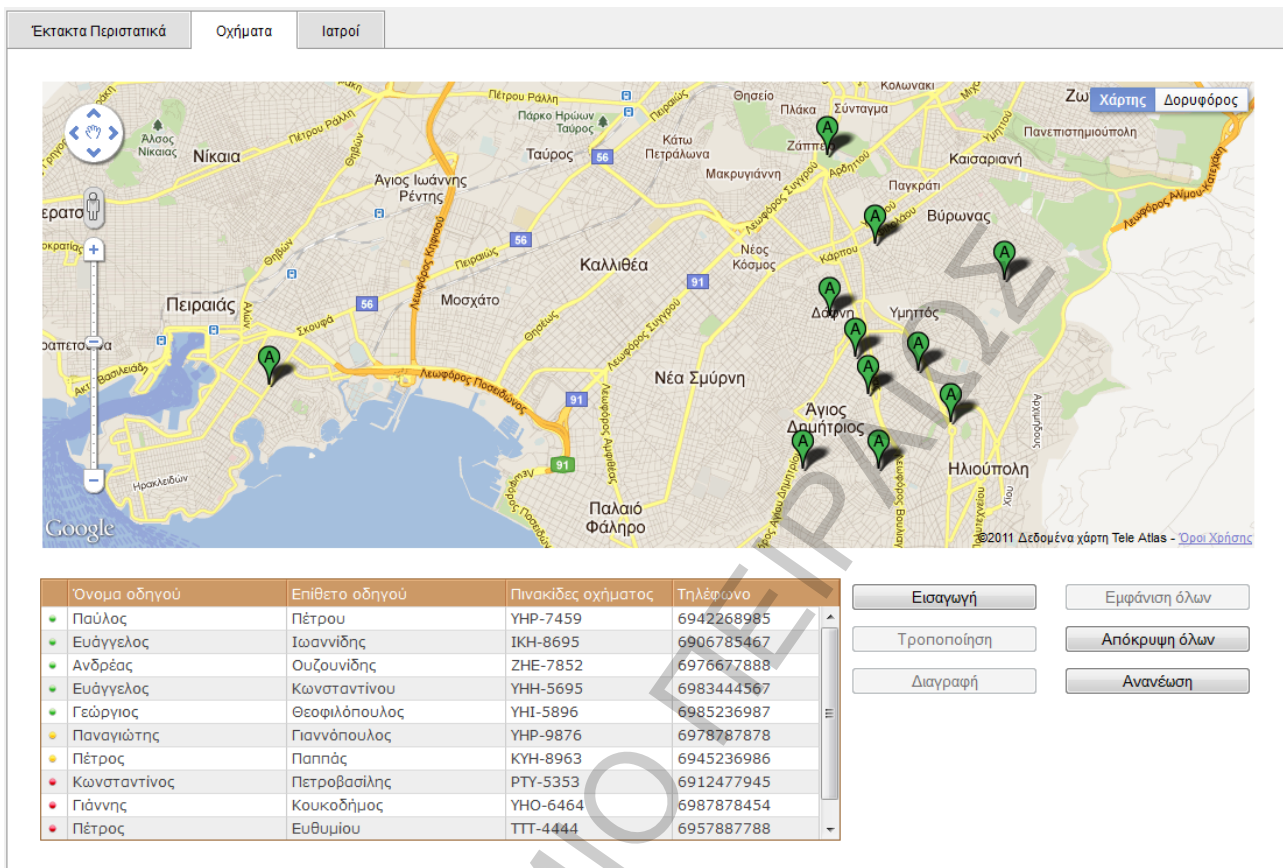
Εικόνα 17: Συντομότερη διαδρομή από έναν γιατρό έως το περιστατικό



Εικόνα 18: Εμφάνιση εκτιμώμενου χρόνου της διαδρομής

Καρτέλα Οχήματα

Η αρχική σελίδα της καρτέλας «Οχήματα» φαίνεται στην εικόνα 19. Όπως παρατηρούμε η καρτέλα αυτή αποτελείται από ένα χάρτη στον οποίο μπορούμε να παρατηρήσουμε τη θέση όλων των ασθενοφόρων διαθέσιμων και μη (όλα τα ασθενοφόρα της βάσης), από ένα πίνακα στον οποίο είναι καταχωρημένα όλα τα ασθενοφόρα, από 6 κουμπιά με διαθέσιμες επιλογές και μία κενή περιοχή κάτω από αυτά για εμφάνιση μηνυμάτων που θα περιγράψουμε παρακάτω πιο αναλυτικά.

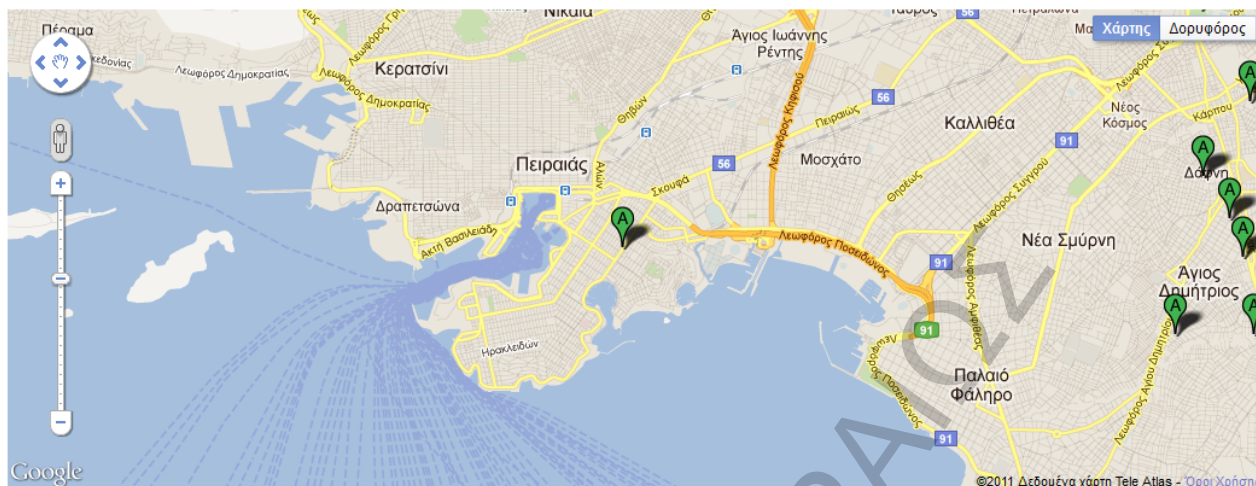


Εικόνα 19: Αρχική σελίδα καρτέλας «Οχήματα»

Ο πίνακας αποτελείται από 5 στήλες.

Η πρώτη στήλη μπορεί να πάρει 3 τιμές. Η πράσινη ένδειξη μας ενημερώνει πως ο οδηγός του συγκεκριμένου ασθενοφόρου έχει συνδεθεί στην εφαρμογή e-EKAB μέσω του android κινητού του και έχει προλάβει να ανανεώσει τη θέση του οπότε και είναι διαθέσιμος για κάποιο περιστατικό. Η κίτρινη ένδειξη μας ενημερώνει ότι ο συγκεκριμένος οδηγός έχει αναλάβει ήδη ένα περιστατικό και οπότε δεν είναι διαθέσιμος. Η κόκκινη ένδειξη, τέλος, μας ενημερώνει ότι ο οδηγός δεν έχει συνδεθεί από το κινητό του στην εφαρμογή e-EKAB οπότε και δεν είναι διαθέσιμος. Η δεύτερη στήλη αφορά το όνομα του οδηγού του ασθενοφόρου, η τρίτη στήλη το επίθετο του, η τέταρτη τις πινακίδες του ασθενοφόρου και η πέμπτη το τηλέφωνο του οδηγού.

Όταν επιλέξουμε μία εγγραφή του πίνακα αυτή χρωματίζεται, ο χάρτης κεντράρεται με βάση τη θέση του ασθενοφόρου που επιλέξαμε και στην κενή περιοχή κάτω από τα διαθέσιμα κουμπιά εμφανίζεται με κόκκινα γράμματα η διεύθυνση στην οποία βρίσκεται εκείνη τη στιγμή το συγκεκριμένο ασθενοφόρο (εικόνα 20).



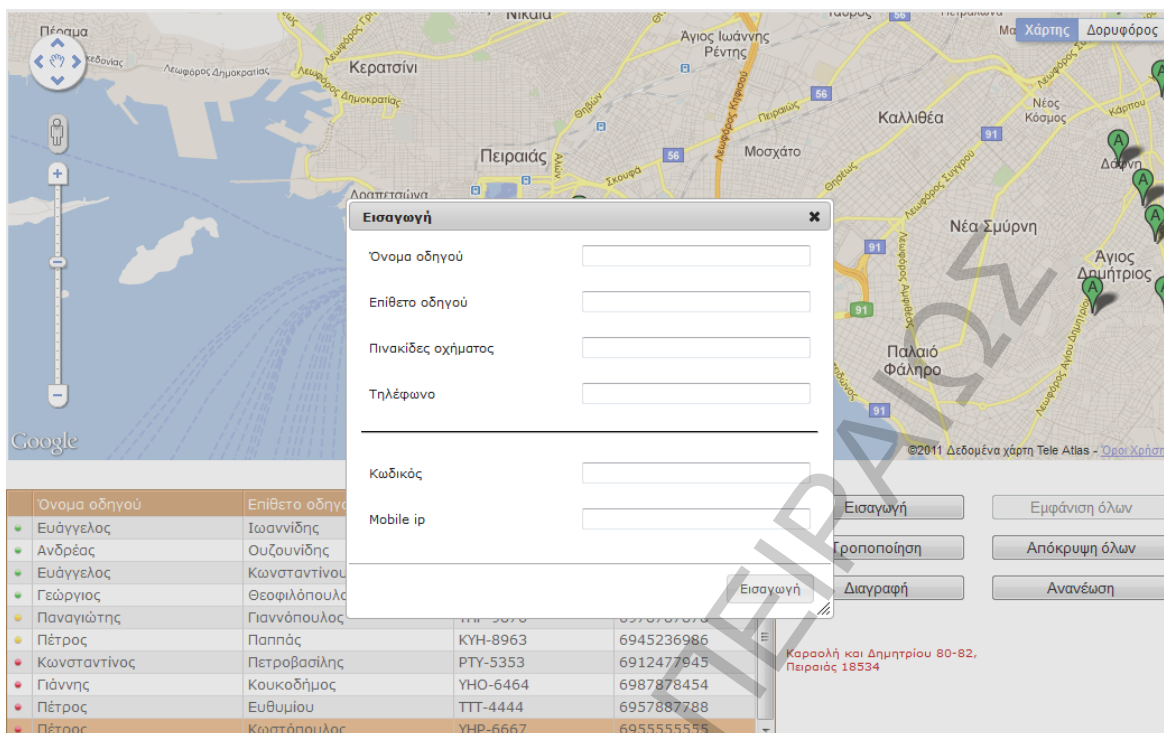
Όνομα οδηγού	Επίθετο οδηγού	Πινακίδες οχήματος	Τηλέφωνο
• Ευάγγελος	Ιωαννίδης	IKH-8695	6906785467
• Ανδρέας	Ουζουνίδης	ZHE-7852	6976677888
• Ευάγγελος	Κωνσταντίνου	YHH-5695	6983444567
• Γεώργιος	Θεοφιλόπουλος	YHI-5896	6985236987
• Παναγιώτης	Γιαννόπουλος	YHP-9876	6978787878
• Πέτρος	Παππάς	KYH-8963	6945236986
• Κωνσταντίνος	Πετροβασιλης	PTY-5353	6912477945
• Γιάννης	Κουκοδήμος	YHO-6464	6987878454
• Πέτρος	Ευθυμίου	TTT-4444	6957887788
• Πέτρος	Κωστόπουλος	YHP-6667	6955555555

Καρσολή και Δημητρίου 80-82,
Πειραιάς 18534

Εικόνα 20: Επιλογή μίας εγγραφής στον πίνακα της καρτέλας «Οχήματα»

Οι διαθέσιμες επιλογές είναι 6 και είναι η «Εισαγωγή», η «Τροποποίηση», η «Διαγραφή», η «Εμφάνιση όλων», η «Απόκρυψη όλων» καθώς και η «Ανανέωση». Αρχικά ενεργοποιημένες είναι μόνο η «Εισαγωγή», η «Απόκρυψη όλων» και η «Ανανέωση». Ας δούμε όμως τι μας προσφέρει η κάθε μία από αυτές τις επιλογές.

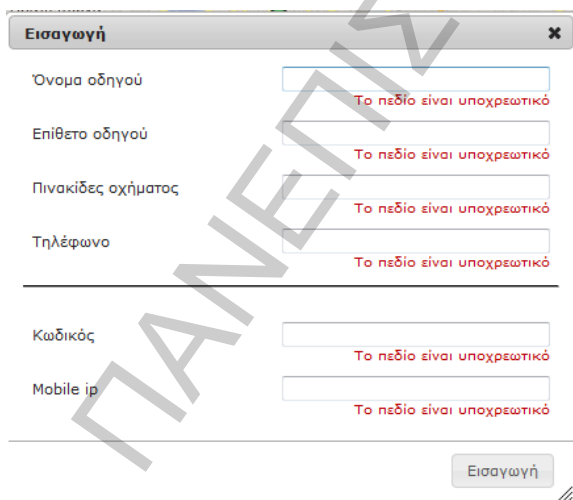
Η «Εισαγωγή» μας επιτρέπει να εισάγουμε ένα καινούριο ασθενοφόρο – οδηγό. Αυτό επιτυγχάνεται με μία φόρμα που εμφανίζεται όταν πατηθεί το κουμπί «Εισαγωγή» (εικόνα 21). Η συγκεκριμένη φόρμα έχει τη δυνατότητα αλλαγής μεγέθους (κάτω δεξιά) και τη δυνατότητα κλεισίματος όπως ένα κανονικό παράθυρο (κουμπί κλεισίματος πάνω δεξιά). Αποτελείται από 6 πεδία. Τα πεδία Όνομα οδηγού και Επίθετο οδηγού στα οποία πρέπει να εισάγουμε το όνομα και το επίθετο του οδηγού αντίστοιχα (πρώτο γράμμα κεφαλαίο και τα υπόλοιπα να αποτελούνται μόνο από ελληνικούς χαρακτήρες), τις πινακίδες οχήματος για τις πινακίδες του ασθενοφόρου (με ελληνικούς κεφαλαίους χαρακτήρες τα πρώτα 3 γράμματα στη συνέχεια μία παύλα και τέλος 4 αριθμοί), το τηλέφωνο για το τηλέφωνο του οδηγού ασθενοφόρου (πρέπει να ξεκινάει από 69). Στο πεδίο κωδικός εισάγουμε έναν μοναδικό κωδικό για το ασθενοφόρο ο οποίος χρησιμεύει στον οδηγό για να κάνει login μέσω του κινητού του και να συνδεθεί στην εφαρμογή e-EKAB (τουλάχιστον 6 χαρακτήρες είτε λατινικοί χαρακτήρες είτε αριθμοί). Στο πεδίο mobile ip εισάγουμε την mobile ip του κινητού του οδηγού (π.χ. 192.168.2.8).



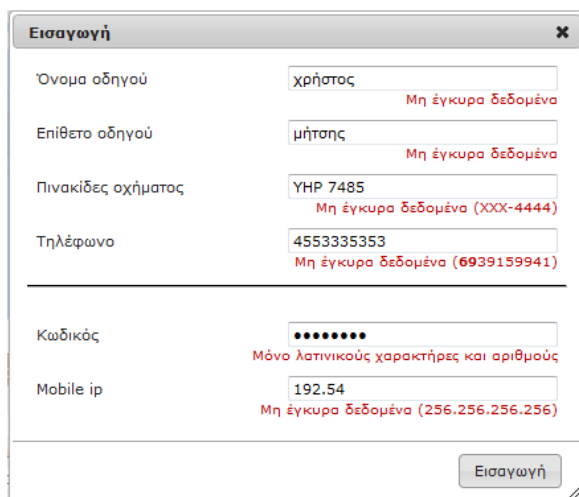
Εικόνα 21: Η φόρμα της εισαγωγής ασθενοφόρου – οδηγού

Και αυτή η φόρμα διαθέτει αυτόματο έλεγχο λαθών όπως ακριβώς και η φόρμα της εισαγωγής περιστατικών. Όταν έχουμε αφήσει κενό κάποιο πεδίο λαμβάνουμε ένα μήνυμα σαν αυτό της εικόνας 22 ενώ αν έχουμε εισάγει λάθος τιμή σε κάποιο από τα πεδία λαμβάνουμε μηνύματα αντίστοιχα με αυτά που φαίνονται στην εικόνα 23.

Μόλις έχουμε εισάγει σωστές τιμές σε όλα τα πεδία πατάμε εισαγωγή. Ένα μήνυμα μας ενημερώνει ότι η εισαγωγή ολοκληρώθηκε (εικόνα 24) και στον πίνακα εμφανίζεται αυτόματα η καινούρια εγγραφή με κόκκινη ένδειξη. Μόλις ο συγκεκριμένος οδηγός συνδεθεί μέσω του κινητού του και στείλει τη θέση του στην εφαρμογή τότε με την επιλογή «Ανανέωση» μπορούμε να δούμε τη θέση του στο χάρτη και την καινούρια του κατάσταση (πράσινη ένδειξη) στον πίνακα.



Εικόνα 22: Μηνύματα λάθους στην περίπτωση κενού πεδίου

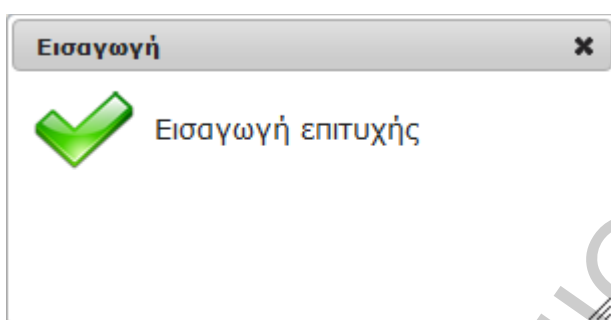


The screenshot shows a registration form with the following fields and error messages:


Field	Value	Error Message
Όνομα οδηγού	χρήστος	Μη έγκυρα δεδομένα
Επίθετο οδηγού	μήτσος	Μη έγκυρα δεδομένα
Πινακίδες οχήματος	ΥΗΡ 7485	Μη έγκυρα δεδομένα (XXX-4444)
Τηλέφωνο	4553335353	Μη έγκυρα δεδομένα (6939159941)
Κωδικός	Μόνο λατινικούς χαρακτήρες και αριθμούς
Mobile ip	192.54	Μη έγκυρα δεδομένα (256.256.256.256)

At the bottom right of the form is a button labeled "Εισαγωγή".

Εικόνα 23: Μηνύματα λάθους στην περίπτωση λανθασμένων τιμών στα πεδία

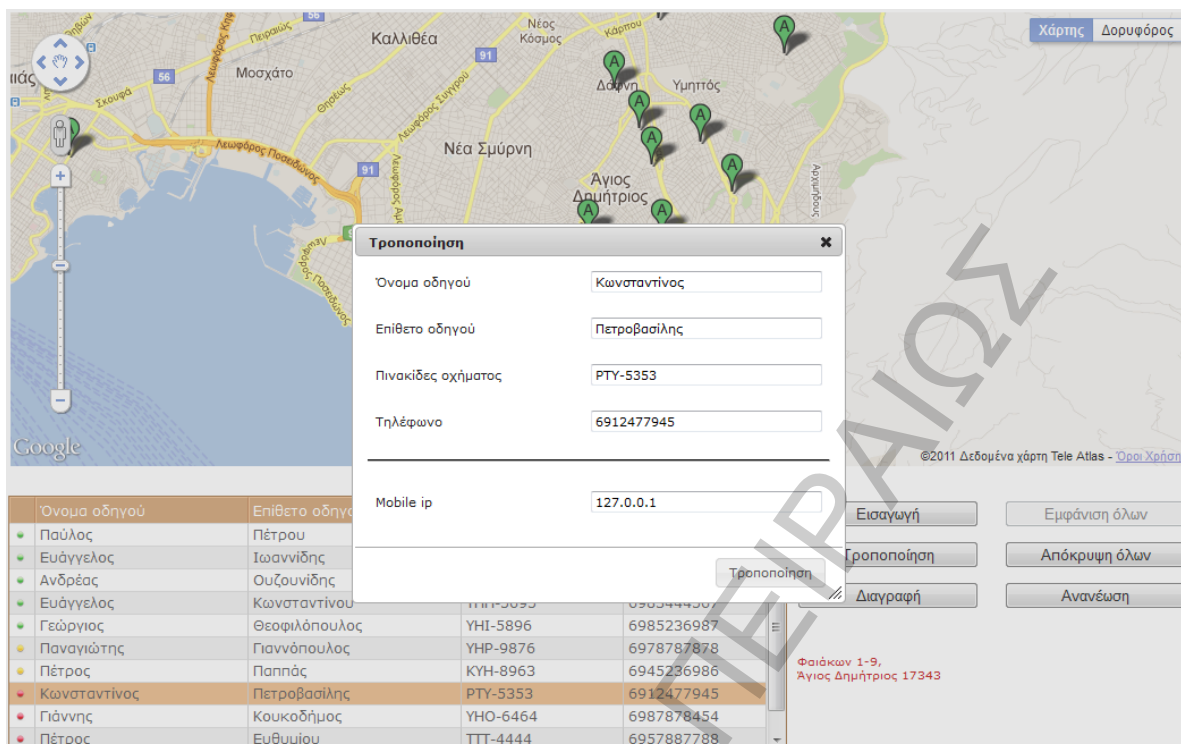


The screenshot shows a registration form with a success message:

 Εισαγωγή επιτυχής

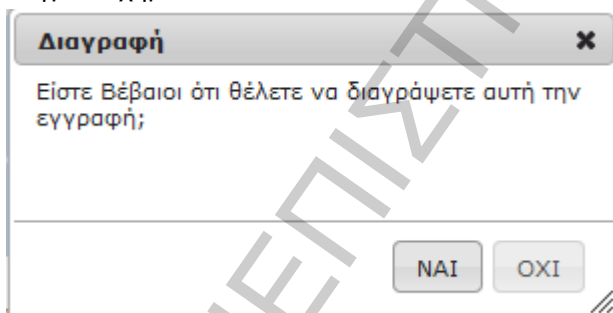
Εικόνα 24: Μήνυμα επιβεβαίωσης ότι η εισαγωγή ήταν επιτυχής

Η επιλογή «Τροποποίηση» είναι αρχικά απενεργοποιημένη. Ενεργοποιείται όταν επιλέξουμε μια εγγραφή από τον πίνακα των Οχημάτων. Απαραίτητη προϋπόθεση είναι η εγγραφή της οποίας τα στοιχεία θέλουμε να τροποποιήσουμε να μην έχει πράσινη ή κίτρινη ένδειξη παρά μόνο κόκκινη ένδειξη. Αυτό γίνεται για μεγαλύτερη ασφάλεια και αποφυγή προβλημάτων που μπορεί να προκύψουν αν τροποποιήσουμε τα στοιχεία ενός οδηγού που είναι συνδεδεμένος εκείνη τη στιγμή στο κεντρικό σύστημα. Αφού λοιπόν επιλέξουμε μία εγγραφή με κόκκινη ένδειξη τότε πατάμε το κουμπί της τροποποίησης και αυτόματα εμφανίζεται η φόρμα της τροποποίησης (εικόνα 25). Η φόρμα είναι ίδια με τη φόρμα της εισαγωγής με τη διαφορά ότι λείπει το πεδίο κωδικός για μεγαλύτερη ασφάλεια καθώς και ότι όλα τα πεδία είναι συμπληρωμένα με τα στοιχεία της εγγραφής για μεγαλύτερη διευκόλυνση στην τροποποίηση τους. Έχει επίσης έλεγχο λαθών ίδιο με τη φόρμα εισαγωγής. Ένα μήνυμα αντίστοιχο της εικόνας 24 μας ειδοποιεί για την επιτυχία της τροποποίησης, μόλις πατήσουμε το κουμπί «τροποποίηση» της φόρμας.



Εικόνα 25: Η φόρμα της τροποποίησης ασθενοφόρου – οδηγού

Η επιλογή «Διαγραφή» είναι και αυτή απενεργοποιημένη και ενεργοποιείται ακριβώς όπως και η επιλογή της τροποποίησης. Επιλέγουμε μια εγγραφή με κόκκινη ένδειξη και πατάμε «Διαγραφή». Αυτόματα εμφανίζεται ένα παράθυρο διαλόγου που ζητάει επιβεβαίωση για τη διαγραφή της εγγραφής που επιλέξαμε (εικόνα 26). Αυτό συμβαίνει για μεγαλύτερη ασφάλεια. Εάν επιλέξουμε «ΝΑΙ» τότε εμφανίζεται ένα μήνυμα αντίστοιχο της εικόνας 24 που μας ειδοποιεί για την επιτυχία της διαγραφής και αυτόματα η εγγραφή διαγράφεται από τον πίνακα των οδηγών-οχημάτων.



Εικόνα 26: Παράθυρο διαλόγου για επιβεβαίωση διαγραφής

Οι επιλογές «Εμφάνιση Όλων» και «Απόκρυψη Όλων» χρησιμεύουν για την εμφάνιση και την απόκρυψη των οχημάτων στο χάρτη.

Η επιλογή «Ανανέωση» υπάρχει έτσι ώστε να μας δείχνει κάθε χρονική στιγμή που πατάμε το κουμπί αυτό, την τρέχουσα κατάσταση των ασθενοφόρων-οδηγών.

Καρτέλα Ιατροί

Η καρτέλα «Ιατροί» είναι ίδια ακριβώς με την καρτέλα «Οχήματα». Υπάρχει μόνο μια διαφορά στην φόρμα εισαγωγής και τροποποίησης. Αντί για το πεδίο «Πινακίδες» υπάρχει το πεδίο «Τομέας». Στο πεδίο αυτό χρησιμοποιείται η τεχνική του auto complete κατευθείαν από τη βάση. Αυτό σημαίνει ότι καθώς ο χρήστης πληκτρολογεί μία ειδικότητα γιατρού στο πεδίο

«Τομέας» αυτόματα εμφανίζεται μία λίστα με τις διαθέσιμες ειδικότητες που υπάρχουν στη βάση και οι οποίες ταιριάζουν με αυτό που πληκτρολογεί ο χρήστης (εικόνα 27).

Εικόνα 27: Φόρμα εισαγωγής ιατρού

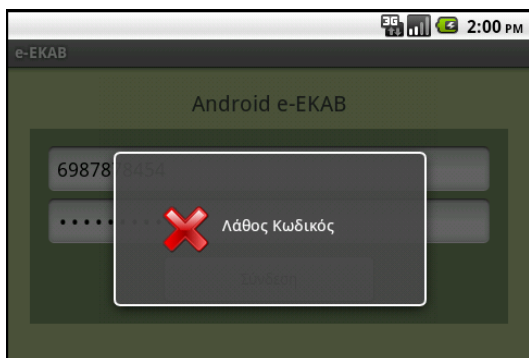
4.4.2 Client

Η κύρια οθόνη της εφαρμογής e-EKAB στο κινητό είναι αυτή που φαίνεται στην εικόνα 28.

Εικόνα 28: Κύρια οθόνη e-EKAB στο κινητό

Αποτελείται ουσιαστικά από 2 πεδία: στο πρώτο πεδίο εισάγουμε το νούμερο της συσκευής στην οποία τρέχει το e-EKAB (μπορεί να είναι και ένα username που θα καθοριστεί από τους διαχειριστές του συστήματος) και στο δεύτερο έναν κωδικό που έχει δοθεί στον κάτοχο του κινητού και μόνο. Μόλις συμπληρωθούν αυτά τα πεδία πατάμε το κουμπί σύνδεση για να συνδεθούμε. Σε περίπτωση που ο αριθμός ή ο κωδικός που εισάγουμε δεν είναι σωστά ή δεν ταιριάζουν τότε εμφανίζεται κάποιο μήνυμα λάθους όπως φαίνεται και στις εικόνες 29 και 30.

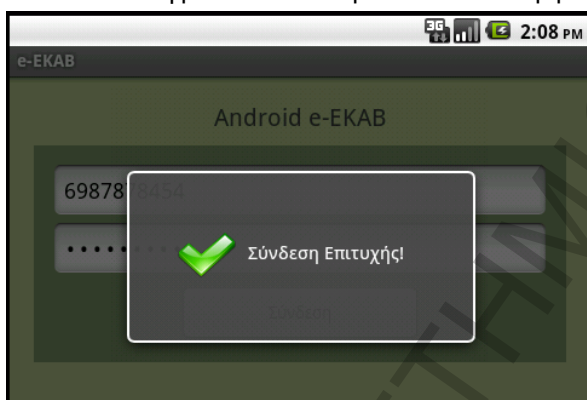
Εικόνα 29: Μήνυμα λάθους όταν ο αριθμός δεν είναι σωστός



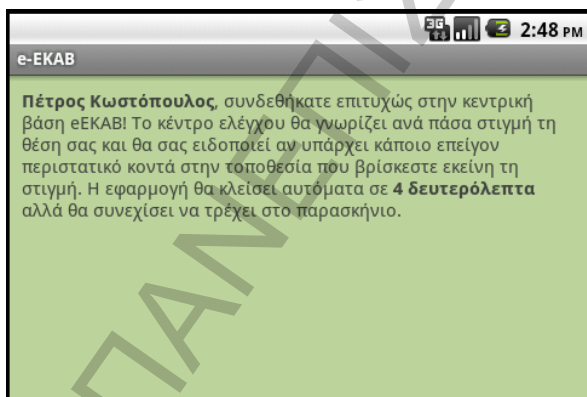
Εικόνα 30: Μήνυμα λάθους όταν ο κωδικός δεν είναι σωστός

Στην περίπτωση σωστής εισαγωγής των πεδίων εμφανίζεται το μήνυμα «Σύνδεση Επιτυχής» (εικόνα 31), στη συνέχεια εμφανίζεται ένα ειδοποιητικό μήνυμα (εικόνα 32) το οποίο κλείνει αυτόματα σε 20 δευτερόλεπτα, που αναφέρει το όνομα του χρήστη και τον ενημερώνει για 4 πράγματα:

- το κινητό συνδέθηκε με την κεντρική εφαρμογή στο server
- η εφαρμογή θα στέλνει ανά τακτά χρονικά διαστήματα την θέση του κινητού στην κεντρική εφαρμογή e-EKAB ενημερώνοντας τη κεντρική βάση.
- η εφαρμογή τρέχει πλέον στο παρασκήνιο περιμένοντας να λάβει κάποιο περιστατικό
- εάν ληφθεί κάποιο περιστατικό τότε εμφανίζεται ειδοποίηση στο κινητό



Εικόνα 31: Μήνυμα επιβεβαίωσης σύνδεσης



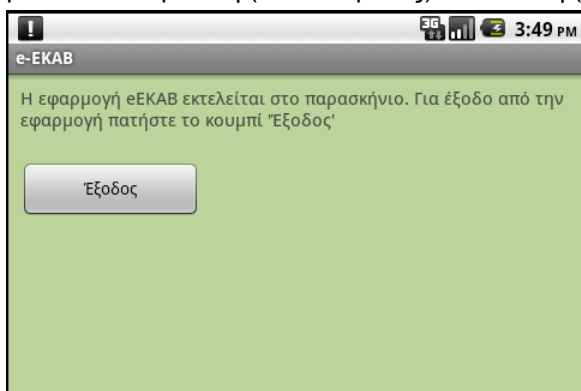
Εικόνα 32: Ειδοποιητικό μήνυμα για τις λειτουργίες της εφαρμογής

Μόλις κλείσει το ειδοποιητικό μήνυμα εμφανίζεται ένα εικονίδιο ειδοποίησης (θαυμαστικό) στην πάνω αριστερή πλευρά του κινητού (εικόνα 33) που μας πληροφορεί για την εκτέλεση της εφαρμογής στο παρασκήνιο.



Εικόνα 33: Εικονίδιο στην πάνω αριστερή μεριά του κινητού που μας πληροφορεί ότι η εφαρμογή τρέχει στο παρασκήνιο

Για να κλείσουμε την εφαρμογή ανοίγουμε την ειδοποίηση και στην οθόνη που εμφανίζεται (εικόνα 34) πατάμε το κουμπί «Έξοδος». Αυτόματα ενημερώνεται η κατάσταση του κινητού στην κεντρική βάση στο server και η ένδειξη στον πίνακα των οχημάτων ή των γιατρών γίνεται από πράσινη (συνδεδεμένος) σε κόκκινη (αποσυνδεδεμένος).



Εικόνα 34: Έξοδος από την εφαρμογή

Η ένδειξη στον πίνακα των οχημάτων ή των γιατρών της κεντρικής εφαρμογής δε γίνεται από κόκκινη σε πράσινη μόλις συνδεθεί το αντίστοιχο κινητό αλλά όταν το κινητό στείλει την τοποθεσία του για πρώτη φορά στην κεντρική βάση. Αυτό πραγματοποιείται γιατί θα ήταν επικίνδυνο για τον ασθενή να επιλεγεί κάποιος γιατρός ή ασθενοφόρο για να μεταβεί στον τόπο του περιστατικού του οποίου η τοποθεσία δεν θα είχε ανανεωθεί. Κι αυτό γιατί π.χ. όταν ο οδηγός ενός ασθενοφόρου αποσυνδεθεί με το τέλος της βάρδιας του από την εφαρμογή μπορεί να απέχει 5 χλμ από το περιστατικό και όταν ξανασυνδεθεί την επόμενη μέρα να απέχει 30 χλμ κάτι το οποίο μπορεί να αποβεί μοιραίο για τον ασθενή.

Η ανανέωση της τοποθεσίας του εκάστοτε οχήματος ή γιατρού στην κεντρική βάση γίνεται κάθε λεπτό ή όταν το όχημα ή ο γιατρός μετακινηθούν πάνω από 1 χιλιόμετρο από την προηγούμενη τοποθεσία.

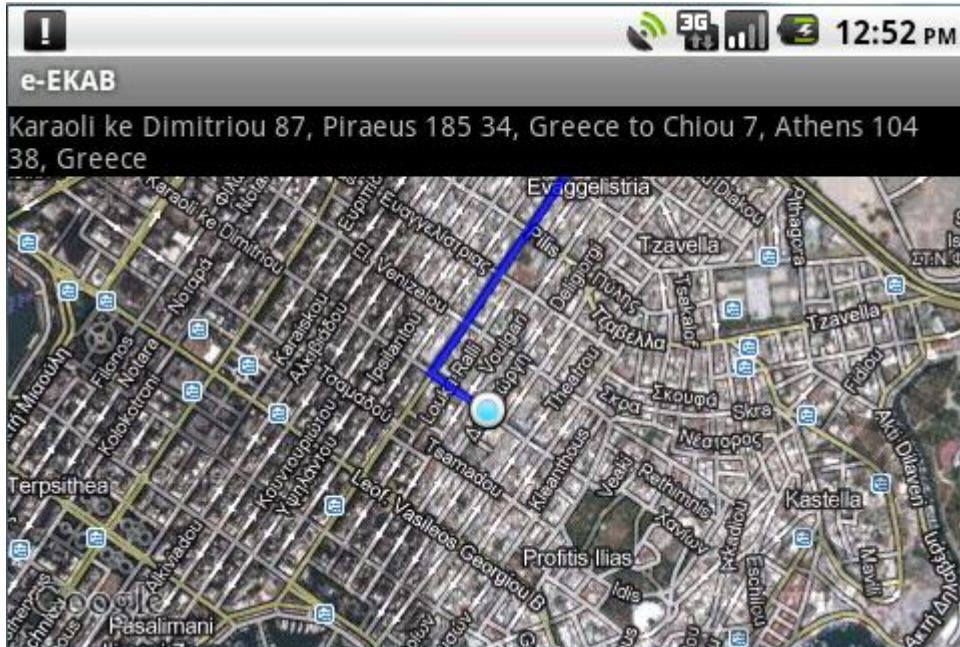
Σε περίπτωση που το κινητό δεχτεί κάποιο περιστατικό τότε μία καινούρια ειδοποίηση στο κινητό μας ενημερώνει ότι ένα καινούριο περιστατικό ελήφθη (εικόνα 35).



Εικόνα 35: Ειδοποίηση άφιξης καινούριου περιστατικού

Αυτόματα η ένδειξη στον πίνακα των οχημάτων της κεντρικής εφαρμογής γίνεται κίτρινη και σταματάει και η ανανέωση της τοποθεσίας του κινητού προς την κεντρική βάση δεδομένων.

Μόλις ανοίξουμε την ειδοποίηση εμφανίζεται στην οθόνη ένας χάρτης (Google map) με την τρέχουσα θέση του ιατρού ή του ασθενοφόρου εκείνη τη στιγμή (γαλάζια κουκίδα) καθώς και η διαδρομή προς το περιστατικό όπως φαίνεται από την εικόνα 36.

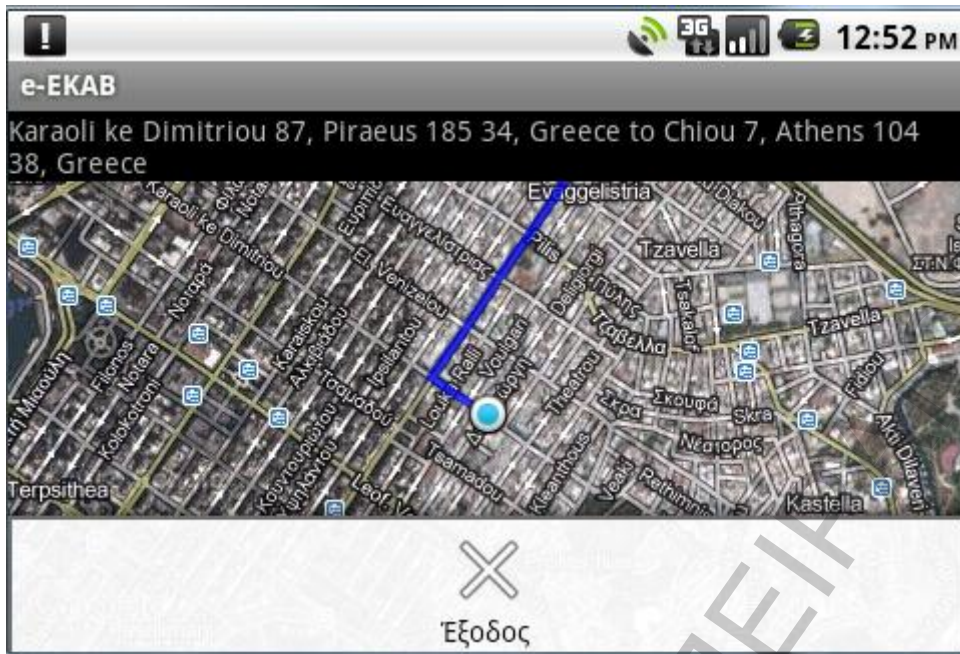


Εικόνα 36: Διαδρομή από το σημείο που βρίσκεται το κινητό εκείνη τη στιγμή (γαλάζια κουκίδα) μέχρι το περιστατικό

Επίσης παρατηρούμε ότι μας παρέχονται και κάποιες άλλες πληροφορίες όπως είναι η διεύθυνση εκκίνησης και προορισμού, η χιλιομετρική απόσταση της διαδρομής και ο μέσος χρόνος που θα κάνει ο γιατρός ή το ασθενοφόρο από το σημείο εκκίνησης μέχρι το περιστατικό. Η θέση του κινητού και κατ' επέκταση ο χάρτης θα ανανεώνεται ανά 3 δευτερόλεπτα ή ανά 100 μέτρα διανυόμενης απόστασης. Θα λειτουργεί δηλαδή σαν ένα gprs navigator.

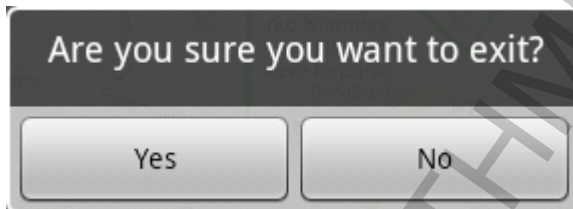
Σε περίπτωση που για οποιοδήποτε λόγο κλείσουμε την εφαρμογή σε αυτό το σημείο την επαναφέρουμε εύκολα από την ειδοποίηση την οποία βρίσκουμε από το θαυμαστικό που υπάρχει στην πάνω αριστερή μεριά της οθόνης του κινητού μας και η οποία δεν διαγράφεται κατά την ανάληψη του περιστατικού.

Μόλις ο κάτοχος του κινητού ή ο γιατρός ή το ασθενοφόρο φτάσουν στο περιστατικό το αναλάβουν και τελειώσουν την εργασία που έχουν να κάνουν πρέπει να κλείσουν την κατάσταση της εφαρμογής έτσι ώστε να καταλάβει η κεντρική εφαρμογή ότι είναι και πάλι διαθέσιμοι για καινούριο περιστατικό. Αυτό επιτυγχάνεται με το πάτημα του κουμπιού μενού του κινητού (εικόνα 37). Στη συνέχεια πατάμε το κουμπί εξόδου που εμφανίζεται.



Εικόνα 37: Επιλογή του μενού για έξοδο

Εμφανίζεται ένα παράθυρο διαλόγου επιβεβαίωσης της εξόδου όπως φαίνεται στην εικόνα 38. Επιλέγουμε Yes και η εφαρμογή επανέρχεται στην προηγούμενη κατάσταση της περιμένοντας πάλι για κάποιο περιστατικό και ανανεώνοντας την θέση του κινητού στην κεντρική εφαρμογή όπως πριν.



Εικόνα 38: Παράθυρο διαλόγου επιβεβαίωσης εξόδου από την εφαρμογή

5. ΑΞΙΟΛΟΓΗΣΗ ΕΦΑΡΜΟΓΗΣ eEKAB

5.1 Περιγραφή μοντέλου αξιολόγησης

Η αξιολόγηση της επιτυχίας των Πληροφοριακών Συστημάτων έχει αναγνωριστεί ως ένα από τα πιο κρίσιμα ζητήματα του τομέα των ΠΣ. Αρκετές εννοιολογικές και εμπειρικές μελέτες έχουν διεξαχθεί με σκοπό να εξερευνήσουν αυτό το συγκεκριμένο πλην όμως σημαντικό ζήτημα. Υπάρχουν πολλά διαφορετικά συστήματα αξιολόγησης και στρατηγικές που χρησιμοποιούνται σήμερα. Η αξιολόγηση με βάση τους στόχους, η ερμηνευτική αξιολόγηση καθώς και η αξιολόγηση με βάση καθορισμένα κριτήρια είναι από τις πιο διαδεδομένες προσεγγίσεις αξιολόγησης καθώς δίνουν τη δυνατότητα καθορισμού κριτηρίων αξιολόγησης και επίσης λαμβάνουν βασικά αποτελέσματα [50, 51]. Σε αυτήν την ενότητα θα περιγράψουμε περιληπτικά την προσέγγιση αξιολόγησης criteria - driven την οποία θα χρησιμοποιήσουμε και στη συνέχεια θα την εφαρμόσουμε για να αξιολογήσουμε το πληροφοριακό μας σύστημα, το eEKAB.

Η προσέγγιση goal-driven όπως παρουσιάζεται στο [51] στηρίζεται στο βαθμό επίτευξης των αρχικών στόχων που είχαν τεθεί πριν την υλοποίηση του πληροφοριακού συστήματος. Οι στόχοι είναι αυτοί που ορίζουν τι θα αξιολογηθεί. Μία άλλη προσέγγιση είναι αυτή που δεν χρησιμοποιεί τους αρχικούς στόχους για να αξιολογήσει το ΠΣ. Αυτή η προσέγγιση είναι πιο περιγραφική, πιο ερμηνευτική. Τέλος, υπάρχει και η προσέγγιση criteria-driven η οποία χρησιμοποιεί κριτήρια για την αξιολόγηση ΠΣ. Όταν χρησιμοποιούνται κριτήρια αυτό σημαίνει ότι η εστίαση πέφτει στον καθορισμό συγκεκριμένων ιδιοτήτων οι οποίες είναι σημαντικές για την αξιολόγηση του ΠΣ.

Επιπρόσθετα, υπάρχουν δύο γενικοί τρόποι αξιολόγησης. Ο ένας είναι να αξιολογήσουμε το ΠΣ όπως είναι, δηλαδή χωρίς τη παρέμβαση των χρηστών και ο άλλος τρόπος είναι να το αξιολογήσουμε κατά τη χρήση, δηλαδή να γίνεται ανάλυση του συστήματος μαζί με τον χρήστη. Στην παρούσα αξιολόγηση θα υιοθετήσουμε τον πρώτο τρόπο καθώς δεν υπήρχαν διαθέσιμοι χρήστες για να συνεισφέρουν στην αξιολόγηση.

Συμπερασματικά η τελική στρατηγική που θα ακολουθήσουμε θα είναι η αξιολόγηση με βάση καθορισμένα κριτήρια χωρίς την παρέμβαση χρηστών. Τα κριτήρια θα χωριστούν σε κατηγορίες και θα δοθούν συγκεκριμένα βάρη σε κάθε κατηγορία και κάθε κριτήριο. Στη

$$Q = \sum_{j=1}^n w_j \sum_{i=1}^{kj} s_{ji} w_{ji} \quad \text{όπου } w_j$$

συνέχεια θα εφαρμόσουμε τον τύπο που παρουσιάζεται στο [51]

το βάρος της κατηγορίας j , s_{ji} η τιμή του κριτηρίου i της κατηγορίας j , w_{ji} το βάρος του κριτηρίου i της κατηγορίας j , n ο αριθμός των κατηγοριών και τέλος kj ο αριθμός των κριτηρίων αξιολόγησης της κατηγορίας j . Οι επιτρεπόμενες τιμές των κριτηρίων θα λαμβάνονται από μία κλίμακα με «άριστα» το 4:

- 0 – δεν πληροί τις προϋποθέσεις
- 1 – Δεν πληροί εξ' ολοκλήρου τις προϋποθέσεις
- 2 – Πληροί τις προϋποθέσεις
- 3 – υπερβαίνει ελαφρώς τις απαιτήσεις
- 4 – υπερβαίνει αισθητά τις απαιτήσεις

Αυτός ο τύπος στο τέλος θα μας δώσει την ποιοτική απόδοση του ΠΣ. Ουσιαστικά θα μας δείξει σε ποιο βαθμό πετύχαμε την εκπλήρωση των απαιτήσεων που είχαμε ορίσει πριν την υλοποίηση του eEKAB. Να τονίσουμε εδώ ότι η τιμή 0 δεν θεωρείται αποδεκτή καθώς εάν ένα σημαντικό κριτήριο δεν πληροί τις προϋποθέσεις τότε όλος ο σχεδιασμός έχει πέσει έξω. Επίσης, τα βάρη των κατηγοριών και των κριτηρίων θα λαμβάνονται από μία κλίμακα με μεγαλύτερη σημαντικότητα το 3:

- 1 – σημαντικό
- 2 – πολύ σημαντικό
- 3 – πάρα πολύ σημαντικό

Οι κατηγορίες αξιολόγησης με τα αντίστοιχα κριτήρια τους παρουσιάζονται παρακάτω.

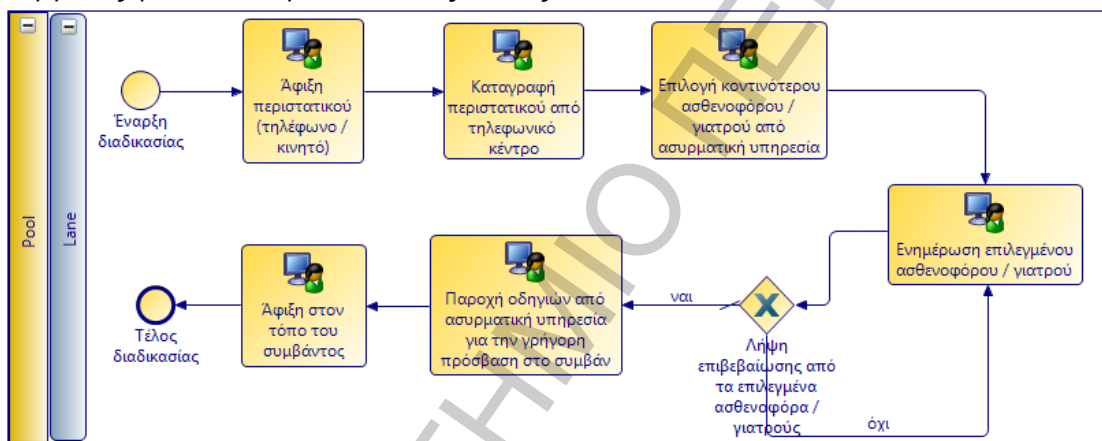
5.2 Κριτήρια αξιολόγησης

5.2.1 Απόδοση

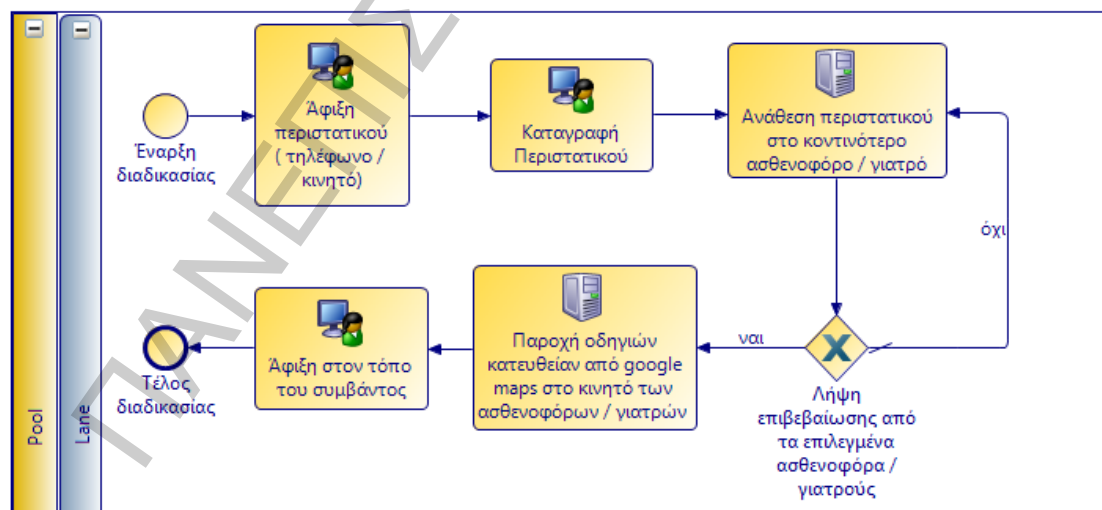
Στην κατηγορία αυτή συμπεριλαμβάνονται τα κριτήρια της μείωσης χρόνου, κόστους και κατανάλωσης πόρων καθώς και της εύκολης εξαγωγής συμπερασμάτων από τα διάφορα περιστατικά και θεωρείται ως η πιο σημαντική της αξιολόγησης μας γι' αυτό και της δίνουμε τιμή βάρους 3. Στη συνέχεια παρουσιάζονται αναλυτικά τα κριτήρια αυτά.

Μείωση χρόνου

Αυτό το κριτήριο προσδιορίζει ουσιαστικά το βαθμό στον οποίο μειώθηκε ο συνολικός χρόνος εκτέλεσης των διαδικασιών του ΕΚΑΒ και κατ' επέκταση εκφράζει τη μείωση του χρόνου από τη στιγμή που κάποιος τηλεφωνεί στο τηλεφωνικό κέντρο του ΕΚΑΒ για να αναφέρει ένα περιστατικό μέχρι τη στιγμή της άφιξης του ασθενοφόρου στον τόπο του συμβάντος. Για να αναλύσουμε σε ποιο βαθμό το πετύχαμε αυτό θα πρέπει να ελέγξουμε τις διεργασίες που εκτελούνται σήμερα στο ΕΚΑΒ και τις διεργασίες με την ενσωμάτωση του ΠΣ eΕΚΑΒ. Οι διεργασίες φαίνονται παραστατικά στις εικόνες 39 και 40.



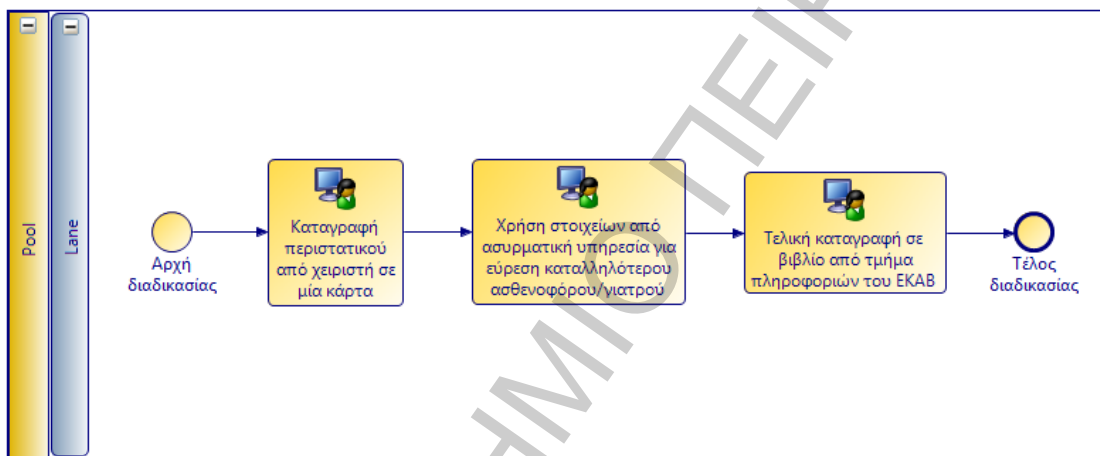
Εικόνα 39: Διεργασίες προνοσοκομειακής ιατρικής φροντίδας πριν την ενσωμάτωση του ΠΣ eΕΚΑΒ



Εικόνα 40: Διεργασίες προνοσοκομειακής ιατρικής φροντίδας μετά την ενσωμάτωση του ΠΣ eΕΚΑΒ

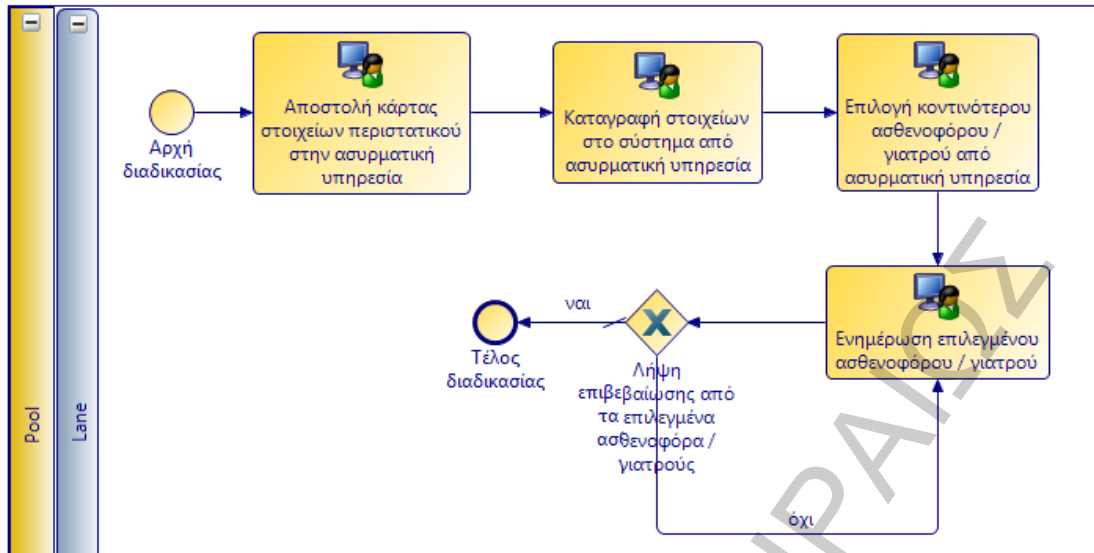
Η διεργασία «άφιξη περιστατικού (τηλέφωνο/ κινητό)» δεν αλλάζει στις δύο περιπτώσεις και παραμένει η ίδια οπότε δεν έχουμε κάποια μείωση χρόνου. Η δεύτερη διεργασία

«καταγραφή περιστατικού» αλλάζει καθώς πριν το eEKAB ο χειριστής του τηλεφωνικού κέντρου έγραφε όλα τα στοιχεία του ασθενούς σε μία κάρτα και την έδινε στην ασυρματική υπηρεσία για να ψάξει για το κοντινότερο ασθενοφόρο/γιατρό. Στη συνέχεια η ασυρματική υπηρεσία το δίνει σε άλλο τμήμα που είναι υπεύθυνο για την καταγραφή των περιστατικών σε ένα κεντρικό βιβλίο. Η διαδικασία φαίνεται στην εικόνα 41. Στο eEKAB όμως η καταγραφή γίνεται κατευθείαν από τον χειριστή του τηλεφωνικού κέντρου μέσω της διαθέσιμης φόρμας της εφαρμογής και η καταγραφή γίνεται σε μία κεντρική βάση δεδομένων. Οπότε ενώ πριν το eEKAB χρειαζόντουσαν τρεις διεργασίες για την διαδικασία της καταγραφής των στοιχείων του περιστατικού τώρα χρειάζεται μόνο μία. Οπότε η διεργασία της καταγραφής είναι τρεις φορές πιο γρήγορη με την ενσωμάτωση του eEKAB (υποθέτουμε ότι κάθε διεργασία χρειάζεται μία μονάδα χρόνου για να εκπληρωθεί). Επίσης η καταγραφή γίνεται πιο γρήγορη αφού η ώρα καταγραφής του περιστατικού συμπληρώνεται αυτόματα, η επιλογή ειδικότητας γιατρού δεν πληκτρολογείται αλλά επιλέγεται από τις ήδη υπάρχουσες ειδικότητες στην κεντρική βάση μέσω μίας λίστας επιλογών και, τέλος, η διεύθυνση δεν χρειάζεται στις περισσότερες των περιπτώσεων να πληκτρολογηθεί ολόκληρη καθώς χρησιμοποιείται η μέθοδος autocomplete η οποία εμφανίζει μία λίστα με διαθέσιμες διευθύνσεις που αντιστοιχούν στη διεύθυνση που έχει πληκτρολογηθεί μέχρι εκείνη τη στιγμή.



Εικόνα 41: Διαδικασία καταγραφής στοιχείων περιστατικού πριν την ενσωμάτωση του ΠΣ eEKAB

Η διαδικασία της ανάθεσης του περιστατικού στο κοντινότερο ασθενοφόρο/γιατρό στο EKAB γίνεται από την ασυρματική υπηρεσία. Η ασυρματική υπηρεσία παραλαμβάνει την κάρτα με τα στοιχεία του περιστατικού από το χειριστή του τηλεφωνικού κέντρου και στη συνέχεια καταγράφει τα στοιχεία σε άλλο σύστημα το οποίο παρουσιάζει τις κοντινότερες ιατρικές μονάδες στο περιστατικό. Στη συνέχεια οι υπάλληλοι της ασυρματικής υπηρεσίας επικοινωνούν με την επιλεγμένη ιατρική μονάδα και την ενημερώνουν για την ανάθεση της αποστολής τους μέσω ασυρμάτου. Η διαδικασία τελειώνει με την λήψη επιβεβαίωσης από την επιλεγμένη ιατρική μονάδα. Η διαδικασία φαίνεται πιο αναλυτικά στην εικόνα 42. Με την ενσωμάτωση του ΠΣ η διαδικασία αυτοματοποιείται. Με την καταγραφή των στοιχείων του περιστατικού στην κεντρική βάση αυτόματα εμφανίζεται η λίστα με τα διαθέσιμα ασθενοφόρα και τους διαθέσιμους ιατρούς φιλτραρισμένους ώστε να εμφανίζονται μόνο της ειδικότητας που χρειάζεται για το περιστατικό. Και οι δύο λίστες είναι ταξινομημένες ανάλογα με τον εκτιμώμενο χρόνο άφιξης τους στον τόπο του συμβάντος. Αυτό σημαίνει ότι η επιλογή είναι γρήγορη διότι τα κοντινότερα ασθενοφόρα και γιατροί είναι οι πρώτοι στη λίστα. Επίσης η επικοινωνία μεταξύ των μερών γίνεται αυτόματα από τους agents οπότε γίνεται ταχύτερη. Τέλος, η διαδικασία της ανάθεσης με το eEKAB επιτυγχάνεται με δύο διεργασίες λιγότερες όπως προκύπτει από την εικόνα 42. Οπότε μπορούμε να πούμε ότι η διαδικασία της ανάθεσης είναι δύο φορές πιο γρήγορη με την ενσωμάτωση του eEKAB. Να τονίσουμε επίσης ότι σε περιπτώσεις που η διεύθυνση που έχει καταγραφεί έχει υπαγορευτεί λάθος από τον καλούντα τότε η καθυστέρηση θα είναι μεγαλύτερη σε αντίθεση με το eEKAB το οποίο μπορεί να κάνει έλεγχο της διεύθυνσης κατά την συμπλήρωση της φόρμας από τον χειριστή του τηλεφώνου όπως περιγράψαμε και νωρίτερα.



Εικόνα 42: Διαδικασία ανάθεσης περιστατικού σε κοντινότερο ασθενοφόρο/γιατρό πριν το eEKAB

Η διαδικασία παροχής οδηγιών προς τα ασθενοφόρα/γιατρούς στο EKAB γίνεται μέσω ασυρμάτου από την ασυρματική υπηρεσία. Αυτό από μόνο του σαν γεγονός είναι χρονοβόρο καθώς εάν για κάποιο λόγο διακοπεί η σύνδεση μεταξύ πληρώματος και ασυρματικής υπηρεσίας το πλήρωμα θα μείνει χωρίς οδηγίες. Με την ενσωμάτωση του eEKAB όταν φτάσει η ανάθεση περιστατικού σε ένα ασθενοφόρο/γιατρό σαν ειδοποίηση, με το άνοιγμα της ειδοποίησης αυτόματα στέλνεται επιβεβαίωση πίσω στην ασυρματική υπηρεσία για να τελειώσει η διαδικασία ανάθεσης που περιγράψαμε προηγουμένως και ταυτόχρονα παρουσιάζεται ένας χάρτης στο κινητό ο οποίος λειτουργεί σαν συσκευή gps και καθοδηγεί το πλήρωμα των ασθενοφόρων προς τον τόπο του συμβάντος. Γενικότερα η χρήση gps είναι ελαφρώς πιο γρήγορη σε σχέση με τη χρήση ασυρμάτου καθώς οι αποφάσεις του οδηγού μπορούν να ληφθούν πιο άμεσα και σωστά αφού εκτός από τις οδηγίες του gps έχει και τη δυνατότητα άμεσης παρακολούθησης της συνολικής πορείας στην οθόνη του κινητού.

Η άφιξη στο περιστατικό ουσιαστικά σηματοδοτεί το τέλος της διαδικασίας της προνοσοκομειακής ιατρικής φροντίδας και δεν υπάρχει κάτι για να αναφερθεί.

Σαν συμπέρασμα έχουμε 4 διαδικασίες που απαρτίζουν την προνοσοκομειακή ιατρική φροντίδα (η τελευταία θεωρείται αμελητέα) από τις οποίες η πρώτη είναι εξίσου γρήγορη όταν ενσωματωθεί το eEKAB στο EKAB, η δεύτερη είναι 3 φορές γρηγορότερη, η τρίτη 2 και η τέταρτη ελαφρώς πιο γρήγορη ας πούμε 1,5 φορές γρηγορότερη. Οπότε έχουμε ότι το ΠΣ

$$\frac{1 + 3 + 2 + 1,5}{4} = 1,875 \approx 1,9$$

eEKAB είναι 1,9 φορές πιο γρήγορο από το προηγούμενο σύστημα του EKAB. Στο κριτήριο του χρόνου δίνουμε τιμή βάρους 3 επειδή το θεωρούμε το σημαντικότερο κριτήριο αξιολόγησης του ΠΣ και τιμή κριτηρίου 2 καθώς θεωρούμε ότι καταφέραμε μία ικανοποιητική μείωση χρόνου των διαδικασιών.

Κόστος

Το κόστος του ΠΣ eEKAB προκύπτει από το κόστος του Web Server το κόστος των κινητών συσκευών android, το κόστος της επαγγελματικής υπηρεσίας των google maps και το κόστος του δικτύου. Το κόστος του web server θα κυμαίνεται από 2000 έως 5000 ευρώ. Επίσης χρειαζόμαστε γύρω στα 1200 κινητά android (με βάση τα οχήματα που διαθέτει το EKAB). Εάν υποθέσουμε ότι το κόστος ενός κινητού android είναι 75 ευρώ τότε το κόστος είναι περίπου $75 \cdot 1200 = 90.000$ ευρώ. Τέλος για τη σύνδεση θα χρησιμοποιήσουμε ένα mobile VPN δίκτυο το οποίο παρέχει τη μεγαλύτερη δυνατή αξιοπιστία και ασφάλεια του οποίου το κόστος είναι περίπου στα 88.800 ευρώ το χρόνο. Τέλος η υπηρεσία των google map ανέρχεται περίπου στα 8.000 το χρόνο.

Οι τιμές είναι ενδεικτικές αλλά βλέπουμε ότι το κόστος ανέρχεται περίπου στα 100.000 ευρώ για τον αρχικό εξοπλισμό συν περίπου 100.000 ευρώ το χρόνο για προσφερόμενες υπηρεσίες. Το κόστος αυτό δεν είναι καθόλου απαγορευτικό για μία δημόσια υπηρεσία όπως το ΕΚΑΒ. Άλλωστε ο δημόσιος τομέας είναι περισσότερο επικεντρωμένος στην εξυπηρέτηση του πελάτη και στην ποιότητα του συστήματος έτσι ώστε αυτό να μπορεί να διαρκέσει για μεγάλο χρονικό διάστημα σε αντίθεση με τον ιδιωτικό τομέα ο οποίος είναι περισσότερο επικεντρωμένος στο να βελτιώσει την παραγωγικότητα και να αποκτήσει κέρδος. Στο κριτήριο αυτό δίνουμε τιμή βάρους 3 καθώς το θεωρούμε πάρα πολύ σημαντικό κομμάτι του ΠΣ και τιμή κριτηρίου 3 καθώς θεωρούμε ότι για αναβάθμιση μίας πολύ κρίσιμης δημόσιας υπηρεσίας το κόστος είναι αρκετά χαμηλό.

Μείωση κατανάλωσης πόρων

Με την ενσωμάτωση του ΠΣ eEKAB στο σύστημα του ΕΚΑΒ πετυχαίνουμε μείωση στην κατανάλωση πόρων ανθρώπινου δυναμικού και πόρων συστήματος. Πιο αναλυτικά μειώνουμε τους πόρους ανθρώπινου δυναμικού στο τμήμα πληροφορίας και στο τμήμα ασυρματικής υπηρεσίας καθώς η διαδικασία καταγραφής των στοιχείων του περιστατικού και η διαδικασία ανάθεσης του περιστατικού στο κοντινότερο ασθενοφόρο/γιατρό γίνονται πολύ γρήγορα από τους χειριστές του τηλεφωνικού κέντρου. Επίσης οι οδηγίες δίνονται κατευθείαν από την κινητή συσκευή των ασθενοφόρων/γιατρών και όχι από την ασυρματική υπηρεσία με αποτέλεσμα να έχουμε μείωση πόρων και στο τμήμα του ανθρώπινου δυναμικού του ασυρμάτου αλλά και στους πόρους συστήματος του ασυρμάτου. Το ανθρώπινο δυναμικό αυτών των υπηρεσιών μπορεί πάρει μετάταξη σε κάποιο άλλο τμήμα του ΕΚΑΒ που έχει έλλειψη και το σύστημα του ασυρμάτου μπορεί να χρησιμοποιηθεί σαν επιπρόσθετη βοήθεια στα πληρώματα των ασθενοφόρων και των γιατρών οποτεδήποτε χρειαστεί. Στο κριτήριο αυτό βάζουμε τιμή βάρους 1 καθώς είναι σημαντικό αλλά όχι και καίριο και επίσης του δίνουμε τιμή κριτηρίου 2 καθώς καταφέραμε να αποφορτίσουμε κάποια τμήματα του ΕΚΑΒ με επιτυχία.

Ευκολία εξαγωγής συμπερασμάτων από τα διάφορα περιστατικά

Είναι πολύ χρήσιμο να μπορούμε να εξαγάγουμε εύκολα συμπεράσματα από το πλήθος των διαφόρων περιστατικών που καταφθάνουν στο τηλεφωνικό κέντρο. Αυτή η διαδικασία μπορεί να μας δώσει σημαντικά στοιχεία τα οποία αργότερα μπορεί να αναλυθούν έτσι ώστε στη συνέχεια με τις απαραίτητες ενέργειες και σχεδιασμό, μελλοντικά, να αποφευχθούν διαφόρων ειδών περιστατικά ή να αλλάξει η κατανομή των ασθενοφόρων σε μία περιοχή ανάλογα με την κατανομή των περιστατικών στην περιοχή κτλ. Κάτι τέτοιο είναι σχεδόν αδύνατο με το σημερινό σύστημα του ΕΚΑΒ καθώς η καταγραφή των περιστατικών γίνεται σε ένα βιβλίο. Δηλαδή η διαδικασία δεν είναι μηχανογραφημένη. Αν σκεφτούμε επίσης ότι τα έκτακτα περιστατικά είναι περίπου 30.000 το μήνα μόνο στην Αττική τότε αντιλαμβανόμαστε πόσο ανέφικτο θα ήταν ένα τέτοιο εγχείρημα. Με το eEKAB τα στοιχεία των περιστατικών θα καταγράφονται στην κεντρική βάση με αποτέλεσμα η εξόρυξη δεδομένων και οπότε και η εξαγωγή συμπερασμάτων να είναι πολύ πιο εύκολη απ' ό,τι πριν. Στο βάρος αυτού του κριτηρίου δίνουμε την τιμή 2 καθώς είναι πολύ σημαντική η μηχανογράφηση των περιστατικών και τιμή κριτηρίου 4 καθώς η εξαγωγή συμπερασμάτων έγινε αισθητά πιο εύκολη από πριν.

5.2.2 Χρηστικότητα

Στην κατηγορία της χρηστικότητας περιλαμβάνονται κριτήρια τα οποία περιγράφουν την αλληλεπίδραση του ΠΣ με το χρήστη. Τέτοια κριτήρια είναι η ευκολία χρήσης, το πλήθος των λειτουργιών που προσφέρει το ΠΣ καθώς και η παράλληλη λειτουργία του από πολλούς χρήστες. Η κατηγορία αυτή είναι αρκετά σημαντική αλλά όχι και κρίσιμη για την εύρυθμη λειτουργία του ΠΣ οπότε θα της δώσουμε τιμή βάρους 1. Στη συνέχεια παρουσιάζονται και αναλύονται τα κριτήρια που σχετίζονται με τη χρηστικότητα.

Ευκολία χρήσης

Ένα σημαντικό χαρακτηριστικό που θα πρέπει να έχει ένα ΠΣ είναι το όσο το δυνατόν πιο εύκολη διεπαφή. Στο eEKAB όλες οι λειτουργίες επικοινωνίας μεταξύ client και server γίνονται από τους agents όπως έχει αναλυθεί στην ενότητα σχεδιασμού της εφαρμογής. Με τον τρόπο οι πολύπλοκες διεργασίες αποκρύβονται από το χρήστη με αποτέλεσμα να έχουμε μία απλή διεπαφή. Σε αυτό το κριτήριο θα δώσουμε τιμή βάρους 1 και τιμή κριτηρίου 4 καθώς η διεπαφή είναι πολύ απλή.

Πλήθος λειτουργιών

Το πλήθος λειτουργιών που προσφέρει το eEKAB στο χρήστη είναι ικανοποιητικές. Παρ' όλα αυτά θα μπορούσε να είχε περισσότερες για μεγαλύτερη χρησιμότητα και προσφορά επιλογών στους χρήστες. Στο κριτήριο δίνουμε τιμή βάρους 2 και τιμή κριτηρίου 2.

Παράλληλη λειτουργία

Σημαντικό κριτήριο για την εύρυθμη λειτουργία του συστήματος είναι η παράλληλη λειτουργία του από πολλούς χρήστες. Αυτό επιτυγχάνεται με τη βοήθεια της αρχιτεκτονικής client-server και της κεντρικής βάσης δεδομένων. Παρ' όλα αυτά δεν έχουν υλοποιηθεί δικλείδες ασφαλείας έτσι ώστε να αποφεύγεται η πιθανότητα την ώρα που κάποιος χρήστης επιλέγει ένα ασθενοφόρο κάποιος άλλος χειριστής να προλάβει να το επιλέξει πρώτος. Στο κριτήριο αυτό δίνουμε τιμή βάρους 3 καθώς είναι πάρα πολύ σημαντικό και τιμή κριτηρίου 1 καθώς εν μέρει πληροί τις προϋποθέσεις.

5.2.3 Αξιοπιστία

Στην κατηγορία αυτή συμπεριλαμβάνονται τα κριτήρια αξιολόγησης της γενικότερης αξιοπιστίας και ασφάλειας του συστήματος και του βαθμού στο οποίο το σύστημα είναι ευάλωτο σε απειλές. Σε αυτή την κατηγορία περιλαμβάνονται τα κριτήρια της ασφάλειας, της επιβεβαίωσης λήψης συντεταγμένων περιστατικού, της έγκυρης διεύθυνσης και του ελέγχου εισαγωγής των δεδομένων. Η κατηγορία αυτή είναι πολύ σημαντική και της δίνουμε τιμή βάρους 2.

Ασφάλεια

Το κριτήριο αυτό αξιολογεί το κατά πόσο το σύστημά μας είναι ευάλωτο σε επιθέσεις από κακόβουλους χρήστες. Στο eEKAB έχουμε πάρει κάποια μέτρα για την ασφάλεια του συστήματος. Το πρώτο μέτρο είναι η αυθεντικοποίηση του χρήστη. Κάθε χρήστης έχει έναν συγκεκριμένο κωδικό και όνομα χρήστη για να συνδέεται στον server. Το δεύτερο είναι η ακεραιότητα της πληροφορίας αφού ο κωδικός του client μεταφέρεται κωδικοποιημένος με τη βοήθεια της συνάρτησης κατακερματισμού sha1. Στη μεριά του server ο κωδικός που είναι αποθηκευμένος στη βάση κωδικοποιείται και αυτός με τη βοήθεια της sha1 και συγκρίνεται με τον κωδικό του client. Με αυτό τον τρόπο πετυχαίνουμε ταυτόχρονα και αυθεντικοποίηση χρήστη και ακεραιότητα ευαίσθητης πληροφορίας. Όσον αφορά την εμπιστευτικότητα του δικτύου σε περίπτωση που χρησιμοποιηθεί mobile VPN θα την παρέχει το δίκτυο με τη βοήθεια του SSL. Στο κριτήριο αυτό δίνουμε τιμή βάρους 3 και τιμή κριτηρίου 2.

Επιβεβαίωση λήψης συντεταγμένων περιστατικού

Για να είναι αξιόπιστο το σύστημα θα πρέπει το κέντρο ελέγχου να γνωρίζει όταν ένα ασθενοφόρο ή γιατρός έλαβε το περιστατικό και δέχτηκε την ανάθεση του περιστατικού. Αυτό γίνεται στέλνοντας μία επιβεβαίωση λήψης συντεταγμένων από τα ασθενοφόρα/γιατρούς προς τον server. Στο eEKAB υλοποιήσαμε αυτή την διαδικασία αλλά δεν υλοποιήθηκε η διαδικασία που ακολουθείται σε περίπτωση μη επιβεβαίωσης. Οπότε στο κριτήριο αυτό δίνουμε τιμή βάρους 3 και τιμή κριτηρίου 1.

Εγκυρη διεύθυνση

Κατά την εισαγωγή της διεύθυνσης του περιστατικού υπάρχει η απαίτηση ελέγχου εγκυρότητας της διεύθυνσης διότι σε περίπτωση μη έγκυρης διεύθυνσης θα χαθεί πολύτιμος χρόνος κατά την διαδικασία προνοσοκομειακής ιατρικής φροντίδας. Στο eEKAB αυτό έχει υλοποιηθεί με live έλεγχο κατευθείαν από τη βάση δεδομένων της Google που περιέχει τις διευθύνσεις του google map. Τιμή βάρους για αυτό το κριτήριο 1 και τιμή κριτηρίου 4.

Έλεγχος φορμών

Πολύ σημαντικό επίσης είναι το κριτήριο του ελέγχου των φορμών. Κατά τη συμπλήρωση όλων των φορμών της εφαρμογής θα πρέπει να υπάρχει έλεγχος έγκυρης εισαγωγής δεδομένων έτσι ώστε τα δεδομένα να αποθηκευτούν σωστά στην κεντρική βάση δεδομένων. Στο eEKAB έχουμε έλεγχο εισαγωγής δεδομένων σε όλες τις φόρμες για πλήρη συνοχή των δεδομένων που αποθηκεύονται στη βάση δεδομένων. Οπότε βάζουμε τιμή βάρους 1 και τιμή κριτηρίου 4.

5.2.4 Αποτελεσματικότητα

Η κατηγορία αυτή έχει τιμή βάρους 1 και περιλαμβάνει τα κριτήρια της συνεργασίας μεταξύ των τμημάτων του EKAB και της χρήσης σύγχρονων τεχνολογιών.

Συνεργασία μεταξύ των τμημάτων του EKAB

Η καλύτερη συνεργασία μεταξύ των τμημάτων του EKAB είναι ουσιαστικά το αποτέλεσμα που θέλουμε να έχουμε με την ενσωμάτωση του eEKAB με απώτερο σκοπό την μείωση του χρόνου. Με το eEKAB η συνεργασία των τμημάτων γίνεται πιο εύκολη και πιο γρήγορη και έτσι δίνουμε τιμή βάρους 2 και τιμή κριτηρίου 3.

Χρήση σύγχρονων τεχνολογιών

Μία αρχική απαίτηση πριν την υλοποίηση του ΠΣ eEKAB ήταν ακριβώς αυτό η χρήση σύγχρονων τεχνολογιών οι οποίες μπορεί να οδηγήσουν σε εύκολη μελλοντική αναβάθμιση του συστήματος. Στο eEKAB το πετύχαμε κατά πολύ αφού χρησιμοποιήσαμε πολύ σύγχρονες τεχνολογίες όπως το λειτουργικό android και οι υπηρεσίες των google maps. Στο κριτήριο αυτό δίνουμε τιμή βάρους 1 και τιμή κριτηρίου 4.

5.3 Αποτελέσματα αξιολόγησης

Σύμφωνα με τον τύπο που αναφέραμε στην περιγραφή του μοντέλου αξιολόγησης μας θα βγάλουμε τον βαθμό στον οποίο πετύχαμε να εκπληρώσουμε τις αρχικές μας απαιτήσεις. Αναλυτικά έχουμε ότι:

$$\begin{aligned}
 Q &= 3 * (3 * 2 + 3 * 3 + 1 * 2 + 2 * 4) + 1 * (1 * 4 + 2 * 2 + 3 * 1) + 2 * (3 * 2 + 3 * 1 + 1 * 4 + \\
 &+ 1 * 4) + 1 * (2 * 3 + 1 * 4) = \\
 &= 3 * (6 + 9 + 2 + 8) + 1 * (4 + 4 + 3) + 2 * (6 + 3 + 4 + 4) + 1 * (6 + 4) = \\
 &= 3 * 25 + 1 * 11 + 2 * 17 + 1 * 10 = 75 + 11 + 34 + 10 = 130 \Leftrightarrow \\
 &\Leftrightarrow Q = 130
 \end{aligned}$$

5.4 Συμπεράσματα

Από την ανάλυση που έγινε στην ενότητα της αξιολόγησης μπορούμε να βγάλουμε αρκετά και χρήσιμα συμπεράσματα. Από την ανάλυση αυτή γίνονται εμφανή με τον πιο αξιόπιστο τρόπο τα δυνατά και τα αδύναμα σημεία του συστήματος μας.

Βλέπουμε ότι η απόδοση του συστήματος μας είναι 130. Αυτό βέβαια από μόνο του δεν μας λέει και πολλά. Γι' αυτό το λόγο θα πρέπει να συγκρίνουμε την απόδοση του με βάση την απόδοση 4 υποτιθέμενων συστημάτων. Έτσι έχουμε ένα σύστημα το οποίο δεν πληροί εξ' ολοκλήρου τις αρχικές απαιτήσεις και το οποίο με εφαρμογή του τύπου της απόδοσης έχει απόδοση $Q=52$. Το δεύτερο σύστημα πληροί τις απαιτήσεις και έχει απόδοση $Q=104$ το τρίτο υπερβαίνει ελαφρώς τις απαιτήσεις και έχει απόδοση $Q=156$ και τέλος το τέταρτο σύστημα πληροί αισθητά τις απαιτήσεις και έχει απόδοση $Q=208$. Οπότε βλέπουμε ότι το σύστημα μας βρίσκεται ανάμεσα στο δεύτερο σύστημα που πληροί τις απαιτήσεις και στο τρίτο το οποίο υπερβαίνει ελαφρώς. Επίσης εάν δεχτούμε ότι το τέλειο σύστημα έχει $Q=208$ τότε με βάση την κλίμακα του 100 το σύστημα μας έχει απόδοση 62,5%.

Σαν συμπέρασμα το μοντέλο που χρησιμοποιήσαμε μας παρουσίασε μία εικόνα του βαθμού στον οποίο καταφέραμε να εκπληρώσουμε τις αρχικές απαιτήσεις και η εικόνα αυτή αξιολογείται σαν ικανοποιητική. Πιο αναλυτικά, μπορούμε να πούμε ότι σε σύγκριση με το σημερινό σύστημα του ΕΚΑΒ το δικό μας είναι πιο αποδοτικό. Πιο συγκεκριμένα το eΕΚΑΒ βελτιώνει το τωρινό σύστημα προνοσοκομειακής ιατρικής φροντίδας και τα κυριότερα σημεία βελτίωσης παρατίθενται παρακάτω:

- Μηχανογράφηση καταγραφής στοιχείων περιστατικών με δυνατότητα ανάλυσης τους και εξαγωγής συμπερασμάτων
- Μείωση χρόνου διαδικασιών του ΕΚΑΒ
- Ευκολότερη διαχείριση του συστήματος του ΕΚΑΒ
- Καλύτερη κατανομή ανθρώπινου δυναμικού
- Καλύτερη γεωγραφική κατανομή ασθενοφόρων/γιατρών μετά από ανάλυση των περιστατικών.

Οπτική απεικόνιση στα ασθενοφόρα/γιατρούς χάρτη και διαδρομής προς το περιστατικό

6. ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΤΑΣΕΙΣ

Παρατηρώντας την αξιολόγηση του πληροφοριακού συστήματος eEKAB καθώς και τον εσωτερικό κανονισμό λειτουργίας του EKAB [4] μπορούμε εύκολα να διακρίνουμε τα σημεία τα οποία χρήζουν βελτίωσης. Η αντιμετώπιση των προβλημάτων αυτών θα αναβαθμίσει την λειτουργία του συστήματος, καθιστώντας το ευκολότερο στην χρήση, γρηγορότερο και κατά συνέπεια αποτελεσματικότερο.

Πιο συγκεκριμένα, ένας τομέας στον οποίο το σύστημα επιδέχεται βελτίωση είναι ο τομέας της ασφάλειας. Το δίκτυο που θα πρέπει να χρησιμοποιηθεί για την υλοποίηση του ΠΣ θα πρέπει να χαρακτηρίζεται από ένα μεγάλο επίπεδο ασφάλειας. Μία επιλογή είναι η χρήση ενός mobile VPN (virtual private network) το οποίο θα επιτρέπει στους clients να παραμένουν συνδεδεμένοι στο δίκτυο του EKAB ακόμα και όταν θα αλλάζουν ασύρματα δίκτυα κατά την κίνησή τους. Επίσης το VPN προσφέρει ένα πιστοποιημένο, και κρυπτογραφημένο κανάλι επικοινωνίας για ασφαλή μεταφορά δεδομένων μέσω δημόσιων δικτύων. Επίσης, θα πρέπει να αναπτυχθούν ειδικές τεχνικές για την υποστήριξη παράλληλης λειτουργίας του συστήματος για την αποφυγή σύγκρουσης δεδομένων. Θα πρέπει να αποφευχθεί δηλαδή η πιθανότητα την ώρα που κάποιος χρήστης επιλέγει ένα ασθενοφόρο κάποιος άλλος χειριστής να προλάβει να το επιλέξει πρώτος. Επίσης όπως αναφέραμε και στην αξιολόγηση τίθεται το θέμα της υλοποίησης της διαδικασίας της περίπτωσης μη επιβεβαίωσης της λήψης της ανάθεσης του περιστατικού από κάποιο ασθενοφόρο ή γιατρό.

Ένα άλλο σημαντικό στοιχείο που είναι απαραίτητο για την μελλοντική αναβάθμιση του συστήματος του EKAB είναι η ανάπτυξη ενός εργαλείου για εξαγωγή συμπερασμάτων από τα περιστατικά. Με τον τρόπο αυτό το σύστημα του EKAB μπορεί να αναβαθμιστεί έτσι ώστε για παράδειγμα τα περισσότερα ασθενοφόρα να τοποθετούνται σε περιοχές με υψηλό αριθμό περιστατικών ή επίσης να εμφανίζεται καρτέλα με το ιστορικό ασθενών κατά την κλήση στο τηλεφωνικό κέντρο με σκοπό την καλύτερη αντιμετώπιση του περιστατικού κτλ. Για να αναπτυχθεί ένα τέτοιο εργαλείο επιβάλλεται οι πληροφορίες για ένα περιστατικό να είναι περισσότερες. Πιο αναλυτικά, θα πρέπει να καταγράφονται η ηλικία, το τηλέφωνο, η περιγραφή πάθησης και πιθανόν και άλλες πληροφορίες έτσι ώστε τα συμπεράσματα που θα εξαχθούν να είναι όσο το δυνατόν πιο αναλυτικά και ολοκληρωμένα.

Τέλος, το σύστημα μπορεί να εξελιχθεί ώστε να περιλαμβάνει καλύτερη επικοινωνία μεταξύ των τμημάτων του EKAB. Πιο συγκεκριμένα το πλήρωμα των ασθενοφόρων καθώς και οι γιατροί θα πρέπει να ενημερώνονται με περισσότερα χαρακτηριστικά του περιστατικού όπως το σήμα του κατεπείγοντος έτσι ώστε να γνωρίζουν αν θα πρέπει να ανοίξουν τη σειρά και επίσης για το ονοματεπώνυμο του ασθενούς και το ονοματεπώνυμο που αναγράφεται στο κουδούνι. Επίσης, τα ασθενοφόρα/γιατροί θα πρέπει να ενημερώνουν το κέντρο ελέγχου για την άφιξη τους στο περιστατικό, την κατάσταση του περιστατικού και για την αναχώρησή τους από τον τόπο του συμβάντος. Με τη σειρά του το κέντρο ελέγχου θα πρέπει να επιλέγει το καταλληλότερο νοσοκομείο ανάλογα με την κατάσταση του περιστατικού και την τοποθεσία του και να αποστέλλει τις συντεταγμένες του στα ασθενοφόρα/γιατρούς.

7. ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Sultan Al - Shaqsi, Models of International Emergency Medical Service (EMS) Systems, Oman Medical Journal 2010, Volume 25, Issue 4, Οκτώβριος 2010. Ανακτήθηκε στις 23 – 11 – 2012.
- [2] Bahman S. Roudsari a, Avery B. Nathens b, Carlos Arreola-Risa c, Peter Cameron d, Ian Civil e, Giouli Grigoriou f, Russel L. Gruen g, Thomas D. Koepsell h,i, Fiona E. Lecky j, Rolf L. Lefering k, Moishe Liberman l, Charles N. Mock m,h, Hans-Jo`rg Oestern n, Elenie Petridou o,p, Thomas A. Schildhauer q, Christian Waydhas r, Moosa Zargar s, Frederick P. Rivara, Emergency Medical Service (EMS) systems in developed and developing countries, International journal of the care of the injured (2007), Ανακτήθηκε στις 26 – 11 – 2012.
- [3] "Design, Origin and Meaning of the Star of Life". NHTSA. Διαθέσιμο στη διεύθυνση: <http://www.ems.gov/vgn-ext-templating/ems/sol/pages/DesignOrigin.htm>. Ανακτήθηκε στις 26 – 11 – 2012.
- [4] Σωματείο εργαζομένων στο ΕΚΑΒ, εσωτερικός κανονισμός λειτουργίας ΕΚΑΒ. Διαθέσιμο στη διεύθυνση: <http://www.somatioekab166.gr/theseis.php?col=3>. Ανακτήθηκε στις 27 – 11 – 2012.
- [5] Παπαευσταθίου Νικόλαος, Οργανωτική & λειτουργική δομή του ΕΚΑΒ. Διαθέσιμο στη διεύθυνση: http://library.tee.gr/digital/m2150/m2150_papaefstathiou.pdf. Ανακτήθηκε στις 26 – 11 – 2012.
- [6] www.ekab.gr
- [7] Dick WF (2003). "Anglo-American vs. Franco-German emergency medical services system". Ανακτήθηκε στις 30 – 11 - 2012.
- [8] Benjamin Schooley, Thomas Horan, Nathan Botts, Aisha Noamani, ITS and Transportation Safety: EMS System Data Integration to Improve Traffic Crash Emergency Response and Treatment – Phase I. Ανακτήθηκε στις 2 – 12 – 2012.
- [9] Tang Paul, Ash Joan, Bates David, Overhage J., Sands Daniel "Personal Health Records: Definitions, Benefits, and Strategies for Overcoming Barriers to Adoption". Ανακτήθηκε στις 2 – 12 – 2012
- [10] 911 service. Διαθέσιμο στη διεύθυνση: <http://transition.fcc.gov/pshs/services/911-services/>. Ανακτήθηκε στις 3 – 12 – 2012.
- [11] Freeman S., How OnStar Works. Διαθέσιμο στη διεύθυνση: <http://auto.howstuffworks.com/onstar.htm> . Ανακτήθηκε στις 4 - 12 – 2012.
- [12] Verma MK, Lange RC, McGarry DC. A Study of US Crash Statistics from Automated Crash Notification Data. The 20th International Technical Conference on the Enhanced Safety of Vehicles. Lyon, 2007. Ανακτήθηκε στις 4 - 12 – 2012.
- [13] [http://www.termwiki.com/EN:computer-aided_dispatch_\(CAD\)](http://www.termwiki.com/EN:computer-aided_dispatch_(CAD)). Ανακτήθηκε στις 5 – 12 – 2012
- [14] Adams, R. Is priority dispatching right for you? Ανακτήθηκε στις 5 – 12 – 2012.
- [15] Computer aided dispatch (CAD). Emergency Medical Services Authority. Διαθέσιμο στη διεύθυνση: <http://www.emsaonline.com/cad.html>. Ανακτήθηκε στις 5 – 12 – 2012.
- [16] Kari Pugh, Define Police Computer Aided Dispatch System. Διαθέσιμο στη διεύθυνση: http://www.ehow.com/facts_5502175_define-computer-aided-dispatch-system.html. Ανακτήθηκε στις 5 – 12 – 2012.
- [17] NCHRP Report, Τεύχος 340, Assessment of Advanced Technologies for Relieving Urban Traffic Congestion. Ανακτήθηκε στις 12 - 12 – 2012
- [18] "Geographic Information Systems as an Integrating Technology: Context, Concepts, and Definitions". ESRI. Διαθέσιμο στη διεύθυνση: <http://www.colorado.edu/geography/gcraft/notes/intro/intro.html>. Ανακτήθηκε στις 12 - 12 – 2012.

- [19] Fire/Rescue and EMS. ESRI. Διαθέσιμο στη διεύθυνση: <http://www.esri.com/Industries/public-safety/fire-rescue-ems>. Ανακτήθηκε στις 12 – 12 – 2012.
- [20] Marshall Brain, Tom Harris. How GPS Receivers Work. Διαθέσιμο στη διεύθυνση: <http://electronics.howstuffworks.com/gadgets/travel/gps.htm>. Ανακτήθηκε στις 12 – 12 – 2012.
- [21] Haque, A. GIS, public service, and the use of democratic governance. Ανακτήθηκε στις 12 – 12 – 2012.
- [22] Lawrence J. Harman, Uma Shama, Mobile Data Terminals: A Synthesis of Transit Practice. Ανακτήθηκε στις 13 – 12 – 2012.
- [23] U.S. Department of Transportation, National Highway Traffic Safety Administration, Star of Life. Ανακτήθηκε στις 22 – 1 – 2013.
- [24] David Jaslow, MD, MPH, Arthur Yancy II, MD, MPH, Andrew Milsten, MD, for the National Association of EMS Physicians Standards and Clinical Practice Committee, MASS GATHERING MEDICAL CARE. Ανακτήθηκε στις 22 – 1 – 2013.
- [25] E Pitt, A Puspongoro, Prehospital care in Indonesia. Ανακτήθηκε στις 22 – 1 – 2013
- [26] Drew E. Dawson, NATIONAL EMERGENCY MEDICAL SERVICES INFORMATION SYSTEM (NEMSIS). Ανακτήθηκε στις 22 – 1 – 2013.
- [27] Bruce J. Walz, Kurt M. Krumperman, Jason J. Zigmont, Foundations of EMS Systems. Ανακτήθηκε στις 22 – 1 – 2013.
- [28] Graham Nichol, Allan S Detsky, Ian G Stiell, Keith O'Rourke, George Wells, Andreas Laupacis, Effectiveness of Emergency Medical Services for Victims of Out-of-Hospital Cardiac Arrest: A Metaanalysis. Ανακτήθηκε στις 23 – 1 – 2013.
- [29] Wei-Meng Lee, Beginning Android 4 Application Development. Ανακτήθηκε στις 13 – 12 – 2012
- [30] "What is Android: Introduction, Features & Applications", EngineersGarage. Διαθέσιμο στη διεύθυνση: <http://www.engineersgarage.com/articles/what-is-android-introduction>. Ανακτήθηκε στις 13 – 12 – 2012
- [31] "Philosophy and Goals". Android Open Source Project. Google. Διαθέσιμο στη διεύθυνση: <http://source.android.com/about/philosophy.html>. Ανακτήθηκε στις 13 – 12 – 2012
- [32] Burnette Ed, Hello, Android: Introducing Google's Mobile Development Platform. Ανακτήθηκε στις 13 – 12 – 2012
- [33] Vogelstein, Fred (April 2011). "How the Android Ecosystem Threatens the iPhone". Wired. Διαθέσιμο στη διεύθυνση: http://www.wired.com/magazine/2011/04/mf_android/all/1. Ανακτήθηκε στις 13 – 12 – 2012.
- [34] «Smartphone statistics and market share». Διαθέσιμο στη διεύθυνση: <http://www.email-marketing-reports.com/wireless-mobile/smartphone-statistics.htm#OS>. Ανακτήθηκε στις 13 – 12 – 2012
- [35] "Google Play hits 600,000 apps, 20 billion total installs". Διαθέσιμο στη διεύθυνση: http://www.engadget.com/2012/06/27/google-play-hits-600000-apps/?icid=eng_latest_art. Ανακτήθηκε στις 13 – 12 – 2012.
- [36] Sheran Gunasekera, Android Apps Security. Ανακτήθηκε στις 14 – 12 – 2012.
- [37] Jeff Friesen, Dave Smith, Android Recipes: A Problem-Solution Approach. Ανακτήθηκε στις 14 – 12 – 2012
- [38] "Application Fundamentals | Android Developers". Διαθέσιμο στη διεύθυνση: <http://developer.android.com/guide/components/fundamentals.html>. Ανακτήθηκε στις 14 – 12 – 2012
- [39] Marko Gargenta, Learning Android. Ανακτήθηκε στις 15 – 12 – 2012
- [40] Mark Murphy, Beginning Android 3. Ανακτήθηκε στις 15 – 12 – 2012
- [41] "Managing the Activity Lifecycle | Android Developers". Διαθέσιμο στη διεύθυνση: <http://developer.android.com/training/basics/activity-lifecycle/index.html>. Ανακτήθηκε στις 15 – 12 – 2012

- [42] "Starting an Activity | Android Developers". Διαθέσιμο στη διεύθυνση: <http://developer.android.com/training/basics/activity-lifecycle/starting.html>. Ανακτήθηκε στις 15 – 12 – 2012
- [43] "Brand Guidelines | Android Developers". Διαθέσιμο στη διεύθυνση: <http://developer.android.com/distribute/googleplay/promote/brand.html>. Ανακτήθηκε στις 23 – 1 – 2013.
- [44] "App Framework | Android Developers". Διαθέσιμο στη διεύθυνση: <http://developer.android.com/about/versions/index.html>. Ανακτήθηκε στις 23 – 1 – 2013.
- [45] J. Jing et al, Client-Server Computing in Mobile Environments. Ανακτήθηκε στις 01 – 12 – 2013.
- [46] Heather Schneider; Valentino Lee; Robbie Schell, Introduction to Mobile Application Architectures. Ανακτήθηκε στις 01 – 12 – 2013.
- [47] Dr. Antonio Moreno, Juan Pavón, Issues in Multi-Agent Systems: The AgentCities.ES Experience. Ανακτήθηκε στις 02 – 12 – 2013.
- [48] Lange Danny, Oshima Mitsuru "Programming and Deploying Java Mobile Agents with aglets". Ανακτήθηκε στις 02 – 12 – 2013.
- [49] Wooldridge Michael, "An introduction to multiagent systems". Ανακτήθηκε στις 02 – 12 – 2013.
- [50] Raid Moh'd Al-adaileh, «An Evaluation of Information Systems Success: A User Perspective - the Case of Jordan Telecom Group», Ανακτήθηκε στις 02 – 12 – 2013.
- [51] Olegas Vasilecas; Algis Saulis; Saulius Dereškevičius, «Evaluation of Information Systems Procurement: Goal and Task-Driven Approaches», Ανακτήθηκε στις 02 – 12 – 2013.
- [52] Emulator Networking. Using the Android Emulator. Διαθέσιμο στη διεύθυνση: <http://developer.android.com/tools/devices/emulator.html#emulatorenetworking>. Ανακτήθηκε στις 18 – 09 – 2012.

ΠΑΡΑΡΤΗΜΑ

A. Επεξήγηση κώδικα

A.1. Κώδικας προγράμματος κεντρικής εφαρμογής

Η εφαρμογή είναι υλοποιημένη με τη βοήθεια της γλώσσας προγραμματισμού php, javascript μέσω της βιβλιοθήκης jquery. Χρησιμοποιήθηκε επίσης η γλώσσα φύλλων στυλ CSS (Cascading Style Sheets) για τον έλεγχο της εμφάνισης της κεντρικής εφαρμογής καθώς και η τεχνική AJAX (Asynchronous Javascript and XML) για δημιουργία ασύγχρονων λειτουργιών στην εφαρμογή. Για την επικοινωνία μεταξύ του server και του mobile client χρησιμοποιήθηκαν sockets. Επίσης χρησιμοποιήθηκε η MySql για την κατασκευή της βάσης δεδομένων. Τέλος χρησιμοποιήθηκε το API της Google, το Google Maps Javascript API V3.

Η εφαρμογή αποτελείται από αρχεία .php, ένα φάκελο που περιέχει τα javascript αρχεία, ένα φάκελο που περιέχει τα .css αρχεία καθώς και ένα φάκελο που περιέχει τις .png εικόνες που περιέχονται στην εφαρμογή. Πιο συγκεκριμένα αποτελείται από:

Δέκα .php αρχεία

- index.php
- mysql.php
- ajaxMysql.php
- search.php
- refreshambtable.php
- refreshdocTable.php
- modifyAmbTable.php
- modifyDocTable.php
- server_socket.php
- SendIncident.php

Ένα .html αρχείο

- dialogs.html

Ένα φάκελο με 6 αρχεία .js

- template.js
- initialize.js
- check_errors.js
- geo_autocomplete.js
- jquery.autocomplete_geomod.js
- jquery-ui.min.js

Ένα φάκελο με 8 αρχεία .css και 1 φάκελο images που περιέχει εικόνες που χρειάζονται για το σχεδιασμό φορμών

- search.css
- tab1.css
- tab2.css
- tab3.css
- tables.css
- tabs.css
- jquery.autocomplete.css
- jquery-ui.css

Ένα φάκελο που περιέχει τις εικόνες που εμφανίζονται στην κεντρική εφαρμογή

- bullet-green.png
- bullet-red.png
- bullet-yellow.png
- confirm.png
- error.png

ΑΡΧΕΙΑ PHP

Το αρχείο που εκτελείται πρώτο κατά την έναρξη της εφαρμογής είναι το αρχείο `index.php`. Ο κώδικας παρατίθεται παρακάτω.

index.php

Αρχείο που εκτελείται πρώτο. Από εδώ ξεκινούν όλες οι λειτουργίες καθώς και η εμφάνιση της εφαρμογής.

Στο τμήμα `<head>` του αρχείου συμπεριλαμβάνουμε όλα τα javascript και css αρχεία καθώς και 2 βιβλιοθήκες μία για τους χάρτες της google και μία για την jquery. Επίσης χρησιμοποιούμε και το αρχείο `mysql.php` για να συνδεθούμε στην βάση δεδομένων.

Στο τμήμα `<body>` του αρχείου ξεκινάει η εκτέλεση στο παρασκήνιο του αρχείου `server_socket.php` το οποίο είναι υπεύθυνο για την επικοινωνία με τους mobile clients και την ενημέρωση της βάσης για την κατάσταση των clients (συνδεδεμένοι, αποσυνδεδεμένοι, επί τω έργω) και στη συνέχεια σχεδιάζεται ο σκελετός της κεντρικής εφαρμογής.

mysql.php

Αρχείο που είναι υπεύθυνο για τη σύνδεση της εφαρμογής με τη βάση `mysql`. Συμπεριλαμβάνεται σε όλα τα αρχεία που χρειάζεται να κάνουν σύνδεση με τη βάση.

```
$link = mysql_connect("localhost", "root", "");
if (!$link) {
    die('Could not connect: ' . mysql_error());
}
$db_selected = mysql_select_db("cad");
if (!$db_selected) {
    die('Could not select database: ' . mysql_error());
}
$greek_char = mysql_query("SET NAMES 'utf8'");
if (!$greek_char) {
    die('Could not query:' . mysql_error());
}
```

ajaxMysql.php

Αρχείο που είναι υπεύθυνο για να κάνει επερωτήσεις στη βάση δεδομένων ασύγχρονα και να επιστρέφει το πρώτο αποτέλεσμα. Χρησιμοποιείται από τη jquery με τη βοήθεια της τεχνικής `ajax` για να επικοινωνεί η javascript με τη βάση δεδομένων.

```
include_once("mysql.php");
$result = mysql_query($_POST["query"]) or die('Could not query:' . mysql_error());
echo mysql_result($result,0,0);
```

search.php

Αρχείο που εκτελείται όταν συμπληρώσουμε τη φόρμα του περιστατικού και πατήσουμε «υποβολή». Είναι υπεύθυνο για να εμφανίσει τους αρμόδιους γιατρούς και ασθενοφόρα ασύγχρονα οι οποίοι θα επιλεγούν στη συνέχεια για να αναλάβουν το περιστατικό.

Αυτό επιτυγχάνεται με την κατασκευή ενός `xml` αρχείου από τη βάση δεδομένων ανάλογα με τις ιδιότητες του τρέχοντος περιστατικού (ειδικότητα γιατρού, συνδεδεμένοι γιατροί και ασθενοφόρα) το οποίο αποστέλλεται με τεχνική `ajax` στο `script` της javascript το οποίο θα το επεξεργαστεί και στη συνέχεια θα κατασκευάσει τους πίνακες της πρώτης καρτέλας της εφαρμογής.

```
header('Content-type: text/xml');
```

```

function build_xml($query,$table)
{
    $result = mysql_query($query);
    if (!$result)
        die('Could not query:' . mysql_error());
    $xml = "";
    $num_rows=mysql_num_rows($result);

    if($table==='doctors')
    {
        for($i = 0; $i < $num_rows; $i++)
        {
            $xml .= "<doctor>";
            $xml .= "<id>" . mysql_result($result,$i,0) . "</id>";
            $xml .= "<name>" . mysql_result($result,$i,1) . "</name>";
            $xml .= "<surname>" . mysql_result($result,$i,2) . "</surname>";
            $xml .= "<lat>" . mysql_result($result,$i,3) . "</lat>";
            $xml .= "<lon>" . mysql_result($result,$i,4) . "</lon>";
            $xml .= "</doctor>";
        }
    }
    else
    {
        for($i = 0; $i < $num_rows; $i++)
        {
            $xml .= "<ambulance>";
            $xml .= "<id>" . mysql_result($result,$i,0) . "</id>";
            $xml .= "<plates>" . mysql_result($result,$i,1) . "</plates>";
            $xml .= "<telephone>" . mysql_result($result,$i,2) . "</telephone>";
            $xml .= "<lat>" . mysql_result($result,$i,3) . "</lat>";
            $xml .= "<lon>" . mysql_result($result,$i,4) . "</lon>";
            $xml .= "</ambulance>";
        }
    }
    return $xml;
}
echo '<?xml version="1.0" encoding="utf-8"?>';
echo "<tables>";
echo build_xml('SELECT id,name,surname,latitude,longtitude from doctors WHERE field="" .
$_POST["doctors"] . "" AND available=2;','doctors');
echo build_xml('SELECT id,vehicleplates,telephone,latitude,longtitude from vehicles WHERE
available=2;','ambulances');
echo "</tables>";

```

refreshambtable.php

Το αρχείο αυτό εκτελείται κάθε φορά που πατάμε το κουμπί «Ανανέωση» της καρτέλας «οχήματα». Είναι υπεύθυνο για να ανατρέχει στην βάση δεδομένων και να ανακατασκευάζει τον πίνακα με τα οχήματα ανανεώνοντας την κατάσταση τους την συγκεκριμένη χρονική στιγμή. Χρησιμοποιείται και αυτό με τη βοήθεια της τεχνικής ajax έτσι ώστε η ανανέωση να γίνεται ασύγχρονα.

refreshdoctable.php

Ισχύει ακριβώς ότι και για το refreshambtable.php αλλά για τον πίνακα των γιατρών αυτή τη φορά.

modifyAmbTable.php

Εκτελείται όταν επιλέξουμε να κάνουμε μία ενέργεια όπως η εισαγωγή, τροποποίηση ή διαγραφή ενός οχήματος. Το αρχείο είναι υπεύθυνο για τη σωστή εκτέλεση της ενέργειας που έχει επιλεγεί, την ανανέωση της βάσης με τα καινούρια δεδομένα καθώς και την ανακατασκευή του πίνακα με τα οχήματα με βάση τα καινούρια δεδομένα. Και αυτό το αρχείο χρησιμοποιείται με τη βοήθεια της τεχνικής ajax έτσι ώστε οι ενέργειες αυτές να γίνονται ασύγχρονα.

```
include("mysql.php");
if($_POST["mode"]=="insert")
{
    //eisagwgi asthenoforou sti vasi
    $result = mysql_query("INSERT INTO vehicles VALUES(null,'" . $_POST["name"] . "','" .
$_POST["surname"] . "','" . $_POST["plates"] . "','" . $_POST["telephone"] . "','" . $_POST["password"] .
"',0.0,0.0,'" . $_POST["mobile_ip"] . "','" . 0,'"');
    if (!$result)
        die('Could not query:' . mysql_error());
}
else if($_POST["mode"]=="delete")
{
    //diagrafi asthenoforou apo ti vasi
    $result = mysql_query("DELETE FROM vehicles WHERE id='" . $_POST["id"] . "'");
    if (!$result)
        die('Could not query:' . mysql_error());
}
else if($_POST["mode"]=="update")
{
    $result = mysql_query("UPDATE vehicles SET name='" . $_POST["name"] . "',
surname='" . $_POST["surname"] . "', vehicleplates='" . $_POST["plates"] . "', telephone =" .
$_POST["telephone"] . "', mobile_ip='" . $_POST["mobile_ip"] . "' WHERE id='" . $_POST["id"] . "'");
    if (!$result)
        die('Could not query:' . mysql_error());
}
else
    die('Server Connection Error');
include("refreshambtable.php");
```

modifyDocTable.php

Ισχύει ακριβώς ότι και για το modifyAmbTable.php αλλά για τον πίνακα των γιατρών αυτή τη φορά.

server_socket.php

Εκτελείται στο παρασκήνιο με την εκκίνηση της εφαρμογής. Είναι υπεύθυνο για την επικοινωνία με τους mobile clients και την ενημέρωση της βάσης για την κατάσταση των clients (συνδεδεμένοι, αποσυνδεδεμένοι, επί τω έργω). Ουσιαστικά καθορίζει το πρωτόκολλο επικοινωνίας με τους clients χρησιμοποιώντας τεχνική sockets για την επικοινωνία. Η μορφή του αρχείου φανερώνεται στις παρακάτω γραμμές κώδικα.

```
$host = "localhost";
```



```
$port = 1235;
```

```
$socket = socket_create(AF_INET, SOCK_STREAM, 0) or die("Could not create
socket\n");
$result = socket_bind($socket, $host, $port) or die("Could not bind to
socket\n");
$result = socket_listen($socket, SOMAXCONN) or die("Could not set up socket
listener\n");
$clients_number = array();
while(true)
{
    $spawn = socket_accept($socket) or die("Could not accept incoming connection\n");
    $input = socket_read($spawn, 1024, PHP_NORMAL_READ);
    ...
}
```

Επειδή η php δεν υποστηρίζει multithreading αυτό που κάνουμε είναι να χρησιμοποιήσουμε την τεχνική ajax έτσι ώστε να αφήσουμε το server_socket.php να τρέχει ασύγχρονα στο παρασκήνιο χωρίς να εμποδίζει τη λειτουργία της κεντρικής εφαρμογής. Αυτό το κάνουμε στο <body> του αρχείου index.php προσθέτοντας τις παρακάτω γραμμές κώδικα.

```
<script type="text/javascript">
    $.ajax({
        url: "server_socket.php",
    });
</script>
```

Όταν ένας mobile client επιθυμεί να συνδεθεί με την εφαρμογή στέλνει την λέξη «Login» μαζί με τον αριθμό τηλεφώνου του ή username και τον κωδικό του ο οποίος στέλνεται κωδικοποιημένος για ασφάλεια. Ο κώδικας του αρχείου server_socket.php αναλαμβάνει στη συνέχεια να συνδέσει τον mobile client στην εφαρμογή. Ελέγχει αν ο κωδικός, το τηλέφωνο ή το username και η mobile ip του client ταιριάζουν μεταξύ τους και αντίστοιχα συνδέει η απορρίπτει τον client. Ενδεικτικά παραθέτουμε τον κώδικα για τη διαδικασία login.

```
if(trim($input) == "Login")
{
    $output = "ok";
    socket_write($spawn,$output."\n", strlen($output) + 1) or die("Could not write output\n");
    $input = socket_read($spawn, 1024, PHP_NORMAL_READ);
    sscanf($input,"%s %s\n", $number,$password);
    $result = mysql_query("SELECT password,mobile_ip,name,surname FROM vehicles WHERE
telephone=' . $number . ' UNION SELECT password,mobile_ip,name,surname FROM doctors WHERE
telephone=' . $number . '"); or die("Could not query:" . mysql_error());
    if(mysql_num_rows($result)==1)
    {
        if(sha1(mysql_result($result,0,0))== $password)
        {
            socket_getpeername($spawn,$addr);
            if(mysql_result($result,0,1)==$addr)
            {
                $clients_number[$addr] = $number;
                $output = mysql_result($result,0,2) . " " . mysql_result($result,0,3);
                $result = mysql_query("SELECT 'vehicles' AS tablename FROM
vehicles WHERE telephone=" . $clients_number[$addr] . " UNION SELECT 'doctors' AS tablename FROM
doctors WHERE telephone=" . $clients_number[$addr] . ""); or die("Could not query:" . mysql_error());
```

```

mysql_query('UPDATE ' . mysql_result($result,0,0) . ' SET available =
2 WHERE telephone=' . $clients_number[$addr] . ';' ) or die('Could not query:' . mysql_error());
    }
    else
        $output = "wrong_ip";
    }
    else
        $output = "wrong_password";
}
else
    $output = "wrong_username";
socket_write($spawn,$output."\n", strlen($output) + 1) or die("Could not write output\n");
}

```

Στην περίπτωση που ο client έχει συνδεθεί και είναι έτοιμος να στείλει τη θέση του στον server στέλνει αρχικά τη λέξη «Send». Το αντίστοιχο τμήμα κώδικα του αρχείου αναλαμβάνει αυτήν την ενέργεια και συγκεκριμένα κάνει έλεγχο για το αν ο client που επιθυμεί να στείλει την τοποθεσία του είναι συνδεδεμένος και η mobile ip του συμφωνεί με την ip της σύνδεσης. Αν αυτά ισχύουν τότε στέλνει «ok» στον client και ο client στέλνει την τοποθεσία του στον server ο οποίος με τη σειρά του αναλαμβάνει να ενημερώσει τη βάση.

Όταν τώρα ο client δεχτεί ένα περιστατικό στέλνει τη φράση «Accept_Incident» στον server ο οποίος ελέγχει πάλι αν ο client που στέλνει είναι συνδεδεμένος και η mobile ip του συμφωνεί με την ip της σύνδεσης και ενημερώνει τη βάση για την κατάσταση του client (ongoing).

Όταν ο client φέρει εις πέρας την αποστολή του στέλνει την φράση «End_Of_Incident» στον server ο οποίος ελέγχει πάλι αν ο client που στέλνει είναι συνδεδεμένος και η mobile ip του συμφωνεί με την ip της σύνδεσης και ενημερώνει τη βάση για την κατάσταση του client (online).

Τέλος, όταν ο client θέλει να αποσυνδεθεί στέλνει την λέξη «Logout» στον server ο οποίος ελέγχει πάλι αν ο client που στέλνει είναι συνδεδεμένος και η mobile ip του συμφωνεί με την ip της σύνδεσης και ενημερώνει τη βάση για την κατάσταση του client (offline).

SendIncident.php

Το αρχείο αυτό εκτελείται όταν επιλεγούν οι γιατροί και τα ασθενοφόρα που θα αναλάβουν το περιστατικό και πατηθεί το κουμπί «Επιλογή επιλεγμένου προσωπικού». Είναι υπεύθυνο για να καταχωρήσει τα στοιχεία του περιστατικού στη βάση και για να στείλει το περιστατικό στους επιλεγμένους clients. Αυτό το επιτυγχάνει λαμβάνοντας από τη βάση την mobile ip των επιλεγμένων clients και χρησιμοποιώντας την ip αυτή καθώς και την κοινή πόρτα επικοινωνίας 1236 στέλνει το latitude και longitude καθώς και τον αριθμό διεύθυνσης του περιστατικού για μεγαλύτερη ακρίβεια στους επιλεγμένους clients με τη βοήθεια τεχνικής sockets. Ενδεικτικά το τμήμα κώδικα αποστολής των συντεταγμένων του περιστατικού στο επιλεγμένο ασθενοφόρο.

```

$result = mysql_query('SELECT mobile_ip FROM vehicles WHERE id = ' . $_POST["Ambld"] . ';' );
$AmbIp = mysql_result($result,0,0);

```

```

set_time_limit(0);
ob_implicit_flush();
$PORT = 1236;

```

```

$sock = socket_create(AF_INET, SOCK_STREAM, 0) //Creating a TCP socket
or die("error: could not create socket\n");
$succ = socket_connect($sock, $AmbIp, $PORT) //Connecting to server using that socket
or die("error: could not connect to host\n");
$text = $_POST["Lat"] . "," . $_POST["Lon"] . "," . $_POST["Number"];
socket_write($sock, $text."\n", strlen($text) + 1) //Writing the text to the socket

```

Διαδικτυακό Πληροφοριακό Σύστημα Διαχείρισης Έκτακτων Περιστατικών με Δυνατότητα Πρόσβασης μέσω Κινητών Συσκευών με Λειτουργικό Android

```
or die("error: failed to write to socket\n");
socket_close($sock);
```

ΑΡΧΕΙΑ HTML

dialogs.html

Το αρχείο αυτό περιέχει την εμφάνιση των φορμών και των παραθύρων διαλόγου που εμφανίζονται κατά την επιλογή εισαγωγής, τροποποίησης και διαγραφής γιατρών ή ασθενοφόρων. Οι φόρμες αυτές καλούνται είναι κρυμμένες από την αρχή και φανερώνονται μέσω javascript κώδικα όταν χρειαστούν.

ΑΡΧΕΙΑ JAVASCRIPT

Το κυριότερο αρχείο javascript είναι το template.js το οποίο περιέχει σχεδόν όλες τις javascript λειτουργίες της εφαρμογής.

template.js

Αυτό το αρχείο περιέχει όλες σχεδόν τις λειτουργίες της εφαρμογής. Εδώ περιέχονται όλες οι ενέργειες που πρέπει να γίνουν ασύγχρονα, όλες οι ενέργειες που εκτελούνται όταν πατηθεί ένα κουμπί, όταν επιλεγεί κάτι από ένα πίνακα, όταν επιλεγεί κάτι από το χάρτη, κοινώς περιέχονται όλα τα events που ενεργοποιούνται από τις ενέργειες του χρήστη. Επίσης σε αυτό το αρχείο κατασκευάζονται και οι χάρτες που φαίνονται και στα 3 καρτέλες.

Πιο συγκεκριμένα, όλο το αρχείο είναι ουσιαστικά μία συνάρτηση (ready) της jquery και η οποία ουσιαστικά εκτελείται μόλις φορτώσει η σελίδα της εφαρμογής. Σε αυτή τη συνάρτηση περιλαμβάνονται άλλες συναρτήσεις καθώς και events τα οποία ενεργοποιούνται όταν ο χρήστης κάνει κάποια ενέργεια.

Ενδεικτικά παραθέτουμε κάποιες συναρτήσεις που χρησιμοποιήθηκαν στο αρχείο αυτό. Αρχικά υπάρχει η συνάρτηση disable η οποία αυτό που κάνει είναι να απενεργοποιεί όλα τα πεδία μίας φόρμας. Καλείται όταν υποβάλουμε ένα περιστατικό έτσι ώστε να μην μπορεί ο χρήστης να πειράξει τα στοιχεία του περιστατικού. Η συνάρτηση enable κάνει ακριβώς το αντίθετο ενώ η συνάρτηση clear_form καθαρίζει τη φόρμα από συμπληρωμένα πεδία. Και οι δύο συναρτήσεις χρησιμοποιούνται όταν σταλεί ένα περιστατικό για να καθαρίσει και να ενεργοποιηθεί η φόρμα για εισαγωγή καινούριου περιστατικού. Ο κώδικας των συναρτήσεων φαίνεται παρακάτω.

```
function disable()
{
    var limit = document.forms[0].elements.length;
    for (i=0;i<limit;i++)
        document.forms[0].elements[i].disabled = true;
}

function enable()
{
    var limit = document.forms[0].elements.length;
    for (i=0;i<2;i++)
        document.forms[0].elements[i].disabled = false;
    for (i=3;i<limit;i++)
        document.forms[0].elements[i].disabled = false;
}

function clear_form(table)
{
    $(table).find("tr").each(function()
    {
        $(this).find("input[type!=button]").val("");
    });
}
```

```

    $(this).find("span").html("&nbsp;");
  });
}

```

Η συνάρτηση distance δέχεται δύο σημεία και επιστρέφει την απόσταση τους με βάση τη μαθηματική φόρμουλα Haversine. Η φόρμουλα αυτή είναι ουσιαστικά ένας μαθηματικός τύπος ο οποίος δοσμένων δύο σημείων πάνω σε μία σφαίρα μας επιστρέφει την μεταξύ τους απόσταση. Η συνάρτηση αυτή χρησιμοποιείται για να βρούμε την απόσταση των γιατρών και των ασθενοφόρων από το εκάστοτε περιστατικό και να τους ταξινομήσουμε ανάλογα με την μικρότερη απόσταση.

```

function distance(lat1,lon1,lat2,lon2)
{
  var R = 6371; // km
  var dLat = (lat2-lat1).toRad();
  var dLon = (lon2-lon1).toRad();

  var a = Math.sin(dLat/2) * Math.sin(dLat/2) +
    Math.cos(lat1.toRad()) * Math.cos(lat2.toRad()) *
    Math.sin(dLon/2) * Math.sin(dLon/2);
  var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
  return R * c * 1.609344;
}

```

Η συνάρτηση findDate επιστρέφει την τρέχουσα ημερομηνία και ώρα και χρησιμοποιείται για να συμπληρώσει το πεδίο ώρα κλήσης της αρχικής φόρμας εισαγωγής περιστατικών της καρτέλας περιστατικών.

```

function findDate()
{
  var d = new Date();
  var hours = d.getHours();
  var minutes = d.getMinutes();
  var seconds = d.getSeconds();

  if(hours<10)
    hours='0'+hours;
  if(minutes<10)
    minutes='0'+minutes;
  if(seconds<10)
    seconds='0'+seconds;

  var time = hours+":"+minutes+":"+seconds;
  document.getElementById("time").value = time;

  var day = d.getDate();
  var month = d.getMonth();

  if(day<10)
    day='0'+day;
  if(month<10)
    month='0'+month;

  return d.getFullYear()+"-"+month+"-"+day;
}

```

initialize.js

Το αρχείο αυτό περιέχει συναρτήσεις που ευθύνονται για την αρχική εμφάνιση των καρτελών των οχημάτων και των γιατρών. Συγκεκριμένα περιέχει τέσσερις συναρτήσεις.

Η συνάρτηση markerClick αναλαμβάνει να δείξει τις πληροφορίες της θέσης ενός οχήματος ή ενός γιατρού όταν επιλεγεί στο χάρτη.

```

function markerClick(marker,map) {
  google.maps.event.addListener(marker, 'click', function()
  {

```

```

    var geocoder = new google.maps.Geocoder();

```

```

var infowindow = new google.maps.InfoWindow({size: new google.maps.Size(40,40)});
var latlng = marker.getPosition();

geocoder.geocode({'latLng': latlng}, function(results, status)
{
    if (status == google.maps.GeocoderStatus.OK)
    {
        if(results[0])
        {
            var address_parts = results[0].formatted_address.split(',');
            infowindow.setContent(address_parts[0] + '<br>' +
address_parts[1]);
            infowindow.open(map, marker);
        }
    }
});
}

```

Η συνάρτηση `getGeoCodeResults` δέχεται τις συντεταγμένες ενός σημείου σε latitude και longitude και επιστρέφει τη διεύθυνση της οδού που βρίσκεται το σημείο αυτό. Καλείται όταν ο χρήστης επιλέξει από τον πίνακα των ασθενοφόρων ή των γιατρών ένα ασθενοφόρο/γιατρό ο οποίος έχει αναλάβει κάποιο περιστατικό (κίτρινη ένδειξη) και εμφανίζει τη διεύθυνση του προορισμού δεξιά του πίνακα.

```

function getGeoCodeResults(latlng, callback)
{
    var geocoder = new google.maps.Geocoder();
    geocoder.geocode({'latLng': latlng}, function(results, status)
    {
        if (status == google.maps.GeocoderStatus.OK)
        {
            if(results[0])
                callback(results[0].formatted_address.split(','));
        }
    });
}

```

Η συνάρτηση `initializetab2` είναι υπεύθυνη για την αρχική κατασκευή της καρτέλας των ασθενοφόρων όπως αντίστοιχα είναι η `initializetab3` για την κατασκευή της καρτέλας των γιατρών. Πιο συγκεκριμένα σε αυτές τις συναρτήσεις κατασκευάζονται οι θέσεις των οχημάτων/γιατρών στο χάρτη, υλοποιείται το `highlighting` στους πίνακες και τέλος υλοποιούνται οι ενέργειες που έπονται της επιλογής μίας εγγραφής στον πίνακα.

check_errors.js

Το αρχείο αυτό είναι υπεύθυνο για τον έλεγχο τυχόν λαθών κατά τη συμπλήρωση των φορμών της εφαρμογής.

Οι συναρτήσεις είναι 6 στον αριθμό. Οι 2 πρώτες είναι υπεύθυνες για την φόρμα των περιστατικών, οι επόμενες 2 για τις φόρμες της τροποποίησης και εισαγωγής της καρτέλας των ασθενοφόρων και οι τελευταίες 2 για τις φόρμες της καρτέλας των γιατρών. Επίσης οι συναρτήσεις χωρίζονται σε δύο κατηγορίες: `onfocusout` και `onsubmitclick`. Στην πρώτη κατηγορία γίνεται έλεγχος της φόρμας κατά τη συμπλήρωση της ενώ στη δεύτερη όταν την έχουμε συμπληρώσει και πατάμε υποβολή της φόρμας. Ενδεικτικά παραθέτουμε την συνάρτηση ελέγχου λαθών των φορμών εισαγωγής και τροποποίησης ασθενοφόρων,

```

function form2_check_errors_onfocusout()
{
    //checking errors

    var patt_names = /^[A-Ω Α'Ε'Η'Ι'Ο'Υ'Ω]+[α-ω άέύίόήώύϊϊύ]*$/;
    var patt_plates = /^[A-Ω][A-Ω][A-Ω]-[0-9][0-9][0-9][0-9]$/;
    var patt_telephone = /^[6][9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]$/i;
    var patt_password = /^[A-Z a-z 0-9]*$/;
    var patt_mobile_ip = /^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}$/;
}

```

```

$("#driver_name").focusout(function()
{
    var driver_name = $(this).val();
    if(!driver_name)
        $("#driver_name_error").html("Το πεδίο είναι υποχρεωτικό");
    else if(!patt_names.test(driver_name))
        $("#driver_name_error").html("Μη έγκυρα δεδομένα");
    else
        $("#driver_name_error").html("&nbsp;");
});

$("#driver_surname").focusout(function()
{
    var driver_surname = $(this).val();
    if(!driver_surname)
        $("#driver_surname_error").html("Το πεδίο είναι υποχρεωτικό");
    else if(!patt_names.test(driver_surname))
        $("#driver_surname_error").html("Μη έγκυρα δεδομένα");
    else
        $("#driver_surname_error").html("&nbsp;");
});

$("#plates").focusout(function()
{
    var plates = $(this).val();
    if(!plates)
        $("#plates_error").html("Το πεδίο είναι υποχρεωτικό");
    else if(!patt_plates.test(plates))
        $("#plates_error").html("Μη έγκυρα δεδομένα (XXX-4444)");
    else
        $("#plates_error").html("&nbsp;");
});

$("#driver_telephone").focusout(function()
{
    var telephone = $(this).val();
    if(!telephone)
        $("#driver_telephone_error").html("Το πεδίο είναι υποχρεωτικό");
    else if(!patt_telephone.test(telephone))
        $("#driver_telephone_error").html("Μη έγκυρα δεδομένα (<b>69</b>39159941)");
    else
        $("#driver_telephone_error").html("&nbsp;");
});

$("#driver_password").focusout(function() {
    var password = $(this).val();
    if(!password)
        $("#driver_password_error").html("Το πεδίο είναι υποχρεωτικό");
    else if(password.length<6)
        $("#driver_password_error").html("Επιλέξτε τουλάχιστον 6 χαρακτήρες");
    else if(!patt_password.test(password))
        $("#driver_password_error").html("Μόνο λατινικούς χαρακτήρες και αριθμούς");
    else
        $("#driver_password_error").html("&nbsp;");
});

$("#driver_mobile_ip").focusout(function() {
    var mobile_ip = $(this).val();
    if(!mobile_ip)
        $("#driver_mobile_ip_error").html("Το πεδίο είναι υποχρεωτικό");
    else if(!patt_mobile_ip.test(mobile_ip))
        $("#driver_mobile_ip_error").html("Μη έγκυρα δεδομένα (256.256.256.256)");
    else

```

```

    $("#driver_mobile_ip_error").html("&nbsp;");
  });
}

```

geo_autocomplete.js, jquery.autocomplete_geomod.js, jquery-ui.min.js

Τα αρχεία αυτά αποτελούν ουσιαστικά βιβλιοθήκες οπότε και παραβλέπουμε τον κώδικα τους. Απλά αναφέρουμε ότι τα δύο πρώτα αρχεία αφορούν βιβλιοθήκες για την αυτόματη λίστα με τις διευθύνσεις που εμφανίζεται κατά την πληκτρολόγηση μίας διεύθυνσης στο πεδίο διεύθυνση της φόρμας των περιστατικών ενώ το τελευταίο αφορά τη βιβλιοθήκη JQuery.

ΑΡΧΕΙΑ CSS και ΕΙΚΟΝΕΣ

Για τα αρχεία αυτά δε θα δώσουμε τον κώδικα ή κάποια επεξήγηση καθώς αφορούν καθαρά την εμφάνιση της εφαρμογής και όχι την υλοποίηση.

A.2. Κώδικας προγράμματος εφαρμογής κινητού

Η εφαρμογή του κινητού είναι υλοποιημένη με τη βοήθεια της γλώσσας προγραμματισμού java και τη βοήθεια αρχείων xml, για την πλατφόρμα Android. Η εφαρμογή αποτελείται από αρχεία .java και αρχεία xml. Πιο συγκεκριμένα αποτελείται από:

12 .java αρχεία

- eEKAB.java
- GoogleParser.java
- Markers.java
- MyMap.java
- MyService.java
- Parser.java
- ReadIncident.java
- Route.java
- RouteOverlay.java
- Segment.java
- WrapMessage.java
- XMLParser.java

1 φάκελο drawable με τις εικόνες της εφαρμογής 5 στον αριθμό:

- confirm.png
- error.png
- green_dot.png
- notification.png
- quit_button.png

1 φάκελο layout με 5 αρχεία xml

- finalview.xml
- main.xml
- map.xml
- response.xml
- timerview.xml

1 φάκελο menu με 1 αρχείο xml

- menu.xml

1 φάκελο values με 2 αρχεία xml

- colors.xml
- strings.xml

Το αρχείο AndroidManifest.xml

eEkab.java

Το αρχείο αυτό είναι το πρώτο που εκτελείται κατά την εκκίνηση της εφαρμογής (το αρχικό αρχείο έχει δηλωθεί στο αρχείο AndroidManifest.xml). Περιλαμβάνει τη συνάρτηση onCreate η οποία εκτελείται όταν αρχίζει η Activity eEkab. Η συνάρτηση αυτή καλεί το αρχείο main.xml το οποίο είναι υπεύθυνο για την αρχική εμφάνιση της εφαρμογής. Η αρχική μορφή της εφαρμογής αποτελείται από ένα τίτλο, 2 πεδία ένα για το όνομα χρήστη και ένα για τον κωδικό και ένα κουμπί.

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
}
```

Από τη στιγμή που ο χρήστης συμπληρώσει τα πεδία και πατήσει το κουμπί καλείται η ConnectToServer που είναι υπεύθυνη για να συνδέσει το χρήστη με το σύστημα. Η συνάρτηση λαμβάνει τα στοιχεία που συμπλήρωσε ο χρήστης και κωδικοποιεί των κωδικό του κωδικοποίηση SHA1. Στη συνέχεια ανοίγει ένα socket για να επικοινωνήσει με τον server, του στέλνει την λέξη «Login» και περιμένει τη λέξη «ok». Αν δεν λάβει το ok τότε ειδοποιεί το χρήστη με ένα αντίστοιχο μήνυμα. Αν ο server στείλει το ok τότε η συνάρτηση στέλνει στον server το τηλέφωνο και τον κωδικό κωδικοποιημένο και περιμένει την απάντηση του server. Ο server ή θα στείλει μήνυμα για λάθος κωδικό οπότε και η συνάρτηση ConnectToServer το γνωστοποιεί στο χρήστη με το αντίστοιχο μήνυμα ή θα στείλει μήνυμα για λάθος τηλέφωνο ή για λάθος ip οπότε και το μήνυμα προς το χρήστη θα είναι αντίστοιχο. Σε αντίθετη περίπτωση που όλα είναι σωστά ο server θα στείλει το όνομα του χρήστη από τη βάση και η συνάρτηση θα γνωστοποιήσει στον χρήστη ότι η σύνδεση επετεύχθη. Στη συνέχεια εμφανίζει στο χρήστη ένα μήνυμα το οποίο μετά από 10 δευτερόλεπτα κλείνει για να συνεχίσει πλέον η εφαρμογή να τρέχει στο παρασκήνιο. Είναι καθαρά ένα μήνυμα που πληροφορεί το χρήστη για την επιτυχή σύνδεση και για την εκτέλεση της εφαρμογής πλέον στο παρασκήνιο. Η εμφάνιση του μηνύματος αυτού πραγματοποιείται από το αρχείο timerview.xml. Όταν περάσουν τα 10 δευτερόλεπτα καλείται η συνάρτηση OnFinish η οποία ξεκινάει ένα service (MyService.java) το οποίο θα είναι υπεύθυνο για την εκτέλεση της εφαρμογής στο background.

Επίσης να επισημάνουμε ότι όλα τα μηνύματα εμφανίζονται με τη μορφή παραθύρων διαλόγου τα οποία κατασκευάζονται από τη συνάρτηση onCreateDialog.

```
@Override
protected Dialog onCreateDialog(int id) {
    LayoutInflater inflater = LayoutInflater.from(this);
    View responseView = inflater.inflate(R.layout.response, (ViewGroup) findViewById(R.id.root));
    TextView text = (TextView) responseView.findViewById(R.id.text);
    ImageView image = (ImageView) responseView.findViewById(R.id.image);
    switch(id)
    {
        case CONNECTION_SUCCESSFUL:
            text.setText("Σύνδεση Επιτυχής!");
            image.setImageResource(R.drawable.confirm);
            return new AlertDialog.Builder(eEKAB.this)
                .setView(responseView)
                .create();
        case WRONG_PASSWORD:
            image.setImageResource(R.drawable.error);
            text.setText("Λάθος Κωδικός");
            return new AlertDialog.Builder(eEKAB.this)
                .setView(responseView)
                .create();
        case WRONG_USERNAME:
            image.setImageResource(R.drawable.error);
            text.setText("Λάθος Αριθμός");
            return new AlertDialog.Builder(eEKAB.this)
                .setView(responseView)
                .create();
        case WRONG_IP:
            image.setImageResource(R.drawable.error);
            text.setText("Η ip σας δεν ταιριάζει");
            return new AlertDialog.Builder(eEKAB.this)
                .setView(responseView)
                .create();
    }
}
```



```

        .setView(responseView)
        .create();
    case CANNOT_CONNECT_TO_SERVER:
        image.setImageResource(R.drawable.error);
        text.setText("Υπήρξε πρόβλημα στη σύνδεση");
        return new AlertDialog.Builder(eEKAB.this)
            .setView(responseView)
            .create();
    }
    return null;
}

```

MyService.java

Αρχικά με την εκκίνηση του service εκτελείται η συνάρτηση onCreate. Μέσα στη συνάρτηση αυτή δημιουργείται το notification (το μικρό θαυμαστικό πάνω αριστερά που φανερώνει την εκτέλεση της εφαρμογής στο παρασκήνιο), ξεκινάει η ανανέωση της τοποθεσίας της κινητής συσκευής η οποία γίνεται κάθε λεπτό ή κάθε φορά που ο κινητός client μετακινηθεί σε απόσταση μεγαλύτερη του 1 χλμ. Στη συνέχεια καλείται για πρώτη φορά η dumpLocation η οποία με τη σειρά της καλεί τη socketConnection η οποία πραγματοποιεί σύνδεση με τον server και στέλνει την τοποθεσία του κινητού client. Τέλος, καλείται μία ασύγχρονη διεργασία η οποία θα τρέχει στο παρασκήνιο μαζί με την εφαρμογή και η οποία θα περιμένει απλά για να δεχθεί κάποιο περιστατικό. Αυτή η ασύγχρονη διεργασία βρίσκεται στο αρχείο ReadIncident.java.

```

@Override
public void onCreate(){
    mNotificationManager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
    showNotification();
    mgr = (LocationManager) getSystemService(LOCATION_SERVICE);
    Criteria criteria = new Criteria();
    best = mgr.getBestProvider(criteria, true);
    mgr.requestLocationUpdates(best, 60000, 1000, this);
    Location location = mgr.getLastKnownLocation(best);
    dumpLocation(location);
    mtask = new ReadIncident(getApplicationContext(), mNotificationManager);
    mtask.execute(null);
}

private void showNotification() {
    Notification notification = new Notification(R.drawable.notification,
    "eEKAB", System.currentTimeMillis());
    notification.flags = Notification.FLAG_ONGOING_EVENT;
    PendingIntent contentIntent = PendingIntent.getActivity(this, 0, new Intent(this,
    WrapMessage.class), 0);
    notification.setLatestEventInfo(this, "eEKAB", "", contentIntent);
    mNotificationManager.notify(NOTIFICATION_ID, notification);
}

```

WrapMessage.java

Αυτό το αρχείο εκτελείται όταν ο χρήστης ανοίξει την ειδοποίηση που δημιουργείται από την onCreate του service.java αρχείου. Η πρώτη συνάρτηση που θα εκτελεστεί θα είναι η onCreate η οποία θα καλέσει το αρχείο finalview.xml για την εμφάνιση της καινούριας σελίδας.

```

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

    LayoutInflater inflater = LayoutInflater.from(WrapMessage.this);
    View finalView = inflater.inflate(R.layout.finalview, (ViewGroup) findViewById(R.id.final_root));
    final TextView finalText = (TextView) finalView.findViewById(R.id.final_text);
    finalText.setText("Η εφαρμογή eEKAB εκτελείται στο παρασκήνιο. Για έξοδο από την εφαρμογή πατήστε το κουμπί "Εξοδος");
    setContentView(finalView);
}

```

}
 Η εμφάνιση της σελίδας αυτής αποτελείται από ένα κείμενο και ένα κουμπί «Έξοδος». Όταν το κουμπί πατηθεί τότε εκτελείται η συνάρτηση ExiteEKAB. Αυτή η συνάρτηση κλείνει την εφαρμογή και στέλνει ειδοποίηση στον server ότι ο συγκεκριμένος client αποσυνδέθηκε.

```
public void ExiteEKAB(View Button)
```

```
{
    finish();
    try {
        Socket s = new Socket("10.0.2.2",1235);
        BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(s.getOutputStream()));
        bw.write("Logout\n");
        bw.flush();
        bw.close();
        s.close();
    } catch (UnknownHostException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    Intent intent = new Intent(WrapMessage.this, MyService.class);
    stopService(intent);
}
```

ReadIncident.java

Το αρχείο που είναι υπεύθυνο για να λάβει ειδοποίηση καινούριου περιστατικού από τον server. Περιλαμβάνει τη συνάρτηση doInBackground η οποία απλά περιμένει για να δεχθεί κάποιο περιστατικό. Μόλις φτάσει κάποιο περιστατικό από το server στέλνεται στον client η τοποθεσία του περιστατικού καθώς και ο αριθμός της οδού.

```
@Override
```

```
protected String[] doInBackground(Void... arg0) {
```

```
    String[] Latlng = null;
    try
    {
        int port = 1236;
        ServerSocket listenSock = null;
        Socket sock = null;

        listenSock = new ServerSocket(port);
        listenSock.setSoTimeout(0);
        sock = listenSock.accept();

        BufferedReader br = new BufferedReader(new
InputStreamReader(sock.getInputStream()));
        String line = "";
        line = br.readLine();

        Latlng = line.split(",");

        br.close();
        sock.close();
        listenSock.close();
    }
    catch (IOException e){
        e.printStackTrace();
    }
    return Latlng;
}
```

Στη συνέχεια εκτελείται η συνάρτηση onPostExecute. Αυτή η συνάρτηση εμφανίζει μία καινούρια ειδοποίηση με τίτλο «New Incident» την οποία αν επιλέξει ο χρήστης εκτελείται στη συνέχεια το αρχείο MyMap.java που θα δούμε στη συνέχεια. Παράλληλα ενημερώνει τον server

ότι αποδέχτηκε το περιστατικό και τερματίζει το service που έτρεχε στο παρασκήνιο για να μην στέλνει πλέον ο client στον server τη τοποθεσία αφού πλέον δεν υπάρχει λόγος.

`@Override`

```
protected void onPostExecute(String[] Latlng) {
    if(Latlng==null)
        Log.i("IOException", "Ok");
    else
    {
        Intent intent = new Intent(context , MyMap.class);
        intent.putExtra("Latlng", Latlng);
        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        Notification notification = new Notification(R.drawable.notification, "New Incident", System.currentTimeMillis());
        notification.defaults |= Notification.DEFAULT_SOUND;
        notification.flags = Notification.FLAG_ONGOING_EVENT;
        PendingIntent contentIntent = PendingIntent.getActivity(context, 0, intent, 0);
        notification.setLatestEventInfo(context, "New Incident", "An incident arrived", contentIntent);
        mNotificationManager.notify(NOTIFICATION_ID, notification);
        try {
            Socket s = new Socket("10.0.2.2", 1235);
            BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(s.getOutputStream()));
            bw.write("Accept_Incident\n");
            bw.flush();

            bw.close();
            s.close();
        } catch (UnknownHostException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        Intent i = new Intent(context, MyService.class);
        context.stopService(i);
    }
}
```

MyMap.java

Το αρχείο αυτό αρχίζει να εκτελείται από τη στιγμή που ο χρήστης επιλέξει την ειδοποίηση «new Incident» που θα εμφανιστεί στο κινητό του. Αρχικά εκτελείται η συνάρτηση onCreate η οποία καλεί το αρχείο map.xml για το σχεδιασμό της εμφάνισης της καινούριας σελίδας καθώς και 2 άλλες συναρτήσεις την initView και initRoute τις οποίες θα αναλύσουμε παρακάτω.

`@Override`

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.map);
    initView();
    initRoute();
}
```

Η σελίδα αποτελείται από ένα κείμενο το οποίο είναι η αφηγηρία και ο προορισμός του client καθώς και ένα χάρτη.

Η initView το μόνο που κάνει είναι να ξεκινάει τον εντοπισμό θέσης του κινητού ανά 3sec ή όταν ο client μετακινηθεί σε απόσταση μεγαλύτερη των 100m.

```
private void initView() {
    mapView = (MapView) findViewById(R.id.mapview);
    controller = mapView.getController();
    mapView.setSatellite(true);
    mapView.setBuiltInZoomControls(true);
```

```
LocationManager mgr = (LocationManager) getSystemService(LOCATION_SERVICE);
```

```

Criteria criteria = new Criteria();
String best = mgr.getBestProvider(criteria, true);
mgr.requestLocationUpdates(best, 3000, 100, this);
}

```

Η `initRoute` βρίσκει την αρχική θέση του κινητού και καλεί την `initIncident` για να σχεδιάσει και την πορεία μέχρι το περιστατικό.

```

private void initRoute() {
    overlay = new MyLocationOverlay(this, mapView);
    overlay.enableMyLocation();
    overlay.runOnFirstFix(new Runnable() {
        public void run() {

            fromLat = (double)overlay.getMyLocation().getLatitudeE6() / 1E6;
            fromLon = (double)overlay.getMyLocation().getLongitudeE6() / 1E6;
            initIncident();
        }
    });
}

```

Επειδή η Google δεν έχει απευθείας υπηρεσία routing για κινητά Android όπως στους google maps ακολουθούμε κάποια άλλη διαδικασία για να το πετύχουμε. Κάνουμε μία κλήση σε ένα web service των google maps το οποίο δέχεται σαν «είσοδο» την αφετηρία και τον προορισμό και επιστρέφει σε ένα αρχείο json (κάτι αντίστοιχο του xml) την ενδιάμεση διαδρομή. Στη συνέχεια επεξεργαζόμαστε το αρχείο με έναν json parser και εμφανίζουμε στον χάρτη ένα πλήθος σημείων που ουσιαστικά είναι η διαδρομή που θέλουμε σαν να το βλέπαμε σε ένα google map. Για να το επιτύχουμε αυτό χρησιμοποιήσαμε ουσιαστικά τον κώδικα που υπάρχει στο παρακάτω σύνδεσμο: <http://stackoverflow.com/questions/11745314/why-retrieving-google-directions-for-android-using-kml-data-is-not-working-anymo/11745316#11745316> ο οποίος μας βοηθάει να υλοποιήσουμε την υπηρεσία routing. Σύμφωνα με τον κώδικα αυτό δημιουργήσαμε 6 αρχεία τα `Parser.java`, `XMLParser.java`, `GoogleParser.java`, `Segment.java`, `Route.java`, `RouteOverlay.java` τα οποία είναι υπεύθυνα για την επεξεργασία του αρχείου json και την αποθήκευση των στοιχείων του αρχείου σε κλάσεις για την χρήση τους στην εφαρμογή.

Η `initIncident` καλεί τη συνάρτηση `directions` η οποία κατασκευάζει το url, με την αφετηρία και τον προορισμό, που είναι απαραίτητο για το web service της google.

```

private Route directions(final GeoPoint start, final GeoPoint dest, String number)
{
    Parser parser;
    String jsonURL = "http://maps.google.com/maps/api/directions/json?";
    final StringBuffer sBuf = new StringBuffer(jsonURL);
    sBuf.append("origin=");
    sBuf.append(start.getLatitudeE6()/1E6);
    sBuf.append(',');
    sBuf.append(start.getLongitudeE6()/1E6);
    sBuf.append("&destination=");
    sBuf.append(dest.getLatitudeE6()/1E6);
    sBuf.append(',');
    sBuf.append(dest.getLongitudeE6()/1E6);
    sBuf.append("&sensor=true&mode=driving");
    parser = new GoogleParser(sBuf.toString());
    Route r = parser.parse(number);
    return r;
}

```

Στη συνέχεια χρησιμοποιεί ένα handler για να σχεδιάσει την διαδρομή. Τέλος έχουμε και δύο συναρτήσεις την `onCreateOptionsMenu` και την `OnOptionsItemSelected` οι οποίες είναι υπεύθυνες για τη δημιουργία του menu. Η πρώτη σχεδιάζει τη μορφή του με βάση το αρχείο `menu.xml` ενώ η δεύτερη, όταν ο χρήστης πατήσει το κουμπί menu του κινητού, εμφανίζει ένα παράθυρο διαλόγου, με τη βοήθεια της συνάρτησης `onCreateDialog`, που ρωτάει το χρήστη εάν θέλει όντως να τερματίσει την εφαρμογή. Εάν ναι τότε ο client ενημερώνει το server για τον τερματισμό του περιστατικού και την επαναφορά της εφαρμογής στην κατάσταση πριν την αποδοχή του περιστατικού.

B. Σχεδίαση Βάσης Δεδομένων

Η βάση δεδομένων της εφαρμογής είναι αρκετά απλή και αποτελείται από 3 πίνακες: doctors, vehicles, incidents. Στο πρώτο αποθηκεύονται τα στοιχεία των γιατρών, στον δεύτερο τα στοιχεία των ασθενοφόρων και στον τρίτο τα στοιχεία των περιστατικών.

doctors

```
CREATE TABLE IF NOT EXISTS `doctors` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(20) COLLATE utf8_unicode_ci NOT NULL,  
  `surname` varchar(20) COLLATE utf8_unicode_ci NOT NULL,  
  `field` varchar(20) COLLATE utf8_unicode_ci NOT NULL,  
  `telephone` varchar(10) COLLATE utf8_unicode_ci NOT NULL,  
  `password` varchar(10) COLLATE utf8_unicode_ci NOT NULL,  
  `latitude` decimal(9,7) NOT NULL,  
  `longitude` decimal(9,7) NOT NULL,  
  `mobile_ip` varchar(15) COLLATE utf8_unicode_ci NOT NULL,  
  `available` int(3) NOT NULL,  
  `address` varchar(40) COLLATE utf8_unicode_ci NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
AUTO_INCREMENT=16 ;
```

incidents

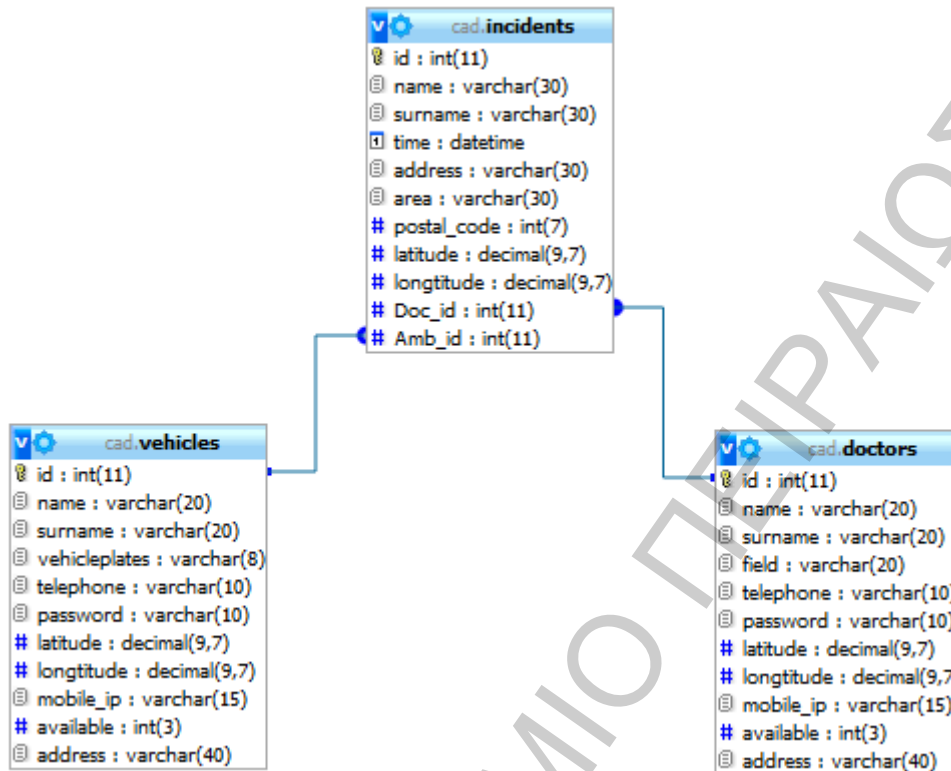
```
CREATE TABLE IF NOT EXISTS `incidents` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(30) NOT NULL,  
  `surname` varchar(30) NOT NULL,  
  `time` datetime NOT NULL,  
  `address` varchar(30) NOT NULL,  
  `area` varchar(30) NOT NULL,  
  `postal_code` int(7) NOT NULL,  
  `latitude` decimal(9,7) NOT NULL,  
  `longitude` decimal(9,7) NOT NULL,  
  `Doc_id` int(11) NOT NULL,  
  `Amb_id` int(11) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=17 ;
```

vehicles

```
CREATE TABLE IF NOT EXISTS `vehicles` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(20) COLLATE utf8_unicode_ci NOT NULL,  
  `surname` varchar(20) COLLATE utf8_unicode_ci NOT NULL,  
  `vehicleplates` varchar(8) COLLATE utf8_unicode_ci NOT NULL,  
  `telephone` varchar(10) COLLATE utf8_unicode_ci NOT NULL,  
  `password` varchar(10) COLLATE utf8_unicode_ci NOT NULL,  
  `latitude` decimal(9,7) NOT NULL,  
  `longitude` decimal(9,7) NOT NULL,  
  `mobile_ip` varchar(15) COLLATE utf8_unicode_ci NOT NULL,  
  `available` int(3) NOT NULL,  
  `address` varchar(40) COLLATE utf8_unicode_ci NOT NULL,  
  PRIMARY KEY (`id`)
```

) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
 AUTO_INCREMENT=136 ;

Οι πίνακες φαίνονται και σχηματικά και στην παρακάτω εικόνα.



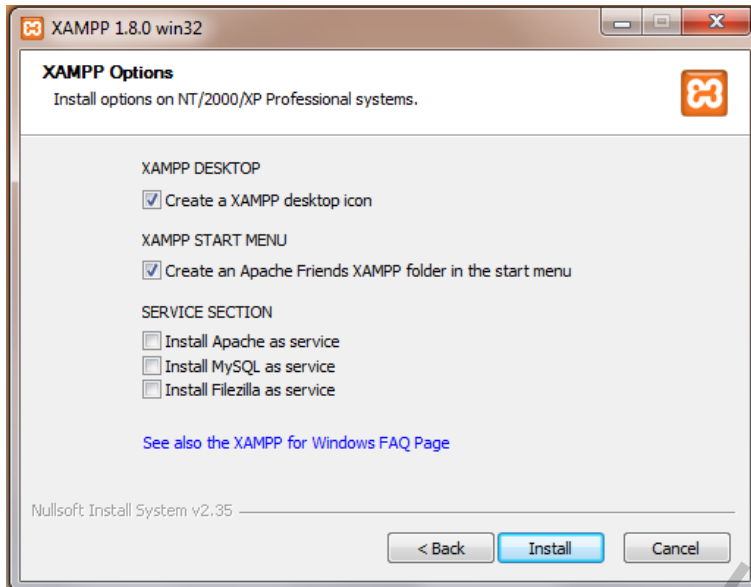
Εικόνα 43: Σχηματική αναπαράσταση της βάσης δεδομένων της εφαρμογής

Γ. Οδηγίες εγκατάστασης

Γ.1. Κεντρική εφαρμογή

Η κεντρική εφαρμογή για να λειτουργήσει χρειάζεται απλά έναν browser firefox 3.6 και πάνω. Στην ιστοσελίδα του Mozilla firefox <http://www.mozilla.org/en-US/firefox/central/> είναι διαθέσιμη η τελευταία έκδοση του δημοφιλή browser. Όπως κάθε web application φυσικά χρειάζεται η εφαρμογή να ανέβει σε κάποιο server που θα υποστηρίζει php και mysql. Στην περίπτωση μας χρησιμοποιήσαμε τον XAMPP server ο οποίος είναι διαθέσιμος από τη διεύθυνση <http://www.apachefriends.org/en/xampp-windows.html>. Επιλέγουμε να κατεβάσουμε τον Installer για πιο εύκολη εγκατάσταση. Μόλις το κατεβάσουμε ανοίγουμε το αρχείο που κατέβηκε και στη συνέχεια ακολουθούμε τις οδηγίες που δίνονται παρακάτω:

1. Στο πρώτο παράθυρο διαλόγου επιλέγουμε next
2. Στο δεύτερο επιλέγουμε την τοποθεσία αποθήκευσης του XAMPP και επιλέγουμε next
3. Στο τρίτο παράθυρο επιλέγουμε τις συντομεύσεις που επιθυμούμε καθώς επίσης και αν επιθυμούμε να χρησιμοποιήσουμε τον Apache και τη MySQL σαν υπηρεσίες. Αυτό σημαίνει ότι ουσιαστικά με την έναρξη του λειτουργικού συστήματος ο Apache και η MySQL θα τρέχουν στο background. Στην παρούσα εργασία δεν επιλέγουμε αυτή τη δυνατότητα



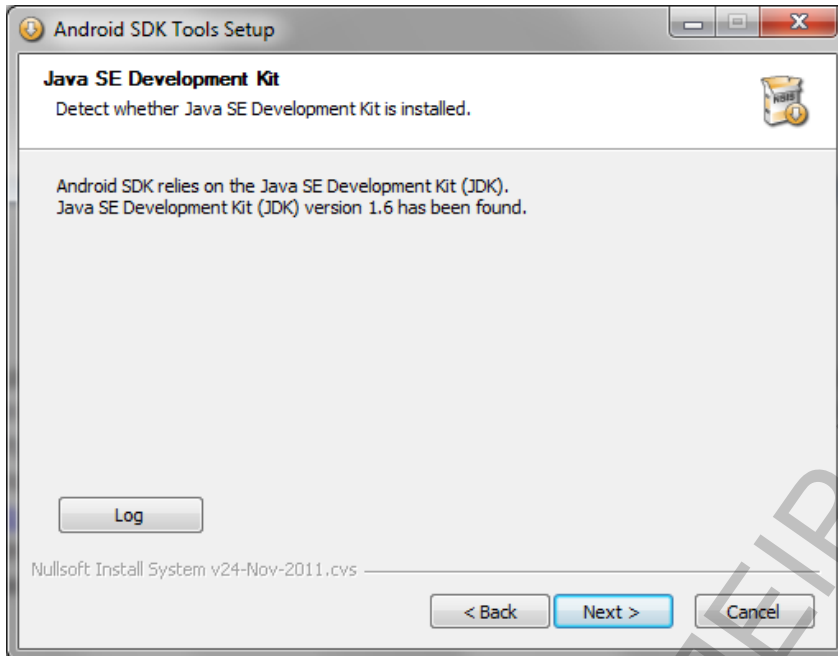
Εικόνα 44: Παράθυρο διαλόγου της εγκατάστασης του XAMPP για επιλογή συντομεύσεων και εκτέλεσης του Apache και της MySQL σαν υπηρεσιών

4. Στη συνέχεια επιλέγουμε Install.

Με την εγκατάσταση του xampp αντιγράφουμε το φάκελο eEKAB που περιέχει τα αρχεία της server μεριάς της εφαρμογής στον φάκελο xampp\htdocs. Επίσης αντιγράφουμε και τα αρχεία του φακέλου cad στο φάκελο xampp\mysql\data.

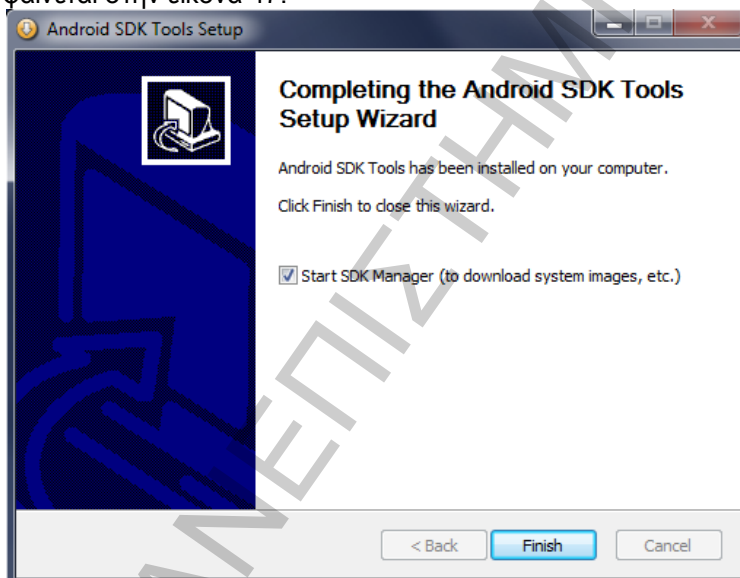
Γ2. Mobile εφαρμογή

Για να υλοποιήσουμε την εφαρμογή χρησιμοποιήσαμε την πλατφόρμα eclipse η οποία διαθέτει emulator του android. Η πλατφόρμα eclipse μπορεί να κατέβει από τη διεύθυνση <http://www.eclipse.org/downloads/>. Οι εκδόσεις που είναι συμβατές με android είναι οι Eclipse IDE for Java Developers, Eclipse IDE for Java EE Developers καθώς και η Eclipse Classic (3.5 και πάνω). Στη συνέχεια πρέπει να κατεβάσουμε ένα java development kit (JDK 5 and above) το οποίο είναι διαθέσιμο από τη σελίδα <http://www.oracle.com/technetwork/java/javase/downloads/index.html>. Καλό θα ήταν να συμπληρώσετε στην μεταβλητή περιβάλλοντος path το μονοπάτι που οδηγεί στην εγκατάσταση του jdk. Στα windows 7 αυτό γίνεται ως εξής: Ο υπολογιστής μου -> ιδιότητες και στη συνέχεια επιλέγουμε Ρυθμίσεις συστήματος για προχωρημένους. Από εκεί επιλέγουμε Μεταβλητές περιβάλλοντος και στις Μεταβλητές συστήματος βρίσκουμε τη μεταβλητή path και συμπληρώνουμε στο τέλος το μονοπάτι μέχρι την εγκατάσταση του jdk. Στη συνέχεια πρέπει να εγκατασταθεί το Android SDK Starter Package το οποίο μπορεί να κατέβει από την url: <http://developer.android.com/sdk/index.html>. Καλό είναι να εγκατασταθεί το Windows Installer καθώς είναι πιο εύκολο στην εγκατάσταση. Μόλις εκκινήσουμε το Windows Installer γίνεται ένα scan για εύρεση του jdk.

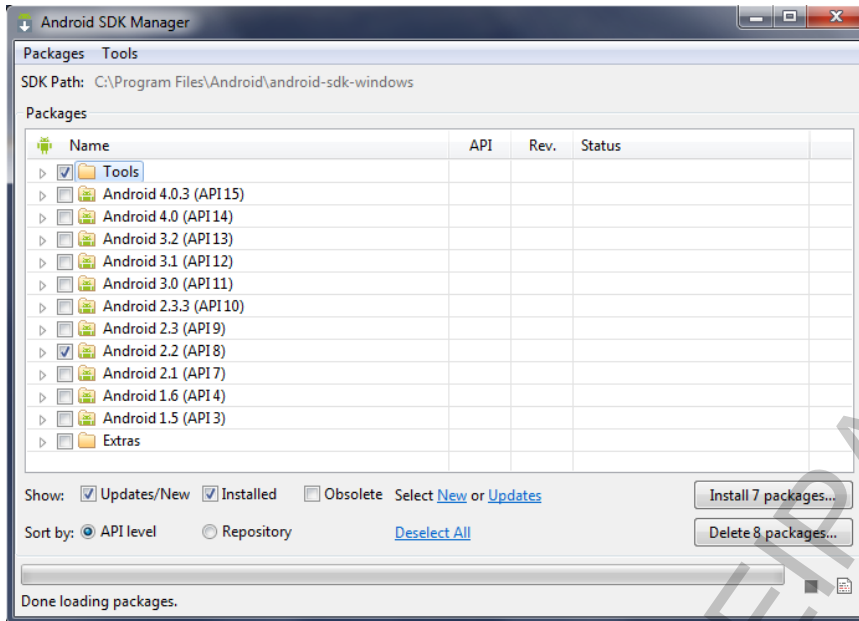


Εικόνα 45: Scan από το Wizard της εγκατάστασης του Android SDK για εύρεση του JDK

Όταν ολοκληρώσουμε την εγκατάσταση επιλέγουμε finish (εικόνα 46) με τσεκαρισμένη την επιλογή Start SDK Manager. Με το τέλος της εγκατάστασης ανοίγει ο Android SDK Manager στον οποίο επιλέγουμε τα api που θέλουμε να εγκαταστήσουμε. Για το project μας αρκεί να κλικάρουμε και να εγκαταστήσουμε το φάκελο tools καθώς και το Android 2.2 (API 8) όπως φαίνεται στην εικόνα 47.



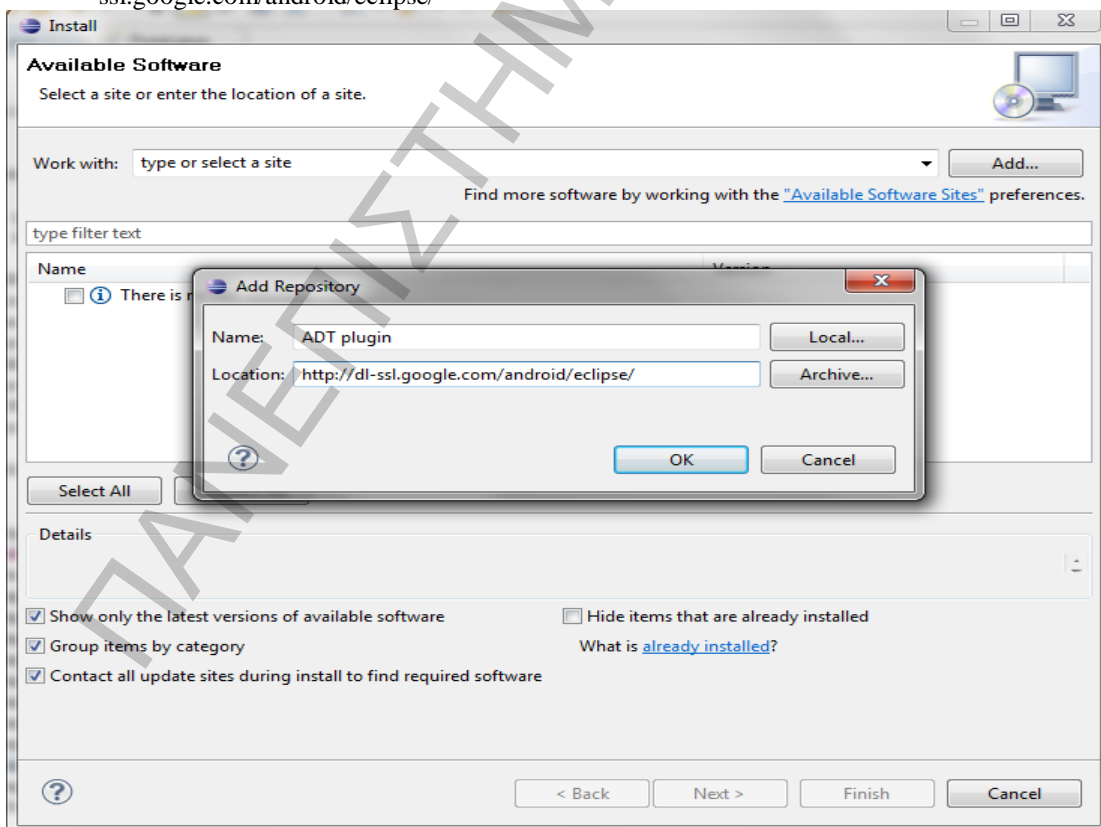
Εικόνα 46: Τέλος εγκατάστασης του Android SDK Starter Package



Εικόνα 47: Επιλογή εγκατάστασης api από τον Android SDK Manager

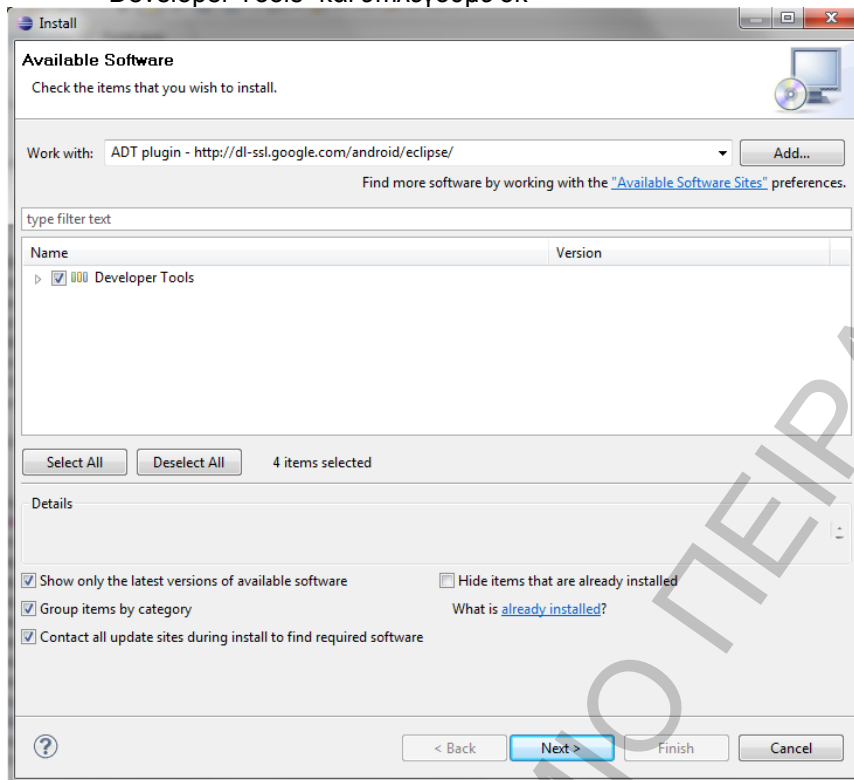
Η εγκατάσταση μπορεί να ολοκληρωθεί και μέσω του eclipse επιλέγοντας window > Android SDK manager. Στη συνέχεια πρέπει να κάνουμε εγκατάσταση του ADT (Android Development Tools) plugin. Αυτό επιτυγχάνεται ως εξής:

1. Ανοίγουμε το eclipse και επιλέγουμε Help > Install New Software....
2. Επιλέγουμε Add στην πάνω δεξιά γωνία.
3. Στο παράθυρο διαλόγου Add Repository που εμφανίζεται, εισάγουμε «ADT plugin» στο πεδίο Name και την παρακάτω url στο πεδίο Location: <http://dl-ssl.google.com/android/eclipse/>



Εικόνα 48: Εγκατάσταση του ADT plugin

4. Επιλέγουμε ok
5. Στο παράθυρο διαλόγου Available Software, επιλέγουμε το checkbox δίπλα από το “Developer Tools” και επιλέγουμε ok



Εικόνα 49: Εγκατάσταση του ADT plugin (2)

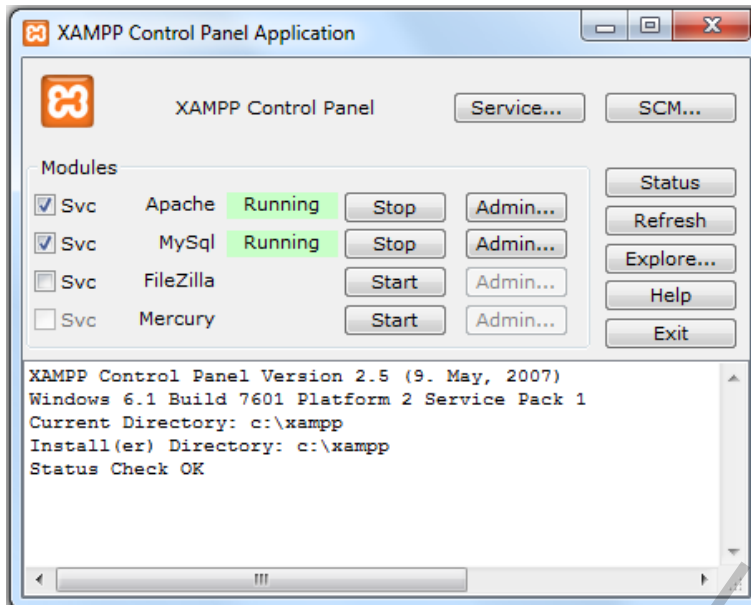
6. Στο επόμενο παράθυρο παρατηρούμε διάφορα εργαλεία που θα κατεβούν. Επιλέγουμε Next.
7. Διαβάζουμε και δεχόμαστε τους όρους της άδειας και επιλέγουμε Finish
8. Όταν η εγκατάσταση ολοκληρωθεί, επανεκκινούμε το Eclipse.
Τέλος, στο Eclipse επιλέγουμε File > Import... και από το φάκελο General επιλέγουμε Existing Projects into Workspace. Από εκεί βρίσκουμε το project eEkab και το κάνουμε import.

Δ. Διαδικασία εκκίνησης της εφαρμογής e-EKAB

Για την εκκίνηση της εφαρμογής απαιτούνται κάποιες ενέργειες οι οποίες είναι αναγκαίες λόγω της ιδιότητας της προσομοίωσης που παρουσιάζει η εν λόγω εφαρμογή. Η προσομοίωση χωρίζεται σε δύο μέρη: στην προσομοίωση του Web Server και στη προσομοίωση του κινητού Android.

Δ1. Web Server

Στην μεριά του web server το μόνο που χρειάζεται είναι το XAMPP. Ανοίγουμε το XAMPP control panel. Στον πίνακα ελέγχου του XAMPP επιλέγουμε ποια στοιχεία θα εκκινήσουμε. Στην περίπτωση μας θέλουμε να εκκινήσουμε τον Apache και την MySQL. Οπότε πατάμε το κουμπί start που βρίσκεται δίπλα στα στοιχεία που προαναφέραμε και περιμένουμε να εμφανιστεί το status “Running” σε κάθε στοιχείο όπως φαίνεται παρακάτω. Σε αυτή την περίπτωση ο Apache και η MySQL τρέχουν στο παρασκήνιο και μπορούν να χρησιμοποιηθούν για να προσομοιώσουμε τον Web Server που χρειάζομαστε.



Εικόνα 50: Πίνακας ελέγχου του XAMPP

Μόλις τελειώσει και αυτό το βήμα το μόνο που μένει είναι να κλείσουμε τον πίνακα ελέγχου και να ανοίξουμε τον Mozilla Firefox και να πληκτρολογήσουμε τη διεύθυνση localhost/eEKAB στη γραμμή διευθύνσεων.

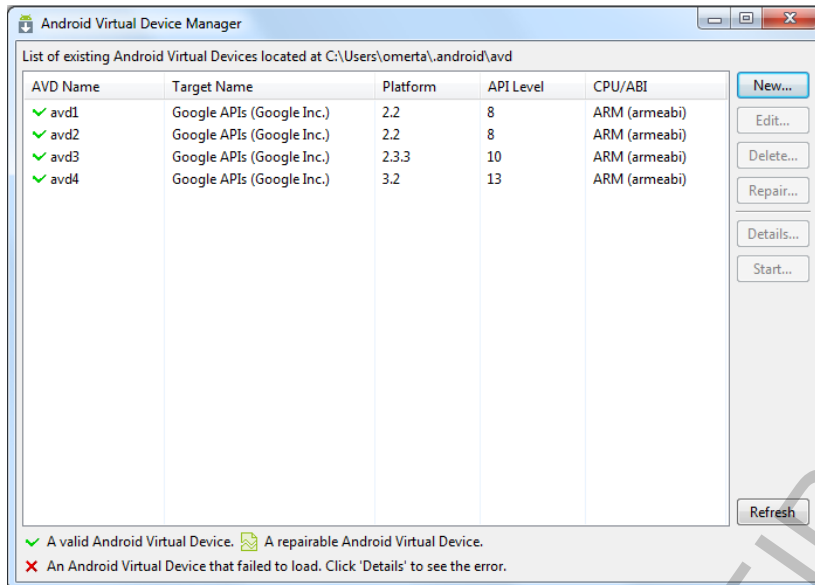
Δ2. Κινητό Android

Απόκτηση κλειδιού για google maps

Για να μπορούμε να έχουμε google maps στην εφαρμογή του κινητού είναι απαραίτητο να αποκτήσουμε ένα κλειδί από τη google. Για να γίνει αυτό θα πρέπει να υπογράψουμε ψηφιακά την εφαρμογή μας. Ευτυχώς αυτό το κάνει το eclipse για μας την πρώτη φορά που εκτελούμε την εφαρμογή. Δημιουργεί ένα αρχείο το "debug.keystore" το οποίο βρίσκεται συνήθως στην τοποθεσία "C:\Users\USER\.android". Στη συνέχεια θα πρέπει να εξάγουμε το MD5 αποτύπωμα του πιστοποιητικού από αυτό το αρχείο. Για να το επιτύχουμε αυτό θα πρέπει να χρησιμοποιήσουμε την εφαρμογή keytool η οποία παρέχεται με κάθε έκδοση της JDK και βρίσκεται συνήθως στην τοποθεσία "%JAVA_HOME%\bin". Για μεγαλύτερη ευκολία προσθέτουμε το μονοπάτι στο οποίο βρίσκεται η εφαρμογή keytool στην μεταβλητή περιβάλλοντος PATH. Στη συνέχεια ανοίγουμε ένα command prompt και πληκτρολογούμε την εντολή "keytool -list -keystore .android/debug.keystore". Αυτή η εντολή μας δίνει το αποτύπωμα MD5 το οποίο το χρησιμοποιούμε στην σελίδα <https://developers.google.com/maps/documentation/android/v1/maps-api-signup> για να αποκτήσουμε το κλειδί. Τέλος το κλειδί αυτό το αντιγράφουμε στο αρχείο map.xml στη περιοχή MapKEY.

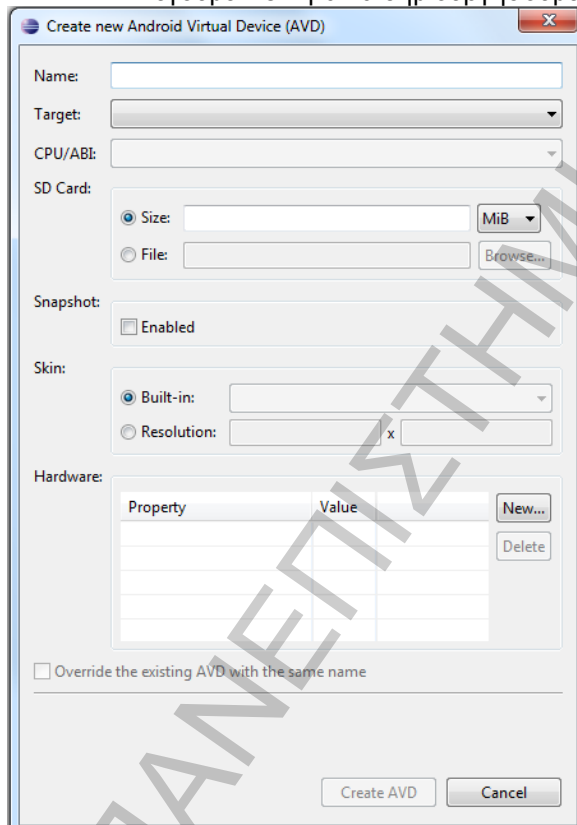
Δημιουργία Εικονικής συσκευής

Αφού κάνουμε import το αρχείο e-EKAB, όπως περιγράφεται στο κεφάλαιο «Οδηγίες εγκατάστασης», το επόμενο βήμα είναι να δημιουργήσουμε μία εικονική συσκευή Android, ένα emulator. Αυτό επιτυγχάνεται από το eclipse επιλέγοντας Window ->AVD Manager (Android Virtual Device). Εμφανίζεται το παράθυρο διαλόγου που φαίνεται στην εικόνα 51.



Εικόνα 51: Android Virtual Device Manager

Επιλέγουμε New για να δημιουργήσουμε μία καινούρια εικονική συσκευή.



Εικόνα 52: Δημιουργία νέας εικονικής συσκευής Android

Στην περιοχή του ονόματος πληκτρολογούμε το όνομα που επιθυμούμε να έχει η εικονική συσκευή π.χ. avd1. Στην περιοχή target επιλέγουμε Google APIs (Google Inc.) – API Level 8 διότι θέλουμε να χρησιμοποιήσουμε google maps στην εικονική μας συσκευή. Τέλος στην περιοχή Built-in επιλέγουμε προαιρετικά το μέγεθος της οθόνης της εικονικής συσκευής. Μόλις τελειώσουμε τις ρυθμίσεις μας επιλέγουμε Create AVD και κλείνουμε τον AVD Manager.

Τώρα που έχουμε δημιουργήσει και την εικονική συσκευή μπορούμε να τρέξουμε την εφαρμογή. Πηγαίνουμε στο eKAB, πατάμε δεξί κλικ και στη συνέχεια επιλέγουμε Run As ->

Android Application. Στη συνέχεια περιμένουμε να φορτώσει η εικονική συσκευή. Μόλις φορτώσει η εικονική συσκευή ξεκινάει αυτόματα η εκτέλεση της εφαρμογής.

Να αναφέρουμε ότι το project θεωρητικά θα πρέπει να χρησιμοποιεί τουλάχιστον δύο κινητές συσκευές έτσι ώστε η μία να παίζει το ρόλο του οδηγού ασθενοφόρου και η άλλη του ιατρού. Κάτι τέτοιο όμως δεν γίνεται διότι όσες εικονικές κινητές συσκευές και αν δημιουργήσουμε θα έχουν όλες την ίδια ip καθώς τρέχουν κάτω από το ίδιο εικονικό δίκτυο [52]. Για αυτό το λόγο χρησιμοποιούμε μόνο έναν emulator για την προσομοίωση.

Εισαγωγή γεωγραφικής θέσης

Επειδή τρέχουμε την εφαρμογή σε emulator θα πρέπει να δώσουμε στον emulator μία γεωγραφική θέση (μία πραγματική συσκευή θα λάμβανε τη θέση της μέσω gps). Αυτό επιτυγχάνεται επιλέγοντας Window -> Show View -> other. Στη συνέχεια Android -> Emulator Control. Από εκεί στην περιοχή Location Controls εισάγουμε ένα γεωγραφικό πλάτος και ένα γεωγραφικό μήκος και πατάμε Send.

Ανακατεύθυνση tcp πακέτων

Τέλος, θα πρέπει να προωθήσουμε τα tcp πακέτα που αποστέλλονται στην πόρτα 1236 του host στην πόρτα 1236 του emulator (Αυτήν την πόρτα χρησιμοποιούμε στην εφαρμογή μας). Αυτό γίνεται γιατί ο emulator είναι host μέσα σε host. Για να γίνει αυτό ανοίγουμε ένα command prompt και πληκτρολογούμε "telnet localhost 5554" (η οποία είναι η πόρτα του emulator που εμφανίζεται πάνω αριστερά). Με αυτή την εντολή συνδεόμαστε στον emulator. Στη συνέχεια πληκτρολογούμε την εντολή "redir add tcp:1236:1236" για να επιτύχουμε αυτό που προαναφέραμε.

Σε περίπτωση που το telnet δεν είναι εγκατεστημένο πριν την εντολή "telnet localhost 5554" πρέπει να πληκτρολογήσουμε την εντολή "pkgmgr /iu:"TelnetClient" η οποία θα εγκαταστήσει το telnet.