

Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης
Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Μηχανικών Πληροφορικής



Πτυχιακή Εργασία

Ανάπτυξη Android Εφαρμογής για τη Διαχείριση
Αγοραπωλησιών

Σπουδαστής:

Χανιωτάκης Ευάγγελος

Επιβλέπων Καθηγητής:

Παπαδάκης Νικόλαος

Ηράκλειο Κρήτης
2017

Περίληψη

Η παρούσα πτυχιακή εργασία έχει ως αντικείμενο την μελέτη και περιγραφή τεχνικών ανάπτυξης εφαρμογών για smartphones με λειτουργικό σύστημα Android, χρησιμοποιώντας το περιβάλλον του Android Studio, τα Android API's (Application Programming Interface) και το SDK (System Development Kit). Αρχικά θα εξηγήσουμε το λόγο για τον οποίο επιλέξαμε το συγκεκριμένο λειτουργικό σύστημα και αργότερα θα κάνουμε μια σύντομη ανασκόπηση στα ενδότερα στοιχεία που το απαρτίζουν.

Θα κάνουμε μία ιστορική αναδρομή στις εκδόσεις και τα χαρακτηριστικά του Android τα οποία εξελίσσονται ραγδαία με την πάροδο του χρόνου προκειμένου να διευκολύνουν τη χρήση του λειτουργικού συστήματος και να το κάνουν πιο αγαπητό στο μέσο χρήστη. Εν συνεχεία εξετάζουμε όλες τις παραμέτρους του και τα πεδία εφαρμογής τους, χρησιμοποιώντας την αντικειμενοστραφή γλώσσα προγραμματισμού Java.

Στόχος της εφαρμογής είναι να διαχειρίζεται τα προϊόντα τα οποία εκθέτει ένας χρήστης προς πώληση ή τα προϊόντα τα οποία αναζητεί ένας χρήστης για αγορά. Δηλαδή η εφαρμογή είναι το μέσο που θα χρησιμοποιούν οι χρήστες προκειμένου να εντοπίσουν αυτό που ψάχνουν κοιτάζοντας μία γκάμα από προϊόντα χωρισμένα σε κατηγορίες ή αντίστοιχα να πουλήσουν κάτι και κατ' επέκταση να έρθουν σε επικοινωνία με τον πωλητή ή τον αγοραστή αντίστοιχα ενός προϊόντος, για να πάρουν παραπάνω πληροφορίες κλπ.

Abstract

The present paper aims to study, develop and show how to create an application for Android based smartphones. For the purpose of the paper an android application was developed using the platform Android Studio. For the beginning we will take a look at the history of Android, how it started and some important characteristics. After that we will move deeper inside and we will use Java which is Object Oriented programming language, in coordination with xml for the visual part of the application. Finally we will create and launch our application.

The application aims to load products that people want to sell or buy. It is the main way to get information about the products that they are looking for and the dealer, in order to get in contact.

Eventually users can claim a product or a service that was being for sale by a dealer, either they can sale a product or a service that was being asked by another dealer inside the application.

Περιεχόμενα

Περίληψη.....	2
Abstract.....	3
Περιεχόμενα.....	4
Κεφάλαιο 1.....	6
1.1 Τι είναι το Android.....	6
1.1.1 Χαρακτηριστικά του Android.....	8
1.2 Εξέλιξη του Android.....	9
1.2.1 Android 1.0 (API Level 1 Android Version 1.0).....	10
1.2.2 Android 1.1 (API Level 2 Android Version 1.1).....	10
1.2.3 Android Cupcake (API Level 3 Android Version 1.5).....	11
1.2.4 Android Donut (API Level 4 Android Version 1.6).....	11
1.2.5 Android Éclair (API Level 5 & 6 & 7 Android Versions 2.0 & 2.1).....	12
1.2.6 Android Froyo (API Level 8 Android Version 2.2 – 2.2.3).....	12
1.2.7 Android Gingerbread (API Level 9 & 10 Android Version 2.3 - 2.3.7).....	13
1.2.8 Android Honeycomb (API Level 11 & 12 & 13 Android Versions 3.0 & 3.1-3.2.6).....	14
1.2.9 Android Ice-cream Sandwich (API Level 14 & 15 Android Version 4.0 - 4.0.4).....	14
1.2.10 Android Jellybean (API Level 16 & 17 & 18 Android Versions 4.1 & 4.2 & 4.3).....	15
1.2.11 Android KitKat (API Level 19 & 20 Android Version 4.4).....	16
1.2.12 Android Lollipop (API Level 21 & 22 Android Version 5.0 – 5.1.1).....	18
1.2.13 Android Marshmallow (API Level 23 Android Version 6.0 – 6.0.1).....	18
1.2.14 Android Nougat (API Level 24 & 25 Android Version 7.0 – 7.1.2).....	19
1.2.15 Android O (API Level 26 Android Version 8.0).....	20
1.3 Γιατι χρησιμοποιούμε Android.....	21
1.4 Δομή του Android.....	23
1.4.1 Πυρήνας (Linux Kernel).....	24
1.4.2 Βιβλιοθήκες.....	25
1.4.3 Πλαίσιο εφαρμογών (Application Framework).....	26
1.4.4 Εφαρμογές (Applicatios).....	26

1.5	Δομικά στοιχεία εφαρμογών Android.....	26
1.5.1	Δραστηριότητα (Activity).....	27
1.5.2	Αναλλαγή μηνυμάτων προθέσεων (Intents).....	27
1.5.3	Κύκλος ζωής μιας δραστηριότητας.....	28
1.5.4	Τεμάχια (Fragments).....	30
1.5.5	Υπηρεσίες (Services).....	31
1.5.6	Πάροχος περιεχομένου (Content Provider).....	31
1.5.7	Παραλήπτες μηνυμάτων πανεκπομπών (Broadcast Receivers).....	32
1.5.8	Μανιφέστο Εφαρμογών (Android Manifest).....	32
1.6	Εσωτερικό Εφαρμογής Android.....	33
1.6.1	Φάκελος src.....	34
1.6.2	Φάκελος res.....	34
1.6.3	Gradle Scripts.....	35
	Κεφάλαιο 2.....	36
2.1	Εγκατάσταση Λογισμικού.....	36
2.2	Διαδικασία Αποσφαλμάτωσης(Debugging).....	37
2.3	Δημοσίευση Εφαρμογής.....	37
2.4	Android System Development Kit.....	38
2.5	Android Virtual Device Manager(AVD).....	39
2.6	Κατευθυντήριες γραμμές σχεδιασμού Android (Design Guidelines).....	41
2.6.1	Υποστήριξη οθονών.....	42
2.7	Android Crash Report Manager – Android Logcat.....	44
2.8	Διαχείριση Δεδομένων.....	45
2.8.1	Κοινόχρηστες προτιμήσεις – Shared Preferences.....	46
2.8.2	Βάση δεδομένων SQLite.....	47
2.8.2.1	Δημιουργία Βάσης SQLite.....	47
2.8.2.2	Διαχείριση δεδομένων σε βάση SQLite.....	48
2.9	Παράθυρα ειδοποιήσεων (Alert Dialogs) και σύντομα μηνύματα (Toasts).....	49
	Κεφάλαιο 3.....	50
3.1	Περιγραφή της εφαρμογής.....	50
3.2	Διαδικασία χρήσης και περιεχόμενο της εφαρμογής.....	50

3.3 Διεκδίκηση προϊόντος με προκαταβολή μέσω Paypal.....	52
3.4 Μεταφορά δεδομένων χρησιμοποιώντας Android Volley.....	53
Κεφάλαιο 4.....	55
4.1 Ανάλυση απαιτήσεων για την δημιουργία της εφαρμογής.....	55
4.2 Ανάλυση της εφαρμογής με διαγράμματα UML.....	55
4.2.1 Διάγραμμα περιπτώσεων χρήσης.....	56
4.3 Δημιουργία Project στο Android Studio.....	57
4.4 Δημιουργία και ανάλυση περιεχομένου εφαρμογής.....	58
Κεφάλαιο 5.....	66
5.1 Συμπεράσματα.....	66
5.2 Μελλοντική Εξέλιξη.....	67
Κεφάλαιο 6.....	67
6.1 Βιβλιογραφία.....	67

Κεφάλαιο 1

1.1 Τι είναι το Android

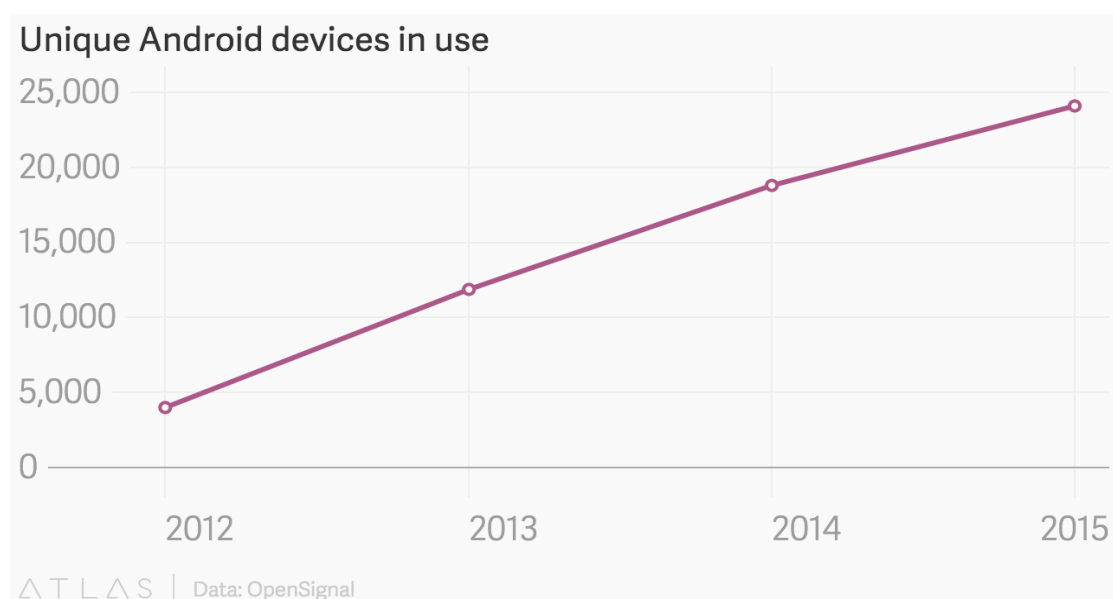
Το **Android** είναι λειτουργικό σύστημα για έξυπνες συσκευές κινητής τηλεφωνίας το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά αναπτύχθηκε από την **Google** [28] και αργότερα από την **Open Handset Alliance** [27]. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google. Το Android είναι κατά κύριο λόγο σχεδιασμένο για συσκευές με οθόνη αφής, όπως τα έξυπνα τηλέφωνα, τα τάμπλετ, με διαφορετικό περιβάλλον χρήσης για τηλεοράσεις (Android TV), αυτοκίνητα (Android Auto) και ρολόγια χειρός (Android Wear). Παρόλο που έχει αναπτυχθεί για συσκευές με οθόνη αφής, έχει χρησιμοποιηθεί σε κονσόλες παιχνιδιών, ψηφιακές φωτογραφικές μηχανές, συνηθισμένους Η/Υ και σε άλλες ηλεκτρονικές συσκευές. Το λογότυπο του Android είναι

ένα πράσινο ανθρωποειδές ρομπότ, όπως αυτό της φωτογραφίας που απεικονίζεται παρακάτω.



Εικόνα 1: Android Λογότυπο.

Τα τηλέφωνα Android πρώτης γενιάς εμφανίστηκαν τον Οκτώβριο του 2008. Σύμφωνα με την Gartner [29] (αμερικανική ερευνητική και συμβουλευτική εταιρεία που παρέχει πληροφόρηση σχετικά με την τεχνολογία της πληροφορίας για τους ηγέτες πληροφορικής και άλλους επιχειρηματίες που βρίσκονται σε ολόκληρο τον κόσμο) οι πωλήσεις τηλεφώνων Android στη Βόρεια Αμερική αυξήθηκαν κατά 707% μέσα στο πρώτο τρίμηνο του 2010 σε σχέση με το προηγούμενο έτος. Μέχρι τον Μάρτιο του 2011, μία μελέτη της Nielsen (παγκόσμια εταιρεία πληροφόρησης, δεδομένων και μετρήσεων) έδειξε ότι το Android κατείχε το 37% της αγοράς τηλεφώνων smartphone στις Η.Π.Α., σε σύγκριση με το 27% του iPhone της Apple και το 22% του Blackberry. Τον Αύγουστο του 2010, ενεργοποιούνταν περισσότερα των 200.000 Android smartphones κάθε μέρα, ενώ πριν από δύο μήνες ενεργοποιούνταν μόνο 100.000 κάθε μέρα. Σήμερα υπάρχουν περισσότερες από 25.000 διαφορετικές συσκευές Android σε όλο τον κόσμο.



Εικόνα 2: Διάγραμμα μοναδικών συσκευών Android που χρησιμοποιούνται (Όπως αναφέρεται στο [23])

1.1.1 Χαρακτηριστικά του Android

Τα τωρινά χαρακτηριστικά και οι λειτουργίες που απαρτίζουν το Android είναι:

- **Λειτουργίες Οθόνης**
Η πλατφόρμα είναι προσαρμόσιμη σε πολλές αναλύσεις οθόνης (από VGA μέχρι 4K), δισδιάστατες ψηφιακές γραφικές βιβλιοθήκες, τρισδιάστατα γραφικά βασισμένα στην OpenGL ES 3.0+ έκδοση χαρακτηριστικών, καθώς και παραδοσιακές απεικονίσεις οθόνης «έξυπνων» συσκευών κινητής τηλεφωνίας.
- **Αποθήκευση Δεδομένων**
Χρήση βάσης δεδομένων SQLite για τις ανάγκες αποθήκευσης.
- **Συνδεσιμότητα**
Το Android υποστηρίζει τεχνολογίες συνδεσιμότητας συμπεριλαμβανομένου GSM / EDGE / UMTS / HSPA / HSPA+ / LTE, 3G, 4G, CDMA, EV-DO, Bluetooth, NFC, και Wi-Fi.
- **Αποστολή μηνυμάτων**
SMS και MMS είναι οι διαθέσιμοι τρόποι ανταλλαγής μηνυμάτων.
- **Περιήγηση στον Ιστό**
Για την περιήγηση στον ιστό το Android διαθέτει φυλλομετρητή βασισμένο στην ανοιχτή τεχνολογία WebKit. Και άλλοι φυλλομετρητές είναι διαθέσιμοι από το Google Play.
- **Υποστήριξη Java**
Λογισμικό γραμμένο στην Java είναι δυνατόν να μεταγλωτιστεί και να εκτελεστεί στην εικονική μηχανή Dalvik/ART, η οποίες αποτελούν εξειδικευμένη υλοποίηση εικονικής μηχανής σχεδιασμένης για χρήση σε φορητές συσκευές, παρόλο που δεν είναι πρότυπη εικονική μηχανή Java.
- **Υποστήριξη Πολυμέσων**
Το λειτουργικό σύστημα Android υποστηρίζει τις ακόλουθα μορφές ήχου, στατικής και κινούμενης εικόνας: H.263, H.264 (σε 3GP ή MP4 container), MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, MP3, MIDI, OGG Vorbis, WAV, JPEG, PNG, GIF, BMP.
- **Επιπλέον υποστήριξη υλικού**
Το λειτουργικό Android μπορεί να συνεργαστεί με κάμερες στατικής ή κινούμενης εικόνας, οθόνες αφής, GPS, αισθητήρες επιτάχυνσης, μαγνητόμετρα, δισδιάστατους καθώς και τρισδιάστατους επιταχυντές γραφικών.

- **Περιβάλλον Ανάπτυξης Λογισμικού**
Περιλαμβάνει ένας προσομοιωτή συσκευής, εργαλεία για διόρθωση σφαλμάτων, μνήμη και εργαλεία ανάλυσης της απόδοσης του εκτελέσιμου λογισμικού καθώς και ένα επιπρόσθετο για το Android Studio.
- **Αγορά και Εγκατάσταση Εφαρμογών**
Το Google Play είναι ένας κατάλογος εφαρμογών που μπορούν να μεταφορτωθούν και εγκατασταθούν στην συσκευή άμεσα μέσω ασύρματων καναλιών, χωρίς την χρήση υπολογιστή. Αρχικά μόνο δωρεάν εφαρμογές ήταν δυνατόν να εγκατασταθούν. Εφαρμογές επί πληρωμή ήταν μετέπειτα διαθέσιμες στο Google play στις ΗΠΑ ύστερα από τις 19 Φεβρουαρίου 2009.
- **Οθόνη Αφής Πολλαπλών Σημείων**
Το λειτουργικό σύστημα Android υποστηρίζει οθόνες αφής πολλαπλών σημείων.

1.2 Εξέλιξη του Android

Η ιστορία εκδόσεων Android, του λειτουργικού συστήματος των έξυπνων κινητών ξεκίνησε με την κυκλοφορία του Android beta το Νοέμβριο του 2007. Η πρώτη εμπορική έκδοση ήταν το Android 1.0 που κυκλοφόρησε το Σεπτέμβριο του 2008. Το Android είναι υπό συνεχή ανάπτυξη από την Google και την Open Handset Alliance (OHA), και έχουν γίνει μια σειρά από ενημερώσεις στην λειτουργία του συστήματος από την αρχική κυκλοφορία του.

Από τον Απρίλιο του 2009, οι εκδόσεις του Android έχουν θέμα από την ζαχαροπλαστική στην κωδική ονομασία τους, και κυκλοφόρησαν σε αλφαβητική σειρά, εξαιρουμένων των εκδόσεων 1.0 και 1.1, που δεν τέθηκαν υπό συγκεκριμένα κωδικά ονόματα. Παρακάτω βλέπουμε έναν πίνακα για να κατανοήσουμε την εξέλιξη πιο αναλυτικά.

Κωδικό όνομα ⇅	Νούμερο έκδοσης ⇅	Ημερομηνία αρχικής κυκλοφορίας ⇅	Επίπεδο API ⇅
N/A	1.0	23 Σεπτεμβρίου 2008	1
	1.1	9 Φεβρουάριου 2009	2
Cupcake	1.5	27 Απριλίου 2009	3
Donut	1.6	15 Σεπτεμβρίου 2009	4
Eclair	2.0 – 2.1	26 Οκτωβρίου 2009	5–7
Froyo	2.2 – 2.2.3	20 Μαΐου 2010	8
Gingerbread	2.3 – 2.3.7	6 Δεκεμβρίου 2010	9–10
Honeycomb	3.0 – 3.2.6	22 Φεβρουάριου 2011	11–13
Ice Cream Sandwich	4.0 – 4.0.4	18 Οκτωβρίου 2011	14–15
Jelly Bean	4.1 – 4.3.1	9 Ιουλίου 2012	16–18
KitKat	4.4 – 4.4.4	31 Οκτωβρίου 2013	19–20
Lollipop	5.0 – 5.1.1	12 Νοεμβρίου 2014	21–22
Marshmallow	6.0 – 6.0.1	5 Οκτωβρίου 2015	23
Nougat	7.0 - 7.1.1	22 Αυγούστου 2016	24
Oreo	8.0	21 Αυγούστου 2017	26

Εικόνα 3: Εκδόσεις του λειτουργικού συστήματος Android (Όπως αναφέρεται στο [20])

1.2.1 Android 1.0 (API Level 1 Android Version 1.0)

Το Android σχεδιάστηκε αρχικά ως μια διεπαφή αφής για κάμερες, πριν η Google δει τις δυνατότητές του και το βγάλει στην αγορά. Η πρώτη εμπορικά διαθέσιμη έκδοση Android για κινητά τηλέφωνα κυκλοφόρησε τον Σεπτέμβριο του 2008 στο HTC Dream handset. Τα ονόματα με γεύση καραμελών δεν είχαν ακόμη εισαχθεί, οπότε αυτή η έκδοση ονομαζόταν τυπικά Android 1.0 (όπως αναφέρει και το [20]).

Τα χαρακτηριστικά του ήταν βασικά, αλλά πολλά από αυτά εξακολουθούν να υπάρχουν σήμερα. Υπήρχε ένα απλό πρόγραμμα περιήγησης ιστού και υποστήριξη για το Gmail, το Ημερολόγιο Google, οι Χάρτες Google και το YouTube, καθώς και η εφαρμογή Messenger, Google Talk.

1.2.2 Android 1.1 (API Level 2 Android Version 1.1)

Μια πολύ απρόσμενη ενημέρωση στο Android 1.0, προσθέτοντας μερικές απλές λειτουργίες στις υπάρχουσες εφαρμογές, όπως η εμφάνιση σχολίων στις αναζητήσεις επιχειρήσεων στο Google Maps (όπως αναφέρει το [20]).

1.2.3 Android Cupcake (API Level 3 Android Version 1.5)



Εικόνα 4: Android Cupcake

Επιτέλους, ένα γλυκό όνομα για ένα όλο και πιο γλυκό κινητό λειτουργικό σύστημα. Το Android 1.5 Cupcake έφερε υποστήριξη widget, κινούμενες μεταβάσεις κατά την παράκαμψη μέσω των επιτραπέζιων υπολογιστών, τη δυνατότητα αυτόματης περιστροφής της οθόνης όταν γυρνάμε το τηλέφωνό και μια κινούμενη εικόνα animation κατά το ξεκίνημα του λειτουργικού όταν ενεργοποιούμε τη συσκευή.

1.2.4 Android Donut (API Level 4 Android Version 1.6)



Εικόνα 5: Android Donut

Το Σεπτέμβριο του 2009 είχαμε τον διάδοχο του Cupcake, με το παρατσούκλι Donut. Το Donut μας έδωσε βελτιωμένη λειτουργικότητα αναζήτησης, πιο γρήγορη εμπειρία και υποστήριξη για οθόνες 480x800 pixel. Επίσης κάποια χαρακτηριστικά ακόμη που προστέθηκαν (όπως αναφέρει το [14],[18],[21]) είναι:

- Εικονικό πληκτρολόγιο οθόνης
- Camcorder mode για εγγραφή και προβολή video
- Stereo Bluetooth
- Εισαγωγή widgets και φακέλων στην αρχική οθόνη
- Αντιγραφή/Επικόλληση και αναζήτηση στον browser
- Άμεση μεταφόρτωση στο Youtube

1.2.5 Android Éclair (API Level 5 & 6 & 7 Android Versions 2.0 & 2.1)



Εικόνα 6: Android Éclair

Τον Οκτώβριο του 2009 έκανε την εμφάνιση του το Android Éclair με εξέλιξη Level στο 2.0 φέρνοντας αλλαγές (όπως αναφέρει το [14],[18],[21]):

- Πολλαπλούς λογαριασμούς χρήστη
- Υποστήριξη λήψης και αποστολής email
- Ανεπτυγμένη κάμερα με χρήση flash και ψηφιακό zoom
- Bluetooth 2.1
- Ανεπτυγμένο πληκτρολόγιο με προσαρμόσιμο λεξικό και προτάσεις από ονόματα επαφών
- Live wallpapers
- Ομιλία σε κείμενο (Speech to text) σε οποιοδήποτε πεδίο κειμένου

1.2.6 Android Froyo (API Level 8 Android Version 2.2 - 2.2.3)



Εικόνα 7: Android Froyo

Το Android 2.2 (που καλείται επίσης Froyo και εκδόθηκε τον Μάιο του 2010) περιελάμβανε αρκετά νέα χαρακτηριστικά και βελτιώσεις. Κάποια από αυτά (όπως αναφέρει το [14],[18],[21]) ήταν:

- Βελτιωμένη μνήμη και απόδοση
- Αυτόματη ανακάλυψη η οποία επιτρέπει σε χρήστες να εισάγουν ένα όνομα χρήστη και ένα κωδικό πρόσβασης, ώστε να συγχρονίζουν γρήγορα τους λογαριασμούς τους.
- Ανεπτυγμένη υποστήριξη Microsoft Exchange: Πολιτικές Ασφαλείας, Global Διευθύνσεις, συγχρονισμός ημερολογίου, απομακρυσμένη διαγραφή και καθαρισμός
- Υποστήριξη Tethering και λειτουργίας Hotspot
- Αναβάθμιση του Android Market εισάγοντας την δυνατότητα εγκατάστασης εφαρμογών σε εξωτερική κάρτα SD
- Υποστήριξη Bluetooth για φωνητική πληκτρολόγηση και διαμοιρασμό επαφών(contact sharing)
- Βελτίωση του προγράμματος πλοήγησης με υποστήριξη HTML5, η οποία περιλαμβάνει χαρακτηριστικά όπως αναπαραγωγή βίντεο και μεταφορά και απόθεση. Παρ' αυτά το πρόγραμμα πλοήγησης υποστηρίζει επίσης το Flash.
- Πληκτρολόγιο σε πολλές γλώσσες

1.2.7 Android Gingerbread (API Level 9 & 10 Android Version 2.3 - 2.3.7)



Εικόνα 8: Android Gingerbread

Το Android 2.3 (Gingerbread) εκδόθηκε το Δεκέμβριο του 2010 (με το Android 2.3.3 -μια ελάχιστη ενημέρωση- που εκδόθηκε τον Φεβρουάριο του 2011), πρόσθεσε περισσότερες βελτιώσεις χρήστη, όπως ένα ανασχεδιασμένο πληκτρολόγιο, βελτιωμένες δυνατότητες πλοήγησης, αυξημένη αποδοτικότητα ισχύος και πολλά περισσότερα. Παρακάτω βλέπουμε (όπως αναφέρει το [14],[18],[21]) μερικά από τα βασικά νέα χαρακτηριστικά και τις βασικές νέες ενημερώσεις.

- Καλύτερη διαχείριση ισχύος
- Συντόμευση Manage Applications όπου μέσα στο μενού Options της αρχικής οθόνης επιτρέπει στους χρήστες να προβάλλουν όλες τις εφαρμογές που εκτελούνται.
- Επικοινωνίες κοντινού πεδίου (Near Field Communication) **NFC**, το οποίο είναι ένα πρότυπο ασύρματων επικοινωνιών μικρής απόστασης, που επιτρέπει την επικοινωνία ανάμεσα σε δύο συσκευές.
- Βελτιωμένη λειτουργικότητα Copy και Paste

- Εφαρμογή Downloads όπου οι χρήστες μπορούν να προσπελάσουν αρχεία που φορτώθηκαν από e-mail, internet κ.λπ μέσω αυτής
- Εναλλαγή σε μπροστινή και πίσω κάμερα μέσα από την εφαρμογή της κάμερας

1.2.8 Android Honeycomb (API Level 11 & 12 & 13 Android Versions 3.0 & 3.1-3.2.6)

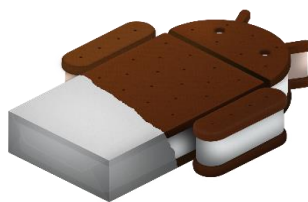


Εικόνα 9: Android Honeycomb

Τον φεβρουάριο του 2011 έκανε την εμφάνιση της μια ιδιαίτερη έκδοση που απευθυνόταν ειδικά για ταμπλέτες. Το Android 3.0 (Honeycomb) περιλαμβάνει βελτιώσεις στην διεπαφή χρήστη, όπως είναι ένα ανασχεδιασμένο πληκτρολόγιο για αποδοτικότερη πληκτρολόγηση, μία οπτικά ελκυστική 3D διεπαφή χρήστη, γραμμές συστήματος και ενέργειας για ευκολότερη πλοήγηση και πολλές άλλες που θα δούμε παρακάτω. Επίσης παρέχει στους προγραμματιστές νέα εργαλεία ώστε να βελτιστοποιούν εφαρμογές για συσκευές με μεγαλύτερες οθόνες. Κάποια λοιπόν απ' τα επιπλέον χαρακτηριστικά (όπως αναφέρει το [14],[18],[21]) που προστέθηκαν είναι:

- Υποστήριξη γραφικών 3D
- Ιδιωτική πλοήγηση - Private Browsing
- Δυνατότητα Video Chat με την χρήση της υπηρεσίας Google Talk
- Bluetooth Tethering
- Υποστήριξη σύνδεσης περιφερειακών συσκευών όπως πληκτρολόγια και Gamepads
- Παραμετροποιήσιμα Widgets
- Δυνατότητα προβολής εικόνων από το άλμπουμ σε πλήρη οθόνη
- Διορθώσεις σφαλμάτων και σταθερότητας συστήματος

1.2.9 Android Ice-cream Sandwich (API Level 14 & 15 Android Version 4.0 - 4.0.4)



Εικόνα 10: Android Ice cream Sandwich

Το **Android 4.0 - 4.0.4 "Ice Cream Sandwich"** έρχεται τον Οκτώβριο του 2011 ως μία ιδιαίτερα ανανεωμένη έκδοση του λειτουργικό συστήματος για κινητά Android με πάρα πολλές καινούργιες δυνατότητες. Με επανασχεδιασμένο User Interface, πιο γρήγορο και αποδοτικό, εισάγει νέα εικονικά κουμπιά που αντικαθιστούν τα αφής ή φυσικά κουμπιά που υπήρχαν στις εκάστοτε συσκευές. Οι αλλαγές που έφερε είναι πολλές όπως θα δούμε παρακάτω (όπως αναφέρει το [14],[18],[21]):

- Δημιουργία φακέλων απλά τραβώντας την μια εφαρμογή επάνω σε μια άλλη
- Διαχωρισμός μεταξύ εφαρμογών και widgets
- Υποστήριξη λειτουργίας pinch-to-zoom στην εφαρμογή ημερολογίου
- Επανασχεδιασμός της εφαρμογής Gmail και παράλληλα προσθέτοντας την λειτουργία offline search καθώς επίσης την λειτουργία swipe μεταξύ email μηνυμάτων
- Επανασχεδιασμός της εφαρμογής Google Chrome η οποία πλέον υποστηρίζει καρτέλες εκτός σύνδεσης, μέχρι 16 συνεχόμενες καρτέλες ανοικτές και συγχρονισμό σελιδοδεικτών
- Διορθώσεις σφαλμάτων πληκτρολογίου οθόνης
- Προσαρμόσιμη οθόνη κλειδώματος, υποστήριξη launcher
- Εισαγωγή γραμματοσειράς Roboto
- Νέα λειτουργία διαγραφής με κύλιση
- Αναπτυγμένη αναγνώριση ομιλίας και βελτιωμένη λειτουργία copy και paste.
- Ξεκλείδωμα οθόνης με λειτουργία αναγνώρισης προσώπου
- Μετρητής δεδομένων κινητής τηλεφωνίας
- Λειτουργία απόκρυψης επιλεγμένων εφαρμογών από το χρήστη
- Δυνατότητα για τερματισμό λειτουργίας εφαρμογών που χρησιμοποιούν πόρους υλικού
- Πρόσθετες λειτουργίες της εφαρμογής της κάμερας: zero shutter lag, continuous focus, zoom while recording, time lapse settings, panorama photos, 1080p recording
- Δυνατότητα αναγνώρισης προσώπου (Face recognition) μέσω της κάμερας με ενσωματωμένο Photo Editor.
- Νέος σχεδιασμός Gallery app (Εφαρμογή άλμπουμ)

- Δυνατότητα αποστολής άμεσου μηνύματος στην περίπτωση που επιθυμεί ο χρήστης να απορρίψει μία κλήση
- Λειτουργία Android Beam, μία ιδιότητα του NFC για αποστολή και λήψη αρχείων όταν οι συσκευές βρίσκονται σε κοντινή απόσταση

1.2.10 Android Jellybean (API Level 16 & 17 & 18 Android Versions 4.1 & 4.2 & 4.3)



Εικόνα 11: Android Jelly Bean

Το λειτουργικό σύστημα Jelly Bean εμφανίζεται τον Ιούλιο του 2012 παρουσιάζοντας τις καλύτερες επιδόσεις όσον αφορά την ταχύτητα και τη λειτουργικότητα των κινητών συσκευών. Μερικά χαρακτηριστικά του (όπως αναφέρει το [14],[18],[21]) είναι τα εξής:

- Υποστήριξη του API OpenGL ES 3.0 για καλύτερη χρήση της GPU και την παροχή καλύτερων γραφικών και καλύτερων επιδόσεων σε παιχνίδια
- Υποστήριξη τεχνολογίας Virtual surround στον ήχο
- Βελτιωμένες λειτουργίες στις λήψεις φωτογραφιών
- Ξεκλείδωμα με αναγνώριση προσώπου
- Υποστήριξη πολλαπλών χρηστών με διαφορετικές διαμορφώσεις για τον καθένα
- Ενσωματωμένη υπηρεσία ασφάλειας δεδομένων, με μεγαλύτερο έλεγχο στην επικοινωνία μεταξύ εφαρμογών
- Περισσότερο ευέλικτα και έξυπνα Notifications
- Καλύτερα και πιο ευέλικτα widgets
- Ασύρματη μετάδοση ήχου και εικόνας
- Έξυπνο Bluetooth για επικοινωνία με έξυπνα περιφερειακά με χαμηλή κατανάλωση ενέργειας
- Βελτίωση στην υποστήριξη της τεχνολογίας NFC (Near Field Communication)
- Προσδιορισμός θέσης μέσω Wi-Fi
- Υποστήριξη οθονών με ανάλυση 4K
- Αναβάθμιση του Android Market σε Google Play Store
- Λειτουργίες προσβασιμότητας όπως (τριπλό χτύπημα οθόνης για μεγέθυνση, zoom με δυο δάκτυλα, φωνητική πλοήγηση για τυφλούς)
- Δυνατότητα εισαγωγής μπάρας γρήγορων ρυθμίσεων στο πάνελ ειδοποιήσεων για πιο εύκολη πρόσβαση στο Wi-Fi, Bluetooth κλπ
- Δυνατότητα λειτουργίας Daydream

1.2.11 Android KitKat (API Level 19 & 20 Android Version 4.4)



Εικόνα 12: Android Kitkat

Τον Οκτώβριο του 2013 έκανε την εμφάνισή του το Android KitKat με νέα, φρέσκα χαρακτηριστικά τα οποία θα αναφέρουμε παρακάτω.

Συγκεκριμένα υπάρχει μία συνολική βελτίωση της απόδοσης των προεγκατεστημένων εφαρμογών αλλά και των εφαρμογών της Google, ενώ η κατάργηση διεργασιών που δεν έκαναν ουσιαστικά τίποτα, έχουν μειώσει το συνολικό αποτύπωμα του λειτουργικού συστήματος στη RAM. Αποτέλεσμα όλων αυτών είναι το Android να τρέχει καλύτερα σε συσκευές με 512MB RAM. Κάποια από τα επί μέρους χαρακτηριστικά που συναντάμε (όπως αναφέρει το [14],[18],[21]) είναι:

- Hangouts
- Εξυπνότερος dialer
Όπως διευκρινίζει και η Google, ο dialer είναι μία εφαρμογή που έχει μείνει στάσιμη τα τελευταία χρόνια. Αυτό αλλάζει πλέον καθώς μέσα από την εφαρμογή κλήσεων και αν είμαστε συνδεδεμένοι στο ίντερνετ, μπορούμε πλέον να ψάζουμε για τηλέφωνα επιχειρήσεων που είναι καταχωρισμένα στο Google Maps!
- Αυτό φυσικά λειτουργεί και αντίστροφα. Αν κάποιος άγνωστος αριθμός μας καλεί και είμαστε συνδεδεμένοι στο ίντερνετ, ο dialer θα μας πει σε ποιον αντιστοιχεί ή από ποια εταιρία μας καλεί! Απαραίτητη προϋπόθεση φυσικά, να είναι καταχωρημένοι στο ίντερνετ οι αριθμοί των καλούντων.
- Νέα λειτουργία immersive mode όπου οι εφαρμογές μπορούν να κρύβουν την πλοήγηση και την μπάρα κατάστασης
- Ευκολότερη εκτύπωση με το Google Cloud Print
- Νέα εικονίδια στο Google Keyboard
- Βιβλιοθήκες υπερύθρων (IR)
- Βελτιώσεις στο Bluetooth
- Screen recording
- Βελτίωση της περιήγησης στο ίντερνετ
- Περισσότερη μουσική, περισσότερη μπαταρία
- Προσθήκη βηματομετρητή
- Καλύτερη διαχείριση αρχείων στο cloud
- Μεγαλύτερη ασφάλεια
- Πληρωμές μέσω NFC version 2.0

1.2.12 Android Lollipop (API Level 21 & 22 Android Version 5.0 – 5.1.1)



Εικόνα 13: Android Lollipop

Νοέμβριος του 2014 και το Android Lollipop κάνει την εμφάνισή του για πρώτη φορά με νέο επανασχεδιασμένο περιβάλλον εργασίας χρήστη, χτισμένο γύρω από μία σχεδιαστική γλώσσα γνωστή ως Material Design. Άλλες αλλαγές περιλαμβάνουν βελτιώσεις στις κοινοποιήσεις, οι οποίες είναι προσβάσιμες από την οθόνη κλειδώματος και εμφανίζονται μέσα σε εφαρμογές. Η Google έκανε επίσης εσωτερικές αλλαγές στην πλατφόρμα, με το Android Runtime (ART) να είναι επίσημο αντικαθιστώντας το Dalvik για βελτιωμένη απόδοση των εφαρμογών, καθώς και με τις αλλαγές που αποσκοπούν στη βελτίωση και τη βελτιστοποίηση της χρήσης της μπαταρίας.

Η νέα έκδοση στη συνέχεια του Lollipop 5.1 φέρνει σημαντικές αλλαγές όπως silent mode το οποίο λείπει από το Android 5.0, βελτίωση στη σταθερότητα του συστήματος, καλύτερη διαχείριση της RAM, διορθώσεις σε κάποια apps τα οποία κλείνουν ξαφνικά, την υπερβολική κατανάλωση δεδομένων όταν είναι συνδεδεμένο σε Wi-Fi, προβλήματα με τη σύνδεση σε ασύρματα δίκτυα, με τη λειτουργία του Okay Google, με τα Notifications, αλλά και άλλες αλλαγές και βελτιώσεις (όπως αναφέρει το [14,18,21]) στο σύνολο του λειτουργικού συστήματος.

1.2.13 Android Marshmallow (API Level 23 Android Version 6.0 – 6.0.1)



Εικόνα 14: Android Marshmallow

Το Lollipop διαδέχεται το Marshmallow, το οποίο κυκλοφόρησε τον Οκτώβριο του 2015 και είναι η έκτη κύρια έκδοση του λειτουργικού συστήματος Android από την Google. Το Marshmallow επικεντρώνεται κυρίως στη βελτίωση της συνολικής εμπειρίας του χρήστη σε σύγκριση με τον προκάτοχο του. Εισάγει μια νέα αρχιτεκτονική στα δικαιώματα των εφαρμογών. Οι άδειες σε εφαρμογές - όπως για χρήση του μικροφώνου, της κάμερας, της περιοχής, του τηλεφώνου, των μηνυμάτων, του ημερολογίου - είχαν σοβαρά κενά ασφαλείας σε προηγούμενες

εκδόσεις του Android. Έπρεπε να επιτρέψεις σε μια εφαρμογή τον πλήρη έλεγχο της συσκευής σου, αν ήθελες να τη χρησιμοποιήσεις. Το Android 6 εισάγει ένα νέο μοντέλο αδειοδοτήσεων που επιτρέπει, όχι μόνο να απαγορέψουμε την πρόσβαση σε συγκεκριμένα στοιχεία, αλλά να την ανοιγοκλείνουμε όποτε θέλουμε.

Επίσης εισάγει νέα APIs όπως το «Now On Tap» το οποίο ενεργοποιείται με παρατεταμένο πάτημα στο πλήκτρο Home και παρέχει βοήθεια βασισμένη στην ενεργό εφαρμογή ή το περιεχόμενο κάποιου μηνύματος που διαβάζουμε. Κάποια επιπλέον νέα χαρακτηριστικά που προσφέρει το Marshmallow (όπως αναφέρει το [14,18,21]) είναι:

- Android Pay (Βασίζεται γύρω από την απλότητα χρήσης, την ασφάλεια και την επιλογή του χρήστη. Πρακτικά, μοιάζει πολύ με το Google Wallet με το οποίο μπορεί κάποιος να πληρώσει μέσω ασύρματων τερματικών).
- Αναγνώριση δακτυλικού αποτυπώματος.
- Παρακολούθηση Συγχρονισμού Συσκευής.
- Υποστήριξη USB Type-C.
- Καλύτερες επιδόσεις σε λειτουργίες κειμένου.

1.2.14 Android Nougat (API Level 24 & 25 Android Version 7.0 – 7.1.2)



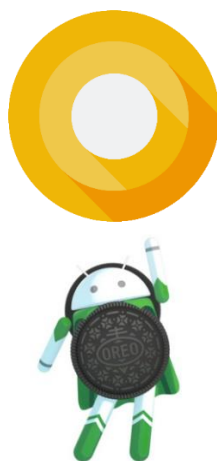
Εικόνα 15: Android Nougat

Το Android Nougat ξεκινάει την πορεία του τον Αύγουστο του 2016 εισάγοντας σημαντικές αλλαγές στο λειτουργικό σύστημα και την πλατφόρμα ανάπτυξης του. Σύμφωνα με την επίσημη σελίδα του Android, στην οποία έχει προστεθεί πλέον και η έκδοση Nougat, η Google μας παρουσιάζει τις ακόλουθες βασικές αλλαγές:

- **Emoji:** Πάνω από 1.500 συμπεριλαμβανομένων 72 νέων.
- **Γλωσσικές προτιμήσεις:** Δυνατότητα να ορίσουμε περισσότερες από μία αγαπημένες γλώσσες τις οποίες λαμβάνουν υπ' όψιν οι εφαρμογές που δεν είναι μεταφρασμένες στην πρωτεύουσα που έχουμε ορίσει.
- **Multi-window:** Ταυτόχρονη λειτουργία δύο εφαρμογών στην ίδια οθόνη.
- **Γρήγορη εναλλαγή εφαρμογών:** Γρήγορη εναλλαγή μεταξύ δύο εφαρμογών με το πλήκτρο της επισκόπησης.
- **Vulkan API:** Ένα νέο API το οποίο προσφέρει καλύτερα γραφικά σε 3D παιχνίδια, τα οποία όμως δε βαραίνουν το σύστημα.
- **Daydream:** Ενσωματωμένο Virtual Reality mode για επιλεγμένες συσκευές.
- **Doze on the Go:** Το Doze λειτουργεί ακόμη καλύτερα βάζοντας τη συσκευή μας σε deep sleep ακόμη και όταν την κουβαλάμε μαζί μας.

- **Παραμετροποιήσιμοι διακόπτες:** Οι διακόπτες της μπάρας ειδοποιήσεων μπορούν πλέον να παραμετροποιηθούν, ενώ οι πέντε κυριότεροι εμφανίζονται πάντα για ευκολία.
- **Άμεσες απαντήσεις:** Οι ειδοποιήσεις που προέρχονται από εφαρμογές μηνυμάτων, μας επιτρέπουν να απαντήσουμε γρήγορα χωρίς να χρειαστεί να τις ανοίξουμε.
- **Ομαδοποιημένες ειδοποιήσεις:** Οι ειδοποιήσεις που προέρχονται από την ίδια εφαρμογή, ομαδοποιούνται για να μπορούμε να τις διαγράψουμε γρηγορότερα, ενώ με ένα άγγιγμα ανοίγουν και μας προσφέρουν περισσότερες πληροφορίες.
- **Εξοικονόμηση δεδομένων:** Με τη λειτουργία εξοικονόμησης δεδομένων οι εφαρμογές μας χρησιμοποιούν λιγότερα δεδομένα στο παρασκήνιο, κάνοντας οικονομία στα data μας.
- **Έλεγχος ειδοποιήσεων:** Κρατώντας πατημένη μία ειδοποίηση εμφανίζεται ένα μενού με το οποίο μπορούμε να ρυθμίσουμε πώς θέλουμε να ειδοποιούμαστε από τη συγκεκριμένη εφαρμογή στο μέλλον.
- **Μέγεθος προβολής:** Με αυτή την επιλογή έχουμε τη δυνατότητα να κάνουμε τη γραμματοσειρά και τα αντικείμενα της οθόνης μεγαλύτερα, αν έχουμε προβλήματα όρασης.
- **Απρόσκοπτες ενημερώσεις:** Σε συγκεκριμένες συσκευές οι ενημερώσεις κατεβαίνουν στο παρασκήνιο και εγκαθίστανται απλά με μία επανεκκίνηση.
- **Κρυπτογράφηση ανά αρχείο:** Το λειτουργικό σύστημα πλέον κρυπτογραφεί τα αρχεία μας ξεχωριστά προσφέροντας μεγαλύτερη ασφάλεια.
- **Άμεση εκκίνηση:** Η εκκίνηση της συσκευής μας γίνεται γρηγορότερα με τις εφαρμογές μας να τρέχουν με ασφάλεια ακόμη και πριν την ξεκλειδώσουμε.

1.2.15 Android O (API Level 26 Android Version 8.0)



Εικόνα 16: Android Oreo

Τον Μάρτιο του 2017 η Google έκανε διαθέσιμο το Developer Preview του Android O και μάθαμε κάποια από τα χαρακτηριστικά που θα διαθέτει. Κατά τη

διάρκεια του διεθνούς συνεδρίου Google I/O η εταιρεία παρουσίασε τα νέα features που προστέθηκαν στο Android O, το οποίο πλέον ονομάζεται Android Oreo. Ξεκινάμε με την διάρκεια ζωής της μπαταρίας. Στην συγκεκριμένη έκδοση του Android η Google δίνει μεγάλη σημασία στην μπαταρία και οι βελτιώσεις θα γίνουν μέσω των νέων αυτόματων ορίων που επιβάλλονται στις ενέργειες που μπορούν να κάνουν οι εφαρμογές στο background.

Χάρη στο Smart Text Selection έχει βελτιωθεί η διαδικασία copy & paste, όσον αφορά στην αντιγραφή και επικόλληση ονομάτων τοποθεσιών, διευθύνσεων και αριθμών τηλεφώνου. Για παράδειγμα μέχρι τώρα έπρεπε να επιλέγεις χειροκίνητα την διεύθυνση την οποία ήθελες να ψάξεις, αλλά με το Android Oreo κάνοντας tap σε οποιαδήποτε λέξη της διεύθυνσης, γίνεται αυτόματη επιλογή. Στη συνέχεια μπορείς να μεταβείς στο Google Maps και να λάβεις τις σχετικές οδηγίες. Το ίδιο γίνεται και με τα ονόματα διάφορων επιχειρήσεων, ενώ όταν κάνεις double tap σε οποιοδήποτε ψηφείο κάποιου αριθμού, σου δίνει τη δυνατότητα κλήσης.

Το νέο feature ονόματι Google Play Protect θα εμφανιστεί στο Play Store και ουσιαστικά είναι η ίδια μηχανή της Google που σκανάρει αυτόματα τις εφαρμογές για κακόβουλη συμπεριφορά, αλλά πλέον θα είναι πιο ορατή στον τελικό χρήστη, δίνοντάς του πληροφορίες για το πότε έγινε το τελευταίο scan. Επιπλέον θα υπάρχει η επιλογή να ξεκινήσεις manual scan οποιαδήποτε στιγμή. Το Find my device θα είναι πλέον υπό την ομπρέλα του Play Protect.

Με το νέο Autofill API, θα μπορούν να γίνουν πιο εύκολα logins σε περισσότερες περιπτώσεις.

Με τα Notification Channels, οι developers θα καθορίζουν τις κατηγορίες για το περιεχόμενο των ειδοποιήσεων και ο χρήστης θα επιλέγει να κάνει “subscribe” μόνο στα channels που τον ενδιαφέρουν. Επιπλέον στο Android Oreo τα notifications θα είναι “snooze-able”. Το Picture-in-Picture Mode θα κάνει αυτό που γίνεται με το YouTube app όταν πατάμε ένα video να παίζει και στη συνέχεια θέλουμε να πάμε πίσω στο OS. Το video θα συνεχίσει να παίζει, ενώ ο χρήστης θα κάνει άλλα πράγματα στο κινητό.

Όταν έρχεται κάποια ειδοποίηση για συγκεκριμένο app, αυτό θα γίνεται εμφανές στο εικονίδιο του.

Μέσα η Google (όπως αναφέρει το [18],[21]) έχει κάνει αρκετές βελτιώσεις όπως το ότι το boot time του Android Oreo είναι δυο φορές ταχύτερο και οι βελτιωμένες επιδόσεις των apps. Αξίζει να σημειωθεί ότι κυκλοφόρησε για το κοινό στις 21 Αυγούστου του 2017.

1.3 Γιατι χρησιμοποιούμε Android

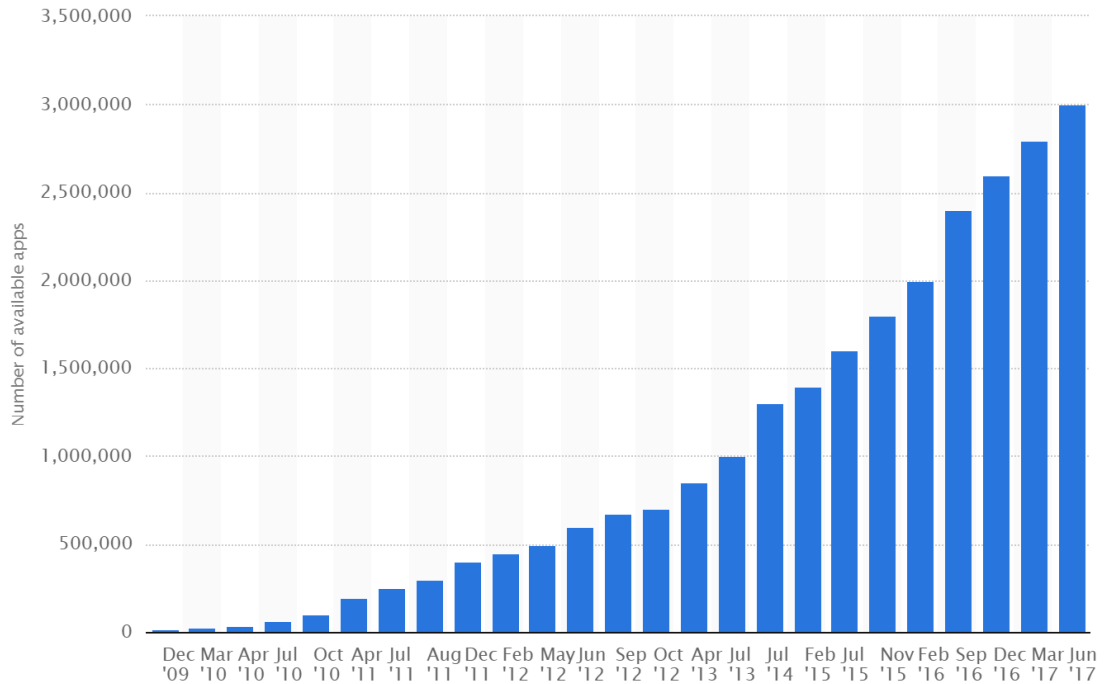
Το Android είναι ένα λογισμικό ιδιαίτερος γνωστό και ευρύτερα διαδεδομένο βασικότερα επειδή διαθέτει μια μεγάλη κοινότητα προγραμματιστών και επιπλέον είναι ανοιχτού κώδικα (open source). Οι εφαρμογές γράφονται σε μια προσαρμοσμένη έκδοση της JAVA χρησιμοποιώντας τα Android API's τα οποία είναι διαθέσιμα μαζί με τις πλατφόρμες [Android Studio](#) και το plugin [ADT](#) για το Eclipse. Έτσι ο προγραμματιστής βασίζεται στην “ελευθερία” του ανοιχτού κώδικα και της διευρυμένης κοινότητας γύρω από αυτόν, δημιουργώντας με αυτό τον τρόπο τις εφαρμογές που επιθυμεί.

Το γεγονός βέβαια ότι το κοινό του Android μπορεί να βρεί μία τεράστια γκάμα δωρεάν εφαρμογών στην αγορά, καλύπτοντας έτσι τις ανάγκες του στο έπακρο χωρίς κανένα χρηματικό κόστος, είναι από μόνο του ένα τεράστιο κίνητρο στο

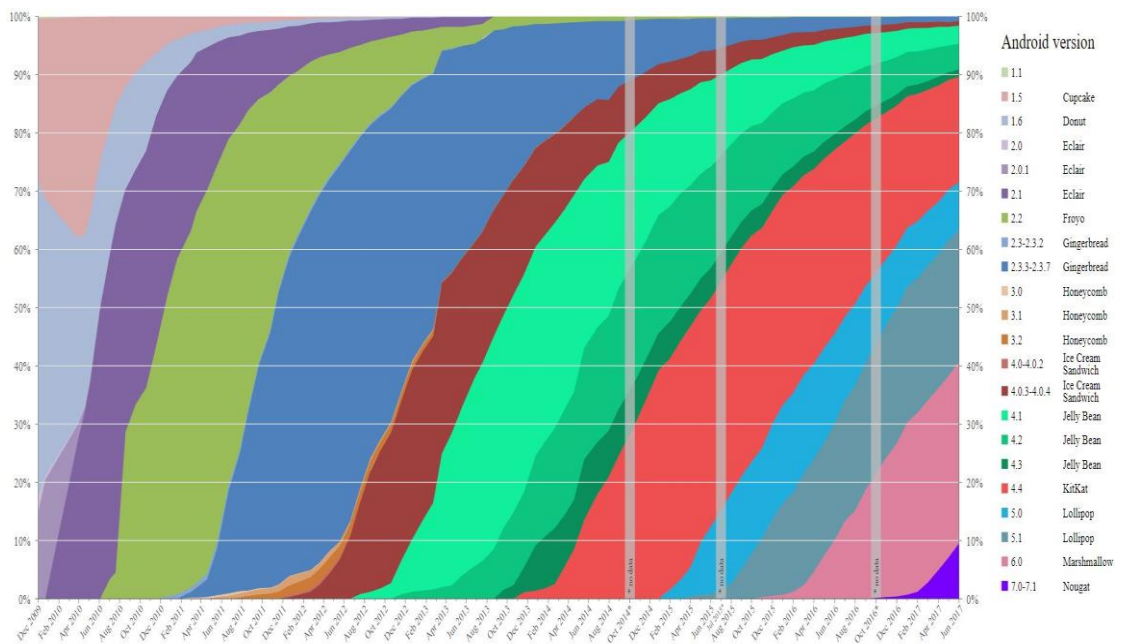
οποίο βασίζεται το “γιατί” χρησιμοποιούμε το συγκεκριμένο λειτουργικό σύστημα.

Ο αριθμός των διαθέσιμων εφαρμογών αυτή τη στιγμή στο Google Play Store ανέρχεται στα 3.000.000 σύμφωνα με στατιστική μελέτη του statista.com.

Παρακάτω θα δούμε πως μεταβλήθηκαν τα νούμερα αυτά με τα χρόνια (σύμφωνα με το [25]):



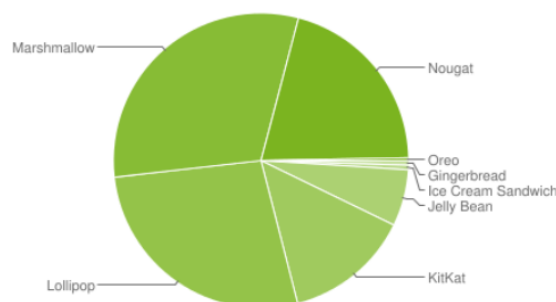
Εικόνα 17: Αριθμός διαθέσιμων εφαρμογών ανά την πάροδο του χρόνου [25]



Εικόνα 18: Χρονικό διάγραμμα χρήσης εκδόσεων Android [10]

Και εδώ παρατηρούμε ποσοστιαία τις προτιμήσεις των χρηστών αναλογικά με τις υπάρχουσες διανομές android αυτή τη στιγμή στην αγορά (σύμφωνα με το [10]).

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.5%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.5%
4.1.x	Jelly Bean	16	2.2%
4.2.x		17	3.1%
4.3		18	0.9%
4.4	KitKat	19	13.8%
5.0	Lollipop	21	6.4%
5.1		22	20.8%
6.0	Marshmallow	23	30.9%
7.0	Nougat	24	17.6%
7.1		25	3.0%
8.0	Oreo	26	0.3%



Data collected during a 7-day period ending on November 9, 2017.

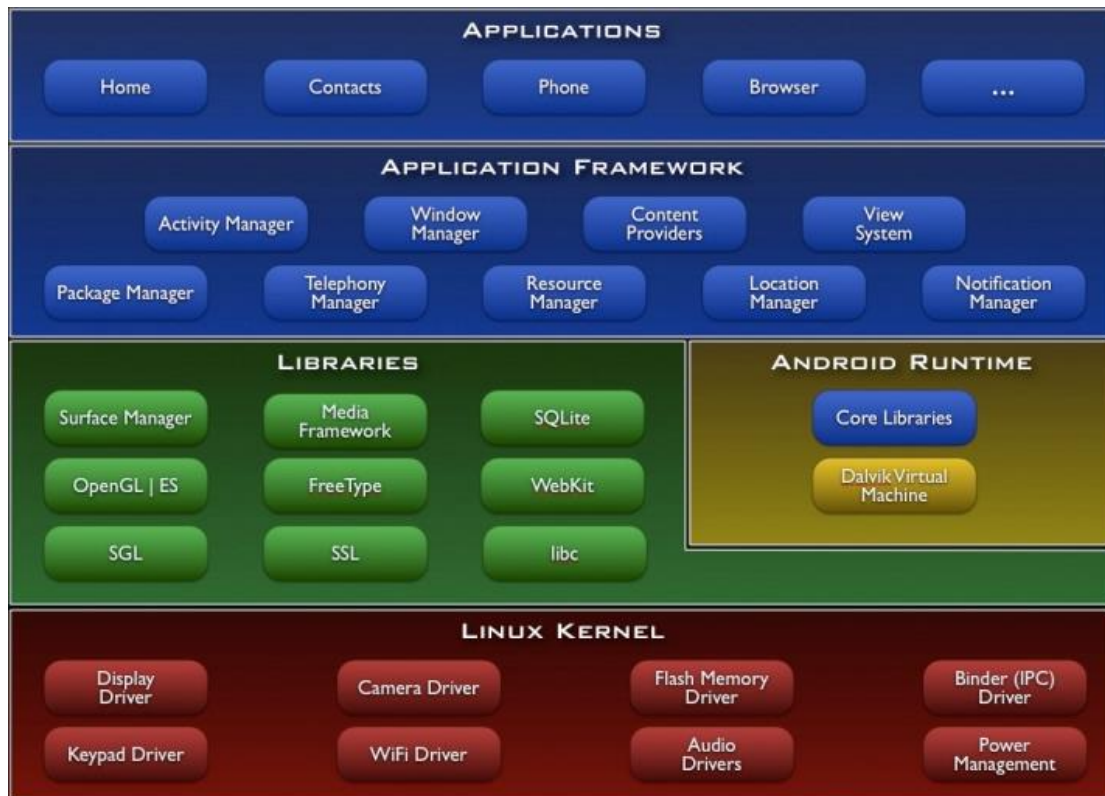
Any versions with less than 0.1% distribution are not shown.

Εικόνα 19: Προτιμήσεις Android χρηστών ποσοστιαία [10]

1.4 Δομή του Android

Το Android αποτελεί ένα ολοκληρωμένο σύστημα λογισμικού το οποίο περιλαμβάνει (α) το λειτουργικό σύστημα, (β) κάποιες βασικές εφαρμογές καθώς και (γ) το επίπεδο λογισμικού με το οποίο γίνεται εφικτή σύνδεση μεταξύ των εφαρμογών και του λειτουργικού συστήματος.

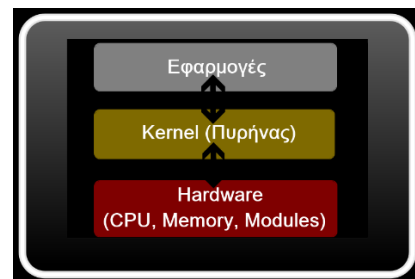
Πιο συγκεκριμένα, το Android αποτελείται από τέσσερα επίπεδα. Το κάθε επίπεδο εξαρτάται από τα υφιστάμενά του καθώς χρησιμοποιεί τις υπηρεσίες που παρέχουν.



Εικόνα 20: Διάγραμμα αρχιτεκτονικής Android [1]

1.4.1 Πυρήνας (Linux Kernel)

Το Android βασίζεται στον πυρήνα Linux, ο οποίος είναι υπεύθυνος να εκτελεί βασικές για το σύστημα υπηρεσίες, όπως είναι η διαχείριση των διεργασιών, η διαχείριση μνήμης, η χρήση των υπηρεσιών δικτύωσης, η χρήση οδηγών (drivers) για διάφορα τμήματα του υλικού κ.ά. Ο πυρήνας αποτελεί το «μεσολαβητή» ανάμεσα στο λογισμικό και το υλικό.

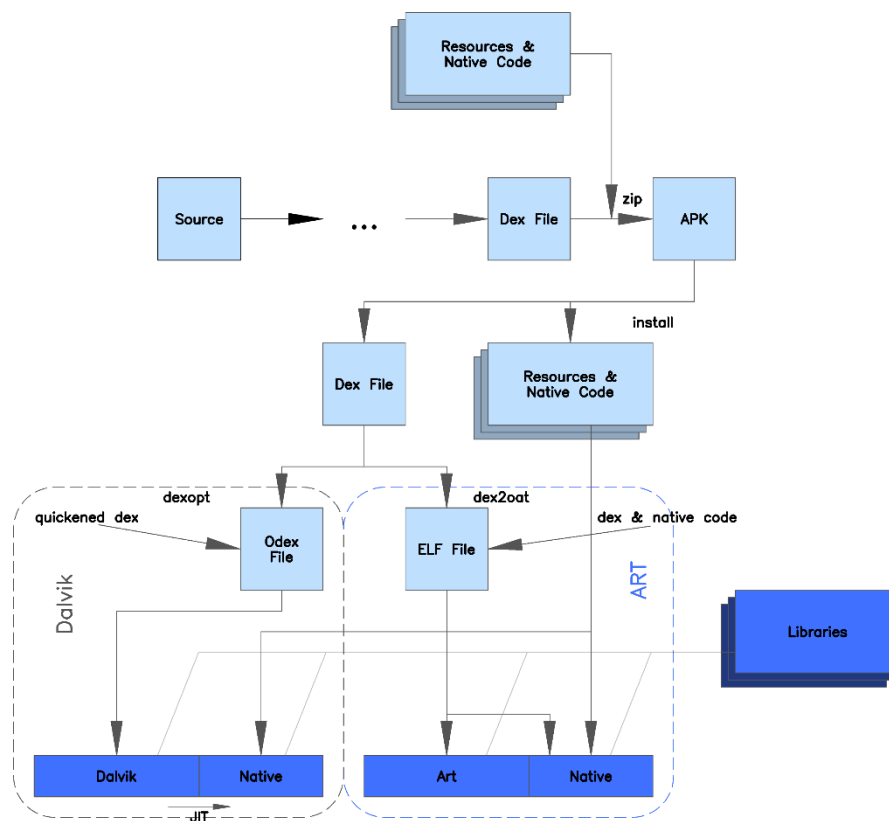


Εικόνα 21: Android Kernel

1.4.2 Βιβλιοθήκες

Στο αμέσως επόμενο (από τον πυρήνα), επίπεδο εντάσσονται οι βιβλιοθήκες, οι οποίες βασίζονται στη γλώσσα προγραμματισμού C/C++ και χρησιμοποιούνται από διάφορα στοιχεία του συστήματος. Από την έκδοση Android Donut (την οποία θα δούμε αναλυτικότερα παρακάτω στο κεφάλαιο εκδόσεις), οι κατασκευαστές έχουν τη δυνατότητα ανάπτυξης δικών τους βιβλιοθηκών μέσω των εργαλείων του NDK (Native Development Kit) με σκοπό την πλήρη εκμετάλλευση των τεχνολογικών δυνατοτήτων των συσκευών τους.

Στο ίδιο επίπεδο βρίσκεται και ο χρόνος εκτέλεσης του Android (Android Runtime) στο οποίο υπάρχουν βασικές βιβλιοθήκες της Java αλλά και η εικονική μηχανή Dalvik. Η **Dalvik** αποτελεί μία εικονική μηχανή Java, κατάλληλη για φορητές συσκευές, η οποία λόγω των περιορισμένων δυνατοτήτων των συσκευών είναι βελτιστοποιημένη να χρησιμοποιεί μικρότερα σε μέγεθος αρχεία ενδιάμεσου κώδικα (.dex) από αυτά της Java SE (.class αρχεία). Αξίζει να σημειωθεί βέβαια ότι από την έκδοση Android 5.0 «Lollipop» η Google αντικατέστησε την εικονική μηχανή Dalvik με την ART. Η βασική διαφορά τους είναι ο τρόπος «μετάφρασης» του κώδικα των εφαρμογών σε γλώσσα μηχανής ώστε να εκτελεστούν από το σύστημα. Η Dalvik εκτελούσε τη μετάφραση τη στιγμή όπου ο χρήστης «έτρεχε» την εκάστοτε εφαρμογή. Αντιθέτως η ART δημιουργεί και αποθηκεύει τη μετάφραση έτσι ώστε να είναι διαθέσιμη πάντα στο λειτουργικό σύστημα.



Εικόνα 22: Dalvik & Art Runtime

1.4.3 Πλαίσιο εφαρμογών (Application Framework)

Αμέσως μετά τοποθετείται το επίπεδο του πλαισίου εφαρμογών ([1],[2],[3]). Πρόκειται για ένα πλαίσιο ανάπτυξης το οποίο επιτρέπει στους προγραμματιστές να χρησιμοποιήσουν τις δυνατότητες του Android στις εφαρμογές τους. Το πλαίσιο ανάπτυξης είναι προεγκατεστημένο αλλά μπορεί και να επεκταθεί καθώς ο κάθε κατασκευαστής μπορεί να προσθέσει τις δικές του βιβλιοθήκες. Κάποια κύρια δομικά συστατικά του πλαισίου είναι:

- Διαχειριστής δραστηριότητας (Activity Manager): Ο διαχειριστής δραστηριότητας ελέγχει το χρόνο ζωής (lifetime) των εφαρμογών.
- Πάροχος περιεχομένου (Content Provider): Πρόκειται για πακέτα δεδομένων τα οποία διαμοιράζονται μεταξύ των εφαρμογών.
- Διαχειριστής πόρων (Resource Manager): Πόροι ορίζονται τα στοιχεία που περιέχει μια εφαρμογή εκτός του κώδικα. Πόροι συστήματος μπορεί να είναι κάποιο πολυμεσικό υλικό, αλφαριθμητικοί χαρακτήρες, κωδικοί χρωμάτων, σχεδιασμένες εικόνες κ.ά.
- Διαχειριστής τοποθεσίας (Location Manager): Αναλαμβάνει να δίνει στίγμα στη συσκευή για την τοποθεσία του χρήστη.
- Διαχειριστής ενημερώσεων (Notification Manager): Αναλαμβάνει να ενημερώνει το χρήστη για γεγονότα που συμβαίνουν χωρίς να διακόπτεται η εργασία του.

1.4.4 Εφαρμογές (Applications)

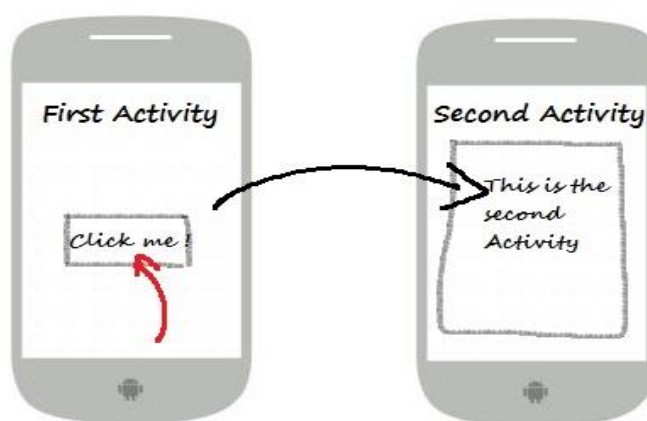
Στο υψηλότερο επίπεδο βρίσκονται οι εφαρμογές τις οποίες χρησιμοποιεί ο χρήστης. Υπάρχουν τόσο προεγκατεστημένες, στη συσκευή εφαρμογές, όπως η εφαρμογή για αποστολή μηνυμάτων, Media Player κ.ά, όσο και εφαρμογές από τρίτους όπου ο χρήστης μπορεί να εγκαταστήσει στη συσκευή ανάλογα με τις ανάγκες του.

1.5 Δομικά στοιχεία εφαρμογών Android

Μία εφαρμογή Android αποτελείται κάποια βασικά δομικά στοιχεία. Τα βασικά δομικά στοιχεία, κάποια από τα οποία μπορεί να περιέχει μία εφαρμογή σύμφωνα με ([1],[2],[3]), είναι τα εξής:

1.5.1 Δραστηριότητα (Activity)

Αποτελεί το πιο σύνηθες δομικό στοιχείο, καθώς μια δραστηριότητα «αντιπροσωπεύει» και μία ξεχωριστή οθόνη της εφαρμογής με την οποία αλληλεπιδρά ο χρήστης. Σε κάθε εφαρμογή υπάρχει ένα σύνολο από activities (σε μορφή στοίβας). Κάθε δραστηριότητα η οποία βρίσκεται στην κορυφή της στοίβας αντιπροσωπεύει την οθόνη που βλέπει εκείνη τη στιγμή ο χρήστης. Η δραστηριότητα η οποία ξεκινάει πρώτη στην εφαρμογή (η πρώτη οθόνη) ονομάζεται συνήθως Κεντρική Δραστηριότητα (Main Activity).



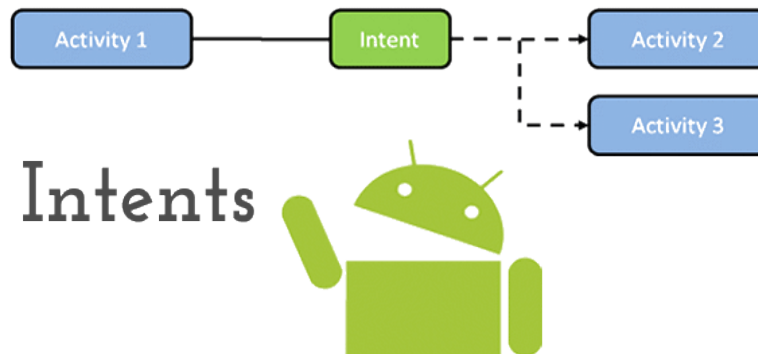
Εικόνα 23: Λειτουργία των Activities.

Το οπτικό περιεχόμενο κάθε δραστηριότητας αποτελείται από Views μέσω των οποίων πραγματοποιείται και η αλληλεπίδραση του χρήστη με το activity.

1.5.2 Ανταλλαγή μηνυμάτων προθέσεων (Intents)

Η ενεργοποίηση κάθε activity γίνεται με τη βοήθεια ασύγχρονων μηνυμάτων, τα οποία ονομάζονται intents. Κάθε intent είναι μια αφηρημένη περιγραφή μιας πράξης (action) που πρόκειται να εκτελεστεί και χρησιμοποιείται συνήθως για εκκίνηση δραστηριοτήτων και υπηρεσιών (Services). Γενικότερα η τεχνική αυτή χρησιμοποιείται από το Android για να επικοινωνεί πληροφορίες ανάμεσα σε δραστηριότητες, μέσα σε μια εφαρμογή ή ανάμεσα σε δραστηριότητες σε διαφορετικές εφαρμογές. Κάθε Activity μπορεί να καθορίζει φίλτρα προθέσεων (intent filters), που δηλώνουν ενέργειες, που μπορεί να χειριστεί η Activity. Τα φίλτρα προθέσεων ορίζονται μέσα στο αρχείο AndroidManifest.xml (το οποίο βλέπουμε παρακάτω).

Ένα intent χρησιμοποιείται για να εκκινήσει μια Activity- δηλώνει μια ενέργεια, που θα εκτελείται και τα δεδομένα, επί των οποίων θα εκτελεστεί αυτή η ενέργεια.



Εικόνα 24: Λειτουργία των intents.

1.5.3 Κύκλος ζωής μιας δραστηριότητας

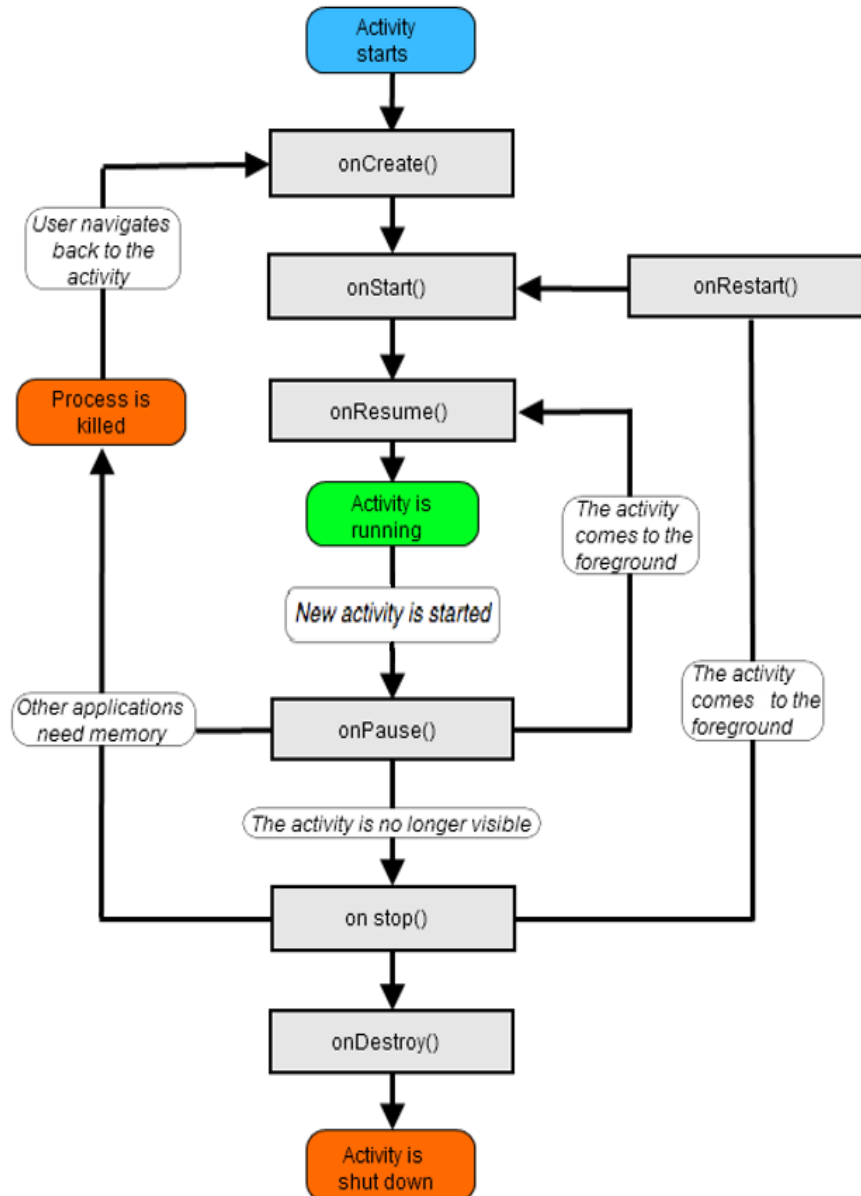
Από την έναρξη μιας activity μέχρι και το τέλος της, η κάθε δραστηριότητα έχει έναν κύκλο ζωής (Lifecycle). Το Lifecycle περιγράφει δηλαδή τις τρεις βασικές καταστάσεις στις οποίες μπορεί να βρεθεί μια activity. Οι καταστάσεις στις οποίες μπορεί να βρεθεί είναι:

- Να είναι ενεργή. Να εκτελείται δηλαδή και να είναι ορατή στο χρήστη.
- Να βρίσκεται σε παύση. Να έχει συνεπώς σταματήσει προσωρινά τη λειτουργία της διότι μια άλλη activity εκτελείται «πάνω» από αυτή.
- Να είναι σταματημένη σε περίπτωση που καλύπτεται από κάποια άλλη activity. Όταν μια activity είναι σταματημένη συνεχίζει να διατηρεί την κατάσταση και τις πληροφορίες των μελών της, και παύει να υφίσταται μόνο όταν μια άλλη εφαρμογή χρειαστεί τη μνήμη που καταλαμβάνει.

Κατά την εναλλαγή των καταστάσεων, η activity ειδοποιείται μέσω κάποιων μεθόδων οι οποίες είναι σε θέση να πραγματοποιήσουν κάποια εργασία όταν αλλάζει η κατάσταση.

Όπως φαίνεται και στο παρακάτω σχήμα ο κύκλος ζωής μιας activity αρχίζει με τη κλήση της μεθόδου onCreate(), όπου ουσιαστικά «αρχίζει να υπάρχει» η δραστηριότητα και τερματίζεται με τη μέθοδο onDestroy(). Ανάμεσα στις δύο μεθόδους βρίσκεται και το χρονικό διάστημα όπου ο χρήστης βλέπει τη συγκεκριμένη activity στην οθόνη του. Ο χρόνος όπου μια activity είναι ορατή αρχίζει με τη μέθοδο onStart() και τελειώνει κάνοντας χρήση της μεθόδου onStop(). Οι δύο παραπάνω μέθοδοι μπορούν να καλεστούν πολλές φορές ανάλογα με το αν η activity είναι ορατή ή όχι. Καθ' όλη τη διάρκεια που ένα activity βρίσκεται στο προσκήνιο και ο χρήστης

αλληλεπιδρά με αυτό, βρίσκεται μεταξύ των μεθόδων `onResume()` και `onPause()`. Η μέθοδος `onPause()` καλείται όταν η συσκευή βρίσκεται σε αδράνεια ή όταν εκκινείται μια άλλη activity. Κατά την επιστροφή στη δραστηριότητα εκτελείται η μέθοδος `onResume()`.



Εικόνα 25: Το Lifecycle μιας Android εφαρμογής ([1],[2],[3]).

1.5.4 Τεμάχια (Fragments)

Ένα τεμάχιο (fragment) τυπικά παριστά ένα επαναχρησιμοποιήσιμο τμήμα της διεπαφής χρήστη μιας Activity, αλλά μπορεί επίσης να παριστά επαναχρησιμοποιήσιμη λογική προγράμματος. Αυτή η εφαρμογή χρησιμοποιεί τεμάχια για να δημιουργήσει και να διαχειριστεί τμήματα της γραφικής διεπαφής χρήστη. Μπορούμε να συνδυάσουμε αρκετά τεμάχια, για να δημιουργήσουμε διεπαφές χρήστη, οι οποίες εκμεταλλεύονται τα μεγέθη οθόνης των κινητών συσκευών. Επίσης μπορούμε εύκολα να εναλλάξουμε τεμάχια, προκειμένου να κάνουμε τις γραφικές διεπαφές χρήστη περισσότερο δυναμικές.

Το Fragment είναι η βασική κλάση για όλα τα τεμάχια. Η εφαρμογή μας περιέχει τις παρακάτω υποκλάσεις της Fragment:

- CategoriesFragment
- CreditsFragment
- HomeFragment
- SearchFragment
- SettingsFragment

Παρακάτω παρατηρούμε την εφαρμογή των παραπάνω fragments έμπρακτα. Κάθε πεδίο στο μενού αντιπροσωπεύει και ένα τεμάχιο (Fragment).



Εικόνα 26: Υλοποίηση Μενού με χρήση Fragment

Και για να συνοψίσουμε, να τονίσουμε πως οι Fragment πρέπει να φιλοξενούνται από μία Activity, δεν μπορούν να εκτελούνται ανεξάρτητα. Για παράδειγμα στην εφαρμογή μας, η Activity που φιλοξενεί τα Fragment είναι η παρακάτω:

```
import tp3646.evangelos.chaniotakis.forsaleorwanted.fragment.CategoriesFragment;
import tp3646.evangelos.chaniotakis.forsaleorwanted.fragment.CreditsFragment;
import tp3646.evangelos.chaniotakis.forsaleorwanted.fragment.HomeFragment;
import tp3646.evangelos.chaniotakis.forsaleorwanted.fragment.SearchFragment;
import tp3646.evangelos.chaniotakis.forsaleorwanted.fragment.SettingsFragment;

/**
 * Created by VAGGELIS on 19/10/2017.
 */
public class HomeActivity extends AppCompatActivity {

    private static final String TAG = HomeActivity.class.getSimpleName();

    private Fragment selectedFragment = null;

    private BottomNavigationView.OnNavigationItemSelectedListener mOnNavigationItemSelectedListener
        = new BottomNavigationView.OnNavigationItemSelectedListener() {

        @Override
        public boolean onNavigationItemSelected(@NonNull MenuItem item) {
            Fragment selectedFragment = null;
            switch (item.getItemId()) {
                case R.id.quiz_mart:
                    selectedFragment = new CategoriesFragment();
                    break;
                case R.id.home:
                    selectedFragment = new HomeFragment();
                    break;
                case R.id.ladder_board:
                    selectedFragment = new SearchFragment();
                    break;
                case R.id.settings:
                    selectedFragment = new SettingsFragment();
                    break;
                case R.id.credits:
                    selectedFragment = new CreditsFragment();
                    break;
            }
            FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();
            transaction.replace(R.id.content, selectedFragment);
            transaction.commit();
            return true;
        }
    };

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

        selectedFragment = new CategoriesFragment();
        FragmentTransaction transactions = getSupportFragmentManager().beginTransaction();
        transactions.replace(R.id.content, selectedFragment);
        transactions.commit();

        BottomNavigationView navigation = (BottomNavigationView) findViewById(R.id.navigation);
        navigation.setOnNavigationItemSelectedListener(mOnNavigationItemSelectedListener);
    }
}
```

Εικόνα 27: Activity φιλοξενίας των Fragment

1.5.5 Υπηρεσίες (Services)

Πρόκειται (σύμφωνα με [1],[2],[3]) για ένα δομικό στοιχείο της εφαρμογής το οποίο βρίσκεται στο παρασκήνιο (background) και συνήθως χρησιμοποιείται για την εκτέλεση χρονοβόρων διαδικασιών. Σε αντίθεση με μία Activity, ένα Service δεν περιέχει κάποιο γραφικό περιβάλλον και συνεχίζει να λειτουργεί ακόμα και αν η εφαρμογή η οποία το ξεκίνησε σταματήσει. Χαρακτηριστικό παράδειγμα ενός service είναι η εφαρμογή για την αναπαραγωγή μουσικής. Ο χρήστης μπορεί να επιλέξει κάποιο μουσικό κομμάτι και να το ακούσει καθώς τρέχει κάποια διαφορετική εφαρμογή από το Music Player της συσκευής.

1.5.6 Πάροχος περιεχομένου (Content Provider)

Οι πάροχοι περιεχομένου (σύμφωνα με [1],[2],[3]) διαχειρίζονται τις τοποθεσίες όπου αποθηκεύονται δεδομένα τα οποία μπορούν να είναι προσπελάσιμα από όλες τις εφαρμογές. Οι τοποθεσίες αποθήκευσης των δεδομένων μπορεί να είναι είτε κάποιοι φάκελοι στο χώρο του κινητού, είτε κάποια βάση δεδομένων SQLite. Χαρακτηριστικό

παράδειγμα αποτελεί η εφαρμογή της κάμερας η οποία αποθηκεύει τις φωτογραφίες σε κάποιο συγκεκριμένο φάκελο ώστε να μπορούν να χρησιμοποιηθούν από άλλες εφαρμογές.

1.5.7 Παραλήπτες μηνυμάτων πανεκπομπών (Broadcast Receivers)

Οι Broadcast Receivers (σύμφωνα με [1],[2],[3]) ενεργοποιούνται όταν λαμβάνουν κάποιο γεγονός από το λειτουργικό σύστημα. Ένα τέτοιο γεγονός μπορεί να είναι η ένδειξη για χαμηλή μπαταρία της συσκευής, η ολοκλήρωση λήψης ενός αρχείου από το internet ακόμα και μία αναπάντητη κλήση. Ένας broadcast receiver μπορεί να ξεκινήσει μια activity όταν λάβει κάποιο γεγονός αλλά και να ενημερώσει το χρήστη μέσω κάποιας ειδοποίησης όπως κάνοντας χρήση της δόνησης της συσκευής ή και μέσω της αναπαραγωγής κάποιου ήχου.

1.5.8 Μανιφέστο Εφαρμογών (Android Manifest)

Για να εκκινήσει το σύστημα ένα δομικό στοιχείο θα πρέπει (σύμφωνα με [1],[2],[3]) πρώτα να γνωρίζει την ύπαρξή του. Η δήλωση των δομικών στοιχείων που βρίσκονται σε μία εφαρμογή γίνεται σε ένα αρχείο που ονομάζεται Manifest και περιέχεται σε κάθε εκτελέσιμο αρχείο .apk μαζί με τον πηγαίο κώδικα και τους πόρους της εφαρμογής. Το αρχείο manifest είναι μορφής XML και έχει την ονομασία AndroidManifest.xml σε όλες τις εφαρμογές. Εκτός από τη δήλωση των δομικών στοιχείων, στο manifest γίνεται η δήλωση των αδειών που πρέπει να αποδεχτεί ο χρήστης ώστε η εφαρμογή να εκτελέσει κάποιες λειτουργίες (π.χ η άδεια για λειτουργία εύρεσης τοποθεσίας του χρήστη), η δήλωση της ελάχιστης έκδοσης Android που απαιτεί η εφαρμογή για να εκτελεστεί κ.ά.


```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="tp3646.evangelos.chaniotakis.forsaleorwanted">

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/logo_icon"
        android:label="For Sale or Wanted"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <meta-data
            android:name="com.google.android.gms.wallet.api.enabled"
            android:value="true" />

        <activity android:name=".LanguageActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

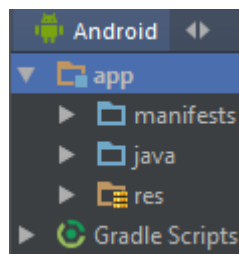
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".LoginOptionActivity"
            android:launchMode="singleTop"
            android:theme="@style/AppThemeFullScreen"
            android:windowSoftInputMode="adjustPan" />
        <activity
            android:name=".LoginActivity"
            android:launchMode="singleTop"
            android:theme="@style/AppThemeFullScreen"
            android:windowSoftInputMode="adjustPan" />
    </application>
</manifest>
```

Εικόνα 28: AndroidManifest.xml

1.6 Εσωτερικό Εφαρμογής Android

Το εσωτερικό μιας εφαρμογής περιλαμβάνει τα ακόλουθα:

- Φάκελος manifests
- Φάκελος java / src
- Φάκελος res
- Gradle Scripts

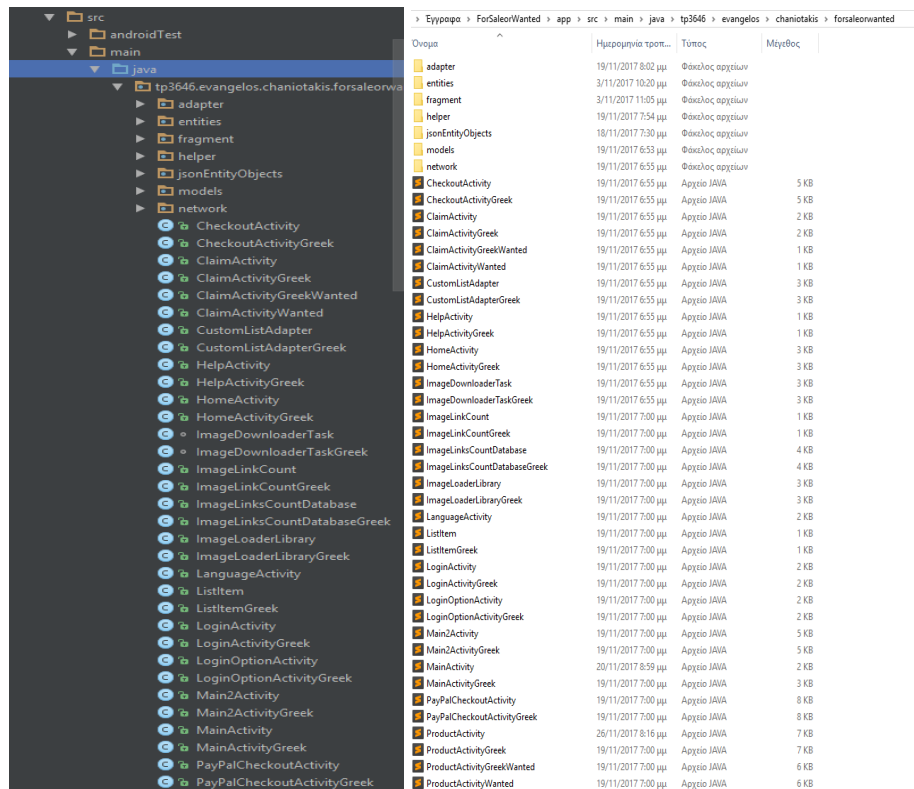


Εικόνα 29: Δομή Project Android

1.6.1 Φάκελος src

Ο φάκελος πηγών «Java» περιλαμβάνει τον πηγαίο κώδικα της εφαρμογής. Σε αυτόν το φάκελο τοποθετούνται οι κλάσεις Java οι οποίες είναι υπεύθυνες για τη λειτουργία της εφαρμογής. Αυτές μετατρέπονται σε αρχεία .dex για να μπορούν να γίνουν bytecode από τους compilers Dalvik και ART.

Διαμορφώνεται κάπως έτσι:

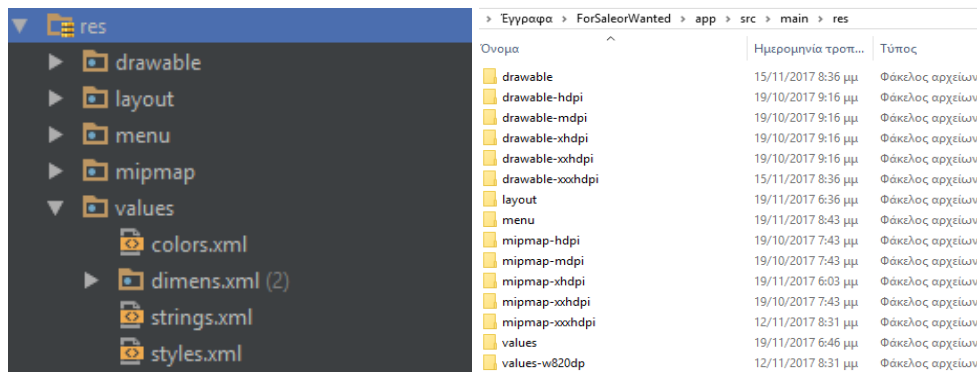


Εικόνα 30: Φάκελος src στο Android Studio (αριστερά) και στο File Explorer (δεξιά)

Παρατηρούμε τα packages, μέσα στα οποία περιέχονται αρχεία java.

1.6.2 Φάκελος res

Ο φάκελος res περιέχει τα απαραίτητα αρχεία για το οπτικό μέρος της εφαρμογής. Πιο συγκεκριμένα τοποθετούνται όλες οι εικόνες οι οποίες θα χρησιμοποιηθούν καταναμημένες βάσει της ανάλυσης της οθόνης στην οποία απευθύνονται, τα xml αρχεία των μενού, τα εικονίδια της εφαρμογής σε διαφορετικές αναλύσεις αλλά και τα xml αρχεία τα οποία περιέχουν πληροφορίες σχετικά με τις διαστάσεις αλλά και τιμές των διαφόρων αντικειμένων που περιλαμβάνονται σε μία Activity. Συγκεκριμένα η εφαρμογή που μελετάμε περιέχει τους εξής φακέλους:

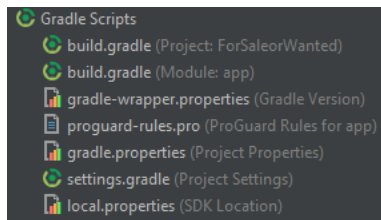


Εικόνα 31: Φάκελος res στο Android Studio (αριστερά) και στο File Explorer (δεξιά)

- **drawable** (Περιέχει τους φακέλους εικόνων (hdpi, mdpi, xhdpi, xxhdpi, xxxhdpi) καθώς και αρχεία .xml που καθορίζουν συγκεκριμένο στυλ για κάποια είδη συμπεριφορών σε διάφορα κομμάτια κώδικα)
- **layout** (Περιέχει τα αρχεία .xml που χρησιμοποιεί η εφαρμογή μας για την οπτικοποίηση των Activities, Fragments κλπ.)
- **menu** (Περιέχει τα .xml αρχεία που είναι υπεύθυνα για την δημιουργία μενού σε συγκεκριμένα σημεία της εφαρμογής μας)
- **mipmap** (Περιέχει τα αναγνωριστικά εικονίδια - λογότυπα τα οποία χρησιμοποιούνται σε συγκεκριμένα σημεία της εφαρμογής μας)
- **values** (Περιέχει τα αρχεία (colors.xml, dims.xml, strings.xml, styles.xml))
 - **colors.xml** (Περιέχει ένα πίνακα στατικών χρωμάτων στα οποία μπορούμε να έχουμε πρόσβαση από κάθε σημείο της εφαρμογής μας)
 - **dims.xml** (Περιέχει αναφορές σε διάφορα περιθώρια και μετακινήσεις μεταξύ αντικειμένων της εφαρμογής μας)
 - **strings.xml** (Περιέχει στατικά πεδία κειμένου τα οποία μπορούν να τοποθετηθούν σε διάφορα σημεία της εφαρμογής χρησιμοποιώντας κάποιο αναγνωριστικό id)
 - **styles.xml** (Περιέχει το στυλ με το οποίο θα φαίνεται η εφαρμογή στον τελικό χρήστη)

1.6.3 Gradle Scripts

Πρόκειται (σύμφωνα με [1],[2],[3]) για το σύστημα εργαλείων το οποίο χρησιμεύει στην παραγωγή, στη δοκιμή και στην εκτέλεση της εφαρμογής. Πιο συγκεκριμένα ο προγραμματιστής είναι σε θέση να διαμορφώσει και να επεκτείνει τη διαδικασία παραγωγής αλλά και να δημιουργήσει πολλαπλά εκτελέσιμα αρχεία χρησιμοποιώντας το ίδιο project και τις ιδιότητές του.



Εικόνα 32: Φάκελος Gradle Scripts στο Android Studio

Κεφάλαιο 2

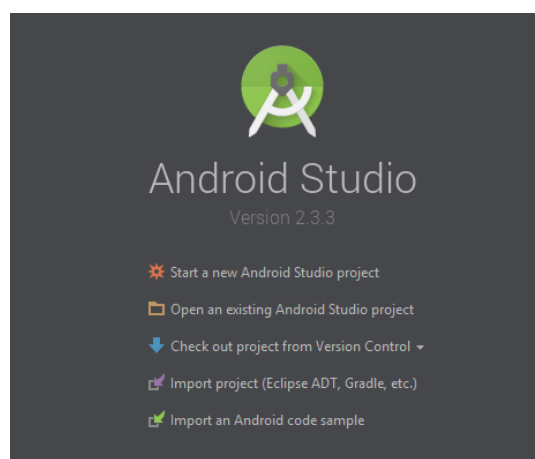
2.1 Εγκατάσταση Λογισμικού

Στην επίσημη σελίδα των Android Developers [10] βρήκα εκτενείς οδηγίες για το πώς να ξεκινήσω, πώς να σχεδιάσω μια εφαρμογή και λεπτομέρειες σχετικά με όλα τα διαθέσιμα Application Programming Interfaces (API) που μας δίνουν τη δυνατότητα να χρησιμοποιούμε υπηρεσίες άλλων εφαρμογών μέσα στη δική μας (π.χ. τους χάρτες του Google Maps). Τα εργαλεία για ανάπτυξη Android εφαρμογών είναι αυτό που λέμε cross-platform, δηλαδή διαθέσιμα για Windows, Linux και OSX, συνεπώς δεν απαιτείται η αγορά κάποιου επιπλέον hardware από εμάς.



Εικόνα 33: Android Developers Ιστοσελίδα

Για την ανάπτυξη Android εφαρμογών είναι απαραίτητη η εγκατάσταση ενός προγράμματος συγγραφής κώδικα. Στις αρχές του 2015 η Google κυκλοφόρησε το επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment, IDE) για την ανάπτυξη Android εφαρμογών, Android Studio [16].



Εικόνα 34: Αρχική οθόνη του Android Studio

Κατά την έναρξη του Android Studio ο προγραμματιστής έχει τη δυνατότητα δημιουργίας ενός νέου ή επεξεργασίας ενός ήδη υπάρχοντος project, της εισαγωγής πρότυπων αρχείων κώδικα με υλοποιημένες υπηρεσίες που προσφέρει το λογισμικό, της λήψης ενός project διαμέσου κάποιων εκ των συστημάτων Ελέγχου Έκδοσης όπως το Git. Επιπροσθέτως είναι σε θέση να εισάγει κάποιο ήδη υπάρχον project το οποίο έχει υλοποιηθεί σε άλλη πλατφόρμα IDE όπως το Eclipse.

2.2 Διαδικασία Αποσφαλμάτωσης(Debugging)

Το κάθε περιβάλλον προγραμματισμού και δημιουργίας εφαρμογών Android περιλαμβάνει κάποια εργαλεία για την αποσφαλμάτωση της εφαρμογής. Ο προγραμματιστής επιλέγει πως θα υλοποιήσει την εφαρμογή αλλά το λειτουργικό σύστημα θα καθορίσει ποιούς πόρους θα χρησιμοποιήσει. Σε αυτήν την διαδικασία χρησιμοποιείται το Logcat το οποίο είναι ένας καταγραφέας για το οτιδήποτε συμβαίνει στην εφαρμογή μας και στους πόρους του συστήματος που χρησιμοποιεί. Είναι υπεύθυνο για να ενημερώσει τον προγραμματιστή για ότι προκύψει (μνήμη που χρησιμοποιεί η εφαρμογή, συντακτικά και λογικά λάθη στον κώδικα, σφάλματα στο μητρώο, παραβίαση αδειών, παραβίαση δεδομένων της εφαρμογής και άλλα). Επίσης ένα άλλο σημαντικό εργαλείο που χρησιμοποιούν οι προγραμματιστές είναι το DDMS το οποίο παρέχει την δυνατότητα ελέγχου της εφαρμογής σε εικονικές συσκευές που δεν έχουν πρόσβαση σε ζωντανά δεδομένα. Αν πούμε ότι για παράδειγμα φτιάχνουμε μια εφαρμογή ανταλλαγής μηνυμάτων κειμένου και στατικών εικόνων, πριν την δοκιμάσουμε σε κανονική συσκευή και μας επιφέρει πρόσθετες χρεώσεις μόνο για τις δοκιμές μπορούμε να στέλνουμε fake notifications ή και μηνύματα στην εικονική συσκευή με στόχο την βελτίωση. Το ίδιο γίνεται και με την γεωγραφική θέση, κλήση αριθμού, προβολή website, προβολή στατιστικών heap memory allocation, thread memory size, network statistics και άλλα.

2.3 Δημοσίευση Εφαρμογής

Αφού αναπτύξουμε και ελέγξουμε την εφαρμογή μας, τόσο στον εξομοιωτή, όσο και σε συσκευή Android, το επόμενο βήμα είναι να την υποβάλουμε στο Google Play για να τη διανείμουμε στο παγκόσμιο κοινό.

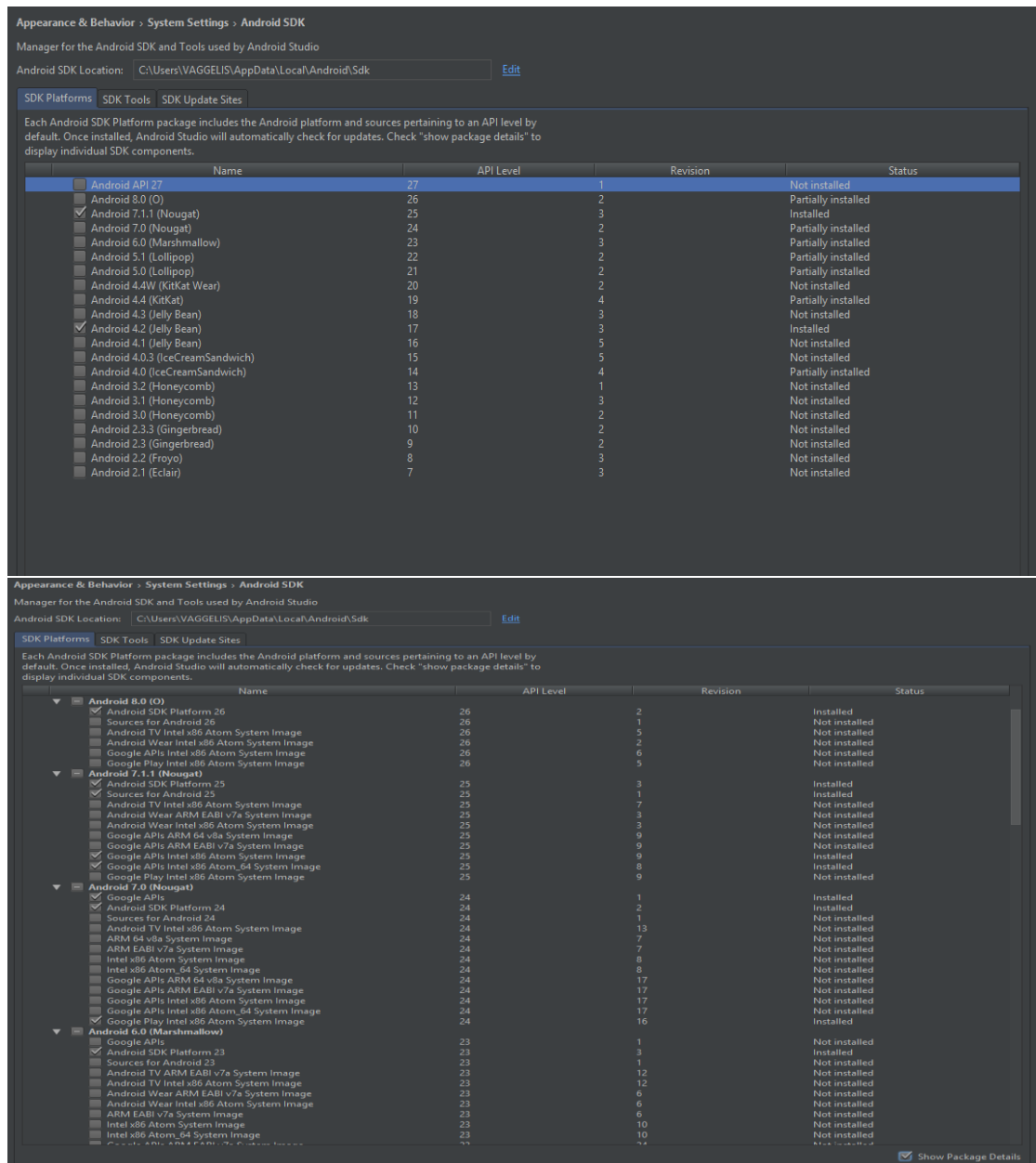
Για να δημοσιεύσουμε μια εφαρμογή (σύμφωνα με [2]) χρειαζόμαστε το .apk αρχείο που θα μας δώσει ο compiler εφόσον περάσει επιτυχώς την διαδικασία της αποσφαλμάτωσης. Αν θέλουμε να δημοσιεύσουμε την εφαρμογή μας στο Google Play Store θα πρέπει να κάνουμε εγγραφή σαν προγραμματιστές και στην συνέχεια δίνοντας μια περιγραφή και μερικά screenshots η εφαρμογή μας ελέγχεται από την Google κι αν περάσει τον έλεγχο δημοσιεύεται και είναι διαθέσιμη για κατέβασμα από τον

χρήστη. Την στιγμή που γράφουμε την παρούσα πτυχιακή εργασία δεν είναι διαθέσιμη η εγγραφή εμπόρου στην Ελλάδα δηλαδή οι προγραμματιστές εφόσον καταβάλλουν τα 25\$ που είναι το registration fee της Google για developer registration θα μπορούν να δημοσιεύουν εφαρμογές αλλά θα διατίθενται δωρεάν στο ευρύ κοινό. Αυτό σημαίνει ότι δεν θα μπορούν να έχουν κέρδος από τα downloads της εφαρμογής παρά μόνο από τα in-app purchases (για παράδειγμα η αναβάθμιση μια εφαρμογής από trial σε pro έκδοση μέσω paypal ή google wallet) ή το admob (διαδικασία πληρωμής από διαφημίσεις της Google μέσα στην εφαρμογή).

2.4 Android System Development Kit

Για να μπορέσει ο προγραμματιστής να δημοσιεύσει μια εφαρμογή η οποία θα είναι διαθέσιμη στο κοινό θα πρέπει να χρησιμοποιήσει κάποιες βιβλιοθήκες και εργαλεία που είναι διαθέσιμα μέσω του Android SDK. Επίσης θα μπορεί να προβεί σε διαθέσιμες ενημερώσεις ανάλογα με τις εκδόσεις του λειτουργικού συστήματος. Μπορεί να κατεβάσει τα απαραίτητα αυτά εργαλεία από τον SDK Manager που είναι μία built-in επέκταση σε κάθε περιβάλλον προγραμματισμού που θα επιλέξει να χρησιμοποιήσει.

Το πως είναι διαθέσιμα τα πακέτα ανάλογα με τα API's, μπορούμε να δούμε στην παρακάτω εικόνα:

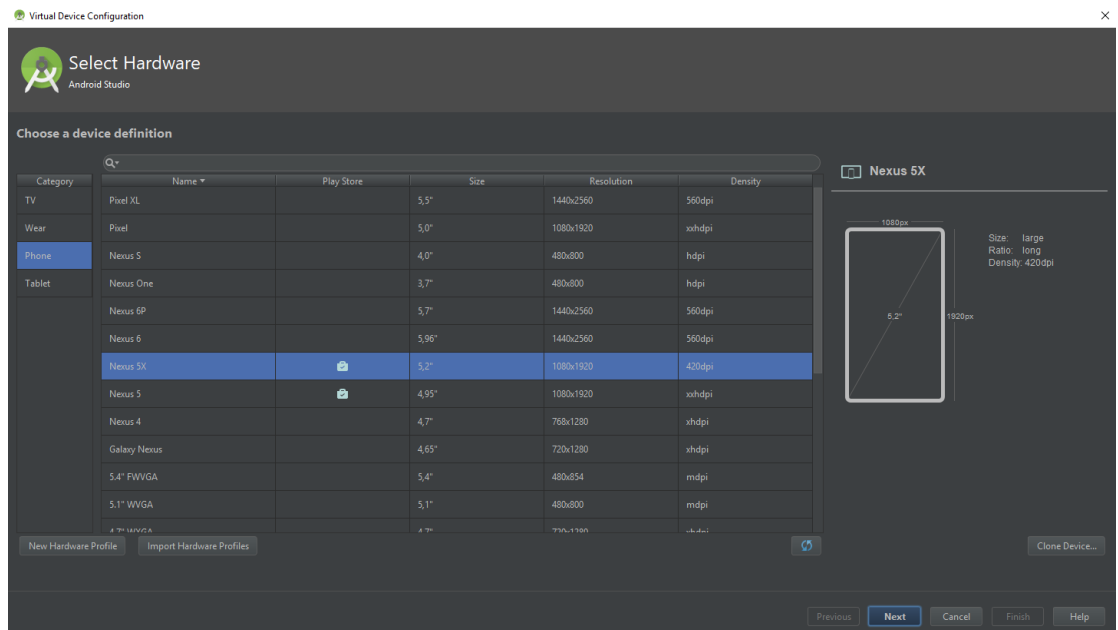


Εικόνα 35: Διαχείριση εικονικών συσκευών

2.5 Android Virtual Device Manager(AVD)

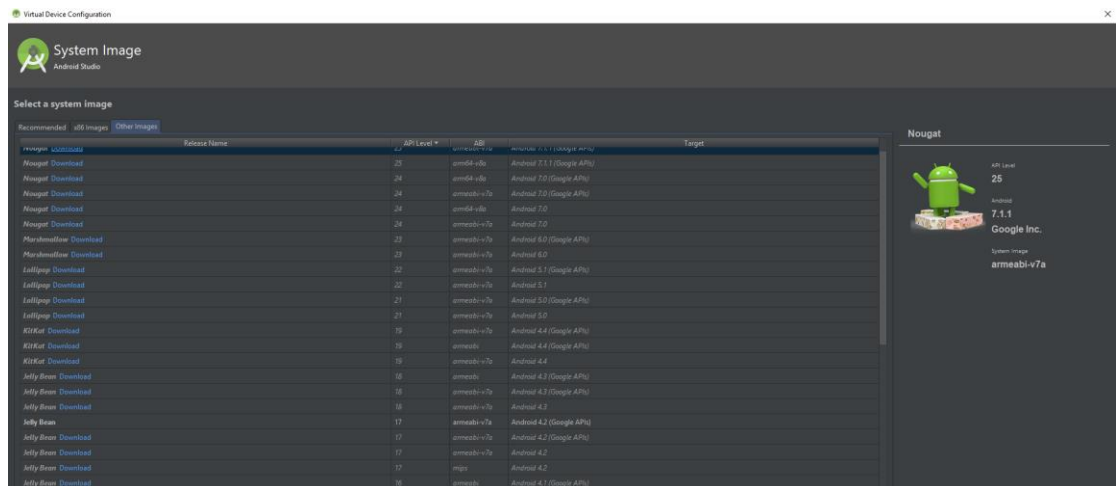
Επίσης θα πρέπει να κατεβάσουμε όλα τα απαραίτητα αρχεία για την δημιουργία εικονικής συσκευής με σκοπό να τεστάρουμε την εφαρμογής μας. Δουλειά η οποία αναλαμβάνει το AVD Android Virtual Device Manager με το κατάλληλο SDK και έκδοση του λειτουργικού συστήματος και με τον κατάλληλο επεξεργαστή. Βεβαίως θα πρέπει να έχουμε εγκατεστημένο το API που θέλουμε να τρέξει και να διαλέξουμε τις ρυθμίσεις οθόνης, μέγεθος μνήμης, χρήση της κάρτας γραφικών του HOST για καλύτερες επιδόσεις ιδιαίτερα σε εφαρμογές με μέγιστη απόδοση γραφικών ή σε παιχνίδια. Παρακάτω θα δούμε πως είναι το AVD:

Διαλέγουμε την συσκευή από την λίστα

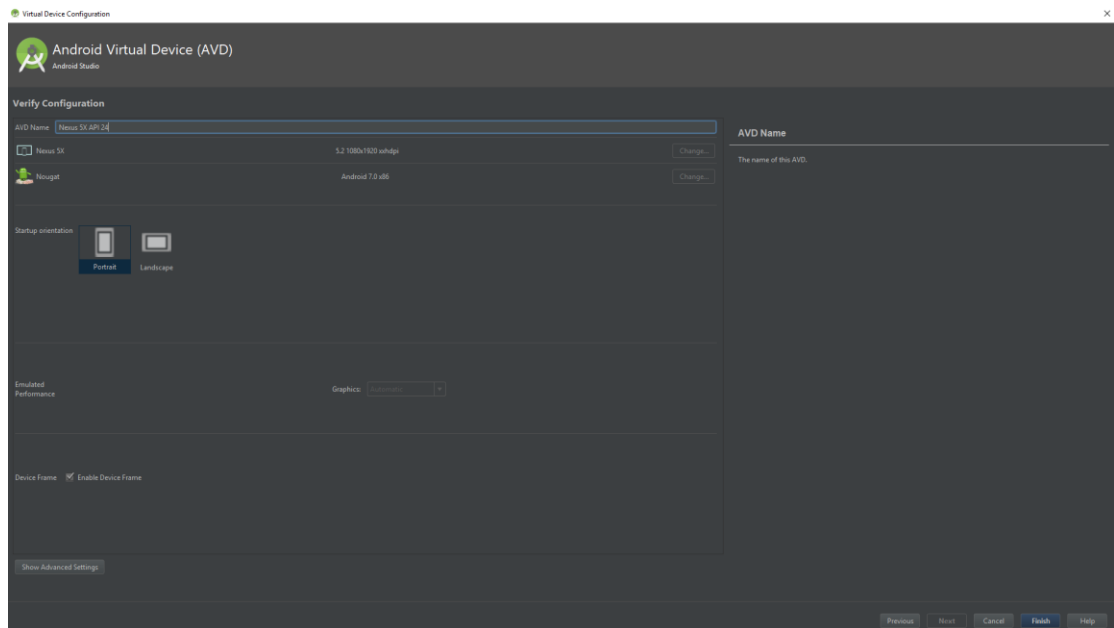


Εικόνα 36: AVD Manager Device Selection

Επιλέγουμε το SDK και την επιθυμητή έκδοση, καθώς και τον επεξεργαστή που θέλουμε να τρέχει η εικονική συσκευή μας.



Εικόνα 37: SDK API Selection



Εικόνα 38: Προσθήκη εικονικής συσκευής

Και κάπως έτσι έχουμε δημιουργήσει μία εικονική συσκευή, την οποία θα χρησιμοποιούμε προκειμένου να κάνουμε άμεσα τις δοκιμές και του απαραίτητους ελέγχους στην εφαρμογή μας.



Εικόνα 39: Εικονική συσκευή προσομοίωσης (AVD)

2.6 Κατευθυντήριες γραμμές σχεδιασμού Android (Design Guidelines)

Με γνώμονα τις βασικές αρχές σχεδιασμού του Android, καταφέρνουμε να δημιουργήσουμε το οπτικό κομμάτι με το οποίο θα έρχεται σε επαφή ο χρήστης. Θα ήταν μάταιο να χρησιμοποιήσουμε αυθαίρετα δικές μας ιδέες έναντι του υλικού που

μας προσφέρει απλόχερα η ίδια η εταιρεία που κατασκεύασε το λειτουργικό σύστημα στο οποίο θα τρέχει η εφαρμογή μας. Γιατί ως γνωστόν μια κακοσχεδιασμένη εφαρμογή απαιτεί περισσότερους πόρους από τη συσκευή που την φιλοξενεί, με αποτέλεσμα να είναι μη παραγωγική και σε πολλές περιπτώσεις να διακόπτεται ακόμη και από το ίδιο το σύστημα. Βασιζόμενοι λοιπόν στο [26] και στο [10] θα αναπτύξουμε την παρακάτω εφαρμογή, η οποία θα υποστηρίζεται σε παλαιότερες εκδόσεις του λειτουργικού συστήματος και σε πολλαπλές συσκευές με διάφορα μεγέθη οθονών. Παρότι οι νεότερες συσκευές που βγαίνουν στην αγορά είναι εξοπλισμένες με όλο και πιο σύγχρονους επεξεργαστές και με περισσότερη μνήμη, εμείς πρέπει να έχουμε ως βασική αρχή την ελάχιστη κατανάλωση πόρων από το σύστημα.

2.6.1 Υποστήριξη οθονών

Το λειτουργικό Android χρησιμοποιείται από πολλές κινητές συσκευές με οθόνες διαφορετικού μεγέθους και ανάλυσης. Ένα πρόβλημα που προκύπτει από το παραπάνω γεγονός δημιουργείται όταν πρέπει να εμφανίζεται σωστά μια γραφική διεπαφή μιας Android δραστηριότητας όταν εκτελείται σε διαφορετικού μεγέθους οθόνες. Για το λόγο αυτό προτείνεται η κατασκευή γραφικών διεπαφών με τη χρήση ιδιοτήτων προσφερόμενων από το Android SDK, όπως `wrap_content` και `match_parent`, ώστε να εξασφαλιστεί η καλύτερη δυνατή προβολή τους. Κατασκευάζοντας γραφικές διεπιφάνειες χρήστη με τη χρήση μονάδων μέτρησης, όπως `pixels`, θα οδηγήσει σχεδόν σίγουρα στη λάθος προβολή της γραφικής διεπαφής σε κάποιες οθόνες. Για να λυθεί αυτό το πρόβλημα χρησιμοποιείται όπου είναι απαραίτητο η μονάδα μέτρησης της πυκνότητας της οθόνης (`Density`).

- ❖ *Ως πυκνότητα οθόνης ορίζεται η αναλογία μεταξύ της ανάλυσής της και του μεγέθους της. Ως μονάδα μέτρησης αυτής της αναλογίας έχουν οριστεί τα σημεία ανά ίντσα (`dots per inch - dpi`).*

Με βάση την παραπάνω μονάδα μέτρησης έχει δημιουργηθεί η εικονική μονάδα μέτρησης εικονοστοιχείων ανεξάρτητων πυκνότητας οθόνης (`density independent pixels - dip`), η οποία αναπαριστά μια αναλογία μεταξύ ενός `pixel` σε μία οθόνη με πυκνότητα `160dpi` και αλλάζει ανάλογα με την πυκνότητα της οθόνης στην οποία εκτελείται η εφαρμογή. Έτσι στις Android εφαρμογές προτείνεται όπου είναι απαραίτητο, όπως για παράδειγμα στον προσδιορισμό του μεγέθους ενός `Space`, η χρήση `dp` αντί για χρήση `pixels`.

Όταν θέλουμε να δημιουργήσουμε μία εφαρμογή που να υποστηρίζει πολλές συσκευές θα πρέπει να λάβουμε υπόψη μας τα διάφορα μεγέθη τα οποία βλέπουμε παρακάτω:

Screen characteristic	Qualifier	Description
Size	<code>small</code>	Resources for <i>small</i> size screens.
	<code>normal</code>	Resources for <i>normal</i> size screens. (This is the baseline size.)
	<code>large</code>	Resources for <i>large</i> size screens.
	<code>xlarge</code>	Resources for <i>extra-large</i> size screens.
Density	<code>ldpi</code>	Resources for low-density (<i>ldpi</i>) screens (~120dpi).
	<code>mdpi</code>	Resources for medium-density (<i>mdpi</i>) screens (~160dpi). (This is the baseline density.)
	<code>hdpi</code>	Resources for high-density (<i>hdpi</i>) screens (~240dpi).
	<code>xhdpi</code>	Resources for extra-high-density (<i>xhdpi</i>) screens (~320dpi).
	<code>xxhdpi</code>	Resources for extra-extra-high-density (<i>xxhdpi</i>) screens (~480dpi).
	<code>xxxhdpi</code>	Resources for extra-extra-extra-high-density (<i>xxxhdpi</i>) screens (~640dpi).
	<code>nodpi</code>	Resources for all densities. These are density-independent resources. The system does not scale resources tagged with this qualifier, regardless of the current screen's density.
	<code>tvdpi</code>	Resources for screens somewhere between <i>mdpi</i> and <i>hdpi</i> ; approximately 213dpi. This is not considered a "primary" density group. It is mostly intended for televisions and most apps shouldn't need it—providing <i>mdpi</i> and <i>hdpi</i> resources is sufficient for most apps and the system will scale them as appropriate. If you find it necessary to provide <i>tvdpi</i> resources, you should size them at a factor of 1.33* <i>mdpi</i> . For example, a 100px x 100px image for <i>mdpi</i> screens should be 133px x 133px for <i>tvdpi</i> .
Orientation	<code>land</code>	Resources for screens in the landscape orientation (wide aspect ratio).
	<code>port</code>	Resources for screens in the portrait orientation (tall aspect ratio).
Aspect ratio	<code>long</code>	Resources for screens that have a significantly taller or wider aspect ratio (when in portrait or landscape orientation, respectively) than the baseline screen configuration.
	<code>notlong</code>	Resources for use screens that have an aspect ratio that is similar to the baseline screen configuration.

Εικόνα 40: Χαρακτηριστικά οθονών

	Low density (120), <i>ldpi</i>	Medium density (160), <i>mdpi</i>	High density (240), <i>hdpi</i>	Extra-high-density (320), <i>xhdpi</i>
<i>Small screen</i>	QVGA (240x320)		480x640	
<i>Normal screen</i>	WQVGA400 (240x400) WQVGA432 (240x432)	HVGA (320x480)	WVGA800 (480x800) WVGA854 (480x854) 600x1024	640x960
<i>Large screen</i>	WVGA800 (480x800) WVGA854 (480x854)	WVGA800 (480x800) WVGA854 (480x854) 600x1024		
<i>Extra-Large screen</i>	1024x600	WXGA (1280x800) [†] 1024x768 1280x768	1536x1152 1920x1152 1920x1200	2048x1536 2560x1536 2560x1600

Εικόνα 41: Μεγέθη οθονών

Υπάρχουν διάφορες μονάδες μέτρησης μεγέθους οθόνης όπως είναι

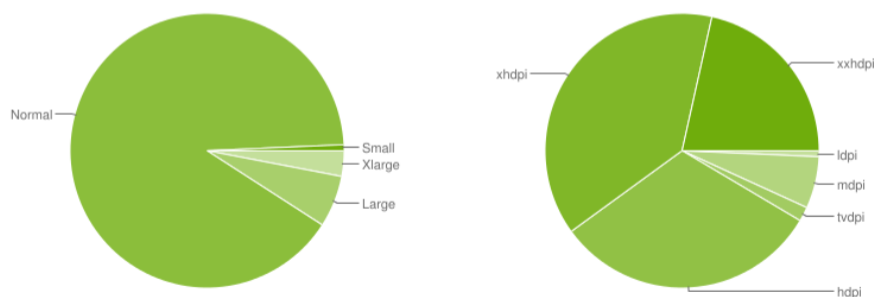
- Px είναι το pixel
- Sp είναι scale-independent pixels
- Dip είναι density-independent pixels
- In είναι η ίντσα και αντιστοιχεί σε πραγματικές διαστάσεις οθόνης 1inch = 2.54 cm
- Pt είναι points και αντιστοιχεί σε πραγματικές διαστάσεις οθόνης 1pt = 1/72 inch
- Dp είναι density-independent pixels αλλά είναι μια abstract μονάδα μέτρησης για το μέγεθος. 1dp = 1px in 160 dpi screen

Density Bucket	Screen Density	Physical Size	Pixel Size
<code>ldpi</code>	120 dpi	0.5 x 0.5 in	0.5 in * 120 dpi = 60 x 60 px
<code>mdpi</code>	160 dpi	0.5 x 0.5 in	0.5 in * 160 dpi = 80 x 80 px
<code>hdpi</code>	240 dpi	0.5 x 0.5 in	0.5 in * 240 dpi = 120 x 120 px
<code>xhdpi</code>	320 dpi	0.5 x 0.5 in	0.5 in * 320 dpi = 160 x 160 px
<code>xxhdpi</code>	480 dpi	0.5 x 0.5 in	0.5 in * 480 dpi = 240 x 240 px

Εικόνα 42: Μονάδες μέτρησης μεγέθους οθόνης

Εδώ βλέπουμε ένα πίνακα που συγκεντρώνει δεδομένα σχετικά με τον αριθμό συσκευών που έχουν μια συγκεκριμένη διαμόρφωση οθόνης, η οποία ορίζεται από ένα συνδυασμό μεγέθους οθόνης και πυκνότητας.

	ldpi	mdpi	tvdpi	hdpi	xhdpi	xxhdpi	Total
Small	0.6%					0.1%	0.7%
Normal		1.4%	0.2%	30.5%	37.1%	21.0%	90.2%
Large	0.1%	2.7%	1.5%	0.5%	0.9%	0.4%	6.1%
Xlarge		2.0%		0.5%	0.5%		3.0%
Total	0.7%	6.1%	1.7%	31.5%	38.5%	21.5%	



Data collected during a 7-day period ending on November 9, 2017.

Any screen configurations with less than 0.1% distribution are not shown.

Εικόνα 43: Ποσοστά συσκευών με συγκεκριμένη διαμόρφωση

2.7 Android Crash Report Manager – Android Logcat

Σημαντικό ρόλο σε ένα προγραμματιστικό περιβάλλον κατέχουν τα εργαλεία που μας πληροφορούν σχετικά με τη λειτουργία της εφαρμογής που αναπτύσσουμε. Εάν κατά τη διαδικασία εκτέλεσης της εφαρμογής προκύψει κάποιο σφάλμα ή ανιχνευθεί μη ορθή λειτουργία της, το Android Monitor με τη βοήθεια του logcat (το οποίο αναφέραμε πιο λεπτομερώς παραπάνω) μας επισημαίνει που βρίσκεται το θέμα προς επίλυση. Γενικά μας παρέχει πληροφορίες σχετικά με κατανάλωση πόρων, μνήμης και χώρου που καταλαμβάνει στην συσκευή. Μας ενημερώνει για τα permissions που δίνουμε στην εφαρμογή καθώς και για το αν το thread μας διεκπεραιώνει μεγάλο φόρτο εργασίας παρότι εκτελείται επιτυχώς. Ένα παράδειγμα βλέπουμε παρακάτω:

2.8.1 Κοινόχρηστες προτιμήσεις – Shared Preferences

Η κλάση `SharedPreferences` αποτελεί μία πολύ αποτελεσματική μέθοδο αποθήκευσης δεδομένων μικρής χωρητικότητας. Οι τύποι των δεδομένων τους οποίους μπορεί ο προγραμματιστής να διαχειριστεί κάνοντας χρήση της `SharedPreferences` είναι πρωταρχικού τύπου, δηλαδή αλφαριθμητικές τιμές (τύπου `Boolean`, `float`, `int`, `long` και `string`). Κάθε τιμή συνδέεται με ένα κλειδί (μεταβλητή τύπου `string`) (key value model).

Στην κλάση `Activity` περιλαμβάνονται δύο μέθοδοι οι οποίες επιτρέπουν την πρόσβαση στις κοινόχρηστες προτιμήσεις. Αυτές είναι η `getPreferences()` και η `getSharedPreferences()`. Η μεταξύ τους διαφορά έχει να κάνει με το πλήθος των ομάδων προτίμησης, των ομαδοποιημένων δηλαδή προτιμήσεων, που χρειάζεται να καλέσουμε. Σε αντίθετη περίπτωση, καλώντας δηλαδή διάφορες ομάδες, χρησιμοποιείται η μέθοδος `getPreferences()` δίνοντας ως όρισμα την ομάδα η οποία είναι προς χρήση.

Για την καλύτερη κατανόηση της χρήσης της `SharedPreferences` ας υποθέσουμε ότι κατά την έναρξη μιας εφαρμογής εκτελείται ένας μετρητής ο οποίος γράφει τις φορές που έχει τερματιστεί ένα `Activity`.

```
Public void onCreate (Bundle savedInstanceState){
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    SharedPreferences myprefs = getPreferences(0);
    int start_times = myprefs.getInt("metritis", 0);
}
```

Για να μετρηθεί ο αριθμός της συγκεκριμένης δραστηριότητας θα πρέπει να εκτελεστεί η μέθοδος `getPreferences()` κατά την εκτέλεση της `onStop()`, με σκοπό τη λήψη της προηγούμενης τιμής του μετρητή. Στη συνέχεια θα πρέπει να δημιουργηθεί ένας `Editor` με τον οποίο θα αυξηθεί η τιμή του μετρητή κατά μία μονάδα. Τέλος, για την επικύρωση των αλλαγών στον `editor`, γίνεται με τη χρήση της `commit()`.

```
Protected void onStop(){
    SharedPreferences myprefs = getPreferences(MODE_PRIVATE);
    // Δημιουργία του Editor για επεξεργασία των προτιμήσεων
    SharedPreferences.Editor editor = myprefs.edit();

    //Αποθήκευση της ήδη υπάρχουσας τιμής στην μεταβλητή start_times
    Int start_times = myprefs.getInt("metritis",0);
    // Αύξηση της τιμής κατά ένα μέσω του Editor
    editor.putInt("metritis",start_times+1);
    // Επικύρωση της αλλαγής
    editor.commit();
    super.onStop();
}
```

2.8.2 Βάση δεδομένων SQLite

Άλλη μία μέθοδος που προσφέρει το Android για την αποθήκευση δεδομένων, είναι η χρήση της SQLite. Πρόκειται για ένα σύστημα διαχείρισης βάσης δεδομένων το οποίο δανείζεται χαρακτηριστικά από άλλες σχεσιακές βάσεις δεδομένων, όπως τη MySQL και την PostgreSQL, υποστηρίζοντας έτσι τη γλώσσα ερωτημάτων SQL.

Το πακέτο `android.database` περιέχει όλες τις απαραίτητες κλάσεις για τη διαχείριση βάσεων δεδομένων. Για τη χρήση SQLite κλάσεων θα πρέπει να μεταβούμε στο πακέτο `SQLiteOpenHelper`.

Αν και δεν είναι απαραίτητη, ενδείκνυται η χρήση υποκλάσεων του `SQLiteOpenHelper` το οποίο θα αντικαταστήσει τη μέθοδο του `onCreate()`. Με αυτόν τον τρόπο κάθε φορά που θα χρειαστεί πρόσβαση στη βάση και σε περίπτωση που δεν υπάρχει, έχουμε τη δυνατότητα δημιουργίας της δομής της και εισαγωγής αρχικών δεδομένων σε αυτή. Επιπροσθέτως, έχουμε τη δυνατότητα μέσω χρήσης της μεθόδου `onUpgrade()`, να ενημερώσουμε τη δομή και τα δεδομένα της βάσης όποτε χρειαστεί.

2.8.2.1 Δημιουργία Βάσης SQLite

Για την καλύτερη κατανόηση των παραπάνω, θα δημιουργήσουμε μία βάση δεδομένων SQLite η οποία θα αποτελείται από έναν πίνακα στον οποίο ο χρήστης θα αποθηκεύει κάποιες φωτογραφίες με βάση το link URL τους.

Το ID θα αποθηκεύει το πρωτεύον κλειδί της κάθε φωτογραφίας(URL link) που θα αποθηκεύει ο χρήστης.

Αρχικά θα πρέπει να δημιουργήσουμε μία κλάση (`ImageLinksCountDatabase.java`) η οποία θα δημιουργήσει τη βάση. Παρακάτω βλέπουμε αποσπάσματα από τον κώδικα στον οποίο χρησιμοποιήσαμε την SQLite για την τρέχουσα εφαρμογή.

```
public class ImageLinksCountDatabase extends SQLiteOpenHelper {

    private static final int DATABASE_VERSION = 1;

    private static final String DATABASE_NAME = "abc";

    private static final String TABLE_LINKS = "link_items";

    private static final String KEY_ID = "id";
    private static final String KEY_URL = "url";
    private static final String KEY_TIMES_SEEN = "times_seen";

    ImageLinksCountDatabase(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String CREATE_ITEMS_TABLE = String.format("CREATE TABLE %s (%s INTEGER PRIMARY KEY, %s TEXT, %s INTEGER)",
            TABLE_LINKS, KEY_ID, KEY_URL, KEY_TIMES_SEEN);
        Log.d("DEBUG", CREATE_ITEMS_TABLE);
    }
}
```



```
        db.execSQL(CREATE_ITEMS_TABLE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
        if (newVersion == 1) {
            db.execSQL("DROP TABLE IF EXISTS " + TABLE_LINKS);
            onCreate(db);
        }
    }
}
```

Η χρήση της `execSQL` μας επιτρέπει την άμεση χρήση SQL ερωτημάτων προς τη βάση.

2.8.2.2 Διαχείριση δεδομένων σε βάση SQLite

Για τη διαχείριση των δεδομένων θα πρέπει να δημιουργηθεί μία σειρά από μεθόδους οι οποίες θα είναι ιδιαίτερα χρήσιμες στην εκτέλεση των διεργασιών «CRUD» (Create Read Update Delete). Μέσω αυτών δίνεται η δυνατότητα στον προγραμματιστή να προσθέσει, να διαβάσει, να επεξεργαστεί αλλά και να διαγράψει εγγραφές από τη βάση.

Πριν τη δημιουργία των συγκεκριμένων μεθόδων θα πρέπει να δημιουργηθεί μια άλλη κλάση (`ImageLinkCount.java`) η οποία θα περιέχει μεθόδους (getter/setters methods) που θα βοηθήσουν στην προσπέλαση των private μεταβλητών της κλάσης `ImageLinkCount`.

```
public class ImageLinkCount {
    private long id;
    private String URL;
    private int timesSeen;

    public ImageLinkCount(String URL, int timesSeen) {
        setURL(URL);
        setTimesSeen(timesSeen);
    }

    public long getId() {
        return id;
    }

    public void setId(long id) {
        this.id = id;
    }

    public String getURL() {
        return URL;
    }

    public void setURL(String URL) {
        this.URL = URL;
    }

    public int getTimesSeen() {
        return timesSeen;
    }
}
```



```
public void setTimesSeen(int timesSeen) {
    this.timesSeen = timesSeen;
}

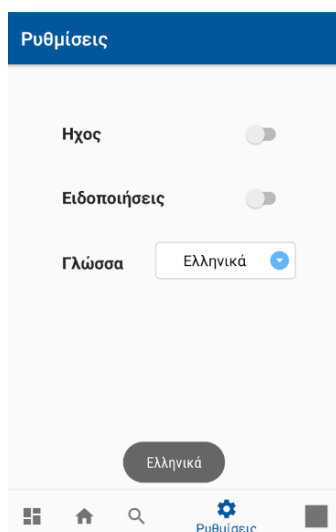
public void incrementTimesSeen() {
    this.timesSeen += 1;
}
}
```

Αφού δημιουργηθεί η παραπάνω κλάση θα πρέπει να κατασκευαστούν οι απαραίτητες μέθοδοι οι οποίοι θα βοηθήσουν στην εκτέλεση των διαδικασιών της ανάγνωσης και εγγραφής δεδομένων από και προς τη βάση.

2.9 Παράθυρα ειδοποιήσεων (Alert Dialogs) και σύντομα μηνύματα (Toasts)

Τα παράθυρα ειδοποιήσεων (AlertDialogs) χρησιμοποιούνται συνήθως όταν ο προγραμματιστής θέλει να ενημερώσει το χρήστη για ένα γεγονός που έλαβε ή πρόκειται να λάβει χώρα σε μία δραστηριότητα.

Το σύντομο μήνυμα (Toast) δίνει τη δυνατότητα στον προγραμματιστή να εμφανίσει ένα μικρό, σύντομο μήνυμα στο χρήστη για ορισμένο χρόνο χωρίς να επηρεάζεται η ανανέωση της γραφικής διεπαφής και ο κύκλος ζωής των δραστηριοτήτων. Παρακάτω βλέπουμε μερικά παραδείγματα από τη χρήση των Toasts μέσα στην εφαρμογή.



Εικόνα 46: Toast για επιλεγμένη γλώσσα



Εικόνα 47: Toast για αφαίρεση προϊόντος



Εικόνα 48: Toast για ενημέρωση κίνησης

Κεφάλαιο 3

3.1 Περιγραφή της εφαρμογής

Στις μέρες μας, με αφορμή τη ραγδαία εξέλιξη της τεχνολογίας, οι ανάγκες για αγορά και πώληση προϊόντων όλο και μεγαλώνουν. Παράλληλα όμως με την οικονομική κρίση που επικρατεί σε πολλές περιοχές ανά τον κόσμο, μεγαλώνει και η δυσκολία των ανθρώπων στο να τα αποκτήσουν. Εδώ λοιπόν έρχεται η εφαρμογή μας να δώσει τη λύση, σε ανθρώπους που θέλουν να πουλήσουν κάποιο προϊόν που δε χρειάζονται έναντι κάποιου ποσού, ή να αγοράσουν αν θέλουν αντίστοιχα κάποιο μεταχειρισμένο ίσως, εξοικονομώντας με αυτόν τον τρόπο αρκετά χρήματα.

Μπορεί ο καθένας λοιπόν να «εμπορευτεί» τα πράγματά του με απώτερο σκοπό την άμεση εξυπηρέτηση των αναγκών του και την εξοικονόμηση των χρημάτων του. Βάζοντας μερικές πληροφορίες και μία φωτογραφία του προϊόντος ή της υπηρεσίας που αναζητεί ο χρήστης, μπαίνει σε μία απλή διαδικασία αναζήτησης και εύρεσης αυτού που ψάχνει ή θέλει να πουλήσει.

3.2 Διαδικασία χρήσης και περιεχόμενο της εφαρμογής

Έχοντας στα χέρια του κανείς ένα έξυπνο τηλέφωνο (smartphone) και πατώντας πάνω στην εφαρμογή οδηγείται σε αυτή την οθόνη όπου καλείται να επιλέξει σε ποια γλώσσα επιθυμεί να χρησιμοποιήσει την εφαρμογή.

Στην συνέχεια πατάει το κουμπί εκκίνησης και μεταβαίνει στην οθόνη εισόδου όπου και εκεί του δίνεται η επιλογή να διαλέξει αν θέλει να δημιουργήσει ένα απλό λογαριασμό προκειμένου να μπαίνει στην εφαρμογή κάθε φορά με τα προσωπικά του στοιχεία. Σε περίπτωση που θέλει να παραλείψει αυτό το βήμα μπορεί απλώς να πατήσει «Without Login» και να αποκτήσει πρόσβαση στο πλήρες περιεχόμενο της εφαρμογής.

Μπαίνοντας λοιπόν αντικρίζουμε ένα Μενού με πέντε διαφορετικές καρτέλες (Tabs). Η πρώτη καρτέλα περιέχει τις δύο κατηγορίες που προκύπτουν άλλωστε και από το όνομα της εφαρμογής. Επιλέγουμε την πρώτη κατηγορία σε περίπτωση που θέλουμε να θέσουμε ένα προϊόν προς πώληση, βάζοντας τις απαραίτητες πληροφορίες που θα βοηθήσουν τον πιθανό αγοραστή και θα τον φέρουν σε επαφή με εμάς.

Τη δεύτερη κατηγορία επιλέγουμε σε περίπτωση που αναζητούμε ένα προϊόν ή μία παροχή υπηρεσίας, βάζοντας πάλι τις απαραίτητες πληροφορίες προκειμένου να γίνουμε αντιληπτοί από το κοινό που διαβάζει την κατά κάποιο τρόπο αγγελία μας και να έρθει σε επικοινωνία μαζί μας.

Και στις δύο κατηγορίες τα πεδία που καλείται να συμπληρώσει ο χρήστης είναι τα ίδια. Όπως βλέπουμε και παρακάτω στην εικόνα τα πεδία είναι τα εξής:

- Όνομα προϊόντος
- Όνομα του διαπραγματευτή
- Κατηγορία
- Τιμή
- Τοποθεσία

- Ηλεκτρονική διεύθυνση επικοινωνίας (email)
- Τηλέφωνο
- Περιγραφή προϊόντος
- Προσθήκη φωτογραφίας

Πατώντας το κουμπί OK, εμφανίζεται ένα μήνυμα που μας λέει «Successful» και στη συνέχεια αν θέλουμε να συνεχίσουμε την περιήγησή μας στην εφαρμογή πατάμε το κουμπί «Main Menu».

Η δεύτερη καρτέλα είναι η αρχική σελίδα όπου βρίσκονται τα αγαπημένα προϊόντα που έχουμε ξεχωρίσει και επιλέξει. Μπορούμε λοιπόν πατώντας το μεγάλο κουμπί να μεταβούμε εκεί και να επιλέξουμε κάποιο από όλα να το κάνουμε «Κράτηση». Με τον όρο αυτό εννοούμε ότι θα πληροφορήσουμε τον αντίστοιχο διαπραγματευτή σχετικά με την επιλογή μας, εκδηλώνοντας το ενδιαφέρον μας. Αυτό μπορεί να γίνει με δύο τρόπους. Ο ένας τρόπος είναι να συμπληρώσουμε απλώς τα στοιχεία μας και να πατήσουμε OK, περιμένοντας να έρθει σε επικοινωνία με εμάς ο διαπραγματευτής. Ο άλλος τρόπος είναι να το διεκδικήσουμε (σε περίπτωση που το θέλουμε οπωσδήποτε) δίνοντας μία μικρή προκαταβολή στον διαπραγματευτή, κάνοντας χρήση της υπηρεσίας Paypal. Παρακάτω εξηγώ ακριβώς από που εμπνεύστηκα αυτή τη λειτουργία.

Η τρίτη καρτέλα είναι η αναζήτηση προϊόντων-υπηρεσιών. Επιλέγουμε την πρώτη ταμπέλα «FOR SALE» σε περίπτωση που ψάχνουμε κάποιο προϊόν αν το πουλάει κάποιος άλλος χρήστης προκειμένου να το αγοράσουμε (αν μας αρέσει) σε προνομιακή τιμή.

Επιλέγουμε τη δεύτερη ταμπέλα «WANTED» σε περίπτωση που θέλουμε να δούμε τι ψάχνει ο κόσμος, μήπως το έχουμε εμείς και θέλουμε να το πουλήσουμε προκειμένου να κερδίσουμε ένα έξτρα εισόδημα.

Στη συνέχεια επιλέγουμε ποια κατηγορία μας ενδιαφέρει και συνεχίζουμε την αναζήτησή μας μέχρις ότου βρούμε αυτό που θέλουμε. Όταν το βρούμε, έχουμε τη δυνατότητα να επικοινωνήσουμε με τον διαπραγματευτή από τα στοιχεία που έχει παραθέσει στις πληροφορίες, προκειμένου να πάρουμε περισσότερες πληροφορίες. Έπειτα επιλέγουμε προσθήκη στα αγαπημένα. Εάν θέλουμε να ολοκληρώσουμε την αναζήτησή μας πατάμε το αστεράκι πάνω δεξιά ή συνεχίζουμε την αναζήτησή μας μέχρι να βρούμε κι άλλα προϊόντα και να αποφασίσουμε στο τέλος ποια θα κρατήσουμε και ποια όχι. Πατώντας λοιπόν το αστεράκι βλέπουμε τι έχουμε βάλει στα αγαπημένα και ανάλογα ποιο μας ενδιαφέρει περισσότερο πατάμε το αντίστοιχο κουμπί. Εάν επιλέξουμε ότι το θέλουμε, η διαδικασία πάλι είναι απλή (όπως την περιγράψαμε και πριν).

Στην τέταρτη καρτέλα είναι οι ρυθμίσεις της εφαρμογής οι οποίες θα μπορούσαν να είναι ενεργές στην περίπτωση που φτάσει ποτέ στην αγορά της Google. Μπορούμε να ρυθμίσουμε αν θέλουμε να μας έρχονται ειδοποιήσεις για νέα προϊόντα που εισάγονται στην εφαρμογή ή για κάποια πιθανή αλλαγή τιμής ενός προϊόντος από τον διαπραγματευτή κλπ.

Στην πέμπτη καρτέλα είναι τα «Credits» όπως γράφει, παρουσιάζοντας τον εξολοκλήρου δημιουργό της εφαρμογής (εμένα), με κάποιες βασικές πληροφορίες.

Αυτή είναι εν ολίγοις η εφαρμογή που αποσκοπεί στη διευκόλυνση αγοραπωλησιών και την εξοικονόμηση-απόκτηση χρημάτων στην οικονομικά κρίσιμη εποχή που διανύουμε.

3.3 Διεκδίκηση προϊόντος με προκαταβολή μέσω PayPal

Με αφορμή την πρόσφατη επέκταση της ψηφιακής συνεργασίας μεταξύ PayPal και Mastercard στην Ελλάδα, πήρα την απόφαση να εμπλουτίσω την εφαρμογή με ένα πρόσθετο χαρακτηριστικό.

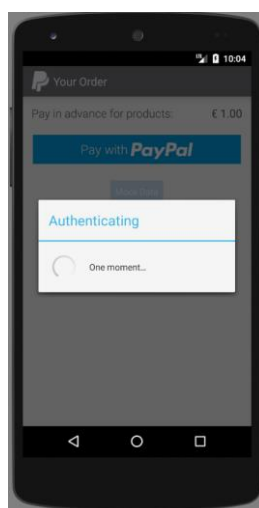
Όπως πληροφορήθηκα από το δελτίο τύπου της PayPal, εκείνη και η Mastercard ανακοίνωσαν πρόσφατα μια σημαντική επέκταση της μακροχρόνιας συνεργασίας τους, συμπεριλαμβάνοντας την Ευρώπη [και την Ελλάδα], τον Καναδά, τη Λατινική Αμερική, την Καραϊβική, την Μέση Ανατολή και την Αφρική, προκειμένου να αυξήσουν τις επιλογές των πελατών, να βελτιστοποιήσουν την εμπειρία του καταναλωτή και να ανάγουν την Mastercard σε μια ξεκάθαρη επιλογή πληρωμής μέσω της PayPal ανά τον κόσμο. Με την προσθήκη των νέων αυτών αγορών – και σε συνέχεια της πρόσφατης διεύρυνσης της συνεργασίας τους στις ΗΠΑ και στις χώρες της περιοχής Ασίας-Ειρηνικού – η PayPal και η Mastercard κατέληξαν σε μια παγκόσμια πλέον συμφωνία.

Όπως ίσχυε και σε προηγούμενες συμφωνίες, η διεθνής επέκταση θα δημιουργήσει μια σειρά συντονισμένων από κοινού ευκαιριών ανάπτυξης, οι οποίες θα δώσουν ώθηση στο όραμα που μοιράζεται η PayPal με την Mastercard για να προσφέρουν στους καταναλωτές μεγαλύτερο εύρος επιλογών και την απαραίτητη ευελιξία για να διαχειρίζονται και να μεταφέρουν τα χρήματά τους.

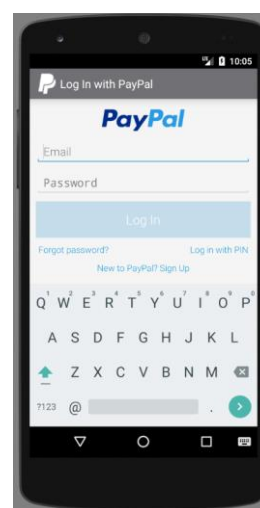
Έτσι λοιπόν ο χρήστης έχει την δυνατότητα να επιλέξει ένα προϊόν που βρίσκεται προς πώληση και να το διεκδικήσει δίνοντας μια μικρή προκαταβολή στον πωλητή χρησιμοποιώντας την υπηρεσία της PayPal.



Εικόνα 49: Είσοδος στο περιβάλλον της Paypal



Εικόνα 50: Authentication



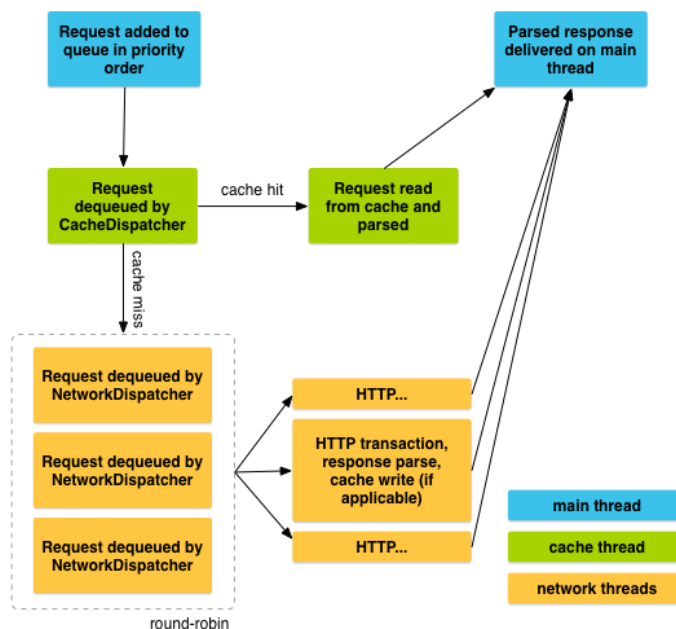
Εικόνα 51: Είσοδος στο λογαριασμό

3.4 Μεταφορά δεδομένων χρησιμοποιώντας Android Volley

Η παραπάνω λειτουργία επιτυγχάνεται με τη χρήση του Android Volley. Το Volley είναι μια βιβλιοθήκη Android της Google, η οποία μπορεί να κάνει τη ζωή του προγραμματιστή πιο εύκολη, σε περίπτωση που θέλει να ασχοληθεί με λειτουργίες του δικτύου. Ας δούμε μερικά χαρακτηριστικά της βιβλιοθήκης και που μπορεί να χρησιμοποιηθεί.

- Προγραμματίζει αυτόματα τις αιτήσεις δικτύου
- Υποστηρίζει την ιεράρχηση των αιτήσεων. Αυτό σημαίνει ότι μπορούμε να φορτώσουμε περιεχόμενο ανάλογα με τις προτεραιότητες, για παράδειγμα το κύριο περιεχόμενο μπορεί να έχει υψηλή προτεραιότητα, αλλά οι εικόνες να έχουν χαμηλότερη προτεραιότητα.
- Παρέχει διαφανή μνήμη cache και μνήμη που επιτρέπει γρήγορη επαναφόρτωση δεδομένων. Η διαφανής κρυφή μνήμη σημαίνει ότι ο καλών δεν χρειάζεται να γνωρίζει την ύπαρξη της κρυφής μνήμης. Δηλαδή η κρυφή μνήμη υλοποιείται αυτόματα. Ωστόσο υπάρχει δυνατότητα να απενεργοποιήσουμε την προσωρινή μνήμη.
- Παρέχει ένα καλό API για την ακύρωση αιτημάτων. Μπορούμε να ακυρώσουμε ένα αίτημα ή πολλά αιτήματα ανάλογα με ορισμένα φίλτρα.

Παρακάτω βλέπουμε σχηματικά πως λειτουργεί.



Εικόνα 52: Είσοδος στο λογαριασμό

```
package tp3646.evangelos.chaniotakis.forsaleorwanted.network;

import android.content.Context;
import android.graphics.Bitmap;
import android.util.LruCache;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.ImageLoader;
import com.android.volley.toolbox.Volley;

/**
 * Created by VAGGELIS on 19/10/2017.
 */
public class VolleySingleton {

    private static VolleySingleton mInstance;
    private RequestQueue mRequestQueue;
    private ImageLoader mImageLoader;
    private static Context mCtx;

    private VolleySingleton(Context context) {
        mCtx = context;
        mRequestQueue = getRequestQueue();

        mImageLoader = new ImageLoader(mRequestQueue, new ImageLoader.ImageCache() {
            private final LruCache<String, Bitmap> cache = new LruCache<String, Bitmap>(20);

            @Override
            public Bitmap getBitmap(String url) {
                return cache.get(url);
            }

            @Override
            public void putBitmap(String url, Bitmap bitmap) {
                cache.put(url, bitmap);
            }
        });
    }

    public static synchronized VolleySingleton getInstance(Context context) {
        if (mInstance == null) {
            mInstance = new VolleySingleton(context);
        }
        return mInstance;
    }

    public RequestQueue getRequestQueue() {
        if (mRequestQueue == null) {
            mRequestQueue = Volley.newRequestQueue(mCtx.getApplicationContext());
        }
        return mRequestQueue;
    }

    public <T> void addToRequestQueue(Request<T> req) {
        getRequestQueue().add(req);
    }

    public ImageLoader getImageLoader() {
        return mImageLoader;
    }
}
```

Εικόνα 53: Χρήση Volley

Το Gson είναι μία βιβλιοθήκη Java που μπορεί να χρησιμοποιηθεί για τη μετατροπή των αντικειμένων Java σε αναπαράσταση JSON. Μπορεί επίσης να χρησιμοποιηθεί για τη μετατροπή μιας συμβολοσειράς JSON σε ένα ισοδύναμο αντικείμενο Java.

Το Gson είναι ένα έργο ανοιχτού κώδικα, το οποίο μπορούμε να βρούμε στη διεύθυνση [24].

Για να χρησιμοποιήσουμε το Gson σε μία δομή Gradle, προσθέτουμε το 'com.google.code.gson:gson:2.6.2' ή μια νεότερη έκδοση ως εξάρτηση (dependency) στο αρχείο build.gradle.

Παρακάτω βλέπουμε ένα χαρακτηριστικό παράδειγμα από τον κώδικα της εφαρμογής, όπου κάναμε χρήση της Gson.

```
import com.google.gson.Gson;
import com.google.gson.JsonSyntaxException;

import java.io.UnsupportedEncodingException;
import java.util.Map;

/**
 * Created by VAGGELIS on 19/10/2017.
 */
public class GsonRequest<T> extends Request<T> {
    private Gson mGson = new Gson();
    private Class<T> clazz;
    private Map<String, String> headers;
    private Map<String, String> params;
    private Response.Listener<T> listener;

    /**
     * Make a GET request and return a parsed object from JSON.
     */
    public GsonRequest(int method, String url, Class<T> clazz, Response.Listener<T> listener, Response.ErrorListener errorListener) {
        super(method, url, errorListener);
        this.clazz = clazz;
        this.listener = listener;
        mGson = new Gson();
    }

    /**
     * Make a POST request and return a parsed object from JSON.
     */
    public GsonRequest(int method, String url, Class<T> clazz, Map<String, String> params, Response.Listener<T> listener, Response.ErrorListener errorListener) {
        super(method, url, errorListener);
        this.clazz = clazz;
        this.params = params;
        this.listener = listener;
        this.headers = null;
        mGson = new Gson();
    }

    @Override
    public Map<String, String> getHeaders() throws AuthFailureError {
        return headers != null ? headers : super.getHeaders();
    }

    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        return params;
    }

    @Override
    protected void deliverResponse(T response) {
        listener.onResponse(response);
    }

    @Override
    protected Response<T> parseNetworkResponse(NetworkResponse response) {
        try {
            String json = new String(response.data, HttpHeaderParser.parseCharset(response.headers));
            return Response.success(mGson.fromJson(json, clazz), HttpHeaderParser.parseCacheHeaders(response));
        } catch (UnsupportedEncodingException e) {
            return Response.error(new ParseError(e));
        } catch (JsonSyntaxException e) {
            return Response.error(new ParseError(e));
        }
    }
}
```

Εικόνα 54: Χρήση Gson

Κεφάλαιο 4

4.1 Ανάλυση απαιτήσεων για την δημιουργία της εφαρμογής

Το βασικότερο στοιχείο σε μία εφαρμογή είναι η ιδέα γύρω από την οποία χτίζεται. Η υλοποίηση της ιδέας όμως προϋποθέτει τη συνεργασία πολλών παραγόντων για να επιτευχθεί το επιθυμητό αποτέλεσμα. Κάποιοι από τους σπουδαιότερους παράγοντες είναι, ο προγραμματιστής που θα υλοποιήσει την ιδέα, το περιβάλλον στο οποίο θα την υλοποιήσει, καθώς και τα εργαλεία που θα χρησιμοποιήσει μέσα σε αυτό.

4.2 Ανάλυση της εφαρμογής με διαγράμματα UML

Ένα από τα βασικά και σημαντικά θέματα στην κατασκευή ενός πληροφοριακού συστήματος είναι η κατασκευή του ορθού συστήματος, δηλαδή του συστήματος που

ικανοποιεί τις απαιτήσεις των χρηστών και των άλλων συμμετεχόντων. Πολλά συστήματα πληροφορικής αποτυγχάνουν να ικανοποιήσουν αυτόν τον φαινομενικά στοιχειώδη στόχο: δεν ικανοποιούν τις προδιαγραφές του χρήστη. Εδώ έρχονται να δώσουν τη λύση οι γλώσσες προδιαγραφών λογισμικού, οι οποίες σε συνδυασμό με τις μεθοδολογίες ανάπτυξης αποτελούν τα δομικά συστατικά για την σωστή ανάπτυξη συστημάτων λογισμικού. Εφόσον λοιπόν μιλάμε για αντικειμενοστραφή προγραμματισμό έχουμε σχεδιάσει -αρχικά στο χαρτί κι έπειτα στον υπολογιστή- το μοντέλο και τα σενάρια λειτουργίας της εφαρμογής μας, προκειμένου να προχωρήσουμε αργότερα στην υλοποίηση.

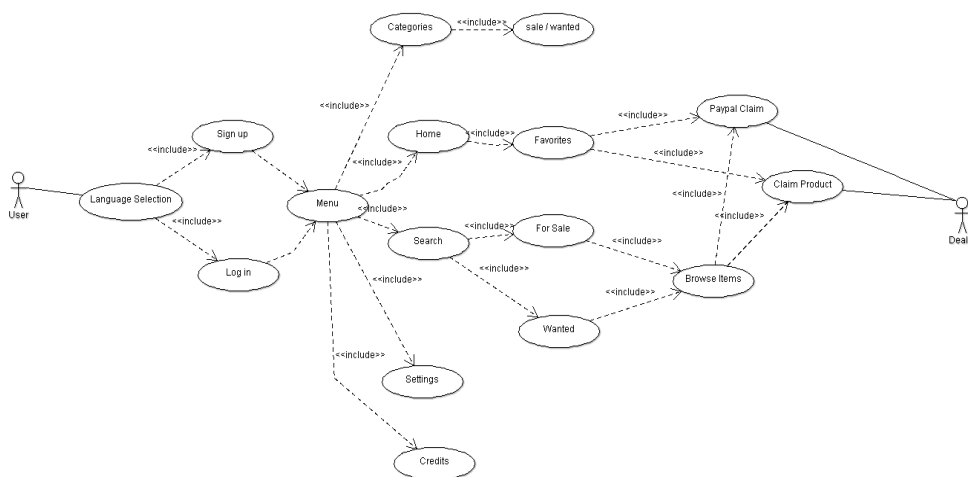
Για τη δημιουργία των διαγραμμάτων UML έγινε χρήση του λογισμικού ArgoUML [22] το οποίο είναι ένα πρόγραμμα ανοιχτού κώδικα εξ ολοκλήρου κατασκευασμένο σε Java.



Εικόνα 55: ArgoUML – Πρόγραμμα υλοποίησης UML

4.2.1 Διάγραμμα περιπτώσεων χρήσης

Το μοντέλο περιπτώσεων χρήσης (σύμφωνα με το [7]) δίνει έμφαση στη λειτουργικότητα ενός συστήματος όπως αυτή είναι ορατή από τους εξωτερικούς χρήστες του. Μία περίπτωση χρήσης διαμερίζει τη λειτουργικότητα ενός συστήματος σε συναλλαγές (περιπτώσεις χρήσης) που έχουν νόημα για τους χρήστες συστήματος. Για να έχουμε λοιπόν μία πρώτη εικόνα της διαδικασίας χρήσης της εφαρμογής μας δούμε το παρακάτω διάγραμμα.

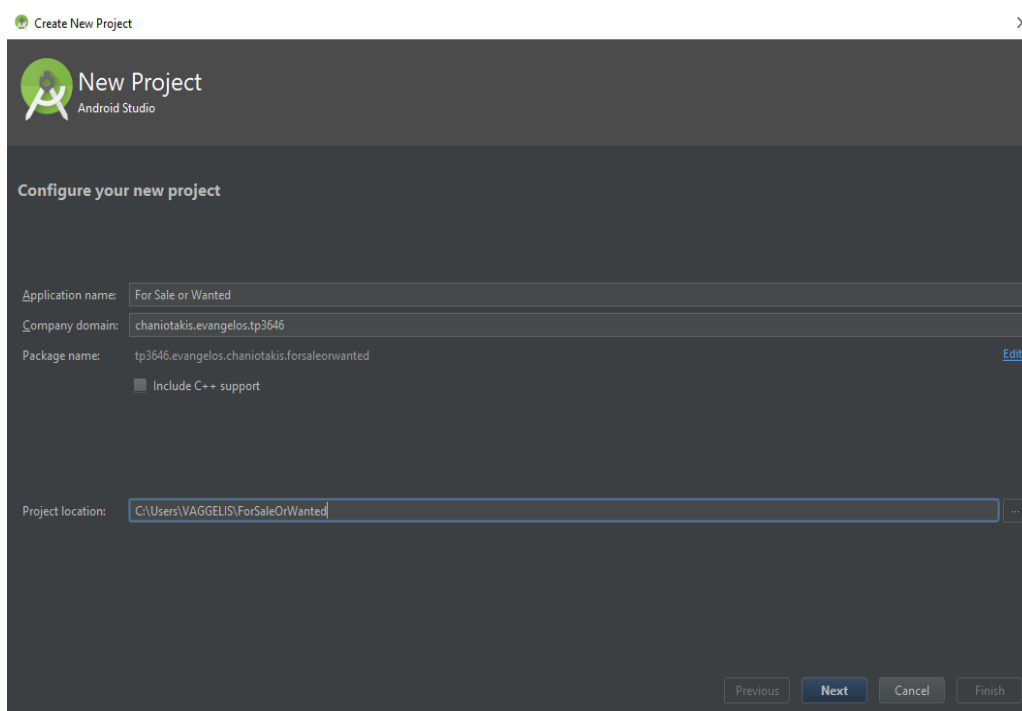


Εικόνα 56: Μοντέλο/Διάγραμμα Περιπτώσεων χρήσης σε UML

Παρατηρούμε τις περιπτώσεις χρήσης της εφαρμογής από το χρήστη (User) ο οποίος θέλει να χρησιμοποιήσει την εφαρμογή άμεσα και από τον χρήστη (Dealer) ο οποίος αλληλεπιδράει με την εφαρμογή στα πλαίσια επικοινωνίας και διαπραγμάτευσης με τον User.

4.3 Δημιουργία Project στο Android Studio

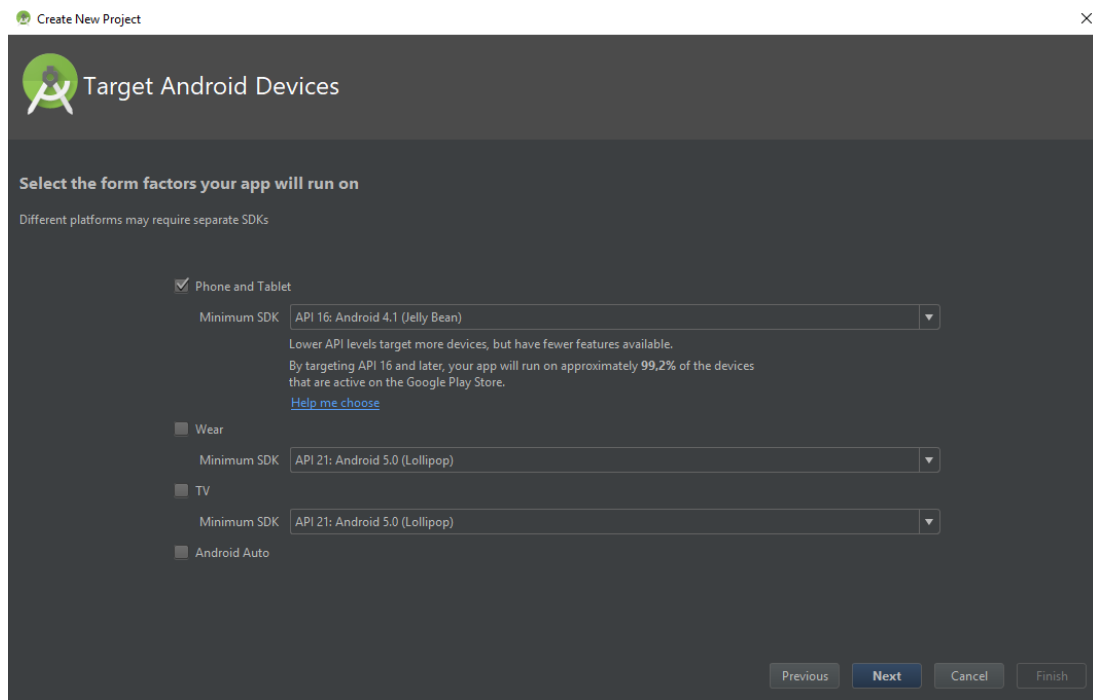
Εφόσον λοιπόν έχουμε ένα πλάνο για το πως θα δημιουργήσουμε την εφαρμογή, ξεκινάμε ένα νέο Project στο Android Studio.



Εικόνα 57: Δημιουργία project στο Android Studio

Δημιουργώντας ένα νέο Project ζητείται στον προγραμματιστή να δηλώσει το όνομα της εφαρμογής του, το όνομα της εταιρείας του καθώς και την τοποθεσία στην οποία θα δημιουργηθεί. Επιπροσθέτως κατά τη συμπλήρωση των πεδίων Application name και Company Domain δημιουργείται αυτόματα το όνομα πακέτου (Package name) της εφαρμογής το οποίο αποτελεί και την «ταυτότητα» της εφαρμογής. Η χρησιμότητα των πακέτων είναι να ομαδοποιούν τις κλάσεις της Java τόσο για την πιο εύκολη διαχείριση σφαλμάτων όσο και του μαζικού ορισμού των δικαιωμάτων πρόσβασης σε αυτές.

Στην επόμενη οθόνη επιλέγουμε την ελάχιστη έκδοση Android με την οποία θα είναι συμβατή η εφαρμογή την οποία θα αναπτύξουμε. Επομένως θα είναι συμβατή από την έκδοση 4.1 και μετά.



Εικόνα 58: Επιλογή Minimum SDK κατά τη δημιουργία του project

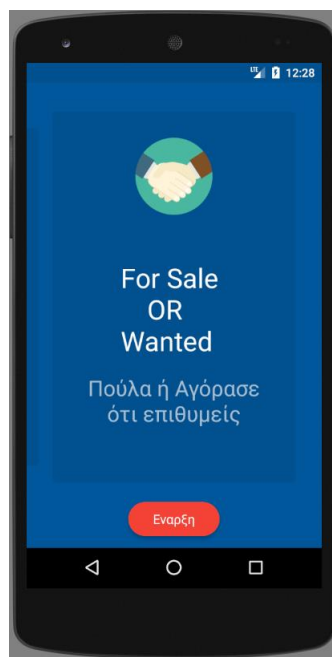
4.4 Δημιουργία και ανάλυση περιεχομένου εφαρμογής

Όπως θα παρατηρήσουμε και παρακάτω, η εφαρμογή υποστηρίζει αγγλικά και ελληνικά. Θα δούμε εικόνες και από τις δύο γλώσσες.

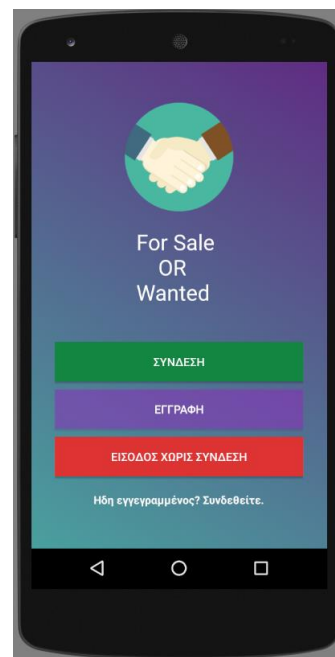
Ανοίγοντας την εφαρμογή βλέπουμε την οθόνη επιλογής γλώσσας. Αφού διαλέξουμε όποια θέλουμε, προχωράμε στην είσοδο. Μπορούμε να χρησιμοποιούμε την εφαρμογή είτε ως εγγεγραμμένοι χρήστες κάνοντας Login, είτε χωρίς.



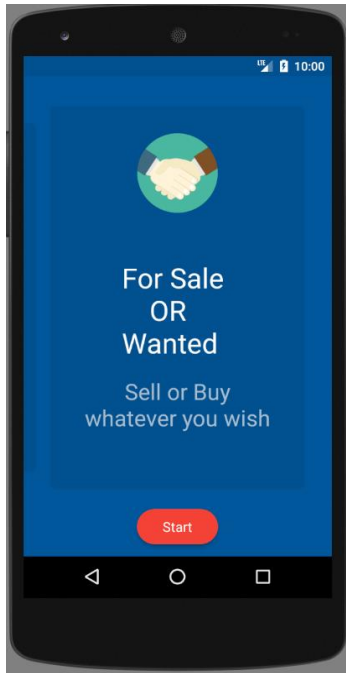
Εικόνα 59: Επιλογή γλώσσας



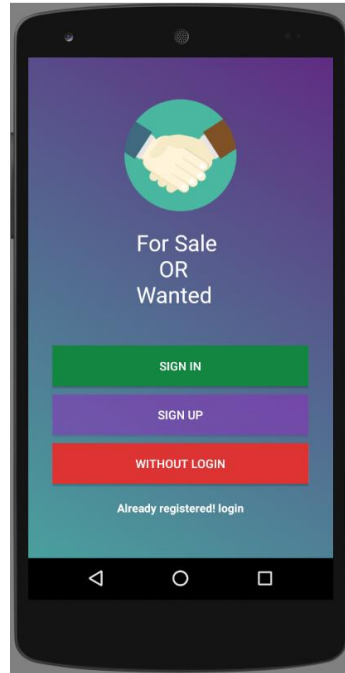
Εικόνα 60: Οθόνη έναρξης



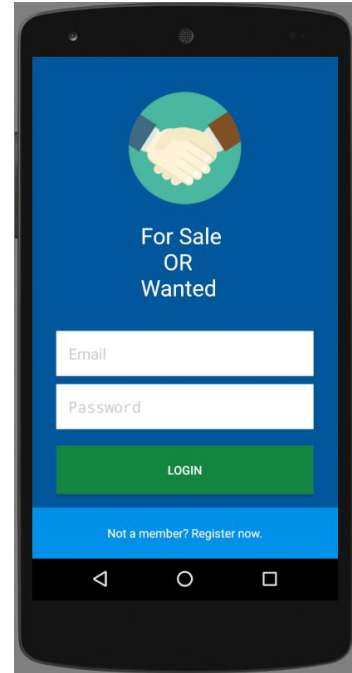
Εικόνα 61: Είσοδος/Εγγραφή



Εικόνα 62: Οθόνη έναρξης



Εικόνα 63: Είσοδος/Εγγραφή



Εικόνα 64: Είσοδος εγγεγραμμένου χρήστη

```

package tp3646.evangelos.chaniotakis.forsaleorwanted;

import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;

/**
 * Created by VAGGELIS on 19/10/2017.
 */

public class LanguageActivity extends AppCompatActivity implements View.OnClickListener{

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

    @Override
    public void onClick(View v) {
        switch (v.getId()) {

            case R.id.imageView3:
                Intent saleIntent = new Intent( packageContext: LanguageActivity.this, MainActivityGreek.class);
                startActivity(saleIntent);
                break;

            case R.id.imageView5:
                Intent wantedIntent = new Intent( packageContext: LanguageActivity.this, MainActivity.class);
                startActivity(wantedIntent);
                break;

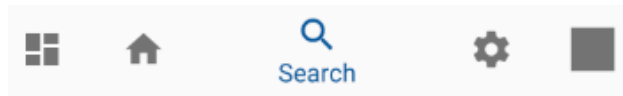
            default:
                break;
        }
    }
}

```

Εικόνα 65: Κομμάτι κώδικα υλοποίησης οθόνης έναρξης

Στη συνέχεια μεταβαίνουμε στο κύριο μενού της εφαρμογής όπου μπορούμε να επιλέξουμε από την μπάρα στο κάτω μέρος της οθόνης ανάμεσα σε πέντε καρτέλες,

αναλόγως τι θέλουμε να κάνουμε. Η μπάρα υλοποιήθηκε με τη χρήση των Fragments του Android.

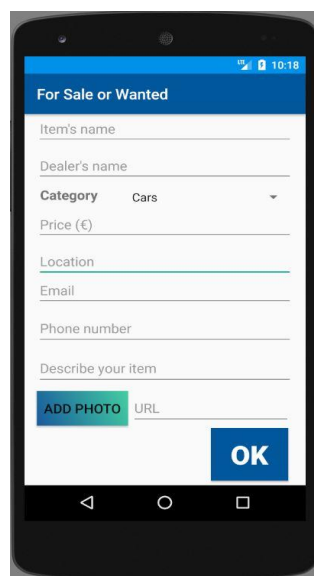


Εικόνα 66: Μπάρα Μενού

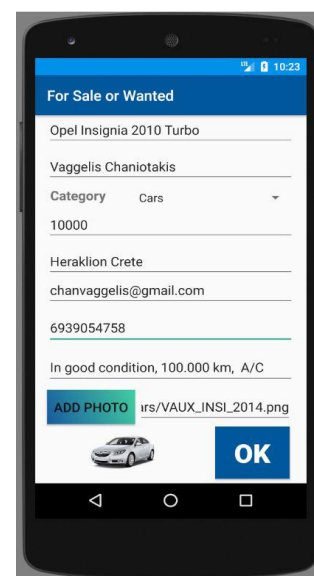
Η πρώτη επιλογή μας πάει σε αυτή την οθόνη όπου διαλέγουμε τι θέλουμε να δημοσιεύσουμε. Αναλόγως εάν θέλουμε να πουλήσουμε ή ψάχνουμε κάτι, πατάμε και το αντίστοιχο. Και στις δύο περιπτώσεις εμφανίζεται μια οθόνη όπου βάζουμε τις απαραίτητες πληροφορίες προκειμένου να προωθηθεί η δημοσίευση με επιτυχία.



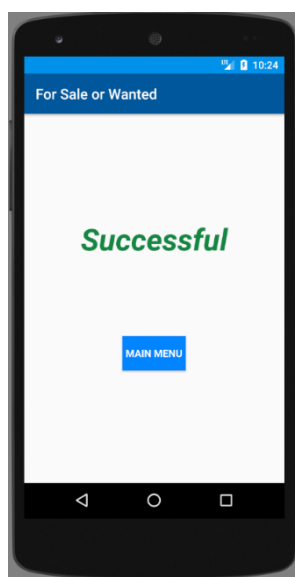
Εικόνα 67: Καρτέλα Κατηγοριών Καταχώρησης



Εικόνα 68: Επιλογή Κατηγορίας



Εικόνα 69: Συμπλήρωση Πληροφοριών

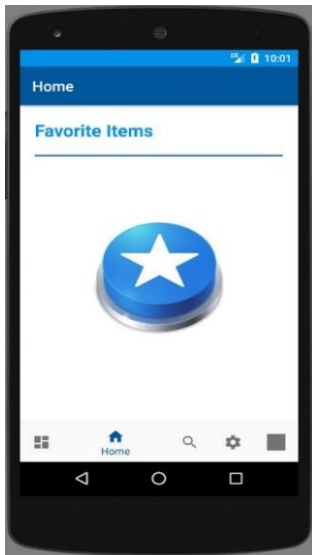


Εικόνα 70: Επιτυχής Καταχώρηση

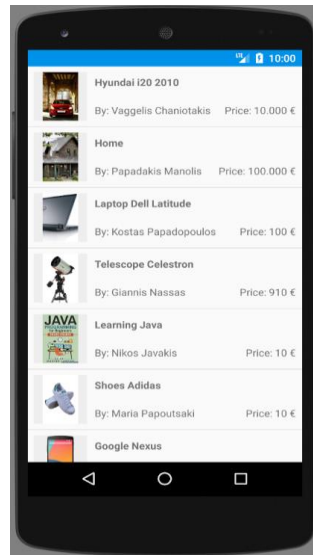
Η προσθήκη φωτογραφίας γίνεται με τον απλό και ευρέως διαδεδομένο τρόπο του copy paste. Δηλαδή εάν έχουμε μία φωτογραφία σε κάποιο από τα social media account μας ή σε κάποια ιστοσελίδα αγγελιών ή γενικότερα οπουδήποτε online, κάνουμε αντιγραφή του συνδέσμου και επικόλληση στο πεδίο URL. Η διαδικασία αυτή υλοποιήθηκε με χρήση της SQLite, όπου εξηγήσαμε παραπάνω τον τρόπο λειτουργίας της.

Αφού πατήσουμε το κουμπί 'Main Menu' επιστρέφουμε στο κύριο μενού προκειμένου να διαλέξουμε αν θέλουμε μια άλλη καρτέλα.

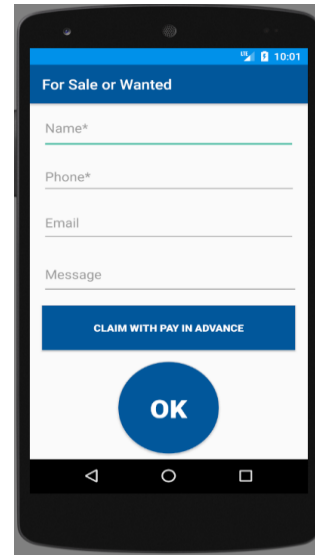
Η δεύτερη για παράδειγμα περιέχει τα αγαπημένα αντικείμενα των χρηστών στο σύνολο, προς πώληση. Έτσι μπορεί να καταλάβει κανείς ποια έχουν τη μεγαλύτερη δημοτικότητα. Μπαίνοντας εκεί μπορούμε να επιλέξουμε ένα από αυτά για να το διεκδικήσουμε από τον διαπραγματευτή είτε με προκαταβολή (Claim with pay in advance) είτε χωρίς. Η διεκδίκηση με προκαταβολή όπως εξηγήσαμε και παραπάνω γίνεται με τη χρήση της υπηρεσίας PayPal.



Εικόνα 71: Καρτέλα Αρχικής Σελίδας

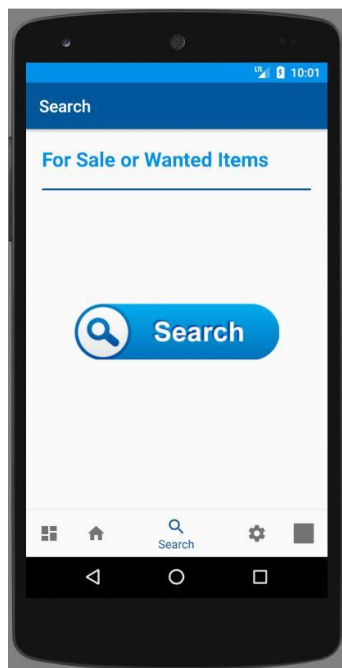


Εικόνα 72: Περιεχόμενο Αρχικής Σελίδας



Εικόνα 73: Διεκδίκηση προϊόντος

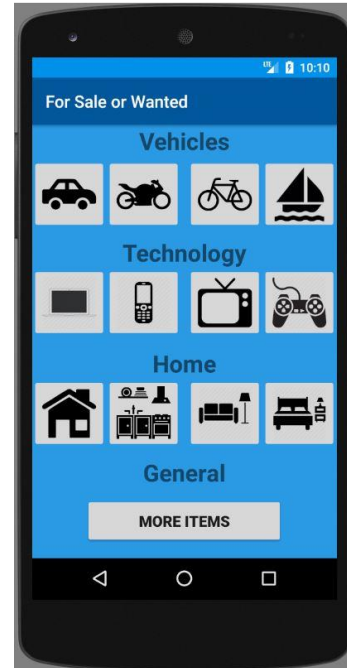
Η Τρίτη καρτέλα περιέχει την αναζήτηση αντικειμένων στην εφαρμογή. Μπαίνοντας, διαλέγουμε ποια κατηγορία προϊόντων-υπηρεσιών μας ενδιαφέρει περισσότερο να ψάξουμε και συνεχίζουμε στην περιήγηση. Από εκεί έχουμε τη δυνατότητα να επιλέξουμε κάποιο αντικείμενο το οποίο είναι προς πώληση και να το διεκδικήσουμε με τους τρόπους που εξηγήσαμε και παραπάνω.



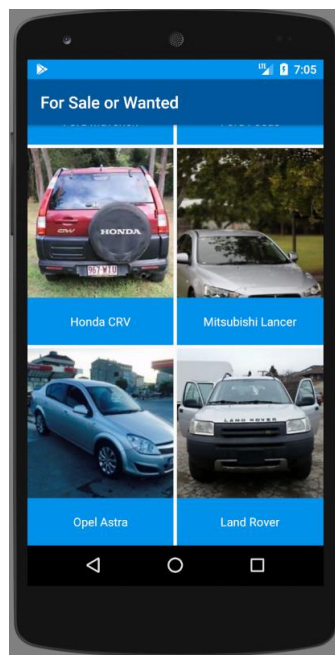
Εικόνα 74: Επιλογή καρτέλας αναζήτησης(Αγγλ.)



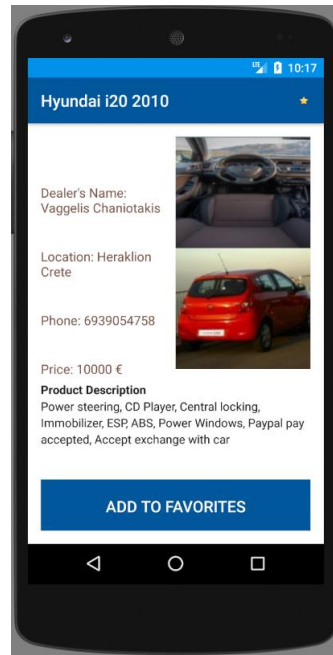
Εικόνα 75: Κατηγορίες αναζήτησης



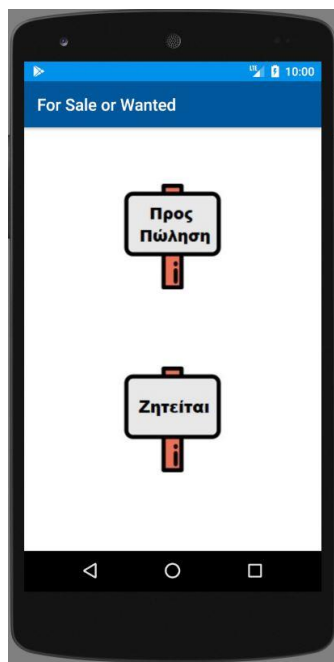
Εικόνα 76: Υποκατηγορίες αναζήτησης



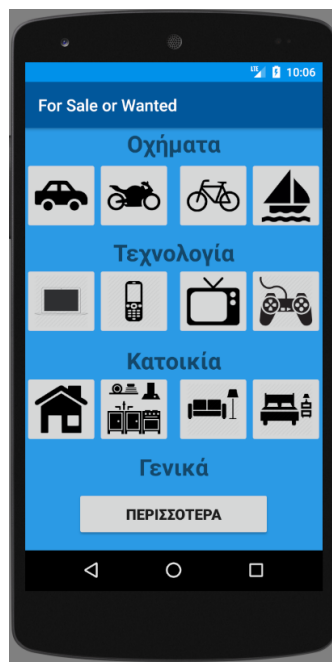
Εικόνα 77: Περιήγηση στα Αυτοκίνητα



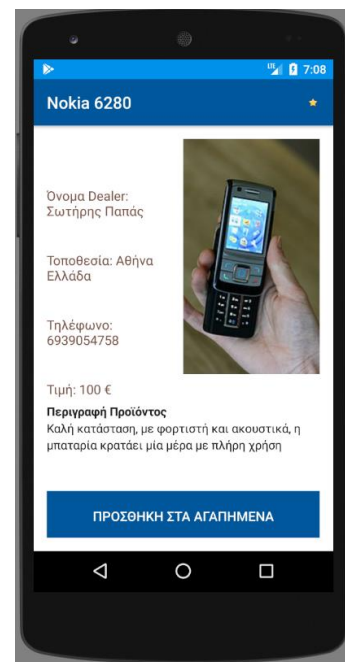
Εικόνα 78: Επιλογή συγκεκριμένου αυτοκινήτου



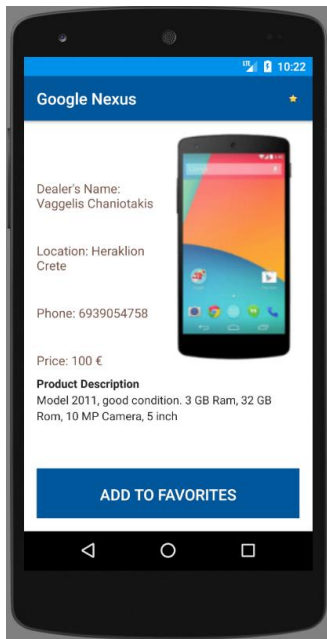
Εικόνα 79: Κατηγορίες Αναζήτησης (Ελλ.)



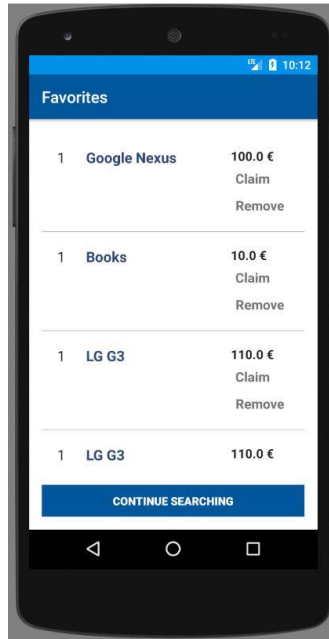
Εικόνα 80: Υποκατηγορίες Αναζήτησης



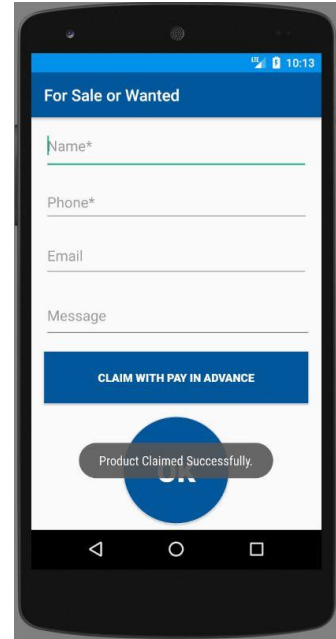
Εικόνα 81: Επιλογή συγκεκριμένου προϊόντος



Εικόνα 82: Επιλογή κινητού

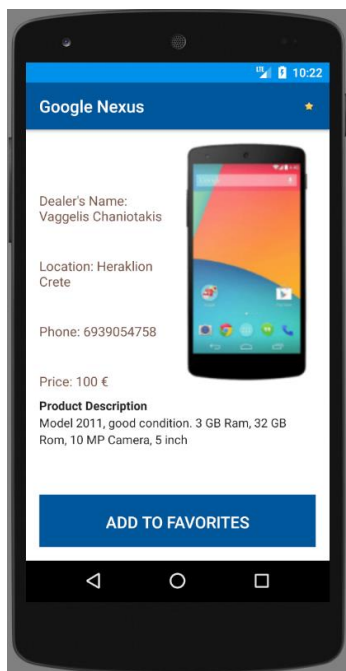


Εικόνα 83: Προσθήκη στα αγαπημένα

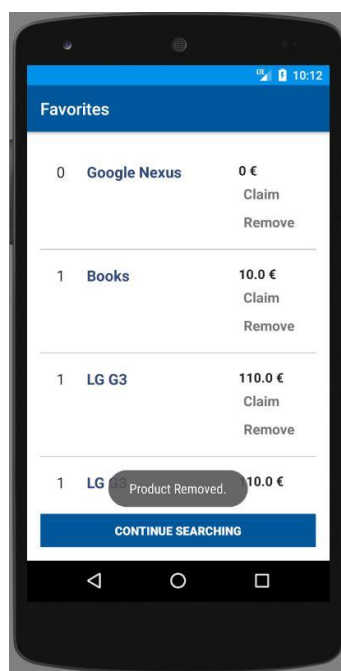


Εικόνα 84: Διεκδίκηση προϊόντος

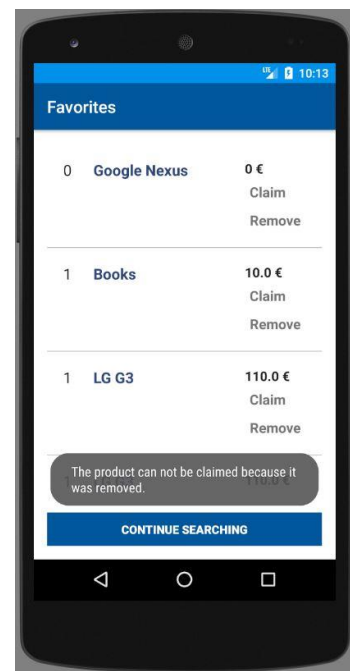
Σε περίπτωση που βάλουμε κάποιο προϊόν στα «Αγαπημένα» χωρίς να το θέλουμε τελικά, το αφαιρούμε ανά πάσα στιγμή. Αυτό έγινε με τη χρήση των intent services, όπου εξηγήσαμε παραπάνω τη λειτουργία τους. Οπότε όταν το αφαιρέσουμε και μετά πατήσουμε το κουμπί «Claim» θα εμφανιστεί το αντίστοιχο μήνυμα για να μας πληροφορήσει ότι πλέον δεν γίνεται να διεκδικηθεί. Το μήνυμα αυτό υλοποιήθηκε με τη βοήθεια των παραθύρων ειδοποιήσεων (Alert Dialogs) και σύντομων μηνυμάτων (Toasts), τα οποία αναλύσαμε παραπάνω.



Εικόνα 85: Επιλογή κινητού



Εικόνα 86: Αφαίρεση από τα αγαπημένα



Εικόνα 87: Άρνηση διεκδίκησης λόγω αφαίρεσης

Παρακάτω βλέπουμε ένα μέρος του κώδικα που υλοποιεί αυτές τις λειτουργίες.

```
@Override
public void onBindViewHolder(final CheckRecyclerViewHolder holder, final int position) {
    //get product quantity
    holder.quantity.setText("1");
    holder.productName.setText(mProductObject.get(position).getProductName());
    holder.productPrice.setText(String.valueOf(mProductObject.get(position).getProductPrice()) + " €");

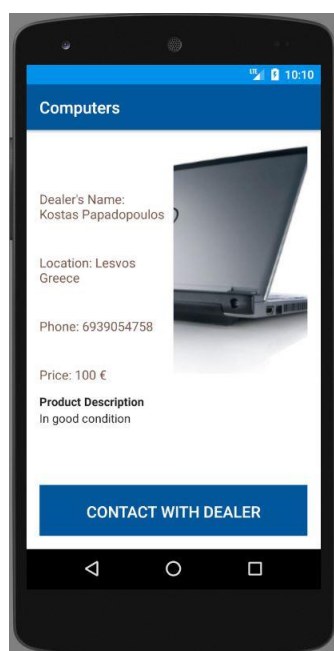
    holder.removeProduct.setOnClickListener((view) -> {
        Toast.makeText(context, "Product Removed.", Toast.LENGTH_SHORT).show();
        holder.quantity.setText("0");
        holder.productName.setText(mProductObject.get(position).getProductName());
        holder.productPrice.setText("0 €");

        holder.claimProduct.setOnClickListener((view) -> {
            Toast.makeText(context, "The product can not be claimed because it was removed.", Toast.LENGTH_SHORT).show();
        });
    });

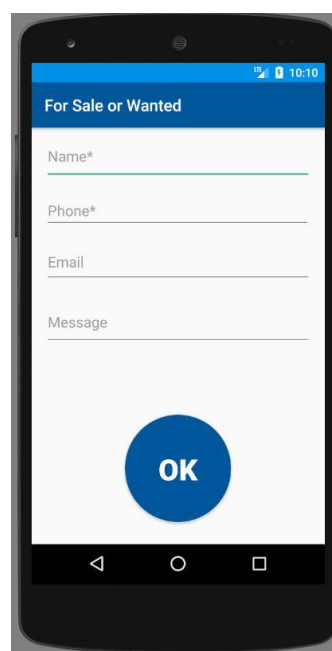
    holder.claimProduct.setOnClickListener((view) -> {
        Toast.makeText(context, "Product Claimed Successfully.", Toast.LENGTH_SHORT).show();
        Intent profileIntent = new Intent(context, ClaimActivity.class);
        context.startActivity(profileIntent);
    });
}
```

Εικόνα 88: Κώδικας που υλοποιεί της παραπάνω λειτουργίες

Στην περίπτωση που βρούμε μια αγγελία που αναζητεί ένα προϊόν ή μία υπηρεσία την οποία έχουμε να δώσουμε, τότε ερχόμαστε σε επικοινωνία με τον διαπραγματευτή μέσω μιας απλής φόρμας επικοινωνίας. Στη φόρμα γράφουμε υποχρεωτικά το όνομα, το τηλέφωνο, προαιρετικά το email και το μήνυμα που θέλουμε να μεταφέρουμε στον διαπραγματευτή και στη συνέχεια περιμένουμε να επικοινωνήσει αυτός μαζί μας. Ένα παράδειγμα βλέπουμε παρακάτω, όπου ο κύριος Κώστας Παπαδόπουλος αναζητεί ένα φορητό υπολογιστή σε καλή κατάσταση και διαθέτει μέχρι 100 ευρώ. Εμείς εάν έχουμε να του πουλήσουμε αυτό το προϊόν που αναζητεί, τότε συμπληρώνουμε τη φόρμα και πατάμε OK.

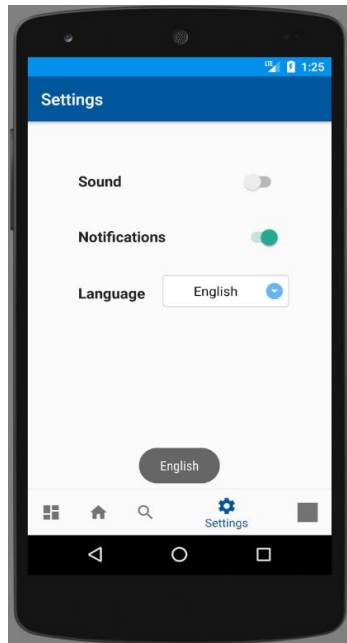


Εικόνα 89: Ζήτηση προϊόντος

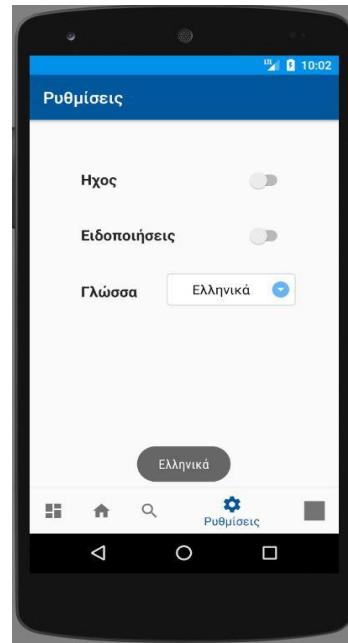


Εικόνα 90: Φόρμα επικοινωνίας

Η τέταρτη καρτέλα περιέχει τις βασικές ρυθμίσεις που μπορεί να έχει μια εφαρμογή, όπως είναι οι ειδοποιήσεις, είτε με ήχο, είτε χωρίς, καθώς και η γλώσσα. Στη συγκεκριμένη έκδοση της εφαρμογής δεν είναι ενεργοποιημένη η λειτουργία των ειδοποιήσεων και του ήχου, όμως σε μεταγενέστερη έκδοση (πιθανόν σε κάποιο update) να προστεθούν ως έξτρα features.



Εικόνα 91: Settings



Εικόνα 92: Ρυθμίσεις

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/textViews"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:layout_marginLeft="50dp"
        android:layout_marginRight="50dp"
        android:layout_marginTop="50dp"
        android:gravity="center"
        android:orientation="horizontal">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentLeft="true"
            android:layout_gravity="center_vertical"
            android:layout_margin="10dp"
            android:text="@string/sounds"
            android:textColor="@color/colorPrimaryText"
            android:textSize="18sp"
            android:textStyle="bold" />

        <Switch
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentRight="true"
            android:layout_gravity="center_vertical"
            android:layout_margin="10dp" />

    </RelativeLayout>

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:layout_marginLeft="50dp"
        android:layout_marginRight="50dp"
        android:layout_marginTop="20dp"
        android:orientation="horizontal">

        <TextView
            android:id="@+id/textViews8"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentLeft="true"
            android:layout_gravity="center_vertical"
            android:layout_margin="10dp"
            android:text="@string/notifications"
            android:textColor="@color/colorPrimaryText"
            android:textSize="18sp"
            android:textStyle="bold" />

        <Switch
            android:id="@+id/notificationsSwitch"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentRight="true"
            android:layout_gravity="center_vertical"
            android:layout_margin="10dp"
            android:checked="false" />

        <TextView
            android:id="@+id/textViews9"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignLeft="@+id/textViews8"
            android:layout_alignParentBottom="true"
            android:layout_alignStart="@+id/textViews" />

    </RelativeLayout>

</LinearLayout>
```

Εικόνα 93: Κώδικας Settings

```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_marginLeft="50dp"
    android:layout_marginRight="50dp"
    android:layout_marginTop="20dp"
    android:orientation="horizontal">

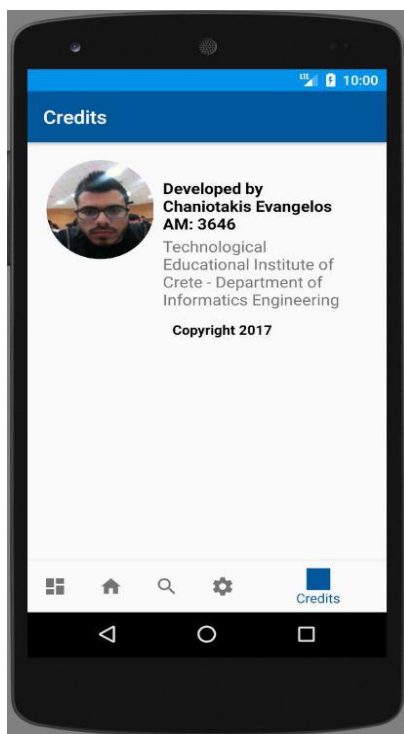
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_gravity="center_vertical"
        android:layout_margin="10dp"
        android:text="@string/language"
        android:textColor="@color/colorPrimaryText"
        android:textSize="18sp"
        android:textStyle="bold" />

    <Spinner
        android:id="@+id/spinner"
        android:layout_width="150dp"
        android:layout_height="40dp"
        android:layout_alignParentRight="true"
        android:background="@drawable/custom_spinner_background"
        android:spinnerMode="dropdown" />

</RelativeLayout>
```

Εικόνα 94: Κώδικας Settings

Η πέμπτη καρτέλα περιέχει κάποιες βασικές πληροφορίες για τον δημιουργό και προγραμματιστή της εφαρμογής.



Εικόνα 95: Καρτέλα Credits

Κεφάλαιο 5

5.1 Συμπεράσματα

Η έμπνευση και ο σχεδιασμός της εφαρμογής στο χαρτί ήταν ίσως το εύκολο κομμάτι της συνολικής προσπάθειας που κατέβαλα προκειμένου να φτάσω στο τελικό αποτέλεσμα. Η υλοποίηση και ανάπτυξη όμως ήταν αρκετά χρονοβόρα και απαιτητική διαδικασία, ειδικά για κάποιον που δεν έχει την απαραίτητη εμπειρία και δεν έχει διδαχτεί κάτι παρόμοιο στο παρελθόν. Χρειάστηκαν πολλές ώρες μελέτης, αναζήτησης και δοκιμών προκειμένου να βρεθούν και να δοκιμαστούν τα κατάλληλα εργαλεία και “υλικά” για την υλοποίηση της εφαρμογής. Επίσης πολλές ώρες δαπανήθηκαν στο να σκεφτώ τρόπους ώστε το περιβάλλον που θα αντικρίζει ο χρήστης να είναι φιλικό και πάνω από όλα εύκολο στη χρήση. Παρ όλες τις δυσκολίες όμως, το τελικό αποτέλεσμα πιστεύω ότι είναι καλό και άξιζε τον κόπο.

5.2 Μελλοντική Εξέλιξη

Αρχικός στόχος της εφαρμογής είναι η εξυπηρέτηση του απλού ανθρώπου-χρήστη, ο οποίος θέλει να πουλήσει ή να αγοράσει κάποιο προϊόν ή κάποια υπηρεσία. Η εξυπηρέτηση αυτή μπορεί να βελτιωθεί ακόμη περισσότερο μελλοντικά, με πολλούς τρόπους. Ένας από αυτούς είναι η προσθήκη ειδοποιήσεων push προκειμένου να γνωστοποιείται στο χρήστη κάποιο συμβάν σχετικά με τη δημοσίευση ενός νέου προϊόντος/υπηρεσίας προς πώληση, ζήτηση ή ακόμα και η αλλαγή τιμής κάποιου ήδη υπάρχοντος.

Επίσης μία καλή λειτουργία η οποία θα μπορούσε να προστεθεί σε κάποια μεταγενέστερη έκδοση της εφαρμογής, είναι η κλασική μπάρα αναζήτησης, όπου ο χρήστης θα μπορούσε να γράφει ένα προϊόν/υπηρεσία που αναζητεί και να του εμφανίζονται τα αντίστοιχα αποτελέσματα.

Τέλος, η δημοσίευση της εφαρμογής στην αγορά της Google (Play Store) είναι ένας από τους επιθυμητούς στόχους, προκειμένου να αξιοποιηθεί από το ευρύ κοινό και να διευκολύνει την καθημερινή διαχείριση αγοραπωλησιών μεταξύ των ανθρώπων. Προφανώς μία τέτοια ενέργεια απαιτεί αρκετά χρήματα, διότι όσο περισσότερες πληροφορίες και φωτογραφίες δημοσιεύονται, τόσο μεγαλύτερες θα είναι και οι ανάγκες για την αποθήκευση αυτών των δεδομένων.

Κεφάλαιο 6

6.1 Βιβλιογραφία

1. Εισαγωγή στον Προγραμματισμό Android , I.N.ΕΛΛΗΝΑΣ - N.I.ΕΛΛΗΝΑΣ, Εκδόσεις Τζιόλα, 2014
2. Android ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ, Paul Deitel - Harvey Deitel - Abbey Deitel, Εκδόσεις: Μ. Γκιούρδας, 2014
3. Ανάπτυξη Εφαρμογών με το Android, Lauren Darcey – Shane Conder, Εκδόσεις: Μ. Γκιούρδας, 2011
4. Android για Προγραμματιστές Μάθετε μέσα από Εφαρμογές, Paul Deitel - Harvey Deitel - Abbey Deitel – Michael Morgano, Εκδόσεις: Μ. Γκιούρδας, 2012
5. Burnette, Ed. Hello, Android: introducing Google’s mobile development platform. Pragmatic Bookshelf, 2009
6. Πλήρες Εγχειρίδιο της Java 7, Rogers Cadenhead, Εκδόσεις: Μ. Γκιούρδας, 2013

7. Βασικές Αρχές Τεχνολογίας Λογισμικού 8, Sommerville, Εκδόσεις ΚΛΕΙΔΑΡΙΘΜΟΣ, 2009
8. Android Programming: The Big Nerd Ranch Guide (3rd Edition) (Big Nerd Ranch Guides)
9. Οδηγός της XML με παραδείγματα, Benoit Marchal, Β.Γκιούρδας Εκδοτική, 2001
10. <https://developer.android.com/training/index.html>
11. <https://www.tutorialspoint.com/android/>
12. Android Application Development All-in-one for Dummies, Barry Burd, 2015
13. <https://www.skillshare.com/browse/The-Complete-Android-Developer-Course-Go-From-Beginner-To-Advanced>
14. <http://www.greeceandroid.gr/development>
15. Developing Android Apps by Google, <https://www.udacity.com/course>
16. Android Studio Download Page
<https://developer.android.com/studio/index.html>
17. <https://developer.android.com/training/volley/simple.html>
18. <https://androidresearch.wordpress.com>
19. Stack Overflow, <https://stackoverflow.com/>
20. <https://el.wikipedia.org>
21. <http://www.androidauthority.com/>
22. <http://argouml.tigris.org/>
23. <https://www.theatlas.com/charts/N1e4Qloq>
24. <https://github.com/google/gson>
25. <https://www.statista.com>
26. <https://developer.android.com/design/index.html>
27. https://en.wikipedia.org/wiki/Open_Handset_Alliance
28. <https://el.wikipedia.org/wiki/Google>
29. <https://www.gartner.com/technology/home.jsp>