



ΤΕΙ Κρήτης

Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

**Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων**



"Εφαρμογή Android για λήψη ανακοινώσεων"
("Android application for receiving announcements")

Πτυχιακή Εργασία

Κρουσταλάκης Νικόλαος

Επιβλέπων Καθηγητής: Παπαδάκης Νικόλαος

Ηράκλειο, Απρίλιος 2015



ΤΕΙ Κρήτης

Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

**Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων**



"Εφαρμογή Android για λήψη ανακοινώσεων"
("Android application for receiving announcements")

Πτυχιακή Εργασία

Κρουσταλάκης Νικόλαος
(Αριθμός Μητρώου: 2781)

Επιβλέπων Καθηγητής: Παπαδάκης Νικόλαος
Επίκουρος Καθηγητής

Ηράκλειο, Απρίλιος 2015

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία έχει στόχο τη μελέτη, κατανόηση και ανάπτυξη μιας εφαρμογής για έξυπνα κινητά (smartphones). Η εφαρμογή έχει αναπτυχθεί για το γνωστό πλέον λειτουργικό σύστημα Android και για τη δημιουργία της έχει μελετηθεί και κατανοηθεί το λειτουργικό αυτό έως ένα επίπεδο.

Αρχικά, αναφέρονται τα πιο δημοφιλή λειτουργικά συστήματα που χρησιμοποιούνται σήμερα στα κινητά τηλέφωνα (και γενικότερα στις κινητές συσκευές). Μέσω μίας σύγκρισης των λειτουργικών αυτών, επεξηγούνται οι λόγοι για τους οποίους επιλέχθηκε η ανάπτυξη μιας εφαρμογής στο Android. Ακολουθεί ανάλυση των ενδότερων επιμέρους στοιχείων του συγκεκριμένου λειτουργικού, καθώς και μια μικρή ιστορική αναφορά στις εκδόσεις και τα χαρακτηριστικά του. Στη συνέχεια, γίνεται μια παρουσίαση των διαφόρων εργαλείων ανάπτυξης λογισμικού και εφαρμογών στη συγκεκριμένη πλατφόρμα.

Τέλος, γίνεται η παρουσίαση της εφαρμογής μου και θα δείτε βήμα προς βήμα πώς αυτή αναπτύχθηκε με τη χρήση του λειτουργικού SDK (Software Development Kit) στο πρόγραμμα Eclipse.

Λέξεις κλειδιά: Εφαρμογή για Κινητά Τηλέφωνα, Λειτουργικό Σύστημα Android.

ABSTRACT

This thesis aims to study, understand and develop an application for smartphones. This application has been developed in the famous Android operating system and for its creation, this operating system has been studied and understood up to a certain level.

First, are the most popular operating systems currently used in mobile phones (and more generally on mobile devices). Through a comparison of these systems, explaining the reasons why I selected the development of an application in Android. Then will follow the interior individual elements of this operating system and a small historical reference to its versions and features. Then, the different software development tools and applications on this platform are presented.

Finally, my application is presented and you will see step by step how it was developed using the operating SDK (Software Development Kit) in the Eclipse program.

Key Words: Mobile Application, Android Operating System.

ΕΥΧΑΡΙΣΤΙΕΣ

Η πτυχιακή εργασία, πραγματοποιήθηκε από τον προπτυχιακό φοιτητή του Τμήματος Εφαρμοσμένης Πληροφορικής και Πολυμέσων του ΤΕΙ Κρήτης, Κρουσταλάκη Νικόλαο.

Μέσω της παρούσας πτυχιακής εργασίας μου δόθηκε η δυνατότητα να διευρύνω τις γνώσεις μου στον προγραμματισμό για κινητά τηλέφωνα, εργαζόμενος πάνω σε μια εφαρμογή για έξυπνα κινητά τηλέφωνα σε λογισμικό Android.

Θα ήθελα να ευχαριστήσω όλους όσους με στήριξαν, καθώς και τον επόπτη Καθηγητή Παπαδάκη Νικόλαο.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ	4
ABSTRACT	5
ΕΥΧΑΡΙΣΤΙΕΣ	6
ΛΙΣΤΑ ΕΙΚΟΝΩΝ.....	9
ΛΙΣΤΑ ΑΚΡΩΝΥΜΙΩΝ	11
1 Εισαγωγή.....	12
1.1 Αντικείμενο της Πτυχιακής Εργασίας.....	12
1.2 Δομή της Πτυχιακής Εργασίας.....	12
2 Εισαγωγή στο Android.....	13
2.1 Τι εστί Android και πού χρησιμοποιείται.....	13
2.2 Σύγκριση του Android με άλλα λειτουργικά συστήματα για smartphones.....	13
2.2.1 Σύγκριση του Android ως προς τη λειτουργικότητά του	14
2.2.2 Σύγκριση του Android ως προς την ευκολία ανάπτυξης εφαρμογής	15
2.3 Το Android και η αρχιτεκτονική του.....	18
2.3.1 Διάγραμμα της αρχιτεκτονικής Android	18
2.3.2 Επίπεδα της αρχιτεκτονικής Android.....	18
2.4 Εκδόσεις Android.....	20
2.4.1 Android 1.0 Apple Pie.....	21
2.4.2 Android 1.1 Banana Bread	21
2.4.3 Android 1.5 Cupcake.....	22
2.4.4 Android 1.6 Donut.....	23
2.4.5 Android 2.0-2.1 Eclair.....	23
2.4.6 Android 2.2-2.2.3 Froyo.....	24
2.4.7 Android 2.3-2.3.7 Gingerbread	25
2.4.8 Android 3.0-3.2.6 Honeycomb.....	26
2.4.9 Android 4.0-4.0.4 Ice Cream Sandwich	27
2.4.10 Android 4.1-4.3.1 Jelly Bean.....	28
2.4.11 Android 4.4-4.4.4 KitKat	28
2.4.12 Android 5.0-5.0.2 Lollipop.....	29
2.5 Το Android στην αγορά.....	30
3 Εργαλεία για δημιουργία Android εφαρμογής.....	31
3.1 Προγραμματιστικό περιβάλλον του Eclipse.....	31
3.2 WiFi ADB	33
4 Εφαρμογή Android για λήψη ανακοινώσεων.....	37
4.1 Αρχική οθόνη της εφαρμογής	37
4.2 Οθόνη με τις τελευταίες ανακοινώσεις της ιστοσελίδας.....	39
4.3 Αποσύνδεση χρήστη ή διατήρηση σύνδεσης.....	40

5	Πηγαίος κώδικας της εφαρμογής	42
5.1	Δομή της εφαρμογής μας	42
5.2	Αναλύοντας την εφαρμογή μας.....	45
5.2.1	Αρχική οθόνη εφαρμογής.....	45
5.2.2	Σύνδεση με το διαδίκτυο, αποστολή και παραλαβή πακέτων	49
5.2.3	JSON	50
5.2.4	Εμφάνιση ανακοινώσεων	52
5.2.5	RSS feed μέσω AsyncTask	55
6	Προτάσεις βελτίωσης της εφαρμογής	59
7	Επίλογος.....	60

ΛΙΣΤΑ ΕΙΚΟΝΩΝ

Εικόνα 1: Android logo	12
Εικόνα 2: Ανταγωνισμός μεταξύ IOS και Android.....	13
Εικόνα 3: Εταιρίες ανάπτυξης λογισμικού και κατασκευής υλικού	14
Εικόνα 4: Προγραμματισμός Android στο Eclipse.....	15
Εικόνα 5: Προγραμματισμός iOS στο Xcode	15
Εικόνα 6: Προγραμματισμός Windows Phone στο Visual Studio Express	16
Εικόνα 7: Διάγραμμα αρχιτεκτονικής Android.....	18
Εικόνα 8: Android OS Architecture	19
Εικόνα 9: Android Ιστορικό Εκδόσεων	19
Εικόνα 10: Android Apple Pie	20
Εικόνα 11: Android Banana Bread.....	20
Εικόνα 12: Android Cupcake	21
Εικόνα 13: Android Donut	22
Εικόνα 14: Android Éclair.....	22
Εικόνα 15: Android Froyo	23
Εικόνα 16: Android Gingerbread	24
Εικόνα 17: Android Honeycomb.....	25
Εικόνα 18: Android Cream Sandwich.....	26
Εικόνα 19: Android Jelly Bean	26
Εικόνα 20: Android KitKat	27
Εικόνα 21: Android Lollipop	28
Εικόνα 22: Εταιρίες ανάπτυξης λογισμικού και κατασκευής υλικού	29
Εικόνα 23: Λογότυπο του προγράμματος	30
Εικόνα 24: Αρχική οθόνη του Eclipse	31
Εικόνα 25: Λογότυπο του WiFi ADB	32
Εικόνα 26: Αρχική οθόνη του WiFi ADB.....	33
Εικόνα 27: Αρχική οθόνη του WiFi ADB (ON)	34
Εικόνα 28: Σύνδεση WiFi ADB με Eclipse	35
Εικόνα 29: Αρχική σελίδα της Notices_APP.....	36
Εικόνα 30: Οθόνη κατά τη σύνδεση με την ιστοσελίδα.....	37
Εικόνα 31: Ανακοινώσεις του AthensWalk	38
Εικόνα 32: Κουμπί συσκευής που βοηθάει στην αποσύνδεση χρήστη.....	39
Εικόνα 33: Κουμπί Log off	39
Εικόνα 34: Αρχική οθόνη.....	40
Εικόνα 35: Δομή εφαρμογής στο Eclipse	41
Εικόνα 36: Περιεχόμενα του src	41
Εικόνα 37: Περιεχόμενα του gen	42
Εικόνα 38: Περιεχόμενα R.java	42

Εικόνα 39: Περιεχόμενα res	43
Εικόνα 40: Περιεχόμενα του strings.xml	43
Εικόνα 41: Δημιουργία xml αρχείου.....	46
Εικόνα 42: Τρόπος εμφάνισης ανακοινώσεων.....	51

ΛΙΣΤΑ ΑΚΡΩΝΥΜΙΩΝ

Application Programming Interface	API
Central Processing Unit	CPU
High Definition	HD
Integrated Development Environment	IDE
Open Handset Alliance	OHA
Open Source Software	OSS
Operating System	OS
Secure Digital	SD
Software Development Kit	SDK
Session Initiation Protocol	SIP
User Interface	UI
Virtual Private Network	VPN

1 Εισαγωγή

1.1 Αντικείμενο της Πτυχιακής Εργασίας

Βασικός στόχος της πτυχιακής εργασίας είναι η υλοποίηση της εφαρμογής που επέλεξα, η οποία υπακούει στους κανόνες του λειτουργικού συστήματος Android. Πρόκειται για μία εφαρμογή στα πλαίσια της οποίας ο χρήστης:

- Δημιουργεί λογαριασμό σύνδεσης.
- Αφού ολοκληρωθεί η επιτυχής σύνδεσή του (login), λαμβάνει αυτόματα τις τελευταίες ανακοινώσεις που αναρτώνται σε συγκεκριμένο ιστοχώρο (website). Η λήψη περιλαμβάνει τον τίτλο της εκάστοτε ανακοίνωσης με το περιεχόμενο αυτής.
- Πραγματοποιεί έξοδο από την εφαρμογή (logout).

Η εφαρμογή δύναται να λειτουργήσει σε κάθε συσκευή που φέρει λειτουργικό Android, ωστόσο έχει δοκιμαστεί σε smartphone.

Επίσης, στα πλαίσια της παρούσας πτυχιακής, αναλύονται τεχνολογίες και εργαλεία για προγραμματισμό σε λειτουργικό Android (Android Development), καθώς και η αρχιτεκτονική του συστήματος με τα κυριότερα τμήματά της.

1.2 Δομή της Πτυχιακής Εργασίας

Η πτυχιακή αυτή εργασία δομείται από 7 κεφάλαια:

Κεφάλαιο 1 – Εισαγωγή: Περικλείει το αντικείμενο της εργασίας και τη συνοπτική δομή αυτής.

Κεφάλαιο 2 – Εισαγωγή στο Android: Εισάγει το χρήστη στο λογισμικό Android, μαζί με μία γενικότερη αναφορά στα υπόλοιπα λειτουργικά συστήματα που χρησιμοποιούνται στα smartphones σήμερα.

Κεφάλαιο 3 – Εργαλεία για δημιουργία Android εφαρμογής: Περιλαμβάνει βασική ανάλυση των εργαλείων που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής.

Κεφάλαιο 4 – Εφαρμογή Android για λήψη ανακοινώσεων: Παρουσιάζει την εφαρμογή μου - προβολή των λειτουργιών αυτής, μαζί με τις αντίστοιχες οθόνες.

Κεφάλαιο 5 – Πηγαίος κώδικας της εφαρμογής: Περιέχει τον πηγαίο κώδικα της εφαρμογής.

Κεφάλαιο 6 – Προτάσεις βελτίωσης της εφαρμογής: Παρουσιάζει πιθανές βελτιώσεις της εφαρμογής.

Κεφάλαιο 7 – Επίλογος: Εκθέτει τα συμπεράσματα της εργασίας.

2 Εισαγωγή στο Android

2.1 Τι εστί Android και πού χρησιμοποιείται

Η λέξη Android αναφέρεται σε ένα λειτουργικό σύστημα για κινητά τηλέφωνα και όχι μόνο. Είναι βασισμένο στο λειτουργικό Linux και αναπτύσσεται από την Google.

Πρόκειται για λογισμικό ανοιχτού κώδικα (Open Source Software - OSS), το οποίο σημαίνει ότι ο πηγαίος κώδικάς του διατίθεται ελεύθερα σε όσους ζητούν να τον εξετάσουν ή να τον τροποποιήσουν για να τον αξιοποιήσουν σε άλλες εφαρμογές. Επιτρέπει στους κατασκευαστές λογισμικού να γράφουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java.

Το Android είναι κατά κύριο λόγο σχεδιασμένο για συσκευές με οθόνη αφής, όπως τα smartphones και τα tablets. Ωστόσο, χρησιμοποιείται ακόμη και σε κονσόλες παιχνιδιών, ψηφιακές φωτογραφικές μηχανές, ηλεκτρονικούς υπολογιστές, όπως επίσης και γυαλιά. Με την εξέλιξή του τα τελευταία χρόνια, έχει χρησιμοποιηθεί και σε τηλεοράσεις, ρολόγια, ακόμα και αυτοκίνητα. Το λογότυπο του Android φαίνεται στην παρακάτω εικόνα.



Εικόνα 1: Android logo

2.2 Σύγκριση του Android με άλλα λειτουργικά συστήματα για smartphones

Αυτή τη στιγμή στην παγκόσμια αγορά χρησιμοποιούνται κατά κύριο λόγο τα εξής λειτουργικά συστήματα στα smartphones: Android (της Google), iOS (της Apple), Windows Phone (της Microsoft), BlackBerry® (της RIM) και Symbian (της Nokia). Τη σύγκριση του λειτουργικού Android με τα υπόλοιπα λειτουργικά, θα μπορούσαμε να τη διαχωρίσουμε σε δύο σκέλη. Το πρώτο σκέλος αφορά τη λειτουργικότητα που απολαμβάνει ο χρήστης, σε συνδυασμό με την ευκολία χρήσης αλλά και απόκτησης της επιθυμητής συσκευής που φέρει κάποιο από τα παραπάνω λειτουργικά. Το δεύτερο σκέλος αφορά την ευκολία συγγραφής κώδικα και χρήσης εργαλείων που προσφέρονται από τα λειτουργικά αυτά για την ανάπτυξη εφαρμογής για smartphone.

2.2.1 Σύγκριση του Android ως προς τη λειτουργικότητά του

Ας ξεκινήσουμε με το πρώτο σκέλος. Το Android προσφέρει μεγάλο αριθμό διαθέσιμων εφαρμογών (πάνω από 700.000 εφαρμογές, δωρεάν ή με πληρωμή), και ο αριθμός αυτών των εφαρμογών αυξάνεται καθημερινά. Είναι εύκολο στη χρήση του, καθώς και στην προσωποποίηση ώστε να το ρυθμίσει ο χρήστης όπως τον βολεύει. Σαν πλατφόρμα είναι η πιο γρήγορα αναπτυσσόμενη σήμερα και χρησιμοποιείται από πολλούς κατασκευαστές κινητών τηλεφώνων, ενώ στην αγορά υπάρχουν πολλές διαθέσιμες συσκευές σε όλες τις κατηγορίες τιμών. Υπάρχει μια Android συσκευή για όλα τα γούστα και τις οικονομικές δυνατότητες. Τέλος, λόγω του ότι η ανάπτυξη του λειτουργικού αυτού γίνεται από την Google, φυσικά συγχρονίζεται εύκολα με τις δημοφιλείς υπηρεσίες τις Google, όπως το Gmail, Contacts και Google Maps κλπ.

Το iOS προσφέρει εξίσου μεγάλο αριθμό διαθέσιμων εφαρμογών (εκατοντάδες χιλιάδες εφαρμογές διαθέσιμες στο App Store της Apple), είναι πολύ εύκολο στο χειρισμό, ωστόσο είναι διαθέσιμο μόνο στις συσκευές της Apple. Η συσκευή τηλεφώνου (iPhone) μπορεί να χρησιμοποιηθεί σαν iPod για μουσική συγχρονίζοντας με το iTunes. Επίσης, υπάρχει η δυνατότητα κλήσεων βίντεο (μέσω του FaceTime) και δυνατότητα πολυδιεργασίας ώστε να χρησιμοποιούν περισσότεροι από ένας χρήστες το ίδιο σύστημα / παιχνίδι, αλλά τα παραπάνω ισχύουν μόνο μεταξύ iPhone συσκευών. Σήμερα το Android θεωρείται σε παγκόσμιο επίπεδο το ίδιο δημοφιλές με το iOS της Apple.

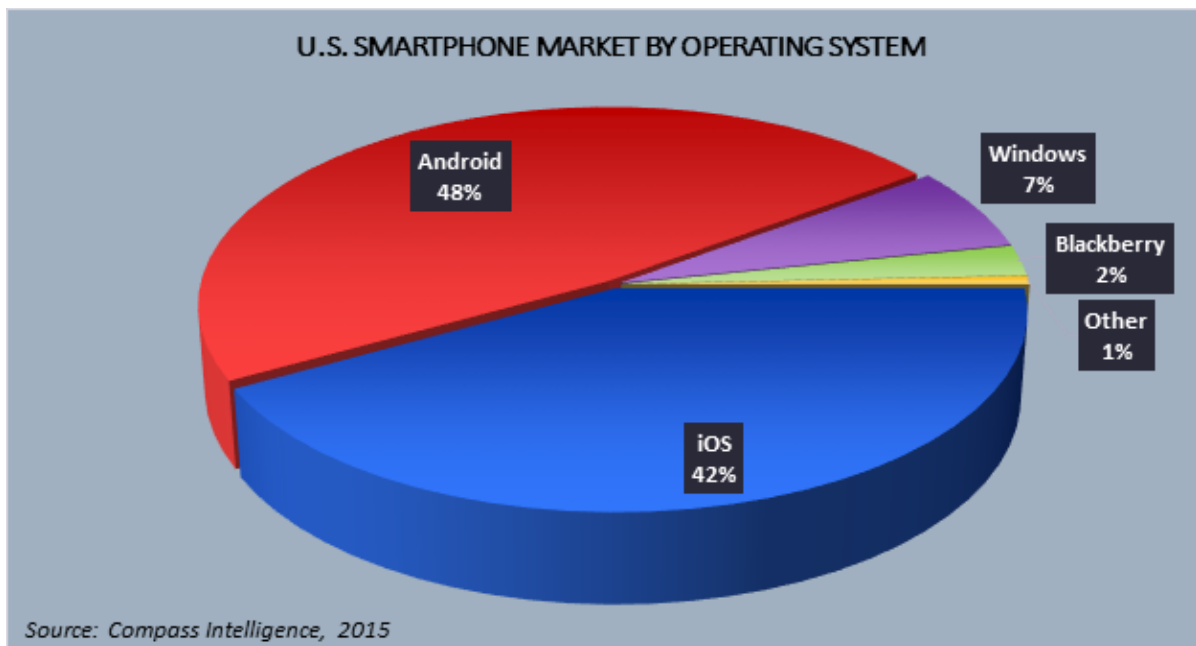


Εικόνα 2: Android vs iOS

Το Windows Phone δίνει έμφαση στον απλούστερο δυνατό χειρισμό της συσκευής και στη σύνδεση με ιστότοπους κοινωνικού περιεχομένου. Τα Windows Phone Smartphones έχουν διασύνδεση με το Zune της Microsoft, Facebook, Windows LIVE καθώς και το Windows Phone Marketplace για εφαρμογές. Κυριότερο χαρακτηριστικό είναι η χρήση του Microsoft Office: Microsoft Office Mobile με Word, Excel και Powerpoint, ώστε να μπορεί ο χρήστης να δει και να επεξεργαστεί Microsoft Office αρχεία όπου κι αν βρίσκεται. Τέλος, διαθέτουν Office hub για διασύνδεση της συσκευής και της αντίστοιχης έκδοσης Microsoft Office στον υπολογιστή.

Η πλατφόρμα BlackBerry® είναι διάσημη κυρίως για την παροχή υψηλής ποιότητας υπηρεσιών email και instant messaging μεταξύ BlackBerry συσκευών, το οποίο σημαίνει ότι είναι διαθέσιμη μόνο σε συσκευές BlackBerry®. Επίσης, πλέον προσφέρει εξαιρετικές υπηρεσίες πολυμέσων και περισσότερες δυνατότητες προσωποποίησης της συσκευής. Οι BlackBerry® συσκευές διαθέτουν πλήρες πληκτρολόγιο (QWERTY διάταξη όπως και στους υπολογιστές), είτε στην οθόνη αφής είτε με πλήκτρα για πιο γρήγορη και άνετη πληκτρολόγηση μηνυμάτων και emails. Τέλος, διαθέτει πάνω από 5.000 εφαρμογές, που διαρκώς μεγαλώνουν σε αριθμό.

Τέλος, το Symbian προσφέρει πολύ καλές δυνατότητες σε πολυμέσα (εγγραφή βίντεο HD), έχει πάνω από 10.000 εφαρμογές και παρέχει δωρεάν δορυφορική πλοήγηση με τα Oni Maps. Ωστόσο, το Symbian χρησιμοποιείται στις περισσότερες συσκευές Nokia και σε ελάχιστες των υπολοίπων κατασκευαστών.



Εικόνα 3: Ανταγωνισμός μεταξύ Android και λοιπών λειτουργικών (2015)

Κλείνοντας το πρώτο σκέλος, μπορούμε να συμπεράνουμε ότι ο μεγαλύτερος ανταγωνισμός έγκειται μεταξύ του Android και του iOS, καθώς τα υπόλοιπα λειτουργικά εστιάζουν σε μικρότερο εύρος επιλογών. Οι Android συσκευές (όπως αποκαλούνται πλέον οι συσκευές που φέρουν λειτουργικό Android), είναι πιο εύκολο να χρησιμοποιηθούν, αλλά και να αποκτηθούν. Επίσης, προσφέρουν όλων των ειδών εφαρμογές και δεν υπάρχει η δέσμευση του κατασκευαστή, καθώς όπως αναφέραμε οι Android συσκευές προσφέρονται από πολλούς κατασκευαστές. Η προτίμηση στις συσκευές Android απεικονίζεται και στην προηγούμενη εικόνα.

2.2.2 Σύγκριση του Android ως προς την ευκολία ανάπτυξης εφαρμογής

Ας περάσουμε τώρα στο δεύτερο σκέλος. Ο προγραμματισμός και η υλοποίηση εφαρμογών σε κινητά τηλέφωνα είναι ένας κλάδος ο οποίος δείχνει να αναβαθμίζεται όλο και περισσότερο με το πέρασ των χρόνων. Αυτό που κάνει το Android περισσότερο αποδεκτό στη δημιουργία εφαρμογών είναι ότι είναι ανοιχτού κώδικα (Open Source Software). Αποτέλεσμα αυτού είναι ότι μπορείς οποιαδήποτε στιγμή να ξεκινήσεις να προγραμματίζεις και να δημιουργείς Android εφαρμογές δωρεάν με τη βοήθεια του Android SDK (Software Development Kit). Για να ξεκινήσεις τον προγραμματισμό σε Android, πρέπει απλά να κατεβάσεις το Eclipse (το οποίο χρησιμοποίησα και στην εφαρμογή μου), ή το Android Studio. Το Eclipse και το Android SDK τρέχουν σε όλα τα λειτουργικά συστήματα και τις εκδόσεις (Windows, Linux, Mac).



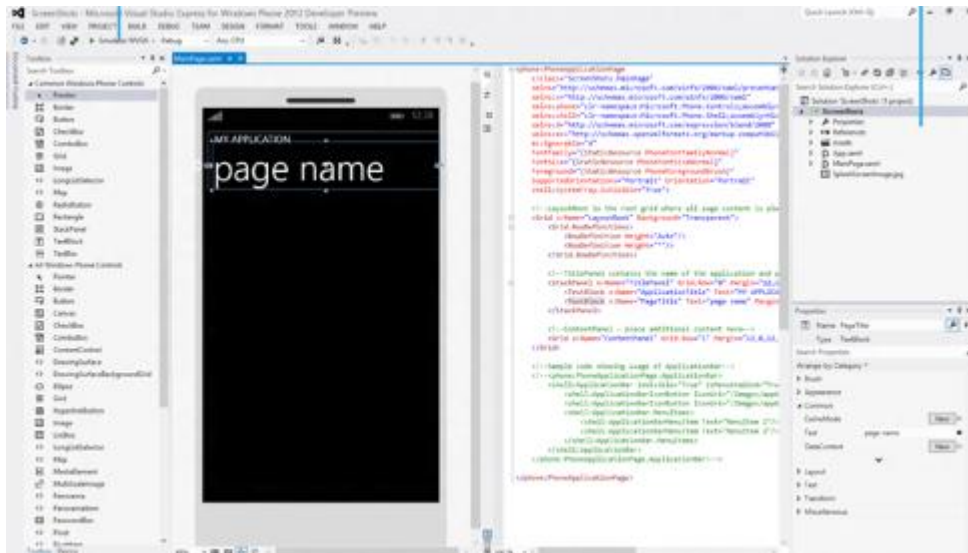
Εικόνα 4: Προγραμματισμός Android στο Eclipse

Η Apple με το λειτουργικό της σύστημα (iOS) που είναι κλειστού τύπου, καθιστά την ανάπτυξη εφαρμογών δυσκολότερη και πιο ακριβή. Το εργαλείο που χρησιμοποιείται για την ανάπτυξη εφαρμογής για iPhone είναι το Xcode, (επίσημη IDE - Integrated Development Environment για iOS και Mac περιβάλλοντα), από το App Store της Mac. Το Xcode τρέχει μόνο σε Mac OS, οπότε απαιτείται η χρήση υπολογιστή της εταιρίας (iMac).



Εικόνα 5: Προγραμματισμός iOS στο Xcode

Για το Windows Phone, το εργαλείο που χρησιμοποιείται είναι το Visual Studio Express, το οποίο τρέχει μόνο σε Windows.



Εικόνα 6: Προγραμματισμός Windows Phone στο Visual Studio Express

Σχετικά με το κόστος, για την πλήρη έκδοση του Visual Studio και αν θέλετε να δημοσιεύσετε μια εφαρμογή ή να αναπτύξετε εφαρμογή στη συσκευή σας, πρέπει να καταβάλετε ένα ετήσιο τέλος. Επίσης, μία ετήσια αμοιβή πρέπει να καταβληθεί και για τη δημιουργία ενός Windows Developer Account, για προγραμματιστές των Windows. Για το iOS απαιτείται η εγγραφή στο Apple Developer Program, όπου η συνδρομή κυμαίνεται σε υψηλότερα επίπεδα κόστους. Αντιθέτως, αν θέλετε να δημοσιεύσετε τις εφαρμογές σας στο κατάστημα Play Store (Android Market – κατάστημα εφαρμογών), το κόστος είναι σημαντικά μικρότερο.

Όλες οι πλατφόρμες έχουν πλούσια τεκμηρίωση, αλλά το Android έχει ένα πλεονέκτημα σε online κοινότητες (όπως το StackOverflow).

Ας λάβουμε υπόψιν στη σύγκρισή μας και πιο τεχνικά στοιχεία. Η Google έλαβε την απόφαση να χρησιμοποιήσει Java ως γλώσσα προγραμματισμού του Android, για να προσελκύσει πολλούς προγραμματιστές Java. Η Microsoft έκανε ακριβώς το ίδιο, μόνο που χρησιμοποιεί C#. Η C# είναι μια πιο σύγχρονη γλώσσα προγραμματισμού, που διαθέτει τα καλά στοιχεία της Java με περισσότερη λειτουργικότητα, αλλά και περιορίζει τις ατέλειές της. Ωστόσο, η ανάπτυξη εφαρμογών σε Windows Phone είναι περιοριστική και επίπονη: δεν προσφέρει λειτουργικότητες του Android όπως «εμπλουτισμένες» ειδοποιήσεις (όπως οι ειδοποιήσεις της έκδοσης Jelly Bean του Android), widgets (γραφικά στοιχεία που δείχνουν "ζωντανά" πληροφορίες που ενδιαφέρουν το χρήστη), live wallpapers, αληθινό multitasking, ραδιοφωνικούς δέκτες. Κύρια γλώσσα για το iOS είναι η Objective-C.

Ακόμη ένας λόγος που κάνει την ανάπτυξη εφαρμογών Android πιο εύκολη και προσβάσιμη, είναι η ευκολία να χρησιμοποιήσεις την εφαρμογή σου σε οποιαδήποτε συσκευή διαθέτει λειτουργικό Android, ανεξαρτήτου μάρκας, ενώ για εφαρμογή σε iOS πρέπει οπωσδήποτε να έχεις iPhone.

Τέλος, για να δοκιμαστεί η εφαρμογή δε χρειάζεται αυτή να ανέβει πρώτα στο Play Store. Μόλις αναπτύξετε το application (app), αυτό μπορεί να δημοσιευθεί με τη χρήση του Google Play Developer Console. Για το Windows Phone, αν θέλετε να δοκιμάσετε το app στο τηλέφωνό σας, πρέπει να το δηλώσετε ως συσκευή δοκιμής, αν και υπάρχουν εξομοιωτές τηλεφώνου που μπορείτε επίσης να χρησιμοποιήσετε. Μόλις ολοκληρωθούν οι δοκιμές, μπορεί να δημοσιευθεί στο κατάστημα στο Windows Phone Store. Αντίθετα, στο λειτουργικό iOS αφού αναπτύξετε και δοκιμάσετε την εφαρμογή σας, πρέπει αυτή να υποβληθεί για έλεγχο από την Apple, πριν τη συμπεριλάβει στο App Store.

Λαμβάνοντας υπόψιν όλα τα σημεία της σύγκρισης που προηγήθηκε, το Android επιλέχτηκε στην πτυχιακή αυτή εργασία διότι είναι μια πλατφόρμα η οποία μπορεί πιο εύκολα και γρήγορα να υποστηρίξει το σχεδιασμό και την υλοποίηση της εφαρμογής μου.

2.3 Το Android και η αρχιτεκτονική του

Η αρχιτεκτονική του Android είναι τέτοια που ενθαρρύνει την επαναχρησιμοποίηση κώδικα και επιτρέπει να δημοσιευτούν και να μοιραστούν δραστηριότητες, υπηρεσίες και δεδομένα με άλλες εφαρμογές. Λόγω αυτού, καθίσταται δυνατή η επέκταση και βελτιστοποίηση εφαρμογών, καθώς και η δημιουργία καινούργιων.

2.3.1 Διάγραμμα της αρχιτεκτονικής Android



Εικόνα 7: Διάγραμμα αρχιτεκτονικής Android

2.3.2 Επίπεδα της αρχιτεκτονικής Android

Όπως φαίνεται και στο διάγραμμα της προηγούμενης παραγράφου, το Android χωρίζεται σε τέσσερα επίπεδα:

1. Linux Kernel (Πυρήνας του Linux):

Αυτός είναι ο πυρήνας στον οποίο βασίζεται το Android και βρίσκεται στο χαμηλότερο επίπεδο. Πρόκειται για τον πυρήνα Linux έκδοση 2.6 για βασικές υπηρεσίες συστήματος όπως ασφάλεια, διαχείριση μνήμης, διαχείριση διεργασιών, στοίβα δικτύου, και οδηγούς συσκευών. Ο πυρήνας λειτουργεί επίσης ως ένα ενδιάμεσο επίπεδο αφαίρεσης μεταξύ της στοίβας λογισμικού και του υλικού.

2. Libraries (Βιβλιοθήκες):

Οι βιβλιοθήκες (γραμμένες σε C/C++) περιέχουν όλο τον κώδικα του Android OS και χρησιμοποιούνται από διάφορα στοιχεία του συστήματος. Οι δυνατότητες που προσφέρουν αυτές οι

βιβλιοθήκες είναι προσβάσιμες στους προγραμματιστές δια μέσου του επιπέδου πλαισίου εφαρμογής.

Για παράδειγμα, η SQLite βιβλιοθήκη παρέχει υποστήριξη ώστε μια εφαρμογή να χρησιμοποιήσει την αποθήκευση δεδομένων, η Webkit βιβλιοθήκη παρέχει λειτουργίες για το διαδικτυακό σερφάρισμα, κλπ.

3. Android Runtime (Επίπεδο Εκτέλεσης):

Στο ίδιο επίπεδο με τις βιβλιοθήκες, το Android Runtime παρέχει ένα σύνολο βασικών βιβλιοθηκών που επιτρέπουν στους προγραμματιστές να γράψουν εφαρμογές χρησιμοποιώντας Java. Επίσης, περιλαμβάνει την Dalvik Virtual Machine, που επιτρέπει κάθε εφαρμογή να τρέξει τη δική της εργασία, με τη δική της ξεχωριστή Dalvik Virtual Machine. (Η Dalvik είναι μια εξειδικευμένη Virtual Machine, ειδικά διαμορφωμένη για κινητές συσκευές που έχουν περιορισμένη μνήμη και ισχύ).

4. Application Framework (Πλαίσιο Εφαρμογών):

Παρέχοντας μια ανοικτή πλατφόρμα ανάπτυξης, το Android προσφέρει στους προγραμματιστές τη δυνατότητα να κατασκευάσουν πλούσιες και καινοτόμες εφαρμογές. Οι προγραμματιστές δύνανται να εκμεταλλευτούν πλήρως το hardware της συσκευής, να έχουν πρόσβαση σε υπηρεσίες εντοπισμού θέσης, να τρέξουν υπηρεσίες στο background, να θέσουν χρονοδιακόπτες για εμφάνιση ειδοποιήσεων και πολλά άλλα. Επίσης έχουν πλήρη πρόσβαση στο ίδιο πλαίσιο από APIs (Application Programming Interface) που έχουν οι βασικές εφαρμογές του Android. Η αρχιτεκτονική είναι διαμορφωμένη με τέτοιο τρόπο που κάθε εφαρμογή μπορεί να χρησιμοποιήσει τις δυνατότητες μιας άλλης και επίσης με τρόπο που δίνει την δυνατότητα στον χρήστη να αλλάξει τα συστατικά κάθε εφαρμογής. Κάτω από το πλαίσιο των εφαρμογών υπάρχει ένα σύστημα από υπηρεσίες και συστήματα που συγκροτούν τη βάση της αρχιτεκτονικής των Android apps και απαρτίζονται ως εξής:

- **Activity Manager:** Διαχειριστή δραστηριοτήτων ο οποίος διαχειρίζεται τον κύκλο ζωής (life cycle) των εφαρμογών.
- **Views:** Ένα σύνολο από γραφικά στοιχεία για τη δημιουργία γραφικού περιβάλλοντος (User Interface - UI). συμπεριλαμβανομένων λιστών (lists), πλεγμάτων (grids), κουτιών κειμένου (text boxes), κουμπιών (buttons) και άλλων.
- **Notification Manager:** Διαχειριστής ειδοποιήσεων ο οποίος επιτρέπει την προβολή ειδοποιήσεων στην μπάρα κατάστασης (status bar).
- **Content Manager:** Διαχειριστής περιεχομένου ο οποίος επιτρέπει στις εφαρμογές την πρόσβαση σε δεδομένα άλλων εφαρμογών ή τον διαμοιρασμό των δικών τους δεδομένων με άλλες εφαρμογές.
- **Resource Manager:** Διαχειριστής πόρων για την πρόσβαση στους πόρους, όπως strings, εικόνες, layout files. Υποστηρίζει συμβολοσειρές και γραφικά.
- **Applications (Εφαρμογές):**
Το υψηλότερο επίπεδο: εδώ βρίσκουμε εφαρμογές που έρχονται μαζί με την Android συσκευή του χρήστη (όπως τηλέφωνο, επαφές, μουσική, κλπ.), όπως επίσης εφαρμογές που κάνει εγκατάσταση. Οποιαδήποτε εφαρμογή έχει είναι σε αυτό το επίπεδο.



Εικόνα 8: Android OS Architecture

2.4 Εκδόσεις Android

Το 2008 βγήκε το πρώτο Android smartphone. Μέχρι σήμερα, έχουν βγει αρκετές εκδόσεις, οι οποίες παίρνουν το όνομά τους από γλυκά και επιδόρπια. Επίσης, η ονομασία των εκδόσεων ακολουθεί αλφαβητική σειρά, πχ Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, κλπ. Στην επόμενη εικόνα βλέπετε τις κυριότερες εκδόσεις Android.



Εικόνα 9: Android Ιστορικό Εκδόσεων

Δείτε μία συνοπτική παρουσίαση των εκδόσεων:

- Έκδοση Android™ 1.5: Cupcake
- Έκδοση Android™ 1.6: Donut
- Έκδοση Android™ 2.0, 2.1: Éclair
- Έκδοση Android™ 2.2: Frozen Yogurt (Froyo)

Έκδοση Android™ 2.3: Gingerbread
Έκδοση Android™ 3.0, 3.1, 3.2: Honeycomb
Έκδοση Android™ 4.0: Ice Cream Sandwich
Έκδοση Android™ 4.1, 4.2, 4.3: Jelly Bean
Έκδοση Android™ 4.4: KitKat
Έκδοση Android™ 5.0: Lollipop

Ακολουθούν τα χαρακτηριστικά της εκάστοτε έκδοσης βάσει χρονολογικής σειράς.

2.4.1 Android 1.0 Apple Pie



Εικόνα 10: Android Apple Pie

Η έκδοση 1.0 πρωτοκυκλοφόρησε το 2008 με το πρώτο Android smartphone HTC Dream.

2.4.2 Android 1.1 Banana Bread



Εικόνα 11: Android Banana Bread

Η έκδοση 1.1 κυκλοφόρησε το Φεβρουάριο του 2009. Για τα τότε δεδομένα, η έκδοση 1.1 ήταν ένα σημαντικό γεγονός. Έδειχνε ότι το Android επιδεχόταν αναβαθμίσεις, (κάτι το οποίο άλλες πλατφόρμες δεν υποστήριζαν ακόμη), και ότι η διαδικασία της αναβάθμισης ήταν εύκολη για το χρήστη.

2.4.3 Android 1.5 Cupcake



Εικόνα 12: Android Cupcake

Η έκδοση 1.5 κυκλοφόρησε στις 30 Απριλίου του 2009. Αποτελείται από πολλές βελτιώσεις:

- Ταχύτερη εκκίνηση κάμερας και λήψη εικόνας
- Πολύ ταχύτερη απόκτηση θέσης GPS
- Ομαλότερη κύλιση της σελίδας στο πρόγραμμα περιήγησης
- Και από πολλά νέα χαρακτηριστικά:
- Λειτουργία αυτόματης συμπλήρωσης κειμένου και δυνατότητα προσανατολισμού του πληκτρολογίου στην οθόνη τόσο σε πορτρέτο όσο και σε τοπίο.
- Βιντεοσκόπηση και αναπαραγωγή βίντεο (MPEG-4 και 3GP μορφές)
- Υποστήριξη Stereo Bluetooth και βελτιωμένη εμπειρία handsfree.
- Ανάπτυξη των widgets της αρχικής οθόνης με εφέ αλλαγής οθονών και μενού, τα οποία περιλαμβάνουν αναλογικό ρολόι, ημερολόγιο, music player, κορνίζα, και αναζήτηση. Επίσης, φάκελοι μπορούν να τοποθετηθούν στην επιφάνεια εργασίας.
- Ενημερωμένο πρόγραμμα περιήγησης με λειτουργία αντιγραφής/επικόλλησης δικτυακών διευθύνσεων, δυνατότητα αναζήτησης μέσα σε μία σελίδα, και προβολής καρτελών με σελιδοδείκτες, ιστορικό περιήγησης και πιο συχνά χρησιμοποιούμενες σελίδες.
- Εμφάνιση εικόνας χρήστη στις αγαπημένες επαφές.
- Καταγραφή ειδικής «σφραγίδας» ημερομηνίας / ώρας για τα γεγονότα στο αρχείο καταγραφής κλήσεων.
- Πρόσβαση με ένα άγγιγμα σε μια κάρτα επαφής από συμβάν καταγραφής κλήσεων.
- Νέα έκδοση (2.6.27) του Πυρήνα του Linux.
- Αυτόματος έλεγχος και επισκευή της SD κάρτας αρχείων.
- Εφαρμογή SIM Toolkit 1.0.
- Βελτιώσεις σε εφαρμογές Google (π.χ. Προβολή της κατάστασης του Google Talk φίλων στις Επαφές, ανέβασμα βίντεο στο Youtube, ανέβασμα φωτογραφιών στο Picasa, κ.ά.).

2.4.4 Android 1.6 Donut



Εικόνα 13: Android Donut

Κυκλοφόρησε στις 15 Σεπτεμβρίου του 2009 και περιλάμβανε πολλές βελτιώσεις, όπως:

- Πλαίσιο γρήγορης αναζήτησης (Quick Search Box).
- Οι χρήστες μπορούν να εναλλάσσουν γρήγορα μεταξύ φωτογραφιών και καταγραφής βίντεο. Επιπλέον, ο φάκελος αποθήκευσης των φωτογραφιών επιτρέπει στον χρήστη να επιλέξει πολλές φωτογραφίες για διαγραφή. Παρέχει επίσης μια πολύ πιο γρήγορη εμπειρία φωτογραφικής μηχανής: σε σύγκριση με την προηγούμενη έκδοση, η κάμερα είναι 39% πιο γρήγορη, και υπάρχει μια βελτίωση 28% στο χρονικό διάστημα από την ολοκλήρωση μιας λήψης στην επόμενη.
- Ένα νέο Virtual Private Network (VPN) επιτρέπει στους χρήστες να συνδέονται με διάφορους τύπους VPNs.
- Ένδειξη χρήσης μπαταρίας: μια νέα οθόνη χρήσης της μπαταρίας επιτρέπει στους χρήστες να δουν ποιες εφαρμογές και υπηρεσίες καταναλώνουν ενέργεια της μπαταρίας. Αν ο χρήστης εντοπίσει ότι μια συγκεκριμένη υπηρεσία ή εφαρμογή χρησιμοποιεί πάρα πολλή ισχύ, μπορεί να σταματήσει την αίτηση ή να απεγκαταστήσει την εφαρμογή.

2.4.5 Android 2.0-2.1 Eclair



Εικόνα 14: Android Éclair

Κυκλοφόρησε μόλις ένα μήνα μετά το Donut (26 Οκτωβρίου του 2009) και περιλάμβανε αρκετά νέα χαρακτηριστικά, καθώς και βελτιώσεις:

- Πολλαπλοί λογαριασμοί μπορούν να προστεθούν σε μια συσκευή για το ηλεκτρονικό ταχυδρομείο και το συγχρονισμό επαφών.
- Άμεση πρόσβαση στις λειτουργίες πληροφόρησης και επικοινωνίας μιας επαφής. Για παράδειγμα, ένας χρήστης μπορεί να αξιοποιήσει μια φωτογραφία επαφής και να επιλέξει να την καλέσει, να στείλει SMS, e-mail.
- Συνδυασμένη προβολή εισερχομένων e-mail για την περιήγηση από πολλούς λογαριασμούς σε μία σελίδα.

- Λειτουργία αναζήτησης για όλα τα αποθηκευμένα μηνύματα SMS και MMS.
- Αυτόματη διαγραφή των παλαιότερων μηνυμάτων σε μια συνομιλία όταν με το πέρασ ενός ορισμένου χρονικού διαστήματος.
- Βελτιώσεις στη φωτογραφική μηχανή: ενσωματωμένη υποστήριξη flash, ψηφιακό ζουμ, λειτουργία σκηνής, ισορροπία λευκού, επίδραση χρώματος, macro εστίασης.
- Εικονικό πληκτρολόγιο: μια βελτιωμένη διάταξη πληκτρολογίου για πιο εύκολο πάτημα των σωστών χαρακτήρων και να βελτίωση της ταχύτητας πληκτρολόγησης. Με την multi-touch υποστήριξη του πλαισίου, εξασφαλίζεται ότι τα πατήματα των πλήκτρων δεν έχουν χαθεί κατά την πληκτρολόγηση γρήγορα με δύο δάχτυλα. Επίσης, το λεξικό μαθαίνει από τη χρήση λέξεων και αυτόματα περιλαμβάνει ονόματα επαφών ως προτάσεις.
- Ανανέωση του User Interface (UI) του προγράμματος περιήγησης που επιτρέπει στο χρήστη να αξιοποιήσει άμεσα την γραμμή διευθύνσεων για αναζήτηση και την πλοήγηση. Επίσης, χρησιμοποιούνται μικρογραφίες των ιστοσελίδων στους σελιδοδείκτες, υποστηρίζεται το ζουμ με διπλό άγγιγμα και υποστηρίζεται και η HTML5.
- Άπειρη κύλιση στην προβολή ατζέντας και δυνατή πρόσκληση σε εκδηλώσεις.

2.4.6 Android 2.2-2.2.3 Froyo



Εικόνα 15: Android Froyo

Κυκλοφόρησε στις 20 Μαΐου του 2010 και είναι συντόμευση της φράσης “Frozen Yogurt” (παγωμένο γιαούρτι). Είναι η πρώτη έκδοση του Android που υποστήριζε Adobe Flash. Μεταξύ άλλων, περιλαμβάνει τα παρακάτω χαρακτηριστικά:

- Νέο widget με συμβουλές για την αρχική οθόνη που βοηθά τους νέους χρήστες για το πώς να ρυθμίσουν την αρχική οθόνη με τις συντομεύσεις και τα widgets, και πώς να κάνουν χρήση πολλαπλών οθονών.
- Το «τηλέφωνο», οι εφαρμογές εκκίνησης, και το πρόγραμμα περιήγησης έχουν πλέον σταθερές συντομεύσεις στην αρχική οθόνη, καθιστώντας εύκολη την πρόσβαση σε αυτά από οποιαδήποτε από τις 5 πολλαπλές οθόνες.
- Αυξημένη ασφάλεια με την προσθήκη αριθμητικού ή αλφαριθμητικού PIN για να το ξεκλείδωμα της συσκευής.
- Δυνατότητα επαναφοράς της συσκευής στις εργοστασιακές ρυθμίσεις μέσω απομακρυσμένης σύνδεσης. για να ασφαλίσετε τα δεδομένα στη συσκευή σε περίπτωση απώλειας ή κλοπής.
- Παγκόσμια λίστα διευθύνσεων διαθέσιμη στην εφαρμογή E-mail, επιτρέποντας στους χρήστες την αυτόματη συμπλήρωση ονομάτων των παραληπτών από τον κατάλογο.
- Στην οθόνη της κάμερας εισάγονται κουμπιά που παρέχουν εύκολη πρόσβαση σε ένα νέο UI, ελέγχουν το ζουμ, το φλας, την ισορροπία λευκού, την εστίαση και την έκθεση. Η βιντεοκάμερα παρέχει επίσης έναν εύκολο τρόπο για να ρυθμίσετε το μέγεθος βίντεο / ποιότητα για MMS και το YouTube, ενώ με το LED φλας μπορεί να γίνει λήψη φωτογραφιών τη νύχτα ή σε συνθήκες χαμηλού φωτισμού.

- Φορητό hotspot: ορισμένες συσκευές, όπως το Nexus One μπορεί να μετατραπεί σε ένα φορητό σημείο πρόσβασης Wi-Fi, η οποία μπορεί να μοιραστεί με έως και 8 συσκευές.
- Μπορείτε να χρησιμοποιήσετε το τηλέφωνό σας ως μια σύνδεση 3G για Windows ή Linux φορητό υπολογιστή με σύνδεση του τηλεφώνου με τον υπολογιστή με ένα καλώδιο USB. Η σύνδεση στη συνέχεια διαμοιράζεται μεταξύ των δύο συσκευών.
- Πολλαπλές γλώσσες πληκτρολογίου: πολύγλωσσοι χρήστες μπορούν να προσθέσουν πολλές γλώσσες στο πληκτρολόγιο και να εναλλάσσονται μεταξύ πολλών γλωσσών εισόδου που βασίζονται στα λατινικά, περνώντας σε όλη την μπάρα διαστήματος.
- Οι επιδόσεις του προγράμματος περιήγησης έχει ενισχυθεί και επιτρέπει την ταχύτερη φόρτωση της JavaScript (βαριές σελίδες).
- Επιτάχυνση των επιδόσεων της CPU (Central Processing Unit – Κεντρική Μονάδα Επεξεργασίας).
- Βελτιωμένη μνήμη, η οποία οδηγεί σε γρηγορότερη εναλλαγή.

2.4.7 Android 2.3-2.3.7 Gingerbread



Εικόνα 16: Android Gingerbread

Κυκλοφόρησε στις 9 Φεβρουαρίου του 2011. Ήταν πολύ πιο γρήγορο και εύχρηστο από τις προηγούμενες εκδόσεις και έδινε στους δημιουργούς εφαρμογών μεγαλύτερες δυνατότητες. Ακολουθούν κάποια από τα χαρακτηριστικά του και βελτιώσεις του:

- Το user interface είναι βελτιωμένο με πολλούς τρόπους σε όλο το σύστημα, πράγμα που καθιστά ευκολότερη την εκμάθησή του και την πλοήγηση, καθώς και πιο γρήγορη τη χρήση. Επίσης, οι αλλαγές στις ρυθμίσεις διευκολύνουν τον έλεγχο των δυνατοτήτων του συστήματος και της συσκευής.
- Το πληκτρολόγιο έχει επανασχεδιαστεί και βελτιστοποιηθεί για ταχύτερη εισαγωγή κειμένου, που καθιστά ευκολότερο να πιέσετε με ακρίβεια, ακόμη και σε υψηλές ταχύτητες.
- Στο πληκτρολόγιο προστίθεται η δυνατότητα να διορθώσει λέξεις που εισάγονται με προτάσεις του λεξικού. Καθώς ο χρήστης επιλέγει μια λέξη που έχει ήδη εγγραφεί, το πληκτρολόγιο εμφανίζει προτεινόμενες λέξεις που ο χρήστης μπορεί να επιλέξει, για να αντικαταστήσει την επιλογή. Ο χρήστης μπορεί επίσης να μεταβεί στη φωνητική λειτουργία εισόδου.
- Επιλογή λέξης με ένα άγγιγμα και copy / paste: κατά την εισαγωγή κειμένου ή την προβολή μιας ιστοσελίδας, ο χρήστης μπορεί να επιλέξει γρήγορα μια λέξη, στη συνέχεια να την αντιγράψει στο πρόχειρο και να κάνει επικόλληση.
- Βελτιωμένη διαχείριση ενέργειας: το σύστημα Android παίρνει έναν πιο ενεργό ρόλο στη διαχείριση των εφαρμογών που κρατούν τη συσκευή ξύπνια για μεγάλο χρονικό διάστημα ή που καταναλώνουν CPU ενώ λειτουργούν στο παρασκήνιο. Με τη διαχείριση αυτών των εφαρμογών το σύστημα βοηθά στο να διασφαλιστεί η καλύτερη δυνατή απόδοση και η μέγιστη διάρκεια ζωής της μπαταρίας.

- Με τη Διαχείριση Εφαρμογών, μια νέα καρτέλα «Running» εμφανίζεται στο χρήστη με μια λίστα με τις ενεργές εφαρμογές και τη μνήμη που χρησιμοποιείται από την καθεμία. Ο χρήστης μπορεί να διαβάσει περισσότερες λεπτομέρειες σχετικά με την κάθε αίτηση και εάν είναι απαραίτητο να σταματήσει μια εφαρμογή ή να αναφέρει σχόλια σχολίων στον προγραμματιστή της.
- Ο χρήστης μπορεί να κάνει φωνητικές κλήσεις μέσω του Διαδικτύου σε άλλους χρήστες που έχουν λογαριασμούς SIP.
- Η εφαρμογή μεταφόρτωσης δίνει στο χρήστη εύκολη πρόσβαση σε οποιοδήποτε αρχείο γίνεται λήψη από το πρόγραμμα περιήγησης, e-mail, ή άλλη εφαρμογή.

2.4.8 Android 3.0-3.2.6 Honeycomb



Εικόνα 17: Android Honeycomb

Κυκλοφόρησε στις 15 Ιουλίου του 2011 και ήταν διαθέσιμη μόνο για tablets. Η πρώτη συσκευή που έτρεχε σε Honeycomb ήταν το tablet Motorola Xoom. Το νέο UI φέρνει νέα πρότυπα για την αλληλεπίδραση, την πλοήγηση και την προσαρμογή, ενώ διατίθεται σε όλες τις εφαρμογές - ακόμη και εκείνες που κατασκευάστηκαν για παλαιότερες εκδόσεις της πλατφόρμας. Περιλαμβάνει:

- System Bar (μπάρα συστήματος): οι χρήστες έχουν γρήγορη πρόσβαση στις κοινοποιήσεις, την κατάσταση του συστήματος, και τα κουμπιά πλοήγησης με μία μπάρα συστήματος που διατίθεται στο κάτω μέρος της οθόνης.
- Action Bar (μπάρα δράσης): οι χρήστες έχουν πρόσβαση στις επιλογές, την πλοήγηση, τα widgets, ή άλλους τύπους περιεχομένου με μία μπάρα δράσης που εμφανίζεται στο πάνω μέρος της οθόνης. Η μπάρα δράσης είναι πάντα παρούσα όταν μια εφαρμογή είναι σε χρήση.
- Προσαρμόζομενη αρχική οθόνη: πέντε προσαρμόσιμες αρχικές οθόνες παρέχουν στους χρήστες άμεση πρόσβαση σε όλα τα μέρη του συστήματος από οποιαδήποτε οθόνη. Οι χρήστες μπορούν να επιλέξουν και να χειριστούν widgets, συντομεύσεις εφαρμογών, wallpapers και οπτική διάταξη. Οπτικά ερεθίσματα και σκιές βελτιώνουν την ορατότητα κατά τη ρύθμιση της διάταξης των συντομεύσεων και των widgets. Επίσης, προσφέρεται ένα πλαίσιο αναζήτησης για καθολική αναζήτηση εφαρμογών, επαφών, αρχείων πολυμέσων, περιεχόμενο ιστοσελίδων και πολλών άλλων.
- Εφαρμογές για πιο εύκολο multitasking.
- Επανασχεδιασμένο πληκτρολόγιο για να κάνει την εισαγωγή κειμένου γρήγορη και ακριβή για τα μεγαλύτερα μεγέθη οθόνης.
- Βελτιωμένη επιλογή κειμένου, αντιγραφή και επικόλληση.
- Νέες επιλογές συνδεσιμότητας: το πρωτόκολλο μεταφοράς δεδομένων / εικόνας επιτρέπει στους χρήστες να συγχρονίσετε άμεσα τα αρχεία πολυμέσων με μια φωτογραφική μηχανή ή έναν επιτραπέζιο υπολογιστή που συνδέεται μέσω USB. Οι χρήστες μπορούν επίσης να συνδέσουν πληκτρολόγιο πάνω στη συσκευή είτε με USB είτε με Bluetooth.

- Πολλαπλές καρτέλες αντικαθιστούν τα παράθυρα του προγράμματος περιήγησης και μια νέα λειτουργία «incognito» επιτρέπει την ανώνυμη περιήγηση. Οι σελιδοδείκτες και το ιστορικό παρουσιάζονται ενιαία.
- Η εφαρμογή της κάμερας έχει επανασχεδιαστεί ώστε να επωφεληθούν οι χρήστες από μια μεγαλύτερη οθόνη για γρήγορη πρόσβαση στην έκθεση, την εστίαση, το φλας, το ζουμ, την μπροστινή κάμερα, και άλλα. Προσθέτει επίσης ενσωματωμένη υποστήριξη για εγγραφή time-lapse video και προβολή φωτογραφιών και βίντεο σε λειτουργία πλήρους οθόνης.
- Βελτιωμένη μορφοποίηση των διεθνών αριθμών και παρουσίαση των στοιχείων επικοινωνίας σε μια κάρτα, καθιστώντας ευκολότερο για τους χρήστες να διαβάζουν και να επεξεργάζονται τις επαφές.

2.4.9 Android 4.0-4.0.4 Ice Cream Sandwich



Εικόνα 18: Android Cream Sandwich

Κυκλοφόρησε στις 16 Δεκεμβρίου του 2011 και έφερε πάρα πολλές αλλαγές στο λειτουργικό σύστημα. Μερικές από αυτές είναι:

- Η δυνατότητα χρήσης «μαλακών» κουμπιών δηλαδή κουμπιών πάνω στην οθόνη (πίσω, αρχική, κλπ.), ενώ μέχρι τότε όλα τα κινητά είχαν εξωτερικά κουμπιά.
- Βελτίωση δυνατοτήτων multitasking: το κουμπί με τις πρόσφατες εφαρμογές επιτρέπει στους χρήστες να πηδήξουν αμέσως από τη μία εργασία στην άλλη χρησιμοποιώντας τη λίστα στη μπάρα συστήματος.
- Νέοι φάκελοι στην αρχική οθόνη προσφέρουν ένα νέο τρόπο για τους χρήστες να ομαδοποιούν εφαρμογές και συντομεύσεις, απλά σύροντας τη μία στην άλλη. Επίσης, σε όλες τις εφαρμογές οι χρήστες μπορούν τώρα απλά να σύρουν την εφαρμογή για να αντλήσουν πληροφορίες σχετικά με αυτή ή να την απεγκαταστήσουν.
- Άλλες δυνατότητες ήταν το Face Unlock, καλύτερη χρήση των φωνητικών εντολών, αναδιαμόρφωση του περιβάλλοντος χρήσης, βελτίωση της ταχύτητας απόκρισης κλπ.

2.4.10 Android 4.1-4.3.1 Jelly Bean



Εικόνα 19: Android Jelly Bean

Κυκλοφόρησε στις 9 Ιουλίου του 2012. Το περιβάλλον χρήσης και η απόκρισή του είναι πιο γρήγορα και καλοφτιαγμένα, ενώ περιλαμβάνει πάρα πολλές μικρές βελτιώσεις σε όλο το σύστημα, όπως για παράδειγμα στην χρήση φωνής (υπαγόρευση κειμένου) και στην κάμερα. Η πρώτη συσκευή που είχε το Jelly Bean ήταν το tablet Google Nexus 7 ενώ η έκδοση Android 4.2 πρωτοεμφανίστηκε στα Nexus 4 και Nexus 10 στις 13 Νοεμβρίου του 2012. Τέλος, η έκδοση 4.3 κυκλοφόρησε στις 24 Ιουλίου του 2013. Μερικές από τις δυνατότητες του Jelly Bean είναι:

- Παρέχει καλύτερες δυνατότητες γραφικών σε 2D και 3D παιχνίδια και εφαρμογές, με τη βοήθεια του Khronos OpenGL ES 3.0.
- Σύνδεση του smartphone με άλλες συσκευές που χρησιμοποιούν Bluetooth, με αποτέλεσμα τη δημιουργία νέων εφαρμογών και υπηρεσιών που μπορούν να υλοποιηθούν μέσω αυτού.
- Δημιουργία πολλαπλών Profile σε tablet συσκευές, με αποτέλεσμα να μπορείς να ελέγχεις τις δυνατότητες που έχει ένας τρίτος χρήστης στη συσκευή σου.
- Οι υπηρεσίες του Google Play προσφέρουν προηγμένα API εντοπισμού που μπορούν να χρησιμοποιηθούν στις εφαρμογές σας. Το Android 4.3 βελτιστοποιεί τα API αυτά για τις υποστηριζόμενες συσκευές με νέο hardware και δυνατότητες software που ελαχιστοποιούν τη χρήση της μπαταρίας.

2.4.11 Android 4.4-4.4.4 KitKat



Εικόνα 20: Android KitKat

Κυκλοφόρησε στις 31 Οκτωβρίου του 2013 και έφερε τα πιο καινοτόμα, όμορφα και χρήσιμα χαρακτηριστικά του Android σε όλο και περισσότερες συσκευές σε όλο τον κόσμο. Η

έκδοση αυτή σχεδιάστηκε για να τρέχει πιο γρήγορα, ομαλά και υπεύθυνα σε ένα ευρύτερο φάσμα συσκευών από ποτέ, ακόμα και σε συσκευές με μόλις 512MB RAM. Δε σταμάτησε όμως εκεί καθώς φέρνει πολλά καινούργια χαρακτηριστικά, μερικά από τα οποία είναι τα εξής:

- Νέες NFC δυνατότητες μέσω Host Card Emulation (HCE), για πληρωμές, πρόσβαση σε κάρτες αλλά και άλλες υπηρεσίες.
- Εκτύπωση οποιουδήποτε περιεχομένου μέσω Wi-Fi ή υπηρεσίες αποθήκευσης cloud, απλά συνδέοντας το smartphone με έναν εκτυπωτή.
- Ένα νέο storage access framework, καθιστά απλό για τους χρήστες να βρουν και να ανοίξουν έγγραφα, εικόνες και άλλα αρχεία πιο γρήγορα, με το περιβάλλον χρήσης του χρήστη να είναι πολύ απλό και εύκολο.

2.4.12 Android 5.0-5.0.2 Lollipop



Εικόνα 21: Android Lollipop

Κυκλοφόρησε στις 3 Νοεμβρίου του 2014. Αυτή η έκδοση είναι γεμάτη με νέα χαρακτηριστικά για τους χρήστες και χιλιάδες νέα API για τους προγραμματιστές. Επεκτείνει το λειτουργικό ακόμη περισσότερο, από κινητά τηλέφωνα, tablets, ρολόγια, σε τηλεοράσεις και αυτοκίνητα. Μερικές από τις δυνατότητες που έχει είναι:

- Νέα 3D στοιχεία που ρίχνουν σκιές σε πραγματικό χρόνο, ακόμα και όταν αυτά κινούνται. Επίσης νέα animations φέρνουν με πολύ όμορφο τρόπο καινούργιες σελίδες και παράθυρα μπροστά στα μάτια του χρήστη.
- Το Android 5.0 τρέχει πλέον αποκλειστικά με το καινούργιο ART runtime, δημιουργημένο από την αρχή για να υποστηρίζει ένα μείγμα από ahead-of-time(AOT), just-in-time(JIT) και interpreted κώδικα. Υποστηρίζεται από ARM, x86 και MIPS αρχιτεκτονικές και είναι απολύτως συμβατό με 64-bit.
- Οι ενημερώσεις (notifications) είναι πλέον πιο ορατές, προσβάσιμες και παραμετροποιημένες.
- Με αυτή την έκδοση, το multitasking είναι ακόμα πιο φιλικό προς τον χρήστη καθώς υπάρχουν πολλές νέες δυνατότητες και χαρακτηριστικά που το πετυχαίνουν αυτό.
- Καλύτερα γραφικά, αφού υποστηρίζεται το Khronos OpenGL ES 3.1, καθώς επίσης και καλύτερο ήχο.

2.5 Το Android στην αγορά

Το Android είναι το πιο ευρέως διαδεδομένο λογισμικό στον κόσμο, με πωλήσεις που ξεπερνούν τις πωλήσεις σε συσκευές Windows, iOS και Mac OS μαζί.

Η πρώτη παρουσίαση του Android έγινε στις αρχές του Νοεμβρίου το 2007. Τότε ανακοινώθηκε και η ίδρυση του οργανισμού Open Handset Alliance (OHA). Μια κοινοπραξία 48 τηλεπικοινωνιακών εταιριών λογισμικού και κατασκευής hardware, οι οποίες αναπτύσσουν ανοιχτά πρότυπα στις συσκευές κινητής τηλεφωνίας. Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache License, η οποία είναι μια ελεύθερη άδεια λογισμικού. Μερικά από τα μέλη του οργανισμού αυτού (OHA) φαίνονται στην εικόνα που ακολουθεί.



Εικόνα 22: Εταιρίες ανάπτυξης λογισμικού και κατασκευής υλικού

Ο OHA ελπίζει να παραδώσει μια καλύτερη εμπειρία λογισμικού κινητών στους καταναλωτές, παρέχοντας καινοτόμες εφαρμογές κινητών, με υψηλότερη ποιότητα χωρίς τέλη αδείας για τους προγραμματιστές καθώς και τους κατασκευαστές κινητών τηλεφώνων.

3 Εργαλεία για δημιουργία Android εφαρμογής

Εδώ θα γίνει η περιγραφή του προγράμματος που χρησιμοποιήθηκε για τη δημιουργία της εφαρμογής, το οποίο είναι το Eclipse για τη συγγραφή του κώδικα.

Το Eclipse είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE). Περιέχει ένα βασικό χώρο εργασίας και ένα επεκτάσιμο plug-in για την προσαρμογή του περιβάλλοντος. Γραμμένο κυρίως σε Java, το Eclipse μπορεί να χρησιμοποιηθεί για την δημιουργία και ανάπτυξη εφαρμογών. Με τη βοήθεια των διάφορων plug-ins που έχει, το Eclipse μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών και σε άλλες γλώσσες προγραμματισμού όπως: Ada, ABAP, C, C++, COBOL, FORTRAN, JavaScript, Perl, PHP, Prolog, Python, Ruby και άλλες.

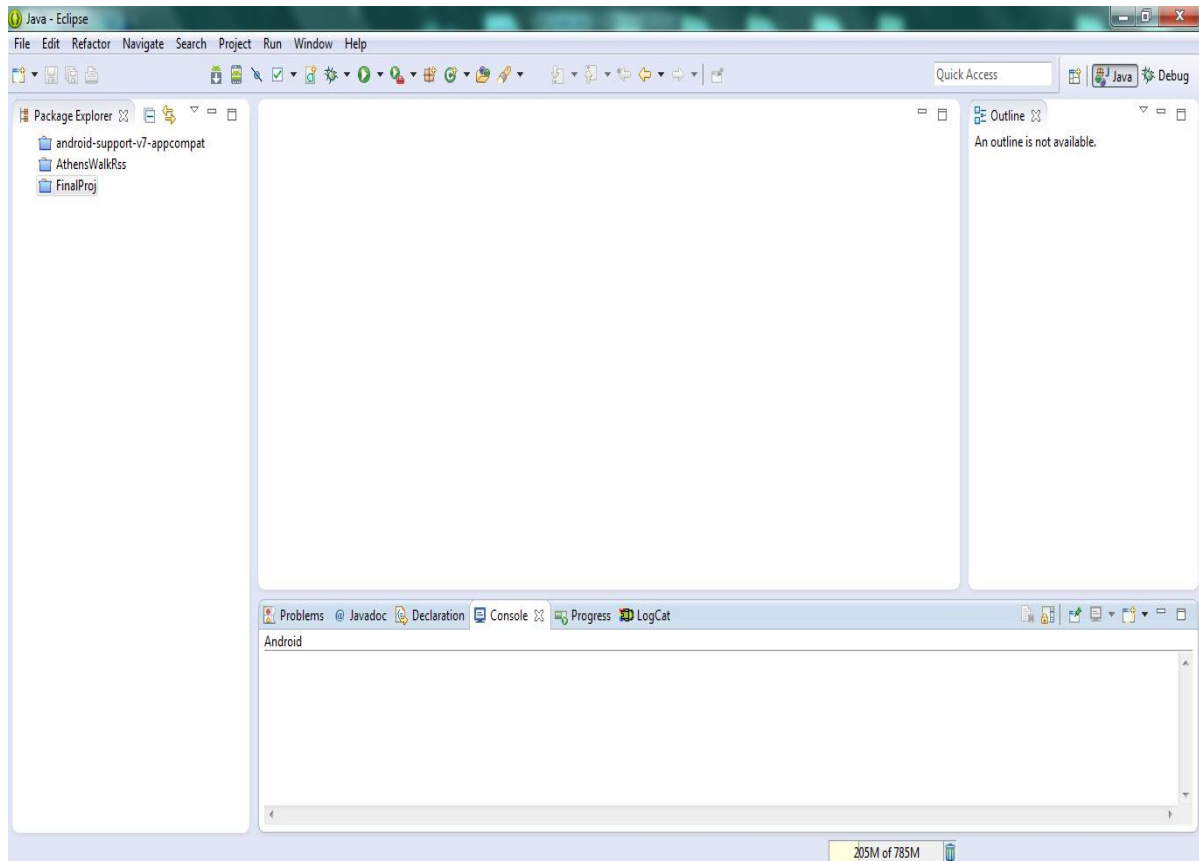
Το Eclipse software development kit (SDK), το οποίο περιλαμβάνει εργαλεία ανάπτυξης της Java, προορίζεται για προγραμματιστές Java. Οι χρήστες μπορούν να επεκτείνουν τις ικανότητές του με την εγκατάσταση plug-ins για την πλατφόρμα του Eclipse, όπως είναι τα εργαλεία για ανάπτυξη άλλων προγραμματιστικών γλωσσών.



Εικόνα 23: Λογότυπο του προγράμματος

3.1 Προγραμματιστικό περιβάλλον του Eclipse

Όπως είπαμε και προηγουμένως, το πρόγραμμα που χρησιμοποιήθηκε για την εφαρμογή είναι το Eclipse και η γλώσσα προγραμματισμού στην οποία υλοποιήθηκε η εφαρμογή είναι η Java. Επίσης το Eclipse είναι δωρεάν και μπορεί να το προμηθευτεί οποιοσδήποτε θέλει να ασχοληθεί με τον προγραμματισμό, από το διαδίκτυο.



Εικόνα 24: Αρχική οθόνη του Eclipse

Πέρα όμως από το πρόγραμμα αυτό, είναι απαραίτητο για την δημιουργία Android εφαρμογών το Android SDK, το οποίο συνδέεται με το γραφικό μας περιβάλλον μέσω ενός plug-in που έπρεπε να εγκατασταθεί, το ADT Plug-in, έτσι ώστε να μπορέσουμε να μεταγλωττίσουμε (compile) και να τρέξουμε το πρόγραμμά μας. Ένα ακόμη σημαντικό στοιχείο, είναι και ο προσομοιωτής κινητού τηλεφώνου (Android Virtual Device – AVD), που μας δίνει τη δυνατότητα να έχουμε στον υπολογιστή μας έναν προσομοιωτή τόσο σε υλικό, όσο και σε λογισμικό ενός κινητού τηλεφώνου με λειτουργικό Android.

Μέσα από το ADT Plug-in, μπορούμε να εγκαταστήσουμε τον προσομοιωτή και να ρυθμίσουμε και πολλές παραμέτρους του, όπως ποια έκδοση να χρησιμοποιεί, το μέγεθος της οθόνης, ακόμη και τη RAM της εικονικής συσκευής. Πέρα όμως από την προσομοίωση αυτή, μπορούμε να χρησιμοποιήσουμε και το κινητό μας τηλέφωνο εφόσον υπάρχει για να τρέξουμε μια εφαρμογή που δημιουργήσαμε.

Το Android SDK περιλαμβάνει επίσης και μερικά άλλα χρήσιμα εργαλεία για την ανάπτυξη των εφαρμογών, μερικά από τα οποία είναι:

Dalvik Debug Monitor Service (DDMS). Επιτρέπει τη διαχείριση των διεργασιών στον εξομοιωτή ή στη συσκευή, πιο συγκεκριμένα δηλαδή, δίνεται η δυνατότητα port-forwarding υπηρεσιών, η λήψη screenshots, εμφάνιση πληροφοριών threads, logcat και άλλα.

Android Debug Bridge (ADB). Επιτρέπει τη διαχείριση της κατάστασης του εξομοιωτή ή της συσκευής. Μέσω του ADB είναι δυνατή η εκτέλεση εντολών φλοιού, η διαχείριση της προώθησης θυρών, καθώς και η αντιγραφή από και προς τη συσκευή ή εξομοιωτή.

Android Asset Packaging Tool (AAPT). Επιτρέπει τη δημιουργία .apk αρχείων, που περιέχουν τα εκτελέσιμα αρχεία και τους πόρους μιας εφαρμογής.

Android Interface Description Language (AIDL). Επιτρέπει τη δημιουργία κώδικα που αφήνει δύο διεργασίες σε μια Android συσκευή να συνομιλούν, χρησιμοποιώντας διαδιεργασιακή επικοινωνία.

Sqlite3. Επιτρέπει την πρόσβαση στα δεδομένα της SQLite που δημιουργούνται από διάφορες εφαρμογές.

Traceview. Επιτρέπει τη γραφική προβολή της ανάλυσης των trace log data που δημιουργούν οι εφαρμογές.

Mksdcard. Βοηθά στη δημιουργία εικονικού δίσκου ο οποίος μπορεί να χρησιμοποιηθεί από τον εξομοιωτή για την προσομοίωση της παρουσίας εξωτερικής αποθηκευτικής κάρτας (SD card).

dx tool. Μετατρέπει τα αρχεία .class από java bytecode σε Android bytecode.

UI/Application Exerciser Monkey. Είναι ένα πρόγραμμα που τρέχει στον εξομοιωτή και παράγει ψευδο-τυχαίες σειρές από συμβάντα χρήστη όπως, clicks, gestures, touches, καθώς επίσης και έναν αριθμό από συμβάντα συστήματος.

Activitycreator. Είναι ένα script που δημιουργεί Ant build αρχεία τα οποία μπορούν να χρησιμοποιηθούν για τη μεταγλώττιση των εφαρμογών.

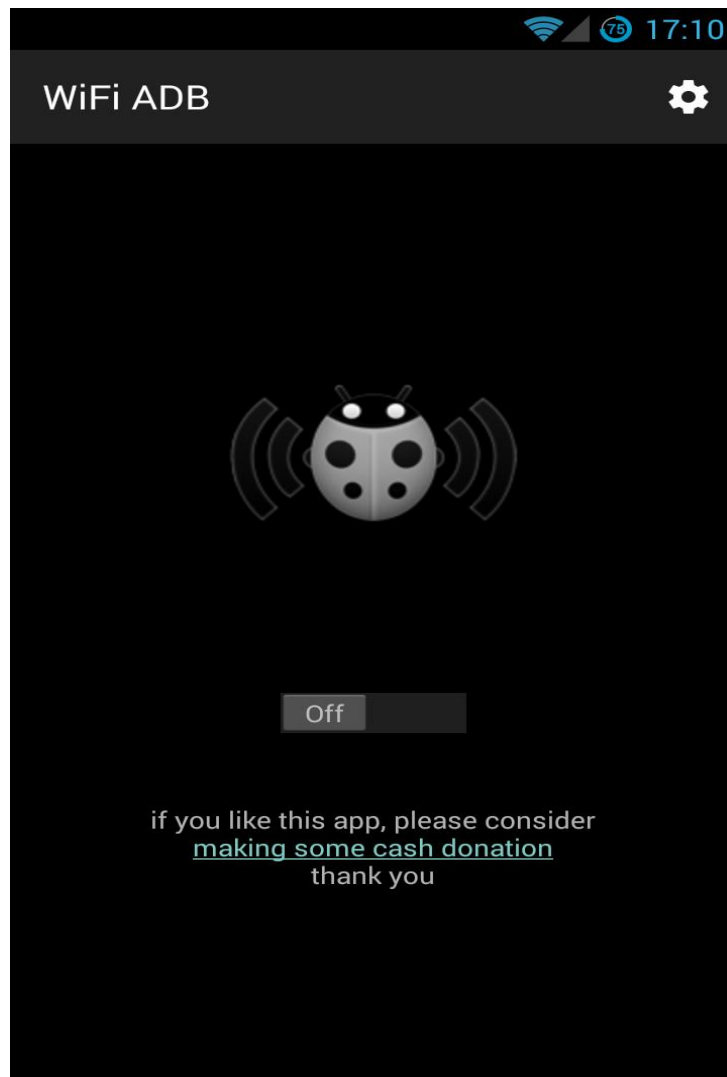
3.2 WiFi ADB

Για να μπορέσω να κάνω compile την εφαρμογή μου, δε χρησιμοποίησα το AVD (Android Virtual Device) που ανέφερα προηγουμένως λόγω κάποιων προβλημάτων που αντιμετώπισα. Χρησιμοποίησα όμως μια άλλη εφαρμογή, η οποία ονομάζεται WiFi ADB.



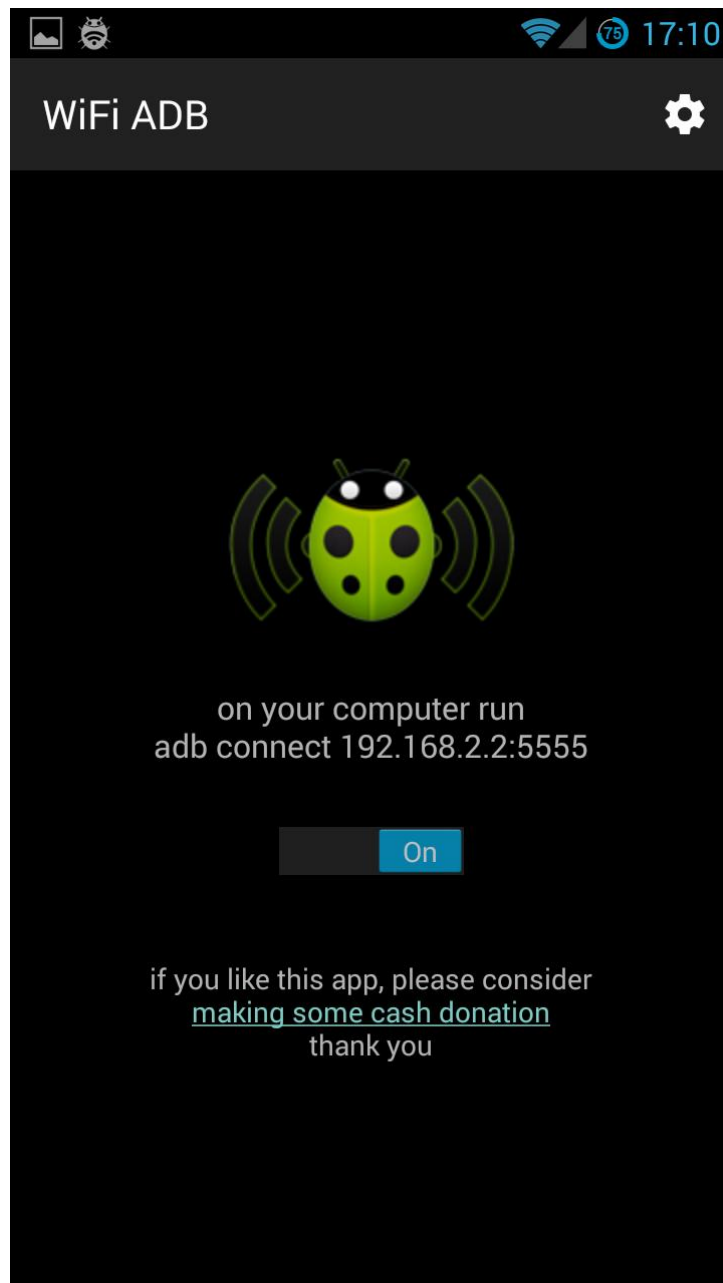
Εικόνα 25: Λογότυπο του WiFi ADB

Με την εφαρμογή αυτή, μπορούμε πολύ εύκολα και γρήγορα να κάνουμε compile το project που δημιουργούμε και να εγκατασταθεί κατευθείαν στο κινητό μας. Το περιβάλλον χρήστη είναι πολύ απλό και φαίνεται στην παρακάτω εικόνα.



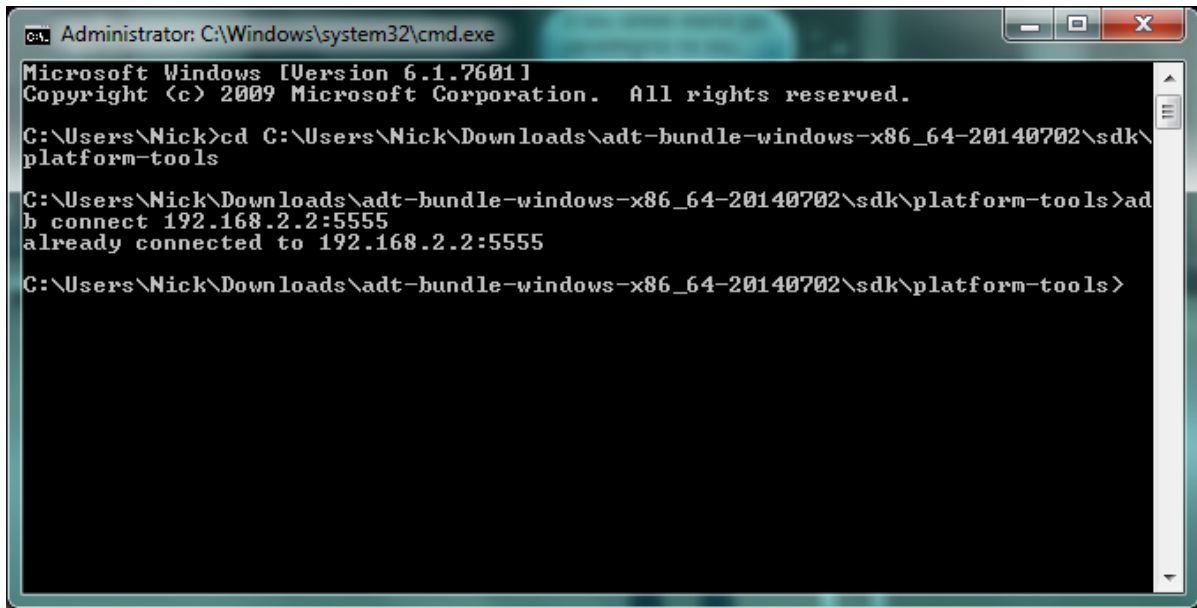
Εικόνα 26: Αρχική οθόνη του WiFi ADB

Στην αρχική οθόνη της εφαρμογής βλέπουμε το λογότυπο, ένα εικονίδιο για τις ρυθμίσεις και ένα κουμπί ON/OFF. Όταν πατήσουμε το κουμπί η οθόνη αλλάζει όπως φαίνεται παρακάτω.



Εικόνα 27: Αρχική οθόνη του WiFi ADB (ON)

Βλέπουμε πώς όταν το κουμπί είναι ON, μας εμφανίζει την IP με την οποία μπορούμε να συνδέσουμε το κινητό μας με το Eclipse. Για να γίνει αυτή η σύνδεση χρειάζεται να ανοίξουμε αρχικά το command prompt και να συνδεθούμε χρησιμοποιώντας την IP που μας δίνει η εφαρμογή. Αρχικά πρέπει να βρούμε όμως το αρχείο του Eclipse που ονομάζεται adb.exe, το οποίο βρίσκεται στους φάκελους του Eclipse. Αφού το βρούμε, αντιγράφουμε το path στο command window και στη συνέχεια κάνουμε τη σύνδεση. Η διαδικασία φαίνεται και παρακάτω στην εικόνα που ακολουθεί.



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Nick>cd C:\Users\Nick\Downloads\adt-bundle-windows-x86_64-20140702\sdk\platform-tools
C:\Users\Nick\Downloads\adt-bundle-windows-x86_64-20140702\sdk\platform-tools>adb connect 192.168.2.2:5555
already connected to 192.168.2.2:5555
C:\Users\Nick\Downloads\adt-bundle-windows-x86_64-20140702\sdk\platform-tools>
```

Εικόνα 28: Σύνδεση WiFi ADB με Eclipse

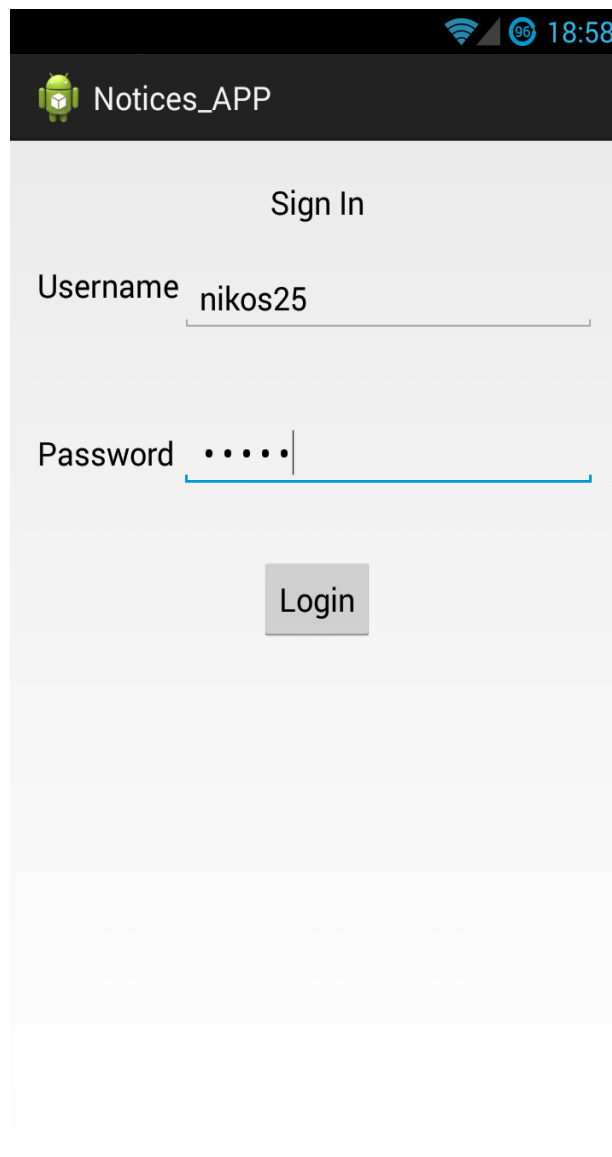
Μετά από το βήμα αυτό μπορούμε να κάνουμε την εφαρμογή μας compile κατευθείαν στον κινητό μας, κατά το οποίο η εφαρμογή εγκαθιστάτε κανονικά όπως και με τις εφαρμογές που κυκλοφορούν στο ψηφιακό μαγαζί της Google (Play Store).

4 Εφαρμογή Android για λήψη ανακοινώσεων

Στο κεφάλαιο αυτό θα ασχοληθούμε με την ανάλυση της εφαρμογής την οποία υλοποιήσαμε. Η εφαρμογή αυτή ονομάζεται **Notices_APP** και η λειτουργία της είναι, με ένα απλό login να λαμβάνεις ανακοινώσεις από νέα μιας συγκεκριμένης ιστοσελίδας, που στη περίπτωση μας είναι το "<http://www.athenswalk.gr>".

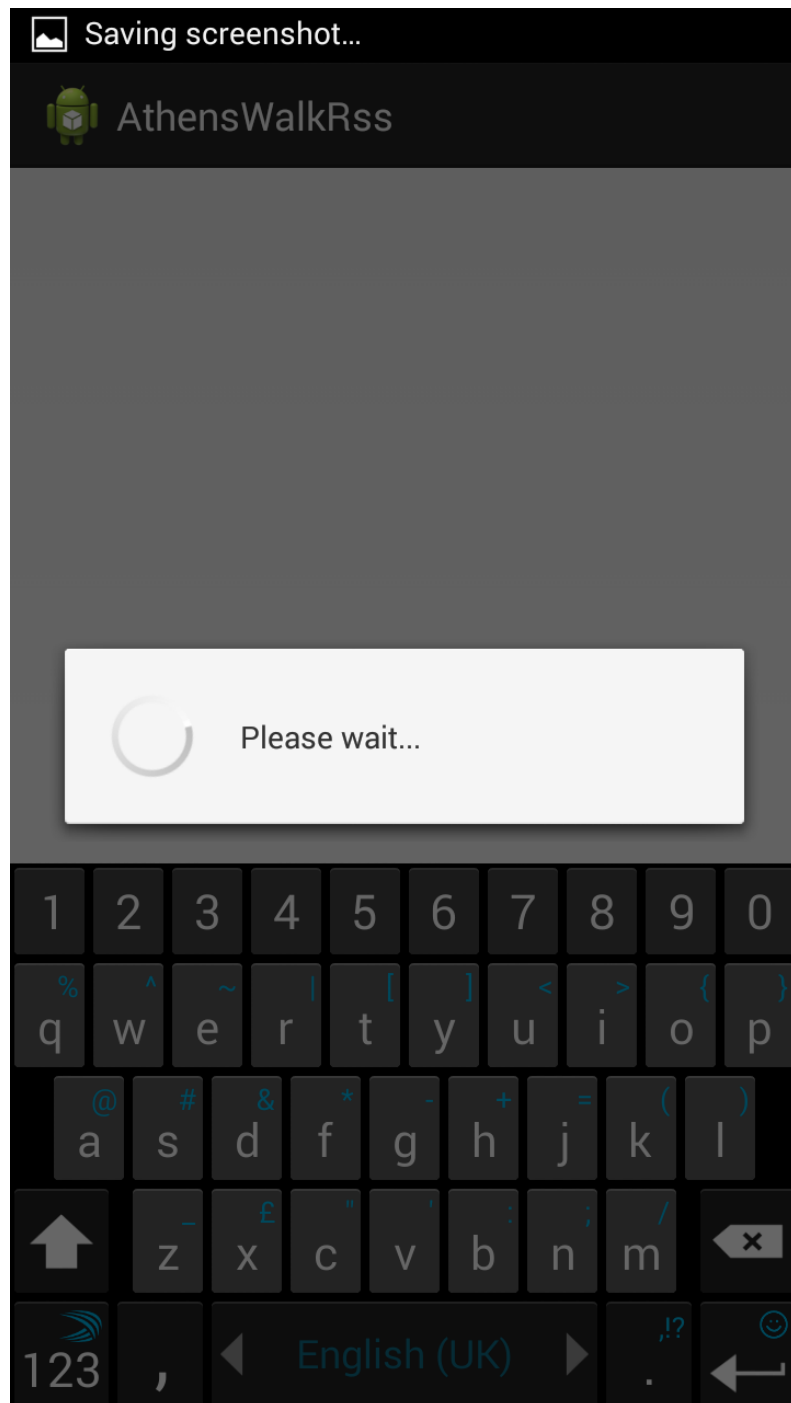
4.1 Αρχική οθόνη της εφαρμογής

Στην αρχική οθόνη της εφαρμογής βλέπουμε το login screen. Βάζουμε το username και το password που επιθυμούμε – όπως βλέπουμε στην εικόνα παρακάτω- και στη συνέχεια θα έχουμε τη δυνατότητα να διαβάζουμε τις ανακοινώσεις και τα νέα της ιστοσελίδας μας.



Εικόνα 29: Αρχική σελίδα της Notices_APP

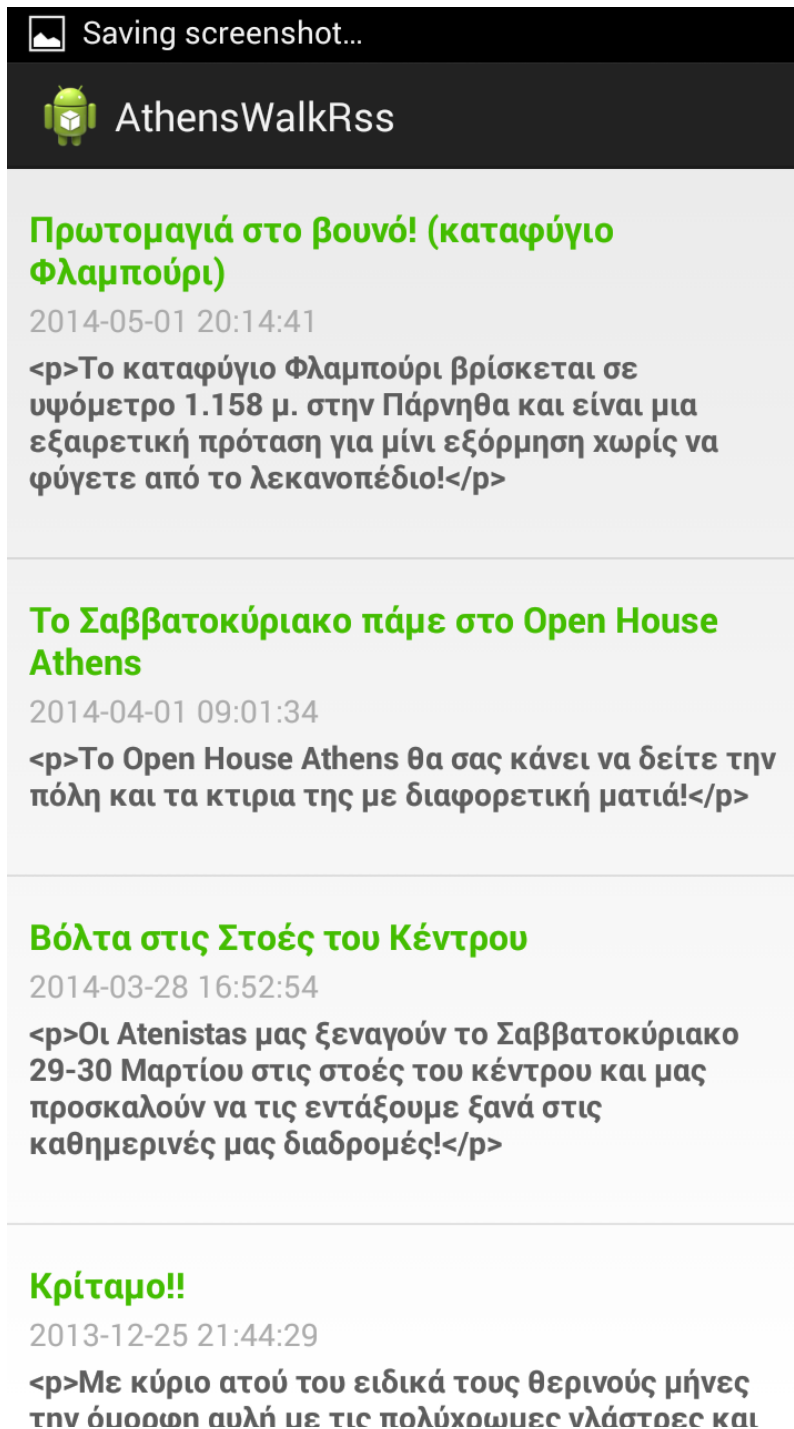
Αφού πατήσουμε το κουμπί LogIn, ακολουθεί η διαδικασία της σύνδεσης με το site μας. Μπορούμε να δούμε την οθόνη με τη διαδικασία αυτή στην ακόλουθη εικόνα.



Εικόνα 30: Οθόνη κατά τη σύνδεση με την ιστοσελίδα

4.2 Οθόνη με τις τελευταίες ανακοινώσεις της ιστοσελίδας

Στην οθόνη αυτή της εφαρμογής μας μπορούμε να δούμε τις ανακοινώσεις της ιστοσελίδας μας. Μπορούμε να διαβάσουμε τον τίτλο της ανακοίνωσης, την ημερομηνία και ώρα όπου βγήκε η εκάστοτε ανακοίνωση, καθώς και το κείμενο που περιλαμβάνει περισσότερες πληροφορίες για το θέμα που μας ενδιαφέρει να διαβάσουμε και να ενημερωθούμε. Η αντίστοιχη εικόνα ακολουθεί παρακάτω.



Εικόνα 31: Ανακοινώσεις του AthensWalk

4.3 Αποσύνδεση χρήστη ή διατήρηση σύνδεσης

Με την εφαρμογή αυτή έχουμε τη δυνατότητα αν θέλουμε να αποσυνδεθούμε από την εφαρμογή, για να υπάρχει δυνατότητα σύνδεσης και τρίτου ατόμου ή να παραμείνουμε συνδεδεμένοι για να μαθαίνουμε πρώτοι τα νέα της ιστοσελίδας μας. Η αποσύνδεση του χρήστη γίνεται πολύ απλά, αφού με το πάτημα ενός κουμπιού του κινητού μας, το οποίο φαίνεται και στην ακόλουθη εικόνα, μπορούμε να αποσυνδεθούμε από την εφαρμογή μας, πατώντας το κουμπί **Log off**.

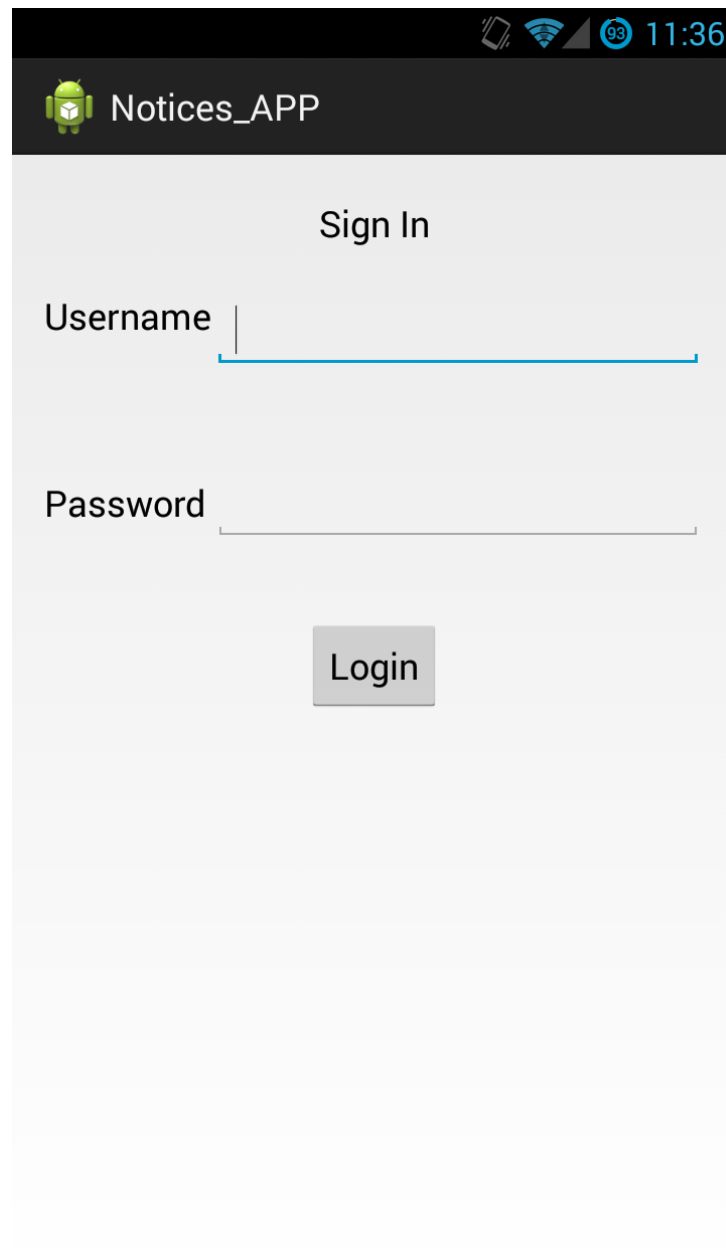


Εικόνα 32: Κουμπί συσκευής που βοηθάει στην αποσύνδεση χρήστη



Εικόνα 33: Κουμπί Log off

Αφού γίνει η αποσύνδεση χρήστη, επιστρέφουμε στην αρχική οθόνη όπου εισάγουμε username και password όπως φαίνεται παρακάτω.



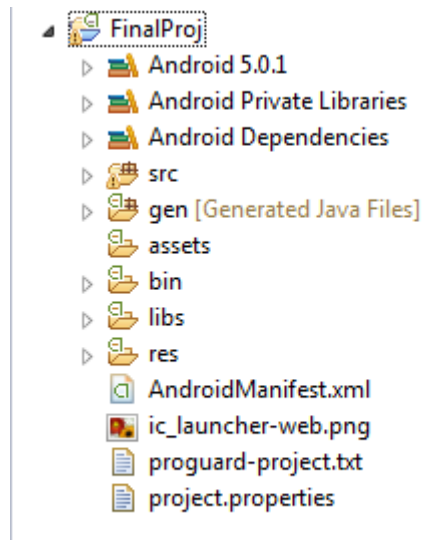
Εικόνα 34: Αρχική οθόνη

Τέλος αν δεν επιθυμούμε να αποσυνδεθούμε, πατώντας το κουμπί της αρχικής οθόνης της συσκευής μας, μπορούμε να παραμείνουμε συνδεδεμένοι για όσο εμείς θέλουμε και να βλέπουμε πρώτοι τα νέα που μας ενδιαφέρουν.

5 Πηγαίος κώδικας της εφαρμογής

5.1 Δομή της εφαρμογής μας

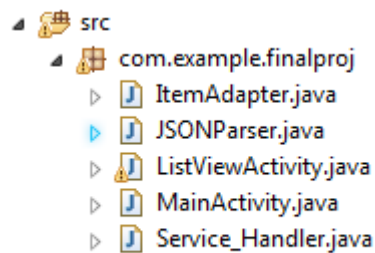
Πριν αναλύσουμε τον κώδικα της εφαρμογής μας, είναι χρήσιμο να κάνουμε μια συνοπτική αναφορά στη δομή της, από ποια αρχεία αποτελείται η εφαρμογή μας και ποια η χρήση τους. Στην επόμενη εικόνα βλέπουμε πώς είναι δομημένη η εφαρμογή μας στο Eclipse.



Εικόνα 35: Δομή εφαρμογής στο Eclipse

Βλέπουμε πως υπάρχουν πολλοί φάκελοι που αναλύονται παρακάτω.

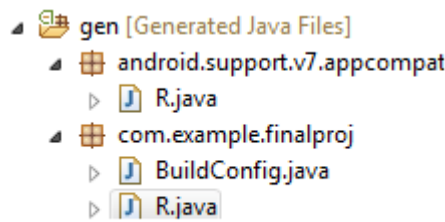
src/



Εικόνα 36: Περιεχόμενα του src

Στην εικόνα αυτή βλέπουμε πως μέσα στον φάκελο src, υπάρχει το πακέτο com.example.finalproj, μέσα στο οποίο υπάρχουν οι κλάσεις που χρησιμοποιήσαμε στην εφαρμογή μας.

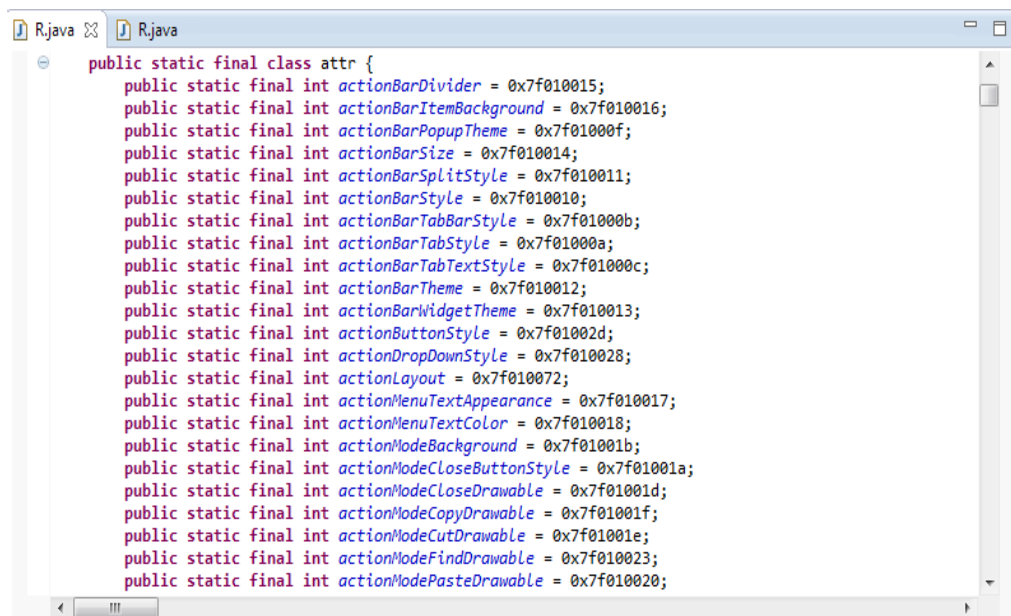
gen/



Εικόνα 37: Περιεχόμενα του gen

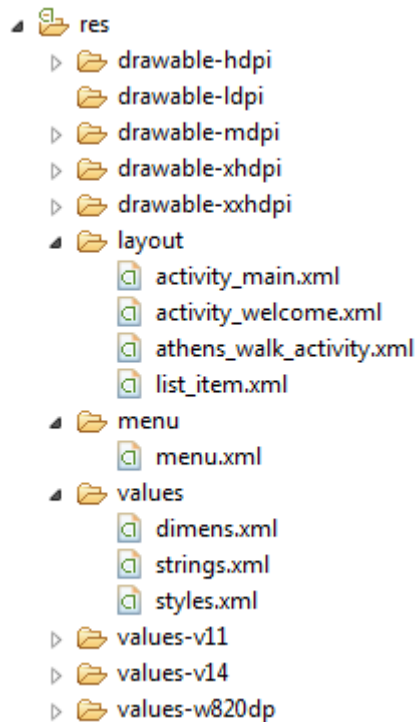
Βλέπουμε στην εικόνα αυτή πως ο φάκελος gen περιέχει δύο πακέτα. Το πρώτο πακέτο αφορά μια βιβλιοθήκη που βοηθάει στη συμβατότητα της εφαρμογής με διάφορες εκδόσεις λογισμικού. Το πακέτο αυτό βλέπουμε πως περιέχει ένα αρχείο R.java που αποτελεί και μια κλάση. Το ίδιο βλέπουμε και στο επόμενο πακέτο που αφορά την εφαρμογή μας καθώς και ένα αρχείο BuildConfig.java, το οποίο όμως δε θα μας απασχολήσει για την παρούσα παρουσίαση.

Οι δύο αυτές κλάσεις (R.java) είναι και αυτές που παίζουν τον κυριότερο ρόλο στο κομμάτι αυτό της δομής της εφαρμογής και αυτό συμβαίνει γιατί, είναι κλάσεις που κατασκευάζονται αυτόματα από το σύστημα και συνδέει όλες της πηγές της εφαρμογής. Αυτό που κάνουν δηλαδή είναι να δηλώνουν κάθε στοιχείο, από εικόνες, γραφικά στοιχεία, αλφαριθμητικά καθώς και θέματα που χρησιμοποιούμε, ως μια ακέραια σταθερά, η οποία έχει τιμή ένα μοναδικό δεκαεξαδικό αριθμό. Το αποτέλεσμα είναι ότι μπορούμε να αναφερόμαστε στα στοιχεία της κλάσης μέσα από τις δικές μας κλάσεις σαν δημόσιες σταθερές.



Εικόνα 38: Περιεχόμενα R.java

res/



Εικόνα 39: Περιεχόμενα res

Στον φάκελο αυτό αποθηκεύονται όλα τα application resources. Στο φάκελο layout για παράδειγμα, αποθηκεύονται όλα τα .xml αρχεία που δηλώνουμε τη διεπαφή χρήστη σε κάθε οθόνη. Στον φάκελο menu, δηλώσαμε ένα xml αρχείο, στο οποίο είναι το περιβάλλον για το activity του μενού της εφαρμογής. Τέλος στον φάκελο values έχουμε δηλώσει όλα τα strings, dimensions που χρειαστήκαμε και styles που αφορούν τη συμβατότητα σε προηγούμενες εκδόσεις λειτουργικού Android.

```

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

    <string name="app_name">Notices_APP</string>
    <string name="Logoff">Log off</string>
    <string name="hello_world">Hello world!</string>
    <string name="Username">Username</string>
    <string name="Password">Password</string>
    <string name="Signin">Sign In</string>
    <string name="Login">Login</string>
    <string name="Logout">Logout</string>
    <string name="title_activity_welcome">Welcome</string>
    <string name="exit">Exit without logout</string>
    <string name="title_activity_listactivity">AthensWalkRss</string>

</resources>

```

Εικόνα 40: Περιεχόμενα του strings.xml

Τέλος κάθε φορά που πάμε να δημιουργήσουμε ένα νέο project, αυτόματα δημιουργείτε και το αρχείο AndroidManifest.xml. Το αρχείο αυτό περιέχει διάφορες πολύ βασικές πληροφορίες

σχετικά με την εφαρμογή μας. Αυτές οι πληροφορίες είναι πολύ σημαντικές, αφού χρειάζονται για τη σωστή εκτέλεση του κώδικά μας. Στη συνέχεια βλέπουμε πώς είναι ο κώδικας μέσα στο xml αρχείο.

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

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="21" />

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

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >

        <activity
            android:name="com.example.finalproj.MainActivity"
            android:label="@string/app_name" >

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

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity
            android:name="com.example.finalproj.ListViewActivity"
            android:label="@string/title_activity_listactivity" >
        </activity>

    </application>
</manifest>
```

5.2 Αναλύοντας την εφαρμογή μας

5.2.1 Αρχική οθόνη εφαρμογής

Σε αυτό το κομμάτι θα ασχοληθούμε με τον κώδικά που γράψαμε για να μπορέσουμε να δημιουργήσουμε την εφαρμογή μας και θα αναλύσουμε τα βασικότερα κομμάτια αυτού. Ξεκινώντας

λοιπόν με την πρώτη μας κλάση, που αφορά την αρχική οθόνη της εφαρμογής μας, βλέπουμε τον ακόλουθο κώδικα.

```
package com.example.finalproj;
```

```
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.SharedPreferences.Editor;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
```

```
public class MainActivity extends Activity {
```

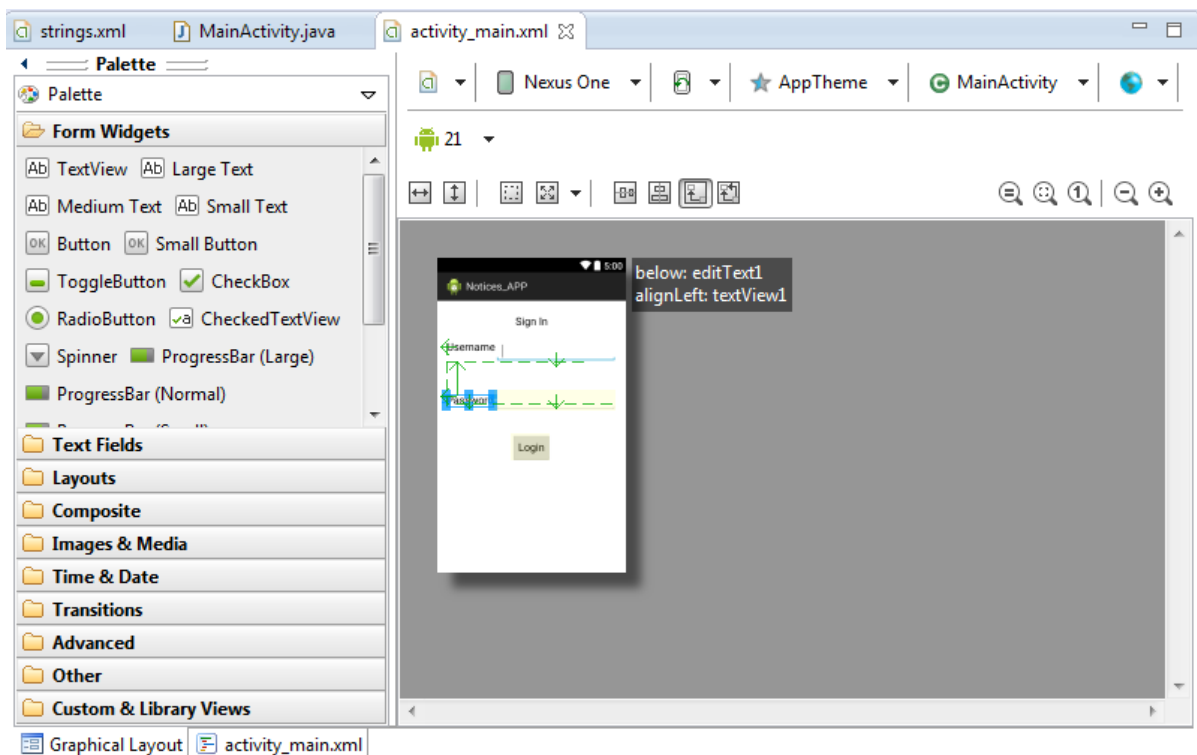
```
    private EditText username,password;
    public static final String MyPREFERENCES = "MyPrefs" ;
    public static final String name = "nameKey";
    public static final String pass = "passwordKey";
    SharedPreferences sharedpreferences;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        username = (EditText)findViewById(R.id.editText1);
        password = (EditText)findViewById(R.id.editText2);
    }
    @Override
    protected void onResume() {
        sharedpreferences=getSharedPreferences(MyPREFERENCES,
        Context.MODE_PRIVATE);
        if (sharedpreferences.contains(name))
        {
            if(sharedpreferences.contains(pass)){
                Intent i = new Intent(this,com.example.finalproj.
                ListViewActivity.class);
                startActivity(i);
            }
        }
        super.onResume();
    }
    public void login(View view){
        Editor editor = sharedpreferences.edit();
        String u = username.getText().toString();
        String p = password.getText().toString();
        editor.putString(name, u);
        editor.putString(pass, p);
        editor.commit();
        Intent i = new Intent(this,com.example.finalproj.ListViewActivity.class);

        startActivity(i);
    }
}
```

Ο κώδικας αυτός ανήκει στην αρχική σελίδα της εφαρμογής και όπως βλέπουμε από το ξεκίνημα του κώδικα έχουμε δηλώσει τις απαραίτητες βιβλιοθήκες που χρειαζόμαστε γράφοντας import και στη συνέχεια το όνομα της βιβλιοθήκης που έχουμε ανάγκη.

Στη συνέχεια δηλώνουμε τις μεταβλητές που θα χρειαστούμε. Σε αυτό το σημείο θα αναφέρουμε πως το username και το password αποθηκεύονται στο sharedpreferences του κινητού μας τηλεφώνου. Είναι κάτι δηλαδή σαν προσωρινή μνήμη, όπου χάνει τις τιμές που έχει πάρει με το κλείσιμο της εφαρμογής μας ή με το να δώσουμε καινούργια στοιχεία για username και password. Σημαντικό στοιχείο στον κώδικα αυτό είναι η Intent που χρησιμοποιούμε. Είναι ένα αντικείμενο που μας παρέχει δεσμευτική εκτέλεση μεταξύ δύο activities. Χρησιμοποιείται κυρίως για αυτό τον λόγο, δηλαδή να ξεκινάει ένα άλλο activity.

Τέλος το user interface της εφαρμογής το δημιουργήσαμε με τη βοήθεια του Eclipse αφού μας δίνει τη δυνατότητα με xml αρχεία να σχεδιάσουμε την εφαρμογή μας όπως εμείς θέλουμε κάνοντας διάφορα αντικείμενα drag and drop πάνω στο σημείο της οθόνης του κινητού που θέλουμε. Φαίνεται στη συνέχεια παράδειγμα μέσα από το Eclipse καθώς και ο κώδικας του xml αρχείου.



Εικόνα 41: Δημιουργία xml αρχείου

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.finalproj.MainActivity" >
```

```
<TextView
```

```
android:id="@+id/textView1"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignParentTop="true"  
android:layout_marginTop="52dp"  
android:text="@string/Username"  
android:textAppearance="?android:attr/textAppearanceMedium" />
```

<TextView

```
android:id="@+id/textView3"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_above="@+id/editText1"  
android:layout_centerHorizontal="true"  
android:layout_marginBottom="22dp"  
android:text="@string/Signin"  
android:textAppearance="?android:attr/textAppearanceMedium" />
```

<Button

```
android:id="@+id/button1"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_below="@+id/editText2"  
android:layout_centerHorizontal="true"  
android:layout_marginTop="45dp"  
android:onClick="login"  
android:text="@string/Login" />
```

<TextView

```
android:id="@+id/textView2"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignLeft="@+id/textView1"  
android:layout_below="@+id/editText1"  
android:layout_marginTop="54dp"  
android:text="@string/Password"  
android:textAppearance="?android:attr/textAppearanceMedium" />
```

<EditText

```
android:id="@+id/editText1"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignParentRight="true"  
android:layout_alignTop="@+id/textView1"  
android:layout_toRightOf="@+id/textView1"  
android:ems="10"  
android:inputType="text" >
```

<requestFocus />

</EditText>

<EditText

```
android:id="@+id/editText2"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"
```



```
android:layout_alignBaseline="@+id/textView2"
android:layout_alignBottom="@+id/textView2"
android:layout_alignLeft="@+id/editText1"
android:layout_alignParentRight="true"
android:ems="10"
android:inputType="textPassword" />
```

</RelativeLayout>

5.2.2 Σύνδεση με το διαδίκτυο, αποστολή και παραλαβή πακέτων

Ο κώδικας που θα δούμε στη συνέχεια αφορά την αποστολή και λήψη πακέτων δεδομένων με τον διακομιστή.

```
package com.example.finalproj;
```

```
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.util.List;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.client.utils.URLEncodedUtils;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.util.EntityUtils;
```

```
public class Service_Handler {
```

```
    static String response = null;
    public final static int GET = 1;
    public final static int POST = 2;
```

```
    public Service_Handler() {
```

```
    }
```

```
    public String makeServiceCall(String url, int method) {
        return this.makeServiceCall(url, method, null);
    }
```

```
    public String makeServiceCall(String url, int method,
        List<NameValuePair> params) {
        try {
```

```
            DefaultHttpClient httpClient = new DefaultHttpClient();
```

```

HttpEntity httpEntity = null;
HttpResponse httpResponse = null;

if (method == POST) {
    HttpPost httpPost = new HttpPost(url);
    if (params != null) {
        httpPost.setEntity(new UrlEncodedFormEntity(params));
    }

    httpResponse = httpClient.execute(httpPost);

} else if (method == GET) {
    if (params != null) {
        String paramString = URLEncodedUtils
            .format(params, "utf-8");
        url += "?" + paramString;
    }
    HttpGet httpGet = new HttpGet(url);

    httpResponse = httpClient.execute(httpGet);

}
httpEntity = httpResponse.getEntity();
response = EntityUtils.toString(httpEntity);

} catch (UnsupportedEncodingException e) {
    e.printStackTrace();
} catch (ClientProtocolException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}

return response;
}
}

```

5.2.3 JSON

Σημαίνει **JavaScript Object Notation** και είναι ένας τρόπος να αποθηκεύεις πληροφορίες με έναν οργανωμένο και εύκολο σε πρόσβαση τρόπο. Μας δίνει δηλαδή μια συλλογή από δεδομένα που είναι εύκολο να διαβάσουμε και να κατανοήσουμε. Ο κώδικας που χρησιμοποιήσαμε είναι ο ακόλουθος.

```

package com.example.finalproj;

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

```

```

import java.io.UnsupportedEncodingException;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.json.JSONException;
import org.json.JSONObject;

import android.util.Log;

public class JSONParser {
    static InputStream is = null;
    static JSONObject jsonObj = null;
    static String json = "";

    // constructor
    public JSONParser() {

    }

    public JSONObject getJSONFromUrl(String url) {

        // Making HTTP request
        try {
            // defaultHttpClient
            DefaultHttpClient httpClient = new DefaultHttpClient();
            HttpPost httpPost = new HttpPost(url);

            HttpResponse httpResponse = httpClient.execute(httpPost);
            HttpEntity httpEntity = httpResponse.getEntity();
            is = httpEntity.getContent();

        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        } catch (ClientProtocolException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }

        try {
            BufferedReader reader = new BufferedReader(new InputStreamReader(
                is, "iso-8859-1"), 8);
            StringBuilder sb = new StringBuilder();
            String line = null;
            while ((line = reader.readLine()) != null) {
                sb.append(line + "\n");
            }
            is.close();
            json = sb.toString();
        } catch (Exception e) {
            Log.e("Buffer Error", "Error converting result " + e.toString());
        }
    }
}

```

```

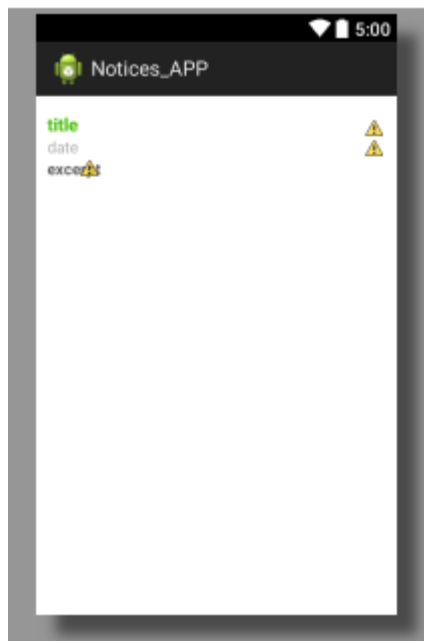
// try parse the string to a JSON object
try {
    jsonObj = new JSONObject(json);
} catch (JSONException e) {
    Log.e("JSON Parser", "Error parsing data " + e.toString());
}

// return JSON String
return jsonObj;
}
}

```

5.2.4 Εμφάνιση ανακοινώσεων

Στο κομμάτι αυτό αποφασίσαμε να δώσουμε μια πολύ κατανοητή και απλή παρουσίαση των ανακοινώσεων που θα λαμβάνουμε στην εφαρμογή μας από την ιστοσελίδα που επιλέξαμε. Δουλέψαμε λοιπόν με τέτοιο τρόπο ώστε κάθε ανακοίνωση να έχει τον τίτλο της, την ημερομηνία και ώρα της καθώς και το σχετικό κείμενο που την αποτελεί. Φαίνεται στην ακόλουθη εικόνα το xml αρχείο κατά τη σχεδίαση της εφαρμογής και τέλος ο κώδικας.



Εικόνα 42: Τρόπος εμφάνισης ανακοινώσεων

```

package com.example.finalproj;

```

```

import java.util.ArrayList;
import java.util.HashMap;

```

```

import android.content.Context;

```

```

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.TextView;

public class ItemAdapter extends BaseAdapter {

    public final Context m_context;
    private final ArrayList<HashMap<String, Object>> m_itemViews;
    LayoutInflater layoutInflater;

    public ItemAdapter(Context _context, ArrayList<HashMap<String, Object>> _itemViews)
    {
        super();
        this.m_context = _context;
        this.m_itemViews = _itemViews;
        layoutInflater = LayoutInflater.from(_context);
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent)
    {
        convertView= layoutInflater.inflate(R.layout.list_item, parent, false);

        TextView textView_title = (TextView) convertView.findViewById(R.id.title);
        TextView textView_date = (TextView) convertView.findViewById(R.id.date);
        TextView textView_excerpt = (TextView) convertView.findViewById(R.id.excerpt);

        textView_title.setText(m_itemViews.get(position).get("title").toString());
        textView_date.setText(m_itemViews.get(position).get("date").toString());
        textView_excerpt.setText(m_itemViews.get(position).get("excerpt").toString());

        return convertView;
    }

    @Override
    public int getCount() {
        // TODO Auto-generated method stub
        return m_itemViews.size();
    }

    @Override
    public Object getItem(int position) {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public long getItemId(int position) {
        // TODO Auto-generated method stub
        return position;
    }
}

```

Τα σημαντικότερα στοιχεία του κώδικα αυτού είναι το `LayoutInflater` και το `ArrayList`. Το πρώτο χρησιμοποιείται για να τοποθετήσει τα αντικείμενα ενός xml αρχείου στην θέση που τους ανήκει ώστε να δημιουργηθεί σωστά μια εφαρμογή. Αυτό το κάνει ανάλογα με τα `id` που έχουμε δώσει σε συγκεκριμένα αντικείμενα του project μας. Με λίγα λόγια χρησιμοποιεί σαν είσοδο ένα αρχείο xml και χτίζει τα αντικείμενα από αυτό.

Με το `ArrayList` δημιουργούμε μια λίστα από γραμμές και πάνω σε κάθε γραμμή της λίστας, τοποθετούμε με τη σειρά τα στοιχεία που έχουμε δημιουργήσει. Μπορεί να είναι `string` αλλά και `object`.

Ακολουθεί ο xml κώδικας που αφορά την εμφάνιση των ανακοινώσεων.

```
<?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="wrap_content"
    android:orientation="vertical"
    android:padding="10dp"
    android:paddingLeft="10dp"
    android:paddingRight="10dp" >

    <TextView
        android:id="@+id/title"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:paddingBottom="2dip"
        android:paddingTop="6dip"
        android:textColor="#43bd00"
        android:textSize="16sp"
        android:text="title"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/date"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:paddingBottom="2dip"
        android:text="date"
        android:textColor="#acacac" />

    <TextView
        android:id="@+id/excerpt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="left"
        android:text="excerpt"
        android:textColor="#5d5d5d"
        android:textStyle="bold" />

</LinearLayout>
```

5.2.5 RSS feed μέσω AsyncTask

Αφού ολοκληρώσαμε και το parser μας, πρέπει να το χρησιμοποιήσουμε κιόλας. Η AsyncTask, ουσιαστικά χρησιμοποιείτε όταν θέλουμε να τρέξει μια διαδικασία στο background του user interface και να εμφανίσει τα επιθυμητά αποτελέσματα. Βλέπουμε παρακάτω τον κώδικα με τον οποίο γίνεται η λήψη RSS μέσω της AsyncTask.

```
package com.example.finalproj;

import java.util.ArrayList;
import java.util.HashMap;

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

import android.app.ListActivity;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.SharedPreferences;
import android.content.SharedPreferences.Editor;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ListView;
import android.widget.TextView;

public class ListViewActivity extends ListActivity {

    private ProgressDialog pDialog;

    // URL to get contacts JSON
    private static String url = "http://www.athenswalk.gr/api/get_recent_posts/";

    // JSON Node names

    private static final String TAG_POSTS = "posts";
    private static final String TAG_URL = "url";
    private static final String TAG_TITLE = "title";
    private static final String TAG_DATE = "date";
    private static final String TAG_CONTENT = "content";
    private static final String TAG_EXCREPT = "excerpt";

    // contacts JSONArray
    JSONArray objView = null;
```

```

// HashMap for ListView
ArrayList<HashMap<String, Object>> objViewsList;

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

    objViewsList = new ArrayList<HashMap<String, Object>>();

    ListView lv = getListView();

    // ListView on item click listener
    lv.setOnItemClickListener(new OnItemClickListener() {

        @Override
        public void onItemClick(AdapterView<?> parent, View view,
            int position, long id) {
            // getting values from selected ListItem
            String title = ((TextView) view.findViewById(R.id.title))
                .getText().toString();
            String date = ((TextView) view.findViewById(R.id.date))
                .getText().toString();
            String excerpt = ((TextView) view.findViewById(R.id.excerpt))
                .getText().toString();

        }
    });

    // Calling async task to get json
    new GetRssPosts().execute();
}

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

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.logoff:
            SharedPreferences
sharedpreferences=getSharedPreferences(MainActivity.MyPREFERENCES,
                Context.MODE_PRIVATE);
            Editor editor = sharedpreferences.edit();
            editor.clear();
            editor.commit();
            super.onBackPressed();
            break;
    }
    return true;
}

```



```

}
/**
 * Async task class to get json by making HTTP call
 * */
private class GetRssPosts extends AsyncTask<Void, Void, Void> {

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        // Showing progress dialog
        progressDialog = new ProgressDialog(ListViewActivity.this);
        progressDialog.setMessage("Please wait...");
        progressDialog.setCancelable(false);
        progressDialog.show();
    }

    @Override
    protected Void doInBackground(Void... arg0) {
        // Creating service handler class instance
        Service_Handler sh = new Service_Handler();

        // Making a request to url and getting response
        String jsonStr = sh.makeServiceCall(url, Service_Handler.GET);

        Log.d("Response: ", ">" + jsonStr);

        if (jsonStr != null) {
            try {
                JSONObject jsonObj = new JSONObject(jsonStr);

                objView = jsonObj.getJSONArray(TAG_POSTS);

                // looping through All objects
                for (int i = 0; i < objView.length(); i++) {
                    JSONObject c = jsonObj.getJSONObject(i);

                    String url = c.getString(TAG_URL);
                    String title = c.getString(TAG_TITLE);
                    String date = c.getString(TAG_DATE);
                    String content = c.getString(TAG_CONTENT);
                    String excerpt = c.getString(TAG_EXCREPT);

                    // tmp hashmap for single itemView
                    HashMap<String, Object> itemView = new
HashMap<String, Object>();

                    // adding each child node to HashMap key => value
                    itemView.put(TAG_URL, url);
                    itemView.put(TAG_TITLE, title);
                    itemView.put(TAG_DATE, date);
                    itemView.put(TAG_CONTENT, content);
                    itemView.put(TAG_EXCREPT, excerpt);

```

```

        // adding contact to contact list
        objViewsList.add(itemView);
    }

    } catch (JSONException e) {
        e.printStackTrace();
    }
    } else {
        Log.e("ServiceHandler", "Couldn't get any data from the url");
    }

    return null;
}

@Override
protected void onPostExecute(Void result) {
    super.onPostExecute(result);
    // Dismiss the progress dialog
    if (pDialog.isShowing())
        pDialog.dismiss();
    /**
     * Updating parsed JSON data into ListView
     */

    //A custom example adapter
    ItemAdapter adapter = new ItemAdapter(
        ListViewActivity.this,
        objViewsList);
    setListAdapter(adapter);
}

}

}

```

Χρησιμοποιώντας τις μεθόδους του JSON Parser εισάγουμε σε μια λίστα τύπου πίνακα (ArrayList<HashMap>), την objViewsList, όλες τις τιμές του RSS που μας ενδιαφέρουν προκειμένου να εμφανίσουμε στην ListView μας.

6 Προτάσεις βελτίωσης της εφαρμογής

Ο τομέας της ανάπτυξης εφαρμογών για σύγχρονες κινητές συσκευές, εξελίσσεται ραγδαία, κάτι που καθιστά σίγουρη τη μελλοντική έρευνα και τη βελτίωση των εφαρμογών. Μερικές προτάσεις για τη βελτίωση αυτή είναι οι εξής:

- Δημιουργία καλύτερου interface της εφαρμογής, προσθέτοντας εικόνες ή ακόμα και 3D γραφικά.
- Δυνατότητα επιλογής περισσότερων ιστοσελίδων της αρεσκείας μας από μια λίστα συνεργαζόμενων ιστότοπων, με τη χρήση ενός και μόνο λογαριασμού.
- Δημιουργία ρυθμίσεων μέσα στην εφαρμογή για καλύτερη εμπειρία χρήστη. Επιλογή για παράδειγμα ανάγνωσης το βράδυ, όπου θα χαμηλώνει ή θα αυξάνεται η φωτεινότητα στην οθόνη καθώς επίσης και αλλαγή χρωμάτων και άλλες.
- Μπορούμε αφού τη φτάσουμε σε ένα ικανοποιητικό επίπεδο βελτίωσης, να την προωθήσουμε και στο Play Store (εικονικό μαγαζί για εφαρμογές) της Google.
- Τέλος, θα ήταν πολύ σημαντικό να υλοποιήσουμε την ίδια ακριβώς εφαρμογή και για άλλα λειτουργικά συστήματα κινητών τηλεφώνων, όπως για παράδειγμα το iOS της εταιρίας Apple και Windows της ομώνυμης εταιρίας.

7 Επίλογος

Η ανάπτυξη μιας εφαρμογής Android είναι μια ιδιαίτερα απαιτητική εργασία, η οποία απαιτεί πολύ καλή γνώση της γλώσσας προγραμματισμού Java, καθώς επίσης και μεγάλη εμπειρία στον προγραμματισμό γενικότερα. Με αυτόν τον τρόπο ο κάθε προγραμματιστής, θα μπορεί να δημιουργήσει κάτι πραγματικό χρήσιμο.

Για την υλοποίηση της εφαρμογής αφιέρωσα αρκετό χρόνο διαβάζοντας τη γλώσσα προγραμματισμού Java, καθώς είχα πολύ βασικές γνώσεις πριν ξεκινήσω τη δημιουργία της εφαρμογής μου. Το project ξεκίνησε 2 μήνες αφού είχα ξεκινήσει την εκμάθηση των βασικών συστατικών της Java. Συνολικά το project μου πήρε 4-5 μήνες για να το υλοποιήσω, μαζί με το Debugging.

Συναντήσαμε αρκετές δυσκολίες καθότι οι δομές της εφαρμογής ήταν απαιτητικές για νέους προγραμματιστές, όπως για παράδειγμα η λήψη και προβολή των xml αρχείων και των RSS. Καταφέραμε όμως να υλοποιήσουμε την εφαρμογή όπως τη θέλαμε και να αποκτήσουμε με αυτό αρκετή εμπειρία και εφόδια που μπορούμε να αξιοποιήσουμε στο μέλλον.

ΠΗΓΕΣ

<https://www.android.com/>
[http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))
<http://www.vodafone.gr/portal/client/cms/viewCmsPage.action?pageId=10297>
<http://el.wikipedia.org/wiki/Smartphone>
<http://blog.compassintelligence.com/?p=832>
<http://www.adappt.co.uk/ios-blackberry-symbian-and-microsoft-vs-android-fair-fight>
<http://blog.bugsense.com/post/26918984231/android-vs-windows-phone-from-a-developers-scope>
<http://www.pcmag.com/article2/0,2817,2476480,00.asp>
<http://standartux.fr/?p=693>
<http://www.itsmac.net/xcode-5-developer-preview-4-tools/>
[https://msdn.microsoft.com/en-us/library/windows/apps/ff630878\(v=vs.105\).aspx](https://msdn.microsoft.com/en-us/library/windows/apps/ff630878(v=vs.105).aspx)
http://www.tutorialspoint.com/android/android_architecture.htm
http://elinux.org/Android_Architecture
<http://www.myphone.gr/forum/showthread.php?t=306146>
http://artemis.cslab.ntua.gr/el_thesis/artemis.ntua.ece/DT2011-0144/DT2011-0144.pdf
<http://nuttycloud.com/android-os-architecture-now-in-question/>
<http://www.mydroid.gr/2013/02/leitourgiko-android-ekdoseis/>
http://el.wikipedia.org/wiki/%CE%99%CF%83%CF%84%CE%BF%CF%81%CE%AF%CE%B1_%CE%B5%CE%BA%CE%B4%CF%8C%CF%83%CE%B5%CF%89%CE%BD_%CF%84%CE%BF%CF%85_Android
<http://www.android.com/history/>
<http://android.wonderhowto.com/inspiration/android-lollipop-all-new-features-you-need-know-about-0158258/>
<http://developer.android.com/about/index.html>
<http://android.appstorm.net/general/opinion/why-i-choose-android-better-open-development/>