ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Σχεδίαση και ανάπτυξη Smartphone εφαρμογών με σύγχρονες τεχνολογίες σε Android Studio

Βιολέττα Σταύρου

Αλέξανδρος Θεοδώρου

Εισηγητής: Δρ. Κωνσταντίνος Κουκουλέτσος, Καθηγητής

ΑΘΗΝΑ

ΙΑΝΟΥΑΡΙΟΣ 2019

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Σχεδίαση και ανάπτυξη Smartphone εφαρμογών με σύγχρονες τεχνολογίες σε Android Studio

Βιολέττα Σταύρου
Α.Μ. 44973
Αλέξανδρος Θεοδώρου
Α.Μ. 44979

Εισηγητής:
Δρ. Κωνσταντίνος Κουκουλέτσος, Καθηγητής

Εξεταστική Επιτροπή:

Ημερομηνία Εξέτασης

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Οι κάτωθι υπογεγραμμένοι Σταύρου Βιολέττα, του Φωτίου, με αριθμό μητρώου 44973 και Θεοδώρου Αλέξανδρος, του Εμμανουήλ, με αριθμό μητρώου 44979 φοιτητές του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών του Πανεπιστημίου Δυτικής Αττικής πριν αναλάβουμε την εκπόνηση της Πτυχιακής Εργασίας μας, δηλώνουμε ότι ενημερωθήκαμε για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο των συγγραφέων, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικοί υπεύθυνοι είναι οι συγγραφείς της Π.Ε., οι οποίοι φέρουν και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών των συγγραφέων σε περίπτωση που το Ίδρυμα τους έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφασης της, μετά από αίτηση των ενδιαφερόμενων, τους αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

ΕΥΧΑΡΙΣΤΙΕΣ

Θα θέλαμε να ευχαριστήσουμε στον καθηγητή μας, κύριο Κ. Κουκουλέτσο για την επικοινωνία του μαζί μας και για την ομαλή πορεία της πτυχιακής μας εργασίας.

Τέλος, να ευχαριστήσουμε πάρα πολύ τις οικογένειές μας για τη στήριξή τους όλα αυτά τα χρόνια της φοιτητικής μας ζωής και που έκαναν ό,τι περνούσε από το χέρι τους για να πάρουμε τα σωστά εφόδια ώστε να έχουμε τη δυνατότητα για ένα καλύτερο μέλλον!

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία έχει ως αντικείμενο την ανάπτυξη μιας εφαρμογής για κινητά τηλέφωνα που χρησιμοποιούν το λογισμικό Android της Google. Η επιλογή του θέματος έγινε σύμφωνα με το γεγονός ότι στη σύγχρονη εποχή πάρα πολλοί χρήστες χρησιμοποιούν το λειτουργικό Android. Οπότε, αποφασίσαμε να δημιουργήσουμε μία εφαρμογή με θέμα το ηλεκτρονικό ξυπνητήρι για συσκευές που υποστηρίζουν το λειτουργικό Android 8.0 και άνω. Σκοπό έχει την παροχή υπηρεσιών αφύπνισης στους χρήστες με έναν πιο σύγχρονο σχεδιασμό και με πολλες διαφορετικές λειτουργίες. Τέλος, για την ανάπτυξη της εφαρμογής μας χρησιμοποιήσαμε το εργαλείο Android Studio καθώς θεωρείται το καλύτερο και ισχυρότερο εργαλείο για την ανάπτυξη εφαρμογών.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Δημιουργία εφαρμογών Android
ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Android, λογισμικό, ξυπνητήρι, εφαρμογή, συσκευές.

ABSTRACT

This project is intented to develop an application for mobile phones using the Google Android software. The choice of this theme was made according to the fact that in modern times too many users use the Android software. So, we decided to create an electronical alarm application for devices that support Android 8.0 and above. The goal is to provide wake-up services to users with a more modern design and many different functions. Finally, for the development of our app we used the Android Studio tool as it is considered the best and most powerful application development tool

SCIENTIFIC AREA: Android application development

KEY WORDS: Android, software, electronic alarm, application, devices.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1

ΛΟΓΙΣΜΙΚΟ ANDROID

1.1 Εισαγωγή

Στο πρώτο κεφάλαιο θα δούμε τι είναι το λογισμικό Android, πως δημιουργήθηκε, ποιες είναι οι εκδόσεις του και οι διαφορές του με το iOS.

1.2 Ιστορική αναδρομή του λογισμικού Android

Η Google όταν εντόπισε την αυξημένη χρήση του internet και των αναζητήσεων στον παγκόσμιο ιστό μέσω κινητών συσκευών (mobile devices) εξαγόρασε, το **2005**, την Android Inc με σκοπό την ανάπτυξη μιας πλατφόρμας για τέτοιου είδους συσκευές. Περίπου την ίδια περίοδο, η Apple παρουσίασε το iPhone (**2007**), το οποίο υποστήριζε multitouch (πολλαπλή αφή). Το Android προσαρμόστηκε γρήγορα με σκοπό να μπορεί να υιοθετήσει και να παρέχει τις δυνατότητες που αναρφέρθηκαν παραπάνω αν και κατά κοινή ομολογία οι πρώτες εκδόσεις του υπολείπονταν του iPhone όσον αφορά τις υποστηριζόμενες λειτουργίες και τη συνολική εμπειρία που πρόσφερε στους χρήστες. Το 2007, τη χρονιά που λανσαρίστηκε το iPhone, δημιουργήθηκε ένας οργανισμός που αποτελούνταν από πολλές εταιρείες τηλεπικοινωνιακού εξοπλισμού καθώς και από εταιρείες πληροφορικής όπως είναι: η Google, η T-Mobile, η Motorola, η Samsung, η Sony Ericsson, η Intel, η Vodafone, η Toshiba κ.α. Με όνομα Open Handset Alliance και σκοπό την έρευνα και την ανάπτυξη τεχνολογιών για την παραγωγή συσκευών που θα διευκολύνουν τους πάροχους κινητής τηλεφωνίας, τους κατασκευαστές κινητών τηλεφώνων, αλλά και τους προγραμματιστές εφαρμογών. Τα μέλη της συμμαχίας δεσμεύτηκαν να παρέχουν τις τεχνολογίες αυτές με βάση το μοντέλο ανοιχτού πηγαίου κώδικα Apache (Android - Digital Academy, n.d.).

Οι συσκευές Android άρχισαν να διαδίδονται πολύ γρήγορα, κυρίως λόγω της δυνατότητας της πλατφόρμας που εκμεταλλευόταν το μοντέλο cloud computing (υπολογιστικό νέφος), αλλά και της έμφυτης υποστήριξης για συνεργασία με μία σχεσιακή βάση δεδομένων (SQLite). Ακολούθησαν αρκετές αναβαθμισμένες

εκδόσεις του Android, όπου κάθε μία προσέθετε νέα χαρακτηριστικά και λειτουργίες. Για κάποιον άγνωστο λόγο, κάθε έκδοση του Android φέρει και μία κωδική ονομασία ενός γλυκού (1.5-Cupcake, 1.6-Donut, 2.0-Eclair κ.α.), ενώ από τις εκδόσεις αυτές, εκείνη που εισήγαγε την υποστήριξη πιο ανεπτυγμένων λειτουργιών ήταν σίγουρα η 2.0, στην οποία ενσωματώθηκε η υποστήριξη multitouch, HTML 5 (Hypertext Markup Language), text-to-speech (κείμενο σε ομιλία) και η δυνατότητα πιο προχωρημένων αναζητήσεων. Η πιο πρόσφατη έκδοση του λειτουργικού Android είναι η 9.0 (**Pie**) (Android - Digital Academy, n.d.).

1.3 Τι είναι το λογισμικό Android

Το Android είναι λειτουργικό σύστημα για συσκευές, το οποίο τρέχει στον πυρήνα του λειτουργικού Linux. Αναπτύχθηκε πρώτα από την Google, τον Οκτώβριο του 2003 και ύστερα απ' την Open Handset Alliance. Δίνει τη δυνατότητα στους κατασκευαστές λογισμικού να συνθέτουν κώδικα χρησιμοποιώντας τη γλώσσα προγραμματισμού Java, ελέγχοντας τη συσκευή τους μέσα απ' τις βιβλιοθήκες λογισμικού, τις οποίες έχει αναπτύξει η Google. Το Android σχεδιάστηκε κυρίως για συσκευές με οθόνη αφής, όπως είναι τα smartphone (έξυπνα κινητά), τα tablet (ταμπλέτες), με διαφορετικό περιβάλλον για τηλεοράσεις (Android TV), αυτοκίνητα (Android Auto) και ρολόγια χειρός (Android Wear). **Το λογισμικό Android είναι το πιο ευρέως διαδεδομένο λογισμικό στον κόσμο.** Επίσης, οι συσκευές με Android έχουν περισσότερες πωλήσεις απ' όλες τις συσκευές Windows, iOS (Operating System) και Mac OS X (Android - Βικιπαίδεια, 2018).

1.4 Android vs Iphone

Τη σήμερον ημέρα, η μάχη για την επικράτηση στην αγορά των smartphones μαίνεται μεταξύ δύο κολοσσών, του **iPhone** και του **Android**. Το iPhone δείχνει να έχει το προβάδισμα και οι κάτοχοί του διατείνονται πως προσφέρει **ανώτερη** εμπειρία στο χρήστη και **μεγαλύτερη** ποικιλία εφαρμογών. Κάτι τέτοιο μπορεί να ισχύει, μιας και το iPhone ήταν αυτό που κυκλοφόρησε πρώτο εισάγοντας

καινοτόμες τεχνολογίες στην αγορά. Ας εστιάσουμε λοιπόν ποια είναι τα θετικά και αρνητικά της κάθε πλατφόρμας (Android - Digital Academy, n.d.).

Εικόνα 1.4.1 – Η μάχη των δύο λογισμικών

Ένα από τα πιο δυνατά σημεία του Android είναι η ανάπτυξη εφαρμογών που γίνεται σχεδόν αποκλειστικά με τη χρήση της γλώσσας προγραμματισμού Java, ίσως την πιο δημοφιλή, πιο ολοκληρωμένη και πιο καλά δομημένη γλώσσα που υπάρχει σήμερα. Επιπλέον, η υλοποίηση των Android εφαρμογών μπορεί να γίνει με μικρό κόστος από την πλευρά του προγραμματιστή. Ο υποψήφιος Android developer (προγραμματιστής) καλό θα είναι να διαθέτει έναν προσωπικό υπολογιστή, σχεδόν απαραίτητο, με οποιοδήποτε λειτουργικό σύστημα επιθυμεί, ένα κινητό Android για τη σωστή δοκιμή των εφαρμογών του και τέλος έναν λογαριασμό developer στο Android Market, για τον οποίο απαιτείται η καταβολή ενός τιμήματος της τάξης των 25$ (Android - Digital Academy, n.d.).

Εικόνα 1.4.2 – Το logo του Android

Από την πλευρά του iPhone, η ανάπτυξη γίνεται σε γλώσσα προγραμματισμού Objective C, μια γλώσσα που αποτελεί υπερσύνολο της γλώσσας C. Νομίζω πως δεν θα διαφωνήσει κανείς πως όσον αφορά τη σύνταξη κώδικα σε C και Java, δεν υπάρχει κανένα περιθώριο σύγκρισης! Βέβαια το κόστος για να μπορεί κάποιος να υλοποιήσει εφαρμογές iPhone και να τις διαθέσει στην αγορά είναι

μεγαλύτερο. Απαιτείται ένας υπολογιστής Mac, ένα iPhone για τη δοκιμή των εφαρμογών του και ένας λογαριασμός developer, για τον οποίο θα πρέπει να καταβάλλει το ποσό των 100$ κάθε χρόνο (ισχύουσα τιμή κατά τη σύνταξη των σημειώσεων). Αν συγκρίνουμε τις τιμές των συσκευών, μία νέα συσκευή iPhone κοστίζει κατά μέσο όρο 450€ (ανάλογα με την έκδοση), ενώ είναι εφικτό να αποκτήσει κάποιος μια αξιοπρεπέστατη συσκευή Android με λιγότερα από 200€ (Android - Digital Academy, n.d.).

Εικόνα 1.4.3 – Το logo της Apple

Συνοψίζοντας όλα τα παραπάνω, διαπιστώνουμε πως το Android έχει σαφές προβάδισμα όσον αφορά το κόστος και τις τεχνολογίες που εμπλέκονται στην υλοποίηση των εφαρμογών του. Από την άλλη πλευρά, φαίνεται πως οι χρήστες των iPhone είναι περισσότερο διατεθιμένοι να πληρώσουν παραπάνω χρήματα για να αγοράσουν μια εφαρμογή απ’ ό,τι οι χρήστες Android (Android - Digital Academy, n.d.).

1.5 Αρχιτεκτονική του Android

Στον πυρήνα της πλατφόρμας Android βρίσκεται ένα Linux kernel (πυρήνας Linux), το οποίο ευθύνεται για τη διαχείριση των device drivers (οδηγοί συσκευών), τον έλεγχο πρόσβασης στους πόρους του συστήματος, τη διαχείριση μνήμης και τις υπόλοιπες υπηρεσίες που παρέχει ένα λειτουργικό σύστημα. Στους device drivers περιλαμβάνονται οι drivers: της οθόνης, του WiFi, της κάμερας, του ήχου κ.α. Ένα επίπεδο πιο επάνω βρίσκονται οι native βιβλιοθήκες του συστήματος που είναι γραμμένες σε γλώσσα προγραμματισμού C++ και περιλαμβάνουν το OpenGL, την SQLite, την Media library κ.α. Οι εφαρμογές που τρέχουν στο κινητό μπορούν να έχουν πρόσβαση σε αυτές τις βιβλιοθήκες μέσω της Dalvik JVM (Java Virtual Machine). Οι εφαρμογές Android έχουν γραφτεί σε

γλώσσα Java και για να μπορέσουν να τρέξουν χρειάζονται το αντίστοιχο περιβάλλον. Για να εκτελέσουμε λοιπόν μία εφαρμογή σε έναν υπολογιστή, θα πρέπει να έχει εγκατασταθεί το κατάλληλο JRE (Java Runtime Environment), για τις εφαρμογές Android. Το ρόλο του JRE παίζει η Dalvik VM (Virtual Machine) (Android - Digital Academy, n.d.).

1.6 Ιστορία εκδόσεων του Android

Σε αυτό το σημείο θα αναφερθούμε σε όλες τις εκδόσεις που έχει δημιουργήσει η Google, από την πρώτη που ήταν το 2007 μέχρι και την νεότερη, δηλαδή το 2019.

1.6.1 Android Alpha – Beta

Alpha

Υπήρχαν δύο εσωτερικές κυκλοφορίες στη Google και την OHA (Open Handset Alliance) πριν η Beta κυκλοφορήσει τον Νοέμβριο του 2007. Οι ονομασίες των εσωτερικών εκδόσεων ήταν "Astro Boy", "Bender” και "R2-D2". Ο Dan Morrill δημιούργησε μερικά πρώτα λογότυπα μασκότ, αλλά το τρέχον πράσινο λογότυπο του Android σχεδιάστηκε από την Irina Blok. Ο διαχειριστής του έργου, ο οποίος σχεδίασε το καθεστώς των ονομασιών γλυκισμάτων που χρησιμοποιείται για την πλειονότητα των δημόσιων κυκλοφοριών του λειτουργικού, ξεκινώντας με το Android 1.5 (Ιστορία εκδόσεων του Android, 2018).

Beta

Η Beta κυκλοφόρησε στις 5 Νοεμβρίου του 2007, ενώ το σετ ανάπτυξης λογισμικού κυκλοφόρησε σχεδόν μετά από ένα μήνα, στις 12 Νοεμβρίου 2007. Η ημερομηνία της 5 Νοέμβρη είναι ευρέως γνωστή ως τα **"γενέθλια"** του Android. Η έκδοση της beta του SDK (Software Development Kit) κυκλοφόρησε με την ακόλουθη σειρά:

- 16 Νοεμβρίου 2007: m3-rc22a
- 14 Δεκέμβρη 2007: m3-rc37a

- 13 Φλεβάρη 2008: M5-Rc14
- 3 Μαρτίου 2008: M5-Rc15
- 18 Αυγούστου 2008: 0.9
- 23 Σεπτεμβρίου 2008: 1,0-R1 (Ιστορία εκδόσεων του Android, 2018).

1.6.2 Android 1.0 (Apple pie) - Android 1.1 (Banana bread)

Οι εκδόσεις 1.0 και 1.1 δεν κυκλοφόρησαν με συγκεκριμένα ονόματα, αν και το Android 1.1 ανεπίσημα ήταν γνωστό ως «Petit Four». Τα ονόματά τους έχουν σαν θέμα διάφορα γλυκίσματα και με αλφαβητική σειρά απ' το Android 1.5 Cupcake του 2009 και ύστερα (Google, 2018).

Εικόνα 1.6.2 – Οι δύο πρώτες εκδόσεις του Android

1.6.3 Android 1.5 (Cupcake)

Το Android 1.5 "**Cupcake**" ήταν η τρίτη έκδοση του Android που αναπτύχθηκε από την Google. Περιλαμβάνει νέα χαρακτηριστικά για τους χρήστες και τους προγραμματιστές της τότε εποχής και αλλαγές στο Android API (Application Program Interface). Για τους προγραμματιστές, η πλατφόρμα του Android 1.5 είναι διαθέσιμη με δυνατότητα λήψης για το Android SDK. Περιλάμβανε νέα χαρακτηριστικά για εκείνη την εποχή. Υποστήριζε νέες λειτουργίες για την κάμερα, δηλαδή κάμερα 5MP (megapixel) με αυτόματη εστίαση, καταγραφή και παρακολούθηση βίντεο από την λειτουργία της κάμερας, αλλά και άμεση μεταφόρτωση του βίντεο και των φωτογραφιών στο Youtube και το Picasa αντίστοιχα, απευθείας από το τηλέφωνο. Νέο εικονικό πληκτρολόγιο με πρόβλεψη λέξεων, πληκτρολόγιο στην οθόνη, υποστήριξη Bluetooth και βελτιωμένα κάποια υπάρχοντα χαρακτηριστικά όπως: αλλαγές στο γραφικό περιβάλλον για τη διαχείριση των εφαρμογών (Android Cupcake, 17).

Εικόνα 1.6.3 – Οι δύο πρώτες εκδόσεις του Android

1.6.4 Android 1.6 (Donut)

Το Android 1.6 “**Donut”** ήταν η δεύτερη μεγάλη αλλαγή της Google στο χώρο του Google OS. Κυκλοφόρησε στις 15 Σεπτεμβρίου 2009 και έφερε μεγάλες αλλαγές όπως: η αναζήτηση φωνής και κειμένου βελτιώθηκε περιλαμβάνοντας ιστορικό σελιδοδεικτών και επαφές. Η gallery, η κάμερα και η βιντεοκάμερα ενσωματώθηκαν πλήρως με γρηγορότερη πρόσβαση στην κάμερα και ήρθε η δυνατότητα πολλαπλής διαγραφής αρχείων. Τέλος, υποστηρίζεται και ανάλυση οθόνης WVGA (Wide Video Graphics Array) (Headlines, 2018).

Εικόνα 1.6.4 – Η έκδοση Donut

1.6.5 Android 2.0 – 2.1 (Eclair)

Το Android 2.0 “**Éclair”** κυκλοφόρησε τον Νοέμβριο του 2009 φέρνοντας τις πιο βασικές οπτικές και αρχιτεκτονικές ανανεώσεις που είδε ποτέ το Android από τότε που ξεκίνησε. Δηλαδή, είχαμε πλέον: πολλαπλή υποστήριξη λογαριασμών Google, πλοήγηση χαρτών Google, μαλακές βελτιώσεις στο πληκτρολόγιο, αναβαθμισμένο πρόγραμμα περιήγησης, ζωντανές ταπετσαρίες, ομιλία σε κείμενο και νέα οθόνη κλειδώματος (Media, 2018).

Εικόνα 1.6.5 - Η έκδοση Éclair

1.6.6 Android 2.2 – 2.2.3 (Froyo)

Το Android 2.2 “**Froyo”** κυκλοφόρησε στα μέσα του 2010 και το πλεονέκτημα του Nexus ήταν σαφές πλέον. Το Nexus One ήταν το πρώτο που ενημερώθηκε. Μερικές αλλαγές που έφερε ήταν οι εξής:

- Δυνατότητα φωνητικών κλήσεων μέσω Bluetooth.
- Βελτίωση φωνητικής αναζήτησης (αναγνώριζε πιο μεγάλες και πολύπλοκες προτάσεις).
- Επαναφορά των αρχικών ρυθμίσεων της συσκευής.
- Κλείδωμα συσκευής με αριθμητικούς ή αλφαριθμητικούς συνδυασμούς.
- Αναβαθμισμένο μενού στη φωτογραφική μηχανή.
- Ενεργοποίηση του φλας στην καταγραφή των βίντεο (Google: Ανακοίνωσε το Android 2.2 (froyo) - myphone.gr, 2018).

Εικόνα 1.6.6 - Η έκδοση Froyo

1.6.7 Android 2.3 – 2.3.7 (Gingerbread)

Το καλοκαίρι του 2010 βγήκε στην αγορά το Android 2.3 “**Gingerbread”** έχοντας γρηγορότερη εισαγωγή κειμένου, βελτιωμένο εργαλείο αντιγραφής-επικόλλησης και επικοινωνία μέσω διαδικτύου (What You Need To Know About Android 2.3 Gingerbread | WIRED, 2018).

Εικόνα 1.6.7 - Η έκδοση Gingerbread

1.6.8 Android 3.0 – 3.2.6 (Honeycomb)

Στις 2 Φεβρουαρίου 2011 παρουσιάστηκε επίσημα το Android 3.0 **“Honeycomb”** ανεβάζοντας πολύ τον ανταγωνισμό του απέναντι στο iOS. Οι αλλαγές του περιλαμβάνουν 3D (3 Dimensions) γραφικά, multitasking (πολυδιεργασία), video chat (βιντεοσυζήτηση), αναβαθμισμένα animations (κινούμενα σχέδια) και νέο περιβάλλον εργασίας. Επιπλέον ο χρήστης μπορεί να διαχειρίζεται διάφορες εφαρμογές που έχει με τη χρήση διαφόρων widgets (πολυεφαρμογές) και να μεταφέρει τα αρχεία ή τους φακέλους του με drag & drop (μεταφορά και απόθεση) (Techgear.gr, 2011).

Εικόνα 1.6.8 - Η έκδοση Honeycomb

1.6.9 Android 4.0 – 4.0.4 (Ice cream sandwich)

Το Android 4.0 “**Ice Cream Sandwich”** έφτασε πρώτα στο Galaxy Nexus και παρουσιάστηκε απο την Google στις 19 Οκτωβρίου 2011 φέρνοντας σημαντικές αλλαγές στο Android OS. Μπορεί να είναι παρόμοιο με το **“Honeycomb”**, αλλά είναι πιο φιλικό ως προς το χρήστη. Τα κανονικά κουμπιά αποτελούν παρελθόν πλέον, εφόσον αντικαταστάθηκαν από εικονικά. Οι ICS (Ice Cream Sandwich) εφαρμογές μπορούν να απεγκατασταθούν απευθείας απ’ το GUI (Graphical User Interface). Οι ειδοποιήσεις είναι προσβάσιμες κατευθείαν από την οθόνη κλειδώματος και οι χρήστες μπορούν να τις διαγράψουν με μία μόνο κίνηση. Βελτίωσε επίσης, το πληκτρολόγιο, την γκαλερί και τις επαφές (Τι νέο υπάρχει στο Android Ice Cream Sandwich 4.0, 2018).

Εικόνα 1.6.9 - Η έκδοση Ice Cream Sandwich

1.6.10 Android 4.1 – 4.3.1 (Jelly bean)

Το Android 4.1 “**Jelly bean”** παρουσιάστηκε για πρώτη φορά στις 27 Ιουνίου 2012, είχε τρεις διαφορετικές εκδόσεις (4.1, 4.2, 4.3.1) και μπορούσε να χρησιμοποιηθεί από smartphone και tablet. Περιλάμβανε μία έξυπνη υποστήριξη για Bluetooth, αυτόματη συμπλήρωση πληκτρολογίου και υποστήριξη δύο ακόμη γλωσσών (αραβικά, εβραϊκά) (Astrium, 2014).

Εικόνα 1.6.10 - Η έκδοση Jelly Bean

1.6.11 Android 4.4 – 4.4.4 (KitKat)

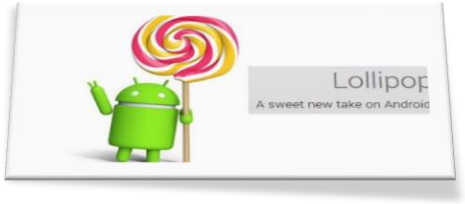
Η Google κυκλοφόρησε το Android 4.4 «**KitKat**» τον Οκτώβριο του 2013 όπου συνέπιπτε με την κυκλοφορία του smartphone Nexus 5. Ήταν απλούστερο απ' την προηγούμενη έκδοση, χωρίς ιδιαίτερες αλλαγές. Σχεδιάστηκε με σκοπό οι εφαρμογές να χρησιμοποιούν όσο γίνεται λιγότερο τη μνήμη RAM (Randrom Access Memory), ενώ βελτιστοποιήθηκε για την καλύτερη διαχείριση της μνήμης όσο αναφορά τα multitasking. Ένα βασικό, καινούριο χαρακτηριστικό είναι πως όσα κινητά διαθέτουν υπέρυθρες μπορούν να χρησιμοποιηθούν ως χειριστήρια τηλεοράσεων (Android 4.4 KitKat: Τι καινούριο φέρνει η νέα έκδοση , 2013).

Εικόνα 1.6.11 - Η έκδοση KitKat

1.6.12 Android 5.0 – 5.1.1 (Lollipop)

Η έκδοση Android 5.0 **"Lollipop"** κυκλοφόρησε στις 3 Νοέμβρη 2014 φέρνοντας αλλαγές τόσο στο σύστημα όσο και στο API. Τα νέα του

χαρακτηριστικά είναι η προηγμένη συλλογή, η βελτιωμένη συλλογή των απορριμάτων και η υποστήριξη εντοπισμού των σφαλμάτων (TechNorms, 2018).



Εικόνα 1.6.12 - Η έκδοση Lollipop

1.6.13 Android 6.0 - 6.0.1 (Marshmallow)

Η έκδοση Android 6.0 **“Marshmallow”** είναι η έκτη και κύρια έκδοση, η οποία κυκλοφόρησε στις 5 Οκτωβρίου του 2015. Εισήγαγε λοιπόν, μία νέα αρχιτεκτονική στα δικαιώματα των εφαρμογών, νέο σύστημα για τη διαχείριση της ενέργειας που μειώνει τη δραστηριότητα παρασκηνίου όταν η συσκευή μας δεν χρησιμοποιείται. Υποστήριξη δακτυλικού αποτυπώματος, usb type C, αλλά και δυνατότητα μεταφοράς δεδομένων και εφαρμογών σε micro sd card (Android Marshmallow, 2017).

Εικόνα 1.6.13 - Η έκδοση Marshmallow

1.6.14 Android 7.0 – 7.1 (Nougat)

Η έκδοση Android 7.0 **“Nougat”** κυκλοφόρησε επίσημα στις 22 Αυγούστου 2016 και έκανε τους χρήστες να θέλουν να το κατεβάσουν δίχως δεύτερη σκέψη. Οι πιο πολλές αλλαγές έμοιαζαν άχρηστες, έως ότου ήρθε η ώρα να τις χρησιμοποιήσουμε. Οι πιο βασικές του αλλαγές ήταν:

- Κλείσιμο όλων των application (εφαρμογών) με τη μία.

- Απάντηση σε μηνύματα κατευθείαν απ' τις ενημερώσεις.
- Περιορισμός στη χρήση των δεδομένων από το πρόγραμμα κινητής τηλεφωνίας για εφαρμογές που τρέχουν στο παρασκήνιο περιμένοντας ενημέρωση.
- Κρατώντας παρατεταμένα το δάχτυλό μας σε κάποια ενημέρωση μπορούμε να σταματήσουμε να λαμβάνουμε ειδοποιήσεις ή να είναι σιωπηλές.
- Οι «Ρυθμίσεις» απλοποιούνται για τη διευκόλυνση του χρήστη.
- Μπορούμε να έχουμε ανοιχτές στην οθόνη μας δύο εφαρμογές. Ταυτόχρονα!
- Οι «Γρήγορες ρυθμίσεις» στο επάνω μέρος είναι αλλαγμένες και μάλιστα με λίγο διαφορετικά εικονίδια και χρώματα.
- Το πληκτρολόγιο της Google έχει πλέον GIF (Graphic Interchange Format).
- Υπάρχει η επιλογή «Night light» (νυχτερινό φως) για τους σκοτεινούς χώρους.
- Αλλάζει το μέγεθος οποιουδήποτε εικονιδίου στην οθόνη.
- Μπορούμε να έχουμε διαφορετικές εικόνες όταν κλειδώνουμε ή ξεκλειδώνουμε το κινητό μας (Τι αλλάζει στο smartphone σας με το Android 7.0 Nougat | in.gr, n.d.).

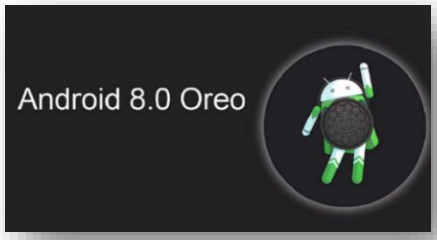
Εικόνα 1.6.14 - Η έκδοση Nougat

1.6.15 Android 8.0 – 8.1 (Oreo)

Η έκδοση Android 8.0 **"Oreo"** κυκλοφόρησε στις 21 Αυγούστου 2017. Έκανε αλλαγές κυρίως σε απόδοση, ασφάλεια, ταχύτητα και ευχρηστία. Οι σημαντικότερες από αυτές είναι οι παρακάτω:

- Picture in Picture (Εικόνα στην εικόνα)

- Autofill API (Αυτόματη συμπλήρωση)
- Κουκίδες ειδοποιήσεων
- Νέο design (σχεδιασμό) στο μενού των ρυθμίσεων και μπαταρίας
- Snooze (αναβολή) στις ειδοποιήσεις
- Κανάλια ειδοποιήσεων
- Background limits (Όρια φόντου)
- Έξυπνη επιλογή κειμένου
- Καλύτερο Bluetooth
- Αυτόματο Wi-fi
- Πιο γρήγορο Boot (κλείσιμο)
- Καλύτερο performance (προετοιμασία) (Happy Byte Blog, n.d.).



Εικόνα 1.6.15 - Η έκδοση Oreo

1.6.16 Android 9.0 (Pie)

Η έκδοση του Android 9.0 **"Pie"** κυκλοφόρησε τον Αύγουστο του 2018, ενώ ανακοινώθηκε για πρώτη φορά τον Μάρτιο του ίδιου έτους. Είναι διαθέσιμη για τις συσκευές Google Pixel και Essential Phone με τα παρακάτω χαρακτηριστικά:

- Χειροκίνητη επιλογή θέματος
- Το κουμπί Περιστροφικού Κλειδώματος δείχνει αν η συσκευή χρησιμοποιεί αυτή τη λειτουργία
- Νέο εικονίδιο πίσω κουμπιού
- Βελτιωμένο χαρακτηριστικό προσαρμοστικής φωτεινότητας
- Προσαρμοστικό χαρακτηριστικό γνώρισμα μπαταριών
- Τοποθετώντας στο κινητό το πρόσωπό μας προς τα κάτω, οι ειδοποιήσεις τίθενται σε σίγαση

- Νέο χαρακτηριστικό «Ψηφιακή Ευημερία»
- Βελτιωμένος ο οριζόντιος πολυεργασιακός εναλλαγέας εφαρμογών
- Νέο σύστημα βασισμένο στις χειρονομίες
- Υποστήριξη HEIF (High Efficiency Image File Format)
- DNS μέσω TLS (Domain Name System), (Transport Layer Security)
- Ενεργοποίηση του Bluetooth κατά την οδήγηση
- Το ποσοστό της μπαταρίας εμφανίζεται και στο “Always on Display” (πάντα στην οθόνη)
- Επανασχεδιασμένη ρύθμιση έντασης
- Υποστήριξη για τις διακοπές προβολή
- Πιο πλούσιες ειδοποιήσεις μηνυμάτων
- Νέες μεταβάσεις για την εναλλαγή μεταξύ εφαρμογών ή δραστηριοτήτων μεταξύ των εφαρμογών
- Στρογγυλεμένες γωνίες
- Νέα λειτουργία κλειδώματος
- Νέο κουμπί ονόματι «Στιγμιότυπο»
- Η εξοικονόμηση μπαταρίας έχει πλέον πορτοκαλί επικάλυψη στην ειδοποίηση και την κατάσταση της μπάρας
- Το ρολόι μετακινήθηκε αριστερά απ’ τη γραμμή των ειδοποιήσεων
- Νέα διεπαφή χρήστη αριστερά απ’ τη γραμμή των γρήγορων ρυθμίσεων (Media, 2018).

Εικόνα 1.6.16 - Η έκδοση Pie

ΚΕΦΑΛΑΙΟ 2

ΠΡΟΓΡΑΜΜΑ ANDROID STUDIO

2.1 Εισαγωγή

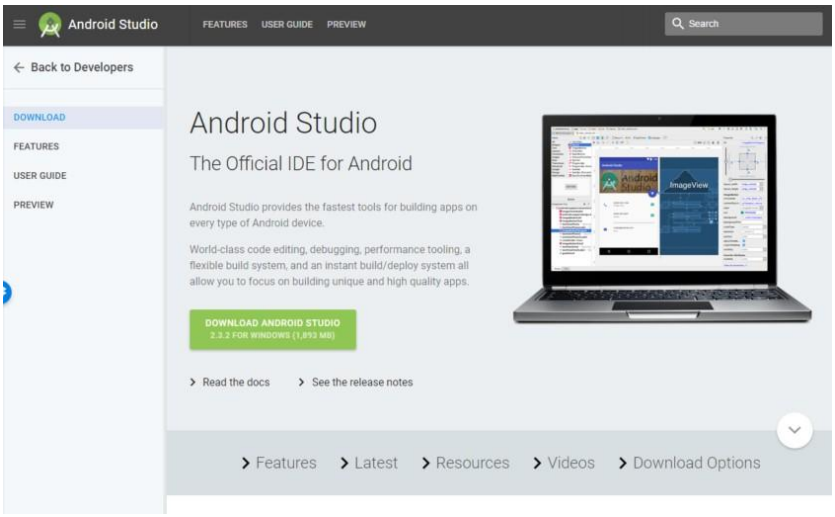
Το κεφάλαιο αυτό θα μας απασχολήσει με το τι είναι το πρόγραμμα Android Studio, πώς το εγκαθιστούμε, με ποιον τρόπο μπορούμε να φτιάξουμε μια νέα εφαρμογή, αλλά και με βασικά στοιχεία που πρέπει να γνωρίζουμε.

2.2 Τι είναι το Android Studio

Το **Android Studio** είναι ένα ολοκληρωμένο προγραμματιστικό περιβάλλον, με το οποίο αναπτύσσονται εφαρμογές αποκλειστικά σε λειτουργικό Android. Ξεκίνησε με την έκδοση 0.1 τον Μάιο του 2013, ύστερα έγινε δοκιμή με την 0.8, ενώ η πρώτη σταθερή έκδοση (1.0) κυκλοφόρησε τον Δεκέμβριο του 2014. Ωστόσο αντικατέστησε τα Android Development Tools (Εργαλεία Ανάπτυξης Android), όντας το κύριο ολοκληρωμένο προγραμματιστικό περιβάλλον για την ανάπτυξη εφαρμογών Android. Τέλος, είναι διαθέσιμο για: Windows, Linux και Mac OS X (Android Studio, 2017).

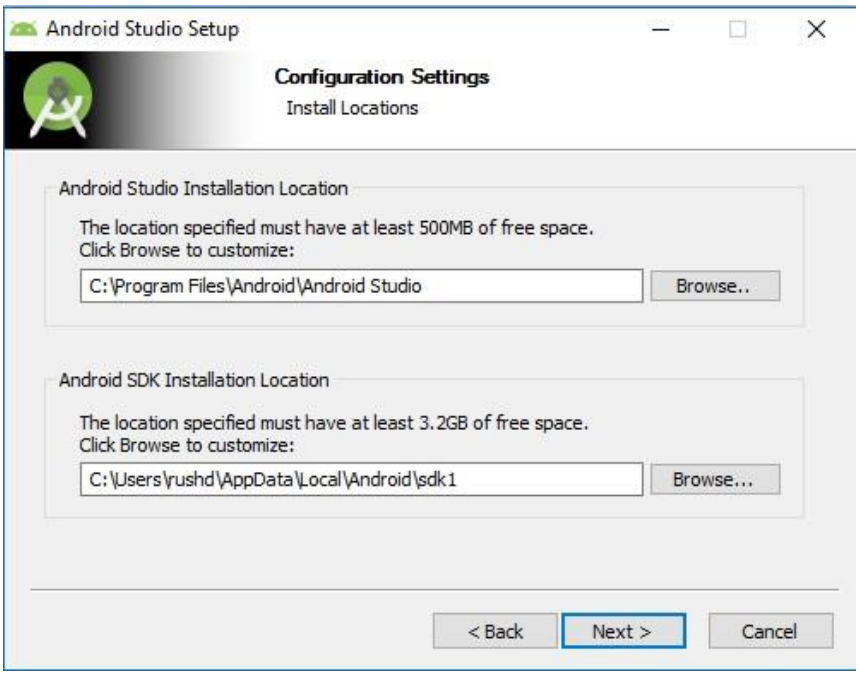
2.3 Εγκατάσταση του προγράμματος

Η εγκατάσταση του Android Studio είναι πολύ απλή και εύκολη χάρη σε σχεδόν όλα τα πακέτα ενός προγράμματος εγκατάστασης. Κατεβάζοντας το Android Studio θα έχουμε και το Android SDK, τον διαχειριστή SDK κ.ά. Το μόνο που θα χρειαστούμε είναι το Java Development Kit. Το Android Studio και το SDK είναι αρκετά μεγάλα, οπότε θα πρέπει να βεβαιωθούμε πως έχουμε αρκετό χώρο στο δίσκο μας πριν ξεκινήσουμε (Android Authority, 2017).



Εικόνα 2.3.1 – Η λήψη του προγράμματος Android Studio

Αν ακολουθήσουμε τις απλές αυτές οδηγίες κατά την εγκατάσταση θα πρέπει στη συνέχεια να δημιουργήσουμε μία πλατφόρμα Android. Μόλις βεβαιωθούμε ότι έχουμε επιλέξει το checkbox που λέει στον installer (εγκαταστάτη) ότι θέλουμε και το Android SDK, διαλέγουμε που θέλουμε να εγκατασταθούν τα αρχεία μας (Android Studio, 2017).

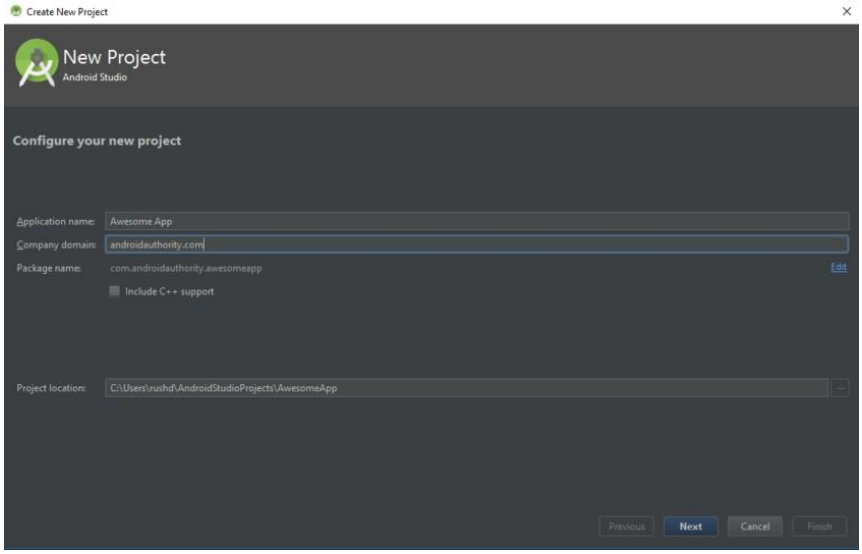


Εικόνα 2.3.2 – Επιλογή διαδρομής αποθήκευσης του προγράμματος

Διαλέγουμε έναν φάκελο για το SDK βάζοντας όποιο όνομα θέλουμε, αρκεί να μην περιλαμβάνει κενούς χαρακτήρες. Στο παράδειγμά μας, για την εγκατάσταση του Android Studio επιλέχθηκε ένας κρυφός φάκελος. Για να βρούμε αυτόν τον φάκελο θα πρέπει να επιλέξουμε το «Εμφάνιση Κρυφών Φακέλων» (Android Authority, 2017).

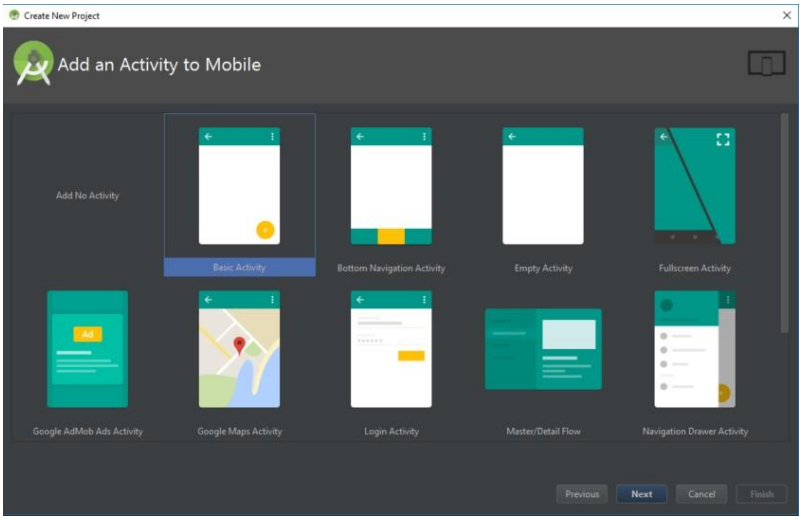
2.4 Ξεκινώντας ένα νέο project

Εφόσον ανοίξει το Android Studio θα πάμε να δημιουργήσουμε ένα νέο Project (Εργασία/Έργο), επιλέγοντας το «Δημιουργία Νέου Project», διαφορετικά η διαδρομή για τη δημιουργία νέου Project είναι η εξής: File>New>New Project (Android Studio, 2017).



Εικόνα 2.4.1 – Δημιουργία νέου project

΄Υστερα, θα έχουμε τη δυνατότητα να διαλέξουμε έναν από τους πολλούς διαφορετικούς τύπους των activities (δραστηριότητες). Τα activities είναι το οπτικό κομμάτι μίας εφαρμογής, κάτι σαν μία οθόνη. Σε κάποιες περιπτώσεις, όλη η εφαμοργή ή ένας μέρος της, μπορεί να μεταβαίνει απ΄την μία οθόνη στην άλλη. Αν θέλουμε, μπορούμε να ξεκινήσουμε νέο activity έχοντας επιλέξει το «Add No Activity», όμως συνήθως χρειαζόμαστε κάποιο. Συχνά οι αρχάριοι developers διαλέγουν για ευκολία το «Empty Activity» (Android Studio, 2017).



Εικόνα 2.4.2 – Activities

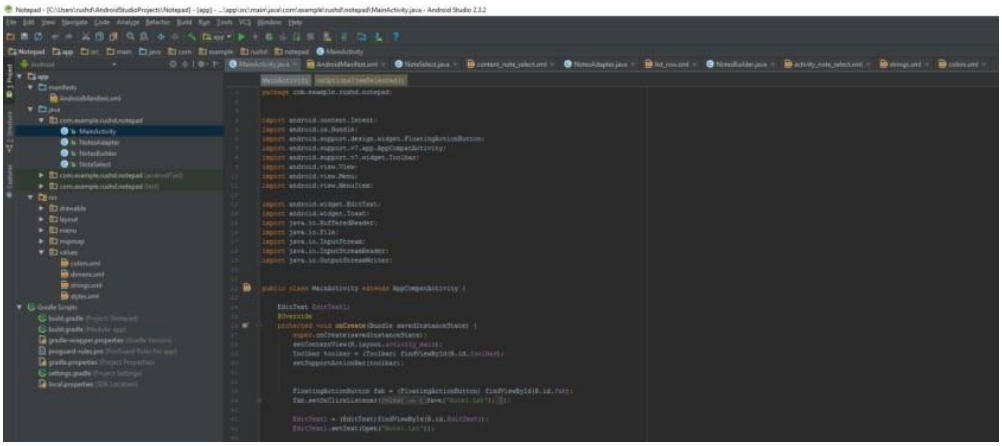
Συνήθως διαλέγουμε το «Basic Activity», διότι διαθέτει μία προεπιλεγμένη εμφάνιση της εφαρμογής Android. Περιλαμβάνει ένα μενού πάνω δεξιά στην οθόνη και ένα πλήκτρο FAB (Floating Action Button), το οποίο είναι μία επιλογή σχεδίασης που προσπαθεί να ενθαρρύνει η Google. Το «Empty Activity» είναι ίδιο με το «Basic Activity», με τη μόνη διαφορά ότι δεν υπάρχει το πρόσθετο κουμπί «+» (Android Authority, 2017).

Διαλέγουμε λοιπόν, το Activity που πιστεύουμε ότι ταιριάζει πιο πολύ στην εφαρμογή που θέλουμε να χτίσουμε. Σε αυτό το σημείο πρέπει να σκεφτούμε το όνομα που θέλουμε να της δώσουμε, το ελάχιστο Android SDK που θέλουμε να υποστηρίξουμε και το όνομα του πακέτου μας. Το όνομα του πακέτου είναι το όνομα που θα έχει η εφαρμογή μας όταν ανέβει στο Play Store. Συνήθως είναι το όνομα του προγραμματιστή σε συνδυασμό με το όνομα της εφαμογής (Android Authority, 2017).

2.5 Μερικοί βασικοί φάκελοι

Ο κυρίως κώδικας βρίσκεται σε ένα Java αρχείο που έχει την ίδια ονομασία με το Activity. Ως προεπιλογή, ονομάζεται «MainActivity.Java», όμως εμείς αν θέλουμε μπορούμε να το αλλάξουμε κατά τα πρώτα βήματα δημιουργίας του Project. Είναι, δηλαδή, το σημείο που εισάγουμε το Java script (Java σενάριο) μας και ορίζουμε τη συμπεριφορά που θα έχουν οι εφαρμογές μας (Android Authority, 2017).

Ωστόσο, την πραγματική διάταξη της εφαρμογής μας τη χειρίζεται ένα άλλο κομμάτι κώδικα. Αυτός ο κώδικας βρίσκεται στο φάκελο ονόματι «activity_main.xml». XML (Extensible Markup Language) ονομάζουμε μία γλώσσα που ορίζει τη διάταξη ενός εγγράφου, παρόμοια με την HTML που χρησιμοποιείται για τη δημιουργία ιστοσελίδων. Δεν είναι καθαρός προγραμματισμός, αλλά κυρίως ένα είδος κώδικα (Android Authority, 2017).



Εικόνα 2.5 – Βασικό περιβάλλον προγράμματος

Τώρα, αν θέλουμε να δημιουργήσουμε ένα κουμπί, θα πρέπει να επεξεργαστούμε το «activity_main.xml». Στην περίπτωση όμως, που θέλουμε να περιγράψουμε τι κάνει, θα πρέπει να το βάλουμε στο «MainActivity.Java». Βέβαια, αν θέλουμε να κάνουμε κάποια πιο περίπλοκα πράγματα, μπορούμε να χρησιμοποιήσουμε οποιοδήποτε XML για να ορίσουμε τη διάταξη όποιου Java script επιθυμούμε με τη βοήθεια της γραμμής **setContentView(R.layout.activity_main)**, η οποία βρίσκεται στην κορυφή του κώδικα Java (Android Studio, 2017).

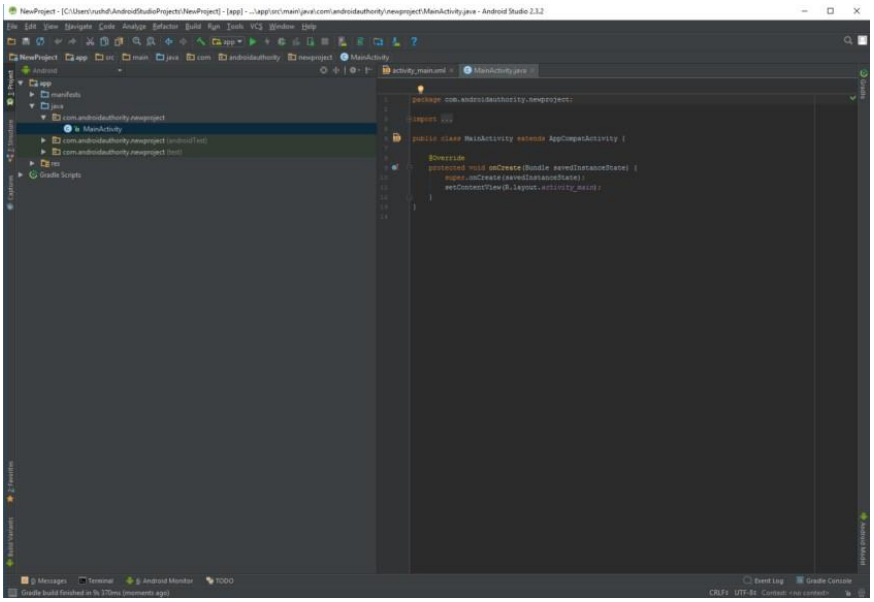
Η παραπάνω γραμμή κώδικα μας λέει πως στο Android Studio το κάθε script μας έχει τη διάταξή του ρυθμισμένη κατά activity_main.xml. Αυτό σημαίνει επίσης, πως θεωρητικά έχουμε τη δυνατότητα να χρησιμοποιήσουμε το ίδιο XML για να ορίσουμε τη διάταξη σε δύο διαφορετικές κλάσεις Java (Android Authority, 2017).

Σε μερικές περιπτώσεις θα έχουμε στην πραγματικότητα περισσότερα από ένα XML αρχεία, τα οποία περιγράφουν διαφορετικές πτυχές από τη διάταξη του activity μας. Για παράδειγμα, αν επιλέξουμε το «Basic Activity», αντί για το «Empty Activity», θα έχουμε ένα «activity_main.xml», το οποίο θα ορίζει τη θέση

του FAB και άλλων UI (User Interface) στοιχείων, αλλά και ένα «content_main.xml», το οποίο θα περιλαμβάνει το περιεχόμενο που θα θέλουμε να προσθέσουμε στη μέση της οθόνης. Ενδεχομένως, εάν θελήσουμε να προσθέσουμε «views» (κουμπιά, lists, text boxes) θα πρέπει να έχουμε λάβει υπόψιν μας πως μπορεί να έχουν τις δικές τους μορφές XML (Android Authority, 2017).

2.6 Φάκελοι και καρτέλες

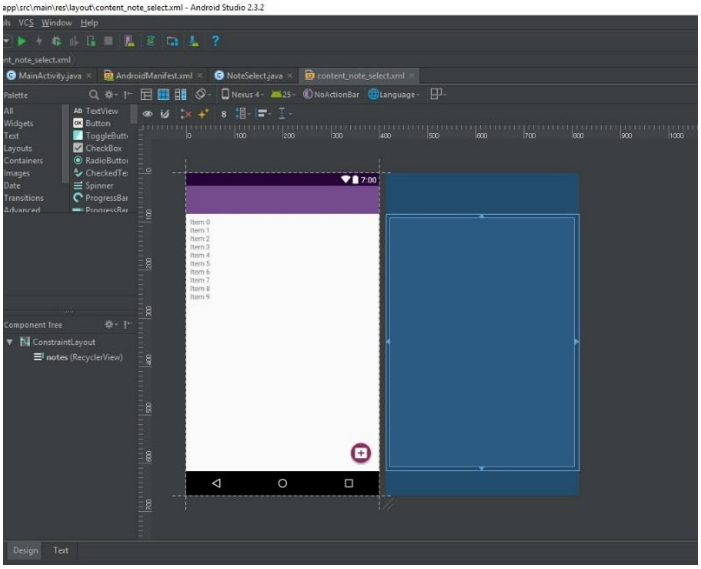
Όπως μπορούμε να δούμε στην παρακάτω φωτογραφία, η Android εφαρμογή αποτελείται από πολλούς φακέλους και είναι καθήκον του προγράμματος να τα διατηρεί σε ένα μέρος για τη διευκόλυνσή μας. Το κύριο παράθυρο στα δεξιά της οθόνης, μας επιτρέπει να προβάλλουμε μεμονωμένα αρχεία και script, ενώ οι καρτέλες κατά μήκος της κορυφής μας επιτρέπουν να τα αλλάζουμε (Android Studio, 2017).



Εικόνα 2.6.1 – Φάκελοι και καρτέλες

Εάν επιθυμούμε να ανοίξουμε μία νέα καρτέλα, μπορούμε να το κάνουμε μέσα απ' την ιεραρχία που έχουν τα αρχεία μας στα αριστερά. Εκεί θα βρούμε όλους τους φακέλους με τους υποφακέλους τους. Όλα τα Java αρχεία βρίσκονται κάτω απ' το όνομα του πακέτου της εφαρμογής μας. Για παράδειγμα, με διπλό κλικ στο

«MainActivity.Java» θα ανοίξει αντίστοιχη καρτέλα στα δεξιά μας (Android Authority, 2017).



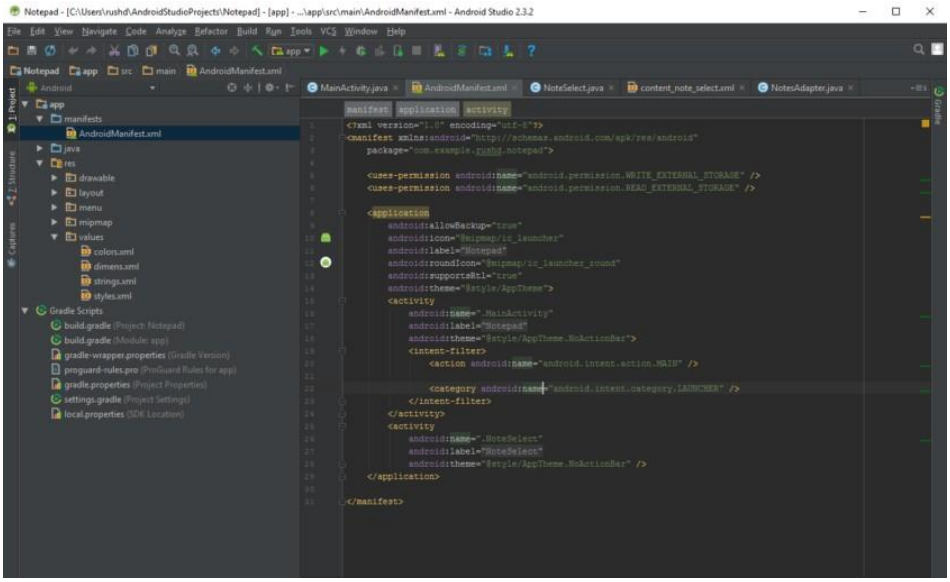
Εικόνα 2.6.2 – Άνοιγμα καρτέλας

Όταν επεξεργαζόμαστε τα αρχεία XML συνήθως παρατηρούμε δύο καρτέλες στο κάτω μέρος της οθόνης που μας επιτρέπουν την εναλλαγή ανάμεσα στο «Text» και το «Design». Στο «Text view» μπορούμε να αλλάξουμε τον XML κώδικα **απευθείας**, προσθέτοντας και επεξεργάζοντας γραμμές. Στο «Design view» (Προβολή σχεδίασης) έχουμε τη δυνατότητα να προσθέσουμε, αφαιρέσουμε ή σύρουμε μεμονωμένα στοιχεία γύρω από την οθόνη για να δούμε πως φαίνονται. Επίσης, το «Text view» (Προβολή κειμένου) διαθέτει ένα παράθυρο προεπισκόπησης για να παρακολουθούμε τι φτιάχνουμε, όσο η οθόνη μας βέβαια είναι αρκετά μεγάλη (Android Studio, 2017).

2.7 Περισσότεροι τύποι φακέλων

Ένας χρήσιμος φάκελος είναι αυτός με το όνομα «res». Είναι συντομογραφία του «resources», δηλαδή «πόροι» στα ελληνικά. Περιλαμβάνει το φάκελο «drawable» (εικόνες που τοποθετούμε στην εφαρμογή μας), «Layout» (εδώ έρχονται τα αρχεία XML) https://developer.android.com/studio/intro/. Οτιδήποτε βρίσκεται μέσα στο «res» πρέπει να είναι γραμμένο με πεζά γράμματα. Γι’ αυτό το λόγο η κάτω παύλα χρησιμοποιείται πολύ έτσι ώστε να διαχωρίζονται τα

ονόματα των αρχείων σε αναγνώσιμους χαρακτήρες. Επίσης, χρήσιμος είναι ο φάκελος «values». Περιέχει περισσότερα αρχεία XML, τα οποία κρατάνε τις τιμές των μεταβλητών, όπως είναι τα ονόματα, τα χρώματα κλπ (Meet Android Studio, 2018).



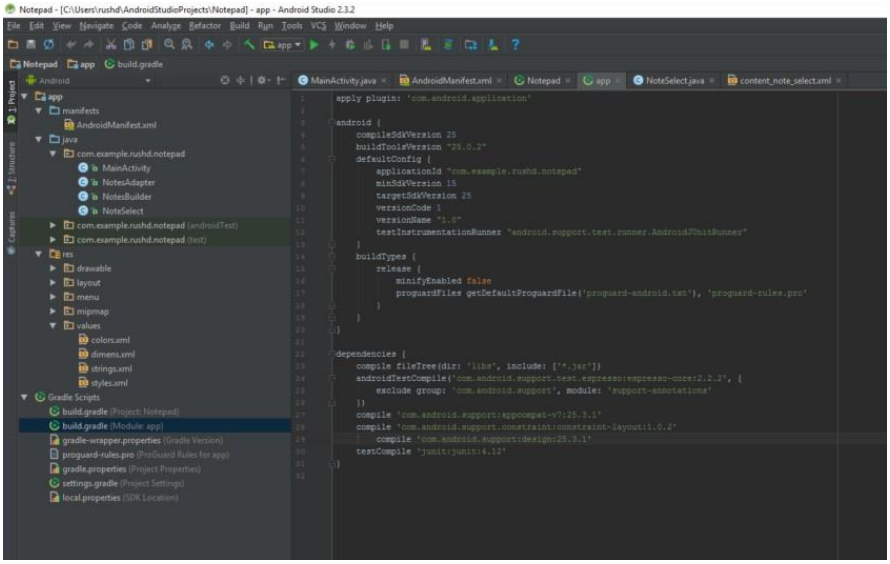
Εικόνα 2.7 – Χρήσιμοι φάκελοι

Ένα πολύ χρήσιμο αρχείο είναι το «AndroidManifest.xml» όπου βρίσκεται μέσα στο φάκελο «manifests». Η δουλειά του είναι να καθορίζει τα κρίσιμα στοιχεία της εφαρμογής μας όπως τα activities που θα συμπεριληφθούν, το όνομα της εφαρμογής έτσι όπως θα το βλέπουν οι χρήστες, τα δικαιώματα της εφαρμογής κ.ά (Meet Android Studio, 2018).

Μπορούμε να δημιουργήσουμε πρόσθετες Java κλάσεις, αρχεία XML ή ολόκληρα activities σε οποιοδήποτε σημείο θέλουμε με σκοπό να προσθέσουμε περισσότερες λειτουργίες στην εφαρμογή μας. Κάνοντας δεξί κλικ στο φάκελο που επιθυμούμε και πατώντας την επιλογή «New» μπορούμε να προσθέσουμε οτιδήποτε θέλουμε. Μπορούμε να ανοίξουμε ακόμα και τον κατάλογο της εφαρμογής μας κάνοντας δεξί κλικ πάνω του και επιλέγοντας το “Show in Explorer” (Android Authority, 2017).

2.8 Γνωρίζοντας τα Gradle αρχεία

Το Android Studio προσπαθεί να κρατάει όμορφα και απλά τα πράγματα για τους χρήστες του, παρέχοντας όλα τα απαραίτητα εργαλεία και χαρακτηριστικά σε ένα μέρος. Τα πράγματα ίσως να γίνουν λίγο περίπλοκα όταν χρειαστεί να πειράξουμε κάποιο απ' τα στοιχεία (Meet Android Studio, 2018).



Εικόνα 2.8 – Αρχεία Gradle

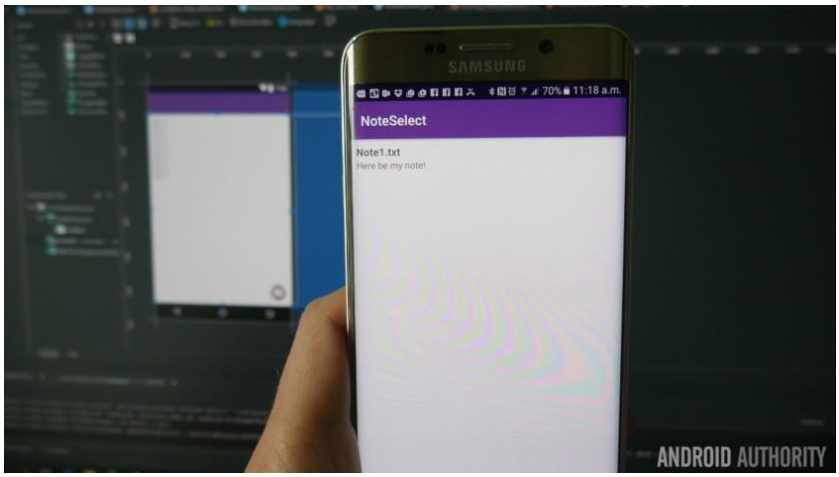
Για παράδειγμα, αν παρατηρήσετε, το Android Studio αναφέρεται περιστασιακά στο Gradle. Στην ουσία είναι ένα εργαλείο δημιουργίας αυτοματισμού, το οποίο βοηθά το πρόγραμμα να μετατρέψει όλα αυτά τα διαφορετικά αρχεία σε ένα APK (Android Package). Καλό είναι να αφήσουμε το Gradle να κάνει την περισσότερη δουλειά στο πρόγραμμα, αλλά περιστασιακά θα πρέπει να μεταβαίνουμε στα «build.gradle» αρχεία αν θέλουμε να προσθέσουμε προηγμένες λειτουργίες στην εφαρμογή μας. Κάποιες φορές, αν κάτι πάει στραβά και δεν δουλέψει η εφαρμογή μας όπως πρέπει, καλό είναι να πάμε στο «Build» και να διαλέξουμε το «Clean Project» για να ξαναβρεί το κάθετι το ρόλο του. Τέλος, συνήθως υπάρχουν δύο αρχεία Gradle build, ένα για ολόκληρο το project κι ένα άλλο για το «module» (η εφαρμογή μας) (Meet Android Studio, 2018).

2.9 Debugging, εικονικές συσκευές, διαχειριστής SDK

Όταν είμαστε έτοιμοι να ελέγξουμε την εφαρμογή μας έχουμε δύο επιλογές. Η μία επιλογή είναι να τη δοκιμάσουμε στη φυσική μας κινητή συσκευή και η άλλη είναι σε μία εικονική κινητή συσκευή (emulator) (Debugging in Android Studio, 2016).

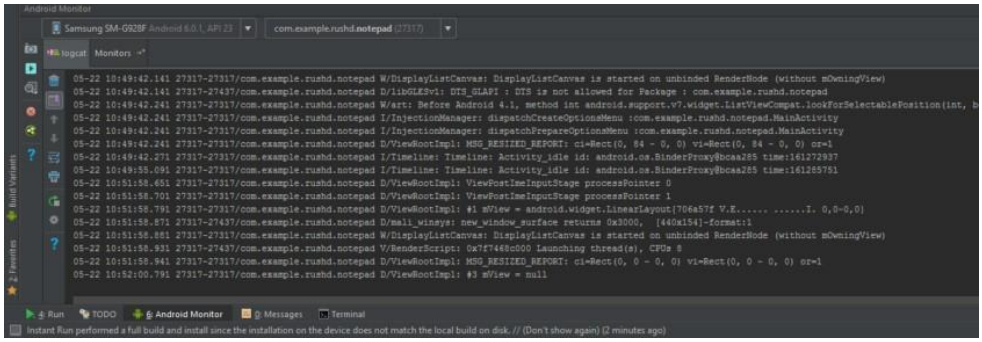
Το να τρέξουμε την εφαρμογή στο κινητό μας είναι πολύ εύκολο. Το συνδέουμε με ένα usb (Universal Serial Bus), ενεργοποιούμε την επιλογή της εγκατάστασης από άγνωστες πηγές και μετά εκτελούμε την εφαμογή μας (Debugging in Android Studio, 2016).

Μετά θα δούμε ένα μήνυμα που θα μας λέει ότι το Gradle εκτελείται. Είναι πολύ γρήγορος τρόπος χάρη στη λειτουργία Instant Run (Άμεση Εκτέλεση) (Debugging in Android Studio, 2016).



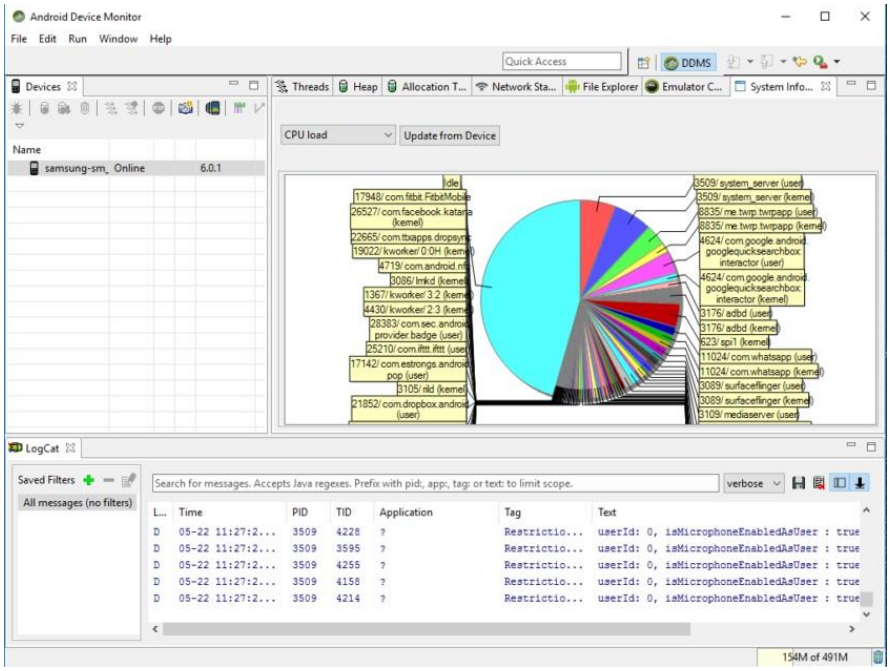
Εικόνα 2.9.1 – Πρώτο στάδιο δοκιμής της εφαρμογής

Όσο τρέχει η εφαρμογή μας μπορεί να λαμβάνουμε διάφορες αναφορές μέσω της καρτέλας «logcat» στο κάτω μέρος της οθόνης. Αν κάτι δεν πάει καλά με την εφαρμογή μας σημαίνει ότι κάπου υπάρχουν λάθη και σταματάει. Τότε εμφανίζεται ένα κόκκικο κείμενο, το οποίο μας περιγράφει το λάθος μας. Αυτό μας εξοικονομεί πολύ χρόνο, διότι μας δείχνει το λάθος μας σε σύγκριση με το να ψάχνουμε μόνοι μας να το βρούμε (Debugging in Android Studio, 2016).



Εικόνα 2.9.2 – Κονσόλα σφαλμάτων

Μπορούμε βέβαια και να μεταβαίνουμε στα monitors (οθόνες) για να βλέπουμε χρήσιμες πληροφορίες, όπως η χρήση της CPU (Control Process Unit) κ.ά. Η οθόνη της εικονικής συσκευής πηγαίνει ένα βήμα παραπέρα και μας επιτρέπει να παρακολουθούμε τα πάντα (Debugging in Android Studio, 2016).

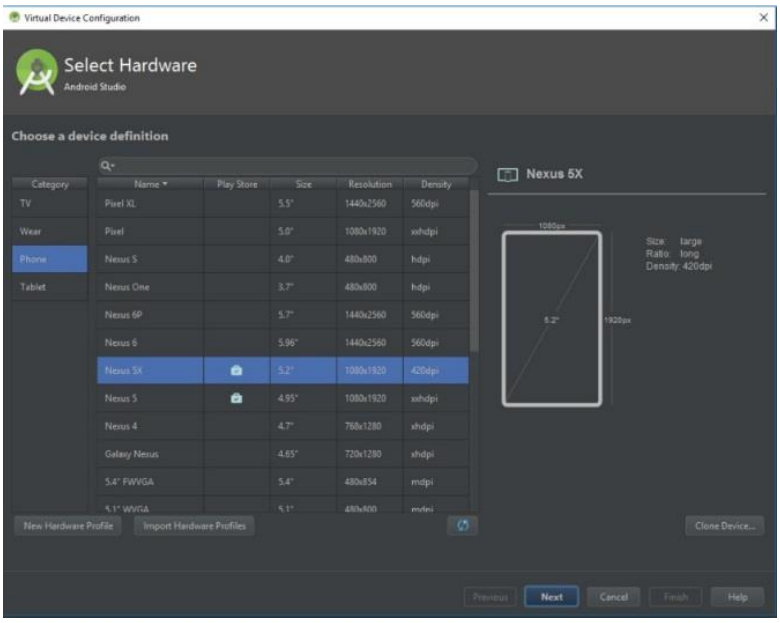


Εικόνα 2.9.3 – Monitor

2.10 Διαχειριστής AVD

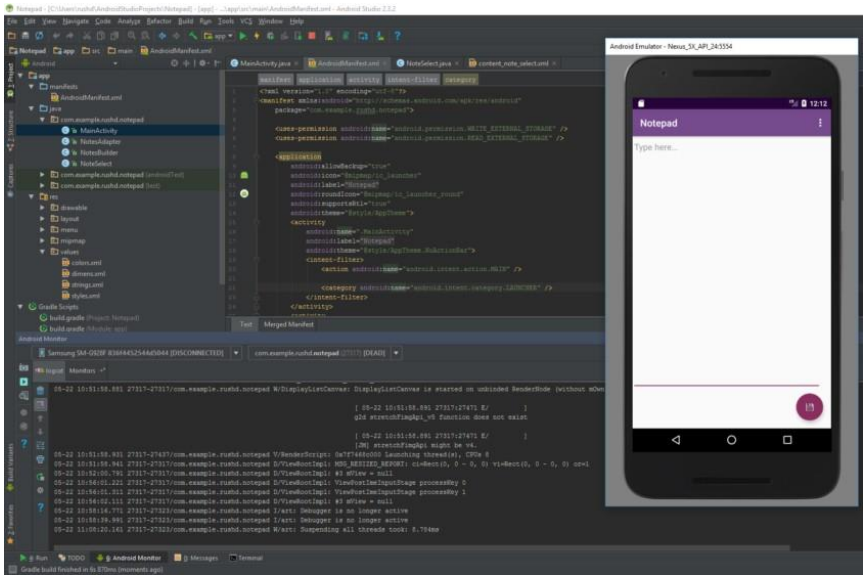
Είναι δύσκολο κάποιος να θέλει να προγραμματίσει σε Android χωρίς να διαθέτει μία συσκευή Android. Ωστόσο μία πολύ μεγάλη πρόκληση είναι ο κατακερματισμός. Διότι δεν είναι καλό να δουλεύει η εφαρμογή μας στη δική μας

συσκευή, αλλά και σε άλλες συσκευές με 10'' και 15''. Πρέπει, λοιπόν, να δουλεύει και σε συσκευές που διαθέτουν παλαιότερες εκδόσεις Android ή είναι υποτιμημένες. Σε αυτό το σημείο έρχεται η εικονική συσκευή. Δηλαδή, μιλάμε για έναν εξομοιωτή που μπορούμε να χρησιμοποιούμε για να μιμηθούμε την εμφάνιση και την απόδοση οποιασδήποτε άλλης συσκευής Android, ρυθμίζοντας πράγματα όπως: μέγεθος οθόνης, έκδοση λογισμικού Android κλπ (AVD Manager, n.d.).



Εικόνα 2.10.1 – Δημιουργία νέου προσομοιωτή

Για να χρησιμοποιήσουμε την εικονική συσκευή, πρώτα πρέπει να έχουμε δημιουργήσει μία, κατεβάζοντας τα απαραίτητα στοιχεία και καθορίζοντας κάποιες ρυθμίσεις. Για να το κάνουμε αυτό ακολουθούμε το εξής μονοπάτι: Tools>Android>AVD Manager. Επιλέγουμε το υλικό και την πλατφόρμα Android που επιθυμούμε. Εάν η έκδοση του Android που θέλουμε δεν έχει κατέβει ακόμα, τότε η επιλογή θα εμφανιστεί δίπλα (AVD Manager, n.d.).



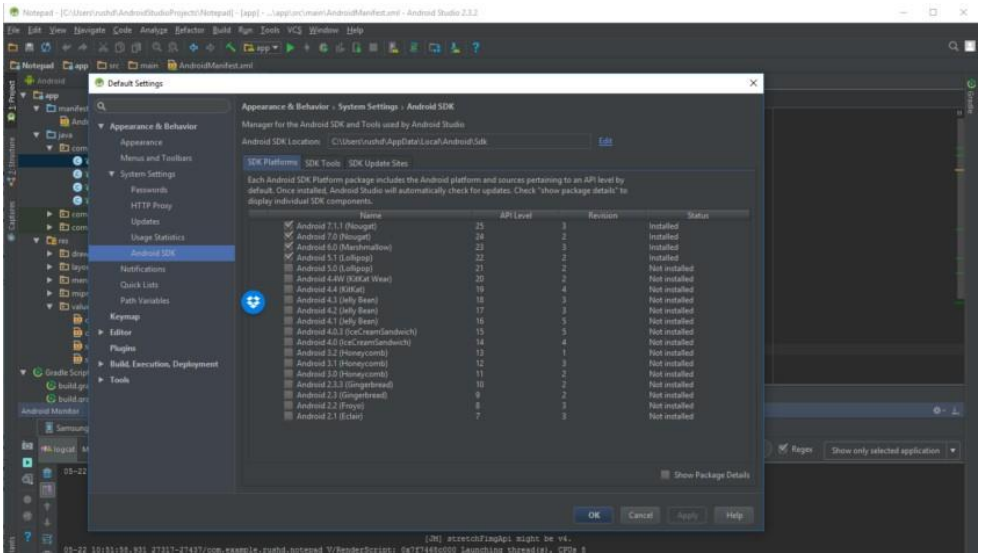
Εικόνα 2.10.2 – Έναρξη του προσομοιωτή

Εφόσον έχουμε ρυθμίσει κάποιες συσκευές προς χρήση, θα μπορούμε να διαλέξουμε μία για να τρέχει την εφαρμογή μας. Να σημειώσουμε σε αυτό το σημείο πως για να εκτελεστεί στην εικονική συσκευή θα χρειαστούμε κάποιες εξειδικευμένες προδιαγραφές (AVD Manager, n.d.).

Εικόνα 2.10.3 – Προσομοιωτής παιχνιδιών

2.11 Διαχειριστής SDK

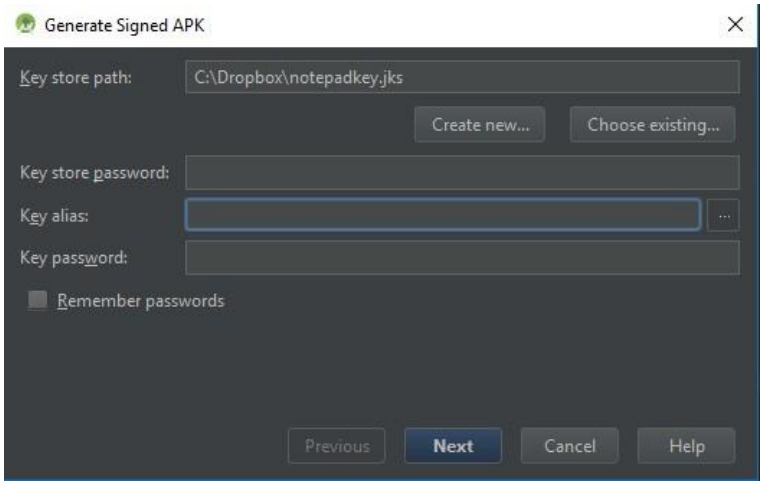
Αν θέλουμε να δημιουργήσουμε μία εικονική συσκευή που εκτελεί συγκεκριμένη έκδοση Android, θα χρειαστεί να κατεβάσουμε απαραίτητα εργαλεία πλατφόρμας και εργαλεία SDK. Μπορούμε να το κάνουμε αυτό μέσω του διαχειριστή SDK, δηλαδή με τη διαδρομή: Tools>SDK Manager (SDK Manager, n.d.).



Εικόνα 2.11 – Διαχειριστής SDK

2.12 Δημιουργία APK's

Μόλις ολοκληρώσουμε τη δοκιμή της εφαρμογής μας και είμαστε έτοιμοι να την κυκλοφορήσουμε, θα πρέπει να επιλέξουμε το μονοπάτι: Build>Generate Signed APK. Απο εκεί θα μας δοθεί το αρχείο που θα χρειαστούμε για να το ανεβάσουμε στο Play Store, το οποίο περιλαμβάνει όλα τα αρχεία, τους πόρους κλπ (Build and share Your APK File Unit, 2018).



Εικόνα 2.12 – Φόρμα στοιχείων για το ανέβασμα της εφαρμογής

Τότε, θα μας ζητηθεί να εισάγουμε ή να δημιουργήσουμε, αν δεν έχουμε, ένα κατάστημα κλειδιού. Αυτό είναι ένα είδος πιστοποιητικού γνησιότητας που αποδεικνύει ότι το APK που ανεβάζουμε είναι η εφαρμογή που λέμε και βέβαια εμποδίζει κάποιον να χακάρει το Google Play λογαριασμό μας με σκοπό να ανεβάσει ένα κακόβουλο αρχείο APK ως «ενημέρωση» της εφαρμογής μας. Θα πρέπει να διατηρούμε ασφαλές αυτό το αρχείο, διότι αν χαθεί δεν υπάρχει περίπτωση να ενημερωθεί ξανά η εφαρμογή. Τέλος, επιλέγουμε σαν τύπο κατασκευής το **«release»** στην περίπτωση που θέλουμε να το απελευθερώσουμε και ύστερα ολοκληρώνουμε τη διαδικασία πατώντας το κουμπί **«finish»** (Build and share Your APK File Unit, 2018).

ΚΕΦΑΛΑΙΟ 3

ΑΝΑΠΤΥΞΗ ΚΩΔΙΚΑ

3.1 Εισαγωγή

Σε αυτό το κεφάλαιο αναλύουμε τον κώδικα που αναπτύξαμε. Αναφερόμαστε πρώτα στα Java αρχεία, μέσα από τα οποία προγραμματίστηκαν βασικές λειτουργίες της εφαρμογής και έπειτα στα xml αρχεία, με τα οποία επεξεργαστήκαμε το οπτικό της κομμάτι.

3.2 MainActivity.java

Η κλάση MainActivity είναι η βασική σελίδα της εφαρμογής μας και θα δούμε αναλυτικά ποιες είναι οι λειτουργίες της.

3.2.1 Δημιουργία FloatingActionButton

Αυτό το κουμπί ασχολείται με τη δημιουργία μίας νέας αφύπνισης κάθε φορά που το πατάμε.

```
actionButton.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v)
    {
        Intent intent_button = new Intent(MainActivity.this, Clock_Activity.class);
        startActivity(intent_button);
    }
});
```

3.2.2 Επιστροφή στην αρχική οθόνη

Μπορούμε να επιστρέψουμε στην αρχική οθόνη της συσκευής μας πατώντας μόνο το **Home Button** (Κουμπί Αφετηρίας). Διαφορετικά, εάν πατήσουμε το κουμπί επιστροφής εμφανίζεται ένα μικρότερο παράθυρο, το οποίο μας δίνει σαφείς πληροφορίες σχετικά με το πως να εξέλθουμε από την εφαρμογή μας.

```
public void useHomeButton()
{
   AlertDialog.Builder builder = new AlertDialog.Builder(this);
   builder.setMessage("Παρακαλώ πατήστε το Home Button για να
εξέλθετε!!!")
         .setTitle("Μήνυμα Εξόδου")
         .setCancelable(false)
         .setIcon(R.drawable.ic_home_black_24dp)
         .setPositiveButton("ΤΟ ΚΑΤΑΛΑΒΑ", new
DialogInterface.OnClickListener()
         {
            public void onClick(DialogInterface dialog, int id)
            {
               //do things
            }
         });
   AlertDialog alert = builder.create();
   alert.show();
}
```

3.2.3 Τι συμβαίνει όταν κάνουμε κλικ σε κάθε listview item;

Κάθε φορά που κάνουμε κλικ σε μία από τις αφυπνίσεις, κατευθείαν μεταφερόμαστε στις ρυθμίσεις της τρέχουσας αφύπνισης (Clock Activity), οι οποίες δείχνουν την ώρα και τον ήχο που είχαμε επιλέξει την τελευταία φορά.

```
listView.setOnItemClickListener(new AdapterView.OnItemClickListener()
{
  @Override
   public void onItemClick(AdapterView<?> parent, View view, int position, long
id)
   {
      if(id == 0)
      {
         Intent intent_listview = new Intent(MainActivity.this,
Clock_Activity.class);
         intent_listview.putExtra("svd_info", saved_info);
         startActivity(intent_listview);
      }
   }
});
```

3.2.4 Πώς γίνεται η διαγραφή ενός listview item;

Η διαγραφή ενός listview item (στοιχείο προβολής λίστας) μπορεί να πραγματοποιηθεί όταν το πατήσουμε παρατεταμένα και εφόσον κάνουμε κλικ στο κουμπί **«ΝΑΙ»** ενός μικρότερου παραθύρου (dialog) που θα μας εμφανιστεί.

```
listView.setOnItemLongClickListener(new
AdapterView.OnItemLongClickListener()
{
   @Override
   public boolean onItemLongClick(AdapterView<?> parent, View view, final
int position, long id)
   {
      AlertDialog.Builder dialog = new AlertDialog.Builder(MainActivity.this);
      dialog.setMessage("Είστε σίγουρος πως θέλετε να σβήσετε αυτή την
  αφύπνιση;");
      dialog.setTitle("ΔΙΑΓΡΑΦΗ ΑΦΥΠΝΙΣΗΣ");
      dialog.setCancelable(false);
```

```
        dialog.setIcon(R.drawable.recyclebin);
        dialog.setPositiveButton("NAI", new DialogInterface.OnClickListener()
        {
            @Override
            public void onClick(DialogInterface dialog, int which)
            {
                modelArrayList.remove(position);
                myAdapter.notifyDataSetChanged();
                Intent intent_alarm_off = new Intent(MainActivity.this,
Clock_Activity.class);
                intent_alarm_off.putExtra("alarm_is_off", alarm_is_off);
                startActivity(intent_alarm_off);
            }
        });
          dialog.setNegativeButton("OXI",
                new DialogInterface.OnClickListener()
        {
            @Override
            public void onClick(DialogInterface dialog, int which)
            {
                dialog.dismiss();
            }
        });
        dialog.show();
        return true;
    }
});
```

3.3 Clock_Activity.java

Η κλάση Clock_Activity κάνει όλες τις βασικές λειτουργίες που χρειάζονται για τη ρύθμιση των αφυπνίσεων.

3.3.1 Ορίζοντας την ώρα

Για να ρυθμίσουμε την ώρα που θέλουμε να χτυπήσει το ξυπνητήρι μας, θα πρέπει απλά να διαλέξουμε την κατάλληλη ώρα και λεπτά απ' τον **Time picker** (Επιλογέας Χρόνου).

```
public void onTimeSet(TimePicker view, int hourOfDay, int minute)
{
    Calendar c = Calendar.getInstance();
    c.set(Calendar.HOUR_OF_DAY,hourOfDay);
    c.set(Calendar.MINUTE,minute);
    c.set(Calendar.SECOND,0);
    startAlarm(c);
}
```

3.3.2 Το ξυπνητήρι εν δράση

Με την παρακάτω μέθοδο το ξυπνητήρι μπαίνει σε λειτουργία, δηλαδή ενημερώνει το λειτουργικό σύστημα Android σχετικά με το πότε πρέπει να ηχήσει.

```
public void startAlarm(Calendar c)
{
    my_intent.putExtra("extra", "alarm on");
    my_intent.putExtra("Droid_choice", choose_droid_sound);
    pending_intent = PendingIntent.getBroadcast(Clock_Activity.this,
            0, my_intent,PendingIntent.FLAG_UPDATE_CURRENT);
    String toastText =
DateFormat.getTimeInstance(DateFormat.SHORT).format(c.getTime());
    if(c.before(Calendar.getInstance()))
    {
  c.add(Calendar.DATE,1);
        Toast.makeText(Clock_Activity.this, "Alarm set on: " + toastText + " Next
Day", Toast.LENGTH_LONG).show();
    }
```

```
    else
    {
        Toast.makeText(Clock_Activity.this, "Alarm set on: " + toastText,
Toast.LENGTH_LONG).show();
    }
alarm_manager.setExact(AlarmManager.RTC_WAKEUP,c.getTimeInMillis(),pe
nding_intent);
}
```

3.3.3 Σταματώντας το ξυπνητήρι

Η συγκεκριμένη μέθοδος ακυρώνει την αφύπνισή μας, ακυρώνοντας τη λειτουργία της και απενεργοποιώντας τον επιλεγμένο ήχο της.

```
public void cancelAlarm()
{
    pending_intent = PendingIntent.getBroadcast(Clock_Activity.this,
            0, my_intent,PendingIntent.FLAG_CANCEL_CURRENT);
    alarm_manager.cancel(pending_intent);
    my_intent.putExtra("extra", "alarm off");
    sendBroadcast(my_intent);
}
```

3.3.4 Spiner επιλογής ήχου

Σε αυτό το σημείο δημιουργούμε ένα spiner, το οποίο θα περιλαμβάνει τους ήχους που θα μπορεί να επιλέξει ο κάθε χρήστης για την αφύπνισή του.

```
spinner = (Spinner) findViewById(R.id.spiner);
ArrayAdapter<CharSequence> adapter =
ArrayAdapter.createFromResource(this,
        R.array.Alarm_Droid, android.R.layout.simple_spinner_item);
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdo
wn_item);
```

```
spinner.setAdapter(adapter);
spinner.setOnItemSelectedListener(this);
```

3.3.5 Κουμπί αποθήκευσης

Πατώντας το κουμπί **«ΑΠΟΘΗΚΕΥΣΗ»** αποθηκεύονται οι τρέχουσες ρυθμίσεις μας και επιστρέφουμε στην προηγούμενη σελίδα, όπου τώρα εμφανίστηκε το ξυπνητήρι που μόλις φτιάξαμε.

```
save_button.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v)
    {
      int hour = time_picker.getHour();
      int minute = time_picker.getMinute();
      newhour = hour;
      newminute = minute;
      preferencesToShare();
      result = hour + ":" + minute;

      if((hour < 10) && (minute > 10))
      {
         result = "0" + hour + ":" + minute;
      }
      else if((hour > 10) && (minute < 10))
        {
         result = hour + ":" + "0" + minute;
      }
      else if((hour < 10) && (minute < 10))
      {
         result = "0" + hour + ":" + "0" + minute;
      }
      else
```

```
        {
            result = hour + ":" + minute;
        }
        backToMain();
        onTimeSet(time_picker, hour, minute);
        soundNotification(result);
    }
});
```

3.3.6 Κουμπί ακύρωσης

Πατώντας το κουμπί **«ΑΚΥΡΩΣΗ»** επιστρέφουμε και πάλι στην προηγούμενη σελίδα, αλλά με τη μόνη διαφορά ότι οι ρυθμίσεις που κάναμε έχουν χαθεί.

```
cancel_button.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v)
    {
        onBackPressed();
    }
});
```

3.3.7 Ειδοποιήσεις

Δημιουργώντας το παρακάτω notification (ειδοποίηση), κάθε φορά που εμείς θα πατάμε το κουμπί **«ΑΠΟΘΗΚΕΥΣΗ»** θα εμφανίζεται ένα μικρό εικονίδιο πολύ ψηλά στην οθόνη της συσκευής μας, το οποίο συνοδευόμενο από ένα πολύ μικρό μήνυμα θα μας υπενθυμίζει την ώρα που έχουμε ρυθμίσει να χτυπήσει το ξυπνητήρι.

```
public void soundNotification(String res)
{
    Intent notificationIntent = new Intent(this, MainActivity.class);
```

```
    PendingIntent contentIntent = PendingIntent.getActivity(this, 0,
notificationIntent, PendingIntent.FLAG_UPDATE_CURRENT);
    NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(this,
SOUNDNOT_CHANNEL_ID)
            .setSmallIcon(R.mipmap.blueblackclock)
            .setContentTitle("Alarm Reminder")
            .setContentText("Alarm set on " + res)
            .setPriority(NotificationCompat.PRIORITY_DEFAULT)
            .setContentIntent(contentIntent)
            .setAutoCancel(false);
       notifiticationManager = NotificationManagerCompat.from(this);
    notifiticationManager.notify(1, mBuilder.build());
}
```

3.4 MyAdapter.java

Αυτή η κλάση είναι ιδιαίτερα σημαντική, διότι προσαρμόζει τα βασικά στοιχεία επεξεργασίας της αφύπνισης στην κύρια σελίδα της εφαρμογής.

3.4.1 Προσαρμόζοντας τον Adapter

Σε αυτό το σημείο δημιουργούμε το δικό μας adapter (προσαρμογέα) με σκοπό να καθορίσουμε την τελική δομή του κάθε listview item. Συγκεκριμένα, περιλαμβάνεται ένα textview και ένα switch button (διακόπτη).

```
public View getView(final int position, View convertView, @NonNull final
ViewGroup parent)
{
    if(convertView == null)
    {
        LayoutInflater layoutInflater = (LayoutInflater)
contexts.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        convertView = layoutInflater.inflate(R.layout.listview_item,null,true);
        mViewHolder.mtime = convertView.findViewById(R.id.textView1);
        mViewHolder.mswitch = convertView.findViewById(R.id.switch1);
```

```
        convertView.setTag(mViewHolder);
    }
    else
    {
        mViewHolder = (ViewHolder) convertView.getTag();
    }
    mViewHolder.mtime.setText(ModelArrayList.get(position).getTime());
    mViewHolder.mswitch.setChecked(true);
    final Intent disabled_intent = new Intent(contexts, Clock_Activity.class);
    final Intent enable_intent = new Intent(contexts,Clock_Activity.class);
    mViewHolder.mswitch.setOnClickListener(new View.OnClickListener()
    {
        @Override
        public void onClick(View v)
        {
            if(mViewHolder.mswitch.isChecked())
            {
                enable_intent.putExtra("enable",en);
                contexts.startActivity(enable_intent);
            }
            else
            {
                 disabled_intent.putExtra("disable", dis);
                contexts.startActivity(disabled_intent);
            }
        }
    });
    return convertView;
}
```

3.4.2 Model.java

Σε αυτήν την κλάση ορίζουμε τις τιμές της ώρας και του switch button, οι οποίες επιστρέφονται στον προσαρμοσμένο (custom) adapter.

```
public class Model
{
    private String time;
    private int switch_btn;
    private int delete_btn;

    public void setTime(String set_time)
    {
        this.time = set_time;
    }
    public String getTime()
    {
        return time;
    }
    public void setSwitch_btn(int set_switch_btn)
    {
        this.switch_btn = set_switch_btn;
    }
    public int getSwitch_btn()
      {
        return switch_btn;
    }
}
```

3.5 FullScreenDialog.java

Μέσα από την κλάση FullScreenDialog δημιουργείται η πλήρης οθόνη που εμφανίζεται όταν ηχεί το ξυπνητήρι.

3.5.1 Σχεδιάζοντας το Full Screen παράθυρο

Σε αυτές τις γραμμές κώδικα σχεδιάζουμε τις διαστάσεις του παραθύρου πλήρης οθόνης.

```
@Override
public void onStart()
{
    super.onStart();
    Dialog dialog = getDialog();
    if(dialog!=null)
    {
        int width = ViewGroup.LayoutParams.MATCH_PARENT;
        int height = ViewGroup.LayoutParams.MATCH_PARENT;
        dialog.getWindow().setLayout(width,height);
    }
}
@Override
public void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setStyle(DialogFragment.STYLE_NORMAL,R.style.FullScreenDialogStyle);
}
```

3.5.2 Εφαρμόζοντας τις λειτουργίες πάνω στο Full Screen παράθυρο

Το παράθυρο πλήρης οθόνης εμφανίζεται τη στιγμή που έχει προγραμματιστεί να χτυπήσει το ξυπνητήρι μας. Επίσης, περιέχει ένα textview που γράφει την τρέχουσα ώρα και δύο κουμπιά ονόματι **«ΑΝΑΒΟΛΗ»** και **«ΜΑΤΑΙΩΣΗ»**. Το κουμπί **«ΑΝΑΒΟΛΗ»** όταν πατηθεί, εξαφανίζει το παράθυρο πλήρης οθόνης προσωρινά, έως ότου ξαναχτυπήσει το ξυπνητήρι με βάση το χρόνο αναβολής που έχουμε επιλέξει μέσω των ρυθμίσεων. Τότε θα εμφανιστεί ξανά. Αντίθετα, το κουμπί **«ΜΑΤΑΙΩΣΗ»** όταν πατηθεί, εξαφανίζει επίσης το παράθυρο πλήρης οθόνης, αλλά όχι προσωρινά. Δηλαδή, ακυρώνει για εκείνη τη μέρα το ξυπνητήρι και θα εμφανιστεί ξανά μόνο εάν στις ρυθμίσεις της αναβολής έχουμε επιλέξει το ξυπνητήρι μας να χτυπάει κάποιες συγκεκριμένες μέρες.

```
@Nullable
@Override
public View onCreateView(final LayoutInflater inflater, @Nullable final
ViewGroup container, final Bundle savedInstanceState)
{
    super.onCreateView(inflater, container, savedInstanceState);
    View view =
getActivity().getLayoutInflater().inflate(R.layout.fullscreendialog,container,false
);
    final Dialog dialog = getDialog();
    int time_hour = getArguments().getInt("time_extra");
    int time_minute = getArguments().getInt("time_extra_2");
    String hour_and_minute = time_hour + ":" + time_minute;

    if((time_hour < 10) && (time_minute < 10))
    {
        hour_and_minute = "0" + time_hour + ":" + "0" + time_minute;
    }
    else if((time_hour < 10) && (time_minute > 10))
      {
        hour_and_minute = "0" + time_hour + ":" + time_minute;
    }
    else if((time_hour > 10) && (time_minute < 10))
    {
        hour_and_minute = time_hour + ":" + "0" + time_minute;
    }
    else
    {
        hour_and_minute = time_hour + ":" + time_minute;
    }

    TextView textview_time_from_clock =
view.findViewById(R.id.textView_time);
    textview_time_from_clock.setText(String.valueOf(hour_and_minute));
```

```
    Button button_dis = view.findViewById(R.id.button_dismiss);
    button_dis.setOnClickListener(new View.OnClickListener()
    {
        @Override
        public void onClick(View v)
        {
            dialog.dismiss();
            ((Clock_Activity)getActivity()).cancelAlarm();
            ((Clock_Activity)getActivity()).finish();
        }
    });
    return view;
}
```

3.6 Alarm_receiver.java

Αυτή η κλάση λειτουργεί κυρίως ως μεσάζων, καθώς δέχεται και στέλνει κάποια απαραίτητα δεδομένα για τη λειτουργία ή τη διακοπή της αφύπνισης αντίστοιχα.

```
@Override
public void onReceive(Context context, Intent intent)
{
    String get_your_string = intent.getExtras().getString("extra");
    Integer get_your_Droid_choice = intent.getExtras().getInt("Droid_choice");
    Intent service_intent = new Intent(context,RingtonePlayingService.class);
    service_intent.putExtra("extra", get_your_string);
    service_intent.putExtra("Droid_choice", get_your_Droid_choice);
    context.startService(service_intent);
}
```

3.7 RingtonePlayingServices.java

Η κλάση RingtonePlayingServices ασχολείται κυρίως με τους ήχους της εφαρμογής, καθώς και με την επεξεργασία τους.

3.7.1 Περιπτώσεις ήχων

Παρακάτω βλέπουμε 4 βασικές περιπτώσεις. Στην πρώτη περίπτωση εάν δεν παίζει κάποιος ήχος, τότε πατώντας το κατάλληλο κουμπί θα ξεκινήσει. Στη δεύτερη περίπτωση, εάν παίζει ήδη κάποιος ήχος, με το πάτημα ενός κουμπιού θα σταματήσει. Στην τρίτη μας περίπτωση, αν δεν παίζει κάποιος ήχος κι εμείς πατάμε το κουμπί «**Turn off**» δεν θα συμβεί τίποτα. Αντίστοιχα, στην τελευταία περίπτωση, πάλι δεν θα συμβεί κάτι, αλλά με τη μόνη διαφορά ότι παίζει ήδη κάποιος ήχος κι εμείς πατάμε το κουμπί «**Turn on**».

```
if (!this.isRunning && startID == 1)
{
 i("There is no music, ", "and you want to START");
 this.isRunning = true;
 this.startID = 0;
 Intent clock_activity = new Intent(this, Clock_Activity.class);
 clock_activity.putExtra("fullscreen",fullscr);
 startActivity(clock_activity);

 if (Droid_sound_choice == 0)
 {
    media_song = MediaPlayer.create(this, R.raw.alarmclock);
    media_song.start();
 }
 else if (Droid_sound_choice == 1)
 {
    media_song = MediaPlayer.create(this, R.raw.catscream);
    media_song.start();
 }
 else if (Droid_sound_choice == 2)
 {
    media_song = MediaPlayer.create(this, R.raw.chicken);
    media_song.start();
 }
```

```
else if (Droid_sound_choice == 3)
{
    media_song = MediaPlayer.create(this, R.raw.cowmoo);
    media_song.start();
}
  else if (Droid_sound_choice == 4)
{
    media_song = MediaPlayer.create(this, R.raw.dixiehorn);
    media_song.start();
}
else if (Droid_sound_choice == 5)
{
    media_song = MediaPlayer.create(this, R.raw.funnyvoices);
    media_song.start();
}
else if (Droid_sound_choice == 6)
{
    media_song = MediaPlayer.create(this, R.raw.glasswater);
    media_song.start();
}
else if (Droid_sound_choice == 7)
{
    media_song = MediaPlayer.create(this, R.raw.hahaha);
    media_song.start();
}
else if (Droid_sound_choice == 8)
{
    media_song = MediaPlayer.create(this, R.raw.heavyrain);
    media_song.start();
}
else if (Droid_sound_choice == 9)
{
    media_song = MediaPlayer.create(this, R.raw.henchicken);
    media_song.start();
```

```
}
else if (Droid_sound_choice == 10)
{
    media_song = MediaPlayer.create(this, R.raw.jollylaugh);
   media_song.start();
}
else if (Droid_sound_choice == 11)
{
   media_song = MediaPlayer.create(this, R.raw.newsintro);
   media_song.start();
}
else if (Droid_sound_choice == 12)
{
   media_song = MediaPlayer.create(this, R.raw.pagerbeeps);
   media_song.start();
}
else if (Droid_sound_choice == 13)
{
   media_song = MediaPlayer.create(this, R.raw.police);
   media_song.start();
}
else if (Droid_sound_choice == 14)
{
   media_song = MediaPlayer.create(this, R.raw.quack);
   media_song.start();
}
else if (Droid_sound_choice == 15)
{
   media_song = MediaPlayer.create(this, R.raw.rooster);
   media_song.start();
}
else if (Droid_sound_choice == 16)
{
```

```
        media_song = MediaPlayer.create(this, R.raw.scaryscream);
        media_song.start();
      }
    else if (Droid_sound_choice == 17)
    {
        media_song = MediaPlayer.create(this, R.raw.sillysnoring);
        media_song.start();
    }
    else if (Droid_sound_choice == 18)
    {
        media_song = MediaPlayer.create(this, R.raw.sleighbells);
        media_song.start();
    }
    else if (Droid_sound_choice == 19)
    {
        media_song = MediaPlayer.create(this, R.raw.tornadosiren);
        media_song.start();
    }
    else if (Droid_sound_choice == 20)
    {
        media_song = MediaPlayer.create(this, R.raw.wakeup);
        media_song.start();
    }
    else if (Droid_sound_choice == 21)
    {
        media_song = MediaPlayer.create(this, R.raw.weatheralert);
        media_song.start();
    }
    else
    {
        media_song = MediaPlayer.create(this, R.raw.sillysnoring);
        media_song.start();
    }
}
```

```
else if (this.isRunning && startID == 0) {
    i("There is music, ", "and you want to STOP");
    media_song.stop();
    media_song.reset();
    this.isRunning = false;
    this.startID = 0;
}
else if (!this.isRunning && startID == 0) {
    i("There is no music, ", "and you want to STOP");
    this.isRunning = false;
    this.startID = 0;
}
else if (this.isRunning && startID == 1) {
    i("There is music, ", "and you want to START");
    this.isRunning = true;
    this.startID = 1;
}
else
  {
    i("Else ", "somehow you reached this");
}
```

3.7.2 Τι κάνει η μέθοδος onDestroy();

Σε περίπτωση που κάτι πάει στραβά με την κλάση, στην οποία βρίσκεται η onDestroy() και δεν λειτουργήσει τίποτα σωστά, τότε καλείται η μέθοδός μας.

```
@Override
public void onDestroy()
{
    i("on Destroy called", "Crashed");
    super.onDestroy();
    this.isRunning = false;
}
```

3.8 activity_main.xml – content_main.xml

Σε αυτά τα δύο xml αρχεία βρίσκεται ο σχεδιασμός της αρχικής σελίδας της εφαρμογής μας. Συγκεκριμένα έχουμε μία λίστα από διάφορα αντικείμενα, τα οποία αντιπροσωπεύουν τα εκάστοτε ξυπνητήρια και ένα κουμπί, το οποίο μας βοηθάει κάθε φορά να δημιουργήσουμε μία νέα αφύπνιση.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/ThemeOverlay.AppCompat.Light"
            app:theme="@style/ThemeOverlay.AppCompat.Dark" />

    </android.support.design.widget.AppBarLayout>
```

```
    <include layout="@layout/content_main" />
</android.support.design.widget.CoordinatorLayout>
```

content_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context=".MainActivity"
    tools:showIn="@layout/activity_main">

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/floatingActionButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:clickable="true"
       android:focusable="true"
        app:backgroundTint="@android:color/holo_blue_dark"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="0.924"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.929"
        app:srcCompat="@drawable/time_fragment"
        tools:backgroundTint="@android:color/holo_blue_light" />

        <ListView
            android:id="@+id/listview"
```

```
        android:focusable="false"
        android:layout_width="368dp"
        android:layout_height="422dp"
        android:choiceMode="multipleChoice"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.056" />
</android.support.constraint.ConstraintLayout>
```

3.9 activity_clock.xml – content_clock.xml

Τα xml μας περιέχουν ένα ρολόι, με το οποίο ρυθμίζουμε την ώρα, το spinner για την επιλογή του ήχου που επιθυμούμε και δύο κουμπιά «**ΑΠΟΘΗΚΕΥΣΗ**» και «**ΑΚΥΡΩΣΗ**».

activity_clock.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Clock_Activity">

  <android.support.design.widget.AppBarLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:theme="@style/AppTheme.AppBarOverlay">
```

```
        <android.support.v7.widget.Toolbar
         android:id="@+id/toolbar"
         android:layout_width="match_parent"
         android:layout_height="?attr/actionBarSize"
         android:background="?attr/colorPrimary"
         app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>
    <include layout="@layout/content_clock_" />
</android.support.design.widget.CoordinatorLayout>
```

content_clock.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app=http://schemas.android.com/apk/res-auto
      xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context=".Clock_Activity">

    <TimePicker
       android:id="@+id/timePicker"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       app:layout_constraintBottom_toBottomOf="parent"
       app:layout_constraintLeft_toLeftOf="parent"
       app:layout_constraintRight_toRightOf="parent"
       app:layout_constraintTop_toTopOf="parent"
       app:layout_constraintVertical_bias="0.73">
    </TimePicker>
```

```
    <Spinner
     android:id="@+id/spiner"
     android:layout_width="368dp"
     android:layout_height="49dp"
     android:layout_alignParentStart="true"
     android:layout_alignParentTop="true"
     android:layout_marginStart="9dp"
     android:layout_marginTop="392dp"
     app:layout_constraintBottom_toBottomOf="parent"
     app:layout_constraintLeft_toLeftOf="parent"
     app:layout_constraintRight_toRightOf="parent"
     app:layout_constraintTop_toTopOf="parent"
     app:layout_constraintVertical_bias="0.73" />

<LinearLayout
     android:layout_width="match_parent"
       android:layout_height="50dp"
     android:layout_alignParentStart="true"
     android:layout_alignParentBottom="true"
     android:layout_marginStart="0dp"
     android:layout_marginBottom="0dp"
     android:orientation="horizontal">

     <Button
        android:id="@+id/save"
        android:layout_width="185dp"
        android:layout_height="62dp"
        android:layout_marginStart="0dp"
        android:layout_marginBottom="283dp"
        android:text="ΑΠΟΘΗΚΕΥΣΗ"
        android:textSize="14sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintLeft_toLeftOf="parent"
```

```
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.53" />

    <Button
        android:id="@+id/cancel"
        android:layout_width="210dp"
        android:layout_height="63dp"
        android:layout_marginTop="-1dp"
        android:layout_marginEnd="0dp"
        android:text="Ακύρωση"
        android:textSize="14sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
          app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.66" />
  </LinearLayout>
</RelativeLayout>
```

3.10 listview_item.xml

Αυτό το xml αντιπροσωπεύει το κάθε αντικείμενο της λίστας της αρχικής μας σελίδας. Αποτελείται από ένα switch button, το οποίο ενεργοποιεί ή απενεργοποιεί αντίστοιχα την κάθε αφύπνιση, αλλά και από ένα textview, το οποίο μας δείχνει την προγραμματισμένη ώρα που θα χτυπήσει το κάθε ξυπνητήρι.

```
<?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:orientation="horizontal">

<TextView
        android:id="@+id/textView1"
        android:layout_width="228dp"
        android:layout_height="48dp"
        android:focusable="false"
        android:text="@string/app_name"
        android:textAlignment="center"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        android:textSize="25sp" />

<Switch
        android:id="@+id/switch1"
        android:layout_width="55dp"
        android:layout_height="52dp"
        android:focusable="false" />
</LinearLayout>
```

3.11 fullscreendialog.xml

Το τελευταίο μας xml δείχνει το οπτικό κομμάτι της οθόνης που εμφανίζεται κάθε φορά που ηχεί το ξυπνητήρι μας.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/relative_outside"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/beach"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
```

```
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <Button
        android:id="@+id/button_dismiss"
        android:layout_width="177dp"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_marginTop="215dp"
        android:layout_marginEnd="112dp"
        android:drawableStart="@drawable/ic_notifications_black_24dp"
        android:text="ΜΑΤΑΙΩΣΗ" />

    <TextView
        android:id="@+id/textView_time"
        android:layout_width="177dp"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_marginTop="173dp"
        android:layout_marginEnd="112dp"
        android:text="TIME:"
        android:textAlignment="center"
        android:textSize="30sp" />
</RelativeLayout>
```

ΚΕΦΑΛΑΙΟ 4

ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ ABCLOCK

4.1 Εισαγωγή

Το κεφάλαιο 4 περιγράφει την εφαρμογή αφύπνισης μέσα από τα μάτια του απλού χρήστη αναλύοντας τα βασικά παράθυρα, τις λειτουργίες, καθώς και τις δυνατότητες που δίνονται σε αυτόν.

4.2 Η κύρια σελίδα δημιουργίας αφύπνισης

Αρχικά, στην πρώτη και κύρια σελίδα της εφαρμογής μας, την πρώτη φορά που μπαίνουμε βλέπουμε πως είναι άδεια. Το μοναδικό αντικείμενο που υπάρχει είναι ένα floating action button (Κουμπί επιπλέουσας ενέργειας). Με το πάτημά του μεταβαίνουμε στην επόμενη σελίδα όπου και δημιουργούμε μία αφύπνιση με τις ρυθμίσεις που επιθυμούμε.

Εικόνα 4.2.1 – Αρχική σελίδα εφαρμογής

΄Υστερα από τη δημιουργία μίας αφύπνισης εμφανίζεται ένα listview item, το οποίο αναγράφει την ώρα που ρυθμίσαμε να ηχήσει το ξυπνητήρι μας και φέρει ένα κουμπί on/off. Με αυτό το κουμπί μπορούμε να ενεργοποιήσουμε ή να απενεργοποιήσουμε το ξυπνητήρι μας. Επίσης, στο πάνω πάνω μέρος της οθόνης εμφανίζεται ένα μικρό εικονίδιο που αντιπροσωπεύει τη ρυθμισμένη αφύπνιση και αναγράφει την ώρα που την έχουμε ρυθμίσει. Εάν πατήσουμε πάνω του θα μεταβούμε στην αρχική μας σελίδα.

Εικόνα 4.2.2 – Αρχική σελίδα με ενεργοποίηση αφύπνισης

Εικόνα 4.2.3 – Αρχική σελίδα με απενεργοποίηση αφύπνισης

Εάν κάποια στιγμή θελήσουμε να διαγράψουμε την αφύπνιση που δημιουργήσαμε, το μόνο που πρέπει να κάνουμε είναι να πατήσουμε παρατεταμένα το listview item. Τότε, θα εμφανιστεί ένα μικρότερο παράθυρο, το οποίο θα μας ρωτάει αν είμαστε σίγουροι πως θέλουμε να τη διαγράψουμε. Με το κουμπί «ΝΑΙ» θα διαγραφεί η αφύπνιση και θα εξαφανιστεί το μικρό εικονίδιο που είχε εμφανιστεί όταν τη δημιουργήσαμε, ενώ με το κουμπί «ΟΧΙ» θα παραμείνουν ως έχουν.

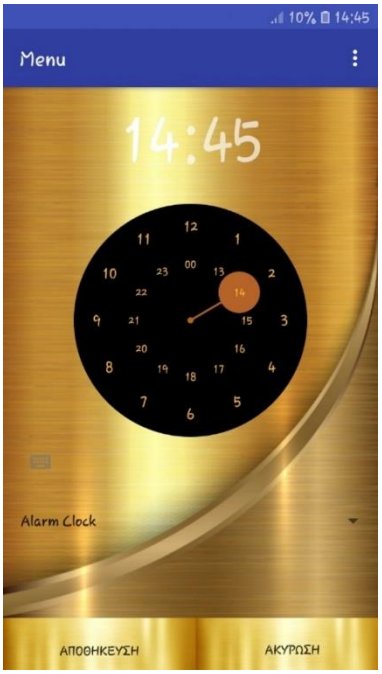
Εικόνα 4.2.4 – Διαγραφή αφύπνισης

Τέλος, αν είμαστε στην πρώτη σελίδα και θελήσουμε να βγούμε από την εφαρμογή ή πατήσουμε καταλάθος το κουμπί «πίσω», θα μας εμφανιστεί άλλο ένα μικρό παράθυρο που θα μας λέει πως μπορούμε να εξέλθουμε μόνο πατώντας το κεντρικό κουμπί.

Εικόνα 4.2.5 – Έξοδος από την εφαρμογή

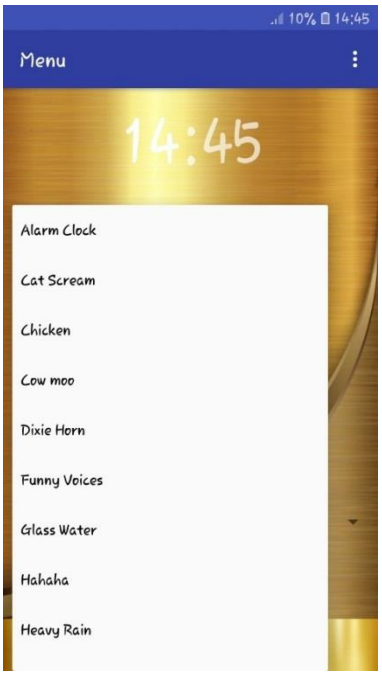
4.3 Η δεύτερη σελίδα της εφαρμογής με τις βασικές ρυθμίσεις

Στη δεύτερη σελίδα διακρίνουμε περισσότερα αντικείμενα. Πρώτα απ' όλα υπάρχει ένα ρολόι, το οποίο μας βοηθάει με την ακριβή επιλογή της ώρας που θέλουμε. Σε αυτό το σημείο να τονίσουμε πως το ρολόι μας, εμφανίζει την ώρα σε δωδεκάωρη και εικοσιτετράωρη μορφή ταυτόχρονα για τη διευκόλυνση των χρηστών. Βέβαια, λίγο πιο κάτω και αριστερά του ρολογιού μπορούμε να διακρίνουμε ένα κουμπάκι. Αν το πατήσουμε θα δούμε μία πιο απλή έκδοση ρύθμισης της ώρας. Με όποιον τρόπο όμως και να τη ρυθμίσουμε, η ώρα αναγράφεται ακριβώς επάνω από το ρολόι μας.



Εικόνα 4.3.1 – Κύρια σελίδα δημιουργίας αφύπνισης

Λίγο πιο κάτω από το ρολόι ρύθμισης της ώρας διακρίνουμε την ύπαρξη ενός πολλαπλού επιλογέα ήχων (spiner). Όταν το πατήσουμε θα δούμε ένα πλήθος εικοσιδύο ήχων. Ο χρήστης μπορεί να επιλέξει όποιον επιθυμεί για την αφύπνισή του.



Εικόνα 4.3.2 – Λίστα ήχων

Τέλος, στο κάτω μέρος της τρέχουσας σελίδας βλέπουμε δύο κουμπιά ονόματι «**ΑΠΟΘΗΚΕΥΣΗ**» και «**ΑΚΥΡΩΣΗ**». Με το πάτημα του κουμπιού «**ΑΠΟΘΗΚΕΥΣΗ**» αποθηκεύονται όλες οι ρυθμίσεις που έχουμε κάνει, δηλαδή η ώρα και ο ήχος που έχουμε επιλέξει και στην πρώτη σελίδα της εφαρμογής δημιουργείται ένα listview item με τις τρέχουσες ρυθμίσεις μας. Αντίθετα, με το κουμπί «**ΑΚΥΡΩΣΗ**» χάνονται όλες οι ρυθμίσεις που έχουμε κάνει. Οπότε, εάν προηγουμένως είχαμε φτιάξει μία αφύπνιση, οι νέες ρυθμίσεις θα χαθούν. Διαφορετικά δεν θα συμβεί απολύτως τίποτα. Και στις δύο περιπτώσεις όμως, υπάρχει μόνο ένα κοινό σημείο. Με το πάτημά τους πηγαίνουμε στην προηγούμενη σελίδα.

4.4 Η πλήρης οθόνη κατά τη διάρκεια της αφύπνισης

Εφόσον είναι όλα ρυθμισμένα, περιμένουμε να χτυπήσει το ξυπνητήρι μας. Όταν έρθει αυτή η στιγμή θα δούμε μία πλήρη οθόνη με δύο κουμπιά ονόματι «**ΜΑΤΑΙΩΣΗ**» και «**ΑΝΑΒΟΛΗ**» και την ώρα να αναγράφεται στο μέσο της οθόνης μας.

Εικόνα 4.4.1 – Η πλήρης οθόνη όταν ηχεί το ξυπνητήρι

Με το κουμπί «**ΜΑΤΑΙΩΣΗ**» η αφύπνισή μας ακυρώνεται για την τρέχουσα ημέρα. Αν την έχουμε ρυθμίσει για κάποιες συγκεκριμένες ημέρες, θα πρέπει να περιμένουμε έως την επόμενη φορά, με την προϋπόθεση να μην την απενεργοποιήσουμε. Η διαφορά με το κουμπί «**ΑΝΑΒΟΛΗ**» είναι πως η αφύπνιση μπαίνει σε αναστολή για λίγα λεπτά και μετά ηχεί ξανά.

ΚΕΦΑΛΑΙΟ 5

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΟΠΤΙΚΕΣ

5.1 Σύνοψη της πτυχιακής εργασίας

Από την πτυχιακή εργασία που ολοκληρώσαμε μπορούμε να διαπιστώσουμε ότι μία εφαρμογή μπορεί να υλοποιηθεί σχετικά εύκολα και με απλά βήματα, έχοντας θέληση, αλλά και υπομονή μιας και δεν είναι ό,τι πιο εύκολο.

Πρώτα απ' όλα, δημιουργήσαμε μία κλάση που περιέχει την αρχική σελίδα της εφαρμογής μας. Σε αυτή, λοιπόν, τη σελίδα μπορεί ο χρήστης να πατήσει ένα κουμπί που προσθέσαμε εμείς, με σκοπό να πάει στην αμέσως επόμενη σελίδα. Εκεί, θα διαλέξει την ώρα που επιθυμεί να χτυπήσει το ξυπνητήρι του, όπως και τον ήχο που του αρέσει και θα ήθελε να ακούσει. Εφόσον επιλέξει ώρα και ήχο, έχει δύο επιλογές. Είτε να πατήσει το κουμπί **«ΑΠΟΘΗΚΕΥΣΗ»**, είτε το κουμπί **«ΑΚΥΡΩΣΗ»**. Εάν πατήσει το κουμπί **«ΑΠΟΘΗΚΕΥΣΗ»** θα πάει πίσω στην αρχική σελίδα, θα εμφανιστεί ένα listview item με τις ρυθμίσεις που επέλεξε ο ίδιος ο χρήστης και θα δημιουργηθεί ένα notification, δηλαδή πάνω πάνω στην οθόνη της συσκευής θα εμφανιστεί ένα μικρό εικονίδιο, στο οποίο θα αναγράφεται η ώρα της προγραμματισμένης αφύπνισης. Σε αυτό το σημείο πρέπει να διευκρινήσουμε πως το κάθε listview item διαθέτει ένα έξυπνο κουμπάκι, το οποίο ενεργοποιεί ή απενεργοποιεί κάθε φορά την αφύπνιση που έχουμε δημιουργήσει. Διαφορετικά εάν επιλέξει το κουμπί **«ΑΚΥΡΩΣΗ»** δύο πράγματα μπορούν να συμβούν. Το ένα είναι να πάει ο χρήστης να δημιουργήσει μία πρώτη αφύπνιση, οπότε με το πάτημα αυτού του κουμπιού θα πάει πίσω στην αρχική σελίδα, χωρίς όμως να υπάρχει αφύπνιση. Το άλλο είναι να υπάρχει ήδη μία τουλάχιστον αφύπνιση και όταν πατηθεί το κουμπί να πάει μεν πίσω, αλλά και πάλι να μην δημιουργηθεί κάποια νέα αφύπνιση. Αυτές οι δύο περιπτώσεις έχουν ένα κοινό σημείο. Σε κάθε πάτημα του κουμπιού **«ΑΚΥΡΩΣΗ»,** οι ρυθμίσεις που έχει κάνει ο χρήστης της εφαρμογής χάνονται.

Απ' τη στιγμή που ο χρήστης έχει φτιάξει μία τουλάχιστον αφύπνιση έχει και τη δυνατότητα να τη διαγράψει. Πώς γίνεται αυτό; Πατώντας παρατεταμένα το

listview item. Εκείνη τη στιγμή θα εμφανιστεί ένα μικρότερο παράθυρο που θα μας ρωτάει αν είμαστε σίγουροι ότι θέλουμε να τη διαγράψουμε. Πατώντας το κουμπί **«ΝΑΙ»** διαγράφεται το listview item και το notification, ενώ με το κουμπί **«ΟΧΙ»** παραμένουν. Παρεπιπτόντως, το notification παύει να εμφανίζεται ακόμα κι όταν απενεργοποιούμε την αφύπνιση.

Όταν, λοιπόν, είναι όλα έτοιμα και προγραμματισμένα απλά περιμένουμε να χτυπήσει το ξυπνητήρι μας. Μόλις έρθει εκείνη η στιγμή ακούγεται ο ήχος που έχει επιλέξει ο χρήστης, η τρέχουσα ώρα και εμφανίζεται ένα μοντέρνο background (φόντο) μαζί με δύο κουμπιά **«ΑΝΑΒΟΛΗ»** και **«ΜΑΤΑΙΩΣΗ»**. Εδώ τώρα, εάν ο χρήστης πατήσει το κουμπί **«ΜΑΤΑΙΩΣΗ»,** το ξυπνητήρι δεν θα χτυπήσει ξανά, τουλάχιστον για την ίδια μέρα. Αν όμως, το έχει προγραμματίσει ώστε να ηχήσει κάποια άλλη ημέρα, θα πρέπει να περιμένει έως τότε. Διαφορετικά, εάν πατήσει το κουμπί **«ΑΝΑΒΟΛΗ»,** το ξυπνητήρι θα ηχήσει μετά από λίγη ώρα, πράγμα που το ρυθμίζει ο χρήστης.

5.2 Μελλοντικές προοπτικές

Σκοπός μας μετά το πέρας της τρέχουσας εργασίας είναι η δημιουργία ακόμα περισσότερων εφαρμογών σε περιβάλλον Android. Έχουμε, λοιπόν, σαν στόχο όλες μας οι εφαρμογές να ανεβαίνουν στο Play Store της Google για να μπορούμε να μοιραζόμαστε τις ιδέες που υλοποιούμε με τον υπόλοιπο κόσμο.

Όσο αναφορά την πτυχιακή μας εργασία, αφού εξεταστούμε σε αυτή, έχουμε τη θέληση να την προχωρήσουμε κι άλλο. Δηλαδή, να προσθέσουμε περισσότερες λειτουργίες ώστε να γίνει πιο πλούσια και δημοφιλής. Τέλος, θα θέλαμε να είναι και οπτικά ενδιαφέρουσα, πράγμα που πολλοί χρήστες παρατηρούν καλά πριν κατεβάσουν μία νέα εφαρμογή.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Android - Βικιπαίδεια. (2018, 10 12). Retrievedfrom ΒΙΚΙΠΑΙΔΕΙΑ: https://el.wikipedia.org/wiki/Android

[2] Android 4.4 KitKat: Τι καινούριο φέρνει η νέα έκδοση . (2013). Retrieved from Techgear.gr: https://www.techgear.gr/android-4-4-kitkat-whats-new-79727/

[3] Android Authority. (2017). Retrieved from Android Authority.

[4] Android Cupcake. (17, 1 17). Retrieved from Wikipedia: https://el.wikipedia.org/wiki/Android_Cupcake

[5] Android Marshmallow. (2017). Retrieved from Βικιπαίδεια: https://el.wikipedia.org/wiki/Android_Marshmallow

[6] Android Studio. (2017). Retrieved from Wikipedia: https://el.wikipedia.org/wiki/Android_Studio

[7] Android - Digital Academy. (n.d.). Retrieved from Digital Academy: http://www.dga.gr/web/publications/files/android.pdf

[8] Astrium, D. (2014). Android - 4.3 Jelly Bean. Retrieved from android: https://www.android.com/versions/jelly-bean-4-3/

[9] AVD Manager. (n.d.). Retrieved from Android Developers: https://stuff.mit.edu/afs/sipb/project/android/docs/tools/help/avd-manager.html

[10] Build and share Your APK File Unit. (2018). Retrieved from Trailhead: https://trailhead.salesforce.com/en/content/learn/projects/workshop-virtual-reality/build-share-apk

[11] Debugging in Android Studio. (2016). Retrieved from STRV: https://www.strv.com/blog/debugging-in-android-studio-as

[12] Google. (2018, 12 3). Retrieved from Βικιπαίδεια: https://translate.google.gr/translate?hl=el&sl=en&tl=el&u=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FAndroid_version_history&anno=2&sandbox=1

[13] Google: Ανακοίνωσε το Android 2.2 (froyo) - myphone.gr. (2018). Retrieved from myphone.

[14] Happy Byte Blog. (n.d.). Retrieved from HappyByte: https://happybyte.gr/blog/android-8-oreo-13-kainoyrgies-allages

[15] Headlines, A. (2018). Android 1.6 Donut. Retrieved from Android Headlines: https://www.androidheadlines.com/android-1-6-donut

[16] Media, V. (2018). Vox Media. Retrieved from The Verge: https://translate.google.gr/translate?hl=el&sl=en&u=https://www.theverge.com/2011/12/7/2585779/android-10th-anniversary-google-history-pie-oreo-nougat-cupcake&prev=search

[17] Meet Android Studio. (2018). Retrieved from Developers: https://developer.android.com/studio/intro/

[18] Meet Android Studio. (2018). Retrieved from Developers: https://developer.android.com/studio/intro/

[19] SDK Manager. (n.d.). Retrieved from Developers: https://stuff.mit.edu/afs/sipb/project/android/docs/tools/help/sdk-manager.html

[20] Techgear.gr. (2011, 2 2). Retrieved from Techgear.gr: https://www.techgear.gr/android-3-0-honeycomb-officially-presented-16752/

[21] TechNorms. (2018). Android 5.1 Lollipop Review. Retrieved from TechNorms: https://www.technorms.com/44194/android-51-lollipop-review

[22] What You Need To Know About Android 2.3 Gingerbread | WIRED. (2018). Retrieved from Wired: https://www.wired.com/2010/12/android-gingerbread-2/

[23] Ιστορία εκδόσεων του Android. (2018, 9 3). Retrieved from Βικιπαίδεια: https://el.wikipedia.org/wiki/%CE%99%CF%83%CF%84%CE%BF%CF%81%CE%AF%CE%B1_%CE%B5%CE%BA%CE%B4%CF%8C%CF%83%CE%B5%CF%89%CE%BD_%CF%84%CE%BF%CF%85_Android

[24] Τι αλλάζει στο smartphone σας με το Android 7.0 Nougat | in.gr. (n.d.). Retrieved from in.gr: https://www.in.gr/2017/03/20/tech/future/ti-allazei-sto-smartphone-sas-me-to-android-7-0-nougat/

[25] Τι νέο υπάρχει στο AndroidIceCreamSandwich 4.0. (2018). Retrieved from amtelefon.com: https://el.amtelefon.com/ce-este-nou-in-android-ice-cream-sandwich-4-0/mobile-apps/

ΠΑΡΑΡΤΗΜΑ – ΚΩΔΙΚΑΣ ΕΦΑΡΜΟΓΗΣ

```
public class Alarm_Receiver extends BroadcastReceiver
{
    @Override
    public void onReceive(Context context, Intent intent)
    {
        // Fetch extra strings from the intent tells the app whether the user pressed the alarm on button or the alarm off button
        String get_your_string = intent.getExtras().getString("extra");
        Log.i("What is the key? " , get_your_string);
        //Fetch extra strings from the intent tells the app which value the user picked from the drop-down menu/spinner
        Integer get_your_Droid_choice = intent.getExtras().getInt("Droid_choice");
        Log.i("Your droid choice is:",get_your_Droid_choice.toString());
        // Pass the extra integer from the Receiver to the Ringtone Playing Service
        service_intent.putExtra("Droid_choice", get_your_Droid_choice);
    }
}

public class Clock_Activity extends AppCompatActivity implements AdapterView.OnItemSelectedListener,TimePickerDialog.OnTimeSetListener
{
    @RequiresApi(api = Build.VERSION_CODES.O)
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_clock_);
        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        setTitle("Menu");
        iniTiaLiSations();
        createChannelNotification();
        if(fullscreen)
        {
            FullScreenDialog dialog = new FullScreenDialog();
getWindow().addFlags(WindowManager.LayoutParams.FLAG_SHOW_WHEN_LOCKED);
            getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
```

```
        getWindow().addFlags(WindowManager.LayoutParams.FLAG_TURN_SCREEN_ON);
        FragmentTransaction ft = getFragmentManager().beginTransaction();
        dialog.show(ft,FullScreenDialog.TAG);
        newhour = sharedpreferences.getInt("message1", 0);
        newminute = sharedpreferences.getInt("message2",0);
        Bundle args = new Bundle(2);
        args.putInt("time_extra",newhour);
        args.putInt("time_extra_2",newminute);
        dialog.setArguments(args);
    }

    boolean switch_btt_on = switch_intent.getBooleanExtra("enable",false);
    if(switch_btt_on)
    {
        newhour = sharedpreferences.getInt("message1", 0);
        newminute = sharedpreferences.getInt("message2",0);
        onTimeSet(time_picker, newhour, newminute);
        String result_switch_btn_on = newhour + ":" + newminute;
        if(newminute < 10)
        {
            result_switch_btn_on = newhour + ":" + "0" + newminute;
        }
        soundNotification(result_switch_btn_on);
    }
    boolean switch_btt_off = switch_intent.getBooleanExtra("disable", false);
    if(switch_btt_off)
    {
        cancelAlarm();
        notifiticationManager.cancel(1);
    }
    boolean alarm_turned_off = info.getBooleanExtra("alarm_is_off", false);
    if(alarm_turned_off)
    {
        cancelAlarm();
        notifiticationManager = NotificationManagerCompat.from(this);
        notifiticationManager.cancel(1);
    }
    boolean saved_informations = info.getBooleanExtra("svd_info", false);
    if(saved_informations)
    {
```

```
        newhour = sharedpreferences.getInt("message1", 0);
        newminute = sharedpreferences.getInt("message2",0);
        choose_droid_sound = sharedpreferences.getInt("sound", 0);
        time_picker.setHour(newhour);
        time_picker.setMinute(newminute);
        spinner.setSelection(choose_droid_sound);
    }

    // Create onClick to start the alarm
    save_button.setOnClickListener(new View.OnClickListener()
    {
        @Override
        public void onClick(View v)
        {
            int hour = time_picker.getHour();
            int minute = time_picker.getMinute();
            newhour = hour;
            newminute = minute;
            preferencesToShare();
            result = hour + ":" + minute;
            if((hour < 10) && (minute > 10))
            {
                result = "0" + hour + ":" + minute;
            }
            else if((hour > 10) && (minute < 10))
            {
                result = hour + ":" + "0" + minute;
            }
            else if((hour < 10) && (minute < 10))
            {
                result = "0" + hour + ":" + "0" + minute;
            }
            else
            {
                result = hour + ":" + minute;
            }
            backToMain();
            onTimeSet(time_picker, hour, minute);
            soundNotification(result);
        }
```

```
        });

        // Create onClick to stop the alarm or undo an alarm set
        cancel_button.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                onBackPressed();
            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu)
    {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.clock_menu, menu);
        return true;
    }

    @Override
    public void onItemSelected(AdapterView<?> parent, View view, int position, long id)
    {
        // An item was selected. You can retrieve the selected item using
        // parent.getItemAtPosition(pos) outputing whatever id the user has selected
        choose_droid_sound = (int) id;
        Log.i("ID ", String.valueOf(choose_droid_sound));
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent)
    {
        // Another interface callback
    }

    public void iniTiaLiSations()
    {
        this.context = this;
        // Initialise our alarm manager
```

```
        alarm_manager = (AlarmManager) getSystemService(Context.ALARM_SERVICE);
        // Create an intent to the Alarm_Receiver class
        my_intent = new Intent(this, Alarm_Receiver.class);
        // Initialise our time picker
        time_picker = findViewById(R.id.timePicker);
        time_picker.setIs24HourView(true);
        // Create the spinner in the main UI
        spinner = findViewById(R.id.spiner);
        // Create an ArrayAdapter using the string array and a default spinner layout
        ArrayAdapter<String> adapter = new ArrayAdapter<>(this,
                R.layout.spiner_item,sounds);
        // Specify the layout to use when the list of choises appears
        adapter.setDropDownViewResource(R.layout.spiner_item);
        // Apply the adapter to the spinner
        spinner.setAdapter(adapter);
        //create an onClick Listener to the Onitemselected method
        spinner.setOnItemSelectedListener(this);
        // Initialise start button
        save_button = findViewById(R.id.save);
        // Initialise stop button
        cancel_button = findViewById(R.id.cancel);
        // Initialise switch button intent switch_intent = getIntent();
        // Initialise our shares preferences
        sharedpreferences = getSharedPreferences("myPrefs", Context.MODE_PRIVATE);
        // Initialise the intent for the saved info
        info = getIntent();
        // Initialise the intent for fullscreen
        fullscreen_intent = getIntent();
    }

    @Override
    public void onBackPressed()
    {
        super.onBackPressed();
    }
    // Ok button
    public void onTimeSet(TimePicker view, int hourOfDay, int minute)
    {
        Calendar c = Calendar.getInstance();
        c.set(Calendar.HOUR_OF_DAY,hourOfDay);
```

```
        c.set(Calendar.MINUTE,minute);
        c.set(Calendar.SECOND,0);
        startAlarm(c);
    }

    public void startAlarm(Calendar c)
    {
        my_intent.putExtra("extra", "alarm on");
        my_intent.putExtra("Droid_choice", choose_droid_sound);
        Log.i("choose_droid_sound", String.valueOf(choose_droid_sound));
        pending_intent = PendingIntent.getBroadcast(Clock_Activity.this,
            0, my_intent, PendingIntent.FLAG_UPDATE_CURRENT);
        String toastText =
DateFormat.getTimeInstance(DateFormat.SHORT).format(c.getTime());
        if (c.before(Calendar.getInstance())) {
          c.add(Calendar.DATE, 1);
          Toast.makeText(Clock_Activity.this, "Alarm set on: " + toastText + " Next Day",
Toast.LENGTH_LONG).show();
        } else {
          Toast.makeText(Clock_Activity.this, "Alarm set on: " + toastText,
Toast.LENGTH_LONG).show();
        }
    }

    public void cancelAlarm()
    {
        pending_intent = PendingIntent.getBroadcast(Clock_Activity.this,
            0, my_intent, PendingIntent.FLAG_CANCEL_CURRENT);
        alarm_manager.cancel(pending_intent);
        my_intent.putExtra("extra", "alarm off");
    }

    public void backToMain()
    {
      // Initialise intent that is going to Main Activity
      Intent intent_save = new Intent(Clock_Activity.this, MainActivity.class);
      intent_save.putExtra("keyList",showList);
      intent_save.putExtra("time",result);
      startActivity(intent_save);
    }
```

```
    public void preferencesToShare()
    {
        SharedPreferences.Editor editor = sharedpreferences.edit();
        editor.putInt("message1", newhour);
        editor.putInt("message2",newminute);
        editor.putInt("sound", choose_droid_sound);
        editor.apply();
    }
    @RequiresApi(api = Build.VERSION_CODES.O)
    public void createChannelNotification()
    {
        // Notifications
        //Create the channel object with the unique ID SOUNDNOT_CHANNEL_ID
        NotificationChannel followersChannel = new
NotificationChannel(SOUNDNOT_CHANNEL_ID,"SoundNotify"
                ,NotificationManager.IMPORTANCE_DEFAULT);
        //Configure the channel's initial settings
        followersChannel.setLightColor(Color.GREEN);
        //Submit the notification channel object to the notification manager
        NotificationManager nm =
(NotificationManager)getSystemService(Context.NOTIFICATION_SERVICE);
        assert nm != null;
        nm.createNotificationChannel(followersChannel);
    }

    public void soundNotification(String res)
    {
        Intent notificationIntent = new Intent(this, MainActivity.class);
        PendingIntent contentIntent = PendingIntent.getActivity(this, 0, notificationIntent,
PendingIntent.FLAG_UPDATE_CURRENT);

        NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(this,
SOUNDNOT_CHANNEL_ID)
                .setSmallIcon(R.drawable.time_fragment)
                .setContentTitle("Alarm Reminder")
                .setContentText("Alarm set on " + res)
                .setPriority(NotificationCompat.PRIORITY_DEFAULT)
                .setContentIntent(contentIntent) // Set the intent that will fire when the user taps the
notification
```

```
                .setAutoCancel(false); //When it pressed, Notification disappears
        //Show the notification
        notifiticationManager = NotificationManagerCompat.from(this);
        notifiticationManager.notify(1, mBuilder.build());
    }

    public void toMain()
    {
        Intent intent = new Intent(this, MainActivity.class);
        int x = 1;
        newhour = sharedpreferences.getInt("message1", 0);
        newminute = sharedpreferences.getInt("message2",0);
        String result = newhour + ":" + newminute;
        if(newminute < 10)
        {
            result = newhour + ":" + "0" + newminute;
        }
        intent.putExtra("keyList",showList);
        intent.putExtra("time",result);
        intent.putExtra("number", x);
        notifiticationManager = NotificationManagerCompat.from(this);
        notifiticationManager.cancel(1);
        startActivity(intent);
    }
}

public class FullScreenDialog extends DialogFragment
{
    @Nullable
    @Override
    public View onCreateView(final LayoutInflater inflater, @Nullable final ViewGroup container,
final Bundle savedInstanceState)
    {
        super.onCreateView(inflater, container, savedInstanceState);
        int time_hour = getArguments().getInt("time_extra");
        int time_minute = getArguments().getInt("time_extra_2");
        String hour_and_minute = time_hour + ":" + time_minute;
        if((time_hour < 10) && (time_minute < 10))
        {
            hour_and_minute = "0" + time_hour + ":" + "0" + time_minute;
```

```
        }
        else if((time_hour < 10) && (time_minute > 10))
        {
            hour_and_minute = "0" + time_hour + ":" + time_minute;
        }
        else if((time_hour > 10) && (time_minute < 10))
        {
            hour_and_minute = time_hour + ":" + "0" + time_minute;
        }
        else
        {
            hour_and_minute = time_hour + ":" + time_minute;
        }
        TextView textview_time_from_clock = view.findViewById(R.id.textView_time);
        textview_time_from_clock.setText(String.valueOf(hour_and_minute));
        Button button_dis = view.findViewById(R.id.button_dismiss);
        button_dis.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                dialog.dismiss();
                ((Clock_Activity)getActivity()).toMain();
                ((Clock_Activity)getActivity()).cancelAlarm();
                ((Clock_Activity)getActivity()).finish();
            }
        });
        return view;
    }

    @Override
    public void onStart()
    {
        super.onStart();
        Dialog dialog = getDialog();
        if(dialog!=null)
        {
            int width = ViewGroup.LayoutParams.MATCH_PARENT;
            int height = ViewGroup.LayoutParams.MATCH_PARENT;
            dialog.getWindow().setLayout(width,height);
```

```
        }
    }

    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setStyle(DialogFragment.STYLE_NORMAL,R.style.FullScreenDialogStyle);
    }
}

public class MainActivity extends AppCompatActivity
{
    @Override
    public void onBackPressed()
    {
        //do nothing
        useHomeButton();
    }

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        Initialisations();
        //get a String value that control our adapter
        Intent intent = getIntent();
        Bundle b = intent.getExtras();
        if(b!= null)
        {
            get_time = b.getString("time");
        }
        int num = intent.getIntExtra("number", 0);
        Log.i("Number", String.valueOf(num));
        //get a boolean value that controls our adapter
        showthelist = intent.getBooleanExtra("keyList",false);
```

```
        if(showthelist)
        {
            //view our custom adapter
            listView.setAdapter(myAdapter);
        }
        listView.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener()
        {
            @Override
            public boolean onItemLongClick(AdapterView<?> parent, View view, final int position,
long id)
            {
                AlertDialog.Builder dialog = new AlertDialog.Builder(MainActivity.this);
                dialog.setMessage("Είστε σίγουρος πως θέλετε να σβήσετε αυτή την αφύπνιση;");
                dialog.setTitle("ΔΙΑΓΡΑΦΗ ΑΦΥΠΝΙΣΗΣ");
                dialog.setCancelable(false);
                dialog.setIcon(R.drawable.recyclebin);
                dialog.setPositiveButton("NAI", new DialogInterface.OnClickListener()
                {
                    @Override
                    public void onClick(DialogInterface dialog, int which)
                    {
                        Intent intent_alarm_off = new Intent(MainActivity.this, Clock_Activity.class);
                        intent_alarm_off.putExtra("alarm_is_off", alarm_is_off);
                        startActivity(intent_alarm_off);
                    }
                });
                dialog.setNegativeButton("OXI",
                        new DialogInterface.OnClickListener()
                        {
                            @Override
                            public void onClick(DialogInterface dialog, int which)
                            {
                                dialog.dismiss();
                            }
                        });
                dialog.show();
                return true;
            }
        });
```

```
        listView.setOnItemClickListener(new AdapterView.OnItemClickListener()
        {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id)
            {
                if(id == 0)
                {
                    Intent intent_listview = new Intent(MainActivity.this, Clock_Activity.class);
                    intent_listview.putExtra("svd_info", saved_info);
                    startActivity(intent_listview);
                }
            }
        });
        actionButton.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                Intent intent_button = new Intent(MainActivity.this, Clock_Activity.class);
                startActivity(intent_button);
            }
        });
    }

    public void Initialisations()
    {
        // Initialise our floatingActionButton
        actionButton = findViewById(R.id.floatingActionButton);
        // Initialise the ListView
        listView = findViewById(R.id.listview);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu)
    {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }
```

```
    @Override
    public boolean onOptionsItemSelected(MenuItem item)
    {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();
        return super.onOptionsItemSelected(item);
    }

    public ArrayList<Model> populateList(int x)
    {
        ArrayList<Model> list = new ArrayList<>();
        Model model = new Model();
        model.setTime(get_time);
        switch (x)
        {
            case 0:
                model.setSwitch_btn(true);
                break;
            case 1:
                model.setSwitch_btn(false);
                break;
            default:
                model.setSwitch_btn(true);
                break;
        }
        list.add(model);
        return list;
    }

    public void useHomeButton()
    {
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setMessage("Παρακαλώ πατήστε το Home Button για να εξέλθετε!!!")
                .setTitle("Μήνυμα Εξόδου")
                .setCancelable(false)
                .setIcon(R.drawable.ic_home_black_24dp)
                .setPositiveButton("ΤΟ ΚΑΤΑΛΑΒΑ", new DialogInterface.OnClickListener()
                {
```

```
                public void onClick(DialogInterface dialog, int id)
                {
                    //do things
                }
            });
    }
}

public class Model
{
    private String time;
    private boolean switch_btn;

    public void setTime(String set_time)
    {
        this.time = set_time;
    }

    public String getTime()
    {
        return time;
    }

    public void setSwitch_btn(boolean set_switch_btn)
    {
        this.switch_btn = set_switch_btn;
    }

    public boolean getSwitch_btn()
    {
        return switch_btn;
    }
}

public class MyAdapter extends BaseAdapter
{
    public void remove(int position)
    {
        ModelArrayList.remove(position);
        notifyDataSetChanged();
```

```
    }

    public void updateList(String time,int num)
    {
        Model model = new Model();
        model.setTime(time);
        switch (num) {
            case 0:
                model.setSwitch_btn(true);
                break;
            case 1:
                model.setSwitch_btn(false);
                break;
            default:
                model.setSwitch_btn(true);
                break;
        }
        ModelArrayList.add(model);
        this.notifyDataSetChanged();
    }

    public int getViewTypeCount()
    {
        return 1;
    }

    public int getItemViewType(int position)
    {
        return position;
    }

    public int getCount()
    {
        return ModelArrayList.size();
    }

    public Object getItem(int position)
    {
        return ModelArrayList.get(position);
    }
```

```
    public long getItemId(int position)
    {
        return 0;
    }

    public View getView(final int position, View convertView, @NonNull final ViewGroup parent)
    {
        if(convertView == null)
        {
            LayoutInflater layoutInflater = (LayoutInflater)
contexts.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            convertView = layoutInflater.inflate(R.layout.listview_item,null,true);
            mViewHolder.mtime = convertView.findViewById(R.id.textView1);
            mViewHolder.mswitch = convertView.findViewById(R.id.switch1);
            convertView.setTag(mViewHolder);
        }
        mViewHolder.mtime.setText(ModelArrayList.get(position).getTime());
        final Intent disabled_intent = new Intent(contexts, Clock_Activity.class);
        final Intent enable_intent = new Intent(contexts,Clock_Activity.class);
        // Switch button - Turn on/Turn off
        mViewHolder.mswitch.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                if(mViewHolder.mswitch.isChecked())
                {
                    enable_intent.putExtra("enable",en);
                    contexts.startActivity(enable_intent);
                }
                else
                {
                    disabled_intent.putExtra("disable", dis);
                    contexts.startActivity(disabled_intent);
                }
            }
        });
        return convertView;
    }
```

```
    static class ViewHolder
    {
        TextView mtime;
        Switch mswitch;
    }
}

public class RingtonePlayingService extends Service
{
    @TargetApi(Build.VERSION_CODES.JELLY_BEAN)
    @Override
    public IBinder onBind(Intent intent)
    {
        return null;
    }
    @RequiresApi(api = Build.VERSION_CODES.O)
    @Override
    public int onStartCommand(Intent intent, int flags, int startID)
    {
        // Fetch the extra string from the alarm on/alarm off values
        state = intent.getExtras().getString("extra");
        i("Ringtone state: extrais", state);
        // Fetch the Droid_choice integer values
        Integer Droid_sound_choice = intent.getExtras().getInt("Droid_choice");
        i("Droid_choice is ", Droid_sound_choice.toString());
        // if else statements
        // If there is no music playing and user presses "alarm on" music should start playing
        if (!this.isRunning && startID == 1) {
            i("There is no music, ", "and you want to START");
            startActivity(clock_activity);
            // Play the Droid sound depending on the passed Droid_choice id
            if (Droid_sound_choice == 0)
            {
                // Create an instance of the media player
                media_song = MediaPlayer.create(this, R.raw.alarmclock);
                // Start the ringtone
                media_song.start();
            }
            else if (Droid_sound_choice == 1)
            {
```

```
            media_song = MediaPlayer.create(this, R.raw.catscream);
            media_song.start();
        }
        else if (Droid_sound_choice == 2)
        {
            media_song = MediaPlayer.create(this, R.raw.chicken);
            media_song.start();
        }
        else if (Droid_sound_choice == 3)
        {
            media_song = MediaPlayer.create(this, R.raw.cowmoo);
            media_song.start();
        }
        else if (Droid_sound_choice == 4)
        {
            media_song = MediaPlayer.create(this, R.raw.dixiehorn);
            media_song.start();
        }
        else if (Droid_sound_choice == 5)
        {
            media_song = MediaPlayer.create(this, R.raw.funnyvoices);
            media_song.start();
        }
        else if (Droid_sound_choice == 6)
        {
            media_song = MediaPlayer.create(this, R.raw.glasswater);
            media_song.start();
        }
        else if (Droid_sound_choice == 7)
        {
            media_song = MediaPlayer.create(this, R.raw.hahaha);
            media_song.start();
        }
        else if (Droid_sound_choice == 8)
        {
            media_song = MediaPlayer.create(this, R.raw.heavyrain);
            media_song.start();
        }
        else if (Droid_sound_choice == 9)
        {
```

```
            media_song = MediaPlayer.create(this, R.raw.henchicken);
            media_song.start();
        }
        else if (Droid_sound_choice == 10)
        {
            media_song = MediaPlayer.create(this, R.raw.jollylaugh);
            media_song.start();
        }
        else if (Droid_sound_choice == 11)
        {
            media_song = MediaPlayer.create(this, R.raw.newsintro);
            media_song.start();
        }
        else if (Droid_sound_choice == 12)
        {
            media_song = MediaPlayer.create(this, R.raw.pagerbeeps);
            media_song.start();
        }
        else if (Droid_sound_choice == 13)
        {
            media_song = MediaPlayer.create(this, R.raw.police);
            media_song.start();
        }
        else if (Droid_sound_choice == 14)
        {
            media_song = MediaPlayer.create(this, R.raw.quack);
            media_song.start();
        }
        else if (Droid_sound_choice == 15)
        {
            media_song = MediaPlayer.create(this, R.raw.rooster);
            media_song.start();
        }
        else if (Droid_sound_choice == 16)
        {
            media_song = MediaPlayer.create(this, R.raw.scaryscream);
            media_song.start();
        }
        else if (Droid_sound_choice == 17)
        {
```

```
            media_song = MediaPlayer.create(this, R.raw.sillysnoring);
            media_song.start();
        }
        else if (Droid_sound_choice == 18)
        {
            media_song = MediaPlayer.create(this, R.raw.sleighbells);
            media_song.start();
        }
        else if (Droid_sound_choice == 19)
        {
            media_song = MediaPlayer.create(this, R.raw.tornadosiren);
            media_song.start();
        }
        else if (Droid_sound_choice == 20)
        {
            media_song = MediaPlayer.create(this, R.raw.wakeup);
            media_song.start();
        }
        else if (Droid_sound_choice == 21)
        {
            media_song = MediaPlayer.create(this, R.raw.weatheralert);
            media_song.start();
        }
        else
        {
            media_song = MediaPlayer.create(this, R.raw.sillysnoring);
            media_song.start();
        }
    }
    // If there is music playing and user presses "alarm off" music should stop playing
    else if (this.isRunning && startID == 0) {
        i("There is music, ", "and you want to STOP");
        // Stop the ringtone
        media_song.stop();
        media_song.reset();
        this.isRunning = false;
        this.startID = 0;
    }
    // These are if the user presses random buttons just to bug-proof the app
    // If there is no music playing and user presses "alarm off" -> do nothing
```

```
        else if (!this.isRunning && startID == 0) {
            i("There is no music, ", "and you want to STOP");
            this.isRunning = false;
            this.startID = 0;
        }
        // If there is music playing and user presses "alarm on" -> do nothing
        else if (this.isRunning && startID == 1) {
            i("There is music, ", "and you want to START");
            this.isRunning = true;
            this.startID = 1;
        }
        // Can't think of anything else, just to catch the odd event
        else {
            i("Else ", "somehow you reached this");
        }
        return START_NOT_STICKY;
    }
    private LayoutInflater getLayoutInflater()
    {
        return null;
    }

    @Override
    public void onDestroy()
    {
        // Tell the user we stopped
        i("on Destroy called", "Crashed");
        super.onDestroy();
        this.isRunning = false;
    }
}
```