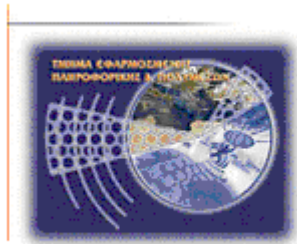




Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων



Πτυχιακή Εργασία

Τίτλος: Ανάπτυξη εφαρμογής σε λειτουργικό σύστημα Android.

Επίσημανση προσώπων και προσθήκη περιγραφής σε φωτογραφία.

Όνοματεπώνυμο: Θεοδωράκη Ανδριανή **ΑΜ:** 1331

Επιβλέπων Καθηγητής: Τριανταφυλλίδης Γεώργιος

ΗΡΑΚΛΕΙΟ 2012

Ευχαριστίες

Στην διεκπεραίωση της πτυχιακής αυτής βοήθησαν έμπρακτα και μη η οικογένεια μου, με την αμέριστη συμπαράσταση και την υπομονή χρόνων. Θα ήθελα επίσης να ευχαριστήσω τον καθηγητή μου Δρ Γεώργιο Τριανταφυλλίδη και τον συνάδελφο Γιώργο Αναγνώσταρο, για την πολύτιμη βοήθεια τους.

Abstract

Over the years mobile users are asking for more, creative and unique applications. Each user wants full control and customization of mobile phone, each mobile operator wants to offer its subscribers a convenient and simultaneously profitable content. Developers like to develop strong applications for mobile phones based on consumer demand, but this is not feasible. Finally, manufacturers of mobile phones want a stable, secure and affordable platform to power their devices.

Entering Android, things change for the mobile development community. Android is an innovative and open platform, well positioned to address the growing needs of the mobile marketplace.

The subject of this thesis is the development of an application, operating system Android, which allows the user to store, in an image that exists on the device, a text and identify persons through the contacts on the device.

Σύνοψη

Με την πάροδο των χρόνων οι χρήστες κινητών τηλεφώνων ζητούν περισσότερες, πιο δημιουργικές και πρωτότυπες εφαρμογές. Ο κάθε χρήστης θέλει τον πλήρη έλεγχο και την ολοκληρωτική παραμετροποίηση του κινητού του τηλεφώνου, η κάθε εταιρία κινητής τηλεφωνίας θέλει να προσφέρει στους συνδρομητές της ένα εύχρηστο και ταυτόχρονα επικερδές περιεχόμενο. Ο προγραμματιστές ήθελαν ελευθερία για την ανάπτυξη ισχυρών εφαρμογών για κινητά τηλέφωνα με βάση την ζήτηση των καταναλωτών, χωρίς αυτό να είναι εφικτό. Τέλος, οι κατασκευαστές συσκευών κινητής τηλεφωνίας θέλουν μια σταθερή, ασφαλή και οικονομικά προσιτή πλατφόρμα για να τροφοδοτήσει τις συσκευές τους.

Μέσα από την πλατφόρμα Android αλλάζουν τα δεδομένα για την κοινότητα ανάπτυξης εφαρμογών για κινητά τηλέφωνα και συσκευές, εφόσον αναφερόμαστε σε μια καινοτόμο και ανοιχτή πλατφόρμα, που είναι σε θέση να αντιμετωπίσει τις αυξανόμενες ανάγκες της αγοράς κινητής τηλεφωνίας.

Το αντικείμενο αυτής της πτυχιακής εργασίας είναι η ανάπτυξη μιας εφαρμογής, σε λειτουργικό σύστημα Android, που θα δίνει τη δυνατότητα στο χρήστη να αποθηκεύει, σε μια εικόνα που υπάρχει στη συσκευή του, ένα κείμενο και να επισημαίνει πρόσωπα μέσα από τις επαφές που υπάρχουν στη συσκευή.

Πίνακας Περιεχομένων

1.1	Εισαγωγή στο περιβάλλον Android	9
1.2	Ανάγκη ανάπτυξης λογισμικού Android	9
1.3	Χαρακτηριστικά και Λειτουργίες Android Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.	
1.4	Αρχιτεκτονική Android	15
1.4.1	Εφαρμογές (Applications)	16
1.4.2	Πλαίσιο εφαρμογών (Application framework)	16
1.4.3	Βιβλιοθήκες (Libraries)	17
1.4.4	Περιβάλλον Android (Runtime Android).....	17
1.4.5	Κέλυφος Kernel (Linux Kernel)	17
1.5	Λογισμικά για κινητές συσκευές	18
1.5.1	Palm OS.....	18
1.5.2	Windows mobile.....	18
1.5.3	Blackberry RIM	18
1.5.4	iOS.....	19
1.5.5	Android	19
1.5.6	Symbian OS	20
1.5.7	Windows phone 7.....	20
1.6	Πλατφόρμες Android Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.	
1.7	Επιτυχία πλατφόρμας Android	21
2.1	Εφαρμογές	24
2.2	Barcode scanner, RFID, χάρτες Google	25
2.2.1	Barcode scanner	25
2.2.2	RFID.....	27
2.2.3	Χάρτης Google	28
2.3	Ασφάλεια και δικαιώματα	28
2.3.1	Αρχιτεκτονική ασφαλείας.....	28
2.3.2	Υπογραφή εφαρμογής.....	28
2.3.3	ID χρήστη και πρόσβαση αρχείων.....	28
2.3.4	Χρησιμοποιώντας τις άδειες	29
2.3.5	Δήλωση και επιβολή αδειών	30

2.3.6	Άδειες URI (Uniform Resource Indicator).....	32
2.4	Διανομή εφαρμογών στους χρήστες.....	32
2.5	Σήμερα	35
3.1	Στάδια ανάπτυξης εφαρμογής.....	36
3.2	Εργαλεία ανάπτυξης	36
3.2.1	Android SDK.....	36
3.2.2	Eclipse IDE.....	36
3.2.3	ADT Plugin για το Eclipse	36
3.2.4	Εικονική μηχανή Android (AVD).....	37
3.2.5	Ανάπτυξη βάσης δεδομένων.....	37
3.3	Εγκατάσταση εργαλείων ανάπτυξης.....	38
3.3.1	<i>Microsoft Visio</i>	38
3.3.2	<i>SQLite Manager</i>	38
3.3.3	<i>Eclipse IDE</i>	38
5.1	Θεωρητικό μέρος εφαρμογής.....	43
5.1.1	Περιγραφή εφαρμογής	43
5.1.2	Καταγραφή λειτουργιών εφαρμογής.....	43
5.1.3	Οντότητες και Σχεσιακό διάγραμμα.....	43
5.1.4	Σχεδιασμός αρχιτεκτονικής	44
5.2	Πρακτικό μέρος εφαρμογής	44
5.2.1	Γραφικό περιβάλλον εφαρμογής	44
5.2.2	AndroidManifest.xml.....	45
5.2.3	Το αρχείο strings.xml.....	45
5.2.4	Οι κλάσεις της εφαρμογής.	45
5.2.5	Απεικόνιση των οθονών της εφαρμογής	49
5.3	Συμπεράσματα, Μελλοντικές επεκτάσεις	54
5.3.1	Συμπεράσματα	54
5.3.2	Μελλοντικές επεκτάσεις	55

Πίνακας εικόνων

Εικόνα 1 - Αρχιτεκτονική πλατφόρμας Android	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Εικόνα 2 - Στατιστικά Αυγούστου 2010.....	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Εικόνα 3 - Στατιστικά Μαρτίου 2011	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Εικόνα 4 - Στατιστικά Ιουλίου 2011.....	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Εικόνα 5 - Στατιστικά Νοεμβρίου 2011.....	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Εικόνα 6 - Νοέμβριος 2009	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Εικόνα 7 - Android vs iOS – Οκτώβριος 2010.....	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Εικόνα 8 - Ιούλιος 2011	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Εικόνα 9 - Στατιστικά Ιανουαρίου– Νοεμβρίου 2011	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Εικόνα 10 - Εντοπισμός QR code.....	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Εικόνα 11 – Ανοίγουμε τη σχετική εφαρμογή και στρέφουμε την κάμερα στο QR code	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Εικόνα 12 - Το κινητό αυτόματα ζητάει να συνδεθεί στη διεύθυνση που του έχουμε υποδείξει	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Εικόνα 13 - RFID ετικέτα.....	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Εικόνα 14 - Εφαρμογές Android στο Android Market	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Εικόνα 15 - Λήψεις Android market.....	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Εικόνα 16 - Δωρεάν εφαρμογές στις διάφορες πλατφόρμες	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Εικόνα 17 - Android market vs Apple App store	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Εικόνα 18 - Free vs Paid.....	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Εικόνα 19 Hello Android.....	40
Εικόνα 20 Αρχική εικόνα εικονικής συσκευής	41
Εικόνα 21 Αφού ξεκλειδώσουμε τη συσκευή βρισκόμαστε στην κεντρική οθόνη.	41
Εικόνα 22 Πατώντας το κουμπί που μας πηγαίνει στο κεντρικό μενού βλέπουμε τις εφαρμογές που είναι εγκατεστημένες στην συσκευή μας. Παρατηρούμε την εφαρμογή που μόλις δημιουργήσαμε Hello Android.	42
Εικόνα 23 Πατώντας το εικονίδιο που αντιστοιχεί στην εφαρμογή μας βλέπουμε τα αποτελέσματα του κώδικα που γράψαμε.....	42
Εικόνα 24 Σχεσιακό διάγραμμα	44

Εικόνα 25 Οθόνη 1: Λήψη ή επιλογή φωτογραφίας.....	49
Εικόνα 26 Λήψη φωτογραφίας.	50
Εικόνα 27 Αποθήκευση φωτογραφίας.....	50
Εικόνα 28 Προβολή φωτογραφιών για επιλογή.	51
Εικόνα 29 Άνοιγμα επιλεγμένης φωτογραφίας.	51
Εικόνα 30 Μενού εικόνας.	52
Εικόνα 31 Εισαγωγή περιγραφής.....	52
Εικόνα 32 Επιλογή επαφής τηλεφώνου.....	53

Κεφάλαιο 1ο - Το περιβάλλον Android.

1.1 Εισαγωγή στο περιβάλλον Android

Το Android είναι μια ολοκληρωμένη, ανοιχτή και ελεύθερη πλατφόρμα για κινητά τηλέφωνα που περιλαμβάνει ένα λειτουργικό σύστημα (OS), το απαραίτητο ενδιάμεσο λογισμικό, βιβλιοθήκες και βασικές εφαρμογές. Δημιουργήθηκε από τον οργανισμό Open Handset Alliance και τη Google και έχει δημιουργήσει αναταραχή στον ασύρματο κόσμο. Η πλατφόρμα αυτή είναι η μόνη που μέχρι τώρα έχει ικανοποιήσει τις διάφορες ανάγκες των χρηστών, των προγραμματιστών αλλά και των κατασκευαστών.

Το Android είναι μια μοναδική πλατφόρμα που επιτρέπει την ανάπτυξη εφαρμογών λογισμικού το οποίο εκμεταλλεύεται πλήρως τις δυνατότητες μιας συμβατής συσκευής. Για παράδειγμα, οι προγραμματιστές εφαρμογών είναι ελεύθεροι να δημιουργήσουν εφαρμογές που χρησιμοποιούν οποιαδήποτε από τις βασικές λειτουργίες του τηλεφώνου όπως η αποστολή SMS, τηλεφωνικές κλήσεις, τη λήψη φωτογραφιών, το GPS κτλ. Έτσι διευκολύνονται στην ανάπτυξη πιο περίπλοκου και πιο πλούσια λειτουργικού λογισμικού. Αυτό το λειτουργικό σύστημα κινητών τηλεφώνων (ή άλλων μικρών φορητών συσκευών που συνδέονται στο διαδίκτυο) στηρίζεται στον ελεύθερο πυρήνα του Linux. Επιπλέον, η πλατφόρμα ανάπτυξης Android είναι μια πλατφόρμα multi tasking, πράγμα που σημαίνει ότι κάθε εφαρμογή μπορεί να τρέξει στο τηλέφωνο ταυτόχρονα κάποια άλλη χωρίς να επηρεαστεί η απόδοσή τους, και αυτό είναι καλύτερο από το να περιορίζεται σε μία εφαρμογή κάθε φορά. Το Android είναι μια πλατφόρμα ανοικτού κώδικα, πράγμα που σημαίνει ότι μπορεί εύκολα να επεκταθεί και να τροποποιηθεί για να συμβαδίζει και να υιοθετεί τις τελευταίες τεχνολογίες και εξελίξεις. Το γεγονός ότι και η πηγή της πλατφόρμας είναι ανοιχτή διασφαλίζει ότι η ανάπτυξη το Android θα έχει συνεχή πρόοδο και θα εξελίσσεται αφού ένας μεγάλος αριθμός ικανών android προγραμματιστών εργάζεται για τη δημιουργία ελεύθερων για χρήση προηγμένων εργαλείων λογισμικού.

Δεν υπάρχει διαφορά μεταξύ των λειτουργιών / εφαρμογών οι οποίες είναι ενσωματωμένες στο τηλέφωνο από τις εφαρμογές που δημιουργούνται και προστίθενται από τρίτους προγραμματιστές Android. Οι τελευταίες μπορούν και έχουν την ίδια πρόσβαση σε όλες τις κύριες λειτουργίες της συσκευής κάτι που επιτρέπει στους τελικούς χρήστες να απολαμβάνουν ένα ευρύ φάσμα εφαρμογών Android που μπορούν να χρησιμοποιηθούν για τη σχεδόν απεριόριστους σκοπούς. Με συσκευές χτισμένες στην πλατφόρμα Android, οι χρήστες έχουν τη δυνατότητα να προσαρμόσουν πλήρως τη συσκευή τους ανάλογα με τις ανάγκες και τις απαιτήσεις τους. Τυχόν εφαρμογές ακόμα και οι βασικές λειτουργίες μπορεί να τροποποιηθούν ή να αντικατασταθούν πλήρως από άλλες. Για παράδειγμα, ο χρήστης μπορεί να χρησιμοποιήσει την επιθυμητή του εφαρμογή για να εμφανίσει τις φωτογραφίες που είναι αποθηκευμένες στο τηλέφωνό του, ή για να έχει πρόσβαση στην αλληλογραφία του.

Οι προγραμματιστές Android μπορούν να δημιουργήσουν πολύπλοκες καινοτόμες εφαρμογές με σχεδόν απεριόριστη λειτουργικότητα. Για παράδειγμα, μια εφαρμογή μπορεί να μεταδώσει τα δεδομένα από το κινητό σας με το διαδίκτυο (κάτι που μπορεί να περιλαμβάνει το ημερολόγιο σας και τις προγραμματισμένες εκδηλώσεις, λίστα με τις επαφές, τις φωτογραφίες σας και ακόμη και την τρέχουσα θέση σας, αλλά και παραγγελίες, τιμολόγια κτλ) και να λάβει όλα όσα μπορεί να χρειαστεί online και να εμφανίζονται στην οθόνη της συσκευής.

Η πλατφόρμα παρέχει στο καθένα που ασχολείται με την ανάπτυξη εφαρμογών τη δυνατότητα χρησιμοποίησης μια μεγάλης ποικιλίας από βιβλιοθήκες και τα χρήσιμα εκείνα εργαλεία που μπορούν να χρησιμοποιηθούν για τη δημιουργία του πιο εξελιγμένου λογισμικού. Αυτή η ολοκληρωμένη δέσμη από έτοιμα εργαλεία αυξάνει σημαντικά την παραγωγικότητα των προγραμματιστών Android εφαρμογών και τους βοηθά να δημιουργήσουν εκπληκτικά πλούσιο λογισμικό γρηγορότερα και με λιγότερα λάθη.

1.2 Ανάγκη ανάπτυξης λογισμικού Android

Όταν εμφανίστηκαν τα κινητά τηλέφωνα έλυσαν βασικά προβλήματα επικοινωνίας που αντιμετωπίζαμε, όπως όταν χανόμασταν με τους φίλους μας σε μία συναυλία, όταν ξεχνούσαμε την λίστα από τα ψώνια ή όταν θα αργούσαμε σε ένα ραντεβού. Πλέον τα κινητά μας τηλέφωνα παρέχουν πολλά περισσότερα από μια

κλήση ή την αποστολή ενός μηνύματος αφού μπορούν να λύσουν όλα σχεδόν τα προβλήματά μας και βασιζόμαστε σε αυτά για τα πάντα.

Η συχνότητα χρήσης των εφαρμογών για κινητά τηλέφωνα από τους χρήστες καθ' όλη τη διάρκεια της ημέρας δημιούργησε και την ανάγκη ανάπτυξης όλο και περισσότερων εφαρμογών. Αρχικά η ανάπτυξη εφαρμογών συνοδευόταν από πολλά προβλήματα και εμπόδια τόσο από την πλευρά των χρηστών όσο και από την πλευρά των προγραμματιστών.

Τα κινητά τηλέφωνα εκείνης της εποχής είχαν εξαιρετικά μικρές οθόνες και με πολύ χαμηλή ανάλυση και σε συνδυασμό με τον περιορισμένο χώρο αποθήκευσης και την μικρή επεξεργαστική ισχύ δεν μπορούσαν να υποστηρίξουν τα εκάστοτε προγράμματα περιήγησης στο web. Τη λύση σε αυτό το πρόβλημα έδωσε το Πρωτόκολλο ασυρμάτων εφαρμογών (WAP). Το WAP ή Wireless Application Protocol σχεδιάστηκε αρχικά για να επιτρέψει στους χρήστες των κινητών τηλεφώνων να έχουν πρόσβαση, ανεξαρτήτως τόπου και χρόνου, σε διάφορες πληροφορίες που αντλούνται από ειδικά διαμορφωμένες «ιστοσελίδες», μέσα στους περιορισμούς μνήμης και εύρους ζώνης του κινητού τηλεφώνου. Φυσικά οι «ασύρματες ιστοσελίδες» δεν έχουν καμία ομοιότητα με τις ιστοσελίδες που οι χρήστες μπορούν να δουν μέσω του World Wide Web και των ηλεκτρονικών υπολογιστών.

Με την πάροδο του χρόνου και λόγω της συνεχούς ζήτησης των χρηστών για όλο και περισσότερες εφαρμογές εμφανίζονται στην αγορά διάφορες ιδιωτικές πλατφόρμες και κάποια smartphones που εκτελούσαν τα Palm OS και RIM Blackberry OS. Άλλες πλατφόρμες όπως το Symbian OS αναπτύχθηκαν από κατασκευαστές κινητών τηλεφώνων όπως οι Nokia, Sony Ericsson, Motorola και Samsung, ενώ το 2008 εμφανίστηκε το λειτουργικό σύστημα του Apple iPhone (OS X iPhone). Αποτέλεσμα της εμφάνισης όλων αυτών των διαφορετικών πλατφορμών είναι ο κατακερματισμός της αγοράς, των προγραμματιστών και των εταιριών κατασκευής κινητών τηλεφώνων.

Οι πρώτες απόπειρες της Google να μπει στον ασύρματο κόσμο στιγματίστηκε από όλα τα προβλήματα που αντιμετώπιζε η κάθε εταιρία εκείνη την εποχή. Οι ελευθερίες που παρείχε το διαδίκτυο στους χρήστες του δεν μπορούσαν να μεταφερθούν στους χρήστες κινητών τηλεφώνων, ενώ οι πρώτοι μπορούσαν να επιλέξουν ανάμεσα από ποικιλία ηλεκτρονικών υπολογιστών, λειτουργικών συστημάτων, παρόχων υπηρεσιών Internet και εφαρμογών περιήγησης στο web. Με την φιλοσοφία της να επικεντρώνεται στον χρήστη και η σχεδίασή της να ακολουθεί δημοκρατικά πρότυπα, η Google επιχειρεί να μετατρέψει την υπάρχουσα κλειστή και φυλασσομένη ασύρματη αγορά σε μια αγορά που οι χρήστες μπορούν να μετακινούνται από τον έναν φορέα στον άλλον και να έχουν ελεύθερη πρόσβαση σε εφαρμογές και υπηρεσίες. Συνδυάζοντας τις επιθυμίες των χρηστών κινητών τηλεφώνων, τις ανάγκες των προγραμματιστών των εφαρμογών κινητών τηλεφώνων και τις απαιτήσεις των κατασκευαστών κινητών συσκευών γεννιέται το εξής ερώτημα: τι θα πρέπει να γίνει, ώστε να κατασκευαστεί ένα καλύτερο κινητό τηλέφωνο;

Απάντηση σε αυτό το ερώτημα καλείται αν δώσει ο οργανισμός Open Handset Alliance (OHA) που σχηματίστηκε τον Νοέμβριο του 2007 γι' αυτόν το σκοπό και αποτελείται από κάποιες από τις μεγαλύτερες και πιο επιτυχημένες εταιρείες κινητών τηλεφώνων. Η εταιρία Android Inc. του Andy Rubin, ο οποίος θεωρείται ο πατέρας της πλατφόρμας Android, εξαγοράστηκε από την Google το 2005. Έτσι τα μέλη του OHA, σε συνεργασία με την Google, ξεκίνησαν να αναπτύσσουν μια μη ιδιωτική ανοιχτή πρότυπη πλατφόρμα η οποία βασίζεται σε τεχνολογία της Android Inc. που στοχεύει στην αντιμετώπιση των προβλημάτων που μαστίζουν την κοινότητα των κινητών τηλεφώνων. Το αποτέλεσμα είναι το Android.

Το Android έγινε αποδεκτό ως «η πρώτη ολοκληρωμένη, ανοικτή και δωρεάν πλατφόρμα κινητών τηλεφώνων». Ολοκληρωμένη γιατί οι προγραμματιστές ακολούθησαν μια περιεκτική προσέγγιση στον σχεδιασμό της πλατφόρμας, ξεκίνησαν μ' ένα ασφαλές λειτουργικό σύστημα και κατασκεύασαν ένα στιβαρό πλαίσιο λογισμικού το οποίο επιτρέπει την πλούσια ανάπτυξη εφαρμογών. Ανοιχτή γιατί παρέχεται μέσω της διαδικασίας ανοιχτής πηγής. Οι προγραμματιστές απολαμβάνουν ανεμπόδιστη πρόσβαση στα χαρακτηριστικά των συσκευών όταν αναπτύσσουν εφαρμογές. Και τέλος δωρεάν γιατί η ανάπτυξη εφαρμογών γίνεται δωρεάν, εφ' όσον δεν απαιτείται η καταβολή χρηματικών ποσών για άδειες χρήσης και πνευματικά δικαιώματα για τον προγραμματισμό σε αυτήν την πλατφόρμα, όπως και στην γενικότερη διακίνηση των εφαρμογών.

1.3 Πλατφόρμες Android: Λειτουργίες και χαρακτηριστικά



Εικόνα 1.3. 1 Cupcake[1]

Η πρώτη εμπορική έκδοσή του Android (Android SDK) κυκλοφόρησε τον Απρίλιο του 2009 με την ονομασία Cupcake, (εικόνα 1.3.1) Android 1.5 και με πυρήνα linux 2.6.27 με τα ακόλουθα χαρακτηριστικά:

- Δυνατότητα καταγραφής και να παρακολουθήσετε τα βίντεο μέσω λειτουργίαβιντεοκάμερας
- Αποστολή βίντεο στο YouTube και φωτογραφίες με το Picasa απευθείας από το τηλέφωνο
- Ένα νέο πληκτρολόγιο και με το κείμενο-πρόβλεψη
- Bluetooth A2DP και AVRCP υποστήριξη
- Δυνατότητα για αυτόματη σύνδεση με τα ακουστικά Bluetooth σε μια ορισμένη απόσταση
- Νέα widgets και τους φακέλους που να συμπληρώσετε τις οθόνες Home
- Κινούμενες μεταβάσεις οθόνης



Εικόνα 1.3. 2 Donut[1]

Η έκδοση 1.6 με την ονομασία Donut κυκλοφόρησε τον Σεπτέμβριο του 2009. Στηρίζεται στον πυρήνα 2.6.29 του linux και διαθέτει τα παρακάτω χαρακτηριστικά:

- Μια βελτιωμένη εμπειρία του Android Market
- Μια διεπαφή ενσωματωμένη φωτογραφική μηχανή, βιντεοκάμερα, και γκαλερί
- Gallery επιτρέπει πλέον στους χρήστες να επιλέξετε πολλές φωτογραφίες για διαγραφή
- Ενημέρωση αναζήτηση Voice, με ταχύτερη απόκριση και βαθύτερη ολοκλήρωση μεεγγενείς εφαρμογές, συμπεριλαμβανομένης της δυνατότητας να καλέσετε επαφές
- Ενημέρωση εμπειρία αναζήτησης για να επιτρέψει την αναζήτηση σελιδοδείκτες, την ιστορία, τις επαφές και το διαδίκτυο από την αρχική οθόνη
- Ενημέρωση για την υποστήριξη της τεχνολογίας CDMA / EVDO, 802.1x, VPNs, και ένα κείμενο σε ομιλία του κινητήρα
- Υποστήριξη για WVGA ανάλυση οθόνης
- βελτιώσεις ταχύτητα στην αναζήτηση και τη φωτογραφική μηχανή εφαρμογές
- πλαίσιο Χειρονομία και GestureBuilder εργαλείο ανάπτυξης



Εικόνα 1.3. 3 Eclair[1]

Η έκδοση 2.0/2.1 με το όνομα Eclair κυκλοφόρησε τον Οκτώβριο του 2009 με πυρήνα τον 2.6.29 (ίδιος με την έκδοση 1.6-Donut) και τα παρακάτω χαρακτηριστικά:

- Βέλτιστη ταχύτητα του υλικού

- Υποστήριξη για περισσότερα μεγέθη οθόνης και τα ψηφίσματα
- ανανεωμένο UI
- Νέα Browser UI και υποστήριξη HTML5
- Νέες λίστες επαφών
- Καλύτερη άσπρο-μαύρο λόγος για φόντο
- Βελτιωμένη Google Maps 3.1.2
- Microsoft Exchange υποστήριξη
- Χτισμένο σε υποστήριξη flash για φωτογραφική μηχανή
- Digital Zoom
- Τάξη MotionEvent ενισχυμένη για να παρακολουθείτε multi-touch γεγονότα
- Βελτιωμένη εικονικό πληκτρολόγιο
- Bluetooth 2.1
- ζωντανά Wallpapers



Εικόνα 1.3. 4 Froyo[1]

Ακολουθεί η έκδοση 2.2 με το όνομα Froyo κυκλοφόρησε τον Μάιο του 2010, στηριζόμενη στον πυρήνα 2.6.32 και διαθέτει τα παρακάτω:

- Γενική Android OS ταχύτητα, τη μνήμη, και βελτιστοποιήσεις επιδόσεων
- Πρόσθετες βελτιώσεις ταχύτητα ευγένεια εφαρμογή της ΚΟΕ εφαρμογή
- Ένταξη των V8 JavaScript μηχανή του Chrome στην εφαρμογή περιήγησης
- Αυξημένη υποστήριξη Microsoft Exchange (πολιτικές ασφάλειας, αυτόματη ανακάλυψη, GAL look up, συγχρονισμό ημερολογίου, απομακρυσμένης διαγραφής)
- Βελτιωμένη εκκίνησης εφαρμογών με συντομεύσεις στο τηλέφωνο και Browser εφαρμογές
- USB πρόσδεση και WiFi hotspot λειτουργικότητα
- Προστέθηκε μια επιλογή για να απενεργοποιήσετε την πρόσβαση σε δεδομένα μέσω δικτύου κινητής τηλεφωνίας
- Ενημέρωση αίτηση αγοράς με την παρτίδα και αυτόματες λειτουργίες ενημέρωσης
- Γρήγορη εναλλαγή μεταξύ πολλών γλωσσών πληκτρολογίου και τα λεξικά τους
- Φωνητική κλήση και κοινή χρήση επαφών μέσω Bluetooth
- Υποστήριξη για τις αριθμητικές και αλφαριθμητικών κωδικών πρόσβασης
- Υποστήριξη για αυτό το αρχείο πεδία ανεβάσετε στην εφαρμογή περιήγησης
- Υποστήριξη για την εγκατάσταση εφαρμογών στο επεκτάσιμη μνήμη
- Adobe Flash 10,1 στήριξη



Εικόνα 1.3. 5 Gingerbread[1]

Στη συνέχεια έχουμε την έκδοση 2.3 με όνομα GingerBread η οποία κυκλοφόρησε στις 6 Δεκεμβρίου του 2010 και έχει τις εξής αλλαγές:

- Ενημερωμένο UI Design

- Υποστήριξη για πολύ μεγάλα μεγέθη οθονών και αναλύσεων(WXGA και μεγαλύτερες)
- Προεγκατεστημένη υποστήριξη για VoIP τηλεφωνία.
- Υποστήριξη για WebM/VP8 αναπαραγωγή βίντεο αλλά και AAC κωδικοποίηση ήχου.
- Νέα ηχητικά εφέ όπως τα reverb, equalization, headphone virtualization, και bass boost.
- Επανασχεδιασμένο Multi-touch πληκτρολόγιο.
- Λειτουργίες αντιγραφής και επικόλλησης σε όλο το λειτουργικό.
- Αυξημένη υποστήριξη για ανάπτυξη εφαρμογών σε περιβάλλον Android.
- Βελτιώσεις ήχου και γραφικών για τους προγραμματιστές παιχνιδιών.
- Προεγκατεστημένη υποστήριξη για περισσότερους αισθητήρες(όπως γυροσκόπιο και βαρόμετρο).
- Διαχείριση λήψεων(Download manager) για λήψη μεγάλων αρχείων.
- Βελτιωμένη διαχείριση ενέργειας και έλεγχος των εφαρμογών.
- Προεγκατεστημένη υποστήριξη για πολλαπλές κάμερες.
- Μετατόπιση από το YAFFS σε ext4 filesystem.

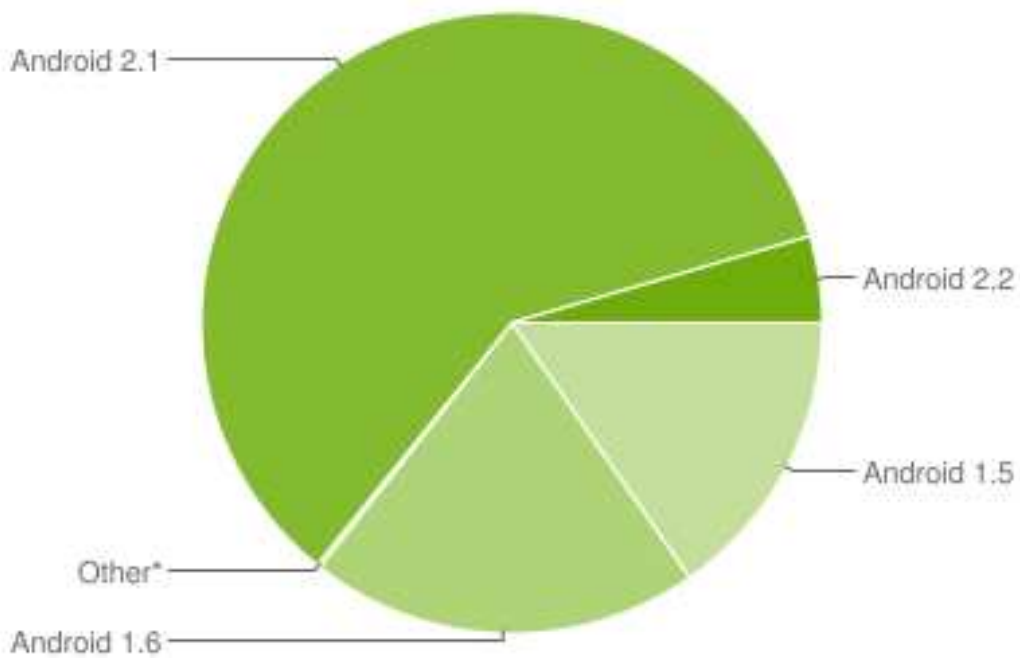
Ακολούθησαν δύο updates για τη συγκεκριμένη έκδοση, τα Android 2.3.3 και Android 2.3.4.

Τόσο η έκδοση Android 3.0 με το όνομα Honeycomb όσο και τα updates αυτής Android 3.1 και Android 3.2 αφορούν tablets.

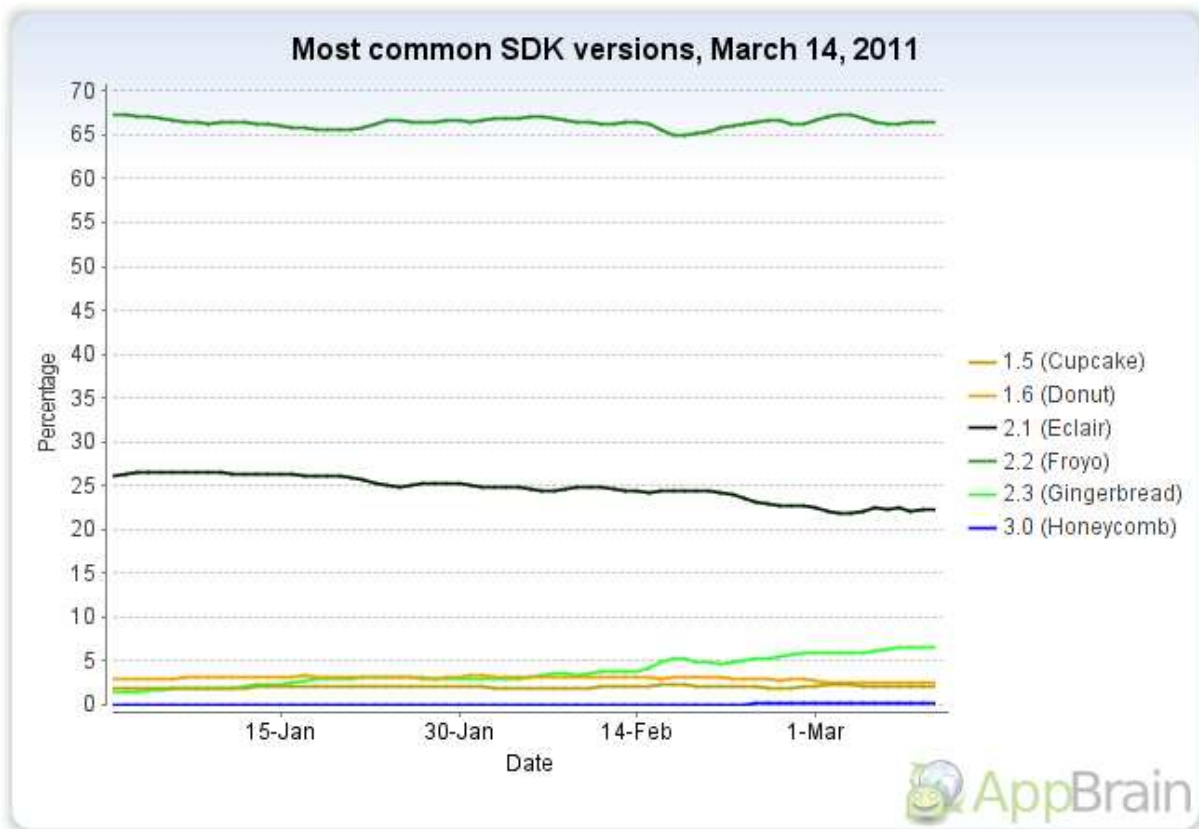
Η τελευταία έκδοση του Android είναι η 4.0 με όνομα Ice Cream Sandwich είναι η πρώτη που καλύπτει τις ανάγκες των κινητών τηλεφώνων αλλά και των tablets. Παρουσιάστηκε το 2011 και κάποια από τα σημαντικότερα χαρακτηριστικά του είναι:

- Face Unlock: Το λειτουργικό ενσωματώνει πλέον τεχνολογία αναγνώρισης προσώπου, που επιτρέπει στο χρήστη να ξεκλειδώνει τη συσκευή απλά και μόνο σκανάροντας το πρόσωπό του με την εμπρόσθια κάμερα.
- Android Beam: Λειτουργία που βασίζεται στο near field communication (NFC) και επιτρέπει στους χρήστες με συμβατές συσκευές να διαμοιράζονται επαφές, sites, apps, χάρτες κλπ.
- Φωνητική πληκτρολόγηση: Λειτουργία που ήδη υπήρχε και τώρα δείχνει να είναι βελτιωμένη.
- Google+: Η νέα υπηρεσία κοινωνικής δικτύωσης ενσωματώνεται στο νέο λειτουργικό και ο χρήστης μπορεί να χρησιμοποιήσει Hangouts για video chat με 9 άτομα ταυτόχρονα.
- Camera Software: Η Google ισχυρίζεται ότι δεν υπάρχει πλέον shutter lag και για να το αποδείξει χρησιμοποίησε το ολοκαίνουριο Samsung Galaxy Nexus.
- Λεπτομέρειες mobile data: Ένα ενσωματωμένο εργαλείο που δίνει τη δυνατότητα για αποτελεσματική παρακολούθηση και διαχείριση των δεδομένων που διακινούνται μέσω του provider.
- Κατάργηση menu button: Το κουμπί του μενού ήταν πάντα ένα μυστήριο στο Android, καθώς το περιεχόμενό του ήταν διαφορετικό σε κάθε περίπτωση. Φαίνεται πώς τώρα εγκαταλείπεται και αντικαθίσταται από χειρισμό που "δένει" με την εκάστοτε οθόνη.

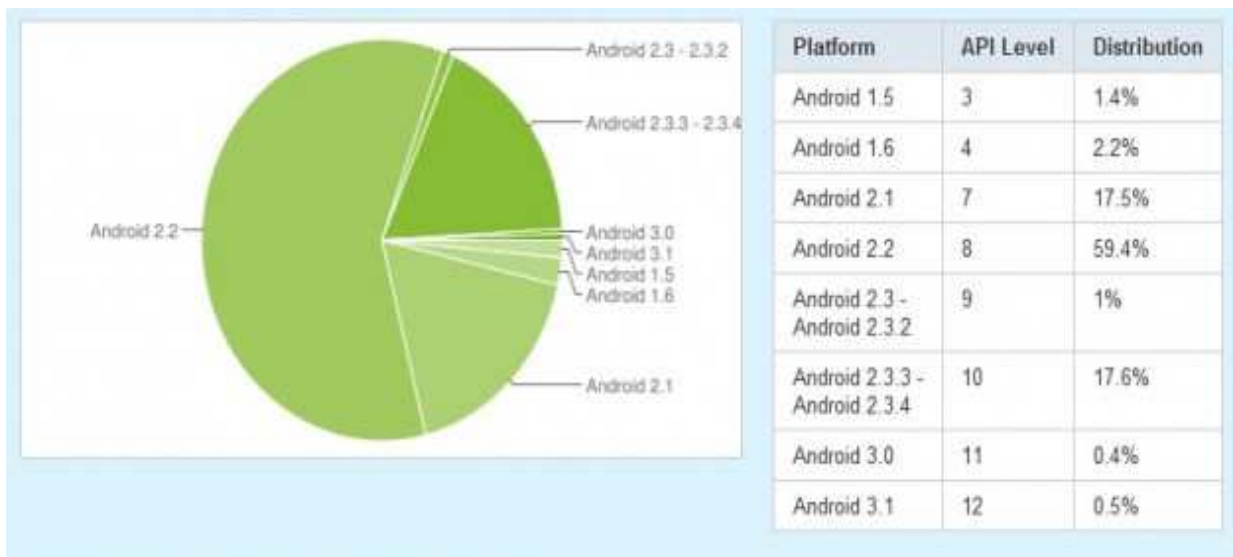
Ακολουθούν κάποια διαγράμματα που αφορούν στατιστικά δεδομένα σχετικά με τη χρήση των εκδόσεων Android.



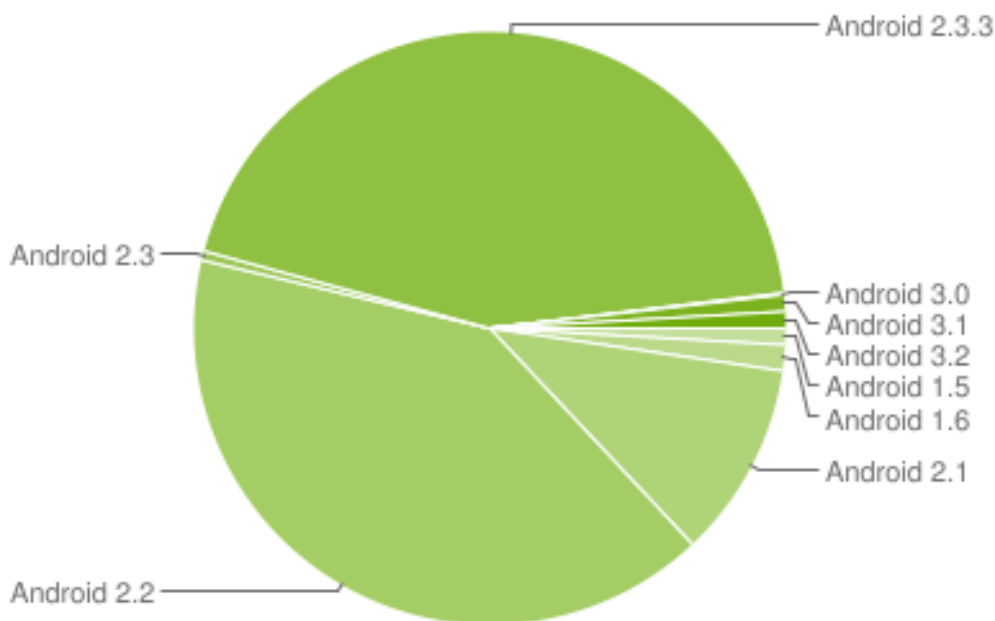
Εικόνα 1.3. 6 Στατιστικά Αυγούστου 2010



Εικόνα 1.3. 7 Στατιστικά Μαρτίου 2011



Εικόνα 1.3. 8 Στατιστικά Ιουλίου 2011



Εικόνα 1.3. 9 Στατιστικά Νοεμβρίου 2011

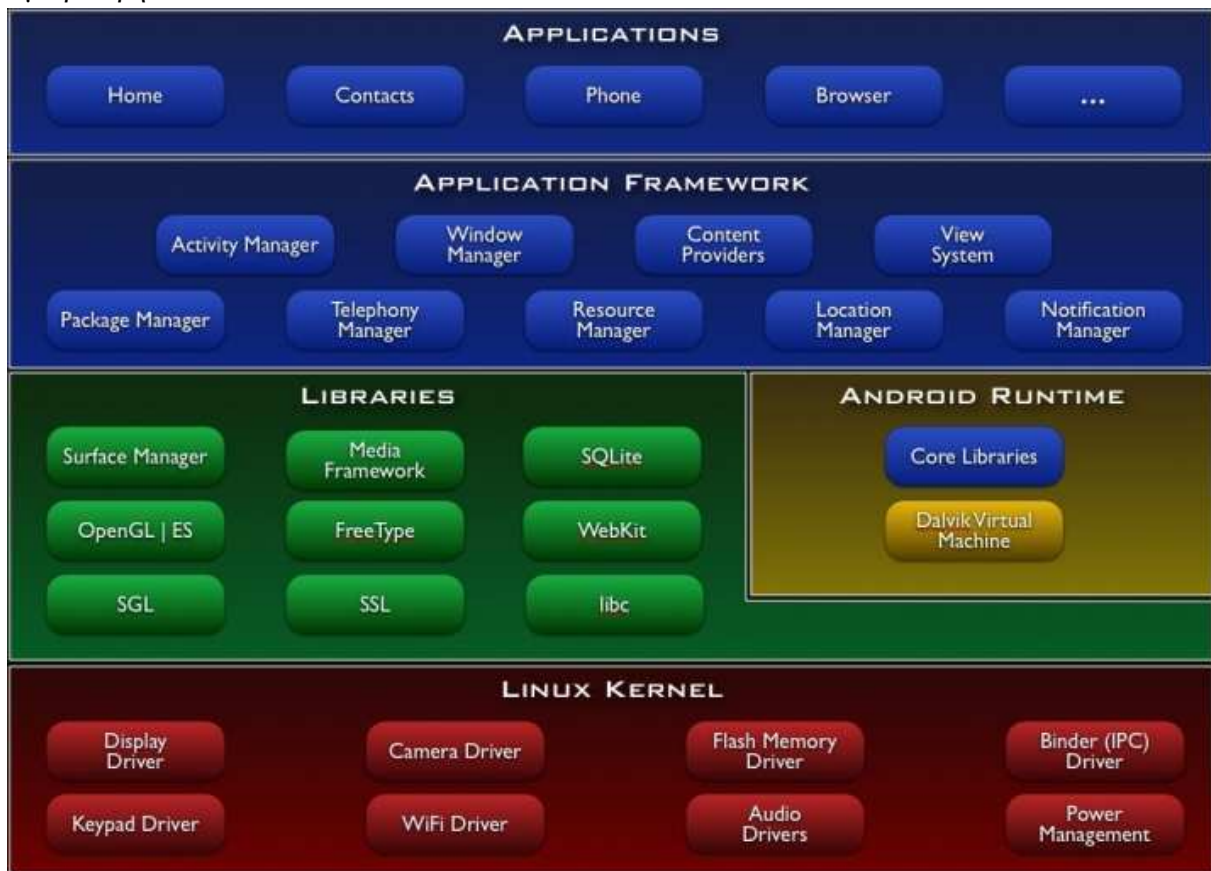
Οι στατιστικές συλλέγονται ανά τακτά χρονικά διαστήματα και αυτό που μπορούμε να παρατηρήσουμε είναι η αισθητή αλλαγή των προτιμήσεων των χρηστών σχετικά με τις εκδόσεις του android. Παρατηρείστε ότι μέσα σε σχεδόν ενάμιση χρόνο η έκδοση 2.1 έχει μειωθεί περίπου 70%.

1.4 Αρχιτεκτονική Android

Τα βασικά συστατικά του λειτουργικού συστήματος Android είναι οι εφαρμογές, τα πλαίσια εφαρμογών, οι βιβλιοθήκες, το περιβάλλον εκτέλεσης εφαρμογών Android (Android Runtime) και το κέλυφος (Kernel) Linux. Αναλυτικότερα οι λειτουργίες φαίνονται στο ακόλουθο διάγραμμα.

Η πλατφόρμα Android σχεδιάστηκε έτσι ούτως ώστε να είναι πιο ανθεκτική σε σφάλματα από τους προκατόχους της. Η τηλεφωνική συσκευή, για την οποία προορίζονται εφαρμογές Android, βασίζεται σε λειτουργικό σύστημα Linux, πάνω στο οποίο εκτελούνται με ασφαλή τρόπο. Κάθε εφαρμογή λειτουργεί την

δική της εικονική μηχανή. Οι εφαρμογές Android είναι κώδικας υπό διαχείριση, δηλαδή είναι λιγότερο πιθανό να προκαλέσουν το κρυστάρισμα του τηλεφώνου και η πιθανότητα «κολλήματος» της συσκευής είναι πολύ μικρότερη.



Εικόνα 1.4. 1 Αρχιτεκτονική πλατφόρμας Android[2]

1.4.1 Εφαρμογές (Applications)

Το Android κυκλοφορεί με ένα σύνολο βασικές εφαρμογές, περιλαμβάνοντας έναν email client, μια εφαρμογή SMS, ημερολόγιο και άλλες. Όλες οι εφαρμογές γράφονται στην γλώσσα προγραμματισμού Java.

1.4.2 Πλαίσιο εφαρμογών (Application framework)

Με την παροχή μίας ανοιχτής πλατφόρμας ανάπτυξης, το Android προσφέρει στους προγραμματιστές τη δυνατότητα να κατασκευάσουν εξαιρετικά πλούσιες και καινοτόμες εφαρμογές. Οι προγραμματιστές μπορούν να επωφεληθούν από το υλικό της συσκευής, να έχουν πληροφορίες για την τοποθεσία, να εκτελούν υπηρεσίες στο παρασκήνιο, να θέτουν συναγερμούς, να προσθέτουν ειδοποιήσεις στη μπάρα κατάστασης και πολλά άλλα.

Οι προγραμματιστές έχουν πλήρη πρόσβαση στις ίδιες εφαρμογές που χρησιμοποιήθηκαν από τις βασικές εφαρμογές. Η αρχιτεκτονική της εφαρμογής έχει σχεδιαστεί για να απλοποιήσει την επαναχρησιμοποίηση των συστατικών (components). Κάθε εφαρμογή μπορεί να δημοσιεύσει τις δυνατότητες της και έτσι οποιαδήποτε άλλη εφαρμογή μπορεί να κάνει χρήση αυτών (με την επιφύλαξη περιορισμών ασφαλείας που επιβάλλονται από το πλαίσιο - framework). Ο ίδιος μηχανισμός επιτρέπει στο χρήστη να αντικαταστήσει τα συστατικά, components.

Η βάση των εφαρμογών περιέχει ένα σύνολο υπηρεσιών και συστημάτων, συμπεριλαμβανομένων των:

- Ένα πλούσιο και επεκτάσιμο σύνολο από Views που μπορούν να χρησιμοποιηθούν για τη δημιουργία μιας εφαρμογής, όπως λίστες (lists), πλέγματα (grids), πλαίσια κειμένου (text boxes), κουμπιά (buttons), ακόμα και έναν embeddable web browser.
- Content Providers (Παρόχους Περιεχομένου) που επιτρέπουν στις εφαρμογές να έχουν πρόσβαση σε δεδομένα από άλλες εφαρμογές (όπως οι επαφές) ή να μοιράζονται τα δικά τους δεδομένα.
- Έναν Resource Manager (Διαχειριστή Πόρων) που παρέχει πρόσβαση σε πηγές χωρίς κώδικα (non-code resources) όπως strings με βάση την τοποθεσία, τα γραφικά και τα αρχεία διάταξης (layout files).
- Έναν Notification Manager (Διαχειριστή Ειδοποιήσεων) που επιτρέπει σε όλες τις εφαρμογές να απεικονίζουν προσαρμοσμένες ειδοποιήσεις στη μπάρα κατάστασης.

- Έναν Activity Manager (Διαχειριστή Δραστηριοτήτων) που διαχειρίζεται τον κύκλο ζωής (lifecycle) των εφαρμογών και παρέχει ένα συνηθισμένο ιστορικό μετάβασης (navigation backstack).

1.4.3 Βιβλιοθήκες (Libraries)

Το Android περιλαμβάνει ένα σύνολο από βιβλιοθήκες C/C++ που χρησιμοποιούνται από διάφορα components του συστήματος. Αυτές οι δυνατότητες εκτίθενται στους προγραμματιστές μέσω του πλαισίου εφαρμογών Android. Μερικές από τις βασικές βιβλιοθήκες αναφέρονται παρακάτω:

- System C library – μία υλοποίηση προερχόμενη από το BSD, της επίσημης βιβλιοθήκης συστήματος C (libc), βελτιστοποιημένη για ενσωματωμένες συσκευές που βασίζονται στο Linux.
- Media Libraries – βασισμένες στο OpenCORE της PacketVideo. Οι βιβλιοθήκες υποστηρίζουν την αναπαραγωγή και καταγραφή πολλών δημοφιλών μορφών ήχου και βίντεο, καθώς και στατικών αρχείων εικόνας. Συμπεριλαμβάνονται τα MPEG4, H.264, MP3, AAC, AMR, JPG, και PNG.
- Surface Manager – διαχειρίζεται την πρόσβαση στο υποσύστημα απεικόνισης και συνθέτει στρώματα 2D και 3D γραφικών από πολλαπλές εφαρμογές.
- LibWebCore – μία σύγχρονη μηχανή web browser θ οποία χρησιμοποιείται και από τον Android browser και από τον embeddable web browser.
- SGL – η βασική μηχανή 2D γραφικών.
- 3D libraries – μια υλοποίηση βασισμένη στις εφαρμογές του OpenGL ES 1.0. Οι βιβλιοθήκες χρησιμοποιούν είτε την επιτάχυνση υλικού για 3D (όπου είναι διαθέσιμο) ή το πολύ καλά βελτιστοποιημένο λογισμικό απεικόνισης 3D (rasterizer).
- FreeType – bitmap και vector γραμματοσειρά φωτοσκίασης.
- SQLite – μία ισχυρή και ελαφριά σχεσιακή βάση δεδομένων, διαθέσιμη σε όλες τις εφαρμογές.

1.4.4 Περιβάλλον Android (Runtime Android)

- Core Libraries

Το Android περιλαμβάνει ένα σύνολο βασικών βιβλιοθηκών που παρέχουν τις περισσότερες από τις διαθέσιμες λειτουργίες των βασικών βιβλιοθηκών της Java.

- Dalvik Virtual Machine

Η Dalvik Virtual Machine πήρε το όνομά της από τον Bornstein. Η Dalvik είναι μία εικονική μηχανή διερμηνέας, η οποία εκτελεί αρχεία της μορφής *.dex (Dalvik Executable), μια μορφή που είναι βελτιστοποιημένη για αποδοτική αποθήκευση και εκτέλεση με χαρτογραφημένη μνήμη (memory-mappable). Η εικονική μηχανή βασίζεται σε καταχωρητές και μπορεί να τρέξει κλάσεις που μεταγλωττίστηκαν από έναν Java compiler και έχουν μετασχηματιστεί στη δική της φυσική μορφή, χρησιμοποιώντας το παρεχόμενο εργαλείο “dx”. Η VM τρέχει πάνω στον πυρήνα του Linux 2.6, στον οποίο βασίζεται για την υποκείμενη λειτουργικότητα (όπως η διαχείριση απειλών -threads- και η διαχείριση μνήμης σε χαμηλό επίπεδο). Κάθε εφαρμογή Android τρέχει στη δική της διαδικασία (process), με το δικό της στιγμιότυπο (instance) της Dalvik VM. Η DalvikVM βελτιστοποιήθηκε επίσης για να τρέχει σε πολλαπλά στιγμιότυπα με πολύ μικρή χρήση μνήμης. Μια σειρά από VM προστατεύουν μια εφαρμογή από το να υπολειπούν εξαιτίας μιας άλλης εφαρμογής που «κόλλησε».

Διαφορές από μια κανονική JavaVM.

Η JavaVM , που είναι πλέον εγκατεστημένη σχεδόν σε όλους τους προσωπικούς υπολογιστές, είναι βασισμένη σε «στοίβες» (stack based). Η DalvikVM από την άλλη είναι βασισμένη σε καταχωρητές (registered based), γιατί οι επεξεργαστές για συσκευές όπως τα κινητά είναι βελτιστοποιημένοι για εκτέλεση εφαρμογών με χρήση καταχωρητών. Επίσης οι VM που βασίζονται σε καταχωρητές επιτρέπουν γρηγορότερη εκτέλεση συχνά σε βάρος προγραμμάτων που είναι μεγαλύτερα σε μέγεθος μετά την μεταγλώττιση τους (compilation).

1.4.5 Κέλυφος Kernel (Linux Kernel)

Το κέλυφος Linux χειρίζεται τις βασικές υπηρεσίες του συστήματος και ενεργεί ως επίπεδο αφαίρεσης υλικού (HAL, hardware abstraction layer) ανάμεσα στο φυσικό υλικό της συσκευής και στη στοιβία λογισμικού Android. Μερικές από τις λειτουργίες που χειρίζεται το κέλυφος Kernel είναι:

- Η επιβολή δικαιωμάτων χρήσης και ασφάλειας των εφαρμογών,

- Η διαχείριση μνήμης χαμηλού επιπέδου,
- Η διαχείριση των διεργασιών και η αλληλουχία των ενεργειών,
- Η στοίβα δικτύου,
- Η οθόνη, η είσοδος από το πληκτρολόγιο, η κάμερα, το Wi Fi, η μνήμη flash και η πρόσβαση του προγράμματος οδήγησης λειτουργίας σύνδεσης (επικοινωνία μεταξύ των διεργασιών).

1.5 Λογισμικά για κινητές συσκευές

Ένα λειτουργικό σύστημα για κινητές πλατφόρμες (Mobile OS) είναι το λειτουργικό σύστημα που τρέχει μια κινητή πλατφόρμα. Έχει τις ίδιες χαρακτηριστικές αρχές με ένα λειτουργικό σύστημα όπως αυτά των ηλεκτρονικών υπολογιστών, με βασική διαφορά ότι είναι πιο «ελαφριά» αφού είναι φτιαγμένα να λειτουργούν με λιγότερους υπολογιστικούς πόρους ενώ έχουν να κάνουν περισσότερο με ασύρματες επικοινωνίες και τοπικά δίκτυα, με διαφορετικά αρχεία πολυμέσων και διαφορετικούς τρόπους εισαγωγής εντολών. Τυπικά παραδείγματα τέτοιων συσκευών που χρησιμοποιούν τέτοια λειτουργικά συστήματα είναι τα smartphones, τα pda, οι ταμπλέτες και γενικώς συσκευές που συνήθως ονομάζουμε έξυπνες συσκευές και άλλες κινητές πλατφόρμες

1.5.1 Palm OS

Θα μπορούσαμε να χαρακτηρίσουμε το Palm OS ως το πρώτο λειτουργικό σύστημα για κινητή πλατφόρμα, το οποίο εμφανίστηκε στην αγορά το 1996. Το Palm OS αναπτύχθηκε από την Palm και χρησιμοποιήθηκε σε pda, σχεδιάστηκε με βάση την ευκολία χρήσης μιας οθόνης αφής και βασίστηκε στην αλληλεπίδραση του χρήστη με αυτήν μέσω ενός γραφικού περιβάλλοντος. Προσέφερε μια σουίτα με βασικές εφαρμογές με σκοπό την διαχείριση προσωπικών δεδομένων.

Με την πάροδο των χρόνων, νέες εκδόσεις του λειτουργικού συστήματος υποστήριξαν και smartphones. Υπήρξαν διάφορες εκδόσεις μέχρι και το 2004, με κάθε μια από αυτές να προσθέτει όλο και περισσότερες λειτουργίες με σκοπό την προσαρμογή του λειτουργικού συστήματος στις νέες απαιτήσεις της αγοράς.

1.5.2 Windows mobile

Τα Windows mobile είναι ένα λειτουργικό σύστημα για κινητές πλατφόρμες που αναπτύχθηκε από την Microsoft και χρησιμοποιείται σε smartphones και pda. Στις μέρες μας έχουν αντικατασταθεί από τα Windows phone 7. Η τελευταία έκδοση των Windows mobile, τα Windows mobile 6.5, είναι βασισμένα στα Windows CE και περιλαμβάνουν μια σουίτα από βασικές εφαρμογές που έχουν αναπτυχθεί από την Microsoft. Τα Windows mobile υποστηριζόντουσαν και από άλλες εταιρίες, που ανέπτυσαν εφαρμογές, και ήταν σχεδιασμένα έτσι ώστε να μοιάζουν όσο το δυνατόν περισσότερο με την έκδοση για ηλεκτρονικούς υπολογιστές των Windows, από άποψη λειτουργιών αλλά και από αισθητικής πλευράς. Τις πρώτες μέρες που εμφανίστηκαν στην αγορά, όπου και χρησιμοποιήθηκαν σε pocket pc, οι περισσότερες συσκευές που έτρεχαν Windows mobile είχαν έναν στυλό (stylus) έτσι ώστε να είναι πιο εύκολη η χρήση της οθόνης αφής. Η Microsoft το 2010 ανακοίνωσε ένα νέο λειτουργικό σύστημα για smartphones, τα Windows phone 7, και παράλληλα σταμάτησε την υποστήριξη των κινητών τηλεφώνων που χρησιμοποιούν Windows mobile.

Το μερίδιο που κατέχουν τα Windows Mobile στην αγορά τα τελευταία χρόνια γίνεται όλο και μικρότερο και σύμφωνα με τις τελευταίες έρευνες έχει φτάσει να είναι μόλις το πέμπτο πιο δημοφιλές κινητό λειτουργικό σύστημα με μόλις το 5% των χρηστών smartphones παγκοσμίως.

1.5.3 Blackberry RIM

Το Blackberry OS εμφανίστηκε το 2005 από την εταιρία Research in Motion και χρησιμοποιείται στα smartphones Blackberry.

Το λειτουργικό αυτό σύστημα δίνει την δυνατότητα χρήσης πολλαπλών εφαρμογών ταυτόχρονα και είναι ειδικά φτιαγμένο ώστε να υποστηρίζει συγκεκριμένες συσκευές εισόδου δεδομένων που χρησιμοποιεί η Research in Motion στα κινητά της τηλέφωνα όπως το trackwheel και το trackball.

Η πλατφόρμα Blackberry είναι αρκετά γνωστή για την υποστήριξή της σε εταιρικές εφαρμογές όπως email και για αυτό το λόγο χρησιμοποιείται κυρίως σε εταιρικό επίπεδο. Αυτός ήταν εξάλλου και ο αρχικός στόχος του λειτουργικού. Τον τελευταίο χρόνο έχει αυξήσει σε μεγάλο βαθμό την υποστήριξή του από τρίτες εταιρίες ανάπτυξης λογισμικού με αποτέλεσμα το ηλεκτρονικό κατάστημα πώλησης εφαρμογών του, το Blackberry App World να διαθέτει προς εγκατάσταση πάνω από 20.000 εφαρμογές. Για την ανάπτυξη εφαρμογών στην συγκεκριμένη πλατφόρμα χρησιμοποιείται το IDE Eclipse ενώ η γλώσσα προγραμματισμού είναι η Java.

1.5.4 iOS

Το iOS είναι το λειτουργικό σύστημα για κινητές πλατφόρμες της Apple. Αν και αρχικά αναπτύχθηκε μόνο για το iPhone έχει από τότε επεκταθεί ώστε να υποστηρίζει και άλλες συσκευές της Apple όπως τα iPod Touch και τα iPad. Το συγκεκριμένο λειτουργικό σύστημα δεν υποστηρίζει άλλες συσκευές εκτός από αυτές της Apple. Στο τελευταίο τετράμηνο του 2010 το iOS κατείχε το 16% της αγοράς των smartphones. Το περιβάλλον χρήσης του είναι βασισμένο στην άμεση αλληλεπίδραση του χρήστη με την οθόνη αφής πολλαπλών σημείων. Με αυτόν τον τρόπο ο χειρισμός γίνεται πολύ ευχάριστος, γρήγορος αλλά και απλός για τον χρήστη αφού υπάρχει άμεση αλληλεπίδραση με τα αντικείμενα που προβάλλονται στην οθόνη. Για παράδειγμα ο χρήστης μέσω της οθόνης αφής πολλαπλών σημείων μπορεί να χρησιμοποιεί διάφορες κινήσεις των δακτύλων του και να παίρνει άμεσα τα αποτελέσματα στην οθόνη. Μπορεί να ζουμάρει σε μια φωτογραφία με το άνοιγμα των δυο δακτύλων του ή μπορεί να αλλάζει φωτογραφίες με μια απλή κίνηση του δακτύλου του από δεξιά προς τα αριστερά. Αυτός ο απλός τρόπος χρήσης έκανε το λειτουργικό σύστημα να ξεχωρίζει σε σχέση με τον ανταγωνισμό ειδικά την περίοδο που παρουσιάστηκε στις αρχές του 2007. Είναι βασισμένο πάνω στα Mac OS X οπότε από την φύση του είναι και βασισμένο στα UNIX. Το λειτουργικό σύστημα χρησιμοποιεί περίπου 500 MB μνήμης από τον αποθηκευτικό χώρο της συσκευής. Το iOS, για την ανάπτυξη εφαρμογών στο περιβάλλον του, χρησιμοποιεί το λογισμικό ανάπτυξης εφαρμογών iOS SDK το οποίο αναπτύχθηκε από την Apple και δόθηκε στους προγραμματιστές τον Φεβρουάριο του 2008. Τους δίνει την δυνατότητα να δημιουργήσουν εφαρμογές και να τις δοκιμάσουν σε ένα εξομοιωτή που ονομάζεται iPhone Simulator. Όμως για την εγκατάσταση μια εφαρμογής στη συσκευή, καθώς και για την πώληση της μέσω του App Store πρέπει ο χρήστης να είναι εγγεγραμμένος στο πρόγραμμα των προγραμματιστών iPhone που κοστίζει 99 ευρώ τον χρόνο. Ο δημιουργός μιας εφαρμογής μπορεί να την πουλήσει σε οποιαδήποτε τιμή πάνω από την μικρότερη επιτρεπτή τιμή (0.99 ευρώ) και να έχει κέρδος το 70% αυτής, με το υπόλοιπο 30% να αντιστοιχεί στο κέρδος της Apple. Εναλλακτικά, μπορεί να δίνει την εφαρμογή δωρεάν και να μην ζημιώνεται καθόλου από τα έξοδα κυκλοφορίας και διανομής, εκτός βεβαίως από τα έξοδα εγγραφής. Το iOS SDK χρησιμοποιεί τον ίδιο πρόγραμμα γραφής κώδικα που χρησιμοποιεί και το Mac OS X, το Xcode, και περιλαμβάνει και τον iPhone Simulator, ένα πρόγραμμα που μπορεί να χρησιμοποιηθεί για να εξομοιώσει το πως θα φαίνονταν οι εφαρμογές και το πως θα δούλευαν αν έτρεχαν στο iPhone, και όλα αυτά από υπολογιστή του προγραμματιστή. Το SDK της Apple έχει ως απαιτήσεις συστήματος για να χρησιμοποιηθεί, έναν Intel Mac με λειτουργικό σύστημα Mac OS X Leopard ή και νεότερο. Όταν έχουμε να κάνουμε με άλλα λειτουργικά όπως τα Windows αλλά και παλιότερες εκδόσεις Mac OS X τα πράγματα περιπλέκονται αφού δεν υπάρχει άμεση υποστήριξη.

1.5.5 Android

Το Android είναι λειτουργικό σύστημα 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. Τα Android αρχικά αναπτύχθηκαν από μια μικρή εταιρία λογισμικού η οποία εξαγοράστηκε από την Google Η πρώτη παρουσίαση της πλατφόρμας Android έγινε τον Νοέμβριο του 2007, παράλληλα με την

ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance, μιας κοινοπραξίας 79 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.

Μια μεγάλη κοινότητα προγραμματιστών ασχολείται με τον προγραμματισμό στο Android και με αυτό τον τρόπο αυξάνει τις δυνατότητες των συσκευών που το χρησιμοποιούν. Αυτή την στιγμή υπάρχουν πάρα πολλές εφαρμογές, οι οποίες αυξάνονται καθημερινά, στο Android Market, το ηλεκτρονικό κατάστημα που έχει φτιάξει η Google, αν και υπάρχει και η δυνατότητα αγοράς εφαρμογών και από τρίτες εταιρίες. Από την στιγμή της εισόδου του στην αγορά το Android έχουν παρουσιάσει μια τεράστια αύξηση και στον αριθμό των συσκευών που το χρησιμοποιούν αλλά και του μεριδίου του στην αγορά, και αυτή την στιγμή θεωρείται το πιο διαδεδομένο λειτουργικό σύστημα για smartphones.

Για την ανάπτυξη εφαρμογών στο περιβάλλον του λειτουργικού χρησιμοποιείται το Android SDK (Software Development Kit) το οποίο περιλαμβάνει ένα μεγάλο σετ από εργαλεία ανάπτυξης. Αυτή την στιγμή οι πλατφόρμες που υποστηρίζονται περιλαμβάνουν υπολογιστές που χρησιμοποιούν Linux (οποιαδήποτε μοντέρνα έκδοση), Mac OS X 10.4.9 ή νεότερο, Windows XP ή νεότερο. Το επίσημο περιβάλλον ανάπτυξης είναι το Eclipse με ταυτόχρονη χρησιμοποίηση των ADT (Android Development Tools) αν και δίνεται η δυνατότητα χρησιμοποίησης οποιουδήποτε κειμενογράφου για την σύνταξη κώδικα Java ή XML και μέσω της γραμμής εντολών, η δημιουργία, κτίσιμο και debug εφαρμογών για Android αλλά και η δυνατότητα ελέγχου των συσκευών Android που έχουν συνδεθεί στον υπολογιστή.

Με κάθε νέα έκδοση του λειτουργικού συστήματος δημιουργείται και μια νέα έκδοση του SDK, με την νέα έκδοση να μην σταματάει την υποστήριξη για ανάπτυξη εφαρμογών για την προηγούμενη έκδοση του λειτουργικού. Όμως υπάρχουν και άλλοι τρόποι δημιουργίας εφαρμογών για το Android όπως το NDT (Native Development Kit) το οποίο μπορεί να συντάξει βιβλιοθήκες γραμμένες σε C και άλλες γλώσσες προγραμματισμού σε κώδικα που χρησιμοποιούν οι επεξεργαστές ARM. Μια από τις καινοτομίες της πλατφόρμας Android είναι η δημιουργία εφαρμογών με το App Inventor, ένα περιβάλλον ανάπτυξης προγραμμάτων το οποίο βασίζεται σε Web τεχνολογίες και προορίζεται για νέους προγραμματιστές. Είναι κάτι που δείχνει τα προτερήματα ενός λειτουργικού που έχει τόσο ανοικτή αρχιτεκτονική.

1.5.6 Symbian OS

Το Symbian OS είναι λειτουργικό σύστημα για φορητές συσκευές το οποίο αποτελεί εξέλιξη του λειτουργικού συστήματος EPOC από την Psion. Το Symbian OS δημιουργήθηκε με τη γλώσσα προγραμματισμού C++ από τη Symbian Ltd. Πριν το 2009 το Symbian OS υποστήριζε διαφορετικά περιβάλλοντα χρήστη. Όμως με την δημιουργία του Symbian Platform, το ίδιο έτος, τα 3 βασικά περιβάλλοντα χρήστη ενώθηκαν σε ένα, το οποίο εξαγοράστηκε από την Nokia και στην συνέχεια μετατράπηκε σε λογισμικό ανοικτού κώδικα. Αν και οι συσκευές με λογισμικό Symbian εξακολουθούν να πωλούνται σε μεγάλους αριθμούς στην αγορά, τα τελευταία χρόνια το μερίδιο του λειτουργικού αυτού συστήματος στην αγορά μειώνεται.

Για την ανάπτυξη εφαρμογών στο περιβάλλον του λειτουργικού υπάρχει το Symbian SDK το οποίο χρησιμοποιεί ως γλώσσα προγραμματισμού την C++ σε συνδυασμό με το Qt, ένα πλαίσιο εφαρμογών που χρησιμοποιείται από πολλές πλατφόρμες. Μπορεί να χρησιμοποιηθεί είτε με το Qt Creator είτε με το Carbide, ένα παλιότερο IDE που χρησιμοποιείται για ανάπτυξη εφαρμογών Symbian. Ένας εξομοιωτής χρησιμοποιείται, για τη δοκιμή των εφαρμογών, που τρέχει τον κώδικα απευθείας αντί να προσομοιώνει την λειτουργία του κινητού τηλεφώνου.

1.5.7 Windows phone 7

Τον Φεβρουάριο του 2010, η Microsoft ανακοίνωσε τον διάδοχο των Windows mobile, την νέα γενιά λειτουργικών συστημάτων για κινητές πλατφόρμες Windows phone 7. Το νέο λειτουργικό σύστημα περιλαμβάνει ένα εντελώς νέο περιβάλλον χρήσης το οποίο έχει δημιουργηθεί με μια γλώσσα σχεδίασης της ίδιας της εταιρίας, που ονομάζεται Metro. Παρέχει πλήρη υποστήριξη των υπηρεσιών της Microsoft όπως

το Windows live, το Zune, το Xbox live και το Bing, αλλά και υπηρεσιών τρίτων εταιριών όπως το Facebook και τα Google accounts.

Αν και αυτή την στιγμή το νέο λειτουργικό βρίσκεται στα πρώτα του βήματα στην αγορά, μελλοντικά, μετά την συμφωνία με την Nokia, όπου θα χρησιμοποιείται ως το βασικό λειτουργικό στα κινητά τηλέφωνα της, δείχνει να είναι ικανό να ανταγωνιστεί τα άλλα 2 μεγάλα λειτουργικά συστήματα, το Android και το iOS. Για τον προγραμματισμό σε αυτή την πλατφόρμα, οι εφαρμογές πρέπει να βασίζονται ή στο XNA, ένα σετ εργαλείων της Microsoft με περιβάλλον διαχείρισης ανάπτυξης εφαρμογών, ή σε μια συγκεκριμένη έκδοση του Silverlight που να υποστηρίζει τα Windows phone 7. Για να υπάρχει η δυνατότητα σχεδίασης και δοκιμής εφαρμογών με το Visual Studio 2010, η Microsoft προσφέρει τα Windows Phone Developer Tools ως επέκταση. Αυτό το σετ εργαλείων υποστηρίζει υπολογιστές που χρησιμοποιούν Windows Vista SP2 ή νεότερα.

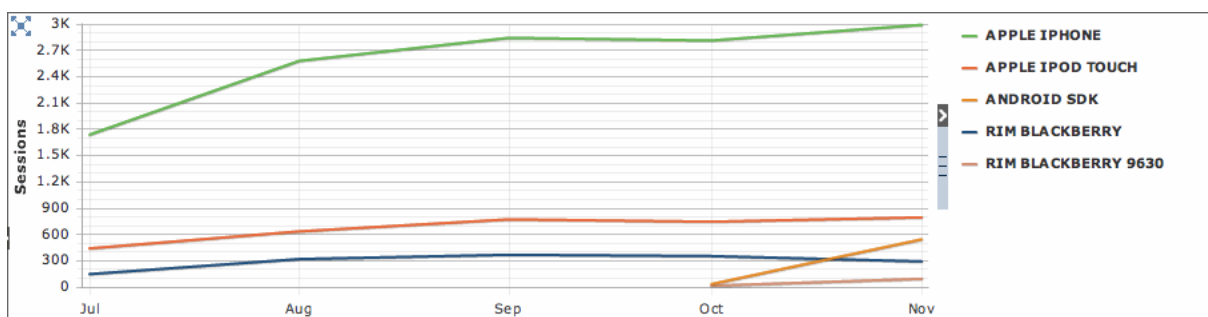
1.6 Επιτυχία πλατφόρμας Android

Οι εταιρείες- μέλη του OHA που ασχολούνται με την κατασκευή κινητών συσκευών συμβάλλουν στην σχεδίαση της πρώτης γενιάς κινητών Android. Η πρώτη συσκευή Android κυκλοφόρησε το 2008 από την HTC και σε χρονικό διάστημα 18 μηνών κυκλοφόρησαν 60 διαφορετικές συσκευές Android από 21 διαφορετικούς κατασκευαστές.

Ένας από τους λόγους επιτυχίας του Android είναι το κόστος των συσκευών του οι οποίες ανά τον κόσμο είναι φθηνότερες από αυτές των ανταγωνιστικών τηλεφώνων όπως το iPhone. Σίγουρα αυτός είναι άλλος ένας βασικός λόγος στον οποίο οφείλεται η γρήγορη εξάπλωση του Android. Εξίσου σημαντικοί λόγοι είναι η πλειονότητα των εφαρμογών που υπάρχουν στο διαδίκτυο, και μάλιστα οι περισσότερες χωρίς κόστος, και ο κώδικας ανοιχτής πηγής, που μέσα από πάρα πολλά βοηθήματα που κυκλοφορούν στο διαδίκτυο μπορεί ο καθένας να φτιάξει εφαρμογές για το κινητό του.

Στατιστικά στοιχεία που ανακοίνωσε η Google αναφέρουν ότι έχουν πραγματοποιηθεί 100 εκατ. ενεργοποιήσεις από την πρώτη ημέρα κυκλοφορίας του Android OS, γίνονται 400.000 νέες ενεργοποιήσεις κάθε ημέρα. Τέλος απαριθμούνται 36 κατασκευαστές συσκευών και 215 πάροχοι σε 112 χώρες. Σε ανακοίνωση που ακολούθησε αναφέρει ότι το Android πλέον λειτουργεί με περισσότερες από 130 εκατομμύρια συσκευές σε όλο τον κόσμο, οι οποίες αν συγκριθούν με τα 200 εκατομμύρια της Apple φαίνεται ότι η Google πλησιάζει.

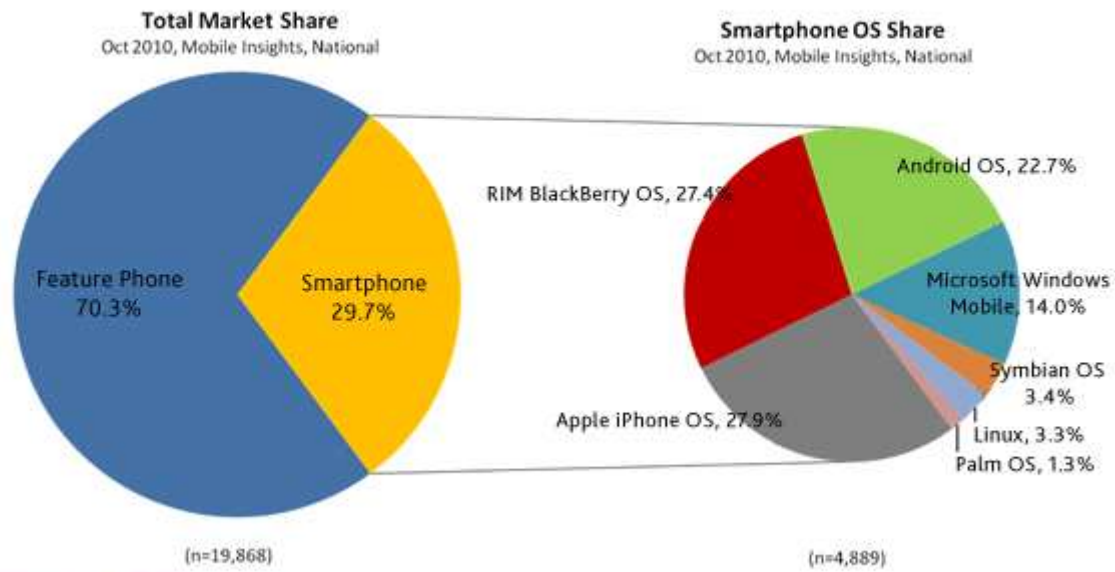
Ακολουθούν στατιστικά στοιχεία σχετικά με την επιτυχία της πλατφόρμας Android και συγκρίσεις με άλλες πλατφόρμες.



Εικόνα 1.6. 1 Νοέμβριος 2009

Total U.S. Market & Smartphone Market

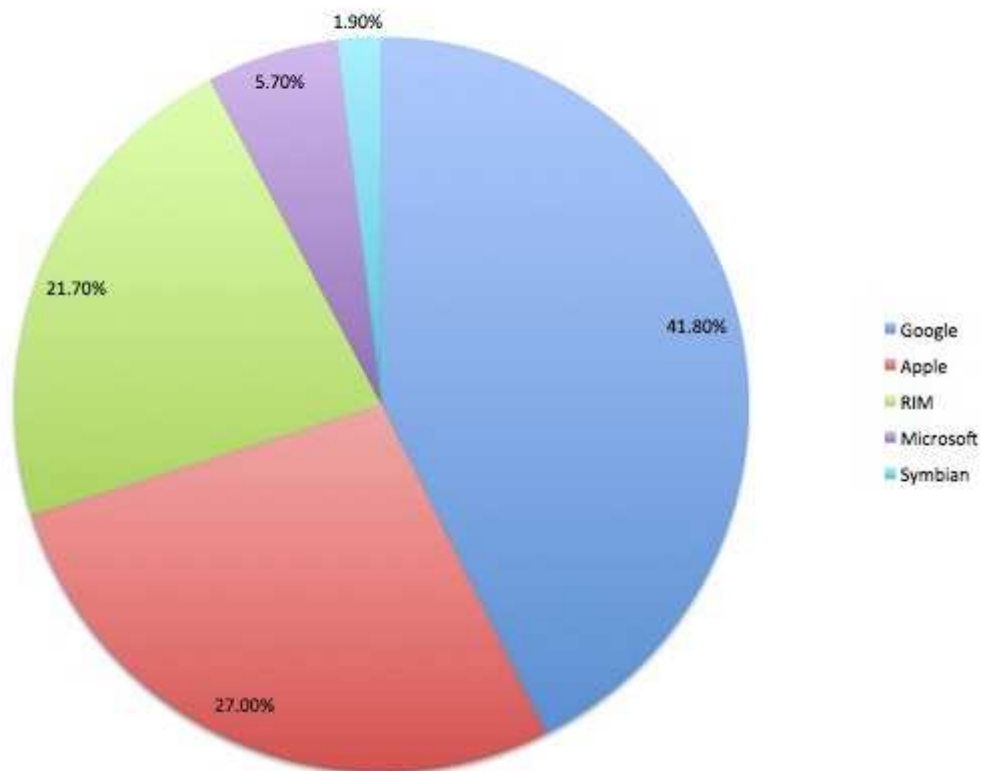
October 2010



Source: The Nielsen Company

Εικόνα 1.6. 2 Android vs iOS – Οκτώβριος 2010

Smartphone Market Share, July 2011 (source: comScore)



Εικόνα 1.6. 3 Ιούλιος 2011

Έχουν υπάρξει πάρα πολλά άρθρα που έχουν γραφτεί συζητώντας ποια πλατφόρμα είναι πιο δημοφιλής, το Android ή iOS. Οι απόψεις δίστανται όταν θα αναφερθεί σε μια παρέα σαν θέμα συζήτησης, ωστόσο η παραπάνω έρευνα (εικόνα 1.6.2) μας δείχνει ότι η διαφορά μεταξύ των δυο είναι μόλις 5,2%. Εάν

συγκρίνουμε με την έρευνα που απεικονίζεται στην εικόνα 1.6.1 βλέπουμε αισθητή μείωση της διαφοράς μεταξύ των iOS και Android, ενώ το RIM Blackberry εξακολουθεί να έχει μεγαλύτερο ποσοστό. Τέλος, βλέπουμε στα στατιστικά του Ιουλίου του 2011 (εικόνα 1.6.3) ότι η ανάπτυξη του Android είναι κάτι παραπάνω από εμφανής αφού σχεδόν καλύπτει το 50% της πίτας.

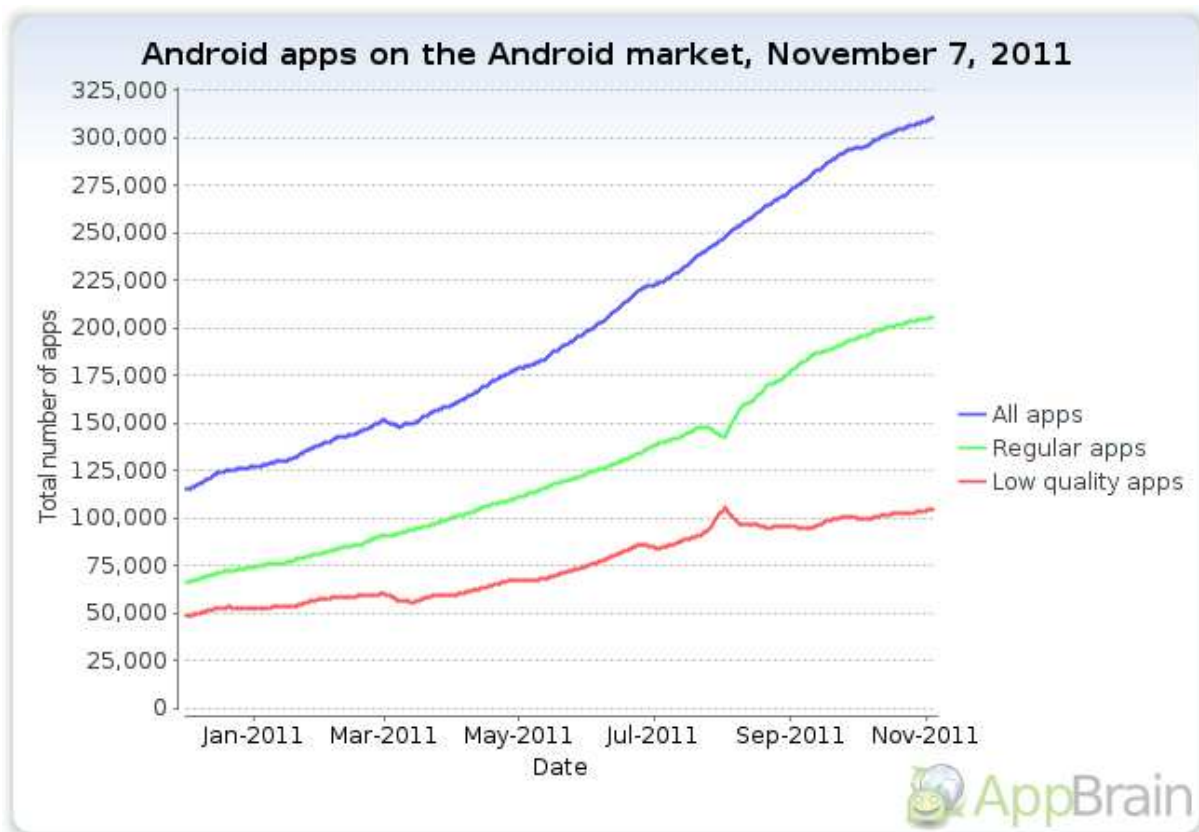
Κεφάλαιο 2ο - Εφαρμογές, Android market, Σήμερα

2.1 Εφαρμογές

Η ανάπτυξη εφαρμογών Android γνώρισε ραγδαία εξέλιξη ξεκινώντας από την Google και τα μέλη του OHA. Στον πρώτο διαγωνισμό έγιναν 1788 αιτήσεις ενώ στον δεύτερο ψήφισαν 26000 χρήστες για την ανάδειξη των 200 καλύτερων εφαρμογών. Οι εφαρμογές αυτές διατίθενται στο Android Market, όπου μέχρι τον Οκτώβριο του 2010 είχαν φτάσει τις 80000, αλλά και σε άλλες αγορές. Αυτή τη στιγμή υπάρχουν περισσότεροι από 180000 προγραμματιστές Android που δημιουργούν νέες εφαρμογές.

Όπως έχουμε ήδη αναφέρει μπορεί σχεδόν ο κάθε προγραμματιστής να αναπτύξει μια εφαρμογή και να την προωθήσει στο διαδίκτυο, έτσι ώστε να την χρησιμοποιήσουν και άλλοι χρήστες. Για την ανάπτυξη των εφαρμογών δεν υπάρχουν επιπλέον κόστη, αφού όλα τα εργαλεία που χρησιμοποιεί ο προγραμματιστής είναι δωρεάν. Η γλώσσα προγραμματισμού που χρησιμοποιείται είναι η Java και το περιβάλλον ανάπτυξης των εφαρμογών περιλαμβάνει παραδοσιακές δομές προγραμματισμού και ειδικά σχεδιασμένες δομές δεδομένων για ενσωμάτωση αντικειμένων που χρησιμοποιούνται σε εφαρμογές κινητών τηλεφώνων.

Στην πλατφόρμα Android δεν διαφοροποιούνται οι κύριες εφαρμογές του τηλεφώνου από εκείνες που έχουν δημιουργηθεί από τρίτους. Μπορούν όλες να κατασκευαστούν ώστε να έχουν την ίδια πρόσβαση στις δυνατότητες του τηλεφώνου παρέχοντας στους χρήστες ένα ευρύ φάσμα εφαρμογών και υπηρεσιών. Βασικό χαρακτηριστικό τους είναι ότι όλες οι εφαρμογές χρησιμοποιούν τις ίδιες βιβλιοθήκες. Στις συσκευές που τρέχουν την πλατφόρμα Android, οι χρήστες θα μπορούν να προσαρμόσουν πλήρως το τηλέφωνο στις ανάγκες τους. Μπορούν να αλλάξουν την αρχική οθόνη, το στυλ του dialer, ή οποιαδήποτε εφαρμογή. Μπορούν επίσης να δώσουν εντολή στα κινητά τους να χρησιμοποιούν την αγαπημένη τους εφαρμογή επισκόπησης εικόνων για να ανοίγει τις φωτογραφίες. Διαθέτουν πρωτοφανή δικαιώματα πρόσβασης στο υποκείμενο υλικό, δίνοντας το δικαίωμα στους προγραμματιστές να δημιουργούν όλο και πιο ισχυρές εφαρμογές. Τέλος δεν υπάρχουν κόστη για τη δημοσίευση μιας εφαρμογής, εκτός από την εγγραφή στο Android market, και η διανομή μπορεί να γίνει είτε μέσω του Android market, είτε μέσω του Handango.com, είτε μέσω δικούς τους μηχανισμούς διανομής και πληρωμής.



Εικόνα 2.1. 1 Στατιστικά Ιανουαρίου– Νοεμβρίου 2011

Παρατηρούμε ότι ο αριθμός των εφαρμογών μέσα σε αυτό το χρονικό διάστημα αυξήθηκε κατά 200,000 περίπου.

2.2 Barcode scanner, RFID, χάρτες Google

Τα barcode scanner, RFID και χάρτες Google είναι τεχνολογίες που σχετίζονται άμεσα με το Android και λαμβάνουν ευρεία χρήση από διάφορου τύπου εφαρμογές του.

2.2.1 Barcode scanner

Ας ξεκινήσουμε με τον ορισμό του QR code. Το QR code είναι στην πραγματικότητα ότι και τα barcodes αλλά είναι αναγνωρίσιμα από κάθε συσκευή που ενσωματώνει κάμερα. Αν αναλύσουμε ένα QR Code προκύπτει κείμενο που μπορεί να φτάσει μέχρι και 3 με 4 χιλιάδες χαρακτήρες.

Μια αρκετά χρήσιμη και έξυπνη εφαρμογή που μπορούμε να βρούμε στο Android είναι το Barcode scanner. Με τη βοήθεια του προγράμματος αυτού μπορείτε να σκανάρετε τα barcode από CD, βιβλία και άλλα προϊόντα, για να βρείτε την τιμή ή να διαβάσετε κάποια σχόλια. Το σημαντικότερο όμως που προσφέρει είναι η δυνατότητα ανάγνωσης των QR Codes, για να μεταφέρεστε σε διευθύνσεις (URL's) με ένα μόνο κλικ.

Παράλληλα το Barcode scanner σας δίνει τη δυνατότητα να κωδικοποιήσετε σε QR Codes, επαφές, διευθύνσεις, μηνύματα και άλλα ώστε να τα μεταφέρετε σε ένα άλλο κινητό, και πάλι με τη βοήθεια του scanner. Το μόνο που έχετε να κάνετε είναι να ανοίξετε το πρόγραμμα, και να κεντράρεται το QR Code ή το barcode στην οθόνη που εμφανίζετε, για να μεταφερθείτε αυτόματα στον προορισμό.

Ας δούμε μέσω εικόνων την χρήση και την λειτουργία της εφαρμογής αυτής.



Εικόνα 2.2.1. 1 Εντοπισμός QR code



Εικόνα 2.2.1. 2 Ανοίγουμε τη σχετική εφαρμογή και στρέφουμε την κάμερα στο QR code



Εικόνα 2.2.1. 3 Το κινητό αυτόματα ζητάει να συνδεθεί στη διεύθυνση που του έχουμε υποδείξει

2.2.2 RFID

Το RFID (Radio Frequency Identification), ταυτοποίηση μέσω ραδιοσυχνότητων, είναι μια πρωτοποριακή τεχνολογία, που αποτελείται από ένα σύστημα ασύρματης αναγνώρισης αντικειμένων με τεράστιες εφαρμογές.



Εικόνα 2.2.2. 1 RFID ετικέτα

Κλασσικό παράδειγμα αποτελούν τα προϊόντα στις αλυσίδες καταστημάτων. Τα προϊόντα αυτά έχουν στην συσκευασία τους μια συρμάτινη ταινία, που στην ουσία είναι ένα ηλεκτρονικό κύκλωμα το οποίο ονομάζεται RFID ετικέτα (Εικόνα 2.2.1).

Η RFID ετικέτα περιέχει τον σειριακό αριθμό του αντικειμένου και οποιαδήποτε άλλη πληροφορία είναι απαραίτητη για την αναγνώριση του. Πιο συγκεκριμένα μια RFID ετικέτα περιέχει τουλάχιστον 2 μέρη:

- Ένα ολοκληρωμένο κύκλωμα για αποθήκευση και επεξεργασία πληροφορίας, διαμόρφωση και αποδιαμόρφωση ενός σήματος ραδιοσυχνότητας και άλλες ειδικές λειτουργίες.
- Μια κεραία για λήψη και μετάδοση του σήματος.

Αντίστοιχα ένας αναγνώστης RFID είναι μια συσκευή που χρησιμοποιείται για ενεργοποίηση της RFID ετικέτας. Ο αναγνώστης διαθέτει μια κεραία που στέλνει ραδιοκύματα και η RFID ετικέτα ανταποκρίνεται αποστέλλοντας τα δεδομένα της. Τα σημαντικότερα πλεονεκτήματα που προσφέρει το RFID είναι:

- Η αναγνώριση μπορεί να γίνει από απόσταση μιας και υπάρχουν RFID ετικέτες που είναι σε θέση, παίρνοντας ενέργεια από κάποια πηγή που συνήθως είναι μπαταρία, να στείλουν τις πληροφορίες στον δέκτη,
- Δυνατότητα αποθήκευσης περισσότερων δεδομένων σε σχέση με τους γραμμικούς κώδικες,
- Μπορούν να μην είναι ορατά στο ανθρώπινο μάτι μιας και για την αναγνώριση τους δεν χρειάζεται οπτικό μέσο,
- Δυνατότητα προγραμματισμού εξ αποστάσεως,

- Επιπρόσθετες λειτουργίες π.χ. παρακολούθηση και καταγραφή της θερμοκρασία.

2.2.3 Χάρτης Google

Οι χάρτες Google, είναι μια διαδικτυακή εφαρμογή υπηρεσιών χαρτογράφησης. Η τεχνολογία αυτή παρέχεται δωρεάν για προσωπική χρήση από την Google. Ανάμεσα στις υπηρεσίες που προσφέρει, είναι η ιστοσελίδα των χαρτών Google, η Google Transit και χάρτες ενσωματωμένους σε ιστοσελίδες τρίτων μέσω των Google Maps API. Παρέχει οδικούς χάρτες, εφαρμογή δρομολόγησης για ταξίδι με τα πόδια, το αυτοκίνητο ή μέσα μαζικής μεταφοράς και εντοπισμό αστικών επιχειρήσεων για πολλές χώρες σε όλο τον κόσμο.

Η τεχνολογία των χαρτών Google χρησιμοποιεί μια στενή παραλλαγή της προβολής Mercator (ένας κυλινδρικής προβολής χάρτης του κόσμου), με αποτέλεσμα να μην μπορεί να δείξει περιοχές γύρω από τους πόλους. Ένα σχετικό προϊόν είναι το Google Earth, ένα αυτόνομο πρόγραμμα το οποίο προσφέρει περισσότερα χαρακτηριστικά ως προς την παγκόσμια προβολή, συμπεριλαμβανομένων των πολικών περιοχών.

2.3 Ασφάλεια και δικαιώματα

Το Android είναι ένα σύστημα πολλαπλών διαδικασιών (multi-process) στο οποίο κάθε εφαρμογή (και μέρη του συστήματος) τρέχουν στη δική τους διαδικασία. Η περισσότερη ασφάλεια μεταξύ των εφαρμογών και του συστήματος γίνεται σε επίπεδο διαδικασιών μέσω των διευκολύνσεων του Linux, όπως τα user και group IDs που εκχωρούνται στις εφαρμογές. Επιπρόσθετα χαρακτηριστικά ασφαλείας παρέχονται μέσω ενός μηχανισμού «Αδειών» (Permission), ο οποίος θέτει τους περιορισμούς για ειδικές λειτουργίες που μια συγκεκριμένη διαδικασία μπορεί να εκτελέσει και τα δικαιώματα που έχει καθένα URI για την επί τούτου χορήγηση πρόσβασης σε συγκεκριμένα τμήματα δεδομένων.

2.3.1 Αρχιτεκτονική ασφαλείας

Ένα κύριο σημείο στην αρχιτεκτονική της ασφαλείας του Android είναι ότι καμία εφαρμογή, από προεπιλογή, δεν έχει άδεια να κάνει οποιοσδήποτε λειτουργίες οι οποίες θα έχουν αρνητικές επιπτώσεις σε άλλες εφαρμογές, στο λειτουργικό σύστημα, ή στον χρήστη. Περιλαμβάνονται η ανάγνωση ή η εγγραφή των προσωπικών δεδομένων του χρήστη (όπως οι επαφές ή τα e-mail), η ανάγνωση ή η εγγραφή αρχείων άλλης εφαρμογής, η πρόσβαση στο δίκτυο, η διατήρηση αναμμένης της οθόνης της συσκευής, κλπ.

Η διαδικασία μιας εφαρμογής είναι ένα ασφαλές «κουτί» (sandbox). Δεν μπορεί να διαταράξει άλλες εφαρμογές, εκτός εάν δηλώσει ρητά τα δικαιώματα που χρειάζεται για πρόσθετες δυνατότητες που δεν παρέχονται από το βασικό «κουτί». Αυτά τα δικαιώματα που ζητάει μπορούν να διεκπεραιώσουν από το λειτουργικό με διάφορους τρόπους, συνήθως αυτόματα επιτρέποντας ή όχι βάση πιστοποιητικών, ή προτρέποντας τον χρήστη να αποφασίσει. Τα δικαιώματα που χρειάζονται από μία εφαρμογή δηλώνονται στατικά στην εφαρμογή, έτσι ώστε να είναι γνωστά κατά την εγκατάσταση της και δεν μπορούν να αλλάξουν μετέπειτα.

2.3.2 Υπογραφή εφαρμογής

Όλες οι Android εφαρμογές (αρχεία .apk) πρέπει να υπογράφονται με ένα πιστοποιητικό του οποίου το ιδιωτικό κλειδί κρατείται από τον προγραμματιστή τους. Το πιστοποιητικό δεν χρειάζεται να έχει υπογραφεί από μία αρχή πιστοποίησης και έτσι είναι απολύτως επιτρεπτό και σύνηθες για τις Android εφαρμογές, να χρησιμοποιούν αυτό-υπογραφόμενα πιστοποιητικά. Το πιστοποιητικό αυτό χρησιμοποιείται μόνο για την επίτευξη σχέσεων εμπιστοσύνης μεταξύ των εφαρμογών και όχι για το γενικό έλεγχο του κατά πόσο μία εφαρμογή μπορεί να εγκατασταθεί. Οι πιο σημαντικοί τρόποι που οι υπογραφές επιδρούν στην ασφάλεια είναι με το να προσδιορίζεται ποιός μπορεί να έχει πρόσβαση στις άδειες βάση πιστοποιητικού και ποιός μπορεί να μοιράζει ID χρήστη.

2.3.3 ID χρήστη και πρόσβαση αρχείων

Κάθε αρχείο Android πακέτο (.apk) που εγκαθίσταται στη συσκευή, λαμβάνει το δικό του μοναδικό Linux ID χρήστη, δημιουργείται γι' αυτό ένα «κουτί» και εμποδίζεται από το να έχει επαφή με άλλες

εφαρμογές (ή οι άλλες εφαρμογές να έχουν επαφή μ' αυτό). Αυτό το ID χρήστη του απονέμεται κατά την εγκατάσταση της εφαρμογής στη συσκευή και παραμένει σταθερό κατά τη διάρκεια της ζωής του σ' αυτή τη συσκευή.

Επειδή η επιβολή ασφάλειας συμβαίνει στο επίπεδο διαδικασίας, ο κώδικας οποιoσδήποτε δύο πακέτα δεν μπορεί να εκτελεστεί κανονικά στην ίδια διαδικασία, δεδομένου ότι πρέπει να τρέχουν ως διαφορετικοί χρήστες Linux. Μπορεί να χρησιμοποιηθεί το χαρακτηριστικό `sharedUserId` στην ετικέτα `manifest` του `AndroidManifest.xml` κάθε πακέτου, για να λάβουν έτσι το ίδιο ID χρήστη. Με τον τρόπο αυτό, για λόγους ασφαλείας τα δύο πακέτα αντιμετωπίζονται στη συνέχεια σαν να είναι η ίδια εφαρμογή, με το ίδιο ID χρήστη και τα ίδια δικαιώματα αρχείου. Αξίζει να σημειωθεί ότι για να διατηρηθεί η ασφάλεια, μόνο δυο εφαρμογές που υπογράφηκαν με το ίδιο πιστοποιητικό (και ζήτησαν το ίδιο `sharedUserId`) θα μπορέσουν να πάρουν το ίδιο ID χρήστη.

Στα δεδομένα που αποθηκεύονται από μία εφαρμογή, απονέμεται το ID χρήστη της εφαρμογής και δεν είναι προσβάσιμα από άλλα πακέτα. Κατά τη δημιουργία ενός νέου αρχείου με τις `getSharedPreferences(String, int)`, `openFileOutput(String, int)`, ή την `openOrCreateDatabase(String, int, SQLiteDatabase.CursorFactory)`, μπορούν να χρησιμοποιηθούν τα flags `MODE_WORLD_READABLE` και/ή `MODE_WORLD_WRITEABLE` για να επιτρέψουν σε οποιοδήποτε πακέτο να διαβάσει/γράψει στο αρχείο. Όταν τίθενται αυτά τα flags, το αρχείο ανήκει ακόμα στην εφαρμογή αλλά τα δικαιώματα ανάγνωσης/εγγραφής έχουν τεθεί κατάλληλα έτσι ώστε οποιαδήποτε εφαρμογή θέλει μπορεί να έχει πρόσβαση σ' αυτό.

2.3.4 Χρησιμοποιώντας τις άδειες

Μία βασική εφαρμογή Android δεν έχει άδειες-δικαιώματα που να συνδέονται μ' αυτή, που σημαίνει ότι δεν μπορεί να κάνει τίποτα που να επηρεάζει αρνητικά την εμπειρία του χρήστη ή οποιαδήποτε δεδομένα στη συσκευή. Για να γίνει χρήση των προστατευόμενων λειτουργιών της συσκευής, θα πρέπει να συμπεριληφθούν στο αρχείο `AndroidManifest.xml` μία ή περισσότερες ετικέτες `<uses-permission>` δηλώνοντας τα δικαιώματα που χρειάζεται η εφαρμογή.

Για παράδειγμα, μία εφαρμογή που χρειάζεται να παρακολουθεί τα εισερχόμενα μηνύματα SMS κα ορίζει τα εξής:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.android.app.myapplication" >
<uses-permission android:name="android.permission.RECEIVE_SMS" />
</manifest>
```

Κατά την εγκατάσταση της εφαρμογής, οι άδειες που ζητήθηκαν απ' αυτή παραχωρήθηκαν από το πρόγραμμα εγκατάστασης πακέτων, με βάση τον έλεγχο του πιστοποιητικού της εφαρμογής και/ή προτρέποντας το χρήστη να αποφασίσει. Από τη στιγμή που τρέχει η εφαρμογή δεν γίνονται έλεγχοι αλληλεπίδρασης με το χρήστη και έτσι είτε χορηγήθηκε μια συγκεκριμένη άδεια όταν εγκαταστάθηκε και μπορεί να χρησιμοποιηθεί η αντίστοιχη λειτουργία, είτε η άδεια απορρίφθηκε και οποιαδήποτε προσπάθεια να χρησιμοποιηθεί εκείνη η λειτουργία θα αποτύχει αφού δεν εγκρίθηκε από το χρήστη.

Πολλές φορές μία αποτυχία άδειας καταλήγει σε *εξαίρεση ασφαλείας* (`SecurityException`) που επιστρέφεται στην εφαρμογή. Ωστόσο, δεν υπάρχει εγγύηση ότι θα συμβεί παντού. Για παράδειγμα, η μέθοδος `sendBroadcast(Intent)` ελέγχει τα δικαιώματα όσο τα δεδομένα παραλαμβάνονται από κάθε παραλήπτη, αφού η κλήση της μεθόδου έχει επιστρέψει, έτσι ώστε να μην ληφθεί *εξαίρεση* εάν υπάρχουν αποτυχίες άδειας. Όμως, σε όλες σχεδόν τις περιπτώσεις, μία αποτυχία άδειας θα τυπωθεί στο αρχείο καταγραφής του συστήματος.

Οι άδειες που παρέχονται από το σύστημα του Android μπορούν να βρεθούν στο `Manifest.permission`. Κάθε εφαρμογή μπορεί επίσης να καθορίσει και να επιβάλλει τα δικά της δικαιώματα, έτσι ώστε να μην ζητείται μία πλήρης λίστα με όλες τις πιθανές άδειες.

Μία συγκεκριμένη άδεια μπορεί να επιβάλλεται σε διάφορα τμήματα της εφαρμογής κατά τη διάρκεια εκτέλεσης της, όπως:

- Κατά τη διάρκεια μιας κλήσης στο σύστημα, για να αποφευχθεί μία εφαρμογή από το να εκτελέσει κάποιες λειτουργίες.
- Κατά την εκκίνηση ενός *Activity*, για εμποδιστούν οι εφαρμογές από το να εκτελέσουν *Activities* άλλων εφαρμογών.
- Τόσο κατά την αποστολή όσο και κατά την λήψη *Broadcasts*, για να ελεγχθεί ποιος μπορεί να στείλει ή να λάβει *Broadcasts* σε/από ποιόν.
- Κατά την πρόσβαση και τον χειρισμό ενός *Content Provider*.
- Συνδέοντας (*Binding*) ή αρχίζοντας ένα *Service*.

2.3.5 Δήλωση και επιβολή αδειών

Για να επιβάλλετε τις δικές σας άδειες, πρέπει πρώτα να τις δηλώσετε στο αρχείο *AndroidManifest.xml* χρησιμοποιώντας μία ή περισσότερες ετικέτες `<permission>`.

Για παράδειγμα, μία εφαρμογή που θέλει να ελέγχει ποιός μπορεί να εκκινήσει κάποιο από τα *Activities* της, κα μπορούσε να δηλώσει μία άδεια για αυτή την λειτουργία, όπως την ακόλουθη:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.me.app.myapplication" >
<permission android:name="com.me.app.myapplication.permission.DEADLY_ACTIVITY"
android:label="@string/permlab_deadlyActivity"
android:description="@string/permdesc_deadlyActivity"
android:permissionGroup="android.permission-group.COST_MONEY"
android:protectionLevel="dangerous" />
</manifest>
```

Το χαρακτηριστικό `<protectionLevel>` είναι απαραίτητο, και λέει στο σύστημα πώς ο χρήστης πρέπει να ενημερώνεται για τις εφαρμογές που χρειάζονται την άδεια, ή ποιός επιτρέπεται να έχει την άδεια.

Το χαρακτηριστικό `<permissionGroup>` είναι προαιρετικό, και χρησιμοποιείται μόνο για να βοηθήσει το σύστημα να απεικονίζει τις άδειες στο χρήστη. Συχνά τίθεται είτε ως ένα τυπικό group του συστήματος (βλέπε `android.Manifest.permission_group`) είτε σε πιο σπάνιες περιπτώσεις ως ένα group καθορισμένο από τον προγραμματιστή της εφαρμογής. Είναι προτιμότερο να χρησιμοποιείται ένα υπάρχον group επειδή απλοποιεί το UI (*User Interface*) που φαίνεται στον χρήστη.

Να σημειωθεί ότι η ετικέτα και η περιγραφή για την άδεια κα πρέπει να παρέχονται. Αυτά είναι strings που μπορούν να εμφανίζονται στο χρήστη όταν βλέπει μία λίστα από δικαιώματα (`android:label`) ή πληροφορίες μίας μόνο άδειας (`android:description`). Η ετικέτα κα πρέπει να είναι σύντομη· λίγες λέξεις που περιγράφουν το βασικό κομμάτι της λειτουργίας που προστατεύει η άδεια. Η περιγραφή θα πρέπει να είναι μία-δύο προτάσεις που περιγράφουν τι θα επιτρέψει η άδεια να κάνει αυτός που την ζητάει. Κατά σύμβαση, η περιγραφή είναι δύο προτάσεις, η πρώτη περιγράφει την άδεια και η δεύτερη προειδοποιεί τον χρήστη τι «στραβά» πράγματα μπορούν να συμβούν εάν χορηγηθεί το δικαίωμα σε μία εφαρμογή.

Ένα παράδειγμα της ετικέτας και της περιγραφής για την άδεια `CALL_PHONE`:

```
<string name="permlab_callPhone">directly call phone numbers</string>
<string name="permdesc_callPhone">Allows the application to call phone numbers without your
intervention. Malicious applications may cause unexpected calls on your phone bill. Note that this does not
allow the application to call emergency numbers.</string>
```

Μπορείτε να δείτε τα δικαιώματα που είναι ορισμένα στο σύστημα με την εντολή `adb shell pm list permissions`. Ειδικότερα, η επιλογή “-s” εμφανίζει τις άδειες σε μορφή παρόμοια με εκείνη που τις βλέπει ο χρήστης:

```
$ adb shell pm list permissions -s
```

All Permissions:

Network communication: view Wi-Fi state, create Bluetooth connections, full Internet access, view network state
Your location: access extra location provider commands, fine (GPS) location, mock location sources for testing, coarse (network-based) location
Services that cost you money: send SMS messages, directly call phone numbers
...

2.3.5.1 Επιβολή αδειών στο *AndroidManifest.xml*

Υψηλού επιπέδου άδειες που να περιορίζουν την πρόσβαση σε ολόκληρα τμήματα του συστήματος ή σε εφαρμογές, μπορούν να εφαρμοστούν μέσω του *AndroidManifest.xml*. Το μόνο που απαιτείται είναι να συμπεριληφθεί ένα χαρακτηριστικό *android:permission* στο επιθυμητό τμήμα του συστήματος και να καθοριστεί η άδεια που κα χρησιμοποιηθεί για να ελέγχει την πρόσβαση σ' αυτό.

Άδειες τύπου *Activity* (εφαρμόζονται με την ετικέτα *<activity>*) περιορίζουν ποιός μπορεί να αρχίσει το σχετικό *Activity*. Η άδεια ελέγχεται κατά την κλήση των *Context.startActivity()* και *Activity.startActivityForResult()*. Εάν αυτός που τις καλεί δεν έχει την απαιτούμενη άδεια τότε επιστρέφεται μία *SecurityException*.

Άδειες τύπου *Service* (εφαρμόζονται με την ετικέτα *<service>*) περιορίζουν ποιός μπορεί να αρχίσει ή να συνδεθεί στο σχετικό *Service*. Η άδεια ελέγχεται κατά την κλήση των *Context.startService()*, *Context.stopService()* και *Context.bindService()*. Εάν αυτός που τις καλεί δεν έχει την απαιτούμενη άδεια τότε επιστρέφεται μία *SecurityException*.

Άδειες τύπου *BroadcastReceiver* (εφαρμόζονται με την ετικέτα *<receiver>*) περιορίζουν ποιός μπορεί να στείλει broadcasts στο σχετικό receiver. Η άδεια ελέγχεται μετά την επιστροφή της *Context.sendBroadcast()*, κακώς το σύστημα προσπαθεί να παραδώσει το broadcast στο σχετικό receiver. Ως αποτέλεσμα, μία αποτυχία άδειας δεν θα επιστρέψει μία εξαίρεση (exception) σ' αυτόν που την κάλεσε, απλά δεν θα παραδώσει το intent. Με τον ίδιο τρόπο, μία άδεια μπορεί να τροφοδοτήσει την *Context.registerReceiver()* για να ελέγχεται ποιός μπορεί να στείλει broadcasts σ' έναν καταχωρημένο (registered) receiver. Ένας άλλος τρόπος, είναι μία άδεια να τροφοδοτήσει την *Context.sendBroadcast()* για να περιορίζει ποιά αντικείμενα *BroadcastReceiver* επιτρέπεται να λαμβάνουν το broadcast (δείτε παρακάτω).

Άδειες τύπου *ContentProvider* (εφαρμόζονται με την ετικέτα *<provider>*) περιορίζουν ποιός μπορεί να έχει πρόσβαση στα δεδομένα ενός *ContentProvider*. (Οι *Content Providers* έχουν μία πρόσθετη δυνατότητα ασφαλείας, τις άδειες *URI*, που περιγράφονται αναλυτικότερα παρακάτω.) Σε αντίκες με τα προηγούμενα, εδώ υπάρχουν δύο ξεχωριστά χαρακτηριστικά άδειας που μπορούν να τεθούν: Το *android:readPermission* το οποίο περιορίζει ποιός μπορεί να διαβάζει από τον provider και το *android:writePermission* το οποίο περιορίζει ποιός μπορεί να γράψει σ' αυτόν. Να σημειωθεί ότι εάν ένας provider προστατεύεται τόσο με το δικαίωμα ανάγνωσης όσο και με της εγγραφής, το να θέσουμε μόνο την άδεια εγγραφής δεν σημαίνει ότι κα μπορούμε να διαβάζουμε κίολας απ' αυτόν. Οι άδειες ελέγχονται όταν ανακτάται για πρώτη φορά ένας provider (εάν δεν έχουμε άδεια, τότε επιστρέφεται μία *SecurityException*) και όσο κάνουμε λειτουργίες στον provider. Χρησιμοποιώντας την *ContentResolver.query()* απαιτείται να έχουμε θέσει το δικαίωμα ανάγνωσης ενώ για τις *ContentResolver.insert()*, *ContentResolver.update()* και *ContentResolver.delete()* το δικαίωμα εγγραφής. Σε όλες τις περιπτώσεις, αν δεν έχει τεθεί η απαιτούμενη άδεια, επιστρέφεται μία *SecurityException*.

2.3.5.2 Επιβολή αδειών κατά την αποστολή broadcast

Εκτός από την άδεια που τροφοδοτεί ποιός μπορεί να στείλει Intends σ' ένα καταχωρημένο *BroadcastReceiver* (όπως περιγράφηκε παραπάνω), μπορούμε επίσης να καθορίσουμε μία απαιτούμενη άδεια όταν στέλνουμε ένα broadcast. Καλίντας την *Context.sendBroadcast()* μ' ένα string άδειας, απαιτούμε από την εφαρμογή του receiver να έχει την κατάλληλη άδεια για να λάβει το broadcast μας.

Να σημειωθεί ότι και ο receiver και ο broadcaster μπορούν να απαιτήσουν μία άδεια. Όταν αυτό συμβαίνει, και οι δύο έλεγχοι άδειας πρέπει να είναι επιτυχείς για να παραδοθεί το Intend στον αντίστοιχο receiver.

2.3.5.3 Επιβολή άλλων αδειών

Άλλες άδειες μπορούν να επιβληθούν σε οποιαδήποτε κλίση σε ένα service. Αυτό επιτυγχάνεται με τη μέθοδο `Context.checkCallingPermission()`. Καλώντας την με ένα επιθυμητό string άδειας, αυτή θα επιστρέψει έναν ακέραιο που δηλώνει εάν χορηγήθηκε άδεια για την τρέχουσα κλίση. Σημειώστε ότι αυτή μπορεί να χρησιμοποιηθεί μόνο όταν εκτελούμε μία κλίση προερχόμενη από μία άλλη διαδικασία, συνήθως μέσω μίας διεπαφής IDL (Interface Definition Language) δημοσιευμένης από ένα service ή με κάποιον άλλο τρόπο δοθέντα σε μία άλλη διαδικασία.

Υπάρχουν πολλοί άλλοι χρήσιμοι τρόποι για τον έλεγχο των αδειών. Αν έχουμε το pid (process id) μιας άλλης διαδικασίας, μπορούμε να χρησιμοποιήσουμε τη μέθοδο `Context.checkPermission(String, int, int)` για να ελέγξουμε την άδεια για εκείνο το pid. Εάν έχουμε το όνομα του package μιας άλλης εφαρμογής, μπορούμε να χρησιμοποιήσουμε τη μέθοδο `PackageManager.checkPermission(String, String)` για να διαπιστώσουμε αν στο συγκεκριμένο πακέτο έχει χορηγηθεί μία ειδική άδεια.

2.3.6 Άδειες URI (Uniform Resource Indicator)

Το τυπικό σύστημα αδειών που περιγράφηκε ως τώρα, συχνά δεν επαρκεί όταν χρησιμοποιείται με Content Providers. Ένας content provider μπορεί να θέλει να προστατέψει τον εαυτό του με δικαιώματα ανάγνωσης και εγγραφής, ενώ οι άμεσοι του clients πρέπει επίσης να θέτουν ειδικά URIs για 'αυτούς, για να λειτουργήσουν οι άλλες εφαρμογές. Ένα χαρακτηριστικό παράδειγμα είναι τα συνημμένα σε μία εφαρμογή ηλεκτρονικού ταχυδρομείου. Η πρόσβαση στο μήνυμα θα πρέπει να προστατεύεται από άδειες, δεδομένου ότι πρόκειται για ευαίσθητα προσωπικά δεδομένα. Ωστόσο, αν ένα URI για μία επισυναπτόμενη φωτογραφία δοθεί σ' ένα πρόγραμμα προβολής εικόνων, αυτό δεν κα έχει το δικαίωμα να την ανοίξει επειδή δεν υπάρχει λόγος να διαθέτει μία άδεια για πρόσβαση σε ολόκληρο το email.

Η λύση γι' αυτό το πρόβλημα είναι οι άδειες ανά-URI: Όταν αρχίζει ένα activity ή επιστρέφεται ένα αποτέλεσμα σ' ένα activity, αυτός που καλεί μπορεί να ορίσει το `Intent.FLAG_GRANT_READ_URI_PERMISSION` και/ή το `Intent.FLAG_GRANT_WRITE_URI_PERMISSION`. Αυτό παραχωρεί στο λαμβάνον activity την άδεια για πρόσβαση στα συγκεκριμένα δεδομένα που το URI δείχνει στο Intent, ανεξάρτητα από το εάν έχει κάποια άδεια πρόσβασης στα δεδομένα του content provider που αντιστοιχεί στο Intent.

Αυτός ο μηχανισμός επιτρέπει σε ένα κοινό μοντέλο όπου η αλληλεπίδραση του χρήστη (άνοιγμα ενός συνημμένου, επιλογή μιας επαφής από την λίστα, κλπ.) οδηγεί στην επί τούτου χορήγηση αδειών. Αυτό μπορεί να αποτελέσει μία βασική διευκόλυνση για τη μείωση των αδειών που χρειάζονται οι εφαρμογές σε εκείνες που συνάζονται μόνο άμεσα με την συμπεριφορά τους.

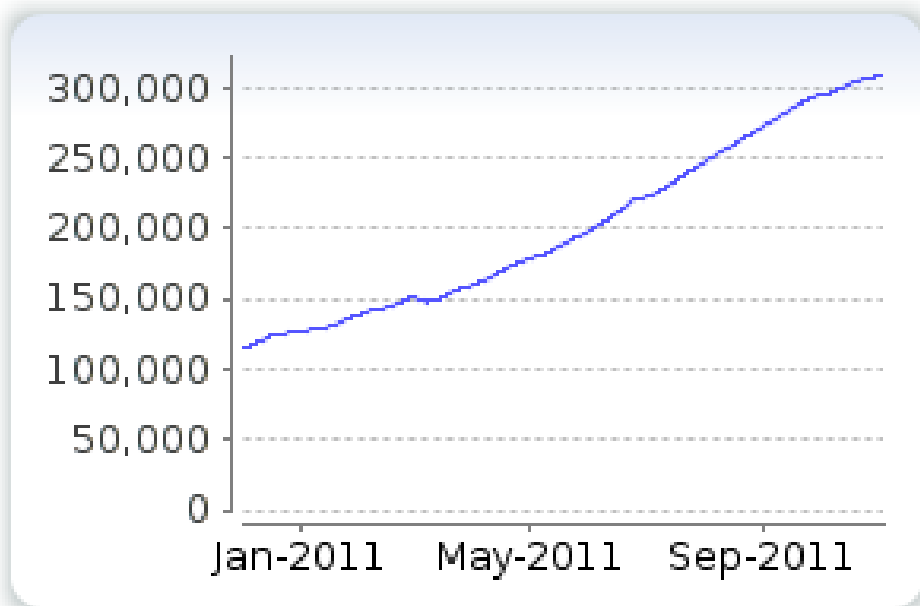
Η χορήγηση αδειών URI, ωστόσο, απαιτεί κάποια συνεργασία με τον content provider για να ελέγχει αυτά τα URIs. Συνιστάται έντονα, οι content providers να εφαρμόζουν αυτό το μηχανισμό και να δηλώνουν ότι τον υποστηρίζουν μέσω του χαρακτηριστικού `android:grantUriPermissions` ή της ετικέτας `<grant-uri-permissions>`.

2.4 Διανομή εφαρμογών στους χρήστες.

Αναφέραμε νωρίτερα τις κινήσεις που μπορεί να κάνει ένας προγραμματιστής για να διαμοιράσει την εφαρμογή του σε άλλους χρήστες και ενδεχομένως να πληρωθεί γι' αυτήν.

Αναλυτικότερα, έχουμε το Android market που είναι ένα ηλεκτρονικό κατάστημα λογισμικού το οποίο αναπτύχθηκε από την Google. Εκεί οι χρήστες μπορούν να περιηγηθούν στην αγορά, να αναζητήσουν, να κατεβάσουν οποιαδήποτε εφαρμογή θέλουν ή και να αντλήσουν πληροφορίες για τις εφαρμογές που επιθυμούν. Η εφαρμογή είναι συνήθως προεγκατεστημένη στα τηλέφωνα Android.

Οι εφαρμογές αναπτύσσονται με ραγδαίους ρυθμούς αλλά οι λήψεις (downloads) θα μπορούσαμε να πούμε ότι τρέχουν με την ταχύτητα του φωτός. Η Google ανακοίνωσε ότι έχει επιτύχει περίπου 7 δισεκατομμύρια λήψεις από το Android Market. Ωστόσο η Apple ανακοίνωσε πρόσφατα τις 15 δισεκατομμύρια λήψεις, κάτι που σημαίνει ότι το Android έχει ακόμα δρόμο να διανύσει, αν και η πλατφόρμα αυξάνεται κατά 4,4% κάθε εβδομάδα. Συνολικά, η Google ανακοίνωσε 9.030.000.000 δολάρια για το δεύτερο τρίμηνο του 2011, και καθαρά έσοδα 2,51 δισ. δολαρίων.



Εικόνα 2.4. 1 Εφαρμογές Android στο Android Market

Ημερομηνία	Εφαρμογές	Λήψεις μέχρι σήμερα
Μάρτιος, 2009	2.300	
Δεκέμβριος του 2009	20.000	
Αύγουστος 2010	80.000	1 δισ. ευρώ
Μάιο του 2011	200.000	3000000000
Ιούλιος 2011	250.000	6 δισ.
Νοέμβριος του 2011	360.000	~ 7 δισ.

Εικόνα 2.4. 2 Λήψεις Android market

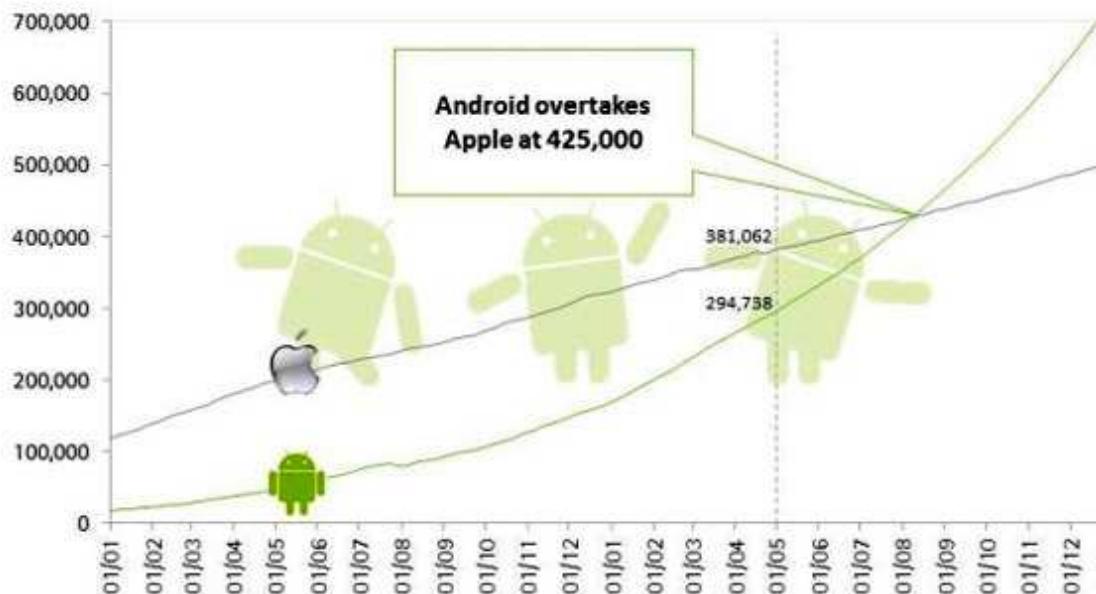
App Store	Ποσοστό δωρεάν εφαρμογές
Android Market	67%
Windows Marketplace τηλέφωνο	52%
BlackBerry App World	26%
Nokia Ovi Store	26%
Η Apple App Store	37%

Εικόνα 2.4. 3 Δωρεάν εφαρμογές στις διάφορες πλατφόρμες

Στις 21 Μαΐου 2011, η Google ανακοίνωσε ότι ο αριθμός των εφαρμογών από το Android Market αυξήθηκαν κατά 28.000 τον Απρίλιο του 2011, δηλαδή συνολικά 200.000 εφαρμογές? Αντιθέτως, τον Απρίλιο του 2011 την εισαγωγή στα χρηματιστήρια του App Store της Apple αυξήθηκαν κατά 11.000 νέες εφαρμογές, συνολικά 381.062 εφαρμογές.

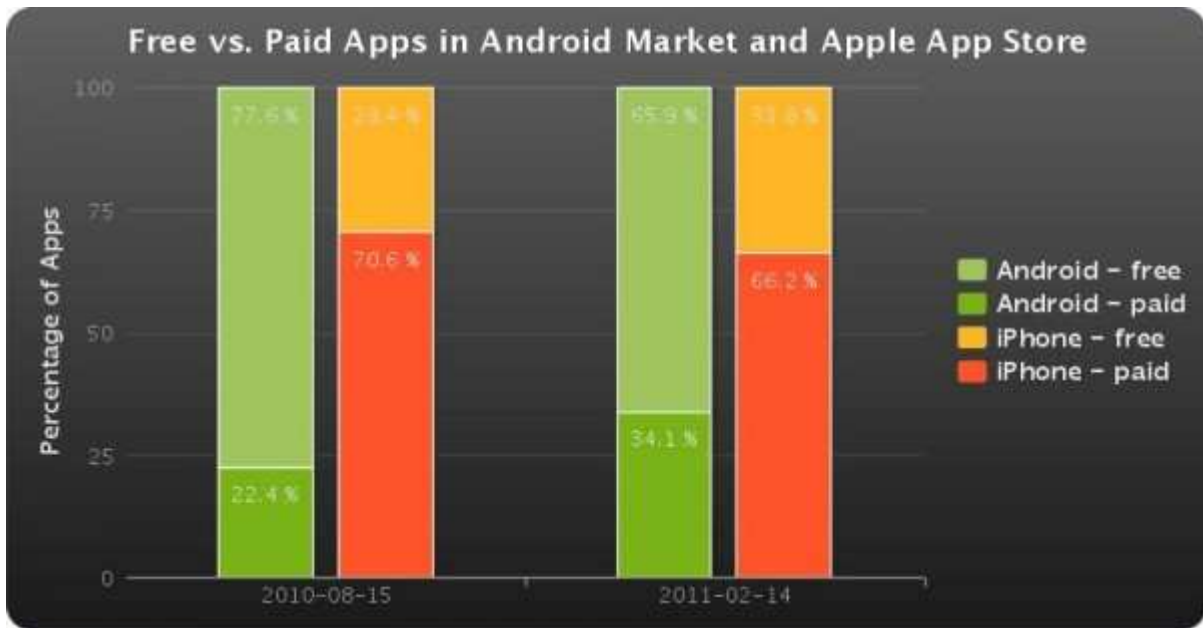
Ας δούμε και μερικές στατιστικές από το 2011 που αφορούν αποκλειστικά τα Android market και Apple App store.

Number of apps in Apple App Store and Android Market (01/2010 – 12/2011E)



research2guidance
the mobile research specialists

Εικόνα 2.4. 4 Android market vs Apple App store



Εικόνα 2.4. 5 Free vs Paid

Στην εικόνα 2.2.2 παρατηρούμε την αύξηση των εφαρμογών στην αγορά Android και την μείωση του ρυθμού αύξησης στο Apple App store. Όταν όμως εξετάζουμε τις εφαρμογές για τις οποίες ο χρήστης πληρώνει βλέπουμε ότι το Android υστερεί όσον αφορά το κέρδος από τις πληρωμένες εφαρμογές.

Το Handango.com είναι ένα άλλο ηλεκτρονικό κατάστημα με το οποίο μπορούν να ασχοληθούν οι χρήστες και οι προγραμματιστές Android. Υποστηρίζει επίσης τα λογισμικά Palm, windows mobile, Symbian και Blackberry. Ο προγραμματιστής εδώ μπορεί να χρησιμοποιήσει τα δικά του μοντέλα χρεώσεων και κοινοχρησίας εισοδήματος.

Τέλος, μπορεί ο κάθε προγραμματιστής να εφεύρει κάποιον δικό του μηχανισμό διανομής και πληρωμής εφαρμογών, δηλαδή ουσιαστικά ένα δικό του ηλεκτρονικό κατάστημα, όπου θα μπορεί να επιβάλει τους δικούς του κανόνες.

2.5 Σήμερα

Στις 18 Οκτωβρίου 2011 έγινε η επίσημη ανακοίνωση της πλατφόρμας Android 4.0. Πλέον όλα τα νέα κινούνται σε καινούργιες εφαρμογές που πρόκειται να κυκλοφορήσουν, όπως ένα πακέτο με τα περισσότερα παιχνίδια του κλασσικού ATTARI, σε καινούργια κινητά που θα έχουν το τελευταίο λογισμικό ή τελευταίες αναβαθμίσεις.



Κεφάλαιο 3ο Εργαλεία Ανάπτυξης εφαρμογών Android

3.1 Στάδια ανάπτυξης εφαρμογής

Αρχικά για την ανάπτυξη μιας εφαρμογής καταγράφουμε σε μία παράγραφο τι ακριβώς θέλουμε εμείς ή ο πελάτης να υλοποιεί η εφαρμογή μας. Στην συνέχεια ακολουθούμε τρία βασικά βήματα που θα μας βοηθήσουν στον σχεδιασμό και στην υλοποίηση.

Πρώτο βήμα είναι η καταγραφή των λειτουργιών με τίτλους και μια μικρή περιγραφή κάτι το οποίο βοηθάει στην καλύτερη κατανόηση αυτών και από τον προγραμματιστή και από τον πελάτη. Δεύτερο βήμα είναι ο εντοπισμός των απαιτούμενων οντοτήτων από τον προγραμματιστή και η δημιουργία ενός σχεσιακού διαγράμματος. Τέλος το τρίτο βήμα είναι ο σχεδιασμός της αρχιτεκτονικής, όπου τα τρία κομμάτια είναι τα model, που αφορά τη βάση δεδομένων, controller, που αφορά την γραφή κώδικα Java, και view, που αφορά κώδικα java για τις δραστηριότητες και xml.

3.2 Εργαλεία ανάπτυξης

Τα εργαλεία που χρειαζόμαστε για την διεκπεραίωση των ανωτέρω δραστηριοτήτων ποικίλουν, οπότε θα αναφέρουμε τα εργαλεία με τα οποία έχουμε δουλέψει για την εφαρμογή που αναπτύσσουμε.

3.2.1 Android SDK

Είναι ένα λογισμικό βοήθημα, θα μπορούσαμε να πούμε, ανάπτυξης που επιτρέπει στους προγραμματιστές να δημιουργήσουν εφαρμογές για την πλατφόρμα Android. Το Android SDK περιλαμβάνει δείγματα με τον πηγαίο κώδικα, εργαλεία ανάπτυξης, έναν εξομοιωτή, και τις βιβλιοθήκες που απαιτούνται για την κατασκευή Android εφαρμογές. Οι αιτήσεις είναι γραμμένες με τη Java γλώσσα προγραμματισμού και λειτουργούν με την εικονική μηχανή Dalvik.

3.2.2 Eclipse IDE

Το Eclipse είναι ένα περιβάλλον ανάπτυξης λογισμικού, το οποίο υποστηρίζει πολλές γλώσσες, περιλαμβάνει ένα ολοκληρωμένο περιβάλλον ανάπτυξης και ένα επεκτάσιμο σύστημα plug-in. Είναι γραμμένο κυρίως σε Java και μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών σε Java και, μέσω των διαφόρων plugins, και σε άλλες γλώσσες προγραμματισμού.



3.2.3 ADT Plugin για το Eclipse

Android Development Toolkit (ADT) είναι ένα plugin για το Eclipse IDE που έχει σχεδιαστεί για να μας δώσει ένα ισχυρό, ενοποιημένο περιβάλλον πάνω στο οποίο θα οικοδομηθούν οι Android εφαρμογές.

Το ADT επεκτείνει τις δυνατότητες του Eclipse για να μας επιτρέπουν να ρυθμίσουμε γρήγορα τα νέα έργα Android, να δημιουργήσουμε μια αίτηση διεπαφής, να προσθέσουμε στοιχεία με βάση το Android πλαίσιο εφαρμογής, να ελέγξουμε τις εφαρμογές μας για λάθη χρησιμοποιώντας τα εργαλεία του Android SDK, και ακόμη και να εξάγουμε τις εφαρμογές μας, με υπογραφή ή χωρίς, σε μορφή .apk αρχείων για να μπορεί να γίνει η διανομή της εφαρμογής μας.

Η ανάπτυξη στο Eclipse με ADT συνιστάται και είναι ο γρηγορότερος τρόπος για να ξεκινήσουμε. Προσφέρει καθοδηγημένη ρύθμιση του έργου, καθώς και εργαλεία ενοποίησης, προσαρμοσμένα

προγράμματα επεξεργασίας XML, και τον εντοπισμό σφαλμάτων. Το ADT δίνει μια απίστευτη ώθηση για την ανάπτυξη Android εφαρμογών.

3.2.4 Εικονική μηχανή Android (AVD)

Μια εικονική συσκευή Android (AVD) είναι ένας διαμορφωμένος εξομοιωτής που επιτρέπει να μοντελοποιήσουμε μια πραγματική συσκευή καθορίζοντας τις επιλογές του υλικού και του λογισμικού.

Ο ευκολότερος τρόπος να δημιουργήσετε ένα AVD είναι μέσω του AVD manager που παρέχει το Eclipse IDE. Μπορούμε επίσης να εκκινήσουμε τον AVD Manager από την γραμμή εντολών με την κλήση του εργαλείου Android μέσα από τον κατάλογο εργαλείων του Android SDK.

3.2.5 Ανάπτυξη βάσης δεδομένων

Η ανάπτυξη της βάσης δεδομένων που χρειαζόμαστε για την εφαρμογή μας μπορούμε να την κατασκευάσουμε είτε μέσω του Microsoft Visio, είτε με την Oracle, είτε με την Access, είτε τέλος να τη δημιουργήσουμε μέσω της εφαρμογής μας.

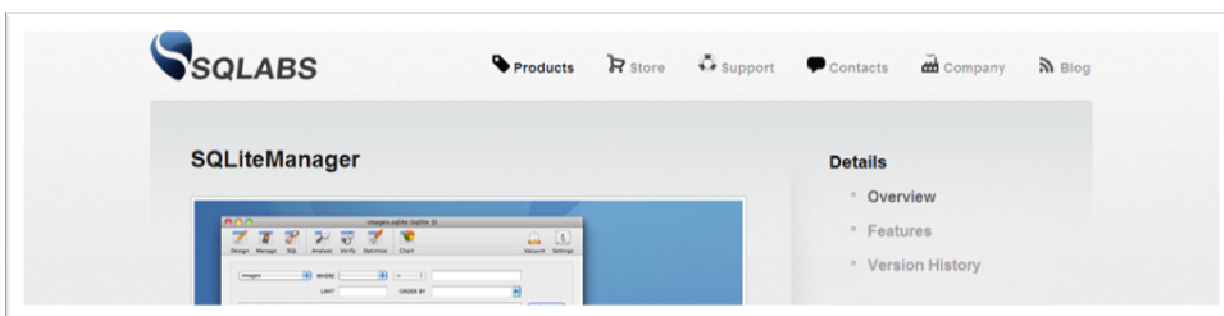
3.2.5.1 Οι βάσεις SQLite

Για την αποθήκευση των δεδομένων της συγκεκριμένης εφαρμογής που αναπτύσσουμε, θα δημιουργήσουμε μια βάση δεδομένων SQLite μέσα από την εφαρμογή μας.

Στην περίπτωση που η εφαρμογή χρειάζεται έναν πιο ανθεκτικό μηχανισμό αποθήκευσης δεδομένων, το σύστημα Android παρέχει υποστήριξη σε σχεσιακές βάσεις δεδομένων, που αφορούν στη συγκεκριμένη εφαρμογή χρησιμοποιώντας SQLite. Οι βάσεις δεδομένων SQLite είναι ελαφριές και βασίζονται σε αρχεία, κάτι που τις καθιστά ιδανικές για συσκευές με ενσωματωμένους επεξεργαστές.

Αυτές οι βάσεις δεδομένων και τα δεδομένα που περιέχουν είναι ιδιωτικά στην εφαρμογή. Για να μοιραστούν τα δεδομένα μεταξύ των εφαρμογών πρέπει να εκταθούν τα δεδομένα που θέλουμε να μοιραστούν μετατρέποντας την εφαρμογή σε πάροχο υλικού. Το Android SDK περιέχει αρκετές χρήσιμες κλάσεις διαχείρισης βάσεων δεδομένων SQLite, πολλές εκ των οποίων βρίσκονται στο πακέτο android.database.sqlite. Εκεί υπάρχουν βοηθητικές κλάσεις για την διαχείριση των διαδικασιών δημιουργίας και ορισμού της έκδοσης βάσεων δεδομένων, διαχείρισης βάσεων δεδομένων, αλλά και χρήσιμες κλάσεις κατασκευής ερωτημάτων που βοηθούν στη διαμόρφωση προτάσεων και ερωτημάτων SQL.

Αναλυτικότερα τον κώδικα για τη δημιουργία της βάσης δεδομένων, την δημιουργία και την ενημέρωση εγγραφών, θα τον δούμε στο κεφάλαιο 5, όπου και αναλύουμε το πρακτικό κομμάτι της εφαρμογής μας.



3.3 Εγκατάσταση εργαλείων ανάπτυξης

3.3.1 Microsoft Visio

Το Microsoft Visio δεν είναι ελεύθερο λογισμικό, άρα μπορεί κανείς να το χρησιμοποιεί ελεύθερα μόνο για 30 ημέρες που διαρκεί το trial και μπορεί να το βρει στην διεύθυνση <http://office.microsoft.com/en-us/visio/>.

3.3.2 SQLite Manager

Είναι πρόσθετη λειτουργία για τον firefox ή τον chrome, είναι ελεύθερο λογισμικό και μπορεί κανείς να το κατεβάσει από την διεύθυνση <http://www.sqlabs.net/sqlitemanager.php>.

3.3.3 Eclipse IDE

Το Eclipse IDE είναι και αυτό ελεύθερο λογισμικό και το κατεβάζεις από το <http://www.eclipse.org/downloads/>. Πρέπει επίσης να γίνει εγκατάσταση του SDK <http://developer.android.com/sdk/index.html> και του πρόσθετου λογισμικού για το Eclipse IDE.

Η εγκατάσταση όλων των προγραμμάτων γίνονται σχεδόν αυτόματα, χωρίς ιδιαιτερότητες, σε αντίθεση με αυτήν του πρόσθετου λογισμικού για το Eclipse IDE που θα την παρουσιάσουμε αναλυτικότερα. (Τα βήματα εξαρτώνται από την έκδοση.)

- Ανοίγουμε το Eclipse IDE και επιλέγουμε Help-> Install new software
- Στο πεδίο Work with πληκτρολογούμε τη διεύθυνση <https://dl-ssl.google.com/android/eclipse/> και κάνουμε κλικ στην επιλογή Add
- Αυτόματα εμφανίζεται ένα checkbox με ετικέτα Developer tools, το οποίο και επιλέγουμε. Πατάμε Next.
- Στην συνέχεια πάλι Next και αφού συμφωνήσουμε με την άδεια χρήσης πατάμε Finish.
- Όταν γίνει η εγκατάσταση κάνουμε επανεκκίνηση στο Eclipse.

Για να ολοκληρώσουμε με το Eclipse και να ξεκινήσουμε την ανάπτυξη της πρώτης εφαρμογής θα πρέπει να δημιουργήσουμε μια εικονική συσκευή Android. Τα βήματα έχουν ως εξής:

- Πατάμε στο μενού Window
- Επιλέγουμε το AVD Manager
- Δημιουργούμε μια εικονική συσκευή Android

Πλέον είμαστε έτοιμοι για τη δημιουργία της πρώτης εφαρμογής μας “Hello Android”.

Κεφάλαιο 4ο - Η Εφαρμογή Hello Android

Ανοίγοντας το Eclipse IDE επιλέγουμε να ξεκινήσουμε ένα νέο Android Project με όνομα helloAndroid, όνομα πακέτου com.example.helloandroid και κλάση HelloAndroid.java.

Αρχικά αν τρέξω το project θα δω την εικονική συσκευή μου, με τις εφαρμογές που είναι εγκατεστημένες ήδη, αλλά και το project. Αναλυτικά μόλις γίνει η δημιουργία του καινούργιου project στην κλάση HelloAndroid βλέπω τον εξής κώδικα:

```
package com.example.helloandroid;
```

```
import android.app.Activity;  
import android.os.Bundle;
```

```
public class HelloAndroid extends Activity {  
    /** Called when the activity is first created. */  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
    }  
}
```

R.java

```
package com.example.helloAndroid;
```

```
public final class R {  
    public static final class attr {  
    }  
    public static final class drawable {  
        public static final int ic_launcher=0x7f020000;  
    }  
    public static final class layout {  
        public static final int main=0x7f030000;  
    }  
    public static final class string {  
        public static final int app_name=0x7f040001;  
        public static final int hello=0x7f040000;  
    }  
}
```

Main.xml

```
<?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="vertical" >  
  
    <TextView  
        android:layout_width="fill_parent"  
        android:layout_height="wrap_content"  
        android:text="@string/hello" />  
  
</LinearLayout>
```

String.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```

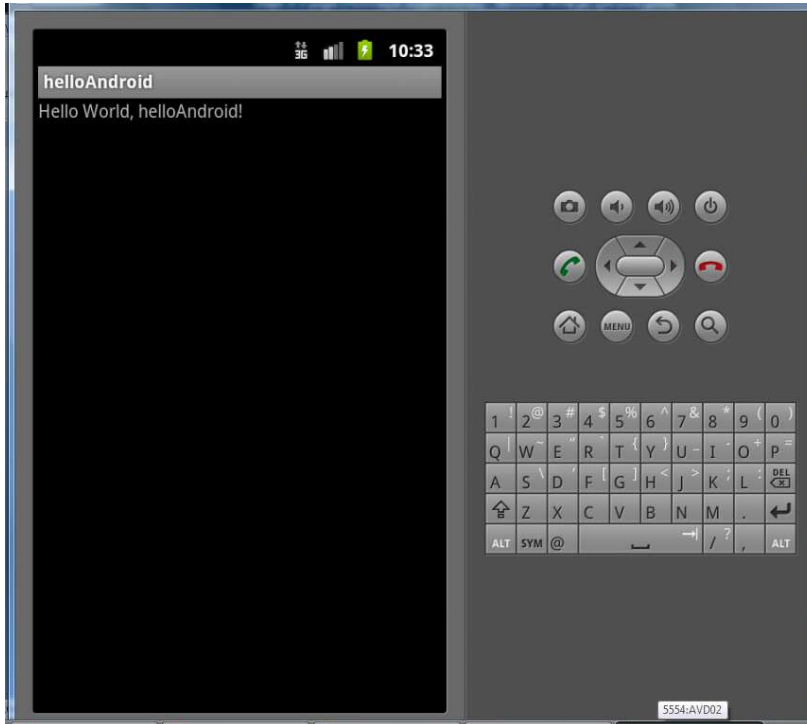
<resources>

    <string name="hello">Hello World, helloAndroid!</string>
    <string name="app_name">helloAndroid</string>

</resources>

```

Ο κώδικας που αυτόματα δημιουργείται μας εμφανίζει στην οθόνη αυτό το οποίο θέλουμε. Ουσιαστικά εμφανίζει ότι υπάρχει στο string.xml.



Εικόνα 1 Hello Android

Ένας άλλος τρόπος είναι να γράψουμε το εξ' ολοκλήρου κώδικα στην κλάση HelloAndroid. Ανοίγω την κλάση που έχω και γράφω τον εξής κώδικα.

```

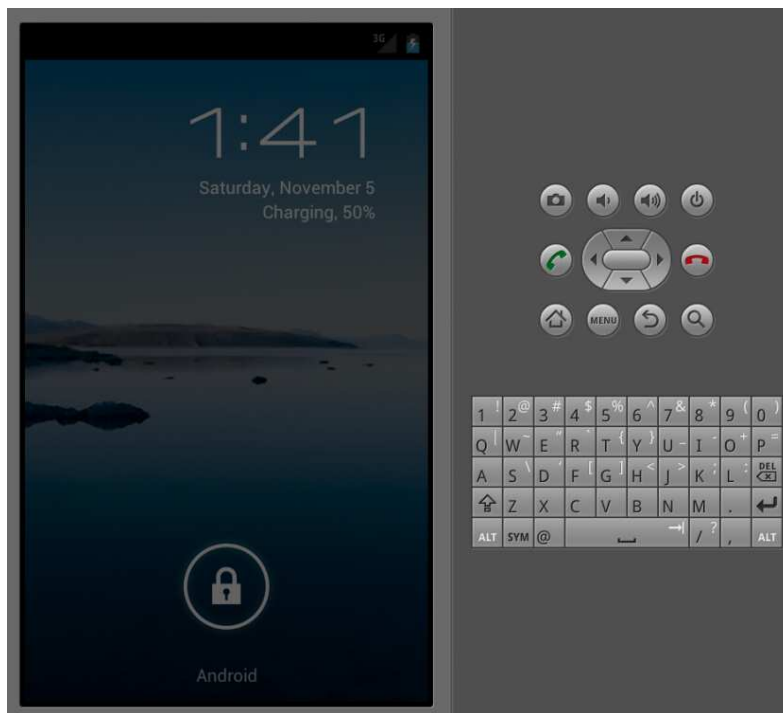
package com.example.helloandroid;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

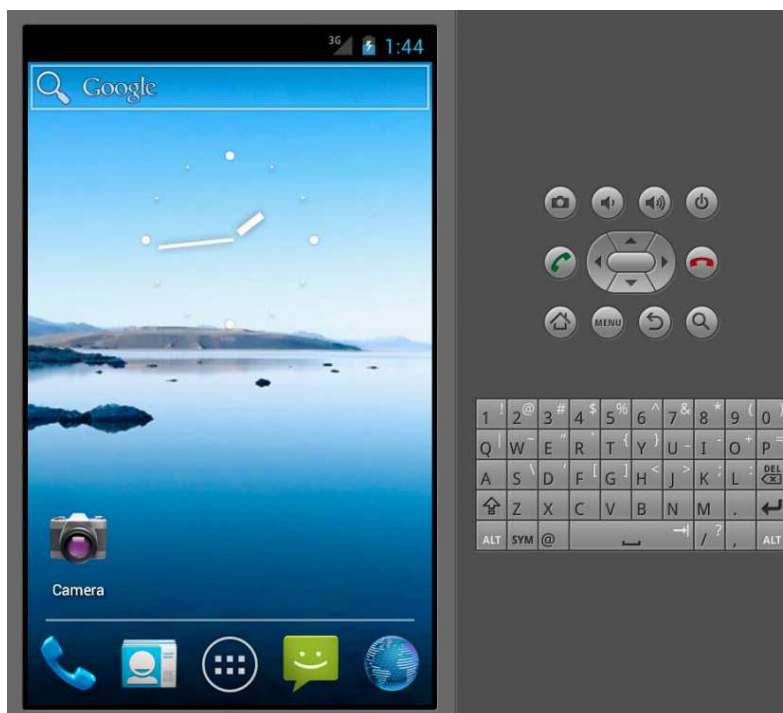
public class HelloAndroid extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TextView tv = new TextView(this);
        tv.setText("Hello, Android");
        setContentView(tv);
    }
}

```

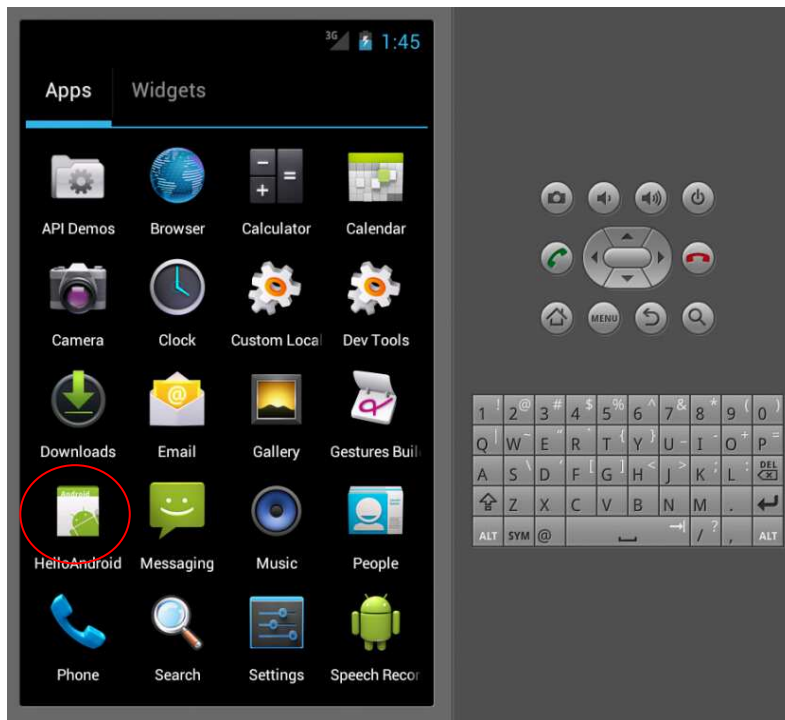

Τρέχω το πρόγραμμά μου. Μόλις ανοίξει η οθόνη ανοίγει κατευθείαν η εφαρμογή μου. Ας δούμε και έναν άλλον τρόπο που μπορούμε να ανοίξουμε την εφαρμογή.



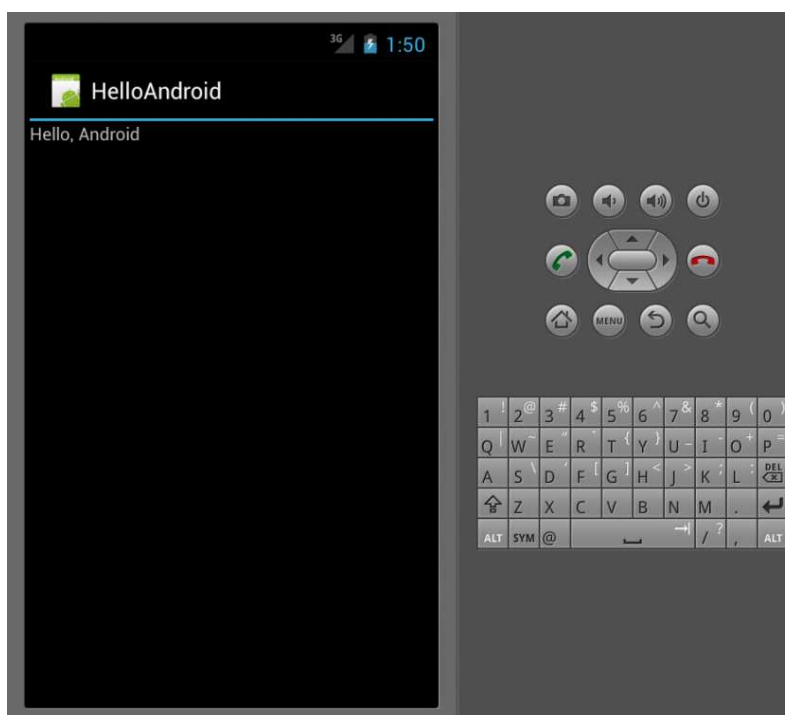
Εικόνα 2 Αρχική εικόνα εικονικής συσκευής



Εικόνα 3 Αφού ξεκλειδώσουμε τη συσκευή βρισκόμαστε στην κεντρική οθόνη.



Εικόνα 4 Πατώντας το κουμπί που μας πηγαίνει στο κεντρικό μενού βλέπουμε τις εφαρμογές που είναι εγκατεστημένες στην συσκευή μας. Παρατηρούμε την εφαρμογή που μόλις δημιουργήσαμε Hello Android.



Εικόνα 5 Πατώντας το εικονίδιο που αντιστοιχεί στην εφαρμογή μας βλέπουμε τα αποτελέσματα του κώδικα που γράψαμε.

Ουσιαστικά γράφοντας τις εξής γραμμές κώδικα

```
TextView tv = new TextView(this);
tv.setText("Hello, Android");
setContentView(tv);
```

δημιουργούμε ένα αντικείμενο τύπου TextView με όνομα tv, του δίνω την τιμή Hello, Android και τέλος με την εντολή setContentView τυπώνω στην οθόνη την τιμή του ορίσματος tv.

5.1 Θεωρητικό μέρος εφαρμογής

5.1.1 Περιγραφή εφαρμογής

Κύριος σκοπός της εφαρμογής που θα αναπτύξουμε είναι η προσθήκη πληροφοριών σε μια φωτογραφία την οποία ο χρήστης είτε θα τη διαλέγει μέσα από την βιβλιοθήκη των φωτογραφιών που υπάρχουν ήδη στην συσκευή του, είτε θα κάνει τη λήψη της εκείνη τη στιγμή. Οι πληροφορίες αυτές αφορούν την προσθήκη μιας οποιασδήποτε περιγραφής σχετικά με τη φωτογραφία, όπως τοποθεσία, ημερομηνία, εποχή και την επισήμανση των προσώπων που υπάρχουν στην φωτογραφία, τα οποία και θα επιλέγει από τις επαφές που ήδη υπάρχουν στην κινητή συσκευή του. Στην συνέχεια μέσα από την εφαρμογή αυτή δίνεται η δυνατότητα στον χρήστη να δει την κάθε φωτογραφία και την πληροφορία που διαθέτει.

5.1.2 Καταγραφή λειτουργιών εφαρμογής

Η πρώτη λειτουργία της εφαρμογής μας αφορά την επιλογή του χρήστη για το αν θέλει να κάνει λήψη μιας καινούργιας φωτογραφίας ή αν θέλει να επεξεργαστεί μια που ήδη υπάρχει στην συσκευή του. Η οθόνη περιέχει τρία κουμπιά που το καθένα αντιστοιχεί σε καθεμία από τις παραπάνω λειτουργίες. Στην πρώτη περίπτωση ο χρήστης πατάει το κουμπί «λήψη νέας φωτογραφίας» και ανοίγει η φωτογραφική μηχανή της συσκευής, πατάει το κουμπί λήψης της φωτογραφίας και αυτή αποθηκεύεται αυτόματα στην συσκευή. Επιστρέφει στην αρχική οθόνη και πατάει το κουμπί «επιλογή φωτογραφίας». Στην περίπτωση που ο χρήστης δεν επιθυμεί τη λήψη νέας φωτογραφίας περνάει κατευθείαν στην δεύτερη επιλογή. Η τρίτη επιλογή αφορά την παρουσίαση των φωτογραφιών που υπάρχουν στην βιβλιοθήκη της συσκευής και την πληροφορία, εάν αυτή υπάρχει.

Μόλις ο χρήστης επιλέξει την φόρτωση των φωτογραφιών η εφαρμογή ανοίγει μια καινούργια οθόνη που παρουσιάζει όλες τις φωτογραφίες που υπάρχουν στην συσκευή και δίνει τη δυνατότητα στον χρήστη να επιλέξει όποια φωτογραφία επιθυμεί. Ωστόσο υπάρχει η δυνατότητα να επιστρέψει στην αρχική οθόνη με το κουμπί back που υπάρχει στην συσκευή.

Η λειτουργία που ακολουθεί είναι αυτή της επιλογής του χρήστη τι είδους πληροφορία θέλει να προσθέσει στην φωτογραφία. Όταν λοιπόν επιλέξει τη φωτογραφία αυτή ανοίγει σε μια καινούργια οθόνη και πατώντας το πλήκτρο του μενού, το οποίο υπάρχει στην συσκευή, ή πατώντας τη φωτογραφία εμφανίζεται ένα μενού τριών επιλογών. Οι επιλογές είναι «Προσθήκη περιγραφής», «Επισήμανση προσώπου» και «Επιλογή νέας φωτογραφίας». Στην τελευταία περίπτωση η εφαρμογή γυρνάει ουσιαστικά ένα βήμα πίσω.

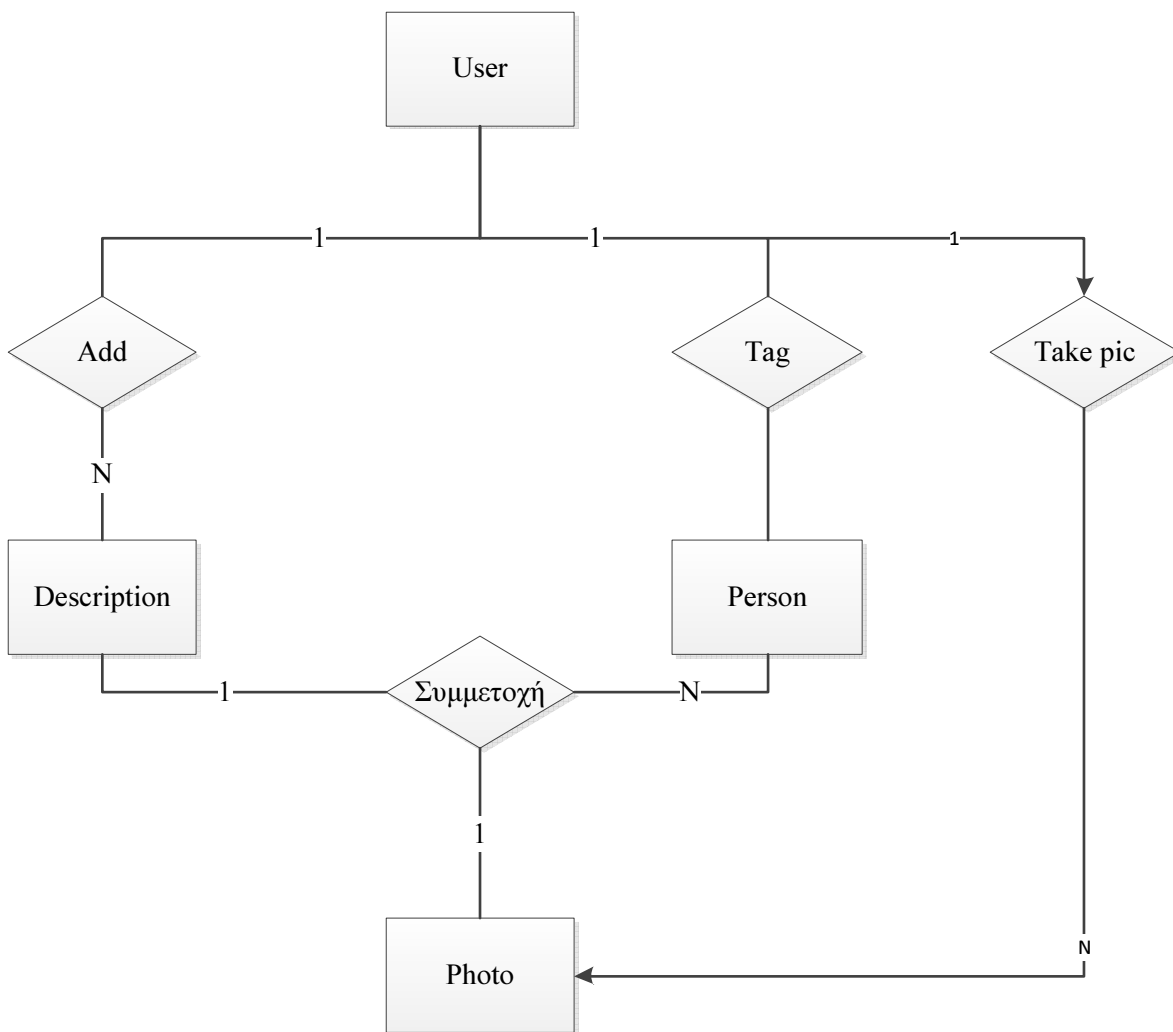
Στην περίπτωση «Προσθήκη περιγραφής» ανοίγει ένας διάλογος που περιέχει ένα κουτί, στο οποίο μπορούμε να γράψουμε την περιγραφή μας, ένα κουμπί «OK», το οποίο αποθηκεύει την περιγραφή και ένα κουμπί «Cancel», το οποίο ακυρώνει την ενέργεια μας και επιστρέφει στην φωτογραφία.

Στην περίπτωση της επιλογής «Επισήμανση προσώπου» η εφαρμογή ανοίγει αυτόματα το κατάλογο των επαφών μας από όπου μπορούμε να επιλέξουμε κάποια από τις επαφές μας. Με το που θα πατήσουμε κάποια επαφή αυτόματα αποθηκεύεται η επισήμανση και επιστρέφουμε στην οθόνη που απεικονίζεται η φωτογραφία. Στην περίπτωση που θέλουμε να επισημάνουμε περισσότερες επαφές απλά πρέπει να επαναλάβουμε το βήμα, ενώ στην περίπτωση που δεν θέλουμε να κάνουμε κάποια επισήμανση χρησιμοποιούμε το πλήκτρο back της συσκευής.

Τέλος, επιστρέφοντας στο αρχικό μενού της εφαρμογής η τρίτη επιλογή αφορά την παρουσίαση των εικόνων και των δεδομένων που συνδέονται με αυτές.

5.1.3 Οντότητες και Σχεσιακό διάγραμμα

Οι οντότητες που εντοπίζουμε στην συγκεκριμένη εφαρμογή είναι ο χρήστης, η περιγραφή, το πρόσωπο και η φωτογραφία. Το σχεσιακό διάγραμμα είναι το ακόλουθο.



Εικόνα 6 Σχεσιακό διάγραμμα

5.1.4 Σχεδιασμός αρχιτεκτονικής

Model	Controller	View
SQLite	Java Code	Java Code for activity & xml
(Σχεσιακό σχήμα DDL) Firefox plugin		

5.2 Πρακτικό μέρος εφαρμογής

5.2.1 Γραφικό περιβάλλον εφαρμογής

Αρχικά φτιάχνουμε τις οθόνες και το μενού που θα χρειαστούμε. Τα αρχεία που αφορούν την εμφάνιση της εφαρμογής μας είναι τύπου xml και τα τοποθετούμε στον κατάλογο res, όπου υπάρχουν όλοι οι πόροι της εφαρμογής μας, στον φάκελο layout. Οπότε από στον φάκελο layout αλλάζουμε το αρχείο main.xml και προσθέτουμε όσα αρχεία .xml χρειαζόμαστε. Ουσιαστικά τα αρχεία αυτά είναι οι οθόνες της εφαρμογής μας. Μπορούμε να φτιάξουμε την οθόνη είτε στο γραφικό περιβάλλον (graphical layout) του xml τοποθετώντας (drag and drop) τα στοιχεία (widgets, text fields, layouts και άλλα) που θα χρειαστούμε και στην συνέχεια μπορούμε να αλλάξουμε τις ιδιότητες τους από το .xml, είτε γράφοντας κατευθείαν κώδικα στο .xml.

Στο αρχείο `main.xml` παρουσιάζεται η πρώτη οθόνη που βλέπει ο χρήστης. Καλείται από την `Selection.java` και περιέχει τρία κουμπιά.

Στην συνέχεια φτιάχνω την οθόνη με όνομα `screen2.xml` η οποία αφορά την κλάση `PhotoViewer.java` και περιέχει ένα `TextView`, για τον τίτλο που θέλω να βάλω, και ένα `ImageView`, για την παρουσίαση της φωτογραφίας που έχει επιλέξει ο χρήστης.

Για τον διάλογο που ανοίγουμε πρέπει να φτιάξουμε ένα `.xml` αρχείο, με όνομα `add_description.xml`, που περιέχει ένα `TextView` και ένα πεδίο `EditText`. Λόγω του ότι είναι διάλογος (`dialog`) δεν χρειάζεται να προσθέσουμε τα κουμπιά αποθήκευσης (`OK`) και ακύρωσης (`Cancel`).

Η τελευταία οθόνη που πρέπει να δημιουργήσουμε είναι αυτή που θα βλέπει ο χρήστης όταν θα ανοίγει την `gallery` της εφαρμογής όπου θα παρουσιάζονται οι φωτογραφίες με τα δεδομένα που περιέχει η κάθε μία. Το αρχείο αυτό είναι το `tagview.xml`.

Στην περιγραφή της εφαρμογής μας αναφέραμε και την ύπαρξη ενός μενού. Το μενού περιγράφεται από το αρχείο `add_info_menu.xml` και βρίσκεται στον φάκελο `res -> menu`.

5.2.2 AndroidManifest.xml

Κάθε εφαρμογή πρέπει να έχει το συγκεκριμένο αρχείο με αυτό ακριβώς το όνομα στο αρχικό `root` του καταλόγου. Στο αρχείο αυτό παρουσιάζονται οι απαραίτητες πληροφορίες της εφαρμογής οι οποίες είναι απαραίτητες στο σύστημα `Android` για να μπορέσει να εκτελέσει τον οποιοδήποτε κώδικα της εφαρμογής. Μεταξύ άλλων οι λειτουργίες που πραγματοποιεί το αρχείο αυτό είναι:

- Ονομάζει το `JAVA` πακέτο της εφαρμογής. Το όνομα του πακέτου λειτουργεί ως μοναδικό αναγνωριστικό `ID` για την εφαρμογή.
- Περιγράφει τις συνιστώσες της εφαρμογής, όπως δραστηριότητες, υπηρεσίες και άλλα.
- Δηλώνει ποια δικαιώματα πρέπει να έχει η εφαρμογή προκειμένου να είναι δυνατή η πρόσβαση σε προστατευμένες περιοχές του `API` και να μπορεί να αλληλεπιδρά με άλλες εφαρμογές.
- Δηλώνει δικαιώματα που οι άλλοι οφείλουν να έχουν προκειμένου να μπορέσουν να αλληλεπιδράσουν με την εφαρμογή.
- Δηλώνει το ελάχιστο επίπεδο του `Android API` που απαιτεί η εφαρμογή.

Άρα πριν γράψουμε περαιτέρω κώδικα πάμε στο `AndroidManifest.xml` για να συμπληρώσουμε τις άδειες (`permissions`), τα δικαιώματα και τις κλάσεις που θα χρειαστούμε και θα χρησιμοποιήσουμε στην εφαρμογή μας. Η κλάση η οποία θα χρησιμοποιηθεί πρώτη από την εφαρμογή μας είναι η `Selection.java` και είναι η πρώτη κλάση που δηλώνεται στο `AndroidManifest.xml`.

5.2.3 Το αρχείο strings.xml

Για την γραφική διεπαφή θα χρειαστούμε κάποιες μεταβλητές οι οποίες θα περιέχουν τις συμβολοσειρές για τα ονόματα στα πλήκτρα, τα χρώματα και ότι άλλο χρειαστούμε. Η δήλωσή των μεταβλητών αυτών γίνεται στο αρχείο `res-> values-> strings.xml` και οι τύποι μεταβλητών είναι οι `string`, `color`, `drawable`, `dimension`, `integer`, `array`.

5.2.4 Οι κλάσεις της εφαρμογής.

Τώρα ας δούμε αναλυτικά τις κλάσεις και τον κώδικα που χρησιμοποιούμε. Αρχικά οι κλάσεις που χρειαζόμαστε είναι οι εξής `Selection.java`, `PhotoViewer.java`, `DatabaseHelper.java` και `TagViewer.java`.

5.2.4.1 Η κλάση Selection.java

Σαν πρώτο βήμα είναι η δημιουργία δυο κουμπιών.

```
Button btnCamera = (Button) findViewById(R.id.btnCamera);  
Button btnGallery = (Button) findViewById(R.id.btnGallery);
```

Πατώντας το κουμπί `btnCamera` ανοίγει η κάμερα της κινητής συσκευής και μόλις γίνει η λήψη της φωτογραφίας αυτή αποθηκεύεται στην `Photo Gallery`.

```
//define the file-name to save photo taken by Camera activity  
String fileName = "new-photo-name.jpg";
```

```

//create parameters for Intent with filename
ContentValues values = new ContentValues();
values.put(MediaStore.Images.Media.TITLE, fileName);
values.put(MediaStore.Images.Media.DESCRPTION, "Image capture by camera");
//imageUri is the current activity attribute, define and save it for later usage (also
in onSaveInstanceState)
Uri imageUri = getResolver().insert(
    MediaStore.Images.Media.EXTERNAL_CONTENT_URI, values);
//create new Intent
Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
intent.putExtra(MediaStore.EXTRA_OUTPUT, imageUri);
startActivityForResult(intent, CAPTURE_IMAGE_ACTIVITY_REQUEST_CODE);

```

Και στο onActivityResult

```

sendBroadcast(new Intent(Intent.ACTION_MEDIA_MOUNTED,
Uri.parse("file://" + Environment.getExternalStorageDirectory())));

```

Πατώντας το κουμπί btnGallery ανοίγει καινούργια οθόνη όπου απεικονίζονται οι φωτογραφίες της συσκευής. Αυτό επιτυγχάνεται ξεκινώντας τη λειτουργία της κλάσης PhotoViewer.java.

```

Intent i = new Intent(Selection.this, PhotoViewer.class);
startActivity(i);

```

5.2.4.2 Η κλάση PhotoViewer.java

Η πρώτη λειτουργία που ξεκινάει με το άνοιγμα της κλάσης είναι το άνοιγμα του συνόλου των φωτογραφιών της photo gallery της συσκευής. Μόλις επιλέξουμε την φωτογραφία που θέλουμε ανοίγει καινούργια οθόνη που απεικονίζει την επιλεγμένη φωτογραφία.

```

Intent photoPickerIntent = new Intent(Intent.ACTION_GET_CONTENT);
photoPickerIntent.setType("image/*");
startActivityForResult(photoPickerIntent, SELECT_IMAGE);
selectedImage = (ImageView) findViewById(R.id.ImageView);

```

```

ImageView imageView ;
imageView = (ImageView)findViewById(R.id.ImageView);
imageView.setImageURI(chosenImageUri);

```

Όταν θα πατήσουμε, ακουμπήσουμε την φωτογραφία ή όταν θα πατήσουμε το κουμπί του μενού της συσκευής μας, εμφανίζουμε το μενού που αναφέραμε παραπάνω (add_info_menu.xml).

```

openOptionsMenu();

```

Η μέθοδος για την δημιουργία του μενού.

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.add_info_menu, menu);
    return true;
}

```

Η μέθοδος που χρησιμοποιούμε για την υλοποίηση των λειτουργιών της κάθε επιλογής του μενού.

```

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle item selection
    switch (item.getItemId()) {

        default:
            return super.onOptionsItemSelected(item);
    }
}

```

```
}
```

Αν ο χρήστης επιλέξει να προσθέσει κάποια περιγραφή στη φωτογραφία τότε ανοίγουμε έναν διάλογο (dialog).

```
case R.id.new_description:
    //Create dialog
    View view = View.inflate(this, R.layout.add_description, null);
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setMessage("Add your description:")
        .setCancelable(false)
        .setPositiveButton("Save", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                DatabaseHelper.addDescription(getApplicationContext(), "Yellow",
"Green");
                dialog.dismiss();
            }
        })
        .setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                Log.d("Cancel desc", "Cancel");
                dialog.dismiss();
            }
        });
    builder.setCancelable(true);
    builder.setView(view);
    builder.setInverseBackgroundForced(true);
    AlertDialog alert = builder.create();
    alert.show();
    return true;
```

Στην περίπτωση που ο χρήστης επιλέξει να επισημάνει κάποιο πρόσωπο ξεκινάει μια διεργασία η οποία ανοίγει τη λίστα με τις επαφές που είναι αποθηκευμένες στη συσκευή.

```
case R.id.new_contact:
    Intent i_contacts= new Intent(Intent.ACTION_PICK,
Contacts.CONTENT_URI);
    startActivity(i_contacts);
    return true;
```

Η τελευταία περίπτωση είναι αυτή που ο χρήστης μπορεί να θέλει την επιλογή μιας άλλης φωτογραφίας, άρα να εμφανίσει το σύνολο των φωτογραφιών της Photo Gallery.

```
case R.id.choose_image:
    Intent photoPickerIntent = new Intent(Intent.ACTION_GET_CONTENT);
    photoPickerIntent.setType("image/*");
    startActivityForResult(photoPickerIntent, SELECT_IMAGE);
    return true;
```

5.2.4.3 Η κλάση DatabaseHelper.java

Η κλάση DatabaseHelper.java αφορά την δημιουργία της βάσης όπου θα αποθηκεύονται τα δεδομένα μας. Όπως έχουμε προαναφέρει, η δημιουργία της βάσης δεδομένων στην εφαρμογή μας γίνεται μέσω SQLite. Η δημιουργία της βάσης μας γίνεται με την εντολή `public void onCreate(SQLiteDatabase db)`, ενώ η ενημέρωση των βάσεων γίνεται με την εντολή `public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)`.

Στην συνέχεια δημιουργούμε τις συναρτήσεις `addDescription` και `addContact` μέσω των οποίων θα συνδέσουμε τα δεδομένα μας με την αντίστοιχη εικόνα. Μόλις ο χρήστης επιλέξει να προσθέσει περιγραφή ή επισήμανση σε μία φωτογραφία δημιουργείται ή ανανεώνεται το αρχείο `image_data12.db`.

Η βάση αποτελείται από δυο πίνακες, τον `IMAGE_DESCRIPTION` και τον `IMAGE_CONTACT`. Η δημιουργία ενός πίνακα γίνεται με την εντολή `CREATE_TABLE_NAME`, ενώ η δημιουργία ενός

στιγμιότυπου της βάσης δεδομένων γίνεται με τον κώδικα `onCreate(SQLiteDatabase db)`. Για την εισαγωγή νέων εγγραφών χρησιμοποιούμε

```
ContentValues contentValues=new ContentValues();
contentValues.put(COL_IMAGE_ID, imageId);
contentValues.put(COL_DESCRIPTION,description);
db.insertOrThrow(TBL_DESCRIPTION,null , contentValues);
```

, όπου η μέθοδος `put` επιστρέφει το `id` της εγγραφής που μόλις δημιουργήθηκε, ενώ η `insertOrThrow` προκαλεί μια εξαίρεση `SQLException` σε περίπτωση αποτυχίας. Το αντικείμενο `ContentValues` χρησιμοποιείται για να αντιστοιχίσουμε τα ονόματα των στηλών με τις τιμές των στηλών για την εγγραφή που θέλουμε να εισάγουμε. Οι εντολές που αναφέραμε παραπάνω αφορούν μια συναλλαγή και βρίσκονται σε ένα `try` μετά από την κλήση της `db.beginTransaction()`; η οποία δηλώνει ότι ή θα εκτελεστούν όλες οι εντολές μέσα στο `try- block` ή καμία. Πριν το κλείσιμο του `try- block` με την κλήση της `db.setTransactionSuccessful()`; Πραγματοποιούμε τις αλλαγές μας. Αν δεν γίνει η κλήση της τότε όλες οι ενέργειες αναιρούνται και δεν υποβάλλονται. Με την κλήση της `db.endTransaction()`; Τερματίζουμε την συναλλαγή. Την συναλλαγή για την εισαγωγή εγγραφών την χρησιμοποιούμε για την δημιουργία εγγραφών σχετικά με την περιγραφή της εικόνας, αφού οι επαφές υπάρχουν ήδη στην συσκευή μας.

Ο `IMAGE_DESCRIPTION` αφορά την προσθήκη περιγραφής και η δημιουργία του γίνεται με τον εξής κώδικα

```
private final String CREATE_DESCRIPTION_TABLE="create table if not exists " +
TBL_DESCRIPTION + "( " + BaseColumns._ID
+ " integer primary key autoincrement, " + COL_IMAGE_ID+ " text not null, "
+ COL_DESCRIPTION + " text);";
```

και περιέχει τα πεδία `id`, που είναι το πρωτεύον κλειδί, το `COL_IMAGE_ID`, που είναι το `id` της φωτογραφίας και το `COL_DESCRIPTION`, που αφορά την περιγραφή που δίνει ο χρήστης. Στην συνέχεια έχουμε τις μεθόδους με τις οποίες είτε δημιουργούμε την βάση μας είτε την ανανεώνουμε.

```
@Override
public void onCreate(SQLiteDatabase db) {
    String sql = CREATE_DESCRIPTION_TABLE;
    db.execSQL(sql);
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    if (oldVersion >= newVersion)
        return;
}

public static void addDescription(Context ctx, String imageId,
    String description) {
    SQLiteOpenHelper sqliteOpenHelper = getHelper(ctx);
    try {
        SQLiteDatabase db=sqliteOpenHelper.getWritableDatabase();

        db.beginTransaction();
        try{
            ContentValues contentValues=new ContentValues();
            contentValues.put(COL_IMAGE_ID, imageId);
            contentValues.put(COL_DESCRIPTION,description);
            db.insertOrThrow(TBL_DESCRIPTION,null , contentValues);
            db.setTransactionSuccessful();
            Log.d("db", "add description");
        }finally{
            db.endTransaction();
        }
    } finally {
        sqliteOpenHelper.close();
    }
}
```



```

}
public static void addContact(Context ctx, String imageId,
    String contactId) {
    SQLiteOpenHelper sqLiteOpenHelper = getHelper(ctx);
    try {
        SQLiteDatabase db=sqLiteOpenHelper.getWritableDatabase();

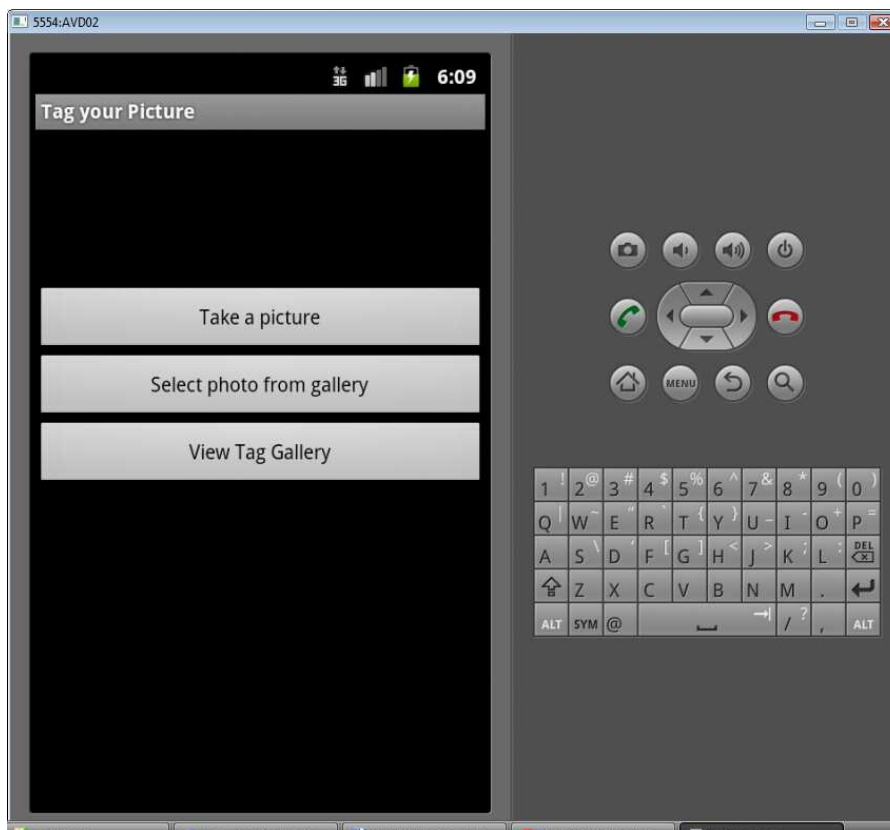
        db.beginTransaction();
        try{
            ContentValues contentValues=new ContentValues();
            contentValues.put(COL_CONTACT,contactId);
            contentValues.put(COL_IMAGE_ID, imageId);
            db.insertOrThrow(TBL_CONTACT,null , contentValues);
            db.setTransactionSuccessful();
            Log.d("", "contact tag");
        }finally{
            db.endTransaction();
        }
    } finally {
        sqLiteOpenHelper.close();
    }
}
}

```

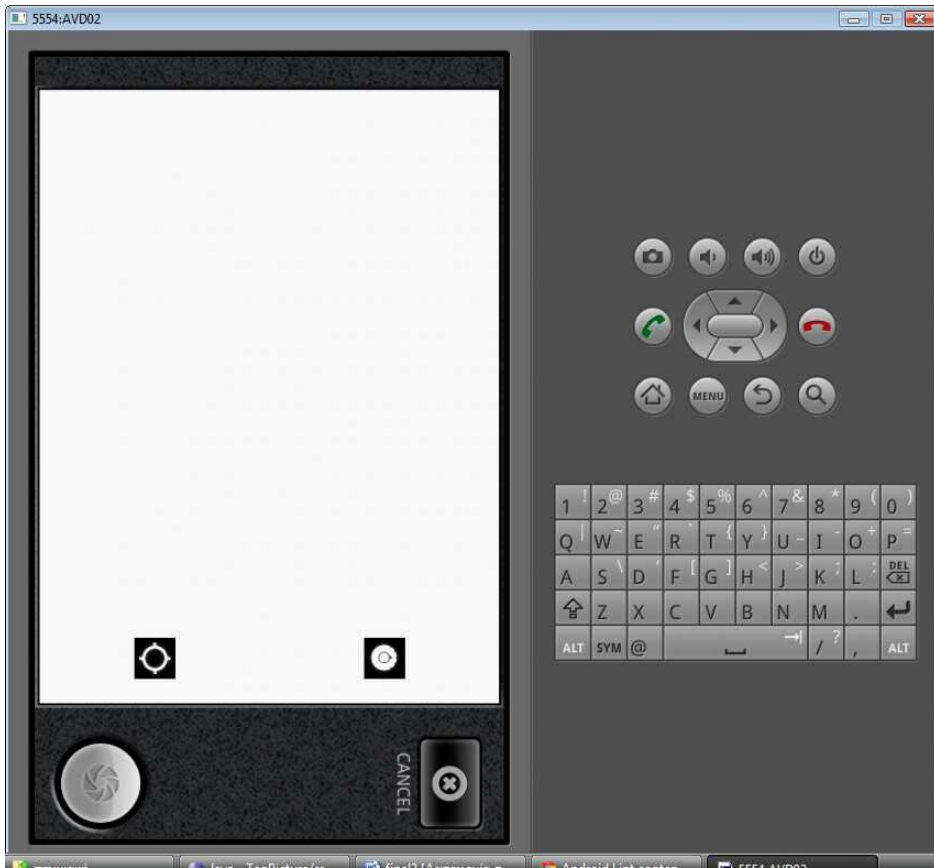
5.2.4.4 Η κλάση TagViewer.java

Η κλάση TagViewer.java αφορά την παρουσίαση των εικόνων και των δεδομένων που περιέχουν.

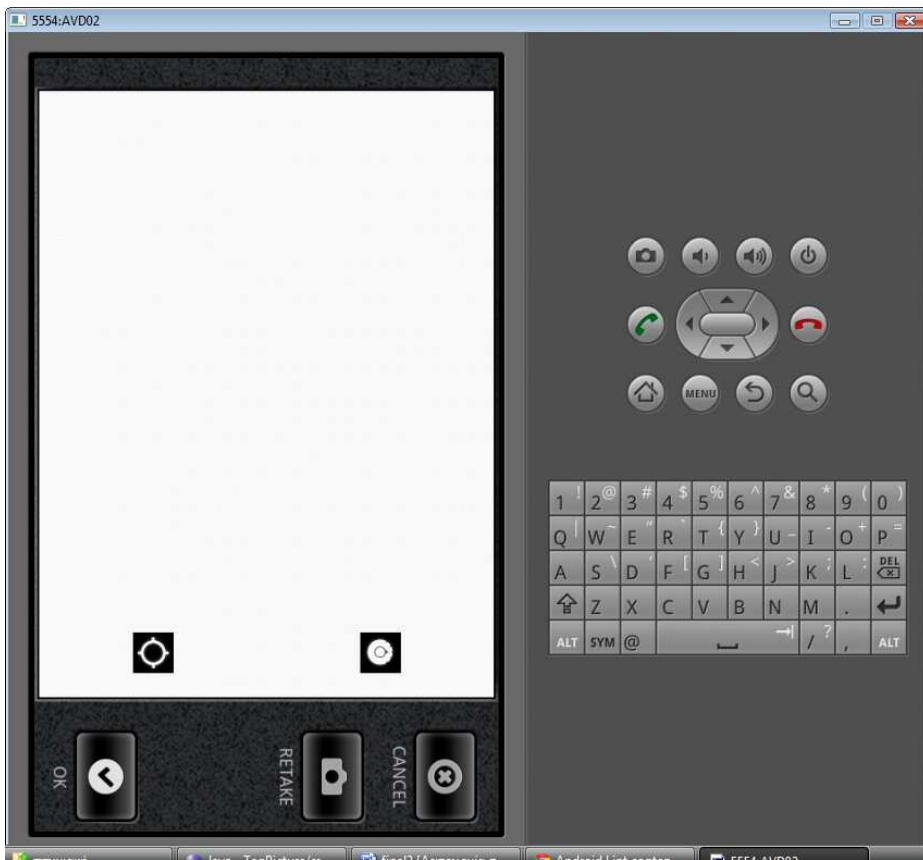
5.2.5 Απεικόνιση των οθονών της εφαρμογής



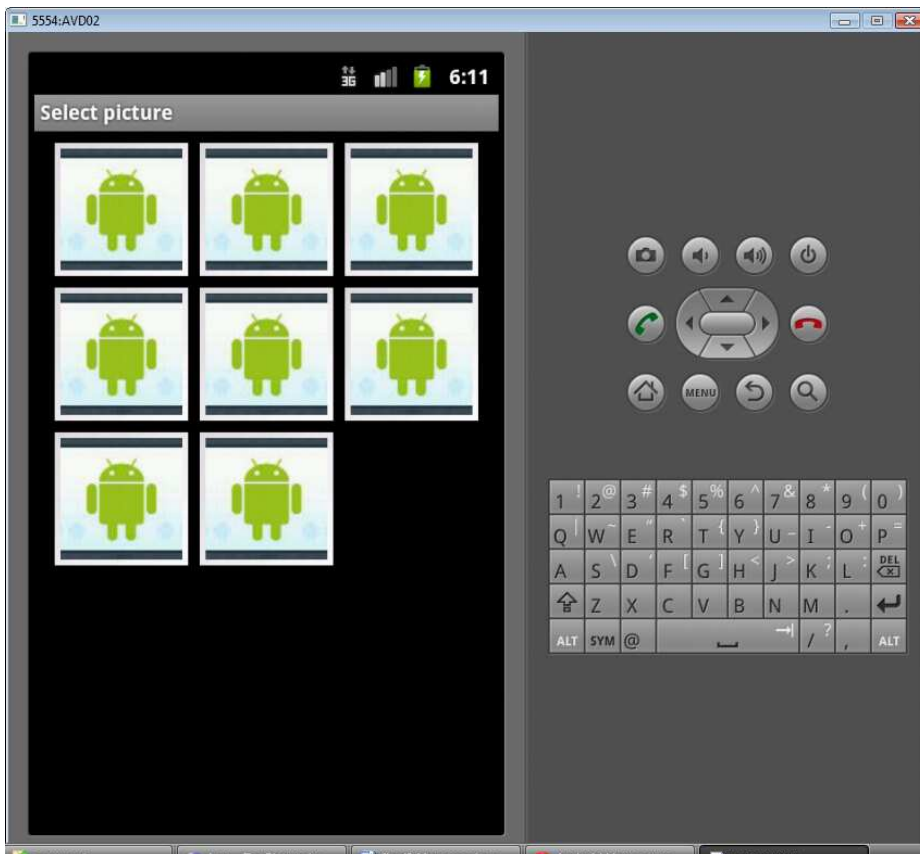
Εικόνα 7 Οθόνη 1: Λήψη ή επιλογή φωτογραφίας.



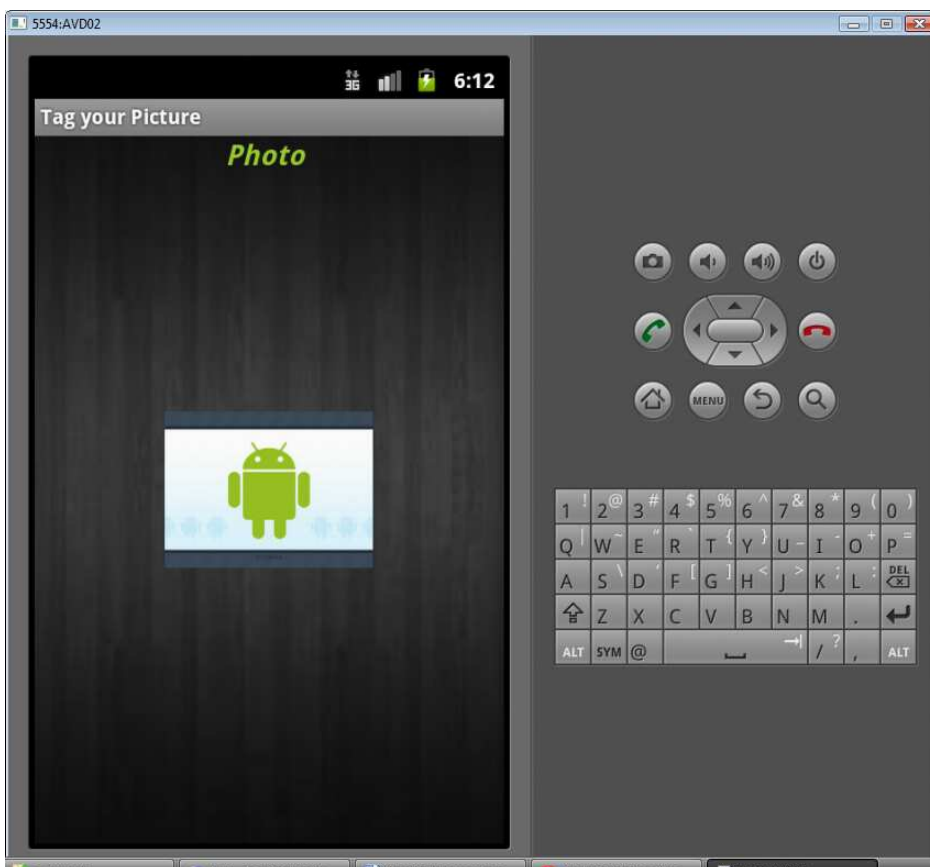
Εικόνα 8 Λήψη φωτογραφίας.



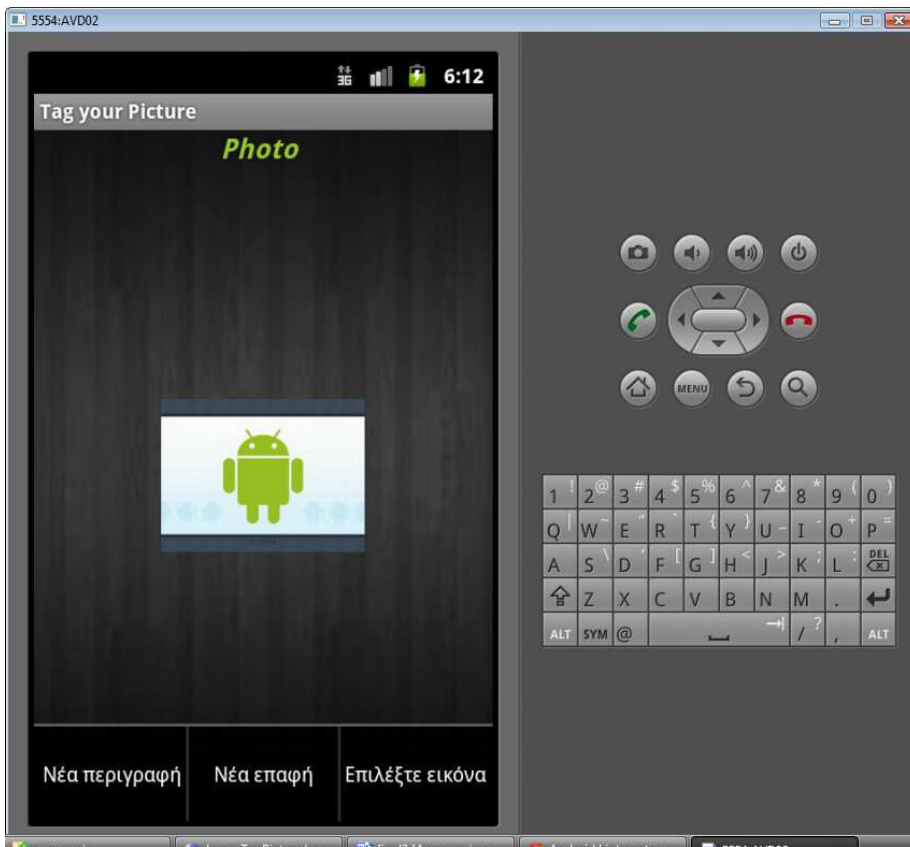
Εικόνα 9 Αποθήκευση φωτογραφίας.



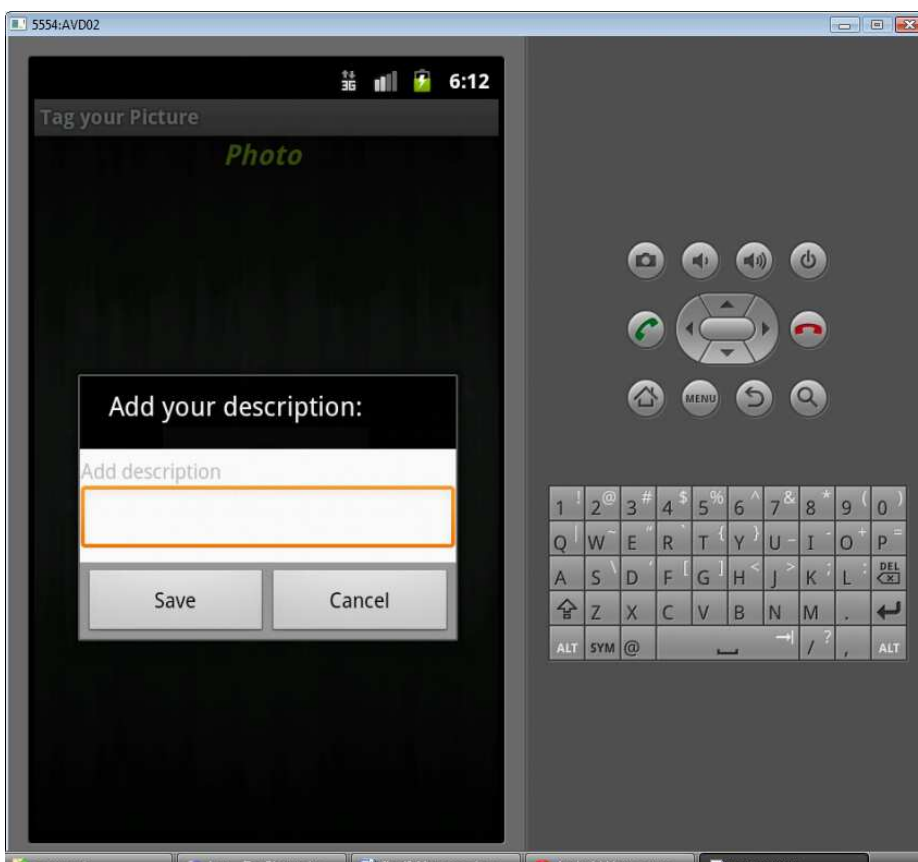
Εικόνα 10 Προβολή φωτογραφιών για επιλογή.



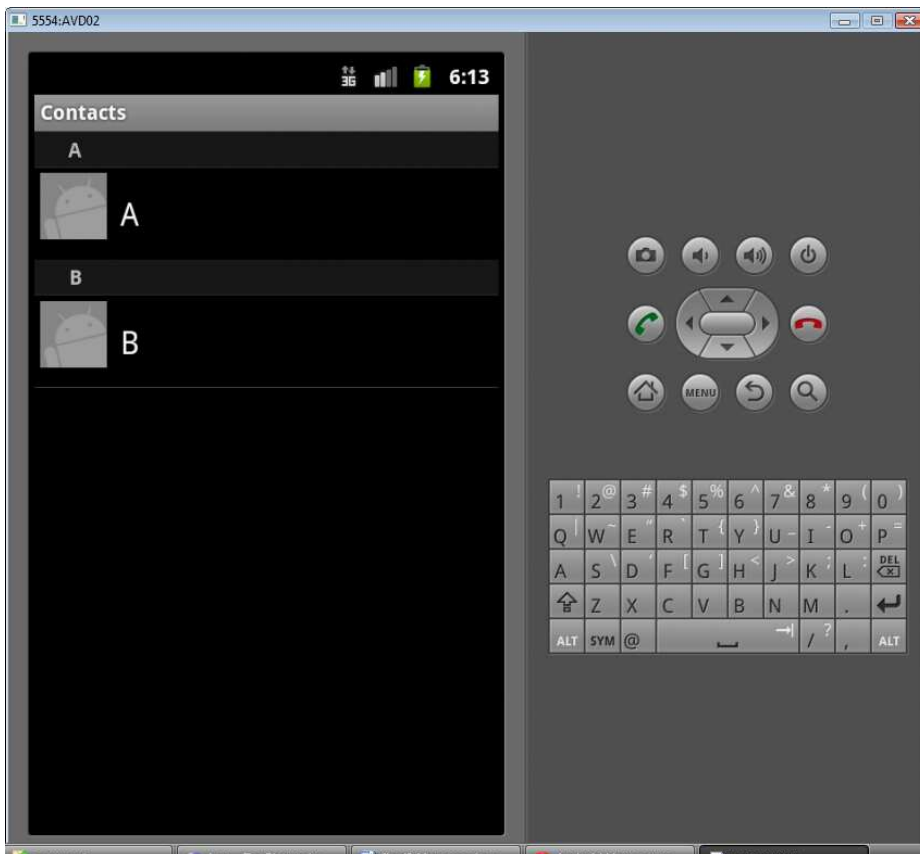
Εικόνα 11 Άνοιγμα επιλεγμένης φωτογραφίας.



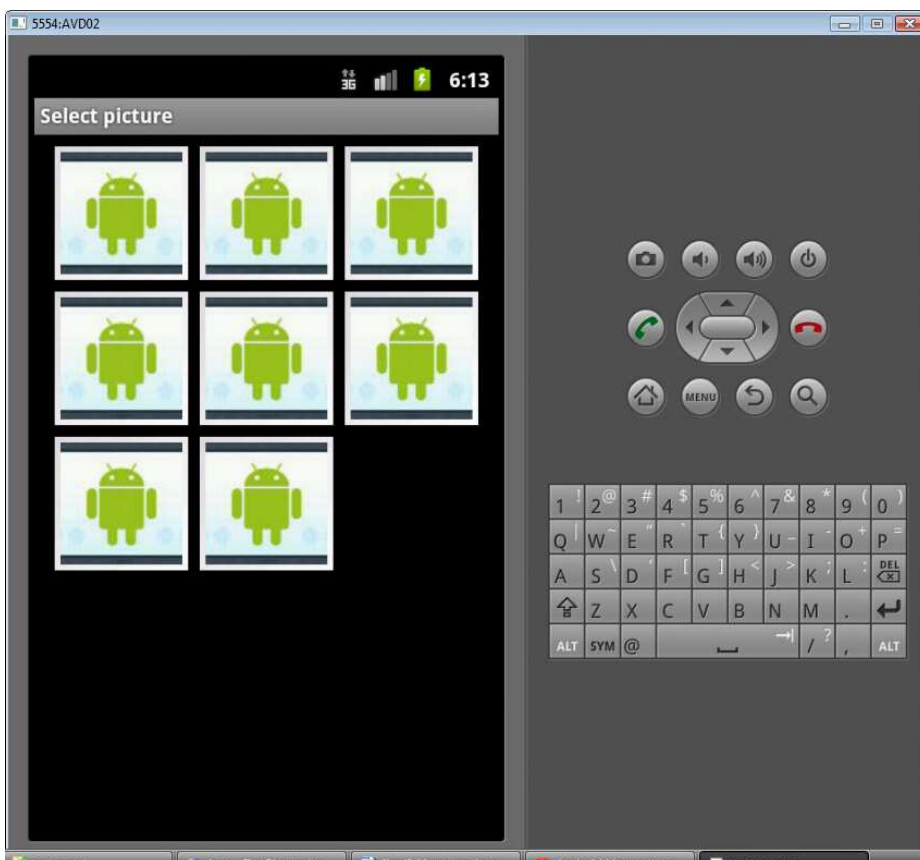
Εικόνα 12 Μενού εικόνας.



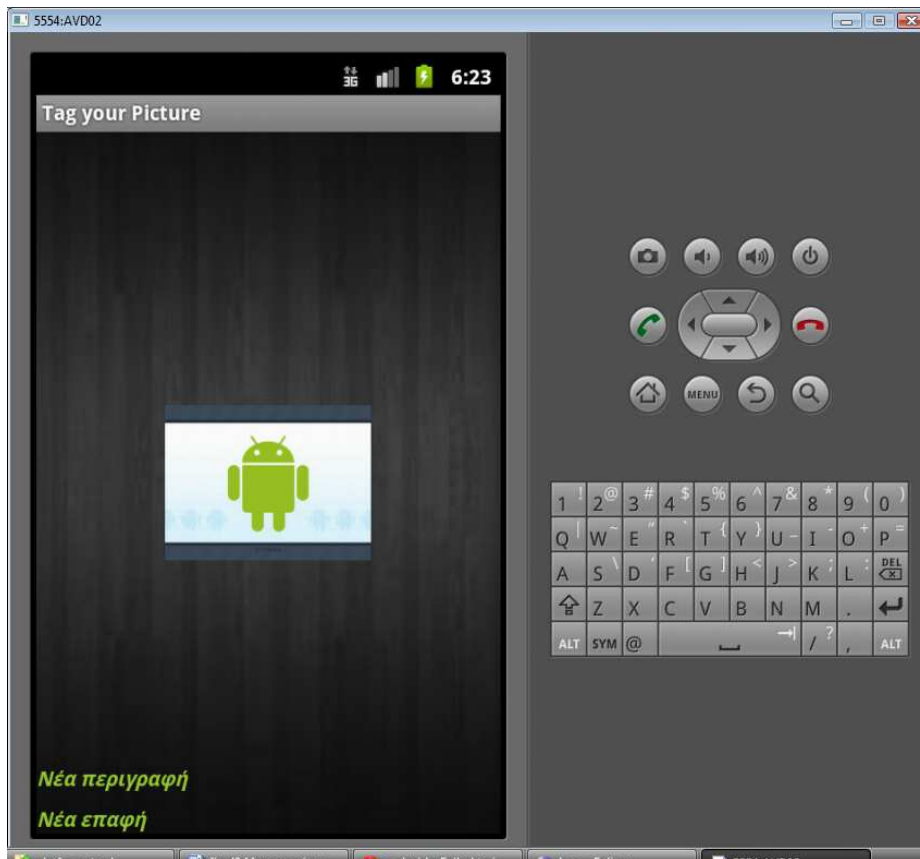
Εικόνα 13 Εισαγωγή περιγραφής.



Εικόνα 14 Επιλογή επαφής τηλεφώνου.



Εικόνα 15 Επιλογή φωτογραφίας για παρουσίαση δεδομένων



Εικόνα 16 Παρουσίαση δεδομένων

5.3 Συμπεράσματα, Μελλοντικές επεκτάσεις

5.3.1 Συμπεράσματα

Στην παρούσα πτυχιακή εργασία ασχοληθήκαμε με την ανάπτυξη μιας εφαρμογής η οποία θα εκτελείται σε κινητές συσκευές που χρησιμοποιούν το λειτουργικό σύστημα Android. Η ανάπτυξη εφαρμογών πάνω σε αυτή την πλατφόρμα κρίνεται επιτυχής για τους ακόλουθους λόγους:

- Το Android εξελίσσεται με ταχύτερους ρυθμούς, αφού οι ενεργοποιήσεις νέων συσκευών έχουν φτάσει περίπου τις 900.000 ημερησίως,
- Οι περισσότερες εταιρίες κατασκευής κινητών τηλεφώνων έχουν ήδη υιοθετήσει το λειτουργικό Android,
- Εταιρίες όπως η ASUS, η SAMSUNG, η SONY και άλλες έχουν ήδη υιοθετήσει το λειτουργικό σύστημα Android για τις ταμπλέτες τους,
- Το Android Market διαθέτει πάνω από 500.000 εφαρμογές, οι περισσότερες δωρεάν και περισσότερες από 18.000.000.000 λήψεις,
- Η ανάπτυξη απλών εφαρμογών δεν απαιτεί εξειδικευμένες γνώσεις προγραμματισμού και ο καθένας μπορεί να ασχοληθεί και να δημιουργήσει μια απλή προσωπική εφαρμογή, αρκεί να υπάρχει μια εξοικίωση με την γλώσσα προγραμματισμού java,
- Τα προγραμματιστικά εργαλεία που χρειάζονται για την ανάπτυξη των εφαρμογών διατίθενται στο διαδίκτυο δωρεάν,
- Διατίθενται αναλυτικοί οδηγοί σε αρκετές ιστοσελίδες στο διαδίκτυο, όπως στην επίσημη ιστοσελίδα της Google, οι οποίοι βοηθάνε τον προγραμματιστή να εξοικιωθεί με την νέα αυτή τεχνολογία.

Κατά τη διάρκεια εκπόνησης αυτής της πτυχιακής εργασίας αποκτήθηκαν σημαντικές γνώσεις σχετικά με τις τεχνολογίες των έξυπνων κινητών τηλεφώνων και τον τρόπο ανάπτυξης εφαρμογών σε αυτά.

5.3.2 Μελλοντικές επεκτάσεις

Οι προγραμματιστές έχουν δυνατότητα δημοσίευσης των εφαρμογών τους στο Android Market ελεύθερα και χωρίς κόστος. Η διαδικασία που ακολουθείτε είναι απλή, εφόσον ανεβάζεις το εκτελέσιμο αρχείο apk της εφαρμογής, και ανά πάσα στιγμή οι χρήστες Android μπορούν να κατεβάσουν την εφαρμογή.

Η δημοσίευση της συγκεκριμένης εφαρμογής στο Android Market σχεδιάζεται να γίνει στο προσεχές μέλλον. Πριν όμως από τη δημοσίευση θα πρέπει να γίνουν δοκιμές και βελτιώσεις.

- Βελτιωμένη διεπαφή χρήστη, ώστε το περιβάλλον της εφαρμογής μας να γίνει πιο φιλικό,
- Λήψη φωτογραφιών,
- Προβολή δεδομένων φωτογραφίας,
- Δυνατότητα μετακίνησης στην επόμενη φωτογραφία,
- Δημιουργία φίλτρου για να γίνεται η προβολή των φωτογραφιών που έχουν είτε περιγραφή είτε επισήμανση.

Βιβλιογραφία

1. www.google.com (search)
2. en.wikipedia.org
3. Ανάπτυξη εφαρμογών με το Android, Δεύτερη έκδοση, Lauren Darcey- Shane Conder, Εκδόσεις Γκιούρδας
4. The developer's guide, <http://developer.android.com>
5. <http://stackoverflow.com/>
6. <http://www.vogella.de/>
7. <http://en.wikipedia.org>
8. <http://www.techi.com/>
9. <http://www.appbrain.com/>
10. <http://www.appbrain.com/>
11. <http://ventrix.nsdg.gr>
12. <http://xsomaras.qv-web.eu/>
13. <http://ifestos.teilar.gr/>

TagPicture/res/layout

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.tagpicture"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="8" />

    <uses-permission android:name="android.permission.CAMERA"/>
    <uses-permission android:name="android.permission.INTERNET"></uses-permission>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"></uses-
permission>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"></uses-
permission>
    <uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"></uses-
permission>
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"></uses-
permission>
    <uses-feature android:name="android.hardware.screen.portrait"/>
    <application
        android:icon="@drawable/icon"
        android:label="@string/app_name" >
        <activity
            android:label="@string/app_name"
            android:name=".Selection" >
            <intent-filter >
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:label="@string/app_name"
            android:name=".PhotoViewer" >
            <intent-filter >
                <action android:name="android.intent.action.SCREEN2" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:label="@string/app_name"
            android:name=".TagViewer" >
            <intent-filter >
                <action android:name="android.intent.action.TAGVIEW" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

```

</activity>
<activity
    android:label="@string/app_name"
    android:name=".AddDescription" >
    <intent-filter >
        <action android:name="android.intent.action.ADD_DESCRIPTION" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>

</application>

```

```

</manifest>

```

Main.xml

```

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

    <Button
        android:id="@+id/btnCamera"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Take a picture"
        android:layout_marginTop="50pt" />

    <Button
        android:id="@+id/btnGallery"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Select photo from gallery" />

    <Button
        android:id="@+id/btnViewGallery"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="View Tag Gallery" />

    <Gallery xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/gallery"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
    />
</LinearLayout>

```

Screen2.xml

```

<?xml version="1.0" encoding="utf-8"?>

```

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/bg"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/label"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Photo"
        android:textColor="#9ACD32"
        android:textSize="28px"
        android:paddingLeft="130dp"
        android:textStyle="bold|italic"/>

    <ImageView
        android:id="@+id/ImageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true" />

</RelativeLayout>

```

Photogallery.xml

```

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

</RelativeLayout>

```

Add_description.xml

```

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

    <TextView
        android:id="@+id/label"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Add description" />

    <EditText
        android:id="@+id/editText"

```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textMultiLine" >
```

```
    <requestFocus />
</EditText>
</LinearLayout>
```

Galleryopen.xml

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

</LinearLayout>
```

Tagview.xml

```
<?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"
    >

<Gallery
    android:id="@+id/Gallery01"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"></Gallery>
<ImageView
    android:id="@+id/ImageView01"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"></ImageView>
</LinearLayout>
```

TagPicture/res/menu

Add_menu_info.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
    <item android:id="@+id/new_description"
        android:title="@string/new_description" />
    <item android:id="@+id/new_contact"
        android:title="@string/new_contact" />
    <item android:id="@+id/choose_image"
        android:title="@string/choose_image" />
</menu>
```

</menu>

TagPicture/res/values

attrs.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <declare-styleable name="Gallery1">
    <attr name="android:galleryItemBackground" />
  </declare-styleable>
</resources>
```

Strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

  <string name="app_name">Tag your Picture</string>
  <string name="new_description">Νέα περιγραφή</string>
  <string name="new_contact">Νέα επαφή</string>
  <string name="choose_image">Επιλέξτε εικόνα</string>

</resources>
```

TagPicture/src/com/example/tagpicture

DatabaseHelper.java

```
package com.example.tagpicture;

import java.util.ArrayList;
import java.util.List;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.provider.BaseColumns;
import android.util.Log;

public abstract class DatabaseHelper {
  private static final String COL_IMAGE_ID="IMAGE_ID";

  private static final String TBL_DESCRIPTION="IMAGE_DESCRIPTION";
  private static final String COL_DESCRIPTION="DESCRIPTION";

  private static final String TBL_CONTACT="IMAGE_CONTACT";
  private static final String COL_CONTACT="CONTACT";
```

```

private static class ImageDataSQLHelper extends SQLiteOpenHelper {
    private static final String DATABASE_NAME = "image_data.db";
    private static final int DATABASE_VERSION = 1;

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

    private final String CREATE_DESCRIPTION_TABLE="create table if not exists " +
TBL_DESCRIPTION + "( " + BaseColumns._ID
        + " integer primary key autoincrement, " + COL_IMAGE_ID+ " text not null, "
        + COL_DESCRIPTION + " text);";

    @Override
    public void onCreate(SQLiteDatabase db) {
        String sql = CREATE_DESCRIPTION_TABLE;
        Log.d("EventsData", "onCreate: " + sql);
        db.execSQL(sql);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        if (oldVersion >= newVersion)
            return;

        //TODO: Find a way to do it!!
    }
}

private static SQLiteOpenHelper getHelper(Context ctx) {
    return new ImageDataSQLHelper(ctx);
}

public static void addDescription(Context ctx, String imageId,
    String description) {
    SQLiteOpenHelper sqlLiteOpenHelper = getHelper(ctx);
    try {
        SQLiteDatabase db=sqlLiteOpenHelper.getWritableDatabase();

        db.beginTransaction();
        try{
            ContentValues contentValues=new ContentValues();
            contentValues.put(COL_IMAGE_ID, imageId);
            contentValues.put(COL_DESCRIPTION,description);
            db.insertOrThrow(TBL_DESCRIPTION,null , contentValues);
            db.setTransactionSuccessful();
            Log.d("db", "add description");
        }
    }
}

```

```

        }finally{
            db.endTransaction();
        }
    } finally {
        sqLiteOpenHelper.close();
    }
}

public static List<String> getDescriptions(Context ctx, String imageId) {
    List<String> result = new ArrayList<String>();

    SQLiteOpenHelper sqLiteOpenHelper = getHelper(ctx);
    try {
        Cursor c = sqLiteOpenHelper.getReadableDatabase().query(
            TBL_DESCRIPTION, new String[] { COL_DESCRIPTION },
            COL_IMAGE_ID+"=? ", new String[] { imageId }, null, null, null);

        while (c.moveToNext()){
            int columnIndex=c.getColumnIndexOrThrow(COL_DESCRIPTION);
            String description=c.getString(columnIndex);
            result.add(description);
        }
    } finally {
        sqLiteOpenHelper.close();
    }
    return result;
}

```

```

public static List<String> getContacts(Context ctx, String imageId) {
    List<String> result = new ArrayList<String>();
    Log.d("db", "Contact Tag");
    SQLiteOpenHelper sqLiteOpenHelper = getHelper(ctx);
    try {
        Cursor c = sqLiteOpenHelper.getReadableDatabase().query(
            TBL_CONTACT, new String[] { COL_CONTACT },
            COL_IMAGE_ID+"=? ", new String[] { imageId }, null, null, null);

        while (c.moveToNext()){
            int columnIndex=c.getColumnIndexOrThrow(COL_CONTACT);
            String contact=c.getString(columnIndex);
            result.add(contact);
        }
    } finally {
        sqLiteOpenHelper.close();
    }
    return result;
}
}

```


PhotoViewer.java

```
package com.example.tagpicture;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.drawable.BitmapDrawable;
import android.net.Uri;
import android.os.Bundle;
import android.provider.ContactsContract.Contacts;
import android.provider.MediaStore.Images.Media;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.ImageView;

/**
 * PhotoViewer
 *
 * @author annie
 *
 */
public class PhotoViewer extends Activity {
    private static final String LATEST_IMAGE_ID = "LATEST_IMAGE_ID";
    private String currentImageId;
    private ImageView selectedImage;
    final int SELECT_IMAGE = 123;

    public void onSaveInstanceState(Bundle savedInstanceState) {
        savedInstanceState.putString(LATEST_IMAGE_ID, currentImageId);
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.screen2);

        if (savedInstanceState != null
            && savedInstanceState.getString(LATEST_IMAGE_ID) != null) {
            String latestImageId = savedInstanceState
                .getString(LATEST_IMAGE_ID);
            Uri chosenImageUri = Uri.parse(latestImageId);
            Intent i = new Intent(getApplicationContext(), PhotoViewer.class);
            i.setData(chosenImageUri); // we add the URI as data, so that screen2
```

```

        // will know the file
        startActivity(i);
    } else {
        Intent photoPickerIntent = new Intent(Intent.ACTION_GET_CONTENT);
        photoPickerIntent.setType("image/*");
        startActivityForResult(photoPickerIntent, SELECT_IMAGE);

        selectedImage = (ImageView) findViewById(R.id.ImageView);
        Log.d("Main screen", "img selection");

        selectedImage.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                openOptionsMenu();
                Log.d("Image Clicked", "on click view");
            }
        });
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (resultCode == RESULT_OK && requestCode == SELECT_IMAGE) { // user selected image
        Uri chosenImageUri = data.getData(); // we retrieve the URI address of the image
        try {
            // Bitmap mBitmap = Media.getBitmap(this.getContentResolver(),
            // chosenImageUri); // we try to load the file as an image
            // Intent i = new Intent(getApplicationContext(), screen2.class);
            // i.setData(chosenImageUri); // we add the URI as data, so that
            // // screen2 will know the file
            // startActivity(i);
            ImageView imageView ;
            imageView = (ImageView) findViewById(R.id.ImageView);
            imageView.setImageURI(chosenImageUri);
            Log.d("", "on image view");
        } catch (Exception e) {
            // in case that we failed loading the file as image...
            // TODO: We will see how to handle exceptions..
        }
    }
}

public void onResume() {
    super.onResume();
    Uri chosenImageUri = getIntent().getData();
    if (chosenImageUri != null) {

```

```

currentImageId = chosenImageUri.toString();
ImageView imageView = (ImageView) findViewById(R.id.ImageView);
Bitmap mBitmap;
try {
    mBitmap = Media.getBitmap(this.getContentResolver(),
        chosenImageUri);
    imageView.setBackgroundDrawable(new BitmapDrawable(mBitmap));
} catch (Exception e) {
    imageView.setBackgroundResource(R.drawable.icon);
}
}
}

```

@Override

```

public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.add_info_menu, menu);
    return true;
}

```

@Override

```

public boolean onOptionsItemSelected(MenuItem item) {
    // Handle item selection
    switch (item.getItemId()) {
        case R.id.new_description:
            //Create dialog
            View view = View.inflate(this, R.layout.add_description, null);
            AlertDialog.Builder builder = new AlertDialog.Builder(this);
            builder.setMessage("Add your description:")
                .setCancelable(false)
                .setPositiveButton("Save", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int id) {
                        DatabaseHelper.addDescription(getApplicationContext(), "Yellow", "Green");
                        Log.d("Add desc", "Description saved!");
                        //List<String> results=DatabaseHelper.getDescriptions(getApplicationContext(),
"Yellow");
                        dialog.dismiss();
                    }
                })
                .setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int id) {
                        Log.d("Cancel desc", "Cancel");
                        dialog.dismiss();
                    }
                });
            builder.setCancelable(true);
            builder.setView(view);
            builder.setInverseBackgroundForced(true);
            AlertDialog alert = builder.create();

```

```

        alert.show();
        return true;
    case R.id.new_contact:
        Intent i_contacts= new Intent(Intent.ACTION_PICK, Contacts.CONTENT_URI);
        startActivity(i_contacts);
        return true;
    case R.id.choose_image:
        Intent photoPickerIntent = new Intent(Intent.ACTION_GET_CONTENT);
        photoPickerIntent.setType("image/*");
        startActivityForResult(photoPickerIntent, SELECT_IMAGE);
        return true;
    default:
        return super.onOptionsItemSelected(item);
    }
}
}
}

```

Selection.java

```

package com.example.tagpicture;

import android.app.Activity;
import android.content.ContentValues;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class Selection extends Activity {

    private static final int CAPTURE_IMAGE_ACTIVITY_REQUEST_CODE = 0;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        Button btnCamera = (Button) findViewById(R.id.btnCamera);
        Button btnGallery = (Button) findViewById(R.id.btnGallery);
        Button btnViewGallery = (Button) findViewById(R.id.btnViewGallery);
        btnCamera.setOnClickListener( new OnClickListener(){
            public void onClick(View v ){
                //define the file-name to save photo taken by Camera activity
                String fileName = "new-photo-name.jpg";
            }
        });
    }
}

```

```

//create parameters for Intent with filename
ContentValues values = new ContentValues();
values.put(MediaStore.Images.Media.TITLE, fileName);
values.put(MediaStore.Images.Media.DESCRPTION,"Image capture by camera");
//imageUri is the current activity attribute, define and save it for later usage (also in
onSaveInstanceState)
Uri imageUri = getContentResolver().insert(
    MediaStore.Images.Media.EXTERNAL_CONTENT_URI, values);
//create new Intent
Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
intent.putExtra(MediaStore.EXTRA_OUTPUT, imageUri);
startActivityForResult(intent, CAPTURE_IMAGE_ACTIVITY_REQUEST_CODE);
}
});
btnGallery.setOnClickListener( new OnClickListener(){
    public void onClick(View v ){
        Intent i = new Intent(Selection.this, PhotoViewer.class);
        startActivity(i);
    }
});
btnViewGallery.setOnClickListener( new OnClickListener(){
    public void onClick(View v ){
        Intent i = new Intent(Selection.this, TagViewer.class);
        startActivity(i);
        Log.d("", "view gal selection");
    }
});
}

protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == CAPTURE_IMAGE_ACTIVITY_REQUEST_CODE) {
        if (resultCode == RESULT_OK) {
            //use imageUri here to access the image
            Log.d("", "access");
            sendBroadcast(new Intent(Intent.ACTION_MEDIA_MOUNTED,
                Uri.parse("file://" + Environment.getExternalStorageDirectory())));
        } else if (resultCode == RESULT_CANCELED) {
            Log.d("", "canceled");
        } else {
            Log.d("", "not taken");
        }
    }
}
}
}

```

TagViewer.java

```
package com.example.tagpicture;
import android.app.Activity;
import android.content.Context;
import android.content.res.TypedArray;
import android.os.Bundle;
import android.os.CountDownTimer;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.Gallery;
import android.widget.ImageView;
import android.widget.Toast;
import android.widget.AdapterView.OnItemClickListener;

public class TagViewer extends Activity {
    Integer[] pics = {
        R.drawable.icon,
        R.drawable.bg,
        R.drawable.c,
        R.drawable.ic_launcher
    };
    ImageView imageView;

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

        Gallery ga = (Gallery)findViewById(R.id.Gallery01);
        ga.setAdapter(new ImageAdapter(this));

        imageView = (ImageView)findViewById(R.id.ImageView01);

        ga.setOnItemClickListener(new OnItemClickListener() {

            @Override
            public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
                long arg3) {

                final Toast tag = Toast.makeText(getBaseContext(),
                    "Tagged people " + (arg2+1) +
                    "Description " + (arg2+1) ,
                    Toast.LENGTH_LONG);

                tag.show();
            }
        });
    }
}
```

```

        new CountdownTimer(9000, 1000)
        {

            public void onTick(long millisUntilFinished) {tag.show();}
            public void onFinish() {tag.show();}

        }.start();

        imageView.setImageResource(pics[arg2]);

    }

});
}

```

```

public class ImageAdapter extends BaseAdapter {

    private Context ctx;
    int imageBackground;

    public ImageAdapter(Context c) {
        ctx = c;
        TypedArray ta = obtainStyledAttributes(R.styleable.Gallery1);
        imageBackground = ta.getResourceId(R.styleable.Gallery1_android_galleryItemBackground, 1);
        ta.recycle();
    }

    @Override
    public int getCount() {

        return pics.length;
    }

    @Override
    public Object getItem(int arg0) {

        return arg0;
    }

    @Override
    public long getItemId(int arg0) {

        return arg0;
    }

    @Override
    public View getView(int arg0, View arg1, ViewGroup arg2) {

```

```
    ImageView iv = new ImageView(ctx);
    iv.setImageResource(pics[arg0]);
    iv.setScaleType(ImageView.ScaleType.FIT_XY);
    iv.setLayoutParams(new Gallery.LayoutParams(250,220));
    iv.setBackgroundResource(imageBackground);
    return iv;
}
}
}
```