



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ταξιδιωτικός οδηγός της Θεσσαλονίκης για Android



Των φοιτητών

Γιακουμή Ζαχαρία

Αρ. Μητρώου: 073232

Παζαρλόγλου Ευαγγελία

Αρ. Μητρώου: 073244

Επιβλέπων καθηγητής

Κεραμόπουλος Ευκλείδης

Θεσσαλονίκη 2014

ΠΕΡΙΛΗΨΗ

Στην παρούσα πτυχιακή εργασία αναπτύχθηκε και παρουσιάζεται μια εφαρμογή με χρήση της πλατφόρμας Android SDK. Η εφαρμογή είναι ένας τουριστικός οδηγός για τη Θεσσαλονίκη με όνομα "ThessVres". Παρέχει πληροφορίες για σημεία ενδιαφέροντος και χάρτη για εύκολη πλοήγηση με βάση την τοποθεσία του χρήστη.

Για την δημιουργία της εφαρμογής χρησιμοποιήθηκε το λογισμικό Eclipse IDE το οποίο είναι ένα πρόγραμμα ανοιχτού κώδικα που σε συνεργασία με τα δωρεάν εργαλεία ανάπτυξης Android λογισμικού που προσφέρει η Google (Android Development Tools) αποτελεί ένα πολύ καλό εργαλείο ανάπτυξης εφαρμογών Android.

ABSTRACT

In this thesis developed and presented an application using the platform Android SDK. The application ThessVres is a travel guide for Thessaloniki that offers information about points of interest and street map for easy navigation based on the user's location.

To create the application we used Eclipse IDE which is an open source project in cooperation with the free Android software development tools offered by Google (Android Development Tools) is a very good development tool Android.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ	iii
ABSTRACT	iv
ΠΕΡΙΕΧΟΜΕΝΑ	v
Ευρετήριο εικόνων.....	vii
ΕΙΣΑΓΩΓΗ	10
ΚΕΦΑΛΑΙΟ 1 Android	11
ΕΙΣΑΓΩΓΗ	11
1.1: Ιστορία του Android	12
1.2 Εκδόσεις του Android.....	14
1.2.1 Android 1.0 “Apple pie”	15
1.2.2 Android 1.1 “Banana bread”	15
1.2.3 Android 1.5 “Cupcake”	16
1.2.4 Android 1.6 “Donut”	17
1.2.5 Android 2.0: 2.1 “Eclair”	18
1.2.6 Android 2.2-2.2.3 “Froyo”	19
1.2.7 Android 2.3-2.3.7 “Gingerbread”	20
1.2.8 Android 3.0-3.2.6 “Honeycomb”	21
1.2.9 Android 4.0-4.0.2 “Ice Cream Sandwich”	22
1.2.10 Android 4.1-4.3 “Jelly Bean”	23
1.2.11 Android 4.4-4.4.2 “Kit Kat”	23
1.3 Αρχιτεκτονική του Android	24
1.3.1 Πυρήνας	26
1.3.2 Βιβλιοθήκες.....	27
1.3.3 Χρόνος εκτέλεσης και εικονική μηχανή Dalvik	28
1.3.4 Πλαίσιο Εφαρμογής	29
1.3.5 Εφαρμογές και Widgets	30
ΕΠΙΛΟΓΟΣ	31
ΚΕΦΑΛΑΙΟ 2 Τεχνολογίες.....	32
ΕΙΣΑΓΩΓΗ	32
2.1 Google API	32
2.2 JavaScript Object Notation (JSON)	33
2.3 SQL Lite.....	34
ΕΠΙΛΟΓΟΣ	34

ΚΕΦΑΛΑΙΟ 3 Δομή Eclipse (ADT Plugin).....	35
ΕΙΣΑΓΩΓΗ	35
3.1 Android SDK	35
3.2 Android SDK Manager	35
3.3 Android Virtual Device	36
3.4 Android Development Tools	36
3.5 Android Debug Bridge (ADB).....	37
3.6 Εγκατάσταση και προετοιμασία του Eclipse	37
ΕΠΙΛΟΓΟΣ	43
ΚΕΦΑΛΑΙΟ 4 Δομή μίας Εφαρμογής.....	44
4.1 Δομή του πηγαίου κώδικα	44
4.2 Android Activities	45
4.3 Το αρχείο AndroidManifest.xml.....	46
4.4 Κατάλογος src.....	47
4.5 Κατάλογος assets	48
4.6 Κατάλογος res	48
4.5.1 Ο κατάλογος res/drawable	50
4.5.2 Κατάλογος res/layout.....	51
4.5.3 Κατάλογος res/values.....	54
4.5.4 Κατάλογος res/menu	57
4.7 Κατάλογος libs.....	58
4.8 Κατάλογος gen	58
4.9 Κατάλογος bin	59
ΕΠΙΛΟΓΟΣ	59
ΚΕΦΑΛΑΙΟ 5 Ανάλυση ThessVres	60
ΕΙΣΑΓΩΓΗ	60
5.1 Βιβλιοθήκες.....	60
5.1.1 Google Play services Library.....	60
5.1.2 Android Support Library	61
5.2 Ανάπτυξη ThessVres.....	63
ΕΠΙΛΟΓΟΣ	88
ΣΥΜΠΕΡΑΣΜΑΤΑ.....	89
ΒΙΒΛΙΟΓΡΑΦΙΑ	90
ΟΔΗΓΟΣ ΧΡΗΣΗΣ ThessVres	92

Ευρετήριο εικόνων

Εικόνα 1. Open Handset Alliance	11
Εικόνα 2. Οι εκδόσεις του Android	14
Εικόνα 3. Έκδοση 1.0 "Apple pie"	15
Εικόνα 4. Έκδοση 1.5 "Cupcake"	16
Εικόνα 5. Έκδοση 1.6 "Donut"	17
Εικόνα 6. Έκδοση 2.0 "Eclair"	18
Εικόνα 7. Έκδοση 2.2 "Froyo"	19
Εικόνα 8. Έκδοση 2.3 "Gingerbread"	20
Εικόνα 9. Έκδοση 3.0 "Honeycomb"	21
Εικόνα 10. Έκδοση 4.0 "Ice Cream Sandwich"	22
Εικόνα 11. Έκδοση 4.1 "Jelly Bean"	23
Εικόνα 12. Έκδοση 4.4 "Kitkat"	23
Εικόνα 13. Η Αρχιτεκτονική του Android	26
Εικόνα 14. Google Maps API στο ThessVres	32
Εικόνα 15. Το περιβάλλον του Eclipse	38
Εικόνα 16. Android SDK Manager	38
Εικόνα 17. Εγκατάσταση του ADT Plugin (1).....	39
Εικόνα 18. Εγκατάσταση του ADT Plugin (2).....	40
Εικόνα 19. Εγκατάσταση του ADT Plugin (3).....	41
Εικόνα 20. Android virtual Device Manager	42
Εικόνα 21. Εικονική συσκευή όπου τρέχει η εφαρμογή.....	42
Εικόνα 22. Η δομή μιας εφαρμογής	44
Εικόνα 23. Οι καταστάσεις ενός Activity	46
Εικόνα 24. Ο κατάλογος src	47
Εικόνα 25. Παράδειγμα μιας βάσης SQLite	48
Εικόνα 26. Ο κατάλογος res	49
Εικόνα 27. Ο κατάλογος layout.....	51
Εικόνα 28. Γραφικό περιβάλλον μίας εφαρμογής.....	54
Εικόνα 29. Ο κατάλογος values.....	55
Εικόνα 30 Παράδειγμα του αρχείου colors.xml που βρίσκεται στον κατάλογο values	56
Εικόνα 31. Το αρχείο strings.xml που βρίσκεται στον κατάλογο values	57
Εικόνα 32. Το αρχείο strings.xml που βρίσκεται στον κατάλογο values-en.....	57
Εικόνα 33. Ο κατάλογος gen	59
Εικόνα 34. Εισαγωγή πακέτου στο Workspace	62
Εικόνα 35.2. Επιλογή Βιβλιοθήκης.....	62
Εικόνα 36. Εισαγωγή της βιβλιοθήκης.....	63
Εικόνα 37. Βάση του ThessVres (Πίνακας PlaceModel).....	68
Εικόνα 38. Αρχικό Action Bar της Εφαρμογής	68
Εικόνα 39. Activity Main.....	74
Εικόνα 40. Μουσεία της Θεσσαλονίκης (ListView)	76
Εικόνα 41. Επεκταση της Κατηγορίας Βυζαντινή (ExpandableListView)	76
Εικόνα 42. Μορφή μιας γραμμής σε μια λίστα	77

Εικόνα 43. Οι πληροφορίες κάθε αντικειμένου (SingleListItem)	79
Εικόνα 44. Διαδρομή που εμφανίζεται στον χάρτη.....	80
Εικόνα 45. Το κύριο Action Bar	80
Εικόνα 46. Το Action Bar που εμφανίζεται στον χάρτη.....	81
Εικόνα 47. Το Action Bar που εμφανίζεται στον χάρτη.....	81
Εικόνα 48. Αναζήτηση Διαδρομής	81
Εικόνα 49. Απλή αναζήτηση.....	82
Εικόνα 50. Το αποτέλεσμα μιας αναζήτησης	83
Εικόνα 51. Εμφανίζει όλες τις περιοχές που υπάρχουν στο χάρτη.....	84
Εικόνα 52. Εμφανίζει ένα σημείο στον χάρτη	84
Εικόνα 53. Google console	85
Εικόνα 54. Android Build.....	86
Εικόνα 55. Δήλωση της εφαρμογής στο Google Console	86
Εικόνα 56. Google API key.....	86
Εικόνα 57. Η αρχική εικόνα (Splash Screen) χωρίς δίκτυο να είναι ανοιχτό.....	92
Εικόνα 58. Η αρχική εικόνα (Splash Screen) με το δίκτυο ανοιχτό	92
Εικόνα 59. Όταν το GPS είναι ανενεργό ζητείται η ενεργοποίηση του	93
Εικόνα 60. Η κύρια οθόνη της εφαρμογής	93
Εικόνα 61. Λίστα εγγραφών (μιας απο τις αρχικές 9).....	94
Εικόνα 62. Λίστα εγγραφών ομαδοποιημένες σε κατηγορίες	94
Εικόνα 63. Πληροφορίες μιας εγγραφής.....	95
Εικόνα 64. Μια διαδρομή στο Google Maps	95
Εικόνα 65. Σύνθετη αναζήτηση.....	96
Εικόνα 66. Οθόνη με χάρτη μετά από αναζήτηση.....	97
Εικόνα 67. Απλή αναζήτηση.....	98

ΕΙΣΑΓΩΓΗ

Τα τελευταία χρόνια η χρήση κινητών τηλεφώνων (και ιδιαίτερα συσκευών τύπου «smartphones») έχει αυξηθεί σημαντικά παγκοσμίως, αλλά και στην Ελλάδα. Φαίνεται να προκαλούν επανάσταση σε διάφορους τομείς της παγκόσμιας οικονομίας, ένας εκ των οποίων είναι και ο ηλεκτρονικός τουρισμός. Μια νέα τάση που έχει επικρατήσει στην Ευρώπη από το 2011 είναι αυτή του τουρισμού μέσω των εφαρμογών σε smartphones ή αλλιώς κινητός τουρισμός (m.Tourism). Η τεχνολογία αυτή χρησιμοποιείται κυρίως για την προώθηση εναλλακτικών μορφών τουρισμού.

Με αφορμή την παραπάνω τεχνολογία δημιουργήσαμε την εφαρμογή "ThessVres". Είναι ένας τουριστικός οδηγός για τη Θεσσαλονίκη που προσφέρει πληροφορίες για σημεία ενδιαφέροντος της πόλης. Επίσης παρέχει χάρτη για την εύκολη πλοήγηση του χρήστη όπου εμφανίζεται η τρέχουσα θέση και οι τοποθεσίες των σημείων ενδιαφέροντος.

Στο πρώτο κεφάλαιο αναπτύσσεται το λειτουργικό Android η ιστορία του, οι εκδόσεις του και η αρχιτεκτονική του.

Στο δεύτερο κεφάλαιο περιγράφουμε τις τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής.

Στο τρίτο κεφάλαιο παρουσιάζουμε το λογισμικό Eclipse και δείχνουμε πως γίνεται η εγκατάσταση του.

Στο τέταρτο κεφάλαιο αναλύουμε την δομή μιας εφαρμογής Android.

Στο πέμπτο κεφάλαιο παρουσιάζουμε την ανάπτυξη της εφαρμογής ThessVres.

Η πτυχιακή έχει στόχο τη μελέτη, τη κατανόηση και τη ανάπτυξη μιας εφαρμογής για συσκευές με λειτουργικό Android. Για την δημιουργία της χρησιμοποιήθηκε το λογισμικό Eclipse IDE και η πλατφόρμα Android SDK.

ΚΕΦΑΛΑΙΟ 1 Android

ΕΙΣΑΓΩΓΗ

Το Android είναι μια ολοκληρωμένη, ανοιχτή και ελεύθερη πλατφόρμα [13] για συσκευές όπως smartphones ή tablets που περιλαμβάνει ένα λειτουργικό σύστημα, το οποίο τρέχει τον πυρήνα του λειτουργικού Linux, το απαραίτητο ενδιάμεσο λογισμικό, βιβλιοθήκες και βασικές εφαρμογές. Έχει την δικιά του εργαλειοθήκη ανάπτυξης συστήματος λογισμικού, το Android System Development Kit παρέχει στους προγραμματιστές όλα τα εργαλεία και APIs για να αρχίσουν να αναπτύσσουν λογισμικό για την πλατφόρμα Android χρησιμοποιώντας τη γλώσσα προγραμματισμού Java.

Αρχικά αναπτύχθηκε από την Google και αργότερα συνεχίστηκε σε συνεργασία με την Open Handset Alliance η οποία είναι μια κοινοπραξία εταιριών λογισμικού, κατασκευής hardware και τηλεπικοινωνιών, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις φορητές συσκευές. Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance. Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού.



Εικόνα 1. Open Handset Alliance

Τον Ιούλιο του 2005, η Google εξαγόρασε την Android Inc, μια μικρή εταιρεία με έδρα το Palo Alto στην California των ΗΠΑ. Εκείνη την εποχή ελάχιστα ήταν γνωστά για τις λειτουργίες της Android Inc, εκτός του ότι ανέπτυσαν λειτουργικό για κινητά τηλέφωνα. Αυτή ήταν η αρχή της φημολογίας περί σχεδίων της Google για να διεισδύσει στην αγορά κινητής τηλεφωνίας.

Στην Google, η ομάδα με επικεφαλής τον Andy Rubin ανέπτυξε μια κινητή πλατφόρμα που στηρίζεται στον πυρήνα του Linux, την οποία προώθησαν με την παροχή ενός ευέλικτου, αναβαθμίσιμου συστήματος. Έχει αναφερθεί ότι η Google είχε ήδη συγκεντρώσει μια σειρά από εταιρείες hardware και software και επισήμανε στους παρόχους ότι ήταν ανοικτή σε διάφορους βαθμούς συνεργασίας εκ μέρους της. Έντυπα και ηλεκτρονικά μέσα ενημέρωσης σύντομα ανέφεραν φήμες ότι η Google ανέπτυσε μια Google-branded συσκευή. Περισσότερες φήμες ακολούθησαν, αναφέροντας ότι η Google καθόριζε τις τεχνικές προδιαγραφές και έδειχνε πρωτότυπα στους κατασκευαστές κινητών τηλεφώνων και τους φορείς δικτύων. Τελικά η Google παρουσίασε το smartphone της Nexus One που χρησιμοποιεί το open source λειτουργικό σύστημα Android. Η συσκευή κατασκευάστηκε από την HTC , και έγινε διαθέσιμη στις 5 Ιανουαρίου 2010.

1.1: Ιστορία του Android

Η εταιρεία Android, Inc. ανακαλύφθηκε στο Palo Alto, California τον Οκτώβριο από τους Andy Rubin, Rich Miner, Nick Sears και Chris White να αναπτύξουν, με τα λόγια του Rubin "smarter mobile devices that are more aware of its owner's location and preferences". Οι αρχικές προθέσεις της εταιρείας ήταν να αναπτύξουν ένα προηγμένο λειτουργικό σύστημα για digital cameras, όταν κυκλοφόρησε η αγορά των συσκευών δεν ήταν αρκετά μεγάλη, και μετέτρεψαν τις προσπάθειες τους στο να παράγουν ένα λειτουργικό σύστημα για smartphones ανταγωνιστικό με συστήματα όπως Symbian και Windows Mobile. Παρά τα παλιά επιτεύγματα της εταιρείας, η Android Inc. λειτουργούσε μυστικά, αποκαλύφθηκε μόνο ότι δούλευε πάνω σε λειτουργικό για κινητά τηλέφωνα. Τον ίδιο χρόνο ο Rubin ξέμεινε από λεφτά. Ο Steve Perlman, ένας κοντινός φίλος του Rubin, του έδωσε \$10,000 μετρητά σε φάκελο και αρνήθηκε μερίδιο από την εταιρεία.

Η Google απέκτησε την εταιρεία Android Inc. 17 Αυγούστου 2005 και τόνισε στους υπαλλήλους της Android Inc. συμπεριλαμβανομένων των Rubin, Miner και White ότι θα μείνουν στην εταιρεία μετά την συγχώνευση. Εκείνη την περίοδο δεν ήξεραν πολλά για την Android Inc. αλλά πολλοί υπέθεταν ότι η Google σχεδίαζε να εισέρθει στην αγορά του κινητού τηλεφώνου με αυτή την κίνηση. Στην Google η ομάδα καθοδηγούμενη από τον Rubin ανέπτυξε μια πλατφόρμα για κινητές συσκευές βασισμένη στον πυρήνα του Linux. Η Google έβγαλε στην αγορά την

πλατφόρμα σε κατασκευαστές και φορείς κινητών συσκευών υπόσχοντας την παροχή ενός ευέλικτου, αναβαθμίσιμου συστήματος. Η Google έφτιαξε μια σειρά hardware component και software partners (πρότυπα λογισμικού) και σηματοδότησε στους φορείς ότι ήταν ανοικτή σε διάφορους βαθμούς συνεργασίας από μέρους τους.

Η θεωρία για την πρόθεση της Google να εισέρθει στην αγορά των κινητών επικοινωνιών συνεχίστηκε να αναπτύσσεται στα μέσα του Δεκεμβρίου του 2006. Η αποκάλυψη του iPhone, ένα touchscreen κινητό της Apple, 9 Ιανουαρίου του 2007 είχε ένα αποδιοργανωτικό αποτέλεσμα στην ανάπτυξη του Android. Την ίδια περίοδο, μια πρωτότυπη συσκευή με κωδική ονομασία "Sooner" που έχει μια ομοιότητα με το BlackBerry κινητό, χωρίς touchscreen αλλά με φυσικό, QWERTY πληκτρολόγιο. Ξεκίνησαν αμέσως την δουλειά για τον ανασχεδιασμό του OS και στα πρωτότυπα του, να συνδυάσει τα χαρακτηριστικά των δικών τους σχεδίων με την συνολική εμπειρία σχεδιασμένο να συναγωνιστεί το iPhone. Τον Σεπτέμβριο του 2007, το διαδικτυακό περιοδικό *InformationWeek* κάλυψε μια αναφορά μελέτης της Evaluateserve ότι η Google έχει καταθέσει αρκετές εφαρμογές στην περιοχή της κινητής τηλεφωνίας.

Στις 5 Νοεμβρίου 2007, η Open Handset Alliance, μια κοινοπραξία των εταιρειών τεχνολογίας, συμπεριλαμβανομένης των Google, κατασκευαστές κινητών όπως HTC, Sony και Samsung, ασύρματων φορείς όπως Sprint Nextel και T-Mobile, και κατασκευαστές chipset όπως Qualcomm και Texas Instruments, δημιουργήθηκε με στόχο την ανάπτυξη ανοικτών προτύπων για τις κινητές συσκευές. Εκείνη την ημέρα η Android αποκάλυψε το πρώτο προϊόν μια πλατφόρμα κινητών συσκευών βασισμένη στο πυρήνα Linux έκδοσης 2.6. Το πρώτο εμπορικά διαθέσιμο smartphone που έτρεχε Android ήταν το HTC Dream, που κυκλοφόρησε 22 Οκτωβρίου 2008.

Το 2010 η Google λάνσαρε την σειρά συσκευών Nexus: μια σειρά από smartphones και tablets που τρέχουν το λειτουργικό σύστημα Android, και φτιαγμένα από συνεταιρο κατασκευαστή. Η HTC συνεργάστηκε με την Google για να κυκλοφορήσει το πρώτο Nexus smartphone, το Nexus One. Η σειρά είχε έκτοτε αναβαθμιστεί με νεότερες συσκευές, όπως το τηλέφωνο Nexus 5 και Nexus 7 tablet, φτιαγμένες από την LG και την Asus αντίστοιχα. Η Google κυκλοφόρησε τα τηλέφωνα και τα tablet Nexus να ενεργούν σαν την «ναυαρχίδα» των συσκευών Android, επιδεικνύοντας το τελευταίο λειτουργικό και χαρακτηριστικά hardware του Android. Στις 13 Μαρτίου 2013 ο Larry Page ανακοίνωσε σε ένα blog ότι ο Andy Rubin μετακινήθηκε από το τμήμα Android για να αναλάβει νέα projects στην Google. Τον αντικατέστησε ο Sundar Pichai, που επίσης συνέχισε το ρόλο του ως αρχηγός του τμήματος Chrome της Google, που ανέπτυξε το λειτουργικό σύστημα Chrome.[4]

1.2 Εκδόσεις του Android

Από το 2008 που βγήκε το πρώτο Android smartphone μέχρι σήμερα έχουν βγει αρκετες εκδόσεις οι οποίες έχουν σταδιακά βελτιώσει το λειτουργικό σύστημα, προσθέτοντας καινούργια χαρακτηριστικά και επιδιορθώνοντας λάθη των προηγούμενων εκδόσεων. Η ονομασία των εκδόσεων αυτών ακολουθεί αλφαβητική σειρά πάντα παίρνουν το όνομά τους από γλυκά και επιδόρπια, πχ Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, κλπ.



Εικόνα 2. Οι εκδόσεις του Android

Η πρώτη έκδοση του Android SDK τον Νοέμβριο του 2007, χαρακτηρίστηκε από τους κατασκευαστές του σαν μια πρώτη ματιά στο SDK του Android, κάτι το οποίο πολλοί παράβλεψαν και βιάστηκαν να κατακρίνουν το Android σαν ένα προβληματικό σύστημα.

Στην ουσία όμως το Android δεν παρουσίαζε προβλήματα τα οποία δεν παρουσιάζει οποιοδήποτε σύστημα σε τέτοια πρώιμη φάση. Έτσι το Σεπτέμβριο του 2008, η T-Mobile ανακοινώνει την διαθεσιμότητα του T-Mobile G1, του πρώτου έξυπνου τηλεφώνου (smartphone), βασισμένο στην πλατφόρμα του Android. Λίγες μέρες αργότερα (Οκτώβριο 2008), η Google ανακοινώνει την απελευθέρωση του SDK Release Candidate 1.0. Ακολούθησε τον Φεβρουάριο του 2009 η έκδοση 1.1 σαν μια ανανεωμένη έκδοση του 1.0. Μέχρι τότε το Android δεν υποστήριζε ακόμη την χρήση κουμπιών αφής, παρά μόνο την χρήση των κλασικών 'σκληρών' κουμπιών της συσκευής.

1.2.1 Android 1.0 “Apple pie”



Εικόνα 3. Έκδοση 1.0 "Apple pie"

Η έκδοση Android 1.0 είναι η πρώτη εμπορική έκδοση του λειτουργικού Android και κυκλοφόρησε 23 Σεπτεμβρίου 2008. Η πρώτη εμπορική συσκευή Android ήταν HTC Dream.

1.2.2 Android 1.1 “Banana bread”

Στις 9 Φεβρουαρίου 2009, κυκλοφόρησε η αναβάθμιση Android 1.1 αρχικά μόνο για το HTC Dream. Android 1.1 ήταν γνωστό ως "Petit Four", αν και αυτό το όνομα δεν χρησιμοποιείται επίσημα. Η αναβάθμιση επίλυσε σφάλματα, άλλαξε το Android API και πρόσθεσε μια σειρά από καινούργια χαρακτηριστικά.[6]

1.2.3 Android 1.5 “Cupcake”



Εικόνα 4. Έκδοση 1.5 "Cupcake"

Στις 30 Απριλίου 2009 κυκλοφόρησε η αναβάθμιση Android 1.5 βασισμένη στον πυρήνα Linux 2.6.27. Αυτή ήταν η πρώτη κυκλοφορία που χρησιμοποιεί όνομα γλυκού ("Cupcake"), ένα θέμα που θα χρησιμοποιείται και στις μετέπειτα εκδόσεις (κυκλοφορίες). Η αναβάθμιση περιλάμβανε πολλά καινούργια χαρακτηριστικά και τροποποιήσεις στην διεπαφή του χρήστη (User Interface). Υποστηρίζει νέες λειτουργίες για την κάμερα της συσκευής, όπως η καταγραφή και παρακολούθηση βίντεο από την λειτουργία της κάμερας και η άμεση μεταφόρτωση του βίντεο αλλά και των φωτογραφιών στο Youtube και το Picasa αντίστοιχα απευθείας από το τηλέφωνο. Έχει νέο έξυπνο πληκτρολόγιο με πρόβλεψη κειμένου. Υποστηρίζει πρότυπο Bluetooth A2DP και AVRCP ενώ έχει και την ικανότητα να συνδέεται αυτόματα σε μικροσυσκευές Bluetooth από μια συγκεκριμένη απόσταση. Ακόμα στην έκδοση αυτή έχει νέο γραφικό περιβάλλον με κινούμενες μεταβάσεις οθόνης.

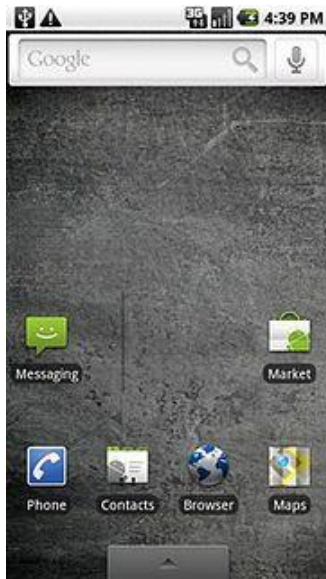
1.2.4 Android 1.6 "Donut"



Εικόνα 5. Έκδοση 1.6 "Donut"

Στις 15 Σεπτεμβρίου 2009, Κυκλοφόρησε το Android 1.6 SDK με το όνομα "Donut", βασισμένο στο πυρήνα Linux 2.6.29. Συμπεριελάμβανε νέες βελτιώσεις όπως αυτόματη περιστροφή οθόνης και εφέ κίνησης κατά την περιήγηση στις διαφορετικές οθόνες. Έχει ταχύτερη απόκριση σε σχέση με την προηγούμενη έκδοση. Υποστηρίζεται πλέον η επιλογή πολλαπλών αρχείων ταυτόχρονα, έχει ανανεωμένο γκάλερι και φωτογραφική μηχανή, καθώς και βελτιωμένο Android Market. Έχει ανανεωμένη φωνητική αναζήτηση, με ταχύτερη απόκριση και βαθύτερη ολοκλήρωση με εγγενείς (native) εφαρμογές, συμπεριλαμβανομένης της δυνατότητας κλήσης επαφών. Δυνατότητα αναζήτησης σελιδοδεικτών, ιστορικού, επαφών αλλά και στο διαδίκτυο από την αρχική οθόνη. Υποστήριξη για ανάλυση οθονών WVGA. Ανανεωμένη υποστήριξη τεχνολογιών για CDMA/EVDO, 802.1x, VPNs και με μηχανή μετατροπής κειμένου σε ομιλία (text-to-speech).

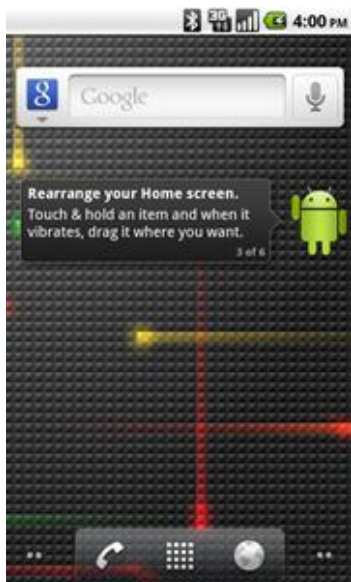
1.2.5 Android 2.0: 2.1 "Eclair"



Εικόνα 6. Έκδοση 2.0 "Eclair"

Η έκδοση Android 1.6 κυκλοφόρησε με το όνομα "Eclair" μόλις ένα μήνα μετά το Donut (26 Οκτώβριου 2009). Σε αυτή την έκδοση υπάρχει ακόμα ταχύτερη απόκριση του υλικού σε σχέση με τις δυο προηγούμενες και πλέον υποστηρίζονται περισσότερες οθόνες και αναλύσεις. Υπάρχει νέος browser ο οποίος υποστηρίζει το πρότυπο HTML5, νέο User Interface, και βελτιωμένοι χάρτες Google (Google Maps 3.1.2). Έχει ενσωματωθεί η υποστήριξη φλας για την κάμερα η οποία έχει πλέον και ψηφιακό zoom. Επίσης έχει βελτιωθεί η κλάση MotionEvent ώστε να υπάρχει η δυνατότητα για γεγονότα πολλαπλής αφής (multitouch events). Υποστηρίζεται Bluetooth 2.1 και έχει βελτιωθεί και το πληκτρολόγιο.

1.2.6 Android 2.2-2.2.3 “Froyo”



Εικόνα 7. Έκδοση 2.2 "Froyo"

Στις 20 Μαΐου 2010 κυκλοφόρησε η έκδοση 2.2 με όνομα "Froyo" και είναι συντόμηση της φράσης "Frozen Yogurt" (παγωμένο γιαούρτι). Είναι η πρώτη έκδοση του Android που υποστήριζε Adobe Flash. Υπάρχουν βελτιστοποιήσεις στην ταχύτητα γενικά του λειτουργικού συστήματος, στην μνήμη και στην απόδοση. Έχει ενσωματωθεί ο μηχανισμός JavaScript του Chrome V8 στον browser και ενώ υποστηρίζεται καλύτερα πλέον το Microsoft Exchange. Έχει γίνει ανανέωση του Android Market. Ο χρήστης μπορεί πλέον να ελέγχει αν θα γίνεται ή όχι κίνηση πακέτων δεδομένων από το δίκτυο κινητής τηλεφωνίας. Υπάρχει η δυνατότητα εγκατάστασης εφαρμογών στην κάρτα μνήμης και η μεταφορά τους εκεί από τη μνήμη του τηλεφώνου. Επίσης το τηλέφωνο πλέον μπορεί να μετατραπεί σε WiFi hotspot.

1.2.7 Android 2.3-2.3.7 "Gingerbread"



Εικόνα 8. Έκδοση 2.3 "Gingerbread"

Κυκλοφόρησε 6 Δεκεμβρίου του 2010 η έκδοση 2.3 με όνομα "Gingerbread". Ήταν πολύ πιο γρήγορο και εύχρηστο από τις προηγούμενες εκδόσεις και έδινε στους δημιουργούς εφαρμογών μεγαλύτερες δυνατότητες. Στην έκδοση αυτή υπάρχουν αλλαγές στο User Interface το οποίο έχει γίνει πιο απλό και ταχύ, ενώ υποστηρίζονται πλέον οθόνες μεγάλων μεγεθών και αναλύσεων. Υπάρχει πλέον το πρωτόκολλο SIP για κλήσεις μέσω VoIP, υποστηρίζεται ο τύπος βίντεο WebM/VP8 και ο κωδικοποιητής AAC, έχει βελτιωθεί ο ήχος καθώς και οι λειτουργίες απεικόνισης για την ανάπτυξη παιχνιδιών. Υπάρχει η δυνατότητα για Copy-Paste σε όλο το σύστημα και όχι μόνο στην ίδια εφαρμογή. Υποστηρίζεται το NFC (Near Field Communication) και η ύπαρξη πολλαπλών καμερών. Επίσης, έχει βελτιωθεί η ενεργειακή υποστήριξη και έχει γίνει μετάβαση από το σύστημα αρχείων YAFFS στο ext4 στις νέες συσκευές.

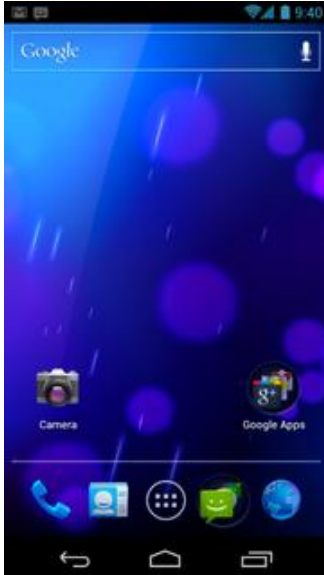
1.2.8 Android 3.0-3.2.6 “Honeycomb”



Εικόνα 9. Έκδοση 3.0 "Honeycomb"

Στις 22 Φεβρουαρίου του 2011 κυκλοφόρησε η έκδοση Android 3.0 με το όνομα "Honeycomb" και ήταν διαθέσιμη μόνο για tablets. Η πρώτη συσκευή που έτρεχε σε Honeycomb ήταν το tablet Motorola Xoom (το οποίο κυκλοφόρησε 24 Φεβρουαρίου 2011). Οι αλλαγές που έγιναν στην έκδοση αυτή έχουν να κάνουν κυρίως με τη βελτίωση της υποστήριξης των tablets. Υπάρχει ένα νέο, εντελώς διαφορετικό, User Interface και υποστηρίζονται διπύρρηνοι και τετραπύρρηνοι επεξεργαστές. Επίσης, έχει απλοποιηθεί το multitasking έτσι ώστε ο χρήστης να μπορεί με τη χρήση ενός πλήκτρου (recent apps) να περνάει από μια εφαρμογή σε άλλη. Υπάρχει η δυνατότητα για Video Chat μέσω της εφαρμογής Google Talk καθώς η ανάγνωση βιβλίων μέσω του Google eBooks. Επιπλέον, μπορούν να κρυπτογραφηθούν όλα τα δεδομένα χρήστη.

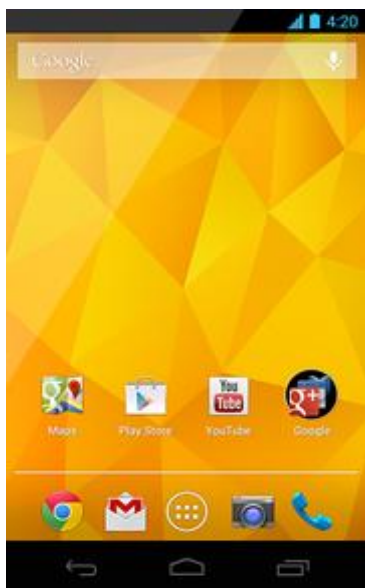
1.2.9 Android 4.0-4.0.2 “Ice Cream Sandwich”



Εικόνα 10. Έκδοση 4.0 "Ice Cream Sandwich"

Στις 19 Οκτωβρίου του 2011 κυκλοφόρησε η έκδοση 4.0 με όνομα "Ice Cream Sandwich" και έφερε πάρα πολλές αλλαγές στο λειτουργικό σύστημα. Μερικές από αυτές είναι η δυνατότητα χρήσης “μαλακών” κουμπιών δηλαδή κουμπιών πάνω στην οθόνη (πίσω, αρχική, κλπ) ενώ μέχρι τότε όλα τα κινητά είχαν εξωτερικά κουμπιά. Άλλες δυνατότητες ήταν το Face Unlock, καλύτερη χρήση των φωνητικών εντολών, αναδιαμόρφωση του περιβάλλοντος χρήσης, βελτίωση της ταχύτητας απόκρισης κλπ. Ο browser μπορεί να ανοίξει ταυτόχρονα μέχρι και 16 καρτέλες. Υπάρχει η δυνατότητα ο χρήστης να τερματίσει εφαρμογές οι οποίες τρέχουν στο background, ενώ μπορεί να θέσει και όρια στην κίνηση πακέτων δεδομένων. Η εφαρμογή Android Beam αξιοποιεί πλέον το NFC αφού επιτρέπει την αποστολή δεδομένων από τη συσκευή σε όσες βρίσκονται εντός μιας μικρής ακτίνας εμβέλειας. Ακόμα με την ύπαρξη του Wi-Fi Direct συσκευές μπορούν να συνδεθούν μεταξύ τους ασύρματα χωρίς την μεσολάβηση κάποιου access point. Επίσης υποστηρίζεται η εγγραφή βίντεο σε 1080p. Τέλος, σε αυτή την έκδοση το market του android αλλάζει σε "play store".

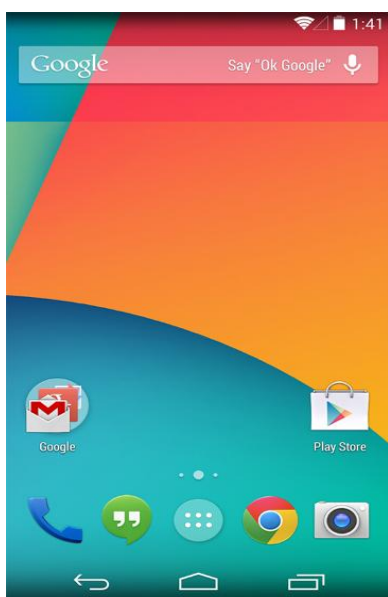
1.2.10 Android 4.1-4.3 “Jelly Bean”



Εικόνα 11. Έκδοση 4.1 "Jelly Bean"

Στις 27 Ιουνίου του 2012 κυκλοφόρησε η έκδοση 4.1 με όνομα “Jelly Bean”. Το περιβάλλον χρήσης και η απόκρισή του είναι πιο γρήγορα και καλοφτιαγμένα ενώ περιλαμβάνει πάρα πολλές μικρές βελτιώσεις σε όλο το σύστημα, όπως για παράδειγμα στην χρήση φωνής (υπαγόρευση κειμένου) και στην κάμερα. Η πρώτη συσκευή που είχε το Jelly Bean ήταν το tablet Google Nexus 7 ενώ η έκδοση Android 4.2 πρωτοεμφανίστηκε στα Nexus 4 και Nexus 10.

1.2.11 Android 4.4-4.4.2 “Kit Kat”



Εικόνα 12. Έκδοση 4.4 "Kitkat"

Στις 31 Οκτωβρίου του 2013. Η νέα έκδοση του Android 4.4 έχει τόσες σημαντικές και πολλές βελτιώσεις που αξίζει να σταθούμε στα βασικά. Το User Interface και το User experience έχουν αλλάξει και έχουν βελτιωθεί στα σημεία. Αυτό σημαίνει ότι τα γραφικά είναι λίγο διαφορετικά και η εμπειρία χρήσης του Android KitKat καλύτερη από ποτέ. Οι διαφορές θα γίνονται αμέσως ορατές όταν οι κάτοχοι Nexus 4 κάνουν αναβάθμιση στο Android 4.4 γιατί για όλους τους άλλους κατόχους Samsung smartphones και LG smartphones με custom skins, τα πράγματα είναι διαφορετικά. Σημαντική αλλαγή που φαίνεται από την πρώτη ματιά είναι η απώλεια των notification bar και control bar έχουν εξαφανιστεί και έχουν γίνει διάφανες. Η ελάχιστη τιμή της απαιτούμενης RAM αυξάνεται από τα 340mb στα 512mb και οι συσκευές που έχουν RAM κάτω από 512 θεωρούνται συσκευές "low RAM". Όμως ο Πρόεδρος του Chrome και Android, Sundar Pichai, δήλωσε ότι παρά τις νέες δυνατότητες του Android 4.4 η νέα έκδοση μπορεί να τρέξει και σε smartphones παλαιότερης τεχνολογίας. Το Google φρόντισε το Android KitKat να καταναλώνει λιγότερη μνήμη από ποτέ. Χαρακτηριστικό παράδειγμα ο Chrome που στο Android 4.4 καταναλώνει 16% λιγότερη RAM σε σύγκριση με το Android 4.3.

Πρέπει να σημειωθεί ότι οι εταιρείες κατασκευής smartphones πολλές φορές προσαρμόζουν το Android βάζοντας νέες δυνατότητες και χαρακτηριστικά. Άλλωστε το Android είναι ανοιχτό λειτουργικό που σημαίνει ότι μπορεί ο καθένας να επεμβεί και να το επεξεργαστεί όπως εκείνος νομίζει.

1.3 Αρχιτεκτονική του Android

Το Android δεν είναι μόνο ένα λειτουργικό σύστημα. Είναι μια στοίβα λογισμικού η οποία αποτελείται από το λειτουργικό σύστημα, τις υπηρεσίες διασύνδεσης με τις εφαρμογές (middleware) και τέλος από τις κύριες (core)εφαρμογές, μεταξύ αυτών, ενός email client, μιας εφαρμογής διαχείρισης SMS, ενός ημερολογίου, ενός browser, εφαρμογή διαχείρισης επαφών, και άλλες οι οποίες έρχονται δεμένες με την υπόλοιπη στοίβα λογισμικού του Android.

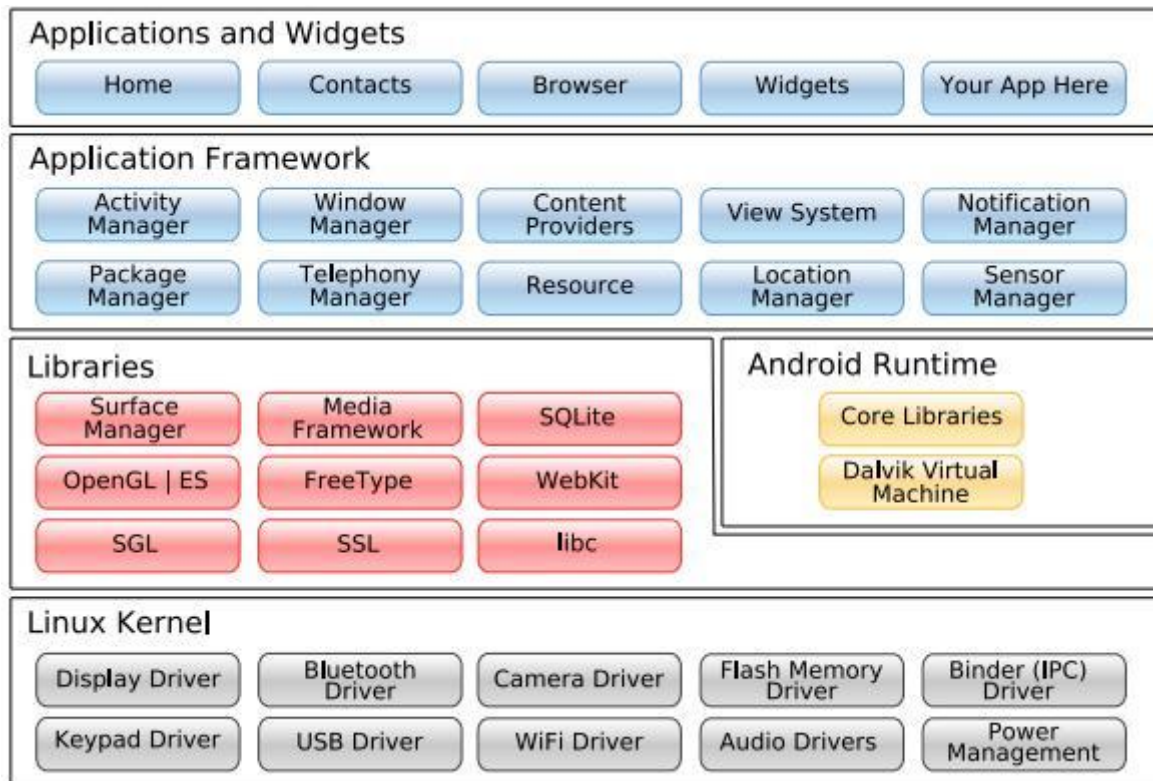
Η αρχιτεκτονική του Android ενθαρρύνει την έννοια επαναχρησιμοποίησης κώδικα, που επιτρέπει να δημοσιευτούν και να μοιραστούν δραστηριότητες, υπηρεσίες και δεδομένα με άλλες εφαρμογές με πρόσβαση που ελέγχεται από τους περιορισμούς ασφάλειας που έχουν τεθεί. Έτσι γίνεται δυνατή η επέκταση και βελτιστοποίηση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

Οι ακόλουθες υπηρεσίες εφαρμογών είναι οι ακρογωνιαίοι λίθοι της αρχιτεκτονικής όλων των εφαρμογών Android, παρέχοντας το πλαίσιο που θα χρησιμοποιηθεί για τη δημιουργία του λογισμικού:

- Activity Manager: Ελέγχει τον κύκλο ζωής των δραστηριοτήτων, συμπεριλαμβανομένης της διαχείρισης του σωρού δραστηριοτήτων.
- Views: Χρησιμοποιούνται για να κατασκευάσουν διεπαφές χρήστη(user interface) για τις δραστηριότητες(activities).
- Notification Manager: Παρέχει ένα συνεπή και αποτελεσματικό μηχανισμό για να προειδοποιεί τον χρήστη
- Content Providers: Επιτρέπουν στις εφαρμογές να μοιραστούν δεδομένα μεταξύ τους.
- Resource Manager: Υποστηρίζει πόρους εκτός κώδικα όπως συμβολοσειρές και γραφικά.

Στην στοίβα του Android παρατηρούμε 4 επίπεδα. Ακολούθως θα περιγράψουμε συνοπτικά τα βασικά αυτά επίπεδα χωρίς να μπούμε σε λεπτομέρειες για όλα τα περιεχόμενα του κάθε επιπέδου.

Κάθε επίπεδο στην αρχιτεκτονική αυτή, χρησιμοποιεί τις υπηρεσίες που του προσφέρονται από τα πιο πάνω επίπεδα. Σας παρουσιάζουμε αυτά τα επίπεδα ξεκινώντας από το πιο χαμηλό.



Εικόνα 13. Η Αρχιτεκτονική του Android

Η αρχιτεκτονική του λειτουργικού συστήματος αποτελείται από 4 βασικά επίπεδα:

- Τον πυρήνα Linux (Linux Kernel)
- Τις εγγενείς και τις προηγμένες βιβλιοθήκες (Libraries)
- Τον χρόνο εκτέλεσης (Android Runtime) και την εικονική μηχανή Dalvik (Dalvik VM)
- Το πλαίσιο εφαρμογής (Application Framework)
- Εφαρμογές και Widgets

1.3.1 Πυρήνας

Η βάση της στοίβας λογισμικού του Android είναι ο πυρήνας Linux. Ο πυρήνας Linux είναι δοκιμασμένος, σταθερός και πετυχημένος και μπορεί να βρεθεί παντού, από ρολόγια χειρός μέχρι υπερυπολογιστές. Ο τροποποιημένος πυρήνας του συστήματος βασίζεται στην έκδοση 2.6 (και στην έκδοση 3.0.1 για το Android 4.0) του Linux Kernel, η οποία υποστηρίζει όλες τις κύριες λειτουργίες του λειτουργικού συστήματος. Οι λειτουργίες αυτές αφορούν διαχείριση μνήμης, διαχείριση διεργασιών, λειτουργίες δικτύου, ασφάλεια του λειτουργικού, και ένα

σύνολο οδηγών υλικού (hardware drivers). Οι οδηγοί αυτοί είναι υπεύθυνοι για την επικοινωνία του software με το hardware της συσκευής. Ενδεικτικά ο πυρήνας του Android περιέχει:

- Οδηγό προβολής οθόνης
- Οδηγό Wifi και BlueTooth
- Οδηγό κάμερας
- κλπ

Ο πυρήνας του Android μπορεί να βασίζεται στον πυρήνα του Linux, αλλά διαφέρει αρκετά από αυτόν. Ο λόγος είναι οι αλλαγές στην αρχιτεκτονική που έχει κάνει η Google για να είναι ελαφρύτερος και βελτιστοποιημένος για χρήση σε κινητές συσκευές. Αυτό σημαίνει ότι παρότι το Android είναι κατά βάση Linux, επί της ουσίας είναι αρκετά δύσκολο να τρέξουν εφαρμογές ή να χρησιμοποιηθούν βιβλιοθήκες από τη μία πλατφόρμα στην άλλη. Το Linux παρέχει στο Android το αφαιρετικό επίπεδο υλικού, επιτρέποντάς του να μπορεί να χρησιμοποιηθεί σε μεγάλη ποικιλία πλατφόρμων στο μέλλον. Ο Linus Torvalds έχει αναφέρει ότι τελικά στο μέλλον το Android και το Linux θα μοιράζονται έναν κοινό πυρήνα.

1.3.2 Βιβλιοθήκες

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

Το σύνολο σχεδόν των βιβλιοθηκών είναι γραμμένο σε C και C++ [11], οι οποίες έχουν μεταγλωττιστεί για τη χρήση τους από το λειτουργικό. Μερικές από τις κύριες βιβλιοθήκες του Android είναι:

- Surface Manager: Διαχειρίζεται την πρόσβαση στο υποσύστημα προβολής, και συνθέτει δισδιάστατα και τρισδιάστατα επίπεδα γραφικών τα οποία προέρχονται από πολλαπλές εφαρμογές.
- Open GL/ ES: Είναι βιβλιοθήκη γραφικών που χρησιμοποιείται για να υποστηρίξει τρισδιάστατα γραφικά βασισμένα στο Open GL ES 1.0 API.

- SGL: Είναι βιβλιοθήκη γραφικών που χρησιμοποιείται για να υποστηρίξει δισδιάστατα γραφικά.
- Media Framework: Προμηθεύτηκε από την PacketVideo, μέλος της Open Handset Alliance.[8] Περιέχει όλα τα codecs για να υποστηρίξει αναπαραγωγή και εγγραφή πολλών δημοφιλών μέσων ήχου και εικόνας, όπως: MPEG4, H.264, MP3, AAC, AMR, JPG, PNG κ.α.
- FreeType: Παρέχει ευκρίνεια γραφικών στις γραμματοσειρές των εφαρμογών του συστήματος.
- SSL: Υποστήριξη για χρήση των Secure Sockets Layer του κρυπτογραφικού πρωτοκόλλου για ασφαλείς επικοινωνίες μέσω διαδικτύου.
- SQLite: Ελαφριά μηχανή σχεσιακών βάσεων δεδομένων που χρησιμοποιείται για να αποθηκεύσει δεδομένα εφαρμογών
- WebKit: Μια opensource μηχανή browser που υποστηρίζει την πλοήγηση στο διαδίκτυο. Χρησιμοποιείται και από τον ενσωματωμένο browser του Android αλλά και από τις WebViews που ενσωματώνονται στις εφαρμογές.
- Libc: μια ενσωμάτωση της standard βιβλιοθήκης συστήματος της C τροποποιημένη για κινητές συσκευές βασισμένες στο Linux.

1.3.3 Χρόνος εκτέλεσης και εικονική μηχανή Dalvik

Ο χρόνος εκτέλεσης των εφαρμογών του Android, βρίσκεται στο ίδιο επίπεδο με τις κύριες βιβλιοθήκες και την μηχανή Dalvik. Σχεδόν το σύνολο των APIs του Android βασίζονται στη γλώσσα προγραμματισμού Java. Στην Java ως γνωστόν υπάρχει η λεγόμενη Java Virtual Machine στην οποία εκτελείται ο κώδικας bytecode των εφαρμογών. Στο Android υπάρχει μια υλοποίηση αυτής της εικονικής μηχανής η οποία ονομάζεται Dalvik. Η Dalvik είναι μια βελτιστοποιημένη υλοποίηση μιας εικονικής μηχανής Java για φορητές συσκευές από την Google. Μέσω της εικονικής μηχανής Dalvik τρέχουν οι εφαρμογές του Android καθε μία με τη δικιά της διεργασία και για αυτό το λόγο καμία εφαρμογή δεν έχει επαφή με άλλη, ενώ εκτελούνται ταυτόχρονα. Η Dalvik τρέχει .dex αρχεία, τα οποία είναι bytecodes που προέρχονται από πηγαίο κώδικα και βιβλιοθήκες (αρχεία .class και .jar) της Java. Τα αρχεία dex ουσιαστικά αποτελούν συμπιεσμένα δεδομένα για εξοικονόμηση χώρου κατά την εκτέλεση, γεγονός σημαντικό για συσκευές με περιορισμένη μνήμη και μπαταρία. Το Android είναι από τη φύση του multitasking λειτουργικό σύστημα και για αυτό επιτρέπει στις εφαρμογές του να τρέχουν σε πολλά νήματα ταυτόχρονα και να απασχολούν πολλές διαδικασίες εάν αυτό είναι

αναγκαίο. Για να γίνει αυτό εφικτό η μηχανή Dalvik είναι σχεδιασμένη για να έχει ελάχιστο αντίκτυπο στη χρήση της μνήμης. Χάρη στον λιτό της σχεδιασμό, το σύστημα είναι σε θέση να τρέχει πολλές εικονικές μηχανές ταυτόχρονα.

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

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

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

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

1.3.4 Πλαίσιο Εφαρμογής

Το Android παρέχει στους developers μια ανοιχτού κώδικα πλατφόρμα ανάπτυξης και τη δυνατότητα να αναπτύξουν με αυτή ιδιαίτερα καινοτόμες και πλούσιες σε υλικό, εφαρμογές. Οι developers έχουν στη διάθεση τους τη δυνατότητα ελέγχου του υλικού της συσκευής και μέσω αυτής μπορούν να αποκτήσουν πρόσβαση σε υπηρεσίες εντοπισμού, εκτέλεση διεργασιών παρασκηνίου, και πάρα πολλές ακόμη δυνατότητες οι οποίες βασίζονται στα APIs που είναι διαθέσιμα.

Στο επόμενο επίπεδο της αρχιτεκτονικής του Android λοιπόν, συναντάμε το πλαίσιο των εφαρμογών. Οι developers έχουν πρόσβαση σε όλα τα APIs μεταξύ αυτών και στα κύρια APIs που χρησιμοποιούν οι ενσωματωμένες εφαρμογές. Η

δομή των εφαρμογών είναι τέτοια που ευνοείται η επαναχρησιμοποίηση δομικών συστατικών, και επίσης επιτρέπεται η χρήση των δυνατοτήτων της μίας εφαρμογής από άλλες εφαρμογές, βέβαια κάτω από τις προδιαγραφές ασφάλειας του Android. Αυτό το πλαίσιο είναι προ-εγκατεστημένο στο Android, αλλά είναι επεκτάσιμο, αφού ο κάθε κατασκευαστής μπορεί να το συμπληρώσει με δικά του κομμάτια. Τα σημαντικότερα δομικά στοιχεία του πλαισίου εφαρμογών είναι:

- **View System:** Αποτελεί ένα εκτενές σύνολο από αντικείμενα GUI τα οποία μπορούν να χρησιμοποιηθούν κατά το σχεδιασμό μιας εφαρμογής. Παραδείγματα προβολών είναι οι λίστες (listView), το πλέγμα (GridView), πεδία εισαγωγής κειμένου, κουμπιά, κλπ
- **Content Providers:** Δίνει τη δυνατότητα στις εφαρμογές να μοιράζονται ή να ανταλλάσσουν δεδομένα μιας συγκεκριμένης μορφής η οποία ορίζεται από τον πάροχο. Παραδείγματα δεδομένων, είναι οι επαφές χρήστη και οι βάσεις δεδομένων των εφαρμογών.
- **Resource Manager:** Παρέχει πρόσβαση σε υλικό το οποίο δεν είναι σε μορφή κώδικα όπως πχ, εικόνες, αρχεία xml, πίνακες χαρακτήρων, κλπ. Για παράδειγμα μπορεί να είναι κωδικοί χρωμάτων, αλφαριθμητικοί χαρακτήρες ή ακόμα και έτοιμα σχεδιαγράμματα οθονών φτιαγμένα σε XML, τα οποία μπορεί το πρόγραμμα να καλεί.
- **Notification Manager:** Δίνει στις εφαρμογές πρόσβαση στις υπηρεσίες ειδοποιήσεων χρήστη. Τέτοιες είναι οι ειδοποιήσεις στη notification bar, τα Toast μηνύματα στο κάτω μέρος της οθόνης, η δόνηση του κινητού και η ενεργοποίηση της οθόνης, κλπ
- **Activity Manager:** Διαχειρίζεται τον κύκλο ζωής των δραστηριοτήτων και παρέχει δυνατότητα πλοήγησης από δραστηριότητα σε δραστηριότητα κρατώντας αποθηκευμένη στη μνήμη τη σειρά εκτέλεσης αυτών.
- **Location Manager:** Χρησιμοποιείται για να μπορεί να ξέρει η συσκευή που βρίσκεται ανά πάσα στιγμή.

1.3.5 Εφαρμογές και Widgets

Στο υψηλότερο επίπεδο της στοίβας Android, βρίσκονται οι εφαρμογές και τα widgets δηλαδή τα προγράμματα είναι που βλέπουν οι χρήστες χωρίς να γνωρίζουν την υπόλοιπη στοίβα του λειτουργικού. Μερικές εφαρμογές έρχονται προεγκατεστημένες με την Android συσκευή (όπως επαφές, μουσική κ.α.), ενώ τις περισσότερες τις εγκαθιστά ο χρήστης ανάλογα με τις ανάγκες του. Οι εφαρμογές είναι προγράμματα που καταλαμβάνουν ολόκληρη την οθόνη και αλληλεπιδρούν

με το χρήστη. Από την άλλη τα widget λειτουργούν σε μικρά τετράγωνα μέσα στην αρχική οθόνη: εφαρμογή.

Για τη διευκόλυνση της διαδικασίας εύρεσης και εγκατάστασης εφαρμογών, σε κάθε Android συσκευή υπάρχει διαθέσιμη και προεγκαταστημένη η εφαρμογή Google Play. Αυτή η εφαρμογή επιτρέπει στους χρήστες να κατεβάζουν εφαρμογές που έχουν δημιουργηθεί από τρίτους προγραμματιστές. Το Σεπτέμβριο του 2011 υπήρχαν περισσότερες από 675.000 εφαρμογές διαθέσιμες και ο αριθμός των εγκατεστημένων ξεπέρασε τα 25 δισεκατομμύρια.

ΕΠΙΛΟΓΟΣ

Σε αυτό το κεφάλαιο, κάναμε μια εισαγωγή στο λειτουργικό σύστημα Android της Google αναφέροντας ιστορικά συμβάντα που αξίζουν να σημειωθούν και μελετήσαμε τις εκδόσεις και την αρχιτεκτονική του.

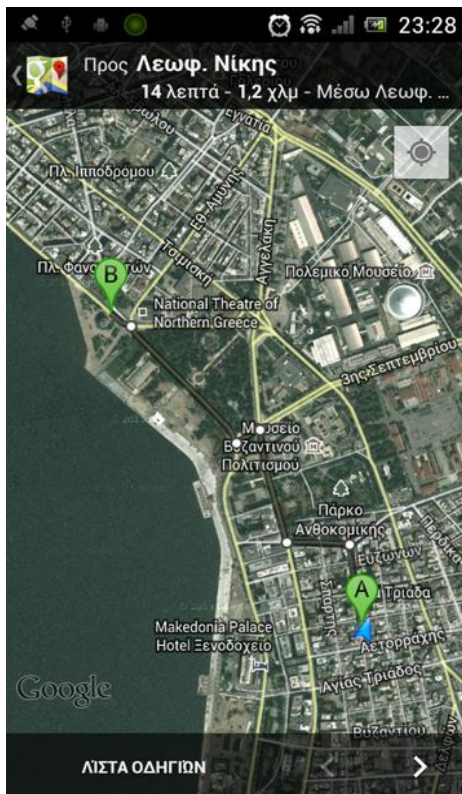
ΚΕΦΑΛΑΙΟ 2 Τεχνολογίες

ΕΙΣΑΓΩΓΗ

Σε αυτό το κεφάλαιο θα αναπτύξουμε κάποιες τεχνολογίες του Android που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής μας.

2.1 Google API

Η Τεχνολογία API είναι μια συντομογραφία του **Application Program Interface** είναι μια συλλογή από μεθόδους, πρωτόκολλα και εργαλεία για την δημιουργία εφαρμογών λογισμικού. Το API καθορίζει το πως τα components λογισμικού θα πρέπει να αλληλεπιδρούν μεταξύ τους και χρησιμοποιούνται για να προγραμματίζονται τα graphical user interface (GUI) components. Δίνει την δυνατότητα στους προγραμματιστές να χρησιμοποιούν προκαθορισμένες λειτουργίες που αλληλεπιδρούν με το λειτουργικό σύστημα.



Εικόνα 14. Google Maps API στο ThessVres

Στην εργασία μας χρησιμοποιούμε API της google και συγκεκριμένα το google maps api v2 για την εμφάνιση του χαρτη, που παρέχει η google, για εύκολη

πλοήγηση του χρήστη στη Θεσσαλονίκη. Το Google Maps είναι υπηρεσία χαρτογράφησης στο Διαδίκτυο. Η εφαρμογή και η τεχνολογία της υπηρεσίας παρέχεται από την Google και υποστηρίζει πολλές υπηρεσίες που βασίζονται σε χάρτες, συμπεριλαμβανομένου της ιστοσελίδας "Χάρτες Google". Προσφέρει χάρτες δρόμων και σχεδιαστή διαδρομών για μεταφορές με τα πόδια, αυτοκίνητο, ποδήλατο (beta) ή μέσα μαζικής μεταφοράς. Περιλαμβάνει επίσης εντοπισμό των επιχειρήσεων που βρίσκονται σε πόλεις σε πολλές χώρες σε όλο τον κόσμο. Οι δορυφορικές εικόνες των Χαρτών Google δεν ανανεώνονται σε πραγματικό χρόνο, ωστόσο η Google προσθέτει δεδομένα στη Κύρια Βάση δεδομένων της σε τακτική βάση και οι περισσότερες από τις εικόνες δεν είναι πάνω από τριών ετών. Το Google Maps API, επιτρέπει στους προγραμματιστές να ενσωματώσουν το Google Maps στις εφαρμογές τους. Το Google Maps API είναι σχεδιασμένο να λειτουργεί σε κινητές συσκευές και desktop browsers.

2.2 JavaScript Object Notation (JSON)

Το JavaScript Object Notation (JSON) είναι μια εναλλακτική λύση ως προς την XML για την αναπαράσταση δεδομένων. Το JSON είναι μια μορφή ανταλλαγής δεδομένων που βασίζεται σε κείμενο που χρησιμοποιείται για να αναπαραστήσει αντικείμενα στην JavaScript συλλογές ζευγαριών ονόματος/ τιμής που αντιπροσωπεύονται ως String. Χρησιμοποιείται συνήθως σε Ajax εφαρμογές. Το JSON είναι μια απλή μορφή που κάνει αντικείμενα εύκολα στο διάβασμα, στην δημιουργία και ανάλυση και, επειδή είναι πολύ λιγότερο "φλύαρο" από την XML, επιτρέπει σε προγράμματα να μεταδίδουν δεδομένα αποτελεσματικά σε ολόκληρο το Διαδίκτυο. Κάθε αντικείμενο JSON αντιπροσωπεύεται ως μία λίστα ονομάτων ιδιοτήτων και τιμών που περιλαμβάνονται σε άγκιστρα, με την παρακάτω μορφή:

```
{propertyName1: value1, propertyName2 : value2}
```

Οι πίνακες αντιπροσωπεύονται στο JSON με τετράγωνες αγκύλες με την παρακάτω μορφή:

```
{value1, value2, value3}
```

Κάθε τιμή σ' έναν πίνακα μπορεί να είναι μία συμβολοσειρά, ένας αριθμός, ένα αντικείμενο JSON, true, false ή null. Για να εκτιμήσετε την απλότητα των δεδομένων JSON,ας εξετάσουμε αυτήν την αναπαράσταση ενός πίνακα καταχωρήσεων σε μία ατζέντα διευθύνσεων:

```
[ { first : 'Cheryl', last: 'Black'},  
  { first : 'James', last: 'Blue'},  
  { first : 'Mike', last: 'Brown'},  
  { first : 'Meg', last: 'Gold'} ]
```

Πολλές γλώσσες προγραμματισμού υποστηρίζουν τώρα την μορφή δεδομένων JSON. Μία εκτενής λίστα των JSON βιβλιοθηκών ταξινομημένων κατά γλώσσα μπορείτε να βρείτε στη διεύθυνση www.json.org [1].

2.3 SQL Lite

Το Android SDK περιλαμβάνει ένα βοηθητικό πρόγραμμα που μας επιτρέπει την αναζήτηση μιας βάσης δεδομένων SQLite. Η SQLite περιέχει όλες τις ρυθμίσεις και τις προτιμήσεις για μια εφαρμογή Android . Μπορούμε να χρησιμοποιήσουμε τη βάση δεδομένων SQLite σε ένα τηλέφωνο Android για να ανακτήσουμε πληροφορίες για το Android app. Οι Android εφαρμογές κωδικοποιούνται στη γλώσσα Java , έτσι ώστε να χρησιμοποιεί τη γλώσσα Java και το πρόγραμμα οδήγησης JDBC για να συνδεθούμε και να ανακτήσουμε τα αρχεία σε ένα κινητό τηλέφωνο ή δισκίο. Μπορούμε να εκτελέσουμε ερωτήματα σε βάση αποθηκευμένη στην συσκευή.

ΕΠΙΛΟΓΟΣ

Σε αυτό το κεφάλαιο είδαμε μερικές από τις τεχνολογίες του Android που χρησιμοποιήσαμε για την υλοποίηση της εφαρμογής.

ΚΕΦΑΛΑΙΟ 3 Δομή Eclipse (ADT Plugin)

ΕΙΣΑΓΩΓΗ

Οι περισσότερες εφαρμογές στο Android βασίζονται στην γλώσσα προγραμματισμού Java. Έτσι, ο κάθε προγραμματιστής μπορεί να χρησιμοποιήσει έναν οποιονδήποτε text editor για να γράψει κώδικα για να επεξεργαστεί τα αρχεία *.Java και *.XML και μετέπειτα να μεταγλωττίσει τα αρχεία μέσω γραμμής εντολών. Ο συγκεκριμένος τρόπος ανάπτυξης δεν είναι ιδιαίτερα φιλικός προς το χρήστη γι' αυτό συνίσταται η χρήση ενός IDE (Integrated Development Environment) που να υποστηρίζει Java, όπως το Eclipse ή το Netbeans.

Το Eclipse είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης λογισμικού με δυνατότητα εύκολης επεκτασιμότητας χάρη στο σύστημα modules που διαθέτει. Είναι γραμμένο κυρίως στη γλώσσα προγραμματισμού Java και μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών σε Java και μέσω των διαφόρων modules μπορεί να υποστηρίξει και επιπλέον γλώσσες προγραμματισμού όπως C, C++, Perl, PHP, Python, Ruby και άλλες. Το Eclipse είναι μια εφαρμογή Ελεύθερου Λογισμικού καθώς διατίθεται κάτω από την άδεια Eclipse Public License. Αναπτύσσεται ραγδαία από την παγκόσμια κοινότητα του Ελεύθερου Λογισμικού ενώ χρηματοδοτείται από το ίδρυμα Eclipse και από διάφορες δωρεές.

3.1 Android SDK

Το πρώτο βήμα στην πορεία ανάπτυξης της εφαρμογής είναι η εγκατάσταση και ρύθμιση του Android SDK. Το Android SDK παρέχει τα εργαλεία που χρειάζονται για την ανάπτυξη εφαρμογών στην πλατφόρμα Android. Περιλαμβάνει παραδείγματα εφαρμογών με τον πηγαίο τους κώδικα, εργαλεία ανάπτυξης, ένα εξομοιωτή και τις απαιτούμενες βιβλιοθήκες για την ανάπτυξη των εφαρμογών στη γλώσσα προγραμματισμού Java. Ακόμη αναλαμβάνει τη μεταγλώττιση του πηγαίου κώδικα ώστε να τρέχει στην εικονική μηχανή Dalvik.

3.2 Android SDK Manager

Το Android SDK παρέχει εργαλεία, πλατφόρμες και αντικείμενα σε πακέτα που μπορούμε να κατεβάσουμε χρησιμοποιώντας το SDK Manager.[12] Το Android SDK Manager εμφανίζει τα SDK πακέτα που είναι διαθέσιμα, που έχουν ήδη εγκατασταθεί, ή για κάθε αναβάθμιση που είναι διαθέσιμη. Για να μπορέσουμε να τρέξουμε την εφαρμογή μας πρέπει να κατεβάσουμε έστω μια SDK πλατφόρμα. Επειδή η εφαρμογή θα τρέξει σε περισσότερες από μια εκδόσεις (τις οποίες είδαμε στο κεφάλαιο 1) προτεινόμενο είναι να την ελέγξουμε σε πολλές πλατφόρμες στον εξομοιωτή, από την μικρότερη στην μεγαλύτερη, που θέλουμε να υποστηρίζεται.

3.3 Android Virtual Device

Ο εξομοιωτής του Android χρησιμοποιεί το Android Virtual Device (AVD). Το AVD επιτρέπει τον ορισμό διαφόρων πτυχών του υλικού του μιμούμενου τηλεφώνου και επιτρέπει τη δημιουργία πολλών συνθέσεων ώστε να υπάρχει ποικιλία παραλλαγών υλικού διασφαλίζοντας την ομαλή λειτουργία της εφαρμογής σε περισσότερες συσκευές Android. Όταν η εφαρμογή λειτουργεί με τον εξομοιωτή, μπορεί να χρησιμοποιήσει τις υπηρεσίες της πλατφόρμας Android για να επικαλεστεί άλλες εφαρμογές, να έχει πρόσβαση στο δίκτυο, να αναπαράγει ήχο και βίντεο, να αποθηκεύει και να ανακτά δεδομένα, να ενημερώνει το χρήστη και άλλα. Ο εξομοιωτής περιλαμβάνει επίσης μια ποικιλία από δυνατότητες εντοπισμού σφαλμάτων, όπως μια κονσόλα στην οποία καταγράφονται τα μηνύματα εξόδου του πυρήνα, προσομοίωση τυχόν διακοπών της εφαρμογής (όπως όταν φθάνουν μηνύματα SMS ή τηλεφωνικές κλήσεις), καθώς και απορρίψεις στο δίκτυο δεδομένων.

3.4 Android Development Tools

Η Google υποστηρίζει επίσημα το Eclipse και έχει αναπτύξει ειδικά για αυτό ένα επιπρόσθετο plug in, το Android Development Tools, το οποίο συνδέει το Eclipse με το Android SDK και όλες τις δυνατότητες του. Επίσης το plug in παρέχει σύνδεση με τον διαχειριστή εικονικών συσκευών του Android SDK, για τη διαχείριση και εκκίνηση των εικονικών συσκευών μέσω γραφικής διεπαφής. Έτσι διευκολύνει τις δοκιμές σε διάφορους τύπους συσκευών καθώς και στην αποσφαλμάτωση τους.

3.5 Android Debug Bridge (ADB)

Android Debug Bridge (ADB) είναι ένα ευέλικτο εργαλείο γραμμής εντολών που μας επιτρέπει να επικοινωνούμε με ένα εξομοιωτή ή μια συνδεδεμένη συσκευή Android. Πρόκειται για ένα πρόγραμμα client-server που περιλαμβάνει 3 αντικείμενα:

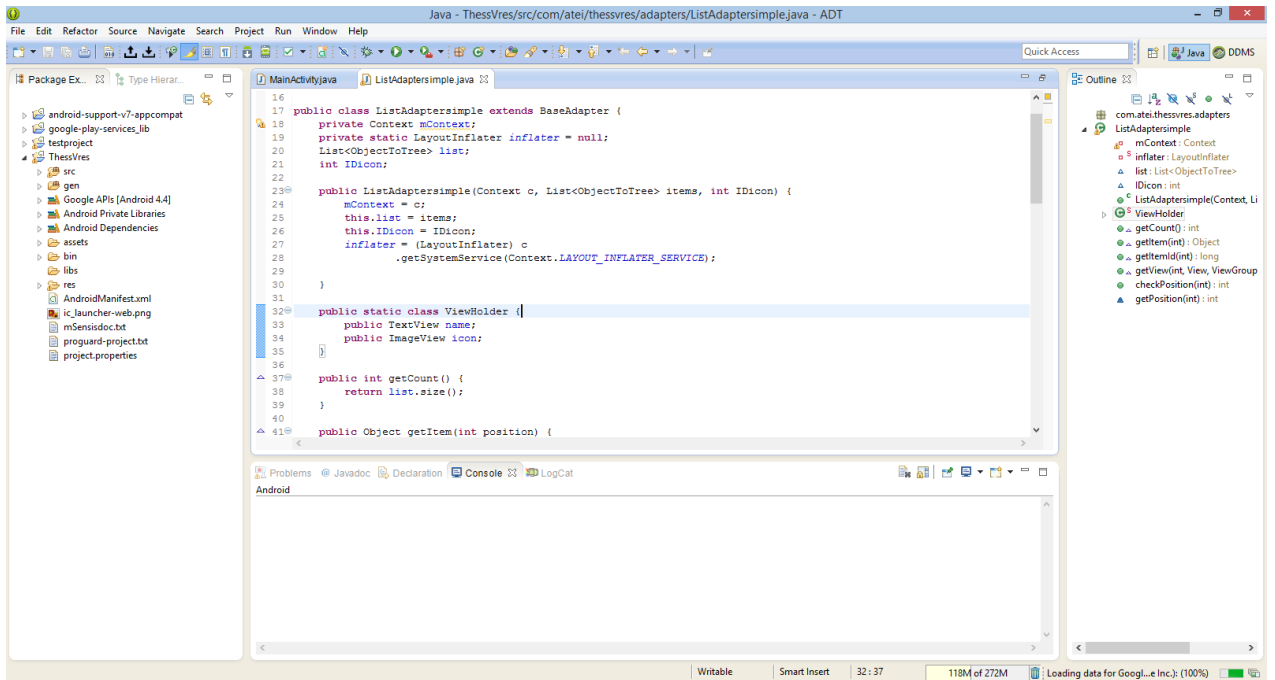
- Έναν client ο οποίος τρέχει στον υπολογιστή που έχουμε στήσει το SDK. Μπορούμε είτε να τον εκκινήσουμε χειροκίνητα είτε να χρησιμοποιήσουμε κάποιο εργαλείο το οποίο ξεκινάει αυτόματα δικό του client, όπως το DDMS ή το ADT Plug in.
- Έναν server ο οποίος τρέχει σαν διεργασία παρασκηνίου στον υπολογιστή. Ο server εξασφαλίζει την επικοινωνία μεταξύ του client και του εργαλείου adb daemon που τρέχει σε έναν εξομοιωτή ή τη κινητή συσκευή.
- Ο daemon που τρέχει σαν διεργασία παρασκηνίου σε κάθε εξομοιωτή ή κινητή συσκευή

3.6 Εγκατάσταση και προετοιμασία του Eclipse

Για να μπορέσουμε να αναπτύξουμε εφαρμογές android θα πρέπει να εγκαταστήσουμε το Eclipse, το SDK του Android και το ADT Plug in.

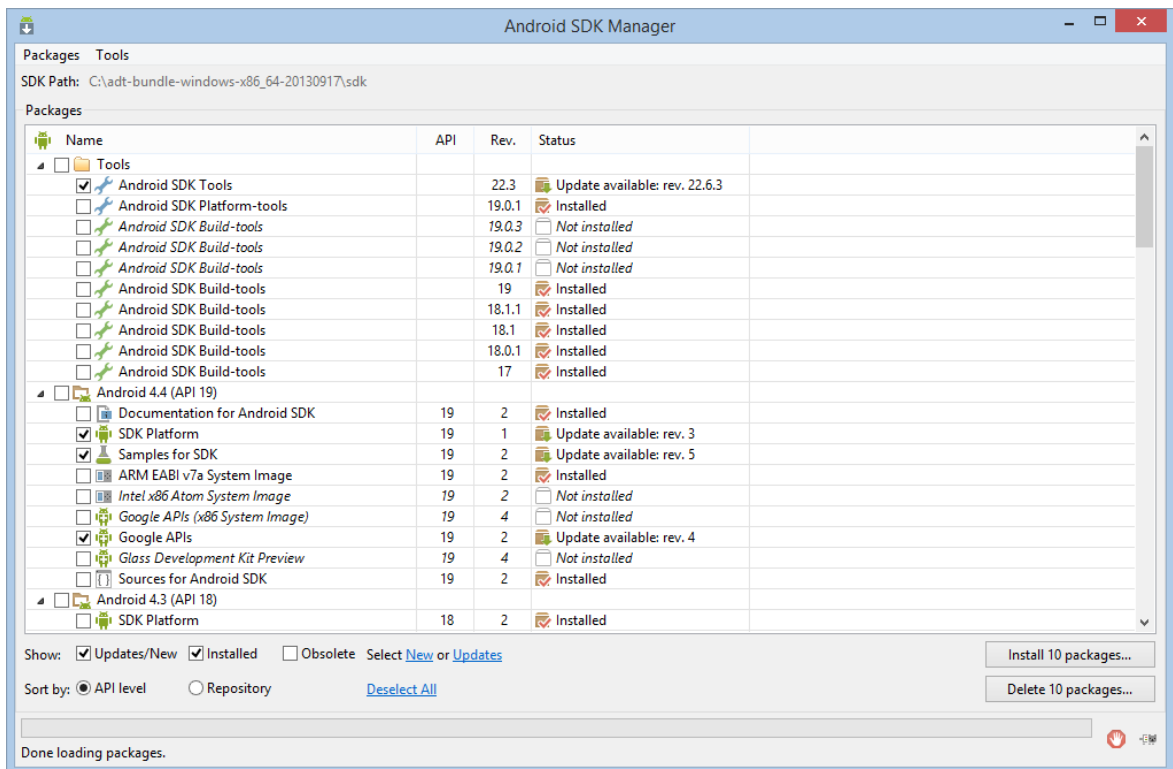
Από την σελίδα <https://www.eclipse.org/downloads/> κατεβάζουμε το Eclipse Standard και το εγκαθιστούμε.

Πτυχιακή εργασία των φοιτητών Ζαχαρία Γιακουμή και Ευαγγελία Παζαρλόγλου



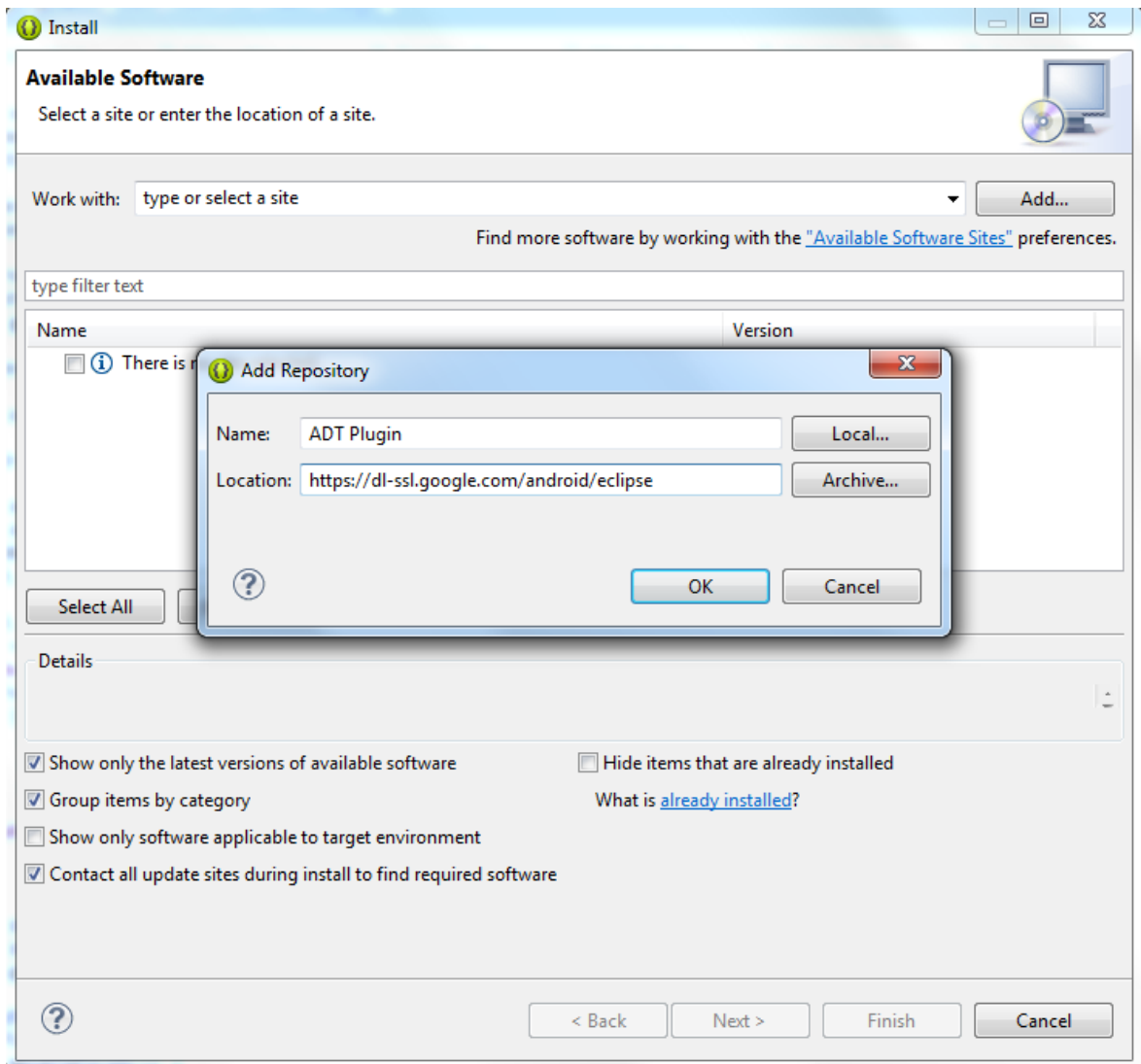
Εικόνα 15. Το περιβάλλον του Eclipse

Στην συνέχεια κατεβάζουμε από τη σελίδα του Android <http://developer.android.com/sdk/index.html> το Android SDK. Αφού κάνουμε την εγκατάσταση του SDK ανοίγουμε το Android SDK Manager και επιλέγουμε να εγκαταστήσουμε τα πακέτα που μας ενδιαφέρουν.



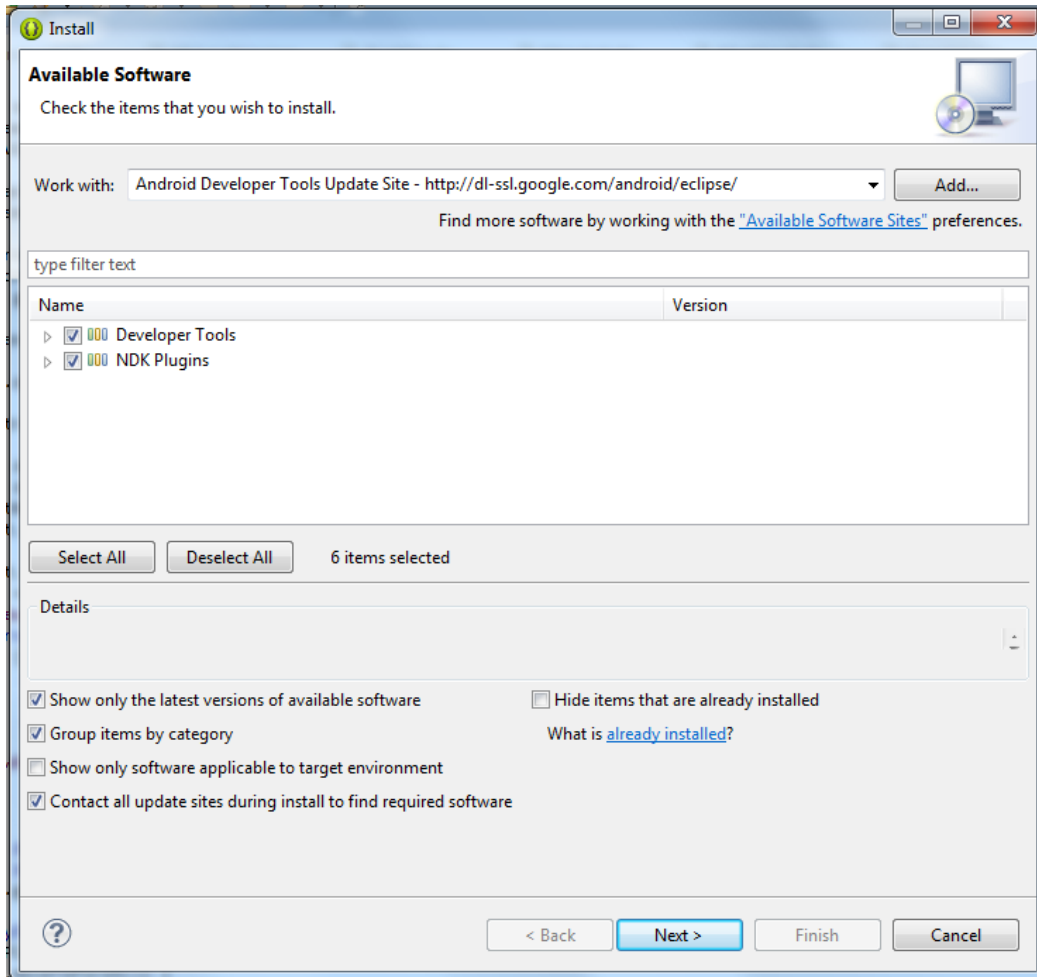
Εικόνα 16. Android SDK Manager

Όταν τελειώσει η εγκατάσταση πάμε στο Eclipse και επιλέγουμε Help>Install New Software. Στο παράθυρο “Add Repository” που θα ανοίξει επιλέγουμε “Add”, βάζουμε ένα όνομα στο “Name” και στο “Location” την <https://dl-ssl.google.com/android/eclipse/> και στη συνέχεια πατάμε “OK”.



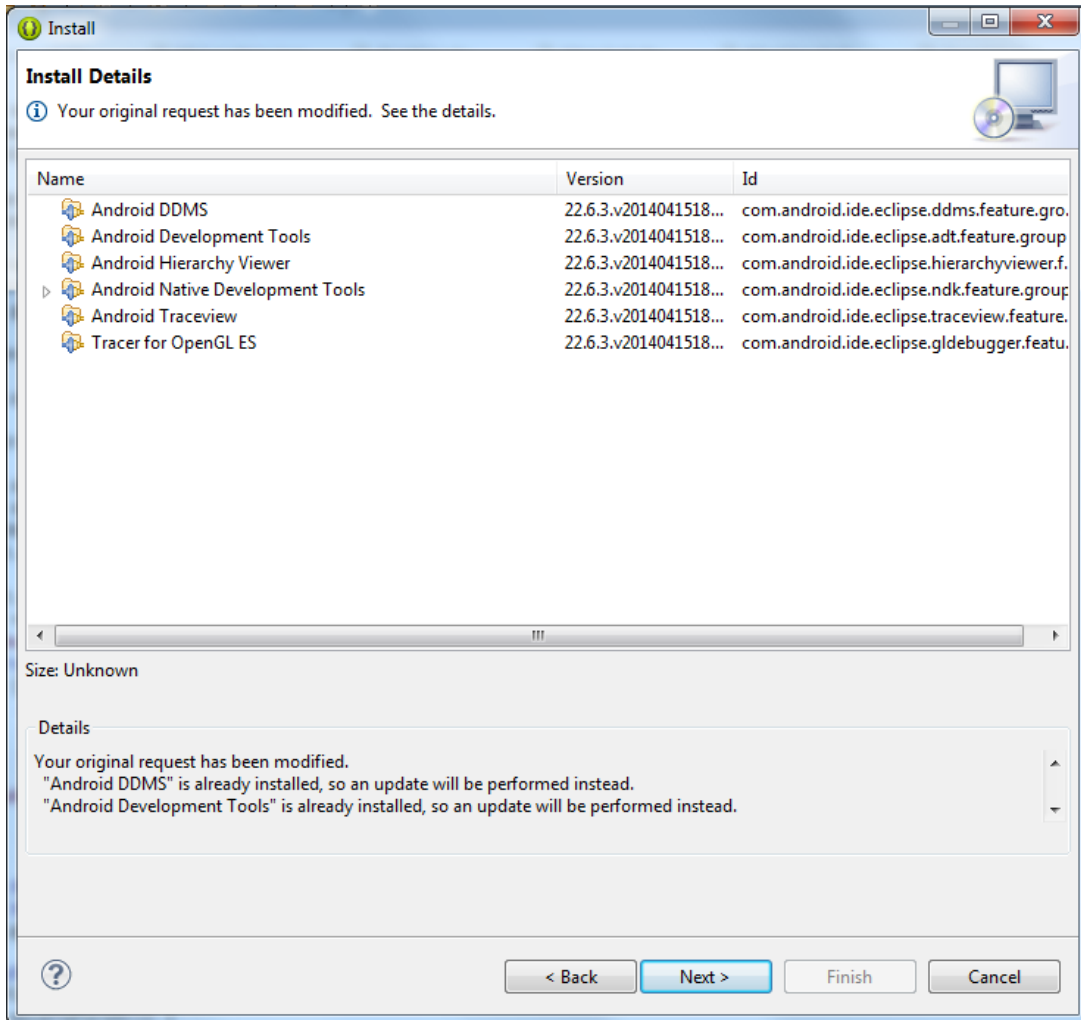
Εικόνα 17. Εγκατάσταση του ADT Plugin (1)

Στο παράθυρο “Available Software”, επιλέγουμε το “Developer tools” και στη συνέχεια πατάμε “Next >” .




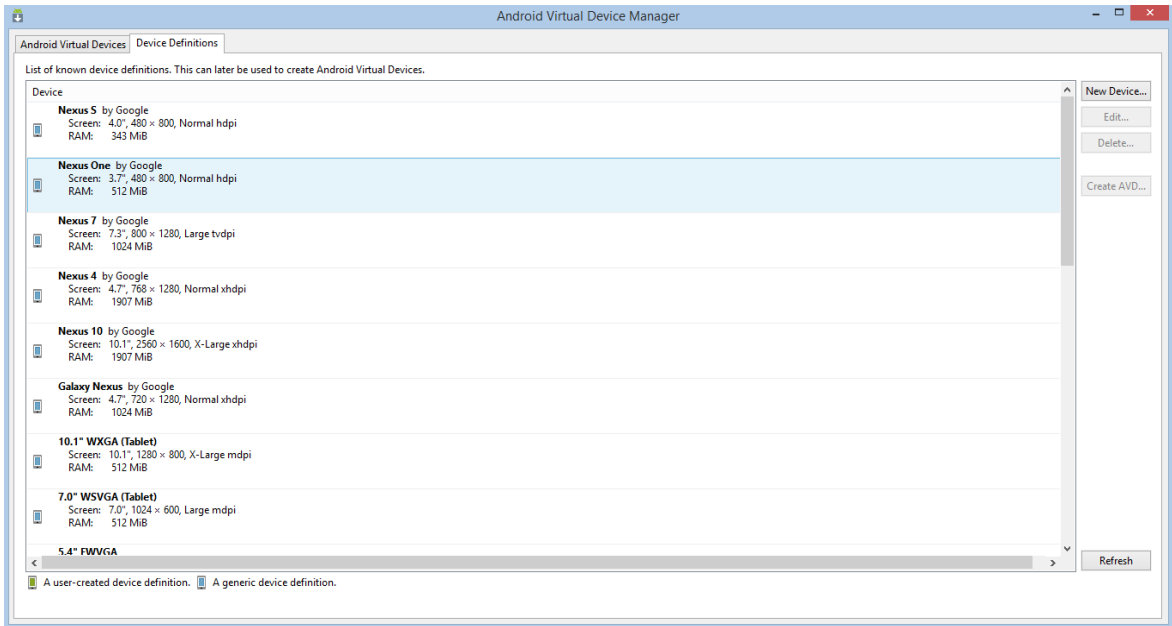
Εικόνα 18. Εγκατάσταση του ADT Plugin (2)

Στο επόμενο παράθυρο βλέπουμε τη λίστα με τα “Developer tools” που θα κατεβάσουμε και πατάμε “Next >”. Έπειτα βλέπουμε την καρτέλα με τους όρους όπου πατάμε “Accept” (αφού διαβάσουμε τους όρους και συμφωνήσουμε με αυτούς) και μετά πατάμε “Finish”.



Εικόνα 19. Εγκατάσταση του ADT Plugin (3)

Στη συνέχεια κάνουμε επανεκκίνηση του eclipse και αν όλα πήγαν καλά τότε βλέπουμε στην οθόνη ότι εμφανιστήκαν τα 2 εικονίδια  το πρώτο είναι το Android SDK Manager (βλ. Εικόνα 16) και το δεύτερο το Android Virtual Device Manager (βλ. Εικόνα 20).



Εικόνα 20. Android virtual Device Manager



Εικόνα 21. Εικονική συσκευή όπου τρέχει η εφαρμογή

ΕΠΙΛΟΓΟΣ

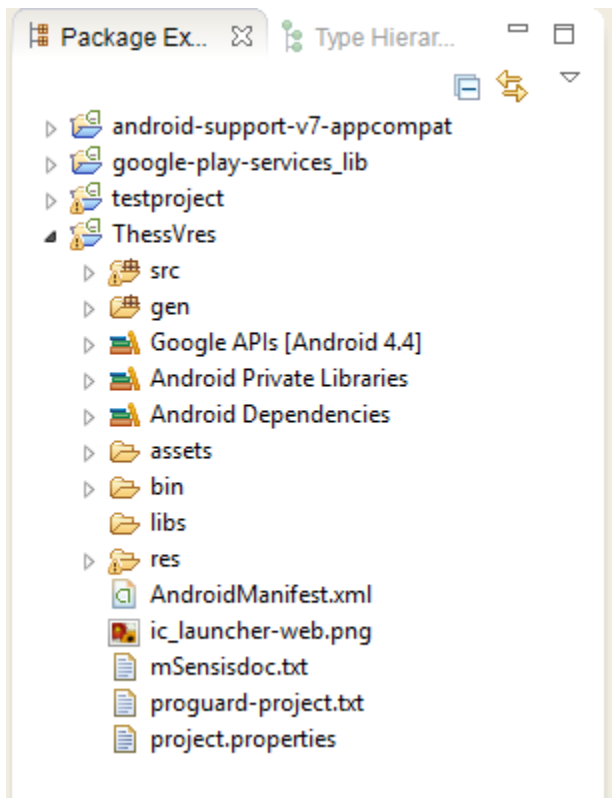
Σε αυτό το κεφάλαιο είδαμε ότι η Google υποστηρίζει επίσημα το Eclipse και έχει αναπτύξει ειδικά για αυτό το ADT plugin, το οποίο παρέχει σύνδεση με το Android SDK με όλες τις δυνατότητες που περιλαμβάνει αυτό. Επίσης το plugin παρέχει σύνδεση με τον AVD Manager, για διαχείριση και εκκίνηση από το GUI του, εικονικών συσκευών Android για δοκιμές και debugging των εφαρμογών.

ΚΕΦΑΛΑΙΟ 4 Δομή μίας Εφαρμογής

Οι εφαρμογές για Android συνήθως είναι γραμμένες στη γλώσσα προγραμματισμού Java. Τα εργαλεία του Android SDK μεταγλωττίζουν τον πηγαίο κώδικα μαζί με τα απαραίτητα συνοδευτικά αρχεία, όπως αρχεία εικόνων ή ήχου, σε ένα πακέτο Android. Αυτό το πακέτο έχει την επέκταση αρχείου .apk και αποτελεί ουσιαστικά την εφαρμογή η οποία χρησιμοποιείται κατά τη διαδικασία της εγκατάστασης.

4.1 Δομή του πηγαίου κώδικα

Ο πηγαίος κώδικας μιας Android εφαρμογής είναι αρκετά καλά δομημένος, λογικά χωρισμένος σε αρκετούς φακέλους και αρχεία.



Εικόνα 22. Η δομή μιας εφαρμογής

4.2 Android Activities

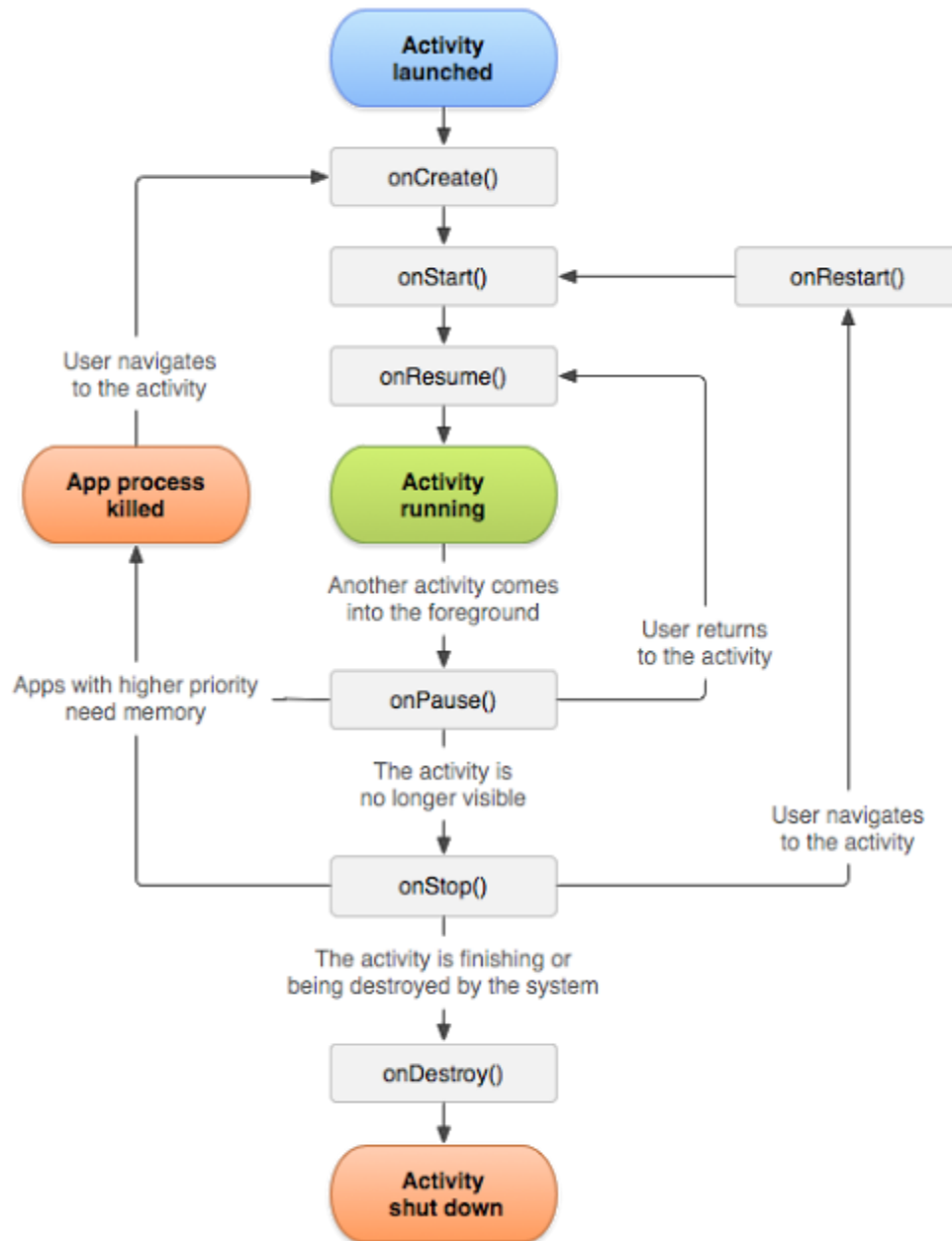
Activity είναι ένα συστατικό της εφαρμογής το οποίο παρέχει τη διεπαφή με την οποία αλληλεπιδρά ο χρήστης ώστε να κάνει κάποιες ενέργειες, όπως για παράδειγμα κλήση μιας επαφής, αποστολή ενός mail, τράβηγμα φωτογραφίας κλπ. Κάθε οθόνη που βλέπει και αλληλεπιδρά ο χρήστης είναι μια κλάση που κληρονομεί την κλάση activity.

Μια εφαρμογή συνήθως αποτελείται από πολλά activities που συνδέονται μεταξύ τους. Το πιο κοινό σενάριο είναι να υπάρχει ένα κεντρικό activity, το οποίο παρουσιάζεται στο χρήστη κατά την εκτέλεση της εφαρμογής. Κάθε activity μπορεί να ξεκινήσει ένα άλλο ώστε να εκτελέσει διαφορετικές ενέργειες. Κάθε φορά που ένα νέο activity ξεκινά, το προηγούμενο σταματάει, αλλά το σύστημα το βάζει σε μια στοίβα. Έτσι όταν ο χρήστης ολοκληρώσει τις ενέργειες του σε αυτή το activity και πατήσει το πλήκτρο επιστροφής, το προηγούμενο activity θα επαναφερθεί από τη στοίβα και θα συνεχίσει την εκτέλεσή της.

Κάθε Activity έχει ένα συγκεκριμένο κύκλο ζωής και κάποιες καταστάσεις, όπως για παράδειγμα Resumed, Paused και Stopped. Όταν μια Activity περνάει σε μια κατάσταση αυτό αναγνωρίζεται από κάποιες callback methods. Μια τέτοια callback method είναι η onCreate την οποία πρέπει να υλοποιήσει ο προγραμματιστής και εκεί να ορίσει με την συνάρτηση setContentView() τη γραφική διεπαφή που θα έχει το Activity.

Πιο αναλυτικά οι καταστάσεις ενός Activity είναι:

- onStart
- onResume
- onRestart
- onPause
- onStop
- onDestroy



Εικόνα 23. Οι καταστάσεις ενός Activity

4.3 Το αρχείο AndroidManifest.xml

Οι εφαρμογές Android αποτελούνται από αρκετά συστατικά τα οποία συνδέονται μεταξύ τους χρησιμοποιώντας ένα κεντρικό XML αρχείο, το AndroidManifest.xml. Το αρχείο αυτό ορίζει χαρακτηριστικά του hardware και του software που χρησιμοποιεί ή απαιτεί η εφαρμογή. Δηλώνει τα δικαιώματα που απαιτεί η εφαρμογή, για παράδειγμα πρόσβαση στο Internet ή δικαιώματα ανάγνωσης των επαφών . Ακόμη, ορίζει την ελάχιστη έκδοση του API που

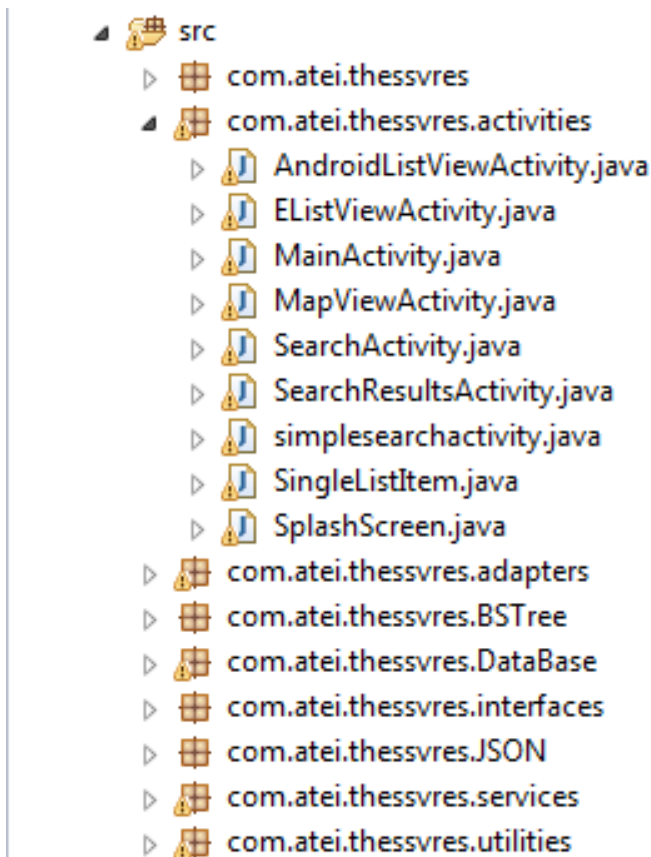
απαιτείται για τη σωστή εκτέλεση της εφαρμογής . Τέλος, δηλώνει τις βιβλιοθήκες του API με τις οποίες πρέπει να συνδεθεί η εφαρμογή πέρα του Android framework API, όπως για παράδειγμα τη βιβλιοθήκη Google Maps.

4.4 Κατάλογος src

Το όνομα αυτού του καταλόγου έχει προέρθει από την συντομογραφία της Αγγλικής λέξης source που σημαίνει «πηγή». Σε αυτόν βρίσκεται όλος ο πηγαίος κώδικας της εφαρμογής, είτε είναι ένα αρχείο δηλαδή μια κλάση είτε είναι πολλές.

Μπορούμε να δημιουργήσουμε πακέτα για να χωρίσουμε τον κώδικα μας σε κατηγορίες όπως adapters, για πιο εύκολη προσβαση

Για την εφαρμογής μας η δομή του είναι η εξής:



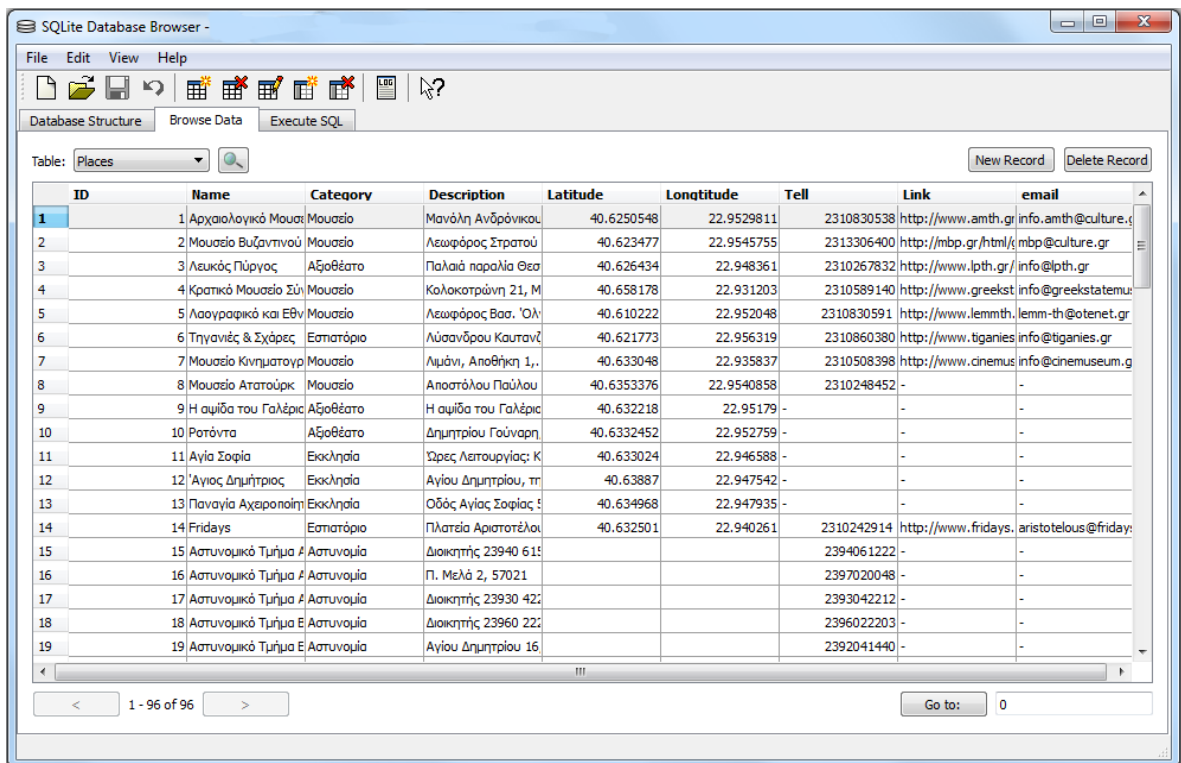
Εικόνα 24. Ο κατάλογος src

4.5 Κατάλογος assets

Ο κατάλογος assets είναι ένας φάκελος στον οποίο ο προγραμματιστής μπορεί να τοποθετήσει read-only αρχεία που υποστηρίζονται από το λειτουργικό σύστημα και τα οποία χρησιμοποιεί στην εφαρμογή για παράδειγμα δεδομένα ενός παιχνιδιού. Τα αρχεία που αποθηκεύονται στον συγκεκριμένο κατάλογο είναι δεν δίνεται κάποια resource ID ώστε να αναφερθούν σε αυτά μέσω της R κλάσης ή κάπου άλλου XML πόρου, όπως γίνεται στα αρχεία του φακέλου res (που θα δούμε παρακάτω). Αντίθετα μπορούμε να τα αναζητήσουμε αρχεία στον κατάλογο assets σαν να είναι ένα κανονικό σύστημα αρχείων.

Τα αρχεία δεν έχουν κάποια επέκταση πχ .txt. Αποθηκεύονται μόνο με το όνομα τους και τα δεδομένα τους ανακτούνται χρησιμοποιώντας την κλάση AssetManager και πιο συγκεκριμένα την μέθοδο getAssets().

Σε αυτό τον κατάλογο μπαίνει η βάση δεδομένων της SQLite που αναφέραμε στο δεύτερο κεφάλαιο (2.3)

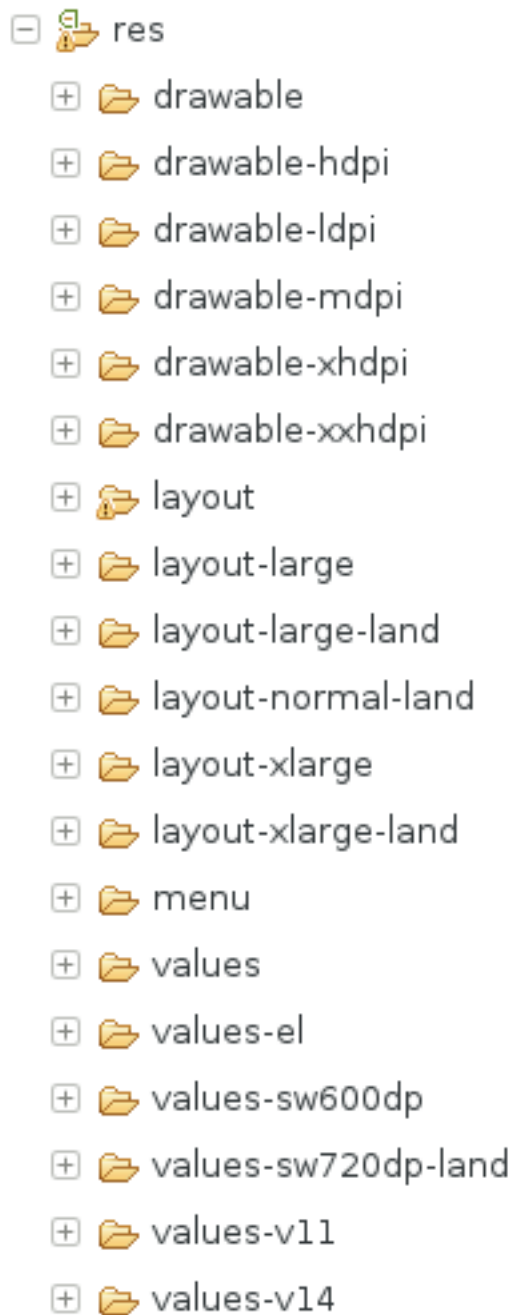


ID	Name	Category	Description	Latitude	Longitude	Tell	Link	email
1	1 Αρχαιολογικό Μουσείο	Μουσείο	Μανόλη Ανδρόνικου	40.6250548	22.9529811	2310830538	http://www.amth.gr	info.amth@culture.gr
2	2 Μουσείο Βυζαντινού	Μουσείο	Λεωφόρος Στρατού	40.623477	22.9545755	2313306400	http://mbp.gr/html/mbp@culture.gr	
3	3 Λευκός Πύργος	Αξιοθέατο	Παλαιά παραλία Θεσ	40.626434	22.948361	2310267832	http://www.lpth.gr	info@lpth.gr
4	4 Κρατικό Μουσείο Σύ	Μουσείο	Κολοκοτρώνη 21, Μ	40.658178	22.931203	2310589140	http://www.greekst	info@greekstatemu
5	5 Λαογραφικό και Εθν	Μουσείο	Λεωφόρος Βασ. Όλ	40.610222	22.952048	2310830591	http://www.lemmth	lemm-th@otenet.gr
6	6 Τηγανιές & Σχάρες	Εσπατόριο	Λύσανδρου Κραυτανή	40.621773	22.956319	2310860380	http://www.tiganies	info@tiganies.gr
7	7 Μουσείο Κινηματογρ	Μουσείο	Λιμάνι, Αποθήκη 1,	40.633048	22.935837	2310508398	http://www.cinemus	info@cinemuseum.g
8	8 Μουσείο Ατατούρκ	Μουσείο	Αποστόλου Παύλου	40.6353376	22.9540858	2310248452	-	-
9	9 Η αμίδα του Γαλέρι	Αξιοθέατο	Η αμίδα του Γαλέρι	40.632218	22.95179	-	-	-
10	10 Ροτόνα	Αξιοθέατο	Δημητρίου Γούναρη	40.6332452	22.952759	-	-	-
11	11 Αγία Σοφία	Εκκλησία	Ώρες Λειτουργίας: Κ	40.633024	22.946588	-	-	-
12	12 Άγιος Δημήτριος	Εκκλησία	Άγιου Δημητρίου, πτ	40.63887	22.947542	-	-	-
13	13 Παναγία Αχειροποίη	Εκκλησία	Οδός Αγίας Σοφίας :	40.634968	22.947935	-	-	-
14	14 Fridays	Εσπατόριο	Πλατεία Αριστοτέλοι	40.632501	22.940261	2310242914	http://www.fridays	aristotelous@friday
15	15 Αστυνομικό Τμήμα Α	Αστυνομία	Διοικητής 23940 615			2394061222	-	-
16	16 Αστυνομικό Τμήμα Α	Αστυνομία	Π. Μελά 2, 57021			2397020048	-	-
17	17 Αστυνομικό Τμήμα Α	Αστυνομία	Διοικητής 23930 422			2393042212	-	-
18	18 Αστυνομικό Τμήμα Ε	Αστυνομία	Διοικητής 23960 222			2396022203	-	-
19	19 Αστυνομικό Τμήμα Ε	Αστυνομία	Άγιου Δημητρίου 16			2392041440	-	-

Εικόνα 25. Παράδειγμα μιας βάσης SQLite

4.6 Κατάλογος res

Ο κατάλογος `res`, το όνομα του οποίου είναι μια συντομογραφία της Αγγλικής λέξης `resources`, περιέχει τους πόρους (`resources`) της εφαρμογής. Οι πόροι μπορούν να είναι κείμενο, εικόνες, ήχος ή πιο απλά ότι δεν είναι κώδικας Java. Σε αυτόν τον κατάλογο αποθηκεύονται οι πόροι ανάλογα με τον τύπο τους. Για παράδειγμα, αν υπάρχει μια εικόνα τύπου `jpeg` ή `png` τότε πρέπει να αποθηκευτεί στον αντίστοιχο κατάλογο.



Εικόνα 26. Ο κατάλογος `res`

Όλοι οι πόροι μεταγλωττίζονται από τον `resource compiler` (μεταγλωττιστή πόρων), ο οποίος τους συμπιέζει και τους πακετάρει δημιουργώντας μια κλάση με το όνομα `R` η οποία περιέχεται στον κατάλογο με το όνομα `gen/`. Η κλάση `R` περιέχει τα αναγνωριστικά που χρησιμοποιούνται στον πηγαίο κώδικα έτσι ώστε να επιτρέπει την αναφορά σε αυτούς τους πόρους μέσα από την εφαρμογή.

Ο κατάλογος `res` χωρίζεται σε αρκετούς υποκαταλόγους, ενώ ο προγραμματιστής είναι ελεύθερος να δημιουργήσει δικούς του ανάλογα με τις ανάγκες του. Μια τυπική εφαρμογή Android χρησιμοποιεί τους καταλόγους `res/drawable`, `res/layout` και `res/values` χωρίς όμως αυτό να είναι δεσμευτικό.

4.5.1 Ο κατάλογος `res/drawable`

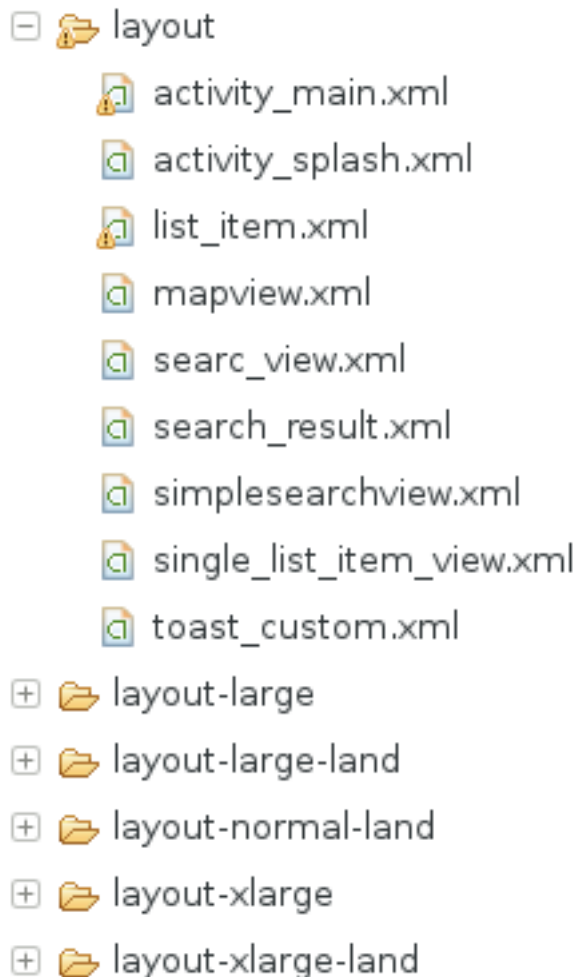
Στον κατάλογο `res/drawable` βρίσκονται όλα τα αρχεία που μπορούν να αποτυπωθούν στην οθόνη της συσκευής. Αυτό σημαίνει ότι εκεί δεν βρίσκονται μόνο οι εικόνες της εφαρμογής, αλλά και σχήματα, προκαθορισμένα στυλ κουμπιών και άλλα. Αναλυτικότερα, στον κατάλογο αυτό μπορούμε να βρούμε :

- Αρχεία εικόνων
Το Android υποστηρίζει τα αρχεία εικόνων τύπου `.png`, `.jpg`, και `.gif`
- Αρχεία Nine-Patch
Πρόκειται για εικόνες τύπου PNG που μπορούν να προσαρμοστούν σε περιοχές, όπως τα κουμπιά, επιτρέποντας την αλλαγή μεγέθους ανάλογα με το περιεχόμενο.
- Λίστες καταστάσεων
Πρόκειται για αρχεία τύπου XML τα οποία ορίζουν διαφορετικές εικόνες για διαφορετικές καταστάσεις. Για παράδειγμα εναλλαγή εικόνας σε ένα κουμπί όταν είναι πατημένο
- Σχέδιο σχήματος
Πρόκειται για αρχεία τύπου XML τα οποία ορίζουν ένα γεωμετρικό σχήμα, συμπεριλαμβανομένων χρωμάτων, περιγράμματος

Ο κατάλογος `res/drawable` μπορεί να υπάρχει σε αρκετές διαφορετικές εκδοχές. Το Android δίνει τη δυνατότητα να ορίζονται και χρησιμοποιούνται διαφορετικοί πόροι τις εφαρμογής ανάλογα με τη συσκευή. Οπότε για παράδειγμα μπορεί να υπάρχει ο κατάλογος `res/drawable-ldpi` ο οποίος να περιέχει εικόνες που προορίζονται για συσκευές με μικρή οθόνη και ταυτόχρονα ο κατάλογος `drawable-hdpi` με εικόνες για συσκευές με μεγάλη οθόνη.

4.5.2 Κατάλογος res/layout

Το Android παρέχει στους προγραμματιστές έναν εύκολο τρόπο να ορίζουν τη γραφική διεπαφή της εφαρμογής τους. Πέρα από τον παραδοσιακό τρόπο της δημιουργίας αντικειμένων προγραμματιστικά, δίνει τη δυνατότητα να δημιουργούνται αρχεία διατάξεων (layout files) τα οποία ορίζουν την δομή της εμφάνισης μιας γραφικής διεπαφής, όπως είναι ένα Activity ή ένα widget. Φυσικά μπορούν να χρησιμοποιηθούν ταυτόχρονα και οι δυο τρόποι αν και προτιμάται η χρήση των αρχείων XML. Τα παραπάνω αρχεία διατάξεων αποθηκεύονται στον κατάλογο res/drawable.



Εικόνα 27. Ο κατάλογος layout

Τα πλεονεκτήματα της δήλωσης της γραφικής διεπαφής σε αρχεία XML είναι αρκετά. Με αυτό τον τρόπο διαχωρίζεται η αναπαράσταση της εφαρμογής

από τον κώδικα που ελέγχει τη συμπεριφορά της. Η περιγραφή της γραφικής διεπαφής είναι ανεξάρτητη από τον πηγαίο κώδικα της εφαρμογής, πράγμα που σημαίνει ότι ανά πάσα στιγμή μπορεί πολύ εύκολα να αλλάξει η εμφάνιση μια εφαρμογής χωρίς να πειραχτεί καθόλου ο πηγαίος της κώδικας. Ακόμη με αυτόν τον τρόπο γίνεται καλύτερη οπτικοποίηση της γραφικής διεπαφής με αποτέλεσμα να είναι πολύ ευκολότερη η εύρεση λαθών και η αποσφαλμάτωση της εφαρμογής.

Με τη χρήση του λεξιλογίου XML του Android μπορεί εύκολα και γρήγορα να οριστεί η διάταξη της εφαρμογής και τα αντικείμενα που την αποτελούν με παρόμοιο τρόπο όπως η δημιουργία ιστοσελίδων με τη χρήση της γλώσσας HTML. Κάθε αρχείο διάταξης πρέπει να περιέχει μόνο ένα γονικό αντικείμενο το οποίο με τη σειρά του μπορεί να περιέχει ένα ή περισσότερα αντικείμενα παιδιά.

Μερικά από τα αντικείμενα που μπορούν να χρησιμοποιηθούν ως γονικά είναι τα παρακάτω ανάλογα με το στυλ της γραφικής διεπαφής της εφαρμογής :

- **Linear Layout**
Τοποθετεί με τη σειρά όλα τα αντικείμενα παιδιά σε μια κατεύθυνση, οριζοντίως ή καθέτως
- **Relative Layout**
Εμφανίζει τα αντικείμενα παιδιά σε σχετικές θέσεις. Η θέση κάθε αντικειμένου μπορεί να οριστεί σε σχέση με ένα άλλο αντικείμενο ή σε σχέση με το αντικείμενο γονέα
- **ListView**
Δημιουργεί μια λίστα από αντικείμενα με δυνατότητα κύλισης. Τα αντικείμενα εισέρχονται αυτόματα στη λίστα με τη χρήση ενός adapter που τραβά το περιεχόμενο από έναν πίνακα ή μια βάση δεδομένων και τα τοποθετεί στη λίστα

Για την δημιουργία των layout files τα xml αρχεία έχουν την παρακάτω μορφή

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

```
<RelativeLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
```

```
<LinearLayout
```

```
android:layout_width="fill_parent"  
android:layout_height="fill_parent"  
android:layout_gravity="center_horizontal|center_vertical"  
android:gravity="center|left"  
android:orientation="horizontal" >
```

```
<LinearLayout  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:gravity="left"  
  android:orientation="horizontal" >
```

```
<ImageView  
  android:id="@+id/imglist"  
  android:layout_width="50dp"  
  android:layout_height="50dp" >  
</ImageView>  
</LinearLayout>
```

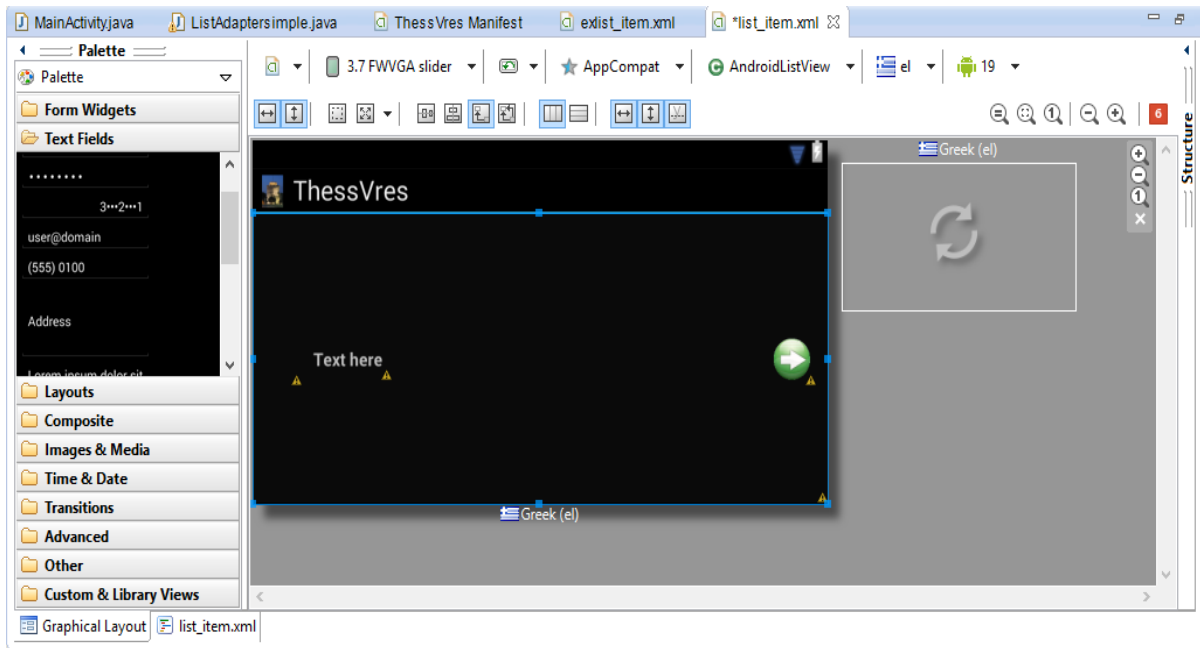
```
<TextView  
  android:id="@+id/label"  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:layout_marginRight="65dp"  
  android:padding="10dip"  
  android:textSize="16dip"  
  android:textStyle="bold" >  
</TextView>  
</LinearLayout>
```

```
<LinearLayout  
  android:layout_width="wrap_content"  
  android:layout_height="fill_parent"  
  android:layout_alignParentRight="true"  
  android:gravity="center" >
```

```
<ImageView  
  android:id="@+id/imgnext"  
  android:layout_width="50dp"  
  android:layout_height="50dp"  
  android:layout_marginRight="10dp"  
  android:gravity="left"  
  android:src="@drawable/next_icon1"  
  android:text="test" />  
</LinearLayout>  
</RelativeLayout>
```

```
</LinearLayout>
```

Στο γραφικό περιβάλλον έχουμε την εξής εμφάνιση:

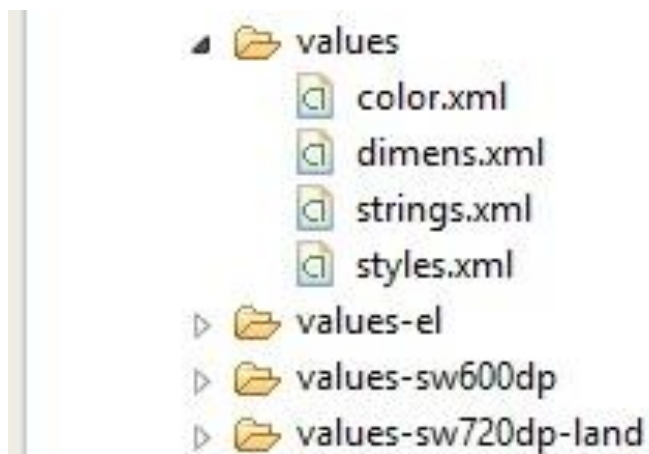


Εικόνα 28. Γραφικό περιβάλλον μίας εφαρμογής

Όπως με τον κατάλογο `res/drawable`, έτσι και με τον κατάλογο `res/layouts` μπορούμε να ορίσουμε διαφορετικές εκδοχές του για την καλύτερη υποστήριξη των διαφόρων συσκευών Android. Για παράδειγμα μπορεί να υπάρχει ο κατάλογος `res/layout-land` στον οποίο να ορίζεται η γραφική διεπαφή για όταν η συσκευή είναι σε οριζόντια θέση.

4.5.3 Κατάλογος `res/values`

Υπάρχουν διάφοροι πόροι που περιγράφουν τη χρήση, τη μορφή και τη σύνταξη για έναν συγκεκριμένο τύπο της εφαρμογής και που μπορούμε να τους προσθέσουμε στον κατάλογο πόρων μας.



Εικόνα 29. Ο κατάλογος values

Στην κατηγορία res/values δύο είναι οι κυριότεροι:

- **String**
Πόροι αυτού του τύπου ορίζουν τα αλφαριθμητικά, πίνακες αλφαριθμητικών, πληθυντικούς αριθμούς και χρώματα. Οι πόροι αυτοί αποθηκεύονται στον κατάλογο res/values/.
- **Style**
Ορίζει το πως θα φαίνονται τα στοιχεία διεπαφής χρήστη (UI elements). Πόροι αυτού του τύπου επίσης αποθηκεύονται στον κατάλογο res/values/ και είναι προσβάσιμοι από την κλάση R.style.

- Color

Είναι ένα αρχείο πόρων τύπου XML που παρέχει τιμές χρωμάτων. Αυτές οι τιμές μπορούν να χρησιμοποιηθούν είτε στον πηγαίο κώδικα της εφαρμογής, είτε στα αρχεία διατάξεων. Τα χρώματα ορίζονται με τη δομή των RGB τιμών και διαφάνειας. Η τιμή πρέπει πάντα να ξεκινά με τον χαρακτήρα # και να ακολουθούν οι τιμές Διαφάνεια-Κόκκινο-Πράσινο-Μπλε.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3
4     <color name="white">#FFFFFF</color>
5     <color name="red">#FF0000</color>
6     <color name="cyan">#00FFFF</color>
7     <color name="yellow">#FFFF00</color>
8     <color name="magenta">#FF00FF</color>
9     <color name="blue">#0000FF</color>
10    <color name="black">#000000</color>
11    <color name="purple">#800080</color>
12    <color name="heavy_gray">#424242</color>
13    <color name="medium_gray">#828282</color>
14    <color name="pale_turquoise">#BBFFFF</color>
15    <color name="orange">#FF7F00</color>
16    <color name="green_lime">#00FF00</color>
17    <color name="spring_green">#00FF7F</color> <!-- #FF383838 -->
18
19    <color name="light_gray">#FFBDBDBD</color>
20    <color name="background_gray">#FFCCCCCC</color>
21    <color name="background_color_screens">#FFCCCCCC</color>
22
23 </resources>
```

Εικόνα 30 Παράδειγμα του αρχείου colors.xml που βρίσκεται στον κατάλογο values

Για κάθε γλώσσα υπάρχει ένας αντίστοιχος φάκελος π.χ. για αγγλικά είναι ο κατάλογος values-en, για ισπανικά είναι ο κατάλογος values-es, για ελληνικά είναι ο κατάλογος values-el κλπ. Ανάλογα με την γλώσσα συστήματος της συσκευής, η εφαρμογή βλέπει τον αντίστοιχο κατάλογο. Αν δεν υπάρχει ο κατάλογος χρησιμοποιείται η προεπιλογή που είναι ο κατάλογος values.


```

1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3   <string name="app_name">ThessVres</string>
4   <string name="action_settings">Settings</string>
5   <string name="textView1"></string>
6   <string name="action_map">Χάρτης</string>
7   <string name="search_title">Αναζήτηση</string>
8   <string name="simple_search">Simple Search</string>
9   <string name="h_search">Αναζήτηση...</string>
10  <string name="from">Από: </string>
11  <string name="to">Προορισμός: </string>
12  <string name="clocation">Τρέχουσα θέση</string>
13  <string name="f_hint">Από...</string>
14  <string name="t_hint">Προορισμός...</string>
15  <string name="source">Αφειτηρία: </string>
16  <string name="s_btn">Αναζήτηση</string>
17  <string name="m_btn">Στο χάρτη</string>
18  <string name="gps">Παρακαλώ ανοίξτε το GPS</string>
19  <string name="gps_settings">Επιλογές</string>
20  <string name="gps_cancel">Ακυρο</string>
21  <string name="gps_msg">Το GPS δεν είναι ενεργοποιημένο. Θέλετε να πάτε στις ρυθμίσεις μενού;</string>
22  <string name="phone">Αρ. Τηλεφώνου: </string>
23  <string name="link">Σελίδα: </string>
24  <string name="events">Εκδηλώσεις</string>
25  <string name="tomail">Θέλετε να στείλετε αυτό το μήνυμα;</string>
26  <string name="tosite">Θέλετε να πάτε σε αυτή τη σελίδα;</string>
27  <string name="tocall">Θέλετε να πραγματοποιήσετε κλήση;</string>
28  <string name="yes">Ναι</string>
29  <string name="no">Όχι</string>
30
31 </resources>
32

```

Εικόνα 31. Το αρχείο strings.xml που βρίσκεται στον κατάλογο values

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3   <string name="app_name">ThessVres</string>
4   <string name="action_settings">Settings</string>
5   <string name="textView1"></string>
6   <string name="action_map">Map</string>
7   <string name="search_title">Search</string>
8   <string name="simple_search">Simple Search</string>
9   <string name="h_search">Search Place...</string>
10  <string name="from">from: </string>
11  <string name="to">Destination: </string>
12  <string name="clocation">Current Location</string>
13  <string name="f_hint">From...</string>
14  <string name="t_hint">Destination</string>
15  <string name="s_hint">Search...</string>
16  <string name="source">Source</string>
17  <string name="s_btn">Search</string>
18  <string name="m_btn">On Map</string>
19  <string name="action_list">List</string>
20  <string name="gps">Pls open your GPS</string>
21  <string name="gps_settings">Settings</string>
22  <string name="gps_cancel">Cancel</string>
23  <string name="gps_msg">GPS is not enabled. Do you want to go to settings menu?</string>
24  <string name="phone">Phone Number: </string>
25  <string name="link">Link: </string>
26  <string name="events">Events</string>
27  <string name="email">E-mail: </string>
28  <string name="tocall">Are you sure for calling ?</string>
29  <string name="tomail">Are you sure for send an email ?</string>
30  <string name="tosite">Open this site?</string>
31  <string name="yes">Yes</string>
32  <string name="no">No</string>
33

```

Εικόνα 32. Το αρχείο strings.xml που βρίσκεται στον κατάλογο values-en

4.5.4 Κατάλογος res/menu

Στον κατάλογο res/menu υπάρχουν τα Action Bar που χρησιμοποιούνται στην εφαρμογή σε μορφή xml. Τα action bar είναι στατικό γραφικό στοιχείο στην κορυφή κάθε οθόνης της εφαρμογής. Κάνει τις σημαντικές ενέργειες της εφαρμογής εμφανέστερες και προσβάσιμες με ένα προβλέψιμο τρόπο (για παράδειγμα την Αναζήτηση). Τα εικονίδια για το action bar έχουν τους δικούς τους σχεδιαστικούς κανόνες όπως για παράδειγμα συγκεκριμένο μέγεθος ανάλογα με το μέγεθος της οθόνης.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto" >

  <item
    android:id="@+id/search"
    android:icon="@drawable/ic_action_search"
    android:title="@string/search_title"
    app:showAsAction="always"/>
  <item
    android:id="@+id/action_list"
    android:icon="@drawable/list_actionbar"
    android:title="@string/action_list"
    app:showAsAction="always"/>

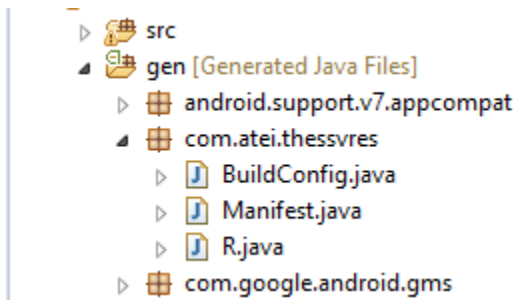
</menu>
```

4.7 Κατάλογος libs

Στον κατάλογο libs/ τοποθετούνται οι εξωτερικές βιβλιοθήκες που μπορεί να χρησιμοποιεί μια εφαρμογή. Χρησιμοποιώντας εξωτερικές βιβλιοθήκες δίνεται η δυνατότητα στους προγραμματιστές να ενσωματώσουν κλάσεις και γενικότερα επιπλέον χαρακτηριστικά σε μια εφαρμογή. Αυτές οι βιβλιοθήκες πρέπει να δηλωθούν ως "linked libraries" μέσα από το Eclipse IDE ώστε να γίνουν link κατά τη μεταγλώττιση της εφαρμογής.

Εφόσον το Android χρησιμοποιεί τη γλώσσα προγραμματισμού Java, έτσι και οι εγγενής βιβλιοθήκες που υποστηρίζει είναι οι βιβλιοθήκες της Java. Πρόκειται για αρχεία που έχουν την κατάληξη .jar και συνήθως περιέχουν πολλές κλάσεις, μετα-πληροφορίες και τους πόρους που χρησιμοποιεί.

4.8 Κατάλογος gen



Εικόνα 33. Ο κατάλογος gen

Στο κατάλογο gen/ βρίσκονται διάφορα αρχεία τα οποία παράγονται αυτόματα από το Android Development Tools plug-in του Eclipse, όπως η κλάση R η οποία κρατάει όλους τους πόρους της εφαρμογής. Κάθε φορά που εισάγεται ένα αρχείο οπουδήποτε στον κατάλογο res το plug in παρατηρεί την αλλαγή και προσθέτει τα κατάλληλα αναγνωριστικά IDs των πόρων στο αρχείο R.java οπότε το αρχείο μένει πάντα συγχρονισμένο.

4.9 Κατάλογος bin

Στον κατάλογο bin βρίσκονται τα αρχεία που παράγονται μετά τη μεταγλώττιση της εφαρμογής. Εκεί βρίσκεται για παράδειγμα το πακέτο ark της εφαρμογής το οποίο μπορεί να χρησιμοποιηθεί κατά την εγκατάσταση. Ωστόσο για να μπορεί να διανεμηθεί το παραπάνω πακέτο μέσω της υπηρεσίας Google Play, θα πρέπει πρώτα να υπογραφεί ψηφιακά. Το Android SDK παρέχει ένα εργαλείο που βοηθά τον προγραμματιστή στη διαδικασία αυτή. Ακόμη πιο εύκολα, ένα πακέτο μπορεί να υπογραφεί ψηφιακά μέσω ενός εύχρηστου γραφικού περιβάλλοντος επιλέγοντας την εξαγωγή της εφαρμογής από το Eclipse IDE.

ΕΠΙΛΟΓΟΣ

Σε αυτό το κεφάλαιο παρουσιάσαμε την δομή μιας εφαρμογής Android στο Eclipse. Περιγράψαμε τους πιο σημαντικούς καταλόγους που έχει οι οποίοι συμβάλουν την δημιουργία του αρχείου .apk που αποτελεί ουσιαστικά την εφαρμογή και χρησιμοποιείται κατά τη διαδικασία της εγκατάστασης.

ΚΕΦΑΛΑΙΟ 5 Ανάλυση ThessVres

ΕΙΣΑΓΩΓΗ

Στα πλαίσια της παρούσας πτυχιακής εργασίας δημιουργήθηκε μια εφαρμογή Android με την ονομασία "ThessVres". Η εφαρμογή εκμεταλλεύεται το GPS των κινητών συσκευών και εντοπίζει την θέση του χρήστη. Σε συνδυασμό με την σύνδεση στο διαδίκτυο ο κάθε χρήστης μπορεί να έχει, στην Android συσκευή του, εγγραφές από μια βάση δεδομένων που έχουμε δημιουργήσει σε server στο internet και ενημερώνεται από εμάς. Στην εφαρμογή υπάρχουν κατηγορίες από σημεία ενδιαφέροντος, από τις οποίες ο χρήστης μπορεί να διαλέξει να και να δει πληροφορίες των εγγραφών στη συσκευή του. Επιπλέον μπορεί να δει την τοποθεσία τους στο χάρτη σε σχέση με την τρέχουσα τοποθεσία του ή την τοποθεσία της επιλογής του που θα ήθελε να κάνει τον "περίπατο" του. Σε συνδυασμό με τους Χάρτες της συσκευής του μπορεί να δει την διαδρομή ανάλογα με το μέσο που διαθέτει (αμάξι, λεωφορείο ή με τα ποδιά.).

5.1 Βιβλιοθήκες

Για την υλοποίηση του ThessVres χρησιμοποιήσαμε δύο εξωτερικές βιβλιοθήκες. Την `google-play-services_lib` και την `android-support-v7-appcompat`. Οι οποίες βρίσκονται στο `sdk/extras`.

5.1.1 Google Play services Library

Η βιβλιοθήκη Google Play service είναι μια επέκταση της πλατφόρμας Android. Παρέχει πρόσβαση στις τελευταίες και καλύτερες υπηρεσίες της Google στην πλατφόρμα Android.

Google Play Services είναι μια πλατφόρμα ανάπτυξης για την ενσωμάτωση των υπηρεσιών της Google στο Android apps. Περιλαμβάνει βασικά μια βιβλιοθήκη πελάτη και ένα APK. Η βιβλιοθήκη πελάτη περιέχει Android APIs για τις επιμέρους υπηρεσίες της Google. Υπηρεσίες της Google δεν είναι το μέρος της πλατφόρμας Android, αλλά είναι τμήμα της πλατφόρμας Google Play Services.[14]

Χρησιμοποιούμε την βιβλιοθήκη Google Play services στην εφαρμογή μας για την παροχή υπηρεσιών όπως την εμφάνιση χάρτη, την εύρεση της τρέχουσα τοποθεσίας του χρήστη, την εμφάνιση τοποθεσιών των σημείων ενδιαφέροντος και την απόσταση αυτόν από τον χρήστη που χρησιμοποιεί την εφαρμογή.

5.1.2 Android Support Library

Το Android Support Library είναι ένα πακέτο το οποίο περιλαμβάνει στατικές "βιβλιοθήκες υποστήριξης", που μπορούν να προστεθούν σε οποιαδήποτε εφαρμογή Android. Με αυτόν τον τρόπο δίνεται η δυνατότητα να χρησιμοποιηθούν APIs που δεν είναι διαθέσιμα για παλαιότερες εκδόσεις πλατφορμών Android ή η δυνατότητα να χρησιμοποιηθούν οφέλη από APIs που δεν αποτελούν μέρος του API που χρησιμοποιείται. Στόχος είναι να απλοποιηθεί η ανάπτυξη της Android εφαρμογής, προσφέροντας περισσότερα APIs που μπορούν να συνδυαστούν με αυτή, χωρίς να συντρέχει λόγος ανησυχίας για την εκάστοτε έκδοση της πλατφόρμας.

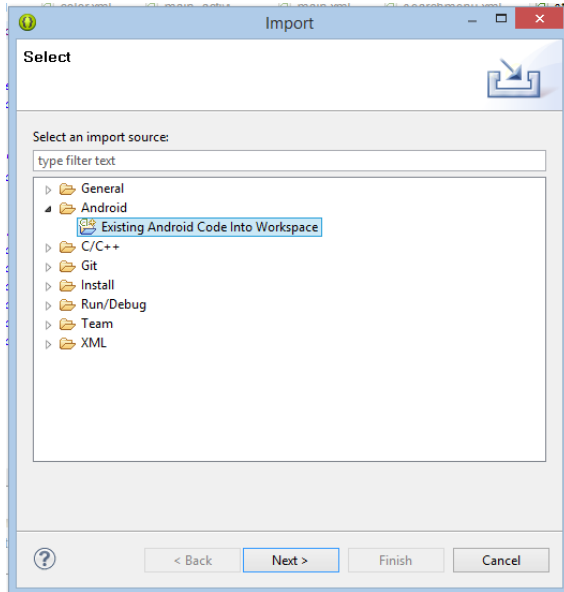
Η Βιβλιοθήκη Υποστήριξης του Android παρέχει ένα αρχείο jar με μια βιβλιοθήκη και ένα API που επιτρέπουν να χρησιμοποιηθούν μερικά από τα πιο πρόσφατα Android APIs από την εφαρμογή, παρόλο που η εφαρμογή τρέχει σε παλαιότερες εκδόσεις του Android.

Στην περίπτωση μας το android-support-v7-appcompat μας προσφέρει την δυνατότητα να έχουμε αντικείμενα που δεν υποστηρίζονται σε παλιότερες εκδόσεις όπως για παράδειγμα τα Action bar τα οποία υποστηρίζονται από το API 11 και μετά. Έτσι με την βοήθεια αυτής της βιβλιοθήκης για κάθε activity που θέλουμε να έχει action bar κάνουμε extent το ActionBarActivity και με την εντολή

```
ActionBar actionBar = getSupportActionBar();
```

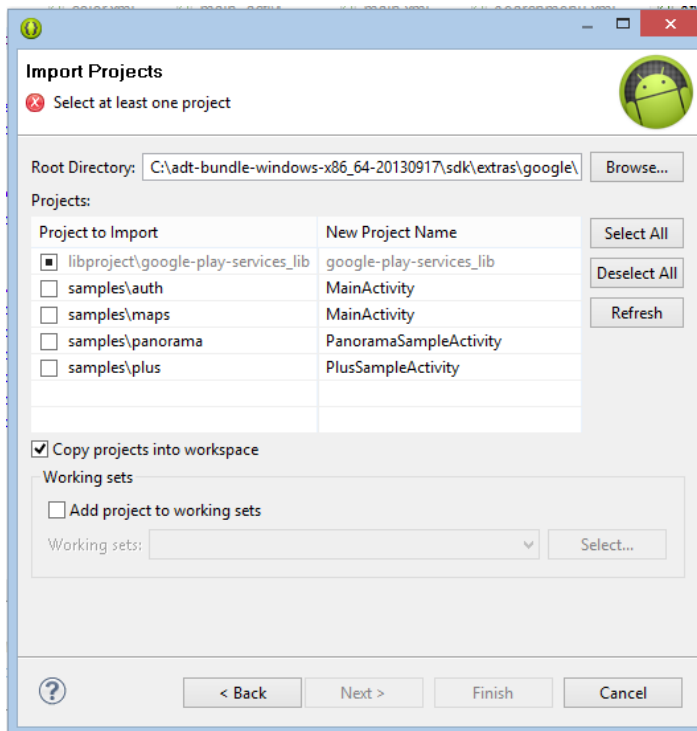
δημιουργούμε το action bar.

Για να περάσουμε μία βιβλιοθήκη στην εφαρμογή μας κάνουμε την βιβλιοθήκη import στο workspace μας και μετά την κάνουμε Linked με το project μας. Για να γίνει αυτό πάμε File > Import και Android>Existing Android Code Into Workspace



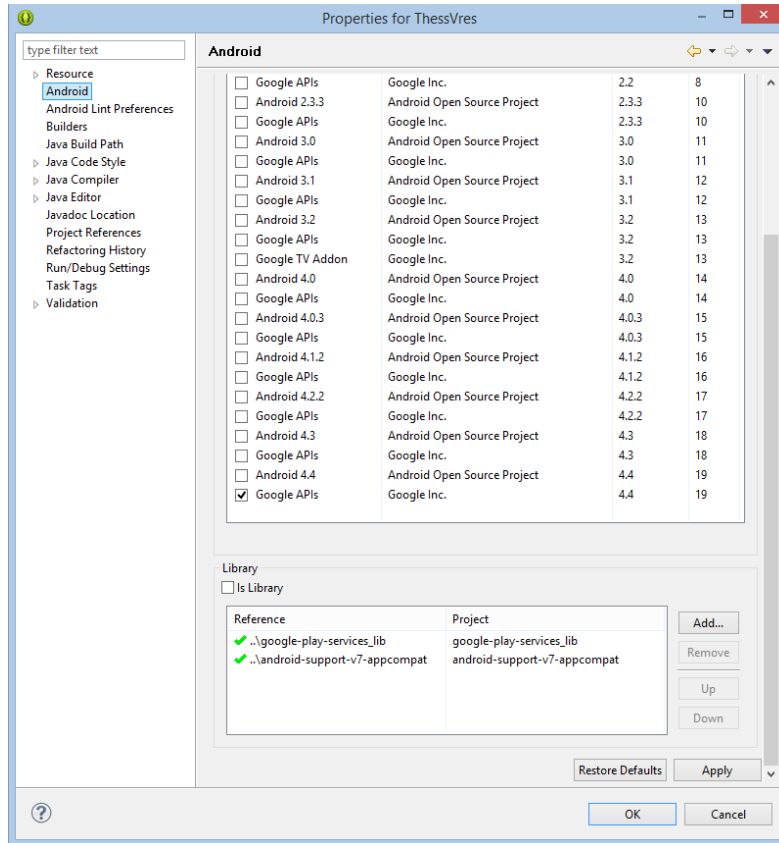
Εικόνα 34. Εισαγωγή πακέτου στο Workspace

Στη συνέχεια επιλέγουμε τον κατάλογο που είναι η βιβλιοθήκη. Για την περίπτωση του Google play service είναι <κατάλογος που είναι το SDK>\extras\google\. Εμείς χρησιμοποιήσαμε την google_play_services_froyo. Προσέχουμε να έχουμε επιλεγμένο το copy project into workspace (σε μερικές περιπτώσεις όταν δεν είναι στο workspace εμφανίζει πρόβλημα). Αφού το κάνουμε αυτό πατάμε “Finish”.



Εικόνα 35. Επιλογή Βιβλιοθήκης

Έτσι θα δούμε την βιβλιοθήκη στο Package Explorer. Επόμενο βήμα είναι να κάνουμε την βιβλιοθήκη linked στο project. Αυτό γίνεται αν πάμε στις ρυθμίσεις του project και επιλέγουμε android, Στο κάτω μέρος επιλέγουμε τις βιβλιοθήκες που θέλουμε και μετά “OK”.



Εικόνα 36. Εισαγωγή της βιβλιοθήκης

5.2 Ανάπτυξη ThessVres

Το ThessVres είναι μια εφαρμογή που βασίζεται στην τοποθεσία του χρήστη και στην εμφάνιση των σημείων ενδιαφέροντος ανάλογα με αυτήν. Αρχικά για να ενημερώσουμε την τοπική βάση της εφαρμογής και πρέπει να ελέγξουμε αν υπάρχει σύνδεση στο internet. Επίσης πρέπει να ελέγξουμε αν το GPS είναι ανοιχτό για να πάρουμε την τρέχουσα τοποθεσία του χρήστη. Ο έλεγχος του GPS και του Internet γίνεται με το service GPSTracker.

```
try {  
    locationManager = (LocationManager) mContext  
        .getSystemService(LOCATION_SERVICE);  
    // getting GPS status  
    isGPSEnabled = locationManager  
        .isProviderEnabled(LocationManager.GPS_PROVIDER);
```

```

Log.e("GPSTracker", "GPSEnabled : " + isGPSEnabled);
// getting network status
isNetworkEnabled = locationManager
.isProviderEnabled(LocationManager.NETWORK_PROVIDER);
Log.e("GPSTracker", "NetworkEnabled : " + isNetworkEnabled);
if (!isGPSEnabled && !isNetworkEnabled) {
// no network provider is enabled
    Log.e("GPSTracker", "no network provider is enabled ");
} else {
this.canGetLocation = true;
if (isNetworkEnabled) {
// Log.e("GPSTracker","NETWORK!!!");
    locationManager.requestLocationUpdates(
        LocationManager.NETWORK_PROVIDER,
        MIN_TIME_BW_UPDATES,
        MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
    Log.e("GPSTracker", "NETWORK!!!");
    if (locationManager != null) {
        location = locationManager
            .getLastKnownLocation(LocationManager.NETWOR
                K_PROVIDER);
        if (location != null) {
            latitude = location.getLatitude();
            longitude = location.getLongitude();
        }
    }
}
}
// if GPS Enabled get lat/long using GPS Services

if (isGPSEnabled) {
    if (location == null) {
        locationManager.requestLocationUpdates(
            LocationManager.GPS_PROVIDER,
            MIN_TIME_BW_UPDATES,
            MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
        Log.e("GPSTracker", "GPS!!!");
        if (locationManager != null) {
            location = locationManager
                .getLastKnownLocation(LocationManager.
                    GPS_PROVIDER);
            if (location != null) {
                latitude = location.getLatitude();
                longitude = location.getLongitude();
            }
        }
    }
}
}
}
} catch (Exception e) {
    e.printStackTrace();
}

```


}

Αφού έχουμε εξασφαλίσει ότι GPS και INTERNET είναι ενεργά, κάνουμε ένα ερώτημα στη βάση μας στο server για να ενημερώσουμε την τοπική βάση SQLite. Αυτό γίνεται με τη βοήθεια ενός ρηρ αρχείου το οποίο κάνει την σύνδεση με την βάση και εκτελεί το ερώτημα που επιστρέφει όλες τις εγγραφές σε μορφή JSON. Με τις κατάλληλες μεθόδους, και συγκεκριμένα με την JSONParser, παίρνουμε τα δεδομένα και τα αποθηκεύουμε στην τοπική βάση SQLite στην συσκευή του κάθε χρήστη που βρίσκεται στον κατάλογο "/data/data/AppName/database/DbName". Σε αυτή την βάση έχουμε πρόσβαση με την κλάση DataBaseHelper οπότε εκεί έχουμε όλα τα ερωτήματα που θα θέλαμε να τρέξουμε στην εφαρμογή μας.

DataBaseHelper.class

```
public class DataBaseHelper extends SQLiteOpenHelper {
    // The Android's default system path of your application database.
    private static String DB_PATH =
        "/data/data/com.atei.thessvres/databases/";
    private static String DB_NAME = "mydb";
    private SQLiteDatabase myDataBase;
    private final Context myContext;
    /**
     * Constructor Takes and keeps a reference of the passed context in order
     * to
     * access to the application assets and resources.
     *
     * @param context
     */
    public DataBaseHelper(Context context) {
        super(context, DB_NAME, null, 1);
        this.myContext = context;
    }
    /**
     * Creates a empty database on the system and rewrites it with your own
     * database.
     */
    public void createDataBase() throws IOException {
        boolean dbExist = checkDataBase();
        if (dbExist) {
            // do nothing - database already exist
        } else {
            // By calling this method and empty database will be created
            //Into the default system path
            // of your application so we are gonna be able to overwrite
            //That database with our database.
            this.getReadableDatabase();
        }
    }
}
```


```
        try {
            copyDataBase();
        } catch (IOException e) {
            throw new Error("Error copying database");
        }
    }
}
/**
 * Check if the database already exist to avoid re-copying the file each
 * time you open the application.
 *
 * @return true if it exists, false if it doesn't
 */
private boolean checkDataBase() {
    SQLiteDatabase checkDB = null;
    try {
        String myPath = DB_PATH + DB_NAME;
        checkDB = SQLiteDatabase.openDatabase(myPath, null,
            SQLiteDatabase.OPEN_READONLY);
    } catch (SQLException e) {
        // database doesn't exist yet.
    }
    if (checkDB != null) {
        checkDB.close();
    }
    return checkDB != null ? true : false;
}
/**
 * Copies your database from your local assets-folder to the just created
 * empty database in the system folder, from where it can be accessed and
 * handled. This is done by transferring bytestream.
 */
private void copyDataBase() throws IOException {
    // Open your local db as the input stream
    InputStream myInput = myContext.getAssets().open(DB_NAME);
    // Path to the just created empty db
    String outFileName = DB_PATH + DB_NAME;
    // Open the empty db as the output stream
    OutputStream myOutput = new FileOutputStream(outFileName);
    // transfer bytes from the inputfile to the outputfile
    byte[] buffer = new byte[1024];
    int length;
    while ((length = myInput.read(buffer)) > 0) {
        myOutput.write(buffer, 0, length);
    }
    // Close the streams
    myOutput.flush();
    myOutput.close();
    myInput.close();
}
```

```
public void openDataBase() throws SQLException {
    // Open the database
    String myPath = DB_PATH + DB_NAME;
    myDataBase = SQLiteDatabase.openDatabase(myPath, null,
        SQLiteDatabase.OPEN_READONLY);
}
@Override
public synchronized void close() {
    if (myDataBase != null)
        myDataBase.close();
    super.close();
}
@Override
public void onCreate(SQLiteDatabase db) {
}
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int
    newVersion) {
}
public Cursor selectRecordBycategory(String categ) {
    Cursor c = myDataBase.rawQuery("Select * from Places where
        Category="
        + categ + "'", null);
    Log.v("test", c.toString());
    return c;
}
public Cursor selectRecordByName(String name) {
    Cursor c = myDataBase.rawQuery("Select * from Places where
        Name LIKE
        "
        + name + "'", null);
    Log.v("test", c.toString());
    return c;
}
public Cursor selectAllRecord() {
    Cursor c = myDataBase.rawQuery("Select * from Places", null);
    Log.v("test", c.toString());
    return c;
}
public Cursor selectChbycatRecord(String cat) {
    Cursor c = myDataBase.rawQuery(
        "Select DISTINCT Ch from Places where Category=" + cat
        + "'",
        null);
    Log.v("test", c.toString());
    return c;
}
public Cursor selectRecordByCh(String cat, String Ch) {
    Cursor c = myDataBase.rawQuery("Select * from Places where
        Category="
```

```

+ cat + "" and Ch="" + Ch + "", null);
Log.v("test", "" + c.getCount());
return c;
}
// Add your public helper methods to access and get content from the
// database.
// You could return cursors by doing "return myDataBase.query(...)" so
it'd
// be easy
// to you to create adapters for your views.
}

```

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	ID	int(11)			No		auto_increment	     
<input type="checkbox"/>	Name	text	utf8_unicode_ci		Yes	NULL		     
<input type="checkbox"/>	Category	text	utf8_unicode_ci		Yes	NULL		     
<input type="checkbox"/>	Description	text	utf8_unicode_ci		Yes	NULL		     
<input type="checkbox"/>	Latitude	text	utf8_unicode_ci		Yes	NULL		     
<input type="checkbox"/>	Longitude	text	utf8_unicode_ci		Yes	NULL		     
<input type="checkbox"/>	Tell	text	utf8_unicode_ci		Yes	NULL		     
<input type="checkbox"/>	Link	text	utf8_unicode_ci		Yes	NULL		     
<input type="checkbox"/>	email	text	utf8_unicode_ci		Yes	NULL		     
<input type="checkbox"/>	menu	text	utf8_unicode_ci		No			     
<input type="checkbox"/>	Ch	text	utf8_unicode_ci		No			     

Εικόνα 37. Βάση του ThessVres (Πίνακας PlaceModel)

Κάθε φορά που ξεκινάει η εφαρμογή ενηρώνεται η βάση. Οπότε τα ερωτήματα στη βάση θα τρέχουν τοπικά και όχι στο server. Παράλληλα με αυτή τη διαδικασία ενημερώνουμε την static μεταβλητή `Location location1 = new Location("")` για να έχουμε και την τρέχουσα θέση του χρήστη στο χάρτη.

Η κάθε οθόνη διαθέτει ActionBar με τη χρήση του `android-support-v7-appcompat` το οποίο επιτρέπει σε συσκευές που διαθέτουν παλιότερες εκδόσεις λειτουργικού Android που δεν υποστηρίζουν τη λειτουργία ActionBar να υποστηρίξουν αυτή την λειτουργία. Στο ActionBar έχουμε επιλογές όπως εμφάνιση στο χάρτη, αναζήτηση και την δυνατότητα επιστροφής σε προηγούμενη οθόνη.



Εικόνα 38. Αρχικό ActionBar της Εφαρμογής

Για την δημιουργία του Action Bar χρειάστηκε να φτιάξουμε ένα xml αρχείο στο AppName >res >menu.

Το αρχείο mainActionBar.xml:

```
<menu>
  <item android:id="@+id/search"
        android:icon="@drawable/ic_action_search"
        android:title="@string/search_title"
        app:showAsAction="always"/>
  <item
        android:id="@+id/action_map"
        android:icon="@drawable/ic_action_map"
        android:title="@string/action_map"
        app:showAsAction="always"/>
</menu>
```

Αφού δημιουργήσαμε το αρχείο με τις επιλογές του ActionBar επόμενο βήμα είναι η εντολή ActionBar actionBar = getSupportActionBar(); στο Activity που θέλουμε να εμφανίζεται (στην περίπτωση μας στην αρχική οθόνη). Στη συνέχεια ορίζουμε ποιο menu θα εμφανίζεται για ActionBar, στη μέθοδο onCreateOptionsMenu(Menu menu).

@Override

```
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.searchmenu, menu);
    return super.onCreateOptionsMenu(menu);
}
```

Με αυτό τον τρόπο καταφέραμε σαν πρώτο βήμα την εμφάνιση του ActionBar. Για τις λειτουργίες του θα πρέπει να υπερβούμε άλλη μια μέθοδο, την onOptionsItemSelected(Menuitem item).

@Override

```
public boolean onOptionsItemSelected(Menuitem item) {
    // Handle presses on the action bar items
    switch (item.getItemId()) {
        case R.id.search:
            Intent i = new Intent(getApplicationContext(),
                SearchActivity.class);
            startActivity(i);
            return true;
        case R.id.action_map:
            // openSettings();
            Intent i1 = new Intent(getApplicationContext(),
                MapViewActivity.class);
```

```
String mapmode = "mapviewmode";
i1.putExtra("mapflag", mapmode);
startActivity(i1);
return true;
default:
return super.onOptionsItemSelected(item);
}
}
```

Έτσι έχουμε καταφέρει να δώσουμε στα “κουμπιά” μας λειτουργίες.

MainActivity

Σε αυτό το Activity εμφανίζουμε τις κατηγορίες που ο χρήστης μπορεί να επιλέξει για να δει τις εγγραφές με τις τοποθεσίες. Οι κατηγορίες αυτές είναι: Εκκλησίες, Μουσεία, Αξιοθέατα, Εστιατόρια, Κέντρα Διασκέδασης, Νοσοκομεία, Αστυνομικά τμήματα, Events και Προξενεία. Για τη εμφάνιση (GUI) του Activity φτιάχνουμε ένα άλλο αρχείο xml στο κατάλογο res>layout.

activity_main.xml:

```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/thess"
    tools:context=".MainActivity" >
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:orientation="vertical" >
        <LinearLayout
            android:id="@+id/l1"
            android:layout_width="fill_parent"
            android:layout_height="0dp"
            android:layout_weight="0.33"
            android:gravity="center"
            android:orientation="horizontal"
            android:padding="5dp" >

        <LinearLayout
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="0.33"
            android:gravity="center_horizontal" >
            <ImageView
                android:id="@+id/Ekklhsies"
                android:layout_width="90dp"
```

```
        android:layout_height="90dp"
        android:src="@string/chbtn" />
</LinearLayout>

<LinearLayout
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="0.33"
    android:gravity="center_horizontal" >
    <ImageView
        android:id="@+id/Mouseio"
        android:layout_width="90dp"
        android:layout_height="90dp"
        android:src="@string/musiabtn" />
</LinearLayout>

<LinearLayout
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="0.33"
    android:gravity="center_horizontal" >
    <ImageView
        android:id="@+id/Aksioth"
        android:layout_width="90dp"
        android:layout_height="90dp"
        android:src="@string/arbbtn" />
</LinearLayout>
</LinearLayout>

<LinearLayout
    android:id="@+id/l2"
    android:layout_width="fill_parent"
    android:layout_height="0dp"
    android:layout_weight="0.33"
    android:gravity="center"
    android:orientation="horizontal"
    android:padding="5dp" >
    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="0.33"
        android:gravity="center_horizontal" >
        <ImageView
            android:id="@+id/Estiator"
            android:layout_width="90dp"
            android:layout_height="90dp"
            android:src="@string/resbtn" />
        </LinearLayout>
    </LinearLayout>
    android:layout_width="0dp"
```



```
        android:layout_height="wrap_content"
        android:layout_weight="0.33"
        android:gravity="center_horizontal" >
        <ImageView
            android:id="@+id/icon5"
            android:layout_width="90dp"
            android:layout_height="90dp"
            android:src="@string/barclup" />
    </LinearLayout>
    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="0.33"
        android:gravity="center_horizontal" >
        <ImageView
            android:id="@+id/hospital_"
            android:layout_width="90dp"
            android:layout_height="90dp"
            android:src="@string/hospitalbtn" />
    </LinearLayout>
</LinearLayout>

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="0dp"
    android:layout_weight="0.33"
    android:gravity="center"
    android:orientation="horizontal"
    android:padding="5dp" >
    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="0.33"
        android:gravity="center_horizontal" >
        <ImageView
            android:id="@+id/police_"
            android:layout_width="90dp"
            android:layout_height="90dp"
            android:src="@string/policebtn" />
    </LinearLayout>
    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="0.33"
        android:gravity="center_horizontal" >
        <ImageView
            android:id="@+id/event_"
            android:layout_width="90dp"
            android:layout_height="90dp"
            android:src="@drawable/event_btn" />
    </LinearLayout>
</LinearLayout>
```



```
        </LinearLayout>
    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="0.33"
        android:gravity="center_horizontal" >
        <ImageView
            android:id="@+id/presvies_"
            android:layout_width="90dp"
            android:layout_height="90dp"
            android:src="@string/presviesbtn" />
    </LinearLayout>
</LinearLayout>
</RelativeLayout>
```

Σαν αποτέλεσμα από το παραπάνω κώδικα θα έχουμε αυτή την οθόνη



Εικόνα 39. Activity Main

Όταν πατήσει ο χρήστης κάποιο από τα κουμπιά αλλάζει εμφάνιση έτσι ώστε να δίνει την αίσθηση ότι πατιέται. Αυτό γίνεται με ένα ακόμα xml στο φάκελο `res>drawable`, στο οποίο ορίζουμε στην ιδιότητα `state_pressed` την εικόνα που θα εμφανίζεται όταν το κουμπί είναι στη κατάσταση `press`

Btn.xml

```
<selector>
```

```
<item android:state_pressed="false" android:drawable="@drawable/btn"/>
<item
android:state_pressed="true"android:drawable="@drawable/btn_pressed"/>
</selector>
```

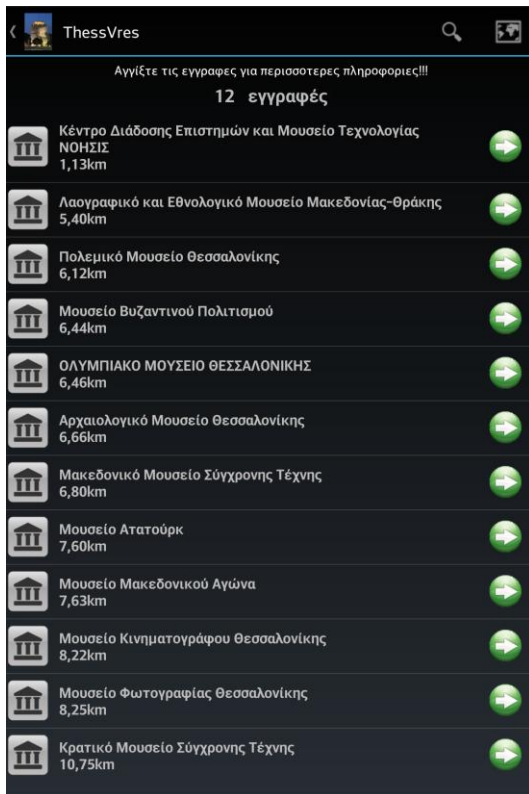
Στην συνέχεια, και αφού δώσαμε στην οθόνη μας την εμφάνιση που θα θέλαμε, δηλώνουμε στην κλάση μας (activity) ότι θα έχει σαν view την συγκεκριμένη οθόνη. Αυτό γίνεται με το παρακάτω κωδικά στην onCreate μέθοδο. setContentView(R.layout.activity_main); Επόμενο βήμα είναι να προσθέσουμε τις λειτουργίες που κάνει το κάθε “κουμπί”. Οπότε υπερβαίνουμε την μέθοδο setOnClickListener με τις λειτουργίες που θέλουμε να δώσουμε.

```
ImageView Btn;
Btn = (ImageView) findViewById(R.id.Btn_1);
Btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String Category = "Μουσείο";
        icon = "" + R.drawable.list_m_icon;
        Intent i = new Intent(getApplicationContext(),
        AndroidListViewActivity.class);
        i.putExtra("MCategory", Category);
        String sh = "2";
        i.putExtra("flag", sh);
        i.putExtra("icon", icon);
        startActivity(i);
    }
});
```

Στον πάρα πάνω κώδικα έχουμε τις εντολές:

```
Intent i = new Intent(getApplicationContext(), AndroidListViewActivity.class);
i.putExtra("MCategory", Category);
startActivity(i);
```

οι οποίες είναι υπεύθυνες για να μας στείλουν στα επόμενα activity και αν θέλουμε να περάσουμε κάποιες μεταβλητές από το ένα στο άλλο. Σε αυτή την περίπτωση το activity που θέλουμε να δούμε είναι το AndroidListViewActivity. Όπου στέλνουμε την κατηγορία που επέλεξε ο χρήστης. Σε αυτό το activity θέλουμε να εμφανίσουμε τις εγγραφές που αντιστοιχούν στην κατηγορία που διάλεξε ο χρήστης. Η μορφή που θέλουμε να έχει η οθόνη του χρήστη είναι η ακόλουθη.



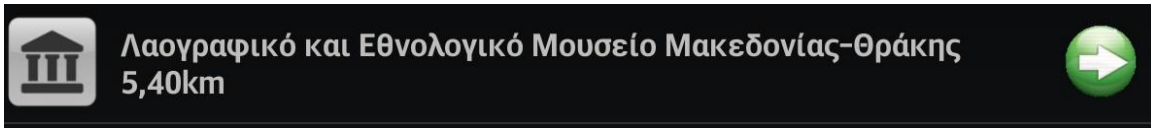
Εικόνα 40. Μουσεία της Θεσσαλονίκης (ListView)



Εικόνα 41. Επέκταση της Κατηγορίας Βυζαντινή (ExpandableListView)

Δημιουργήσαμε δυο είδη λίστας: μια απλή λίστα που περιέχει τις εγγραφές μιας κατηγορίας όπως στην Εικόνα 40 για την οποία χρησιμοποιήσαμε ListView και μια λίστα με ομαδοποιημένες εγγραφές που επεκτείνεται η κάθε ομάδα για την

αποκάλυψη των εγγραφών όπως στην εικόνα 41 για την οποία χρησιμοποιήσαμε `ExpandableListView`.



Εικόνα 42. Μορφή μιας γραμμής σε μια λίστα

Η μορφή που έχει η κάθε γραμμή της λίστας όπως φαίνεται εικόνα 39 φτιάχνουμε ακόμα ένα `xml` το οποίο είναι υπεύθυνο να δώσει μορφή στα πεδία της λίστας μας με την βοήθεια του `adapter` που έχουμε φτιάξει.

```
ListAdapter adapter = new ListAdaptersimple(this, pModel_Alist, icon);  
list = (ListView) findViewById(R.id.listviewls);  
list.setAdapter(adapter); .
```

Οι εγγραφές ταξινομούνται ανάλογα με την απόσταση τους από την τρέχουσα τοποθεσία του χρήστη. Το αντικείμενο `PlaceModel` κάνει `implement` το `interface ObjectToTree`.

```
public interface ObjectToTree {  
  
    public double getCompareItem();  
}
```

Υπερβαίνουμε την μέθοδο `getCompareItem()`; στην κλάση `PlaceModel.java` ώστε να γίνεται η ταξινόμηση με τη μεταβλητή `km` που είναι η απόσταση.

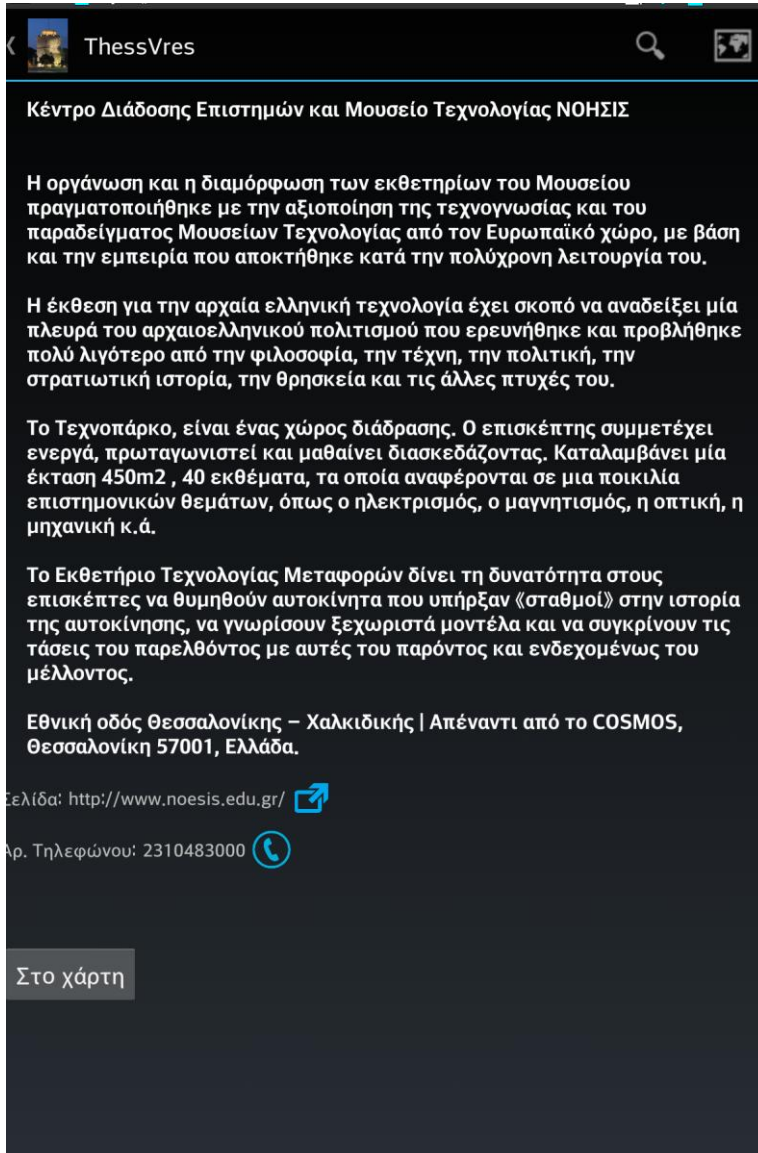
```
@Override  
public double getCompareItem() {  
    return km;  
}
```

Στη συνέχεια δίνουμε στη κάθε εγγραφή της λίστας την ιδιότητα `onclick` ώστε όταν αγγίζει ο χρήστης να μπορεί να δει πληροφορίες.

```
list.setOnItemClickListener(new OnItemClickListener() {  
    public void onItemClick(AdapterView<?> parent, final View view,  
        final int position, long id) {  
        // selected item  
        String place = ((TextView) view.findViewById(R.id.label))  
            .getText().toString();  
        // Launching new Activity on selecting single List Item
```

```
Intent i = new Intent(getApplicationContext(),
SingleListItem.class);
// sending data to new activity
i.putExtra("product", place);
i.putExtra("title", ((PlacesModel) pMtree
.inOrderTraversallList().get(position)).getName());//
i.putExtra("info", ((PlacesModel) pMtree.inOrderTraversallList()
.get(position)).getDescription());
i.putExtra("la", ""+ ((PlacesModel) pMtree.inOrderTraversallList()
.get(position)).getLatitude());
i.putExtra("lo", ""+ ((PlacesModel) pMtree.inOrderTraversallList()
.get(position)).getLongitude());
i.putExtra("tell", ((PlacesModel) pMtree.inOrderTraversallList()
.get(position)).getTell());
i.putExtra("link", ((PlacesModel) pMtree.inOrderTraversallList()
.get(position)).getLink());
i.putExtra("email", ((PlacesModel) pMtree
.inOrderTraversallList().get(position)).getEmail());
i.putExtra("menu", ((PlacesModel) pMtree.inOrderTraversallList()
.get(position)).getmenu());
startActivity(i);
}
});
```

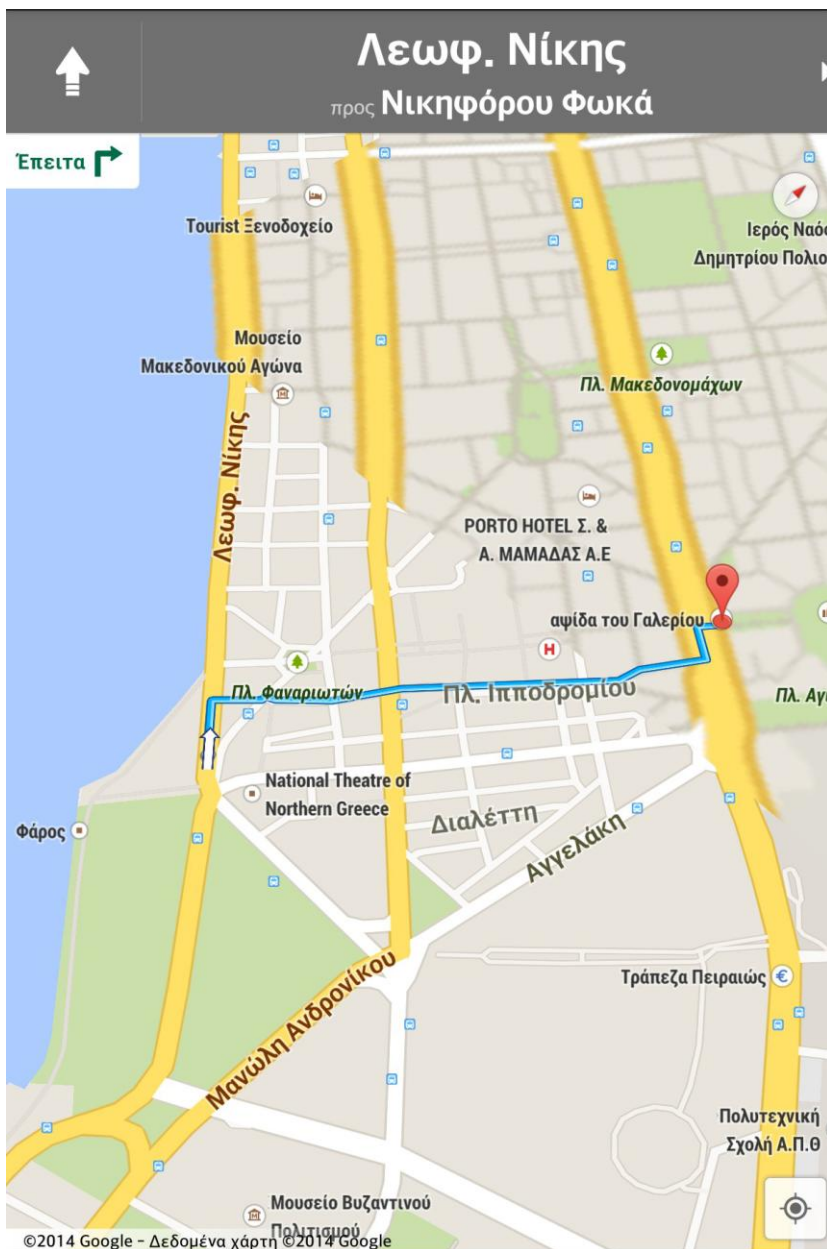
Με τον παραπάνω κώδικα όταν ο χρήστης αγγίξει κάποια εγγραφή ανοίγει το επόμενο αντίστοιχο activity SingleListItem και στέλνουμε σε μεταβλητές τις πληροφορίες για να τις εμφανίσει. Το SingleListItem έχει την μορφή που φαίνεται στην εικόνα 43.



Εικόνα 43. Οι πληροφορίες κάθε αντικειμένου (SingleListItem)

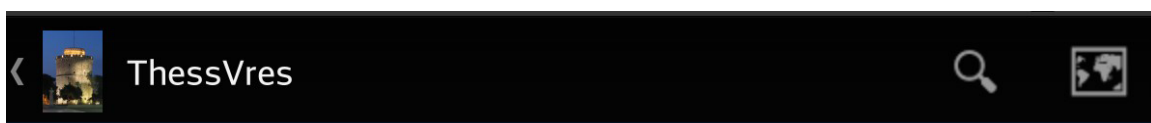
Αν η εγγραφή έχει διαθέσιμο τηλέφωνο, σελίδα στο διαδίκτυο, μενού και e-mail εμφανίζεται η πληροφορία. Αγγίζοντας στο μπλε εικονίδιο δίπλα εκτελείται η αντίστοιχη ενέργεια. Με το κουμπί “Στο χάρτη” θέλουμε να ανοίγουν οι χάρτες του android και να δίνουμε την θέση του χρήστη και την θέση της τοποθεσίας που έχει επιλέξει. Αυτό γίνεται με τον παρακάτω κώδικα.

```
Intent intent = new Intent(android.content.Intent.ACTION_VIEW,Uri.parse("http://maps.google.com/maps?saddr="+AndroidListViewActivity.location1.getLatitude()+","+AndroidListViewActivity.location1.getLongitude()+ "&daddr=" +tolat + "," + tolon));startActivity(intent);
```

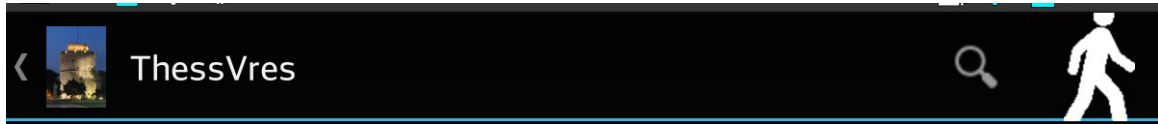



Εικόνα 44. Διαδρομή που εμφανίζεται στον χάρτη

Σε μερικές περιπτώσεις έχουμε διαφορετικό ActionBar ανάλογα με τις απαιτήσεις των Views. Τα διαφορετικά ActionBar που θα έχουμε είναι αυτά.



Εικόνα 45. Το κύριο ActionBar

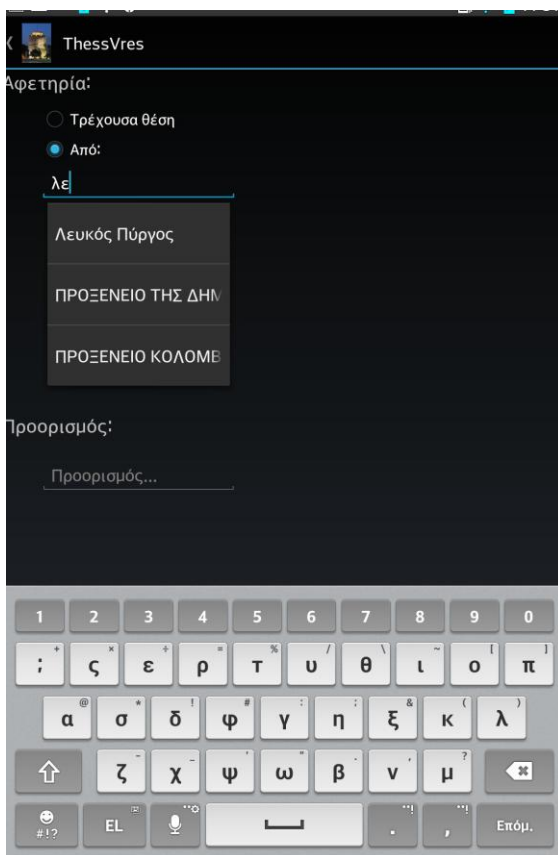


Εικόνα 46. Το Action Bar που εμφανίζεται στον χάρτη μετά από αναζήτηση διαδρομής

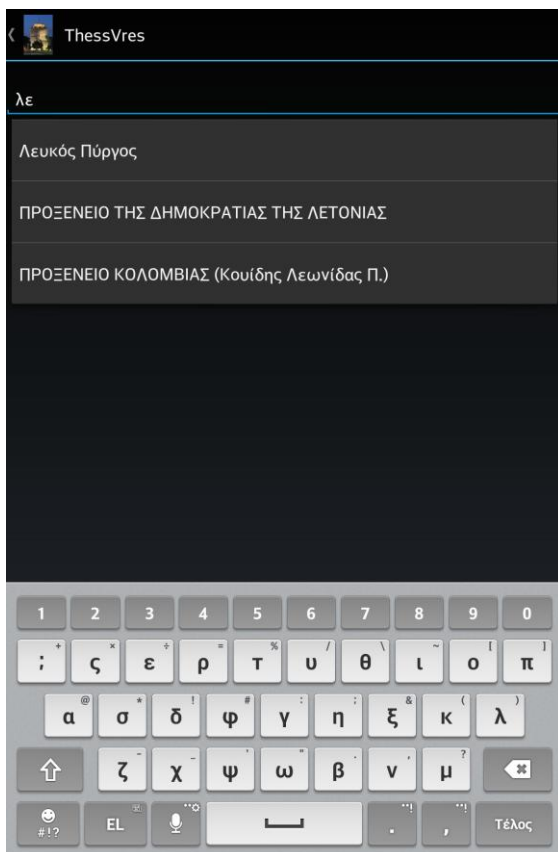


Εικόνα 47. Το Action Bar που εμφανίζεται στον χάρτη

Όταν ο χρήστης επιλέγει το εικονίδιο της αναζήτησης, ανάλογα με το activity, βγαίνει το αντίστοιχο search menu.

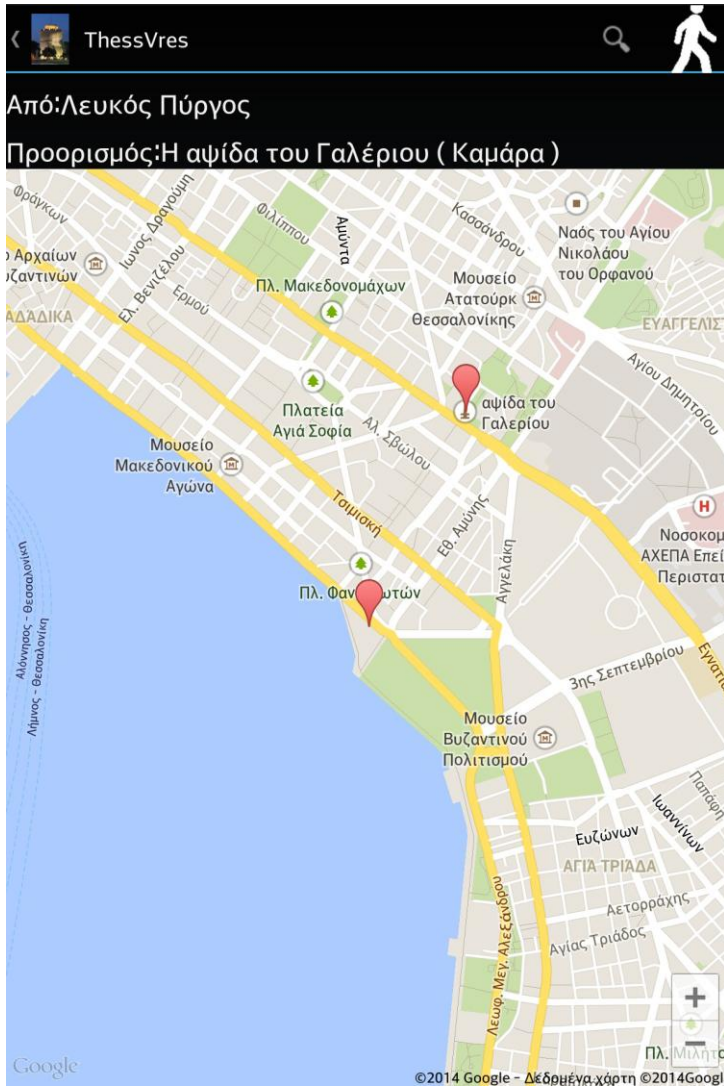


Εικόνα 48. Αναζήτηση Διαδρομής




Εικόνα 49. Απλή αναζήτηση

Για την εικόνα 48 το επόμενο view θα είναι το `ListViewActivity` με τη λίστα η οποία θα περιέχει το αποτέλεσμα της αναζήτησης. Για την εικόνα 49 το επόμενο View θα είναι το `SearchResaultActivity` το οποίο εμφανίζει στο χάρτη τα δύο σημεία που επιλέξαμε.

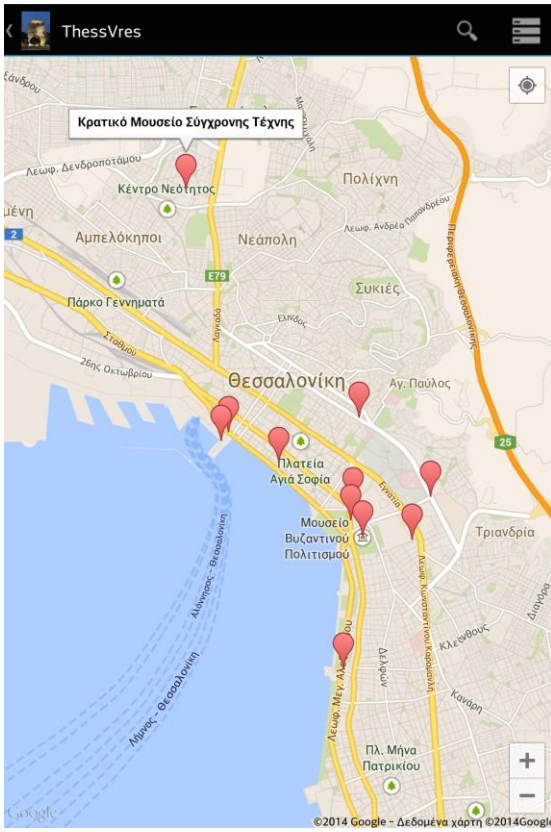


Εικόνα 50. Το αποτέλεσμα μιας αναζήτησης

Πατώντας το εικονίδιο με το ανθρωπάκι ανοίγουν οι χάρτες του android με τα 2 σημεία και ο χρήστης μπορεί να δει την διαδρομή. Στο ActionBar έχουμε και το εικονίδιο 

το οποίο άπλα “σκοτώνει” το Activity και εμφανίζεται το view του προηγούμενου activity. Το εικονίδιο του χάρτη είναι υπεύθυνο για να εμφανιστεί το MapViewActivity το οποίο ανάλογα με κάποια flags, που παίρνουμε από το activity που το καλούμε, εμφανίζει τα σημεία των εγγραφών της λίστας ή το σημείο που επιλέξαμε και βλέπουμε στο SingleListActivity. Τα διαφορετικά views είναι :

Πτυχιακή εργασία των φοιτητών Ζαχαρία Γιακουμή και Ευαγγελία Παζαρλόγλου



Εικόνα 51. Εμφανίζει όλες τις περιοχές που υπάρχουν στο χάρτη



Εικόνα 52. Εμφανίζει ένα σημείο στον χάρτη

Για να έχουμε την δυνατότητα να δούμε τους χάρτες πρέπει να πάρουμε ένα κλειδί από την Google στη περίπτωση μας το Google API V2. Για να το αποκτήσουμε πρέπει να έχουμε ένα λογαριασμό Google από με το οποίο κάνουμε log in στο Google console <https://console.developers.google.com/project>. από εκεί δημιουργούμε το project. Από την καρτέλα Apis επιλέγουμε το Google Maps Android Api V2 και αποδεχόμαστε τους όρους.

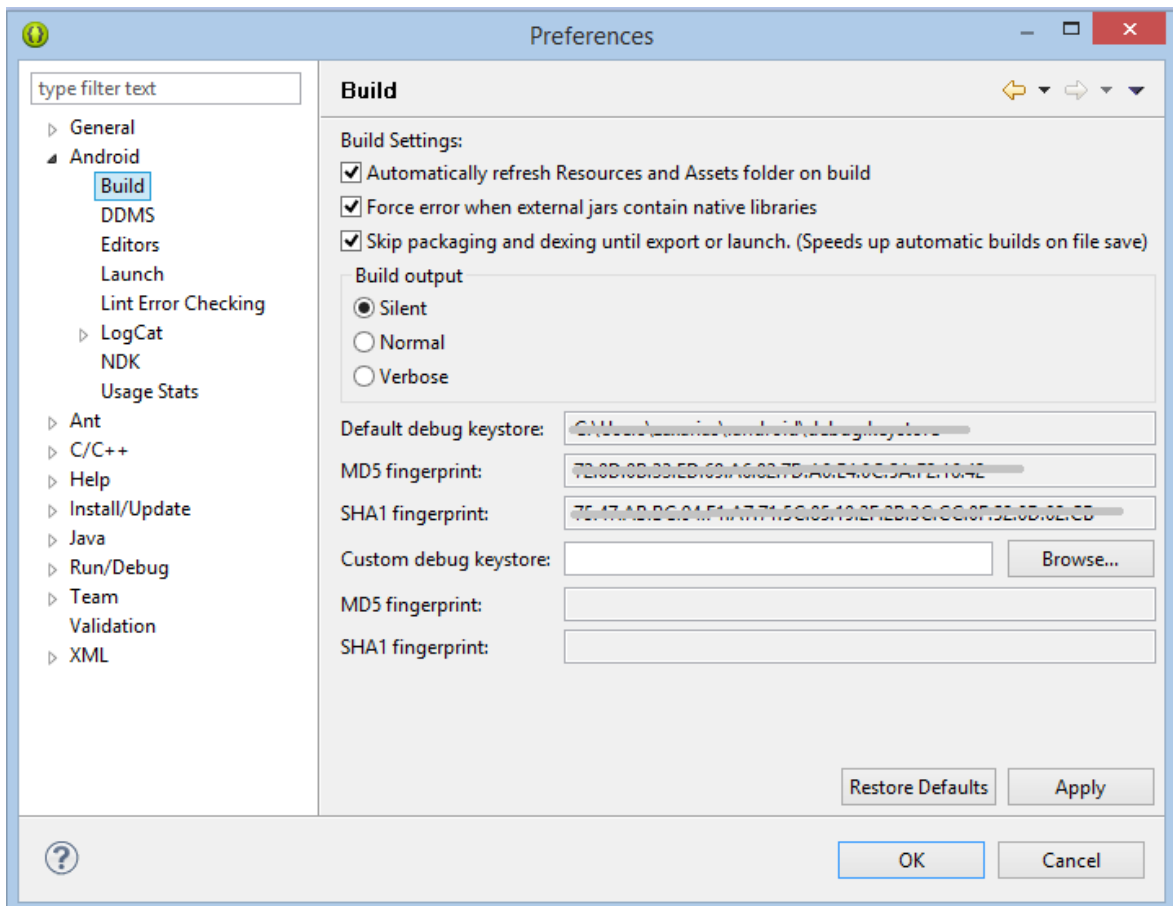


	NAME	QUOTA	STATUS
	Google Maps Android API v2		ON
	Google Maps JavaScript API v3	0%	ON
	Ad Exchange Buyer API	1,000 requests/day	OFF
	Ad Exchange Seller API	10,000 requests/day	OFF
	Admin SDK	150,000 requests/day	OFF
	AdSense Host API	100,000 requests/day	OFF
	AdSense Management API	10,000 requests/day	OFF
	Analytics API	50,000 requests/day	OFF
	Audit API	10,000 requests/day	OFF
	BigQuery API	10,000 requests/day	OFF
	Blockly API v3	10,000 requests/day	OFF

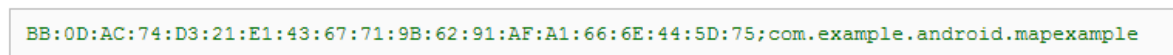
Εικόνα 53. Google console

Για να κάνουμε την σύνδεση με το project μας περνούμε από το eclipse το κλειδί SHA1 fingerprint από το μενού window>preference android-build και προσθέτουμε στο τέλος ένα «;» και στη συνέχεια το όνομα του project. Στη συνέχεια το τοποθετούμε στη καρτέλα Credentials του Google Console στο Create New Key - Android key και μετά Create. Παίρνουμε το αποτέλεσμα και το δηλώνουμε στο xml αρχείο AndroidManifest.xml.

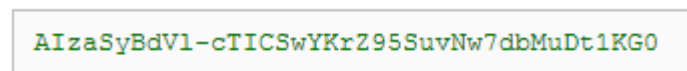
```
<meta-data  
android:name="com.google.android.maps.v2.API_KEY"  
android:value="API_V2_KEY"/>
```



Εικόνα 54. Android Build



Εικόνα 55. Δήλωση της εφαρμογής στο Google Console



Εικόνα 56. Google API key

Μετά από αυτό το βήμα μπορούμε να δούμε τους χάρτες. Για να εμφανίσουμε στο χάρτη ένα Market (στίγμα στο χάρτη) χρειαζόμαστε ένα αντικείμενο Mark και του περνάμε τις παραμέτρους ανάλογα με τις πληροφορίες που θέλουμε να δώσουμε σε αυτό.

```
double plat = location1.getLatitude(); double plon =location1.getLongitude();  
drawMarker(new LatLng(plat, plon),getResources().getString(R.string.clocation));
```

Με τις παρακάτω εντολές μπορούμε να δείξουμε στο χάρτη που να εστιάζει και το είδος του.

```
mMap.moveCamera(CameraUpdateFactory.newLatLng(new LatLng(plat, plon)));  
mMap.animateCamera(CameraUpdateFactory.zoomTo(15), 4000, null);
```

Ένα από τα σημαντικότερα συστατικά για μια εφαρμογή android είναι το αρχείο που αναφέραμε πριν. Το XML manifest. Περιέχει μέσα όλες τις πληροφορίες της εφαρμογής τις απαιτήσεις, σε ποιες εκδόσεις μπορεί να λειτουργήσει όπως και σε ποιες λειτουργίες θέλει να έχει πρόσβαση η εφαρμογή για να εκτελεί τις λειτουργίες της. Το manifest έχει την παρακάτω μορφή.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="AppName"
    android:versionCode="1"
    android:versionName="1.0.0" >
    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="8" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
        android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission
        android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission
        android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
    <uses-permission
        android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <!-- Map api V2 -->
    <permission
        android:name="com.atei.Thessvres.permission.MAPS_RECEIVE"
        android:protectionLevel="signature" />
    <uses-permission
        android:name="com.atei.Thessvres.permission.MAPS_RECEIVE"
    />
    <uses-feature
        android:glEsVersion="0x00020000"
        android:required="true" />
    <application
        android:name="com.atei.Thessvres"
        android:configChanges="keyboardHidden|orientation"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:screenOrientation="portrait"
        android:theme="@android:style/Theme.NoTitleBar" >
    <uses-library android:name="com.google.android.maps" />
    <activity
        android:name="com.atei.Thessvres.SplashActivity"
        android:configChanges="orientation"
        android:label="@string/title_activity_main"
        android:screenOrientation="portrait" >
    <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
```



```
</intent-filter>
</activity>
<!--
<activity
.....
.....
.....
<!-- </activity>
-->
<!--
<activity
.....
.....
.....
<activity>
.....
.....
.....
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.DEFAULT" />
</intent-filter>
</activity>
-->
<service android:name="com.atei.Thessvres.services.LoaderService" />
<receiver android:name="com.atei.Thessvres.notifications.AlarmReceiver"
/>
<meta-data
android:name="com.google.android.maps.v2.API_KEY"
android:value="API_V2_KEY" />
</application>
</manifest>
```

ΕΠΙΛΟΓΟΣ

Σε αυτό το κεφάλαιο παρουσιάσαμε την ανάπτυξη της εφαρμογής. Δείξαμε πως χρησιμοποιήθηκαν οι τεχνολογίες που αναφέραμε στο δεύτερο κεφάλαιο. Περιγράψαμε τις σημαντικότερες λειτουργίες της εφαρμογής. Και τέλος εμφανίσαμε τα κυριότερα σημεία του κώδικα που αναπτύξαμε.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Σύμφωνα με τις ανάγκες που προέκυψαν λόγω της εξέλιξης των τεχνολογιών και του μεγάλου ανταγωνισμού που δημιουργήθηκε, όλο και περισσότερες τουριστικές επιχειρήσεις αναπτύσσουν διαδικτυακές υπηρεσίες και εφαρμογές για Smartphone και χρησιμοποιούν αυτές τις νέες τεχνολογίες.

Τα τελευταία χρόνια το Android έχει καταφέρει να εισβάλλει για τα καλά στην παγκόσμια αγορά. Κυριαρχεί σχεδόν σε όλες τις κατασκευάστριες εταιρείες κινητών τηλεφώνων και όχι μόνο, που το επιλέγουν ως κύριο λειτουργικό σύστημα για τις φορητές συσκευές τους. Υπάρχουν χιλιάδες δωρεάν και επί πληρωμή εφαρμογές διαθέσιμες για όλους τους χρήστες του Android.

Αποφασίσαμε να ασχοληθούμε με αυτό το αντικείμενο και είδαμε πολλές ενδιαφέρουσες τεχνολογίες που σχετίζονται με το android. Το android εξελίσσεται κάθε μέρα και λόγω των συχνών του ανανεώσεων αν κάποιος θέλει να ασχοληθεί σοβαρά με αυτό θα πρέπει να είναι ενημερωμένος για τις νέες εκδόσεις του android. Η συχνή ενασχόληση με το αντικείμενο απαιτείται καθώς για το καλύτερο αποτέλεσμα χρειάζεται μεγάλη εμπειρία.

Με την πτυχιακή λάβαμε μέρος στο διαγωνισμό apps 4 thesaloniki η συμμετοχή μας βρίσκεται σε αυτή την ιστοσελίδα <http://goo.gl/ev9hRL>. από εδώ μπορεί κάποιος να κατεβάσει την εφαρμογή και υπάρχει διαθέσιμο βίντεο. Ο κώδικας της εφαρμογής είναι ανεβασμένος στο github <https://github.com/ssatoo/ThessVres>. Επιπλέον φτιάξαμε μία ιστοσελίδα <http://thessvres.netii.net/> που έχουμε το ark της εφαρμογής και την βάση που μπορούμε να προσθέτουμε δεδομένα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

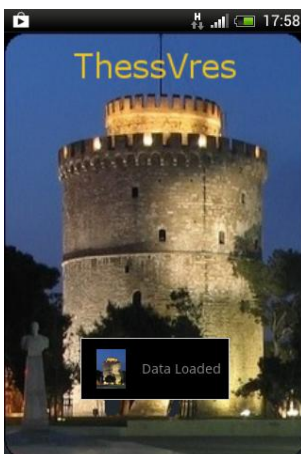
- [1] Deitel P., Deitel H.(2010) *Java προγραμματισμό*, Μ.Γκιούρδας Αθήνα.
- [2] Scott Davis (2006) *Google Maps API, V2: Adding Where to Your Applications*, The Pragmatic Bookshelf, Dallas, Texas
- [3]*Andoid*
<http://el.wikipedia.org/wiki/Android>
- [4]*Android*
<http://ventrix.nsd.c.gr/stuff/AndroidFoscommWorkshop.pdf>
- [5] *Android (operating system)*
http://en.wikipedia.org/wiki/Android_%28operating_system%29
- [6]*Android version history*
http://en.wikipedia.org/wiki/Android_version_history
- [7]*Android/Introduction*
<http://en.wikibooks.org/wiki/Android/Introduction>
- [8]*Architecture of Android*
<http://www.androidaspect.com/2012/10/architecture-of-android.html>
- [9]*Directory Structure of an Android Project*
<http://www.compiletimeerror.com/2013/01/directorystructure-of-android-project.html#.U3FLYPmSxEk>
- [10]*Google Play Services*
<http://developer.android.com/google/play-services/index.html#newfeatures>
- [11]N. Hari Krishna, B.S.S Prasad C (2012) *Android Architecture*
<http://www.slideshare.net/kittu565/android-architecture>
- [12]*SDK Manager*
<http://developer.android.com/tools/help/sdk-manager.html>
- [13]*Τεχνολογία » Τι είναι το Android*
<http://www.adds.gr/company/technology/what-is-android/>
- [14]*Working with Google Play Services Library in Android*
<http://www.androidaspect.com/2014/02/google-play-services.html>

ΟΔΗΓΟΣ ΧΡΗΣΗΣ ThessVres

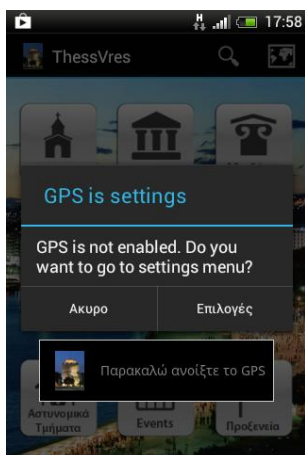
Στην αρχή της εφαρμογής εμφανίζεται το splash screen και παράλληλα ελέγχεται αν είναι ανοιχτά το δίκτυο και το GPS. Το δίκτυο πρέπει να είναι ανοιχτό για να μπορέσουν να κατέβουν τα δεδομένα από το server. Από την άλλη το GPS πρέπει να είναι ανοιχτό για να μπορέσει η εφαρμογή να εντοπίσει την τρέχουσα τοποθεσία του χρήστη στην Θεσσαλονίκη για ευκολότερη πλοήγηση. Σε περίπτωση που δεν είναι κανένα από τα δυο ανοιχτά αναγράφεται μήνυμα και εμφανίζεται επιλογή για να ανοίξουν οι ρυθμίσεις.



Εικόνα 57. Η αρχική εικόνα (Splash Screen) χωρίς δίκτυο να είναι ανοιχτό



Εικόνα 58. Η αρχική εικόνα (Splash Screen) με το δίκτυο ανοιχτό



Εικόνα 59. Όταν το GPS είναι ανενεργό ζητείται η ενεργοποίηση του

Εφόσον εξασφαλίσουμε ότι το GPS και το δίκτυο είναι ανοιχτά και τα αποθηκεύσαμε τα δεδομένα στη μνήμη του κινητού και τη τρέχουσα τοποθεσία του χρήστη. Ύστερα εμφανίζεται η κύρια οθόνη όπου υπάρχουν οι κατηγορίες που είναι ταξινομημένες οι εγγραφές.

Σε αυτό το σημείο ο χρήστης επιλέγει την κατηγορία που θέλει. Οι κατηγορίες που μπορεί να επιλέξει ο χρήστης είναι : Εκκλησίες, Μουσεία, Αξιοθέατα, Εστιατόρια, Κέντρα διασκέδασης, Νοσοκομεία, Αστυνομικά Τμήματα, Events και Προξενεία.



Εικόνα 60. Η κύρια οθόνη της εφαρμογής

Όταν ο χρήστης επιλέξει μια από τις παραπάνω κατηγορίες τότε εμφανίζεται η οθόνη με τις εγγραφές της κατηγορίας που επέλεξε. Οι εγγραφές

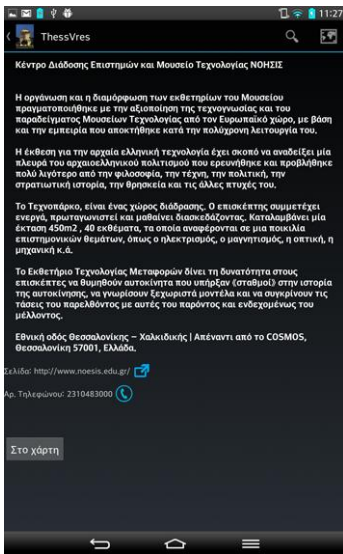
είναι ταξινομημένες ανάλογα με την απόσταση που έχει ο χρήστης από αυτές. Από τη λίστα ο χρήστης μπορεί να επιλέξει την εγγραφή που θέλει ώστε να δει τις πληροφορίες. Οι πληροφορίες αυτές είναι η περιγραφή της εγγραφής, ιστοσελίδα όπου ο χρήστης μπορεί να την μεταβεί σε αυτήν, αριθμός τηλεφώνου με την δυνατότητα κλήσης σε αυτόν, αποστολή email και μενού για τα εστιατόρια.



Εικόνα 61. Λίστα εγγραφών (μιας από τις αρχικές 9)

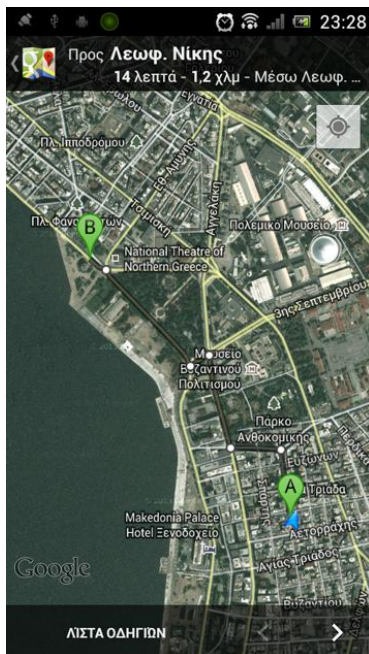


Εικόνα 62. Λίστα εγγραφών ομαδοποιημένες σε κατηγορίες

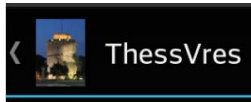


Εικόνα 63. Πληροφορίες μιας εγγραφής

Στην οθόνη με τις πληροφορίες υπάρχει το κουμπί "Στο χάρτη" που παραπέμπει τον χρήστη στην εφαρμογή Google maps όπου εμφανίζεται διαδρομή με αφετηρία την τρέχουσα τοποθεσία και προορισμό την τοποθεσία της εγγραφής.



Εικόνα 64. Μια διαδρομή στο Google Maps



Όταν το βελάκι είναι ενεργοποιημένο τότε ο χρήστης μπορεί να πάει στην προηγούμενη οθόνη.



Σε περίπτωση που ο χρήστης πατήσει το εικονίδιο εμφανίζεται ο χάρτης ανάλογα με την οθόνη που βρίσκεται:

- Στην αρχική οθόνη εμφανίζεται στον χάρτη η τρέχουσα τοποθεσία του χρήστη.
- Στην οθόνη με τις λίστες στον χάρτη εμφανίζονται όλες οι τοποθεσίες των εγγραφών.
- Στην οθόνη με τις πληροφορίες μιας εγγραφής εμφανίζεται η τρέχουσα τοποθεσία του χρήστη και η τοποθεσία της εγγραφής.

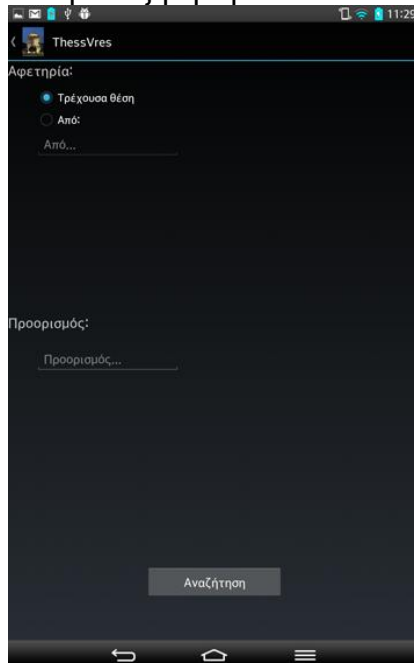


Το συγκεκριμένο εικονίδιο εμφανίζεται στην περίπτωση που η εφαρμογή είναι στην οθόνη του χάρτη και επιστρέφει στην προηγούμενη οθόνη.



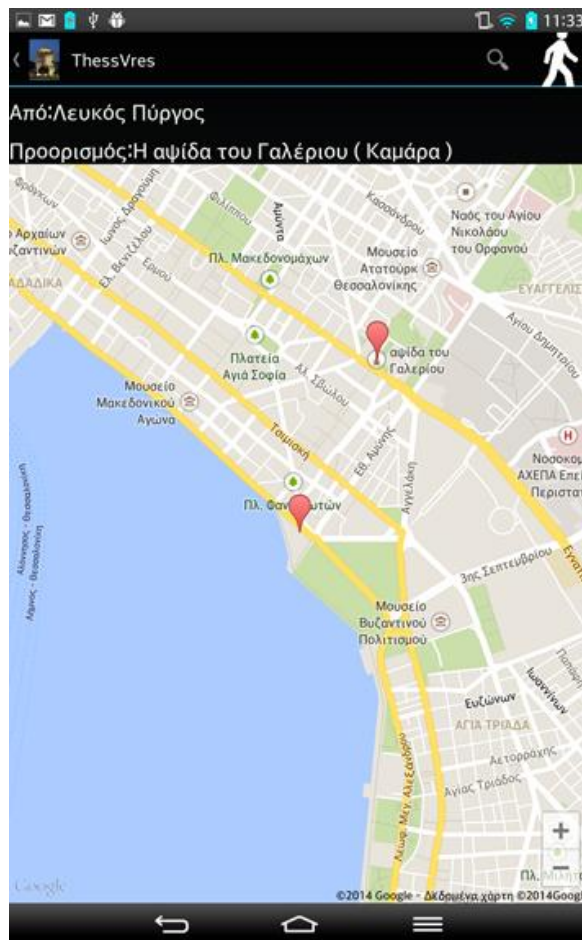
Υπάρχουν 2 τύποι αναζήτησης.

- Σύνθετη αναζήτηση



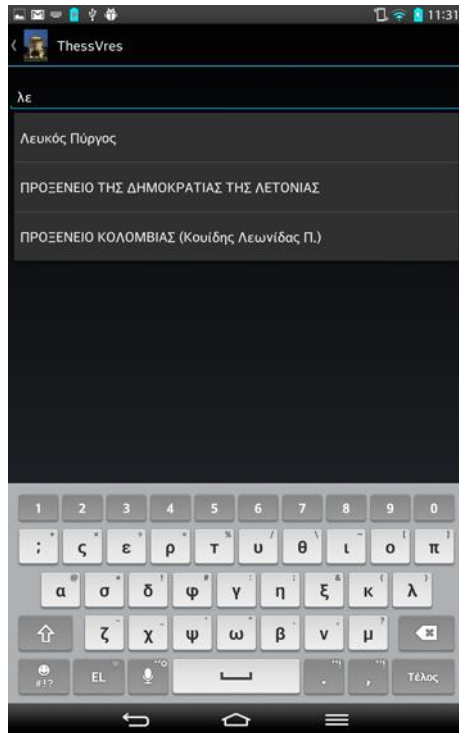
Εικόνα 65. Σύνθετη αναζήτηση

Ο χρήστης γράφει την αφετηρία και τον προορισμό που θέλει (Πρέπει να γράψει κάποιο σημείο ενδιαφέροντος και όχι διεύθυνση). Πατώντας αναζήτηση εμφανίζεται ο χάρτης με την αφετηρία και τον προορισμό που διάλεξε ο χρήστης




Εικόνα 66. Οθόνη με χάρτη μετά από αναζήτηση

- Απλή αναζήτηση



Εικόνα 67. Απλή αναζήτηση

Όταν ο χρήστης βρίσκεται σε μια λίστα και πατήσει το εικονίδιο αναζήτηση τότε εμφανίζεται η απλή αναζήτηση.

Το εικονίδιο  εμφανίζεται στην οθόνη του χάρτη όταν πατηθεί ανοίγει το Google maps και εμφανίζεται η διαδρομή που έχει επιλεγθεί.