

**Πανεπιστήμιο Πειραιώς**

**Τμήμα Ψηφιακών Συστημάτων**

**Π.Μ.Σ. " Ψηφιακά Συστήματα & Υπηρεσίες "**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**ΘΕΜΑ: «ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ ANDROID ΓΙΑ ΤΗΝ  
ΛΗΨΗ ΜΕΤΡΗΣΕΩΝ ΠΟΙΟΤΗΤΑΣ ΔΙΚΤΥΟΥ ΣΕ ΑΣΥΡΜΑΤΑ  
ΔΙΚΤΥΑ»**



**ΟΝΟΜΑΤΕΠΩΝΥΜΟ : ΜΑΡΙΟΣ ΝΤΙΝΟΣ**

**ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΑΘΑΝΑΣΙΟΣ ΚΑΝΑΤΑΣ**

**ΠΕΙΡΑΙΑΣ 2016**

**Πανεπιστήμιο Πειραιώς**

**Τμήμα Ψηφιακών Συστημάτων**

**Π.Μ.Σ. " Ψηφιακά Συστήματα & Υπηρεσίες "**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**ΘΕΜΑ: «ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ ANDROID ΓΙΑ ΤΗΝ  
ΛΗΨΗ ΜΕΤΡΗΣΕΩΝ ΠΟΙΟΤΗΤΑΣ ΔΙΚΤΥΟΥ ΣΕ ΑΣΥΡΜΑΤΑ  
ΔΙΚΤΥΑ»**



**ΟΝΟΜΑΤΕΠΩΝΥΜΟ : ΜΑΡΙΟΣ ΝΤΙΝΟΣ**

**ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΑΘΑΝΑΣΙΟΣ ΚΑΝΑΤΑΣ**

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την .....

(Υπογραφή)

.....

(Υπογραφή)

.....

(Υπογραφή)

.....

**ΠΕΙΡΑΙΑΣ 2016**

## Ευχαριστίες

Πρώτα απ' όλα, θέλω να ευχαριστήσω τον επιβλέποντα της διπλωματικής εργασίας μου, Καθηγητή κ. Αθανάσιο Κανάτα, για την πολύτιμη βοήθεια και καθοδήγησή του κατά τη διάρκεια της δουλειάς μου. Η κατανόηση στα προβλήματα που παρουσιάστηκαν κατά την εκπόνηση της εργασίας ήταν υψίστης σημασίας.

Πάνω απ' όλα, είμαι ευγνώμων στους γονείς μου, Χρήστο και Ροδοθέα Ντίνου για την ολόψυχη αγάπη και υποστήριξή τους όλα αυτά τα χρόνια. Αφιερώνω αυτή την εργασία στην μητέρα μου και στον πατέρα μου.

Μάριος Ντίνος

## Περίληψη

Παρακάτω περιγράφονται εφαρμογές οι οποίες είναι παρόμοιες με την εφαρμογή η οποία θα αναπτυχθεί στην πτυχιακή μου εργασία. Επιπλέον, παρουσιάζονται στατιστικά στοιχεία σχετικά με τον αριθμό των κινητών συσκευών οι οποίες λειτουργούν με android και της απήχησης των εφαρμογών σε android στους χρήστες των κινητών συσκευών η οποία είναι μετρήσιμη βάσει τον αριθμό των μεταφορτώσεων.

Στο επόμενο κεφάλαιο θα περιγραφεί η πλατφόρμα Android και τα βασικά στοιχεία μίας εφαρμογής Android. Πιο συγκεκριμένα, θα περιγραφούν όλες οι εκδόσεις των λειτουργικών Android (τα κύρια χαρακτηριστικά τους), η αρχιτεκτονική του συστήματος και η δομή μίας εφαρμογής (φάκελοι source/layout/configuration). Στο επόμενο κεφάλαιο θα περιγραφεί το περιβάλλον σχεδίασης Android Studio και τα σημαντικότερα εργαλεία που διαθέτει για την ανάπτυξη της εφαρμογής.

Στο τέταρτο κεφάλαιο θα περιγράφει η εφαρμογή και τα κυριότερα κομμάτια κώδικα της εφαρμογής. Στη συνέχεια θα παρουσιαστούν οι οθόνες της εφαρμογής.

# Περιεχόμενα

Ευχαριστίες .....	2
Περίληψη .....	3
Περιεχόμενα .....	4
Κεφάλαιο 1 .....	7
1.1. Περίληψη .....	7
1.2. Βιβλιογραφική Ανασκόπηση.....	7
1.2.1 Εφαρμογές Android.....	7
1.2.2. Εφαρμογές σχετικές με μετρήσεις QOS.....	10
Κεφάλαιο 2 Θεωρητικό.....	11
2.1. Εισαγωγή.....	11
2.2. Εκδόσεις και χαρακτηριστικά .....	13
Συνολικός Πίνακας Εκδόσεων .....	19
2.3. Αρχιτεκτονική του Android .....	20
2.3.1 Εργαλεία προγραμματισμού Android.....	21
2.3.2 Συστατικά μέρη μίας Android εφαρμογής.....	23
Δραστηριότητες (Activities).....	24
Υπηρεσίες (Services) .....	25
Πάροχοι Περιεχομένου (Content Providers).....	26
Broadcast Receivers.....	27
Διεπαφή Χρήστη (User Interface).....	27
Διάταξη των γραφικών στοιχείων (Layout).....	29
Έλεγχοι Εισόδου (Input Controls) .....	30
Menu .....	30
Διάλογοι (Dialogs).....	31
Ειδοποιήσεις (Notifications).....	32
Στοιχεία Εφαρμογής (Application Resources).....	33
Προσανατολισμός (Mode).....	33
Μέγεθος οθόνης (Screen Size) .....	34
2.4. Android Manifest.....	35
Σύνοψη .....	37
Κεφάλαιο 3ο Πρακτικό .....	38

3.1. Android Studio.....	38
3.1.1 Χαρακτηριστικά του Android Studio IDE.....	39
3.1.2 Άλλα χαρακτηριστικά .....	42
3.2 Εργαλεία ανάπτυξης λογισμικού Android (Android Software Development Kit SDK) ...	42
3.2.2 Εργαλεία ανάπτυξης (Development Tools).....	45
3.2.3 Εργαλεία αποσφαλμάτωσης (Debugging Tools).....	47
3.2.4 Εργαλεία Κατασκευής (Build Tools) .....	48
3.2.5 Εργαλεία Πλατφόρμας (Platform Tools) .....	51
3.3 Εφαρμογή.....	52
3.3.1 Απαιτήσεις εφαρμογής .....	52
3.3.2 Κώδικας εφαρμογής.....	52
3.3.3 Οθόνη .....	54
Κεφάλαιο 4 Συμπεράσματα Μελλοντικές Βελτιώσεις .....	58
Βιβλιογραφία .....	60
Παράρτημα .....	62

Εικόνα 1 - Google play number of apps.....	8
Εικόνα 2 - Android OS market share.....	9
Εικόνα 3 - Εταιρίες λογισμικού και κατασκευής υλικού παγκόσμιας εμβέλειας.....	12
Εικόνα 4 - Λογότυπο πλατφόρμας android .....	12
Εικόνα 5 -Λογότυπο Android 1.5 CupCake .....	13
Εικόνα 6 - Λογότυπο android 1.6 DONUT.....	14
Εικόνα 7 – Λογότυπο Android 2.0 ECLAIR .....	14
Εικόνα 8 - Λογότυπο Android 2.2 Froyo .....	15
Εικόνα 9 - Λογότυπο Android 2.3 GingerBread .....	16
Εικόνα 10 - Λογότυπο Android 3.0 Honeycomb.....	17
Εικόνα 11 - Λογότυπο Android Ice Cream Sandwich.....	17
Εικόνα 12 - Λογότυπο Android Jelly Bean .....	18
Εικόνα 13 - Λογότυπο Android Kitkat.....	19
Εικόνα 14 - Λογότυπο Android Lollipop .....	19
Εικόνα 15 - Αναλογία Συσκευών Λειτουργικών .....	19
Εικόνα 16 - Η Αρχιτεκτονική του Android.....	21
Εικόνα 17 - Το περιβάλλον του Eclipse.....	22
Εικόνα 18 - Η εικονική συσκευή του Eclipse .....	23
Εικόνα 19 - Ο κύκλος ζωής των Activities .....	25
Εικόνα 20 - Η ιεραρχική δομή της Διεπαφής Χρήστη .....	28
Εικόνα 21 - Linear Layout.....	29
Εικόνα 22 - Android Manifest .....	36
Εικόνα 23 - Λογότυπο του Android Studio IDE.....	38
Εικόνα 24 -Android Studio IDE.....	39
Εικόνα 25 - Παράδειγμα έξυπνης αυτοσυμπλήρωσης κώδικα.....	40
Εικόνα 26 - Παράδειγμα ανάλυσης κώδικα κατά την επεξεργασία .....	41
Εικόνα 27 - Διαχειριστής εικονικών συσκευών .....	43
Εικόνα 28 - Εξομοιωτής Android συσκευών.....	44
Εικόνα 29 - Παράδειγμα Θέασης ιεραρχίας.....	45
Εικόνα 30 - Παράδειγμα ελέγχου σφαλμάτων με τη χρήση του εργαλείου Lint .....	46
Εικόνα 31 - Παράδειγμα ανάλυσης μνήμης με το εργαλείο παρακολούθησης συσκευής.....	47
Εικόνα 32 - Παράδειγμα διαγράμματος της στοίβας κλήσεων .....	48
Εικόνα 33 - Παράδειγμα χρήσης του Draw 9 .....	49
Εικόνα 34 - Παράδειγμα χρήσης του Vector Asset Studio .....	50
Εικόνα 35 - Παράδειγμα χρήσης του logcat.....	51
Εικόνα 36 - Οθόνη της εφαρμογής από το Android Studio(1/2).....	55
Εικόνα 36 - Οθόνη της εφαρμογής από το Android Studio(2/2).....	55
Εικόνα 36 - Οθόνη της εφαρμογής από το κινητό τηλέφωνο android (1/2) .....	56
Εικόνα 36 - Οθόνη της εφαρμογής από το κινητό τηλέφωνο android (2/3) .....	57
Εικόνα 40 - Οθόνη της εφαρμογής από το κινητό τηλέφωνο android (3/3) .....	58

# Κεφάλαιο 1

## 1.1. Περίληψη

Παρακάτω περιγράφονται εφαρμογές οι οποίες είναι παρόμοιες με την εφαρμογή η οποία θα αναπτυχθεί στην πτυχιακή μου εργασία. Επιπλέον, παρουσιάζονται στατιστικά στοιχεία σχετικά με τον αριθμό των κινητών συσκευών οι οποίες λειτουργούν με android και της απήχησης των εφαρμογών σε android στους χρήστες των κινητών συσκευών η οποία είναι μετρήσιμη βάσει τον αριθμό των μεταφορτώσεων.

Στο επόμενο κεφάλαιο θα περιγραφεί η πλατφόρμα Android και τα βασικά στοιχεία μίας εφαρμογής Android. Πιο συγκεκριμένα, θα περιγραφούν όλες οι εκδόσεις των λειτουργικών Android (τα κύρια χαρακτηριστικά τους), η αρχιτεκτονική του συστήματος και η δομή μίας εφαρμογής (φάκελοι source/layout/configuration). Στο επόμενο κεφάλαιο θα περιγραφεί το περιβάλλον σχεδίασης Android Studio και τα σημαντικότερα εργαλεία που διαθέτει για την ανάπτυξη της εφαρμογής.

Στο τέταρτο κεφάλαιο θα περιγράψει η εφαρμογή και τα κυριότερα κομμάτια κώδικα της εφαρμογής. Στη συνέχεια θα παρουσιαστούν οι οθόνες της εφαρμογής.

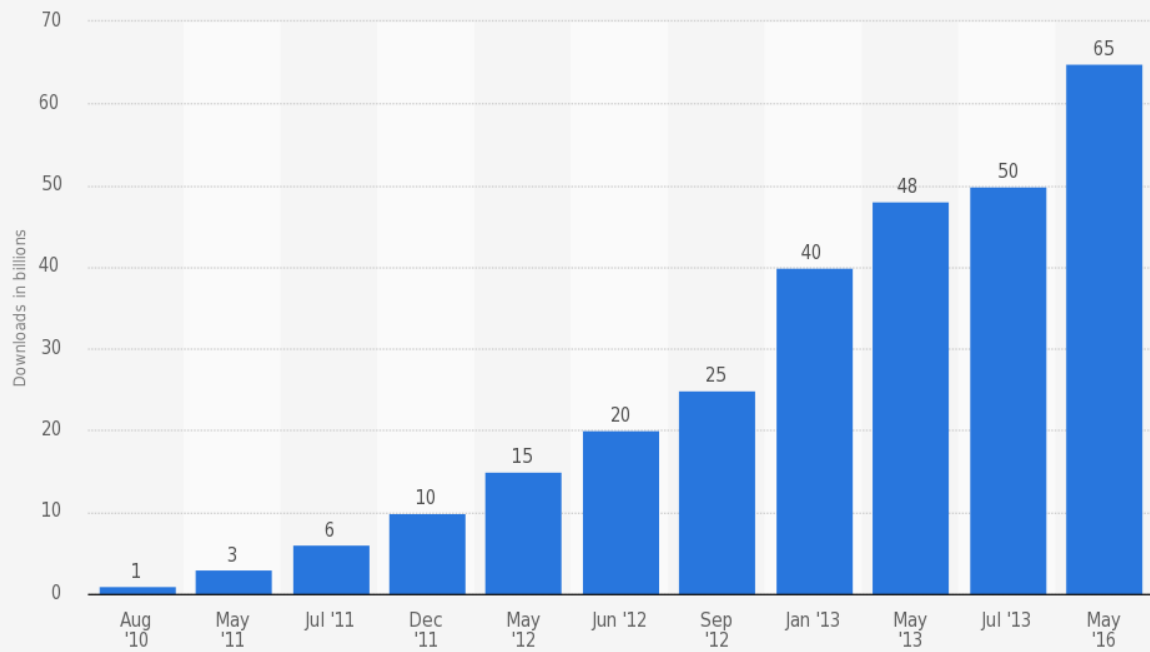
## 1.2. Βιβλιογραφική Ανασκόπηση

### 1.2.1 Εφαρμογές Android

Το λειτουργικό σύστημα Android OS υποστηρίζεται από μία μεγάλη κοινότητα προγραμματιστών που αναπτύσσουν εφαρμογές, οι οποίες επεκτείνουν τη λειτουργικότητα των κινητών συσκευών android. Οι εφαρμογές αναπτύσσονται σε μια προσαρμοσμένη έκδοση της JAVA η οποία διατίθεται δωρεάν από το διαδικτυακό κατάστημα Google Play (πρώην Android Market) όπως και από άλλους διαδικτυακούς τόπους. Μέχρι τον Ιανουάριο του 2015 περισσότερες από 1.300.000 (Εικόνα 1) εφαρμογές ήταν διαθέσιμες για κινητές συσκευές Android ενώ εκτιμάτε ότι ο αριθμός των μεταφορτώσεων από το Google Play μέχρι το Μάιο του 2016 είχε υπερβεί τα 65 δισεκατομμύρια. Το Android είναι η πρώτη σε πωλήσεις παγκοσμίως πλατφόρμα για smartphones (έξυπνες κινητές συσκευές) (Εικόνα 2 **Error! Reference source not found.**).



### Cumulative number of apps downloaded from the Google Play as of May 2016 (in billions)



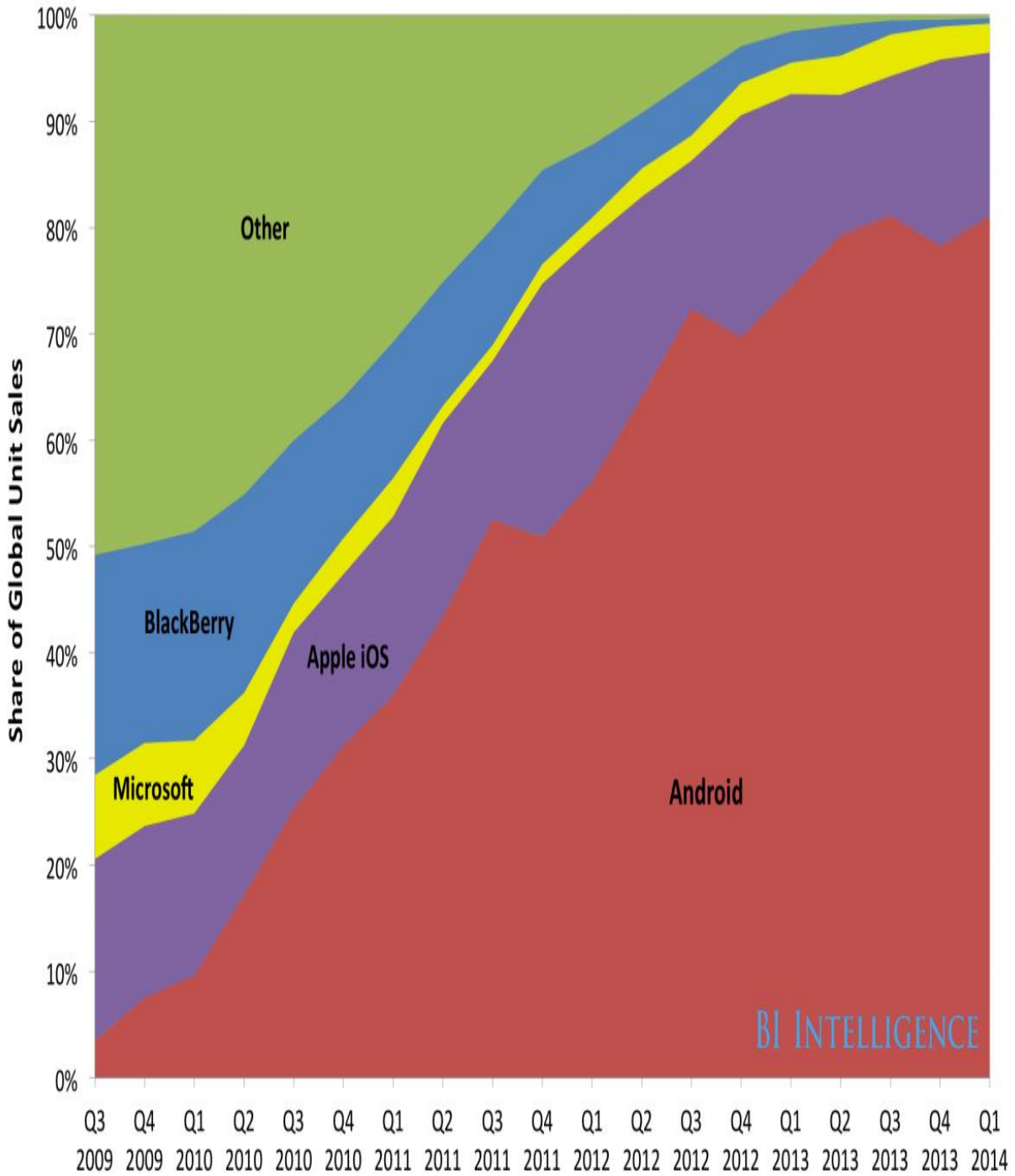
Sources:  
Mashable; Android; Google  
© Statista 2016

Additional Information:  
Worldwide; Google; Android; August 2010 to May 2016



Εικόνα 1 - Google play number of apps

# Global Smartphone Market Share By Platform



Source: IDC, Strategy Analytics

Εικόνα 2 - Android OS market share

## 1.2.2. Εφαρμογές σχετικές με μετρήσεις QoS

### 1. Speed Test & QoS 3G 4G WiFi

- Τρέχουσα έκδοση: 2.2.13
- Εγκαταστάσεις: 100.000 - 500.000
- Απαιτεί Android: 2.3.3 και νεότερες εκδόσεις

### 2. Network Signal Guru

- Τρέχουσα έκδοση: 1.1.4
- Εγκαταστάσεις: 10.000 - 50.000
- Απαιτεί Android: 4.0.3 και νεότερες εκδόσεις

### 3. 4Gmark (3G / 4G speed test)

- Τρέχουσα έκδοση: 1.1.4
- Εγκαταστάσεις: 1.000.000 – 5.000.000
- Απαιτεί Android: 4.0.3 και νεότερες εκδόσεις

### 4. Traffic Monitor & 3G/4G Speed

- Τρέχουσα έκδοση: 2.1.4
- Εγκαταστάσεις: 1.000.000 – 5.000.000
- Απαιτεί Android: 4.0.3 και νεότερες εκδόσεις

### 5. SPEED TEST 4G LTE 3G MAP QoS

- Τρέχουσα έκδοση: 1.17.9
- Εγκαταστάσεις: 1.000.000 – 5.000.000
- Απαιτεί Android: 3.0.1 και νεότερες εκδόσεις

## Κεφάλαιο 2 Θεωρητικό

Σε αυτήν την ενότητα θα περιγραφεί η πλατφόρμα Android και τα βασικά στοιχεία μίας εφαρμογής Android. Πιο συγκεκριμένα, θα περιγράψουν όλες οι εκδόσεις των λειτουργικών android (τα κύρια χαρακτηριστικά τους), η αρχιτεκτονική του συστήματος και η δομή μίας εφαρμογής (φάκελοι source/layout/configuration).

### 2.1. Εισαγωγή

Το Android είναι μια στοίβα λογισμικού για έξυπνες κινητές συσκευές η οποία αποτελείται από λειτουργικό σύστημα, ενδιάμεσο λογισμικό (middleware) και κάποιες βασικές εφαρμογές. Επιπλέον, ο πυρήνας του Android βασίζεται στο λειτουργικό Linux και μέσω της δικιά του εργαλειοθήκης ανάπτυξης συστήματος λογισμικού (Software Development Kit), δίνει τη δυνατότητα στους κατασκευαστές να δημιουργούν πρωτοποριακές καινοτόμες εφαρμογές. Το Android αρχικά αναπτύχθηκε από την Google και αργότερα συνεχίστηκε σε συνεργασία με την Open Handset Alliance (OHA). **Error! Reference source not found.** Η πλατφόρμα του Android παρουσιάστηκε για πρώτη φορά στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού OHA, μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής υλικού, οι οποίες έχουν ως κύριο αντικείμενο την ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές ανοιχτής τηλεφωνίας.

Ενδεικτικά αναφέρουμε μερικά μέλη του οργανισμού αυτού (Εικόνα 1) για να δείξουμε την τεράστια προοπτική που δημιουργείται:

- Sprint Nextel
- T-Mobile
- Motorola
- Samsung
- Sony Ericsson
- Vodafone
- Google

- Verizon
- Texas Instruments
- Htc



Εικόνα 3 - Εταιρίες λογισμικού και κατασκευής υλικού παγκόσμιας εμβέλειας

Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android 0, υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού.



Εικόνα 4 - Λογότυπο πλατφόρμας android

## 2.2. Εκδόσεις και χαρακτηριστικά

Τον Νοέμβριο του 2007 εμφανίστηκε η πρώτη έκδοση του Android SDK **Error! Reference source not found.**, η οποία χαρακτηρίστηκε από τους κατασκευαστές του σαν μια πρώτη ματιά στο SDK του Android. Τότε πολλοί κατασκευαστές βιάστηκαν να το κατακρίνουν το Android σαν ένα προβληματικό σύστημα παραβλέποντας ότι ήταν ακόμα σε νηπιακή μορφή. Στην πραγματικότητα όμως το Android δεν παρουσίαζε προβλήματα τα οποία δεν παρουσιάζει οποιοδήποτε σύστημα σε τέτοια πρώιμη φάση. Η T-Mobile το Σεπτέμβριο του 2008 **Error! Reference source not found.**, με ένα δελτίο τύπου ανακοίνωσε την διαθεσιμότητα του T-Mobile G1, του πρώτου έξυπνου τηλεφώνου (smartphone), το οποίο λειτουργεί με την πλατφόρμα του Android. Λίγες μέρες αργότερα (Οκτώβριο 2008), η Google ανακοινώνει την απελευθέρωση του SDK Release Candidate 1.0.

Τον Φεβρουάριο του 2009 **Error! Reference source not found.** παρουσιάστηκε η έκδοση Android 1.1 σαν μια ανανεωμένη έκδοση του 1.0. Σε εκείνη την χρονική το Android δεν είχε την δυνατότητα χρήσης κουμπιών αφής, παρά μόνο της χρήσης των κλασικών ‘σκληρών’ κουμπιών της συσκευής.

Τον Μάιο του 2009 είχαμε την έκδοση Android 1.5 **Error! Reference source not found.**, εν ονόματι ‘Cupcake’.



Εικόνα 5 -Λογότυπο Android 1.5 CupCake

Το ‘Cupcake’[2] προσθέτει επιπλέον δυνατότητες και χαρακτηριστικά καθώς και μία ανανεωμένη διεπαφή (κινούμενες μεταβάσεις οθόνης) χρήστη (User Interfac). Πιο συγκεκριμένα, στην καινούρια έκδοση οι κινητές συσκευές έχουν την ικανότητα για καταγραφή και παρακολούθηση βίντεο με τη χρήση της βιντεοκάμερας του κινητού, μεταφόρτωση βίντεο στο διαδικτυακό τόπο

YouTube και φωτογραφιών στο διαδικτυακό τόπο Picasa, ανανεωμένο πληκτρολόγιο (αφής) με πρόβλεψη κειμένου(T9), επικοινωνίας μέσω του προτύπου Bluetooth A2DP και AVRCP, αυτόματης σύνδεση σε μικροσυσκευή Bluetooth από μια συγκεκριμένη απόσταση.

Το 'Donut', Android 1.6, ήρθε τον Σεπτέμβριο του 2009.



Εικόνα 6 - Λογότυπο android 1.6 DONUT

Η έκδοση αυτή εισάγει κάποια καινούργια χαρακτηριστικά όπως το βελτιωμένο Android Market, η "τρία σε ένα" φωτογραφική μηχανή, η εμφάνιση βιντεοκάμερα, η διεπαφή (interface) σε μορφή γκαλερί, η δυνατότητα επιλογής και διαγραφής πολλών φωτογραφιών ταυτόχρονα, η ικανότητα αναζήτησης με φωνή, η ταχύτερη απόκριση και βαθύτερη ολοκλήρωση με εγγενής (native) εφαρμογές, συμπεριλαμβανομένης της δυνατότητας να καλούμε επαφές, η υποστήριξη τεχνολογιών για CDMA/EVDO, 802.1x, VPNs και με μηχανή μετατροπής κειμένου σε ομιλία (text-to-speech) , η υποστήριξη για ανάλυση οθονών WVGA και οι βελτιώσεις στην ταχύτητα της φωτογραφικής μηχανής.

Ακολουθεί το 'Eclair', Android 2.0 **Error! Reference source not found.** τον Νοέμβριο 2009, με τις επανεκδόσεις του σε Android 2.0.1 τον Δεκέμβριο 2009 (Eclair 0.1) και τον Ιανουάριο 2010 με το Android 2.1 (Eclair MR1).



**Εικόνα 7 – Λογότυπο Android 2.0 ECLAIR**

Οι πιο σημαντικές αλλαγές [4] είναι η ταχύτερη ανταπόκριση του υλικού, η υποστήριξη πολλαπλών οθονών και αναλύσεων, η πιο φιλική προς το χρήστη διεπαφή, η υποστήριξη του προτύπου html5, η υποστήριξη Microsoft Exchange, η ενσωματωμένη υποστήριξη flash για την Camera, η υποστήριξη ψηφιακής μεγέθυνσης (zoom), το ανανεωμένο εικονικό πληκτρολόγιο και το Bluetooth 2.1

Ακολουθεί το Android 2.2 με το όνομα 'Froyo' τον Μάιο του 2010.



**Εικόνα 8 - Λογότυπο Android 2.2 Froyo**

Οι σημαντικότερες αλλαγές στην έκδοση FROYO [5] είναι οι βελτιώσεις στον πυρήνα του λειτουργικού συστήματος με τις οποίες επιτεύχθηκαν βελτιώσεις στην ταχύτητα επεξεργασία και απόκρισης, στην μνήμη και στην απόκριση. Επιπλέον, υποστηρίζεται σχεδόν πλήρως ο Microsoft Exchange (σε πολιτικές ασφαλείας, συγχρονισμού ημερολογίου, auto – discovery, GAL look-up, remote wipe). Δίνεται στον χρήστη η δυνατότητα χρήση συντομεύσεων προς τις εφαρμογές τηλεφώνου και της Μηχανής Αναζήτησης. Επιπροσθέτως, είναι εφικτή η πρόσδεση USB και λειτουργία δυναμικής ζώνης (hotspot) WiFi. Ο χρήστης έχει ακόμα την δυνατότητα να επιλέξει την μη πρόσβαση δεδομένων πάνω από ένα συσκευασμένο δίκτυο κινητής τηλεφωνίας. Το



πληκτρολόγιο γίνεται πιο εύχρηστο και επιτρέπεται η εναλλαγή σε πολλαπλές γλώσσες του πληκτρολογίου και των λεξικών τους. Ο χρήστης έχει ακόμα τη δυνατότητα Φωνητικής κλήσης. Από αυτήν την έκδοση του Android και εφεξής είναι εφικτή η εγκατάσταση εφαρμογών στην επεκτάσιμη μνήμη και η υποστήριξη εφαρμογών και ιστοσελίδων με χρήση Adobe Flash.

Η επόμενη έκδοση για κινητά smartphones, από τον Δεκέμβριο του 2010 είναι η έκδοση Android 2.3 με το όνομα “Gingerbread” με την επανέκδοση του σε Android 2.3.3 τον Φεβρουάριο του 2011.



Εικόνα 9 - Λογότυπο Android 2.3 GingerBread

Η 2.3 έκδοση (Gingerbread) **Error! Reference source not found.** [6] βελτίωσε τη διεπαφή χρήστη, βελτίωσε το πληκτρολόγιο αφής και τις δυνατότητες αντιγραφής-επικόλλησης, βελτίωσε τις επιδόσεις στα παιχνίδια, πρόσθεσε SIP υποστήριξη (κλήσεις VOIP) και πρόσθεσε υποστήριξη για την επικοινωνία κοντινού πεδίου (Near Field Communication).

Η έκδοση 3.0 (με όνομα Honeycomb) **Error! Reference source not found.** προοριζόταν κυρίως για χρήση σε tablet. Η έκδοση Honeycomb υποστηρίζει συσκευές με μεγαλύτερες οθόνες και αναλύσεις και παρουσιάζονται σε αυτήν πολλά νέα χαρακτηριστικά διεπαφής χρήστη. Επιτρέπει την υποστήριξη επεξεργαστών πολλών πυρήνων (multicore processors), την επιτάχυνση υλικού γραφικών για βελτιωμένες επιδόσεις του λειτουργικού συστήματος και των εφέ που χρησιμοποιεί ενώ έχει υποστήριξη πλήρους συστήματος κρυπτογράφησης. Η βελτιωμένη έκδοση 3.1 κυκλοφόρησε το Μάιο του 2011 και πρόσθεσε υποστήριξη για περισσότερες συσκευές εισόδου, λειτουργία USB για τη μεταφορά δεδομένων απευθείας από φωτογραφικές μηχανές και άλλες συσκευές, καθώς και υποστήριξη για Google Movies και Books Apps. Η επόμενη έκδοση η 3.2

**Error! Reference source not found.** που κυκλοφόρησε τον Ιούλιο του 2011 πρόσθεσε βελτιστοποίηση για ένα ευρύτερο φάσμα μεγεθών οθόνης, νέο “zoom-to-fill” χαρακτηριστικό έτσι ώστε να μεγεθύνεται η οθόνη σε Full screen χωρίς να υπάρχουν σφάλματα στην εικόνα ή προβλήματα συμβατότητας, πρόσθεσε υποστήριξη για φόρτωση αρχείων πολυμέσων απευθείας από την κάρτα SD και ένα εκτεταμένο API υποστήριξης οθόνης. Το HuaweiMediaPad ήταν το πρώτο tablet 7 ιντσών που χρησιμοποίησε αυτήν την έκδοση.



Εικόνα 10 - Λογότυπο Android 3.0 Honeycomb

Μερικά από τα χαρακτηριστικά του [6] είναι η υποστήριξη διπύρηνων και τετραπύρηνων επεξεργαστών, η βελτιωμένη υποστήριξη των ταμπλετών ανάπτυξη λογισμικού (scripting) για 3D, σε γλώσσα η οποία καλείται "Renderscript", η εμφάνιση του Video chat μέσω Google Talk, τα Google eBooks και η "Ιδιωτική περιήγηση".

Η επόμενη έκδοση όπως ανακοινώθηκε στο Google I/O 2011 στο San Francisco, πρόκειται να είναι το “Ice Cream Sandwich”, το οποίο αποτελεί την προσπάθεια της εταιρίας για ενιαίο λειτουργικό σύστημα για όλες τις συσκευές.



**Εικόνα 11 - Λογότυπο Android Ice Cream Sandwich**

Η 4.0 έκδοση (Ice Cream Sandwich) **Error! Reference source not found.** ανακοινώθηκε στις 19 Οκτωβρίου 2011, έφερε τα χαρακτηριστικά της Honeycomb στα smartphones. Έχει προσθέσει νέα χαρακτηριστικά όπως αναγνώριση προσώπων μέσω της κάμερας, παρακολούθηση και έλεγχο των δεδομένων και της χρήσης του δικτύου, ενιαίες κοινωνικές επαφές δικτύωσης, βελτιώσεις των φωτογραφιών, αναζήτηση e-mail ακόμα και σε offline κατάσταση, φακέλους για τα προγράμματα και διαμοιρασμό πληροφοριών με χρήση της τεχνολογίας NFC (NearFieldCommunication). Η έκδοση 4.0.3 Ice Cream Sandwich είναι η πιο πρόσφατη έκδοση του Android που είναι διαθέσιμη σε κινητά τηλέφωνα. Ο πηγαίος κώδικας του Android 4.0.1 κυκλοφόρησε στις 14 Νοεμβρίου 2011.

Η επόμενη έκδοση του Android είναι η 4.1 **Error! Reference source not found.** η οποία ονομάζεται Jelly Bean. Η έκδοση Jelly Bean περιλαμβάνει τις εκδόσεις 4.1 έως 4.3.1[8]. Το Android Jelly Bean, έχει επιπλέον βελτιώσεις στην επίδοση και ανανεωμένη διεπαφή χρήστη. Επιπλέον, παρέχει τη δυνατότητα δημιουργίας προφίλ και εκχώρησης συγκεκριμένων δικαιωμάτων, υποστηρίζει Bluetooth Smart support, Dial pad autocomplete, υποστηρίζει γλώσσες όπως Αραβικά/Εβραϊκά και γραφικά OpenGL ES 3.0



**Εικόνα 12 - Λογότυπο Android Jelly Bean**

Η επόμενη έκδοση του Android είναι η 4.4 KitKat.

Μερικά από τα νέα χαρακτηριστικά αυτής της έκδοσης[9] είναι η ταχύτερη υποστήριξη και επεξεργασία πολλαπλών εργασιών (multitasking), η φιλικότερη προς το χρήστη κατηγοριοποίηση των μηνυμάτων και των επαφών, η σουίτα εργαλείων QuickOffice, η υποστήριξη Bluetooth MAP, η ενσωματωμένη διαχείριση συσκευών και η καλύτερη κατανάλωση ενέργειας.



Εικόνα 13 - Λογότυπο Android Kitkat

Η τελευταία έκδοση του Android είναι η 5.0 η οποία ονομάζεται lollipop.



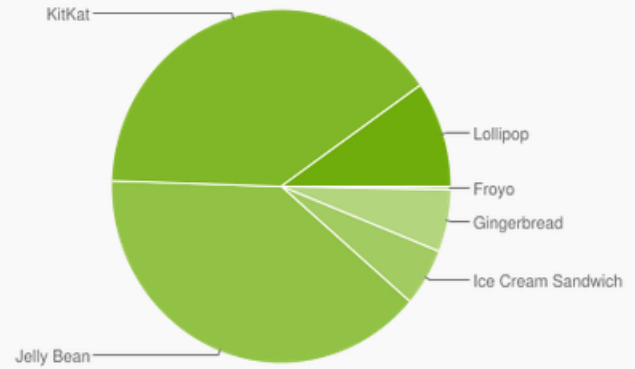
Εικόνα 14 - Λογότυπο Android Lollipop

[10]

### ***Συνολικός Πίνακας Εκδόσεων***

Στον παρακάτω πίνακα εμφανίζονται το μερίδιο των συσκευών οι οποίες λειτουργούν με την κάθε έκδοση του λειτουργικού Android.

Version	Codename	API	Distribution
2.2	Froyo	8	0.3%
2.3.3 - 2.3.7	Gingerbread	10	5.7%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	5.3%
4.1.x	Jelly Bean	16	15.6%
4.2.x		17	18.1%
4.3		18	5.5%
4.4	KitKat	19	39.8%
5.0	Lollipop	21	9.0%
5.1		22	0.7%



Εικόνα 15 - Αναλογία Συσκευών Λειτουργικών

## **2.3. Αρχιτεκτονική του Android**

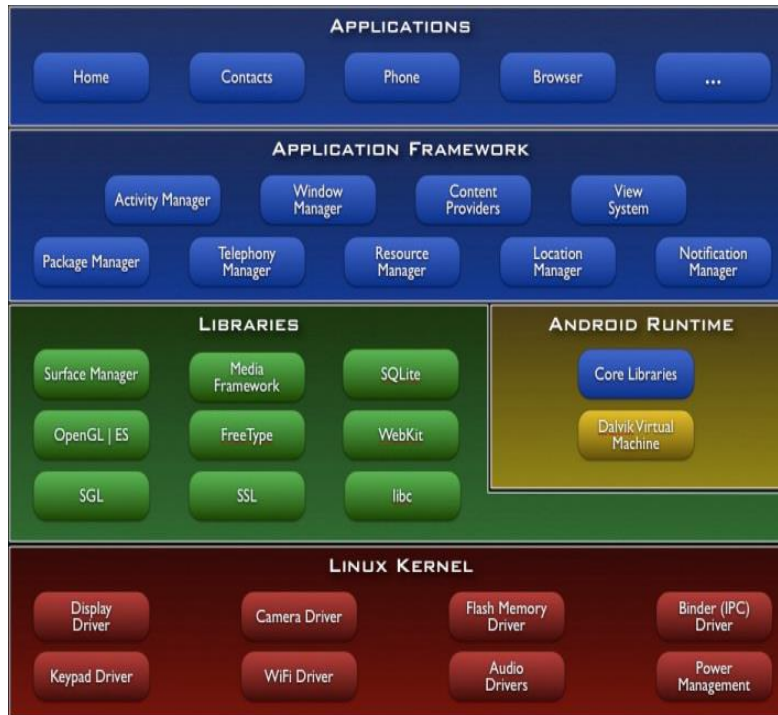
Η αρχιτεκτονική του Android στηρίζεται στα παρακάτω επίπεδα :

**Πυρήνας του Linux (Linux Kernel):** Το λειτουργικό σύστημα έχει παρόμοιες αρχές λειτουργίας με τον πυρήνα του Linux. Με τη χρήση του πυρήνα του Linux, το Android εγκαθίσταται σε μια σειρά από πλατφόρμες. Επιπλέον, στο Android ο πυρήνας του Linux χρησιμοποιείται για την διεκπαιρέωση λειτουργιών όπως την διαχείριση μνήμης, την διαχείριση διεργασιών, την δικτύωση και άλλες.

**Επίπεδο Βιβλιοθηκών (Libraries Layer):** Οι βιβλιοθήκες του Android έχουν αναπτυχθεί σε γλώσσα προγραμματισμού C/C++. Όπως παρουσιάζεται και από την παραπάνω εικόνα οι βιβλιοθήκες διαχειρίζονται το Surface Manager, το Media Framework, το SQLite, το OpenGL κ.α.

**Επίπεδο Εφαρμογών (Applications Layer):** Σε αυτό το επίπεδο βρίσκονται οι εφαρμογές από τις οποίες αποτελείται το Android. Όλες οι εφαρμογές είναι γραμμένες στην γλώσσα προγραμματισμού Java. Τέτοιες εφαρμογές είναι ο περιηγητής, οι επαφές, το ημερολόγιο κ.α.

**Επίπεδο Πλαισίου Εφαρμογών (Applications Framework Layer):** Μέσω αυτού του επιπέδου είναι εφικτή η χρήση των πόρων της συσκευής αλλά και η κλήση APIs. Το Android παρέχει τη δυνατότητα χρήσης υπηρεσιών όπως Google Maps, Notifications (ειδοποιήσεις), Contacts(Επαφές) κ.α.



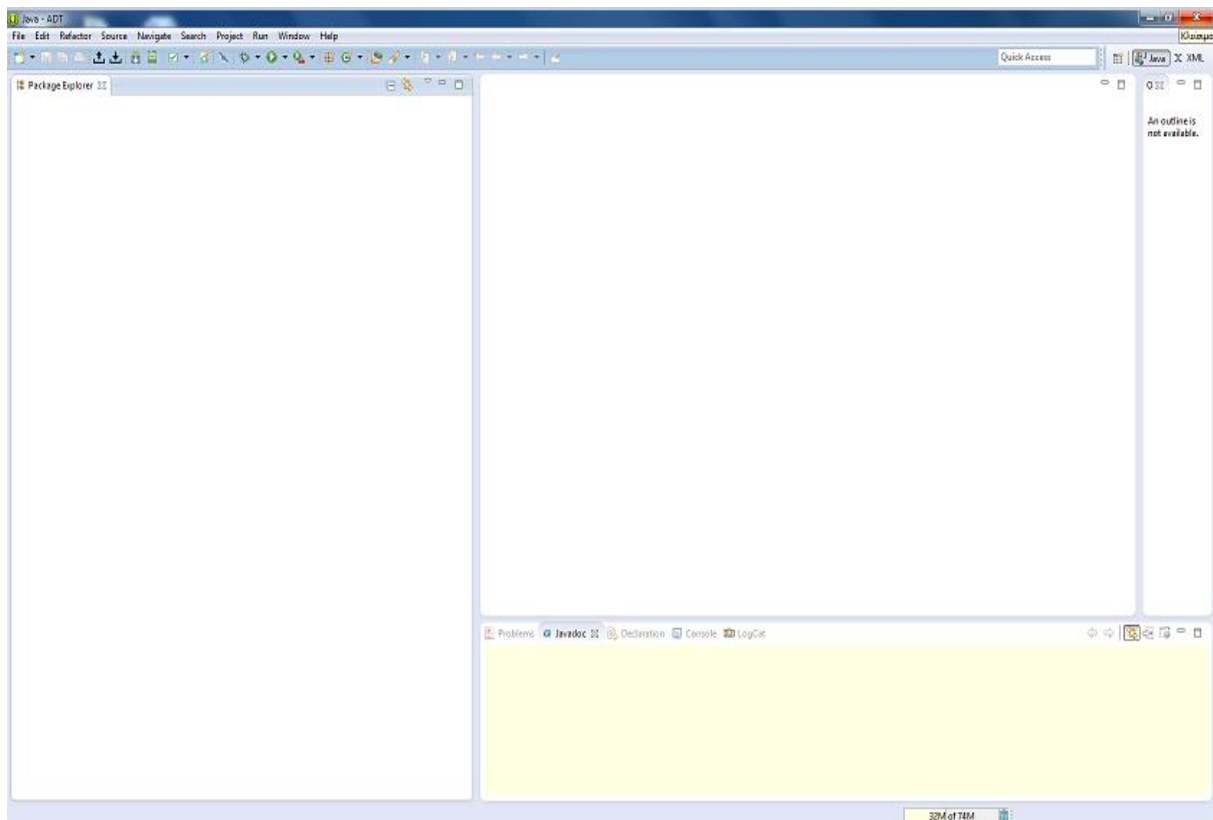
Εικόνα 16 - Η Αρχιτεκτονική του Android

### 2.3.1 Εργαλεία προγραμματισμού Android

Για την ανάπτυξη εφαρμογών Android είναι απαραίτητη η γνώση της γλώσσας προγραμματισμού JAVA ή η χρήση ενός application framework για τη δημιουργία εφαρμογών για κινητές συσκευές. Επιπροσθέτως, είναι απαραίτητη η χρήση ενός περιβάλλοντος ανάπτυξης εφαρμογής (Integrated Development Environment, IDE) για την μεταγλώττιση και εκτέλεση των εφαρμογών όπως το Eclipse ή NetBeans μέσω του οποίου θα χρησιμοποιείται το Java Development Kit (JDK).

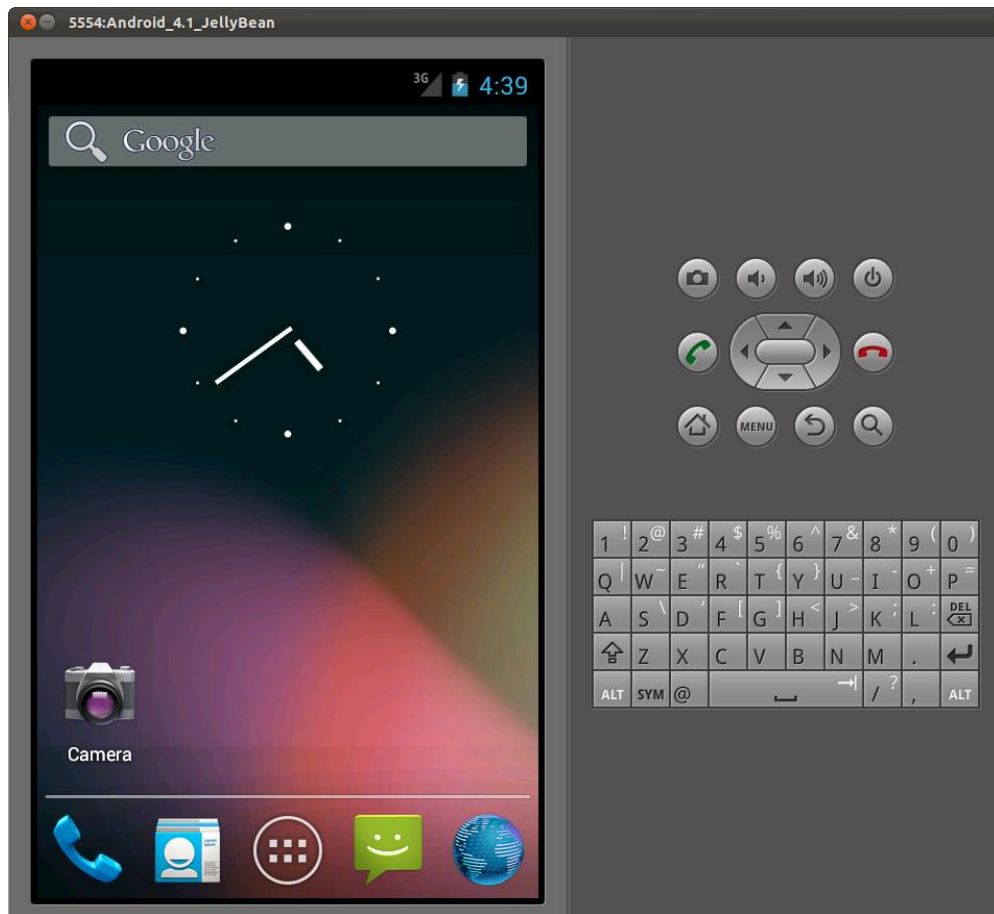
Μέσω του IDE εγκαθίσταται όλες τις απαραίτητες βιβλιοθήκες καθώς και εργαλεία ώστε να είναι ευκολότερος ο προγραμματισμός σε περιβάλλον Android όπως το Android Software Development Kit (AndroidSDK). Η σύνδεση του Android SDK με το γραφικό περιβάλλον του IDE πραγματοποιείται μέσω της επέκτασης Android Development Tools Plugin ή αλλιώς ADT Plugin. Σε παρακάτω ενότητα θα περιγραφεί η εγκατάσταση του Eclipse, του ADT plugin καθώς και του framework το οποίο είναι απαραίτητο για την εφαρμογή που θα αναπτυχθεί στα πλαίσια της πτυχιακής.

Το Eclipse προσφέρει την δυνατότητα κατά την διάρκεια της ανάπτυξης της εφαρμογής της άμεσης εκτέλεσης σε περιβάλλον Android στον υπολογιστή, μέσω μίας εικονικής συσκευής του Android Virtual Device (AVD), ενός εικονικού προσομοιωτή τόσο του software όσο και του hardware ενός κινητού τηλεφώνου με λειτουργικό σύστημα Android. Ο εικονικός προσομοιωτής εγκαθίσταται μέσω της επέκτασης ADT Plugin. Κατά την εξατομίκευση της εικονικής συσκευής βάση των απαιτήσεων της εφαρμογής έχουμε την δυνατότητα επιλογής των χαρακτηριστικών των οποίων είναι απαραίτητα όπως παραδείγματος χάρη την έκδοση Android που θα χρησιμοποιεί, το μέγεθος της κάρτας SD και της cache, καθώς και άλλα χαρακτηριστικά όπως του δέκτη GPS κλπ.



Εικόνα 17 - Το περιβάλλον του Eclipse





Εικόνα 18 - Η εικονική συσκευή του Eclipse

### 2.3.2 Συστατικά μέρη μίας Android εφαρμογής

Η εκάστοτε εφαρμογή Android είναι απαραίτητο να εμπεριέχει κάποια συγκεκριμένα δομικά στοιχεία ώστε αυτή να είναι λειτουργική. Τα δομικά στοιχεία παρέχουν ένα τρόπο πρόσβασης του λειτουργικού συστήματος στην εφαρμογή. Τα βασικά στοιχεία μιας Android εφαρμογής αντιστοιχίζονται στα παρακάτω τέσσερα βασικά συστατικά:

- τις δραστηριότητες (activities),
- τις υπηρεσίες (services),
- τους παρόχους περιεχομένου (content providers) και
- τους παραλήπτες μηνυμάτων (broadcast receivers).

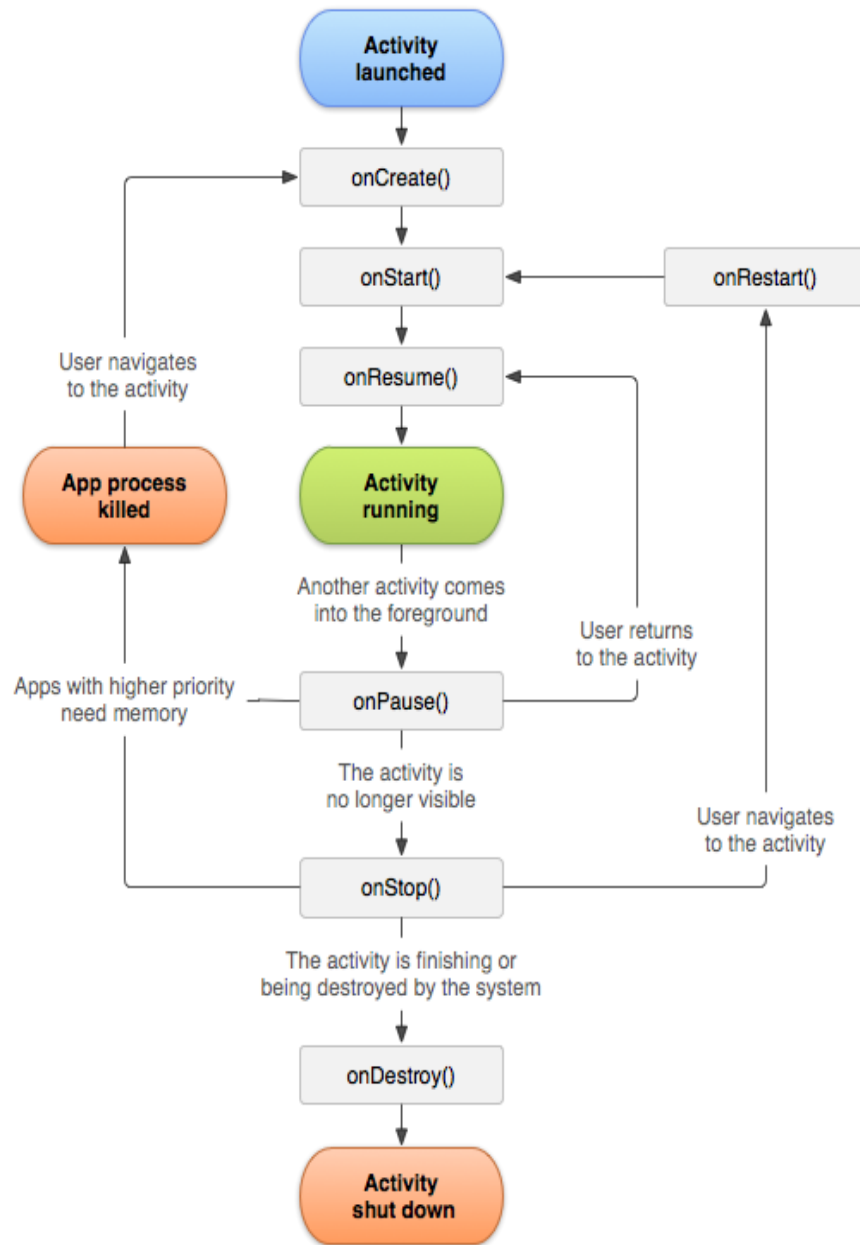
## Δραστηριότητες (Activities)

Είναι το σημαντικότερο στοιχείο κάθε εφαρμογής. Κάθε οθόνη αντιστοιχεί σε μία διαφορετική οθόνη (παράθυρο) της εφαρμογής με την οποία ο χρήστης αλληλεπιδρά. Το γραφικό περιβάλλον της εφαρμογής φορτώνεται από τον φάκελο Activity. Κάθε εφαρμογή μπορεί να αποτελείται από πολλές activities, οι οποίες συνδέονται μεταξύ τους και αλληλεπιδρούν.

Η πρώτη οθόνη της εφαρμογής αποτελεί την main activity (κύρια δραστηριότητα), της οποίας έπονται οι υπόλοιπες. Όταν καλείται από τον χρήστη μία Activity τότε παρουσιάζεται μία νέα οθόνη, ενώ με την επιλογή "back" υπάρχει η δυνατότητα επιστροφής στην προηγούμενη Activity.

Κάθε μία activity υλοποιεί τη μέθοδο onCreate() η οποία καλείται τη στιγμή που ξεκινάει η activity. Ακόμη εντός της μεθόδου πρέπει να δημιουργηθεί η διεπαφή χρήστη (User Interface ή UI), η οποία δηλώνεται με την εντολή setContentView() και η findViewById() για να ανακτήσει τα διάφορα widgets προγραμματιστικά. [13]

Το Android διαχειρίζεται ολοκληρωτικά τον κύκλο ζωής της κάθε Activity.



Εικόνα 19 - Ο κύκλος ζωής των Activities

## Υπηρεσίες (Services)

Η εκτέλεση των υπηρεσιών (services) είναι ανεξάρτητη και δεν συνδέεται με την εκάστοτε Activity. Η κύρια διαφορά της service με την activity είναι ότι μία υπηρεσία δεν παρέχει περιβάλλον διεπαφής χρήστη (user interface). Επίσης, η υπηρεσία μπορεί να εκτελείται ακόμα και όταν εκτελείται κάποια διαφορετική εφαρμογή από αυτήν που την ξεκίνησε.

Ένα παράδειγμα υπηρεσίας είναι η εφαρμογή για αναπαραγωγή μουσικής. Δηλαδή μπορεί ο χρήστης να ακούει μουσική (δίχως την προβολή κάποιου γραφικού περιβάλλοντος) και ταυτόχρονα να εκτελεί άλλη εφαρμογή. Μία υπηρεσία μπορεί να βρίσκεται σε δύο διαφορετικές καταστάσεις.

- **H Started:** Μία activity εκτελεί την συνάρτηση `startService()`. Η συνάρτηση `startService()` ενεργοποιεί κάποια λειτουργία του συστήματος ή μια διαδικασία γενικά και δεν επηρεάζεται από το αν η activity που την κάλεσε τερματιστεί. Παράδειγμα μιας υπηρεσίας (service) είναι η μεταφόρτωση ενός αρχείου από το διαδίκτυο.
- **H Bound:** Η `bindService()` «δένει» τη service με κάποια άλλη μονάδα της εφαρμογής, όπως με μία activity. Μεταξύ τους υπάρχει αλληλοεξάρτηση και αλληλοεπίδραση π.χ. αποστολή και λήψη αποτελεσμάτων. Οπότε σε αυτήν την περίπτωση αν τερματιστεί η activity τερματίζεται και η service. Η service δεν διαθέτει διεπαφή χρήστη άρα για τα αποτελέσματα της υπηρεσίας ενημερωνόμαστε με δυο τρόπους: η ειδοποίηση Toast και η ειδοποίηση μέσω Status Bar μέσω της διεπαφής της activity. [17]

## Πάροχοι Περιεχομένου (Content Providers)

Οι πάροχοι περιεχομένου (Content Providers) είναι υπεύθυνοι για τη διαχείριση των αποθηκευτικών χώρων οι οποίοι είναι προσβάσιμοι από οποιαδήποτε εφαρμογή της συσκευής. Για παράδειγμα μία εφαρμογή «Πελάτες» θα ήταν θεμιτό να αποθηκεύει τους πελάτες μας ώστε να μπορούν προσβάσιμο αποθηκευτικό χώρο. Στη συνέχεια μία εφαρμογή «Ραντεβού» να μπορεί να εισάγει δεδομένα από την εφαρμογή «Πελάτες». Το Android διαθέτει την ενσωματωμένη βάση δεδομένων (SQLite database), στην οποία μπορούν να αποθηκεύουν ή να διαβάζουν δεδομένα οι Content Providers.

## **Broadcast Receivers**

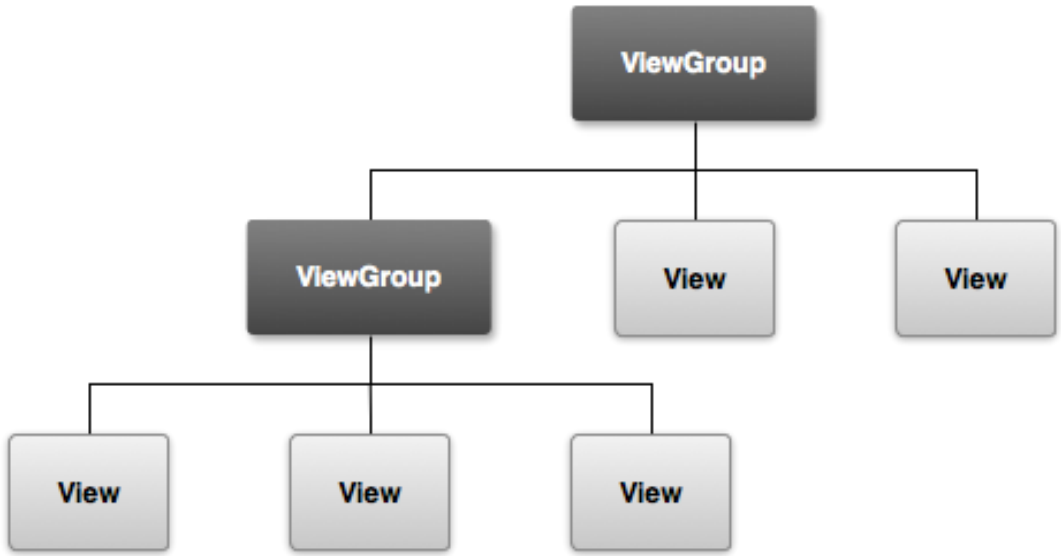
Οι Broadcast Receivers είναι υπεύθυνοι για την ενημέρωση του χρήστη για ανάληψη κάποιων ενεργειών. Για παράδειγμα όταν η στάθμη της μπαταρίας είναι χαμηλή ενημερώνεται ο χρήστης ότι πρέπει να συνδέσει τη συσκευή για φόρτιση. Όταν η στάθμη της μπαταρίας είναι χαμηλή τότε απενεργοποιείται το φλας στη λήψη φωτογραφιών για την εξοικονόμηση ενέργειας.[13]

## **Διεπαφή Χρήστη (User Interface)**

Η διεπαφή χρήστη (User Interface, UI) είναι το γραφικό περιβάλλον που εμφανίζεται στην συσκευή του χρήστη. Όλα τα αντικείμενα της διεπαφής χρήστη σε μία εφαρμογή Android δημιουργούνται χρησιμοποιώντας αντικείμενα View και ViewGroup. Ένα αντικείμενο View εμφανίζεται στην Οθόνη ως ένα γραφικό στοιχείο στην οθόνη μέσω του οποίου ο χρήστης μπορεί να αλληλεπιδράσει με την εφαρμογή. Ένα αντικείμενο Viewgroup συγκεντρώνει το σύνολο των αντικειμένων View και καθορίζει τη διάταξη της διεπαφής χρήστη και το περιβάλλον της εφαρμογής γενικότερα.

Το Android προσφέρει μία συλλογή από View και ViewGroup υπο-κλάσεις (subclasses) τα οποία προσφέρουν στοιχεία ελέγχου και ποικίλα μοντέλα διάταξης όπως το linear και το relative layout τα οποία θα εξηγήσουμε στην συνέχεια.

Η διεπαφή χρήστη για κάθε μέρος της Android εφαρμογής καθορίζεται από αυστηρή ιεραρχία δένδροειδούς ιεραρχικής δομής η οποία δύναται να είναι όσο απλή ή πολύπλοκη γίνεται αναλογικά με τις ανάγκες και τις απαιτήσεις της εκάστοτε εφαρμογής. [13]



Εικόνα 20 - Η ιεραρχική δομή της Διεπαφής Χρήστη

Ακολουθεί ένα απλό παράδειγμα για κάθετη διάταξη

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="horizontal" >

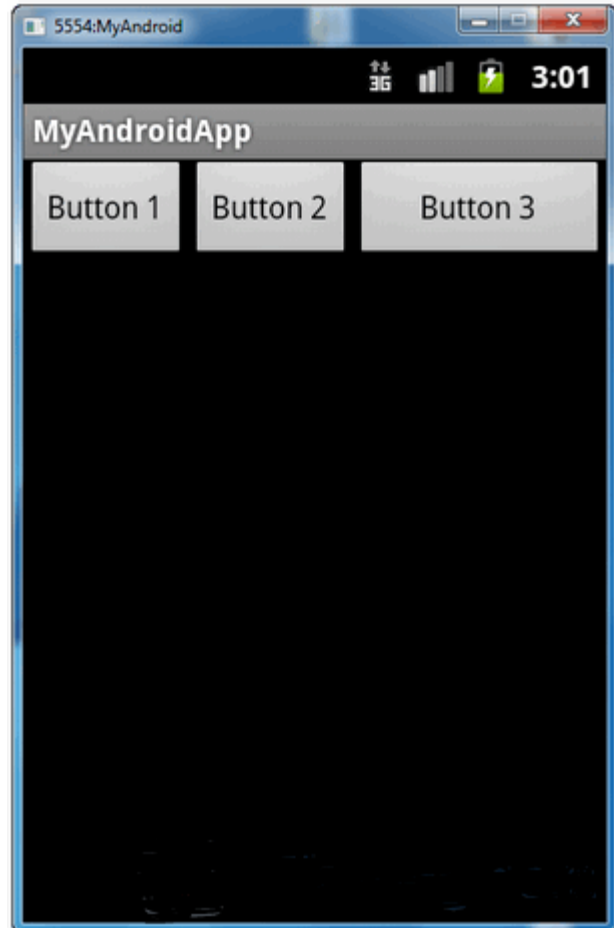
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 1" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 2" />

    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 3"
        android:layout_weight="1"/>

</LinearLayout>

```



Εικόνα 21 - Linear Layout

## Διάταξη των γραφικών στοιχείων (Layout)

Κάθε οθόνη μιας εφαρμογής Android αποτελείται από ένα σύνολο γραφικών στοιχείων. Όλα τα γραφικά στοιχεία της εφαρμογής αποτελούν το layout της εφαρμογής. Υπομάδες των γραφικών στοιχείων μπορεί να συγκροτούν επιμέρους layouts. Τα layout χωρίζονται σε:

- **LinearLayout(γραμμική διάταξη):** Αποτελεί διάταξη στοιχείων σε οριζόντια ή κατακόρυφη σειρά. Αν δηλώσουμε τρία στοιχεία A,B,C σε ένα οριζόντιο LinearLayout τότε τα στοιχεία αυτά θα εμφανίζονται στην οθόνη σε μία οριζόντια διάταξη με τη σειρά που τα δηλώσαμε το ένα δίπλα στο άλλο.

- **RelativeLayout(σχετική διάταξη):** Είναι αρκετά ελαστικό στη δήλωση των αντικειμένων και των γραφικών στοιχείων. Δίνεται η δυνατότητα να επιλεγθεί οποιοδήποτε στοιχείο και να εμφανιστεί σε οποιαδήποτε θέση της οθόνης.[14]
- **List View (διάταξη λίστας):** Απαρτίζεται από μία κυλιόμενη λίστα αντικειμένων στην οποία τα γραφικά στοιχεία της εφαρμογής εισέρχονται αυτόματα μέσω ενός προσαρμογέα (Adapter) ο οποίος τα εισάγει από έναν πίνακα ή μία βάση δεδομένων. [14]
- **Grid View (διάταξη πλέγματος):** Τα γραφικά στοιχεία και αντικείμενα της εφαρμογής εμφανίζονται σε ένα δισδιάστατο κυλιόμενο πλέγμα. [15]

Τα παραπάνω layout ορίζονται με τους παρακάτω τρόπους:

α) Δηλώνοντας τα αρχεία μας με XML. Το Android προσφέρει ένα απλό XML λεξιλόγιο που αντιστοιχεί τις View κλάσεις και υποκλάσεις όπως αυτές των widgets και των διατάξεων.

β) Με στοιχεία διάταξης κατά τον χρόνο εκτέλεσης της εφαρμογής μας. Η εφαρμογή μας δύναται να δημιουργήσει View και ViewGroup αντικείμενα προγραμματιστικά.

Το Android μας δίνει την δυνατότητα να χρησιμοποιήσουμε μαζί ή ξεχωριστά τις παραπάνω μεθόδους κατά την δήλωση και την διαχείριση της διεπαφής χρήστη της εφαρμογής μας. [16]

## Έλεγχοι Εισόδου (Input Controls)

Είναι τα διαδραστικά στοιχεία που μπορούμε να συμπεριλάβουμε στο περιβάλλον της εφαρμογής μας. Το Android μας παρέχει ένα εύρος τέτοιων Ελέγχων Εισόδου όπως κουμπιά (buttons), πεδία κειμένου (text fields), πεδία επιλογής (checkboxes) και άλλα. [18]

## Menu

Τα μενού αποτελούν ένα σημαντικό κομμάτι της διεπαφής χρήστη για κάθε οθόνη της εφαρμογής, διότι παρέχουν στο χρήστη ένα γνωστό τρόπο για να εισάγει τις επιλογές του. Στο λειτουργικό σύστημα Android υπάρχουν τρία διαφορετικά είδη μενού: το μενού επιλογών (options menu), το μενού πλαισίου (context menu), το υπομενού (submenu) και τα τρία δηλώνονται σε αρχεία xml.



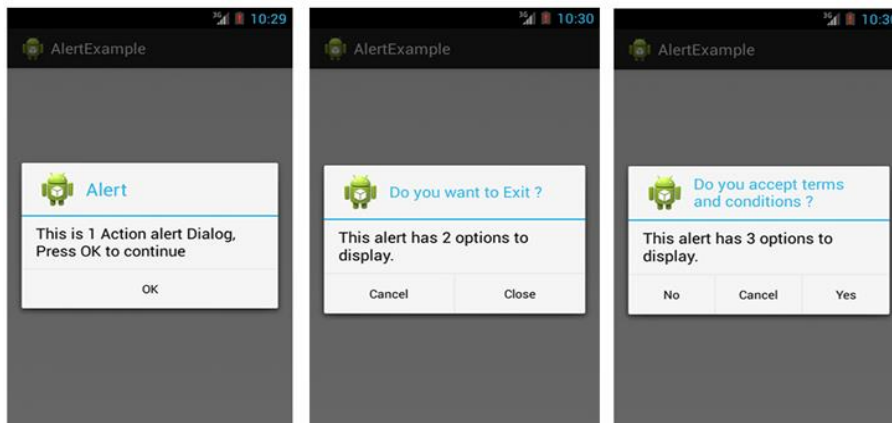
- **Options Menu:** Στο Options Menu περιέχονται οι βασικές επιλογές μιας εφαρμογής. Εμφανίζεται κατά την επιλογή του κουμπιού μενού του κινητού μας. Μέσω αυτού του μενού περιηγούμαστε μεταξύ των διαφορετικών οθονών και activities της εφαρμογής μας.
- **Context Menu:** Το context menu είναι το αντίστοιχο δεξί κλικ των windows. Υπάρχει η δυνατότητα να οριστεί σε οποιοδήποτε γραφικό στοιχείο (εικόνα, κείμενο, κλπ). Στις κινητές συσκευές που δεν υπάρχει ποντίκι για την ενεργοποίησή του αρκεί το παρατεταμένο πάτημα (press and hold ή long press) του στοιχείου αυτού. Για παράδειγμα, στο context menu ενός κειμένου θα ορίζαμε τις επιλογές «αντιγραφή», «αποκοπή», «επικόλληση».
- **Submenu:** Το submenu συνδέεται με ένα από τα δύο μενού παρέχοντας στο χρήστη επιπλέον επιλογές και λειτουργίες. Το χρησιμοποιούμε κυρίως όταν είναι απαραίτητο να χωριστεί μία λειτουργία σε επιμερους λειτουργίες για διευκόλυνση του χρήστη. [19]

## Διάλογοι (Dialogs)

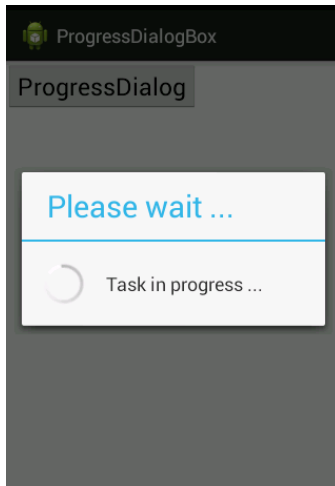
Ο dialog είναι συνήθως ένα μικρό παράθυρο που εμφανίζεται στην οθόνη μπροστά από την activity που την κάλεσε. Το παράθυρο του διαλόγου είναι το μοναδικό με το οποίο μπορεί να αλληλεπιδράσει ο χρήστης. Χρησιμοποιείται είτε για να ορίσει ο χρήστης κάποια επιλογή του είτε για ενημέρωση του χρήστη για κάποιο γεγονός.

Τα δύο πιο σημαντικά είδη διαλόγων είναι:

- **ο AlertDialog (διάλογος ειδοποίησης):** Ο AlertDialog περιέχει τίτλο, κάποιο μήνυμα (συνήθως ερώτηση) και είτε ορισμένα κουμπιά είτε μια λίστα από επιλογές. Ανάλογα με το ποιο κουμπί ή επιλογή επιλέξει ο χρήστης εκτελείται κάποια συγκεκριμένη ενέργεια.



- **ο ProgressDialog (διάλογος προόδου):** Ο ProgressDialog παρουσιάζει την πρόοδο μιας ενέργειας. Για παράδειγμα όσο συγχρονίζει ο email client της κινητής μας συσκευής για να μην υπάρχει κενή οθόνη εμφανίζεται ένας ProgressDialog.



Ακυρώνουμε έναν διάλογο με τον πιο φυσικό τρόπο στο Android δηλαδή με το back button. Αυτό το επιτυγχάνουμε ορίζοντας τον διάλογο μας cancellable(ακυρώσιμο).[20]

## Ειδοποιήσεις (Notifications)

Σε κάποιες περιπτώσεις θέλουμε να ενημερώσουμε το χρήστη για ένα αποτέλεσμα σχετικό με την εφαρμογή μας. Μερικά από αυτά τα αποτελέσματα απαιτούν κάποια απάντηση από το χρήστη και κάποια άλλα όχι. Δύο είναι τα είδη των μηνυμάτων που χρησιμοποιούμε:

- **To toast notification:** Είναι μία ειδοποίηση που παρουσιάζεται στην τρέχουσα οθόνη του χρήστη (για λίγα δευτερόλεπτα ανάλογα με την παραμετροποίηση).



- **To status bar notification:** Είναι ένα μήνυμα το οποίο παρουσιάζεται στην status bar της έξυπνης συσκευής. Το μήνυμα είναι προσβάσιμο είτε από το κεντρικό μενού είτε από οποιαδήποτε εφαρμογή.



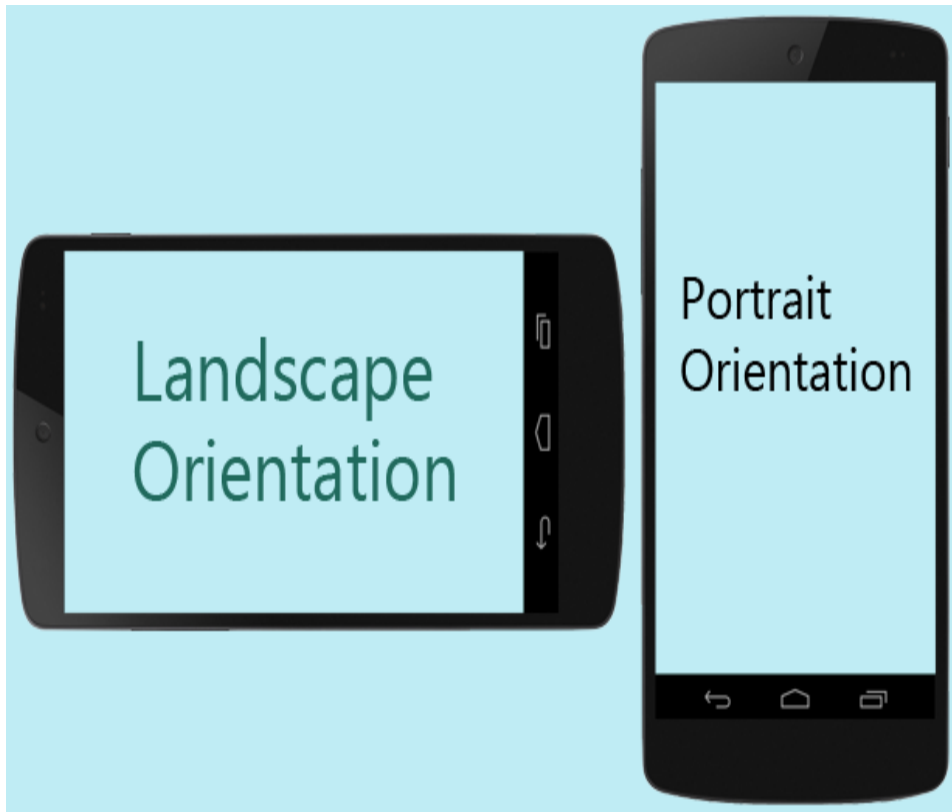
Τις περισσότερες φορές οι toast notification ενεργοποιούνται από activities ενώ οι status bar notification από services.

## Στοιχεία Εφαρμογής (Application Resources)

Τα Application Resources είναι συνήθως εικόνες, κείμενα κ.α. τα οποία χρησιμοποιούνται κατά την εκτέλεση της εφαρμογής. Για λόγους συμβατότητας είναι καλή πρακτική αυτά τα στοιχεία να ορίζονται σε διαφορετικά σημεία από τις κλάσεις της εφαρμογής. Οπότε έτσι είναι πιο εύκολο η γλώσσα να υποστηρίζει πολλές γλώσσες ή διαφορετικά μεγέθη οθόνης.

## Προσανατολισμός (Mode)

Οι κινητές συσκευές Android λειτουργούν με δύο διαφορετικούς προσανατολισμούς. Είτε σε landscape mode (προσανατολισμός τοπίου), είτε με τη διάταξη portrait mode (προσανατολισμός πορτρέτου). Όταν το κινητό αλλάζει θέση αλλάζει αυτόματα και ο προσανατολισμός. Τα γραφικά στοιχεία προσαρμόζονται αυτόματα. Ο χρήστης έχει τη δυνατότητα απενεργοποίησης της δυνατότητας αλλαγής προσανατολισμού.



Το layout της εφαρμογής δηλώνεται σε αρχεία XML. Στον φάκελο `res/layout/` όπου τοποθετούνται όλα τα αρχεία XML τα οποία σχετίζονται με τη διάταξη αρχείων. Για τον καθορισμό της διάταξης των γραφικών στοιχείων δηλώνεται το κατάλληλο xml αρχείο με την εντολή `setContentView()`.

### **Μέγεθος οθόνης (Screen Size)**

Το Android μπορεί με δυναμικό τρόπο να διαχειρίζεται τα γραφικά στοιχεία ή τη διάταξη στοιχείων ανάλογα με το μέγεθος της οθόνης και τον προσανατολισμό της οθόνης.

```
<supports-screens android:resizeable=["true" | "false"]
    android:smallScreens=["true" | "false"]
    android:normalScreens=["true" | "false"]
    android:largeScreens=["true" | "false"]
    android:xlargeScreens=["true" | "false"]
    android:anyDensity=["true" | "false"]
    android:requiresSmallestWidthDp="integer"
    android:compatibleWidthLimitDp="integer"
    android:largestWidthLimitDp="integer"/>
```

## 2.4. Android Manifest

Το αρχείο AndroidManifest.xml είναι απαραίτητο για κάθε εφαρμογή. Στο αρχείο Android Manifest περιέχονται βασικές και κρίσιμες πληροφορίες σχετικά με την εκτέλεση και λειτουργία του κώδικα της εφαρμογής οι οποίες μεταβιβάζονται στο λειτουργικό σύστημα πριν εκτελεσθεί η εφαρμογή. Πιο αναλυτικά περιέχει τα εξής:

- Την ονομασία του πακέτου της Java εφαρμογής.
- Την έκδοση της εφαρμογής.
- Την έκδοση του λειτουργικού συστήματος Android που απαιτεί η εφαρμογή.
- Το όνομα της εφαρμογής καθώς και το εικονίδιο της.
- Τις άδειες που χρειάζονται για να εκτελεστούν ορισμένες λειτουργίες της εφαρμογής.
- Τα συστατικά στοιχεία (activities, services, content providers, broadcast receivers) της εφαρμογής.
- Τις εξωτερικές βιβλιοθήκες που χρησιμοποιεί η εφαρμογή μας.

Ο παρακάτω κώδικας δηλώνει την γενική δομή του Android Manifest καθώς και κάθε στοιχείο που μπορεί να περιέχει. [22]

```
<manifest>

    <uses-permission />
    <permission />
    <permission-tree />
    <permission-group />
    <instrumentation />
    <uses-sdk />
    <uses-configuration />
    <uses-feature />
    <supports-screens />
    <compatible-screens />
    <supports-gl-texture />

    <application>

        <activity>
            <intent-filter>
                <action />
                <category />
                <data />
            </intent-filter>
            <meta-data />
        </activity>

        <activity-alias>
            <intent-filter> . . . </intent-filter>
            <meta-data />
        </activity-alias>

        <service>
            <intent-filter> . . . </intent-filter>
            <meta-data/>
        </service>

        <receiver>
            <intent-filter> . . . </intent-filter>
            <meta-data />
        </receiver>

        <provider>
            <grant-uri-permission />
            <meta-data />
            <path-permission />
        </provider>

        <uses-library />

    </application>

</manifest>
```

Εικόνα 22 - Android Manifest

## ***Σύνοψη***

Στις παραπάνω ενότητες περιεγράφηκε το λειτουργικό σύστημα και η αρχιτεκτονική του (πυρήνας και κυριότερες βιβλιοθήκες). Επιπλέον, περιεγράφηκαν οι εκδόσεις του android από το 2007 έως σήμερα, τα πλεονεκτήματα και μειονεκτήματα κάθε έκδοσης και στο τέλος οι φάκελοι και τα αρχεία από τα οποία αποτελείται μία εφαρμογή android.

## Κεφάλαιο 3ο Πρακτικό

### 3.1. Android Studio

Το Android Studio είναι ένα Ολοκληρωμένο Περιβάλλον Ανάπτυξης (IDE). Χρησιμοποιείται αποκλειστικά για την συγγραφή κώδικα και ανάπτυξη εφαρμογών για την πλατφόρμα Android. Αρχικά παρουσιάστηκε στις 16 Μαΐου του 2013 κατά τη διάσκεψη Google I/O (Innovation in the Open) στο Σαν Φρανσίσκο της Καλιφόρνιας όπου ανακοινώθηκε από την Διευθύντρια Προϊόντων της Google (Product Manager) Κάθριν Τσου (Katherine Chou). Είναι διαθέσιμο στο κοινό εντελώς δωρεάν υπό την Άδεια Έκδοσης 2.0 της Apache (Apache License 2.0).

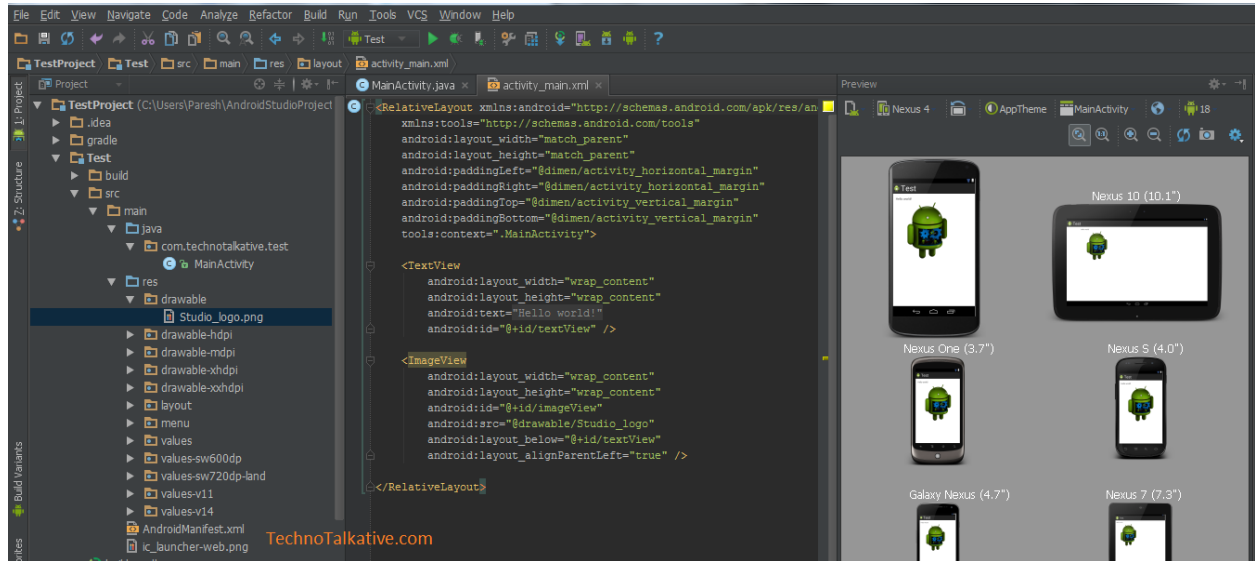
Το Android Studio ήταν σε αρχικό στάδιο προεπισκόπησης ξεκινώντας από την έκδοση 0.1 τον Μάιο του 2013. Στη συνέχεια δημοσιεύτηκε το δοκιμαστικό στάδιο (Beta) ξεκινώντας από την έκδοση 0.8, που κυκλοφόρησε τον Ιούνιο του 2014. Το πρώτο σταθερό build κυκλοφόρησε το Δεκέμβριο του 2014, ξεκινώντας από την έκδοση 1.0. Η τρέχουσα έκδοση του Android Studio είναι η 2.2.

Το Android Studio βασίστηκε στο λογισμικό JetBrains IntelliJ της IDEA και εξυπηρετεί την ανάπτυξη εφαρμογών σε Android. Είναι συμβατό με τα περισσότερα λειτουργικά συστήματα όπως Windows, Mac OS X και Linux. Κύριος σκοπός του είναι η αντικατάσταση των Εργαλείων Ανάπτυξης Android του Eclipse IDE (Eclipse Android Development Tools ADT).



Εικόνα 23 - Λογότυπο του Android Studio IDE





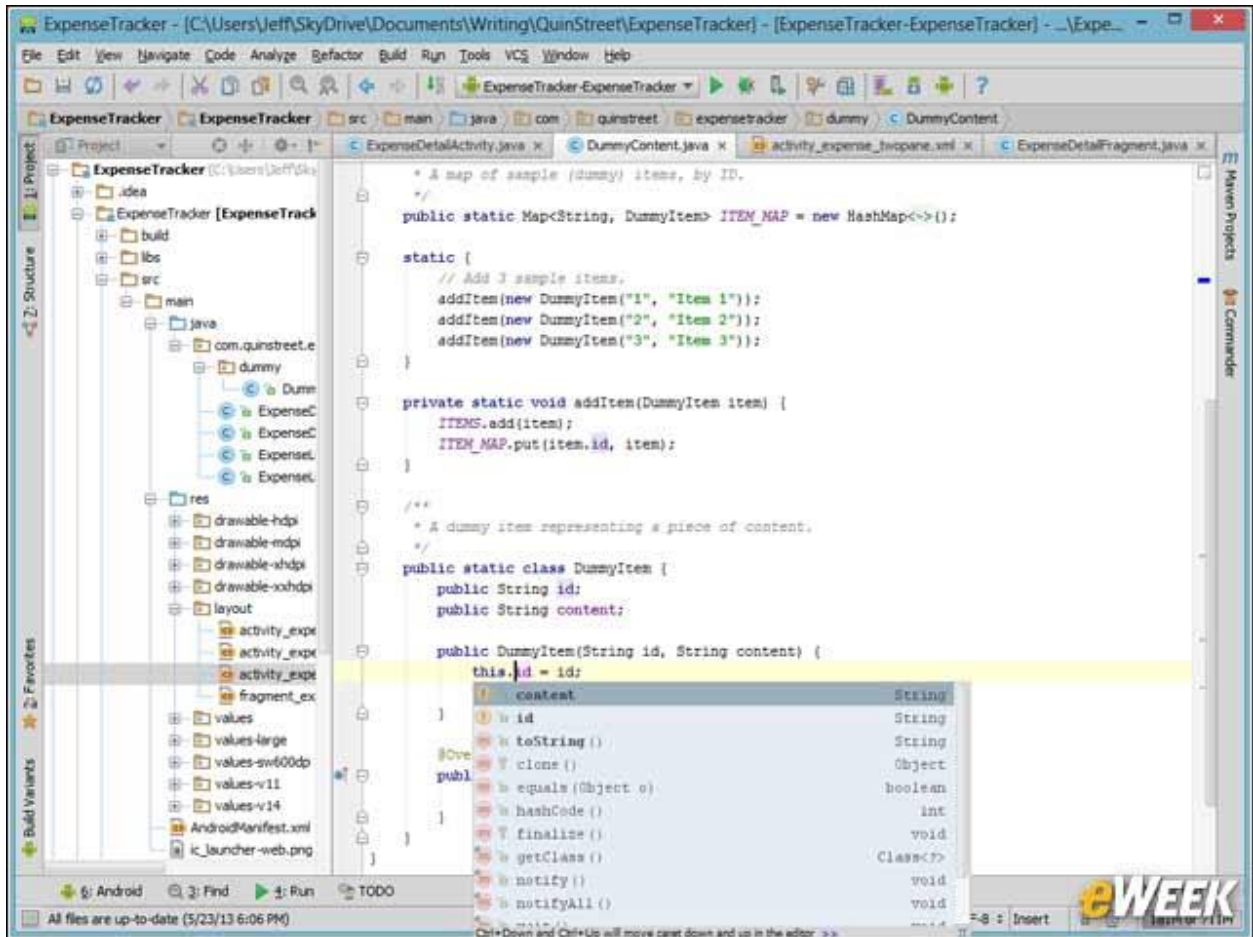
Εικόνα 24 -Android Studio IDE

[23]

### 3.1.1 Χαρακτηριστικά του Android Studio IDE

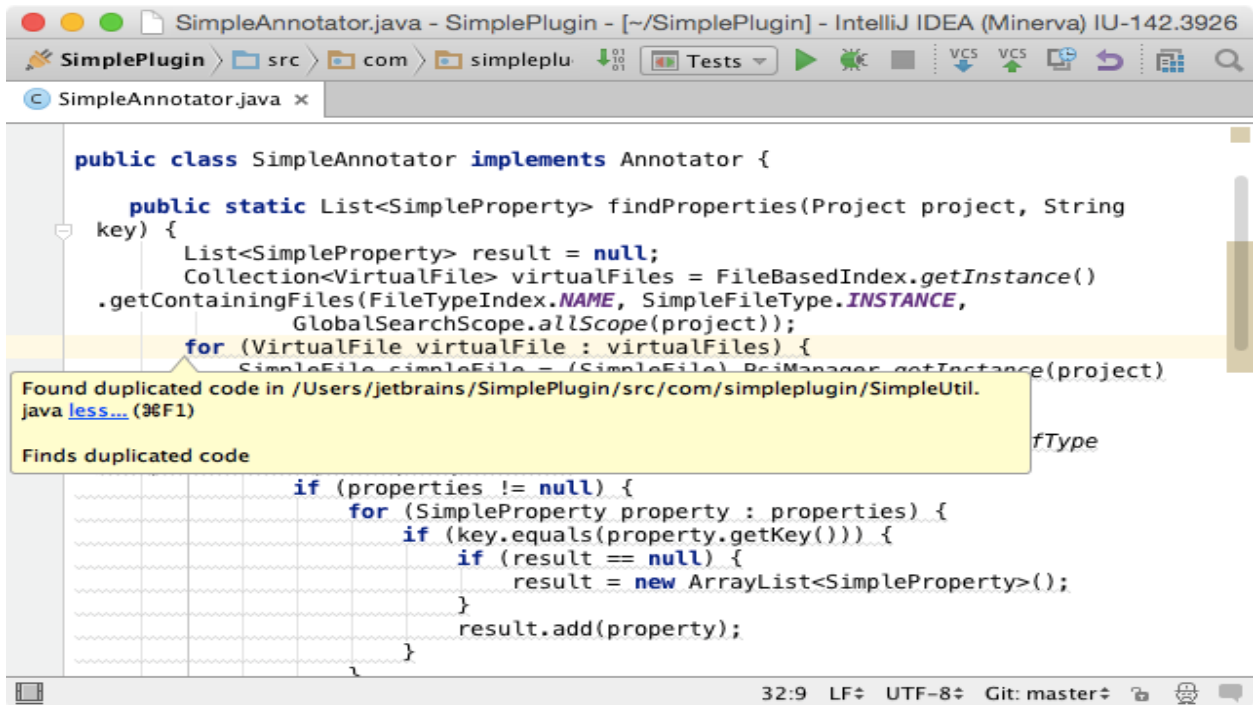
Ο πυρήνας του Android Studio είναι το IntelliJ IDEA οπότε τα κύρια χαρακτηριστικά του προέρχονται από αυτό το λογισμικό. Κάποια από τα πιο σημαντικά χαρακτηριστικά του περιγράφονται παρακάτω :

- **Ισχυρή Επεξεργασία Κώδικα (Powerful Code Editing)** η οποία χαρακτηρίζεται από έξυπνο σύστημα αυτοσυμπλήρωσης κώδικα (autocomplete), ριζική πλοήγηση μεταξύ των αρχείων του έργου (Profound navigation between project files), προχωρημένη και ασφαλή επεξεργασία (π.χ. μετονομασία ενός αρχείου και εφαρμογή του νέου ονόματος σε όλα τα μέρη του κώδικα μας που αναφέρεται αυτόματα) και προεπισκόπηση πόρων. Στην παρακάτω εικόνα παρουσιάζεται παράδειγμα αυτοσυμπλήρωσης κώδικα. [24]



Εικόνα 25 - Παράδειγμα έξυπνης αυτοσυμπλήρωσης κώδικα

- **Ανάλυση κώδικα κατά την επεξεργασία (On-the-fly Code Analysis).** Ο ενσωματωμένος συντάκτης εμφανίζει προειδοποιήσεις και σφάλματα στον κώδικα κατά την πληκτρολόγηση και δίνει τη δυνατότητα της εφαρμογής μιας αυτόματης γρήγορης λύσης. [24]



Εικόνα 26 - Παράδειγμα ανάλυσης κώδικα κατά την επεξεργασία

- **Ενσωματωμένα εργαλεία του Android (Built-in Android Tools)** τα οποία αποτελούνται από:

ο Ισχυρό σχεδιαστή διεπαφής χρήστη με drag-n-drop ο οποίος υποστηρίζει διατάξεις (layouts) διαφορετικών διατάξεων και μεγεθών οθόνης.

ο Το εργαλείο LogCat είναι ένα σύστημα καταγραφής με επιλογή αναζήτησης και εφαρμογής διάφορων φίλτρων εμφάνισης.

ο Το εργαλείο αποσφαλμάτωσης Dalvik Debug Monitor Server (DDMS).

ο Η λειτουργία θέασης ιεραρχίας (Hierarchy View) η οποία δίνει τη παρουσιάζει σε εύπεπτη μορφή την ιεραρχία των στοιχείων της διεπαφής χρήστη.

ο Δίνει τη δυνατότητα εκτέλεσης και αποσφαλμάτωσης για εξομοιώσεις κινητών συσκευών καθώς και πραγματικών συσκευών.

ο Με το εργαλείο Draw 9 δίνεται η δυνατότητα επεξεργασίας-δημιουργίας εικόνων μορφής Bitmap και της αυτόματης προσαρμογής τους σε κάθε μεγέθους οθόνη που τις φιλοξενούν. [24]

### 3.1.2 Άλλα χαρακτηριστικά

- Το ενοποιημένο σύστημα Proguard που δίνει την δυνατότητα να συρρικνωθεί, να βελτιστοποιηθεί, και να συσκοτιστεί ο κώδικας μας με την αφαίρεση αχρησιμοποίητου κώδικα και την μετονομασία κατηγοριών, πεδίων και μεθόδων με σημασιολογικά σκοτεινές ονομασίες. Το αποτέλεσμα είναι ένα μικρότερου μεγέθους αρχείο .apk που είναι πιο δύσκολο να αντιστραφεί μηχανικά (Reverse Engineered) οπότε και καλύπτεται η ευπάθεια υποκλοπής του κώδικα.
- Δυνατότητα ψηφιακής υπογραφής της παραχθείσας εφαρμογής με πιστοποιητικό (Certificate). Το Android απαιτεί ότι όλες οι εφαρμογές είναι ψηφιακά υπογεγραμμένες με ένα πιστοποιητικό για να μπορέσουν να εγκατασταθούν. Το Android χρησιμοποιεί αυτό το πιστοποιητικό για την ταυτοποίηση του συγγραφέα/προγραμματιστή ενός app, χωρίς να χρειάζεται να υπογραφτεί από κάποια αρχή έκδοσης πιστοποιητικών. Οι Android εφαρμογές χρησιμοποιούν συχνά αυτό-υπογεγραμμένα πιστοποιητικά όπου ο προγραμματιστής της εφαρμογής κατέχει το ιδιωτικό κλειδί του πιστοποιητικού.
- Η ενσωματωμένη υποστήριξη για την πλατφόρμα της Google Cloud, καθιστώντας εύκολη την ενσωμάτωση του Google Cloud Messaging και του App Engine.
- Τα εργαλεία **Lint (Lint tools)** τα οποία δίνουν την δυνατότητα καταγραφής της απόδοσης, της χρηστικότητας, της συμβατότητας εκδόσεων του Android, και άλλων προβλημάτων όπως καταγραφή μερικών ύποπτων και μη δομικών τμημάτων του πηγαίου κώδικα που μπορούν να προκαλέσουν σφάλματα.
- **Πρότυπα κώδικα (Code templates)** τα οποία προορίζονται για την ανάπτυξη εφαρμογών με κοινά χαρακτηριστικά.

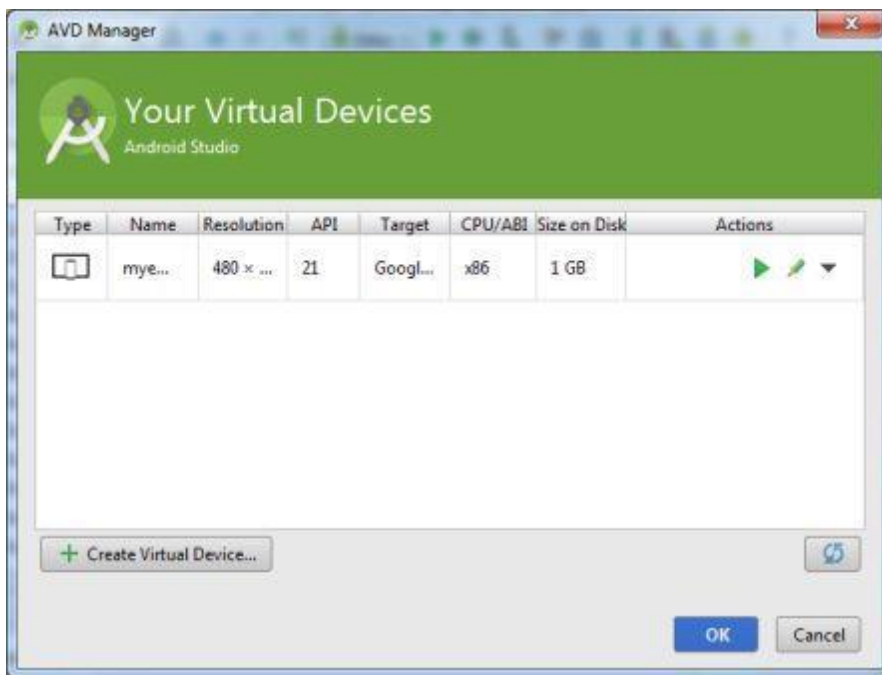
[24]

## 3.2 Εργαλεία ανάπτυξης λογισμικού Android (Android Software Development Kit SDK)

Παρακάτω θα δούμε τα εργαλεία ανάπτυξης λογισμικού που χρησιμοποιούνται πιο συχνά, θα τα ταξινομήσουμε με βάση την κατηγορία τους και θα επεκταθούμε σε μερικά από αυτά σε επόμενα κεφάλαια.

### 3.2.1 Εργαλεία Εικονικών Συσκευών (Android Virtual Device Tools AVD)

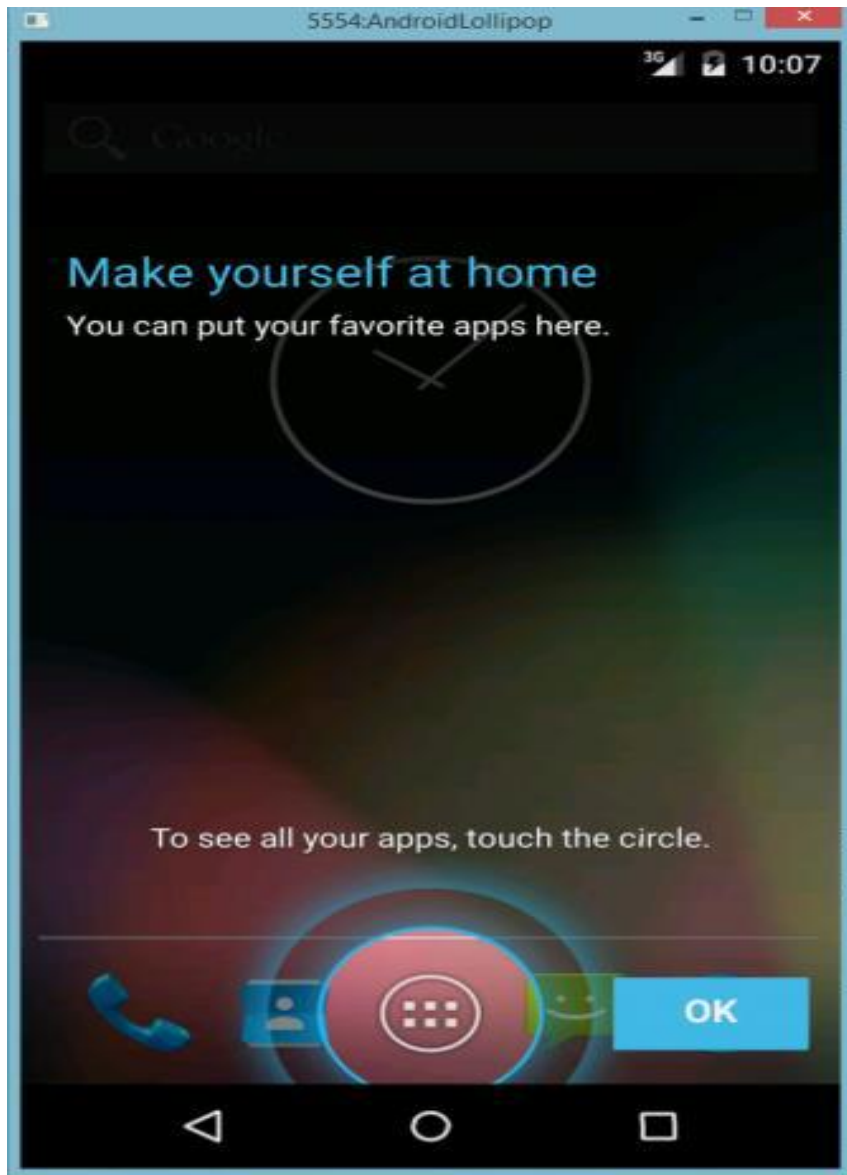
- Διαχειριστής Εικονικών Συσκευών (Android Virtual Device Manager).



Εικόνα 27 - Διαχειριστής εικονικών συσκευών

[25]

- Εξομοιωτής των Android συσκευών (Android Emulator).



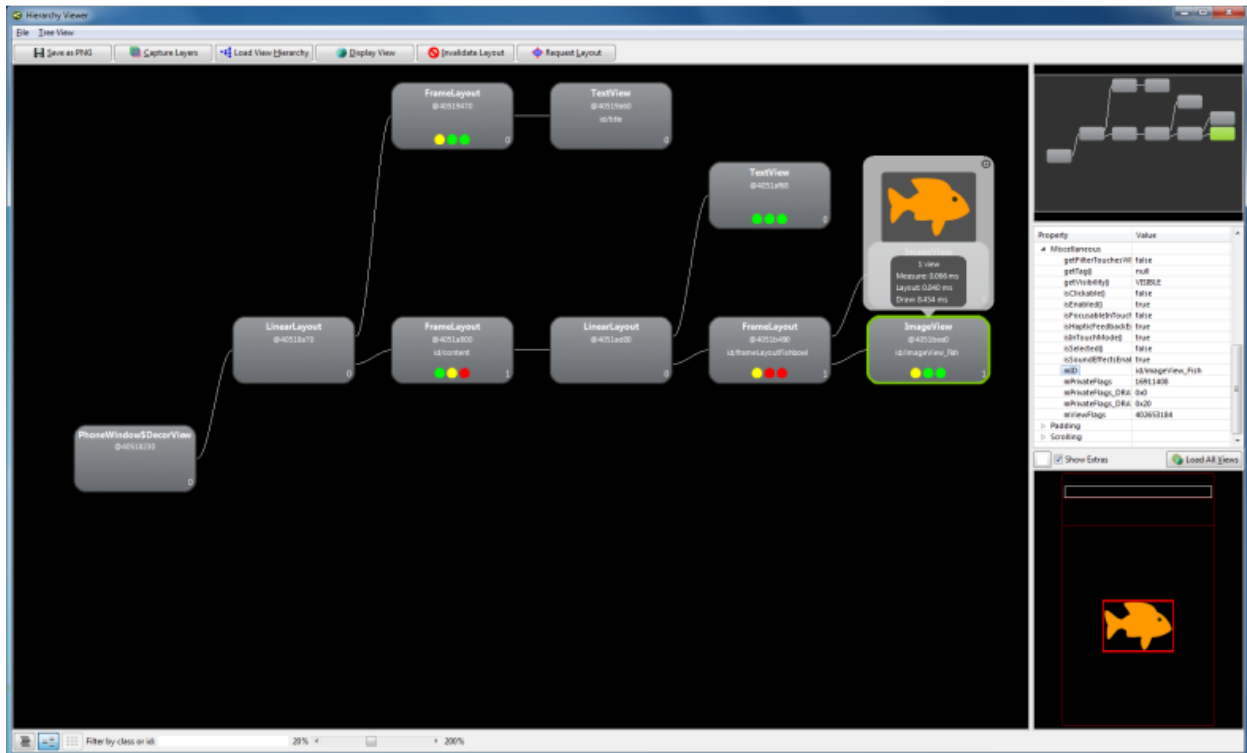
Εικόνα 28 - Εξομοιωτής Android συσκευών

[26]

- Εργαλείο δημιουργίας εξωτερικής μνήμης για τις εικονικές συσκευές Android (mkcard) πχ. Εξωτερική κάρτα μνήμης (SD card).

### 3.2.2 Εργαλεία ανάπτυξης (Development Tools)

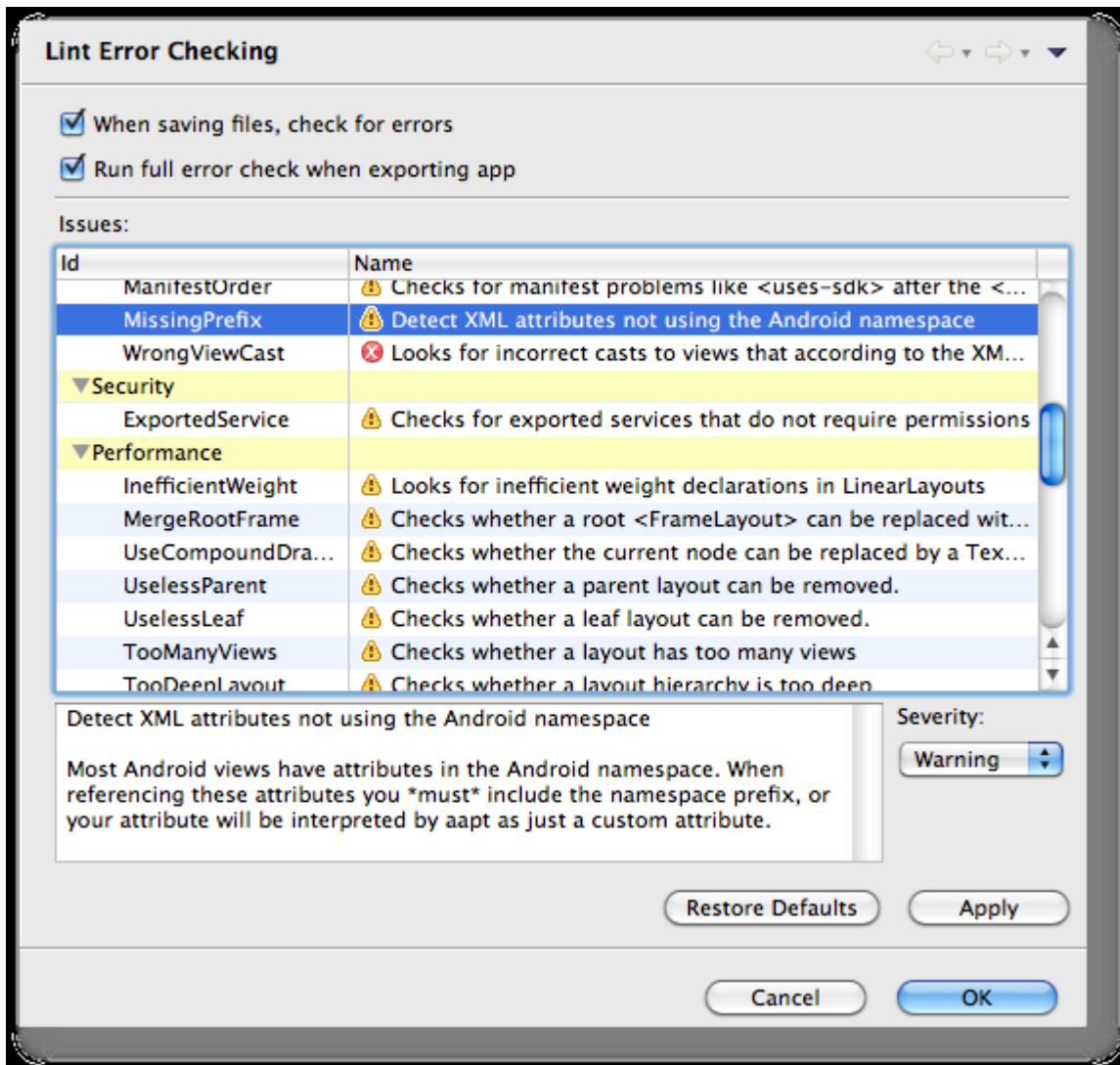
- Λειτουργία θέασης ιεραρχίας (Hierarchy View) που αναφέρθηκε στα παραπάνω χαρακτηριστικά παρουσιάζεται στην παρακάτω εικόνα.



Εικόνα 29 - Παράδειγμα Θέασης ιεραρχίας

[27]

- Τα εργαλεία Lint (Lint tools) που και αυτά ανήκουν στα εργαλεία ανάπτυξης παρουσιάζονται στην εικόνα παρακάτω



Εικόνα 30 - Παράδειγμα ελέγχου σφαλμάτων με τη χρήση του εργαλείου Lint

[28]

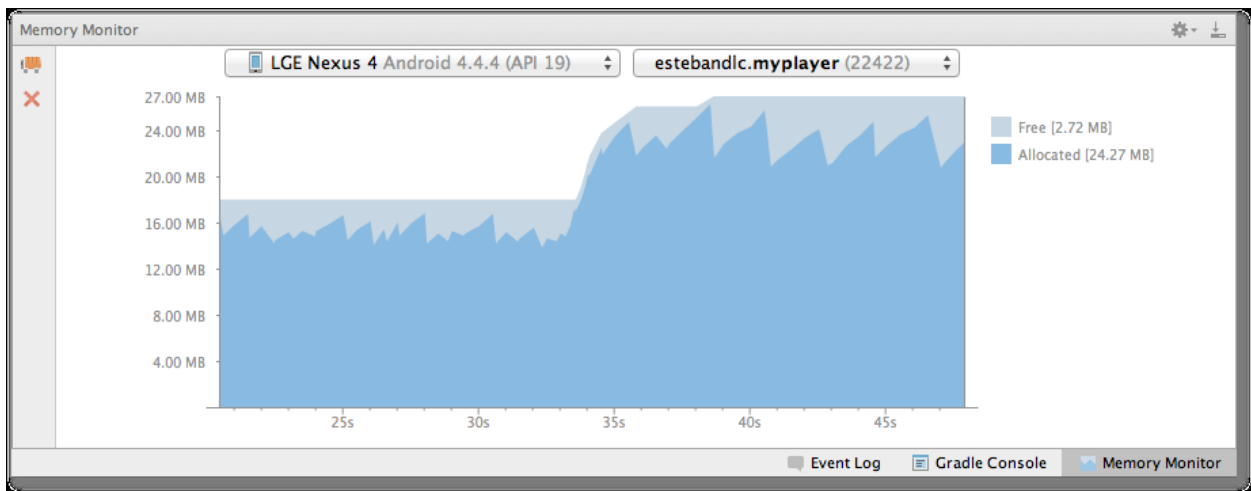
- Ο Διαχειριστής Εργαλείων Ανάπτυξης Εφαρμογών (SDK Manager).
- Το sqlite3 το οποίο μας δίνει τη δυνατότητα της διαχείρισης βάσεως δεδομένων (Databases) που δημιουργήθηκαν από εφαρμογές του Android μέσω ενός απομακρυσμένου κελύφους ή μέσω του υπολογιστή σας. Κάποιες από τις πιο κοινές εντολές, είναι οι .dump η οποία χρησιμοποιείται για την εμφάνιση των περιεχομένων ενός πίνακα και η .schema για την εξαγωγή της SQL δήλωσης CREATE (SQL CREATE statement) για έναν υπάρχοντα πίνακα. Το εργαλείο αυτό σας δίνει επίσης τη δυνατότητα να εκτελέσετε τις εντολές SQLite κατά την επεξεργασία.



### 3.2.3 Εργαλεία αποσφαλμάτωσης (Debugging Tools)

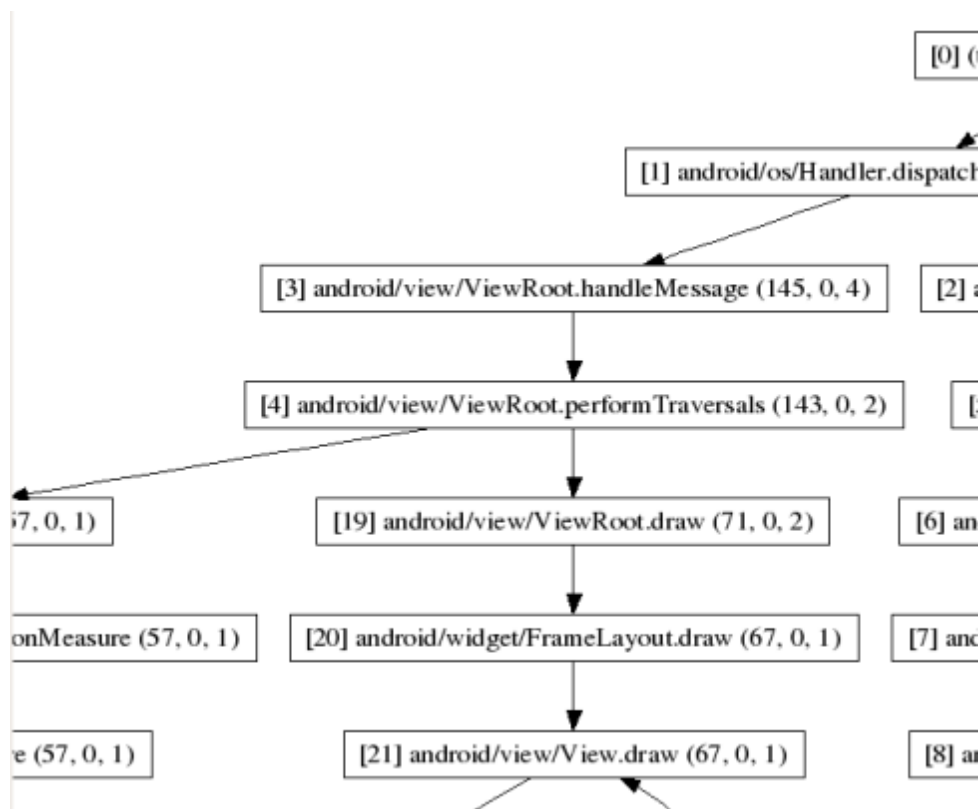
Παρακάτω παρουσιάζονται τα βασικά εργαλεία αποσφαλμάτωσης

- **Android Debug Bridge ADB.** Το ADB είναι ένα ευέλικτο εργαλείο γραμμής εντολών που δίνει τη δυνατότητα να επικοινωνείτε με έναν ενεργό εξομοιωτή ή μια συνδεδεμένη συσκευή Android. Επιπλέον δίνει τη δυνατότητα πρόσβασης στο φλοιό της συσκευής (Device Shell). Προσφέρει επιπλέον μια σειρά από εντολές για την αποσφαλμάτωση.
- Εργαλείο αποσφαλμάτωσης **Dalvik Debug Monitor Server (DDMS).**
- Εργαλείο παρακολούθησης συσκευής (**Android Device Monitor**) με δυνατότητες αποσφαλμάτωσης και ανάλυσης χρήσης πόρων του συστήματος (κύριας μνήμης, κεντρικής μονάδας επεξεργασίας ΚΜΕ κ.α.).



Εικόνα 31 - Παράδειγμα ανάλυσης μνήμης με το εργαλείο παρακολούθησης συσκευής

- Το **Dmtracedump (Trace Dump Diagram)** το οποίο αναπτύσσει γραφικά διαγράμματα της στοίβας κλήσεων (Call Stack) από τα αρχεία καταγραφής (Log Files) της παρακολούθησης. Το εργαλείο αυτό χρησιμοποιεί το βοηθητικό λογισμικό Graphviz Dot για να δημιουργήσει την γραφική έξοδο.



Εικόνα 32 - Παράδειγμα διαγράμματος της στοίβας κλήσεων

- Το **Traceview** είναι εργαλείο με αρκετές ομοιότητες με το Dmtracedump.
- **SysTrace (System Trace)** το οποίο δίνει τη δυνατότητα της ανάλυσης της απόδοσης της εφαρμογής, με την καταγραφή του χρόνου εκτέλεσης των διεργασιών των εφαρμογών και άλλων διεργασιών του Λειτουργικού συστήματος. Το εργαλείο βασίζεται σε δεδομένα από τον πυρήνα του Android, όπως το χρονοδιάγραμμα της CPU, την δραστηριότητα του δίσκου, και τα νήματα της εφαρμογής (Application Threads) για να δημιουργήσει μια HTML έκθεση που παρουσιάζει μια συνολική εικόνα των διεργασιών του συστήματος της συσκευής Android για μια δεδομένη χρονική περίοδο.

[29]

### 3.2.4 Εργαλεία Κατασκευής (Build Tools)

Σε αυτήν την ενότητα παρουσιάζονται τα κυριότερα εργαλεία κατασκευής.

- Το **Java Opaque Binary Blob (JOBB)** δίνει τη δυνατότητα δημιουργίας κρυπτογραφημένων και μη κρυπτογραφημένων αρχείων επέκτασης .apk σε αδιαφανείς δυαδική

μορφή (Opaque Binary Blob, \*.obb). Τα αρχεία που προκύπτουν συνήθως περιέχουν έτοιμα προς χρήση στοιχεία του αρχείου .apk (όπως γραφικά, ήχους και βίντεο).

- Το **ProGuard** που παρουσιάστηκε στην παραπάνω ενότητα.
- Το **Zipalign** (.zip align) το οποίο έχει ως κύρια λειτουργία την βελτιστοποίηση των .apk αρχείων με την προϋπόθεση ότι όλα τα ασυμπιεστα δεδομένα ξεκινούν με μια ιδιαίτερη ευθυγράμμιση σχετική με την αρχή του αρχείου. Αυτό θα πρέπει πάντα να χρησιμοποιείται για την ευθυγράμμιση των .apk αρχείων αφού έχουν υπογραφεί.

### Εργαλεία Επεξεργασίας Εικόνων (Image Tools)

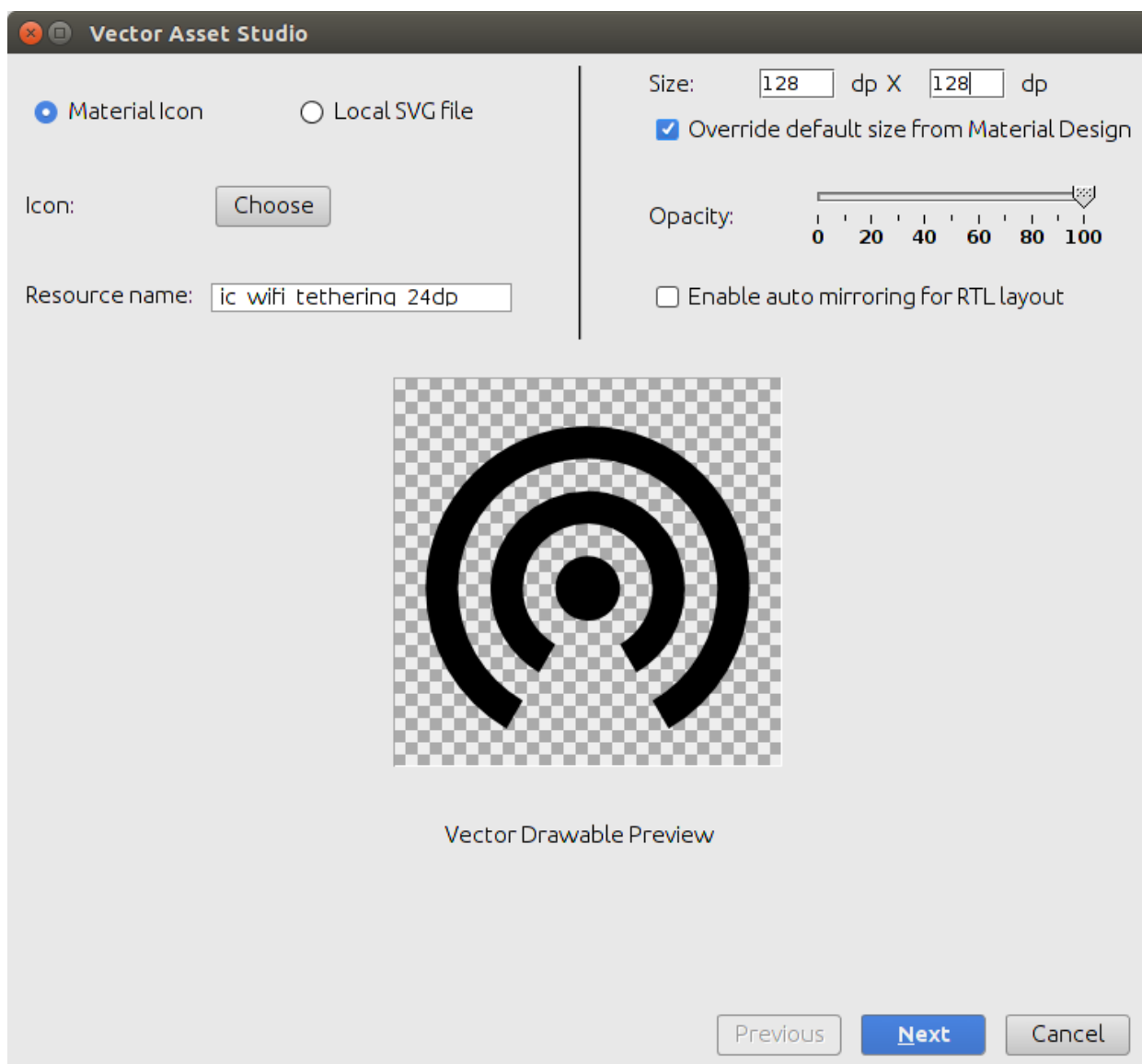
- Το Draw 9 που ανήκει στα εργαλεία επεξεργασίας εικόνων παρουσιάζεται στην παρακάτω εικόνα.



Εικόνα 33 - Παράδειγμα χρήσης του Draw 9

- **Vector Asset Studio.**

Το Vector Asset Studio επιτρέπει την δημιουργία και προσθήκη Material Icons και την εισαγωγή αρχείων τύπου Scalable Vector Graphic (SVG) ως όλκιμους πόρους. Σε σχέση με τις εικόνες raster, διανυσματικά drawables χαρακτηρίζονται για το μικρό τους μέγεθος και τη δυνατότητα αλλαγής μεγέθους της εικόνας χωρίς απώλεια. Θα σας βοηθήσει να υποστηρίξετε πιο εύκολα διαφορετικές συσκευές Android με διαφορετικά μεγέθη οθόνης και ανάλυσης, επειδή μπορείτε να εμφανίσετε ένα εικονίδιο κοινού φορέα σε όλα αυτά.



Εικόνα 34 - Παράδειγμα χρήσης του Vector Asset Studio

- **etc1tool (Etc1 Compression Tool).**

Το etc1tool είναι ένα βοηθητικό πρόγραμμα γραμμής εντολών που σας επιτρέπει να κωδικοποιήσετε PNG εικόνες με το πρότυπο συμπίεσης ETC1 και να αποκωδικοποιήσει ETC1 συμπιεσμένες εικόνες πίσω σε PNG.

- **Tracer for OpenGL ES.**

Το εργαλείο σας επιτρέπει να συλλάβετε τις εντολές OpenGL ES (Embedded Systems) και καρέ-καρέ τις εικόνες για να σας βοηθήσει να καταλάβετε πώς εκτελούνται οι εντολές γραφικών σας.

[30]

### 3.2.5 Εργαλεία Πλατφόρμας (Platform Tools)

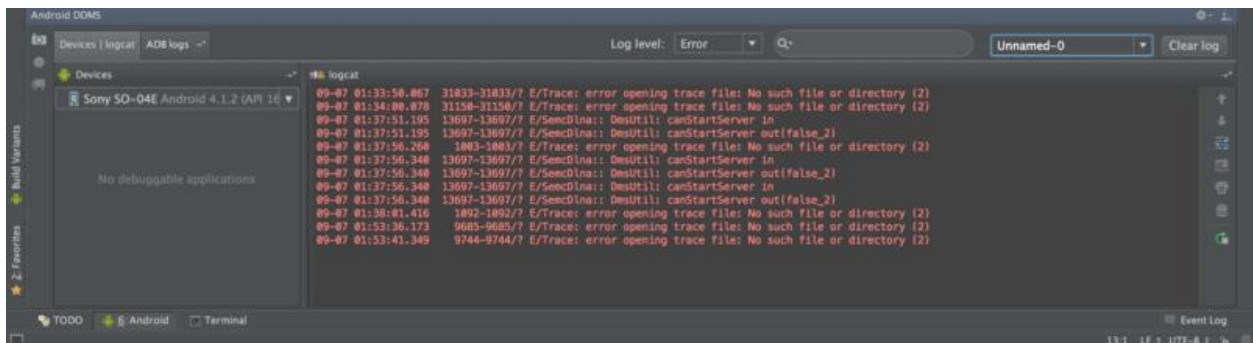
- **Android Debug Bridge (adb).**

Το adb είναι ένα ευέλικτο εργαλείο γραμμής εντολών που σας επιτρέπει να επικοινωνείτε με ένα στιγμιότυπο εξομοιωτή ή μια συνδεδεμένη συσκευή Android καθώς και να ελέγχετε την κατάσταση τους. Μέσω αυτού μπορείτε κάλλιστα να εγκαταστήσετε μια εφαρμογή Android (.apk).

- **Bmgr (Back Up Manager).**

Το bmgr είναι ένα εργαλείο φλοιού το οποίο επιτρέπει την λειτουργία και αλληλεπίδραση με τον Διαχειριστή αντιγράφων ασφαλείας (Back-up Manager) σε συσκευές Android που υποστηρίζουν το API Level 8 (FROYO) ή μεγαλύτερο.

- **LogCat** όπως προαναφέραμε παραπάνω.



Εικόνα 35 - Παράδειγμα χρήσης του logcat

[31]

## 3.3 Εφαρμογή

### 3.3.1 Απαιτήσεις εφαρμογής

Στην διπλωματική μου εργασία ανέπτυξα εφαρμογή η οποία έχει τα παρακάτω χαρακτηριστικά

- Υπολογίζει την ταχύτητα μεταφόρτωσης(download) αρχείου.
- Υπολογίζει την ταχύτητα μεταφόρτωσης(upload) αρχείου.
- Υπολογίζει την καθυστέρηση του δικτύου (network latency).
- Βρίσκει εάν το κινητό με λειτουργικό σύστημα Android είναι συνδεδεμένο σε ασύρματο δίκτυο. (wifi network)
- Βρίσκει εάν το κινητό με λειτουργικό σύστημα Android είναι συνδεδεμένο με πάροχο κινητής τηλεφωνίας.
- Βρίσκει την ισχύ του σήματος της κινητής τηλεφωνίας, το id του κελιού της κινητής τηλεφωνίας και την τοποθεσία του (cell location)

### 3.3.2 Κώδικας εφαρμογής

Σε αυτήν την ενότητα θα περιγραφούν τα σημαντικότερα κομμάτια κώδικα

#### 'Έλεγχος εάν υπάρχει wifi

Ο έλεγχος πραγματοποιείται με τον παρακάτω κώδικα

```
ConnectivityManager connMgr = (ConnectivityManager)
context.getSystemService(Context.CONNECTIVITY_SERVICE);
NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();
return (networkInfo != null && networkInfo.isConnected());
```

[32]

Η κλάση ConnectionManager διακρίνεται για τα παρακάτω χαρακτηριστικά

- Καταγράφει & παρακολουθεί τις συνδέσεις δικτύου (Wi-Fi, GPRS, UMTS, κ.α.)
- Καταγράφει τις αλλαγές στις ιδιότητες της δικτυακής συνδεσιμότητας.
- Δυνατότητα σύνδεσης σε διαθέσιμο δίκτυο ανάλογα την σειρά προτεραιότητας η οποία έχει οριστεί.

- Παρέχει ένα σετ εντολών για την σύνδεση και την αναγνώριση ενεργών συνδέσεων.

[33]

### Έλεγχος εάν υπάρχει σύνδεση με πάροχο κινητής

Ο έλεγχος είναι παρόμοιος με την υπάρξει wifi σύνδεσης

### Εύρεση ισχύς σήματος (signal strength)

Η έρευνα της ισχύς του σήματος πραγματοποιείται με τον παρακάτω κώδικα

```
//get signal strenght  
psListener = new myPhoneStateListener();  
tm.listen(psListener, PhoneStateListener.LISTEN_SIGNAL_STRENGTHS);
```

Η ισχύς του σήματος ορίζεται σε dbm και για την καταγραφή της είναι απαραίτητη η ανάπτυξη ενός PhoneStateListener (βλ. παράρτημα). Στην εφαρμογή μας χρησιμοποιούμε τον παραπάνω Listener μόνο για την καταγραφή των μεταβολών της αλλαγής της ισχύς του σήματος. Με τον ίδιο PhoneStateListener υπάρχει επιπλέον η δυνατότητα της εύρεσης του τύπου του δικτύου που είναι συνδεδεμένη η κινητή συσκευή.

### Εύρεση του throughput (ρυθμαπόδοσης)

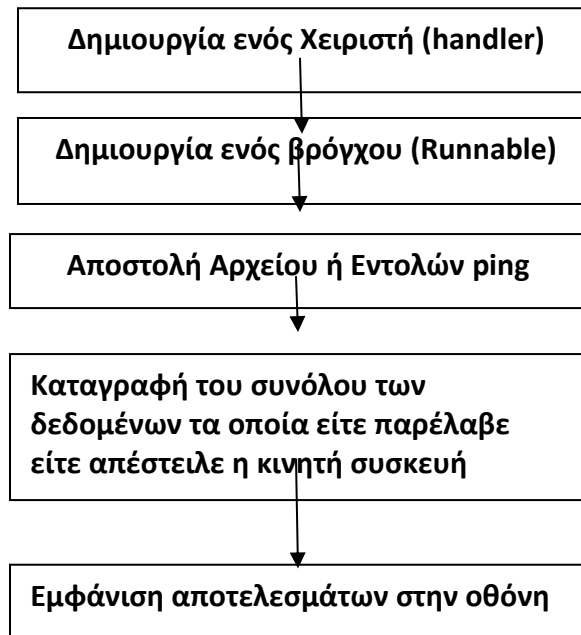
Ο τύπος υπολογισμού του throughput είναι windows size / RTT. Το μέγεθος σε bits του παραθύρου είναι 65536\*1024 το μέγιστο. Τίθενται όμως κάποιοι περιορισμοί για αποφυγή προβλημάτων latency ορίζεται το μέγεθος του παραθύρου στα 524288 bits. Για την εύρεση του RTT χρησιμοποιείται η εντολή ping. Από το αποτέλεσμα της προκύπτει το RTT. Στη συνέχεια γίνεται ο υπολογισμός της ρυθμαπόδοσης. Για την εντολή ping χρησιμοποιείται ο παρακάτω κώδικας

```
Runtime runtime = Runtime.getRuntime();  
Process proc = runtime.exec("ping -c 1 " + host);  
proc.waitFor();  
int exit = proc.exitValue();  
return exit;
```

### **Εύρεση ταχύτητας μεταφόρτωσης (Downlink/Uplink)**

Η εύρεση της ταχύτητας μεταφόρτωσης πραγματοποιείται με δύο τρόπους. Ο ένας είναι με την κλάση androidstats και ο άλλος είναι με την μεταφόρτωση αρχείο συγκεκριμένου μεγέθους από διαδικτυακό τόπο.

Και οι δύο τρόποι βασίζονται στο παρακάτω διάγραμμα ροής

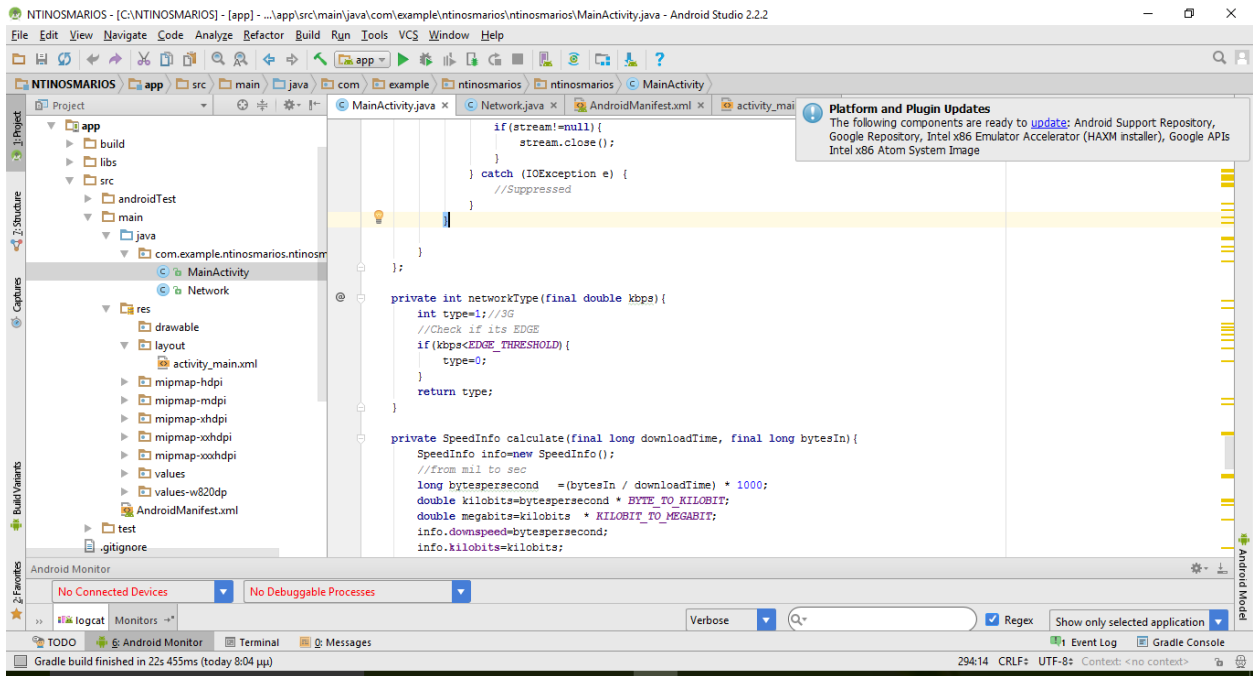


Αναλυτικά ο κώδικας της εφαρμογής βρίσκεται στο παράρτημα.

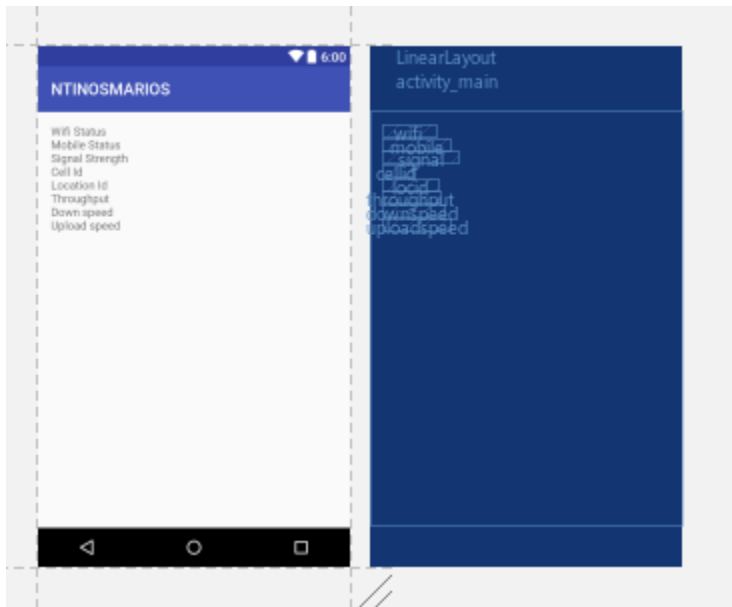
### **3.3.3 Οθόνη**

Σε αυτήν την ενότητα παρουσιάζονται οι βασικές οθόνες της εφαρμογής.

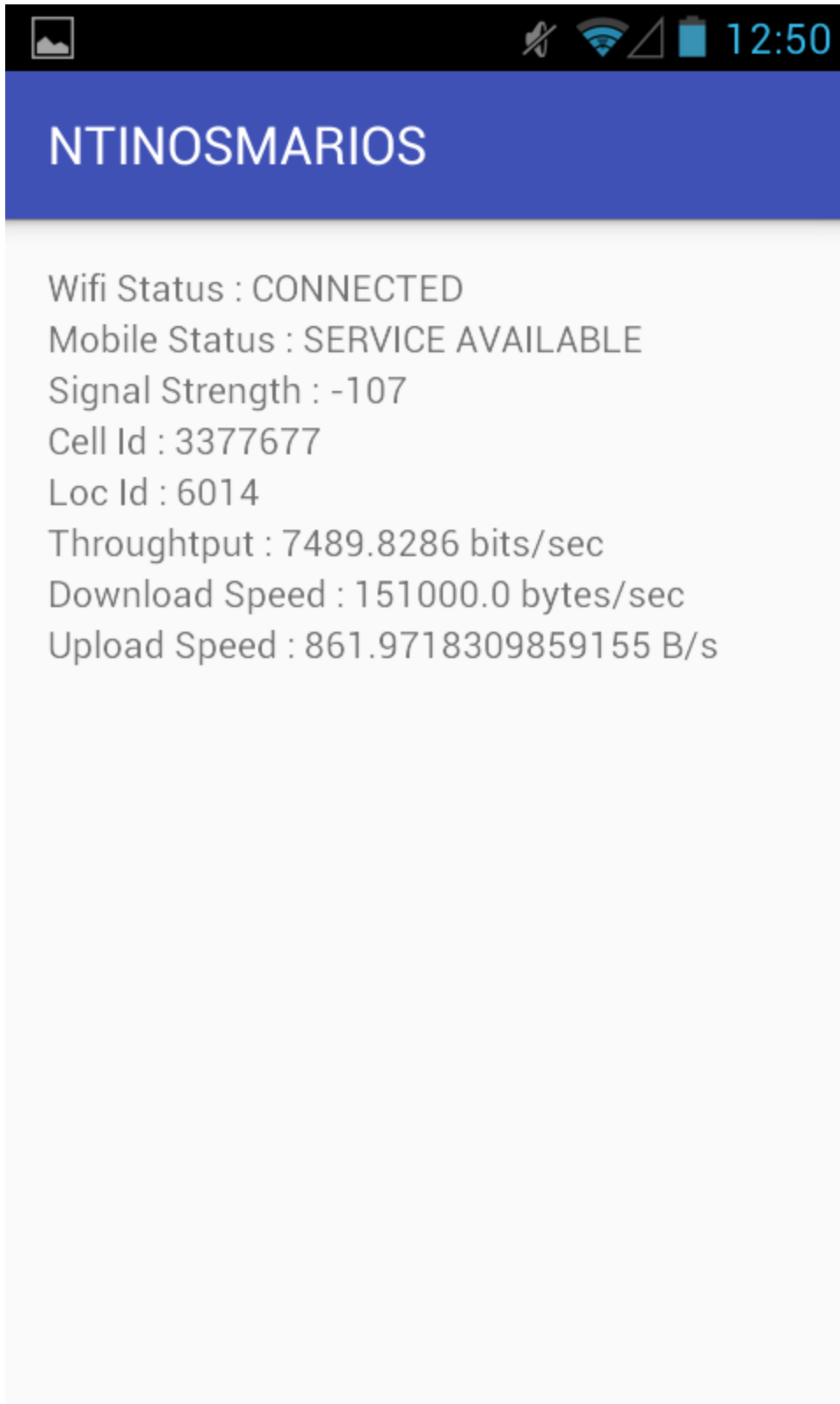




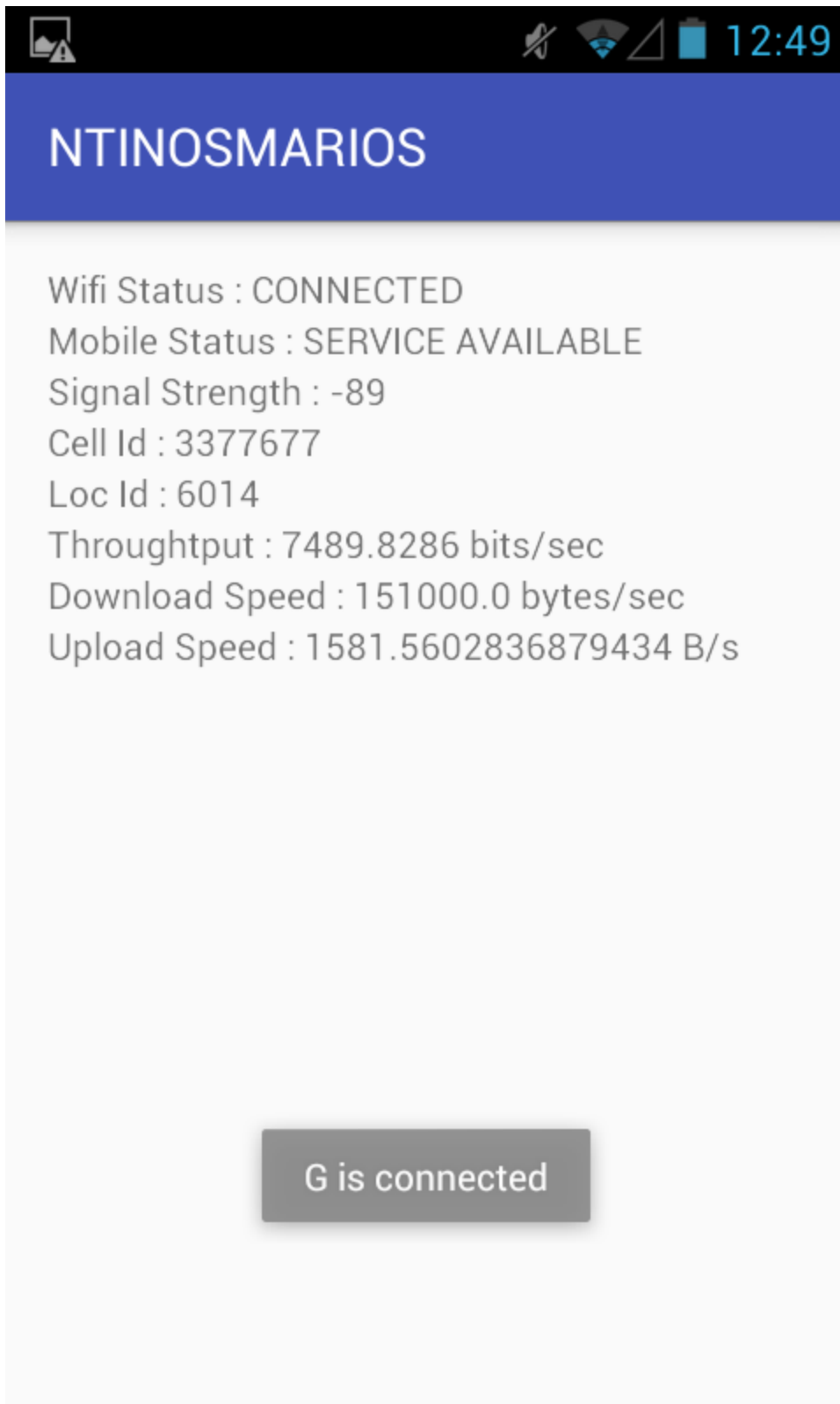
Εικόνα 36 - Θέση της εφαρμογής από το Android Studio(1/2)



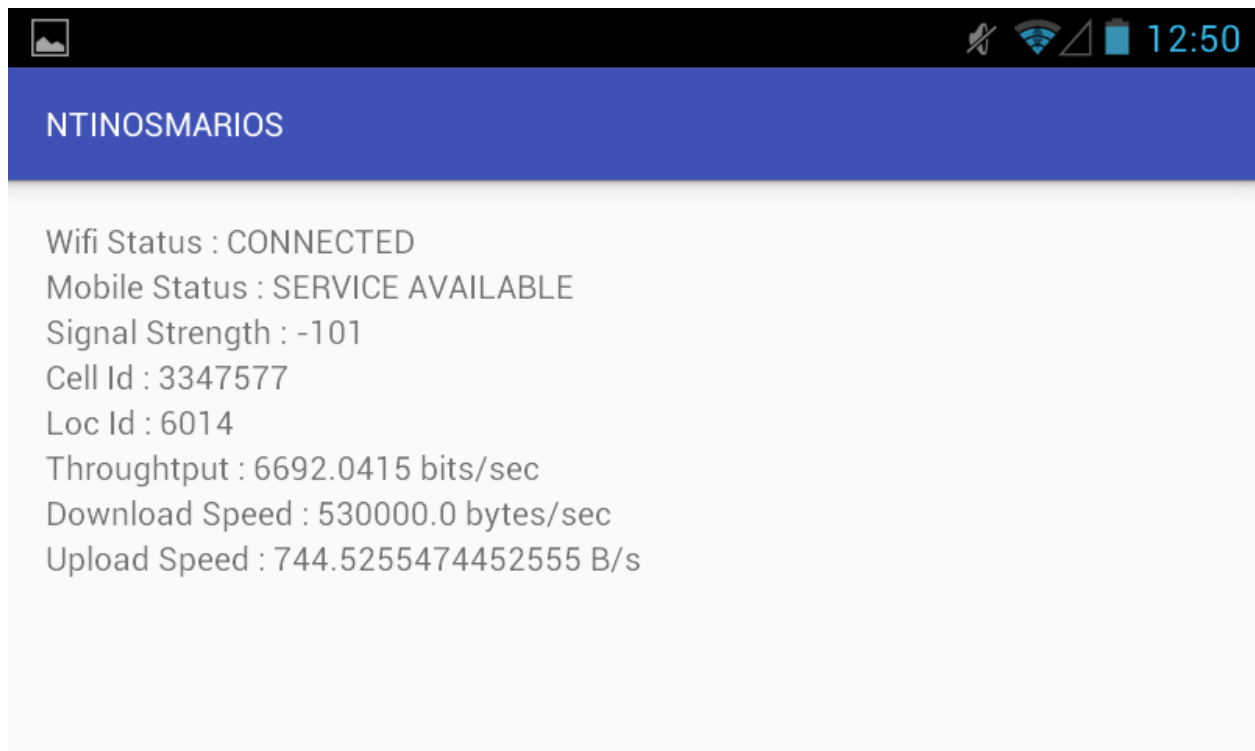
Εικόνα 37 - Θέση της εφαρμογής από το Android Studio(2/2)



Εικόνα 38 - Οθόνη της εφαρμογής από το κινητό τηλέφωνο android (1/2)



Εικόνα 39 - Οθόνη της εφαρμογής από το κινητό τηλέφωνο android (2/3)



Εικόνα 40 - Οθόνη της εφαρμογής από το κινητό τηλέφωνο android (3/3)

## Κεφάλαιο 4 Συμπεράσματα Μελλοντικές Βελτιώσεις

Η εφαρμογή η οποία παρουσιάστηκε στην παρούσα πτυχιακή εργασία είναι ολοκληρωμένη και λειτουργική. Η δομή της είναι απλή και μετράει μια σειρά από χαρακτηριστικά τα οποία δείχνουν την ποιότητά του δικτύου. Η δομή της είναι απλή και υπολογίζει κάποιες τιμές με διαφορετικούς τρόπου για πιο ακριβή αποτελέσματα.

Ολοκληρώνοντας την πτυχιακή εργασία και το “ταξίδι” στην ανάπτυξη Android εφαρμογών είχα τη δυνατότητα να εξοικειωθώ με τεχνολογίες ανάπτυξης εφαρμογών σε κινητές συσκευές. Πέρα από τις απαιτήσεις της πλατφόρμας και την εξερεύνηση των δυσκολιών της, η ενασχόληση με το αντικείμενο προσφέρει την κατανόηση της γλώσσας προγραμματισμού Java και γενικά τον λογισμό που χρειάζεται από την σχεδίαση έως την υλοποίηση του έργου. Επίσης, αποκτήθηκαν γνώσεις πάνω στο περιβάλλον ανάπτυξης λογισμικού Android Studio και στα εργαλεία που προσφέρει το Android SDK.

Για την βελτίωση της εργασίας θα πρότεινα κάποιες μελλοντικές αλλαγές

- Όλες οι τιμές να υπολογίζονται σε τακτά χρονικά διαστήματα.
- Οι τιμές να αποθηκεύονται σε τοπική βάση δεδομένων.

Στη συνέχεια με την επεξεργασία των δεδομένων θα μπορούσαν να εξαχθούν συμπεράσματα όπως σε ποια τηλεφωνικά κελιά το σήμα του κινητού είναι ασθενή ή σε ποιες περιοχές είναι γρηγορότερη η μεταφόρτωση αρχείων με τη χρήση της κινητής τηλεφωνίας.

## Βιβλιογραφία

- [1] "Android", <https://www.android.com/>
- [2] "Android 1.5", <http://developer.android.com/about/versions/android-1.5.html>
- [3] "Android 1.6", <http://developer.android.com/about/versions/android-1.6-highlights.html>
- [4] "Android 2.0", <http://developer.android.com/about/versions/android-2.0-highlights.html>
- [5] "Android 2.2", <http://developer.android.com/about/versions/android-2.2-highlights.html>
- [6] "Android 3.0", <http://developer.android.com/about/versions/android-3.0-highlights.html>
- [7] "Android 4.0", <http://developer.android.com/about/versions/android-4.0-highlights.html>
- [8] "Android Jelly bean", <http://developer.android.com/about/versions/jelly-bean.html>
- [9] "Android KitKat", <http://developer.android.com/about/versions/kitkat.html>
- [10] "Android Lollipop", <http://developer.android.com/about/versions/lollipop.html>
- [11] "Android Architecture", <http://www.android-app-market.com/android-architecture.html>
- [12] "Android Fundamentals",  
<http://developer.android.com/guide/components/fundamentals.html>
- [13] "Android Activity", <http://developer.android.com/reference/android/app/Activity.html>
- [14] "Android ListView", <http://developer.android.com/guide/topics/ui/layout/listview.html>
- [15] "Android Gridview", <http://developer.android.com/guide/topics/ui/layout/gridview.html>
- [16] "Android Layout", <http://developer.android.com/guide/topics/ui/declaring-layout.html>
- [17] "Android Service", <http://developer.android.com/guide/components/services.html>
- [18] "Android Control", <http://developer.android.com/guide/topics/ui/controls.html>
- [19] "Android menu", <http://developer.android.com/guide/topics/ui/menus.html>
- [20] "Android Dialog", <http://developer.android.com/guide/topics/ui/dialogs.html>

- [21] "Android Notification",  
<http://developer.android.com/guide/topics/ui/notifiers/notifications.html>
- [22] "Android Manifest", <http://developer.android.com/guide/topics/manifest/manifest-intro.html>
- [23] "Android Studio", <https://developer.android.com/studio/index.html>
- [24] "Android Studio Features", <https://developer.android.com/studio/features.html>
- [25] "(Android Virtual Device Tools AVD ".  
<https://developer.android.com/studio/run/managing-avds.html>
- [26] "Android Emulator", <https://developer.android.com/studio/run/emulator.html>
- [27] "Hierarchy Viewer", <https://developer.android.com/studio/profile/hierarchy-viewer.html>
- [28] "Lint Tools", <https://developer.android.com/studio/write/lint.html>
- [29] "Android Studio Debug Tools", <https://developer.android.com/studio/debug/index.html>
- [30] "Android Studio Build Tools", <https://developer.android.com/studio/releases/build-tools.html>
- [31] "Android Studio Platform Tools", <https://developer.android.com/studio/releases/sdk-tools.html>
- [32] "Android Connection Status example", <http://www.androidhive.info/2012/07/android-detect-internet-connection-status/>
- [33] <https://developer.android.com/reference/android/net/ConnectivityManager.html>

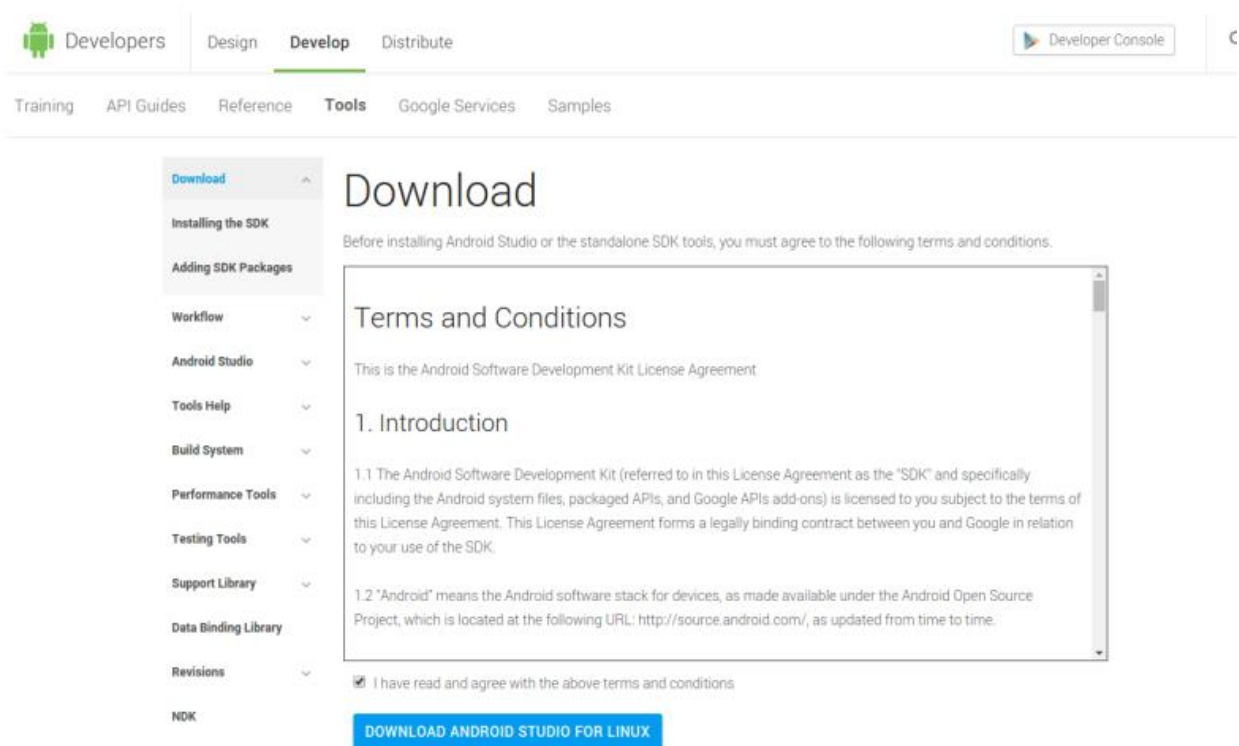
# Παράρτημα

## Εγκατάσταση Android Studio

Τα παρακάτω βήματα είναι απαραίτητα για την εγκατάσταση και παραμετροποίηση του Android Studio

Βήμα 1 μεταφόρτωση του αρχείου από τον διαδικτυακό τόπο

<https://developer.android.com/studio/index.html>



The screenshot shows the Android Developers website's 'Download' page. The navigation bar includes 'Developers', 'Design', 'Develop', and 'Distribute'. Below the navigation bar, there are links for 'Training', 'API Guides', 'Reference', 'Tools', 'Google Services', and 'Samples'. The main content area is titled 'Download' and contains a warning: 'Before installing Android Studio or the standalone SDK tools, you must agree to the following terms and conditions.' A dialog box titled 'Terms and Conditions' is open, displaying the Android Software Development Kit License Agreement. The dialog box includes sections for '1. Introduction' and '1.1 The Android Software Development Kit (referred to in this License Agreement as the "SDK" and specifically including the Android system files, packaged APIs, and Google APIs add-ons) is licensed to you subject to the terms of this License Agreement. This License Agreement forms a legally binding contract between you and Google in relation to your use of the SDK.' Below the dialog box, there is a checkbox labeled 'I have read and agree with the above terms and conditions' which is checked. At the bottom of the dialog box, there is a blue button labeled 'DOWNLOAD ANDROID STUDIO FOR LINUX'.

Ο υπολογιστής πρέπει να πληρεί τις παρακάτω προϋποθέσεις,

### Windows

- Microsoft® Windows® 7/8/10 (32- or 64-bit)
- 2 GB RAM minimum, 8 GB RAM recommended



- 2 GB of available disk space minimum,  
4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution
- Java Development Kit (JDK) 8

### **Mac**

- Mac® OS X® 10.8.5 or higher, up to 10.11.4 (El Capitan)
- 2 GB RAM minimum, 8 GB RAM recommended
- 2 GB of available disk space minimum,  
4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution
- Java Development Kit (JDK) 6

### **Linux**

- GNOME or KDE desktop *Tested on Ubuntu® 12.04, Precise Pangolin (64-bit distribution capable of running 32-bit applications)*
- 64-bit distribution capable of running 32-bit applications
- GNU C Library (glibc) 2.11 or later
- 2 GB RAM minimum, 8 GB RAM recommended
- 2 GB of available disk space minimum,  
4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution
- Java Development Kit (JDK) 8

Ακολούθησε τα βήματα του video για την ολοκλήρωση της εγκατάστασης.

<https://developer.android.com/studio/install.html>

### **MainActivity.java**

```

package com.example.ntinosmarios.ntinosmarios;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.content.Context;
import android.os.Build;
import android.os.Message;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.os.Handler;
import android.net.*;
import android.telephony.CellIdentityGsm;
import android.telephony.CellInfo;
import android.telephony.CellInfoGsm;
import android.telephony.CellLocation;
import android.telephony.CellSignalStrengthGsm;
import android.telephony.NeighboringCellInfo;
import android.telephony.PhoneStateListener;
import android.telephony.SignalStrength;
import android.telephony.TelephonyManager;
import android.telephony.cdma.CdmaCellLocation;
import android.telephony.gsm.GsmCellLocation;
import android.util.Log;
import android.view.*;
import android.widget.*;

import java.io.IOException;
import java.io.InputStream;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLConnection;
import java.text.DecimalFormat;
import java.util.Calendar;
import java.util.List;
import java.util.Random;
import java.util.logging.LogRecord;

import android.os.Message;

import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.DefaultHttpClient;

public class MainActivity extends AppCompatActivity {
    private Context mContext;
    private TelephonyManager tm;
    myPhoneStateListener psListener;
    private static String status = "";
    private Handler mHandlerRxTx = new Handler();
    private long mStartRX = 0;
    private long mStartTX = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

```

```

        setContentView(R.layout.activity_main);
        mContext = getApplicationContext();
        tm = (android.telephony.TelephonyManager)
getSystemService(Context.TELEPHONY_SERVICE);

        //check wifi connection
        if (isWifiConnected(getApplicationContext())) {
            Toast.makeText(mContext, "wifi is connected",
Toast.LENGTH_LONG).show();
            TextView temp = (TextView)findViewById(R.id.wifi);
            temp.setText("Wifi Status : CONNECTED");

        }
        else {
            Toast.makeText(mContext, "wifi is connected",
Toast.LENGTH_LONG).show();
            TextView temp = (TextView)findViewById(R.id.wifi);
            temp.setText("Wifi Status : NOT CONNECTED");
        }

        //check mobile connection
        if (isMobileConnected(getApplicationContext())) {
            Toast.makeText(mContext, "G is connected",
Toast.LENGTH_LONG).show();
            TextView temp = (TextView)findViewById(R.id.mobile);
            temp.setText("Mobile Status : SERVICE AVAILABLE");
        }
        else {
            Toast.makeText(mContext, "G is not connected",
Toast.LENGTH_LONG).show();
            TextView temp = (TextView)findViewById(R.id.mobile);
            temp.setText("Mobile Status : SERVICE AVAILABLE");
        }

        //get signal strenght
        psListener = new myPhoneStateListener();
        tm.listen(psListener, PhoneStateListener.LISTEN_SIGNAL_STRENGTHS);

        //find location details
        CellLocation location = tm.getCellLocation();
        if (location instanceof GSMCellLocation) {
            GSMCellLocation gsmLocation = (GSMCellLocation) location;
            int cellid = gsmLocation.getCellId();
            int lacid = gsmLocation.getCellId();
            TextView temp = (TextView)findViewById(R.id.cellid);
            temp.setText("Cell Id : " + cellid);
            TextView temp1 = (TextView)findViewById(R.id.locid);
            temp1.setText("Loc Id : " + lacid);
        } else if (location instanceof CDMACellLocation) {
            CDMACellLocation cdmaLocation = (CDMACellLocation) location;
            int cellid = cdmaLocation.getBaseStationId();//.getCellId();
            int latid = cdmaLocation.getBaseStationLatitude();
            int lonid = cdmaLocation.getBaseStationLongitude();
            TextView temp = (TextView)findViewById(R.id.cellid);
            temp.setText("Cell Id : " + cellid);
            TextView temp1 = (TextView)findViewById(R.id.locid);
            temp1.setText("Lat : " + latid + "Lon : " + lonid);
        }
    }
}

```

```

    } else {
        //do nothing
    }
    //compute throughput
    //throughput = windowssize/RTT
    if (isWifiConnected(getApplicationContext()) ||
isMobileConnected(getApplicationContext())) {

        Network test = new Network();
        try {

            float rtt = Float.parseFloat(test.ping("8.8.8.8"));
            float throughput = 524288 / rtt;

            TextView temp = (TextView)findViewById(R.id.throughput);
temp.setText("Throughput : " + String.valueOf(throughput) +
" bits/sec");

        } catch (Exception ex) {
            Log.d(ex.toString(), "<-ping");
        }

    }
    //compute dl speed
    new Thread(mWorker).start();

    //compute ul speed
    mStartRX = TrafficStats.getTotalRxBytes();
    mStartTX = TrafficStats.getTotalTxBytes();

    if (mStartRX == TrafficStats.UNSUPPORTED || mStartTX ==
TrafficStats.UNSUPPORTED) {

        Log.d("no traffic stats", "no traffic stats");
    } else {
        mHandlerRxTx.postDelayed(mRunnable1, 1000);
    }

}

//check if wifi is connected
public static boolean isWifiConnected(@NonNull Context context) {
    return isConnected(context, ConnectivityManager.TYPE_WIFI);
}
//check if data gsm/3g/4g/... is connected
public static boolean isMobileConnected(@NonNull Context context) {
    return isConnected(context, ConnectivityManager.TYPE_MOBILE);
}

public static boolean isConnected(@NonNull Context context, int
typeOfNetwork) {
    ConnectivityManager connMgr = (ConnectivityManager)
context.getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();
    return (networkInfo != null && networkInfo.isConnected());
}

```

```

}

public class myPhoneStateListener extends PhoneStateListener {
    public int signalStrengthValue;

    public void onSignalStrengthsChanged(SignalStrength signalStrength) {
        super.onSignalStrengthsChanged(signalStrength);
        if (signalStrength.isGsm()) {
            if (signalStrength.getGsmSignalStrength() != 99)
                signalStrengthValue =
signalStrength.getGsmSignalStrength() * 2 - 113;
            else
                signalStrengthValue =
signalStrength.getGsmSignalStrength();
        } else {
            signalStrengthValue = signalStrength.getCdmaDbm();
        }

        TextView temp = (TextView)findViewById(R.id.signal);
        temp.setText("Signal Strength : " + signalStrengthValue);
    }
}

private final android.os.Handler mHandler=new android.os.Handler(){

    public void publish(LogRecord record) {

    }

    public void flush() {

    }

    public void close() throws SecurityException {

    }

    @Override
    public void handleMessage(final Message msg) {
        switch(msg.what){
            case MSG_UPDATE_STATUS:
                final SpeedInfo info1=(SpeedInfo) msg.obj;

                break;
            case MSG_UPDATE_CONNECTION_TIME:

                break;
            case MSG_COMPLETE_STATUS:
                final SpeedInfo info2=(SpeedInfo) msg.obj;

                TextView temp = (TextView)findViewById(R.id.downspeed);
                temp.setText("Download Speed : " +
String.valueOf(info2.downspeed) + " bytes/sec");

                if(networkType(info2.kilobits)==1){

```

```

        // network_3g);
    }else{
        // network edge);
    }

    break;
default:
    super.handleMessage(msg);
}
}
};
/**
 * Our Slave worker that does actually all the work
 */
protected final Runnable mWorker=new Runnable(){

    @Override
    public void run() {
        InputStream stream=null;
        try {
            int bytesIn=0;
            String
downloadFileUrl="ftp://ftp.ntua.gr/mirror/3drealms/sw/maps/n-z/nympho.txt";
            long startCon=System.currentTimeMillis();
            URL url=new URL(downloadFileUrl);
            URLConnection con=url.openConnection();
            con.setUseCaches(false);
            stream=con.getInputStream();
            long connectionLatency=System.currentTimeMillis()- startCon;

            Message msgUpdateConnection= Message.obtain(mHandler,
MSG_UPDATE_CONNECTION_TIME);
            msgUpdateConnection.arg1=(int) connectionLatency;
            mHandler.sendMessage(msgUpdateConnection);

            long start=System.currentTimeMillis();
            int currentByte=0;
            long updateStart=System.currentTimeMillis();
            long updateDelta=0;
            int bytesInThreshold=0;

            while((currentByte=stream.read())!=-1){
                bytesIn++;
                bytesInThreshold++;
                if(updateDelta>=UPDATE_THRESHOLD){
                    int
progress=(int) ((bytesIn/(double)EXPECTED_SIZE_IN_BYTES)*100);
                    Message msg=Message.obtain(mHandler,
MSG_UPDATE_STATUS, calculate(updateDelta, bytesInThreshold));
                    msg.arg1=progress;
                    msg.arg2=bytesIn;
                    mHandler.sendMessage(msg);
                    //Reset
                    updateStart=System.currentTimeMillis();
                    bytesInThreshold=0;
                }
                updateDelta = System.currentTimeMillis()- updateStart;

```

```

    }

    long downloadTime=(System.currentTimeMillis()-start);
    //Prevent ArithmeticException
    if(downloadTime==0){
        downloadTime=1;
    }

    Message msg=Message.obtain(mHandler, MSG_COMPLETE_STATUS,
calculate(downloadTime, bytesIn));
    msg.arg1=bytesIn;
    mHandler.sendMessage(msg);
    //Toast.makeText(mContext, "worker ended",
Toast.LENGTH_LONG).show();
    }
    catch (MalformedURLException e) {

    } catch (IOException e) {

    }finally{
        try {
            if(stream!=null){
                stream.close();
            }
        } catch (IOException e) {
            //Suppressed
        }
    }
}
};

private int networkType(final double kbps){
    int type=1;//3G
    //Check if its EDGE
    if(kbps<EDGE_THRESHOLD){
        type=0;
    }
    return type;
}

private SpeedInfo calculate(final long downloadTime, final long bytesIn){
    SpeedInfo info=new SpeedInfo();
    //from mil to sec
    long bytespersecond =(bytesIn / downloadTime) * 1000;
    double kilobits=bytespersecond * BYTE_TO_KILOBIT;
    double megabits=kilobits * KILOBIT_TO_MEGABIT;
    info.downspeed=bytespersecond;
    info.kilobits=kilobits;
    info.megabits=megabits;

    return info;
}

private static class SpeedInfo{
    public double kilobits=0;
    public double megabits=0;
}

```

```

    public double downspeed=0;
}

private final Runnable mRunnable1 = new Runnable() {
    public void run() {

        //compute upload speed
        mStartRX = TrafficStats.getTotalRxBytes();
        mStartTX = TrafficStats.getTotalTxBytes();
        if (mStartRX == TrafficStats.UNSUPPORTED || mStartTX ==
TrafficStats.UNSUPPORTED) {

            Log.d("no traffic stats", "no traffic stats");
        } else {

            long BeforeTime = System.currentTimeMillis();
            long TotalTxBeforeTest = TrafficStats.getTotalTxBytes();
            long TotalRxBeforeTest = TrafficStats.getTotalRxBytes();

            Network test = new Network();

            try {
                test.ping("8.8.8.8");
                test.ping("8.8.8.8");
                test.ping("8.8.8.8");
            } catch (IOException e) {
                e.printStackTrace();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }

            long TotalTxAfterTest = TrafficStats.getTotalTxBytes();
            long TotalRxAfterTest = TrafficStats.getTotalRxBytes();
            long AfterTime = System.currentTimeMillis();

            double TimeDifference = AfterTime - BeforeTime;

            double rxDiff = TotalRxAfterTest - TotalRxBeforeTest;
            double txDiff = TotalTxAfterTest - TotalTxBeforeTest;

            if((rxDiff != 0) && (txDiff != 0)) {
                double rxBPS = (rxDiff / (TimeDifference/1000)); // total
rx bytes per second.
                double txBPS = (txDiff / (TimeDifference/1000)); // total
tx bytes per second.

                TextView temp = (TextView)findViewById(R.id.uploadspeed);
                temp.setText("Upload Speed : " + String.valueOf(txBPS) +
" B/s");
            }
            else {
                Log.d("0,", "No uploaded or downloaded bytes.");
            }
        }

        mHandler.postDelayed(mRunnable1, 1000);
    }
}

```



```

};

private static final int EXPECTED_SIZE_IN_BYTES = 1024;//1KB 1024*1024

private static final double EDGE_THRESHOLD = 176.0;
private static final double BYTE_TO_KILOBIT = 0.0078125;
private static final double KILOBIT_TO_MEGABIT = 0.0009765625;
private final int MSG_UPDATE_STATUS=0;
private final int MSG_UPDATE_CONNECTION_TIME=1;
private final int MSG_COMPLETE_STATUS=2;

private final static int UPDATE_THRESHOLD=300;

private DecimalFormat mDecimalFormater;
}

```

## Network.java

```

package com.example.ntinosmarios.ntinosmarios;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

import android.util.Log;

public class Network {

    private static final String TAG = "Network.java";

    public static String pingError = null;

    /**
     * Ping a host and return an int value of 0 or 1 or 2 0=success, 1=fail,
     2=error
     *
     * Does not work in Android emulator and also delay by '1' second if host
     not pingable
     * In the Android emulator only ping to 127.0.0.1 works
     *
     * @param String host in dotted IP address format
     * @return
     * @throws IOException
     * @throws InterruptedException
     */
    public static int pingHost(String host) throws IOException,
    InterruptedException {
        Runtime runtime = Runtime.getRuntime();
        Process proc = runtime.exec("ping -c 1 " + host);
        proc.waitFor();
        int exit = proc.exitValue();
        return exit;
    }
}

```

```

    public static String ping(String host) throws IOException,
    InterruptedException {
        StringBuffer echo = new StringBuffer();
        Runtime runtime = Runtime.getRuntime();
        Log.v(TAG, "About to ping using runtime.exec");
        Process proc = runtime.exec("ping -c 1 " + host);
        proc.waitFor();
        int exit = proc.exitValue();
        if (exit == 0) {
            InputStreamReader reader = new
    InputStreamReader(proc.getInputStream());
            BufferedReader buffer = new BufferedReader(reader);
            String line = "";
            while ((line = buffer.readLine()) != null) {

                echo.append(line + "\n");
            }
            Log.d("echo.toString()",echo.toString());
            return getPingStats(echo.toString());
        } else if (exit == 1) {
            pingError = "failed, exit = 1";
            Log.d("failed, exit = 1", "failed, exit = 1");
            return null;
        } else {
            pingError = "error, exit = 2";
            Log.d("error, exit = 2", "error, exit = 2");
            return null;
        }
    }
}

/**
 * getPingStats interprets the text result of a Linux ping command
 *
 * Set pingError on error and return null
 *
 * http://en.wikipedia.org/wiki/Ping
 *
 * PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
 * 64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.251 ms
 * 64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.294 ms
 * 64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.295 ms
 * 64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.300 ms
 *
 * --- 127.0.0.1 ping statistics ---
 * 4 packets transmitted, 4 received, 0% packet loss, time 0ms
 * rtt min/avg/max/mdev = 0.251/0.285/0.300/0.019 ms
 *
 * PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data.
 *
 * --- 192.168.0.2 ping statistics ---
 * 1 packets transmitted, 0 received, 100% packet loss, time 0ms
 *
 * # ping 321321.
 * ping: unknown host 321321.
 *
 * 1. Check if output contains 0% packet loss : Branch to success -> Get
stats

```

```

    * 2. Check if output contains 100% packet loss : Branch to fail -> No
stats
    * 3. Check if output contains 25% packet loss : Branch to partial
success -> Get stats
    * 4. Check if output contains "unknown host"
    *
    * @param s
    */
public static String getPingStats(String s) {

    if (s.contains("0% packet loss")) {
        int start = s.indexOf("/mdev = ");
        int end = s.indexOf(" ms\n", start);
        s = s.substring(start + 8, end);
        String stats[] = s.split("/");
        return stats[2];
    } else if (s.contains("100% packet loss")) {
        pingError = "100% packet loss";
        return null;
    } else if (s.contains("% packet loss")) {
        pingError = "partial packet loss";
        return null;
    } else if (s.contains("unknown host")) {
        pingError = "unknown host";
        return null;
    } else {
        pingError = "unknown error in getPingStats";
        return null;
    }
}
}
}

```