

Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

**Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Μηχανικών Πληροφορικής**



Πτυχιακή Εργασία

«Δημιουργία εφαρμογής για έξυπνα κινητά που υποστηρίζει διασύνδεση με Portals»

Παντελής Ζαχαρίας ΑΜ: 1110

Επιβλέπων καθηγητής: Βιδάκης Νικόλαος

Ηράκλειο 2016

Ευχαριστίες

Ευχαριστώ πάρα πολύ τους καθηγητές μου κ. Βιδάκη Νικόλαο και κ. Ακουμιανάκη Δημοσθένη για την συμπαράσταση και την υπομονή που έδειξαν, όλα τα παιδιά του iSTLab και κυρίως τον Γιώργο για την συνεχή υποστήριξη και καθοδήγηση του. Τέλος θέλω να ευχαριστήσω όλους αυτούς που με στήριξαν και πίστεψαν σε μένα.

Abstract

In this thesis we dealt with an implementation for mobile devices using the software "Android" issued by Google as an "open source software" in October 2008 and adopted by many leading companies in the field of mobile. The purpose of the application is to provide faster and better access to the portal created by iSTLab under the research program EuroCancerComs. The portal provides to the users, access to articles concerning breast cancer and the ability to interact with other users by using objects of "social networking" (Social widgets) as "tagging", "evaluation" and more.

Σύνοψη

Στην παρούσα πτυχιακή εργασία ασχοληθήκαμε με την υλοποίηση μιας εφαρμογής για κινητές συσκευές που χρησιμοποιούν το λογισμικό «Android» το οποίο εκδόθηκε από την Google ως ένα «λογισμικό ανοιχτού κώδικα» (open source) τον Οκτώβριο του 2008 και υιοθετήθηκε από πολλές και κορυφαίες εταιρείες στον χώρο της κινητής τηλεφωνίας. Σκοπός της εφαρμογής είναι να δώσει γρηγορότερη και καλύτερη πρόσβαση στο portal το οποίο δημιουργήθηκε από το iSTLab στα πλαίσια του ερευνητικού προγράμματος EuroCancerComs. Το portal δίνει στους χρήστες την πρόσβαση σε άρθρα που αφορούν τον καρκίνο του μαστού και την δυνατότητα να αλληλεπιδράσουν με τους υπόλοιπους χρήστες χρησιμοποιώντας αντικείμενα «κοινωνικής δικτύωσης» (social widgets) όπως «tagging», «rating» και πολλά άλλα.

Πίνακας περιεχομένων

ΕΥΧΑΡΙΣΤΙΕΣ	- 1 -
ABSTRACT	- 2 -
ΣΥΝΟΨΗ	- 3 -
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ	- 4 -
ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ	- 6 -
1. ΕΙΣΑΓΩΓΗ	- 8 -
2. ΚΙΝΗΤΕΣ ΣΥΣΚΕΥΕΣ ΚΑΙ MOBILE INTERNET – ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ	- 9 -
2.1 Εισαγωγή στις κινητές συσκευές	- 9 -
2.2 Εμφάνιση έξυπνων κινητών συσκευών	- 10 -
2.3. Κινητά δίκτυα δεδομένων	- 11 -
3. MOBILE APPLICATION DEVELOPMENT	- 14 -
3.1 Εισαγωγή στο mobile application development	- 14 -
3.2 Λειτουργικά συστήματα για έξυπνες κινητές συσκευές	- 15 -
3.2.1 Symbian OS.....	- 16 -
3.2.2 BlackBerry OS.....	- 19 -
3.2.3 Apple iOS και iPhone.	- 19 -
4. ANDROID OS	- 22 -
4.1 Ιστορική αναδρομή στο Android	- 22 -
4.1.1 Η προϊστορία	- 22 -
4.1.2 Τα πρώτα χρόνια.	- 23 -
4.1.3 Εκδόσεις του Android και Nexus.	- 24 -
4.2 Αρχιτεκτονική του Android	- 26 -
4.2.1 Ανάλυση αρχιτεκτονικής.	- 26 -
4.2.2 Πυρήνας Linux (Linux Kernel).	- 26 -
4.2.3 Βιβλιοθήκες (Libraries).	- 27 -
4.2.4 Χρόνος εκτέλεσης (Android Runtime).	- 27 -
4.2.5 Πλαίσιο εφαρμογής (Application Framework).....	- 28 -
4.2.6 Εφαρμογές (Applications).....	- 29 -

4.3 Ανάλυση μιας εφαρμογής Android	- 29 -
4.3.1 AndroidManifest.xml	- 29 -
4.3.2 Ο φάκελος src	- 30 -
4.3.3 Ο φάκελος res	- 30 -
4.3.4 Ο φάκελος gen	- 30 -
4.3.5 Ο φάκελος bin	- 30 -
4.3.6 Ο φάκελος assets	- 31 -
4.4 Android και Γλώσσες Προγραμματισμού	- 31 -
4.4.1 Ποιές Γλώσσες προγραμματισμού χρησιμοποιούνται στο Android	- 31 -
4.4.2 Android και Java	- 31 -
5. ΠΡΟΓΡΑΜΜΑΤΙΖΟΝΤΑΣ ΜΕ ΤΟ ANDROID	- 33 -
5.1 Επιλογή IDE και εγκατάσταση παραμέτρων	- 33 -
5.1.1 Εγκατάσταση του Eclipse	- 33 -
5.1.2 Εγκατάσταση του ADT Plugin και του Android SDK Manager	- 34 -
5.1.3 Ρύθμιση του AVD Virtual Manager	- 35 -
5.2 Ανάπτυξη εφαρμογών στο Eclipse	- 39 -
5.2.1 Η πρώτη μου εφαρμογή	- 39 -
6. EUROCANCERCOMS ΚΑΙ ANDROID	- 47 -
6.1 Τι είναι το EuroCancerComs	- 47 -
6.2 Σκέψεις και ανάπτυξη στον ιστό	- 47 -
6.2.1 Clinical practice guideline management information systems	- 47 -
6.2.2 Social Widgets	- 49 -
6.3 Το EuroCancerComs στο Android	- 51 -
6.3.1 JSON και ευελιξία στο EuroCancerComs	- 51 -
6.3.2 Parsing του JSON	- 53 -
6.3.3 Παρουσίαση της εφαρμογής	- 57 -
7. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΒΕΛΤΙΩΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	- 61 -
7.1 Συμπεράσματα	- 61 -
7.2 Βελτιώσεις	- 61 -
ACKNOWLEDGMENT	- 62 -
BIBLIOΓΡΑΦΙΑ	- 63 -

Πίνακας εικόνων

ΕΙΚΟΝΑ 1. MOTOROLA DYNATAC	- 9 -
ΕΙΚΟΝΑ 2. IBM & BELLSOUTH SIMON	- 10 -
ΕΙΚΟΝΑ 3. ΝΟΚΙΑ 9110 OPEN	- 11 -
ΕΙΚΟΝΑ 4. ΔΟΜΗ ΕΝΟΣ ΔΙΚΤΥΟΥ GSM.	- 13 -
ΕΙΚΟΝΑ 5. SYMBIAN ΚΑΙ U'S.	- 16 -
ΕΙΚΟΝΑ 6. ΕΠΙΠΕΔΑ ΣΧΕΔΙΑΣΜΟΥ SYMBIAN	- 18 -
ΕΙΚΟΝΑ 7. ΤΟ ΠΡΩΤΟ IPHONE.	- 19 -
ΕΙΚΟΝΑ 8. ΟΙ ΕΦΑΡΜΟΓΕΣ ΠΟΛΛΑΠΛΑΣΙΑΖΟΝΤΑΙ ΜΕ ΡΑΓΔΑΙΟ ΡΥΘΜΟ ΜΕΣΑ ΣΤΟ APP STORE.	- 20 -
ΕΙΚΟΝΑ 9. Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ IPHONE.	- 21 -
ΕΙΚΟΝΑ 10. Ο ΠΡΩΤΟΣ ΕΜULATOR ΤΟΥ ANDROID ΘΥΜΙΖΕ ΠΟΛΥ ΚΛΩΝΟ BLACKBERRY ΚΑΙ ΒΑΣΙΖΟΤΑΝ ΣΕ ΕΝΑ ΠΡΩΤΟΤΥΠΟ ΜΟΝΤΕΛΟ ΤΗΣ HTC ΤΟ SOONER ΤΟ ΟΠΟΙΟ ΔΕΝ ΚΥΚΛΟΦΟΡΗΣΕ ΠΟΤΕ.	- 22 -
ΕΙΚΟΝΑ 11. ΑΡΙΣΤΕΡΑ ΤΟ HTC DREAM ΚΑΙ ΔΕΞΙΑ ΤΟ T-MOBILE G1. ΕΛΑΧΙΣΤΕΣ ΟΙ ΑΙΣΘΗΤΙΚΕΣ ΔΙΑΦΟΡΕΣ ΜΕΤΑΞΥ ΤΟΥΣ.	- 24 -
ΕΙΚΟΝΑ 12. ΟΙ ΕΚΔΟΣΕΙΣ ΤΩΝ NEXUS ΣΕ ΤΗΛΕΦΩΝΑ ΚΑΙ ΤΑΜΠΛΕΤΕΣ ΑΠΟ ΔΕΞΙΑ ΠΡΟΣ ΤΑ ΑΡΙΣΤΕΡΑ. ΜΕ ΤΑ ΧΡΟΝΙΑ ΟΧΙ ΜΟΝΟ Η ΥΠΟΛΟΓΙΣΤΙΚΗ ΔΥΝΑΜΗ ΑΛΛΑ ΚΑΙ ΤΟ ΜΕΓΕΘΟΣ ΤΗΣ ΣΥΣΚΕΥΗΣ ΑΥΞΑΝΕΤΑΙ.	- 25 -
ΕΙΚΟΝΑ 13. ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ANDROID.	- 26 -
ΕΙΚΟΝΑ 14. ACTIVITY LIFECYCLE MANAGER.	- 28 -
ΕΙΚΟΝΑ 15. ECLIPSE IDE.	- 33 -
ΕΙΚΟΝΑ 16. ΕΓΚΑΤΑΣΤΑΣΗ ADT PLUGIN.	- 34 -
ΕΙΚΟΝΑ 17. ANDROID SDK MANAGER.	- 35 -
ΕΙΚΟΝΑ 18. AVD MANAGER ΜΕ ΤΗΝ ΛΙΣΤΑ ΤΩΝ ΣΥΣΚΕΥΩΝ ΠΟΥ ΕΧΟΥΜΕ ΔΗΜΙΟΥΡΓΗΣΕΙ.	- 36 -
ΕΙΚΟΝΑ 19. ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΕΙΣ ΓΙΑ ΤΗΝ ΔΗΜΙΟΥΡΓΙΑ ΝΕΑΣ ΕΙΚΟΝΙΚΗΣ ΣΥΣΚΕΥΗΣ.	- 36 -
ΕΙΚΟΝΑ 20. Ο ΕΜULATOR ΜΑΣ ΤΡΕΧΕΙ.	- 37 -
ΕΙΚΟΝΑ 21. ΧΡΗΣΙΜΟΠΟΙΩΝΤΑΣ ΤΟΝ ΚΕΡΣΟΡΑ ΜΠΟΡΟΥΜΕ ΝΑ ΞΕΚΛΕΙΔΩΣΟΥΜΕ ΤΗΝ ΣΥΣΚΕΥΗ ΚΑΙ ΝΑ ΜΠΟΥΜΕ ΣΤΟ ΜΕΝΟΥ ΜΕ ΤΙΣ ΕΦΑΡΜΟΓΕΣ ΟΠΩΣ ΦΑΙΝΕΤΑΙ ΣΤΙΣ ΕΙΚΟΝΕΣ ΚΑΙ Ο ΕΜULATOR ΕΙΝΑΙ ΠΛΗΡΩΣ ΛΕΙΤΟΥΡΓΗΣΙΜΟΣ.	- 38 -
ΕΙΚΟΝΑ 22. ΣΕ ΚΑΠΟΙΕΣ ΠΕΡΙΠΤΩΣΕΙΣ ΕΧΟΥΜΕ ΚΑΙ ΠΙΟ ΡΕΑΛΙΣΤΙΚΟΥΣ ΕΜULATOR.	- 38 -
ΕΙΚΟΝΑ 23. ΔΗΜΙΟΥΡΓΙΑ ΕΝΟΣ ΝΕΟΥ ANDROID PROJECT.	- 39 -
ΕΙΚΟΝΑ 24. ΔΗΜΙΟΥΡΓΙΑ ΕΝΟΣ ΝΕΟΥ ANDROID PROJECT - 2.	- 40 -
ΕΙΚΟΝΑ 25. ΔΗΜΙΟΥΡΓΙΑ ΕΝΟΣ ΝΕΟΥ ANDROID PROJECT - 3.	- 40 -
ΕΙΚΟΝΑ 26. ΕΠΙΛΟΓΗ ΕΙΚΟΝΙΔΙΟΥ ΕΦΑΡΜΟΓΗΣ.	- 41 -
ΕΙΚΟΝΑ 27. ΔΗΜΙΟΥΡΓΙΑ ACTIVITY.	- 41 -
ΕΙΚΟΝΑ 28. ΔΗΜΙΟΥΡΓΙΑ ACTIVITY – 2.	- 42 -
ΕΙΚΟΝΑ 29. ΔΗΜΙΟΥΡΓΙΑ ACTIVITY – 3.	- 42 -
ΕΙΚΟΝΑ 30. ΔΗΜΙΟΥΡΓΙΑ ACTIVITY – 4.	- 43 -
ΕΙΚΟΝΑ 31. XML ΣΕ UI.	- 44 -
ΕΙΚΟΝΑ 32. Η ΠΡΩΤΗ ΜΑΣ ΕΦΑΡΜΟΓΗ ΤΡΕΧΕΙ ΣΤΟΝ ΕΜULATOR ΤΟΥ ANDROID.	- 45 -
ΕΙΚΟΝΑ 33. Η ΕΦΑΡΜΟΓΗ ΜΑΣ ΣΤΟΝ LAUNCHER.	- 46 -
ΕΙΚΟΝΑ 34. ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΩΝ TRANSFORMABLE BOUNDARY ARTIFACTS.	- 48 -
ΕΙΚΟΝΑ 35. TRANSFORMABLE BOUNDARY ARTIFACTS.	- 48 -
ΕΙΚΟΝΑ 36. WEB VERSION OF GUIDELINE MANAGEMENT TOOL WITH WIDGETS.	- 50 -
ΕΙΚΟΝΑ 37. RESULT WIDGETS.	- 51 -
ΕΙΚΟΝΑ 38. ΔΟΜΗ ΤΟΥ JSON ΩΣ OBJECT.	- 52 -
ΕΙΚΟΝΑ 39. ΔΟΜΗ ΤΟΥ JSON ΩΣ ARRAY.	- 52 -
ΕΙΚΟΝΑ 40. ΠΕΡΙΕΧΟΜΕΝΑ ΤΟΥ VALUE.	- 52 -
ΕΙΚΟΝΑ 41. COMMENT DISPLAY ON ANDROID.	- 54 -

ΕΙΚΟΝΑ 42. LOGIN SCREEN.	- 57 -
ΕΙΚΟΝΑ 43. ΕΠΙΛΟΓΗ DOCUMENT ΜΕΣΑ ΑΠΟ ΛΙΣΤΑ.	- 58 -
ΕΙΚΟΝΑ 44. ΕΠΙΛΟΓΗ ΚΑΡΤΕΛΑΣ ΚΑΙ ΣΤΗΝ ΣΥΝΕΧΕΙΑ ΑΡΘΡΟΥ.	- 59 -
ΕΙΚΟΝΑ 45. ΠΡΟΒΟΛΗ ΑΡΘΡΟΥ.	- 59 -
ΕΙΚΟΝΑ 46. ΠΡΟΒΟΛΗ SOCIAL WIDGETS.	- 60 -

1. Εισαγωγή.

Τα τελευταία χρόνια βάσει της ανάπτυξης της πληροφορικής δημιουργήθηκε η ανάγκη για γρηγορότερη και άμεση πληροφόρηση γεγονός που συνέβαλε στην εξέλιξη της κινητής τεχνολογίας και την δημιουργία έξυπνων κινητών συσκευών (smartphones). Οι εταιρείες αποφάσισαν να προσφέρουν στον χρήστη την ικανότητα για άμεση πρόσβαση στο Διαδίκτυο και την αλληλεπίδραση τους με άλλους χρήστες ενσωματώνοντας υπηρεσίες σε συσκευές που ήδη υπάρχουν στην καθημερινότητα μας. Σταδιακά άρχισαν να κυκλοφορούν στην αγορά κινητές συσκευές με χαρακτηριστικά όπως μεγαλύτερες οθόνες, πληκτρολόγιο qwerty, οθόνες αφής κλπ. που επέτρεψαν στον χρήστη να μπορεί να χρησιμοποιήσει το κινητό του τηλέφωνο αποτελεσματικότερα για να συμμετέχει σε μια συζήτηση, να μπορέσει γρήγορα και εύκολα να αποστείλει ένα e-mail ή ακόμα και να παρακολουθήσει κάποιο βίντεο από το Διαδίκτυο. Αυτό είχε σαν αποτέλεσμα την ραγδαία αύξηση των κινητών συσκευών που κυκλοφορούσαν στην αγορά και την στροφή των προγραμματιστών σε ένα νέο μεγάλο κομμάτι που ξεκίνησε να αναπτύσσεται.

Στο επόμενο κεφάλαιο θα ξεκινήσουμε με μια ιστορική αναδρομή στις κινητές συσκευές και στο mobile internet και θα ασχοληθούμε λίγο με την εμφάνιση των smartphones. Έπειτα θα δώσουμε κάποιες πληροφορίες για τα κινητά δίκτυα δεδομένων και μετά θα μιλήσουμε για το Mobile Application Development. Θα δούμε τα δημοφιλέστερα λειτουργικά συστήματα που υπάρχουν για κινητές συσκευές και τι χρειάζεται για να αναπτύξουμε εφαρμογές σε αυτά. Μετά θα επικεντρωθούμε στο Android το οποίο χρησιμοποιήσαμε για την κατασκευή της εφαρμογής μας. Θα δώσουμε ιστορικά στοιχεία για το πώς ξεκίνησε το Android και θα μάθουμε πώς να δημιουργούμε μια εφαρμογή από την αρχή και τέλος θα δούμε για ποιον σκοπό αναπτύχθηκε η δική μας εφαρμογή και τι τεχνικές χρησιμοποιήσαμε για να το πετύχουμε.

2. Κινητές συσκευές και Mobile Internet – ιστορική αναδρομή.

2.1 Εισαγωγή στις κινητές συσκευές.

Η δύναμη που μας δίνει μια κινητή συσκευή στον τομέα της επικοινωνίας έχει φτάσει σε εκπληκτικά επίπεδα τα τελευταία χρόνια . Μια μικρή συσκευή με απίστευτη υπολογιστική ισχύ που έχει την δυνατότητα να κρατάς ανά πάσα στιγμή στην παλάμη του χεριού σου. Έχει σταματήσει να είναι ένα απλό κινητό τηλέφωνο που χρησιμοποιούσαμε για να επικοινωνήσουμε και έχει μετατραπεί σε μια συσκευή ενημέρωσης, ψυχαγωγίας και ένα πολύτιμο εργαλείο καθημερινής εργασίας επιτρέποντας μας να εργαστούμε εν κινήσει και να γλιτώσουμε πολύτιμο χρόνο και να αυξήσουμε την αποδοτικότητα της εργασίας μας.

Η πρώτη κλήση από κινητό τηλέφωνο έγινε από ένα συσκευή ενσωματωμένη σε αυτοκίνητο στις 17 Ιουνίου 1946 στις ΗΠΑ. 10 χρόνια αργότερα το πρώτο αυτόματο σύστημα τηλεφωνικής κλήσης από αυτοκίνητο ευρείας παραγωγής λανσαρίστηκε στην Σουηδία. Ένας ερευνητής της Motorola, ο Δρ. Martin Cooper θεωρείται ο εφευρέτης της πρώτης αυτούσιας κινητής συσκευής και έκανε την πρώτη κλήση από αυτήν στις 3 Απριλίου του 1973. Η πρώτη συσκευή ευρείας παραγωγής έκανε την εμφάνιση της στις 6 Μαρτίου του 1983 και ήταν το Motorola DynaTAC (Dynamic Adaptive Total Area Coverage). Κόστιζε 3.995 δολάρια χρειαζόταν 10 ώρες φόρτιση και μπορούσε να πραγματοποιήσει κλήσεις για μισή ώρα. Η Motorola συνέχισε την ευρεία παραγωγή και συνέχισε τις ριζοσπαστικές και καινοτόμες ιδέες στο χώρο με άλλα μοντέλα όπως το MicroTAC και αργότερα το StarTAC.



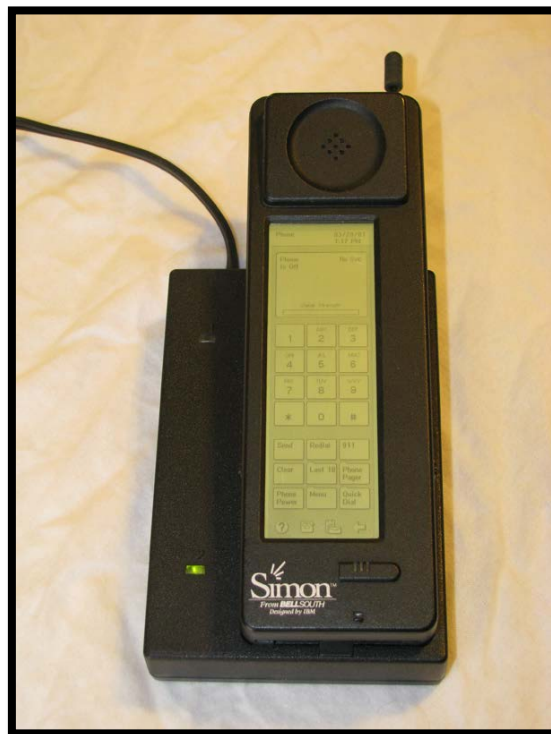
Εικόνα 1. Motorola DynaTAC

Το πρώτο δίκτυο κινητής τηλεφωνίας είχε ήδη κάνει την εμφάνιση του το 1979 στην Ιαπωνία από την NTT (Nippon Telegraph and Telephone) και ακολουθήθηκε το 1981 από την NMT (Nordic Mobile Telephone). Το 1983 παράλληλα με την κυκλοφορία του DynaTAC δημιουργήθηκε το 1^ο δίκτυο 1^{ης} γενιάς (1G) για να ακολουθήσουν το 1991 το 2^{ης} γενιάς (2G), το 2001 το 3^{ης} γενιάς (3G) και το 4^{ης} γενιάς (4G) το 2006 τα όποια έφεραν μαζί τους τεχνολογίες που μας επιτρέπουν την ανταλλαγή μεγαλύτερων πακέτων δεδομένων σε πολύ μεγαλύτερες ταχύτητες και έχουν βοηθήσει πάρα πολύ στην ανάπτυξη της τεχνολογίας των Smartphone. Θα αναφερθούμε εκτενέστερα στις τεχνολογίες αυτές παρακάτω.

2.2 Εμφάνιση έξυπνων κινητών συσκευών.

Μαζί με την κατασκευή του πρώτου κινητού τηλεφώνου από τον Cooper χρονολογούνται και οι πρώτες βάσεις για την ανάπτυξη του πρώτου Smartphone. Ο έλληνας Θεόδωρος Παρασκευάκος παρουσίασε ένα concept εισάγοντας σε μια κινητή συσκευή την ικανότητα έξυπνης συλλογής και επεξεργασίας δεδομένων μαζί με οθόνες γραφικών αναπαραστάσεων και κατοχύρωσε πατέντες για τραπεζική χρήση κινητού και πληρωμών μέσω της κινητής συσκευής. Είκοσι χρόνια πριν την πώληση της πρώτης έξυπνης κινητής συσκευής είχαν μπει τα θεμέλια για την ανάπτυξη μιας νέας αγοράς..

Το πρώτο Smartphone θεωρείται το Simon, ένα αποτέλεσμα μιας συνεργασίας της IBM και της BellSouth. Παρουσιάστηκε πρώτη φορά σε μια έκθεση στο Λας Βέγκας το 1992 και βγήκε στην αγορά το 1993. Το Simon ενσωμάτωνε λειτουργίες κινητού τηλεφώνου, βομβητή, PDA (Personal Digital Assistant) και φαξ. Επίσης το Simon εισήγαγε κάποιες πολύ πρωτοποριακές ιδέες στην αγορά ενσωματώνοντας λειτουργίες μέσω εφαρμογών όπως ημερολόγιο, αριθμομηχανή, σημειωματάριο κ.α. δεν είχε φυσικά κουμπιά αλλά οθόνη αφής και είχε την δυνατότητα αναβάθμισης προσθέτοντας εφαρμογές κατασκευασμένες από τρίτους (third party applications) στην εσωτερική μνήμη του τηλεφώνου. Χαρακτηριστικά και δυνατότητες που ενσωματώνονται σε όλα τα σύγχρονα Smartphone της αγοράς και θεωρούνται πλέον απαραίτητα. Παρ' όλα αυτά μόνο μια εφαρμογή κατασκευάστηκε για το Simon το Dispatchit από την PDA Dimensions και πωλήθηκε σε περιορισμένο αριθμό πελατών διότι η BellSouth ανακοίνωσε την απόσυρση του Simon λίγο αργότερα.



Εικόνα 2. IBM & BellSouth Simon

Το 1996 η Nokia κυκλοφόρησε την σειρά της “Nokia Communicator” ξεκινώντας με το Nokia 9000 που ήταν και το πρώτο Smartphone της εταιρείας. Η Nokia σε συνεργασία με την Hewlett Packard είχε φέρει στην αγορά μια συσκευή PDA και κινητού τηλεφώνου που χαρακτηριζόταν από πολύ κομψό σχεδιασμό και πολύ δυνατές λειτουργίες. Το λειτουργικό της σειράς των communicator ήταν βασισμένο στο GEOS V3.0 και προσέφερε επικοινωνία μέσω e-mail και text-based web browsing. Το 1998 κυκλοφόρησε το Nokia 9110 και 2 χρόνια μετά το Nokia 9110i.



Εικόνα 3. Nokia 9110 open

Το 1997 χρησιμοποιήθηκε για πρώτη φορά ο όρος Smartphone όταν η Ericsson παρουσίασε το Concept τηλέφωνο της GS88 το πρώτο που έφερε την ετικέτα του Smartphone. Έπειτα έγινε μια μεγάλη στροφή από τις εταιρείες προς την ανάπτυξη λειτουργικών συστημάτων για Smartphone και την κατασκευή ολοένα και περισσότερων συσκευών που ενσωματώνουν λειτουργίες μιας έξυπνης κινητής συσκευής όπως θα δούμε στο κεφάλαιο 3.

2.3. Κινητά δίκτυα δεδομένων.

Μεγάλο ρόλο στην ανάπτυξη των Smartphone και την διάδοση τους είχε και συνεχίζει να έχει η ανάπτυξη των δικτύων κινητών δεδομένων που μας επιτρέπουν να μπορούμε να έχουμε πρόσβαση στον ιστό και σε υπηρεσίες δεδομένων σε πολύ γρήγορες ταχύτητες από οποιοδήποτε σημείο και αν βρισκόμαστε. Η αρχή των Internet Phone έγινε από την AT&T το 1997 με την παρουσίαση του PocketNet ένα τηλέφωνο που κατασκεύασε η Pacific Communications Sciences Inc. το οποίο έμοιαζε με όλα τα τηλέφωνα της τότε εποχής με μια μικρή διαφορά, περιείχε ένα 19.2 Kbps μόντεμ το οποίο συνδεόταν μέσω του καινούριου δικτύου της AT&T το CDPD (Cellular Digital Packet Data). Η AT&T όμως εγκατέλειψε περίπου το 2000 τις προσπάθειες με τα δικά της πρωτόκολλα και δίκτυα και στράφηκε και αυτή στο πιο διαδεδομένο εκείνη την στιγμή πρότυπο το οποίο έβαλε τα θεμέλια για την μεγάλη άνθηση των κινητών δικτύων το GSM.

Το GSM (Global System for Mobile communications) ή (Group Special Mobile) όπως έγινε γνωστό αρχικά ήταν ένα πρότυπο το οποίο αναπτύχθηκε από το Ευρωπαϊκό Τηλεπικοινωνιακό Συμβούλιο ETSI (European Telecommunications Standards Institute). Το ETSI ξεκίνησε την μελέτη του GSM το 1982 με σκοπό να φέρει ένα ψηφιακό δίκτυο δεύτερης γενιάς να αντικαταστήσει τα αναλογικά δίκτυα πρώτης γενιάς που υπήρχαν. Το 1990 ανακοινώθηκε επίσημα το πρότυπο και τα χαρακτηριστικά του GSM. Το 1991 άρχισε η εμπορική διάθεση του GSM στην Ευρώπη ενώ στην Ελλάδα ήρθε το 1993. Το GSM είναι ένα κυψελοειδές ψηφιακό σύστημα κινητής τηλεφωνίας, το οποίο χρησιμοποιεί ηλεκτρομαγνητικά σήματα και την τεχνική πολλαπλής πρόσβασης με διαχωρισμό του διαθέσιμου φάσματος συχνοτήτων σε ένα αριθμό καναλιών και την διαίρεση αυτών σε χρονοθυρίδες για την μετάδοση σημάτων. Οι ζώνες συχνοτήτων που ορίστηκαν να χρησιμοποιούν τα δίκτυα GSM είναι 3:

- **GSM 900**

Η Διεθνής Ένωση Τηλεπικοινωνιών (ITU) παραχώρησε ένα ζεύγος συχνοτήτων, από τα 890 έως τα 915 MHz και από τα 935 έως τα 960 MHz. Η πρώτη περιοχή χρησιμοποιείται για την επικοινωνία του κινητού με τον σταθμό βάσης (Up link), ενώ η δεύτερη για την επικοινωνία του σταθμού βάσης με το κινητό (down link). Οι περιοχές (ζώνες) των 25MHz υποδιαιρούνται η καθεμία σε 124 + (1 ελεύθερο) κανάλια συχνότητας και κάθε κανάλι έχει εύρος ζώνης 200 KHz. Όλο αυτό το σύστημα ονομάστηκε GSM 900 ή Standard GSM.

- **GSM 1800**

Στη συνέχεια, το 1991, αναπτύχθηκε το σύστημα DCS 1800, στο οποίο διατηρείται η δομή ενός GSM 900 δικτύου αλλά χρησιμοποιούνται διαφορετικά ζεύγη συχνοτήτων, από τα 1710 έως τα 1785 MHz Up link και από τα 1805 έως τα 1880 MHz Down link. Οι περιοχές των 75MHz υποδιαιρούνται η καθεμία σε 374 (+ 1 ελεύθερο) κανάλια και κάθε κανάλι έχει εύρος ζώνης 200 KHz. Αυτή η αλλαγή στην ζώνη συχνοτήτων έγινε διότι οι ζώνες του GSM 900 στην Ευρώπη ήταν πιασμένες από άλλους πάροχους κινητής τηλεφωνίας. Σήμερα, όλες οι εταιρίες κινητής τηλεφωνίας χρησιμοποιούν και τα δύο συστήματα(GSM 900/GSM 1800) στα δίκτυα τους αυξάνοντας αισθητά τη χωρητικότητά στα δίκτυα τους. Στα τέλη δεκαετίας του 1990 η GSM World Association αποφάσισε να μετονομάσει το DCS 1800 σε GSM 1800 για να φανεί η δυναμικότητα και η παγκοσμιότητα του GSM.

- **GSM 1900**

Στο GSM 1900 τέλος το οποίο χρησιμοποιείται σε αρκετές χώρες της Αμερικής, διατηρείται και πάλι η δομή ενός GSM 900 δικτύου, αλλά χρησιμοποιούνται και εδώ διαφορετικά ζεύγη συχνοτήτων: Από τα 1850 έως τα 1910 MHz για Up link και από τα 1930 έως τα 1990 MHz για Down link. Οι περιοχές των 60MHz υποδιαιρούνται η καθεμία σε 299+ (1 ελεύθερο) κανάλια συχνότητας και κάθε κανάλι έχει εύρος ζώνης 200KHz.

Για την ανάπτυξη των δικτύων GSM με σκοπό την αύξηση χωρητικότητας του δικτύου προσφέροντας περισσότερες και ποιοτικότερες υπηρεσίες και την ανάπτυξη του mobile internet εμφανίστηκαν 2 νέες τεχνολογίες:

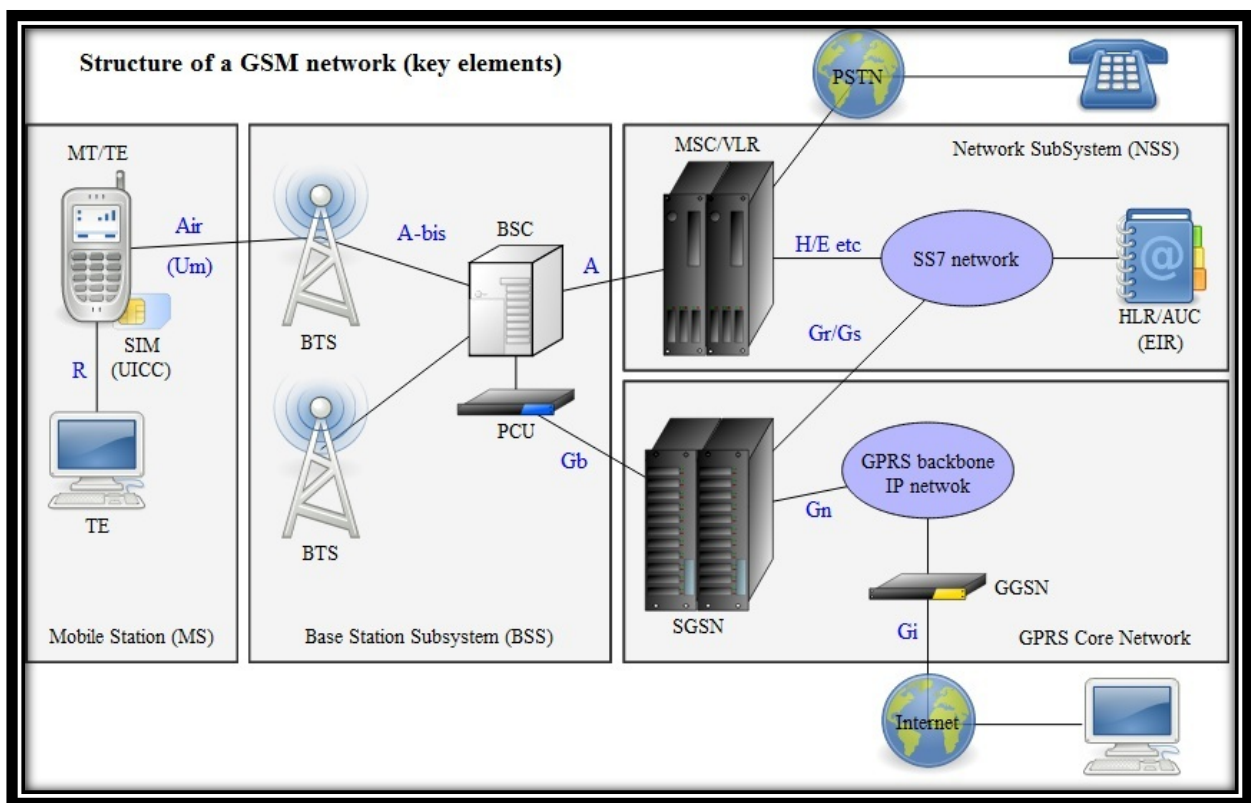
- **Τεχνολογία GPRS - (General Packet Radio Service)**

Το GPRS (General Packet Radio Service) γενικά είναι το τεχνολογικό πρότυπο που επιτρέπει την ταχύτερη αποστολή και λήψη δεδομένων μέσω των δικτύων κινητής τηλεφωνίας GSM μέσω της τεχνολογίας μεταγωγής πακέτων. Το GPRS επιτρέπει την ταυτόχρονη χρήση περισσοτέρων του ενός χρονοθυρίδων έτσι η μεταφορά δεδομένων μπορεί να φτάσει θεωρητικά ως και τα $153,6 = 16 \times 9.6$ kbps για 16 χρονοθυρίδες ή $21,4 \times 8 = 171,2$ kbps για 8 χρονοθυρίδες. Οι πόροι του δικτύου χρησιμοποιούνται πιο αποδοτικά γιατί οι χρονοθυρίδες δεσμεύονται μόνο κατά την ώρα μετάδοσης και αποδεσμεύονται όταν τελειώνει η μετάδοση. Στην πράξη όμως το GPRS χρησιμοποιεί 3 με 4 χρονοθυρίδες για κατέβασμα και μια χρονοθυρίδα για ανέβασμα (μη συμμετρική σύνδεση).

- **Τεχνολογία EDGE - (Enhanced Data rates for GSM Evolution)**

Το EDGE είναι μια ενδιάμεση μεταβατική τεχνολογία πριν το 3G και αυτό είναι τεχνολογικό πρότυπο που επιτρέπει στα δίκτυα 2G να έχουν τριπλάσια χωρητικότητα δικτύου με πολύ υψηλές ταχύτητες μετάδοσης για την παροχή υπηρεσιών 3G, όπως video streaming, πραγματικό Internet browsing κτλ.. Το EDGE είναι μια αναβάθμιση του GPRS αλλά δεν μπορεί να λειτουργήσει αυτόνομα ενώ η αναβάθμιση και η εγκατάσταση του EDGE δεν απαιτεί την χρήση νέου εξοπλισμού από τις εταιρίες κινητής τηλεφωνίας αλλά την βελτίωση του ήδη υπάρχοντος.

Βασικό πλεονέκτημα της τεχνολογίας EDGE σε σχέση με το ήδη υπάρχον GSM δίκτυο, είναι η χρήση μίας διαφορετικής μεθόδου διαμόρφωσης των δεδομένων. Η μέθοδος αυτή ονομάζεται 8PSK (8 Phase Shift Keying modulation) επιτρέποντας τη μεταφορά 3 bit δεδομένων σε κάθε μοναδικό παλμό του δικτύου. Η τεχνολογία που παρέχουν τα απλά δίκτυα GSM με υποστήριξη υπηρεσιών GPRS, χρησιμοποιεί τη μέθοδο GMSK (Gaussian pre-filtered Minimum Shift Keying) η οποία βασίζεται στη μέθοδο Gauss για την εκθετική μείωση των πιθανοτήτων λάθους κατά τη μεταφορά των δεδομένων, αλλά επιτρέπει τη μεταφορά μόνο ενός bit δεδομένων σε κάθε μοναδικό παλμό του δικτύου. Οι ταχύτητες που επιτυγχάνονται είναι 384Kbps ή και 768kbps με στόχο όμως να φτάσει τα 2Mbps. Επίσης το EDGE έχει την ικανότητα αναμετάδοσης ενός πακέτου πληροφοριών, που δεν κωδικοποιήθηκε σωστά, με ένα περισσότερο ισχυρό σχήμα κωδικοποίησης, ενώ στο GPRS τα πακέτα θα έπρεπε να αποστέλλονται με το ίδιο σχήμα κωδικοποίησης ακόμη και αν το περιβάλλον μεταβάλλεται με αποτέλεσμα τις αποσυνδέσεις και τα προβλήματα, ιδιαίτερα σε περιοχές με αυξημένη ζήτηση.



Εικόνα 4.Δομή ενός δικτύου GSM.

3. Mobile Application Development.

3.1 Εισαγωγή στο mobile application development.

Το mobile application development (ανάπτυξη εφαρμογών για κινητές συσκευές, θα χρησιμοποιήσουμε και τους δύο όρους καθώς τους συναντάμε πολύ συχνά στην ελληνική ορολογία) περιγράφει την διαδικασία ανάπτυξης εφαρμογών για χαμηλής κατανάλωσης φορητές συσκευές όπως είναι τα PDA και οι έξυπνες κινητές συσκευές. Αυτές οι εφαρμογές μπορούν είτε να βρίσκονται προεγκατεστημένες στη συσκευή από τον κατασκευαστή είτε να γίνεται λήψη στην συσκευή και εγκατάσταση από τον χρήστη ώστε να βοηθάει στην παραμετροποίηση της κάθε συσκευής στις ανάγκες του χρήστη. Τα τελευταία χρόνια το mobile application development έχει εξελιχθεί σε ένα καινούριο κλάδο στον τομέα της πληροφορικής. Υπάρχουν εταιρείες που έχουν δημιουργηθεί και ασχολούνται αποκλειστικά με αυτό όπως επίσης και πάρα πολλές μεγάλες εταιρείες στον χώρο έχουν αναγκαστεί να δημιουργήσουν τμήματα τα οποία φροντίζουν για το κομμάτι της εταιρείας όσον αφορά mobile συσκευές. Ένας κλάδος που έχει αποδειχθεί εκτός των άλλων και αρκετά επικερδής λόγω της ραγδαίας ανάπτυξης που έχει γνωρίσει τα τελευταία χρόνια. Συμβολή σε αυτό έχει και το γεγονός ότι το mobile gaming έχει γνωρίσει μεγάλη άνθηση με πολλές εταιρείες να έχουν στραφεί αποκλειστικά στην ανάπτυξη του και να βλέπουν το μέλλον εκεί. Αυτό σπρώχνει τις εταιρείες κατασκευής κινητών να δημιουργήσουν συσκευές με μεγαλύτερη επεξεργαστική ισχύ ώστε να βοηθήσουν στην ανάπτυξη όλο και πιο δυνατότερων εφαρμογών και φυσικά να φροντίσουν για την συνεχή βελτιστοποίηση των λειτουργικών συστημάτων που συνοδεύουν αυτές τις συσκευές.

Για την ανάπτυξη εφαρμογών σε έξυπνες κινητές συσκευές χρειάζεται να δουλέψουμε στην κάθε πλατφόρμα της εκάστοτε εταιρείας όπως θα δούμε παρακάτω για τα διαφορετικά λειτουργικά συστήματα που υπάρχουν. Για να γίνει αυτό θα χρειαστούμε κάποια εργαλεία. Το σύνολο των Εργαλείων Ανάπτυξης Λογισμικού που μας παρέχεται για να μπορέσουμε να δημιουργήσουμε μια εφαρμογή σε ένα συγκεκριμένο λειτουργικό σύστημα ονομάζεται SDK (Software Development Kit). Συνήθως οι εταιρείες παρέχουν τα SDK δωρεάν για να ενθαρρύνουν τους χρήστες να αναπτύξουν εφαρμογές για τα δικά τους λειτουργικά συστήματα. Κάθε SDK συνοδεύεται από μια Διεπαφή ή Διασύνδεση Προγραμματισμού Εφαρμογών γνωστή και ως API (Application Programming Interface). Το API είναι ένα ολοκληρωμένο σύνολο από ρουτίνες, πρωτόκολλα και εργαλεία τα οποία πρέπει να χρησιμοποιήσει ο προγραμματιστής για να έχει την διασύνδεση μεταξύ της εφαρμογής και του συστήματος. Περιλαμβάνει όλες τις κλήσεις συστήματος και τις διαχειρίσεις των απαντήσεων. Συνήθως έχει την μορφή μια βιβλιοθήκης που μας παρέχει τις οδηγίες για να κατασκευάσουμε όλα αυτά που χρειαζόμαστε. Εκτός από το SDK και το API αυτό που χρειαζόμαστε είναι ένα IDE (Integrated Development Environment) το οποίο είναι μια ολοκληρωμένη εφαρμογή λογισμικού για να μπορέσουμε να διαχειριστούμε το SDK που πήραμε και να αναπτύξουμε την δική μας εφαρμογή. Το IDE αποτελείται συνήθως από τα εξής:

- *source code editor*
- *build automation tools*
- *debugger*

Το source code editor μας επιτρέπει να γράψουμε κατευθείαν τον κώδικα για την εφαρμογή, τα build automation tools μας βοηθούν να εξοικονομήσουμε χρόνο κάνοντας αυτοματοποιημένα κάποιες απαραίτητες διαδικασίες για να μπορεί να μετατραπεί ο κώδικας που έχουμε γράψει στην εφαρμογή που θέλουμε να φτιάξουμε και τέλος ο debugger μας βοηθάει να εντοπίσουμε τυχόν λάθη που έχουμε κάνει στον κώδικα μας. Συνήθως ένα IDE χρησιμοποιείται για μια γλώσσα προγραμματισμού όμως υπάρχουν και κάποια που μπορούμε να τα χρησιμοποιήσουμε για διαφορετικές γλώσσες προγραμματισμού. Μερικά από τα πιο γνωστά που θα αναφερθούμε και παρακάτω είναι τα Eclipse, NetBeans, Microsoft Visual Studio και Xcode.

3.2 Λειτουργικά συστήματα για έξυπνες κινητές συσκευές.

Όλες οι μεγάλες εταιρείες όπως αναφέραμε παραπάνω στράφηκαν στην ανάπτυξη διαφορετικών λογισμικών για κινητές συσκευές προσφέροντας η κάθε μια την δικιά της πρόταση για συσκευή που συνοδευόταν από το δικό της λειτουργικό σύστημα. Έτσι διάφορα λειτουργικά συστήματα εμφανίστηκαν από τις εταιρείες όπως το Symbian OS που ξεκίνησε σαν μια συνεργασία από μεγάλες εταιρείες και κατέληξε αποκλειστικά στην Nokia, το Palm OS και το BlackBerry OS που χρησιμοποιήθηκαν από την RIM, το Windows CE κα. Όλα άλλαξαν το 2007 όταν η Apple ανακοίνωσε την είσοδο της στην αγορά των κινητών συσκευών με το διάσημο iPhone και το iOS που το συνόδευε και η Google διέθεσε το Android το οποίο υιοθέτησαν μεγάλες εταιρείες δίνοντας έτσι στον χρήστη μια καλύτερη εικόνα για το τι θα πρέπει να θεωρούμε smartphone ανεβάζοντας πάρα πολύ ψηλά τον πήχη για τις υπόλοιπες εταιρείες. Τα δύο αυτά λειτουργικά συστήματα έφεραν την επανάσταση στην αγορά και καθιέρωσαν ως επικρατέστερες τις συσκευές με οθόνη αφής δίνοντας στον χρήστη μεγάλη ταχύτητα και ευχρηστία και έχουν φτάσει να κατέχουν το 79% του μεριδίου της αγοράς. Οι υπόλοιπες εταιρείες μετά την μεγάλη επιτυχία αυτών των δύο λογισμικών έχουν στρέψει την πολιτική τους και τις τεχνικές υλοποίησης ώστε να προσπαθήσουν να τα ακολουθήσουν. Παρακάτω θα κάνουμε μια εκτενής αναφορά σε μερικά από μεγαλύτερα λειτουργικά συστήματα της αγοράς για το πώς ξεκίνησαν και αναπτύχθηκαν και θα επικεντρωθούμε στο Android το οποίο αυτή την στιγμή είναι το κυρίαρχο της αγοράς και έχει υιοθετηθεί από πάρα πολλές μεγάλες εταιρείες στον χώρο.

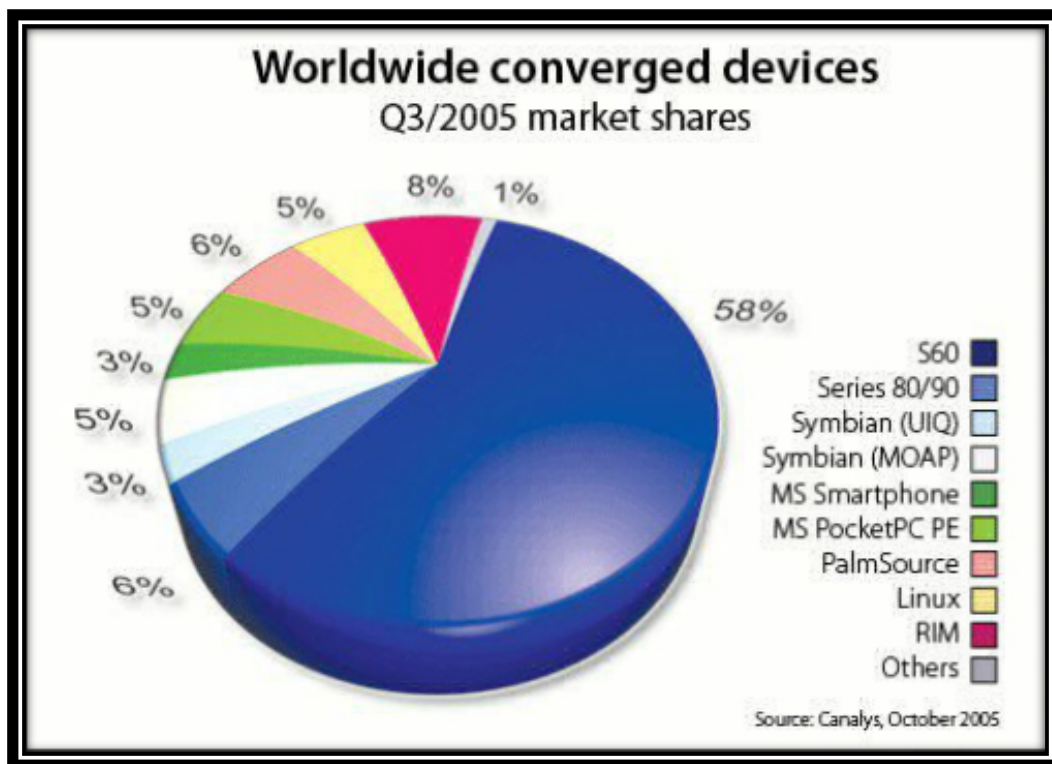
World-Wide Smartphone Sales							
Quarter	Android	iOS	Symbian	RIM	Bada	Windows Phone	Other
2012 Q1	56.1%	22.9%	8.6%	6.9%	2.7%	1.9%	0.9%
2011 Q4	50.9%	23.9%	11.7%	8.8%	2.1%	1.9%	0.8%
2011 Q3	52.5%	15.0%	16.9%	11.0%	2.2%	1.5%	0.9%
2011 Q2	43.4%	18.2%	22.1%	11.7%	1.9%	1.6%	1.0%
2011 Q1	36.0%	16.8%	27.4%	12.9%	1.7%	3.6%	1.6%
2010 Q4	31.1%	16.1%	32.9%	13.1%	1.3%	3.4%	2.2%
2010 Q3	25.3%	16.6%	36.3%	15.4%	1.1%	2.8%	2.5%
2010 Q2	17.2%	14.2%	41.2%	18.2%	0.9%	5.0%	3.3%
2010 Q1	9.6%	15.3%	44.2%	19.7%		6.8%	4.4%
2009 Q4	7.6%	16.2%	44.7%	19.7%		7.9%	4.0%
2009 Q3	3.4%	17.0%	44.2%	20.5%		7.9%	7.0%
2009 Q2	1.8%	13.0%	51.0%	19.0%		9.3%	5.9%
2009 Q1	1.6%	10.5%	48.8%	20.6%		10.2%	8.2%
2008 Q4	1.1%	10.6%	46.5%	19.3%		12.2%	9.1%
2008 Q3	0.6%	13.1%	50.3%	16.1%		11.2%	9.8%
2008 Q2		2.8%	57.5%	17.5%		12.1%	10.8%
2008 Q1		4.6%	49.5%	11.6%		10.4%	11.6%
2007 Q4		5.2%	62.3%	10.9%		11.9%	9.6%
2007 Q3		3.4%	63.1%	9.7%		12.8%	11.5%
2007 Q2		1.0%	65.6%	8.9%		11.5%	13.0%
2007 Q1			61.2%	8.7%		13.4%	16.8%

Πίνακας 1. Μεριδίο αγοράς λειτουργικών συστημάτων (mobile data usage).

3.2.1 Symbian OS.

Το 2000 η Ericsson κυκλοφόρησε το R380, το πρώτο κινητό που έτρεχε σε λειτουργικό σύστημα Symbian και το πρώτο εμπορικό τηλέφωνο της αγοράς που χαρακτηρίστηκε ως Smartphone. Το Symbian όμως έχει τις ρίζες του πολύ πιο παλιά μέχρι να φτάσουμε στο 2000. Στα τέλη της δεκαετίας του 80 και στις αρχές του 90 το λειτουργικό σύστημα EPOC16 είχε κατασκευαστεί από την εταιρεία Psion για τις SIBO (Sixteen Bit Organizers) συσκευές της εταιρείας. Οι συσκευές αυτές είχαν έναν 8086 επεξεργαστή της Intel και μια 16-bit αρχιτεκτονική. Το λειτουργικό ήταν γραμμένο σε γλώσσα μηχανής για τον συγκεκριμένο επεξεργαστή και σε C και σχεδιασμένο για να διανέμεται σε ROM. Υποστήριζε μια απλή γλώσσα προγραμματισμού την OPL (Open Programming Language) και ένα ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών το OVAL. Το 1997 η εταιρεία προχώρησε ακόμη παρακάτω παρουσιάζοντας το EPOC32 το οποίο ήταν γραμμένο σε αντικειμενοστραφής γλώσσα προγραμματισμού την C++. Το R380 της Ericsson έτρεχε στην 5^η έκδοση του EPOC32. Το 1998 η Psion Software μετονομάστηκε σε Symbian Ltd. και η 6^η έκδοση του EPOC έγινε γνωστή ως Symbian OS με την συνεργασία της εταιρείας με τις Nokia, NTT DoCoMo, και Ericsson.

Πάνω από 250 εκατομμύρια συσκευές με το Symbian OS κυκλοφόρησαν τα χρόνια που ακολούθησαν από τις Nokia, Siemens, Samsung, Ericsson, Sony Ericsson, Motorola και Panasonic. Κάθε εταιρεία είχε αναπτύξει το δικό της user interface το οποίο έτρεχε επάνω στο Symbian OS με την Nokia να έχει το μεγαλύτερο μερίδιο με το Series 60, την Sony Ericsson με το UIQ και την NTT DoCoMo με το MOAP και το OPP.



Εικόνα 5. Symbian και UI's.

Το 2009 οι εταιρείες Nokia, Sony Ericsson, NTT DoCoMo, Motorola, Texas Instruments, Vodafone, LG Electronics, Samsung Electronics, STMicroelectronics and AT&T ίδρυσαν την Symbian Foundation, ένα μη κερδοσκοπικό οργανισμό για να διαθέσουν την πλατφόρμα Symbian ως λογισμικό ανοιχτού κώδικα (open source). Το Symbian foundation δεν είχε ως αντικείμενο την εξέλιξη του λογισμικού αλλά τα παρακάτω:

- *platform development kits and tools*
- *documentation and example code*
- *discussion forums and mailing lists*
- *application signing (Symbian Signed)*
- *application distribution (Symbian Horizon)*
- *idea gathering and feedback (Symbian Ideas)*
- *an annual conference (Symbian Exchange and Exposition, abbreviated "SEE")*

Λόγω αλλαγής πολιτικής η LG Electronics και η Motorola άφησαν γρήγορα τον οργανισμό λίγο μετά την δημιουργία του. Αντικαταστάθηκαν από τις Fujitsu και Qualcomm Innovation Center. Το Νοέμβριο του 2010 το Symbian Foundation ανέστειλε την λειτουργία του λόγω της ταχύτατης εξέλιξης της παγκόσμιας οικονομίας και των αλλαγών στις παγκόσμιες αγορές όπως αναρτήθηκε στο site. Έπειτα η Nokia πήρε εξ' ολόκληρου την διαχείριση της πλατφόρμας στα χέρια της και έκτοτε είναι αποκλειστικά υπεύθυνη για την εξέλιξη της.

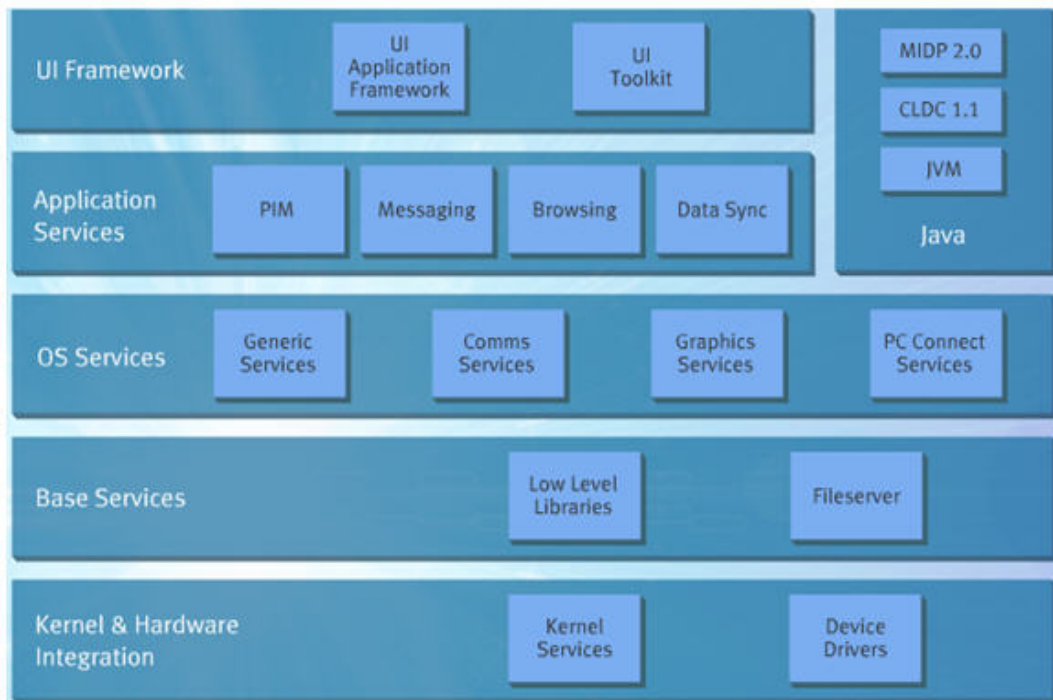
Το Symbian OS δημιουργήθηκε διατηρώντας 3 αρχές που είχαν στο μυαλό τους οι προγραμματιστές:

1. Η ακεραιότητα και ασφάλεια των δεδομένων των χρηστών είναι υψίστης σημασίας
2. Δεν πρέπει να σπαταλιέται χρόνος χρήσης
3. Όλοι οι πόροι σπανίζουν.

Για να το επιτύχει αυτό το Symbian χρησιμοποιεί ένα microkernel και τεχνικές προσέγγισης των services μέσω αιτήσεων και απαντήσεων. Έτσι πετυχαίνει τον διαχωρισμό μεταξύ του User Interface και της μηχανής και την εξοικονόμηση ενέργειας βάζοντας τον επεξεργαστή σε μια χαμηλή ισχύος λειτουργία όταν δεν αλληλεπιδρά με κάποια εφαρμογή. Αυτό επιτυγχάνεται με μια τεχνική η οποία ονομάζεται active objects. Η τεχνική των active objects έχει να κάνει με το πόσο καλά διαχειρίζεται το λειτουργικό, τους πόρους του συστήματος για τις εφαρμογές που τρέχουν στο προσκήνιο και για αυτές που τρέχουν στο παρασκήνιο. Το Symbian σχεδιάστηκε για συσκευές που δεν χρησιμοποιούν μεγάλη ενέργεια και για ROM-based συστήματα. Οι εφαρμογές και το λειτουργικό χρησιμοποιούν ένα αντικειμενοστραφές μοντέλο σχεδιασμού: Model-View-Controller (MVC Μοντέλο-Όψη-Χρήστης). Σε μεταγενέστερες εκδόσεις του λειτουργικού λόγω ανταγωνισμού παρουσιάστηκε μια διαφορετική προσέγγιση με ένα πυρήνα πραγματικού χρόνου και ένα μοντέλο ασφαλής πλατφόρμας στο Symbian 8 και 9.

Το μοντέλο σχεδιασμού του Symbian περιλαμβάνει τα παρακάτω στρώματα στον σχεδιασμό του:

- *UI Framework Layer*
- *Application Services Layer*
 - *JAVA ME*
- *OS Services Layer*
 - *Generic OS services*
 - *Communications services*
 - *Multimedia and graphics services*
 - *Connectivity services*
- *Base Services Layer*
- *Kernel Services & Hardware Interface Layer*



Εικόνα 6. Επίπεδα σχεδιασμού Symbian

Το Base Services Layer είναι το χαμηλότερο στρώμα που μπορεί να έχει πρόσβαση ο χρήστης, από κάτω στο Kernel Services & Hardware Interface Layer βρίσκεται ο πυρήνας και το hardware, στρώμα το οποίο αναλαμβάνει την διαχείριση των πόρων του συστήματος για να υπάρχει σωστή ροή των δεδομένων και των διεργασιών. Στα πιο πάνω στρώματα συναντάμε όλες τις βασικές υπηρεσίες που ενσωματώνει το OS, τις υπηρεσίες που έρχονται μέσω εφαρμογών και στο πάνω επίπεδο συναντάμε την εργαλειοθήκη ανάπτυξης των διαφορετικών User Interface όπως μιλήσαμε παραπάνω (Series 60, UIQ). Εδώ φαίνεται ότι στην διαμόρφωση του UI που έχει κάνει κάθε εταιρεία δεν έχει να κάνει με τα βασικά κομμάτια σχεδιασμού του Symbian γιατί είναι στο πάνω επίπεδο της αρχιτεκτονικής.

Για την ανάπτυξη εφαρμογών το SDK του Symbian είναι γραμμένο σε αντικειμενοστραφή γλώσσα προγραμματισμού C++ χρησιμοποιώντας το Qt framework. Το Qt είναι ένα framework που χρησιμοποιείται σε διαφορετικές πλατφόρμες εφαρμογών και μας βοηθάει να ξεκινήσουμε έχοντας μια στάνταρ δομή στην εφαρμογή μας. Στις παλιές εκδόσεις δημοφιλής IDE για το Symbian ήταν ο CodeWarrior ο οποίος το 2006 αντικαταστάθηκε από τον Carbide.c++ ένα IDE βασισμένο στο Eclipse. Ο Carbide.c++ προσφέρεται σε 4 εκδόσεις Express, Developer, Professional και OEM. Με την έκδοση Express η οποία διατίθεται δωρεάν μπορεί να δημιουργηθεί μια ολοκληρωμένη εφαρμογή για Symbian. Σε άλλες εκδόσεις μπορούμε να έχουμε έξτρα χαρακτηριστικά όπως UI design, crash debugging και άλλα. Εναλλακτικές λύσεις για την ανάπτυξη εφαρμογών είναι το Qt Creator όπως επίσης και το Microsoft Visual Studio 2003 και 2005 με την χρήση του plugin Carbide.vs. Μια τελειοποιημένη εφαρμογή για Symbian γίνεται ένα πακετάκι σε ένα .sis αρχείο και μπορεί να εγκατασταθεί στην συσκευή over-the-air, μέσω Bluetooth, συνδέοντας την συσκευή στο pc ή και μέσω κάρτας μνήμης. Στις τελευταίες εκδόσεις του Symbian για να μπορεί μια εφαρμογή να έχει πρόσβαση σε όλες τις δυνατότητες που παρέχει η συσκευή πρέπει να είναι Symbian Signed κάτι το οποίο τώρα μπορεί να γίνει πλέον δωρεάν. Για τον σχεδιασμό εφαρμογών μπορούμε επίσης να χρησιμοποιήσουμε και άλλες γλώσσες προγραμματισμού όπως Python, Java ME, Flash Lite, Ruby, .NET, Web Runtime και C. Σε μια εφαρμογή Java έχουμε ένα .jar ή και το οποίο εγκαθίσταται στην συσκευή και τρέχει κανονικά. Στην περίπτωση που θέλουμε να τρέξουμε κώδικα Python μπορούμε να εγκαταστήσουμε τον μεταγλωττιστή Python for S60 και μετά μπορούμε και να τρέξουμε και να γράψουμε κώδικα κατευθείαν από την συσκευή.

3.2.2 BlackBerry OS.

Το BlackBerry OS αναπτύχθηκε από την εταιρεία RIM (Research In Motion) για να συνοδεύσει την σειρά BlackBerry των έξυπνων κινητών της εταιρείας. Οι πρώτες συσκευές προσέφεραν στον χρήστη μια πρωτόγνωρη εμπειρία όσον αφορά την χρήση του κινητού ενσωματώνοντας στην συσκευή trackwheel ή scroll wheel και trackball συσκευές εισαγωγής που μέχρι τότε είχαμε συναντήσει μόνο σε υπολογιστή. Σε μεταγενέστερα μοντέλα η RIM ενσωμάτωσε και touchpad ή trackpad και οθόνη αφής. Επίσης η RIM ήταν η πρώτη εταιρεία η οποία προσέφερε σε όλα της τα μοντέλα φυσικό QWERTY διάταξης πληκτρολόγιο, λόγος για τον οποίο έγινε πολύ γνωστή και τύχαινε μεγάλης προτίμησης στους επαγγελματίες χρήστες. Σε τελευταία μοντέλα η RIM έχει αρχίσει να ενσωματώνει μεγάλες οθόνες αφής χωρίς φυσικό πληκτρολόγιο ακολουθώντας την τάση σχεδιασμού των άλλων μεγάλων εταιρειών.

3.2.3 Apple iOS και iPhone.

Τον Ιούνιο του 2007 μια καινούρια πλατφόρμα παράλληλα με μια καινούρια έξυπνη κινητή συσκευή έκανε την εμφάνιση της και αναμενόταν να ταραξεί τα νερά στην αγορά. Ο λόγος για το iPhone της Apple που συνοδευόταν από το λειτουργικό iOS. Η Apple λάνσαρε το νέο της λειτουργικό αποκλειστικά για να συνοδεύει την συσκευή της ενώ αργότερα επέκτεινε την χρήση του και σε άλλες συσκευές της εταιρείας (iPod, iPad κτλ). Το iPhone βασιζόταν αποκλειστικά σε οθόνη πολλαπλής αφής χρησιμοποιώντας ελάχιστα φυσικά κουμπιά και άνοιξε τον δρόμο για μια τεράστια αλλαγή στην παραγωγή έξυπνων κινητών τηλεφώνων που κατευθύνθηκε στο να επωφεληθεί ο χρήστης από την αύξηση μεγέθους της οθόνης και τι μπορεί να αναπτυχθεί πάνω σε αυτή και να περιοριστούν οι χρήσεις των φυσικών κουμπιών. Το iOS ήταν πολύ φιλικό και εύχρηστο προσφέροντας την δυνατότητα στον χρήστη μόνο με τα δάχτυλα του να μπορεί να διαβάσει ένα κείμενο κάνοντας κύλιση στην οθόνη, να μεγεθύνει μια φωτογραφία πάνω στην οθόνη ακόμα και να παρακολουθήσει ένα βίντεο σε μεγαλύτερο μέγεθος περιστρέφοντας την οθόνη.



Εικόνα 7. Το πρώτο iPhone.

Βασική λογική της λειτουργίας iOS ήταν ότι για οτιδήποτε ενέργεια θελήσει να κάνει ο χρήστης θα χρειαστεί να τρέξει μια εφαρμογή. Τον Φεβρουάριο του 2008 λοιπόν η Apple ανακοίνωσε και λίγους μήνες αργότερα έδωσε στην κυκλοφορία το iPhone SDK (μετέπειτα ονομάστηκε σε iOS SDK) και εισήγαγε τα «third party applications» (τα αναφέραμε πιο πάνω) στην συσκευή. Τον Ιούλιο του ίδιου έτους η εταιρεία για να μπορεί να ελέγξει και να βάλει περιορισμούς στις εφαρμογές έκανε διαθέσιμο ένα ηλεκτρονικό κατάστημα στην συσκευή κάθε χρήστη το περίφημο «App Store».

Κάθε εφαρμογή λοιπόν μπορούσε να γίνει διαθέσιμη στους χρήστες αποκλειστικά και μόνο μέσω του App Store αφού πρώτα είχε περάσει από τον έλεγχο και όλους τους περιορισμούς..



Εικόνα 8. Οι εφαρμογές πολλαπλασιάζονται με ραγδαίο ρυθμό μέσα στο App Store.

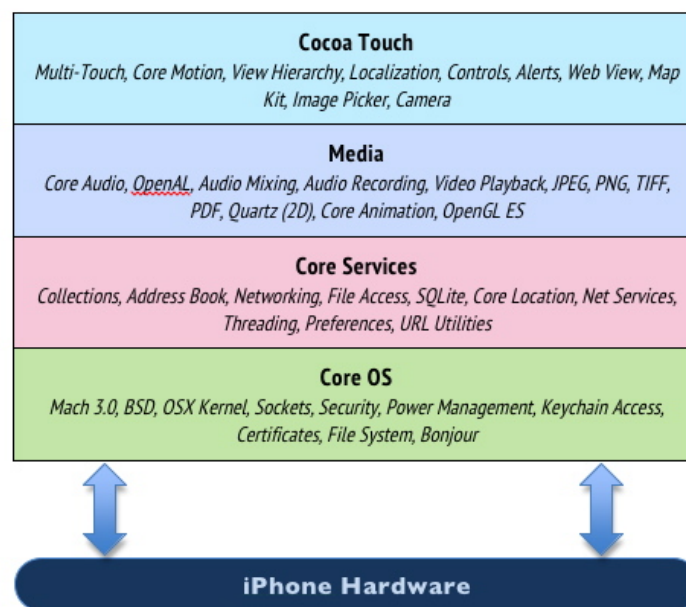
Με αυτή της την ενέργεια η Apple μπόρεσε να ελέγξει πολύ καλά τι θα διακινηθεί στις συσκευές και να έχει ένα πολύ καλό έλεγχο ασφαλείας που υποσχόταν και στον τελικό χρήστη που θα προτιμούσε κάποια από τις συσκευές της. Από την άλλη οι developers βρήκαν μια μεγάλη αγορά και άρχισε μια μεγάλη ανάπτυξη για το Mobile Application Development με την Apple να βάζει πάντα τους περιορισμούς της. Έτσι λοιπόν για να μπορέσει κάποιος να αναπτύξει μια εφαρμογή σε iOS θα πρέπει να έχει στην κατοχή του ένα υπολογιστή της εταιρείας που να τρέχει Mac OS για να μπορεί να εκμεταλλεύεται το iOS SDK το οποίο προσφέρει έναν εικονικό Simulator για την δοκιμή της εφαρμογής. Παρ' όλα αυτά αν ο developer επιθυμούσε να φορτώσει στην συσκευή την εφαρμογή έπρεπε να πληρώσει ένα αντίτιμο της τάξεως των 99\$/χρόνο. Επίσης για να γίνει διαθέσιμη μια εφαρμογή είναι δυνατό μόνο μέσω του App Store είτε δωρεάν είτε με κάποιο αντίτιμο από το οποίο το 70% αντιστοιχεί στον ίδιο και το 30% στην Apple.

Η Apple από το 2003 αναπτύσσει το Xcode, ένα ολοκληρωμένο IDE για την δημιουργία εφαρμογών για Mac OS. Το ίδιο έκανε διαθέσιμο και στους developers για την δημιουργία εφαρμογών για iOS συσκευές. Η τελευταία έκδοση του Xcode είναι διαθέσιμη πάντα σε χρήστες που χρησιμοποιούν και την τελευταία έκδοση του Mac OS ενώ σε παλιότερους χρήστες η Apple βάζει πάλι τους περιορισμούς της και αφήνει μόνο παλιότερες εκδόσεις του Xcode.

Η γλώσσα προγραμματισμού που χρησιμοποιείται για την ανάπτυξη των εφαρμογών είναι η Objective-C με κάποια κομμάτια να επιτρέπεται να γραφτούν σε C ή C++. Η Objective-C είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού που αναπτύχθηκε το 1980 και χρησιμοποιούσε η Apple και για το Mac OS. Η Objective-C συνοδεύτηκε από το API Cocoa Touch το οποίο είναι βασισμένο στο Cocoa που επίσης είχε η εταιρεία για το Mac OS. Το Cocoa Touch περιλαμβάνει όλα αυτά τα κομμάτια λοιπόν που χρειάζονται για την ανάπτυξη μιας εφαρμογής και την διαχείριση των κυρίων δυνατοτήτων και τεχνολογιών που προσφέρει το iOS όπως:

- *App Extension*
- *Handoff*
- *Document Picker*
- *AirDrop*
- *UIKit*
- *UIKit Dynamics*
- *Multitasking*
- *Auto Layout*
- *Storyboards*
- *UI State Preservation*
- *Apple Push Notification Service*
- *Local Notifications*
- *Gesture Recognisers*
- *Standard System View Controllers*

Αναλύοντας το iOS θα δούμε ότι το Cocoa Touch όπως είναι φυσικό βρίσκεται στα ψηλά στρώματα που ο χρήστης έχει άμεση πρόσβαση μετά ακολουθούν όλα τα στοιχεία των πολυμέσων, οι κεντρικές υπηρεσίες του συστήματος και έπειτα ο πυρήνας.



Εικόνα 9. Η αρχιτεκτονική του iPhone.

Στον πυρήνα του iOS βρίσκεται το Darwin, ένα λειτουργικό σύστημα ανοιχτού κώδικα βασισμένο στο Unix που είχε αναπτυχθεί από την εταιρεία Next. Τον Δεκέμβριο του 1996 η Apple εξαγόρασε την εταιρεία και αποφάσισε να κάνει ένα βήμα για την ανοιχτή κοινότητα όμως μετά από κάποιες εκδόσεις που βγήκαν το 2000 πλέον το Darwin προσφέρεται μόνο σαν Source Code και όχι σε έκδοση που μπορεί να γίνει build. Ο πυρήνας του λειτουργικού συστήματος Darwin είναι ο XNU (X is Not Unix) ένας υβριδικός πυρήνας που περιελάμβανε τον πυρήνα Mach, κάποια στοιχεία από το λειτουργικό σύστημα (επίσης βασισμένο στο Unix) BSD και ένα API σε Objective-C για drivers το Driver Kit. Όταν η Apple εξαγόρασε την NeXT η έκδοση του Mach αναβαθμίστηκε στην 3.0, προστέθηκαν στοιχεία στο BSD από το FreeBSD Project και το Driver Kit αντικαταστάθηκε από ένα API σε C++ που ονομαζόταν I/O Kit. Αξίζει να αναφέρουμε ότι ιδρυτής της NeXT ήταν ο Steve Jobs, πρώην CEO της Apple που επανήλθε στην εταιρεία μετά την εξαγορά της NeXT και ήταν ο εμπνευστής του iPhone.

4. Android OS.

4.1 Ιστορική αναδρομή στο Android.

4.1.1 Η προϊστορία.

Η εταιρεία “Android Inc.” ιδρύθηκε το 2003 από τον Andy Rubin ένα πρώην υπάλληλο της Apple με στόχο να παρουσιάσει μια πλατφόρμα ανοιχτή σε όλο τον κόσμο. Ο Rubin είχε μόλις αποχωρήσει από την εταιρεία Danger, την οποία είχε ιδρύσει ο ίδιος, έχοντας στο ενεργητικό του μια καλή συνεργασία με την τεράστια εταιρεία κινητής τηλεφωνίας T-Mobile και την παραγωγή του επιτυχημένου Smartphone, Sidekick το οποίο αποτελούσε μια εξέλιξη του πρωτότυπου τηλεφώνου HiTop του Rubin. Η Android για 2 χρόνια δεν είχε κυκλοφορήσει και δεν είχε πουλήσει τίποτα στην αγορά αλλά ο Rubin με μία μικρή ομάδα από προγραμματιστές είχε ένα σχέδιο για ένα ανοιχτό λογισμικό επόμενης γενιάς συνδυάζοντας πολλές από τις ιδέες που είχε στην προηγούμενη εταιρεία του. Το 2005 η εταιρεία ψάχνοντας για επενδυτές βρήκε ότι καλύτερο έβγαχνε. Ο Larry Page και ο Sergey Brin συνιδρυτές της Google αναζητούσαν να μπουν στην αγορά της κινητής τηλεφωνίας και του λογισμικού γιατί ήθελαν όλο και πιο πολλές συσκευές να χρησιμοποιούν την δικιά τους μηχανή αναζήτησης και να ανταγωνιστούν τις RIM και Microsoft. Έτσι η Google προχώρησε στην εξαγορά της εταιρείας και μέχρι το φθινόπωρο του 2005 είχαν ήδη πάρει θέση σε καινούρια γραφεία στην Καλιφόρνια συνεχίζοντας να υλοποιούν το έργο τους. Στις 5 Νοεμβρίου του 2007 η Google σε συνεργασία με άλλες 34 μεγάλες εταιρείες όπως HTC, Sony, Samsung, Qualcomm, Texas Instruments, T-Mobile κ.α. αναπτύσσουν ένα συνεταιρισμό με σκοπό την ανάπτυξη του ανοικτού κώδικα σε κινητές συσκευές. Ο συνεταιρισμός ονομάστηκε Open Handset Alliance και ανέλαβε την ανάπτυξη του Android. Στις 11 Νοεμβρίου του 2007 δια στόματος του Sergey Brin η Google παρουσίασε ένα Demo του Android με 2 πρωτότυπες συσκευές και την επόμενη μέρα στις 12 Νοεμβρίου έκανε διαθέσιμο το SDK στους developers με την ονομασία "m3-rc20a" με το m3 να σημαίνει Milestone 3. Μόνο μέσα στον browser μπορούσε κάποιος να δει την ονομασία για την πρώτη έκδοση του Android που δεν δόθηκε ποτέ από την Google ως “Android 0.5”.



Εικόνα 10. Ο πρώτος emulator του Android θύμιζε πολύ κλώνο Blackberry και βασιζόταν σε ένα πρωτότυπο μοντέλο της HTC το Sooner το οποίο δεν κυκλοφόρησε ποτέ.

4.1.2 Τα πρώτα χρόνια.

Περίπου ένα χρόνο μετά την ανακοίνωση και την διάθεση του SDK, η αγορά ήταν έτοιμη να υποδεχθεί την πρώτη συσκευή Android. Στις 20 Οκτωβρίου του 2008 κυκλοφόρησε λοιπόν το πρώτο Android Smartphone με το όνομα “T-Mobile G1”. Η συσκευή ήταν φτιαγμένη από την HTC και είχε το κωδικό όνομα “Dream”. Η T-Mobile είχε την αποκλειστική διάθεση στις Ηνωμένες πολιτείες και η HTC στον υπόλοιπο κόσμο. Πολλές πρωτότυπες συσκευές εγκαταλείφθηκαν μέχρι το λανσάρισμα της πρώτης με την HTC επηρεασμένη από την εμφάνιση του iPhone να εγκαταλείψει την ιδέα του Sooner για να εμπιστευθεί ένα μοντέλο με μεγαλύτερη οθόνη αφής κρατώντας όμως το φυσικό πληκτρολόγιο επηρεασμένη λίγο από το πρώιμο Sidekick. Τα χαρακτηριστικά ήταν εντυπωσιακά για την εποχή:

Display:

Type: TFT capacitive touchscreen, 65K colors

Size: 3.2 inches (~46.8% screen-to-body ratio)

Resolution: 320 x 480 pixels (~180 ppi pixel density)

Platform:

OS: Android OS, v.1.0 upgradable to v1.6 (Donut)

Chipset: Qualcomm MSM7201A

CPU: 528 MHz ARM 11

GPU: Adreno 130

Memory:

Card slot: microSD (up to 16gb)

Internal: 192 MB RAM, 256 MB

Camera:

Primary: 3.15 MP, 2048 x 1536 pixels, autofocus, Video

Comms:

WLAN: Wi-Fi 802.11 b/g

Bluetooth: v2.0, A2DP (headset support only)

GPS

USB: miniUSB

Features:

Sensors: Accelerometer, compass

Messaging: SMS(threaded view), MMS, Email, Instant Messaging

Browser: HTML

Java: Yes, via Java MIDP emulator

Voice memo, Mp3 Player

Η T-Mobile παρ' όλα αυτά διέθεσε την συσκευή μόνο σε επιλεγμένες αγορές μέσα στις Ηνωμένες πολιτείες και η HTC παγκοσμίως δεν έδωσε τόσο σημασία στο μάρκετινγκ στο να φανεί ότι η συσκευή ήταν σε συνεργασία με την Google όπως το G1. Έτσι το G1 δεν είχε προφανώς την τεράστια επιτυχία που είχε το iPhone της Apple στην πρώτη του εμφάνιση όμως όλοι ήξεραν ότι είχε γίνει μια αρχή για κάτι πολύ μεγάλο που θα ακολουθούσε.



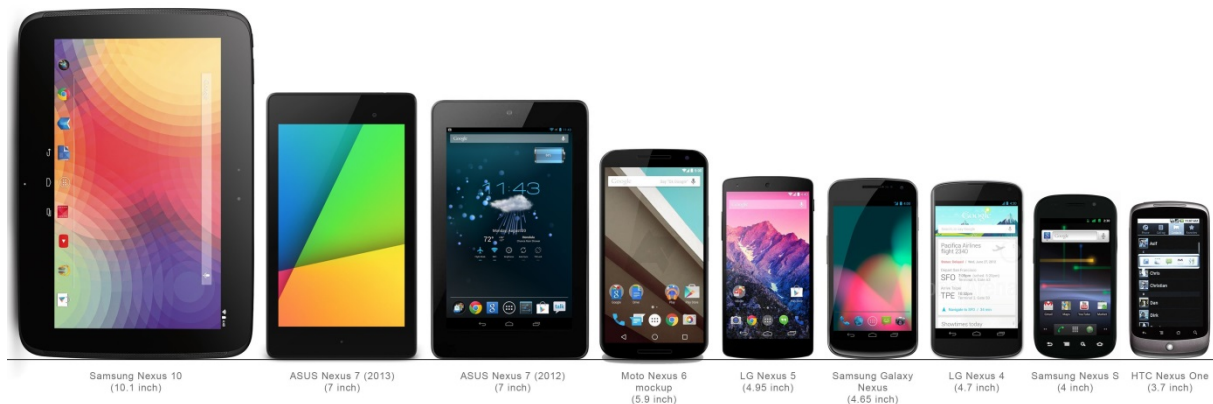
Εικόνα 11. Αριστερά το HTC Dream και δεξιά το T-Mobile G1. Ελάχιστες οι αισθητικές διαφορές μεταξύ τους.

Μαζί βέβαια με την πρώτη συσκευή έκανε την εμφάνιση της και η πρώτη ολοκληρωμένη έκδοση του Android η v.1.0. Στην πρώτη έκδοση του Android βρίσκουμε υπηρεσίες όπως αναφέραμε πιο πάνω στα χαρακτηριστικά του G1, GPS και Bluetooth, ενσωμάτωση αρκετών υπηρεσιών της Google όπως το Gmail, Maps, Calendar, Youtube και βέβαια την πρώτη εμφάνιση του Android Market (το αντίστοιχο App Store της Apple) το οποίο περιείχε 35 εφαρμογές περίπου στην έναρξη του. Η Google έδωσε την πρώτη αναβάθμιση τον Φεβρουάριο του 2009 (v. 1.1) και φρόντισε μέσα στο 2009 να δώσει τις επόμενες μεγάλες αναβαθμίσεις, v.1.5 Απρίλιος 2009 και v.1.6 Σεπτέμβριος 2009, και να αρχίσει τις περιφημες “γλυκίες” κωδικές ονομασίες. Για τις 2 προαναφερθείσες οι κωδικές ονομασίες ήταν “Cupcake” και “Donut”.

4.1.3 Εκδόσεις του Android και Nexus.

Τα χρόνια που ακολούθησαν η Google φρόντιζε να δίνει τουλάχιστον μια μεγάλη αναβάθμιση στο Android κάθε χρόνο μαζί με τις υπόλοιπες μικρές που ήταν περίπου 3 φορές τον χρόνο. Για να γιορτάσει κάθε μεγάλη αναβάθμιση του Android η Google αποφάσισε να λανσάρει μια σειρά συσκευών σε συνεργασία με άλλες εταιρείες με την ονομασία “Nexus”. Η Google ανέλαβε τον σχεδιασμό, το μάρκετινγκ και την υποστήριξη των συσκευών και η δημιουργία και η ανάπτυξη γινόταν σε συνεργασία με διαφορετικές εταιρείες κορυφαίες στην κατασκευή κινητών τηλεφώνων. Η ιδέα ήταν ότι το Nexus θα ήταν μια συσκευή που πρώτο θα περιελάμβανε την καινούρια έκδοση του Android πριν αυτή γίνει διαθέσιμη στους υπόλοιπους χρήστες και η κυκλοφορία του γινόταν ταυτόχρονα με την κυκλοφορία της καινούριας έκδοσης του Android. Επίσης η Google φρόντιζε ότι οι Nexus συσκευές θα έπαιρναν πρώτες τις επόμενες αναβαθμίσεις που θα ακολουθούσαν και δεν θα υποστήριζε όσο ήταν εφικτό την συσκευή. Η αρχή έγινε τον Ιανουάριο του 2010 με το Nexus One την πρώτη συσκευή που κυκλοφόρησε με την έκδοση 2.1 (Eclair), η πρώτη που αναβαθμίστηκε στην έκδοση 2.2 (Froyo) τον Μάιο του ίδιου έτους και στην έκδοση 2.3 (Gingerbread) μέχρι το τέλος του έτους. Το Nexus One δεν θα μπορούσε να γίνει σε συνεργασία με καμιά άλλη εταιρεία παρά την πρώτη που συνεργάστηκε με την Google για την δημιουργία και του G1 την HTC. Είχαμε αναφέρει

ότι η εταιρεία εργάστηκε πολύ με την Google για την δημιουργία πρωτότυπων συσκευών και ο Andy Rubin είχε καλές σχέσεις με τον Peter Chou τον CEO της HTC. Μετά από λίγο καιρό η εταιρεία κυκλοφόρησε την ίδια συσκευή με το δικό της λογότυπο αλλά και μερικές πολύ μικρές αλλαγές όπως είχε γίνει και με το G1.



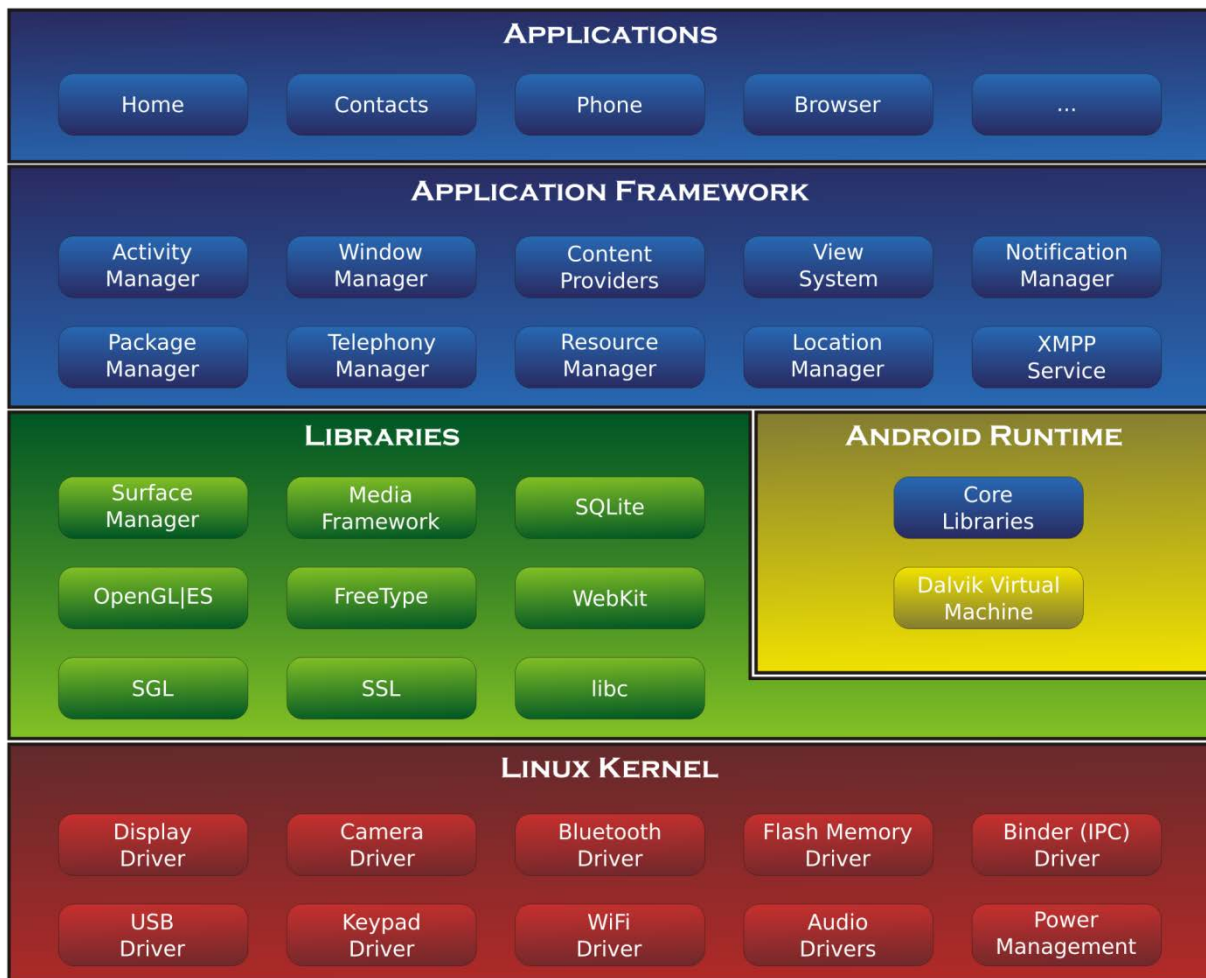
Εικόνα 12. Οι εκδόσεις των Nexus σε τηλέφωνα και ταμπλέτες από δεξιά προς τα αριστερά. Με τα χρόνια όχι μόνο η υπολογιστική δύναμη αλλά και το μέγεθος της συσκευής αυξάνεται.

Η ίδια φιλοσοφία συνεχίζεται από την Google όλα αυτά τα χρόνια για την εξέλιξη και την κυκλοφορία κάθε νέας αναβάθμισης να συνοδεύεται από μια πανίσχυρη Nexus συσκευή. Οι πρώτες εκδόσεις του Android ωστόσο είχαν πάρα πολλά κενά γι αυτό τον λόγο η Google φρόντιζε να έχει συνεχή υποστήριξη και αναβαθμίσεις ώστε να τα καλύπτει στον αντίποδα όμως είχαν κάνει την εμφάνιση τους κάποιες λειτουργίες που πολύς κόσμος αναρωτιόταν γιατί να λείπουν και να μην υποστηρίζονται από τον μεγάλο αντίπαλο του Android το iOS της Apple. Λειτουργίες όπως Multitasking, μεταφορά αρχείων μέσω Bluetooth κ.α. που έλαμπαν δια της απουσίας τους στο iOS στο Android ήταν δεδομένες από την πρώτη στιγμή και γενικά η Google φρόντιζε να ήταν πρωτοπόρα στις υπηρεσίες που θα διέθετε στον τελικό χρήστη με την Apple να έρχεται σχεδόν πάντα δεύτερη. Το Android εκτός των άλλων είχε την υποστήριξη από τόσες εταιρείες και καταλαμβάνοντας το μεγαλύτερο μερίδιο της αγοράς πολύ γρήγορα όπως είδαμε και σε ένα πίνακα πιο πάνω είχε συνεχώς αυξανόμενες απαιτήσεις για όλο και πιο πολλές υπηρεσίες και λειτουργίες. Όλο και πιο ισχυρές Android συσκευές έκαναν την εμφάνιση τους αυξάνοντας τις απαιτήσεις σύμφωνα πάντα με την ανάπτυξη του λειτουργικού. Τον πρώτο καιρό οι προγραμματιστές αντιμετώπισαν διάφορα προβλήματα που έπρεπε να ξεπεραστούν και σιγά σιγά προσαρμόστηκαν γιατί είχαν να αντιμετωπίσουν πολλές ταυτόχρονες διαφορετικές εκδόσεις του Android στην αγορά, συσκευές με πολύ μεγάλη διαφορά στην υπολογιστική ισχύ και διαφορετικές αναλύσεις στις οθόνες. Μέχρι τώρα αυτά τα προβλήματα στο iOS δεν ήταν εμφανή αλλά στο Android τα πράγματα ήταν διαφορετικά.

4.2 Αρχιτεκτονική του Android.

4.2.1 Ανάλυση αρχιτεκτονικής.

Το Android είναι μια στοίβα λογισμικού στην οποία έχουμε ένα διαχωρισμό σε διαφορετικά τμήματα και επίπεδα που ενώνονται με τις υπηρεσίες διασύνδεσης. Στο παρακάτω επίσημο διάγραμμα για την αρχιτεκτονική του Android θα δούμε και θα αναλύσουμε όλα αυτά τα κομμάτια.



Εικόνα 13. Αρχιτεκτονική του Android.

Από το διάγραμμα λοιπόν μπορούμε να δούμε ότι το Android χωρίζεται σε 5 τμήματα και σε 4 επίπεδα. Από πάνω προς τα κάτω λοιπόν έχουμε:

4.2.2 Πυρήνας Linux (Linux Kernel).

Στον πυρήνα του Android βρίσκουμε την βάση του που είναι το Linux. Δεν θα μπορούσε να γίνει κάποια άλλη επιλογή αφού το Android είναι ένα open source project όπως είναι και το Linux το οποίο έχει αποδείξει τόσα χρόνια την σταθερότητα του και την αξιοπιστία του όταν χρειάζεται να στηριχτεί ένα project πάνω του. Η πρώτη έκδοση του Android ήταν βασισμένη στην έκδοση 2.6.25 από τότε όμως ως επί το πλείστον στις συσκευές υπάρχουν οι εκδόσεις 3.4 και 3.10 του πυρήνα. Ο πυρήνας επίσης δέχτηκε κάποιες αλλαγές για να μπορεί να προσαρμοστεί στις ανάγκες του Android. Σκοπός της Google είναι να επιφέρει αυτές τις αλλαγές μόνιμα στο Linux κάτι που σύμφωνα με τον δημιουργό του Linux Linus Torvalds δεν είναι εφικτό στην παρούσα φάση αλλά κάποια στιγμή το Android και το Linux θα καταλήξουν σε ένα κοινό πυρήνα. Ο πυρήνας του Linux περιλαμβάνει όλες τις κύριες λειτουργίες του συστήματος όπως είναι η διαχείριση της μνήμης, η ασφάλεια, οι λειτουργίες του δικτύου όπως επίσης και ένα σύνολο από οδηγών λογισμικού (drivers) που είναι

υπεύθυνοι για την επικοινωνία μεταξύ του software και του hardware της συσκευής. Από εκεί περνάει η διαχείριση για την κάμερα, τον ήχο, το Bluetooth κ.α.

4.2.3 Βιβλιοθήκες (Libraries).

Στο αμέσως επάνω επίπεδο συναντάμε το σύνολο των βιβλιοθηκών που είναι διαθέσιμες στο Android. Αυτές οι βιβλιοθήκες περιλαμβάνουν το σύνολο των APIs που είναι διαθέσιμα στους προγραμματιστές για την ανάπτυξη των εφαρμογών. Τα APIs περιλαμβάνουν όλες αυτές τις πληροφορίες που χρειάζεται κάποιος για να αναπτύξει μια εφαρμογή στο Android και αυτό που χρειάζεται να κάνει είναι να καλέσει αυτές τις βιβλιοθήκες μέσα από την εφαρμογή του. Κάποιες από αυτές τις βιβλιοθήκες είναι για παράδειγμα:

- android.app – Εδώ περιλαμβάνονται όλα τα Services και Activities και είναι η βάση για όλες τις εφαρμογές.
- android.telephony – Επιτρέπει την άμεση αλληλεπίδραση για την δυνατότητα λήψης, τον έλεγχο των τηλεφωνημάτων και των SMS.
- android.text – Εργαλεία επεξεργασίας κειμένου.
- android.widget – Περιλαμβάνει στοιχεία διεπαφής, λίστες, κουμπιά κ.α.

Εκτός από το σύνολο των APIs βρίσκουμε σε αυτό το επίπεδο και τις εγγενείς βιβλιοθήκες (Native Libraries) οι οποίες είναι γραμμένες σε C και C++ και έχουν μεταγλωττιστεί για την συγκεκριμένη αρχιτεκτονική υλικού που χρησιμοποιείται από το Android. Κάποιες από αυτές τις βιβλιοθήκες είναι:

- OpenGL – Υποστήριξη 3D γραφικών.
- FreeType – Γραφικά για γραμματοσειρές
- SGL – Δισδιάστατα γραφικά.
- SQLite – Ελαφριά σχεσιακή βάση δεδομένων.

Μετά την έκδοση 1.6 (Donut) η Google οι developers μπορούν να χρησιμοποιήσουν την εργαλειοθήκη NDK (Native Development Kit) και να γράψουν μόνοι τους τις δικές τους βιβλιοθήκες.

4.2.4 Χρόνος εκτέλεσης (Android Runtime).

Στο ίδιο επίπεδο με τις βιβλιοθήκες βρίσκουμε τον χρόνο εκτέλεσης. Εδώ περιλαμβάνονται οι βασικές βιβλιοθήκες της Java και η εικονική μηχανή Dalvik (Dalvik Virtual Machine). Η Dalvik είναι στην ουσία μια διαφορετική βελτιστοποιημένη έκδοση της εικονικής μηχανής της Java (Java Virtual Machine). Η JVM κάνει compile τον κώδικα μας και είναι αυτή που τρέχει τις εφαρμογές της Java στην συσκευή μας. Η Dalvik δημιουργήθηκε για να κάνει την μετατροπή των αρχείων της Java σε δικά της .dex (Dalvik Executables) και .odex (Optimized Dalvik Executables) αρχεία τα οποία προέρχονται από .class και .jar της Java και είναι πιο συμπαγή και αποδοτικά για να μπορούν να τρέξουν σε κινητές συσκευές που έχουν περιορισμένη μνήμη και μπαταρία. Αφού κάνει την μετατροπή λοιπόν στα δικά της αρχεία μετά τα τρέχει. Η Google ανέφερε ότι η δημιουργία της Dalvik επιτρέπει σε μια συσκευή να τρέξει πολλαπλές φορές ταυτόχρονα την εικονική μηχανή με πολύ καλή απόδοση.

4.2.6 Εφαρμογές (Applications).

Στο ανώτερο και τελευταίο επίπεδο του Android βρίσκουμε τις εφαρμογές. Αυτές περιλαμβάνουν και τις προεγκατεστημένες εφαρμογές που βρίσκονται στην συσκευή (Phone, Browser, Contacts, Messaging κ.α.) και τις υπόλοιπες που μπορεί να εγκαταστήσει ο χρήστης από μόνος του. Όταν μια εφαρμογή είναι έτοιμη γίνεται μετατροπή για να την χρησιμοποιήσουμε σε ένα .apk αρχείο. Μέχρι να φτάσουμε όμως σε αυτό το σημείο η εφαρμογή μας είναι αλλιώς δομημένη. Στο επόμενο κεφάλαιο θα αναλύσουμε πως ακριβώς είναι δομημένη η εφαρμογή, τι περιλαμβάνει και τι θα συναντήσουμε εμείς όταν ξεκινήσουμε ένα καινούριο project.

4.3 Ανάλυση μιας εφαρμογής Android.

4.3.1 AndroidManifest.xml.

Το AndroidManifest.xml είναι ένα αρχείο που συνοδεύει κάθε καινούριο Project που σχεδιάζουμε για να φτιάξουμε μια εφαρμογή. Το αρχείο εμπεριέχει κάποιες πληροφορίες που χρειάζεται η συσκευή για να τρέξει την εφαρμογή όπως το όνομα του πακέτου που έχουμε δημιουργήσει για την εφαρμογή (1), το μίνιμουμ απαιτούμενο SDK (2) που μπορεί να τρέξει η εφαρμογή (εξαρτάται από την έκδοση του Android) και σε ποιο SDK (3) έχουμε στοχεύσει εμείς για να δουλεύει σωστά.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
  xmlns:android=
    "http://schemas.android.com/apk/res/android"
  package="com.example.myfirstandroidapp" (1)
  android:versionCode="1"
  android:versionName="1.0" >
  <uses-sdk
    android:minSdkVersion="8" (2)
    android:targetSdkVersion="16" /> (3)
  <application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <activity (4)
      android:name=
        "com.example.myfirstandroidapp.MainActivity"
      android:label="@string/app_name" >
      <intent-filter>
        <action android:name=
          "android.intent.action.MAIN" /> (5)
        <category android:name=
          "android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>
</manifest>
```

Παράδειγμα ενός AndroidManifest.xml

Πάμε να δούμε τα δομικά στοιχεία που πρέπει να περιλαμβάνονται μέσα σε ένα `AndroidManifest.xml` παρακάτω:

- **Δραστηριότητα – Activity (4)**
Είναι το πιο σημαντικό στοιχείο μέσα σε ένα `AndroidManifest.xml`. Το Activity είναι μια κλάση Java δημιουργημένη από εμάς που επεκτείνει την βασική κλάση Activity (Activity Base Class). Το Activity είναι στην ουσία αυτό που έχει την άμεση συσχέτιση με τον χρήστη γιατί μέσα από αυτό θα σχεδιάσουμε το UI (User Interface) χρησιμοποιώντας τις Όψεις (Views) όπως θα δούμε παρακάτω και είναι αυτό που θα διαχειριστεί τα Συμβάντα (Events) που δημιουργούνται από τον χρήστη.
- **Προθέσεις – Intents (5)**
Για να κάνει το Activity αυτά που χρειάζεται χρησιμοποιεί τα Intents που καθορίζουν τι ακριβώς θέλουμε να παρέχουμε στον χρήστη της εφαρμογής όπως το πώς θα γίνει η προβολή των δεδομένων.
- **Υπηρεσίες – Services**
Τα Services είναι ένα κομμάτι κώδικα το οποίο τρέχει στο παρασκήνιο της εφαρμογής και επιστρέφει αποτελέσματα ακόμα και όταν η εφαρμογή δεν είναι στο προσκήνιο.
- **Πάροχος Περιεχομένου – Content Provider**
Διαχειρίζεται συγκεκριμένα δεδομένα τα οποία έχουμε ορίσει από πριν στην εφαρμογή μας.
- **Δέκτες Μετάδοσης – Broadcast Receivers**
Είναι ένα είδος υπηρεσίας το οποίο αναλαμβάνει να διαχειριστεί τις Προθέσεις, να ενημερώσει την εφαρμογή ή άλλες εφαρμογές και να προβάλλει μια ειδοποίηση αν χρειαστεί στον χρήστη.

4.3.2 Ο φάκελος `src`.

Ο φάκελος `src` (source) εμπεριέχει τις κλάσεις τις οποίες εμείς δημιουργούμε για την εφαρμογή στα οποία αναφέρονται αναλυτικά όπως θα δούμε και παρακάτω όλα τα Activities, Intents, Services κλπ.

4.3.3 Ο φάκελος `res`.

Στο φάκελο `res` (resources) βρίσκουμε καταχωρημένα σε υποφακέλους αρχεία εικόνας και `xml` τα οποία χρησιμοποιούνται στο γραφικό κομμάτι κυρίως της εφαρμογής από τις κλάσεις που βρίσκονται στο φάκελο `src`.

4.3.4 Ο φάκελος `gen`.

Ο φάκελος εμπεριέχει κώδικα που έχει δημιουργηθεί αυτόματα από τα όλα τα resources που έχουμε δηλώσει εμείς στον φάκελο `res` και γίνονται διαθέσιμα σε εμάς μέσω του αρχείου `R.java`.

4.3.5 Ο φάκελος `bin`.

Στον φάκελο `bin` επίσης εμπεριέχονται αρχεία τα οποία έχουν δημιουργηθεί αυτόματα από τις κλάσεις που έχουμε γράψει εμείς και χρησιμοποιούνται από τον compiler για να συντεθεί το `.apk` αρχείο που είπαμε ότι είναι η τελική μορφή της εφαρμογής.

4.3.6 Ο φάκελος assets.

Στον φάκελο assets τοποθετούμε εμείς ότι ακατέργαστα δεδομένα (raw data) θέλουμε να χρησιμοποιήσουμε στην εφαρμογή μας.

4.4 Android και Γλώσσες Προγραμματισμού.

4.4.1 Ποιές Γλώσσες προγραμματισμού χρησιμοποιούνται στο Android.

Αναφέραμε ήδη παραπάνω ότι στο Android γίνεται χρήση της C και της C++ στο επίπεδο των βιβλιοθηκών και την Dalvik Virtual Machine στο Android Runtime. Πάμε να δούμε όμως ποιες γλώσσες προγραμματισμού και εργαλεία μπορούν να χρησιμοποιηθούν για μια εφαρμογή Android.

- **Java**
Από τις πιο γνωστές γλώσσες προγραμματισμού και βασική στο Android την οποία θα αναλύσουμε παρακάτω.
- **C & C++**
Η C και C++ μπορούν να χρησιμοποιηθούν όπως αναφέραμε και πιο πάνω χρησιμοποιώντας το NDK αλλά και για εφαρμογές χωρίς όμως να το συστήνει η Google γιατί αυξάνει την πολυπλοκότητα και οι πιο πολλές εφαρμογές δεν θα έχουν κάποιο όφελος.
- **Basic4Android**
Εργαλείο εμπνευσμένο από την Microsoft Visual Basic 6 και το Microsoft Visual Studio που κάνει τον προγραμματισμό πιο εύκολο σε κάποιον που είναι ήδη γνώστης της γλώσσας και δυσκολεύεται με τη Java.
- **Corona SDK**
Το Corona SDK προσφέρει την δυνατότητα κατασκευής εφαρμογών για iOS και Android χρησιμοποιώντας γραφικό περιβάλλον.
- **Delphi**
Για εφαρμογή Android με Objective Pascal.
- **HyperNext Android Creator**
Πρόκειται για μια εφαρμογή η οποία απευθύνεται σε άτομα που ξεκινούν την ενασχόληση τους με τον προγραμματισμό. Η βασική γλώσσα προγραμματισμού είναι η HyperNext που προέρχεται από την HyperTalk.
- **Qt For Android**
Το Qt για Android επιτρέπει Qt εφαρμογές να τρέξουν στο Android. Το Qt είναι ένα πλαίσιο εφαρμογών που μπορεί να χρησιμοποιηθεί σε διαφορετικές πλατφόρμες (cross-platform application framework). Χρησιμοποιεί C++ και QML και ο Qt Creator είναι ο IDE που χρησιμοποιείται.
- **HTML, JavaScript, CSS**
Επίσης μια ακόμα επιλογή στο να προγραμματίσεις στο Android είναι φορτώνοντας έναν προβολέα για ιστό (WebView) να χρησιμοποιήσεις τις ήδη γνωστές υπάρχουσες γλώσσες που χρησιμοποιούνται στο Web.

4.4.2 Android και Java

Στο Android επιλέχθηκε από την αρχή η βασική γλώσσα προγραμματισμού να είναι η Java. Η Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού που αναπτύχθηκε από την Sun Microsystems στις αρχές του 1990. Η Sun αναζητούσε ένα εργαλείο για να μπορεί να αναπτύξει εφαρμογές από οικιακές συσκευές μέχρι πολύπλοκα γραφικά συστήματα και άρχισε να αναπτύσσει και να δουλεύει πάνω στην C++. Το 1995 παρουσίασε την Java, μέσα στον ίδιο χρόνο πολλές μεγάλες εταιρείες έδειξαν το ενδιαφέρον τους να την χρησιμοποιήσουν για την δημιουργία λογισμικού και στις 13 Νοεμβρίου του 1996 η Java έγινε πλέον μια γλώσσα ανοιχτού κώδικα (open source) και για τον μεταγλωττιστή (compiler) της και για το πακέτο ανάπτυξης JDK (Java Development Kit). Ένα από τα μεγάλα πλεονεκτήματα που βρήκε η Google στην Java είναι η ανεξαρτησία του λειτουργικού συστήματος και της πλατφόρμας. Ότι πρόγραμμα γραφτεί σε Java μπορεί να τρέξει σε όλα τα

λειτουργικά συστήματα χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση και χωρίς να αλλάξει ο πηγαίος κώδικας. Για να γίνει όμως αυτό χρειάζεται η εγκατάσταση εικονικής μηχανής (Java Virtual Machine) στο σύστημα μας όπως είδαμε και πιο πάνω με την Dalvik στο Android. Εκτός από αυτό η Google βρήκε και άλλα πλεονεκτήματα στην Java για να την χρησιμοποιήσει όπως:

- Η Java είναι γλώσσα ανοιχτού κώδικα όπως και το Android.
- Ήταν ήδη γνωστή στους προγραμματιστές, υπήρχε έτοιμο κοινό να δουλέψει κατευθείαν για εφαρμογές Android.
- Υπήρχε ήδη ένα μεγάλο σύνολο από εργαλεία ανάπτυξης λογισμικού.
- Πολλές κινητές συσκευές χρησιμοποιούσαν ήδη την έκδοση Java ME οπότε οι εταιρείες ήταν ήδη εξοικειωμένες με την γλώσσα.

Η Google έδειξε από την αρχή τις προθέσεις της και επεδίωξε να έχει μια συνεργασία με την Sun για την χρησιμοποίηση της Java, μια συνεργασία η οποία περιελάμβανε:

- Η Sun βοήθησε στο να γίνει open source το Android και έγινε μέλος του Open Handset Alliance.
- Μπήκε σε ένα πρόγραμμα συνεργασίας με την Google για να φέρουν την πλατφόρμα του Android στην αγορά.
- Η Google θα πλήρωνε στην Sun 25-50 εκατομμύρια δολάρια και θα συζητούσαν ενδεχόμενα κέρδη από διαφημίσεις.

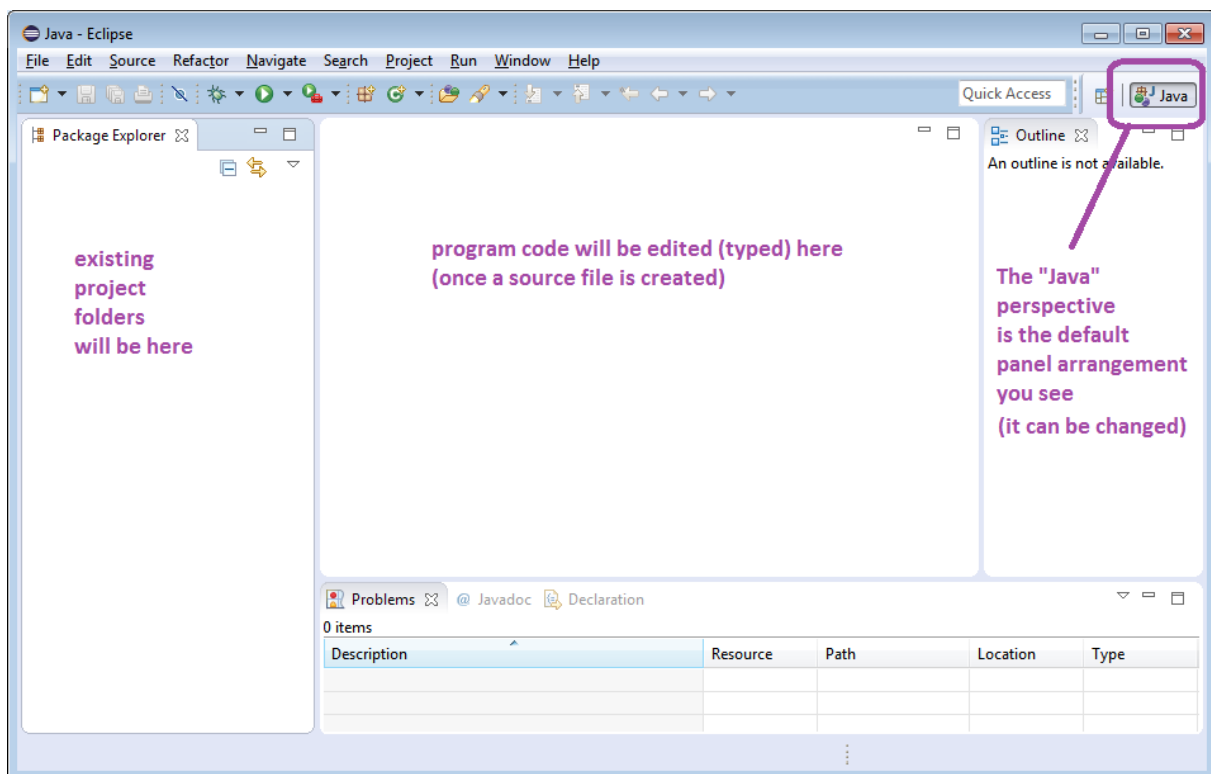
Ο Andy Rubin μάλιστα είχε δηλώσει ότι αν η Sun δεν ήθελε να συνεργαστεί μαζί τους, είχαν 2 επιλογές, είτε να υιοθετήσουν άλλη Virtual Machine και να δουλέψουν με την Objective C ή να χρησιμοποιήσουν την Java έτσι κι αλλιώς. Τα πράγματα δεν εξελίχθηκαν όπως θα ήθελαν όλοι μιας και η Oracle εξαγόρασε την Sun το 2010 και μήνυσε την Google για παραβίαση πατεντών και πνευματικής ιδιοκτησίας.

5. Προγραμματίζοντας με το Android.

5.1 Επιλογή IDE και εγκατάσταση παραμέτρων.

5.1.1 Εγκατάσταση του Eclipse.

Για να ξεκινήσουμε να προγραμματίζουμε στο Android το πρώτο βασικό που θα έπρεπε να επιλέξουμε είναι το IDE μας που μιλήσαμε και στο εισαγωγικό κεφάλαιο. Δύο κορυφαίες επιλογές στον χώρο ήταν διαθέσιμες για εμάς, το Eclipse και το NetBeans. Διαλέξαμε το Eclipse γιατί το πρόσθετο (plugin) του Android που προσφερόταν το οποίο ήταν και το επίσημο από την Google ήταν πολύ καλύτερο από αυτό του NetBeans. Πρώτη μας δουλειά λοιπόν πριν να εγκαταστήσουμε το Eclipse είναι η εγκατάσταση του Java JDK στο σύστημα μας ώστε να είναι ικανό πλέον το μηχάνημα μας να τρέξει την Java. Μετά την εγκατάσταση του JDK περνάμε στην εγκατάσταση του Eclipse. Από την ιστοσελίδα του Eclipse μπορούμε να κατεβάσουμε την πλήρη έκδοση ή την έκδοση για Java EE Developers. Κατά την εγκατάσταση του το Eclipse θα μας ζητήσει να δώσουμε μια διαδρομή σε ένα φάκελο στο σύστημα μας (workspace) όπου θα αποθηκεύονται τα Project που δημιουργούμε στο Eclipse.

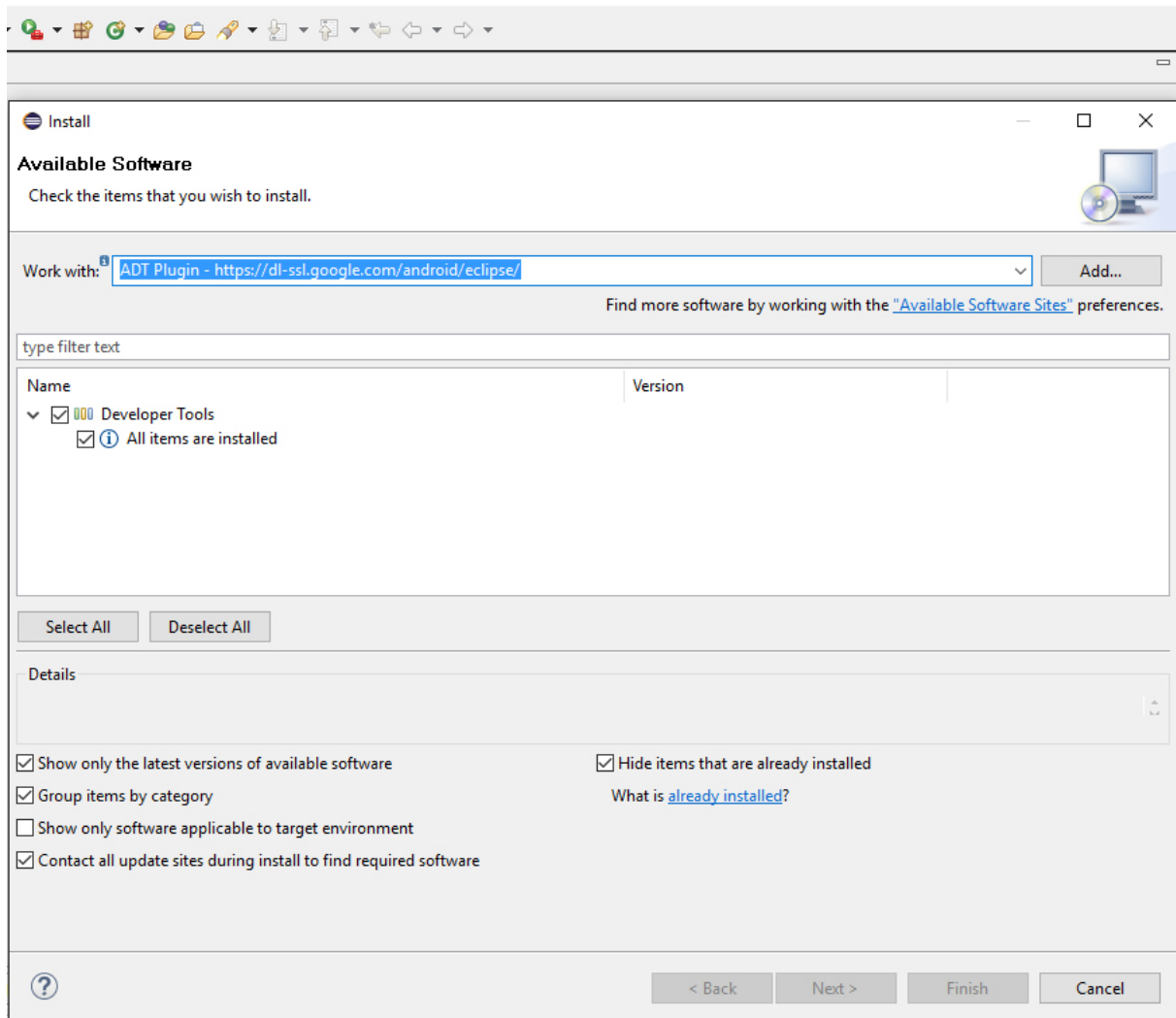


Εικόνα 15. Eclipse IDE.

Μετά την εγκατάσταση και την πρώτη μας είσοδο στο Eclipse θα δούμε όπως φαίνεται και στην παραπάνω εικόνα ότι ο χώρος που εργαζόμαστε είναι χωρισμένος σε διάφορα κομμάτια. Αριστερά είναι ο χώρος που θα εμφανίζονται τα Project που έχουμε στο workspace folder που δηλώσαμε κατά την εγκατάσταση. Στο κεντρικό κομμάτι είναι που θα εμφανίζεται ο κώδικας που μπορούμε να δημιουργήσουμε από την αρχή ή να τροποποιήσουμε. Στο δεξί κομμάτι είναι η όψη για τον προγραμματισμό σε Java που αναγνωρίζει τον κώδικα μας και μπορεί να μας δώσει μια γρήγορη πρόσβαση σε κάποια σημεία. Στο κάτω μέρος συναντάμε το κομμάτι του LogFile και όλα τα μηνύματα και οι ειδοποιήσεις του Eclipse για οποιοδήποτε πρόβλημα προκύψει.

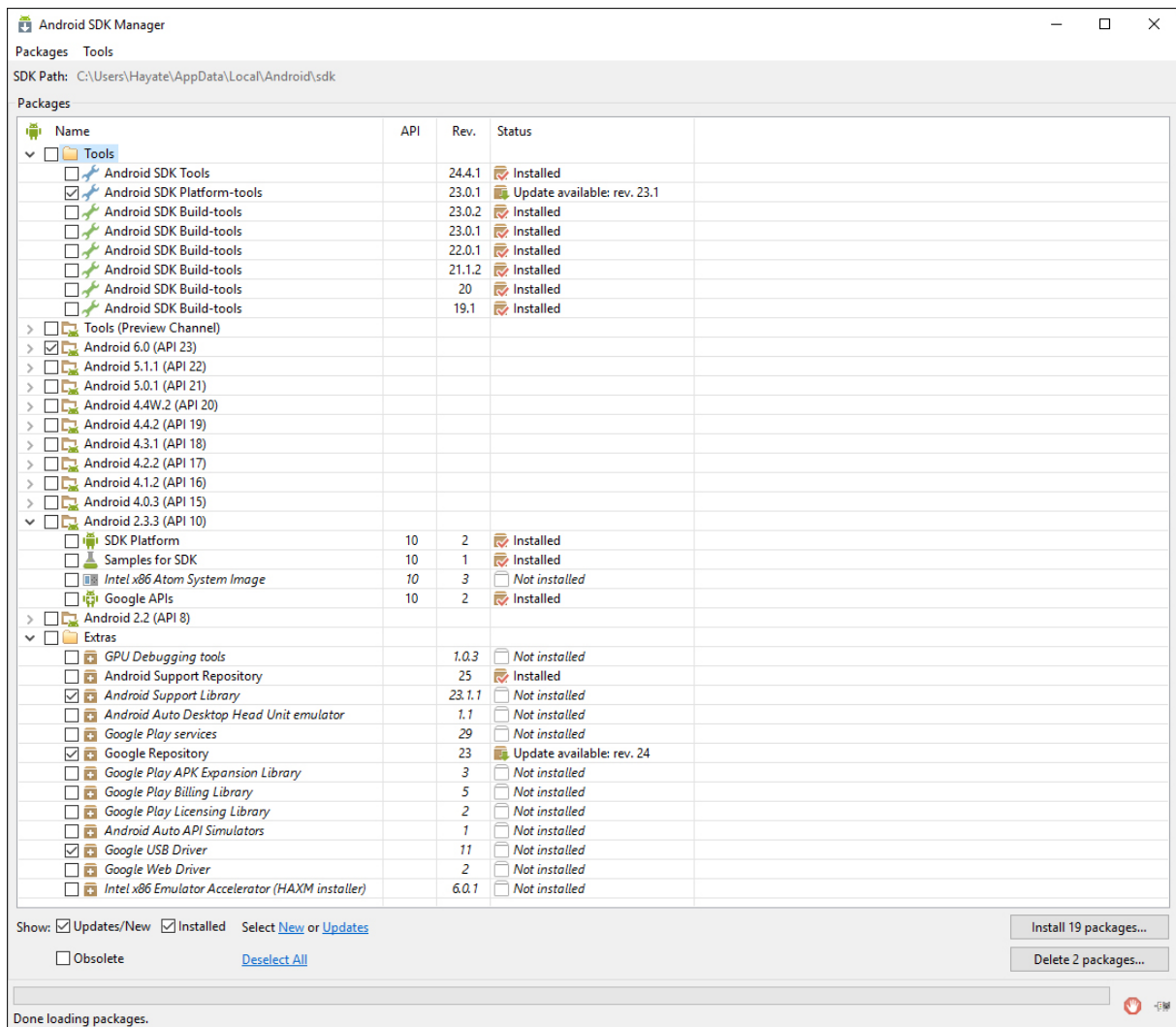
5.1.2 Εγκατάσταση του ADT Plugin και του Android SDK Manager.

Το επόμενο μας βήμα είναι να προχωρήσουμε στην εγκατάσταση του ADT (Android Development Tools) Plugin κατευθείαν από το site της Google. Ξεκινάμε το Eclipse και επιλέγουμε “**Help->Install New Software**” και στο πεδίο “**work with**” πληκτρολογούμε <https://dl-ssl.google.com/android/eclipse/>.



Εικόνα 16. Εγκατάσταση ADT Plugin.

Επιλέγουμε τα **Developer Tools** πατάμε **Next** βλέπουμε την λίστα των εργαλείων που θα κατεβάσουμε αποδεχόμαστε τους όρους και κάνουμε επανεκκίνηση στο Eclipse. Ταυτόχρονα μπορούμε να κατεβάσουμε από την τελευταία έκδοση του Android SDK στον υπολογιστή μας και να την αποθηκεύσουμε σε κάποιο φάκελο. Μετά την επανεκκίνηση το θα μας ζητηθεί να δείξουμε στο Eclipse σε ποιο path είναι αποθηκευμένο το SDK. Αν δεν εμφανιστεί το μενού το κάνουμε μόνοι μας από το **Window->Preferences->Android SDK Location** πατάμε Browse και δίνουμε την διαδρομή στο Eclipse. Έπειτα θα πρέπει να ανοίξουμε τον Android SDK Manager και να επιλέξουμε μέσα από μια μεγάλη λίστα τα SDK και τα API's για τις εκδόσεις του Android που θέλουμε να δουλέψουμε. Κάθε διαφορετική έκδοση του Android έχει διαθέσιμη την πλατφόρμα διάφορα Samples έτοιμα για να πάρουμε κάποια παραδείγματα και το API βέβαια. Εκτός από αυτό μπορούμε να κατεβάσουμε και κάποια άλλα εργαλεία που θα μας βοηθήσουν στον προγραμματισμό μας.

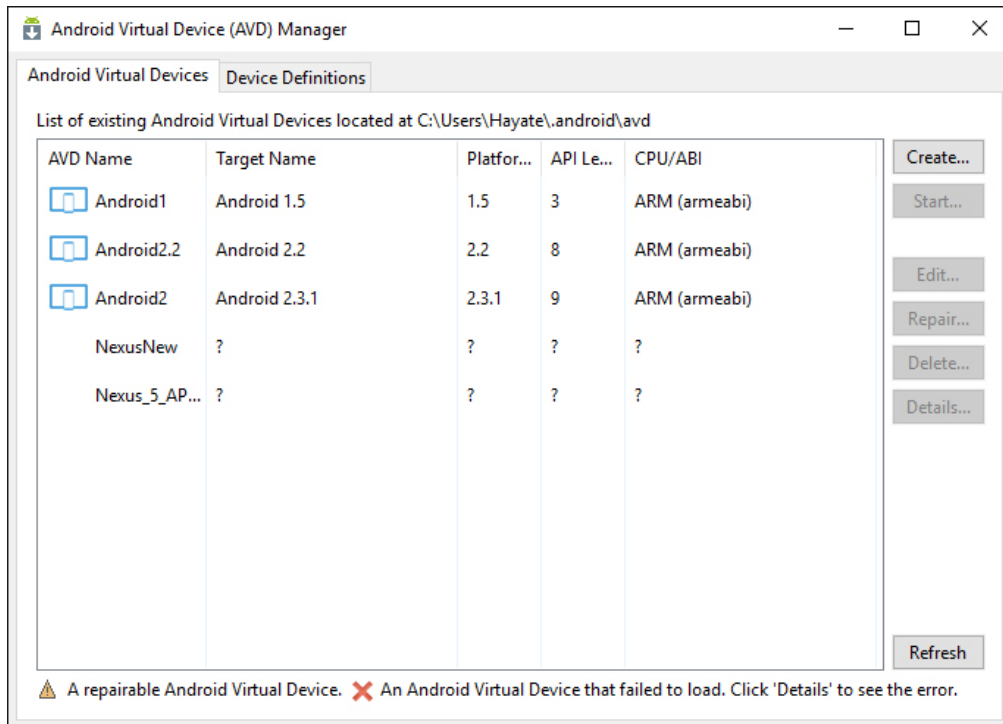


Εικόνα 17. Android SDK Manager.

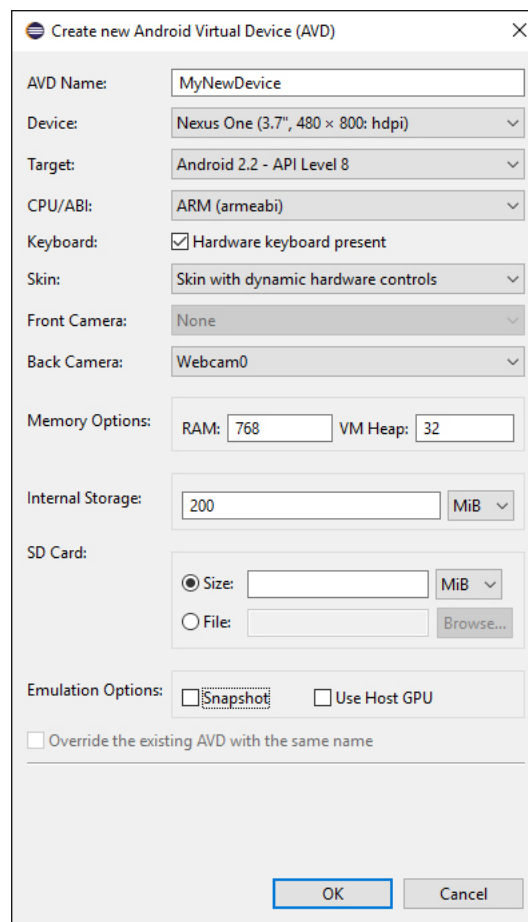
Στην παραπάνω εικόνα μπορούμε να δούμε αυτό που περιγράψαμε ότι εμφανίζεται ότι έχουμε εγκαταστήσει δηλαδή μέχρι τώρα και τι είναι ακόμα διαθέσιμο για μεταφόρτωση ώστε να δουλέψουμε σε άλλες εκδόσεις όπως επίσης και τα διαφορετικά εργαλεία.

5.1.3 Ρύθμιση του AVD Virtual Manager.

Το Android Virtual Device Manager είναι ένα εργαλείο το οποίο μας επιτρέπει να δημιουργήσουμε μια εικονική συσκευή στον υπολογιστή μας και να εγκαταστήσουμε εκεί τις εφαρμογές που έχουμε δημιουργήσει ώστε να έχουμε άμεσα μια εικόνα της εφαρμογής μας σε λειτουργία. Το σημαντικό με το AVD είναι ότι μας δίνει πάρα πολλές επιλογές για να δημιουργήσουμε πολλές διαφορετικές συσκευές οι οποίες θα έχουν διαφορετικό μέγεθος οθόνης, διαφορετική ανάλυση οθόνης, διαφορετική αρχιτεκτονική επεξεργαστή, διαφορετική μνήμη RAM, διαφορετική έκδοση του Android και έτσι να μπορούμε να δοκιμάσουμε την εφαρμογή μας σε πολλά διαφορετικά συστήματα και να αποκλείσουμε τυχόν δυσλειτουργίες να βελτιώσουμε την εφαρμογή μας και να έχουμε μια πλήρη εικόνα για αυτό που έχουμε δημιουργήσει.

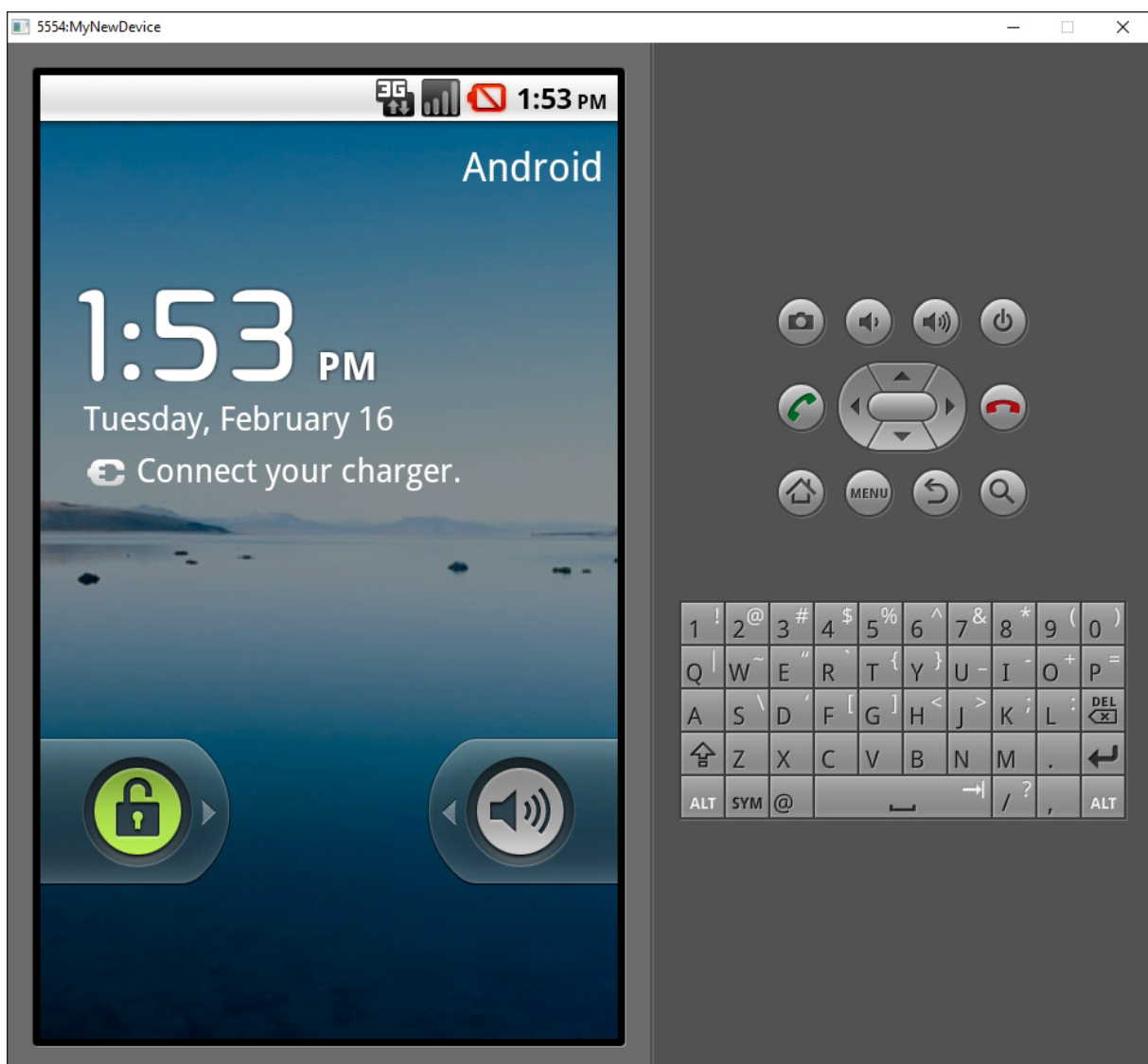


Εικόνα 18. AVD Manager με την λίστα των συσκευών που έχουμε δημιουργήσει.

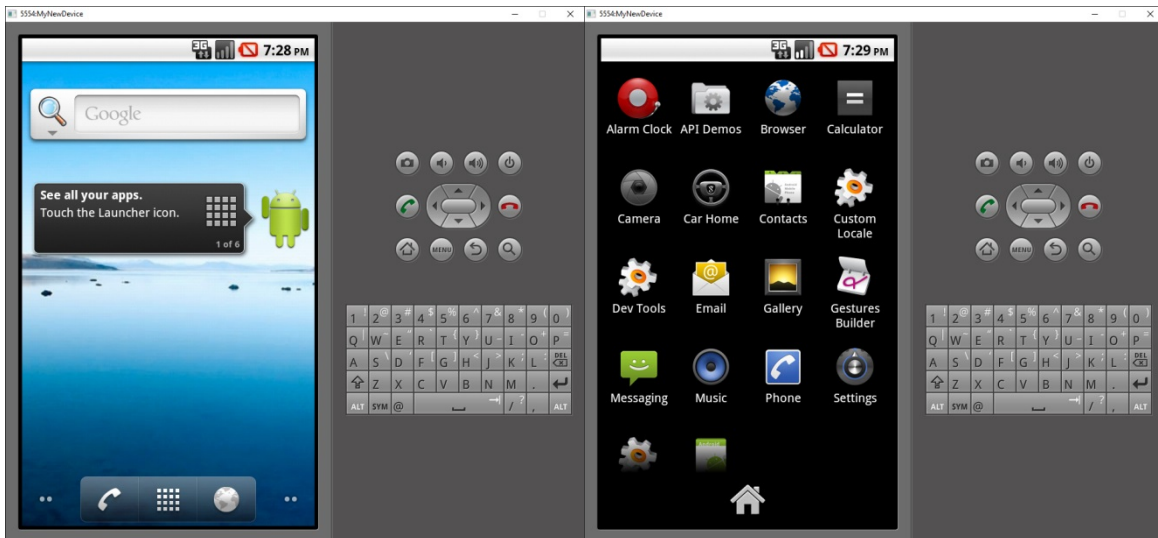


Εικόνα 19. Παραμετροποιήσεις για την δημιουργία νέας εικονικής συσκευής.

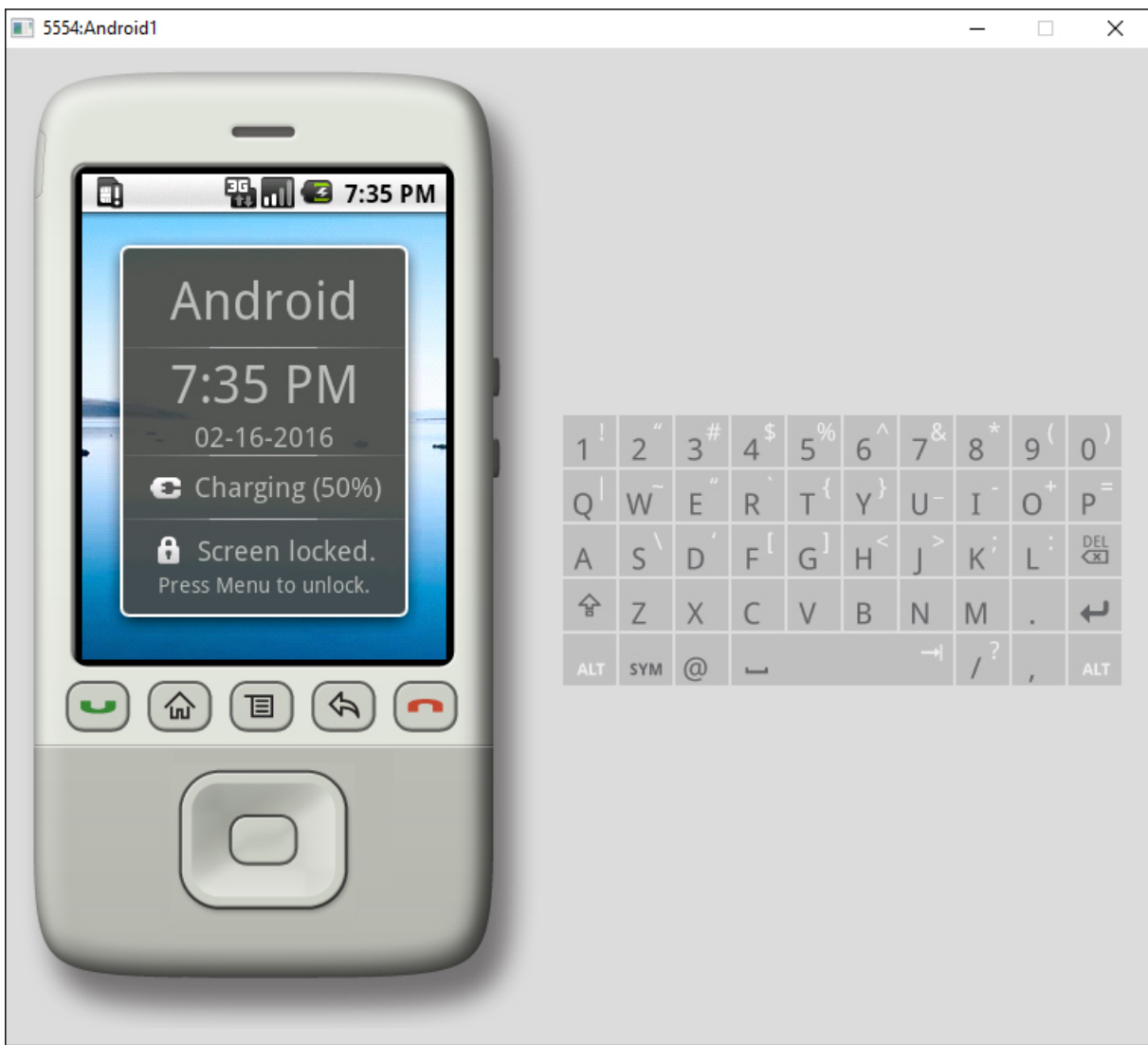
Πατώντας πάνω στο κουμπί “Create” στην Εικόνα 18 μας εμφανίζεται ένα καινούριο παράθυρο όπως φαίνεται στην Εικόνα 19 στο οποίο εμφανίζονται όλες οι παραμετροποιήσεις που έχουμε να επιλέξουμε. Δίνουμε ένα όνομα στην συσκευή μας, διαλέγουμε ένα τύπο συσκευής μέσα από μια λίστα που είναι διαθέσιμη με διαφορετικές συσκευές Nexus και άλλες πρωτότυπες συσκευές με διαφορετικά μεγέθη οθόνης και αναλύσεις, εμείς όπως φαίνεται επιλέξαμε το Nexus One μια συσκευή με μέγεθος 3.7 Έπειτα διαλέγουμε την έκδοση του Android που θέλουμε, το σύστημα έβαλε μόνο του την Froyo 2.2 βάσει της συσκευής που επιλέξαμε. Κάνουμε τις τελευταίες ρυθμίσεις για το αν θα χρειαστούμε κάποιο skin, να πειράζουμε λίγο το scale ώστε να φαίνεται καλύτερα στον υπολογιστή μας την RAM που θα χρειαστούμε, χώρο εσωτερικής αποθήκευσης και εικονική κάρτα SD για αποθήκευση. Μόλις τελειώσουμε πατάμε πάνω στο κουμπί “OK” και δημιουργείται η συσκευή μας. Επιστρέφουμε στο προηγούμενο παράθυρο επιλέγουμε την συσκευή πατάμε Start και ο προσομοιωτής (Emulator) εκκινείται για να δημιουργήσει την συσκευή μας. Το αποτέλεσμα είναι το παρακάτω:



Εικόνα 20. Ο Emulator μας τρέχει.



Εικόνα 21. Χρησιμοποιώντας τον κέρσορα μπορούμε να ξεκλειδώσουμε την συσκευή και να μπούμε στο μενού με τις εφαρμογές όπως φαίνεται στις εικόνες και ο emulator είναι πλήρως λειτουργήσιμος.



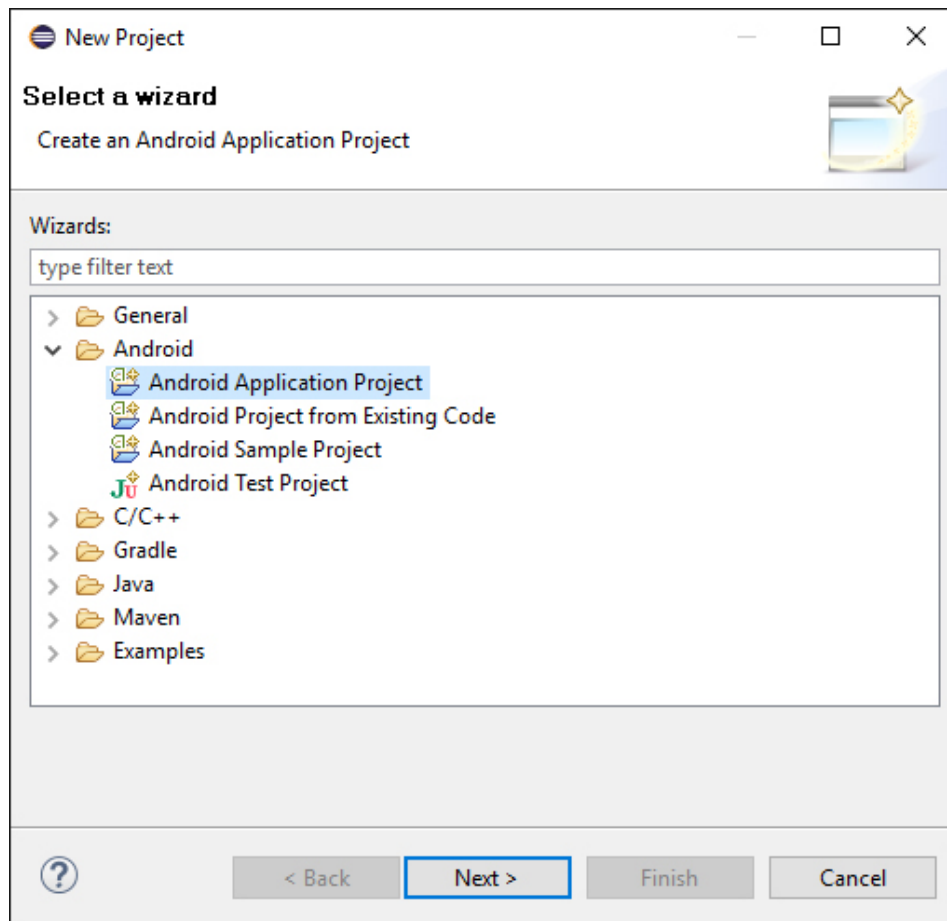
Εικόνα 22. Σε κάποιες περιπτώσεις έχουμε και πιο ρεαλιστικούς emulator.

Τώρα πλέον που έχουμε δημιουργήσει και την εικονική συσκευή όταν θα τρέξουμε την εφαρμογή μας το Eclipse θα ψάξει για συμβατές συσκευές που να μπορεί να εγκατασταθεί και να τρέξει αυτόματα αλλιώς θα μας ζητήσει εμάς να επιλέξουμε συσκευή. Κάθε φορά που τρέχουμε την εφαρμογή μας έχουμε την δυνατότητα να το αλλάξουμε αυτό όπως και κάποιες άλλες παραμέτρους.

5.2 Ανάπτυξη εφαρμογών στο Eclipse.

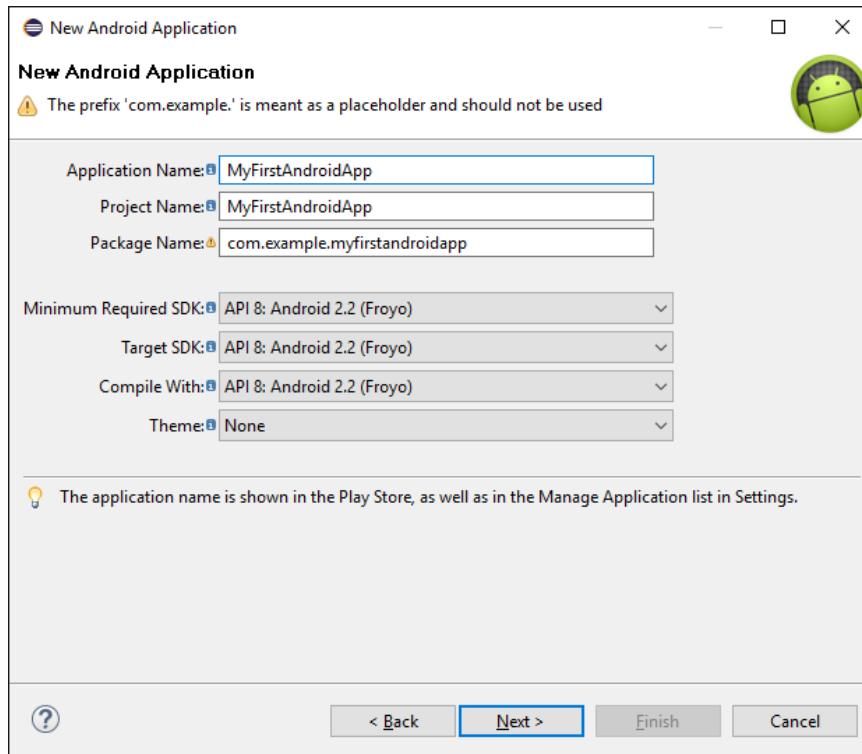
5.2.1 Η πρώτη μου εφαρμογή.

Για να ξεκινήσουμε μια καινούρια εφαρμογή Android θα πρέπει να δημιουργήσουμε ένα καινούριο Android Project στο Eclipse. Πάμε στο **File->New->Project** και θα συναντήσουμε την παρακάτω εικόνα:



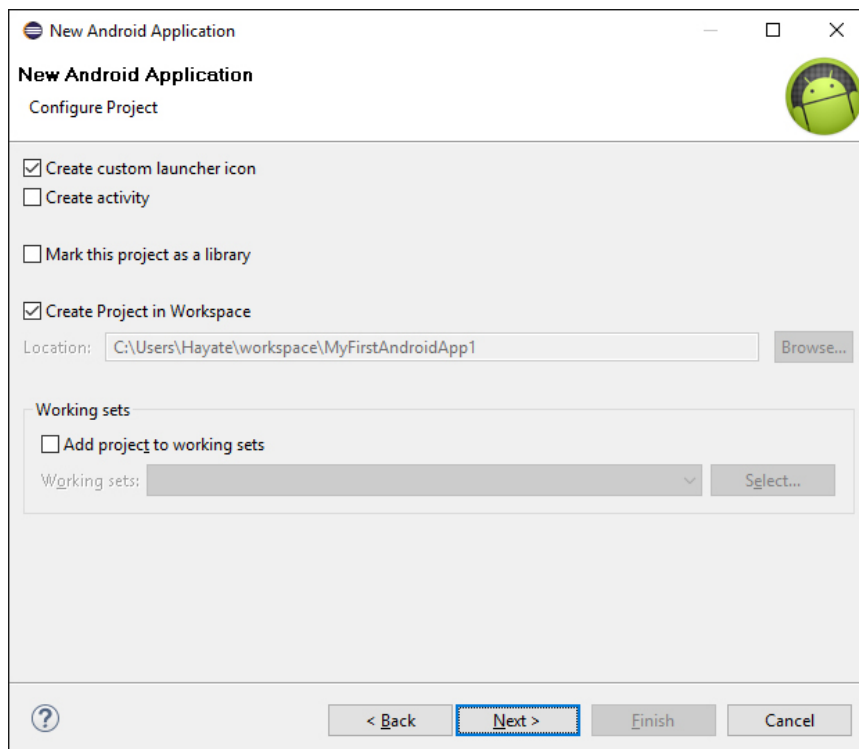
Εικόνα 23. Δημιουργία ενός νέου Android Project.

Από το πάνω παράθυρο βλέπουμε ότι μια επιλογή είναι να επιλέξουμε ένα “Android Sample Project” το οποίο θα μας ανοίξει μια λίστα από ήδη υπάρχουσες εφαρμογές που θα μπορέσουμε να διαλέξουμε και να τρέξουμε κανονικά να δούμε τον κώδικα ώστε να δούμε πράγματα που θα θέλουμε κι εμείς να υλοποιήσουμε. Εμείς θα επιλέξουμε “Android Application Project” και θα πατήσουμε “Next” για να δημιουργήσουμε ένα κενό Project.



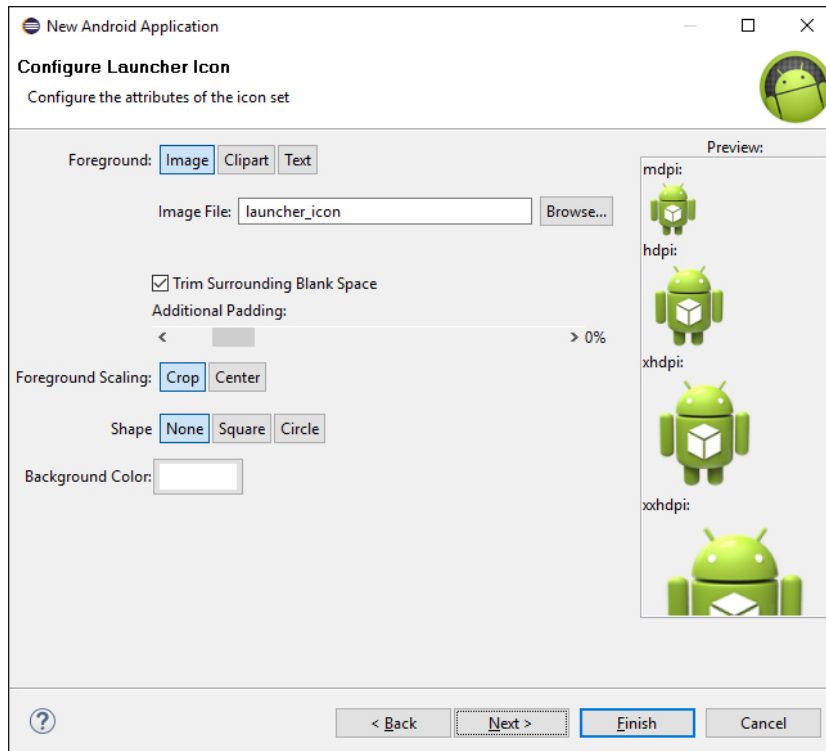
Εικόνα 24. Δημιουργία ενός νέου Android Project - 2.

Εδώ δίνουμε τις ονομασίες που χρειάζεται και διαλέγουμε την έκδοση για την εφαρμογή. Το σύστημα μας βοηθάει σε τι πρέπει να επιλέξουμε. Αφού τα έχουμε επιλέξει όλα πατάμε “Next”.



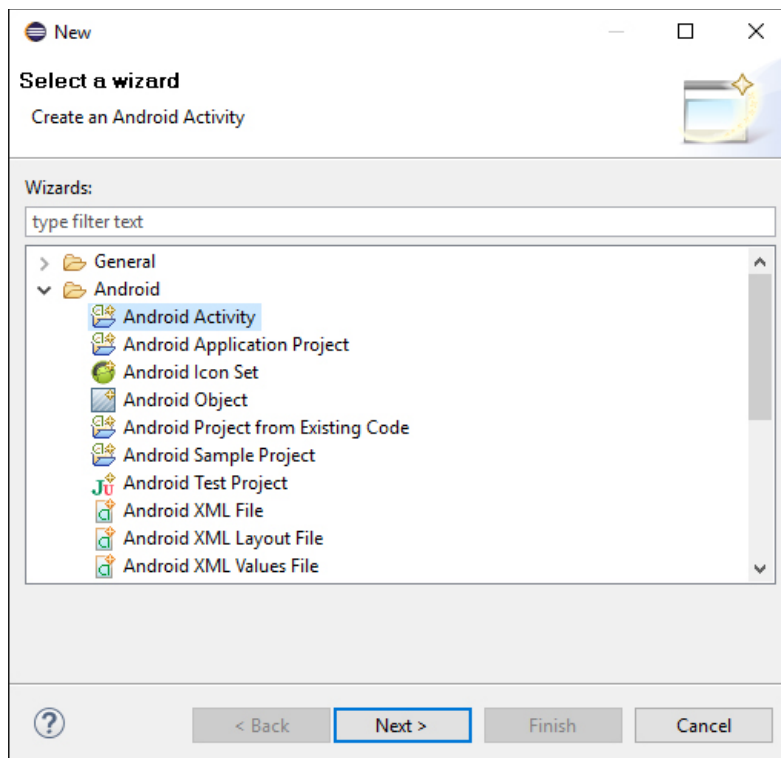
Εικόνα 25. Δημιουργία ενός νέου Android Project - 3.

Αποεπιλέγουμε το “Create activity” γιατί θα το κάνουμε σε άλλο σημείο και πατάμε “Next”.



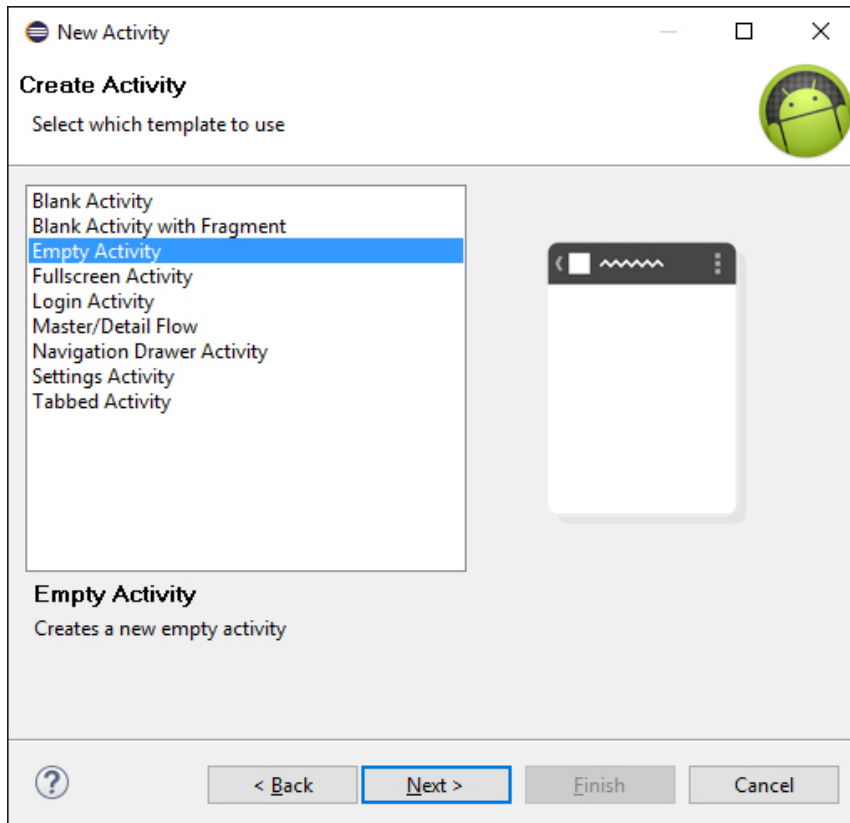
Εικόνα 26. Επιλογή εικονιδίου εφαρμογής.

Στην τελευταία εικόνα βλέπουμε ότι μπορούμε να επιλέξουμε και να τροποποιήσουμε το εικονίδιο της εφαρμογής μας και πατώντας απλά μετά “Finish” το Project μας δημιουργείται μέσα στο workspace. Έπειτα πατάμε πάνω στο Project μας και επιλέγουμε **File->New->Other->Android Activity** για να δημιουργήσουμε το Activity μας που εξηγήσαμε πόσο σημαντικό είναι στο Android.



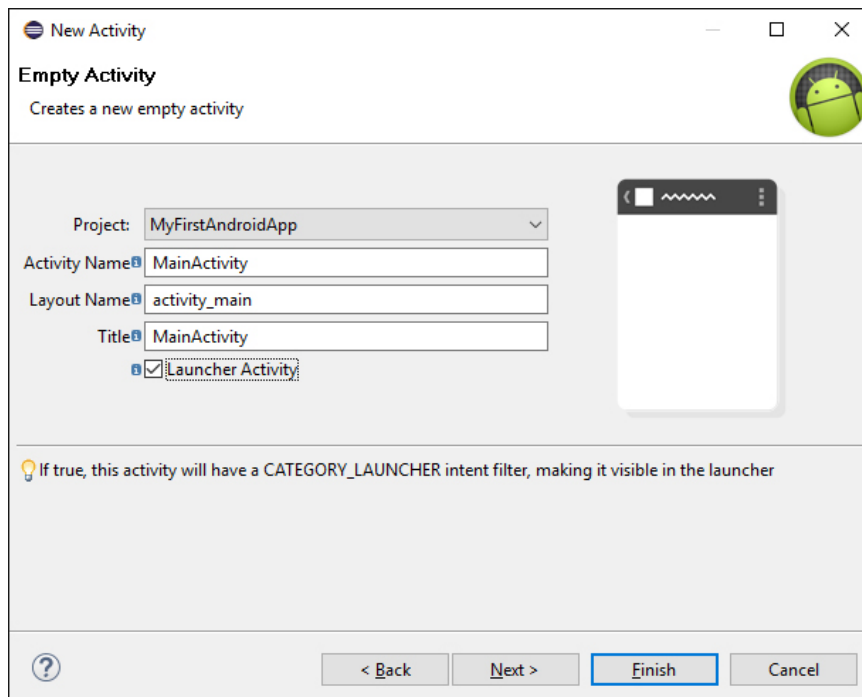
Εικόνα 27. Δημιουργία Activity.

Παρακάτω θα επιλέξουμε να ξεκινήσουμε με ένα κενό Activity το οποίο μπορούμε να διαμορφώσουμε εμείς.



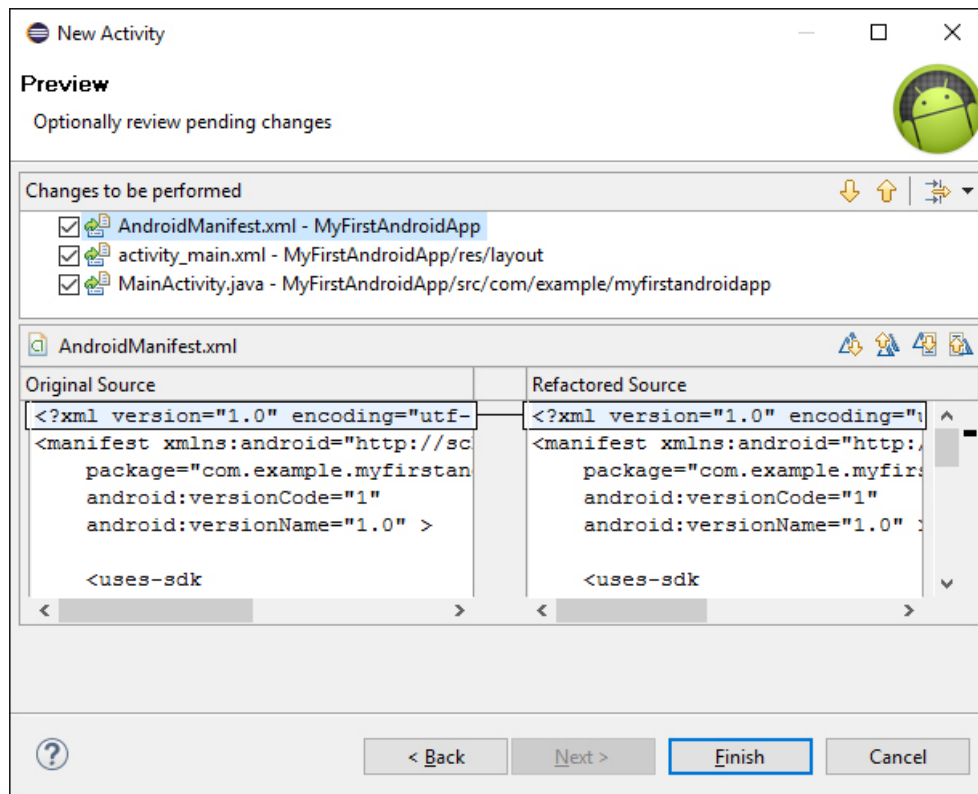
Εικόνα 28. Δημιουργία Activity – 2.

Έπειτα ονομάζουμε αυτά που θα δημιουργηθούν.



Εικόνα 29. Δημιουργία Activity – 3.

Στο τελευταίο βήμα μας γίνεται μια ανασκόπηση για τα αρχεία που θα δημιουργηθούν και αυτά που θα τροποποιηθούν.



Εικόνα 30. Δημιουργία Activity – 4.

Όπως βλέπουμε στην παραπάνω εικόνα μας εμφανίζεται σε ποιον φάκελο θα δημιουργηθεί κάθε αρχείο και το τι περιέχει ώστε να μπορούμε να κάνουμε τις τελευταίες αλλαγές. Στο `activity_main.xml` όπως θα αναλύσουμε παρακάτω έχει γραφτεί αυτόματα το “Hello World” σαν πρώτη μας εφαρμογή. Πατάμε “Finish” και όλα δημιουργούνται μέσα στο Project. Πηγαίνουμε να δούμε πρώτα το αρχείο `MainActivity.java` το οποίο είπαμε ότι είναι από τα πιο σημαντικά στοιχεία μας στο Android. Μέσα στο αρχείο που έχει δημιουργηθεί βρίσκεται ο παρακάτω κώδικας.

```
public class MainActivity extends Activity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```

MainActivity.java

Βλέπουμε αυτό που είχαμε περιγράψει πιο πάνω όταν μιλούσαμε για το Activity ότι η κλάση μας επεκτείνει την βασική κλάση Activity και πιο κάτω βλέπουμε ότι με την εντολή `setContentView` δίνουμε την όψη που θέλουμε στο παράθυρο που έχουμε δημιουργήσει. Στην συγκεκριμένη εντολή βλέπουμε ότι μέσω του `R.java` έχουμε πρόσβαση στο `activity_main.xml` που δημιουργήσαμε πιο πάνω και το βάζουμε σαν βασική όψη του παραθύρου μας. Είπαμε ότι μέσω του `R.java` που δημιουργείται αυτόματα μπορούμε να έχουμε πρόσβαση στα αρχεία που βρίσκονται στον φάκελο `res`. Βλέπουμε ότι το `activity_main.xml` έχει δημιουργηθεί μέσα στο φάκελο και πάμε να δούμε τι περιέχει το αρχείο.

```

<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"
    tools:context=".${relativePackage}.${activityClass}" >

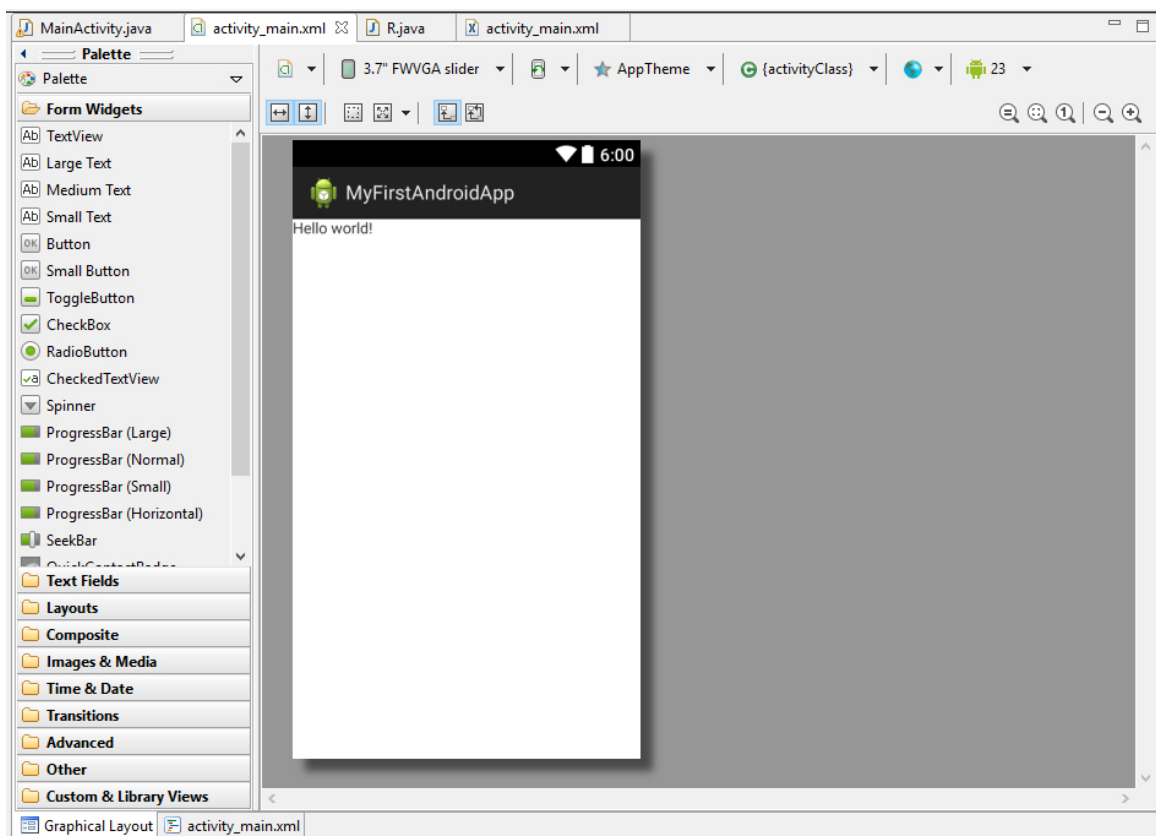
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />

</RelativeLayout>

```

activity_main.xml

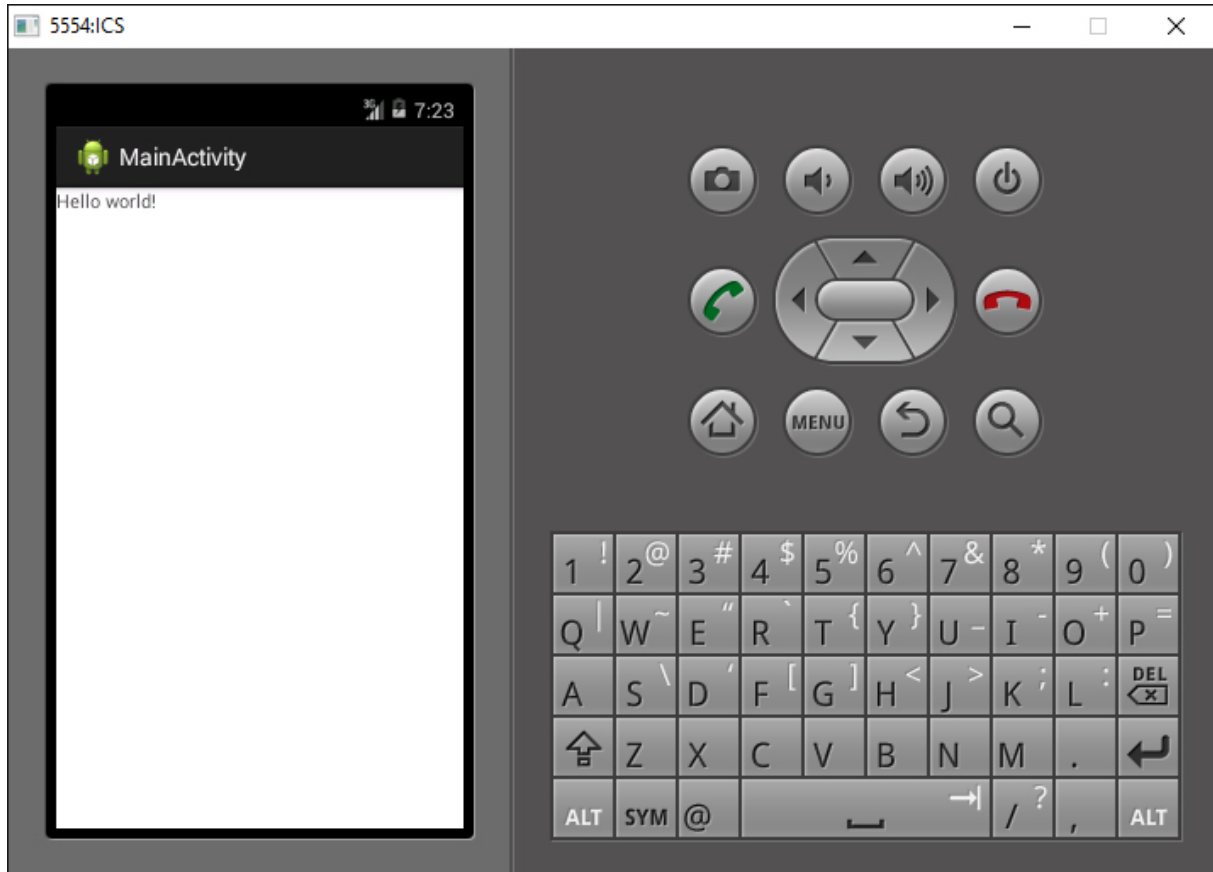
Μέσα στο xml απλά βλέπουμε ότι όταν το θέσουμε σαν View αυτό θα γεμίσει όλο το παράθυρο και επίσης ότι έχει προστεθεί και ένα text το οποίο καλεί το string “hello_world” που αν ανατρέξουμε μέσα στα strings που υπάρχουν μέσα στο strings.xml μέσα στα Resources βλέπουμε ότι περιέχει την πρόταση “Hello World!”. Αυτό μας δείχνει το πώς μπορούμε να έχουμε τον καταμερισμό των Resources, να δημιουργήσουμε τα δικά μας strings να τα αποθηκεύσουμε και να τα καλέσουμε όποτε τα χρειαστούμε στην εφαρμογή μας. Το Android μας δίνει κι άλλο ένα xml editor για να μας βοηθήσει στο να δημιουργήσουμε και να τροποποιήσουμε αρχεία xml όπως επίσης και ένα απευθείας γραφικό περιβάλλον για να απλοποιήσει την δουλειά μας.



Εικόνα 31. Xml σε UI.

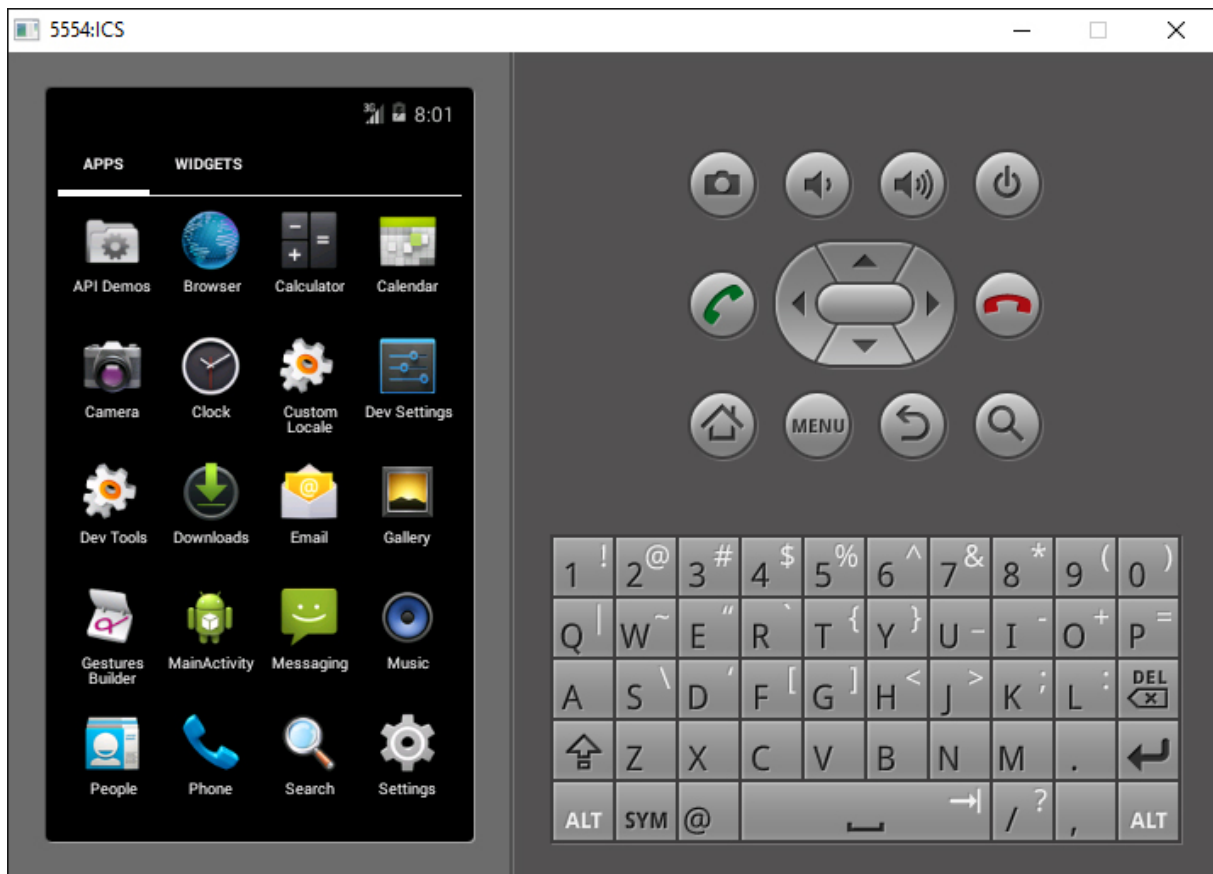
Στο τελικό μας βήμα εφόσον έχουμε δημιουργήσει μια εικονική συσκευή που να είναι συμβατή με την εφαρμογή που έχουμε δημιουργήσει πατάμε δεξί κλικ πάνω στο Project μας και επιλέγουμε **Run as -> Run Configurations**. Από εδώ μπορούμε να επιλέξουμε και να ρυθμίσουμε τις

τελικές παραμέτρους όπως το αν θα γίνει απλά η εγκατάσταση της εφαρμογής στο κινητό η θα τρέξουμε και το Activity κατευθείαν, ποια εικονική συσκευή θέλουμε να είναι προεπιλεγμένη για την εφαρμογή και διάφορα άλλα. Μόλις κάνουμε Run θα ξεκινήσει η αποσφαλμάτωση (debug) της εφαρμογής μας και θα γίνει η κατασκευή του .apk το οποίο θα εγκατασταθεί στην εικονική συσκευή. Ο Emulator εκκινείται και το αποτέλεσμα που παίρνουμε είναι το παρακάτω:



Εικόνα 32. Η πρώτη μας εφαρμογή τρέχει στον Emulator του Android.

Βλέπουμε αυτό που είχαμε σχεδιάσει από πριν μέσα στην εφαρμογή με μια μικρή διαφορά πάνω να γράφει MainActivity. Είναι το όνομα που είχαμε δώσει πριν όταν δημιουργούσαμε το Activity στο πεδίο “Title” στην Εικόνα 29. Στην ίδια εικόνα φαίνεται ότι έχουμε επιλέξει το Launcher Activity που σημαίνει ότι θα δημιουργηθεί εικονίδιο στον Launcher για να μπορούμε να εκκινούμε την εφαρμογή και με το εικονίδιο που έχουμε επιλέξει σε προηγούμενα βήματα όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 33. Η εφαρμογή μας στον Launcher.

6. EuroCancerComs και Android.

6.1 Τι είναι το EuroCancerComs.

Το EuroCancerComs είναι ένα Project που ξεκίνησε το 2009 το οποίο είχε σαν στόχο να εγκαθιδρύσει ένα ενοποιημένο μοντέλο για ένα Portal το οποίο θα έδινε την δυνατότητα σύνδεσης σε διαφορετικές κοινότητες που εμπλέκονται στην καταπολέμηση του καρκίνου, από ασθενείς και ιατρούς μέχρι και ερευνητικούς επιστήμονες. Η πρωτοβουλία ήταν του Ευρωπαϊκού Ινστιτούτου Ογκολογίας και επικεφαλής ήταν ο ECCO (European CanCer Organisation). Σκοπός του Project ήταν χρησιμοποιώντας την τεχνολογία του ιστού να ενοποιήσει κάτω από ένα Portal όλες αυτές τις διαφορετικές κοινότητες που υπάρχουν στην Ευρώπη και να τους προσφέρει την πρόσβαση σε πληροφορίες και την ικανότητα να προσθέσουν και να διαμοιραστούν τις δικές τους. Το EuroCancerComs εξέτασε τα θέματα που αφορούν την επικοινωνία και τη διάδοση των πληροφοριών για τον καρκίνο σε όλη την Ευρώπη, εντοπίστηκαν τα σημεία συμφόρησης και προτάθηκαν λύσεις. Το EuroCancerComs φρόντισε:

- να εξετάσει και να διαχωρίσει ποια είναι η ροή των πληροφοριών μέσα από επιστήμονες και από άλλους οργανισμούς υγείας,
- να αξιολογήσει τα αποτελέσματα των ερευνών και τον τρόπο τα οποία αυτά μεταφράζονται και διαδίδονται ώστε να ξεπεραστεί αυτό το εμπόδιο,
- να δημιουργήσει βάσεις δεδομένων για τις κλινικές δοκιμές και τις κατευθυντήριες γραμμές στις οποίες στις οποίες οι εμπλεκόμενοι, συμπεριλαμβανομένων των ασθενών, θα μπορούν να κάνουν αναζήτηση και να βρουν άμεσα αυτό που τους ενδιαφέρει,
- να χρησιμοποιήσει νέες τεχνολογίες που να έχουν βάση το Διαδίκτυο ώστε να κάνει ευκολότερη την χρήση και τον διαμοιρασμό των πληροφοριών.

Στο πλαίσιο αυτό επικεντρώθηκε ώστε η δημιουργία του Portal να περιλαμβάνει λογαριασμούς από τους χρήστες και να μπορεί ο κάθε χρήστης:

- να δημιουργήσει το προσωπικό του blog που θα έχει τον χώρο του να καταχωρήσει περιεχόμενο που θα είναι προσβάσιμο από τους υπόλοιπους,
- να δημιουργήσει διάφορες διασκέψεις, τηλεδιασκέψεις η και events,
- να ψάξει και να ενημερωθεί για κέντρα με ειδικότητα στον καρκίνο, ινστιτούτα και οργανισμούς υγείας σε ένα παγκόσμιο επίπεδο.

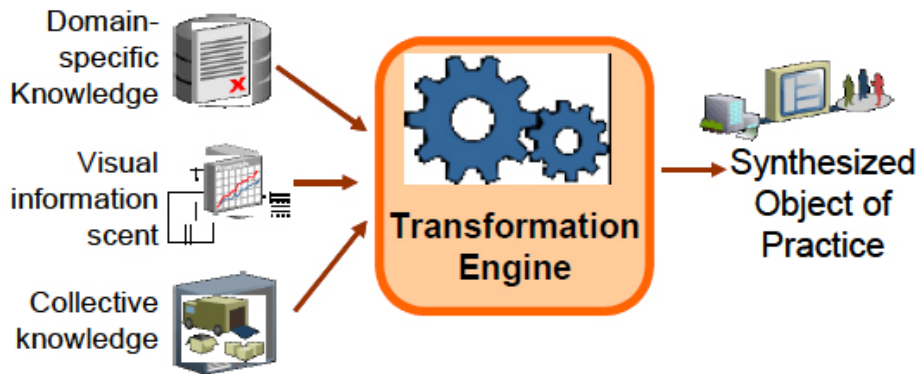
Επίσης το Portal θα παρείχε ένα φόρουμ όπου θα μπορούσε να γίνουν διάφορες συζητήσεις όσον αφορά το θέμα του καρκίνου και την δημιουργία ανοιχτών η κλειστών γκρουπ για πιο εξειδικευμένα θέματα που θα μπορούσαν να ασχοληθούν μόνο άτομα με κοινά ενδιαφέροντα.

6.2 Σκέψεις και ανάπτυξη στον ιστό.

6.2.1 Clinical practice guideline management information systems.

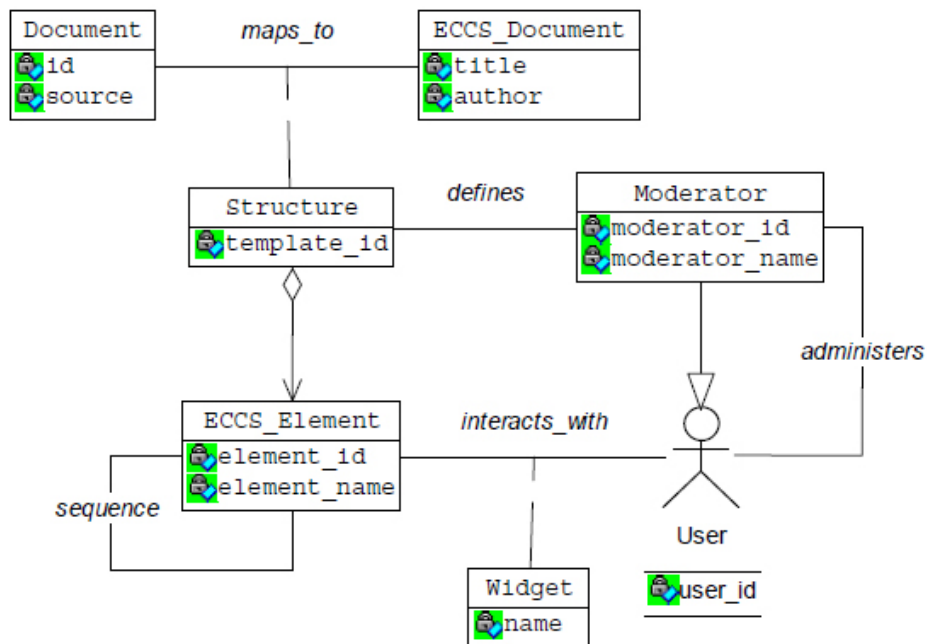
Η ιδέα πίσω από την ανάπτυξη της εφαρμογής μας η οποία έπρεπε να γίνει στο web για να είναι εύκολα και άμεσα προσβάσιμη από όλους ήταν να γίνει ένα σύστημα στο οποίο θα μπορεί να γίνει ένας διαχωρισμός των πληροφοριών και να μπορούμε να διαχειριζόμαστε εύκολα τα δεδομένα με μια απλοποιημένη διαδικασία για τις διαφορετικές ομάδες χρηστών. Αυτό που δόθηκε σαν αρχή στο Project ως κύριο αντικείμενο της έρευνας μας ήταν οι κλινικές μελέτες που έχουν αναπτυχθεί για τον καρκίνο. Στο θέμα αυτό έγινε προσπάθεια να προβάλλουμε αυτές τις κλινικές μελέτες κατηγοριοποιημένες δίνοντας σε κάθε άρθρο το χαρακτηριστικό μιας νέας δραστηριότητας και μια επεκτάσιμη μορφή με κοινωνικό χαρακτήρα. Προσπαθήσαμε να κατασκευάσουμε αντικείμενα τα οποία θα μπορούν να έχουν μια εναλλάξιμη μορφή και μέσα σε αυτά να προσδώσουμε τον κοινωνικό χαρακτήρα. Τα transformable boundary artifacts (διασυνοριακά τεχνουργήματα) είναι αυτά τα αντικείμενα τα οποία κατασκευάσαμε και χαρακτηρίζονται από 2 βασικά συστατικά:

1. Ευέλικτα πρότυπα
2. Social Widgets (Κοινωνικές Μικροεφαρμογές).



Εικόνα 34. Αρχιτεκτονική των Transformable Boundary Artifacts.

Βλέπουμε στην παραπάνω εικόνα ότι στην υλοποίηση αυτό που κάνουμε είναι να μαζέψουμε όλη την γνώση που υπάρχει γύρω από το αντικείμενο και να την εμπλουτίζουμε συνεχώς με καινούρια πράγματα και μέσα από μια μηχανή μετασχηματισμού να δίνουμε αυτό το σύνθετο αντικείμενο στο κοινό να αναπτυχθεί ακόμα παραπάνω χρησιμοποιώντας τα Social Widgets. Το τελικό αντικείμενο που θα παραχθεί είναι αυτό που θα παρουσιάσουμε με διαφορετικούς τρόπους γι αυτό θέλουμε να του προσδώσουμε μια πλαστικότητα.



Εικόνα 35. Transformable Boundary Artifacts.

Στην εικόνα 35 βλέπουμε πως μπορούμε να αναπαραστήσουμε σχηματικά αυτό το πράγμα την κατασκευή δηλαδή ενός document με μια συγκεκριμένη δομή μέσα από ένα πρότυπο (template) που ορίζει τον διαχειριστή του (moderator) και έχει αλληλεπίδραση (interaction) μέσω του χρήστη (user) με τα Widgets.

6.2.2 Social Widgets.

Τα social widgets είναι ένα από τα πιο βασικά κομμάτια της ανάπτυξης των κοινωνικών δικτύων τα τελευταία χρόνια. Είναι ένα τμήμα που δίνει ζωή σε όλο τον ιστό και την συνεχόμενη εξέλιξη του. Τα social widgets δίνουν την δυνατότητα στους χρήστες να αλληλεπιδράσουν μεταξύ τους και να τους δώσουν παραπάνω πληροφορίες γι αυτό που έχουν μπροστά τους. Είναι παντού, σε ένα βίντεο στο YouTube με την μορφή ψήφου, στο Facebook με την μορφή Like, στο Twitter με την μορφή σχολίου, στο Imdb με την μορφή βαθμολογίας, στο TripAdvisor με την μορφή κριτικής και σε πάρα πολλά άλλα με πάρα πολλές μορφές. Εφόσον η σκέψη μας λοιπόν στην υλοποίηση της εφαρμογής είναι να δώσουμε την ικανότητα στους χρήστες να αλληλεπιδράσουν μεταξύ τους ώστε να έχουμε το μέγιστο των πληροφοριών θα έπρεπε κι εμείς να προσθέσουμε αυτά τα widgets στην εφαρμογή μας. Πάμε να δούμε παρακάτω μια λίστα από αυτά τα widgets που έγιναν διαθέσιμα στην web εφαρμογή:

- **Tag Widget:**
Επιτρέπει στους χρήστες να χρησιμοποιούν λέξεις κλειδιά ώστε να χαρακτηρίσουν το κείμενο και να μπορεί ο επόμενος χρήστης να δει γρήγορα τι περιέχει.
- **Rate Widget:**
Οι χρήστες μπορούν να βαθμολογήσουν το αντικείμενο.
- **Word Cloud Widget:**
Ένα γραφικό στοιχείο που δημιουργεί ένα συννεφάκι από λέξεις με τις λέξεις που έχουν χρησιμοποιηθεί περισσότερο στο κείμενο να είναι πιο εμφανείς και να μπορούμε να πραγματοποιούμε αντίστροφη αναζήτηση στον ιστό.
- **Comment Widget:**
Οι χρήστες μπορούν να σχολιάσουν για το αντικείμενο.
- **Voting Widget:**
Επιτρέπει στους χρήστες να δημιουργούν μικρές ψηφοφορίες και έρευνες.
- **Dispute Widget:**
Με αυτό το widget δίνουμε την δυνατότητα στον χρήστη να διατυπώσει μια διαφωνία στο αντικείμενο και να μπορέσει να ξεκινήσει ένα θέμα στο οποίο θα μπορούν να προταθούν λύσεις.
- **Uploading Widget:**
Επιτρέπουμε στους χρήστες να μεταφορτώσουν πολυμέσα όπως σε ένα θέμα που έχει δημιουργηθεί η αντίστοιχα documents.
- **Questionnaire / assessment widget:**
Οι χρήστες έχουν την δυνατότητα να κατασκευάσουν ένα μικρό ηλεκτρονικό ερωτηματολόγιο με προδιαγραφές XML.

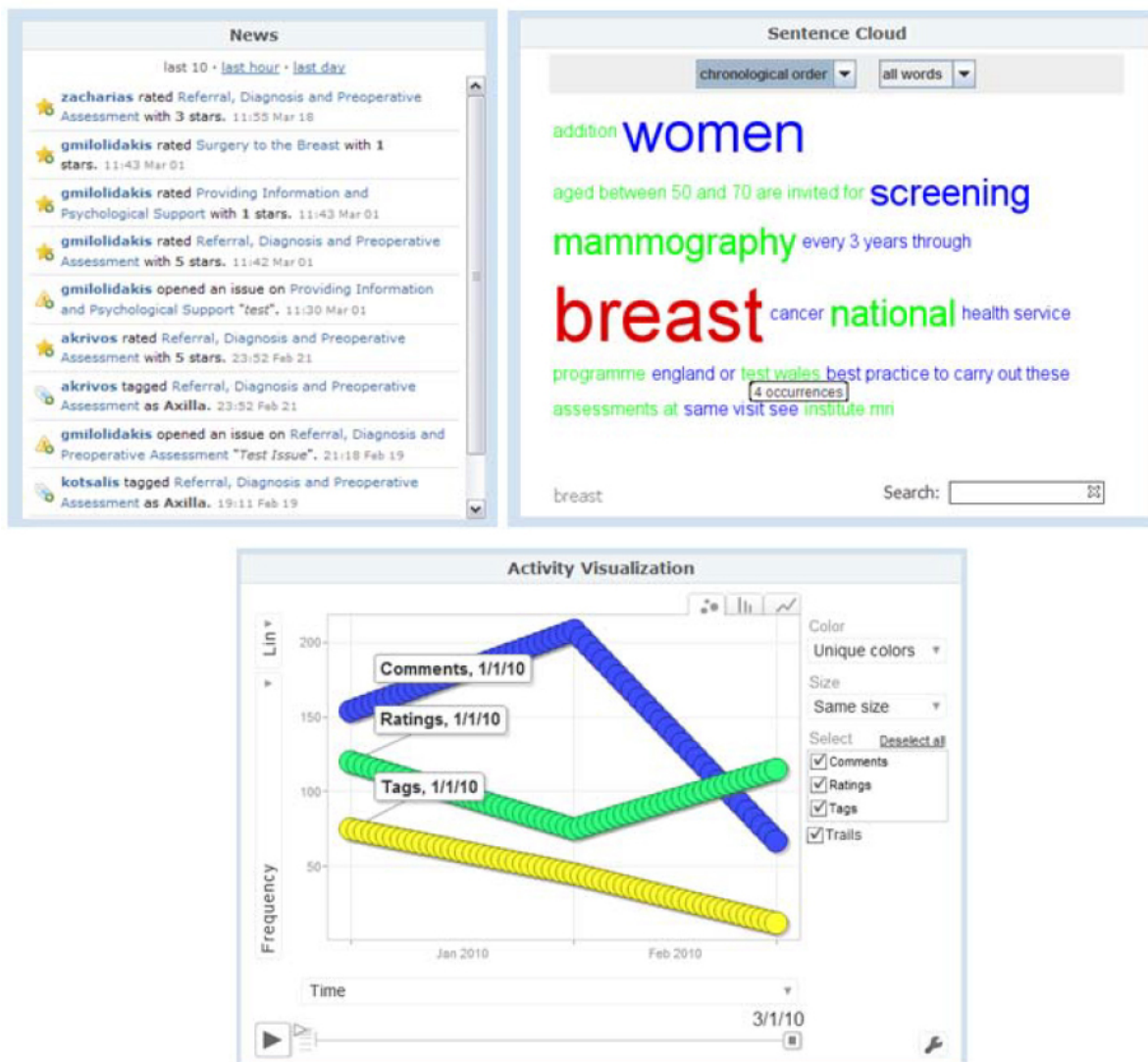
The screenshot displays the web version of the Guideline Management Tool. The main document is titled "Early and locally advanced breast cancer. Diagnosis and treatment." and is categorized under "Referral, Diagnosis and Preoperative Assessment". The document content includes text about breast screening programs and clinical assessment methods. Several widgets are overlaid on the document:

- Tags:** A box containing "assessments Axilla breast cancer BTWSP DCIS Diagnosis disease examination".
- Comments (5):** A list of user comments with their names, dates, and content. For example, "vellis" comments "more multimedia content should be inserted." on Fri, 19 Feb 2010 18:10.
- Issues:** A list of user-reported issues with their names, dates, and descriptions. For example, "insufficient data" by "vellis" on Fri, 19 Feb 2010 18:29.
- Psychological distress inventory (PDI) questionnaire:** A form with four questions and radio button options. The first question is "1. Do you think your desire to speak with others has diminished?" with options "Not at all", "A little little", "Quite a bit", "Much", and "Very much". The second question is "2. Have you felt calm?" with the same options. The third question is "3. Have you experienced moments of anxiety or inner tension?" with options "Not at all", "A little little", "Quite a bit", "Much", and "Very much". The fourth question is "4. Have you felt tired or lacking in energy?" with the same options. The questionnaire shows a progress bar for each question.
- Word Cloud:** A cloud of words related to the document, including "breast cancer screening", "preoperative assessment", "clinical", "invasive", "mammography", "patients", "Axilla", "NICE", "National Programme Service", "Wales", "needle", "BTWSP", "DCIS", "England", "Eoslerne".
- Google Search:** A search bar with results for "screening" and "screening - wikipedia, the encyclopedia".

Εικόνα 36. Web version of Guideline management tool with widgets.

Στην παραπάνω εικόνα βλέπουμε την λειτουργία του Guideline Management Tool που ενσωματώνει μια λίστα πάνω με όλα τα διαθέσιμα documents. Όταν επιλέξουμε το document που θέλουμε να διαβάσουμε και πατήσουμε το "Load Document" φορτώνεται το κείμενο και ταυτόχρονα από δίπλα φορτώνονται και όλα τα social widgets που είναι διαθέσιμα για το συγκεκριμένο document και να δούμε τα tags, το rating, τα σχόλια και να έχουμε άμεση αλληλεπίδραση κατευθείαν πάνω σε αυτό που διαβάζουμε, να συνεχίσουμε με τα σχόλια ή τα issues και ότι άλλο θέλουμε.

Άλλο ένα κομμάτι που μπορούμε να ενσωματώσουμε στο εργαλείο είναι widgets τα οποία δεν θα έχουμε άμεση αλληλεπίδραση αλλά θα μας δίνουν αποτελέσματα. Αυτό είναι πιο πολύ χρήσιμο όχι για τους τελικούς χρήστες αλλά για τους διαχειριστές για να βλέπουν τι κίνηση υπάρχει και για ποιο κομμάτι ενδιαφέρεται πιο πολύ ο κόσμος. Έτσι μπορούμε να προβούμε σε βελτιώσεις και καλύτερες περαιτέρω τροποποιήσεις. Στις παρακάτω εικόνες απεικονίζονται κάποια από αυτά τα widgets.



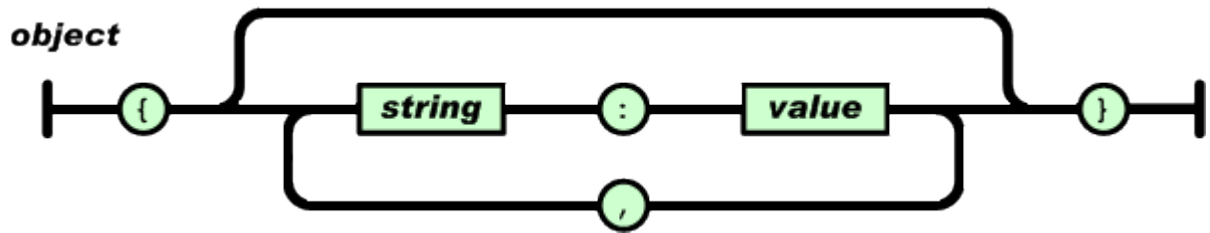
Εικόνα 37. Result Widgets.

6.3 Το EuroCancerComs στο Android.

6.3.1 JSON και ευελιξία στο EuroCancerComs.

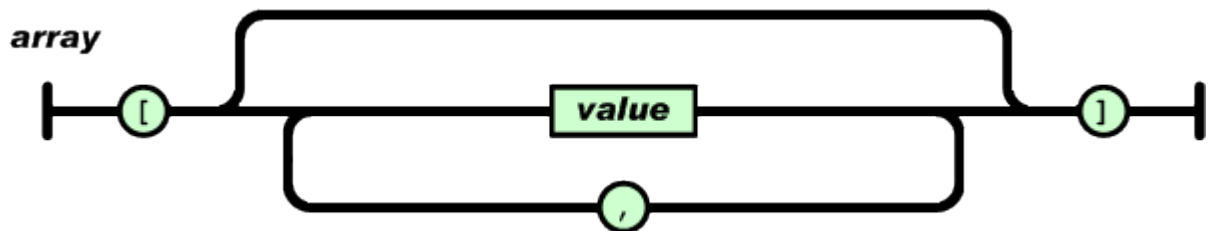
Το JSON (JavaScript Object Notation) είναι ένα ελαφρύ πρότυπο ανταλλαγής δεδομένων, το οποίο είναι βασισμένο πάνω σε ένα υποσύνολο της γλώσσας προγραμματισμού JavaScript. Είναι πολύ εύκολο για τις μηχανές να το αναλύσουν (parse) και να το παράγουν (generate). Το JSON είναι ένα πρότυπο κειμένου το οποίο είναι τελείως ανεξάρτητο από γλώσσες προγραμματισμού αλλά χρησιμοποιεί πρακτικές οι οποίες είναι γνωστές στους προγραμματιστές της οικογένειας προγραμματισμού C, συμπεριλαμβανομένων των C, C++, C#, Java, JavaScript, Perl, Python, και πολλών άλλων. Αυτές οι ιδιότητες κάνουν το JSON μια ιδανική γλώσσα προγραμματισμού ανταλλαγής δεδομένων. Το JSON χρησιμοποιεί δυο δομές, είτε να είναι χτισμένο με καταχωρήσεις σε ζευγάρια ονομάτων/τιμών που οι πιο πολλές γλώσσες προγραμματισμού το αντιλαμβάνονται σαν ένα αντικείμενο (objet), καταχώριση, δομή, λεξικό, πίνακα hash (hash table), λίστα κλειδιών, είτε σαν μια ταξινομημένη λίστα τιμών που οι περισσότερες γλώσσες προγραμματισμού το αντιλαμβάνονται σαν ένα πίνακα (array), διάνυσμα, λίστα, ή ακολουθία.

Στην πρώτη περίπτωση που έχουμε ένα object η δομή μας ξεκινάει με { (αριστερό άγκιστρο) και τελειώνει με } (δεξιό άγκιστρο). Κάθε όνομα ακολουθείται από : (άνω-κάτω τελεία) και κάθε ζευγάρι ονόματος/τιμής χωρίζεται από , (κόμμα).



Εικόνα 38. Δομή του JSON ως object.

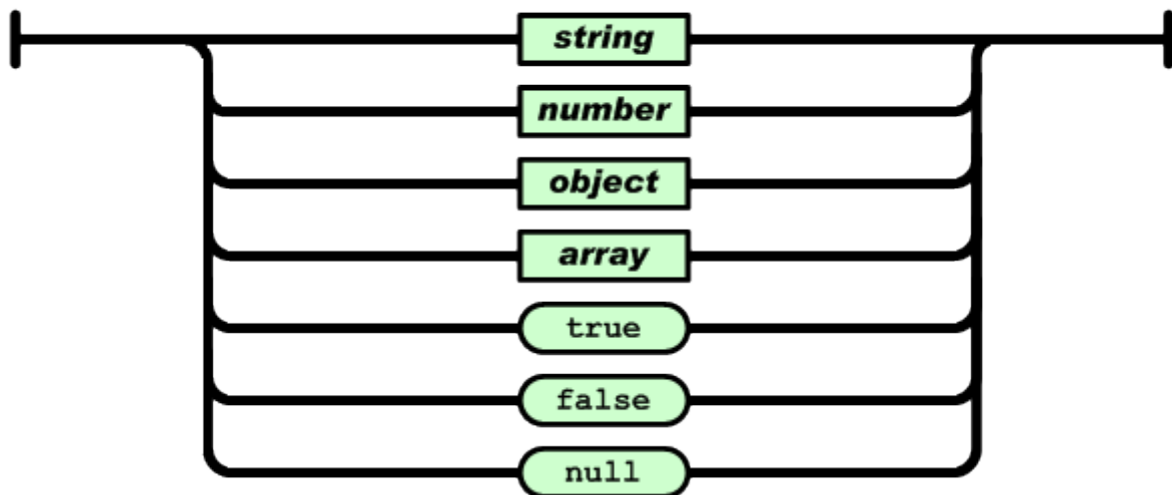
Στην δεύτερη περίπτωση έχουμε έναν πίνακα (array) που είναι μια συλλογή από τιμές σε σειρά. Ένας πίνακας ξεκινάει με [(αριστερή αγγύλη) και τελειώνει με] (δεξιά αγγύλη). Οι τιμές χωρίζονται με , (κόμμα).



Εικόνα 39. Δομή του JSON ως array.

Μία τιμή (value) μπορεί να είναι string μέσα σε διπλά quotes, ή αριθμός (number), ή true ή false ή null, ή αντικείμενο (object) ή πίνακας (array). Αυτές οι τιμές μπορεί να είναι και ανακατεμμένες.

value



Εικόνα 40. Περιεχόμενα του value.

Η απλότητα και η ευκολία του JSON είναι αυτή που μας οδήγησε στην επιλογή του ώστε να χρησιμοποιήσουμε αυτή την μορφή για να έχουμε τα δεδομένα και έπειτα να μπορέσουμε να τα κάνουμε parse και να τα απεικονίσουμε όπως εμείς θέλουμε, είτε σε μια web έκδοση είτε σε ένα smartphone. Αυτό που ήταν βολικό με το JSON ήταν ότι μπορούσαμε να έχουμε όλα τα δεδομένα είτε πρόκειται για κάποιο document είτε πρόκειται για δεδομένα που αφορούν social widget σε αυτή την μορφή και να τα χρησιμοποιούμε στην γραφική απεικόνιση.

6.3.2 Parsing του JSON.

Στο κομμάτι της web εφαρμογής υπήρχε ήδη προκατασκευασμένη η διαδικασία της αποθήκευσης των δεδομένων μέσα σε μια βάση και από εκεί και πέρα το εργαλείο για να κατασκευαστούν τα JSON αρχεία. Εμείς αυτό που πρέπει να δούμε από την μεριά μας στο Android είναι η ανάλυση των δεδομένων του JSON που έχει παραχθεί και η γραφική απεικόνιση τους στην κινητή συσκευή μας.

```
[
  {
    "comId": 39,
    "userName": "kotsalis",
    "comContent": "the link is broken",
    "comParentId": -1,
    "comCreationDate": "Fri, 19 Feb 2010 19:00"
  },
  {
    "comId": 37,
    "userName": "vellis",
    "comContent": "more multimedia content should be
inserted.",
    "comParentId": -1,
    "comCreationDate": "Fri, 19 Feb 2010 18:10"
  },
  {
    "comId": 36,
    "userName": "vidakis",
    "comContent": "too scientific",
    "comParentId": -1,
    "comCreationDate": "Fri, 19 Feb 2010 18:00"
  },
  {
    "comId": 35,
    "userName": "anargyros",
    "comContent": "Breast cancer is the second most common
type of non-skin cancer after lung cancer and the fifth
most common cause of cancer death.",
    "comParentId": -1,
    "comCreationDate": "Fri, 19 Feb 2010 17:52"
  },
  {
    "comId": 34,
    "userName": "milolidakis",
    "comContent": "the image is too small",
    "comParentId": -1,
    "comCreationDate": "Fri, 19 Feb 2010 17:46"
  }
]
```

comment.json

Στο παραπάνω παράδειγμα βλέπουμε ένα JSON το οποίο έχει δημιουργηθεί και αφορά τα σχόλια που έχουν γίνει σε ένα document. Όπως εξηγήσαμε πιο πάνω το JSON αυτό είναι με την μορφή array γιατί βλέπουμε ότι ξεκινάει με αγκύλη και μέσα βλέπουμε τα περιεχόμενα του είναι σε μορφή αντικειμένου. Το parse του συγκεκριμένου JSON θα μας δώσει την παρακάτω γραφική απεικόνιση στο Android σύμφωνα πάντα με τις δικές μας επιλογές και μορφοποιήσεις.



Εικόνα 41. Comment Display on Android.

Όλα τα σχόλια όπως τα βλέπουμε στο JSON έχουν φορτωθεί στην συσκευή μας και μπορούμε να κάνουμε κύλιση (scroll) πάνω κάτω με το δάχτυλο μας για να τα διαβάσουμε με μια ταξινόμηση βάσει της ημερομηνίας καταχώρησης. Κάτω από τα σχόλια βρίσκουμε ένα Text Field, ένα πεδίο που μπορούμε να πληκτρολογήσουμε το δικό μας σχόλιο και να το δημοσιεύσουμε μαζί με τα άλλα. Κάτω από το Text Field βρίσκουμε δύο κουμπιά το ένα για να γίνει η δημοσίευση και το άλλο για να κλείσουμε τα σχόλια και να επιστρέψουμε στο κείμενο.

Πάμε να δούμε μέσα στον κώδικα μας πως γίνεται να δημιουργήσουμε αυτό το παράθυρο και πως θα αναλύσουμε τα δεδομένα που παίρνουμε από το JSON για να τα παρουσιάσουμε με αυτό τον τρόπο. Βλέπουμε στην εικόνα 41 ότι το παράθυρο που εμφανίζει τα δεδομένα μας έχει εμφανιστεί επάνω από το παράθυρο που ήμασταν πριν, το οποίο είχε το document που διαβάζαμε και από κάτω τα κουμπιά για τα αντίστοιχα tags, rates, comments. Το προηγούμενο φαίνεται στο παρασκήνιο αχνά πίσω και το άλλο που έχει έρθει από πάνω το βρίσκουμε στο Android ως Dialog box (παράθυρο διαλόγου). Επιλέξαμε αυτού του είδους την εμφάνιση για όλα τα social widgets πατώντας το αντίστοιχο κουμπί ώστε να μην χάνουμε την επαφή με το κείμενο που διαβάζουμε καθώς όπως βλέπουμε στην εικόνα, ο τίτλος του κειμένου του οποίου τα σχόλια έχουμε ανοίξει είναι αρκετά ευδιάκριτος από πίσω.

```

final Dialog comments = new Dialog(this);
comments.setTitle("Comments");
comments.setContentView(R.layout.commentsdialog);
Button commentsbutton = (Button)
findViewById(R.id.comment);
Button commentscancel = (Button) comments
    .findViewById(R.id.commentscancel);
TableLayout table = (TableLayout) comments
    .findViewById(R.id.TableLayout01);

```

Δημιουργία του Dialog “Comments”.

Στο παραπάνω κομμάτι κώδικα βλέπουμε την δημιουργία του Dialog Box που φαίνεται στην εικόνα 41. Βάζουμε τον τίτλο “Comments” με το “set.Title” και εμφανίζεται με άσπρα έντονα γράμματα στην κορυφή του Dialog, ορίζουμε το View δημιουργούμε τα κουμπιά και δίνουμε ένα συγκεκριμένο layout.

Στο επόμενο σημείο αυτό που κάνουμε είναι να δημιουργήσουμε μια συνάρτηση για το parse των δεδομένων που υπάρχουν στο JSON. Για να το κάνουμε αυτό πρέπει να χτυπήσουμε ένα συγκεκριμένο URL που βρίσκεται το JSON. Το Android μας βοηθάει με τις συναρτήσεις που έχει έτοιμες για αυτή την δουλειά για να ανοίξουμε ένα stream και να έχουμε ροή των δεδομένων όπως βλέπουμε παρακάτω. Από εκεί και πέρα χρησιμοποιούμε ακόμα μια δική μας συνάρτηση την “convertStreamToString” η οποία μετατρέπει το stream σε συμβολοσειρά (string).

```

void parseComments() {

    HttpClient httpclient = new DefaultHttpClient();
    HttpPost post = new HttpPost(
        "http://users.otenet.gr/~hayate7/json/comment");
    try {

        HttpResponse response = httpclient.execute(post);
        HttpEntity entity = response.getEntity();
        InputStream instream = entity.getContent();
        String commentstream =
            convertStreamToString(instream);
        JSONArray comments = new JSONArray(commentstream);
        for (int i = 0; i < comments.length(); i++) {
            JSONObject co = comments.getJSONObject(i);

            commentDate.add(co.getString("comCreationDate"));
            comUserName.add(co.getString("userName"));
            comContent.add(co.getString("comContent"));
        }

    } catch (IOException ex) {
        System.out.println("Connection failed: " +
            ex.getMessage());
    } catch (JSONException ex) {
        System.out.println("JSON failed: " +
            ex.getMessage());
    }
}

```

Parse των δεδομένων από το JSON.

Χρησιμοποιώντας ένα βρόγχο μέσα στην συνάρτηση μας μπορούμε να πάρουμε τα δεδομένα που υπάρχουν στο JSON και να τα αποθηκεύσουμε σε μια μεταβλητή που έχουμε δηλώσει εμείς σε μορφή Array γιατί έχουμε πολλά δεδομένα. Με αυτό τον τρόπο μπορούμε να πάρουμε την τιμή του value “username” και να την αποθηκεύσουμε στο Array μας “comUserName”. Στο τέλος έχουμε δηλώσει τα απαραίτητα exception που χρειάζονται σε περίπτωση που δεν υπάρχει σύνδεση ή δεν βρεθεί το JSON.

```
ArrayList<String> commentDate = new ArrayList<String>();  
ArrayList<String> comUserName = new ArrayList<String>();  
ArrayList<String> comContent = new ArrayList<String>();
```

Η δήλωση των μεταβλητών έχει γίνει ως ArrayList που περιέχουν String.

```
for (int j = 0; j < commentDate.size(); j++) {  
    System.out.println(commentDate.get(j).toString());  
    System.out.println(comUserName.get(j).toString());  
    System.out.println(comContent.get(j).toString());  
  
    View thinLine = new View(getApplicationContext());  
    thinLine.setBackgroundColor(Color.GRAY);  
    thinLine.setLayoutParams(new  
        ViewGroup.MarginLayoutParams(320, 1));  
  
    View bMargin = new View(getApplicationContext());  
    bMargin.setLayoutParams(new  
        ViewGroup.MarginLayoutParams(320, 8));  
  
    TableRow row01 = new TableRow(this);  
  
    TextView cname = new TextView(this);  
    cname.setTypeface(null, Typeface.BOLD);  
    cname.setText(comUserName.get(j).toString());  
    cname.setPadding(5, 0, 0, 0);  
  
    TextView cdate = new TextView(this);  
    cdate.setTextSize(11);  
    cdate.setGravity(Gravity.RIGHT | Gravity.TOP);  
    cdate.setText(commentDate.get(j).toString());  
    cdate.setPadding(0, 0, 5, 0);  
  
    TextView cContent = new TextView(this);  
    cContent.setTextSize(14);  
    cContent.setText(comContent.get(j).toString());  
    cContent.setPadding(5, 0, 0, 0);  
  
    row01.addView(cname, new TableRow.LayoutParams(1));  
    row01.addView(cdate, new TableRow.LayoutParams());  
  
    table.addView(row01, new TableLayout.LayoutParams());  
    table.addView(thinLine);  
    table.addView(cContent, new TableLayout.LayoutParams());  
    table.addView(bMargin);  
}
```

Γέμισμα του Dialog Box με τα σχόλια.

Στο τέλος αυτό που κάνουμε χρησιμοποιώντας έναν βρόγχο και πάλι κάνουμε την παραμετροποίηση για την εμφάνιση των σχολίων και προσθέτουμε τα δεδομένα που έχουμε πάρει από πριν, με τον τρόπο που δείξαμε, στον πίνακα “table” που δημιουργήσαμε στην αρχή μέσα στην δημιουργία του Dialog Box. Την ίδια ακριβώς διαδικασία ακολουθούμε και για τα υπόλοιπα Social Widgets και γενικά για την διαδικασία μεταφοράς και απεικόνισης των δεδομένων του

EuroCancerComs σε κάθε κομμάτι της εφαρμογής που θέλουμε να εμφανίσουμε στην κινητή συσκευή. Όλα μας τα δεδομένα είναι σε JSON οπότε όπως θα δούμε και παρακάτω στην παρουσίαση της εφαρμογής κάνουμε το ίδιο για να εξερευνήσουμε τα διαθέσιμα documents, να επιλέξουμε κάποιο για να διαβάσουμε και έπειτα να έχουμε αλληλεπίδραση με αυτό.

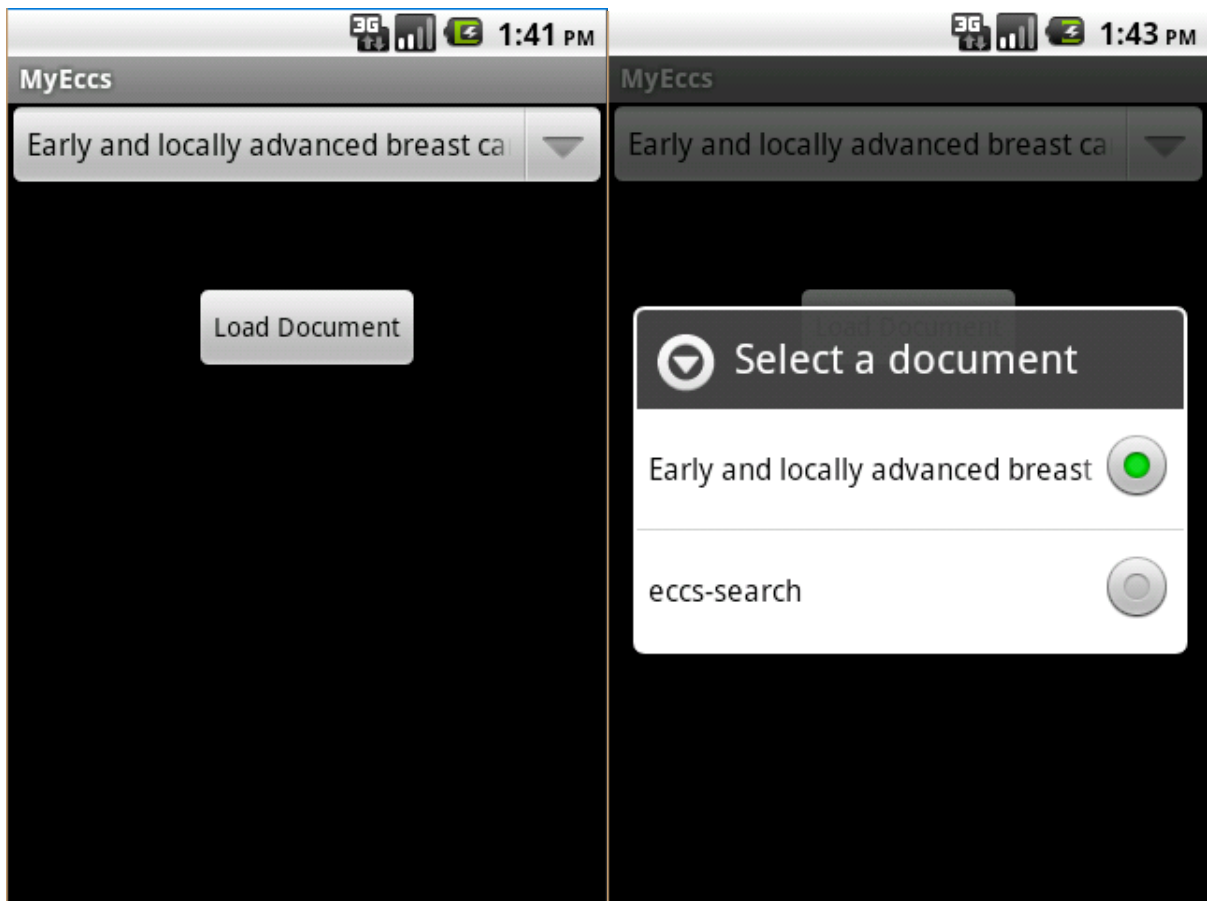
6.3.3 Παρουσίαση της εφαρμογής.

Στο EuroCancerComs εξετάσαμε τα πιθανά σενάρια χρήσης ώστε το δίκτυο να είναι ανοικτό και να μην χρειάζεται κάποιος να έχει λογαριασμό για να έχει πρόσβαση στα διαθέσιμα documents. Ακόμα και σε αυτό το σενάριο θα χρειαστεί ένας λογαριασμός αν ο χρήστης επιθυμεί να συμμετέχει στο δίκτυο χρησιμοποιώντας τα Social Widgets. Για το σενάριο χρήσης που ο λογαριασμός ήταν απαραίτητος για κάθε χρήση θα ξεκινούσαμε μέσα στην εφαρμογή με μια “Login Screen”.



Εικόνα 42. Login Screen.

Με μια απλή διαδικασία ο χρήστης του οποίου η ταυτοποίηση υπάρχει στον Server, δεν έχουμε προσθέσει δηλαδή ακόμη διαδικασία εγγραφής χρήστη από εδώ, μπορεί να εισάγει το όνομα χρήστη και τον κωδικό που του έχει δοθεί και να κάνει την αρχική εισαγωγή μέσα στην εφαρμογή. Η παραπάνω οθόνη δεν βρίσκεται στην τελική μας εφαρμογή την αναφέρουμε όμως γιατί θα μπορούσε να υπάρχει μια πιστοποίηση εξαρχής ώστε να μπορούμε να έχουμε και τον έλεγχο και τις χρήσιμες αναλύσεις για το ποιοι χρήστες έχουν χρησιμοποιήσει τις κινητές συσκευές και τι έχουν κάνει από εκεί ώστε να μας βοηθήσει περαιτέρω στην έρευνα μας και στην βελτιστοποίηση τη εφαρμογής. Σε κάθε περίπτωση στόχος μας για να ξεκινήσουμε είναι η εξερεύνηση των documents που ξεκινάει μέσα από την επόμενη μας οθόνη.



Εικόνα 43. Επιλογή Document μέσα από λίστα.

Μέσα από μια αναπτυσσόμενη λίστα επιλέγουμε το document που θέλουμε να φορτώσουμε από τον Server και πατάμε Load Document. Σε περίπτωση που ο τίτλος είναι πολύ μεγάλος όπως στο πρώτο document το κείμενο γίνεται scroll για να το διαβάσουμε όλο. Αν γυρίσουμε λίγο πίσω στην εικόνα 36 που είχαμε την αναπαράσταση του guideline management tool θα βλέπαμε ότι το ίδιο γίνεται και στον ιστό. Έχουμε επάνω δεξιά μια αναπτυσσόμενη λίστα που επιλέγουμε το document που θέλουμε και πατάμε Load. Στο επόμενο ακριβώς κομμάτι θα φορτώσουμε τις διαφορετικές καρτέλες που φαίνονται στο εργαλείου του ιστού και αμέσως μετά επιλέγοντας καρτέλα πάμε στο επόμενο κομμάτι που είναι όλα τα άρθρα που περιέχει η καρτέλα. Επιλέγουμε μέσα στις λίστες δηλαδή μέχρι να φτάσουμε στο τελικό άρθρο που θέλουμε και μέσα σε αυτό το άρθρο θα προβάλλουμε όλα τα social widgets όπως δείξαμε και πριν.

MyEccs	MyEccs
RECOMMENDATIONS	Referral, Diagnosis and Preoperative Assessment
EVIDENCE SUPPORTING THE RECOMMENDATIONS	Providing Information and Psychological Support
IDENTIFYING INFORMATION AND AVAILABILITY	Surgery to the Breast
DISCLAIMER	Surgery to the Axilla
	Postoperative Assessment and Adjuvant Therapy Planning

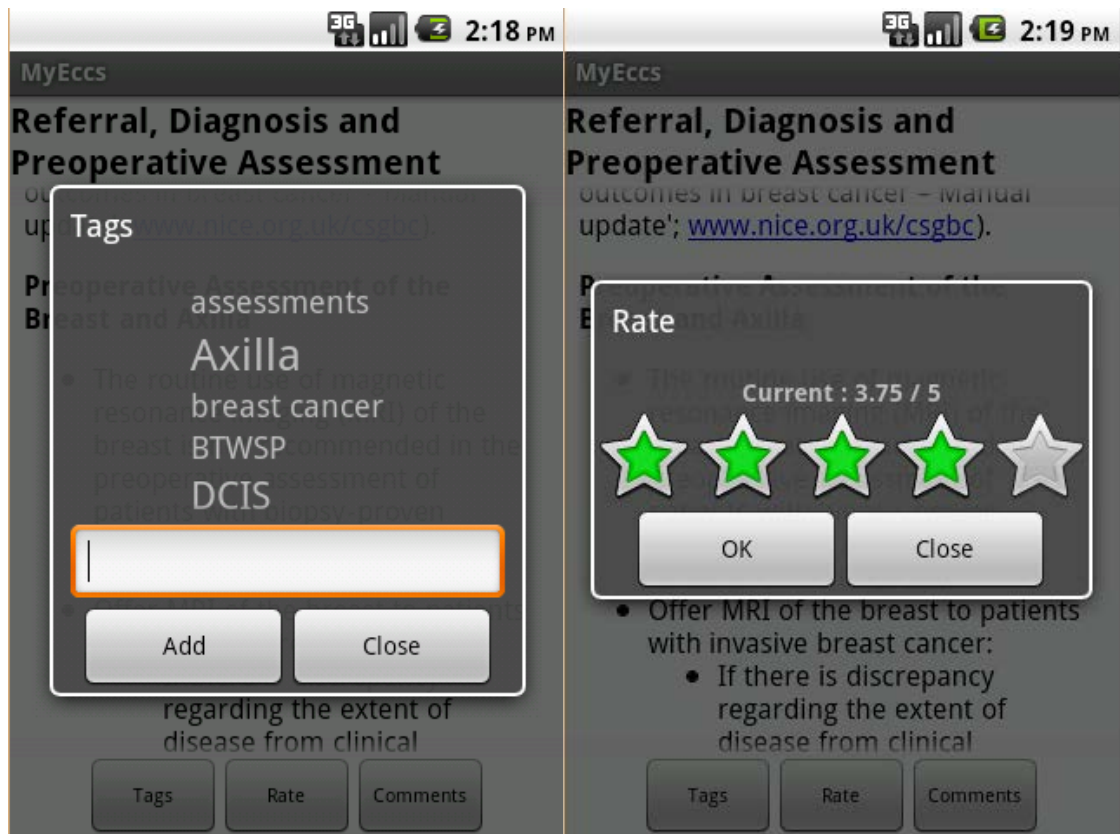
Εικόνα 44. Επιλογή καρτέλας και στην συνέχεια άρθρου.

The screenshot shows the MyEccs mobile application interface. At the top, the status bar displays '3G', signal strength, battery, and the time '2:13 PM'. Below the status bar, the app header reads 'MyEccs'. The main content area features the title 'Referral, Diagnosis and Preoperative Assessment' in bold black text. Below the title, there is a subtitle 'outcomes in breast cancer – Manual update'; www.nice.org.uk/csgbc. The main heading is 'Preoperative Assessment of the Breast and Axilla'. The content consists of a bulleted list:

- The routine use of magnetic resonance imaging (MRI) of the breast is not recommended in the preoperative assessment of patients with biopsy-proven invasive breast cancer or ductal carcinoma in situ (DCIS).
- Offer MRI of the breast to patients with invasive breast cancer:
 - If there is discrepancy regarding the extent of disease from clinical

At the bottom of the article view, there are three buttons: 'Tags', 'Rate', and 'Comments'.

Εικόνα 45. Προβολή άρθρου.



Εικόνα 46. Προβολή Social Widgets.

7. Συμπεράσματα και προτάσεις για βελτίωση της εφαρμογής.

7.1 Συμπεράσματα.

Ολοκληρώνοντας την εφαρμογή μας αυτό που μπορούμε να πούμε είναι ότι πετύχαμε τον αρχικό μας στόχο να προσδώσουμε στο κοινωνικό μας δίκτυο την φορητότητα (portability) που χρειαζόταν και να έχουμε έτσι μια ηλεκτρονική κοινότητα στην οποία θα μπορεί να υπάρχει άμεση πρόσβαση από παντού. Η κινητή συσκευή χρησιμοποιείται πλέον αποκλειστικά από αρκετό κόσμο για την πρόσβαση στο διαδίκτυο και για την επαφή σε τέτοιου είδους κοινωνικά δίκτυα οπότε ήταν αναγκαία η κατασκευή μιας εφαρμογής για αυτό τον σκοπό. Στόχος μας ήταν επίσης η μεταφορά στην κινητή συσκευή να είναι όσο πιο κοντά γίνεται στην απεικόνιση του ιστού έτσι ώστε να μπορεί ο κάθε χρήστης να κινηθεί με τον ίδιο τρόπο το ίδιο εύκολα μέσα στο δίκτυο και να αντλήσει τις πληροφορίες που θέλει. Έτσι κι αλλιώς θέλαμε το δίκτυο μας να απευθύνεται σε απλούς χρήστες που σημαίνει ότι δεν έχουν σχετική εμπειρία από τέτοιες ηλεκτρονικές κοινότητες.

7.2 Βελτιώσεις.

Στην μελλοντική δουλειά που πρέπει και μπορεί να γίνει μέσα στην εφαρμογή είναι να έχουμε την διασύνδεση της εφαρμογής για να μπορεί ο χρήστης να χρησιμοποιήσει τα social widgets για να πραγματοποιήσει κάποια εγγραφή. Ο κώδικας και τα γραφικά είναι έτοιμα όπως είδαμε στις εικόνες πιο πάνω αλλά η διασύνδεση δεν έχει γίνει γιατί υπήρξε μια καθυστέρηση όχι από την δική μας πλευρά. Επίσης μελλοντικά αυτό που μπορεί να γίνει είναι μια καλύτερη απεικόνιση των γραφικών στοιχείων μας και φυσικά η ενσωμάτωση ακόμα περισσότερων social widgets όπως έχουμε δείξει και στον ιστό ότι μπορούμε να έχουμε. Επίσης θα μπορούσαμε να δώσουμε την δυνατότητα στον χρήστη να μεταπηδά στα άρθρα που διάβασε τελευταία πιο γρήγορα απ' ό,τι να τα ψάξει όπως πριν και θα μπορούσαμε να δώσουμε και ειδοποιήσεις στον χρήστη για διάφορους λόγους όπως όταν υπάρχει ένα καινούριο άρθρο που τον ενδιαφέρει ή απάντηση σε κάποιο σχόλιο που έγραψε. Επίσης θα μπορούσαμε να προσθέσουμε έναν πίνακα χρήστη που να μπορεί να έχει μια επισκόπηση όλων αυτών που έχει κάνει με τις ρυθμίσεις του λογαριασμού που επιθυμεί, να προσθέσουμε και αυτά που αναφέραμε πριν και να προβάλλουμε και στον χρήστη στατιστικά σχετικά με την περιήγηση του στην κοινότητα και πόσο έχει συνεισφέρει σε αυτή.

ACKNOWLEDGMENT

The work was carried out in the context of the EuroCancerComs project (<http://www.eurocancercoms.eu>) funded by the EU FP 7 Science in Society Project Grant No 230548. Special thanks are due to the project's guidelines workgroup.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Akoumianakis, D., Milolidakis, G., Akrivos, A., Panteris, Z., & Ktistakis, G. (2010). Clinical practice guideline management information systems: Cancer guidelines as boundary spanning transformable objects of practice, 2nd International Conference on Intelligent Networking and Collaborative Systems (INCoS 2010), November 24-26, Thessaloniki, Greece, IEEE Computer Society, pp. 230-237.
- [2] Παρδάλης Ν. , Καραδήμος Θ. (2012) Ανάπτυξη mobile εφαρμογής διεπαφής χρήστη, για τις υπηρεσίες του Τμήματος Πληροφορικής & Τηλεπικοινωνιών, χρησιμοποιώντας την πλατφόρμα ανοιχτού κώδικα Android SDK. Πτυχιακή εργασία. Ανώτατο Εκπαιδευτικό Ίδρυμα Λάρισας, Σχολή Τεχνολογικών Εφαρμογών, Τμήμα Τεχνολογίας Πληροφορικής και Τηλεπικοινωνιών.
- [3] Δασκαλάκης Ε. (2012) Ανάλυση και χρήση εργαλείων για προγραμματισμό εφαρμογών σε Android. Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης, Σχολή Τεχνολογιών Εφαρμογών, Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων.
- [4] Android – <http://www.android.com>.
- [5] Google Code Project – <http://code.google.com>.
- [6] Android Developer – <http://developer.android.com>.
- [7] Wikipedia Org. The Free Encyclopedia – <http://www.wikipedia.org>.
- [8] JSON Org. – <http://www.json.org>.
- [9] Android Central – <http://www.androidcentral.com>.
- [10] Cnet – <http://www.cnet.com>.
- [11] JavaTpoint – [http:// www.javatpoint.com](http://www.javatpoint.com).