



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Αθήνας  
Σχολή Τεχνολογικών Εφαρμογών  
Τμήμα Μηχανικών Πληροφορικής Τ.Ε.

## Πτυχιακή Εργασία

«Ηλεκτρονικός φάκελος ασθενών σε περιβάλλον Android»

Εμμανουήλ Παπαδημητρίου  
ΑΜ: 131004

Επιβλέπουσα Καθηγήτρια:  
Ελένη Γαλιώτου

Αθήνα, Ιανουάριος 2018

## Περίληψη

Η παρούσα πτυχιακή εργασία αφορά την ανάπτυξη μιας εφαρμογής κινητών συσκευών σε περιβάλλον Android με στόχο την υλοποίηση ηλεκτρονικού ιατρικού φακέλου των ασθενών στα πλαίσια του περιβάλλοντος του νοσοκομείου. Η εφαρμογή αυτή δίνει την δυνατότητα στους χρήστες (γιατροί, νοσοκομειακό προσωπικό κτλ.), να προβάλλουν, να διαχειρίζονται και να επεξεργάζονται απαραίτητες πληροφορίες που αφορούν τον κάθε ασθενή. Ειδικότερα, θα υπάρχει η δυνατότητα προβολής λεπτομερών στοιχείων του ασθενή, διαχείριση φαρμάκων, εξετάσεων και εισαγωγή σημειώσεων για παρατηρήσεις ή οτιδήποτε άλλο κρίνει ο χρήστης. Επίσης, είναι διαθέσιμη η προβολή ιστορικού για όλες τις αλλαγές που έγιναν στον ηλεκτρονικό φάκελο του ασθενή. Τέλος, δίνεται η δυνατότητα επικοινωνίας μεταξύ των χρηστών με άμεση ανταλλαγή μηνυμάτων.

Αρχικά θα γίνει αναλυτική περιγραφή του προβλήματος. Στην συνέχεια, θα αναλυθεί το λειτουργικό σύστημα Android και θα γίνει αναφορά στις νέες εκδόσεις. Συνεχίζοντας, θα περιγραφεί η γλώσσα Java και η νέα έκδοση της (Java 8) που προσφέρει νέες δυνατότητες αλλά και η γλώσσα προγραμματισμού Kotlin που είναι μια ανερχόμενη γλώσσα επιλογής για το Android. Στο δεύτερο μέρος, παρουσιάζεται η δομή που έχει ένα android project και θα αναλυθούν τα αρχεία και οι φάκελοι του. Σε επόμενο μέρος, θα αναλυθεί το Android Studio και οι emulators. Έπειτα στο τρίτο μέρος, θα γίνει αναλυτική περιγραφή στο Firebase που είναι μια NoSQL βάση και θα παρουσιαστούν οι λόγοι της επιλογής της, τα πλεονεκτήματα που έχει με τις Android εφαρμογές και θα γίνει πλήρης περιγραφή της σύνδεσης Firebase με το Android Studio.

Στο τέταρτο μέρος, παρουσιάζεται η εφαρμογή αναλυτικά, δείχνοντας την βάση στο firebase και την δομή της, ανάλυση εφαρμογής (κεντρική οθόνη, σύνδεση στο σύστημα, fragments με τα στοιχεία και με εικόνες παραδειγμάτων, λεκτική περιγραφή). Πλήρης ανάλυση των μερών της εφαρμογής και τα σημαντικά αρχεία.

Τελειώνοντας, θα γίνει σύγκριση με άλλες παρόμοιες εφαρμογές/πληροφοριακά συστήματα και παραθέτονται εντυπώσεις και προτάσεις για πιθανή βελτίωση της εφαρμογής και εξέλιξη της.

## Abstract

This diploma thesis discusses the development of a mobile application in an Android environment with the aim of implementing an electronic medical file for patients within the hospital environment. This application enables users (doctors, hospital staff, etc.) to view, manage and process necessary patient information. In particular, the user will be able to view patient details, manage medications, examine and insert notes for comments or whatever the user deems fit. Also, the history view is available for all changes made to the patient's electronic folder. Finally, users can communicate with each other by instant messaging.

First, a detailed description of the problem will be made. The Android operating system will then be analyzed and new versions will be referenced. Continuing, the Java language and its new version (Java 8) offer new features and the programming language Kotlin, which is an upcoming choice language for Android. In the second part, the structure of an android project is presented and its files and folders will be analyzed. In the next part, we analyze Android Studio and emulators. Then in the third part, a detailed description will be made about Firebase which is a NoSQL database and the reasons that it was selected will be presented as well, its advantages with Android apps and a full description of the Firebase connection with Android Studio.

In the fourth part, the application is presented in detail, showing the database scheme in the firebase and its structure, application analysis (central screen, connection to the system, data fragments and examples of images, verbal description). Full analysis of the parts of the application and important records.

Finally, other similar applications / information systems will be compared and impressions and suggestions will be given for possible improvement of the application and its evolution.

# Περιεχόμενα

Περίληψη.....	2
Abstract.....	3
Κεφάλαιο 1: Εισαγωγή.....	7
1.1. Παρουσίαση Εφαρμογής “HealthApp”.....	7
1.1.1. Σκοπός Εφαρμογής.....	7
1.1.2. Συνοπτική παρουσίαση εφαρμογής.....	8
1.2. Γνωριμία με το Android.....	12
1.2.1. Λίγα λόγια για το Android.....	12
1.2.2. Γλώσσες Προγραμματισμού.....	13
1.3. Γνωριμία με το Firebase.....	13
Κεφάλαιο 2: Λειτουργικό Σύστημα Android.....	14
2.1. Παρουσίαση του Android.....	14
2.1.1. Αναλυτική περιγραφή.....	14
2.1.2. Υλικό.....	15
2.1.3. Χαρακτηριστικά.....	15
2.1.3. Ασφάλεια και εμπιστευτικότητα.....	16
2.1.4. Open-Source κοινότητα.....	19
2.2. Τεχνικά χαρακτηριστικά.....	20
2.2.1. Linux kernel.....	20
2.2.1. Ανάλυση λειτουργικού συστήματος.....	21
2.3. Εκδόσεις Android.....	22
2.3.1. Android 1.0.....	22
2.3.2. Android 1.1.....	23
2.3.3. Android 1.5 (Cupcake).....	23
2.3.4. Android 1.6 (Donut).....	23
2.3.5. Android 2.0 – 2.1 (Eclair).....	23
2.3.6. Android 2.2 – 2.2.3 (Froyo).....	24
2.3.7. Android 2.3 – 2.3.7 (Gingerbread).....	24
2.3.8. Android 3.0 – 3.2.6 (Honeycomb).....	25
2.3.9. Android 4.0 – 4.0.4 (Ice Cream Sandwich).....	25
2.3.10. Android 4.1 – 4.3.1 (Jelly Bean).....	26
2.3.11. Android 4.4 – 4.4.4 (KitKat).....	27
2.3.12. Android 5.0 – 5.1.1 (Lollipop).....	27
2.3.13. Android 6.0 – 6.0.1 (Marshmallow).....	28
2.3.14. Android 7.0 – 7.1.2 (Nougat).....	28
2.3.15. Android 8.0 – 8.1 (Oreo).....	28
2.4. Δομή Εφαρμογής.....	29
2.4.1. Βασικές αρχές εφαρμογής.....	29
2.4.2. Στοιχεία εφαρμογής.....	29
2.4.2.1. Activities.....	30
2.4.2.2. Services.....	31
2.4.2.3. Broadcast Receivers.....	31
2.4.2.4. Content Providers.....	32
2.4.2.5. Fragments.....	32
2.4.2.6. Διαδικασίες.....	32
2.4.2.7. Threads.....	32
2.4.3. Αρχείο manifest.....	33
2.4.4. Πόροι εφαρμογής.....	33

2.4.4.1. Layout.....	33
2.4.4.2. Drawable.....	34
2.4.4.3. Values.....	34
<b>2.5. Android Studio.....</b>	<b>34</b>
2.5.1. Περιγραφή Android Studio.....	34
2.5.2. Emulators.....	38
2.5.2.1. AVD.....	38
2.5.2.2. Genymotion.....	39
2.5.2.3. Σύγκριση AVD και Genymotion.....	39
<b>Κεφάλαιο 3: Firebase.....</b>	<b>40</b>
<b>3.1. NoSQL Βάσεις Δεδομένων.....</b>	<b>40</b>
<b>3.2. Αναλυτική περιγραφή του Firebase.....</b>	<b>42</b>
3.2.1. Αυθεντικοποίηση.....	43
3.2.2. Βάση δεδομένων πραγματικού χρόνου.....	44
3.2.3. Αποθηκευτικός χώρος.....	46
3.2.4. Λόγος επιλογής.....	47
3.2.5. Χρήση του Firebase.....	47
3.2.5.1. Κονσόλα Firebase.....	47
3.2.5.1.1. Δημιουργία project.....	47
3.2.5.1.1. Αυθεντικοποίηση χρήστη.....	48
3.2.5.1.2. Δημιουργία βάσης δεδομένων.....	48
3.2.5.2. Σύνδεση με το Android Studio.....	49
<b>3.3. Σύγκριση με άλλες βάσεις δεδομένων.....</b>	<b>49</b>
3.3.1. Σύγκριση με MySQL.....	49
3.3.2. Σύγκριση με SQLite.....	50
3.3.3. Σύγκριση με Couchbase.....	50
3.3.3. Σύγκριση με MongoDB.....	50
<b>Κεφάλαιο 4: Υλοποίηση Εφαρμογής HealthApp.....</b>	<b>51</b>
<b>4.1. Λεκτική παρουσίαση.....</b>	<b>51</b>
<b>4.2. Πλατφόρμα Firebase.....</b>	<b>52</b>
4.2.1. Αυθεντικοποίηση χρηστών.....	52
4.2.2. Βάση δεδομένων.....	53
4.2.3. Αποθηκευτικός χώρος.....	59
<b>4.3. Παρουσίαση εφαρμογής.....</b>	<b>61</b>
4.3.1. Φόρτωση εφαρμογής.....	65
4.3.2. Σύνδεση χρήστη.....	66
4.3.3. Navigation Drawer.....	68
4.3.4. PatientsActivity.....	71
4.3.5. Προφίλ.....	73
4.3.6. Εξετάσεις.....	74
4.3.7. Φάρμακα.....	79
4.3.8. Σημειώσεις.....	82
4.3.9. Ιστορικό.....	85
4.3.10. Ανταλλαγή μηνυμάτων.....	87
<b>Κεφάλαιο 5: Παρόμοιες εφαρμογές.....</b>	<b>90</b>
<b>5.1. MediFile EMR.....</b>	<b>90</b>
<b>5.2. ΗΦΑ.....</b>	<b>90</b>
<b>Κεφάλαιο 6: Συμπεράσματα-Προτάσεις.....</b>	<b>91</b>
<b>6.1. Συμπεράσματα.....</b>	<b>91</b>
<b>6.2. Προτάσεις.....</b>	<b>91</b>

Κεφάλαιο 7: Βιβλιογραφία.....92

Κεφάλαιο 8: Παράρτημα.....96

# Κεφάλαιο 1: Εισαγωγή

## 1.1. Παρουσίαση Εφαρμογής “HealthApp”

### 1.1.1. Σκοπός Εφαρμογής

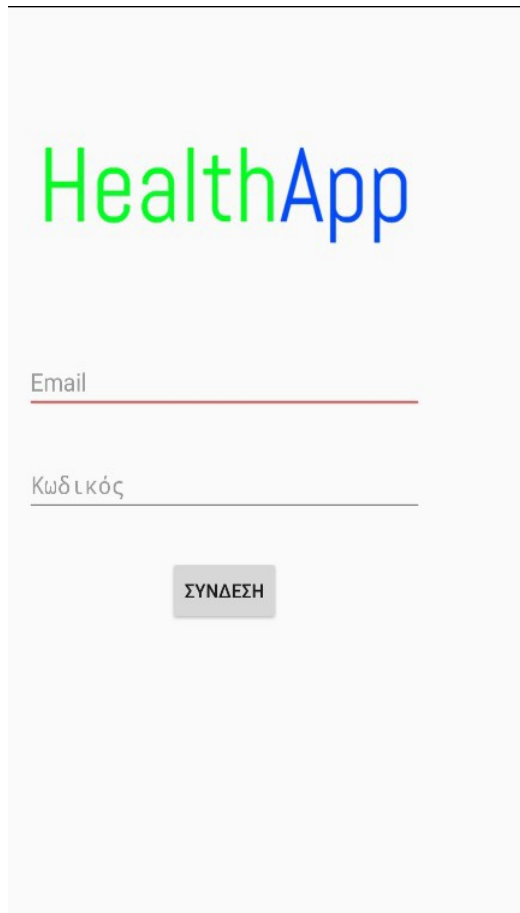
Η εφαρμογή “HealthApp” έχει ως σκοπό την άμεση διευκόλυνση των χρηστών (γιατρών και νοσοκομειακού προσωπικού) στο περιβάλλον ενός νοσοκομείου ως προς την διαχείριση των ασθενών τους. Δίνεται η δυνατότητα στους χρήστες να μπορούν να προβάλουν και να επεξεργαστούν ανά πάσα στιγμή οποιαδήποτε πληροφορία αφορά τον ασθενή τους. Στην ουσία, η εφαρμογή “HealthApp” είναι ένα σύστημα που περιέχει όλους τους ιατρικούς φακέλους των ασθενών που έχουν ανατεθεί στον κάθε χρήστη με τις παρακάτω δυνατότητες:

- Προβολή όλων των ασθενών ονομαστικά.
- Επιλογή ενός ασθενούς για προβολή ηλεκτρονικού ιατρικού φακέλου.
- Παρουσίαση βασικών στοιχείων του ασθενή (όνομα, φύλο, ημ. γέννησης κτλ.).
- Προβολή και προσθήκη φαρμάκων.
- Προβολή πληροφοριών εξετάσεων και των αποτελεσμάτων τους αν υπάρχουν, και προσθήκη νέων εξετάσεων.
- Εισαγωγή σημειώσεων – δυνατότητα στον χρήστη να εισάγει οτιδήποτε κρίνει ότι πρέπει να σημειωθεί για τον ασθενή.
- Δυνατότητα προβολής ιστορικού για όλες τις προσθαφαίρεσεις που έγιναν στα παραπάνω.
- Δυνατότητα ανταλλαγής άμεσων μηνυμάτων με τους χρήστες στα πλαίσια του ηλεκτρονικού ιατρικού φακέλου κάθε ασθενούς

Είναι σημαντικό να τονιστεί ότι η εφαρμογή αυτή δεν λειτουργεί μονομερώς, αλλά οι αλλαγές που γίνονται από τον κάθε χρήστη προβάλλονται και στους υπόλοιπους, για αυτό έχει ιδιαίτερη σημασία το ιστορικό. Ειδικότερα, ένας ασθενής μπορεί να ανήκει σε περισσότερους από έναν χρήστες και οι χρήστες μπορούν να προβάλουν και να επεξεργάζονται τον ηλεκτρονικό φάκελο του ασθενούς μόνο όταν έχει ανατεθεί από τον διαχειριστή του συστήματος ο ασθενής στον χρήστη. Δεν μπορεί δηλαδή, ένας χρήστης να έχει πρόσβαση στον ηλεκτρονικό φάκελο ενός ασθενούς που δεν του έχει ανατεθεί.

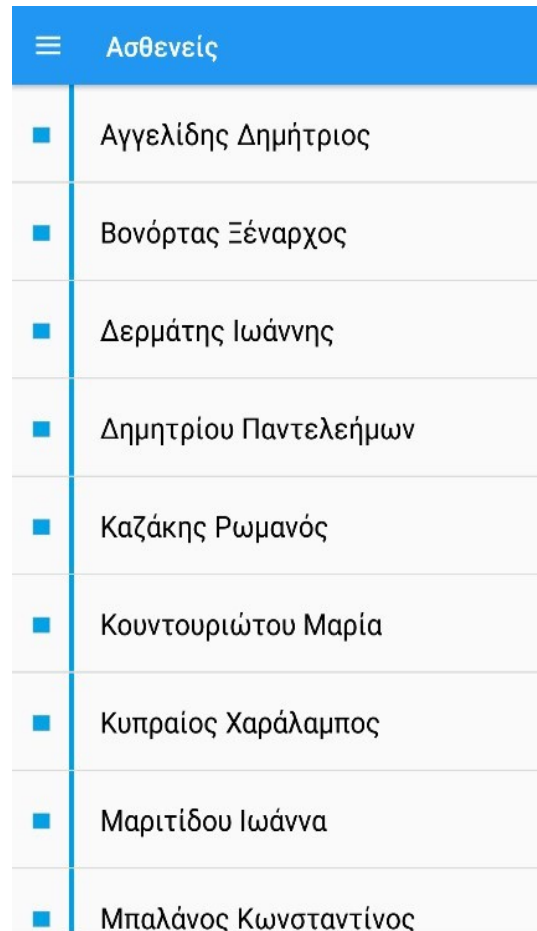
Ιδιαίτερη μνεία γίνεται στο ιατρικό απόρρητο, καθώς η εφαρμογή προφυλάσσεται από την αθέμιτη χρήση εξωτερικών χρηστών. Συγκεκριμένα, δεν υπάρχει η δυνατότητα εγγραφής καθώς τα στοιχεία σύνδεσης αναθέτονται στους χρήστες από τον διαχειριστή. Επίσης, τα στοιχεία των ασθενών δεν είναι διαθέσιμα πουθενά αλλού παρα μόνο στα πλαίσια της εφαρμογής και στο ασφαλές σύστημα της βάσης δεδομένων της πλατφόρμας Firebase.

### 1.1.2. Συνοπτική παρουσίαση εφαρμογής



The image shows a login screen for an application named "HealthApp". The title "HealthApp" is displayed in a large, green and blue font. Below the title, there are two input fields: "Email" and "Κωδικός" (Password). A red underline is visible under the "Email" field. Below the input fields, there is a grey button with the text "ΣΥΝΔΕΣΗ" (Login).

Εικόνα 1.1: Σύνδεση χρήστη



Ασθενείς	
<input type="checkbox"/>	Αγγελίδης Δημήτριος
<input type="checkbox"/>	Βονόρτας Ξέναρχος
<input type="checkbox"/>	Δερμάτης Ιωάννης
<input type="checkbox"/>	Δημητρίου Παντελεήμων
<input type="checkbox"/>	Καζάκης Ρωμανός
<input type="checkbox"/>	Κουντουριώτου Μαρία
<input type="checkbox"/>	Κυπραίος Χαράλαμπος
<input type="checkbox"/>	Μαριτίδου Ιωάννα
<input type="checkbox"/>	Μπαλάνος Κωνσταντίνος

Εικόνα 1.2: Λίστα Ασθενών



ΠΡΟΦΙΛ		ΕΞΕΤΑΣΕΙΣ		ΦΑΡΜΑΚΑ		ΣΗΜΕΙΩΣΕΙΣ		ΙΣΤΟ	
Όνομα	Δημήτριος								
Επώνυμο	Αγγελίδης								
Πατρώνυμο	Γεώργιος								
Φύλο	Άρρεν								
Η.Μ. Γέννησης	12/5/1971								
Εισαγωγή	2/11/2017								
Εθνικότητα	Ελληνική								
Πόλη Διαμονής	Αθήνα								
Διεύθυνση	Αιόλου 32								
T.K.	10437								
Τηλέφωνο 1	6973335517								
Τηλέφωνο 2	2102735650								

ΠΡΟΦΙΛ

ΕΞΕΤΑΣΕΙΣ

ΦΑΡΜΑΚΑ

ΣΗΜΕΙΩΣΕΙΣ


ΙΣΤΟ

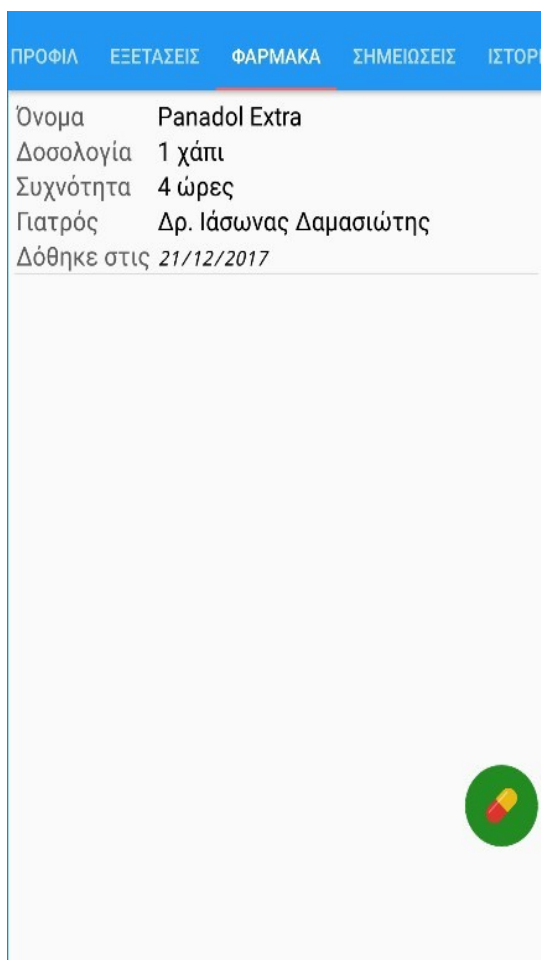
Όνομα

Ακτινογραφία γνάθου κάτω

Ημερομηνία

23/12/2017

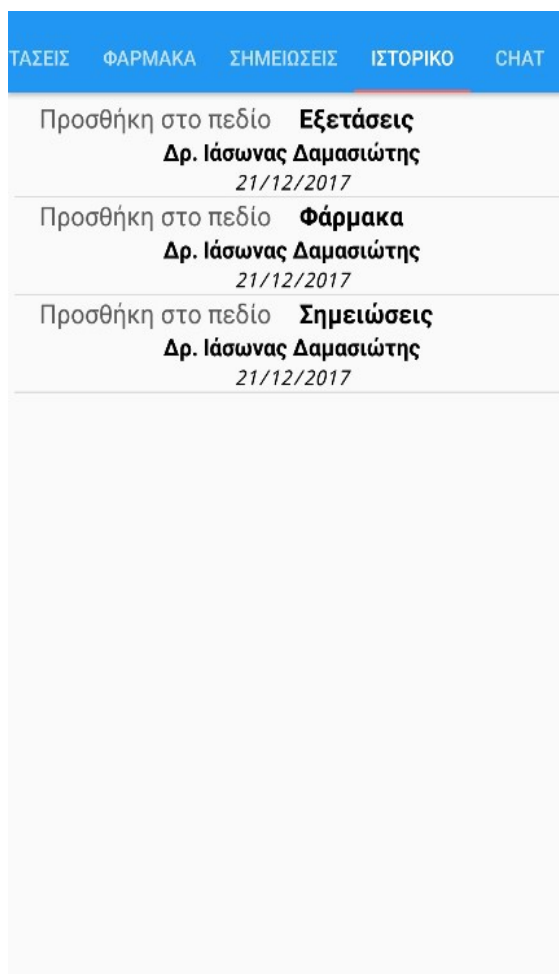




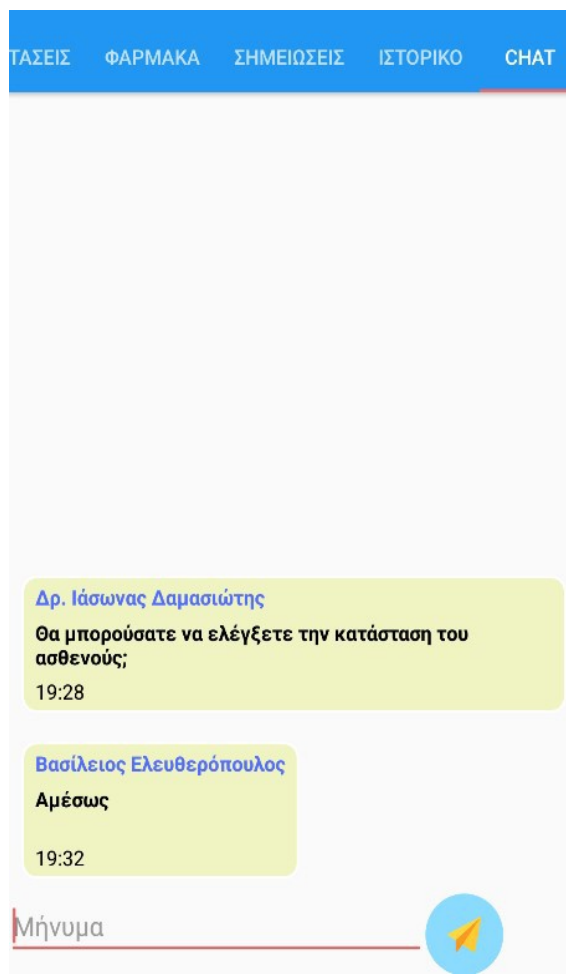
Εικόνα 1.5: Φάρμακα ασθενούς



Εικόνα 1.6: Σημειώσεις ασθενούς



Εικόνα 1.7: Ιστορικό αλλαγών



Εικόνα 1.8: Επικοινωνία μεταξύ χρηστών

## 1.2. Γνωριμία με το Android

### 1.2.1. Λίγα λόγια για το Android

Το Android είναι ένα λειτουργικό σύστημα που το συναντάμε κυρίως σε φορητές συσκευές (smartphones, tablets, ρολόγια χειρός) αλλά και στις τηλεοράσεις (android tv) και στα αυτοκίνητα (android auto) [1]. Είναι ένα από τα πιο διαδεδομένα λειτουργικά συστήματα, καθώς χρησιμοποιείται σε εκατομμύρια συσκευές στον κόσμο. Η πρώτη επίσημη έκδοση (1.0) κυκλοφόρησε το 2008 [2], χωρίς να έχει κάποιο συγκεκριμένο όνομα όπως οι επόμενες εκδόσεις με πρώτη την Cupcake (1.5) το 2009 [3] μέχρι και την τελευταία έκδοση Oreo (8.1.0). Χαρακτηριστικό του Android είναι η πολυπύρηνη επεξεργασία δεδομένων με την χρήση οθονών αφής, αισθητήρων και αναδυόμενων τεχνολογιών που αφορούν τις κινητές συσκευές.



*Εικόνα 1.9: Λογότυπο Android*

Το Android είναι βασισμένο στον πυρήνα (kernel) του Linux το οποίο είναι open-source. Αυτό οδήγησε στο να βοηθήσουν την δημιουργία και την συνεχή εξέλιξη του εκατοντάδες προγραμματιστές, ειδικοί του hardware κτλ, με πρώτη την Android Inc. που εξαγοράστηκε από την Google το 2005 [4]. Το Android δίνει την δυνατότητα δημιουργίας εφαρμογών που ικανοποιεί κάθε ανάγκη όπως:

- Επικοινωνία
- Ψυχαγωγία (μουσική, παιχνίδια)
- Ενημέρωση
- Καλύτερευση του βίου (χάρτες και πλοήγηση, εφαρμογές υγείας)

Το Android συνεχίζει να εξελίσσεται με ραγδαίους ρυθμούς ώστε να μπορεί να δώσει νέες εμπειρίες στους χρήστες αλλά και περισσότερες δυνατότητες στους δημιουργούς.

### 1.2.2. Γλώσσες Προγραμματισμού

Το Android επιτρέπει στους προγραμματιστές και στους δημιουργούς να υλοποιούν τις εφαρμογές χρησιμοποιώντας κατά κύριο λόγο την αντικειμενοστρεφής γλώσσα προγραμματισμού Java με την χρήση των απαραίτητων βιβλιοθηκών που παρέχει η Google.

Όμως, η Kotlin που είναι μια στατική γλώσσα προγραμματισμού που μπορεί να χρησιμοποιηθεί και ως αντικειμενοστρεφής αλλά και ως λειτουργική. Είναι μια ανερχόμενη γλώσσα που τρέχει στο Java Virtual Machine (JVM) και μπορεί να μεταγλωττιστεί σε JavaScript. Λειτουργεί μαζί με τον κώδικα Java και είναι εξαρτημένη από υπάρχουσες βιβλιοθήκες σε Java. Το Android Studio 3.0 υποστήριζει πλήρως Kotlin.[\[5\]](#)

### 1.3. Γνωριμία με το Firebase

Το Firebase είναι μια πλατφόρμα ανάπτυξης κινητών και διαδικτυακών εφαρμογών που αναπτύχθηκε το 2011 από την Firebase Inc. και εξαγοράστηκε από την Google το 2011 [\[6\]](#). Το Firebase προσφέρει υπηρεσίες όπως:

- Αναλυτικών στοιχείων (Analytics)
- Realtime Βάση Δεδομένων
- Αυτόματης εξουσιοδότησης χρηστών (auth)
- Αποθήκευσης διαφόρων ειδών αρχείων
- Φιλοξενία στατικών σελίδων του web (HTML,CSS,JavaScript) [\[7\]](#)

Είναι ουσιαστικά μια backend υπηρεσία που παρέχει εύκολη και άμεση πρόσβαση σε υπηρεσίες που διαφορετικά θα έπρεπε να γραφτούν APIs και να γίνει δημιουργία και διαχείριση των servers. Ελευθερώνει τους προγραμματιστές στο να επικεντρωθούν στην υλοποίηση της εφαρμογής. Το Firebase είναι μια NoSQL cloud βάση δεδομένων ιδανική για συγχρονισμό και αποθήκευση δεδομένων. Δεν υπάρχει δηλαδή, η ανάγκη ενός application server καθώς ο χρήστης μπορεί να έχει πρόσβαση στα δεδομένα από την φορητή συσκευή ή μέσω του web browser. [\[8\]](#)

Το Firebase παρέχει ρυθμίσεις για την ασφάλεια των στοιχείων που δίνουν ευελιξία στον χρήστη να προσαρμόσει την εφαρμογή του με την βάση δεδομένων. Κλείνοντας, οι εφαρμογές του Firebase παραμένουν responsive ακόμα και όταν δεν υπάρχει σύνδεση στο internet, και αυτό είναι ένα πολύ μεγάλο πλεονέκτημα έναντι στην custom δημιουργία server καθώς το σύνηθες είναι οι προγραμματιστές να απαγορεύουν την χρήση μιας online εφαρμογής όταν δεν υπάρχει σύνδεση. Το Firebase δίνει την απόλυτη ελευθερία στον χρήστη και στον δημιουργό.

## Κεφάλαιο 2: Λειτουργικό Σύστημα Android

### 2.1. Παρουσίαση του Android

#### 2.1.1. Αναλυτική περιγραφή

Όπως είναι γνωστό, το Android ανήκει στην Google και ο κόσμος άρχισε να μαθαίνει για το Android λειτουργικό σύστημα στα τέλη της προηγούμενης δεκαετίας. Όμως, το 2003 ο Andy Rubin είπε “Ο στόχος του Android είναι να επιτρέψει στις έξυπνες κινητές συσκευές να γνωρίζουν καλύτερα την τοποθεσία του ιδιοκτήτη και τις προτιμήσεις του”. Έτσι, μαζί με τον Rich Miner, Nick Sears και Chris White ιδρύσανε την Android Inc τον Οκτώβριο του 2003.<sup>[3]</sup> Το 2005, όταν ακόμα η Google αποκαλούνταν “γίγαντας της αναζήτησης”, ήθελε να κάνει το βήμα παραπάνω στην ασύρματη τεχνολογία και έτσι απέκτησε την Android Inc. Η πρώτη επίσημη και σταθερή έκδοση του Android η 1.0 ανακοινώθηκε στις 23 Σεπτεμβρίου 2008, με το πρώτο smartphone διαθέσιμο για το κοινό να είναι το HTC Dream <sup>[9]</sup>. Έπειτα, συνεχίστηκαν οι ενημερώσεις και οι διορθώσεις προβλημάτων αλλά και οι νέες εκδόσεις μέχρι την πιο πρόσφατη την Android 8.1.0 με το χαρακτηριστικό όνομα Oreo όπως συνηθίζεται να δίνεται το όνομα κάποιου γλυκίσματος σε κάθε νέα έκδοση.

Το Android είναι ένα λειτουργικό σύστημα που είναι σχεδιασμένο για smartphones, tablets, έξυπνα ρολόγια και ακόμα αυτοκίνητα. Υποστηρίζει οθόνη αφής με την οποία ο χρήστης μπορεί άμεσα και εύκολα να πραγματοποιήσει επιθυμητές ενέργειες όπως:

- Τηλεφωνική κλήση
- Ανταλλαγή μηνυμάτων
- Χρησιμοποίηση εφαρμογών

Κύριο χαρακτηριστικό του λειτουργικού συστήματος είναι οι εφαρμογές (applications), οι οποίες διευρύνουν τις δυνατότητες των συσκευών και ικανοποιούν όλο και περισσότερες ανάγκες του χρήστη με το πέρασμα του χρόνου. Οι εφαρμογές έχουν κατάληξη “.apk” και μπορούν να βρεθούν στο Google Play, το οποίο είναι ένα ηλεκτρονικό κατάστημα που επιτρέπει την πρόσβαση σε εφαρμογές διαφόρων ειδών. Επίσης, οι εφαρμογές μπορούν να βρεθούν σε διάφορους ιστοτόπους ή σε άλλα ηλεκτρονικά καταστήματα όπως το .

Σημαντικό να τονιστεί ότι το Android είναι open source και η δομή του είναι πιο ελεύθερη συγκριτικά με άλλα λειτουργικά συστήματα, μπορεί δηλαδή ο χρήστης να προσαρμόσει το ίδιο το λειτουργικό σύστημα στα μέτρα του και στις ανάγκες του. Αυτό έχει οδηγήσει στην συνεχή εξέλιξη του ίδιου του Android και στην γέννηση νέων λειτουργικών συστημάτων βασισμένα στην δομή του Android. <sup>[10]</sup>

Το 2017 η Google ανακοίνωσε στο I/O developer conference, ότι το Android τρέχει σε περισσότερες από 2 δις ενεργές συσκευές. Ξεπέρασε δηλαδή τους 1.5 δις υπολογιστές που τρέχουν Windows παγκοσμίως. Αν δει κανείς ότι το 2009 ότι μόλις το 2.8% των smartphones έτρεχαν σε Android και το 2010 το 33% παγκοσμίως σε μια εποχή που ξεπέρασε το λειτουργικό σύστημα Symbian. <sup>[11]</sup>

Το Android παρέχει στους προγραμματιστές τα εφόδια για ανάπτυξη εφαρμογών. Μέσω της επίσημης ιστοσελίδας υπάρχουν οδηγοί εκμάθησης δημιουργίας εφαρμογών αλλά και αναλυτικοί οδηγοί για οποιαδήποτε δυνατότητα. Το κύριο περιβάλλον ανάπτυξης (IDE) είναι

το Android Studio, στο οποίο ο developer μπορεί να χρησιμοποιήσει την γλώσσα προγραμματισμού Java ώστε να γράψει την λογική της εφαρμογής του και με την χρήση της XML υλοποιεί το γραφικό περιβάλλον [12]. Ιδιαίτερο γνώρισμα της Java και κατ' επέκταση του Android, είναι η χρήση βιβλιοθηκών. Η Google παρέχει επίσημες βιβλιοθήκες στον developer για προσθήκη λειτουργιών. Τον Μάιο του 2017, ανακοινώθηκε ότι το Android υποστηρίζει και την γλώσσα προγραμματισμού Kotlin[5]. Τα εργαλεία της Kotlin , προστέθηκαν στην έκδοση 3.0 του Android Studio, την περίοδο δηλαδή που η Kotlin βγήκε από την beta έκδοση τον Νοέμβριο του 2017.

### 2.1.2. Υλικό

Το Android, καθώς είναι open source, δεν περιορίζεται σε μία και μόνο αρχιτεκτονική όπως συμβαίνει σε ανταγωνιστικά λειτουργικά συστήματα. Έτσι, υπάρχει η ανάγκη συμβατότητας του λειτουργικού με διάφορες αρχιτεκτονικές υλικού και επεξεργαστών. [13] Με την χρήση του NDK ο δημιουργός είναι σίγουρος ότι η μεταγλώττιση γίνεται για την σωστή αρχιτεκτονική, παρέχοντας του δυνατότητα επιλογής πολλών ABI(application binary interface). ABI είναι ο συνδυασμός του επεξεργαστή με τα σετ εντολών του. Το Android υποστηρίζει τα παρακάτω ABIs :

- armeabi
- armeabi-v7a
- arm64-v8a
- x86
- x86\_64
- mips
- mips64 [14]

Οι συσκευές Android περιέχουν υλικό όπως κάμερα, γυροσκόπιο, επιταχυνσιόμετρο, GPS, θερμομέτρο, μαγνητόμετρο, βαρόμετρο, αισθητήρα προσέγγισης κτλ. Κάποια από αυτά μπορούν να παραλειφθούν, αλλά υπάρχουν εφαρμογές που τα χρησιμοποιούν άρα όταν λείπουν από την συσκευή , η χρήση της εφαρμογής είναι αδύνατη.

Το Android 4.4 KitKat μπορεί να τρέξει σε συσκευές με RAM στα 512 MB αλλά προτείνεται πλέον πάνω από 2 GB RAM. Όσες περισσότερες εκδόσεις βγαίνουν, τόσο περισσότερη είναι η ανάγκη για καλύτερη τεχνολογία στις συσκευές.

Αν και το Android είναι διαδεδομένο κυρίως σε smartphones και tablets, είναι δυνατή η χρήση του και σε σταθερούς υπολογιστές όπως για παράδειγμα το λειτουργικό σύστημα Remix OS[15] το οποίο τρέχει σε οποιοδήποτε υπολογιστή με αρχιτεκτονική x86 και είναι βασισμένο στο Android.

### 2.1.3. Χαρακτηριστικά

Τα κύρια χαρακτηριστικά που συνθέτουν το λειτουργικό σύστημα του Android είναι:

- Υποστήριξη Ήχου
- Υποστήριξη Video
- Υποστήριξη Εικόνας
- Πρωτόκολλα Δικτύου [16]



Η μεμονωμένη χρήση αλλά και ο συνδυασμός των παραπάνω, μας δίνει το σύνολο του λειτουργικού συστήματος. Για παράδειγμα, η βιντεοκλήση είναι συνδυασμός ήχου, video και δικτύου. Σημαντικό χαρακτηριστικό που προστέθηκε το 2010 είναι το multitouch το οποίο το υλοποιήθηκε στο Android στο επίπεδο του kernel [17] καθώς η Apple το 2007 έκανε δημοφιλή την έννοια multitouch και με αυτόν τον τρόπο η Google ήθελε να αποφύγει προβλήματα ευρεσιτεχνίας [18].

Σημαντικό χαρακτηριστικό επίσης είναι η φωνητική αναγνώριση, καθώς γίνεται εφικτή η χρήση της συσκευής χωρίς τα δάχτυλα. Την λειτουργία αυτήν την συναντάμε κυρίως στην πραγματοποίηση τηλεφωνημάτων, πλοήγηση, αναζήτηση και φωνητική πληκτρολόγηση. Είναι από τα πιο σημαντικά χαρακτηριστικά γιατί βελτιώνει την ποιότητα ζωής των ανθρώπων αλλά και των ανθρώπων με προβλήματα όρασης και ακοής.

Το Android επιτρέπει στις εφαρμογές να έχουν πρόσβαση στις υπηρεσίες τοποθεσίας όπως οι εφαρμογές πλοήγησης και χαρτών. Με το Google Maps Android API, μπορεί ο προγραμματιστής να προσθέσει λειτουργίες χαρτών με βάση τα δεδομένα του Google Maps. Το API διαχειρίζεται αυτόματα την πρόσβαση στους servers του Google Maps, το κατέβασμα δεδομένων, την προβολή του χάρτη και τις χειρονομίες αφής στον χάρτη. Μπορεί επίσης να χρησιμοποιήσει το API για να βάλει σημάδια στους χάρτες ή να περιορίσει την προβολή σε ένα μια συγκεκριμένη περιοχή του χάρτη.[19]

Ο προγραμματιστής έχει την δυνατότητα να προσθέσει διαφόρων ειδών γραφικά μέσω των APIs στο GUI της εφαρμογής δημιουργώντας τα δικά του γραφικά είτε 2D είτε 3D όπως σχήματα, γραφήματα και τα οποία μπορεί να τα κάνει animated. Θέτει τις παραμέτρους στο XML αρχείο και επεξεργάζεται την λογική του γραφικού στο κώδικα Java.[20][21]

Υπάρχει η δυνατότητα αποθήκευσης δεδομένων της εφαρμογής με κάποιες από αυτές τις επιλογές να είναι:

- Αποθήκευση δεδομένων με την χρήση κλειδιών
- Εσωτερική Αποθήκευση
- Εξωτερική Αποθήκευση
- SQLite Βάσεις δεδομένων
- Αποθήκευση δεδομένων στον προσωπικό network server

### 2.1.3. Ασφάλεια και εμπιστευτικότητα

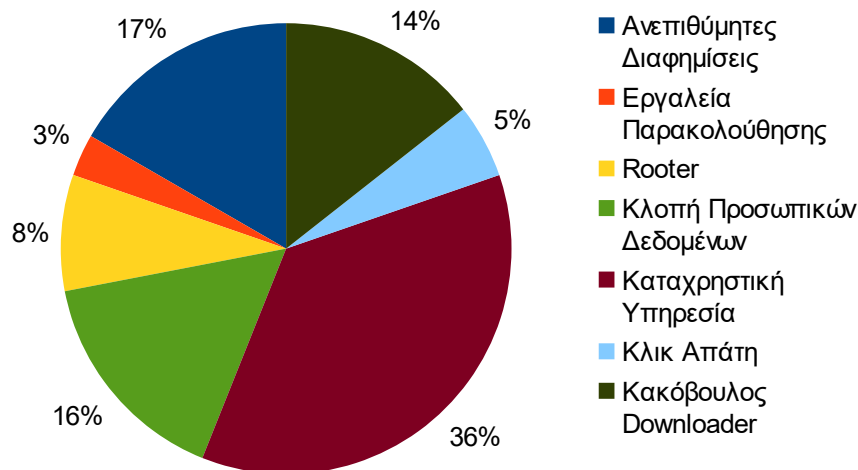
Με την αύξηση των συσκευών και των χρηστών, ήταν αναγκαία η αύξηση των μέτρων για την ασφάλεια. Τον Φεβρουάριο του 2012 ο Hiroshi Lockheimer, VP of Engineering του Android, ανακοίνωσε ότι στόχος εκτός από τα νέα χαρακτηριστικά και καινοτομίες εξακολουθεί να είναι και η ασφάλεια [22]. Το 2012 ήταν μια χρονιά κατά την οποία οι μολύνσεις των κινητών συσκευών με malwares αυξήθηκαν δραματικά, καθώς όλο και περισσότερες εφαρμογές έκαναν την εμφάνιση τους ακόμα και στο επίσημο κατάστημα του Google Play, όμως το ποσοστό αυτό είναι μικρό, μόλις στο 0.5% σύμφωνα με την έρευνα του F-Secure [23]. Σημαντικό να τονιστεί ότι εκείνη την περίοδο υπήρχαν διαμάχες μεταξύ της Google και διαφόρων εταιρειών για τις κατηγορίες περί malware, καθώς η Google θεωρεί ότι είναι πολύ μικρό το ποσοστό των επικίνδυνων malware.

Οι πιο διαδεδομένοι τύποι κακόβολου λογισμικού είναι:

- Ανεπιθύμητες Διαφημίσεις
- Εργαλεία παρακολούθησης



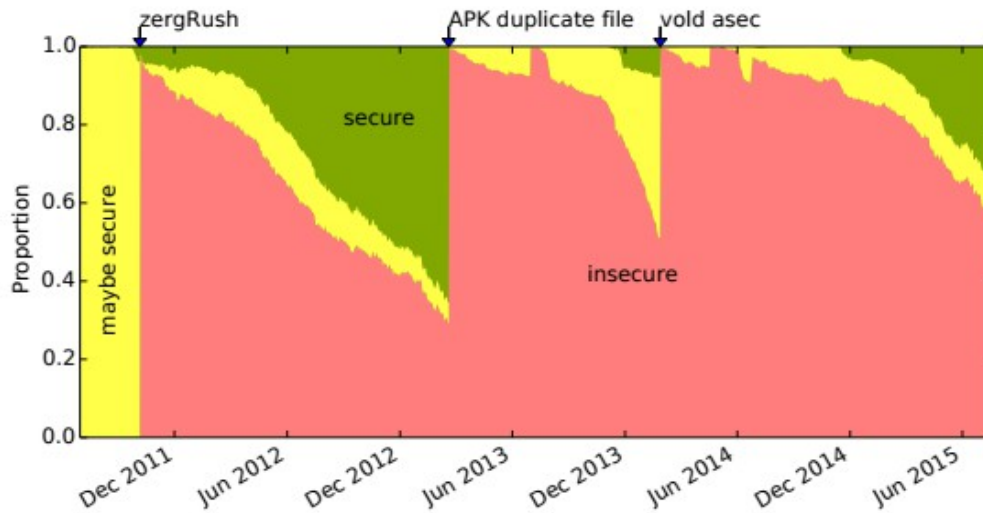
- Router – επιτρέπει την εκτέλεση εντολών στην μολυσμένη συσκευή
  - Κλοπή προσωπικών δεδομένων
  - Καταχρηστική υπηρεσία τηλεφωνημάτων χωρίς την συγκατάθεση του χρήστη
  - Κλικ απάτη – διαφημίσεις ή άλλες κακόβουλες λειτουργίες κρύβονται πίσω από τα κλικ
  - Κακόβουλος Downloader – κατεβάζει κακόβουλες εφαρμογές στη μολυσμένη συσκευή
- [24]



Εικόνα 2.1: Στατιστικά στοιχεία των malwares για το 2012

Όπως είναι εμφανές, εκείνη η περίοδος ήταν δύσκολη για το όνομα του Android και έπρεπε να απαντήσει η Google με πράξεις, έτσι και έγινε. Με την 4.2 έκδοση του Android που κυκλοφόρησε τον Νοέμβριο του 2012, ήρθαν αλλαγές στην ασφάλεια με κύριο γνώρισμα το σκανάρισμα Sideloadιng εφαρμογών. Με τον όρο Sideloadιng εννοούμε τις εφαρμογές που έχουν εγκατασταθεί στην συσκευή, μεταφέροντας το αρχείο apk από τον υπολογιστή στην συσκευή, εφαρμογές δηλαδή που δεν προέρχονται από το ηλεκτρονικό κατάστημα Google Play. Αν ένα αποτύπωμα μιας εφαρμογής ταιριάζει με κάποιο γνωστό malware, τότε το λειτουργικό σύστημα μπλοκάρει την εγκατάσταση της εφαρμογής. [25]

Έρευνα του πανεπιστημίου του Cambridge το 2015, έδειξε ότι το 87% των Android συσκευών είναι ευάλωτες σε επιθέσεις από μολυσμένες εφαρμογές. Στην έρευνα αυτή χρησιμοποιήθηκαν στοιχεία από την εφαρμογή Device Analyzer που βρίσκεται διαθέσιμη στο Google Play. Σύμφωνα με τον κύριο συγγραφέα της μελέτης Daniel Thomas, αυτή η εφαρμογή συλλέγει δεδομένα από χρήστες που δέχθηκαν να συμμετάσχουν στην εφαρμογή, και στην συνέχεια δημιουργεί τα στατιστικά στοιχεία που χρειάζονται. [26]



Εικόνα 2.2: Ποσοστό συσκευών που τρέχουν "μη ασφαλές", "μάλλον ασφαλές" και "ασφαλές" εκδόσεις του Android

Σύμφωνα με την έρευνα, ευθύνη έχουν οι κατασκευαστές των συσκευών καθώς δεν παρέχουν ενημερώσεις ασφαλείας και διορθώσεις σε κρίσιμα τρωτά σημεία. Υπολογίστηκε μέσω αλγορίθμου, σε κλίμακα 1 έως 10 κατά πόσο είναι μία συσκευή είναι ασφαλής. Ο αλγόριθμος αυτός υπολογίζει τις μέρες που οι συσκευές τρέξαν χωρίς να έχουν μολυνθεί, τις συσκευές που τρέχουν την νέα έκδοση του Android και το μέσο όρο των ευπαθειών που δεν έχουν επιλυθεί σε καμία συσκευή που διαθέτει στην αγορά η εταιρεία. Αυτός ο αλγόριθμος ονομάζεται FUM, κάθε γράμμα αντιστοιχεί στα παραπάνω.

Name	<i>f</i>	<i>u</i>	<i>m</i>	score (out of 10)
nexus	0.39 ± 0.00	0.48 ± 0.00	0.56 ± 0.01	5.17 ± 0.02
LG	0.22 ± 0.00	0.33 ± 0.00	0.62 ± 0.01	3.97 ± 0.02
Motorola	0.18 ± 0.00	0.12 ± 0.00	0.71 ± 0.02	3.07 ± 0.02
Samsung	0.13 ± 0.00	0.04 ± 0.00	0.61 ± 0.00	2.75 ± 0.00
Sony	0.14 ± 0.00	0.19 ± 0.00	1.09 ± 0.02	2.63 ± 0.02
HTC	0.14 ± 0.00	0.10 ± 0.00	0.87 ± 0.01	2.63 ± 0.02
asus	0.20 ± 0.00	0.51 ± 0.01	6.01 ± 0.07	2.35 ± 0.02
other	0.06 ± 0.00	0.05 ± 0.00	1.04 ± 0.01	1.97 ± 0.02
alps	0.03 ± 0.00	0.19 ± 0.01	3.99 ± 0.08	0.80 ± 0.02
Symphony	0.00 ± 0.00	0.08 ± 0.00	5.00 ± 0.05	0.30 ± 0.01
walton	0.00 ± 0.00	0.09 ± 0.00	6.00 ± 0.08	0.27 ± 0.01

Εικόνα 2.3: Αποτέλεσμα ασφάλειας βάση κατασκευαστών

Η Google παρουσίασε τα κύρια μέρη από τα οποία αποτελείται το πρόγραμμα ασφαλείας του Android, αυτά είναι:

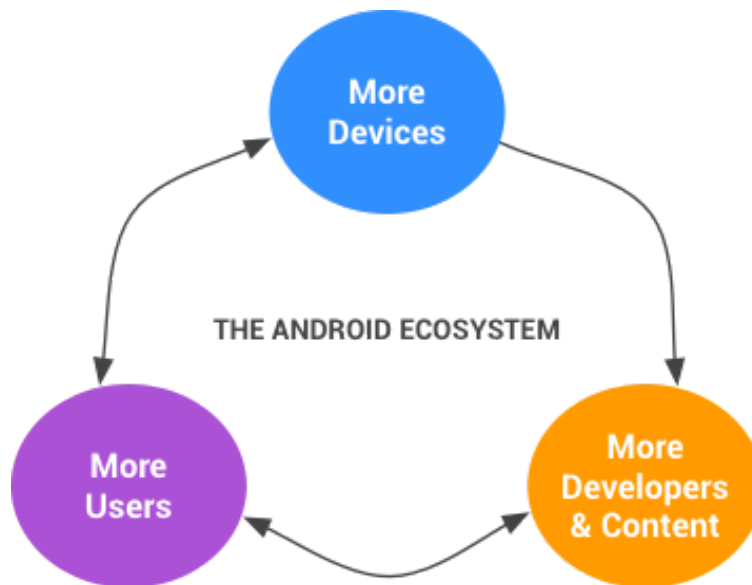
- Αναθεώρηση σχεδιασμού
- Ανάλυση κώδικα και testing
- Αντιμετώπιση περιστατικών
- Μηνιαίες ενημερώσεις ασφαλείας
- Αναθεώρηση από την κοινότητα ανοιχτού κώδικα [27]

#### 2.1.4. Open-Source κοινότητα

Το λειτουργικό σύστημα του Android δεν θα ήταν ο γίγαντας που είναι τώρα αν δεν ήταν open-source, AOSP (Android Open Source Project) [28]. Η Google έχει διαθέσει τον κώδικα του Android σε όλους και αυτό έχει οδηγήσει σε μια τεράστια ανάπτυξη διαφορετικών έργων αλλά και στην εξέλιξη του ίδιου του Android. Όπως αναφέρθηκε προηγουμένως, κάποιοι κατασκευαστές καθυστερούν χαρακτηριστικά ή και ακόμα δεν διαθέτουν ποτέ νέες ενημερώσεις για τις συσκευές. Έτσι, η κοινότητα έρχεται να καλύψει αυτό το κενό με εξίσου αντίστοιχη ποιότητα. Παλαιές συσκευές που δεν επιδέχονται νέες ενημερώσεις, μπορούν μέσω της ανοιχτής κοινότητας να βρουν ενημερώσεις ώστε να μπορούν να είναι λειτουργικές. Τα πιο διαδεδομένα projects ανοιχτού κώδικα είναι:

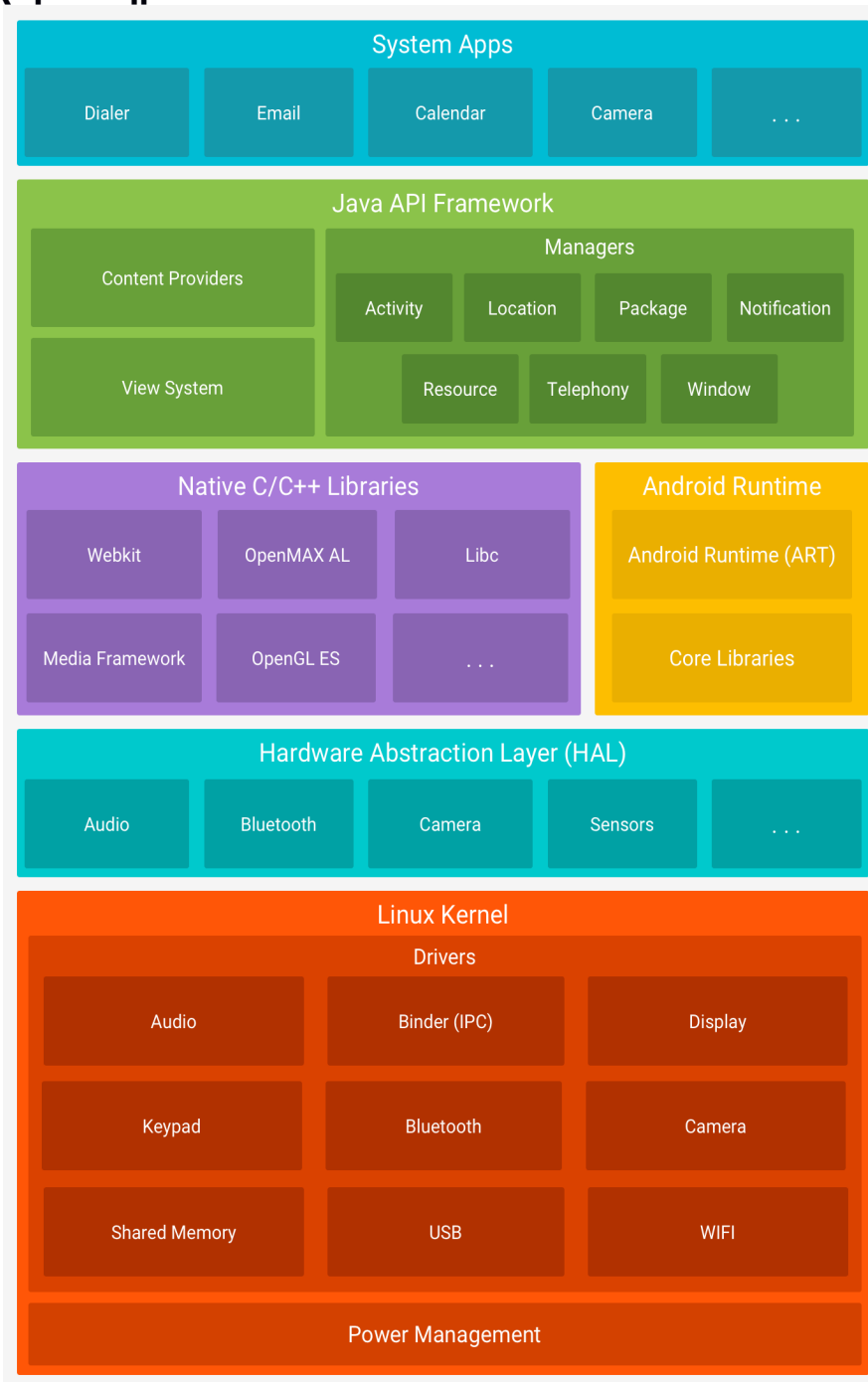
- Δημιουργία custom firmware
- Root συσκευής
- Διαχείριση του hardware (overclock, over/undervolt)
- Επίλυση bugs στο λειτουργικό σύστημα του Android
- Προσθήκη περαιτέρω λειτουργιών στο λειτουργικό σύστημα [29]

Έχοντας το Android open-source είναι θετικό για όλους. Για τους χρήστες γιατί μπορούν να έχουν την απόλυτη ελευθερία και να προσαρμόσουν το λειτουργικό σύστημα στις ανάγκες τους, αλλά και για την Google καθώς μπορεί να αποκομίσει τα οφέλη σε αντίθεση με άλλα λειτουργικά συστήματα που υπάρχουν στην διαθεσιμότητα του κοινού ως standalone συστήματα.



Εικόνα 2.4: Το οικοσύστημα Android ευδοκιμεί με τη συμβατότητα των συσκευών

## 2.2. Τεχνικά χαρακτηριστικά



Εικόνα 2.5: Αρχιτεκτονική του Android

### 2.2.1. Linux kernel

Όλα τα λειτουργικά συστήματα έχουν τον πυρήνα τους (kernel), έτσι και το Android χρησιμοποιεί τον πυρήνα του Linux αλλά κάθε συσκευή έχει την δικιά της έκδοση. Ο πυρήνας υποστηρίζει λειτουργίες του λειτουργικού συστήματος όπως διεργασίες, drivers (wifi, bluetooth, εικόνας, ήχου, usb, πληκτρολογίου, κάμερας), διαχείριση μνήμης κτλ [30].

Η εκδοχή του πυρήνα του Linux για το Android, έχει υποστεί περαιτέρω αλλαγές που πραγματοποιήθηκαν από την Google όπως:

- ashmem(Android Shared Memory) – διευκολύνει την κοινή χρήση και διατήρηση της μνήμης
- ION – Ασύγχρονη δικτύωση [31]
- Διαφορετικούς χειρισμούς του OOM (Out of memory)
- device trees
- pmem (Physical Memory) – ένας driver για τη χαρτογράφηση μεγάλων κομματιών (chunks) φυσικής μνήμης
- wakelocks – Είναι μηχανισμοί διαχείρισης ενέργειας, οι οποίοι διασφαλίζουν ότι η συσκευή δεν απενεργοποιείται επειδή μια συγκεκριμένη εφαρμογή χρησιμοποιεί τους πόρους του συστήματος. [32]

Το Linux kernel χρησιμοποιεί έναν διαφορετικό τρόπο διάθεσης νέων λειτουργιών από ότι το AOSP (Android Open Source Project). Μέχρι και το 2003, υπήρχαν δύο branch που συντηρούσε η κοινότητα του πυρήνα, το ένα ήταν η σταθερή έκδοση του kernel και το άλλο η development έκδοση. Με την διάθεση της 2.3 έκδοσης του πυρήνα τον Δεκέμβρη του 2003 αποφασίστηκε να συνεχιστεί μόνο η σταθερή (stable) έκδοση καθώς ήταν δύσκολη και χρονοβόρα η συντήρηση δύο branch και τα μεγάλα χρονικά διαστήματα μεταξύ την διάθεση των εκδόσεων. [33]

Στο επίπεδο του λειτουργικού συστήματος, η πλατφόρμα του Android παρέχει την ασφάλεια του πυρήνα του Linux, όπως και ακόμα ασφαλή επικοινωνία μεταξύ διαδικασιών IPC ώστε να παρέχουν ασφαλή επικοινωνία μεταξύ εφαρμογών που τρέχουν σε διαφορετικές διαδικασίες. Ακόμα, ο πυρήνας του Linux παρέχει στο Android κάποια σημαντικά χαρακτηριστικά:

- Μοντέλο αδειών βάσει χρηστών
- Επέκταση του μηχανισμού για ασφαλή IPC
- Δυνατότητα αφαίρεσης περιττών και πιθανών μη ασφαλών τμημάτων του πυρήνα [34]

Με την έκδοση 8.1.0 του Android προστέθηκε μια λειτουργία προστασίας του πυρήνα με όνομα KASLR (kernel Address Space Layout Randomization), με την οποία είναι πιο δύσκολο οι εισβολείς να εκμεταλλευτούν εξ'ολοκλήρου τον πυρήνα. Το KASLR διατίθεται στους πυρήνες των εκδόσεων του Android από την 4.4 και πάνω. [35]

Το partition του συστήματος περιέχει τον πυρήνα του Android και ακόμα βιβλιοθήκες του λειτουργικού συστήματος, το χρόνο εκτέλεσης της εφαρμογής, το πλαίσιο εφαρμογής και τις εφαρμογές. Το partition αυτό μπορεί μόνο να διαβαστεί και όχι να τροποποιηθεί/επεξεργαστεί, αλλά όμως είναι εφικτή η προσθήκη δικαιωμάτων για αλλαγές από τον κάτοχο της συσκευής με διάφορους τρόπους, αλλά όχι εξ αρχής καθώς είναι ξεκάθαρο ότι αυτό θα έκανε το λειτουργικό σύστημα εξαιρετικά ευάλωτο σε επιθέσεις που αφορούν το ίδιο το Android και την ακεραιότητα της συσκευής.

### 2.2.1. Ανάλυση λειτουργικού συστήματος

HAL – Hardware Abstraction Layer, παρέχει μια μέθοδο αλληλεπίδρασης του λειτουργικού συστήματος του Android και του υλικού. Το HAL μπορεί να κληθεί είτε από τον πυρήνα είτε από έναν οδηγό συσκευής. [36]

ART- Android runtime, είναι η διαχείριση του χρόνου εκτέλεσης(runtime) που χρησιμοποιείται από εφαρμογές και υπηρεσίες του συστήματος τους Android. Κάποια από τα κύρια χαρακτηριστικά του ART είναι:

- AOT(ahead-of-time) compilation
- Βελτιωμένη συλλογή περιττών δεδομένων
- Βελτίωση στην ανάπτυξη και επίλυση(debugging) [37]

Native C/C++ Βιβλιοθήκες. Το android runtime και το hardware abstraction layer, επειδή είναι low level υπηρεσίες, είναι γραμμένα σε C και C++, έτσι η πλατφόρμα του Android παρέχει Java framework APIs ώστε να δώσει την δυνατότητα χρησιμοποίησης τους στο επίπεδο των εφαρμογών. [38]

Java API Framework. Το Android παρέχει στον χρήστη ολόκληρο το λειτουργικό σύστημα μέσω APIs γραμμένα σε Java. Στόχος είναι η απλοποίηση των βασικών λειτουργιών και υπηρεσιών του συστήματος ώστε να μπορούν να χρησιμοποιηθούν χωρίς να χρειάζεται εξειδικευμένη γνώση τους. [38]

Εφαρμογές συστήματος. Το Android περιέχει προεγκατεστημένες κάποιες εφαρμογές που είναι αναγκαίες για την χρήση της συσκευής, όπως:

- Τηλεφωνική κλήση
- Επαφές
- Περιήγηση στο internet
- Κάμερα
- Ρολόι
- Email
- Google Store [38]

## 2.3. Εκδόσεις Android

Τα χαρακτηριστικά που αναφέρονται παρακάτω, είναι συνδυασμός προσωπικής έρευνας και της παραπομπής [40] της βιβλιογραφίας.

### 2.3.1. Android 1.0

Η πρώτη σταθερή έκδοση του Android (API 1) που κυκλοφόρησε τον Σεπτέμβρη του 2008 περιλαμβάνοντας τα ακόλουθα χαρακτηριστικά:

- Κατέβασμα και ενημέρωση εφαρμογών από το Android Market
- Βασικές εφαρμογές όπως ξυπνητήρι, αριθμομηχανή, τηλεφωνικές κλήσεις, φωτογραφίες, ρυθμίσεις, home screen
- Ανταλλαγή μηνυμάτων SMS και MMS
- Υποστήριξη Wi-Fi και Bluetooth
- Αναπαραγωγή τραγουδιών και βίντεο
- Φυλλομετρητής
- Υποστήριξη κάμερας
- Συγχρονισμός εφαρμογών της Google όπως Gmail, Επαφές και Google Agenda
- Google Maps

- Εφαρμογή YouTube

### 2.3.2. Android 1.1

Η έκδοση του Android 1.1 (API 2) κυκλοφόρησε τον Φεβρουάριο του 2009, που κυρίως επίλυσε σημαντικά bugs και περιλάμβανε τα ακόλουθα χαρακτηριστικά:

- Δυνατότητα αποθήκευσης συννημένων αρχείων που στάλθηκαν μέσω MMS
- Προβολή λεπτομερειών και κριτικών κατά την προβολή επιχειρήσεων στους χάρτες
- Προβολή και απόκρυψη πληκτρολογίου με αριθμούς στην εφαρμογή της τηλεφωνικής κλήσης
- Αυξήθηκε ο χρόνος που σβήνει η οθόνη όταν πραγματοποιείται κλήση και χρησιμοποιείται η ανοιχτή ακρόαση

### 2.3.3. Android 1.5 (Cupcake)

Το Android 1.5 (API 3) είναι η πρώτη έκδοση που χρησιμοποιεί κωδικοποιημένο όνομα, το Cupcake. Η έκδοση περιλαμβάνει τα ακόλουθα χαρακτηριστικά:

- Καταγραφή και προβολή video
- Υποστήριξη Bluetooth A2DP και AVRCP
- Υποστήριξη εφαρμογών πληκτρολογίου από τρίτους
- Λειτουργία πρόβλεψης λέξεων
- Αυτόματη περιστροφή συσκευής
- Δυνατότητα μεταμόρφωσης video στο YouTube και Picasa
- Λειτουργίες αντιγραφής επικόλλησης στο web
- Υποστήριξη widgets – συντομεύσεις των εφαρμογών για άμεση πρόσβαση σε αυτές
- Κινούμενες μεταβάσεις οθόνης

### 2.3.4. Android 1.6 (Donut)

Το Android 1.6 (API 4) με κωδική ονομασία Donut, κυκλοφόρησε τον Σεπτέμβριο του 2009. Η έκδοση περιλαμβάνει τα ακόλουθα χαρακτηριστικά:

- Υποστήριξη WVGA αναλύσεις οθονών
- Βελτιώσεις ταχύτητας στην αναζήτηση και στις εφαρμογές της κάμερας
- Δυνατότητα στην επιλογή πολλών φωτογραφιών για διαγραφή
- Ευκολότερη αναζήτηση και δυνατότητα προβολής screenshot στο Android Market
- Δυνατότητα στις εφαρμογές να “μιλήσουν” ένα string κειμένου
- Υποστήριξη φωνητικής γραφής (text-to-speech)
- Gesture framework

### 2.3.5. Android 2.0 – 2.1 (Eclair)

Η έκδοση Eclair του Android, περιείχε τις εκδόσεις 2.0 (API 5) κυκλοφόρησε τον Οκτώβριο του 2009, 2.0.1 (API 6) κυκλοφόρησε τον Δεκέμβριο του 2009 και 2.1 (API 7) κυκλοφόρησε τον Ιανουάριο του 2010. Οι εκδόσεις περιλαμβάνουν τα ακόλουθα χαρακτηριστικά:

- Υποστήριξη περισσότερων οθονών και αναλύσεων



- Βελτίωση GoogleMaps 3.1.2
- Προσθήκη νέων λειτουργιών στην κάμερα, όπως το φλας, zoom, εφέ χρωμάτων κτλ.
- Υποστήριξη Bluetooth 2.1
- Επιτρέπει στους χρήστες να έχουν πολλούς λογαριασμούς
- Δυνατότητα αναζήτησης αποθηκευμένων SMS και MMS
- Βελτίωση ταχύτητας πληκτρολόγησης και με εξυπνότερο λεξιλόγιο
- Υποστήριξη HTML5
- Βελτίωση UI του φυλλομετρητή
- Βελτίωση της ταχύτητας του hardware
- Υποστήριξη live wallpaper
- Microsoft Exchange
- Βελτιώσεις στο UI
- Αλλαγές και τροποποιήσεις στο API
- Επίλυση προβλημάτων και συμπεριφοράς του framework

### **2.3.6. Android 2.2 – 2.2.3 (Froyo)**

Η έκδοση Froyo του Android, περιείχε τις εκδόσεις 2.2 (API 8) κυκλοφόρησε τον Μάιο του 2010, 2.2.1 κυκλοφόρησε τον Ιανουάριο του 2011, 2.2.2 κυκλοφόρησε τον Ιανουάριο του 2011 και 2.2.3 κυκλοφόρησε τον Νοέμβριο του 2011. Οι εκδόσεις περιλαμβάνουν τα ακόλουθα χαρακτηριστικά:

- Βελτιώσεις ταχύτητας
- Υποστήριξη JIT
- USB Tethering και Wi-Fi hotspot
- Υποστήριξη του Android Cloud to Device Messaging (C2DM), με δυνατότητα push ειδοποιήσεων
- Γρήγορη εναλλαγή μεταξύ πολλαπλών γλωσσών στο πληκτρολόγιο
- Υποστήριξη animated GIF
- Οι εφαρμογές μπορούν να εγκατασταθούν στην επεκτάσιμη μνήμη
- Υποστήριξη Adobe Flash
- Υποστήριξη ανεβάσματος αρχείων μέσω φυλλομετρητή
- Υποστήριξη οθονών 720p
- Επίλυση προβλημάτων
- Ενημερώσεις στην ασφάλεια
- Βελτιώσεις στην επίδοση
- Επίλυση προβλημάτων στο SMS routing
- Ενημερώσεις ασφάλειας

### **2.3.7. Android 2.3 – 2.3.7 (Gingerbread)**

Η έκδοση Gingerbread του Android, περιείχε τις εκδόσεις 2.3 (API 9) κυκλοφόρησε τον Δεκέμβριο του 2010, 2.3.1 κυκλοφόρησε τον Δεκέμβριο του 2010, 2.3.2 (API 9) κυκλοφόρησε τον Ιανουάριο του 2011, 2.3.3 (API 10) κυκλοφόρησε τον Φεβρουάριο του 2011, 2.3.4 κυκλοφόρησε τον Απρίλιο του 2010, 2.3.5 κυκλοφόρησε τον Ιούλιο του 2011, 2.3.6



κυκλοφόρησε τον Σεπτέμβριο του 2011 και 2.3.7 κυκλοφόρησε τον Σεπτέμβριο του 2011 . Οι εκδόσεις περιλαμβάνουν τα ακόλουθα χαρακτηριστικά:

- Βελτιώσεις στο UI με έμφαση στην ταχύτητα και την απλότητα
- Βελτιώσεις στην χρήση του πληκτρολογίου
- Βελτιώσεις στην αντιγραφή/επικόλληση
- Βελτιώσεις στην διαχείριση ενέργειας
- Υποστήριξη πολλών καμερών, όπως της μπροστινής κάμερας
- Βελτιώσεις στον ήχο, γραφικά και input για game developers
- Υποστήριξη ομιλίας και βιντεοκλήσης με το Google Talk
- Βελτιώσεις στην εφαρμογή του Gmail
- Διορθώσεις προβλημάτων στην φωνητική λειτουργία
- Υποστήριξη Google Wallet για το Nexus S 4G

### **2.3.8. Android 3.0 – 3.2.6 (Honeycomb)**

Η έκδοση Honeycomb του Android, περιείχε τις εκδόσεις 3.0 (API 11) κυκλοφόρησε τον Φεβρουάριο του 2011, 3.1 (API 12) κυκλοφόρησε τον Μάρτιο του 2011, 3.2 (API 13) κυκλοφόρησε τον Ιούλιο του 2011, 3.2.1 κυκλοφόρησε τον Σεπτέμβριο του 2011, 3.2.2 κυκλοφόρησε τον Σεπτέμβριο του 2011, 3.2.3 κυκλοφόρησε τον Σεπτέμβριο του 2011, 3.2.4 κυκλοφόρησε τον Σεπτέμβριο του 2011, 3.2.5 κυκλοφόρησε τον Σεπτέμβριο του 2011 και 3.2.5 κυκλοφόρησε τον Σεπτέμβριο του 2011. Οι εκδόσεις περιλαμβάνουν τα ακόλουθα χαρακτηριστικά:

- Υποστήριξη πολλών πυρήνων(core)
- Καλύτερη υποστήριξη στα tablet
- Ενημέρωση του 3D UI
- Google talk video chat
- Google eBooks
- Ανώνυμη περιήγηση
- HTTP Live Streaming
- Βελτιώσεις στο UI
- USB host API
- Υποστήριξη ποντικιού, joystick και gamepad
- Βελτιώσεις για περισσότερα tablets
- Συγχρονισμός τραγουδιών, βίντεο από SD κάρτα
- Βελτιώσεις Wi-Fi
- Βελτιώσεις του Android Market
- Βελτιώσεις στο hardware
- Βελτιώσεις στα Google βιβλία
- Βελτιώσεις στο Adobe Flash

### **2.3.9. Android 4.0 – 4.0.4 (Ice Cream Sandwich)**

Η έκδοση Ice Cream Sandwich του Android, περιείχε τις εκδόσεις 4.0 (API 14) κυκλοφόρησε τον Οκτώβριο του 2011, 4.0.1 κυκλοφόρησε τον Οκτώβριο του 2011, 4.0.2 κυκλοφόρησε τον Νοέμβριο του 2011, 4.0.3 (API 15) κυκλοφόρησε τον Δεκέμβριο του 2011, 4.0.4 κυκλοφόρησε τον Μάρτιο του 2012. Οι εκδόσεις περιλαμβάνουν τα ακόλουθα χαρακτηριστικά:

- Νέες λειτουργίες κλειδώματος οθόνης, όπως η πρόσβαση σε εφαρμογές
- Βελτιώσεις στην εισαγωγή κειμένου και στον έλεγχο λαθών
- Ευκολότερη δημιουργία φακέλων
- Δυνατότητα λήψης στιγμιοτύπου οθόνης
- Καλύτερος χειρισμός των δεδομένων δικτύου
- 1080p καταγραφή video
- Επεξεργασία εικόνων
- Αυτόματος συγχρονισμός δεδομένων του Chrome
- Επίλυση προβλημάτων
- Βελτιώσεις στα γραφικά, βάσεις δεδομένων και Bluetooth
- Νέα APIs για τους προγραμματιστές
- Βελτιώσεις στο ημερολόγιο
- Καλύτερη περιστροφή οθόνης
- Βελτιώσεις στην σταθερότητα του συστήματος

### 2.3.10. Android 4.1 – 4.3.1 (Jelly Bean)

Η έκδοση Ice Cream Sandwich του Android, περιείχε τις εκδόσεις 4.1 (API 16) κυκλοφόρησε τον Ιούλιο του 2012, 4.1.1 κυκλοφόρησε τον Ιούλιο του 2012, 4.1.2 κυκλοφόρησε τον Οκτώβριο του 2012, 4.2 (API 17) κυκλοφόρησε τον Νοέμβριο του 2012, 4.2.1 κυκλοφόρησε τον Νοέμβριο του 2012, 4.2.2 κυκλοφόρησε τον Φεβρουάριο του 2013, 4.3 (API 18) κυκλοφόρησε τον Ιούλιο του 2013 και 4.3.1 κυκλοφόρησε τον Οκτώβριο του 2013. Οι εκδόσεις περιλαμβάνουν τα ακόλουθα χαρακτηριστικά:

- Προσθήκη του Google Now
- Φωνητική αναζήτηση
- Βελτιώσεις ταχύτητας
- CPU boost
- Βελτιωμένη προσβασιμότητα
- Βελτιώσεις στην εφαρμογή της κάμερας
- Προσβασιμότητα: λειτουργία χειρονομίας, ενεργοποίηση εξωτερικών πληκτρολογίων braille
- Οι συντομεύσεις των εφαρμογών αλλάζουν σχήμα δυναμικά για καλύτερη διαχείριση του χώρου
- Επεκτάσιμες ειδοποιήσεις
- Ενεργοποίηση περιστροφής της αρχικής οθόνης
- Widgets όταν είναι κλειδωμένο το κινητό (lockscreen)
- Εικόνες 360 μοιρών με το Photo Sphere
- Νέα εφαρμογή ρολογιού
- Group Messaging
- Ασύρματη φόρτιση
- Υποστήριξη 4K ανάλυσης
- Bluetooth Audio/Video
- Προβολή ποσοστού που έχει κατεβασθεί για μια εφαρμογή
- Βελτιώσεις στην ασφάλεια
- Αυτόματη συμπλήρωση για το πληκτρολόγιο κλήσης

### 2.3.11. Android 4.4 – 4.4.4 (KitKat)

Η έκδοση KitKat του Android, περιείχε τις εκδόσεις 4.4 (API 19) κυκλοφόρησε τον Οκτώβριο του 2013 , 4.4.1 κυκλοφόρησε τον Δεκέμβριο του 2013, 4.4.2 κυκλοφόρησε τον Δεκέμβριο του 2013, 4.4.3 κυκλοφόρησε τον Ιούνιο του 2014 και 4.4.4 κυκλοφόρησε τον Ιούνιο του 2014. Οι εκδόσεις περιλαμβάνουν τα ακόλουθα χαρακτηριστικά:

- Βελτιώσεις στην επίδοση σε συσκευές με χαμηλότερες προδιαγραφές
- Δυνατότητα ασύρματης εκτύπωσης
- Περιορισμός στις εφαρμογές κατά την πρόσβαση εξωτερικού αποθηκευτικού χώρου πέρα από τον δικό τους κατάλογο
- Δυνατότητα εγγραφής οθόνης
- Νέο διαφανές σύστημα UI
- Βελτιωμένη πρόσβαση στις ειδοποιήσεις
- Επίλυση της ευπάθειας του Heartbleed / OpenSSL
- Storage Access Framework, API που επιτρέπει στις εφαρμογές να λαμβάνουν αρχεία με σταθερό τρόπο
- Υποστήριξη MAP
- Το Android Runtime (ART) εισήχθη ως ένα νέο πειραματικό περιβάλλον χρόνου εκτέλεσης εφαρμογών
- Ανανέωση του της εφαρμογής των τηλεφωνικών κλήσεων

Ακόμα, η έκδοση Android 4.4W (API 20) , κυκλοφόρησε αποκλειστικά για έξυπνα ρολόγια android που περιλάμβανε:

- Offline διαθεσιμότητα μουσικής
- Υποστήριξη GPS
- Ενημερώσεις UI για πλοήγηση και συναγερμούς στο Google Maps

### 2.3.12. Android 5.0 – 5.1.1 (Lollipop)

Η έκδοση Lollipop του Android, περιείχε τις εκδόσεις 5.0 (API 21) κυκλοφόρησε τον Νοέμβριο του 2014, 5.0.1 κυκλοφόρησε τον Δεκέμβριο του 2014, 5.0.2 κυκλοφόρησε τον Δεκέμβριο του 2014, 5.1 (API 22) κυκλοφόρησε τον Μάρτιο του 2015, 5.1.1 κυκλοφόρησε τον Απρίλιο του 2015. Οι εκδόσεις περιλαμβάνουν τα ακόλουθα χαρακτηριστικά:

- Android Runtime (ART) με προχρονολόγηση (ahead-of-time compilation) AOT και βελτιωμένη συλλογή απορριμάτων
- Υποστήριξη 64-bit επεξεργαστών
- Βελτίωση της κατανάλωσης μπαταρίας
- Βελτίωση απόδοσης
- Διορθώσεις σφαλμάτων
- Υποστήριξη πολλαπλών SIM καρτών
- Προστασία συσκευής αν κλαπεί ή αν χαθεί
- Δυνατότητα αναζήτησης στις ρυθμίσεις
- Εφαρμογές εκτός συστήματος μπορούν ξανά να διαβάζουν και να επεξεργάζονται δεδομένα σε εξωτερικό χώρο αποθήκευσης
- Φωνητική κλήση υψηλής ευκρίνειας
- Δυνατότητα δημιουργίας λογαριασμών επισκεπτών
- Η οθόνη κλειδώματος δεν υποστηρίζει πλέον widgets

- Διανυσματικά γραφικά στοιχεία, τα οποία κλιμακώνονται χωρίς να χάσουν ποιότητα στην ανάλυση τους
- Προσθήκη εφαρμογής φακού που χρησιμοποιεί το φλας της συσκευής

### **2.3.13. Android 6.0 – 6.0.1 (Marshmallow)**

Η έκδοση Marshmallow του Android, περιείχε τις εκδόσεις 6.0 (API 23) κυκλοφόρησε τον Οκτώβριο του 2015 και 6.0.1 κυκλοφόρησε τον Δεκέμβριο του 2015. Οι εκδόσεις περιλαμβάνουν τα ακόλουθα χαρακτηριστικά:

- Υποστήριξη USB-C
- Υποστήριξη ταυτοποίησης με δακτυλικό αποτύπωμα
- Καλύτερη διάρκεια ζωής της μπαταρίας με το “deep sleep”
- Android pay
- Υποστήριξη MIDI
- Βελτιώσεις Google Now
- Πίνακας ελέγχου δικαιωμάτων
- Doze mode, μειώνει την ταχύτητα του επεξεργαστή όταν η οθόνη είναι σβηστή
- Μεγαλύτεροι φάκελοι εφαρμογών με πολλές σελίδες
- Οι εφαρμογές μπορούν να μπαίνουν σε standby mode
- Γραμμή αναζήτησης εφαρμογών

### **2.3.14. Android 7.0 – 7.1.2 (Nougat)**

Η έκδοση Nougat του Android, περιείχε τις εκδόσεις 7.0 (API 24) κυκλοφόρησε τον Αύγουστο του 2016, 7.1 (API 25) κυκλοφόρησε τον Οκτώβριο του 2016, 7.1.1 κυκλοφόρησε τον Δεκέμβριο του 2016 και 7.1.2 κυκλοφόρησε τον Απρίλιο του 2017. Οι εκδόσεις περιλαμβάνουν τα ακόλουθα χαρακτηριστικά:

- Δυνατότητα εστίασης οθόνης
- Daydream, πλατφόρμα εικονικής πραγματικότητας
- Καλύτερο multitasking
- Υποστήριξη multi-window mode
- Αδιάλειπτες ενημερώσεις συστήματος
- Βελτιώσεις στην περιήγηση αρχείων
- Περισσότερες γρήγορες ρυθμίσεις
- Αλλαγές στο UI
- Βελτιώσεις απόδοσης της αφής και προβολής
- Βελτιώσεις διαχείρισης αποθήκευσης
- Φωτάκι νυκτός
- Ειδοποιήσεις μπαταρίας

### **2.3.15. Android 8.0 – 8.1 (Oreo)**

Η έκδοση Oreo του Android, περιείχε τις εκδόσεις 8.0 (API 26) κυκλοφόρησε τον Αύγουστο του 2017 και 8.1 (API 27) κυκλοφόρησε τον Δεκέμβριο του 2017. Οι εκδόσεις περιλαμβάνουν τα ακόλουθα χαρακτηριστικά:

- Project Treble, αρθρωτή αρχιτεκτονική που διευκολύνει και επιταγχύνει τους κατασκευαστές του hardware να παραδίδουν ενημερώσεις του Android
- Υποστήριξη εικόνας σε εικόνα
- Ανασχεδιασμός γρήγορων ρυθμίσεων
- Βελτιωμένο σύστημα ειδοποιήσεων
- Βελτιωμένες ρυθμίσεις συστήματος
- Ανασχεδιασμός οθόνη κλειδώματος
- Υποστήριξη πολλαπλών οθονών
- Google Play Protect
- Μπορούν να κατεβασθούν γραμματοσειρές
- Βοηθός Wi-Fi
- Νευρωνικό Δίκτυο API
- Κοινόχρηστη μνήμη API
- Android Go
- Μηνύματα Toast τώρα έχουν λευκό background
- Βελτίωση στον χρόνο εκκίνησης συσκευής
- Προσαρμοστικά εικονίδια
- Ανακατασκευάστηκαν οι ρυθμίσεις έτσι ώστε να χωρίζονται σε ενότητες οι παρόμοιες καταχωρήσεις

## 2.4. Δομή Εφαρμογής

### 2.4.1. Βασικές αρχές εφαρμογής

Μια εφαρμογή Android μπορεί να γραφτεί με τις γλώσσες προγραμματισμού Java, C++ και πιο πρόσφατα με την Kotlin. Η τελική μορφή της υλοποιημένης εφαρμογής έχει κατάληξη “apk”, αυτό γίνεται με την βοήθεια του Android SDK που μεταγλωττίζει τον κώδικα μαζί με τα υπόλοιπα στοιχεία και αρχεία σε ένα Android πακέτο. Κάθε εφαρμογή περιέχει τα δικά της στοιχεία ασφάλειας, που είναι:

- Το λειτουργικό σύστημα του Android είναι ένα σύστημα Linux πολλαπλών χρηστών όπου κάθε εφαρμογή δουλεύει ως διαφορετικός χρήστης
- Το σύστημα αναθέτει σε κάθε εφαρμογή ένα ID μαζί με τα αντίστοιχα δικαιώματα
- Κάθε διαδικασία έχει την δικιά του εικονική μηχανή έτσι ώστε, ο κώδικας της εφαρμογής να τρέχει μεμονωμένα, να μην επηρεάζεται δηλαδή από άλλες εφαρμογές
- Κάθε εφαρμογή τρέχει στην δικιά της Linux διαδικασία. Το Android χειρίζεται την εφαρμογή έτσι ώστε να μην καταναλώνει άσκοπα πόρους όταν δεν χρειάζεται να λειτουργήσει [\[41\]](#)

### 2.4.2. Στοιχεία εφαρμογής

Τα στοιχεία της εφαρμογής είναι τα βασικά δομικά στοιχεία μιας εφαρμογής Android. Κάποια από αυτά εξαρτώνται από άλλα. Το Android εφαρμόζει την αρχή του least privilege, δηλαδή τα λιγότερα προνόμια. Αυτό σημαίνει, κάθε εφαρμογή έχει πρόσβαση μόνο στα στοιχεία που χρειάζεται για να δουλέψει, με αυτόν τον τρόπο πετυχαίνεται η προστασία της συσκευής και της λειτουργικότητας της, καθώς κάποιες κακόβουλες εφαρμογές με αποκλειστικό σκοπό την μόλυνση της συσκευής, δεν έχουν τα απαραίτητα δικαιώματα ώστε να κάνουν καταστροφικές ενέργειες. Υπάρχουν τέσσερα βασικά στοιχεία μιας εφαρμογής:

- Activities
- Services
- Broadcast Receivers
- Content Providers

Κάθε τύπος εξυπηρετεί τον δικό του σκοπό έχοντας ξεχωριστό κύκλο ζωής, ο οποίος είναι υπεύθυνος για την δημιουργία και καταστροφής του συγκεκριμένου στοιχείου. [41]

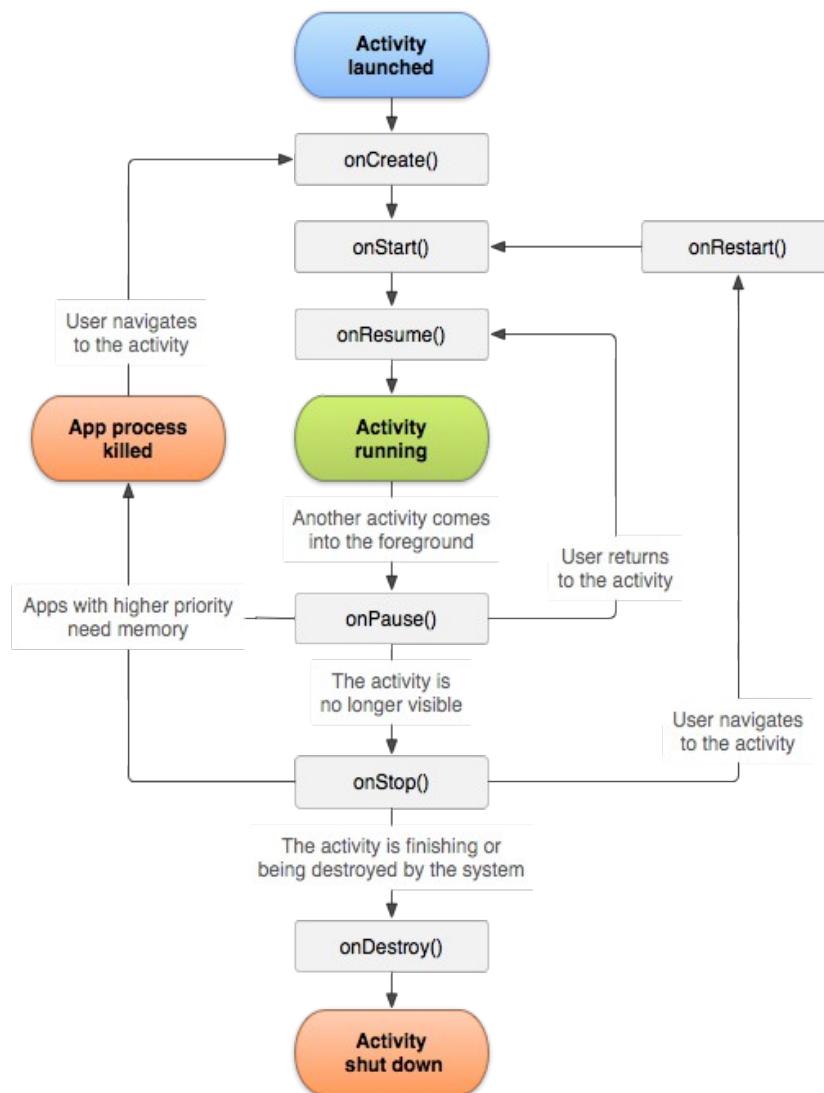
Υπάρχουν όμως και περισσότερα στοιχεία, αλλά εκείνα χρησιμοποιούνται στα πλαίσια των παραπάνω βασικών στοιχείων για να υλοποιηθεί η λογική τους και η σύνδεση μεταξύ τους. Αυτά τα στοιχεία είναι:

- Fragments
- Διαδικασίες
- Threads
- Views
- Layouts
- Intents
- Resources
- Manifest [42]

#### 2.4.2.1. Activities

Ένα activity αναπαριστά μια οθόνη με ένα περιβάλλον χρήστη, δηλαδή ένα activity εκτελεί ενέργειες στην οθόνη. Μια εφαρμογή μπορεί να έχει πολλές δραστηριότητες(activities) και μπορούν να συνεργαστούν μεταξύ τους, αλλά είναι αυτόνομες χωρίς να εξαρτώνται από άλλες δραστηριότητες. Όταν μία εφαρμογή έχει περισσότερες από μια δραστηριότητες, τότε πρέπει μία από αυτές να οριστεί ως η κύρια activity που θα παρουσιαστεί όταν θα εκτελεστεί η εφαρμογή για πρώτη φορά. Η κλάση “Activity” παρέχει μια σειρά επανακλήσεων που επιτρέπουν στο activity να γνωρίζει ότι η κατάσταση έχει αλλάξει. Μέσα στον κύκλο ζωής, πρέπει να δηλωθεί η κανονική συμπεριφορά της δραστηριότητας, όταν ο χρήστης φύγει και όταν επιστρέψει σε αυτήν. Είναι απαραίτητο κάθε δραστηριότητα να σεβαστεί τον κύκλο ζωής διαφορετικά θα υπάρξουν προβλήματα όπως το απότομο κλείσιμο της όταν ο χρήστης δεχθεί τηλεφωνική κλήση, κατανάλωση σημαντικών πόρων της συσκευής όταν δεν γίνεται χρήση της εφαρμογής, να μην αποθηκευθούν οι επιλογές του χρήστη και να κλείσει η εφαρμογή ή να χαθούν οι επιλογές του χρήστη όταν γίνεται περιστροφή της οθόνης. [42]





Εικόνα 2.6: Κύκλος ζωής ενός Activity

#### 2.4.2.2. Services

Μια υπηρεσία (service) είναι ένα στοιχείο της εφαρμογής που τρέχει στο background με στόχο την εκτέλεση μακροχρόνιων λειτουργιών. Μπορεί δηλαδή ο χρήστης να τρέχει μια εφαρμογή χωρίς να είναι απαραίτητα στην οθόνη, χρησιμοποιώντας άλλες εφαρμογές ή γενικότερα την συσκευή. Τέτοιου είδους υπηρεσίες συναντάμε σε εφαρμογές μουσικής ή κατέβασμα δεδομένων από το διαδίκτυο χωρίς να διακόπτει την λειτουργία μιας δραστηριότητας. [43]

#### 2.4.2.3. Broadcast Receivers

Ένας broadcast receiver, είναι ένα στοιχείο που επιτρέπει στο σύστημα να εκπέμπει γεγονότα στην εφαρμογή, έξω από τα πλαίσια της κανονικής ροής, επιτρέποντας στην εφαρμογή να ανταποκρίνεται σε μηνύματα που προέρχονται από άλλες εφαρμογές ή από το σύστημα. Για παράδειγμα, εφαρμογές μπορούν να επιτρέψουν την εκπομπή μηνυμάτων έτσι ώστε να επιτρέψουν σε άλλες εφαρμογές να γνωρίζουν ότι κάποια δεδομένα είναι διαθέσιμα για χρήση. Πολλές εκπομπές προέρχονται από το σύστημα, όπως οι εκπομπές που ενημερώνουν τον

χρήστη για την κατάσταση της μπαταρίας, την παραλαβή ενός email ή ενός μηνύματος, την απενεργοποίηση της οθόνης κτλ. Γνώρισμα των broadcast receivers είναι ότι δεν προβάλλουν κάποιο περιβάλλον χρήστη, αλλά μπορούν να δημιουργήσουν μια γραμμή ειδοποίησης ώστε να ενημερώσουν τον χρήστη ότι μια εκπομπή συνέβη. [43]

#### 2.4.2.4. Content Providers

Ένας πάροχος περιεχομένου(content provider), παρέχει δεδομένα από μια εφαρμογή σε άλλους μετά από αίτηση. Τα δεδομένα μπορούν να αποθηκευτούν στο σύστημα αρχείων, στη βάση δεδομένων ή σε οποιαδήποτε άλλη τοποθεσία αποθήκευσης στην οποία έχει πρόσβαση η εφαρμογή. Οι πάροχοι περιεχομένου είναι χρήσιμοι και για το διάβασμα και αποθήκευση δεδομένων που είναι ιδιωτικά και δεν μοιράζονται, όπως ένα offline σημειωματάριο. [43]

#### 2.4.2.5. Fragments

Ένα fragment είναι ένα κομμάτι μιας δραστηριότητας (activity), που επιτρέπουν μια πιο ευέλικτη σχεδίαση της δραστηριότητας. Ένα fragment θα μπορούσε να χαρακτηριστεί και ως υπο-δραστηριότητα. Μπορούν να συνδυαστούν πολλά fragments σε μία δραστηριότητα και να χρησιμοποιηθεί ένα fragment σε πολλές δραστηριότητες. Το fragment, έχει το δικό του κύκλο ζωής, δέχεται τα δικά του δεδομένα την ώρα που τρέχει η δραστηριότητα. Ένα fragment, πρέπει πάντα να ενσωματώνεται σε μια δραστηριότητα και ο κύκλος ζωής του επηρεάζεται άμεσα από τον κύκλο ζωής της κεντρικής δραστηριότητας. Ακόμα, έχει το δικό του σχεδιασμό και μπορεί να λειτουργήσει χωρίς να έχει απαραίτητα μια διεπαφή χρήστη. Πριν υπάρξουν τα fragments, υπήρχε περιορισμός στην προβολή μιας δραστηριότητας στην οθόνη, οπότε ήταν αδύνατη η διαίρεση της οθόνης σε ξεχωριστά κομμάτια και έτσι αυτή η ανάγκη έφερε την δημιουργία των fragments. Υπάρχουν τρία βασικά είδη fragments:

- Μονό fragment
- Λίστα fragments
- Μεταφορά fragment από το ένα στο άλλο [44][45]

#### 2.4.2.6. Διαδικασίες

Όλα τα στοιχεία της εφαρμογής εκτελούνται στην ίδια διαδικασία και προτείνεται οι εφαρμογές να μην κάνουν κάποια αλλαγή σε αυτό, αλλά υπάρχει η δυνατότητα τροποποίησης της λειτουργίας τους μέσω του αρχείου manifest. Το Android μπορεί να διακόψει την λειτουργία μιας διαδικασίας αν υπάρχει πρόβλημα με την χωρικότητα της μνήμης και πρέπει να διατεθεί σε πιο σημαντικές εφαρμογές. Το Android έχει ως κριτήριο για το ποια εφαρμογή θα κλείσει, αν βρίσκεται στην οθόνη εκείνη την συγκεκριμένη στιγμή. [46]

#### 2.4.2.7. Threads

Κατά την εκτέλεση μιας εφαρμογής, το σύστημα δημιουργεί ένα thread για την εφαρμογή το οποίο ονομάζεται “main” (κύριο). Αυτό το thread, είναι υπεύθυνο για την αλληλεπίδραση της εφαρμογής με το Android UI. Τα threads δίνουν την δυνατότητα στις εφαρμογές να τρέξουν κομμάτια κώδικα ξεχωριστά από το υπόλοιπο πρόγραμμα και με την προσθήκη χρόνου εκτέλεσης. Επίσης χρησιμοποιούνται συχνά σε σε δικτυακές υπηρεσίες και πρέπει να χρησιμοποιηθούν τα αποτελέσματα της κλήσης αυτής σε ένα επόμενο σημείο του κώδικα. Χωρίς την χρήση threads είναι εξαιρετικά πιθανή να πάρουμε κάποιο exception, και σε



περίπτωση που δούλευε τότε θα ήταν καθαρά θέμα τύχης και όχι σωστή πρακτική. Έτσι τα threads επιτρέπουν τον έλεγχο της δομής του κώδικα. [47]

### 2.4.3. Αρχείο manifest

Το αρχείο AndroidManifest.xml είναι γραμμένο σε γλώσσα XML και ο λόγος ύπαρξής του είναι να περιέχει δηλωμένα όλα τα στοιχεία της εφαρμογής γιατί από αυτό το αρχείο το σύστημα γνωρίζει ποια στοιχεία υπάρχουν. Ακόμα το αρχείο manifest, περιέχει το ελάχιστο επίπεδο API που απαιτεί η εφαρμογή για την λειτουργία της, δηλώνει τι θα χρησιμοποιήσει από το υλικό η εφαρμογή(κάμερα, bluetooth κτλ), δήλωση από που πηγάζουν οι βιβλιοθήκες που θα χρησιμοποιηθούν (όπως του Firebase) και προσδιορίζει τα δικαιώματα χρήστη που απαιτεί η εφαρμογή (όπως η χρήστη του internet, πρόσβαση στις επαφές του χρήστη κτλ.). Το AndroidManifest περιέχει ακόμα πληροφορίες όπως:

- Όνομα εφαρμογής
- Εικονίδιο εφαρμογής
- Θέμα/Στυλ εφαρμογής
- Όνομα πακέτου
- Κωδικοποίηση
- Έκδοση του manifest
- Ενεργοποιημένα services
- Ενεργοποιημένα receivers
- Πάροχους
- Έκδοση XML

Όταν κάποιες απαιτήσεις που βρίσκονται στο manifest δεν καλύπτονται από την συσκευή, τότε είναι αδύνατη η εγκατάσταση της εφαρμογής από το Google Play. Μπορούν βέβαια οι απαιτήσεις να γίνουν πιο ευέλικτες έτσι ώστε να μην αποκλείεται κάποιο ποσοστό συσκευών. [48]

### 2.4.4. Πόροι εφαρμογής

Μια εφαρμογή Android, εκτός από κώδικα, απαιτεί και πόρους. Πόροι μπορεί να είναι εικόνες, αρχεία ήχου και ότι συσχετίζεται την οπτική παρουσίαση της εφαρμογής. Κάθε πόρος, έχει το δικό του μοναδικό ID που του δίνεται από το SDK, με το οποίο μπορεί να αναφερθεί σε άλλα κομμάτια και σημεία του project κατά την υλοποίηση. Σημαντική πτυχή της ύπαρξης των πόρων σε διαφορετικό σημείο από ότι ο κώδικας, είναι η δυνατότητα παροχής εναλλακτικών πόρων για διαφορετικές διαμορφώσεις συσκευών (διαφορετικό μέγεθος οθόνης, διαφορετική γλώσσα κτλ.). Οι πόροι της εφαρμογής βρίσκονται στον φάκελο res του συστήματος

#### 2.4.4.1. Layout

Τα layouts είναι αρχεία XML, τα οποία χρησιμοποιούνται για την απεικόνιση των γραφικών που αντιστοιχούν στα activities και fragments, αλλά μπορεί ένα layout να ανήκει σε πολλές δραστηριότητες. Παρέχεται δυνατότητα τοποθέτησης γραφικών στοιχείων, βασισμένα σε layouts(relative,linear,grid,constraint, πίνακας). Κάποια από αυτά τα γραφικά στοιχεία είναι:

- Εισαγωγή και προβολή κειμένου
- Εικόνες και video
- Ημερομηνία

- RadioButtons
- CheckBox
- ProgressBar
- Προβολή Χάρτη

Τα παραπάνω , και τα υπόλοιπα διαθέσιμα στοιχεία από την παλέτα, μπορούν να τροποποιηθούν με την χρήση κώδικα αλλά και με την αλλαγή των λεπτομερειών τους από το πεδίο “Attributes”.

#### 2.4.4.2. Drawable

Ο φάκελος drawable περιέχει όλα τα γραφικά στοιχεία που αποτελούνται από εικόνες, σχήματα και animations. Τα σχήματα είναι αρχεία XML και μπορούν να συνδυαστούν με τα layouts. Ο φάκελος drawable έχει διαφορετικές κατηγορίες ανάλογα με την ανάλυση και το μέγεθος της οθόνης , όπως hdpi, mdpi,xhdpi,xxhdpi και xxxhdpi.

#### 2.4.4.3. Values

Ο φάκελος values περιέχει αρχεία XML, για την αποθήκευση συμβολοσειρών (strings.xml), διαστάσεων (dimens.xml), χρωμάτων (colors.xml), γραφικών (drawables.xml), στυλ θεμάτων (styles.xml). Ο χρήστης μπορεί να τα χρησιμοποιήσει καλώντας κάθε αντικείμενο με το όνομα του αντί για στατική χρήση. Με αυτόν τον τρόπο γίνεται πιο εύκολη η μαζική τροποποίηση αντικειμένων.

## 2.5. Android Studio

### 2.5.1. Περιγραφή Android Studio

Για την δημιουργία εφαρμογών είναι απαραίτητη η ύπαρξη ενός αναπτυξιακού περιβάλλον. Το Android Studio [1] είναι το επίσημο υποστηριζόμενο αναπτυξιακό περιβάλλον (IDE) με το οποίο μπορεί ο προγραμματιστής να δημιουργήσει εφαρμογές. Το Android Studio είναι βασισμένο στο IntelliJ IDEA που δημιουργήθηκε από την ομάδα JetBrains, και είναι αποκλειστικά σχεδιασμένο για ανάπτυξη εφαρμογών Android και τίποτα άλλο. Το IDE αυτό είναι διαθέσιμο σε Windows, macOS και Linux συστήματα [49]. Ήρθε για να αντικαταστήσει το ADT του Eclipse ως το κύριο περιβάλλον ανάπτυξης android εφαρμογών. Η έκδοση 0.1 ανακοινώθηκε τον Μάιο του 2013 στο Google I/O συνέδριο, και στην συνέχεια η beta έκδοση 0.8 ανακοινώθηκε τον Ιούνιο του 2014 [50] και έξι μήνες μετά , ανακοινώθηκε η σταθερή έκδοση 1.0 [51] .

Κάποια χαρακτηριστικά του Android Studio είναι:

- Σύστημα βασισμένο στο Gradle
- Emulator για τρέξιμο της εφαρμογής χωρίς την ανάγκη ύπαρξη συσκευής
- Δυνατότητα ανάπτυξης εφαρμογών για όλες τις Android συσκευές
- Άμεσο τρέξιμο της εφαρμογής χωρίς να χρειάζεται η δημιουργία apk αρχείου
- Υποστήριξη C++ και NDK
- Εργαλεία για testing
- Layout editor που δίνει την δυνατότητα στους χρήστες να κάνουν drag-and-drop γραφικά στοιχεία, με δυνατότητα προεπισκόπησης των αποτελεσμάτων σε διάφορες οθόνες

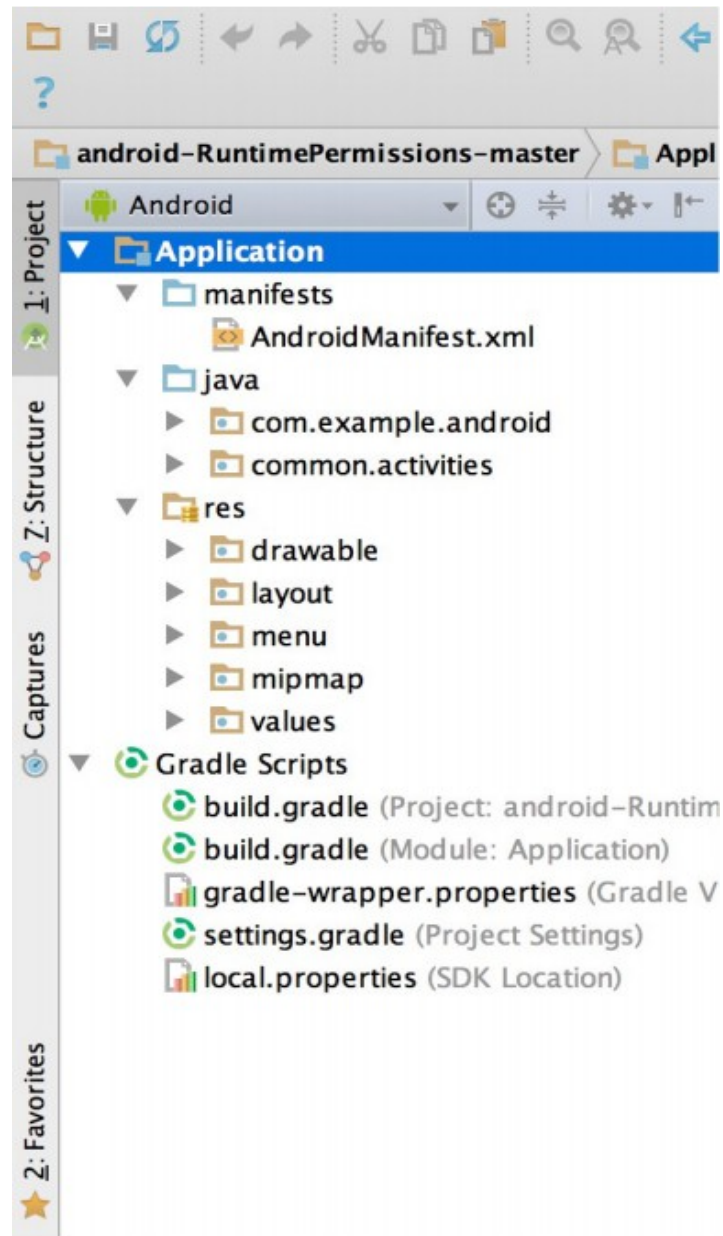
- Ενσωματωμένη υποστήριξη της πλατφόρμας του Google Cloud
- Αναλυτής APK [52]

Το Android Studio, παρουσιάζει τα αρχεία που ο χρήστης μπορεί να τροποποιήσει και αυτά βρίσκονται στον φάκελο Application, και αποτελούνται από τρεις κύριους υποφακέλους:

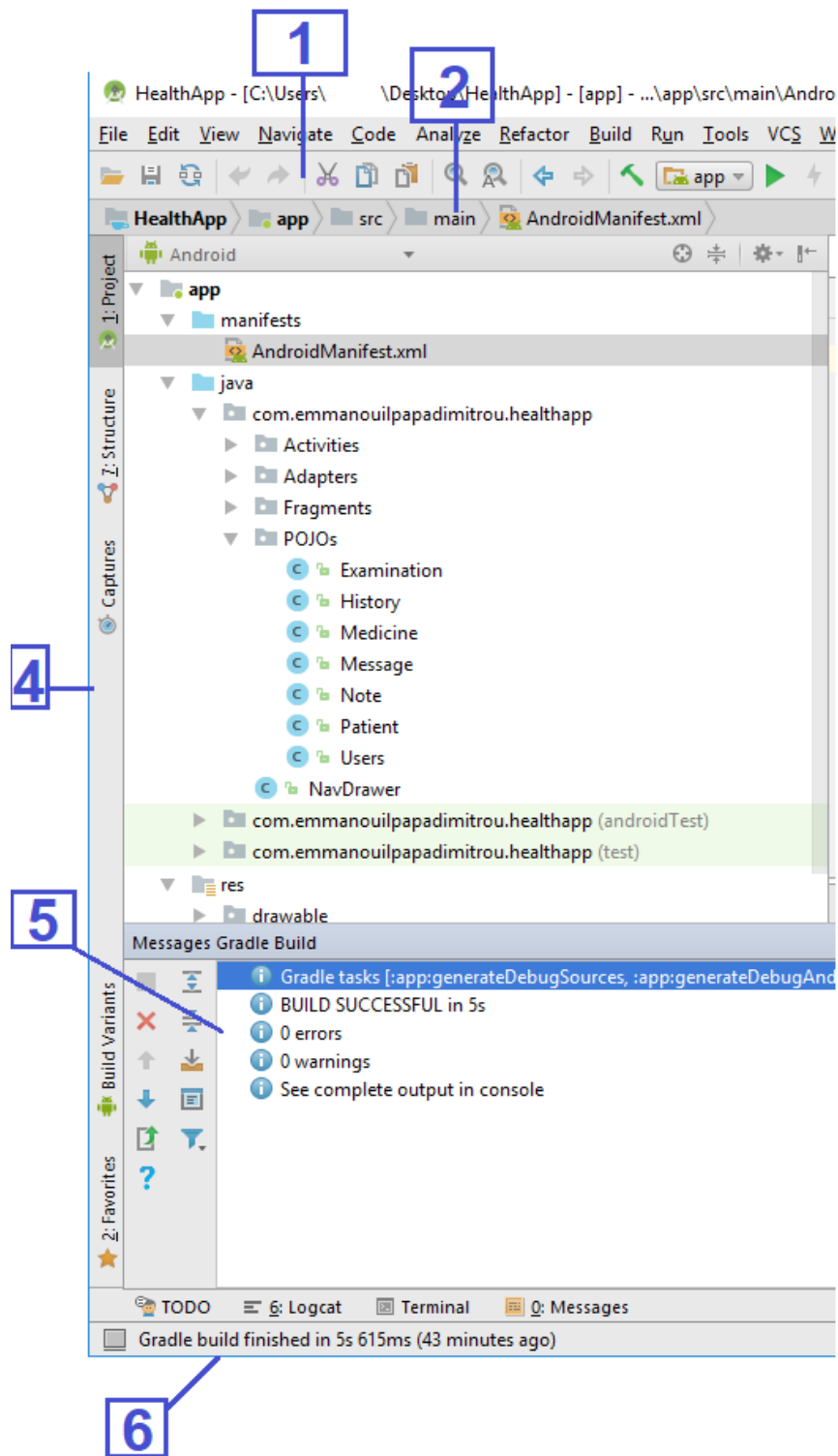
- manifests : περιέχει το αρχείο AndroidManifest.xml
- java: περιέχει τον πηγαίο κώδικα της εφαρμογής
- res: περιέχει όλους τους πόρους της εφαρμογής, όπως layouts, εικόνες, values και άλλα [52]

Το user interface (βλ. *Εικόνα 2.8 και 2.9*) της πλατφόρμας αποτελείται από:

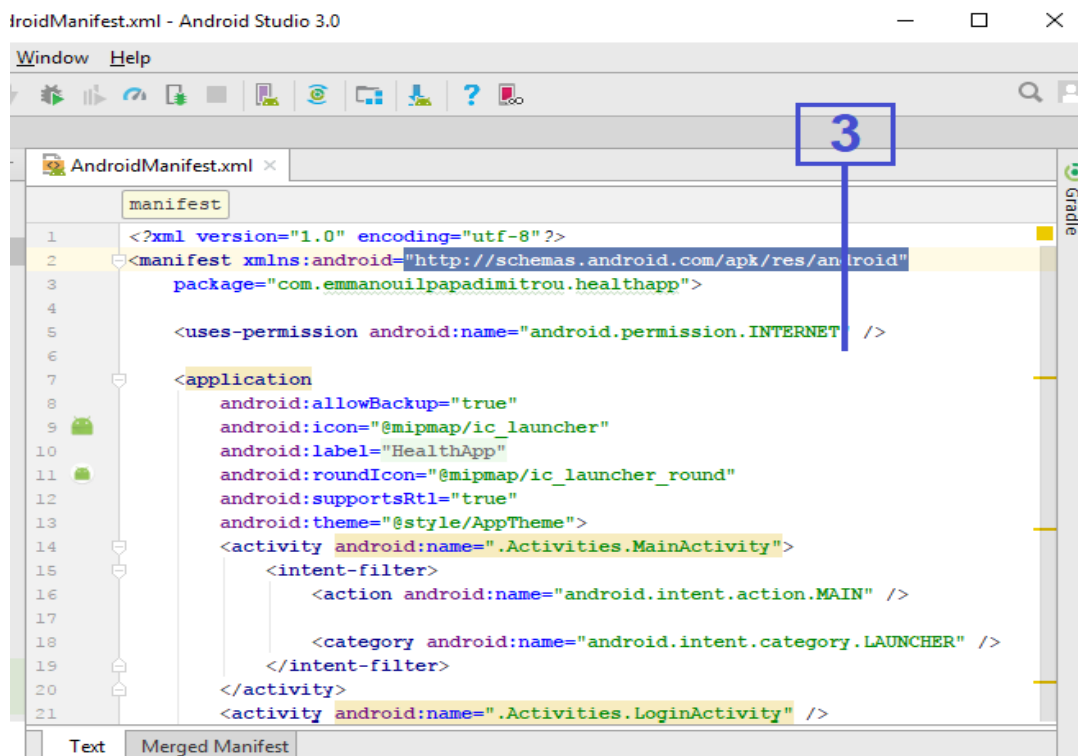
1. Γραμμή εργαλείων: δίνει την δυνατότητα άμεσης χρησιμοποιήσεως εργαλείων, όπως το τρέξιμο της εφαρμογής, το σταμάτημα και το debugging
2. Γραμμή πλοήγησης: βοηθά στην γρήγορη μεταφορά από αρχείο σε αρχείο και από φάκελο σε φάκελο
3. Παράθυρο επεξεργασίας: ο χώρος στον οποίο ο χρήστης μπορεί να επεξεργαστεί τα αρχεία του έργου, όπως τον κώδικα java, αρχεία XML κτλ.
4. Μπάρα παράθυρο εργαλείων: περιέχει τα κουμπιά που επιτρέπουν την επέκταση ή την σύμπτυξη μεμονωμένων παραθύρων εργαλείων
5. Παράθυρο εργαλείων: επιτρέπει την πρόσβαση σε εργασίες όπως debugger, αναζήτηση, έλεγχος εκδόσεων και άλλα
6. Γραμμή κατάστασης: εμφανίζει την κατάσταση που βρίσκεται το έργο (όπως ότι ολοκληρώθηκε το χτίσιμο της εφαρμογής) αλλά και μηνύματα διαφόρων ειδών, είτε για τον IDE είτε από το σύστημα [52]



Εικόνα 2.7: Δομή του project στο Android Studio



Εικόνα 2.9: User Interface-1



Εικόνα 2.9: User Interface-2

## 2.5.2. Emulators

Emulator σε περιβάλλον Android είναι η προσομοίωση μιας Android συσκευής στον υπολογιστή, έτσι ώστε ο δημιουργός να μπορεί να τρέχει την εφαρμογή του σε διαφορετικές συσκευές και να κάνει τις απαραίτητες αλλαγές. Έτσι, δεν είναι απαραίτητο ο δημιουργός να έχει στην κατοχή του μια Android συσκευή, και ακόμα η μεταφορά δεδομένων από το Android Studio στον emulator είναι πιο γρήγορη από ότι στην πραγματική συσκευή μέσω USB. Ένας emulator έχει πολλά χαρακτηριστικά που καλύπτουν τις ανάγκες, όπως μοντέλο συσκευής, ανάλυση οθόνης, μέγεθος οθόνης, πυκνότητα σε dpi, είδος συσκευής (smartphone, tablet, tv, ρολόι κτλ), χωρητικότητα RAM, σένσορες (γυροσκόπιο, gps κτλ.), κάμερα και έκδοση Android. Είναι εμφανές, ότι ένας emulator παρέχει όλες τις δυνατότητες μιας συσκευής και ακόμα την προσομοίωση τηλεφωνικό κλήσεων, μηνυμάτων, χρήση Google Play, χρήστη δικτύου και άλλα. [53]

### 2.5.2.1. AVD

Το Android Studio περιέχει emulator που χρησιμοποιούν AVD (Android Virtual Device) με δυνατότητα επιλογής πολλών χαρακτηριστικών συσκευής. Κάθε AVD λειτουργεί σαν μια αυτόνομη συσκευή με το δικό του αποθηκευτικό χώρο, SD κάρτα και επιλογές χρηστών. Το AVD έχει κάποιες απαιτήσεις όπως:

- Παρόμοιες απαιτήσεις συστήματος με το Android Studio, αλλά ο υπολογιστής πρέπει να έχει έναν 64-bit επεξεργαστή
- Εργαλεία SDK μεγαλύτερα από την έκδοση 25.0.10
- Ενεργοποιημένο το adb
- Η επιτάχυνση του emulator στα Windows απαιτεί Intel επεξεργαστή, ενώ στα Linux μπορεί να χρησιμοποιηθεί και AMD επεξεργαστής [53]

### 2.5.2.2. Genymotion

Ένας διαδεδομένος third-party emulator είναι το Genymotion {2}. Δεν περιλαμβάνεται στο Android Studio, αλλά μπορεί να χρησιμοποιηθεί αφού πρώτα γίνει εγκατάσταση του απαραίτητου plugin. Το Genymotion είναι ένας multi-platform emulator Android που τρέχει πάνω από το VirtualBox και βασικό του γνώρισμα είναι ότι υπερτερεί σε ταχύτητα έναντι του AVD. Το περιβάλλον χρήστη του προγράμματος είναι πολύ εύκολο στην χρήση του με δυνατότητα αλλαγής μεγέθους του emulator κατά βούληση. Ακόμα, σημαντικό στοιχείο του είναι η άμεση μεταφορά αρχείων με την χρήση drag and drop. Τέλος, δεν έχει μεγάλες απαιτήσεις στο μέγεθος των emulator. [54]

### 2.5.2.3. Σύγκριση AVD και Genymotion

Κύριος λόγος που προτιμάται το Genymotion έναντι του AVD, είναι η μεγάλη διαφορά ταχύτητας. Πραγματοποιήθηκαν τριών ειδών tests στους δύο emulators.

- **Test 1:** Boot συσκευής
- **Test 2:** Boot συσκευής και απευθείας ξεκίνημα εφαρμογής
- **Test 3:** Ξεκίνημα εφαρμογής με ανοιχτή την συσκευή

Η εφαρμογή που δοκιμάστηκε είναι η εφαρμογή που αφορά την πτυχιακή.

Tests	AVD (λεπτά)	Genymotion (λεπτά)	Διαφορά (δευτερόλεπτα)
Test 1	01:22:7	00:49:1	33
Test 2	02:10:5	01:15:4	55
Test 3	00:20:7	00:05:6	15

Το AVD περιέχει τα APIs της Google ενώ το Genymotion όχι, άρα χρειάζεται εγκατάσταση τους από τον χρήστη ώστε να μπορεί η εφαρμογή του να δουλέψει σωστά αν χρησιμοποιεί αυτά τα APIs, όπως το Firebase.

Το AVD δίνει περισσότερες επιλογές χαρακτηριστικών για τις συσκευές από ότι το Genymotion στο οποίο ο χρήστης επιλέγει μια συσκευή και την κατεβάζει και δεν μπορεί να κάνει πολλές αλλαγές.

Το AVD υποστηρίζεται άμεσα από την Google και περιέχει πάντα τις νέες εκδόσεις από το SDK, ενώ το Genymotion δεν έχει αυτήν την ευχέρεια.

Το Genymotion λειτουργεί σε όλους τους επεξεργαστές, σε αντίθεση με το AVD που στα Windows δουλεύει μόνο με intel και η υποστήριξη των AMD επεξεργαστών συμβαίνει μόνο στα Linux.

Συνοψίζοντας, αν υπάρχει η ανάγκη ώστε τα χαρακτηριστικά των συσκευών να μπορούν να τροποποιηθούν ανά πάσα στιγμή, τότε καλύτερο είναι το AVD. Όμως, το Genymotion είναι πιο γρήγορο και καταναλώνει λιγότερο χώρο στον υπολογιστή. Ανάλογα τις ανάγκες του ο χρήστης επιλέγει το καταλληλότερο.

{2} Ιστοσελίδα για λήψη και πληροφορίες του Genymotion <https://www.genymotion.com/>



## Κεφάλαιο 3: Firebase

### 3.1. NoSQL Βάσεις Δεδομένων

Οι NoSQL βάσεις δεδομένων είναι βάσεις δεδομένων που δεν χρησιμοποιούν πίνακες, και κατ'επέκταση ούτε στήλες, γραμμές ή σχήμα βάσης, για τον χειρισμό των δεδομένων τους όπως οι σχεσιακές βάσεις δεδομένων SQL (RDBMS). NoSQL σημαίνει “όχι SQL” ή “όχι μόνο SQL” [55]. Αυτές οι βάσεις, ενώ υπάρχουν αρκετές δεκαετίες, μόλις πρόσφατα άρχισαν να χρησιμοποιούνται και να υποστηρίζονται από μεγάλες εταιρείες όπως η Google, δεδομένου ότι οι RDBMS έχουν προβλήματα στην ταχύτητα, στην επεκτασιμότητα και την ευελιξία [56][57]. Οι NoSQL βάσεις είναι χρήσιμες για την αποθήκευση μη δομημένων δεδομένων τα οποία χρησιμοποιούνται στις μέρες όλο και πιο πολύ και δεν ταιριάζουν με την φιλοσοφία των RDBMS βάσεων, όπως ανταλλαγή μηνυμάτων, δεδομένα χρήστη, δεδομένα καταγραφής, δεδομένα που απαιτούνται στο Internet Of Things και multimedia όπως εικόνες και βίντεο [58].

Υπάρχουν τέσσερα διαφορετικά είδη NoSQL βάσεων που δημιουργήθηκαν για να καλύψουν διαφορετικές ανάγκες:

- Key-value βάσεις: Οι βάσεις αυτές χαρακτηρίζονται για την απλότητα τους, την ικανότητα τους να δίνουν και να παίρνουν δεδομένα πολύ γρήγορα στην εφαρμογή ειδικά σε εκείνα που θα χρειαστούν περισσότερες φορές. Τα δεδομένα είναι μπορούν να είναι αποθηκευμένα σε μορφή JSON, απλού κειμένου και βίντεο. Συναντάμε συχνά JSON αρχεία, καθώς μπορούν να ομαδοποιούν τα δεδομένα και να κάνουν την πρόσβαση τους σε αυτά πιο εύκολη και άμεση χωρίς να χρειάζεται να ψάξουν όλο το έγγραφο για όλα τα δεδομένα με την χρήση κλειδιού (key) [56]. Κάποιες τέτοιες βάσεις είναι:
  - DynamoDB
  - Azure Table Storage
  - Riak
  - Redis
  - LevelDB
  - Scalien
  - allegro-C [55]
- Βάσεις δεδομένων εγγράφου: Μια βάση δεδομένων εγγράφων χρησιμοποιείται για την αποθήκευση, την ανάκτηση και τη διαχείριση ημιδομημένων δεδομένων. Στα ημιδομημένα δεδομένα οι οντότητες που ανήκουν στην ίδια κλάση μπορεί να έχουν διαφορετικές ιδιότητες. Αυτές οι βάσεις χρησιμοποιούν JSON, XML και BSON αρχεία για την αποθήκευση των δεδομένων τους. Έχουν ομοιότητες με τις key-value βάσεις, αλλά η σημαντική διαφορά τους είναι ότι μία τιμή (value) είναι ένα έγγραφο που αποθηκεύει όλα τα δεδομένα που σχετίζονται με το συγκεκριμένο κλειδί. Για παράδειγμα, σε μία βάση το πρώτο έγγραφο να περιέχει το όνομα και την διεύθυνση ενός ανθρώπου ενώ το δεύτερο έγγραφο να περιέχει και παραπάνω στοιχεία όμως ημερομηνία γέννησης, χώρα κτλ [56]. Κάποιες τέτοιες βάσεις είναι :
  - MongoDB
  - Cloud Datastore
  - Elastic
  - ArangoDB



- NeDB [55]
- Ευρείας στήλης: Οι βάσεις ευρείας στήλης χρησιμοποιούν παρόμοιο τρόπο αποθήκευσης δεδομένων με τις RDBMS βάσεις, αλλά η σημαντική διαφορά είναι ότι κάθε στήλη μπορεί να έχει διαφορετική μορφή, δεν περιορίζεται δηλαδή όπως ένας πίνακας σε μια SQL βάση που πρέπει κάθε γραμμή να περιέχει τα ίδια στοιχεία. Τα πεδία μπορούν να ανακτηθούν χωρίς την γνώση κάποιου κλειδιού. Κάθε έγγραφο έχει ίδια ή διαφορετική δομή, δεν υπάρχει περιορισμός [56]. Κάποιες τέτοιες βάσεις είναι :
  - Cassandra
  - Scylla
  - Cloudata
  - Amazon SimbleDB
  - eXtremeDB
  - HBase [55]
- Γράφος: Μια τέτοια βάση δεδομένων βασίζεται στην μορφή του παραδοσιακού γράφου που είναι γνωστός από την πληροφορική. Με την χρήση του γράφου γίνεται αποθήκευση, χαρτογράφηση και διερεύνηση των σχέσεων μεταξύ των οντοτήτων. Παρέχουν γειτνίαση χωρίς την ύπαρξη κάποιου δείκτη, έτσι ώστε τα αντικείμενα να είναι συνδεδεμένα μεταξύ τους χωρίς κάποιου παραπάνω πεδίου. Στις σχεσιακές βάσεις δεδομένων, υπάρχει ο όρος του ξένου κλειδιού (foreign key) όταν υπάρχει η ανάγκη σύνδεσης δεδομένων μεταξύ πινάκων [56]. Κάποιες τέτοιες βάσεις είναι :
  - ArangoDB
  - GraphBase
  - Trinity
  - Onyx Database
  - weaver
  - InfoGrid [55]

Οι NoSQL βάσεις δεδομένων έχουν σημαντικά πλεονεκτήματα έναντι των παραδοσιακών RDBMS βάσεων, όπως τα παρακάτω:

- Ευελιξία: Οι βάσεις NoSQL κάνουν χρήση της μεθοδολογίας οριζόντιας κλιμάκωσης. Με αυτόν τον τρόπο γίνεται πιο εύκολη η προσθήκη ή αφαίρεση δεδομένων χωρίς να επηρεάσει οτιδήποτε άλλο. Στις RDBMS βάσεις δεν είναι μηδενικό το κόστος αλλαγών, καθώς με την προσθήκη ή την αφαίρεση πρέπει να γίνουν και άλλες ενέργειες που ενημερώνουν το σύστημα για αλλαγές.
- Επίδοση: Έχοντας την άνεση άμεσων αλλαγών στην βάση, αυτό φέρνει και την καλύτερευση του user experience καθώς κάνει την χρήση της εφαρμογής πιο γρήγορη.
- Μεγάλη διαθεσιμότητα: Οι βάσεις NoSQL είναι σχεδιασμένες ώστε να παρέχουν άμεσα δεδομένα χωρίς την περιπλοκότητα που έχουν οι RDBMS βάσεις που χρειάζονται δηλαδή να ενημερώσουν δευτερεύοντα στοιχεία σε οποιαδήποτε αλλαγή, είτε μικρή είτε μεγάλη. Γνώρισμα των NoSQL είναι η διαθεσιμότητα των δεδομένων ακόμα και όταν ένα πεδίο δεν είναι διαθέσιμο για οποιοδήποτε λόγο.
- Διαθεσιμότητα παντού: Κύριο γνώρισμα τους είναι η διάδοση δεδομένων σε πολλούς servers, κέντρα δεδομένων και στο cloud έτσι ώστε να ελαχιστοποιηθεί η καθυστέρηση και να εξασφαλιστεί η ομαλή λειτουργία της εφαρμογής χωρίς προβλήματα που θα υπήρχαν αν τα δεδομένα δεν ήταν διαθέσιμα ανά πάσα στιγμή.
- Ευέλικτη μοντελοποίηση δεδομένων: Το NoSQL προσφέρει τη δυνατότητα εφαρμογής ευέλικτων μοντέλων και πιο ρευστών δεδομένων. Δηλαδή, η αλληλεπίδραση ανάμεσα στην εφαρμογή και στην βάση δεδομένων να γίνει απλούστερη και ταχύτερη. [58]

### 3.2. Αναλυτική περιγραφή του Firebase

Κατά την δημιουργία μιας εφαρμογής είτε Android, iOS ή web, η βάση δεδομένων που είναι αναγκαία για την εφαρμογή αυτή μπορεί να καταναλώσει σε αρκετές περιπτώσεις πολλούς πόρους του δικτύου. Για παράδειγμα, αν μια εφαρμογή κοινοποιεί κάποιες εικόνες και ο χρήστης θέλει να έχει πρόσβαση σε όλες τις εικόνες, δεν είναι εύκολη η υποστήριξη όλων αυτών των αιτημάτων προς την βάση. Ακόμα ένα κομμάτι που έχει αρκετά λεπτά σημεία, είναι η αυθεντικοποίηση των χρηστών. Οι χρήστες χρειάζονται την δυνατότητα σύνδεσης και εγγραφής μέσω διαφόρων έτοιμων υπηρεσιών που προσφέρουν τα social media, αλλά και μια απλή σύνδεση του συστήματος δεν είναι εύκολη πολλές φορές να σχεδιαστεί από το μηδέν, είναι εφικτό αλλά πολύ χρονοβόρο και σε εφαρμογές που ο στόχος δεν είναι η συντήρηση κάποιου server, άρα δεν υπάρχει κάποια ομάδα που δουλεύει αποκλειστικά πάνω στον server αυτόν, σπαταλάει χρόνο από την υλοποίηση και την βελτίωση της λειτουργικότητας της εφαρμογής.



*Εικόνα 3.1: Λογότυπο Firebase*

Είναι λάθος να χαρακτηριστεί το Firebase ως αποκλειστικά και μόνο σαν βάση δεδομένων γιατί πολλά περισσότερα από αυτό μόνο. Το Firebase δίνει μια πλήρη λύση όλων των παραπάνω προβλημάτων. Είναι μια back-end υπηρεσία που ξεκίνησε ως μία startup εταιρεία (Firebase Inc.) το 2011 και πλέον έχει εξελιχθεί σε μια πλατφόρμα ανάπτυξης εφαρμογών νέας γενιάς του Google Cloud Platform [6]. Με το Firebase μπορεί ο δημιουργός να υλοποιήσει εύκολα τους πιο συχνούς τρόπους αυθεντικοποίησης χρηστών, όπως το email και κωδικό αλλά και την άμεση χρησιμοποίηση των APIs για σύνδεση και εγγραφή στο σύστημα μέσω Google, Facebook, GitHub κτλ. Εκτός από την αυθεντικοποίηση, το Firebase χειρίζεται το πρόβλημα των βάσεων δεδομένων, καθώς παρέχει μια βάση δεδομένων NoSQL πραγματικού χρόνου, αυτό σημαίνει ότι όταν ο χρήστης ζητήσει μια συγκεκριμένη λειτουργία από την εφαρμογή, το αίτημα που γίνεται στην βάση πραγματοποιείται σε πολύ μικρό χρόνο και η προβολή των αιτούμενων στοιχείων γίνεται πάρα πολύ γρήγορα. Οι δυνατότητες του Firebase δεν περιορίζονται στην αυθεντικοποίηση και στον χειρισμό βάσεων δεδομένων, αλλά και σε υπηρεσίες όπως cloud messaging, αποθήκευση αρχείων στο cloud, φιλοξενία εφαρμογών δικτύου, ειδοποιήσεις εφαρμογών, προγράμματα διαφήμισης (AdMob) και πολλά ακόμη που βρίσκονται σε beta κατάσταση όπως παρακολούθηση των επιδόσεων και εργαλεία καταγραφής όταν αποτυγχάνει η συσκευή να εκτελέσει κάποιες λειτουργίες (Crashlytics). [59]

Κύριο γνώρισμα του Firebase είναι ότι παρέχει APIs για τις γλώσσες προγραμματισμού που υλοποιούν εφαρμογές android, iOS, web, αλλά και Unity όπως Java, Swift και JavaScript αντίστοιχα. Δεν χρειάζεται δηλαδή ο χρήστης να μάθει κάποια καινούργια γλώσσα προγραμματισμού για να συνδέσει το έργο του με το Firebase, καθώς μπορεί να καλέσει τα επιθυμητά APIs στην γλώσσα που χρειάζεται.

Τελειώνοντας, το Firebase ελευθερώνει τους προγραμματιστές από το βάρος της δημιουργίας και συντήρησης ενός server που θα περιέχουν όλα τα δεδομένα προς ανάκτηση και αποθήκευση. Δεν απαιτεί την δημιουργία APIs καθώς παρέχει όλα όσα χρειάζονται για την πρόσβαση και την διαχείριση των δεδομένων. Η δομή του Firebase επιτρέπει την προσαρμογή των λειτουργιών της στις ανάγκες τις εκάστοτε εφαρμογής.



### 3.2.1. Αυθεντικοποίηση

Στις εφαρμογές που ο χρήστης έχει την ανάγκη να αποθηκεύσει τις επιλογές τους ώστε να μπορεί να έχει πρόσβαση ή να τις τροποποιήσει οποιαδήποτε στιγμή, εμφανίζεται η ανάγκη της προστασίας αυτών των επιλογών. Το κομμάτι της προστασίας υλοποιείται με την δημιουργία ενός συστήματος αυθεντικοποίησης χρήστη. Ο πιο συνηθισμένος τρόπος αυθεντικοποίησης των χρηστών είναι με την απαίτηση τους στο να γίνει η εγγραφή τους στο σύστημα, ζητώντας τους κατά κύριο λόγο email, όνομα χρήστη και κωδικό. Στην συνέχεια να συνδεθούν ώστε να μπορούν να δημιουργήσουν, να προβάλουν και να αλλάξουν τα δεδομένα που επιθυμούν.

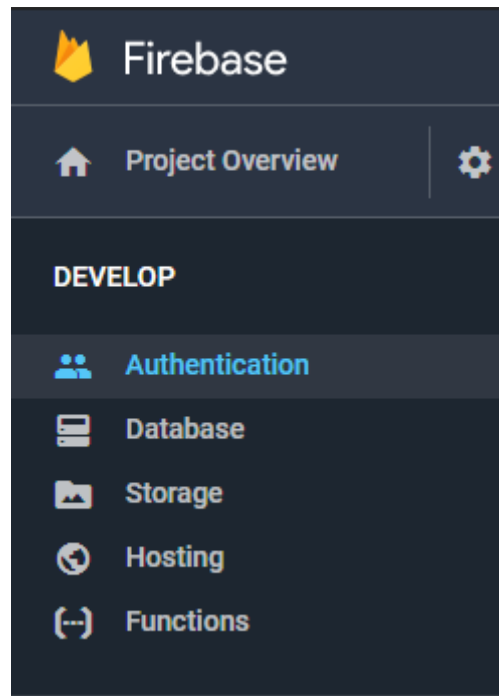
Το Firebase παρέχει την υπηρεσία Αυθεντικοποίησης (Authentication) περιέχοντας μια ενσωματωμένη λειτουργία αυθεντικοποίησης χρήστη με την χρήση email και κωδικού πρόσβασης αλλά επίσης υποστηρίζει παρόχους social media όπως Google, Facebook, GitHub έτσι ώστε ο χρήστης να μην χρειάζεται να κάνει και εγγραφή. Το Firebase Auth είναι μια υπηρεσία που μπορεί να αυθεντικοποιήσει χρήστες με την υπάρξη μόνο client-side κώδικα. Επιπλέον, παρέχει υπηρεσίες αυθεντικοποίησης μέσω τηλεφώνου και μέσω ανώνυμων λογαριασμών. Ανώνυμοι λογαριασμοί είναι προσωρινοί λογαριασμοί για την πρόσβαση στην εφαρμογή χωρίς να απαιτείται από τον χρήστη να κάνουν σύνδεση. Αν ο χρήστης το επιθυμεί στην συνέχεια μπορεί να εγγραφεί στο σύστημα. [60]

Τα βήματα για την χρήση του SDK της αυθεντικοποίησης είναι:

1. Ορισμός μεθόδων σύνδεσης
2. Δημιουργία του UI για την σύνδεση του χρήστη
3. Αποστολή στοιχείων του χρήστη με την χρήση του Firebase Authentication SDK [60]

Authentication				
USERS	SIGN-IN METHOD	TEMPLATES	USAGE	
<div><div><div><div></div></div><div>Search by email address, phone number, or user UID</div></div><div>ADD USER</div><div><div></div><div></div></div></div>				
Identifier	Providers	Created	Signed In	User UID ↑
manuelpap@gmail.com		Oct 24, 2017		m4T7HmDb7QeJL4ZXjpCW4E4Zw...
valkon.gr@gmail.com		Oct 11, 2017	Oct 11, 2017	yFi22O60j8cBySw8apYFz0vHGg82 <div><div></div><div></div></div>
Rows per page: 50 1-2 of 2 < >				

Εικόνα 3.2: Κονσόλα Αυθεντικοποίησης



Εικόνα 3.3: Καρτέλα  
Αυθεντικοποίησης (Authentication)

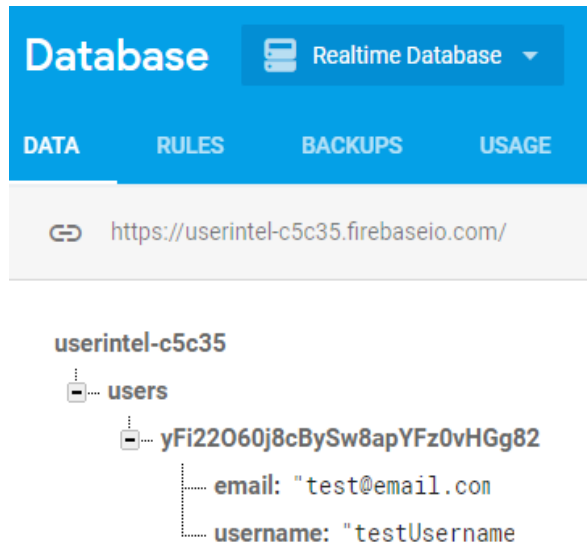
### 3.2.2. Βάση δεδομένων πραγματικού χρόνου

Οι περισσότερες βάσεις δεδομένων επιστρέφουν δεδομένα όταν και μόνο ο χρήστης τα ζητήσει. Η βάση δεδομένων πραγματικού χρόνου (realtime database) του Firebase, αποθηκεύει και ενημερώνει τα δεδομένα με την NoSQL cloud βάση δεδομένων. Κύριο γνώρισμα της βάσης αυτής είναι διαθεσιμότητα των δεδομένων ακόμη και όταν η συσκευή δεν έχει σύνδεση στο internet. Το Firebase στέλνει δεδομένα την στιγμή που αυτά ενημερώνονται. Τα δεδομένα αυτά αποθηκεύονται σε JSON αρχεία και ενημερώνονται σε πραγματικό χρόνο σε όλους τους συνδεδεμένους χρήστες, ακόμη και σε εφαρμογές που είναι cross-platform. Σύμφωνα με τα παραπάνω, ο τύπος NoSQL είναι document store (βάση δεδομένων εγγράφου). Αναλυτικότερα, τα κύρια χαρακτηριστικά είναι:

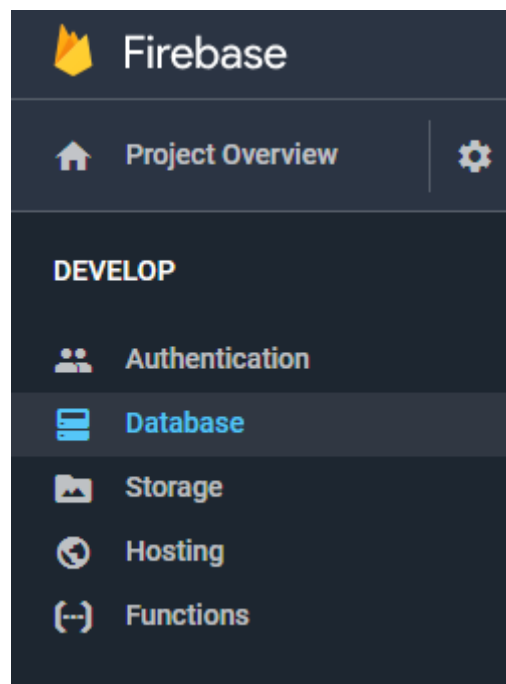
- **Πραγματικός χρόνος:** Οι περισσότερες βάσεις δεδομένων χρησιμοποιούν HTTP για να πραγματοποιήσουν τα αιτήματά τους. Το Firebase χρησιμοποιεί WebSockets, τα οποία είναι πολύ πιο γρήγορα από το HTTP, για την στιγμιαία ενημέρωση των δεδομένων σε κλάσματα του δευτερολέπτου.
- **Διαθεσιμότητα εκτός σύνδεσης:** Το SDK το Firebase έχει την δυνατότητα αποθήκευσης των δεδομένων στην συσκευή, έτσι τα δεδομένα είναι διαθέσιμα ακόμα και όταν δεν υπάρχει σύνδεση στο διαδίκτυο. Στις RDBMS βάσεις για να υλοποιηθεί αυτό θέλει περισσότερο χρόνο και έχει πολλές δυσκολίες στην υλοποίηση όπως με ποιον τρόπο θα αποθηκευτούν τα δεδομένα με την καλύτερη εκμετάλλευση του χώρου. Είναι εφικτό, αλλά ξεφεύγει από τα πλαίσια της σχεσιακής βάσης.
- **Διαθεσιμότητα:** Τα δεδομένα της βάσης μπορούν επεξεργαστούν είτε από μια κινητή συσκευή είτε από τον φυλλομετρητή. Τα δεδομένα δηλαδή είναι διαθέσιμα για όλες τις συσκευές και για όλες τις γλώσσες προγραμματισμού. Ακόμα το Firebase προσφέρει πολλά επίπεδα ασφάλειας και κανόνων, πράγμα το οποίο στις SQL βάσεις θέλει μεγάλη προσοχή γιατί υπάρχουν πολλοί τρόποι παραβίασης τους όπως SQL Injection, κάτι τέτοιο που φυσικά δεν υφίσταται στο Firebase. [61]

Τα βήματα της υλοποίησης είναι:

1. Ενσωμάτωση του SDK της βάσης δεδομένων πραγματικού χρόνου στην εφαρμογή
2. Δημιουργία της βάσης με JSON αρχείο
3. Ορισμός των δεδομένων και ενημέρωσή τους όταν γίνονται αλλαγές
4. Ενεργοποίηση διαθεσιμότητας των δεδομένων εκτός δικτύου
5. Ασφάλεια των δεδομένων με την χρήση των κανόνων ασφαλείας (Security Rules) [61]



Εικόνα 3.4: Κονσόλα βάσης δεδομένων



Εικόνα 3.5: Καρτέλα βάσης δεδομένων

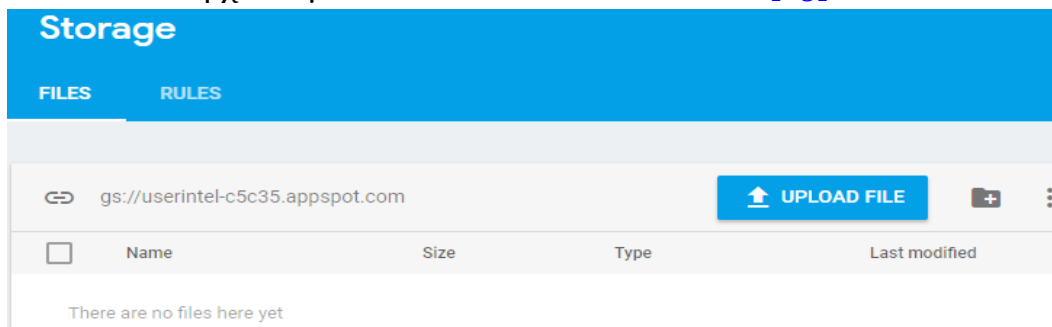
### 3.2.3. Αποθηκευτικός χώρος

Το Firebase παρέχει υπηρεσία αποθήκευσης αρχείων όπως εικόνες, video ή οποιοδήποτε τύπο αρχείου χρειαστεί. Έτσι μπορεί ο προγραμματιστής να χρησιμοποιήσει και αρχεία εκτός από δεδομένα κειμένου, με την χρησιμοποίηση απλώς του SDK. Οι βάσεις RDBMS, και πιο συγκεκριμένα η MySQL δεν μπορούν να αποθηκεύσει αρχεία από μόνη της, χρειάζεται περαιτέρω βιβλιοθήκες όπως το BLOB [62]. Τα κύρια χαρακτηριστικά του αποθηκευτικού χώρου του Firebase είναι:

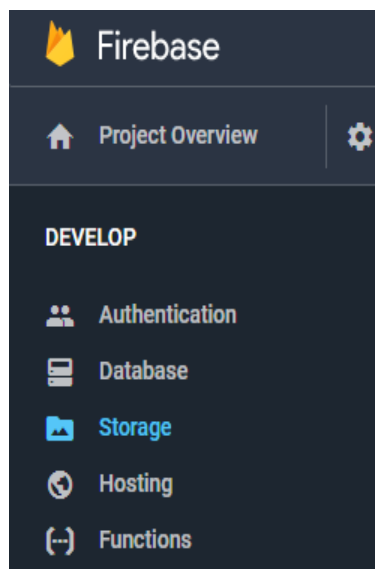
- Ισχυρές λειτουργίες για κατέβασμα και ανέβασμα αρχείων ανεξαρτήτως της ποιότητας του δικτύου.
- Προστασία των αρχείων σε συνδυασμό με το Firebase Authentication.
- Υψηλή επεκτασιμότητα όταν τα δεδομένα χρειάζονται να καλεστούν πολλές φορές από πολλούς χρήστες. [63]

Τα βήματα της υλοποίησης είναι:

1. Ενσωμάτωση του SDK για τον αποθηκευτικό χώρο
2. Εισαγωγή τοποθεσίας του αρχείου
3. Ανέβασμα ή κατέβασμα του αρχείου
4. Προστασία των αρχείων μέσω των κανόνων του Firebase [63]



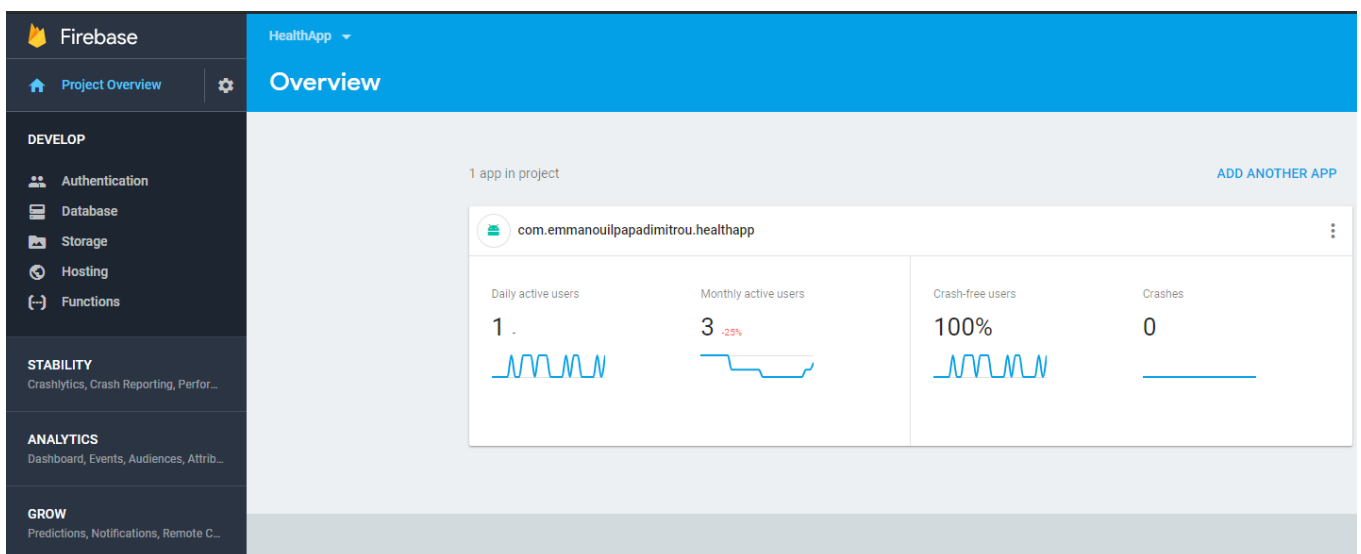
Εικόνα 3.6: Κονσόλα διαχείρισης αρχείων



Εικόνα 3.7: Καρτέλα διαχείρισης αρχείων

### 3.2.4. Λόγος επιλογής

Ο κύριος λόγος που επιλέχθηκε η πλατφόρμα του Firebase για την υποστήριξη των δεδομένων και των αρχείων της εφαρμογής που αφορά την πτυχιακή, είναι η προστασία και η ασφάλεια που παρέχει στα δεδομένα. Στην βάση δεδομένων μπορούν να προστεθούν, να αφαιρεθούν ή να τροποποιηθούν κανόνες ανάλογα με το αν ο χρήστης είναι επιβεβαιωμένος ώστε να μην είναι εφικτή η πρόσβαση στα δεδομένα από τρίτους. Αυτό είναι πολύ σημαντικό στην εφαρμογή, γιατί γίνεται διαχείριση των προσωπικών δεδομένων των ασθενών και πρέπει γίνεται σεβαστό το ιατρικό απόρρητο σε όλη την δομή της εφαρμογής. Επιπλέον, το Firebase παρέχει μια πλατφόρμα για την άμεση και εύκολη διαχείριση της αυθεντικοποίησης των χρηστών, βάσης δεδομένων και των αρχείων για ανέβασμα και κατέβασμα χωρίς να χρειάζεται να δημιουργηθεί μια web εφαρμογή για την διαχείριση τους όπως θα ήταν αναγκαίο να γίνει σε περίπτωση που η βάση ήταν SQL.



Εικόνα 3.7: Πλατφόρμα Firebase

### 3.2.5. Χρήση του Firebase

#### 3.2.5.1. Κονσόλα Firebase

##### 3.2.5.1.1. Δημιουργία project

Για την δημιουργία του project είναι απαραίτητος ο λογαριασμός Google. Στην συνέχεια, πηγαίνουμε στην ιστοσελίδα {1} που περιέχει την κονσόλα του Firebase για την επιλογή ή την δημιουργία νέου project και επιλέγεται το Add project για την δημιουργία. Μετά εμφανίζεται ένα παράθυρο στο οποίο καλείται ο χρήστης να εισάγει το όνομα του project. Από κάτω εμφανίζεται το μοναδικό ID του project του και έπειτα επιλέγει την χώρα του. Τέλος υπάρχει η επιλογή “CREATE PROJECT” που θα του δημιουργήσει το project του και θα τον οδηγήσει στην κονσόλα του Firebase, ή μπορεί να πατήσει “CANCEL” για να μην γίνει η δημιουργία του project.

{1} Ιστοσελίδα κονσόλας Firebase <https://console.firebase.google.com/>



### 3.2.5.1.1. Αυθεντικοποίηση χρήστη

Για την αυθεντικοποίηση χρήστη επιλέγουμε “Authentication” {2}. Αρχικά πρέπει να οριστεί μια μέθοδος σύνδεσης στο σύστημα {3}, κάποιες επιλογές είναι το email/κωδικός, τηλέφωνο, Google, Facebook, GitHub και ανώνυμη σύνδεση. Θα γίνει εστίαση μόνο στο email και κωδικό πρόσβασης. Οπότε, το επιλέγουμε από την λίστα και πατάμε “Enable”. Μπορούν να οριστούν και templates, δηλαδή πως θα είναι τα email που θα λαμβάνει ο χρήστης όταν πρέπει να επιβεβαιώσει το email του, όταν θα γίνει αναίρεση κωδικού και όταν θα γίνει αλλαγή του email. Μέσω του project μπορούμε να υλοποιήσουμε εγγραφή χρήστη στο σύστημα που θα τον εμφανίσει στο Authentication, αλλά δεν μας αφορά αυτό τώρα, οπότε βάζουμε έναν νέο χρήστη επιλέγοντας “ADD USER”, που εκεί θα μας ζητηθεί να πληκτρολογήσουμε email και κωδικό. Έτσι έχουμε τον πρώτο αυθεντικοποιημένο μας χρήστη.

### 3.2.5.1.2. Δημιουργία βάσης δεδομένων

Για την δημιουργία της βάσης δεδομένων επιλέγουμε “Database” {4} και στην συνέχεια “GET STARTED”. Γίνεται ο ορισμός της βάσης που έχει το δικό της URL. Μπορεί να γίνει προσθήκη δεδομένων είτε γράφοντας τα απευθείας στην online κονσόλα, είτε δημιουργώντας ένα JSON αρχείο και στην συνέχεια να γίνει η εισαγωγή του. Υπάρχει η δυνατότητα εξαγωγής της βάσης σε αρχείο JSON οποιαδήποτε στιγμή. Στην συνέχεια, συναντάμε μία πάρα πολύ βασική δυνατότητα του Firebase, την επιλογή κανόνων {5}. Εδώ πέρα μπορούν να οριστούν ποιοι χρήστες έχουν την δυνατότητα προβολής, προσθήκης και επεξεργασία δεδομένων από και προς την βάση. Από τις πιο ασφαλείς επιλογές είναι η ανάθεση δικαιωμάτων διαβάσματος και εγγραφής μόνο από αυθεντικοποιημένους χρήστες, και αυτό επιτυγχάνεται με τον παρακάτω τρόπο:

```
1 {
2   "rules": {
3     ".read": "auth != null",
4     ".write": "auth != null"
5   }
6 }
```

Εικόνα 3.8: Κανόνες της βάσης δεδομένων

Με αυτόν τον πολύ απλό και γρήγορο τρόπο, έχει στηθεί η βάση και το σύστημα αυθεντικοποίησης των χρηστών και τώρα μπορεί ο χρήστης να συνδέσει το Firebase με το project του στο Android Studio.

{2} Authentication <https://console.firebase.google.com/project/ID/authentication/users>

{3} Λίστα επιλογών σύνδεσης <https://console.firebase.google.com/project/ID/authentication/providers>

{4} Database <https://console.firebase.google.com/project/ID/database/ID/data>

{5} Κανόνες <https://console.firebase.google.com/database/ID/rules>

### 3.2.5.2. Σύνδεση με το Android Studio

Για να μπορεί να συνδεθεί το project στο firebase με το project στο Android πρέπει να γίνει καταχώρηση της εφαρμογής. Με την επιλογή “Add Firebase to your Android app” ο χρήστης καλείται να καταχωρήσει το android πακέτο της εφαρμογής του, το οποίο μπορεί να το βρει από το αρχείο build.gradle του Android Studio. Μόλις το συμπληρώσει πατάει “REGISTER APP”.

Σε επόμενη οθόνη, κατεβάζει το “google-services.json” αρχείο το οποίο πρέπει να το τοποθετήσει στο project του στο Android Studio και πιο συγκεκριμένα μέσα στον φάκελο “app”. Έπειτα πρέπει να δηλωθούν οι υπηρεσίες google play στο build.gradle `com.google.gms.google-services`. Επίσης, πρέπει να δηλωθεί το ίδιο plugin στο build.gradle του επιπέδου app. Στην συνέχεια, στο ίδιο αρχείο, δηλώνονται τα dependencies που είναι απαραίτητα για την χρήση των λειτουργιών του Firebase, όπως το

- `com.google.firebase:firebase-auth:ΕΚΔΟΣΗ`
- `com.google.firebase:firebase-database:ΕΚΔΟΣΗ`
- `com.firebaseui:firebase-ui-database:ΕΚΔΟΣΗ`
- `com.google.firebase:firebase-storage:ΕΚΔΟΣΗ`

Εφόσον χρειαστεί περεταίρω λειτουργικότητα, τα docs του Firebase περιέχουν αναλυτικά όλα τα dependencies που υπάρχουν με την νεότερη έκδοση τους.

Όπως είναι φανερό, η σύνδεση Firebase και Android Studio είναι πάρα πολύ εύκολη και άμεση και ο προγραμματιστής πλέον μπορεί να χρησιμοποιήσει τα APIs του Firebase σαν βιβλιοθήκες Java.

## 3.3. Σύγκριση με άλλες βάσεις δεδομένων

### 3.3.1. Σύγκριση με MySQL

Η MySQL είναι ίσως η πιο διαδεδομένη σχεσιακή βάση δεδομένων (RDBMS). Είναι επίσης ανοιχτού κώδικα, που εκτός από σχεσιακή βάση μπορεί να υλοποιήσει τα μοντέλα δεδομένου εγγράφου και key-value. Δημιουργήθηκε από την Oracle το 1995 με γλώσσες προγραμματισμού C και C++. Δεν έχει την δικιά της υπηρεσία cloud, αλλά σύνηθες είναι η συνεργασία με το rhemyadmin, όπου ο χρήστης δημιουργεί εκεί μια βάση δεδομένων ή χειρίζεται μια ήδη υπάρχουσα, και μπορεί να χρησιμοποιήσει όλες τις λειτουργίες που προσφέρει η MySQL αλλά σε γραφικό περιβάλλον. Η MySQL απαιτεί εγκατάσταση στο σύστημα του δημιουργού για την χρήση της ενώ το Firebase όπως φάνηκε χρησιμοποιείται από τον φυλλομετρητή και ο server δεν είναι εγκατεσμένος στον υπολογιστή του δημιουργού, όπως είναι απαραίτητο με την MySQL να έχει δικό της server. Επιπλέον, δεν παρέχει έτοιμα APIs όπως το Firebase που παρέχει για Android, iOS, JavaScript και γενικά RESTful APIs. Η MySQL παρέχει connectors για την εκάστοτε γλώσσα αλλά δεν υποστηρίζει έτοιμες πλατφόρμες, οπότε οι ρυθμίσεις που πρέπει να γίνουν οι περισσότερες από ότι στο Firebase. Το Firebase είναι υπηρεσία back-end και ο χρήστης δεν μπορεί να αλλάξει πολλά και ούτε χρειάζεται εφόσον τα παρέχει εύκολα, το μόνο σημείο που μπορεί να αλλάξει είναι οι κανόνες για το ποιος θα έχει πρόσβαση που. Σε αντίθεση με την MySQL, είναι αναγκαία η δημιουργία custom back-end λειτουργιών για την διαχείριση της βάσης και αυτό συνήθως παίρνει πολύ χρόνο και για την δημιουργία αλλά και για την συντήρηση του. Επίσης δεν υπάρχει η έννοια των “κανόνων” όπως στο Firebase, αλλά πρέπει ο δημιουργός να δημιουργήσει τους περιορισμούς. [64][65]

### 3.3.2. Σύγκριση με SQLite

Η SQLite είναι επίσης μια σχεσιασκή βάση δεδομένων (RDBMS) που έχει κάποια από τα χαρακτηριστικά SQL βάσης, αλλά με μια σημαντική διαφορά που έρχεται να γίνει κοινό σημείο με το Firebase, δεν απαιτεί την εγκατάσταση της στο σύστημα του δημιουργού. Δημιουργήθηκε το 2000 από τον Dwayne Richard Hipp και στην πραγματικότητα δεν ανήκει σε κάποια εταιρεία όπως το Firebase που ανήκει στην Google και η SQL στην Oracle. Εφόσον δεν απαιτεί εγκατάσταση, εμφανές είναι ότι δεν έχει κάποιον τοπικό server που πρέπει να τρέχει συνεχώς στο σύστημα. Το SQLite δεν είναι scalable, δηλαδή δεν μπορεί να χειριστεί με αποτελεσματικά περισσότερες εργασίες από ότι συνήθως χειρίζεται, οπότε δεν προτιμάται για μεγάλες εφαρμογές. [66]

### 3.3.3. Σύγκριση με Couchbase

Το Couchbase δημιουργήθηκε το 2011 από την Couchbase Inc. και είναι επίσης μια NoSQL βάση δεδομένων που διατηρεί τα δεδομένα της σε αρχείο της μορφής JSON. Περιέχει επίσης λειτουργίες για ασφάλεια δεδομένων και αυτόματο συγχρονισμό τους σε όλους τους χρήστες. Το Couchbase ταιριάζει πιο πολύ σε εφαρμογές με τεράστιο όγκο δεδομένων και με περίπλοκη επεξεργασία δεδομένων στην πλευρά του πελάτη (client-side). Επίσης, είναι απαραίτητη η εγκατάσταση και η διαχείριση του server και υπάρχουν προβλήματα ταχύτητα και βελτιστοποίησης. [67][68]

### 3.3.3. Σύγκριση με MongoDB

Το MongoDB είναι μια NoSQL βάση δεδομένων που δημιουργήθηκε και διαχειρίζεται από την MongoDB Inc. Η δομή της βάσης είναι ευέλικτη με ελάχιστους περιορισμούς, μπορεί να αποθηκεύσει δεδομένα με μηδενικό κόστος και θεωρείται πολύ ασφαλής. Χαρακτηρίζεται από τις επιδόσεις του σε εφαρμογές με πολλά αιτήματα διαβάσματος και γραψίματος. Ένα από τα αρνητικά είναι ότι η σύνταξη των queries είναι αρκετά πολύπλοκη. Ακόμα, υπάρχει μεγάλη καμπύλη εκμάθησης για προγραμματιστές που πρέπει να χειρίζονται εφαρμογές με πολλούς χρήστες, καθώς ο κώδικας μπορεί να γίνει αρκετά περίπλοκος και η διόρθωση του να είναι κάποιες φορές πάρα πολύ δύσκολη ή και ακόμα να μην μπορεί να εντοπιστεί ακριβώς το σημείο του προβλήματος. Τα αιτήματα στην βάση μπορούν να έχουν αρκετά περίπλοκη μορφή, ενώ στο Firebase είναι δομημένα με κατανοητό τρόπο χωρίς να πάρουν περίπλοκες μορφές. [69]

## Κεφάλαιο 4: Υλοποίηση Εφαρμογής HealthApp

### 4.1. Λεκτική παρουσίαση

Ο χρήστης, την πρώτη φορά που θα χρησιμοποιήσει την εφαρμογή, θα κληθεί να εισάγει το email και τον κωδικό του, χωρίς την δυνατότητα κάποιας εγγραφής. Αυτό συμβαίνει γιατί μόνο ο διαχειριστής μπορεί να δημιουργήσει λογαριασμούς για τον κάθε χρήστη (γιατρό ή νοσοκομειακό προσωπικό) και να ενημερώσει τους ενδιαφερόμενους για τα στοιχεία τους.

Με την επιτυχημένη εισαγωγή στο σύστημα, εμφανίζεται στον χρήστη η λίστα με τους διαθέσιμους ασθενείς που του έχουν ανατεθεί από τον διαχειριστή. Οι ασθενείς εισάγονται επίσης από τον διαχειριστή μέσω της πλατφόρμας Firebase, και συνδέονται με τον ή τους χρήστες που πρέπει. Ο χρήστης επιλέγει έναν ασθενή από την λίστα.

Μετά την επιλογή του ασθενούς, προβάλλονται στον οι λειτουργίες της εφαρμογής. Πρώτα του εμφανίζεται η καρτέλα “ΠΡΟΦΙΛ” που περιέχει τις βασικές πληροφορίες του ασθενούς, όπως:

- Όνομα
- Επώνυμο
- Πατρώνυμο
- Φύλο
- Η.Μ. Γέννησης
- Εισαγωγή
- Εθνικότητα
- Πόλη Διαμονής
- Διεύθυνση
- Τ.Κ.
- Τηλέφωνο 1
- Τηλέφωνο 2

Δεν υπάρχει η δυνατότητα τροποποίησης των πληροφοριών παρά μόνο η προβολή τους.

Επόμενη καρτέλα είναι οι “ΕΞΕΤΑΣΕΙΣ”. Εδώ εμφανίζονται όλες οι εξετάσεις που έχουν ανατεθεί από τους χρήστες, με πληροφορίες όπως το όνομα της εξέτασης και της ημερομηνίας που θα γίνει. Ο χρήστης μπορεί επίσης να πατήσει πάνω στην εξέταση για να προβάλει τα αποτελέσματα της αν αυτά είναι διαθέσιμα, αν δεν είναι θα του εμφανιστεί μήνυμα “Δεν υπάρχουν διαθέσιμα αποτελέσματα ακόμα”. Επί προσθέτως, ο χρήστης δημιουργεί νέα εξέταση που θέλει, εισάγοντας το όνομα και την ημερομηνία.

Η καρτέλα “ΦΑΡΜΑΚΑ” παρουσιάζει όλα τα φάρμακα που έχουν δημιουργηθεί με πληροφορίες όπως:

- Όνομα
- Δοσολογία
- Συχνότητα
- Γιατρός που το ανέθεσε
- Ημερομηνία που δόθηκε

Ο χρήστης έχει την δυνατότητα εισαγωγής νέου φαρμάκου εισάγοντας τα παραπάνω στοιχεία εκτός της ημερομηνίας και του ονόματος του που εισάγονται αυτόματα από την εφαρμογή. Επίσης, παρέχεται η δυνατότητα διαγραφής κάποιου φαρμάκου που θέλει ο χρήστης.

Στην καρτέλα “ΣΗΜΕΙΩΣΕΙΣ”, μπορεί ο κάθε χρήστης να εισάγει ένα κείμενο το οποίο προβληθεί μαζί με τις υπόλοιπες σημειώσεις. Ο χρήστης χρειάζεται μόνο να πληκτρολογήσει το κείμενο καθώς το όνομα του και η ημερομηνία εισάγονται αυτόματα. Μπορούν οι χρήστες να διαγράψουν σημειώσεις επίσης.

Η καρτέλα “ΙΣΤΟΡΙΚΟ” περιέχει όλες τις αλλαγές που έχουν γίνει στις προηγούμενες καρτέλες, προσθήκες και διαγραφές όλες παρουσιάζονται σε μια λίστα με τα ακόλουθα χαρακτηριστικά:

- Προσθήκη ή Αφαίρεση από πια καρτέλα
- Ποιος χρήστης το έκανε
- Ημερομηνία της πράξης

Επίσης, ο χρήστης μπορεί να πατήσει πάνω στο ιστορικό που επιθυμεί και θα προβληθούν οι λεπτομέρειες της προσθήκης ή της αφαίρεσης.

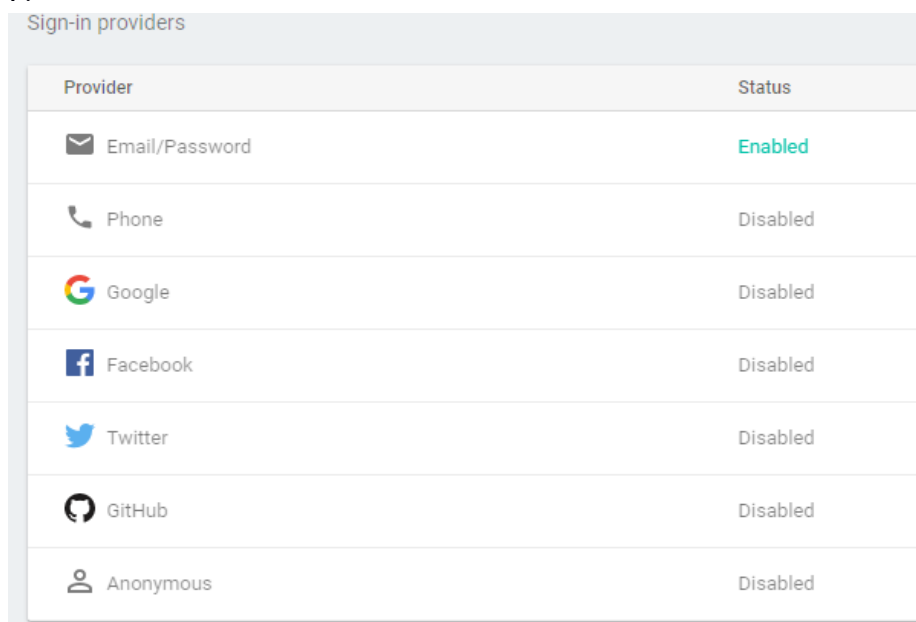
Τελευταία καρτέλα είναι το “CHAT”, εδώ μπορούν όλοι οι χρήστες να επικοινωνήσουν μεταξύ τους σε πραγματικό χρόνο ανταλλάζοντας μηνύματα. Εκτός από το μήνυμα, προβάλλεται το όνομα του αποστολέα και η ώρα που το έστειλε.








Σημαντικό να τονιστεί ότι όλες οι αλλαγές που κάνουν οι χρήστες εμφανίζονται σε όλους τους χρήστες. Όλα τα παραπάνω είναι αποθηκευμένα στην πλατφόρμα του Firebase με την χρήση των τριών χαρακτηριστικών (Αυθεντικοποίηση, Βάση Δεδομένων και Χώρος αποθήκευσης).

## 4.2. Πλατφόρμα Firebase

### 4.2.1. Αυθεντικοποίηση χρηστών

Ο διαχειριστής εισάγει τους χρήστες που μπορούν να έχουν πρόσβαση στην εφαρμογή στην πλατφόρμα του Firebase στην κονσόλα Αυθεντικοποίηση (Authentication). Αρχικά γίνεται επιλογή της μεθόδου εισαγωγής στο σύστημα, και στην συγκεκριμένη περίπτωση επιλέγεται μόνο η σύνδεση μέσω email και κωδικό.



Sign-in providers	
Provider	Status
 Email/Password	Enabled
 Phone	Disabled
 Google	Disabled
 Facebook	Disabled
 Twitter	Disabled
 GitHub	Disabled
 Anonymous	Disabled

Εικόνα 4.1: Μέθοδοι εισαγωγής των χρηστών

Ο διαχειριστής μπορεί να εισάγει και να τροποποιήσει χρήστες στην καρτέλα “USERS” της ίδιας κονσόλας. Παρουσιάζεται μια λίστα με όλους τους χρήστες περιέχοντας τα ακόλουθα στοιχεία ανά χρήστη:

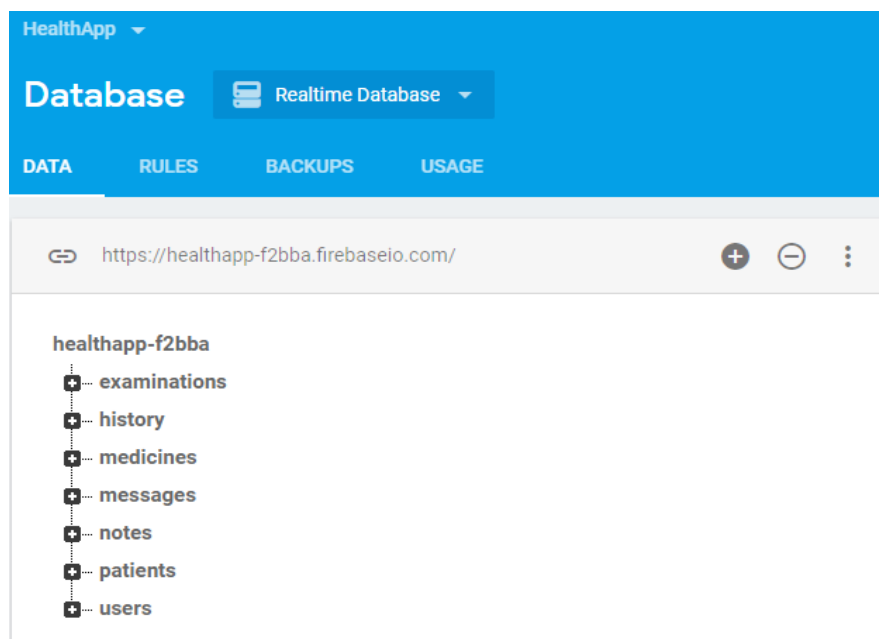
- Αναγνωριστικό (π.χ. Email του χρήστη)
- Πάροχος (π.χ. Email, τηλέφωνο)
- Ημερομηνία δημιουργίας
- Ημερομηνία τελευταία σύνδεση
- ID χρήστη (UID)

Όπως είναι λογικό, μόνο οι χρήστες που βρίσκονται σε αυτήν την λίστα μπορούν να συνδεθούν στην εφαρμογή με καμία άλλη εξαίρεση. Σε περίπτωση που κάποιος χρήστης θέλει να αλλάξει τον κωδικό του ή τον έχει ξεχάσει, επικοινωνεί με τον διαχειριστή και μέσω των επιλογών που διαθέτει το Firebase γίνονται οι ενέργειες για την επίλυση οποιουδήποτε προβλήματος.

#### 4.2.2. Βάση δεδομένων

Στην βάση δεδομένων περιέχονται όλα τα δεδομένα που θα χρησιμοποιηθούν στην εφαρμογή. Ο διαχειριστής έχει πρόσβαση στην βάση μέσω της κονσόλας “Database” που εκεί περιέχονται οι καρτέλες Data, Rules, Backups και Usage.

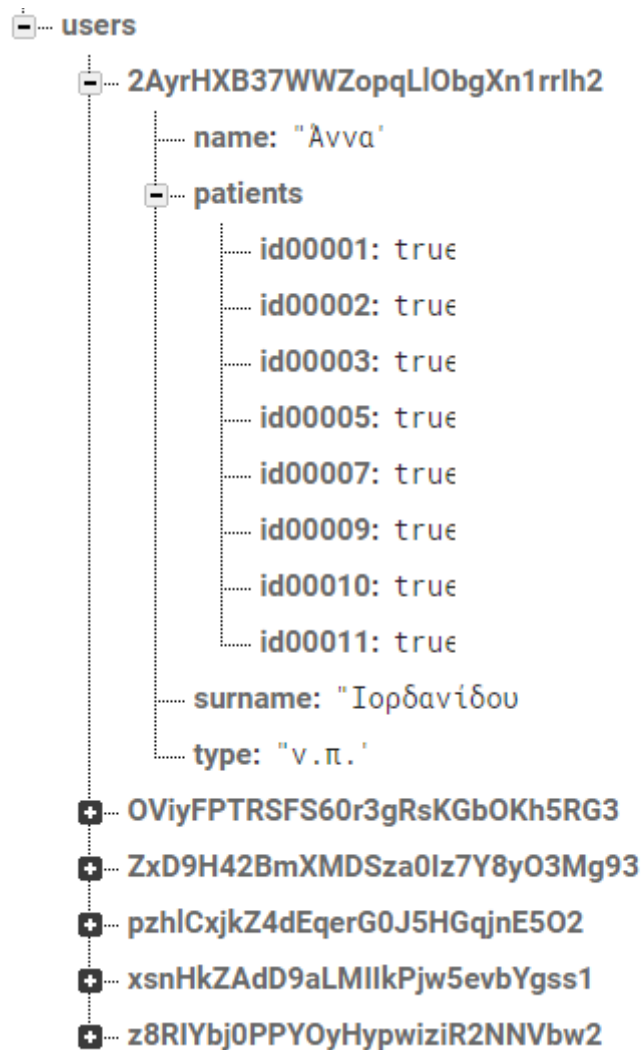
Στην καρτέλα Data, βρίσκονται όλα τα δεδομένα της εφαρμογής, είτε αυτά τα έχουν προστεθεί από τον διαχειριστή, είτε έχουν δημιουργηθεί και επεξεργαστεί από την εφαρμογή. Όπως είναι γνωστό, η βάση δεδομένων του Firebase είναι μια NoSQL βάση άρα δεν υπάρχει η έννοια των πινάκων όπως γνωρίζουμε από την SQL. Αντίθετα, η οργάνωση θυμίζει πολλά δέντρα με τις ρίζες του, κάθε “δέντρο” είναι η ομαδοποίηση των δεδομένων ανά κατηγορίες. Ο διαχειριστής μπορεί να προσθέσει, επεξεργαστεί και να διαγράψει τα δεδομένα μέσω του editor, αλλά μπορεί να εισάγει το δικό του αρχείο JSON το οποίο θα παρουσιαστεί στην βάση αν δεν έχει συντακτικά λάθη. Μπορεί επίσης ανά πάσα στιγμή να κάνει export ολόκληρη την βάση σε αρχείο JSON για ευκολότερη διαχείριση της βάσης. Η εφαρμογή, έχει επτά δέντρα που αντιστοιχούν στους χρήστες, ασθενείς, εξετάσεις, φάρμακα, σημειώσεις, ιστορικό και μηνύματα.



Εικόνα 4.2: Βάση δεδομένων

Το πεδίο “users” αντιστοιχεί στους χρήστες της εφαρμογής, εκείνους δηλαδή που έχουν λογαριασμό και μπορούν να συνδεθούν στην εφαρμογή (γιατροί και νοσοκομειακό προσωπικό). Το κλειδί κάθε χρήστη είναι το UID που τους αντιστοιχεί από την αυθεντικοποίηση (Authentication). Οι τιμές που περιέχονται είναι:

- όνομα
- επώνυμο
- τύπος (γιατρός ή νοσοκομειακό προσωπικό)
- ασθενείς που του αντιστοιχούν

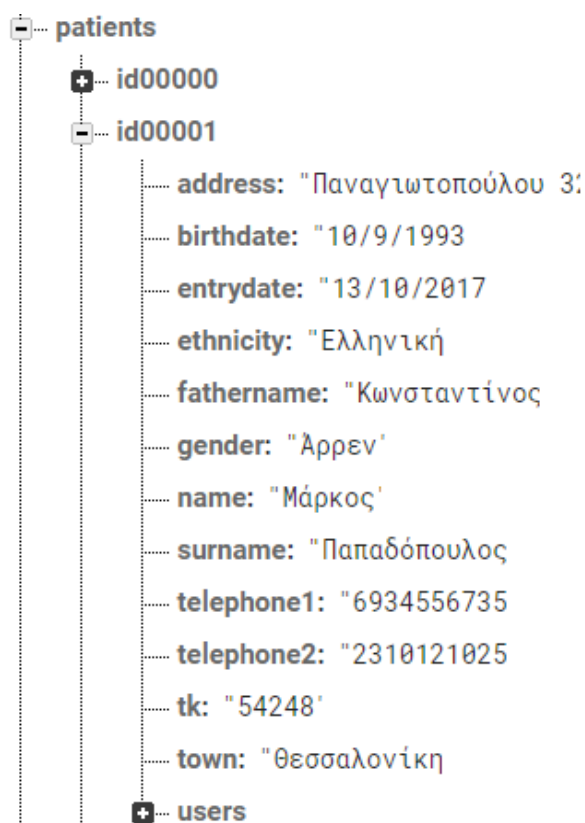


Εικόνα 4.3: Στοιχεία πεδίου “users”



Το πεδίο “patients” αντιστοιχεί στους ασθενείς του νοσοκομείου. Το κλειδί κάθε ασθενούς είναι ένα ID με τον μοναδικό τους αριθμό και περιέχει τις τιμές:

- όνομα
- επώνυμο
- πατρώνυμο
- φύλο
- ημερομηνία γέννησης
- διεύθυνση
- ημερομηνία εισαγωγής
- εθνικότητα
- πόλη
- διεύθυνση
- τ.κ.
- τηλέφωνο 1
- τηλέφωνο 2
- τους χρήστες που έχουν πρόσβαση στα δεδομένα του



Εικόνα 4.4: Στοιχεία πεδίου “patients”

Το πεδίο “examinations” αντιστοιχεί στις εξετάσεις του κάθε ασθενούς. Το κλειδί κάθε εξέτασης είναι ένα ID(ex) με τον μοναδικό τους αριθμό και περιέχει τις τιμές:

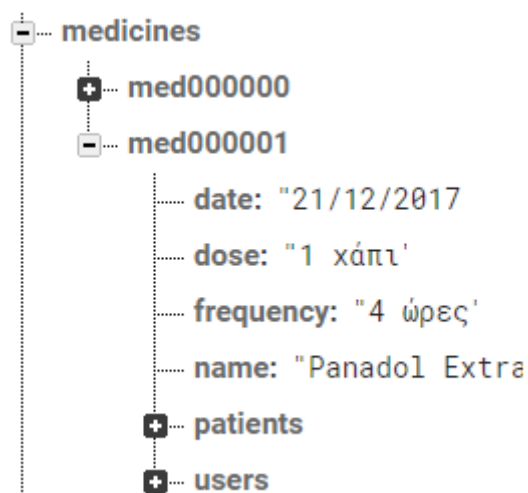
- όνομα
- ημερομηνία που θα πραγματοποιηθεί η εξέταση
- κλειδί ασθενούς
- κλειδί χρήστη



Εικόνα 4.5: Στοιχεία πεδίου "examinations"

Το πεδίο “medicines” αντιστοιχεί στα φάρμακα του κάθε ασθενούς. Το κλειδί κάθε εξέτασης είναι ένα ID(med) με τον μοναδικό τους αριθμό και περιέχει τις τιμές:

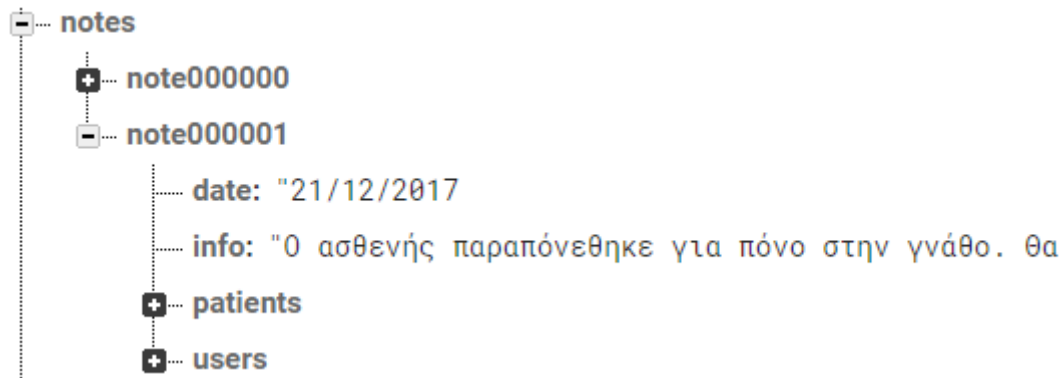
- όνομα
- δοσολογία
- συχνότητα
- ημερομηνία που δόθηκε το φάρμακο
- κλειδί ασθενούς
- κλειδί χρήστη



Εικόνα 4.6: Στοιχεία πεδίου "medicines"

Το πεδίο “notes” αντιστοιχεί στις σημειώσεις του κάθε ασθενούς που έχουν πληκτρολογηθεί από τον κάθε χρήστη. Το κλειδί κάθε εξέτασης είναι ένα ID(note) με τον μοναδικό τους αριθμό και περιέχει τις τιμές:

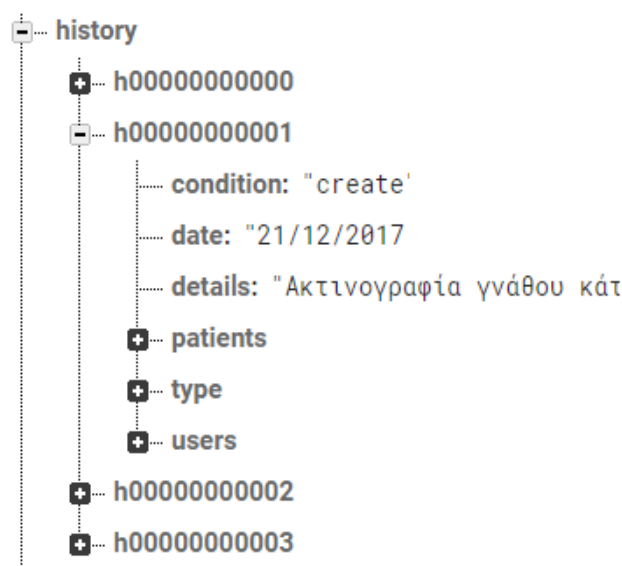
- κείμενο σημείωσης
- ημερομηνία που δημιουργήθηκε
- κλειδί ασθενούς
- κλειδί χρήστη



Εικόνα 4.7: Στοιχεία πεδίου "notes"

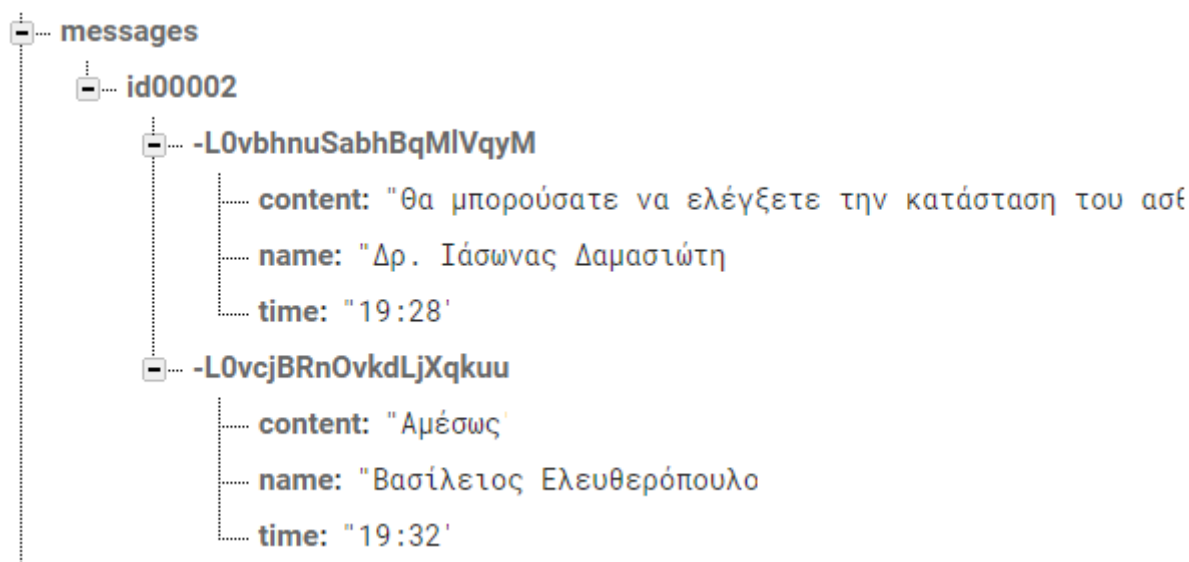
Το πεδίο “history” αντιστοιχεί στο ιστορικό κάθε ασθενούς, δηλαδή στις προσθήκες και αφαιρέσεις που έγιναν στα προηγούμενα. Το κλειδί κάθε εξέτασης είναι ένα ID(h) με τον μοναδικό τους αριθμό και περιέχει τις τιμές:

- είδος ιστορικού (προσθήκη-διαγραφή)
- ημερομηνία
- λεπτομέρειες
- κλειδί εγγραφής που προστέθηκε ή διαγράφηκε
- κλειδί ασθενούς
- κλειδί χρήστη



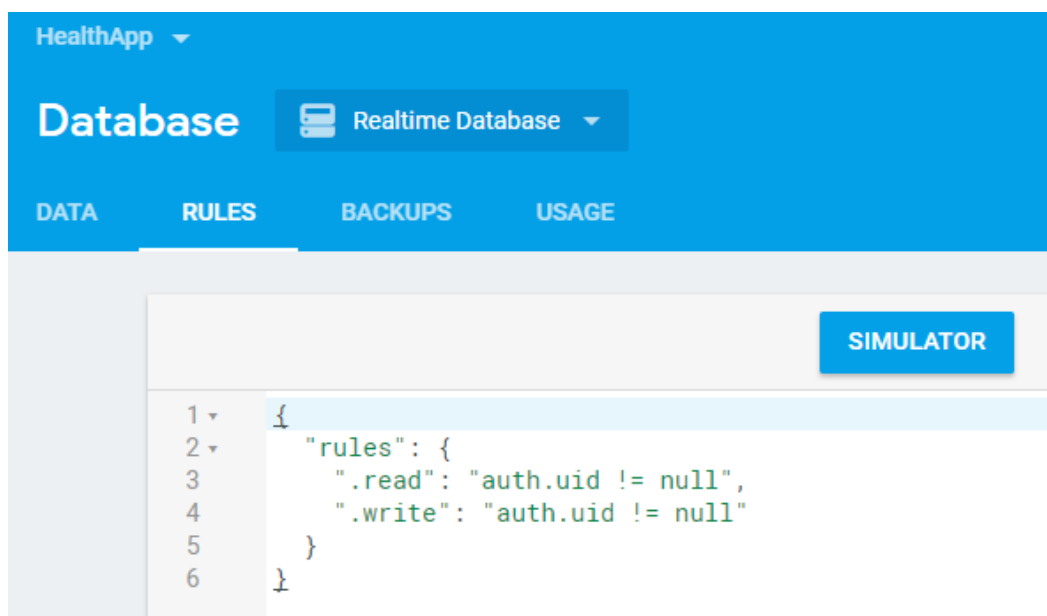
Εικόνα 4.8: Στοιχεία πεδίου "history"

Το πεδίο “messages” αντιστοιχεί στα άμεσα μηνύματα(chat) που ανταλλάχθηκαν για τον ασθενή. Τα μηνύματα έχουν ομαδοποιηθεί με βάση τον ασθενή, άρα το κλειδί των ομαδοποιημένων μηνυμάτων είναι το κλειδί του ασθενούς στο πεδίο patients και κάθε τιμή περιέχει τα κλειδιά του κάθε μηνύματος, και αυτά με την σειρά τους περιέχουν σαν τιμές το μήνυμα, όνομα αποστολέα και ώρα που στάλθηκε το μήνυμα.



Εικόνα 4.8: Στοιχεία πεδίου “messages”

Όπως τονίστηκε, η πλατφόρμα του Firebase προσφέρει στον διαχειριστή την ελευθερία να δώσει δικαιώματα πρόσβασης σε όποιους χρήστες επιθυμεί, αυτές τις ρυθμίσεις μπορεί να τις αλλάξει στην καρτέλα Rules. Στην περίπτωση της εργασίας, δίνονται δικαιώματα μόνο στους χρήστες που είναι επιβεβαιωμένοι και αυτό γίνεται με τον παρακάτω κώδικα:

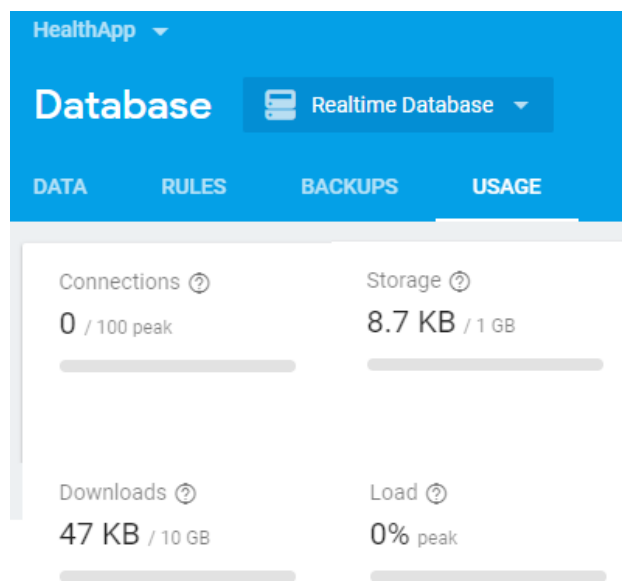


Εικόνα 4.9: Κανόνες βάσεις

- Το “.read” αφορά την ανάγνωση των δεδομένων της βάσης
- Το “.write” αφορά την προσθήκη, επεξεργασία και διαγραφή των δεδομένων της βάσης

Με την εντολή “auth.uid” παίρνουμε το κλειδί του κάθε αυθεντικοποιημένου χρήστη, αν αυτό δεν υπάρχει, δηλαδή είναι null, τότε ο χρήστης δεν έχει καμία πρόσβαση στην βάση. Μπορούν να προστεθούν και άλλοι περιορισμοί για συγκεκριμένα πεδία της βάσης και γενικά υπάρχουν πολλές επιλογές που έχει ο διαχειριστής. Στην περίπτωση της εργασίας όμως είναι αρκετός ο περιορισμός αυτός. Θα υπήρχε ανάγκη για προσθήκη περισσότερων μηχανισμών, αν υπήρχε περίπτωση να έχουν πρόσβαση στην εφαρμογή και τρίτοι χρήστες.

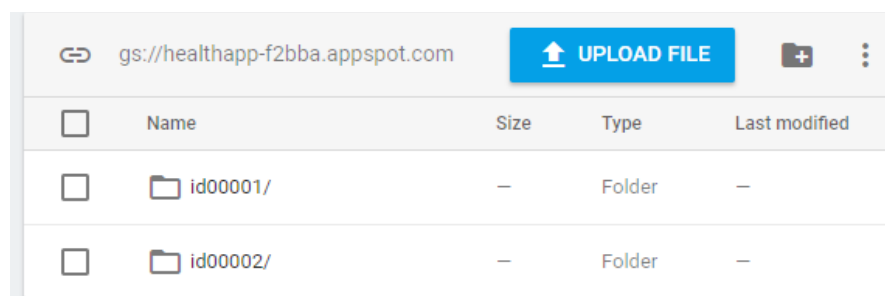
Στην καρτέλα Usage, βλέπουμε πόσες συνδέσεις έχουν γίνει ταυτόχρονα στο σύστημα, πόσο χώρο πίνουν τα δεδομένα της βάσης, πόσα κατεβάσματα έχουν γίνει για αυτά τα δεδομένα και πόσα αιτήματα βρίσκονται σε αναμονή για την ολοκλήρωσή τους.



Εικόνα 4.10: Καρτέλα USAGE

### 4.2.3. Αποθηκευτικός χώρος

Στην εφαρμογή γίνεται χρήση του αποθηκευτικού χώρου του Firebase για την αποθήκευση και την προβολή των διαθέσιμων εξετάσεων. Ο αποθηκευτικός χώρος περιέχει φακέλους με το μοναδικό κλειδί του κάθε ασθενούς και μέσα στους φακέλους αυτούς περιέχονται ή διαθέσιμες εξετάσεις. Οι εξετάσεις είναι σε μορφή εικόνας και το όνομα τους περιέχει το κλειδί της κάθε εξέτασης. Έτσι, όταν είναι διαθέσιμα τα αποτελέσματα, ο χρήστης μπορεί να τα προβάλει στην εφαρμογή.



Εικόνα 4.11: Φάκελοι αποθηκευτικού χώρου

<a href="#">gs://healthapp-f2bba.appspot.com</a> > id00001 <div> <b>UPLOAD FILE</b> </div>				
<input type="checkbox"/>	Name	Size	Type	Last modified
<input type="checkbox"/>	ex000001.jpg	39.67 KB	image/jpeg	Nov 29, 2017
<input type="checkbox"/>	ex000002.jpg	91.45 KB	image/jpeg	Dec 5, 2017

Εικόνα 4.12: Αρχεία υποφάκελου

<a href="#">gs://healthapp-f2bba.appspot.com</a> > id00001 <div> <b>UPLOAD FILE</b> </div>				
<input type="checkbox"/>	Name	Size	Type	Last modified
<input type="checkbox"/>	ex000001.jpg	39.6...	image/jpeg	Nov 29, 2...
<input type="checkbox"/>	ex000002.jpg	91.4...	image/jpeg	Dec 5, 2017

**ex000001.jpg** ×

Name  
ex000001.jpg

Size  
39.67 KB

Type  
image/jpeg

Created  
Nov 29, 2017, 4:51:49 PM

Updated  
Nov 29, 2017, 4:51:49 PM

File location ▼

Other metadata ▼

Εικόνα 4.13: Περιεχόμενα αρχείου

### 4.3. Παρουσίαση εφαρμογής

Η εφαρμογή {a} υλοποιήθηκε με την χρήση του Android Studio 3.0 και με την γλώσσα προγραμματισμού Java. Επίσης σαν emulator για δοκιμές χρησιμοποιήθηκε το Genymotion και τα στιγμιότυπα που θα προβληθούν στην παρουσίαση, έχουν παρθεί από τον emulator της συσκευής Google Nexus 5x. Στην ενότητα αυτή θα παρουσιαστούν όλες οι οθόνες που θα συναντήσει ο χρήστης κατά την χρήση της εφαρμογής και σε κάθε οθόνη θα παρουσιάζονται τα εξής:

- Στιγμιότυπο οθόνης (εικόνα)
- Ανάλυση λειτουργικότητας
- Κλάσεις Java που την υλοποιούν
- Σημαντικά κομμάτια κώδικα καθώς ο πλήρης πηγαίος κώδικας της εφαρμογής βρίσκεται στο παράρτημα

Οι κλάσεις που περιέχονται στον φάκελο java, έχουν χωριστεί σε κατηγορίες:

- Activities – είναι οι δραστηριότητες της εφαρμογής δηλαδή οι οθόνες.
- NavDrawer – Navigation Drawer, είναι ένα πάνελ που περιέχει επιλογές για την εφαρμογή στα αριστερά της οθόνης. Είναι κρυμμένο το πάνελ αυτό, και ο χρήστης για να έχει πρόσβαση στις επιλογές, πατάει πάνω στο χαρακτηριστικό εικονίδιο.
- Fragments – στην εφαρμογή έχουν χρησιμοποιηθεί fragments τα οποία περιέχονται στις αρχικές οθόνες και παρουσιάζουν δεδομένα όπως τα φάρμακα του ασθενούς.
- Adapters – διαχείριση των δεδομένων κάθε activity και fragment, σε συνδυασμό με τις αντίστοιχες κλάσεις τους.
- POJOs – περιέχονται όλες οι κλάσεις που αντιστοιχούν στα στοιχεία κάθε οντότητας που θα χρησιμοποιηθεί από την εφαρμογή περιέχοντας τις μεταβλητές τους και τους getters και setters.
- Στον φάκελο drawable που βρίσκεται στον φάκελο res, περιέχονται όλες οι εικόνες, εικονίδια και αρχεία XML που περιέχουν τα σχήματα που χρησιμοποιήθηκαν στην εφαρμογή.
- Στον φάκελο layout, περιέχονται όλα τα layouts των activities, fragments, navigation drawer και λιστών της εφαρμογής.
- Στον φάκελο menu περιέχονται όλα τα XML αρχεία που χρησιμοποιήθηκαν στο πάνελ του NavDrawer.
- Στον φάκελο mipmap περιέχεται το εικονίδιο της εφαρμογής σε διάφορες αναλύσεις για διαφορετικές συσκευές.
- Στον φάκελο values περιέχονται όλα τα δεδομένα που χρησιμοποιήθηκαν στα layouts, όπως συμβολοσειρές (strings.xml), διαστάσεις για διάφορα γραφήματα (dimens.xml), χρώματα(colors.xml), το στυλ του θέματος της εφαρμογής όπως το είδος (styles.xml) και ένα αρχείο (drawables.xml) για κάποια έτοιμα εικονίδια που χρησιμοποιεί η εφαρμογή.

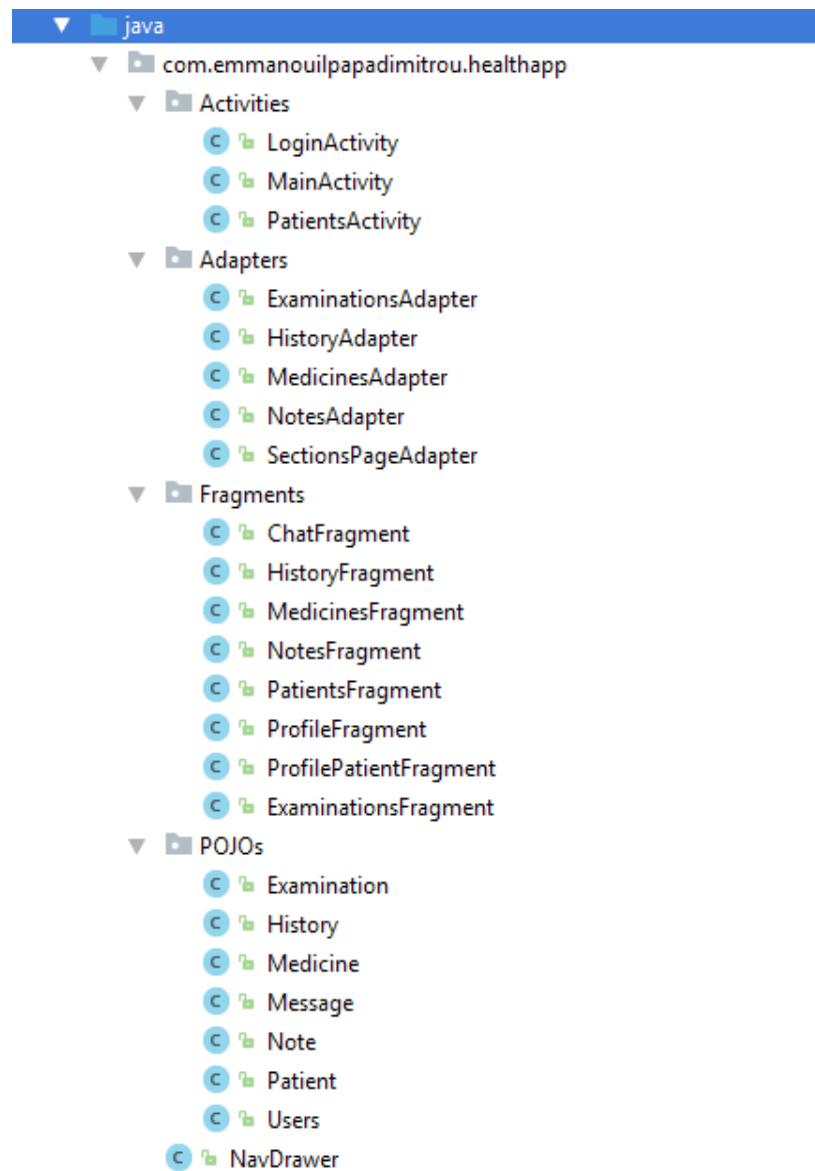
{b}

Στο build.gradle (επίπεδο project) περιέχονται οι δηλώσεις για τις υπηρεσίες της google.

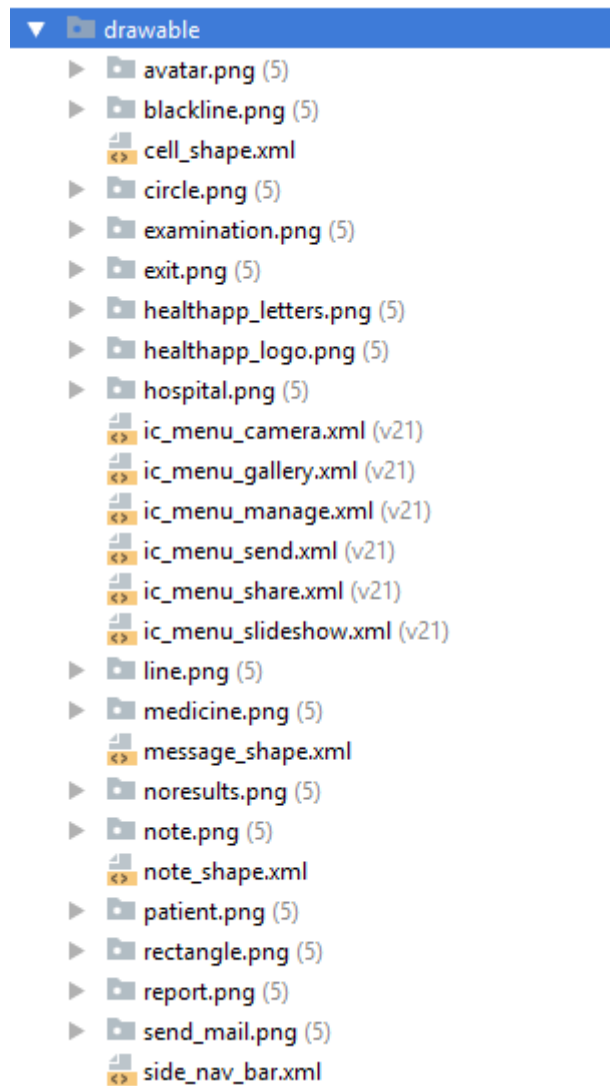
Στο build.gradle (επίπεδο app) περιέχονται οι εξαρτήσεις(dependencies) της εφαρμογής για τις βιβλιοθήκες που είναι αναγκαίο να χρησιμοποιηθούν, όπως οι βιβλιοθήκες του Firebase.

Τέλος, στο αρχείο AndroidManifest περιέχεται το όνομα του πακέτου της εφαρμογής, το όνομα, οι δραστηριότητες, στυλ, εικονίδιο, όνομα εφαρμογής και η άδεια χρήσης του internet.

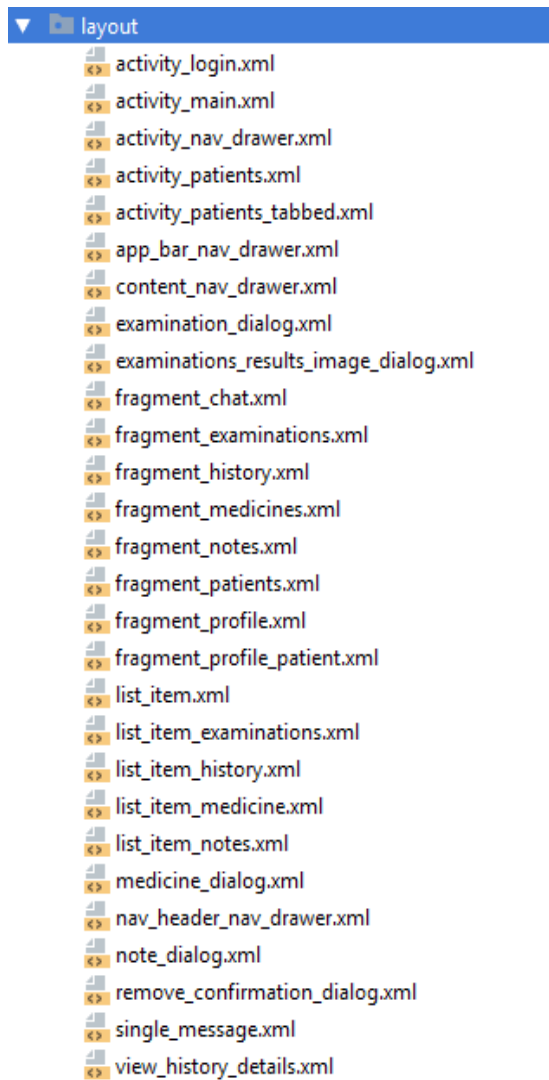




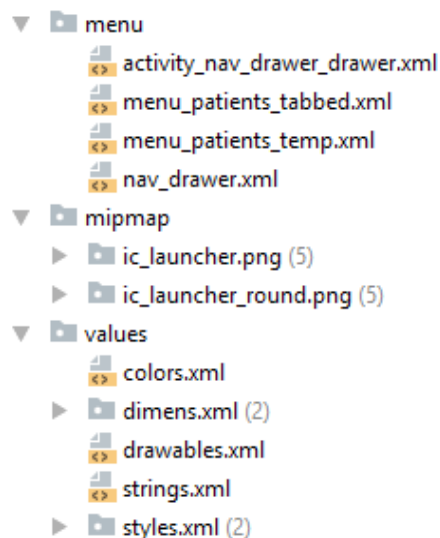
Εικόνα 4.14: Java κλάσεις της εφαρμογής



Εικόνα 4.15: Φάκελος *drawable*



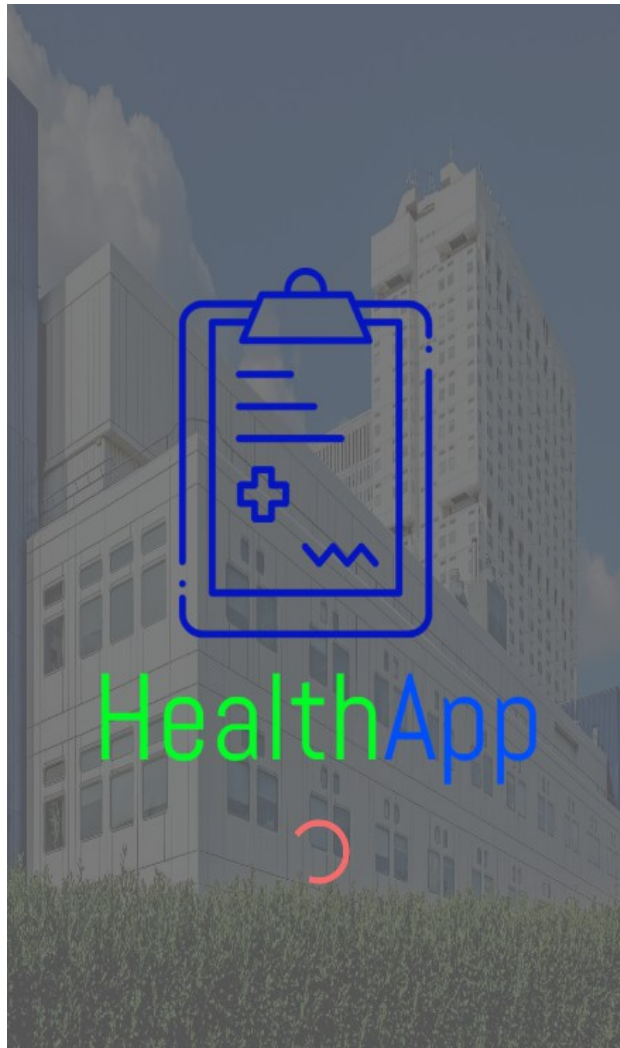
Εικόνα 4.16: Φάκελος layout



Εικόνα 4.17: Φάκελος menu, mipmap και values

### 4.3.1. Φόρτωση εφαρμογής

Η πρώτη οθόνη που αντικρίζει ο χρήστης είναι η φόρτωση της εφαρμογής. {c}



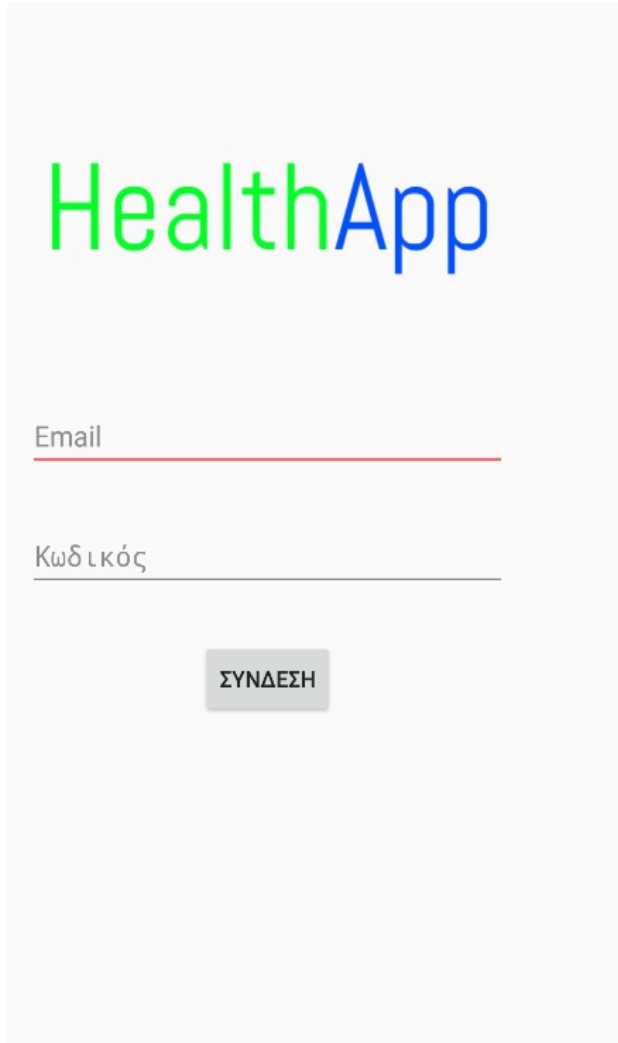
Εικόνα 4.18: Φόρτωση εφαρμογής

Αυτή η οθόνη υλοποιείται στην κεντρική activity(δραστηριότητα) της εφαρμογής `MainActivity`. Στην κλάση αυτή χρησιμοποιείται ένας `Handler` για την παρουσίαση της οθόνης αυτής για συγκεκριμένο χρονικό διάστημα, και στην συνέχεια ανακατευθύνει στην δραστηριότητα `LoginActivity` με την χρήση `Intent`. Χρησιμοποιεί το layout `activity_main`.

```
new Handler().postDelayed(new Runnable() {  
    @Override  
    public void run() {  
        Intent loginIntent = new Intent(MainActivity.this, LoginActivity.class);  
        finish();  
        startActivity(loginIntent);  
    }  
}, SPLASH_TIME_OUT);
```

### 4.3.2. Σύνδεση χρήστη

Στην ανακατεύθυνση, ο χρήστης οδηγείται στην οθόνη για της σύνδεσης του στο σύστημα. Εδώ πρέπει να εισάγει το email και τον κωδικό που του έχουν δοθεί από τον διαχειριστή. Σε περίπτωση που τα εισάγει λάθος, του εμφανίζεται μήνυμα ότι κάποιο από τα πεδία είναι λάθος.



Εικόνα 4.19: Οθόνη σύνδεσης του χρήστη

Τα στοιχεία που εισάγει ο χρήστης, ελέγχονται με την αυθεντικοποίηση της πλατφόρμας του Firebase από το πεδίο Authentication. Η οθόνη αυτή υλοποιείται στην κλάση `LoginActivity` και χρησιμοποιεί το layout `activity_login`.

```
firebaseAuth = FirebaseAuth.getInstance();
if(firebaseAuth.getCurrentUser() != null){
    finish();
    startActivity(new Intent(getApplicationContext(), NavDrawer.class));
}
```

Η μεταβλητή `firebaseAuth` περιέχει όλα τα στοιχεία του χρήστη που είναι αυθεντικοποιημένος. Στην συνέχεια γίνεται έλεγχος του `firebaseAuth.getCurrentUser()` με το `null` έτσι ώστε άμα έχει κάνει προηγουμένως σύνδεση στο σύστημα να γίνει ανακατεύθυνση στο κύριο μέρος της εφαρμογής, αν ισούται με `null` σημαίνει ότι ο χρήστης δεν έχει κάνει σύνδεση ή αποσυνδέθηκε.

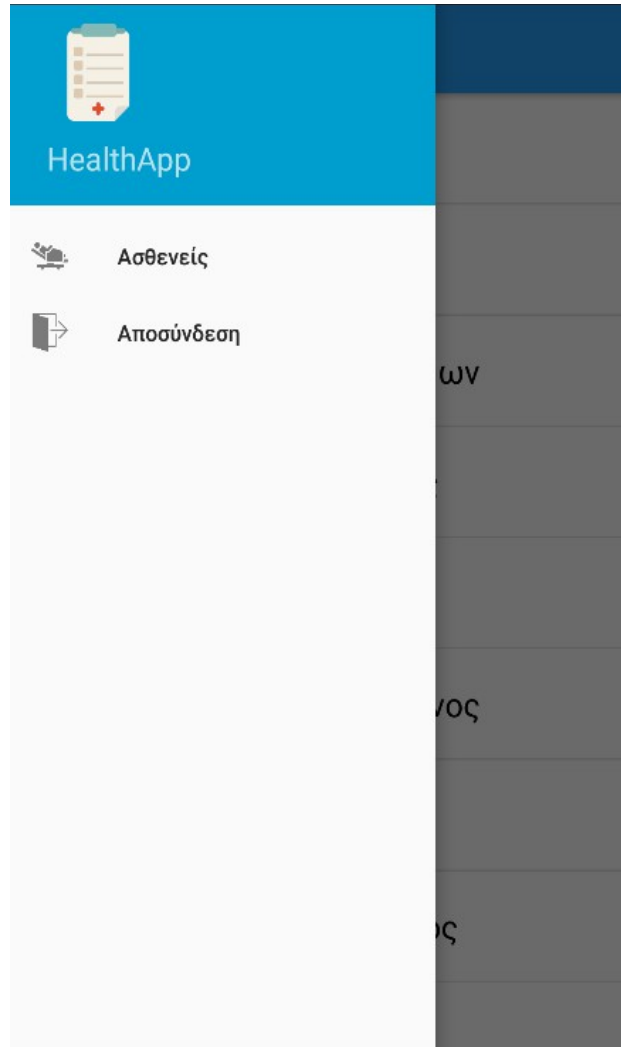
```
//Το firebase δίνει δυνατότητα για άμεση διαχείριση του email και του κωδικού του
//χρήστη στην σύνδεση στο σύστημα
firebaseAuth.signInWithEmailAndPassword(email,password).addOnCompleteListener(this, new
OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {
        //Παίρνουμε τα στοιχεία του χρήστη από την βάση για να κάνουμε ελέγχους
        FirebaseUser user = firebaseAuth.getCurrentUser();
        //Κλείνουμε το παράθυρο για το progress
        progressDialog.dismiss();
        //Αν η σύνδεση είναι επιτυχής
        if(task.isSuccessful()){
            //Κλείνει το τωρινό activity
            finish();
            //Και ανοίγει το βασικό activity της εφαρμογής
            startActivity(new Intent(getApplicationContext(),NavDrawer.class));
        }
        //Αν η σύνδεση δεν ήταν επιτυχής τα παρακάτω catches θα μας εμφανίσουν το
        //είδος του προβλήματος
        else{
            try {
                throw task.getException();
            } catch (FirebaseAuthWeakPasswordException e) {
                editTextPassword.setError(getString(R.string.error_weak_password));
                editTextPassword.requestFocus();
            } catch (FirebaseAuthInvalidCredentialsException e) {
                editTextEmail.setError(getString(R.string.error_data));
                editTextEmail.requestFocus();
            } catch (FirebaseAuthInvalidUserException e) {
                editTextEmail.setError(getString(R.string.error_data));
                editTextEmail.requestFocus();
            } catch (Exception e) {
                Log.e("TAG", e.getMessage());
            }
        }
    }
});
```

Με την συνάρτηση `firebaseAuth.signInWithEmailAndPassword` περνάμε σαν ορίσματα το email και τον κωδικό που έδωσε ο χρήστης ώστε να ψάξει η συνάρτηση μέσα στο Authentication για το αν ο χρήστης υπάρχει και αν είναι σωστά τα στοιχεία ή αντιστοιχούν σε χρήστη τότε γίνεται ανακατεύθυνση στην κλάση `NavDrawer`, αν δεν υπάρχει ή έχει δώσει λανθασμένα στοιχεία τότε του εμφανίζεται μήνυμα αποτυχημένης προσπάθειας.

### 4.3.3. Navigation Drawer

Στην οθόνη του Navigation Drawer που το υλοποιεί η κλάση `NavDrawer`, εμφανίζεται στον χρήστη η λίστα με τους διαθέσιμους ασθενείς του που το υλοποιεί το `fragment PatientsFragment`. Πάνω αριστερά ο χρήστης μπορεί να πατήσει το εικονίδιο για να του εμφανιστούν οι επιλογές που έχει το menu. Αυτές είναι η προβολή λίστας ασθενών και η αποσύνδεση από το σύστημα.

{e}



Εικόνα 4.20: Επιλογές του menu

Το Firebase παρέχει την συνάρτηση `signOut()` για την αποσύνδεση του χρήστη από το σύστημα και στην συνέχεια γίνεται ανακατεύθυνση στην κλάση `MainActivity` ώστε να συνδεθεί ξανά.

```
FirebaseAuth.getInstance().signOut();  
Intent loginIntent = new Intent(this, MainActivity.class);  
startActivity(loginIntent);  
finish();
```



Όπως αναφέρθηκε, ο χρήστης βλέπει απευθείας την λίστα των ασθενών που του αντιστοιχούν.

☰	Ασθενείς
■	Αγγελίδης Δημήτριος
■	Δερμάτης Ιωάννης
■	Δημητρίου Παντελεήμων
■	Κυπραίος Χαράλαμπος
■	Μαριτίδου Ιωάννα
■	Μπαλάνος Κωνσταντίνος
■	Παπαδήμου Αγαθή
■	Παπαδόπουλος Μάρκος

Εικόνα 4.21: Λίστα διαθέσιμων ασθενών

Η υλοποίηση της λειτουργικότητας αυτής συμβαίνει στο `PatientsFragment {f}`. Είναι η πρώτη φορά που χρησιμοποιείται στον κώδικα η βάση δεδομένων, η χρήση της γίνεται με την μεταβλητή και `referenceDB` μέσω αυτής μπορούμε να περιηγηθούμε σε όλα τα δεδομένα της. Το layout του fragment είναι `fragment_patients`. Ο ορισμός της βάσης δεδομένων γίνεται με τον παρακάτω τρόπο.

```
referenceDB = FirebaseDatabase.getInstance().getReference();
```

Γίνεται κλήση στην βάση με την μεταβλητή `referenceDB` στο πεδίο `patients` ώστε να παρθούν όλοι οι ασθενείς που βρίσκονται στην βάση δεδομένων και στην συνέχεια να ελεγχούν αν αντιστοιχούν στον χρήστη που είναι συνδεδεμένος, γίνονται σύγκριση των κλειδών με την χρήση της παρακάτω συνάρτησης.

```
firebaseAuth.getCurrentUser().getUid()
```

Γίνεται χρήση `HashMap` για εύκολη προσθήκη πολλών στοιχείων στην λίστα αντί για ένα. Η κλήση στο `patients` πεδίο γίνεται κάνοντας χρήση listener. Σημαντικό να τονιστεί ότι όλες οι κλήσεις στην βάση, είναι ασύγχρονες, άρα η υλοποίηση των απαραίτητων λειτουργιών γίνεται μέσα στα πλαίσια της κάθε κλήσης και ποτέ έξω από αυτήν γιατί ο κώδικας ολοκληρώνεται χωρίς να έχει γίνει η λήψη των αποτελεσμάτων.

```
//Παίρνουμε όλους τους ασθενείς από την βάση
referenceDB.child("patients").addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        for(DataSnapshot patient : dataSnapshot.getChildren()){
            Patient p = new Patient();
            p.setId(String.valueOf(patient.getKey()));
            p.setName(String.valueOf( patient.child("name").getValue()));
            p.setSurname(String.valueOf( patient.child("surname").getValue()));
            p.setFathurname(String.valueOf( patient.child("fathurname").getValue()));
            p.setGender(String.valueOf( patient.child("gender").getValue()));
            p.setBirthdate(String.valueOf( patient.child("birthdate").getValue()));
            p.setEntrydate(String.valueOf( patient.child("entrydate").getValue()));
            p.setEthnicity(String.valueOf( patient.child("ethnicity").getValue()));
            p.setTown(String.valueOf( patient.child("town").getValue()));
            p.setAddress(String.valueOf( patient.child("address").getValue()));
            p.setTk(String.valueOf( patient.child("tk").getValue()));
            p.setTelephone1(String.valueOf( patient.child("telephone1").getValue()));
            p.setTelephone2(String.valueOf( patient.child("telephone2").getValue()));
            allPatients.add(p);
            for(DataSnapshot patientChild : patient.child("users").getChildren()){
                if(firebaseAuth.getCurrentUser().getUid().equals(String.valueOf( patientChild.getKey()))){
                    patientInfoForList.put(p.getSurname() + " " + p.getName(),p.getId());
                }
            }
        }
    }
});
```

Με την χρήση της κλάσης `Patient {g}`, φτιάχνουμε ένα αντικείμενο που θα περιέχει όλα τα δεδομένα του κάθε χρήστη και στην συνέχεια βλέπουμε αν περιέχεται το κλειδί του χρήστη στην λίστα των κλειδιών του ασθενούς από την βάση. Η `Patient` κλάση έχει τις παρακάτω μεταβλητές:

```
private String id;
private String name;
private String surname;
private String fathurname;
private String gender;
private String birthdate;
private String entrydate;
private String ethnicity;
private String town;
private String address;
private String tk;
private String telephone1;
private String telephone2;
```

Στην συνέχεια, γίνεται αναγνώριση ποιον ασθενή επέλεξε ο χρήστης από την λίστα και μεταφέρεται το αντικείμενο της κλάσης `Patient` μέσω της συνάρτησης `intent.putExtra` του `intent` για την ανακατεύθυνση στο κύριο μέρος της εφαρμογής. Αυτό γίνεται για να ξέρει η επόμενη οθόνη ποιος είναι ο ασθενής.

```
userID = firebaseAuth.getCurrentUser().getUid();
referenceDB.child("users").addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        for(DataSnapshot doctorTemp : dataSnapshot.getChildren()){
            if(userID.equals(String.valueOf(doctorTemp.getKey()))){
                currentUser = new
Users(String.valueOf(doctorTemp.child("name").getValue()),
        String.valueOf(doctorTemp.child("surname").getValue()),
        String.valueOf(doctorTemp.child("type").getValue()));
        //Στέλνουμε την κλάση του ασθενούς στο PatientsActivity για χρήση των
στοιχείων του
                intent.putExtra("Patient", tempPatient);
                intent.putExtra("User", currentUser);
                break;
            }
        }
        //Εκκινάει το activity που αφορά το κύριο μέρος της εφαρμογής για κάθε ασθενή
        //Στέλνοντας τα απαραίτητα στοιχεία για διαχείριση
        startActivity(intent);
    }
    @Override
    public void onCancelled(DatabaseError databaseError) {
    }
});
```

Γίνεται κλήση στην βάση στο πεδίο ώστε να παρθεί το όνομα και το επώνυμο του χρήστη και να δημιουργηθεί νέο αντικείμενο της κλάσης `Users {h}` και να σταλθεί στην επόμενη οθόνη ώστε να μην χρειαστεί ξανά η κλήση στην βάση για να γίνει εύρεση του ονόματος του χρήστη. Η κλάση `Users` περιέχει τις παρακάτω μεταβλητές

```
public String name;
public String surname;
public String type;
```

- Όνομα χρήστη
- Επώνυμο χρήστη
- Τύπος χρήστη, αν είναι γιατρός ή νοσοκομειακό προσωπικό

#### 4.3.4. PatientsActivity

Η ανακατεύθυνση γίνεται στην δραστηριότητα `PatientsActivity {i}`. Σε αυτήν την οθόνη, γίνεται ο ορισμός των fragments που θα χρησιμοποιηθούν μαζί με τους τίτλους τους και ο τρόπος που θα παρουσιαστούν στην οθόνη αυτή. Τα fragments παρουσιάζονται ως καρτέλες και ο χρήστης μπορεί να επιλέξει όποια θέλει και να σύρει την μπάρα με τις καρτέλες, δεξιά ή αριστερά ανάλογα ποια καρτέλα θέλει που δεν φαίνεται στην οθόνη την συγκεκριμένη στιγμή. Η δραστηριότητα χρησιμοποιεί το layout `activity_patients_tabbed`. Για την παρουσίαση των

fragments, χρησιμοποιείται ένας πίνακας που περιέχει έναν `ViewPager`. Στο `onCreate` γίνεται ο ορισμός των αντικειμένων που θα χρησιμοποιηθούν για την παρουσίαση των fragments.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_patients_tabbed);
    mSectionsPagerAdapter = new SectionsPagerAdapter(getSupportFragmentManager());

    mViewPager = (ViewPager) findViewById(R.id.container);
    setupViewPager(mViewPager);
    TabLayout tabLayout = (TabLayout) findViewById(R.id.tabs);
    tabLayout.setupWithViewPager(mViewPager);
}

public void setupViewPager(ViewPager viewPager){
    SectionsPagerAdapter adapter = new
SectionsPagerAdapter(getSupportFragmentManager());
    adapter.addFragment(new ProfilePatientFragment(), "Προφίλ");
    adapter.addFragment(new ExaminationsFragment(), "Εξετάσεις");
    adapter.addFragment(new MedicinesFragment(), "Φάρμακα");
    adapter.addFragment(new NotesFragment(), "Σημειώσεις");
    adapter.addFragment(new HistoryFragment(), "Ιστορικό");
    adapter.addFragment(new ChatFragment(), "Chat");
    viewPager.setAdapter(adapter);
}
```

{}

Επίσης υπάρχουν τρεις συναρτήσεις που χρησιμοποιούνται από τα fragments

`getCurrentPatient` Επιστρέφει το αντικείμενο του ασθενούς που περιέχει τις πληροφορίες του.

```
public Patient getCurrentPatient(){
    patient = getIntent().getExtras().getParcelable("Patient");
    return patient;
}
```

`getCurrentUser` Επιστρέφει το αντικείμενο του χρήστη που περιέχει τις πληροφορίες του.

```
public Users getCurrentUser(){
    //Παίρνουμε την κλάση του ασθενούς που τέθηκε από το PatiensFragment
    currentUser = getIntent().getExtras().getParcelable("User");
    return currentUser;
}
```

`reload` Λειτουργεί ως ανανέωση της εφαρμογής για την προβολή των δεδομένων μετά από αλλαγές.

```
public void reload(){
    mViewPager = (ViewPager) findViewById(R.id.container);
    setupViewPager(mViewPager);
    TabLayout tabLayout = (TabLayout) findViewById(R.id.tabs);
    tabLayout.setupWithViewPager(mViewPager);
    setupViewPager(mViewPager);
}
```

#### 4.3.5. Προφίλ

Το πρώτο fragment που βλέπει ο χρήστης με το που επιλέξει τον ασθενή είναι το Προφίλ. Σε αυτό το fragment, γίνεται παρουσίαση των βασικών πληροφοριών του ασθενούς σε μορφή πίνακα. Η κλάση που το υλοποιεί αυτό το fragment είναι η `ProfilePatientFragment` {k} και το layout που χρησιμοποιεί είναι `fragment_profile_patient`.

ΠΡΟΦΙΛ   ΕΞΕΤΑΣΕΙΣ   ΦΑΡΜΑΚΑ   ΣΗΜΕΙΩΣΕΙΣ   ΙΣΤΟΡΙΑ	
Όνομα	Δημήτριος
Επώνυμο	Αγγελίδης
Πατρώνυμο	Γεώργιος
Φύλο	Άρρεν
Η.Μ. Γέννησης	12/5/1971
Εισαγωγή	2/11/2017
Εθνικότητα	Ελληνική
Πόλη Διαμονής	Αθήνα
Διεύθυνση	Αιόλου 32
T.K.	10437
Τηλέφωνο 1	6973335517
Τηλέφωνο 2	2102735650

Εικόνα 4.22: Προφίλ ασθενούς

Αρχικά, γίνεται δημιουργία ενός αντικειμένου της κλάσης `Patient` και μετά γίνεται κλήση της συνάρτησης `getCurrentPatient` της δραστηριότητας `PatientsActivity` ώστε να αντιστοιχηθούν τα δεδομένα. Στην συνέχεια γίνεται σύνδεση των μεταβλητών `TextView` με τα γραφικά στοιχεία του layout και τέλος γίνεται η προβολή των αντίστοιχων στοιχείων.

```
Patient patient = ((PatientsActivity) getActivity()).getCurrentPatient();

(παράλειψη της σύνδεσης των μεταβλητών με το layout)

name_input.setText(patient.getName());
surname_input.setText(patient.getSurname());
fathername_input.setText(patient.getFathername());
gender_input.setText(patient.getGender());
```

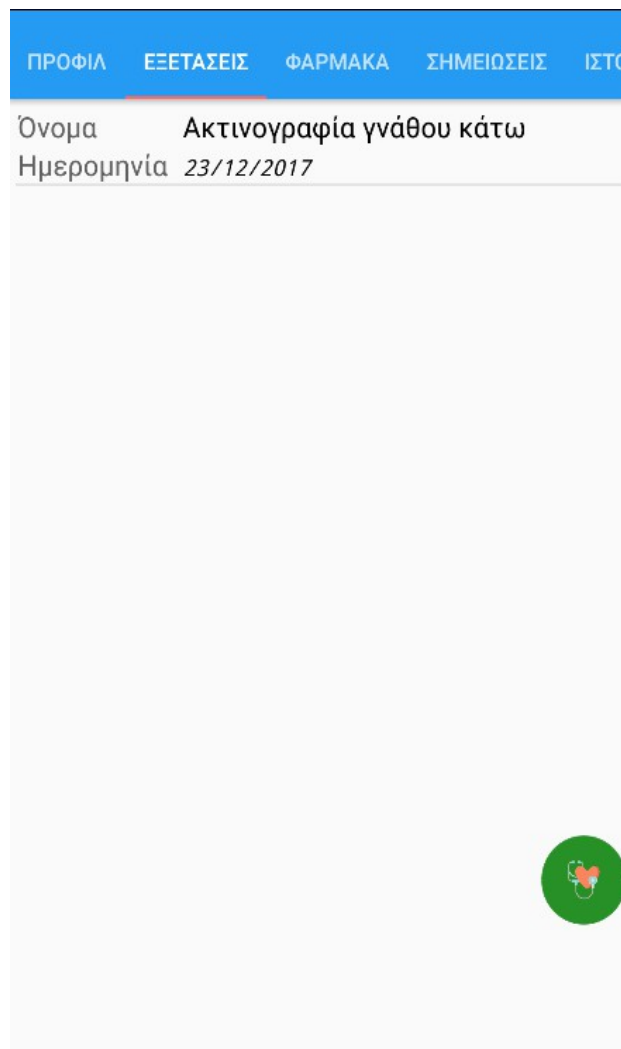
```

birthdate_input.setText(patient.getBirthdate());
entrydate_input.setText(patient.getEntrydate());
ethnicity_input.setText(patient.getEthnicity());
town_input.setText(patient.getTown());
address_input.setText(patient.getAddress());
tk_input.setText(patient.getTk());
telephone1_input.setText(patient.getTelephone1());
telephone2_input.setText(patient.getTelephone2());

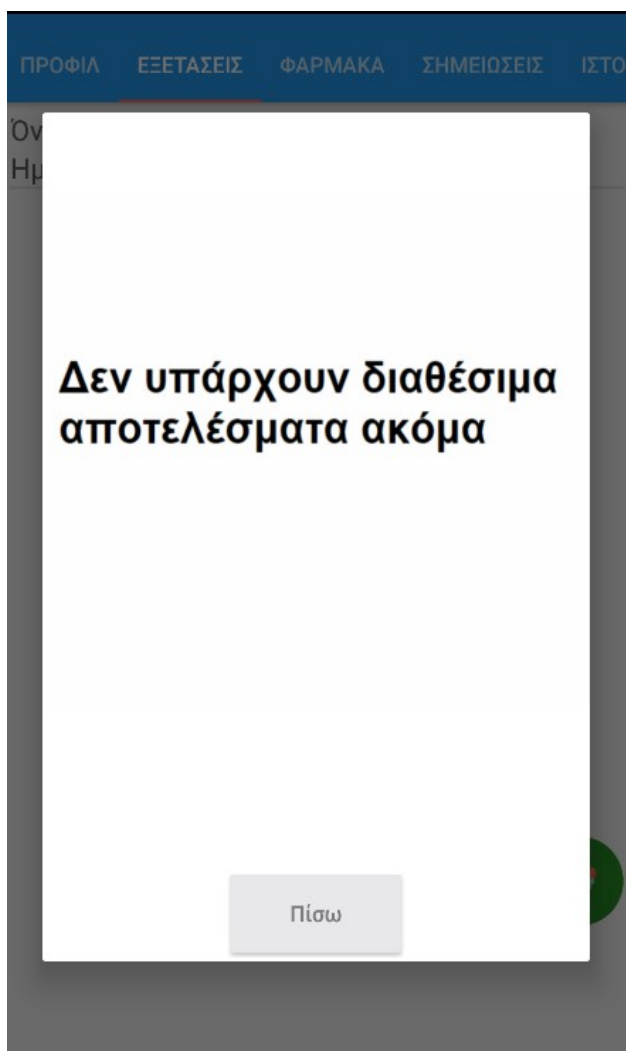
```

#### 4.3.6. Εξετάσεις

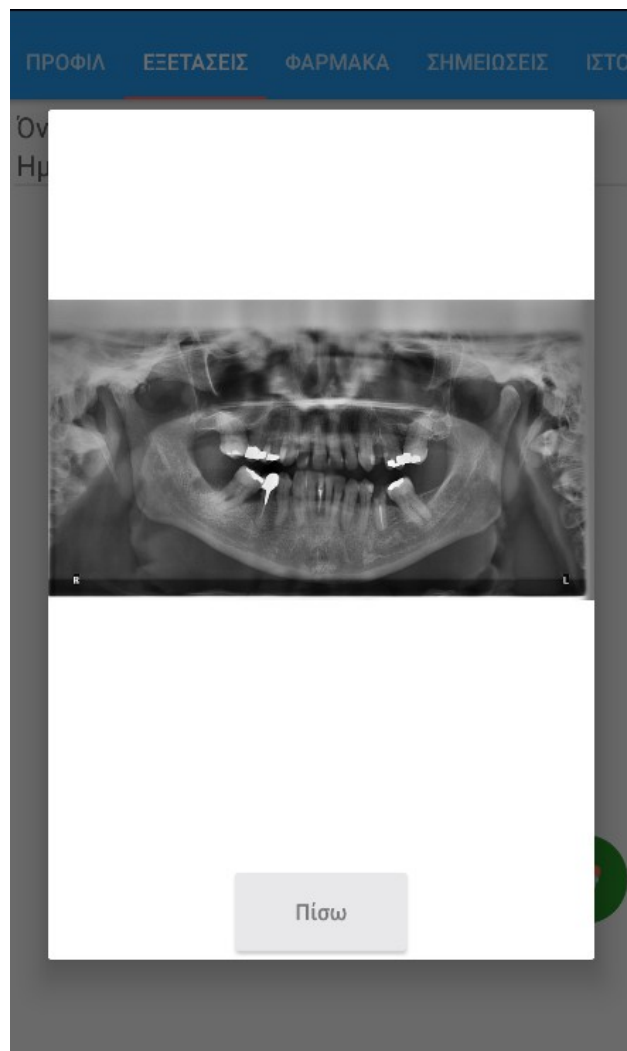
Στην καρτέλα των εξετάσεων, ο χρήστης έχει την δυνατότητα προβολής όλων των διαθέσιμων εξετάσεων και των αποτελεσμάτων τους αν υπάρχουν. Επίσης μπορεί να δημιουργήσει νέα εξέταση εισάγοντας το όνομα της εξέτασης και την ημερομηνία που θα γίνει. Μόνο ο γιατρός μπορεί να εισάγει νέα εξέταση, αν προσπαθήσει κάποιος χρήστης που είναι νοσοκομειακό προσωπικό, τότε του εμφανίζεται μήνυμα αποτυχίας.



Εικόνα 4.23: Προβολή εξετάσεων



Εικόνα 4.24: Μη διαθέσιμα αποτελέσματα



Εικόνα 4.25: Διαθέσιμα αποτελέσματα

The screenshot shows a mobile application interface with a dark blue header containing navigation tabs: ΠΡΟΦΙΛ, ΕΞΕΤΑΣΕΙΣ, ΦΑΡΜΑΚΑ, ΣΗΜΕΙΩΣΕΙΣ, and ΙΣΤΟΡΙΑ. Below the header, the user's name is 'Ακτινογραφία γνάθου κάτω' and the date is '23/12/2017'. A white dialog box titled 'Εισαγωγή Εξέτασης' is displayed in the foreground. It contains a red underline for the 'Όνομα Εξέτασης' field. Below this, the text 'Επιλέξτε Ημερομηνία Εξέτασης' is shown. A date picker is visible with three rows of options: 'Dec 01 2017', 'Jan 02 2018', and 'Feb 03 2019'. At the bottom of the dialog are two buttons: 'Πίσω' (Back) in a light gray box and 'Εισαγωγή' (Add) in a green box.

Εικόνα 4.26: Εισαγωγή νέας εξέτασης

Η υλοποίηση των λειτουργιών στην καρτέλα εξετάσεις, πραγματοποιείται στην κλάση `ExaminationsFragment` [1]. Η κλάση που αντιστοιχεί στις πληροφορίες κάθε εξέτασης είναι η `Examination` {m} και το layout που χρησιμοποιεί είναι `fragment_examinations`. Οι μεταβλητές της κλάσης `Examination` είναι:

```
private String id;  
private String name;  
private String date;  
private String doctor;
```

Για την προβολή όλων των εξετάσεων σε λίστα γίνεται κλήση στην βάση με την μεταβλητή `referenceDB` στο πεδίο `examinations`, και δημιουργείται ένα αντικείμενο της κλάσης `Examination` με τα απαραίτητα στοιχεία της κάθε εξέτασης τα οποία προστίθεται στην λίστα. Για το γέμισμα της λίστας με πολλές πληροφορίες, γίνεται η χρήση μιας custom κλάσης που λειτουργεί ως τροφοδότης της λίστας. Αυτό συμβαίνει και σε όλα τα επόμενα fragments. Επίσης κάθε



adapter χρησιμοποιεί την δικιά του λίστα που βρίσκεται σε ξεχωριστό αρχείο XML, και στην συγκεκριμένη περίπτωση των εξετάσεων η λίστα βρίσκεται στο αρχείο `list_item_examinations`.

## Προβολή φαρμάκων από την βάση

```
//Παίρνουμε όλες τις εξετάσεις από την βάση
referenceDB.child("examinations").addListenerForSingleValueEvent(new ValueEventListener() {
    @Override

    (παράλειψη κάποιων γραμμών)

    //Κλήση της βάσης στο πεδίο users
referenceDB.child("users").addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot1) {
        for(DataSnapshot doctorTemp : dataSnapshot1.getChildren()){
            //Βρίσκουμε τον χρήστη που ανέθεσε την εξέταση ώστε να πάρουμε το όνομα του
            if(docID.equals(String.valueOf(doctorTemp.getKey()))){
                e.setId(String.valueOf(examination.getKey()));
                e.setName(String.valueOf( examination.child("name").getValue()));
                e.setDate(String.valueOf( examination.child("date").getValue()));
                e.setDoctor(String.valueOf(doctorTemp.child("name").getValue())+" " +
String.valueOf(doctorTemp.child("surname").getValue()));

                for(DataSnapshot patientTempID : examination.child("patients").getChildren()){
                    //Αν είναι ο χρήστης που επιλέχθηκε τότε μπαίνει η εξέταση στην λίστα
                    if(patient.getId().equals(String.valueOf(patientTempID.getKey()))){
                        allExaminations.add(e);
                    }
                }
            }
        }
    }
    //Εμφανίζουμε όλες τις εξετάσεις του ασθενούς με τις απαραίτητες πληροφορίες
    //Δημιουργία του adapter για την μετατροπή της λίστας σε views
    ExaminationsAdapter adapter = new ExaminationsAdapter(getActivity(),allExaminations);
    //Σύνδεση του adapter με το αντικείμενο της λίστας στο layout
    examinationsList.setAdapter(adapter);
}

});
```

Adapter για τις εξετάσεις {n}. Τονίζεται ότι δεν θα τοποθετηθεί κώδικας των επόμενων adapters κατά την παρουσίαση της εφαρμογής, καθώς μοιάζουν στην δομή τους και περιέχονται αναλυτικά στο παράρτημα.

```
public class ExaminationsAdapter extends ArrayAdapter<Examination> {
    //Constructor για τον ορισμό των στοιχείων
    public ExaminationsAdapter(Context context, ArrayList<Examination> examinations) {
        super(context, 0, examinations);
    }
    @NonNull
    @Override
    public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent)
    {
        Examination examination = getItem(position);
        if (convertView == null) {
            //Σύνδεση του adapter με το layout που περιέχει την λίστα

```

```

        convertView = LayoutInflater.from(getContext())
            .inflate(R.layout.list_item_examinations, parent, false);
    }

    //Σύνδεση των TextView με τις μεταβλητές
    TextView examinationTextView = (TextView)
convertView.findViewById(R.id.examinationTextView);
    TextView dateExaminationTextView = (TextView)
convertView.findViewById(R.id.dateExaminationTextView);

    //Φόρτωση των TextView με τις πληροφορίες που στάλθηκαν με την χρήση της κλάσης
    Examination

    examinationTextView.setText(examination.getName());
    dateExaminationTextView.setText(examination.getDate());
    return convertView;
}
}

```

Για την προβολή των αποτελεσμάτων της εξέτασης αν υπάρχουν, γίνεται αναγνώριση ποια εξέταση επέλεξε ο χρήστης από την λίστα και με την χρήση παραθύρου διαλόγου `AlertDialog`, φορτώνεται η εικόνα. Η φόρτωση της εικόνας γίνεται από τον χώρο αποθήκευσης των εικόνων στην πλατφόρμα Firebase.

Κύριο κομμάτι κώδικα για την χρήση του storage του Firebase. Τα ανεβασμένα αρχεία είναι χωρισμένα σε φακέλους με ονομασία το κλειδί του κάθε ασθενούς, και μέσα στον φάκελο βρίσκονται τα αποτελέσματα των εξετάσεων σε μορφή εικόνας. Κάθε εικόνα έχει σαν ονομασία το κλειδί της εξέτασης.

```

StorageReference filePath = storage.child(patient.getId() + "/" + examinationSelected.getId()
+ ".jpg" );

filePath.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
    @Override
    public void onSuccess(Uri uri) {
        Glide.with(getContext()).load(uri.toString()).into(resultsImage);
    }
}).addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception exception) {
        resultsImage.setImageResource(R.drawable.noresults);
    }
});

```

Για την δημιουργία της νέας εξέτασης, ο χρήστης εισάγει το όνομα της και την ημερομηνία που θα γίνει. Στην βάση αποθηκεύονται τα δεδομένα αυτά σαν αντικείμενο της κλάσης `Examination`, μαζί με το κλειδί του ασθενούς και το κλειδί του χρήστη που την ανέθεσε.

Εισαγωγή της νέας εξέτασης στην βάση δεδομένων.

```

//Το αντικείμενο περιέχει όλα τα δεδομένα της νέας εξέτασης
referenceDB.child("examinations")
    .child(newExamination.getId())
    .setValue(newExamination);
//Εισαγωγή του κλειδιού του ασθενούς
referenceDB.child("examinations")
    .child(newExamination.getId())

```

```

        .child("patients")
        .child(patient.getId())
        .setValue(true);

//Εισαγωγή του κλειδιού του χρήστη που δημιούργησε την νέα εξέταση
referenceDB.child("examinations")
        .child(newExamination.getId())
        .child("users")
        .child(userID)
        .setValue(true);

```

#### 4.3.7. Φάρμακα

Στην καρτέλα των φαρμάκων, ο χρήστης έχει την δυνατότητα προβολής όλων των διαθέσιμων φαρμάκων και την διαγραφή τους. Επίσης μπορεί να δημιουργήσει νέο φάρμακο εισάγοντας το όνομα του, την δοσολογία και την συχνότητα. Η ημερομηνία που δόθηκε το φάρμακο αποθηκεύεται εισάγεται αυτόματα, όπως και επίσης το όνομα του χρήστη που την δημιούργησε. Μόνο ο γιατρός μπορεί να εισάγει νέα φάρμακα, αν προσπαθήσει κάποιος χρήστης που είναι νοσοκομειακό προσωπικό, τότε του εμφανίζεται μήνυμα αποτυχίας.

Η υλοποίηση των λειτουργιών των φαρμάκων γίνεται στην κλάση `MedicinesFragment {o}` και χρησιμοποιεί το layout `fragment_medicines`.

Η προβολή των φαρμάκων γίνεται με παρόμοιο τρόπο όπως οι εξετάσεις, άρα θα παραλειφθεί η τοποθέτηση κώδικα στην παρουσίαση. Γίνεται και πάλι χρήση adapter για την εισαγωγή πολλών τιμών ανά θέση της λίστας, ο adapter είναι `MedicinesAdapter {r}`.

Η κλάση που χρησιμοποιείται για την δημιουργία αντικειμένων φαρμάκου είναι η `Medicine {q}`. Και περιέχονται οι παρακάτω τιμές:

```

private String id;
private String name;
private String doctor;
private String date;
private String dose;
private String frequency;

```

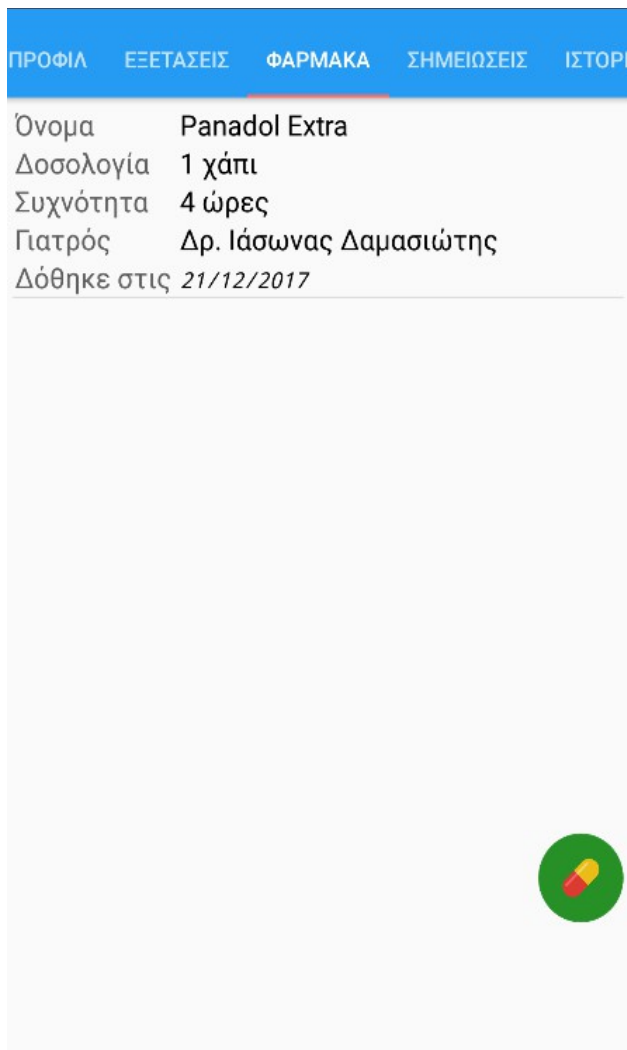
- ID – είναι το μοναδικό κλειδί του φαρμάκου
- name – το όνομα του φαρμάκου
- doctor – ο γιατρός που το δημιούργησε
- date – ημερομηνία που δόθηκε
- dose – η δοσολογία
- frequency – συχνότητα

Η κλήση στην βάση με την μεταβλητή `referenceDB` για τα φάρμακα γίνεται στο πεδίο `medicines` με τον παρακάτω τρόπο:

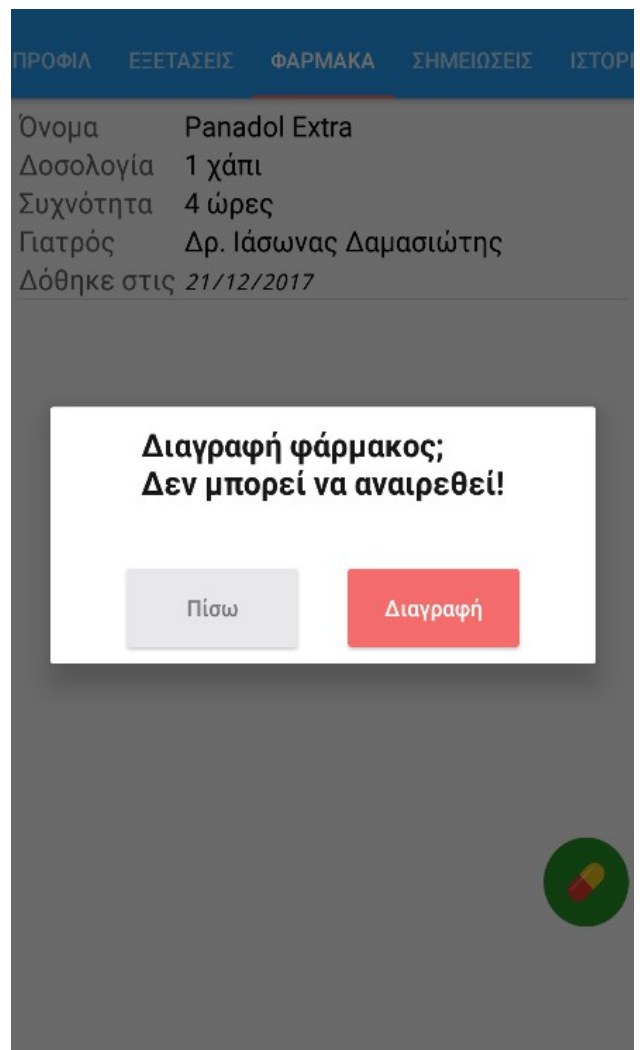
```

referenceDB.child("medicines").addListenerForSingleValueEvent(new ValueEventListener() {

```



Εικόνα 4.27: Προβολή φαρμάκων



Εικόνα 4.28: Διαγραφή επιλεγμένου φαρμάκου

Όπως αναφέρθηκε, ο χρήστης έχει την δυνατότητα διαγραφής φαρμάκων. Μόλις επιλέξει το επιθυμητό φάρμακο για διαγραφή τότε του εμφανίζεται μήνυμα επιβεβαίωσης διαγραφής του φαρμάκου. Το Firebase περιέχει την κατάλληλη συνάρτηση για την διαγραφή μιας τιμής στην βάση δεδομένων ανάλογα με το κλειδί της, η συνάρτηση αυτή είναι η `removeValue`. Η διαγραφή γίνεται με τον παρακάτω τρόπο:

```
referenceDB.child("medicines").child(medicineSelected.getId()).removeValue();
```

ΠΡΟΦΙΛ ΕΞΕΤΑΣΕΙΣ ΦΑΡΜΑΚΑ ΣΗΜΕΙΩΣΕΙΣ ΙΣΤΟΡΙΑ

Όνομα Panadol Extra  
 Δοσολογία 1 χάπι  
 Συχνότητα 4 ώρες  
 Γιατρός Δρ. Ιάσωνας Δαμασιώτης  
 Δόση


### Εισαγωγή Φάρμακος

Όνομα Φάρμακος

Δοσολογία

Συχνότητα

Πίσω Εισαγωγή



Εικόνα 4.29: Δημιουργία νέου φαρμάκου

### Εισαγωγή νέου φάρμακος στην βάση δεδομένων.

```
//Το αντικείμενο περιέχει όλα τα δεδομένα του νέου φάρμακος
referenceDB.child("medicines")
    .child(newMedicine.getId())
    .setValue(newMedicine);
//Εισαγωγή του κλειδιού του ασθενούς
referenceDB.child("medicines")
    .child(newMedicine.getId())
    .child("patients")
    .child(patient.getId())
    .setValue(true);
//Εισαγωγή του κλειδιού του χρήστη που δημιούργησε το νέο φάρμακο
referenceDB.child("medicines")
    .child(newMedicine.getId())
    .child("users")
    .child(userID)
    .setValue(true);
```

#### 4.3.8. Σημειώσεις

Στην καρτέλα των σημειώσεων, ο χρήστης έχει την δυνατότητα προβολής όλων των διαθέσιμων σημειώσεων που έχουν εισάγει όλοι οι χρήστες. Επίσης μπορεί να εισάγει νέα σημείωση πληκτρολογώντας το κείμενο της σημείωσης. Η ημερομηνία που δημιουργήθηκε η σημείωση αποθηκεύεται εισάγεται αυτόματα, όπως και επίσης το όνομα του χρήστη που την δημιούργησε. Μπορούν όλοι οι χρήστες να δημιουργήσουν νέες σημειώσεις.

Η υλοποίηση των λειτουργιών των σημειώσεων γίνεται στην κλάση `NotesFragment {r}` και χρησιμοποιεί το layout `fragment_notes`.

Η προβολή των φαρμάκων γίνεται με παρόμοιο τρόπο όπως τα φάρμακα και οι εξετάσεις, άρα θα παραλειφθεί η τοποθέτηση κώδικα στην παρουσίαση. Γίνεται και πάλι χρήση adapter για την εισαγωγή πολλών τιμών ανά θέση της λίστας, ο adapter είναι `NotesAdapter {s}`.

Η κλάση που χρησιμοποιείται για την δημιουργία αντικειμένων φαρμάκου είναι η `Note {t}`. Και περιέχονται οι παρακάτω τιμές:

```
private String id;  
private String date;  
private String info;  
private String user;
```

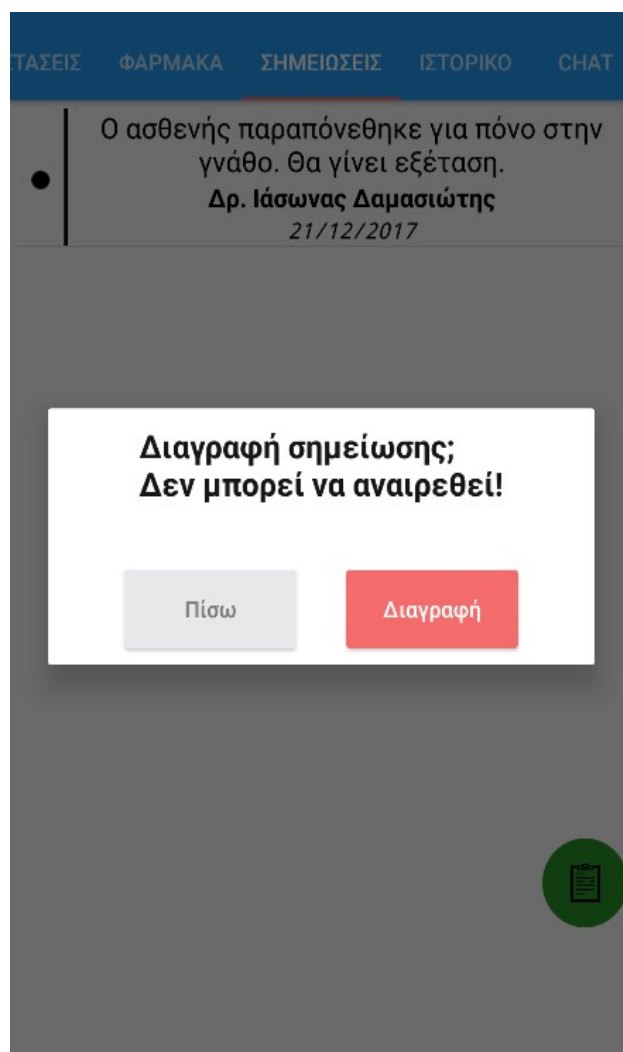
- ID – το μοναδικό κλειδί της σημείωσης
- date – ημερομηνία δημιουργίας της
- info – τα περιεχόμενα της σημείωσης
- user – ο χρήστης που την δημιούργησε

Η κλήση στην βάση με την μεταβλητή `referenceDB` για τα φάρμακα γίνεται στο πεδίο `notes` με τον παρακάτω τρόπο:

```
referenceDB.child("notes").addListenerForSingleValueEvent(new ValueEventListener() {
```



Εικόνα 4.30: Προβολή σημειώσεων



Εικόνα 4.31: Διαγραφή επιλεγμένης σημείωσης

Και στην περίπτωση των σημειώσεων ο χρήστης έχει την δυνατότητα διαγραφής σημειώσεων. Μόλις επιλέξει την επιθυμητή σημείωση για διαγραφή τότε του εμφανίζεται μήνυμα επιβεβαίωσης διαγραφής της σημείωσης. Η διαγραφή γίνεται με τον παρακάτω τρόπο:

```
referenceDB.child("notes").child(noteSelected.getId()).removeValue();
```



Εικόνα 4.32: Δημιουργία νέας σημείωσης

Εισαγωγή νέας σημείωσης στην βάση δεδομένων.

```
//Το αντικείμενο περιέχει όλα τα δεδομένα της νέας σημείωσης
referenceDB.child("notes")
    .child(newNote.getId())
    .setValue(newNote);
//Εισαγωγή του κλειδιού του ασθενούς
referenceDB.child("notes")
    .child(newNote.getId())
    .child("patients")
    .child(patient.getId())
    .setValue(true);
//Εισαγωγή του κλειδιού του χρήστη που δημιούργησε τη νέα σημείωση
referenceDB.child("notes")
    .child(newNote.getId())
    .child("users")
    .child(userID)
    .setValue(true);
```



### 4.3.9. Ιστορικό

Στην καρτέλα του ιστορικού, ο χρήστης έχει την δυνατότητα προβολής όλων των αλλαγών που έγιναν στις προηγούμενες καρτέλες από όλους του χρήστες, προσθήκη ή διαγραφή. Δεν υπάρχει δυνατότητα διαγραφής ή προσθήκης ιστορικού από και προς την λίστα. Εμφανίζεται η ενέργεια που έγινε (Προσθήκη ή Διαγραφή), το όνομα του χρήστη που εκτέλεσε την συγκεκριμένη ενέργεια, και την ημερομηνία που έγινε. Επίσης όλοι οι χρήστες μπορούν να προβάλουν το ιστορικό και ακόμα μπορούν να δουν τις λεπτομέρειες του νέου ιστορικού πατώντας στο αντικείμενο της λίστας που επιθυμούν. Η προβολή του ιστορικού γίνεται στην κλάση `HistoryFragment {u}` και χρησιμοποιεί το layout `fragment_history`. Όμως, η καταγραφή του ιστορικού (οι προσθήκες και οι αλλαγές) συμβαίνει στις κλάσεις:

- `ExaminationsFragment`
- `MedicinesFragment`
- `NotesFragment`

Σε αυτές τις κλάσεις υλοποιούνται οι λειτουργίες του συστήματος άρα η ενημέρωση της βάσης στο πεδίο `history` γίνεται από τις κλάσεις αυτές με κομμάτια κώδικα όπως για παράδειγμα συμβαίνει στην στην κλάση `ExaminationsFragment` για την δημιουργία νέας εξέτασης:

```
//Το αντικείμενο περιέχει όλα τα δεδομένα του ιστορικού. Γίνεται προσθήκη του νέου
//ιστορικού βάση της δομής που έχει η βάση
referenceDB.child("history")
    .child(newHistory.getId())
    .setValue(newHistory);
referenceDB.child("history")
    .child(newHistory.getId())
    .child("type")
    .child("examinations")
    .setValue(finalNewStringExamId);
referenceDB.child("history")
    .child(newHistory.getId())
    .child("patients")
    .child(patient.getId())
    .setValue(true);
referenceDB.child("history")
    .child(newHistory.getId())
    .child("users")
    .child(userID)
    .setValue(true);
```

Η προβολή των φαρμάκων γίνεται με παρόμοιο τρόπο όπως οι σημειώσεις, τα φάρμακα και οι εξετάσεις, άρα θα παραλειφθεί η τοποθέτηση κώδικα στην παρουσίαση. Γίνεται και πάλι χρήση adapter για την εισαγωγή πολλών τιμών ανά θέση της λίστας, ο adapter είναι `HistoryAdapter {v}`.

Η κλάση που χρησιμοποιείται για την δημιουργία αντικειμένων φαρμάκου είναι η `History {w}` Και περιέχονται οι παρακάτω τιμές:

```
private String id;
private String date;
private String type;
private String typeId;
private String user;
private String condition;
private String details;
```

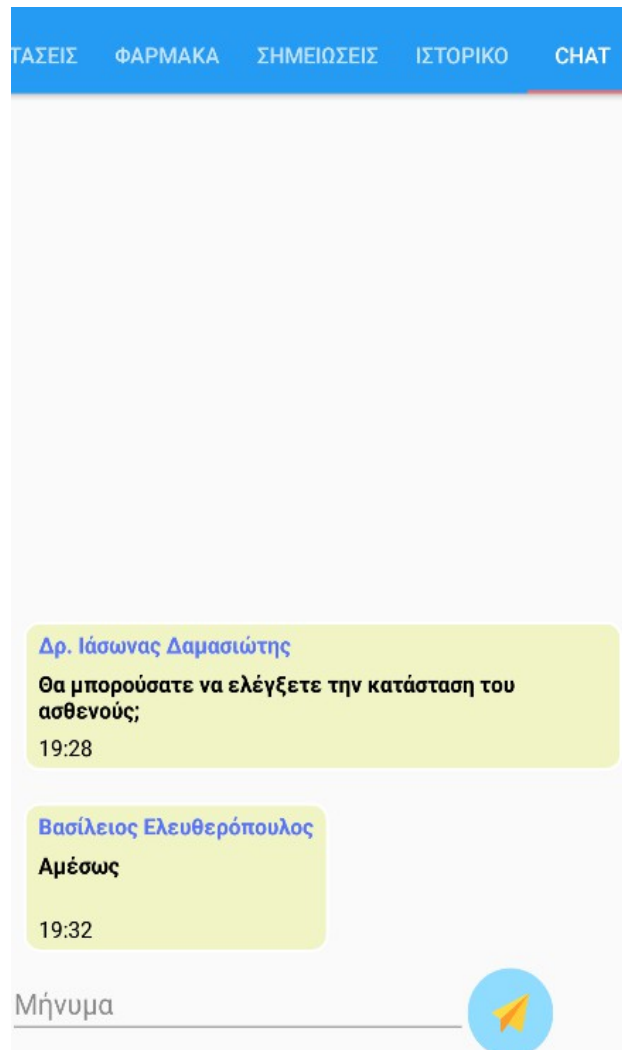
- ID – το μοναδικό κλειδί του ιστορικού
- date – ημερομηνία δημιουργίας
- type – τύπος αλλαγής (π.χ. Εξετάσεις)
- typeId – κλειδί αλλαγής (π.χ. ex000002)
- user – ο χρήστης που έκανε την αλλαγή
- condition – προσθήκη ή αφαίρεση
- details – οι λεπτομέρειες (π.χ. κείμενο σημείωσης, πληροφορίες φαρμάκου και εξετάσεων)

ΕΞΕΤΑΣΕΙΣ	ΦΑΡΜΑΚΑ	ΣΗΜΕΙΩΣΕΙΣ	ΙΣΤΟΡΙΚΟ	CHAT
Προσθήκη στο πεδίο <b>Εξετάσεις</b>				
<b>Δρ. Ιάσωνας Δαμασιώτης</b>				
21/12/2017				
Προσθήκη στο πεδίο <b>Φάρμακα</b>				
<b>Δρ. Ιάσωνας Δαμασιώτης</b>				
21/12/2017				
Προσθήκη στο πεδίο <b>Σημειώσεις</b>				
<b>Δρ. Ιάσωνας Δαμασιώτης</b>				
21/12/2017				

Εικόνα 4.33: Προβολή ιστορικού

#### 4.3.10. Ανταλλαγή μηνυμάτων

Η εφαρμογή δίνει την δυνατότητα στους χρήστες να ανταλλάξουν άμεσα μηνύματα μεταξύ τους. Αυτή η λειτουργία υλοποιείται στην καρτέλα Chat της κλάσης `ChatFragment {x}`. Στην οθόνη εμφανίζεται κάτω ένα πεδίο για πληκτρολόγηση μηνύματος και δεξιά του το κουμπί αποστολής και πάνω από το πεδίο αυτό εμφανίζονται τα μηνύματα που υπάρχουν ή που στέλνονται εκείνη την στιγμή.



Εικόνα 4.34: Ανταλλαγή άμεσων μηνυμάτων

Η `ChatFragment` κλάση χρησιμοποιεί το layout `fragment_chat`, αλλά κάθε μήνυμα έχει το δικό του layout με όνομα `single_message`. Εκεί πέρα εμφανίζονται μορφοποιημένες οι πληροφορίες του ονόματος του αποστολέα, το κείμενο και η ώρα που στάλθηκε. Επίσης, χρησιμοποιείται η κλάση `Message {y}` για την δημιουργία αντικειμένων για κάθε μήνυμα, που περιέχει τις παρακάτω μεταβλητές:

```
private String content;  
private String name;  
private String time;
```

- Content – κείμενο του μηνύματος
- name – όνομα αποστολέα
- time – ώρα αποστολής μηνύματος

Η παρουσίαση των μηνυμάτων στην βάση γίνεται με κλήση στο πεδίο `messages` και με την χρήση `FirestoreRecyclerAdapter` της κλάσης του `FirestoreUI` με το οποίο γίνεται αυτόματη ανανέωση της οθόνης όταν έρχονται νέα μηνύματα. Δίνονται ως ορίσματα η κλάση `Message` και το layout `single_message` για κάθε μήνυμα όπως αναφέρθηκε, και στην συνέχεια με ασύγχρονη κλήση γίνεται γέμισμα των δεδομένων με τα περιεχόμενα του μηνύματος κάνοντας χρήση των setters των μεταβλητών της κλάσης.

```
final FirestoreRecyclerAdapter<Message,MessageViewHolder> FBRA = new
FirestoreRecyclerAdapter<Message, MessageViewHolder>(
    Message.class,
    R.layout.single_message,
    MessageViewHolder.class,
    referenceDB
) {
    @Override
    protected void populateViewHolder(MessageViewHolder viewHolder, Message model, int
position) {
        viewHolder.setContent(model.getContent());
        viewHolder.setName(model.getName());
        viewHolder.setTime(model.getTime());
    }
};
messageList = (RecyclerView) view.findViewById(R.id.messageRec);
messageList.setHasFixedSize(true);
LinearLayoutManager linearLayoutManager = new LinearLayoutManager(getActivity());
linearLayoutManager.setStackFromEnd(true);
messageList.setLayoutManager(linearLayoutManager);
messageList.setAdapter(FBRA);
```

Κατά την αποστολή νέου μηνύματος γίνεται αποστολή των πληροφοριών του κάθε μηνύματος, στο πεδίο `messages`. Χρησιμοποιείται η μέθοδος `push` για αυτόματη δημιουργία κλειδιών του κάθε μηνύματος. Στην συνέχεια αποθηκεύεται το κείμενο του μηνύματος, ελέγχεται αν είναι γιατρός ο χρήστης ώστε να γίνει προσθήκη του “Δρ.” μπροστά από το όνομα του, αποθήκευση ονόματος χρήστη με την πλήρη μορφή, αποθήκευση ώρας που στάλθηκε το μήνυμα με την χρήση `Calendar` και `SimpleDateFormat`.

```
final DatabaseReference newPost = referenceDB.push();
usersDB.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        //Αποθήκευση κειμένου μηνύματος στην βάση
        newPost.child("content").setValue(messageValue);
        //Έλεγχος αν είναι γιατρός ο χρήστης
        getType = dataSnapshot.child("type").getValue().toString();
        if(getType.equals("doctor")){
            nameToWrite = "Δρ. " + dataSnapshot.child("name").getValue() + " " +
dataSnapshot.child("surname").getValue();
        }
        else{
            nameToWrite = dataSnapshot.child("name").getValue() + " " +
dataSnapshot.child("surname").getValue();
        }
        //Αποθήκευση ονόματος χρήστη στην βάση
        newPost.child("name").setValue(nameToWrite).addOnCompleteListener(new
OnCompleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> task) {
```

```

    }
    });

    //Δημιουργία ώρας που στάλθηκε το μήνυμα
    Calendar calendar = Calendar.getInstance();
    SimpleDateFormat simpleDateFormat = new SimpleDateFormat("HH:mm");
    String time = simpleDateFormat.format(calendar.getTime());
    //Αποθήκευση ώρας στην βάση
    newPost.child("time").setValue(time).addOnCompleteListener(new
    OnCompleteListener<Void>() {
        @Override
        public void onComplete(@NonNull Task<Void> task) {
        }
    });
    editMessage.setText("");
}
});

```

## Κεφάλαιο 5: Παρόμοιες εφαρμογές

### 5.1. MediFile EMR

Ο ηλεκτρονικός φάκελος ασθενών “MideFile EMR” , δημιουργήθηκε από την εταιρεία CCS για την κάλυψη των αναγκών ενός νοσοκομείου. Είναι υλοποιημένη για web περιβάλλον και έχει γίνει export σε κινητές συσκευές. Κύριο γνώρισμα αυτής της πλατφόρμας, είναι η δυνατότητα σύνδεσης σε πληροφοριακά συστήματα όπως:

- HIS -hospital information system
- ΔΠΣΝ – διοικητικό-οικονομικό πληροφοριακό σύστημα
- LIS – laboratory information system
- RIS/PAC – radiology information system
- Παθολογοανατομικά συστήματα
- Συστήματα αιμοδοσίας και τράπεζα αίματος

Η πλατφόρμα αυτή έχει δυνατότητες διαχείριση της άφιξης των ασθενών, ιστορικό τους πάσης φύσεως, παρακολούθηση ασθενών ανά μονάδα θεραπείας, εξετάσεις, διαγνώσεις, αγωγές, παρακολούθηση αποτελεσμάτων, ορισμό εξετάσεων και άλλα.

Η βάση δεδομένων που χρησιμοποιεί είναι Oracle, MySQL.

Έχει την δυνατότητα export των περιεχομένων σε έντυπα. [70]

### 5.2. ΗΦΑ

Η εφαρμογή “Ηλεκτρονικός Ιατρικός Φάκελος Ασθενών - ΗΦΑ”, ξεκίνησε να υλοποιείται το 2006 από τον ιατρό Ευτύχιο Κωνσταντινιδάκη. Είναι μια εφαρμογή δημιουργημένη με το Microsoft Access 2003. Δίνει την δυνατότητα στον χρήστη εισαγωγής των βασικών του στοιχείων του ασθενούς όπως το όνομα, φύλο κτλ.

Η εφαρμογή παρέχει τις εξής δυνατότητες στους χρήστες:

- Εισαγωγή φαρμακευτικής αγωγής
- Επισκέψεις στο ιατρείο
- Ιατρικό ιστορικό
- Ιστορικό συγγενών
- Εξετάσεις αίματος/ούρων

Ακόμα, γίνεται εξαγωγή στατιστικών στοιχείων διαφόρων κατηγοριών, π.χ. Ποια άτομα έχουν συγκεκριμένη πάθηση ή συνδυασμός φαρμάκου και φύλου, συνδυασμός φαρμάκων, συνδυασμός πάθησης και φαρμάκου κτλ. Το πρόγραμμα εξάγει τα αποτελέσματα σε αρχεία που μπορούν να επεξεργαστούν από το Excel ή σε έντυπη μορφή για εύκολη αρχειοθέτηση.

Τέλος, παρέχει στατιστικά στοιχεία για το ιατρείο όπως στατιστικά ημέρας και μήνα. Η αναφορά που εξάγεται, περιέχει την κίνηση του ιατρείου από άποψης ασθενών βάσει των στοιχείων που έχουν καταχωρηθεί στην ιστοσελίδα με την επισκεψιμότητα στο ιατρείο. [71]

## Κεφάλαιο 6: Συμπεράσματα-Προτάσεις

### 6.1. Συμπεράσματα

Κατά την δημιουργία της Android εφαρμογής “HealthApp” και την συγγραφή της πτυχιακής αποκτήθηκαν πολλές γνώσεις και εμπειρίες. Αρχικά έγινε το πλάνο της δομής της εφαρμογής και έρευνα αναγκών που έχουν τα νοσοκομεία και το ιατρικό προσωπικό. Ερευνήθηκε η πλατφόρμα Firebase και οι λειτουργίες που παρέχει σε μια εφαρμογή. Αποκτήθηκαν περισσότερες γνώσεις πάνω στην γλώσσα προγραμματισμού Java για γράψιμο κώδικα με τρόπο που αρμόζει σε αντικειμενοστρεφείς γλώσσες προγραμματισμού. Επιπροσθέτως, στην διάρκεια της συγγραφής του θεωρητικού μέρους της πτυχιακής, κατανοήθηκε η λειτουργία του λειτουργικού συστήματος του Android και η δομή του. Ακόμη, μελετήθηκαν οι βάσεις δεδομένων NoSQL καθώς η βάση δεδομένων της πλατφόρμας Firebase ανήκει στην κατηγορία αυτήν. Κατά την έρευνα της πλατφόρμας Firebase, παρουσιάστηκαν λεπτομερώς οι κύριες λειτουργίες της που χρησιμοποιήθηκαν από την εφαρμογή.

### 6.2. Προτάσεις

Η Android εφαρμογή “HealthApp”, αν και πλήρως λειτουργική, δέχεται βελτιώσεις ώστε να μπορεί να χρησιμοποιηθεί από τα νοσοκομεία για την κάλυψη περισσότερων αναγκών και βελτιστοποίησης των ήδη υπάρχων λύσεων.

Κάποιες βελτιώσεις και προσθήκες μπορούν να είναι:

- Παρουσίαση στατιστικών ανά ασθενή. Διαγράμματα που δείχνουν την πορεία του ασθενούς κατά την παρουσία του στο νοσοκομείο.
- Web πλατφόρμα για τον διαχειριστή. Δημιουργία σελίδας ώστε ο διαχειριστής να κάνει τις αλλαγές πιο εύκολα και κατανοητά και όχι μέσα από το περιβάλλον της πλατφόρμας του Firebase.
- Εξαγωγή ιστορικού σε έντυπη μορφή.
- Προφίλ για τον χρήστη που θα περιέχει τα στοιχεία του, πόσους ασθενείς έχει και κάποια στατιστικά.
- Προσθήκη κάποιων φωνητικών λειτουργιών, κυρίως για την αντικατάσταση της πληκτρολόγησης.
- Διαθεσιμότητα εφαρμογής και σε συσκευές iOS
- Δυνατότητα αναζήτησης των εξετάσεων, φαρμάκων, σημειώσεων και ιστορικού αντί για μόνο προβολή λίστας με τα στοιχεία.
- Προσθήκη κάποιου στρώματος προστασίας κατά την προσθήκη νέων εξετάσεων και φαρμάκων, όπως να τα επιβεβαιώσει ο διαχειριστής ή κάποιος προϊστάμενος.
- Χρέωση και τιμολόγηση
- Περαιτέρω διαχωρισμός ανά είδος, όπως φυσικοθεραπεία, παιδιατρικό, ογκολογικό κτλ.

## Κεφάλαιο 7: Βιβλιογραφία

- [1] Περιγραφή Android <https://developer.android.com/about/index.html>
- [2] Πρώτη έκδοση Android :<https://android-developers.googleblog.com/2008/09/announcing-android-10-sdk-release-1.html>
- [3] Ιστορική αναδρομή του Android [https://www.phonearena.com/news/Googles-Android-OS-Past-Present-and-Future\\_id21273](https://www.phonearena.com/news/Googles-Android-OS-Past-Present-and-Future_id21273)
- [4] Εξαγορά του Android Inc. από την Google  
[https://web.archive.org/web/20110205190729/http://www.businessweek.com/technology/content/aug2005/tc20050817\\_0949\\_tc024.htm](https://web.archive.org/web/20110205190729/http://www.businessweek.com/technology/content/aug2005/tc20050817_0949_tc024.htm)
- [5] Υποστήριξη Kotlin στο Android <https://www.theverge.com/2017/5/17/15654988/google-jet-brains-kotlin-programming-language-android-development-io-2017>
- [6] Εξαγορά firebase Inc από την Google <https://firebase.googleblog.com/2014/10/firebase-is-joining-google.html>
- [7] Πληροφορίες για το Firebase <https://howtofirebase.com/what-is-firebase-fcb8614ba442>
- [8] Εισαγωγικά στοιχεία για το Firebase <https://firebase.googleblog.com/2012/04/developers-meet-firebase.html>
- [9] Ανακοίνωση πρωτότυπης συσκευής για το Android  
[https://www.pcworld.com/article/254539/original\\_android\\_prototype\\_revealed\\_during\\_google\\_oracle\\_trial.html](https://www.pcworld.com/article/254539/original_android_prototype_revealed_during_google_oracle_trial.html)
- [10] Android open source <https://source.android.com/>
- [11] Android Studio εισαγωγικά <https://developer.android.com/studio/features.html>
- [12] Το Android ξεπέρασε τους 2 δις χρήστες <https://venturebeat.com/2017/05/17/android-passes-2-billion-monthly-active-devices/>
- [13] Επεξεργαστές και αρχιτεκτονική <https://developer.android.com/ndk/guides/arch.html>
- [14] Πληροφορίες για ABI <https://developer.android.com/ndk/guides/abis.html>
- [15] Remix OS <http://www.jide.com/remixos-for-pc>
- [16] Μέσα που υποστηρίζει το Android  
<https://developer.android.com/guide/topics/media/media-formats.html#audio-formats>
- [17] Multitouch στο Android <https://www.engadget.com/2010/02/02/nexus-one-gets-a-software-update-enables-multitouch/>
- [18] Apple multitouch 2007 <https://www.cnet.com/news/report-apple-nixed-androids-multitouch/>
- [19] Google Maps API και χρήση τοποθεσίας  
<https://developer.android.com/guide/topics/location/index.html>
- [20] Animation και Γραφικά στο Android  
<https://developer.android.com/guide/topics/graphics/overview.html>



- [21] Animation στο Android <https://developer.android.com/guide/topics/graphics/view-animation.html>
- [22] Android Security Ανακοίνωση <http://googlemobile.blogspot.gr/2012/02/android-and-security.html>
- [23] Έρευνα F-Secure για μολυσμένες εφαρμογές στο Play Store [https://www.phonearena.com/news/Android-malware-perspective-only-0.5-comes-from-the-Play-Store\\_id36696](https://www.phonearena.com/news/Android-malware-perspective-only-0.5-comes-from-the-Play-Store_id36696)
- [24] Malware στο Android το 2012 <http://www.zdnet.com/article/android-malware-numbers-explode-to-25000-in-june-2012/>
- [25] Sideloaded Εφαρμογές <https://phandroid.com/2013/07/20/android-101-sideloaded-apps/>
- [26] Έρευνα πανεπιστημίου Cambridge <https://www.cl.cam.ac.uk/~drt24/papers/spsm-scoring.pdf>
- [27] Android Security <https://source.android.com/security/>
- [28] Λίγα λόγια για το AOSP <https://www.androidcentral.com/aosp>
- [29] Νέες δυνατότητες του Android από την open source κοινότητα <https://www.wired.com/2011/04/cyanogenmod-android/>
- [30] Linux kernel άρθρο <https://www.androidcentral.com/android-z-what-kernel>
- [31] ION <https://github.com/koush/ion/blob/master/README.md>
- [32] Αλλαγές στον πυρήνα από την Google <https://www.forbes.com/sites/quora/2013/05/13/what-are-the-major-changes-that-android-made-to-the-linux-kernel/#1e8c7fa27bb6>
- [33] Android Linux Kernel <https://source.android.com/devices/architecture/kernel/>
- [34] Ασφάλεια του kernel <https://source.android.com/security/overview/kernel-security>
- [35] Υλοποίηση KASLR <https://android-developers.googleblog.com/2017/08/hardening-kernel-in-android-oreo.html>
- [36] Hardware Abstraction Layer – HAL <https://source.android.com/devices/#Hardware%20Abstraction%20Layer>
- [37] Android runtime – ART <https://source.android.com/devices/tech/dalvik/>
- [38] Αρχιτεκτονική Android <https://developer.android.com/guide/platform/index.html>
- [39] Ιστορικό εκδόσεων Android <https://www.android.com/history>
- [40] Εκδόσεις Android και χαρακτηριστικά <http://socialcompare.com/en/comparison/android-versions-comparison>
- [41] Χαρακτηριστικά εφαρμογής <https://developer.android.com/guide/components/fundamentals.html>
- [42] Κύκλος ζωής ενός activity <https://developer.android.com/guide/components/activities/activity-lifecycle.html>
- [43] Χαρακτηριστικά εφαρμογής αναλυτικότερα [https://www.tutorialspoint.com/android/android\\_application\\_components.htm](https://www.tutorialspoint.com/android/android_application_components.htm)

- [44] Fragments <https://developer.android.com/guide/components/fragments.html>
- [45] Fragments ανάλυση [https://www.tutorialspoint.com/android/android\\_fragments.htm](https://www.tutorialspoint.com/android/android_fragments.htm)
- [46] Διαδικασίες και Threads <https://developer.android.com/guide/components/processes-and-threads.html#Threads>
- [47] Threads <https://academy.realm.io/posts/android-threading-background-tasks/>
- [48] Περιγραφή αρχείου manifest <https://developer.android.com/guide/topics/manifest/manifest-intro.html>
- [49] Android Studio διαθέσιμο σε windows, os x και linux <http://www.redmondpie.com/download-android-studio-ide-for-windows-os-x-and-linux/>
- [50] Ανακοίνωση early access έκδοση του Android Studio <https://android-developers.googleblog.com/2013/05/android-studio-ide-built-for-android.html>
- [51] Android Studio 1.0 έκδοση <https://venturebeat.com/2014/12/08/google-releases-android-studio-1-0-the-first-stable-version-of-its-ide/>
- [52] Android Studio αναλυτικά <https://developer.android.com/studio/intro/index.html>
- [53] Android Emulator <https://developer.android.com/studio/run/emulator.html>
- [54] Genymotion Guide [https://docs.genymotion.com/pdf/PDF\\_User\\_Guide/Genymotion-2.11-User-Guide.pdf](https://docs.genymotion.com/pdf/PDF_User_Guide/Genymotion-2.11-User-Guide.pdf)
- [55] Ιστοσελίδα NoSQL <http://nosql-database.org/>
- [56] Περιγραφή NoSQL <http://searchdatamanagement.techtarget.com/definition/NoSQL-Not-Only-SQL>
- [57] Παραπάνω στοιχεία για τις NoSQL βάσεις <https://www.mongodb.com/nosql-explained>
- [58] Περισσότερες πληροφορίες για τις NoSQL βάσεις <http://basho.com/resources/nosql-databases/>
- [59] Δυνατότητες του Firebase <https://firebase.google.com/products/>
- [60] Firebase Authentication <https://firebase.google.com/docs/auth/>
- [61] Realtime Database Firebase <https://firebase.google.com/docs/database/>
- [62] BLOB Mysql για αποθήκευση αρχείων <https://dev.mysql.com/doc/refman/5.7/en/blob.html>
- [63] Αποθηκευτικός χώρος <https://firebase.google.com/docs/storage/>
- [64] Τι είναι MySQL <https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>
- [65] Σύγκριση Firebase και MySQL <https://db-engines.com/en/system/Firebase+Realtime+Database%3BMySQL>
- [66] Σύγκριση Firebase και SQLite <https://db-engines.com/en/system/Firebase+Realtime+Database%3BSQLite>
- [67] Σύγκριση Firebase και Couchbase <https://db-engines.com/en/system/Firebase+Realtime+Database%3BCouchbase>
- [68] Σύγκριση Firebase και Couchbase [https://rozdoum.com/blog\\_articles/items/firebase-vs-couchbase-feel-the-difference.html](https://rozdoum.com/blog_articles/items/firebase-vs-couchbase-feel-the-difference.html)

- [69] Σύγκριση Firebase με MongoDB <https://db-engines.com/en/system/Firebase+Realtime+Database%3BMongoDB>
- [70] Ηλεκτρονικός φάκελος ασθενών MediFile EMR <https://www.ccs.gr/products/medifile/?lang=el>
- [71] Ηλεκτρονικός ιατρικός φάκελος ασθενών ΗΦΑ  
[http://users.otenet.gr/~kyiteas7/ilektronikos\\_fakelos\\_asthenon.htm](http://users.otenet.gr/~kyiteas7/ilektronikos_fakelos_asthenon.htm)

## Κεφάλαιο 8: Παράρτημα

{a} Το project της εφαρμογής βρίσκεται στην τοποθεσία

<https://github.com/valkonzero/Thesis/tree/master/Android/HealthApp>

{b} Φάκελος resources

<https://github.com/valkonzero/Thesis/tree/master/Android/HealthApp/app/src/main/res>

{c} Αρχική δραστηριότητα MainActivity.java

<https://github.com/valkonzero/Thesis/blob/master/Android/HealthApp/app/src/main/java/com/emmanouilpapadimitrou/healthapp/Activities/MainActivity.java>

{d} Δραστηριότητα σύνδεσης χρήστη στο σύστημα

<https://github.com/valkonzero/Thesis/blob/master/Android/HealthApp/app/src/main/java/com/emmanouilpapadimitrou/healthapp/Activities/LoginActivity.java>

{e} Δραστηριότητα που περιέχει την λίστα των ασθενών και την λειτουργία της αποσύνδεσης

<https://github.com/valkonzero/Thesis/blob/master/Android/HealthApp/app/src/main/java/com/emmanouilpapadimitrou/healthapp/NavDrawer.java>

{f} Fragment με την λίστα των ασθενών του χρήστη

<https://github.com/valkonzero/Thesis/blob/master/Android/HealthApp/app/src/main/java/com/emmanouilpapadimitrou/healthapp/Fragments/PatientsFragment.java>

{g} Κλάση Ασθενής

<https://github.com/valkonzero/Thesis/blob/master/Android/HealthApp/app/src/main/java/com/emmanouilpapadimitrou/healthapp/POJOs/Patient.java>

{h} Κλάση Χρήστης

<https://github.com/valkonzero/Thesis/blob/master/Android/HealthApp/app/src/main/java/com/emmanouilpapadimitrou/healthapp/POJOs/Users.java>

{i} Δραστηριότητα που περιέχει τα βασικά fragments της εφαρμογής και συναρτήσεις που χρησιμοποιούνται από αυτά

<https://github.com/valkonzero/Thesis/blob/master/Android/HealthApp/app/src/main/java/com/emmanouilpapadimitrou/healthapp/Activities/PatientsActivity.java>

{j} Αντάπτορας Fragments

<https://github.com/valkonzero/Thesis/blob/master/Android/HealthApp/app/src/main/java/com/emmanouilpapadimitrou/healthapp/Adapters/SectionsPagerAdapter.java>

{k} Fragment Προφίλ ασθενούς

<https://github.com/valkonzero/Thesis/blob/master/Android/HealthApp/app/src/main/java/com/emmanouilpapadimitrou/healthapp/Fragments/ProfilePatientFragment.java>

{l} Fragment Εξετάσεων

<https://github.com/valkonzero/Thesis/blob/master/Android/HealthApp/app/src/main/java/com/emmanouilpapadimitrou/healthapp/Fragments/%CE%95xaminationsFragment.java>

{m} Κλάση Εξέτασης

<https://github.com/valkonzero/Thesis/blob/master/Android/HealthApp/app/src/main/java/com/emmanouilpapadimitrou/healthapp/POJOs/Examination.java>

{n} Αντάπτορας Εξετάσεων

<https://github.com/valkonzero/Thesis/blob/master/Android/HealthApp/app/src/main/java/com/emmanouilpapadimitrou/healthapp/Adapters/ExaminationsAdapter.java>

{o} Fragment Φαρμάκων

<https://github.com/valkonzero/Thesis/blob/master/Android/HealthApp/app/src/main/java/com/emmanouilpapadimitrou/healthapp/Fragments/MedicinesFragment.java>

{p} Αντάπτορας Φαρμάκων

<https://github.com/valkonzero/Thesis/blob/master/Android/HealthApp/app/src/main/java/com/emmanouilpapadimitrou/healthapp/Adapters/MedicinesAdapter.java>

{q} Κλάση Φαρμάκου

<https://github.com/valkonzero/Thesis/blob/master/Android/HealthApp/app/src/main/java/com/emmanouilpapadimitrou/healthapp/POJOs/Medicine.java>

{r} Fragment Σημειώσεων

<https://github.com/valkonzero/Thesis/blob/master/Android/HealthApp/app/src/main/java/com/emmanouilpapadimitrou/healthapp/Fragments/NotesFragment.java>

{s} Αντάπτορας Σημειώσεων

<https://github.com/valkonzero/Thesis/blob/master/Android/HealthApp/app/src/main/java/com/emmanouilpapadimitrou/healthapp/Adapters/HistoryAdapter.java>

{t} Κλάση Σημείωσης

<https://github.com/valkonzero/Thesis/blob/master/Android/HealthApp/app/src/main/java/com/emmanouilpapadimitrou/healthapp/POJOs/Note.java>

{u} Fragment Ιστορικού

<https://github.com/valkonzero/Thesis/blob/master/Android/HealthApp/app/src/main/java/com/emmanouilpapadimitrou/healthapp/Fragments/HistoryFragment.java>

{v} Αντάπτορας Ιστορικού

<https://github.com/valkonzero/Thesis/blob/master/Android/HealthApp/app/src/main/java/com/emmanouilpapadimitrou/healthapp/Adapters/HistoryAdapter.java>

{w} Κλάση Ιστορικού

<https://github.com/valkonzero/Thesis/blob/master/Android/HealthApp/app/src/main/java/com/emmanouilpapadimitrou/healthapp/POJOs/History.java>

{x} Fragment Ανταλλαγής μηνυμάτων

<https://github.com/valkonzero/Thesis/blob/master/Android/HealthApp/app/src/main/java/com/emmanouilpapadimitrou/healthapp/Fragments/ChatFragment.java>

{y} Κλάση μηνύματος

<https://github.com/valkonzero/Thesis/blob/master/Android/HealthApp/app/src/main/java/com/emmanouilpapadimitrou/healthapp/POJOs/Message.java>

{z} Apache License 2.0 <https://github.com/valkonzero/Thesis/blob/master/LICENSE>

{aa} Πιστώσεις-Credits <https://github.com/valkonzero/Thesis/blob/master/CREDITS>

{ab} Flaticon License <https://github.com/valkonzero/Thesis/blob/master/flaticonlicense.pdf>