

ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



Κατασκευή διαδικτυακής εφαρμογής στην αρχιτεκτονική ANDROID που υλοποιεί ένα παιχνίδι ερωταπαντήσεων

Μαργαρίτη-Δουβίτσα Ελένη
AM 05/2767

Επιβλέποντες καθηγητές:
Δημήτρης Μιχαλόπουλος
Δρ. Δασυγένης Μηνάς



ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία έχει σαν βασικό στόχο την ανάπτυξη μιας διαδικτυακής ηλεκτρονικής εφαρμογής σε περιβάλλον Google Android. Η εφαρμογή επιτρέπει στον κάτοχο κινητού τηλεφώνου ή pc-tablet με εγκατεστημένο το λειτουργικό σύστημα Android να παίξει ένα παιχνίδι ερωτήσεων γενικού περιεχομένου. Αυτό το παιχνίδι αποσκοπεί τόσο στην διασκέδαση όσο και στην απόκτηση γνώσεων μέσω των απαντήσεων που δίνουν οι παίκτες.

Ο τομέας της ανάπτυξης λογισμικού για κινητά τηλέφωνα έχει εξελιχθεί σημαντικά τα τελευταία χρόνια. Το Android έκανε την εμφάνισή του ως μια νέα πλατφόρμα ανάπτυξης εφαρμογών κινητών τηλεφώνων, βασιζόμενο σε επιτυχίες του παρελθόντος και αποφεύγοντας τα λάθη των άλλων πλατφορμών. Σχεδιάστηκε ώστε να δώσει στον προγραμματιστή όλα όσα χρειάζεται, ώστε να δημιουργήσει καινοτόμες εφαρμογές. Ένας λόγος που επέλεξα αυτή την πλατφόρμα για να αναπτύξω την εφαρμογή μου είναι ότι παρέχεται δωρεάν, ενώ οι προγραμματιστές απολαμβάνουν πολλά οφέλη σε σχέση με τις ανταγωνιστικές πλατφόρμες.

Συνοπτικά, τα βασικά συστατικά στοιχεία αυτής της πτυχιακής εργασίας επικεντρώνονται σε τρεις βασικούς τομείς. Ο πρώτος είναι αυτός που αφορά την ανάπτυξη της εφαρμογής σε περιβάλλον Android με την βοήθεια του Eclipse IDE (Client). Επέλεξα το Eclipse γιατί η ομάδα ανάπτυξης του Android έχει ενσωματώσει πολλά εργαλεία ανάπτυξης στο Eclipse, αλλά μπορούν να χρησιμοποιηθούν και άλλα IDE. Αυτό το κομμάτι «τρέχει στις συσκευές μας». Ο δεύτερος αφορά την βάση δεδομένων (Data Base) που αποθηκεύονται τα δεδομένα μας. Ο τρίτος αφορά το κομμάτι που εκτελείται στον διακομιστή (Server).

Στο επόμενο κεφάλαιο θα αναλύσουμε τα τρία τμήματα, ολόκληρο το περιβάλλον ανάπτυξης λογισμικού Android, τι σημαίνουν τα xml και php αρχεία και πώς δουλεύουν οι υπηρεσίες διαδικτύου.

ABSTRACT

The main goal of this diploma thesis is the development of a web-based application within the Google Android environment. The developed application allows the owner of a mobile smartphone or pc tablet, equipped with the Android operating system to play a general knowledge quiz questions. The aim of this game is to have fun the players and learn of them answers.

Mobile software development has evolved over time. Android has emerged as a new mobile development platform, building on past successes and avoiding past failures of other platforms. This was designed to empower the developer to write innovative applications. One reason that i chose this platform to develop my application is that the platform is open source and developers enjoy many benefits over other competing platforms.

In order to summarize the key elements of this diploma thesis, they focus on three main aspects. The first one has to do with the development of Android environment using Eclipse IDE. I chose the Eclipse because the Android development team has integrated the Android development tools directly into the Eclipse IDE. However, developers are not constrained to using Eclipse, they can also use other IDEs. This part runs on our devices. The second one concerns the Database witch stores our data. The third one concerns the part which runs on Server.

In the next chapters we will analyze the three parts, the Android Environment, what it means the xml and php files, and how works the web services.

Περιεχόμενα

ΠΕΡΙΛΗΨΗ	1
ABSTRACT	2
Ευρετήριο Σχημάτων και Πινάκων.....	5
ΚΕΦΑΛΑΙΟ 1: Εισαγωγή	7
1.1 Εισαγωγή στο Android	7
1.2 Ιστορία δημιουργίας	7
1.3 Συσκευές Android στο μερίδιο της παγκόσμιας αγοράς	9
ΚΕΦΑΛΑΙΟ 2: Περιγραφή Λειτουργικού Συστήματος	12
2.1 Αρχιτεκτονική	12
2.1.1 Εφαρμογές και Widgets	12
2.1.2 Πλαίσιο Εφαρμογής	13
2.1.3 Χρόνος Εκτέλεσης	14
2.1.4 Εγγενείς Βιβλιοθήκες	15
2.1.5 Πυρήνας	15
2.2 Εκτέλεση και Συνεργασία Εφαρμογών	15
ΚΕΦΑΛΑΙΟ 3: Ανάπτυξη Εφαρμογών σε περιβάλλον Android	17
3.1 Προγραμματιστικά Εργαλεία	17
3.2 Δομικά Στοιχεία Του Android SDK.....	19
3.2.1 Δραστηριότητες (Activities)	19
3.2.2 Υπηρεσίες (Services).....	21
3.2.3 Πάροχοι Περιεχομένου (Content Providers)	21
3.2.4 Καθολικούς Παραλήπτες Μηνυμάτων (Broadcast Receivers)	22
3.3 Διεπαφή Χρήστη - User Interface	22
3.3.1 Προβολές (Views)	23
3.3.2 Στοιχεία Ελέγχου (Controls).....	23
3.3.3 Ομάδες Προβολής (View Group)	24
3.3.4 Διάταξη (Layout)	24
3.3.5 Στοιχεία Ελέγχου Που Περιέχουν Προβολές	25
ΚΕΦΑΛΑΙΟ 4 : Ανάλυση Επιμέρους Καταλόγων Στο Android Project.....	27
4.1.1 Κατάλογος src.....	28
4.1.2 Κατάλογος res	28
4.1.3 Κατάλογος gen (Generated Java Files)	28

4.1.4	Κατάλογος assets	29
4.1.5	Androidmanifest.xml	29
ΚΕΦΑΛΑΙΟ 5: Ανάπτυξη Εφαρμογής Παιχνιδιού		31
5.1	Περιγραφή Εφαρμογής.....	31
5.2	Έκδοση Πλατφόρμας Εφαρμογής	31
5.3	Υλοποίηση Εφαρμογής	32
5.4	Λογισμικό που χρησιμοποιήθηκε για την κατασκευή του Client Side τμήματος	33
5.4.1	Αρχεία xml της Εφαρμογής στο Android Project.....	33
5.4.2	Αρχεία java της Εφαρμογής στο Android Project.....	37
5.5	Λογισμικό που χρησιμοποιήθηκε για την κατασκευή του Server Side τμήματος	45
5.5.1	Σύστημα Διαχείρισης Βάσης Δεδομένων- MySQLServer.....	47
5.5.2	Αρχεία Php	51
Συμπεράσματα και Μελλοντικές Επεκτάσεις		56
Οδηγός Χρήσης Λογισμικού		58
Βιβλιογραφία		65

Ευρετήριο Σχημάτων και Πινάκων

Εικόνα 1:Πωλήσεις smartphone (πλατφόρμες)	9
Εικόνα 2:Το Android στα πρώτα του βήματα (2009)	9
Εικόνα 3:Οι 20 συσκευές με τις περισσότερες πωλήσεις	10
Εικόνα 4:Όλες οι εκδόσεις συγκριτικά με την χρήση τους το 2012	11
Εικόνα 5:Ακριβεί ποσοστά χρήσης εκδόσεων που κυκλοφορούν	11
Εικόνα 6: Επίπεδα Αρχιτεκτονικής του Λειτουργικού Συστήματος Android	12
Εικόνα 7:Android Runtime	14
Εικόνα 8:Βασικές Βιβλιοθήκες	15
Εικόνα 9:Στοιβα Δραστηριοτήτων Εκτέλεσης Εφαρμογών	16
Εικόνα 10:Eclipse IDE	17
Εικόνα 11:Android SDK Manager.....	18
Εικόνα 12:Android Virtual Device Manager	18
Εικόνα 13:Εικονική Συσκευή με λειτουργικό Android.....	19
Εικόνα 14:Ο κύκλος ζωής μιας Δραστηριότητας Android και οι βασικές μέθοδοι .	20
Εικόνα 15:Content Providers.....	21
Εικόνα 16:Παράδειγμα του Status bar notification.....	22
Εικόνα 17:Ιεραρχία κλάσεων Text View	23
Εικόνα 18:Παράδειγμα Edit Text	24
Εικόνα 19:Παράδειγμα στοιχείου ελέγχου Gallery	25
Εικόνα 20:Ιεραρχία κλάσεων για δημιουργία Gallery.....	25
Εικόνα 21:Παράδειγμα στοιχείου ελέγχου TabHost.....	26
Εικόνα 22:Βασικοί κατάλογοι νέου έργου	27
Εικόνα 23:Καρτέλα επεξεργασίας AndroidManifest.xml	29
Εικόνα 24:Επεξεργαστής Eclipse Manifest File	30
Εικόνα 25:Ιεραρχία καταλόγου layout του project.....	33
Εικόνα 26:Αρχείο main.xml.....	34
Εικόνα 27:Αρχείο main.xml.....	35
Εικόνα 28:Ιεραρχία αρχείων java του project	37
Εικόνα 29:HomeScreen Java class	39
Εικόνα 30:Κλάση Mobpplication	43
Εικόνα 31:Κλάση Xmlclass.....	44
Εικόνα 32:Client Server Connection.....	45
Εικόνα 33:Σύστημα Διαχείρισης Βάσης Δεδομένων MySQL Server.....	47
Εικόνα 34:Πίνακας Ερωτήσεων.....	48
Εικόνα 35:Διαχείριση Τοπικής Βάσης Δεδομένων.....	49
Εικόνα 36:Δομή του πίνακα answers	49
Εικόνα 37:Προσθήκη μιας εγγραφής σε πίνακα	50
Εικόνα 38:MySql insert into questions.....	50
Εικόνα 39:Εγγραφές του πίνακα Ερωτήσεων	51
Εικόνα 40:Αρχείο dbconn.php	51
Εικόνα 41:Αρχείο question.php	53
Εικόνα 42:Μορφή xml αρχείου α' φάση.....	53

Εικόνα 43:Τελική μορφή του thequestion xml αρχείου	53
Εικόνα 44:Αρχείο users.php	55
Εικόνα 45:Δημιουργία εικονιδίου στην κύρια οθόνη	58
Εικόνα 46:Φόρτωση Εφαρμογής.....	59
Εικόνα 47:Εισαγωγή Παίκτη	59
Εικόνα 48:Κύριο Μενού	60
Εικόνα 49:Οθόνη με των ερωτήσεων	60
Εικόνα 50:Γρήγορη και σωστή απάντηση	61
Εικόνα 51:Λάθος απάντηση	61
Εικόνα 52:Επιλογή Εμφάνισης Λίστας Κατάταξης.....	62
Εικόνα 53:Επιλογή Σύντομης Περιγραφής	63

ΚΕΦΑΛΑΙΟ 1: Εισαγωγή

1.1 Εισαγωγή στο Android

Ο προσωπικός υπολογιστής και το διαδίκτυο προσφέρουν με επαναστατικούς τρόπους στους ανθρώπους επικοινωνία, ψυχαγωγία και ανταλλαγή πληροφοριών. Πιο συγκεκριμένα, το διαδίκτυο επηρεάζει κάθε ανθρώπινη δραστηριότητα μέσω των υπηρεσιών που διαθέτει. Για παράδειγμα, κάνει ευκολότερη την απόκτηση γνώσης χωρίς να βάζει χρονικούς ή χωρικούς περιορισμούς, παρέχει άμεση ενημέρωση καθώς επίσης επιτρέπει την εύκολη και γρήγορη επικοινωνία (μέσω των υπηρεσιών κοινωνικής δικτύωσης που είναι ευρέως διαδεδομένοι στις μέρες μας). Ακόμη, μπορεί να χρησιμοποιηθεί τόσο ως εργαλείο στον κλάδο της εργασίας, όσο και της ψυχαγωγίας (δεδομένου ότι εύκολα κάποιος μπορεί να ακούσει μουσική, ραδιόφωνο, να δει τα video που θέλει, να παρακολουθήσει ταινίες ακόμη και τηλεόραση ή να ασχοληθεί με τα ηλεκτρονικά παιχνίδια).



Η εταιρία **Google** έκανε προσπάθεια να δημιουργήσει μια πλατφόρμα ,η οποία να εγκαθίσταται σε μια φορητή συσκευή ώστε να συνδυάζει όλα τα πλεονεκτήματα του διαδικτύου και να μπορεί να χρησιμοποιείται από όλους. Η συσκευή που είναι πλέον διαδεδομένη, προσιτή και χρησιμοποιείται σχεδόν από όλους είναι το κινητό τηλέφωνο. Είναι φανερό πως η συσκευή κινητού τηλεφώνου επηρεάζει σε μεγάλο βαθμό τη ζωή των ανθρώπων και αποτελεί μέρος της καθημερινότητάς τους. Έτσι, η Google ανέπτυξε μια πλατφόρμα με το όνομα “Android” που περιλαμβάνει ένα λειτουργικό σύστημα χρήστη και εφαρμογές που τρέχουν σε αυτό.

Στον τομέα των λειτουργικών συστημάτων κινητής τηλεφωνίας υπάρχουν μόνο τρεις βασικοί ανταγωνιστές που είναι η Apple με το “iPhone”, η Symbian με “Symbian OS” και η Microsoft με το “Windows Mobile”. Το Android τα ανταγωνίζεται με πολύ μεγάλη επιτυχία και αυτό γιατί συνεργάζεται με πολλές εταιρείες κατασκευής κινητών τηλεφώνων και συνδυάζει ένα τεράστιο πλήθος νέων εφαρμογών ακόμη και σε αρκετά οικονομικές συσκευές. Εκτός όμως από τα κινητά (smartphones) η πλατφόρμα Android έχει εφαρμοστεί και στις λεγόμενες συσκευές “Android PCs Tablets”. Τα Android PCs TABLETS είναι ο μεγάλος αντίπαλος του τύπου συσκευών της Apple, το λεγόμενο iPad. Είναι μια συσκευή που συνδυάζει τις δυνατότητες ενός κινητού τηλεφώνου και ενός υπολογιστή στο μέγεθος ενός μικρού βιβλίου.

1.2 Ιστορία δημιουργίας

Τον Ιούλιο του 2005, η Google απέκτησε την Android Inc , μια νέα μικρή εταιρία με έδρα το Palo Alto, της California των ΗΠΑ . Οι πρώτο-ιδρυτές είναι ο Andy Rubin

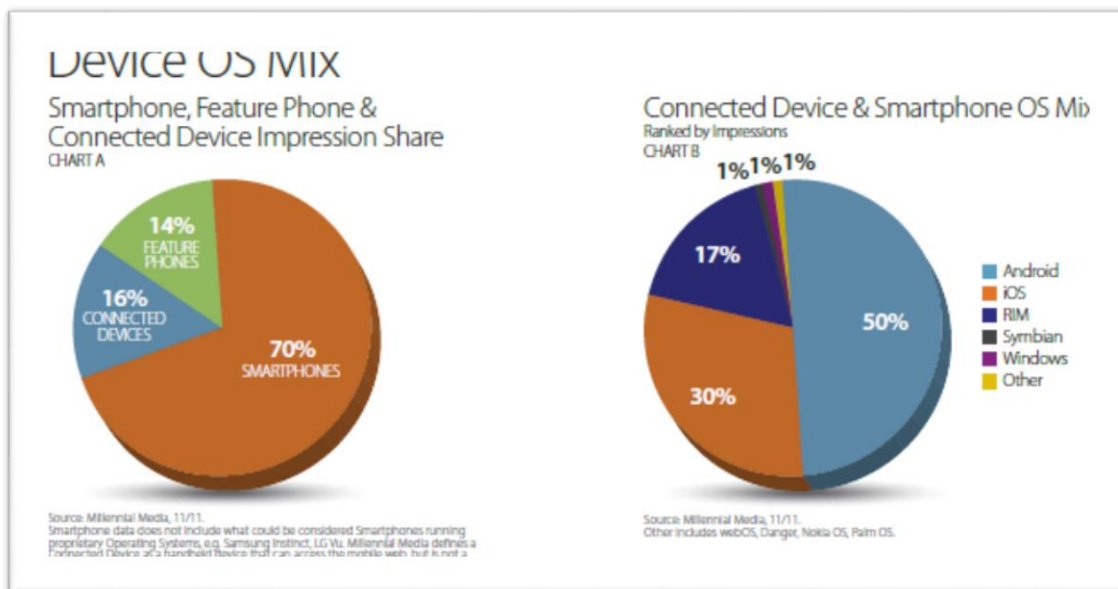
(συν-ιδρυτής της Danger), Rich Miner (συν-ιδρυτής της wildfire Communications), ο Nick Sears (VP σε T-Mobile), και ο Chris White (επικεφαλής σχεδιασμού και ανάπτυξης διεπαφής της WebTV). Την εποχή εκείνη, λίγα ήταν γνωστά για τη λειτουργία και τις εφαρμογές της Android , εκτός από ότι παρήγαγε λογισμικό για κινητά τηλέφωνα. Αυτό ξεκίνησε τις φήμες ότι η Google σχεδιάζει να εισέλθει στην αγορά κινητής τηλεφωνίας. Η ομάδα με επικεφαλής τον Rubin ανέπτυξε μια πλατφόρμα για κινητές συσκευές που βασίζεται στον πυρήνα του Linux[5].

Τον Δεκέμβριο του 2006 το BBC και η Wall Street Journal ανέφεραν ότι η Google θα ήθελε να παράγει εφαρμογές για κινητά τηλέφωνα και δούλευε σκληρά για να το πετύχει. Έντυπα και ηλεκτρονικά μέσα ενημέρωσης ανέφεραν σύντομα φήμες ότι είχε αναπτύξει ένα «Google-κινητό». Τον Σεπτέμβριο του 2007 δημοσιεύτηκε μελέτη αναφοράς ότι η Google είχε καταθέσει αρκετές αιτήσεις για διπλώματα ευρεσιτεχνίας στον τομέα της κινητής τηλεφωνίας. Στις 5 Νοεμβρίου του 2007 έγινε η πρώτη παρουσίαση της πλατφόρμας Android, πολλές εταιρείες συμπεριλαμβανομένων των Google, HTC, Motorola, η Intel, η Qualcomm, η Sprint Nextel, η T-Mobile, Samsung, Sony Ericsson, Toshiba και NVIDIA, παράλληλα ανακοινώθηκε πως ενώθηκαν για να σχηματίσουν τον οργανισμό Open Handset Alliance (OHA). Δήλωσαν το στόχο τους για την ανάπτυξη ανοικτού κώδικα για κινητές συσκευές. Η πρώτη συσκευή που βασίζεται στην πλατφόρμα Android κυκλοφόρησε τον Οκτώβριο του 2008 με την ονομασία T-Mobile G1 το οποίο κατασκεύασε η HTC και η T-Mobile παρείχε τις απαραίτητες υπηρεσίες τηλεφωνίας. Ταυτόχρονα η Google ανακοίνωσε ότι στην καινούρια πλατφόρμα θα ενσωματώνεται και τμήμα ανοιχτού κώδικα του εξυπηρετητή του παγκόσμιου ιστού Apache [2].

Λίγες μέρες αργότερα (Οκτώβριο 2008) , η Google ανακοίνωσε 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 επαφής, παρά μόνο την χρήση των κλασικών 'σκληρών' κουμπιών της συσκευής. Τον Μάιο του 2009 είχαμε την έκδοση Android 1.5, εν ονόματι 'Cupcake'. Η πλατφόρμα απέκτησε μεγάλη φήμη σχετικά γρήγορα καθώς μέσα στους επόμενους 18 μήνες, 60 διαφορετικές συσκευές από 21 διαφορετικούς κατασκευαστές έκαναν το ντεμπούτο τους σε 59 φορείς σε 48 χώρες σε όλο τον κόσμο. Χαρακτηριστικό στοιχείο που φανερώνει την ταχεία ανάπτυξη της πλατφόρμας είναι ότι τον Ιούνιο του 2010 η Google ανακοίνωσε πως περισσότερες από 160.000 συσκευές ενεργοποιούνται κάθε μέρα. Ο όρος "Android" έχει τις ρίζες του στην ελληνική ΑΝΔΡ- που σημαίνει «άνθρωπος ή αρσενικό» και την κατάληξη -ΕΙΔΕΣ, που χρησιμοποιείται για να σημάνει "ομοειδή ή του είδους".

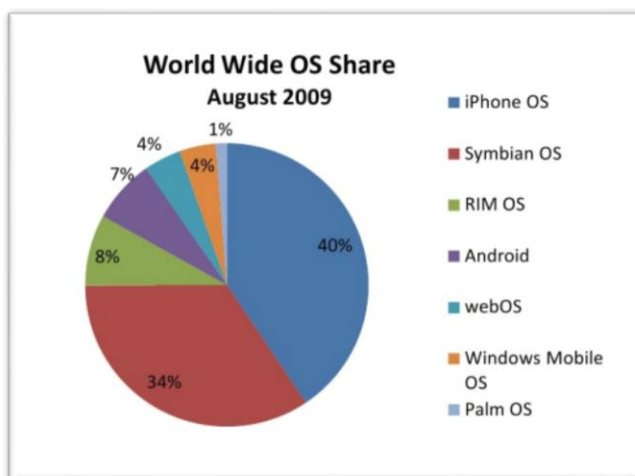
1.3 Συσκευές Android στο μερίδιο της παγκόσμιας αγοράς

Σύμφωνα με την έρευνα της εταιρίας Millennial Media που πραγματοποιήθηκε τον Νοέμβριο του 2011, το Android έχει αναδειχθεί ως το κορυφαίο σε πωλήσεις λειτουργικό σύστημα στον κόσμο για το τρίτο τρίμηνο του 2011. Ενδεικτικά, στην παρακάτω εικόνα παρουσιάζονται σε γραφική αναπαράσταση τα αποτελέσματα της έρευνας αυτής:



Εικόνα 1: Πωλήσεις smartphone (πλατφόρμες)

Όπως φαίνεται χαρακτηριστικά το 70% των κινητών τηλεφώνων που χρησιμοποιούνται είναι smartphone, και από αυτά το 50% διαθέτουν πλατφόρμα Android. Σαν δεύτερη σε πωλήσεις εμφανίζεται ο μεγάλος ανταγωνιστής του Android, η Apple με το iOS με ποσοστό 30%. Κάτι που είναι πολύ αναμενόμενο, αφού οι πωλήσεις συσκευών android αυξάνονται συνεχώς κερδίζοντας έδαφος στην αγορά των έξυπνων κινητών.



Εικόνα 2: Το Android στα πρώτα του βήματα (2009)

Συγκριτικά με το 2009 που το Android ήταν στα πρώτα του βήματα, με ποσοστό μόλις 7% ήταν στην τέταρτη θέση, όμως βλέπουμε με πόσο μεγάλους ρυθμούς έχει εδραιωθεί στην παγκόσμια αγορά.

TOP 20 MOBILE PHONES
Ranked by Impressions
CHART B

Rank	Devices	November 2011	Type	OS
1	Apple iPhone	13.54%	Smartphone	iOS
2	BlackBerry Curve	5.87%	Smartphone	BlackBerry OS
3	Motorola Droid X	5.27%	Smartphone	Android
4	HTC Evo	3.18%	Smartphone	Android
5	LG Optimus	3.08%	Smartphone	Android
6	BlackBerry Bold	3.03%	Smartphone	BlackBerry OS
7	HTC Desire	2.99%	Smartphone	Android
8	BlackBerry Torch	2.77%	Smartphone	BlackBerry OS
9	Samsung Nexus S	2.48%	Smartphone	Android
10	Samsung Vibrant Galaxy S	2.36%	Smartphone	Android
11	Samsung Galaxy S	1.66%	Smartphone	Android
12	HTC Droid Incredible	1.53%	Smartphone	Android
13	BlackBerry Pearl	1.25%	Smartphone	BlackBerry OS
14	ZTE Score	1.16%	Smartphone	Android
15	Motorola Droid	1.06%	Smartphone	Android
16	Samsung Fascinate	1.03%	Smartphone	Android
17	HUAWEI Ascend	1.02%	Smartphone	Android
18	HUAWEI Ideos	0.87%	Smartphone	Android
19	HTC MyTouch 4G Glacier	0.84%	Smartphone	Android
20	BlackBerry Bold Touch	0.82%	Smartphone	BlackBerry OS

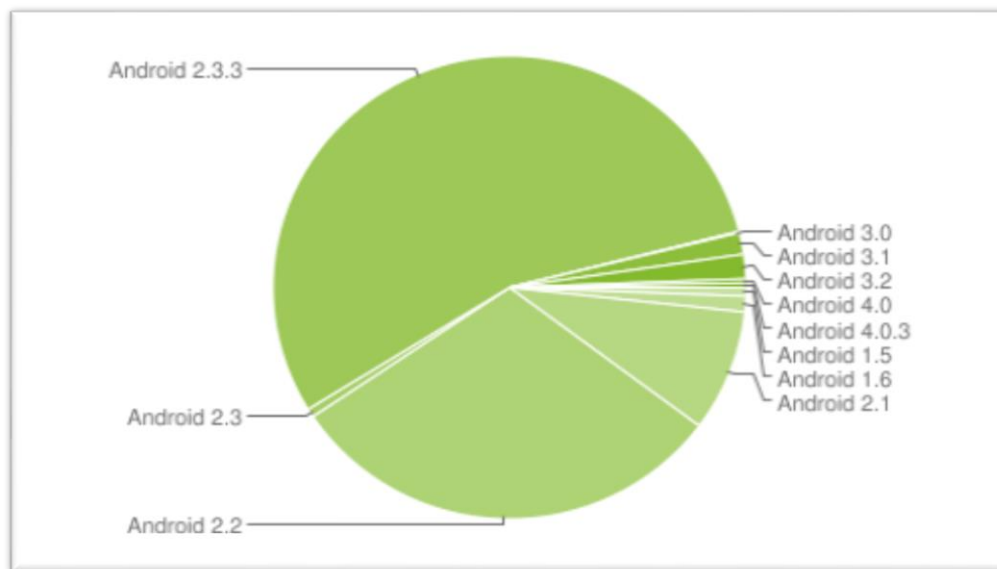
Source: Millennial Media, 11/11.

Εικόνα 3:Οι 20 συσκευές με τις περισσότερες πωλήσεις

Επιπλέον, στον παραπάνω πίνακα φαίνονται οι είκοσι συσκευές με τις περισσότερες πωλήσεις στον κόσμο. Αξιοσημείωτο είναι πως οι δεκατέσσερις από τις είκοσι υποστηρίζουν το λειτουργικό σύστημα Android. Αυτό είναι ένα δείγμα για το ότι αυτές οι συσκευές είναι ανταγωνιστικές σε πολλά χαρακτηριστικά, τόσο στην τιμή όσο και στις δυνατότητες που έχουν. Είναι σημαντικό να αναφέρουμε πως αυτή η πλατφόρμα αποτελεί ανοιχτού κώδικα λογισμικό, δεν βάζει φραγμούς και αυτός είναι ένας λόγος που το κάνει ακόμη πιο ξεχωριστό. Στο επόμενο κεφάλαιο θα αναλυθεί η αρχιτεκτονική και τα βασικά χαρακτηριστικά της πλατφόρμας αυτής.

Από την επίσημη ιστοσελίδα του Android Developers έχουμε μια εικόνα με το ποσοστό των εκδόσεων που χρησιμοποιούνται στην αγορά. Η πρώτη σε κυκλοφορία είναι η έκδοση 2.3 (τον Ιανουάριο του 2012), με ποσοστό 58.8% ενώ ακολουθεί με ποσοστό 30% η έκδοση που έχει επιλεγεί να αναπτυχθεί η

συγκεκριμένη εφαρμογή του παιχνιδιού. Αυτά τα στοιχεία αποτυπώνονται στο παρακάτω γράφημα:



Εικόνα 4:Όλες οι εκδόσεις συγκριτικά με την χρήση τους το 2012

Platform	Codename	API Level	Distribution
Android 1.5	Cupcake	3	0.6%
Android 1.6	Donut	4	1.1%
Android 2.1	Eclair	7	8.5%
Android 2.2	Froyo	8	30.4%
Android 2.3 - Android 2.3.2	Gingerbread	9	0.6%
Android 2.3.3 - Android 2.3.7		10	54.9%
Android 3.0	Honeycomb	11	0.1%
Android 3.1		12	1.5%
Android 3.2		13	1.7%
Android 4.0 - Android 4.0.2	Ice Cream Sandwich	14	0.3%
Android 4.0.3		15	0.3%

Εικόνα 5:Ακριβεί ποσοστά χρήσης εκδόσεων που κυκλοφορούν

ΚΕΦΑΛΑΙΟ 2: Περιγραφή Λειτουργικού Συστήματος

2.1 Αρχιτεκτονική

Το Android είναι μια “συλλογή λογισμικού” που περιλαμβάνει λειτουργικό σύστημα, εφαρμογές, ενδιάμεσο λογισμικό και διεπαφή χρήστη. Βασίζεται σε πυρήνα Linux, ο οποίος δρα σαν διαχωριστικό επίπεδο μεταξύ του επιπέδου του υλικού και του υπολοίπου λογισμικού. Η αρχιτεκτονική αυτή αποτελείται συνολικά από πέντε επίπεδα, όπου κάθε επίπεδο χρησιμοποιεί τις υπηρεσίες που του προσφέρονται από τα πιο κάτω επίπεδα. Στην παρακάτω εικόνα παρουσιάζεται η αρχιτεκτονική του λειτουργικού συστήματος Android:



Εικόνα 6: Επίπεδα Αρχιτεκτονικής του Λειτουργικού Συστήματος Android

2.1.1 Εφαρμογές και Widgets

Το επίπεδο των βασικών εφαρμογών (**Applications and Widgets**) περιλαμβάνει εφαρμογές όπως browser, e-mail client, SMS, χάρτες, ημερολόγιο, επαφές και πολλά άλλα. Αυτό είναι που βλέπουν οι χρήστες χωρίς να γνωρίζουν την όλη από κάτω διαδικασία. Αυτές είναι εφαρμογές που γράφουν οι κατασκευαστές λογισμικού, εφαρμογές που ήδη είναι εγκατεστημένες στο τηλέφωνο ή που ο χρήστης παίρνει από το Android Market. Οι εφαρμογές είναι προγράμματα που

καταλαμβάνουν ολόκληρη την οθόνη και αλληλεπιδρούν με το χρήστη. Από την άλλη τα widgets λειτουργούν σε μικρά τετράγωνα μέσα στην αρχική οθόνη – εφαρμογή . Όλες αυτές οι εφαρμογές είναι γραμμένες σε γλώσσα προγραμματισμού Java, και θεωρούνται ισότιμες μεταξύ τους. Θα πρέπει να αναφερθεί ότι οι εφαρμογές μπορούν να τρέχουν ταυτόχρονα, είναι δυνατόν ο χρήστης να ακούσει μουσική και να διαβάσει ένα e-mail την ίδια στιγμή.

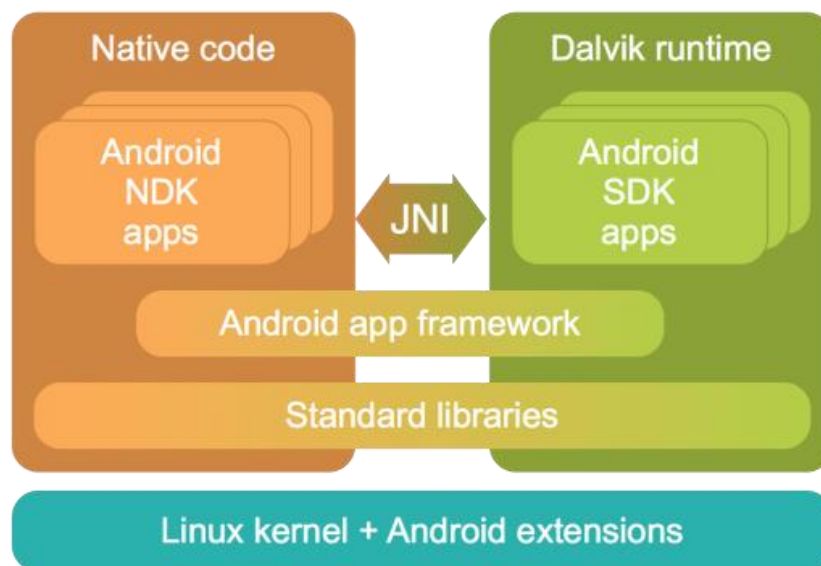
2.1.2 Πλαίσιο Εφαρμογής

Το επίπεδο του πλαισίου εφαρμογής (**Application Framework**) βρίσκεται δομικά πάνω από τις εγγενής βιβλιοθήκες και το χρόνο εκτέλεσης. Οι εφαρμογές έχουν πρόσβαση στις βασικές βιβλιοθήκες του λειτουργικού συστήματος μέσω των κατάλληλων διεπαφών και μέσω του πλαισίου. Αυτό το επίπεδο μας παρέχει υψηλού επιπέδου δομικές μονάδες τις οποίες μπορούμε να χρησιμοποιούμε για την κατασκευή των εφαρμογών μας. Είναι προ-εγκατεστημένο στο Android, αλλά είναι επεκτάσιμο, αφού ο κάθε κατασκευαστής μπορεί να το συμπληρώσει με δικά του κομμάτια. Τα σημαντικότερα δομικά στοιχεία του πλαισίου αυτού είναι:

- i. **Σύστημα Απεικόνισης (View System):** Δίνει την δυνατότητα χρήσης λιστών, πεδίων κειμένου, κουμπιών κλπ.
- ii. **Διαχειριστής Δραστηριοτήτων (Activity Manager):** Είναι υπεύθυνο για τον έλεγχο του χρόνου ζωής κάθε εφαρμογής και για την διατήρηση μιας στοίβας που επιτρέπει την πλοήγηση του χρήστη σε προηγούμενες οθόνες - καταστάσεις.
- iii. **Πάροχος Περιεχομένου (Content Provider):** Αυτά τα αντικείμενα έχουν πρόσβαση σε δεδομένα άλλων εφαρμογών ή περιέχουν δεδομένα που μπορούν να διαμοιραστούν μεταξύ εφαρμογών όπως για παράδειγμα οι επαφές του χρήστη.
- iv. **Διαχειριστής Πόρων (Resource Manager):** Παρέχει πρόσβαση σε πόρους. Οι πόροι, είναι οτιδήποτε υπάρχει σε ένα πρόγραμμα το οποίο δεν είναι κώδικας. Για παράδειγμα μπορεί να είναι κωδικοί χρωμάτων, αλφαριθμητικοί χαρακτήρες ή ακόμα και έτοιμα σχεδιαγράμματα οθονών φτιαγμένα σε XML, τα οποία μπορεί το πρόγραμμα να καλεί.
- v. **Διαχειριστής Τοποθεσίας (Location Manager):** Χρησιμοποιείται για να μπορεί να ξέρει το τηλέφωνο που βρίσκεται ανά πάσα στιγμή.
- vi. **Διαχειριστής Κοινοποιήσεων (Notification Manager):** Αποτελεί ιδανικό τρόπο για να ενημερώνεται ο χρήστης για γεγονότα που συμβαίνουν, διακριτικά χωρίς να διακόπτεται η εργασία του. Με άλλα λόγια διαχειρίζεται τα μηνύματα εφαρμογών που εμφανίζονται στην status bar, όπως εισερχόμενα μηνύματα, ειδοποιήσεις κλπ.

2.1.3 Χρόνος Εκτέλεσης

Ο χρόνος εκτέλεσης (**runtime**) του Android αποτελείται από δύο στοιχεία. Πρώτον, ένα σύνολο βασικών βιβλιοθηκών πυρήνα και δεύτερον, η εικονική μηχανή (Virtual Dalvik Machine) η οποία λειτουργεί σαν μεταφραστής μεταξύ του επιπέδου της εφαρμογής και του λειτουργικού συστήματος. Κάθε εφαρμογή που τρέχει στο Android είναι γραμμένη σε Java. Δεδομένου ότι το λειτουργικό σύστημα δεν είναι σε θέση να κατανοήσει άμεσα τα προγράμματα σε Java, μεταφράζονται από τη Virtual Dalvik Machine. Η Dalvik τρέχει .dex αρχεία, τα οποία είναι bytetimes που προέρχονται από αρχεία .class και .jar. Εν αντιθέσει όμως με τα .class αρχεία, τα .dex είναι πολύ πιο συμπαγή και αποδοτικά, γεγονός σημαντικό για συσκευές με περιορισμένη μνήμη και μπαταρία. Ο μεταφρασμένος κώδικας μπορεί στη συνέχεια να εκτελεστεί από το λειτουργικό σύστημα. Κάθε εφαρμογή ενθυλακώνεται στη δική της Virtual Dalvik Machine ακόμη και αν ορισμένες εφαρμογές τρέχουν παράλληλα. Το πλεονέκτημα είναι ότι τα διάφορα προγράμματα δεν επηρεάζουν το ένα το άλλο, έτσι, ένα σφάλμα προγράμματος για παράδειγμα, μπορεί να οδηγήσει σε κατάρρευση του προγράμματος, αλλά όχι ολόκληρου του συστήματος.



Εικόνα 7: Android Runtime

2.1.4 Εγγενείς Βιβλιοθήκες

Οι διαθέσιμες βιβλιοθήκες (**Native Libraries**) είναι γραμμένες σε C / C ++ και μεταγλωττίστηκαν για την συγκεκριμένη αρχιτεκτονική υλικού που χρησιμοποιείται από το τηλέφωνο. Αυτές καλούνται μέσω Java interface, περιλαμβάνουν την Επιφάνεια Διαχείρισης (για compositing παράθυρα), 2D και 3D γραφικών, codecs για αναπαραγωγή πολυμέσων, όπως MPEG-4 και MP3, η βάση δεδομένων SQL SQLite και το διαδίκτυο μηχανής περιήγησης WebKit. Οι βιβλιοθήκες αυτές δεν είναι εφαρμογές που μπορούν να σταθούν από μόνες τους. Υπάρχουν για να μπορούν να κληθούν από προγράμματα υψηλότερου επιπέδου. Από την έκδοση Donut και μετά , οι κατασκευαστές μπορούν να γράφουν τις δικές τους τέτοιες βιβλιοθήκες με την χρήση της Εργαλειοθήκης NDK (Native Development Kit).



Εικόνα 8:Βασικές Βιβλιοθήκες

2.1.5 Πυρήνας

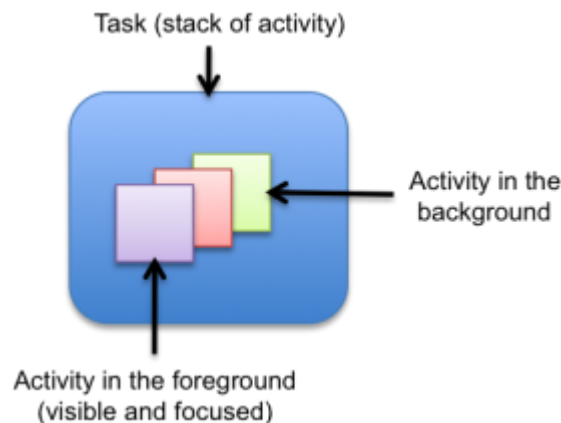
Τέλος, ο πυρήνας (**linux kernel**) είναι δοκιμασμένος, σταθερός και πετυχημένος και μπορεί να χρησιμοποιηθεί παντού, από ρολόγια χειρός μέχρι υπερυπολογιστές. Το Linux παρέχει στο Android το αφαιρετικό επίπεδο υλικού, επιτρέποντάς του να μπορεί να χρησιμοποιηθεί σε μεγάλη ποικιλία πλατφόρμων στο μέλλον. Ειδικότερα, το Android χρησιμοποιεί τον συγκεκριμένο πυρήνα για την διαχείριση υπηρεσιών του λειτουργικού όπως την διαχείριση της μνήμης, την διαχείριση των διεργασιών, την δικτύωση, την χρήση πληκτρολογίου μέσω οθόνης αφής, την χρήση αισθητήρων κ.α.

2.2 Εκτέλεση και Συνεργασία Εφαρμογών

Οι εφαρμογές που προορίζονται να «τρέχουν» σε φορητές συσκευές είναι πολύ σημαντικό να περιορίζουν τις απαιτήσεις τους σε πόρους στο ελάχιστο δυνατό. Το

λειτουργικό σύστημα Android έχει σχεδιαστεί έτσι ώστε να επιτρέπει σε πολλές εφαρμογές να εκτελούνται ταυτόχρονα αρκεί να υπάρχει η απαραίτητη διαθέσιμη μνήμη και επεξεργαστική ισχύς. Κάθε εφαρμογή πρέπει να μπορεί ανά πάσα στιγμή να είναι ενεργή, να παύεται, να τερματίζεται ή να ανακτάται από το παρασκήνιο χωρίς να προκαλεί προβλήματα στη λειτουργία της συσκευής και χωρίς να επηρεάζει τις λειτουργίες των υπολοίπων εφαρμογών. Έτσι είναι σημαντικό οι εφαρμογές να είναι σχεδιασμένες ώστε να είναι υπεύθυνες για την διαχείριση της κατάστασης, της μνήμης, των πόρων και των δεδομένων τους. Αυτό επιτυγχάνεται με τον διαχωρισμό των εφαρμογών σε επιμέρους δραστηριότητες (activities) που θα αναπτύξουμε αναλυτικά στο επόμενο κεφάλαιο.

Όλα τα αντικείμενα των δραστηριοτήτων παρακολουθούνται και τοποθετούνται σε μία στοίβα δραστηριοτήτων όπως ορίζεται από το λειτουργικό σύστημα. Μπορεί να υπάρχει μόνο μία ενεργή εφαρμογή η οποία να είναι ορατή στο χρήστη κάθε φορά – συγκεκριμένα, σε οποιαδήποτε δεδομένη στιγμή υπάρχει μόνο μία **Activity** στο προσκήνιο. Όταν μια νέα δραστηριότητα ξεκινά, η activity που βρίσκεται στην κορυφή της στοίβας διακόπτεται προσωρινά και η νέα activity προωθείται στην κορυφή της στοίβας. Αντίστοιχα, όταν η activity σταματά αφαιρείται από την στοίβα, και η προηγούμενη δραστηριότητα ξεκινά και πάλι.

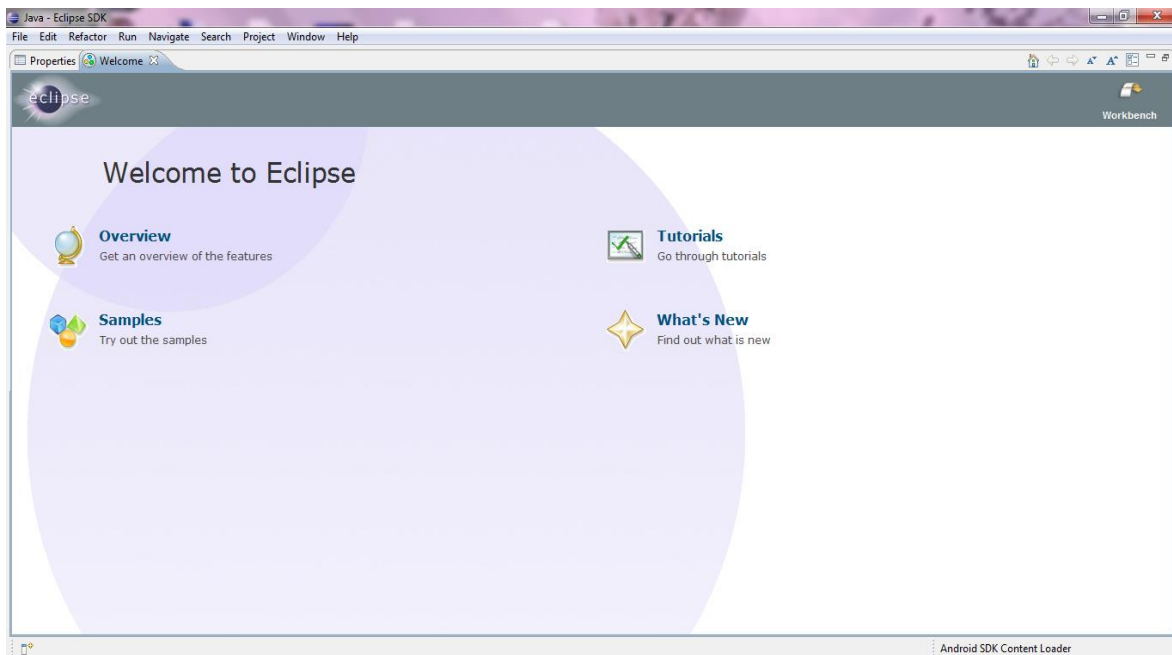


Εικόνα 9:Στοίβα Δραστηριοτήτων Εκτέλεσης Εφαρμογών

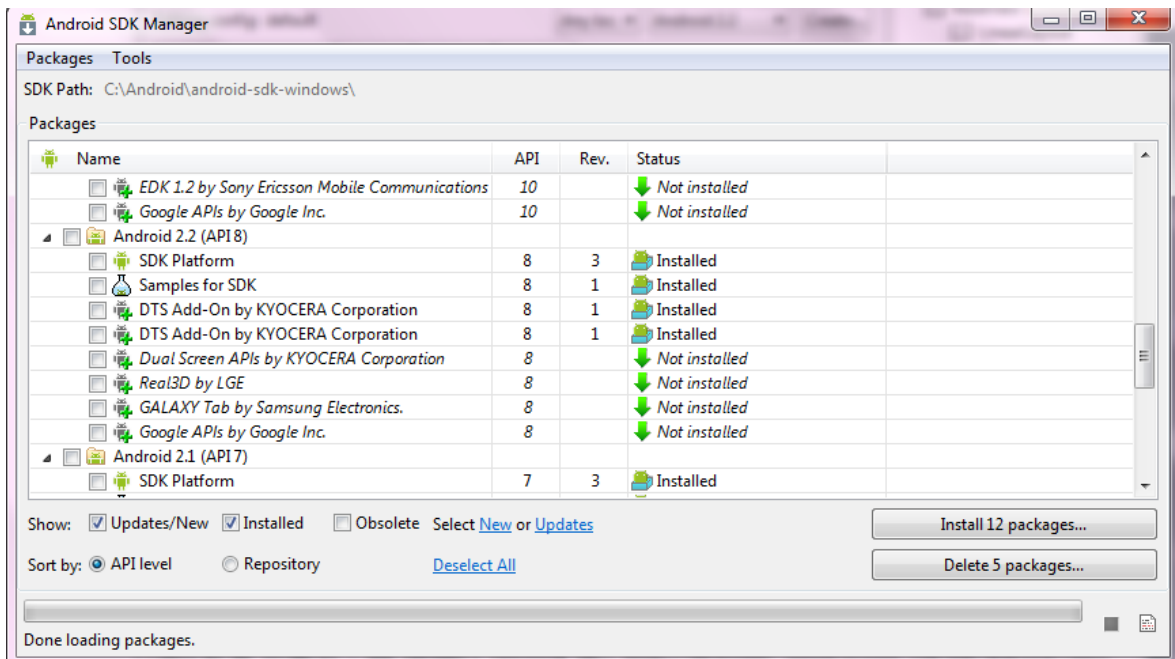
ΚΕΦΑΛΑΙΟ 3: Ανάπτυξη Εφαρμογών σε περιβάλλον Android

3.1 Προγραμματιστικά Εργαλεία

Η γλώσσα προγραμματισμού για την ανάπτυξη εφαρμογών είναι η Java, οπότε πρωταρχικό κομμάτι είναι να διαθέτουμε τα εργαλεία αυτής της γλώσσας που περιέχονται στο Java Development Kit (**JDK**). Επιπλέον, χρειαζόμαστε ένα ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment ή **IDE**) για να μεταγλωττίσουμε και να τρέχουμε τα προγράμματα. Το βασικότερο όμως εργαλείο είναι το **Android SDK** (Software Development Kit) το οποίο μας παρέχει τα εξειδικευμένα εργαλεία για την γραφή κώδικα εφαρμογής Android. Η σύνδεση του SDK με το γραφικό περιβάλλον του IDE γίνεται μέσω μιας επέκτασης (Android Development Tools ή **ADT Plug-in**) που εγκαθιστάμε στο IDE για να μπορεί να μεταγλωττιστεί η εφαρμογή και στην συνέχεια να 'τρέξει'. Το λειτουργικό σύστημα Android συνεχώς εξελίσσεται και κυκλοφορεί σε διάφορες εκδόσεις, κάθε έκδοση χρησιμοποιεί κάποια συγκεκριμένα διαφορετικά στοιχεία από τις υπόλοιπες.

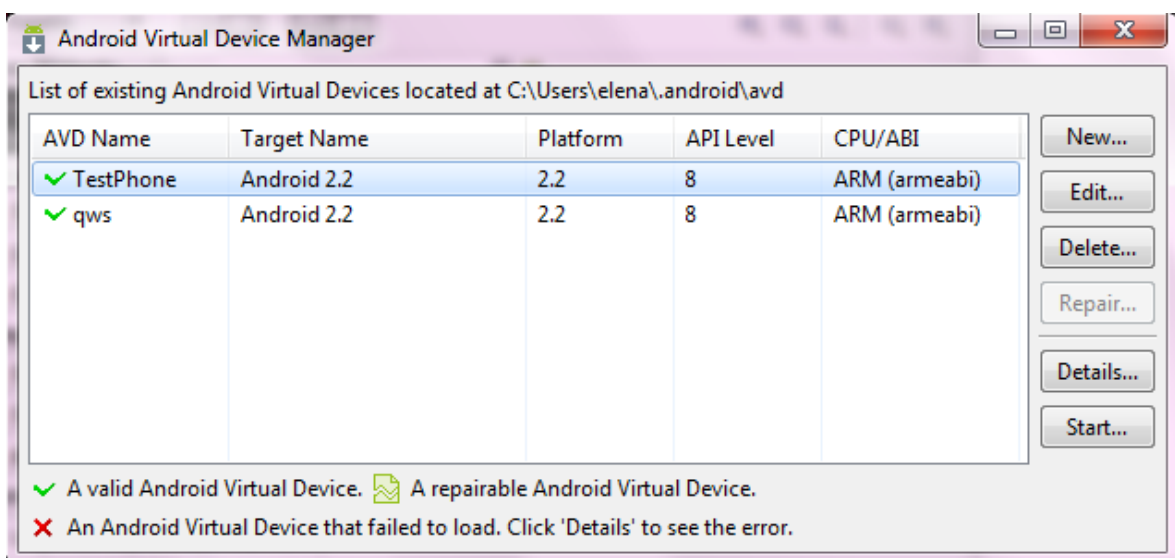


Εικόνα 10: Eclipse IDE



Εικόνα 11:Android SDK Manager

Το ADT Plug-in μας παρέχει τα εργαλεία για κάθε έκδοση. Τέλος, για να δοκιμάσουμε την εφαρμογή και να δούμε τα αποτελέσματά της υπάρχει το αντίστοιχο εργαλείο προσομοίωσης και αυτό είναι μία εικονική συσκευή Android (Android Virtual Device ή **AVD**). Πρόκειται για προσομοιωτή κινητής συσκευής στον υπολογιστή μας τόσο σε software όσο και hardware καθώς υπάρχει δυνατότητα ρύθμισης πολλών παραμέτρων 'όπως η έκδοση, το μέγεθος μνήμης, οι διαστάσεις της οθόνης κλπ.



Εικόνα 12:Android Virtual Device Manager



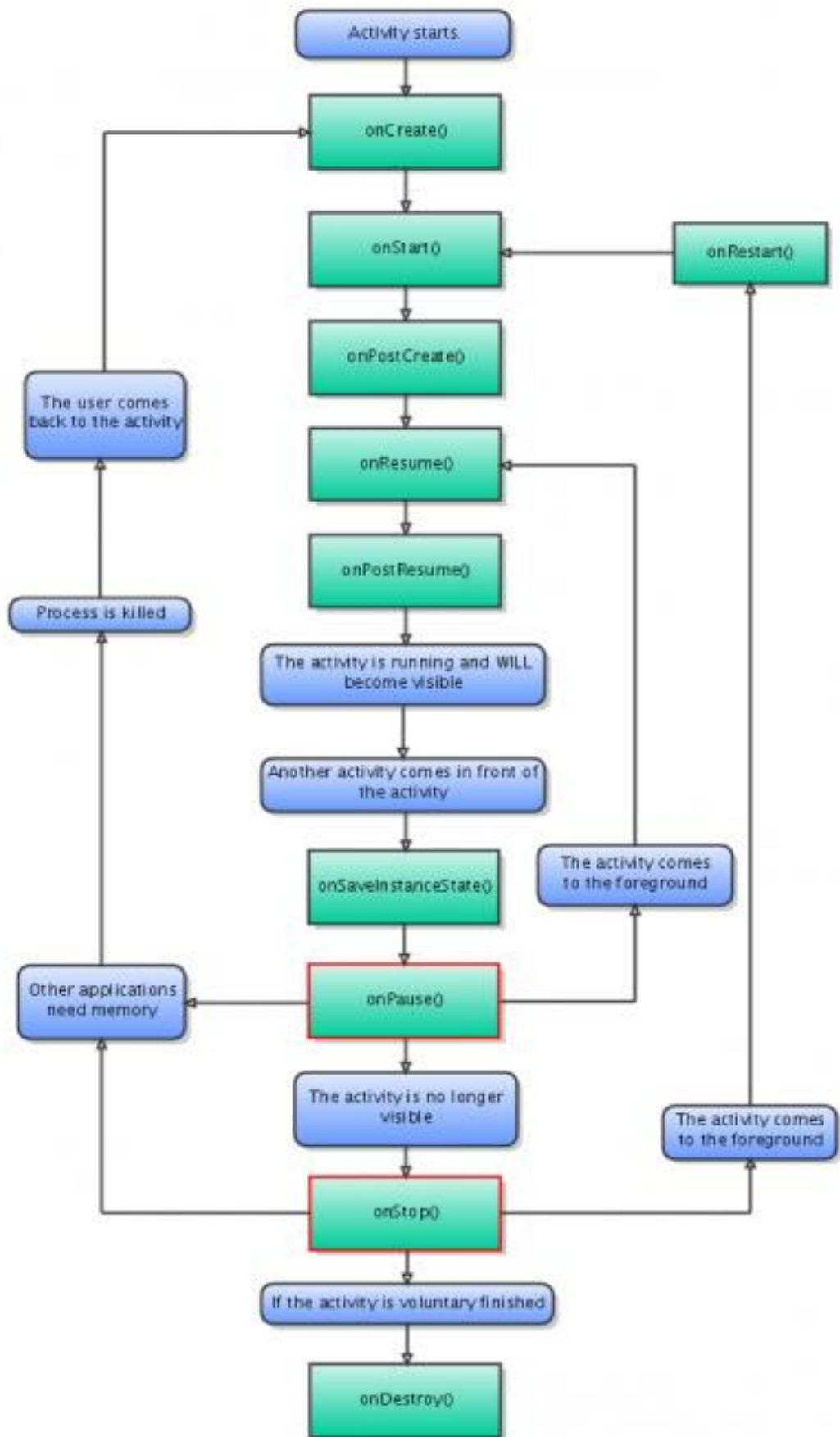
Εικόνα 13:Εικονική Συσκευή με λειτουργικό Android

3.2 Δομικά Στοιχεία Του Android SDK

Κάθε εφαρμογή Android αποτελείται από κάποια βασικά στοιχεία που την κάνουν αποδεκτή και της προσδίδουν επικοινωνία με το λειτουργικό σύστημα του τηλεφώνου. Τα δομικά στοιχεία είναι:

3.2.1 Δραστηριότητες (Activities)

Αποτελούν το κυριότερο στοιχείο κάθε εφαρμογής. Μία εφαρμογή είναι μια συλλογή από δραστηριότητες. Κάθε δραστηριότητα σε μια εφαρμογή αναλαμβάνει μια μοναδική εργασία ή έχει ένα μοναδικό σκοπό. Πιο συγκεκριμένα, αντιπροσωπεύει μια διαφορετική οθόνη (παράθυρο) της εφαρμογής με την οποία έρχεται σε αλληλεπίδραση ο χρήστης. Οι δραστηριότητες συνδέονται μεταξύ τους και μπορεί η μία να καλέσει την άλλη. Επιπλέον, είναι υπεύθυνες να αποθηκεύουν την «κατάσταση» τους ώστε να μπορεί να πραγματοποιηθεί η αλλαγή από την μία στην άλλη εύκολα, χωρίς να διαταραχθεί ο κύκλος ζωής της ίδιας της εφαρμογής.



Εικόνα 14:Ο κύκλος ζωής μιας Δραστηριότητας Android και οι βασικές μέθοδοι

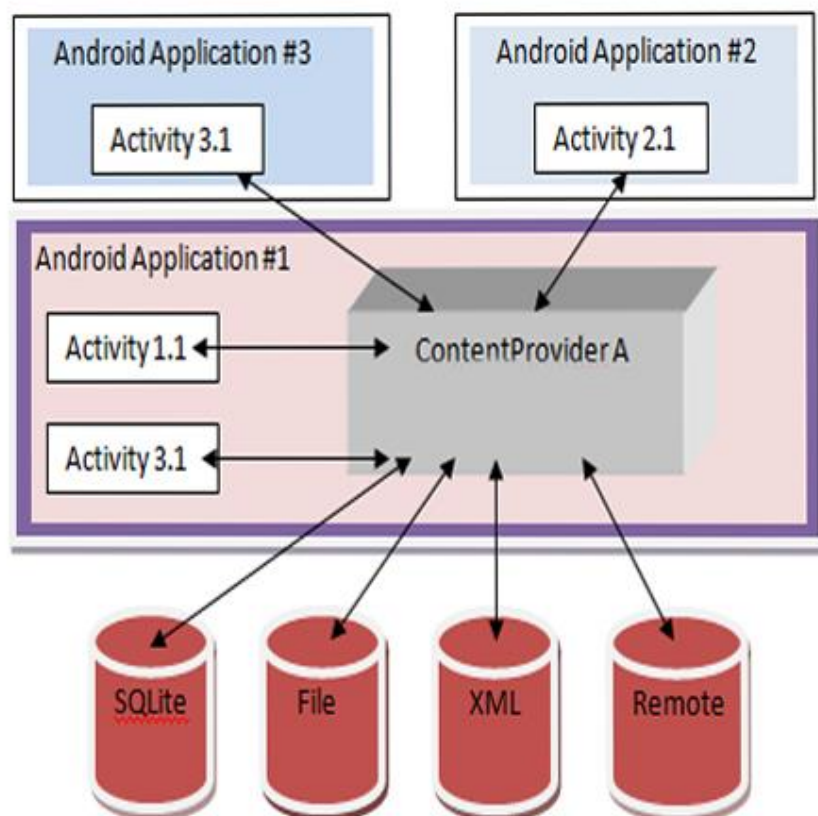
3.2.2 Υπηρεσίες (Services)

Οι εργασίες που δεν απαιτούν αλληλεπίδραση με τον χρήστη μπορούν να ενθυλακωθούν σε μια υπηρεσία. Μια υπηρεσία είναι περισσότερο χρήσιμη όταν κάποιες λειτουργίες διαρκούν πολύ ή εκτελούνται στο παρασκήνιο, πρέπει να πραγματοποιούνται τακτικά (πχ όπως ο έλεγχος ενός διακομιστή για νέα αλληλογραφία) ή να ενεργεί περίπου όπως μια υπηρεσία web εκτελώντας την επεξεργασία αιτήσεων από τρίτους.

3.2.3 Πάροχοι Περιεχομένου (Content Providers)

Οι content providers διαχειρίζονται αποθηκευτικούς χώρους για δεδομένα, οι οποίοι είναι προσπελάσιμοι από όλες τις εφαρμογές. Για παράδειγμα τέτοιο αποθηκευτικός χώρος είναι τα αρχεία ήχου, βίντεο, εικόνων, οι επαφές του τηλεφωνικού καταλόγου κτλ. Μια εφαρμογή μπορεί να χρησιμοποιήσει μια εικόνα από το αρχείο εικόνων και να το αντιστοιχίσει με μια επαφή ώστε με κάθε κλήση στο συγκεκριμένο αριθμό να εμφανίζεται η ίδια εικόνα.

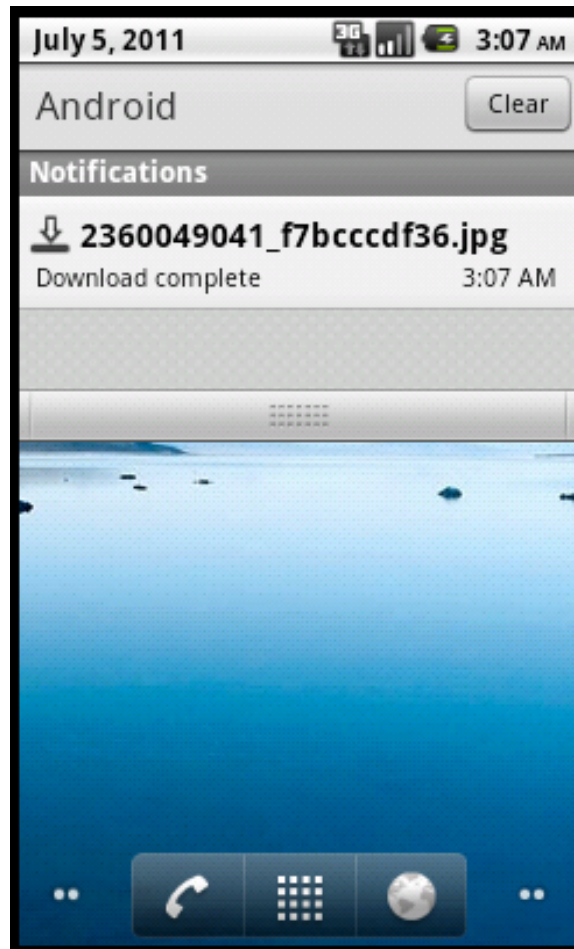
Ακόμη, οι content providers μπορούν να γράφουν και να διαβάζουν δεδομένα από μια ενσωματωμένη στο λειτουργικό σύστημα βάση δεδομένων την SQLite database.



Εικόνα 15:Content Providers

3.2.4 Καθολικούς Παραλήπτες Μηνυμάτων (Broadcast Receivers)

Οι προθέσεις ενημερώνονται για κάποιο συγκεκριμένο γεγονός από το λειτουργικό σύστημα και τότε ενεργοποιούνται. Πιο συγκεκριμένα, γίνεται εκπομπή ενός αντικειμένου στο σύστημα και οποιαδήποτε εφαρμογή ενδιαφέρεται μπορεί να λάβει αυτή την εκπομπή. Τέτοια γεγονότα μπορεί να είναι ότι η οθόνη έχει σβήσει, ότι η στάθμη της μπαταρίας είναι χαμηλή κλπ. Η οπτική επαφή που μπορεί να έχει ο χρήστης με τους broadcast receivers είναι και μέσω των status bar notifications.



Εικόνα 16: Παράδειγμα του Status bar notification

3.3 Διεπαφή Χρήστη - User Interface

Το μεγαλύτερο ποσοστό των εφαρμογών Android αποτελούνται από κάποιο είδος διεπαφής χρήστη. Η διεπαφή χρήστη έχει τεράστια σημασία για την εφαρμογή

διότι αποτελεί την τελική εικόνα που βλέπει ο χρήστης, το γραφικό περιβάλλον που περιηγείται και ενεργοποιεί όλες τις λειτουργίες της εφαρμογής. Τα στοιχεία διεπαφής χρήστη, καθορίζουν την επικοινωνία και αλληλεπίδραση του χρήστη και επηρεάζουν άμεσα την λειτουργικότητα της κάθε εφαρμογής. Κάποια από αυτά τα στοιχεία εμφανίζουν πληροφορίες στον χρήστη ενώ άλλα συλλέγουν πληροφορίες από αυτόν. Οι απαραίτητες κλάσεις για την δημιουργία των στοιχείων αυτών τις βρίσκουμε στο πακέτο ανάπτυξης λογισμικού (SDK). Θεμελιώδης κλάσεις για τον σχεδιασμό διεπαφής χρήστη είναι: οι προβολές, οι ομάδες προβολής και οι διεργασίες.

3.3.1 Προβολές (Views)

Το Android SDK έχει ένα πακέτο Java που ονομάζεται `android.view`. Αυτό το πακέτο περιέχει πολλές διεπαφές και κλάσεις που σχετίζονται με την σχεδίαση στην οθόνη. Ειδικότερα, το αντικείμενο `View` είναι μια υποκλάση του πακέτου `android.view`. Η κλάση `android.view.View` είναι το βασικό συστατικό των διεπαφών χρήστη και αναπαριστά ένα ορθογώνιο τμήμα της οθόνης, και αποτελεί την βασική κλάση για όλα τα στοιχεία ελέγχου της διεπαφής χρήστη και τις διατάξεις της οθόνης.

3.3.2 Στοιχεία Ελέγχου (Controls)

Στο Android SDK περιέχεται ένα πακέτο Java που ονομάζεται `android.widget`. Σε αυτό το πακέτο ανήκουν αντικείμενα τα οποία είτε εμφανίζουν κάτι στον χρήστη, είτε τον βοηθούν στην χρήση-περιήγηση στην εφαρμογή, είτε παίρνουν πληροφορίες από αυτόν και σύμφωνα με τις επιλογές που κάνει έχει και τα ανάλογα αποτελέσματα. Ουσιαστικά αποτελούν στοιχεία αλληλεπίδρασης. Μερικά παραδείγματα τέτοιων στοιχείων είναι τα `Text View`, `Edit Text`, `Button`, `Image View` κλπ.



Εικόνα 17:Ιεραρχία κλάσεων `Text View`



Εικόνα 18: Παράδειγμα Edit Text

3.3.3 Ομάδες Προβολής (View Group)

Τα στοιχεία έλεγχου View Group αναπαριστούν ένα ορθογώνιο στην οθόνη, με την διαφορά ότι περιέχουν άλλα αντικείμενα View τα οποία εμφανίζουν με ένα οργανωμένο τρόπο. Οι υποκλάσεις View Group χωρίζονται σε δύο κατηγορίες: στις διατάξεις και στα στοιχεία ελέγχου που περιέχουν προβολές.

3.3.4 Διάταξη (Layout)

Ένας ειδικός τύπος στοιχείου ελέγχου είναι η διάταξη (**Layout**), η οποία είναι ένα αντικείμενο View αλλά δεν σχεδιάζει κάτι στην οθόνη απλά χρησιμοποιείται σαν στοιχείο γονέας για την οργάνωση άλλων στοιχείων ελέγχου. Οι διατάξεις προέρχονται από την κλάση `android.view.ViewGroup`. Υπάρχουν οι εξής τύποι διατάξεων:

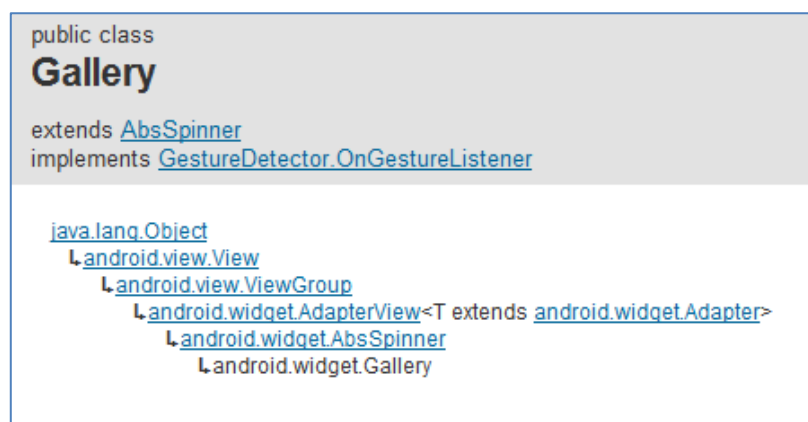
- **LinearLayout** (Γραμμική Διάταξη): Με αυτή την προβολή οργανώνονται τα αντικείμενα σε μια οριζόντια ή κατακόρυφη γραμμή
- **TableLayout** (Διάταξη Πίνακα): Αυτή η οργάνωση τοποθετεί τα αντικείμενα όπως θα έμπαιναν σε ένα πίνακα, σε πολλές γραμμές (TableRow) και σε στήλες
- **RelativeLayout** (Σχετική Διάταξη): Είναι μια οργάνωση των προβολών σύμφωνα με την οποία η θέση των αντικειμένων καθορίζεται σε σχέση με τα υπόλοιπα αντικείμενα στην συγκεκριμένη οθόνη. Για παράδειγμα, το νέο αντικείμενο View θα τοποθετηθεί πάνω, κάτω, δεξιά ή αριστερά ενός άλλου View
- **FrameLayout** (Διάταξη Πλαισίου): Κάθε αντικείμενο τοποθετείται στην πάνω αριστερή γωνία κάθε προβολής

3.3.5 Στοιχεία Ελέγχου Που Περιέχουν Προβολές

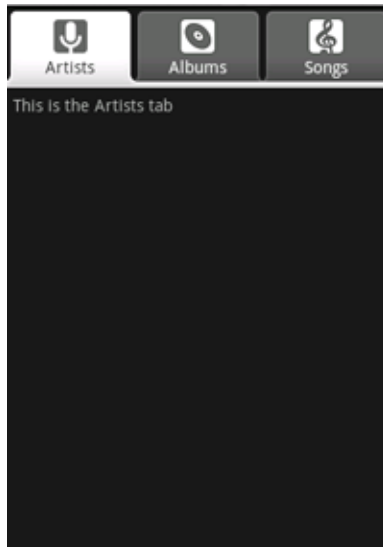
Η δεύτερη κατηγορία υποκλάσεων του View Group είναι κλάσεις που περιέχουν αντικείμενα View, όπως και οι διατάξεις με την μόνη διαφορά ότι παρέχουν κάποιες λειτουργίες που επιτρέπουν στους χρήστες να αλληλεπιδρούν με αυτά σαν να ήταν κανονικά στοιχεία ελέγχου. Κάποιες κλάσεις που ανήκουν σε αυτή την κατηγορία είναι: Gallery, GridView, ListView, ScrollView, TabHost κπλ. Στην παρακάτω εικόνα παρουσιάζεται ένα τέτοιο στοιχείο της υποκλάσης Gallery, όπου εμφανίζεται μια λίστα με εικόνες, και ο χρήστης έχει την δυνατότητα να περιηγηθεί στη λίστα με την ιδιότητα της αριστερής και δεξιάς κύλισης.



Εικόνα 19:Παράδειγμα στοιχείου ελέγχου Gallery



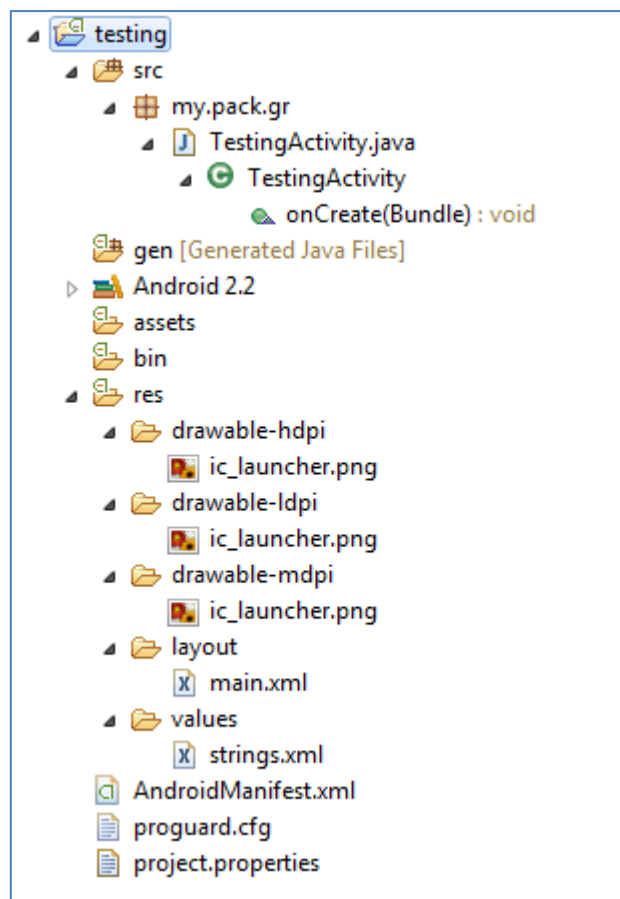
Εικόνα 20:Ιεραρχία κλάσεων για δημιουργία Gallery



Εικόνα 21:Παράδειγμα στοιχείου ελέγχου TabHost

ΚΕΦΑΛΑΙΟ 4: Ανάλυση Επιμέρους Καταλόγων Στο Android Project

Κάθε νέα εφαρμογή Android προέρχεται από την δημιουργία ενός έργου «project» (στην παρούσα πτυχιακή χρησιμοποιήθηκε το Eclipse IDE). Καθένα έργο έχει το δικό του μοναδικό και αναγνωριστικό όνομα, καθώς και ένα ή περισσότερα πακέτα για την συγκεκριμένη εφαρμογή. Αρχικά, εκτός από το όνομα καθορίζεται σε ποια έκδοση πλατφόρμας θα τρέχει η εφαρμογή μας (Android 2.2). Όλες οι εφαρμογές αποτελούνται από δύο πράγματα: λειτουργίες (οδηγίες κώδικα) και δεδομένα (πόρους). Οι λειτουργίες είναι ο κώδικας που ορίζει την συμπεριφορά της εφαρμογής, οι πόροι περιλαμβάνουν εικόνες, συμβολοσειρές κειμένου, αρχεία ήχου και βίντεο, και άλλα δεδομένα που χρησιμοποιούνται από την εφαρμογή. Οι πιο κοινοί τύποι πόρων αποθηκεύονται σε αρχεία xml. Στην παρακάτω εικόνα αποτυπώνεται η ιεραρχία των καταλόγων κάθε έργου:



Εικόνα 22:Βασικοί κατάλογοι νέου έργου

4.1.1 Κατάλογος src

Η ονομασία «**src**» προέρχεται από την Αγγλική λέξη «source» που σημαίνει «πηγή». Αυτός ο κατάλογος περιλαμβάνει αλγορίθμους που συμβάλλουν στην εκτέλεση της εφαρμογής, και είναι αρχεία κλάσεων java. Σε αυτόν τον κατάλογο μπορεί να γίνει δήλωση και διαχωρισμός των κλάσεων σε ξεχωριστά πακέτα (package). Για τους προγραμματιστές ο διαχωρισμός σε πακέτα κάνει την δημιουργία της εφαρμογής πιο καλά ιεραρχημένη, με ξεκάθαρη δομή και τους βοηθά όχι μόνο στην ανάπτυξη και την διόρθωση λαθών αλλά και στην επαναχρησιμοποίηση κομματιών κώδικα.

4.1.2 Κατάλογος res

- **res/drawable-***

Σε αυτόν τον κατάλογο αποθηκεύονται οι πόροι των γραφικών. Η οργάνωση χωρίζεται σε επιμέρους κατηγορίες όπως /res/drawable-hdpi για γραφικά που ταιριάζουν σε οθόνες με υψηλή ανάλυση, /res/drawable-mdpi με μεσαία ανάλυση και /res/drawable-ldpi για χαμηλή ανάλυση.

- **res/layout**

Αυτός ο υποκατάλογος αποτελείται από πόρους διεπαφής χρήστη. Κάθε νέα οθόνη είναι και ένα αρχείο xml με όλα τα στοιχεία ελέγχου που το αποτελούν.

- **res/values**

Τέλος, σε αυτόν τον υποκατάλογο αποθηκεύονται απλά δεδομένα όπως συμβολοσειρές, τιμές χρωμάτων .

4.1.3 Κατάλογος gen (Generated Java Files)

Όλοι οι πόροι μεταγλωττίζονται από τον μεταγλωττιστή πόρων (resource compiler). Με αυτόν τον τρόπο συμπιέζονται και πακετάρονται σε μια κλάση με το όνομα **R.java** μέσα στο φάκελο gen , ώστε να μας επιτρέπεται να έχουμε πρόσβαση στις μεταβλητές και τους πόρους μέσα από την εφαρμογή μας. Όμως είναι σημαντικό να τονιστεί πως αυτή η κλάση τροποποιείται αυτόματα από το Android plug-in του Eclipse. Κάθε φορά που κάνουμε κάποια αλλαγή στον κατάλογο res, το plug-in συγχρονίζει την R και κάνει τις απαραίτητες αλλαγές. Κάθε πόρος έχει το δικό του ξεχωριστό id το οποίο ορίζεται αυτόματα από το plug-in.

4.1.4 Κατάλογος assets

Είναι ένας φάκελος που περιέχει όλους τους πόρους της εφαρμογής που ανοίγουμε και διαβάζουμε και είναι δεδομένα που έχουν ομαδοποιηθεί σε μια ροή από bytes (πχ αρχεία raw, αρχεία κειμένου, βάση δεδομένων κλπ).

4.1.5 Androidmanifest.xml



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

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

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:label="@string/app_name"
            android:name=".TestingActivity" >
            <intent-filter >
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

Εικόνα 23:Καρτέλα επεξεργασίας AndroidManifest.xml

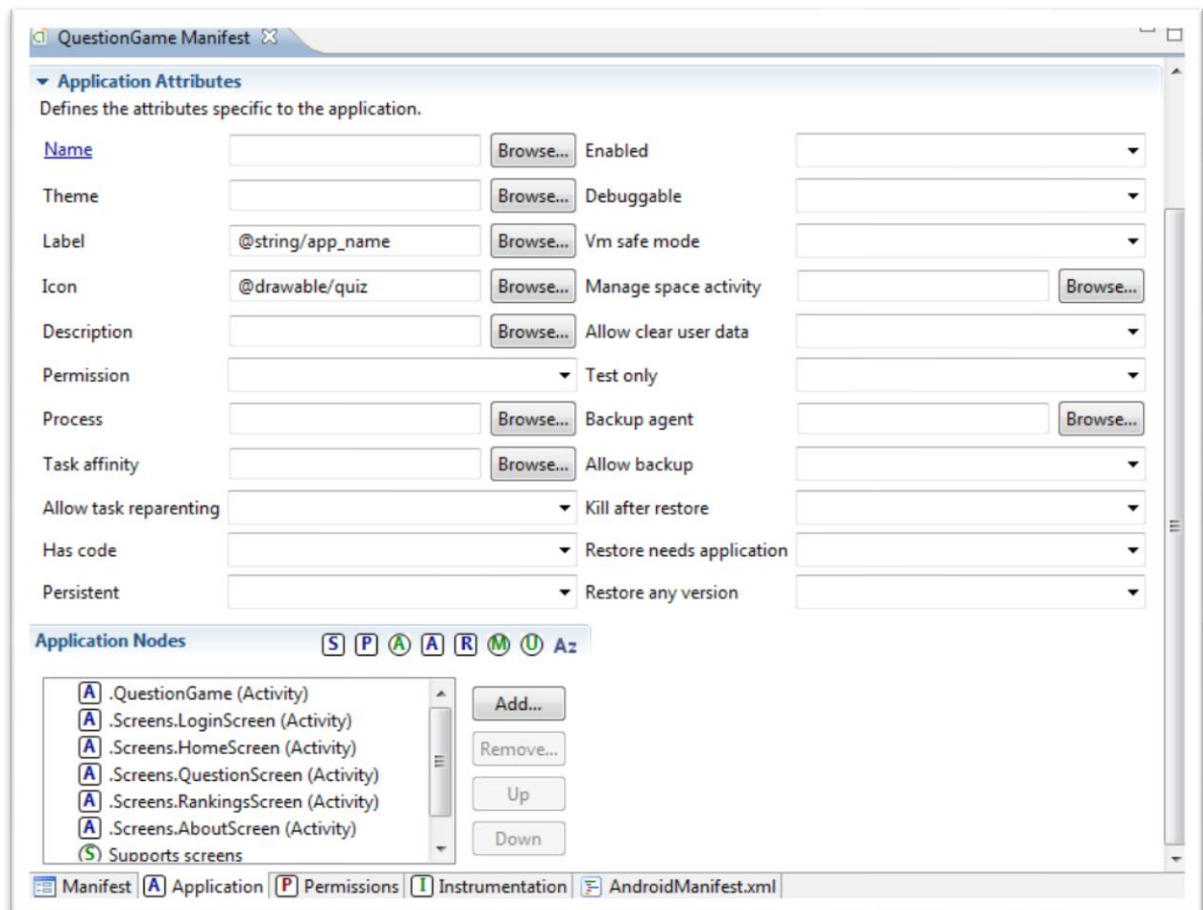
Τα έργα Android χρησιμοποιούν ένα ειδικό αρχείο ρύθμισης παραμέτρων, το οποίο ονομάζεται manifest. Τέτοιοι παράμετροι είναι το όνομα της εφαρμογής, η έκδοση και τα δικαιώματα για την εκτέλεσή της. Ιεραρχικά, το αρχείο αυτό βρίσκεται στο κορυφαίο επίπεδο του έργου, και μπορούμε να το επεξεργαστούμε χρησιμοποιώντας τον επεξεργαστή πόρων Eclipse Manifest File (πρόσθετο ADT) ή διαφορετικά τροποποιείται χειροκίνητα από τον xml editor. Κάποιες ρυθμίσεις είναι ορισμένες από το ίδιο το IDE, και ο προγραμματιστής τις προσαρμόζει όπως αυτός επιθυμεί. Ο επεξεργαστής πόρων οργανώνει τις ρυθμίσεις σε κατηγορίες, όπου κάθε καρτέλα αφορά και διαφορετικό τμήμα ρυθμίσεων:

- **Καρτέλα manifest:** αφορά το όνομα του πακέτου της εφαρμογής, τη έκδοση της, τον κωδικό της έκδοσης κλπ
- **Καρτέλα Application** (Εφαρμογή): αφορά στοιχεία της εφαρμογής όπως το εικονίδιο της, το όνομά της, μια περιγραφή της κλπ
- **Καρτέλα Permission** (Δικαιώματα): περιέχουν ρυθμίσεις που έχουν να κάνουν με κάποια απαραίτητα χαρακτηριστικά που πρέπει να διαθέτει η εφαρμογή από το λειτουργικό σύστημα που θα εγκατασταθεί, ώστε να

μπορεί να λειτουργήσει καλά όπως για παράδειγμα είναι απαραίτητο να υπάρχει πρόσβαση στο internet

- **Καρτέλα Instrumentation** (Συσκευές): παρέχει δυνατότητα για συγκεκριμένες ρυθμίσεις που αφορούν την οθόνη των συσκευών αλλά είναι διαφορετικές από αυτές που ορίζονται αυτόματα από το λειτουργικό σύστημα
- **Καρτέλα AndroidManifest.xml** : είναι το αρχείο στο οποίο ο προγραμματιστής γράφει μέσα σε ετικέτες, τον κώδικα για τις ρυθμίσεις της εφαρμογής που κατασκευάζει.

Στην παρακάτω εικόνα βλέπουμε το παράθυρο διαλόγου με το οποίο ο προγραμματιστής επιλέγοντας τον αντίστοιχο κατάλογο (Tab), καθορίζει τις δικές του ρυθμίσεις για την δική του εφαρμογή.



Εικόνα 24:Επεξεργαστής Eclipse Manifest File

ΚΕΦΑΛΑΟ 5: Ανάπτυξη Εφαρμογής Παιχνιδιού

5.1 Περιγραφή Εφαρμογής



Η εφαρμογή υποστηρίζει από ένα και πάνω παίκτες μέσω ισάριθμων φορητών συσκευών που υποστηρίζουν το λειτουργικό σύστημα ANDROID 2.2 (ή νεότερο). Ξεκινώντας η εφαρμογή εμφανίζεται ένα μενού με επιλογές (κουμπιά) : Νέο Παιχνίδι, Κατάταξη, και Σύντομη Περιγραφή. Λειτουργεί ως εξής:

Αρχικά, οι παίκτες βάζουν ένα όνομα username. Από την βάση δεδομένων εξάγονται ερωτήσεις (τυχαία επιλογή από 100 πιθανές ερωτήσεις) με 4 πιθανές απαντήσεις. Η ερώτηση μαζί με τις απαντήσεις εμφανίζεται σε κάθε φορητή συσκευή. Από την έναρξη του παιχνιδιού φαίνονται στην οθόνη οι πόντοι που έχει ο χρήστης μέχρι τώρα, ο αριθμός ερώτησης που εμφανίζεται την δεδομένη στιγμή και ο μετρητής χρόνου. Ο χρήστης μπορεί να επιλέγει μέσω της οθόνης αφής την απάντηση που θεωρεί σωστή. Μόλις επιλέξει μια απάντηση αν η απάντηση είναι σωστή ο χρήστης παίρνει 10 βαθμούς, διαφορετικά του αφαιρούνται 5 βαθμοί (όχι αρνητικό σκορ) και μπορούν να συνεχίσουν στην ίδια ερώτηση οι υπόλοιποι παίκτες. Ταυτόχρονα, υπάρχει και ένας timer όπου θα μετράει τον χρόνο απάντησης των παικτών. Όταν απαντηθεί η ερώτηση μέσα σε 10 δεύτερα και είναι σωστή τότε παίρνει συν 5 βαθμούς bonus, αν απαντήσει μέσα σε 20 δεύτερα παίρνει bonus 3 βαθμούς. Κάθε φορά που απαντάει την ερώτηση τονίζεται. Όταν τελειώσουν οι ερωτήσεις εμφανίζονται τα usernames και τα σκορ των παικτών. Επιπλέον υπάρχει η δυνατότητα εμφάνισης στατιστικών (πχ απάντησε στις 5 σωστά από τις 10, και ο μέσος χρόνος απάντησης του κάθε παίκτη). Τέλος, υπάρχει πίνακας με παλαιότερα σκορ που φαίνονται οι χρόνοι και οι πόντοι των παικτών σε κατάταξη με βάση τους πόντους τους.

5.2 Έκδοση Πλατφόρμας Εφαρμογής



Η συγκεκριμένη εφαρμογή έχει αναπτυχθεί ώστε να μπορεί να είναι συμβατή με από την έκδοση **Android 2.2** ή νεότερη. Η έκδοση 2.2 ονομάζεται «**Froyo**» κυκλοφόρησε τον Μάιο του 2010, στηριζόμενη στον πυρήνα 2.6.32 και διαθέτει τα παρακάτω:

- Βελτιστοποιημένες επιδόσεις ολόκληρου του λειτουργικού συστήματος, και ταχύτερη απόκριση της μνήμης
- Μεγαλύτερη ταχύτητα εκτέλεσης εφαρμογών από την προηγούμενη έκδοση
- Ενσωμάτωση του Chrome V8 JavaScript engine στον Browser
- Παρέχει μεγάλη ασφάλεια στο χρήστη
- USB διασύνδεση και χρήση σαν WiFi hotspot

- Μια ρύθμιση για απενεργοποίηση πρόσβασης δεδομένων μέσω δικτύου κινητής τηλεφωνίας
- Αυτόματη εύρεση και εκτέλεση ενημερώσεων εφαρμογών
- Γρήγορη εναλλαγή μεταξύ πολλών γλωσσών πληκτρολογίου και των λεξικών τους
- Φωνητική κλήση και χρήση Bluetooth
- Αποδοχή αλφαριθμητικών χαρακτήρων για χρήση κλειδώματος
- Αποδεκτή εγκατάσταση εφαρμογών από την επεκτάσιμη μνήμη
- Διαθέτει Adobe Flash 10.1

5.3 Υλοποίηση Εφαρμογής

Η παρούσα εφαρμογή αποτελείται από δυο τμήματα: το ένα τμήμα αφορά το κομμάτι που εκτελείται στον διακομιστή (**server side**) και το άλλο κομμάτι εκτελείται στην συσκευή του χρήστη (**client side**).

Λέγοντας client side εννοούμε το τμήμα της εφαρμογής που «τρέχει» στη φορητή συσκευή του χρήστη. Ουσιαστικά, είναι το βασικό κομμάτι της εφαρμογής και το μόνο που γίνεται αντιληπτό από τον χρήστη αφού έρχεται σε επαφή μόνο με αυτό. Δημιουργείται από το IDE, αποτελεί ένα project και γίνεται η εξαγωγή του σε ένα αρχείο με κατάληξη .apk, το οποίο ο χρήστης κατεβάζει και εγκαθιστά στη συσκευή.

Το σύστημα Android επιτρέπει σε μεμονωμένες εφαρμογές να έχουν ιδιωτικές βάσεις δεδομένων στις οποίες αποθηκεύουν, ανακτούν και επεξεργάζονται δεδομένα. Υπάρχουν δύο διαφορετικοί τύποι βάσεων δεδομένων που χρησιμοποιούνται για διαφορετικές ανάγκες αποθήκευσης και χρήσης δεδομένων των εφαρμογών. Ο ένας τύπος είναι το SQLite, είναι αποτελεί έναν συμπαγή, ελαφρύ και πολύ ισχυρό μηχανισμό που τον καθιστά ιδανικό για εφαρμογές κινητών τηλεφώνων. Σύμφωνα με αυτόν το μηχανισμό η βάση δεδομένων αποθηκεύεται τοπικά σε ένα αυτόνομο αρχείο τοπικά μέσα στο δίσκο. Ο δεύτερος τύπος είναι το MySQL [1], που βασίζεται σε SQL server, και συγκεκριμένα στον διακομιστή Apache[2]. Στην παρούσα εφαρμογή έχει χρησιμοποιηθεί ο δεύτερος τύπος και αυτό γιατί το παιχνίδι έχει να κάνει με μεγάλο πλήθος παικτών οι οποίοι μπορούν να παίζουν ταυτόχρονα (online) και να εμφανίζονται τα σκορ τους σύμφωνα με τις επιδόσεις τους. Όπως είναι λογικό, θα έπρεπε να υπάρχει μια βάση δεδομένων προσβάσιμη από όλους τους παίκτες ανεξάρτητα από το μέρος που βρίσκονται για αυτό το λόγο τέτοιες εφαρμογές είναι απαραίτητο να αποθηκεύουν και να ανανεώνουν τα δεδομένα τους σε βάση που βρίσκεται στον server .

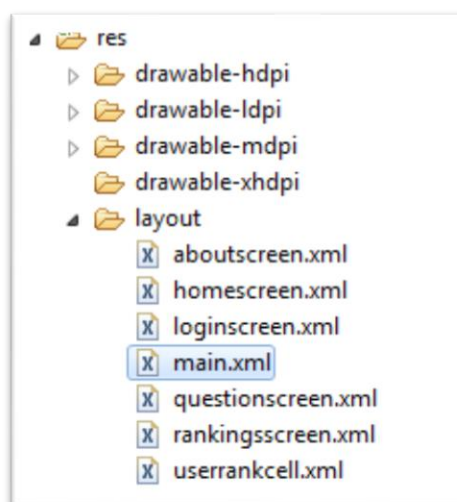
Επιπλέον, για την συγκεκριμένη εφαρμογή επιλέχθηκε για την συγγραφή κώδικα που εκτελείται στον διακομιστή, η γλώσσα Php [3], διότι είναι συμβατή με την βάση δεδομένων MySQL, είναι ανοιχτού κώδικα λογισμικό, και είναι δωρεάν η χρήση της. Σε συνδυασμό η php και η MySQL μας παρέχουν την δυνατότητα να αναπτύξουμε κώδικα σε Windows πλατφόρμα αλλά να εκτελείται και σε Linux. Ακόμη, όλα τα δεδομένα που μεταφέρονται από τον διακομιστή στην φορητή συσκευή ή και το αντίθετο είναι σε μορφή xml [4] αρχείων.

5.4 Λογισμικό που χρησιμοποιήθηκε για την κατασκευή του Client Side τμήματος

Ο κώδικας γράφηκε χρησιμοποιώντας Windows 7 με εγκατεστημένο το παρακάτω λογισμικό:

- Eclipse Java IDE Version 3.7 (Indigo)
- Java SE Development Kit (JDK) 6 update 20
- Android SDK Version 2.2, API Level 8 (FroYo)
- ADT Plug in for Eclipse (Android Development Tools)
- SDK Tools Revision 7

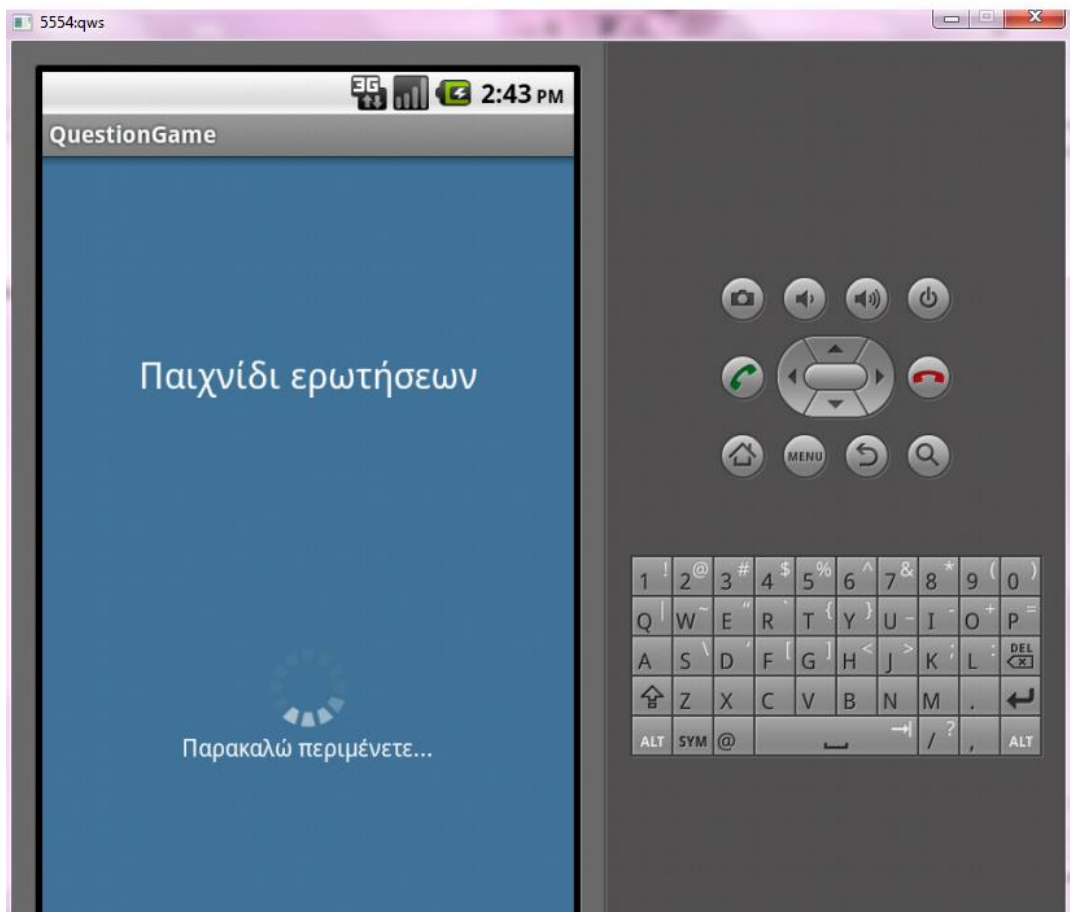
5.4.1 Αρχεία xml της Εφαρμογής στο Android Project



Εικόνα 25:Ιεραρχία καταλόγου layout του project

Η εφαρμογή του παιχνιδιού αποτελείται από συνολικά επτά αρχεία xml, όπου κάθε αρχείο αντιστοιχεί σε κάθε μια οθόνη που βλέπει ο χρήστης. Πιο συγκεκριμένα, η

πρώτη επαφή του παίκτη είναι μια οθόνη όπου γίνεται φόρτωση του παιχνιδιού.



Εικόνα 26:Αρχείο main.xml

Παρακάτω χαρακτηριστικά αποτυπώνεται ο κώδικας αυτού του αρχείου **main.xml** στον xml editor του eclipse:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/BaseView"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center_vertical"
    android:background="#ff447799">
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:orientation="vertical"
        android:gravity="center_vertical|center_horizontal">
        <TableLayout
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:stretchColumns="0"
            android:gravity="top"
            android:layout_marginBottom="20dp"
            android:layout_marginLeft="40dp"
            android:layout_marginTop="20dp"
```

```

        android:layout_marginRight="40dp"
        android:layout_gravity="center_vertical|center_horizontal">
        <TableRow android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginBottom="150dp">
            <TextView android:text="@string/app_title"
                android:textSize="22sp"
                android:textColor="#ffffff"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:layout_gravity="center_horizontal"
                android:gravity="center_horizontal" />
        </TableRow>
        <TableRow android:layout_width="wrap_content"
            android:layout_height="wrap_content">
            <ProgressBar android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center_horizontal"
                android:gravity="center_horizontal"
                android:layout_marginBottom="2dp" />
        </TableRow>
        <TableRow android:layout_width="wrap_content"
            android:layout_height="wrap_content">
            <TextView android:text="@string/str_wait"
                android:textSize="14sp"
                android:textColor="#ffffff"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:layout_gravity="center_horizontal"
                android:gravity="center_horizontal" />
        </TableRow>
    </TableLayout>
</LinearLayout>
</LinearLayout>

```

Εικόνα 27:Αρχείο main.xml

Αναλύοντας κάποια βασικά στοιχεία αυτού του αρχείου αξίζει να αναφέρουμε:

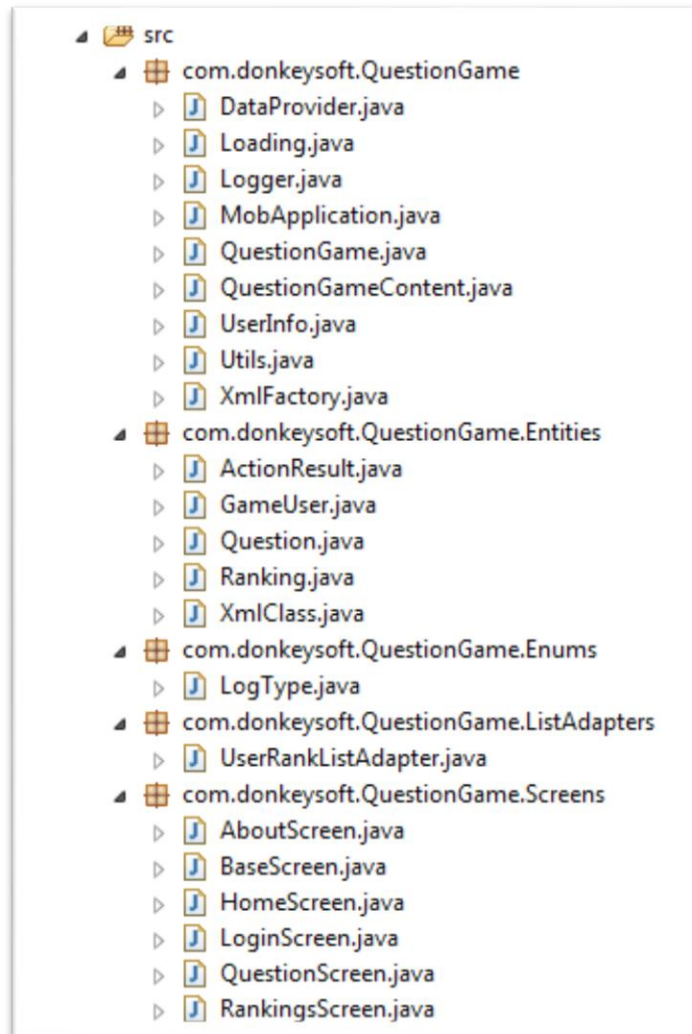
Η επικέτα **LinearLayout** αφορά το πλαίσιο (την διάταξη) που ορίζει η οθόνη. Έχει κάποιες ιδιότητες όπως έναν συγκεκριμένο κωδικό (**android:id**), προσανατολισμό (**android:orientation**), πλάτος (**android:layout_width**), ύψος (**android:layout_height**), πως στοιχίζονται τα τμήματα που περιέχει (**android:gravity**), και το χρώμα που θα έχει το φόντο (**android:background**). Μέσα σε αυτό το πλαίσιο ορίζεται ένα νέο πλαίσιο που προσαρμόζει τον προσανατολισμό ανάλογα με το πώς βλέπει την οθόνη ο χρήστης από την συσκευή του πχ οριζόντια. Τα στοιχεία ελέγχου που τοποθετούνται στο πλαίσιο μπαίνουν σε ένα πίνακα. Με αυτή τη μέθοδο, που είναι γνωστή και από την εγγραφή html κώδικα, είναι πολύ εύκολη η ακριβής και πλήρως στοιχισμένη τοποθέτηση των δύο πλαισίων κειμένου (**TextView**) και της **ProgressBar** . Σε γενικές γραμμές κάθε xml αρχείο που έχει να κάνει με την σχεδίαση της οθόνης ορίζει με παρόμοιο τρόπο την τοποθέτηση των αντικειμένων που περιέχει. Με βάση αυτού του είδους τις ιδιότητες το αποτέλεσμα ορίζεται με τον καλύτερο

τρόπο, και ελέγχεται η εικόνα (User Interface) που έρχεται σε επαφή ο χρήστης. Δεν αφήνεται κανένα περιθώριο για αυθαιρεσίες γιατί μια εφαρμογή δεν μας ενδιαφέρει να είναι μόνο λειτουργική και σωστά λογικά εκτελέσιμη χωρίς σφάλματα στον χρόνο εκτέλεσης αλλά να είναι και εύχρηστη και οπτικά προσιτή στον χρήστη. Μια καλή εφαρμογή είναι απαραίτητο να διαθέτει και ένα καλά σχεδιασμένο περιβάλλον αλληλεπίδρασης. Μια εφαρμογή με κακό σχεδιασμό και οπτικά άσχημη σχεδίαση, προκαλεί δύσκολη επικοινωνία με τον χρήστη, δεν την κάνει καθόλου άνετη στη χρήση και είναι σίγουρο ότι αποτρέπει τον χρήστη σε πολύ μεγάλο βαθμό.

Εκτός από την **main.xml** έχουν δημιουργηθεί και οι εξής:

- **loginscreen.xml**: Γίνεται η εισαγωγή του ονόματος του παίκτη
- **homescreen.xml**: Είναι η βασική οθόνη πλοήγησης στην εφαρμογή, με δυνατότητα μετάβασης του παίκτη σε νέο παιχνίδι, στην λίστα κατάταξης ή στο κείμενο με την σύντομη περιγραφή του παιχνιδιού.
- **questionscreen.xml**: Εμφανίζονται με την σειρά οι ερωτήσεις, ενώ μετράει ο χρόνος, και οι διαθέσιμοι πόντοι
- **rankingscreen.xml**: Αρχείο που κρατάει τα στατιστικά του κάθε χρήστη
- **userrankincell.xml**: Τέλος, εμφάνιση της γενικής κατάταξης όλων των παικτών με βάση τους πόντους τους

5.4.2 Αρχεία java της Εφαρμογής στο Android Project



Εικόνα 28:Ιεραρχία αρχείων java του project

Εκτός όμως από την οπτική επαφή του χρήστη με κάθε xml αρχείο ορίζεται και μία κλάση η οποία είναι υπεύθυνη για την εκτέλεση των απαραίτητων λειτουργιών. Ο χρήστης εκτελώντας την εφαρμογή, κινεί αφενός την φόρτωση του παιχνιδιού, αυτό είναι και που βλέπει στην οθόνη της συσκευής, αφετέρου η εφαρμογή προετοιμάζεται για την είσοδο του χρήστη. Η πρώτη κλάση που είναι υπεύθυνη για το φόρτωμα της εφαρμογής είναι η **QuestionGame.java**. Όμως η πιο βασική κλάση που ελέγχει όλες τις λειτουργίες της εφαρμογής είναι η **MobApplication.java** (Mobile Application). Η εφαρμογή αποτελείται από αρκετά τμήματα-κλάσεις, γι' αυτό θεώρησα καλό να δημιουργήσω επιμέρους υποκατηγορίες πακέτων, και να διαχωρίσω σύμφωνα με τις λειτουργίες που αντιπροσωπεύουν. Το έργο όπως φαίνεται στην παραπάνω εικόνα διαχωρίστηκε στα εξής πακέτα:

- **com.donkeysoft.QuestionGame**
- **com.donkeysoft.QuestionGame.Entities**

- **com.donkeysoft.QuestionGame.Enums**
- **com.QuestionGame.ListAdapters**
- **com.QuestionGame.Screens**

5.4.2.1 *To πακέτο com.donkeysoft.QuestionGame*

Οι κλάσεις που βρίσκονται στο πακέτο **com.donkeysoft.QuestionGame.Screens** συνδέονται με τις ενέργειες που απαιτούνται για κάθε μία οθόνη που βλέπει ο χρήστης.

Χαρακτηριστικό παράδειγμα είναι η παρακάτω κλάση:

```

package com.donkeysoft.QuestionGame.Screens;

import com.donkeysoft.QuestionGame.MobApplication;
import com.donkeysoft.QuestionGame.QuestionGame;
import com.donkeysoft.QuestionGame.R;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.Window;
import android.widget.Button;

public class HomeScreen extends BaseScreen
{
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.homescreen);

        if
(MobApplication.Instance().getUserInfo().getUserName().equalsIgnoreCase(""))
        {
            Intent l_intent = new Intent(this, QuestionGame.class);
            startActivity(l_intent);
            finish();
            return;
        }
        //koumbia gia neo paixnidi, katataksi kai sintomi perigrifi
        Button btn_newGame = (Button)
findViewById(R.id.ButtonHomeNewGame);
        Button btn_rankings = (Button)
findViewById(R.id.ButtonHomeRankings);
        Button btn_about = (Button) findViewById(R.id.ButtonHomeAbout);

        btn_newGame.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View arg0)
            {

```

```
        MobApplication.Instance().startNewGame();

        Intent l_intent = new Intent(HomeScreen.this,
QuestionScreen.class);
        startActivityForResult(l_intent, 0);
        finish();
    }
});

    btn_rankings.setOnClickListener(new View.OnClickListener()
    {
@Override
public void onClick(View arg0){
Intent l_intent = new Intent(HomeScreen.this, RankingsScreen.class);
startActivityForResult(l_intent, 0);
finish();
}
});

    btn_about.setOnClickListener(new View.OnClickListener()
    {
@Override
public void onClick(View arg0){
Intent l_intent = new Intent(HomeScreen.this, AboutScreen.class);
startActivityForResult(l_intent, 0);
finish();
}
});
}
}
```

Εικόνα 29: HomeScreen Java class

Η **homescreen.java** είναι μια κλάση της εφαρμογής η οποία φορτώνεται μετά την εισαγωγή του χρήστη, και έχει να κάνει κυρίως με την διαχείριση `onClick` ενεργειών του χρήστη. Για παράδειγμα το κουμπί για την έναρξη νέου παιχνιδιού είναι : `Button btn_newGame = (Button) findViewById(R.id.ButtonHomeNewGame)` Όταν επιλέγει Νέο παιχνίδι, καλείται η συνάρτηση `btn_newGame.setOnClickListener`, όπου φορτώνεται η κλάση που αντιστοιχεί στις ερωτήσεις. Με τον ίδιο τρόπο ενεργούμε και για τα άλλα δυο κουμπιά.

Παρόμοια λειτουργία με αυτή την κλάση στην εφαρμογή έχουμε ορίσει για το κάθε αρχείο `xml` και μια κλάση. Για τον ίδιο σκοπό όλες οι κλάσεις είναι:

- **AboutScreen.java:** αντιστοιχεί στο σχεδιαστικό `aboutscreens.xml`, και εμφανίζει την σύντομη περιγραφή του παιχνιδιού
- **HomeScreen.java:** αντιστοιχεί με το `homescreen.xml`, καθορίζει τις ενέργειες της εφαρμογής όταν ο χρήστης επιλέξει από την βασική οθόνη μια επιλογή (Νέο παιχνίδι, Κατάταξη ή Σύντομη περιγραφή) να φορτώσει το αντίστοιχο παράθυρο
- **LoginScreen.java:** συνδέεται με το `loginscreens.xml`. Μέσω αυτής ελέγχεται αν ο χρήστης δώσει ένα `username` για την είσοδο στην εφαρμογή και

επίσης ελέγχει και για το αν υπάρχει δυνατότητα σύνδεσης με την βάση στο διακομιστή

- **QuestionScreen.java:** συνεργάζεται με το questionscreen.xml, εμφανίζει τις ερωτήσεις με τις απαντήσεις, ελέγχει για την επιλογή του χρήστη, μετράει το χρόνο απάντησης και τον αριθμό της απαντημένης ερώτησης καθώς επίσης είναι υπεύθυνη για την εμφάνιση του παράθυρου διαλόγου με την ενημέρωση για τους πόντους που αποκτά ο παίκτης μετά από κάθε απάντηση
- **RankingsScreen.java:** κρατάει τα στατιστικά στοιχεία των χρηστών, και τα συγχρονίζει με την βάση
- **BaseScreen.java:** είναι υπεύθυνη για την αρχικοποίηση, παύση και ανάκτηση δραστηριοτήτων

5.4.2.2 *To πακέτο com.donkeysoft.QuestionGame*

Περιλαμβάνει τις κλάσεις:

- **DataProvider.java:** Κάνει την φόρτωση με υπηρεσίες από τον server σε συνεργασία με τα αρχεία rhp, ενημερώνει και για τις accept_answers
- **Loading.java:** Σχετίζεται με την φόρτωση της εφαρμογής και το main.xml
- **Logger.java:** Χρησιμοποιείται για καταγραφή σφαλμάτων
- **MobApplication.java :** Επιλογή τυχαίων δέκα ερωτήσεων, και εμφάνισή τους με την σειρά
- **QuestionGame.java:** Κάνει την εφαρμογή να επικοινωνεί με το διακομιστή, διαφορετικά εμποδίζει την εκτέλεση
- **QuestionGameContent.java:** Δημιουργεί λίστα ερωτήσεων και λίστα βαθμολογιών
- **UserInfo.java:** Προσθέτει βαθμολογία για κάθε ερώτηση στον χρήστη
- **Utils.java:** Δημιουργεί και κάνει ανάγνωση του τοπικού αρχείου questiongame.dat
- **XmlFactory.java:** Αναλαμβάνει να κάνει parse (ανάλυση) το xml που έρχεται από τον server, και να λίστα από οντότητες (Entities)

```
package com.donkeysoft.QuestionGame;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.List;
import java.util.Random;

import com.donkeysoft.QuestionGame.Entities.Question;
import com.donkeysoft.QuestionGame.Screens.BaseScreen;

import android.content.Context;
```

```

public class MobApplication
{
    private static MobApplication m_instance;

    private boolean m_initialized;
    private BaseScreen m_currentActivity;
    private QuestionGameContent m_content;
    private DataProvider m_dataProvider;
    private UserInfo m_userInfo;

    private int m_questionsPerGame;
    private int m_currentQuestionIndex;
    private int m_currentGamePoints;

    private MobApplication()
    {
        m_initialized = false;
        m_content = new QuestionGameContent();
        m_dataProvider = new DataProvider();
        m_userInfo = new UserInfo();

        m_questionsPerGame = 10;
    }

    public static MobApplication Instance()
    {
        if (m_instance == null)
            m_instance = new MobApplication();

        return m_instance;
    }

    public void initialize(Context a_context)
    {
        if (m_initialized)
            return;

        Logger.Instance().setup("questiongame.log");
        //arxeio katagrafis sfalmatwn

        m_initialized = true;
    }

    public void setCurrentActivity(BaseScreen a_activity)
    {
        m_currentActivity = a_activity;
    }

    public DataProvider getDataProvider()
    {
        return m_dataProvider;
    }

    public UserInfo getUserInfo()
    {
        return m_userInfo;
    }
}

```

```

public QuestionGameContent getContent()
{
    return m_content;
}

public BaseScreen getCurrentActivity()
{
    return m_currentActivity;
}

public int getQuestionsPerGame()
{
    return m_questionsPerGame;
}

public int getCurrentQuestionIndex()
{
    return m_currentQuestionIndex;
}

public int getCurrentGamePoints()
{
    return m_currentGamePoints;
}

public void ShowWaitingMessage(Context context)
{
    Loading.Instance().Show(context, "",
context.getResources().getString(R.string.str_wait));
}

public void HideWaitingMessage()
{
    Loading.Instance().Hide();
}

public void saveUserInfo(Context a_context)
{
    List<UserInfo> l_list = new ArrayList<UserInfo>();
    l_list.add(m_userInfo);
    Utils.saveLocalData(a_context, l_list, UserInfo.class);
}

public void loadUserInfo(Context a_context)
{
    List<UserInfo> l_lastUserInfo = Utils.loadLocalData(a_context,
UserInfo.class);
    if ((l_lastUserInfo != null) && (l_lastUserInfo.size() > 0))
        m_userInfo = l_lastUserInfo.get(0);
}

public void clearLocalUserSavedData(Context a_context)
{
    // reset user info
    getUserInfo().clearUserInfo();
    saveUserInfo(a_context);
}

public void startNewGame()

```

```

{
    m_currentQuestionIndex = 0;
    m_currentGamePoints = 0;

    // randomize questions
    Random l_random = new Random();
    for (Question l_question : m_content.getQuestions())
        l_question.setOrder(l_random.nextInt());

    Comparator<Question> l_comparator = new Comparator<Question>()
    {
        @Override
        public int compare(Question object1, Question object2)
        {
            if (object1.getOrder() > object2.getOrder())
                return 1;
            else if (object1.getOrder() < object2.getOrder())
                return -1;
            else
                return 0;
        }
    };

    Collections.sort(m_content.getQuestions(), l_comparator);
}

public void goToNextQuestion()
{
    m_currentQuestionIndex++;
}
}

```

Εικόνα 30:Κλάση Mobpplication

5.4.2.3 Το πακέτο *com.donkeysoft.QuestionGame.Entities*

Οι κλάσεις που ανήκουν στο πακέτο **com.donkeysoft.QuestionGame.Entities** (οντότητες) είναι:

- **ActionResult.java:** Ενημερώνει αν έχει ολοκληρωθεί σωστά η μεταφορά-ανάγωση των xml αρχείων από τον διακομιστή
- **GameUser.java:** Αναλαμβάνει την ανάγνωση από xml του userID, username, points του χρήστη, κληρονομεί την xmlclass
- **Question.java:** Κάνει ανάγνωση από το xml των στοιχείων questioned, description, rightAnswerId, answer1, answer2, answer3, answer4
- **Ranking.java:** Κληρονομώντας την xmlclass κάνει ανάγνωση των βαθμολογιών των χρηστών
- **XmlClass.java:** Είναι μια κλάση γενικού τύπου όλων των οντοτήτων, ώστε να μπορούν να μπουν στο xmlFactory

```

package com.donkeysoft.QuestionGame.Entities;

import java.io.Serializable;

public class XmlClass implements Serializable {
    private static final long serialVersionUID = 2331014032348830239L;

    public XmlClass()
    {
    }

    public void xSetFields(String a_fieldName, String a_value)
    {
    }
}

```

Εικόνα 31:Κλάση Xmlclass

Στην παραπάνω εικόνα φαίνεται η κλάση XmlClass, η οποία αποτελεί υλοποίηση της κλάσης του συστήματος Serializable. είναι μια κλάση που αναλύει τα xml αρχεία που έρχονται από τον διακομιστή, από την σειριακή μορφή σε μορφή κατανοητή από την εφαρμογή στην φορητή συσκευή. Αν παρατηρήσουμε όλες οι κλάσεις αυτού του πακέτου κληρονομούν(inherit) την XmlClass, γιατί όλες έχουν τον ίδιο σκοπό, την ανάγνωση αρχείων xml, και είναι τα αντικείμενα που επιστρέφει η XmlFactory.java.

5.4.2.4 *To πακέτο com.QuestionGame.ListAdapters*

Σε αυτό το πακέτο ανήκει μία κλάση η **UserRankListAdapter.java**, η οποία συμβάλει στην δημιουργία λίστας με τα στατιστικά αποτελέσματα όλων των παικτών

5.4.2.5 *To πακέτο com.donkeysoft.QuestionGame.Enums*

Αυτό το πακέτο έχει να κάνει με την καταγραφή σφαλμάτων και αποτελείται μόνο από την κλάση **LogType.java**

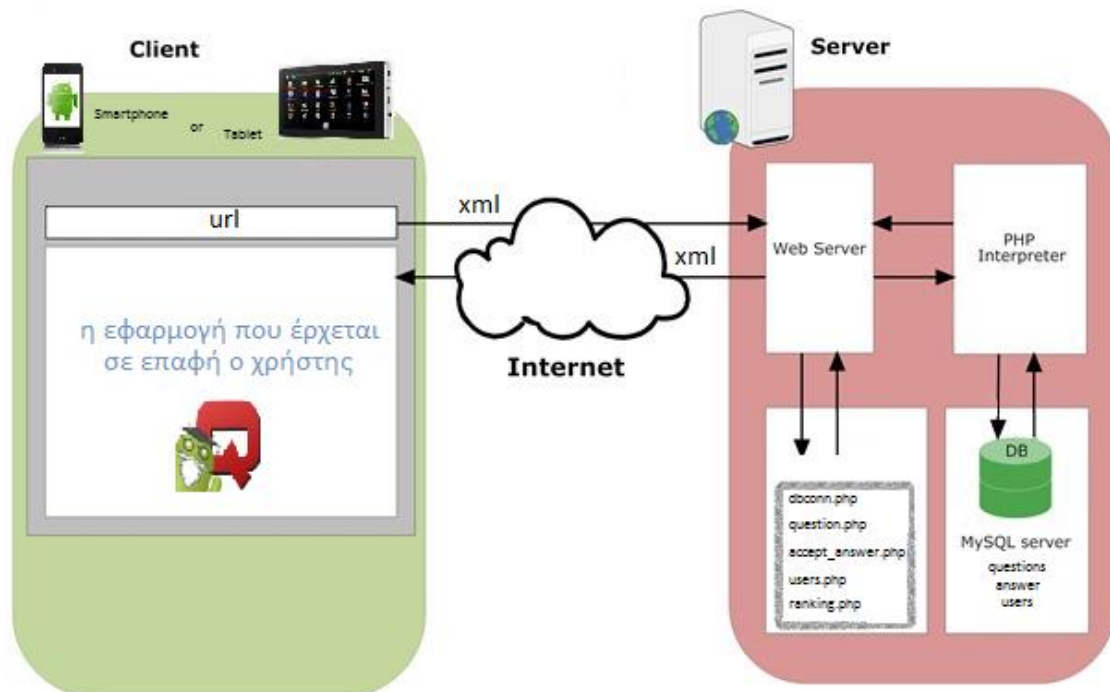
5.5 Λογισμικό που χρησιμοποιήθηκε για την κατασκευή του Server Side τμήματος

EasyPHP-5.3.8.1 το οποίο περιλαμβάνει:

- Apache/2.2.21 (Web Server)
- PHP/5.3.8 (Γλώσσα Προγραμματισμού)
- MySQL: 5.5.16 (Σύστημα Βάσης Δεδομένων)
- PHP MyAdmin Database Interface (Διασύνδεση Βάσης Δεδομένων PHP)

και το **Notepad++**: 5.9.3 (ανάπτυξη php κώδικα)

Βασική δομή υλοποίησης του τμήματος του διαδικτυακού μέρους της εφαρμογής αποτυπώνεται στο σχήμα που ακολουθεί:



Εικόνα 32: Client Server Connection

Αρχικά, ο χρήστης κάνει εγκατάσταση της εφαρμογής στην συσκευή του (smartphone ή tablet). Με το άνοιγμα της η εφαρμογή απαιτεί την είσοδο του χρήστη για να επιτύχει την σύνδεση του συγκεκριμένου χρήστη στο διακομιστή. Αυτός ο έλεγχος γίνεται μέσω του url που είναι ο συνδετικός κρίκος του παίκτη και του διακομιστή που παρέχει τα στοιχεία της εφαρμογής. Λέγοντας url, αναφερόμαστε στη διεύθυνση που μας έχει δοθεί από το server για την τοποθέτηση της βάσης δεδομένων μας. Στην περίπτωση που ο χρήστης είναι νέος, κάνει δημιουργία νέας εγγραφής στον πίνακα της βάσης με τους χρήστες

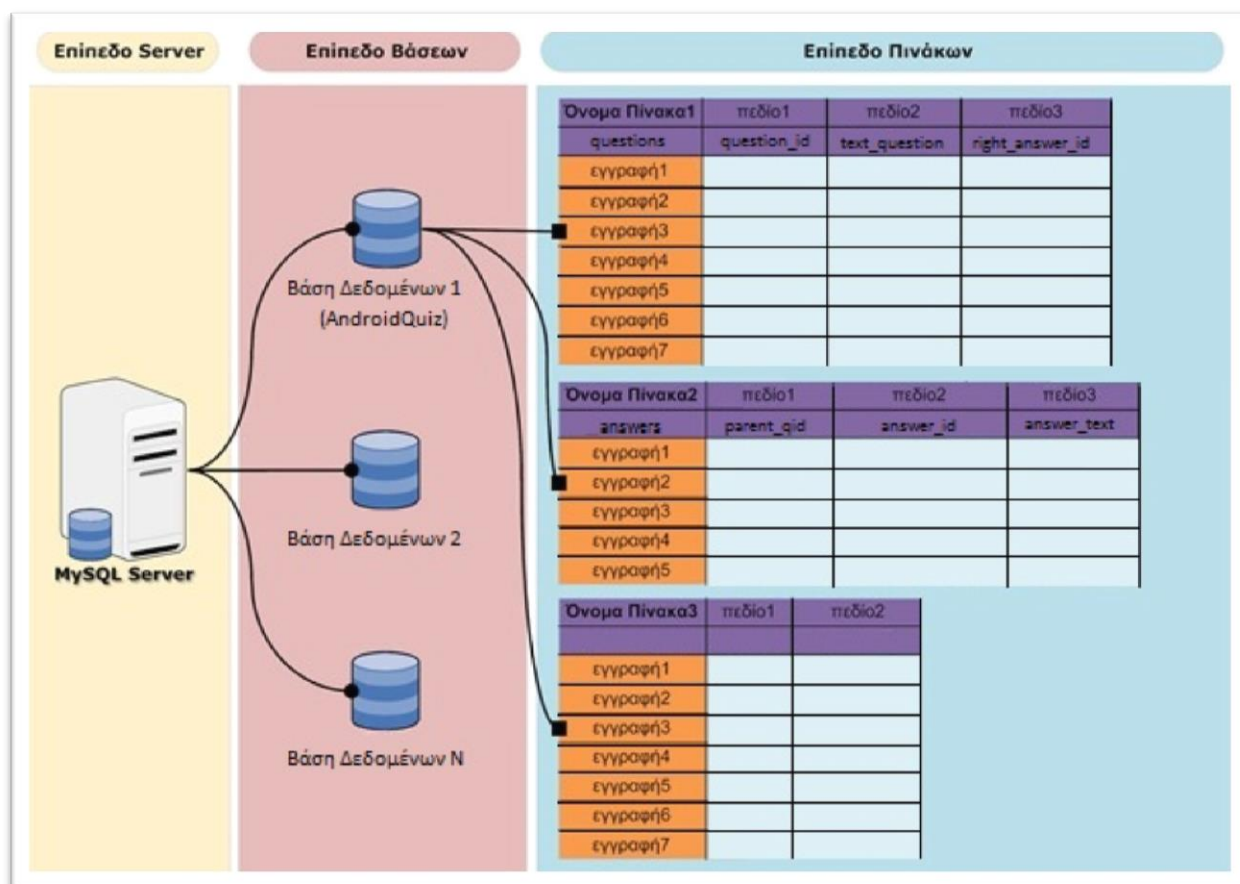
(users.php). Διαφορετικά, αναγνωρίζεται το username, και ταυτίζεται ο χρήστης με τα στατιστικά στοιχεία που δημιουργήθηκαν από προηγούμενες φορές εισόδου του στο παιχνίδι.

Όλα τα δεδομένα που μεταφέρονται από και προς τον server είναι στην μορφή xml αρχείων. Τα αρχεία xml περιέχουν πληροφορίες καλά δομημένες και η εφαρμογή τα «διαβάζει» σαν μια σειρά από stream. Επέλεξα αυτή την μορφή για τους εξής λόγους:

- τα xml αρχεία είναι πολύ εύκολο να διαβαστούν και να επεξεργαστούν λόγω της απλής μορφής τους
- είναι αρχεία ASCII, είναι συμβατά με πολυγλωσσικές εφαρμογές, επειδή δεν περιορίζονται από συγκεκριμένες γλώσσες προγραμματισμού, λειτουργικά συστήματα, πλατφόρμες ή βάσεις δεδομένων
- το πρότυπο αυτό είναι ευρέως διαδεδομένο, και υποστηρίζεται από ολόκληρη την σύγχρονη βιομηχανία λογισμικού

Το αρχείο dbconn.php κάνει την σύνδεση με την βάση, και το αρχείο users.php ελέγχει το όνομα που έχει δώσει ο χρήστης, στην περίπτωση που είναι νέος χρήστης κάνει την εγγραφή νέου user στην βάση ώστε να μπορεί να τον χειριστεί. Στην συνέχεια αναλόγως με την επιλογή του παίκτη, όταν θα διαλέξει την έναρξη νέου παιχνιδιού το αρχείο question.php κατεβάζει τυχαία 10 ερωτήσεις από την βάση (πίνακας questions) με τις πιθανές απαντήσεις που αντιστοιχούν στην ερώτηση. Όπως είναι κατανοητό η επικοινωνία της συσκευής με την βάση και τον διακομιστή είναι αρκετά συχνή. Με το αρχείο accept_answer.php ρυθμίζεται η βαθμολογία ανάλογα με την απάντηση του παίκτη. Τέλος, το αρχείο rankings.php επιστρέφει τα στατιστικά αποτελέσματα των όλων των παικτών στο σύνολο τους.

5.5.1 Σύστημα Διαχείρισης Βάσης Δεδομένων- MySQLServer



Εικόνα 33:Σύστημα Διαχείρισης Βάσης Δεδομένων MySQL Server

Στην εφαρμογή του παιχνιδιού η βάση δεδομένων ονομάζεται «androidquiz», έχει την παραπάνω δομή και αποτελείται από τους εξής πίνακες :

- **questions:** αποθηκεύεται το αναγνωριστικό κάθε ερώτησης(question_id), το κείμενο της κάθε ερώτησης (text_question) και ποια είναι η σωστή απάντηση (right_answer_id)
- **answers:** αυτός ο πίνακας αποτυπώνει τις πιθανές απαντήσεις που αντιστοιχούν σε κάθε ερώτηση, πιο συγκεκριμένα τα πεδία είναι το αναγνωριστικό της κάθε ερώτησης (parent_qid), αναγνωριστικό απάντησης (answer_id), κείμενο της απάντησης (answer_text)
- **users :** καταγράφονται όλοι οι χρήστες της εφαρμογής καθένας με το δικό του αναγνωριστικό (id), όνομα χρήστη(username), πόντοι (points), σωστές απαντήσεις(right_answers), συνολικές προσπάθειες (total_attempts), συνολικός χρόνος (totaltime)

Οι πίνακες questions και answers έχουν δεδομένα τα οποία δεν τροποποιούνται κατά την διάρκεια της εφαρμογής για αυτό το λόγο επέλεξα να τους δημιουργήσω με την βοήθεια του **EasyPhp** και του **PhpMyAdmin**. Επίσης, επέλεξα να τα χωρίσω με αυτόν τον τρόπο ώστε τα αρχεία xml που μεταφέρονται και σχετίζονται

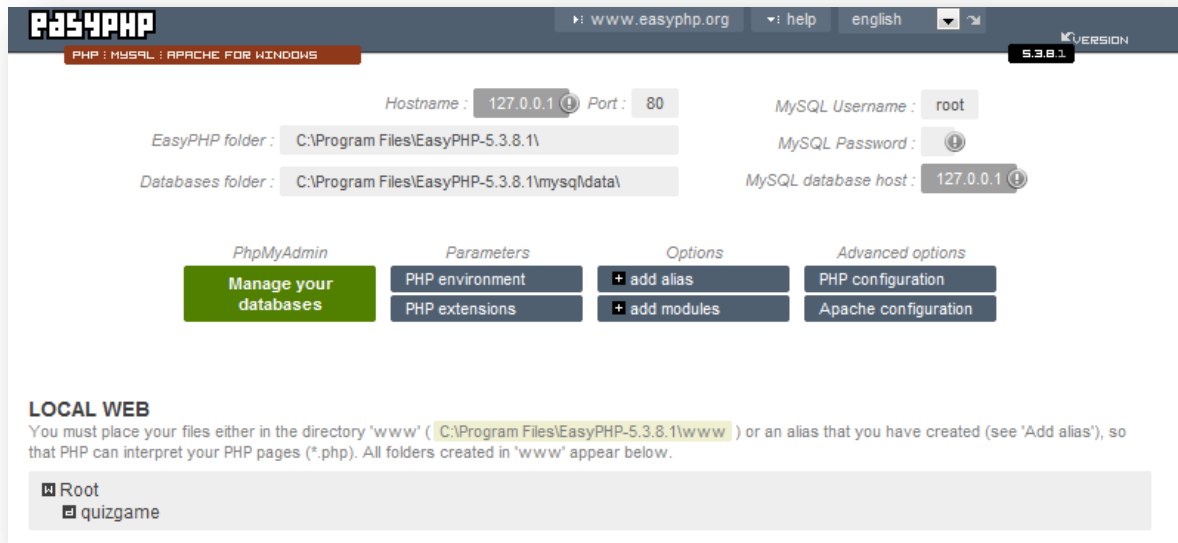
άμεσα με τις πληροφορίες και τα δεδομένα της βάσης, να είναι σε απλή μορφή και σχετικά να διαθέτουν σύντομο περιεχόμενο.

Στην παρακάτω εικόνα παρουσιάζεται η δομή και ένα τμήμα από τα δεδομένα του αρχείου Excel με τα στοιχεία του πίνακα ερωτήσεων.

	A	B	C
1	question_id	text_question	right_answer_id
2	1	Τι ύψος έχει ο Όλυμπος	3
3	2	Το μεγαλύτερο σε μήκος ποτάμι είναι	1
4	3	Ποιο νησί δεν ανήκει στα Δωδεκάνησα	1
5	4	Πόσοι νομοί υπάρχουν στην Ελλάδα	3
6	5	Η μεγαλύτερη λίμνη της Ελλάδας είναι	1
7	6	Το μεγαλύτερο νησί στην Ελλάδα είναι	1
8	7	Η βιομηχανική επανάσταση ξεκίνησε από	2
9	8	Η έξοδος του Μεσολογγίου έγινε το	2
10	9	Η απελευθέρωση των Ιωαννίνων έγινε	3
11	10	Η Χούντα των Συνταγματαρχών έληξε	3
12	11	Πότε έληξε ο Β Παγκόσμιος Πόλεμος	3
13	12	Πότε γκρεμίστηκε το τείχος του Βερολίνου	3
14	13	Μεταξύ ποιών ξέσπασε ο Ψυχρός Πόλεμος	3
15	14	Η σφαγή των Καλαβρύτων έγινε το	3
16	15	Τα ηχητικά κύματα δε διαδίδονται	4
17	16	Πότε πάτησε ο πρώτος άνθρωπος στο φεγγάρι	3
18	17	Φυσικός του 17ου και 18ου αιώνα δεν ήταν	4
19	18	Ποιός ποταμός διασχίζει το Παρίσι	1
20	19	Περισσότερες ετήσιες ημέρες ηλιοφάνειας έχει	2
21	20	Η πιο πυκνοκατοικημένη πόλη του κόσμου είναι	3
22	21	Σε μία σφαίρα διπλασιάζουμε την ακτίνα της	4
23	22	Η κομποστοποίηση είναι η ανακύκλωση	2
24	23	Ποιό υλικό ονομάζεται Τράπεζα Ενέργειας	1
25	24	Το μεγαλύτερο νησί της Μεσογείου είναι	3

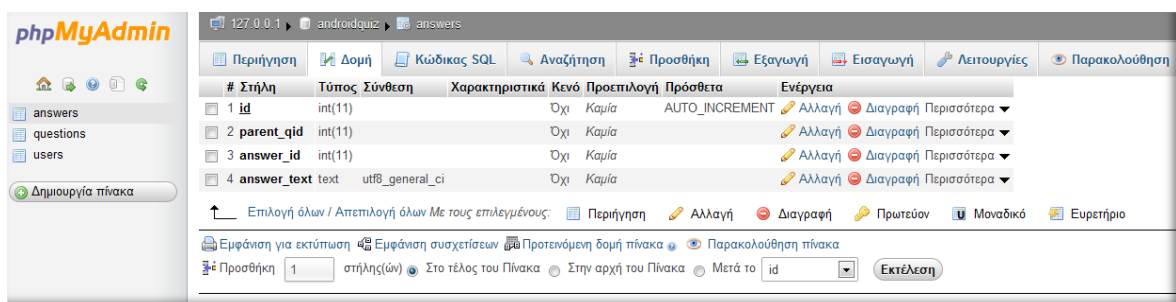
Εικόνα 34:Πίνακας Ερωτήσεων

Στην συνέχεια αφού δημιουργήσαμε τα δεδομένα των πινάκων, πάμε με την επιλογή «**Manage your Databases**» διαχειριζόμαστε αρχικά μια βάση δεδομένων η οποία αποθηκεύεται τοπικά στον σκληρό μας δίσκο.



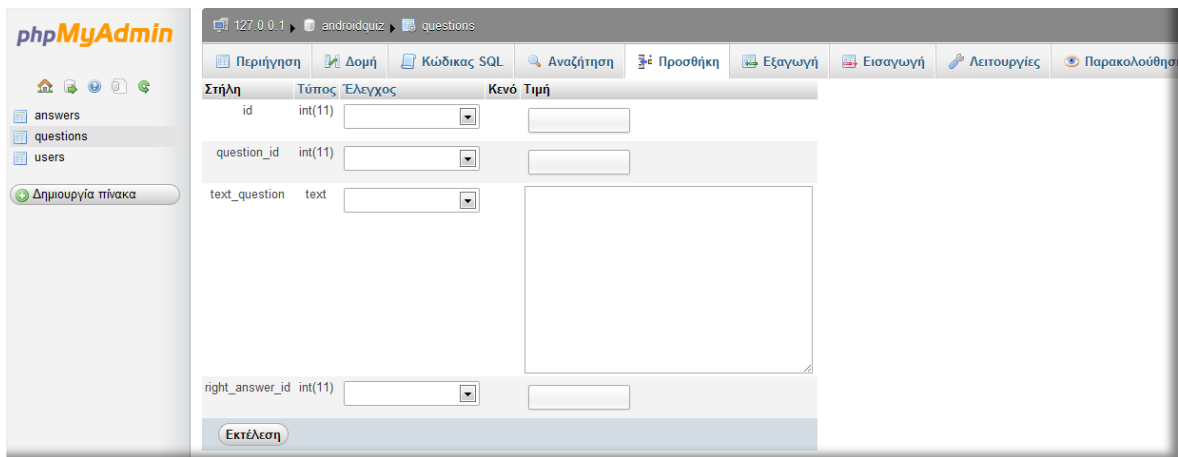
Εικόνα 35: Διαχείριση Τοπικής Βάσης Δεδομένων

Η διεύθυνση 127.0.0.1 είναι εικονική, και είναι πάντα ίδια επιλεγμένη σαν αρχική τιμή ορισμένη από την εφαρμογή EasyPhp. Με την επιλογή «Δημιουργία πίνακα» ορίζουμε το όνομα και τα πεδία του κάθε πίνακα. Στην εικόνα που ακολουθεί φαίνεται η δομή του πίνακα answers.



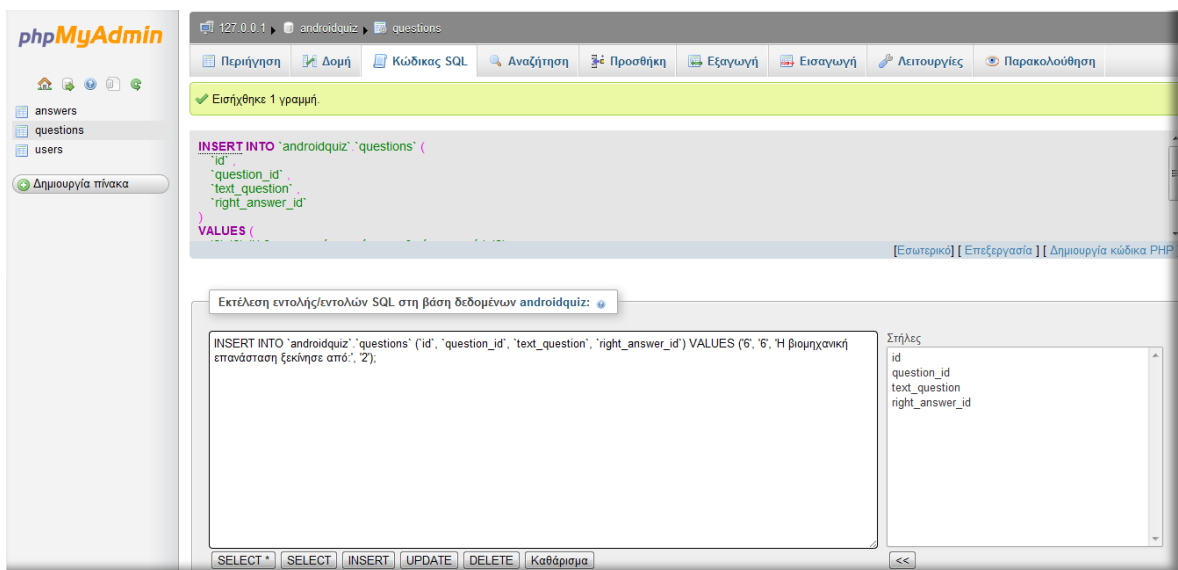
Εικόνα 36: Δομή του πίνακα answers

Στην δική μας βάση χρειαζόμαστε τρεις πίνακες. Αφού λοιπόν δηλώσουμε τα παραπάνω χαρακτηριστικά για τους πίνακες το επόμενο βήμα να γίνει η εισαγωγή των δεδομένων. Επιλέγοντας τον πίνακα που θέλουμε με την επιλογή «Προσθήκη», κάνουμε την εισαγωγή των δεδομένων και πατάμε «Εκτέλεση».



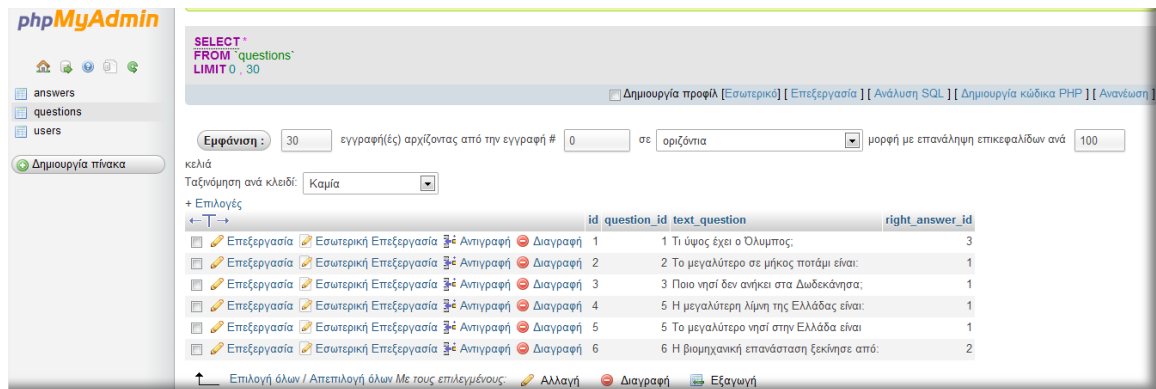
Εικόνα 37: Προσθήκη μιας εγγραφής σε πίνακα

Με αυτόν τον τρόπο γίνεται η χειροκίνητη εισαγωγή, μίας μίας της κάθε εγγραφής. Στην επόμενη εικόνα φαίνεται ο sql κώδικας που εκτελείται για την δημιουργία των εγγραφών. Η πάνω εικόνα δείχνει μια εύκολη τυπική διαδικασία, συμπληρώνοντας πεδία, σχεδιασμένη για τον κάθε προγραμματιστή σε πολύ απλή μορφή, όμως στην εικόνα που ακολουθεί παρουσιάζεται ο κώδικας που τρέχει στο background για την εκτέλεση της κάθε εισαγωγής.



Εικόνα 38: MySql insert into questions

Όταν ολοκληρωθεί η διαδικασία (στην δική μας περίπτωση 100 εγγραφές για τους δύο πίνακες), ο κάθε πίνακας έχει την εξής μορφή:



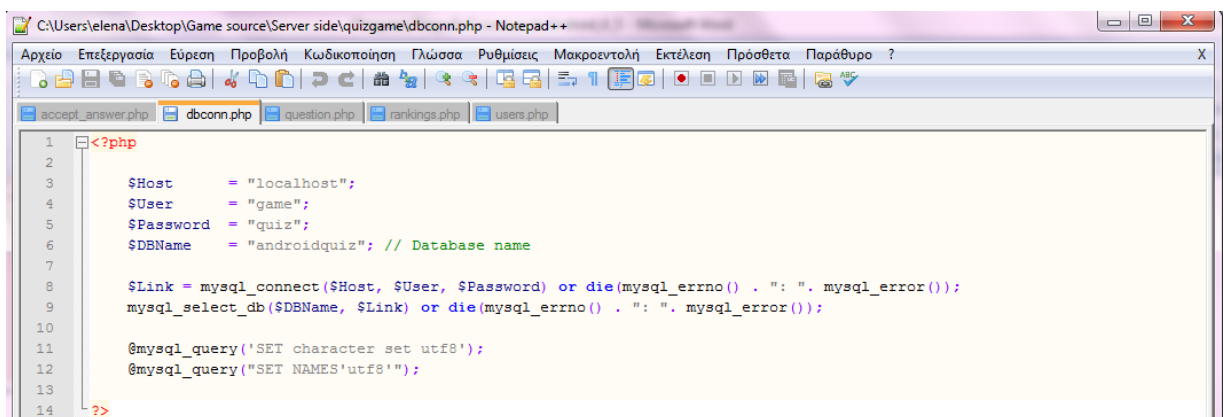
Εικόνα 39:Εγγραφές του πίνακα Ερωτήσεων

Σε αυτό το στάδιο η πολλή δουλειά έχει γίνει, σε περίπτωση που θέλουμε να τροποποιήσουμε το περιεχόμενο κάποιου πεδίου, απλά το επιλέγουμε και πατάμε «επεξεργασία». Τέλος, κάνουμε μια εξαγωγή σε ένα αρχείο της μορφής gzip, το οποίο είναι πια σε μορφή επιπέδου κατανοητού από τον διακομιστή, και πάλι χρησιμοποιώντας το πρόγραμμα PhpMyAdmin το κάνουμε την εισαγωγή στον Server.

5.5.2 Αρχεία Php

Τα αρχεία php χρησιμοποιούνται στον προγραμματισμό του server side κομματιού, και πιο συγκεκριμένα είναι αυτά που μας δίνουν πρόσβαση στην βάση δεδομένων. Σε αυτά τα αρχεία γράφονται τα ερωτήματα σε mysql που μας παρέχουν εισαγωγή, επεξεργασία και ανάκτηση των δεδομένων της βάσης. Μέσω αυτών τα στοιχεία μετατρέπονται σε xml ώστε να γίνουν μια μορφή κατανοητή για την εφαρμογή μας.

- **dbconn.php:** Κάνει την σύνδεση με την βάση, δίνοντας τα απαραίτητα αναγνωριστικά για να αποκτήσουμε πρόσβαση όπως πχ το όνομα της βάσης, το username, και τον κωδικό.



Εικόνα 40:Αρχείο dbconn.php

- **question.php:** Αφού συνδεθεί με την βάση, κατεβάζει την εκφώνηση των ερωτήσεων και τις πιθανές απαντήσεις.

Παρακάτω δίνεται μια εικόνα με τον κώδικα γραμμένο σε php του αρχείου question.php:

```

<?php
include("dbconn.php"); //sundesimevasi

$get_que = "SELECT * //erotimasemysql
            FROM questions
            ORDER BY id ASC";

$result = mysql_query($get_que) or die (mysql_error()); // apotelesmata
echo header("Content-Type: application/xml");
$data = "<?xml version='1.0' encoding='UTF-8' ?>\n";
$data .= "<questions>";
while($record = mysql_fetch_array($result)){
    extract($record);
    $data .= "<thequestion>";

        $data .= "<qid>";
        $data .= $question_id;
        $data .= "</qid>";
        $data .= "<Description>";
        $data .= $text_question;
        $data .= "</Description>";
        $data .= "<rightanswerid>";
        $data .= $right_answer_id;
        $data .= "</rightanswerid>";

    $get_answers = "SELECT * //erotimasemysql
                   FROM answers
                   WHERE parent_qid=$question_id
                   ORDER BY answer_id ASC";

    $i=0;
        $result_answers = mysql_query($get_answers) or die (mysql_error());
        //apotelesmata
        while($answers_record = mysql_fetch_array($result_answers)){

            extract($answers_record);
            $i+=1;
                $data .= "<answerid".$i.">";
                $data .= $answer_id;
                $data .= "</answerid".$i.">";
                $data .= "<answertext".$i.">";
                $data .= $answer_text;
                $data .= "</answertext".$i.">";
            }
        $data .= "</thequestion>";
    }
    $data .= "</questions>";
}

```

```
echo $data;
mysql_close($Link); //aposundesl apo tin vasi
?>
```

Εικόνα 41:Αρχείο question.php

Στην αρχή του αρχείου βλέπουμε πως κάνουμε include το αρχείο dbconn.php, με αυτή την εντολή συμπεριλαμβάνουμε στο αρχείο μας το αρχείο που είναι υπεύθυνο για την σύνδεση στην βάση. Μόνο μέσω αυτού μπορούμε να δούμε και να επεξεργαστούμε τα δεδομένα. Αυτό συμβαίνει για να αποφευχθούν εισβολές και τροποποιήσεις των στοιχείων από άτομα που δεν έχουν εξουσιοδότηση. Στην συνέχεια η μεταβλητή **\$get_que** ορίζει ένα mysql ερώτημα στην βάση. Τα αποτελέσματα από την εκτέλεση εκχωρούνται στην μεταβλητή **\$result**. Με την βοήθεια της μεταβλητής **\$data** όσο υπάρχουν αποτελέσματα, δημιουργούμε xml αρχείο, με τις ετικέτες :

```
<thequestion>
  <qid>
    question_id
  </qid>
  <Description>
    text_question
  </Description>
  <rightanswerid>
    right_answer_id
  </rightanswerid>
</thequestion>
```

Εικόνα 42:Μορφή xml αρχείου α' φάση

Αντίστοιχα η μεταβλητή **\$get_answers** δέχεται τις πιθανές απαντήσεις κάθε ερώτησης και τις ενσωματώνει στο προηγούμενο xml αρχείο. Η τελική μορφή του αρχείου xml είναι η εξής:

```
<thequestion> // κάθε ερώτηση
  <qid>question_id</qid>
  <Description> text_question</Description>
  <rightanswerid> right_answer_id</rightanswerid>
  </answer1>answer_id</answer1>
  <answertext>answer_text<answertext>
  </answer2>answer_id</answer2>
  <answertext>answer_text<answertext>
  </answer3> answer_id</answer3>
  <answertext> answer_text<answertext>
  </answer4> answer_id</answer4>
  <answertext> answer_text<answertext>
</thequestion>
```

Εικόνα 43:Τελική μορφή του thequestion xml αρχείου

Κάθε ερώτηση είναι ένα τέτοιου τύπου αρχείο. Φτάνει από τον server μέσω υπηρεσίας διαδικτύου στην κινητή συσκευή σαν xml, και αναλύεται από την κλάση DataProvider.java της εφαρμογής. Έχουμε ακόμη τα παρακάτω αρχεία php:

- **accept_answer.php:** Μετά την απάντηση που έδωσε ο χρήστης είναι υπεύθυνο να προσαρμοστεί ο αριθμός των πόντων του, και να γίνει κατανοητό αν απάντησε σωστά να προστεθούν οι βαθμοί, διαφορετικά αφού ελέγξει να μην γίνει το score αρνητικός αριθμός, αφαιρεί τους κατάλληλους πόντους.
- **users.php:** Με αυτό το αρχείο καθορίζεται οι λίστα με τους εγγεγραμμένους χρήστες και τους πόντους που έχει συγκεντρώσει ο καθένας.
- **rankings.php:** Συσχετίζονται οι χρήστες με τους πόντους, πόσες ερωτήσεις απάντησε ο καθένας και με πόσο μέσο χρόνο απάντησε σε κάθε ερώτηση. Με αυτόν τον τρόπο είναι σαν να δημιουργείται προφίλ για κάθε παίκτη ξεχωριστά.

Ας μελετήσουμε και το αρχείο users.php :

```
<?php
echo header("Content-Type: application/xml");
$data = "<?xml version='1.0' encoding='UTF-8' ?>\n";

include('dbconn.php');

$username=$_GET["username"];

$sql_find_user="SELECT * FROM users WHERE username='".$username."'";
$result = mysql_query($sql_find_user) or die (mysql_error());
if (mysql_num_rows($result) > 0) {
    $record = mysql_fetch_array($result);

    $data .= "<user>";
    extract($record);

        $data .= "\n<username>$username\n</username>";
        $data .= "\n\t<userid>$id</userid>";
        $data .= "\n\t<points>$points</points>";
        $data .= "\n</user>";
        echo $data;
    }

else //an den yparxei o xrhsths tote prepei na ton kataxwrhsoume

{
    $sql_new_user= "INSERT INTO users
                    (username,points,right_answers,total_attempts,totaltime)
                    VALUES
                    ('".$username."',0,0,0,0)";
```

```

mysql_query($sql_new_user);

$sql_find_user="SELECT * FROM users WHERE id=".mysql_insert_id()."";
//epistrefoume ta stoixeia tou neou xrhsth
$result = mysql_query($sql_find_user) or die (mysql_error());
$record = mysql_fetch_array($result);

$data .= "<user>";
extract($record);

$data .= "\n<username>$username\n</username>";
$data .= "\n\t<userid>$id</userid>";
$data .= "\n\t<points>$points</points>";
$data .= "\n</user>";
echo $data;
}

mysql_close($Link);
?>

```

Εικόνα 44:Αρχείο users.php

Όπως και πριν αφού γίνει η σύνδεση με την βάση (include'dbconn.php'), δημιουργούμε ερώτημα στην βάση, το οποίο το εκχωρούμε σε μια μεταβλητή με το όνομα \$sql_find_user. Σύμφωνα με το uname, ψάχνει στον πίνακα users αν υπάρχει ο χρήστης, διαφορετικά κάνει εγγραφή αρχικοποιώντας με μηδενικές τιμές τα πεδία points,right_answers,total_attempts, totaltime.

Συμπεράσματα και Μελλοντικές Επεκτάσεις

Στην παρούσα πτυχιακή εργασία ασχολήθηκα με την ανάπτυξη μιας εφαρμογής για φορητές συσκευές με λειτουργικό σύστημα Android. Καταρχήν, επέλεξα να δημιουργήσω ένα παιχνίδι με την προϋπόθεση πως όλοι κάποια στιγμή έχουμε δοκιμάσει να παίξουμε παιχνίδια αφιερώνοντας λίγο χρόνο σε αυτά, έστω και από περιέργεια. Είναι μια σύντομη και ευχάριστη ενασχόληση. Όμως στο μυαλό μου δεν είχα μόνο το κομμάτι της διασκέδασης αλλά ήθελα να δημιουργήσω παροχή γνώσεων με διασκεδαστικό και σύντομο χαρακτήρα. Σκέφτηκα πως πιθανόν κάποιοι χρήστες, ανεξαρτήτως ηλικίας θα ήθελαν να δοκιμάσουν τις γνώσεις τους. Κάποιοι θα προχωρούσαν πιο πολύ ώστε να μάθουν τις σωστές απαντήσεις και να απαντούν εύκολα και γρήγορα με στόχο να κατακτήσουν τις πρώτες θέσεις στα τελικά αποτελέσματα βαθμολογιών. Με αυτή την συναγωνιστική διαδικασία, χωρίς να το καταλάβουν μπαίνουν σε μια διαδικασία απόκτησης γνώσεων. Αυτή ήταν η πρώτη ιδέα που με ώθησε στην κατασκευή εφαρμογής quiz test.

Μετά από την ενασχόλησή μου με αυτό το αντικείμενο, κατάλαβα πως το Android SDK είναι ένα εργαλείο που έχει δημιουργηθεί με γνώμονα την δημιουργία καλά δομημένων, ισχυρών και ευέλικτων εφαρμογών, όσο και ολόκληρη την διαδικασία σχεδίασης και προγραμματισμού της εφαρμογής από την οπτική των developers. Χρησιμοποιεί την Java, μια γλώσσα ιδιαίτερα γνωστή, και τις δίνει νέα χαρακτηριστικά με τις βιβλιοθήκες του sdk. Χάρη στο κέλυφος Linux, που διαχειρίζεται τις βασικές υπηρεσίες συστήματος, διαχωρίζει το φυσικό υλικό από την στοίβα λογισμικού του Android επιτυγχάνοντας επιβολή δικαιωμάτων χρήσης και ασφάλεια εφαρμογών. Κάτι που είναι πολύ σημαντικό είναι πως πολλοί κατασκευαστές έχουν εμπιστευτεί το Android και δημιουργούν όλο και περισσότερα μοντέλα με αυτό το λειτουργικό. Φαίνεται πως αυτή η πλατφόρμα θα είναι η πλατφόρμα του μέλλοντος αναβαθμίζοντας συνεχώς τις εκδόσεις χαράζει νέους δρόμους στον χώρο της τεχνολογίας φορητών συσκευών.

Μια επέκταση που θα μπορούσε να προστεθεί στην εφαρμογή, θα ήταν ο διαχωρισμός των ερωτήσεων σε κατηγορίες. Με αυτόν τον τρόπο, οι παίκτες θα μπορούσαν να απαντούν και να μαθαίνουν ξεχωριστές ενότητες σύμφωνα με τα προσωπικά τους ιδιαίτερα χαρακτηριστικά όπως για παράδειγμα κάποιος που είναι καλός στην γεωγραφία να επιλέγει να διαγωνίζεται μόνο σε αυτό το επίπεδο. Με άλλα λόγια να διαχωριστούν οι ερωτήσεις σε επιστήμες (πχ Φυσική, Μαθηματικά, Χημεία κλπ) αλλά και σε επίπεδα ανάλογα με το γνωστικό επίπεδο που διαθέτει.

Ένας άλλος τρόπος προσέγγισης θα ήταν να δημιουργηθεί μια επέκταση η οποία θα δίνει έμφαση στα γραφικά της εφαρμογής, και να τεστάρει την ταχύτητα απόκρισης των παικτών. Ή ένα τμήμα του παιχνιδιού να περιλαμβάνει απλές ερωτήσεις αντίληψης, που θα απευθύνονται σε παιδιά μικρής ηλικίας. Θεωρώ πως

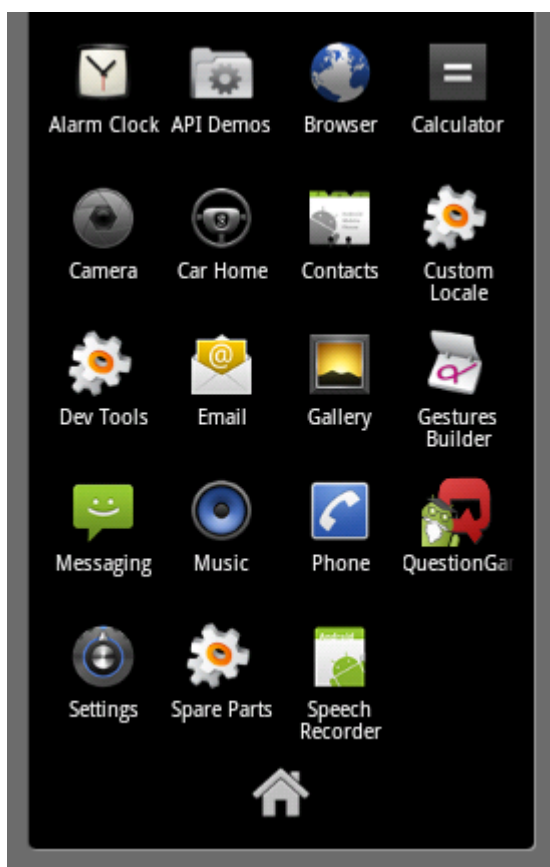
με μία τέτοια εφαρμογή θα προσελκύαμε μια μεγάλη ηλικιακά κατηγορία ανθρώπων με κοινό παρονομαστή την μάθηση. Με την ίδια εφαρμογή θα μπορούσε να περνάει ευχάριστα την ώρα του τεστάροντας τις γνώσεις του ένας ενήλικας, ενώ ταυτόχρονα θα μπορούσε να ακονίζει το μυαλό του και ένα μικρό παιδί. Ακόμη, μια καλή ιδέα θα ήταν ο κάθε χρήστης μελλοντικά να μπορεί να δημιουργήσει ένα προφίλ, βάζοντας προσωπικά χαρακτηριστικά (πχ κάποια εικόνα, ενδιαφέροντα, πόλη κατοικίας κλπ).

Τέλος, η έρευνα που πραγματοποίησα για να αναπτύξω αυτή την εφαρμογή με έκανε να καταλάβω το πώς λειτουργεί η πλατφόρμα Android, πως ανταλλάσσονται δεδομένα από τον server στις συσκευές μας . Μέσα από αυτή μου δόθηκε η ευκαιρία να δω πώς λειτουργεί και συμπεριφέρεται μία πραγματική βάση δεδομένων και να ανακαλύψω πόσες ενέργειες γίνονται στο background των εφαρμογών που ο χρήστης δεν μπορεί να αντιληφθεί.

Οδηγός Χρήσης Λογισμικού

Το συγκεκριμένο λογισμικό απευθύνεται σε όλες τις ηλικίες. Πρόκειται για μία εφαρμογή με το όνομα **QuestionGame** που προορίζεται για φορητές συσκευές που υποστηρίζουν το λειτουργικό σύστημα Android 2.2 ή μεταγενέστερο. Το QuestionGame είναι ένα παιχνίδι με ερωτήσεις γενικού περιεχομένου που υποστηρίζει ταυτόχρονα πολλούς παίκτες. Ο στόχος του είναι οι παίκτες να συγκεντρώνουν μεγάλο αριθμό πόντων με όσο το δυνατόν ταχύτερο χρόνο απάντησης. Διαθέτει ένα απλό και ευχάριστο περιβάλλον χωρίς υπερβολές, που δίνει στόχο στην γνώση και τον ευγενή συναγωνισμό.

Το πρώτο βήμα είναι η εγκατάσταση της εφαρμογής, κατά την οποία δημιουργείται το αντίστοιχο εικονίδιο στο κεντρικό μενού της συσκευής. Αυτό δείχνει ότι η εγκατάσταση ολοκληρώθηκε με επιτυχία και είμαστε έτοιμοι για την χρήση της.



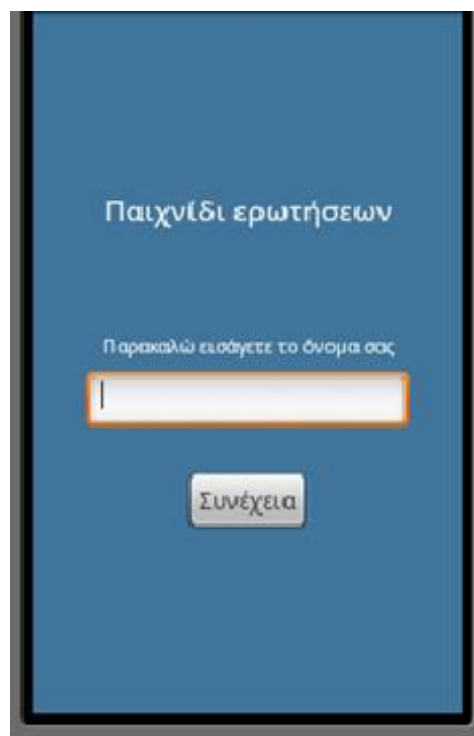
Εικόνα 45: Δημιουργία εικονιδίου στην κύρια οθόνη

Κάνοντας διπλό κλικ, φορτώνεται η εφαρμογή μας όπως φαίνεται στην παρακάτω εικόνα:



Εικόνα 46:Φόρτωση Εφαρμογής

Κατά την έναρξη σας ζητείται να πληκτρολογήσετε ένα όνομα χρήστη ώστε να μπορούμε να κάνουμε ένα προφίλ χρήστη που μας ξεχωρίζει από τους υπόλοιπους παίκτες. Και μετέπειτα πατήστε στο πλήκτρο «**Συνέχεια**».

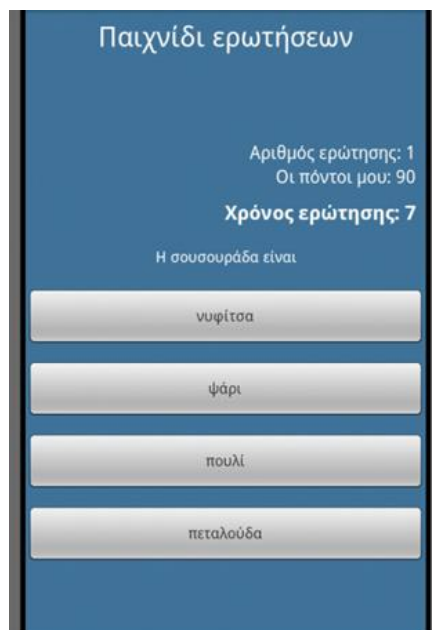


Εικόνα 47:Εισαγωγή Παίκτη



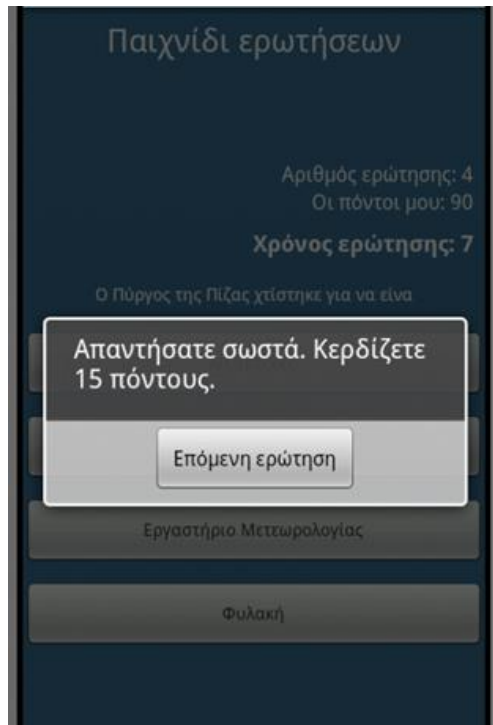
Εικόνα 48:Κύριο Μενού

Στην παραπάνω εικόνα φαίνεται το Κεντρικό Μενού της εφαρμογής. Με την επιλογή «**Νέου Παιχνιδιού**» το παιχνίδι αρχίζει, η πρώτη ερώτηση εμφανίζεται. Κάθε ερώτηση έχει τέσσερις πιθανές απαντήσεις που εμφανίζονται ταυτόχρονα, επιπλέον ο χρόνος απάντησης αρχίζει να μετρά και ο αριθμός ερώτησης, και οι πόντοι εμφανίζονται στο πάνω δεξιά κομμάτι της οθόνης. Κάθε φορά που επιλέγεται νέο παιχνίδι, τυχαίες δέκα ερωτήσεις συνολικά θα εμφανιστούν με τυχαία σειρά προς απάντηση. Μια χαρακτηριστική εικόνα είναι η παρακάτω:

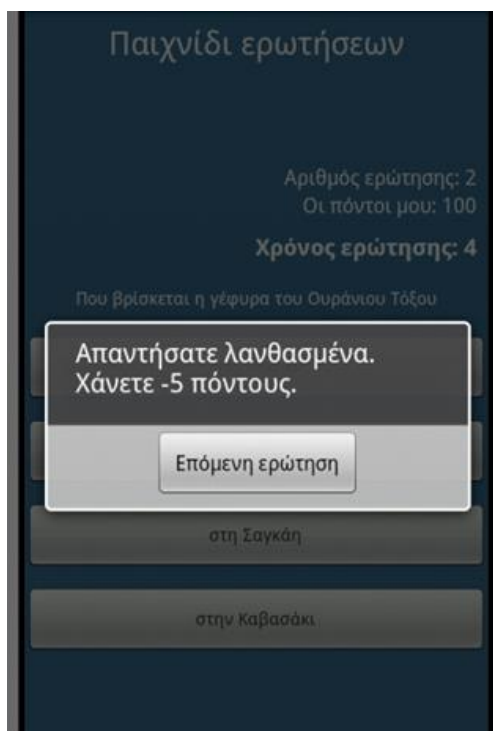


Εικόνα 49:Οθόνη με των ερωτήσεων

Ανάλογα με τον χρόνο απάντησης κερδίζετε ή χάνετε ένα αριθμό πόντων. Συγκεκριμένα, αν απαντήσει ο παίκτης σωστά μέσα σε δέκα δευτερόλεπτα κερδίζει 15 πόντους, σε 20 δευτερόλεπτα 13 πόντους, αν απαντήσει σωστά μετά από 21 δευτερόλεπτα τότε παίρνει μόνο 10 βαθμούς.



Εικόνα 50:Γρήγορη και σωστή απάντηση



Εικόνα 51:Λάθος απάντηση

Όταν απαντηθούν και οι 10 ερωτήσεις θα εμφανιστεί αυτόματα η λίστα με τα ονόματα των παικτών και τους πόντους τους. Στην ίδια λίστα μπορείτε να μεταφερθείτε και αν πατήσετε το πλήκτρο «**Κατάταξη**» από το βασικό μενού των επιλογών. Σε αυτή τη λίστα εμφανίζονται οι παίκτες ταξινομημένοι με βάση τους πόντους τους, και παράλληλα εμφανίζονται και πόσες ήταν οι ερωτήσεις που απαντήθηκαν από τον κάθε παίκτη, πόσες ήταν σωστές και σε πόσο χρόνο απαντήθηκαν.



Παίκτης	Πόντοι	Σωστές απαντ.	Συνολ. απαντ.	Μέσος χρόνος απάντησης
ελενα	128	12	21	7.0952
eIena	128	11	16	12.6875
Helen	123	9	10	12.5
	0	0	0	0.0
alex	0	0	1	4.0

Εικόνα 52:Επιλογή Εμφάνισης Λίστας Κατάταξης

Επιπλέον, υπάρχει άλλη μία επιλογή στο κύριο μενού, η «**Σύντομη Περιγραφή**» , όπου εξηγεί με σύντομο και κατανοητό τρόπο πως λειτουργούν οι κανόνες του παιχνιδιού, όπως στην εικόνα που ακολουθεί:

Παιχνίδι ερωτήσεων

Το παιχνίδι λειτουργεί ως εξής:

Με την έναρξη «Νέου Παιχνιδιού» εμφανίζονται τυχαία 10 ερωτήσεις με 4 πιθανές απαντήσεις η κάθε μία. Στην οθόνη εκτός από την ερώτηση φαίνεται ο αριθμός ερώτησης που εμφανίζεται την δεδομένη στιγμή, οι πόντοι που έχει μέχρι τώρα ο κάθε παίκτης και ο μετρητής χρόνου.

Όταν ο παίκτης απαντήσει σωστά σε λιγότερο από 10 δευτέρα τότε κερδίζει 15 πόντους, όταν απαντήσει σωστά μέσα σε 10 έως 20 δευτέρα τότε κερδίζει 13 πόντους και αν απαντήσει σωστά σε χρόνο πάνω από 20 δευτέρα τότε παίρνει 10 βαθμούς στην ερώτηση. Στην περίπτωση που απαντήσει λάθος του αφαιρούνται 5 πόντοι.

Όταν τελειώσουν οι 10 πιθανές ερωτήσεις θα εμφανίζονται τα usernames, οι πόντοι όλων των παικτών, ο αριθμός ερωτήσεων που απάντησε κάθε παίκτης και ο μέσος χρόνος απαντήσής του. Το παιχνίδι αρχίζει!!

Καλή επιτυχία!!

Εικόνα 53:Επιλογή Σύντομης Περιγραφής

[1] Οι συναρτήσεις που είναι γραμμένες σε **MySQL** μας επιτρέπουν να έχουμε πρόσβαση στην βάση δεδομένων.

[2] Ο **Apache** HTTP γνωστός και απλά σαν Apache είναι ένας εξυπηρετητής του παγκόσμιου ιστού (web). Όποτε ένας χρήστης επισκέπτεται ένα ιστότοπο το πρόγραμμα πλοήγησης (browser) επικοινωνεί με έναν διακομιστή (server) μέσω του πρωτοκόλλου HTTP, ο οποίος παράγει τις ιστοσελίδες και τις αποστέλλει στο πρόγραμμα πλοήγησης. Ο Apache είναι ένας από τους δημοφιλέστερους γιατί λειτουργεί σε διάφορες πλατφόρμες όπως τα Windows, το Linux, το Unix και το Mac OS X. Συντηρείται τώρα από μια κοινότητα ανοικτού κώδικα κάτω από την επιτήρηση του ιδρύματος Λογισμικού Apache (Apache Software Foundation). Ο Apache χρησιμοποιείται και σε τοπικά δίκτυα σαν διακομιστής συνεργαζόμενος με συστήματα διαχείρισης Βάσης Δεδομένων π.χ. Oracle, MySQL.

[3] **Hypertext Preprocessor**- Είναι μια γλώσσα που την χρησιμοποιούμε για εφαρμογές που εκτελούνται στον διακομιστή (όπως και η ASP).

[4] **XML**- Extensible Markup Language: Η γλώσσα xml εκτός από την χρήση της στον κατάλογο layout του έργου, χρησιμοποιείται σαν ιεραρχικό μοντέλο για αποθήκευση και ανάκτηση δεδομένων. Το xml πρότυπο έχει δημιουργηθεί από το W3C (World Wide Web Consortium). Πρόκειται για αρχεία που περιέχουν δεδομένα και μεταδεδομένα μαζί, καθώς επίσης αναπαριστώνται και οι σχέσεις μεταξύ των δεδομένων αυτών. Ονομάζονται extensible γιατί παρέχουν την δυνατότητα στους χρήστες να ορίσουν τα δικά τους στοιχεία. Όλα τα έγγραφα xml είναι γραμμένα με χαρακτήρες Unicode. Τα αρχεία xml προκύπτουν από την βάση δεδομένων στο διακομιστή και γίνονται κατανοητά από την εφαρμογή Android. Για να πραγματοποιηθεί αυτή η επικοινωνία σημαντικό ρόλο έχουν οι υπηρεσίες διαδικτύου (Web Services).

[5] Ο **πυρήνας Linux** είναι ένας πυρήνας του λειτουργικού συστήματος που χρησιμοποιείται από την ευρύτερη οικογένεια των Unix λειτουργικών συστημάτων. Είναι το πιο χαρακτηριστικό παράδειγμα για κώδικα ανοιχτού λογισμικού. Ο πυρήνας Linux δημιουργήθηκε αρχικά από τον Φιλανδό φοιτητή πληροφορικής Λίνους Τόρβαλντς το 1991. Τα πνευματικά δικαιώματα παρέχονται από τη «GNU Free Documentation License». Γενικότερα το λειτουργικό σύστημα που βασίζεται σε πυρήνα Linux είναι το λειτουργικό σύστημα Linux, είναι γραμμένο σε γλώσσα C, και είναι μεταφέρσιμο σε κάθε πλατφόρμα υλικού. Ξεχωρίζει για την σταθερότητά του ως λειτουργικό και για το ισχυρό τείχος προστασίας που παρέχει.

Βιβλιογραφία

<http://developer.android.com/index.html>

<http://www.w3schools.com>

Σημειώσεις Βάσεων Δεδομένων-Κεραμόπουλος Ευκλείδης

<http://techblog.gr/tablet/android-honeycomb-google-statistics-3786/>

<http://www.millennialmedia.com/mobile-intelligence/mobile-mix/>

<http://www.freewebmasterhelp.com/tutorials/phpmysql>

<http://dev.mysql.com/doc/>

http://www.siteground.com/tutorials/phpmyadmin/phpmyadmin_create_database.htm

<http://www.easyphp.org/>

<http://www.w3.org/standards/xml/>

http://cgi.di.uoa.gr/~ys01/using_php.pdf

Beginning Android, Mark L. Murphy 2009

Android Wireless Application Development, Second Edition 2011

<http://www.anddev.org/>

Beginning Android 2, Mark L. Murphy