ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΗΠΕΙΡΟΥ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

TECHNOLOGICAL EDUCATIONAL INSTITUTE OF EPIRUS
SCHOOL OF APPLIED TECHNOLOGY
DEPARTMENT OF COMPUTER ENGINEERING

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

"Ανάπτυξη εφαρμογής ασφαλείας για τον έλεγχο, περιορισμό και διαχείριση της διεπαφής χρήστη σε περιβάλλον Android"

Σπουδάστρια

Παρασκευή Μπαράκου ΑΜ-9100

Επιβλέπων Καθηγητής

Ιωάννης Τσούλος

ΠΡΟΛΟΓΟΣ

Η παρούσα πτυχιακή εργασία γράφτηκε από τη φοιτήτρια Παρασκευή-Βιβή Μπαράκου στα πλαίσια των σπουδών στο τμήμα Μηχανικών Πληροφορικής Τ.Ε της σχολής Τεχνολογικών Εφαρμογών του ΤΕΙ Ηπείρου. Η πτυχιακή εργασία εκπονήθηκε υπό την εποπτεία του καθηγητή κύριου Ιωάννη Τσούλο.

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου για τη καθοδήγηση και την φιλική συμπεριφορά του. Επιπλέον θα ήθελα να ευχαριστήσω την οικογένεια μου για τη συμπαράσταση και την στήριξη τους σε αυτή μου την προσπάθεια. Επί

ΠΕΡΙΛΗΨΗ

Το θέμα της εργασίας αφορά την ανάπτυξη μιας εφαρμογής σε περιβάλλον Android. Συγκεκριμένα , η εφαρμογή αφορά μια οθόνη κλειδώματος. Με την εγκατάσταση της εφαρμογής , ο χειριστής έχει τη δυνατότητα να επιλέξει αν θα γίνεται εκκίνηση με την οθόνη συστήματος ή με την οθόνη κλειδώματος (αυτή η επιλογή γίνεται μια φορά). Στην ουσία όταν γίνει αυτή η επιλογή , η εφαρμογή λειτουργεί σαν μια επιφάνεια εργασίας. Επιπλέον η εφαρμογή μας δίνει τη δυνατότητα να τερματίζονται αυτόματα οι εφαρμογές ή οι διεργασίες που θα επιλέξουμε.

Ο χειριστής θα έχει πρόσβαση (με κωδικό ασφαλείας) σε ένα κεντρικό μενού από όπου θα μπορεί :

- Να επιλέγει ποιες εφαρμογές θα εμφανίζονται στην οθόνη , χωρίς κάποιος άλλος χρήστης να μπορεί να έχει πρόσβαση σε κάποια άλλη εφαρμογή.
- Να απενεργοποιεί διεργασίες κ εφαρμογές που τρέχουν στη συσκευή , ώστε να επιτυγχάνεται εξοικονόμηση μπαταρίας.
- Να αλλάζει τον κωδικό ασφαλείας της εφαρμογής .
- Να έχει πρόσβαση σε όλες τις εγκατεστημένες εφαρμογές της συσκευής.

Η αρχική οθόνη κλειδώματος θα δίνει στο χρήστη την δυνατότητα να έχει πρόσβαση μόνο στις εφαρμογές που έχουν επιλεγεί , και δεν θα μπορεί να κάνει κάποια παραμετροποίηση αφού απαιτείται κωδικός πρόσβασης.
Σκοπός της εφαρμογής αυτής είναι :

- Εξοικονόμηση μπαταρίας
- Χρήση της εφαρμογής ως πρόγραμμα περιορισμένης πρόσβασης στις λειτουργίες του κινητού.
- Προστασία της συσκευής από μη εξουσιοδοτημένη πρόσβαση.
- Βελτιστοποίηση της συσκευής και εξοικονόμηση επεξεργαστικής ισχύς λόγω τερματισμού περιττών εφαρμογών.

Λέξεις – Κλειδιά : Android, Εκδόσεις Android , Αρχιτεκτονική Android, Eclipse, Ανάπτυξη Εφαρμογών Android

ABSTRACT

The subject of this diploma thesis is the development of an application within the Android Environment. Specifically, the application concerns a lock screen. After the installation of the application, the user has the ability to choose if the start up will be with the screen of the operating system or with the lock screen of the application (This choice is made when the application is started) . Substantially when the choice is given , the application works like a desktop. Furthermore, the application can automatically stop applications or processes which are selected.

The operator will have access (with a security password) to a main menu from where he can :

- Choose which applications will be appeared to the home screen.
- Disable processes or applications that run in the device.
- Change the security password.
- Access to all the installed applications of the device.

The main screen lets the user have access only to applications which are already chosen from the setting menu and cannot make any customizing as it demands an access password.

The goal of the application is :

- Battery saving.
- To be used as a program of constraint access of the device.
- Protection of the device of no authorized access
- Optimization of the device and gain of the processing power because of the termination of excess applications.

Keywords : Android, Android versions , Αρχιτεκτονική Android, Eclipse, Android development

Πίνακας περιεχομένων

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

1

Εισαγωγή στο λειτουργικό σύστημα Android

1.1 Περιγραφή του Android

Το Android είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας το οποίο τρέχει τον πυρήνα (kernel) του λειτουργικού Linux . Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance.[1] Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google.

Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, και από τότε έχει γίνει σημαντικό κομμάτι του τεχνολογικού κόσμου. Το Android έπαιξε πολύ σημαντικό ρόλο και αυτό στην εξέλιξη της τεχνολογίας, όπου πλέον κυκλοφορούν στην αγορά περισσότερες από 1 δισεκατομμύριο συσκευές smartphone και tablet .

Ο Όρος ανδροιδ έχει ελληνική προέλευση καθώς προέρχεται από τη λέξη ανδρ- που έχει την έννοια του άνδρα ή του ανθρώπου και την κατάληξη –ειδές που χαρακτηρίζει κάποιο είδος. Συνεπώς η έννοια που δίνεται στη λέξη android είναι τα ανδροειδές και συμβολίζει το ρομπότ με τη μορφή ανθρώπου σε χρώμα πράσινου μήλου το όποιο σχεδιάστηκε από τη γραφίστρια Irina.

Το Android έχει μια μεγάλη κοινότητα προγραμματιστών που γράφουν εφαρμογές οι οποίες επεκτείνουν τη λειτουργικότητα των συσκευών. Μέσω του Google play (Android Market) μπορεί να γίνει άμεση εγκατάσταση εφαρμογών χωρίς τη χρήση υπολογιστή. Το Google Play Store προσφέρει πάνω 700.000 εφαρμογές,

1.2 Εξέλιξη του Android

Η πρώτη έκδοση του Android SDK που εμφανίστηκε 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. Σε σύντομο χρονικό διάστημα έχει παρουσιάσει ραγδαία εξέλιξη και από έκδοση σε έκδοση έχει σημαντικές αλλαγές και βελτιώσεις τόσο στην εμφάνιση όσο και στις λειτουργίες του.

Παρακάτω φαίνονται οι εκδόσεις του Λειτουργικού Συστήματος Android και τα χαρακτηριστικά τους. Η κωδική ονομασία των εκδόσεων είναι σε μορφή επιδόρπιου.

1.2.1 Έκδοση 1.5 (CupCake)

Τον Απρίλιο του 2009 εμφανίστηκε η έκδοση Android 1.5, ονόματι ‘Cupcake’

Εικόνα 1: Λογότυπο Android 1.5 CUPCAKE

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

1.2.2 Έκδοση 1.6 (Donut)

Στη συνέχεια τον Σεπτέμβριο του 2009 εμφανίστηκε η έκδοση Android 1.6, ονόματι 'Donut'.

Εικόνα 2: Λογότυπο Android 1.6 DONUT

Νέα χαρακτηριστικά της έκδοσης αυτής :

- Βελτιωμένο Android Market
- Ενσωματωμένη φωτογραφική μηχανή, βιντεοκάμερα και διεπαφή (interface) γκαλερί
- Η γκαλερί επιτρέπει πλέον στους χρήστες την επιλογή πολλαπλών φωτογραφιών προς διαγραφή
- Ανανεωμένη φωνητική αναζήτηση, με ταχύτερη απόκριση και βαθύτερη ολοκλήρωση με εγγενείς (native) εφαρμογές, συμπεριλαμβανομένης της δυνατότητας κλήσης επαφών
- 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 αναζήτησης και των εφαρμογών της φωτογραφικής μηχανής

1.2.3 Έκδοση 2.0-2.1 (Eclair)

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

Εικόνα 3: Λογότυπο Android 2.0 ECLAIR

Αλλαγές της συγκεκριμένης έκδοσης:

- Βέλτιστη ταχύτητα υλικού
- Υποστήριξη για περισσότερες οθόνες και αναλύσεις
- Βελτιωμένη διεπαφή χρήστη
- Καινούργια διεπαφή 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

1.2.4 Έκδοση 2.2 (Froyo)

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

Εικόνα 4: Λογότυπο Android 2.2 FROYO

Η έκδοση αυτή, 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
- Σύνδεση USB και λειτουργία δυναμικής ζώνης (hotspot) WiFi
- Ανανεωμένη εφαρμογή Αγοράς (Market) με αυτόματη ανανέωση
- Επιλογή για απαγόρευση πρόσβασης δεδομένων μέσω ενός δικτύου 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 14
- Υποστήριξη Adobe Flash 10.1

1.2.5 Έκδοση 2.3 (GingerBread)

Τον Φεβρουάριο του 2010 κυκλοφορεί η έκδοση το Android 2.3 με το όνομα ‘GingerBread’.

Εικόνα 5: Λογότυπο Android 2.3 GINGERBREAD

Χαρακτηριστικά:

- Βελτιωμένο UI (User Interface) για απλότητα και ταχύτητα
- Υποστήριξη για πολύ μεγάλα μεγέθη οθονών και αναλύσεων(WXGA και μεγαλύτερες)
- 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
- Download manager για κατέβασμα μεγάλων αρχείων
- Υποστήριξη του πρωτόκολλου WebM για αναπαραγωγή video

1.2.6 Έκδοση 3.0 (Honeycomb)

Η Android 3.0 με το όνομα 'Honeycomb', βγήκε τον Ιούλιο του 2011 με την ιδιαιτερότητα ότι προοριζόταν αποκλειστικά για tablets.

Εικόνα 6: 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"

1.2.7 Έκδοση 4.0 (Ice cream Sandwich)

Μια από τις πιο βελτιωμένες εκδόσεις είναι το Android 4.0 ονόματι 'Ice Cream Sandwich' η οποία κυκλοφόρησε τον Δεκέμβριο του 2011.

Εικόνα 7: Λογότυπο Android 4.0 Ice Cream Sandwich

Επιπρόσθετες αλλαγές της έκδοσης αυτής :

- Εισήγαγε την νέα γραμματοσειρά Roboto
- Πατώντας τα "Volume Down" + "Power" μπορούμε να πάρουμε screenshot της οθόνης,
- Βελτιστοποίηση στο πληκτρολόγιο
- Προσθήκη voice-to-text που επιτρέπει πλέον να μιλάμε με παύσεις ανάμεσα στις λέξεις.
- Καθαρισμός μπάρας ειδοποιήσεων
- Προσθήκη συγχρονισμού των bookmarks από τον dekstop Chrome και δυνατότητα εναλλαγής από mobile σε desktop περιβάλλον των σελίδων .
- Face Unlock , αναγνώριση προσώπου στα κινητά μας.
- Gmail νέο τρόπο εμφάνισης και νέες λειτουργίες.
- Ενσωμάτωση στο ICS τη δική της εφαρμογή για την διαχείριση των δεδομένων από/προς το διαδίκτυο.
- Βελτιώσεις κ προσθήκες για Φωτογραφίες, Video & Gallery
- Android Beam, μπορεί να γίνεται ανταλλαγή δεδομένων με τα Android κινητά με NFC, με μια απλή επαφή των δύο συσκευών

1.2.8 Έκδοση 4.1-4.2 (Jelly Bean)

Τον Ιούλιο του 2012 βγήκε το Android 4.1 'Jelly Bean' και τον Νοέμβριο 2012 ακλούθησε το Android 4.2 με όνομα "Jelly Bean".

Εικόνα 8: Λογότυπο Android 4.1-4.2 Jelly Bean

Χαρακτηριστικά :

- Ασύρματο streaming μέσω Wi-Fi
- Πανοραμικές φωτογραφίες , Street View
- Βελτιωμένο πληκτρολόγιο
- Πολλαπλοί χρήστες
- Άλλες βελτιώσεις
- Daydream, απεικονίζει ειδήσεις από το Google Currents ή φωτογραφίες από τη γκαλερί σας.
- Lock Screen Widgets
- Επιτρέπει στον χρήστη να πραγματοποιεί κάποιες ενέργειες χωρίς να χρειαστεί το ξεκλείδωμα της συσκευής

1.2.9 Έκδοση 4.3 (Jelly Bean)

Μια πιο ενημερωμένη έκδοση του Jelly Bean βγήκε τον Ιούλιο του 2013 , το Android 4.3 ‘Jelly Bean’.

Εικόνα 9: Λογότυπο Android 4.3 Jelly Bean

Προσθήκες-Αλλαγές στη έκδοση Jelly Bean

- Περιορισμένα προφίλ
- Μπλοκάρισμα αγοράς εφαρμογών
- Bluetooth Smart ή αλλιώς Bluetooth Low energy για χαμηλότερη κατανάλωση σε αυτού του τύπου τις συνδέσεις
- OpelGL ES 3.0 υποστήριξη, για καλύτερα γραφικά

1.2.10 Έκδοση 4.4 (Kitkat)

Η τελευταία έκδοση που έχει κυκλοφορήσει έως τώρα είναι το Android 4.4 με όνομα ‘Kitkat’ , τον Οκτώβριο του 2013.

Εικόνα 10: Λογότυπο Android 4.4 Kitkat

Αυτή η έκδοση περιλαμβάνει :

- Google Now, χαμηλότερη κατανάλωση RAM, αναβάθμιση για περισσότερα smartphones
- Σπάει" το φράγμα των περιορισμών μνήμης που υπήρχε έως τώρα στο Android και εμπόδιζε low-end μοντέλα να επωφεληθούν από αναβαθμίσεις
- Μίνι "λίφτινγκ", ώστε το αποτέλεσμα να είναι πιο καθαρό και πιο απλό
- Το status και το notification bar της οθόνης έναρξης είναι πλέον διάφανα
- Phone app, που κάνει τις κλήσεις σας πιο εύκολες από ποτέ
- Hangouts, που πλέον διαχειρίζεται όλα τα SMS και MMS σας στο ίδιο σημείο με τις "Internet" επικοινωνίες σας
- Βελτιώσεις υπάρχουν και στην υπηρεσία φωνητικών αναζητήσεων
- Οι αναζητήσεις στο νέο λειτουργικό είναι καλύτερες από ποτέ
- Ευκολότερη εκτύπωση με το Google Cloud Print
- Βελτιώσεις στο Bluetooth
- Βιντεοσκόπησή της επιφάνειας εργασίας
- Καλύτερη διαχείριση αρχείων στο cloud
- Μεγαλύτερη ασφάλεια
- Πληρωμές μέσω NFC version 2.0

1.2.11 Έκδοση 5.0 (Key Lime Pie)

Μια πολλά υποσχόμενη έκδοση είναι το Andoid 5.0 με όνομα ‘Key Lime Pie’ η οποία αναμένεται στο επόμενο έτος.

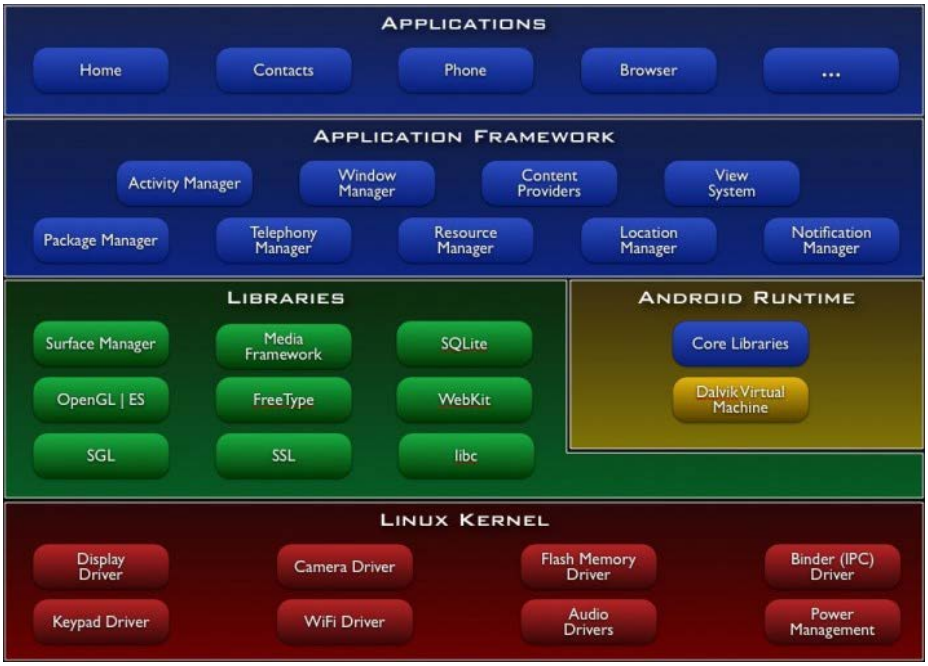
Εικόνα 11: Λογότυπο Android 5.0 Key Lime Pie

Προσθήκες και βελτιώσεις :

- Ενημέρωση φωτογραφικής μηχανής με την τεχνολογία Φωτόσφαιρας σε κάποιες συσκευές Google Nexus
- Βελτιωμένο πληκτρολόγιο με έξυπνη πληκτρολόγηση και τεχνολογία Swype
- Google Babel , θα ενοποιήσει περισσότερες επιλογές μηνυμάτων (Όπως Συζήτηση / Στέκια, Φωνή, SMS si Gmail)
- Εκμετάλλευση δυνατοτήτων ισχυρών επεξεργαστών
- Βελτίωση Multi-tasking και GPU
- Διάρκεια ζωής της μπαταρίας ελαφρώς υψηλότερη

1.3 Αρχιτεκτονική του Android

Το Android δεν είναι μόνο ένα λειτουργικό σύστημα. Είναι μια στοίβα λογισμικού η οποία αποτελείτε από το λειτουργικό σύστημα, τις υπηρεσίες διασύνδεσης με τις εφαρμογές (middleware) και τέλος από τις κύριες (core) εφαρμογές. Στο επίσημο σχεδιάγραμμα που ακολουθεί θα δούμε οπτικά την αρχιτεκτονική αυτή.



Εικόνα 12: Επίσημο σχεδιάγραμμα επιπέδων στοίβας Android

Τα επίπεδα της στοίβας του Android παρουσιάζονται παρακάτω από το υψηλότερο προς το χαμηλότερο :

- Επίπεδο εφαρμογών (Applications)
- Επίπεδο πλαισίου εφαρμογών (Application Framework)
- Επίπεδο βιβλιοθηκών (Libraries)
- Επίπεδο χρόνου εκτέλεσης (Android Runtime)
- Επίπεδο πυρήνα Linux (Linux Kernel)

1.3.1 Επίπεδο εφαρμογών (Applications)

Στο επίπεδο αυτό περιλαμβάνεται ένα σύνολο από βασικές εφαρμογές μερικές από τις οποίες είναι e-mail client, πρόγραμμα sms , ημερολόγιο , χάρτες , browser ,επαφές κ.α. Όλες οι εφαρμογές είναι γραμμένες με χρήση της γλώσσας προγραμματισμού Java.

1.3.2 Επίπεδο Επίπεδο πλαισίου εφαρμογών (Application Framework)

Το Android παρέχει στους developers μια ανοιχτού κώδικα πλατφόρμα ανάπτυξης και τη δυνατότητα να αναπτύξουν με αυτή ιδιαίτερα καινοτόμες και πλούσιες σε υλικό, εφαρμογές. Στο επόμενο επίπεδο της αρχιτεκτονικής του Android λοιπόν, συναντάμε το πλαίσιο των εφαρμογών το οποίο αποτελείται από ένα σύνολο συστημάτων και εφαρμογών. Τα σημαντικότερα δομικά στοιχεία του πλαισίου εφαρμογών είναι:

- **Σύστημα προβολών (View System)** – αποτελεί ένα εκτενές σύνολο από αντικείμενα GUI τα οποία μπορούν να χρησιμοποιηθούν κατά το σχεδιασμό μιας εφαρμογής. Παραδείγματα προβολών είναι οι λίστες (listView), το πλέγμα (GridView), πεδία εισαγωγής κειμένου, κουμπιά, κλπ
- **Πάροχος Περιεχομένου (Content Provider) –** δίνει τη δυνατότητα στις εφαρμογές να μοιράζονται ή να ανταλλάσσουν δεδομένα μιας συγκεκριμένης μορφής η οποία ορίζεται από τον πάροχο. Παραδείγματα δεδομένων, είναι οι επαφές χρήστη και οι βάσεις δεδομένων των εφαρμογών.
- **Διαχειριστής Πόρων (Resource Manager) –** παρέχει πρόσβαση σε υλικό το οποίο δεν είναι σε μορφή κώδικα όπως πχ, εικόνες, αρχεία xml, πίνακες χαρακτήρων, κλπ
- **Διαχειριστής Ειδοποιήσεων (Notification Manager) –** δίνει στις εφαρμογές πρόσβαση στις υπηρεσίες ειδοποιήσεων χρήστη. Τέτοιες είναι οι ειδοποιήσεις στη notification bar, τα toast μηνύματα στο κάτω μέρος της οθόνης, η δόνηση του κινητού και η ενεργοποίηση της οθόνης, κλπ
- **Διαχειριστής Δραστηριοτήτων (Activity Manager) –** διαχειρίζεται τον κύκλο ζωής των δραστηριοτήτων και παρέχει δυνατότητα πλοήγησης από δραστηριότητα σε δραστηριότητα κρατώντας αποθηκευμένη στη μνήμη τη σειρά εκτέλεσης αυτών. Στο σχεδιάγραμμα (Εικόνα ?) φαίνεται λεπτομερώς ο κύκλος ζωής κάθε δραστηριότητας.

1.3.3 Επίπεδο βιβλιοθηκών (Libraries)

Στο δεύτερο επίπεδο της στοίβας έχουμε τις βιβλιοθήκες του Android. Αυτό περιλαμβάνει ένα σύνολο από βιβλιοθήκες γραμμένες σε C και C++. Αυτές ουσιαστικά αποτελούν τα APIs που είναι διαθέσιμα στους προγραμματιστές για την ανάπτυξη των εφαρμογών. Οι δυνατότητες των βιβλιοθηκών του Android γίνονται εμφανείς στους προγραμματιστές στην στοίβα του πλαισίου εφαρμογής. Μερικές από τις κύριες βιβλιοθήκες του Android είναι:

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

1.3.4 Επίπεδο χρόνου εκτέλεσης (Android Runtime)

Αποτελείται από ένα σύνολο από βασικές βιβλιοθήκες και την Dalvik Virtual Machine.Η Dalvik Virtual Machine είναι η εικονική μηχανή μέσω της οποίας τρέχουν οι εφαρμογές του Android.

1.3.5 Επίπεδο πυρήνα Linux (Linux Kernel)

Η βάση της στοίβας λογισμικού του Android είναι ο πυρήνας Linux. Βασίζεται στο πυρήνα Linux έκδοση 2.6 (και στην έκδοση 3.0.1 για το Android 4.0) του Linux Kernel, η οποία υποστηρίζει λειτουργίες του λειτουργικού συστήματος που αφορούν διαχείριση μνήμης, διαχείριση διεργασιών, λειτουργίες δικτύου, ασφάλεια του λειτουργικού, και ένα σύνολο οδηγών υλικού (hardware drivers). Οι οδηγοί αυτοί είναι υπεύθυνοι για την επικοινωνία του software με το hardware της συσκευής. Ενδεικτικά ο πυρήνας του Android περιέχει:

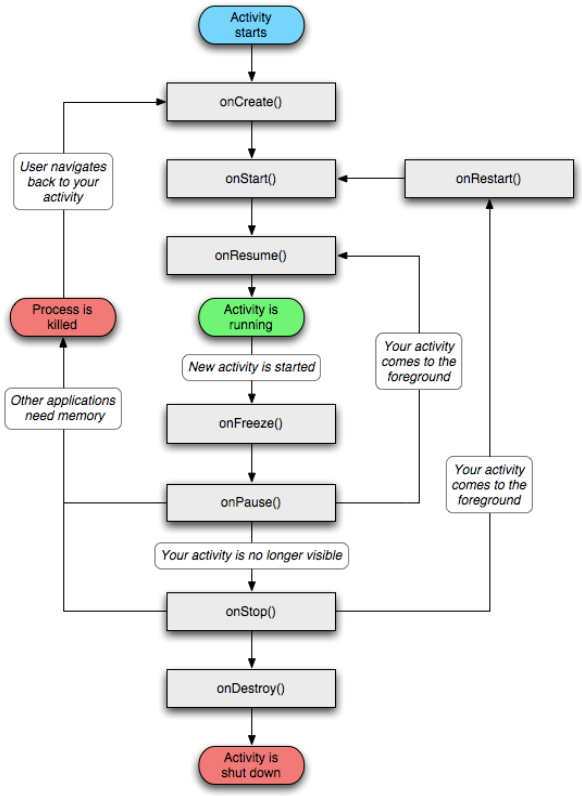
• Οδηγό προβολής οθόνης
• Οδηγό Wifi και Bluetooth

• Οδηγό κάμερας, κλπ

1.4 Στο εσωτερικό μιας εφαρμογής του Android

Κάθε εφαρμογή αποτελείτε από ένα σύνολο αρχείων και φακέλων δομημένα σε μορφή project, τα οποία αφού γίνουν compiled μέσω του Android SDK μας δίνουν το αρχείο .apk. Το αρχείο αυτό αποτελεί την εφαρμογή και μπορούμε να εγκαταστήσουμε στις συσκευές μας.

Ξεκινώντας, η κάθε εφαρμογή αποτελείτε όπως είπαμε από πολλά αρχεία δομημένα σε φακέλους. Όλες οι εφαρμογές πρέπει να έχουν ένα μοναδικό όνομα πακέτου (package name) το οποίο χρησιμοποιείτε από το λειτουργικό σύστημα για αναγνώριση της εφαρμογής. Μια εφαρμογή μπορεί να αποτελείτε από πολλά υποπακέτα, εφόσον αυτό είναι απαραίτητο λόγω της πολυπλοκότητας τις εφαρμογής, αλλά μόνο από ένα κύριο.



Εικόνα 13: Κύκλος ζωής μιας δραστηριότητας (Activity lifecycle)

1.4.1 Το αρχείο AndroidManifest.xml

Κάθε project εφαρμογής περιέχει ένα αρχείο στο οποίο βρίσκονται καταχωρημένες οι σημαντικότερες πληροφορίες της εφαρμογής, και το αρχείο αυτό ονομάζεται AndroidManifest.xml. Πρόκειται όπως λέει και το όνομα του για ένα αρχείο xml μέσα στο οποίο ο προγραμματιστής καταχωρεί τις σημαντικότερες πληροφορίες της εφαρμογής για χρήση από το λειτουργικό σύστημα. Κάποιες από αυτές τις πληροφορίες είναι:

- Το όνομα του πακέτου της εφαρμογής
- Το κανονικό της όνομα που φαίνεται στον χρήστη
- Η έκδοση των APIs που χρησιμοποιούνται
- Ο αριθμός έκδοσης της εφαρμογής
- Οι άδειες χρήσης που ζητάει η εφαρμογή
- Όλες οι δραστηριότητες, πάροχοι περιεχομένου, υπηρεσίες, κλπ, που περιέχει και χρησιμοποιεί η εφαρμογή. Όπως αντιλαμβανόμαστε πρόκειται για πολύ σημαντικό αρχείο και αποτελεί κύριο συστατικό κάθε εφαρμογής.

1.4.2 Οι φάκελοι src & res

Στον φάκελο src (εκ του source) περιέχονται τα αρχεία κλάσης τις Java όλων των Activities, Services, Content Providers, βοηθητικά αρχεία, κλπ. Ο φάκελος περιέχει το πακέτο ή τα πακέτα της εφαρμογής τα οποία περιέχουν τα αρχεία Java, και αποτελεί τον μοναδικό φάκελο στο project στον οποίο αποθηκεύονται τα αρχεία του κώδικα μας.

Ο φάκελος res (εκ του resources) περιέχει όλα τα αρχεία εικόνας, κειμένου, xml layout, κλπ τα οποία χρησιμοποιούνται από τις Activities που βρίσκονται στον φάκελο src. Φυσικά δεν βρίσκονται όλα τα αρχεία πόρων, σε έναν φάκελο, αλλά είναι χωρισμένα και ταξινομημένα σε υποφακέλους ανάλογα με το είδος τους. Συνηθισμένοι υποφάκελοι του κύριου φακέλου res, είναι ο φάκελος drawable ο οποίος περιέχει τα αρχεία εικόνας (.png, .jpg, .gif) τα οποία χρησιμοποιεί η εφαρμογή μας, ο φάκελος layout ο οποίος περιέχει όλα τα αρχεία xml τα οποία ορίζουν τα διάφορα layouts που υπάρχουν στην εφαρμογή, και τέλος ο φάκελος values στον οποίο αποθηκεύονται όλοι οι πόροι κειμένου που χρησιμοποιούνται
στην εφαρμογή.

1.4.3 Οι υπόλοιποι φάκελοι του project

Ένα project αποτελείτε από περισσότερους από τους 3 βασικούς φακέλους, κάποιοι από τους οποίους μπορεί να θεωρηθούν και περιττοί αναλόγως την περίπτωση. Στο project λοιπόν περιλαμβάνονται και ο φάκελος με τα διαθέσιμα APIs αναλόγως την έκδοση που έχουμε επιλέξει να δουλέψουμε, ο φάκελος με τις διαθέσιμες βιβλιοθήκες που έχουμε εισάγει στο build path του project μας, και επίσης περιλαμβάνει και τις διαβαθμίσεις του φακέλου res, όπως είναι οι φάκελοι drawable-hdpi, drawable-mdpi, layout-port, menu, κλπ. Σε αυτούς περιλαμβάνονται τα ειδικά διαμορφωμένα αρχεία πόρων που έχουμε τοποθετήσει ώστε να είναι διαθέσιμα από το λειτουργικό σύστημα, αναλόγως την περίσταση.

1.4.4 Δομικά Μέρη μιας Εφαρμογής

Παραπάνω αναφέραμε ότι όλα τα δομικά μέρη της εφαρμογής πρέπει να αναφέρονται αναλυτικά στο αρχείο AndroidManifest.xml, πια είναι όμως αυτά τα δομικά μέρη και πια η λειτουργία του καθενός;

- **Δραστηριότητες (Activities)** – Πρόκειται ίσως για το κύριο δομικό στοιχείο μιας εφαρμογής. Δραστηριότητα είναι μια οθόνη διεπαφής χρήστη (GUI) και

προβολής πληροφοριών. Κάθε εφαρμογή έχει τόσες Activities όσες και οι διαφορετικές οθόνες οι οποίες εμφανίζονται στον χρήστη. Όλες οι δραστηριότητες συνεργάζονται μεταξύ τους για να δώσουν στον χρήστη μια συνολική εμπειρία χρήσης της εφαρμογής.

- **Προθέσεις (Intents) –** Οι δραστηριότητες επικοινωνούν και εναλλάσσουν την λειτουργία τους μέσω των Intents. Ουσιαστικά τα Intents εξασφαλίζουν την μετάβαση από την μία δραστηριότητα σε μια άλλη και επίσης χρησιμοποιούνται για ανταλλαγή δεδομένων. Η ανταλλαγή δεδομένων, μπορεί να γίνει είτε μεταξύ των Activities μιας εφαρμογής, είτε από τη μία εφαρμογή στην άλλη.
- **Υπηρεσίες (Services) –** Πρόκειται για λειτουργίες της εφαρμογής οι οποίες είναι σχεδιασμένες να τρέχουν στο παρασκήνιο και να επιστρέφουν αποτελέσματά ακόμη και όταν η εφαρμογή δεν είναι στο προσκήνιο.
- **Πάροχος Περιεχόμενου (Content Providers) -** Η ανταλλαγή δεδομένων από μια εφαρμογή στην άλλη όπως είπαμε παραπάνω μπορεί να γίνει μέσω ενός Intent, ένας πάροχος περιεχομένου όμως έχει πιο σύνθετη λειτουργία. Οι content providers μιας εφαρμογής διαχειρίζονται συγκεκριμένα δεδομένα της εφαρμογής τα οποία έχει ορίσει ο προγραμματιστής κατά την κατασκευή του. Συνηθισμένα δεδομένα τα οποία μοιράζονται μέσω Content Providers, είναι οι βάσεις δεδομένων SQLite μιας εφαρμογής, και οι επαφές του χρήστη.
- **Δέκτες Μετάδοσης (Broadcast Receivers) –** Πρόκειται για ένα είδους υπηρεσίας η οποία αντιλαμβάνεται κάποια γεγονότα του συστήματος και αναλαμβάνει να ενημερώσει το σύστημα ή τις υπόλοιπες εφαρμογές. Ο σκοπός τους είναι διπλός καθότι μπορούν και να ενημερωθούν για κάποιο συμβάν από άλλες εφαρμογές, αλλά και να ειδοποιήσουν τις υπόλοιπες εφαρμογές και το σύστημα για κάποιο συμβάν που τις ενεργοποίησε. Δεν έχουν γραφικό περιβάλλον αλλά μπορούν να προβάλουν ειδοποίηση στον χρήστη μέσω της μπάρας ειδοποιήσεων. Συνήθως χρησιμοποιούνται ως διαμεσολαβητές μεταξύ των Activities και των Services μιας εφαρμογής.

1.4 Ασφάλεια στο Android

Τη στιγμή που μια εφαρμογή εγκαθίσταται στη συσκευή, λειτουργεί αποκλειστικά στο δική της εικονική μηχανή η οποία αποτελεί και το πλαίσιο ασφαλείας (sandbox) της εφαρμογής. Το Android είναι ένα λειτουργικό σύστημα πολλών χρηστών στο οποίο:

- Η κάθε εφαρμογή αντιμετωπίζεται σαν διαφορετικός χρήστης
- Από προεπιλογή το σύστημα δίνει έναν μοναδικό αριθμό ID ο οποίος είναι άγνωστος στην εφαρμογή. Το σύστημα αναθέτει συγκεκριμένες άδειες χρήσης στα αρχεία της εφαρμογής, και μόνο η εφαρμογή με το σωστό ID μπορεί να έχει πρόσβαση σε αυτά.
- Κάθε εφαρμογή τρέχει στην δική της εικονική μηχανή (VM) απομονωμένη από τις υπόλοιπες εφαρμογές. Η κάθε VM εκκινείτε μόλις ζητηθεί από το σύστημα και κλείνει είτε επειδή δεν χρησιμοποιείτε πλέον, είτε επειδή το σύστημα θέλει να ελευθερώσει τους πόρους της μνήμης για χρήση από άλλη εφαρμογή.

Με αυτό τον τρόπο το Android χρησιμοποιεί την αρχή των ελαχίστων δικαιωμάτων. Η κάθε εφαρμογή έχει πρόσβαση μέσω του AndroidManifest μόνο σε όσους πόρους συστήματος χρειάζεται και κανέναν περισσότερο. Οι πόροι και τα δικαιώματα που απαιτούνται από μία εφαρμογή γίνονται γνωστά στον χρήστη τη στιγμή της εγκατάστασης της, και ο χρήστης μπορεί να επιλέξει να μην εγκαταστήσει μια εφαρμογή εφόσον δεν συμφωνεί να τις παρέχει πρόσβαση στους πόρους που ζητάει.

2

Περιβάλλον Eclipse

2.1 Βήματα ανάπτυξης εφαρμογών στο Android

Η ανάπτυξη εφαρμογών στο Android είναι μια σύνθετη διαδικασία η οποία συνοψίζεται σε 4 βασικά στάδια .

2.1.1 Εγκατάσταση Λογισμικού

Στο πρώτο στάδιο της ανάπτυξης ο προγραμματιστής καλείτε να στήσει το περιβάλλον εργασίας στο οποίο θα γίνει ο σχεδιασμός, η ανάπτυξη, ο έλεγχος, και η λειτουργία των εφαρμογών. Μπορεί να επιλέξει όποιο περιβάλλον ανάπτυξης (IDE) τον εξυπηρετεί καλύτερα και να χρησιμοποιήσει όλα τα εργαλεία του Android SDK μηδενός εξαιρουμένου.

Στη συνέχεια θα πρέπει να δημιουργήσει έναν αριθμό από εικονικές συσκευές στην διαχείριση εικονικών συσκευών (AVD) για να δοκιμάσει την λειτουργία της εφαρμογής σε διαφορετικές πραγματικές συνθήκες λειτουργίας.

2.1.2 Ανάπτυξη Πηγαίου Κώδικα Εφαρμογής

Σε αυτό το στάδιο ο προγραμματιστής πρέπει να αποφασίσει για τις δυνατότητες και το περιεχόμενο που θα περιλαμβάνει η εφαρμογή, να σχεδιάσει το layout με γνώμονα την λειτουργικότητα και να αποφύγει υπερβολές στο σχεδιασμό, και τέλος να δέσει αρμονικά τον κώδικα με το layout για να φέρει το τελικό αποτέλεσμα. Η διαδικασία ξεκινάει με ένα νέο Project το οποίο θα περιέχει τον πηγαίο κώδικα, τις εικόνες, τα κείμενα και γενικά ότι χρειάζεται η εφαρμογή για να τρέξει ως οφείλει. Στο project του ο developer θα πρέπει να φροντίσει ώστε το υλικό του να είναι τακτοποιημένο και ο κώδικας του ευανάγνωστος ώστε να ακολουθήσει η διαδικασία του Debugging.

2.1.3 Αποσφαλμάτωση (Debugging) και Δοκιμαστική Φάση Εφαρμογής

Η διαδικασία του debugging είναι εξίσου κρίσιμη και μερικές φορές και εξίσου χρονοβόρα με την διαδικασία ανάπτυξης του πηγαίου κώδικα της εφαρμογής. Αποτελείτε από αρκετά επί μέρους στάδια τα οποία αναλύονται παρακάτω.

Το πρώτο στάδιο αφορά το αρχικό χτίσιμο της εφαρμογής και η λειτουργία αυτής σε debug mode. Για να γίνει το compile της εφαρμογής φυσικά τα περισσότερα περιβάλλοντα ανάπτυξης (IDE) προϋποθέτουν ότι ο κώδικας δεν έχει κανένα συντακτικό λάθος, αλλιώς ειδοποιούν τον χρήστη να τα διορθώσει. Αφού γίνει το compile η εφαρμογή μπορεί να δοκιμαστεί είτε σε εικονική συσκευή μέσω του AVD Manager, είτε απευθείας σε φυσική συσκευή μέσω ADB push εντολής. Για το ADB θα μιλήσουμε εκτενώς παρακάτω.

Στο δεύτερο στάδιο ο προγραμματιστής καλείτε να αντιμετωπίσει τα λειτουργικά και αισθητικά προβλήματα της εφαρμογής του, πρώτα εντοπίζοντας τα στην λειτουργία της συσκευής και μετά διορθώνοντας τα κομμάτια του κώδικα που δημιουργούν τα σφάλματα. Το κύριο εργαλείο που κάνει αυτή τη διαδικασία εφικτή είναι το “LogCat” το οποίο μας επιστρέφει το stack trace του κώδικα στο σημείο εκείνο που συνέβη το σφάλμα. Υπάρχουν φυσικά και άλλα εργαλεία τα οποία θα αναλυθούν εκτενώς παρακάτω.

Στο τρίτο στάδιο ο προγραμματιστής επιστρέφει στο compile και τη δοκιμή της εφαρμογής σε εικονική η φυσική συσκευή ώστε να διαπιστώσει τα αποτελέσματα του βήματος, της αποσφαλμάτωσης.

2.1.4 Τελική έκδοση και δημοσίευση της εφαρμογής στο κοινό

Στο τέταρτο και τελευταίο στάδιο της ανάπτυξης ο προγραμματιστής πρέπει να έχει διορθώσει όλα τα σφάλματα που προέκυψαν από την διαδικασία αποσφαλμάτωσης, να κάνει τις τελευταίες ρυθμίσεις και tweaks της εφαρμογής, και να κάνει το τελικό compile της εφαρμογής σε κανονική λειτουργία αυτή τη φορά και όχι debug.

2.2 Εργαλεία ανάπτυξης λογισμικού

Το Android SDK αποτελεί μια συλλογή εργαλείων και βιβλιοθηκών που καθιστούν εφικτή την ανάπτυξη εφαρμογών στο Android, χρησιμοποιώντας την γλώσσα προγραμματισμού Java. Τα χαρακτηριστικά του παρέχουν τεράστια ευελιξία και δυνατότητα ανάπτυξης έξυπνων εφαρμογών για κινητά τηλέφωνα. Ακόμα, παρέχει APIs για την χρήση web browser, εμφάνιση δισδιάστατων και τρισδιάστατων γραφικών, δομημένη αποθήκευση δεδομένων σε βάση δεδομένων, εμφάνιση πολυμεσικού υλικού (ήχος, βίντεο, εικόνες), χρήση των τεχνολογιών GSM, Bluetooth, EDGE, 3G και WiFi, χρήση συσκευών όπως φωτογραφική μηχανή, GPS, πυξίδα, επιταχυνσιόμετρο.

Ένα σημαντικό γεγονός που ευνοεί την ανάπτυξη εφαρμογών είναι πως το πακέτο Android SDK συνεργάζεται με το Eclipse και συνεπώς ο προγραμματιστής

μπορεί εύκολα και γρήγορα να βλέπει τις αλλαγές του κώδικα στον emulator που του παρέχει το Android SDK, τον οποίο αναλύουμε παρακάτω. Ένα άλλο σημαντικό πλεονέκτημα είναι το γεγονός πως σε αναβαθμίσεις του λογισμικού, η εφαρμογή εξακολουθεί να δουλεύει χωρίς την ανάγκη επανασχεδιασμού κάποιων σημαντικών κομματιών του κώδικα, που αφορούν την αλληλεπίδραση της εφαρμογής με τα δομικά μέρη-hardware- του κινητού τηλεφώνου.

Το SDK υποστηρίζει πολλά δημοφιλή λειτουργικά συστήματα συμπεριλαμβανομένων όλων των σύγχρονων διανομών Linux, το MAC OS , Windows. Επιπλέον περιλαμβάνει μια μεγάλη λίστα με εργαλεία ανάπτυξης. Σε αυτά περιλαμβάνονται:

- Εργαλεία Debugging των εφαρμογών Βιβλιοθήκες
- Εξομοιωτής συσκευών (Android Virtual Machines)
- Documentation
- Δείγματα ΚώδικαTutorials

2.2.1 Μερικά από τα χαρακτηριστικά του SDK

Το Android SDK είναι ένα πολύ χρήσιμο εργαλείο για την εξερεύνηση των «ενδοτέρων» του Android. Παρακάτω, φαίνονται τα βασικότερα χαρακτηριστικά του.

- Εφαρμογή πλαισίου που επιτρέπει την επαναχρησιμοποίηση και την αντικατάσταση στοιχείων
- Βελτιστοποιημένη Dalvik εικονική μηχανή για κινητές συσκευές
- Ολοκληρωμένο πρόγραμμα περιήγησης, το οποίο βασίζεται στο open source WebKit Βελτιστοποιημένα γραφικά, τα οποία τροφοδοτούνται από μια προσαρμοσμένη βιβλιοθήκη 2D και 3D γραφικών με βάση τις προδιαγραφές του OpenGL ES 1.0 (επιτάχυνση hardware προαιρετικά)
- SQLite για δομημένη αποθήκευση δεδομένων
- Υποστήριξη πολυμέσων για αρχεία ήχου, βίντεο, ακόμα και εικόνων (MPEG4, H.264, MP3,AAC, AMR, JPG, PNG, GIF)
- GSM Τηλεφωνία (εξαρτώμενη από το hardware)
- Bluetooth, EDGE, 3G, WiFi (εξαρτώμενα από το hardware).
- Φωτογραφική μηχανή, GPS, πυξίδα, και επιταχυνσιόμετρο (εξαρτώμενα από το hardware)
- Πλούσιο περιβάλλον ανάπτυξης, συμπεριλαμβανομένου ενός εξομοιωτή συσκευής, εργαλεία
- για τον εντοπισμό σφαλμάτων, μνήμη και προφίλ απόδοσης όπως επίσης και ένα plugin για το
- Eclipse IDE

2.2.2 Ο εξομοιωτής (emulator).

Προκειμένου να γίνει ευκολότερη η διαδικασία της ανάπτυξης και αποσφαλμάτωσης μιας εφαρμογής, το Android SDK περιλαμβάνει έναν εξομοιωτή μιας εικονικής κινητής συσκευής, η οποία τρέχει το λειτουργικό του Android.

Εικόνα 14: Ο εξομοιωτής (emulator)

Έτσι δεν είναι η αναγκαία η ύπαρξη πραγματικής κινητής συσκευής για την εκτέλεση και δοκιμή των εφαρμογών. Ο εξομοιωτής προσομοιώνει ένα μεγάλο πλήθος λειτουργιών μιας τυπικής συσκευής, η οποία τρέχει το Android:

- Παρέχει μια ποικιλία πλήκτρων πλοήγησης και ελέγχου
- Παρέχει μια οθόνη για την προβολή των εφαρμογών που τρέχουν στον εξομοιωτή
- Επιτρέπει στις εφαρμογές την χρήση των υπηρεσιών που προσφέρει η πλατφόρμα του Android, δηλαδή την κλήση άλλων εφαρμογών, την πρόσβαση στο δίκτυο, την αναπαραγωγή ήχου και βίντεο, την αποθήκευση και επαναφορά δεδομένων, την ειδοποίηση χρήστη, το γραφικό περιβάλλον του Android.

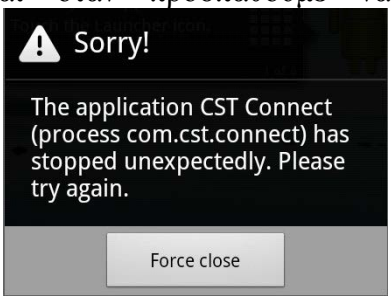
Επίσης παρέχει ένα πλήθος λειτουργιών για την ευκολότερη αποσφαλμάτωση:

- Κονσόλα για την καταγραφή της εξόδου του πυρήνα
- Προσομοίωση διακοπών (όπως η άφιξη SMS μηνύματος ή τηλεφωνικής κλήσης)
- Προσομοίωση καθυστέρησης και απώλειας στο κανάλι δεδομένων
- Προσομοίωση λήψης δεδομένων θέσης από την συσκευή GPS.

2.2.3 Εργαλείο καταγραφής συμβάντων – LogCat

Το Android διαθέτει ένα μηχανισμό καταγραφής συμβάντων, σκοπός του οποίου είναι η συλλογή και προβολή των αρχείων αποσφαλμάτωσης του συστήματος. Τα δεδομένα των διάφορων εφαρμογών αλλά και του λειτουργικού συστήματος συγκεντρώνονται σε μια σειρά από buffers, τους οποίους μετά μπορούμε να προβάλουμε και να φιλτράρουμε με την εντολή “logcat”. Στον προγραμματισμό υπάρχουν οι λεγόμενες “εξαιρέσεις” (exceptions), καταστάσεις δηλαδή που προκύπτουν όταν κάτι δεν πάει καλά, και αυτό έχει σαν αποτέλεσμα την διακοπή λειτουργίας του προγράμματος σε περίπτωση που δεν έχουμε φροντίσει να “χειριστούμε” την εξαίρεση.

Συνηθισμένο παράδειγμα εξαίρεσης λειτουργίας του Android είναι η "NullPointerException", η οποία μας εμφανίζεται όταν προσπαθούμε να προσπελάσουμε κάποια μεταβλητή η αντικείμενο που έχει μηδενική (Null) τιμή. Προγραμματιστικά βέβαια υπάρχει η δυνατότητα να βάλουμε δικλείδα ασφαλείας σε μερικά επίφοβα σημεία του κώδικα μας, και να σταματήσουμε την απότομη διακοπή λειτουργίας της εφαρμογής, εμφανίζοντας έναντι μόνο το μήνυμα σφάλματος στα logs του συστήματος.Τα σφάλματα λειτουργίας μιας εφαρμογής στο Android, που προκύπτουν από εξαιρέσεις λειτουργίας, συνήθως προκαλούν τον άμεσο τερματισμό της εφαρμογής προβάλλοντας ένα παράθυρο με το όνομα της εφαρμογής που τερματίστηκε, και ένα απλό μήνυμα σφάλματος , δίνοντας μας την "επιλογή" να πατήσουμε "Force Close".



Το LogCat λοιπόν είναι μια εντολή γραμμής εντολών η οποία μπορεί να χρησιμοποιηθεί μέσω του ADB για να δούμε τα debug logs της συσκευής που δουλεύουμε, και άρα ως συνεπακόλουθο, της εφαρμογής που αναπτύσσουμε ώστε να εντοπίσουμε τις πηγές των σφαλμάτων, οι οποίες συνήθως αν όχι πάντα, είναι exceptions στον κώδικα μας. Το eclipse ενσωματώνει μία GUI έκδοση του LogCat για αποτελεσματικότερη αποσφαλμάτωση του κώδικα μας.

2.2.4 Άλλα εργαλεία του Android

Το Android SDK περιλαμβάνει μερικά ακόμη εργαλεία για την ανάπτυξη εφαρμογών:

- Το Dalvik Debug Monitor Service (DDMS) το οποίο επιτρέπει την διαχείριση των διεργασιών στον εξομοιωτή ή στην συσκευή. Συγκεκριμένα δίνεται η δυνατότητα port-forwarding υπηρεσιών, λήψη screenshots, εμφάνιση πληροφοριών για τον σωρό και τα νήματα, logcat, εμφάνιση πληροφοριών ράδιο και πληροφοριών διεργασιών, προσομοίωση εισερχόμενων κλήσεων και μηνυμάτων, προσομοίωση δεδομένων θέσης κ.α.
- Την Android Debug Bridge (ADB) η οποία επιτρέπει την διαχείριση της κατάστασης του εξομοιωτή ή της συσκευής. Μέσω του ADB είναι δυνατή η εκτέλεση εντολών φλοιού, η διαχείριση της προώθησης θυρών και η αντιγραφή από και προς την συσκευή ή τον εξομοιωτή.
- Το Android Asset Packaging Tool (AAPT) το οποίο δίνει την δυνατότητα δημιουργίας .apk αρχείων τα οποία περιέχουν τα εκτελέσιμα αρχεία και τους πόρους μιας εφαρμογής.
- Την Android Interface Description Language (AIDL) η οποία επιτρέπει την δημιουργία κώδικα που επιτρέπει σε δύο διεργασίες σε μια συσκευή βασισμένη στο Android να συνομιλούν χρησιμοποιώντας διαδιεργασιακή επικοινωνία.
- Το sqlite3 το οποίο επιτρέπει την πρόσβαση στα δεδομένα της SQLite που δημιουργούνται από τις διάφορες εφαρμογές.
- Το Traceview που επιτρέπει την γραφική προβολή της ανάλυσης των trace log data που δημιουργούν οι διάφορες εφαρμογές.

- Το mksdcard το οποίο βοηθά στην δημιουργία εικονικού δίσκου ο οποίος μπορεί να χρησιμοποιηθεί από τον εξομοιωτή για την προσομοίωση της παρουσίας εξωτερικής αποθηκευτικής κάρτας (όπως η SD card).
- Το dx tool το οποίο μετατρέπει τα αρχεία .class από java bytecode σε Android bytecode.
- Το UI/Application Exerciser Monkey το οποίο είναι ένα πρόγραμμα που τρέχει στον εξομοιωτή και παράγει ψευδό-τυχαίες σειρές από συμβάντα χρήστη όπως clicks, touches, gestures καθώς επίσης και έναν αριθμό από συμβάντα συστήματος.
- Το activitycreator το οποίο είναι ένα script που δημιουργεί Ant build αρχεία τα οποία μπορούν να χρησιμοποιηθούν για την μεταγλώττιση των εφαρμογών.

2.3 Χρήση του Eclipse IDE μαζί με το ADT (Android Development Tools)

Ο προγραμματισμός στο Android βασίζεται στην γλώσσα Java και ο κάθε προγραμματιστής μπορεί να χρησιμοποιήσει έναν οποιονδήποτε text editor για να γράψει κώδικα για να επεξεργαστεί τα αρχεία *.Java και *.XML και μετέπειτα να τα κάνει compile μέσω γραμμής εντολών χρησιμοποιώντας το JDK (Java Development Kit). Ο συγκεκριμένος τρόπος ανάπτυξης δεν είναι ιδιαίτερα φιλικός στον χρήστη γιαυτό συνίσταται η χρήση ενός IDE (Integrated Development Environment) που να υποστηρίζει Java, όπως το Eclipse ή το Netbeans.

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

Φυσικά όπως είπαμε και παραπάνω, ο κάθε προγραμματιστής μπορεί να χρησιμοποιήσει τον Text Editor ή IDE της επιλογής του για τη δημιουργία του κώδικα και μετέπειτα να χρησιμοποιήσει τα εργαλεία JDK και Apache Ant μέσω γραμμής εντολών για να κάνει compile την εφαρμογή του ώστε να την τεστάρει με όλες τις δυνατότητες που το παρέχει το Android SDK.

Η επιλογή ενός IDE που κάνει όλη την πολύπλοκη δουλεία για μας είναι προφανής λοιπόν. Επίσης τα περισσότερα παραδείγματα και άρθρα για το Android στηρίζονται στο γεγονός ότι η πλειονότητα των developers χρησιμοποιεί το Eclipse μαζί με το ADT plugin οπότε ξεκινάνε με αυτό σαν δεδομένο.

3

Αρχιτεκτονική Εφαρμογής

3.1 Περιγραφή Εφαρμογής

Το θέμα της εργασίας αφορά την ανάπτυξη μιας εφαρμογής σε περιβάλλον Android. Συγκεκριμένα , η εφαρμογή αφορά μια οθόνη κλειδώματος. Με την εγκατάσταση της εφαρμογής , ο χειριστής έχει τη δυνατότητα να επιλέξει αν θα γίνεται εκκίνηση με την οθόνη συστήματος ή με την οθόνη κλειδώματος (αυτή η επιλογή γίνεται μια φορά). Στην ουσία όταν γίνει αυτή η επιλογή , η εφαρμογή λειτουργεί σαν μια επιφάνεια εργασίας.

Ο χειριστής θα έχει πρόσβαση (με κωδικό ασφαλείας) σε ένα κεντρικό μενού από όπου θα μπορεί :

- Να επιλέγει ποιες εφαρμογές θα εμφανίζονται στην οθόνη , χωρίς κάποιος άλλος χρήστης να μπορεί να έχει πρόσβαση σε κάποια άλλη εφαρμογή.
- Να απενεργοποιεί διεργασίες κ εφαρμογές που τρέχουν στη συσκευή , ώστε να επιτυγχάνεται εξοικονόμηση μπαταρίας.
- Να αλλάζει τον κωδικό ασφαλείας της εφαρμογής .
- Να έχει πρόσβαση σε όλες τις εγκατεστημένες εφαρμογές της συσκευής.

Η αρχική οθόνη κλειδώματος θα δίνει στο χρήστη την δυνατότητα να έχει πρόσβαση μόνο στις εφαρμογές που έχουν επιλεγεί , και δεν θα μπορεί να κάνει κάποια παραμετροποίηση αφού απαιτείται κωδικός πρόσβασης.
Σκοπός της εφαρμογής αυτής είναι :

- Εξοικονόμηση μπαταρίας
- Χρήση της εφαρμογής ως πρόγραμμα περιορισμένης πρόσβασης στις λειτουργίες του κινητού.
- Προστασία της συσκευής από μη εξουσιοδοτημένη πρόσβαση.
- Βελτιστοποίηση της συσκευής και εξοικονόμηση επεξεργαστικής ισχύς λόγω τερματισμού περιττών εφαρμογών.

3.2 Απαιτήσεις Εφαρμογής

Αρχικά χρειαζόμαστε μια επιφάνεια εργασίας από την όποια θα έχει πρόσβαση ο χρήστης στις ρυθμίσεις της εφαρμογής και στα εικονίδια των επιλεγμένων εφαρμογών που θα εμφανίζονται .

Στη συνέχεια θα χρειαστεί μια διεργασία η οποία θα εμφανίζει όλες τις εφαρμογές που είναι εγκατεστημένες στο σύστημα και από κει ο χρήστης θα επιλέγει την εφαρμογή ή τις εφαρμογές που θέλει να εμφανίζονται στην επιφάνεια εργασίας του και να έχει πρόσβαση σε αυτές.

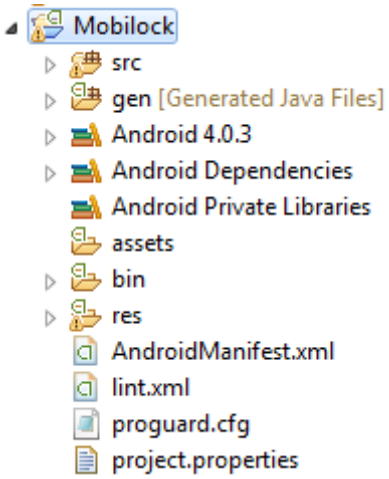
Επιπλέον, για το τερματισμό εφαρμογών θέλουμε μια συνάρτηση η οποία θα φέρνει τη λίστα των εφαρμογών που εκτελούνται στο σύστημα . Από κει ο χρήστης θα έχει τη δυνατότητα να επιλέγει ποιες εφαρμογές θέλει να τερματίζουν . Με αυτό τον τρόπο σε κάθε άνοιγμα κ κλείσιμο στης οθόνης θα υπάρχει μια διεργασία η οποία θα τερματίζει τις επιλεγμένες εφαρμογές.

Ακόμα , για να μπορεί να έχει ο χρήστης πλήρη έλεγχο της συσκευής , μετά την εγκατάσταση της εφαρμογής, θα χρειαστεί μια διεργασία η οποία θα εμφανίζει όλες τις εφαρμογές της συσκευής κ από κει θα μπορεί να έχει απευθείας πρόσβαση σε όποια επιλέξει.

Όλες οι ρυθμίσεις που κάνει ο χρήστης στην εφαρμογή αποθηκεύονται σε μια βάση δεδομένων , τύπου SQLite.

3.3 Δομή Εφαρμογής

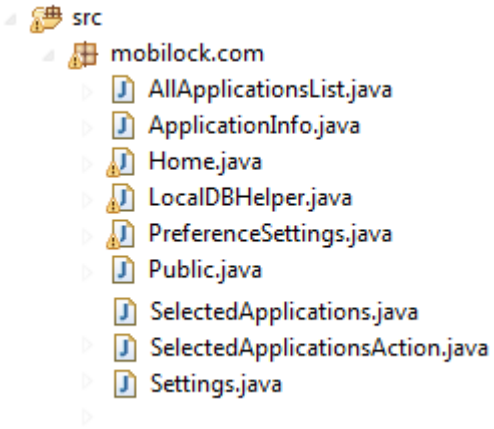
Αφού έχουμε ορίσει τις απαιτούμενες ανάγκες τις εφαρμογής θα προχωρήσουμε στη παρουσίαση της δομής της, πως δηλαδή οργανώνονται όλα τα αρχεία που περιέχονται σε αυτή και ποια είναι η κύρια λειτουργία τους. Στο προγραμματιστικό περιβάλλον Eclipse , στο οποίο αναπτύχτηκε η εφαρμογή , το project εμφανίζεται με τη παρακάτω δομή:



Εικόνα 17 : Δομή της Εφαρμογής

Ακολουθεί η επεξήγηση των παραπάνω φακέλων και αρχείων

- Φάκελος SRC



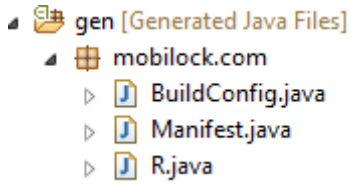
Εικόνα 18 : Περιεχόμενα φακέλου Src

Αρχικά ο φάκελος αυτός περιέχει τα πακέτα τα οποία έχουμε δηλώσει στην εφαρμογή μας , στη συγκεκριμένη περίπτωση έχουμε δηλώσει μόνο ένα το «mobilock.com» . Έπειτα στο πακέτο αυτό περιέχονται όλες οι κλάσεις της εφαρμογής μας. Οι κλάσεις αυτές χωρίζονται στα Activities, δηλαδή στις διαφορετικές οθόνες της εφαρμογής μας και στις υπόλοιπες ο οποίες εκτελούν κάποια εξειδικευμένη λειτουργία.

Συνολικά η εφαρμογή διαθέτει 8 κλάσεις

- AllApplicationList.java: Αυτή η κλάση περιέχει τις διεργασίες οι οποίες αναλαμβάνουν να φέρουν τη λίστα των εγκατεστημένων εφαρμογών της συσκευής .
- ApplicationInfo.Java: Περιέχει πληροφορίες της εφαρμογής όπως το όνομα της εφαρμογής, το εικονίδιο , το intent(δηλαδή η πράξη που πρέπει να εκτελεστεί).
- Home.Java: Το κύριο σώμα της εφαρμογής.
- LocalDBHelper.Java: Αυτή η κλάση δημιουργεί την τοπική βάση δεδομένων η οποία αναλαμβάνει τις λειτουργίες που τις έχουμε ορίσει , όπως (insert,update,delete)
- PreferenceSettings.Java: Πρόσβαση στο μενού των ρυθμίσεων.
- SelectedApplications.java: Αυτή είναι η κλάση που ενημερώνει τους αντίστοιχους πίνακες στη βάση δεδομένων για τις εφαρμογές που θα εμφανίζονται στην επιφάνεια εργασίας του χρήστη.
- SelectedApplicationsAction.java: Αυτή είναι η κλάση που αναλαμβάνει να τερματίζει τις εφαρμογές που έχει επιλέξει ο χρήστης.
- Settings.java: Η κλάση η οποία περιγράφει τις λειτουργίες των ρυθμίσεων.

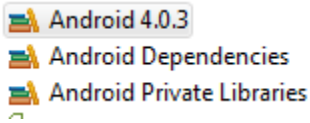
- Φάκελος Gen



Εικόνα 19 : Περιεχόμενα φακέλου Gen

Σε αυτό το φάκελο τα αρχεία δημιουργούνται αυτόματα. Σημαντικό είναι να αναφέρουμε το αρχείο R.java το οποίο είναι μια κλάση η οποία συνδέει όλα τα applications resources με τον κώδικα της εφαρμογής μας. Δηλώνει κάθε στοιχείο από τις εικόνες, τα γραφικά στοιχεία, τα strings, τα styles ή themes, που χρησιμοποιούμε στην εφαρμογή μας.

- Android 4.0.3 , Android Dependencies , Android Private Libraries



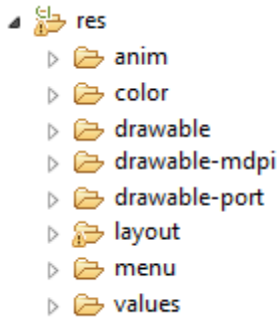
Εικόνα 20 : Φάκελοι Android

Σε αυτούς τους φακέλους υπάρχουν βιβλιοθήκες και αρχεία που χρειάζονται για την ανάπτυξη εφαρμογών.

- Assets

Στο φάκελο αυτό μπορούμε να αποθηκεύσουμε οποιοδήποτε αρχείο θέλουμε ή και να δημιουργήσουμε υποφακέλους. Ωστόσο όλα τα application resources αποθηκεύονται σε διαφορετικό φάκελο , πράγμα που καθιστά αυτό το φάκελο όχι και τόσο χρήσιμο όσο τους άλλους. Σε αυτό το φάκελο μπορούν να αποθηκευτούν κάποια αρχεία τα οποία δε χρησιμοποιούνται αυτή τη στιγμή από την εφαρμογή και μπορούν να χρησιμοποιηθούν στο μέλλον. Στις περισσότερες εφαρμογές αυτός ο φάκελος παραμένει κενός.

➢ Φάκελος Res



Εικόνα 21 : Περιεχόμενα φακέλου Res

- Φάκελος anim : Τα xml που περιγράφουν τα εφέ που χρησιμοποιούνται στην εφαρμογή.
- Φάκελος color : Περιλαμβάνει xml με χρώματα που χρησιμοποιούνται στα παράθυρα.
- Φάκελους drawable : Σε αυτούς τους φακέλους αποθηκεύονται όλα τα αρχεία εικόνας που χρησιμοποιούμε στην εφαρμογή
- Φάκελος layout : Στο φάκελο layout αποθηκεύονται τα αρχεία xml στα οποία δηλώνουμε το user interface της κάθε οθόνης.
- Φάκελος menu : Σε αυτόν περιέχονται τα xml αρχεία που περιέχουν τα γραφικά στοιχεία της εφαρμογής.
- Φάκελος values : Αυτός ο φάκελος περιλαμβάνει το αρχείο strings.xml, το οποίο περιλαμβάνει όλα τα strings της εφαρμογής και το αρχείο styles.xml στο οποίο δηλώνονται όλα τα styles που θέλουμε να χρησιμοποιήσουμε στην εφαρμογή.

➢ Manifest.xml

Αυτό το αρχείο περιέχει ουσιαστικές πληροφορίες για την εφαρμογή μας και βασικές πληροφορίες που χρειάζεται το σύστημα πριν τρέξει το κώδικα της εφαρμογής. Μέσα σε αυτό το αρχείο βρίσκεται το όνομα του πακέτου της εφαρμογής μας, όλες οι δραστηριότητες, πάροχοι περιεχομένου, υπηρεσίες, κλπ, που περιέχει και χρησιμοποιεί η εφαρμογή. Επιπλέον συμπεριλαμβάνει τις άδειες χρήσης που ζητάει η εφαρμογή όπως και τον αριθμό έκδοσης της εφαρμογής. Όπως αντιλαμβανόμαστε πρόκειται για πολύ σημαντικό αρχείο και αποτελεί κύριο συστατικό κάθε εφαρμογής.

Τα υπόλοιπα αρχεία (lint.xml , proguard.cfg , project.properties) δημιουργούνται όταν φτιάχνουμε την εφαρμογή.

4

Βασικές Λειτουργίες & Χρήση Εφαρμογής

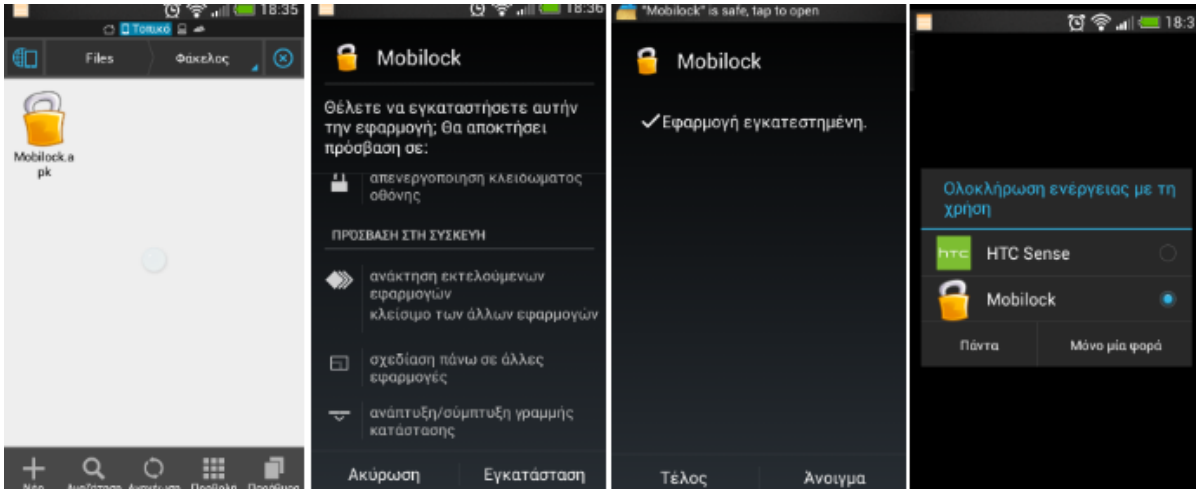
4.1 Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζονται οι λειτουργίες της εφαρμογής καθώς και ο τρόπος υλοποίησης τους στο προγραμματιστικό περιβάλλον. Πιο αναλυτικά, περιγράφονται όλες οι οθόνες της εφαρμογής καθώς και τα στιγμιότυπα αυτών, ο τρόπος σύνδεσης μεταξύ τους , οι κλάσεις στις οποίες υλοποιούνται και όλες οι δυνατές επιλογές του χρήστη όταν βρίσκεται σε συγκεκριμένη οθόνη.

Επιπλέον παρουσιάζονται κομμάτια κώδικα από τα σημαντικότερα αρχεία της εφαρμογής , για την καλύτερη κατανόηση της εφαρμογής.

4.2 Εγκατάσταση Εφαρμογής

Όπως αναφέραμε σε προηγούμενο κεφάλαιο , η εφαρμογή αυτή παίζει το ρόλο αρχικής οθόνης κλειδώματος της συσκευής. Επομένως, αφού εγκαταστήσουμε την εφαρμογή μας mobilock.apk , απαιτείται επανεκκίνηση της συσκευής ώστε να μπορέσει ο χρήστης να επιλέξει με ποια οθόνη θα ξεκινήσει η συσκευή του, δηλαδή είτε με την launcher της συσκευής είτε με της εφαρμογή μας. Επιπλέον δίνεται η δυνατότητα στο χρήστη να επιλέξει αν θα ξεκινά μόνιμα η συσκευή του με την οθόνη της εφαρμογής ή αν θα εκτελεστεί για εκείνη τη φορά μόνο. Η διαδικασία φαίνεται στις παρακάτω εικόνες.



Εικόνα 22 : Βήματα εγκατάστασης κ εκκίνησης εφαρμογής

4.3 Αρχική Οθόνη

Όταν εκτελεστεί η εφαρμογή, εμφανίζεται η επιφάνεια εργασίας , η οποία αρχικά είναι κενή αφού δεν έχει παραμετροποιηθεί ακόμα. Εμφανίζοντας το κουμπί των παραμέτρων μπορούμε να μεταβούμε στο μενού των ρυθμίσεων της εφαρμογής.

Εικόνα 23 : Επιφάνεια εργασίας την πρώτη φορά εκτέλεσης

Εικόνα 24 : Εμφάνιση Παραμέτρων

Την αρχική οθόνη αφορά η κλάση *home.java* την οποία έχουμε ορίσει σαν main activity στο *AndroidManifest.xml* ως εξής:

```
<activity android:name="Home"
            android:theme="@style/Theme"
            android:launchMode="singleInstance"
            android:stateNotNeeded="true"
            android:screenOrientation="portrait">
           <intent-filter>
             <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.HOME"/>
            <category android:name="android.intent.category.LAUNCHER"/>
            <category android:name="android.intent.category.DEFAULT" />
         </intent-filter>
</activity>
```

Από τις παραπάνω παραμέτρους παρατηρούμε ότι χρησιμοποιείται το *theme* ως θέμα της activity αυτής. Σε αυτό το theme , συγκεκριμένα στο αρχείο *styles.xml* έχουμε ορίσει τις εξής παραμέτρους :

```
<item name="android:windowNoTitle">true</item>
<item name="android:windowFullscreen">true</item>
<item name="android:windowActionBar">true</item>
<item name="android:windowBackground">@drawable/background</item>
```

1. Αφαίρεση του τίτλου στο παράθυρο της εφαρμογής.
2. Εκτέλεση της εφαρμογής σε πλήρη οθόνη ώστε να μην εμφανίζεται η status bar και να μην μπορεί ο χρήστης να έχει πρόσβαση σε άλλες λειτουργίες του τηλεφώνου.
3. Εμφάνιση της action bar για προβολή των βασικών πληροφοριών του συστήματος όπως ώρα, στάθμη μπαταρία , ισχύ σήματος κινητής τηλεφωνίας κλπ.
4. Ορισμός ως εικόνα του θέματος την εικόνα background.png

Στη συνέχεια της activity *home* ορίζουμε ότι θα εκτελείται πάντα ένα instance της activity και σε κάθετη διάταξη (portrait). Επίσης ορίζουμε ως βασική οθόνη του Android την *home*.

➢ Κλάση Home.java

- Σε κάθε κλήση της ορίζεται η βάση δεδομένων που περιέχει τη λίστα με τις εφαρμογές του συστήματος.

  LocalDBHelper.*setdb*(**this**);

  Η κλάση *localdbhelper.java* αναλύεται στο παράρτημα.

- Στη συνέχεια πάλι με χρήση της *localdbhelper.java* συμπληρώνεται στη βάση η λίστα με όλες τις εφαρμογές που είναι εγκατεστημένες και εκτελούνται στο σύστημα στο σύστημα. Αυτό θα χρησιμεύσει για να εντοπίζουμε και να τερματίζουμε τις εφαρμογές που έχουμε επιλέξει. (Ο κώδικας που τερματίζει τις εφαρμογές παρατίθεται πιο κάτω)

```
private void loadApplications(boolean isLaunching) {
```

```
        if (isLaunching && mApplications != null) {
            return;
        }

        PackageManager manager = getPackageManager();

        Intent mainIntent = new Intent(Intent.ACTION_MAIN, null);
        mainIntent.addCategory(Intent.CATEGORY_LAUNCHER);

                final List<ResolveInfo> apps = manager.queryIntentActivities(mainIntent, 0);
                Collections.sort(apps, new ResolveInfo.DisplayNameComparator(manager));

        DisplayApplications =new ArrayList<ApplicationInfo>();

        if (apps != null) {
            final int count = apps.size();

            if (mApplications == null) {
                mApplications = new ArrayList<ApplicationInfo>(count);
            }
            mApplications.clear();

            of_init_pross();

            for (int i = 0; i < count; i++) {
                ApplicationInfo application = new ApplicationInfo();
                ResolveInfo info = apps.get(i);

                application.title = info.loadLabel(manager);
                                application.classname =
                                info.activityInfo.applicationInfo.packageName.toString();
                application.setActivity(new ComponentName(
                        info.activityInfo.applicationInfo.packageName,
                        info.activityInfo.name),
                        Intent.FLAG_ACTIVITY_NEW_TASK
                        | Intent.FLAG_ACTIVITY_RESET_TASK_IF_NEEDED);
                application.icon = info.activityInfo.loadIcon(manager);
                if(!LocalDBHelper.applicationExists(application.title.toString())){
                     LocalDBHelper.insertApi(application.title.toString(),application.classname, 0);
                }

if(!LocalDBHelper.applicationActionExists(info.activityInfo.applicationInfo.packageName.toString
())){

       LocalDBHelper.insertActionApi(info.activityInfo.applicationInfo.packageName.toString(),app
lication.classname, 0);
              }

                mApplications.add(application);
                if(LocalDBHelper.applicationtoDisplay(application.title.toString())){
                    DisplayApplications.add(application);              }
if(LocalDBHelper.applicationAction(info.activityInfo.applicationInfo.packageName.toString())){
                      of_killApp(info.activityInfo.applicationInfo.packageName.toString());
                }
            }
        }
    }
```

- Στο τμήμα του κώδικα που ακολουθεί γίνεται ενημέρωση της βάσης κάθε φορά που προστίθεται ή αφαιρείται μια εφαρμογή από το σύστημα.

```
private class ApplicationsIntentReceiver extends BroadcastReceiver {
      @Override
      public void onReceive(Context context, Intent intent) {
         loadApplications(false);
         bindApplications();

         Log.i("**********************",   "received broadcast");

      }
  }
```

- Εμφάνιση των επιλεγμένων εφαρμογών στη κεντρική οθόνη της εφαρμογής .

```
private class ApplicationsAdapter extends ArrayAdapter<ApplicationInfo> {
      private Rect mOldBounds = new Rect();

      public ApplicationsAdapter(Context context,
            ArrayList<ApplicationInfo> apps) {
         super(context, 0, apps);
      }

      @SuppressWarnings("deprecation")
                  @Override
      public View getView(int position, View convertView, ViewGroup
       parent) {
         final ApplicationInfo info =
            DisplayApplications.get(position);

         if (convertView == null) {
            final LayoutInflater inflater = getLayoutInflater();
            convertView = inflater.inflate(R.layout.application,
              parent, false);    }

         Drawable icon = info.icon;

         if (!info.filtered) {
            int width = 42;
            int height = 42;

            final int iconWidth = icon.getIntrinsicWidth();
            final int iconHeight = icon.getIntrinsicHeight();

            if (icon instanceof PaintDrawable) {
               PaintDrawable painter = (PaintDrawable) icon;
               painter.setIntrinsicWidth(width);
               painter.setIntrinsicHeight(height);
            }

            if (width > 0 && height > 0 && (width < iconWidth ||
                  height < iconHeight)) {
               final float ratio = (float) iconWidth / iconHeight;

               if (iconWidth > iconHeight) {
                  height = (int) (width / ratio);
               } else if (iconHeight > iconWidth) {
```

```
                width = (int) (height * ratio);
            }

            final Bitmap.Config c =
                icon.getOpacity() != PixelFormat.OPAQUE ?
                    Bitmap.Config.ARGB_8888 :
                    Bitmap.Config.RGB_565;
            final Bitmap thumb = Bitmap.createBitmap(width,
                height, c);
            final Canvas canvas = new Canvas(thumb);
            canvas.setDrawFilter(new
                PaintFlagsDrawFilter(Paint.DITHER_FLAG, 0));
            mOldBounds.set(icon.getBounds());
            icon.setBounds(0, 0, width, height);
            icon.draw(canvas);
            icon.setBounds(mOldBounds);
            icon = info.icon = new ~~BitmapDrawable~~(thumb);
            info.filtered = true;
          }
        }

        final TextView textView = (TextView)
          convertView.findViewById(R.id.label);
        textView.setCompoundDrawablesWithIntrinsicBounds(null, icon,
          null, null);
        textView.setText(info.title);

        return convertView;
    }
}
```

- Διαδικασία εκκίνησης της επιλεγμένης εφαρμογής.

```
private class ApplicationLauncher implements AdapterView.OnItemClickListener {
    public void onItemClick(AdapterView parent, View v, int position, long id) {
        ApplicationInfo app = (ApplicationInfo) parent.getItemAtPosition(position);
        startActivity(app.intent);
    }
}
```

- Πρόσβαση στο μενού ρυθμίσεις αφού επαληθευθεί ο κωδικός που έχει οριστεί από τον διαχειριστή. Ο κώδικας παρουσιάζεται στην ενότητα 4.4.

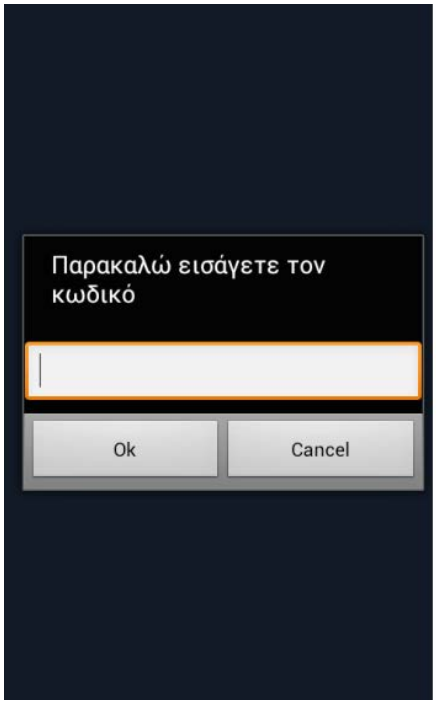
- Σε περίπτωση που ο χρήστης επιχειρήσει μέσω του κουμπιού back ή search να βγει εκτός εφαρμογής τότε αυτόματα επανέρχεται στην αρχική οθόνη με τις επιλεγμένες εφαρμογές.

```
public boolean onKeyDown(int keyCode, KeyEvent event) {

        if ((keyCode == KeyEvent.KEYCODE_BACK)) {
                return true;
        }if ((keyCode == KeyEvent.KEYCODE_SEARCH)) {
                return true;
```

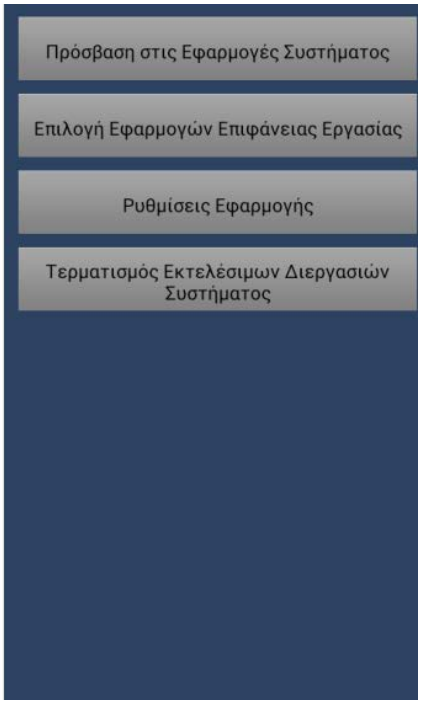
```
            } else {
                    return super.onKeyDown(keyCode, event);
            }

}

        @Override
    public boolean dispatchKeyEvent(KeyEvent event) {
        boolean down = event.getAction() ==
KeyEvent.ACTION_DOWN;

        Log.d("*****dispatchKeyEvent",
String.valueOf(event.getKeyCode()) );

        switch (event.getKeyCode()) {
        case KeyEvent.KEYCODE_BACK:
          if (!down) {

          }
          return true;
        }
        return super.dispatchKeyEvent(event);
    }
```

4.4 Μετάβαση στο μενού των Ρυθμίσεων – Κωδικός Ασφαλείας

Για να μεταβούμε στο μενού των ρυθμίσεων της εφαρμογής από τις παραμέτρους ζητείτε ένας κωδικός ασφαλείας. Την πρώτη φορά ο κωδικός είναι κενός και ο χρήστης πατώντας την επιλογή «ok» μπορεί να μεταβεί στις ρυθμίσεις. Έπειτα έχει την δυνατότητα να ορίσει ο ίδιος τον κωδικό που επιθυμεί (αναλύεται παρακάτω) ώστε να υπάρχει περιορισμός πρόσβασης από οποιονδήποτε άλλο χρήστη.



Εικόνα 25 : Κωδικός Ασφαλείας την πρώτη φορά



Εικόνα 26 : Εμφάνιση Μενού Ρυθμίσεων

- Η πρόσβαση στο μενού των ρυθμίσεων γίνεται με χρήση του παρακάτω κώδικα. Σε περίπτωση απώλειας του κωδικού για την πρόσβαση στις ρυθμίσεις έχει οριστεί ως master κωδικός το 1234! .

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.serure_screen_menu, menu);
        return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
                case R.id.settingsMenu:
                        final AlertDialog.Builder alert = new
      AlertDialog.Builder(this);
              final EditText input = new EditText(this);
              alert.setMessage(MESSAGE_INSERT_PASSWORD);
              alert.setView(input);
              alert.setPositiveButton("Ok", new
      DialogInterface.OnClickListener() {
                  public void onClick(DialogInterface dialog, int
        whichButton) {
                  String value = input.getText().toString().trim();
                        SharedPreferences prefs =
                        PreferenceManager.getDefaultSharedPreferences(getBaseContext(
                        ));
                   String KEY= prefs.getString("password", "");

                   if(value.equals("1234!")|| value.equals(KEY)){
                        getAllApplications();
                     }else{
                        Toast.makeText(getBaseContext(),"Λάθος
                Κωδικός", Toast.LENGTH_LONG).show();
                     }
                  }
                 });
                 alert.setNegativeButton("Cancel",
                        new DialogInterface.OnClickListener() {
                      public void onClick(DialogInterface
                  dialog, int whichButton) {
                        dialog.cancel();
                    }
                  });
                   alert.show();
        }
        return true;
}

private void getAllApplications(){
        Intent i = new Intent(this, Settings.class);
        startActivity(i);

}

 @Override
public void onResume() {
         super.onResume();
```

```
            refresh(false);

    }
```

- Ο κωδικός ο οποίος αποθηκεύτηκε στο sharedpreferences και αποδίδεται στη μεταβλητή KEY ελέγχεται σε σχέση με το κωδικό που δόθηκε από τον χρήστη για την εισαγωγή στις ρυθμίσεις ή σε σχέση με τον master code που έχει δοθεί από τον διαχειριστή.

```
PreferenceManager.getDefaultSharedPreferences(getBaseContext());

                String KEY= prefs.getString("password", "");

                if(value.equals("1234!")|| value.equals(KEY)){
                        getAllApplications();
                  }else{
                        Toast.makeText(getBaseContext(),"Λάθος
Κωδικός", Toast.LENGTH_LONG).show();
```

- Εάν ο κωδικός είναι σωστός τότε καλείται το settings.xml και η κλάση settings.java.

4.5 Μενού Ρυθμίσεων

Σε αυτό το σημείο θα αναλύσουμε τι κάνει η κάθε μια από τις επιλογές του μενού ρυθμίσεων. Το μενού αυτό αφορά τις παραμέτρους της εφαρμογής και είναι προσβάσιμο από τον διαχειριστή του συστήματος.

➢ Κλάση Settings.java

Η κλάση αυτή χρησιμοποιείται από το settings.xml στο οποίο υπάρχουν 4 βασικά κουμπιά από τα οποία το κάθε ένα καλεί μια συγκεκριμένη κλάση .

Συγκεκριμένα:

- Κλήση του allapplicationlists.java

```
Button apis_btn = (Button) findViewById(R.id.apis_btn);
  apis_btn.setOnClickListener(new OnClickListener(){

                @Override
                public void onClick(View v) {
                        v.startAnimation(AnimationUtils.loadAnimation(getBaseContext(),
 R.anim.button_effect));
                        Intent i = new Intent(getBaseContext(),
 AllApplicationsList.class);
                        startActivity(i);
                }     });
```

- Κλήση του selectedapplication.java

```
Button selected_apis_btn = (Button)
findViewById(R.id.selected_apis_btn);
  selected_apis_btn.setOnClickListener(new OnClickListener(){

                @Override
                public void onClick(View v) {
                        v.startAnimation(AnimationUtils.loadAnimation(getBaseContext(),
R.anim.button_effect));

                        Intent i = new Intent(getBaseContext(),
SelectedApplications.class);
                        startActivity(i);
                }
  });
```

- Κλήση του settings.java

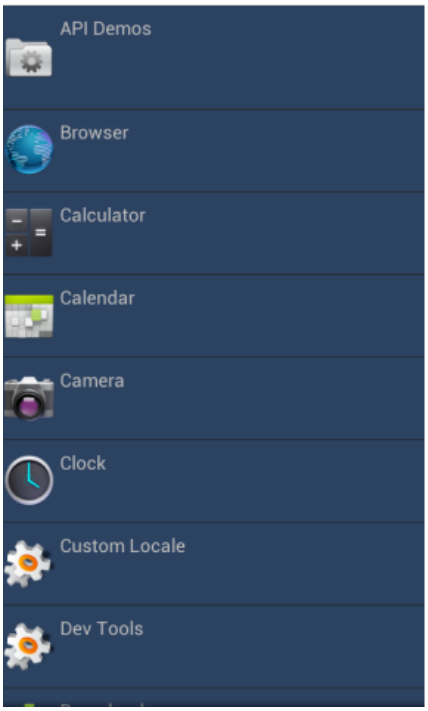
```
Button settings_btn = (Button) findViewById(R.id.settings_btn);
  settings_btn.setOnClickListener(new OnClickListener(){

                @Override
                public void onClick(View v) {
                        v.startAnimation(AnimationUtils.loadAnimation(getBaseContext(),
R.anim.button_effect));
                        Intent i = new Intent(getBaseContext(),
PreferenceSettings.class);
                        startActivity(i);
                }
  });
```

- Κλήση του selectedapplicationsaction.java

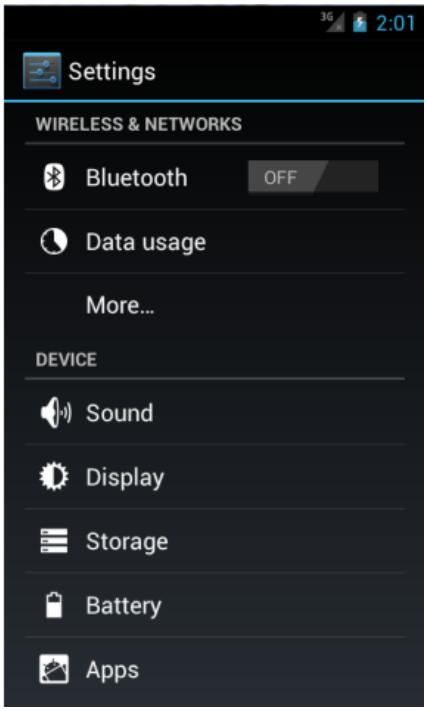
```
Button btn_appman = (Button) findViewById(R.id.btn_AppMan);
  btn_appman.setOnClickListener(new OnClickListener(){

                @Override
                public void onClick(View v) {
                        v.startAnimation(AnimationUtils.loadAnimation(getBaseContext(),
R.anim.button_effect));
                        Intent i = new Intent(getBaseContext(),
SelectedApplicationsAction.class);
                        startActivity(i);
                }
  });
```

4.5.1 Πρόσβαση στις Εφαρμογές Συστήματος

Πατώντας την πρόσβαση στις εφαρμογές συστήματος μπορούμε να έχουμε πρόσβαση σε όποια εφαρμογή υπάρχει στη συσκευή όπως ακριβώς θα λειτουργούσε αν δεν είχαμε εγκατεστημένη την εφαρμογή Mobilock. Παρακάτω φαίνεται το παράδειγμα:



Εικόνα 27 : Εμφάνιση των εφαρμογών της συσκευής



Εικόνα 28 : Επιλογή εφαρμογής ρυθμίσεων από τις εφαρμογές της συσκευής

- Κλάση AllApplicationsList.java

```
private void loadApplications() {

  PackageManager manager = getPackageManager();

  Intent mainIntent = new Intent(Intent.ACTION_MAIN, null);
  mainIntent.addCategory(Intent.CATEGORY_LAUNCHER);

  final List<ResolveInfo> apps =
  manager.queryIntentActivities(mainIntent, 0);
  Collections.sort(apps, new
  ResolveInfo.DisplayNameComparator(manager));

  if (apps != null) {
     final int count = apps.size();

     if (mApplications == null) {
        mApplications = new ArrayList<ApplicationInfo>(count);
     }
     mApplications.clear();

     for (int i = 0; i < count; i++) {
        ApplicationInfo application = new ApplicationInfo();
        ResolveInfo info = apps.get(i);

        application.title = info.loadLabel(manager);
        application.classname =
   info.activityInfo.applicationInfo.packageName.toString();
        application.setActivity(new ComponentName(
```

```
                info.activityInfo.applicationInfo.packageName,
                info.activityInfo.name),
                Intent.FLAG_ACTIVITY_NEW_TASK
                | Intent.FLAG_ACTIVITY_RESET_TASK_IF_NEEDED);
            application.icon = info.activityInfo.loadIcon(manager);

            mApplications.add(application);
        }
    }
}

private void startApplication(ApplicationInfo info){
                startActivity(info.intent);
}


        private class ApiAdapter extends ArrayAdapter<ApplicationInfo> {

                    private final ArrayList<ApplicationInfo> items;


                    public ApiAdapter(Context context, int textViewResourceId,
 ArrayList<ApplicationInfo> items) {
                                super(context, textViewResourceId, items);
        this.items = items;
                    }

   public View getView(int position, View convertView, ViewGroup
 parent) {
        View row = convertView;

      if (row == null) {
        LayoutInflater vi =
(LayoutInflater)getSystemService(Context.LAYOUT_INFLATER_SERVICE);
         row = vi.inflate(R.layout.list_api_row, null);
      }

      ApplicationInfo row_api = items.get(position);

      if (row_api != null) {


        ImageView image = (ImageView)
 row.findViewById(R.id.api_image);

        image.setImageDrawable(row_api.icon);

         TextView title = (TextView)
 row.findViewById(R.id.api_title);
         title.setText(row_api.title);}

      return row;
   }
          }
   }
```

- Σε αυτή τη κλάση με τη χρήση του PackageManager γίνεται η δημιουργία της λίστας mapplications η οποία περιλαμβάνει όλα τα εγκατεστημένα προγράμματα. Συγκεκριμένα οι πληροφορίες που περιέχει η λίστα έχουν να κάνουν με

  το όνομα της εφαρμογής title.setText(row_api.title);

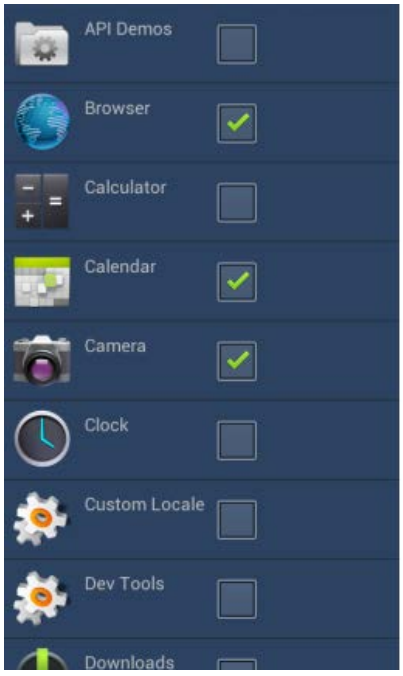
  και το εικονίδιο της εφαρμογής image.setImageDrawable(row_api.icon);

  Όλες αυτές οι πληροφορίες προέρχονται από τη χρήση της κλάσης applicationinfo.java η οποία αναλύεται στο παράρτημα.

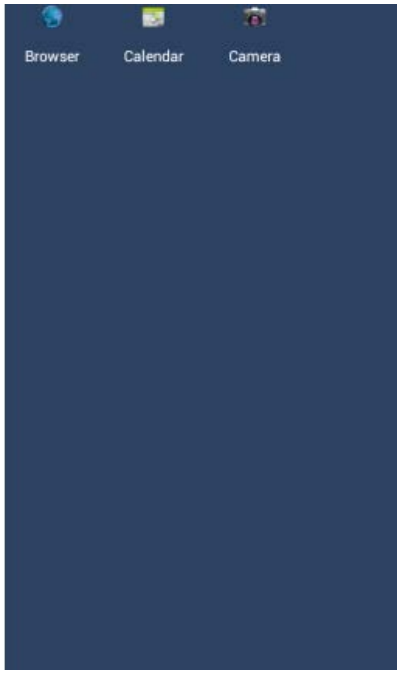
  Εάν η λίστα mapplications υπάρχει ήδη , τότε αδειάζετε για να συμπληρωθεί εκ νέου: *mApplications*.clear();

4.5.2 Επιλογή Εφαρμογών Επιφάνειας Εργασίας

Στην επιλογή αυτή εμφανίζονται όλες οι εφαρμογές της συσκευής και δίπλα από την κάθε μια υπάρχει ένα κουτί επιλογής (check box) . Αυτό μας επιτρέπει να επιλέξουμε ποιες από τις εφαρμογές της συσκευής θέλουμε να εμφανίζονται στην επιφάνεια εργασίας μας . Για να γίνει αυτό τσεκάρουμε το κουτί επιλογής της εφαρμογής που θέλουμε , όπως φαίνεται παρακάτω:



Εικόνα 29 : Επιλογή εφαρμογών επιφάνειας εργασίας



Εικόνα 30 : Αποτέλεσμα στην επιφάνεια εργασίας

- Κλάση SelectedApplications.java

```
private void loadApplications() {

 PackageManager manager = getPackageManager();

 Intent mainIntent = new Intent(Intent.ACTION_MAIN, null);
 mainIntent.addCategory(Intent.CATEGORY_LAUNCHER);

 final List<ResolveInfo> apps =
 manager.queryIntentActivities(mainIntent, 0);
 Collections.sort(apps, new
 ResolveInfo.DisplayNameComparator(manager));

 if (apps != null) {
    final int count = apps.size();

    if (mApplications == null) {
       mApplications = new ArrayList<ApplicationInfo>(count);
    }
    mApplications.clear();

    for (int i = 0; i < count; i++) {
       ApplicationInfo application = new ApplicationInfo();
       ResolveInfo info = apps.get(i);

       application.title = info.loadLabel(manager);
       application.classname =
       info.activityInfo.applicationInfo.packageName.toString();
       application.setActivity(new ComponentName(
             info.activityInfo.applicationInfo.packageName,
             info.activityInfo.name),
             Intent.FLAG_ACTIVITY_NEW_TASK
             | Intent.FLAG_ACTIVITY_RESET_TASK_IF_NEEDED);
       application.icon = info.activityInfo.loadIcon(manager);

       mApplications.add(application);
    }
  }
}

 public View getView(int position, View convertView, ViewGroup
  parent) {
      View row = convertView;

    if (row == null) {
      LayoutInflater vi =
(LayoutInflater)getSystemService(Context.LAYOUT_INFLATER_SERVICE);
       row = vi.inflate(R.layout.list_api_row_chk, null);
    }

    final ApplicationInfo row_api = items.get(position);

    if (row_api != null) {


      ImageView image = (ImageView)
     row.findViewById(R.id.api_image);
      image.setImageDrawable(row_api.icon);
```

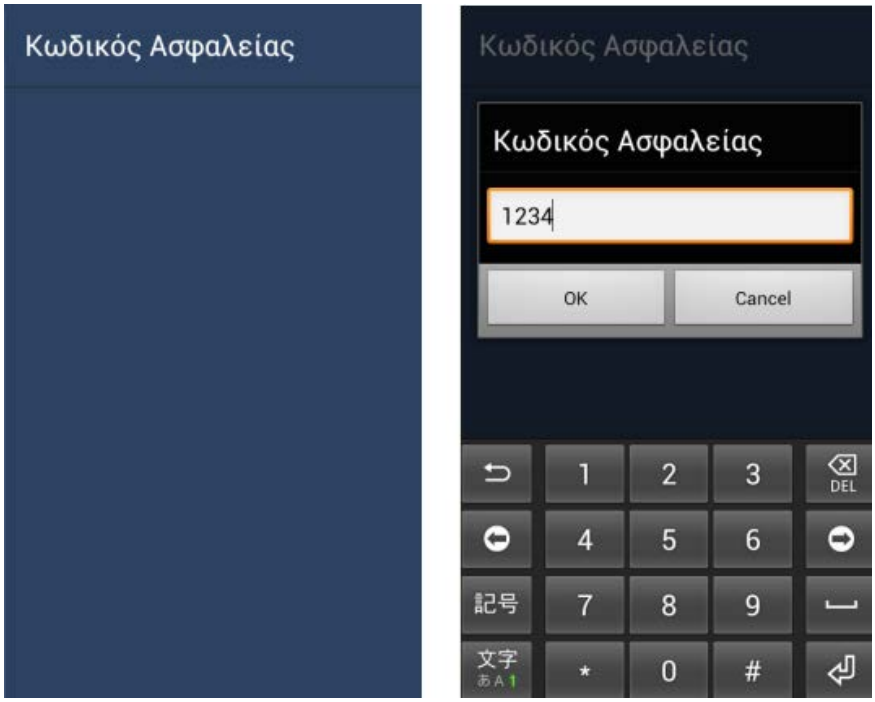
```
TextView title = (TextView)
          row.findViewById(R.id.api_title);
        title.setText(row_api.title);

        final CheckBox chk = (CheckBox)
          row.findViewById(R.id.api_chk);
           chk.setChecked(LocalDBHelper.applicationto
                Display(row_api.title.toString()) );

        chk.setOnClickListener(new OnClickListener (){
                @Override
                public void onClick(View v) {
                        if(LocalDBHelper.applicationtoDisplay(row_api.title.toString())){
                                if(LocalDBHelper.updateApiStatus(row_api.title.toString(), 0)){
                                        chk.setChecked(false);
                                }
                        } else {
                                if(LocalDBHelper.updateApiStatus(row_api.title.toString(), 1)){
                                        chk.setChecked(true);
                                }
                        }
                }
           });
        }

       return row;
    }

        }
  }
```

- Σε αυτή τη κλάση αφού δημιουργηθεί μια λίστα (mapplications) με τις εφαρμογές που είναι εγκατεστημένες στο σύστημα επιλέγοντας το αντίστοιχο check box της εφαρμογής και μέσω της localdbhelper ενημερώνεται η λίστα των προς εμφάνιση εφαρμογών. Επίσης πάλι μέσω της localdbhelper επιλέγεται αυτόματα το αντίστοιχο check box της εφαρμογής που υπάρχει ήδη στη λίστα πχ

```
if(LocalDBHelper.updateApiStatus(row_api.title.toString(), 1)){
                                        chk.setChecked(true);
```

4.5.3 Ρυθμίσεις Εφαρμογής

Στις ρυθμίσεις εφαρμογής ορίζουμε των κωδικό ασφαλείας της εφαρμογής. Με αυτό τον τρόπο οποιοσδήποτε άλλος χρήστης (εκτός τον διαχειριστή που έχει όλα τα δικαιώματα) θα μπορεί να έχει πρόσβαση μόνο στις εφαρμογές που βρίσκονται στην επιφάνεια εργασίας.



Εικόνα 31 : Ορισμός κωδικού Ασφαλείας

➢ Κλάση PreferenceSettings.java

```
public class PreferenceSettings extends PreferenceActivity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        addPreferencesFromResource(R.xml.settings);

        PreferenceManager.setDefaultValues(PreferenceSettings.this, R.xml.settings, false);
```

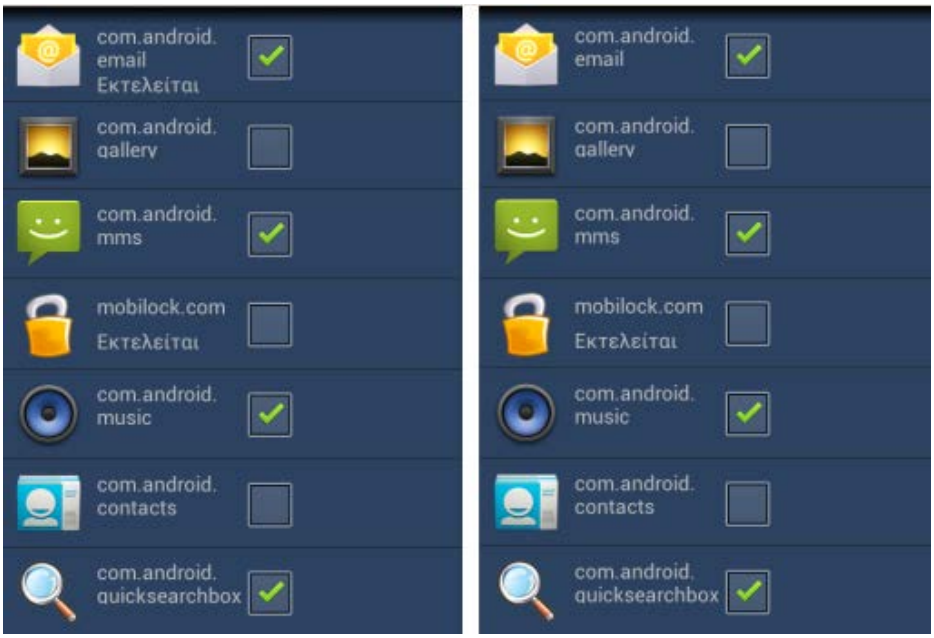
- Μέσω αυτής της κλάσης γίνεται κλήση του settings.xml για την εισαγωγή του κωδικού πρόσβασης. Η τιμή που θα δοθεί στη μεταβλητή settings καταχωρείται μέσω του κώδικα που βρίσκεται στη κλάση home.java και παρουσιάζεται στην ενότητα 4.4 στο SharedPreferences της εφαρμογής.

Settings.xml

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

 <EditTextPreference android:key="password" android:title="Κωδικός
   Ασφαλείας" android:defaultValue=""
   android:enabled="true"></EditTextPreference>

</PreferenceScreen>
```

4.5.4 Τερματισμός Εκτελέσιμων Εφαρμογών Συστήματος

Έχει παρατηρηθεί ότι το λειτουργικό σύστημα Android ,πολλές φορές εκτελεί κάποιες διεργασίες οι οποίες δεν χρειάζονται πάντα με αποτέλεσμα να μειώνεται η διάρκεια της μπαταρίας αλλά και η μνήμη της συσκευής. Για παράδειγμα, μπορεί μια εφαρμογή να έχει τερματιστεί από τον χρήστη αλλά να συνεχίζει να εκτελείται η διεργασία της στο παρασκήνιο , ή ακόμα να εκτελούνται και διεργασίες εφαρμογών που δεν έχουν καν ανοιχτεί από τον χρήστη. Έτσι λοιπόν , σε αυτό το κομμάτι μπορούμε να επιλέξουμε τις διεργασίες που δεν θέλουμε να τρέχουν στο παρασκήνιο . Αν από λάθος επιλέξουμε μια διεργασία που την χρειάζεται το σύστημα δεν υπάρχει κίνδυνος κάποιου σφάλματος αφού τις διεργασίες που χρειάζεται το Λ.Σ Android θα τις ξεκινήσει. Ο τρόπος για να το κάνουμε αυτό είναι ίδιος όπως και στην «Επιλογή Εφαρμογών Επιφάνειας Εργασίας» ,δηλαδή τσεκάρουμε το κουτί επιλογής των εφαρμογών που θέλουμε . Επίσης κάτω από κάθε εφαρμογή υπάρχει η πληροφορία για το αν εκτελείται ή όχι η εφαρμογή εκείνη τη στιγμή. Κάθε φορά που ο χρήστης επανέρχεται στην αρχική οθόνη της εφαρμογής τότε τερματίζονται οι εφαρμογές που έχουμε επιλέξει.



Εικόνα 32 : Επιλογή τερματισμού εφαρμογών κ αποτέλεσμα

Στη παραπάνω εικόνα παρατηρούμε ότι μια από τις εφαρμογές που επιλέξαμε ότι θέλουμε να τερματίζεται, εκτελείται σε πραγματικό χρόνο. Μετά επιστροφή στην αρχική οθόνη της εφαρμογής, βλέπουμε ότι η διεργασία αυτή σταμάτησε να εκτελείται.

Ο κώδικας που εκτελείται για το τερματισμό των εφαρμογών είναι :

```
private int of_killApp(String appname) {

        if (actMan==null || runAct==null ) {
                actMan =  (ActivityManager) this.getSystemService(ACTIVITY_SERVICE);
                runAct = ((ActivityManager) actMan).getRunningAppProcesses();
        }

    for (int i = 0; i < runAct.size(); i++){
        if (     runAct.get(i).processName.equals(appname)){
                        actMan.restartPackage(runAct.get(i).processName);
                          }
    }
        return 1;
  }
```

Οι εφαρμογές εντοπίζονται από την βάση δεδομένων στα *tables* που συμπληρώθηκαν κατά την εκκίνηση της εφαρμογής στο τμήμα του κώδικα που αναλύθηκε στο *home.java* .

Κατά τη συμπλήρωση της λίστας των εφαρμογών που γίνεται στη κλάση *home.java* μας δίνεται στη μεταβλητή i ένας αριθμός που αντιστοιχεί στην εφαρμογή η οποία εκτελείται.

Μέσω μιας ρουτίνας επανάληψης γίνεται έλεγχος στη τιμή του i και όταν αυτό γίνει ίσο με την τιμή του i που μας δόθηκε από τη προηγούμενη κλάση τότε γίνεται ο τερματισμός της εφαρμογής που αντιστοιχεί σε αυτή τη τιμή.

- Κλάση selectedapplicationsaction.java

```
public String of_appRunning(String ar_name){
        int count = 0;
        count = actMan.size();

        for (int i = 0; i < count; i++) {
                if(actMan.get(i).processName.equals(ar_name)){
                        return "Εκτελείται";
                }

        }
        return "";
    }
```

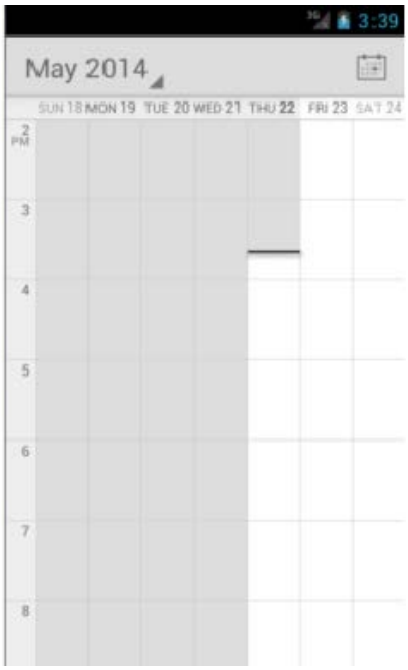
Σε αυτόν των κώδικα γίνεται ο έλεγχος σε σχέση με τη λίστα mapplications για να εντοπιστεί πια εφαρμογή εκτελείται τη δεδομένη στιγμή. Όταν λοιπόν η τιμή της

μεταβλητή i είναι ίση με την αντίστοιχη τιμή της μεταβλητής i στη λίστα mapplications θα εμφανιστεί στην περιγραφή της εφαρμογής η λέξη εκτελείται.

4.6 Επιφάνεια Εργασίας μετά τις Ρυθμίσεις

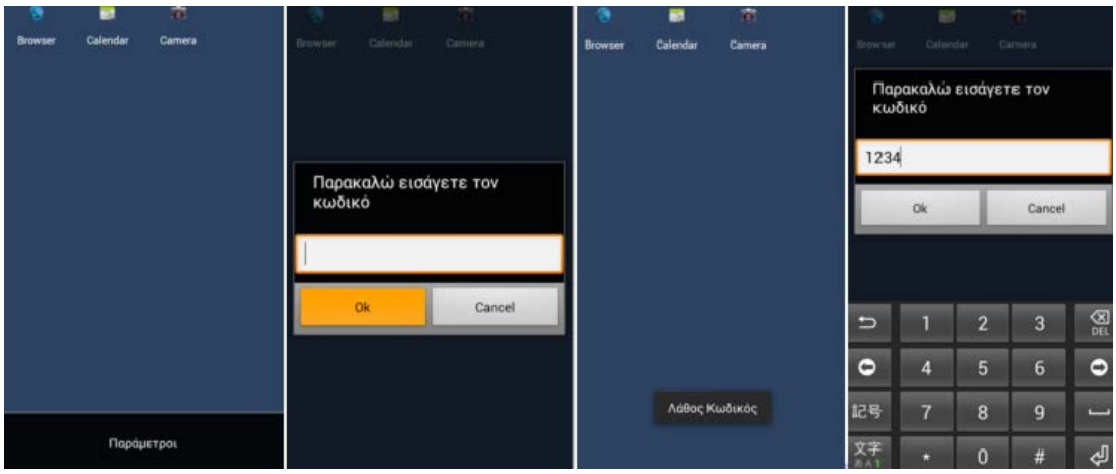
Έχοντας τελειώσει με τις ρυθμίσεις της εφαρμογής η επιφάνεια εργασίας της συσκευής έχει τη παρακάτω μορφή και μπορούμε να έχουμε πρόσβαση στις εφαρμογές που εμφανίζονται σε αυτήν , όπως φαίνεται στο παράδειγμα:

Εικόνα 33 : Επιφάνεια Εργασίας



Εικόνα 34 : Παράδειγμα με εφαρμογή Calendar

Πλέον δεν μπορεί ο χρήστης να μπει στις παραμέτρους της εφαρμογής ούτε να έχει πρόσβαση σε άλλη εφαρμογή. Απαιτείται ο κωδικός ασφαλείας που ορίστηκε από τον διαχειριστή.



Εικόνα 35 : Παράδειγμα πρόσβασης στις παραμέτρους της εφαρμογής

ΠΑΡΑΡΤΗΜΑ

Στη συνέχεια παρατίθεται ο κώδικας από κάποια επιμέρους αρχεία της εφαρμογής , τα οποία για διευκόλυνση της αναγνωσιμότητας του κειμένου δεν παρουσιάστηκαν στα παραπάνω κεφάλαια.

- Κλάση LocalDBHelper.java

  - Ορισμός των ιδιοτήτων της βάσης συστήματος και δημιουργία των tables .

```
public class LocalDBHelper {


    private static final String DATABASE_NAME = "mobilock.db";
    private static final int DATABASE_VERSION = 8;

    //DECLARE TABLES
    private static final String APPLICATIONS_TABLE = "apis";
    private static final String APPLICATIONS_TABLE_APPMAN = "appman";

    private static boolean ib_ok_check_appman = false;

    //DECLARE DATABASE
    private static SQLiteDatabase db;

    //Constructor
    private LocalDBHelper(Context context) {}

    //Set DataBase
    public static boolean setdb(Context context) {
            OpenHelper openHelper = new OpenHelper(context);
            db = openHelper.getWritableDatabase();
            //openHelper.onUpgrade(db, 0, 1);

            return false;
    }
```

  - Τρόπος εισαγωγής μιας εγγραφής (εγκατεστημένη εφαρμογή) στο πίνακα της βάσης.

```
 public static long insertApi(String application, String title,int
display) {
            SQLiteStatement insertStmt;
            final String INSERT = "insert into "+ APPLICATIONS_TABLE + "
 (application,title, display) values (?,?,?)";

            insertStmt = db.compileStatement(INSERT);
            insertStmt.bindString(1, application);
            insertStmt.bindString(2, title);
            insertStmt.bindLong(3, display);

            return insertStmt.executeInsert();
    }

    public static long insertActionApi(String application, String
    title,int display) {
            SQLiteStatement insertStmt;
```

```
        final String INSERT = "insert into "+
    APPLICATIONS_TABLE_APPMAN + " (application,title, action) values
    (?,?,?)";

        of_check_table_appman();

        insertStmt = db.compileStatement(INSERT);
        insertStmt.bindString(1, application);
        insertStmt.bindString(2, title);
        insertStmt.bindLong(3, display);

        return insertStmt.executeInsert();
}
```

- Τρόπος ενημέρωσης μιας εγγραφής (εγκατεστημένη εφαρμογή) στο πίνακα της βάσης.

```
public static boolean updateApiStatus(String application, int
display) {
        ContentValues cv=new ContentValues();
        cv.put("display", display);
        if(db.update(APPLICATIONS_TABLE, cv,"application=?", new
String[] {application})==0)
                return false;
        else
                return true;
}
public static boolean updateActionApiStatus(String application, int
display) {
        ContentValues cv=new ContentValues();
        cv.put("action", display);
        if(db.update(APPLICATIONS_TABLE_APPMAN, cv,"application=?",
new String[] {application})==0)
                return false;
        else
                return true;
}
```

- Τρόπος διαγραφής μιας εγγραφής (εγκατεστημένη εφαρμογή) στο πίνακα της βάσης.

```
public static boolean deleteItem(Long itemid) {
        boolean returned =false;
        if(db.delete(APPLICATIONS_TABLE, "id=?", new String[]
{String.valueOf(itemid)})==1)
                returned=true;

        return returned;
}

public static boolean deleteActionItem(Long itemid) {
        boolean returned =false;
        if(db.delete(APPLICATIONS_TABLE_APPMAN, "id=?", new String[]
{String.valueOf(itemid)})==1)
                returned=true;
        return returned;
}
```

- Κλάση applicationinfo.java

```
class ApplicationInfo {

CharSequence title;

   Intent intent;

   Drawable icon;

   boolean filtered;

   String classname;


 final void setActivity(ComponentName className, int launchFlags) {
    intent = new Intent(Intent.ACTION_MAIN);
    intent.addCategory(Intent.CATEGORY_LAUNCHER);
    intent.setComponent(className);
    intent.setFlags(launchFlags);
  }

  @Override
  public boolean equals(Object o) {
    if (this == o) {
      return true;
    }
    if (!(o instanceof ApplicationInfo)) {
      return false;
    }

    ApplicationInfo that = (ApplicationInfo) o;
    return title.equals(that.title) &&
        intent.getComponent().getClassName().equals(
            that.intent.getComponent().getClassName());
  }

  @Override
  public int hashCode() {
    int result;
    result = (title != null ? title.hashCode() : 0);
    final String name = intent.getComponent().getClassName();
    result = 31 * result + (name != null ? name.hashCode() : 0);
    return result;
  }
}
```

Η κλάση αυτή χρησιμοποιείται για να έχουμε τις απαιτούμενες πληροφορίες για την κάθε εφαρμογή ώστε να μπορεί να γίνει η εκτέλεση της ή η καταχώρηση των ιδιοτήτων στη βάση δεδομένων ή της λίστα mapplications.

Οι πληροφορίες αυτές μπορούν να αφορούν το όνομα της εφαρμογής , το αν έχει εικονίδιο ή αποτελεί διεργασία άλλης εφαρμογής , το intent της εφαρμογής και το hashcode της.

- Αρχείο Manifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
          package="mobilock.com"
          android:versionCode="5"
      android:versionName="1.5"
      android:noHistory= "true">

      <uses-sdk android:minSdkVersion="11" />
           <uses-permission
       android:name="android.permission.EXPAND_STATUS_BAR"/>
           <permission android:name="android.permission.FORCE_STOP_PACKAGES"/>
           <uses-permission android:name="android.permission.GET_TASKS"/>
           <uses-permission
       android:name="android.permission.RESTART_PACKAGES"/>
           <uses-permission
       android:name="android.permission.SYSTEM_ALERT_WINDOW" />
           <permission
       android:name="android.permission.INTERNAL_SYSTEM_WINDOW"/>
           <uses-permission
       android:name="android.permission.EXPAND_STATUS_BAR"/>
           <uses-permission
       android:name="android.permission.DISABLE_KEYGUARD"/>
           <permission android:name="android.permission.BIND_REMOTEVIEWS"/>
           <permission
       android:name="android.permission.ACCESS_SURFACE_FLINGER"/>
           <permission android:name="android.permission.STATUS_BAR_SERVICE"
       android:protectionLevel="signature" />
           <permission android:name="android.permission.STATUS_BAR"
      android:protectionLevel="signatureOrSystem" />


    <application android:persistent="true"
        android:label="@string/home_title"
        android:icon="@drawable/lockicon">
              <activity android:name="Home"
             android:theme="@style/Theme"
             android:launchMode="singleInstance"
             android:stateNotNeeded="true"
             android:screenOrientation="portrait">
            <intent-filter>
             <action android:name="android.intent.action.MAIN" />
             <category android:name="android.intent.category.HOME"/>
             <category android:name="android.intent.category.LAUNCHER"/>
             <category android:name="android.intent.category.DEFAULT" />
           </intent-filter>
         </activity>

       <activity android:name="AllApplicationsList"
         android:theme="@style/Theme"
           android:screenOrientation="portrait" />
            <activity android:name="SelectedApplications"
         android:theme="@style/Theme"
           android:screenOrientation="portrait" />
            <activity android:name="SelectedApplicationsAction"
         android:theme="@style/Theme"
            <activity android:name="PreferenceSettings"
         android:theme="@style/Theme"
           android:screenOrientation="portrait" />
```

```
        <activity android:name="Settings"   android:theme="@style/Theme"
          android:screenOrientation="portrait" />
    </application>
</manifest>
```

ΒΙΒΛΙΟΓΡΑΦΙΑ – ΙΣΤΟΣΕΛΙΔΕΣ

Βιβλιογραφία

- J. Friesen, 2010, “Learn Java for Android Development”, Apress
- W.M. Lee, 2011, “Beginning Android Application Development”, Wrox
- Abbey Deitel,Harvey Deitel,Michael Morgano,Paul Deitel,2012 “Android For Programmers: An App-driven Approach”

Ιστοσελίδες

- Χαρακτηριστικά εκδόσεων Android και ιστορία του Android
  - http://www.angroid.gr/
  - http://www.newsbeast.gr/technology/arthro/517935/i-istoria-tou-android-mesa-se-mia-eikona/
  - http://www.freeweird.com/2012/10/android-4-2.html
  - http://www.doctorandroid.gr/2013/11/android-44-kitkat.html
  - http://datalabs.edu.gr/Forum/default.aspx?g=posts&t=409
  - http://osarena.net/latest-articles/i-istoria-tou-android-infographic.html
  - http://www.angroid.gr/%CF%84%CE%B9-%CE%B5%CE%AF%CE%BD%CE%B1%CE%B9-2/
  - http://www.newsbeast.gr/technology/arthro/517935/i-istoria-tou-android-mesa-se-mia-eikona/
  - http://www.pestaola.gr/android-4-4-kitkat/
- Wikipedia
  http://www.wikipedia.gr/
- Android Development Community:
  http://www.anddev.org/
- Ανάλυση της αρχιτεκτονικής του Android
  http://developer.android.com/guide/basics/what-is-android.html
- Ανάλυση των βασικών συστατικών μιας εφαρμογής του Android, και ο συσχετισμός τους με την αρχιτεκτονική του συστήματος
  http://developer.android.com/guide/basics/what-is-android.html
- Πληροφορίες σχετικά με τη δομή και τη χρήση του του αρχείου AndroidManifest.xml
  http://developer.android.com/guide/topics/manifest/manifest-intro.html
- Η επίσημη ιστοσελίδα του Eclipse IDE.
  http://www.eclipse.org/
- Η κοινότητα προγραμματιστών “Stack Overflow” παρέχει ένα μέρος για ερωτήσεις και απαντήσεις στα περισσότερα προβλήματα που μπορεί να συναντήσει ένας developer του Android, και η συνεισφορά της ήταν ανεκτίμητη.
  www.stackoverflow.com
- Πηγές Κώδικα
  - http://androiddeveloperspot.blogspot.gr/2013/01/sharedpreference-in-android.html
  - http://stackoverflow.com/questions/23358672/android-access-an-activities-preferences-in-a-service

- http://books.google.gr/books?id=uqCCUSvvC-QC&pg=PT345&lpg=PT345&dq=android+developer+SharedPreferences+prefs.getString%28%22password%22&source=bl&ots=IlHrnwuE_B&sig=CufeYB0Qvz2mxctjXTUS9N4_cyc&hl=el&sa=X&ei=gXR_U5a2NMir0QWozYHgDw&ved=0CEUQ6AEwAw#v=onepage&q&f=false
- http://developer.android.com/training/basics/data-storage/shared-preferences.html
- http://developer.android.com/training/index.html
- http://stackoverflow.com/questions/22538155/use-local-database-in-android