



**ΤΕΙ ΚΡΗΤΗΣ**

**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**The library quiz**

**ΦΟΙΤΗΤΡΙΑ:** Κωνσταντούλα Μπακέλα (2308)

**ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:** Παπαδουράκης Γεώργιος.

**Ηράκλειο 2016**

.

<<Δηλώνουμε υπεύθυνα ότι το παρόν κείμενο αποτελεί προϊόν προσωπικής μελέτης και εργασίας. Οι πηγές που χρησιμοποιήθηκαν αναφέρονται στην βιβλιογραφία>>

### ***Ευχαριστίες***

Ευχαριστώ ιδιαίτερος το συνάδελφο κο. Κωνσταντίνο Δρακωνάκη για τη σημαντική βοήθεια στην κατασκευή της android εφαρμογής. Τον κο. Μιχαήλ Νικητάκη για το θέμα της εργασίας και τον επιβλέπων καθηγητή μου κο. Γεώργιο Παπαδουράκη για την καθοδήγηση της πορείας της πτυχιακής εργασίας. Τέλος, ευχαριστώ τους δικούς μου ανθρώπους για την στήριξη τους κατά τη διάρκεια της φοίτησής μου.

**Abstract**

Information literacy (IL) is world widely acknowledged these days as a set of basic skills every citizen of a modern society should have in order to participate in it and solve problems on personal or professional basis.

Academic librarians, as facilitators of the educational procedure, design and deliver IL programs. In most cases they even take up a new role as instructors in these programs, bringing the academic library in center of the institution's educational procedure. This necessitates the acquisition of new skills, besides the ones gained by their typical library training, such as instruction skills, pedagogical skills, educational design skills in order to become effective instructors

The present paper aims to study, develop educate and learn how to create application for Android based smartphones.

For the purposes of the paper an android application has developed using the platform Android Studio 4.0. In the beginning of the thesis we will see the Android history, how it started how it has been developed, and further characteristics of this technology. Later on, we will move to the major parts and we will use JAVA programming language which is Object Oriented.XML, PHP data objects, MYSQL Database. Furthermore, after all the studying and experiences we had during studying the characteristics we will create our application.

A game designed and programmed to provide the knowledge and the education to all library users. An interesting game with triggering questions and answers providing basic knowledge of how we should behave in a respectable environment a library even if is part of a foundation or a public library. Useful information about the book. How to categorize the book according to the title, the indicated code, and many other tips, that serve with proper education of each guest reader of a library.

## Περίληψη

Πτυχιακή εργασία που έχει υλοποιηθεί για της ανάγκες της βιβλιοθήκης του ΤΕΙ Ηρακλείου. Υλοποίηση μιας ιδέας του υπεύθυνου της, Μιχάλη Νικητάκη. Αρχικός συντάκτης των ερωτήσεων, απαντήσεων και κατηγοριών του περιεχομένου της εφαρμογής.

Στην παρούσα πτυχιακή εργασία γίνεται αναφορά της πληροφοριακής παιδείας στην εκπαίδευση περιγράφοντας κάποιες από της βασικές έννοιες με στόχο τον προσδιορισμό μιας σαφής κατεύθυνσης για ένα πρόγραμμα πληροφοριακής παιδείας. Τι είναι η πληροφορία ως στοιχείο, ποια είναι η ανάγκη για την αποτελεσματική χρήση της πληροφορίας, και ποια είναι η έννοια της πληροφοριακής παιδείας.

Η εκπαιδευτική μέθοδος που έχουμε επιλέξει ώστε να επηρεάσουμε τη διδασκαλία της πληροφοριακής παιδείας, είναι η δημιουργία μιας Android εφαρμογής. Ένα παιχνίδι ερωτήσεων – απαντήσεων που αφορούν κατά κύριο λόγο τους χρήστες των βιβλιοθηκών ανά τον κόσμο με κύριο γνώμονα τη γνώση και την εκπαίδευση.

Κατασκευάστηκε με ερωτήσεις και απαντήσεις που προσφέρουν βασικές γνώσεις για το πώς θα πρέπει να συμπεριφέρονται οι χρήστες σε ένα περιβάλλον βιβλιοθήκης, είτε αποτελεί μέρος ενός ιδρύματος ή μιας δημόσιας βιβλιοθήκης. Χρήσιμες πληροφορίες για το βιβλίο. Πώς να κατηγοριοποιήσουμε το βιβλίο σύμφωνα με τον τίτλο, τον υποδεικνύοντα κώδικα, και πολλές άλλες συμβουλές, που εξυπηρετούν με την κατάλληλη εκπαίδευση του κάθε αναγνώστη επισκεπτών μιας βιβλιοθήκης.

Για να αναφερθούμε στα στάδια της τεχνολογίας που χρησιμοποιήσαμε, έχει γίνει μια σύντομη ιστορική αναδρομή στο λειτουργικό σύστημα Android, από την πρώτη της εμφάνιση μέχρι και την πιο πρόσφατη. Αναφερόμαστε λοιπόν, στην αρχιτεκτονική της δομής επεξηγώντας με απλό τρόπο τη λειτουργία της. Η ανάπτυξη του παιχνιδιού έγινε με χρήση της εφαρμογής Android Studio και χρησιμοποιώντας τα Android API's (Application Programming Interface) και SDK(System Development Kit). Αρχικά θα δούμε γιατί επιλέξαμε το συγκεκριμένο λειτουργικό σύστημα και αργότερα θα δούμε τα ενδότερα στοιχεία του.

Τέλος, βλέπουμε το τελικό αποτέλεσμα της εφαρμογής έτσι όπως φαίνεται στο χρήστη και τα συμπεράσματα που βγάζουμε μέσα από την πορεία της πτυχιακής αυτής εργασίας..

## Περιεχόμενα

<b>1</b>	<b>Εισαγωγή .....</b>	<b>7</b>
1.1	Γενικά.....	7
1.2	Σκοπός.....	8
1.3	Περιγραφή .....	8
1.4	Δομή της εργασίας .....	9
<b>2</b>	<b>Πληροφορική παιδεία εκπαίδευσης χρηστών βιβλιοθήκης.....</b>	<b>10</b>
2.1	Η έννοια της πληροφοριακής παιδείας στην εκπαίδευση. ....	10
2.2	Τι είναι πληροφορία.....	11
2.3	Πρότυπα σχεδίασης εκπαιδευτικών στόχων για τα προγράμματα πληροφοριακής παιδείας.....	11
2.3.1	Πρόσβαση .....	12
2.3.2	Αξιολόγηση.....	12
2.3.3	Χρήση της πληροφορίας.....	13
2.4	Εκπαίδευση χρηστών.....	13
2.4.1	Στόχοι και σκοποί .....	14
2.4.2	Αξιολόγηση.....	14
2.5	Ψηφιακά παιχνίδια σε ακαδημαϊκές βιβλιοθήκες.....	14
2.5.1	Gamification (Παιχνιδοποίηση) στις βιβλιοθήκες .....	14
2.5.2	Εφαρμογές της παιχνιδοποίησης στις βιβλιοθήκες. ....	16
	<b>18</b>	
2.6	Αναφορές.....	19
<b>3.</b>	<b>Android development - History of infrastructure .....</b>	<b>22</b>
3.1	Τι είναι το Android .....	22
3.2	Χαρακτηριστικά του Android .....	23
3.3	Γιατί επιλέγω Android .....	23
3.3.1	Εφαρμογές Android .....	25
3.4	Ιστορική αναδρομή, πορεία εξέλιξης του Android. ....	25
3.4.1	Android 1.5 Cupcake .....	25
3.4.2	Android 1.6 Donut .....	26
3.4.3	Android 2.0 Eclair.....	27
3.4.4	Android 2.2 Froyo.....	27
3.4.5	Android 3.0 Honeycomb.....	29
3.4.6	Android 4.0 Ice Cream Sandwich .....	29
3.4.7	Android 4.1 Jelly Bean.....	30
3.4.8	Android 4.4 KitKat .....	32
3.4.9	Android 5.0 Lollipop.....	32
3.4.10	Android 6.0 Marshmallow .....	34
3.5	Η αρχιτεκτονική του Android .....	35
3.5.1	Σχήμα αρχιτεκτονικής τεχνολογίας Android .....	35
3.5.2	Application Framework .....	35
3.5.2.6	Βιβλιοθήκες (Libraries) .....	38
3.5.3	Linux Kernel .....	40
3.5.4	ART και Dalvik.....	40
3.6	Βασική Δομή του Android project .....	41
3.7	Κάθε project Android περιέχει:.....	41
3.7.1	src:.....	41
3.7.2	Res: .....	41

<b>3.8</b>	<b>Σημαντικοί κατάλογοι στο Android project:</b>	<b>43</b>
3.8.1	AndroidManifest.xml	43
3.8.2	MainLayout.xml:	45
3.8.3	Gradle Scripts	47
<b>3.9</b>	<b>Διαδικασίες ανάπτυξης Android.</b>	<b>49</b>
3.9.1	Δημιουργία Project στο Android Studio	49
<b>3.10</b>	<b>Official development tools in android studio</b>	<b>50</b>
3.10.1	Android SDK (System Development Kit)	50
3.10.2	Διαδικασία Αποσφαλμάτωσης (Debugging)	52
3.10.3	Android Debug Bridge	53
3.10.4	Δημοσίευση Εφαρμογής στο κοινό (Publishing)	53
3.10.5	Android Virtual Device Manager(AVD)	54
<b>3.11</b>	<b>Αναφορές</b>	<b>86</b>
<b>4</b>	<b>Δημιουργίας της εφαρμογής – τελικό αποτέλεσμα</b>	<b>56</b>
<b>4.1</b>	<b>Ανάλυση της Δημιουργίας της εφαρμογής</b>	<b>56</b>
4.1.1	Η βάση δεδομένων My SQL	56
4.1.2	Εφαρμογή και ανάπτυξη της MySQL	57
4.1.3	PHP Script for exporting Data in JSON Array	58
4.1.4	Ανάλυση της PHP	59
4.1.5	JSON	63
4.1.6	Εξάγοντας τα δεδομένα σε JSON Array στον browser	64
<b>4.2</b>	<b>Δημιουργία Project στο Android Studio</b>	<b>65</b>
4.2.1	Δήλωση αδειών χρήστη στο AndroidManifest	66
4.2.2	Έλεγχος της συσκευής για πρόσβαση στο Ίντερνετ	67
<b>4.3</b>	<b>To Layout του Library quiz</b>	<b>70</b>
4.3.1	Welcome screen.	70
4.3.2	Δημιουργία εμφανισιακού περιεχομένου της πρώτης οθόνης	71
4.3.3	Κύριο μενού επιλογών- Κύρια οθόνη.	71
4.3.4	Αρχική οθόνη (Start menu).	73
4.3.5	Οθόνη κατηγοριών - λίστα κατηγοριών	76
4.3.6	Επιλογές –οθόνη ρυθμίσεων	77
<b>5</b>	<b>Συμπεράσματα</b>	<b>85</b>

# 1 Εισαγωγή

## 1.1 Γενικά

Η παρούσα πτυχιακή εργασία αναπτύχθηκε για λογαριασμό της βιβλιοθήκης του ΤΕΙ Ηρακλείου και αφορά την εισαγωγή δημιουργικών και νέων εφαρμογών στις ελληνικές βιβλιοθήκες. Αντικείμενο της εργασίας αυτής αποτελεί η πληροφοριακή παιδεία στην εκπαίδευσή και τον τρόπο που επιλέξαμε να την εφαρμόσουμε.

Αφορμή στάθηκε η ιδέα υπεύθυνου της βιβλιοθήκης ΤΕΙ Ηρακλείου Κρήτη, Μιχάλη Νικιτάκη να κατασκευαστεί μια ηλεκτρονική εφαρμογή – οδηγός, στην εκπαίδευσης νέων χρηστών βιβλιοθήκης.

Η εφαρμογή που επιλέξαμε αφορά ένα παιχνίδι ερωτήσεων-απαντήσεων δημιουργώντας ένα βασικό οδηγό βιβλιοθήκης για φοιτητές, βασισμένο στον τρόπο εύρεσης βιβλίων, άρθρα περιοδικών, και άλλων πόρων σε ένα περιβάλλον βιβλιοθήκης για τις αναθέσεις και ερευνητικές εργασίες.

## 1.2 Σκοπός

Με το πέρασμα των τελευταίων χρόνων, στην ελληνική πραγματικότητα, οι ακαδημαϊκές βιβλιοθήκες υπέστησαν αρκετές διαφοροποιήσεις με κύριο σκοπό την υποστήριξη των αναγκών των χρηστών τους για πληροφόρηση, επιμόρφωση, εκπαίδευση, ψυχαγωγία κ.ά. Οι διαφοροποιήσεις αυτές οφείλονται στην υιοθέτηση των νέων τεχνολογιών, κυρίως με τη χρήση των ηλεκτρονικών υπολογιστών, του διαδικτύου αλλά και με την ανάπτυξη σχέσεων. Οι διαφοροποιήσεις αυτές έθεσαν και θέτουν μια νέα σημαντική παράμετρο, τη διαμόρφωση του **ρόλου των σύγχρονων βιβλιοθηκών** ως ουσιαστικών εταίρων στις διαδικασίες πληροφοριακής παιδείας / διαβίου μάθησης. Ως εκ τούτου λοιπόν, ο κύριος σκοπός της εργασίας αυτής η να εισάγουμε ένα παιχνίδι επιδεξιότητας ενημέρωσης και διασκέδασης, στην καθημερινότητα του χρήστη, αναπτύσσοντας έτσι την πληροφοριακή παιδεία σε ένα περιβάλλον βιβλιοθήκης.

## 1.3 Περιγραφή

Το παιχνίδι που κατασκευάσαμε χαρακτηρίζεται ως ένα παιχνίδι γνώσεων. Περιγράφεται ως ένα android developed παιχνίδι και έχει χαρακτήρα ερωτήσεων – απαντήσεων και επιλογής θεμάτων των κατηγοριών. Ο χρήστης μπορεί να επιλέξει ένα γρήγορο παιχνίδι έχοντας τη δυνατότητα να ερωτηθεί σε θέματα διαφορετικών κατηγοριών ερωτήσεων, η να επιλέξει συγκεκριμένες κατηγορίες ερωτήσεων – απαντήσεων, παρέχοντας συγκεκριμένη γκάμα γνώσεων. Η εφαρμογή έχει κατασκευαστεί και στην αγγλική γλώσσα. Μέσω το μενού των ρυθμίσεων ο χρήστης μπορεί να επιλέξει τη γλώσσα στην οποία θέλει να παίξει, καθώς έχεις κατασκευαστεί και στην Αγγλική γλώσσα, και διάφορες άλλες παραμέτρους που



επηρεάζουν στο παίξιμο της εφαρμογής. Τέλος, μέσω την εφαρμογής αυτής, ο χρήστης μπορεί ανά πάσα στιγμή να επικοινωνήσει με την βιβλιοθήκης του ΤΕΙ μας, είτε καλώντας το προσωπικό της βιβλιοθήκης, είτε αποστέλλοντας ένα ηλεκτρονικό μήνυμα ταχυδρομείου, μέσω της ειδικής φόρμας που έχεις σχεδιαστεί γι' αυτό το λόγο.

## 1.4 Δομή της εργασίας

Με κύριο μέλημα τη διευκόλυνση της πληροφοριακής παιδείας, την προσέλκυση των χρηστών και τη μετατροπή της σε σημείο αναφοράς των νέων και όχι μόνο, χωρίσαμε την παρούσα εργασία σε δύο μέρη. Στο πρώτο μέρος θα αναφερθούμε, στην Πληροφοριακή παιδεία ως ένα θεμελιώδες συστατικό της εκπαιδευτικής διαδικασίας, Στο δεύτερο μέρος θα επεκταθούμε στην τεχνολογία που χρησιμοποιήθηκε για την κατασκευή του παιχνιδιού.

**Η πτυχιακή εργασία είναι δομημένη σε 5 βασικά κεφάλαια.**

- ✓ **Κεφάλαιο 1:** Εισαγωγή
- ✓ **Κεφάλαιο 2:** Πληροφορική παιδεία εκπαίδευσης χρηστών βιβλιοθήκης.
- ✓ **Κεφάλαιο 3:** Adnroid development - History of infrustructure  
Στο κεφάλαιο αυτό γίνεται μια ιστορική αναφορά στην τεχνολογία του Android από τη γέννησή της μέχρι και σήμερα. Γίνεται αναφορά στην επεξήγηση της τεχνολογίας, γιατί επιλέγω αυτή τη την τεχνολογία, και που εφαρμόζεται.
- ✓ **Κεφάλαιο 4:** Διαδικασίες ανάπτυξης της εφαρμογής. – τελικό αποτέλεσμα Επεξήγηση και ανάλυση της εφαρμογής
- ✓ **Κεφάλαιο 5:** Συμπεράσματα

## 2 Πληροφορική παιδεία εκπαίδευσης χρηστών βιβλιοθήκης.

### 2.1 Η έννοια της πληροφοριακής παιδείας στην εκπαίδευση.

Πληροφορική παιδεία θεωρείται η γνώση και οι δεξιότητες που είναι απαραίτητες για να εντοπίζει κανείς τις πληροφορίες που χρειάζονται για να εκτελέσει ένα συγκεκριμένο έργο ή για να λύσει κάποιο πρόβλημα, να αναζητά πληροφορίες οικονομικά ανταποδοτικές, να τις οργανώνει και να τις αναδιοργανώνει, να τις ερμηνεύει και να τις αναλύει, εφόσον έχουν βρεθεί και ανακτηθεί (π.χ. από το διαδίκτυο), να αξιολογεί την ακρίβεια και την αξιοπιστία των πληροφοριών, λαμβάνοντας υπόψη την ηθική αναγνώριση των πηγών από τις οποίες αποκτήθηκαν οι πληροφορίες, να διακινεί και να παρουσιάζει τα αποτελέσματα της ανάλυσης και της ερμηνείας αυτών σε άλλους, εάν είναι απαραίτητο, και μετά να τις χρησιμοποιεί για την επίτευξη ενεργειών και αποτελεσμάτων.[1]

Αναγνωρίζεται πλέον σε παγκόσμιο επίπεδο ως ένα σύνολο βασικών δεξιοτήτων για την αποτελεσματική συμμετοχή των πολιτών στις σύγχρονες κοινωνίες και την επίλυση προβλημάτων τόσο στην προσωπική όσο και στην επαγγελματική ζωή τους. Η αλλαγή στην εκπαιδευτική σκέψη, η εκρηκτική αύξηση της ηλεκτρονικής πληροφόρησης και οι αλματώδεις εξελίξεις στην τεχνολογία, αποτελούν πλέον τη μεγάλη πρόκληση για τις ακαδημαϊκές βιβλιοθήκες όσον αφορά στον εκπαιδευτικό τους ρόλο.

Για να προσδιοριστεί μία σαφής κατεύθυνση για ένα πρόγραμμα πληροφοριακής παιδείας, είναι σημαντικό να γνωρίζει κανείς τις διαφορετικές έννοιες πληροφοριακής παιδείας που σχετίζονται με αυτή.

Η πληροφορική παιδεία στηρίζει και επεκτείνει τη δια βίου μάθηση, μέσω ικανοτήτων που μπορούν να χρησιμοποιήσουν τεχνολογίες, αλλά βασικά ανεξάρτητες από αυτές.

Ένα άτομο με πληροφορική παιδεία είναι ικανό να :

- Προσδιορίζει την έκταση της απαιτούμενης πληροφόρησης.
- Έχει πρόσβαση στην απαιτούμενη πληροφόρηση αποτελεσματικά και ικανοποιητικά.

- Αποτιμά την πληροφόρηση και τις πηγές της με κριτικό πνεύμα.
- Ενσωματώνει επιλεγμένη πληροφόρηση στη γνωστική του βάση.
- Χρησιμοποιεί την πληροφόρηση αποτελεσματικά για να επιτελέσει ένα συγκεκριμένο σκοπό

## 2.2 Τι είναι πληροφορία

Πληροφορία είναι ένα στοιχείο, που ορίζεται με ποικίλους τρόπους ανάλογα με το μορφότυπο και το μέσο που χρησιμοποιείται για την αποθήκευση ή μεταφορά του, καθώς και το επιστημονικό πεδίο που το ορίζει. Ο Case δίνει έναν ευρύτερο ορισμό.

Εδώ ο όρος είναι συνώνυμος με:

- Συμπυκνωμένη γνώση
- Οργανωμένη ανθρώπινη εμπειρία
- Ένα στοιχείο που μπορεί να προσφέρει αναρίθμητα δεδομένα
- Ένα στοιχείο που μπορεί να εμφανιστεί σε διαφορετικά μορφότυπα, με διαφορετικές μορφές οργάνωσης, με διαφορετικούς τρόπους μεταφοράς και με ποικίλες μεθόδους διανομής
  - Ανθρώπους: οικογένεια, φίλοι, εκπαιδευτές, συμφοιτητές
  - Ιδρύματα, π.χ. επαγγελματίες του εθνικού συστήματος υγείας ή υπηρεσιών βοήθειας.

## 2.3 Πρότυπα σχεδίασης εκπαιδευτικών στόχων για τα προγράμματα πληροφοριακής παιδείας

Ο βασικός εκπαιδευτικός στόχος των προγραμμάτων πληροφοριακής παιδείας, ειδικά στην τριτοβάθμια εκπαίδευση, είναι να βοηθήσουν τους σπουδαστές να αναπτύξουν τις γνώσεις τους και τις δεξιότητες τους σε όλα τα ζητήματα που αφορούν την πληροφορία. Η σύνδεση των εκπαιδευτικών στόχων των προγραμμάτων με το τελικό αποτέλεσμα, την απόκτηση δηλ. των επιθυμητών γνώσεων και δεξιοτήτων, εναπόκειται εξολοκλήρου στο σωστό προσδιορισμό και σχεδιασμό της πορείας που θα ακολουθηθεί προς την επίτευξη των στόχων που θα τεθούν.

Τα πρότυπα αυτά είναι δημιουργημένα από οργανισμούς συνήθως εθνικούς για να καλύψουν το κενό που υπήρχε στην ύπαρξη ενός κοινού μοντέλου εκπαίδευσης. Κάποια από αυτά δε χρησιμοποιούνται μόνο σε εθνικό επίπεδο πλέον αλλά και παγκοσμίως. Η αποδεδειγμένη χρησιμότητά τους τα καθιστά έγκυρα εργαλεία για κάθε βιβλιοθήκη που επιθυμεί και επιδιώκει να προσφέρει μια καλού επιπέδου εκπαίδευση στους χρήστες της.

Τα πρότυπα της πληροφοριακής παιδείας, για να είναι αποτελεσματικά, περιέχουν τρία βασικά συστατικά : πρόσβαση, αξιολόγηση και χρήση της πληροφορίας. Αυτά τα τρία μαζί θεωρούνται ο «πυρήνας» των περισσότερων προτύπων που δημιουργήθηκαν από οργανώσεις/ ενώσεις βιβλιοθηκών, όπως είναι σχετικές συνεργασίες του AASL, ACRL, SCOUNL ANZIIL\*.

Τα πρότυπα πληροφοριακής παιδείας βασίζονται σ' αυτές τις διεθνείς εμπειρίες και συνεργασίες. Αυτά τα πρότυπα είναι κατηγοριοποιημένα σε τρία βασικά συστατικά της πληροφοριακής παιδείας.

### 2.3.1 Πρόσβαση

Ο χρήστης έχει πρόσβαση στην πληροφορία αποτελεσματικά και ικανοποιητικά

#### 1. Ορισμός και άρθρωση της πληροφοριακής ανάγκης

- Ορίζει ή αναγνωρίζει την ανάγκη του για πληροφόρηση
- Αποφασίζει να κάνει κάτι ώστε να βρει σχετικές πληροφορίες
- Εκφράζει και ορίζει την πληροφοριακή ανάγκη
- Αρχίζει τη διαδικασία αναζήτησης/ έρευνας

#### 2. Εντοπισμός της πληροφορίας

- Αναγνωρίζει και αξιολογεί πηγές πληροφόρησης
- Αναπτύσσει στρατηγικές αναζήτησης
- Έχει πρόσβαση στις επιλεγμένες πηγές
- Επιλέγει και ανακτά την πληροφορία που εντόπισε

### 2.3.2 Αξιολόγηση

Ο χρήστης αξιολογεί κριτικά και επιδέξια την πληροφορία

#### 1. Εκτίμηση της πληροφορίας

- Αναλύει, εξετάζει και διαλέγει πληροφορίες
- Γενικεύει και ερμηνεύει την πληροφορία
- Επιλέγει και συνθέτει την πληροφορία
- Αξιολογεί την ακρίβεια και τη σχετικότητα των ανακτηθέντων πληροφοριών

## 2. Οργάνωση της πληροφορίας

- Τακτοποιεί και κατηγοριοποιεί τις πληροφορίες
- Ομαδοποιεί και οργανώνει τις πληροφορίες που ανέκτησε
- Αποφασίζει πια είναι η καλύτερη και πιο χρήσιμη πληροφορία

### 2.3.3 Χρήση της πληροφορίας

Ο χρήστης χρησιμοποιεί την πληροφορία δημιουργικά και με ακρίβεια.

#### 1. Χρήση της πληροφορίας

- Βρίσκει νέους τρόπους να μεταδίδει, να παρουσιάζει και να χρησιμοποιεί την πληροφορία
  - Ανακτά την ανακτώμενη πληροφορία
  - Μαθαίνει ή εμπεδώνει την πληροφορία ως προσωπική γνώση
  - Παρουσιάζει το προϊόν πληροφόρησης

#### 2. Μετάδοση και ηθική χρήση της πληροφορίας

- Καταλαβαίνει την ηθική χρήση της πληροφορίας
- Σέβεται τη νομική/ νόμιμη χρήση της πληροφορίας
- Μεταδίδει το προϊόν μάθησης με αναγνώριση/ ομολογία της διανοητικής ιδιότητας.
  - Χρησιμοποιεί τις σχετικές γνώσεις
  - Χρησιμοποιεί τα σχετικά πρότυπα

## 2.4 Εκπαίδευση χρηστών.

Η εκπαίδευση χρηστών είναι αποτελεσματική μόνο όταν χρειάζεται (at the moment of need) και είναι ή πρέπει να θεωρείται ως μια διαρκής διαδικασία. Η διαδικασία αυτή συνήθως πραγματοποιείται σε δύο επίπεδα. Το πρώτο, ο προσανατολισμός, είναι ουσιαστικά μια εισαγωγή στις χρήσεις των προσφερόμενων υπηρεσιών. Το δεύτερο, η εκπαίδευση, αναφέρεται στη γνώση ή εκμάθηση των πληροφοριακών πηγών που διατίθενται σε συγκεκριμένες θεματικές ενότητες-κατηγορίες.

### 2.4.1 Στόχοι και σκοποί

Για να αποδώσει ένα πρόγραμμα εκπαίδευσης χρηστών οι στόχοι και οι αντικειμενικοί σκοποί του πρέπει να έχουν καθοριστεί με βάση τις ανάγκες των χρηστών.

Έτσι οι χρήστες γίνονται ικανοί, πρώτα απ' όλα, να βρουν τις πληροφορίες που αναζητούν πάνω σε διάφορα θέματα σε σχέση με τις σπουδές τους. Σ' αυτό το σημείο βρίσκεται και το ενδιαφέρον των ακαδημαϊκών που προσπαθούν να υποκινήσουν τους φοιτητές να ανατρέξουν στη βιβλιοθήκη για να βρουν τις πληροφορίες που χρειάζονται.

Επιθυμούν οι σπουδαστές να συγκεντρώσουν υλικό που θα τους βοηθήσει να ανταπεξέλθουν στα γνωστικά τους καθήκοντα. Αυτή είναι μια πιο παιδαγωγική προσέγγιση του θέματος της ανάκτησης, σε αντίθεση με τους βιβλιοθηκονόμους που ενδιαφέρονται περισσότερο για τη χρήση των σωστών πηγών, παρά για τη μέθοδο ανάκτησης που χρησιμοποιείται απ' τους σπουδαστές

### 2.4.2 Αξιολόγηση

Αξιολόγηση είναι η διαδικασία που πραγματοποιείται μετά το πέρας του προγράμματος εκπαίδευσης χρηστών ή και κατά τη διάρκειά του, κατά την οποία γίνεται υγκέντρωση πληροφοριών για τα αποτελέσματα που προέκυψαν. Η ανάλυση των πληροφοριών αυτών οδηγεί σε λογικά και χρήσιμα συμπεράσματα που βοηθούν στη λήψη αποφάσεων.

## 2.5 Ψηφιακά παιχνίδια σε ακαδημαϊκές βιβλιοθήκες.

### 2.5.1 Gamification (Παιχνιδοποίηση) στις βιβλιοθήκες

Ο όρος gamification εμφανίστηκε για πρώτη φορά στη βιβλιογραφία των ψηφιακών μέσων ενημέρωσης το 2008 και διαδόθηκε ευρύτερα το δεύτερο εξάμηνο του 2010 όταν και υιοθετήθηκε από ηλεκτρονικές επιχειρήσεις ως το νέο πολύτιμο εργαλείο και πραγματοποιήθηκαν σχετικά συνέδρια και έρευνες.

Κύριος στόχος του είναι να προσελκύσει νέους χρήστες, να αυξήσει την αλληλεπίδραση και τη δέσμευση μαζί τους, δημιουργώντας αποσιωμένους χρήστες.

Με το gamification οι βιβλιοθήκες και οι βιβλιοθηκονόμοι προβάλλουν και προωθούν τις υπηρεσίες τους, τις συλλογές τους και τα τεκμήρια τους. Με αποτέλεσμα να διατηρούν, να προσελκύουν μακροπρόθεσμα τους χρήστες και τέλος να τους παρακινούν ώστε να γνωρίσουν καλύτερα τη βιβλιοθήκη της γειτονιάς τους.

Το να σχεδιάσει κάποιος,, ένα ψυχαγωγικό παιχνίδι το οποίο στηρίζεται στην εμπειρία και με αντικείμενο την ανάδειξη της πληροφοριακής παιδείας, είναι ένα αρκετά δύσκολο εγχείρημα. Για αυτό το λόγο οι βιβλιοθήκες, πριν ξεκινήσουν μία εφαρμογή, θα πρέπει να εξετάσουν προσεχτικά το περιεχόμενό και το σχεδιασμό της.

Σύμφωνα με τη βιβλιογραφία οι βέλτιστες πρακτικές που θα πρέπει να λάβουν υπόψη τους οι βιβλιοθήκες στη δημιουργία μιας εφαρμογής.

1) Το παιχνίδι θα πρέπει να είναι απλό. Τα πιο επιτυχημένα online παιχνίδια είναι αυτά που είναι απλά.

2) Ο σχεδιασμός θα πρέπει να ακολουθεί πάντα μία συγκεκριμένη στρατηγική και την ανάλογη προώθηση. Αυτή η στρατηγική θα πρέπει να συνδυάζει το διασκεδαστικό παιχνίδι με το εκπαιδευτικό, έτσι ώστε να δίνει στους χρήστες της βιβλιοθήκης ένα ισχυρό κίνητρο για να τη χρησιμοποιούν.

3) Αν στα πλαίσια του σεναρίου του παιχνιδιού κριθεί αναγκαία η αφομοίωση μιας δραστηριότητας το παιχνίδι θα πρέπει να σχεδιασθεί έτσι ώστε οι χρήστες να μην μπορούν να προχωρήσουν σε επόμενο στάδιο χωρίς την επίτευξη του πρώτου.

4) Δεν αρκεί να σχεδιαστεί ένα διαδραστικό τεστ εκμάθησης και να ονομαστεί παιχνίδι. Οι χρήστες θα πρέπει να διασκεδάζουν Αυτό μπορεί να πραγματοποιηθεί με την προσθήκη ανταγωνισμού.

5) Η παροχή μεγάλου όγκου πληροφοριών είναι ένα από τα μεγαλύτερα πλεονεκτήματα του gamification. Διότι, τα παιχνίδια μεταφέρουν πληροφορίες άμεσα. Ο παίχτης δοκιμάζει κάτι, πληροφορείται για το πόσο καλά απέδωσε σε αυτό, και είτε χρησιμοποιεί τις πληροφορίες που έλαβε για να κάνει μια πιο "έμπειρη" δεύτερη προσπάθεια, είτε έχει τη δυνατότητα να συνεχίσει στην επόμενη πρόκληση. Η

ανατροφοδότηση πληροφοριών στα παιχνίδια των βιβλιοθηκών μπορεί να εμπεριείχε μικρά τεστ εκμάθησης.

6) Δοκιμές και αξιολόγηση παιχνιδιού. Αυτό είναι ένα πολύ κρίσιμο στάδιο στη δημιουργία των παιχνιδιών. Για να επιτύχει το καλύτερο δυνατό αποτέλεσμα το παιχνίδι θα πρέπει δοκιμαστεί και να αξιολογηθεί από τους χρήστες της βιβλιοθήκης.

## 2.5.2 Εφαρμογές της παιχνιδοποίησης στις βιβλιοθήκες.

Οι αρχές του gamification έχουν ήδη εφαρμοσθεί με ιδιαίτερα αποτελεσματικό τρόπο σε διάφορες βιβλιοθήκες. Ενδεικτικά θα αναφερθούμε σε κάποιες από αυτές.

### *New York Public Library: Find the Future*

Πρόκειται για πρωτοποριακή διαδραστική εμπειρία που δημιουργήθηκε ειδικά για τη συγκεκριμένη βιβλιοθήκη από διάσημους σχεδιαστές παιχνιδιών. Απευθύνεται σε νέους άνω των 18 ετών και πρόκειται για ένα παιχνίδι θησαυρού που περιλαμβάνει αποστολές στον πραγματικό κόσμο σε συνδυασμό με εικονικά στοιχεία και online συνεργασίες, όλα εμπνευσμένα από 100 διαφορετικά έργα των συλλογών της New York Public Library. Το παιχνίδι φέρνει σε επαφή τους χρήστες με το έργο και τη ζωή σημαντικών ιστορικών προσωπικοτήτων, σε μια προσπάθεια αναζήτησης του μέλλοντος.

### *Carnegie Mellon University: Library Arcade*

Το Library Arcade περιέχει mini-games σχεδιασμένα έτσι ώστε να βοηθηθούν οι χρήστες να αναπτύξουν τις ερευνητικές τους δεξιότητες, μέσω επαναλαμβανόμενων και ψυχαγωγικών δραστηριοτήτων. Μέσα από τα συγκεκριμένα παιχνίδια οι χρήστες μαθαίνουν πως είναι οργανωμένες οι πληροφορίες σε μια ακαδημαϊκή βιβλιοθήκη π.χ. ταξινόμηση, θεματική ευρετηρίαση, καθώς και πώς να εντοπίσουν πληροφορίες για το θέμα της εκάστοτε έρευνάς τους.

### *The University of North Carolina – Greensboro. The Information Literacy Game.*

Πρόκειται για ένα online παιχνίδι γνώσεων. Σχεδιάστηκε από τους ίδιους τους βιβλιοθηκονόμους της βιβλιοθήκης, με στόχο τη «γνωριμία» της βιβλιοθήκης με τους πρωτοετείς φοιτητές. Μπορούν να συμμετάσχουν έως και τέσσερις παίχτες, οι οποίοι



με τη βοήθεια ενός ζαριού μετακινούνται γύρω από ένα πίνακα απαντώντας σε ερωτήσεις τεσσάρων διαφορετικών κατηγοριών. Οι χρήστες που έπαιξαν το παιχνίδι κατά τη διάρκεια της δοκιμής - αξιολόγησης, ανέφεραν ότι διασκέδασαν το παιχνίδι αλλά και γνώρισαν τη βιβλιοθήκη. Οι δημιουργοί διανέμουν το παιχνίδι με άδεια χρήσης Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License, ώστε οποιοσδήποτε (ή οποιαδήποτε βιβλιοθήκη) να μπορεί να το κατεβάσει, να το τροποποιήσει και να το χρησιμοποιήσει σύμφωνα με τους στόχους και τις ανάγκες του.

***Huddersfield university - Library Game,***

Στο σημείο αυτό κρίνουμε σκόπιμο να αναφερθούμε εκτενέστερα και στο Library Game, το οποίο δημιουργήθηκε από τη βιβλιοθήκη του Πανεπιστημίου του Huddersfield και προσφέρει δύο παιχνίδια. Το Orange Tree, που είναι ειδικά σχεδιασμένο για δημοτικές βιβλιοθήκες και το Lemon Tree που απευθύνεται σε πανεπιστημιακές. Πρόκειται για παιχνίδια που συνδέονται με το σύστημα διαχείρισης της βιβλιοθήκης και συλλέγουν δεδομένα αναφορικά με τις δραστηριότητες των χρηστών (δανεισμούς, επιστροφές κ.ά.). Οι χρήστες της βιβλιοθήκης που συμμετέχουν στο παιχνίδι επιβραβεύονται για τις υπηρεσίες που χρησιμοποιούν π.χ. κερδίζοντας πόντους με το δανεισμό ενός συγκεκριμένου βιβλίου, με την επίσκεψη στη βιβλιοθήκη σε περίεργες ώρες κ.ά. Ο Iman Moradi, ένας από τους δημιουργούς του παιχνιδιού, επισημαίνει ότι ο κύριος σκοπός του είναι η παρότρυνση των χρηστών να διαβάζουν περισσότερο, η παροχή βοήθειας προς τους βιβλιοθηκονόμους ώστε να καταλάβουν καλύτερα τους χρήστες τους και η δυνατότητα στους ίδιους τους χρήστες να μοιραστούν τις εμπειρίες τους.



## 2.6 Αναφορές

Lau, Jesus, IFLA (2006) . [Guidelines on Information Literacy for Lifelong Learning](#).

Mary J. Snyder Broussard, (2012) "*Digital games in academic libraries: a review of games and suggested best practices*", Reference Services Review, Vol. 40 Iss: 1, pp.75 – 89.

Associations and Institutions. 29 Μαΐου 2007 <http://www.ifla.org/VII/s42/pub/IL-Guidelines2006.pdf>

AASL, (American Association of School Librarians) Η Αμερικανική Ένωση Σχολικών Βιβλιοθηκονόμων (AASL) είναι η μόνη εθνική οργάνωση επαγγελματική ένταξη επικεντρώθηκε σε σχολείο των βιβλιοθηκονόμων και η σχολική κοινότητα βιβλιοθηκών. AASL έχει πάνω από 7.000 μέλη και υπηρετεί το σχολείο των βιβλιοθηκονόμων στις Ηνωμένες Πολιτείες, τον Καναδά, και σε όλο τον κόσμο. επίσημη ιστοσελίδα <http://www.ala.org/aasl/about>

ACRL. (2004, February). Information Literacy Competency Standards for Higher Education. Ημερομηνία ανάκτησης 26 Ιουλίου 2004, από το δικτυακό τόπο: <http://www.ala.org/ala/acrl/acrlstandards/informationliteracycompetency.htm>

The Society of College, National and University Libraries (SCONUL). Αντιπροσωπεύει όλες τις πανεπιστημιακές βιβλιοθήκες στο Ηνωμένο Βασίλειο και την Ιρλανδία, ανεξάρτητα από την ομάδα της αποστολής, καθώς και οι εθνικές βιβλιοθήκες και πολλά από τα κολλέγια της τριτοβάθμιας εκπαίδευσης του Ηνωμένου Βασιλείου. Από την επίσημη ιστοσελίδα <http://www.sconul.ac.uk/page/about-sconul>  
SCONUL. (2001). The Seven Pillars of Higher Education. London: SCONUL

Bundy, A. (2004). Australian and New Zealand Information Literacy Framework: Principles, Standards and Practice. Ημερομηνία ανάκτησης 26 Ιουλίου 2004, από το δικτυακό τόπο του Australian and New Zealand Institute for Information Literacy: <http://www.caul.edu.au/infoliteracy/InfoLiteracyFramework.Pdf>

Information Literacy Standards. (2001). Ημερομηνία ανάκτησης 27 Ιουλίου 2004, από το δικτυακό τόπο του Council of Australian University Librarians: <http://www.caul.edu.au/caul-doc/InfoLitStandards2001.doc>

Genser, Stefan (2012). A Theory of The Effects of Gamification on Social Software: Bachelor thesis. p. 19.

Bohyun Kim (2012) [Harnessing the power of game dynamics1:Why, how to, and how not to gamify the library experience.](#) College & Research Libraries News vol. 73 no. 8 465-469.

Skelton Val (2012) [Gamification, augmented reality and new generation library software: The role of new technologies in libraries.](#) Proceedings of Internet Librarian International Conference 2012, London, UK, October 29–31.

*New York Public Library: Find the Future* Διαθέσιμο στην ηλεκτρονική διεύθυνση: [Find the future: The game](#)

*Carnegie Mellon University: Library Arcade.* Διαθέσιμο στην ηλεκτρονική διεύθυνση: <https://libwebspace.library.cmu.edu/libraries-and-collections/Libraries/etc/index.html>

The University of North Carolina – Greensboro. *The Information Literacy Game.* Διαθέσιμο στην ηλεκτρονική διεύθυνση: [The Information Literacy Game](http://library.uncg.edu/game/)  
<http://library.uncg.edu/game/>

*Library Game.* Διαθέσιμο στην ηλεκτρονική διεύθυνση: <http://librarygame.co.uk/> Επίσης, περισσότερες πληροφορίες στην ηλεκτρονική διεύθυνση: <http://librarygame.tumblr.com/>

Deterding, Sebastian / Khaled, Rilla / Nacke, Lennart E. / Dixon, Dan: “Gamification: Toward a Definition”, Vancouver, 2011 (CHI 2011 Conference).

Rubin, Richard, Foundations of library and information science. New York: Neal-Schuman Publishers, c1998.

*Πρότυπα δεξιοτήτων πληροφοριακής παιδείας στην ανώτατη εκπαίδευση.* Chicago: American Library Association, 2000. American College and Research Libraries Association. 15 Δεκ 2006. <http://www.ala.org/ala/acrl/acrlstandards/infolit.PDF>

Bawden, D. (2001, March). Information and Digital Literacies: A Review of Concepts. στο *Journal of Documentation*, No. 57, σσ. 218-259.

Behrens, S.J. (1994, April). A Conceptual Analysis and Historical Overview of Information Literacy. *College and Research Libraries*, Vol. 55, No. 4, σσ.309-322.

Bligh, Donald A. (2000). What's the Use of Lectures? Στο Gibbs, *Teaching in Higher Education: Theory and Evidence*. San Francisco: Jossey-Bass Publishers, σελ. 346.

Bruce, C. and Candy, P. (Eds.) (2000). *Information Literacy Around the World: Advances in Programs and Research*. Wagga, Wagga, Australia, Centre for Information Studies Charles Sturt University.

Bruce, C. (1997). *The Seven Faces of Information Literacy: Seven Faces of Information Literacy*. AULSIB Press, Adelaide Auslib Press.

Νικητάκης, Μιχάλης and Βραχλιώτη, Μαρία (2006). Πληροφοριακή Παιδεία: μέθοδοι και η ελληνική πραγματικότητα στο χώρο των ακαδημαϊκών βιβλιοθηκών. Στο *Πρακτικά Ημερίδας Οι ακαδημαϊκές βιβλιοθήκες της περιφέρειας ως μοχλοί υποστήριξης των εκπαιδευτικών και πολιτιστικών δράσεων των δημοσίων και σχολικών βιβλιοθηκών*, 12 Απριλίου 2006, Καλαμάτα. Διαθέσιμο στο: <http://eprints.rclis.org/archive/00011903/01/Nikitakis-Brachlioti.pdf>

## 3. Android development - History of infrastructure

### 3.1 Τι είναι το Android

Το Android είναι ένα λειτουργικό σύστημα για <<έξυπνες>> συσκευές γνωστές και ως smartphones βασισμένο στο Linux. Αρχικά αναπτύχθηκε από την Google και επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα ανοιχτού λογισμικού χρησιμοποιώντας την γλώσσα προγραμματισμού JAVA. Το λογότυπο είναι ένα ρομποτάκι χρώματος ανοιχτό πράσινο.



*Εικόνα 1: Android Logo*

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. 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 λογισμικό για κινητά τηλέφωνα. Αυτή ήταν η αρχή της φημολογίας περί σχεδίων της Google να διεισδύσει στην αγορά κινητής

τηλεφωνίας. Σ την 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 εταιρους 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 ανέπτυξε 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 NexusOne που χρησιμοποιεί το open source λειτουργικό σύστημα Android. Η συσκευή κατασκευάστηκε από την HTC , και έγινε διαθέσιμη στις 5 Ιανουαρίου 2010.

## 3.2 Χαρακτηριστικά του Android

Η πλατφόρμα είναι προσαρμόσιμη σε μεγαλύτερες αναλύσεις (VGA, QVGA).

- Χρήση βάσης δεδομένων SQLite.
- Τεχνολογίες συνδεσιμότητας GSM/CDMA, EDGE, EV-DO, UMTS, Bluetooth, Wi-Fi.
- Διαθέσιμους τρόπους ανταλλαγής μηνυμάτων SMS και MMS.
- Διαθέτει web browser βασισμένο στην τεχνολογία Web Kit.
- Λογισμικό γραμμένο σε JAVA το οποίο γίνεται compile στην εικονική μηχανή Dalvik και ART.
  - Υποστηρίζει τις ακόλουθες επεκτάσεις στατικής κινούμενης εικόνας [H.263](#), [H.264](#), [3GP](#), [MP4](#), [AMR](#), [AMR-WEB](#), [JPEG](#), [PNG](#), [GIF](#), [BMP](#).
  - Αρχεία ήχου [AAC](#), [MP3](#), [MIDI](#), [OGG](#), [WAV](#).
  - Χρησιμοποιεί ψηφιακές βιβλιοθήκες και τρισδιάστατα γραφικά χρησιμοποιώντας την τεχνολογία OpenGL ES 1.0 .

Μπορεί να διαθέτει γυροσκόπιο, κάμερα, GPS, αισθητήρες επιτάχυνσης, μαγνητόμετρο, δισδιάστατους και τρισδιάστατους επιταχυντές γραφικών

- Υποστηρίζει την τεχνολογία multi-touch

## 3.3 Γιατί επιλέγω Android

Υπάρχουν πολλά λειτουργικά συστήματα στην αγορά. Υπάρχουν τα Windows Phone, το Symbian, το iOS, το Blackberry, το Android, το Web OS και άλλα που δεν αναφέρω εδώ. Από όλα τα λειτουργικά συστήματα που υπάρχουν στην αγορά, χωρίς

αμφιβολία, τα δύο επικρατέστερα είναι το Android και το iOS. Google και Apple αντίστοιχα, δύο εταιρίες με παντελώς διαφορετική φιλοσοφία αλλά και διαφορετικές προσεγγίσεις στο θέμα του software. Το Android από την μια υποστηρίζει τον “ανοιχτό” κώδικα και δίνει την ελευθερία στον οποιονδήποτε έχει τις απαραίτητες γνώσεις ή μη να επεμβαίνει στο λειτουργικό και να κάνει ότι θέλει και από την άλλη έχουμε την Apple που υποστηρίζει το ακριβώς αντίθετο καθώς θέλει να έχει τον πλήρη έλεγχο τόσο σε hardware όσο και σε software επίπεδο. Προσωπικά επιλέγω το Android της Google.



*Εικόνα 2: Android vs Apple*



### 3.3.1 Εφαρμογές Android

Το Android έχει μια μεγάλη κοινότητα προγραμματιστών που γράφουν εφαρμογές, οι οποίες επεκτείνουν τη λειτουργικότητα των συσκευών. Οι εφαρμογές γράφονται σε μια προσαρμοσμένη έκδοση της JAVA και μπορεί κανείς να κατεβάσει από το online κατάστημα Google Play (πρώην Android Market) της Google όπως και από άλλα sites. Μέχρι τον Φεβρουάριο του 2012 περισσότερες από 450000 εφαρμογές ήταν διαθέσιμες για Android ενώ εκτιμάτε ότι ο αριθμός των downloads από το Android Market μέχρι το Δεκέμβριο του 2011 είχε υπερβεί τα 10 δισεκατομμύρια. Το Android είναι η πρώτη σε πωλήσεις παγκοσμίως πλατφόρμα για smartphones καθώς μέχρι το Φεβρουάριο του 2012 μετρούσε περισσότερες από 300 εκατομμύρια συσκευές σε χρήση.

## 3.4 Ιστορική αναδρομή, πορεία εξέλιξης του Android.

### Διανομές λογισμικού:

#### 3.4.1 Android 1.5 Cupcake

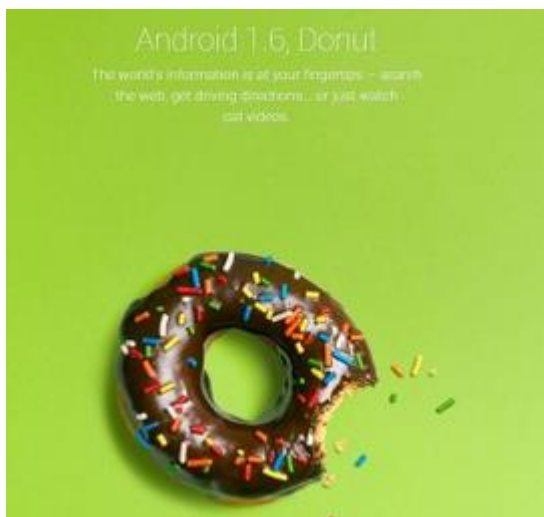
Αρχικά να αναφέρουμε πως θα ήταν ψέμα να πούμε πως το **Android 1.5 Cupcake** ή οποιαδήποτε άλλη έκδοση του Android είχε μια και μοναδική βελτίωση, αλλά αν πρέπει να επιλέξουμε το χαρακτηριστικό που το ανέδειξε, είναι η εισαγωγή ψηφιακού πληκτρολογίου. Ίσως σας φαίνεται περίεργο, αλλά το 2008/2009, τα περισσότερα smartphones είχαν φυσικό πληκτρολόγιο QWERTY.



Εικόνα 3: Android 1.5 Cupcake

### 3.4.2 Android 1.6 Donut

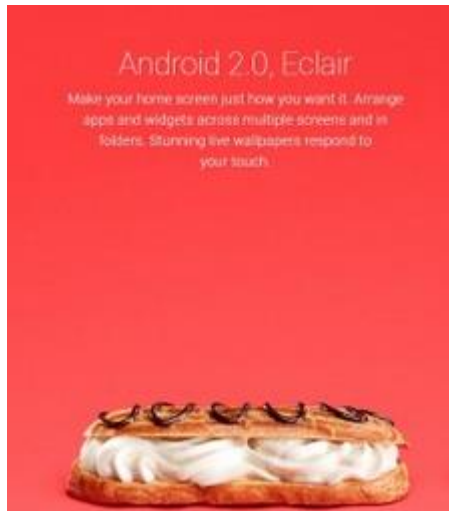
Το **Android 1.6 Donut**, είχε αλλαγές στο εσωτερικό του λειτουργικού και αυτό είχε ως αποτέλεσμα να υποτιμηθεί αρκετά από τους καταναλωτές. Παρόλα αυτά, ήταν πολύ σημαντική αναβάθμιση, καθώς έφερε υποστήριξη διαφορετικών αναλύσεων οθόνης, ανεξάρτητα από την πυκνότητα pixel, κάτι που καθόρισε πραγματικά το μέλλον των Android συσκευών.



Εικόνα 4: Android 1.6 Donut

### 3.4.3 Android 2.0 Eclair

Ξεχωρίζοντας ένα μόνο από τα χαρακτηριστικά του **Eclair**, είναι αρκετά δύσκολο, αλλά αν πρέπει να επιλέξουμε κάτι, αυτό είναι οι βελτιώσεις στην κάμερα. Μέχρι τότε το Android δεν είχε υποστήριξη LED flash στην κάμερα, ούτε δυνατότητες που σήμερα θεωρούμε δεδομένες, όπως χρωματικά εφέ, λειτουργία σκηνών, λειτουργίες εστίασης, ή ακόμη και ρύθμιση της ισορροπίας λευκού.

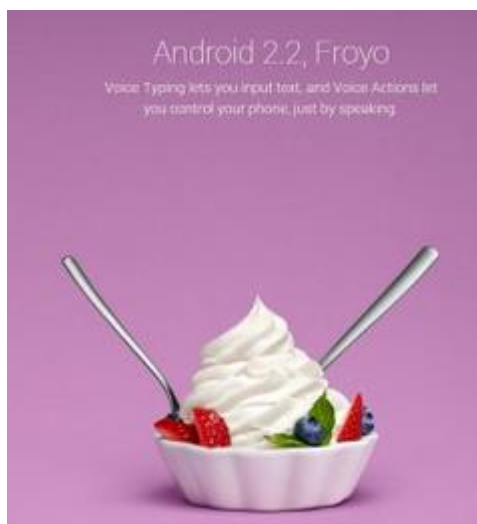


Εικόνα 5: Android 2.0 Eclair

### 3.4.4 Android 2.2 Froyo

Για το **Froyo** έχουμε να επιλέξουμε ανάμεσα σε δύο χαρακτηριστικά και αυτά είναι η ταχύτητα και υποστήριξη για Wi-Fi hotspots. Αλλά δεδομένου ότι πρέπει να επιλέξουμε μόνο ένα, θα πρέπει να πάμε με την ταχύτητα, διότι το Wi-Fi Hotspot δεν ήταν διαθέσιμο παγκοσμίως. Στο Froyo λοιπόν, ορίστηκε σαν runtime ο Dalvik, που έφερε σοβαρές βελτιώσεις στην απόδοση της συσκευής μας, που σύμφωνα με την Google γίνεται από 2 έως 5 φορές ταχύτερη.

Μία από τις πιο εξέχουσες αλλαγές στην απελευθέρωση Froyo ήταν tethering USB και λειτουργία Wi-Fi hotspot. Άλλες αλλαγές περιλαμβάνουν υποστήριξη για το Android Cloud στα Μηνύματα (C2DM) υπηρεσία συσκευής, επιτρέποντας τα Push Notifications, Πρόσθετες βελτιώσεις ταχύτητα εφαρμογής, υλοποιείται μέσω σύνταξη KOE και εμφανίζεται μέσα σε εφαρμογές όπως top-of-the-οθόνη πανό.



*Εικόνα 6: Android 2.2 Froyo*

### 3.4.5 Android 3.0 Honeycomb

Χωρίς αμφιβολία, το **Honeycomb**, ήταν μια από τις λιγότερο δημοφιλείς εκδόσεις του Android, αλλά αυτό σίγουρα δεν έχει να κάνει με την συνεισφορά του στην πλατφόρμα. Το Android, είχε φρικτή εμπειρία χρήσεως στα tablets, κάτι που άλλαξε με την κυκλοφορία του Honeycomb, όπου ήταν σχεδιασμένο για συσκευές με μεγαλύτερη οθόνη.

Μια ακόμα διαθέσιμη έκδοση της πλατφόρμας Android που έχει σχεδιαστεί για συσκευές με τα μεγαλύτερα μεγέθη οθόνης, ιδιαίτερα δισκία. Εκτός από την προσθήκη νέων χαρακτηριστικών, Honeycomb εισήγαγε ένα νέο λεγόμενο « interface theme και ένα μοντέλο αλληλεπίδρασης που χτίστηκε πάνω στα κύρια χαρακτηριστικά του Android, όπως το multitasking, τα notifications και τα widgets.



Εικόνα 7: Android 3.0 Honeycomb

### 3.4.6 Android 4.0 Ice Cream Sandwich

Το **Ice Cream Sandwich**, έφερε πραγματικά τεράστια ανανέωση στον σχεδιασμό του Android με το Holo UI. Το ICS, ήταν η πρώτη έκδοση του Android που έδωσε μοντέρνα χαρακτηριστικά στο λειτουργικό, που ακόμη και σήμερα συναντάμε πολλά στοιχεία του στις Android συσκευές.

Android 4.0 βασίζεται στις σημαντικές αλλαγές που έγιναν από το tablet-only release το Android 3.0 "Honeycomb", σε μια προσπάθεια να δημιουργήσει μια

ενοποιημένη πλατφόρμα για τα δύο smartphones και tablets, καθώς και την απλούστευση και τον εκσυγχρονισμό της συνολικής εμπειρίας του Android γύρω από ένα νέο σύνολο κατευθυντήριων γραμμών της ανθρώπινης διεπαφής. Ως μέρος αυτών των προσπαθειών, το Android 4.0 εισήγαγε μια νέα οπτική εμφάνιση με την κωδική ονομασία "Holo», το οποίο είναι χτισμένο γύρω από ένα καθαρότερο, μινιμαλιστική σχεδίαση, και μια νέα προεπιλεγμένη γραμματοσειρά που ονομάζεται Roboto.



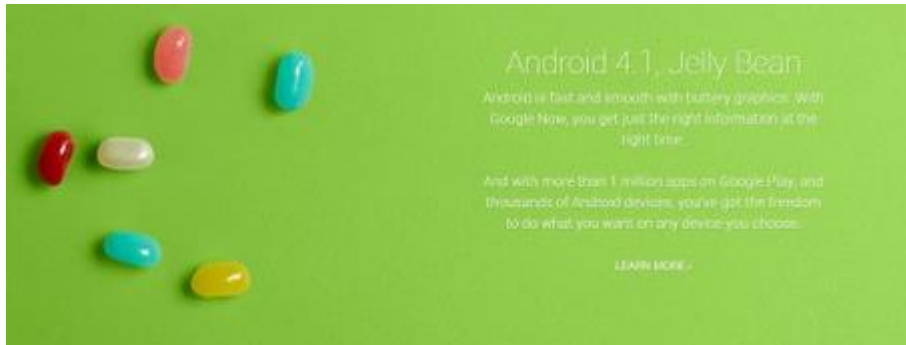
Εικόνα 8: Android Ice Cream Sandwich

### 3.4.7 Android 4.1 Jelly Bean

Το **Jelly Bean** είναι μεγάλη υπόθεση για πολλούς λόγους, ιδιαίτερα αν σκεφτούμε ότι κατέχει περίπου το 50% των Android συσκευών. Οι συνεισφορές του στο Android είναι πολλές, αλλά αυτή που ξεχώρισε και αναπτύσσετε μέχρι και σήμερα, είναι το Google Now. Ο ψηφιακός βοηθός της Google όχι μόνο παρέχει γρήγορες απαντήσεις για τα πάντα, αλλά και εμφανίζει νέα και ειδήσεις σύμφωνα με τα ενδιαφέροντα μας. Το Google Now έχει οργανωμένες κάρτες, που μας δίνουν πληροφορίες σχετικές με την ημέρα, τον καιρό αλλά και την περιοχή που βρισκόμαστε.

Το όνομα που δίνεται σε τρεις μεγάλες κυκλοφορίες σημείο του Android κινητό λειτουργικό σύστημα που αναπτύχθηκε από την Google, που εκτείνονται σε εκδόσεις μεταξύ 4.1 και 4.3.1. Το πρώτο από αυτά τα τρία, 4.1, παρουσιάστηκε σε συνέδριο / O developer της Google που τον Ιούνιο του 2012, εστιάζοντας σε βελτιώσεις στην απόδοση σχεδιαστεί για να δώσει το λειτουργικό σύστημα μια ομαλότερη και πιο δεκτικά αίσθηση, βελτιώσεις στο

σύστημα κοινοποίησης που επιτρέπει για "επεκτάσιμη" κοινοποιήσεων με κουμπιά δράσης, και άλλες εσωτερικές αλλαγές. Δύο ακόμη απελευθερώσεις έγιναν υπό την ονομασία Jelly Bean τον Οκτώβριο του 2012 και τον Ιούλιο του 2013 αντίστοιχα, συμπεριλαμβανομένων των 4,2-η οποία περιλαμβάνει περαιτέρω βελτιστοποιήσεις, υποστήριξη multi-user για τις ταμπλέτες, widgets οθόνη κλειδώματος, γρήγορη ρυθμίσεις και προφυλάξεις οθόνης, και 4.3-περιείχαν περαιτέρω βελτιώσεις και ενημερώσεις για την υποκείμενη πλατφόρμα Android.



Εικόνα 9: Android 4.1 Jelly Bean

### 3.4.8 Android 4.4 KitKat

Με το **KitKat**, η Google όχι απλά ενίσχυσε το UI, αλλά και ανέδειξε την δύναμη του λειτουργικού ακόμη και σε φθηνές συσκευές. Χάρη στις πολλές βελτιώσεις επιδόσεων που έγιναν στο KitKat, η Google υποστήριξε πως ακόμη και συσκευές με μόλις 512MB RAM θα είναι σε θέση να τρέξουν το KitKat απροβλημάτιστα. Αυτό ήταν μια αρκετά σοβαρή υπόθεση, δεδομένου ότι οι εταιρείες έδειχναν αδιαφορία στις οικονομικές συσκευές. Έτσι πλέον, δίνεται η δυνατότητα και σε αναπτυσσόμενες χώρες να αποκτήσουν Android συσκευές και κατά συνέπεια να αυξηθούν και οι πωλήσεις φθηνότερων συσκευών.



Εικόνα 10: Android 4.4 KitKat

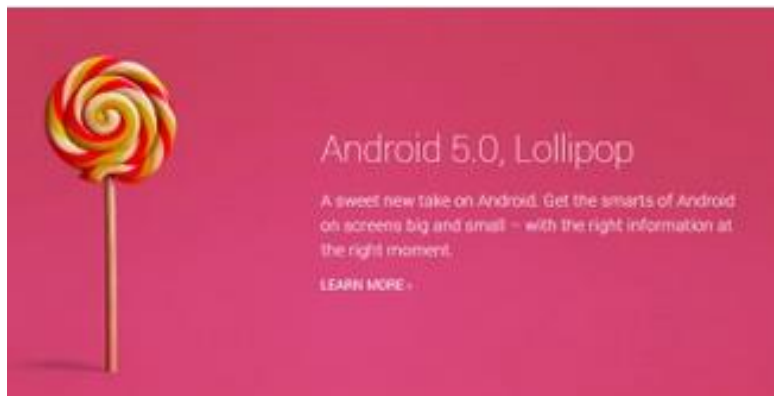
### 3.4.9 Android 5.0 Lollipop

Με το **Lollipop** να είναι η πιο ελκυστική έκδοση του Android που έχει υπάρξει ποτέ! Με το Material Design της Google, η εμφάνιση του λειτουργικού έγινε επίπεδη, με περισσότερα χρώματα και φυσικά με πάρα πολλά εφέ κίνησης.

Μία από τις πιο εξέχουσες αλλαγές στην απελευθέρωση Lollipop είναι ένα επανασχεδιασμένο περιβάλλον εργασίας χρήστη χτισμένο γύρω από μια σχεδιαστική γλώσσα γνωστή ως «υλικό σχεδιασμού». Άλλες αλλαγές περιλαμβάνουν βελτιώσεις



στις κοινοποιήσεις, το οποίο μπορεί να προσεγγιστεί από την lockscreen και εμφανίζεται μέσα σε εφαρμογές όπως top-of-the-οθόνη πανό. Η Google έκανε επίσης εσωτερικές αλλαγές στην πλατφόρμα, με το Android Runtime (ART) επίσημα αντικατάσταση Dalvik για να βελτιώσει την απόδοση των εφαρμογών, καθώς και με τις αλλαγές που αποσκοπούν στη βελτίωση και τη βελτιστοποίηση της χρήσης της μπαταρίας.



*Εικόνα 11: Android 5.0 Lollipop*

### 3.4.10 Android 6.0 Marshmallow

Το Android 6.0 Marshmallow[6] λανσάρεται ως “η πιο περιποιημένη έκδοση Android που έχουμε δει μέχρι σήμερα”, υπάρχουν ενδιαφέρουσες προσθήκες και αλλαγές τόσο στην εμφάνιση, όσο και στη λειτουργικότητα.

- Υποστήριξη USB Type-C
- Επιστροφή του Silence Mode
- Μόνιμη εμφάνιση των εργαλείων Copy-Paste επάνω από το κείμενο
- Δυνατότητα ρύθμισης του ήχου σε διαφορετικά επίπεδα για multitasking (ήχος κλήσης, ήχος εφαρμογής, ήχος ειδοποιήσεων)
  - Κάθετη εμφάνιση εφαρμογών με δυνατότητα αναζήτησης και επισήμανσης αγαπημένων
- Εμφάνιση όλων των επιλογών στις Ρυθμίσεις
- Νέα λίστα εμφάνισης της κατανάλωσης μνήμης RAM για κάθε εφαρμογή

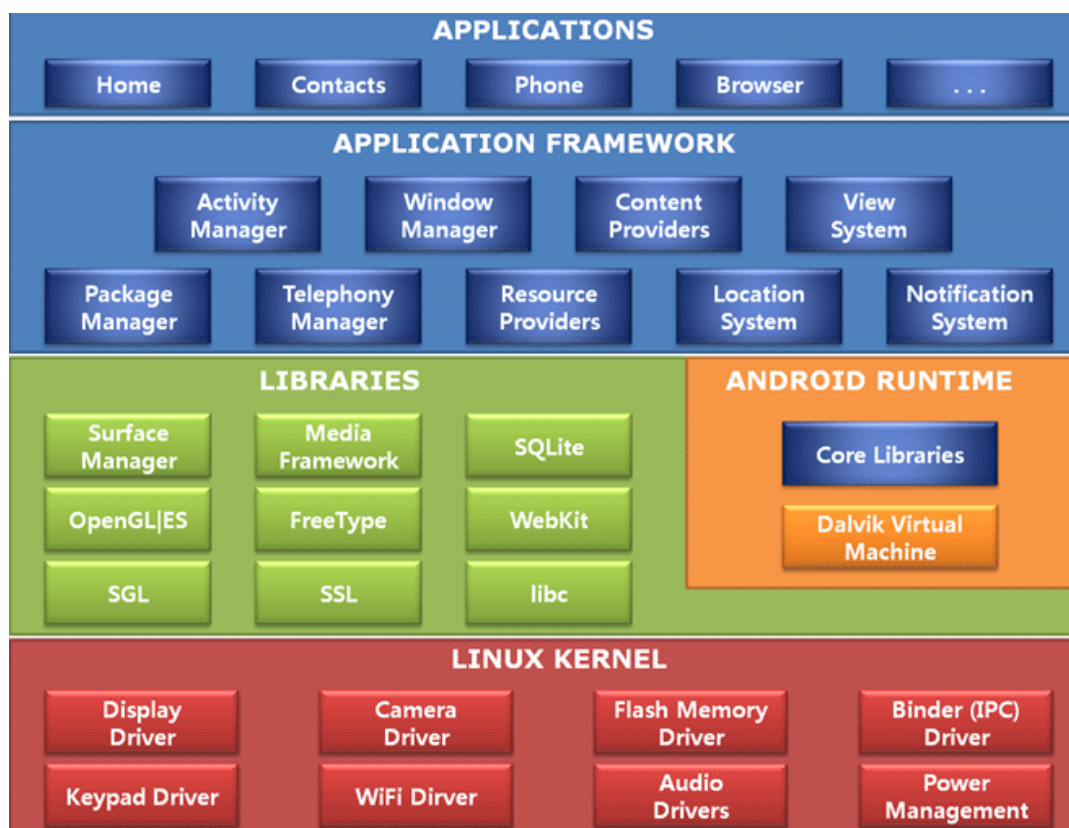


Εικόνα 12: Android 6.0 Marshmallow

## 3.5 Η αρχιτεκτονική του Android

### 3.5.1 Σχήμα αρχιτεκτονικής τεχνολογίας Android

Android τρέχει βασικά για **DVM** (Dalvik Virtual Machine), η οποία είναι **runtime Android** και φορτώνεται στο Linux kernel. Το παρακάτω σχήμα απεικονίζει όλα τα στοιχεία. Περαιτέρω θα εξηγήσω κάθε ένα από αυτά από πάνω προς τα κάτω.



Εικόνα 13: Android Architecture Layers

### 3.5.2 Application Framework

Πλαίσιο εφαρμογής βρίσκεται στην κορυφή μητρική βιβλιοθήκες, Android runtime και Linux kernel. Το πλαίσιο αυτό έρχονται προ-εγκατεστημένο με υψηλού επιπέδου δομικά στοιχεία που οι προγραμματιστές μπορούν να χρησιμοποιήσουν για να προγραμματίσετε τις εφαρμογές. Μετά είναι τα πιο σημαντικά συστατικά πλαίσιο εφαρμογής για την εφαρμογή μας και την ανάπτυξη του Android σε γενικές γραμμές.

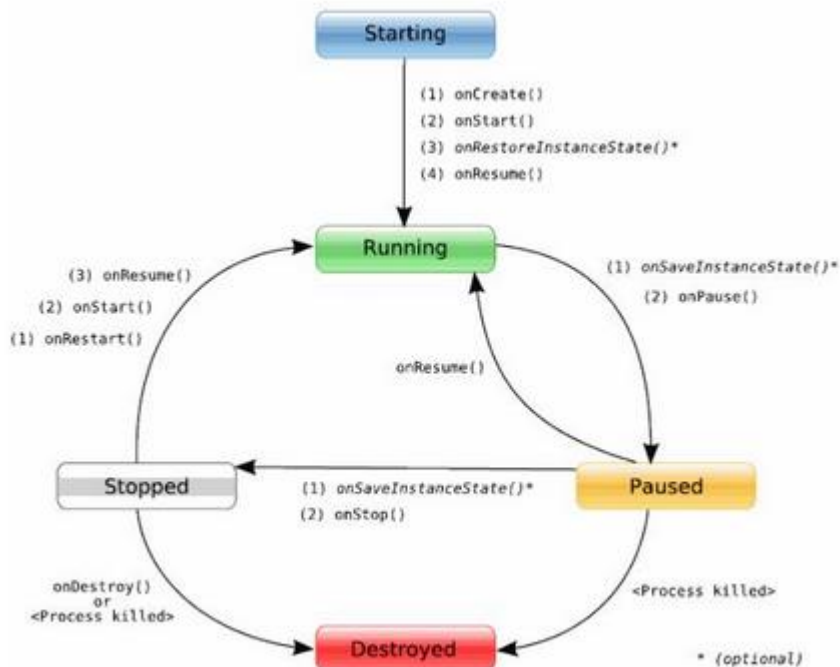
&lt;



Εικόνα 14:Application Framework

### 3.5.2.1 Activity lifecycle

Το Activity είναι ένα ενιαίο εστιασμένο πράγμα. Τα Activities μπορούν να τρέξουν στο προσκήνιο δίνοντας άμεση αλληλεπίδραση με το χρήστη, π.χ. τρέχον παράθυρο / καρτέλα, που μπορεί να τρέξει ως background services ή μπορούν να ενσωματωθούν σε άλλες activities.



Εικόνα 15: Android Activity Lifecycle

Το σύνολο του κύκλου ζωής [8] ορίζεται από ορισμένες μεθόδους ή καταστάσεις, όπως φαίνεται στην εικόνα 16.

```
public class Activity extends XYZ {  
    protected void onCreate(Bundle savedInstanceState);  
  
    protected void onStart();  
  
    protected void onResume();  
  
    protected void onPause();  
  
    protected void onStop();  
  
    protected void onDestroy();  
}
```

Εικόνα 16: Important methods of Activity

Όλες οι τάξεις δραστηριοτήτων που πρέπει να έχει μια αντίστοιχη <δραστηριότητας> δήλωση AndroidManifest.xml.

### 3.5.2.2 Contents Provider

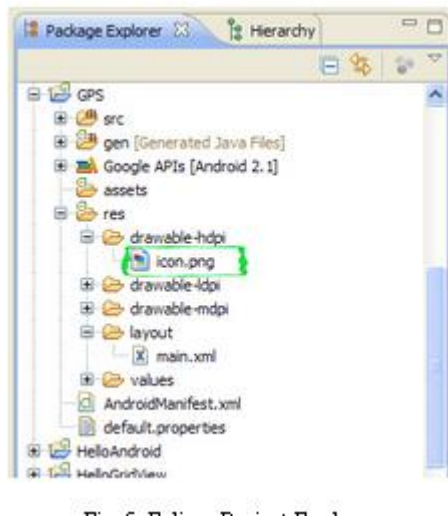
Περιεχόμενο παρέχει στοιχεία λαβές σε όλες τις εφαρμογές σε παγκόσμιο επίπεδο. Android έρχεται με μια σειρά από χτίστηκε το παρόχους περιεχομένου να χειριστεί δεδομένα πολυμέσων ή επαφές κλπ προγραμματιστές μπορούν να κάνουν τη δική τους παρόχους για ευελιξία ή μπορούν να ενσωματώσουν τα δεδομένα τους σε έναν από τους υφιστάμενους παρόχους. Για τη συγκεκριμένη εφαρμογή μας, μας ενδιαφέρει το περιεχόμενο: // browser για να έχετε πρόσβαση σε online δεδομένα μέσω της διεπαφής του προγράμματος περιήγησης.

### 3.5.2.3 View System

Το View System δεσμεύει όλες τις τάξεις μαζί που χειρίζονται γραφική διεπαφή χρήστη (GUI) που σχετίζονται με στοιχεία. Όλα απόψεις στοιχεία διατεταγμένα σε ένα ιεραρχικό τρόπο ενιαίο δέντρο. Μπορούν να κληθεί από έναν κώδικα java ή περιλαμβάνονται σε αρχεία διάταξης XML. Ένα καλό πράγμα που παρατήρησα σχετικά με την ανάπτυξη του Android είναι εκτεταμένη χρήση των αρχείων XML. Αυτά τα αρχεία παρέχουν μια ωραία αφαίρεσης μεταξύ κωδικό backend java και τα στοιχεία διάταξης. Σχεδιάζοντας στοιχεία που σχετίζονται UI γίνεται με τον ίδιο τρόπο, όπως με βάση HTML web σχεδιασμό.

### 3.5.2.4 Resource Manager

Διαχείριση πόρων χειρίζεται όλα τα πράγματα μη-κώδικα. Αυτά μπορεί να είναι οτιδήποτε, κυμαίνονταν από εικόνες, γραφικά ή κείμενο. Οι πόροι αυτοί βρίσκονται κάτω από τον RES κατάλογο, όπως μπορεί να δει κάτω από το Eclipse project Explore στην παρακάτω εικόνα.



Εικόνα 17: Eclipse Project Explorer

### 3.5.2.5 Location Services

Αυτό το πακέτο υποστηρίζει παροχή τοποθεσίας location based, όπως το GPS και διευρυσμένων δικαιωμάτων παροχή τοποθεσίας, όπως το κινητό τηλέφωνο τριγωνισμού. Το Location Manager είναι το κεντρικό στοιχείο του πλαισίου τοποθεσίας. Αυτή η υπηρεσία συστήματος παρέχει μια υποκείμενη API για την πρόσβαση σε πληροφορίες τοποθεσίας της συσκευής. Εκτός Location Manager κλάσης, είναι πολλές άλλες σημαντικές κλάσεις από android.location που είναι σημαντικά για εφαρμογές location aware όπως Geocoder, GPS Satellite και LocationProvider.

### 3.5.2.6 Βιβλιοθήκες (Libraries)

Στο δεύτερο επίπεδο της στοίβας έχουμε τις βιβλιοθήκες του Android. Αυτός ο τομέας παρέχει διάφορες C / C ++ βιβλιοθήκες που μπορούν να χρησιμοποιηθούν για το Android. Όλες οι βιβλιοθήκες σε αυτόν τον τομέα μπορεί να χρησιμοποιηθεί από τους προγραμματιστές μέσω του πλαισίου εφαρμογής. Οι τυπικές βιβλιοθήκες C σύστημα βασίζεται σε BSD (Berkeley Διανομής Λογισμικού) αναθεωρήθηκαν για να είναι κατάλληλο για συσκευές που βασίζονται σε Linux. Η βιβλιοθήκη πολυμέσων με

βάση OpenCore της PacketVideo υποστηρίζει MPEG4, H.264, MP3, AAC, AMR, JPG και PNG αρχεία. Ο διαχειριστής επιφάνειας υποστηρίζει 2D και 3D γραφικά, ενώ το WebKit υποστηρίζει τις λειτουργίες του προγράμματος περιήγησης. Το SQLite είναι επίσης διαθέσιμο ως μια μηχανή βάσης δεδομένων η οποία μπορεί να χρησιμοποιηθεί από εφαρμογές.

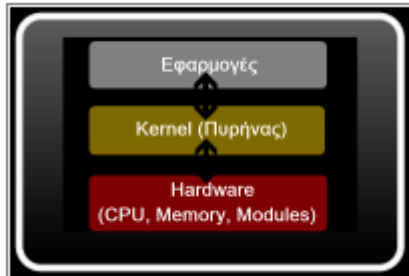
Μερικές από τις κύριες βιβλιοθήκες του Android είναι:

- ✓ **System C library** – μια ενσωμάτωση της standard βιβλιοθήκης συστήματος της C(libc) τροποποιημένη για κινητές συσκευές βασισμένες στο Linux.
- ✓ **Βιβλιοθήκες Πολυμέσων** – Υποστηρίζει αναπαραγωγή και εγγραφή πολλών δημοφιλών μέσων ήχου και εικόνας, όπως: MPEG4, H.264, MP3, AAC, AMR, JPG, και PNG
- ✓ **Surface Manager** – διαχειρίζεται την πρόσβαση στο υποσύστημα προβολής, και συνθέτει πρόσκοπτα δισδιάστατα και τρισδιάστατα επίπεδα γραφικών τα οποία προέρχονται από πολλαπλές εφαρμογές.
- ✓ **LibWebCore** – μια μοντέρνα μηχανή υποστήριξης πλοήγηση στο διαδίκτυο (browserengine) η οποία χρησιμοποιείτε και από τον ενσωματωμένο browser του Android αλλά και από τις WebViews που ενσωματώνονται στις εφαρμογές.
- ✓ **SGL** – η γνωστή μηχανή δισδιάστατων γραφικών
- ✓ **Βιβλιοθήκες 3D** – μια υλοποίηση βασισμένη στα APIs του OpenGL ES 1. Οι βιβλιοθήκες χρησιμοποιούν είτε τρισδιάστατη επιτάχυνση υλικού, όπου αυτή είναι διαθέσιμη, είτε μια υψηλά βελτιωμένη τρισδιάστατη επιτάχυνση λογισμικού σε περίπτωση που η πρώτη δεν είναι διαθέσιμη.
- ✓ **FreeType** – παρέχει ευκρίνεια γραφικών στα bitmaps και τις γραμματοσειρές των εφαρμογών του συστήματος.
- ✓ **SQLite** – μια πανίσχυρη και συνάμα πολύ ελαφριά σχεσιακή βάση δεδομένων Android Runtime

Παρά το γεγονός ότι το Android έχει αναπτυχθεί σε γλώσσα Java, DVM (Dalvik Virtual Machine) χρησιμοποιείται αντί της JVM (Java Virtual Machine). Έτσι, ένα πηγαίο κώδικα (Java) συγκεντρώνονται σε ένα αρχείο τάξη (.class) με Java compiler το οποίο αργότερα μετασχηματίστηκε σε Dalvik εκτελέσιμο από τα εργαλεία DX. Ο λόγος για αυτό είναι ότι Dalvik έχει βελτιστοποιηθεί για να λειτουργεί με μικρού μεγέθους συσκευές με περιορισμένη μνήμη.

### 3.5.3 Linux Kernel

Android Kernel ονομάζουμε τον πυρήνα του λειτουργικού συστήματος και είναι υπεύθυνος για την διασύνδεση εφαρμογών με το hardware όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 18: Linux Based Kernel

### 3.5.4 ART και Dalvik

Android RunTime (ART) είναι η διαχείριση χρόνου εκτέλεσης που χρησιμοποιείται από εφαρμογές και ορισμένες υπηρεσίες του συστήματος στο Android. Το ART και ο προκάτοχός του Dalvik είχαν αρχικά δημιουργηθεί ειδικά για το Android project. ART ως runtime, εκτελεί τα Dalvik εκτελέσιμα αρχεία και τις προδιαγραφές bytecode Dex.

Τα ART και Dalvik είναι συμβατά runtimes που εκτελούν Dex bytecode, έτσι ώστε οι εφαρμογές που αναπτύχθηκαν για Dalvik θα πρέπει να εκτελούνται όταν τρέχει με την ART. Ωστόσο, ορισμένες τεχνικές που δουλεύουν σε Dalvik δεν λειτουργούν με ART.

Με άλλα λόγια, η Dalvik είναι η εικονική μηχανή που χρησιμοποιεί η κάθε εφαρμογή για να τρέξει. Έτσι γίνεται το λεγόμενο multi-tasking χωρίς οι εικονικές μηχανές να επηρεάζονται μεταξύ τους και με αποτέλεσμα οι εφαρμογές να εκτελούνται ταυτόχρονα. Τα αρχεία της εφαρμογής πρέπει να είναι σε μορφή .dex διότι η Dalvik δεν μπορεί να κάνει compile σε κώδικα byte έτσι ώστε να τα τρέξει το Virtual Machine. Μια από τις σημαντικότερες προσθήκες στην έκδοση 4.4.x ήταν η προσθήκη του Compiler ART πράγμα το οποίο πέρασε απαρατήρητο από πολλούς χρήστες κατά το Runtime της συσκευής αλλά και της εφαρμογής. Αρχικά ήρθε σε πειραματικό στάδιο πράγμα το οποίο σήμαινε ότι για να το ενεργοποιήσει κάποιος χρήστης θα έπρεπε να ανοίξει την κρυφή λειτουργία Developer Options και να αλλάξει το Runtime απο



Dalvik σε ART. Με την έκδοση 5.0 το ART έγινε πλέον το default runtime των συσκευών.

## 3.6 Βασική Δομή του Android project

Αφού έχουμε εγκαταστήσει όλα τα plug-ins που απαιτούνται για την ανάπτυξη ενός αρχείου Android, μπορεί τώρα να αρχίσετε να αναπτύξει μια εφαρμογή Android. Από το επάνω μενού, επιλέγω Αρχείο -> new project, και από το «παράθυρο New Project», επιλέγω "Android Project". Ακολουθώ τον οδηγό εγκατάστασης του project και μετά την ολοκλήρωση του οδηγού, θα έχετε μια βασική εφαρμογή του Android.

## 3.7 Κάθε project Android περιέχει:

### 3.7.1 src:

Αυτός ο φάκελος περιέχει τα αρχεία πηγαίου κώδικα της Java. Περιέχει όλα τα αρχεία .java στο κάθε package και είναι υπεύθυνος για τον κώδικα της εφαρμογής. Αυτά μετατρέπονται σε αρχεία .dex για να μπορούν να γίνουν bytecode από τους compilers Dalvik και ART.

### 3.7.2 Res:

Εδώ μπορούμε να αποθηκεύσει τα αρχεία των πόρων, όπως εικόνες, αρχεία XML για τον καθορισμό σχεδιαγράμματα, και ούτω καθεξής. Μέσα σε αυτό το φάκελο υπάρχουν επιπλέον φακέλους, όπως Drawable, Layout και Value.

#### 3.7.2.1 Drawable:

Εδώ έχουμε αποθηκεύσει τα διάφορα αρχεία γραφικών. Μπορούμε να δούμε τρεις τύπους σχεδιάσιμα φακέλους. Αυτό είναι επειδή υπάρχουν πολλές συσκευές Android με διαφορετικές αναλύσεις οθόνης. Από προεπιλογή, υπάρχουν πολλές εκδόσεις αυτού του φακέλου, όπως: σχεδιάσιμα-mdpi, σχεδιάσιμα-πυκνότητα οθόνης hdpi, και ούτω καθεξής. Αυτό είναι απαραίτητο, προκειμένου να προσαρμοστούν σε διαφορετικές αναλύσεις οθόνης.

### 3.7.2.2 Layout:

Αυτή είναι η θέση για τα αρχεία διάταξης XML. Διάταξη αρχεία είναι αρχεία XML που καθορίζουν το πώς διάφορα Android αντικείμενα (όπως πλαίσια κειμένου, κουμπιά, κλπ) οργανώνονται στην οθόνη.

### 3.7.2.3 Values:

XML αρχεία που αποθηκεύουν διάφορες τιμές συμβολοσειράς (τίτλους, ετικέτες, κλπ).

## 3.8 Σημαντικοί κατάλογοι στο Android project:

### 3.8.1 AndroidManifest.xml

Αυτό είναι το αρχείο ορισμού Android. Περιέχει πληροφορίες σχετικά με την εφαρμογή του Android, όπως η ελάχιστη έκδοση του Android, δικαίωμα πρόσβασης σε Android δυνατότητες της συσκευής, όπως η δυνατότητα πρόσβασης στο διαδίκτυο, δυνατότητα να χρησιμοποιήσει την άδεια τηλέφωνο, κλπ.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.library.quizgame">

    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.CALL_PHONE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
    <uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_app_book"
        android:label="QuizGame"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".StartActivity"
            android:configChanges="locale|orientation">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".StartGameActivity" />
        <activity android:name=".CategoryActivity" />
        <activity android:name=".OptionsActivity"
            android:configChanges="locale|orientation"/>
        <activity android:name=".QuestionsActivity"/>
    </application>

</manifest>
```

Εικόνα 19: AndroidManifest.xml

Κάποια από τα κυριότερα δομικά στοιχεία του manifest.xml αναφέρονται αναλυτικά παρακάτω περιγράφοντας την τη λειτουργία του καθενός.

### 3.8.1.1 Δραστηριότητες (Activities)

Ένα κύριο δομικό στοιχείο μιας εφαρμογής. Δραστηριότητα είναι μια οθόνη διεπαφής χρήστη (GUI) και προβολής πληροφοριών. Κάθε εφαρμογή έχει τόσες Activities όσες και οι διαφορετικές οθόνες οι οποίες εμφανίζονται στον χρήστη. Όλες οι δραστηριότητες συνεργάζονται μεταξύ τους για να δώσουν στον χρήστη μια συνολική εμπειρία χρήσης της εφαρμογής.

```
package com.library.quizgame;

import ...

public class CategoryActivity extends AppCompatActivity {
    private Toolbar toolbar;
    private CategoriesAdapter adapter;
    private CategoriesReadTask task;
    private RecyclerView categoriesList;
    private GridLayoutManager gridLayoutManager;
    private List<SingleCategories> categories;
    private ProgressDialog progressDialog;
    private String locale;
    private String langText;
    private SharedPreferences sharedPreferences;
    private ConnectivityManager connectivityManager;
    private NetworkInfo networkInfo;
    private boolean network_connected;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        loadFromPrefs();
        StringGenerator.setLocale(langText, CategoryActivity.this);
        setContentView(R.layout.activity_category);
    }
}
```

Εικόνα 20: CategoryActivity.java

### 3.8.1.2 Υπηρεσίες (Services)

Πρόκειται για background Threads που τρέχουν για όση διάρκεια απαιτείται για να γίνει μια διεργασία, όπως είναι να φορτωθούν δεδομένα απο το διαδίκτυο ή να μετράμε τον χρόνο για κάποια διεργασία ή να παρέχουμε ζωντανές πληροφορίες όλο το 24ώρο. Δηλώνονται και αυτά στο Manifest και δεν μπορεί να τα δει ο χρήστης στο τελικό αποτέλεσμα.

### 3.8.1.3 Προθέσεις (Intents)

Οι δραστηριότητες επικοινωνούν και εναλλάσσουν την λειτουργία τους μέσω των Intents. Ουσιαστικά τα Intents εξασφαλίζουν την μετάβαση από τη μία δραστηριότητα

σε μια άλλη και επίσης χρησιμοποιούνται για ανταλλαγή δεδομένων. Η ανταλλαγή δεδομένων, μπορεί να γίνει είτε μεταξύ των Activities μιας εφαρμογής, είτε από τη μία εφαρμογή στην άλλη. Παραδείγματος χάρη μπορούμε μέσω ενός Intent να εκκινήσουμε έναν browser ώστε να μας ανοίξει απευθείας ένα url το οποίο έχουμε παρέχει εμείς μέσω ενός Intent.

#### 3.8.1.4 Πάροχος Περιεχομένου (Content Providers)

Οι πάροχοι περιεχομένου είναι ένα από τα κύρια δομικά στοιχεία του Android εφαρμογές, παρέχοντας περιεχόμενο σε εφαρμογές. Ενσωματώνουν τα δεδομένα και τα παρέχουν στις εφαρμογές μέσω της ενιαίας διεπαφής ContentResolver. Ένας πάροχος περιεχομένου απαιτείται μόνο αν χρειαστεί να μοιράζονται τα δεδομένα μεταξύ πολλαπλών εφαρμογών. Για παράδειγμα, τα δεδομένα επαφών χρησιμοποιείται από πολλαπλές εφαρμογές και πρέπει να αποθηκεύονται σε έναν πάροχο περιεχομένου.

#### 3.8.1.5 Οι κύριες μέθοδοι που πρέπει να υλοποιηθούν είναι:

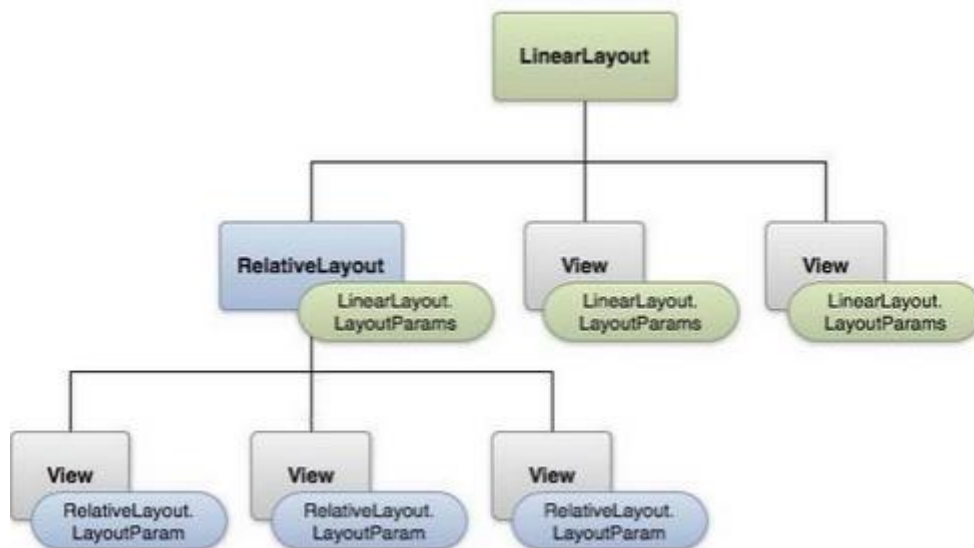
- **onCreate ()**, η οποία καλείται να προετοιμάσει τον πάροχο
- **query (Uri, String [], String, String [], String)**, η οποία επιστρέφει τα δεδομένα στον καλούντα
- **insert (Uri, ContentValues)**, η οποία εισάγει νέα δεδομένα στην υπηρεσία παροχής περιεχομένου
- **update (Uri, ContentValues, String, String [])** που αναβαθμίζουν τα υπάρχοντα δεδομένα στην υπηρεσία παροχής περιεχομένου
- **delete (Uri, String, String [])** που διαγράφει τα δεδομένα από τον πάροχο περιεχομένου
- **getType (URI)** η οποία επιστρέφει τον τύπο MIME των δεδομένων στον πάροχο περιεχομένου

### 3.8.2 MainLayout.xml:

Ως βασικό δομικό στοιχείο για τη διεπαφή χρήστη είναι ένα αντικείμενο View το οποίο δημιουργείται από την Προβολή κατηγορίας (View class) και καταλαμβάνει μια ορθογώνια περιοχή στην οθόνη και είναι υπεύθυνη για την κατάρτιση και το χειρισμό εκδήλωση. View είναι η βασική κλάση για τα widgets, τα οποία χρησιμοποιούνται για να δημιουργήσουν διαδραστικά στοιχεία UI, όπως κουμπιά, πεδία κειμένου, κλπ

Το ViewGroup είναι μια υποκατηγορία των Προβολή και παρέχει αόρατο δοχείο που κατέχουν άλλες Εμφανίσεις ή άλλες ViewGroup και να καθορίσει τις ιδιότητες διάταξή τους.

Στο τρίτο επίπεδο έχουμε διαφορετικές διατάξεις που είναι υποκατηγορίες ViewGroup τάξη και μια τυπική διάταξη ορίζει την οπτική δομή για ένα Android διεπαφή χρήστη και μπορούν να δημιουργηθούν είτε κατά το χρόνο εκτέλεσης, χρησιμοποιώντας Προβολή / ViewGroup αντικείμενα ή μπορείτε να δηλώσετε τη διάταξή σας, χρησιμοποιώντας απλό αρχείο XML στο layout.xml που βρίσκεται στο res / layout φάκελο του project.



Εικόνα 21 Linear Layout structure

Με λίγα λόγια, αυτό το αρχείο περιγράφει τη διάταξη της σελίδας. Αυτό σημαίνει ότι η τοποθέτηση του κάθε στοιχείου (όπως πλαίσια κειμένου, ετικέτες, κουμπιά επιλογής, ορίζονται από το χρήστη εξαρτήματα, κ.λπ.) στην οθόνη της εφαρμογής. Κατηγορία δραστηριότητας: Κάθε αίτηση που καταλαμβάνει ολόκληρη την οθόνη της συσκευής χρειάζεται τουλάχιστον μια τάξη που κληρονομεί από την κλάση Activity. Μια σημαντική μέθοδος ονομάζεται onCreate. Αυτή η μέθοδος ξεκινά την εφαρμογή και φορτώνει τη σελίδα διάταξης.

Ακολουθεί ένα απλό παράδειγμα (μέσα από το προτζεκ το οποίο θα αναλυθεί σε μια από τις περαιτέρω ενότητες ) .από το αρχείου XML που έχει LinearLayout.



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/welcome_fragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:clipChildren="false"
    android:background="@drawable/abc1"
    android:orientation="vertical"
    android:layout_weight="6"
    android:paddingTop="48dp">

    <FrameLayout
        android:id="@id/image_container"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="4"
        android:clipChildren="false">

        <ImageView
            android:id="@+id/img"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:scaleType="centerInside"
            android:src="@drawable/ic_local_library"
            android:visibility="invisible"
            android:paddingLeft="18dp"
            android:paddingRight="18dp"
            android:layout_gravity="center"/>

    </FrameLayout>
```

Εικόνα 22: MainLayout.xml

### 3.8.3 Gradle Scripts

Gradle είναι ένα ανοικτό σύστημα κατασκευής που κάνει το build process και βασίζεται στις έννοιες του Apache Ant και Apache Maven και εισάγει την Groovy γλώσσα που βασίζεται σε συγκεκριμένους τομείς (DSL) αντί της μορφής XML που χρησιμοποιείται από τον Apache Maven της δηλώνοντας τη διαμόρφωση του σχεδίου. Το Gradle χρησιμοποιεί ένα κατευθυνόμενο άκυκλο γράφημα ("DAG") για να καθορίσει τη σειρά με την οποία μπορεί να εκτελεστεί εργασίες που είναι up-to-date, έτσι ώστε κάθε projet που εξαρτάται από αυτά τα μέρη δεν θα πρέπει να είναι re-executed.

Δημιουργία του ακόλουθου σενάριού κατασκευής που ονομάζεται build.gradle.

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 23
    buildToolsVersion "23.0.2"

    defaultConfig {
        applicationId "com.library.quizgame"
        minSdkVersion 14
        targetSdkVersion 23
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.2.0'
    compile 'com.android.support:cardview-v7:23.2.0'
    compile 'com.android.support:gridlayout-v7:23.2.0'
    compile 'com.android.support:recyclerview-v7:23.2.0'
    compile 'org.jbundle.util.osgi.wrapped:org.jbundle.util.osgi.wrapped.org.apache.http.client:4.1.2'
}
```

Εικόνα 23: Build.gradle (Module app)

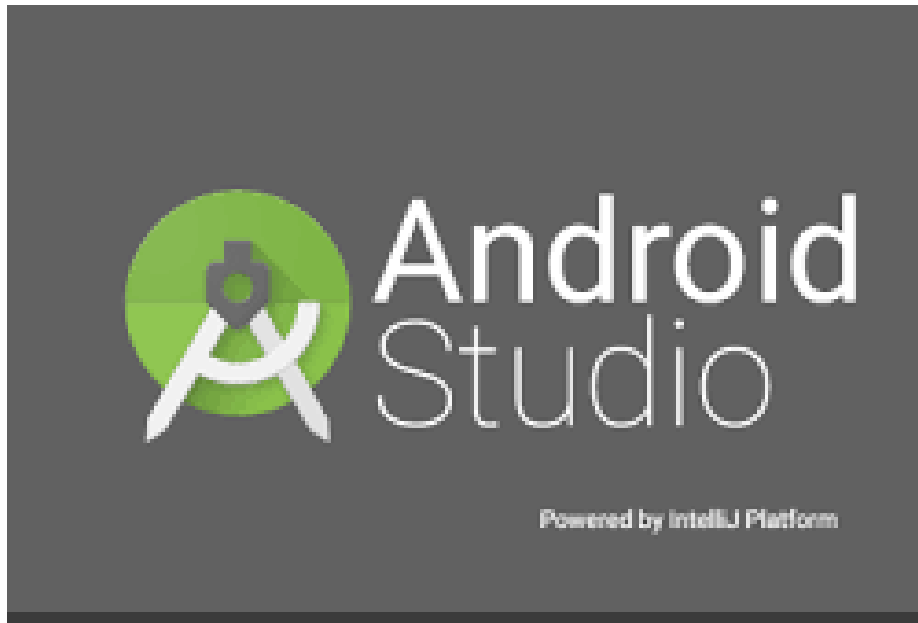


## 3.9 Διαδικασίες ανάπτυξης Android.

### 3.9.1 Δημιουργία Project στο Android Studio

Για να αναπτύξουμε την εφαρμογή θα πρέπει να έχουμε εγκαταστήσει ένα ολοκληρωμένο περιβάλλον διαχείρισης και δημιουργίας εφαρμογών Android τα λεγόμενα **IDE**. Τα περιβάλλοντα αυτά διαθέτουν πολλά εργαλεία αποσφαλμάτωσης, built-in compiler, auto-complete text editor και επίσης αυτόματη δημιουργία όλων των χρήσιμων αρχείων και φακέλων που καθιστούν την εφαρμογή έτοιμη να τρέξει και να χρησιμοποιηθεί από τον τελικό χρήστη. Ως βασικό εργαλείο ανάπτυξης του κώδικά για την υλοποίηση της εφαρμογής, είναι το Android studio.

Το Android Studio είναι το επίσημος ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για την ανάπτυξη της πλατφόρμας Android. Αυτό ανακοινώθηκε στις 16 Μαΐου 2013, στο συνέδριο της Google I / O. Το Android Studio είναι ελεύθερα διαθέσιμο υπό την άδεια χρήσης Apache 2.0. Android Studio ήταν σε πρώιμο στάδιο προεπισκόπησης πρόσβασης ξεκινώντας από την έκδοση 0.1 Μάιο του 2013, στη συνέχεια άρχισε beta στάδιο ξεκινώντας από την έκδοση 0.8 που κυκλοφόρησε τον Ιούνιο του 2014. Η πρώτη σταθερή build έκδοση, κυκλοφόρησε τον Δεκέμβριο του 2014, αρχίζοντας από την έκδοση 1.0. Βασισμένο σε λογισμικό IDEA JetBrains »IntelliJ, Android Studio έχει σχεδιαστεί ειδικά για την ανάπτυξη του Android είναι διαθέσιμο για download σε Windows, Mac OS X και Linux, για να αντικαταστήσει το Eclipse Android Development Tools (ADT) ως κύριο IDE της Google για την native Android.



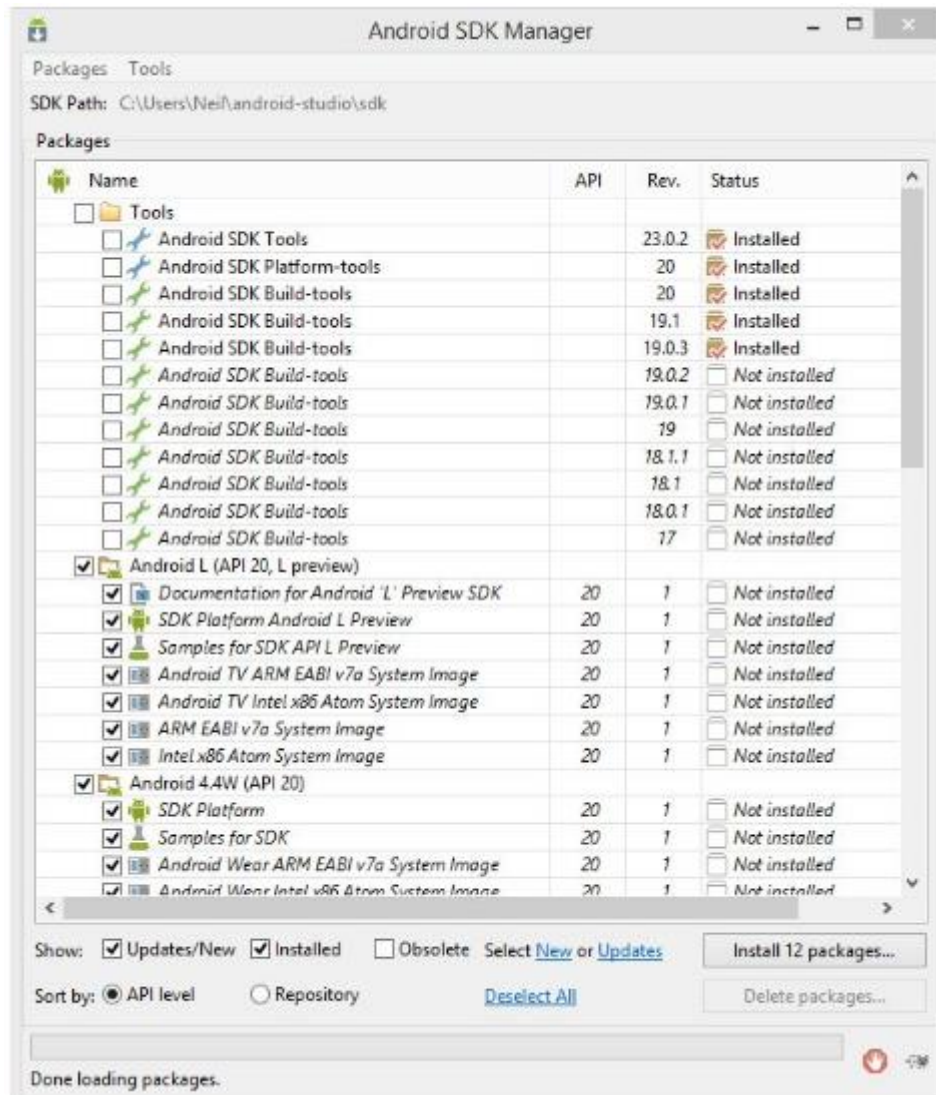
Εικόνα 24: Android studio Logo

## 3.10 Official development tools in android studio

### 3.10.1 Android SDK (System Development Kit)

το Android System Development Kit (SDK) περιλαμβάνει ένα ολοκληρωμένο σύνολο εργαλείων ανάπτυξης. Αυτά περιλαμβάνουν ένα πρόγραμμα εντοπισμού σφαλμάτων, βιβλιοθήκες, έναν handset emulator που βασίζεται σε QEMU, documentation, δείγματα κώδικα, και tutorials. Προς το παρόν οι υποστηρίζομενες πλατφόρμες ανάπτυξης περιλαμβάνουν υπολογιστές με λειτουργικό σύστημα Linux (οποιαδήποτε νεότερη desktop διανομή Linux), Mac OS X 10.5.8 ή νεότερη έκδοση και Windows XP ή νεότερη έκδοση. Από το Μάρτιο του 2015, η SDK δεν είναι διαθέσιμο για το Android, αλλά η ανάπτυξη λογισμικού είναι δυνατή με τη χρήση εξειδικευμένων εφαρμογών Android.

Το πως είναι διαθέσιμα τα πακέτα ανάλογα με τα API's του μπορούμε να δούμε στην παρακάτω εικόνα:



Εικόνα 25: Android SDK

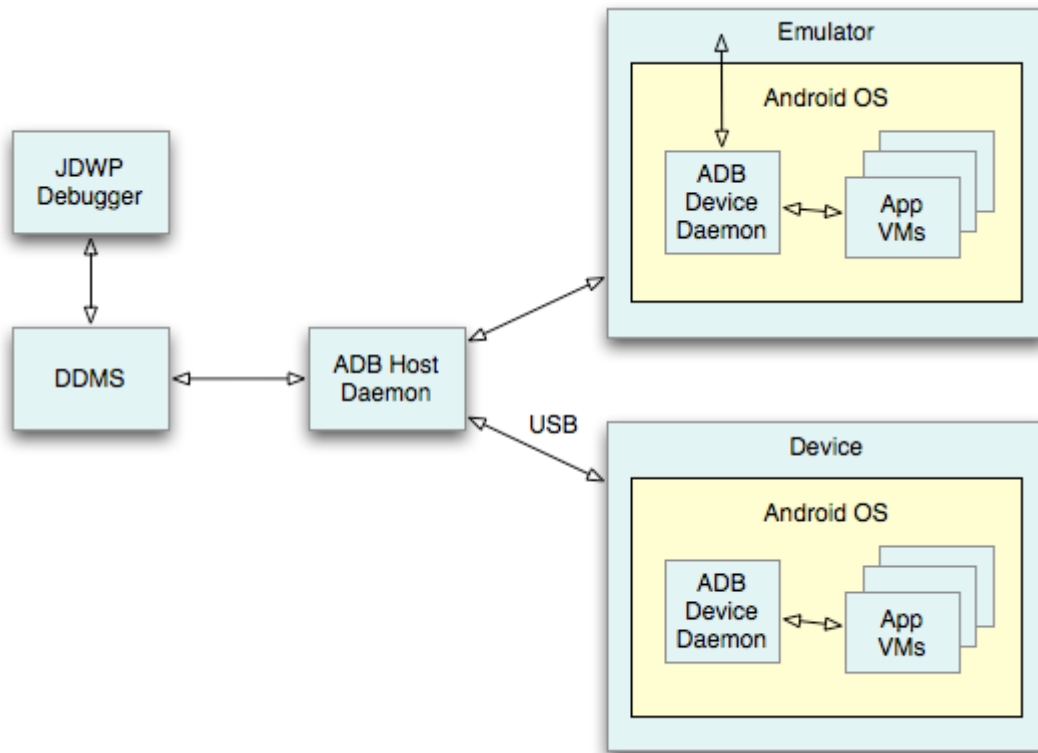
### 3.10.2 Διαδικασία Αποσφαλμάτωσης (Debugging)

Είναι η διαδικασία για την εξεύρεση και την επίλυση των ελαττωμάτων που εμποδίζουν τη σωστή λειτουργία του λογισμικού ηλεκτρονικών υπολογιστών ή ένα σύστημα. Debugging τείνει να είναι πιο δύσκολο όταν τα διάφορα υποσυστήματα στενά συνδεδεμένες, όπως αλλαγές σε ένα μπορεί να προκαλέσει σφάλματα να βγει στην άλλη. Ο προγραμματιστής επιλέγει πως θα υλοποιήσει την εφαρμογή αλλά το λειτουργικό σύστημα θα καθορίσει ποιους πόρους θα χρησιμοποιήσει.

Σε αυτήν την διαδικασία χρησιμοποιείται το Logcat το οποίο είναι ένας καταγραφέας για το οτιδήποτε συμβαίνει στην εφαρμογή σου και στους πόρους του συστήματος που χρησιμοποιεί. Είναι υπεύθυνο για να ενημερώσει τον προγραμματιστή για ότι προκύψει(μνήμη που χρησιμοποιεί η εφαρμογή, συντακτικά και λογικά λάθη στον κώδικα, σφάλματα στο μητρώο, παραβίαση αδειών, παραβίαση δεδομένων της εφαρμογής και άλλα).

Επίσης ένα άλλο σημαντικό εργαλείο που χρησιμοποιούν οι προγραμματιστές είναι το DDMS το οποίο παρέχει την δυνατότητα ελέγχου της εφαρμογής σε εικονικές συσκευές που δεν έχουν πρόσβαση σε ζωντανά δεδομένα. Για παράδειγμα, φτιάχνουμε μια εφαρμογή ανταλλαγής μηνυμάτων κειμένου και στατικών εικόνων πριν τη δοκιμάσουμε σε κανονική συσκευή και μας επιφέρει πρόσθετες χρεώσεις μόνο για τις δοκιμές μπορούμε να στέλνουμε fake notifications ή και μηνύματα στην εικονική συσκευή με στόχο την βελτίωση. Το ίδιο γίνεται και με την γεωγραφική θέση, κλήση αριθμού, προβολή website, προβολή στατιστικών heap memory allocation, thread memory size, network statistics και άλλα.

Η παρακάτω εικόνα πώς τα διάφορα εργαλεία εντοπισμού σφαλμάτων εργάζονται μαζί σε ένα τυπικό εντοπισμού σφαλμάτων:



Εικόνα 26: Debugging Environment

### 3.10.3 Android Debug Bridge

Το Android Debug Bridge (ADB) είναι ένα σύνολο εργαλείων που περιλαμβάνονται στο πακέτο του Android SDK. Αποτελείται από δύο προγράμματα client και server-side που επικοινωνούν μεταξύ τους. Η ADB είναι συνήθως προσβάσιμες μέσω της διεπαφής γραμμής εντολών, αν και υπάρχουν πολλές γραφικές διεπαφές χρήστη να ελέγχει ADB

### 3.10.4 Δημοσίευση Εφαρμογής στο κοινό (Publishing)

Η διαδικασία δημοσίευσης τυπικά εκτελείται αφού ολοκληρώσουμε τον έλεγχο της εφαρμογής μας σε ένα περιβάλλον εντοπισμού σφαλμάτων. Επίσης, ως βέλτιστη πρακτική, η αίτησή μας θα πρέπει να πληροί όλα τα κριτήρια απελευθέρωσης για τη λειτουργικότητα, την απόδοση και τη σταθερότητα πριν ξεκινήσουμε τη διαδικασία έκδοσης.

Για να δημοσιεύσουμε μια εφαρμογή χρειαζόμαστε το .apk αρχείο που θα μας δώσει ο compiler εφόσον περάσει επιτυχώς την διαδικασία της αποσφαλμάτωσης. Αν

θέλουμε να δημοσιεύσουμε την εφαρμογή μας στο Google Play Store θα πρέπει να κάνουμε εγγραφή σαν προγραμματιστές και στην συνέχεια δίνοντας μια περιγραφή και μερικά screenshots η εφαρμογή μας δημοσιεύεται και είναι διαθέσιμη για κατέβασμα απο τον χρήστη.

Την στιγμή που γράφουμε την παρούσα πτυχιακή εργασία δεν είναι διαθέσιμη η εγγραφή εμπόρου στην Ελλάδα δηλαδή οι προγραμματιστές εφόσον καταβάλλουν τα 25\$ που είναι το registration fee της Google για developer registration θα μπορούν να δημοσιεύουν εφαρμογές αλλά θα διατίθενται δωρεάν στο ευρύ κοινό. Αυτό σημαίνει ότι δεν θα μπορούν να έχουν κέρδος απο τα downloads της εφαρμογής παρά μόνο από τα in-app purchases (για παράδειγμα η αναβάθμιση μια εφαρμογής απο trial σε pro έκδοση μέσω paypal ή google wallet) ή το admob(διαδικασία πληρωμής από διαφημίσεις της Google μέσα στην εφαρμογή).

Η παρακάτω εικόνα δείχνει πώς η διαδικασία έκδοσης ταιριάζει στη συνολική διαδικασία ανάπτυξης εφαρμογών Android.

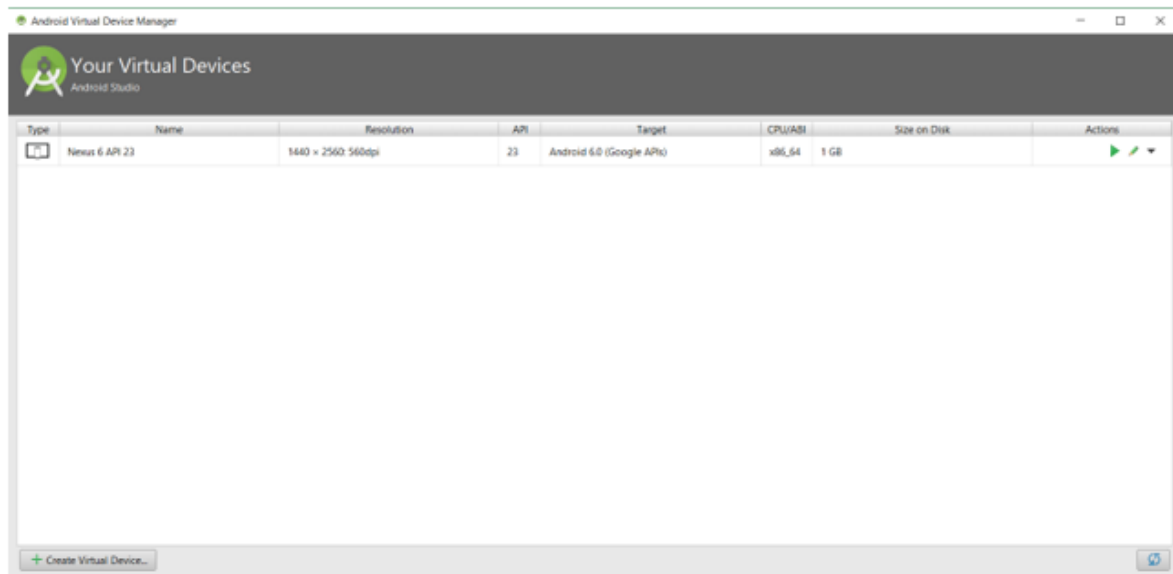


Εικόνα 27: Publishing Overview

### 3.10.5 Android Virtual Device Manager(AVD)

Επίσης θα πρέπει να κατεβάσουμε όλα τα απαραίτητα αρχεία για την δημιουργία εικονικής συσκευής με σκοπό να τεστάρουμε την εφαρμογής μας. Δουλεία η οποία αναλαμβάνει το AVD Android Virtual Device Manager με το κατάλληλο SDK και έκδοση του λειτουργικού συστήματος και με τον κατάλληλο επεξεργαστή. Βεβαίως θα πρέπει να έχουμε εγκατεστημένο το API που θέλουμε να τρέξει και να διαλέξουμε τις ρυθμίσεις οθόνης, μέγεθος μνήμης, χρησιμοποίηση της κάρτας γραφικών του HOST για καλύτερες επιδόσεις ιδιαίτερα σε εφαρμογές με μέγιστη απόδοση γραφικών ή σε παιχνίδια. Παρακάτω θα δούμε πως είναι το AVD:

Διαλέγουμε την συσκευή από την λίστα:



Εικόνα 28: AVD Manager Device Selection

## 4 Δημιουργία της εφαρμογής – τελικό αποτέλεσμα

### 4.1 Ανάλυση της Δημιουργίας της εφαρμογής

#### 4.1.1 Η βάση δεδομένων My SQL

Η **MySQL** είναι μια γλώσσα προγραμματισμού ειδικού σκοπού σχεδιασμένη για τη διαχείριση των δεδομένων που τηρούνται σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS), ή για επεξεργασία ροή σε ένα σύστημα relational data stream management system (RDBMS). Αρχικά με βάση σχεσιακή άλγεβρα και πλειάδα σχεσιακό λογισμό, SQL αποτελείται από μια γλώσσα ορισμού δεδομένων, τη γλώσσα χειρισμού δεδομένων, καθώς και τη γλώσσα ελέγχου δεδομένων. Το πεδίο εφαρμογής της SQL περιλαμβάνει εισαγωγή δεδομένων, το ερώτημα, ενημέρωση και διαγραφή, δημιουργία σχήματος και τροποποίηση, και τον έλεγχο πρόσβασης δεδομένων. Αν και SQL είναι συχνά περιγράφεται ως, και σε μεγάλο βαθμό είναι μια δηλωτική γλώσσα (4GL), περιλαμβάνει επίσης διαδικαστικά στοιχεία.

Η γλώσσα SQL υποδιαιρείται σε πολλά στοιχεία της γλώσσας, μεταξύ των οποίων:

**Clauses**, οι οποίες είναι συστατικά στοιχεία των δηλώσεων και ερωτήματα. (Σε ορισμένες περιπτώσεις, αυτές είναι προαιρετικές.)

**Expressions**, οι οποίες μπορούν να παράγουν είτε βαθμωτές τιμές ή πίνακες που αποτελείται από τις στήλες και τις σειρές των δεδομένων



**Pradicates**, τα οποία καθορίζουν τους όρους που μπορούν να αξιολογηθούν σε SQL τριών αποτιμώνται λογική (3VL) τιμές (true / false / άγνωστη) ή Boolean αλήθεια και χρησιμοποιούνται για να περιορίσουν τις επιπτώσεις των δηλώσεων και ερωτήματα, ή να αλλάξει τη ροή του προγράμματος.

**Queries**, που να ανακτήσετε τα δεδομένα με βάση συγκεκριμένα κριτήρια. Αυτό είναι ένα σημαντικό στοιχείο της SQL.

**Statements**, οι οποίες έχουν μια επίμονη επίδραση στα σχήματα και τα δεδομένα, ή να μπορεί να ελέγχει τις συναλλαγές, τη ροή του προγράμματος, συνδέσεις, συνεδρίες, ή διαγνωστικά.

**SQL Statements**, περιλαμβάνουν επίσης το ερωτηματικό ("?") Δήλωση τερματισμού. Αν και δεν απαιτείται σε κάθε πλατφόρμα, ορίζεται ως ένα τυπικό μέρος της SQL γραμματικής. .

#### 4.1.2 Εφαρμογή και ανάπτυξη της MySQL.

Για τις ανάγκες του Project της πτυχιακής μας, δημιουργήσαμε τρεις πίνακες. Πίνακας ερωτήσεις (questions). Απαντήσεις (answers) και κατηγορίες.(categories). Παρακάτω βλέπουμε ένα παράδειγμα απεικόνισης του πίνακα των απαντήσεων που χρησιμοποιούμε στο Project μας..

Κώδικας για την δημιουργία της βάσης:

```
CREATE TABLE `answers` (  
  `quest_id` int(11) NOT NULL,  
  `answer` text NOT NULL,  
  `incorrect` tinyint(3) UNSIGNED ZEROFILL NOT NULL,  
  `anser_id` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Έχοντας ως αποτέλεσμα την εμφάνιση του παρακάτω πίνακα.

quest_id	answer	isincorrect	anser_id
1	Μας ενημερώνει που βρίσκετε το βιβλίο στο ράφι	001	1
1	Μας ενημερώνει ότι υπάρχει το βιβλίο	000	2
1	Μας ενημερώνει για το θέμα του βιβλίου	000	3
2	είναι ένας μοναδικός διεθνής αριθμός για μονογραφί...	001	4
2	είναι ένας μοναδικός διεθνής αριθμός για μουσικές ...	000	5
2	είναι ένας μοναδικός διεθνής αριθμός για περιοδικέ...	000	6
3	Στην Agricola	000	7
3	Στην MathScienceNet online	000	8
3	Στην Medline	001	9
4	Διατριβή	001	10
4	Ευρετήριο	000	11
4	Αλμανάκ	000	12
5	Βιβλιογραφία	001	13
5	Απομνημονεύματα	000	14
5	Αλμανάκ	000	15
6	Άρθρο περιοδικού	001	16
6	Εγκυκλοπαίδεια	000	17
6	Ευρετήριο	000	18
7	Στον κατάλογο της βιβλιοθήκης	000	37
7	Στον αλφαβητικό κατάλογο του healink	000	38
7	Στις βάσεις δεδομένων που υπάρχουν στην ιστοσελίδα...	001	39
8	Ταξινομικό αριθμό	000	40
8	Ταξινομικό αριθμό και διαθεσιμότητα	000	41
8	Ταξινομικό αριθμό διαθεσιμότητα και συλλογή	001	42
9	Είναι πολύ γενική και θα μας επιφέρει πολλά άσχετα...	001	43

Εικόνα 29:Table of answers

### 4.1.3 PHP Script for exporting Data in JSON Array

Η εκτέλεση των 2 παραπάνω κομματιών κώδικα μας δίνει το αποτέλεσμα που είδαμε παραπάνω. Βεβαίως για να επικοινωνήσει η εφαρμογή μας με τα αποτελέσματα από την βάση δεδομένων θα πρέπει να εξάγουμε τα δεδομένα μας σε μορφή JSON πίνακα. Αυτό το επιτυγχάνουμε με τον παρακάτω κώδικα:

```

<?php
error_reporting(E_ALL ^ E_NOTICE);
ini_set("default_charset", "UTF-8");
header('Content-type: text/html; charset=UTF-8');
try {
    $handler = new PDO('mysql:host=localhost;dbname=ptuxiakli', 'root', 'password');
    $handler->setAttribute(PDO::MYSQL_ATTR_INIT_COMMAND, "SET NAMES utf8");
    $handler->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $handler->exec("SET CHARACTER SET 'utf8'");
} catch (Exception $e) {
    echo $e->getMessage();
    die();
}

if (isset($_POST['category_id'])) {
    $cat_id = $_POST['category_id'];
    $query = $handler->query("SELECT DISTINCT c.cat_description, c.cat_name, c.cat_id, q.question, q.q_id
    FROM 'categories' c
    INNER JOIN 'questions' q ON c.cat_id = q.cat_id
    WHERE c.cat_id = '$cat_id'");
    $records = array(); $records = $query->fetchAll(PDO::FETCH_ASSOC); $first = array(); $second = array(); $third = array(); $query->closeCursor();
    foreach ($records as $k => $v) {
        $first[] = array("category_name" => $v['cat_name'], "category_id" => $v['cat_id'], "category_description" => $v['cat_description'],
        "question_name" => $v['question'],
        "question_id" => $v['q_id'], "question_answers" => array());
        $second[] = $v['question'];
    }
    foreach ($second as $key => $value) {
        $ques = $value;
        $qu = $handler->query("SELECT a.quest_id, a.answer, a.iscorrect, a.anser_id
        FROM 'answers' a
        INNER JOIN 'questions' q ON a.quest_id = q.q_id
        WHERE q.question = '$ques'");
        $third = $qu->fetchAll(PDO::FETCH_ASSOC);
        foreach ($third as $tk => $tv) {
            $third[$tk]['answer' . ($tk+1)] = $tv['answer'];
            $third[$tk]['incorrect' . ($tk+1)] = $tv['incorrect'];
        }
        foreach ($first as $k => $v) {
            $first[$key]['question_answers'] = $third;
        }
    }
    $j['quiz'] = $first;
    echo json_encode($j);
} else {echo "You have to make a POST Request for id";}?>

```

Εικόνα 30:Κώδικας PHP για export σε JSON πίνακα

Χρησιμοποιώντας PDO (php data objects) αντικειμενοστραφής php για να συνδεθώ στην βάση δεδομένων και να πιάσουμε τα Exceptions. Έγινε χρήση της μεθόδου query() για να κάνουμε το «ερώτημα» στην βάση δεδομένων και να συλλέξουμε όλα τα δεδομένα από τον συγκεκριμένο πίνακα φέρνοντας τα σαν Associative array. Τα κάνουμε encode σε json array ονομαζόμενο Categories και μας προκύπτει το παρακάτω αποτέλεσμα στον browser.

#### 4.1.4 Ανάλυση της PHP

Ως γνωστόν, η **PHP** είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή (Server) του Παγκόσμιου Ιστού (όπως για παράδειγμα ο Apache, ), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML. Ένα αρχείο με κώδικα PHP θα πρέπει να έχει την κατάλληλη επέκταση (π.χ. \*.php, \*.php4, \*.phpml κ.ά.).

Η ενσωμάτωση κώδικα σε ένα αρχείο επέκτασης .html δεν θα λειτουργήσει και θα εμφανίσει στον browser τον κώδικα χωρίς καμία επεξεργασία, εκτός αν έχει γίνει η

κατάλληλη ρύθμιση στα MIME types του server. Επίσης ακόμη κι όταν ένα αρχείο έχει την επέκταση .php, θα πρέπει ο Server να είναι ρυθμισμένος για να επεξεργάζεται και να μεταγλωττίζει τον κώδικα PHP σε HTML που καταλαβαίνει το πρόγραμμα πελάτη.

Ο διακομιστής Apache, που χρησιμοποιείται σήμερα ευρέως σε συστήματα με τα λειτουργικά συστήματα GNU/Linux, Microsoft Windows, Mac OS X υποστηρίζει εξ ορισμού την εκτέλεση κώδικα PHP, είτε με την χρήση ενός πρόσθετου (mod\_php) ή με την αποστολή του κώδικα προς εκτέλεση σε εξωτερική διεργασία CGI ή FCGI ή με την έλευση της php5.4 υποστηρίζονται η εκτέλεση σε πολυάσχολους ιστοχώρους, FastCGI Process Manager (FPM).

Έτσι και στο δικό μας πρότζεκτ, προσαρμόσαμε και αναπτύξαμε όλες τις απαραίτητες τεχνικές για την πραγματοποίηση του τελικού αποτελέσματος.

Θα χρειαστεί για αρχή να δημιουργηθεί η σύνδεση με την βάση δεδομένων που έχουμε δημιουργήσει στη MySQL, Χρησιμοποιώντας το σωστό encoding για τη σωστή ανάγνωση του κειμένου. Και όπως γίνεται σε κάθε περίπτωση του προγράμματος πάνουμε τα πιθανά Exceptions, όπως βλέπουμε στον παρακάτω κώδικα.

```
try {  
    $handler = new PDO('mysql:host=localhost;dbname=ptuxiaki', 'root', '');  
    $handler->setAttribute(PDO::MYSQL_ATTR_INIT_COMMAND, "SET NAMES utf8");  
    $handler->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
    $handler->exec("SET CHARACTER SET 'utf8'");  
}  
catch (Exception $e) {  
    echo $e->getMessage();  
    die();  
}
```

Εικόνα 31: create connetion the database to the servre

**Associative array** είναι ένας συσχετισμένος πίνακας, που αποθηκεύει τις τιμές των κλειδιών χωρίς να είναι απαραίτητα σε γραμμική σειρά. Μπορούμε δηλαδή να προσθέσουμε να αφαιρέσουμε να τροποποιήσουμε ή και να αναζητήσουμε μια τιμή που έχει δηλωθεί ως κειδί στον πίνακα μας. Αναλόγως λοιπόν έχουμε χτίσει τον κώδικά μας.

```
$records = $query->fetchAll(PDO::FETCH_ASSOC);  
$first = array();  
$second = array();  
$third = array();  
$query->closeCursor();
```

Εικόνα 32: create associative arrays

Χρησιμοποιούμε δύο συσχετισμένους πίνακες. Ένα πίνακα για τις κατηγορίες και τις ερωτήσεις και ένα μόνο για τις ερωτήσεις.

```

foreach($records as $k => $v){
    $first[] = array("category_name" => $v['cat_name'],
                    "category_id" => $v['cat_id'],
                    "category_description" => $v['cat_description'],
                    "question_name" => $v['question'],
                    "question_id" => $v['q_id'],
                    "question_answers" => array());
    $second[] = $v['question'];
}

```

Εικόνα 33: create tow associate table of Categories and questions

Εδώ χρησιμοποιούμε τον πίνακα με τις ερωτήσεις για να κάνουμε query για κάθε ερώτηση για να πάρουμε τις απαντήσεις της κάθε ερώτησης.

```

foreach ($second as $key => $value) {
    $ques = $value;
    $qu = $handler->query("SELECT a.quest_id, a.answer, a.iscorrect, a.anser_id
                          FROM `answers` a INNER JOIN `questions` q ON a.quest_id = q.q_id
                          WHERE q.question = '$ques' ");
    $third = $qu->fetchAll(PDO::FETCH_ASSOC);
}

```

Εικόνα 34: Query for question selection for gettin answers of each question

Πρέπει να χρησιμοποιήσουμε και τον πίνακα που θα αποθηκεύονται οι απαντήσεις. Για κάθε ερώτηση λοιπόν, δημιουργούμε ένα πίνακα question\_answers που περιέχει τις απαντήσεις καθώς επίσης και το πιά ερώτηση είναι η σωστή.

```

foreach($third as $tk => $tv){
    $third[$tk]['answer' . ($tk+1)] = $tv['answer'];
    $third[$tk]['incorrect' . ($tk+1)] = $tv['incorrect'];
}
foreach ($first as $k => $v) {
    $first[$key]['question_answers'] = $third;
}

```

Εικόνα 35: create Get\_answers associate array

### 4.1.5 JSON

JSON (JavaScript Object Notation) είναι μια μεταγλώσσα η οποία μας παρέχει τον τρόπο για να αποθηκεύσουμε τις πληροφορίες με έναν οργανωμένο, εύκολο στην πρόσβαση τρόπο. Με λίγα λόγια, μας δίνει μια ανθρώπινη αναγνώσιμη από τη συλλογή των δεδομένων που μπορούμε να έχουμε πρόσβαση σε ένα πραγματικά λογικό τρόπο.

```
- question_answers: [  
  - {  
    quest_id: "9",  
    answer: "Είναι πολύ γενική και θα μας επιφέρει πολλά άσχετα αποτελέσματα σε σχέση με το θέμα μας",  
    iscorrect: "001",  
    anser_id: "43",  
    answer1: "Είναι πολύ γενική και θα μας επιφέρει πολλά άσχετα αποτελέσματα σε σχέση με το θέμα μας",  
    iscorrect1: "001"  }  
]
```

Εικόνα 36: Json Object

Από την παραπάνω εικόνα, βλέπουμε τη δημιουργία ενός αντικείμενου που έχουμε πρόσβαση χρησιμοποιώντας τη μεταβλητή Json. Περικλείοντας τιμή της μεταβλητής σε άγκιστρα, είμαστε δείχνει ότι η τιμή είναι ένα αντικείμενο. Μέσα από το αντικείμενο, μπορούμε να δηλώσουμε οποιοδήποτε αριθμό των ιδιοτήτων χρησιμοποιώντας ένα "name": "value" pair, χωρισμένα με κόμμα.

Για να αποκτήσουμε πρόσβαση στις πληροφορίες που αποθηκεύονται σε json, μπορούμε απλά να αναφερόμαστε στο όνομα του αντικειμένου που χρειαζόμαστε.

#### 4.1.6 Εξάγοντας τα δεδομένα σε JSON Array στον browser

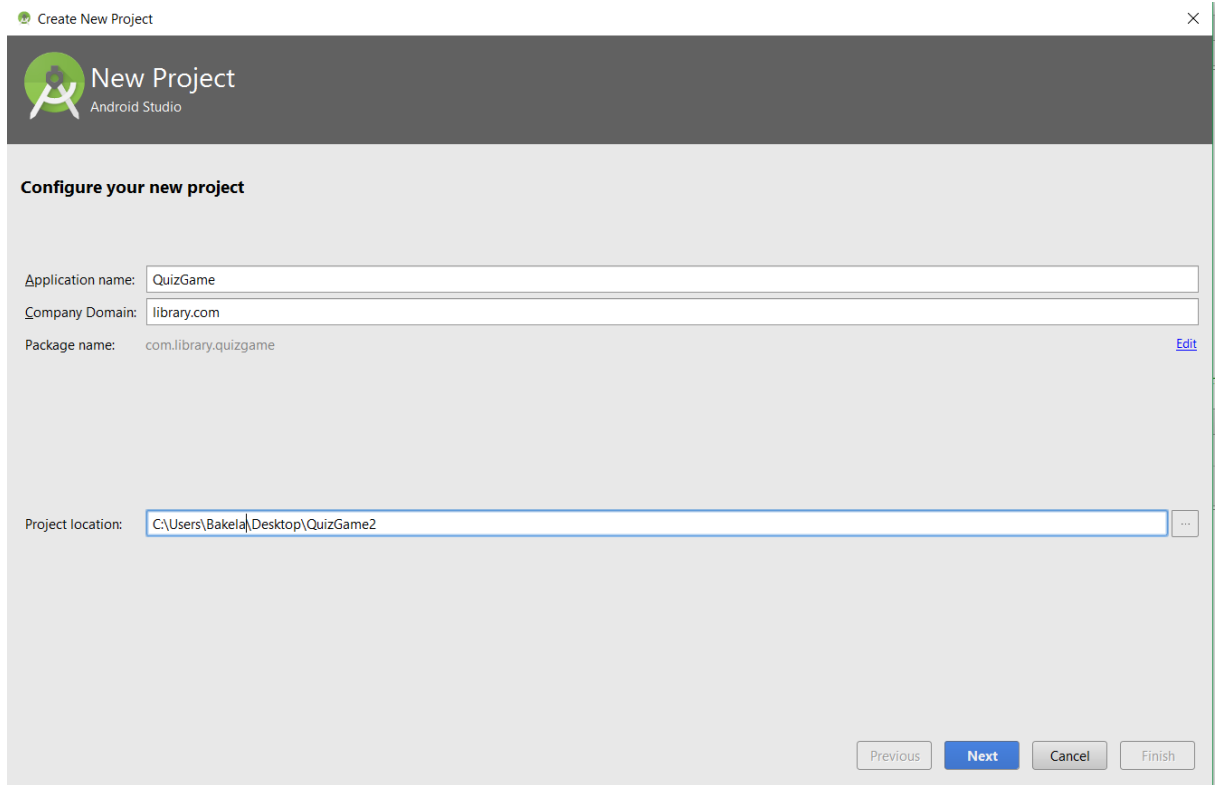
```
{
  - categories: [
    - {
      cat_id: "1",
      cat_name: "Όργανωση Πληροφοριων",
      cat_description: ""
    },
    - {
      cat_id: "2",
      cat_name: "Ηλεκτρονικος Καταλογος",
      cat_description: ""
    },
    - {
      cat_id: "3",
      cat_name: "Περιοδικα",
      cat_description: ""
    },
    - {
      cat_id: "4",
      cat_name: "Βασεις Δεδομενων",
      cat_description: ""
    },
    - {
      cat_id: "5",
      cat_name: "Διαδικοτο",
      cat_description: ""
    },
    - {
      cat_id: "6",
      cat_name: "Βιβλιογραφικες Παραπομετες",
      cat_description: ""
    }
  ]
}
```

Εικόνα 37:Exporting Data in JSON Array into the browser



## 4.2 Δημιουργία Project στο Android Studio

Αφού έχουμε κάνει όλα αυτά που περιγράψαμε πιο πάνω είμαστε έτοιμοι να δημιουργήσουμε το καινούργιο μας project. Πάμε λοιπόν Create New Project και μας εμφανίζεται η παρακάτω οθόνη:



Εικόνα 38: New project with Android studio

Στην συνέχεια επιλέγουμε τα SDK's και για τι πλατφόρμα θέλουμε να δημιουργήσουμε την εφαρμογή και αφού επιλέξουμε για κινητό ή τάμπλετ τότε θα μας βγάλει στο πρώτο μας Activity το οποίο λέγεται StartActivity.java.

### 4.2.1 Δήλωση αδειών χρήστη στο AndroidManifest

Επειδή φορτώνουμε δεδομένα από το Ίντερνετ θα πρέπει να έχουμε τα εξής permissions στο Manifest.xml.

Η υλοποίηση της Activity έχει ολοκληρωθεί και πλέον έχουμε έναν πλήρη μηχανισμό λήψης, αποθήκευσης και προβολής των ανακοινώσεων του τμήματος. Για να λειτουργήσουν όλα τα αντικείμενα που δημιουργήσαμε όμως, χρειάζονται κάποιες άδειες χρήσης του συστήματος, οι οποίες θα μας δώσουν την δυνατότητα να:

- Πρόσβαση στο Ίντερνετ
- Ελέγχουμε της κατάστασης του δικτύου
- Αποθήκευση των αρχείων στην κάρτα SD της συσκευής

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.library.quizgame">

    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.CALL_PHONE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
    <uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" />

</manifest>
```

Εικόνα 39: Permissions

Κατά την εγκατάσταση της εφαρμογής, ο χρήστης ενημερώνεται γι' αυτές τις τρεις άδειες που θα ζητηθούν από το σύστημα.

## 4.2.2 Έλεγχος της συσκευής για πρόσβαση στο Ίντερνετ

Λόγω του ότι η εφαρμογή μας δυναμική, που σημαίνει ότι πρέπει να υπάρχει real time connection με τον server μας ο οποίος μας συνδέει στη βάση μας για την άντληση των δεδομένων. Για να πραγματοποιήσουμε τον έλεγχο της σύνδεσης της συσκευής μας, χρησιμοποιούμε τον παρακάτω κώδικα.

```
private void checkNetwork() {
    connectivityManager = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
    networkInfo = connectivityManager.getActiveNetworkInfo();
    network_connected = networkInfo != null && networkInfo.isAvailable() && networkInfo.isConnectedOrConnecting();
    if (!network_connected) {
        DialogMessageDisplay.displayWifiSettingsDialog(CategoryActivity.this, CategoryActivity.this, "No Internet Connection",
    }else {
        //φιτιάχνουμε το view μας. Δηλαδή τι θα βλέπει ο χρήστης
        setupCategoryView();
    }
}
```

Εικόνα 40: Check for network connection

Επίσης, ελέγχουμε αν ο τύπος του ενεργού δικτύου είναι μέσω Wifi ώστε να κάνουμε την φόρτωση των δεδομένων κανονικά, ενώ αν γίνεται μέσω δεδομένων κινητής τηλεφωνίας θα πρέπει να προειδοποιήσουμε τον χρήστη για τη χρέωση που μπορεί να επιφέρει ανάλογα με το πακέτο που χρησιμοποιεί. Αυτό το επιτυγχάνουμε με το παρακάτω κώδικα:

```
@Override
public void onConfigurationChanged(Configuration newConfig) {
    super.onConfigurationChanged(newConfig);
    if (newConfig.orientation == Configuration.ORIENTATION_LANDSCAPE) {
        if (pDialog.isShowing()) {
            pDialog.show();
        } else {
            pDialog.dismiss();
        }
        if (onAlertMobileData() != null)
            if (onAlertMobileData().isShowing()) {
                onAlertMobileData().show();
            } else {
                onAlertMobileData().dismiss();
            }
        else {
            onAlertMobileData().dismiss();
        }
        if (onDetectNetworkState().isShowing()
            && onDetectNetworkState() != null) {
            onDetectNetworkState().show();
        } else {
            onDetectNetworkState().dismiss();
        }
    }
    if (newConfig.orientation == Configuration.ORIENTATION_PORTRAIT) {

        if (pDialog.isShowing()) {
            pDialog.show();
        } else {
            pDialog.dismiss();
        }

        if (onAlertMobileData().isShowing()) {
            onAlertMobileData().show();
        } else {
            onAlertMobileData().dismiss();
        }
        if (onDetectNetworkState().isShowing()) {
            onDetectNetworkState().show();
        } else {
            onDetectNetworkState().dismiss();
        }
    }
}
```

Εικόνα 41: Alert on internet connection

Εφόσον έχουμε ελέγξει τα πάντα, είμαστε έτοιμοι να φτιάξουμε την μέθοδο που θα μας επιτρέψει να φορτώσουμε τα δεδομένα σε μορφή Json είτε xml από το ίντερνετ. Αυτό θα γίνει με την χρήση AsyncTask η οποία είναι μια μέθοδος ασύγχρονης διεργασίας για φόρτωση δεδομένων:

```

public class JsonReadTask extends AsyncTask<String, Void, String> {
    public JsonReadTask() { super(); }

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        progressDialog = new ProgressDialog(ListLoaderActivity.this);
        progressDialog.setTitle("Please wait");
        progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
        progressDialog.setMessage("Loading Stocks...");
        progressDialog.setIndeterminate(true);
        progressDialog.setCancelable(false);
        progressDialog.setInverseBackgroundForced(true);
        progressDialog.show();
    }

    @Override
    protected String doInBackground(String... params) {
        HttpClient httpClient = new DefaultHttpClient();
        HttpPost httpPost = new HttpPost(params[0]);
        try {
            HttpResponse response = httpClient.execute(httpPost);
            jsonResult = inputStreamToString(
                response.getEntity().getContent()).toString();
        } catch (Exception e) {
            Intent intent1 = new Intent(ListLoaderActivity.this,
                RefreshActivity.class);
            startActivity(intent1);
            ListLoaderActivity.this.finish();
        }
        return null;
    }

    private StringBuilder inputStreamToString(InputStream is) {...}

    @Override
    protected void onPostExecute(String result) {
        ListDrawer();
        progressDialog.dismiss();
    }
} // end async task

public void accessWebService() {...}

public void ListDrawer() {...}

```

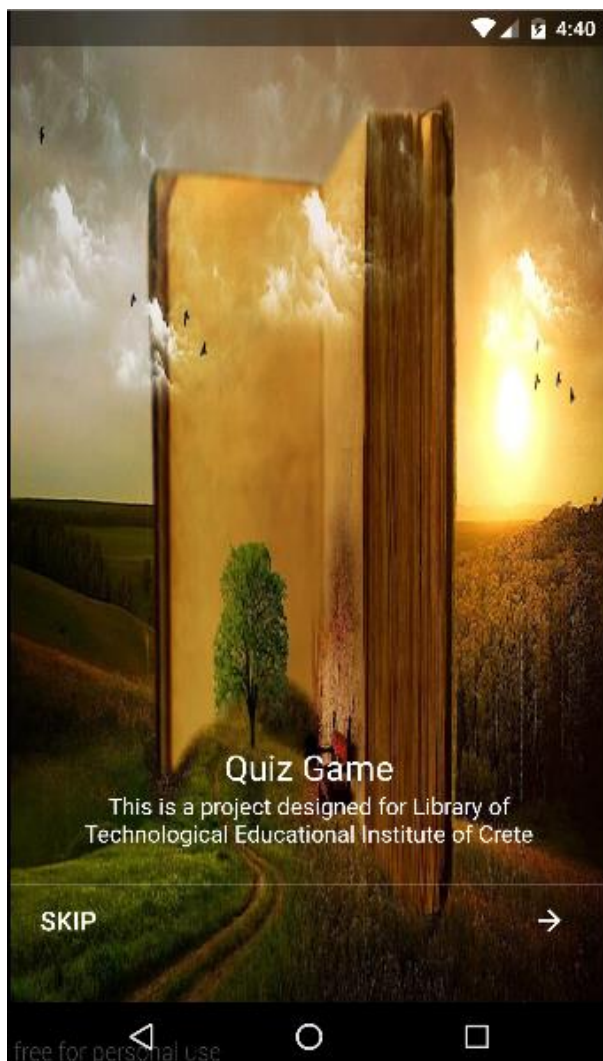
Εικόνα 42: Method onPostExecute

Η μέθοδος αυτή περιέχει 3 μεθόδους *preExecute*, *doInBackground*, *postExecute*. Πρέπει να προειδοποιήσουμε τον χρήστη για το χρονικό διάστημα που η εφαρμογή φορτώνει δεδομένα. Αυτό γίνεται με ένα *ProgressDialog* παράθυρο διαλόγου που ενημερώνει τον χρήστη για την πρόοδο για να αποφύγουμε το πάγωμα της οθόνης και τις κακές κριτικές από τους χρήστες αν έχουμε μεγάλο όγκο δεδομένων.

## 4.3 Το Layout του Library quiz

### 4.3.1 Welcome screen.

Στην οθόνη αυτή βλέπουμε κάποιες πληροφορίες για την εργασία μας. Μπορούμε να επιλέξουμε SKIP για να μπούμε κατευθείαν στο κύριο μένουν της εφαρμογής, Να πατήσουμε το βελάκι και να μεταφέρει σε μια δεύτερη οθόνη για να πατήσουμε DONE. Και το οποίο θα μας μεταφέρει και πάλι στην κύρια οθόνη για να ξεκινήσουμε ο παιχνίδι.



Εικόνα 43: Welcome panel

### 4.3.2 Δημιουργία εμφανισιακού περιεχομένου της πρώτης οθόνης

Στην πρώτη οθόνη θα δούμε το κύριο μενού που αποτελείται από τρία κουμπιά. Τα όποια το κάθε ένα από αυτά επεξεργάζεται τα δεδομένα της βάσης μας ώστε να «τρέξουν» σωστά και να μας μεταφέρουν σε μια επόμενη οθόνη συνέχισης του παιχνιδιού. Ως εκ τούτου λοιπόν θα χρειαστεί να δημιουργήσουμε μια linear λίστα με σκοπό να φορτώσουμε τα δεδομένα. Που σημαίνει ότι μπορούμε να στοιχίσουμε την όψη των κουμπιών ένα προς ένα Αυτό επιτυγχάνεται με τον παρακάτω κώδικα:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/welcome_fragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:clipChildren="false"
    android:background="@drawable/abc1"
    android:orientation="vertical"
    android:layout_weight="6"
    android:paddingTop="48dp">

    <FrameLayout
        android:id="@+id/image_container"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="4"
        android:clipChildren="false">

        <ImageView
            android:id="@+id/img"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:scaleType="centerInside"
            android:src="@drawable/ic_local_library"
            android:visibility="invisible"
            android:paddingLeft="18dp"
            android:paddingRight="18dp"
            android:layout_gravity="center"/>

    </FrameLayout>
```

Εικόνα 44: Linear listing

### 4.3.3 Κύριο μενού επιλογών- Κύρια οθόνη.

Ως εκ τούτου μέσω του παραπάνω κώδικα βλέπουμε τη μορφή που παίρνουν τα στοιχεία που έχουμε εισάγει στο κύριο μενού της εφαρμογής. Το κύριο μενού, αποτελείται από τις εξής επιλογές πλοήγησης:

- ✚ Νέο παιχνίδι (start game)
- ✚ Κατηγορίες (categories)
- ✚ Επιλογές (options)

Και το layout θα έχει το εξής αποτέλεσμα

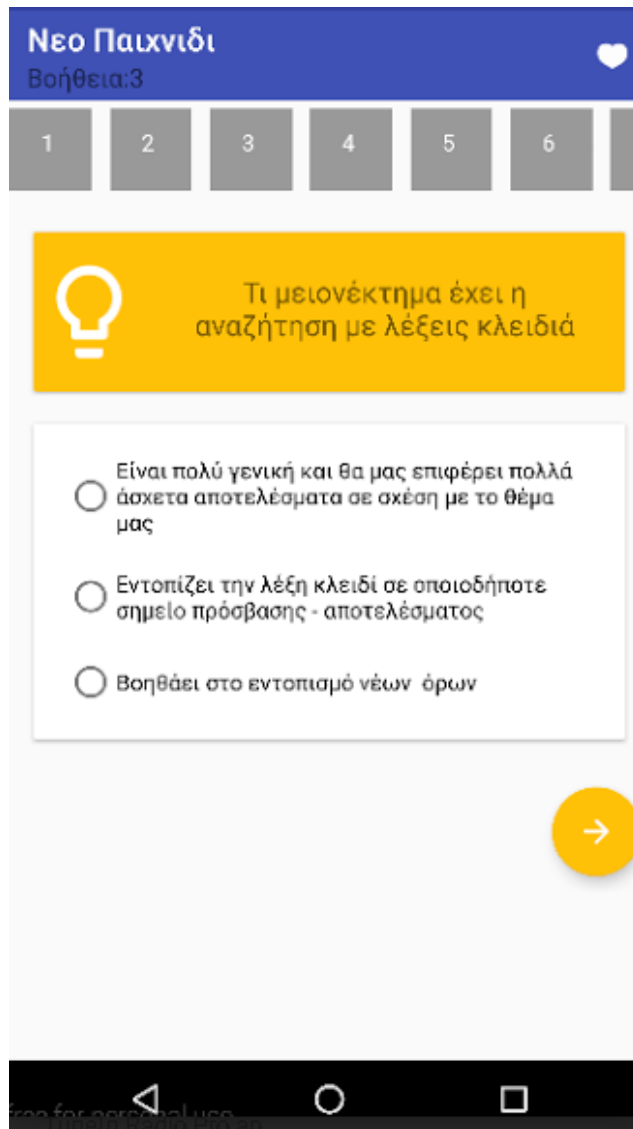


Εικόνα 45: Main menu Layout



#### 4.3.4 Αρχική οθόνη (Start menu).

Οι ερωτήσεις εμφανίζονται ανα διαφορετική κατηγορία τη φορά. Η οθόνη των ερωτήσεων θα είναι υπο την διάταξη λίστας, και θα υπάρχει το κουμπί που θα μας μεταφέρει στην επόμενη. Επίσης μπορούμε να βλέπουμε αριθμητικά την ακολουθία των ερωτήσεων στο πάνω μέρος της οθόνης μας. Η παρακάτω εικόνα μας δείχνει το αποτέλεσμα

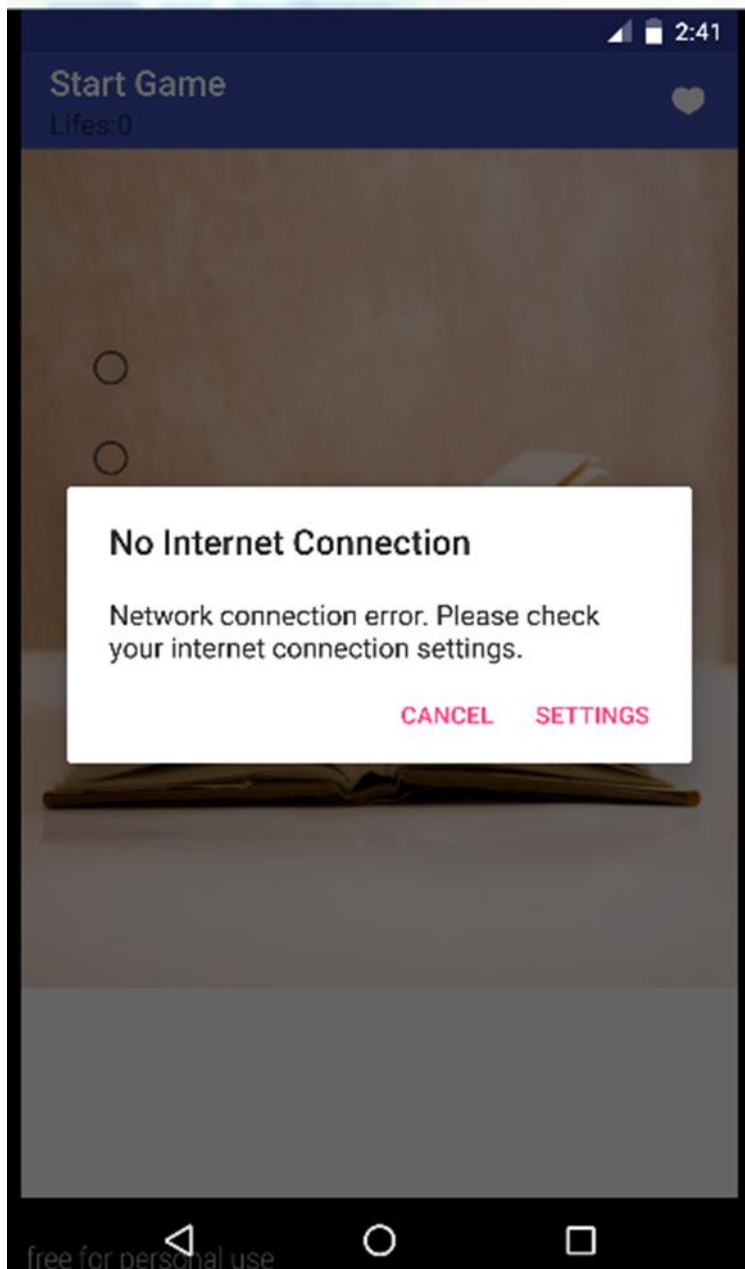


Εικόνα 46: New game

Λόγο του ότι η εφαρμογή μας είναι μια δυναμική εφαρμογή, καθώς επεξεργάζεται τα δεδομένα μας από τον 'server, σε real time υπάρχει προαπαίτηση να είναι συνδεδεμένη στο διαδίκτυο ώστε να γίνεται η σωστή σύνδεση και να παίρνουμε τα δεδομένα από τη βάση. Έτσι λοιπόν κατά την προσπάθεια του χρήστη να ξεκινήσει ένα νέο παιχνίδι στο κύριο μενού, ελέγχεται αν η συσκευή είναι συνδεδεμένη στο Ίντερνετ πράγμα τα οποίο επιτυγχάνεται με τον παρακάτω κώδικα: (όπως επίσης έχουμε αναλύσει στην ενότητα 3.2.2)

```
private void checkNetwork() {
    connectivityManager = (ConnectivityManager)
    getSystemService(Context.CONNECTIVITY_SERVICE);
    networkInfo = connectivityManager.getActiveNetworkInfo();
    network_connected = networkInfo != null && networkInfo.isAvailable() &&
    networkInfo.isConnectedOrConnecting();
    if (!network_connected) {
        DialogMessageDisplay.displayWifiSettingsDialog(CategoryActivity.this,
        CategoryActivity.this, getString(R.string.wifi_off_title),
        getString(R.string.wifi_off_message));
    } else {
        setupCategoryView();
    }
}
```

Στην περίπτωση που η συσκευή είναι εκτός δικτύου θα υπάρχει και το ανάλογο μήνυμα ενημέρωσης. Ακριβώς όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 47: No internet

### 4.3.5 Οθόνη κατηγοριών - λίστα κατηγοριών.

Στη δεύτερη οθόνη βλέπουμε παρατιθέμενη την λίστα των κατηγοριών που απαρτίζουν συγκεκριμένο σύνολο ερωτήσεων. Όπως βλέπουμε στην παρακάτω εικόνα.

Οι κατηγορίες τις οποίες έχουμε συμπεριλάβει είναι οι εξής.

- Οργάνωση πληροφοριών
- Ηλεκτρονικός κατάλογος
- Περιοδικά
- Βάσης δεδομένων
- Διαδίκτυο
- Βιβλιογραφικές παραπομπές



Εικόνα 48: Categories Layout



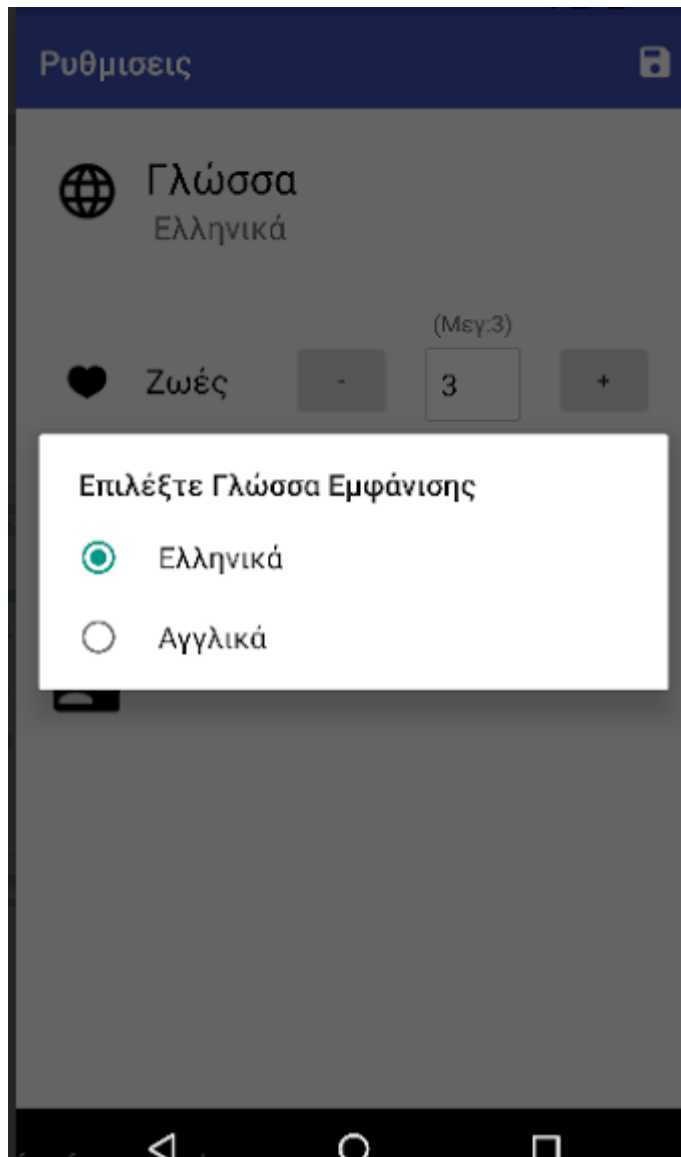
#### 4.3.6 Επιλογές –οθόνη ρυθμίσεων

Το τελευταίο κουμπί του Μενού μας μεταφέρει σε στην οθόνη των ρυθμίσεων. Από εκεί μπορούμε να αλλάξουμε γλώσσα καθώς η βάση των δεδομένων μας έχει μεταφραστεί και στα αγγλικά. Μπορούμε να επιλέξουμε ζώες επιτυχίας να βρούμε πληροφορίες για την εφαρμογή καθώς και να επικοινωνήσουμε με την βιβλιοθήκη του ΤΕΙ Ηρακλείου Κρήτης.



Εικόνα 49: settings

❖ **Γλώσσα:** Στις επιλογές της εφαρμογής μας έχουμε τη δυνατότητα να αλλάξουμε τη γλώσσα απεικόνισης των επιλογών είτε στα Ελληνικά είτε στα Αγγλικά.



Εικόνα 50: Language selection

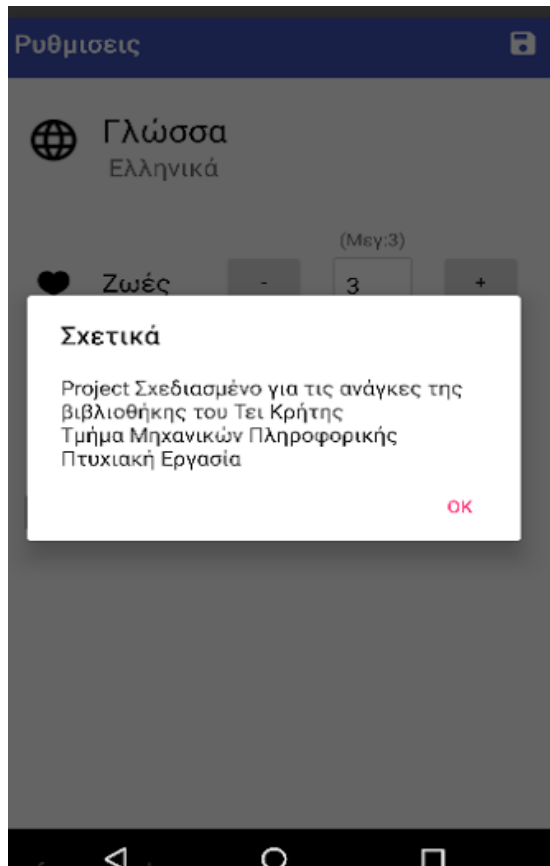
Αφού έχουμε επιλέξει τη γλώσσα μπορούμε να επιλέξουμε πόσες ζωές θέλουμε αφού έχουμε επιλέξει τη γλώσσα πατάμε save. Εάν προσπαθήσουμε να αλλάξουμε μόνο το πεδίο “ζωές” και πατήσουμε αποθήκευση, η εφαρμογή για γρασάρει καθώς κάθε φορά που ορίζεται η γλώσσα τότε ένα αρχείο “γεννιέται” στο οποίο αποθηκεύονται οι αλλαγές και το οποίο αποθηκεύεται στα sharefolders.

Η μέθοδος που χρησιμοποιούμε στο πρόγραμμα μας για να φορτώσουμε τις ρυθμίσεις του χρήστη από το αρχείο μας είναι η εξής:

```
private void loadUserPrefs() {
    sharedPreferences = getSharedPreferences(Constants.PREFERENCES_FILE,
MODE_PRIVATE);
    langFromPrefs =
sharedPreferences.getString(Constants.LANGUAGE_PREFS_FILE,
getString(R.string.ta_to_select));
}
```

```
livesFromPrefs =  
sharedPreferences.getString(Constants.LIFES_PREFS_FILE, String.valueOf(0));  
langFromPrefs = StringGenerator.revertLanguageCode(langFromPrefs,  
OptionsActivity.this);  
displayLanguage.setText(langFromPrefs);  
lives.setText(livesFromPrefs);
```

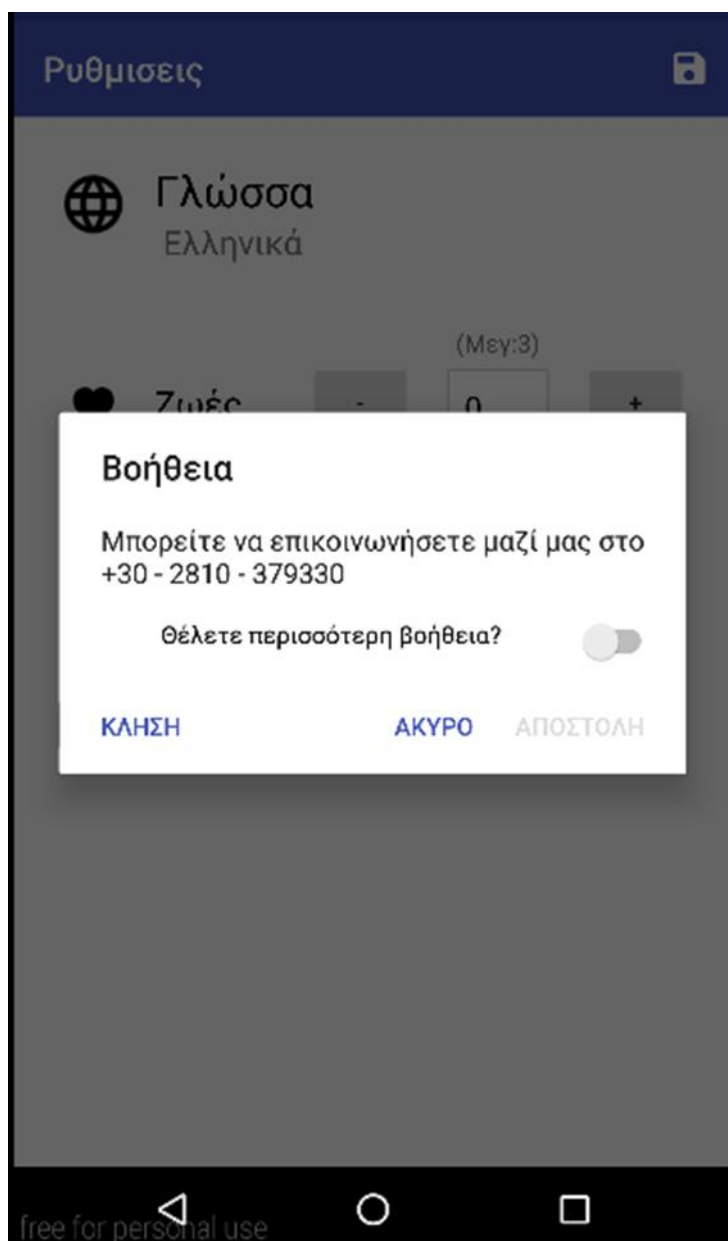
❖ **Σχετικά:** Βρίσκουμε πληροφορίες σχετικά με το παρών προτζεκτ.



Εικόνα 51: info alert

❖ **Επικοινωνία** Ένα επίσης σπουδαίο κατασκεύασμα της εφαρμογής είναι η δυνατότητα επικοινωνίας που μπορεί να αποκτήσει ο χρήστης μέσω της πλατφόρμας επικοινωνίας με την βιβλιοθήκη του Ιδρύματος μας.

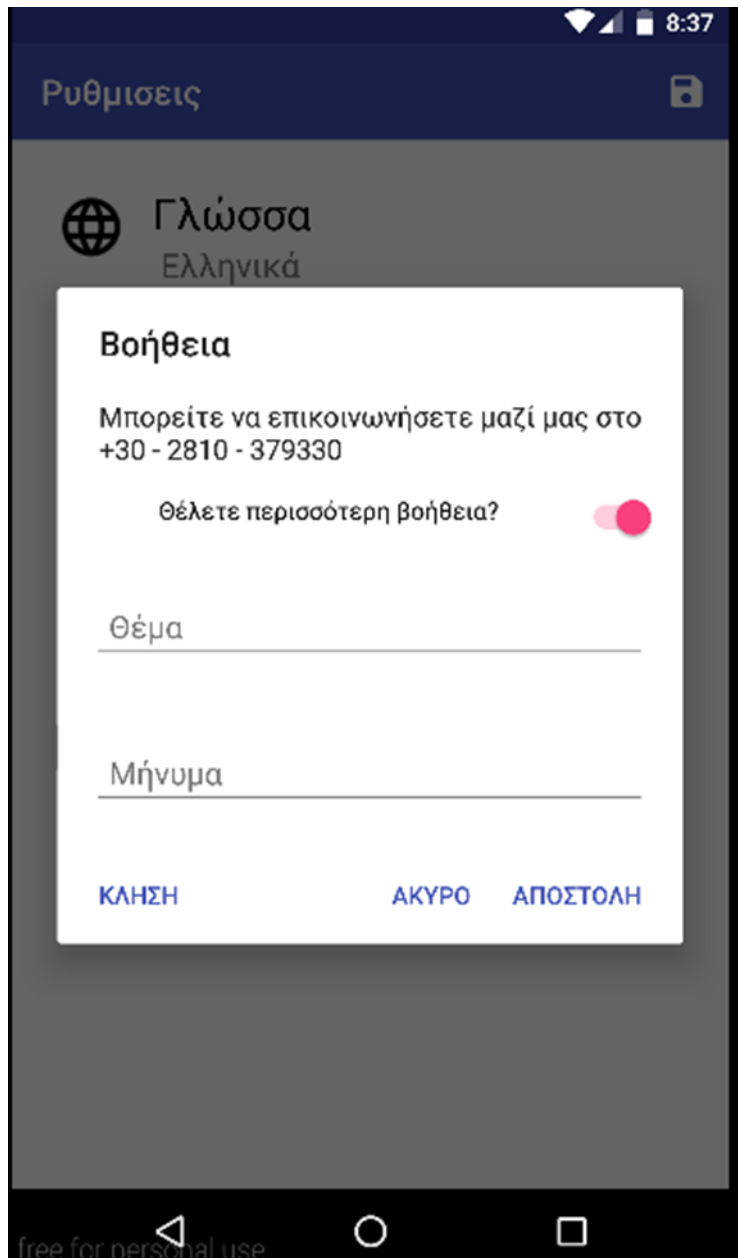




Εικόνα 52:contact Layout

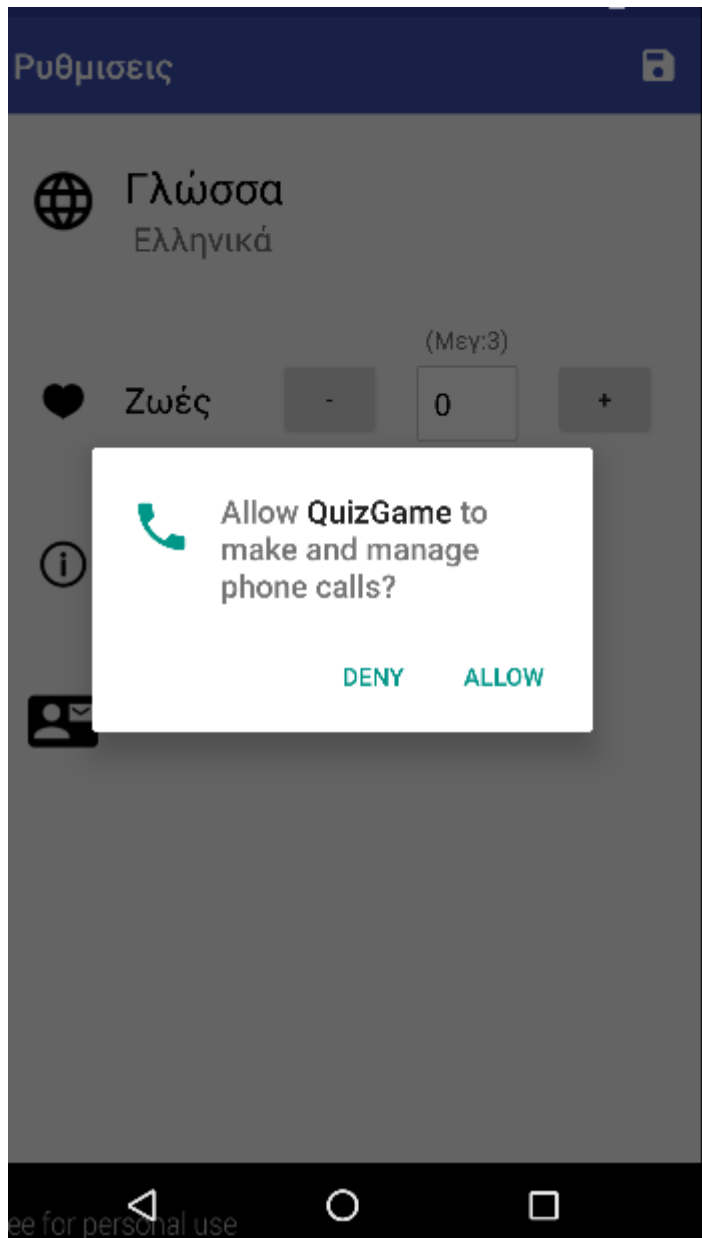
Η πλατφόρμα επικοινωνίας μας εμφανίζει ένα παράθυρο που αναγράφεται το τηλέφωνο επικοινωνίας της βιβλιοθήκης του ΤΕΙ Κρήτης καθώς και τη δυνατότητα για περαιτέρω βοήθεια μέσω ενός διακόπτη on/off.

Ανοίγοντας το διακόπτη βλέπουμε ένα ποιο αναπτυγμένο μενού, στο οποίο μπορούμε να συντάξουμε ένα μήνυμα ηλεκτρονικού ταχυδρομείου με θέμα και περιεχόμενο. Πατώντας αποστολή θα δρομολογηθεί το μήνυμά μας στην ηλεκτρονική διεύθυνση libreq@staff.teicrete.gr που ανταποκρίνεται στην e-mail της βιβλιοθήκης .



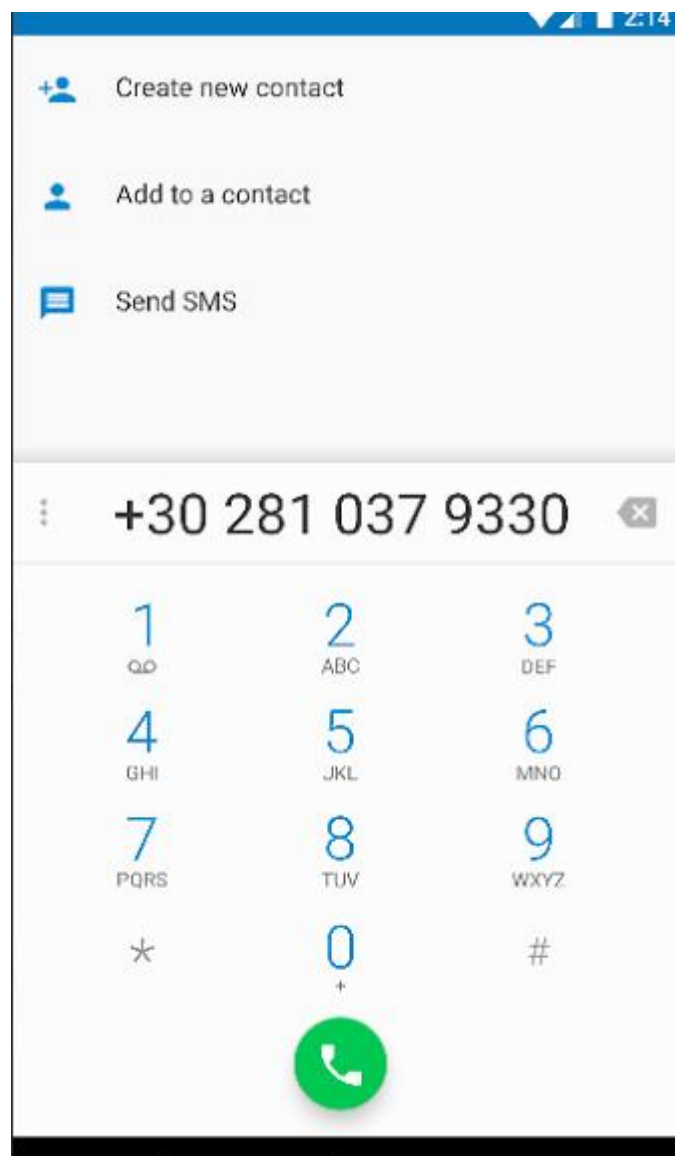
Εικόνα 53: compose an email

Πατώντας την επιλογή κλήση, ο χρήστης λαμβάνει ένα μήνυμα κειμένου που τον ρωτάει για την άδεια του χρήστη που θα σώσει στην συσκευή του ώστε να πραγματοποιήσει την συγκεκριμένη κλήση.



Εικόνα 54: Warning message

Στην περίπτωση που ο χρήστης δεχθεί να πραγματοποιήσει αυτή την κλήση αυτόματα θα τον παραπέμψει στο call menu της συσκευής του. όπως βλέπουμε στο παρακάτω σχήμα.



Εικόνα 55: Call number

## 5 Συμπεράσματα.

Εν κατακλείδι, διαπιστώνουμε ότι η διάθεση ενός προγράμματος πληροφοριακής παιδείας αποτελεί προϊόν που σχετίζεται με διάφορες εσωτερικές και εξωτερικές παραμέτρους της ακαδημαϊκής βιβλιοθήκης.

Από τα παραπάνω κεφάλαια παρατηρούμε ότι η πλατφόρμα της Google, Android παρέχει σε μια γενική έννοια ένα ολοκληρωμένο λειτουργικό σύστημα για κινητές συσκευές. Αποτελεί μια ολοκληρωμένη πλατφόρμα ανάπτυξης εφαρμογών και προσφέρει πληθώρα εργαλείων και μεθόδων και παρέχει αρκετά και τεκμηριωμένα κείμενα ανάπτυξης εφαρμογών και του τρόπος λειτουργίας του συστήματος. Μπορεί να καλύψει τις ανάγκες τόσο των μέσων όσο και των εξειδικευμένων χρηστών. Επίσης, οι κατασκευαστές των κινητών συσκευών έχουν τη δυνατότητα να προσαρμόσουν ανάλογα με τις ανάγκες τους το λειτουργικό σύστημα android χωρίς κόστος.

Μέσα από την ενασχόληση των παραπάνω τεχνολογιών που μελετήσαμε καταλήγουμε στην επίτευξη του στόχο μας.

Στόχος μας λοιπόν, ήταν η εφαρμογή που αναπτύχθηκε να λειτουργεί γενικότερα σε συσκευές που υποστηρίζονται από λειτουργικό σύστημα android. Συνεπώς, η εφαρμογή δοκιμάστηκε σε εικονικές συσκευές αλλά και σε πραγματικά τηλέφωνα όπως περιγράψαμε και στην εισαγωγή και λειτούργησε με επιτυχία. Επιπλέον, η εφαρμογή που αναπτύχθηκε θα μπορούσε με αρκετή επιτυχία να είναι διαθέσιμη, ως μια εφαρμογή διαπαιδαγώγησης και γνώσης απευθυνόμενη σε ευρύ κοινό. Επιπροσθέτως δε, με χρησιμοποιώντας κάποια επιπλέον χαρακτηριστικά στην εφαρμογή, θα μπορούσε εύκολα να εξελιχθεί σε ένα σπουδαίο βοήθημα, για την απόκτησης της γνώσης που αφορά την λειτουργία μια βιβλιοθήκης,

## 5.1 Αναφορές

- "Android - UI Layouts." *TutoriasPoint*. n.d.  
[http://www.tutorialspoint.com/android/android\\_user\\_interface\\_layouts.htm](http://www.tutorialspoint.com/android/android_user_interface_layouts.htm).
- Burnette, Ed. "Android Introducing Google's Development Platform." In *Hello*. 2nd edition, n.d. Developers. "Debugging." n.d.  
<https://stuff.mit.edu/afs/sipb/project/android/docs/tools/debugging/index.html>.
- . "Platform Versions." *Data collected during a 7-day period ending on September 5, 2016*. n.d.  
<https://developer.android.com/about/dashboards/index.html>.
- Developers, Android. *Android Supported Media Formats*, 2009: May.
- . "What is Android?" July 21, 2009.
- Krasnoff. *CODE PROJECT, for those who code*. May 31, 2012.  
<http://www.codeproject.com/Articles/395614/Basic-structure-of-an-Android-project>.
- Madfortech. *Γιατί επιλέγω το Android*. January 10, 2013.  
<http://www.madfortech.com/2013/01/giati-epilegw-android/>.
- Sharma, Rupali . "Developing for Android - An Introduction." *Cprogramming.com*. n.d.  
[http://www.cprogramming.com/android/android\\_getting\\_started.html](http://www.cprogramming.com/android/android_getting_started.html).
- Studio, Android. *The Official IDE for Android*. copyright (c) 1995-2015 International Business Machines Corporation and others. <https://developer.android.com/studio/index.html>.
- Topolsky, Joshua. "Google's Android OS early look SDK now available." *Engadget*. , 2007: November.
- Wiki. *PHP*. August 18, 2016. <https://en.wikipedia.org/wiki/PHP>.

*Wikipedia*. n.d. [https://en.wikipedia.org/wiki/Android\\_%28operating\\_system%29](https://en.wikipedia.org/wiki/Android_%28operating_system%29).

01.

