Ατομική Διπλωματική Εργασία

# “ΕΦΑΡΜΟΓΗ ANDROID - ΥΛΟΠΟΙΗΣΗ ΒΙΒΛΙΟΘΗΚΗΣ”

**Γεωργία Ανδρέου**

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ**

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

Μάιος 2013

# ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**


## “Εφαρμογή Android - Υλοποίηση βιβλιοθήκης”


## Γεωργία Ανδρέου

Επιβλέπων Καθηγητής

Παρασκευάς Ευριπίδου



Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2013

# Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω θερμά τον επιβλέπον καθηγητή της παρούσας διπλωματικής εργασίας, κύριο Παρασκευά Ευριπίδου, για την ευκαιρία που μου πρόσφερε μιας και μέσα από την καθοδήγηση του αλλά και με τον εξοπλισμό που μου παρείχε κατάφερα να εργαστώ εποικοδομητικά και να φέρω εις πέρας την εν λόγω διπλωματική εργασία. Πρόκειται για έναν εξαίρετο καθηγητή με απεριόριστες γνώσεις σε πολλούς τομείς, και ιδιαίτερα στους τομείς που αφορούν την επιστήμη της Πληροφορικής, αλλά πάνω από όλα για έναν εξαίρετο άνθρωπο πάντα πρόθυμο και με κατανόηση.

Στη συνέχεια θα ήθελα να ευχαριστήσω τους φίλους και συναδέλφους μου για την πολύτιμη βοήθεια αλλά και για τις συμβουλές τους στις δυσκολίες που αντιμετώπισα κατά την υλοποίηση της διπλωματικής μου εργασίας.

Τέλος, οφείλω ένα τεράστιο ευχαριστώ στην οικογένεια μου, και ιδιαίτερα στους γονείς μου Παναγιώτη και Μαρία που με στήριξαν ηθικά και οικονομικά καθ'όλη την διάρκεια των σπουδών μου, όπως επίσης και στους φίλους μου για όλη τη στήριξη και τη συμπαράσταση που μου παρείχαν όλο αυτό τον καιρό.

# Περίληψη

Από την ίδρυσή τους, τα κινητά τηλέφωνα έχουν γίνει αναπόσπαστο μέρος της ζωής σχεδόν όλων μας στον ανεπτυγμένο κόσμο. Οι πρόοδοι στην τεχνολογία κινητών επικοινωνιών έχουν οδηγήσει σε τεράστια αύξηση, εξαιρετικά ικανά έξυπνα τηλέφωνα. Τα έξυπνα κινητά τηλέφωνα επιτρέπουν στους χρήστες να τρέχουν εφαρμογές λογισμικού που χρησιμοποιούνται για την οργάνωση, εξοικονόμηση χρόνου ή απλά για διασκέδαση.

Στην παρούσα πτυχιακή εργασία αναπτύχθηκε μια εφαρμογή κινητού τηλεφώνου με χρήση της πλατφόρμας Google Android. Στόχος της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη και υλοποίηση μιας εφαρμογής Android , η οποία έχει ως θέμα μια βιβλιοθήκη. Στην εφαρμογή αυτή, υπάρχουν ήδη ενσωματωμένα βιβλία τύπου pdf και ο χρήστης είναι σε θέση να κάνει αναζήτηση για το βιβλίο που τον ενδιαφέρει. Σε αυτή την εφαρμογή περιέχονται μόνο ελληνικά βιβλία. Επίσης, ο χρήστης έχει την δυνατότητα να ανεβάσει οποιοδήποτε βιβλίο επιθυμεί στην συγκεκριμένη εφαρμογή, το οποίο παρομοίως πρέπει να είναι τύπου pdf. Στην εφαρμογή υπάρχει επίσης μία κατηγορία About, στην οποία υπάρχει μια μικρή περιγραφή αυτής της εφαρμογής. Και τέλος, υπάρχει μία κατηγορία Settings, στην οποία ο χρήστης μπορεί να προσαρμόσει την φωτεινότητα της εφαρμογής είτε σε πιο φωτεινή είτε σε πιο σκοτεινή οθόνη. Επίσης, δίνεται η δυνατότητα στον χρήστη να στείλει email σε μια προκαθορισμένη διεύθυνση που έχει οριστεί, τοποθετώντας σχόλια για την εφαρμογή είτε κάποιες απορίες που μπορεί να υπάρξουν σχετικά με την λειτουργία της εφαρμογής.

Ο κύριος στόχος της πτυχιακής εργασίας ήταν η ανάπτυξη και ο σχεδιασμός του ενός λογισμικού κινητού τηλεφώνου για το λειτουργικό σύστημα Android της Google με σκοπό την ανάδειξη των δυνατοτήτων αυτής της ραγδαίας αναπτυσσόμενης πλατφόρμας. Για την επίτευξη του στόχου χρησιμοποιήθηκε το λογισμικό Eclipse το οποίο είναι ένα πρόγραμμα ανοιχτού κώδικα που σε συνεργασία με τα δωρεάν εργαλεία ανάπτυξης Android λογισμικού που προσφέρει η Google (Android Development Tools) αποτελεί ένα πολύ ισχυρό εργαλείο ανάπτυξης εφαρμογών Android.

# Περιεχόμενα

# Κεφάλαιο 1

## Εισαγωγή

---



---

### 1.1 Αντικείμενο της διπλωματικής

Τα τελευταία χρόνια βιώνουμε μία επανάσταση των έξυπνων τηλεφώνων (smartphones), τα οποία ενσωματώνουν πλέον δυνατότητες, οι οποίες δεν υπάρχουν στα συμβατικά κινητά τηλέφωνα. Περιλαμβάνουν ολοκληρωμένο λειτουργικό σύστημα, ισχυρό επεξεργαστή ικανό για εκτέλεση πολύπλοκων και χρονοβόρων υπολογισμών, δυνατότητα σύνδεσης στο διαδίκτυο είτε μέσω ασύρματων δικτύων (Wi-Fi) είτε δικτύων τρίτης γενιάς (3G networks), κάμερα υψηλής ανάλυσης, συστήματα εντοπισμού θέσης (GPS) κ.α. Τα λειτουργικά συστήματα των συσκευών αυτών ποικίλουν, με το Android της Google να περιλαμβάνεται τη στιγμή αυτή στο μεγαλύτερο ποσοστό των smartphones της αγοράς. Μάλιστα επίσημα στοιχεία δείχνουν οτι η ιστοσελίδα της Google, Android Market, που περιέχει όλες τις εφαρμογές για Android περιλαμβάνει αυτή τη στιγμή περισσότερες από 250.000 εφαρμογές.

Στο πλαίσιο αυτό, σκοπός της παρούσας διπλωματικής εργασίας, είναι η ανάπτυξη μίας εφαρμογής που προορίζεται για τα παραπάνω κινητά τηλέφωνα. Φυσικά, η εφαρμογή μπορεί να εκτελεστεί όχι μόνο σε κινητά τηλέφωνα αλλά και σε οποιαδήποτε άλλη ηλεκτρονική

συσκευή που περιλαμβάνει λειτουργικό σύστημα Android (όπως tablet pc). Η εφαρμογή iReader παρέχει τη δυνατότητα αναζήτησης των βιβλίων καθώς και ανέβασμα βιβλίων και μπορεί να χρησιμοποιηθεί από οποιονδήποτε χρήστη κατέχει Android έξυπνη κινητή συσκευή. Βασική προϋπόθεση, όπως και για τις πλείστες εφαρμογές που απευθύνονται σε έξυπνες κινητές συσκευές, είναι η πρόσβαση στο διαδίκτυο.

## 1.2 Σχετικά Άρθρα και Μελέτη

Πρώτη εκτενής εισαγωγή κατά την περίοδο προσανατολισμού, για να πάρω μια ιδέα σχετικά με την πλατφόρμα Android έχω διαβάσει το Hello Android, Introducing Google's Mobile Development Platform, 3rd edition από Ed Burnette. Στην συνέχεια έχω διαβάσει Professional Android Application Development από Reto Meier. Ακολούθως, άρχισα να προγραμματίζω σε πραγματική εφαρμογή. Ένα μήνα στον προγραμματισμό, άρχισα να τεκμηριώνω τα βασικά για αυτή την εργασία, συνεχίζοντας παράλληλα την περαιτέρω ανάπτυξη της εφαρμογής.

## 1.3 Οργάνωση κειμένου

Η διπλωματική αυτή εργασία αναλύεται συνολικά σε 6 κεφάλαια. Συγκεκριμένα έχουμε:
Στο 1ο Κεφάλαιο ασχολούμαστε με την εισαγωγή στο αντικείμενο που πραγματεύεται η διπλωματική εργασία. Εισάγουμε τον αναγνώστη στις νέες τεχνολογίες που χρησιμοποιούνται στις σημερινές κινητές συσκευές (mobile devices) και τον πληροφορούμε σχετικά με τον σκοπό συγγραφής της διπλωματικής εργασίας και τον τρόπο που συνδέεται αυτή με τις τεχνολογίες.

Στο 2ο Κεφάλαιο εντάσσουμε τον χρήστη στο λειτουργικό σύστημα Android περιγράφοντας το τι είναι, τις εκδόσεις και τα ιστορικά μέχρι και σήμερα του λειτουργικού αυτού, την αρχιτεκτονική του λειτουργικού συστήματος Android και γιατί επιλέγηκε το λειτουργικό σύστημα Android για την ανάπτυξη αυτής της εφαρμογής.

Το 3$^{ο}$ Κεφάλαιο περιγράφει το λειτουργικό σύστημα Android και εισάγει τον αναγνώστη στον προγραμματισμό στο περιβάλλον αυτό. Εξηγούνται τα προγραμματιστικά εργασλεία που χρειάζονται και αναλύονται τα βασικά βήματα που απαιτούνται, για την ανάπτυξη μίας εφαρμογής που θα εκτελείται σε λειτουργικό σύστημα Android. Το κεφάλαιο αυτό, λοιπόν, αποτελεί ουσιαστικά έναν εισαγωγικό οδηγό που πρέπει να ακολουθήσει ο προγραμματιστής ώστε να κατασκευάσει μία εφαρμογή Android.

Στο 4$^{ο}$ Κεφάλαιο παρουσιάζεται αναλυτικά η αρχιτεκτονική της εφαρμογής που υλοποιήθηκε στο πλαίσιο της διπλωματικής αυτής εργασίας. Αρχικά αναφέρονται οι διάφορες λειτουργίες και δυνατότητες της εφαρμογής και έπειτα περιγράφεται η δομή της, το πώς δηλαδή οργανώνονται όλα τα αρχεία που περιέχονται σε αυτήν και ποια είναι η κύρια λειτουργία τους.

Στο 5$^{ο}$ Κεφάλαιο παρουσιάζονται διεξοδικά όλες οι λειτουργίες της εφαρμογής και οι επιλογές που έχει ο χρήστης αλληλεπιδρώντας με αυτήν. Περιγράφεται κάθε διαφορετική οθόνη της εφαρμογής, οι επιλογές που έχει ο χρήστης στη συγκεκριμένη οθόνη. Όλα τα αποτελέσματα παρουσιάζονται σε εικόνες, οι οποίες ουσιαστικά εμφανίζουν ένα στιγμιότυπο από την αντίστοιχη οθόνη της συσκευής που εκτελεί την εφαρμογή, και βοηθάνε τον αναγνώστη στην κατανόηση των παραπάνω λειτουργιών.

Στο 6$^{ο}$ Κεφάλαιο περιγράφονται τα συμπεράσματα που προκύπτουν από την εκπόνηση της διπλωματικής εργασίας, καθώς και οι μελλοντικές εξελίξεις της εφαρμογής. Αναφέρονται, ακόμα, συγκεκριμένες επεκτάσεις που σχεδιάζουμε να συμπεριλάβουμε στο προσεχές διάστημα στην εφαρμογή αυτή.

Στο Παράρτημα της εργασίας αυτής, παρατίθεται αναλυτικά ο κώδικας της εφαρμογής που αναπτύχθηκε και έπειτα τα υπόλοιπα αρχεία που περιλαμβάνει το project της εφαρμογής.

Τέλος, παρουσιάζεται η Βιβλιογραφία που χρησιμοποιήθηκε για τη συγγραφή του παρόντος κειμένου, καθώς και όλες οι ιστοσελίδες που βοήθησαν στην ανάπτυξη της εφαρμογής.

# Κεφάλαιο 2

# Android

---



---

## Android

Ένας από τους πρωταρχικούς σκοπούς σε αυτήν την εργασία, ήταν η επιλογή μιας πλατφόρμας, η οποία θα μπορούσε να υποστηρίξει τον σχεδιασμό και την υλοποίηση μίας βιβλιοθήκης. Ταυτόχρονα αναζητούσαμε κάτι το καινούργιο στον χώρο που θα άξιζε την μελέτη του και παράλληλα θα ήταν πολλά υποσχόμενο. Μέσα σε λίγα λεπτά καταλήξαμε σε δύο υποψήφιες πλατφόρμες, το iPhone και το Android. Οι δύο αυτές πλατφόρμες φαίνεται ότι θα πρωταγωνιστήσουν στον χώρο των έξυπνων τηλεφώνων (smart phones), αφού διαθέτουν εξαιρετικά χαρακτηριστικά και απίστευτες δυνατότητες.


Εικόνα 2.1: Η μάχη μεταξύ Android και iphone βρίσκεται σε εξέλιξη

Στις αρχές Οκτωβρίου 2009, το iPhone κατείχε τα σκήπτρα στις πωλήσεις. Παρόλα αυτά με την ραγδαία ανάπτυξη που είχε πάρει το Android, οι ερευνητές μιλούσαν ότι τα δεδομένα θα έχουν ανατραπεί μέχρι το 2012 το γρηγορότερο. Προς έκπληξη αρκετών, την ώρα που γράφεται αυτή η εργασία (Μάιος 2013), το Android έχει ήδη ξεπεράσει προπολλού σε πωλήσεις το iPhone και συνεχίζει να επεκτείνεται. Ωστόσο είναι μια μάχη η οποία αναμένετε να συνεχιστεί, με τις δύο πλατφόρμες να εκσυγχρονίζουν ανά τακτά χρονικά διαστήματα τα χαρακτηριστικά τους.

## 2.1 Τι είναι το Android

Το Android είναι μια στοίβα λογισμικού για κινητές συσκευές η οποία περιλαμβάνει λειτουργικό σύστημα, ενδιάμεσο λογισμικό (middleware) και βασικές εφαρμογές. Το Android τρέχει τον πυρήνα του λειτουργικού Linux και μέσω της δικιάς του εργαλειοθήκης ανάπτυξης συστήματος λογισμικού (Software Development Kit), επιτρέπει στους κατασκευαστές να δημιουργούν πρωτοποριακές εφαρμογές. Αρχικά αναπτύχθηκε από την Google και αργότερα συνεχίστηκε σε συνεργασία με την Open Handset Alliance (OHA). Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την

ανακοίνωση της ίδρυσης του οργανισμού OHA, μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής υλικού, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές ανοιχτής τηλεφωνίας.

Ενδεικτικά αναφέρουμε μερικά μέλη του οργανισμού αυτού (Εικόνα 2.2) για να δείξουμε την τεράστια προοπτική που δημιουργείται:

- Sprint Nextel
- T-Mobile
- Motorola
- Samsung
- Sony Ericsson
- Vodafone
- Google
- Verizon
- Texas Instruments
- Htc


Εικόνα 2.2: Εταιρίες λογισμικού και κατασκευής υλικού παγκόσμιας εμβέλειας

Η Google δημοσίευσε το μεγαύτερο μέρος του κώδικα του Android, υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού.


Εικόνα 2.3: Λογότυπο πλατφόρμας Android

## 2.2 Ιστορικά - Εκδόσεις και χαρακτηριστικά

Η πρώτη έκδοση του Android SDK τον Νοέμβριο του 2007, χαρακτηρίστηκε από τους κατασκευαστές του σαν μια πρώτη ματιά στο SDK του Android, κάτι το οποίο πολλοί παράβλεψαν και βιάστηκαν να κατακρίνουν το Android σαν ένα προβληματικό σύστημα. Στην ουσία όμως το Android δεν παρουσίαζε προβλήματα τα οποία δεν παρουσιάζει οποιοδήποτε σύστημα σε τέτοια πρώιμη φάση. Έτσι το Σεπτέμβριο του 2008, η T-Mobile ανακοινώνει την διαθεσιμότητα του T-Mobile G1, του πρώτου έξυπνου τηλεφώνου (smartphone), βασισμένο στην πλατφόρμα του Android. Λίγες μέρες αργότερα (Οκτώβριο 2008), η Google ανακοινώνει την απελευθέρωση του SDK Release Candidate 1.0. Ακολούθησε τον Φεβρουάριο του 2009 η έκδοση 1.1 σαν μια ανανεωμένη έκδοση του 1.0. Μέχρι τότε το Android δεν υποστήριζε ακόμη την χρήση κουμπιών αφής, παρά μόνο την χρήση των κλασσικών 'σκληρών' κουμπιών της συσκευής.

Τον Μάιο του 2009 είχαμε την έκδοση Android 1.5, εν ονόματι 'Cupcake'.

Εικόνα 2.4: Λογότυπο Android 1.5 CUPCAKE

Το ‘Cupcake’ εισάγει κάποια καινούργια χαρακτηριστικά και ανανεώσεις στην διεπιφάνεια χρήστη (User Interface):

- Ικανότητα για καταγραφή και παρακολούθηση βίντεο μέσα από την λειτουργία της βιντεοκάμερας, μεταφόρτωση βίντεο στο YouTube και φωτογραφιών στο Picasa απευθείας από το τηλέφωνο, καινούργιο μαλακό πληκτρολόγιο (αφής) με πρόβλεψη κειμένου
- Υποστήριξη προτύπου Bluetooth A2DP και AVRCP
- Ικανότητα αυτόματης σύνδεσης σε μικροσυσκευή Bluetooth από μια συγκεκριμένη απόσταση
- Καινούργια widgets και φάκελοι που μπορούν να δημοσιευτούν στην αρχική οθόνη
- Κινούμενες μεταβάσεις οθόνης

Το ‘Donut’, Android 1.6, ήρθε τον Σεπτέμβριο του 2009.

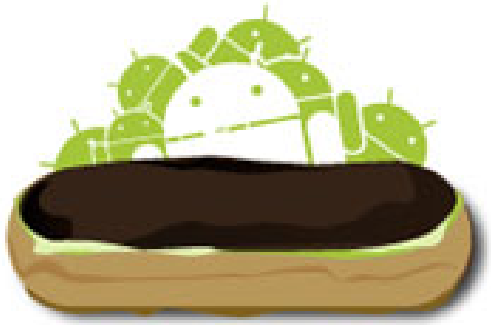
Εικόνα 2.5: Λογότυπο Android 1.6 DONUT

Η έκδοση αυτή εισάγει κάποια καινούργια χαρακτηριστικά όπως:

- Βελτιωμένο Android Market
- Ενσωματωμένη φωτογραφική μηχανή, βιντεοκάμερα και διεπαφή (interface) γκαλερί

- Η γκαλερί επιτρέπει πλέον στους χρήστες την επιλογή πολλαπλών φωτογραφιών για διαγραφή
- Ανανεωμένη αναζήτηση με φωνή, με ταχύτερη απόκριση και βαθύτερη ολοκλήρωση με εγγενής (native) εφαρμογές, συμπεριλαμβανομένης της δυνατότητας να καλούμε επαφές
- Ανανεωμένη αναζήτηση με την δυνατότητα αναζήτησης σελιδοδεικτών, ιστορικού, επαφών και στο διαδίκτυο από την αρχική οθόνη
- Ανανεωμένη υποστήριξη τεχνολογιών για CDMA/EVDO, 802.1x, VPNs και με μηχανή μετατροπής κειμένου σε ομιλία (text-to-speech)
- Υποστήριξη για ανάλυση οθονών WVGA
- Βελτιώσεις στην ταχύτητα για αναζήτηση και για εφαρμογές φωτογραφικής μηχανής

Ακολουθεί το 'Eclair', Android 2.0 τον Νοέμβριο 2009, με τις επανεκδόσεις του σε Android 2.0.1 τον Δεκέμβριο 2009 (Eclair 0.1) και τον Ιανουάριο 2010 με το Android 2.1 (Eclair MR1).



Εικόνα 2.6: Λογότυπο Android 2.0 ECLAIR

Ανάμεσα στις άλλες αλλαγές είναι και:

- Βέλτιστη ταχύτητα υλικού
- Υποστήριξη για περισσότερες οθόνες και αναλύσεις

- Βελτιωμένη διεπιφάνεια χρήστη
- Καινούργια διεπιφάνεια χρήσης για την μηχανή αναζήτησης και υποστήριξη του προτύπου HTML5
- Καινούργιες λίστες επαφών
- Καλύτερος λόγος άσπρου – μαύρου για φόντα
- Βελτιωμένοι χάρτες Google (google maps) 3.1.2
- Υποστήριξη Microsoft Exchange
- Ενσωματωμένη υποστήριξη flash για την Camera
- Ψηφιακή μεγέθυνση (zoom)
- Κλάση MotionEvent βελτιωμένη ώστε οι κατασκευαστές να μπορούν να παρακολουθούν αποτελεσματικότερα τα γεγονότα πολλαπλής αφής
- Ανανεωμένο εικονικό πληκτρολόγιο
- Bluetooth 2.1

Ακολουθεί το Android 2.2 με το όνομα ‘Froyo’ τον Μάιο του 2010.

Εικόνα 2.7: Λογότυπο Android 2.2 FROYO

Η έκδοση FROYO ανάμεσα σε άλλες αλλαγές περιλαμβάνει:

- Βελτιστοποιήσεις στην ταχύτητα γενικά του λειτουργικού συστήματος, στην μνήμη και στην απόδοση
- Ενσωμάτωση στην μηχανή αναζήτησης, της μηχανής Javascript του Chrome V8
- Αυξημένη υποστήριξη Microsoft Exchange (σε πολιτικές ασφαλείας, συγχρονισμού ημερολογίου, auto – discovery, GAL look-up, remote wipe)
- Βελτιωμένος προωθητής εφαρμογής (application launcher), με συντομεύσεις προς τις εφαρμογές τηλεφώνου και εφαρμογές της Μηχανής Αναζήτησης
- Πρόσδεση USB και λειτουργία δυναμικής ζώνης (hotspot) WiFiΑνανεωμένη εφαρμογή Αγοράς (Market) με αυτόματη ανανέωση
- Επιλογή για απαγόρευση πρόσβασης δεδομένων πάνω από ένα δίκτυο κινητής τηλεφωνίας
- Γρήγορη εναλλαγή ανάμεσα σε πολλαπλές γλώσσες του πληκτρολογίου και των λεξικών τους
- Φωνητική κλήση και διαμοιρασμός επαφών με Bluetooth
- Υποστήριξη για αριθμητικούς και αλφαριθμητικούς κωδικούς
- Η μηχανή αναζήτησης μπορεί να αποτυπώσει κινούμενα GIFs
- Υποστήριξη για πεδία μεταφόρτωσης αρχείων στην μηχανή αναζήτησης
- Υποστήριξη για εγκατάσταση εφαρμογών στην επεκτάσιμη μνήμη
- Υποστήριξη Adobe Flash 10.1

Ακολουθεί το Android 2.3 με το όνομα ‘Gingerbread’ τον Δεκέμβριο του 2010 με την επανέκδοση του σε Android 2.3.3 τον Φεβρουάριο του 2011.

Εικόνα 2.8: Λογότυπο Android 2.3 GINGERBREAD

Οι αλλαγές που έχουν γίνει είναι οι ακόλουθες:

- Βελτιωμένο UI για  απλότητα και ταχύτητα
- Πιο γρήγορη, πιο διαισθητική εισαγωγή κειμένου
- Επιλογή λέξεων και αντιγραφή/επικόλληση με ένα άγγιγμα
- Βελτιωμένη ενεργειακή διαχείριση υποστήριξη NFC (Near Field Communication)
- Υποστήριξη video κλήσης
- Υποστήριξη του πρωτόκολλου WebM για αναπαραγωγή video

Ακολουθεί η έκδοση Android 3.0 με το όνομα “Honeycomb” όπου είναι στην διάθεση των χρηστών και προγραμματιστών από τον Φεβρουάριο του 2011, λίγες μέρες μετά την επανέκδοση του Android 2.3.3, και προορίζεται αποκλειστικά για ταμπλέτες.

Εικόνα 2.9: Λογότυπο Android 3.0 HONEYCOMB

Μερικά από τα χαρακτηριστικά του είναι:

- Υποστηρίζει διπύρηνους και τετραπύρηνους επεξεργαστές
- Βελτιωμένη υποστήριξη των ταμπλετών
- Ανάπτυξη λογισμικού (scripting) για 3D, σε γλώσσα η οποία καλείται "Renderscript"
- Video chat μέσω Google Talk
- Google eBooks
- "Ιδιωτική περιήγηση"

Ακολουθεί η έκδοση Android 4.0 με το όνομα “Ice Cream Sandwich” , τον Οκτώβριο του 2011.

Εικόνα 2.10: Λογότυπο Android Ice Cream Sandwich

Μερικά από τα χαρακτηριστικά του είναι:

- Soft buttons από το Android 3.x είναι διαθέσιμα για χρήση σε κινητά τηλέφωνα
- Διαχωρισμός των widgets σε μία νέα καρτέλα, χωρισμένα με παρόμοιο τρόπο σε εφαρμογές
- Καλύτερη ενοποίηση φωνής και συνέχειας, real-time ομιλία για υπαγόρευση κείμενου
- Μια νέα οικογένεια γραματοσειράς για το UI, Roboto
- Νέο layout γκαλερί, οργανωμένο από την θέση και το πρόσωπο
- Υποστήριξη για το WebP image format

Ακολουθεί η έκδοση Android 4.1 με το όνομα “Jelly Bean” , τον Ιούλιο του 2012.

Εικόνα 2.11: Λογότυπο Android Jelly Bean

Μερικά από τα χαρακτηριστικά του είναι:

- Ομαλότερη διεπαφή χρήστη
- Χάρτες πληκτρολογίου εγκατεστημένοι από τον χρήστη
- Ικανότητα να απενεργοποιηθούν οι ειδοποιήσεις για μια εφαρμογή
- Βελτιωμένη φωνητική αναζήτηση
- Βελτιωμένη εφαρμογή κάμερας

Ακολουθεί η έκδοση Android 4.2 με το όνομα “Jelly Bean” , τον Νοέμβριο του 2013.

Μερικά από τα χαρακτηριστικά του είναι:

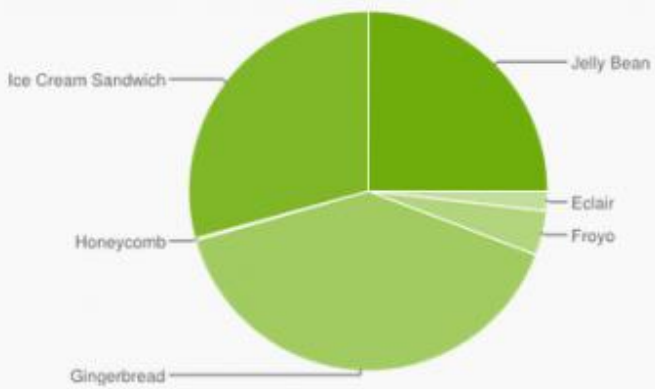
- “Photo Sphere” πανοραμικές φωτογραφίες
- Βελτιώσεις στην οθόνη κλειδώματος
- Υποστήριξη για ασύρματη οθόνη
- Βελτίωση της προσβασιμότητας

- Πολλαπλοί λογαριασμοί χρηστών (σε tablets μόνο)

Ένα ακόμα ενδιαφέρον στατιστικό που πρέπει να δούμε είναι τα ποσοστά των android εκδόσεων που είναι εγκατεστημένες σε όλες τις android συσκευές όπως αυτά ανακοινώθηκαν για τον Απρίλιο του 2013 (Σχήμα 2.12).

| Version | Codename | API | Distribution |
|---|---|---|---|
| 1.6 | Donut | 4 | 0.1% |
| 2.1 | Eclair | 7 | 1.7% |
| 2.2 | Froyo | 8 | 4.0% |
| 2.3 - 2.3.2 | Gingerbread | 9 | 0.1% |
| 2.3.3 - 2.3.7 | | 10 | 39.7% |
| 3.2 | Honeycomb | 13 | 0.2% |
| 4.0.3 - 4.0.4 | Ice Cream Sandwich | 15 | 29.3% |
| 4.1.x | Jelly Bean | 16 | 23.0% |
| 4.2.x | | 17 | 2.0% |

*Data collected during a 14-day period ending on April 2, 2013.*
*Any versions with less than 0.1% distribution are not shown.*




Σχήμα 2.12: Ποσοστά των εγκατεστημένων εκδόσεων στις android συσκευές

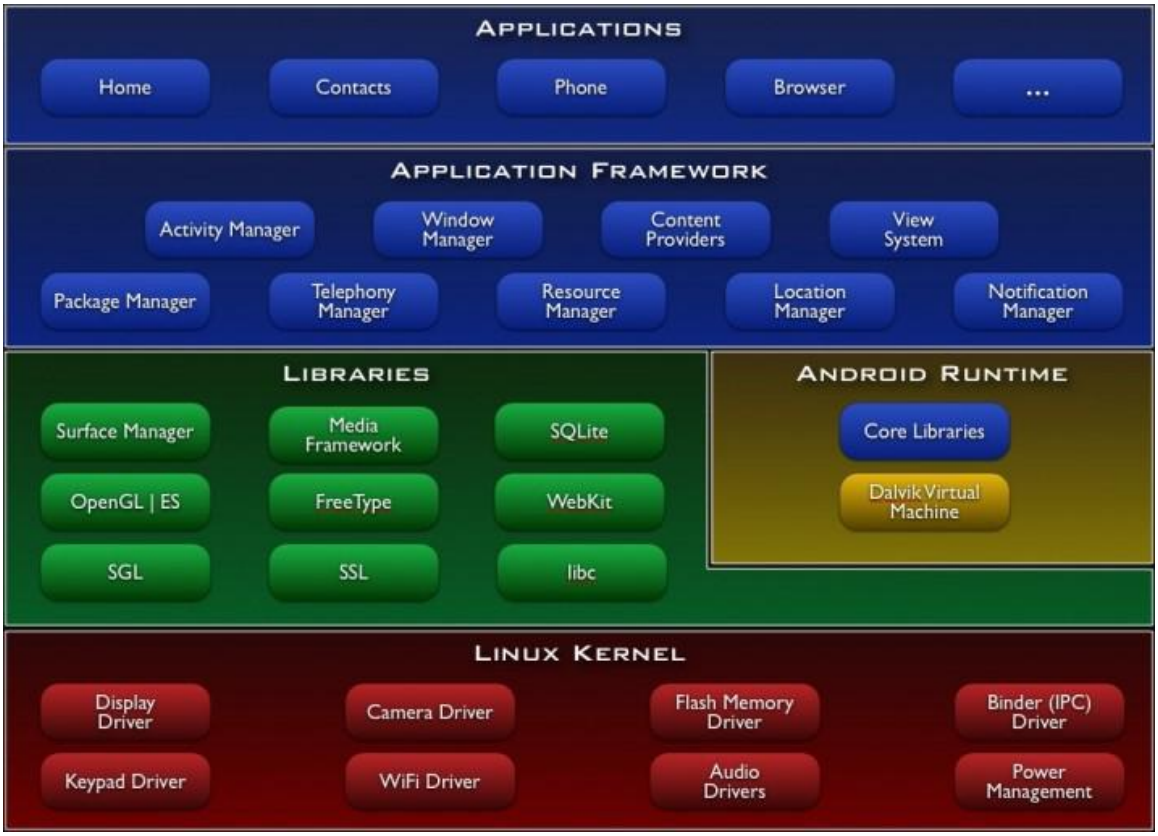
Όπως φαίνεται στην (Σχήμα 2.12) πιο πάνω, το Gingerbread είναι εγκατεστημένο στο μεγαλύτερο μέρος των συσκευών συγκεντρώνοντας το 39.7% αυτών. Το νεότερο από αυτό, Honeycomb ως μια έκδοση αποκλειστικά για ταμπλέτες και ουσιαστικά με μηδαμινό χρόνο κυκλοφορίας βρίσκεται στο 0.2%. Το Ice Cream Sandwich έχει φτάσει σε κυκλοφορία μέχρι και 29.3%, καθώς το νεότερο από όλα Jelly Bean παίρνει 23.0%, ένα ικανοποιητικό ποσοστό. Οι παλαιότερες εκδόσεις, Donut, Éclair και Froyo δεν έχουν μεγάλα ποσοστά.

Το Android έχει καταπληκτικά χαρακτηριστικά και πολλαπλές δυνατότητες, ενώ συνεχίζει να εκσυγχρονίζεται. Επιπρόσθετα παρέχει καταπληκτικά εργαλεία για την ανάπτυξη εφαρμογών που κάνουν την ζωή του κατασκευαστή πραγματικά πολύ πιο εύκολη. Για να κατανοήσουμε

την πρωτοτυπία αυτής της πλατφόρμας, θα δούμε παρακάτω την βασική της αρχιτεκτονική και θα συζητήσουμε γιατί το Android ευκολύνει την ζωή του προγραμματιστή ενώ ταυτόχρονα του παρέχει πάμπολλες επιλογές και δυνατότητες για την δημιουργία πρωτοποριακών εφαρμογών.

## 2.3 Αρχιτεκτονική του Android

Όπως προανέφερα, το Android είναι μια στοίβα λογισμικού. Η λογική πίσω από αυτήν την έκφραση και σε όλη την φιλοσοφία του Android, κρύβεται στο ακόλουθο διάγραμμα με τα βασικά συστατικά του (Σχήμα 2.13).




Σχήμα 2.13: Τα βασικά περιεχόμενα του λειτουργικού συστήματος Android

Στην στοίβα του Android (Σχήμα 2.13), παρατηρούμε 4 επίπεδα. Ακολούθως θα περιγράψουμε συνοπτικά τα βασικά αυτά επίπεδα χωρίς να μπούμε σε λεπτομέρειες για όλα τα περιεχόμενα του κάθε επιπέδου.

Κάθε επίπεδο στην αρχιτεκτονική αυτή, χρησιμοποιεί τις υπηρεσίες που του προσφέρονται από τα πιο κάτω επίπεδα. Ας δούμε τώρα αυτά τα επίπεδα ξεκινώντας από το πιο χαμηλό.

### 2.3.1 Πυρήνας Linux ( Linux Kernel)

Το Android είναι βασισμένο στα γερά θεμέλια του Linux. Ο πυρήνας Linux είναι δοκιμασμένος, σταθερός και πετυχημένος και μπορεί να βρεθεί παντού, από ρολόγια χειρός μέχρι υπερυπολογιστές. Το Linux παρέχει στο Android το αφαιρετικό επίπεδο υλικού, επιτρέποντάς του να μπορεί να χρησιμοποιηθεί σε μεγάλη ποικιλία πλατφόρμων στο μέλλον. Ειδικότερα, το Android χρησιμοποιεί τον πυρήνα Linux για την διαχείριση μνήμης, την διαχείριση διεργασιών, την δικτύωση και άλλες υπηρεσίες του λειτουργικού συστήματος.

### 2.3.2 Εγγενής Βιβλιοθήκες – Native Libraries

Στο αμέσως ψηλότερο επίπεδο βρίσκουμε τις Native Libraries – Εγγενής Βιβλιοθήκες. Όλες αυτές είναι γραμμένες στην γλώσσα προγραμματισμού C και C++ και μεταγλωττίστηκαν για την συγκεκριμένη αρχιτεκτονική υλικού που χρησιμοποιείται από το τηλέφωνο. Οι βιβλιοθήκες αυτές δεν είναι εφαρμογές που μπορούν να σταθούν από μόνες τους. Υπάρχουν για να μπορούν να κληθούν από προγράμματα υψηλότερου επιπέδου. Από την έκδοση Donut και μετά, οι κατασκευαστές μπορούν να γράφουν τις δικές τους τέτοιες βιβλιοθήκες με την χρήση της Εργαλειοθήκης NDK ( Native Development Kit ).

### 2.3.3 Χρόνος Εκτέλεσης – Android Runtime

Στο ίδιο επίπεδο με τις εγγενής βιβλιοθήκες, βρίσκουμε και τον χρόνο εκτέλεσης Android. Εδώ ζουν βασικές βιβλιοθήκες της Java και η εικονική μηχανή Dalvik. Η Dalvik είναι μια βελτιστοποιημένη υλοποίηση μιας εικονικής μηχανής Java για φορητές συσκευές από την Google. Η Dalvik τρέχει .dex αρχεία, τα οποία είναι bytecodes που προέρχονται από αρχεία .class και .jar. Εν αντιθέσει όμως με τα .class αρχεία, τα .dex είναι πολύ πιο συμπαγή και αποδοτικά, γεγονός σημαντικό για συσκευές με περιορισμένη μνήμη και μπαταρία.

Το Android περιλαμβάνει ένα σύνολο βασικών βιβλιοθηκών που παρέχουν τις περισσότερες από τις διαθέσιμες λειτουργίες των βασικών βιβλιοθηκών της Java. Κάποια πακέτα και κλάσεις υπάρχουν και στο Android κάποια άλλα δεν υποστηρίζονται καθόλου, ενώ ταυτόχρονα το Android παρέχει και επιπρόσθετα προσαρμοσμένα στις δικές του ανάγκες.

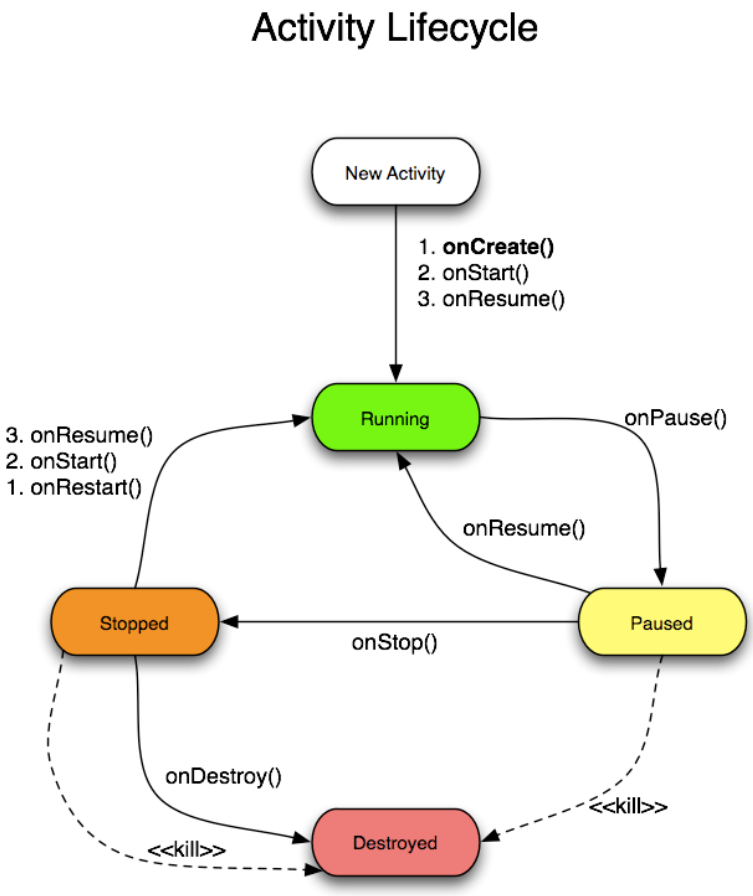
### 2.3.4 Πλαίσιο Εφαρμογής – Application Framework

Πάνω από τις εγγενής βιβλιοθήκες και το χρόνο εκτέλεσης Android, είναι το πλαίσιο εφαρμογής. Αυτό το επίπεδο μας παρέχει υψηλού επιπέδου δομικές μονάδες τις οποίες μπορούμε να χρησιμοποιούμε για την κατασκευή των εφαρμογών μας. Αυτό το πλαίσιο είναι προ-εγκατεστημένο στο Android, αλλά είναι επεκτάσιμο, αφού ο κάθε κατασκευαστής μπορεί να το συμπληρώσει με δικά του κομμάτια.

Τα σημαντικότερα δομικά στοιχεία του πλαισίου αυτού είναι:

- Διαχειριστής Δραστηριοτήτων - Activity Manager: Υπεύθυνο για τον έλεγχο του χρόνου ζωής (Σχήμα 2.14) των εφαρμογών και για την διατήρηση μιας στοίβας που επιτρέπει την πλοήγηση του χρήστη σε προηγούμενες οθόνες.

- Παροχέας Περιεχομένου - Content Providers: Αυτά τα αντικείμενα περιέχουν δεδομένα που μπορούν να διαμοιραστούν μεταξύ εφαρμογών.
- Διαχειριστής Πόρων - Resource Manager: Οι πόροι, είναι οτιδήποτε υπάρχει σε ένα πρόγραμμα και δεν είναι κώδικας. Για παράδειγμα μπορεί να είναι κωδικοί χρωμάτων, αλφαριθμητικοί χαρακτήρες ή ακόμα και έτοιμα σχεδιαγράμματα οθονών φτιαγμένα σε XML, τα οποία μπορεί το πρόγραμμα να καλεί.
- Διαχειριστής Τοποθεσίας - Location Manager: Χρησιμοποιείται για να μπορεί να ξέρει το τηλέφωνο που βρίσκεται ανά πάσα στιγμή.
- Διαχειριστής Κοινοποιήσεων - Notification Manager: Ιδανικός τρόπος για να ενημερώνεις τον χρήστη για γεγονότα που συμβαίνουν, διακριτικά χωρίς να διακόπτεις την εργασία του.




Σχήμα 2.14: Ο κύκλος ζωής μιας Δραστηριότητας Android

### 2.3.5 Εφαρμογές και Widgets

Στο υψηλότερο επίπεδο της στοίβας Android, φιγουράρουν οι εφαρμογές και τα widgets. Αυτό είναι που βλέπουν οι χρήστες χωρίς να γνωρίζουν την όλη από κάτω διαδικασία. Αυτές είναι εφαρμογές που γράφουν οι κατασκευαστές λογισμικού, εφαρμογές που ήδη είναι εγκατεστημένες στο τηλέφωνο ή που ο χρήστης παίρνει από το Android Market.

Οι εφαρμογές είναι προγράμματα που καταλαμβάνουν ολόκληρη την οθόνη και αλληλεπιδρούν με το χρήστη. Από την άλλη τα widget λειτουργούν σε μικρά τετράγωνα μέσα στην αρχική οθόνη – εφαρμογή.

## 2.4 Γιατί Android

Εν κατακλείδι παραθέτουμε εδώ κάποια σημεία και σκέψεις, για το Android, σαν μια τελευταία προσπάθεια να πείσουμε για την πρωτοπορία του στο χώρο και την ισχυρή δυναμική του. Γιατί Android λοιπόν;

- Είναι μια πραγματικά ανοιχτή, ελεύθερη πλατφόρμα ανάπτυξης, βασισμένη στο Linux

- Διαθέτει αρχιτεκτονική βασισμένη σε δομικά στοιχεία τα οποία μπορούν να τροποποιηθούν, να ολοκληρωθούν και να προσαρμοστούν στις ανάγκες κάθε κατασκευαστή και κατά συνέπεια χρήστη

- Πολλές ενσωματωμένες υπηρεσίες που μπορούν να κάνουν την εμπειρία του χρήστη μοναδική, όπως υπηρεσίες βασισμένες στην τοποθεσία, πανίσχυρη SQL βάση δεδομένων, μηχανή αναζήτησης και χάρτες

- Αυτόματη διαχείριση του κύκλου ζωής μιας εφαρμογής, με πολλαπλές δικλίδες ασφαλείας ανάμεσα στα προγράμματα. Βελτιστοποιήσεις στον τομέα διαχείρισης μνήμης και χαμηλής κατανάλωσης σε τέτοιο βαθμό που δεν έχει ξανασυναντηθεί σε άλλο έξυπνο κινητό τηλέφωνο (smartphone)

- Υψηλής ποιότητας γραφικά και ήχος

- Φορητότητα ανάμεσα σε ένα ευρύ φάσμα ήδη υπάρχοντος υλικού αλλά και μελλοντικού. Αυτό έρχεται σαν απόρροια του γεγονότος ότι όλα τα προγράμματα γράφονται σε Java και εκτελούνται από την εικονική μηχανή Dalvik. Επιπρόσθετα οι οθόνες μπορούν να τροποποιηθούν κατάλληλα για να υποστηρίζουν οποιαδήποτε ανάλυση, μέγεθος και προσανατολισμό οθόνης

Αυτοί είναι μερικοί μόνο από τους λόγους για τους οποίους έγινε η επιλογή του Android ως πλατφόρμα ανάπτυξης σε αυτή την πτυχιακή εργασία.

# Κεφάλαιο 3

## Προγραμματισμός σε περιβάλλον Android

---



---

### 3.1 Εισαγωγή

Το Android OS αποτελεί ένα λειτουργικό σύστημα το οποίο χρησιμοποείται τα τελευταία χρόνια σε ολοένα και αυξανόμενο αριθμό κινητών τηλεφώνων (smartphones), καθώς και σε ηλεκτρονικές ταμπλέτες (tablet pc's). Όλες οι λειτουργίες του κινητού τηλεφώνου ελέγχονται από εφαρμογές (applications) τις οποίες οι χρήστες μπορούν να κατεβάσουν και να εγκαταστήσουν ελεύθερα.

Αξίζει να σημειωθεί ότι το Android OS αποτελεί ελεύθερο λογισμικό ανοιχτού κώδικα (free and open source software), πράγμα που σημαίνει ότι οι κατασκευαστές κινητών τηλεφώνων ή άλλων ηλεκτρονικών συσκευών έχουν τη δυνατότητα να το χρησιμοποιήσουν στα προιόντα τους ελεύθερα και χωρίς κόστος. Το γεγονός αυτό αποτέλεσε τον καταλύτη στη διάδοση του λειτουργικού αυτού συστήματος. Παράλληλα τα εργαλεία για την κατασκευή εφαρμογών

υπάρχουν ελεύθερα στο διαδίκτυο, έτσι ώστε οποιοσδήποτε προγραμματιστής να έχει τη δυνατότητα να κατασκευάσει και να δημοσιεύσει εφαρμογές.

Ακολουθεί μία σύντομη περιγραφή στα εργαλεία αυτά και μία ανάλυση των βασικών στοιχείων που αφορούν στην ανάπτυξη εφαρμογών σε λειτουργικό σύστημα Android.

## 3.2 Προγραμματιστικά εργαλεία

Η γλώσσα προγραμματισμού που χρησιμοποιείται για την κατασκευή εφαρμογών Android είναι η Java, ενώ τον τελευταίο καιρό γίνονται προσπάθειες για να συμπεριληφθούν και άλλες γλώσσες όπως η C και C++. Οπότε βασική προϋπόθεση της κατασκευής είναι να διαθέτουμε τα αντίστοιχα εργαλεία της γλώσσας προγραμματισμού που θα χρησιμοποιήσουμε και συγκεκριμένα το Java Development Kit (JDK). Ακόμη, χρειαζόμαστε ένα ολοκληρωμένο περιβάλλον ανάπτυξης ( integrated development environment ή IDE) για να μεταγλωττίζουμε και να τρέχουμε τα προγράμματα μας, όπως το Eclipse (Εικόνα 3.1).

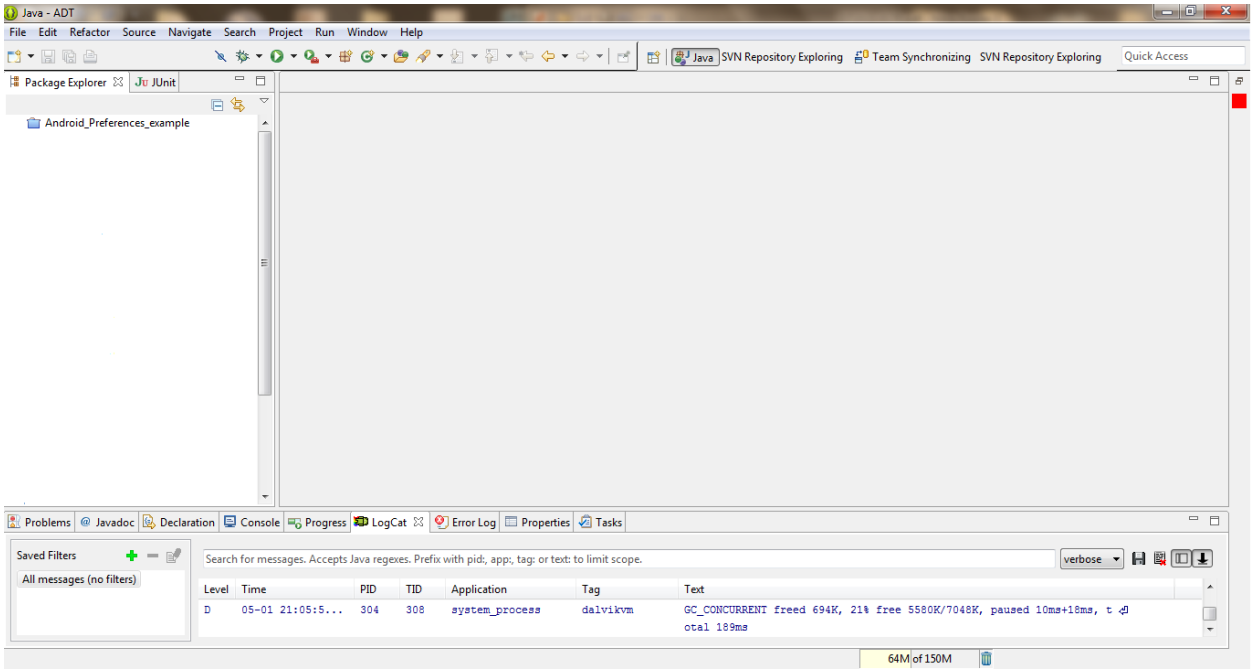
Το βασικότερο εργαλείο αποτελεί το Android SDK (software development kit) το οποίο ουσιαστικά μας παρέχει τα επιπλέον εργαλεία ώστε να μπορούμε να γράψουμε κώδικα συγκεκριμένα για την κατασκευή εφαρμογής Android.

Καθώς το λειτουργικό σύστημα Android κυκλοφορεί σε διάφορες εκδόσεις όπως είδαμε και αναλύσαμε σε προηγούμενο κεφάλαιο, καθίσταται σαφές ότι η κάθε έκδοση θα χρησιμοποιεί και ορισμένα διαφορετικά προγραμματιστικά εργαλεία. Έτσι μέσα από την ADT Plugin μπορούμε να εγκαταστήσουμε τα εργαλεία για την υλοποίηση εφαρμογής σε οποιαδήποτε έκδοση. Για παράδειγμα αν θέλουμε η εφαρμογή μας να είναι συμβατή με παλαιότερες εκδόσεις του Android, τότε θα πρέπει να εγκαταστήσουμε τα αντίστοιχα εργαλεία και να δοκιμάσουμε την εφαρμογή μας στις εκδόσεις αυτές.

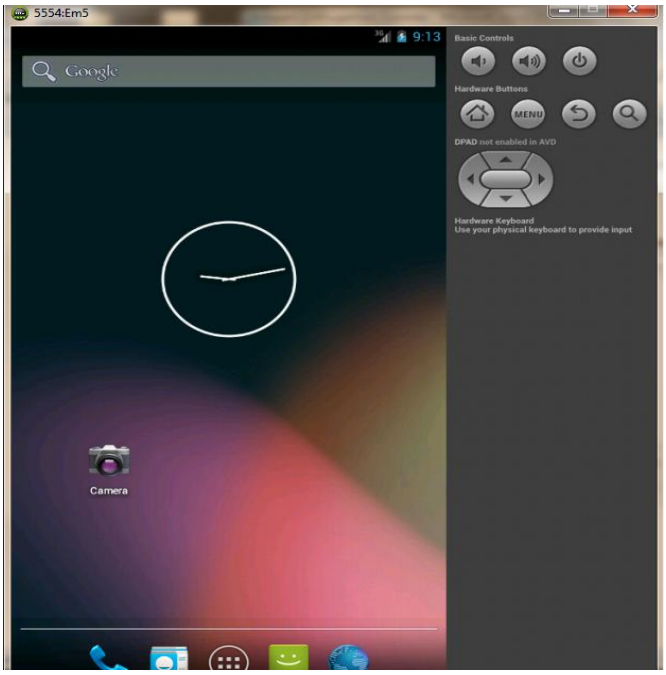
Τέλος, για να δοκιμάσουμε την εφαρμογή μας και να δούμε τα αποτελέσματα της θα χρειαστούμε κάποιον προσομοιωτή κινητού τηλεφώνου στον υπολογιστή μας. Τη λύση μας δίνει μία εικονική συσκευή Android (Android Virtual Device ή AVD) η οποία ουσιαστικά αποτελεί προσομοιωτή τόσο software όσο και hardware ενός κινητού τηλεφώνου με

λειτουργικό σύστημα Android (Εικόνα 3.2). Τη συσκευή αυτή, την εγκαθιστάμε μέσα από την ADT Plugin, από όπου μπορούμε και να ρυθμίσουμε πολλές παραμέτρους της, όπως την έκδοση Android που θα χρησιμοποιεί, το μέγεθος τηε οθόνης, το μέγεθος της κάρτας SD και της cache, καθώς και άλλα χαρακτηριστικά όπως δυνατότητα λήψης φωτογραφιών (κάμερας), δέκτη GPS. Απαιτούνται βέβαια αρκετοί υπολογιστικοί πόροι για την εκτέλεση της AVD, πράγμα που καθιστά τις περισσότερες φορές σχετικά αργή την εκτέλεση της εφαρμογής μας στη συσκευή αυτή. Φυσικά, σε κάθε περίπτωση μπορούμε να εγκαθιστάμε και να εκτελούμε τις εφαρμογές μας σε φυσική συσκευή, όπως κινητό τηλέφωνο ή tablet pc, που χρησιμοποιεί το λειτουργικό σύστημα Android.

Στο σημείο αυτό, πρέπει να σημειώσουμε ότι όλα τα παραπάνω προγραμματιστικά εργαλεία διατίθενται ελεύθερα στο διαδίκτυο από τους κατασκευαστές τους, τα οποία μπορούμε να κατεβάσουμε και να χρησιμοποιήσουμε χωρίς κόστος.




Εικόνα 3.1: Το περιβάλλον ανάπτυξης λογισμικού Eclipse




Εικόνα 3.2: Εικονική συσκευή με λειτουργικό σύστημα Android (AVD)

## 3.3 Συστατικά στοιχεία εφαρμογής

Με τον όρο αυτό, εννοούμε τα απαραίτητα δομικά στοιχεία μίας εφαρμογής Android. Το κάθε στοιχείο από αυτά είναι ουσιαστικά ένας τρόπος πρόσβασης του λειτουργικού συστήματος στην εφαρμογή μας.

Μπορούμε να τα διακρίνουμε σε τέσσερα βασικά στοιχεία: τις δραστηριότητες (activities) , τις υπηρεσίες (services), τους παρόχους περιεχομένου (content providers) και τους καθολικούς παραλήπτες μηνυμάτων (broadcast receivers).

### 3.3.1 Activities

Αποτελούν το βασικότερο στοιχείο κάθε εφαρμογής. Η κάθε activity αποτελεί μία διαφορετική οθόνη (παράθυρο) της εφαρμογής μας, με την οποία ο χρήστης αλληλεπιδρά. Σε αυτήν δηλαδή φορτώνεται το γραφικό περιβάλλον το οποίο τελικά βλέπει ο χρήστης.

Κάθε εφαρμογή αποτελείται συνήθως από αρκετές activities , οι οποίες συνδέονται μεταξύ τους, δηλαδή κάθε activity μπορεί να καλέσει κάποια άλλη. Η πρώτη οθόνη που θα δει ο χρήστης όταν τρέξει την εφαρμογή αποτελεί την κεντρική (main) activity, από την οποία προέρχονται όλες οι υπόλοιπες. Τη στιγμή που η εφαρμογή μας καλεί νέα activity, ο χρήστης βλέπει τη νέα οθόνη και με το πλήκτρο back μπορεί να επιστρέψει στην προηγούμενη, δηλαδή έχουμε μία μορφή στοίβας (LIFO). Αν δηλαδή η activity A καλέσει την activity B και αυτή την activity C (A->B->C) τότε στην activity C, με το πλήκτρο back επιστέφουμε στην B και από αυτήν επιστρέφουμε στην A. Φυσικά κάποιες activities μπορούμε να επιλέξουμε να μην τις αποθηκεύουμε στη στοίβα, δηλαδή αν στην περίπτωση μας επιλέξουμε η B να μην αποθηκευτεί στη στοίβα, τότε με το πλήκτρο back από την C θα επιστρέψουμε στην A. Αν πατηθεί το πλήκτρο back στη main activity τότε βγαίνουμε από την εφαρμογή μας.

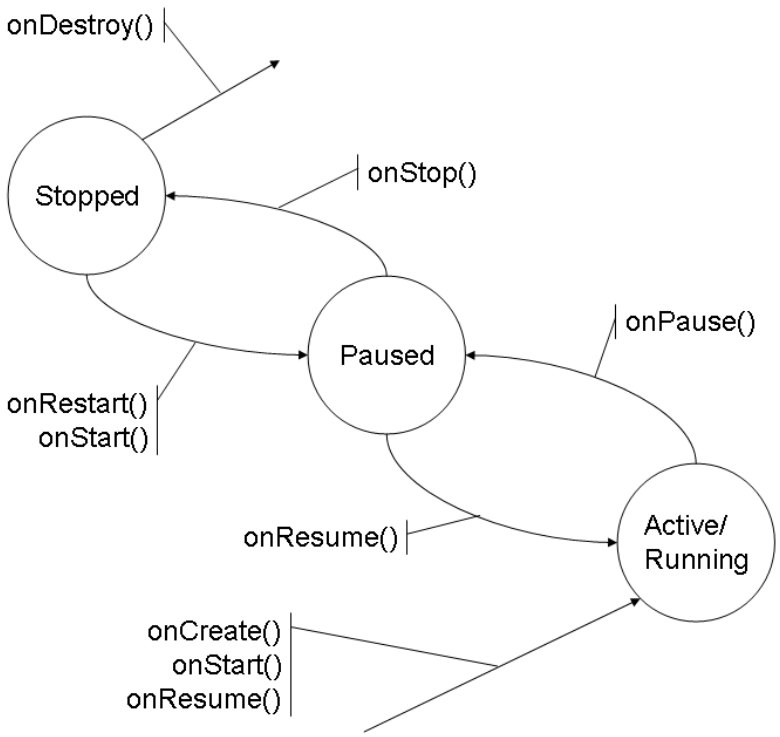
Θεωρούμε το παράδειγμα μιας απλής εφαρμογής ηλεκτρονικού ταχυδρομείου, η οποία μόλις ανοίγει μας εμφανίζει 3 επιλογές ή κουμπιά: Inbox, Compose, Sent Mail. Η συνηθέστερη υλοποίηση της εφαρμογής αυτής είναι να δημιουργήσουμε 5 διαφορετικές οθόνες, δηλαδή 5 activities: 1. Η main activity που εμφανίζει τις επιλογές, 2. Inbox, δηλαδή η λίστα με τα εισερχόμενα μηνύματα, 3. ReadMail, δηλαδή η οθόνη που διαβάζουμε ένα email, 4. Compose, δηλαδή η οθόνη που γράφουμε ένα email για να το στείλουμε και 5. SentMail, δηλαδή η λίστα με τα εξερχόμενα μηνύματα μας.

Κάθε activity πρέπει να υλοποιεί τη μέθοδο onCreate() η οποία καλείται τη στιγμή που ξεκινάει η activity. Η μέθοδος αυτή καλείται μόνο όταν τρέξουμε μία νέα activity είτε από κάποιο κουμπί είτε από κάποια επιλογή, δηλαδή δεν καλείται αν επιστρέψουμε με το πλήκτρο back σε κάποια προηγούμενη. Όταν επιστρέφουμε σε κάποια προηγούμενη activity βλέπουμε το στιγμιότυπο που αφήσαμε και η activity δεν ξαναδημιουργείται. Επίσης μέσα στη μέθοδο αυτή (συνήθως στην αρχή) πρέπει να δημιουργηθεί η διεπαφή χρήστη (User Interface ή UI), η οποία δηλώνεται με την εντολή setContentView(). (Το UI περιγράφεται στην επόμενη παράγραφο).

Για να ξεκινήσουμε μία activity B μέσα από μία activity A χρησιμοποιούμε τις παρακάτω εντολές: - Intent intent = new Intent (this, MyActivityB.class);

- startActivity(intent);

Για να τερματίσουμε μία activity, αρκεί η εντολή finish(). Σε γενικές γραμμές πάντως, το Android διαχειρίζεται τον κύκλο ζωής της κάθε activity, οπότε δε χρειάζεται να τις τερματίζουμε εμείς, παρά μόνο όταν θέλουμε οπωσδήποτε να μη μπορεί να επιστρέψει ο χρήστης σε αυτό το στιγμιότυπο της activity. Ο κύκλος ζωής και οι αντίστοιχες μέθοδοι των activities φαίνονται στην εικόνα 3.3.




Εικόνα 3.3: Ο κύκλος ζωής των activities και οι αντίστοιχες μέθοδοι

### 3.3.2 Services

Μία service αποτελεί ένα στοιχείο της εφαρμογής, το οποίο τρέχει στο παρασκήνιο (background) και μπορεί να εκτελέσει αρκετές και χρονοβόρες λειτουργίες. Ειδοποιός διαφορά με την activity είναι το ότι η service δεν παρέχει user interface. Επίσης, η service μπορεί να εκτελείται ακόμα και όταν τρέξουμε κάποια διαφορετική εφαρμογή από αυτήν που την ξεκίνησε.

Παράδειγμα υπηρεσίας αποτελεί η εφαρμογή για αναπαραγωγή μουσικής. Όταν ακούμε μουσική, είναι πιθανό να θέλουμε να συνεχίσουμε να την ακούμε ακόμα και αν τρέξουμε κάποια άλλη εφαρμογή και δίχως να απαιτούμε γραφικό περιβάλλον για την αναπαραγωγή. Επιπλέον όλη η επικοινωνία με το δίκτυο θέλουμε να συνεχίσει όταν ανοίξουμε κάποια εφαρμογή και να μην σταματήσει, άρα θα πρέπει να λειτουργεί σαν service.

Δύο βασικές λειτουργίες σχετίζονται με τις υπηρεσίες: η startService() και η bindService(). Η πρώτη καλείται από κάποια activity για να ξεκινήσει η service και θα συνεχίσει να εκτελείται ακόμα και αν η activity που την κάλεσε τερματιστεί. Για το λόγο αυτό η service δεν επιστρέφει κάποιο αποτέλεσμα στην activity που την κάλεσε, αλλά εκτελεί κάποια λειτουργία όπως το κατέβασμα κάποιου αρχείου από το διαδίκτυο. Η δεύτερη λειτουργία «δένει» τη service με κάποιο άλλο στοιχείο της εφαρμογής, π.χ. αποστολή και λήψη αποτελεσμάτων. Στην περίπτωση αυτή, αν τερματιστεί το στοιχείο που έχει «δεθεί» με τη service, τότε τερματίζεται και η service.

Προηγουμένως, αναφέραμε ότι η service δεν παρέχει διεπαφή χρήστη, επομένως δημιουργείται το ερώτημα, πώς θα ενημερώνεται ο χρήστης για κάποιο αποτέλεσμα της υπηρεσίας. Υπάρχουν δύο τρόποι: η ειδοποίηση Toast και η ειδοποίηση μέσω της Status Bar. Οι ειδοποιήσεις αυτές αναλύονται στην παράγραφο 3.4.4, καθώς αφορούν στο user interface.

Σε αντίθεση με τις activities, το λειτουργικό σύστημα δε διαχειρίζεται τον κύκλο ζωής των services (παρά μόνο όταν υπάρχει έλλειψη υπολογιστικών πόρων), επομένως η κάθε service θα πρέπει να δηλώνει το πότε θα σταματήσει με την εντολή stopSelf().

### 3.3.3 Content Providers

Οι content providers διαχειρίζονται αποθηκευτικούς χώρους για δεδομένα, οι οποίοι είναι προσπελάσιμοι από οποιαδήποτε εφαρμογή. Αποτελούν το μοναδικό τρόπο για αποθήκευση δεδομένων, τα οποία θέλουμε να είναι προσβάσιμα από αρκετές εφαρμογές. Για παράδειγμα μία εφαρμογή τηλεφωνικού καταλόγου, θα θέλαμε να αποθηκεύει σε συγκεκριμένο χώρο τις επαφές μας, ώστε αργότερα αν θελήσουμε να χρησιμοποιήσουμε διαφορετική εφαρμογή, να μπορεί αυτή να εντοπίσει και να επεξεργαστεί τα ήδη υπάρχοντα δεδομένα. Παρόμοια παραδείγματα αποτελούν τα αρχεία ήχου, βίντεο, οι εικόνες, τα οποία θέλουμε να αποθηκεύονται σε κάποιους προεπιλεγμένους χώρους, ώστε να είναι προσβάσιμα από περισσότερες από μία εφαρμογές.

Στο πλαίσιο αυτό υπάρχει ενσωματωμένη βάση δεδομένων στο λειτουργικό σύστημα Android (SQLite database), στην οποία μπορούν να αποθηκευτούν ή να διαβάζουν δεδομένα οι content providers.

### 3.3.4 Broadcast Receivers

Οι broadcast receivers ενημερώνονται για κάποιο συγκεκριμένο γεγονός από το λειτουργικό σύστημα και τότε ενεργοποιούνται. Τέτοια γεγονότα μπορεί να είναι ότι η οθόνη έχει σβήσει, ότι η στάθμη της μπαταρίας είναι χαμηλή και άλλα. Για παράδειγμα, αρκετές εφαρμογές λήψης φωτογραφιών απενεργοποιούν το flash σε περίπτωση χαμηλής μπαταρίας για λόγους εξοικονόμησης ενέργειας. Ενημερώνονται δηλαδή από τον broadcast receiver για το γεγονός αυτό και πράττουν ανάλογα. Οι broadcast receivers δεν παρέχουν user interface, αλλά μπορούν να ενημερώσουν το χρήστη για κάποιο γεγονός μέσω των status bar notifications.

## 3.4 User Interface

Η διεπαφή χρήστη έχει τεράστια σημασία για κάθε εφαρμογή. Αποτελεί την τελική εικόνα που βλέπει ο χρήστης, το γραφικό περιβάλλον στο οποίο θα περιηγείται, και ενεργοποιεί όλες τις λειτουργίες της εφαρμογής. Το user interface και η λειτουργικότητα της εφαρμογής αποτελούν αλληλένδετα στοιχεία και χωρίς το ένα δε μπορεί να υπάρξει και το άλλο. Πολλές φορές μάλιστα είναι δυσκολότερος ο σχεδιασμός ενός όμορφου και εύχρηστου περιβάλλοντος εργασίας, παρά η ίδια η λειτουργικότητα της εφαρμογής. Καθίσταται σαφές, λοιπόν, ότι απαιτεί μεγάλη προσπάθεια και προσοχή η δημιουργία ενός γραφικού περιβάλλοντος που θα προσελκύει τους χρήστες και θα τους ωθεί να χρησιμοποιούν μία συγκεκριμένη εφαρμογή έναντι μίας άλλης, με τις ίδιες λειτουργίες.

### 3.4.1 Layout

Σε κάθε οθόνη της εφαρμογής, πρωταρχικό στοιχείο του γραφικού περιβάλλοντος αποτελεί η διάταξη των γραφικών στοιχείων ή layout. Το layout περιλαμβάνει όλα τα γραφικά στοιχεία της οθόνης, τα οποία μπορεί να είναι διατεταγμένα σε επιμέρους layouts.

Υπάρχουν 4 είδη layout, τα LinearLayout (γραμμική διάταξη) , RelativeLayout (σχετική διάταξη), FrameLayout (διάταξη πλαισίου) και TableLayout (διάταξη πίνακα). (Υπάρχει και το AbsoluteLayout, το οποίο όμως έχει προταθεί να μην χρησιμοποιείται πλέον, διότι ορίζει τις απόλυτες θέσεις κάθε στοιχείου, οι οποίες όμως διαφέρουν ανάλογα με την κινητή συσκευη και την οθόνη που χρησιμοποιείται η εφαρμογή). Το LinearLayout αποτελεί διάταξη στοιχείων σε οριζόντια ή κατακόρυφη σειρά. Αν δηλαδή δηλώσουμε τρία στοιχεία A, B, C μέσα σε ένα οριζόντιο LinearLayout τότε τα στοιχεία αυτά θα εμφανίζονται στην οθόνη σε μία οριζόντια διάταξη με τη σειρά που τα δηλώσαμε το ένα δίπλα στο άλλο. Το RelativeLayout μας δίνει περισσότερη ελευθερία στη δήλωση των γραφικών στοιχείων, με την έννοια ότι κάθε στοιχείο μπορούμε να επιλέξουμε να το εμφανίσουμε σε συγκεκριμένο σημείο της οθόνης, όπως π.χ. στην αρχή, στο κέντρο, στο τέλος ή να επιλέξουμε τη θέση του σε σχέση με κάποιο άλλο

στοιχείο. Το FrameLayout αποτελεί την απλούστερη διάταξη στοιχείων. Είναι απλά ένας κενός χώρος τον οποίο μπορούμε να γεμίσουμε με κάποιο αντικείμενο, π.χ. μία εικόνα. Για το λόγο αυτό συνήθως χρησιμοποιείται σαν ρίζα (root) στο δέντρο των γραφικών στοιχείων της οθόνης. Τέλος, το TableLayout όπως φανερώνει και η ονομασία του αποτελεί διάταξη πίνακα, δηλαδή μπορεί να διατάσσει τα παιδιά του (children) σε σειρές και στήλες. Σε όλα τα παραπάνω στοιχεία μπορούμε να ρυθμίσουμε αρκετές παραμέτρους όπως μέγεθος (πλάτος και ύψος), οριζόντια ή κατακότυφη διάταξη, βαρύτητα (layout gravity) και άλλες.

Υπάρχουν δύο τρόποι για να δηλώσουμε τα layouts (όπως και κάθε άλλο γραφικό στοιχείο) της εφαρμογής μας : α) μέσω αρχείων xml ή β) μέσα στις activities της εφαρμογής μας. Τα αρχεία xml αποτελούν στατικό τρόπο δημιουργίας των γραφικών στοιχείων. Τα αποθηκεύουμε σε συγκεκριμένο φάκελο του project μας και τα καλούμε για δημιουργία του γραφικού περιβάλλοντος μέσα από τις activities. Παρουσιάζουν δενδρική δομή για εύκολη συγγραφή και κατανόηση του περιεχομένου τους. Πολλές φορές ωστόσο δε γνωρίζουμε εξαρχής τη διάταξη που θα χρησιμοποιήσουμε, διότι ίσως να εξαρτάται από ορισμένες επιλογές του χρήστη. Στις περιπτώσεις αυτές δημιουργούμε δυναμικά τα layouts μέσα στις activities και ορίζουμε εκεί τις παραμέτρους αυτών. Ωστόσο ο πιο εύκολος και συνηθέστερος τρόπος είναι να δημιουργήσουμε για κάθε οθόνη ένα διαφορετικό xml αρχείο που θα περιλαμβάνει τη διάταξη όλων των γραφικών της στοιχείων, και να το καλέσουμε μέσα από την αντίστοιχη activity.

Ακολουθούν δύο παραδείγματα διάταξης των γραφικών στοιχείων της οθόνης.

Παράδειγμα 1:

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"

xmlns:android="http://schemas.android.com/apk/res/android"
   >
```

```
    <TextView
        android:layout_width="105px"
        android:layout_height="wrap_content"
        android:text="Hello World,\nUIActivity"
        />
    <Button
        android:layout_width="100px"
        android:layout_height="wrap_content"
        android:text="Button"
        />
</LinearLayout>
```


Εικόνα 3.4: Παράδειγμα κατασκευής ενός LinearLayout που περιλαμβάνει ένα κείμενο και ένα κουμπί

Παράδειγμα 2:

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_height="fill_parent"
android:layout_width="fill_parent"
android:background="#000044">
<TableRow>
  <TextView
    android:text="User Name:"
    android:width ="120px"
    />
  <EditText
    android:id="@+id/txtUserName"
    android:width="200px" />
</TableRow>
<TableRow>
  <TextView
    android:text="Password:"
    />
  <EditText
    android:id="@+id/txtPassword"
    android:password="true"

    />
</TableRow>
<TableRow>
  <TextView />
  <CheckBox android:id="@+id/chkRememberPassword"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Remember Password"
    />
```

```
    </TableRow>
    <TableRow>
      <Button
        android:id="@+id/buttonSignIn"
        android:text="Log In" />
    </TableRow>
</TableLayout>
```


Εικόνα 3.5: Παράδειγμα κατασκευής ενός TableLayout το οποίο ζητάει όνομα χρήστη και κωδικό πρόσβασης

### 3.4.2 Μενού

Τα μενού αποτελούν ένα σημαντικό κομμάτι της διεπαφής χρήστη για κάθε οθόνη της εφαρμογής, διότι παρέχουν στο χρήστη ένα γνωστό και φιλικό τρόπο για να εισάγει τις επιλογές του. Στο λειτουργικό σύστημα Android υπάρχουν τρία διαφορετικά είδη μενού, το μενού επιλογών (options menu),το μενού πλαισίου (context menu) και το υπομενού (submenu), τα οποία δηλώνονται και αυτά σε αρχεία xml.

### 3.4.2.1 Μενού επιλογών

Το μενού επιλογών αποτελεί το βασικότερο μενού μίας εφαρμογής. Εμφανίζεται τη στιγμή που πατάμε το κουμπί menu της συσκευής μας και περιέχει όλες τις βασικές επιλογές της εφαρμογής μας. Στον κώδικα της εφαρμογής μας ορίζουμε κάθε επιλογή του μενού σε ποια activity θα οδηγήσει το χρήστη. Παράδειγμα ενός μενού επιλογών φαίνεται στην παρακάτω εικόνα:


Εικόνα 3.6: Το μενού επιλογών μίας εφαρμογής περιήγησης στο διαδίκτυο (browser)

### 3.4.2.2 Μενού πλαισίου

Γνωρίζουμε όλοι τις επιλογές που μας δίνει το δεξί κλικ σε διάφορα προγράμματα ή εικονίδια στα λειτουργικά συστήματα που χρησιμοποιούνται στους υπολογιστές. Το ζήτημα είναι πώς θα εμφανίσουμε παρόμοιες επιλογές στις εφαρμογές Android, αφού διαθέτουμε οθόνη αφής. Τη λύση έρχεται να μας δώσει το μενού πλαισίου το οποίο περιλαμβάνει αυτές ακριβώς τις επιλογές σε οποιοδήποτε γραφικό στοιχείο το ορίσουμε (εικόνα, κείμενο κλπ) και

ενεργοποιείται από τον χρήστη με παρατεταμένο πάτημα (press and hold ή long press) του στοιχείου αυτού. Για παράδειγμα στο μενού πλαισίου ενός κειμένου θα ορίζαμε επιλογές «αντιγραφή», «αποκοπή», στο μενού πλαισίου μίας διεύθυνσης URL σε εφαρμογή web browser θα ορίζαμε επιλογές «άνοιγμα», «άνοιγμα σε νέα καρτέλα».

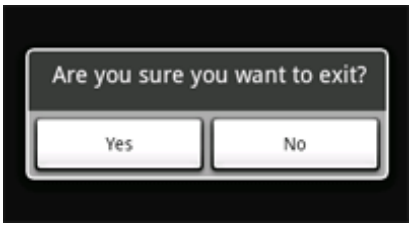
#### 3.4.2.3 Υπομενού

Το υπομενού μπορεί να προστεθεί σαν επιλογή στα δύο παραπάνω μενού και όπως δείχνει και το όνομα του ανοίγει ένα επιπλέον μενού για περισσότερες επιλογές. Χρησιμοποιείται σε περιπτώσεις που η εφαρμογή μας εκτελεί αρκετές λειτουργίες και θέλουμε να τις οργανώσουμε σε μενού.

### 3.4.3 Διάλογος (dialog)

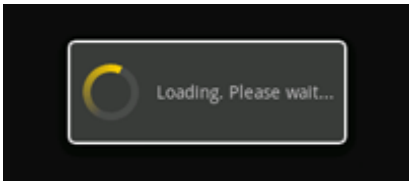
Ο διάλογος (dialog) είναι συνήθως ένα μικρό παράθυρο που εμφανίζεται στην οθόνη μπροστά από την activity που τον κάλεσε. Η activity αυτή χάνει την εστίαση που είχε (focus) και το παράθυρο του διαλόγου είναι το μοναδικό με το οποίο μπορεί να αλληλεπιδράσει ο χρήστης. Χρησιμοποιείται είτε για ενημέρωση του χρήστη για κάποιο γεγονός είτε γα να ορίσει ο χρήστης κάποια επιλογή του. Τα κυριότερα είδη διαλόγων είναι ο AlertDialog (διάλογος ειδοποίησης) και ο ProgressDialog (διάλογος προόδου).

Ο AlertDialog είναι ο πιο συνηθισμένος διάλογος. Αποτελείται από ένα τίτλο, ένα μήνυμα, ορισμένα κουμπιά ή μια λίστα από επιλογές. Για κάθε κουμπί του διαλόγου, ορίζουμε μέσα στην activity τις ενέργειες που θα ακολουθήσουν όταν το πατήσει ο χρήστης.




Εικόνα 3.7: Παράδειγμα AlertDialog

Ο ProgressDialog αποτελεί ουσιαστικά επέκταση του AlertDialog και χρησιμοποιείται όταν θέλουμε να εμφανίσουμε στο χρήστη την πρόοδο για κάποια ενέργεια. Για παράδειγμα, όταν θέλουμε να κατεβάσουμε κάποιες εικόνες από το διαδίκτυο και να τις εμφανίσουμε στο χρήστη, θα χρειαστούμε κάποιο χρονικό διάστημα για να ολοκληρωθεί αυτή η ενέργεια. Οπότε για να μη βλέπει ο χρήστης μία κενή μαύρη οθόνη, εμφανίζουμε έναν ProgressDialog και στο background εκτελούμε τις χρονοβόρες διαδικασίες. Παρακάτω φαίνεται η απλούστερη μορφή ενός ProgressDialog.




Εικόνα 3.8: Παράδειγμα ProgressDialog

### 3.4.4 Ειδοποιώντας το χρήστη

Σε πολλές περιπτώσεις θέλουμε να ενημερώσουμε το χρήστη για κάποιο γεγονός ή αποτέλεσμα σχετικό με την εφαρμογή μας. Ορισμένα από αυτλα τα γεγονότα απαιτούν κάποια απάντηση από το χρήστη και άλλα όχι. Για παράδειγμα, όταν ο χρήστης αποθηκεύει ένα αρχείο, θα θέλαμε να δει κάποιο μήνυμα ότι το αρχείο αποθηκεύτηκε επιτυχώς. Ή όταν η εφαρμογή μας τρέχει στο background και θέλει να ενημερώσει το χρήστη για κάποιο γεγονός, θα πρέπει να στείλει κάποια ειδοποίηση την οποία ο χρήστης να μπορεί να «ανοίξει» όταν αυτός επιθυμεί. Στην πρώτη περίπτωση χρησιμοποιούμε toast notification, ενώ στη δεύτερη status bar notification.

#### 3.4.4.1 Toast Notification

Η toast notification είναι ένα μήνυμα που εμφανίζεται για λίγα δευτερόλεπτα στο παράθυρο που βρίσκεται ο χρήστης, οποιασδήποτε εφαρμογής και αν είναι αυτό. Ο χώρος που καταλαμβάνει είναι ο ελάχιστος απαιτούμενος ώστε το μήνυμα να είναι εμφανές, ενώ ο χρήστης μπορεί όσο εμφανίζεται το μήνυμα να αλληλεπιδρά με την activity στην οποία βρίσκεται. Δεν υπάρχει κάποια επιλογή σε αυτή την ειδοποίηση, παρά μόνο ενημέρωση, δηλαδή ο χρήστης δε μπορεί να αλληλεπιδράσει με την ειδοποίηση. Χρησιμοποιείται συνήθως για μικρά μηνύματα που δεν απαιτούν κάποια ενέργεια από το χρήστη.

#### 3.4.4.2 Status Bar Notification

Η status bar notification, όπως φανερώνει και το όνομα της, είναι μία ειδοποίηση η οποία εμφανίζεται στη status bar του κινητού τηλεφώνου μας, και την οποία μπορούμε να ανοίξουμε είτε βρισκόμαστε στο κεντρικό μενού του τηλεφώνου μας, είτε σε κάποια εφαρμογή. Αντίθετα με την toast, η status bar notification μπορεί να επιλεχθεί και να ξεκινήσει κάποια λειτουργία ανάλογα με τις ενέργειες που έχουμε ορίσει στον κώδικα της εφαρμογής. Για παράδειγμα, όταν κατεβάζουμε ένα αρχείο από το διαδίκτυο, όταν η λήψη ολοκληρωθεί, θα θέλαμε να επιλέξουμε την ειδοποίηση αυτή και με τον τρόπο αυτό είτε να ανοίξουμε το φάκελο που βρίσκεται το αρχείο, είτε να το τρέξουμε. Τις περισσότερες φορές οι toast notifications ενεργοποιούνται από activities, ενώ οι status bar notifications από services.

### 3.5 Application Resources & Συμβατότητα

Με τον όρο application resources εννοούμε όλα τα στατικά στοιχεία τα οποία χρησιμοποιεί η εφαρμογή μας, όπως για παράδειγμα οι εικόνες και τα strings που χρησιμοποιούμε στον κώδικα. Όλα αυτά τα στοιχεία θα πρέπει να είναι ανεξάρτητα από τον κώδικα της εφαρμογής, δηλαδή να δηλώνονται σε διαφορετικά σημεία από τις κλάσεις μας, για λόγους συμβατότητας

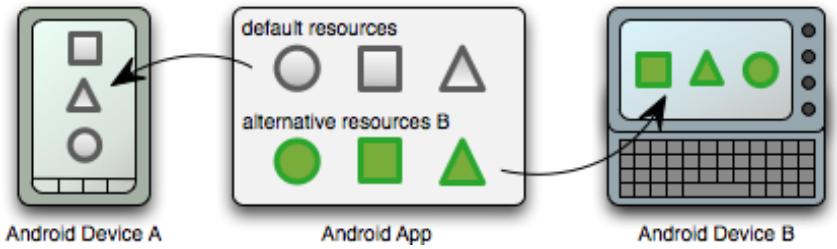
της εφαρμογής μας. Δηλαδή αν η εφαρμογή μας εμφανίζει σε μία οθόνη κάποιες εικόνες, θα θέλαμε ίσως να τις εμφανίσουμε με διαφορετικό τρόπο όταν βρισκόμαστε σε portrait mode και με διαφορετικό όταν βρισκόμαστε σε landscape. Επίσης, για να κάνουμε την εφαρμογή να υποστηρίζει αρκετές γλώσσες θα μπορούσαμε να κρατάμε σε διαφορετικό αρχείο τα strings για κάθε γλώσσα και να τα προσπελαύνουμε με τον ίδιο τρόπο από τον κώδικα της εφαρμογής μας. Έτσι λοιπόν, για κάθε στοιχείο που θα χρησιμοποιήσουμε ορίζουμε κάθε φορά το προεπιλεγμένο (default) και έπειτα αν θέλουμε και άλλα εναλλακτικά (alternative), τα οποία μπορεί να εξαρτώνται από τον προσανατολισμό (orientation) της συσκευής, το μέγεθος της οθόνης, τη γλώσσα που χρησιμοποιεί ο χρήστης στη συσκευή και άλλα.

### 3.5.1 Προσανατολισμός

Οι συσκευές με λειτουργικό σύστημα Android μπορούν να τοποθετηθούν είτε σε προσανατολισμό πορτρέτου (portrait mode) είτε σε προσανατολισμό τοπίου (landscape mode). Η πρώτη περίπτωση αφορά τον πιο συνηθισμένο τρόπο χρήσης του κινητού, αυτόν που χρησιμοποιούμε το κινητό μας τηλέφωνο σε όρθια θέση, δηλαδή η μεγαλύτερη πλευρά της οθόνης είναι κατακόρυφη. Η δεύτερη περίπτωση είναι η αντίστροφη, δηλαδή όταν η μεγαλύτερη πλευρά της οθόνης είναι οριζόντια. Αντιλαμβάνεται κανείς ότι τα γραφικά στοιχεία της οθόνης θα θέλαμε κάποιες φορές να εμφανίζονται με διαφορετικό τρόπο, ανάλογα τον προσανατολισμό της συσκευής. Το λειτουργικό σύστημα Android έχει τη δυνατότητα να επανασχεδιάζει τα γραφικά αυτά στοιχεία, όταν αλλάζουμε προσανατολισμό, και να τα εμφανίζει με τον τρόπο που λεχουμε ορίσει εμείς. Ο προεπιλεγμένος τρόπος είναι να παρουσιάζεται το user interface με την ίδια διάταξη και στους δύο προσανατολισμούς. Έστω λοιπόν ότι θέλουμε να εμφανίσουμε τρεις εικόνες σε κάποια οθόνη της εφαρμογής μας, οι οποίες να καλύπτουν όλο το χώρο της οθόνης. Η καλύτερη λύση θα ήταν όταν ο χρήστης χρησιμοποιεί το κινητό του σε portrait mode, να εμφανίζονται η μία κάτω από την άλλη, ενώ όταν αλλάζει σε landscape mode, να εμφανίζονται η μία δίπλα στην άλλη. Στις παρακάτω εικόνες φαίνεται η διάταξη για το παραπάνω παράδειγμα σε portrait mode και σε landscape mode. Στην πρώτη εικόνα έχουμε ορίσει μόνο μία διάταξη, η οποία χρησιμοποιείται και για

τους δύο προσανατολισμούς, ενώ στη δεύτερη εικόνα έχουμε ορίσει διαφορετική διάταξη για την περίπτωση του landscape mode, η οποία οπτικά είναι και καλύτερη.


Εικόνα 3.8: Κοινή διάταξη για portrait και landscape προσανατολισμό




Εικόνα 3.9: Διαφορετική διάταξη για portrait και landscape προσανατολισμό

Προηγουμένως στο user interface αναφέραμε ότι η διάταξη των γραφικών στοιχείων δηλώνεται σε αρχεία xml. Συγκεκριμένα στο project της εφαρμογής μας υπάρχει ο φάκελος res/layout/ στον οποίο τοποθετούμε όλα τα αρχεία xml που αφορούν στη διεπαφή χρήστη της εφαρμογής μας (το res προέρχερται από το resources). Έστω ότι η εφαρμογή μας περιέχει τρεις διαφορετικές οθόνες (δηλαδή τρεις διαφορετικές activities), με κάθε μία να χρησιμοποιεί το δικό της user interface. Τοποθετούμε λοιπόν τα αρχεία myfirstscreen.xml,

mysecondscreen.xml και mythirdscreen.xml στον παραπάνω φάκελο. Όμως η Τρίτη οθόνη της εφαρμογής μας εμφανίζει τρεις εικόνες όπως το παράδειγμα παραπάνω, άρα θέλουμε να χρησιμοποιήσουμε διαφορετική διάταξη για το landscape mode της τρίτης οθόνης. Όλα τα αρχεία xml που αφορούν στο user interface και συγκεκριμένα στην περίπτωση του landscape mode, τοποθετούνται στο φάκελο res/layout-land/ (δηλαδή layout landscape). Άρα λοιπόν, δημιουργούμε ένα ακόμα αρχείο xml με το ίδιο ακριβώς όνομα mythirdscreen.xml το οποίο όμως τοποθετούμε στο φάκελο res/layout-land/. Η εντολή που χρησιμοποιούμε για να δηλώσουμε το αρχείο xml που θα χρησιμοποιηθεί σε μία activity είναι η setContentView(). Επομένως, σε κάθε μία από τις activities μας myFirstActivity.java, mySecondActivity.java και myThirdActivity.java γράφουμε την εντολή setContentView(R.layout.myfirstscreen), setContentView(R.layout.mysecondscreen) και setContentView(R.layout.mythirdscreen) αντίστοιχα. Όταν θα εκτελεστεί η εφαρμογή μας και συγκεκριμένα η εντολή setContentView() το λειτουργικό σύστημα Android θα ελέγξει τον προσανατολισμό της συσκευής μας και θα επιλέξει το κατάλληλο αρχείο xml. Αν βρισκόμαστε στην πρώτη οθόνη σε προσανατολισμό portrait και αλλάξουμε προσανατολισμό σε landscape, τότε το Android θα ελέγξει αν υπάρχχει το αρχείο myfirstscreen.xml στον φάκελο res/layout-land/. Δε θα το βρει όμως διότι δεν το έχουμε ορίσει, άρα θα χρησιμοποιήσει το προεπιλεγμένο αρχείο το οποίο βρίσκεται στο φάκελο res/layout/ και τελικά η διάταξη των στοιχείων θα είναι κοινή και για τους δύο προσανατολισμούς. Τα ίδια ισχύουν και για τη δεύτερη οθόνη. Στην Τρίτη οθόνη, όταν αλλάξουμε προσανατολισμό από portrait σε landscape τότε θα βρεθεί το αρχείο mythirdscreen.xml στο φάκελο res/layout-land/ άρα τελικά θα χρησιμοποιηθεί αυτό για τη διάταξη των γαρφικών στοιχείων. Φυσικά όταν επιστρέψουμε πάλι σε portrait mode θα αλλάξει και η διάταξη.

### 3.5.2 Μέγεθος Οθόνης

Παραπάνω μιλήσαμε για την περίπτωση αλλαγής του προσανατολισμού της συσκευής. Με τον ίδιο τρόπο μπορούμε να χρησιμοποιήσουμε διαφορετικά γραφικά στοιχεία ή διαφορετική διάταξη των στοιχείων αυτών ανάλογα με το μέγεθος της οθόνης. Για παράδειγμα αν θέλουμε

να εμφανίσουμε κάποιες εικόνες, τότε στην οθόνη του κινητού τηλεφώνου (από 3 εώς 4 ίντσες) θα χρησιμοποιήσουμε διαφορετική ανάλυση για την κάθε εικόνα, από αυτήν που θα χρησιμοποιούσαμε για ένα tablet pc (περίπου 10 ίντσες οθόνη). Ομοίως αν θέλουμε να εμφανίσουμε αρκετές εικόνες σε μία οθόνη, στο κινητό τηλέφωνο για να είναι ευδιάκριτες μπορούμε να εμφανίσουμε πολλές παραπάνω.

Ο τρόπος που υλοποιείται το παραπάνω σενάριο είναι παρόμοιος με τον τρόπο διάταξης σε portrait και landscape προσανατολισμό. Όλες οι εικόνες αποθηκεύονται στο φάκελο res/drawable/. Για να χρησιμοποιήσουμε διαφορετικές εικόνες ανάλογα με το μέγεθος της οθόνης, τις αποθηκεύουμε σε διαφορετικούς φακέλους όπως res/drawable-small, res/drawable-normal (είναι ουσιαστικά ίδιος με τον προεπιλεγμένο φάκελο res/drawable/), res/drawbale-large και res/drawable-xlarge/. Το λειτουργικό σύστημα θα εντοπίσει αυτόματα την οθόνη της συσκευής του χρήστη και θα επιλέξει υα κατάλληλα αρχεία. Στην περίπτωση της διαφορετικής διάταξης, π.χ. για πολύ μεγάλες οθόνες, θα πρέπει να αποθηκεύουμε επιπλέον αρχεία xml που ορίζουν τη διάταξη, εκτός από τον φάκελο res/layout/, και στον φάκελο res/layout-xlarge/.

Στις activities, λοιπόν, θα χρησιμοποιούμε ενιαίες εντολές, χωρίς να δηλώνουμε κάτι σχετικό με το μέγεθος της οθόνης ή άλλη παράμετρο, και το λειτουργικό σύστημα θα επιλέγει αυτόματα τα κατάλληλα αρχεία από αυτά που έχουμε δηλώσει. Για παράδειγμα, εμφανίζουμε μία εικόνα στην οθόνη μας με τις παρακάτω εντολές:

```
ImageView imageView = (ImageView) findViewById(R.id.myimageview);
imageView.setImageResource(R.drawable.myimage);
```

Αν έχουμε αποθηκεύσει το αρχείο myimage.jpg σε διαφορετικούς φακέλους για τα διάφορα μεγέθη της οθόνης, το λειτουργικό σύστημα θα εντοπίσει το κατάλληλο αυτόματα για να το εμφανίσει.

### 3.5.3 Γλώσσα

Οι εφαρμογές Android πρέπει να απευθύνονται σε ευρύ κοινό και όχι μόνο στη χώρα κατασκευής τους. Το σημαντικότερο ρόλο σε αυτό διαδραματίζει η φυσική γλώσσα που θα είναι γραμμένη η εφσρμογή. Αξίζει, λοιπόν, να προσπαθήσουμε να μεταφράσουμε την εφαρμογή μας σε πολλές διαφορετικές γλώσσες, ειδικά μάλιστα όταν το Android μας προσφέρει τέτοια δυνατότητα με αρκετά εύκολο τρόπο.

Το κείμενο που εμφανίζεται στηνεφαρμογή είναι ουσιαστικά ένα σύνολο από strings. Είτε αυτό αφορά κείμενο σε κάποιο κουμπί, κείμενο κάτω από μία εικόνα, κείμενο σε μία ειδοποίηση ή ένα διάλογο. Έχουμε λοιπόν δύο τρόπους για να εμφανίσουμε ένα string σε κάποιο σημείο της εφαρμογής μας. Ο πρώτος τρόπος είναι ο συνηθισμένος που εφαρμόζεται στα περισσότερα προγράμματα. Δηλαδή ορίζουμε στον κώδικα μας ένα string με την τιμή του, για παράδειγμα String helloWorld = “Hello World”. Έπειτα εμφανίζουμε το string αυτό κάνοντας αναφορά στη μεταβλητή μας, δηλαδή την helloWorld. Σκεφτείτε, όμως, σε μία εφαρμογή που έχουμε χιλιάδες γραμμές κώδικα, να θέλουμε να μεταφράσουμε όλα αυτά τα strings, ψάχνοντας τα ένα-ένα. Τη λύση έρχεται να μας δώσει το Android, η οποία αποτελεί και το δεύτερο (και σωστό) τρόπο εμφάνισης των strings. Στο φάκελο res/values/ του project μας υπάρχει το αρχείο strings.xml. Στο αρχείο αυτό δηλώνουμε όλα τα string που χρησιμοποιεί η εφαρμογή μας με τον τρόπο <string name=”helloWorld”>Hello World</string>, και αναφερόμαστε σε αυτα μέσα στις κλάσεις μας με την εντολή getString(R.string.helloWorld). Με τον τρόπο αυτό ανεξαρτητοποιούμε τη γλώσσα της εφαρμογής μας, με τον κώδικα που γράφουμε στις activities.

Έστω, λοιπόν, ότι θέλουμε προεπιλεγμένη γλώσσα της εφαρμογής μας να είναι η αγγλική, άρα στο αρχείο res/values/strings.xml δηλώνουμε τα strings με το όνομα τους και την τιμή τους στην αγγλική γλώσσα. Για λόγους συμβατότητας, θέλουμε οι Έλληνες χρήστες να χρησιμοποιούν την εφαρμογή μας στην ελληνικλη γλώσσα. Δημιουργούμε λοιπόν, νέο φάκελο res/values-el/ και μέσα στο φάκελο αυτόν δημιουργούμε το αρχείο strings.xml. Χρησιμοποιούμε ακριβώς το ίδιο όνομα για κάθε string αντίστοιχα με το προηγούμενο αρχείο

strings.xml, αλλά τώρα με διαφορετική τιμή, δηλαδή στην ελληνική γλώσσα. Όταν ο χρήστης εγκαταστήσει και εκτελέσει την εφαρμογή, το Android θα εντοπίσει αυτόματα τη γλώσσα που χρησιμοποιεί ο χρήστης στη συσκευή του. Αν, λοιπόν ο χρήστης χρησιμοποιεί την ελληνική γλώσσα τότε το Android για κάθε string που γίνεται αναφορά στον κώδικα, θα χρησιμοποιεί αυτό που βρίσκεται στο αρχείο strings.xml στο φάκελο res/values-el/. Με τον ίδιο τρόπο μπορούμε να δημιουργήσουμε φακέλους και για άλλες γλώσσες όπως res/values-de/, res/values-fr/ για γερμανική και γαλλική γλώσσα. Σε περίπτωση που δε βρεθεί φάκελος για τη γλώσσα που χρησιμοποιεί ο χρήστης, τότε χρησιμοποιούνται οι τιμές που έχουμε στον προεπιλεγμένο φάκελο res/values/.

Εκτός από τη γλώσσα του κειμένου της εφαρμογής μας, τα παραπάνω μπορούν να εφαρμοστούν και σε κάποιο άλλο φάκελο των resources μας, όπως π.χ. το φάκελο που δηλώνουμε τη διεπαφή χρήστη, δηλαδή τον res/layout/. Για παράδειγμα η γερμανική γλώσσα συνήθως χρησιμοποιεί μεγαλύτερες σε μήκος λέξεις από την αγγλική. Επομένως, ορισμένα στοιχεία της εφαρμογής μας όπως κουμπιά ή κείμενο, μπορεί να εμφανίζονται με διαφορετικό τρόπο από αυτόν που θέλουμε, διότι τα strings που χρησιμοποιούμε είναι τώρα υπερβολικά μεγάλα. Επομένως, μπορούμε να δημιουργήσουμε νέο φάκελο res/layout-de/ και τα αρχεία xml (δηλαδή η διάταξη) που θα ορίσουμε μέσα σε αυτόν, θα χρησιμοποιούνται μόνο σε περίπτωση γερμανικής γλώσσας της συσκευής του χρήστη.

Στο παρακάτω παράδειγμα ορίζουμε στο αρχείο res/values/strings.xml το string <string name=”hello”>Hello!</string> και στο αρχείο res/values-el/strings.xml το string <string name=”hello”>Γεια σου!</string> (ίδιο όνομα, διαφορετική τιμή). Όταν το εμφανίσουμε στην οθόνη το αποτέλεσμα θα είναι το παρακάτω:


Εικόνα 3.10: Η εφαρμογή όπως φαίνεται σε συσκευή που χρησιμοποιεί οποιαδήποτε γλώσσα εκτός της ελληνικής (χρησιμοποιείται το προεπιλεγμένο αρχείο res/values/strings.xml)


Εικόνα 3.11: Η ίδια εφαρμογή όπως φαίνεται σε συσκευή που χρησιμοποιεί ελληνική γλώσσα (χρησιμοποιείται το αρχείο res/values-el/strings.xml)

Εκτός από τα παραπάνω (προσανατολισμός, μέγεθος οθόνης, γλώσσα) υπάρχουν και άλλες παράμετροι που δηλώνονται με τον ίδιο τρόπο και το λειτουργικό σύστημα τις εντοπίζει αυτόματα. Μερικές από αυτές είναι η πυκνότητα της οθόνης, η νυχτερινή λειτουργία, αν υπάρχει οθόνη αφής, αν υπάρχει φυσικό πληκτρολόγιο και η έκδοση του λειτουργικού συστήματος. Δηλαδή ανάλογα με τις παραπάνω παραμέτρους της συσκευής του χρήστη, μπορεί να αλλάζει αυτόματα η διάταξη των γραφικών στοιχείων, η γλώσσα του κειμένου, οι εικόνες που χρησιμοποιούμε και άλλα.

Τα παραπάνω μπορούν να λειτουργήσουν και συνδυαστικά. Για παράδειγμα αν δηλώσουμε το user interface μίας οθόνης στο φάκελο res/layout-el-port/, τότε αυτό θα χρησιμοποιηθεί αν η συσκευή του χρήστη χρησιμοποιεί την ελληνικλη γλώσσα και βρίσκεται σε προσανατολισμό πορτρέτου.

Το συμπέρασμα που προκύπτει για το σωστό προγραμματισμό της εφαρμογής μας είναι πάντα να εξωτερικεύουμε τα application resources από το κομμάτι του κώδικα των κλάσεων. Υπάρχουν κατάλληλοι φάκελοι που αποθηκεύουμε ή δηλώνουμε τα παραπάνω στοιχεία και πρέπει να χρησιμοποιούνται. Έτσι η εφαρμογή μας καθίσταται κατανοητή στον προγραμματιστή και εύκολα επεξεργάσιμη, ενώ παράλληλα γίνεται συμβατή με όλες τις δυνατές παραμέτρους της συσκευής του κάθε χρήστη.

## 3.6 Android Manifest

Σε κάθε εφαρμογή πρέπει να υπάρχει το αρχείο AndroidManifest.xml, το οποίο δημιουργείται αυτόματα όταν ξεκινάμε καινούργιο project μίας Android εφαρμογής. Το αρχείο αυτό περιέχει βασικές πληροφορίες σχετικά με την εφαρμογή, τις οποίες το λειτουργικό σύστημα πρέπει να γνωρίζει προτού τρέξει οποιοδήποτε άλλο κώδικα. Οι σημαντικότερες από αυτές τις πληροφορίες περιγράφονται παρακάτω:

- Η ονομασία του πακέτου Java της εφαρμογής (Java package)
- Η έκδοση της εφαρμογής (π.χ. 1.0, 1.4.2, 2.7 κ.λ.π)
- Η ελάχιστη έκδοση του λειτουργικού συστήματος Android που απαιτεί η εφαρμογή (min sdk version). Για παράδειγμα αν έχει δηλωθεί min sdk version ο αριθμός 7, που ισοδυναμεί με την έκδοση Android 2.1, τότε η εφαρμογή θα μπορεί να εκτελεστεί σε συσκευές με έκδοση Android μεγαλύτερη ή ίση της 2.1
- Το όνομα της εφαρμογής, καθώς και το εικονίδιο της.
- Οι άδειες που απαιτούνται για να εκτελεστούν ορισμένες λειτουργίες της εφαρμογής. Για παράδειγμα αν η εφαρμογή μας χρησιμοποιεί την κάμερα της συσκευής, θα πρέπει να δηλώνεται και η αντίστοιχη άδεια. Αν θέλουμε να έχουμε πρόσβαση στο διαδίκτυο από την εφαρμογή μας, θα πρέπει να δηλώνεται επίσης και η αντίστοιχη άδεια. Όμοια ισχύουν για το αν θέλουμε να αποθηκεύσουμε αρχεία στην κάρτα SD, αν θέλουμε να στείλουμε μηνύματα SMS, αν θέλουμε να πραγματοποιήσουμε τηλεφωνικές κλήσεις και άλλες λειτουργίες. Αν δεν δηλώσουμε τις άδειες που απαιτούνται για τις διάφορες

λειτουργίες της εφαρμογής, τότε θα εμφανίζεται σφάλμα και η εφαρμογή δεν θα λειτουργεί. Οι άδειες αυτές περιγράφονται στο χρήστη τη στιγμή που εγκαθιστά την εφαρμογή, και πρέπει να συμφωνήσει με αυτές για να ολοκληρωθεί η εγκατάσταση. Με τον τρόπο αυτό, ο χρήστης αισθάνεται ασφαλής απέναντι σε κακόβουλες εφαρμογές

- Όλα τα συστατικά στοιχεία (activities, services, content providers, broadcast receivers) της εφαρμογής. Για παράδειγμα, αν δημιουργήσουμε κάποια κλάση για activity της εφαρμογής, χωρίς να τη δηλώσουμε στο AndroidManifest.xml, δεν θα μπορέσει να λειτουργήσει
- Οι εξωτερικές βιβλιοθήκες που χρησιμοποιεί η εφαρμογή μας. Για παράδειγμα, αν η εφαρμογή μας, σε κάποιο σημείο του κώδικα τρέχει την εφαρμογή google maps, τότε θα πρέπει να δηλωθεί η βιβλιοθήκη com.google.android.maps

Ακολουθεί ένα παράδειγμα του AndroidManifest.xml μίας εφαρμογής:

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

  <uses-sdk android:minSdkVersion="7" />

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

  <uses-feature android:name="android.hardware.camera" />

  <application android:icon="@drawable/icon" android:label="@string/app_name"
```

```
android:theme="@style/myStyle">

    <uses-library android:name="com.google.android.maps" />

    <activity android:name=".myFirstActivity" android:label="@string/app_name">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
    <activity android:name=".mySecondActivity"></activity>
    <activity android:name=".myThirdActivity"></activity>

  </application>
</manifest>
```

# Κεφάλαιο 4

## Αρχιτεκτονική της εφαρμογής

---



---

### 4.1 Εισαγωγή

Στο κεφάλαιο αυτό που ακολουθεί αρχικά θα περιγράψουμε την εφαρμογή που αναπτύχθηκε για το λειτουργικό σύστημα Android, αναφέροντας όλες τις δυνατότητες που προσφέρει η εφαρμογή στο χρήστη. Έπειτα θα αναλύσουμε τη δομή της, τα αρχεία δηλαδή στα οποία οργανώνεται, καθώς και τη χρήση του καθενός. Η εφαρμογή που κατασκευάστηκε ονομάστηκε “iReader”.

### 4.2 Περιγραφή

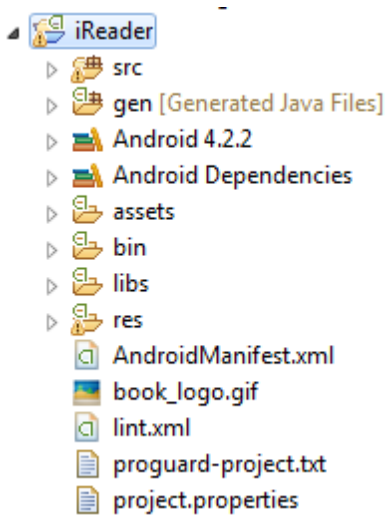
Κύριος σκοπός της εφαρμογής είναι να δίνει την δυνατότητα στον χρήστη να αναζητήσει κάποιο βιβλίο που είναι ήδη ενσωματωμένο στην βάση δεδομένων της εφαρμογής. Με αυτή την επιλογή, μπορεί ο χρήστης να διαβάσει το βιβλίο μέσω αυτής της εφαρμογής.

Παράλληλα, ο χρήστης μπορεί να ανεβάσει κάποιο βιβλίο, το οποίο είναι ήδη αποθηκευμένο στην SD card της συσκευής του χρήστη. Έτσι, την επόμενη φορά που ο χρήστης θα

χρησιμοποιήσει την εφαρμογή και αναζητήσει το βιβλίο που είχε ανεβάσει θα παρατηρήσει ότι είναι ήδη καταχωρημένο πλέον.

## 4.3 Δομή

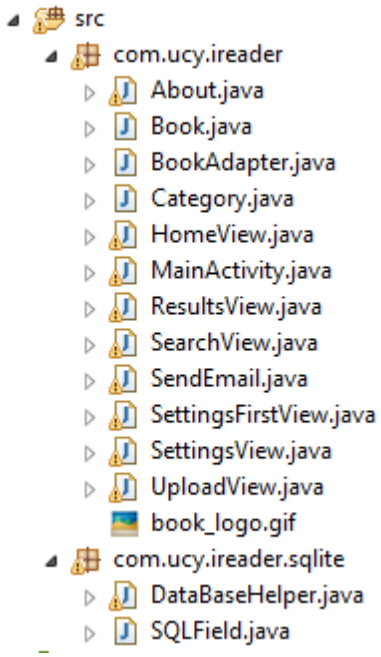
Πριν προχωρήσουμε στη λεπτομερή ανάλυση των διάφορων λειτουργιών της εφαρμογής που αναπτύχθηκε, θα ήταν χρήσιμο να παρουσιάσουμε πρώτα τη δομή της, πώς δηλαδή οργανώνονται όλα τα αρχεία που περιέχονται σε αυτήν και ποια είναι η κύρια λειτουργία τους. Στο προγραμματιστικό περιβάλλον Eclipse στο οποίο αναπτύχθηκε η εφαρμογή, το project εμφανίζεται με την παρακάτω δομή:




Εικόνα 4.1: Δομή της εφαρμογής

Ακολουθεί η επεξήγηση των παραπάνω φακέλων και αρχείων καθώς και του περιεχομένου τους:

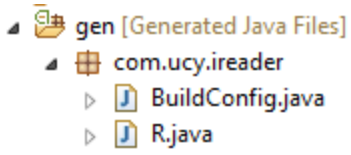
- **scr/**




Εικόνα 4.2: Περιεχόμενα φακέλου src/

Ο φακέλος αυτός αρχικά περιέχει τα πακέτα τα οποία έχουμε δηλώσει στην εφαρμογή μας. Έχουμε δηλώσει δύο πακέτα, το com.ucy.ireader και το com.ucy.ireader.sqlite. Στο πρώτο πακέτο περιέχονται όλες οι κλάσεις της εφαρμογής μας. Οι κλάσεις αυτές χωρίζονται στις activities, δηλαδή στις διαφορετικές οθόνες της εφαρμογής μας. Στο πακέτο com.ucy.ireader.sqlite υπάρχουν οι κλάσεις οι οποίες αφορούν την βάση δεδομένων που υπάρχει, στην οποία βρίσκονται καταχωρημένα τα βιβλία και πιο συγκεκριμένα η κατηγορία που ανήκουν, το έτος κυκλοφορίας τους, τον συγγραφέα και τον τίτλο τους.

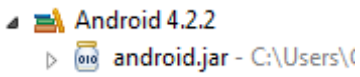
- **gen/**




Εικόνα 4.3: Περιεχόμενα φακέλου gen/

Ο φάκελος αυτός περιέχει για το πακέτο που χρησιμοποιούμε στην εφαρμογή μας com.ucy.ireader, το αρχείο R.java το οποίο ουσιαστικά είναι μία κλάση η οποία κατασκευάζεται αυτόματα από το σύστημα όπως παρομοίως και η κλάση BuildConfig.java. Η R.java συνδέει όλα τα application resources με τον κώδικα της εφαρμογής μας. Έχουμε εξηγήσει ότι όλα αυτά τα resources δηλώνονται σε ξεχωριστά αρχεία και όχι μέσα στις κλάσεις της εφαρμογής μας. Η κλάση, λοιπόν, R.java δηλώνει κάθε στοιχείο από τις εικόνες, τα γραφικά στοιχεία, τα strings, τα styles ή themes, που χρησιμοποιούμε στην εφαρμογή μας, σαν μία ακέραια σταθερά (int) και με τιμή έναν μοναδικό δεκαεξαδικό αριθμό, ώστε να αναφερόμαστε σε αυτές μέσα από τις δικές μας κλάσεις σαν δημόσιες σταθερές (public constants).

- **Google APIs**




Εικόνα 4.4: Βιβλιοθήκη του λειτουργικού συστήματος Android

Πρόκειται για την βιβλιοθήκη που επιλέγεται αυτόματα από το σύστημα, όταν εμείς δηλώσουμε την ελάχιστη έκδοση Android με την οποία θα είναι συμβατή η εφαρμογή μας. Συγκεκριμένα, όταν ξεκινάμε ένα project για την κατασκευή Android application τότε ανάμεσα στις διάφορες επιλογές, πρέπει να επιλέξουμε και το Build Target της εφαρμογής μας, που σημαίνει την ελάχιστη έκδοση Android με την οποία θέλουμε να είναι συμβατή η εφαρμογή μας. Έτσι το σύστημα, τοποθετεί τις κατάλληλες βιβλιοθήκες για να λειτουργεί η εφαρμογή μας στις εκδόσεις αυτές. Προφανώς, το ιδανικό θα ήταν η εφαρμογή μας να είναι συμβατή με όλες τις εκδόσεις Android, όμως κάτι τέτοιο δεν είναι πάντα εφικτό, διότι κάθε νέα έκδοση Android έχει τις ίδιες και ορισμένες επιπλέον δυνατότητες από την αμέσως προηγούμενη. Επομένως, όταν στην εφαρμογή μας θέλουμε να χρησιμοποιήσουμε μία εξειδικευμένη λειτουργία του συστήματος, τότε πιθανόν η λειτουργία αυτή να μην υπάρχει για παλαιότερες εκδόσεις Android, πράγμα το οποίο θα αποκλείσει την εφαρμογή μας από τις εκδόσεις αυτές. Διαισθητικά μπορούμε να σκεφτούμε ότι γενικά οι απλές εφαρμογές είναι συμβατές για όλες τις εκδόσεις Android, οι λίγο πιο εξειδικευμένες είναι συμβατές για εκδόσεις μεγαλύτερες ή ίσες της 2.1, ενώ άλλες ακόμα πιο εξειδικευμένες είναι συμβατές για εκδόσεις μεγαλύτερες ή ίσες της 2.2. Το αρχείο, λοιπόν, android.jar αποτελεί τη βασική βιβλιοθήκη του λειτουργικού συστήματος.

- **assets/**

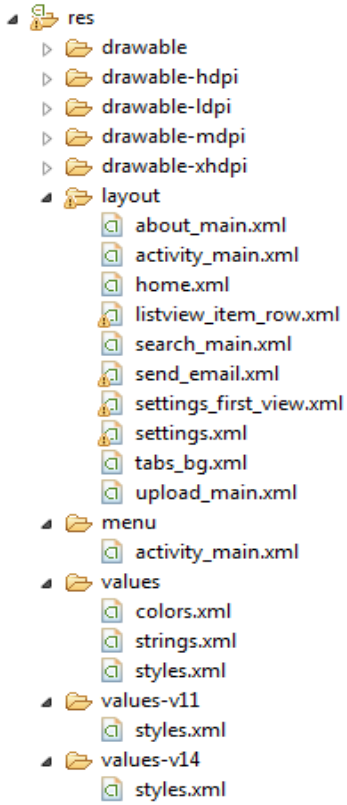
Στο φάκελο αυτό μπορούμε να αποθηκεύσουμε οποιοδήποτε αρχείο θέλουμε ή και να δημιουργήσουμε υποφακέλους. Ωστόσο όλα τα application resources αποθηκεύονται σε διαφορετικό φάκελο. Στην συγκεκριμένη εφαρμογή, στον φάκελο αυτό έχουμε αποθηκευμένα τα βιβλία που είναι καταχωρημένα καθώς και την βάση δεδομένων που χρησιμοποιούμε.

- **libs/**

Ο φάκελος αυτός αποτελεί τη φυσική τοποθεσία των βιβλιοθηκών. Δηλαδή ο προγραμματιστής τοποθετεί στο φάκελο αυτό τα αρχεία .jar τα οποία θέλει να χρησιμοποιήσει, και έπειτα τα συνδέει με το λειτουργικό σύστημα Android. Τα συνδεδεμένα

αρχεία, τα οποία τελικά χρησιμοποιούνται αποτελούν τις Android Dependecies Libraries. Στην συγκεκριμένη εφαρμογή χρησιμοποιείται η βιβλιοθήκη android-support-v4.jar.

- **res/**




Εικόνα 4.5: Όλα τα resources της εφαρμογής

Στο φάκελο res/ αποθηκεύονται όλα τα application resources, τα οποία χρησιμοποιούμε. Συγκεκριμένα, στο φάκελο drawable αποθηκεύουμε όλα τα αρχεία εικόνας που χρησιμοποιούνται στην εφαρμογή. Στο φάκελο layout αποθηκεύονται τα αρχεία xml στα οποία δηλώνουμε το user interface της κάθε οθόνης. Ο φάκελος values περιλαμβάνει το αρχείο strings.xml με όλα τα strings της εφαρμογής δηλωμένα εκεί, και το αρχείο styles.xml, στο οποίο δηλώνονται τα styles τα οποία θέλουμε να χρησιμοποιήσουμε είτε σε όλη την εφαρμογή, είτε σε συγκεκριμένη οθόνη, είτε σε συγκεκριμένο γραφικό στοιχείο. Στο αρχείο colors.xml υπάρχουν τα χρώματα που θα χρησιμοποιηθούν στην εφαρμογή.

Τέλος, παρατηρούμε τρία αρχεία τα οποία δημιουργούνται αυτόματα όταν ξεκινάμε καινούργιο project για μία εφαρμογή Android. Το αρχείο AndroidManifest.xml έχει αναλυθεί σε προηγούμενη ενότητα. Τα άλλα δύο αρχεία proguard-project.txt και project.properties.txt αφορούν το σύστημα και δεν θα πρέπει να τα επεξεργαζόμαστε.

# Κεφάλαιο 5

## Εγχειρίδιο χρήσης εφαρμογής – User Guide

---



---

### 5.1 Εισαγωγή

Στην εφαρμογή αυτή, υπάρχουν ήδη ενσωματωμένα βιβλία τύπου pdf και ο χρήστης είναι σε θέση να κάνει αναζήτηση για το βιβλίο που τον ενδιαφέρει. Σε αυτή την εφαρμογή περιέχονται μόνο ελληνικά βιβλία. Επίσης, ο χρήστης έχει την δυνατότητα να ανεβάσει οποιοδήποτε βιβλίο επιθυμεί στην συγκεκριμένη εφαρμογή, το οποίο παρομοίως πρέπει να είναι τύπου pdf. Στην εφαρμογή υπάρχει επίσης μία κατηγορία About, στην οποία υπάρχει μια μικρή περιγραφή αυτής της εφαρμογής. Και τέλος, υπάρχει μία κατηγορία Settings, στην οποία ο χρήστης μπορεί να προσαρμόσει την φωτεινότητα της εφαρμογής είτε σε πιο φωτεινή είτε σε πιο σκοτεινή οθόνη. Επίσης, δίνεται η δυνατότητα στον χρήστη να στείλει email σε μια προκαθορισμένη διεύθυνση που έχει οριστεί, τοποθετώντας σχόλια για την εφαρμογή είτε κάποιες απορίες που μπορεί να υπάρξουν σχετικά με την λειτουργία της εφαρμογής.

## 5.2 Αρχική Οθόνη

Η αρχική οθόνη χρησιμεύει ως καλωσόρισμα στην εφαρμογή. Η συγκεκριμένη οθόνη περιλαμβάνει τα εξής:

- Εμφανίζει το όνομα της εφαρμογής
- Εμφανίζει το λογότυπο της εφαρμογής
- Εμφανίζει μία σειρά από εξώφυλλα βιβλίων, τα οποία φαίνονται μέσα από slideshow

Η Εικόνα 5. 1 παρουσιάζει το πρότυπο της Κύριας Οθόνης αυτής της εφαρμογής.


Εικόνα 5.1: Αρχική Οθόνη

Ο χρήστης στην οθόνη αυτή, έχει την δυνατότητα να δει μία ροή από εξώφυλλα γνωστών βιβλίων, καθώς επίσης μπορεί να σταματήσει αυτή την ροή πατώντας μόνο ένα κουμπί (Stop SlideShow).

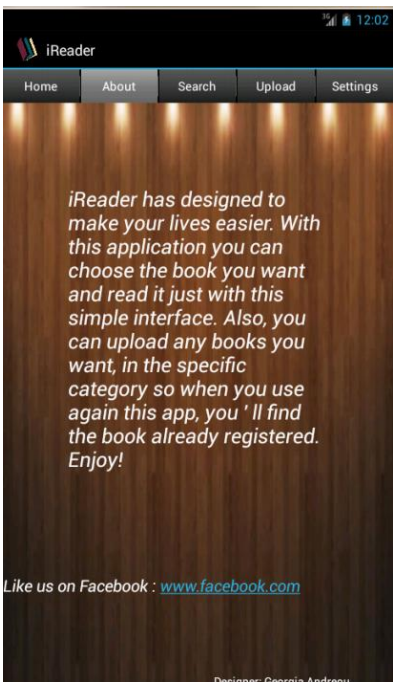
Στην οθόνη αυτή, όπως και σε κάθε άλλη οθόνη, εμφανίζονται τα tabs της εφαρμογής, τα οποία απαρτίζουν το action bar και κατ'επέκταση τις επιλογές που δίνονται στον χρήστη.

## 5.3 Οθόνη «Σχετικά»

Η οθόνη αυτή χρησιμεύει στο να δίνει μια γεύση στον χρήστη περί τίνος πρόκειται αυτή η εφαρμογή. Περιγράφει δηλαδή την λειτουργία της και την χρησιμότητα της. Αυτή η οθόνη εμφανίζει τα εξής:

- Ένα κείμενο στο οποίο γίνεται η περιγραφή της εφαρμογής iReader
- Ένα link στο οποίο μπορεί ο χρήστης να μεταφερθεί στην σελίδα που δημιουργήθηκε στο κοινωνικό δίκτυο Facebook με θέμα αυτή την εφαρμογή

Η Εικόνα 5.2 παρουσιάζει το πρότυπο αυτής της οθόνης.



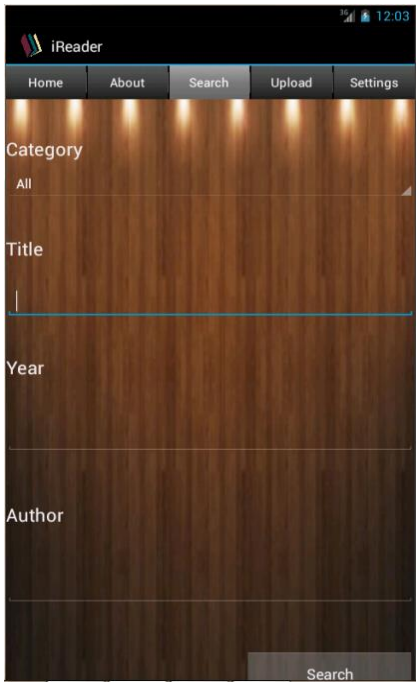

Εικόνα 5.2: Οθόνη Σχετικά (About)

## 5.4 Οθόνη Αναζήτησης

Η οθόνη αναζήτησης δίνει στον χρήστη την δυνατότητα να αναζητήσει το βιβλίο που επιθυμεί, δίνοντας του τις επιλογές αναζήτησης είτε με κατηγορία που ανήκει το βιβλίο, είτε με τον τίτλο του βιβλίου, είτε με χρονιά κυκλοφορίας, είτε με το όνομα του συγγραφέα. Αυτή η οθόνη περιλαμβάνει τα εξής:

- Επιλογή από διάφορες κατηγορίες βιβλίων
- Χώρο για εισαγωγή τίτλου βιβλίου
- Χώρο για εισαγωγή χρονιάς κυκλοφορίας
- Χώρο για εισαγωγή του ονόματος του συγγραφέα

Η Εικόνα 5.3 παρουσιάζει το πρότυπο της Οθόνης Αναζήτησης και η Εικόνα 5.4 την οθόνη αποτελεσμάτων.




Εικόνα 5.3: Οθόνη Αναζήτησης


Εικόνα 5.4: Οθόνη Αποτελεσμάτων

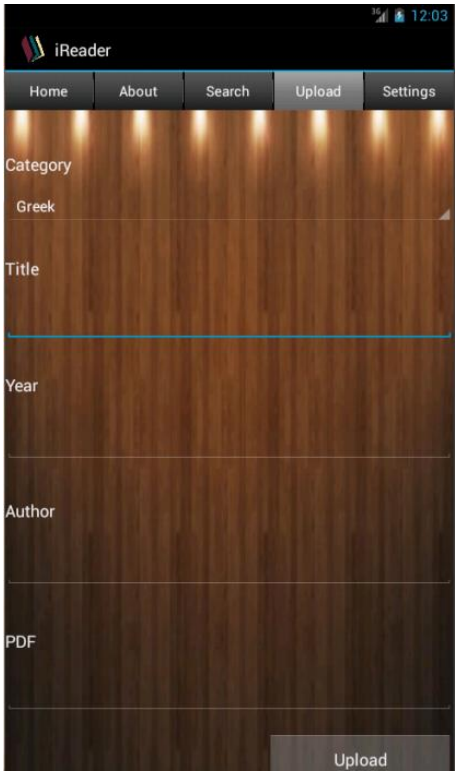
Ο χρήστης σε αυτή την οθόνη συμπληρώνει τα πεδία που επιθυμεί να γίνει αναζήτηση με βάση αυτά. Όταν ο χρήστης πατήσει το κουμπί “Search” τότε θα του εμφανιστεί μία νέα οθόνη με τα αποτελέσματα που βρέθηκαν βασισμένα στα πεδία, τα οποία έχει συμπληρώσει ο χρήστης. Αν όμως ο χρήστης, επιθυμεί να δει όλα τα βιβλία που είναι καταχωρημένα στην εφαρμογή, τότε στο Category επιλέγει All και εμφανίζονται όλα τα βιβλία χωρισμένα ανάλογα με την κατηγορία που ανήκουν.

## 5.5 Οθόνη Upload

Η οθόνη αυτή δίνει την δυνατότητα στον χρήστη να ανεβάσει κάποιο βιβλίο στην εφαρμογή, έτσι ώστε την επόμενη φορά που θα χρησιμοποιήσει την εφαρμογή το βιβλίο αυτό θα είναι ήδη καταχωρημένο. Το βιβλίο που θα ανεβάσει, επίσης, θα πρέπει να είναι ήδη αποθηεκυμένο στην κάρτα μνήμης της συσκευής του χρήστη (sdcard). Η οθόνη αυτή περιλαμβάνει τα εξής:

- Την κατηγορία του βιβλίου που επιθυμεί ο χρήστης να ανεβάσει
- Τον τίτλο του βιβλίου
- Την χρονιά που κυκλοφόρησε
- Τον συγγραφέα του βιβλίου
- Το μονοπάτι (path) στο οποίο υπάρχει το βιβλίο pdf στην συσκευή του χρήστη

Η Εικόνα 5.5 παρουσιάζει το πρότυπο της Οθόνης Upload.



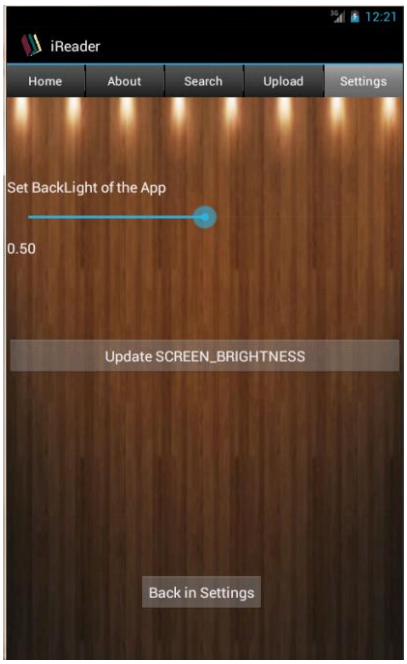

Εικόνα 5.5: Οθόνη Upload

Με αυτό τον τρόπο το βιβλίο καταχωρείται πλέον στα βιβλία που παρέχει η εφαρμογή.

## 5.6 Οθόνη Ρυθμίσεων

Η οθόνη αυτή δίνει την δυνατότητα στον χρήστη να ρυθμίσει την φωτεινότητα της συγκεκριμένης εφαρμογής. Επίσης, σε αυτή την οθόνη ο χρήστης μπορεί να στείλει ηλεκτρονικό ταχυδρομικό μήνυμα (email) σε μια προκαθορισμένη διεύθυνση, η οποία είναι δημιουργημένη αποκλειστικά για τις ανάγκες της εφαρμοφής. Με αυτό τον τρόπο, ο χρήστης μπορεί να δώσει σχόλια για την εφαρμογή είτε να διατυπώσει τυχόν διευκρινήσεις που χρειάζεται για την εγαρμογή. Η οθόνη αυτή περιλαμβάνει τα εξής:

- Κουμπί για την ρύθμιση της φωτεινότητας της εφαρμογής
- Κουμπί για αποστολή ηλεκτρονικού μηνύματος


Η Εικόνα 5.6 παρουσιάζει το πρότυπο της Οθόνης Ρυθμίσεων.

Για την επιλογή “Send Email”, ο χρήστης επιλέγει με ποιο τρόπο επιθυμεί να στείλει το μήνυμα του (Gmail, Hotmail) και στην συνέχεια μεταφέρεται στην αντίστοιχη σελίδα.

# Κεφάλαιο 6

## Συμπεράσματα και μελλοντικές εξελίξεις

---



---

### 6.1 Σύνοψη και συμπεράσματα

Στην παρούσα διπλωματική εργασία ασχοληθήκαμε με την ανάπτυξη μίας εφαρμογής, η οποία θα εκτελείται σε κινητές συσκευές που χρησιμοποιούν το λειτουργικό σύστημα Android. Η επιλογή της ανάπτυξης της εφαρμογής για το συγκεκριμένο λειτουργικό σύστημα κρίνεται επιτυχής για αρκετούς λόγους:

- Το Android τη στιγμή αυτή χρησιμοποιείται στο μεγαλύτερο ποσοστό smartphones της αγοράς, ενώ περίπου 550.000 νέες συσκευές Android ενεργοποιούνται καθημερινά
- Οι μεγαλύτερες εταιρείες κατασκευής κινητών τηλεφώνων, όπως Sony Ericsson, Samsung, LG, HTC, Motorola, υιοθετούν το λειτουργικό αυτό σύστημα σε όλο και περισσότερες συσκευές τους
- Το Android Market τη στιγμή αυτή διαθέτει πάνω από 250.000 εφαρμογές, με περισσότερα από 6 δισεκατομμύρια downloads συνολικά
- Σε γενικές γραμμές, κρίνεται εύκολη η εκμάθηση της φιλοσοφίας του λειτουργικού αυτού συστήματος, καθώς και ο τρόπος ανάπτυξης εφαρμογών
- Δεν απαιτούνται εξειδικευμένες γνώσεις προγραμματισμού για την ανάπτυξη απλών εφαρμογών. Ο καθένας μπορεί να ασχοληθεί και να δημιουργήσει μία προσωπική

εφαρμογή, αρκεί να έχει εξοικειωθεί με τη γλώσσα προγραμματισμού Java. Ωστόσο, για απλές λειτουργίες, δεν απαιτούνται εξειδικευμένες γνώσεις της γλώσσας αυτής

- Όλα τα προγραμματιστικά εργαλεία που χρειάζονται για την ανάπτυξη εφαρμογών κυκλοφορούν ελεύθερα στο διαδίκτυο και καθένας μπορεί να τα κατεβάσει και να τα χρησιμοποιήσει χωρίς κανένα κόστος
- Η επίσημη ιστοσελίδα της Google σχετικά με την κατασκευή εφαρμογών σε Android, περιλαμβάνει αναλυτικούς οδηγούς που βοηθάνε τον προγραμματιστή στην εξοικείωση με τη νέα αυτή τεχνολογία
- Υπάρχουν πάρα πολλές διαδικτυακές κοινότητες που ασχολούνται καθαρά με τον προγραμματισμό εφαρμογών Android. Χαρακτηριστικό παράδειγμα αποτελεί το γεγονός ότι στις περισσότερες ιστοσελίδες ερωταπαντήσεων γενικού προγραμματιστικού ενδιαφέροντος, η ετικέτα “Android” βρίσκεται ανάμεσα στις δημοφιλέστερες. Αυτό σημαίνει ότι συνεχώς προγραμματιστές ανταλλάσουν ιδέες και πληροφορίες σχετικές με την κατασκευή εφαρμογών στο σύστημα αυτό

Η έρευνα που πραγματοποιήθηκε για την εκπόνηση της διπλωματικής αυτής εργασίας κρίνεται άκρως ενδιαφέρουσα. Κατανοήθηκε η έννοια της πρωσοποποίησης περιεχομένου και εφαρμόστηκαν στην πράξη βασικές έννοιες σχετικά με την αλληλεπίδραση ανθρώπου-μηχανής και την ανάλυση περιεχομένου. Αποκτήθηκαν πολύ σημαντικές γνώσεις για τις τεχνολογίες των έξυπνων κινητών τηλεφώνων και τον τρόπο ανάπτυξης εφαρμογών σε αυτά. Προτείνεται ανεπιφύλακτα η ενασχόληση με τις τεχνολογίες αυτές, σε φοιτητές, προγραμματιστές ή μηχανικούς, καθώς αποτελεί έναν καινούριο και συνεχώς αναπτυσσόμενο κλάδο της επιστήμης υπολογιστών, ο οποίος σε καμία περίπτωση δεν είναι κορεσμένος, αλλά αντίθετα δημιουργεί συνεχώς νέες θέσεις εργασίας.

## 6.2 Μελλοντικές εξελίξεις

Το Android Market δίνει τη δυνατότητα στους προγραμματιστές να δημοσιεύσουν ελεύθερα και χωρίς κόστος τις εφαρμογές τις οποίες έχουν κατασκευάσει. Η διαδικασία είναι απλή και το μόνο αρχείο που χρειάζεται για την υπηρεσία συτή είναι το εκτελέσιμο apk της εφαρμογής. Με τον τρόπο αυτό, η εφαρμογή γίνεται προσβάσιμη σε όλους τους χρήστες Android, οι οποίοι μπορούν να την κατεβάσουν ανά πάσα στιγμή από το Android Market, να τη βαθμολογήσουν ή να γράψουν κριτική.

Στο πλαίσιο αυτό, σχεδιάζεται η δημοσίευση της εφαρμογής μας στο Android Market, καθώς και η συνεχής ενημέρωση (update) αυτής για προσθήκη νέων δυνατοτήτων ή διόρθωση ορισμένων σφαλμάτων τα οποία αναφέρονται από χρήστες (feedback). Συγκεκριμένα προτείνονται οι παρακάτω επεκτάσεις και βελτιώσεις, προτού δημοσιευτεί η εφαρμογή στο Android Market, και οι οποίες είναι εύκολα υλοποιήσιμες:

- Βελτιωμένο user interface, ώστε να γίνει ακόμα πιο φιλικό προς το χρήστη
- Να δίνεται η δυνατότητα στον χρήστη να ανεβάζει βιβλίο από το διαδίκτυο απευθείας χωρίς να χρειάζεται να το έχει ήδη κατεβασμένο στην συσκευή του
- Μετάφραση της γλώσσας της εφαρμογής στις πιο δημοφιλείς ξένες γλώσσες, με στόχο να αποκτήσουν οι αντίστοιχοι χρήστες επιπλέον κίνητρο να τη χρησιμοποιήσουν

# ΠΑΡΑΡΤΗΜΑ

Στη συνέχεια παρατίθεται αναλυτικά ο κώδικας από όλα τα επιμέρους αρχεία της εφαρμογής. Αρχικά παρουσιάζονται οι κλάσεις της εφαρμογής, έπειτα τα αρχεία xml τα οποία βρίσκονται στο φάκελο res/layout/ και ορίζουν τη διεπαφή χρήστη για κάθε activity, τα αρχεία strings.xml και τέλος το AndroidManifest.xml.

### MainActivity.java

```
package com.ucy.ireader;

import java.io.IOException;

import com.ucy.ireader.sqlite.DataBaseHelper;

import android.os.Bundle;
import android.app.Activity;
import android.app.TabActivity;
import android.content.Intent;
import android.database.SQLException;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View.OnClickListener;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.Button;
import android.widget.TabHost;
import android.widget.TextView;
import android.widget.TabHost.TabContentFactory;
import android.widget.TabHost.TabSpec;
import android.widget.ViewFlipper;

public class MainActivity extends TabActivity {


	public static DataBaseHelper DatabaseHelper;
	private TabHost mTabHost;
	 int mFlipping = 0 ; // Initially flipping is off
	 Button mButton ; // Reference to button available in the layout
to start and stop the flipper

	private void setupTabHost() {
		mTabHost = (TabHost) findViewById(android.R.id.tabhost);
		mTabHost.setup();
	}

	@SuppressWarnings("deprecation")
	@Override
```

```
	protected void onCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);
		setContentView(R.layout.activity_main);

		setupTabHost();

	mTabHost.getTabWidget().setDividerDrawable(R.drawable.tab_divider
);

		setupTab(new TextView(this), "Home");
		setupTab(new TextView(this), "About");
		setupTab(new TextView(this), "Search");
		setupTab(new TextView(this), "Upload");
		setupTab(new TextView(this), "Options");









		DatabaseHelper = new DataBaseHelper(this);
		//DatabaseHelper.drop_database();
		try {
			DatabaseHelper.createDataBase();
			Log.e(">>>>>>", "Create SQL Database");
		} catch (IOException ioe) {
			throw new Error("Unable to create database");
		}

		try {
			if (DatabaseHelper.is_open())
				DatabaseHelper.close();

			DatabaseHelper.openDataBase();
			Log.e(">>>>>>", "Open SQL Database");
		} catch (SQLException sqle) {
			throw sqle;
		}

	}

	private void setupTab(final View view, final String tag) {
		View tabview = createTabView(mTabHost.getContext(), tag);

		TabSpec setContent =
mTabHost.newTabSpec(tag).setIndicator(tabview)
				.setContent(new TabContentFactory() {
					public View createTabContent(String tag) {
						return view;
```

```
                              }
                        });
            if (tag.equals("Home")) {
                  Intent intentAbout = new Intent(MainActivity.this,
HomeView.class);
                  setContent.setContent(intentAbout);
            }

            if (tag.equals("About")) {
                  Intent intentAbout = new Intent(MainActivity.this,
About.class);
                  setContent.setContent(intentAbout);
            }

            if (tag.equals("Search")) {
                  Intent intentAbout = new Intent(MainActivity.this,
SearchView.class);
                  setContent.setContent(intentAbout);
            }

            if (tag.equals("Upload")) {
                  Intent intentAbout = new Intent(MainActivity.this,
UploadView.class);
                  setContent.setContent(intentAbout);
            }

            if (tag.equals("Options")) {
                  Intent intentAbout = new Intent(MainActivity.this,
                              SettingsFirstView.class);
                  setContent.setContent(intentAbout);

            }

            mTabHost.addTab(setContent);
      }

      private static View createTabView(final Context context, final
String text) {
            View view = LayoutInflater.from(context)
                        .inflate(R.layout.tabs_bg, null);
            TextView tv = (TextView) view.findViewById(R.id.tabsText);
            tv.setText(text);
            return view;
      }

      @Override
      public boolean onCreateOptionsMenu(Menu menu) {
            // Inflate the menu; this adds items to the action bar if
it is present.

            getMenuInflater().inflate(R.menu.activity_main, menu);
```

```
            return true;
      }

      @Override
      public boolean onOptionsItemSelected(MenuItem item) {

            // Handle item selection
            switch (item.getItemId()) {
            case R.id.itm_search:

                  Log.e("search", "hello");
                  Intent intentSearch = new Intent(MainActivity.this,
                              SearchView.class);

                  startActivity(intentSearch);

                  return true;
            case R.id.itm_upload:

                  Intent intentUpload = new Intent(MainActivity.this,
                              UploadView.class);

                  startActivity(intentUpload);

                  return true;
            case R.id.itm_about:

                  Intent intentAbout = new Intent(MainActivity.this,
About.class);

                  startActivity(intentAbout);

                  return true;
            case R.id.itm_home:

                  Intent intentHome = new Intent(MainActivity.this,
HomeView.class);

                  startActivity(intentHome);

                  return true;
            case R.id.itm_settings:

                  Intent intentSettings = new Intent(MainActivity.this,
SettingsFirstView.class);

                  startActivity(intentSettings);



                  return true;
```

```
            default:
                  return super.onOptionsItemSelected(item);
            }
      }

      public void currentTab() {
            mTabHost.setCurrentTab(2);
      }

}
```

**HomeView.java**

```
package com.ucy.ireader;



import java.io.File;
import java.util.Date;

import com.ucy.ireader.sqlite.DataBaseHelper;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Matrix;
import android.graphics.drawable.BitmapDrawable;

import android.os.Bundle;
import android.os.Environment;
import android.view.Gravity;
import android.view.Menu;
import android.view.View;
import android.view.View.MeasureSpec;
import android.view.View.OnClickListener;
import android.view.ViewGroup;

import android.widget.Button;
import android.widget.ImageView.ScaleType;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.RelativeLayout.LayoutParams;
import android.widget.Spinner;
import android.widget.TabHost;
import android.widget.TextView;
```

```
import android.widget.ImageView;
import android.widget.Toast;
import android.widget.ViewFlipper;


public class HomeView extends Activity{

      Button btn_back;
      ;
      int mFlipping = 0 ; // Initially flipping is off
    Button mButton ;

      public void onCreate(Bundle savedInstanceState){
            super.onCreate(savedInstanceState);

          setContentView(R.layout.home);


            /** Click event handler for button */
        OnClickListener listener = new OnClickListener() {

            @Override
            public void onClick(View v) {
                ViewFlipper flipper = (ViewFlipper)
findViewById(R.id.flipper1);

                if(mFlipping==0){
                    /** Start Flipping */
                    flipper.startFlipping();
                    mFlipping=1;
                    mButton.setText(R.string.str_btn_stop);
                }
                else{
                    /** Stop Flipping */
                    flipper.stopFlipping();
                    mFlipping=0;
                    mButton.setText(R.string.str_btn_start);
                }
            }
        };

        /** Getting a reference to the button available in the
resource */
        mButton = (Button) findViewById(R.id.btn);

        /** Setting click event listner for the button */
        mButton.setOnClickListener(listener);

    }
```

```
//OnStart Method
      public void onStart() {
            super.onStart();
      }

      // On restart method
      public void OnRestart() {
            super.onRestart();
      }

      // OnResume Function
      public void OnResume() {
            super.onResume();
      }

      // OnPause function
      public void OnPause() {
            super.onPause();
      }

      // OnStop function
      public void OnStop() {
            super.onStop();
      }

      // On Destroy Method
      protected void onDestroy() {
            super.onDestroy();
      }
}
```

**About.java**

```
package com.ucy.ireader;



import java.util.Date;

import com.ucy.ireader.sqlite.DataBaseHelper;

import android.app.Activity;
import android.content.Intent;

import android.os.Bundle;
import android.text.Html;
```

```
import android.text.method.LinkMovementMethod;
import android.view.View;

import android.widget.Button;
import android.widget.Spinner;
import android.widget.TabHost;
import android.widget.TextView;



public class About extends Activity{

      Button btn_back;
      TextView about;
      TextView link;
      public void onCreate(Bundle savedInstanceState){
            super.onCreate(savedInstanceState);
            setContentView(R.layout.about_main);

            about= (TextView) findViewById(R.id.textView1);
            link = (TextView) findViewById(R.id.textView2);
            link.setClickable(true);
            link.setText(Html.fromHtml(getString(R.string.my_link)));
            link.setMovementMethod(LinkMovementMethod.getInstance());







      }


//OnStart Method
      public void onStart() {
            super.onStart();
      }

      // On restart method
      public void OnRestart() {
            super.onRestart();
      }

      // OnResume Function
      public void OnResume() {
            super.onResume();
      }

      // OnPause function
      public void OnPause() {
            super.onPause();
      }
```

```
	// OnStop function
	public void OnStop() {
		super.onStop();
	}

	// On Destroy Method
	protected void onDestroy() {
		super.onDestroy();
	}
}
```

**SearchView.java**

```
package com.ucy.ireader;

import java.util.ArrayList;
import java.util.Date;

import android.app.Activity;
import android.app.ActivityGroup;
import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;

public class SearchView extends ActivityGroup {
	EditText title;
	EditText year;
	EditText author;
	Spinner category;

	Button btn_search;
	Button btn_cancel;

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

		category = (Spinner) findViewById(R.id.list_category);
		title = (EditText) findViewById(R.id.txt_title);
		year = (EditText) findViewById(R.id.txt_year);
		author = (EditText) findViewById(R.id.txt_author);
```

```
            btn_search = (Button) findViewById(R.id.btn_search);
//          btn_cancel = (Button) findViewById(R.id.btn_cancel);

            ArrayList<Category> dataSource = new ArrayList<Category>();

            Cursor t = MainActivity.DatabaseHelper.select("SELECT
id,name FROM category");

            dataSource.add(new Category("0", "All"));

            while (t.moveToNext()) {
                  dataSource.add(new Category(t.getString(0),
t.getString(1)));
            }

            ArrayAdapter<Category> adapter = new
ArrayAdapter<Category>(this,
                        android.R.layout.simple_spinner_item,
dataSource);


      adapter.setDropDownViewResource(android.R.layout.simple_spinner_d
ropdown_item);
            category.setAdapter(adapter);

            btn_search.setOnClickListener(new View.OnClickListener() {
                  @Override
                  public void onClick(View v) {


                        Intent resultsView = new Intent(v.getContext(),
ResultsView.class);

                        resultsView.putExtra("category",
((Category)category.getItemAtPosition(category.getSelectedItemPosition
())).getId());
                        resultsView.putExtra("title",
title.getText().toString());
                        resultsView.putExtra("year",
year.getText().toString());
                        resultsView.putExtra("author",
author.getText().toString());

                        replaceContentView("ResultsView", resultsView);

                  }
            });


      }
```

```
	public void replaceContentView(String id, Intent newIntent) {
		View view =
getLocalActivityManager().startActivity(id,newIntent.addFlags(Intent.F
LAG_ACTIVITY_CLEAR_TOP)) .getDecorView();
		this.setContentView(view);
		}


	// OnStart Method
	public void onStart() {
		super.onStart();
	}

	// On restart method
	public void OnRestart() {
		super.onRestart();
	}

	// OnResume Function
	public void OnResume() {
		super.onResume();
	}

	// OnPause function
	public void OnPause() {
		super.onPause();
	}

	// OnStop function
	public void OnStop() {
		super.onStop();
	}

	// On Destroy Method
	protected void onDestroy() {
		super.onDestroy();
	}
}
```

**ResultsView.java**

```
package com.ucy.ireader;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
```

```
import java.io.OutputStream;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;

import com.ucy.ireader.sqlite.DataBaseHelper;


import android.app.ActivityGroup;
import android.app.ListActivity;

import android.content.Context;
import android.content.Intent;
import android.database.Cursor;
import android.database.SQLException;
import android.graphics.drawable.Drawable;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.util.Log;
import android.view.KeyEvent;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.Toast;

public class ResultsView extends ListActivity {
      private int per_page = 50;
      private int index = 0;

      private int total_results = 0;

      private String category = "";
      private String title = "";
      private String year = "";
      private String author = "";

      private ArrayList<Book> books = new ArrayList<Book>();
      BookAdapter adapter;

      public DataBaseHelper DatabaseHelper;



      public static int getImageId(Context context, String imageName) {
            return context.getResources().getIdentifier("drawable/" +
imageName,
                        null, context.getPackageName());
```

```
	}

	@Override
	public void onCreate(Bundle icicle) {
		super.onCreate(icicle);

		ListView listView = getListView();

		Drawable background =
getResources().getDrawable(R.drawable.back);

		//set background to Drawable
		listView.setBackgroundDrawable(background);

		Bundle extras = getIntent().getExtras();
		category = extras.getString("category");
		title = extras.getString("title");
		year = extras.getString("year");
		author = extras.getString("author");

		DatabaseHelper = new DataBaseHelper(this);

		try {
			if (DatabaseHelper.is_open())
				DatabaseHelper.close();

			DatabaseHelper.openDataBase();
			Log.e(">>>>>>", "Open SQL Database");
		} catch (SQLException sqle) {
			throw sqle;
		}

		find_results();

	}

	public boolean onCreateOptionsMenu(Menu menu) {
		menu.add(0, 0, 0, "Prev " + per_page);
		menu.add(0, 1, 0, "Next " + per_page);

		return true;
	}

	public boolean onOptionsItemSelected(MenuItem item) {

		switch (item.getItemId()) {
		case 0:
			/* Actions in case that Search is pressed */
			if (index > 0) {
				index = index - 1;
				find_results();
```

```
                }
                return true;
            case 1:
                /* Actions in case that Search is pressed */
                if ((index * per_page) + per_page < total_results) {
                    index = index + 1;
                    find_results();
                }
                return true;
            }

            return false;

        }

        private void find_results() {
            // Clear List
            books.clear();

            String sql = "SELECT
id,title,author,year_release,path,category_fk,picture FROM book";
            String sql_rows = "SELECT COUNT(*) FROM book";

            String where = "";
            String and = "";

            if (!category.equals("0")) {
                where += " category_fk=" + category;
                and = " AND ";
            }

            if (!title.equals("")) {
                where += and + " UPPER(title) LIKE '" +
title.toUpperCase() + "%'";
                and = " AND ";
            }

            if (!author.equals("")) {
                where += and + " UPPER(author) LIKE '" +
author.toUpperCase()
                        + "%'";
                and = " AND ";
            }

            if (!year.equals("")) {
                where += and + " year_release=" + year;
                and = " AND ";
            }

            if (!where.equals("")) {
                sql += " WHERE " + where;
```

```
                sql_rows += " WHERE " + where;
          }

          sql += " LIMIT " + (index * per_page) + "," + per_page;

          // DatabaseHelper
          Cursor t = DatabaseHelper.select(sql);
          Cursor t_rows = DatabaseHelper.select(sql_rows);

          if (t_rows.moveToFirst()) {
                total_results = t_rows.getInt(0);
          } else {
                total_results = 0;
          }

          while (t.moveToNext()) {
                books.add(new Book(t.getString(0), t.getString(1),
t.getString(2),
                            t.getString(3), t.getString(4),
t.getString(5),getResources().getIdentifier( t.getString(6),
"drawable",getPackageName())));
          }

          if (books.size() == 0) {
                books.add(new Book("0", "No Results", "", "", "", "",
-1));
          }

          Collections.sort(books,new Comparator<Book>() {
                public int compare(Book book1, Book book2) {
                      return
book1.getCategory_fk().compareTo(book2.getCategory_fk());
                }
          });

          if (category.equals("0")) {
                ArrayList<Book> books2 = new ArrayList<Book>();

                for(int i = 0; i < books.size(); i++)
                {
                      if(i == 0)
                      {
                            Cursor t_cat =
MainActivity.DatabaseHelper.select("SELECT name FROM category WHERE
id=" + books.get(i).getCategory_fk());

                            if (t_cat.moveToFirst()) {
                                  books2.add(new Book("0",
t_cat.getString(0),"", "","", "",-1));
                            } else {
                                  books2.add(new Book("0",
```

```
books.get(i).getCategory_fk(),"", "","", "",-1));
                              }

                              books2.add(books.get(i));
                              continue;
                        }


      if(!books.get(i).getCategory_fk().equals(books.get(i -
1).getCategory_fk()))
                        {
                              Cursor t_cat =
MainActivity.DatabaseHelper.select("SELECT name FROM category WHERE
id=" + books.get(i).getCategory_fk());

                              if (t_cat.moveToFirst()) {
                                    books2.add(new Book("0",
t_cat.getString(0),"", "","", "",-1));
                              } else {
                                    books2.add(new Book("0",
books.get(i).getCategory_fk(),"", "","", "",-1));
                              }

                              books2.add(books.get(i));
                        }
                        else
                              books2.add(books.get(i));
                  }

                  adapter = new BookAdapter(this,
R.layout.listview_item_row, books2);
            }
            else
            {
                  adapter = new BookAdapter(this,
R.layout.listview_item_row, books);
            }
            // Create an ArrayAdapter, that will actually make the
Strings above
            // appear in the ListView
            setListAdapter(adapter);


      }

      @Override
      protected void onListItemClick(ListView l, View v, int position,
long id) {
            super.onListItemClick(l, v, position, id);
            if (books.size() == 0) {
                  Toast.makeText(this,
```

```
                              "There are not any Results! Press Back
Button",
                              Toast.LENGTH_LONG).show();
                    return;
               }

               // Get the item that was clicked
               Book o = (Book) this.getListAdapter().getItem(position);

               if(o.getPicture() == -1)
               {
                    return;
               }

               String keyword = o.toString();
               Toast.makeText(this, "You selected: " + keyword,
Toast.LENGTH_LONG)
               .show();

               InputStream is;
               try {
                    if(!o.getPath().contains("/"))
                         is = getAssets().open(o.getPath());
                    else{
                         File myFile = new File(o.getPath());
                         is = new FileInputStream(myFile);
                    }
                    File cacheDirectory;

                    if
(android.os.Environment.getExternalStorageState().equals(
                              android.os.Environment.MEDIA_MOUNTED))
                         cacheDirectory = new File(

          android.os.Environment.getExternalStorageDirectory(),
                                   "iReader/");
                    else
                         cacheDirectory = this.getCacheDir();

                    if (!cacheDirectory.exists())
                         cacheDirectory.mkdirs();



                    File file = new File(cacheDirectory,
o.getPath().substring(o.getPath().lastIndexOf("/") + 1));

                    OutputStream out = new FileOutputStream(file);
                    byte buf[] = new byte[1024];
                    int len;
                    while ((len = is.read(buf)) > 0)
```

```
                    out.write(buf, 0, len);
                out.close();
                is.close();

                Uri path = Uri.fromFile(file);
                System.out.println("File Path:" +
file.getCanonicalPath());

                Intent intent = new Intent(Intent.ACTION_VIEW);
                intent.setDataAndType(path, "application/pdf");
                intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                startActivity(intent);


           } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
           }

     }

     @Override
     public void onBackPressed(){
           Intent intentSearch = new Intent(ResultsView.this,
                      SearchView.class);

           SearchView m = (SearchView)getParent();
           m.replaceContentView("SearchView", intentSearch);


     }


}
```

**UploadView.java**

```
package com.ucy.ireader;

import java.io.File;
import java.util.ArrayList;
import java.util.Date;

import android.app.Activity;
import android.content.ContentValues;
import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
import android.os.Environment;
```

```
import android.util.Log;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

public class UploadView extends Activity {
      EditText title;
      EditText year;
      EditText author;
      EditText pdf;
      Spinner category;
      Button btn_upload;
      Button btn_browse;

      UploadView self;
      @Override
      public void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            self=this;

            setContentView(R.layout.upload_main);

            category = (Spinner) findViewById(R.id.list_category_up);
            title = (EditText) findViewById(R.id.txt_title_up);
            year = (EditText) findViewById(R.id.txt_year_up);
            author = (EditText) findViewById(R.id.txt_author_up);
            pdf = (EditText) findViewById(R.id.txt_pdf_up);

            btn_upload = (Button) findViewById(R.id.btn_upload);
            btn_browse = (Button) findViewById(R.id.btn_browse);


            ArrayList<Category> dataSource = new ArrayList<Category>();

            Cursor t = MainActivity.DatabaseHelper.select("SELECT
id,name FROM category");

            while (t.moveToNext()) {
                  dataSource.add(new Category(t.getString(0),
t.getString(1)));
            }

            ArrayAdapter<Category> adapter = new
ArrayAdapter<Category>(this,
                        android.R.layout.simple_spinner_item,
dataSource);
```

```
     adapter.setDropDownViewResource(android.R.layout.simple_spinner_d
ropdown_item);
           category.setAdapter(adapter);

           btn_upload.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                     ContentValues values=new ContentValues();

                     values.put("category_fk",
((Category)category.getItemAtPosition(category.getSelectedItemPosition
())).getId());
                     values.put("title", title.getText().toString());
                     values.put("year_release",
year.getText().toString());
                     values.put("author",
author.getText().toString());

                     values.put("path", pdf.getText().toString());

                     values.put("picture", "book_1");

                     MainActivity.DatabaseHelper.insert("book",
values);



                     Toast.makeText(self, "Upload was Successful!",
                               Toast.LENGTH_LONG).show();

                     category.setSelection(0, true);
                     title.setText("");
                     year.setText("");
                     author.setText("");
                     pdf.setText("");
                }
           });



           btn_browse.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {

                     File mPath = new
File(Environment.getExternalStorageDirectory() + "//DIR//");
                   FileDialog fileDialog = new
FileDialog(UploadView.this, mPath);
                   fileDialog.setFileEndsWith(".pdf");
```

```
                    fileDialog.addFileListener(new
FileDialog.FileSelectedListener() {
                        public void fileSelected(File file) {
                            Log.d(getClass().getName(), "selected file
" + file.toString());
                            pdf.setText(file.toString());
                        }
                    });

                    fileDialog.showDialog();


                }
            });
      }

      // OnStart Method
      public void onStart() {
            super.onStart();
      }

      // On restart method
      public void OnRestart() {
            super.onRestart();
      }

      // OnResume Function
      public void OnResume() {
            super.onResume();
      }

      // OnPause function
      public void OnPause() {
            super.onPause();
      }

      // OnStop function
      public void OnStop() {
            super.onStop();
      }

      // On Destroy Method
      protected void onDestroy() {
            super.onDestroy();
      }
}
```

**SettingsView.java**

```
package com.ucy.ireader;

import android.app.Activity;
import android.app.ActivityGroup;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.SeekBar;
import android.widget.TabHost;
import android.widget.TextView;

public class SettingsView extends ActivityGroup {

       float BackLightValue = 0.5f; //dummy default value
       Button btn_back;
     @Override
     public void onCreate(Bundle savedInstanceState) {
         super.onCreate(savedInstanceState);

         setContentView(R.layout.settings);
         btn_back= (Button) findViewById(R.id.button2);

         SeekBar BackLightControl =
(SeekBar)findViewById(R.id.backlightcontrol);
         final TextView BackLightSetting =
(TextView)findViewById(R.id.backlightsetting);
         Button UpdateSystemSetting =
(Button)findViewById(R.id.updatesystemsetting);



         UpdateSystemSetting.setOnClickListener(new
Button.OnClickListener(){

     @Override
     public void onClick(View arg0) {
      int SysBackLightValue = (int)(BackLightValue * 255);
      android.provider.Settings.System.putInt(getContentResolver(),
        android.provider.Settings.System.SCREEN_BRIGHTNESS,
        SysBackLightValue);

     }});
     BackLightControl.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener(){

     @Override
     public void onProgressChanged(SeekBar arg0, int arg1, boolean
```

```
arg2) {
       // TODO Auto-generated method stub
       BackLightValue = (float)arg1/100;
       BackLightSetting.setText(String.valueOf(BackLightValue));

       WindowManager.LayoutParams layoutParams =
getWindow().getAttributes();
       layoutParams.screenBrightness = BackLightValue;
       getWindow().setAttributes(layoutParams);
      }
      @Override
      public void onStartTrackingTouch(SeekBar arg0) {
        arg0.setSecondaryProgress(arg0.getProgress()); // set the shade
of the previous value.

      }
      @Override

      public void onStopTrackingTouch(SeekBar arg0) {
      }});
      btn_back.setOnClickListener(new View.OnClickListener() {
                  @Override
                  public void onClick(View v) {
                        Intent intentSettingsView = new
Intent(v.getContext(), SettingsFirstView.class);

                        StringBuffer urlString = new StringBuffer();
                        replaceContentView("SettingsFirstView",
intentSettingsView);

                  }
            });
       }
      public void replaceContentView(String id, Intent newIntent) {
            View view =
getLocalActivityManager().startActivity(id,newIntent.addFlags(Intent.F
LAG_ACTIVITY_CLEAR_TOP)) .getDecorView(); this.setContentView(view);
            }

      }
```

**SettingsFirstView.java**

```
package com.ucy.ireader;

import java.util.Date;

import com.ucy.ireader.sqlite.DataBaseHelper;
```

```
import android.app.Activity;
import android.app.ActivityGroup;
import android.content.Intent;

import android.os.Bundle;
import android.text.Html;
import android.text.method.LinkMovementMethod;
import android.view.View;

import android.widget.Button;
import android.widget.Spinner;
import android.widget.TabHost;
import android.widget.TextView;

public class SettingsFirstView extends ActivityGroup {

      Button btn_brightness;
      Button btn_send_email;
      Button btn_set_fontsize;
      public void onCreate(Bundle savedInstanceState){
            super.onCreate(savedInstanceState);
            setContentView(R.layout.settings_first_view);



            btn_brightness= (Button) findViewById(R.id.button1);
            btn_send_email = (Button) findViewById(R.id.button3);


            btn_brightness.setOnClickListener(new
View.OnClickListener() {
                  @Override
                  public void onClick(View v) {
                        Intent intentSettingsView = new
Intent(v.getContext(), SettingsView.class);

                        StringBuffer urlString = new StringBuffer();
                        replaceContentView("SettingsView",
intentSettingsView);

                  }
            });

            btn_send_email.setOnClickListener(new
View.OnClickListener() {
                  @Override
                  public void onClick(View v) {
                        Intent email = new Intent(Intent.ACTION_SEND);
                        email.setType("message/rfc822");
                        email.putExtra(Intent.EXTRA_EMAIL ,new
```

```
String[]{"georgiandreoucs@gmail.com"});
                        email.putExtra(Intent.EXTRA_SUBJECT, "Comments");
                        try {

                              startActivity(Intent.createChooser(email,
"Send mail..."));

                        } catch
(android.content.ActivityNotFoundException ex) {
                        }
                  }
            });


      }
      public void replaceContentView(String id, Intent newIntent) {
            View view =
getLocalActivityManager().startActivity(id,newIntent.addFlags(Intent.F
LAG_ACTIVITY_CLEAR_TOP)) .getDecorView(); this.setContentView(view);
            }


//OnStart Method
      public void onStart() {
            super.onStart();
      }

      // On restart method
      public void OnRestart() {
            super.onRestart();
      }

      // OnResume Function
      public void OnResume() {
            super.onResume();
      }

      // OnPause function
      public void OnPause() {
            super.onPause();
      }

      // OnStop function
      public void OnStop() {
            super.onStop();
      }

      // On Destroy Method
      protected void onDestroy() {
            super.onDestroy();
      }
```

```
}
```

**FileDialog.java**

```
package com.ucy.ireader;

import java.io.*;
import java.util.*;

import com.ucy.ireader.ListenerList.FireHandler;

import android.app.*;
import android.content.DialogInterface;
import android.content.DialogInterface.OnClickListener;
import android.os.Environment;
import android.util.Log;


public class FileDialog {
    private static final String PARENT_DIR = "..";
    private final String TAG = getClass().getName();
    private String[] fileList;
    private File currentPath;
    public interface FileSelectedListener {
        void fileSelected(File file);
    }
    public interface DirectorySelectedListener {
        void directorySelected(File directory);
    }
    private ListenerList<FileSelectedListener> fileListenerList = new
ListenerList<FileDialog.FileSelectedListener>();
    private ListenerList<DirectorySelectedListener> dirListenerList =
new ListenerList<FileDialog.DirectorySelectedListener>();
    private final Activity activity;
    private boolean selectDirectoryOption;
    private String fileEndsWith;

    /**
     * @param activity
     * @param initialPath
     */
    public FileDialog(Activity activity, File path) {
        this.activity = activity;
        if (!path.exists()) path =
Environment.getExternalStorageDirectory();
        loadFileList(path);
    }
```

```
    /**
     * @return file dialog
     */
    public Dialog createFileDialog() {
        Dialog dialog = null;
        AlertDialog.Builder builder = new
AlertDialog.Builder(activity);

        builder.setTitle(currentPath.getPath());
        if (selectDirectoryOption) {
            builder.setPositiveButton("Select directory", new
OnClickListener() {
                public void onClick(DialogInterface dialog, int which)
{
                    Log.d(TAG, currentPath.getPath());
                    fireDirectorySelectedEvent(currentPath);
                }
            });
        }

        builder.setItems(fileList, new
DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                String fileChosen = fileList[which];
                File chosenFile = getChosenFile(fileChosen);
                if (chosenFile.isDirectory()) {
                    loadFileList(chosenFile);
                    dialog.cancel();
                    dialog.dismiss();
                    showDialog();
                } else fireFileSelectedEvent(chosenFile);
            }
        });

        dialog = builder.show();
        return dialog;
    }


    public void addFileListener(FileSelectedListener listener) {
        fileListenerList.add(listener);
    }

    public void removeFileListener(FileSelectedListener listener) {
        fileListenerList.remove(listener);
    }

    public void setSelectDirectoryOption(boolean
selectDirectoryOption) {
        this.selectDirectoryOption = selectDirectoryOption;
```

```
    }

    public void addDirectoryListener(DirectorySelectedListener
listener) {
        dirListenerList.add(listener);
    }

    public void removeDirectoryListener(DirectorySelectedListener
listener) {
        dirListenerList.remove(listener);
    }

    /**
     * Show file dialog
     */
    public void showDialog() {
        createFileDialog().show();
    }

    private void fireFileSelectedEvent(final File file) {
        fileListenerList.fireEvent(new
FireHandler<FileDialog.FileSelectedListener>() {
            public void fireEvent(FileSelectedListener listener) {
                listener.fileSelected(file);
            }
        });
    }

    private void fireDirectorySelectedEvent(final File directory) {
        dirListenerList.fireEvent(new
FireHandler<FileDialog.DirectorySelectedListener>() {
            public void fireEvent(DirectorySelectedListener listener)
{
                listener.directorySelected(directory);
            }
        });
    }

    private void loadFileList(File path) {
        this.currentPath = path;
        List<String> r = new ArrayList<String>();
        if (path.exists()) {
            if (path.getParentFile() != null) r.add(PARENT_DIR);
            FilenameFilter filter = new FilenameFilter() {
                public boolean accept(File dir, String filename) {
                    File sel = new File(dir, filename);
                    if (!sel.canRead()) return false;
                    if (selectDirectoryOption) return
sel.isDirectory();
                    else {
                        boolean endsWith = fileEndsWith != null ?
```

```
filename.toLowerCase().endsWith(fileEndsWith) : true;
                        return endsWith || sel.isDirectory();
                    }
                }
            };
            String[] fileList1 = path.list(filter);
            for (String file : fileList1) {
                r.add(file);
            }
        }
        fileList = (String[]) r.toArray(new String[]{});
    }

    private File getChosenFile(String fileChosen) {
        if (fileChosen.equals(PARENT_DIR)) return
currentPath.getParentFile();
        else return new File(currentPath, fileChosen);
    }

    public void setFileEndsWith(String fileEndsWith) {
        this.fileEndsWith = fileEndsWith != null ?
fileEndsWith.toLowerCase() : fileEndsWith;
    }
 }

class ListenerList<L> {
private List<L> listenerList = new ArrayList<L>();

public interface FireHandler<L> {
    void fireEvent(L listener);
}

public void add(L listener) {
    listenerList.add(listener);
}

public void fireEvent(FireHandler<L> fireHandler) {
    List<L> copy = new ArrayList<L>(listenerList);
    for (L l : copy) {
        fireHandler.fireEvent(l);
    }
}

public void remove(L listener) {
    listenerList.remove(listener);
}

public List<L> getListenerList() {
    return listenerList;
}
}
```

**DialogView.java**

```
package com.ucy.ireader;

import java.io.*;

import android.app.Activity;
import android.os.*;
import android.util.Log;

public class DialogView extends Activity {
     protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        File mPath = new
File(Environment.getExternalStorageDirectory() + "//DIR//");
        FileDialog fileDialog = new FileDialog(this, mPath);
        fileDialog.setFileEndsWith(".pdf");
        fileDialog.addFileListener(new
FileDialog.FileSelectedListener() {
            public void fileSelected(File file) {
                Log.d(getClass().getName(), "selected file " +
file.toString());
            }
        });

        fileDialog.showDialog();
    }
}
```

**Category.java**

```
package com.ucy.ireader;

public class Category {
     private String id;
     private String name;


     public Category(String id_v,String name_v){
          id=id_v;
          name=name_v;
     }
```

```
	public String getId() {
		return id;
	}


	public void setId(String id) {
		this.id = id;
	}


	public String getName() {
		return name;
	}


	public void setName(String name) {
		this.name = name;
	}

	public String toString(){
		return name;
	}
}
```

**BookAdapter.java**

```
package com.ucy.ireader;

import java.util.ArrayList;

import android.app.Activity;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.TextView;

public class BookAdapter extends ArrayAdapter<Book>{

    Context context;
    int layoutResourceId;
    ArrayList<Book>data = null;

    public BookAdapter(Context context, int layoutResourceId,
ArrayList<Book> data) {
```

```
        super(context, layoutResourceId, data);
        this.layoutResourceId = layoutResourceId;
        this.context = context;
        this.data = data;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup
parent) {
        View row = convertView;
        BookHolder holder = null;

        if(row == null)
        {
           Book book = data.get(position);

           if(book != null)
            {
                if(book.getPicture() == -1)
                {
                     LayoutInflater inflater =
((Activity)context).getLayoutInflater();
                    row =
inflater.inflate(R.layout.listview_item_category, parent, false);

                    row.setOnClickListener(null);
                    row.setOnLongClickListener(null);
                    row.setLongClickable(false);

                    final TextView sectionView = (TextView)
row.findViewById(R.id.listview_item_category_text);
                    sectionView.setText(book.toString());
                }
                else
                {
                     LayoutInflater inflater =
((Activity)context).getLayoutInflater();
                    row = inflater.inflate(R.layout.listview_item_row,
parent, false);

                    holder = new BookHolder();
                    holder.imgIcon =
(ImageView)row.findViewById(R.id.imgIcon);
                    holder.txtTitle =
(TextView)row.findViewById(R.id.txtTitle);

                    row.setTag(holder);

                    holder.txtTitle.setText(book.toString());

holder.imgIcon.setImageResource(book.getPicture());
```

```
                }
            }
        }
        else
        {
            holder = (BookHolder)row.getTag();
        }

        return row;
    }

    static class BookHolder
    {
        ImageView imgIcon;
        TextView txtTitle;
    }
}
```

**Book.java**

```
package com.ucy.ireader;

public class Book {

      private String id;
      private String title;
      private String author;
      private String year;
      private String category_fk;
      private String path;
      private int picture;

      public Book(String id,String title,String author,String
year,String path,String category_fk,int picture){
            this.id=id;
            this.title=title;
            this.author=author;
            this.year=year;
            this.category_fk=category_fk;
            this.path=path;
            this.picture=picture;
      }

      public int getPicture() {
            return picture;
      }

      public void setPicture(int picture) {
            this.picture = picture;
```

```
	}

	public String toString(){
		if (!year.equals(""))
			return title+" / "+year;

		return title;
	}

	public String getId() {
		return id;
	}
	public void setId(String id) {
		this.id = id;
	}
	public String getTitle() {
		return title;
	}
	public void setTitle(String title) {
		this.title = title;
	}
	public String getAuthor() {
		return author;
	}
	public void setAuthor(String author) {
		this.author = author;
	}
	public String getYear() {
		return year;
	}
	public void setYear(String year) {
		this.year = year;
	}
	public String getCategory_fk() {
		return category_fk;
	}
	public void setCategory_fk(String category_fk) {
		this.category_fk = category_fk;
	}
	public String getPath() {
		return path;
	}
	public void setPath(String path) {
		this.path = path;
	}

	public int compareTo(Book text) {
		// TODO Auto-generated method stub
		return 0;
	}
```

```
}
```

**DatabaseHelper.java**

```
package com.ucy.ireader.sqlite;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteException;
import android.database.sqlite.SQLiteOpenHelper;

public class DataBaseHelper extends SQLiteOpenHelper {

      // The Android's default system path of your application
database.
      private static String DB_PATH;
      private static String DB_NAME = "sql_values.db";
      private SQLiteDatabase myDataBase;
      private final Context myContext;

      /**
       * Constructor Takes and keeps a reference of the passed context
in order to
       * access to the application assets and resources.
       *
       * @param context
       */
      public DataBaseHelper(Context context) {

            super(context, DB_NAME, null, 1);
            this.myContext = context;
            DB_PATH = "/data/data/" + this.myContext.getPackageName()
                        + "/databases/";
      }

      /**
```

```
       * Creates a empty database on the system and rewrites it with
your own
       * database.
       * */
      public void createDataBase() throws IOException {

            boolean dbExist = checkDataBase();

            if (dbExist) {
                  // do nothing - database already exist
            } else {

                  // By calling this method and empty database will be
created into
                  // the default system path
                  // of your application so we are gonna be able to
overwrite that
                  // database with our database.
                  this.getReadableDatabase();

                  try {
                        copyDataBase();
                  } catch (IOException e) {
                        throw new Error("Error copying database");
                  }
            }

      }

      /**
       * Check if the database already exist to avoid re-copying the
file each
       * time you open the application.
       *
       * @return true if it exists, false if it doesn't
       */
      private boolean checkDataBase() {

            SQLiteDatabase checkDB = null;

            try {
                  String myPath = DB_PATH + DB_NAME;
                  checkDB = SQLiteDatabase.openDatabase(myPath, null,
                              SQLiteDatabase.OPEN_READONLY);
            } catch (SQLiteException e) {
                  // database does't exist yet.
            }

            if (checkDB != null) {
                  checkDB.close();
            }
```

```
            return checkDB != null ? true : false;
      }

      /**
       * Copies your database from your local assets-folder to the just
created
       * empty database in the system folder, from where it can be
accessed and
       * handled. This is done by transfering bytestream.
       * */
      private void copyDataBase() throws IOException {

            // Open your local db as the input stream
            InputStream myInput = myContext.getAssets().open(DB_NAME);

            // Path to the just created empty db
            String outFileName = DB_PATH + DB_NAME;

            // Open the empty db as the output stream
            OutputStream myOutput = new FileOutputStream(outFileName);

            // transfer bytes from the inputfile to the outputfile
            byte[] buffer = new byte[1024];
            int length;
            while ((length = myInput.read(buffer)) > 0) {
                  myOutput.write(buffer, 0, length);
            }

            // Close the streams
            myOutput.flush();
            myOutput.close();
            myInput.close();
      }

      public void drop_database(){
            // Path to the just created empty db
            File file = new File( DB_PATH + DB_NAME);
            file.delete();
      }

      public void openDataBase() throws SQLException {

            // Open the database
            String myPath = DB_PATH + DB_NAME;
            myDataBase = SQLiteDatabase.openDatabase(myPath, null,
                        SQLiteDatabase.OPEN_READWRITE);
      }

      @Override
      public synchronized void close() {
```

```
            if (myDataBase != null)
                  myDataBase.close();

            super.close();

      }

      @Override
      public void onCreate(SQLiteDatabase db) {

      }

      @Override
      public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {

      }

      /* Insert Function - Return the Row ID */
      public long insert(String table, ContentValues initialValues) {
            // TODO Auto-generated method stub
            return myDataBase.insert(table, null, initialValues);
      }

      /* Update Function - Return the number of affected Rows */
      public int update(String table, ContentValues updateValues,
                  String whereCondition) {
            // TODO Auto-generated method stub
            return myDataBase.update(table, updateValues,
whereCondition, null);
      }

      /* Delete Function - Return the number of deleted Rows */
      public int delete(String table, String whereCondition) {
            // TODO Auto-generated method stub
            return myDataBase.delete(table, whereCondition, null);
      }

      public Cursor select(String sql) {
            return myDataBase.rawQuery(sql, null);
      }

      public boolean is_open() {
            if (myDataBase == null || !myDataBase.isOpen())
                  return false;

            return true;
      }
}
```

**SQLField.java**

```
package com.ucy.ireader.sqlite;

public class SQLField {
      public String field;
      public String value;

      public SQLField(String Field,String Value){
            this.field=Field;
            this.value=Value;
      }

      public SQLField(){

      }

}
```

**about_main.xml**

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/back"
    tools:context=".MainActivity">

<TextView
    android:id="@+id/textView1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginLeft="60dp"
    android:layout_marginRight="60dp"
    android:layout_marginTop="80dp"
    android:text="@string/about"
    android:textColor="#FFFFFF"
    android:textSize="20dip"
    android:textStyle="italic" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
```

```
    android:layout_alignParentRight="true"
    android:layout_marginBottom="12dp"
    android:layout_marginRight="54dp"
    android:textColor="#FFFFFF"
    android:text="@string/designer"
    android:textAppearance="?android:attr/textAppearanceSmall" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/textView3"
    android:layout_alignParentLeft="true"
    android:layout_marginBottom="12dp"
    android:autoLink="web"
    android:clickable="true"
    android:text="@string/my_link"
    android:textColor="#FFFFFF"
    android:textSize="20dip"
    android:textStyle="italic" />

</RelativeLayout>
```

**activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<TabHost xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@android:id/tabhost"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:orientation="vertical" >

        <TabWidget
            android:id="@android:id/tabs"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content" />

        <FrameLayout
            android:id="@android:id/tabcontent"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent" >
        </FrameLayout>
    </LinearLayout>

</TabHost>
```

**home.xml**

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/back" >

    <Button
        android:id="@+id/btn"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/str_btn_start"
    />

    <ViewFlipper
        android:id="@+id/flipper1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/btn"
        android:flipInterval="3000"
        android:inAnimation="@android:anim/slide_in_left"
        android:outAnimation="@android:anim/slide_out_right" >

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_gravity="center_horizontal"
            android:adjustViewBounds="true"
            android:contentDescription="@string/str_img1"
            android:layout_marginTop="60dp"
            android:scaleType="fitCenter"
            android:src="@drawable/image1" />

        <ImageView
            android:src="@drawable/image2"
           android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:contentDescription="@string/str_img2"
            android:layout_marginTop="60dp"
            android:adjustViewBounds="true"
            android:scaleType="fitCenter"
            android:layout_gravity="center_horizontal"
        />
```

```
          <ImageView
            android:src="@drawable/image5"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:contentDescription="@string/str_img3"
            android:layout_marginTop="60dp"
            android:adjustViewBounds="true"
            android:scaleType="fitCenter"
            android:layout_gravity="center_horizontal"
        />

           <ImageView
            android:src="@drawable/image6"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:contentDescription="@string/str_img3"
            android:layout_marginTop="60dp"
            android:adjustViewBounds="true"
            android:scaleType="fitCenter"
            android:layout_gravity="center_horizontal"
        />

            <ImageView
            android:src="@drawable/image7"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:contentDescription="@string/str_img3"
            android:layout_marginTop="60dp"
            android:adjustViewBounds="true"
            android:scaleType="fitCenter"
            android:layout_gravity="center_horizontal"
        />



              <ImageView
            android:src="@drawable/image9"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:contentDescription="@string/str_img3"
            android:layout_marginTop="60dp"
            android:adjustViewBounds="true"
            android:scaleType="fitCenter"
            android:layout_gravity="center_horizontal"
        />

               <ImageView
            android:src="@drawable/image10"
                android:layout_width="wrap_content"
            android:layout_height="match_parent"
```

```
            android:contentDescription="@string/str_img3"
            android:layout_marginTop="60dp"
            android:adjustViewBounds="true"
            android:scaleType="fitCenter"
            android:layout_gravity="center_horizontal"
        />

                <ImageView
            android:src="@drawable/image11"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:contentDescription="@string/str_img3"
            android:layout_marginTop="60dp"
            android:adjustViewBounds="true"
            android:scaleType="fitCenter"
            android:layout_gravity="center_horizontal"
        />

                 <ImageView
            android:src="@drawable/image12"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:contentDescription="@string/str_img3"
            android:layout_marginTop="60dp"
            android:adjustViewBounds="true"
            android:scaleType="fitCenter"
            android:layout_gravity="center_horizontal"
        />

                  <ImageView
            android:src="@drawable/image13"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:contentDescription="@string/str_img3"
            android:layout_marginTop="60dp"
            android:adjustViewBounds="true"
            android:scaleType="fitCenter"
            android:layout_gravity="center_horizontal"
        />

    </ViewFlipper>


</RelativeLayout>
```

**listview_item_category.xml**

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout
      xmlns:android="http://schemas.android.com/apk/res/android"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:orientation="vertical">

      <TextView
          android:id="@+id/listview_item_category_text"
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          layout="@android:layout/preference_category"
          android:textColor="#FFFFFF"
          android:textSize="20sp"
          android:textStyle="italic"
          />

</LinearLayout>
```

**listview_item_row.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="10dp">



     <ImageView android:id="@+id/imgIcon"
        android:layout_width="wrap_content"
        android:layout_height="fill_parent"
        android:gravity="center_vertical"
        android:layout_alignParentTop="true"
        android:layout_alignParentBottom="true"
        android:layout_marginRight="15dp"
        android:layout_marginTop="5dp"
        android:layout_marginBottom="5dp" />

     <TextView android:id="@+id/txtTitle"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:gravity="center_vertical"
        android:layout_alignParentTop="true"
        android:layout_alignParentBottom="true"
        android:textStyle="bold"
        android:textSize="22dp"
        android:textColor="#FFFFFF"
```

```
        android:layout_marginTop="5dp"
        android:layout_marginBottom="5dp"
        android:minHeight="18dp"  />

</LinearLayout>
```

**search_main.xml**

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/back"
    tools:context=".MainActivity">


    <Button
        android:id="@+id/btn_search"
        android:layout_width="200dp"
        android:layout_height="60dp"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/txt_author"
        android:layout_marginTop="55dp"
        android:minHeight="0dp"
        android:minWidth="0dp"
        android:text="Search" />

    <TextView
        android:id="@+id/textView1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="14dp"
        android:text="Category"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <Spinner
        android:id="@+id/list_category"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/textView1" />

    <EditText
        android:id="@+id/txt_title"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
```

```
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/TextView01"
        android:ems="10" />

    <TextView
        android:id="@+id/TextView02"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/txt_title"
        android:layout_marginTop="17dp"
        android:text="Year"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <EditText
        android:id="@+id/txt_year"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_centerVertical="true"
        android:ems="10"
        android:inputType="number" />

    <TextView
        android:id="@+id/TextView03"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/txt_year"
        android:layout_marginTop="22dp"
        android:text="Author"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <EditText
        android:id="@+id/txt_author"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/TextView03"
        android:ems="10" />

    <TextView
        android:id="@+id/TextView01"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/list_category"
        android:layout_marginTop="16dp"
        android:text="Title"
        android:textAppearance="?android:attr/textAppearanceLarge" />
```

```
</RelativeLayout>
```

**settings_first_view.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/back"
    android:orientation="vertical" >

    <Button
        android:id="@+id/button1"
        android:layout_width="match_parent"
        android:layout_height="76dp"
        android:layout_marginTop="100dp"
        android:scaleType="fitCenter"
        android:adjustViewBounds="true"
        android:layout_gravity="center_horizontal"
        android:text="Set the Brightness" />

    <Button
        android:id="@+id/button3"
        android:layout_width="match_parent"
        android:layout_height="76dp"
        android:layout_marginTop="60dp"
        android:adjustViewBounds="true"
        android:scaleType="fitCenter"
        android:text="Send Email" />



</LinearLayout>
```

**settings.xml**

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

    <TextView
```

```
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="100dp"
        android:text="Set BackLight of the App"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <SeekBar
        android:id="@+id/backlightcontrol"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10px"
        android:max="100"
        android:progress="50" />

    <TextView
        android:id="@+id/backlightsetting"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
         android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="0.50" />

    <Button
        android:id="@+id/updatesystemsetting"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="100dp"
        android:gravity="center"
        android:text="Update SCREEN BRIGHTNESS" />

    <Button
        android:id="@+id/button2"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="70dp"
        android:gravity="center"
        android:text="Back in Settings" />

</LinearLayout>
```

**tabs_bg.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Copyright (c) 2010 Josh Clemm -->

<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/tabsLayout"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
```

```
    android:background="@drawable/tab_bg_selector"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="10dip" >

    <TextView
        android:id="@+id/tabsText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Title"
        android:textColor="@drawable/tab_text_selector"
        android:textSize="15dip" />

</LinearLayout>
```

**upload_main.xml**

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/back"
    tools:context=".MainActivity">

    <Spinner
        android:id="@+id/list_category_up"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="50dp" />

    <TextView
        android:id="@+id/textView1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="30dp"
        android:text="Category"
        android:textAppearance="?android:attr/textAppearanceMedium" />


    <EditText
        android:id="@+id/txt_title_up"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
```

```
        android:layout_below="@+id/TextView01"
        android:ems="10" />

    <EditText
        android:id="@+id/txt_year_up"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/TextView02"
        android:ems="10"
        android:inputType="number" />

    <EditText
        android:id="@+id/txt_author_up"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/TextView03"
        android:ems="10" >

        <requestFocus />
    </EditText>

    <EditText
        android:id="@+id/txt_pdf_up"
        android:layout_width="190dp"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/TextView04"
        android:layout_toLeftOf="@+id/btn_browse"
        android:ems="10" />

    <TextView
        android:id="@+id/TextView01"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/list_category_up"
        android:text="Title"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <TextView
        android:id="@+id/TextView02"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/txt_title_up"
        android:text="Year"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <TextView
```

```
        android:id="@+id/TextView03"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/txt_year_up"
        android:text="Author"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <TextView
        android:id="@+id/TextView04"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/txt_author_up"
        android:text="PDF"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <Button
        android:id="@+id/btn_upload"
        android:layout_width="200dp"
        android:layout_height="60dp"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:text="Upload" />

    <Button
        android:id="@+id/btn_browse"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/TextView04"
        android:text="Browse" />

</RelativeLayout>
```

**res/values/strings.xml**

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

    <string name="app_name">iReader</string>
    <string name="hello_world">Hello world!</string>
    <string name="menu_settings">Settings</string>
      <string name="about">iReader has designed to make your lives
easier. With this application you can choose the book you want and
read it just with this simple interface. Also, you can upload any
books you want, in the specific category so when you use again this
app, you \' ll find the book already registered. Enjoy!</string>
    <string name="welcome">WELCOME TO iReader</string>
    <color name="purple">#E3E4FA</color>
    <string name="str_img1">Image1</string>
```

```
    <string name="str_img2">Image2</string>
    <string name="str_img3">Image3</string>
    <string name="str_img4">Image4</string>
    <string name="str_img5">Image5</string>
    <string name="designer">Designer: Georgia Andreou</string>

    <string name="str_btn_start">Start SlideShow</string>
    <string name="str_btn_stop">Stop SlideShow</string>
    <string name="my_link">Like us on Facebook :  <![CDATA[<a
href="www.facebook.com/csiReader">www.facebook.com/csiReader</a>]]></s
tring>

     <string name="location">My path:</string>
</resources>
```

**AndroidManifest.xml**

```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.ucy.ireader"
    android:versionCode="1"
    android:versionName="1.0"
     >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17"
        />



    <uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:allowBackup="true"
        android:icon="@drawable/book_logo"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.ucy.ireader.MainActivity"
            android:label="@string/app_name" >

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

```
                <category
android:name="android.intent.category.LAUNCHER" />

            </intent-filter>
        </activity>
        <activity
android:name="com.ucy.ireader.SearchView"></activity>
        <activity
android:name="com.ucy.ireader.ResultsView"></activity>
        <activity
android:name="com.ucy.ireader.UploadView"></activity>
        <activity android:name="com.ucy.ireader.About"></activity>
        <activity android:name="com.ucy.ireader.HomeView"></activity>
        <activity
android:name="com.ucy.ireader.SettingsFirstView"></activity>
        <activity
android:name="com.ucy.ireader.SettingsView"></activity>
        <activity
android:name="com.ucy.ireader.FileDialog"></activity>
        <activity
android:name="com.ucy.ireader.DialogView"></activity>

    </application>

</manifest>
```

# ΒΙΒΛΙΟΓΡΑΦΙΑ - ΙΣΤΟΣΕΛΙΔΕΣ


- **Hello Android, Introducing Google's Mobile Development Platform, 3rd edition , Ed Burnette**
- **Professional Android Application Development, Reto Meier**
- **Android Developers, The Developer's Guide:** http://developer.android.com/guide/index.html
- **Google Groups, Android Developers:** https://groups.google.com/forum/?fromgroups#!forum/android-developers
- **Android Development Community:** http://www.anddev.org/
- **Code Project:** http://www.codeproject.com/
- **Android Operating System** https://en.wikipedia.org/wiki/Android_(operating_system)