

ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2019-2020

ΤΕΛΙΚΗ ΕΡΓΑΣΙΑ - ANDROID

ΣΥΓΧΡΟΝΑ ΘΕΜΑΤΑ ΤΕΧΝΟΛΟΓΙΑΣ ΛΟΓΙΣΜΙΚΟΥ

Ανάλυση απαιτήσεων

Π16036 – ΙΩΑΝΝΙΔΗΣ ΠΑΝΑΓΙΩΤΗΣ

Π16097 – ΝΙΚΑΣ ΔΙΟΝΥΣΙΟΣ

Π16112 – ΠΑΡΑΒΑΝΤΗΣ ΑΘΑΝΑΣΙΟΣ

Περιεχόμενα

1	Εισαγωγή.....	2
1.1	Στόχοι της εργασίας.....	2
1.2	Ορισμός του προβλήματος προς επίλυση.....	4
2	Σύντομη παρουσίαση της RUP	5
2.1	Φάση Σύλληψης.....	5
2.2	Φάση Επεξεργασίας.....	5
2.3	Φάση Κατασκευής	6
2.4	Φάση Μετάβασης.....	6
3	Φάση: Έναρξη (Inception).....	7
3.1	Σύλληψη απαιτήσεων.....	7
3.2	Ανάλυση-Σχεδιασμός.....	7
3.2.1	Διαγράμματα Περιπτώσεων Χρήσης	7
4	Φάση: Εκπόνηση Μελέτης (Elaboration).....	8
4.1	Ανάλυση-Σχεδιασμός.....	8
4.1.1	Διαγράμματα Περιπτώσεων Χρήσης (2 ^η έκδοση)	8
4.2	Υλοποίηση-Έλεγχος.....	8
4.2.1	Υλοποίηση: 1 ^η εκτελέσιμη έκδοση	8
4.2.2	Αναφορά ελέγχου για την 1 ^η εκτελέσιμη έκδοση	8
5	Φάση: Κατασκευή (Construction).....	9
5.1	Ανάλυση-Σχεδιασμός.....	9
5.1.1	Διαγράμματα Περιπτώσεων Χρήσης (3 ^η έκδοση)	9
5.2	Υλοποίηση-Έλεγχος.....	9
5.2.1	Υλοποίηση: Τελική εκτελέσιμη έκδοση	9
5.2.2	Αναφορά ελέγχου για την τελική εκτελέσιμη έκδοση	10
6	Διαγράμματα κλάσεων	11
6.1	Διάγραμμα κλάσεων Alert	11
6.2	Διάγραμμα κλάσεων Doctor	12
6.3	Διάγραμμα κλάσεων Drive	13
6.4	Διάγραμμα κλάσεων Disability	14
6.5	Διάγραμμα κλάσεων Home	15

1 Εισαγωγή

1.1 Στόχοι της εργασίας

ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ: Με βάση το ειδικό θέμα της κάθε ομάδας, το οποίο αναλύεται στη βασική του μορφή στην επόμενη σελίδα, καλείστε να αναπτύξετε λογισμικό, καθώς και να το συνοδεύσετε με την αντίστοιχη ανάπτυξή του, βάσει των παρακάτω απαιτήσεων.

Σύμφωνα με το ειδικό θέμα της εργασίας που έχει δοθεί στην ομάδα σας να γίνουν τα εξής:

1. Ανάλυση απαιτήσεων. Έγγραφο των βασικών λειτουργικών απαιτήσεων της εφαρμογής που θα αναπτύξετε, με δομημένο τρόπο.
2. Ανάπτυξη μέρους του μοντέλου ανάπτυξης λογισμικού (Rational Unified Process) χρησιμοποιώντας 2 διαγράμματα της UML, του διαγράμματος τάξεων και του διαγράμματος περιπτώσεων χρήσης. Μπορεί να γίνει χρήση ενός εργαλείου CASE της επιλογής σας. Προτεινόμενα εργαλεία CASE: Rational Rose, Visual Studio Ultimate, ArgoUML, κ.λπ.
3. Ανάπτυξη της εργασίας. Χρήση της γλώσσας ή του εργαλείου που έχει δηλώσει η ομάδα σας. Σε κάθε περίπτωση, προτείνεται η χρήση κάποιας αντικειμενοστραφούς γλώσσας προγραμματισμού.

Διευκρινήσεις:

- Τα διαγράμματα που θα υλοποιήσετε κατά την ανάπτυξη του ερωτήματος 2 θα πρέπει υποχρεωτικά να σχετίζονται με την υλοποιημένη εργασία του ερωτήματος 3 και το αντίστροφο.
- Για οποιοσδήποτε διευκρινίσεις/απορίες σε σχέση με τη γενική περιγραφή του θέματός σας να προσέλθετε στο μάθημα για να τις συζητήσουμε.

1. Δομή εργασίας

- Σκοπός της εργασίας σας είναι να υλοποιήσετε μια εφαρμογή “mobile application” για το λειτουργικό σύστημα Android. Το ακριβές περιεχόμενο της τελικής σας εφαρμογής θα εξαρτάται από το ειδικό θέμα εργασίας που έχει αναλάβει η ομάδα σας.
- Η εφαρμογή σας θα πρέπει να διαθέτει βάση δεδομένων, SQLite ή SharedPreferences, την οποία φυσικά να χρησιμοποιεί/αξιοποιεί. Επίσης επιβάλλεται και η χρήση κάποιας απομακρυσμένης βάσης δεδομένων, π.χ. Firebase, ή κάποιο web service της επιλογής σας.
- Η εφαρμογή σας θα πρέπει να αξιοποιεί τουλάχιστον 2 αισθητήρες της Android συσκευής, με επιθυμητή τη χρήση 3 (ή και παραπάνω εφόσον το επιθυμείτε) αισθητήρων. Είναι επίσης υποχρεωτικό να χρησιμοποιήσετε τον αισθητήρα gps της συσκευής σας, αξιοποιώντας έτσι και πληροφορίες γεωεντοπισμού.
- Η εφαρμογή σας θα πρέπει να υποστηρίζει ταυτόχρονα δύο γλώσσες (π.χ. Ελληνικά και Αγγλικά) σε σχέση με την πληροφορία που υπάρχει στο UI της.
- Είναι επιθυμητό, το mobile application που θα δημιουργήσετε να έχει όσο το δυνατόν πιο φιλικό User Interface και να είναι εύχρηστο ως προς τη χρήση του. Ο κώδικας να είναι βελτιστοποιημένος σε σχέση με τη χρήση των πόρων του κινητού και φυσικά να μην κάνει αλόγιστη σπατάλη ενέργειας.
- Για να αξιολογηθεί η εργασία σας με άριστα θα πρέπει να αναπτύξετε την εφαρμογή σας με τη χρήση του συνδυασμού Java-Android SDK (π.χ. μέσω Eclipse, ή Android Studio). Αν ωστόσο

χρησιμοποιήσετε κάποια hybrid (Phone Gap) ή web-based (html/html5) προσέγγιση, η εργασία σας θα αξιολογηθεί με άριστα το 80% του βαθμού της εργασίας.

2. Συγκέντρωση Πληροφοριών

Πριν την υλοποίηση της ιστοσελίδας ζητείται να προηγηθεί ανάλυση απαιτήσεων και έρευνα για παρόμοιες εφαρμογές. Έτσι θα καταγράψετε καλύτερα τις απαιτήσεις για τη δημιουργία της εφαρμογής σας και ταυτόχρονα θα συλλέξετε και ιδέες ή/και πρόσθετες πληροφορίες για αυτήν.

3. Τρόπος Παράδοσης

Για την παράδοση της τελικής εργασίας, θα τοποθετήσετε σε ένα αρχείο zip τα εξής: – Όλα τα αρχεία κώδικα που χρησιμοποιήσατε για τη δημιουργία του project – Το παραγόμενο εκτελέσιμο αρχείο .apk – Βάση δεδομένων με τους πίνακες και τα δεδομένα (τη ΒΔ θα πρέπει να την έχετε κάνει export) – Σύντομο βίντεο επίδειξης των λειτουργιών της εφαρμογής που υλοποιήσατε.

Ειδικό θέμα εργασίας ομάδας: Εφαρμογή εκτάκτων αναγκών

1.2 Ορισμός του προβλήματος προς επίλυση

Η εργασία που καλούμαστε να υλοποιήσουμε επιχειρεί να λύσει ένα κομβικό ζήτημα ηλικιωμένων ανθρώπων: η ασφάλειά τους. Έχοντας υπ’ όψη αυτό το σκεπτικό, μπορούμε να εκκινήσουμε μια διαδικασία ανάλυσης των απαιτήσεων μιας τέτοιας εφαρμογής. Οι λειτουργίες που θα περιέχονται, πρέπει να σχετίζονται με καθημερινές δραστηριότητες ανθρώπων αλλά και με περιπτώσεις έκτακτης ανάγκης. Μια τέτοια εφαρμογή οφείλει να περιέχει εργαλεία ουσιαστικά αλλά και αποτελεσματικά. Η προσέγγιση και η επίλυση του προβλήματος είναι εφικτή, εφόσον τα έξυπνα κινητά τηλέφωνα που διαθέτουν όλοι στις μέρες μας, περιέχουν ήδη χρήσιμες δυνατότητες που μπορούμε να εκμεταλλευτούμε.

2 Σύντομη παρουσίαση της RUP

Ο κύκλος ζωής του λογισμικού (software life-cycle) σκιαγραφεί τη ζωή του προγράμματος λογισμικού από τη στιγμή της γέννησής του μέχρι τη στιγμή της αντικατάστασης ή της εγκατάλειψής του. Ο κύκλος ζωής του λογισμικού στην RUP υποδιαιρείται σε τέσσερις συνεχόμενες φάσεις, οι οποίες είναι:

- Η Φάση Σύλληψης (Inception Phase)
- Η Φάση Επεξεργασίας (Elaboration Phase)
- Η Φάση Κατασκευής (Construction Phase)
- Η Φάση Μετάβασης (Transition Phase)

Ο κύκλος ανάπτυξης είναι ένα πέρασμα από τέσσερις φάσεις. Το κάθε πέρασμα από αυτές τις φάσεις ονομάζεται **γενιά (generation)**.

Σε περίπτωση που το προϊόν συνεχίσει να εξελίσσεται στην επόμενη γενιά του, έχουμε έναν νέο διαδοχικό κύκλο. Αυτοί οι κύκλοι ονομάζονται **κύκλοι εξέλιξης (evolution cycles)**.

Καθεμία από τις φάσεις ολοκληρώνεται με την παράδοση κάποιων ορόσημων. Στο τέλος κάθε φάσης πραγματοποιείται μια αποτίμηση η οποία κρίνει εάν οι στόχοι της φάσης έχουν ικανοποιηθεί προκειμένου να αρχίσει η επόμενη φάση.

2.1 Φάση Σύλληψης

Τα αποτελέσματα της φάσης Σύλληψης είναι τα εξής :

- Μία γενική περιγραφή των βασικών απαιτήσεων του έργου, τα σημεία κλειδιά και οι βασικοί περιορισμοί.
- Ένα αρχικό μοντέλο περιπτώσεων χρήσης (ολοκληρωμένο κατά 10%-20%)
- Ένα αρχικό γλωσσάριο του έργου.
- Μια αρχική εκτίμηση των κινδύνων.
- Μία αρχική επιχειρηματική περίπτωση, η οποία περιέχει το επαγγελματικό πλαίσιο, κριτήρια επιτυχίας και οικονομική πρόβλεψη.
- Ένα σχέδιο του έργου (project plan), το οποίο δείχνει φάσεις και επαναλήψεις.
- Ένα επιχειρηματικό μοντέλο (business model) αν είναι απαραίτητο.
- Ένα ή διάφορα πρωτότυπα.

2.2 Φάση Επεξεργασίας

Το αποτέλεσμα της φάσης επεξεργασίας είναι:

- Ένα μοντέλο περιπτώσεων χρήσης (ολοκληρωμένο τουλάχιστον κατά 80%)
- Συμπληρωματικές απαιτήσεις για τις μη λειτουργικές απαιτήσεις και απαιτήσεις που δεν έχουν σχέση με μια συγκεκριμένη περίπτωση χρήσης.
- Ένα εκτελέσιμο αρχιτεκτονικό πρωτότυπο
- Μια περιγραφή της αρχιτεκτονικής του λογισμικού

- Επανεξέταση της λίστας κινδύνων και της επιχειρηματικής περίπτωσης
- Ένα σχέδιο ανάπτυξης για το ολικό έργο
- Το έργο μπορεί να ματαιωθεί ή να επανεξεταστεί σοβαρά η πραγματοποίησή του αν αποτύχει να περάσει από αυτό το ορόσημο

2.3 Φάση Κατασκευής

Αποτελείται τουλάχιστον από τα εξής:

- Το προϊόν λογισμικού
- Το εγχειρίδιο του χρήστη
- Μια περιγραφή της παρούσας άδειας έκδοσης

Η μετάβαση μπορεί να αναβληθεί αν το έργο αποτύχει να περάσει αυτό το ορόσημο.

2.4 Φάση Μετάβασης

Ο σκοπός της φάσης είναι η μετάβαση του λογισμικού προϊόντος στην κοινότητα των χρηστών. Αφού το προϊόν παραδοθεί στους τελικούς χρήστες, συνήθως εντοπίζονται προβλήματα που απαιτούν την ανάπτυξη νέων εκδόσεων.

Σε αυτό το σημείο αποφασίζεται αν οι στόχοι έχουν επιτευχθεί και αν θα πρέπει να ξεκινήσει ο επόμενος κύκλος εξέλιξης.

3 Φάση: Έναρξη (Inception)

3.1 Σύλληψη απαιτήσεων

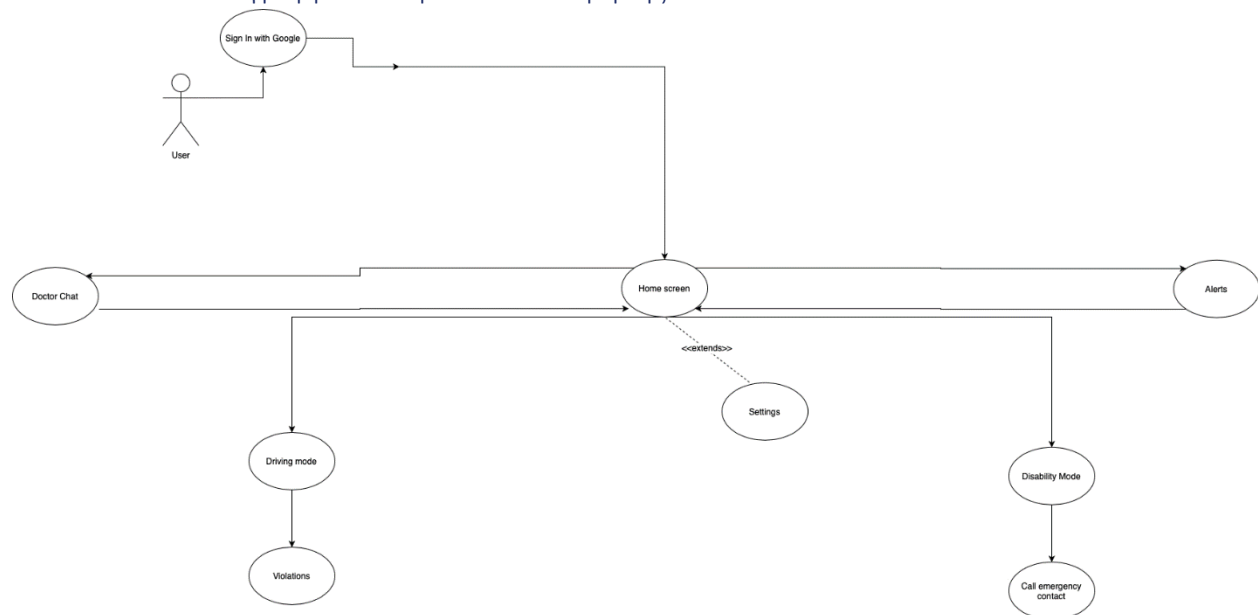
Η εφαρμογή “Instant Service” που πρέπει να υλοποιήσουμε μπορεί να περιέχει τις εξής δυνατότητες:

- Άμεση επικοινωνία με προσωπικό γιατρό για ιατρικές συμβουλές όλων των θεμάτων.
- Ειδοποίηση χρηστών για φυσικές καταστροφές, ασθένειες και υπενθυμίσεις.
- Ειδοποίηση άλλων προσώπων σε περίπτωση έκτακτης ανάγκης.
- Έλεγχος ορίων ταχύτητας στην οδήγηση.
- Εύχρηστη κλήση κινητών τηλεφώνων, αποστολή μηνυμάτων και επικοινωνία με νοσοκομείο.
- Αποφυγή ατυχημάτων (πχ. πτώσης) κυρίως για ηλικιωμένους με προβλήματα αστάθειας.

Έχοντας υπ’ όψη μια αρχική ιδέα της εφαρμογής, είμαστε έτοιμοι να προχωρήσουμε από τη 1^η στη 2^η φάση σχεδίασης.

3.2 Ανάλυση-Σχεδιασμός

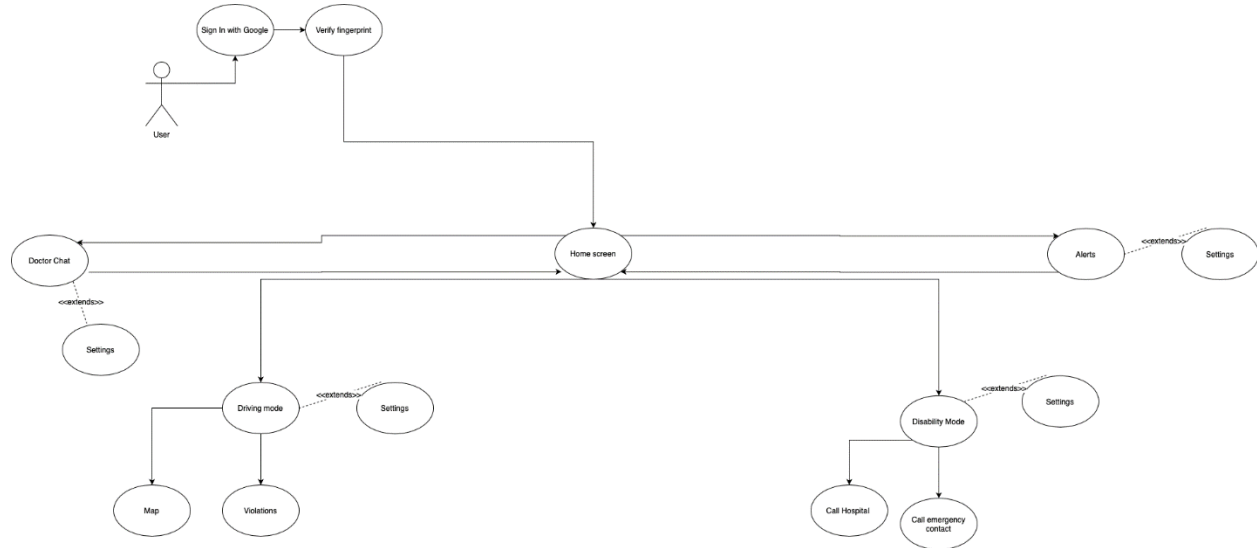
3.2.1 Διαγράμματα Περιπτώσεων Χρήσης



4 Φάση: Εκπόνηση Μελέτης (Elaboration)

4.1 Ανάλυση-Σχεδιασμός

4.1.1 Διαγράμματα Περιπτώσεων Χρήσης (2^η έκδοση)



4.2 Υλοποίηση-Έλεγχος

4.2.1 Υλοποίηση: 1^η εκτελέσιμη έκδοση

Στη πρώτη εκτελέσιμη έκδοση της εφαρμογής μας, παρατηρούνται μερικά τεχνικά ζητήματα αλλά προκύπτουν και νέες απαιτήσεις.

- Ανάγκη προσδιορισμού των αισθητήρων κινητού που θα χρησιμοποιούνται.
- Ανάγκη σελίδας ρυθμίσεων για την προσαρμογή της εφαρμογής.
- Ανάγκη σελίδων βοήθειας, ήχων και δονήσεων.
- Ανάγκη ταυτοποίησης για την αύξηση της ασφάλειας του χρήστη.

Μεταβαίνοντας στη 3^η φάση, ολοκληρώνουμε όλο και περισσότερο τις γενικές απαιτήσεις της εφαρμογής.

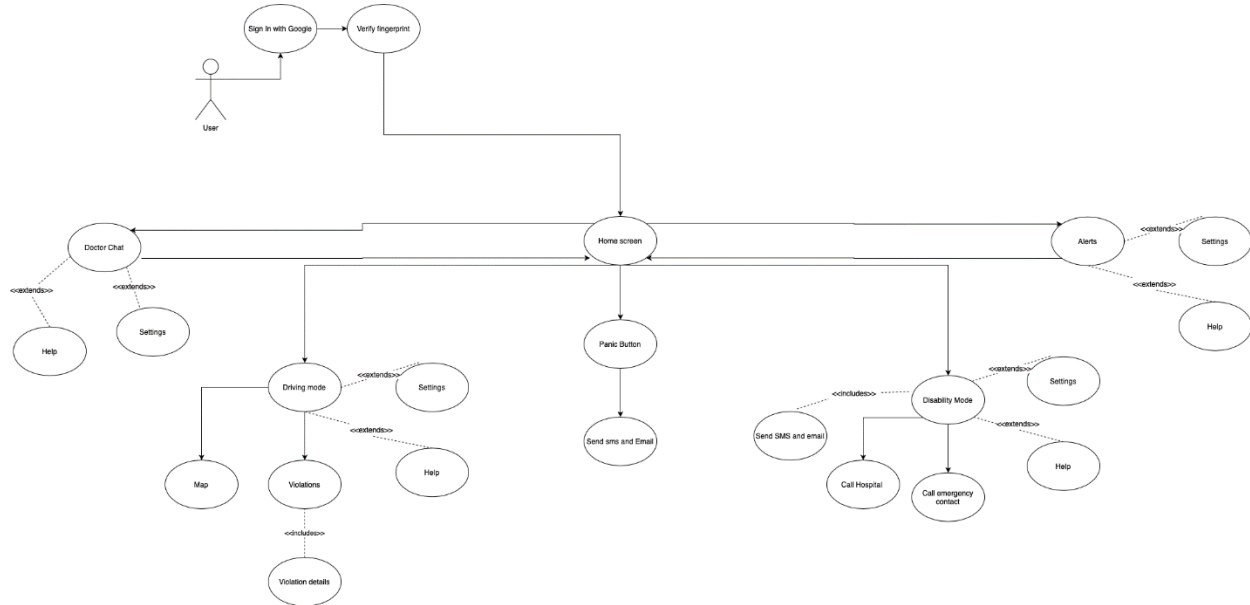
4.2.2 Αναφορά ελέγχου για την 1^η εκτελέσιμη έκδοση

Διαπιστώθηκαν μερικά προβλήματα στη λειτουργία της εφαρμογής, τα οποία αντιμετωπίζονται καθώς η ομάδα μας αναπτύσσει την εφαρμογή.

5 Φάση: Κατασκευή (Construction)

5.1 Ανάλυση-Σχεδιασμός

5.1.1 Διαγράμματα Περιπτώσεων Χρήσης (3^η έκδοση)



5.2 Υλοποίηση-Έλεγχος

5.2.1 Υλοποίηση: Τελική εκτελέσιμη έκδοση

Στη τελική εκτελέσιμη έκδοση της εργασίας διακρίνουμε τις εξής λειτουργίες:

- Σύνδεση με τον λογαριασμό Google του χρήστη.
- Επαλήθευση στοιχείων με δακτυλικό αποτύπωμα του χρήστη.
- Επικοινωνία με γιατρό οποιαδήποτε στιγμή μέσω chat.
- Ειδοποίηση χρήστη με alerts ακόμα και όταν η εφαρμογή είναι κλειστή.
- Σελίδες βοήθειας για την επεξήγηση λειτουργιών.
- Ρυθμίσεις εφαρμογής για τις απαιτήσεις των χρηστών.
- Μετάφραση εφαρμογής σε γλώσσες: Ελληνικά, Αγγλικά.
- Ρύθμιση ταχύτητας με τη λειτουργία οδήγησης.
- Ιστορικών παραβιάσεων ταχύτητας με google maps.
- Κουμπί έκτακτης ανάγκης σε περίπτωση κινδύνου.
- Επικοινωνία με συγγενικά πρόσωπα και υπηρεσίες για την ασφάλεια του χρήστη.
- Ειδοποίηση τρίτων σε περίπτωση απώλειας ισορροπίας του χρήστη.
- Δόνηση για μηνύματα γιατρού.
- Προσδιορισμός τηλεφώνου έκτακτης ανάγκης για κλήσεις και μηνύματα.

Τεχνολογίες και Sensors:

- Google authentication για τη σύνδεση του χρήστη.
- Firebase Realtime Database για τη λειτουργία γιατρού και ειδοποιήσεων.
- SQLite database για τη λειτουργία οδήγησης.
- SharedPreferences για τις ρυθμίσεις της εφαρμογής.
- Fingerprint Sensor για την επαλήθευση των στοιχείων του χρήστη.
- Location Sensor για τη λειτουργία οδήγησης σχετικά με την ταχύτητα και τη τοποθεσία.
- Acceleration Sensor για τη λειτουργία ειδοποίησης σε απώλεια ισορροπίας.

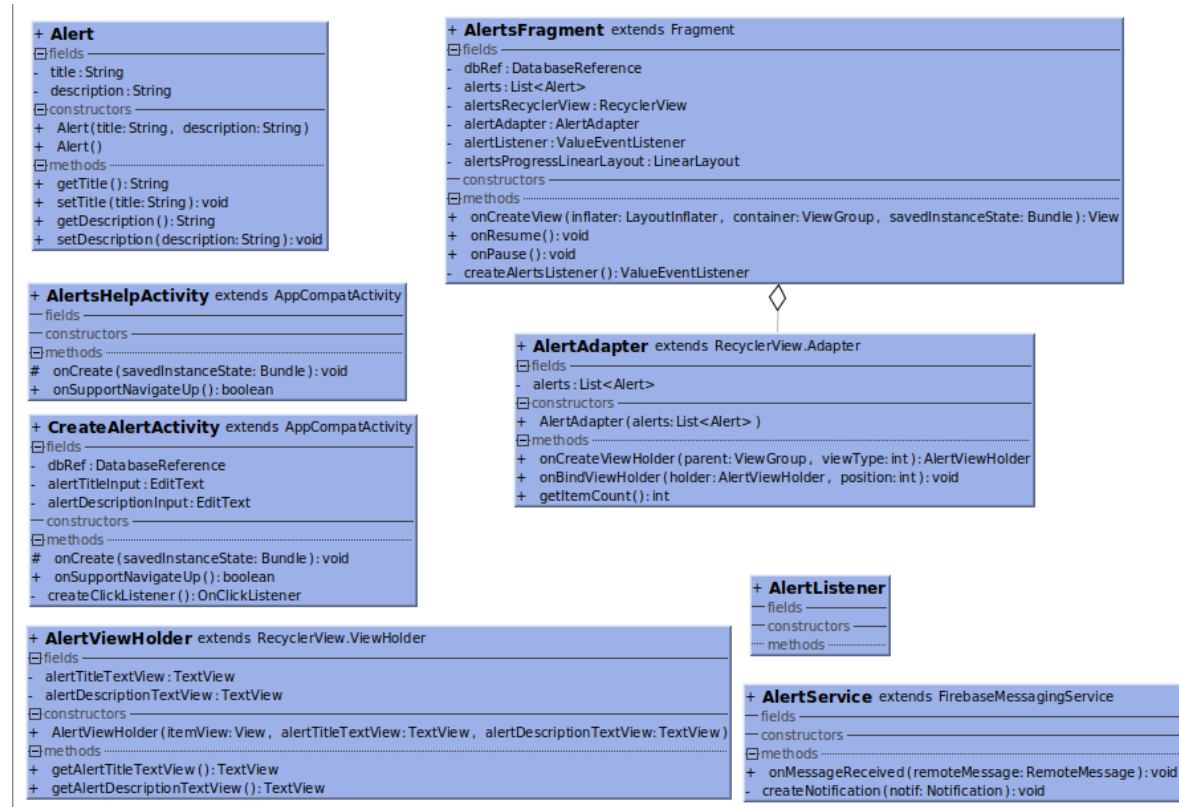
5.2.2 Αναφορά ελέγχου για την τελική εκτελέσιμη έκδοση

Στη τελική έκδοση της εφαρμογής αντιμετωπίσαμε τεχνικές δυσκολίες και ζητήματα που αφορούσαν τόσο το λειτουργικό σύστημα Android, όσο και services της Google. Κληθήκαμε να δημιουργήσουμε βάσεις δεδομένων για την αποθήκευση στοιχείων του χρήστη αλλά και να επεξεργαστούμε με σωστό τρόπο δεδομένα από σένσορες.

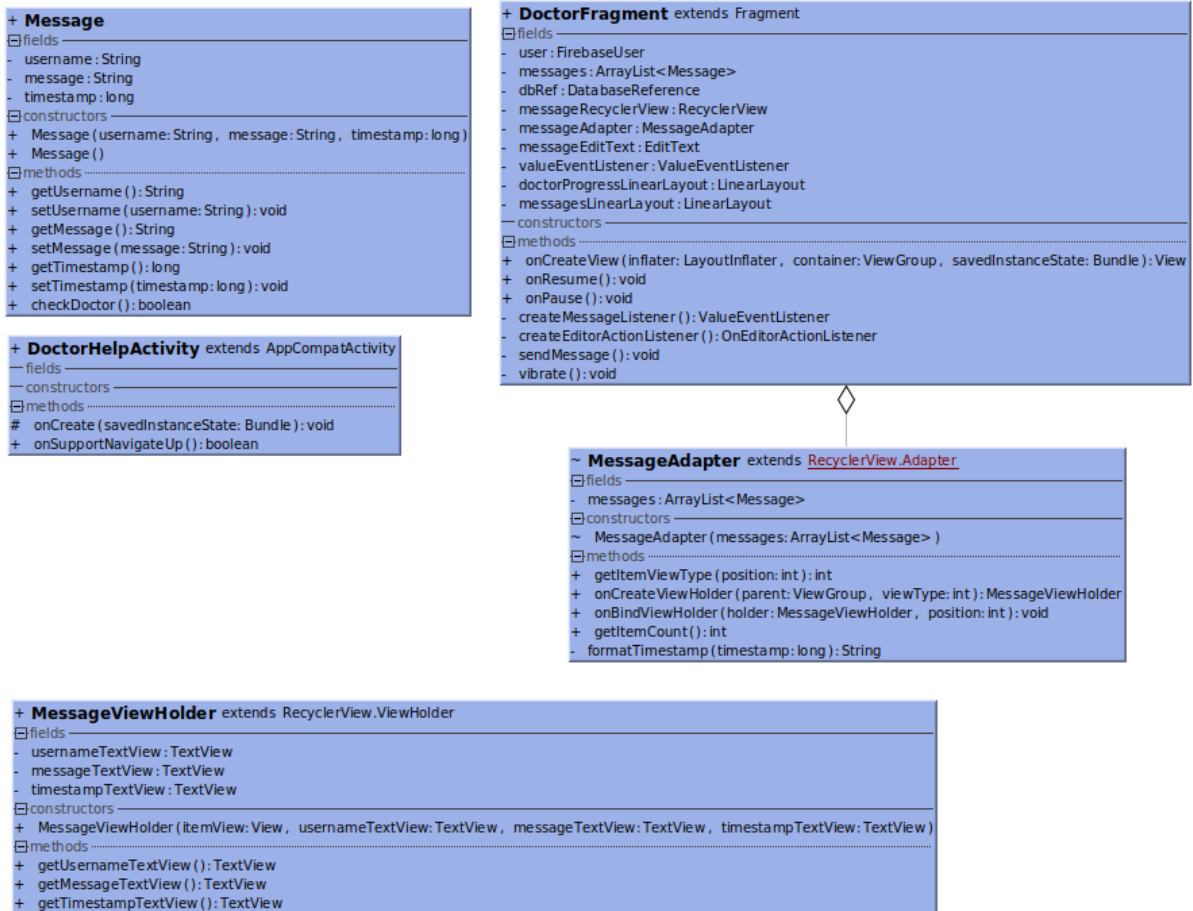
Μετά από αρκετές δοκιμαστικές λειτουργίες, καταλήξαμε σε μια λειτουργική έκδοση, η οποία είναι και αυτή που παρουσιάζεται σε αυτό το στάδιο.

6 Διαγράμματα κλάσεων

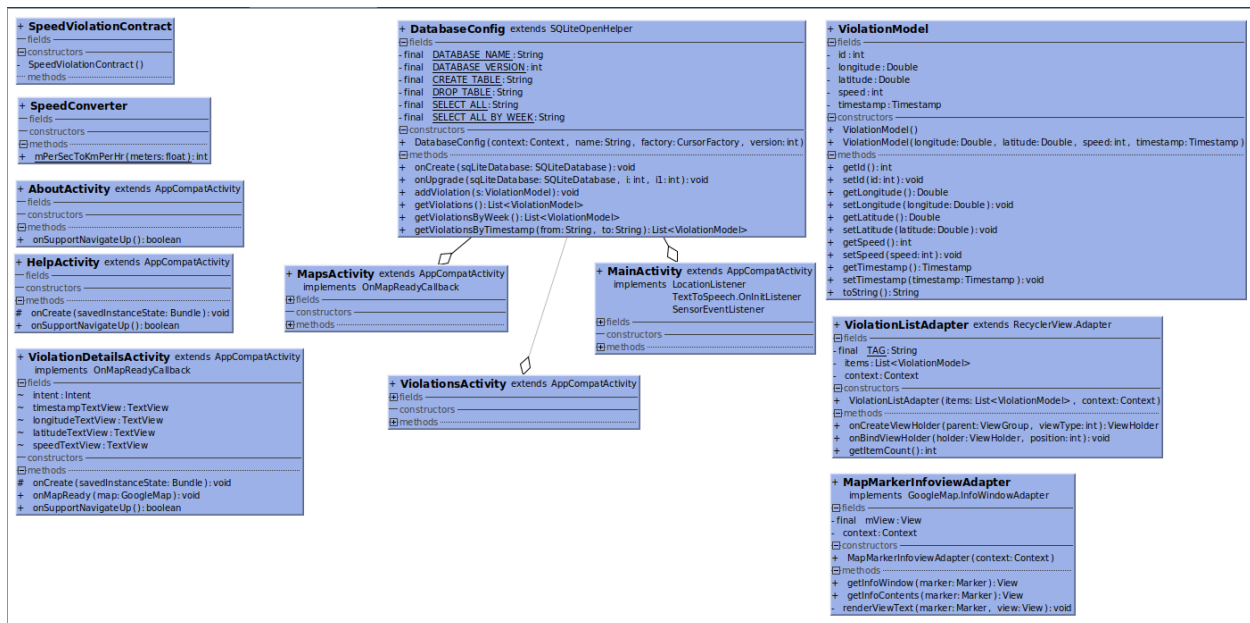
6.1 Διάγραμμα κλάσεων Alert



6.2 Διάγραμμα κλάσεων Doctor



6.3 Διάγραμμα κλάσεων Drive



6.4 Διάγραμμα κλάσεων Disability

```

+ PanicActivity extends AppCompatActivity
    implements TextToSpeech.OnInitListener
               SensorEventListener

fields -----
- emergencyPhone : String
- message : String
- emergencyEmail : String
- messagesButton : Button
- hospitalButton : Button
- callButton : Button
- speechRecognitionButton : FloatingActionButton
- intent : Intent
- actionBar : ActionBar
- sharedPreferences : SharedPreferences
- locale : Locale
- sensorManager : SensorManager
- accelerometer : Sensor
- textToSpeech : TextToSpeech
- speechRecognizer : SpeechRecognizer
- final SHAKE_THRESHOLD : float
- final MIN_TIME_BETWEEN_SHAKES_MILLISECS : int
- mLastShakeTime : long
constructors -----
methods -----
# onCreate (savedInstanceState: Bundle) : void
# onResume() : void
# onRestart() : void
# onDestroy() : void
+ onRequestPermissionsResult (requestCode: int, permissions: String[], grantResults: int[]): void
- initializeSpeechRecognition() : void
- processSpeechResult (command: String) : void
+ onSensorChanged (sensorEvent: SensorEvent) : void
+ onCreateOptionsMenu (menu: Menu) : boolean
+ onOptionsItemSelected (item: MenuItem) : boolean
+ onAccuracyChanged (sensor: Sensor, i: int) : void
+ checkPermission (permission: String) : int
+ onSupportNavigateUp() : boolean
- getPreferences() : void
~ checkLocale (lang: String) : void
# onActivityResult (requestCode: int, resultCode: int, data: Intent) : void
- sendMessages() : void
+ onInit (status: int) : void
+ onConfigurationChanged (newConfig: Configuration) : void

```

```

+ HelpActivity extends AppCompatActivity
fields -----
constructors -----
methods -----
# onCreate (savedInstanceState: Bundle) : void
+ onSupportNavigateUp() : boolean

```

6.5 Διάγραμμα κλάσεων Home

