



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ανάπτυξη Android Εφαρμογής «Πρώτες Βοήθειες (First Aid)» Development of Android Application «First Aid (First Aid)»
Όνοματεπώνυμο Φοιτητή	Κάσο Λευτέρης
Πατρώνυμο	Κωνσταντίνος
Αριθμός Μητρώου	ΜΠΠΛ/14027
Επιβλέπων	Αλέξης Ευθύμιος, Επίκουρος Καθηγητής

Ημερομηνία Παράδοσης: **Σεπτέμβριος 2016**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Όνομα Επώνυμο
Βαθμίδα

Όνομα Επώνυμο
Βαθμίδα

Όνομα Επώνυμο
Βαθμίδα

ΑΦΙΕΡΩΣΗ

**Αφιερωμένη σε όσους με
στήριξαν καθ' όλη τη διάρκεια
της προσπάθειας μου**



ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα διπλωματική εργασία δεν θα είχε ολοκληρωθεί χωρίς την ουσιαστική συμβολή του κ. Ευθύμιου Αλέπη με τον οποίο συνεργαστήκαμε στενά κατά την εκπόνηση της συγκεκριμένης διπλωματικής εργασίας. Θα ήθελα να τον ευχαριστήσω θερμά για την καθοδήγηση του και τις συμβουλές που απλόχερα μου έδωσε. Επιπλέον θα ήθελα να ευχαριστήσω την οικογένεια μου για την υποστήριξη τους όλα αυτά τα χρόνια, που ήταν δίπλα μου και με στηρίζουν να πέτυχω τους στόχους μου.

ΠΕΡΙΛΗΨΗ

Στόχος της παρούσας διπλωματικής εργασίας είναι να χρησιμοποιήσει τις δυνατότητες που προσφέρουν οι έξυπνες κινητές συσκευές και συγκεκριμένα το λειτουργικό σύστημα Android, για την ανάπτυξη μιας εφαρμογής για κινητές συσκευές.

Η εφαρμογή αυτή έχει σκοπό να βοηθήσει τον χρήστη σε μια έκτακτη ανάγκη, παρέχοντας του οδηγίες για τις Πρώτες Βοήθειες. Η εφαρμογή διαθέτει διάφορες λειτουργίες που εξυπηρετούν το στόχο αυτό, όπως διάφορες ηχητικές οδηγίες, καταχωρημένα επείγοντα τηλεφώνά, ιατρικό φάκελο ασθενούς κτλ. Πέραν από την εφαρμογή δημιουργήθηκε και μια βάση δεδομένων όπου αποθηκεύεται όλος ο ιατρικός φάκελος του χρήστη.

Μετά την ολοκλήρωση της εφαρμογής, η εφαρμογή δόθηκε σε φοιτητές για εκτενείς δοκιμές. Η δοκιμαστική χρήση από τους υπολοίπους σε πραγματικές συνθήκες βοήθησε στην διόρθωση και στην βελτίωση της εφαρμογής. Τα πρώτα μηνύματα ήταν ενθαρρυντικά και λήφθηκαν χρήσιμα σχόλια.

Αρχικά, αναφέρονται τα βασικά χαρακτηριστικά των κινητών τηλεφώνων, και συγκεκριμένα μια ιστορική ανάδρομη, τι περιλαμβάνει ένα κινητό τηλέφωνο, πια λειτουργικά συστήματα υπάρχουν, καθώς επίσης και των αριθμό των πωλήσεων των κινητών τηλεφώνων.

Στα επόμενα δυο κεφάλαια εμβαθύνουμε στο λειτουργικό σύστημα Android ,κάνοντας μια ιστορική ανασκόπηση, περιγράφοντας την αρχιτεκτονική του Android καθώς επίσης και όλες τις εκδόσεις του Android.

Στο τέταρτο μέρος, παρουσιάζονται οι απαιτήσεις που υπάρχουν για την ανάπτυξη εφαρμογών και τα εργαλεία που κάνουν εύκολη την ζωή του προγραμματιστή.

Έπειτα, παρουσιάζονται ο σχεδιασμός και η υλοποίηση της εφαρμογής που αναπτύχθηκε για το σκοπό της διπλωματικής εργασίας.

Κλείνοντας, παραθέτοντας τα συμπεράσματα του σπουδαστή και ιδέες για μελλοντική εξέλιξη της εφαρμογής καθώς αποτυπώνονται στο παράρτημα οι κώδικες που χρησιμοποιήθηκαν.

Τέλος ίσως ο βασικότερος σκοπός της εφαρμογής ήταν η εκμάθηση προγραμματισμού σε περιβάλλον Android. Αυτό επετεύχθη σε μεγάλο βαθμό καθώς χρησιμοποιήθηκαν πολλά διαφορετικά στοιχεία του Android, όπως *TextViews*, *ScrollViews*, *WebViews*, *EditTexts*, *Alert Dialogs*, *ImageSlider*, *Spinners*, *Tabs*, *Toast Messages* κ.α. Έχει δοθεί μεγάλη προσοχή και στο *User interface* της εφαρμογής ώστε να είναι ελκυστικό για τον χρήστη, σαν εφαρμογή στο Android Market.

ABSTRACT

The aim of this thesis is to utilize the capabilities that mobile smartphones provide -in particular the Android operating system - in order to develop an application for mobile devices.

This application's purpose is to help out the user in a case of emergency, by providing him the imperative information regarding medical first aid. The application disposes various operations that serve the purpose, such as separate and different audio instructions, registered phone numbers of emergency, patient medical history etc. Beyond the application itself, a database was created as well, wherein the medical record of the user is saved.

Once the application took its final form, it was handed out to university students for extensive testing. This testing usage of the application under real conditions assisted in the revision and the enhancement of it. The first reactions were encouraging and useful comments were received.

Initially, the key features of mobile phones are mentioned, and specifically, a retrospection of mobile phones, what constitutes a mobile phone, which operating systems occur, and as well as, the number of sales of mobile phones.

In the next two chapters, we are looking into the Android operating system by reviewing it. This was carried out through the description of the Android architecture and its versions.

In the fourth chapter, are presented the requirements for the application development and the tools that make the programmer's life easier.

Afterwards, the planning and implementation of the application are presented, which was developed for the purpose of this thesis.

In the end, the writer's conclusions and ideas are cited for the potential future development of the application as the codes used by the writer/student are written in the appendix.

And, finally, the main objective of this application was, probably, the learning of programming in an Android environment. This was fulfilled at a great scale as a lot of different elements of Android were used, such as *TextViews*, *ScrollViews*, *WebViews*, *EditTexts*, *Alert Dialogs*, *ImageSlider Spinners*, and *Tabs*, *Toast Messages*, among others. A special attention was paid on the *User Interface* of the application, too, in order to be visually attractive to the user as an application of the Android Market.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΚΕΦΑΛΑΙΟ 1^ο	σελ 10
1.1 ΕΙΣΑΓΩΓΗ	σελ 10
1.2 ΤΙ ΕΙΝΑΙ ΕΞΥΠΝΟ ΤΗΛΕΦΩΝΟ	σελ 11
1.3 ΥΛΙΚΟ ΤΩΝ ΕΞΥΠΝΩΝ ΚΙΝΗΤΩΝ	σελ 11
1.3.1 Γενικά.....	σελ 12
1.3.2 Επεξεργαστής.....	σελ 12
1.3.3 Μνήμη.....	σελ 12
1.3.4 Είσοδος.....	σελ 12
1.3.5 Οθόνη.....	σελ 12
1.3.6 Ασύρματη Δικτύωση.....	σελ 13
1.3.7 Φωτογραφική Μηχανή.....	σελ 13
1.3.8 Υλικό Εύρεσης Θέσης.....	σελ 13
1.3.9 Συνδεσιμότητα.....	σελ 13
1.4 ΕΚΔΟΣΕΙΣ ΛΕΙΤΟΥΡΓΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ	σελ 14
1.5 ΑΡΙΘΜΟΣ ANDROID APPLICATION	σελ 14
1.6 ΠΩΛΗΣΕΙΣ ΕΞΥΠΝΩΝ ΤΗΛΕΦΩΝΩΝ	σελ 15
ΚΕΦΑΛΑΙΟ 2^ο	σελ 17
2.1 ΕΙΣΑΓΩΓΗ-ΤΟ ΛΕΙΤΟΥΡΓΙΚΟ ANDROID	σελ 17
2.2 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ	σελ 17
2.3 ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ANDROID	σελ 19
2.4 Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΩΝ ANDROID	σελ 20
2.4.1 Το επίπεδο του πυρήνα του Linux.....	σελ 21
2.4.2 Το επίπεδο των βασικών βιβλιοθηκών.....	σελ 22
2.4.3 Το επίπεδο των Android Runtime.....	σελ 24
2.4.4 Το επίπεδο των Application Framework.....	σελ 27
2.4.5 Το επίπεδο των εφαρμογών.....	σελ 29
ΚΕΦΑΛΑΙΟ 3^ο	σελ 30
3.1 ΟΙ ΕΚΔΟΣΕΙΣ ΤΟΥ ANDROID	σελ 30
3.1.1 Έκδοση Android 1.0.....	σελ 31
3.1.2 Έκδοση Android 1.1.....	σελ 31
3.1.3 Έκδοση Android 1.5 (Cupcake).....	σελ 31
3.1.4 Έκδοση Android 1.6 (Donut).....	σελ 32
3.1.5 Έκδοση Android 2.0/2.1 (Éclair).....	σελ 33
3.1.6 Έκδοση Android 2.2 (Froyo).....	σελ 33
3.1.7 Έκδοση Android 2.3/2.3.2 (Gingerbread).....	σελ 34
3.1.8 Έκδοση Android 3.0/3.1/3.2 (Honeycomb).....	σελ 35
3.1.9 Έκδοση Android 4.0/4.0.2 (Ice Cream Sandwich).....	σελ 36
3.1.10 Έκδοση Android 4.1/4.2/4.3/ (Jelly Bean).....	σελ 37
3.1.11 Έκδοση Android 4.4 (Kit Kat).....	σελ 37
3.1.12 Έκδοση Android 5.0/5.0.2/5.1/5.1.1 (Lollipop).....	σελ 38
3.1.13 Έκδοση Android 6.0/6.0.1 (Marshmallow).....	σελ 39
3.1.14 Έκδοση Android 7.0 (Nougat).....	σελ 41
ΚΕΦΑΛΑΙΟ 4^ο	σελ 41
4.1 ΑΠΑΙΤΗΣΕΙΣ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ	σελ 41
4.1.1 Υποστήριξη πολλών συσκευών.....	σελ 41
4.1.2 Υποστήριξη διαφορετικών διαστάσεων οθόνης.....	σελ 42
4.1.3 Υποστήριξη παλιότερων εκδόσεων.....	σελ 42
4.2 ΒΗΜΑΤΑ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ	σελ 42
4.2.1 Εγκατάσταση Λογισμικού.....	σελ 42
4.2.2 Ανάπτυξη.....	σελ 43
4.2.3 Αποσφαλμάτωση και δοκιμαστική φάση.....	σελ 43
4.2.4 Δημοσίευση.....	σελ 44

4.3 ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ.....σελ 44	
4.3.1 Android Sdk.....σελ 44	
4.3.2 Eclipse Ide.....σελ 45	
4.3.3 Android Studio.....σελ 45	
4.3.4 Εγκατάσταση του Android Studio.....σελ 46	
ΚΕΦΑΛΑΙΟ 5^ο.....σελ 53	
5.1 ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ ΜΙΑΣ ΕΦΑΡΜΟΓΗΣ.....σελ 53	
5.1.1 Activities.....σελ 53	
5.1.2 Services.....σελ 53	
5.1.3 Content Providers.....σελ 53	
5.1.4 Broadcast Receivers.....σελ 53	
5.2 ΕΦΑΡΜΟΓΗ ΠΡΩΤΕΣ ΒΟΗΘΕΙΕΣ (FIRST AID).....σελ 53	
5.3 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΠΡΩΤΩΝ ΒΟΗΘΕΙΩΝ (FIRST AID).....σελ 55	
ΚΕΦΑΛΑΙΟ 6^ο.....σελ 68	
6.1 ΔΟΜΗ ΕΦΑΡΜΟΓΗΣ ΠΡΩΤΩΝ ΒΟΗΘΕΙΩΝ(FIRST AID).....σελ 68	
6.2 ΦΑΚΕΛΟΣ RES.....σελ 68	
6.3 ΥΠΟΦΑΚΕΛΟΣ RES/DRAWABLE.....σελ 68	
6.4 ΥΠΟΦΑΚΕΛΟΣ RAW.....σελ 70	
6.5 ΔΙΑΦΟΡΟΙ ΤΥΠΟΙ ΠΟΡΩΝσελ 70	
6.5.1 Colors.xml.....σελ 71	
6.5.2 Dimens.xml.....σελ 71	
6.5.3 Google_maps_api.xml.....σελ 71	
6.5.4 Strings.xml.....σελ 71	
6.5.5 Styles.xml.....σελ 71	
6.6 ΥΠΟΦΑΚΕΛΟΣ RES/LAYOUT.....σελ 70	
6.7 ΦΑΚΕΛΟΣ JAVA.....σελ 73	
6.8 ΑΡΧΕΙΟ ANDROIDMANIFEST.....σελ 74	
6.9 ΦΑΚΕΛΟΣ BIN.....σελ 74	
ΚΕΦΑΛΑΙΟ 7^ο.....σελ 74	
7.1 ΑΝΑΛΥΣΗ ΚΩΔΙΚΑ ΒΟΗΘΗΤΙΚΩΝ ΚΛΑΣΕΩΝ.....σελ 74	
7.1.1 Βοηθητικές κλάσεις.....σελ 74	
7.1.2 mMySQL.....σελ 75	
7.2 ΑΝΑΛΥΣΗ ΚΩΔΙΚΑ ΚΥΡΙΩΝ ΚΛΑΣΕΩΝ.....σελ 75	
7.2.1 Android Manifest.....σελ 75	
7.2.2 ActivityMain.xml.....σελ 77	
ΚΕΦΑΛΑΙΟ 8^ο.....σελ 77	
8.1 ΣΥΜΠΕΡΑΣΜΑΤΑ.....σελ 77	
8.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ.....σελ 77	
ΚΕΦΑΛΑΙΟ 9^ο.....σελ 78	
9.1 ΒΙΒΛΙΟΓΡΑΦΙΑσελ 78	
9.1.1 Ελληνική Βιβλιογραφία.....σελ 78	
9.1.2 Ξενόγλωσση Βιβλιογραφία.....σελ 78	
9.1.3 Ηλεκτρονική Βιβλιογραφία.....σελ 78	
ΚΕΦΑΛΑΙΟ 10^ο.....σελ 80	
10.1 ΠΑΡΑΡΤΗΜΑ- ΚΩΔΙΚΕΣ.....σελ 80	
10.1.1 Colors.xml.....σελ 80	
10.1.2 Dimens.xml.....σελ 80	
10.1.3 Google_maps_api.xml.....σελ 80	
10.1.4 Strings.xml.....σελ 81	
10.1.5 Styles.xml.....σελ 82	
10.1.6 Aimoragia.java.....σελ 82	
10.1.7 Apofraksiaeragwgu.java.....σελ 83	
10.1.8 aRegisterRequest.java.....σελ 84	
10.1.9 Axinos.java.....σελ 85	

10.1.10 BasicActivity.java.....	σελ 86
10.1.11 Epeigontaefhmer.java.....	σελ 87
10.1.12 Epilipikikr.java.....	σελ 88
10.1.13 Fidi.java.....	σελ 89
10.1.14 Istorikosfakelos.java.....	σελ 70
10.1.15 Kardio.java.....	σελ 93
10.1.16 Katagma.java.....	σελ 94
10.1.17 Krpanikou.java.....	σελ 95
10.1.18 Lipothimia.java.....	σελ 96
10.1.19 LoginRequest.java.....	σελ 97
10.1.20 MainActivity.java.....	σελ 98
10.1.21 Methis.java.....	σελ 101
10.1.22 Pnigmos.java.....	σελ 102
10.1.23 Protesb.java.....	σελ 103
10.1.24 Psixoypo.java.....	σελ 107
10.1.25 RegisterActivity.java.....	σελ 109
10.1.26 RegisterRequest.java.....	σελ 110
10.1.27 Rinoragia.java.....	σελ 111
10.1.28 Stoixeia.java.....	σελ 112
10.1.29 Thermopliksia.java.....	σελ 114
10.1.30 Topothesia.java.....	σελ 115
10.1.31 Trofikid.java.....	σελ 116
10.1.32 Tsimpouri.java.....	σελ 117
10.1.33 Tsouextra.java.....	σελ 118
10.1.34 UserAreaActivity.java.....	σελ 119
10.1.35 Yprothermia.java.....	σελ 125
10.1.36 Connector.java.....	σελ 126
10.1.37 DataParser.java.....	σελ 127
10.1.38 Downloader.java.....	σελ 129
10.1.39 MyAdapter.java.....	σελ 131
10.1.40 MyHolder.java.....	σελ 131
10.1.41 Spacecraft.java.....	σελ 132
10.2 ΚΩΔΙΚΕΣ-ΡΗΡ.....	σελ 132
10.2.1 Register.php.....	σελ 132
10.2.2 Login.php.....	σελ 133
10.2.3 aRegister.php.....	σελ 134
10.2.4 nRegister.php.....	σελ 135
10.3 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ.....	σελ 136
10.3.1 First Aid.sql.....	σελ 136
10.3.2 IatrikosFakelos.sql.....	σελ 137
10.3.3 PhpMyAdmin.jpeg.....	σελ 139
10.4 ΚΩΔΙΚΑΣ-ANDROID MANIFEST.....	σελ 141
10.4.1 AndroidManifest.xml.....	σελ 141
ΚΕΦΑΛΑΙΟ 11°.....	σελ 142
11.1 ΑΝΑΣΚΟΠΗΣΗ ΠΕΔΙΟΥ.....	σελ 142

ΚΕΦΑΛΑΙΟ 1°

1.1 ΕΙΣΑΓΩΓΗ-ΚΙΝΗΤΑ ΤΗΛΕΦΩΝΑ

Μετά την λήξη του Β' Παγκοσμίου πολέμου ξεκίνησαν οι πρώτες προσπάθειες από τους Σουηδούς, Φιλανδούς, και Αμερικάνους να δημιουργήσουν μια συσκευή η οποία δεν θα εξαρτάται από καλωδιακή σύνδεση με δίκτυο παροχής τηλεφωνίας, ούτε από κάποια τοπική ασύρματη συσκευή εκπομπής ραδιοφωνικού σήματος χαμηλής συχνότητας.

Έτσι γεννάται το κινητό τηλέφωνο στις 3 Απριλίου του 1973 με την κατασκευή και την έναρξη της χρήσης από το κοινό. Το πρώτο σύγχρονο κινητό τηλέφωνο δημιουργήθηκε από την Motorola με το κωδικό MotorolaDynaTac με χαρακτηριστικά, ύψος 25 εκατοστά και βάρος 900 γραμμάρια, η πρώτη γενιά (1G) των κινητών τηλεφώνων ήταν γεγονός. Τα χαρακτηριστικά της πρώτης γενιάς κινητών τηλεφώνων ήταν μεγάλες και ογκώδεις συσκευές, με χαμηλότερες ταχύτητες, χαμηλή ποιότητα όσον αφορά την μετάδοση της φωνής η οποία γινόταν με αναλογικό τρόπο αλλά και με πολλά προβλήματα σύνδεσης.

Στην αρχή της δεκαετίας του 90' άρχισε η απογείωση των κινητών τηλεφώνων, με την ψηφιοποίηση δικτύων (GSM) και συσκευών. Τα κινητά τηλέφωνα ήταν ελαφρύτερα κατά 100 έως 200 γραμμάρια, χωρούσαν στην παλάμη και έμπαιναν έστω και με δυσκολία στην τσέπη του χρήστη τους. Πέρασαμε έτσι στα κινητά δεύτερης γενιάς (2G) όπου παρείχαν περισσότερες λειτουργίες όπως αναγνώριση κλήσης, αποστολή γραπτών μηνυμάτων καθώς και λήψη φωτογραφιών.

Στις αρχές του 21^{ου} αιώνα έπειτα από την αλματώδη πορεία των κινητών τηλεφώνων προχωράμε στα κινητά τρίτης γενιάς (3G), με τις απεριόριστες δυνατότητες πολυμέσων. Πιο συγκεκριμένα τα κινητά τηλέφωνα συνδέονται στο διαδίκτυο, μπορούν να πραγματοποιήσουν βιντεοκλήσεις καθώς είναι εφικτή και η παρακολούθηση τηλεοπτικών προγραμμάτων.

Τα τελευταία χρόνια τα κινητά τηλέφωνα έχουν προχωρήσει στην τέταρτη γενιά (4G) τα οποία διαθέτουν την πιο εξελιγμένη μέχρι τώρα τεχνολογία. Γενικά οι χρήστες αυτών των κινητών τηλεφώνων έχουν πολύ πιο γρήγορη πρόσβαση στο διαδίκτυο η οποία θα δώσει μια δυναμική στις εφαρμογές της κινητής τηλεφωνίας, τόσο σε υπηρεσίες (μετεωρολογικές προβλέψεις, GPS κτλ.) όσο και σε περιεχόμενο (ειδήσεις, ραδιόφωνο, παιχνίδια κτλ.). Τα κινητά τηλέφωνα τέταρτης γενιάς (4G) είναι ελαφρύτερα με πολλές δυνατότητες όπως διαθέτουν ενσωματωμένο Gps, Bluetooth, Wi-Fi και με μεγαλύτερες ταχύτητες καθώς περιλαμβάνουν πολυπύρηνους επεξεργαστές με μεγάλες ταχύτητες και εκτελούν πολλές εντολές ταυτόχρονα, διαθέτουν οθόνες τελευταίας τεχνολογίας σε πολλές διαστάσεις ανάλογα με τις ανάγκες των χρηστών. Αποτελούν στην ουσία κινητά τηλέφωνα με δυνατότητες ενός Ηλεκτρονικού Υπολογιστή. Η πρώτη εταιρεία που έφερε στο προσκήνιο κινητά τέταρτης γενιάς ήταν η Apple.

Για να υποστηριχθούν όλα τα παραπάνω θα έπρεπε να αναπτυχθούν λειτουργικά συστήματα όπου να μπορούν να υποστηρίξουν όλες αυτές τις δυνατότητες των νέων έξυπνων τηλεφώνων (smartphones).

1.2 ΤΙ ΕΙΝΑΙ ΕΞΥΠΝΟ ΤΗΛΕΦΩΝΟ

Το έξυπνο τηλέφωνο (smartphone) είναι ένα κινητό τηλέφωνο βασισμένο σε λειτουργικό σύστημα κινητής τηλεφωνίας με περισσότερη προηγμένη υπολογιστική ικανότητα και συνδεσιμότητα σε σχέση με ένα απλό κινητό τηλέφωνο. Οι κλασσικές συσκευές κινητής τηλεφωνίας αποτελούν τον «πρόγονο» και τα έξυπνα κινητά τηλέφωνα (smartphones), αποτελούν την φυσική εξέλιξη των προκατόχων τους είτε τον «απόγονο» τους.

Ένα έξυπνο κινητό τηλέφωνο είναι ένα «σούπερ» τηλέφωνο που μπορεί ανάλογα με τα χαρακτηριστικά, δυνατότητες, λειτουργικό αλλά και επιθυμίες και ανάγκες του χρήστη να διευκολύνει την καθημερινότητα του, προσφέροντας του μια πληθώρα από ευκολίες που μέχρι προσφάτως αυτά τα έβρισκε ο χρήστης εάν ήταν κάτοχος ηλεκτρονικού υπολογιστή, όπως την επεξεργασία κειμένων, την λήψη και αποστολή μηνυμάτων ηλεκτρονικού ταχυδρομείου (e-mail), καθώς και την πλοήγηση στο διαδίκτυο.

Φυσικά, οι ανάγκες και οι απαιτήσεις του αγοραστικού κοινού αυξήθηκαν με αποτέλεσμα να προστεθούν σε πολλά μεταγενέστερα μοντέλα περισσότερες δυνατότητες, όπως σύγχρονες ψηφιακές φωτογραφικές μηχανές, οι λειτουργίες των φορητών media players, οι βιντεοκάμερες τσέπης, σχεδόν όλα τα κινητά τηλέφωνα διαθέτουν και αισθητήρες κίνησης (Accelerometer) προσφέροντας έτσι την δυνατότητα για πιο εξελιγμένο και φυσικό χειρισμό της συσκευής, καθώς και μονάδες πλοήγησης Gps, με αποτέλεσμα το έξυπνο κινητό τηλέφωνο να διαμορφωθεί σε μια πολυχρηστική συσκευή.

Η συνεχής πρόσβαση στο διαδίκτυο μέσω έξυπνων κινητών συσκευών έχουν αλλάξει τα δεδομένα στην πληροφόρηση και στην διασκέδαση. Η πλοήγηση μέσω έξυπνων συσκευών κινητής τηλεφωνίας είναι μεγαλύτερη σε σχέση από κάποιον Ηλεκτρονικό Υπολογιστή ή από κάποιο Τάμπλετ.

1.3 ΥΛΙΚΟ ΤΩΝ ΕΞΥΠΝΩΝ ΚΙΝΗΤΩΝ

1.3.1 Γενικά

Μια βόλτα στην αγορά θα μας πείσει ότι υπάρχουν πολλά έξυπνα κινητά τηλέφωνα με διάφορα χαρακτηριστικά και δυνατότητες, παρακάτω θα αναφερθούμε στις δυνατότητες των έξυπνων κινητών τηλεφώνων όσον αφορά το υλικό κομμάτι (Hardware) με σκοπό να διαλευκάνουμε κάποιες σημαντικές διαφορές έτσι ώστε να βοηθήσουμε κάποιον που ενδιαφέρεται να γίνει ιδιοκτήτης ενός Smartphone και θα έπρεπε να γνωρίζει ποια χαρακτηριστικά στο Hardware τους είναι σημαντικά για αυτόν.

1.3.2 Επεξεργαστής

Αρχικά θα αναφερθούμε στους επεξεργαστές, οι σύγχρονοι επεξεργαστές των κινητών τηλεφώνων είναι πολυπύρρηνοι με πολύ μεγάλες ταχύτητες, ποιο συγκεκριμένα ένας επεξεργαστής μπορεί να είναι διπύρηνος και να φτάνει έως δέκα πυρήνες όπως το έξυπνο κινητό Xiaomi Redmi Note 4, ενώ παράλληλα βλέπουμε ότι η λειτουργία συχνότητας αυξάνεται συνεχώς και μπορεί να κυμανθεί από 1.0 Ghz έως 2.3

Ghz.Οι σύγχρονοι επεξεργαστές υποστηρίζουν το Multitasking δηλαδή την παράλληλη εξυπηρέτηση ολοένα και περισσότερων λειτουργιών.

1.3.3 Μνήμη

Η ανάγκη για συσκευές με περισσότερη μνήμη αποθήκευσης δημιουργήθηκε εξαιτίας των απαιτητικών λειτουργιών των Smartphones. Η μνήμη αυτή ποικίλει από συσκευή σε συσκευή και μπορεί να φτάσει σε μεγάλη ισάζια ενός επιτραπέζιου Ηλεκτρονικού Υπολογιστή, σήμερα οι μνήμες ξεκινάνε από 512 mb μέχρι και 3 Gb.

Η μνήμη είναι απαραίτητη διότι ένα σύγχρονο κινητό τηλέφωνο δεν χρησιμοποιείται μόνο για την αναπαραγωγή πολυμέσων αλλά χρησιμοποιείται και για παιχνίδια αλλά και για επεξεργασία των περισσότερων μορφών δεδομένων (αρχεία κειμένου, εικόνας, ήχου, βίντεο).Όσο περισσότερη είναι η μνήμη στο κινητό μας τόσο καλύτερα ανταποκρίνεται το κινητό στο άνοιγμα εφαρμογών καθώς και στην λειτουργία τους.

1.3.4 Είσοδος

Εάν ανατρέξουμε στο παρελθόν και κάνουμε μια σύγκριση μεταξύ ενός «απλού» κινητού και ενός «έξυπνου» κινητού θα παρατηρήσουμε αμέσως ότι ο χειρισμός του τηλεφώνου και η είσοδος των δεδομένων γίνεται μέσω αφής. Οι περισσότερες λειτουργίες πραγματοποιούνται με το άγγιγμα, τα πλήκτρα έχουν καταργηθεί σχεδόν εξ' ολοκλήρου με την μονή εξαίρεση τα υποτυπώδη πλήκτρα οπου είναι συνήθως το κλείσιμο το κινητού, η αύξηση και η μείωση της έντασης καθώς και το κουμπί που μας οδηγεί στην αρχική οθόνη του κινητού.

Η έννοια και η ανάγκη πλέον του πληκτρολογίου έχει σχεδόν εξαλειφθεί ως υποχρεωτική και αποτελεί πλέον προαιρετική καθώς σε κάποιες συσκευές υπάρχει πλήρες συρόμενο πληκτρολόγιο πίσω από την οθόνη του κινητού ή για κάποιες άλλες προσφέρεται ένα αποσπώμενο πληκτρολόγιο σαν επιπλέον αξεσουάρ.

1.3.5 Οθόνη

Ένα άλλο σημαντικό χαρακτηριστικό ενός «απλού» και ενός «έξυπνου» κινητού τηλεφώνου είναι η οθόνη. Το εικονικό πληκτρολόγιο (αφής), η αναπαραγωγή των πολυμέσων, οι λίστες με τα μηνύματα, οι χάρτες και τα παράθυρα του φυλλομετρητή φιλοξενούνται στην διαγώνιο της οθόνης.

Οι οθόνες στα κινητά τηλέφωνα ποικίλουν και μπορεί να είναι από 2.8 ίντσες έως 10 ίντσες αναλόγως την συσκευή, επίσης ιδιαίτερη σημασία έχει η τεχνολογία οθόνης καθώς επίσης και η ανάλυση της οθόνης. Πιο συγκεκριμένα όσο μεγαλύτερη είναι η ανάλυση της οθόνης τόσο πιο καθαρό θα παρουσιάζεται και το περιεχόμενο όπως φωτογραφίες, βίντεο, οι συχνότερες για οθόνες Smartphone ποικίλουν από 240 x 320(QVGA) pixels έως 1080 x 1920 (FULL HD).Τα τηλέφωνα Android, έχουν συνήθως ανάλυση ίση ή μεγαλύτερη από 1280 x 720 (720p HD).

Ιδιαίτερη σημασία έχει η φωτεινότητα και κυρίως ποσό ευανάγνωστο είναι το περιεχόμενο κάτω από το φως του ηλίου. Μια σημαντική παράμετρος που διαφοροποιεί τις οθόνες είναι η ποιότητα κατασκευής με ποιο σημαντική αυτή του αθραύστου γυαλιού που συναντάμε στα ακριβά Smartphone. Οποία συσκευή έχει οθόνη με κρύσταλλο Corning Gorilla Glass είναι σιγουρά πολύ ανθεκτική, αντέχει την σκληρή μεταχείριση και ως ένα βαθμό την αδεξιότητα.

Τεχνολογίες όπως το Multi-Touch (εκτέλεση λειτουργίας με παραπάνω από ένα ταυτόχρονο άγγιγμα της οθόνης) δημιουργούν νέες δυνατότητες και λειτουργίες σε ένα έξυπνο κινητό τηλέφωνο αφού πλέον η οθόνη δεν σημαίνει μόνο αναπαραγωγή του περιεχομένου αλλά και είσοδος με την αφή.

1.3.6 Ασύρματη Δικτύωση

Η ευκολία πρόσβασης στο διαδίκτυο είτε μέσω τηλεπικοινωνιακών δικτύων (3G,4G,4G+) είτε μέσω ασυρμάτων δικτύων Wi-Fi είναι μια άλλη σημαντική διάφορα. Ο συνδυασμός μιας πληθώρας εφαρμογών με την γρήγορη πλοήγηση στο διαδίκτυο, ανάλογα βέβαια με το λειτουργικό που διαθέτει η συσκευή, προσφέρει άμεση πρόσβαση σε σελίδες κοινωνικής δικτύωσης αλλά και γενικότερα προσφέρει πρόσβαση σε όλες τις γνωστές σελίδες αναπαραγωγής πολυμέσων και διαδικτύου.

1.3.7 Φωτογραφική Μηχανή

Σχεδόν όλα τα έξυπνα κινητά τηλέφωνα είναι εξοπλισμένα με φωτογραφική μηχανή. Ο χρήστης έχει την δυνατότητα να τραβήξει εικόνες και βίντεο υψηλής ανάλυσης και ευκρίνειας. Στο σημείο αυτό, θα μπορούσαμε να σχολιάσουμε ότι η συνεχής βελτίωση των συσκευών και παράλληλα αύξησης δυνατοτήτων του έχει ως αποτέλεσμα πολλές φορές οι δυνατότητες μιας απλής φωτογραφικής μηχανής να ξεπερνιούνται από τις αντίστοιχες των ενσωματωμένων ψηφιακών μηχανών. Πολλές έξυπνες συσκευές προσφέρουν την δυνατότητα στον χρήστη πραγματοποίησης τηλεδιασκέψεων είτε παρακολούθησης βίντεο μέσω μιας δευτερεύουσας κάμερας χαμηλότερης ανάλυσης στην πλευρά της οθόνης που έχουν ενσωματώσει.

1.3.8 Υλικό Εύρεσης Θέσης

Η υποστήριξη λειτουργιών πλοήγησης με χάρτες (Google Maps) είτε με GPS (σύστημα εντοπισμού θέσης), είτε με A-GPS (σύστημα εντοπισμού θέσης με την βοήθεια του δικτυού κινητής τηλεφωνίας) είναι μια χρήσιμη λειτουργία που παρέχεται σχεδόν σε όλα τα έξυπνα κινητά τηλέφωνα.

1.3.9 Συνδεσιμότητα

Μια άλλη σημαντική παράμετρος που διαφοροποιεί τις συσκευές αυτές, είναι η πληθώρα επιλογών συνδεσιμότητας (Bluetooth, Καλώδιο σύνδεσης Usb, TV-Out) με άλλες ηλεκτρονικές συσκευές, πράγμα που διευκολύνει την διαχείριση και την αναπαραγωγή του περιεχομένου τους.

1.4 ΕΚΔΟΣΕΙΣ ΛΕΙΤΟΥΡΓΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Όσον αφορά την χρήση των έξυπνων κινητών τηλεφώνων πολλές εταιρίες έχουν δημιουργήσει το δικό τους λειτουργικό σύστημα και το οποίο το έχουν ενσωματώσει στα έξυπνα κινητά τηλέφωνα, πιο συγκεκριμένα η κάθε εταιρία προσπαθεί να κερδίσει το δικό της μερίδιο στην αγορά. Λειτουργικά όπως το IOS της Apple το Firefox OS της Mozilla, το Windows Phone της Microsoft και το Blackberry OS της Blackberry και άλλες μικρότερες εταιρίες, ανταγωνίζονται για να αυξήσουν το μερίδιο τους και να «κλέψουν» κάτι από το μερίδιο του Android OS της Google. Παρακάτω θα δούμε αναλυτικότερα σε τι ποσοστό καλύπτουν την αγορά τα λειτουργικά συστήματα τώρα.

About OS								
Feature	iOS	Android	Firefox OS	Windows Phone	BlackBerry 10	Tizen	Sailfish OS	Ubuntu Touch
Company	Apple Inc.	Open Handset Alliance	Mozilla Foundation	Microsoft	BlackBerry Ltd.	Linux Foundation, Tizen Association, Samsung, Intel	Sailfish Alliance, Mer, Jolla and Sailfish community contributors	Canonical Ltd. and Ubuntu community contributors
Market share ^[1]	14.8%	84.1%	N/A	0.7%	0.2%	N/A	N/A	N/A
Current version	9.3.4	7.0	2.2.0	Windows Mobile 10(10.0.10586.420) ^[2]	10.3.2.2876	2.4	2.0.1.11	15.04 (OTA-10)

Εικόνα 1: Εκδόσεις Λειτουργικών Συστημάτων & Μερίδιο στην Αγορά
(https://en.wikipedia.org/wiki/Comparison_of_mobile_operating_systems)

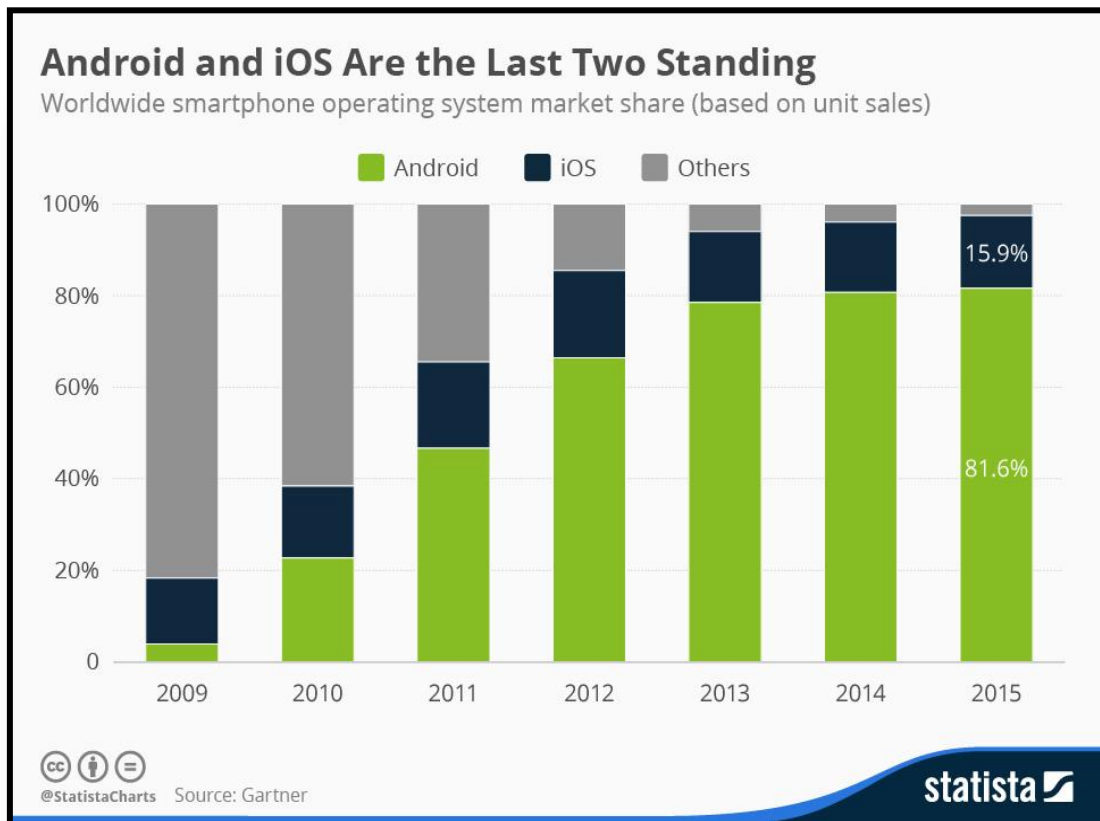
Βλέποντας το παραπάνω πίνακα παρατηρούμε ότι το μεγαλύτερο μερίδιο της αγοράς το καλύπτει το Android με ποσοστό 84.1%, δεύτερο έρχεται το λειτουργικό IOS της Apple με ποσοστό 14.8% και τέλος ακολουθεί το Windows Phone με 0.7% και το Blackberry 10 με ποσοστό 0.2%.

Η ομορφιά του Android είναι η ανοικτή πλατφόρμα που επιτρέπει την εύκολη δημιουργία εφαρμογών, η δυνατότητα πρόσβασης σε αρχεία πολυμέσων εν κίνηση και η μεγάλη ποικιλία των εφαρμογών που είναι διαθέσιμες, είναι αρκετά να τοποθετήσουν το λειτουργικό σύστημα Android στις υψηλές θέσης προτίμησης των χρηστών. Η πρόσβαση σε χιλιάδες Android εφαρμογές που κάθε μέρα γνωρίζει και περισσότερη άνθηση καθώς και το φιλικό προς το χρήστη περιβάλλον (GUI) το κάνει ελκυστικό και ενδιαφέρον.

1.5 ΑΡΙΘΜΟΣ ANDROID APPLICATION

Η ραγδαία ανάπτυξη του Android φαίνεται και από την τρομερή αύξηση του αριθμού των εφαρμογών που γίνονται διαθέσιμες για τους χρήστες καθημερινά στο Google Play. Πιο συγκεκριμένα τον Ιούλιο του 2013 υπήρχαν διαθέσιμες 888.563 εφαρμογές ενώ λίγους μήνες αργότερα ο αριθμός αυτός αυξήθηκε κατά πολύ αγγίζοντας το 1.207.586. Συμπεραίνουμε ότι το Android σαν εργαλείο ανάπτυξης εφαρμογών είναι διαδεδομένο παντού και χρησιμοποιείται από τους περισσότερους Developer.

Παρακάτω βλέπουμε σε στατιστικά αυτή την ραγδαία αύξηση και διαπιστώνουμε οι εφαρμογές Android καλύπτουν μεγάλο ποσοστό της αγοράς, και σύμφωνα με το διάγραμμα τα τελευταία χρονιά αυτή η πρωτιά είναι σταθερή έναντι του IOS της Apple που συνεχώς χάνει έδαφος.













Εικόνα 2: Αριθμός Android Applications

(<http://www.statista.com/statistics/272307/market-share-forecast-for-smartphone-operating-systems/>)

1.6 ΠΩΛΗΣΕΙΣ ΕΞΥΠΝΩΝ ΤΗΛΕΦΩΝΩΝ

Κάτι ανάλογο συμβαίνει και από την μεριά των χρηστών που προτιμούν να χρησιμοποιούν Android Smartphones. Αυτό παρατηρείτε από τι πωλήσεις των έξυπνων κινητών ανά το κόσμο. Πιο συγκεκριμένα με στατιστικά στοιχεία που προκύπτουν το πρώτο τρίμηνο του 2013 παρατηρούμε μεγάλα ποσοστά στη χρήση των κινητών τηλεφώνων με λειτουργικό σύστημα Android, στη Γερμάνια με 73.6%, στη Κίνα με 69.4%, ακόμα και στην Αμερική την πατρίδα της Apple το Android επικρατεί με ποσοστό 49.3% έναντι 43.7%.

Smartphone OS Sales Share (%)				Smartphone OS Sales Share (%)			
Germany 	3 m/e Mar 2012	3 m/e Mar 2013	% pt. Change	USA 	3 m/e Mar 2012	3 m/e Mar 2013	% pt. Change
iOS	18.7	16.9	- 1.8	iOS	44.6	43.7	- 0.9
Android	63.4	73.6	10.2	Android	47.9	49.3	1.4
RIM	1.8	0.5	- 1.3	RIM	2.6	0.9	- 1.7
Symbian	8.2	2.5	- 5.7	Symbian	0.5	0.2	- 0.3
Windows	6.6	6.1	- 0.5	Windows	3.7	5.6	1.9
Other	1.2	0.4	- 0.8	Other	0.6	0.3	- 0.3
GB 	3 m/e Mar 2012	3 m/e Mar 2013	% pt. Change	China 	3 m/e Mar 2012	3 m/e Mar 2013	% pt. Change
iOS	30.1	28.7	- 1.4	iOS		24.6	
Android	49.3	58.4	9.1	Android		69.4	
RIM	15.3	5.1	- 10.2	RIM		0.3	
Symbian	2.1	0.6	- 1.5	Symbian		2.9	
Windows	2.9	7.0	4.1	Windows		2.0	
Other	0.2	0.3	0.1	Other		0.7	
France 	3 m/e Mar 2012	3 m/e Mar 2013	% pt. Change	Australia 	3 m/e Mar 2012	3 m/e Mar 2013	% pt. Change
iOS	20.5	21.2	0.7	iOS	33.8	31.0	- 2.8
Android	57.8	63.3	5.5	Android	52.9	61.7	8.8
RIM	7.4	4.0	- 3.4	RIM	0.8	0.5	- 0.3
Symbian	4.3	0.6	- 3.7	Symbian	6.7	1.1	- 5.6
Windows	2.8	7.2	4.4	Windows	3.3	4.1	0.8
Other	7.2	3.6	- 3.6	Other	2.6	1.5	- 1.1
Italy 	3 m/e Mar 2012	3 m/e Mar 2013	% pt. Change	Japan 	3 m/e Mar 2012	3 m/e Mar 2013	% pt. Change
iOS	22.6	19.9	- 2.7	iOS		49.2	
Android	48.4	62.5	14.1	Android		45.8	
RIM	3.7	2.5	- 1.2	RIM		0.7	
Symbian	17.7	3.5	- 14.2	Symbian		0.2	
Windows	5.9	10.9	5.0	Windows		0.3	
Other	1.6	0.7	- 0.9	Other		3.7	
Spain 	3 m/e Mar 2012	3 m/e Mar 2013	% pt. Change	EUS 	3 m/e Mar 2012	3 m/e Mar 2013	% pt. Change
iOS	5.4	3.2	- 2.2	iOS	20.4	19.4	- 1.0
Android	76.8	93.5	16.7	Android	58.1	68.8	10.7
RIM	7.8	0.2	- 7.6	RIM	8.1	2.7	- 5.4
Symbian	8.1	1.5	- 6.6	Symbian	7.2	1.6	- 5.6
Windows	1.8	1.3	- 0.5	Windows	4.1	6.5	2.5
Other	0.0	0.3	0.3	Other	2.2	1.1	- 1.2

Εικόνα 3: Πωλήσεις Έξυπνων Τηλεφώνων ανά Λειτουργικό Σύστημα
<http://www.kumulos.com/2013/05/08/kantar-backend-as-a-service>

Παρότι το Android βρίσκεται κοντά μας από το 2007, αναπτύσσεται συνεχώς, είναι μια τεχνολογία αιχμής και θα μας απασχολήσει και στο μέλλον. Έτσι αποτελεί μεγάλο προσόν για έναν Developer να γνωρίζει την τεχνολογία του Android και πως να δημιουργήσει εφαρμογές για την πλατφόρμα αυτή.

ΚΕΦΑΛΑΙΟ 2°

2.1 ΕΙΣΑΓΩΓΗ-ΤΟ ΛΕΙΤΟΥΡΓΙΚΟ ANDROID

Όπως γνωρίζουμε το λειτουργικό σύστημα android έχει ενσωματωθεί σε κινητές συσκευές, τα οποία διαθέτουν οθόνη αφής, τρέχουν τον πυρήνα (Kernel) του λειτουργικού Linux και επιπλέον επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με τη χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας έτσι την συσκευή μέσω βιβλιοθηκών λογισμικού της Google.



Εικόνα 4: Η οικογένεια των Smartphones

(<https://techcompose.com/wp-content/uploads/2016/05/Android-Family1.png>)

Στη σύγχρονη εποχή υπάρχουν πολλές συσκευές με λειτουργικό σύστημα android, όπου η κάθε μια έχει τα δικά της χαρακτηριστικά. Πιο συγκεκριμένα μεγάλες εταιρίες όπως είναι η Samsung, η HTC, η Motorola, η Sony Ericsson, η LG χρησιμοποιούν το λειτουργικό Android και προσθέτουν στα κινητά τους τηλέφωνα διάφορες ιδιότητες και χαρακτηριστικά με αποτέλεσμα να δημιουργείτε ένας μεγάλος ανταγωνισμός.

2.2 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

Οι Rubin, Miner, Sears και White δημιούργησαν το Android, το 2003 στην Καλιφόρνια. Η δυνατότητα εντοπισμού και αναγνώρισης της τοποθεσίας του ιδιοκτήτη ενός έξυπνου κινητού τηλεφώνου οποιαδήποτε στιγμή και εάν χρειαστεί, ήταν η ανάγκη -αφορμή που οδήγησε στην δημιουργία αυτού.

Αρχικά όλες οι ενέργειες από την μεριά των δημιουργών του Android γινόντουσαν με απολυτή μυστικότητα έως ότου ο Andy Rubin αποφάσισε να συνεργαστεί με την Google ώστε να την χρησιμοποιήσει ως κυρία μηχανή αναζήτησης για το T-Mobile Sidekick.

Έτσι, πραγματοποιήθηκε μια συνάντηση του Andy Rubin με τον Larry Page, ο οποίος ήταν ένας από τους δυο ιδρυτές της Google. Σε αυτή την συνάντηση έγινε η παρουσίαση του Android ως ένα παγκόσμιο ανοιχτό λειτουργικό σύστημα για έξυπνα κινητά τηλέφωνα. Η συνάντηση αυτή είχε ως αποτέλεσμα η Google να εξαγοράσει το Android και η εποχή της μεγάλης ανάπτυξης του ξεκίνησε.

Παρ' όλο που το Android είχε περάσει στα χεριά της Google η εταιρεία συνέχισε να κρατάει μυστικό το λειτουργικό σύστημα καθώς δούλευε πάνω σε αυτό ,το βελτίωνε και κατοχύρωνε πατέντες και έκανε επαγγελματικές συναντήσεις ψάχνοντας για συνεργάτες και επενδυτές μέχρι το 2007 όπου η εποχή του απλού κινητού τηλεφώνου ως αναγκαίο μέσο επικοινωνίας σε συνδυασμό με τις δυνατότητες που μπορούσε να προσφέρει ένα έξυπνο κινητό τηλέφωνο, θα μπορούσε να κατακλίσει την αγορά και να ανταγωνιστεί για πολλούς χρήστες και την ανάγκη αγοράς ενός υπολογιστή, πράγμα που επιβεβαιώνεται καθημερινά εάν παρατηρηθούν οι πωλήσεις των εκάστοτε προϊόντων.



Εικόνα 5: Υπάρχει μια παραδοσιακή αντιμαχία ανάμεσα στο Android και την Apple (<http://flameofudun.net/forum/general/38355-android-vs-apple>)

Παράλληλα, η συγύρια του αιώνιου αντίπαλου, ενώνοντας την Apple, η οποία το καλοκαίρι του 2007, προωθεί στην αγορά το iPhone. Ο επιχειρηματικός και ο τεχνολογικός κόσμος νομίζει ότι η Google θα απαντήσει με το gPhone πράγμα που δεν έγινε. Το φθινόπωρο του 2007, ανακοινώνεται το μυστικό καθώς και οι μυστικές συνομιλίες στο κόσμο της τεχνολογίας και στην αγορά αλλά όχι έτσι απλά. Πιο συγκριμένα, δημιουργείτε ένας συνεταιρισμός εταιρειών με το όνομα Open Handset Alliance, όπου αποτελείτε από πολλούς εκπροσώπους τόσο της Google όσο και της Texas Instruments, της Intel, της T-Mobile, της Sprint Nextel, της HTC, της Samsung, της Qualcomm (κατασκευάστρια εταιρία των επεξεργαστών) και άλλους με σκοπό να μετατρέψουν το Android σε μια πλατφόρμα ανοιχτού κώδικα που θα έχει ενσωματωμένο το λογισμικό Linux και σιγά σιγά να κυκλοφορήσουν στην αγορά τα πρώτα

Smartphones με Android. Εταιρείες όπως η HTC, η Motorola και η LG, ανακοινώνουν ότι πρόκειται να υπάρξουν στην αγορά τα Smartphones, όπου θα διαθέτουν λειτουργικό σύστημα Android άλλα θα είναι κατάλληλα προσαρμοσμένες στις ανάγκες του αγοραστικού τους κοινού τόσο όσο αφορά τον καλλωπισμό σε διαφορά σχήματα και μεγέθη όσο και τις δυνατότητες που μπορεί να προσφέρει μέσω των πολλαπλών εφαρμογών που θα μπορούν να υλοποιηθούν και να υποστηριχθούν από αυτά, προσαρμοσμένες σε οποιαδήποτε ανάγκη και αν έχει.

Η HTC κυκλοφορεί ένα χρόνο μετά το πρώτο Smartphone με λειτουργικό σύστημα Android το οποίο ονομαζόταν HTC Dream. Στη συνέχεια ακολουθήσαν και οι υπόλοιπες εταιρείες καθώς και δεκάδες άλλες που εμπιστευτήκαν το Android ως λειτουργικό σύστημα και κατάφεραν να το κάνουν το κυρίαρχο λειτουργικό αυτή την στιγμή στην αγορά. Η συνεργασία με την Samsung, η οποία είναι η πρώτη εταιρεία σε πωλήσεις κινητών παγκοσμίως αυτή την στιγμή κατάφερε και εδραίωσε το Android, και η εξάπλωση του οφείλετε κατά κύριο λόγο σε αυτήν. Από εκεί και μετά η ιστορία είναι λίγο πολύ γνωστή σε όλους μας.

2.3 ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ANDROID

Οι συσκευές που χρησιμοποιούν Android είναι Multimedia, δηλαδή παρέχουν την δυνατότητα να αναπαράγουν πολλαπλά μέσα όπως ήχο, εικόνα, βίντεο, καθώς και Multitasking δηλαδή δίνουν την δυνατότητα εκτέλεσης πολλών εφαρμογών ταυτόχρονα, όπως παραδείγματος χάριν οι χρήστες των συσκευών να σερφάρουν στο διαδίκτυο ενώ παράλληλα μπορούν να ακούσουν μουσική ή να δεχτούν μια κλήση χωρίς να χρειαστεί να κλείσουν καμία εφαρμογή ή να χάσουν την σελίδα που επισκέπτονται στο διαδίκτυο.

Το Android παρέχει την δυνατότητα κάλυψης αναγκών ενός απαιτητικού χρήστη σε επίπεδο υψηλό όσον αφορά το Web Browsing καθώς είναι ταχύτατο, μπορεί να υποστηρίξει τόσο Flash όσο και την χρήση οποιουδήποτε φυλλομετρητή προτιμά. Σχεδόν όλες οι συσκευές Android διαθέτουν ενσωματωμένο Gps καθώς και Wi-fi, ανεξαρτήτως του κόστους της συσκευής και δικαιολογώντας έτσι τον βασικό λόγο «γέννησης» του λειτουργικού συστήματος Android, που είναι η εύκολη γρήγορη και ανεμπόδιστη πρόσβαση στο διαδίκτυο σε συνδυασμό με πάμπολες εφαρμογές (Apps), όπως αναζήτηση, Chat, E-mail, και χάρτες που επιτρέπουν στο χρήστη να μένει διαρκώς δικτυωμένος και ενημερωμένος.

Επιπλέον, το Android δίνει την δυνατότητα στους χρήστες να προσθέσουν Widgets, δηλαδή εικονίδια για την ταχύτερη πρόσβαση στα προγράμματα, τα οποία τοποθετούνται στην αρχική οθόνη (Home Screen) του κινητού (Launcher).

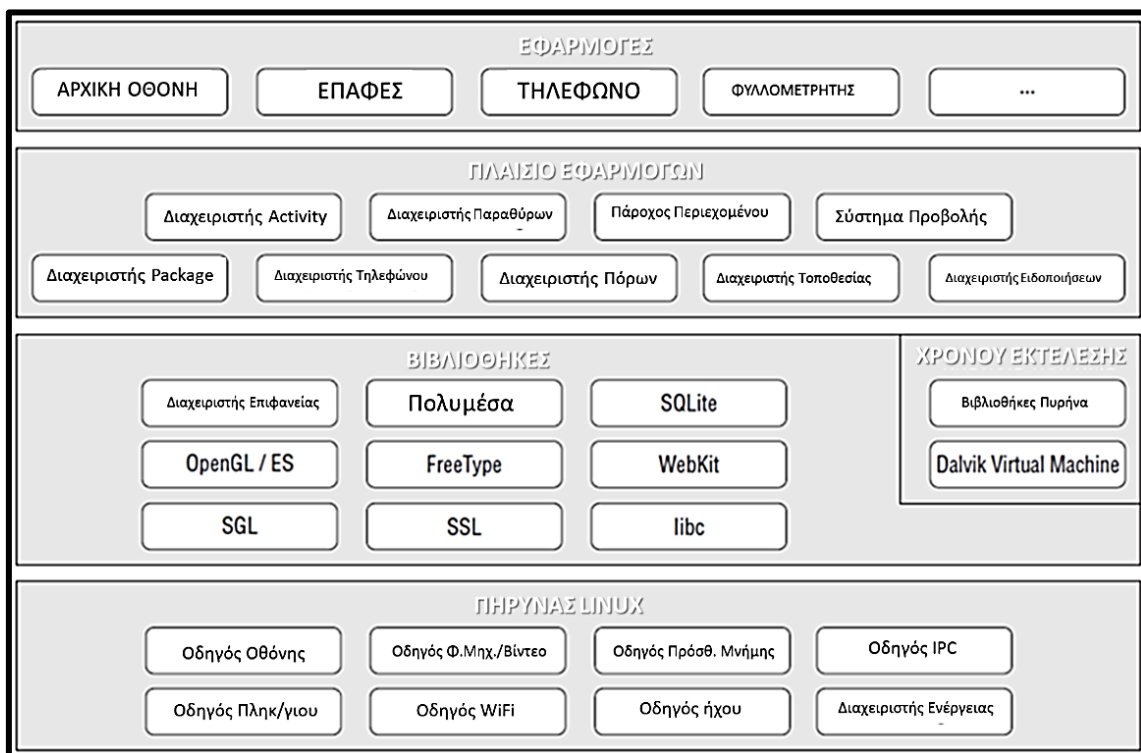
Βασικό χαρακτηριστικό του Android, επίσης είναι η ευκαιρία επιλογής μεταξύ πολλών εφαρμογών που διατηρούν την συνεχή σύνδεση με κοινωνικά δίκτυα όπως Facebook, MySpace, Twitter κλπ.

Επιπλέον εξαιρετικά χρήσιμη είναι η Notification Bar, καθώς με ένα απλό Drag οι χρήστες μπορούν να δουν όλες τις ιδιοποιήσεις σχετικά με τη συσκευή τους, άλλα και τα προγράμματα (Applications) που έχουν εγκαταστήσει εάν έχουν πχ. ενημερώσεις η οτιδήποτε άλλο. Σχετικά με το Υλικό

(Hardware) οι εξελίξεις στην τεχνολογία είναι ραγδαίες καθώς οι διπύρηντοι και τετραπύρηντοι επεξεργαστές καθώς και οι διακεκριμένες GPU είναι πλέον γεγονός.

2.4 Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΩΝ ANDROID

Το Android όπως έχουμε αναφέρει παραπάνω είναι ένα λειτουργικό σύστημα, και σαν λειτουργικό σύστημα ο σκοπός του είναι να παρέχει ένα επίπεδο αφαιρετικότητας ανάμεσα στο υλικό και στον χρήστη. Πιο απλά, ο ρόλος ενός λειτουργικού συστήματος είναι να δίνει την δυνατότητα στον χρήστη να χρησιμοποιεί τους πόρους του συστήματος προς όφελος του με μια διεπαφή, ένα περιβάλλον που είναι ακόμα πιο κατανοητό στον άνθρωπο. Για να το καταφέρει αυτό, το Android αποτελείται από μια στοίβα λογισμικών τμημάτων (Software Stack) με αυστηρά καθορισμένους ρόλους.(Εικόνα 5)



Εικόνα 6: Η δομή της αρχιτεκτονικής του λειτουργικού συστήματος Android (<http://docplayer.gr/docs-images/25/6629751/images/35-0.png>)

Στο κατώτερο επίπεδο βρίσκεται ο πυρήνας του Linux. Ο πυρήνας του Linux είναι γνωστός για την μεγάλη ποικιλία αρχιτεκτονικών επεξεργαστών με τις οποίες είναι συμβατός. Χτίζοντας, λοιπόν, τα υπόλοιπα τμήματα του Android πάνω από το πυρήνα του Linux επιτυγχάνουμε την συμβατότητά του με μια ευρεία γκάμα αρχιτεκτονικών και άρα συσκευών που μπορούν να υποστηρίξουν.

Στο επίπεδο ακριβώς πάνω από το πυρήνα του Linux βρίσκονται οι βασικές βιβλιοθήκες του συστήματος. Εδώ βρίσκονται οι υλοποιήσεις για βιβλιοθήκες γραφικών, για συστήματα βάσεων δεδομένων, η βασική βιβλιοθήκη της γλωσσάς προγραμματισμού C και άλλα. Όλες αυτές οι βιβλιοθήκες «τρέχουν» στο πυρήνα του Linux. Το επίπεδο αυτό προσφέρει τα απαραίτητα στοιχεία ώστε να είναι δυνατή η υλοποίηση του Android Runtime, του βασικού μηχανισμού, δηλαδή που απαιτείται για την εκτέλεση των εφαρμογών που αναπτύσσονται για το περιβάλλον Android.

Το Android Runtime περιλαμβάνει την εικονική μηχανή (Virtual Machine) που διερμηνεύει τις εφαρμογές Android καθώς και την υλοποίηση των βασικών βιβλιοθηκών της Java που προσφέρονται στον προγραμματιστή κατά τη διαδικασία ανάπτυξης των δικών του εφαρμογών.

Βασιζόμενο στο Android Runtime και στις βασικές βιβλιοθήκες του συστήματος, το προτελευταίο επίπεδο είναι το ονομαζόμενο Application Framework. Το επίπεδο αυτό προσφέρει στον προγραμματιστή μια πληθώρα δυνατοτήτων σχετιζόμενων με το λειτουργικό σύστημα και την συσκευή που το φιλοξενεί στο σύνολο τους, όπως πρόσβαση στα διάφορα περιφερικά της συσκευής, πρόσβαση σε λειτουργικότητες του ίδιου του λειτουργικού και άλλα.

Τέλος, το ανώτερο επίπεδο της στοίβας είναι το επίπεδο όπου βρίσκονται οι εφαρμογές για το περιβάλλον του Android. Σε αυτό το επίπεδο αναπτύσσονται οι εφαρμογές που περιγράφονται στο παρόν κείμενο και εδώ μπορούν να αξιοποιηθούν όλες οι δυνατότητες που μας προσφέρουν τα δυο προαναφερθέντα επίπεδα.

2.4.1 Το επίπεδο του πυρήνα του Linux

Το κατώτερο επίπεδο της λογικής στοίβας του Android είναι ο πυρήνας του Linux. Η Google χρησιμοποίησε μια τροποποιημένη μορφή του πυρήνα του Linux έκδοσης 2.6 για το λειτουργικό σύστημα Android. Η επιλογή της αυτή είχε σαν αποτέλεσμα κάθε καινούργια έκδοση του πυρήνα που κυκλοφόρησε να πρέπει να προσαρμοστεί, αντίστοιχα με την έκδοση 2.6, πριν μπορέσουν οι χρήστες του Android να επωφεληθούν από τα νέα χαρακτηριστικά που τυχόν θα έχουν εισαχθεί. Με την έκδοση 3.3 του πυρήνα του Linux, έχει γίνει πολυπόθητη από πολλούς συγχώνευση του κώδικα του Android. Πλέον τα νέα χαρακτηριστικά που εισάγονται στο πυρήνα γίνονται άμεσα διαθέσιμα και στο λειτουργικό σύστημα Android. Επιπλέον, ανοίγουν οι πόρτες για την εκτέλεση του Android ακόμα και σε προσωπικούς υπολογιστές.



Εικόνα 7: Το επίπεδο του πυρήνα του Linux

(<http://docplayer.gr/docs-images/25/6629751/images/35-0.png>)

Ο πυρήνας του Linux (Εικόνα 6) δρα στην περίπτωση του Android ως το επονομαζόμενο επίπεδο αφαίρεσης υλικού (Hardware Abstraction Layer). Χτίζοντας τα υπόλοιπα επίπεδα του Android πάνω από το πυρήνα του Linux διασφαλίζεται ένα πολύ σημαντικό χαρακτηριστικό για το λειτουργικό αυτό σύστημα. Πλέον, αν κάποιος κατασκευαστής κινητών συσκευών επιθυμεί να διαθέσει τις συσκευές του με το λειτουργικό σύστημα Android το μόνο που χρειάζεται, θεωρητικά να κάνει είναι να υλοποιήσει τους οδηγούς που απαιτούνται για τα περιφερικά που διαθέτει κάθε συσκευή του. Όλα τα υπόλοιπα επίπεδα της στοιβάς μπορούν να μείνουν ως έχουν μιας και αυτά βασίζονται αποκλειστικά στο κατώτατο επίπεδο του πυρήνα του Linux και καθόλου στο υπάρχον υλικό (εξίσου και η ονομασία επίπεδο αφαίρεσης υλικού). Στο επίπεδο αυτό ανήκουν οι οδηγοί για την οθόνη, για τις κάρτες μνήμης που μπορεί να διαθέτει η εκάστοτε συσκευή, για τον ήχο, για το υποσύστημα ασύρματης σύνδεσης στο διαδίκτυο (Wi-Fi), για την κάμερα και για όποιο άλλο περιφερειακό συναντάμε σήμερα στις σύγχρονες κινητές συσκευές.

Εκτός από το επίπεδο αφαιρετικότητας που προσφέρει ο πυρήνας, το Android, επωφελείται και από τις μεθόδους διαχείρισης μνήμης, διαχείρισης ενέργειας και διαχείρισης διεργασιών που ακολουθεί το Linux. Οι παρακάτω μέθοδοι έχουν τροποποιηθεί για να χρησιμοποιηθούν στο Android κυρίως γιατί το Android αποτελεί ένα λειτουργικό σύστημα το οποίο καλείτε να «τρέξει» σε κινητές συσκευές οι οποίες εξ ορισμού έχουν περιορισμένους πόρους. Οι αλλαγές που έχουν γίνει μπορούν να κατηγοριοποιηθούν σε διορθώσεις σφαλμάτων του πυρήνα, προσθήκες για βελτίωση της αποδοτικότητας στο πεδίο του χρήστη (Kernel Space), εισαγωγή των νέων δομών διαχείριση και επέκταση της υποστήριξης σε νέες αρχιτεκτονικές (msm7k, msm8k, Nvidia Tegra κ.λπ.)

2.4.2 Το επίπεδο των βασικών βιβλιοθηκών

Οι βασικές βιβλιοθήκες του συστήματος (Εικόνα 7) είναι αυτές που απέχουν τις βασικές λειτουργικότητες όπως αυτή των γραφικών, του ήχου και άλλες. Οι βιβλιοθήκες που βρίσκονται σε αυτό το επίπεδο είναι ουσιαστικά βιβλιοθήκες υλοποιημένες στην γλώσσα προγραμματισμού C ή C++ και τρέχουν, όπως αναφέραμε, πάνω στον πυρήνα του Linux (Εικόνα 6).



Εικόνα 8: Το επίπεδο των βασικών βιβλιοθηκών του συστήματος
(<http://docplayer.gr/docs-images/25/6629751/images/35-0.png>)

Μια από τις βιβλιοθήκες του επιπέδου αυτού είναι ο διαχειριστής σχεδιαστικών επιφανειών (Surface Manager). Ο ρόλος του διαχειριστή αυτού είναι η διαχείριση των σχεδιαστικών επιφανειών

(Drawable Surfaces). Μια σχεδιαστική επιφάνεια μπορεί να είναι η οθόνη μιας συσκευής ή ακόμη και η περιοχή της μνήμης που έχει δεσμευθεί προκειμένου να κρατήσει τα δεδομένα προς απεικόνιση πριν αυτά μεταφερθούν στην πραγματική οθόνη για προβολή. Η διαχείριση των σχεδιαστικών επιφανειών έχει να κάνει με την διαχείριση των παραθύρων των εφαρμογών καθώς και με τον τρόπο τον οποίο αυτές εμφανίζονται στην οθόνη της συσκευής. Ο διαχειριστής σχεδιαστικών επιφανειών σε συνδυασμό με τον διαχειριστή παραθύρων (Window Manager), που βρίσκεται στο επόμενο επίπεδο της στοίβας, είναι υπεύθυνος για την λεγόμενη σύνθεση των παραθύρων των εφαρμογών ανάλογα με την διάταξη που αυτές έχουν στην οθόνη. Πιο συγκεκριμένα, αποφασίζουν με ποια σειρά θα σχεδιαστούν τα παράθυρα των εφαρμογών προκειμένου να επικαλύπτονται με την σωστή αλληλουχία ανάλογα με τις επιλογές του χρήστη. Επιπλέον αν κάποια παράθυρα είναι ημιδιαφανή τότε τα παράθυρα που βρίσκονται από πίσω τους πρέπει να είναι μερικώς ορατά και αυτό είναι επίσης δουλειά των διαχειριστών αυτών.

Μένοντας στο κομμάτι των γραφικών και συμπεριλαμβάνοντας τον ήχο στο τρέχων επίπεδο βρίσκεται το ονομαζόμενο Media Framework του Android. Σε όλες τις σύγχρονες συσκευές αυτό αποτελεί την καρδιά του συστήματος από την άποψη ότι είναι υπεύθυνο για την σωστή αναπαραγωγή ήχου και βίντεο σε μια συσκευή, των δυο πιο δημοφιλών τύπων πολυμέσων σήμερα. Στο συγκεκριμένο Framework βρίσκονται όλες οι διαθέσιμες ρουτίνες για την κωδικοποίηση (ή αποκωδικοποίηση) του ήχου και του βίντεο (ή από) τα διάφορα φορμάτ που αυτά είναι διαθέσιμα. Φορμάτ ήχου όπως AAC, AC-3, AMR, AMR-WB, MP2, MP3, MP4, OGG Vorbis, WMA, RealAudio και ALAC αλλά και φορμάτ βίντεο όπως MPEG1, MPEG2, MPEG4, H.264, VC-1, VC-2, VC-3 όλα μπορούν να αναπαραχθούν από το Android. Βασικό μέρος της εργασίας της συγκεκριμένης βιβλιοθήκης είναι και η διαχείριση μνήμης που απαιτείται κατά την διαδικασία κωδικοποίησης ή αποκωδικοποίησης. Επιπλέον εδώ λαμβάνεται η απόφαση αν θα χρησιμοποιηθεί η πλήρως λογισμική υλοποίηση ενός κωδικοποιητή (ή αποκωδικοποιητή) ή αν η κωδικοποίηση (ή αποκωδικοποίηση) θα υποβοηθηθεί από το υπάρχον υλικό (αν αυτό υποστηρίζει τέτοια δυνατότητα).

Δυο πολύ σημαντικές βιβλιοθήκες στο επίπεδο αυτό είναι η υλοποίηση του προτύπου OpenGL ES και του SGL. Οι δυο αυτές βιβλιοθήκες προσφέρουν την δυνατότητα, στα παραπάνω επίπεδα της στοίβας, σχεδίασης τρισδιάστατων και δισδιάστατων γραφικών αντίστοιχα. Τα πρότυπα αυτά είναι αρκετά εξειδικευμένα και για αυτό δεν θα επεκταθούμε περισσότερο στο πως αυτά λειτουργούν και τι δυνατότητες δίνουν σε κάποιον που αναπτύσσει εφαρμογές για Android. Παρόλα αυτά, αξίζει να σημειωθεί ότι και εδώ, όπως και στην περίπτωση του Media Framework, αν υπάρχει διαθέσιμη υποβοήθηση από το υλικό τότε αυτή χρησιμοποιείται ειδάλλως γίνεται χρήση της πλήρους λογισμικής υλοποίησης κάθε προτύπου.

Παρότι το Android θα μπορούσε να χρησιμοποιήσει μια από τις δυο βιβλιοθήκες γραφικών που διαθέτει για να αποτυπώσει γραμματοσειρές στην οθόνη, επέλεξε να συμπεριλάβει στο επίπεδο αυτό, μια βιβλιοθήκη αποκλειστικά για τον σκοπό αυτό. Αυτή είναι η βιβλιοθήκη FreeType και παρέχει την δυνατότητα για αποτύπωση γραμματοσειρών που αναπαρίστανται από απλά Bitmaps (Raster Fonts) και γραμματοσειρών που περιγράφονται μαθηματικά με καμπύλες Bezier (Vector Fonts). Μια σημαντική διαφορά μεταξύ των δυο τύπων γραμματοσειρών είναι η ικανότητα μεγέθυνσης που η κάθε μια έχει. Στην περίπτωση των Bitmap γραμματοσειρών, η αύξηση του μεγέθους ενός γράμματος σημαίνει εκ νέου

τοποθέτηση των εικονοστοιχείων που το αποτελούν στο νέο μέγεθος και στην συνέχεια προσέγγιση (Interpolation) των εικονοστοιχείων που λείπουν, ενδιάμεσα, με μαθηματικούς υπολογισμούς. Το αποτέλεσμα μετά από αρκετή μεγάλη μεγέθυνση είναι γνωστό σε όλους «πιξέλιασμα» των γραμμμάτων. Στην αντίθετη περίπτωση των διανυσματικών γραμματοσειρών, η μόνη διαθέσιμη πληροφορία για κάθε γράμμα είναι τα σημεία που ορίζουν τις καμπύλες Bezier που συνθέτουν την τελική εμφάνιση του γράμματος. Όλα τα υπόλοιπα εικονοστοιχεία υπολογίζονται κατά την σχεδίαση του γράμματος. Έτσι η ποιότητα απεικόνισης μιας διανυσματικής γραμματοσειράς δεν μειώνεται καθώς αυξάνουμε το μέγεθος της.

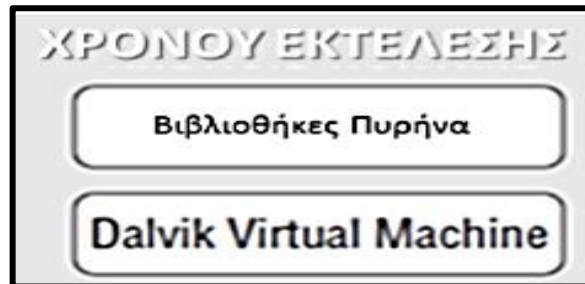
Επειδή στις εφαρμογές για Android συνήθως γίνεται επεξεργασία και αποθήκευση μεγάλου όγκου πληροφοριών το λειτουργικό σύστημα παρέχει μια βιβλιοθήκη που μπορεί να χρησιμοποιηθεί σαν ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων. Με τον όρο σχεσιακή βάση δεδομένων εννοείτε μια συλλογή δεδομένων οργανωμένη σε συσχετισμένους πίνακες που παρέχει ταυτόχρονα ένα μηχανισμό για ανάγνωση, εγγραφή, τροποποίηση ή και πιο πολύπλοκες διαδικασίες πάνω στα δεδομένα. Μια άλλη σημαντική βιβλιοθήκη στο επίπεδο αυτό είναι η μηχανή διάταξης γραφικών (Layout Engine) Web Kit. Ο ρόλος της μηχανής αυτής είναι να δώσει την δυνατότητα σε έναν περιηγητή του διαδικτύου να απεικονίσει τις σελίδες που ο χρήστης επισκέπτεται με γραφικό τρόπο έχοντας στην διάθεση του τον κώδικα HTML που παρέχει ο ιστότοπος. Η Web Kit είναι μια πολύ γνωστή μηχανή διάταξης γραφικών εδώ και αρκετά χρόνια γιατί είναι η μηχανή που χρησιμοποιεί ο περιηγητής της Apple , ο Safari.

Τέλος, η Google υλοποίησε την βασική βιβλιοθήκη της C (Lib) με στόχο την μείωση της απαιτούμενης μνήμης από αυτήν και έχει συμπεριλάβει μια βιβλιοθήκη που υλοποιεί το Open SSL πρότυπο για την διαχείριση πιστοποιητικών κατά την δημιουργία ασφαλών συνδέσεων με απομακρυσμένους εξυπηρετητές.

2.4.3 Το επίπεδο του Android Runtime

Οι εφαρμογές του περιβάλλοντος Android καθώς και πολλές από τις βιβλιοθήκες υψηλού επιπέδου (στο επίπεδο Application Framework) είναι γραμμένες στην γλώσσα προγραμματισμού Java. Ως γνωστόν προγράμματα που είναι γραμμένα σε αυτή την γλώσσα απαιτούν την ύπαρξη διερμηνευτή Java για την εκτέλεση τους. Στο Android δεν έχει χρησιμοποιηθεί ο τυπικός Java διερμηνευτής για ενσωματωμένα συστήματα Java ME αλλά υλοποιήθηκε ένας διερμηνευτής ειδικά για το Android βελτιστοποιημένος για μικρά ενσωματωμένα συστήματα με περιορισμένους πόρους (απαιτεί ένα ελάχιστο 64MB μνήμης). Το χαρακτηριστικό αυτό του Android το διαφοροποιεί από όλα τα άλλα ανταγωνιστικά λειτουργικά συστήματα κινητών συσκευών που χρησιμοποιούν γλώσσες όπως C, C++ και Objective C.

Στο επίπεδο του Android Runtime βρίσκονται και οι βασικές βιβλιοθήκες της Java οι οποίες όμως είναι γραμμένες ειδικά για τον διερμηνευτή του Android (Εικόνα 8).



Εικόνα 9: Το επίπεδο του Android Runtime
(<http://docplayer.gr/docs-images/25/6629751/images/35-0.png>)

Το όνομα του διερμηνευτή του Android είναι Dalvik (Εικόνα 8) που ονομάστηκε έτσι από τον προγραμματιστή που τον ανέπτυξε. Κάθε εφαρμογή στο Android εκτελείτε σε ένα δικό της στιγμιότυπο του Dalvik απομονώνοντας την από τις υπόλοιπες εφαρμογές διασφαλίζοντας έτσι την ευστάθεια και την ασφάλεια του λειτουργικού συστήματος. Τα αρχεία με τα bytecodes που παράγονται για την εκτέλεση από τον Dalvik είναι αρκετά μικρότερα από τα αντίστοιχα Java bytecodes που παράγονται για την εκτέλεση του JVM (Java Virtual Machine) μειώνοντάς έτσι σημαντικά τον χώρο που απαιτεί κάθε εφαρμογή για την εγκατάστασή της. Τα αρχεία εγκατάστασης των εφαρμογών έχουν κατάληξη .apk και περιέχουν τα αρχεία .dex, που περιέχουν τα bytecodes προς εκτέλεση από τον Dalvik, καθώς και διάφορα συνοδευτικά αρχεία που η κάθε εφαρμογή χρειάζεται (εικόνες, ήχους κ.λπ.). Προκειμένου τα .dex αρχεία να είναι μικρά τυχόν δεδομένα που εμφανίζονται πολλές φορές στις κλάσεις μιας εφαρμογής αποθηκεύονται μονάχα μια φορά και όπου χρειάζεται δημιουργούνται αναφορές σε αυτά.

Κατά την εκτέλεση των εφαρμογών στο Android πρέπει να γίνει επαλήθευση της δομής του .dex αρχείου και να εκτελεστούν κάποιες βελτιστοποιήσεις που αφορούν το μέγεθος του και την πολυπλοκότητα του κώδικα του. Το εργαλείο Dexopt είναι αυτό που πραγματοποιεί τις παραπάνω βελτιστοποιήσεις. Η διαδικασία επαλήθευσης ορθότητας της εφαρμογής είναι γενικά χρονοβόρα διαδικασία και απαιτείται να γίνει μόνο μια φορά εκτός και αν τα αρχεία της εφαρμογής αλλάξουν (λόγο αναβάθμισης της εφαρμογής ή επανεγκατάστασης της). Στο Android, ο απαιτούμενος έλεγχος, προτιμάται να γίνεται κατά την εγκατάσταση της εφαρμογής προκειμένου να μειωθεί ο χρόνος της πρώτης εκκίνησης της εφαρμογής από τον χρήστη. Το εργαλείο Dexopt για να επαληθεύσει την ορθότητα της δομής των αρχείων .dex τα φορτώνει σε ένα στιγμιότυπο του διερμηνευτή και ελέγχει όλες τις εντολές, όλων των κλάσεων που αυτό περιέχει προκειμένου να εντοπίσει προβληματικές αλλοιώσεις πριν αυτές δημιουργήσουν πρόβλημα στην εκτέλεση της εφαρμογής. Στο τέλος της διαδικασίας δημιουργεί ένα άθροισμα ελέγχου (checksum) και αποθηκεύει το αρχείο.

Εκτός από τον έλεγχο δομικής ορθότητας των αρχείων, το Dexopt, πραγματοποιεί κάποιες βελτιστοποιήσεις που σαν σκοπό έχουν την αύξηση των επιδόσεων της εφαρμογής μέσω της μείωσης του μεγέθους και της πολυπλοκότητας του κώδικα της. Η διαδικασία βελτιστοποίησης είναι άρρηκτα

συνδεδεμένη με την εκάστοτε έκδοση του διερμηνευτή και για αυτό δεν μπορούν να πραγματοποιηθούν αλλού παρά μόνο στην συγκεκριμένη συσκευή που εγκαθιστάτε η εφαρμογή. Για την μείωση του μεγέθους του κώδικα τυχόν άδειες μέθοδοι αντικαθίστανται από εντολές μη ενέργειας (Nop Instructions) και συχνά χρησιμοποιούμενες μέθοδοι γίνονται Inline κερδίζοντας έτσι τον χρόνο που απαιτείτε για την διαδικασία κλήσης μιας μεθόδου. Επιπλέον, για να πέτυχει καλύτερη εκμετάλλευση της βοηθητικής μνήμης του επεξεργαστή (Cache Memory), τύποι δεδομένων που έχουν μέγεθος μικρότερο των 32bit συγχωνεύονται σε λέξεις των 32bit.

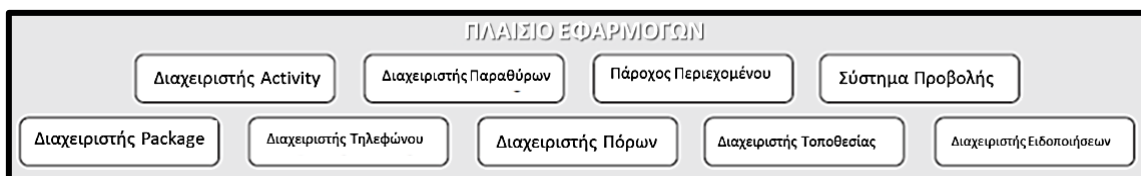
Επιπλέον των βελτιστοποιήσεων και ελέγχων που γίνονται στο αρχείο .dex το Android Runtime περιβάλλον προσπαθεί να χειρίζεται την μνήμη με όσο το δυνατόν καλύτερο τρόπο. Στον Dalvik υπάρχουν τμήματα μνήμης που είναι κοινά και άλλα που ανήκουν μονάχα στην εκάστοτε εφαρμογή. Για την διαχείριση των κοινών τμημάτων μνήμης το Android χρησιμοποιεί το Zygote. Το Zygote είναι μια διεργασία που ξεκινά κατά την εκκίνηση του λειτουργικού συστήματος και είναι γονική διεργασία όλων των στιγμιότυπων του διερμηνευτή που ξεκινούν για εκτέλεση εφαρμογών. Το Zygote φορτώνει και αρχικοποιεί κλάσεις που χρησιμοποιούνται πολύ συχνά από τις εφαρμογές. Για παράδειγμα, τα .dex αρχεία των βιβλιοθηκών του συστήματος που είναι διαθέσιμες στις εφαρμογές.

Ένας άλλος μηχανισμός του Android για σωστή διαχείριση μνήμης είναι ο συλλέκτης σκουπιδιών (Garbage Collector). Κάθε στιγμιότυπο του διερμηνευτή έχει τον δικό του συλλέκτη και άρα η περιοχή μνήμης που έχει στην διάθεση του καθορίζεται ανεξάρτητα από τους άλλους διερμηνευτές. Κύριος ρόλος του συλλέκτη είναι να ελευθερώνει περιοχές μνήμης που καταλαμβάνονται από δεδομένα που δεν θα χρειαστούν ξανά και την πορεία εκτέλεσης της εφαρμογής.

Τέλος, ένα σημαντικό χαρακτηριστικό του Android Runtime είναι η ύπαρξη του JIT (Just in Time) μεταφραστή. Ο JIT έχει σαν σκοπό την μετάφραση από bytecodes σε κώδικα μηχανής προκειμένου να αυξηθεί η ταχύτητα εκτέλεσης των αντίστοιχων τμημάτων της εφαρμογής. Υπάρχουν δυο ειδών μεταφραστές αυτής της κατηγορίας, οι βασισμένοι σε μεθόδους (Method Based) και οι βασισμένοι σε ακολουθίες εκτέλεσης (Trace Based). Οι πρώτοι μεταφράζουν ολόκληρες μεθόδους από τον εκτελούμενο κώδικα σε γλώσσα μηχανής πριν την εκτέλεση τους ενώ οι δεύτεροι μεταφράζουν ένα σύνολο εντολών που ανήκουν σε ένα κοινό δρόμο εκτέλεσης. Λόγω περιορισμών μνήμης στο Android ο JIT μεταφραστής που χρησιμοποιήθηκε είναι βασισμένος σε ακολουθία εκτέλεση και έχει περιορισμό σε ένα βάθος εκατό εντολών. Ο μεταφραστής και ο διερμηνευτής λειτουργούν αρμονικά σε ένα αυστηρά καθορισμένο περιβάλλον συνεργασίας. Σε περίπτωση εμφάνισης κάποιου σφάλματος στον κώδικα που κατασκεύασε ο μεταφραστής, ο διερμηνευτής αναλαμβάνει να επαναφέρει την εκτέλεση στην κατάσταση που ήταν πριν αναμιχθεί ο μεταφραστής και να εκτελέσει τον κώδικα ξανά χωρίς την βοήθεια του τελευταίου. Ο μηχανισμός αυτός επεκτείνεται και δίνει την δυνατότητα παράλληλης εκτέλεσης (κώδικα JIT και διερμηνευμένου κώδικα) με σκοπό την επιβεβαίωση της ορθής λειτουργίας του παραγομένου από τον μεταφραστή κώδικα. Οι πρώτες εκδόσεις του JIT μεταφραστή παράγαν κώδικα μόνο για την αρχιτεκτονική ARM, καθώς η εξέλιξη του Android προχωράει θα δούμε μεταφραστές αυτού του είδους που θα παράγουν κώδικα για μια μεγαλύτερη γκάμα διαθέσιμων επεξεργαστών.

2.4.4 Το επίπεδο του Application Framework

Το Application Framework (Εικόνα 9) είναι το επίπεδο της στοίβας του Android που ενδιαφέρει περισσότερο τον προγραμματιστή εφαρμογών. Το επίπεδο αυτό παρέχει μια πληθώρα από API τα οποία δίνουν την δυνατότητα στον προγραμματιστή να ενσωματώσει στις εφαρμογές του δυνατότητες όπως συνδεσιμότητα στο διαδίκτυο, δημιουργία διεπαφών χρήστη, πρόσβαση στα αποθηκευτικά μέσα της συσκευής και άλλα. Όλες οι βιβλιοθήκες του επιπέδου αυτού είναι γραμμένες στην γλώσσα προγραμματισμού Java και αξιοποιούν τις δυνατότητες που προσφέρει το Android Runtime και οι βασικές βιβλιοθήκες του λειτουργικού συστήματος. Ο τρόπος που είναι οργανωμένα τα API στο επίπεδο αυτό ακολουθεί την λογική του διαχειριστή (Manager). Για κάθε υπηρεσία που προσφέρει το επίπεδο αυτό υπάρχουν διαχειριστές που οι εφαρμογές μπορούν να καλέσουν για να τους παρασχεθεί η αντίστοιχη υπηρεσία. Οι εφαρμογές πρέπει να χρησιμοποιούν αυτό το σχήμα επικοινωνίας με τις βιβλιοθήκες του συστήματος και να μην χρησιμοποιούν απευθείας τις βασικές βιβλιοθήκες. Έτσι το Android έχει την δυνατότητα να επιβάλει περιορισμούς στο ποιες λειτουργίες μπορεί κάθε εφαρμογή να εκτελεί.



Εικόνα 10: Το επίπεδο Application Framework

(<http://docplayer.gr/docs-images/25/6629751/images/35-0.png>)

Από τον μεγάλο αριθμό API που παρέχει το Android στο επίπεδο αυτό, για την ανάπτυξη εφαρμογών, αναφερόμαστε συνοπτικά στα βασικότερα.

Η κλάση View στο Android είναι η βασική κλάση τα στιγμιότυπα της οποίας συνθέτουν μια διεπαφή χρήστη. Όλα τα ορατά στην οθόνη στοιχεία μιας διεπαφής αποτελούν στιγμιότυπα της κλάσης αυτής. Επιπλέον των ορατών στοιχείων, το Android, προσφέρει και κάποια μη ορατά στοιχεία, που ανήκουν στην κλάση ViewGroup, τα οποία έχουν σαν σκοπό, την χωρική οργάνωση των στοιχείων της διεπαφής στην οθόνη της συσκευής. Για την κάλυψη της ανάγκης αλληλεπίδρασης του χρήστη με την διεπαφή, η κλάση View, παρέχει ένα σύνολο μεθόδων που εκτελούνται όταν ανιχνευθεί κάποιο συμβάν από τον χρήστη (πάτημα κουμπιού, παρατεταμένο πάτημα κουμπιού κ.λπ.). Οι μέθοδοι αυτοί, φυσικά, μπορούν να επικαλυφθούν από τον προγραμματιστή προκειμένου η εφαρμογή να ανταποκρίνεται με τον επιθυμητό τρόπο στα διάφορα συμβάντα της διεπαφής.

Παράλληλα με τον έλεγχο της απόκρισης της εφαρμογής στα διάφορα συμβάντα, ο προγραμματιστής, έχει στα χεριά του ένα σύνολο κλάσεων (υποκλάσεις της Dialog) οι οποίες του επιτρέπουν να εμφανίζει στον χρήστη παράθυρα με σύντομα μηνύματα και να λαμβάνει κάποια απόκριση σε αυτά. Για την ενημέρωση του χρήστη, εκτός από τα παράθυρα διαλόγου, υπάρχουν και λιγότερο παρεμβατικοί τρόποι ενημερώσεων. Αυτά είναι τα λεγόμενα Toast μηνύματα και οι ενημερώσεις στην

μπάρα κατάστασης. Τα Toast μηνύματα είναι μικρά μηνύματα που εμφανίζονται στο κάτω μέρος της οθόνης σε ένα μικρό πλαίσιο και εξαφανίζονται μετά από κάποια δευτερόλεπτα. Σε αντίθεση με ένα πλαίσιο διαλόγου τα Toast μηνύματα δεν επικαλύπτουν την τρέχουσα εφαρμογή στο σύνολο της. Τα μηνύματα στην μπάρα κατάστασης είναι τα λιγότερο επεμβατικά μιας και δεν επηρεάζουν την εμφάνιση της τρέχουσας εφαρμογής στο ελάχιστο. Το μόνο που ένα μήνυμα της μπάρας κατάστασης κάνει είναι να εμφανίσει ένα εικονίδιο σε αυτήν και ένα σύντομο μήνυμα το οποίο ο χρήστης μπορεί να διαβάσει και να διαγράψει οποτεδήποτε το επιθυμεί.

Εκτός από τις δυνατότητες που αφορούν την διεπαφή χρήστη υπάρχουν και API που παρέχουν την δυνατότητα στις εφαρμογές να αναπαράγουν πολυμέσο περιεχόμενο. Οι κλάσεις που παρέχουν τις δυνατότητες αυτές αξιοποιούν το Media Framework που αναφέραμε προηγουμένα στο επίπεδο με τις βασικές βιβλιοθήκες του Android. Χαρακτηριστικά παραδείγματα τέτοιων κλάσεων είναι οι MediaPlayer και οι MediaEncoder που, όπως δηλώνουν τα ονόματά τους, χρησιμοποιούνται για την αναπαραγωγή ή καταγραφή πολυμέσων των υποστηριζόμενων τύπων.

Δεδομένης της σημασίας που έχει η ικανότητα δικτύωσης των κινητών συσκευών, δεν θα μπορούσαν να λείπουν από το επίπεδο αυτό οι κλάσεις που προσφέρουν τέτοιου είδους δυνατότητες στις εφαρμογές. Χαρακτηριστική κλάση είναι η ConnectivityManager που παρέχει πληροφορίες για τις δυνατές συνδέσεις μιας συσκευής καθώς και για την κατάσταση της κάθε σύνδεσης. Το Android παρέχει ένα σύνολο πακέτων για την διαχείριση συνδέσεων και μεταφοράς των δεδομένων σε διάφορα δίκτυα. Παρακάτω βλέπουμε μερικά από αυτά:

- **Java.net:** Οι βιβλιοθήκες της Java για δημιουργία συνδέσεων με sockets η χρησιμοποιώντας το πρωτόκολλο HTTP
- **Android.net:** βιβλιοθήκες του Android που επεκτείνουν τις δυνατότητες των πακέτων της java.net
- **Android.net.http:** Βιβλιοθήκες του Android για την διαχείριση ασφαλών συνδέσεων με πρωτοκόλλα HTTPS
- **Org.apache:** Πακέτα του Apache Foundation για HTTP συνδέσεις
- **Android.telephony:** Πακέτα του Android για πρόσβαση σε δίκτυα τηλεφωνίας, όπως GSM και CDMA
- **Android.net.wifi:** Πακέτα του Android για σύνδεση σε δίκτυα με χρήση Wi-Fi

Τέλος, δίνεται η δυνατότητα στις εφαρμογές να χρησιμοποιούν (σε όσες συσκευές είναι διαθέσιμο) το Bluetooth για να συνδέονται με άλλες συσκευές και να ανταλλάσσουν δεδομένα. Σε αυτή την κατεύθυνση βοηθάει το Bluetooth API του Android που επιτρέπει την αναζήτηση άλλων συσκευών, την σύνδεση με αυτές καθώς και την ανταλλαγή δεδομένων.

Όπως και οι εφαρμογές στους προσωπικούς μας υπολογιστές, έτσι και οι εφαρμογές στο Android χρειάζονται πολλές φορές να αποθηκεύουν πληροφορίες σε μια πιο μόνιμη μορφή. Στο Android οι εφαρμογές έχουν τις εξής επιλογές:

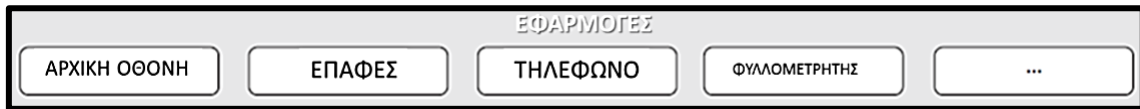
- **SharedPreferences:** Η κλάση αυτή παρέχει την δυνατότητα αποθήκευσης ζευγών ονόματος-τιμής όπου οι τιμές είναι κάποιου βασικού τύπου (ακέραιοι αριθμοί, αλφαριθμητικά κ.λπ.). Τα δεδομένα αυτά είναι διαθέσιμα σε άλλες εφαρμογές για ανάγνωση ή εγγραφή.
- **Internal/External Storage:** Αν η εφαρμογή επιθυμεί να μην έχει κανένας άλλος πρόσβαση στα δεδομένα της μπορεί να τα αποθηκεύσει στον εσωτερικό αποθηκευτικό χώρο. Δεδομένα που είναι πιθανόν να διαμοιραστούν μεταξύ εφαρμογών μπορούν να αποθηκευτούν στο εξωτερικό αποθηκευτικό χώρο ο οποίος συνήθως είναι οι κάρτες αποθήκευσης που φέρουν οι συσκευές.
- **Database Storage:** Όπως αναφέραμε, το Android παρέχει τις απαραίτητες υποδομές για την δημιουργία και διαχείριση βάσεων δεδομένων. Αν λοιπόν, μια εφαρμογή επιθυμεί να έχει μια οργανωμένη βάση δεδομένων μπορεί να αξιοποιήσει αυτές τις δυνατότητες για την αποθήκευση των δεδομένων της.
- **Network Storage:** Αν καμία από τις παραπάνω μεθόδους δεν καλύπτει τις ανάγκες μιας εφαρμογής τότε μπορεί να αποθηκεύσει τα δεδομένα της σε κάποιον απομακρυσμένο εξυπηρετητή χρησιμοποιώντας τις δυνατότητες δικτύωσης που παρέχει το Android.

Όσον αφορά στο θέμα της αποθήκευσης, τέλος, αξίζει να αναφερθεί ότι από την έκδοση του Android 2.2 και έπειτα παρέχεται ένας μηχανισμός στις εφαρμογές να αποθηκεύουν τις ρυθμίσεις τους και άλλα δεδομένα σε κάποιο απομακρυσμένο εξυπηρετητή. Έτσι, σε μια περίπτωση που γίνει επανεγκατάσταση της εφαρμογής η αυτή εγκατασταθεί σε άλλη συσκευή του ιδίου ατόμου, τα δεδομένα μπορούν να ανακτηθούν και η εφαρμογή να έλθει σε μια κατάσταση ίδια με αυτή που ήταν πριν την επανεγκατάσταση, εξασφαλίζοντας έτσι καλύτερη εμπειρία χρήση για τον χρήστη.

Επιπρόσθετα, το Android παρέχει και API τα οποία δίνουν ένα σύνολο χρήσιμων δυνατοτήτων και βοηθούν τις εφαρμογές να γίνουν περισσότερο ελκυστικές και ανταγωνιστικές. Τέτοια είναι το Location & Maps API, που δίνει την δυνατότητα στις εφαρμογές να κάνουν το περιεχόμενο τους διαθέσιμο προς αναζήτηση από τους χρήστες τους, Το Web Kit API, με το οποίο οι εφαρμογές μπορούν να προσφέρουν υπηρεσίες περιηγητών, το Speech API (Cloud to Device Messaging API) που δίνει την δυνατότητα αποστολής μηνυμάτων στην συσκευή από κάποια απομακρυσμένη τοποθεσία.

2.4.5 Το επίπεδο των εφαρμογών

Τελειώνοντας την περιγραφή της στοίβας του Android φτάνουμε στο ανώτερο επίπεδο όπου βρίσκονται οι εφαρμογές του Android (Εικόνα 10). Οι εφαρμογές του Android είναι ο σκοπός της παρούσας εργασίας και οι απαραίτητες δομές για την ανάπτυξη εφαρμογών Android θα παρουσιαστούν αναλυτικότερα στην συνέχεια. Στο παρόν σημείο αρκεί να αναφέρουμε ότι όλες οι δυνατότητες που έχουμε αναφέρει μέχρι τώρα είναι στην διάθεση των εφαρμογών. Όλα τα API που διαθέτει το Application Framework είναι στα χεράκια του προγραμματιστή για να συνδυάσει και να δημιουργήσει εφαρμογές για το περιβάλλον Android.



Εικόνα 11: Το επίπεδο των εφαρμογών Android
(<http://docplayer.gr/docs-images/25/6629751/images/35-0.png>)

Στο επίπεδο αυτό βρίσκονται και οι προεγκατεστημένες εφαρμογές του Android, όπως η εφαρμογή για την αποστολή SMS, η εφαρμογή για την πραγματοποίηση κλήσεων και άλλες.

ΚΕΦΑΛΑΙΟ 3^ο

3.1 ΟΙ ΕΚΔΟΣΕΙΣ ΤΟΥ ANDROID

Το Android έχει δημιουργηθεί πάνω στο λειτουργικό σύστημα Linux αλλά είναι προσαρμοσμένο πάνω σε κινητές συσκευές. Ξεκινώντας από το 2008 με την κυκλοφορία του πρώτου Android Smartphone μέχρι σήμερα έχουν βγει αρκετές εκδόσεις όπως είναι φυσικό και υπάρχει το χαρακτηριστικό γνώρισμα ότι όλες έχουν πάντα ονόματα κάποιου γλυκού είτε επιδορπίου. Η ονομασία των εκδόσεων αυτών πχ Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, κλπ.



Εικόνα 12: Όλες οι εκδόσεις του Android
(<http://www.compareraja.in/blog/android-nougat-features/>)

Από τη μεριά του ο χρήστης μπορεί να κάνει ενημέρωση, όποτε είναι εφικτό, και να αξιοποιήσει τις νέες λειτουργίες και δυνατότητες που του προσφέρονται. Βέβαια, δεν μπορούν όλες οι συσκευές να ενημερώνονται με τις τελευταίες εκδόσεις καθώς υπάρχουν περιορισμοί ανάλογα με τις δυνατότητες της εκάστοτε συσκευής. Ας δούμε τις κυριότερες εκδόσεις Android που κυκλοφόρησαν από το 2008 μέχρι σήμερα με χρονολογική σειρά.

3.1.1 Έκδοση Android 1.0

Η πρώτη εμπορική έκδοση του λογισμικού κυκλοφόρησε στις 23 Σεπτεμβρίου 2008 και διατέθηκε με την πρώτη συσκευή Android το HTC Dream έχοντας τα ακόλουθα χαρακτηριστικά:

- Την εφαρμογή Android Market, όπου ο χρήστης μπορεί να εγκαταστήσει στο κινητό του παιχνίδια και εφαρμογές ή να ενημερώσει τις ήδη υπάρχουσες.
- Πρόγραμμα περιήγησης, όπου επιτυγχάνεται πρόσβαση σε ιστοσελίδες.
- Υποστήριξη κάμερας.
- Πρόσβαση σε E-mail Servers, υποστηρίζοντας πρωτοκολλά POP3,IMP4,SMTP.
- Υποστήριξη ασυρμάτου δικτύου (Wi-Fi) και Bluetooth.
- Συγχρονισμός με την εφαρμογή Gmail, τις επαφές, το ημερολόγιο.
- YouTube Video Player και φωνητική κλήση επιτρέποντας την κλήση χωρίς την πληκτρολόγηση ονόματος ή αριθμού.
- Λοιπές εφαρμογές: Ειδοποίηση, Αριθμομηχανή, Dialer, Αρχική Οθόνη, Εικόνες και Ρυθμίσεις.

3.1.2 Έκδοση Android 1.1

Η έκδοση Android 1.1 κυκλοφόρησε στις 9 Φεβρουάριου 2009. Έλυσε αρκετά σφάλματα της πρώτης έκδοσης και πρόσθεσε νέα χαρακτηριστικά:

- Λεπτομερή σχόλια για επιχειρήσεις στους χάρτες.
- Δυνατότητα αποθήκευση συνημμένων αρχείων.
- Υποστήριξη για μαρκίζα σε διατάξεις συστήματος.

3.1.3 Έκδοση Android 1.5 (Cupcake)



Εικόνα 13: Το λογότυπο του Android 1.5 (Cupcake)

(<http://www.androidust.com/what-are-different-android-versions-released/>)

Στις 27 Απριλίου 2009 ανακοινώθηκε η έκδοση 1.5, η οποία ήταν η πρώτη που πήρε κωδική ονομασία εδέσματος και βασίστηκε στον Linux Kernel 2.6.27:

- Αυτόματη υποστήριξη προτύπου Bluetooth A2DP και AVRCP.
- Ικανότητα αυτόματης σύνδεσης σε μικροσυσκευές Bluetooth από συγκεκριμένη απόσταση.
- Προστέθηκε αυτόματη περιστροφή κατά επιλογή.
- Προστέθηκε Animation κατά την εκκίνηση.
- Δυνατότητα ανεβάσματος βίντεο στο YouTube.
- Δυνατότητα ανεβάσματος φωτογραφιών στο Picasa.
- Νέο γραφικό περιβάλλον με κινούμενες μεταβάσεις οθόνης.
- Υποστήριξη εικόνων έξυπνων πληκτρολογίων με πρόβλεψη και αυτόματο ορθογραφικό έλεγχο κειμένου.
- Προστέθηκαν λειτουργίες αντιγραφής και επικόλλησης στο πρόγραμμα περιήγησης.
- Υποστήριξη εγγραφής και αναπαραγωγής βίντεο 3GP και MPEG4 μορφές.
- Δυνατότητα αγαπημένων με εικόνα στις επαφές.

3.1.4 Έκδοση Android 1.6 (Donut)



Εικόνα 14: Το λογότυπο του Android 1.4 (Donut)

(<http://www.androidust.com/what-are-different-android-versions-released/>)

Τον Σεπτέμβρη του 2009 δημοσιεύτηκε το Android 1.6 βασισμένο στον Linux Kernel 2.6.29 και συμπεριλάμβανε πολλές νέες λειτουργίες:

- Υποστήριξη για WVGA ανάλυση οθόνης.
- Υποστήριξη πολλαπλών αρχείων ταυτόχρονα.
- Ανανεωμένη φωνητική αναζήτηση με ταχύτερη απόκριση.
- Βελτιωμένο Android Market με την ικανότητα ο χρήστης να βλέπει στιγμιότυπα εφαρμογών.
- Μηχανή μετατροπής κειμένου σε ομιλία (text-to-speech).

3.1.5 Έκδοση Android 2.0/2.1 (Eclair)



Εικόνα 15: Το λογότυπο του Android 2.0/2.1(Eclair)

(<http://www.androidust.com/what-are-different-android-versions-released/>)

Και αυτή η έκδοση βασισμένη στον Linux kernel 2.6.29, παρουσιάστηκε στο κοινό στις 26 Οκτωβρίου του 2009 και 3 μήνες αργότερα, τον Ιανουάριο του 2010, η έκδοση 2.1. Οι αλλαγές στα χαρακτηριστικά παρουσιάζονται παρακάτω. Αξίζει να σημειωθεί ότι η 2.1 δεν είχε νέα χαρακτηριστικά πάρα διορθώσεις σε σφάλματα.

- ✚ Υποστήριξη Bluetooth 2.1.
- ✚ Προσθήκη ζωντανών ταπετσαριών (Live Wallpapers).
- ✚ Βελτιωμένη η εφαρμογή Google Maps 3.1.2
- ✚ Υποστηρίζει περισσότερες αναλύσεις οθονών με ποιοτικότερη αναλογία αντίθεσης.
- ✚ Βελτιωμένη Motion Event κλάση ώστε να πραγματοποιούνται Multi-Touch γεγονότα.
- ✚ Ενσωματώθηκε φλας για την κάμερα, η οποία πλέον υποστηρίζει ψηφιακό Zoom.
- ✚ Δυνατότητα αναζήτησης αποθηκευμένων SMS και MMS και αυτόματη διαγραφή παλαιότερων μηνυμάτων συνομιλίας όταν ένα καθορισμένο από το χρήστη όριο επιτευχθεί.
- ✚ Δυνατότητα πρόσθεσης πολλών λογαριασμών E-mail σε μια συσκευή και συγχρονισμό επαφών.

3.1.6 Έκδοση Android 2.2 (Froyo)



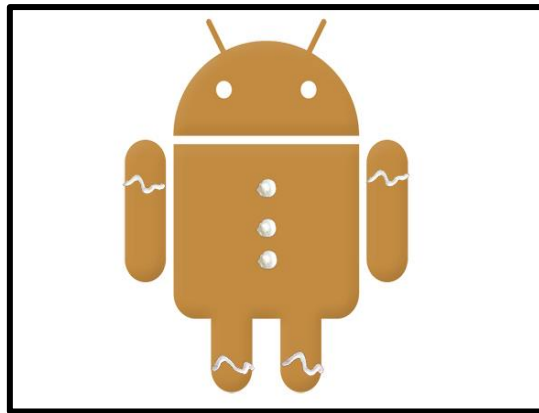
Εικόνα 16: Το λογότυπο του Android 2.2 (Froyo)

(<http://www.androidust.com/what-are-different-android-versions-released/>)

Το Android 2.2 παρουσιάστηκε τον Μάιο του 2010 και πήρε την ονομασία του από τα αρχικά του εδέσματος Frozen Yogurt. Βασίστηκε στον Linux Kernel 2.6.32 και είχε τα εξής νέα χαρακτηριστικά:

- ✓ Ενημέρωση της Εφαρμογής Android Market.
- ✓ Καλύτερη ταχύτητα, μεγαλύτερη μνήμη και απόδοση.
- ✓ Υποστήριξη Adobe Flash.
- ✓ Γρήγορη εναλλαγή πληκτρολόγιων διαφορετικής γλώσσας και των λεξικών τους.
- ✓ Μεταφορά αρχείων μέσω Bluetooth και επιλογή απενεργοποίησης της πρόσβασης σε δεδομένα μέσω του δικτύου κινητής τηλεφωνίας.
- ✓ Υποστήριξη αριθμητικών και αλφαριθμητικών κωδίκων ασφάλειας.
- ✓ Συμβατότητα USB και λειτουργία δυναμικής ζώνης Wi-Fi (Hotspot).
- ✓ Ανανεωμένη εφαρμογή εκκίνησης με συντομεύσεις στο τηλέφωνο και στις εφαρμογές Browser.
- ✓ Δυνατότητα εγκατάστασης εφαρμογών στην κάρτα μνήμης.
- ✓ Υποστηρίζεται το Microsoft Exchange και έχει ενσωματωθεί ο μηχανισμός JavaScript του Chrome V8 στον Browser.

3.1.7 Έκδοση Android 2.3/2.3.2 (Gingerbread)



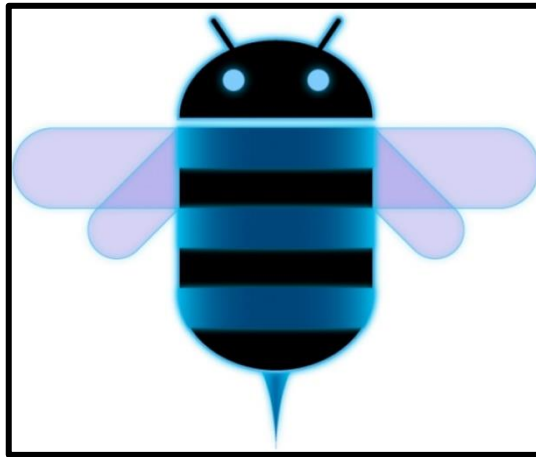
Εικόνα 17: Το λογότυπο του Android 2.3/2.3.2 (Gingerbread)
(<http://www.androidust.com/what-are-different-android-versions-released/>)

Κυκλοφόρησε τον Δεκέμβρη του 2010 και βασίστηκε στον Linux Kernel 2.6.35. Η έκδοση αυτή είχε μεγάλη απήχηση καθώς ήταν πιο ευχάριστη και πιο γρήγορη από τις προγενέστερες. Έφτασε ως την 2.3.7 εκδοχή της και οι προγραμματιστές εφαρμογών είχαν περισσότερες δυνατότητες από ποτέ.

- ❖ Το User Interface έγινε πιο απλό και γρήγορο.
- ❖ Υποστήριξη οθονών μεγάλων αναλύσεων και μεγεθών.
- ❖ Ενισχυμένη αντιγραφή/επικόλληση λειτουργικότητα.

- ❖ Νέα ηχητικά εφέ και Download Manager.
- ❖ Υποστηρίζεται η αναπαραγωγή βίντεο για WebM/VP8 και AAC κωδικοποίηση ήχου.
- ❖ Το Google Talk υποστηρίζει την συνομιλία με φωνή και βίντεο.
- ❖ Υποστηρίζονται εξαιρετικά μεγάλες αναλύσεις και μεγέθη οθονών.
- ❖ Υποστήριξη VoIP τηλεφωνίας μέσω Internet.
- ❖ Υποστηρίζεται το NFC (Near Field Communication)
- ❖ Εγγενής υποστήριξη σε περισσότερους αισθητήρες όπως βαρόμετρο και γυροσκόπιο.
- ❖ Βελτιώσεις σε ήχο και στις λειτουργίες απεικόνισης για την ανάπτυξη παιχνιδιών.

3.1.8 Έκδοση Android 3.0/3.1/3.2 (Honeycomb)



Εικόνα 18: Το λογότυπο του Android 3.0/3.1/3.2 (Honeycomb)
(<http://www.androidust.com/what-are-different-android-versions-released/>)

Τον Φεβρουάριο του 2011 παρουσιάστηκε το Android 3.0 και βασίστηκε στον Linux Kernel 2.6.36. Ήταν η πρώτη έκδοση για Tablets και οι αλλαγές που περιλαμβάνονται:

- ✓ Υποστηρίζονται διπύρηνιοι και τετραπύρηνιοι επεξεργαστές.
- ✓ Υποστήριξη Tablet με ολογραμμιακό User Interface.
- ✓ Υποστήριξη συνομιλίας βίντεο με την εφαρμογή Google Talk.
- ✓ Τα παράθυρα στον Browser αντικαταστάθηκαν από καρτέλες.
- ✓ Βελτιωμένη υποστήριξη Adobe Flash στον Browser.
- ✓ Δυνατότητα κρυπτογράφησης των δεδομένων του χρήστη.
- ✓ Το Multitasking βελτιώθηκε και ο χρήστης μπορεί να βλέπει τις εφαρμογές που τρέχουν από την μπάρα συστήματος.
- ✓ Δυνατότητα ανάγνωσης βιβλίων μέσω της εφαρμογής Google eBooks.
- ✓ Υποστήριξη HTTP Proxy για κάθε συνδεδεμένο Wi-fi.

3.1.9 Έκδοση Android 4.0/4.0.2 (Ice Cream Sandwich)

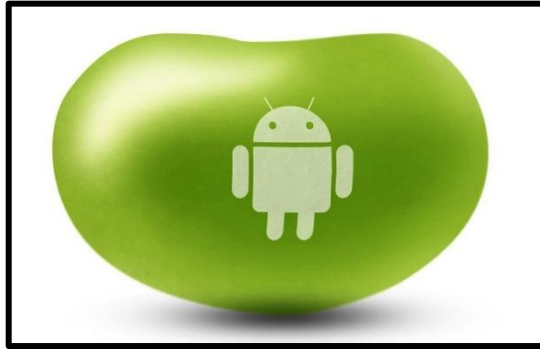


Εικόνα 19: Το λογότυπο του Android 4.0/4.0.2 (Ice Cream Sandwich)
(<http://www.androidust.com/what-are-different-android-versions-released/>)

Η έκδοση Android 4.0 (Ice Cream Sandwich) βασίστηκε στον Linux Kernel 3.0.1 και παρουσιάστηκε τον Οκτώβρη του 2011. Ο πηγαίος κώδικας δημοσιεύτηκε έναν μηνά αργότερα. Τα νέα του χαρακτηριστικά παρουσιάζονται παρακάτω:

- Το User Interface ανανεώθηκε πάλι και πλέον υπάρχουν εικονικά πλήκτρα που αντικατέστησαν τα αφής και τα φυσικά.
- Ευκολότερη δημιουργία φακέλων, με Drag and Drop.
- Υποστήριξη εγγραφής βίντεο σε 1080 pixel.
- Ανανεωμένη η εφαρμογή «επαφές», η οποία επικοινωνεί με εφαρμογές κοινωνικής δικτύωσης και υποστηρίζει εικόνες υψηλής ευκρίνειας.
- Δυνατότητα απενεργοποίησης εφαρμογών που λειτουργούν στο περιθώριο (Background).
- Οι συσκευές μπορούν να συνδεθούν μεταξύ τους ασύρματα χωρίς κάποιο Access Point, με την ύπαρξη Wi-Fi Direct.
- Λειτουργία Face Unlock, που ο χρήστης έχει την δυνατότητα να ξεκλειδώνει το κινητό του με αναγνώριση χαρακτηριστικών προσώπου.
- Η εφαρμογή Android Beam αξιοποιεί το NFC και επιτρέπει πλέον την αποστολή και λήψη δεδομένων από τη συσκευή σε άλλες εντός μικρής εμβέλειας.
- Ο Browser έχει την ικανότητα να ανοίγει μέχρι και 16 καρτέλες ταυτόχρονα.

3.1.10 Έκδοση Android 4.1/4.2/4.3 (Jelly Bean)



Εικόνα 20: Το λογότυπο του Android 4.1/4.2/4.3 (Jelly Bean)
(<http://www.androidust.com/what-are-different-android-versions-released/>)

Η Google ανακοίνωσε την έκδοση 4.1 στις 27 Ιονίου 2012. Βασίστηκε στον Linux Kernel 3.0.31 και ήταν μια σημαντική ενημέρωση που στόχευε στην βελτίωση της λειτουργικότητας και της εμφάνισης του User Interface. Το Tablet Nexus 7 ήταν η πρώτη συσκευή που έτρεξε αυτή την έκδοση και παρουσιάστηκε τον Ιούλιο του 2012. Οι νέες λειτουργίες παρουσιάζονται παρακάτω:

- ✓ Πλήρες ανανεωμένο το σύστημα ειδοποιήσεων.
- ✓ Βελτιωμένες λειτουργίες κάμερας και προηγμένη αναζήτηση με ομιλία.
- ✓ Βελτίωση των γραφικών μέσω της υποστήριξης OpenGL ES 3.0
- ✓ Βελτιωμένες δυνατότητες αισθητήρα.
- ✓ Προηγμένη ασφάλεια και βελτιώσεις στην απόδοση.
- ✓ Δυνατότητα χρήσης εξωτερικής συσκευής ήχου μέσω USB.
- ✓ Υποστηρίζονται τα Tablets μικρών οθονών με χρήση βελτιωμένης έκδοσης για κινητά τηλεφονά.

3.1.11 Έκδοση Android 4.4 (Kit Kat)



Εικόνα 21: Το λογότυπο του Android 4.4 (Kit Kat)
(<https://plus.google.com/+NeilMoore/posts/845BMXD3z6F>)

Το Android 4.4 ανακοινώθηκε τον Σεπτέμβρη του 2013 στην συσκευή της Google Nexus 5. Αναμενόταν για καιρό ως Android 5.0 με κωδική ονομασία Key Lime Pie. Αναπτύχθηκε για να τρέχει σε καλύτερο εύρος συσκευών από ότι οι προηγούμενες εκδόσεις, έχοντας 512 Mb ελάχιστη απαίτηση μνήμης RAM. Βασικά νέα χαρακτηριστικά της ήταν:

- Δυνατότητα εκτύπωσης μέσω ασυρμάτου δικτύου.
- Ανανεωμένος σχεδιασμός με λευκά αντί μπλε στοιχεία.
- Δυνατότητα επιλογής εισαγωγής μηνύματος μέσω της εφαρμογής Ρυθμίσεις.
- Υποστήριξη Bluetooth Message Access Profile (MAP).
- Απενεργοποιήθηκε η πρόσβαση στα στατιστικά της μπαταρίας από εφαρμογές που δεν έχουν εγκατασταθεί από το Android Market.
- Εικονικά κουμπιά του μενού είναι πάντα ορατά ακόμα και σε συσκευές που έχουν φυσικά κουμπιά.
- Έγιναν βελτιώσεις στην ασφάλεια και λυθήκαν σφάλματα προηγούμενων εκδόσεων.

3.1.12 Έκδοση Android 5.0/5.0.2/5.1/5.1.1 (Lollipop)



Εικόνα 22: Το λογότυπο του Android 5.0 (Lollipop)

(http://logos.wikia.com/wiki/Android/Version_History)

Το Android Lollipop 5.0/5.0.2 έκανε την εμφάνιση του στις 3 Νοέμβριου 2014, ενώ η αναβαθμισμένη έκδοση του δηλαδή η 5.1/5.1.1 έκανε την εμφάνιση του στις 9 Μάρτιου 2015 εξοπλισμένο με μια πληθώρα νέων χαρακτηριστικών το Lollipop φέρνει νέες δυνατότητες που θα αναφερθούμε παρακάτω:

- Τα νέα χαρακτηριστικά περιλαμβάνουν, υποστήριξη για πολλαπλές κάρτες SIM, Προστασία της συσκευής και υποστήριξη φωνής υψηλής ευκρίνειας (HD) σε συμβατά κινητά τηλέφωνα.

- Βασικά, το Android 5.1 Lollipop θα επιτρέπει πλέον σε smartphones να χρησιμοποιήσετε περισσότερες από μία κάρτα SIM όταν η συσκευή διαθέτει πολλαπλές υποδοχές SIM.
- Το δεύτερο νέο χαρακτηριστικό μείζονος σημασίας που έρχεται σε αυτήν την ενημέρωση είναι προστασία της συσκευής, η οποία παρέχει ιδιοκτήτες Smartphones με ένα επιπλέον επίπεδο ασφάλειας.
- Όταν κλαπεί ή χαθεί, κάποια συσκευή που έχει εγκατεστημένο το Android 5.1 Lollipop θα παραμείνει κλειδωμένη μέχρι ο ιδιοκτήτης της να εισέλθει με το λογαριασμό του στη Google. Αυτό ισχύει ακόμα και αν κάποιος επαναφέρει τη συσκευή στις εργοστασιακές ρυθμίσεις.
- Σύμφωνα με την Google, η δυνατότητα προστασίας της συσκευής θα είναι διαθέσιμη στα περισσότερα Android Smartphones και Tablets που θα κυκλοφορήσουν «φορώντας» το Android 5.1 Lollipop, εκτός από τα Nexus 6 και 9.
- Επιπλέον, το Android 5.1 Lollipop φέρνει το HD voice calling στις συμβατές συσκευές, όπως το Nexus 6 της T-Mobile και της Verizon.
- Τελευταίο, και επίσης σημαντικό, είναι η δυνατότητα εισόδου σε δίκτυα Wi-Fi και ο έλεγχος των συνδεδεμένων συσκευών με Bluetooth απευθείας από τις Γρήγορες ρυθμίσεις

3.1.13 Έκδοση Android 6.0/6.0.1 (Marshmallow)



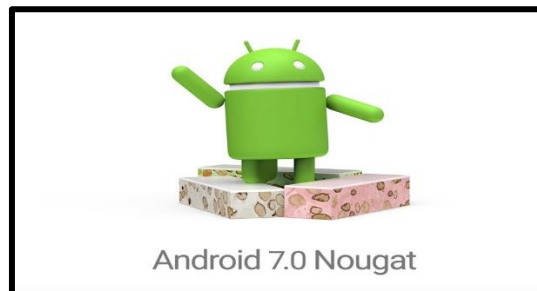
Εικόνα 23: Το λογότυπο του Android 6.0 (Marshmallow)

(<http://www.pcmag.com/slideshow/story/339347/9-sweet-hidden-features-inside-android-6-0-marshmallow>)

Το Android Marshmallow 6.0 έκανε την εμφάνιση του στις 5 Οκτώβριου 2015, Η Google επιβεβαίωσε πως το Android M, η επόμενη έκδοση του Android, θα ονομάζεται Marshmallow. Ένα χρόνο, λοιπόν, μετά την κυκλοφορία του Android 5.0 Lollipop, έρχεται το Android 6.0 Marshmallow για να διορθώσει τις αδυναμίες του προκατόχου του και να προσφέρει μερικά νέα χαρακτηριστικά. Το Lollipop ήταν μεγάλη αναβάθμιση, τόσο αισθητικά με το Material Design, όσο και λειτουργικά. Ομολογουμένως όμως, ενώ η Google υποσχόταν πολλά με το Lollipop, λίγα κατάφερε να υλοποιήσει. Το Project Volta αν θυμάστε υποσχόταν μεγάλη αναβάθμιση στη διάρκεια ζωής της μπαταρίας, η πλειοψηφία όμως των συσκευών είδαν χειρότερη διάρκεια μπαταρίας από αυτή που είχαν με το Android 4.4 KitKat. Αντίστοιχα, Bugs και άλλες αδυναμίες έκαναν τους χρήστες, ή έστω την πλειοψηφία αυτών να μην είναι ικανοποιημένοι με το Android 5.0. Καλό θα είναι η Google να μάθει από τα λάθη της και αυτή τη φορά να παραδώσει ένα ολοκληρωμένο λειτουργικό. Καλό θα είναι βέβαια να μάθουμε και εμείς από τα λάθη μας και να κρατάμε μικρό καλάθι στις όποιες ανακοινώσεις, μέχρι να τις δούμε υλοποιημένες. Οι όποιες πληροφορίες έχουμε σήμερα για τα νέα χαρακτηριστικά του Android 6.0 Marshmallow, προέρχονται από ένα changelog του API level 23. Οι δύο μεγαλύτερες αλλαγές που έρχονται είναι τα App Permission Controls και η υποστήριξη δακτυλικών αποτυπωμάτων. Η σύντομη περιγραφή της Google έχει ως εξής:

- Visual Voicemail.
- Rotating Home screen
- Τέλος στο Dark theme
- Διορθώθηκε το App drawer
- Δυνατότητα διαγραφής screenshot
- RAM Manager
- Επιλογή Network Settings Reset
- Ανανέωση του αποθηκευτικού χώρου
- App permissions
- Υποστήριξη δακτυλικών αποτυπωμάτων
- Mobile πληρωμές
- App links
- Φόρτιση και μπαταρία
- Αυτόματο Backup και Restore εφαρμογών
- Προσαρμόσιμες συσκευές αποθηκευτικού χώρου
- Google υπηρεσίες

3.1.14 Έκδοση Android 7.0 (Nougat)



Εικόνα 24: Το λογότυπο του Android 7.0 (Nougat)

(<http://www.androidauthority.com/android-7-0-nougat-confirmed-701079/>)

Η τελευταία έκδοση του Android είναι διαθέσιμη από τις 22 Αυγούστου 2016 με βελτιωμένα τα χαρακτηριστικά:

- ✚ Το Android 7.0 κάνει τις ενημερώσεις γρηγορότερες
- ✚ Multi-Window Multitasking
- ✚ Βελτιωμένο σύστημα ειδοποιήσεων
- ✚ Μεγαλύτερη διάρκεια μπαταρίας & οικονομία δεδομένων
- ✚ Βελτιώσεις στην ασφάλεια

ΚΕΦΑΛΑΙΟ 4°

4.1 ΑΠΑΙΤΗΣΕΙΣ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ

Η ανάπτυξη των εφαρμογών στο περιβάλλον του Android είναι μια συνθέτη εργασία, αρκετά απαιτητική, καθώς ο προγραμματιστής θα πρέπει να έχει κατά νου κάποιους άτυπους κανόνες για την ομαλή υλοποίηση της εφαρμογής. Ο σχεδιασμός και οι λειτουργίες της είναι το σημαντικότερο κομμάτι καθώς πρέπει να ισορροπούν ώστε να επιτευχθεί το επιθυμητό αποτέλεσμα. Αν μια εφαρμογή έχει τρομερές λειτουργίες αλλά κακό σχεδιασμό ο χρήστης θα μπερδευτεί και πιθανότατα να μην την χρησιμοποιήσει ξανά. Οι απαιτήσεις που προκύπτουν αφορούν την υποστήριξη πολλών συσκευών, διαφορετικών διαστάσεων οθόνης και παλαιότερων εκδόσεων λειτουργικού. Παρακάτω θα αναλυθούν περαιτέρω οι απαιτήσεις.

4.1.1 Υποστήριξη πολλών συσκευών

Οι συσκευές που χρησιμοποιούν το Android κυκλοφορούν σε πολλά σχήματα και μεγέθη και έχουν μεγάλη ποικιλία στον τύπο τους. Για να γίνει μια πετυχημένη εφαρμογή θα πρέπει να υποστηρίζει την πλειονοψηφία των συσκευών και να εναρμονίζεται με τις προδιαγραφές τους. Ο προγραμματιστής λοιπόν για να κάνει προσβάσιμη την εφαρμογή του σε όσο το δυνατόν περισσότερες συσκευές, θα πρέπει να την σχεδιάσει με τέτοιο τρόπο ώστε να ανταποκρίνεται στις πολλές εκδόσεις που υπάρχουν και στην μεγάλη ποικιλία υλικού (Hardware) που κυκλοφορεί. Αυτό συνεπάγεται με συνεχομένη ενημέρωση της εφαρμογής, προσαρμογή

της σε νέες τεχνολογίες που προκύπτουν και χρήση των δυνατοτήτων που θα αποφέρουν οι ενημερώσεις του λειτουργικού.

4.1.2 Υποστήριξη διαφορετικών διαστάσεων οθόνης

Όπως αναφέραμε και πιο πριν, το γραφικό περιβάλλον της εφαρμογής είναι μείζονος σημασίας για την αποδοχή της από τους χρήστες. Το πρόβλημα που καλείτε να λύσει ο προγραμματιστής είναι η υποστήριξη όλων των διαστάσεων και πυκνοτήτων Pixel που διαθέτουν οι συσκευές. Το Android κατηγοριοποιεί τις οθόνες σε δυο γενικές κατηγορίες διαστάσεων και πυκνότητας. Η πρώτη χωρίζεται σε τέσσερα μεγέθη small, normal, large και xlarge, ενώ η δεύτερη χωρίζεται σε ldpi, mdpi, hdpi, xhdpi. Αυτό διευκολύνει τους προγραμματιστές να βελτιώσουν την εμφάνιση των εφαρμογών τους δημιουργώντας διαφορετικούς σχεδιασμούς για κάθε κατηγορία.

4.1.3 Υποστήριξη παλιότερων εκδόσεων

Η συνεχόμενη εξέλιξη της πλατφόρμας εκτός από πλεονέκτημα κρύβει και προκλήσεις που θα πρέπει να αντιμετωπίσει ο προγραμματιστής. Θα πρέπει να ακολουθεί τις εξελίξεις και να αξιοποιεί τις νέες λειτουργίες των νέων εκδόσεων, χωρίς όμως να χάνεται η υποστήριξη του στις παλιότερες. Αυτό το πρόβλημα είναι από τα σημαντικότερα καθώς οι βιβλιοθήκες συμβατότητας που εκδίδει η Google μαζί με τις νέες εκδόσεις δεν αρκούν και έτσι υπάρχουν χαρακτηριστικά που δεν υποστηρίζονται στα παλαιότερα APIs. Η καλύτερη πρακτική που μπορεί να ακολουθηθεί είναι η εκάστοτε εφαρμογή να υποστηρίζει το 90% των ενεργών συσκευών σχετικά με την έκδοση που τρέχουν. Η Google βοηθάει σε αυτό, παρουσιάζοντας στατιστικά στοιχεία κάθε δυο εβδομάδες στοχεύοντας να ελαττώσει το φαινόμενο της διάσπασης (Fragmentation).

4.2 ΒΗΜΑΤΑ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΗΣ

Για την ανάπτυξη των εφαρμογών πρέπει να ακολουθηθεί συγκεκριμένη διαδικασία ώστε το τελικό αποτέλεσμα να συνάδει με την φιλοσοφία του Android. Ο προγραμματιστής πρέπει να χρησιμοποιήσει εργαλεία που θα τον βοηθήσουν στον σχεδιασμό της εφαρμογής, στη μεταγλώττιση του πηγαίου κώδικα έως την δημοσίευσή της. Η διαδικασία αυτή χωρίζεται σε τέσσερα βήματα και αναλύεται εκτενώς παρακάτω:

4.2.1 Εγκατάσταση Λογισμικού

Το πρώτο βήμα που απαιτείται είναι το στήσιμο του περιβάλλοντος εργασίας (IDE) του προγραμματιστή ώστε να μπορεί να σχεδιάσει, να αναπτύσσει και να ελέγξει την λειτουργία των εφαρμογών του. Η επιλογή του κατάλληλου IDE, είναι προσωπική για κάθε προγραμματιστή καθώς πρέπει να επιλέξει ανάλογα με τις προτιμήσεις και το σύστημα που διαθέτει. Τα εργαλεία που φέρει το Android SDK πρέπει να χρησιμοποιηθούν χωρίς εξαιρέσεις.

Για τον έλεγχο της εφαρμογής σε πραγματικές συνθήκες και την δοκιμή για τυχόν υπολειτουργίες, η εφαρμογή θα πρέπει να τρέξει σε αρκετές φυσικές συσκευές ώστε να ληφθούν συμπεράσματα σχετικά με την συμπεριφορά της σε διαφορετικές περιπτώσεις. Αυτή η πρακτική όμως είναι χρονοβόρα και πολυδάπανη και ευτυχώς χάριν στην διαχείριση εικονικών συσκευών (AVD) μπορεί να δημιουργηθεί ένας αριθμός από εικονικές συσκευές που τεστάρει ανά πάσα στιγμή την πρόοδο της εφαρμογής.

4.2.2 Ανάπτυξη

Πρόκειται για το πιο σημαντικό στάδιο της διαδικασίας το οποίο απαιτεί χρόνο και λήψεις αποφάσεων. Συγκεκριμένα, ο προγραμματιστής θα πρέπει να αποφασίσει για την λειτουργικότητα, τις δυνατότητες, το περιεχόμενο και το σχεδιασμό της εφαρμογής. Θα πρέπει να ερευνηθεί η εξέλιξη που θα έχει και να αποφευχθούν υπερβολές στο σχεδιασμό.

Η διαδικασία ξεκινάει με την δημιουργία νέου Project το οποίο θα περιέχει τον πηγαίο κώδικα, τις εικόνες, τα κείμενα και γενικά όλους τους πόρους που χρειάζονται για να υλοποιηθεί η εφαρμογή. Ο προγραμματιστής οφείλει να έχει το υλικό του τακτοποιημένο ώστε να ακολουθήσει το επόμενο στάδιο της αποσφαλμάτωσης (Debugging).

4.2.3 Αποσφαλμάτωση και δοκιμαστική φάση

Στη συνέχεια ακολουθεί η διαδικασία της αποσφαλμάτωσης, η οποία είναι εξίσου σημαντική και πολλές φορές χρονοβόρα ανάλογα με την ποιότητα οργάνωσης της προηγούμενης διαδικασίας. Αποτελείτε από αρκετά επιμέρους στάδια που αναλύονται παρακάτω:

- Στο πρώτο στάδιο βρίσκεται ο αρχικός σχεδιασμός της εφαρμογής και η λειτουργία της σε διαδικασία αποσφαλμάτωσης (Debug Mode). Για να γίνει η μεταγλώττιση (Compile) πρέπει να μην υπάρχουν συντακτικά λάθη στον κώδικα. Σε αντίθετη περίπτωση, ο προγραμματιστής καλείτε να διορθώσει για να συνεχίσει στο επόμενο στάδιο. Αφού γίνει επιτυχώς η μεταγλώττιση, η εφαρμογή θα πρέπει να δοκιμαστεί σε συσκευές ώστε να γίνουν αντιληπτά τυχόν προβλήματα λειτουργικά και αισθητικά. Υπάρχουν δυο τρόποι να γίνει αυτό είτε με εικονικές συσκευές μέσω του AVD Manager, είτε απευθείας σε φυσική συσκευή μέσω ADB Push εντολής. Το ADB θα αναλυθεί εκτενώς παρακάτω.
- Στο επόμενο στάδιο ο προγραμματιστής πρέπει να αντιμετωπίσει τα προβλήματα που εντοπίστηκαν στο προηγούμενο στάδιο και να διορθώσει τα κομμάτια του κώδικα που δημιουργούν σφάλματα. Το εργαλείο LogCat μας επιτρέπει να κάνουμε αυτή την διαδικασία, καθώς μας επιστρέφει στο σημείο του κώδικα που συνέβη το σφάλμα.

- Στο τρίτο στάδιο, ο προγραμματιστής αφού έχει τελειώσει την αποσφαλμάτωση επιστρέφει πίσω στο πρώτο στάδιο, δηλαδή στην μεταγλώττιση και έπειτα στην δοκιμή της εφαρμογής σε εικονική ή φυσική συσκευή ώστε να διαπιστώσει τα αποτελέσματα της αποσφαλμάτωσης.
- Στο τελευταίο και προαιρετικό στάδιο, βρίσκεται η Δοκιμαστική φάση. Συγκεκριμένα, η εφαρμογή προσφέρεται σε εθελοντές για τη δοκιμή των λειτουργικών της στις φυσικές συσκευές τους και στη συνέχεια αναφέρουν τα προβλήματα και τις προτάσεις που μπορεί να προκύψουν από την χρήση της εφαρμογής. Η διαδικασία της αποσφαλμάτωσης μοιάζει σαν έναν βρόγχο που επαναλαμβάνεται συνέχεια μέχρι να μηδενιστούν όλα τα σφάλματα της εφαρμογής.

4.2.4 Δημοσίευση

Στο τελευταίο στάδιο της ανάπτυξης, ο προγραμματιστής πρέπει να κάνει μερικές κινήσεις ώστε να ολοκληρώσει με επιτυχία τη διαδικασία αυτή. Αρχικά, η εφαρμογή πρέπει να έχει διορθωθεί και να μην περιέχει σφάλματα και να γίνει η τελική μεταγλώττιση σε κανονική λειτουργία και όχι σε αποσφαλμάτωση. Στη συνέχεια, ακολουθεί η δημοσίευση της εφαρμογής στο μέσο της επιλογής του προγραμματιστή. Αυτό μπορεί να είναι το επίσημο ψηφιακό κατάστημα της Google (Google Play) δημιουργώντας λογαριασμό ή μπορεί να το διαθέσει σε αλλά εναλλακτικά ψηφιακά καταστήματα. Η εφαρμογή μπορεί να διατεθεί με τιμή πώλησης ή δωρεάν.

Επίσης, ένα άλλο μέσο διάθεσης είναι η προσωπική ή εταιρική ιστοσελίδα του προγραμματιστή. Αυτή η επιλογή όμως κρύβει μειονεκτήματα καθώς δεν θα υπάρχει έλεγχος για ενημερώσεις της εφαρμογής σε αντίθεση με τα ψηφιακά καταστήματα όπου υπάρχει αυτόματος μηχανισμός.

4.3 ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ

4.3.1 Android SDK

Το Android SDK (Software Development Kit) είναι το επίσημο εργαλείο της Google για την ανάπτυξη εφαρμογών στο Android. Περιέχει μια συλλογή από εργαλεία και βιβλιοθήκες απαραίτητα για τους προγραμματιστές. Πρώτο βήμα στην πορεία της ανάπτυξης είναι η εγκατάσταση και ρύθμιση του. Περιλαμβάνει παραδείγματα εφαρμογών με τον πηγαίο τους κώδικα, βοηθήματα, πληροφορίες και εξομοιωτή για την εκτίμηση της προόδου της εργασίας. Ακόμη, αναλαμβάνει την μεταγλώττιση του κώδικα ώστε να μπορεί να τρέχει στην εικονική μηχανή Dalvik. Η τελευταία έκδοση μέχρι τη στιγμή που γράφεται η παρούσα εργασία είναι η 24 και είναι γραμμένη σε γλώσσα προγραμματισμού Java. Οι πλατφόρμες που μπορούν να υποστηρίξουν το Android SDK είναι τα συστήματα που τρέχουν MAC OS X 15.5 ή νεότερο, Windows XP ή νεότερο και οποιαδήποτε σύγχρονη με λειτουργικό Linux. Οι γλώσσες που μπορούν να χρησιμοποιήσουν οι προγραμματιστές για να αναπτύξουν εφαρμογές είναι η Java και C++. Το επίσημο περιβάλλον εργασίας που υποστηρίζεται είναι το Eclipse IDE χρησιμοποιώντας το Android Development Tools (ADT) και το NetBeans IDE.

Το Android SDK περιλαμβάνει εξομοιωτή που χρησιμοποιεί το Android Virtual Device (AVD). Το AVD επιτρέπει την δημιουργία διαφορετικού τύπου υλικού της μιμούμενης συσκευής και τη δημιουργία πολλών συνθέσεων ώστε η εφαρμογή να δοκιμάζεται σε ποικιλία παραλλαγών υλικού και να διασφαλίζεται η ομαλή λειτουργία της.

Για την υποστήριξη προηγούμενων εκδόσεων λειτουργικού, το SDK δίνει την δυνατότητα στον προγραμματιστή να επιλέξει το ελάχιστο API που θα στοχεύει η εφαρμογή του. Δηλαδή, είναι εφικτή η επιλογή του εύρους των εκδόσεων που θα τρέχει η εφαρμογή, επιλέγοντας την παλαιότερη και την νεότερη. Αυτό είναι αναγκαίο επειδή πολλοί χρήστες κατέχουν συσκευές με παλαιότερες εκδόσεις Android και ο κατασκευαστής δεν έχει ή δεν πρόκειται να βγάλει ενημέρωση για τις συσκευές αυτές.

4.3.2 Eclipse IDE

Το Eclipse IDE είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης λογισμικού το οποίο είναι γραμμένο στη γλώσσα Java. Χρησιμοποιείται για την ανάπτυξη εφαρμογών Java και μέσω διάφορων επεκτάσεων υποστηρίζει γλώσσες προγραμματισμού όπως C, C++,PHP,Python,Ruby κ.α. Οι πλειοψηφία των εφαρμογών του Android βασίζονται στην γλώσσα Java και ο προγραμματιστής μπορεί να χρησιμοποιήσει έναν Text Editor για να γράψει πηγαίο κώδικα, να επεξεργαστεί αρχεία *.Java και *.XML και να τα μεταγλωττίσει μέσω της γραμμής εντολών χρησιμοποιώντας το JDK (Java Development Kit).Λόγω του ότι ο συγκεκριμένος τρόπος δεν είναι γραφικός και φιλικός στον χρήστη, η καλύτερη επιλογή είναι η χρήση IDE (Integrated Development Environment) που να υποστηρίζει τη Java, όπως το Eclipse και το NetBeans.

Το Eclipse υποστηρίζεται επισημά από την Google, η οποία έχει αναπτύξει το ADT Plugin, που συνδέει τις δυνατότητες του με το Android SDK.Επίσης, παρέχει σύνδεση με τον AVD Manager ώστε να διαχειρίζεται τις εικονικές συσκευές για αποσφαλμάτωση.

Είναι μια εφαρμογή ελεύθερου λογισμικού που υποστηρίζεται και αναπτύσσεται ραγδαία από την παγκόσμια κοινότητα ελεύθερου λογισμικού. Διατίθεται κάτω από την άδεια Eclipse Public License και χρηματοδοτείται από το ίδρυμα Eclipse όπως και από άλλες δωρεές.

4.3.3 Android Studio

Το Android Studio είναι ένα ολοκληρωμένο προγραμματιστικό περιβάλλον (IDE) για ανάπτυξη εφαρμογών στην πλατφόρμα Android. Ανακοινώθηκε στις 16 Μαΐου 2013 στο συνέδριο Google I/O από την Google Product Manager, Katherine Chou. Το Android Studio είναι διαθέσιμο ελεύθερα με την άδεια Apache License 2.0.Το Android Studio ήταν διαθέσιμο σε πρώιμο στάδιο για προεπισκόπηση ξεκινώντας από την έκδοση 0.1 τον Μάιο του 2013, έπειτα ξεκίνησε το δοκιμαστικό στάδιο από την έκδοση 0.8 που βγήκε τον Ιούνιο του 2014. Η πρώτη σταθερή έκδοση βγήκε το Δεκέμβριο του 2014, με την έκδοση 1.0. Βασισμένο στο λογισμικό της JetBrains IntelliJ IDEA, το Android Studio σχεδιάστηκε αποκλειστικά για προγραμματισμό Android. Είναι διαθέσιμο για Windows, Mac OS X και Linux, και αντικατέστησε τα Eclipse Android Development Tools (ADT) ως το κύριο IDE της Google για ανάπτυξη εφαρμογών Android.

4.3.4 Εγκατάσταση του Android Studio

Το Android Studio είναι το βασικό εργαλείο της Google για την ανάπτυξη εφαρμογών Android και διατίθεται δωρεάν από την ιστοσελίδα της Google και συγκεκριμένα από το παρακάτω σύνδεσμο (<https://developer.android.com/studio/index.html>). Ο Προγραμματιστής έχει την δυνατότητα να κατεβάσει την πλατφόρμα και να την εγκαταστήσει. Παρακάτω θα δούμε τα βήματα της εγκατάστασης και τι υλικό (Hardware) χρειάζεται για να δουλέψει ομαλά η πλατφόρμα.

Αρχικά θα πρέπει να σιγουρευτούμε ότι ο Ηλεκτρονικός Υπολογιστής, διαθέτει ένα από τα παρακάτω λειτουργικά συστήματα ή νεότερα από αυτά:

- Microsoft® Windows® 8/7/Vista/2003 (32 or 64-bit).
- Mac® OS X® 10.8.5 ή πάνω από 10.9 (Mavericks)
- GNOME ή KDE desktop

Στη συνέχεια θα πρέπει για να δουλέψει η πλατφόρμα μας να έχουμε εγκαταστήσει και το πακέτο Java το οποίο κατεβάζουμε και αυτό δωρεάν από την ιστοσελίδα της Oracle παρακάτω δίνεται ο σύνδεσμος (<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>).

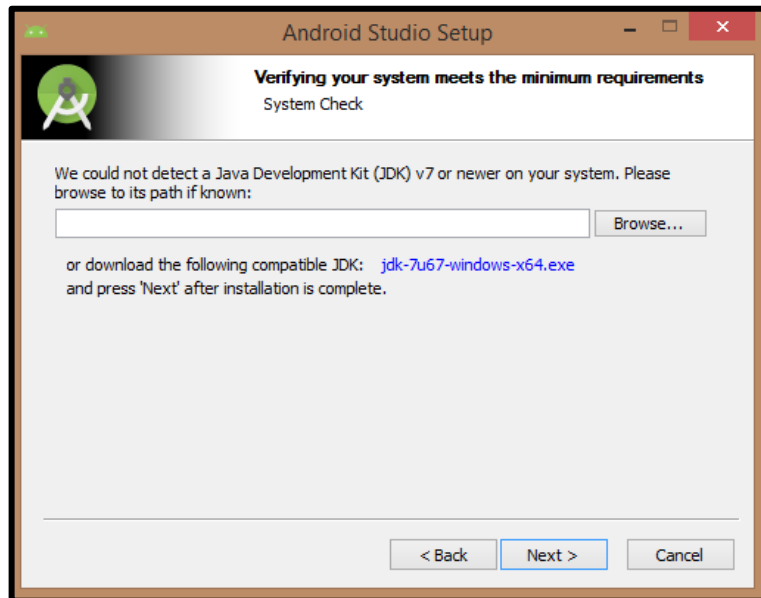
Αναφέρουμε κάποια χαρακτηριστικά που θα πρέπει να έχει το πακέτο Java.

- Java JDK5 or later version
- Java Runtime Environment (JRE) 6
- Android Studio

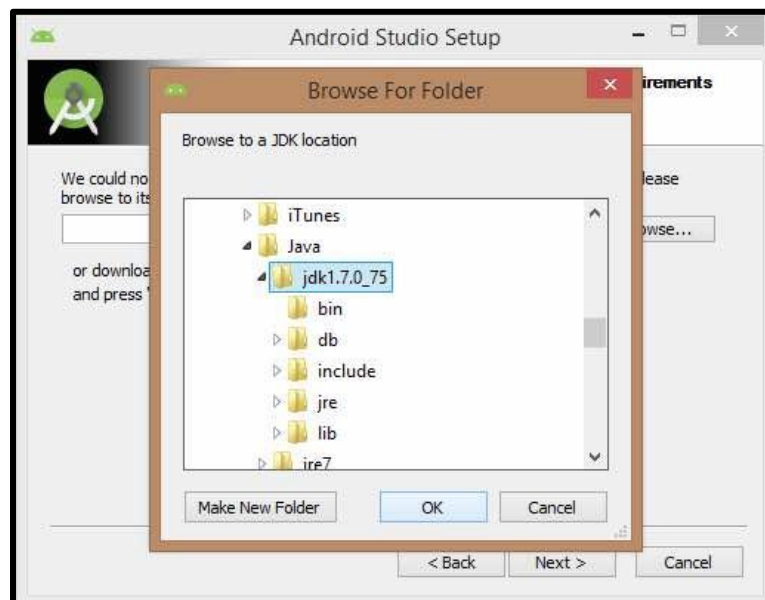


Εικόνα 25: Διαδικασία εγκατάστασης του Android Studio
(http://www.tutorialspoint.com/android/android_studio.htm)

Καθώς έχουμε κατεβάσει το αρχείο *Android studio.exe* στον Υπολογιστή μας πατάμε κλικ για να ξεκινήσει η εγκατάσταση. Στη συνέχεια πατάμε Next (Επόμενο). Και οδηγούμαστε στο παρακάτω παράθυρο, στην συγκεκριμένη περίπτωση θα πρέπει να διαλέξουμε το αρχείο Java που κατεβάσαμε από την ιστοσελίδα της Oracle.

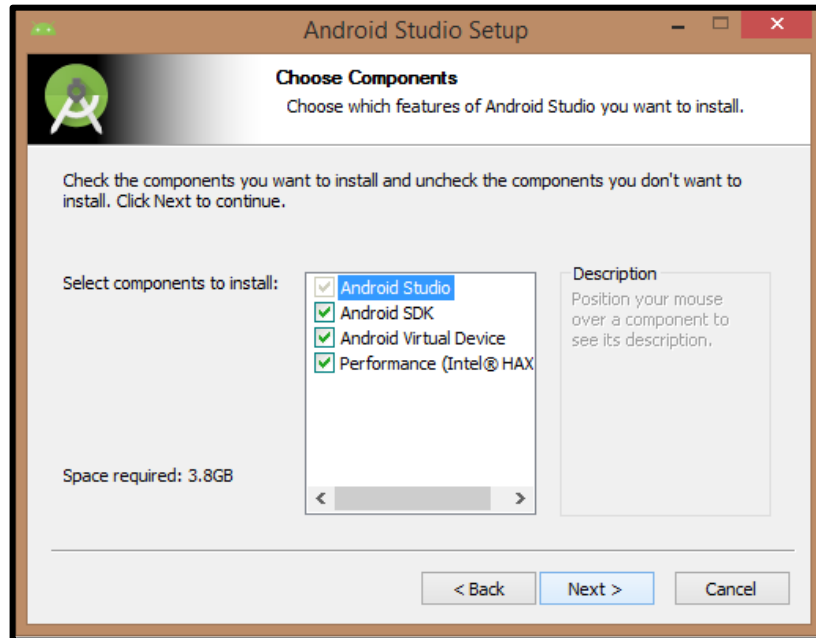


Εικόνα 26: Διαδικασία επιλογής του αρχείου java
(http://www.tutorialspoint.com/android/android_studio.htm)



Εικόνα 26: Επιλογή του αρχείου java
(http://www.tutorialspoint.com/android/android_studio.htm)

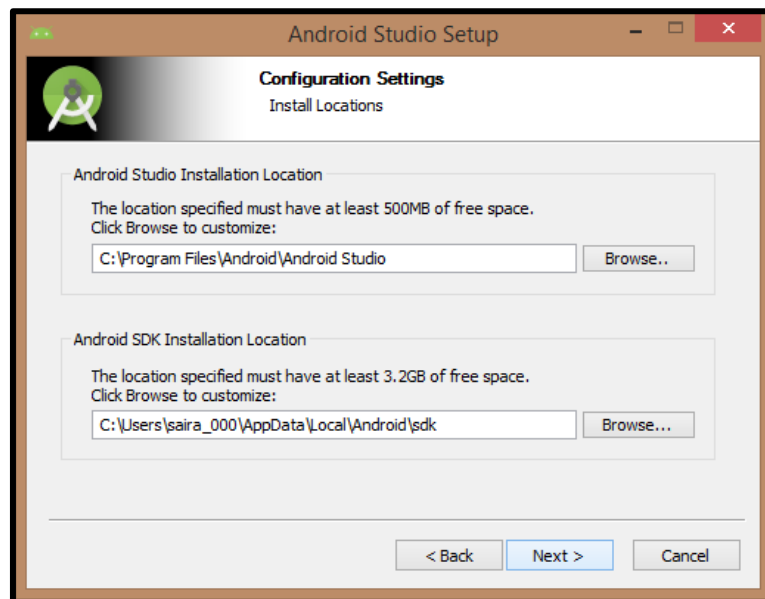
Επιλέγοντας το κουμπί Browse (Αναζήτηση) βρίσκουμε το αρχείο Jdk που έχουμε κατεβάσει στη συνέχεια πατάμε το κουμπί Ok. Οδηγούμαστε στην αρχική οθόνη και πατάμε το κουμπί Next (Επόμενο).



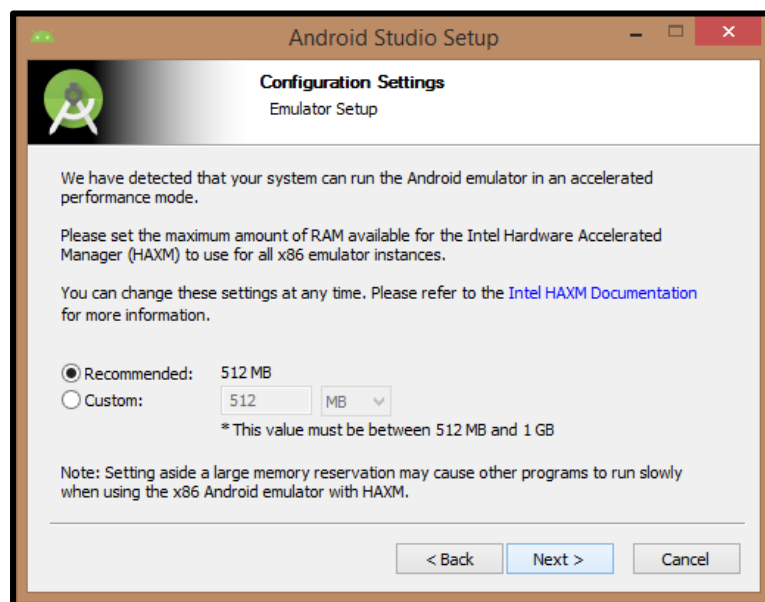
Εικόνα 27: Επιλογή των παραμέτρων

(http://www.tutorialspoint.com/android/android_studio.htm)

Στη συγκεκριμένη περίπτωση επιλέγουμε τις ιδιότητες που θα έχει η πλατφόρμα μας. Κάτω δεξιά μπορούμε να ελέγξουμε και το χώρο που πιάνουν τα χαρακτηριστικά που προσθέτουμε. Κλικ άρουμε στο Next (Επόμενο).

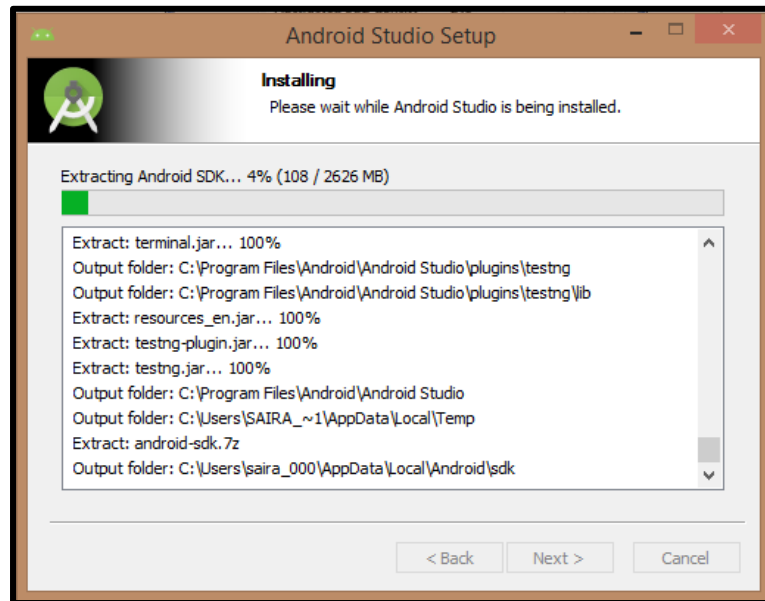


Εικόνα 28: Επιλογή της διαδρομής εγκατάστασης
(http://www.tutorialspoint.com/android/android_studio.htm)



Εικόνα 29: Ρυθμίσεις του Emulator
(http://www.tutorialspoint.com/android/android_studio.htm)

Ρυθμίζουμε το Emulator στο οποίο θα πραγματοποιηθούν οι δοκιμές των εφαρμογών. Παρατηρούμε ότι μπορούμε να αυξήσουμε την μνήμη Ram ή να την ελαττώσουμε αντίστοιχα. Κλικ άρουμε το κουμπί Next (Επόμενο).



Εικόνα 30: Εγκατάσταση της πλατφόρμας

(http://www.tutorialspoint.com/android/android_studio.htm)

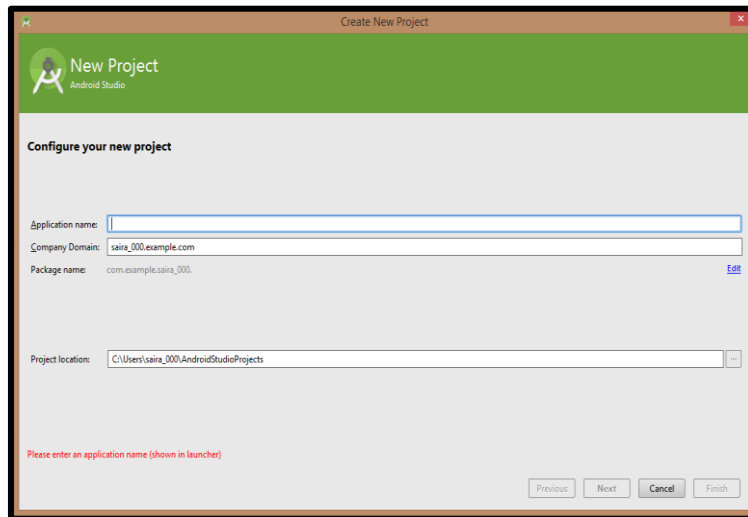
Πραγματοποιείται η εγκατάσταση, μόλις ολοκληρωθεί πατάμε το κουμπί Next (Επόμενο).



Εικόνα 31: Αρχικό περιβάλλον της πλατφόρμας

(http://www.tutorialspoint.com/android/android_studio.htm)

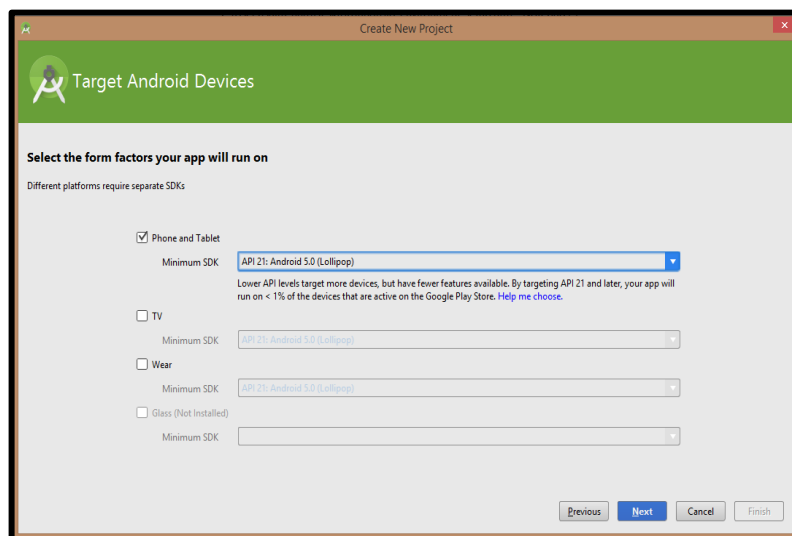
Η εγκατάσταση έχει ολοκληρωθεί με επιτυχία πλέον, ο προγραμματιστής μπορεί να προγραμματίσει τις δίκες του εφαρμογές. Στη συνέχεια θα αναλύσουμε τα βήματα πως μπορούμε να φτιάξουμε ένα Project. Αρχικά επιλέγουμε το *Start a new Android Project* (Δημιουργία νέου Android Project).



Εικόνα 32: Δημιουργία Project

(http://www.tutorialspoint.com/android/android_studio.htm)

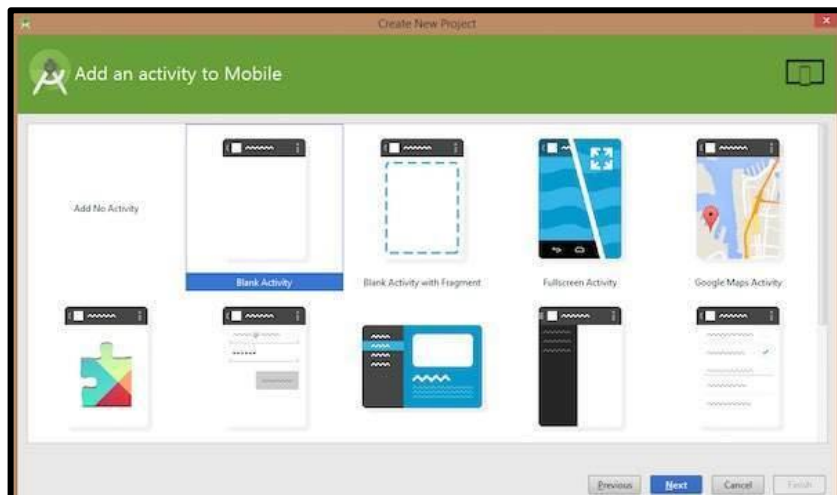
Στο παράθυρο αυτό συμπληρώνουμε το όνομα της εφαρμογής πχ First Aid, βάζουμε το όνομα της εταιρίας παραδείγματος χάρη My Home και στο Project Location βάζουμε την διαδρομή της εφαρμογής που θα αποθηκευτεί πχ Desktop (Επιφάνεια Εργασίας).



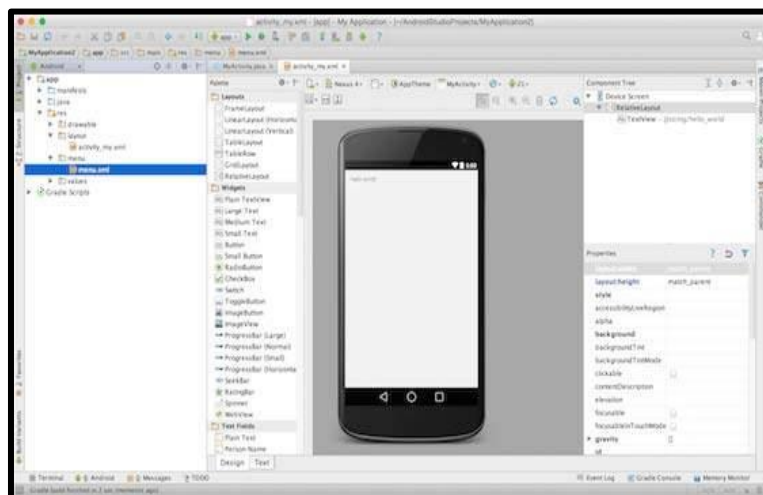
Εικόνα 33: Επιλογή παραμέτρων για την εφαρμογή

(http://www.tutorialspoint.com/android/android_studio.htm)

Στο παράθυρο αυτό επιλέγουμε εάν η εφαρμογή που δημιουργείτε προορίζεται για έξυπνο κινητό τηλέφωνο (Smartphone) για Tablet για Έξυπνη τηλεόραση ή για έξυπνο ψηφιακό ρολόι. Επίσης έχουμε την δυνατότητα να επιλέξουμε και την έκδοση του Android στο οποίο θα τρέξει η εφαρμογή. Επιλέγουμε το Next(Εεπόμενο)



Εικόνα 34: Επιλογή οθόνης της εφαρμογής
(http://www.tutorialspoint.com/android/android_studio.htm)



Εικόνα 35: Περιβάλλον εργασίας
(http://www.tutorialspoint.com/android/android_studio.htm)

Τέλος, φτάνουμε στο περιβάλλον εργασίας όπου εδώ ξεκινάμε να υλοποιήσουμε την εφαρμογή.

ΚΕΦΑΛΑΙΟ 5°

5.1 ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ ΜΙΑΣ ΕΦΑΡΜΟΓΗΣ

Τα δομικά στοιχεία μιας εφαρμογής Android μπορούμε να τα διακρίνουμε σε τέσσερις κατηγορίες, στις δραστηριότητες (Activities), στις υπηρεσίες (Services), στους πάροχους περιεχομένου (content providers) και τέλος στους καθολικούς παραλήπτες μηνυμάτων (Broadcast receivers).

5.1.1 Activities

Είναι το σημαντικότερο στοιχείο μιας εφαρμογής. Κάθε Activity αποτελεί ένα παράθυρο της εφαρμογής, ουσιαστικά μια διαφορετική οθόνη, είναι αυτό που φαίνεται στον χρήστη, εδώ φορτώνεται το γραφικό περιβάλλον της εφαρμογής.

Η πρώτη Activity που βλέπει ο χρήστης είναι η Main Activity από την οποία θα έχει πρόσβαση και σε όλες τις υπόλοιπες. Ουσιαστικά πολλές Activities μαζί μπορούν να συνδέονται μεταξύ τους.

Όταν ο χρήστης καλέσει μια νέα οθόνη ουσιαστικά καλείται μια νέα Activity, με το πλήκτρο Back μπορεί να επιστρέψει στην προηγούμενη Activity (Οθόνη), δηλαδή έχουμε μια μορφή στοίβας. Δεν είναι όμως απαραίτητο όλες οι Activity να είναι αποθηκευμένες στην στοίβα, ένα παράδειγμα στο οποίο είναι απαραίτητο να μην είναι αποθηκευμένη στην στοίβα είναι όταν ο χρήστης πατήσει το Back και ενώ θα είναι στην Main Activity θα πρέπει να βγει από την εφαρμογή.

5.1.2 Services

Εκτός από την Activity με την οποία έχει άμεση επαφή ο χρήστης, από πίσω κάποιο στοιχείο της εφαρμογής εκτελεί κάποιες διεργασίες αυτή είναι η Service, η οποία μπορεί να εκτελείται από μια διαφορετική εφαρμογή από αυτήν που αρχικά την κάλεσε ενώ δεν έχει κάποιο User Interface.

Παράδειγμα υπηρεσίας είναι η εφαρμογή πλοήγησης, χωρίς απαραίτητα να βλέπουμε κάποιο γραφικό περιβάλλον, θέλουμε να τρέχει η εφαρμογή ενώ παράλληλα μπορούμε να χρησιμοποιήσουμε το κινητό τηλέφωνο για μια εισερχόμενη κλήση. Στην περίπτωση αυτή θέλουμε να τρέχει η εφαρμογή, ώστε όταν χρειαστεί να την τρέξουμε, χωρίς όμως να την φορτώσουμε ξανά από την αρχή.

Δυο είναι οι βασικές λειτουργίες που έχουν άμεση σχέση με τις services η `startService()` και η `bindService()`. Η πρώτη καλείται από κάποια Activity για να ξεκινήσει η Service και συνεχίζει να εκτελείται ακόμα και όταν η Activity που την κάλεσε τερματίσει. Ένα παράδειγμα που χρησιμοποιείται η `startService()` είναι όταν θέλουμε να κατεβάσουμε ένα αρχείο από το Play Store. Μπορεί εμείς να έχουμε κλείσει την εφαρμογή παρόλα αυτά το αρχείο συνεχίζει να κατεβαίνει. Στην δεύτερη είναι δεμένη από μια Activity, όταν για κάποιο λόγο τερματιστεί η Activity τερματίζει και η Service.

Τέλος, πρέπει να τονίσουμε ότι είναι απαραίτητο να δηλώνεται ο τερματισμός της Service διότι συνεχίζει να τρέχει καταναλώνοντας υπολογιστικούς πόρους.

5.1.3 Content Providers

Οι content providers διαχειρίζονται αποθηκευτικούς χώρους δεδομένων που είναι προσπελάσιμοι από περισσότερες εφαρμογές. Είναι ο μοναδικός τρόπος με τον οποίο μπορούν να αποθηκευτούν δεδομένα από αρκετές εφαρμογές. Για παράδειγμα το ενδεχόμενο να χρησιμοποιηθούν οι επαφές ενός τηλεφώνου από μια άλλη εφαρμογή θα είναι πολύ πιθανό, οπότε θα πρέπει οι επαφές να αποθηκευτούν σε κάποιο συγκεκριμένο χώρο ώστε μελλοντικά να μπορεί μια άλλη εφαρμογή να μπορέσει να τις επεξεργαστεί. Το αρχείο βίντεο, εικόνας και ήχου είναι επίσης αρχεία τα οποία πρέπει και αυτά να αποθηκευτούν σε κάποιο αντίστοιχο χώρο.

Στο πλαίσιο αυτό υπάρχει ενσωματωμένη βάση δεδομένων στο λειτουργικό σύστημα Android (SQLite Database), στην οποία μπορούν να αποθηκευτούν η να διαβάζουν δεδομένα οι Content Providers.

5.1.4 Broadcast Receivers

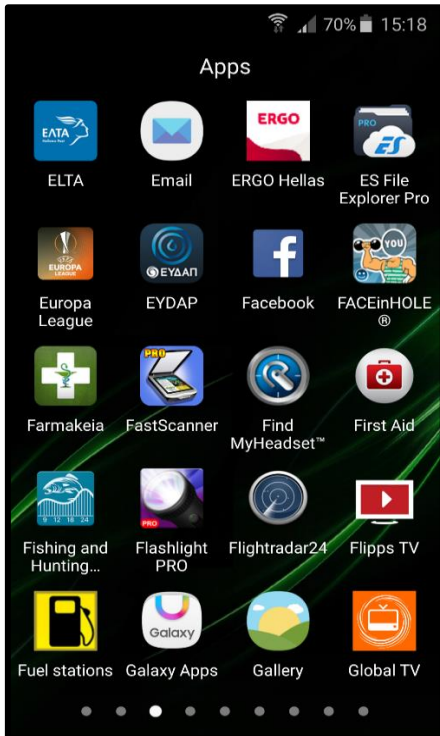
Οι Broadcast Receivers ενημερώνονται για κάποιο συγκεκριμένο γεγονός από το λειτουργικό σύστημα και τότε ενεργοποιούνται. Τέτοια γεγονότα μπορεί να είναι ότι η οθόνη έχει σβήσει, ότι η στάθμη της μπαταρίας είναι χαμηλή κτλ. Για παράδειγμα αρκετές εφαρμογές λήψης φωτογραφιών απενεργοποιούν το Flash, σε περίπτωση χαμηλής μπαταρίας για λογούς εξοικονόμησης ενέργειας. Ενημερώνονται δηλαδή από τον Broadcast Receiver για το γεγονός αυτό και πράττουν ανάλογα. Οι Broadcast Receivers δεν παρέχουν User Interface, αλλά μπορούν να ενημερώσουν το χρήστη για κάποιο γεγονός μέσω του Status Bar Notifications.

5.2 ΕΦΑΡΜΟΓΗ ΠΡΩΤΕΣ ΒΟΗΘΕΙΕΣ (First Aid)

Η εφαρμογή Πρώτων βοηθειών (First Aid) παρέχει άμεσα οπτικοακουστικές οδηγίες ώστε ο χρήστης να διαχειριστεί με ασφάλεια και αποτελεσματικά τα πιο κοινά επείγοντα περιστατικά. Η εφαρμογή περιλαμβάνει:

- ✓ Απλές οπτικοακουστικές οδηγίες, βήμα προς βήμα, τα οποία καθοδηγούν το χρήστη να μάθει και να αντιμετωπίζει με ασφάλεια καθημερινά επείγοντα περιστατικά.
- ✓ Καταχωρημένα επείγοντα τηλεφωνα όπως ΕΚΑΒ ,Αστυνομία, Πυροσβεστική, Κέντρο Δηλητηριάσεων κλπ.
- ✓ Τηλεφωνα Ψυχολογικής υποστήριξης.
- ✓ Τοποθεσία Gps έτσι ώστε ο χρήστης σε μια έκτακτη ανάγκη να γνωρίζει που βρίσκεται.
- ✓ Ιατρικός Φάκελος Ασθενή όπου αποθηκεύονται όλα τα στοιχεία του χρήστη καθώς και ο ιατρικός του φάκελος, πχ τι εξετάσεις έχει κάνει, τα αποτελέσματα των εξετάσεων και ποιες εξετάσεις έχει προγραμματίσει, και υποθετικά αυτά ελέγχονται από ιατρό. Αυτό έχει ως σκοπό ο χρήστης να μην χάνει τις εξετάσεις του και να έχει μια συνεχή παρακολούθηση από ιατρό.
- ✓ Επίσης περιλαμβάνει και τα Στοιχεία κατόχου, όπου ο κάθε χρήστης της εφαρμογής έχει την δυνατότητα να αποθηκεύει τα στοιχεία του όπως το ονοματεπώνυμο του, την διεύθυνση κατοικίας, τα τηλεφωνα του καθώς και διάφορα σχόλια, όλα αυτά αποθηκεύονται στο κινητό του χρήστη.

5.3 ΠΑΡΣΟΥΣΙΑΣΗ ΕΦΑΡΜΟΓΗΣ ΠΡΩΤΩΝ ΒΟΗΘΕΙΩΝ (First Aid)



Εικόνα 36: Εικονίδιο Εφαρμογής



Εικόνα 37: Αρχική Οθόνη Εφαρμογής

Στη συνέχεια θα αναλύσουμε τα κουμπιά της εφαρμογής. Αρχικά πατάμε στο κουμπί **Πρώτες Βοήθειες** και οδηγούμαστε στην παρακάτω οθόνη.



Εικόνα 38: Περιβάλλον Πρώτων Βοηθειών 1/3

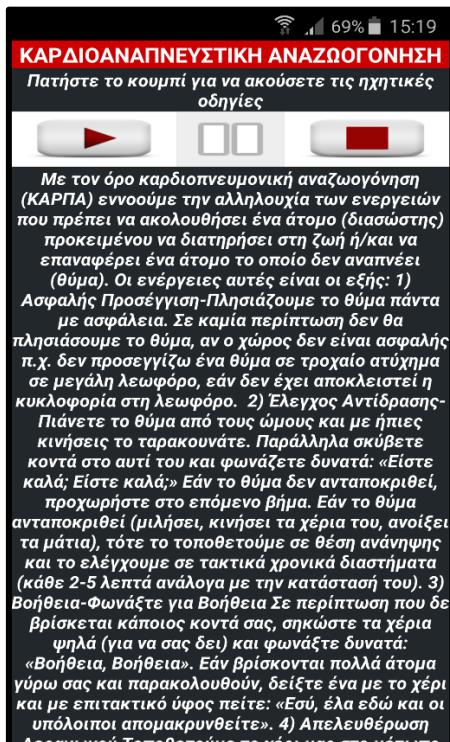


Εικόνα 39: Περιβάλλον Πρώτων Βοηθειών 2/3



Εικόνα 40: Περιβάλλον Πρώτων Βοηθειών 3/3

Στην προκειμένη περίπτωση βλέπουμε τις διαθέσιμες επιλογές ανάλογα με το περιστατικό που αντιμετωπίζουμε, στην συνέχεια εάν επιλέξουμε κάποιο περιστατικό όπως Καρδιοαναπνευστική αναζωογόνηση οδηγούμαστε στο παρακάτω παράθυρο όπου διαπιστώνουμε σε τι ενέργειες θα πρέπει να προβούμε, όσοι έχουν προβλήματα οράσεως έχουν την δυνατότητα των ηχητικών οδηγιών.

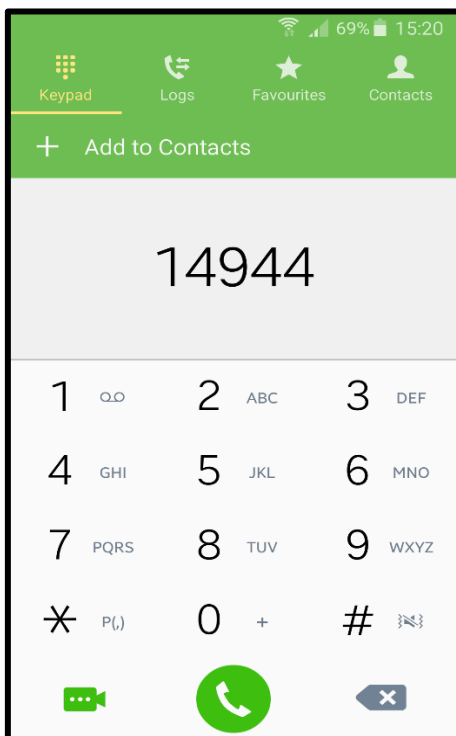


Εικόνα 41: Οδηγίες Καρδιοαναπνευστικής Αναζωογόνησης

Στη συνέχεια οδηγούμαστε στο κουμπί *Επείγοντα και Εφημερεύοντα* όπου είναι διαθέσιμα όλα τα τηλεφωνα έκτακτης ανάγκης. Τα τηλεφωνα είναι καταχώρησα στην εφαρμογή οπότε ο χρήστης δεν χρειάζεται να πληκτρολόγησε κάποιο τηλέφωνο, αρκεί να επιλέξει το κουμπί που χρειάζεται βοήθεια και η εφαρμογή καλεί αμέσως τον αριθμό.

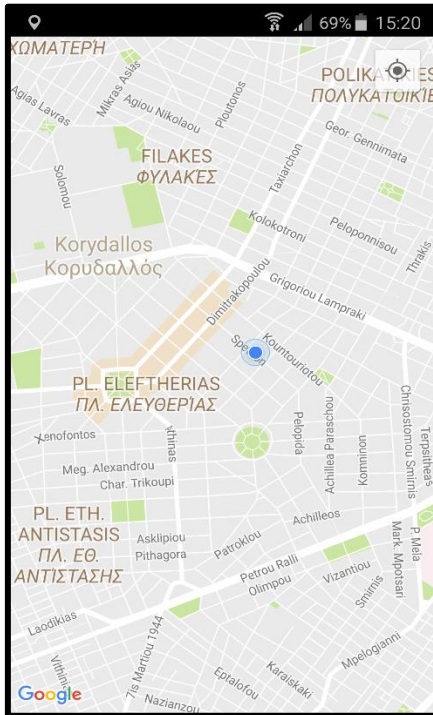


Εικόνα 42: Τηλέφωνα Έκτακτης Ανάγκης Ελλάδος

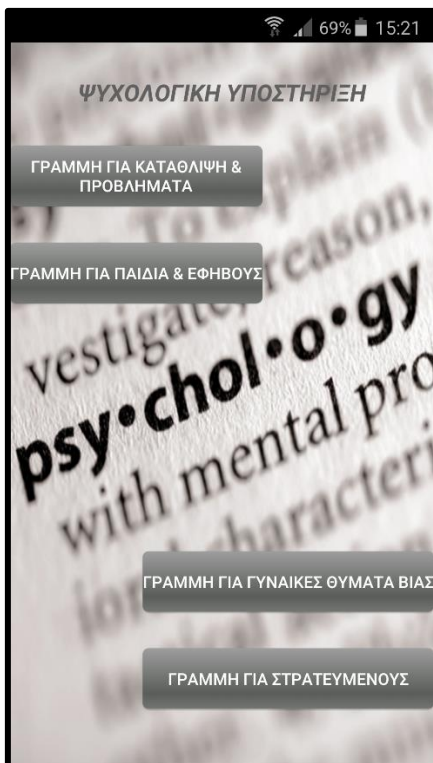


Εικόνα 43: Τηλέφωνο για Εφημερεύοντα Νοσοκομεία

Πολύ χρήσιμο κουμπί είναι και το κουμπί της **Τοποθεσίας Gps** όπου ο χρήστης έχει την δυνατότητα να δει που βρίσκεται εκείνη την στιγμή την τοποθεσία του. Επίσης το κουμπί **Ψυχολογική Υποστήριξη** περιλαμβάνει διάφορα τηλεφώνια ψυχολογικής υποστήριξης ανά κατηγορία.



Εικόνα 44: Τοποθεσία Χρήστη

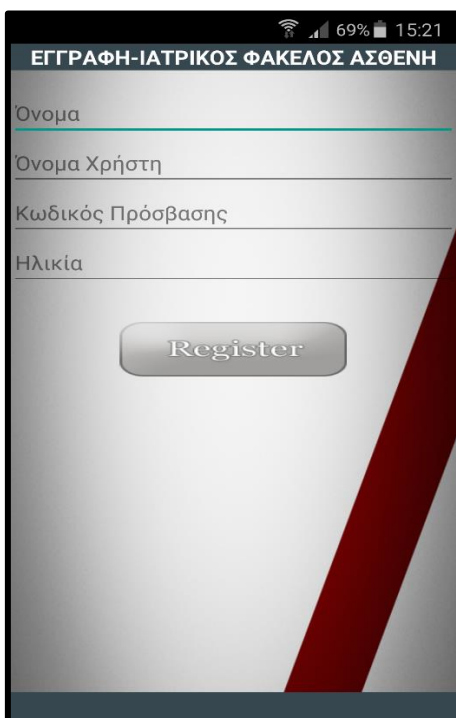


Εικόνα 45: Τηλέφωνα Ψυχολογικής Υποστήριξης

Στη συνέχεια θα δούμε τον **Ιατρικό Φάκελο Ασθενή** όπου ο χρήστης έχει την δυνατότητα να δημιουργήσει το δικό του προφίλ όπως βλέπουμε παρακάτω. Για να το επιτύχει αυτό επιλέγει το κουμπί **Create Account** από την στιγμή που έχει συμπληρώσει όνομα, όνομα χρήστη κωδικό πρόσβασης και ηλικία πατώντας το κουμπί **Register** καταχωρεί τα στοιχεία του στην βάση δεδομένων με αποτέλεσμα να έχει δημιουργήσει το δικό του λογαριασμό.



Εικόνα 46: Ιατρικός Φάκελος Ασθενή- Αρχική Οθόνη

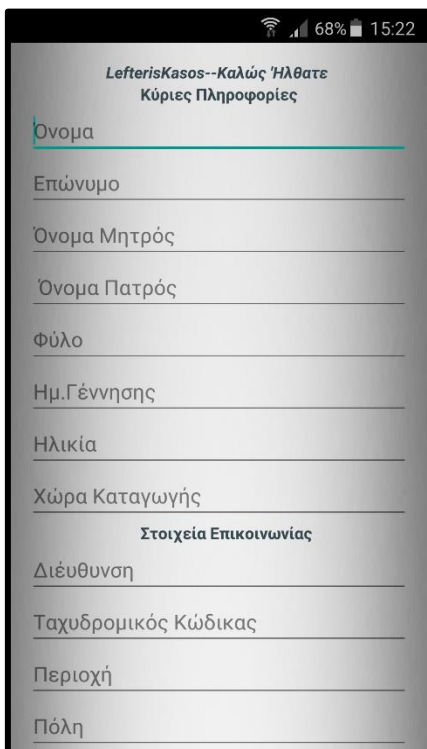


Εικόνα 47: Ιατρικός Φάκελος Ασθενή – Εγγραφή

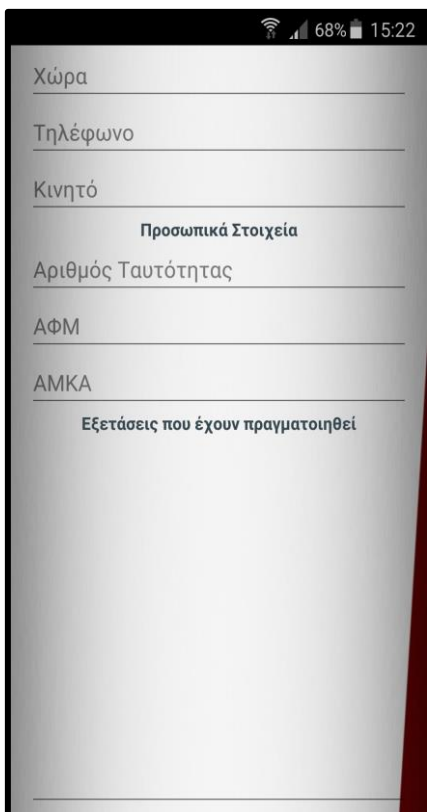
Καθώς έχουμε δημιουργήσει το δικό μας λογαριασμό, στην αρχική οθόνη του Ιατρικού φακέλου ασθενή γράφουμε το Όνομα Χρήστη και τον Κωδικό Πρόσβασης και πλέον έχουμε πραγματοποιήσει επιτυχώς εισαγωγή στο σύστημα της βάσης δεδομένων. Συμπληρώνουμε όλο το Ιατρικό Φάκελο και στο τέλος πατάμε **Εγγραφή στη Βάση Δεδομένων**. Γίνεται επιτυχής εγγραφή στην βάση δεδομένων όπου αυτό σημαίνει ότι την βάση δεδομένων που την χειρίζεται ο γιατρός έχει ενημερωθεί πλήρως. Εάν σε περίπτωση θέλουμε να δούμε τις εγγραφές πατάμε στο κουμπί **Ανάκτηση Πληροφοριών από την Βάση Δεδομένων**, αυτό έχει ως αποτέλεσμα να μας επιστραφεί το **Όνομα το Επώνυμο και το Πατρώνυμο**, όλα τα υπόλοιπα στοιχεία λόγω ότι είναι προσωπικά δεδομένα τα διαχειρίζεται μόνο ο γιατρός.



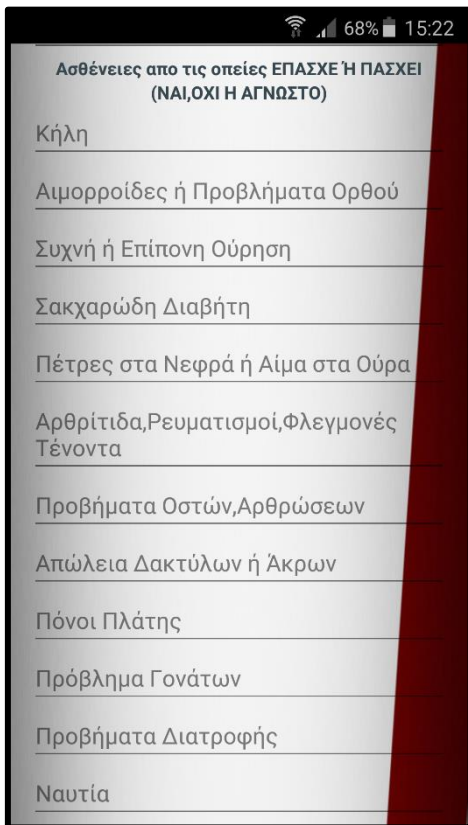
Εικόνα 48: Ιατρικός Φάκελος- Είσοδος Χρήστη



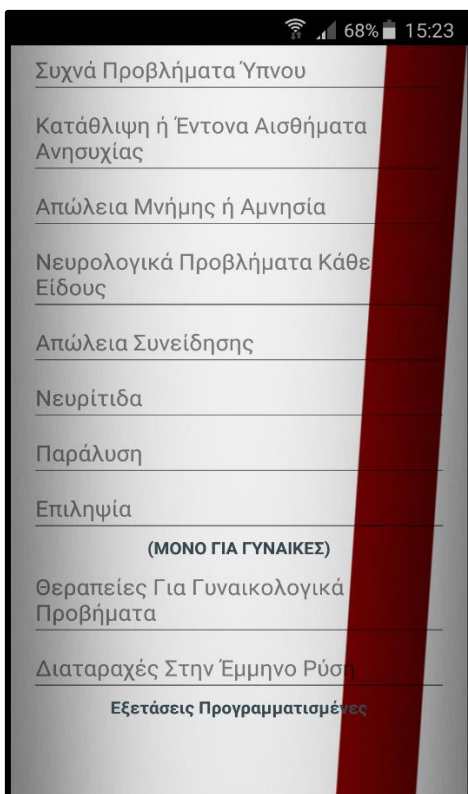
Εικόνα 49: Ιατρικός Φάκελος Ασθενή- Στοιχεία 1/5



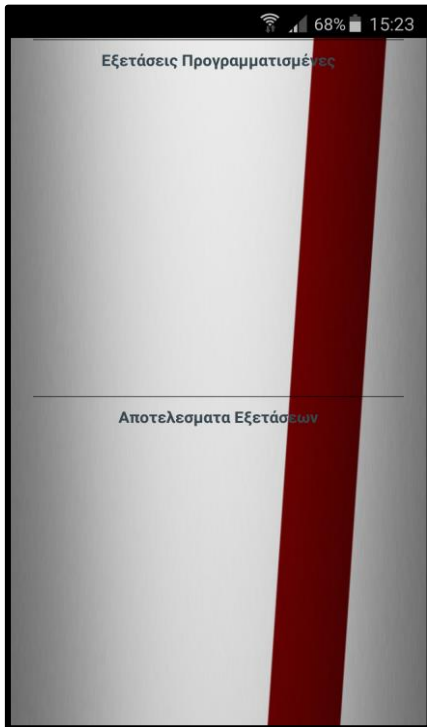
Εικόνα 50: Ιατρικός Φάκελος Ασθενή- Στοιχεία 2/5



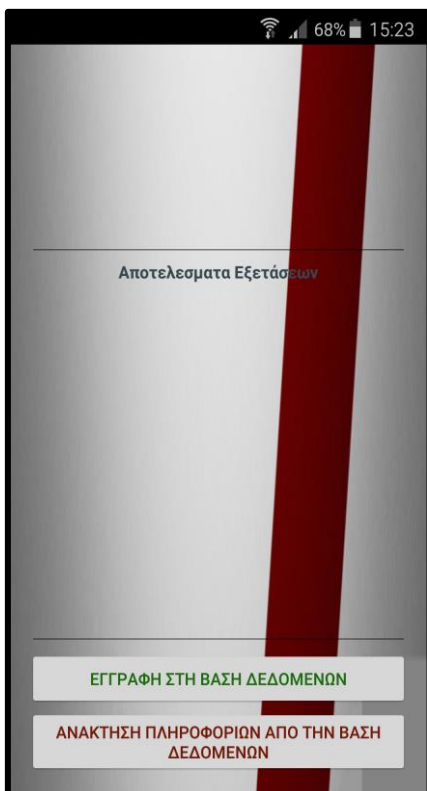
Εικόνα 51: Ιατρικός Φάκελος Ασθενή- Στοιχεία 3/5



Εικόνα 52: Ιατρικός Φάκελος Ασθενή- Στοιχεία 4/5

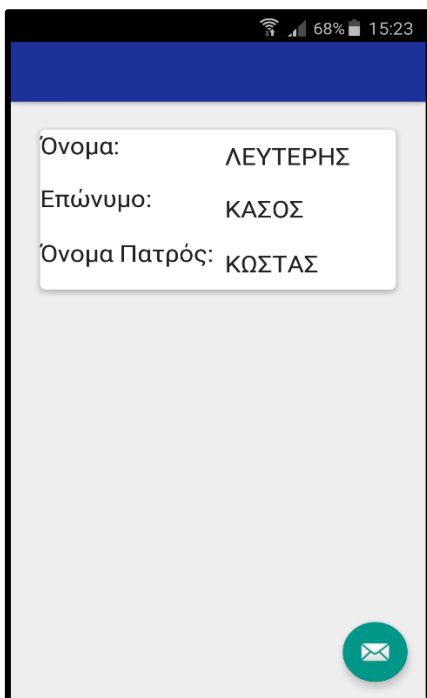


Εικόνα 53: Ιατρικός Φάκελος Ασθενή- Στοιχεία 5/5



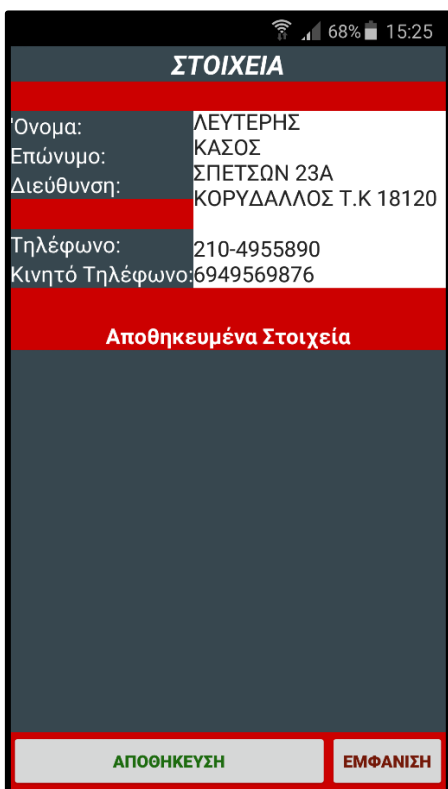
Εικόνα 54: Ιατρικός Φάκελος Ασθενή- Εγγραφή στη Βάση

Δεδομένων - Ανάκτηση Πληροφοριών από την Βάση Δεδομένων

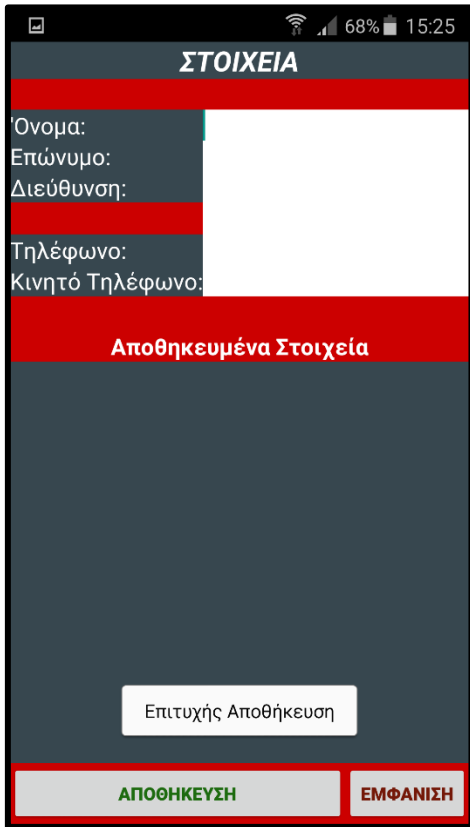


Εικόνα 55: Ανάκτηση Εγγεγραμμένων Χρηστών

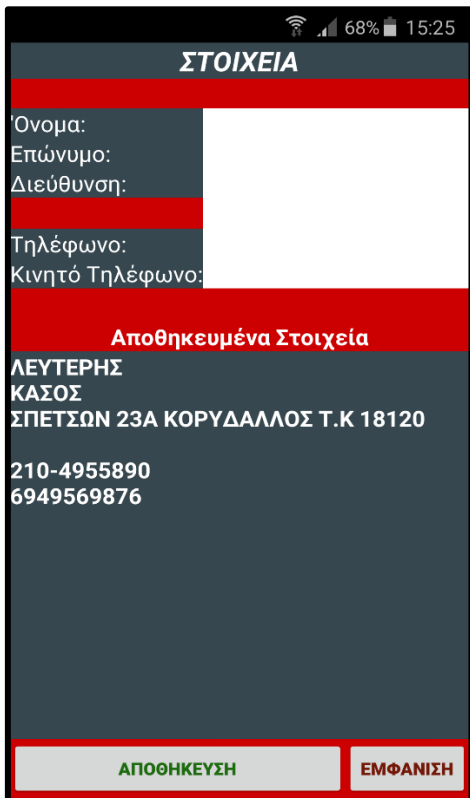
Στη συνέχεια προχωράμε στο κουμπί **Στοιχεία Κατόχου** όπου ο κάθε χρήστης της εφαρμογής μπορεί να συμπληρώσει τα στοιχεία του και αυτά αποθηκεύονται τοπικά στο κινητό.



Εικόνα 56: Στοιχεία Κατόχου Εφαρμογής

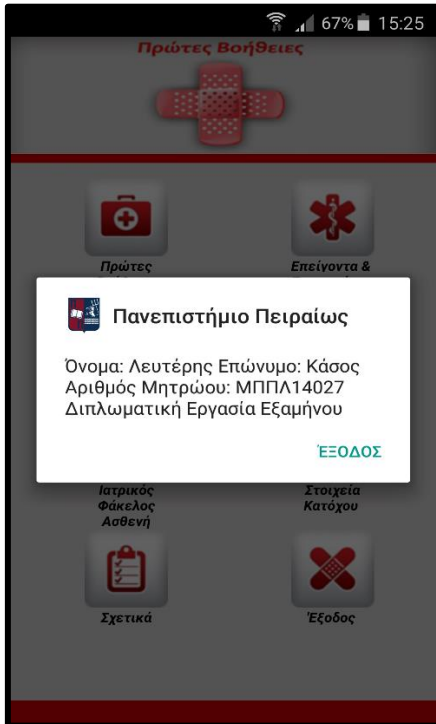


Εικόνα 56: Αποθήκευση Στοιχείων Κατόχου Εφαρμογής

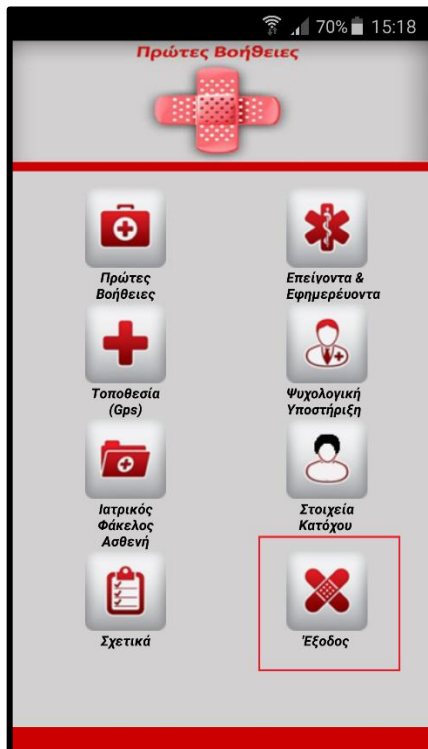


Εικόνα 57: Εμφάνιση στοιχείων Κατόχου Εφαρμογής

Τέλος μας απομένει το κουμπί **Στοιχεία** όπου γράφουμε κάποια πράγματα όπως τον δημιουργό της εφαρμογής και για πιο σκοπό προορίζεται η εφαρμογή, και τέλος το κουμπί έξοδος όπου μας οδηγεί στην αρχική οθόνη του Android.



Εικόνα 58: Στοιχεία Δημιουργού και Προορισμός Εργασίας



Εικόνα 60: Έξοδος Εφαρμογής

ΚΕΦΑΛΑΙΟ 6°

6.1 ΔΟΜΗ ΕΦΑΡΜΟΓΗΣ ΠΡΩΤΩΝ ΒΟΗΘΕΙΩΝ (FIRST AID)

Η εφαρμογή μας έχει γραφεί σε *Java*, *Xml* και σε ένα σημείο χρησιμοποιείται *Php* και *JavaScript*. Αποτελείτε από 36 κλάσεις. Περιλαμβάνει *Xml* αρχεία, 5 για τον ορισμό τιμών των πόρων που χρησιμοποιεί το σύστημα (*colors*, *dimens*, *strings*, *styles*, *google_maps_api*), 1 για το αρχείο *Manifest*, 31 για να υλοποιούνται τα *layouts* των οθονών της εφαρμογής και 1 που αφορά το μενού της κεντρικής οθόνης. Ακόμα περιλαμβάνονται 37 πόροι εικόνων, καθώς και 17 αρχεία ήχου .mp3.

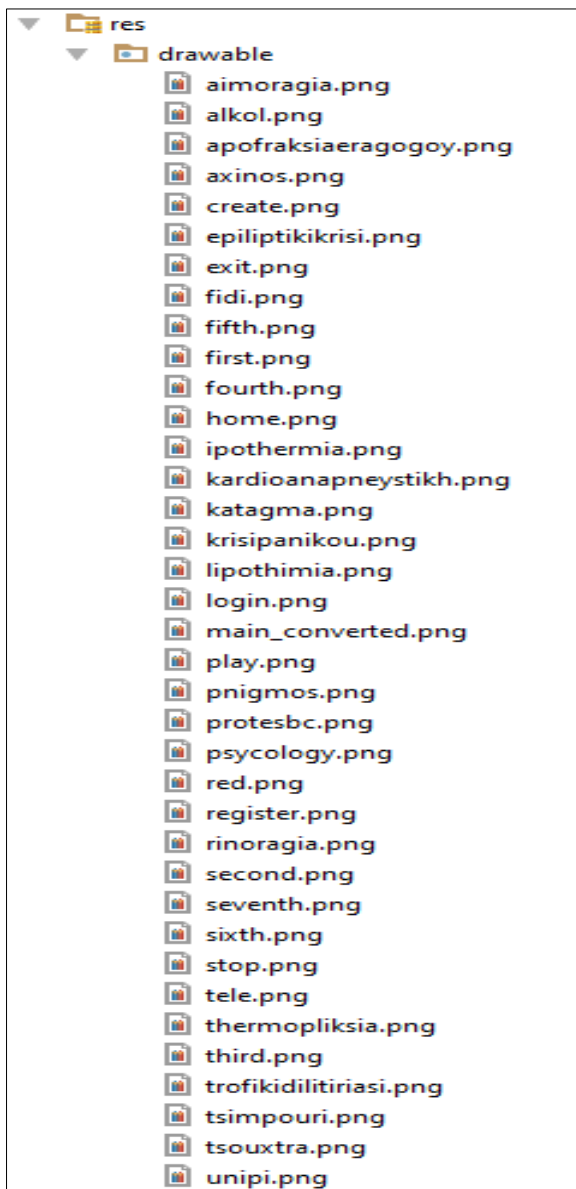
6.2 ΦΑΚΕΛΟΣ RES

Ο φάκελος *res* (*resources*) περιέχει τους πόρους της εφαρμογής οι οποίοι μπορεί να είναι αρχεία *Xml*, εικόνες, ήχοι, *animation* και *string* τιμές. Ο μεταγλωττιστής πόρων (*resource compiler*) κατηγοριοποιεί τους πόρους βάσει του τύπου τους, για παράδειγμα οι εικόνες βρίσκονται στον υποφάκελο *drawable*. Σε περίπτωση που ο προγραμματιστής χρειάζεται να δημιουργήσει νέο υποφάκελο για της ανάγκες της εφαρμογής του είναι ελεύθερος να το κάνει, καθώς το *Android Studio* δεν έχει κανένα περιορισμό και δίνει αυτή την δυνατότητα.

6.3 ΥΠΟΦΑΚΕΛΟΣ RES/DRAWABLE

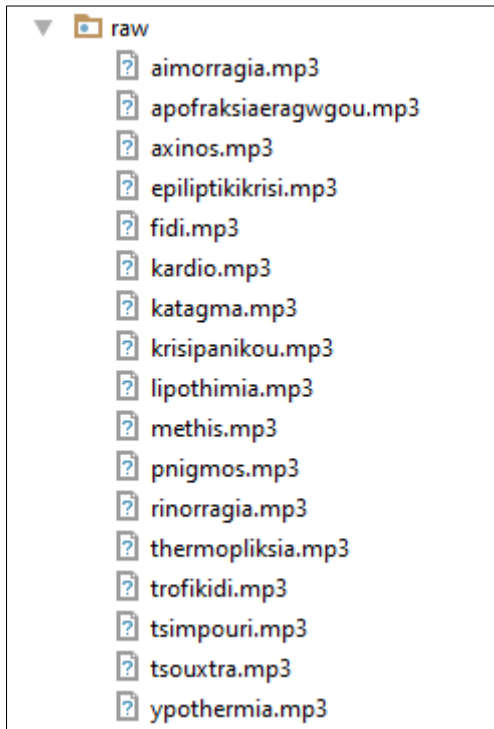
Στον υποφάκελο *drawable* βρίσκονται κυρίως τα αρχεία που είναι στατικά γραφικά, όπως για παράδειγμα εικόνες, σχήματα, και *animation*. Το *Android* υποστηρίζει αρχεία εικόνων τύπου *PNG*, *JPG*, *GIF*, και αρχεία *XML* τα οποία ορίζουν σχήματα, χρώματα, διαγράμματα ,σχέδια που αλλάζουν το μέγεθος τους ανάλογα με την οθόνη και εικόνες που εναλλάσσουν την κατάσταση τους σε καθορισμένα γεγονότα.

Με την δημιουργία νέου *Project* το *Android Studio* δημιουργεί τέσσερεις υποφακέλους *drawable* οι οποίοι κατηγοριοποιούνται σε *ldpi* (120 dpi) 36 x 36 px, *mdpi* (160 dpi) 48x 48 px, *hdpi* (240 dpi) 72 x 72 px και *xhdpi* (320 dpi) 96 x 96 px.

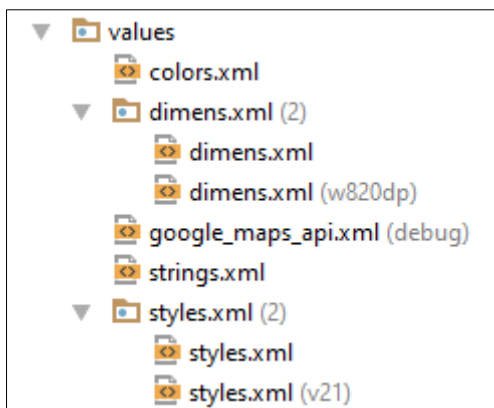


6.4 ΥΠΟΦΑΚΕΛΟΣ RAW

Ο υποκατάλογος raw οργανώνει τα ακατέργαστα αρχεία που μπορεί να χρησιμοποιεί η εφαρμογή, π.χ. αρχεία ήχου, βίντεο. Στη περίπτωση μας χρησιμοποιούμε αρχεία ήχου .mp3



6.5 ΔΙΑΦΟΡΟΙ ΤΥΠΟΙ ΠΟΡΩΝ (ΧΡΩΜΑΤΑ, ΣΥΜΒΟΛΟΣΕΙΡΕΣ ΚΕΙΜΕΝΟΥ, ΣΤΥΛ, ΔΙΑΣΤΑΣΕΙΣ, GOOGLE API KEY)



Αυτός ο υποκατάλογος οργανώνει τους διαφορετικούς τύπους πόρων . Στην εφαρμογή μας περιέχονται πόροι χρωμάτων , διαστάσεων , συμβολοσειρών στυλ μορφοποίησης καθώς και ο χάρτης.

6.5.1 Colors.xml

Το αρχείο «*colors.xml*» περιέχει τα ονόματα των χρωμάτων, την δεκαεξαδική τους σημειογραφία (ένας συνδυασμός μπλε χρώματος-μπλε βαθύ χρώματος -μαύρου χρώματος).Ο κώδικας του αρχείου βρίσκεται στο παράρτημα της εργασίας.

6.5.2 Dimens.xml

Το αρχείο «*dimens.xml*» περιλαμβάνει τα ονόματα και τις διαστάσεις που χρησιμοποιούνται στην εφαρμογή. Ο κώδικας του αρχείου βρίσκεται στο παράρτημα της εργασίας.

6.5.3 Google_maps_api.xml

Το αρχείο «*google_maps_api.xml*» είναι απαραίτητο αρχείο για να τρέξει ο χάρτης του Google στην εφαρμογή μας. Το συγκεκριμένο αρχείο περιλαμβάνει ένα μοναδικό κλειδί το οποίο είναι απαραίτητο για την εφαρμογή μας. Ο κώδικας του αρχείου βρίσκεται στο παράρτημα της εργασίας.

6.5.4 Strings.xml

Το αρχείο «*string.xml*» περιλαμβάνει όλες εκείνες τις συμβολοσειρές που χρησιμοποιούνται στο πρόγραμμα μας. Φαίνονται τα ονόματα των συμβολοσειρών, τα ονόματα των πινάκων συμβολοσειρών, οι ίδιες οι συμβολοσειρές, επίσης φαίνεται που γίνεται η χρήση των συμβολοσειρών. Ο κώδικας του αρχείου βρίσκεται στο παράρτημα της εργασίας.

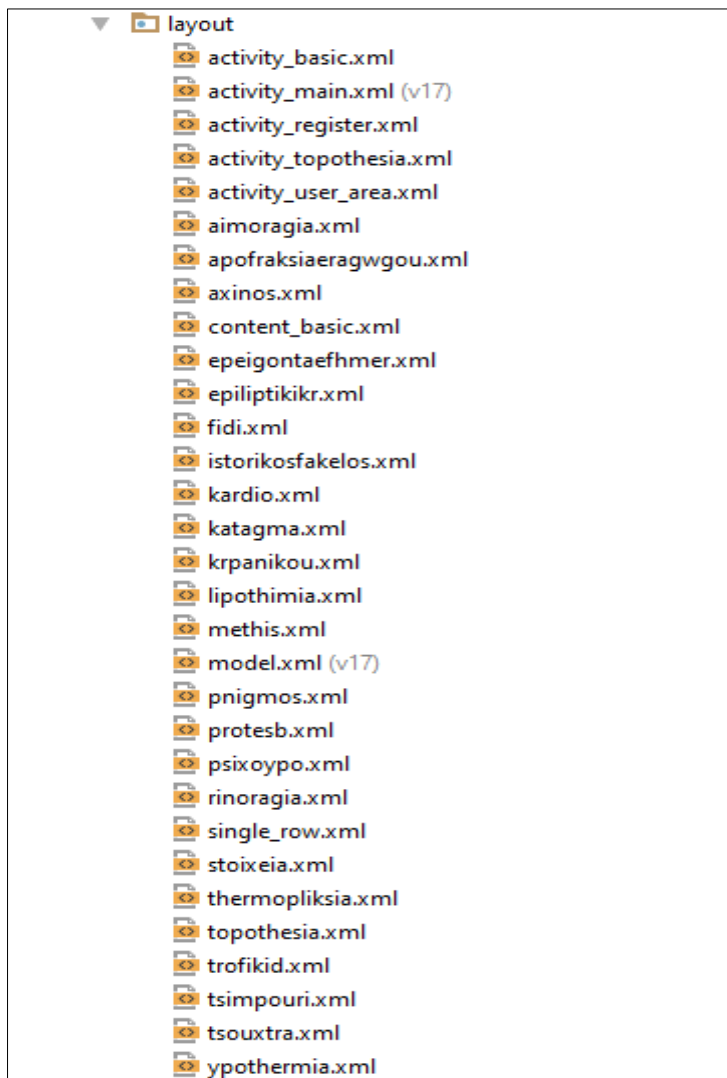
6.5.5 Styles.xml

Το αρχείο «*styles.xml*» περιλαμβάνει όλα τα διαφορετικά στυλ εμφάνισης που χρησιμοποιεί η εφαρμογή μας. Έχουμε προσθέσει αρκετά δικά μας στυλ, που χρησιμοποιούνται στα alter dialogs, στα text views και inputtexts (πχ. Theme.AppCompat.Light.NoActionBar, ThemeOverlay.AppCompat.Dark.ActionBar, ThemeOverlay.AppCompat.Light). Επίσης έχει δημιουργηθεί και ένα απλό στυλ το οποίο χρησιμοποιείται σε όλες τις activities της εφαρμογής προκειμένου να έχουν παρόμοια εμφάνιση (ThemeForActivities – η χρήση της γίνεται στο Android Manifest). Σε κάθε στυλ που δημιουργήθηκε έχουν προστεθεί .xml attributes με την ονομασία του και με την τιμή του χαρακτηριστικού που επιθυμούμε. Ο κώδικας του αρχείου βρίσκεται στο παράρτημα της εργασίας.

6.6 ΥΠΟΦΑΚΕΛΟΣ RES/LAYOUT

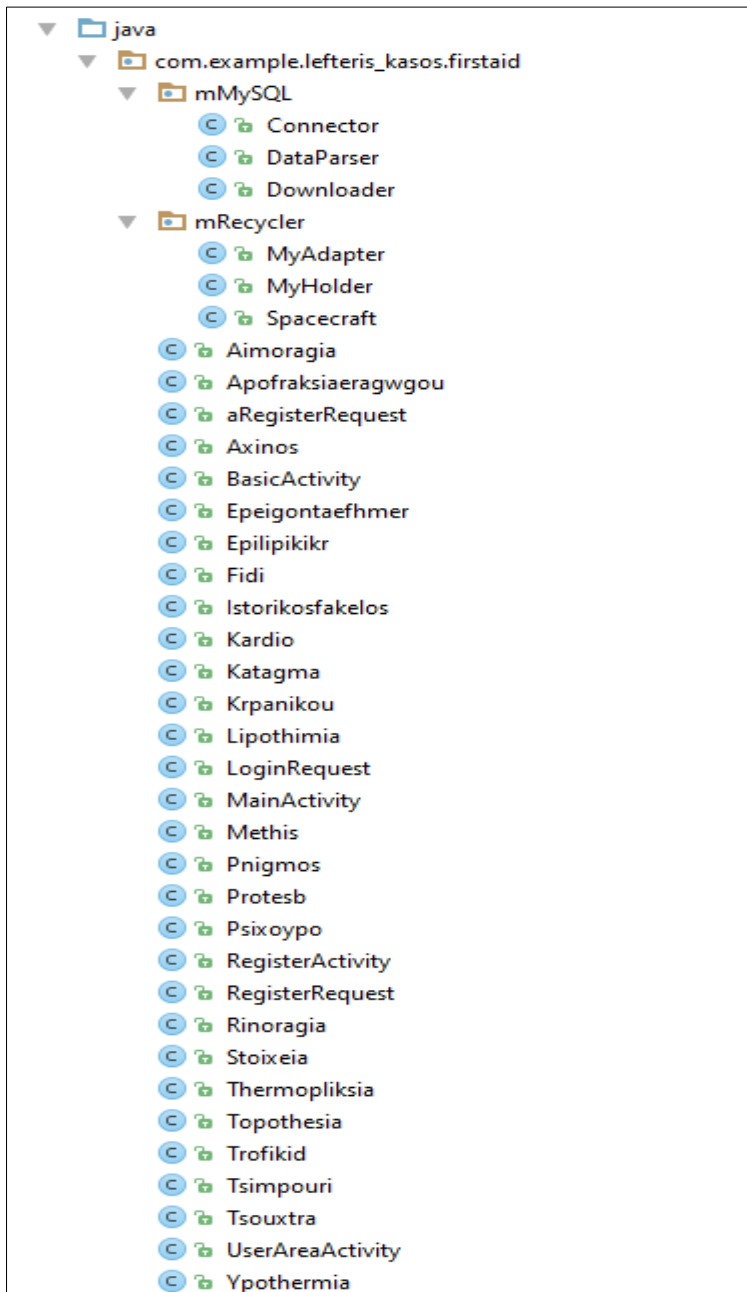
Η χρήση των αρχείων Xml είναι ο ευκολότερος τρόπος για την απεικόνιση γραφικών στοιχείων στην εφαρμογή. Μεταγλωττίζονται σε διατάξεις οθόνης ή μέρους της οθόνης. Αυτός, ο τρόπος, υπερτερεί της κλασσικής δημιουργίας αντικειμένων προγραμματιστικά επειδή διαχωρίζει τον πηγαίο κώδικα που αφορά τις λειτουργίες της εφαρμογής με τον σχεδιασμό της και γίνεται εύκολη η αλλαγή της εμφάνισης της εφαρμογής χωρίς να ασχοληθούμε με πηγαίο κώδικα. Ο σχεδιασμός υλοποιείται ευκολά και γρηγορά

καθώς η διάταξη των αντικειμένων ορίζεται παρόμοια με την φιλοσοφία δημιουργίας ιστοσελίδων με τη γλώσσα Html. Τα αρχεία αυτά αποθηκεύονται στο υποφάκελο res/layout και αποτελούν την γενική μορφοποίηση της εφαρμογής.



6.7 ΦΑΚΕΛΟΣ JAVA

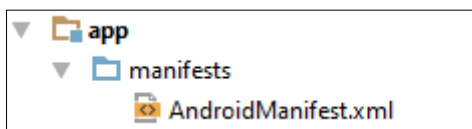
Στο φάκελο Java περιέχεται όλος ο πηγαίος κώδικας και συγκεκριμένα όλα τα αρχεία Java της εφαρμογής. Όλες οι κλάσεις βρίσκονται κάτω από το όνομα του πακέτου. Η εφαρμογή που πραγματεύεται η παρούσα εργασία περιέχει 36 αρχεία Java, τα οποία συμπεριλαμβάνονται στο πακέτο εφαρμογής ονόματι *com.example.lefteris_kasos.firstaid*.



6.8 ΑΡΧΕΙΟ ANDROIDMANIFEST

Το αρχείο AndroidManifest αποτελεί το άλφα και το ωμέγα της εφαρμογής. Σε αυτό ο προγραμματιστής ορίζει τις απαιτήσεις της, δηλώνει τα δικαιώματα που απαιτούνται και καθορίζει ποιες εκδόσεις θα μπορούν να την τρέχουν. Συγκεκριμένα, δηλώνονται ποια χαρακτηριστικά Hardware ή Software θα χρησιμοποιηθούν (Camera, Bluetooth κλπ.), δηλώνονται τα δικαιώματα πρόσβασης που θα έχει η εφαρμογή στα δεδομένα του χρήστη και τέλος ορίζεται η ελάχιστη απαιτούμενη έκδοση.

Υποχρεωτικά, κάθε εφαρμογή πρέπει να έχει αυτό το αρχείο για να μπορεί να γίνει εκτέλεση. Εδώ, δηλώνονται επίσης οι δραστηριότητάς (Activities) μαζί με τις ενέργειες τους. Με την δημιουργία κάθε νέου Project, δηλώνεται αυτόματα η Activity της κεντρικής οθόνης (Main Screen).



6.9 ΦΑΚΕΛΟΣ BIN

Τέλος εδώ βρίσκονται τα αρχεία που παράγονται μετά την μεταγλώττιση του πηγαίου κώδικα. Συγκεκριμένα το τελικό αρχείο .apk το οποίο χρησιμοποιείται για την εγκατάσταση της εφαρμογής στις συσκευές. Αξίζει να σημειωθεί ότι το αρχείο αυτό από μόνο του δεν μπορεί να δημοσιευτεί στο Google Play, γιατί απαιτείται να φέρει ψηφιακή υπογραφή.

ΚΕΦΑΛΑΙΟ 7°

7.1 ΑΝΑΛΥΣΗ ΚΩΔΙΚΑ ΒΟΗΘΗΤΙΚΩΝ ΚΛΑΣΕΩΝ

7.1.1 Βοηθητικές Κλάσεις

Παρακάτω ακολουθεί η παρουσίαση των βοηθητικών κλάσεων της εφαρμογής μας. Βοηθητικές κλάσεις είναι αυτές που καλούνται από τις κύριες κλάσεις μας, για να πραγματοποιήσουν μια ενέργεια, μια δράση. Οι ενέργειες αυτές και κατά συνέπεια και ο κώδικας των βοηθητικών κλάσεων θα μπορούσε να έχει συγχωνευτεί με αυτόν των κυρίων κλάσεων, αλλά είναι καλύτερη πρακτική ο διαμερισμός του κώδικα και η χρήση με κλάσεις στο σημείο που επιθυμούμε, ιδίως και αφού η χρήση του κώδικα των βοηθητικών κλάσεων επαναλαμβάνεται.

Οι βοηθητικές μας κλάσεις δεν έχουν κάποια μορφή, απεικόνιση στο User Interface του κινητού. Είναι ενέργειες που συμπληρώνουν τα κυρία κομμάτια κώδικα. Κατά συνέπεια δεν σχετίζονται με κάποιο Layout αρχείο, για αυτό και παρουσιάζονται μονές τους στο κομμάτι αυτό.

7.1.2 mMySQL

Στο φάκελο mMySQL παρατηρούμε ότι έχουμε τις βοηθητικές κλάσεις Connector, DataParser και Downloader, όπου αυτό μας βοηθάει να δημιουργήσουμε ένα δικό μας λογαριασμό και να πραγματοποιηθεί η σύνδεση με το Server και με την βάση δεδομένων που έχουμε δημιουργήσει μέσω της ιστοσελίδας www.hostinger.gr

7.2 ΑΝΑΛΥΣΗ ΚΩΔΙΚΑ ΚΥΡΙΩΝ ΚΛΑΣΕΩΝ

Παρακάτω ακολουθεί η ανάλυση του κώδικα των κυρίων κλάσεων της εφαρμογής μας. Ο κώδικας αυτός είναι ουσιαστικός για την εφαρμογή μας, είναι η ραχοκοκαλιά της, σχετίζεται με το τι βλέπει και το τι κάνει ο χρήστης στην εφαρμογή, αποτυπώνει τις διάφορες οθόνες μέσω των **Layout** που χρησιμοποιούνται από την καθεμιά κλάση. Έτσι δεν μπορεί να παρουσιαστεί η κάθε κλάση ξεχωριστά από τα **Layouts.xml** αρχεία που σχετίζεται. Αξίζει να σημειωθεί ότι κάθε ξεχωριστή κυρία οθόνη της εφαρμογής μας σχετίζεται με μια κυρία κλάση/ δραστηριότητα της εφαρμογής μας.

7.2.1 Android Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.lefteris_kasos.firstaid">

    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.CALL_PHONE" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />

    <!--
        The ACCESS_COARSE/FINE LOCATION permissions are not required to use
        Google Maps Android API v2, but you must specify either coarse or fine
        location permissions for the 'MyLocation' functionality.
    -->
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

Ορίζουμε την έκδοση της εφαρμογής και το όνομα της έκδοσης της εφαρμογής, καθώς επίσης και τα δικαιώματα της εφαρμογής, παραδείγματος χάρι το δικαίωμα η εφαρμογή να καλεί στο τηλέφωνο, ή να έχει πρόσβαση στο διαδίκτυο.

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="First Aid"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".MainActivity">
```

Εδώ αναφέρουμε το θέμα της εφαρμογής μας **@style/AppTheme** , το όνομα της εφαρμογής **First Aid**.

```

</activity>
<activity android:name=".Psixoypo" />
<activity android:name=".Protesb" />
<activity android:name=".Epeigontaeafhmer" />
<activity
  android:name=".Topothesisia"
  android:label="Map" />
<activity android:name=".Istorikosfakelos" />
<activity android:name=".Stoixeia" />
<activity android:name=".Kardio" />
<activity android:name=".Apofraksiaeragwgou" />
<activity android:name=".Pnigmos" />
<activity android:name=".Aimoragia" />
<activity android:name=".Rinoragia" />
<activity android:name=".Trofikid" />
<activity android:name=".Katagma" />
<activity android:name=".Thermopliksia" />
<activity android:name=".Ypothemia" />
<activity android:name=".Lipothimia" />
<activity android:name=".Axinos" />
<activity android:name=".Fidi" />
<activity android:name=".Tsimpouri" />
<activity android:name=".Tsouextra" />
<activity android:name=".Methis" />
<activity android:name=".Kspanikou" />
<activity android:name=".Epilipikikr" />

```

Παραπάνω αποτυπώνονται όλες οι δραστηριότητες της εφαρμογής μας.

```

<meta-data
  android:name="com.google.android.gms.version"
  android:value="9256000" />

<activity android:name=".RegisterActivity" />
<activity android:name=".UserAreaActivity" />
<activity
  android:name=".BasicActivity"
  android:label="BasicActivity"
  android:theme="@style/AppTheme" />

<!--
  The API key for Google Maps-based APIs is defined as a string resource.
  (See the file "res/values/google_maps_api.xml").
  Note that the API key is linked to the encryption key used to sign the APK.
  You need a different API key for each encryption key, including the release key that is used to
  sign the APK for publishing.
  You can define the keys for the debug and release targets in src/debug/ and src/release/.
-->
<meta-data
  android:name="com.google.android.geo.API_KEY"
  android:value="AIzaSyDLwQvAAv10gwzvUKYJQsLs7O-Lth4Tkug" />
</application>

```

Αποτυπώνονται οι δραστηριότητες που έχουν σχέση με την εγγραφή του χρήστη στο Ιατρικό φάκελο καθώς επίσης, και το κλειδί της google που έχει σχέση με τον χάρτη.

7.2.2 ActivityMain.xml

Η κλάση MainActivity.java αντιπροσωπεύει μια δραστηριότητα στην εφαρμογή μας. Ουσιαστικά είναι η πρώτη δραστηριότητα και η πρώτη οθόνη που βλέπουμε. Η κλάση αυτή χρησιμοποιεί το αρχείο ActivityMain.xml για να σχηματιστεί η οθόνη της δραστηριότητας. Η Κλάση αυτή πέρα από την υλοποίηση του μενού περιβάλλοντος και του μενού επίλογων , που χρησιμοποιούνται για να μετάβουμε σε άλλες δραστηριότητες -οθόνες, η κλάση αυτή έχει και άλλη χρήσιμη λειτουργία. Όταν εγκαταστήσουμε για πρώτη φορά την εφαρμογή μας και την τρέξουμε, όταν αυτή φτάσει στο μενού για πρώτη φορά δημιουργείτε η βάση μας.

ΚΕΦΑΛΑΙΟ 8°

8.1 ΣΥΜΠΕΡΑΣΜΑΤΑ

Ολοκληρώνοντας την διπλωματική μου εργασία και το «ταξίδι» στην ανάπτυξη Android εφαρμογής αποκτήθηκαν χρήσιμα συμπεράσματα και γνώσεις. Πέρα από τις απαιτήσεις της πλατφόρμας και την εξερεύνηση δυσκολιών της, η ενασχόληση με το αντικείμενο προσφέρει την κατανόηση της γλώσσας προγραμματισμού Java και γενικά τον λογισμό που χρειάζεται από την σχεδίαση έως την υλοποίηση ενός έργου. Επίσης, αποκτήθηκαν γνώσεις πάνω στην πλατφόρμα Android Studio και στα εργαλεία που προσφέρει.

Τα κυρία πλεονεκτήματα όμως είναι η ικανοποίηση και η ευχαρίστηση του προγραμματιστή για την υλοποίηση ενός χρήσιμου εργαλείου που εξυπηρετεί τους συνάνθρωπους μας σε μια έκτακτη ανάγκη και επίσης δίνεται ένα νέο παράδειγμα στους νέους προγραμματιστές.

8.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Η εφαρμογή που πραγματεύεται η παρούσα διπλωματική εργασία είναι ολοκληρωμένη και λειτουργική. Έχει απλή δομή για βέλτιστη απόδοση χρόνου και προσφέρει όλες τις απαραίτητες λειτουργίες που αφορούν τις Πρώτες Βοήθειες. Η συνεχόμενη αύξηση των υπηρεσιών που προσφέρει το Android στους χρήστες δεν επιτρέπει σε καμία εφαρμογή παραμένει στάσιμη στο χρόνο. Έτσι, και η συγκριμένη εφαρμογή μπορεί να εξελιχθεί και να προσφέρει περισσότερες δυνατότητες που δεν προστέθηκαν είτε λόγω της λήξης προθεσμίας της εργασίας είτε λόγω των ελλείπων τεχνολογιών την δεδομένη στιγμή. Τέτοιες δυνατότητες θα μπορούσε να ήταν, να περιέχει ένα κουμπί Έκτακτης Ανάγκης (Panic Button) το οποίο να έστελνε αυτοματοποιημένο μήνυμα μαζί με την τοποθεσία του χρήστη εκείνη την στιγμή αυτόματα σε κάποιον αριθμό τον οποίο θα δηλώναμε στην αρχή όταν κατεβάζαμε την εφαρμογή, θα μπορούσε στις Πρώτες Βοήθειες αντί για εικόνες να περιέχει βίντεο που να εξηγούσε το κάθε βήμα χωριστά.

ΚΕΦΑΛΑΙΟ 9°

9.1 ΒΙΒΛΙΟΓΡΑΦΙΑ

9.1.1 ΕΛΛΗΝΙΚΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Lauren Darcey , Shane Conder , «Μάθετε την Ανάπτυξη Εφαρμογών για το Android σε 24 Ωρες» , Εκδόσεις Μ.Γκιούρδας , Αθήνα 2012

[2] Paul Deitel, Harvey Deitel , Abbey Deitel «Android για Προγραμματιστές, Μάθετε μέσα από τις Εφαρμογές», Εκδόσεις Μ. Γκιούρδας, Αθήνα 2012

[3] Conder Shane, «Ανάπτυξη Εφαρμογών με το Android», Εκδόσεις Μ. Γκιούρδας, Αθήνα 2011

[4] Rogers Cadenhead , Laura Lemay , «Πλήρες Εγχειρίδιο της Java 2 (τρίτη έκδοση)» , Εκδόσεις Μ.Γκιούρδας , Αθήνα 2003

9.1.2 ΞΕΝΟΓΛΩΣΣΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Wei-Meng Lee 2012. Beginning Android 4 Application Development. 1st Edition, John Wiley& Sons, USA

[2] Paul Deitel, Harvey Deitel, Abbey Deitel, Michael Morgano 2011. Android for Programmers, An App-Driven Approach. 1st Edition, Prentice Hall, USA.

[3] Erik T. Ray 2003. Learning XML. 2nd Edition, O'Reilly, USA.

9.1.3 ΗΛΕΚΤΡΟΝΙΚΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

- 1) <http://lyk-avlon.att.sch.gr/wp-content/uploads/ceb7-ceb5cebeceb5cebbceb9cebeceb7-cf84cf89cebd-cebaceb9cebdceb7cf84cf89cebd-cf84ceb7cebbceb5cf86cf89cebdcf89cebd.pdf>
- 2) 2epal-aigal.att.sch.gr/images/yliko/projects/paroyusiasikinita4epal97g.ppt
- 3) http://5lyk-koryd.att.sch.gr/wp-content/uploads/Istoria_Kinhtwn.pdf
- 4) <https://dsewiki.wikispaces.com/4%CE%B7+%CE%B3%CE%B5%CE%BD%CE%AF%CE%B1+%CE%BA%CE%B9%CE%BD%CE%B7%CF%84%CF%8E%CE%BD+%CF%84%CE%B7%CE%BB%CE%B5%CF%86%CF%8E%CE%BD%CF%89%CE%BD>
- 5) <http://smartphoneproject1.weebly.com/epsilonxiomegatauepsilonrhoiotakappaalphanialpharhoalphakappatauetarhoiotasigmatauiotakappaalpha.html>
- 6) <http://www.statista.com/statistics/272307/market-share-forecast-for-smartphone-operating-systems/>
- 7) <http://www.kumulos.com/2013/05/08/kantar-backend-as-a-service>
- 8) <https://el.wikipedia.org/wiki/Android>
- 9) <http://unboxholics.com/stories/10064-ta-panta-gia-to-android-6-marshmallow>

- 10) <http://www.myphone.gr/forum/showthread.php?t=440335>
- 11) <https://iguru.gr/2015/03/10/45402/android-5-1-lollipop-45402/>
- 12) www.tutorialspoint.com/android/android_studio.htm
- 13) https://el.wikipedia.org/wiki/Android_Studio
- 14) <https://developer.android.com/reference/android/app/Activity.html>
- 15) <https://developer.android.com/guide/topics/manifest/manifest-intro.html>
- 16) <https://developer.android.com/guide/topics/ui/declaring-layout.html#CommonLayouts>
- 17) <https://developer.android.com/guide/topics/ui/custom-components.html>
- 18) https://developer.android.com/guide/practices/screens_support.html
- 19) https://developer.android.com/guide/practices/screens_support.html
- 20) <https://developer.android.com/guide/topics/sensors/index.html>
- 21) <http://www.techinsider.gr/software/mobile-os/pies-ine-i-diafores-metaxi-android-4-0-ke-android-2-3/>
- 22) https://en.wikipedia.org/wiki/Google_Maps
- 23) <http://www.myphone.gr/forum/showthread.php?t=306146>.
- 24) <https://dzone.com/>
- 25) www.hostinger.gr

ΚΕΦΑΛΑΙΟ 10°

10.1 ΠΑΡΑΡΤΗΜΑ-ΚΩΔΙΚΕΣ

10.1.1 colors.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<resources>
  <color name="colorPrimary">#1d319a</color>
  <color name="colorPrimaryDark">#303f9f</color>
  <color name="colorAccent">#000000</color>
</resources>
```

10.1.2 dimens.xml

```
<resources>
  <!-- Default screen margins, per the Android Design guidelines. -->
  <dimen name="activity_horizontal_margin">16dp</dimen>
  <dimen name="activity_vertical_margin">16dp</dimen>
  <dimen name="fab_margin">16dp</dimen>
</resources>
```

10.1.3 google_maps_api.xml

```
<resources>
  <!--
  TODO: Before you run your application, you need a Google Maps API key.

  To get one, follow this link, follow the directions and press "Create" at the end:
  https://console.developers.google.com/flows/enableapi?apiid=maps_android_backend&keyType=CLIENT_SIDE_ANDROID&r=93:3E:88:00:F6:8E:D5:9

  You can also add your credentials to an existing key, using this line:
  93:3E:88:00:F6:8E:D5:97:7E:EB:CD:6E:21:BB:26:52:37:3E:B1:75;com.example.lefteris_kasos.firstaid

  Alternatively, follow the directions here:
  https://developers.google.com/maps/documentation/android/start#get-key

  Once you have your key (it starts with "AIza"), replace the "google_maps_key"
  string in this file.
  -->
  <string name="google_maps_key" templateMergeStrategy="preserve" translatable="false">AIzaSyDLwQvAAv10gwzyUKYJQsLs70-Lth4Tkug</string>
</resources>
```


10.1.4 strings.xml

```
<resources>
  <string name="app_name">First Aid</string>
  <string name="name">Name:</string>
  <string name="amp"><![CDATA[Επείγοντα & Εφημερεύοντα]]></string>
  <string name="list">LIST</string>

  <string-array name="titles">
    <item>ΚΑΡΔΙΟΑΝΑΠΝΕΥΣΤΙΚΗ ΑΝΑΖΩΟΓΟΝΗΣΗ</item>
    <item>ΑΠΟΦΡΑΞΗ ΑΕΡΑΓΩΓΟΥ</item>
    <item>ΠΝΙΓΜΟΣ-ΕΞΑΓΩΓΗ ΞΕΝΟΥ</item>
    <item>ΑΙΜΟΡΡΑΓΙΑ</item>
    <item>ΡΙΝΟΡΡΑΓΙΑ</item>
    <item>ΤΡΟΦΙΚΗ ΔΗΛΗΤΗΡΙΑΣΗ</item>
    <item>ΚΑΤΑΓΜΑ</item>
    <item>ΘΕΡΜΟΠΛΗΣΙΑ</item>
    <item>ΥΠΟΘΕΡΜΙΑ</item>
    <item>ΛΙΠΟΘΥΜΙΑ</item>
    <item>ΤΣΙΜΠΗΜΑ ΑΠΟ ΑΧΙΝΟ</item>
    <item>ΤΣΙΜΠΗΜΑ ΑΠΟ ΦΙΔΙ</item>
    <item>ΤΣΙΜΠΗΜΑ ΑΠΟ ΤΣΙΜΠΟΥΡΙ</item>
    <item>ΤΣΙΜΠΗΜΑ ΑΠΟ ΤΣΟΥΧΤΡΑ</item>
    <item>ΑΝΤΙΜΕΤΩΠΙΣΗ ΜΕΘΗΣ</item>
    <item>ΚΡΙΣΗ ΠΑΝΙΚΟΥ</item>
    <item>ΕΠΙΛΗΠΤΙΚΗ ΚΡΙΣΗ</item>
  </string-array>

  <string-array name="descriptions">
    <item>lefteris</item>
    <item>lefteris</item>
    <item>lefteris</item>
    <item>lefteris</item>
    <item>lefteris</item>
    <item>lefteris</item>
    <item>lefteris</item>
    <item>lefteris</item>
    <item>lefteris</item>
    <item>lefteris</item>
    <item>lefteris</item>
    <item>lefteris</item>
    <item>lefteris</item>
    <item>lefteris</item>
    <item>lefteris</item>
    <item>lefteris</item>
    <item>lefteris</item>
    <item>lefteris</item>
    <item>lefteris</item>
    <item>lefteris</item>
    <item>lefteris</item>
    <item>lefteris</item>
    <item>lefteris</item>
  </string-array>
  <string name="title_activity_basic">BasicActivity</string>
  <string name="action_settings">Settings</string>
  <string name="title_activity_topothesia">Map</string>
</resources>
```

10.1.5 styles.xml

```

<resources>

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">
        <!-- Customize your theme here. -->
    </style>

    <style name="AppTheme.AppBarOverlay" parent="ThemeOverlay.AppCompat.Dark.ActionBar" />

    <style name="AppTheme.PopupOverlay" parent="ThemeOverlay.AppCompat.Light" />

</resources>

```

10.1.6 Aimoragia.java

```

package com.example.lefteris_kasos.firstaid;

import android.app.Activity;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

/**
 * Created by Lefteris_Kasos on 26/06/2016.
 */
public class Aimoragia extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.aimoragia);
        Button play = (Button) findViewById(R.id.play);
        Button pause = (Button) findViewById(R.id.pause);
        Button stop = (Button) findViewById(R.id.stop);

        final MediaPlayer mp = MediaPlayer.create(Aimoragia.this, R.raw.aimorragia);
        play.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.start();
            }
        });

        pause.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.pause();
            }
        });

        stop.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.stop();
                Aimoragia.this.finish();
            }
        });
    }
}

```

10.1.7 Apofraksiaeragwou.java

```
package com.example.lefteris_kasos.firstaid;

import ...

/**
 * Created by Lefteris Kasos on 26/06/2016.
 */
public class Apofraksiaeragwou extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.apofraksiaeragwou);

        Button play = (Button) findViewById(R.id.play);
        Button pause = (Button) findViewById(R.id.pause);
        Button stop = (Button) findViewById(R.id.stop);

        final MediaPlayer mp = MediaPlayer.create(Apofraksiaeragwou.this, R.raw.apofraksiaeragwou);
        play.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.start();
            }
        });

        pause.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.pause();
            }
        });

        stop.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.stop();
                Apofraksiaeragwou.this.finish();
            }
        });
    }
}
```

10.1.8 aRegisterRequest.java

```

package com.example.lefteris_kasos.firstaid;

import com.android.volley.Response;
import com.android.volley.toolbox.StringRequest;

import java.util.HashMap;
import java.util.Map;

/**
 * Created by Lefteris_Kasos on 01/07/2016.
 */
public class aRegisterRequest extends StringRequest {

    private static final String REGISTER_REQUEST_URL = "http://firstaid.esy.es/aRegister.php";
    private Map<String, String> params;

    public aRegisterRequest(String onoma, String eponymo, String onomamitros,
        String onomapatros, String fulo, String hmerominiagenisis,
        String hlikia, String xwratagwghs, String dieythinsi,
        String taxydromikoskwdikas, String perioxh, String polh,
        String xwra, String thlefwno, String kinhto, String arithmostaytothtas,
        String afm, String amka, String eksetaseiseginan, String khlh,
        String aimoroides, String ourhsh, String sakxarwdhdiavitis,
        String petres, String arthritida, String problmataostwn,
        String apoleiadakylwn, String ponoiplaths, String problmatagonatwn,
        String problmatadiatrofhs, String nautia, String problmataypnou,
        String katathlipsi, String apwleiamnhmhs, String neurologikaproblmata,
        String apwleiasynidisis, String neuritida, String paralysh, String epilipsia,
        String gynaikologikaproblmata, String emhnorysh, String eksetaseisprogrammatismenes,
        String apotelesmata, Response.Listener<String>listener) {
        super(Method.POST, REGISTER_REQUEST_URL, listener, null);
        params = new HashMap<>();
        params.put("onoma", onoma);
        params.put("eponymo", eponymo);
        params.put("onomamitros", onomamitros);
        params.put("onomapatros", onomapatros);
        params.put("fulo", fulo);
        params.put("hmerominiagenisis", hmerominiagenisis);
        params.put("hlikia", hlikia);
        params.put("xwratagwghs", xwratagwghs);
        params.put("dieythinsi", dieythinsi);
        params.put("taxydromikoskwdikas", taxydromikoskwdikas);
        params.put("perioxh", perioxh);
        params.put("polh", polh);
        params.put("xwra", xwra);
        params.put("thlefwno", thlefwno);
        params.put("kinhto", kinhto);
        params.put("arithmostaytothtas", arithmostaytothtas);
        params.put("afm", afm);
        params.put("amka", amka);
        params.put("eksetaseiseginan", eksetaseiseginan);
        params.put("khlh", khlh);
        params.put("aimoroides", aimoroides);
        params.put("ourhsh", ourhsh);
        params.put("sakxarwdhdiavitis", sakxarwdhdiavitis);
        params.put("petres", petres);
        params.put("arthritida", arthritida);
        params.put("problmataostwn", problmataostwn);
    }
}

```

```

        params.put("apoleiadakylwn", apoleiadakylwn);
        params.put("ponoiplaths", ponoiplaths);
        params.put("problhmatagonatwn", problhmatagonatwn);
        params.put("problhmatadiatrofhs", problhmatadiatrofhs);
        params.put("nautia", nautia);
        params.put("problhmataypnou", problhmataypnou);
        params.put("katathlipsi", katathlipsi);
        params.put("apwleiamnhmhs", apwleiamnhmhs);
        params.put("neurologikaproblhmata", neurologikaproblhmata);
        params.put("apwleiasynidisis", apwleiasynidisis);
        params.put("neuritida", neuritida);
        params.put("paralysh", paralysh);
        params.put("epilipsia", epilipsia);
        params.put("gynaikologikaproblhmata", gynaikologikaproblhmata);
        params.put("emhnrorysh", emhnrorysh);
        params.put("eksetaseisprogrammatismenes", eksetaseisprogrammatismenes);
        params.put("apotelesmata", apotelesmata);
    }

    @Override
    public Map<String, String> getParams() { return params; }
}

```

10.1.9 Axinos.java

```

package com.example.lefteris_kasos.firstaid;

import android.app.Activity;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

/**
 * Created by Lefteris_Kasos on 26/06/2016.
 */
public class Axinos extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.axinos);

        Button play = (Button) findViewById(R.id.play);
        Button pause = (Button) findViewById(R.id.pause);
        Button stop = (Button) findViewById(R.id.stop);

        final MediaPlayer mp = MediaPlayer.create(Axinos.this, R.raw.axinos);
        play.setOnClickListener((view) - { mp.start(); });

        pause.setOnClickListener((view) - { mp.pause(); });

        stop.setOnClickListener((view) - {
            mp.stop();
            Axinos.this.finish();
        });
    }
}

```

10.1.10 BasicActivity.java

```
package com.example.lefteris_kasos.firstaid;

import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.DefaultItemAnimator;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.view.View;
import android.view.Window;

import com.example.lefteris_kasos.firstaid.mMySQL.Downloader;

public class BasicActivity extends AppCompatActivity {

    String urlAddress="http://firstaid.esy.es/nRegister.php";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate (savedInstanceState);
        this.requestWindowFeature (Window.FEATURE_NO_TITLE);
        setContentView (R.layout.activity_basic);

        final RecyclerView rv= (RecyclerView) findViewById(R.id.rv);
        rv.setLayoutManager(new LinearLayoutManager(this));
        rv.setItemAnimator(new DefaultItemAnimator());

        FloatingActionButton fab = (FloatingActionButton) findViewById (R.id.fab);
        fab.setOnClickListener ((view) - {

            new Downloader (BasicActivity.this,urlAddress,rv).execute ();

        });
    }
}
```

10.1.11 Ephgontaefhmer.java

```
public class Epeigontaefhmer extends Activity implements View.OnClickListener {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.epeigontaefhmer);  
  
        Button button0 = (Button) this.findViewById(R.id.button0);  
        Button button5 = (Button) this.findViewById(R.id.button5);  
        Button button6 = (Button) this.findViewById(R.id.button6);  
        Button button7 = (Button) this.findViewById(R.id.button7);  
        Button button8 = (Button) this.findViewById(R.id.button8);  
  
        button0.setOnClickListener(this);  
        button5.setOnClickListener(this);  
        button6.setOnClickListener(this);  
        button7.setOnClickListener(this);  
        button8.setOnClickListener(this);  
    }  
  
    @Override  
    public void onClick(View v) {  
        switch (v.getId()) {  
            case R.id.button0:  
                Intent d = new Intent();  
                d.setAction(Intent.ACTION_DIAL);  
                d.setData(Uri.parse("tel:" + Uri.encode("166")));  
                startActivity(d);  
                break;  
            case R.id.button5:  
                Intent f = new Intent();  
                f.setAction(Intent.ACTION_DIAL);  
                f.setData(Uri.parse("tel:" + Uri.encode("2107793777")));  
                startActivity(f);  
                break;  
            case R.id.button6:  
                Intent g = new Intent();  
                g.setAction(Intent.ACTION_DIAL);  
                g.setData(Uri.parse("tel:" + Uri.encode("14944")));  
                startActivity(g);  
                break;  
            case R.id.button7:  
                Intent h = new Intent();  
                h.setAction(Intent.ACTION_DIAL);  
                h.setData(Uri.parse("tel:" + Uri.encode("100")));  
                startActivity(h);  
                break;  
            case R.id.button8:  
                Intent k = new Intent();  
                k.setAction(Intent.ACTION_DIAL);  
                k.setData(Uri.parse("tel:" + Uri.encode("199")));  
                startActivity(k);  
                break;  
        }  
    }  
};
```

10.1.12 Epilipikikr.java

```
package com.example.lefteris_kasos.firstaid;

import android.app.Activity;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

/**
 * Created by Lefteris_Kasos on 26/06/2016.
 */
public class Epilipikikr extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.epiliptikikr);

        Button play = (Button) findViewById(R.id.play);
        Button pause = (Button) findViewById(R.id.pause);
        Button stop = (Button) findViewById(R.id.stop);

        final MediaPlayer mp = MediaPlayer.create(Epilipikikr.this, R.raw.epiliptikikrisi);
        play.setOnClickListener((view) -> { mp.start(); });

        pause.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.pause();
            }
        });

        stop.setOnClickListener((view) -> {
            mp.stop();
            Epilipikikr.this.finish();
        });
    }
}
```


10.1.13 Fidi.java

```
package com.example.lefteris_kasos.firstaid;

import ...

/**
 * Created by Lefteris_Kasos on 26/06/2016.
 */
public class Fidi extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.fidi);

        Button play = (Button) findViewById(R.id.play);
        Button pause = (Button) findViewById(R.id.pause);
        Button stop = (Button) findViewById(R.id.stop);

        final MediaPlayer mp = MediaPlayer.create(Fidi.this, R.raw.fidi);
        play.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.start();
            }
        });

        pause.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.pause();
            }
        });

        stop.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.stop();
                Fidi.this.finish();
            }
        });
    }
}
```

10.1.14 Istorikosfakelos.java

```

package com.example.lefteris_kasos.firstaid;

import ...

/**
 * Created by Lefteris_Kasos on 03/06/2016.
 */
public class Istorikosfakelos extends AppCompatActivity {

    /**
     * ATTENTION: This was auto-generated to implement the App Indexing API.
     * See https://g.co/AppIndexing/AndroidStudio for more information.
     */
    private GoogleApiClient client;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.istorikosfakelos);

        final EditText etUsername = (EditText) findViewById(R.id.etUsername);
        final EditText etPassword = (EditText) findViewById(R.id.etPassword);
        final Button bLogin = (Button) findViewById(R.id.bLogin);
        final TextView registerLink = (TextView) findViewById(R.id.tvRegisterHere);

        registerLink.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {

                Intent registerIntent = new Intent(Istorikosfakelos.this, RegisterActivity.class);
                Istorikosfakelos.this.startActivity(registerIntent);

            }

        });

        bLogin.setOnClickListener(new View.OnClickListener() {
            @Override

            public void onClick(View v) {

                final String username = etUsername.getText().toString();
                final String password = etPassword.getText().toString();

                Response.Listener<String> responseListener = new Response.Listener<String>() {

```

```
@Override
public void onClick(View v) {

    Intent registerIntent = new Intent(Istorikosfakelos.this, RegisterActivity.class);
    Istorikosfakelos.this.startActivity(registerIntent);

}

});

bLogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        final String username = etUsername.getText().toString();
        final String password = etPassword.getText().toString();

        Response.Listener<String> responseListener = new Response.Listener<String>() {

            @Override
            public void onResponse(String response) {

                try {
                    JSONObject jsonResponse = new JSONObject(response);
                    boolean success = jsonResponse.getBoolean("success");

                    if (success) {

                        String name = jsonResponse.getString("name");
                        int age = jsonResponse.getInt("age");

                        Intent intent = new Intent(Istorikosfakelos.this, UserAreaActivity.class);
                        intent.putExtra("name", name);
                        intent.putExtra("username", username);
                        intent.putExtra("age", age);

                        Istorikosfakelos.this.startActivity(intent);

                    } else {
                        AlertDialog.Builder builder = new AlertDialog.Builder(Istorikosfakelos.this);
                        builder.setMessage("Η ΕΙΣΟΔΟΣ ΑΝΕΤΥΧΕ")
                                .setNegativeButton("ΠΡΟΣΠΑΘΗΤΕ ΞΑΝΑ", null)
                                .create()
                                .show();
                    }

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

            }

        };

    }

});
```

```
        LoginRequest loginRequest = new LoginRequest(username, password, responseListener);
        RequestQueue queue = Volley.newRequestQueue(Istorikosfakelos.this);
        queue.add(loginRequest);
    }
}

// ATTENTION: This was auto-generated to implement the App Indexing API.
// See https://g.co/AppIndexing/AndroidStudio for more information.
client = new GoogleApiClient.Builder(this).addApi(AppIndex.API).build();
}

@Override
public void onStart() {
    super.onStart();

    // ATTENTION: This was auto-generated to implement the App Indexing API.
    // See https://g.co/AppIndexing/AndroidStudio for more information.
    client.connect();
    Action viewAction = Action.newAction(
        Action.TYPE_VIEW, // TODO: choose an action type.
        "Istorikosfakelos Page", // TODO: Define a title for the content shown.
        // TODO: If you have web page content that matches this app activity's content,
        // make sure this auto-generated web page URL is correct.
        // Otherwise, set the URL to null.
        Uri.parse("http://host/path"),
        // TODO: Make sure this auto-generated app URL is correct.
        Uri.parse("android-app://com.example.lefteris_kasos.firstaid/http/host/path")
    );
    AppIndex.AppIndexApi.start(client, viewAction);
}

@Override
public void onStop() {
    super.onStop();

    // ATTENTION: This was auto-generated to implement the App Indexing API.
    // See https://g.co/AppIndexing/AndroidStudio for more information.
    Action viewAction = Action.newAction(
        Action.TYPE_VIEW, // TODO: choose an action type.
        "Istorikosfakelos Page", // TODO: Define a title for the content shown.
        // TODO: If you have web page content that matches this app activity's content,
        // make sure this auto-generated web page URL is correct.
        // Otherwise, set the URL to null.
        Uri.parse("http://host/path"),
        // TODO: Make sure this auto-generated app URL is correct.
        Uri.parse("android-app://com.example.lefteris_kasos.firstaid/http/host/path")
    );
    AppIndex.AppIndexApi.end(client, viewAction);
    client.disconnect();
}
}
```

10.1.15 Kardio.java

```
package com.example.lefteris_kasos.firstaid;

import android.app.Activity;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

/**
 * Created by Lefteris_Kasos on 26/06/2016.
 */
public class Kardio extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.kardio);

        Button play = (Button) findViewById(R.id.play);
        Button pause = (Button) findViewById(R.id.pause);
        Button stop = (Button) findViewById(R.id.stop);

        final MediaPlayer mp = MediaPlayer.create(Kardio.this, R.raw.kardio);
        play.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.start();
            }
        });

        pause.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.pause();
            }
        });

        stop.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.stop();
                Kardio.this.finish();
            }
        });
    }
}
```

10.1.16 Katagma.java

```
package com.example.lefteris_kasos.firstaid;

import android.app.Activity;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

/**
 * Created by Lefteris Kasos on 26/06/2016.
 */
public class Katagma extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.katagma);

        Button play = (Button) findViewById(R.id.play);
        Button pause = (Button) findViewById(R.id.pause);
        Button stop = (Button) findViewById(R.id.stop);

        final MediaPlayer mp = MediaPlayer.create(Katagma.this, R.raw.katagma);
        play.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.start();
            }
        });

        pause.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.pause();
            }
        });

        stop.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.stop();
                Katagma.this.finish();
            }
        });
    }
}
```

10.1.17 Krpanikou.java

```
package com.example.lefteris_kasos.firstaid;

import android.app.Activity;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

/**
 * Created by Lefteris Kasos on 26/06/2016.
 */
public class Krpanikou extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.krpanikou);

        Button play = (Button) findViewById(R.id.play);
        Button pause = (Button) findViewById(R.id.pause);
        Button stop = (Button) findViewById(R.id.stop);

        final MediaPlayer mp = MediaPlayer.create(Krpanikou.this, R.raw.krisipanikou);
        play.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.start();
            }
        });

        pause.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.pause();
            }
        });

        stop.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.stop();
                Krpanikou.this.finish();
            }
        });
    }
}
```

10.1.18 Lipothimia.java

```
package com.example.lefteris_kasos.firstaid;

import ...

/**
 * Created by Lefteris Kasos on 26/06/2016.
 */
public class Lipothimia extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.Lipothimia);

        Button play = (Button) findViewById(R.id.play);
        Button pause = (Button) findViewById(R.id.pause);
        Button stop = (Button) findViewById(R.id.stop);

        final MediaPlayer mp = MediaPlayer.create(Lipothimia.this, R.raw.Lipothimia);
        play.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.start();
            }
        });

        pause.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.pause();
            }
        });

        stop.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.stop();
                Lipothimia.this.finish();
            }
        });
    }
}
```


10.1.19 LoginRequest.java

```
package com.example.lefteris_kasos.firstaid;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.toolbox.StringRequest;

import java.util.HashMap;
import java.util.Map;

/**
 * Created by Lefteris Kasos on 30/06/2016.
 */
public class LoginRequest extends StringRequest {

    private static final String LOGIN_REQUEST_URL = "http://firstaid.esy.es/Login.php";
    private Map<String, String> params;

    public LoginRequest( String username, String password, Response.Listener<String>listener) {
        super(Request.Method.POST, LOGIN_REQUEST_URL, listener, null);
        params = new HashMap<>();
        params.put("username", username);
        params.put("password", password);
    }

    @Override
    public Map<String, String> getParams() {
        return params;
    }
}
```

10.1.20 MainActivity.java

```

package com.example.lefteris_kasos.firstaid;
import android.app.Activity;
import android.content.DialogInterface;
import android.net.Uri;
import android.preference.DialogPreference;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.os.Process;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ImageButton;
import android.view.View;
import android.content.Intent;

import com.google.android.gms.appindexing.Action;
import com.google.android.gms.appindexing.AppIndex;
import com.google.android.gms.common.api.GoogleApiClient;

public class MainActivity extends Activity {

    ImageButton imageButton8;
    /**
     * ATTENTION: This was auto-generated to implement the App Indexing API.
     * See https://g.co/AppIndexing/AndroidStudio for more information.
     */
    private GoogleApiClient client;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.activity_main);

        imageButton8 = (ImageButton) findViewById(R.id.imageButton8);
        imageButton8.setOnClickListener((v) -> { finish(); });
        // ATTENTION: This was auto-generated to implement the App Indexing API.
        // See https://g.co/AppIndexing/AndroidStudio for more information.
        client = new GoogleApiClient.Builder(this).addApi(AppIndex.API).build();
    }

    @Override
    protected void onDestroy() {
        Process.killProcess(Process.myPid());
        super.onDestroy();
    }

    public void onClick4(View v) {

        if (v.getId() == R.id.imageButton4) {
            Intent i = new Intent(MainActivity.this, Psixoypo.class);
            startActivity(i);
        }
    }
}

```

```

public void onClick1(View v) {
    if (v.getId() == R.id.imageButton1) {
        Intent j = new Intent(MainActivity.this, Protesb.class);
        startActivity(j);
    }
}

public void onClick2(View v) {
    if (v.getId() == R.id.imageButton2) {
        Intent k = new Intent(MainActivity.this, Epeigontaefhmer.class);
        startActivity(k);
    }
}

public void onClick3(View v) {
    if (v.getId() == R.id.imageButton3) {
        Intent m = new Intent(MainActivity.this, Topothesia.class);
        startActivity(m);
    }
}

public void onClick5(View v) {
    if (v.getId() == R.id.imageButton5) {
        Intent n = new Intent(MainActivity.this, Istorikosfakelos.class);
        startActivity(n);
    }
}

public void onClick6(View v) {
    if (v.getId() == R.id.imageButton6) {
        Intent z = new Intent(MainActivity.this, Stoixeia.class);
        startActivity(z);
    }
}

public void showAlert(View view) {
    AlertDialog.Builder myAlert = new AlertDialog.Builder(this);
    myAlert.setMessage("Όνομα: Λευτέρης Επάνυμο: Κάσος Αριθμός Μητρώου: ΜΠΠΛ14027 Διπλωματική Εργασία Εξαμήνου")
        .setPositiveButton("Εξοδος", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
            }
        })
        .setTitle("Πανεπιστήμιο Πειραιώς")
        .setIcon(R.drawable.unipi)
        .create();
    myAlert.show();
}

```

```
@Override
public void onStart() {
    super.onStart();

    // ATTENTION: This was auto-generated to implement the App Indexing API.
    // See https://g.co/AppIndexing/AndroidStudio for more information.
    client.connect();
    Action viewAction = Action.newAction(
        Action.TYPE_VIEW, // TODO: choose an action type.
        "Main Page", // TODO: Define a title for the content shown.
        // TODO: If you have web page content that matches this app activity's content,
        // make sure this auto-generated web page URL is correct.
        // Otherwise, set the URL to null.
        Uri.parse("http://host/path"),
        // TODO: Make sure this auto-generated app URL is correct.
        Uri.parse("android-app://com.example.lefteris_kasos.firstaid/http/host/path")
    );
    AppIndex.AppIndexApi.start(client, viewAction);
}

@Override
public void onStop() {
    super.onStop();

    // ATTENTION: This was auto-generated to implement the App Indexing API.
    // See https://g.co/AppIndexing/AndroidStudio for more information.
    Action viewAction = Action.newAction(
        Action.TYPE_VIEW, // TODO: choose an action type.
        "Main Page", // TODO: Define a title for the content shown.
        // TODO: If you have web page content that matches this app activity's content,
        // make sure this auto-generated web page URL is correct.
        // Otherwise, set the URL to null.
        Uri.parse("http://host/path"),
        // TODO: Make sure this auto-generated app URL is correct.
        Uri.parse("android-app://com.example.lefteris_kasos.firstaid/http/host/path")
    );
    AppIndex.AppIndexApi.end(client, viewAction);
    client.disconnect();
}
}
```

10.1.21 Methis.java

```
package com.example.lefteris_kasos.firstaid;

import ...

/**
 * Created by Lefteris_Kasos on 26/06/2016.
 */
public class Methis extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.methis);

        Button play = (Button) findViewById(R.id.play);
        Button pause = (Button) findViewById(R.id.pause);
        Button stop = (Button) findViewById(R.id.stop);

        final MediaPlayer mp = MediaPlayer.create(Methis.this, R.raw.methis);
        play.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.start();
            }
        });

        pause.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.pause();
            }
        });

        stop.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.stop();
                Methis.this.finish();
            }
        });
    }
}
```

10.1.22 Pnigmos.java

```
package com.example.lefteris_kasos.firstaid;

import ...

/**
 * Created by Lefteris_Kasos on 26/06/2016.
 */
public class Pnigmos extends Activity{
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.pnigmos);

        Button play = (Button) findViewById(R.id.play);
        Button pause = (Button) findViewById(R.id.pause);
        Button stop = (Button) findViewById(R.id.stop);

        final MediaPlayer mp = MediaPlayer.create(Pnigmos.this, R.raw.pnigmos);
        play.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.start();
            }
        });

        pause.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.pause();
            }
        });

        stop.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.stop();
                Pnigmos.this.finish();
            }
        });
    }
}
```

10.1.23 Protesb.java

```

package com.example.lefteris_kasos.firstaid;

import ...

/**
 * Created by Lefteris Kasos on 03/06/2016.
 */
public class Protesb extends Activity {

    ListView list;

    String[] memeTitles;
    String[] memeDescriptions;
    int[] images = {R.drawable.kardioanapneystikh, R.drawable.apofraksiaeragogoy, R.drawable.pnigmos}
    /**
     * ATTENTION: This was auto-generated to implement the App Indexing API.
     * See https://g.co/AppIndexing/AndroidStudio for more information.
     */
    private GoogleApiClient client;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.protesb);

        Resources resources = getResources();
        memeTitles = resources.getStringArray(R.array.titles);
        memeDescriptions = resources.getStringArray(R.array.descriptions);

        list = (ListView) findViewById(R.id.listView);
        lefterisAdapter adapter = new lefterisAdapter(this, memeTitles, images, memeDescriptions);
        list.setAdapter(adapter);

        list.setOnItemClickListener(new AdapterView.OnItemClickListener() {

            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                switch (position) {
                    case 0:
                        Intent a = new Intent(Protesb.this, Kardio.class);
                        startActivity(a);

                        break;

                    case 1:
                        Intent b = new Intent(Protesb.this, Apofraksiaeragwou.class);
                        startActivity(b);

                        break;

                    case 2:
                        Intent c = new Intent(Protesb.this, Pnigmos.class);
                        startActivity(c);

                        break;

                    case 3:
                        Intent d = new Intent(Protesb.this, Aimoragia.class);
                        startActivity(d);

                        break;
                }
            }
        });
    }
}

```

```
case 4:
    Intent e = new Intent(Protesb.this, Rinoragia.class);
    startActivity(e);

    break;
case 5:
    Intent f = new Intent(Protesb.this, Trofikid.class);
    startActivity(f);

    break;
case 6:
    Intent g = new Intent(Protesb.this, Katagma.class);
    startActivity(g);

    break;
case 7:
    Intent h = new Intent(Protesb.this, Thermopliksia.class);
    startActivity(h);

    break;
case 8:
    Intent k = new Intent(Protesb.this, Ypothemia.class);
    startActivity(k);

    break;
case 9:
    Intent l = new Intent(Protesb.this, Lipothimia.class);
    startActivity(l);

    break;
case 10:
    Intent m = new Intent(Protesb.this, Axinos.class);
    startActivity(m);

    break;
case 11:
    Intent n = new Intent(Protesb.this, Fidi.class);
    startActivity(n);

    break;
case 12:
    Intent p = new Intent(Protesb.this, Tsimpouri.class);
    startActivity(p);

    break;
case 13:
    Intent q = new Intent(Protesb.this, Tsouextra.class);
    startActivity(q);

    break;
```



```
        case 14:
            Intent r = new Intent(Protesb.this, Methis.class);
            startActivity(r);

            break;
        case 15:
            Intent v = new Intent(Protesb.this, Krpanikou.class);
            startActivity(v);

            break;
        case 16:
            Intent w = new Intent(Protesb.this, Epilipikikr.class);
            startActivity(w);

            break;

        default:
    }

    });
    // ATTENTION: This was auto-generated to implement the App Indexing API.
    // See https://g.co/AppIndexing/AndroidStudio for more information.
    client = new GoogleApiClient.Builder(this).addApi(AppIndex.API).build();
}

@Override
public void onStart() {
    super.onStart();

    // ATTENTION: This was auto-generated to implement the App Indexing API.
    // See https://g.co/AppIndexing/AndroidStudio for more information.
    client.connect();
    Action viewAction = Action.newAction(
        Action.TYPE_VIEW, // TODO: choose an action type.
        "Protesb Page", // TODO: Define a title for the content shown.
        // TODO: If you have web page content that matches this app activity's content,
        // make sure this auto-generated web page URL is correct.
        // Otherwise, set the URL to null.
        Uri.parse("http://host/path"),
        // TODO: Make sure this auto-generated app URL is correct.
        Uri.parse("android-app://com.example.lefteris_kasos.firstaid/http/host/path")
    );
    AppIndex.AppIndexApi.start(client, viewAction);
}
```

```
@Override
public void onStop() {
    super.onStop();

    // ATTENTION: This was auto-generated to implement the App Indexing API.
    // See https://g.co/AppIndexing/AndroidStudio for more information.
    Action viewAction = Action.newAction(
        Action.TYPE_VIEW, // TODO: choose an action type.
        "Protesb Page", // TODO: Define a title for the content shown.
        // TODO: If you have web page content that matches this app activity's content,
        // make sure this auto-generated web page URL is correct.
        // Otherwise, set the URL to null.
        Uri.parse("http://host/path"),
        // TODO: Make sure this auto-generated app URL is correct.
        Uri.parse("android-app://com.example.lefteris_kasos.firstaid/http/host/path")
    );
    AppIndex.AppIndexApi.end(client, viewAction);
    client.disconnect();
}

class lefterisAdapter extends ArrayAdapter<String> {
    Context context;
    int images[];
    String[] titleArray;
    String[] descriptionArray;

    lefterisAdapter(Context c, String[] titles, int imgs[], String[] desc) {
        super(c, R.layout.single_row, R.id.textView, titles);
        this.context = c;
        this.images = imgs;
        this.titleArray = titles;
        this.descriptionArray = desc;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        View row = inflater.inflate(R.layout.single_row, parent, false);
        ImageView myImage = (ImageView) row.findViewById(R.id.imageView3);
        TextView myTitle = (TextView) row.findViewById(R.id.textView20);
        TextView myDescription = (TextView) row.findViewById(R.id.textView21);

        myImage.setImageResource(images[position]);
        myTitle.setText(titleArray[position]);
        myDescription.setText(descriptionArray[position]);

        return row;
    }
}
```

10.1.24 Psixoypo.java

```

package com.example.lefteris_kasos.firstaid;

import ...

/**
 * Created by Lefteris Kasos on 02/06/2016.
 */
public class Psixoypo extends Activity implements View.OnClickListener {

    /**
     * ATTENTION: This was auto-generated to implement the App Indexing API.
     * See https://g.co/AppIndexing/AndroidStudio for more information.
     */
    private GoogleApiClient client;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.psixoypo);

        Button button1 = (Button) this.findViewById(R.id.button1);
        Button button2 = (Button) this.findViewById(R.id.button2);
        Button button3 = (Button) this.findViewById(R.id.button3);
        Button button4 = (Button) this.findViewById(R.id.button4);

        button1.setOnClickListener(this);
        button2.setOnClickListener(this);
        button3.setOnClickListener(this);
        button4.setOnClickListener(this);

        // ATTENTION: This was auto-generated to implement the App Indexing API.
        // See https://g.co/AppIndexing/AndroidStudio for more information.
        client = new GoogleApiClient.Builder(this).addApi(AppIndex.API).build();
    }

    @Override
    public void onStart() {
        super.onStart();

        // ATTENTION: This was auto-generated to implement the App Indexing API.
        // See https://g.co/AppIndexing/AndroidStudio for more information.
        client.connect();
        Action viewAction = Action.newAction(
            Action.TYPE_VIEW, // TODO: choose an action type.
            "Psixoypo Page", // TODO: Define a title for the content shown.
            // TODO: If you have web page content that matches this app activity's content,
            // make sure this auto-generated web page URL is correct.
            // Otherwise, set the URL to null.
            Uri.parse("http://host/path"),
            // TODO: Make sure this auto-generated app URL is correct.
            Uri.parse("android-app://com.example.lefteris_kasos.firstaid/http/host/path")
        );
        AppIndex.AppIndexApi.start(client, viewAction);
    }
}

```

```
@Override
public void onStop() {
    super.onStop();

    // ATTENTION: This was auto-generated to implement the App Indexing API.
    // See https://g.co/AppIndexing/AndroidStudio for more information.
    Action viewAction = Action.newAction(
        Action.TYPE_VIEW, // TODO: choose an action type.
        "Psixoypo Page", // TODO: Define a title for the content shown.
        // TODO: If you have web page content that matches this app activity's content,
        // make sure this auto-generated web page URL is correct.
        // Otherwise, set the URL to null.
        Uri.parse("http://host/path"),
        // TODO: Make sure this auto-generated app URL is correct.
        Uri.parse("android-app://com.example.lefteris_kasos.firstaid/http/host/path")
    );
    AppIndex.AppIndexApi.end(client, viewAction);
    client.disconnect();
}

@Override
public void onClick(View v) {
    switch(v.getId()){
        case R.id.button1:
            Intent out = new Intent();
            out.setAction(Intent.ACTION_DIAL);
            out.setData(Uri.parse("tel:"+Uri.encode("197")));
            startActivity(out);
            break;
        case R.id.button2:
            Intent a = new Intent();
            a.setAction(Intent.ACTION_DIAL);
            a.setData(Uri.parse("tel:"+Uri.encode("1056")));
            startActivity(a);
            break;
        case R.id.button3:
            Intent b = new Intent();
            b.setAction(Intent.ACTION_DIAL);
            b.setData(Uri.parse("tel:"+Uri.encode("15900")));
            startActivity(b);
            break;
        case R.id.button4:
            Intent c = new Intent();
            c.setAction(Intent.ACTION_DIAL);
            c.setData(Uri.parse("tel:"+Uri.encode("2108105068")));
            startActivity(c);
            break;
    }
}
}
```

10.1.25 RegisterActivity.java

```

package com.example.lefteris_kasos.firstaid;

import ...

public class RegisterActivity extends AppCompatActivity {

    /**
     * ATTENTION: This was auto-generated to implement the App Indexing API.
     * See https://g.co/AppIndexing/AndroidStudio for more information.
     */
    private GoogleApiClient client;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);

        final EditText etAge = (EditText) findViewById(R.id.etAge);
        final EditText etName = (EditText) findViewById(R.id.etName);
        final EditText etUsername = (EditText) findViewById(R.id.etUsername);
        final EditText etPassword = (EditText) findViewById(R.id.etPassword);
        final Button bRegister = (Button) findViewById(R.id.bRegister);

        assert bRegister != null;
        bRegister.setOnClickListener((view) - {
            final String name;
            name = etName.getText().toString();
            final String username;
            username = etUsername.getText().toString();
            final String password;
            password = etPassword.getText().toString();
            final int age;
            age = Integer.parseInt(etAge.getText().toString());

            Response.Listener<String> responseListener = (response) - {
                try {
                    JSONObject jsonResponse = new JSONObject(response);

                    boolean success = jsonResponse.getBoolean("success");

                    if (success) {
                        Intent intent = new Intent(RegisterActivity.this, Istorikosfakelos.class);
                        RegisterActivity.this.startActivity(intent);
                    } else {
                        AlertDialog.Builder builder = new AlertDialog.Builder(RegisterActivity.this);
                        builder.setMessage("Η εγγραφή απέτυχε")
                                .setNegativeButton("ΠΡΟΣΠΑΘΗΤΕ ΞΑΝΑ", null)
                                .create()
                                .show();
                    }
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }

        });
    }
}

```

```

        RegisterRequest registerRequest = new RegisterRequest(name, username, age, password, responseListener);

        RequestQueue queue = Volley.newRequestQueue(RegisterActivity.this);
        queue.add(registerRequest);
    });
    // ATTENTION: This was auto-generated to implement the App Indexing API.
    // See https://g.co/AppIndexing/AndroidStudio for more information.
    client = new GoogleApiClient.Builder(this).addApi(AppIndex.API).build();
}

@Override
public void onStart() {
    super.onStart();

    // ATTENTION: This was auto-generated to implement the App Indexing API.
    // See https://g.co/AppIndexing/AndroidStudio for more information.
    client.connect();
    Action viewAction = Action.newAction(
        Action.TYPE_VIEW, // TODO: choose an action type.
        "Register Page", // TODO: Define a title for the content shown.
        // TODO: If you have web page content that matches this app activity's content,
        // make sure this auto-generated web page URL is correct.
        // Otherwise, set the URL to null.
        Uri.parse("http://host/path"),
        // TODO: Make sure this auto-generated app URL is correct.
        Uri.parse("android-app://com.example.lefteris_kasos.firstaid/http/host/path")
    );
    AppIndex.AppIndexApi.start(client, viewAction);
}

@Override
public void onStop() {
    super.onStop();

    // ATTENTION: This was auto-generated to implement the App Indexing API.
    // See https://g.co/AppIndexing/AndroidStudio for more information.
    Action viewAction = Action.newAction(
        Action.TYPE_VIEW, // TODO: choose an action type.
        "Register Page", // TODO: Define a title for the content shown.
        // TODO: If you have web page content that matches this app activity's content,
        // make sure this auto-generated web page URL is correct.
        // Otherwise, set the URL to null.
        Uri.parse("http://host/path"),
        // TODO: Make sure this auto-generated app URL is correct.
        Uri.parse("android-app://com.example.lefteris_kasos.firstaid/http/host/path")
    );
    AppIndex.AppIndexApi.end(client, viewAction);
    client.disconnect();
}
}

```

10.1.26 RegisterRequest.java

```

package com.example.lefteris_kasos.firstaid;

import ...

/**
 * Created by Lefteris Kasos on 30/06/2016.
 */
public class RegisterRequest extends StringRequest {

    private static final String REGISTER_REQUEST_URL = "http://firstaid.esy.es/Register.php";
    private Map<String, String> params;

    public RegisterRequest(String name, String username, int age, String password, Response.Listener<String>listener) {
        super(Method.POST, REGISTER_REQUEST_URL, listener, null);
        params = new HashMap<>();
        params.put("name", name);
        params.put("username", username);
        params.put("password", password);
        params.put("age", age + "");
    }

    @Override
    public Map<String, String> getParams() {
        return params;
    }
}

```

10.1.27 Rinoragia.java

```
package com.example.lifteris_kasos.firstaid;

import android.app.Activity;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

/**
 * Created by Lifteris_Kasos on 26/06/2016.
 */
public class Rinoragia extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.rinoragia);
        Button play = (Button) findViewById(R.id.play);
        Button pause = (Button) findViewById(R.id.pause);
        Button stop = (Button) findViewById(R.id.stop);

        final MediaPlayer mp = MediaPlayer.create(Rinoragia.this, R.raw.rinorragia);
        play.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.start();
            }
        });

        pause.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.pause();
            }
        });

        stop.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.stop();
                Rinoragia.this.finish();
            }
        });
    }
}
```

10.1.28 Stoixeia.java

```

package com.example.lefteris_kasos.firstaid;

import ...

/**
 * Created by Lefteris_Kasos on 03/06/2016.
 */
public class Stoixeia extends Activity {
    EditText editText;
    TextView textView;
    /**
     * ATTENTION: This was auto-generated to implement the App Indexing API.
     * See https://g.co/AppIndexing/AndroidStudio for more information.
     */
    private GoogleApiClient client;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.stoixeia);
        editText = (EditText) findViewById(R.id.editText1);
        textView = (TextView) findViewById(R.id.textView16);
        // ATTENTION: This was auto-generated to implement the App Indexing API.
        // See https://g.co/AppIndexing/AndroidStudio for more information.
        client = new GoogleApiClient.Builder(this).addApi(AppIndex.API).build();
    }

    public void read(View view) {
        try {
            FileInputStream fileInputStream= openFileInput("Stoixeia.txt");
            InputStreamReader inputStreamReader= new InputStreamReader(fileInputStream);
            BufferedReader bufferedReader = new BufferedReader(inputStreamReader);
            StringBuffer stringBuffer= new StringBuffer();
            String lines;
            while((lines=bufferedReader.readLine())!=null) {

                stringBuffer.append(lines+"\n");
            }

            textView.setText(stringBuffer.toString());
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public void write(View view) {
        String mytextmessage = editText.getText().toString();
        try {
            FileOutputStream fileOutputStream = openFileOutput("Stoixeia.txt", MODE_PRIVATE);
            fileOutputStream.write(mytextmessage.getBytes());
            fileOutputStream.close();
            Toast.makeText(getApplicationContext(), "Επιτυχής Αποθήκευση", Toast.LENGTH_LONG).show();
            editText.setText("");
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```



```
    }

    @Override
    public void onStart() {
        super.onStart();

        // ATTENTION: This was auto-generated to implement the App Indexing API.
        // See https://g.co/AppIndexing/AndroidStudio for more information.
        client.connect();
        Action viewAction = Action.newAction(
            Action.TYPE_VIEW, // TODO: choose an action type.
            "Stoixeia Page", // TODO: Define a title for the content shown.
            // TODO: If you have web page content that matches this app activity's content,
            // make sure this auto-generated web page URL is correct.
            // Otherwise, set the URL to null.
            Uri.parse("http://host/path"),
            // TODO: Make sure this auto-generated app URL is correct.
            Uri.parse("android-app://com.example.lefteris_kasos.firstaid/http/host/path")
        );
        AppIndex.AppIndexApi.start(client, viewAction);
    }

    @Override
    public void onStop() {
        super.onStop();

        // ATTENTION: This was auto-generated to implement the App Indexing API.
        // See https://g.co/AppIndexing/AndroidStudio for more information.
        Action viewAction = Action.newAction(
            Action.TYPE_VIEW, // TODO: choose an action type.
            "Stoixeia Page", // TODO: Define a title for the content shown.
            // TODO: If you have web page content that matches this app activity's content,
            // make sure this auto-generated web page URL is correct.
            // Otherwise, set the URL to null.
            Uri.parse("http://host/path"),
            // TODO: Make sure this auto-generated app URL is correct.
            Uri.parse("android-app://com.example.lefteris_kasos.firstaid/http/host/path")
        );
        AppIndex.AppIndexApi.end(client, viewAction);
        client.disconnect();
    }
}
```

10.1.29 Thermopliksia.java

```
package com.example.lefteris_kasos.firstaid;

import ...

/**
 * Created by Lefteris Kasos on 26/06/2016.
 */
public class Thermopliksia extends Activity{
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.thermopliksia);
        Button play = (Button) findViewById(R.id.play);
        Button pause = (Button) findViewById(R.id.pause);
        Button stop = (Button) findViewById(R.id.stop);

        final MediaPlayer mp = MediaPlayer.create(Thermopliksia.this, R.raw.thermopliksia);
        play.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.start();
            }
        });

        pause.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.pause();
            }
        });

        stop.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.stop();
                Thermopliksia.this.finish();
            }
        });
    }
}
```

10.1.30 Topothesia.java

```
package com.example.lefteris_kasos.firstaid;

import android.content.pm.PackageManager;
import android.os.Bundle;
import android.support.v4.app.ActivityCompat;
import android.support.v4.app.FragmentActivity;

import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;

public class Topothesia extends FragmentActivity implements OnMapReadyCallback {

    private GoogleMap mMap;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_topothesia);

        SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
    }

    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;

        if (ActivityCompat.checkSelfPermission(this, android.Manifest.permission.ACCESS_FINE_LOCATION)
                != PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(
                this, android.Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            // TODO: Consider calling
            // ActivityCompat#requestPermissions
            // here to request the missing permissions, and then overriding
            // public void onRequestPermissionsResult(int requestCode, String[] permissions,
            // int[] grantResults)
            // to handle the case where the user grants the permission. See the documentation
            // for ActivityCompat#requestPermissions for more details.
            return;
        }
        mMap.setMyLocationEnabled(true);
    }
}
```

10.1.31 Trofikid.java

```
package com.example.lefteris_kasos.firstaid;

import ...

/**
 * Created by Lefteris Kasos on 26/06/2016.
 */
public class Trofikid extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.trofikid);
        Button play = (Button) findViewById(R.id.play);
        Button pause = (Button) findViewById(R.id.pause);
        Button stop = (Button) findViewById(R.id.stop);

        final MediaPlayer mp = MediaPlayer.create(Trofikid.this, R.raw.trofikidi);
        play.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.start();
            }
        });

        pause.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.pause();
            }
        });

        stop.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.stop();
                Trofikid.this.finish();
            }
        });
    }
}
```

10.1.32 Tsimpouri.java

```
package com.example.lefteris_kasos.firstaid;

import ...

/**
 * Created by Lefteris_Kasos on 26/06/2016.
 */
public class Tsimpouri extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.tsimpouri);

        Button play = (Button) findViewById(R.id.play);
        Button pause = (Button) findViewById(R.id.pause);
        Button stop = (Button) findViewById(R.id.stop);

        final MediaPlayer mp = MediaPlayer.create(Tsimpouri.this, R.raw.tsimpouri);
        play.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.start();
            }
        });

        pause.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.pause();
            }
        });

        stop.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.stop();
                Tsimpouri.this.finish();
            }
        });
    }
}
```

10.1.33 Tsouxttra.java

```
package com.example.lefteris_kasos.firstaid;

import ...

/**
 * Created by Lefteris_Kasos on 26/06/2016.
 */
public class Tsouxttra extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.tsouxttra);

        Button play = (Button) findViewById(R.id.play);
        Button pause = (Button) findViewById(R.id.pause);
        Button stop = (Button) findViewById(R.id.stop);

        final MediaPlayer mp = MediaPlayer.create(Tsouxttra.this, R.raw.tsouxttra);
        play.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.start();
            }
        });

        pause.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.pause();
            }
        });

        stop.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.stop();
                Tsouxttra.this.finish();
            }
        });
    }
}
```

10.1.34 UserAreaActivity.java

```

package com.example.lefteris_kasos.firstaid;

import android.content.DialogInterface;
import android.content.Intent;
import android.net.Uri;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.toolbox.Volley;
import com.google.android.gms.appindexing.Action;
import com.google.android.gms.appindexing.AppIndex;
import com.google.android.gms.common.api.GoogleApiClient;

import org.json.JSONException;
import org.json.JSONObject;

public class UserAreaActivity extends AppCompatActivity {

    EditText onoma, eponymo;
    TextView textViewLefteris;

    private GoogleApiClient client;
    /**
     * ATTENTION: This was auto-generated to implement the App Indexing API.
     * See https://g.co/AppIndexing/AndroidStudio for more information.
     */
    private GoogleApiClient client2;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate (savedInstanceState);
        setContentView (R.layout.activity_user_area);

        final EditText tv1 = (EditText) findViewById (R.id.tv1);
        final EditText tv2 = (EditText) findViewById (R.id.tv2);
        final EditText tv3 = (EditText) findViewById (R.id.tv3);
        final EditText tv4 = (EditText) findViewById (R.id.tv4);
        final EditText tv5 = (EditText) findViewById (R.id.tv5);
        final EditText tv6 = (EditText) findViewById (R.id.tv6);
        final EditText tv7 = (EditText) findViewById (R.id.tv7);
        final EditText tv8 = (EditText) findViewById (R.id.tv8);
        final EditText tv9 = (EditText) findViewById (R.id.tv9);
        final EditText tv10 = (EditText) findViewById (R.id.tv10);
        final EditText tv11 = (EditText) findViewById (R.id.tv11);
        final EditText tv12 = (EditText) findViewById (R.id.tv12);
        final EditText tv13 = (EditText) findViewById (R.id.tv13);
        final EditText tv14 = (EditText) findViewById (R.id.tv14);
        final EditText tv15 = (EditText) findViewById (R.id.tv15);
        final EditText tv16 = (EditText) findViewById (R.id.tv16);
        final EditText tv17 = (EditText) findViewById (R.id.tv17);
        final EditText tv18 = (EditText) findViewById (R.id.tv18);
        final EditText tv19 = (EditText) findViewById (R.id.tv19);
        final EditText tv20 = (EditText) findViewById (R.id.tv20);
    }
}

```

```

final EditText tv21 = (EditText) findViewById (R.id.tv21);
final EditText tv22 = (EditText) findViewById (R.id.tv22);
final EditText tv23 = (EditText) findViewById (R.id.tv23);
final EditText tv24 = (EditText) findViewById (R.id.tv24);
final EditText tv25 = (EditText) findViewById (R.id.tv25);
final EditText tv26 = (EditText) findViewById (R.id.tv26);
final EditText tv27 = (EditText) findViewById (R.id.tv27);
final EditText tv28 = (EditText) findViewById (R.id.tv28);
final EditText tv29 = (EditText) findViewById (R.id.tv29);
final EditText tv30 = (EditText) findViewById (R.id.tv30);
final EditText tv31 = (EditText) findViewById (R.id.tv31);
final EditText tv32 = (EditText) findViewById (R.id.tv32);
final EditText tv33 = (EditText) findViewById (R.id.tv33);
final EditText tv34 = (EditText) findViewById (R.id.tv34);
final EditText tv35 = (EditText) findViewById (R.id.tv35);
final EditText tv36 = (EditText) findViewById (R.id.tv36);
final EditText tv37 = (EditText) findViewById (R.id.tv37);
final EditText tv38 = (EditText) findViewById (R.id.tv38);
final EditText tv39 = (EditText) findViewById (R.id.tv39);
final EditText tv40 = (EditText) findViewById (R.id.tv40);
final EditText tv41 = (EditText) findViewById (R.id.tv41);
final EditText tv42 = (EditText) findViewById (R.id.tv42);
final EditText tv43 = (EditText) findViewById (R.id.tv43);
final Button aRegister = (Button) findViewById (R.id.aRegister);
final TextView welcomeMessage = (TextView) findViewById (R.id.tvWelcomeMsg);

assert aRegister != null;

aRegister.setOnClickListener ((view) - {
    final String onoma;
    onoma = tv1.getText ().toString ();

    final String eponymo;
    eponymo = tv2.getText ().toString ();

    final String onomamitros;
    onomamitros = tv3.getText ().toString ();

    final String onomapatros;
    onomapatros = tv4.getText ().toString ();

    final String fulo;
    fulo = tv5.getText ().toString ();

    final String hmerominiagenisis;
    hmerominiagenisis = tv6.getText ().toString ();

    final String hlikia;
    hlikia = tv7.getText ().toString ();

    final String xwrakatagwghs;
    xwrakatagwghs = tv8.getText ().toString ();

    final String dieythinsi;
    dieythinsi = tv9.getText ().toString ();

    final String taxydromikoskwdikas;
    taxydromikoskwdikas = tv10.getText ().toString ();

```



```
final String perioxh;  
perioxh = tv11.getText ().toString ();  
  
final String polh;  
polh = tv12.getText ().toString ();  
  
final String xwra;  
xwra = tv13.getText ().toString ();  
  
final String thlefwno;  
thlefwno = tv14.getText ().toString ();  
  
final String kinhto;  
kinhto = tv15.getText ().toString ();  
  
final String arithmostaytohtas;  
arithmostaytohtas = tv16.getText ().toString ();  
  
final String afm;  
afm = tv17.getText ().toString ();  
  
final String amka;  
amka = tv18.getText ().toString ();  
  
final String eksetaseiseginan;  
eksetaseiseginan = tv19.getText ().toString ();  
  
final String khlh;  
khlh = tv20.getText ().toString ();  
  
final String aimoroides;  
aimoroides = tv21.getText ().toString ();  
  
final String ourhsh;  
ourhsh = tv22.getText ().toString ();  
  
final String sakxarwdhdiavitis;  
sakxarwdhdiavitis = tv23.getText ().toString ();  
  
final String petres;  
petres = tv24.getText ().toString ();  
  
final String arthritida;  
arthritida = tv25.getText ().toString ();  
  
final String problmataostwn;  
problmataostwn = tv26.getText ().toString ();  
  
final String apoleiadakylwn;  
apoleiadakylwn = tv27.getText ().toString ();  
  
final String ponoiplaths;  
ponoiplaths = tv28.getText ().toString ();  
  
final String problmatagonatwn;  
problmatagonatwn = tv29.getText ().toString ();  
  
final String problmatadiatrofhs;  
problmatadiatrofhs = tv30.getText ().toString ();
```

```
final String nautia;
nautia = tv31.getText ().toString ();

final String problhmatayprou;
problhmatayprou = tv32.getText ().toString ();

final String katathlipsi;
katathlipsi = tv33.getText ().toString ();

final String apwleiamnhmhs;
apwleiamnhmhs = tv34.getText ().toString ();

final String neurologikaproblhmata;
neurologikaproblhmata = tv35.getText ().toString ();

final String apwleiasynidisis;
apwleiasynidisis = tv36.getText ().toString ();

final String neuritida;
neuritida = tv37.getText ().toString ();

final String paralysh;
paralysh = tv38.getText ().toString ();

final String epilipsia;
epilipsia = tv39.getText ().toString ();

final String gynaikologikaproblhmata;
gynaikologikaproblhmata = tv40.getText ().toString ();

final String emmhnoorysh;
emmhnoorysh = tv41.getText ().toString ();

final String eksetaseisprogrammatismenes;
eksetaseisprogrammatismenes = tv42.getText ().toString ();

final String apotelesmata;
apotelesmata = tv43.getText ().toString ();

final Response.Listener<String> responseListener = new Response.Listener<String> () {
```

```

@Override
public void onResponse(String response) {

    try {
        JSONObject jsonResponse = new JSONObject (response);

        boolean success = jsonResponse.getBoolean ("success");
        if (success) {

            AlertDialog.Builder builder = new AlertDialog.Builder (UserAreaActivity.this);
            builder.setMessage ("Η εγγραφή στη βάση δεδομένων ολοκληρώθηκε με επιτυχία")
                .create ()
                .show ();

        } else {

            AlertDialog.Builder builder = new AlertDialog.Builder (UserAreaActivity.this);
            builder.setMessage ("Η εγγραφή απέτυχε")
                .setNegativeButton ("ΠΡΟΣΠΑΘΗΣΤΕ ΞΑΝΑ", null)
                .create ()
                .show ();

        }

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

}

};

aRegisterRequest aregisterRequest = new aRegisterRequest (onoma, eponymo, onomamitros,
    onomapatros, fulo, hmerominiagenisis, hlikia, xwrakatagwghs, dieythinsi,
    taxydromikoskwdikas, perioxh, polh, xwra, thlefwno, kinhto, arithmostaytothtas,
    afm, amka, eksetaseiseginan, khlh, aimoroides, ourhsh, sakxarwdhdiavitis,
    petres, arthritida, problhmataostwn, apoleiadakylwn, ponoiplaths,
    problhmatagonatwn, problhmatadiatrofhs, nautia, problhmataypnou,
    katathlipsi, apwleiamnhmhs, neurologikaproblhmata, apwleiasynidisis,
    neuritida, paralysh, epilipsia, gynaikologikaproblhmata, emmhnorvsh,
    eksetaseisprogrammatismenes, apotelesmata, responseListener);

RequestQueue aqueue = Volley.newRequestQueue (UserAreaActivity.this);
aqueue.add (aregisterRequest);

});

Intent intent = getIntent ();
String name = intent.getStringExtra ("name");

String message = name + "--Καλώς 'Ηλθατε ";
welcomeMessage.setText (message);

// ATTENTION: This was auto-generated to implement the App Indexing API.
// See https://g.co/AppIndexing/AndroidStudio for more information.
client2 = new GoogleApiClient.Builder (this).addApi (AppIndex.API).build ();
}

```

```
@Override
public void onStart() {
    super.onStart ();

    // ATTENTION: This was auto-generated to implement the App Indexing API.
    // See https://g.co/AppIndexing/AndroidStudio for more information.
    client2.connect ();
    Action viewAction = Action.newAction (
        Action.TYPE_VIEW, // TODO: choose an action type.
        "UserArea Page", // TODO: Define a title for the content shown.
        // TODO: If you have web page content that matches this app activity's content,
        // make sure this auto-generated web page URL is correct.
        // Otherwise, set the URL to null.
        Uri.parse ("http://host/path"),
        // TODO: Make sure this auto-generated app URL is correct.
        Uri.parse ("android-app://com.example.lefteris_kasos.firstaid/http/host/path")
    );
    AppIndex.AppIndexApi.start (client2, viewAction);
}

@Override
public void onStop() {
    super.onStop ();

    // ATTENTION: This was auto-generated to implement the App Indexing API.
    // See https://g.co/AppIndexing/AndroidStudio for more information.
    Action viewAction = Action.newAction (
        Action.TYPE_VIEW, // TODO: choose an action type.
        "UserArea Page", // TODO: Define a title for the content shown.
        // TODO: If you have web page content that matches this app activity's content,
        // make sure this auto-generated web page URL is correct.
        // Otherwise, set the URL to null.
        Uri.parse ("http://host/path"),
        // TODO: Make sure this auto-generated app URL is correct.
        Uri.parse ("android-app://com.example.lefteris_kasos.firstaid/http/host/path")
    );
    AppIndex.AppIndexApi.end (client2, viewAction);
    client2.disconnect ();
}

public void onClick(View view) {

    if (view.getId () == R.id.button_list) {

        Intent zk = new Intent (UserAreaActivity.this, BasicActivity.class);
        startActivity (zk);
    }
}
}
```

10.1.35 Ypothemia.java

```
package com.example.lefteris_kasos.firstaid;

import ...

/**
 * Created by Lefteris_Kasos on 26/06/2016.
 */
public class Ypothemia extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.ypothemia);

        Button play = (Button) findViewById(R.id.play);
        Button pause = (Button) findViewById(R.id.pause);
        Button stop = (Button) findViewById(R.id.stop);

        final MediaPlayer mp = MediaPlayer.create(Ypothemia.this, R.raw.ypothemia);
        play.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.start();
            }
        });

        pause.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.pause();
            }
        });

        stop.setOnClickListener((view) - {
            mp.stop();
            Ypothemia.this.finish();
        });
    }
}
```

10.1.36 Connector.java

```
package com.example.lefteris_kasos.firstaid.mMySQL;

import ...

/**
 * Created by Lefteris_Kasos on 04/07/2016.
 */
public class Connector {

    public static HttpURLConnection connect(String urlAddress) {
        try {
            URL url = new URL(urlAddress);
            HttpURLConnection con = (HttpURLConnection) url.openConnection();

            //SET PROPS
            con.setRequestMethod("GET");
            con.setConnectTimeout(20000);
            con.setReadTimeout(20000);
            con.setDoInput(true);

            return con;
        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }

        return null;
    }
}
```

10.1.37 DataParser.java

```
package com.example.lefteris_kasos.firstaid.mMySQL;

import ...

/**
 * Created by Lefteris Kasos on 04/07/2016.
 */
public class DataParser extends AsyncTask<Void,Void,Integer> {

    Context c;
    RecyclerView rv;
    String jsonData;

    ProgressDialog pd;
    ArrayList<Spacecraft> spacecrafts=new ArrayList<>();

    public DataParser(Context c, RecyclerView rv, String jsonData) {
        this.c = c;
        this.rv = rv;
        this.jsonData = jsonData;
    }

    @Override
    protected void onPreExecute() {
        super.onPreExecute();

        pd = new ProgressDialog(c);
        pd.setTitle("Ανάλυση");
        pd.setMessage("Ανάλυση...Παρακαλώ Περιμένετε");
        pd.show();
    }

    @Override
    protected Integer doInBackground(Void... params) {
        return this.parseData();
    }

    @Override
    protected void onPostExecute(Integer result) {
        super.onPostExecute(result);

        pd.dismiss();

        if(result==0)
        {
            Toast.makeText(c,"Unable To Parse",Toast.LENGTH_SHORT).show();
        }else
        {
            //BIND DATA USING ADAPTER
            MyAdapter adapter=new MyAdapter(c,spacecrafts);
            rv.setAdapter(adapter);
        }
    }
}
```

```
private int parseData()
{
    try {
        JSONArray ja=new JSONArray(jsonData);
        JSONObject jo=null;

        spacecrafts.clear();
        Spacecraft s=null;

        for(int i=0;i<ja.length();i++)
        {
            jo=ja.getJSONObject(i);

            int user_id=jo.getInt("user_id");
            String onoma=jo.getString("onoma");
            String eponymo=jo.getString("eponymo");
            String onomapatros=jo.getString("onomapatros");

            s=new Spacecraft();
            s.setUser_id(user_id);
            s.setOnoma(onoma);
            s.setEponymo(eponymo);
            s.setOnomapatros(onomapatros);

            spacecrafts.add(s);
        }

        return 1;
    } catch (JSONException e) {
        e.printStackTrace();
    }

    return 0;
}
```


10.1.38 Downloader.java

```

package com.example.lefteris_kasos.firstaid.mMySQL;

import android.app.ProgressDialog;
import android.content.Context;
import android.os.AsyncTask;
import android.support.v7.widget.RecyclerView;
import android.widget.Toast;

import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;

/**
 * Created by Lefteris_Kasos on 04/07/2016.
 */
public class Downloader extends AsyncTask<Void,Void,String> {

    Context c;
    String urlAddress;
    RecyclerView rv;

    ProgressDialog pd;

    public Downloader(Context c, String urlAddress, RecyclerView rv) {
        this.c = c;
        this.urlAddress = urlAddress;
        this.rv = rv;
    }

    @Override
    protected void onPreExecute() {
        super.onPreExecute();

        pd=new ProgressDialog(c);
        pd.setTitle("Ανάκτηση Δεδομένων");
        pd.setMessage("Ανάκτηση Δεδομένων...Παρακαλώ Περιμένετε");
        pd.show();
    }

    @Override
    protected String doInBackground(Void... params) {
        return this.downloadData();
    }

    @Override
    protected void onPostExecute(String s) {
        super.onPostExecute(s);

        pd.dismiss();
        if(s==null)
        {
            Toast.makeText(c,"Ανεπιτυχής",Toast.LENGTH_SHORT).show();
        }else
        {
            //CALL DATA PARSER TO PARSE IT
            DataParser parser=new DataParser(c,rv,s);
            parser.execute();
        }
    }
}

```

```
private String downloadData()
{
    HttpURLConnection con=Connector.connect(urlAddress);
    if(con==null)
    {
        return null;
    }

    InputStream is=null;
    try {
        is=new BufferedInputStream(con.getInputStream());
        BufferedReader br=new BufferedReader(new InputStreamReader(is));

        String line=null;
        StringBuffer response=new StringBuffer();

        if(br != null)
        {
            while ((line=br.readLine()) != null)
            {
                response.append(line+"\n");
            }

            br.close();

        }else {
            return null;
        }

        return response.toString();
    } catch (IOException e) {
        e.printStackTrace();
    }finally {
        if(is != null)
        {
            try {
                is.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }

    return null;
}
}
```

10.1.39 MyAdapter.java

```

package com.example.lefteris_kasos.firstaid.mRecycler;

import ...

/**
 * Created by Lefteris_Kasos on 04/07/2016.
 */
public class MyAdapter extends RecyclerView.Adapter<MyHolder> {

    Context c;
    ArrayList<Spacecraft> spacecrafts;

    public MyAdapter(Context c, ArrayList<Spacecraft> spacecrafts) {
        this.c = c;
        this.spacecrafts = spacecrafts;
    }

    @Override
    public MyHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View v = LayoutInflater.from(parent.getContext()).inflate(R.layout.model, parent, false);
        return new MyHolder(v);
    }

    @Override
    public void onBindViewHolder(MyHolder holder, int position) {
        holder.onomaTxt.setText(spacecrafts.get(position).getOnoma());
        holder.eponymoTxt.setText(spacecrafts.get(position).getEponymo());
        holder.onomapatrosTxt.setText(spacecrafts.get(position).getOnomapatros());
    }

    @Override
    public int getItemCount() {
        return spacecrafts.size();
    }
}

```

10.1.40 MyHolder.java

```

package com.example.lefteris_kasos.firstaid.mRecycler;

import ...

/**
 * Created by Lefteris_Kasos on 04/07/2016.
 */
public class MyHolder extends RecyclerView.ViewHolder {

    TextView onomaTxt, eponymoTxt, onomapatrosTxt;

    public MyHolder(View itemView) {
        super(itemView);

        onomaTxt = (TextView) itemView.findViewById(R.id.onomaTxt);
        eponymoTxt = (TextView) itemView.findViewById(R.id.eponymoTxt);
        onomapatrosTxt = (TextView) itemView.findViewById(R.id.onomapatrosTxt);
    }
}

```

10.1.41 Spacecraft.java

```

package com.example.lefteris_kasos.firstaid.mRecycler;

/**
 * Created by Lefteris_Kasos on 04/07/2016.
 */
public class Spacecraft {
    int user_id;
    String onoma, eponymo, onomapatros;

    public String getEponymo() {
        return eponymo;
    }

    public void setEponymo(String eponymo) {
        this.eponymo = eponymo;
    }

    public String getOnoma() {
        return onoma;
    }

    public void setOnoma(String onoma) {
        this.onoma = onoma;
    }

    public String getOnomapatros() {
        return onomapatros;
    }

    public void setOnomapatros(String onomapatros) {
        this.onomapatros = onomapatros;
    }

    public int getUser_id() {
        return user_id;
    }

    public void setUser_id(int user_id) {
        this.user_id = user_id;
    }
}

```

10.2 ΚΩΔΙΚΕΣ-PHP

10.2.1 Register.php

```

<?php
$con = mysqli_connect("mysql.hostinger.gr", "u952517515_user1", "lef123", "u952517515_db1");

$name = $_POST["name"];
$age = $_POST["age"];
$username = $_POST["username"];
$password = $_POST["password"];

$stmt = mysqli_prepare($con, "INSERT INTO FirstAid (name, username, age, password) VALUES (?, ?, ?, ?)");
mysqli_stmt_bind_param($stmt, "ssis", $name, $username, $age, $password);
mysqli_stmt_execute($stmt);

$response = array();
$response["success"] = true;

echo json_encode($response);
?>

```

10.2.2 Login.php

```
<?php
$con = mysqli_connect("mysql.hostinger.gr", "u952517515_user1", "1ef123", "u952517515_db1");

$username = $_POST["username"];
$password = $_POST["password"];

$stmt = mysqli_prepare($con, "SELECT * FROM FirstAid WHERE username = ? AND password = ?");
mysqli_stmt_bind_param($stmt, "ss", $username, $password);
mysqli_stmt_execute($stmt);

mysqli_stmt_store_result($stmt);
mysqli_stmt_bind_result($stmt, $userID, $name, $username, $age, $password);

$response = array();
$response["success"] = false;

while(mysqli_stmt_fetch($stmt)){
    $response["success"] = true;
    $response["name"] = $name;
    $response["age"] = $age;
    $response["username"] = $username;
    $response["password"] = $password;
}

echo json_encode($response);
?>
```

10.2.3 aRegister.php

```

<?php
    $con = mysqli_connect("mysql.hostinger.gr", "u952517515_user1", "lef123", "u952517515_db1");

    $onoma = $_POST["onoma"];
    $eponymo= $_POST["eponymo"];
    $onomamitros = $_POST["onomamitros"];
    $onomapatros= $_POST["onomapatros"];
    $fulo = $_POST["fulo"];
    $hmerominiagenisis = $_POST["hmerominiagenisis"];
    $hlikia = $_POST["hlikia"];
    $xwrakatagwghs = $_POST["xwrakatagwghs"];
    $dieythinsi = $_POST["dieythinsi"];
    $staxydromikoskwdikas = $_POST["staxydromikoskwdikas"];
    $perioxh = $_POST["perioxh"];
    $polh = $_POST["polh"];
    $xwra = $_POST["xwra"];
    $thlefwno = $_POST["thlefwno"];
    $kinhto = $_POST["kinhto"];
    $arithmostaytohtas = $_POST["arithmostaytohtas"];
    $afm = $_POST["afm"];
    $amka = $_POST["amka"];
    $eksetaseiseginan = $_POST["eksetaseiseginan"];
    $khlh = $_POST["khlh"];
    $aimoroides = $_POST["aimoroides"];
    $ourhsh = $_POST["ourhsh"];
    $sakxarwdhdiavitis = $_POST["sakxarwdhdiavitis"];
    $petres = $_POST["petres"];
    $arthritida= $_POST["arthritida"];
    $problmataostwn = $_POST["problmataostwn"];
    $apoleiadakylwn= $_POST["apoleiadakylwn"];
    $ponoiplaths = $_POST["ponoiplaths"];
    $problmatagonatwn = $_POST["problmatagonatwn"];
    $problmatadiatrofhs = $_POST["problmatadiatrofhs"];
    $nautia= $_POST["nautia"];
    $problmataypnou = $_POST["problmataypnou"];
    $katathlipsi= $_POST["katathlipsi"];
    $apwleiamnhmhs= $_POST["apwleiamnhmhs"];
    $neurologikaproblmata = $_POST["neurologikaproblmata"];
    $apwleiasynidisis= $_POST["apwleiasynidisis"];
    $neuritida = $_POST["neuritida"];
    $paralysh = $_POST["paralysh"];
    $epilipsia= $_POST["epilipsia"];
    $gynaikologikaproblmata = $_POST["gynaikologikaproblmata"];
    $emmhnoorysh= $_POST["emmhnoorysh"];
    $eksetaseisprogrammatismenes= $_POST["eksetaseisprogrammatismenes"];
    $apotelesmata= $_POST["apotelesmata"];

```

```

$statement = mysqli_prepare($con, "INSERT INTO IatrikosFakelos (onoma, eponymo,
onomamitros, onomapatros,fulo,hmerominiagenisis,hlikia,xwratagwghs,dieythinsi,
taxydromikoskwdikas,perioxh,polh,xwra,thlefwno,kinhto,arithmostaytohtas,afm,
amka,eksetaseiseginan, khlh,aimoroides,ourhsh,sakxarwdhdiavitis,petres,
arthritida,problmataostwn,apoleiadakylwn,ponciplaths,problmatagonatwn,
problmatadiatrofhs,nautia,problmataypnou,katathlipsi,apwleiamnhmhs,
neurologikaproblmata,apwleiasynidisis,neuritida,paralysh,epilipsia,gynaikologikaproblmata,
emmhorysh,eksetaseisprogrammatismenes,apotelesmata)
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)");
mysqli_stmt_bind_param($statement, "ssssssssssssssssssssssssssssssssssssss",
$onoma, $eponymo, $onomamitros, $onomapatros,$fulo, $hmerominiagenisis,$hlikia,
$xwratagwghs, $dieythinsi, $taxydromikoskwdikas, $perioxh, $polh, $xwra,
$thlefwno, $kinhto, $arithmostaytohtas,$afm, $amka,$eksetaseiseginan,$khlh,
$aimoroides, $ourhsh,$sakxarwdhdiavitis, $petres, $arthritida,$problmatagonatwn ,
$apoleiadakylwn,$ponciplaths,$problmatagonatwn,$problmatadiatrofhs,$nautia,
$problmataypnou,$katathlipsi,$apwleiamnhmhs,$neurologikaproblmata,$apwleiasynidisis,
$neuritida,$paralysh,$epilipsia,$gynaikologikaproblmata,$emmhorysh,$eksetaseisprogrammatismenes,
$apotelesmata);
mysqli_stmt_execute($statement);

$response = array();
$response["success"] = true;

echo json_encode($response);
?>

```

10.2.4 nRegister.php

```

<?php
$con = mysqli_connect("mysql.hostinger.gr", "u952517515_user1", "lef123", "u952517515_db1");

$user_id=$_POST['Query'];
$sql="SELECT * FROM IatrikosFakelos WHERE user_id LIKE '%$user_id%'";
$query=mysqli_query($con,$sql);
if($query)
{
    while($row=mysqli_fetch_array($query))
    {
        $data[]=$row;
    }
    print(json_encode($data));
}
else
{
    echo('Not Found ');
}

$response = array();
$response["success"] = true;

echo json_encode($response);
?>

```

10.3 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

10.3.1 First Aid

```

-- phpMyAdmin SQL Dump
-- version 3.5.2.2
-- http://www.phpmyadmin.net
--
-- Host: localhost
-- Generation Time: Jul 04, 2016 at 07:49 PM
-- Server version: 10.0.22-MariaDB
-- PHP Version: 5.2.17

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Database: `u952517515_db1`
--

-----

--
-- Table structure for table `FirstAid`
--

CREATE TABLE IF NOT EXISTS `FirstAid` (
  `user_id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(16) COLLATE utf8_unicode_ci NOT NULL,
  `username` varchar(16) COLLATE utf8_unicode_ci NOT NULL,
  `age` tinyint(4) NOT NULL,
  `password` varchar(16) COLLATE utf8_unicode_ci NOT NULL,
  PRIMARY KEY (`user_id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=5 ;

--
-- Dumping data for table `FirstAid`
--

INSERT INTO `FirstAid` (`user_id`, `name`, `username`, `age`, `password`) VALUES
(3, 'LefterisKasos', 'lefteris', 27, 'lef123');

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```


10.3.2 IatrikosFakelos

```

-- phpMyAdmin SQL Dump
-- version 3.5.2.2
-- http://www.phpmyadmin.net
--
-- Host: localhost
-- Generation Time: Jul 04, 2016 at 06:50 PM
-- Server version: 10.0.22-MariaDB
-- PHP Version: 5.2.17

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Database: `u952517515_db1`
--
-----

--
-- Table structure for table `IatrikosFakelos`
--
CREATE TABLE IF NOT EXISTS `IatrikosFakelos` (
  `user_id` int(11) NOT NULL AUTO_INCREMENT,
  `onoma` varchar(20) COLLATE utf8_unicode_ci NOT NULL,
  `eponymo` varchar(20) COLLATE utf8_unicode_ci NOT NULL,
  `onomamitros` varchar(20) COLLATE utf8_unicode_ci NOT NULL,
  `onomapatros` varchar(20) COLLATE utf8_unicode_ci NOT NULL,
  `fulo` varchar(10) COLLATE utf8_unicode_ci NOT NULL,
  `hmerominiagenisis` varchar(20) COLLATE utf8_unicode_ci NOT NULL,
  `hlikia` varchar(4) COLLATE utf8_unicode_ci NOT NULL,
  `xwrakatagwghs` varchar(20) COLLATE utf8_unicode_ci NOT NULL,
  `dieythinsi` varchar(20) COLLATE utf8_unicode_ci NOT NULL,
  `taxydromikoskwdikas` varchar(20) COLLATE utf8_unicode_ci NOT NULL,
  `perioxh` varchar(20) COLLATE utf8_unicode_ci NOT NULL,
  `polh` varchar(20) COLLATE utf8_unicode_ci NOT NULL,
  `xwra` varchar(20) COLLATE utf8_unicode_ci NOT NULL,
  `thlefwno` varchar(10) COLLATE utf8_unicode_ci NOT NULL,
  `kinhto` varchar(20) COLLATE utf8_unicode_ci NOT NULL,
  `arithmostaytohtas` varchar(8) COLLATE utf8_unicode_ci NOT NULL,
  `afm` varchar(9) COLLATE utf8_unicode_ci NOT NULL,
  `amka` varchar(11) COLLATE utf8_unicode_ci NOT NULL,
  `eksetaseiseginan` varchar(1000) COLLATE utf8_unicode_ci NOT NULL,
  `khlh` varchar(15) COLLATE utf8_unicode_ci NOT NULL,
  `aimoroides` varchar(15) COLLATE utf8_unicode_ci NOT NULL,
  `ourhsh` varchar(15) COLLATE utf8_unicode_ci NOT NULL,
  `sakxarwdhdiavitis` varchar(15) COLLATE utf8_unicode_ci NOT NULL,
  `petres` varchar(15) COLLATE utf8_unicode_ci NOT NULL,
  `arthritida` varchar(15) COLLATE utf8_unicode_ci NOT NULL,

```

```

`problhmataostwn` varchar(15) COLLATE utf8_unicode_ci NOT NULL,
`apoleiadakylwn` varchar(15) COLLATE utf8_unicode_ci NOT NULL,
`ponoiplaths` varchar(15) COLLATE utf8_unicode_ci NOT NULL,
`problhmatagonatwn` varchar(15) COLLATE utf8_unicode_ci NOT NULL,
`problhmatadiatrofhs` varchar(15) COLLATE utf8_unicode_ci NOT NULL,
`nautia` varchar(15) COLLATE utf8_unicode_ci NOT NULL,
`problhmataypnou` varchar(15) COLLATE utf8_unicode_ci NOT NULL,
`katathlipsi` varchar(15) COLLATE utf8_unicode_ci NOT NULL,
`apwleiamnhmhs` varchar(15) COLLATE utf8_unicode_ci NOT NULL,
`neurologikaproblhmata` varchar(15) COLLATE utf8_unicode_ci NOT NULL,
`apwleiasynidisis` varchar(15) COLLATE utf8_unicode_ci NOT NULL,
`neuritida` varchar(15) COLLATE utf8_unicode_ci NOT NULL,
`paralysh` varchar(15) COLLATE utf8_unicode_ci NOT NULL,
`epilipsia` varchar(15) COLLATE utf8_unicode_ci NOT NULL,
`gynaikologikaproblhmata` varchar(15) COLLATE utf8_unicode_ci NOT NULL,
`emmhnyrysh` varchar(15) COLLATE utf8_unicode_ci NOT NULL,
`eksetaseisprogrammatismenes` varchar(1000) COLLATE utf8_unicode_ci NOT NULL,
`apotelesmata` varchar(1000) COLLATE utf8_unicode_ci NOT NULL,
PRIMARY KEY (`user_id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=28 ;

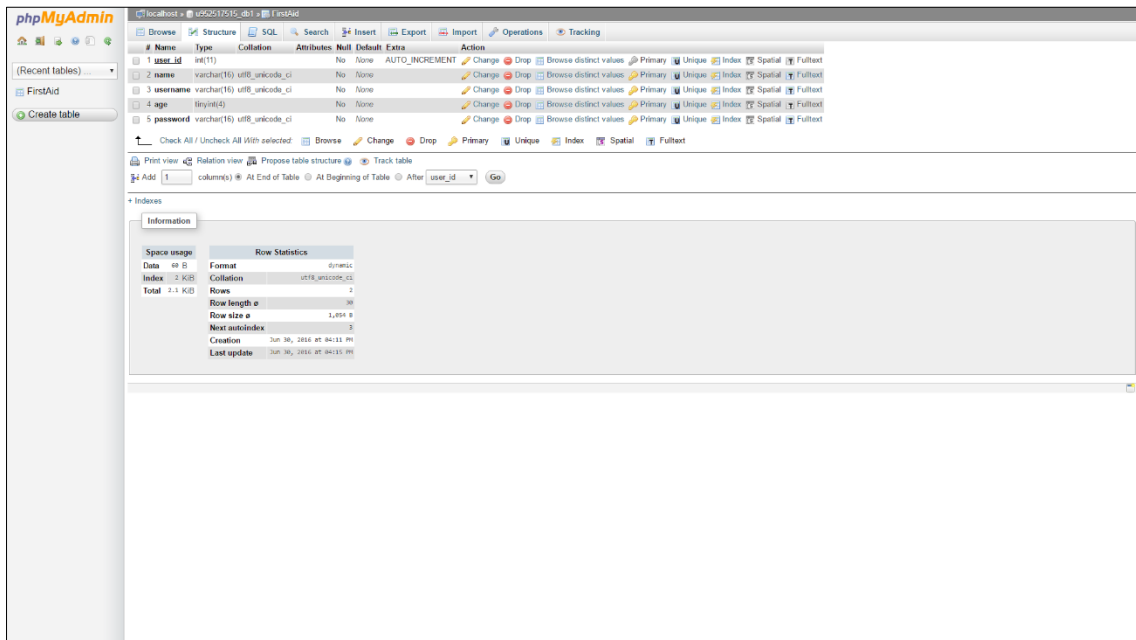
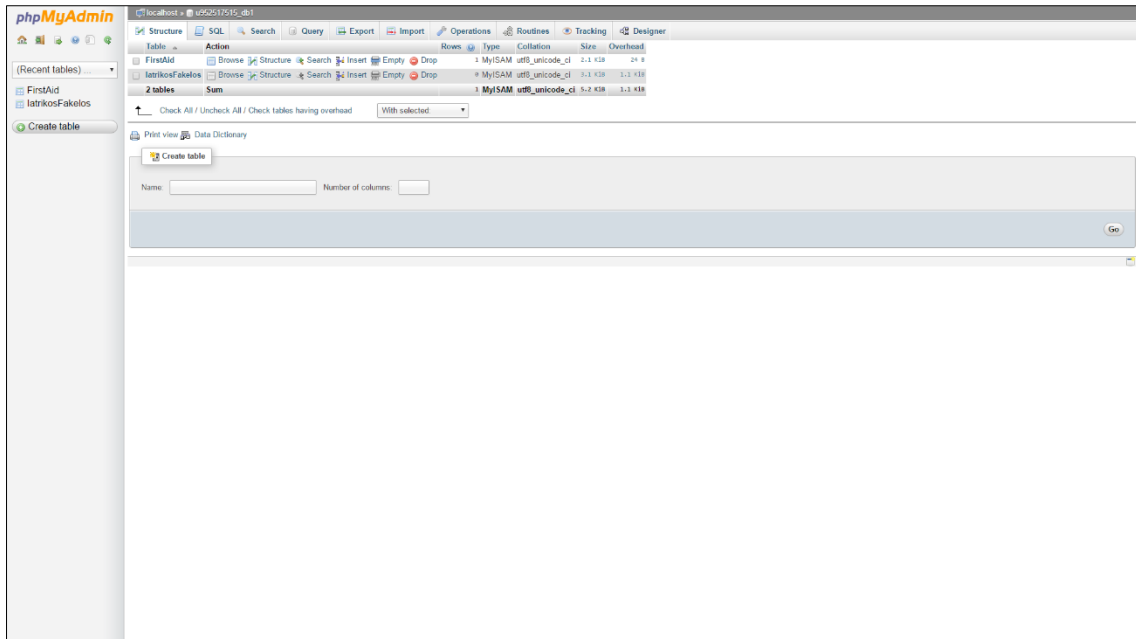
--
-- Dumping data for table `IatrikosFakelos`
--

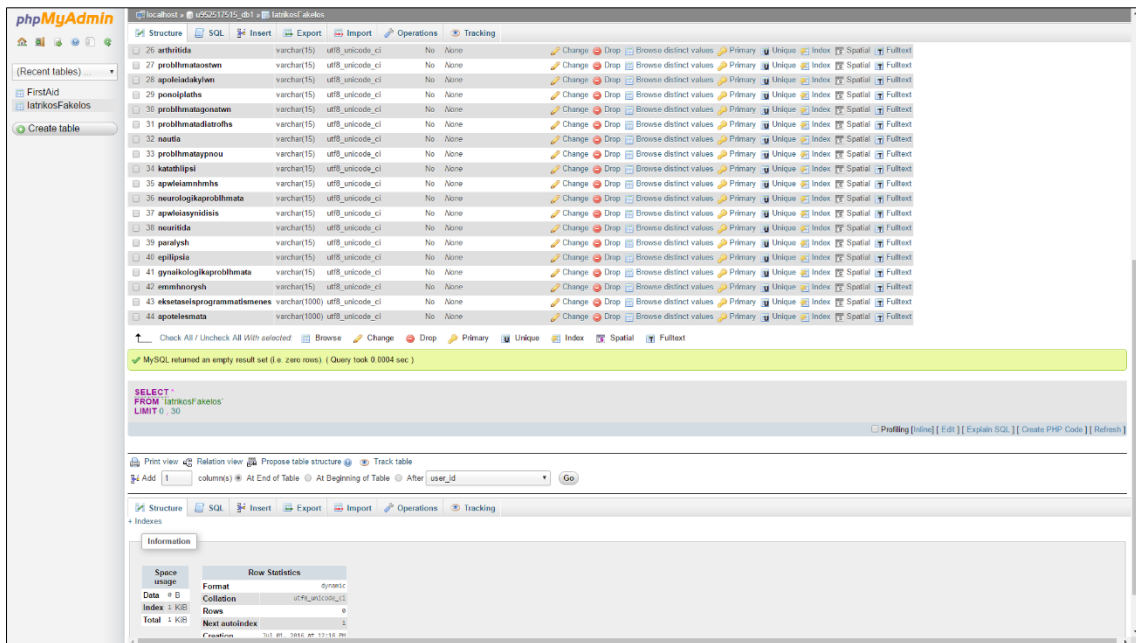
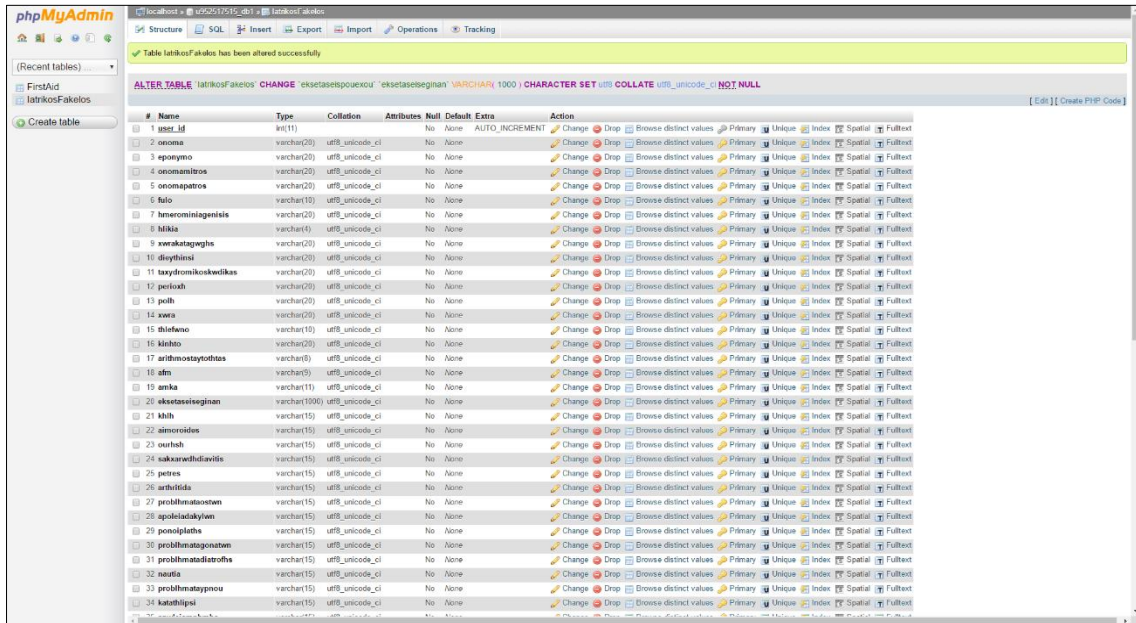
INSERT INTO `IatrikosFakelos` (`user_id`, `onoma`, `eponymo`, `onomamitros`,
`onomapatros`, `fulo`, `hmerominiagenisis`, `hlikia`, `xwrakatagwghs`, `dieythinsi`,
`taxydromikoskwdikas`, `perioxh`, `polh`, `xwra`, `thlefwno`, `kinhto`,
`arithmostaytohtas`, `afm`, `amka`, `eksetaseiseginan`, `khlh`, `aimoroides`,
`ourhsh`, `sakxarwdhdiavitis`, `petres`, `arthritida`, `problhmataostwn`,
`apoleiadakylwn`, `ponoiplaths`, `problhmatagonatwn`, `problhmatadiatrofhs`,
`nautia`, `problhmataypnou`, `katathlipsi`, `apwleiamnhmhs`, `neurologikaproblhmata`,
`apwleiasynidisis`, `neuritida`, `paralysh`, `epilipsia`, `gynaikologikaproblhmata`,
`emmhnyrysh`, `eksetaseisprogrammatismenes`, `apotelesmata`) VALUES
(22, 'leftferis', 'kasos', 'elh ', 'konstantinos', '', '', '',
'', '', '', '', '', '', '', '', '', '', '', '', '',
'', '', '', '', '', '', '', '', '', '', '', '',
'', '', ''),
(27, 'leftferis', 'kasos', '', 'kwstantinos', '', '', '',
'', '', '', '', '', '', '', '', '', '', '', '',
'', '', '', '', '', '', '', '', '', '', '', '',
'', '', '');

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

10.3.3 PhpMyAdmin





10.4 ΚΩΔΙΚΑΣ-ANDROID MANIFEST

10.4.1 Android Manifest

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.lefteris_kasos.firstaid">

    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.CALL_PHONE" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />

    <!--
        The ACCESS_COARSE/FINE_LOCATION permissions are not required to use
        Google Maps Android API v2, but you must specify either coarse or fine
        location permissions for the 'MyLocation' functionality.
    -->
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="First Aid"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".Psixoypo" />
        <activity android:name=".Protesb" />
        <activity android:name=".Epeigontaeafhmer" />
        <activity
            android:name=".Topothesisia"
            android:label="Map" />
        <activity android:name=".Istorikosfakelos" />
        <activity android:name=".Stoixeia" />
        <activity android:name=".Kardio" />
        <activity android:name=".Apofraksiaeragwou" />
        <activity android:name=".Pnigmos" />
        <activity android:name=".Aimoragia" />
        <activity android:name=".Rinoragia" />
        <activity android:name=".Trofikid" />
        <activity android:name=".Katagma" />
        <activity android:name=".Thermoplksia" />
        <activity android:name=".Ypothemia" />
        <activity android:name=".Lipothimia" />
        <activity android:name=".Axinos" />
        <activity android:name=".Fidi" />
        <activity android:name=".Tsimpouri" />
        <activity android:name=".Tsouxtra" />
        <activity android:name=".Methis" />
        <activity android:name=".Krpanikou" />
        <activity android:name=".Epilipikikr" />
    </application>

```

```

<!--
ATTENTION: This was auto-generated to add Google Play services to your project for
App Indexing. See https://g.co/AppIndexing/AndroidStudio for more information.
-->
<meta-data
    android:name="com.google.android.gms.version"
    android:value="9256000" />

<activity android:name=".RegisterActivity" />
<activity android:name=".UserAreaActivity" />
<activity
    android:name=".BasicActivity"
    android:label="BasicActivity"
    android:theme="@style/AppTheme" />

<!--
The API key for Google Maps-based APIs is defined as a string resource.
(See the file "res/values/google_maps_api.xml").
Note that the API key is linked to the encryption key used to sign the APK.
You need a different API key for each encryption key, including the release key that is used to
sign the APK for publishing.
You can define the keys for the debug and release targets in src/debug/ and src/release/.
-->
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="AIzaSyDLwQvAAv10gwzvuKYJQsLs7O-Lth4Tkug" />
</application>
</manifest>

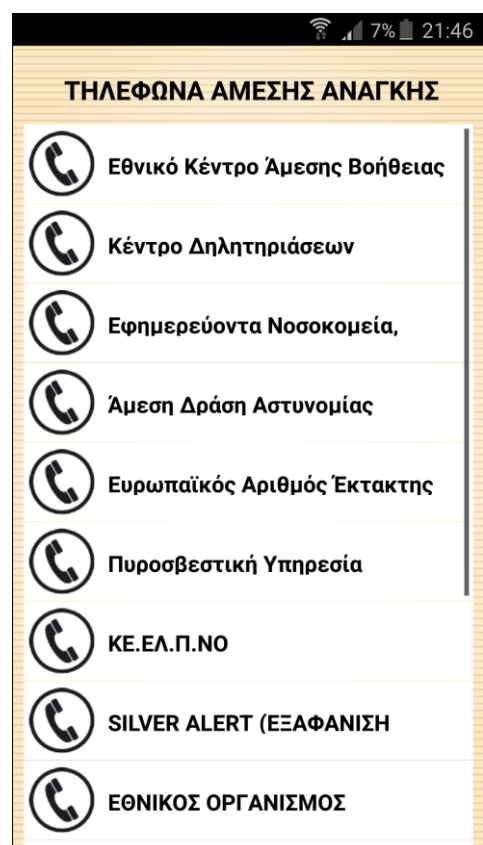
```

ΚΕΦΑΛΑΙΟ 11°

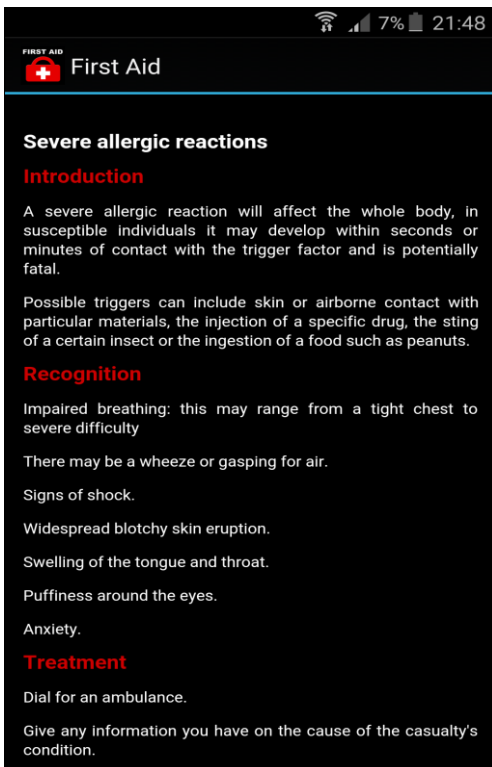
11.1 ΑΝΑΣΚΟΠΗΣΗ ΠΕΔΙΟΥ

Στο συγκεκριμένο κεφάλαιο θα κάνουμε μια ανασκόπηση σε παρεμφερή εφαρμογές, οι οποίες αποδείχτηκαν χρήσιμες καθώς μου έδωσαν αρκετές ιδέες έτσι ώστε να δημιουργήσω την δικιά μου εφαρμογή. Αρχικά συμβουλευτήκα την ελληνική εφαρμογή Πρώτων Βοηθειών από την Πανελλήνια ΜΚΟ ΑΓΩΓΗΣ ΥΓΕΙΑΣ “ΑΛΛΗΛΕΓΓΥΗ ΥΓΕΙΑΣ”. Στη συνέχεια διάφορες εφαρμογές από το Google Store και τέλος την ελληνική Εφαρμογή Πρώτων βοηθειών από τον Ελληνικό Ερυθρό Σταυρό. Πιο συγκεκριμένα παρακάτω θα δούμε κάποια στιγμιότυπα οθόνης των εφαρμογών.

1^η ΕΦΑΡΜΟΓΗ



2^η ΕΦΑΡΜΟΓΗ



3^η ΕΦΑΡΜΟΓΗ

