



*Τεχνολογικό Εκπαιδευτικό Ίδρυμα Μεσολογγίου  
Τμήμα Τηλεπικοινωνιακών Συστημάτων και Δικτύων*

## *Πτυχιακή Εργασία*

*Θέμα: Ανάλυση και Σχεδιασμός Εφαρμογής Android  
καταγραφής Location των χρηστών κοινωνικών δικτύων με  
χρήση GPS.*

*Σπουδάστρια:  
Γκούβα Πολυτίμη*

*Επιβλέπων Καθηγητής:  
Σπυρίδων Λούβρος*

*Ναύπακτος 2013*



## ΠΕΡΙΕΧΟΜΕΝΑ

<b>Πίνακας Εικόνων</b> .....	5
<b>Περίληψη</b> .....	7
<b>1 Εισαγωγή για το λογισμικό Android</b> .....	9
<b>1.1</b> Τι είναι το Android.....	9
<b>1.2</b> Εφαρμογές Android.....	10
<b>1.3</b> Εξέλιξη του Android- Εκδόσεις και χαρακτηριστικά.....	11
<b>1.3.1</b> Android 1.5 Cupcake.....	12
<b>1.3.2</b> Android 1.6 Donut.....	13
<b>1.3.3</b> Android 2.0/2.1 Eclair.....	14
<b>1.3.4</b> Android 2.2 Froyo.....	16
<b>1.3.5</b> Android 2.3 Gingerbread.....	17
<b>1.3.6</b> Android 3.0 Honeycomb.....	19
<b>1.3.7</b> Android 4.0 Ice Cream Sandwich.....	22
<b>1.3.8</b> Android 4.1/4.2 Jelly Beans.....	23
<b>1.3.9</b> Android 5.0 Key Lime Pie.....	24
<b>1.4</b> Εφαρμογές και Widgets.....	25
<b>1.5</b> Γιατί Android;.....	25
<b>2 Αρχιτεκτονική του Android</b> .....	27
<b>2.1</b> Applications.....	28
<b>2.2</b> Πυρήνας Linux ( Linux Kernel).....	28
<b>2.3</b> Βιβλιοθήκες – Libraries.....	28
<b>2.3</b> Προηγμένες Βιβλιοθήκες Android.....	30
<b>2.4</b> Χρόνος εκτέλεσης (Runtime Android).....	31
<b>2.5</b> Εικονική μηχανή Dalvik.....	31
<b>2.6</b> Πλαίσιο εφαρμογής (App Frameworks).....	32
<b>2.7</b> Βασικά συστατικά του Android.....	34
<b>2.7.1</b> Δραστηριότητα- Activity.....	34
<b>2.7.2</b> Υπηρεσία – Service.....	35
<b>2.7.3</b> Παροχέας Περιεχομένων- Content Provider.....	35
<b>2.7.4</b> Προθέσεις- Intent.....	36

<b>3</b>	<b>Εργαλεία ανάπτυξης λογισμικού.....</b>	<b>39</b>
3.1	Android SDK.....	39
3.1.1	Χαρακτηριστικά του SDK.....	39
3.1.2	Εξομοιωτής Emulator.....	40
3.1.3	Άλλα εργαλεία του Android.....	41
3.2	<b>Ανάπτυξη εφαρμογών Android.....</b>	<b>42</b>
3.2.1	Εγκατάσταση λογισμικού.....	42
3.2.2	Ανάπτυξη πηγαίου κώδικα εφαρμογής.....	42
3.2.3	Αποσφαλμάτωση Debugging.....	43
3.2.4	Android Debug Bridge (ADB).....	43
3.2.5	Εικονικές συσκευές Android.....	44
<b>4</b>	<b>Android Application.....</b>	<b>47</b>
4.1	<b>Android App με την χρήση SDK και Eclipse.....</b>	<b>47</b>
4.1.1	Eclipse IDE.....	47
4.1.2	Εγκατάσταση του ADT plugin για το Eclipse.....	47
4.1.3	Εγκατάσταση του Android SDK.....	49
<b>5</b>	<b>Υλοποίηση Εφαρμογής .....</b>	<b>51</b>
5.1	Περιγραφή εφαρμογής.....	51
5.2	Εφαρμογή GPS_Post_Locations.....	53
5.3	Δημιουργία GPS Retrieval εφαρμογής.....	63
i.	<b>Βιβλιογραφία.....</b>	<b>67</b>

## **Πίνακας Εικόνων**

- Εικόνα 1.1 :** Μερίδιο Αγοράς
- Εικόνα 1.2 :** Εξέλιξη των εκδόσεων Android.
- Εικόνα 1.3 :** Λογότυπο Android 1.5 CUPCAKE
- Εικόνα 1.4 :** Λογότυπο Android 1.6 DONUT
- Εικόνα 1.5 :** Λογότυπο Android 2.0 ECLAIR
- Εικόνα 1.6 :** Λογότυπο Android 2.2 FROYO
- Εικόνα 1.7 :** Λογότυπο Android 2.3 GINGERBREAD
- Εικόνα 1.8 :** Λογότυπο Android 3.0 HONEYCOMB
- Εικόνα 1.9 :** Λογότυπο Android Ice Cream Sandwich
- Εικόνα 1.10:** Λογότυπο Android Jelly Beans
- Εικόνα 1.11:** Λογότυπο Android Key Lime Pie
- Εικόνα 2.1 :** Τα βασικά περιεχόμενα του λειτουργικού συστήματος Android
- Εικόνα 2.2 :** Κύκλος ζωής μιας δραστηριότητας(Activity lifecycle)
- Εικόνα 4.1 :** Επιλογή αποθήκευσης project
- Εικόνα 4.2 :** Εγκατάσταση του Eclipse.
- Εικόνα 4.3 :** Εγκατάσταση SDK
- Εικόνα 5.1 :** Διαθέσιμα LocationProvider.
- Εικόνα 5.2 :** Posting Location εφαρμογή στον emulator.
- Εικόνα 5.3 :** GPS Retrieve Location εφαρμογή στον Emulator.



## **ΠΕΡΙΛΗΨΗ**

Στην παρούσα πτυχιακή εργασία γίνεται ανάλυση και περιγραφή του λογισμικού Android . Στόχος είναι η υλοποίηση μιας εφαρμογής Android η οποία θα καταγράφει το location , με την χρήση του GPS, των χρηστών κοινωνικών δικτύων.

Αρχικά αναφέρονται ορισμένα βασικά ιστορικά στοιχεία και στην συνέχεια αναλύεται η εξέλιξη των εκδόσεων του Android λογισμικού. Επίσης παρουσιάζονται τα βασικά στοιχεία του λογισμικού και τα χαρακτηριστικά της αρχιτεκτονικής του.

Στην συνέχεια περιγράφεται η χρήση των εργαλείων στο σύστημα ανάπτυξης εφαρμογών Android. Τέλος χρησιμοποιώντας τα παραπάνω εργαλεία και μετά απο μελέτη αυτών , αναπτύχθηκε μια εφαρμογή καταγραφής Location με χρήση GPS. Για την επίτευξη του στόχου χρησιμοποιήθηκε το λογισμικό Eclipse, το οποίο είναι πρόγραμμα ανοιχτού κώδικα, και το εργαλείο ανάπτυξης λογισμικού Android SDK.



## **1 ΕΙΣΑΓΩΓΗ ΓΙΑ ΤΟ ΛΟΓΙΣΜΙΚΟ ANDROID**

### **1.1 Τι είναι το Android ;**

Το Android είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα για συσκευές κινητής τηλεφωνίας, το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google.

Ξεκινώντας την περιήγηση στον κόσμο του Android, θα ήταν καλό να αναφερθούν ορισμένα σημαντικά ιστορικά στοιχεία. Τον Ιούλιο του 2005, η Google εξαγόρασε την Android Inc, μια μικρή εταιρεία με έδρα το Palo Alto στην California των ΗΠΑ. Οι συνιδρυτές της Android πήγαν να εργαστούν στην Google συμπεριλαμβανομένων των Andy Rubin (συν-ιδρυτής της Danger), Rich Miner (συν-ιδρυτής της Wildfire Communications, Inc), Nick Sears (πρώην αντιπρόεδρος της T-Mobile), και Chris White (επικεφαλής σχεδιασμού και ανάπτυξης interface στο WebTV). Εκείνη την εποχή ελάχιστα ήταν γνωστά για τις λειτουργίες της Android Inc, εκτός του ότι ανέπτυσαν λογισμικό για κινητά τηλέφωνα. Αυτή ήταν η αρχή της φημολογίας περί σχεδίων της Google για να διεισδύσει στην αγορά κινητής τηλεφωνίας.

Στην Google, η ομάδα με επικεφαλής τον Rubin ανέπτυξε μια κινητή πλατφόρμα που στηρίζεται στον πυρήνα του Linux, την οποία προώθησαν με την παροχή ενός ευέλικτου, αναβαθμίσιμου συστήματος. Έχει αναφερθεί ότι η Google έχει ήδη συγκεντρώσει μια σειρά από εταιρίες hardware και software και επισήμανε στους παρόχους ότι ήταν ανοικτή σε διάφορους βαθμούς συνεργασίας εκ μέρους της. Περισσότερες εικασίες ότι η Google θα εισέλθει στην αγορά κινητής τηλεφωνίας άρχισαν τον Δεκέμβριο του 2006.

Τον Σεπτέμβριο του 2007, η InformationWeek κάλυψε μια μελέτη αξιολόγησης αναφέροντας ότι η Google έχει καταθέσει αρκετές πατέντες στον τομέα της κινητής τηλεφωνίας. Τελικά η Google παρουσίασε το smartphone της Nexus One που χρησιμοποιεί το open source λειτουργικό σύστημα Android. Η συσκευή κατασκευάστηκε από την HTC Corporation της Ταϊβάν, και έγινε διαθέσιμη στις 5 Ιανουαρίου 2010.

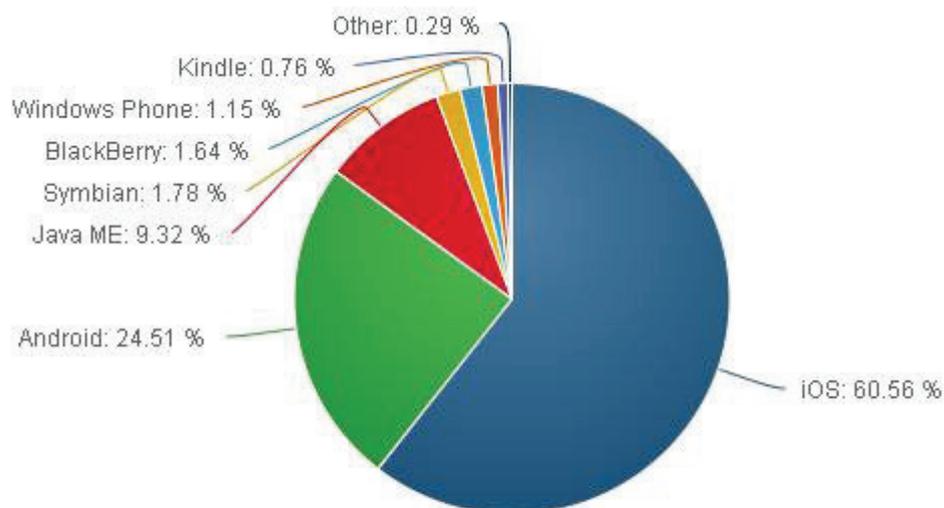
Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού **Open Handset Alliance**, μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής hardware, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές κινητής τηλεφωνίας.

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

## **1.2 Εφαρμογές Android**

Το Android έχει μια μεγάλη κοινότητα προγραμματιστών που αναπτύσσουν εφαρμογές, οι οποίες επεκτείνουν τη λειτουργικότητα των συσκευών. Όπως προαναφέρθηκε, οι προγραμματιστές γράφουν κατά κύριο λόγο σε μια προσαρμοσμένη έκδοση της Java. Σήμερα υπάρχουν πάνω από 520.000 εφαρμογές διαθέσιμες για το Android, τις οποίες οι χρήστες μπορούν να προμηθευτούν από δικτυακούς τόπους τρίτων ή μέσω online καταστημάτων όπως το Android Market, το online app κατάστημα της Google.

Εν κατακλείδι, η ευελιξία, η ευκολία και η ελευθερία πρόσβασης στον κώδικα του Android δικαιολογούν απόλυτα την εντυπωσιακή αύξηση της δημοτικότητας του. Στο ακόλουθο γράφημα αποτυπώνεται το μερίδιο αγοράς που καταλαμβάνει το κάθε λειτουργικό σύστημα κινητών συσκευών, το οποίο δημοσιεύτηκε τον Φεβρουάριο του 2013.



Εικόνα1.1 Μερίδιο Αγοράς

### **1.3 Εξέλιξη του Android- Εκδόσεις και χαρακτηριστικά**



Εικόνα 1.2. Εξέλιξη των εκδόσεων Android.

Το λειτουργικό Android ξεκίνησε με την απελευθέρωση της έκδοσης 1.0 το Σεπτέμβριο του 2008. Ωστόσο, στην αγορά κυκλοφορούν σε τακτά χρονικά διαστήματα ενημερώσεις, οι οποίες επεκτείνουν με εντυπωσιακό τρόπο την αρχική έκδοση. Η βασική λογική μιας ενημέρωσης ωστόσο παραμένει η διόρθωση των σφαλμάτων προηγούμενων κυκλοφοριών. Τα ονόματα της κάθε έκδοσης προκύπτουν κατά αλφαβητική σειρά από ονόματα επιδορπίων. Στη συνέχεια παρατίθενται αναλυτικά όλες οι εκδόσεις και τα κυριότερα χαρακτηριστικά τους.

### **1.3.1 Android 1.5 Cupcake**



Εικόνα 1.3.Λογότυπο Android 1.5 CUPCAKE

Με βάση το Linux kernel 2.6.27, στις 30 Απριλίου 2009 κυκλοφόρησε επίσημα η έκδοση 1.5, η επονομαζόμενη και Cupcake:

- Υποστήριξη εικονικού πληκτρολογίου με δυνατότητα πρόβλεψης λεξικού χρήστη για προσαρμοσμένες λέξεις.
- Υποστήριξη **widgets**, δηλαδή μικροεφαρμογές οι οποίες μπορούν να ενσωματωθούν σε άλλες εφαρμογές (όπως στην αρχική οθόνη) και να λαμβάνουν περιοδικές ενημερώσεις.
- Εγγραφή βίντεο.
- Αναπαραγωγή βίντεο αρχείων με επεκτάσεις **MPEG-4** και **3GP**
- Στην υπηρεσία Bluetooth προστέθηκε stereo υποστήριξη, βάσει των προφίλ **A2DP** και **AVRCP**.
- Στο περιηγητή internet (*web browser*) έγιναν προσθήκες που αφορούν την αντιγραφή και την επικόλληση
- Στις επαφές εμφανίζεται η εικόνα του άλλου χρήστη.
- Σε ό,τι αφορά το dialer, υπάρχει συγκεκριμένη σφραγίδα ημερομηνίας και ώρας για τα γεγονότα στο αρχείο κλήσεων και δυνατότητα πρόσβασης σε μια επαφή μέσω του εν λόγω αρχείου.
- Εφέ κίνησης συνοδεύουν τις διάφορες μεταβάσεις στην οθόνη.
- Ανέβασμα (*upload*) βίντεο στο YouTube.
- Ανέβασμα φωτογραφιών στο **Picasa**.

### **1.3.2 Android 1.6 Donut**

Με βάση το Linux kernel 2.6.29, στις 15 Σεπτέμβρη 2009 κυκλοφόρησε η επόμενη νέα έκδοση, κατά την οποία:



Εικόνα 1.4.Λογότυπο Android 1.6 DONUT

- Η φωνητική αναζήτηση και το κείμενο αναζήτησης βελτιώνονται έτσι ώστε να περιλαμβάνονται οι σελιδοδείκτες, το ιστορικό, οι επαφές, το διαδίκτυο, κ.α.
- Οι προγραμματιστές μπορούν να συμπεριλάβουν πλέον το προϊόν τους στα αποτελέσματα αναζήτησης.
- Υπηρεσία κειμένου σε ομιλία, κατά την οποία διατίθεται μια πολύγλωσση μηχανή σύνθεσης ομιλίας που επιτρέπει σε κάθε Android εφαρμογή να παράγει μια συμβολοσειρά κειμένου.
- Στο Android Market η αναζήτηση γίνεται πλέον ευκολότερα.
- Ενσωματώθηκαν ενημερώσεις για ταχύτερη πρόσβαση στην κάμερα.

- Στην gallery εικόνων επιτρέπεται πλέον στους χρήστες να επιλέξουν πολλές φωτογραφίες για ταυτόχρονη διαγραφή.
- Ενημέρωση του συστήματος για την υποστήριξη των τεχνολογιών **CDMA**, **EVDO**, **802.1x** και **VPNs**.
- Υποστήριξη **WVGA** ανάλυση οθόνης.
- Βελτιώσεις στην ταχύτητα αναζήτησης καθώς και στις εφαρμογές φωτογραφικής μηχανής
- Διεύρυνση του **Gesture framework** και υποστήριξη του νέου εργαλείου ανάπτυξης **GestureBuilder** για **face-to-face επικοινωνία**.
- Δωρεάν δυνατότητα από την Google για **turn-by-turn** πλοήγηση, κατά την οποία το σύστημα ενημερώνει συνεχώς τον χρήστη σχετικά με την βέλτιστη διαδρομή προς τον προορισμό του.

### **1.3.3 Android 2.0/2.1 Eclair**

Με βάση το Linux kernel 2.6.29 και πάλι, στις 26 Οκτώβρη 2009 κυκλοφορεί η πρώτη έκδοση από την σειρά Eclair, κατά την οποία υπάρχουν τα ακόλουθα χαρακτηριστικά:



Εικόνα 1.5.Λογότυπο Android 2.0 ECLAIR

- Διεύρυνση του συγχρονισμού λογαριασμού, κατά την οποία πολλαπλοί λογαριασμοί μπορούν να προστεθούν σε μια συσκευή για ηλεκτρονική αλληλογραφία και συγχρονισμό επαφών.
- Επέκταση στις υπηρεσίες ηλεκτρονικού ταχυδρομείου, όπως είναι η υποστήριξη ανταλλαγών, κ.α.
- Τεχνολογία Bluetooth 2.1.
- Επιλογή μιας φωτογραφίας επαφής και δυνατότητα αποστολής email, SMS και τηλεφωνικής κλήσης.
- Αναζήτηση όλων των αποθηκευμένων μηνυμάτων (SMS, MMS). Αυτόματη διαγραφή παλαιότερων μηνυμάτων μιας συνομιλίας, όταν ξεπερνούν ένα προκαθορισμένο όριο.
- Σε ότι αφορά την κάμερα, πλέον υπάρχει υποστήριξη flash, ψηφιακό zoom, λειτουργία σκηνής, ισορροπία λευκού, εφέ χρωμάτων, macro εστίαση.
- Βελτιωμένη ταχύτητα δακτυλογράφησης σε ότι αφορά το εικονικό πληκτρολόγιο, εξυπνότερο λεξικό που μαθαίνει από την χρήση των λέξεων και περιλαμβάνει τα ονόματα των επαφών.
- Στον web browser υπάρχει ανανεωμένο UI, μικρογραφίες σελιδοδεικτών, zoom με διπλό χτύπημα και υποστήριξη **HTML5**.
- Στο ημερολόγιο ενισχύθηκε η προβολή ατζέντας, ευκολότερη ενημέρωση της κατάστασης των υπολοίπων αλλά και εύκολη πρόσκληση σε νέα γεγονότα.
- Βελτιωμένη ταχύτητα υλικού συστήματος καθώς και καλύτερο UI.
- Υποστήριξη οθόνης μεγαλύτερου μεγέθους, υψηλότερης ανάλυσης και με καλύτερη αναλογία αντίθεσης.
- Βελτιωμένοι και πληρέστεροι χάρτες με το Google Maps 3.1.2.
- Η κλάση **MotionEvent** παρουσιάζεται ενισχυμένη για **multi-touch** ενέργειες.
- Οι εικόνες φόντου και αρχικής οθόνης μπορούν πλέον να είναι κινουμένων σχεδίων.

Στις 3 Δεκεμβρίου 2009 κυκλοφορεί η έκδοση 2.0.1, η οποία περιλαμβάνει μικρές αλλαγές στο API, διορθώσεις σφαλμάτων και αλλαγές στην συμπεριφορά του framework. Περίπου ένα μήνα αργότερα, στις 12 Ιανουαρίου 2010 κυκλοφορεί η 2.1 έκδοση για να εισάγει μερικές ακόμη αλλαγές στο API και κάποιες διορθώσεις σφαλμάτων και να ολοκληρώσει την σειρά εκδόσεων Eclair.

### **1.3.4 Android 2.2 Froyo**

Βάσει του Linux kernel 6.2.32, στις 20 Μαΐου 2010 κυκλοφορεί μια ακόμη έκδοση με νέα ονομασία (*Froyo*). Οι αλλαγές που περιλαμβάνονται είναι:

- Βελτιστοποίηση συστήματος σε ότι σχετίζεται με ταχύτητα, μνήμη και απόδοση.
- Πρόσθετες βελτιώσεις ταχύτητας στη JIT μεταγλώττιση.



Εικόνα 1.6. Λογότυπο Android 2.2 FROYO

- Ενσωμάτωση της μηχανής **V8 JavaScript** του **Chrome** στην εφαρμογή περιήγησης.
- Βελτιωμένη υποστήριξη **Microsoft Exchange**.
- Βελτιωμένη εκκίνηση εφαρμογών με έξυπνες συντομεύσεις στο τηλέφωνο.
- USB διασύνδεση με άλλες συσκευές και λειτουργία σαν **Wi-Fi hotspot**.
- Προσθήκη επιλογής για απενεργοποίηση πρόσβασης σε δεδομένα μέσω του δικτύου κινητής τηλεφωνίας.
- Ενημέρωση Android Market και προσθήκη λειτουργιών αυτόματης ενημέρωσης.
- Γρήγορη εναλλαγή μεταξύ των πολλών γλωσσών πληκτρολογίου αλλά και των λεξικών τους.
- Φωνητική κλήση και κοινή χρήση επαφών μέσω Bluetooth.
- Υποστήριξη αριθμητικών και αλφαριθμητικών κωδικών πρόσβασης.
- Υποστήριξη πεδίων για ανέβασμα αρχείων στην εφαρμογή περιήγησης
- Δυνατότητα εγκατάστασης εφαρμογών στην επεκτάσιμη μνήμη.
- Υποστήριξη **Adobe Flash**.
- Υποστήριξη οθονών ακόμα υψηλότερων DPIs

### **1.3.5 Android 2.3 Gingerbread**

Η εν λόγω έκδοση κυκλοφόρησε στις 6 Δεκεμβρίου 2010 με το Linux kernel 2.6.35. Συνοπτικά, τα νέα χαρακτηριστικά αφορούν:



Εικόνα 1.7 Λογότυπο Android 2.3 GINGERBREAD

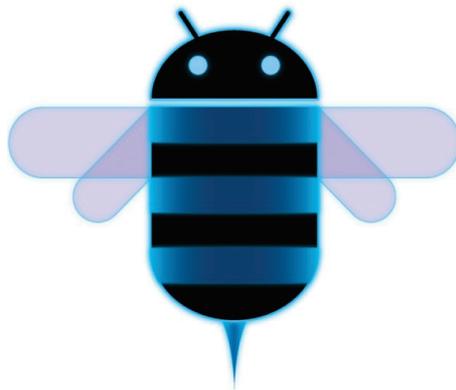
- Ενημερωμένη διεπαφή χρήστη – μηχανής για απλότητα και ταχύτητα
- Υποστήριξη πολύ μεγάλων μεγεθών οθόνης καθώς και αναλύσεων (**WXGA** και άνω)
- Υποστήριξη για **VoIP** τηλεφωνία.
- Αναφορικά με το εικονικό πληκτρολόγιο, υπάρχει εξέλιξη που εμπίπτει στην γρηγορότερη και εξυπνότερη εισαγωγή κειμένου, με βελτιωμένη ακρίβεια.
- Βελτιστοποιημένη η λειτουργία της αντιγραφής – επικόλλησης. Δυνατότητα επιλογής της επιθυμητής λέξης με απλό **press-hold**.
- Το **άμεσο πεδίο επικοινωνίας (Near Field Communication, NFC)** επιτρέπει στο χρήστη να διαβάσει μια NFC ετικέτα ενσωματωμένη σε μια αφίσα, σε ένα αυτοκόλλητο ή μια διαφήμιση.
- Νέα εφέ ήχου, όπως είναι η αντήχηση, η εξίσωση, κτλ.
- Ο νέος οδηγός λήψεων (**Download Manager**) παρέχει στους χρήστες εύκολη πρόσβαση σε οποιοδήποτε ληφθέν αρχείο.
- Πρόσβαση σε πολλαπλές κάμερες της συσκευής, συμπεριλαμβανομένης και αυτής που βρίσκεται στο μπροστινό μέρος του κινητού (*εφόσον υπάρχει*).

- Υποστήριξη αναπαραγωγής βίντεο για επεκτάσεις **WebM/VP8** και **AAC** κωδικοποίηση ήχου.
- Βελτιωμένη διαχείριση ενέργειας με έναν πιο ενεργό ρόλο στη διαχείριση των εφαρμογών που κρατούν τη συσκευή σε έντονη λειτουργία για πάρα πολύ ώρα.
- Ενισχυμένη υποστήριξη για ανάπτυξη κώδικα.
- Μεταπήδηση του συστήματος αρχείων από **YAFFS** σε **ext4** στις νεώτερες συσκευές.
- Βελτιώσεις σε ήχο και γραφικά που βοηθούν τους προγραμματιστές παιχνιδιών.
- Δυνατότητα συλλογής «απορριμμάτων» για αύξηση της απόδοσης.
- Υποστήριξη περισσότερων αισθητήρων (*sensors*), όπως είναι τα γυροσκόπια και τα βαρόμετρα.

Η έκδοση 2.3.3 επέφερε ορισμένες αλλαγές στο API σε σχέση με την αρχική πλατφόρμα 2.3. Η 2.3.4 εισήγαγε στο Google Talk τη δυνατότητα συνομιλίας με φωνή ή βίντεο. Η επόμενη έκδοση, μεταξύ άλλων, βελτίωσε την απόδοση του δικτύου στο Nexus S 4G διόρθωσε τα σφάλματα του Bluetooth στο Samsung Galaxy S και βελτίωσε την εφαρμογή για το Gmail.

### **1.3.6 Android 3.0 Honeycomb**

Στις 22 Φεβρουαρίου 2011, η πρώτη έκδοση από την σειρά Honeycomb κυκλοφόρησε για tablet computer με βάση το Linux kernel 2.6.36. Η πρώτη συσκευή που χαρακτηρίζει αυτή την έκδοση είναι το tablet Motorola Xoom, το οποίο κυκλοφόρησε στις 24 Φεβρουαρίου 2011. Οι αλλαγές περιλαμβάνουν:



Εικόνα 1.8. Λογότυπο Android 3.0 HONEYCOMB

- Βελτιστοποιημένη υποστήριξη tablet τεχνολογίας με νέο εικονικό περιβάλλον και διεπαφή χρήστη υπό μορφή ολογράμματος.
- Γρήγορη πρόσβαση σε ανακοινώσεις και στην κατάσταση του τηλεφώνου με την βοήθεια μαλακών πλήκτρων στο κάτω μέρος της οθόνης.
- Πρόσβαση σε ρυθμίσεις, πλοήγηση, widgets ή άλλο περιεχόμενο από την κορυφή της οθόνης.
- Καρφίτσωμα πρόσφατων εφαρμογών στην γραμμή συστήματος, προκειμένου να υπάρχει η δυνατότητα πρόσβασης σε εν εξελίξει εργασίες, αλλά και γρήγορη μετάβαση από τη μία εφαρμογή στην άλλη.
- Επανασχεδιασμένο πληκτρολόγιο.
- Απλοποίηση των διαδικασιών αντιγραφής και επικόλλησης.
- Πολλαπλές καρτέλες αντικατέστησαν τα παράθυρα του προγράμματος περιήγησης και δυνατότητα για ανώνυμη περιήγηση.
- Αναφορικά με την κάμερα, υπάρχει δυνατότητα γρήγορης πρόσβαση στην έκθεση, δυνατότητα εστίασης, flash, zoom, κάμερα στο μπροστινό μέρος της συσκευής, κ.α.
- Προβολή άλμπουμ και άλλων συλλογών σε πλήρη οθόνη, με εύκολη πρόσβαση σε μικρογραφίες άλλων φωτογραφιών.
- Το **UI two-pane** επιτρέπει στους χρήστες να οργανώνουν και να εντοπίζουν τις επαφές τους εύκολα.
- Επίσης, η εν λόγω τεχνολογία κάνει την προβολή και την οργάνωση των μηνυμάτων ηλεκτρονικού ταχυδρομείου περισσότερο αποτελεσματική.
- Υποστήριξη video chat χρησιμοποιώντας το Google Talk.
- Επιταχυνόμενο hardware.
- Υποστήριξη για επεξεργαστές πολλών πυρήνων.

### **Έκδοση 3.1 (Honeycomb)**

Το Android 3.1 κυκλοφόρησε στις 10 Μαΐου 2011. Οι αλλαγές που περιλαμβάνονται αφορούν:

- Βελτιώσεις στο UI.
- Συνδεσιμότητα αξεσουάρ μέσω **USB**.

- Διευρυμένη λίστα προσφάτων εφαρμογών.
- Δυνατότητα αλλαγής μεγέθους των widgets στην αρχική οθόνη.
- Υποστήριξη για εξωτερικά πληκτρολόγια και ποντίκια.
- Υποστήριξη για χειριστήρια παιχνιδιών.
- Υποστήριξη για **FLAC** αναπαραγωγή ήχου.
- Η υψηλή απόδοση κλειδώματος του Wi-Fi διατηρεί τις Wi-Fi συνδέσεις σε υψηλό επίπεδο, όταν η οθόνη της συσκευής είναι απενεργοποιημένη.
- Υποστήριξη **HTTP proxy** για κάθε συνδεδεμένο **Wi-Fi access point**.

### **Έκδοση 3.2 (Honeycomb)**

Στις 15 Ιουλίου 2011 κυκλοφόρησε η 3.2 έκδοση με μερικές ακόμη αλλαγές:

- Βελτιωμένη υποστήριξη υλικού, συμπεριλαμβανομένων ορισμένων βελτιστοποιήσεων με σκοπό την κάλυψη ενός ευρύτερου φάσματος tablet συσκευών.
- Ευκολότερη πρόσβαση των εφαρμογών σε αρχεία που βρίσκονται στην κάρτα SD.
- Συμβατότητα στη λειτουργία της προβολής για εφαρμογές που δεν έχουν ακόμη βελτιστοποιηθεί ώστε να υποστηρίξουν την ανάλυση οθόνης των tablets.
- Νέες λειτουργίες υποστήριξης οθόνης που δίνουν στους προγραμματιστές περισσότερο έλεγχο σε διαφορετικές συσκευές Android.

### **Έκδοση 3.2.1 (Honeycomb)**

Η έκδοση 3.2.1 κυκλοφόρησε στις 20 Σεπτεμβρίου 2011 περιλαμβάνοντας ορισμένες ακόμη τροποποιήσεις:

- Διορθώσεις σφαλμάτων ασφάλειας, σταθερότητας και βελτιώσεις στο Wi-Fi.
- Αναβάθμιση του Android Market με δυνατότητα αυτόματου update και ευκολότερα στην ανάγνωση κείμενα συνθηκών κατά την εγκατάσταση μιας εφαρμογής.
- Αναβάθμιση των **Google Books**.
- Βελτιώσεις στο στοιχείο Adobe Flash του browser.
- Βελτιώσεις στην πρόβλεψη λέξεων του κινέζικου αλφαβήτου.

### **1.3.7 Android 4.0 Ice Cream Sandwich**

Η έκδοση 4.0 Ice Cream Sandwich κυκλοφόρησε στις 19 Οκτωβρίου του 2011 και περιλαμβάνει:



Εικόνα 1.9.Λογότυπο Android Ice Cream Sandwich

- Εικονικά κουμπιά στο UI.
- Διαχωρισμός των widgets.
- Ευκολότερη δημιουργία φακέλων, με τη βοήθεια ενός απλού drag and drop.
- Προσαρμόσιμο launcher.
- Βελτιωμένος τηλεφωνητής.
- Λειτουργικότερη εφαρμογή ημερολογίου.
- Νέες δυνατότητες του Gmail, όπως δυνατότητα αναζήτησης εκτός σύνδεσης.
- Ευκολότερη εναλλαγή συνομιλιών στο Gmail.
- Βελτιωμένη διόρθωση λαθών στο πληκτρολόγιο.
- Δυνατότητα πρόσβασης σε εφαρμογές απευθείας από την οθόνη κλειδώματος.
- Βελτιωμένες λειτουργίες αντιγραφής – επικόλλησης.
- Υποστήριξη face unlock, που επιτρέπει στους χρήστες να ξεκλειδώνουν τις συσκευές μέσω λογισμικού αναγνώρισης προσώπου.

- Υποστήριξη μέχρι και 16 καρτελών στον web browser.
- Αυτόματος συγχρονισμός του browser με τους σελιδοδείκτες του chrome.
- Υποστήριξη της μοντέρνας γραμματοσειράς **Roboto**.
- Ύπαρξη τομέα «Χρήσης Δεδομένων» στις ρυθμίσεις, που προειδοποιεί τους χρήστες όταν πλησιάζουν ένα συγκεκριμένο όριο χρήσης των δεδομένων και απενεργοποιεί τα δεδομένα όταν παρατηρείται υπέρβαση του εν λόγω ορίου.
- Δυνατότητα να τερματίζονται οι εφαρμογές που χρησιμοποιούν δεδομένα στο background.
- Βελτιωμένη εφαρμογή κάμερας.
- Ενσωματωμένο πρόγραμμα επεξεργασίας φωτογραφιών.
- Νέο layout στη γκαλερί.
- Ανανεωμένη εφαρμογή People.
- Επιτάχυνση hardware του UI.
- Δυνατότητα αλλαγής μεγέθους των widgets πλέον και στα κινητά τηλέφωνα.
- Υποστήριξη τεχνολογίας **Wi-Fi Direct**, κατά την οποία επιτρέπεται σε ασύρματες συσκευές να επικοινωνούν μεταξύ τους χωρίς την ανάγκη για ασύρματα σημεία πρόσβασης.

### **1.3.8 Android 4.1 /4.2 Jelly Beans**



Εικόνα 1.10.Λογότυπο Android Jelly Beans

Η **Google** παρουσίασε στην Google I/O 27 Ιουνίου 2012 την επόμενη έκδοση του λειτουργικού συστήματος Android, το Android 4.1 Jelly Bean, με πολλές βελτιώσεις, προσθήκες και λειτουργίες που αποδεικνύουν ότι η εταιρία δεν πρόκειται να αποδειχτεί εύκολος αντίπαλος για την Apple.

Η έκδοση αυτή έχει μερικά πραγματικά ενδιαφέροντα νέα χαρακτηριστικά να προσφέρει στον χρήστη του :

- **Project Butter** , τελευταία καινοτομία στο Jelly Bean, χάριν στην οποία τριπλασιάζεται το buffering των γραφικών (60fps), με αποτέλεσμα ομαλότερη και ταχύτερη απεικόνιση κατά την πλοήγηση στο UI του συστήματος.
- Υποστηρίζει το **wireless display**, το οποίο επιτρέπει το διαμοιρασμό της εικόνας της συσκευής με μία HDTV, η οποία μπορεί να συνδεθεί με έναν ειδικό αντάπτορα και να μετατραπεί σε δεύτερη ασύρματα συνδεδεμένη οθόνη.
- Το **daydream** από την μεριά του είναι ένα χαρακτηριστικό που κάνει το Android να προβάλλει όμορφες εικόνες, ή άλλες χρήσιμες πληροφορίες όταν είναι σε κατάσταση αναμονής.
- Τα **notifications** έχουν ανανεωθεί και είναι όλα σε ένα σημείο συγκεντρωμένα, επιτρέποντας την άμεση εκτέλεση των εργασιών από εκεί.
- Αναβάθμιση υπάρχει τόσο για το **Google Now**, όσο και για το **Google Search**, με το πρώτο να έχει γίνει πιο έξυπνο να ανταποκρίνεται αυτόματα στις συνήθειες και τις ανάγκες σας, ενώ το δεύτερο έχει ενσωματώσει το Google Knowledge Graph για να παίρνετε όλο τον πλούτο των πληροφοριών που ζητάτε ανά πάσα στιγμή.
- Εύκολος διαμοιρασμός φωτογραφιών και video με το **Android Beam**.
- Η πληκτρολόγηση είναι πιο εύκολη με το **Swype**.
- Πανοραμικές φωτογραφίες με την χρήση του **Photo Spheres**.

### **1.3.9 Android 5.0 Key Lime Pie**



Εικόνα 1.11.Λογότυπο Android Key Lime Pie

Η Google δεν έχει κυκλοφορήσει μέχρι στιγμής καμία επίσημη πληροφορία για την έκδοση 5.0 Key Lime Pie. Η παρουσίαση της νέας έκδοσης θα πραγματοποιηθεί τον Μάιο του 2013.

## **1.4 Εφαρμογές και Widgets**

Στο υψηλότερο επίπεδο της στοίβας Android, φιγουράρουν οι εφαρμογές και τα widgets. Αυτό είναι που βλέπουν οι χρήστες χωρίς να γνωρίζουν την όλη από κάτω διαδικασία. Αυτές είναι εφαρμογές που γράφουν οι κατασκευαστές λογισμικού, εφαρμογές που ήδη είναι εγκατεστημένες στο τηλέφωνο ή που ο χρήστης παίρνει από το Android Market. Οι εφαρμογές είναι προγράμματα που καταλαμβάνουν ολόκληρη την οθόνη και αλληλεπιδρούν με το χρήστη. Από την άλλη τα widget λειτουργούν σε μικρά τετράγωνα μέσα στην αρχική οθόνη – εφαρμογή

## **1.5 Γιατί Android;**

Εν κατακλείδι, παρατίθενται εδώ και κάποια σημεία και σκέψεις για το Android, σαν μία τελευταία απόδειξη της πρωτοπορίας του στο χώρο και της ισχυρής δυναμικής του. Γιατί Android λοιπόν ;

- Είναι μια πραγματικά ανοιχτή, ελεύθερη πλατφόρμα ανάπτυξης, βασισμένη στο Linux.
- Διαθέτει αρχιτεκτονική βασισμένη σε δομικά στοιχεία τα οποία μπορούν να τροποποιηθούν, να ολοκληρωθούν και να προσαρμοστούν στις ανάγκες κάθε κατασκευαστή και κατά συνέπεια χρήστη.
- Πάμπολλες ενσωματωμένες υπηρεσίες που μπορούν να κάνουν την εμπειρία του χρήστη μοναδική, όπως υπηρεσίες βασισμένες στην τοποθεσία, πανίσχυρη SQL βάση δεδομένων, μηχανή αναζήτησης και χάρτες.
- Αυτόματη διαχείριση του κύκλου ζωής μιας εφαρμογής, με πολλαπλές δικλείδες ασφαλείας ανάμεσα στα προγράμματα. Βελτιστοποιήσεις στον τομέα διαχείρισης μνήμης και χαμηλής κατανάλωσης σε τέτοιο βαθμό που δεν έχει ξανασυναντηθεί σε άλλο smartphone.

- Υψηλής ποιότητας γραφικά και ήχου.
  
- Φορητότητα ανάμεσα σε ένα ευρύ φάσμα ήδη υπάρχοντος υλικού αλλά και μελλοντικού. Αυτό έρχεται σαν απόρροια του γεγονότος ότι όλα τα προγράμματα γράφονται σε Java και εκτελούνται από την εικονική μηχανή Dalvik. Επιπρόσθετα οι οθόνες μπορούν να τροποποιηθούν κατάλληλα για να υποστηρίξουν οποιαδήποτε ανάλυση, μέγεθος και προσανατολισμό οθόνης.

Αυτοί είναι μερικοί μόνο από τους λόγους για τους οποίους έγινε η επιλογή του Android ως πλατφόρμα ανάπτυξης σε αυτή την πτυχιακή εργασία. Ο οποιοσδήποτε χρήστης θα μπορούσε να επαληθεύσει όλα τα παραπάνω, χρησιμοποιώντας απλά ένα τηλέφωνο ή μια ταμπλέτα που τρέχει το λειτουργικό αυτό σύστημα.

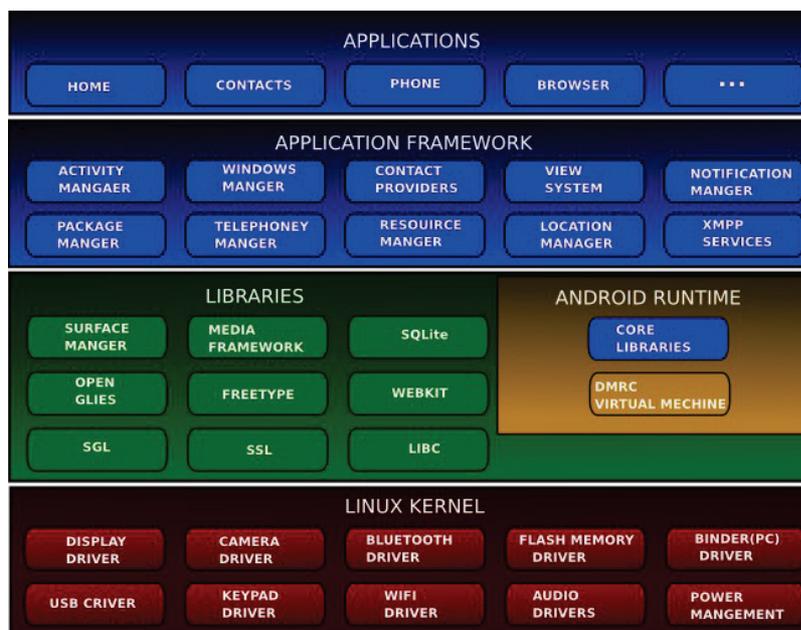
## 2.Αρχιτεκτονική Android

Όπως αναφέραμε πρίν, το Android είναι μια στοίβα λογισμικού. Η αρχιτεκτονική του Android ενθαρρύνει την έννοια επαναχρησιμοποίησης κώδικα, που επιτρέπει να δημοσιευτούν και να μοιραστούν δραστηριότητες, υπηρεσίες και δεδομένα με άλλες εφαρμογές με πρόσβαση που ελέγχεται από τους περιορισμούς ασφάλειας που έχουν τεθεί. Έτσι γίνεται δυνατή η επέκταση και βελτιστοποίηση υπαρχόντων εφαρμογών ή δημιουργίας καινούργιων, χρησιμοποιώντας κώδικα από αυτές.

Ο ακόλουθος πίνακας παρουσιάζει τα κυριότερα συστατικά του λειτουργικού συστήματος Android. Κάθε τμήμα περιγράφεται αναλυτικότερα παρακάτω.

Από ότι βλέπουμε λοιπόν η αρχιτεκτονική του λειτουργικού συστήματος αποτελείτε από 5 βασικά επίπεδα.

- Τις εφαρμογές (Applications ).
- Το πλαίσιο εφαρμογής (Application Framework).
- Τον χρόνο εκτέλεσης (Android Runtime) και Την εικονική μηχανή Dalvik (Dalvik VM).
- Τις εγγενείς και τις προηγμένες βιβλιοθήκες (Libraries).
- Τον πυρήνα Linux (Linux Kernel).



Εικόνα 2.1. Τα βασικά περιεχόμενα του λειτουργικού συστήματος Android

## **2.1 Application**

Το Android κυκλοφορεί με ένα σύνολο βασικών εφαρμογών , περιλαμβάνοντας ένα email client , μια εφαρμογή SMS, ημερολόγιο, χάρτες, φυλλομετρητή, επαφές κ.α. Η δημιουργία των εφαρμογών γίνεται με την χρήση της γλώσσας προγραμματισμού JAVA.

## **2.2 Πυρήνας Linux - Linux Kernel**

Η βάση της στοίβας λογισμικού του Android είναι ο πυρήνας Linux. Ο τροποποιημένος πυρήνας του συστήματος βασίζεται στην έκδοση 2.6 (και στην έκδοση 3.0.1 για το Android 4.0) του Linux Kernel, η οποία υποστηρίζει όλες τις κύριες λειτουργίες του λειτουργικού συστήματος. Οι λειτουργίες αυτές αφορούν διαχείριση μνήμης, διαχείριση διεργασιών, λειτουργίες δικτύου, ασφάλεια του λειτουργικού, και ένα σύνολο οδηγών υλικού (hardware drivers).

Οι drivers αυτοί είναι υπεύθυνοι για την επικοινωνία του software με το hardware της συσκευής. Ενδεικτικά ο πυρήνας του Android περιέχει driver προβολής οθόνης, Wifi και Bluetooth, driver κάμερας κλπ.

Ο πυρήνας του Android μπορεί να βασίζεται στον πυρήνα του Linux, αλλά διαφέρει αρκετά από αυτόν. Ο λόγος είναι οι αλλαγές στην αρχιτεκτονική που έχει κάνει η Google για να είναι ελαφρύτερος και βελτιστοποιημένος για χρήση σε κινητές συσκευές. Αυτό σημαίνει ότι παρότι το Android είναι κατά βάση Linux, επί της ουσίας είναι αρκετά δύσκολο να τρέξουν εφαρμογές ή να χρησιμοποιηθούν βιβλιοθήκες από τη μία πλατφόρμα στην άλλη. Ο Linus Torvalds έχει αναφέρει ότι τελικά στο μέλλον το Android και το Linux θα μοιράζονται έναν κοινό πυρήνα, αλλά αυτό θα αργήσει 4-5 χρόνια ακόμα.

## **2.3 Βιβλιοθήκες – Libraries**

Στο δεύτερο επίπεδο της στοίβας έχουμε τις βιβλιοθήκες του Android. Αυτές ουσιαστικά αποτελούν τα APIs που είναι διαθέσιμα στους προγραμματιστές για την ανάπτυξη των εφαρμογών. Οι βιβλιοθήκες από μόνες τους δεν αποτελούν εφαρμογές αλλά ενσωματώνονται και χρησιμοποιούνται από τις εφαρμογές για τις διάφορες λειτουργίες που παρέχει η καθεμία από αυτές. Ουσιαστικά αποτελούν ένα από τα δομικά υλικά των εφαρμογών, και άρα είναι αναπόσπαστο κομμάτι τους. Οι δυνατότητες των βιβλιοθηκών του Android γίνονται εμφανείς στους προγραμματιστές στην στοίβα του πλαισίου εφαρμογής. Όλες οι συσκευές Android υποστηρίζουν το λιγότερο τα παρακάτω APIs :

- android.util - Το πακέτο βοηθήματος (utility) πυρήνων περιέχει χαμηλού επιπέδου κλάσεις, όπως τα εξειδικευμένα containers, μορφοποιητές (formatters) συμβολοσειρών, και βοηθήματα για parsing XML αρχείων.
- android.os - Το πακέτο λειτουργικού συστήματος παρέχει την πρόσβαση στις βασικές υπηρεσίες του λειτουργικού συστήματος όπως τη διαβίβαση μηνυμάτων, την επικοινωνία μεταξύ των επικοινωνιών, τις λειτουργίες ρολογιών, και την αποσφαλμάτωση (debugging).
- android.graphics - Το API γραφικών παρέχει τις χαμηλού επιπέδου κλάσεις γραφικών που υποστηρίζουν τους καμβάδες (canvases), τα χρώματα, και τα βασικά στοιχεία ζωγραφικής, και παρέχει τη δυνατότητα να ζωγραφίσει κανείς πάνω σε καμβά (canvas).
- android.text - Τα εργαλεία επεξεργασίας κειμένων για αναπαράσταση και ανάλυση κειμένου.
- android.database - Παρέχει τις χαμηλού επιπέδου κλάσεις που απαιτούνται για το χειρισμό των δρομέων (cursors) κατά τη λειτουργία τους με τις βάσεις δεδομένων.
- android.content - Το API περιεχομένου (content) χρησιμοποιείται για να διαχειριστεί την πρόσβαση στα δεδομένα και την έκδοσή τους, παρέχοντας υπηρεσίες για την διαχείριση των πόρων, των Παρόχων Περιεχομένου (Content Provider), και των πακέτων.
- android.view – Οι Όψεις (Views) είναι οι κλάσεις πυρήνα διεπαφών χρήστη. Όλες οι διεπαφές χρήστη κατασκευάζονται χρησιμοποιώντας μια σειρά από Όψεις που παρέχουν τα συστατικά της αλληλεπίδρασης χρηστών.
- android.widget - Τοποθετημένες στο πακέτο View, οι κλάσεις widget είναι τα στοιχεία της διεπαφής χρήστη (user-interface) που χρησιμοποιούνται στις εφαρμογές. Περιλαμβάνουν τις λίστες, τα κουμπιά, και τις διατάξεις (layouts).
- com.google.android.maps - Ένα υψηλού επιπέδου API που παρέχει πρόσβαση στις εγγενείς λειτουργίες χαρτών που μπορούν να χρησιμοποιηθούν μέσα σε εφαρμογές. Περιλαμβάνει τη λειτουργία MapView, καθώς επίσης και τις κλάσεις Overlay και MapController που χρησιμοποιούνται για να διαχειριστούν και να ελέγξουν τους ενσωματωμένους χάρτες.
- android.app - Ένα πακέτο υψηλού επιπέδου που παρέχει πρόσβαση στο μοντέλο εφαρμογών. Το πακέτο εφαρμογών περιλαμβάνει τα Activity και Service APIs που αποτελούν τη βάση για όλες τις Android εφαρμογές.
- android.provider - Για να διευκολύνει την πρόσβαση των προγραμματιστών σε ορισμένους τυποποιημένους Παρόχους Περιεχομένου (Content Providers) (όπως η βάση δεδομένων επαφών), προσφέρονται κλάσεις που παρέχουν πρόσβαση σε τυποποιημένες βάσεις δεδομένων που περιλαμβάνονται σε όλες τις εκδόσεις Android.
- android.telephony - Τα APIs τηλεφωνίας επιτρέπουν την άμεση αλληλεπίδραση με το τηλεφωνικό σωρό (phone stack) της συσκευής, δίνοντας τη δυνατότητα, να γίνει λήψη, και έλεγχος των τηλεφωνημάτων, της κατάστασης του τηλεφώνου, και των μηνυμάτων SMS.

- android.webkit - Το πακέτο WebKit περιλαμβάνει APIs για την εργασία με περιεχόμενο βασισμένο στο διαδίκτυο, συμπεριλαμβανομένης της λειτουργίας wampree για την ενσωμάτωση μηχανών αναζήτησης και ενός διαχειριστή για cookies.

Εκτός από το Android APIs, ο σωρός του Android περιλαμβάνει ένα σύνολο βιβλιοθηκών C/C++. Αυτές οι βιβλιοθήκες περιλαμβάνουν:

- OpenGL - Βιβλιοθήκη που χρησιμοποιείται για να υποστηρίξει 3D γραφικά βασισμένη στο Open GL ES 1.0 API.
- FreeType - Υποστήριξη για τη bitmap και τη vector απόδοση γραμματοσειράς.
- SGL - Βιβλιοθήκη πυρήνων που χρησιμοποιείται για να παρέχει μια μηχανή 2D γραφικών .
- libc - Τυποποιημένη βιβλιοθήκη C που βελτιστοποιείται για συσκευές βασισμένες σε Linux.
- SQLite - Ελαφριά μηχανή σχεσιακών βάσεων δεδομένων που χρησιμοποιείται για να αποθηκεύσει δεδομένα εφαρμογών.
- SSL - Υποστήριξη για χρήση των Secure Sockets Layer του κρυπτογραφικού πρωτοκόλλου για ασφαλείς επικοινωνίες μέσω ιαδικτύου.
- Surface Manager – διαχειρίζεται την πρόσβαση στο υποσύστημα προβολής, και συνθέτει απρόσκοπτα δισδιάστατα και τρισδιάστατα επίπεδα γραφικών τα οποία προέρχονται από πολλαπλές εφαρμογές.
- LibWebCore – μια μοντέρνα μηχανή υποστήριξης πλοήγηση στο διαδίκτυο (browser engine) η οποία χρησιμοποιείτε και από τον ενσωματωμένο browser του Android αλλά και από τις WebViews που ενσωματώνονται στις εφαρμογές.

### **2.3.1 Προηγμένες Βιβλιοθήκες Android**

Οι βιβλιοθήκες πυρήνα παρέχουν όλες τις λειτουργίες που είναι απαραίτητες για να ξεκινήσει η δημιουργία μιας εφαρμογής για Android, αλλά οι προηγμένες βιβλιοθήκες APIs προσφέρουν πραγματικά συναρπαστικές λειτουργίες.

Επειδή το Android στοχεύει να χρησιμοποιείται από ένα ευρύ φάσμα κινητών, η καταλληλότητα και η εφαρμογή των ακόλουθων APIs ποικίλουν ανάλογα με τη συσκευή επάνω στην οποία εφαρμόζονται.

- android.location -Το τοποκεντρικό API (location-based) δίνει πρόσβαση των εφαρμογών σε πληροφορίες για τη τρέχουσα φυσική θέση της συσκευής. Οι υπηρεσίες βασισμένες στη τοποθεσία παρέχουν πρόσβαση σε πληροφορίες τοποθεσίας χρησιμοποιώντας οποιοδήποτε υλικό ή τεχνολογία είναι διαθέσιμη στη συσκευή και βοηθούν στον προσδιορισμό θέσης.
- android.media -Τα APIs πολυμέσων παρέχουν υποστήριξη για την αναπαραγωγή και εγγραφή αρχείων ήχου και βίντεο, συμπεριλαμβανομένων των πολυμέσων ροής

(streamed).

- android.opengl -Το Android προσφέρει μια ισχυρή αποδοτική 3D μηχανή που χρησιμοποιεί το OpenGL ES API με το οποίο δημιουργούνται 3D διεπαφές χρήστη για τις εφαρμογές.
- android.hardware - Όπου είναι διαθέσιμο, το API υλικού εκθέτει υλικό αισθητήρων συμπεριλαμβανομένης της φωτογραφικής μηχανής, του επιταχύμετρου (accelerometer) και των αισθητήρων πυξίδας. .
- android.bluetooth, android.net.wifi, και android.telephony - Το Android επίσης παρέχει χαμηλού επιπέδου πρόσβαση στην πλατφόρμα υλικού, συμπεριλαμβανομένου του Bluetooth, του WI-FI, και του υλικού τηλεφωνίας.

## **2.4 Χρόνος Εκτέλεσης – Android Runtime**

Στο ίδιο επίπεδο με τις εγγενείς βιβλιοθήκες, βρίσκουμε και τον χρόνο εκτέλεσης Android. Εδώ υπάρχουν οι βασικές βιβλιοθήκες της Java και η εικονική μηχανή Dalvik. Η Dalvik είναι μια βελτιστοποιημένη υλοποίηση μιας εικονικής μηχανής Java για φορητές συσκευές από την Google. Η Dalvik τρέχει .dex αρχεία, τα οποία είναι bytecodes που προέρχονται από αρχεία .class και .jar. Εν αντιθέσει όμως με τα .class αρχεία, τα .dex είναι πολύ πιο συμπαγή και αποδοτικά, γεγονός σημαντικό για συσκευές με περιορισμένη μνήμη και μπαταρία. Το Android περιλαμβάνει ένα σύνολο βασικών βιβλιοθηκών που παρέχουν τις περισσότερες από τις διαθέσιμες λειτουργίες των βασικών βιβλιοθηκών της Java. Κάποια πακέτα και κλάσεις υπάρχουν και στο Android, κάποια άλλα δεν υποστηρίζονται καθόλου, ενώ ταυτόχρονα το Android παρέχει και επιπρόσθετα προσαρμοσμένα στις δικές του ανάγκες.

## **2.5 Εικονική Μηχανή Dalvik**

Σχεδόν το σύνολο των APIs του Android βασίζονται στη γλώσσα προγραμματισμού Java. Στην Java ως γνωστόν υπάρχει η λεγόμενη Java Virtual Machine στην οποία εκτελείτε ο κώδικας bytecode των εφαρμογών. Στο Android υπάρχει κάτι παρόμοιο και δεν είναι άλλο από την εικονική μηχανή Dalvik μέσω της οποίας τρέχουν οι εφαρμογές του Android.

Η Dalvik VM (Virtual Machine) χρησιμοποιεί τον πυρήνα Linux της συσκευής για να χειριστεί τις χαμηλού επιπέδου λειτουργίες που περιλαμβάνουν την ασφάλεια, τον πολυνηματισμό και τη διαχείριση διαδικασιών και μνήμης. Είναι επίσης δυνατό να γραφτούν εφαρμογές C/C++ που τρέχουν άμεσα στο εσωτερικό του λειτουργικού Linux. Αν και μπορεί να γίνει αυτό, στις περισσότερες περιπτώσεις δεν υπάρχει κανένας λόγος.

Μέσω της Dalvik VM επιτυγχάνεται η ρύθμιση της πρόσβασης στο υλικό και στις υπηρεσίες του συστήματος. Με τη χρησιμοποίηση αυτής της εικονικής μηχανής στην εκτέλεση εφαρμογής, η οποία προσφέρει ένα αφαιρετικό στρώμα, οι κατασκευαστές δεν χρειάζεται να ανησυχήσουν για κάποια υλοποίηση υλικού (hardware implementation).

Η Dalvik VM εκτελεί τα Dalvik εκτελέσιμα αρχεία, ένα format βελτιστοποιημένο έτσι ώστε να καταλαμβάνει την ελάχιστη μνήμη. Τα .dex εκτελέσιμα αρχεία δημιουργούνται μετασχηματίζοντας κλάσεις που έχουν μεταγλωττιστεί από Java χρησιμοποιώντας εργαλεία που παρέχονται μέσα στο SDK.

Μια απλή Java VM είναι μια εικονική μηχανή βασισμένη σε στοίβα (stack-based). Η Dalvik VM από την άλλη είναι μια εικονική μηχανή βασισμένη σε μητρώα (registerbased). Με τον τρόπο αυτό αυξάνεται η αποδοτικότητα του επεξεργαστή του κινητού. Επίσης, οι εικονικές μηχανές που είναι βασισμένες σε καταχωρητές (registers) επιτρέπουν ταχύτερους χρόνους εκτέλεσης των μεγάλων προγραμμάτων.

## **2.6 Πλαίσιο Εφαρμογής – App Framework**

Πάνω από τις εγγενής βιβλιοθήκες και το χρόνο εκτέλεσης Android, είναι το πλαίσιο εφαρμογής, App Framework. Αυτό το επίπεδο μας παρέχει υψηλού επιπέδου δομικές μονάδες τις οποίες μπορούμε να χρησιμοποιούμε για την κατασκευή των εφαρμογών μας. Αυτό το πλαίσιο είναι προ-εγκατεστημένο στο Android, αλλά είναι επεκτάσιμο, αφού ο κάθε κατασκευαστής μπορεί να το συμπληρώσει με δικά του κομμάτια.

Τα σημαντικότερα δομικά στοιχεία του πλαισίου αυτού είναι:

- Διαχειριστής Δραστηριοτήτων - Activity Manager: Υπεύθυνο για τον έλεγχο του χρόνου ζωής (Εικόνα 2.2) των εφαρμογών και για την διατήρηση μιας στοίβας που επιτρέπει την πλοήγηση του χρήστη σε προηγούμενες οθόνες.
- Παροχέας Περιεχομένου - Content Providers: Αυτά τα αντικείμενα περιέχουν δεδομένα που μπορούν να διαμοιραστούν μεταξύ εφαρμογών.
- Διαχειριστής Πόρων - Resource Manager: Οι πόροι, είναι οτιδήποτε υπάρχει σε ένα πρόγραμμα και δεν είναι κώδικας. Για παράδειγμα μπορεί να είναι κωδικοί χρωμάτων, αλφαριθμητικοί χαρακτήρες ή ακόμα και έτοιμα σχεδιαγράμματα οθονών φτιαγμένα σε XML, τα οποία μπορεί το πρόγραμμα να καλεί.
- Διαχειριστής Τοποθεσίας - Location Manager: Χρησιμοποιείται για να μπορεί να



## 2.7 Βασικά Συστατικά του Android

Οι εφαρμογές Android αποτελούνται από κάποια συστατικά, που συνδέονται χρησιμοποιώντας ένα XML αρχείο, το `AndroidManifest.xml` του κάθε project, το οποίο περιγράφει κάθε συστατικό και πώς αλληλεπιδρά με τα άλλα. Τα συστατικά που παρέχουν τις δομικές μονάδες για τις εφαρμογές είναι:

- *Activity (Δραστηριότητα)*
- *Service (Υπηρεσία)*
- *Content Provider (Παροχέας Περιεχομένου)*
- *Intent (Προθέσεις)*

### 2.7.1 Activity (Δραστηριότητα)

Τα Activities είναι τα πιο κοινά από τα 4 τμήματα που προαναφέρθηκαν. Μια δραστηριότητα είναι συνήθως μια απλή οθόνη της εφαρμογής. Κάθε δραστηριότητα υλοποιείται σαν μια κλάση που επεκτείνει (*extends*) την βασική κλάση *Δραστηριότητας* (*Activity base class*). Η δική σας κλάση θα προβάλει μια *διεπαφή χρήστη* (*user interface*) αποτελούμενη από *Εικόνες* (*Views*) και θα απαντά σε *Συμβάντα* (*Events*). Οι περισσότερες εφαρμογές αποτελούνται από πολλαπλές οθόνες. Για παράδειγμα, μια εφαρμογή ανταλλαγής γραπτών μηνυμάτων, θα μπορούσε να έχει μια οθόνη που δείχνει μια λίστα από τις επαφές για να διαλέξετε σε ποιον θα στείλετε το μήνυμα, μια δεύτερη οθόνη για να γράφετε το μήνυμα στην επιλεγμένη επαφή και άλλες οθόνες για να βλέπετε παλιά μηνύματα ή να αλλάζετε τις ρυθμίσεις. Κάθε μια από αυτές τις οθόνες θα υλοποιούτα σαν μια δραστηριότητα. Η μετάβαση σε άλλη οθόνη επιτυγχάνεται με την έναρξη μιας νέας δραστηριότητας. Σε μερικές περιπτώσεις μια δραστηριότητα μπορεί να επιστρέφει μια τιμή σε μια προηγμένη δραστηριότητα – για παράδειγμα μια δραστηριότητα που επιτρέπει στον χρήστη να επιλέξει μια φωτογραφία θα επέστρεφε την επιλεγμένη φωτογραφία στην δραστηριότητα που την κάλεσε.

Όταν μια νέα οθόνη ανοίγει, η πορηγούμενη οθόνη μπαίνει σε μια στοίβα που κρατά το ιστορικό (*history stack*). Ο χρήστης μπορεί να πλοηγηθεί πίσω σε οθόνες που άνοιξε προηγουμενος μέσω του ιστορικού. Οι οθόνες μπορούν επίσης να αφαιρεθούν από το ιστορικό σε περιπτώσεις που δεν χρειάζεται να παραμείνουν προσβάσιμες. Το Android διατηρεί ιστορικό για κάθε εφαρμογή που εκκίνησε από την κεντρική οθόνη (*home screen*).

### 2.7.2 **Service (Υπηρεσία)**

Μια Υπηρεσία είναι κώδικας που τρέχει για μεγάλο χρονικό διάστημα και χωρίς διεπαφή χρήστη (UI), ενημερώνοντας τις πηγές δεδομένων και τις ορατές δραστηριότητες (Activities), ενεργοποιώντας Ειδοποιήσεις (Notifications). Αυτές τρέχουν και επεξεργάζονται δεδομένα ακόμα και όταν οι δραστηριότητες της εφαρμογής δεν είναι ενεργές ή ορατές. Ένα καλό παράδειγμα είναι μια εφαρμογή που αναπαράγει μουσική από μια λίστα μουσικών κομματιών (media player). Σε μια τέτοια εφαρμογή, θα υπήρχαν κατά πάσα πιθανότητα μία ή και παραπάνω δραστηριότητες που επιτρέπουν στον χρήστη να επιλέξει τραγούδια και να τα αναπαράγει. Ωστόσο, η αναπαραγωγή από μόνη της δεν θα έπρεπε να διαχειρίζεται από την δραστηριότητα γιατί ο χρήστης θα περίμενε την μουσική να συνεχίζει να παίζει ακόμη και μετά την πλοήγησή του σε μια νέα οθόνη. Σε αυτή τη περίπτωση, η δραστηριότητα της αναπαραγωγής μουσικής θα ξεκινούσε μια Υπηρεσία (Service) για να τρέξει στο παρασκήνιο και να συνεχίσει η μουσική να παίζει. Το σύστημα τότε θα κρατά την Υπηρεσία αναπαραγωγής ενεργή μέχρι να τελειώσει το κομμάτι. Όταν πραγματοποιηθεί σύνδεση σε μια Υπηρεσία, μπορεί να υπάρξει επικοινωνία με αυτή μέσω μιας διεπαφής που προσφέρεται από την Υπηρεσία. Για την Υπηρεσία μουσικής, αυτό θα επέτρεπε την παύση ή την επιστροφή πίσω στο κομμάτι (rewind) κλπ.

### 2.7.3 **Content Provider (Παροχέας Περιεχομένου)**

Παροχέας Περιεχομένου (Content Provider) - Οι εφαρμογές μπορούν να σώζουν τα δεδομένα τους σε αρχεία, σε μια βάση δεδομένων SQLite ή με οποιοδήποτε άλλο μηχανισμό μπορούν. Ένας Παροχέας Περιεχομένου, ωστόσο, είναι χρήσιμος ώστε τα δεδομένα μιας εφαρμογής να είναι διαθέσιμα και σε άλλες εφαρμογές. Ένας Παροχέας Περιεχομένου είναι μια κλάση που υλοποιεί μια συγκεκριμένη ομάδα μεθόδων που επιτρέπουν σε άλλες εφαρμογές να αποθηκεύουν και να επανακτούν δεδομένα του τύπου που διαχειρίζεται ο Παροχέας Περιεχομένου.

Οι συσκευές Android περιλαμβάνουν διάφορους εγγενείς Παροχείς Περιεχομένου (Content Providers) που εκθέτουν τις χρήσιμες βάσεις δεδομένων, όπως για παράδειγμα των στοιχείων των επαφών του χρήστη.

## 2.7.4 Intent (Προθέσεις)

Το Android χρησιμοποιεί μια ειδική κλάση που λέγεται Πρόθεση (Intent) για να κινείται από οθόνη σε οθόνη. Η Πρόθεση περιγράφει τι θέλει η εφαρμογή να γίνει στη συνέχεια. Τα δυο πιο σημαντικά μέρη της δομής δεδομένων της Πρόθεσης είναι η δράση (Action) και τα δεδομένα βάσει των οποίων αυτή θα εκτελεστεί. Τυπικές τιμές για μια δράση είναι η MAIN (η κεντρική είσοδος της εφαρμογής), VIEW, PICK, EDIT κλπ. Τα δεδομένα εκφράζονται ως URI (Uniform Resource Indicator). Για παράδειγμα, για να εμφανιστεί μια ιστοσελίδα στον φυλλομετρητή (browser), δημιουργείται μια Πρόθεση με δράση VIEW και τα δεδομένα ως ένα URL

Υπάρχει μια σχετική κλάση που λέγεται Φίλτρο Πρόθεσης (IntentFilter). Ενώ μια Πρόθεση είναι στην ουσία ένα αίτημα για να γίνει κάτι, το Φίλτρο Πρόθεσης είναι μια περιγραφή του τι είναι δυνατόν να διαχειριστεί ένας έκτης Πρόθεσης (Intent Receiver). Μια ραστηριότητα που είναι σε θέση να προβάλει πληροφορίες επικοινωνίας για ένα άτομο, θα ανακοινώνει με ένα Φίλτρο Πρόθεσης (IntentFilter) ότι γνωρίζει πως να διαχειριστεί την VIEW\_ACTION όταν τα δεδομένα αντιπροσωπεύουν ένα άτομο. Οι ραστηριότητες ανακοινώνουν τα Φίλτρα Πρόθεσης (IntentFilters) στο AndroidManifest.xml αρχείο.

Η πλοήγηση από οθόνη σε οθόνη πετυχαίνεται με Προθέσεις. Για να πλοηγηθεί κανείς προς τα μπρος, μια δραστηριότητα καλεί την startActivity(myIntent). Το σύστημα τότε κοιτά στα Φίλτρα Προθέσεων (Intent Filters) για όλες τις εγκατεστημένες εφαρμογές και διαλέγει την δραστηριότητα που τα Φίλτρα Πρόθεσης ταιριάζουν καλύτερα με την παράμετρο 'MyIntent' της κλήσης. Η νέα δραστηριότητα ενημερώνεται για την Πρόθεση και ξεκινά. Η διαδικασία της υλοποίησης των Προθέσεων συμβαίνει κατά τον χρόνο εκτέλεσης της εφαρμογής, όταν δηλαδή καλείται η startActivity, πράγμα που προσφέρει 2 πλεονεκτήματα – κλειδιά:

1. Οι δραστηριότητες μπορούν να επαναχρησιμοποιούν κάποια λειτουργικότητα από άλλα τμήματα του κώδικα απλά κάνοντας ένα αίτημα υπό την μορφή μιας πρόθεσης.
2. Οι δραστηριότητες μπορούν να αντικατασταθούν οποιαδήποτε στιγμή από μια νέα ραστηριότητα με ένα αντίστοιχο Φίλτρο Πρόθεσης.

➤ **Δέκτες Πρόθεσης (Intent Receiver)** - Χρησιμοποιούνται για να εκτελεστεί μια εφαρμογή σε απάντηση ενός εξωτερικού συμβάντος (external event), για παράδειγμα, όταν το τηλέφωνο χτυπά, ή όταν το δίκτυο είναι διαθέσιμο, ή όταν είναι μεσάνυχτα. Οι δέκτες Πρόθεσης δεν προβάλουν μια διεπαφή χρήστη (UI), ωστόσο μπορούν να προβάλουν Ειδοποιήσεις (Notifications) για να ειδοποιήσουν τον χρήστη για κάτι σημαντικό που συνέβη.

Οι δέκτες Πρόθεσης είναι επίσης καταχωρημένοι στο AndroidManifest.xml, αλλά μπορούν επίσης καταχωρηθούν από τον κώδικα χρησιμοποιώντας την Context.registerReceiver().

Η εφαρμογή δεν χρειάζεται να τρέχει για να κληθούν οι δέκτες Πρόθεσης που έχει. Το σύστημα θα κινήσει την εφαρμογή, αν χρειαστεί, όταν ένας δέκτης Πρόθεσης ενεργοποιηθεί. Οι εφαρμογές μπορούν επίσης να στέλνουν τις δικές τους Ανακοινώσεις Πρόθεσης (Intent Broadcasts) σε άλλους με την `Context.broadcastIntent()`.

- **Δέκτες Μετάδοσης (Broadcast Receivers)** - Βασική Κλάση για τον κώδικα, που θα λαμβάνει τις Προθέσεις (Intents) που στέλνονται από το `sendBroadcast ()`. Μπορεί είτε δυναμικά να καταχωρηθεί μια περίπτωση αυτής της κατηγορίας με `Context.registerReceiver()`, είτε να καταχωρηθεί ένα στιγμιότυπο αυτής της κλάσης σε μια εφαρμογή μέσω της ετικέτας `<receiver>` στο `AndroidManifest.xml`. Οι έκτες Μετάδοσης αρχίζουν αυτόματα την εφαρμογή αποκρινόμενοι σε μια εισερχόμενη Πρόθεση, που στην ουσία κάνει την εφαρμογή, μια εφαρμογή οδηγούμενη από τα γεγονότα. Υπάρχουν δύο σημαντικές κατηγορίες Μεταδόσεων (Broadcasts) που μπορούν να παραληφθούν:

- Οι Κανονικές Μεταδόσεις - Normal broadcasts (που στέλνονται με `Context.sendBroadcast`) είναι απολύτως ασύγχρονες. Όλοι οι δέκτες της Μετάδοσης οργανώνονται σε μια απροσδιόριστη διαταγή, συχνά συγχρόνως. Αυτό είναι αποδοτικότερο, αλλά σημαίνει ότι οι έκτες δεν μπορούν να χρησιμοποιήσουν το αποτέλεσμα ή να αποβάλουν APIs που συμπεριλαμβάνονται εδώ.
- Οι Μεταδόσεις διαταγής - Ordered broadcasts (που στέλνονται με `Context.sendOrderedBroadcast`) παραδίδονται σε έναν δέκτη τη φορά. Δεδομένου ότι κάθε δέκτης εκτελείται στη συνέχεια, μπορεί να διαδώσει ένα αποτέλεσμα στον επόμενο έκτη, ή μπορεί να αποβάλει τη Μετάδοση έτσι ώστε να μη περάσει σε άλλους δέκτες. Ο χειρισμός της εκτέλεσης των δεκτών διαταγής μπορεί να γίνει με την ιδιότητα του Φίλτρου Προθέσεων Ταιριάσματος (Matching Intent-Filter). Οι δέκτες με την ίδια προτεραιότητα τρέχουν με τυχαία σειρά.



### **3.Εργαλεία Ανάπτυξης Λογισμικού**

Μέσω του Android software development προκύπτουν ολοένα και νέες εφαρμογές που δημιουργούνται για το λειτουργικό σύστημα Android. Οι εφαρμογές αναπτύσσονται συνήθως στη γλώσσα προγραμματισμού Java που χρησιμοποιεί το Android. Από τον Απρίλιο του 2011, έχουν αναπτυχθεί για το Android πάνω από 200.000 εφαρμογές , με πάνω από 3 δισεκατομμύρια downloads. Η χρήση της πλατφόρμας Android έχει επίσης αυξηθεί, γιατί προτιμάται από τους προγραμματιστές των κινητών. Μια έρευνα τον Ιούνιο του 2011 έδειξε ότι πάνω από το 67% των κινητών χρησιμοποιούν εφαρμογές που αναπτύχθηκαν με χρήση της εν λόγω πλατφόρμας.

#### **3.1 Android SDK**

Το Android SDK παρέχει τα απαραίτητα εργαλεία για την ανάπτυξη προγραμμάτων χρησιμοποιώντας την γλώσσα προγραμματισμού Java. Τα χαρακτηριστικά του παρέχουν τεράστια ευελιξία και δυνατότητα ανάπτυξης έξυπνων εφαρμογών για κινητά τηλέφωνα, παρέχοντας δυνατότητες ανάπτυξης σε επιχειρηματίες όλων των κλάδων, είτε πρόκειται για αυτοματοποίηση πωλήσεων, ψυχαγωγία, παιχνίδια αλλά και κάθε άλλου είδους επιχειρήσεις. Ακόμα, παρέχει APIs για την χρήση web browser, εμφάνιση δισδιάστατων και τρισδιάστατων γραφικών, δομημένη αποθήκευση δεδομένων σε βάση δεδομένων, εμφάνιση πολυμεσικού υλικού (ήχος, βίντεο, εικόνες), χρήση των τεχνολογιών GSM, Bluetooth, EDGE, 3G και WiFi, χρήση συσκευών όπως φωτογραφική μηχανή, GPS, πυξίδα, επιταχυνσιόμετρο.

Ένα σημαντικό γεγονός που ευνοεί την ανάπτυξη εφαρμογών είναι πως το πακέτο Android SDK συνεργάζεται με το Eclipse και συνεπώς ο προγραμματιστής μπορεί εύκολα και γρήγορα να βλέπει τις αλλαγές του κώδικα στον emulator που του παρέχει το Android SDK, χωρίς να χρειάζεται να εξάγει κάθε φορά την εφαρμογή και να την εγκαθιστά σε κινητό. Επίσης, ο emulator είναι πολύ αξιόπιστος, καθώς έχει ακριβώς την ίδια συμπεριφορά με αυτή που θα είχε η εφαρμογή εάν είχε εγκατασταθεί σε ένα κινητό τηλέφωνο Android. Ένα άλλο σημαντικό πλεονέκτημα είναι το γεγονός πως σε περίπτωση αναβάθμισης του λογισμικού, η εφαρμογή εξακολουθεί να δουλεύει χωρίς την ανάγκη επανασχεδιασμού κάποιων σημαντικών κομματιών του κώδικα, που αφορούν την αλληλεπίδραση της εφαρμογής με τα δομικά μέρη-hardware- του κινητού τηλεφώνου. Παρακάτω, θα γίνει η ανάπτυξη του εξομοιωτή (emulator).

##### **3.1.1 Χαρακτηριστικά του SDK:**

Το Android SDK είναι ένα πολύ χρήσιμο εργαλείο για την εξερεύνηση των «ενδοτέρων» του Android. Παρακάτω, φαίνονται τα βασικότερα χαρακτηριστικά του.

- Εφαρμογή πλαισίου που επιτρέπει την επαναχρησιμοποίηση και την αντικατάσταση στοιχείων.
  - Βελτιστοποιημένη Dalvik εικονική μηχανή για κινητές συσκευές.
  - Ολοκληρωμένο πρόγραμμα περιήγησης, το οποίο βασίζεται στο open source WebKit
  - Βελτιστοποιημένα γραφικά, τα οποία τροφοδοτούνται από μια προσαρμοσμένη βιβλιοθήκη 2D και 3D γραφικών με βάση τις προδιαγραφές του OpenGL ES 1.0 (επιτάχυνση hardware προαιρετικά)
  - SQLite για δομημένη αποθήκευση δεδομένων
  - Υποστήριξη πολυμέσων για αρχεία ήχου, βίντεο, ακόμα και εικόνων (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
  - GSM Τηλεφωνία (εξαρτώμενη από το hardware)
  - Bluetooth, EDGE, 3G, WiFi (εξαρτώμενα από το hardware).
  - Φωτογραφική μηχανή, GPS, πυξίδα, και επιταχυνσιόμετρο (εξαρτώμενα από το hardware)
  - Πλούσιο περιβάλλον ανάπτυξης, συμπεριλαμβανομένου ενός εξομοιωτή συσκευής, εργαλεία για τον εντοπισμό σφαλμάτων, μνήμη και προφίλ απόδοσης όπως επίσης και ένα plugin για το Eclipse IDE.

### **3.1.2 Εξομοιωτής Emulator**

Προκειμένου να γίνει ευκολότερη η διαδικασία της ανάπτυξης και αποσφαλμάτωσης μιας εφαρμογής, το Android SDK περιλαμβάνει έναν εξομοιωτή μιας εικονικής κινητής συσκευής, η οποία τρέχει το λειτουργικό του Android. Έτσι δεν είναι η αναγκαία η ύπαρξη πραγματικής κινητής συσκευής για την εκτέλεση και δοκιμή των εφαρμογών. Ο εξομοιωτής προσομοιώνει ένα μεγάλο πλήθος λειτουργιών μιας τυπικής συσκευής, η οποία τρέχει το Android:

- Παρέχει μια ποικιλία πλήκτρων πλοήγησης και ελέγχου
- Παρέχει μια οθόνη για την προβολή των εφαρμογών που τρέχουν στον εξομοιωτή
- Επιτρέπει στις εφαρμογές την χρήση των υπηρεσιών που προσφέρει η πλατφόρμα του Android, δηλαδή την κλήση άλλων εφαρμογών, την πρόσβαση στο δίκτυο, την αναπαραγωγή ήχου και βίντεο, την αποθήκευση και επαναφορά δεδομένων, την ειδοποίηση χρήστη, το γραφικό περιβάλλον του Android.

Επίσης παρέχει ένα πλήθος λειτουργιών για την ευκολότερη αποσφαλμάτωση:

- Κονσόλα για την καταγραφή της εξόδου του πυρήνα
- Προσομοίωση διακοπών (όπως η άφιξη SMS μηνύματος ή τηλεφωνικής κλήσης)
- Προσομοίωση καθυστέρησης και απώλειας στο κανάλι δεδομένων

### **3.1.3 Άλλα εργαλεία του Android**

Το Android SDK περιλαμβάνει μερικά ακόμη εργαλεία για την ανάπτυξη εφαρμογών:

- Το Dalvik Debug Monitor Service (DDMS) το οποίο επιτρέπει την διαχείριση των διεργασιών στον εξομοιωτή ή στην συσκευή. Συγκεκριμένα δίνεται η δυνατότητα port-forwarding υπηρεσιών, λήψη screenshots, εμφάνιση πληροφοριών για τον σωρό και τα νήματα, logcat εμφάνιση πληροφοριών ράδιο και πληροφοριών διεργασιών, προσομοίωση εισερχόμενων κλήσεων και μηνυμάτων, προσομοίωση δεδομένων θέσης κ.α.
- Την Android Debug Bridge (ADB) η οποία επιτρέπει την διαχείριση της κατάστασης του εξομοιωτή ή της συσκευής. Μέσω του ADB είναι δυνατή η εκτέλεση εντολών φλοιού, η διαχείριση της προώθησης θυρών και η αντιγραφή από και προς την συσκευή ή τον εξομοιωτή.
- Το Android Asset Packaging Tool (AAPT) το οποίο δίνει την δυνατότητα δημιουργίας .apk αρχείων τα οποία περιέχουν τα εκτελέσιμα αρχεία και τους πόρους μιας εφαρμογής.
- Την Android Interface Description Language (AIDL) η οποία επιτρέπει την δημιουργία κώδικα, που επιτρέπει σε δύο διεργασίες σε μια συσκευή βασισμένη στο Android, να συνομιλούν χρησιμοποιώντας διαδιεργασιακή επικοινωνία.
- Το sqlite3 το οποίο επιτρέπει την πρόσβαση στα δεδομένα της SQLite που δημιουργούνται από τις διάφορες εφαρμογές.
- Το Traceview που επιτρέπει την γραφική προβολή της ανάλυσης των trace log data που δημιουργούν οι διάφορες εφαρμογές.
- Το mkshcard το οποίο βοηθά στην δημιουργία εικονικού δίσκου ο οποίος μπορεί να χρησιμοποιηθεί από τον εξομοιωτή για την προσομοίωση της παρουσίας εξωτερικής αποθηκευτικής κάρτας (όπως η SD card).
  - Το dx tool το οποίο μετατρέπει τα αρχεία .class από java bytecode σε Android bytecode.
  - Το UI/Application Exerciser Monkey το οποίο είναι ένα πρόγραμμα που τρέχει στον εξομοιωτή και παράγει ψευδό-τυχαίες σειρές από συμβάντα χρήστη όπως clicks, touches, gestures καθώς επίσης και έναν αριθμό από συμβάντα συστήματος.
  - Το activitycreator το οποίο είναι ένα script που δημιουργεί Ant build αρχεία τα οποία μπορούν να χρησιμοποιηθούν για την μεταγλώττιση των εφαρμογών.

## **3.2 Ανάπτυξη εφαρμογών Android**

### **3.2.1 Εγκατάσταση λογισμικού**

Στο πρώτο στάδιο της ανάπτυξης ο προγραμματιστής καλείτε να στήσει το περιβάλλον εργασίας στο οποίο θα γίνει ο σχεδιασμός, η ανάπτυξη, ο έλεγχος, και η λειτουργία των εφαρμογών. Μπορεί να επιλέξει όποιο περιβάλλον ανάπτυξης (IDE), τον εξυπηρετεί καλύτερα και να χρησιμοποιήσει όλα τα εργαλεία του Android SDK μηδενός εξαιρουμένου.

Στη συνέχεια θα πρέπει να δημιουργήσει έναν αριθμό από εικονικές συσκευές στην διαχείριση εικονικών συσκευών (AVD) για να δοκιμάσει την λειτουργία της εφαρμογής σε διαφορετικές πραγματικές συνθήκες λειτουργίας. Ιδανικά ο developer θα διαθέτει έναν αριθμό διαφορετικών φυσικών συσκευών ώστε να δοκιμάσει ο ίδιος πως συμπεριφέρεται η εφαρμογή του σε κάθε περίπτωση, όμως αυτή η πρακτική μπορεί να αποδειχθεί πολυδάπανη και χρονοβόρα. Εδώ αναλαμβάνουν δράση η ευελιξία των AVDs, για τις οποίες θα γράψουμε περισσότερα παρακάτω.

### **3.2.2 Ανάπτυξη πηγαίου κώδικα εφαρμογής**

Πρόκειται αν μη τι άλλο για τη πιο χρονοβόρα και πολύπλοκη διαδικασία. Σε αυτό το στάδιο ο προγραμματιστής πρέπει να αποφασίσει για τις δυνατότητες και το περιεχόμενο που θα περιλαμβάνει η εφαρμογή, να εντοπίσει ποιες από αυτές τις δυνατότητες είναι εφικτές και ποιες θέλουν παραπάνω έρευνα για να προστεθούν στο μέλλον, να σχεδιάσει το layout με γνώμονα την λειτουργικότητα και να αποφύγει υπερβολές στο σχεδιασμό, και τέλος να δέσει αρμονικά τον κώδικα με το layout για να φέρει το τελικό αποτέλεσμα.

Η διαδικασία ξεκινάει με ένα νέο Project το οποίο θα περιέχει τον πηγαίο κώδικα, τις εικόνες, τα κείμενα και γενικά ότι χρειάζεται η εφαρμογή για να τρέξει ως οφείλει. Στο project του ο developer θα πρέπει να φροντίσει ώστε το υλικό του να είναι τακτοποιημένο και ο κώδικας του ευανάγνωστος ώστε να ακολουθήσει η διαδικασία του Debugging.

### **3.2.3 Αποσφαλμάτωση Debugging**

Η διαδικασία του debugging είναι εξίσου κρίσιμη και μερικές φορές και εξίσου χρονοβόρα με την διαδικασία ανάπτυξης του πηγαίου κώδικα της εφαρμογής. Αποτελείτε από αρκετά επί μέρους στάδια τα οποία αναλύονται παρακάτω.

Το πρώτο στάδιο αφορά το αρχικό χτίσιμο της εφαρμογής και η λειτουργία αυτής σε debug mode. Για να γίνει το compile της εφαρμογής φυσικά τα περισσότερα περιβάλλοντα ανάπτυξης (IDE) προϋποθέτουν ότι ο κώδικας δεν έχει κανένα συντακτικό λάθος, αλλιώς ειδοποιούν τον χρήστη να τα διορθώσει. Αφού γίνει το compile η εφαρμογή μπορεί να δοκιμαστεί είτε σε εικονική συσκευή μέσω του AVD Manager, είτε απευθείας σε φυσική συσκευή μέσω ADB push εντολής.

Στο δεύτερο στάδιο ο προγραμματιστής καλείτε να αντιμετωπίσει τα λειτουργικά και αισθητικά προβλήματα της εφαρμογής του, πρώτα εντοπίζοντας τα στην λειτουργία της συσκευής και μετά διορθώνοντας τα κομμάτια του κώδικα που δημιουργούν τα σφάλματα. Το κύριο εργαλείο που κάνει αυτή τη διαδικασία εφικτή είναι το “LogCat” το οποίο μας επιστρέφει το stack trace του κώδικα στο σημείο εκείνο που συνέβη το σφάλμα. Υπάρχουν φυσικά και άλλα εργαλεία τα οποία θα αναλυθούν εκτενώς παρακάτω.

Στο τρίτο στάδιο ο προγραμματιστής αφού έχει τελειώσει την αποσφαλμάτωση (debugging) επιστρέφει στο βήμα ένα, δηλαδή στο compile και τη δοκιμή της εφαρμογής σε εικονική η φυσική συσκευή ώστε να διαπιστώσει τα αποτελέσματα του 2 βήματος, της αποσφαλμάτωσης.

Ένα προαιρετικό στάδιο είναι η “Δημόσια δοκιμαστική φάση” της εφαρμογής. Σε αυτή τη φάση εθελοντές προσφέρονται να δοκιμάσουν τις λειτουργίες της εφαρμογής στις συσκευές τους και να αναφέρουν προβλήματα, παρατηρήσεις, προτάσεις και άλλα σχόλια που μπορεί προκύψουν από τη χρήση της εφαρμογής.

Φυσικά η διαδικασία του debugging είναι σαν ένα βρόγχος (loop) που επαναλαμβάνεται συνέχεια μέχρι να εντοπιστούν και να διορθωθούν όλα τα σφάλματα της εφαρμογής, και για αυτό το λόγο μπορεί να αποδειχθεί εξαιρετικά χρονοβόρα.

### **3.2.4 Android Debug Bridge (ADB)**

Για να λειτουργήσουν σωστά τα εργαλεία που αναφέραμε παραπάνω, χρειάζεται κάποιο είδος προγράμματος client-server που να συνδέει τον υπολογιστή με τις συσκευές μας, εικονικές και μη. Τον ρόλο αυτό αναλαμβάνει το Android Debug Bridge (ADB). Πρόκειται για ένα εργαλείο γραμμής εντολών που έρχεται μαζί με το Android SDK και το οποίο αποτελείτε από 3 μέρη:

➤ Έναν client ο οποίος τρέχει στον υπολογιστή που έχουμε στήσει το SDK. Μπορούμε είτε να τον εκκινήσουμε χειροκίνητα είτε να χρησιμοποιήσουμε κάποιο εργαλείο το οποίο ξεκινάει αυτόματα δικό του client, όπως το DDMS ή το ADT Plugin.

- Έναν server ο οποίος τρέχει σαν υπηρεσία παρασκηνίου στον υπολογιστή που βρίσκεται το SDK, όπως και ο client. Ο server εξασφαλίζει την επικοινωνία μεταξύ του client και του εργαλείου “δαίμονα” (daemon) που τρέχει στη συσκευή.
- Ο “δαίμονας” (daemon) που τρέχει σαν διεργασία παρασκηνίου στην εικονική η πραγματική συσκευή που χρησιμοποιείτε για εξομοίωση.

Όταν ξεκινάει το ADB, ο client ελέγχει αν υπάρχει κάποια υπάρχουσα διεργασία του server που να εκτελείτε ήδη, αλλιώς δημιουργεί μια νέα. Μετά δημιουργεί μια τοπική TCP σύνδεση στην θύρα 5037 και είναι έτοιμος να δεχτεί εντολές. Μετά ελέγχει το εύρος θυρών TCP από 5555 μέχρι 5585, στο οποίο επικοινωνούν οι συσκευές εξομοίωσης, και ελέγχει αν υπάρχουν διαθέσιμες και πόσες είναι αυτές. Αφού εντοπίσει κάποια συσκευή ελέγχει αν σε αυτή τη συσκευή τρέχει ο “δαίμονας”, και αν ναι δημιουργείτε σύνδεση adb με την συσκευή.

Αφότου πραγματοποιηθεί σύνδεση μεταξύ συσκευής και client μπορούμε να χρησιμοποιήσουμε όλες τις δυνατότητες που μας παρέχει το ADB για να ασκήσουμε πλήρη έλεγχο της συσκευής μας. Οι εντολές μέσω γραμμής εντολών δίνονται ως εξής:

**adb [-d|-e|-s <serialNumber>] <command>**

όπου **-d** είναι η απευθείας εντολή εάν υπάρχει μόνο μία συνδεδεμένη συσκευή, **-e** η απευθείας εντολή εάν υπάρχει μόνο μία συνδεδεμένη εικονική συσκευή, **-s <serialNumber>** η απευθείας εντολή στην συσκευή που έχει το **<serialNumber>**, και **<command>** είναι η εντολή που θέλουμε να εκτελεστεί.

### **3.2.5 Εικονικές Συσκευές Android (Android Virtual Devices – AVD)**

Όπως γράψαμε και παραπάνω, ο developer πρέπει πριν να εκδώσει την εφαρμογή του να την δοκιμάσει σε ένα αριθμό συσκευών για να εξασφαλίσει την ομαλή λειτουργία της σε όλες τις συνθήκες. Φυσικά το κόστος των συσκευών είναι αρκετά μεγάλο για να αποθαρρύνει τον προγραμματιστή να έχει στην κατοχή του 10-20 συσκευές για να ελέγξει σε όλες τις λειτουργίες και την εμφάνιση της εφαρμογής του. Το πρόβλημα αυτό έρχεται να λύσει η ύπαρξη των εικονικών συσκευών του Android.

Πρόκειται για μια συσκευή εξομοίωσης η οποία μας επιτρέπει να εξομοιώσουμε την λειτουργία και συμπεριφορά μιας κανονικής συσκευής, ορίζοντας τις επιλογές υλικού και λογισμικού που θέλουμε στον εξομοιωτή του Android. Με αυτό τον τρόπο ο developer μπορεί να ελέγξει την εφαρμογή του σε μια σειρά από πραγματικά σενάρια λειτουργίας και να πάρει γρήγορα και άμεσα feedback για τη λειτουργία της εφαρμογής του. Μια εικονική συσκευή αποτελείτε από:

- Το προφίλ του υλικού: Σε αυτό προσδιορίζονται οι ιδιότητες και τα χαρακτηριστικά της εικονικής συσκευής. Μπορούμε παραδείγματος χάρη να ορίσουμε την ανάλυση της οθόνης και την πυκνότητα σε pixel (dpi), το μέγεθος της μνήμης RAM, αν η συσκευή θα έχει κάμερα, υποστήριξη GPS, κλπ.
- Την έκδοση του Android: Επιλογή της έκδοσης της πλατφόρμας του Android που θέλουμε να εξομοιώσει η εικονική συσκευή. Μπορούμε επίσης να επιλέξουμε και μεταξύ ειδικών εκδόσεων της πλατφόρμας, μεταξύ των οποίων τις Google TV, και άλλων.
- Έξτρα χώρος αποθήκευσης: Εδώ αποθηκεύονται όλα τα δεδομένα της εφαρμογής, και επίσης μπορούμε να ορίσουμε μια εικονική κάρτα μνήμης ώστε να επεκτείνουμε τον αποθηκευτικό χώρο, όπως θα κάναμε και σε μια πραγματική συσκευή.



## **4.Android Application**

### **4.1 Android App με την χρήση των SDK και Eclipse**

Για την προετοιμασία της εφαρμογής απαιτείται η εγκατάσταση μερικών λειτουργικών προγραμμάτων που αναφέρονται λεπτομερώς πιο κάτω. Δεν θα υπάρξουν δραματικές αλλαγές αν χρησιμοποιείτε άλλη έκδοση των Windows ή Linux.

- Εγκατάσταση του Eclipse IDE
- Εγκατάσταση ADT plugin για το Eclipse
- Εγκατάσταση JDK και JRE
- Εγκατάσταση του Android SDK

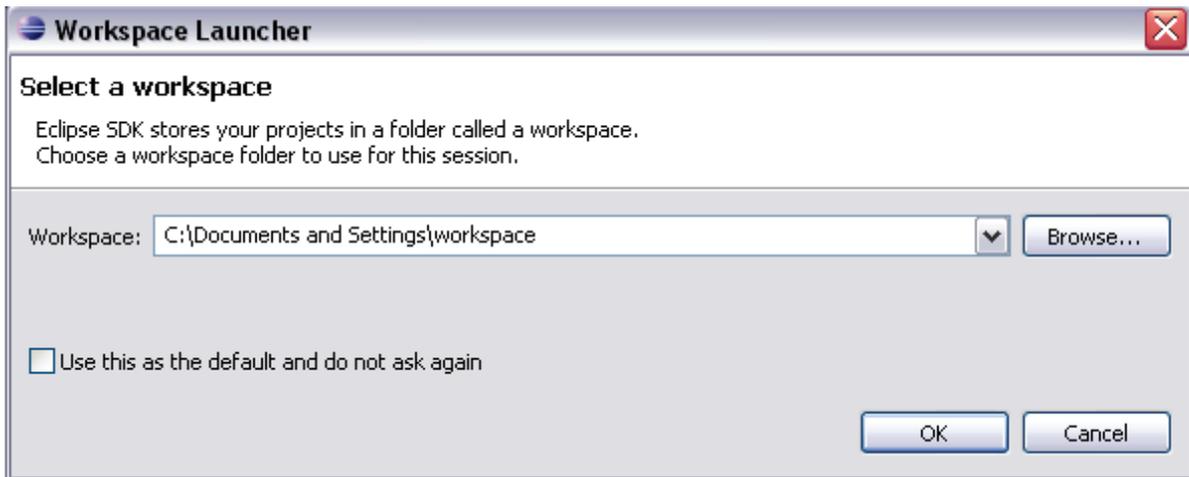
#### **4.1.1 Eclipse IDE**

Το Android προσφέρει ένα προσαρμοσμένο plugin για το Eclipse IDE, το οποίο ονομάζεται Android Development Tools (ADT). Αυτό το plugin προσφέρει ένα ισχυρά, ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών Android το οποίο επεκτείνει τις δυνατότητες του Eclipse για να σας επιτρέπουν να ρυθμίσετε γρήγορα καινούργια Android Project , την κατασκευή app UI και την απασφαλμάτωση (debug) των εφαρμογών.

Είναι απαραίτητη η εγκατάσταση των Java Development Kit (JDK) και Java Runtime Environment (JRE)

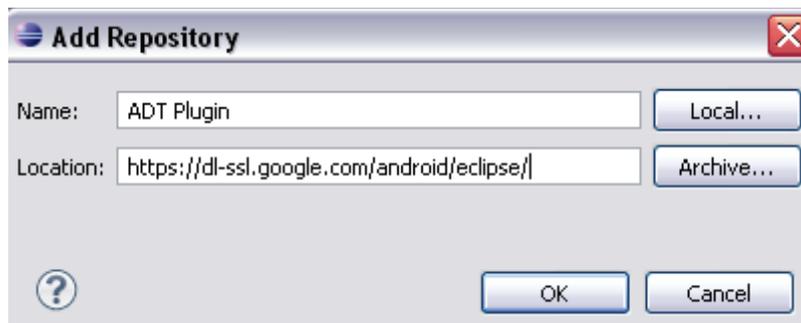
#### **4.1.2 Εγκατάσταση του ADT plugin για το Eclipse**

Τρέχουμε το Eclipse. Αρχικά θα μας ζητηθεί να επιλέξουμε τον φάκελο που θα αποθηκεύονται τα projects μας.



Εικόνα 4.1 Επιλογή αποθήκευσης project

Τώρα θα πρέπει να γίνει η εγκατάσταση του plugin επιλέγοντας **Help > Install New Software**. Στην συνέχεια **Add > Add Repository** , θα προσθέσουμε το ADT Plugin στο πεδίο όνομα και στο πεδίο Location το URL : <https://dl-ssl.google.com/android/eclipse/> .



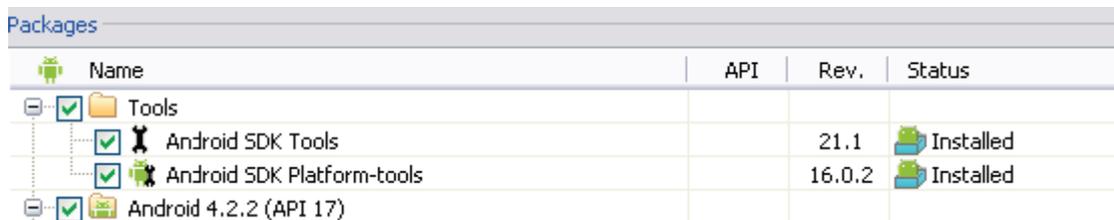
Εικόνα 4.2 Εγκατάσταση του Eclipse.

### 4.1.3 Εγκατάσταση του Android SDK

Κατεβάζουμε το SDK απο το επίσημο site του Android <http://developer.android.com>. Τρέχουμε το αρχείο SDK και στην συνέχεια κάνουμε την εγκατάσταση.

Το Android SDK διαχωρίζει τα εργαλεία , τις πλατφόρμες, και άλλα εξαρτήματα σε πακέτα που μπορούμε να τα κάνουμε λήψη χρησιμοποιώντας το Android SDK Manager. Το πρωτότυπο SDK πακέτο που έχουμε κατεβάσει περιλαμβάνει μόνο τα SDK εργαλεία. Για να αναπτύξει μια Android app, μπορούμε επίσης να πραγματοποιήσουμε λήψη τουλάχιστον μίας πλατφόρμας Android και το τελευταίο SDK.

Το SDK Manager δείχνει όλα τα διαθέσιμα πακέτα για να προσθέσουμε στο Android SDK, **window > Android SDK Manager**.



Packages				
	Name	API	Rev.	Status
<input checked="" type="checkbox"/>	Tools			
<input checked="" type="checkbox"/>	Android SDK Tools		21.1	Installed
<input checked="" type="checkbox"/>	Android SDK Platform-tools		16.0.2	Installed
<input checked="" type="checkbox"/>	Android 4.2.2 (API 17)			

Εικόνα 4.3 Εγκατάσταση SDK

Αφού επιλέξουμε τα πακέτα , πατάμε Install. Το Android SDK Manager εγκαθιστά τις επιλεγμένες συσκευασίες στο Android SDK περιβάλλον.



## **5.Υλοποίηση εφαρμογής Android καταγραφής Location των χρηστών κοινωνικού δικτύου με τη χρήση GPS**

### **5.1 Περιγραφή Εφαρμογής**

Για την υλοποίηση της εφαρμογής θα πρέπει δημιουργήσουμε δύο εφαρμογές. Αρχικά δημιουργούμε την Android εφαρμογή Gps\_PostLocations, η οποία κάνει post το GPS Location και το id σε ένα event ενός κοινωνικού δικτύου, στην περίπτωση μας το Facebook, στο οποίο θα μπορούμε να χρησιμοποιήσουμε το graph API του Facebook για την διαχείριση των δεδομένων προς και από το event που θα δημιουργήσουμε.

Θα γίνει χρήση social authAndroid library για να καλέσουμε κάποια rest calls του Facebook και να στέλνουμε στο event το location της συσκευής.

Το SocialAuth Android είναι η Android έκδοση της δημοφιλούς SocialAuth Java library. Με την χρήση της βιβλιοθήκης αυτής γίνεται εφικτή η ενσωμάτωση της εφαρμογής με πολλαπλά κοινωνικά δίκτυα.

Το API επιτρέπει την ταυτοποίηση των χρηστών και την κοινοποίηση των ενημερώσεων μέσω διαφόρων κοινωνικών δικτύων και παρέχει έναν εύκολο μηχανισμό για την δημιουργία social apps με ασφάλεια.

Τέλος θα δημιουργήσουμε μια άλλη εφαρμογή , GPSRetrieveLocation η οποία θα συνδέεται με το Facebook event και θα ανακτά τα στοιχεία του Location που έχουν κοινοποιηθεί και θα τα εξάγει στην οθόνη του κινητού.

Οι περισσότερες Android συσκευές επιτρέπουν τον καθορισμό της τρέχον θέσης. Αυτό μπορεί να γίνει μέσω GPS (Global Positioning System) ή μέσω δικτύων WiFi.

Το Android περιέχει το android.location package που παρέχει το API για να προσδιοριστεί η γεωγραφική θέση.

Η LocationManager class παρέχει την πρόσβαση στην υπηρεσία Android Location. Αυτή η υπηρεσία επιτρέπει την πρόσβαση στους πάροχους θέσης και την δήλωση ενημέρωσης θέσης.

Η LocationProvider class είναι υπερκλάση των διαφορων πάροχων οι οποίοι δίνουν πληροφορίες για την τρέχουσα θέση. Οι πληροφορίες αυτές αποθηκεύονται στο Location class.

Οι Android συσκευές ενδέχεται να έχουν αρκετά LocationProvider διαθέσιμα και μπορούμε να επιλέξουμε πια θα χρησιμοποιήσουμε. Στον παρακάτω πίνακα φέρονται κάποια απο τα ποιά συνηθισμένα.

LocationProvider	Description
network	Uses the mobile network or WI-FI to determine the best location. Might have a higher precision in closed rooms than GPS.
gps	Use the GPS receiver in the Android device to determine the best location via satellites. Usually better precision than network.
passive	Allows to participate in location of updates of other components to save energy

Εικόνα 5.1 Διαθέσιμα LocationProvider.

Για την καλύτερη επιλογή των Location Provider χρησιμοποιούμε το Criteria object, με τα οποία ορίζουμε εμείς τον τρόπο με τον οποίο θα γίνει η επιλογή.

Η Geocoder class επιτρέπει τον προσδιορισμό των συντεταγμένων διευθύνσεων (γεωγραφικό μήκος και πλάτος ), χρησιμοποιώντας μια online Google υπηρεσία.

Το Google Map Activity ενεργοποιεί αυτόματα το GPS ή χειροκίνητα με την χρήση του Location Manager.

Χρησιμοποιούμε το DDMS Perspective απο το Eclipse για να σταλεί η θέση του εξομοιωτή ή της συνδεδεμένης συσκευής. Αυτό γίνεται επιλέγοντας Window → Open Perspective → Other → DDMS. Στο Emulator Control εισάγουμε τις συντεταγμένες.

Μπορούμε επίσης να ρυθμίσουμε το geoposition μέσω telnet.

```
telnet localhost 5554
```

Ρυθμίζουμε την θέση μέσω της παρακάτω εντολής.

```
geo fix 13.4 52.3
```

## 5.2 Εφαρμογή GPS\_Post\_Locations

Για την υλοποίηση της εφαρμογή δημιουργούμε αρχικά ένα ένα Facebook event , απο το οποίο θα κρατήσουμε το id του event που θα δημιουργήσουμε.

Διουργούμε ένα project με όνομα GPS\_PostLocation και Activity με όνομα PostGPSActivity.

### ■ gps.postlocation / PostGPSActivity.java

```
package gps.postlocation;

import experimedia.socialintegrator.R;

import gps.datatransferobjects.EventAttributesDTO;
import gps.datatransferobjects.SNUserAttributeDTO;
import gps.util.GPS_Util;

import android.app.Activity;
import android.content.Context;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;

public class PostGPSActivity extends Activity implements LocationListener {
    private TextView latitudeField;
    private TextView longitudeField;
    private Location location;
    private LocationManager locationManager;
    private String provider;

    private String uid = "001";

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.postlocation);

        // latitudeField = (TextView) findViewById(R.id.TextView02);
        // longitudeField = (TextView) findViewById(R.id.TextView04);

        // Get the location manager
        locationManager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);
        // Define the criteria how to select the locatioin provider -> use
        // default
```

```
Criteria criteria = new Criteria();
provider = locationManager.getBestProvider(criteria, false);
location = locationManager.getLastKnownLocation(provider);
// location = locationManager.getLastKnownLocation("gps");
// Initialize the location fields
if (location != null) {
    System.out.println("Provider " + provider + " has been selected.");
    onLocationChanged(location);
} else {
    latitudeField.setText("Location not available");
    longitudeField.setText("Location not available");
    Toast.makeText(this, "Location not available",
        Toast.LENGTH_LONG).show();
}
}

public void showAboutMessage(View v) {

    String event_wall_location = uid + "," + location.getLongitude() + "," +
location.getLatitude();
        GPS_Util.sendMessage(SNUserAttributeDTO.ADAPTER.getCurrentProvider(),
event_wall_location, EventAttributesDTO.EVENT_ID, "");

    Toast.makeText(this, "Location was posted",
        Toast.LENGTH_LONG).show();
}

/* Request updates at startup */
@Override
protected void onResume() {
    super.onResume();
    locationManager.requestLocationUpdates(provider, 400, 1, this);
}

/* Remove the locationlistener updates when Activity is paused */
@Override
protected void onPause() {
    super.onPause();
    locationManager.removeUpdates(this);
}

@Override
public void onLocationChanged(Location location) {
    int lat = (int) (location.getLatitude());
    int lng = (int) (location.getLongitude());
    latitudeField.setText(String.valueOf(lat));
    longitudeField.setText(String.valueOf(lng));
}

@Override
public void onStatusChanged(String provider, int status, Bundle extras) {
    // TODO Auto-generated method stub
}

@Override
public void onProviderEnabled(String provider) {
    Toast.makeText(this, "Enabled new provider " + provider,
        Toast.LENGTH_SHORT).show();
}
```

```
}  
  
@Override  
public void onProviderDisabled(String provider) {  
    Toast.makeText(this, "Disabled provider " + provider,  
        Toast.LENGTH_SHORT).show();  
}  
}
```

Στην συνέχεια δημιουργούμε την κλάση Location και Util , τις οποίες θα τις χρησιμοποιήσουμε και στις δύο εφαρμογές.

- gps.object/ Location.java το οποίο θα παρέχει το lon και το lan. Με τις κατάλληλες μεθόδους θέτει και παίρνει το Location.

```
package gps.object;  
  
public class Location {  
  
    private String lat_  
    private String lon_  
    private String uid_  
  
    public Location() {}  
  
    /**  
     * return the latitude of the location  
     *  
     * @return lat    location latitude  
     */  
    public String getLatitude() {  
        return lat_  
    }  
  
    /**  
     * return the longitude of the location  
     *  
     * @return lon    location longitude  
     */  
    public String getLongitude() {  
        return lon_  
    }  
  
    /**  
     * return the user id of the mobile device  
     * posting the location  
     *  
     * @return uid    user id  
     */  
    public String getUid() {  
        return uid_  
    }  
}
```

```
/**
 * set the latitude of the place
 *
 * @param lat    location latitude
 */
public void setLatitude(String lat) {
    lat_ = lat;
}

/**
 * set the longitude of the place
 *
 * @param lon    location longitude
 */
public void setLongitude(String lon) {
    lon_ = lon;
}

/**
 * set the user id that maps to the location posting
 * posting
 *
 * @param uid    user id
 */
public void setUid(String uid) {
    uid_ = uid;
}
}
```

## ■ gps.util/ GPS\_Util.java

```
package gps.util;

import gps.object.Location;
import gr.ntua.experimedia.socialintegrator.providers.AugmentedFacebookImpl;
// import gr.ntua.experimedia.socialintegrator.objects.Message;
// import gr.ntua.experimedia.socialintegrator.providers.AugmentedFacebookImpl;

import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.util.LinkedList;
import java.util.List;
import java.lang.StringBuilder;

import org.brickred.socialauth.AuthProvider;
import org.brickred.socialauth.util.Constants;
import org.brickred.socialauth.util.MethodType;
import org.brickred.socialauth.util.Response;
import org.json.JSONArray;
import org.json.JSONObject;

import org.apache.log4j.Logger;
```

```
public class GPS_Util {
    final static String BASE_URL_FACEBOOK = "https://graph.facebook.com/";

    static final String FACEBOOK_POST_LOG_MESSAGE = " returned by facebook
post query ";

    private static final Logger LOGGER = Logger
        .getLogger(GPS_Util.class.getName());

    public static List<Location> getMessages(AuthProvider provider, String
source, String type, String lim, String until) throws
UnsupportedEncodingException{

        List<Location> eventLocations = null;
        String next = null;
        String limit = null;
        List<Location> locations = new LinkedList<Location>();
        // if (provider.getProviderId() == Constants.FACEBOOK) {

            String event_id = "308578935938695";
            String fql_query = "SELECT post_id,message FROM stream WHERE
source_id=" + event_id;

            StringBuilder url = new StringBuilder();
            url.append(BASE_URL_FACEBOOK).append("/fql").append("?");

            url.append("q").append("=").append(URLEncoder.encode(fql_query,
Constants.ENCODING)).append("&");

            try {
                Response response = provider.api(url.toString(),
MethodType.GET.toString(), null, null, null);
                //
                System.out.print(response.getResponseBodyAsString("UTF-8"));
                // Log("AndroidAppPhotoUtil", "Status " +
response.getStatus() + " returned by facebook get query " + url);
                if (response.getStatus() == 200) {
                    //
                    System.out.print(response.getResponseBodyAsString("UTF-8") + "\n");
                    String respStr =
response.getResponseBodyAsString(Constants.ENCODING);
                    // System.out.print(respStr + " " + respStr +
"\n");

                    JSONObject resp = new JSONObject(respStr);
                    System.out.print(resp + "\n");
                    JSONArray data = resp.optJSONArray("data");
                    System.out.print(data + "\n");
                    if (data != null)
                        for (int i = 0; i < data.length(); i++) {
                            JSONObject obj =
data.getJSONObject(i);

                            Location l = new Location();

                            String msg = obj.optString("message");
                            /** TODO Parse the string [lon=...,
lat=..., uid=...] */

                            String lon = null;
                            String lat = null;
                            String uid = null;
                        }
                }
            }
        }
    }
}
```

```
        String delims = "[,]";
        String[] tokens = msg.split(delims);
        lon = tokens[0];
        lat = tokens[1];
        uid = tokens[2];

        l.setLatitude(lat);
        l.setLongitude(lon);
        l.setUid(uid);

        if (type.equals(obj.optString("type"))) { //
(isEmpty(type) || type.equals(obj.optString("type")))
        System.out.print(l.getLatitude() + "\n");
        System.out.print(l.getLongitude() + "\n");
        System.out.print(l.getUid() + "\n");
            locations.add(l);
        }
    }
}
} catch (Exception e) {
    e.printStackTrace();
}
//}
return locations;
}

/**
 * Method to send message or reply.<br>
 * Facebook: Needs publish_stream permission +
 * user_groups, friends_groups, user_events or friends_events
 * depending on where the message is sent.
 *
 * See <a href="https://developers.facebook.com/docs/authentication/permissions/">
 * https://developers.facebook.com/docs/authentication/permissions/</a>
 * for details.
 * @param provider
 * @param message The message to be sent. Must be non empty.
 * @param dest The identification code of message being replied.
 * For Facebook it can also be a group's id, a friend's id
 * or an event's id depending on where we want to send a
message
 * @param receiver The username of the message sender being replied. (only
for Twitter if applicable)
 * @return the id of the message
 */
public static String sendMessage(AuthProvider provider, String message,
String dest, String receiver ) {
    String result = null;
    try {
        String providerId = provider.getProviderId();
        if ((Constants.FACEBOOK).equals(providerId) ||

(AugmentedFacebookImpl.CLASSID.equals(providerId)) ) {
            if (isEmpty(dest)) dest = "me";
            StringBuilder url = new StringBuilder();
            StringBuilder body = new StringBuilder();

            if (dest.indexOf("_") < 0) {
```

```
url.append(BASE_URL_FACEBOOK).append(dest).append("/feed");
    } else {

url.append(BASE_URL_FACEBOOK).append(dest).append("/comments");
    }
    body.append("message=").append(message);

    Response response = provider.api(url.toString(),
MethodType.POST.toString(),
        null, null, body.toString());

    LOGGER.info("Status " + response.getStatus() +
FACEBOOK_POST_LOG_MESSAGE + url);
    if (response.getStatus() == 200) {
        String respStr =
response.getResponseBodyAsString(Constants.ENCODING);
        JSONObject resp = new JSONObject(respStr);
        result = resp.optString("id");
    }
    } else if (Constants.TWITTER.equals(providerId)) {
        /* No twitter code needed */
    }
    } catch (Exception e) {
        LOGGER.error(e.getMessage());
        e.printStackTrace();
    }
    return result;
}

private static boolean isEmpty(String str) {
    return ( (str == null) || "".equals(str.trim()) );
}
}
```

## ■ **gps.Tokenretriever/ RetrieveFacebookTokenActivity.java**

```
package gps.tokenretriever;

import java.io.Serializable;
import org.brickred.socialauth.Profile;
import org.brickred.socialauth.android.DialogListener;
import org.brickred.socialauth.android.SocialAuthError;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.Intent;
import android.graphics.Color;
import android.graphics.drawable.AnimationDrawable;
import android.os.Bundle;
import android.os.StrictMode;
import android.util.Log;
import android.view.Gravity;
import android.view.View;
```

```
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.TextView;
import experimedia.eu.socialauth.android.extension.SocialAuthAdapterExperimedia;
import experimedia.socialintegrator.R;
import gps.datatransferobjects.EventAttributesDTO;
import gps.datatransferobjects.SNUserAttributeDTO;
import gps.postlocation.PostGPSActivity;

public class RetrieveFacebookTokenActivity extends Activity implements
Serializable
{

    /* Serial Version UID */
    private static final long serialVersionUID = -4116399871405578909L;

    /* Event id */
    private static final String EVENT_ID = "308578935938695";

    /* SocialAuth Components */
    private SocialAuthAdapterExperimedia adapter;

    private final String TAG = "SocialIntegrator";

    Profile profileMap;

    /**
     * Reference to the ImageView which will display the animation.
     */
    ImageView animation;

    /* Android Components */
    ListView listview;
    AlertDialog dialog;

    /* Variables */
    boolean status;
    public static int pos;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        /* Adapter initialization */
        adapter = new SocialAuthAdapterExperimedia(new ResponseListener());

        listview = (ListView) findViewById(R.id.listview);
        listview.setAdapter(new CustomAdapter(this, adapter));

        animation = (ImageView) findViewById(R.id.animated_si);

        /* Welcome Message */
        TextView textview = (TextView) findViewById(R.id.text);
        textview.setText("© Copyright 2012 National Technical University of
Athens");
        textview.setTextColor(Color.GRAY);
        textview.setGravity(Gravity.CENTER);
        textview.setPadding(0, 0, 0, 0);
    }
}
```

```
}

/* To receive the response after authentication */
private final class ResponseListener implements DialogListener
{
    public void onComplete(Bundle values) {
        StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);

        Log.d(TAG , "Successful");

        /* Changing Sign In Text to Sign Out */
        /* Code to refresh Single ListView Item : You can remove it
for your app */
        View v = listview.getChildAt(pos -
listview.getFirstVisiblePosition());
        TextView pText = (TextView) v.findViewById(R.id.signstatus);
        pText.setPadding(0, 30, 30, 10);
        pText.setText("Sign Out");

        /* Get the provider */
        final String providerName =
values.getString(SocialAuthAdapterExperimedia.PROVIDER);
        Log.d(TAG, "providername = " + providerName);

        /* Set global adapter */
        SNUUserAttributeDTO.ADAPTER = adapter;

        /* Get first event from page (the first one is the latest
created?TODO) */

        // Event e =
SNUUserAttributeDTO.ADAPTER.getEvents(PAGE_ID).get(0);
        EventAttributesDTO.EVENT_ID = EVENT_ID;// e.getId();

        /* Join User to Group */
        SNUUserAttributeDTO.ADAPTER.joinEvent(EVENT_ID);
        // SNUUserAttributeDTO.ADAPTER.joinEvent(e.getId());

        // Log.d("Custo-UI", e.getRsvp_status().toString());

        /* Get messages and photos of event */
        // List<Message> eventMessages =
AndroidAppPhotoUtil.getMessages(SNUUserAttributeDTO.ADAPTER.getCurrentProvider(),
EventAttributesDTO.EVENT_ID, null);
        // List<Photo> eventPhotos =
AndroidAppPhotoUtil.getPictures(SNUUserAttributeDTO.ADAPTER.getCurrentProvider(),
EventAttributesDTO.EVENT_ID, null);

        /* Set the global event attributes */
        // EventAttributesDTO.InitiallizeEventVars(eventMessages,
eventPhotos);

        /* Start New activity completing this part of the application
view */
        Intent myIntent = new Intent(RetrieveFacebookTokenActivity.this,
PostGPSActivity.class);
        startActivityForResult(myIntent, 0);
    }
}
```

```
    }

    public void onError(SocialAuthError error) {
        Log.d(TAG , "Error");
    }

    public void onCancel() {
    }

}

/**
 * This method is called whenever the Activity becomes visible or invisible
to the user.
 * During this method call its possible to start the animation.
 */
@Override
public void onWindowFocusChanged (boolean hasFocus) {
    super.onWindowFocusChanged(hasFocus);
    AnimationDrawable frameAnimation =
        (AnimationDrawable) animation.getBackground();
    //frameAnimation.clearColorFilter();

    frameAnimation.setOneShot(true);

    if(hasFocus) {
        frameAnimation.start();
    } else {
        frameAnimation.stop();
    }
}
}
```

Στον folder res/layout δημιουργούμε το posttlocation.xml το οποίο θα δίχνει το layout όταν ο χρήστης θα κάνει post το location.

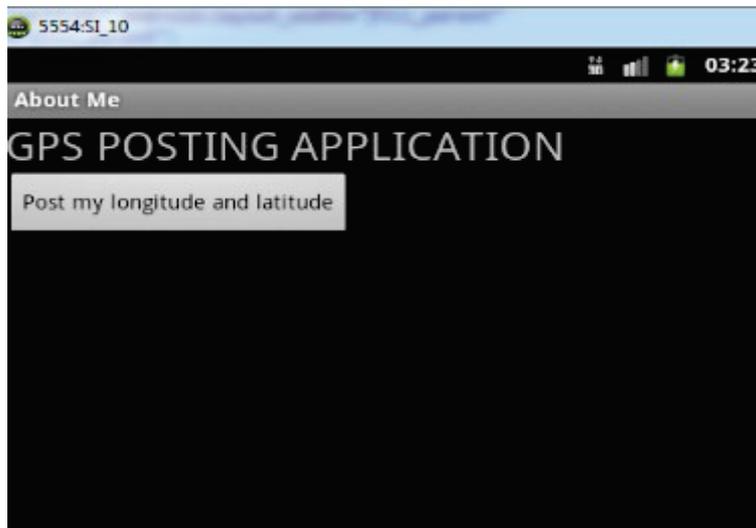
```
<?xml version="1.0" encoding="UTF-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <TextView android:textSize="28dp" android:text="GPS POSTING APPLICATION"
        android:id="@+id/textView1" android:layout_height="wrap_content"
        android:layout_width="wrap_content">
    </TextView>

    <Button android:id="@+id/aboutMeButton"
    android:layout_height="wrap_content"
```

```
        android:onClick="showAboutMessage"  
android:layout_width="wrap_content"  
        android:text="Post my longitude and latitude">  
    </Button>  
</LinearLayout>
```

Στην συνέχεια δημιουργούμε έναν emulator από το AND Manager και τρέχουμε το πρόγραμμα. Το αποτέλεσμα της εφαρμογής φέρεται στην παρακάτω εικόνα:



Εικόνα 5.2 Posting Location εφαρμογή στον emulator.

### 5.3 Δημιουργία GPS\_Retrieval εφαρμογής

Δημιουργούμε ένα project με όνομα GPS\_Post\_Locations. Στον folder src/ δημιουργούμε την κλάση SimpleListViewActivity στο gps.retrieveLocations package.

#### ■ Gps.retrieveLocations/ SimpleListViewActivity.java

```
package gps.retrieveLocations;  
  
import java.io.UnsupportedEncodingException;  
import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.List;
```

```
import experimedia.socialintegrator.R;
import gps.datatransferobjects.EventAttributesDTO;
import gps.datatransferobjects.SNUserAttributeDTO;
import gps.object.Location;
import gps.util.GPS_Util;

import android.app.Activity;
import android.os.Bundle;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ListView;

public class SimpleListViewActivity extends Activity {

    private ListView mainListView ;
    private ArrayAdapter<String> listAdapter ;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.retrieve_loc_list);

        // Find the ListView resource.
        mainListView = (ListView) findViewById( R.id.mainListView );

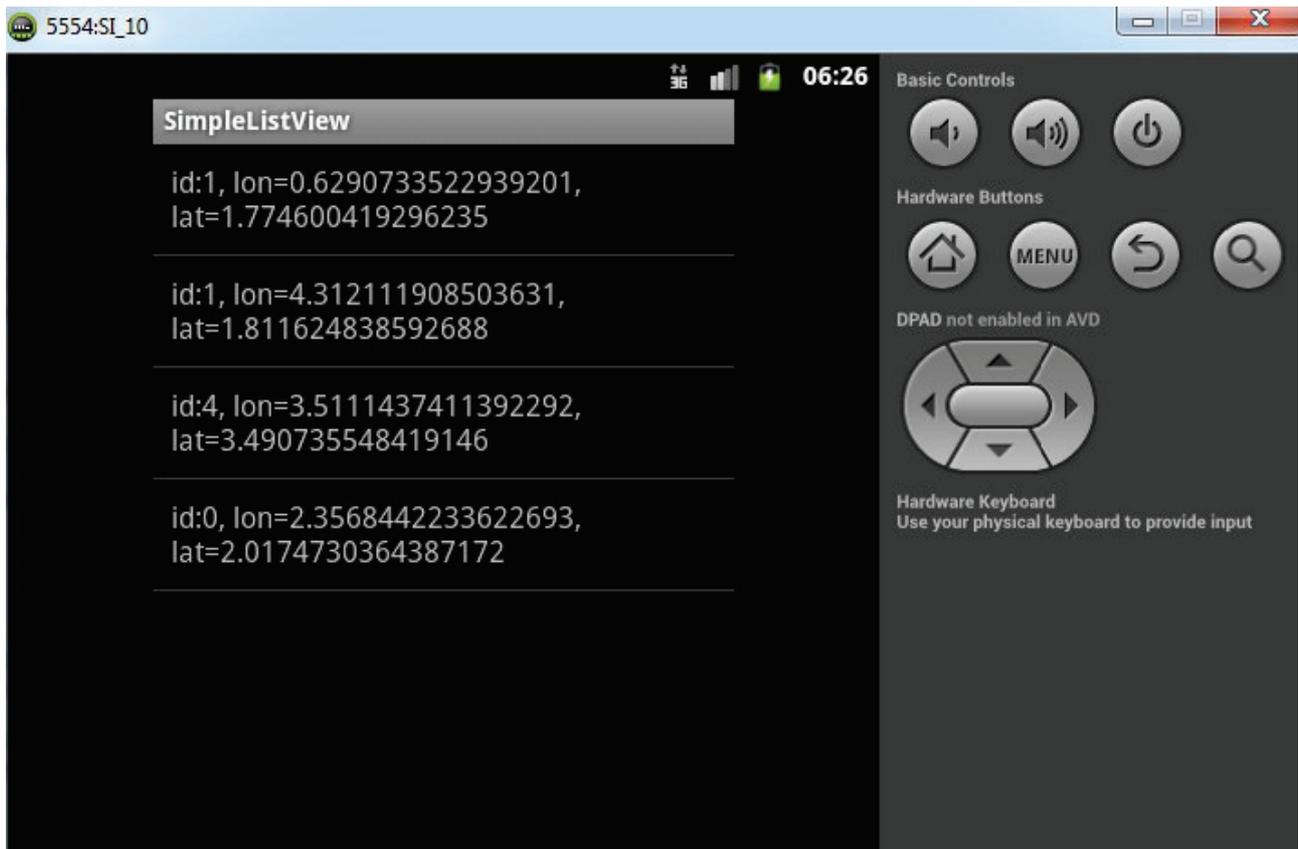
        // Populate a List of locations taken from facebook event.
        // String facebook_source = SNProps.getProperty("facebook_event_id");
        List<String> locations_str = new ArrayList<String>();
        List<Location> lm = new ArrayList<Location>();
        try {
            lm =
GPS_Util.getMessages(SNUserAttributeDTO.ADAPTER.getCurrentProvider(),
EventAttributesDTO.EVENT_ID, "", "", "");
        } catch (UnsupportedEncodingException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        for (Location post: lm) {
            locations_str.add("UID: " + post.getUid() + "Lon: " +
post.getLongitude() + ", Lat: " + post.getLatitude() + '\n');
        }

        // Create ArrayAdapter using the planet list.
        listAdapter = new ArrayAdapter<String>(this, R.layout.simplerow,
locations_str);

        // Set the ArrayAdapter as the ListView's adapter.
        mainListView.setAdapter( listAdapter );
    }
}
```

Τέλος τρέχουμε την εφαρμογή και το αποτέλεσμα της φέεται στην παρακάτω εικόνα:



Εικόνα 5.3 GPS Retrieve Location εφαρμογή στον Emulator.



## **ΒΙΒΛΙΟΓΡΑΦΙΑ**

- [1] <http://developer.android.com/sdk/installing/installing-adt.html> .
- [2] <http://developer.android.com/guide/components/activities.html> .
- [3] <http://www.androidauthority.com/android-4-1-vs-android-4-2-jelly-bean-149740/> .
- [4] [http://www.gsmarena.com/state\\_of\\_android\\_jan\\_2013\\_jb\\_market\\_share\\_still\\_under\\_15-news-5489.php](http://www.gsmarena.com/state_of_android_jan_2013_jb_market_share_still_under_15-news-5489.php) .  
[http://en.wikipedia.org/wiki/Android\\_%28operating\\_system%29](http://en.wikipedia.org/wiki/Android_%28operating_system%29) .
- [5] <http://www.techradar.com/news/phone-and-communications/mobile-phones/android-5-0-key-lime-pie-release-date-news-and-rumours-1091500> .
- [6] <http://andbook.anddev.org/>
- [7] [http://reviews.cnet.com/8301-19736\\_7-57560057-251/android-in-2013-what-to-expect/](http://reviews.cnet.com/8301-19736_7-57560057-251/android-in-2013-what-to-expect/) .
- [8] <http://techblog.gr/software/android-4-2-jelly-bean-new-characteristics-454545/> .
- [9] <http://www.scribd.com/doc/14180203/An-Introduction-to-Android> .
- [10] <http://www.netmarketshare.com/#>
- [11] <http://www.edumobile.org/android/android-tutorial/what-is-ddms-and-how-it-works/> .
- [12] <https://source.android.com/tech/index.html> .
- [13] <http://developer.android.com/tools/debugging/index.html> .
- [14] <https://code.google.com/p/socialauth/> .
- [15] <http://www.android-app-market.com/android-architecture.html> .