



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ανάπτυξη android εφαρμογής που αποστέλλει ειδοποιήσεις σε περιπτώσεις έκτακτης ανάγκης Emergency Call android application development
Όνοματεπώνυμο Φοιτήτριας	Ευθαλία Αμαργιανού
Πατρώνυμο	Χρήστος
Αριθμός Μητρώου	ΜΠΣΠ/ 13006
Επιβλέπων	Κωνσταντίνος Πατσάκης, Επίκουρος Καθηγητής



Ημερομηνία Παράδοσης **Οκτώβριος 2017**





Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Κωνσταντίνος Πατσάκης
Επίκουρος Καθηγητής

Ευθύμιος Αλέπης
Επίκουρος Καθηγητής

Γεώργιος Τσιχριτζής
Καθηγητής





Περίληψη

Στόχος της εργασίας είναι η ανάπτυξη μιας Android εφαρμογής που να επιτρέπει στους χρήστες να στέλνουν ειδοποιήσεις σε περιπτώσεις έκτακτης ανάγκης ή καταστάσεις πανικού. Οι χρήστες μπορούν επίσης να στέλνουν μηνύματα κειμένου και μηνύματα ηλεκτρονικού ταχυδρομείου με το πάτημα ενός μόνο κουμπιού.

Οι τηλεφωνικοί αριθμοί, οι λογαριασμοί ηλεκτρονικού ταχυδρομείου και το κείμενο των μηνυμάτων και των ηλεκτρονικών μηνυμάτων μπορούν να οριστούν μέσα από την εφαρμογή. Όλες οι ειδοποιήσεις που στέλνονται, έχουν επίσης και την τελευταία τοποθεσία στην οποία βρισκόταν ο χρήστης τη στιγμή που απέστειλε την ειδοποίηση. Αυτό θα βοηθήσει στον εντοπισμό του ατόμου που στέλνει την ειδοποίηση έκκλησης για βοήθεια. Οι χρήστες μπορούν να καλέσουν το 112 κατευθείαν μέσα από την εφαρμογή αν το κρίνουν απαραίτητο.

Επιπλέον, με την αποστολή των ειδοποιήσεων θα αποστέλλεται και το επίπεδο ισχύς της μπαταρίας της συσκευής του χρήστη, ώστε οι παραλήπτες να γνωρίζουν το διάστημα που έχουν ώστε να μπορέσουν να επικοινωνήσουν με το χρήστη της εφαρμογής. Επιπροσθέτως, παρέχεται η δυνατότητα καταγραφής αρχείου ήχου ώστε να μπορεί ο χρήστης να δημιουργήσει κάποιο ηχογραφημένο μήνυμα και να το αποστείλει. Οι χρήστες μπορούν επίσης να κάνουν χρήση υπηρεσίας πολυμεσικών μηνυμάτων, όπου παρέχεται η δυνατότητα αποστολής αρχείου εικόνας, βίντεο και ήχου, εκτός από κείμενο. Σε πολύ μεγάλες καταστροφές και αναταραχές, ειδοποιήσεις θα μπορούν να σταλούν και στις επαφές του χρήστη στα μέσα κοινωνικής δικτύωσης, εφόσον ο χρήστης επιτρέψει αυτή τη δυνατότητα μέσα από την εφαρμογή.

Μια πολύ σημαντική δυνατότητα που δίνεται στους χρήστες της εφαρμογής, είναι η τηλεφωνική κλήση μέσω διαδικτύου ή σωστότερα «Φωνή επί διαδικτυακού πρωτοκόλλου», η οποία προσφέρει φωνητική συνομιλία σε πραγματικό χρόνο με σχετικά καλή ποιότητα πλέον και στην ουσία χωρίς κόστος, καθότι στην κλήση αυτή δεν μεσολαβεί κάποιος παραδοσιακός φορέας τηλεπικοινωνιών, παρά μόνο το διαδίκτυο.

Abstract

The aim of the project is to develop an Android application that enables its users to send notifications in case of an emergency or in a panic situation. The users can also send messages and emails on the press of a single button.

The phone numbers, email ids and the contents of the text and email messages can be set from within the application. All notifications sent, also have the last known location of the user. This would help in tracking the location of the person. Users can also call the 112 directly from within the application, if the nature of the situation demands it.

Furthermore, by sending notifications will be sent and the level of battery power of the user's device, so recipients know the time they should be able to communicate with the application user. In addition, is provided the possibility of recording audio file so that the user can create a recorded message and send. Users can also make use of multimedia messaging service, where it is possible to image file upload, video and audio, besides text.

A very important option that is given to users of the application is the phone call via internet or more accurately "Voice over Internet Protocol", which offers voice chat in real time with fairly good quality and

Emergency Call Android Application Development



most essentially no cost, since in this call is not a traditional telecom operator that intervenes, but only the internet.



ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1.	Εισαγωγή	12
1.1	Περιγραφή του υπό μελέτη προβλήματος	12
1.2	Σκοπός και στόχοι της εργασίας	13
1.3	Βασικοί ορισμοί	14
1.4	Παραδοτέα της εργασίας.....	16
1.5	Δομή της εργασίας.....	16
2.	Επισκόπηση του Android.....	17
2.1	Android Architecture.....	17
2.1.1	Google Maps Android API v2.....	19
2.1.2	Java	20
2.1.3	Kotlin	24
2.2	MongoDB	27
2.2.1	MongoDB και Node.js	28
2.2.2	MongooseJS	31
2.3	BackEnd - Node.js.....	33
2.3.1	Express.js	35
2.3.2	Firebase	36
2.3.3	FrontEnd – AngularJS	37
2.4	Security.....	38
2.5	Android app – Περιγραφή εφαρμογής	40
2.5.1	Web app – Περιγραφή FrontEnd	63
2.6	Αιτιολόγηση και συναφείς εφαρμογές.....	68
2.7	Μελλοντικές εργασίες εξέλιξης της εφαρμογής.....	100
3.	Συμπεράσματα	101
4.	Βιβλιογραφικές Πηγές.....	102



ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

Εικόνα 1. Android Stack.....	18
Εικόνα 2. Παράδειγμα εγγράφου mongodb.....	27
Εικόνα 3. Παράδειγμα δημιουργίας node.js server.....	34
Εικόνα 4. Περιγραφή λειτουργιών Firebase	37
Εικόνα 5. Κεντρική οθόνη της mobile εφαρμογής.....	41
Εικόνα 6. Αρχική οθόνη της mobile εφαρμογής	42
Εικόνα 7. Οθόνη δημιουργίας λογαριασμού χρήστη στη mobile εφαρμογή.....	51
Εικόνα 8. Οθόνη επαναφοράς κωδικού πρόσβασης.....	52
Εικόνα 9. Αρχική οθόνη μετά την επαναφορά κωδικού	53
Εικόνα 10. Κεντρική οθόνη εφαρμογής - επιλογή κωδικού πρόσβασης.....	54
Εικόνα 11. Οθόνη αλλαγής κωδικού πρόσβασης	55
Εικόνα 12. Κεντρική οθόνη εφαρμογής - Διαθέσιμες επιλογές	56
Εικόνα 13. Οθόνη εισαγωγής χρησικών προσωπικών πληροφοριών χρήστη	57
Εικόνα 14. Οθόνη αποστολής ειδοποιήσεων έκτακτης ανάγκης	58
Εικόνα 15. Κεντρική οθόνη εφαρμογής - μήνυμα ενημέρωσης παραλαβής "alert".....	61
Εικόνα 16. Οθόνη αποστολής alert μετά τη λήψη φωτογραφίας	62
Εικόνα 17. Οθόνη διαχείρισης σημάτων κινδύνου - web εφαρμογής	63
Εικόνα 18. Πλαίσιο πληροφοριών σήματος κινδύνου στη σελίδα διαχείρισης του Web app	64
Εικόνα 19. Web app - Ειδοποίηση λήψης μηνύματος κινδύνου	67
Εικόνα 20. Web app - Αποστολή ειδοποίησης προς το χρήστη της Mobile εφαρμογής	67
Εικόνα 21. Mobile εφαρμογή - λήψη ειδοποίησης από το Web app.....	68



Εικόνα 22. Κεντρική οθόνη 112 Iceland εφαρμογής.....	71
Εικόνα 23. Οθόνη προσθήκης πληροφοριών του χρήστη στην 112 Iceland app.....	72
Εικόνα 24. Οθόνη επιλογής γλώσσας του περιβάλλοντος της 112 Iceland εφαρμογής.....	73
Εικόνα 25. Οθόνη ενημέρωσης του χρήστη από το 112 για το επιτυχές Check in στην 112 Iceland app.....	74
Εικόνα 26. Οθόνη εισόδου στην εφαρμογή 112 BE.....	75
Εικόνα 27. Οθόνη συμπλήρωσης προσωπικών πληροφοριών του χρήστη 112 BE app.....	76
Εικόνα 28. Οθόνη συμπληρωματικών πληροφοριών χρήστη - 112 BE app.....	77
Εικόνα 29. Οθόνη συμπλήρωσης πρόσθετων πληροφοριών του χρήστη - 112 BE app.....	78
Εικόνα 30. Οθόνη συμπλήρωσης στοιχείων πρόσθετης επαφής επικοινωνίας - 112 BE app.....	79
Εικόνα 31. Οθόνη επιβεβαίωσης πρόσβασης της εφαρμογής σε λειτουργίες της συσκευής - 112 BE app.....	80
Εικόνα 32. Οθόνη επιλογής κινδύνου - κλήση έκτακτης ανάγκης - 112 BE app.....	81
Εικόνα 33. Κεντρική οθόνη της εφαρμογής S.O.S.....	82
Εικόνα 34. Μενού επιλογών - S.O.S. app.....	83
Εικόνα 35. Οθόνη δήλωσης αριθμού τηλεφώνου στην εφαρμογή - 112 Where are U app.....	85
Εικόνα 36. Επαλήθευση αριθμού τηλεφώνου από την εφαρμογή - 112 Where are U app.....	86
Εικόνα 37. Αποστολή κωδικού με sms για την ολοκλήρωση της εγγραφής στην εφαρμογή - 112 Where are U app.....	87
Εικόνα 38. Οθόνη ρυθμίσεων - δυνατότητα διαγραφής του χρήστη από τη ΒΔ της εφαρμογής - 112 Where are U app.....	88
Εικόνα 39. Οθόνη προσθήκης επαφών ενημέρωσης σε περίπτωση έκτακτης ανάγκης - 112 Where are U app.....	89
Εικόνα 40. Επιλογές προσθήκης επαφών - 112 Where are U app.....	90
Εικόνα 41. Οθόνη κλήσεων έκτακτης ανάγκης - 112 Where are U app.....	91



Εικόνα 42. Επιλογή "VOICE CALL" κλήσης έκτακτης ανάγκης - 112 Where are U app.....	92
Εικόνα 43. "SILENT CALL" - 112 Where are U app.....	93
Εικόνα 44. Κεντρική οθόνη εφαρμογής - 911 USA	94
Εικόνα 45. Κλήση προς το 911	95
Εικόνα 46. Κλήση επαφής κοντινού προσώπου σε περίπτωση ανάγκης - 911 USA app	96
Εικόνα 47. Οθόνη πρόσθετων πληροφοριών - 911 USA app	97

ΠΙΝΑΚΑΣ ΠΙΝΑΚΩΝ

Πίνακας 1. Εφαρμογές με λειτουργίες κλήσης προς το 112/911 και αποστολή ακριβούς γεωγραφικού σημείου.....	98
-----------------------------------------------------------------------------------------------------------	----



Κεφάλαιο 1^ο

1. Εισαγωγή

Όσο και να θέλαμε να αποφύγουμε καταστάσεις πανικού ή έκτακτης ανάγκης, δυστυχώς είναι αναπόφευκτες και συνήθως απροσδόκητες. Η φύση και οι συνέπειες αυτών των καταστάσεων μπορεί να ποικίλλουν σημαντικά και μπορεί να γίνουν απειλητικές για την ανθρώπινη ζωή. Παραδείγματα τέτοιων καταστάσεων έκτακτης ανάγκης μπορεί να είναι, φυσικές, περιβαλλοντικές ή καιρικές καταστροφές, όπως: φωτιά, κατολίθωση, σεισμός, πλημμύρα κ.α, μπορεί να είναι και καταστάσεις που αφορούν ξαφνική επιδείνωση της υγείας ενός ανθρώπου, εγκληματικές ενέργειες ή και οδικά και οποιασδήποτε φύσης ατυχήματα. Ως εκ τούτου, θα ήταν πραγματικά ωραία να υπάρχει ένας μηχανισμός με τον οποίο τα άτομα που βρίσκονται σε τέτοιες περιπτώσεις έκτακτης ανάγκης, να μπορούν να τις κοινοποιούν, να στέλνουν δηλαδή μια ειδοποίηση για την κατάσταση έκτακτης ανάγκης στην οποία βρίσκονται ώστε να λαμβάνουν βοήθεια το συντομότερο δυνατόν.

Η ανάγκη για ένα τέτοιο μηχανισμό αυξάνεται όλο και περισσότερο καθώς στην εποχή της αυξημένης τεχνολογικής ανάπτυξης που βρισκόμαστε, υπάρχουν και οι πλατφόρμες για να υποστηρίξουν ένα τέτοιο μηχανισμό. Μια τέτοια πλατφόρμα και πολύ διαδεδομένη είναι η λεγόμενη «έξυπνη» κινητή τηλεφωνική συσκευή ή κοινώς Smartphone. Σχεδόν όλοι οι άνθρωποι ή οι περισσότεροι, ανεξαρτήτου ηλικίας, στις μέρες μας, χρησιμοποιούν κινητές τηλεφωνικές συσκευές Smartphone, καθότι γίνονται ολοένα και πιο προσιτές οικονομικά και είναι πλέον εύκολα διαθέσιμες. Επίσης, εντός της αγοράς των Smartphone, οι συσκευές με λειτουργικό σύστημα Android κατέχουν το μεγαλύτερο μερίδιο. Σύμφωνα με μια στατιστική μελέτη, το δεύτερο τρίμηνο του 2016, το 87,6% του μεριδίου της αγοράς των συσκευών smartphone καταλάμβαναν οι συσκευές με λειτουργικό σύστημα Android. Ως εκ τούτου, η ανάπτυξη μια android εφαρμογής είναι και η πιο προφανής επιλογή καθότι το συγκεκριμένο λειτουργικό σύστημα χρησιμοποιείται από τους περισσότερους χρήστες smartphone συσκευών παγκοσμίως.

1.1 Περιγραφή του υπό μελέτη προβλήματος

Emergency call είναι μια εφαρμογή που προορίζεται να «τρέξει» σε Android smartphone συσκευές κυρίως αλλά και σε συσκευές tablets που υποστηρίζουν κυψελοειδή τηλεφωνική επικοινωνία. Οι κύριες λειτουργίες και τα χαρακτηριστικά της εφαρμογής είναι:

- i. Ο χρήστης της εφαρμογής θα πρέπει να συνδεθεί εισάγοντας ένα όνομα χρήστη και κωδικό πρόσβασης την πρώτη φορά που ανοίγει την εφαρμογή στη συσκευή του. Στη συνέχεια παραμένει συνδεδεμένος στην εφαρμογή μέχρι να αποσυνδεθεί εκτελώντας τη διαδικασία αποσύνδεσης.
- ii. Εάν ο χρήστης δεν έχει δημιουργήσει λογαριασμό, μπορεί να εγγραφεί μετά το άνοιγμα της εφαρμογής, στην οθόνη εισόδου στην εφαρμογή.
- iii. Ο χρήστης μπορεί να επιλέξει την επαναφορά κωδικού πρόσβασης σε περίπτωση που δεν θυμάται τον κωδικό πρόσβασης του. Με τη διαδικασία αυτή ένας νέος κωδικός πρόσβασης ορίζεται για το



χρήστη και ένα μήνυμα που περιέχει αυτό το νέο κωδικό πρόσβασης αποστέλλεται στο λογαριασμό ηλεκτρονικού ταχυδρομείου που έχει καταχωρήσει ο χρήστης.

- iv. Μόλις συνδεθεί ο χρήστης κατευθύνεται στην κύρια οθόνη της εφαρμογής. Αυτή είναι η οθόνη που θα ανοίξει όταν ο χρήστης ανοίγει την εφαρμογή. Ο χρήστης μπορεί να πατήσει το κουμπί έκτακτης ανάγκης για να στείλει μηνύματα κειμένου και μηνύματα ηλεκτρονικού ταχυδρομείου στις επαφές που έχει ορίσει. Ο χρήστης έχει τη δυνατότητα να στείλει ένα άμεσο μήνυμα «είμαι καλά» στις επαφές που έχει ορίσει κάνοντας κλικ στο κουμπί OK. Επίσης, μπορεί να καλέσει απευθείας το 112 μέσα από την εφαρμογή πατώντας το κουμπί 112. Για να αποφευχθεί η άσκοπη και τυχαία χρήση των κουμπιών αυτών παρέχεται η δυνατότητα ενεργοποίησης και απενεργοποίησης τους από το χρήστη.
- v. Ο χρήστης θα μπορεί να δει στην κύρια οθόνη την γεωγραφική θέση που βρίσκεται εκείνη τη στιγμή, ώστε σε περίπτωση κλήσης προς το 112 να μπορεί να δώσει και αυτή την πολύ σημαντική πληροφορία.
- vi. Ο χρήστης μπορεί να ρυθμίσει τις επαφές για να στείλει τα μηνύματα κειμένου και μηνύματα ηλεκτρονικού ταχυδρομείου μέσα από την εφαρμογή και να ορίσει και τα αντίστοιχα μηνύματα. Μπορεί επίσης, να επιλέξει να ορίσει τις επαφές μέσα από τον κατάλογο επαφών ή να τις εισάγει μόνος του.
- vii. Ο χρήστης μέσα από την εφαρμογή μπορεί να ελέγχει το επίπεδο ισχύς της μπαταρίας της συσκευής του. Η πληροφορία αυτή θα αποστέλλεται και στα μηνύματα έκτακτης ανάγκης.
- viii. Ο χρήστης μπορεί να αποστέλλει, εκτός από μηνύματα απλού κειμένου και μηνύματα ήχου και φωτογραφιών και βίντεο.
- ix. Ο χρήστης θα μπορεί να ενεργοποιήσει τη δυνατότητα σύνδεσης της εφαρμογής με μέσα κοινωνικής δικτύωσης και σε περίπτωση έκτακτης ανάγκης, αν κρίνει ότι πρέπει, να χρησιμοποιήσει τη δυνατότητα αποστολής μηνύματος σε φίλους του στα μέσα αυτά.
- x. Μέσα από την εφαρμογή παρέχεται στο χρήστη η δυνατότητα να πραγματοποιήσει κλήσεις μέσω διαδικτύου (VOIP).

1.2 Σκοπός και στόχοι της εργασίας

Στόχος της εργασίας είναι η χρησιμοποίηση όλων των απαραίτητων τεχνολογιών για την ανάλυση, των σχεδιασμό και την υλοποίηση της Android εφαρμογής αποστολής ειδοποιήσεων σε περιπτώσεις έκτακτης ανάγκης. Στο πλαίσιο αυτό γίνεται η παρουσίαση και ανάλυση του λειτουργικού συστήματος Android και της αρχιτεκτονικής δομής του. Επίσης, παρουσιάζονται η τεχνολογίες που χρησιμοποιήθηκαν στο στάδιο υλοποίησης της εφαρμογής, σε επίπεδο server, βάσεων δεδομένων, web services και το γραφικό τμήμα της εφαρμογής.

Επιπρόσθετα, γίνεται προσέγγιση του ζητήματος σε ευρύτερο επίπεδο, ώστε να γίνουν κατανοητές σε όσο το δυνατόν μεγαλύτερο βαθμό οι ανάγκες των χρηστών που βρίσκονται σε κατάσταση έκτακτης ανάγκης. Σε συνδυασμό με τις τεχνολογίες που χρησιμοποιήθηκαν για την καλύτερη υλοποίηση της εφαρμογής, σκοπός



της εργασίας είναι η εφαρμογή να παρέχει όλες εκείνες τις χρήσιμες δυνατότητες στους χρήστες που βρίσκονται σε κατάσταση πανικού ή έκτακτης ανάγκης, ώστε να επιτευχθεί ο άμεσος εντοπισμός και η αποστολή της κατάλληλης βοήθειας για την προστασία του χρήστη.

1.3 Βασικοί ορισμοί

Ακολουθούν μερικές λέξεις κλειδιά μαζί με τους ορισμούς τους που θα χρησιμοποιηθούν κατά τη διάρκεια της διπλωματικής και θα πρέπει να είναι από πριν κατανοητές.

Web application – Ονομάζεται κάθε εφαρμογή η οποία είναι διαθέσιμη στους χρήστες της μέσω του διαδικτύου ή του ενδοδικτύου μια εταιρίας και ο χρήστης χρησιμοποιεί μόνο τον περιηγητή του για να τη χρησιμοποιήσει.

Standalone application – Λογισμικό υπολογιστή ικανό να λειτουργεί ανεξάρτητα και εκτός δικτύου.

Back-End / Front-End Development – Είναι ο διαχωρισμός της φάσης της δημιουργίας μίας εφαρμογής ανάμεσα στο κομμάτι που θα τρέχει στον απομακρυσμένο server και σε αυτό που θα τρέχει στον υπολογιστή του χρήστη.

API – Είναι η Διασύνδεση Προγραμματισμού Εφαρμογών, ή αλλιώς διεπαφή, με την οποία ένα σύστημα μπορεί να επικοινωνήσει με ένα άλλο και να ζητήσει δεδομένα ή λειτουργίες.

Framework – Είναι ένα πρόγραμμα, ή σειρά προγραμμάτων που σκοπός τους είναι να προσφέρουν περισσότερες επιλογές και ευκολίες στη χρήση μιας γενικότερης πλατφόρμας.

Βιβλιοθήκη – Είναι μια συλλογή από εφόδια που έχουν φτιαχτεί για να κάνουν πιο εύκολη την διαδικασία του Development. Μοιάζει νοητικά με το Framework αλλά συνήθως είναι πολύ μικρότερου μήκους και δεν δημιουργούνται για συγκεκριμένες τεχνολογίες.

IDE – Ολοκληρωμένο περιβάλλον ανάπτυξης, είναι μια σουίτα λογισμικού που βοηθάει στην ανάπτυξη προγραμμάτων υπολογιστή.

JDK – Είναι ένα περιβάλλον ανάπτυξης προγραμμάτων για τη σύνταξη μικροεφαρμογών και εφαρμογών Java. Αποτελείται από ένα περιβάλλον χρόνου εκτέλεσης που βρίσκεται "επάνω" στο επίπεδο του λειτουργικού συστήματος καθώς και από τα εργαλεία και τον προγραμματισμό που χρειάζονται οι προγραμματιστές για να μεταγλωττίζουν, να εντοπίζουν σφάλματα και να εκτελούν εφαρμογές και εφαρμογές γραμμένες στη γλώσσα Java.

ORM – Object-relational mapping: αντικείμενο-σχεσιακή χαρτογράφηση/ αντιστοίχιση, είναι μια τεχνική προγραμματισμού για τη μετατροπή δεδομένων μεταξύ μη συμβατών τύπων συστημάτων, χρησιμοποιώντας αντικειμενοστραφείς γλώσσες προγραμματισμού. Αυτό δημιουργεί, στην πραγματικότητα, μια "βάση δεδομένων εικονικών αντικειμένων" που μπορεί να χρησιμοποιηθεί από τη γλώσσα προγραμματισμού. Υπάρχουν διαθέσιμα δωρεάν και εμπορικά πακέτα που εκτελούν αντικειμενοστραφή χαρτογράφηση, παρόλο που ορισμένοι προγραμματιστές επιλέγουν να κατασκευάσουν τα δικά τους εργαλεία ORM.

CRUD – create, read, update και delete: Στον προγραμματισμό υπολογιστών, δημιουργώ, διαβάζω, ενημερώνω και διαγράφω, είναι οι τέσσερις βασικές λειτουργίες της συνεχούς αποθήκευσης. Εναλλακτικές λέξεις χρησιμοποιούνται μερικές φορές κατά τον ορισμό των τεσσάρων βασικών λειτουργιών CRUD.



Ανάκτησης, αντί για ανάγνωση, τροποποίηση αντί για ενημέρωση ή καταστροφή αντί για διαγραφή. Το CRUD χρησιμοποιείται επίσης μερικές φορές για να περιγράψει τις συμβάσεις διεπαφής χρήστη που διευκολύνουν την προβολή, την αναζήτηση και την ανταλλαγή των πληροφοριών συχνά χρησιμοποιώντας υπολογιστικές φόρμες και αναφορές.

Database Schema – Είναι η λογική σχεδίαση του διαχωρισμού των δεδομένων σε πίνακες και της εύρεσης των συνδέσεων αυτών μεταξύ τους. Σύμφωνα με αυτό θα δημιουργηθεί και η βάση δεδομένων που θα γεμίσει με αυτά τα δεδομένα.

JSON – Είναι μορφή αναπαράστασης δεδομένων και δημιουργήθηκε για την ευκολότερη μεταφορά αυτών στο Διαδίκτυο.

Smartphone – Έξυπνο τηλέφωνο, είναι ένα κινητό τηλέφωνο βασισμένο σε ένα λειτουργικό σύστημα κινητής τηλεφωνίας με περισσότερη προηγμένη υπολογιστή ικανότητα και συνδεσιμότητα σε σχέση με τα συμβατικά κινητά.

GNSS – Ο όρος «παγκόσμιο δορυφορικό σύστημα πλοήγησης» (GNSS) αναφέρεται σε έναν αστερισμό δορυφόρων που παρέχει σήματα από δεδομένα θέσης και χρονισμού μετάδοσης διαστήματος. Εξ ορισμού, ένα GNSS παρέχει παγκόσμια κάλυψη.

GPS – Το παγκόσμιο σύστημα εντοπισμού θέσης NAVSTAR (GPS) των Η.Π.Α. είναι το ευρύτερα χρησιμοποιούμενο παγκόσμιο δορυφορικό σύστημα πλοήγησης (GNSS) στον κόσμο. Χρησιμοποιείται ευρέως για πλοήγηση από το κοινό, τις επιχειρήσεις και άλλους οργανισμούς. Σήμερα, το GPS ενσωματώνεται σε μια αυξανόμενη ποικιλία συσκευών εντοπισμού θέσης, όπως τα έξυπνα τηλέφωνα.

EENA – European Emergency Number Association. Αριθμός κλήσης έκτακτης ανάγκης της Ευρωπαϊκής Ένωσης.

Advanced Mobile Location (AML) – Είναι μια τεχνική εντοπισμού του καλούντος σε περίπτωση έκτακτης ανάγκης. Αναπτύχθηκε στο Ηνωμένο Βασίλειο από την British Telecom, την EE Limited και την HTC ως λύση για την προβληματική τοποθεσία καλούντος σε καταστάσεις έκτακτης ανάγκης. Όταν ένα άτομο σε κατάσταση ανάγκης καλεί τις υπηρεσίες έκτακτης ανάγκης με ένα έξυπνο τηλέφωνο όπου είναι ενεργοποιημένη η λειτουργία AML, το τηλέφωνο ενεργοποιεί αυτόματα την υπηρεσία εντοπισμού θέσης για να καθορίσει τη θέση του και στέλνει τις πληροφορίες αυτές στις υπηρεσίες έκτακτης ανάγκης μέσω SMS. Οι υπηρεσίες χρησιμοποιούν είτε GNSS είτε WiFi, ανάλογα με το ποια είναι καλύτερη στη δεδομένη στιγμή. Εκτιμάται ότι αυτή η τεχνική είναι έως 4000 φορές ακριβέστερη από το προηγουμένως χρησιμοποιούμενο σύστημα.



1.4 Παραδοτέα της εργασίας

1 Το έντυπο κείμενο της πτυχιακής εργασίας, το οποίο περιλαμβάνει την επισκόπηση , ανάλυση και μελέτη των τεχνολογιών που περιλαμβάνει το Android περιβάλλον προγραμματισμού, καθώς και τη διαδικασία δημιουργίας της εφαρμογής Emergency Call Android Application.

2 Τον κώδικα που αναπτύχθηκε για την δημιουργία της εφαρμογής Emergency Call, σε Back-end , Front-end και Database επίπεδο.

1.5 Δομή της εργασίας

Η παρούσα εργασία θα αναλυθεί σε επιμέρους τμήματα για να γίνει όσο το δυνατό περισσότερο κατανοητό κάθε κομμάτι αυτής. Στα πρώτα κομμάτια θα γίνει η περιγραφή και η ανάλυση των εκάστοτε τεχνολογιών μαζί με κομμάτια κώδικα που θα κάνουν πιο εύκολη την παρακολούθηση του κειμένου. Στα επόμενα κομμάτια , θα δούμε τον κώδικα και την διαδικασία υλοποίησης της εφαρμογής Project Management. Τέλος, θα περιγραφούν οι προβληματισμοί και τα σχόλια γύρω από την εφαρμογή και τις τεχνολογίες που χρησιμοποιήθηκαν.

Στην παρούσα εργασία συμπεριλαμβάνεται :



Κεφάλαιο 2^ο

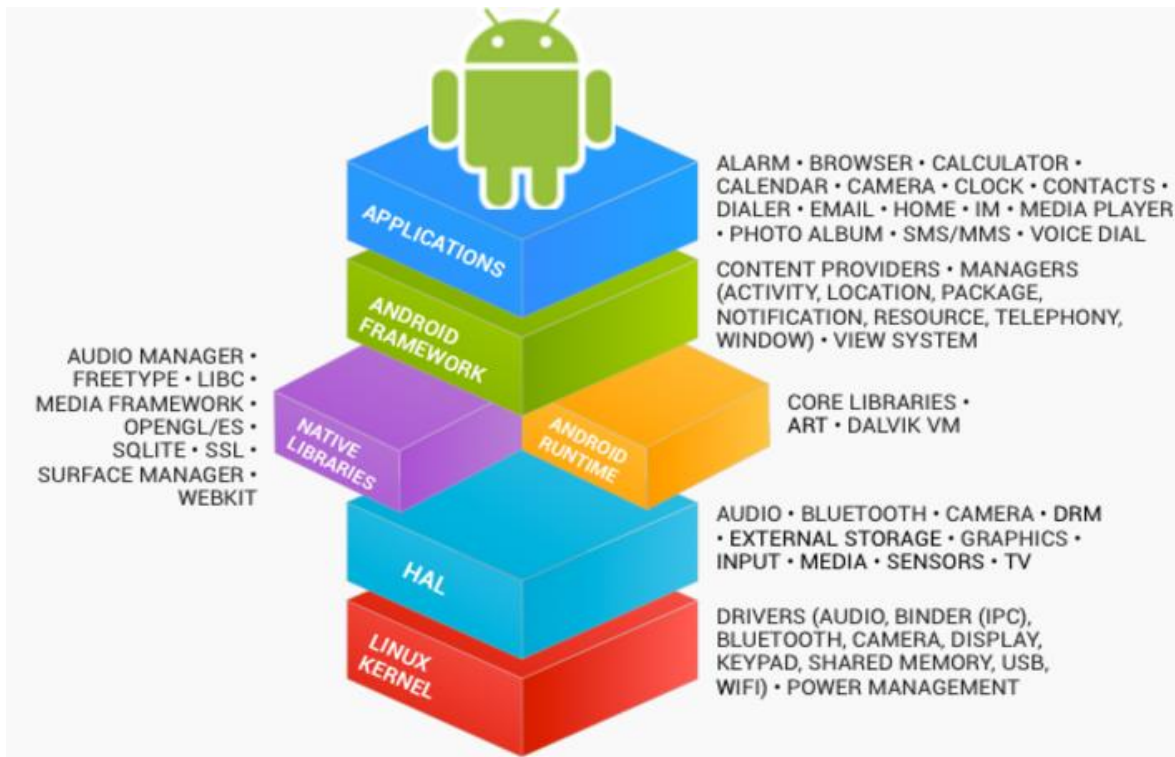
2. Επισκόπηση του Android

Το Android είναι το πιο ευρέως διαδεδομένο λειτουργικό στον κόσμο. Είναι ένα λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση τη γλώσσας προγραμματισμού Java, ελέγχοντας τη συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από τη Google. Το Android είναι σχεδιασμένο κυρίως για συσκευές με οθόνες αφής, όπως τα έξυπνα τηλέφωνα (smartphones) και τα τάμπλετ (tablet), με διαφορετικό περιβάλλον χρήσης για τηλεοράσεις (Android TV), αυτοκίνητα (Android Auto) και ρολόγια χειρός (Android Wear). Παρόλο που η συνήθης χρήση του είναι σε συσκευές με οθόνη αφής, έχει χρησιμοποιηθεί και σε κονσόλες παιχνιδιών, ψηφιακές φωτογραφικές μηχανές, ηλεκτρονικούς υπολογιστές και άλλες ηλεκτρικές συσκευές.

Συγκεκριμένα, στα επόμενα κεφάλαια θα αναλυθούν τα παρακάτω :

2.1 Android Architecture

Το Android είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα και έχει υλοποιηθεί σαν μια στοίβα λογισμικού διαφορετικών επιπέδων. Στην εικόνα που ακολουθεί διακρίνονται τα διαφορετικά επίπεδα της στοίβας.



Εικόνα 1. Android Stack

Linux Kernel. Αυτό είναι το επίπεδο στο κάτω μέρος της αρχιτεκτονικής του Android. Όλα τα άλλα επίπεδα τρέχουν στην κορυφή του πυρήνα του Linux και βασίζονται σε αυτό τον πυρήνα για να αλληλοεπιδρούν με το υλικό (hardware). Αυτό το επίπεδο περιέχει όλους του απαραίτητους hardware drivers, οι οποίοι βοηθούν στον έλεγχο και την επικοινωνία με το υλικό (hardware). Παρέχει βασικές λειτουργικότητες όπως, Διαχείριση των Διαδικασιών (process management), Διαχείριση Μνήμης και Διαχείριση Συσκευών όπως κάμερα, οθόνη, Flash κ.α.

Libraries. Οι βιβλιοθήκες είναι ένα σύνολο κοινών λειτουργιών του σκελετού της εφαρμογής που επιτρέπει στη συσκευή να χειριστεί διαφορετικούς τύπους δεδομένων. Μερικά από τα πιο σημαντικά σύνολα βιβλιοθηκών που περιλαμβάνονται είναι το Web Kit, που είναι ο κινητήρας του προγράμματος περιήγησης για την εκκίνηση της HTML, OpenGL χρησιμεύει για την απόδοση 2D ή 3-D γραφικών επί της οθόνης, SQLite είναι μια δημοφιλής επιλογή ως ενσωματωμένη βάση δεδομένων για τοπική αποθήκευση.

Android Runtime. Αποτελείται κυρίως από την εικονική μηχανή Dalvik (DVM), η οποία μοιάζει πολύ με το Java εικονικό μηχάνημα (JVM) εκτός από το ότι έχει βελτιστοποιηθεί για κινητές συσκευές που έχουν χαμηλή επεξεργαστική ισχύ και χαμηλή μνήμη. Το DVM δημιουργεί ένα αρχείο .dex από το αρχείο .class κατά τη μεταγλώττιση και παρέχει υψηλότερη απόδοση σε συσκευές με χαμηλούς πόρους. Κάθε εφαρμογή έχει τη δική της διαδικασία και ένα στιγμιότυπο του DVM. Το Android Runtime παρέχει επίσης βασικές βιβλιοθήκες που επιτρέπουν στους προγραμματιστές να δημιουργήσουν Java εφαρμογές.

Application Framework. Είναι μερικά πρότυπα αρχεία κλάσης (.class) τα οποία είναι διαθέσιμα να τα χρησιμοποιήσουν οι προγραμματιστές. Μια εφαρμογή μπορεί να αλληλοεπιδρά άμεσα μαζί τους και να τα

Emergency Call Android Application Development



χρησιμοποιεί. Το Framework της εφαρμογής παρέχει τις πιο βασικές λειτουργίες του τηλεφώνου όπως Location Manager, Content Providers κ.α.

Applications. Αυτό είναι το ανώτατο επίπεδο της Android αρχιτεκτονικής και το επίπεδο όπου βρίσκεται η εφαρμογή που αναπτύσσουμε. Αυτό το επίπεδο παρέχει αρκετές προεγκατεστημένες εφαρμογές που είναι προεπιλεγμένες για ορισμένα πράγματα, όπως το Βιβλίο επαφών, ο Φυλλομετρητής κ.α.

2.1.1 Google Maps Android API v2

Η Google παρέχει ένα πολύ χρήσιμο API στους προγραμματιστές Android εφαρμογών που θέλουν να χρησιμοποιήσουν Google Maps στην εφαρμογή τους. Το API χειρίζεται αυτόματα την πρόσβαση στους Google Maps servers, στο κατέβασμα δεδομένων, στην απεικόνιση του χάρτη και απαντά σε κινήσεις στο χάρτη. Οι κλήσεις στο API μπορούν επίσης να χρησιμοποιηθούν ώστε να προστεθούν δείκτες, πολύγωνα και επικαλύψεις σε ένα βασικό χάρτη και να αλλάξει την εικόνα του χρήστη για μια συγκεκριμένη περιοχή στο χάρτη. Το API επιτρέπει να προστεθούν τα παρακάτω γραφικά σε ένα χάρτη:

- Εικόνες τοποθετημένες σε συγκεκριμένες θέσεις πάνω στο χάρτη (δείκτες)
- Σύνολα ευθύγραμμων τμημάτων (πολλαπλές γραμμές)
- Εσώκλειστα τμήματα (πολύγωνα)
- Γραφικά Bitmap τοποθετημένα σε συγκεκριμένες θέσεις πάνω στο χάρτη (επικαλύψεις εδάφους)
- Σύνολα εικόνων που εμφανίζονται στην κορυφή του χάρτη ως βάση πλακιδίων (επικαλύψεις πλακιδίων)

Για να αρχίσει κάποιος να χρησιμοποιεί το Google Maps API v2 πρέπει να αποκτήσει ένα Google Maps API κλειδί, ώστε να αποκτήσει πρόσβαση στους Google Maps servers. Ο τύπος κλειδιού που χρειάζεται να αποκτηθεί είναι ένα API κλειδί αποκλειστικά για Google εφαρμογές. Για τη λήψη του API key πρέπει να γίνει εγγραφή του project της εφαρμογής στην κονσόλα API της Google. Επίσης, θα πρέπει να προστεθεί το Google Play services SDK ως πρόγραμμα στον τοπικό δίσκο και να προστεθεί ως εξωτερική βιβλιοθήκη στην εφαρμογή.

Το επόμενο βήμα είναι να προστεθεί η έκδοση του Google play services στο αρχείο Manifest της εφαρμογής προσθέτοντας:

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

Μόλις αποκτηθεί το API key της εφαρμογής, προσθέτουμε στο αρχείο Manifest:

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

Πρέπει επίσης να προστεθεί και η ακόλουθη άδεια στο αρχείο Manifest:

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



```
<uses-permission Android:name="Android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission Android:name="com.google.Android.providers.gsf.permission.READ_GSERVICES"/>
```

Για να προστεθεί ο χάρτης Google θα πρέπει επίσης να προστεθεί και ένα fragment στον Activity χρησιμοποιώντας:

```
<fragment xmlns:Android="http://schemas.Android.com/apk/res/Android"
    Android:id="@+id/map"
    Android:layout_width="match_parent"
    Android:layout_height="match_parent"
    Android:name="com.google.Android.gms.maps.MapFragment"/>
```

Για ένα στιγμιότυπο google map θα πρέπει να καλέσουμε τη συνάρτηση `getMap()` σε ένα αντικείμενο `MapFragment`:

```
GoogleMap map = ((MapFragment)
getFragmentManager().findFragmentById(R.id.map)).getMap();
```

Στη συνέχεια μπορούμε να προσθέσουμε δείκτες στο χάρτη καλώντας τη μέθοδο `addMarker`:

```
map.addMarker(new MarkerOptions()
    .title(address)
    .position(new LatLng(latitude, longitude))
    .snippet(time));
```

2.1.2 Java

Η Java είναι μια γλώσσα προγραμματισμού υψηλού επιπέδου που δημιουργήθηκε από τον James Gosling, από τη Sun Microsystems (Sun) το 1991. Ο στόχος της Java είναι να γράψει ένα πρόγραμμα μια φορά και στη συνέχεια να εκτελέσει αυτό το πρόγραμμα σε διάφορα λειτουργικά συστήματα. Η πρώτη έκδοση της Java (Java 1.0) που κυκλοφόρησε δημοσίως, κυκλοφόρησε το 1995. Η Sun Microsystems αποκτήθηκε από την Oracle Corporation το 2010. Η Oracle έχει τώρα τη διαχείριση της Java. Το 2006, η Sun άρχισε να καθιστά διαθέσιμη την Java υπό την Γενική Άδεια Δημόσιας Χρήσης του GNU (GPL). Η Oracle συνεχίζει αυτό το έργο με το όνομα OpenJDK.

Με την πάροδο του χρόνου νέες βελτιωμένες εκδόσεις της Java έχουν κυκλοφορήσει. Η τρέχουσα έκδοση είναι η 1.8 η οποία είναι επίσης γνωστή και ως Java 8.

Η Java ορίζεται από κάποιες προδιαγραφές και αποτελείται από μια γλώσσα προγραμματισμού, έναν μεταγλωττιστή, τις βασικές βιβλιοθήκες και ένα runtime (εικονική μηχανή Java, JVM). Το Java runtime επιτρέπει στους προγραμματιστές να γράφουν κώδικα προγράμματος σε άλλες γλώσσες από τη γλώσσα προγραμματισμού Java που εξακολουθεί να τρέχει στην Java εικονική μηχανή. Η πλατφόρμα Java συνδέεται με την εικονική μηχανή Java και τις βασικές βιβλιοθήκες Java. Η Java εικονική μηχανή (JVM) εκτελεί κώδικα Java, αλλά είναι γραμμένο σε άλλες γλώσσες όπως η C / C++ / ASM κτλ. Η JVM δεν είναι γραμμένη σε Java



και ως εκ τούτου δεν μπορεί να είναι ανεξάρτητη από την πλατφόρμα και ο διερμηνέας Java είναι στην πραγματικότητα μέρος του JVM.

Παλαιότερα η Java χρησιμοποιήθηκε για το σχεδιασμό και τον προγραμματισμό μικρών υπολογιστικών συσκευών, αλλά αργότερα υιοθετήθηκε ως μια ανεξάρτητη πλατφόρμας γλώσσα προγραμματισμού και τώρα σύμφωνα με τη Sun 3 δισεκατομμύρια τρέχουν Java.

Η Java είναι μια από τις πιο σημαντικές γλώσσες στις σύγχρονες βιομηχανίες της πληροφορικής.

Χρησιμοποιείται:

- JSP: Η Java χρησιμοποιείται για τη δημιουργία διαδικτυακών εφαρμογών όπως PHP και ASP, JSP (Σελίδες διακομιστών Java) που χρησιμοποιούνται με τις συνήθεις ετικέτες HTML, γεγονός που βοηθά στη δημιουργία δυναμικών ιστοσελίδων.
- Applets: Αυτός είναι ένας άλλος τύπος προγράμματος Java, που χρησιμοποιείται σε μια ιστοσελίδα, για να προσθέσει πολλά νέα χαρακτηριστικά σε ένα πρόγραμμα περιήγησης ιστού.
- J2EE: Το λογισμικό Java 2 Enterprise Edition χρησιμοποιείται από διάφορες εταιρίες για τη μεταφορά δεδομένων που βασίζονται σε δομημένα έγγραφα XML μεταξύ τους.
- JavaBeans: Είναι ένα επαναχρησιμοποιήσιμο στοιχείο λογισμικού που μπορεί να χρησιμοποιηθεί για να δημιουργήσει κάποια νέα, προηγμένη εφαρμογή.
- Mobile: Εκτός από τις παραπάνω τεχνολογίες η Java χρησιμοποιείται επίσης σε κινητές συσκευές, πολλά είδη υπηρεσιών και παιχνιδιών που είναι βασισμένα στη Java. Σήμερα όλοι οι κορυφαίοι πάροχοι υπηρεσιών κινητής τηλεφωνίας όπως η Nokia, η Siemens, η Vodafone χρησιμοποιούν τεχνολογίες Java.

Τύποι εφαρμογών Java:

- Web application: Η Java χρησιμοποιείται για τη δημιουργία διαδικτυακών εφαρμογών από πλευράς διακομιστή. Προς το παρόν χρησιμοποιούνται οι τεχνικές servlet, jsp, struts, jsf κλπ.
- Standalone application: Είναι επίσης γνωστή και ως εφαρμογή επιφάνειας εργασίας (desktop) ή εφαρμογή που βασίζεται σε παράθυρα (window based application). Μια εφαρμογή που πρέπει να εγκαταστήσουμε σε κάθε μηχάνημα ή διακομιστή, όπως media player, antivirus κλπ. AWT και Swing χρησιμοποιούνται για τη δημιουργία αυτόνομων εφαρμογών.
- Enterprise application: Μια εφαρμογή που παρέχεται στο κοινό για ιδιαίτερη χρήση, όπως οι τραπεζικές εφαρμογές κλπ. Έχει το πλεονέκτημα της ασφάλειας υψηλού επιπέδου, της εξισορρόπησης φορτίου και της ομαδοποίησης. Στη Java το EJB χρησιμοποιείται για τη δημιουργία επιχειρηματικών εφαρμογών.
- Mobile application: Η Java χρησιμοποιείται για τη δημιουργία λογισμικών εφαρμογών για κινητές συσκευές. Αυτή τη στιγμή το Java ME χρησιμοποιείται για τη δημιουργία εφαρμογών για μικρές συσκευές και επίσης η Java είναι η γλώσσα προγραμματισμού που χρησιμοποιείται για την ανάπτυξη εφαρμογών Google Android.

Δεδομένα χαρακτηριστικά της Java:



- Αντικειμενοστραφής: Στη Java όλα είναι ένα Αντικείμενο, για αυτό το λόγο μπορεί και εύκολα να επεκταθεί γιατί βασίζεται στο μοντέλο Αντικείμενο.
- Ανεξάρτητη πλατφόρμας: C και C++ είναι γλώσσες εξαρτημένες από την πλατφόρμα στην οποία γράφονται εξ ου και τα προγράμματα εφαρμογών που γράφονται σε ένα λειτουργικό σύστημα δεν μπορούν να τρέξουν σε οποιοδήποτε άλλο λειτουργικό σύστημα. Η Java καθώς είναι μια γλώσσα ανεξάρτητη πλατφόρμας, τα προγράμματα εφαρμογών Java που ένα γραμμένα σε ένα λειτουργικό σύστημα μπορούν να τρέξουν και σε οποιοδήποτε άλλο.
- Απλή: Η Java έχει σχεδιαστεί με τέτοιο τρόπο ώστε να είναι εύκολη στην εκμάθηση. Αν καταλάβετε τη βασική ιδέα του αντικειμενοστραφούς προγραμματισμού θα είναι εύκολο να κατανοήσετε και τη Java.
- Ασφαλής: Με την ασφαλή λειτουργία της Java, επιτρέπει την ανάπτυξη συστημάτων χωρίς ιούς, χωρίς παραβίαση. Οι τεχνικές ελέγχου ταυτότητας βασίζονται σε κρυπτογράφηση δημόσιου κλειδιού.
- Αρχιτεκτονικά ουδέτερη: Ο μεταγλωτιστής Java δημιουργεί μια αρχιτεκτονική – ουδέτερη μορφή αρχείου αντικειμένων που καθιστά τον επεξεργασμένο κώδικα εκτελέσιμο σε πολλούς επεξεργαστές, με το σύστημα Java Runtime.
- Φορητή: Το γεγονός ότι η Java είναι αρχιτεκτονικά ουδέτερη και δεν έχει εξαρτήσεις υλοποίησης την καθιστά φορητή – portable. Ο Compiler και η Java έχουν γραφτεί σε ANSI C με ένα καθαρό όριο φορητότητας το οποίο είναι υποσύνολο της POSIX.
- Ισχυρή: Η Java προσπαθεί να εξαλείψει τις επιρρεπείς σε σφάλματα καταστάσεις δίνοντας έμφαση κυρίως στον έλεγχο σφαλμάτων κατά τη μεταγλώττιση και στον έλεγχο χρόνου εκτέλεσης.
- Πολυνηματική: Με τη δυνατότητα πολλαπλών νημάτων της Java είναι δυνατή η εγγραφή προγραμμάτων που μπορούν να κάνουν πολλές εργασίες ταυτόχρονα. Αυτό το σχεδιαστικό χαρακτηριστικό επιτρέπει στους προγραμματιστές να κατασκευάσουν ομαλά διαδραστικές εφαρμογές.
- Διερμηνευμένη: Ο κώδικας byte της Java μεταφράζεται σε μη αυτόματες εντολές μηχανής και δεν αποθηκεύεται οπουδήποτε. Η διαδικασία ανάπτυξης είναι ταχύτερη και αναλυτικότερη, καθώς η σύνδεση είναι μια διαδοχική και ελαφρού βάρους διαδικασία.
- Υψηλή απόδοση: Με τη χρήση μεταγλωτιστών Just – In –Time, η Java επιτρέπει υψηλή απόδοση.
- Διαμοιραζόμενη: Η Java έχει σχεδιαστεί για να διαμοιράζεται στο περιβάλλον του διαδικτύου.
- Δυναμική: Η Java θεωρείται πιο δυναμική από την C ή από τη C++, αφού έχει σχεδιαστεί για να προσαρμόζεται σε ένα εξελισσόμενο περιβάλλον. Τα προγράμματα Java μπορούν να μεταφέρουν ένα εκτεταμένο ποσό πληροφοριών χρόνου εκτέλεσης που μπορούν να χρησιμοποιηθούν για την επαλήθευση και την επίλυση προσπελάσεων σε αντικείμενα κατά την εκτέλεση.

Διαφορετικές εκδόσεις τεχνολογίας Java:



- Java SE: Η Java Standard Edition παρέχει εργαλεία και API που μπορείτε να χρησιμοποιήσετε για να δημιουργήσετε εφαρμογές διακομιστών, εφαρμογές επιφάνειας εργασίας, ακόμα και μικροεφαρμογές. Αυτά τα προγράμματα που αναπτύσσονται χρησιμοποιώντας Java SE μπορούν να τρέξουν σε σχεδόν κάθε δημοφιλές λειτουργικό σύστημα, συμπεριλαμβανομένων των Linux, Macintosh, Solaris και Windows.
- JEE: Βασισμένο στο πλαίσιο θεμελίωσης της πρότυπης έκδοσης, το Java Enterprise Edition, βοηθά στην υπηρεσία διαδικτυακών εφαρμογών, το μοντέλο εξαρτημάτων και την αρχιτεκτονική προσανατολισμένη στις υπηρεσίες επιχειρησιακής κλάσης (SOA).
- JME: Η Java Micro Edition είναι μια συσσώρευση των API Java που χρησιμοποιούνται για την ανάπτυξη λογισμικού για συσκευές όπως τα κινητά τηλέφωνα, τα PDA, τους τηλεοπτικούς δέκτες και τον προγραμματισμό παιχνιδιών. Η πλατφόρμα της μικροεπεξεργασίας αποτελείται γενικά από ένα εύκολο περιβάλλον εργασίας για το χρήστη, ένα ισχυρό μοντέλο ασφάλειας και μια μεγάλη ποικιλία ενσωματωμένων δικτύων για την εκτέλεση εφαρμογών που βασίζονται στη Java.

Για την εκτέλεση της Java χρειάζεται:

- Προγραμματιστές λογισμικού: Να εγκαταστήσουν το JDK (Java Development Kit) το οποίο περιλαμβάνει ένα πλήρες εργαλείο JRE plus, για την ανάπτυξη, τον εντοπισμό σφαλμάτων και την παρακολούθηση εφαρμογών Java.
- Διαχειριστές που εκτελούν εφαρμογές σε ένα διακομιστή: Server JRE (Java RunTime Environment), εγκαθίσταται για την ανάπτυξη εφαρμογών Java σε διακομιστές. Περιλαμβάνει εργαλεία για την παρακολούθηση JVM και εργαλεία που συνήθως απαιτούνται για εφαρμογές διακομιστή, αλλά δεν περιλαμβάνει την ενσωμάτωση του προγράμματος περιήγησης (Plug-in Java), την αυτόματη ενημέρωση και το πρόγραμμα εγκατάστασης.
- Τελικός χρήστης που εκτελεί την Java σε επιφάνεια εργασίας: JRE (περιβάλλον Java Runtime). Καλύπτει τις περισσότερες ανάγκες των τελικών χρηστών και περιέχει όλα όσα απαιτούνται για την εκτέλεση εφαρμογών Java στο σύστημα σας.

Δημοφιλείς επεξεργαστές κώδικα Java:

- Notepad: Ανάλογα το λειτουργικό σύστημα που χρησιμοποιεί ο χρήστης μπορεί να χρησιμοποιήσει οποιοδήποτε απλό πρόγραμμα επεξεργασίας κειμένου όπως το Notepad++, το Sublime, Vim κ.α.
- Netbeans: Είναι ένα ανοιχτού κώδικα Java IDE το οποίο μπορείτε να κατεβάσετε από <https://netbeans.org/index.html>.
- Eclipse: Είναι επίσης ένα ανοιχτού κώδικα Java IDE το οποίο μπορείτε να κατεβάσετε από <http://www.eclipse.org/>.
- Android Studio: Είναι το επίσημο Java IDE κατασκευή εφαρμογών Android. Παρέχει τα ταχύτερα εργαλεία για την κατασκευή εφαρμογών σε κάθε τύπο συσκευής Android. Οι επεξεργασίες κώδικα παγκόσμιας κλάσης, οι εντοπισμοί σφαλμάτων, τα εργαλεία απόδοσης, το ευέλικτο σύστημα δημιουργίας και το άμεσο σύστημα δημιουργίας και ανάπτυξης



επιτρέπουν την κατασκευή υψηλής ποιότητας εφαρμογών. Μπορείτε να το κατεβάσετε από <https://developer.android.com/studio/index.html>.

2.1.3 Kotlin

Το Μάιο του 2017, η ομάδα του Android ανακοίνωσε ότι αρχίζουν να παρέχουν επίσημα υποστήριξη για την γλώσσα προγραμματισμού Kotlin και ότι η Kotlin θα γίνει η επίσημη γλώσσα προγραμματισμού για την πλατφόρμα Android. Είναι μια σύγχρονη γλώσσα προγραμματισμού που τρέχει σε Java Virtual Machine. Όπως ανέφεραν, η Kotlin είναι μια λαμπρά σχεδιασμένη και ώριμη γλώσσα που πιστεύουν ότι θα κάνει την ανάπτυξη του Android πιο γρήγορη και πιο διασκεδαστική. Έχει ήδη υιοθετηθεί από διάφορους σημαντικούς προγραμματιστές όπως - Expedia, Flipboard, Pinterest και άλλοι – για τις εφαρμογές παραγωγής τους. Η Kotlin είναι διαλειτουργική με τη Java και το Android. Αυτή η διαλειτουργικότητα των δυο αυτών γλωσσών προγραμματισμού έκανε και την Kotlin ελκυστική. Το plug-in Kotlin είναι τώρα συνδεδεμένο με το Android studio 3.0 και είναι έτοιμο για άμεση λήψη.

Για να πούμε hello world στην Kotlin:

```
package helloWorld
fun main(args: Array) {
    println("Hello World!")
}
```

Με τη πρώτη ματιά θα δείτε ανακουφιστικά στοιχεία όπως άγκιστρα, κλάσεις, πακέτα, συναρτήσεις και μεθόδους. Αλλά καθώς πηγαίνετε βαθύτερα, θα ανακαλύψετε πως αν και η Kotlin βασίζεται σε γνωστές έννοιες, είναι μοναδικά μοντέρνα, κομψή και ρεαλιστική. Συγκεκριμένα η kotlin είναι εξαιρετικά εκφραστική με ελάχιστη συντακτική τριβή ανάμεσα στις σκέψεις σας και αυτό που πρέπει να πληκτρολογήσετε για να εκφράσετε αυτές τις σκέψεις. Αν κατά τη σύνταξη κώδικα έχετε αναρωτηθεί πολλές φορές για ερωτήσεις που ξεκίνησαν «γιατί πρέπει να...;» θα είστε ευτυχείς να μάθετε ότι στην kotlin η απάντηση σε πολλές από αυτές τις ερωτήσεις είναι «δεν το κάνετε!».

Για παράδειγμα, ίσως έχετε αναρωτηθεί γιατί θα πρέπει να πληκτρολογήσετε ένα σορό getters και setters καθώς και να κάνετε overriding equals(), hashCode() και toString() κατά την υλοποίηση μιας απλής κλάσης. Εδώ είναι ένα τυπικό παράδειγμα από τη γλώσσα προγραμματισμού Java.

```
public class Customer {
    private String name;
    private String email;
    private String company;

    public Customer(String name) {
        this(name, "", "");
    }
}
```




```
public Customer(String name, String email) {
    this(name, email, "");
}

public Customer(String name, String email, String company) {
    this.name = name;
    this.email = email;
    this.company = company;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public String getEmail() {
    return email;
}

public void setEmail(String email) {
    this.email = email;
}

public String getCompany() {
    return company;
}

public void setCompany(String company) {
    this.company = company;
}
```



```
@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (o == null || getClass() != o.getClass()) return false;

    Customer customer = (Customer) o;

    if (name != null ? !name.equals(customer.name) : customer.name != null) return false;
    if (email != null ? !email.equals(customer.email) : customer.email != null) return false;
    return company != null ? company.equals(customer.company) : customer.company == null;
}
```

```
@Override
public int hashCode() {
    int result = name != null ? name.hashCode() : 0;
    result = 31 * result + (email != null ? email.hashCode() : 0);
    result = 31 * result + (company != null ? company.hashCode() : 0);
    return result;
}
```

```
@Override
public String toString() {
    return "Customer{" +
        "name=" + name + '\n' +
        ", email=" + email + '\n' +
        ", company=" + company + '\n' +
        '}';
}
}
```

Στην Kotlin δεν χρειάζεται να πληκτρολογήσετε όλα αυτά. Αυτή η ενιαία γραμμή είναι ισοδύναμη με ολόκληρη την παραπάνω κλάση.

```
data class Customer(var name: String, var email: String = "",
    var company: String = "")
```



Θα μπορούσε να γίνει εκτενέστερή αναφορά στη γλώσσα προγραμματισμού Kotlin, αλλά το σίγουρο είναι ότι αρχίζει να μπαίνει δυναμικά στον κόσμο του προγραμματισμού και ειδικά στις android εφαρμογές. Πρόκειται να γραφτούν πολλές αναλύσεις, να γίνουν πολλές συγκρίσεις και να απασχολήσει πολύ κόσμο του προγραμματισμού εφαρμογών.

2.2 MongoDB

Η mongoDB είναι μια εγγραφοκεντρική βάση δεδομένων που παρέχει, υψηλή απόδοση, υψηλή διαθεσιμότητα και εύκολη επεκτασιμότητα. Η mongoDB λειτουργεί υπό την έννοια της «συλλογής» και του «εγγράφου». Ο εξυπηρετητής της βάσης και τα υποστηρικτικά εργαλεία που διατίθενται μαζί με τη mongoDB είναι «ανοιχτού κώδικα» και υποστηρίζονται από μια μεγάλη κοινότητα προγραμματιστών.

Το αντίστοιχο της εγγραφής (record) των σχεσιακών βάσεων, ονομάζεται έγγραφο (document) στη mongoDB και περιέχει πεδία που απαρτίζονται από ζεύγη κλειδιών – τιμών (key – value pairs). Τα mongoDB έγγραφα είναι παρόμοια με τα αντικείμενα της JSON. Οι τιμές των πεδίων μπορεί να περιέχουν άλλα έγγραφα, πίνακες και πίνακες από έγγραφα. Το αντίστοιχο του πίνακα (table) των σχεσιακών βάσεων ονομάζεται συλλογή (collection) στη mongoDB.

```
{
  name: "sue",
  age: 26,
  status: "A",
  groups: [ "news", "sports" ]
}
```

← field: value
← field: value
← field: value
← field: value

Εικόνα 2. Παράδειγμα εγγράφου mongoDB



Τα πλεονεκτήματα της χρήσης εγγράφων είναι ότι:

- Τα έγγραφα (δηλαδή αντικείμενα) αντιστοιχούν στους native (έμφυτους) τύπους δεδομένων πολλών γλωσσών προγραμματισμού όπως η Javascript.
- Η δυνατότητα για ενσωματωμένα έγγραφα και πίνακες μειώνει την ανάγκη για δαπανηρές πράξεις τύπου join.
- Η δυνατότητα χρήσης αδόμητου ή ημιδομημένου σχήματος υποστηρίζει την εύκολη επέκταση των εφαρμογών.

Η mongoDB παρέχει υψηλή απόδοση, υψηλή διαθεσιμότητα και αυτόματη κλιμάκωση. Μερικά από τα χαρακτηριστικά της βάσης που συντελούν τα αναφερθέντα είναι:

- Υποστήριξη ενσωμάτωσης αντικειμένων στο μοντέλο δεδομένων της (nested documents), δυνατότητα η οποία μειώνει την ανάγκη για πολλαπλές αναγνώσεις / εγγραφές στους χώρους αποθήκευσης.
- Παρέχει δείκτες (indexes) οι οποίοι μπορούν να δεικτοδοτήσουν κλειδιά και σε ενσωματωμένα έγγραφα.
- Η υπηρεσία λειτουργίας αντιγράφων της βάσης (replication) παρέχει αυτόματη ανάκαμψη από βλάβες (automatic failover) και πλεονασμό δεδομένων (data redundancy).
- Παρέχει οριζόντια κλιμάκωση ως βασική της υπηρεσία και παρέχει τη δυνατότητα κατακερματισμού των δεδομένων (sharding) σε ένα σύνολο (cluster) υπολογιστών.

2.2.1 MongoDB και Node.js

Όπως προαναφέρθηκε, η mongoDB κάνει την ανάπτυξη εφαρμογών να φαίνεται εύκολη. Το μοντέλο εγγράφων της mongoDB είναι απλό για τους προγραμματιστές να το μάθουν και να το χρησιμοποιούν, παρέχοντας παράλληλα όλες τις δυνατότητες που χρειάζονται για να καλύψουν τις πιο σύνθετες απαιτήσεις σε οποιαδήποτε κλίμακα. Παρέχονται οδηγοί για 10 γλώσσες προγραμματισμού και οι κοινότητα έχει δημιουργήσει δεκάδες ακόμα. Στη συνέχεια παρατίθενται μερικά παραδείγματα ανάπτυξης με τη γλώσσα Node.js σε συνδιασμό με τη mongoDB:

```
//1. Connect to MongoDB instance running on localhost
// Connection URL
var url = 'mongodb://localhost:27017/test';

co(function*() {
  const db = yield MongoClient.connect(url);
```

Emergency Call Android Application Development



```
console.log("Connected successfully to server");

yield insertDocuments(db, null);
yield findDocuments(db, null);
yield indexCollection(db, null);
yield aggregateDocuments(db, null);

db.close();
}).catch(err => console.log(err));

// 2. Insert
var insertDocuments = function(db, callback) {
  return co(function*() {
    const results = yield db.collection('restaurants').insertMany([
      {
        "name": "Sun Bakery Trattoria",
        "stars": 4,
        "categories": [
          "Pizza", "Pasta", "Italian", "Coffee", "Sandwiches"
        ]
      }, {
        "name": "Blue Bagels Grill",
        "stars": 3,
        "categories": [
          "Bagels", "Cookies", "Sandwiches"
        ]
      }, {
        "name": "Hot Bakery Cafe",
        "stars": 4,
        "categories": [
          "Bakery", "Cafe", "Coffee", "Dessert"
        ]
      }, {
        "name": "XYZ Coffee Bar",
        "stars": 5,
```



```
"categories": [
  "Coffee", "Cafe", "Bakery", "Chocolates"
]
}, {
  "name": "456 Cookies Shop",
  "stars": 4,
  "categories": [
    "Bakery", "Cookies", "Cake", "Coffee"
  ]
}
]);

console.log(results)
return results;
});
};

// 3. Query Collection
var findDocuments = function(db) {
  return co(function*() {
    // Get the documents collection
    const collection = db.collection('restaurants');
    const docs = yield collection.find({}).toArray();
    console.log("Found the following records");
    console.log(docs)
    return docs;
  });
};

// 4. Create Index
var indexCollection = function(db) {
  return co(function*() {
    const results = yield db.collection('restaurants').createIndex(
      { "name": 1 },
      null
    );
  });
};
```



```
);

console.log(results);
return results;
});
};

// 5. Perform Aggregation
var aggregateDocuments = function(db, callback) {
return co(function*() {
const results = yield db.collection('restaurants')
.aggregate([
{$match: { "categories": "Bakery" }},
{$group: { '_id': "$stars", 'count': { '$sum': 1 } }}
])
.toArray();

console.log(results)
return results;
});
};
```

2.2.2 MongooseJS

Η mongooseJS είναι ένα πακέτο μοντελοποίησης αντικειμένων για τη Node.js το οποίο λειτουργεί ουσιαστικά σαν ORM. Η mongoose μας επιτρέπει να έχουμε πρόσβαση στις εντολές της mongoDB για το CRUD απλά και εύκολα. Για να χρησιμοποιήσουμε τη mongoose θα πρέπει να την προσθέσουμε στο Node έργο που αναπτύσσουμε κάθε φορά χρησιμοποιώντας την ακόλουθη εντολή:

```
$ npm install mongoose --save
```

Αφού έχουμε εγκαταστήσει το πακέτο της mongooseJS μένει να την εισάγουμε στο πρόγραμμα που αναπτύσσουμε:

```
var mongoose = require('mongoose');
```

Πρέπει επίσης να συνδεθούμε σε μια βάση δεδομένων (τοπικά, ή φιλοφενούμενη).

```
mongoose.connect('mongodb://localhost/myappdatabase');
```



Πριν μπορέσουμε να χειριστούμε τις λειτουργίες CRUD, θα χρειαστούμε ένα μοντέλο mongoose. Αυτά τα μοντέλα είναι κατασκευαστές που ορίζουμε. Αντιπροσωπεύουν έγγραφα τα οποία μπορούν να αποθηκευτούν και να ανακτηθούν από τη βάση δεδομένων μας. Το σχήμα Mongoose είναι αυτό που χρησιμοποιείται για τον ορισμό των χαρακτηριστικών για τα έγγραφά μας. Οι μέθοδοι μπορούν επίσης να οριστούν σε ένα σχήμα mongoose.

```
// grab the things we need
var mongoose = require('mongoose');
var bcrypt = require('bcryptjs');

var UsersSchema = new mongoose.Schema(
  {
    id : String ,
    username : {
      type:String ,
      unique: true
    },
    password : String,
    email : {
      type:String ,
      unique: true
    },
    lastname : String ,
    firstname : String ,
    history : String ,
    blood : String ,
    cellphone : String ,
    scellphone : String ,
    home : String ,
    language : String
  }
);
```

```
UsersSchema.methods.comparePassword = function(candidatePassword, cb) {
  bcrypt.compare(candidatePassword, this.password, function(err, isMatch) {
```




```
    if (err) return cb(err);
    cb(null, isMatch);
  });
};

UsersSchema.pre('save', function(next) {
  var user = this;
  if (!user.isModified('password')) return next();
  bcrypt.genSalt(10, function(err, salt) {
    if (err) return next(err);
    bcrypt.hash(user.password, salt, function(err, hash) {
      if (err) return next(err);
      user.password = hash;
      next();
    });
  });
});

var User = mongoose.model('User', UsersSchema);
// make this available to our users in our Node applications
module.exports = User;
```

2.3 Backend - Node.js

Το Node.js δημιουργήθηκε από τον Ryan Dahl το 2009. Η δημιουργία και η συντήρηση του έργου χορηγήθηκε από την εταιρία Joyent. Η ιδέα για την ανάπτυξη του node προήλθε από την ανάγκη του Ryan Dahl να βρει τον πιο αποδοτικό τρόπο να ενημερώνει τον χρήστη σε πραγματικό χρόνο για την κατάσταση ενός αρχείου που ανέβαζε στο διαδίκτυο. Επίσης επηρεάστηκε από το Mongrel του Zed Shaw. Επιπροσθέτως μετά από αποτυχημένα έργα σε C, Lua, Haskell η κυκλοφορία της μηχανής V8 (V8 JavaScript Engine) της Google τον ώθησε να ασχοληθεί με την Javascript.

Το Node χαρακτηρίζεται από την έμφαση στην ασύγχρονη επικοινωνία μεταξύ των υπολογιστικών πόρων. Αυτό επιτυγχάνεται με την χρήση συμβάντων (events) που προσφέρει η Javascript και ονομάζονται callbacks. Για παράδειγμα όταν ένας περιηγητής ιστού φορτώσει πλήρως ένα αρχείο, ένας χρήστης πατάει κάποιο κουμπί, ολοκληρώνεται ένα αίτημα AJAX, τα συμβάντα αυτά πυροδοτούν ένα συγκεκριμένο callback.



Αυτό με την σειρά του επιτρέπει την ροή του κώδικα χωρίς να αφήνει ανενεργό τον επεξεργαστή προκειμένου να εκτελεστεί μια λειτουργία, όπως μια επιτυχής ανάγνωση αρχείου από τον δίσκο.

```
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World\n');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Εικόνα 3. Παράδειγμα δημιουργίας node.js server

Ο παραπάνω κώδικας δημιουργεί έναν εξυπηρετητή ο οποίος τρέχει τοπικά στη θύρα «3000». Ο χρήστης μπορεί στη μπάρα του φυλλομετρητή να γράψει τη διεύθυνση url <http://127.0.0.1:3000> και να του εμφανίσει μια σελίδα με το “Hello World” σαν κείμενο. Για την εκτέλεση του παραπάνω προγράμματος απαιτείται η αποθήκευση του παραπάνω κώδικα σε ένα αρχείο (έστω example.js) με την εντολή:

```
$ node example.js
```

```
Server running at http://127.0.0.1:3000/
```

Ένα παράδειγμα δημιουργίας αρχείου είναι:

```
var fs = require('fs')
```

```
fs.writeFile("D:\myfile.txt", "Hello World", function(err, data) {
  if (err) throw err;
  console.log(data);
})
```



Ο παραπάνω κώδικας χρησιμοποιεί την έτοιμη βιβλιοθήκη του node για την επεξεργασία αρχείων με την εντολή "require([library_name])" και δημιουργεί ή αντικαθιστά στην περίπτωση που υπάρχει στο δίσκο D το αρχείο "myfile.txt" με περιεχόμενο "hello world". Όταν ολοκληρωθεί η λειτουργία αυτή της εγγραφής, πυροδοτείται το callback το οποίο ορίστηκε, και αυτό με την σειρά του είτε στέλνει ένα μήνυμα προβλήματος στην περίπτωση αποτυχίας, είτε στέλνει τα δεδομένα του αρχείου στην κονσόλα του χρήστη.

Η κοινότητα έχει δημιουργήσει ένα ολόκληρο οικοσύστημα από βιβλιοθήκες που προορίζονται ή είναι συμβατές με το node. Ανάμεσά τους εργαλεία που ξεχώρισαν όπως το node-mysql, το MongoDB και το Express παίζουν σημαντικό ρόλο υποστηρίζοντας την ασύγχρονη διάδραση με τις παραδοσιακές και NoSQL μεθόδους βάσεων δεδομένων. Αυτό επιτυγχάνεται με την χρήση του node package manager το οποίο επιτρέπει την εγκατάσταση των παραπάνω βιβλιοθηκών. Χρησιμοποιείται συνήθως σε εφαρμογές Chat, Proxy, Http Server καθώς και για παρακολούθηση εφαρμογών και του συστήματος (monitoring).

2.3.1 Express.js

Express.js είναι ένα web application framework για Node.js, κυκλοφόρησε ως δωρεάν και ανοιχτού κώδικα λογισμικό υπό την άδεια MIT. Έχει σχεδιαστεί για τη δημιουργία διαδικτυακών εφαρμογών και APIs. Είναι το βασικό πρότυπο server framework για Node.js. Express είναι το backend μέρος του MEAN stack, μαζί με τη βάση δεδομένων MongoDB και την AngularJS frontend framework.

Έχοντας πρώτα εγκαταστήσει τη Node.js για να εγκατασταθεί το express framework πρέπει να δημιουργηθεί ένας κατάλογος (directory) όπου θα υπάρχει η εφαρμογή και ο κατάλογος που θα δουλέψουμε.

```
$ mkdir myapp
```

```
$ cd myapp
```

Στη συνέχεια πρέπει να γίνει χρήση της εντολής npm init ώστε να δημιουργηθεί ένα αρχείο package.json για την εφαρμογή.

```
$ npm init
```

Πληκτρολογούμε app.js ή όποιο θέλουμε να είναι το όνομα του αρχείου. Αν θέλουμε το όνομα αρχείου να είναι index.js πληκτρολογούμε RETURN ώστε να αποδεχτούμε το προεπιλεγμένο όνομα αρχείου. Τώρα εγκαθιστούμε το Express στο MyApp κατάλογο και αποθηκεύουμε στη λίστα εξαρτήσεων.

```
$ npm install express --save
```

Για να εγκαταστήσουμε το Express προσωρινά χωρίς να προστεθεί στη λίστα εξαρτήσεων πρέπει να παραλείψτε την επιλογή --save.

Παρατίθεται στη συνέχεια το αρχείο package.json από την υλοποίηση του Web app του έργου. Εκεί βλέπουμε σημαντικές πληροφορίες για τις βιβλιοθήκες που χρησιμοποιήθηκαν και κατά το npm install διαβάζονται και γίνονται "download".

```
{
```

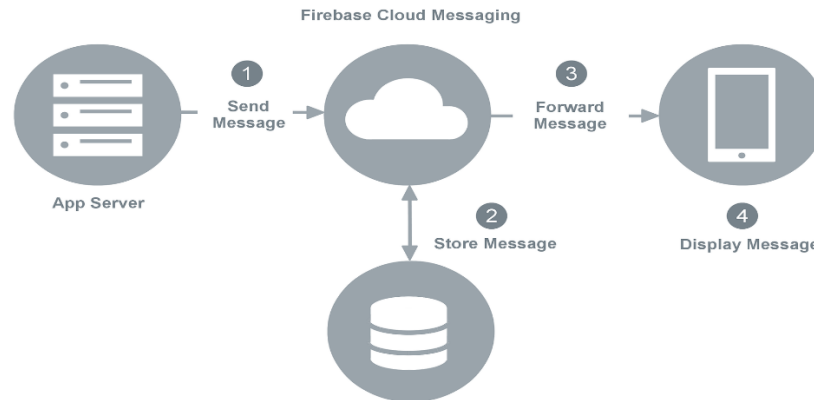


```
"name": "thalia_thesis",
"version": "1.0.0",
"description": "",
"main": "index.js",
"scripts": {
  "start": "node index.js",
  "postinstall": "bower install"
},
"author": "",
"license": "ISC",
"dependencies": {
  "express": "^4.14.0",
  "body-parser": "^1.13.3",
  "bcryptjs": "^2.2.1",
  "npm": "^3.3.8",
  "bower": "*",
  "multer": "^1.2.0",
  "multiparty": "*",
  "mongoose": "^4.1.0"
}
}
```

2.3.2 Firebase

Η Firebase είναι μια backend πλατφόρμα για τη δημιουργία εφαρμογών Web, Android και Ios η οποία παρέχει τρεις βασικές υπηρεσίες: βάση δεδομένων σε πραγματικό χρόνο, έλεγχο ταυτότητας χρήστη και φιλοξενία (hosting). Δηλαδή, υποστηρίζει δεδομένα JSON και όλοι χρήστες που συνδέονται σε αυτή λαμβάνουν ζωντανές ενημερώσεις μετά από κάθε αλλαγή. Μπορεί να χρησιμοποιηθεί ανώνυμα, κωδικό πρόσβασης, ή αυθεντικοποίηση με χρήση των λογαριασμών διαφόρων μέσων κοινωνικής δικτύωσης. Οι εφαρμογές μπορούν να αναπτυχθούν μέσω ασφαλούς σύνδεσης με τους Firebase servers.

Για να χρησιμοποιηθεί η Firebase σε υπάρχουσα εφαρμογή, στην περίπτωση μας android εφαρμογή, επειδή χρησιμοποιήθηκε η τελευταία έκδοση του android studio, υπήρχε η δυνατότητα να γίνει χρήση το Firebase assistant για να συνδέσουμε την εφαρμογή μας με την Firebase. Έτσι εγκαθίστανται αυτόματα όλες οι grandle dependencies. Με τη χρήση της Firebase υπάρχει ζωντανή αλληλεπίδραση – επικοινωνία μεταξύ του διαχειριστή του Web app και του χρήστη της Android εφαρμογής.



Εικόνα 4. Περιγραφή λειτουργιών Firebase

2.3.3 FrontEnd – AngularJS

AngularJS είναι ένα ολοκληρωμένο frontend web application framework βασισμένο στη Javascript. Διατηρείται κυρίως από την Google και από την κοινότητα ιδιωτών και επιχειρήσεων για την αντιμετώπιση των προκλήσεων που ανέκυψαν κατά την ανάπτυξη εφαρμογών μιας σελίδας. Η AngularJS αποτελεί ένα διαρθρωτικό framework για δυναμικές διαδικτυακές εφαρμογές. Αυτό επιτρέπει να χρησιμοποιηθεί η HTML ως πρότυπη γλώσσα και επιτρέπει να επεκταθεί η σύνταξη της HTML ώστε να εκφράσει τις συνιστώσες της εφαρμογής συνοπτικά και με σαφήνεια. Τα δεσμευτικά δεδομένα και οι ενέσεις εξαρτήσεων της Angular μειώνουν κατά πολύ τον κώδικα που θα έπρεπε να γραφτεί σε διαφορετική περίπτωση. Όλα αυτά συμβαίνουν στο πρόγραμμα περιήγησης (browser), γεγονός που το καθιστά ιδανικό συνεργάτη με οποιαδήποτε τεχνολογία διακομιστή.

Η Angular είναι αυτό που η HTML θα ήταν αν είχε σχεδιαστεί για εφαρμογές. Η HTML είναι μια πολύ καλή δηλωτική γλώσσα για στατικά έγγραφα. Η αναντιστοιχία μεταξύ δυναμικών εφαρμογών και στατικών εγγράφων στην HTML συχνά επιλύονται με:

- **Μια βιβλιοθήκη (library):** ένα σύνολο συναρτήσεων που είναι χρήσιμες όταν δημιουργούμε διαδικτυακές εφαρμογές. Ο κώδικας είναι υπεύθυνος να καλέσει τη βιβλιοθήκη όταν το κρίνει σκόπιμο. (πχ. jQuery)
- **Frameworks:** μια συγκεκριμένη υλοποίηση μιας διαδικτυακής εφαρμογής όπου ο κώδικας συμπληρώνει τις λεπτομέρειες. Το framework είναι υπεύθυνο και καλεί μέσα από τον κώδικα όταν χρειαστεί μια συγκεκριμένη εφαρμογή, πχ durantal, ember κ.α.

Η Angular ακολουθεί μια διαφορετική προσέγγιση. Προσπαθεί να ελαχιστοποιήσει την αναντιστοιχία μεταξύ κεντρικών HTML εγγράφων και τι χρειάζεται μια εφαρμογή με τη δημιουργία νέων HTML κατασκευών. Η Angular μαθαίνει το πρόγραμμα περιήγησης νέα σύνταξη μέσω ενός μορφώματος που αποκαλούμαι νιρεκτίβες. Τα παραδείγματα περιλαμβάνουν:

- Τα δεδομένα δέσμευσης, όπως στο `{{}}`
- Δομές ελέγχου DOM για την επανάληψη, εμφάνιση και απόκρυψη DOM fragments.
- Υποστήριξη για φόρμες και επιβεβαίωση φόρμας.



- Επισυνάπτοντας νέα συμπεριφορά σε DOM στοιχεία, όπως χειρισμός γεγονότος στον DOM.
- Ομαδοποίηση της HTML σε επαναχρησιμοποιήσιμα στοιχεία.

Παράδειγμα με AngularJS:

```
<!DOCTYPE html>
<html>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
<body>
<div ng-app="">
  <p>Name: <input type="text" ng-model="name"></p>
  <p ng-bind="name"></p>
</div>
</body>
</html>
```

2.4 Security

Η ασφάλεια είναι το κομμάτι του Development που όλοι συμφωνούν ότι είναι πολύ σημαντικό αλλά λίγοι ασχολούνται όπως θα έπρεπε με αυτό. Τα τελευταία χρόνια παρόλα αυτά, ο τομέας της ασφάλειας έχει αρχίσει να παίρνει την προσοχή που του χρειάζεται. Στην NodeJS η ασφάλεια δεν έχει ανελιχθεί ακόμα ως πρωτεύον ζήτημα, λόγω του νέου της τεχνολογίας, αλλά όσο αυτή κερδίζει χώρο στην αγορά, τόσο θα εξελίσσεται ο εν λόγω τομέας τομέα. Παρακάτω θα δούμε μερικές από τις τακτικές που πρέπει να ακολουθούν οι χρήστες τόσο στο Back-End κομμάτι όσο και στο Front-End.

Αρχικά, το μεγάλο πρόβλημα της Node είναι το εργαλείο npm που όπως έχουμε δει προσφέρει πρόσβαση σε χιλιάδες άλλες βιβλιοθήκες και εργαλεία που βοηθούν απίστευτα τη διαδικασία του Development. Ο χρυσός κανόνας χρήσης του npm είναι να γνωρίζεις τι κατεβάζεις καθώς, αυτές οι βιβλιοθήκες μπορούν να έχουν δημιουργηθεί από τον οποιονδήποτε. Αυτό σημαίνει πως μία τέτοια βιβλιοθήκη μπορεί ταυτόχρονα να κάνει πολλά άλλα πράγματα στο Back-End περιβάλλον, ή κάτι σίγουρα πιο πιθανό, να έχει κάποια ευπάθεια η οποία να επιτρέπει σε χρήστες να δημιουργήσουν πρόβλημα στην εφαρμογή μας. Γι' αυτό, θα πρέπει να είμαστε πολύ σίγουροι για τον κώδικα τρίτων ατόμων που ενσωματώνουμε στις εφαρμογές μας. Ως βοηθητικό αυτού του προβλήματος δημιουργήθηκε το Node Security Project, μέσα από το οποίο ελέγχονται όσα περισσότερα από τα εργαλεία που υπάρχουν στο npm γίνεται, ώστε να βρίσκονται οι

Emergency Call Android Application Development



ευπάθειές τους και να λύνονται. Μέσω αυτού έχουν βρεθεί και λυθεί τέτοια προβλήματα σε μεγάλα εργαλεία που είδαμε και παραπάνω όπως το Hari και το Express.

Σε συνέχεια του παραπάνω προβλήματος, δεν πρέπει να αμελούμε την αναβάθμιση αυτών των βιβλιοθηκών. Πολλές φορές όταν η εφαρμογή μας δουλεύει σωστά με μία συγκεκριμένη έκδοση μιας βιβλιοθήκης δεν μπαίνουμε στην διαδικασία αναβάθμισής της, γιατί μπορεί να χρειαστούν αλλαγές για να δουλέψει πάλι σωστά. Αυτό είναι λάθος γιατί οι αναβαθμίσεις των βιβλιοθηκών εκτός από νέα χαρακτηριστικά συχνά φέρνουν και διορθώσεις σε ευπάθειες των μέχρι τότε εκδόσεων.

Μετά, δεν πρέπει να ξεχνάμε πως η Node παραμένει να είναι JavaScript, οπότε και όλες τις ευπάθειες που αυτή έχει στο Front-End, η Node τις περνάει στο Back-End, όπου και μπορούν να δημιουργήσουν πολύ μεγαλύτερα προβλήματα. Για παράδειγμα, η χρήση της `eval` μπορεί να προκαλέσει σοβαρά προβλήματα και θα πρέπει να αποφεύγεται, όπως γίνεται δηλαδή και στην JavaScript μέχρι τώρα.

Αντίστοιχα, όπως η JavaScript στον browser όταν βρει κάποιο error σταματάει να δουλεύει, έτσι και στη Node, όταν βρεθεί κάποιο error τότε ο server θα πέσει. Έτσι, αν κάποιος βρει έναν τρόπο για να κάνει την JavaScript στο Back-End να βρει error, για παράδειγμα στέλνοντας λάθος δεδομένα, θα μπορεί να ρίχνει συνέχεια τον server και ουσιαστικά να κάνει επιτυχημένες DOS (Denial of Service) επιθέσεις. Γι' αυτό, πρέπει είτε να προσπαθούμε να προλάβουμε τα σφάλματα αυτά, πράγμα δύσκολο, είτε να ελέγχουμε συνέχεια την κατάσταση του server ώστε με μια αυτοματοποιημένη διαδικασία να ξαναλειτουργεί. Ένα εργαλείο που κάνει ακριβώς αυτή τη δουλειά είναι το `forever`.

Τέλος, υπάρχουν κάποια ακόμα πράγματα που μπορούμε να κάνουμε για να αποφύγουμε ένα συγκεκριμένο κομμάτι επιθέσεων. Οι επιθέσεις τύπου CSRF (Cross-Site Request Forgery) προσπαθούν να εκμεταλλευτούν την υπάρχουσα σύνδεση του χρήστη με κάποια άλλη εφαρμογή και την ύπαρξη του ενεργού cookie αυτού στον browser. Μέσω αυτού, μπορεί να γίνει κλίση της άλλης εφαρμογής χωρίς ο χρήστης να το καταλάβει. Ο τρόπος για να αποφευχθεί αυτό είναι απλός. Η εφαρμογή στο Back-End δημιουργεί ένα τυχαίο κλειδί και το στέλνει στο Front-End. Από εκεί και πέρα, κάθε επικοινωνία μεταξύ Back-End και Front-End θα πρέπει να περιλαμβάνει το συγκεκριμένο κλειδί. Αν κάποια άλλη εφαρμογή προσπαθήσει να καλέσει το Back-End της δικιάς μας, δεν θα γνωρίζει το συγκεκριμένο κλειδί άρα δεν θα μπορέσει να αποκτήσει πρόσβαση.

Το CSP (Content Security Policy) είναι ουσιαστικά μια λίστα από επιτρεπτές πηγές για κώδικα JavaScript, CSS, φωτογραφίες και άλλα για την εφαρμογή μας. Δηλαδή, δημιουργώντας αυτή την λίστα, ένα αρχείο JavaScript που προέρχεται από εξωτερική πηγή, αν αυτή η πηγή δεν είναι δηλωμένη τότε δεν θα τρέξει ο κώδικάς του. Αντίστοιχα, αν μια φωτογραφία πρέπει να φορτωθεί από άλλον server, ο οποίος δεν έχει δηλωθεί σε αυτή τη λίστα, δεν θα εμφανιστεί μπροστά στην εφαρμογή. Με αυτό το εργαλείο αποκτά μεγάλη ασφάλεια η εφαρμογή, αφού μπορούμε να ελέγξουμε ακριβώς από ποιες πηγές θα δεχόμαστε να τρέχει κώδικας στην εφαρμογή μας.

Δύο άλλες σημαντικές ρυθμίσεις είναι το HTTPS , για ασφαλή σύνδεση και κρυπτογράφηση των δεδομένων που μεταφέρονται, και την επιλογή X-Frame, με την οποία μπορούμε να αποτρέψουμε άλλες εφαρμογές να εμφανίζουν – ενσωματώνουν τη δικιά μας εφαρμογή μέσα σε iframes.

Αυτά είναι μερικά από τα βήματα που πρέπει να ακολουθούμε στο Back-End κομμάτι. Στο Front-End κομμάτι το μεγάλο πρόβλημα είναι οι επιθέσεις τύπου XSS (Cross Site Scripting). Οι επιθέσεις αυτές προσπαθούν να προσθέσουν κακόβουλο κώδικα σε κομμάτια της σελίδας που θα δουν αργότερα άλλοι χρήστες. Η λύση σε αυτό το πρόβλημα είναι ο συνεχής έλεγχος των εισαγωγών του χρήστη και η αποφυγή

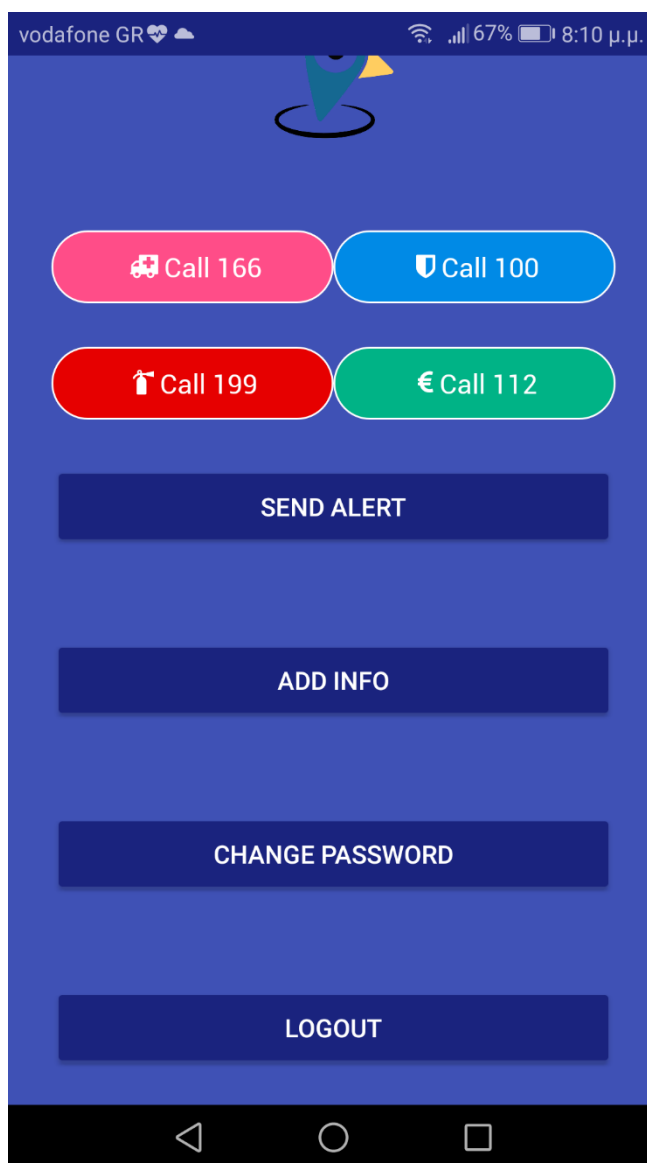


των επικίνδυνων κομματιών του. Αυτό πρέπει να γίνεται, τόσο στις εισαγωγές του χρήστη όσο και στα δεδομένα που έρχονται στο Front-End από APIs, είτε από το Back-End της ίδιας της εφαρμογής είτε από κάποιο εξωτερικό.

2.5 Android app – Περιγραφή εφαρμογής

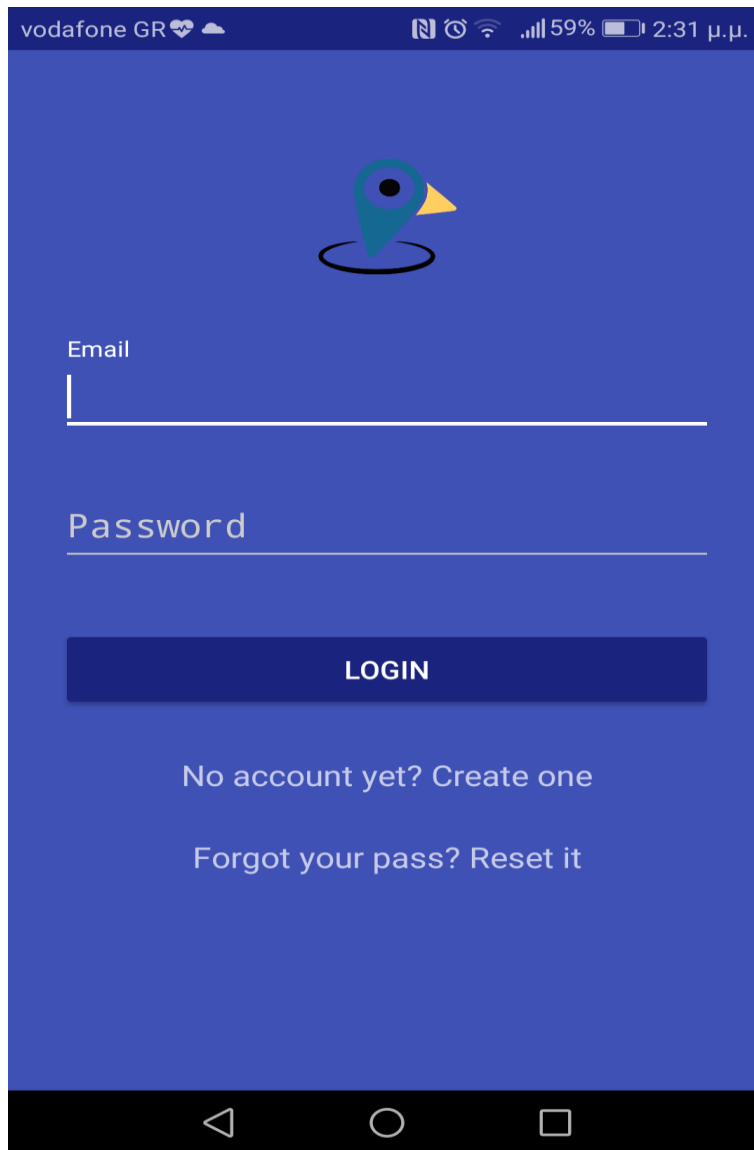
Για να γίνουν όλα τα παραπάνω πιο κατανοητά, θα δημιουργήσουμε μια εφαρμογή αποστολής ειδοποιήσεων σε περιπτώσεις έκτακτης ανάγκης, χρησιμοποιώντας όλες τις τεχνολογίες που αναλύθηκαν στις προηγούμενες ενότητες. Τεχνολογίες που αφορούσαν το backend, το frontend και τη Βάση Δεδομένων της εφαρμογής.

Αρχικά, ο χρήστης θα μπορεί να συνδεθεί στην εφαρμογή είτε με το email του είτε με κάποιο social λογαριασμό του. Μετά από την επιτυχή εγγραφή και σύνδεση του χρήστη, αυτός θα έχει πρόσβαση στην κεντρική σελίδα της εφαρμογής από την οποία θα βλέπει μια σύνοψη των πληροφοριών που διαθέτει και θα μπορεί να περιηγηθεί στα επιμέρους κομμάτια της από όπου θα μπορεί να στείλει ειδοποίηση έκτακτης ανάγκης είτε στέλνοντας τα στοιχεία του και την τοποθεσία που βρίσκεται, είτε στέλνοντας όλα αυτά συν μια φωτογραφία. Επίσης, από την κεντρική σελίδα της εφαρμογής έχει τη δυνατότητα να προσθέσει προσωπικές χρήσιμες πληροφορίες, οι οποίες θα αποστέλλονται κατά την κάθε αποστολή ειδοποίησης έκτακτης ανάγκης. Παρέχεται επίσης, η δυνατότητα αλλαγής του κωδικού πρόσβασης στην εφαρμογή και υπάρχει και η επιλογή αποσύνδεσης από αυτή. Οι βασικότερες λειτουργίες που παρέχονται στην κεντρική οθόνη της εφαρμογής είναι η κλήση προς το 112 Ευρωπαϊκό αριθμό κλήσεων έκτακτης ανάγκης, αλλά και απευθείας κλήσεις προς σημαντικές υπηρεσίες όπως η Πυροσβεστική, το ΕΚΑΒ και η αστυνομία.



Εικόνα 5. Κεντρική οθόνη της mobile εφαρμογής

Για να φτάσουμε όμως στην κεντρική σελίδα της εφαρμογής θα πρέπει πρώτα να κάνουμε εγγραφή και να δημιουργήσουμε ένα λογαριασμό χρήστη. Αυτό επιτυγχάνεται στην πρώτη σελίδα της εφαρμογής.



Εικόνα 6. Αρχική οθόνη της mobile εφαρμογής

Στη σελίδα αυτή, υπάρχουν κάποιες δυνατές επιλογές. Σε περίπτωση που ο χρήστης είχε κάνει ήδη την εγγραφή και έχει ήδη λογαριασμό, με τη χρήση του email και το κωδικού πρόσβασης μπορεί να συνδεθεί και να εισέλθει στην κεντρική σελίδα της εφαρμογής. Αν ο χρήστης εισέρχεται για πρώτη φορά στην εφαρμογή τότε θα πρέπει να δημιουργήσει ένα λογαριασμό. Ένα δείγμα κώδικα της υλοποίησης της συγκεκριμένης λειτουργίας ακολουθεί στη συνέχεια παραθέτοντας τη `LoginActivity.java`:

```
package com.sourcey.alertsignal;  
  
import android.app.ProgressDialog;
```



```
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.util.Patterns;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import com.google.gson.Gson;
import com.google.gson.JsonObject;
import com.google.gson.JsonParser;
import java.io.IOException;
import java.util.HashMap;
import butterknife.Bind;
import butterknife.ButterKnife;
import okhttp3.ResponseBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class LoginActivity extends AppCompatActivity {
    private static final String TAG = "LoginActivity";
    private static final int REQUEST_SIGNUP = 0;
    private static final int REQUEST_RESETPASS = 1;

    @Bind(R.id.input_email)
    EditText _emailText;
    @Bind(R.id.input_password)
    EditText _passwordText;
    @Bind(R.id.btn_login)
    Button _loginButton;
    @Bind(R.id.link_signup)
    TextView _signupLink;
```



```
@Bind(R.id.link_reset)
TextView linkReset;

SessionHandler session;
HashMap<String, String> userdetails;

UserModel loginUser = new UserModel();

ApiInterface apiService = ApiClient.getClient().create(ApiInterface.class);

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);
    ButterKnife.bind(this);

    _loginButton.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            login();
        }
    });

    _signupLink.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            // Start the Signup activity
            Intent intent = new Intent(getApplicationContext(), SignupActivity.class);
            startActivityForResult(intent, REQUEST_SIGNUP);
        }
    });

    linkReset.setOnClickListener(new View.OnClickListener() {
```



```
@Override
public void onClick(View v) {
    // Start the Signup activity
    Intent intent = new Intent(getApplicationContext(), ResetPasswordActivity.class);
    startActivityForResult(intent, REQUEST_RESETPASS);
}
});

}

public void login() {
    Log.d(TAG, "Login");

    if (!validate()) {
        onLoginFailed();
        return;
    }

    _loginButton.setEnabled(false);

    final ProgressDialog progressDialog = new ProgressDialog(LoginActivity.this,
        R.style.AppTheme_Dark_Dialog);
    progressDialog.setIndeterminate(true);
    progressDialog.setMessage("Authenticating...");
    progressDialog.show();

    String email = _emailText.getText().toString();
    String password = _passwordText.getText().toString();

    loginUser.setLogin(email, password);
    Call<ResponseBody> call = apiService.loginUser(loginUser);

    System.out.println("Request is " + new Gson().toJson(call.request()));
    call.enqueue(new Callback<ResponseBody>() {
        @Override
```



```
public void onResponse(Call<ResponseBody> call, Response<ResponseBody> response) {
    int statusCode = response.code();
    System.out.println("Recieved " + statusCode);
    String responseString;
    try {
        responseString = response.body().string();
        Log.e("test6", responseString);
        //JSONObject responseObject = JSONObject.
        JsonParser parser = new JsonParser();
        JsonObject responseObject = parser.parse(responseString).getAsJsonObject();

        if (responseObject.get("status").getAsInt() == 200) {

            onLoginSuccess();
            //----eno edo xrisimopioume session, pou apothikevi mono key values----//

            session = new SessionHandler(LoginActivity.this);
            session.createLoginSession(loginUser.getEmail(), loginUser.getEmail());
            progressDialog.dismiss();
        } else {

            onLoginFailed();
            progressDialog.dismiss();
            Log.e("test2", responseObject.get("message").getString());
        }

    } catch (IOException e) {
        e.printStackTrace();
    }

}

@Override
public void onFailure(Call<ResponseBody> call, Throwable t) {
```



```
        // Log error here since request failed
        System.out.println("Failed call " + t);
        System.out.println("stack trace: ");
        onLoginFailed();
        progressDialog.dismiss();
        Log.e("test2", t.toString());
    }
});

}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == REQUEST_SIGNUP) {
        if (resultCode == RESULT_OK) {

            String email = (String) data.getExtras().getString("email");
            session = new SessionHandler(LoginActivity.this);
            session.createLoginSession(email, email);
            // TODO: Implement successful signup logic here
            // By default we just finish the Activity and log them in automatically
            this.finish();
        }
    } else if (requestCode == REQUEST_RESETPASS) {
        if (resultCode == RESULT_OK) {
            Toast.makeText(getBaseContext(), "Your Password now is the same as your username",
                Toast.LENGTH_LONG).show();
        }
    }
}

@Override
public void onBackPressed() {
```



```
// Disable going back to the MainActivity
moveTaskToBack(true);
}

public void onLoginSuccess() {
    _loginButton.setEnabled(true);
    finish();
}

public void onLoginFailed() {
    Toast.makeText(getBaseContext(), "Login failed", Toast.LENGTH_LONG).show();

    _loginButton.setEnabled(true);
}

public boolean validate() {
    boolean valid = true;

    String email = _emailText.getText().toString();
    String password = _passwordText.getText().toString();

    if (email.isEmpty() || !Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
        _emailText.setError("enter a valid email address");
        valid = false;
    } else {
        _emailText.setError(null);
    }

    if (password.isEmpty() || password.length() < 4 || password.length() > 10) {
        _passwordText.setError("between 4 and 10 alphanumeric characters");
        valid = false;
    } else {
        _passwordText.setError(null);
    }
}
```




```
        return valid;
    }
}
```

Στη συνέχεια, παρατίθεται το αντίστοιχο αρχείο xml που αφορά το “Layout”, την γραφική αποτύπωση της παραπάνω λειτουργίας στην οθόνη της εφαρμογής.

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

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="56dp"
        android:paddingLeft="24dp"
        android:paddingRight="24dp">

        <ImageView android:src="@drawable/logo"
            android:layout_width="wrap_content"
            android:layout_height="72dp"
            android:layout_marginBottom="24dp"
            android:layout_gravity="center_horizontal" />

        <!-- Email Label -->
        <android.support.design.widget.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="8dp"
            android:layout_marginBottom="8dp">
            <EditText android:id="@+id/input_email"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:inputType="textEmailAddress"/>
```



```
        android:hint="Email" />
</android.support.design.widget.TextInputLayout>

<!-- Password Label -->
<android.support.design.widget.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:layout_marginBottom="8dp">
    <EditText android:id="@+id/input_password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword"
        android:hint="Password"/>
</android.support.design.widget.TextInputLayout>

<android.support.v7.widget.AppCompatButton
    android:id="@+id/btn_login"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    android:layout_marginBottom="24dp"
    android:padding="12dp"
    android:text="Login"/>

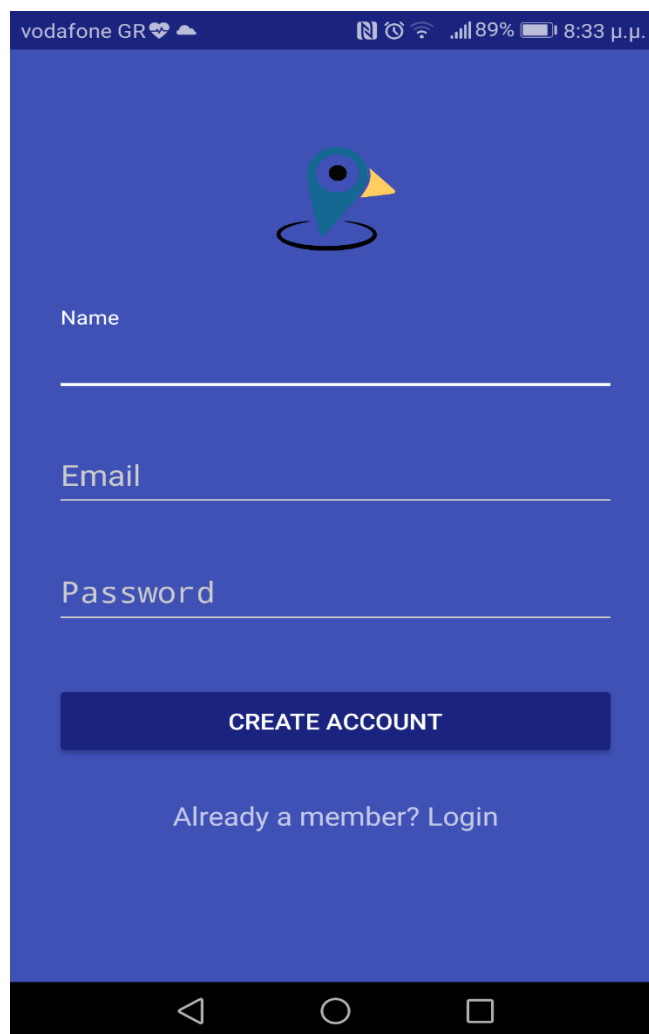
<TextView android:id="@+id/link_signup"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="24dp"
    android:text="No account yet? Create one"
    android:gravity="center"
    android:textSize="16dip"/>

<TextView android:id="@+id/link_reset"
    android:layout_width="fill_parent"
```



```
android:layout_height="wrap_content"  
android:layout_marginBottom="24dp"  
android:text="Forgot your pass? Reset it"  
android:gravity="center"  
android:textSize="16dp"/>
```

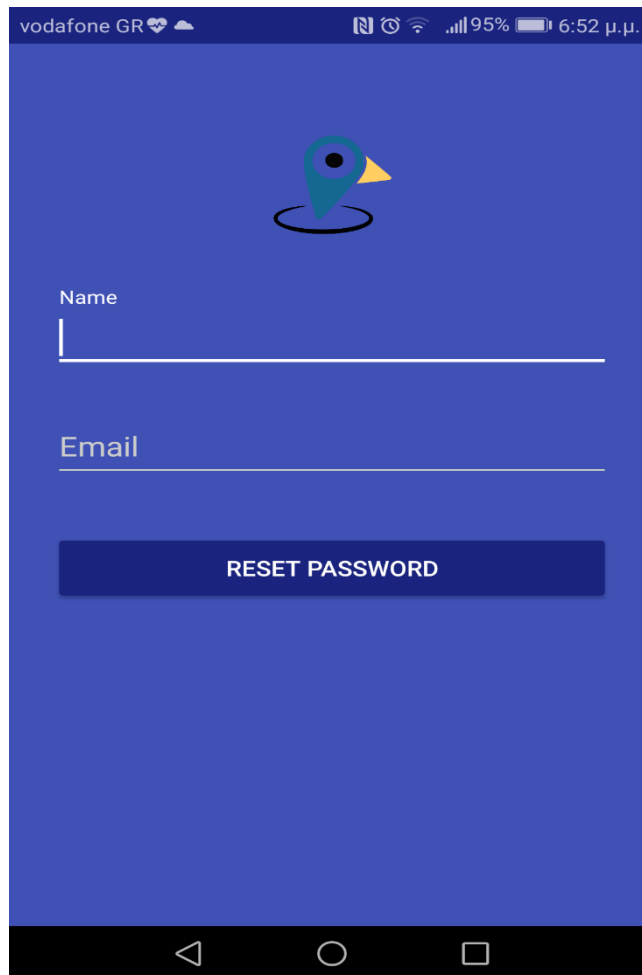
```
</LinearLayout>  
</ScrollView>
```



Εικόνα 7. Οθόνη δημιουργίας λογαριασμού χρήστη στη mobile εφαρμογή

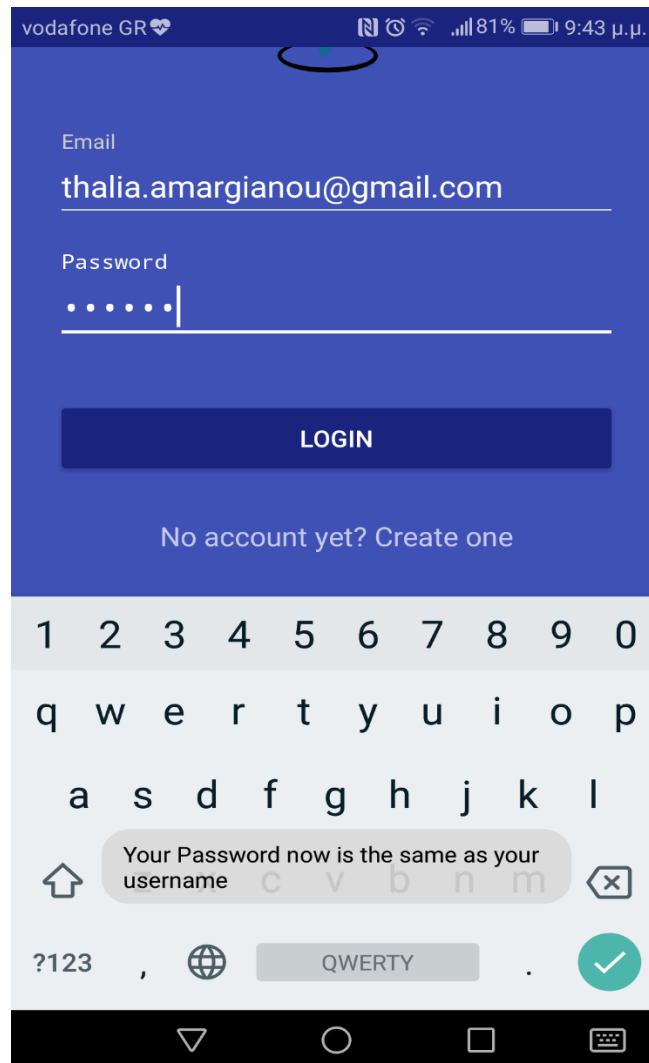


Για τη δημιουργία του λογαριασμού χρειάζεται το όνομα χρήστη, το email του και ένας κωδικός πρόσβασης. Με επιλογή στο κουμπί “create account” δημιουργείται ο λογαριασμός και ο χρήστης εισέρχεται στην κεντρική σελίδα της εφαρμογής. Σε περίπτωση που ο χρήστης της εφαρμογής είχε ήδη δημιουργήσει λογαριασμό στην εφαρμογή αλλά έχει ξεχάσει τον κωδικό πρόσβασης, δίνεται η δυνατότητα από την αρχική σελίδα της εφαρμογής να κάνει “reset” τον κωδικό και να δηλώσει ένα νέο κωδικό πρόσβασης.



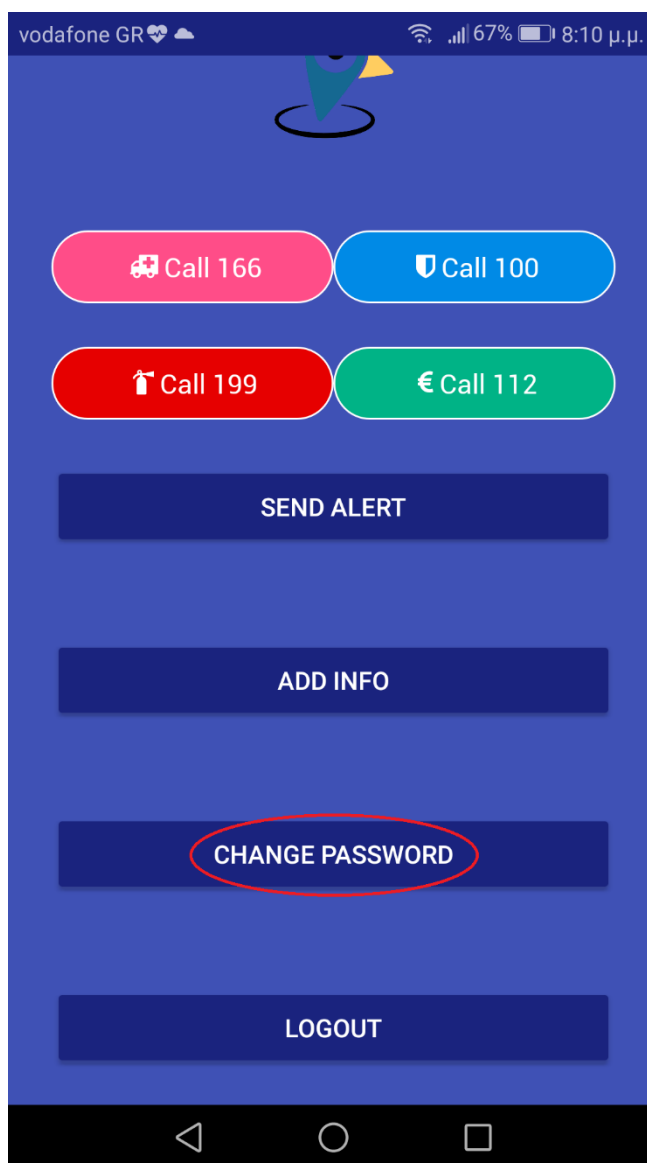
Εικόνα 8. Οθόνη επαναφοράς κωδικού πρόσβασης

Με χρήση του ονόματος χρήστη και του email του, κάνοντας επιλογή στο κουμπί “reset password”, επαναφέρεται στη σελίδα σύνδεσης στην εφαρμογή, όπου θα πρέπει να χρησιμοποιήσει το email του και ως κωδικό πρόσβασης το username του με σκοπό να εισέλθει στην κεντρική σελίδα της εφαρμογής και να κάνει αλλαγή κωδικού πρόσβασης από την επιλογή “change password”.



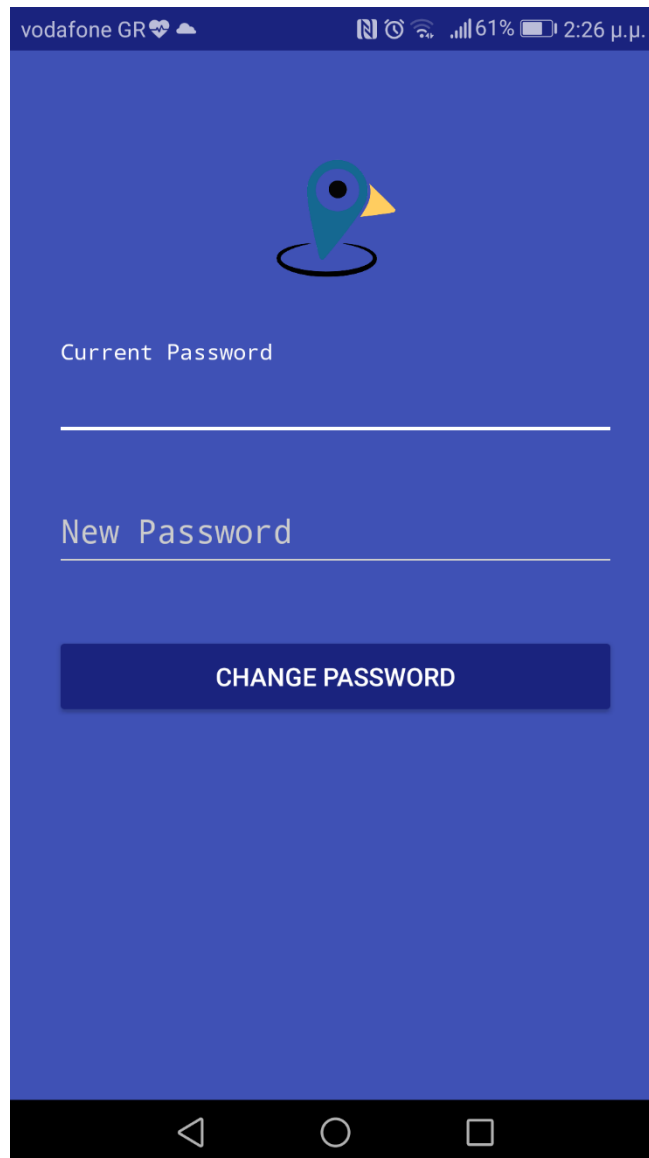
Εικόνα 9. Αρχική οθόνη μετά την επαναφορά κωδικού

Όπως προαναφέρθηκε, με για να ολοκληρωθεί η διαδικασία απόκτησης νέου κωδικού πρόσβασης, μετά την επαναφορά κωδικού, με χρήση του email και του ονόματος χρήστη ως κωδικό, ο χρήστης εισέρχεται στην κεντρική σελίδα της εφαρμογής, όπου εκεί μπορεί να ορίσει νέο κωδικό πρόσβασης.



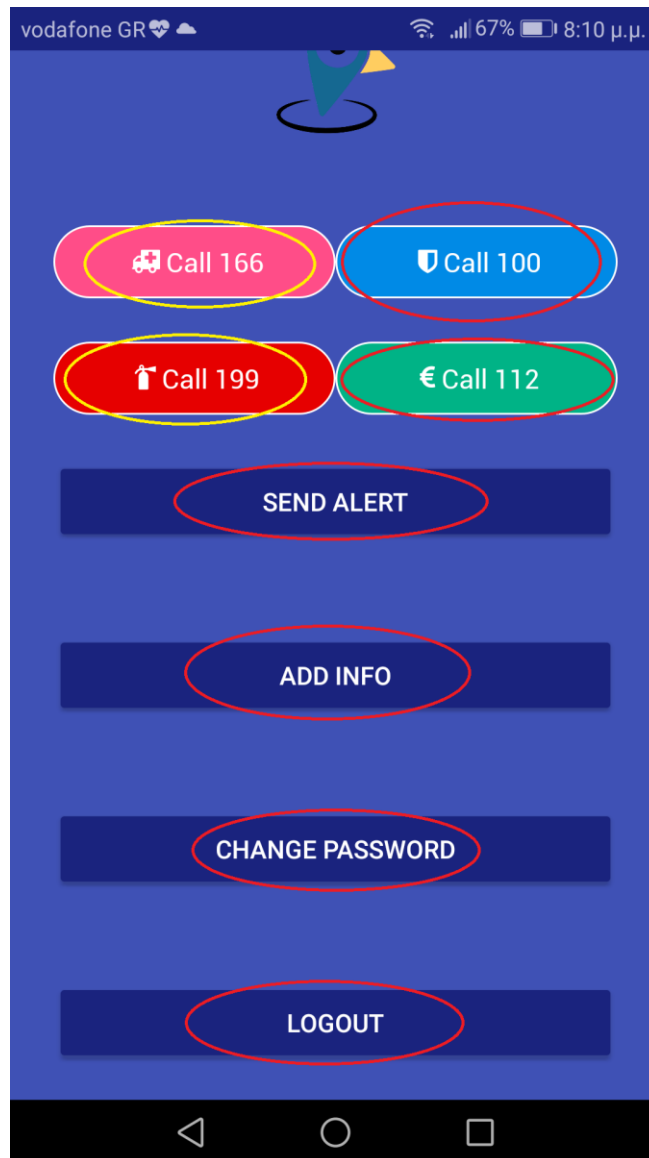
Εικόνα 10. Κεντρική οθόνη εφαρμογής - επιλογή κωδικού πρόσβασης

Αφού ο χρήστης εισέλθει στην κεντρική σελίδα της εφαρμογής, επιλέγει “change password” και μεταφέρεται στη σελίδα αλλαγής κωδικού πρόσβασης.



Εικόνα 11. Οθόνη αλλαγής κωδικού πρόσβασης

Με εισαγωγή στα πεδία, του τρέχοντος κωδικού και του νέου κωδικού πρόσβασης, επιλέγοντας “change password” επιτυγχάνεται η αλλαγή – επαναφορά του κωδικού πρόσβασης. Πλέον ο χρήστης βρίσκεται στην κεντρική σελίδα της εφαρμογής, όπου από κει θα μπορεί να προσθέσει χρήσιμες προσωπικές πληροφορίες, να αλλάξει τον κωδικό πρόσβασης όποια στιγμή το κρίνει απαραίτητο, να αποσυνδεθεί από την εφαρμογή και το πιο σημαντικό να αποστείλει ειδοποίηση έκτακτης ανάγκης.



Εικόνα 12. Κεντρική οθόνη εφαρμογής - Διαθέσιμες επιλογές

Έχει ήδη γίνει περιγραφή της επιλογής αλλαγής κωδικού πρόσβασης και η επιλογή “Logout”, μας δίνει τη δυνατότητα αποσύνδεσης από την εφαρμογή. Η επιλογή “add info” εισάγει το χρήστη σε μια νέα σελίδα της εφαρμογής, όπου εκεί συμπληρώνοντας τα πεδία, αποθηκεύει χρήσιμες προσωπικές πληροφορίες οι οποίες αντλούνται κατά την αποστολή ειδοποίησης έκτακτης ανάγκης, όπως θα διαπιστώσετε στη συνέχεια.

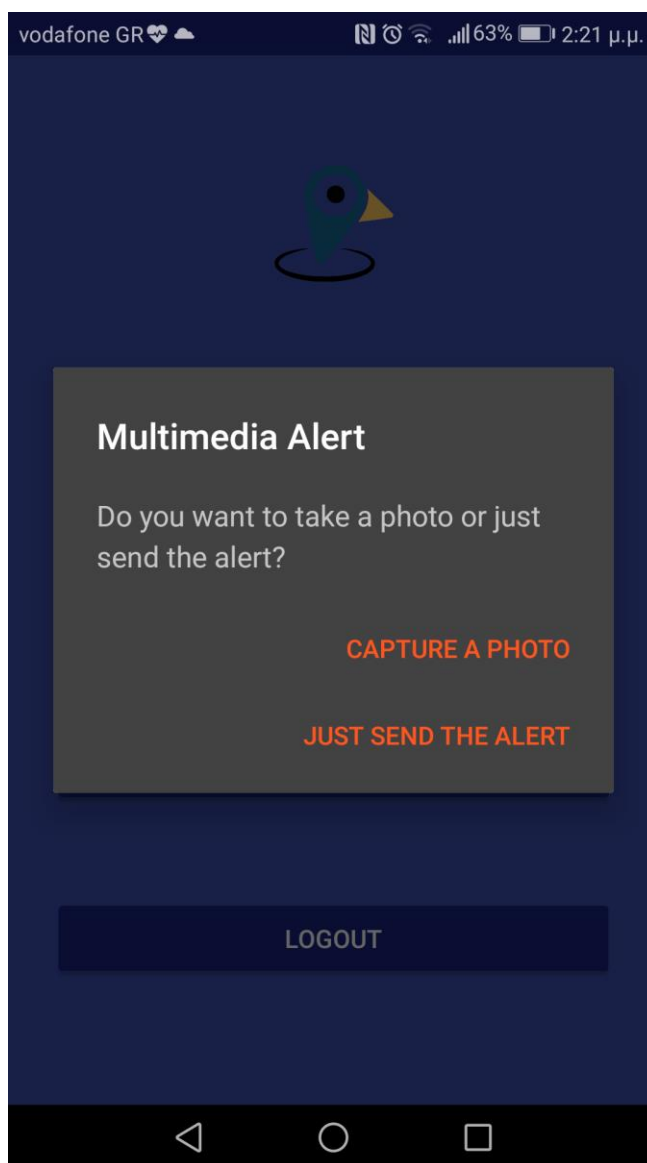


The screenshot shows a registration form with the following elements:

- Header: "vodafone GR" with signal, Wi-Fi, and battery icons, and time "2:23 μ.μ.".
- Form fields: "Name", "Surname", "History", "Cellphone", "Closest Person Cellphone", and "Home".
- Dropdown menus: "English" and "0+".
- Bottom navigation bar: Standard Android navigation icons.

Εικόνα 13. Οθόνη εισαγωγής χρήσιμων προσωπικών πληροφοριών χρήστη

Όπως βλέπετε και στην εικόνα 12, ο χρήστης στη σελίδα αυτή έχει τη δυνατότητα να εισάγει στα πεδία πληροφορίες όπως, το όνομα, επώνυμο, κάποια σημαντική πληροφορία από το ιστορικό του που μπορεί να αφορά θέμα υγείας ή κάποια αλλεργία, τους αριθμούς του δικού του κινητού τηλεφώνου και ενός κοντινού του προσώπου, τη διεύθυνση κατοικίας του και την ομάδα αίματός του. Στη συνέχεια επιλέγοντας "submit personal info" αποθηκεύει τις πληροφορίες στη Βάση Δεδομένων, ώστε να αντληθούν και να αποσταλούν σε περιπτώσεις έκτακτης ανάγκης από το χρήστη. Ένα πολύ σημαντικό στοιχείο της εφαρμογής είναι η επιλογή "send alert", όπου μέσω αυτής στέλνονται και ειδοποιήσεις έκτακτης ανάγκης.



Εικόνα 14. Οθόνη αποστολής ειδοποιήσεων έκτακτης ανάγκης

Ένα μικρό δείγμα του κώδικα υλοποίησης της συγκεκριμένης λειτουργίας είναι το παρακάτω:

```
public void showDialog() {
```

```
    new MaterialDialog.Builder(this)
        .title("Multimedia Alert")
        .content("Do you want to take a photo or just send the alert?")
        .positiveText("Capture a photo")
```



```
.negativeText("Just send the alert")
.onPositive(new MaterialDialog.SingleButtonCallback() {
    @Override
    public void onClick(@NonNull MaterialDialog dialog, @NonNull DialogAction which) {
        // TODO
        takePhoto();
    }
})
.onNegative(new MaterialDialog.SingleButtonCallback() {
    @Override
    public void onClick(@NonNull MaterialDialog dialog, @NonNull DialogAction which) {
        // TODO
        showAlert(null);
    }
})
.show();

}

public void openCamera() {
    EasyImage.openCamera(this, 0);
}

public void deleteLastPhoto() {
    File photoFile = EasyImage.lastlyTakenButCanceledPhoto(this);
    if (photoFile != null) photoFile.delete();
}

public void takePhoto() {

    EasyImage.clearConfiguration(this);
```



```

EasyImage.configuration(this.getApplicationContext())
    .setCopyExistingPicturesToPublicLocation(true)
    .setImagesFolderName("Alert")
    .saveInRootPicturesDirectory();

/**Permission check only required if saving pictures to root of sdcard*/
int permissionCheck = ContextCompat.checkSelfPermission(this, filePermissions[0]);
int permissionCheck2 = ContextCompat.checkSelfPermission(this, filePermissions[1]);
if (permissionCheck == PackageManager.PERMISSION_GRANTED && permissionCheck2 ==
PackageManager.PERMISSION_GRANTED) {
    openCamera();
} else {

    Nammu.askForPermission(this, filePermissions, new PermissionCallback() {
        @Override
        public void permissionGranted() {
            openCamera();
        }

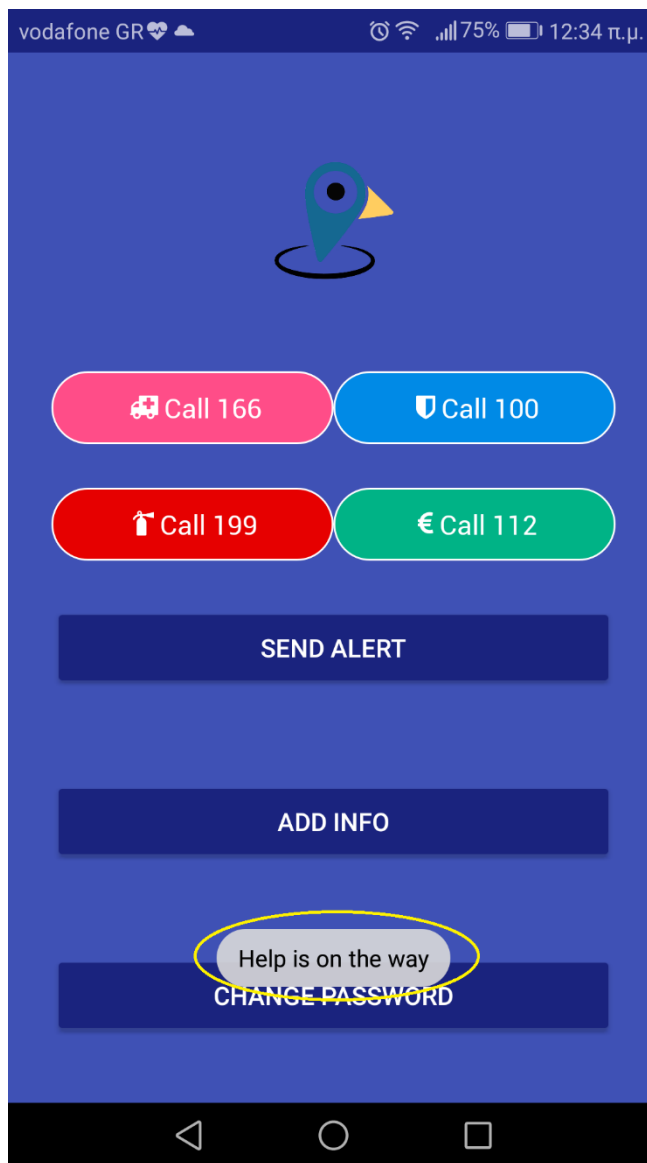
        @Override
        public void permissionRefused() {
            Log.d("Permission is :", "Denied");
        }
    });
}
}

```

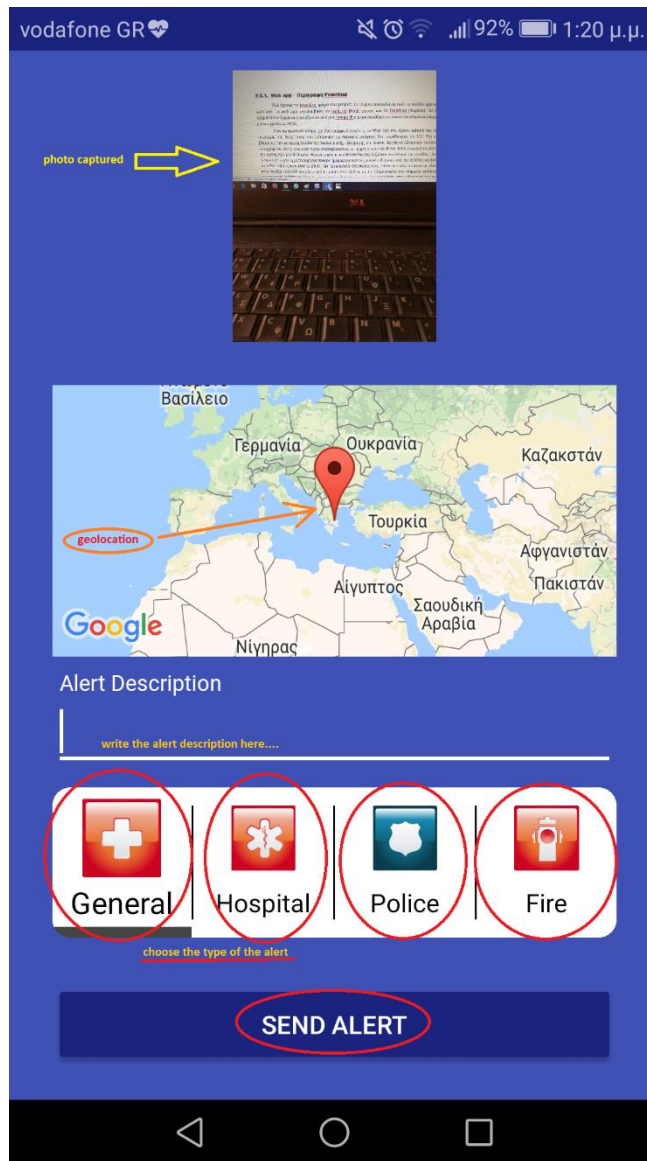
Στη σελίδα αυτή ο χρήστης έχει τη δυνατότητα να στείλει ειδοποίηση έκτακτης ανάγκης με δύο τρόπους. Πρώτον, με την λήψη φωτογραφίας που ενδεχομένως να δίνει περισσότερες πληροφορίες για την κατάσταση του χρήστη και παράλληλη αποστολή της φωτογραφίας, των προσωπικών πληροφοριών και το γεωγραφικό στίγμα του χρήστη, επιλέγοντας το πεδίο “capture a photo”. Μετά τη λήψη της φωτογραφίας και την επιλογή της, στο επόμενο βήμα μεταφερόμαστε στη σελίδα επιλογής του τύπου ανάγκης και εκεί υπάρχει η δυνατότητα να δούμε τη φωτογραφία που τραβήξαμε, το χάρτη με το σημείο στο οποίο βρισκόμαστε και μας δίνεται και η δυνατότητα να επιλέξουμε αν πιστεύουμε ότι χρειαζόμαστε ιατρική βοήθεια, αν χρειαζόμαστε τη βοήθεια της αστυνομίας ή της πυροσβεστικής. Επίσης, κατά την επιλογή του τύπου κινδύνου ανάλογα με το περιστατικό, μπορούμε να γράψουμε και να αποστείλουμε κάποιο σχόλιο, κείμενο που θα διευκολύνει το χειριστή που θα λάβει το μήνυμα να αποστείλει την κατάλληλη βοήθεια.



Δεύτερον, με αποστολή των προσωπικών πληροφοριών του χρήστη και του γεωγραφικού του στίγματος, επιλέγοντας το πεδίο “send alert”, ακολουθείτε η προηγούμενη διαδικασία εκτός από την λήψη και αποστολή φωτογραφίας. Με την επιλογή αποστολής σήματος κινδύνου, εμφανίζεται η ειδοποίηση στην οθόνη της κεντρικής σελίδας “Help is on the way”, όπου ενημερώνει το χρήστη ότι το μήνυμα κινδύνου που απέστειλε έχει παραλειφθεί από την αρμόδια υπηρεσία.



Εικόνα 15. Κεντρική οθόνη εφαρμογής - μήνυμα ενημέρωσης παραλαβής "alert"



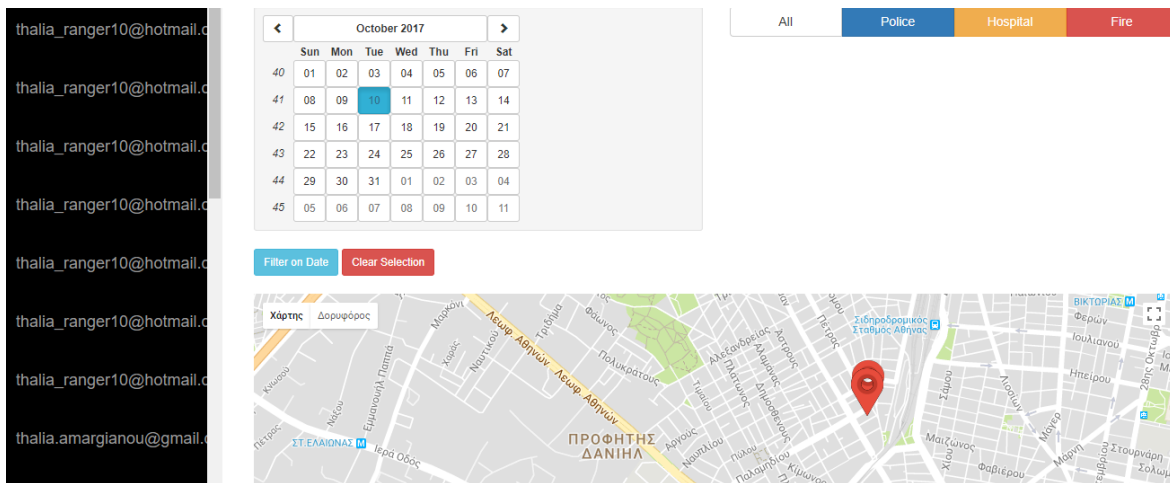
Εικόνα 16. Οθόνη αποστολής alert μετά τη λήψη φωτογραφίας



2.5.1 Web app – Περιγραφή FrontEnd

Εδώ έχουμε το FrontEnd τμήμα του project. Το project αποτελείται από το mobile app και από το web app. Το web app περιλαμβάνει το Backend (Node server) και το FrontEnd (Angular). Τα δύο αυτά τμήματα του έργου υποστηρίζονται από μια mongoDB για την αποθήκευση των απαιτούμενων πληροφοριών μέσω εγγράφων JSON.

Στον πραγματικό κόσμο, με ένα υπαρκτό σενάριο, το Web app του έργου αφορά την υπηρεσία υποδοχής και διαχείρισης των ειδοποιήσεων έκτακτης ανάγκης. Για παράδειγμα, το 112. Στη συνέχεια βλέπουμε την κεντρική σελίδα της διαδικτυακής εφαρμογής του έργου. Αριστερά βλέπουμε τη λίστα με τα εισερχόμενα alerts, όπου στη λίστα αναπαρίστανται με τη χρήση του email του. Κάθε ένα από τα alert signals της λίστας έχει μια δεδομένη θέση στο χάρτη που βρίσκεται στα δεξιά και στο κέντρο της σελίδας. Δεξιά και πάνω από το χάρτη βλέπουμε ένα πίνακα ημερολογίου όπου μπορεί ο διαχειριστής της σελίδας να φιλτράρει τις ειδοποιήσεις κινδύνου με βάση την ημερομηνία αποστολής τους. Πάνω στο χάρτη κάνοντας κλικ πάνω στην πινέζα εμφανίζεται κάτω από το χάρτη ένα πλαίσιο με τις πληροφορίες του σήματος κινδύνου. Τα στοιχεία του χρήστη και όλες τις προσωπικές πληροφορίες που είχε προσθέσει στην εφαρμογή του κινητού του ο χρήστης. Επίσης, αποστέλλεται και το γεωγραφικό σημείο, με τις γεωγραφικές συντεταγμένες, από το σημείο από το οποίο στάλθηκε η ειδοποίηση. Τις ίδιες πληροφορίες μπορούμε να εμφανίσουμε στο πλαίσιο αυτό αν επιλέξουμε κάποιο από τα email που βρίσκονται στη λίστα με τις ειδοποιήσεις κινδύνου.



Εικόνα 17. Οθόνη διαχείρισης σημάτων κινδύνου - web εφαρμογής

Στα δεξιά της εικόνας 17 βλέπουμε διαφορετικούς τύπους σημάτων κινδύνων που μπορεί να φτάσουν στο σύστημα διαχείρισης, και έτσι ο διαχειριστής μπορεί να επιλέξει να του εμφανίσει τα alerts ενός από αυτά. Οι τύποι κινδύνων αφορούν, περιστατικά υγείας, την άμεσση δράση και την Πυροσβεστική. Υπάρχει η δυνατότητα να εμφανίσει ο διαχειριστής και όλους τους τύπους μαζί, όπου εκεί περιλαμβάνονται και τα μηνύματα γενικού τύπου – General.

Στη συνέχεια βλέπουμε στην εικόνα 18, το πλαίσιο με τις πληροφορίες ενός επιλεγμένου σήματος ειδοποίησης έκτακτης ανάγκης. Ακολουθεί τμήμα του κώδικα υλοποίησης.



```
vm.info.push( [ "Email" , vm.alerts[vm.selected].user.email ] );
vm.info.push( [ "First Name" , vm.alerts[vm.selected].user.firstname ] );
vm.info.push( [ "Last Name" , vm.alerts[vm.selected].user.lastname ] );
vm.info.push( [ "Username" , vm.alerts[vm.selected].user.username ] );
vm.info.push( [ "Bio" , vm.alerts[vm.selected].user.history ] );
vm.info.push( [ "Blood Type" , vm.alerts[vm.selected].user.blood ] );
vm.info.push( [ "CellPhone" , vm.alerts[vm.selected].user.cellphone ] );
vm.info.push( [ "Next Kin's Cellphone" , vm.alerts[vm.selected].user.scellphone ] );
vm.info.push( [ "Language" , vm.alerts[vm.selected].user.language ] );
vm.info.push( [ "Home" , vm.alerts[vm.selected].user.home ] );
vm.info.push( [ "Latitude" , vm.alerts[vm.selected].latitude ] );
vm.info.push( [ "Longitude" , vm.alerts[vm.selected].longitude ] );

}

if ( vm.alerts[vm.selected].imgname ){
    vm.alertimage=vm.alerts[vm.selected].imgname;
}

vm.map.panTo({lat: vm.alerts[vm.selected].latitude , lng: vm.alerts[vm.selected].longitude });

};

var generateMarkers = function() {

    $http({method: 'GET', url: '/api/alerts'}).
    success(function(data, status, headers, config) {
        // this callback will be called asynchronously
        // when the response is available.
        console.log(data);
    });
}
```



```
console.log(vm.selected);
// vm.positions = [];
// vm.alerts = data;
for (i = 0; i < data.length; i++) {
  if (vm.alertsIDs.indexOf(data[i]._id) > -1) {

  }else {

    vm.alertsIDs.push(data[i]._id);

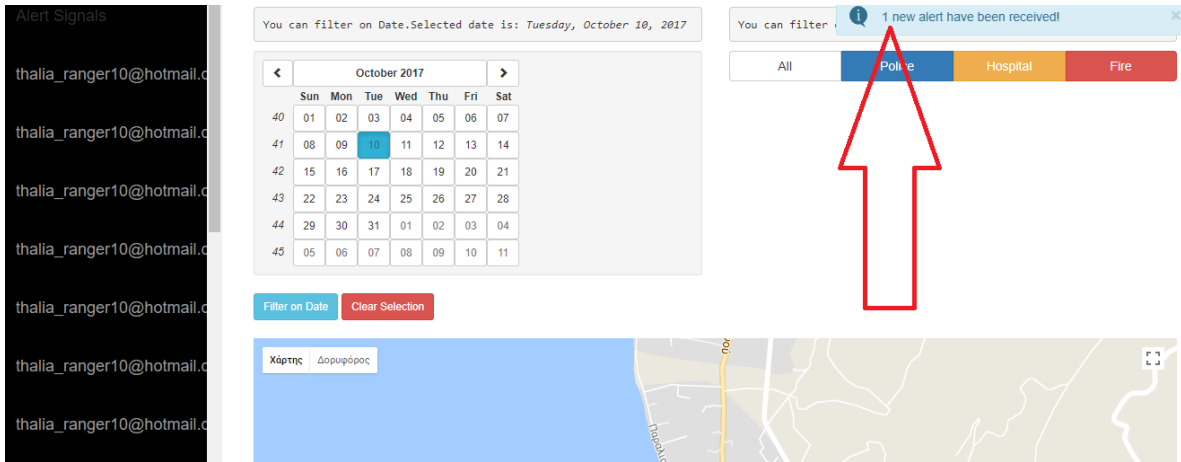
    vm.alerts.push(data[i]);

    var lat = data[i].latitude;
    var lng = data[i].longitude;
    vm.positions.push(data[i]);
  }
}
//users = data;
}).
error(function(data, status, headers, config) {
  // called asynchronously if an error occurs
  // or server returns response with an error status.
  console.log('Oops and error', data);

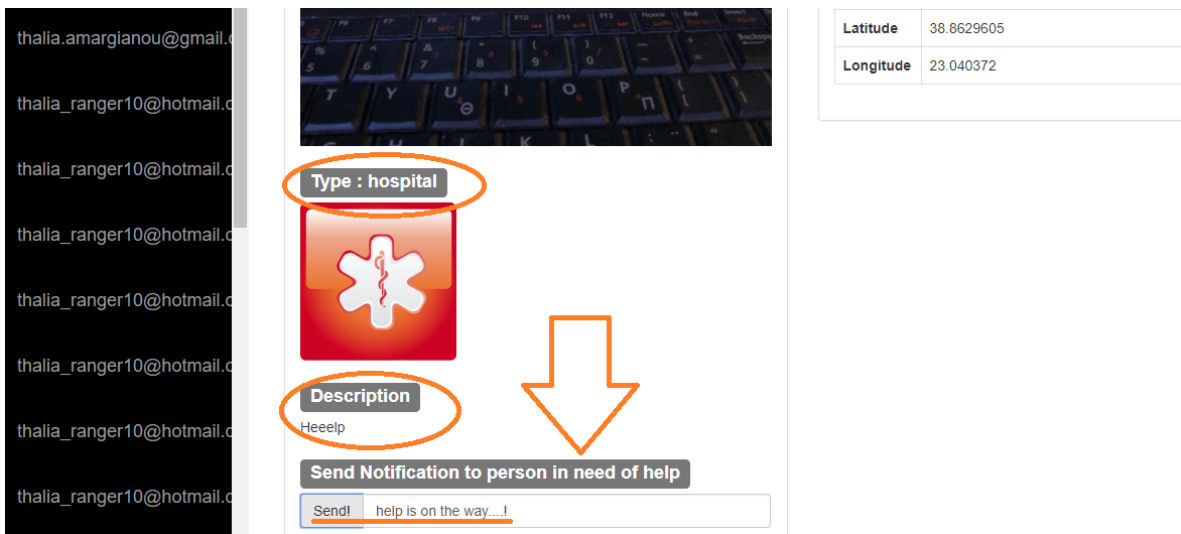
});

console.log("vm.positions", vm.positions);
};

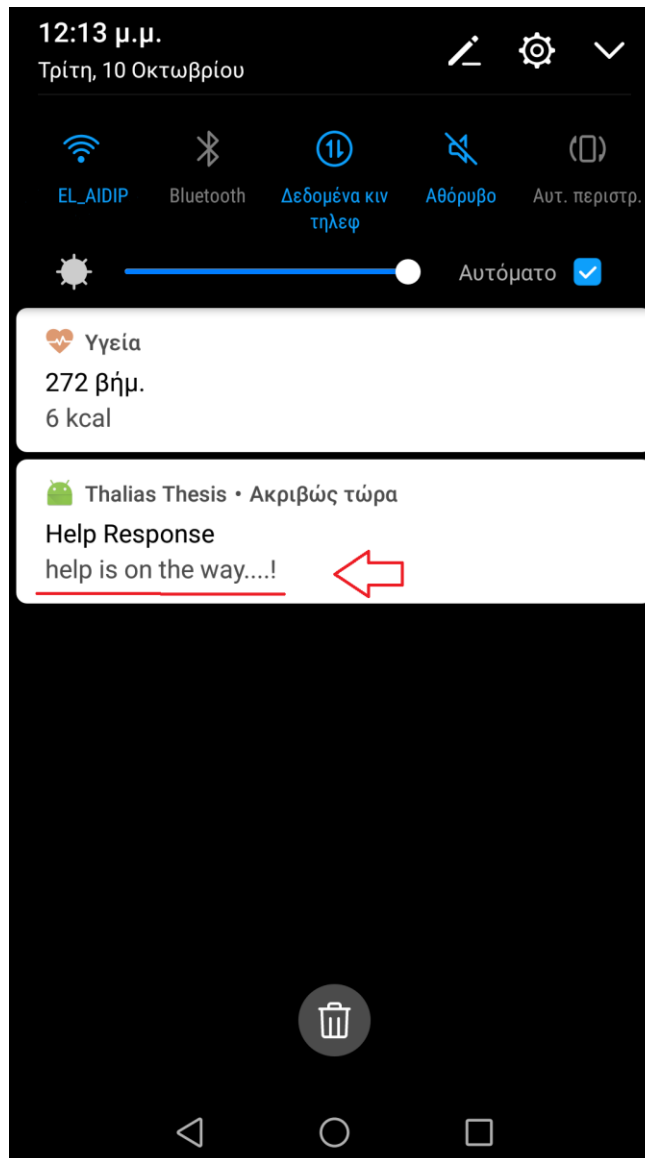
vm.highlightedIcon = function() {
  return {
    url: 'micon.png'
  };
};
};
```



Εικόνα 19. Web app - Ειδοποίηση λήψης μηνύματος κινδύνου



Εικόνα 20. Web app - Αποστολή ειδοποίησης προς το χρήστη της Mobile εφαρμογής



Εικόνα 21. Mobile εφαρμογή - λήψη ειδοποίησης από το Web app

2.6 Αιτιολόγηση και συναφείς εφαρμογές

Το 112 είναι ο Ευρωπαϊκός αριθμός έκτακτης ανάγκης και στα 28 κράτη μέλη της ΕΕ, καθώς και σε άλλες χώρες της Ευρώπης και αλλού. Τα άτομα που βρίσκονται σε κίνδυνο μπορούν να καλέσουν το 112, οποιαδήποτε στιγμή μέσα στις 24 ώρες της ημέρας, για να ειδοποιήσουν την πυροσβεστική, την αστυνομία ή να καλέσουν σε ιατρική βοήθεια. Ο Ευρωπαϊκός αριθμός κλήσης έκτακτης ανάγκης είναι δωρεάν και μπορεί να κληθεί από σταθερά καθώς και από κινητά τηλέφωνα. Το αυξημένο επίπεδο των Ευρωπαίων που



ταξιδεύουν από τη μια χώρα στην άλλη, οδήγησε το Συμβούλιο της Ευρωπαϊκής Ένωσης να εισαγάγει έναν κοινό αριθμό έκτακτης ανάγκης σε όλα τα κράτη προκειμένου να αποφευχθεί η ανάγκη να θυμόμαστε διαφορετικούς εθνικούς αριθμούς έκτακτης ανάγκης ανάλογα με τη τοποθεσία του ατόμου.

Στην ιστοσελίδα της η EENA, δίνει μερικές συμβουλές προς τους πολίτες που θα καλέσουν το 112. Συμβουλές όπως:

- Παρέμεινε ψύχραιμος. Θα πρέπει να παρέχεις σημαντικές πληροφορίες στην υπηρεσία έκτακτης ανάγκης, οπότε είναι επιβεβλημένο να διατηρήσεις την ψυχραιμία σου.
- Εάν είναι δυνατό κάνε την κλήση μόνος σου. Δεν μπορεί να περιγράψει κανείς καλύτερα από εσένα το συμβάν.
- Περίμενε μέχρι ο διαχειριστής να απαντήσει στην κλήση σου. Μπορεί να χρειαστεί κάποιος χρόνος μέχρι να απαντηθεί η κλήση, αλλά θα πρέπει να περιμένεις διότι κάθε επαναλαμβανόμενη κλήση θεωρείται ως νέα και μπαίνει στο τέλος της ουράς αναμονής, οπότε μπορεί να παραταθεί ο χρόνος διέλευσης.
- Δήλωσε το όνομα σου, τι συνέβη, ποιος εμπλέκεται και υπέδειξε την τοποθεσία σου.
- Ακολούθησε τις συμβουλές του διαχειριστή. Απάντησε στις ερωτήσεις του και ακολούθησε τις συμβουλές του. Μην κλήσεις το τηλέφωνο εάν δεν στο ζητήσει ο διαχειριστής. Μην απασχολείς τη γραμμή του τηλεφώνου σου μέχρι να καταφθάσουν οι αρμόδιες υπηρεσίες. Μπορεί να χρειαστεί να επικοινωνήσουν ξανά μαζί σου.
- Αν αλλάξει κάτι ενημέρωσε ξανά την υπηρεσία 112.

Όλες αυτές οι πληροφορίες μπορεί να φαίνονται ως λεπτομέρειες ή υπερβολική πληροφορία σε σχέση με την εφαρμογή που πραγματευόμαστε, αλλά επί της ουσίας είναι ο συνδεδετικός κρίκος ανάμεσα σε ένα σύστημα PSAP (Public Safety Answering Point) όπως το 112 και σε μια mobile εφαρμογή κλήσεων έκτακτης ανάγκης όπως η Emergency call android app. Οι παραπάνω συμβουλές μπορούν εύκολα να μετατραπούν σε καιρία ερωτήματα προς τους διαχειριστές του συστήματος και κατ' επέκταση στην EENA. Για παράδειγμα, αν το συμβάν είναι τέτοιο που ο χρήστης δεν μπορεί να διατηρήσει την ψυχραιμία του ώστε να έχει μια πλήρη συνομιλία με το διαχειριστή του 112 και δεν μπορεί να του δώσει τις πληροφορίες που χρειάζεται. Με ποιο τρόπο θα μπορούσε να στείλει μια ειδοποίηση προς το 112, ότι βρίσκεται σε κίνδυνο, χωρίς να χρειαστεί να μιλήσει στον διαχειριστή του συστήματος που λαμβάνει τις κλήσεις προς το 112; Αν δεν είναι δυνατό να δηλώσει τηλεφωνικά το όνομα του, να περιγράψει το περιστατικό, να υποδείξει την τοποθεσία στην οποία βρίσκεται και να δώσει περισσότερα στοιχεία για το συμβάν τηλεφωνικά. Πως μπορεί να αποστείλει αυτά τα στοιχεία στο 112; Αντιστοίχως, μπορεί κανείς να αναρωτηθεί και για τις υπόλοιπες συμβουλές που δίνονται από την EENA (European Emergency Number Association - 112), για το πως θα μπορούσε κάποιος που δεν είναι σε θέση να επικοινωνήσει τηλεφωνικά είτε λόγω της σοβαρότητας του συμβάντος, είτε λόγω έλλειψης ικανότητας λόγω κάποιας μορφής αναπηρίας, να επικοινωνήσει με το 112 και να δώσει τις απαραίτητες πληροφορίες.

Η τεχνολογία αιχμής που χρησιμοποιείται πλέον στα έξυπνα κινητά τηλέφωνα, έρχεται να δώσει τις λύσεις σε όλα εκείνα τα ερωτήματα και κατά επέκταση τα προβλήματα που αποτρέπουν έναν άνθρωπο που βρίσκεται σε κίνδυνο να στείλει ή να δώσει τις απαραίτητες σωστές και ακριβείς πληροφορίες προς τις υπηρεσίες διαχείρισης κλήσεων εκτάκτων αναγκών.



Η πλατφόρμα Android υποστηρίζει τρεις μεγάλες κατηγορίες αισθητήρων:

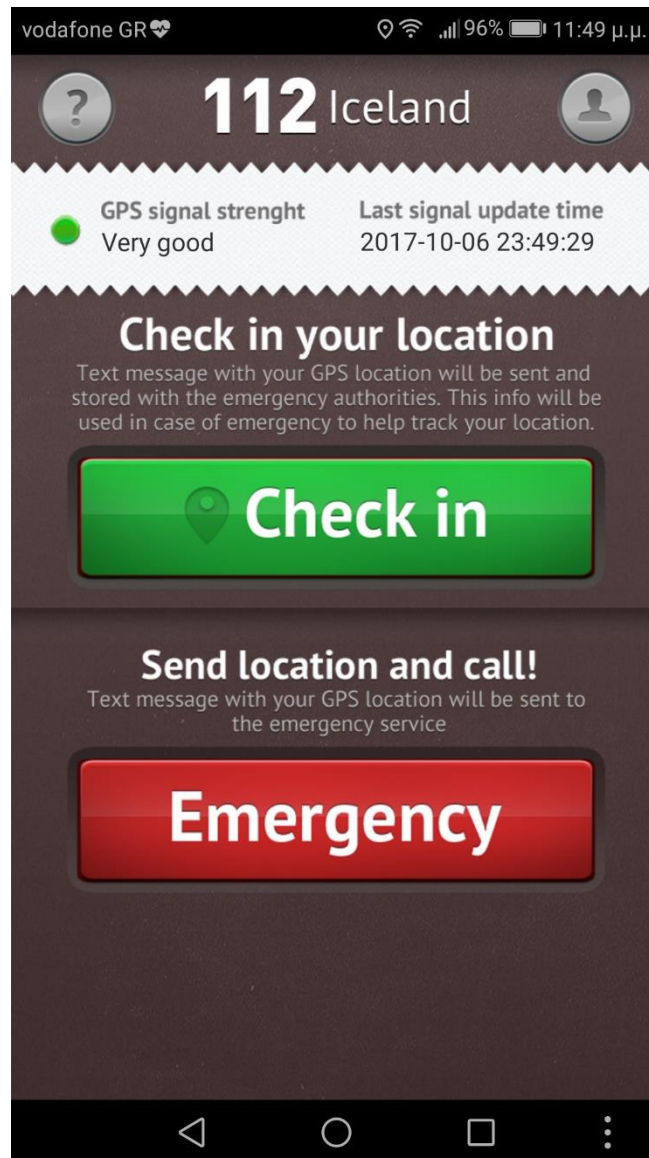
- Αισθητήρες κίνησης. Αυτοί οι αισθητήρες μετρούν τις δυνάμεις επιτάχυνσης και τις δυνάμεις περιστροφής κατά μήκος τριών αξόνων. Αυτή η κατηγορία περιλαμβάνει επιταχυνσιόμετρα, αισθητήρες βαρύτητας, γυροσκόπια και αισθητήρες περιστροφικού φορέα.
- Περιβαλλοντικοί αισθητήρες. Αυτοί οι αισθητήρες μετρούν διάφορες περιβαλλοντικές παραμέτρους, όπως θερμοκρασία και πίεση του αέρα του περιβάλλοντος, φωτισμό και υγρασία. Αυτή η κατηγορία περιλαμβάνει βαρόμετρα, φωτομέτρων και θερμόμετρα.
- Αισθητήρες θέσης. Αυτοί οι αισθητήρες μετρούν τη φυσική θέση μιας συσκευής. Αυτή η κατηγορία περιλαμβάνει αισθητήρες προσανατολισμού και μαγνητόμετρα.

Επίσης, περιλαμβάνει κάποια πολύ σημαντικά χαρακτηριστικά από τα οποία ενδεικτικά να αναφέρουμε, την αποστολή μηνυμάτων SMS και MMS, είναι διαθέσιμες μορφές μηνυμάτων, συμπεριλαμβανομένων των μηνυμάτων κειμένου με χρονολογική σειρά στην ίδια οθόνη και των μηνυμάτων Android Cloud To Device Messaging (C2DM) και τώρα βελτιωμένης έκδοσης του C2DM, ενώ το Google Cloud Messaging Android (GCM) είναι επίσης ένα μέρος των υπηρεσιών Android Push Messaging. Υπηρεσίες βασισμένες στο χαρακτηριστικό της φωνής, όπως φωνητικές ενέργειες για κλήσεις, γραπτά μηνύματα και πλοήγηση. Αποστολή αρχείων, βιντεοκλήσεις, ζωντανή λήψη και αποστολή πολυμεσικού στοιχείου κ.α.

Λαμβάνοντας λοιπόν υπόψη, τις δυνατότητες που παρέχονται για χρήση από την πλατφόρμα Android και την ανάγκη για άμεση και με ακριβή στοιχεία ειδοποίηση του 112 σε περίπτωση έκτακτης ανάγκης, προγραμματιστές android εφαρμογών έχουν προσπαθήσει να προσεγγίσουν όσο πιο ολοκληρωμένα τους ήταν δυνατό τον στόχο αυτό.

Στη συνέχεια γίνεται μια παρουσίαση – περιγραφή κάποιων εφαρμογών που αφορούν κλήσεις προς το 112 και μια συγκριτική μελέτη όσο αφορά τα χαρακτηριστικά τους σε σχέση με την εφαρμογή Emergency call android app που υλοποιείται στα πλαίσια αυτής της εργασίας.

Η εφαρμογή 112 Iceland, είναι η επίσημη εφαρμογή στην Ισλανδία που χρησιμοποιείται σε έξυπνα κινητά τηλέφωνα android για κλήσεις προς το 112. Η εφαρμογή αυτή στέλνει ένα SMS με το γεωγραφικό σημείο στο οποίο βρίσκεται το κινητό τηλέφωνο, πριν τη κλήση προς το 112, εντοπίζοντας το σημείο μέσω GPS.



Εικόνα 22. Κεντρική οθόνη 112 Iceland εφαρμογής

Όπως βλέπουμε, στην κεντρική οθόνη της εφαρμογής 112 Iceland, δίνονται πληροφορίες, όπως το επίπεδο ποιότητας του σήματος του GPS, την ημερομηνία και ώρα της τελευταίας ενημέρωσης του GPS σήματος, δυνατότητα επιλογής check in your location κουμπιού και του κουμπιού emergency. Επιλέγοντας Check in, αποστέλλεται μήνυμα προς το 112 με το γεωγραφικό σημείο που βρίσκεται ο χρήστης της συσκευής και αποθηκεύεται από την υπηρεσία έκτακτης ανάγκης ώστε να χρησιμοποιηθεί για τον εντοπισμό του χρήστη όποια στιγμή χρειαστεί. Με την επιλογή Emergency, αποστέλλεται μήνυμα με τα γεωγραφικό σημείο που βρίσκεται ο χρήστης της εφαρμογής και γίνεται και κλήση τηλεφωνική προς το 112.



vodafone GR 96% 11:49 μ.μ.

112 Iceland

Please enter your information:

Your full name

Contact in case of emergency

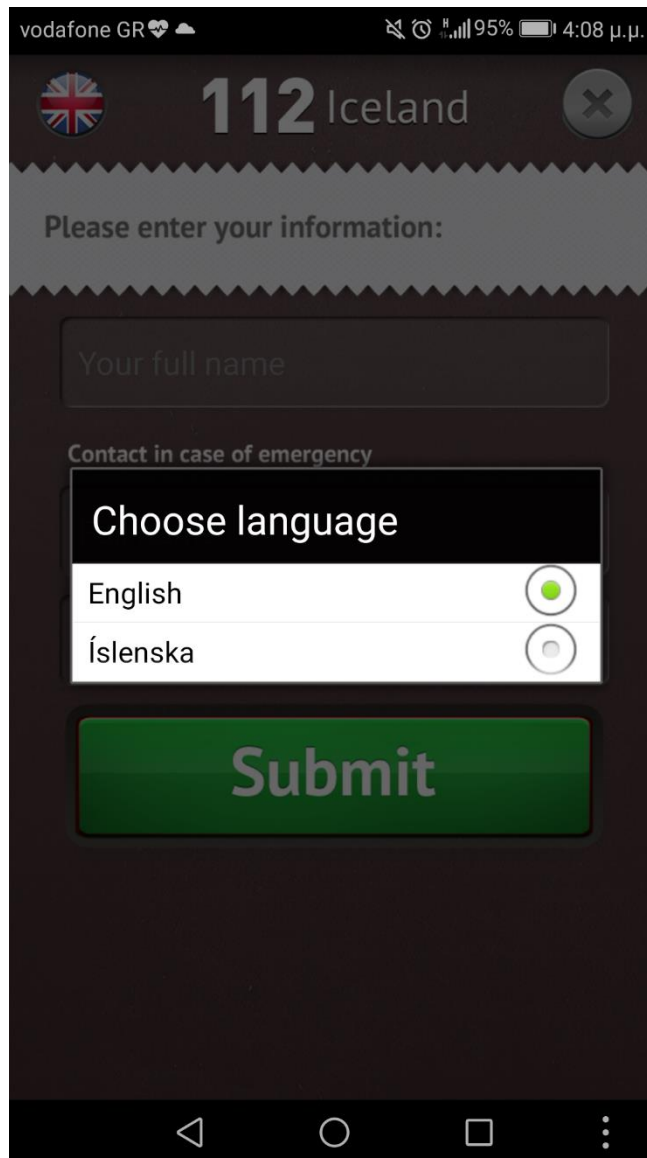
Full name

Phone number

Submit

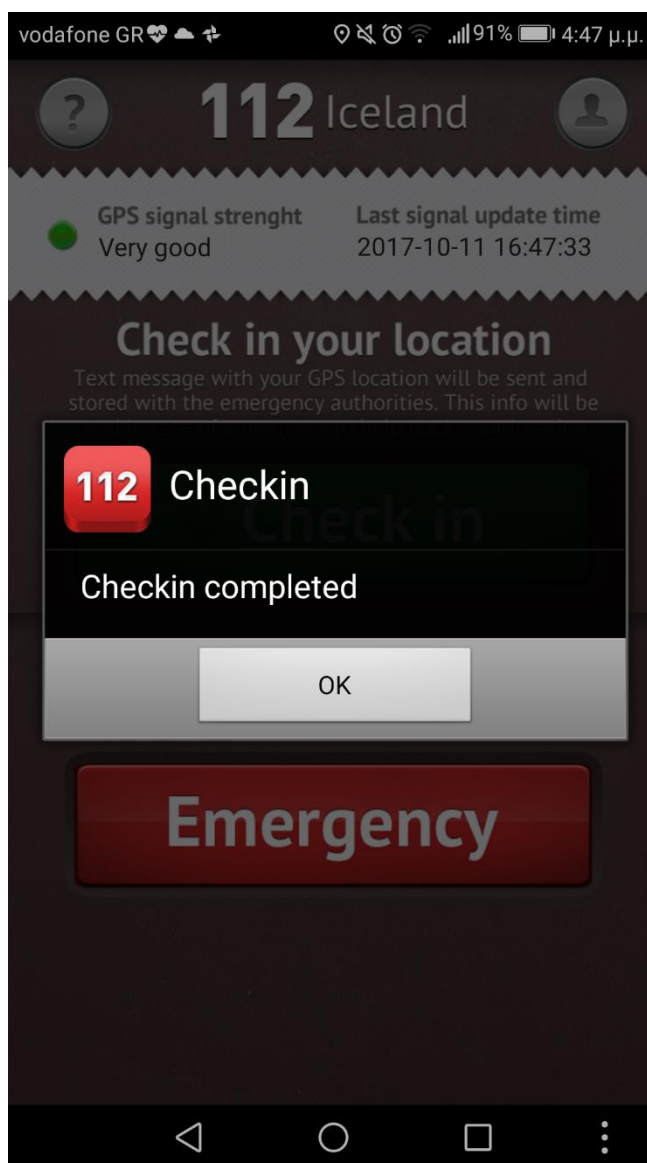
Εικόνα 23. Οθόνη προσθήκης πληροφοριών του χρήστη στην 112 Iceland app

Στην οθόνη αυτή βλέπουμε ότι ο χρήστης έχει τη δυνατότητα να εισάγει το πλήρες όνομα του και το τηλέφωνο και το όνομα ενός δικού του ανθρώπου για να ειδοποιηθεί σε περίπτωση έκτακτης ανάγκης. Από την περιγραφή της εφαρμογής στην σελίδα του καταστήματος της Google (Google Play store) δεν γίνεται σαφές αν με την αποστολή του γεωγραφικού σημείου προς το 112 αποστέλλονται και οι παραπάνω πληροφορίες.



Εικόνα 24. Οθόνη επιλογής γλώσσας του περιβάλλοντος της 112 Iceland εφαρμογής

Με την επιλογή του εικονιδίου της σημαίας πάνω αριστερά στην ίδια οθόνη, επιλέγουμε την γλώσσα προτίμησης χρήσης στην εφαρμογή. Οι δυνατές επιλογές είναι Αγγλικά και Ισλανδικά. Αν επανέλθουμε στην κεντρική οθόνη, επιλέγοντας Check in your location, εμφανίζεται παράθυρο μηνύματος προς το χρήστη από το 112 όπου τον ενημερώνει ότι το Check in ολοκληρώθηκε επιτυχώς.



Εικόνα 25. Οθόνη ενημέρωσης του χρήστη από το 112 για το επιτυχές Check in στην 112 Iceland app

Η επόμενη εφαρμογή 112 BE, είναι η επίσημη εφαρμογή του Βελγίου για κλήσεις έκτακτης ανάγκης προς την υπηρεσία 112.



Εικόνα 26. Οθόνη εισόδου στην εφαρμογή 112 ΒΕ



vodafone GR 84% 6:27 μ.μ.

112 Registration

Please fill in all the fields, so that in case of an emergency we know who we're talking to.

First name* Thalia

Last name* Amargianou

Date of birth* 10/04/1983

Add more details

Mobile phone

Phone country code* Greece (+30)

Phone number*

I am deaf, hearing-impaired or have a speech impediment.

Next

Εικόνα 27. Οθόνη συμπλήρωσης προσωπικών πληροφοριών του χρήστη 112 BE app

Αφού ο χρήστης εισέλθει στην 1^η οθόνη της εφαρμογής, θα πρέπει να συμπληρώσει τα προσωπικά του στοιχεία, πλήρες όνομα, ημερομηνία γέννησης, αριθμό τηλεφώνου προσθέτοντας και τον αριθμό που προσδιορίζει τον κωδικό της εκάστοτε χώρας, συν την επιλογή ή μη του πεδίου που αφορά κάποιο πρόβλημα με την ακοή ή την ομιλία του χρήστη.



vodafone GR 84% 6:27 μ.μ.

☰ **112 Personal**

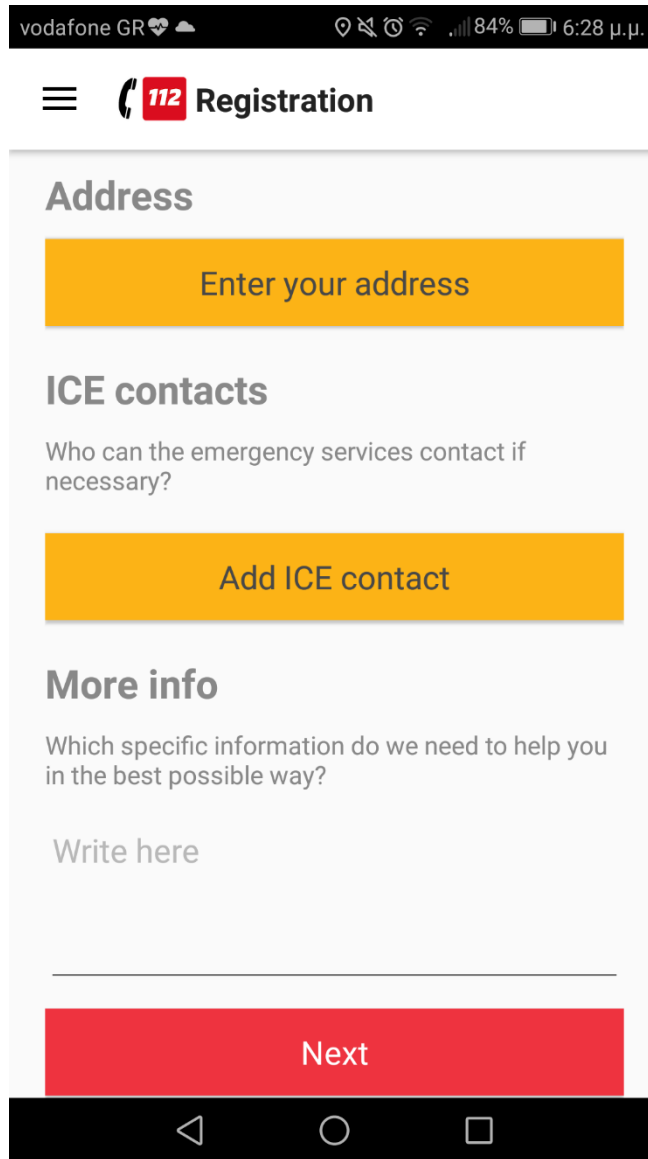
Gender Female ▼

Mother tongue Greek ▼

OK



Εικόνα 28. Οθόνη συμπληρωματικών πληροφοριών χρήστη - 112 BE app



Εικόνα 29. Οθόνη συμπλήρωσης πρόσθετων πληροφοριών του χρήστη - 112 BE app



vodafone GR 83% 6:29 μ.μ.

☰ **112 Add ICE contact**

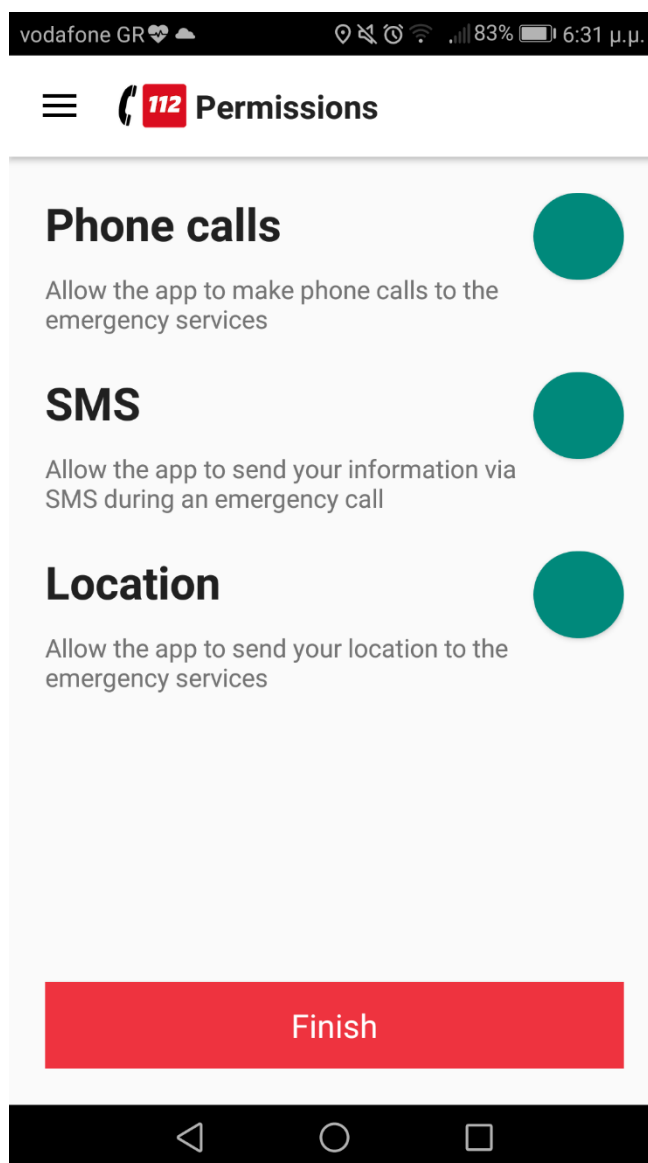
First name*	Panagiotis
Last name*	Amargianos
Phone country code*	Greece (+30) ▼
Phone number*	XXXXXXXXXX

Add

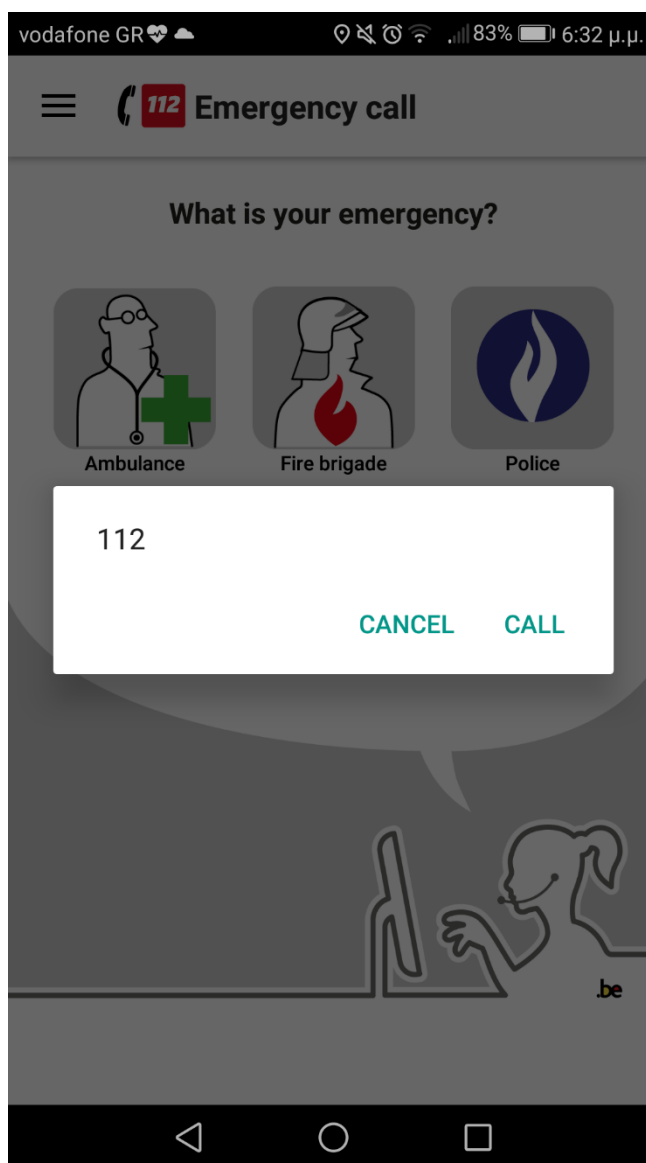
1	2 ABC	3 DEF	-
4 GHI	5 JKL	6 MNO	⌋
7 PRQS	8 TUV	9 WXYZ	⊗
* #	0 +	.	✓

▽ ○ □ ☰

Εικόνα 30. Οθόνη συμπλήρωσης στοιχείων πρόσθετης επαφής επικοινωνίας - 112 BE app



Εικόνα 31. Οθόνη επιβεβαίωσης πρόσβασης της εφαρμογής σε λειτουργίες της συσκευής - 112 BE app



Εικόνα 32. Οθόνη επιλογής κινδύνου - κλήση έκτακτης ανάγκης - 112 BE app

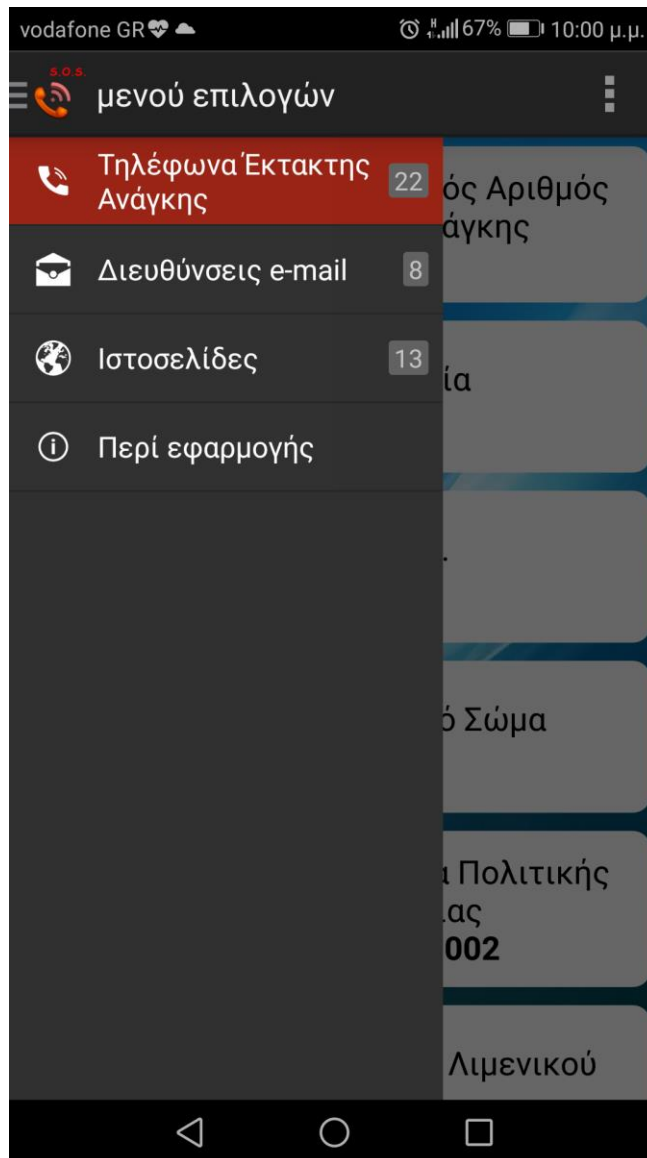
Εφόσον ο χρήστης έχει συμπληρώσει όλες τις προσωπικές σημαντικές πληροφορίες μπορεί να κάνει χρήση των επιλογών έκτακτης ανάγκης, σε περίπτωση που θεωρηθεί αναγκαίο. Δηλαδή από την οθόνη κλήσης έκτακτης ανάγκης ο χρήστης επιλέγει τον τύπο κινδύνου και κάνει την κλήση προς το 112, την Πυροσβεστική ή την αστυνομία.

Μια ακόμη εφαρμογή η οποία παρέχει τη δυνατότητα κλήσης του Ευρωπαϊκού αριθμού έκτακτης ανάγκης 112 σε περίπτωση κινδύνου είναι η S.O.S. Τηλέφωνα Έκτακτης Ανάγκης, μια εφαρμογή που έχει φτιαχτεί από Έλληνα προγραμματιστή και αφορά της κλήσεις έκτακτης ανάγκης στην Ελλάδα.



Εικόνα 33. Κεντρική οθόνη της εφαρμογής S.O.S.

Με την είσοδο στην εφαρμογή εισερχόμαστε στην κεντρική οθόνη, η οποία περιέχει μια λίστα από τηλέφωνα έκτακτης ανάγκης συμπεριλαμβανομένου και του Ευρωπαϊκού αριθμού έκτακτης ανάγκης 112 το οποίο βρίσκεται πρώτο στη λίστα. Επιλέγοντας κάποιο από αυτά τα τηλέφωνα, κάνουμε κλήση σε αυτό.

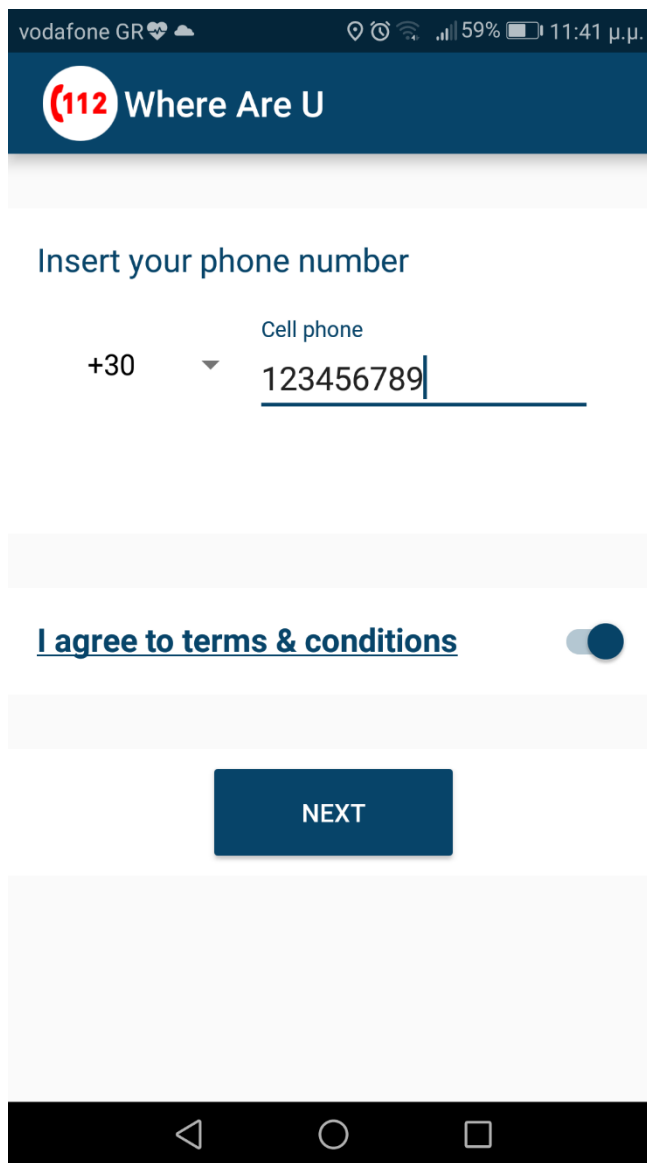


Εικόνα 34. Μενού επιλογών - S.O.S. app

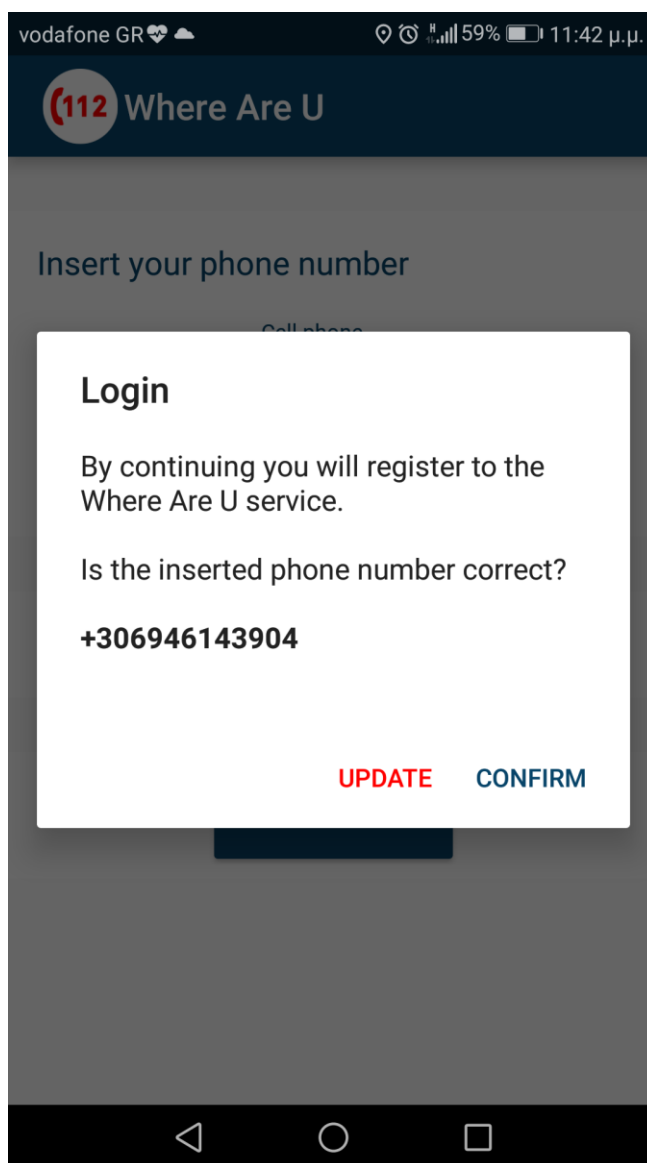
Εκτός από τη λίστα με τα τηλέφωνα έκτακτης ανάγκης που μπορεί να κάνει κλήση ο χρήστης σε περίπτωση που χρειαστεί, υπάρχει ένα μενού επιλογών που μας δίνει τη δυνατότητα να επιλέξουμε και να δούμε σε επόμενες οθόνες λίστες με διευθύνσεις email και ιστοσελίδες σημαντικών υπηρεσιών. Ομοίως, επιλέγοντας ένα email ή μια ιστοσελίδα, μας δίνεται η δυνατότητα να αποστείλουμε email μέσω των διαθέσιμων εφαρμογών του κινητού μας και αντιστοίχως να επιλέξουμε και να περιηγηθούμε σε κάποια από τις ιστοσελίδες των υπηρεσιών έκτακτης ανάγκης.



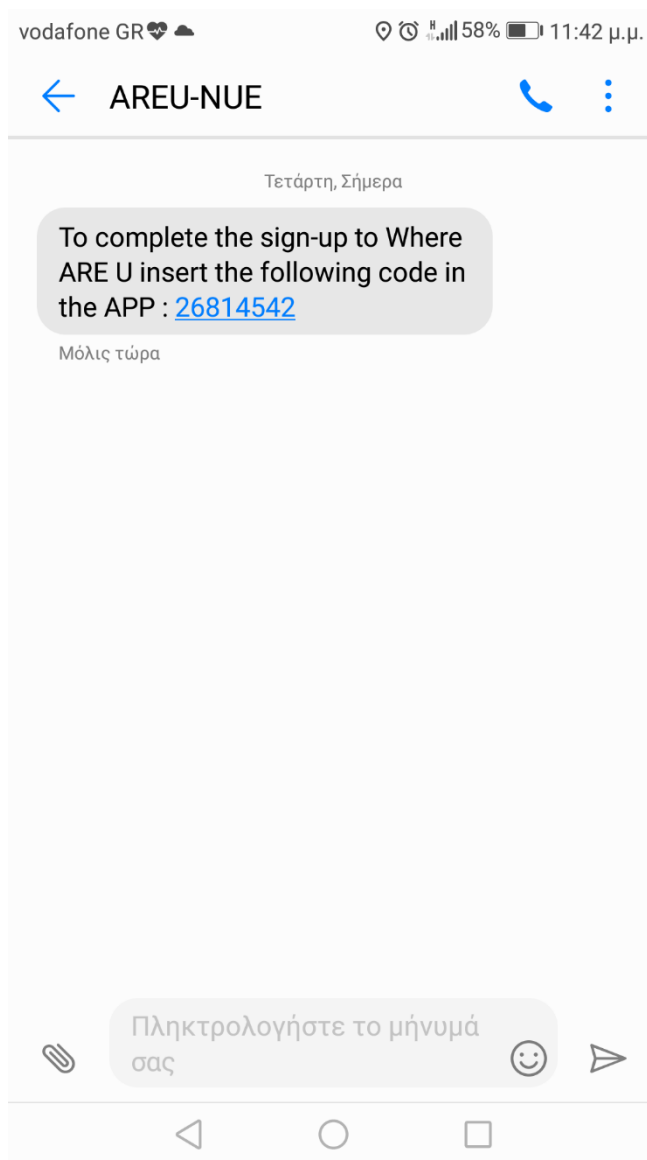
Η επόμενη εφαρμογή είναι η 112 Where are U, όπου κατά την είσοδο στην εφαρμογή ζητείται η δήλωση του αριθμού τηλεφώνου με προσδιορισμός της χώρας προέλευσης. Στο επόμενο βήμα και εφόσον γίνει επαλήθευση του αριθμού, αποστέλλεται στον αριθμό αυτό ένα μήνυμα με ένα κωδικό τον οποίο θα πρέπει υποτίθεται να χρησιμοποιήσουμε για να ολοκληρωθεί η εγγραφή του χρήστη στην εφαρμογή. Δεν εντόπισα κάποιο πεδίο εισαγωγής του συγκεκριμένου κωδικού. Στη συνέχεια δίνεται η δυνατότητα εισαγωγής επαφών από τις υπάρχουσες επαφές του τηλεφώνου ή και χειροκίνητα ώστε να ειδοποιηθούν σε περίπτωση έκτακτης ανάγκης. Κάθε φορά που ο χρήστης καλεί το 112 οι πληροφορίες αυτές θα είναι ορατές στο διαχειριστή του συστήματος διαχείρισης κλήσεων έκτακτης ανάγκης μαζί με την τοποθεσία που βρίσκεται ο χρήστης. Στην κεντρική οθόνη της εφαρμογής, την οθόνη κλήσεων έκτακτης ανάγκης, ο χρήστης μπορεί να καλέσει το 112 δηλώνοντας και τον τύπο άμεσης ανάγκης, ανάλογα με τον κίνδυνο που αντιμετωπίζει (Πυροσβεστική, Αστυνομία, Νοσοκομείο). Υπάρχει η δυνατότητα να δηλώσει με την κλήση του ότι δεν μπορεί να συνομιλήσει με το διαχειριστή του 112, οπότε επιλέγει “SILENT CALL” αντί για “VOICE CALL”, οπότε ενημερώνεται ο διαχειριστής ώστε να αντιμετωπίσει την κλήση κατάλληλα και να προβεί στις απαραίτητες ενέργειες.



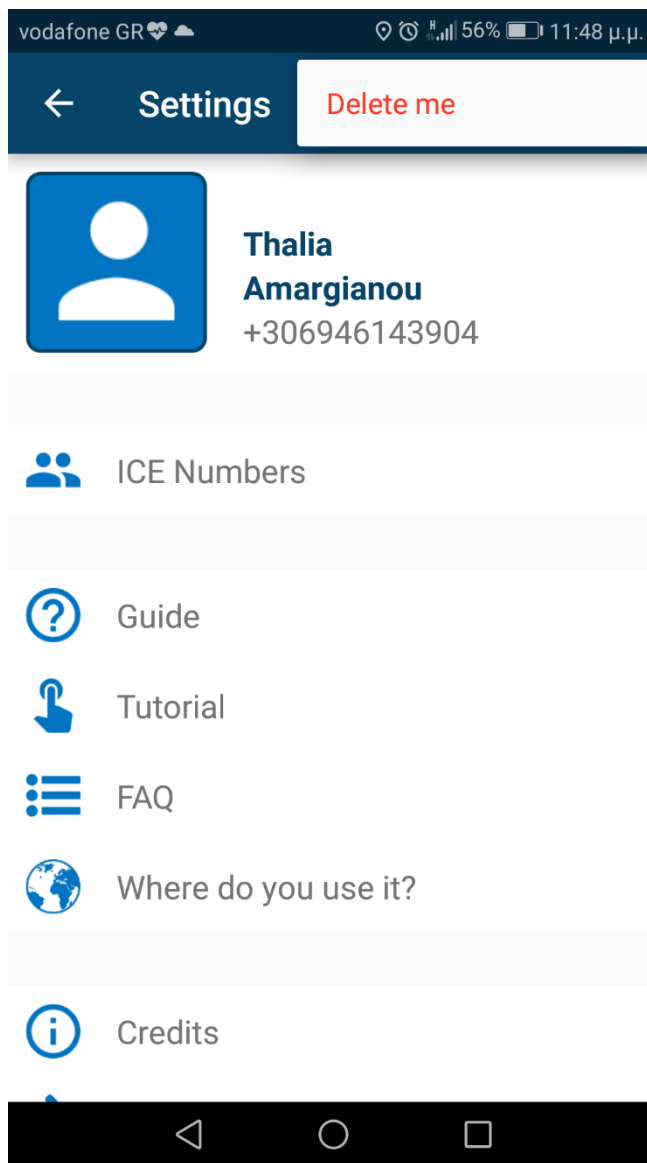
Εικόνα 35. Οθόνη δήλωσης αριθμού τηλεφώνου στην εφαρμογή - 112 Where are U app



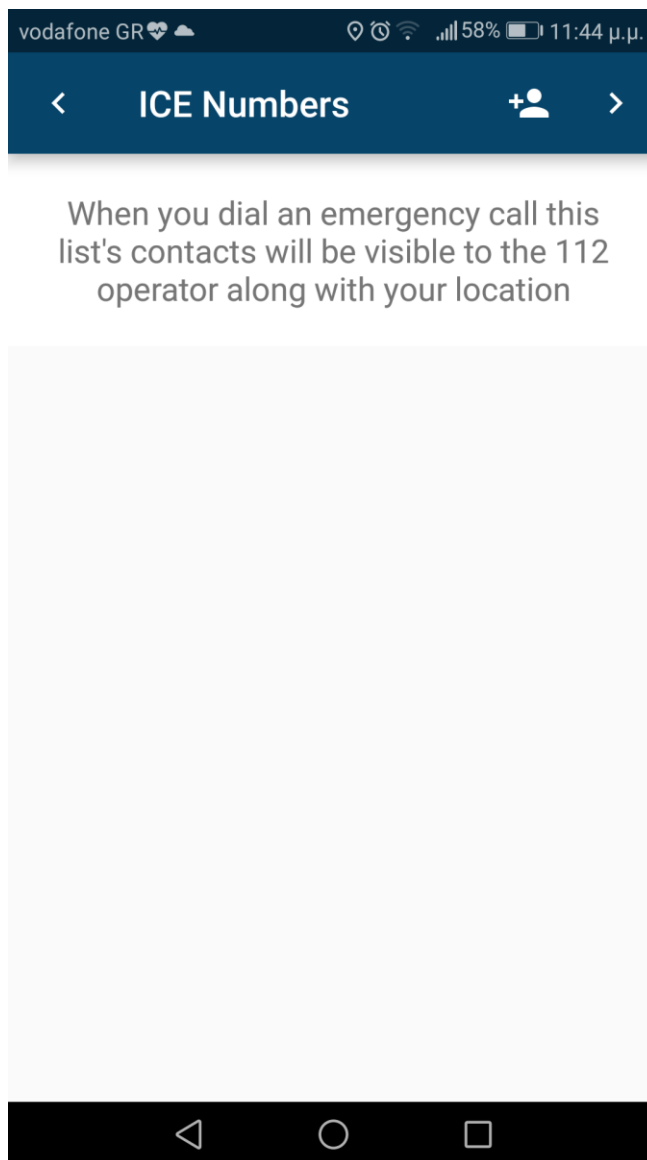
Εικόνα 36. Επαλήθευση αριθμού τηλεφώνου από την εφαρμογή - 112 Where are U app



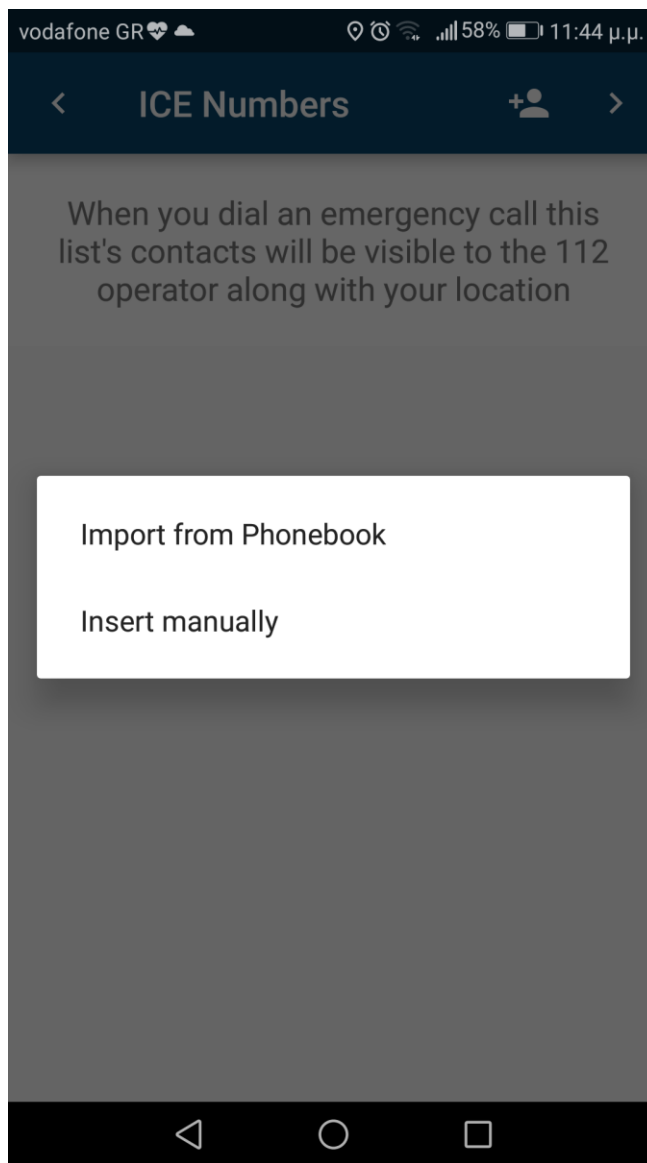
Εικόνα 37. Αποστολή κωδικού με sms για την ολοκλήρωση της εγγραφής στην εφαρμογή - 112 Where are U app



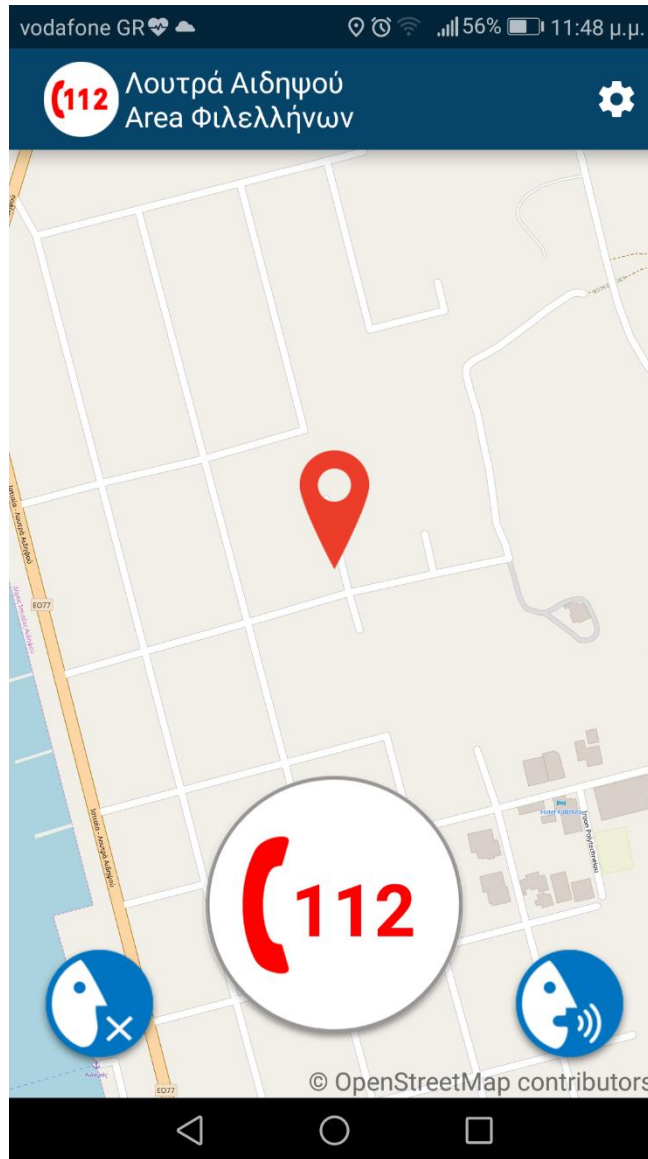
Εικόνα 38. Οθόνη ρυθμίσεων - δυνατότητα διαγραφής του χρήστη από τη ΒΔ της εφαρμογής - 112
Where are U app



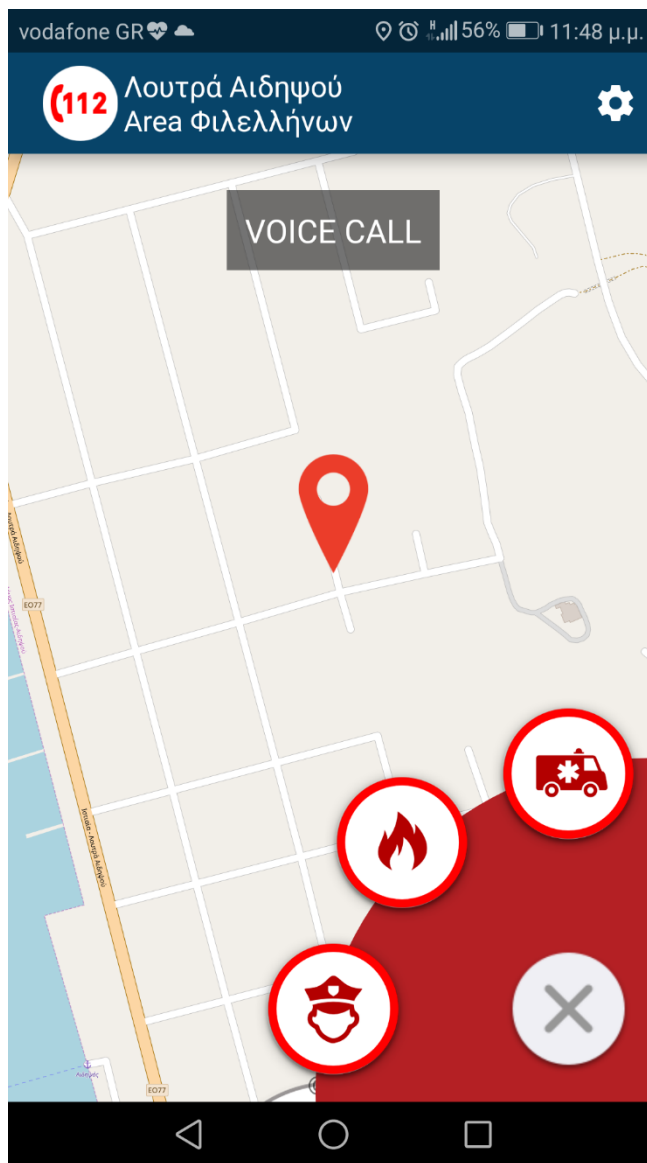
Εικόνα 39. Οθόνη προσθήκης επαφών ενημέρωσης σε περίπτωση έκτακτης ανάγκης - 112 Where are U app



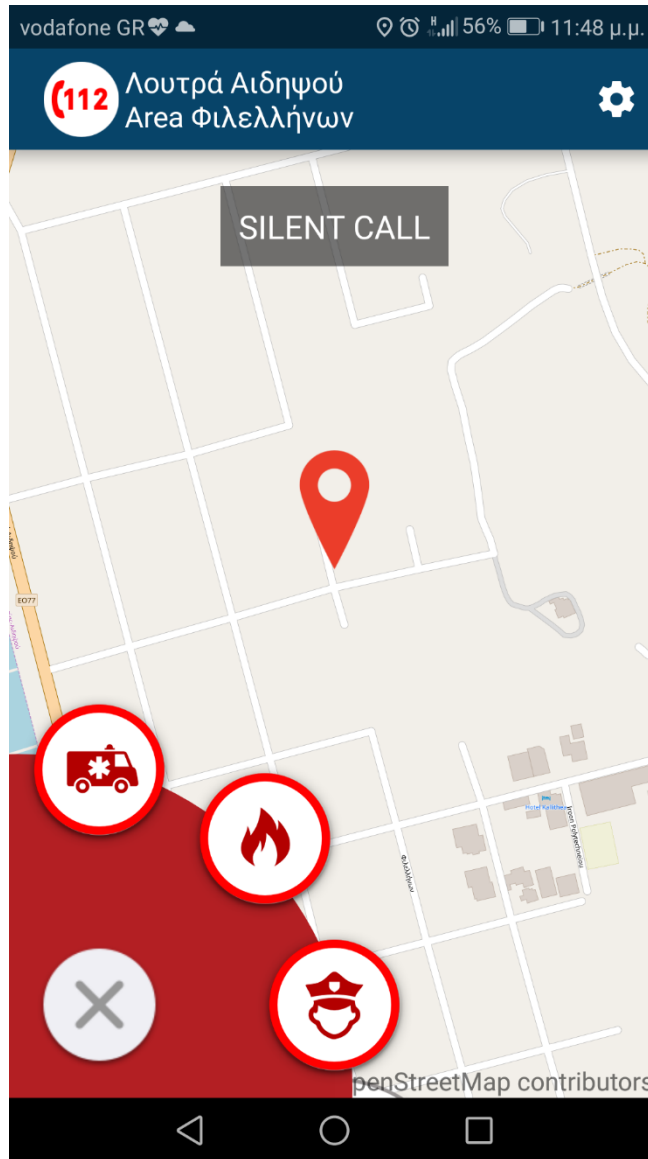
Εικόνα 40. Επιλογές προσθήκης επαφών - 112 Where are U app



Εικόνα 41. Οθόνη κλήσεων έκτακτης ανάγκης - 112 Where are U app



Εικόνα 42. Επιλογή "VOICE CALL" κλήσης έκτακτης ανάγκης - 112 Where are U app

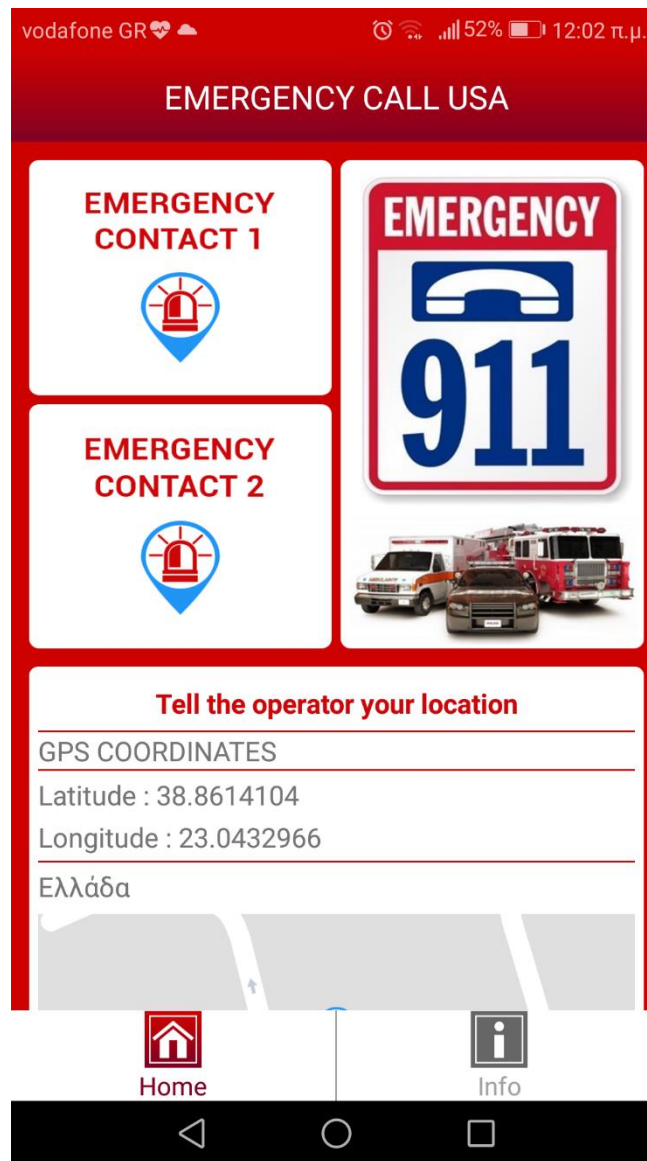


Εικόνα 43. "SILENT CALL" - 112 Where are U app

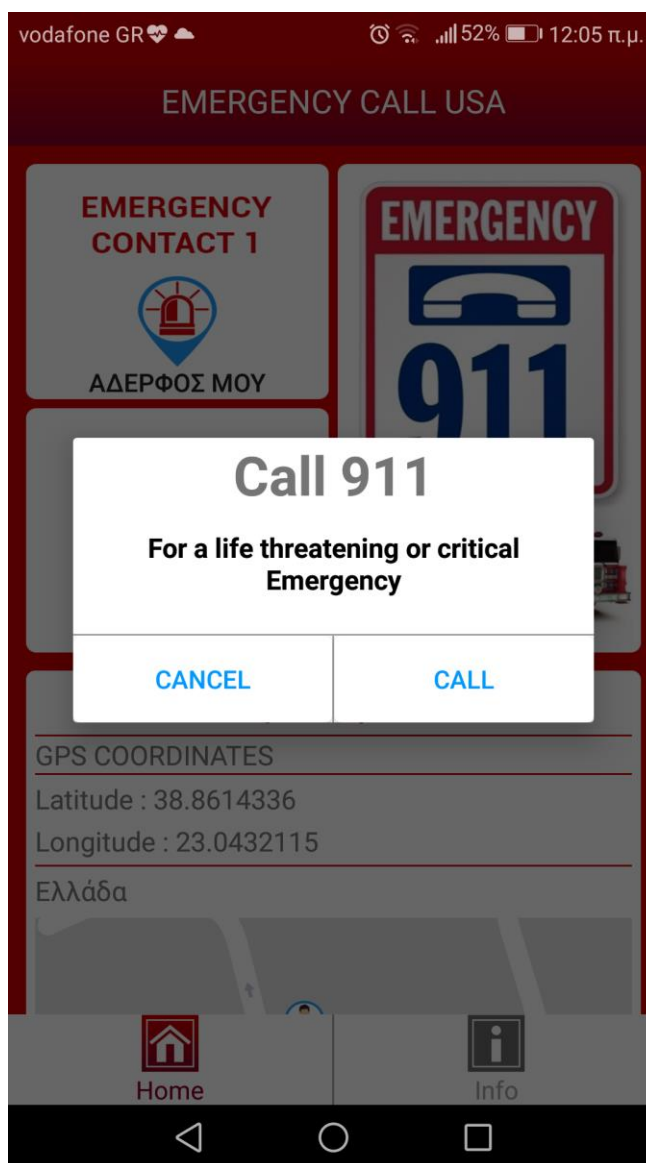
Μια τελευταία περιγραφή εφαρμογής κλήσης έκτακτης ανάγκης, πριν γίνει μια συνολική εκτίμηση των διαφοροποιήσεων και των χαρακτηριστικών της κάθε μιας από αυτές, είναι η εφαρμογή Emergency Call USA 9-1-1 που αφορά κλήσεις έκτακτης ανάγκης προς την υπηρεσία διαχείρισης κλήσεων 911 των Ηνωμένων Πολιτειών. Με την είσοδο στην εφαρμογή αυτή γίνεται άμεσος εντοπισμός του σημείου που βρίσκεται ο χρήστης. Δίνεται η δυνατότητα από την κεντρική οθόνη της εφαρμογής να προστεθούν δύο επαφές ανθρώπων που θα θέλαμε να έχουμε τη δυνατότητα να τους καλέσουμε άμεσα σε περίπτωση ανάγκης. Επίσης δίνεται η δυνατότητα άμεσης κλήσης προς το 911 και υπάρχει ο χάρτης και οι γεωγραφικές συντεταγμένες με το σημείο που βρίσκεται ο χρήστης τη δεδομένη στιγμή ώστε μόλις χρειαστεί να μπορεί



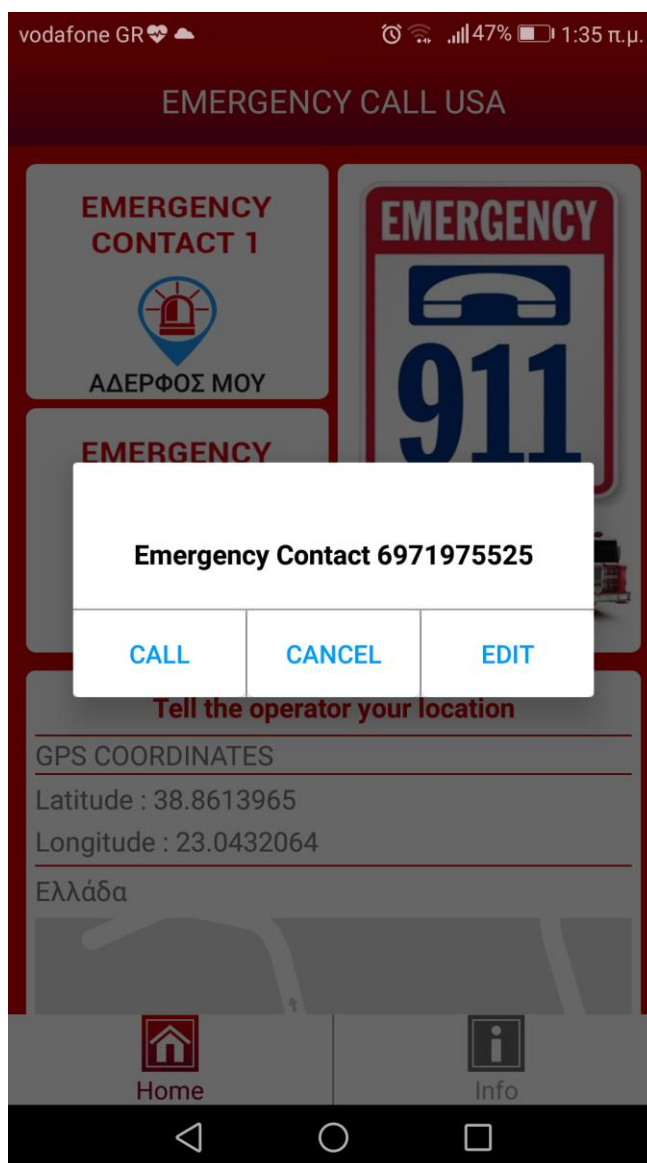
να δώσει αυτές τις πληροφορίες. Στην κεντρική οθόνη υπάρχουν δύο κουμπιά – επιλογές, όπου το ένα σε επαναφέρει στην κεντρική οθόνη της εφαρμογής και άλλο “info”, παραπέμπει σε οθόνη με πληροφορίες και τηλέφωνα από άλλες σημαντικές υπηρεσίες που μπορεί να χρειαστεί να επικοινωνήσει ο χρήστης.



Εικόνα 44. Κεντρική οθόνη εφαρμογής - 911 USA



Εικόνα 45. Κλήση προς το 911



Εικόνα 46. Κλήση επαφής κοντινού προσώπου σε περίπτωση ανάγκης - 911 USA app



Εικόνα 47. Οθόνη πρόσθετων πληροφοριών - 911 USA app

Η επιλογή μελέτης και σύγκρισης των παραπάνω εφαρμογών με την Emergency call android app της παρούσας εργασίας, έγινε αρχικά με βάσει τις δυνατότητες που παρέχουν στους χρήστες, με βασική λειτουργία τη δυνατότητα κλήσης προς την υπηρεσία διαχείρισης κλήσεων έκτακτης ανάγκης, τη δυνατότητα αποστολής μηνύματος κειμένου προς τις υπηρεσίες έκτακτης ανάγκης, αλλά και αποστολής του γεωγραφικού σημείου που βρίσκεται ο χρήστης της δεδομένη στιγμή. Επιλέχθηκε η εφαρμογή που αφορά τις Η.Π.Α με σκοπό να δούμε κατά πόσο έχουν προχωρήσει στη χρήση των τεχνολογιών αιχμής για την επικοινωνία με τη δική τους υπηρεσία διαχείρισης κλήσεων έκτακτης ανάγκης 911. Τέλος επιλέχθηκε μια ελληνική εφαρμογή για να γίνει η σύγκριση και με μια προσπάθεια που έχει γίνει για να καλύψει τις ανάγκες



επικοινωνίας με το 112, στην Ελλάδα. Οι τέσσερις από τις πέντε εφαρμογές που αναλύθηκαν αφορούν κλήσεις προς το 112 και η 5^η εφαρμογή αφορά κλήσεις προς το 911 της Αμερικής, όπως ήδη αναφέρθηκε. Στις τρεις πρώτες εφαρμογές έχουμε εγκαταστάσεις 100000 και πάνω και στην εφαρμογή για το 911 καθότι νέα εφαρμογή είναι κοντά στις 10000 οι εγκαταστάσεις σε έξυπνα κινητά. Ομοίως και στην ελληνική εφαρμογή. Όλες οι εφαρμογές έχουν ενημερωθεί μετά το 2016.

Εφαρμογή	Κλήση 112 /911	Αποστολή ακριβούς γεωγραφικής τοποθεσίας – GPS-based location	Προφίλ χρήστη	Περιοχή λειτουργίας	Χρήση και αποστολή πολυμεσικών στοιχείων	Push notifications / Αμφίδρομη επικοινωνία χρήστη - διαχειριστή με μηνύματα
112 Where are U	NAI	NAI	NAI	112 / 911	OXI	OXI
112 BE	NAI	NAI	NAI	Βέλγιο	OXI	NAI
112 Iceland	NAI	NAI	OXI	Ισλανδία	OXI	OXI
9-1-1 USA Emergency Calls	NAI (911)	NAI	OXI	Η.Π.Α	OXI	OXI
S.O.S. Τηλέφωνα εκτάκτων αναγκών	NAI	Λογικά μέσω των τηλεφωνικών πυλώνων	OXI	Ελλάδα	OXI	OXI
Emergency call android app	NAI	NAI	NAI	Πανευρωπαϊκά	NAI	NAI

Πίνακας 1. Εφαρμογές με λειτουργίες κλήσης προς το 112/911 και αποστολή ακριβούς γεωγραφικού σημείου

Με βάση τη σύγκριση που έχει γίνει ανάμεσα στις παραπάνω εφαρμογές και την ανάλυση των λειτουργιών τους ξεχωριστά παρατηρείται η ξεκάθαρη αλλαγή και προσθήκη δυνατοτήτων που φέρνει η εφαρμογή Emergency call android app που δημιουργήθηκε στα πλαίσια αυτής της εργασίας. Έχουμε μια εφαρμογή που δίνει στο χρήστη τη δυνατότητα να κάνει κλήση προς στο 112, να αποστείλει το γεωγραφικό σημείο στο οποίο βρίσκεται τη δεδομένη στιγμή, έχοντας ήδη δημιουργήσει το προφίλ του ο χρήστης με αποτέλεσμα να αποστέλλονται παράλληλα με τις παραπάνω πληροφορίες και οι προσωπικές πληροφορίες του χρήστη.

Emergency Call Android Application Development



Επίσης, υπάρχει η δυνατότητα επιλογής του τύπου κινδύνου που αντιμετωπίζει ο χρήστης της εφαρμογής και η επικοινωνία του με το διαχειριστή της εφαρμογής μέσω μηνυμάτων και ειδοποιήσεων.

Βέβαια, πολύ σημαντικό ή θα μπορούσε να πει κανείς και το σημαντικότερο όλων σε αυτή την εφαρμογή αλλά και στις παραπάνω εφαρμογές που ερευνηθήκαν είναι ο όσο το δυνατόν πιο ακριβής εντοπισμός του σημείου που βρίσκεται ο χρήστης τη στιγμή που επικοινωνεί με τις υπηρεσίες διαχείρισης εκτάκτων αναγκών. Στην περίπτωση της παρούσας εφαρμογής, δίνουμε τη δυνατότητα εντοπισμού μέσω το GPS based location και συμπληρωματικά στον εντοπισμό που γίνεται από το 112 μέσω της κλασσικής μεθόδου εντοπισμού από τους τηλεφωνικούς πυλώνες και την απόσταση από αυτούς της συσκευής ώστε να γίνει πιο ακριβής υπολογισμός. Συνδυαστικά και με τη χρήση της IP της συσκευής όταν βρίσκεται σε σημείο που υπάρχει σύνδεση στο διαδίκτυο και αποθήκευση του σημείου αυτής της τελευταίας IP μπορεί να γίνει ακόμα πιο ακριβής εντοπισμός.

Το ζήτημα εδώ είναι βέβαια, σε πραγματικά δεδομένα, έχοντας απέναντι μας το πραγματικό σύστημα διαχείρισης και υποδοχής των κλήσεων 112, σε ποιες από αυτές τις δυνατότητες και τεχνολογίες των έξυπνων κινητών τηλεφώνων είναι σε θέση να ανταποκριθεί η Ευρωπαϊκή υπηρεσία κλήσεων έκτακτης ανάγκης 112, λαμβάνοντας υπόψη ότι πάνω από το 70% των κλήσεων προς το 112 γίνεται πλέον από έξυπνα κινητά τηλέφωνα.

Χωρίς να υπάρχει λόγος να φορτωθεί ο αναγνώστης με πλεονάζουσα πληροφορία, για να γίνει μια ολοκληρωμένη σύνδεση με τα προηγούμενα, θα πρέπει να αναφερθεί ότι ο οργανισμός του Ευρωπαϊκού αριθμού έκτακτης ανάγκης 112 είναι σε συνεχείς προσπάθειες βελτίωσης των τεχνολογιών του και των υπηρεσιών που προσφέρει στους πολίτες που βρίσκονται σε κίνδυνο. Το Ευρωπαϊκό έργο HELP112 έχει σκοπό να βελτιώσει τον εντοπισμό της τοποθεσίας του χρήστη σε περιπτώσεις έκτακτης ανάγκης. Είναι ένα πιλοτικό έργο για το σχεδιασμό την υλοποίηση και την εκτέλεση της μεταφοράς δεδομένων GNSS κατά τη διάρκεια μιας κλήσης στο 112 σε ένα σημείο απάντησης δημόσιας ασφάλειας (PSAP). Η EENA σε συνεργασία με την Google εργάζονται επίσης για αυτό το σκοπό. Έτσι η Google ανακοίνωσε ότι από το Ιούλιο του 2016 όλες οι Android συσκευές με από την έκδοση Gingerbread και μετά θα περιλαμβάνουν την τεχνική Advanced Mobile Location, για τη βελτίωση της ποιότητας εντοπισμού των κλήσεων έκτακτης ανάγκης.

Έτσι λοιπόν σε σχέση και με τις όλο και πιο βελτιωμένες προσπάθειες που γίνονται από εφαρμογές αποστολής ειδοποιήσεων και κλήσεων έκτακτης ανάγκης προς τα σημεία απάντησης δημόσιας ασφάλειας, για την καλύτερη και πιο σαφή πληροφόρηση των διαχειριστών προς όφελος των χρηστών των εφαρμογών, γίνονται και οι αντίστοιχες προσπάθειες εξέλιξης και εξομίσωσης των τεχνολογιών που χρησιμοποιούνται από τα σημεία απάντησης των κλήσεων αυτών.



2.7 Μελλοντικές εργασίες εξέλιξης της εφαρμογής

Η συγκεκριμένη εργασία, emergency call android app, παρέχει πολύ σημαντικές δυνατότητες αποστολής ειδοποιήσεων σε περιπτώσεις έκτακτης ανάγκης. Δυνατότητες όπως, η αποστολή σήματος έκτακτης ανάγκης, η οποία ειδοποίηση θα αποστείλει προσωπικές πληροφορίες του χρήστη και το γεωγραφικό σημείο αποστολής της ειδοποίησης. Επιπλέον, αποστολή σήματος έκτακτης ανάγκης με δυνατότητα χρήσης της κάμερας και αποστολής φωτογραφίας από το σημείο που βρίσκεται ο χρήστης, συν την αποστολή των βασικών πληροφοριών του χρήστη και του γεωγραφικού σημείου αποστολής και στις δυο περιπτώσεις με επιλογή του τύπου κινδύνου που αντιμετωπίζει ο χρήστης και με τη δυνατότητα να αποστείλει και κάποιο μήνυμα προς το διαχειριστή. Βασική λειτουργία της εφαρμογής είναι και η απευθείας κλήση προς το 112 και σε άλλες υπηρεσίες διαχείρισης εκτάκτων αναγκών. Μια εφαρμογή όπως αυτή όμως με τέτοιο σκοπό έχει πολλές δυνατότητες εξέλιξης. Σε επόμενο επίπεδο υλοποίησης θα μπορούσαν να προστεθούν περισσότερες λειτουργίες, όπως:

- Λειτουργία κλήσεων μέσω internet. Δηλαδή κλήσεις μέσω voip τηλεφωνίας.
- Αποστολή μηνυμάτων ηλεκτρονικών ταχυδρομείου (emails), σε επιλεγμένες υπηρεσίες και πρόσωπα.
- Με χρήση αισθητήρων κίνησης και άλλων τεχνικών, θα μπορούσε να προστεθεί και λειτουργία ανίχνευσης πτώσης του χρήστη της συσκευής. Οπότε να αποστέλλονται ειδοποιήσεις έκτακτης ανάγκης, καταγράφοντας την κίνηση και την επιτάχυνση, σε περίπτωση διακοπής της κίνησης του χρήστη, να ενεργοποιείται κάποιος συναγερμός από την εφαρμογή και αν δεν υπάρξει απόκριση μετά από συγκεκριμένο χρόνο, να αποστέλλονται αυτόματα μηνύματα, ειδοποιήσεις έκτακτης ανάγκης σε τηλέφωνα που έχει επιλέξει ο χρήστης.
- Άλλη μια σημαντική λειτουργία που θα μπορούσε να προστεθεί είναι η επικοινωνία του χρήστη με τις επαφές που έχει ορίσει ως τηλέφωνα επικοινωνίας σε περιπτώσεις έκτακτης ανάγκης κατευθείαν μέσα από την εφαρμογή και η αυτόματη ενημέρωση των επαφών αυτών ταυτόχρονα με το 112.



Κεφάλαιο 3^ο

3. Συμπεράσματα

Η Emergency call android app, είναι μια πολύ βασική, σημαντική εφαρμογή που θα έπρεπε να έχει ο καθένας στο “smartphone” του. Είναι μια προσωπική εφαρμογή που μπορεί να έχει ο χρήστης τη συσκευή του για λόγους ασφαλείας, καθότι μέσα από αυτή μπορεί να στείλει ειδοποιήσεις έκτακτης ανάγκης σε επιλεγμένες επαφές και υπηρεσίες καθώς και να καλέσει απευθείας στον Ευρωπαϊκό αριθμό έκτακτης ανάγκης 112 αλλά και άλλες σημαντικές υπηρεσίες. Η δυνατότητα εντοπισμού του γεωγραφικού σημείου που βρίσκεται ο χρήστης τη δεδομένη στιγμή έκτακτης ανάγκης, προστίθεται ως βασική και κεντρική πληροφορία στο σύνολο των πληροφοριών που αποστέλλονται σε κάθε μήνυμα ειδοποίησης προς την υπηρεσία διαχείρισης των ειδοποιήσεων έκτακτων αναγκών.

Η εφαρμογή τμηματικά χωρίζεται στο Mobile app τμήμα της που αφορά τον εκάστοτε χρήστη της εφαρμογής από το smartphone του, όπου η υλοποίηση έγινε για android περιβάλλον με χρήση των τεχνολογιών Java για android και στο Web app τμήμα της, όπου υλοποιήθηκε με τη χρήση τεχνολογιών της NodeJS για το server και Angular framework για το FrontEnd. Τα δύο αυτά τμήματα της εφαρμογής αποθηκεύουν τα δεδομένα όποτε ορίζεται ως απαραίτητο με τη χρήση εγγράφων δεδομένων σε μια βάση MongoDB.

Πρόκειται ξεκάθαρα, για μια εφαρμογή που με την υλοποίηση και των πιο προηγμένων λειτουργιών που αναφέρθηκαν στην ενότητα 2.7, μπορεί να θεωρηθεί μέγιστης σημασίας για τους χρήστες έξυπνων κινητών των οποίων το επάγγελμά τους έχει μεγάλο βαθμό επικινδυνότητας αλλά και οι δραστηριότητες στις οποίες συμμετέχουν, επίσης. Όπως και για ανθρώπους ηλικιωμένους ή ανθρώπους με ιδιαίτερες παθήσεις, όπως αλτσχάιμερ.

Η εφαρμογή emergency calls, είναι μια εφαρμογή που έχει πυροδοτήσει το ενδιαφέρον μου να ασχοληθώ περισσότερο με mobile εφαρμογές και ειδικά τέτοιου τύπου. Δηλαδή με εφαρμογές που μπορούν να προσφέρουν μια δυνατότητα σε ανθρώπους οι οποίοι μια δεδομένη στιγμή δεν είναι σε θέση να βοηθήσουν το εαυτό τους. Δεν ήταν η πρώτη μου απόπειρα να ασχοληθώ με τη δημιουργία android εφαρμογής. Στα πλαίσια των μαθημάτων χρειάστηκε να ασχοληθώ ξανά. Η τελευταία ενασχόληση μου όμως είναι η emergency call app και θα ήθελα να γίνει η αρχή ώστε να μπορέσω να εντρυφήσω σε καινούργιες τεχνολογίες ώστε να υλοποιήσω πιο καινοτόμες και χρήσιμες εφαρμογές.



Κεφάλαιο 4^ο

4. Βιβλιογραφικές Πηγές

1. **Kathy Sierra, Bert Bates.** Head First Java 2nd Edition, 2009
2. **Down Griffiths, David Griffiths.** Head First Android Development 2nd Edition, A Brain - Friendly Guide, 2017
3. **Ian Darwin.** Android Cookbook 2nd Edition, Problems and Solutions for Android Developers, 2017
4. **Simon Holmes.** Getting MEAN with Mongo, Express, Angular and Node, 2015
5. **Brad Dayley.** Node.js, MongoDB and AngularJS Web Development (Developer's Library) 1st Kindle Edition, 2014
6. **Stephen Samuel, Stefan Bocutiu.** Programming Kotlin, 2017
7. **European Emergency Number Association.** <http://www.eena.org>
8. **112 Where are U app.**
<https://play.google.com/store/apps/details?id=it.Beta80Group.whereareu>
9. **112 Iceland app.**
<https://play.google.com/store/apps/details?id=is.stokkur.savage.android&hl=el>
10. **Emergency Call USA 9-1-1 (911).**
<https://play.google.com/store/apps/details?id=com.familytrak.USA>
11. **112 BE app.**
<https://play.google.com/store/apps/details?id=be.Nextel.EmergencyApp112&hl=el>
12. **S.O.S. app τηλέφωνα κλήσης έκτακτης ανάγκης.**
<https://play.google.com/store/apps/details?id=com.themelix.mysocalls&hl=el>

