



ΤΕΙ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

## ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη Android Εφαρμογής για πρόβλεψη τιμών μετοχών

ΣΠΟΥΔΑΣΤΗΣ:

ΔΡΑΚΩΝΑΚΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ

ΠΑΠΑΔΑΚΗΣ ΝΙΚΟΛΑΟΣ

Κρήτη 2014

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

## Περίληψη

Στην παρούσα πτυχιακή εργασία περιγράφονται τεχνικές δημιουργίας εφαρμογών για smartphones με λειτουργικό σύστημα Android με χρήση της εφαρμογής Android Studio και χρησιμοποιώντας τα Android API's (Application Programming Interface) και SDK (System Development Kit). Αρχικά θα δούμε γιατί επιλέξαμε το συγκεκριμένο λειτουργικό σύστημα και αργότερα θα δούμε τα ενδότερα στοιχεία του.

Ακολουθεί μια ιστορική αναδρομή σχετικά με τις εκδόσεις και τα χαρακτηριστικά του Android και στην συνέχεια προχωράμε στις παραμέτρους του χρησιμοποιώντας την αντικειμενοστραφής γλώσσα προγραμματισμού JAVA. Τέλος μετα απο την μελέτη όλων αυτών και χρησιμοποιώντας τα κατάλληλα εργαλεία καταφέραμε να φτιάξουμε την παρακάτω εφαρμογή.

Στόχος της εφαρμογής είναι να φορτώνει τις μετοχές καθώς και τις τιμές τους απο το διαδύκτιο και να κάνει μια πρόβλεψη για την μελλοντική τιμή υλοποιώντας ενα αλγόριθμο. Ο χρήστης μπορεί να μένει ενημερωμένος για τις 15 τελευταίες τιμές και να βλέπει ένα ζωντανό γράφημα για το πώς μεταβλήθηκε η τιμή της κάθε μετοχής το τελευταίο διάστημα.

## Abstract

The present paper aims to study, develop and learn how to create application for Android based smartphones. For the purposes of the paper an android application has developed using the platform Android Studio. For start we will see the history of Android, how it started and characteristics. Later we will move to the deeper parts and we will use JAVA programming language which is Object Oriented. Furthermore after all the studying and experiences we had during studying the characteristics we will create our application.

The Application aims to load the stocks and stock prices from a web feed and will try to predict the future price. The application keeps record of the last 15 days and the user will be able to predict the future price with a push of a button and also view a live graph on how the stock price was changed during those 15 days.

## Περιεχόμενα

Περίληψη .....	3
Abstract .....	4
Κεφάλαιο 1.....	7
1.1    Τι είναι το Android .....	7
1.2    Χαρακτηριστικά του Android .....	8
1.3    Γιατι χρησιμοποιούμε Android .....	9
1.4    Εξέλιξη του Android .....	10
1.4.1    Android Cupcake (API Level 3 Android Version 1.5) .....	11
1.4.2    Android Donut (API Level 4 Android Version 1.6).....	12
1.4.3    Android Éclair (API Level 5 & 6 & 7 Android Versions 2.0 & 2.1) .....	12
1.4.4    Android Froyo (API Level 8 Android Version 2.2).....	13
1.4.5    Android Gingerbread (API Level 9 & 10 Android Version 2.3.x) .....	14
1.4.6    Android Honeycomb (API Level 11 & 12 & 13 Android Versions 3.0 & 3.1-3.2.6).....	15
1.4.7    Android Ice-cream Sandwich (API Level 14 & 15 Android Version 4.0.x) .....	16
1.4.8    Android Jellybean (API Level 16 & 17 & 18 Android Versions 4.1 & 4.2 & 4.3) .....	17
1.4.9    Android KitKat (API Level 19 & 20 Android Version 4.4).....	18
1.4.10    Android Lollipop (API Level 21 Android Version 5.0) .....	19
1.5    Αρχιτεκτονική του Android .....	20
1.5.1    Πυρήνας (Linux Based Kernel) .....	20
1.5.2    Η Εικονική μηχανή Dalvik.....	21
1.5.3    Πλαίσιο Εφαρμογής Android .....	22
1.6    Εσωτερικό Εφαρμογής Android .....	26
1.6.1    Φάκελος src.....	26
1.6.2    Φάκελος res .....	27
1.6.3    AndroidManifest.xml .....	28
1.6.4    Gradle Scripts .....	29
1.7    Δομικά μέρη μιας εφαρμογής .....	30
1.7.1    Activity.....	30
1.7.2    Services .....	30
1.7.3    Intents .....	30
1.7.4    Broadcast Receivers .....	31
Κεφάλαιο 2.....	31
2.1    Εγκατάσταση Λογισμικού .....	31

2.2	Διαδικασία Αποσφαλμάτωσης(Debugging).....	31
2.3	Δημοσίευση Εφαρμογής.....	32
2.4	Android System Development Kit.....	32
2.5	Android Virtual Device Manager(AVD) .....	33
2.6	Android Design Guidelines.....	35
2.7	Android Logcat – Android Crash Report Manager.....	37
2.8	Notifications.....	39
2.9	Developing In Android from Programmer View .....	43
Κεφάλαιο 3.....		44
3.1	Ανάλυση απαιτήσεων για την δημιουργία της εφαρμογής .....	44
3.1.1	Δημιουργία Βάσης Δεδομένων.....	44
3.1.2	PHP Script for exporting Data in JSON Array .....	47
3.1.3	Exporting Data in JSON Array into the browser.....	47
3.2	Δημιουργία Project στο Android Studio .....	48
3.3	Δημιουργία εμφανισιακού περιεχομένου της πρώτης οθόνης .....	49
3.4	Δημιουργία κώδικα πρώτης οθόνης.....	51
3.5	Δημιουργία εμφανισιακού περιεχομένου δευτερης οθόνης.....	58
3.6	Δημιουργία κώδικα της δεύτερης οθόνης.....	59
3.7	Δημιουργία Animation εναλλαγή οθονών.....	62
3.8	Τελικό Αποτέλεσμα.....	63
Κεφάλαιο 4.....		64
4.1	Μελλοντική Εξέλιξη .....	64
4.2	Συμπεράσματα.....	64
Κεφάλαιο 5.....		65
Βιβλιογραφία .....		65

## Κεφάλαιο 1

### 1.1 Τι είναι το Android

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



Εικόνα 1: Android Λογότυπο

## 1.2 Χαρακτηριστικά του Android

Τα χαρακτηριστικά του είναι:

- Η πλατφόρμα είναι προσαρμόσιμη σε μεγαλύτερες αναλύσεις (VGA, QVGA).
- Χρήση βάσης δεδομένων SQLite.
- Τεχνολογίες συνδεσημοτητας GSM/CDMA, EDGE, EV-DO, UMTS, Bluetooth, Wi-Fi.
- Διαθέσιμους τρόπους ανταλλαγής μηνυμάτων SMS και MMS.
- Διαθέτει web browser βασισμένο στην τεχνολογία WebKit.
- Λογισμικό γραμμένο σε JAVA το οποίο γίνεται compile στην εικονική μηχανή Dalvik και ART.
- Υποστηρίζει τις ακόλουθες επεκτάσεις στατικής κινούμενης εικόνας [H.263](#), [H.264](#), [3GP](#), [MP4](#), [AMR](#), [AMR-WEB](#), [JPEG](#), [PNG](#), [GIF](#), [BMP](#).
- Αρχεία ήχου [AAC](#), [MP3](#), [MIDI](#), [OGG](#), [WAV](#).
- Χρησιμοποιεί ψηφιακές βιβλιοθήκες και τρισδιάστατα γραφικά χρησιμοποιώντας την τεχνολογία OpenGL ES 1.0 .
- Μπορεί να διαθέτει γυροσκόπιο, κάμερα, GPS, αισθητήρες επιτάχυνσης, μαγνητόμετρο, δισδιάστατους και τρισδιάστατους επιταχυντές γραφικών.



- Υποστηρίζει την τεχνολογία multi-touch

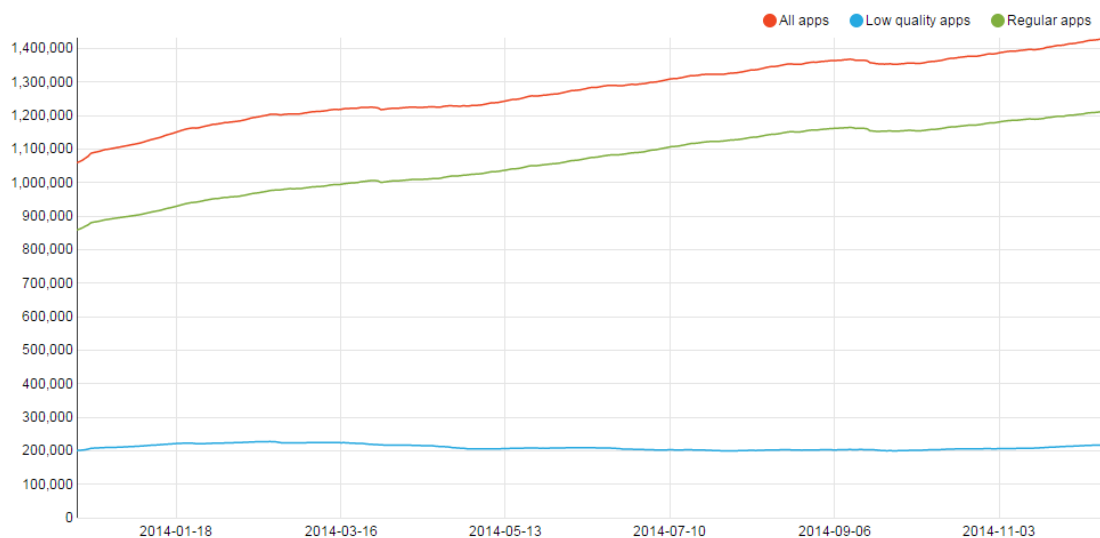
### 1.3 Γιατι χρησιμοποιούμε Android

Το Android είναι ένα λογισμικό ιδιαίτερος γνωστό και ευρύτερα διαδεδομένο βασικότερα επειδή διαθέτει μια μεγάλη κοινότητα προγραμματιστών και επειδή είναι ανοιχτού κώδικα(open source). Οι εφαρμογές γράφονται σε μια προσαρμοσμένη έκδοση της JAVA χρησιμοποιώντας τα Android API's τα οποία είναι διαθέσιμα μαζί με τις πλατφόρμες [Android Studio](#) και το plugin [ADT](#) για το Eclipse.

Διαθέτει πάνω απο ενα εκατομύριο εφαρμογές στο Google Play Store και για την ακρίβεια 1.432.671

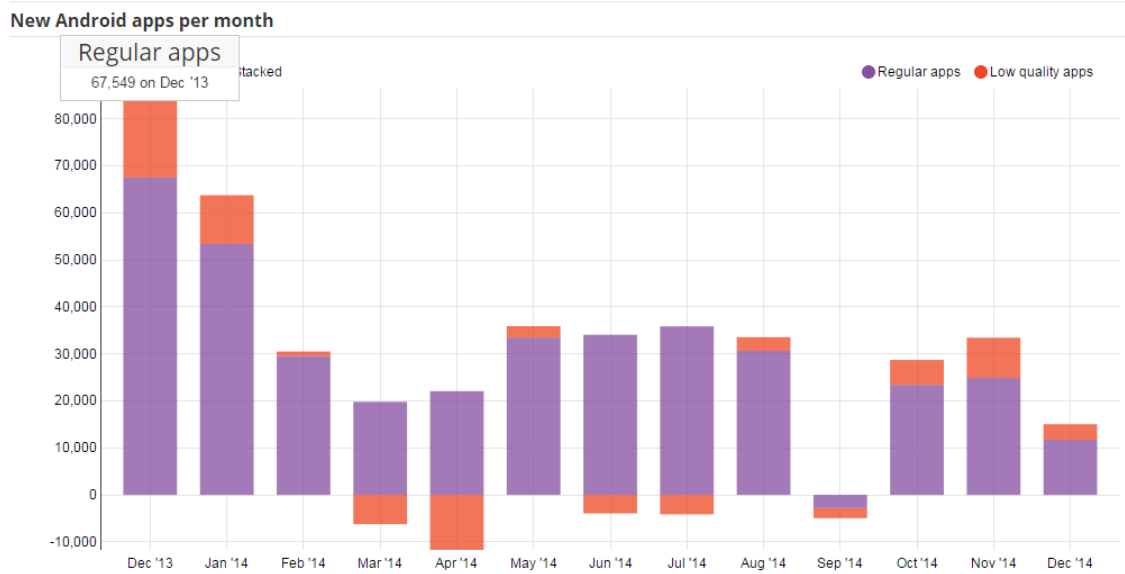
Παρακάτω θα δούμε πως μεταβλήθηκαν τα νούμερα αυτά με τα χρόνια:

Android apps on Google Play



Εικόνα 2: Διάγραμμα 1

Και εδώ θα δούμε την αυξομείωση των εφαρμογών σε σχέση με την ποιότητα τους:



Εικόνα 3: Διάγραμμα 2

### 1.4 Εξέλιξη του Android

Η εξέλιξη του Android ήταν πολύ μεγάλη και αναμενόμενη λόγω της διαθεσιμότητας του στο ευρύ κοινό χωρίς πληρωμή. Αυτό φαίνεται στο ότι οι 9 κύριες εκδόσεις του έχουν κυκλοφορήσει σε διάστημα 5 ετών από τον Σεπτέμβριο του 2009 με την πρώτη έκδοση Android (Cupcake).

Version ↕	Code name ↕	Release date ↕	API level ↕	Distribution <sup>[a]</sup> ↕
4.4	<i>KitKat</i>	October 31, 2013	19	33.9%
4.3	<i>Jelly Bean</i>	July 24, 2013	18	7.0%
4.2.x		November 13, 2012	17	20.4%
4.1.x		July 9, 2012	16	21.3%
4.0.3–4.0.4	<i>Ice Cream Sandwich</i>	December 16, 2011	15	7.8%
2.3.3–2.3.7	<i>Gingerbread</i>	February 9, 2011	10	9.1%
2.2	<i>Froyo</i>	May 20, 2010	8	0.5%

Εικόνα 4: Εξέλιξη του Android

#### 1.4.1 Android Cupcake (API Level 3 Android Version 1.5)



Εικόνα 5: Cupcake

Τον Μάιο του 2009 ήρθε η έκδοση Cupcake με την εισαγωγή νέων χαρακτηριστικών όπως:

- Παγκόσμια αναζήτηση
- Αναδιάταξη και Επανασχεδιασμός του τότε Android Market με κατηγορίες όπως (Εφαρμογές, Λήψεις, Παιχνίδια) και φίλτρα (Κορυφαίες δωρεάν, Κορυφαίες επι πληρωμή)

- Επέκταση του SDK για να υποστηρίζει κινήσεις με το χέρι(gestures) και ομιλία σε κείμενο(voice to text)

#### 1.4.2 Android Donut (API Level 4 Android Version 1.6)



Εικόνα 6: Android Donut

Τον Σεπτέμβριο του 2009 ήρθε το Android Donut το οποίο πρόσθεσε τα εξής χαρακτηριστικά στην διαθεσιμότητα του κοινού:

- Εικονικό πληκτρολόγιο οθόνης
- Camcorder mode για εγγραφή και προβολή video
- Stereo Bluetooth
- Εισαγωγή widgets και φακέλων στην αρχική οθόνη
- Αντιγραφή/Επικόλληση και αναζήτηση στον browser
- Άμεση μεταφόρτωση στο YouTube και στο Picasa

#### 1.4.3 Android Éclair (API Level 5 & 6 & 7 Android Versions 2.0 & 2.1)



Εικόνα 7: Android Éclair

Τον Οκτώμβριο του 2009 έκανε την εμφάνιση του το Android Éclair με αλλαγή Level στο 2.0 φέρνοντας αλλαγές όπως:

- Πολλαπλούς λογαριασμούς χρήστη
- Υποστήριξη λήψης και αποστολής email
- Ανεπτυγμένη κάμερα με χρήση flash και ψηφιακό zoom
- Bluetooth 2.1
- Ανεπτυγμένο πληκτρολόγιο με προσαρμόσιμο λεξικό και προτάσεις απο ονόματα επαφών
- Live wallpapers
- Ομιλία σε κείμενο(Speech to text) σε οποιοδήποτε πεδίο κειμένου

#### 1.4.4 Android Froyo (API Level 8 Android Version 2.2)



Εικόνα 8: Android Froyo

Τον Μάϊο του 2010 έκανε την εμφάνιση του το Android Frogo εισάγωντας νέες λειτουργίες στην κοινότητα όπως:

- Υποστήριξη Tethering και λειτουργίας Hotspot
- Αναβάθμιση του Android Market εισάγωντας την δυνατότητα εγκατάστασης εφαρμογών σε εξωτερική κάρτα SD
- Adobe Flash 10.1
- Ανέβασμα αρχείων σε πόρους του browser
- Ανεπτυγμένη υποστήριξη Microsoft Exchange: Πολιτικές Ασφαλείας, Global Διευθύνσεις, συγχρονισμός ημερολογίου, απομακρυσμένη διαγραφή και καθαρισμός
- Υποστήριξη Bluetooth για φωνητική πληκτρολόγηση και διαμοιρασμό επαφών(contact sharing)

#### 1.4.5 Android Gingerbread (API Level 9 & 10 Android Version 2.3.x)



Εικόνα 9: Android Gingerbread

Τον Δεκέμβριο του 2010 ήρθε έκανε την εμφάνιση του το Android Gingerbread φέρνοντας σημαντικές αλλαγές στην κοινότητα του Android όπως:

- Επανασχεδιασμός του Copy/Paste
- Υποστήριξη για συμπίεση WebM video
- NFC(near field communication)
- Αλλαγή σε μπροστινή κάμερα μέσα απο την εφαρμογή της κάμερας
- Συντομέυσεις πληκτρολογίου οθόνης

#### 1.4.6 Android Honeycomb (API Level 11 & 12 & 13 Android Versions 3.0 & 3.1-3.2.6)



Εικόνα 10: Android Honeycomb

Τον φεβρουάριο του 2011 έκανε την εμφάνιση της μια ιδιαίτερη έκδοση που απευθυνόταν αποκλειστικά μόνο για tablets, ενσωματώνει ένα νέο σχεδιασμένο με απλούστερο User Interface και υποστήριξη διπύρηνων και τετραπύρηνων επεξεργαστών.

Επίσης έχουμε την ύπαρξη του κουμπιού recent apps έτσι ώστε ο χρήστης να περνάει εύκολα απο την μια εφαρμογή σε μια άλλη.

Οι καινούργιες αλλαγές που έφερε όμως δεν ήταν μόνο αυτές, έτσι βλέπουμε αλλαγές όπως:

- Υποστήριξη γραφικών 3D
- Private browsing
- Δυνατότητα Video Chat με την χρήση της υπηρεσίας Google Talk
- Δυνατότητα προβολής εικόνων απο την γκαλερί σε πλήρης οθόνη

- Bluetooth Tethering
- Υποστήριξη περιφερειακών συσκευών όπως πληκτρολόγια και game pads
- Παραμετροποιήσιμα widgets
- Δυνατότητα υποστήριξης της λειτουργίας πληρωμή στον δρόμο (Pay as you go) 3G/4G tablets
- Διορθώσεις σφαλμάτων και σταθερότητας

#### 1.4.7 Android Ice-cream Sandwich (API Level 14 & 15 Android Version 4.0.x)



Εικόνα 11: Android Ice-cream Sandwich

Με πάρα πολλές καινούργιες δυνατότητες κάνει την εμφάνιση του τον Οκτώβριο του 2011 το Android Ice-cream Sandwich.

Με επανασχεδιασμένο User interface πιο γρήγορο και αποδοτικό εισάγει νέα εικονικά κουμπιά που αντικαθιστούν τα αφής ή φυσικά που υπήρχαν στις εκάστοτε συσκευές.

Οι αλλαγές που έφερε είναι πολλές όπως θα δούμε παρακάτω:

- Διμιουργία φακέλων απλά τραβώντας την μια εφαρμογή επάνω σε μια άλλη
- Διαχωρισμός μεταξύ εφαρμογών και widgets
- Υποστήριξη λειτουργίας pinch-to-zoom στην εφαρμογή ημερολογίου
- Επανασχεδιασμός της εφαρμογής Gmail και παράλληλα προσθέτοντας την λειτουργία offline search καθώς επίσης την λειτουργία swipe μεταξύ email μηνυμάτων
- Επανασχεδιασμός της εφαρμογής Google Chrome η οποία πλέον υποστηρίζει καρτέλες εκτός σύνδεσης, μέχρι 16 συνεχόμενες καρτέλες ανοικτές, συγχρονισμός σελιδοδεικτών
- Διορθώσεις σφάλματων πληκτρολογίου οθόνης
- Προσαρμόσιμη οθόνη κλειδώματος, υποστήριξη launcher
- Εισαγωγή γραμματοσειράς Roboto
- Νέα λειτουργία διαγραφής με κύλιση
- Αναπτυγμένη αναγνώριση ομιλίας και λειτουργία copy/paste



- Ξεκλείδωμα οθόνης με λειτουργία αναγνώρισης προσώπου
- Μετρητής δεδομένων κινητής τηλεφωνίας
- Λειτουργία απόκρυψης επιλεγμένων εφαρμογών απο τον χρήστη
- Ικανότητα για τερματισμό λειτουργίας εφαρμογών που χρησιμοποιούν πόρους υλικού
- Προσθετες λειτουργίες της εφαρμογής κάμερας: zero shutter lag, continuous focus, zoom while recording, time lapse settings, panorama photos, 1080p recording
- Δυνατότητα αναγνώρισης προσώπου (Face recognition) μεσα απο την κάμερα με ενσωματωμένο Photo Editor
- Νέος σχεδιασμός του gallery app
- Δυνατότητα αποστολής άμεσου μηνύματος στην απόριψη κλήσεων
- Λειτουργία Android Beam μια ιδιότητα του NFC για αποστολή αρχείων οταν τα κινητά βρίσκονται σε κοντινή απόσταση

#### 1.4.8 Android Jellybean (API Level 16 & 17 & 18 Android Versions 4.1 & 4.2 & 4.3)



Εικόνα 12: Android Jellybean

Τον Ιούλιο του 2012 έκανε την εμφάνιση του το Android Jellybean φέρνοντας ραγδαίες αλλαγές όχι μόνο στον σχεδιασμό των εφαρμογών αλλά και σε ολόκληρο το λειτουργικό σύστημα. Οι αλλαγές είναι οι εξής:

- Πιο γρήγορη και ομαλή απόδοση με το λεγόμενο project Butter
- Επεκτάσιμες ειδοποιήσεις με μεγαλύτερη διαδραστικότητα
- Φωνητική αναζήτηση με κύλιση απο το κάτω μέρος της οθόνης
- Εισαγωγή της υπηρεσίας Google Now
- Λειτουργία Offline Dictation
- Επαναπροσαρμοσιμα widgets εφαρμογών
- Αναβάθμιση της λειτουργίας Android Beam στο να υποστηρίζει μεγαλύτερου όγκου αρχια όπως εικόνες και video
- Αναβάθμιση του Android Market σε Google Play Store
- Φωνητική αναζήτηση μουσικών κομματιών και καλιτεχνών
- Υψηλής ανάλυσης φωτογραφίες επαφών
- Επιλογές μεγαλύτερης προσβασιμότητας

- Επέκταση υποστήριξης γλωσσών σε Αραβικά και Εβραϊκά
- Interface Tweaks
- Δυνατότητας εισαγωγής widgets στην οθόνη κλειδώματος και άνοιγμα κάμερας απευθείας απο την οθόνη κλειδώματος
- Δυνατότητα εισαγωγής μπάρας γρήγορων ρυθμίσεων στο πάνελ ειδοποιήσεων για πιο εύκολη πρόσβαση στο Wi-Fi, Bluetooth, etc.
- Δυνατότητα λειτουργίας Daydream
- Wireless Display(Miracast)
- Λειτουργίες προσβασιμότητας όπως (τριπλό χτύπημα οθόνης για μεγένθυση, zoom με δυο δάκτυλα, φωνητική πλοήγηση για τυφλούς)
- Ενοποιημένο User Interface για όλες τις συσκευές
- Υποστήριξη Bluetooth Low Energy
- Εντοπισμός θέσης με την υποστήριξη Wi-Fi χωρίς την ενεργοποίηση του Wi-Fi
- Υποστήριξη οθονών με ανάλυση 4K

#### 1.4.9 Android KitKat (API Level 19 & 20 Android Version 4.4)



Εικόνα 13: Android KitKat

Εισάγωντας νέες αλλαγές στο design τον Οκτώβριο του 2013 έκανε την εμφάνισή του το Android KitKat όπως είναι οι παρακάτω:

- Μεγάλες αλλαγές στο design ειδικότερα για Nexus συσκευές
- Διαφανής μπάρα κατάστασης για το λειτουργικό σύστημα και για εφαρμογές

- Νέα λειτουργία immersive mode όπου οι εφαρμογές μπορούν να κρύβουν την πλοήγηση και την μπάρα κατάστασης
- Μείωση του μεγέθους του λειτουργικού συστήματος για γρηγορότερη λειτουργία και απόδοση σε συσκευές με μικρές ποσότητες μνήμης RAM
- Ασύρματη εκτύπωση με της υπηρεσίας Google Cloud Print

#### 1.4.10 Android Lollipop (API Level 21 Android Version 5.0)



Εικόνα 14: Android Lollipop

Πολλά υποσχόμενο και νέες αλλαγές πρόκειται να φέρει η νέα και πολυσηζητημένη έκδοση Android Lollipop. Πλήρως επανασχεδιασμένο Material Design και ανανεωμένο User interface έτσι ώστε να παρέχει την βέλτιστη απόδοση για χρήστες με μικρές ποσότητες μνήμης RAM. Δεν έχει κυκλοφορήσει ακόμα σε όλα τις συσκευές τις αγοράς παρα μόνο στα Nexus και μερικά μοντέλα της Samsung.

Οι αλλαγές που θα γίνουν είναι οι παρακάτω:

- Ειδοποιήσεις κατευθείαν στην οθόνη κλειδώματος και pop-up μηνυματάκια
- Λειτουργία priority mode η οποία εμφανίζει τις πιο σημαντικές ειδοποιήσεις
- Πολλαπλοί λογαριασμοί χρηστών για smartphones και tablets
- Νέο menu recent apps με ονομασία Overview
- Λειτουργία Guest mode

- Screen Pinning
- Battery Saver mode
- Προεπιλεγμένη κρυπτογράφηση συσκευής
- Έξυπνο κλείδωμα/ξεκλείδωμα ξεκλειδώνει συσκευές Bluetooth ή NFC

## 1.5 Αρχιτεκτονική του Android

Το Android δεν είναι απλά ένα λειτουργικό σύστημα αλλά είναι μια στοίβα απο διεργασίες και υπηρεσίες που επικοινωνούν με τις εφαρμογές γνωστές και ως middleware, τις κύριες εφαρμογές γνωστές και ως core, λειτουργικό σύστημα.

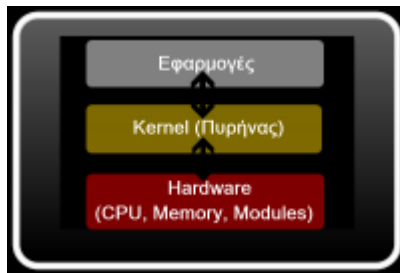
Στο παρακάτω γράφημα θα δούμε την αρχιτεκτονική αυτή:



Εικόνα 15: Αρχιτεκτονική του Android

### 1.5.1 Πυρήνας (Linux Based Kernel)

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



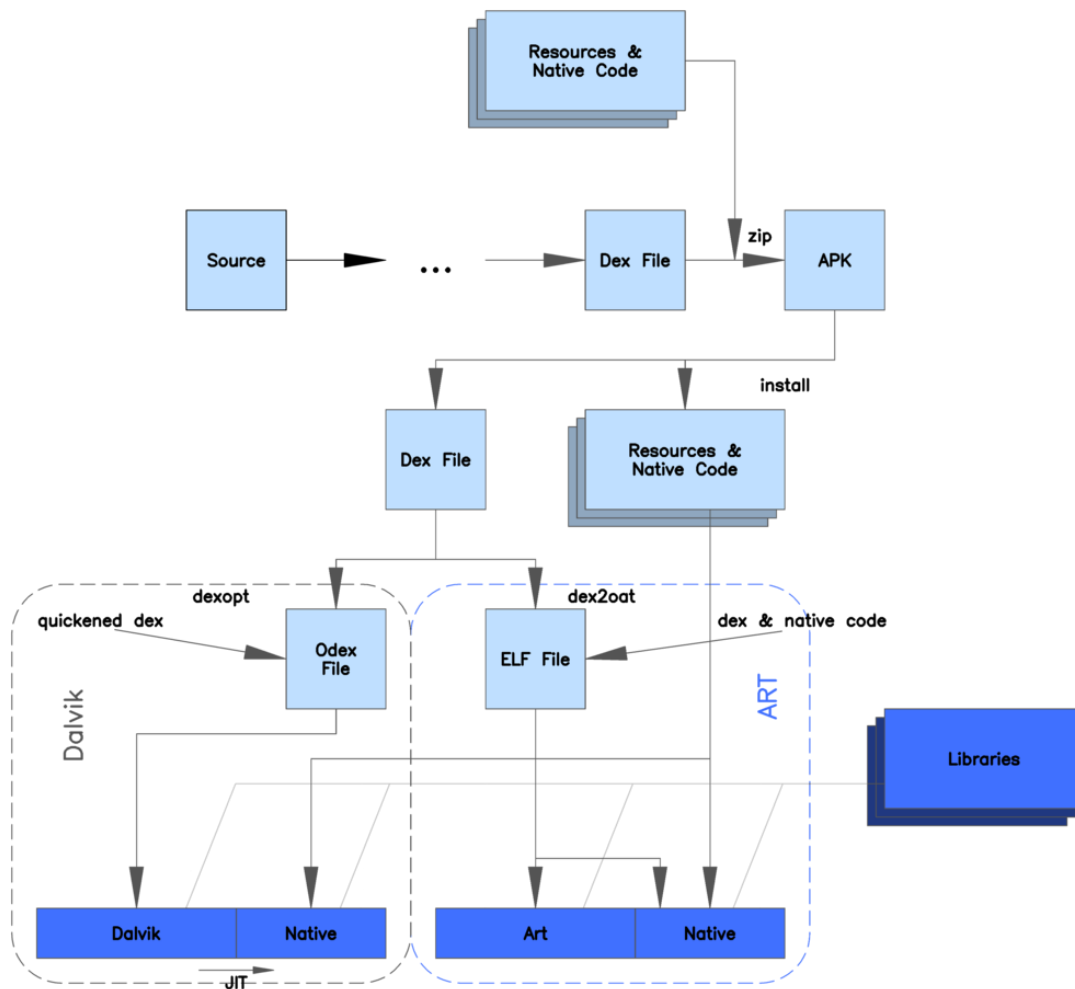
Εικόνα 16: Android Kernel

Η κατασκευάστρια εταιρία καθε συσκευής πρέπει να ενσωματώνει διαφορετικούς drivers για το hardware πράγμα το οποίο καθιστά το kernel της κάθε συσκευής διαφορετικό. Πολλοί προγραμματιστές αλλάζουν το kernel για πολλούς λόγους όπως είναι (bug fixing, overclocking, better RAM performance, e.t.c).

### 1.5.2 Η Εικονική μηχανή Dalvik

Η Dalvik είναι η εικονική μηχανή που χρησιμοποιεί η κάθε εφαρμογή για να τρέξει. Έτσι γίνεται το λεγόμενο multi-tasking χωρίς οι εικονικές μηχανές να επηρεάζονται μεταξύ τους και με αποτέλεσμα οι εφαρμογές να εκτελούνται ταυτόχρονα. Τα αρχεία της εφαρμογής πρέπει να είναι σε μορφή .dex διότι η Dalvik δεν μπορεί να κάνει compile σε κώδικα byte έτσι ώστε να τα τρέξει το Virtual Machine. Μια απο τις σημαντικότερες προσθήκες στην έκδοση 4.4.x ήταν η προσθήκη του Compiler ART πράγμα το οποίο πέρασε απαρατήρητο απο πολλούς χρήστες κατα το Runtime της συσκευής αλλα και της εφαρμογής. Αρχικά ήρθε σε πειραματικό στάδιο πράγμα το οποίο σήμαινε οτι για να το ενεργοποιήσει κάποιος χρήστης θα έπρεπε να ανοίξει την κρυφή λειτουργία Developer Options και να αλλάξει το Runtime απο Dalvik σε ART. Με την έκδοση 5.0 το ART έγινε πλέον το default runtime των συσκευών.

Παρακάτω θα δούμε μια σύγκριση του Dalvik και του ART Runtime.



Εικόνα 17: Σύγκριση του Dalvik και του ART Runtime.

### 1.5.3 Πλαίσιο Εφαρμογής Android

Το πλαίσιο εφαρμογής περιλαμβάνει τα εξής

- 1) View System
- 2) Content Providers
- 3) Resource Manager
- 4) Notification Manager
- 5) Activity Manager

Οι προγραμματιστές έχουν πρόσβαση σε όλα τα API's καθιστώντας έτσι την δημιουργία πολύπλοκων εφαρμογών με συγκεκριμένες άδειες δίνοντας πρόσβαση σε όλα τα μέρη του λειτουργικού συστήματος.

### 1.5.3.1 View System

Είναι όλα τα Γραφικά Περιβάλλοντα του Προγραμματιστή που θα χρησιμοποιηθούν για την δημιουργία της εφαρμογής. Μπορούν να υλοποιηθούν μέσω XML μέσω της εντολής `setContentView(int resourceId)` η οποία παίρνει όρισμα ένα id του αρχείου XML που υλοποιείται ή προγραμματιστικά μέσω

```
TextView tv = new TextView(this);  
tv.setLayoutParams(new ViewGroup.LayoutParams(  
    ViewGroup.LayoutParams.WRAP_CONTENT,  
    ViewGroup.LayoutParams.WRAP_CONTENT));
```

Στην οποία βλέπουμε πως υλοποιείται ένα πεδίο κειμένου(ετικέτας). Τα γραφικά περιβάλλοντα μπορούν να περιλαμβάνουν πολύπλοκα GUI's χρησιμοποιώντας πολλούς Layout Managers(Linear Layout, Relative Layout, Table Layout, Listview, Spinner View, Frame Layout, Buttons, Edittext, Textview, Checkbox, Switch, ToggleButton, Imageview, e.t.c.).

### 1.5.3.2 Content Providers

Οι content providers δίνουν την δυνατότητα στην εφαρμογή να έχει πρόσβαση σε δομημένα δεδομένα. Ενθυλακώνουν μηχανισμούς παραλαβής, αποστολής και ασφάλισης δεδομένων. Για να μπορεί να γίνει λήψη των δεδομένων χρειάζονται κάποιες συγκεκριμένες άδειες που επιτρέπουν στην εφαρμογή να κάνει συγκεκριμένα πράγματα. Για παράδειγμα αν θέλαμε να δημιουργήσουμε μια εφαρμογή ανταλλαγής μηνυμάτων κειμένου θα έπρεπε να βάλουμε το `<uses-feature>` στο Manifest.xml το `hasSystemFeature (PackageManager.FEATURE_TELEPHONY)`. Και τα permissions

- `<uses-permission android:name="android.permission.WRITE_SMS"/>`
- `<uses-permission android:name="android.permission.READ_SMS"/>`
- `<uses-permission android:name="android.permission.SEND_SMS"/>`
- `<uses-permission android:name="android.permission.RECEIVE_SMS"/>`
- `<uses-permission android:name="android.permission.BROADCAST_SMS"/>`
- `<permission android:name="android.providers.Telephony"/>`
- `<permission android:name="android.providers.Telephony.SMS"/>`

### 1.5.3.3 Resource Manager

Δίνει πρόσβαση στην εφαρμογή απο εξωτερικούς παράγοντες όπως είναι(αρχεία, εικόνες απο την μνήμη(εσωτερική ή εξωτερική), αρχεία xml, και άλλα.

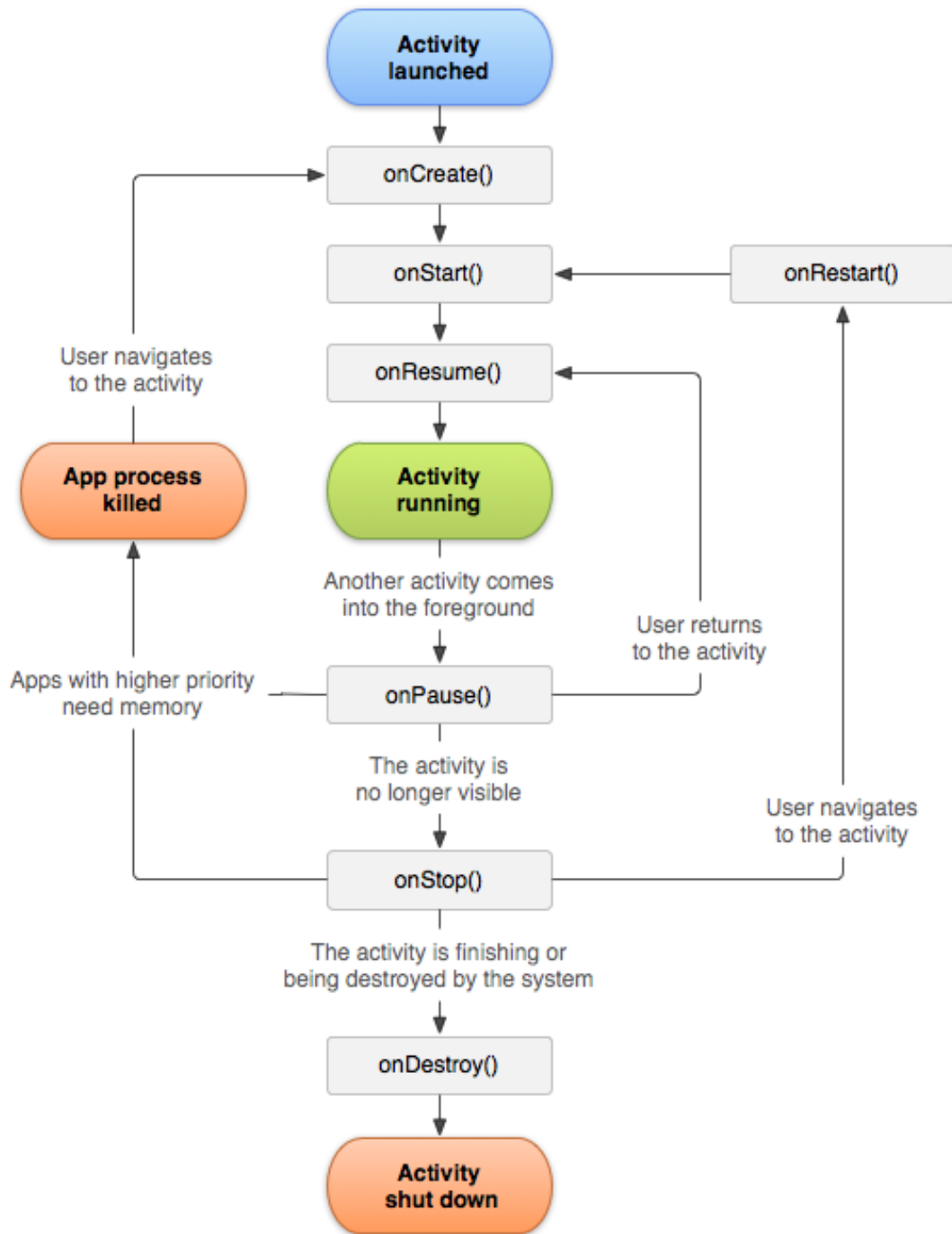
#### ***1.5.3.4 Notification Manager***

Δίνει την δυνατότητα στην εφαρμογή να ενημερώνει τον χρήστη για αλλαγές ή νέες προσθήκες στην εφαρμογή χωρίς ο χρήστης να χρειάζεται να ανοίξει την εφαρμογή και να φορτώσει τα δεδομένα απασχολώντας πολύτιμους πόρους του συστήματος. Για παράδειγμα Notifications μας στέλνει το Google Play Store για το ποιές εφαρμογές μας είναι έτοιμες για ενημέρωση χωρίς να ανοίγουμε εμείς κάθε φορά να κάνουμε έλεγχο για το αν κάποια εφαρμογή έχει διαθέσιμη ενημέρωση.

#### ***1.5.3.5 Activity Manager***

Διαχειρίζεται το πως συμπεριφέρεται μια εφαρμογή ανάλογα με το πως χρησιμοποιείται απο τον χρήστη. Για παράδειγμα τι μεθόδους θα καλέσει όταν ο χρήστης την έχει ανοικτή, όταν βρίσκεται στο παρασκήνιο και όταν τερματίζεται η λειτουργία της. Θα δούμε τον κύκλο ζωής της εφαρμογής στο παρακάτω διάγραμμα.





Εικόνα 18: Κύκλος ζωής της εφαρμογής

## 1.6 Εσωτερικό Εφαρμογής Android

Το εσωτερικό μιας εφαρμογής περιλαμβάνει τα ακόλουθα:

- Φάκελος src
- Φάκελος res
- AndroidManifest.xml
- Gradle Scripts

### 1.6.1 Φάκελος src

Ο πιο σημαντικός φάκελος μέσα στην εφαρμογή. Περιέχει όλα τα αρχεία .java στο κάθε package και είναι υπευθυνος για τον κώδικα της εφαρμογής. Αυτα μετατρέπονται σε αρχεία .dex για να μπορούν να γίνουν bytecode απο τους compilers Dalvik και ART.

Διαμορφώνεται κάπως έτσι:

Όνομα	Ημερομηνία τροπ...	Τύπος	Μέγεθος
adapters	23/1/2015 10:14 μμ	Φάκελος αρχείων	
com	23/1/2015 10:14 μμ	Φάκελος αρχείων	
dialogs	23/1/2015 10:14 μμ	Φάκελος αρχείων	
filter	23/1/2015 10:14 μμ	Φάκελος αρχείων	
graphs	23/1/2015 10:14 μμ	Φάκελος αρχείων	
lists	23/1/2015 10:14 μμ	Φάκελος αρχείων	
search	23/1/2015 10:14 μμ	Φάκελος αρχείων	

Εικόνα 20: Φάκελος src

Βλέπουμε τα packages ενω μέσα στο καθένα εμπεριέχονται τα αρχεία .java

## 1.6.2 Φάκελος res

Ο φάκελος res περιέχει διάφορους φακέλους αναλόγως το περιεχόμενο. Περιέχει τους φακέλους:

- Anim (Καθορίζει τα animations που θα χρησιμοποιηθούν για τις εναλλαγές μεταξύ activities ή για το πώς θα εμφανιστεί ένα πλαίσιο κειμένου και άλλα.)
- Drawable(Περιέχει τους φακέλους εικόνων (hdpi, mdpi, ldpi, xhdpi, xxhdpi) καθώς και αρχεία .xml που καθορίζουν συγκεκριμένο στυλ για κάποιες συμπεριφορές σε διάφορα κομμάτια κώδικα)
- Layout(Υπεύθυνος για τα αρχεία .xml που χρησιμοποιεί η εφαρμογή μας)
- Menu(Περιέχονται τα .xml που είναι υπεύθυνα για την δημιουργία μενού και υπομενού σε συγκεκριμένα σημεία της εφαρμογής μας)
- Values(Περιέχει τα αρχεία (styles.xml, colors.xml, dimens.xml, strings.xml) )
- Styles.xml(Περιέχει το πως θα είναι και θα φαίνεται η εφαρμογή στον τελικό χρήστη)
- Colors.xml(Περιέχει ένα πίνακα στατικών χρωμάτων στα οποία μπορούμε να έχουμε πρόσβαση από κάθε σημείο της εφαρμογής μας)
- Dimens.xml(Περιέχει αναφορές σε διάφορα περιθώρια και μετακινήσεις μεταξύ αντικειμένων της εφαρμογής μας)
- Strings.xml(Περιέχει στατικά πεδία κειμένου τα οποία μπορούν να μεταφραστούν με το οποίο μπορούμε να πετύχουμε το globalization και localization της εφαρμογής μας)

Η δομή φαίνεται στην παρακάτω εικόνα:

Όνομα	Ημερομηνία τροπ...	Τύπος	Μέγεθος
anim	23/1/2015 10:14 μμ	Φάκελος αρχείων	
drawable	23/1/2015 10:14 μμ	Φάκελος αρχείων	
drawable-hdpi	23/1/2015 10:14 μμ	Φάκελος αρχείων	
drawable-mdpi	23/1/2015 10:14 μμ	Φάκελος αρχείων	
drawable-xhdpi	23/1/2015 10:14 μμ	Φάκελος αρχείων	
drawable-xxhdpi	23/1/2015 10:14 μμ	Φάκελος αρχείων	
filter	17/12/2014 12:20 πμ	Φάκελος αρχείων	
layout	23/1/2015 10:14 μμ	Φάκελος αρχείων	
menu	23/1/2015 10:14 μμ	Φάκελος αρχείων	
values	23/1/2015 10:14 μμ	Φάκελος αρχείων	
values-el	23/1/2015 10:14 μμ	Φάκελος αρχείων	
values-w820dp	23/1/2015 10:14 μμ	Φάκελος αρχείων	
xml	23/1/2015 10:14 μμ	Φάκελος αρχείων	

Εικόνα 21: Φάκελος res

### 1.6.3 AndroidManifest.xml

Ίσως το πιο σημαντικό αρχείο μέσα σε όλη την εφαρμογή. Περιέχει το package name, αναφορές στα υπόλοιπα packages, κλάσεις, activities, permissions. Η δομή του φαίνεται παρακάτω:



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

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

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:configChanges="orientation|screenSize|keyboardHidden"
            android:screenOrientation="portrait"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:configChanges="orientation|screenSize|keyboardHidden"
            android:name=".CategoryActivity"
            android:screenOrientation="portrait"
            android:label="@string/title_activity_category" >
        </activity>
        <activity
            android:configChanges="orientation|screenSize"
            android:name="albums.Album1"
            android:screenOrientation="portrait"
            android:label="@string/title_album" >
        </activity>
        <activity
            android:configChanges="orientation|screenSize|keyboardHidden"
            android:name=".AndroidGridLayoutActivity"
            android:screenOrientation="portrait"
            android:label="@string/title_activity_grid_view" >
        </activity>
        <activity
            android:screenOrientation="portrait"
            android:name=".FullScreenViewActivity">
        </activity>
    </application>
</manifest>
```

Εικόνα 22: AndroidManifest.xml

### 1.6.4 Gradle Scripts

Το σύστημα διαθέτει ένα toolkit το οποίο παρέχει κάποιες συγκεκριμένες δυνατότητες (Τροποποίηση, Ρύθμιση, Επέκταση) της διαδικασίας του build. Παρέχεται η δυνατότητα δημιουργίας πολλών και πολύπλοκων εφαρμογών με την χρήση ίδιων προσθέτων (plugins), βιβλιοθηκών (libraries, dependencies) διαφόρων επεκτάσεων (modules). Το αρχείο build.gradle είναι υπεύθυνο για την προσθήκη όλων των παραπάνω, έτσι ώστε εάν θέλουμε να χρησιμοποιήσουμε μια custom library να μην κάνουμε εισαγωγή όλο το project στην εφαρμογή μας παρα μόνο μια αναφορά στο groupId, ArtifactId της βιβλιοθήκης που θέλουμε να χρησιμοποιήσουμε.

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

android {
    compileSdkVersion 21
    buildToolsVersion "21.1.1"

    defaultConfig {
        applicationId "com.providers.package"
        minSdkVersion 17
        targetSdkVersion 21
        versionCode 1
        versionName "1.0"
    }

    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_7
        targetCompatibility JavaVersion.VERSION_1_7
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:21.0.2'
    compile 'com.astuetz:pagerslidingtabstrip:1.0.1'
    compile 'com.koushikdutta.ion:ion:2.+@aar'
    compile 'com.mcxiaoke.volley:library:1.0.+@aar'
}
```

Εικόνα 23: Gradle Script

## 1.7 Δομικά μέρη μιας εφαρμογής

Τα δομικά μέρη μιας εφαρμογής περιλαμβάνουν τα εξής:

- Activities
- Services
- Intents
- Broadcast Receivers

### 1.7.1 Activity

Είναι οι διάφορες οθόνες μιας εφαρμογής. Εκεί μέσα ο κάθε προγραμματιστής επιλέγει πως θα συνδεθεί το γραφικό περιβάλλον της εφαρμογής με τον πηγαίο κώδικα ή εγκέφαλο της εφαρμογής. Όλα τα Activities δηλώνονται στο Manifest καθώς και τα packages στα οποία ανήκουν.

Όλα τα Activities μαζί αποτελούν το τελικό αποτέλεσμα που βλέπει ο χρήστης στην συσκευή του.

### 1.7.2 Services

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

### 1.7.3 Intents

Πρόκειται για διεργασίες που επιτρέπουν και καθορίζουν την επικοινωνία μεταξύ Activities. Δηλώνονται στο Manifest ή και προγραμματιστικά. Καθορίζουν ποιο Activity θα αρχίσει πρώτο καθώς και ποιο θα ακολουθήσει και επίσης χρησιμοποιούνται για να μεταφέρουν δεδομένα απο ενα Activity σε ένα άλλο. Επίσης χρησιμοποιούνται για πρόσβαση σε άλλες εφαρμογές όπως είναι οι ρυθμίσεις συστήματος της συσκευής ή στο να παραπέμψουν τον χρήστη να εγκαταστήσει μια άλλη εφαρμογή ή πρόσθετο που απαιτείται για την ορθή λειτουργία της εφαρμογής.

#### 1.7.4 Broadcast Receivers

Πρόκειται για υπηρεσίες που χρησιμοποιούνται με στόχο την λήψη, αποστολή, ενημέρωση του χρήστη ή και των άλλων εφαρμογών του συστήματος. Δεν είναι ορατές στον χρήστη και δηλώνονται στο Manifest. Χρησιμοποιούνται συχνά για ανταλλαγή μηνυμάτων, ενημέρωση του χρήστη μέσω push notifications και άλλα.

## Κεφάλαιο 2

### 2.1 Εγκατάσταση Λογισμικού

Για να μπορέσει ο προγραμματιστής να αναπτύξει την δικιά του εφαρμογή θα πρέπει να έχει εγκαταστήσει ένα ολοκληρωμένο περιβάλλον διαχείρισης και δημιουργίας εφαρμογών Android τα λεγόμενα IDE. Τα περιβάλλοντα αυτά διαθέτουν πολλά εργαλεία αποσφαλμάτωσης, built-in compiler, auto-complete text editor και επίσης αυτόματη δημιουργία όλων των χρήσιμων αρχείων και φακέλων που καθιστούν την εφαρμογή έτοιμη να τρέξει και να χρησιμοποιηθεί από τον τελικό χρήστη. Για την εγκατάσταση του περιβάλλοντος διαχείρισης απαιτείται να έχει εγκατεστημένο το JDK(Java Development Kit). Μετα θα μπορέσει να επιλέξει όποιο περιβάλλον προγραμματισμού αυτός θεωρεί πιο βολικό(Android Studio, Eclipse IDE(Android Developer Tools Built-in), Netbeans IDE, IntelliJ IDEA IDE) για να φτιάξει την εφαρμογή του. Θα πρέπει όμως να κατεβάσει όλα τα API's που θεωρεί απαραίτητα για την υλοποίηση καθώς και τα απαραίτητα αρχεία για την εικονική συσκευή που θα γίνει το debugging.

### 2.2 Διαδικασία Αποσφαλμάτωσης(Debugging)

Το κάθε περιβάλλον προγραμματισμού και δημιουργίας εφαρμογών Android περιλαμβάνει κάποια εργαλεία για την αποσφαλμάτωση της εφαρμογής. Ο προγραμματιστής επιλέγει πως θα υλοποιήσει την εφαρμογή αλλά το λειτουργικό σύστημα θα καθορίσει ποιούς πόρους θα χρησιμοποιήσει. Σε αυτήν την διαδικασία χρησιμοποιείται το Logcat το οποίο είναι ένας καταγραφέας για το οτιδήποτε συμβαίνει στην εφαρμογή σου και στους πόρους του συστήματος που χρησιμοποιεί. Είναι υπεύθυνο για να ενημερώσει τον προγραμματιστή για ότι προκύψει(μνήμη που χρησιμοποιεί η εφαρμογή, συντακτικά και λογικά λάθη στον κώδικα, σφάλματα στο μητρώο, παραβίαση αδειών, παραβίαση δεδομένων της εφαρμογής και άλλα). Επίσης ένα άλλο σημαντικό εργαλείο που χρησιμοποιούν οι προγραμματιστές είναι το DDMS το οποίο παρέχει την δυνατότητα τεσταρίσματος της εφαρμογής σε εικονικές συσκευές που δεν έχουν πρόσβαση σε ζωντανά δεδομένα. Αν πούμε ότι για παράδειγμα φτιάχνουμε μια εφαρμογή

ανταλλαγής μηνυμάτων κειμένου και στατικών εικόνων πριν την δοκιμάσουμε σε κανονική συσκευή και μας επιφέρει πρόσθετες χρεώσεις μόνο για τις δοκιμές μπορούμε να στέλνουμε fake notifications ή και μηνύματα στην εικονική συσκευή με στόχο την βελτίωση. Το ίδιο γίνεται και με την γεωγραφική θέση, κλήση αριθμού, προβολή website, προβολή στατιστικών heap memory allocation, thread memory size, network statistics και άλλα.

## 2.3 Δημοσίευση Εφαρμογής

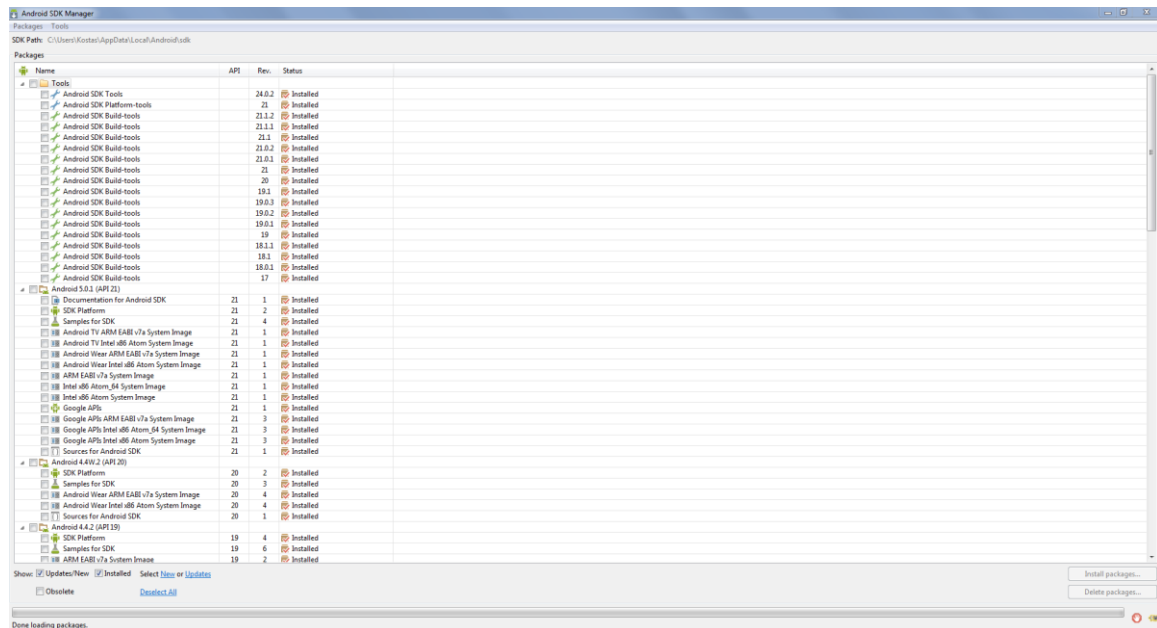
Για να δημοσιεύσουμε μια εφαρμογή χρειαζόμαστε το .apk αρχείο που θα μας δώσει ο compiler εφόσον περάσει επιτυχώς την διαδικασία της αποσφαλμάτωσης. Αν θέλουμε να δημοσιεύσουμε την εφαρμογή μας στο Google Play Store θα πρέπει να κάνουμε εγγραφή σαν προγραμματιστές και στην συνέχεια δίνοντας μια περιγραφή και μερικά screenshots η εφαρμογή μας δημοσιεύεται και είναι διαθέσιμη για κατέβασμα απο τον χρήστη. Την στιγμή που γράφουμε την παρούσα πτυχιακή εργασία δεν είναι διαθέσιμη η εγγραφή εμπόρου στην Ελλάδα δηλαδή οι προγραμματιστές εφόσον καταβάλλουν τα 25\$ που είναι το registration fee της Google για developer registration θα μπορούν να δημοσιεύουν εφαρμογές αλλά θα διατίθοντε δωρεάν στο ευρύ κοινό. Αυτό σημαίνει οτι δεν θα μπορούν να έχουν κέρδος απο τα downloads της εφαρμογής παρα μόνο απο τα in-app purchases(για παράδειγμα η αναβάθμιση μια εφαρμογής απο trial σε pro έκδοση μέσω paypal ή google wallet) ή το admob(διαδικασία πληρωμής απο διαφημίσεις της Google μέσα στην εφαρμογή).

## 2.4 Android System Development Kit

Για να μπορέσει ο προγραμματιστής να δημοσιεύσει μια εφαρμογή η οποία θα είναι διαθέσιμη στο κοινό θα πρέπει να χρησιμοποιήσει κάποιες βιβλιοθήκες και εργαλεία που είναι διαθέσιμα μεσω του Android SDK. Επίσης θα μπορεί να προβεί σε διαθέσιμες ενημερώσεις αναλογα με τις εκδόσεις του λειτουργικού συστήματος. Μπορεί να κατεβάσει τα απαραίτητα αυτά εργαλεία απο τον SDK Manager που είναι ενα built-in επέκταση σε κάθε περιβάλλον προγραμματισμού που θα επιλέξει να χρησιμοποιήσει.

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



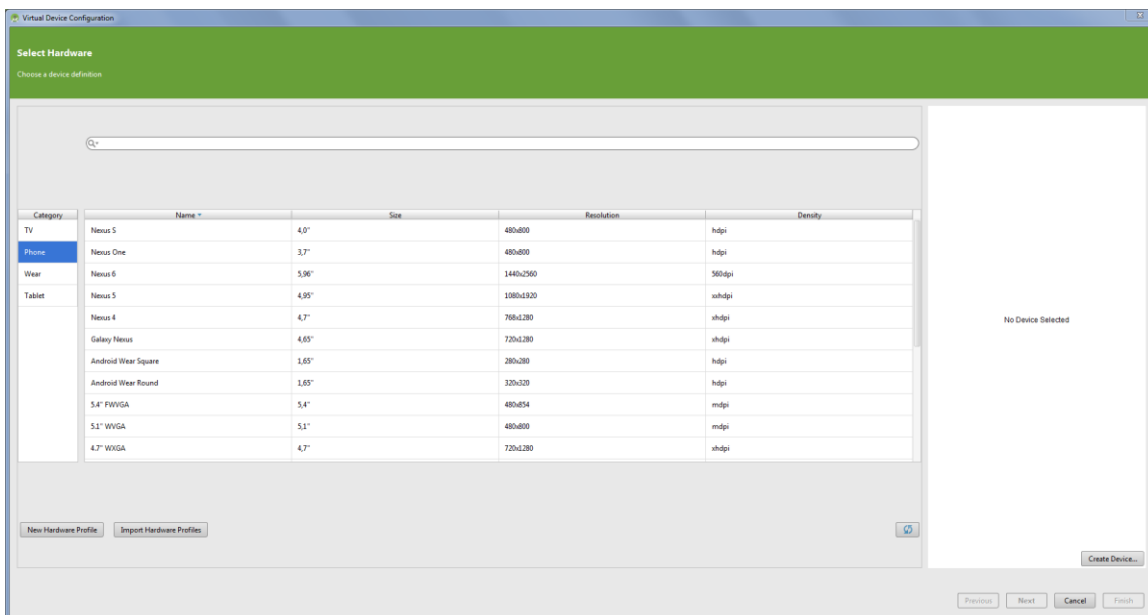


Εικόνα 24: Android SDK Manager

## 2.5 Android Virtual Device Manager(AVD)

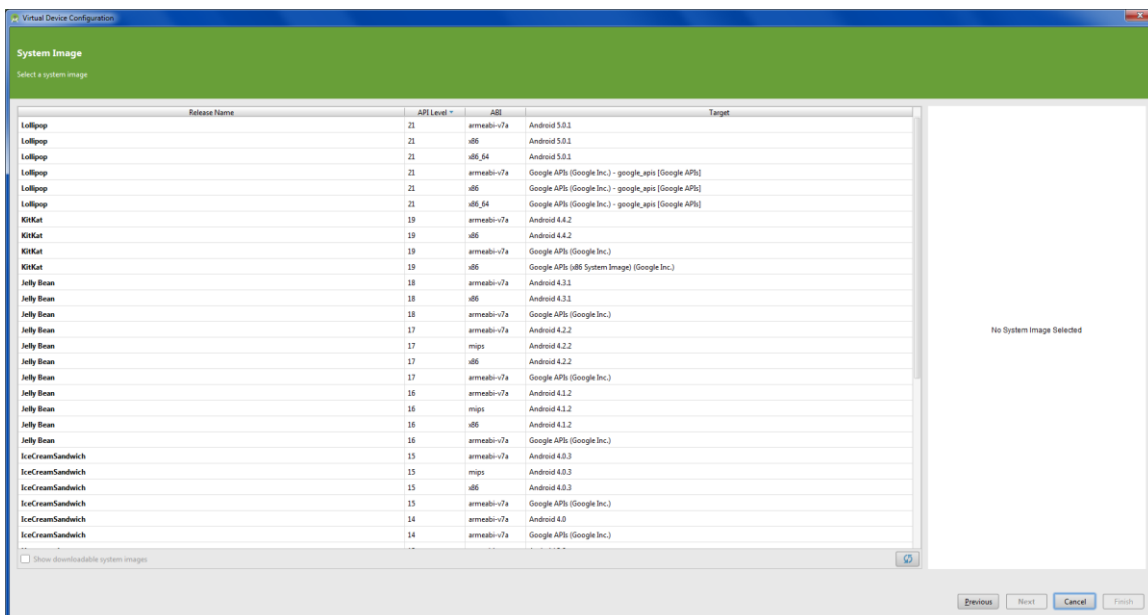
Επίσης θα πρέπει να κατεβάσουμε όλα τα απαραίτητα αρχεία για την δημιουργία εικονικής συσκευής με σκοπό να τεστάρουμε την εφαρμογής μας. Δουλειά η οποία αναλαμβάνει το AVD Android Virtual Device Manager με το κατάλληλο SDK και έκδοση του λειτουργικού συστήματος και με τον κατάλληλο επεξεργαστή. Βεβαίως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

Διαλέγουμε την συσκευή απο την λίστα



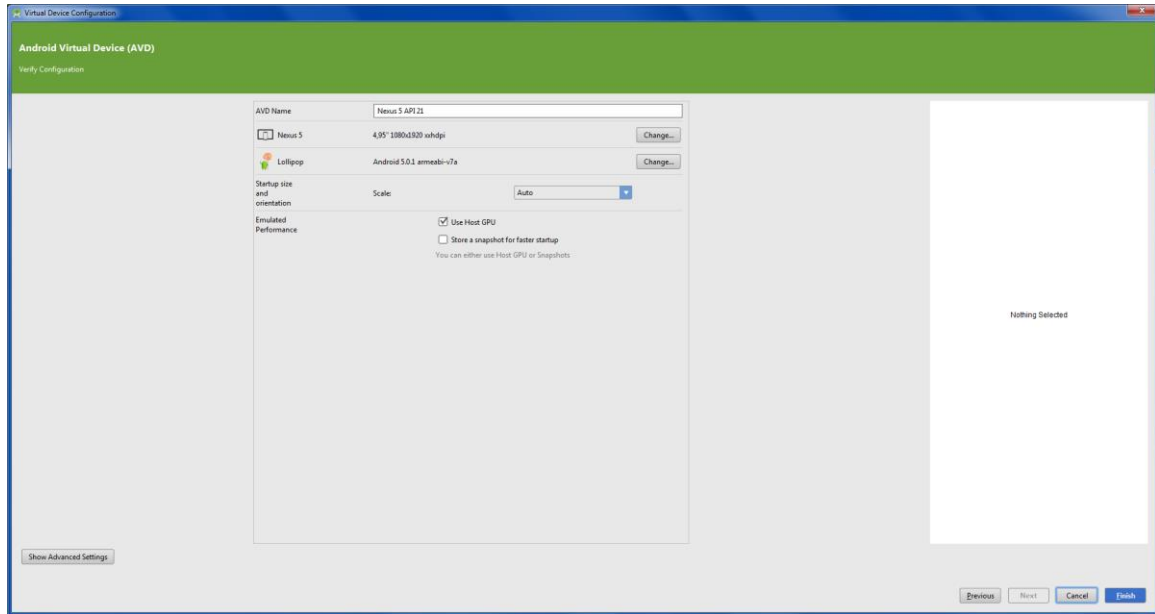
Εικόνα 25: AVD Manager Device Selection

Επιλέγουμε το SDK και την έκδοση καθώς και τον επεξεργαστή που θέλουμε να τρέχει η εικονική συσκευή μας:



Εικόνα 26: SDK and Processor Selection

Επιθεώρηση και δημιουργία της εικονικής συσκευής



Εικόνα 27: AVD inspection and Creation

## 2.6 Android Design Guidelines

Βασιζόμενοι στο <http://www.developers.android.com/design/> θα προσπαθήσουμε να ακολουθήσουμε τις υποδοχές για τον σχεδιασμό και υλοποίηση με βάση τις προτάσεις της εταιρίας που κατασκεύασε το λειτουργικό σύστημα. Βεβαίως μπορούμε να χρησιμοποιήσουμε τις δικές μας ιδέες απλώς ο σχεδιασμός μιας κακοσχεδιασμένης εφαρμογής θα απαιτεί πολύ περισσότερους πόρους στην συσκευή με σκοπό μια πολύπλοκη εφαρμογή να διακοπεί από το σύστημα εάν δεν είναι διαθέσιμοι οι πόροι ή ακόμα και περισσότερο χρόνο αποσφαλμάτωσης για τυχόν λάθη. Θα πρέπει να αποφασίσουμε αν η εφαρμογή μας θα καλύψει παλαιότερες εκδόσεις του λειτουργικού συστήματος, αν θα υποστηρίζει πολλαπλές συσκευές όπως (smartphones, tablets, tv, wearable) με σκοπό τον σχεδιασμό εικόνων και γραφικών για πολλά μεγέθη οθονών.

Οι εταιρίες πάντα θα βγάζουν συσκευές μεγαλύτερων αναλύσεων έτσι εάν θέλουμε η εφαρμογή μας να καλύπτει ένα μεγάλο ποσοστό συσκευών θα πρέπει να αναβαθμίζουμε συνεχώς την εφαρμογή.

Όταν θέλουμε να σχεδιάσουμε εφαρμογή που να υποστηρίζει πολλές συσκευές θα πρέπει να συμπεριλάβουμε υπόψη τα διάφορα μεγέθη τα οποία φαίνονται παρακάτω:

Density Bucket	Screen Density	Physical Size	Pixel Size
ldpi	120 dpi	0.5 x 0.5 in	0.5 in * 120 dpi = 60 x 60 px
mdpi	160 dpi	0.5 x 0.5 in	0.5 in * 160 dpi = 80 x 80 px
hdpi	240 dpi	0.5 x 0.5 in	0.5 in * 240 dpi = 120 x 120 px
xhdpi	320 dpi	0.5 x 0.5 in	0.5 in * 320 dpi = 160 x 160 px
xxhdpi	480 dpi	0.5 x 0.5 in	0.5 in * 480 dpi = 240 x 240 px

Εικόνα 28: Διαφορές σε διάφορα μεγέθη οθονών

Υπάρχουν διάφορες μονάδες μέτρησης μεγέθους οθόνης όπως είναι

- Px Είναι το pixel
- Sp Είναι scale-independent pixels
- Dip Είναι density-independent pixels
- In Είναι η ιντσα και αντιστοιχεί σε πραγματικές διαστάσεις οθόνης 1inch = 2.54 cm
- Pt Είναι points και αντιστοιχεί σε πραγματικές διαστάσεις οθόνης 1pt= 1/72 inch
- Dp Είναι density-independent pixels αλλά είναι μια abstract μονάδα μέτρησης για το μέγεθος. 1dp = 1px in 160 dpi screen

Παρακάτω θα δούμε ενα συνοπτικό πίνακα που εξηγεί καλύτερα τα μεγέθη των διαφόρων οθονών:

<b>Dimension</b>	<b>Description</b>	<b>Units / Physical Inch</b>	<b>Density Independent</b>	<b>Same Physical Size on Every Screen</b>
px	Pixels	Varies	No	No
in	Inches	1	Yes	Yes
mm	Millimeters	25.4	Yes	Yes
pt	Points	72	Yes	Yes
dp	Density independent pixels	~160	Yes	No
sp	Scale independent pixels	~160	Yes	No

Εικόνα 29: Αντιστοιχία μονάδων μέτρησης για μεγέθη οθονών

Εδω θα δούμε ενα πίνακα που συγκεντρώνει τα ποσοστά χρήσης μεταξύ εκδόσεων του Android μέχρι και τον Ιούνιο του 2014.



Εικόνα 30: Χρήση των Εκδόσεων Android

## 2.7 Android Logcat – Android Crash Report Manager

Ένα από τα βασικότερα εργαλεία μέσα στα περιβάλλοντα προγραμματισμού είναι το Android Logcat. Παρέχει πληροφορίες για το οτιδήποτε γίνεται μέσα στην εφαρμογή καθώς και τους πόρους, μνήμη, χώρο που καταλαμβάνει στην συσκευή. Επίσης μας παρέχει πληροφορίες για το εάν η εφαρμογή διακοπεί κατά την διάρκεια της αποσφαλμάτωσης κατά την διάρκεια του debugging καθώς και για τον λόγο διακοπής. Μας ενημερώνει για τα permissions που δίνουμε στην εφαρμογή καθώς και για το αν το thread μας καταλαμβάνει πάρα πολύ δουλειά παρόλο που εκτελείται επιτυχώς. Μπορούμε να δούμε το αποτέλεσμα του Logcat με μηνύματα verbose, info, warning και άλλα. Μια οθόνη Logcat είναι η παρακάτω:

```

02-01 19:18:05.510 1391-1381/? D/AndroidRuntime: Shutting down VM
02-01 19:18:05.516 1390-1390/? I/art: Late-enabling -Xcheck:jni
02-01 19:18:05.516 1391-1389/? E/art: Thread stratching while runtime is shutting down: Binder_1
02-01 19:18:05.516 1391-1389/? I/AndroidRuntime: NOTE: attach of thread 'Binder_1' failed
02-01 19:18:05.564 451-474/system_process D/: HostConnection:connect() New Host Connection established 0xac82bc9f0, tid 474
02-01 19:18:05.820 1390-1406/? D/OpenGLRenderer: Render dirty regions requested: true
02-01 19:18:05.824 1390-1390/? D/: HostConnection:connect() New Host Connection established 0xac863800, tid 1390
02-01 19:18:05.831 1390-1390/? D/Atlas: Validating map...
02-01 19:18:05.857 1390-1406/? D/libEGL: loaded /system/lib/egl/libEGL_emulation.so
02-01 19:18:05.858 1390-1406/? D/libEGL: loaded /system/lib/egl/libGLESv1_CM_emulation.so
02-01 19:18:05.864 1390-1406/? D/libEGL: loaded /system/lib/egl/libGLESv2_emulation.so
02-01 19:18:05.881 1390-1406/? D/: HostConnection:connect() New Host Connection established 0xac8639e0, tid 1406
02-01 19:18:05.902 1390-1406/? I/OpenGLRenderer: Initialized EGL, version 1.4
02-01 19:18:05.939 1390-1406/? D/OpenGLRenderer: Enabling debug mode 0
02-01 19:18:05.965 1390-1406/? W/EGL_emulation: eglSurfaceAttrib not implemented
02-01 19:18:05.965 1390-1406/? W/OpenGLRenderer: Failed to set EGL_SWAP_BEHAVIOR on surface 0xac87c460, error=EGL_SUCCESS
02-01 19:18:06.558 1390-1406/com.kostas.stockpredictions W/EGL_emulation: eglSurfaceAttrib not implemented
02-01 19:18:06.558 1390-1406/com.kostas.stockpredictions W/OpenGLRenderer: Failed to set EGL_SWAP_BEHAVIOR on surface 0xac87c460, error=EGL_SUCCESS
02-01 19:18:06.589 1390-1390/com.kostas.stockpredictions I/Choreographer: Skipped 46 frames! The application may be doing too much work on its main thread.
02-01 19:18:06.841 451-474/system_process I/ActivityManager: Displayed com.kostas.stockpredictions/.ListLoaderActivity: +1s339ms
02-01 19:18:10.214 451-508/system_process W/AudioTrack: AUDIO_OUTPUT_FLAG_FAST denied by client
02-01 19:18:10.220 451-863/system_process I/ActivityManager: START u0 [cmp=com.kostas.stockpredictions/.StockItem (has extras)] from uid 10061 on display 0
02-01 19:18:10.329 1390-1406/com.kostas.stockpredictions W/EGL_emulation: eglSurfaceAttrib not implemented
02-01 19:18:10.329 1390-1406/com.kostas.stockpredictions W/OpenGLRenderer: Failed to set EGL_SWAP_BEHAVIOR on surface 0xa4fe3040, error=EGL_SUCCESS
02-01 19:18:13.288 451-792/system_process I/ActivityManager: Killing 1009:com.android.dialer/u0e4 (adj 15): empty #17
02-01 19:18:13.305 451-863/system_process W/libprocessgroup: failed to open /acct/uid_10004/pid_1009/cgroup.procs: No such file or directory
02-01 19:18:15.789 451-469/system_process W/art: Long monitor contention event with owner method=void com.android.server.am.ActivityManagerService$AppDeathRecipient.binderDied() from ActivityManagerService.java:1163 waiters=0 for 2.054s
02-01 19:18:15.790 451-498/system_process W/art: Long monitor contention event with owner method=void com.android.server.am.ActivityManagerService$AppDeathRecipient.binderDied() from ActivityManagerService.java:1163 waiters=1 for 866ms
02-01 19:18:15.794 1390-1390/com.kostas.stockpredictions I/Choreographer: Skipped 322 frames! The application may be doing too much work on its main thread.
02-01 19:18:15.832 451-474/system_process I/ActivityManager: Displayed com.kostas.stockpredictions/.StockItem: +5s61ms
02-01 19:18:19.348 451-508/system_process W/AudioTrack: AUDIO_OUTPUT_FLAG_FAST denied by client
02-01 19:18:19.391 1390-1406/com.kostas.stockpredictions W/EGL_emulation: eglSurfaceAttrib not implemented
02-01 19:18:19.391 1390-1406/com.kostas.stockpredictions W/OpenGLRenderer: Failed to set EGL_SWAP_BEHAVIOR on surface 0xa4899700, error=EGL_SUCCESS
02-01 19:18:20.281 1160-1199/com.android.calendar D/InitialsService: Clearing and rescheduling alarms.
02-01 19:18:20.284 451-870/system_process I/ActivityManager: Killing 1079:com.android.managedprovisioning/u0a7 (adj 15): empty #17
02-01 19:18:20.294 451-753/system_process W/libprocessgroup: failed to open /acct/uid_10007/pid_1079/cgroup.procs: No such file or directory
02-01 19:18:20.622 451-469/system_process W/art: Long monitor contention event with owner method=void com.android.server.am.ActivityManagerService.serviceDoneExecuting(android.os.IBinder, int, int, int) from ActivityManagerService.java:1491
02-01 19:18:20.626 451-469/system_process W/art: Long monitor contention event with owner method=void com.android.server.am.ActivityManagerService$AppDeathRecipient.binderDied() from ActivityManagerService.java:1163 waiters=1 for 278ms
02-01 19:18:48.022 1326-1326/com.android.musicfx V/MusicFXControlPanelReceiver: onReceive
02-01 19:18:48.022 1326-1326/com.android.musicfx V/MusicFXControlPanelReceiver: Action: android.media.action.CLOSE_AUDIO_EFFECT_CONTROL_SESSION
02-01 19:18:48.023 1326-1326/com.android.musicfx V/MusicFXControlPanelReceiver: Package name: com.android.music
02-01 19:18:48.023 1326-1326/com.android.musicfx V/MusicFXControlPanelReceiver: Audio session: 8
02-01 19:18:48.023 1326-1326/com.android.musicfx V/MusicFXControlPanelEffect: closeSession(android.app.ReceiverRestrictedContext@1e1a10e5, com.android.music, 8)
02-01 19:18:48.025 451-863/system_process I/MediaFocusControl: AudioFocus abandonAudioFocus() from android.media.AudioManager@1e1a10e5com.android.music.MediaIFlybackServices$83895d27ba
02-01 19:18:48.027 451-821/system_process I/ActivityManager: Killing 884:com.android.music/u0e42 (adj 15): empty #17
02-01 19:18:48.044 451-752/system_process W/libprocessgroup: failed to open /acct/uid_10042/pid_884/cgroup.procs: No such file or directory

```

Εικόνα 31: Logcat Output

Μπορούμε να καταγράψουμε τα δικά μας δεδομένα για αποσφαλμάτωση χρησιμοποιώντας τις παρακάτω εντολές

- `Log.v("INTENT_TAG", "Intent Successful");` για να δούμε αποτέλεσμα εαν το intent ήταν επιτυχές σε verbose level
- `Log.d("INTENT_TAG", "Intent Successful");` για να δούμε αποτέλεσμα εαν το intent ήταν επιτυχές σε debug level
- `Log.e("INTENT_TAG", "Intent Successful");` για να δούμε αποτέλεσμα εαν το intent ήταν επιτυχές σε error level

*`Toast.makeText(Activity.this, "This is just a plain Message", Toast.LENGTH_SHORT).show();`*

- `Log.i("INTENT_TAG", "Intent Successful");` για να δούμε αποτέλεσμα εαν το intent ήταν επιτυχές σε info level
- `Log.w("INTENT_TAG", "Intent Successful");` για να δούμε αποτέλεσμα εαν το intent ήταν επιτυχές σε warning level







Σηνήτως το χρησιμοποιούμε σε Exception Handlers και Catch Statements.

## 2.8 Notifications


Πολύ σημαντικό είναι για ένα προγραμματιστή να μπορεί να ενημερώνει τον χρήστη για κάποια λογικά ή συντακτικά λάθη με βάση την ορθή χρήση της εφαρμογής. Αυτό μπορεί να επιτευχθεί με πολλούς τρόπους 3 εξ αυτών είναι οι πιο διαδεδομένοι.

- Toast Message Πρόκειται για floating μηνύματα που προσπαθούν να βελτιώσουν την ορθή λειτουργία της εφαρμογής.  
Ενα παράδειγμα υποδομής ενός τέτοιου μηνύματος είναι

Και ο χρήστης βλέπει το ακόλουθο:

	Stock Name: AAAK Stock Price: 2.81
	Stock Name: ABE Stock Price: 0.341
	Stock Name: ALPHA Stock Price: 1.586
	Stock Name: AKPIT Stock Price: 0.115
	Stock Name: ANEK Stock Price: 0.118
	Stock Name: ATE Stock Price: 0.155

This is just a plain Message



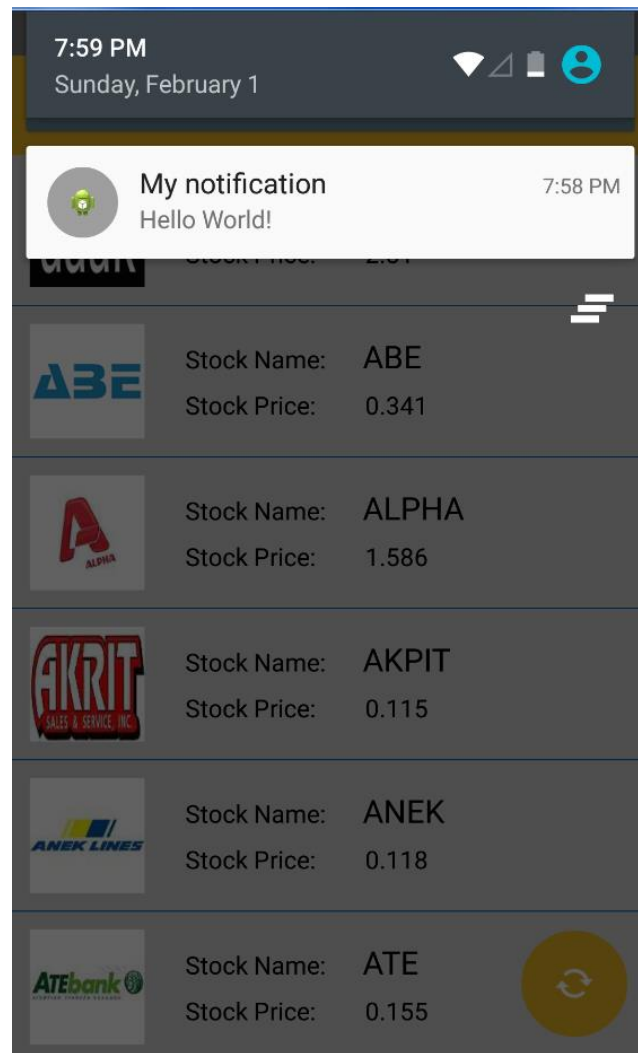
## Εικόνα 32: Toast Message

- Push Notifications Χρησιμοποιούνται για την ενημέρωση του χρήστη έτσι ώστε να μην χρειάζεται να ελέγχει την εφαρμογή καθημερινά. Με κώδικα υλοποιείται ως εξής:

Ενω ο χρήστης βλέπει το τελικό αποτέλεσμα ως εξής:

```
NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(this)
    .setSmallIcon(R.drawable.ic_launcher)
    .setContentTitle("My notification")
    .setContentText("Hello World!");
Intent resultIntent = new Intent(this, ListLoaderActivity.class);
TaskStackBuilder stackBuilder = TaskStackBuilder.create(this);
    stackBuilder.addParentStack(ListLoaderActivity.class);
    stackBuilder.addNextIntent(resultIntent);
PendingIntent resultPendingIntent = stackBuilder.getPendingIntent(
    0,
    PendingIntent.FLAG_UPDATE_CURRENT
);
mBuilder.setContentIntent(resultPendingIntent);
NotificationManager mNotificationManager =(NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
mNotificationManager.notify(1, mBuilder.build());
```





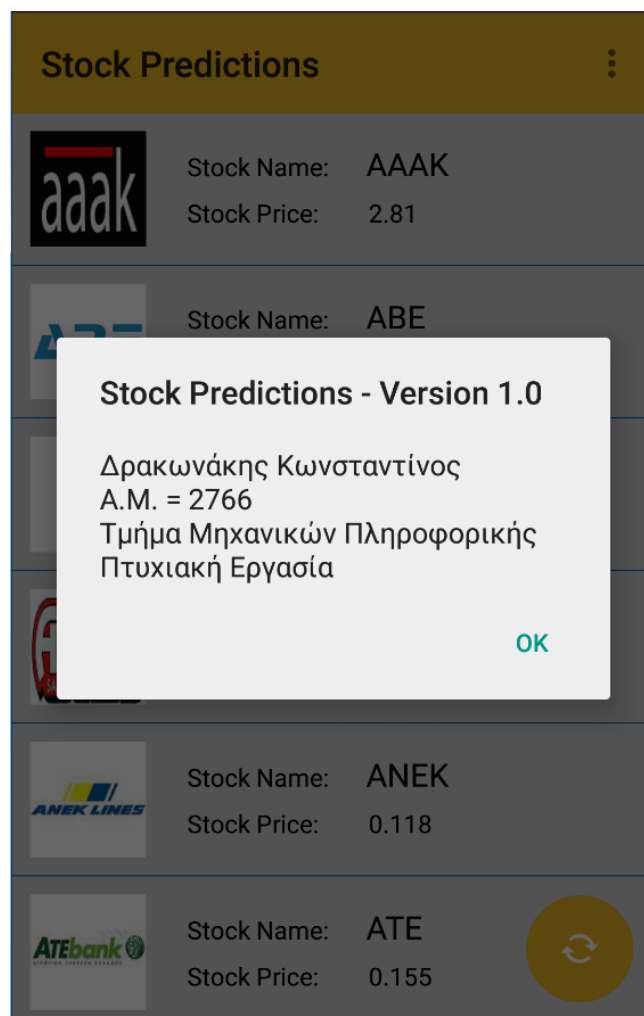
Εικόνα 33: Notification Message

- Alert Dialog Χρησιμοποιείται όταν θέλουμε να εμφανίσουμε μια πληροφορία η ένα μήνυμα λάθους σε μια φόρμα.

Με κώδικα υλοποιείται ως εξής:

```
public AlertDialog showMessage(Context context, String title, String message){
    AlertDialog.Builder builder = new AlertDialog.Builder(context);
        builder.setTitle(title);
        builder.setMessage(message);
    builder.setPositiveButton("Ok", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.cancel();
        }
    });
    builder.show();
}
```

Ο χρήστης βλέπει το τελικό αποτέλεσμα ως εξής:



Εικόνα 34: Alert Dialog

## 2.9 Developing In Android from Programmer View

Πιο πάνω αναλύσαμε όλα αυτά τα guidelines που χρησιμοποιεί ένας προγραμματιστής προκειμένου να μην χάσει ένα μεγάλο μέρος χρηστών διότι αυτοί μπορεί να είναι πιθανοί πελάτες του. Δεν μπορεί να αγνοήσει όλα τα μεγέθη οθόνης γιατί είναι μέρος των χρηστών που απευθύνεται μιας και δεν έχουν όλοι οι χρήστες την οικονομική δυνατότητα να αγοράζουν καινούργιες συσκευές με στόχο να καταργούνται κάποια API's. Βέβαια αυτό μπορεί να αποβεί ασύμφορο για έναν προγραμματιστή, πόσο μάλλον μια εταιρία όπως είναι η Killjo Games στην περίπτωση του "Subway Surfers" ενός παιχνιδιού που διατίθεται δωρεάν και περιλαμβάνει in-app purchases για πιο γρήγορο gameplay και λιγότερο ανταγωνισμό. Βέβαια δείτε το λίγο και από την μεριά της εταιρίας, ότι θα πρέπει να αποσύρει κάποιες εκδόσεις αλλιώς θα προβεί σε πτώχευση έτσι όπως εξελίσσεται το λειτουργικό σύστημα.

Το ερώτημα το οποίο πρέπει να απασχολεί κάθε προγραμματιστή είναι "Πόσο εύκολο είναι να δημιουργηθεί μια εφαρμογή σε μια πλατφόρμα?". Ας υποθέσουμε ότι είστε χρήστης των Windows και θα πρέπει απλώς να κατεβάσετε το JDK, έπειτα ένα περιβάλλον προγραμματισμού Eclipse, Android Studio, NetBeans, IntelliJ Idea, να κατεβάσετε τα απαραίτητα API's, και τα απαραίτητα αρχεία που θα πρέπει να δημιουργήσουν την εικονική συσκευή. Έννοείτε βέβαια ότι το Android παρέχει αρκετό documentation και μια μεγάλη online κοινότητα καθώς και διάφορα φόρουμ όπου θα προσφερθεί βοήθεια όπως το Stack Overflow, Android Enthusiasts, Super User καθώς και blogs με διάφορα tutorials που πλέον παρέχουν κάθε λύση σε κάθε πρόβλημα ή ακόμα και ένα παράδειγμα πολύ κοντά σε αυτό που ψάχνουμε.

Αυτό που αντιμετωπίζει συχνά ένας προγραμματιστής είναι οι κριτικές που δέχονται οι εφαρμογές του και οι πιο πολλές περιλαμβάνουν λέξεις όπως «διακόπηκε», «δεν δουλεύει», «πολλές διαφημίσεις» και άλλα. Για την επιτυχία μιας εφαρμογής πρέπει να παρθούν οι σωστές αποφάσεις στον σωστό χρόνο. Ο προγραμματιστής πρέπει να λάβει υπόψη ότι μια εφαρμογή δεν γίνεται έτσι απλά να επιτύχει απλώς βεβαιώνοντας τον ότι δουλεύει. Οι χρήστες θα προτιμήσουν εφαρμογές που έχουν καλό performance, low battery consumption, κατανάλωση λιγότερων πόρων.

Η κάθε εφαρμογή είναι ένα software. Στην αγορά υπάρχουν πολλές εφαρμογές που μπορούν να κάνουν την δικιά σου εφαρμογή να συμπεριφέρεται λανθασμένα. Ένας λόγος που το λειτουργικό σύστημα Android είναι το διαδεδομένο είναι ότι βρίσκεται πολύ πιο μπροστά από το iOS and Windows Mobile, παρέχεται δωρεάν η πλατφόρμα ανάπτυξης εφαρμογών σε αντίθεση με τα άλλα 2 που έχουν ετήσια συνδρομή μόνο για να ανάπτυξη εφαρμογής 99\$ ετησίως.

## Κεφάλαιο 3

### 3.1 Ανάλυση απαιτήσεων για την δημιουργία της εφαρμογής

Ο σκοπός είναι να φτιάξουμε μια εφαρμογή πρόβλεψης τιμών μετοχών. Η εφαρμογή θα πρέπει να φορτώνει τις μετοχές καθώς και τις τιμές των 15 τελευταίων ημερών και επίσης θα πρέπει να φτιάχνει ένα γράφημα με σκοπό να βλέπει ο χρήστης πως κυμαίνεται η τιμή της μετοχής.

#### 3.1.1 Δημιουργία Βάσης Δεδομένων

				id	name	price	datecreated	price1	price2	price3	price4	price5	price6	price7	price8	price9	price10	price11	price12	price13	price14	price15	image
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	1	AAAK	2.81	2014-01-17 01:27:52	3.33	2.65	6.21	0.021	4.89	10.58	6.21	7.25	6.147	0.0254	6.88	9.21	11.25	10.93	11.01	http://spinner.bl.ee/images/aa.png
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	2	ABE	0.341	2014-01-17 01:29:50	0.289	0.24	0.98	2.15	3.174	0.258	6.25	88.025	63.04	2.014	3.25	9.01	4.102	8.025	70.95	http://spinner.bl.ee/images/ABE.png
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	3	ALPHA	1.586	2014-01-17 01:30:29	0.721	0.698	0.658	0.632	0.025	3.24	1.201	9.41	41.025	3.025	9.02	8.056	4.23	4.052	2.01	http://spinner.bl.ee/images/ALPHA.png
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	4	AKPIT	0.115	2014-01-17 01:31:00	0.025	0.47	1.025	3.214	0.369	2.041	3.14	6.214	3.025	9.024	6.025	6.014	5.02	3.01	4.12	http://spinner.bl.ee/images/AKPIT.png
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	5	ANEK	0.118	2014-01-17 01:31:26	0.119	14.2	7.89	5.64	3.28	2.21	1.01	0.025	3.012	6.012	4.05	2.13	6.144	9.04	10.54	http://spinner.bl.ee/images/ANEK.png
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	6	ATE	0.155	2014-01-17 01:32:03	0.155	0.254	0.365	0.147	0.21	1.02	2.54	5.22	9.32	7.01	8.23	8.89	4.32	10.201	3.214	http://spinner.bl.ee/images/ATE.png
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	7	ATEK	2.14	2014-01-17 01:32:15	1.9	2.4	5.654	3.214	0.089	4.85	7.12	4.021	6.023	3.12	6.14	5.14	8.201	9.01	7.01	http://spinner.bl.ee/images/ATEK.png
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	8	ATT	0.248	2014-01-17 01:32:38	0.216	0.987	1.002	6.25	4.25	9.521	15.03	11.02	52.014	3.14	9.012	12.36	8.96	16.201	22.36	http://spinner.bl.ee/images/ATT.png
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	9	ATTIK	0.028	2014-01-17 01:32:49	0.67	0.365	2.35	0.254	1.254	3.214	0.024	0.014	6.54	7.25	6.301	10.25	14.55	30.01	15.02	http://spinner.bl.ee/images/ATTIK.png
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	10	ATTIKA	0.76	2014-01-17 01:33:18	0.356	0.854	1.023	4.521	6.21	7.98	4.321	25.01	3.02	4.025	2.66	9.55	4.01	3.33	2.52	http://spinner.bl.ee/images/ATTIKA.png
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	11	AXON	0.139	2014-01-17 01:33:51	1.658	0.254	2.143	3.214	8.25	12.021	3.21	11.02	3.66	6.563	4.201	3.2	1.025	3.98	7.85	http://spinner.bl.ee/images/AXON.png
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	12	BINTA	3.66	2014-01-17 01:34:06	0.548	6.201	4.052	30.21	25.14	4.14	0.254	6.2	3.21	6.044	6.22	3.14	3.05	6.21	3.02	http://spinner.bl.ee/images/BINTA.png
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	13	BIOT	0.177	2014-01-17 01:34:21	6.25	0.998	1.254	3.336	4.563	7.88	9.987	9.201	4.015	12.36	5.124	6.047	9.203	11.201	32.05	http://spinner.bl.ee/images/BIOT.png
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	14	BOX	1.3	2014-01-17 01:34:39	2.18	0.214	7.521	6.214	8.963	5.214	0.0214	6.014	2.02	20.11	18.52	17.63	12.01	13.054	18.66	http://spinner.bl.ee/images/BOX.png
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	15	HPAK	1.28	2014-01-17 01:36:01	0.069	0.089	1.023	4.521	3.62	4.89	7.25	10.23	5.14	3.215	9.154	5.15	6.045	7.04	65.02	http://spinner.bl.ee/images/HPAK.png
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	16	IATP	0.85	2014-01-17 01:36:19	1.29	6.25	5.14	1.236	9.521	7.321	6.25	1.025	6.05	4.045	7.523	9.02	10.053	6.45	3.04	http://spinner.bl.ee/images/IATP.png
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	17	MPOKKA	0.055	2014-01-17 01:37:12	0.5874	0.24	0.147	1.02	2.52	3.414	5.124	6.0154	7.05	9.025	12.032	3.045	5.04	6.25	3.0174	http://spinner.bl.ee/images/MPOKKA.png
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	18	NHP	0.325	2014-01-17 01:38:56	3.17	4.36	5.21	3.214	3.258	8.25	9.52	5.014	6.98	9.58	6.05	1.023	0.015	9.05	4.02	http://spinner.bl.ee/images/NHP.png
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	19	OTE	11.46	2014-01-17 01:39:25	12.76	13.56	18.96	25.3	100.24	96.52	75.301	56.056	45.05	63.012	47.512	85.05	96.201	63.05	20.045	http://spinner.bl.ee/images/OTE.png
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	20	OPAP	10.54	2014-01-17 01:39:35	11.77	47.56	35.21	10.26	3.54	8.645	19.24	15.023	14.06	17.056	22.01	29.065	35.026	45.01	68.025	http://spinner.bl.ee/images/OPAP.png
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	21	ATOX	5.246	2014-01-17 02:05:39	218.054	521.36	410.23	300.214	10.24	95.32	75.396	21.056	45.065	35.054	88.025	66.045	84.02	63.021	84.253	http://spinner.bl.ee/images/ATOX.png
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	22	ALOXC	2.69	2014-03-17 17:52:33	5.36	9.214	3.25	0.024	96.25	8.14	5.201	2.02	6.01	6.0145	5.041	3.021	1.02	36.025	56.02	http://spinner.bl.ee/images/ALOXC.png
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	23	HAVOK	69.24	2014-03-17 20:52:57	5.36	75.14	52.36	4.21	0.029	8.65	7.302	3.05	1.02	5.301	6.05	4.05	5.23	9.041	7.05	http://spinner.bl.ee/images/HAVOK.png
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	24	VENDOK	5.39	2014-03-17 20:52:57	8.201	9.35	4.05	5.24	9.25	16.88	8.936	5.63	4.05	3.33	6.05	8.05	7.056	9.99	10.54	http://spinner.bl.ee/images/VENDOK.png

Εικόνα 35: Βάση δεδομένων για μετοχές

#### Κώδικας για την δημιουργία της βάσης:

```
CREATE TABLE IF NOT EXISTS `metoxes` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `name` varchar(50) CHARACTER SET latin1 NOT NULL,
```

```
`price` float NOT NULL,  
`datecreated` datetime NOT NULL,  
`price1` float NOT NULL,  
`price2` float NOT NULL,  
`price3` float NOT NULL,  
`price4` float NOT NULL,  
`price5` float NOT NULL,  
`price6` float NOT NULL,  
`price7` float NOT NULL,  
`price8` float NOT NULL,  
`price9` float NOT NULL,  
`price10` float NOT NULL,  
`price11` float NOT NULL,  
`price12` float NOT NULL,  
`price13` float NOT NULL,  
`price14` float NOT NULL,  
`price15` float NOT NULL,  
`image` varchar(250) NOT NULL,  
PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=25 ;
```

**Κώδικας για την εισαγωγή των δεδομένων:**

```
INSERT INTO `metoxes` (`id`, `name`, `price`, `datecreated`, `price1`, `price2`, `price3`, `price4`, `price5`,  
`price6`, `price7`, `price8`, `price9`, `price10`, `price11`, `price12`, `price13`, `price14`, `price15`, `image`)  
VALUES  
(1, 'AAAK', 2.81, '2014-01-17 01:27:52', 3.33, 2.65, 6.21, 0.021, 4.89, 10.58, 6.21, 7.25, 6.147, 0.0254, 6.88,  
9.21, 11.25, 10.93, 11.01, 'http://spinner.bl.ee/images/aa.png'),  
(2, 'ABE', 0.341, '2014-01-17 01:29:50', 0.289, 0.24, 0.98, 2.15, 3.174, 0.258, 6.25, 88.025, 63.04, 2.014, 3.25,  
9.01, 4.102, 8.025, 70.95, 'http://spinner.bl.ee/images/ABE.png'),  
(3, 'ALPHA', 1.586, '2014-01-17 01:30:29', 0.721, 0.698, 0.658, 0.632, 0.025, 3.24, 1.201, 9.41, 41.025, 3.025,  
9.02, 8.056, 4.23, 4.052, 2.01, 'http://spinner.bl.ee/images/ALPHA.png'),  
(4, 'AKPIT', 0.115, '2014-01-17 01:31:00', 0.025, 0.47, 1.025, 3.214, 0.369, 2.041, 3.14, 6.214, 3.025, 9.024,  
6.025, 6.014, 5.02, 3.01, 4.12, 'http://spinner.bl.ee/images/AKPIT.png'),  
(5, 'ANEK', 0.118, '2014-01-17 01:31:26', 0.119, 14.2, 7.89, 5.64, 3.28, 2.21, 1.01, 0.025, 3.012, 6.012, 4.05,  
2.13, 6.144, 9.04, 10.54, 'http://spinner.bl.ee/images/ANEK.png'),  
(6, 'ATE', 0.155, '2014-01-17 01:32:03', 0.155, 0.254, 0.365, 0.147, 0.21, 1.02, 2.54, 5.22, 9.32, 7.01, 8.23, 8.89,  
4.32, 10.201, 3.214, 'http://spinner.bl.ee/images/ATE.png'),  
(7, 'ATEK', 2.14, '2014-01-17 01:32:15', 1.9, 2.4, 5.654, 3.214, 0.089, 4.85, 7.12, 4.021, 6.023, 3.12, 6.14, 5.14,  
8.201, 9.01, 7.01, 'http://spinner.bl.ee/images/ATEK.png'),  
(8, 'ATT', 0.248, '2014-01-17 01:32:38', 0.216, 0.987, 1.002, 6.25, 4.25, 9.521, 15.03, 11.02, 52.014, 3.14, 9.012,  
12.36, 8.96, 16.201, 22.36, 'http://spinner.bl.ee/images/ATT.png'),  
(9, 'ATTIK', 0.028, '2014-01-17 01:32:49', 0.67, 0.365, 2.35, 0.254, 1.254, 3.214, 0.024, 0.014, 6.54, 7.25, 6.301,  
10.25, 14.55, 30.01, 15.02, 'http://spinner.bl.ee/images/ATTIK.png'),  
(10, 'ATTIKA', 0.76, '2014-01-17 01:33:18', 0.356, 0.854, 1.023, 4.521, 6.21, 7.98, 4.321, 25.01, 3.02, 4.025,  
2.66, 9.55, 4.01, 3.33, 2.52, 'http://spinner.bl.ee/images/ATTIKA.png'),  
(11, 'AXON', 0.139, '2014-01-17 01:33:51', 1.658, 0.254, 2.143, 3.214, 8.25, 12.021, 3.21, 11.02, 3.66, 6.563,  
4.201, 3.2, 1.025, 3.98, 7.85, 'http://spinner.bl.ee/images/AXON.png'),  
(12, 'BINTA', 3.66, '2014-01-17 01:34:06', 0.548, 6.201, 4.052, 30.21, 25.14, 4.14, 0.254, 6.2, 3.21, 6.044, 6.22,  
3.14, 3.05, 6.21, 3.02, 'http://spinner.bl.ee/images/BINTA.png'),  
(13, 'BIOT', 0.177, '2014-01-17 01:34:21', 6.25, 0.998, 1.254, 3.336, 4.563, 7.88, 9.987, 9.201, 4.015, 12.36,  
5.124, 6.047, 9.203, 11.201, 32.05, 'http://spinner.bl.ee/images/BIOT.png'),  
(14, 'BOX', 1.3, '2014-01-17 01:34:39', 2.18, 0.214, 7.521, 6.214, 8.963, 5.214, 0.0214, 6.014, 2.02, 20.11, 18.52,  
17.63, 12.01, 13.054, 18.66, 'http://spinner.bl.ee/images/BOX.png'),  
(15, 'HPAK', 1.28, '2014-01-17 01:36:01', 0.069, 0.089, 1.023, 4.521, 3.62, 4.89, 7.25, 10.23, 5.14, 3.215, 9.154,  
5.15, 6.045, 7.04, 65.02, 'http://spinner.bl.ee/images/HPAK.png'),  
(16, 'IATP', 0.85, '2014-01-17 01:36:19', 1.29, 6.25, 5.14, 1.236, 9.521, 7.321, 6.25, 1.025, 6.05, 4.045, 7.523,  
9.02, 10.053, 6.45, 3.04, 'http://spinner.bl.ee/images/IATP.png'),  
(17, 'MPOKKA', 0.055, '2014-01-17 01:37:12', 0.5874, 0.24, 0.147, 1.02, 2.52, 3.414, 5.124, 6.0154, 7.05, 9.025,  
12.032, 3.045, 5.04, 6.25, 3.0174, 'http://spinner.bl.ee/images/MPOKKA.png'),  
(18, 'NHP', 0.325, '2014-01-17 01:38:56', 3.17, 4.36, 5.21, 3.214, 3.258, 8.25, 9.52, 5.014, 6.98, 9.58, 6.05,  
1.023, 0.015, 9.05, 4.02, 'http://spinner.bl.ee/images/NHP.png'),  
(19, 'OTE', 11.46, '2014-01-17 01:39:25', 12.76, 13.56, 18.96, 25.3, 100.24, 96.52, 75.301, 56.056, 45.05,  
63.012, 47.512, 85.05, 96.201, 63.05, 20.045, 'http://spinner.bl.ee/images/OTE.png'),  
(20, 'OPAP', 10.54, '2014-01-17 01:39:35', 11.77, 47.56, 35.21, 10.26, 3.54, 8.645, 19.24, 15.023, 14.06, 17.056,  
22.01, 29.065, 35.026, 45.01, 68.025, 'http://spinner.bl.ee/images/OPAP.png'),  
(21, 'ATOX', 5.246, '2014-01-17 02:05:39', 218.054, 521.36, 410.23, 300.214, 10.24, 95.32, 75.396, 21.056,  
45.065, 35.054, 88.025, 66.045, 84.02, 63.021, 84.253, 'http://spinner.bl.ee/images/ATOX.png'),  
(22, 'ALOXC', 2.69, '2014-03-17 17:52:33', 5.36, 9.214, 3.25, 0.024, 96.25, 8.14, 5.201, 2.02, 6.01, 6.0145, 5.041,  
3.021, 1.02, 36.025, 56.02, 'http://spinner.bl.ee/images/ALOXC.png'),  
(23, 'HAVOK', 69.24, '2014-03-17 20:52:57', 5.36, 75.14, 52.36, 4.21, 0.029, 8.65, 7.302, 3.05, 1.02, 5.301, 6.05,  
4.05, 5.23, 9.041, 7.05, 'http://spinner.bl.ee/images/HAVOK.png'),  
(24, 'VENDOK', 5.39, '2014-03-17 20:52:57', 8.201, 9.35, 4.05, 5.24, 9.25, 16.88, 8.936, 5.63, 4.05, 3.33, 6.05,  
8.05, 7.056, 9.99, 10.54, 'http://spinner.bl.ee/images/VENDOK.png');
```

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

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

```

1 <?php
2 try {
3     $handler = new PDO('mysql:host=mysql.hostinger.gr;dbname=u749704378_stock', 'u749704378_metox', 'kostas123');
4     $handler->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
5 } catch (Exception $e) {
6     echo $e->getMessage();
7     die();
8 }
9
10 $query = $handler->query('SELECT * FROM metoxes');
11 $records = array();
12 $records = $query->fetchAll(PDO::FETCH_ASSOC);
13 $json['metoxes'] = $records;
14 echo json_encode($json);

```

Εικόνα 36: Κώδικας PHP για export σε JSON πίνακα

Χρησιμοποίησα PDO(php data objects) αντικειμενοστραφής php για να συνδεθώ στην βάση δεδομένων και να πιάσω τα Exceptions. Χρησιμοποίησα την μέθοδο query() για να κάνω το «ερώτημα» στην βάση δεδομένων και να σθλλέξω όλα τα δεδομένα απο τον συγκεκριμένο πίνακα φέρνοντας τα σαν Associative array. Τα κάνουμε encode σε json array ονομαζόμενο metoxes και μας προκύπτει το παρακάτω αποτέλεσμα στον browser.

### 3.1.3 Exporting Data in JSON Array into the browser

```

["metoxes":[{"id":"1","name":"AAAK","price":"2.81","datecreated":"2014-01-17
01:27:52","price1":"3.33","price2":"2.65","price3":"6.21","price4":"0.021","price5":"4.89","price6":"10.58","price7":"6.21","price8":"7.25","price9":"6.147","price10":"0.0254","price11":"6.88","price12":"9.21","price13":"11.25","price14":"10.93","price15":"11.01","image":"http://spinner.bl
[id":"2","name":"ABE","price":"0.341","datecreated":"2014-01-17
01:29:50","price1":"0.389","price2":"0.17","price3":"0.98","price4":"3.15","price5":"3.174","price6":"0.258","price7":"6.25","price8":"88.025","price9":"63.04","price10":"2.0147","price11":"3.25","price12":"9.01","price13":"4.102","price14":"8.025","price15":"70.95","image":"http://spinner.bl
[id":"3","name":"ALPHA","price":"1.586","datecreated":"2014-01-17
01:30:29","price1":"0.721","price2":"0.698","price3":"0.658","price4":"0.632","price5":"0.025","price6":"3.24","price7":"1.201","price8":"9.41","price9":"41.025","price10":"3.025","price11":"9.02","price12":"8.056","price13":"4.23","price14":"4.052","price15":"2.01","image":"http://spinner.b
[id":"4","name":"AKPIT","price":"0.115","datecreated":"2014-01-17
01:31:00","price1":"0.025","price2":"0.47","price3":"1.025","price4":"3.214","price5":"0.369","price6":"2.041","price7":"3.147","price8":"6.214","price9":"3.025","price10":"9.024","price11":"6.025","price12":"6.014","price13":"5.02","price14":"3.01","price15":"4.12","image":"http://spinner.bl
[id":"5","name":"ANEKI","price":"0.118","datecreated":"2014-01-17
01:31:26","price1":"0.119","price2":"14.2","price3":"7.89","price4":"5.64","price5":"3.28","price6":"2.21","price7":"1.01","price8":"0.025","price9":"3.012","price10":"6.012","price11":"4.05","price12":"2.13","price13":"6.144","price14":"9.04","price15":"10.54","image":"http://spinner.bl/ee/u
[id":"6","name":"ATE","price":"0.155","datecreated":"2014-01-17
01:32:03","price1":"0.155","price2":"0.254","price3":"0.365","price4":"0.147","price5":"0.21","price6":"1.02","price7":"2.54","price8":"5.22","price9":"9.32","price10":"7.01","price11":"8.23","price12":"8.89","price13":"4.32","price14":"10.201","price15":"3.214","image":"http://spinner.bl/ee/ima
[id":"7","name":"ATEKI","price":"2.14","datecreated":"2014-01-17
01:32:15","price1":"1.9","price2":"2.4","price3":"5.684","price4":"3.214","price5":"0.089","price6":"4.85","price7":"7.12","price8":"4.021","price9":"6.023","price10":"3.12","price11":"6.14","price12":"5.14","price13":"8.201","price14":"9.01","price15":"7.01","image":"http://spinner.bl/ee/ima
[id":"8","name":"ATT","price":"0.248","datecreated":"2014-01-17
01:32:38","price1":"0.216","price2":"0.987","price3":"1.002","price4":"6.25","price5":"4.25","price6":"9.521","price7":"15.03","price8":"11.02","price9":"52.014","price10":"3.14","price11":"9.012","price12":"12.36","price13":"8.96","price14":"16.201","price15":"22.36","image":"http://spinn
[id":"9","name":"ATTIK","price":"0.028","datecreated":"2014-01-17
01:32:49","price1":"0.67","price2":"0.365","price3":"2.35","price4":"0.254","price5":"1.254","price6":"3.214","price7":"0.024","price8":"0.014","price9":"6.54","price10":"7.25","price11":"6.301","price12":"10.25","price13":"14.55","price14":"30.01","price15":"15.02","image":"http://spinner.b
[id":"10","name":"ATTIKA","price":"0.76","datecreated":"2014-01-17
01:33:18","price1":"0.356","price2":"0.854","price3":"1.02","price4":"4.521","price5":"6.21","price6":"7.98","price7":"4.321","price8":"25.01","price9":"3.02","price10":"4.025","price11":"2.66","price12":"9.55","price13":"4.01","price14":"3.33","price15":"2.52","image":"http://spinner.bl/ee/
[id":"11","name":"AXOX","price":"0.139","datecreated":"2014-01-17
01:33:51","price1":"1.658","price2":"0.254","price3":"2.143","price4":"3.214","price5":"8.25","price6":"12.021","price7":"3.21","price8":"11.02","price9":"3.66","price10":"6.563","price11":"4.201","price12":"3.2","price13":"1.025","price14":"3.98","price15":"7.85","image":"http://spinner.bl/ee
[id":"12","name":"BINTA","price":"3.66","datecreated":"2014-01-17
01:34:06","price1":"0.548","price2":"6.201","price3":"4.052","price4":"30.21","price5":"25.14","price6":"4.14","price7":"0.254","price8":"6.2","price9":"3.21","price10":"6.044","price11":"6.22","price12":"3.147","price13":"3.05","price14":"6.21","price15":"3.02","image":"http://spinner.bl/ee/ii
[id":"13","name":"BOF","price":"0.177","datecreated":"2014-01-17
01:34:21","price1":"6.25","price2":"0.998","price3":"1.254","price4":"3.336","price5":"4.563","price6":"7.88","price7":"9.987","price8":"9.201","price9":"4.015","price10":"12.36","price11":"5.124","price12":"6.047","price13":"9.203","price14":"11.201","price15":"32.05","image":"http://spinn
[id":"14","name":"BOXX","price":"1.3","datecreated":"2014-01-17
01:34:39","price1":"2.18","price2":"0.214","price3":"0.21","price4":"6.214","price5":"8.963","price6":"5.214","price7":"0.0214","price8":"6.014","price9":"2.02","price10":"20.11","price11":"18.52","price12":"17.63","price13":"12.01","price14":"13.054","price15":"18.66","image":"http://spinn
[id":"15","name":"BPAX","price":"1.146","datecreated":"2014-01-17
01:36:01","price1":"0.069","price2":"0.089","price3":"1.023","price4":"4.521","price5":"3.62","price6":"4.89","price7":"7.25","price8":"10.23","price9":"5.14","price10":"3.215","price11":"9.154","price12":"5.15","price13":"6.045","price14":"7.04","price15":"65.02","image":"http://spinner.bl/ee
[id":"16","name":"IATP","price":"0.85","datecreated":"2014-01-17
01:36:19","price1":"1.29","price2":"4.36","price3":"5.21","price4":"3.214","price5":"9.521","price6":"7.321","price7":"6.25","price8":"1.025","price9":"6.05","price10":"4.045","price11":"7.523","price12":"9.02","price13":"10.053","price14":"6.457","price15":"3.04","image":"http://spinner.bl/ee
[id":"17","name":"MPOKKA","price":"0.055","datecreated":"2014-01-17
01:37:12","price1":"0.5874","price2":"0.24","price3":"0.147","price4":"1.02","price5":"2.52","price6":"3.414","price7":"5.124","price8":"6.0154","price9":"7.05","price10":"9.025","price11":"12.032","price12":"3.045","price13":"5.04","price14":"6.25","price15":"3.0174","image":"http://spinn
[id":"18","name":"NHP","price":"0.325","datecreated":"2014-01-17
01:38:56","price1":"1.77","price2":"4.36","price3":"5.21","price4":"3.214","price5":"3.258","price6":"8.25","price7":"9.521","price8":"5.014","price9":"6.98","price10":"9.58","price11":"6.05","price12":"1.023","price13":"0.015","price14":"9.05","price15":"4.02","image":"http://spinner.bl/ee/ii
[id":"19","name":"OPAX","price":"5.246","datecreated":"2014-01-17
01:39:25","price1":"12.76","price2":"13.56","price3":"18.96","price4":"4.521","price5":"10.24","price6":"96.52","price7":"75.302","price8":"75.301","price9":"56.056","price10":"63.012","price11":"47.512","price12":"85.05","price13":"96.201","price14":"84.02","price15":"20.045","image":"http://
[id":"20","name":"OPAP","price":"10.54","datecreated":"2014-01-17
01:39:35","price1":"11.77","price2":"4.56","price3":"5.21","price4":"10.26","price5":"3.54","price6":"8.645","price7":"19.24","price8":"15.023","price9":"14.06","price10":"17.056","price11":"22.01","price12":"29.065","price13":"35.026","price14":"45.01","price15":"68.025","image":"http://
[id":"21","name":"ATOX","price":"5.246","datecreated":"2014-01-17
02:05:39","price1":"218.034","price2":"521.36","price3":"410.23","price4":"300.214","price5":"10.24","price6":"95.32","price7":"75.396","price8":"21.056","price9":"45.065","price10":"35.054","price11":"88.025","price12":"66.045","price13":"84.02","price14":"63.021","price15":"84.253","au
[id":"22","name":"ALOXOX","price":"2.69","datecreated":"2014-01-17
17:53:33","price1":"3.26","price2":"9.214","price3":"3.25","price4":"0.024","price5":"96.25","price6":"8.14","price7":"5.201","price8":"2.02","price9":"6.01","price10":"6.0145","price11":"5.041","price12":"3.021","price13":"1.02","price14":"36.025","price15":"56.02","image":"http://spinner.b
[id":"23","name":"HAXOKK","price":"69.24","datecreated":"2014-03-17
20:52:57","price1":"5.36","price2":"7.14","price3":"52.36","price4":"4.21","price5":"0.029","price6":"8.65","price7":"7.302","price8":"3.05","price9":"1.02","price10":"5.301","price11":"6.05","price12":"4.05","price13":"5.23","price14":"9.041","price15":"7.05","image":"http://spinner.bl/ee/ii
[id":"24","name":"VENDOK","price":"5.39","datecreated":"2014-03-17
20:52:57","price1":"8.201","price2":"9.35","price3":"4.05","price4":"5.24","price5":"9.25","price6":"16.88","price7":"8.936","price8":"5.63","price9":"4.05","price10":"3.33","price11":"6.05","price12":"8.05","price13":"7.056","price14":"9.99","price15":"10.54","image":"http://spinner.bl/ee/ii

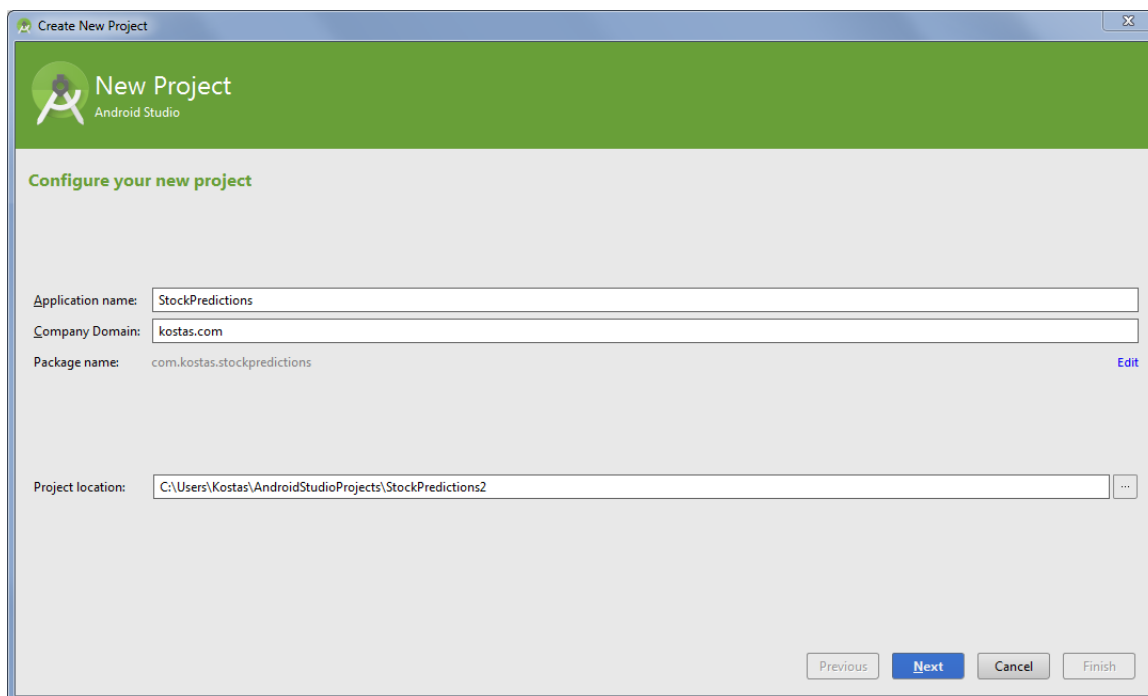
```



Εικόνα 37: Πίνακας δεδομένων JSON

## 3.2 Διμιουργία Project στο Android Studio

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

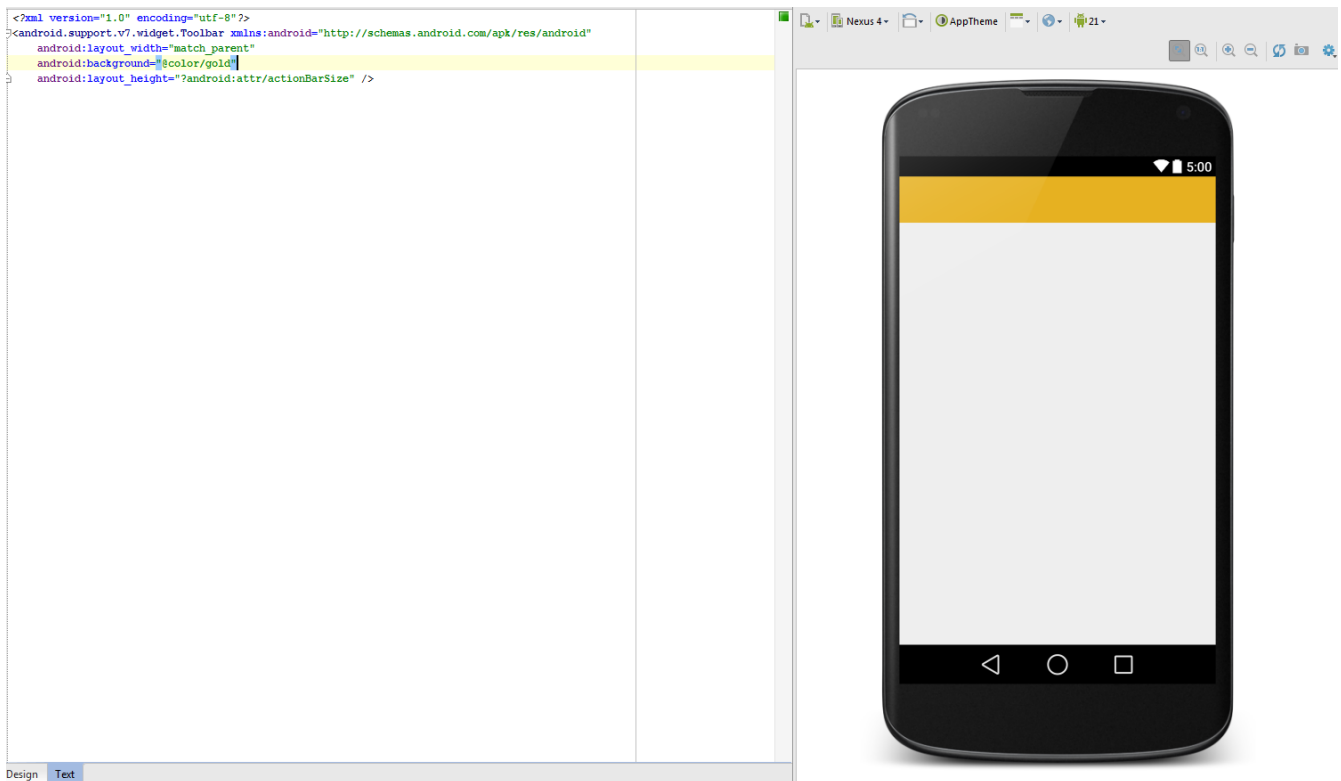


Εικόνα 38: Διμιουργία νέου πρότζεκτ

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



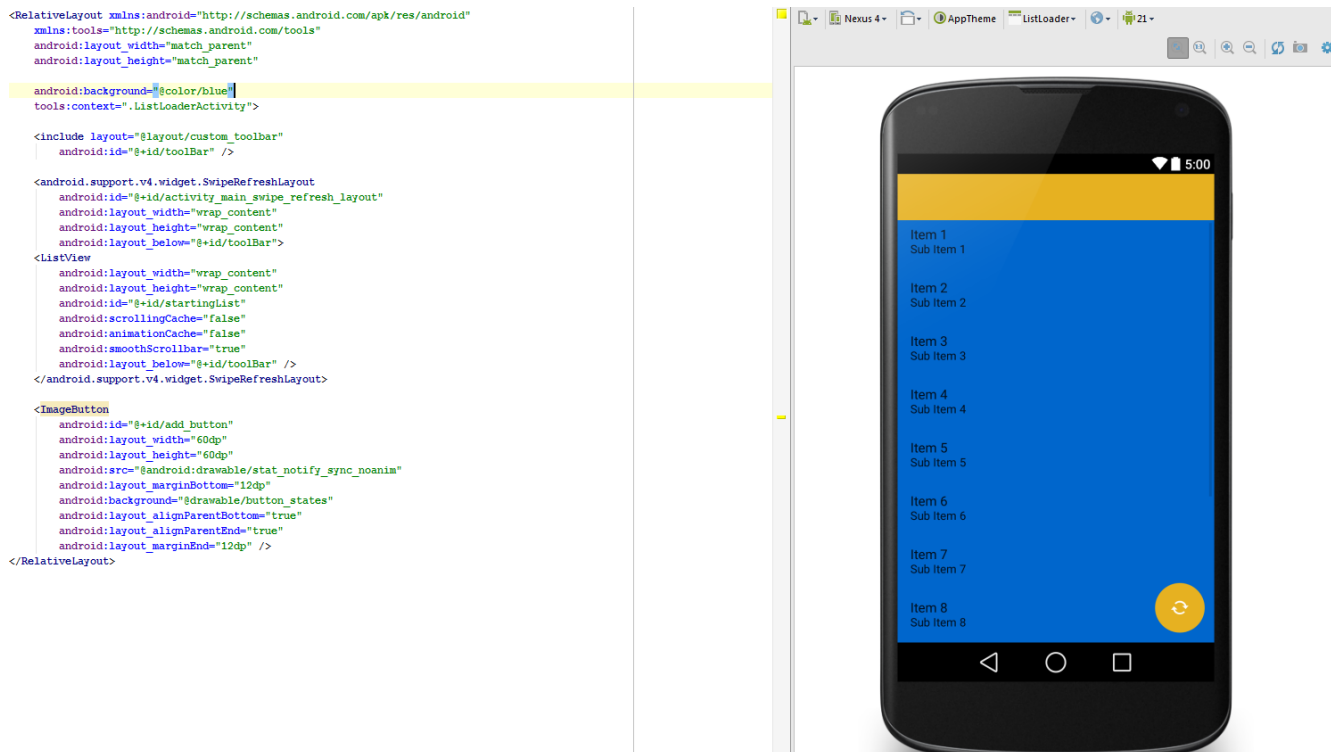




Εικόνα 40: Διμιουργία custom Toolbar

Η μπάρα που διμιουργήσαμε θα περιέχει το όνομα της εφαρμογής στην πρώτη οθόνη καθώς και κάποιες πληροφορίες ανάλογα την μετοχή που θα επιλέγει ο χρήστης και επίσης θα περιέχει το Menu που θα περιέχει επίσης η πρώτη οθόνη.

Βεβαίως πρέπει να διμιουργήσουμε και το layout αρχείο που θα χρησιμοποιήσει η πρώτη οθόνη δηλαδή το πως θα φαίνεται στον χρήστη.



Εικόνα 41: Layout πρώτης οθόνης

### 3.4 Δημιουργία κώδικα πρώτης οθόνης

Επειδή φορτώνουμε δεδομένα απο το Ιντερνετ θα πρέπει να έχουμε τα εξής permissions στο Manifest.xml.

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

Βεβαίως θα πρέπει να ελέγξουμε για το αν η συσκευή είναι συνδεδεμένη στο Ιντερνετ πράγμα τα οποίο επιτυγχάνεται με τον παρακάτω κώδικα.

```
ConnectivityManager cm = (ConnectivityManager) getSystemService(CONNECTIVITY_SERVICE);
NetworkInfo activeNetwork = cm.getActiveNetworkInfo();
boolean network_connected = activeNetwork != null && activeNetwork.isAvailable() && activeNetwork.isConnectedOrConnecting();
if (!network_connected) {
    onDetectNetworkState().show();
} else {
    if (activeNetwork.getType() == ConnectivityManager.TYPE_WIFI) {
        accessWebService();
        registerCallClickBack();
        mSwipeRefreshLayout.setOnRefreshListener(() -> {
            accessWebService();
            mSwipeRefreshLayout.setRefreshing(false);
        });
    }
    if (activeNetwork.getType() == ConnectivityManager.TYPE_MOBILE) {
        SharedPreferences settings = getSharedPreferences(PREFS_NAME, 0);
        String skipMessage = settings.getString("skipMessage", "NOT checked");
        if (!skipMessage.equals("checked")) {
            onAlertMobileData().show();
        }
        if (skipMessage.equals("checked")) {
            accessWebService();
            registerCallClickBack();
            mSwipeRefreshLayout.setOnRefreshListener(() -> {
                accessWebService();
                mSwipeRefreshLayout.setRefreshing(false);
            });
        }
    }
}
}
```

Εικόνα 42: Κώδικας για έλεγχο σύνδεσης της συσκευής στο διαδίκτυο

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

```

private AlertDialog onAlertMobileData() {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    LayoutInflater adbInflater = LayoutInflater.from(this);
    View eulaLayout = adbInflater.inflate(R.layout.checkbox, null);
    dontShowAgain = (CheckBox) eulaLayout.findViewById(R.id.skip);
    builder.setTitle("Connect via mobile networks")
        .setView(eulaLayout)
        .setMessage("Wi-Fi is not available. Connecting to mobile networks m...")
        .setNegativeButton("Cancel",
            (dialog, which) -> {
                String checkBoxResult = "NOT checked";
                if (dontShowAgain.isChecked())
                    checkBoxResult = "checked";
                SharedPreferences settings = getSharedPreferences(
                    PREFS_NAME, 0);
                SharedPreferences.Editor editor = settings
                    .edit();
                editor.putString("skipMessage", checkBoxResult);
                // Commit the edits!
                editor.commit();
                ListLoaderActivity.this.finish();
            })
        .setPositiveButton("Ok",
            (dialog, which) -> {
                String checkBoxResult = "NOT checked";
                if (dontShowAgain.isChecked())
                    checkBoxResult = "checked";
                SharedPreferences settings = getSharedPreferences(
                    PREFS_NAME, 0);
                SharedPreferences.Editor editor = settings
                    .edit();
                editor.putString("skipMessage", checkBoxResult);
                editor.commit();
                accessWebService();
                registerCallClickBack();
            });
    return builder.create();
}

```

Εικόνα 43: Dialog για δεδομένα κινητής τηλεφωνίας

Βεβαίως θα πρέπει να προγραμματίσουμε τι θα κάνει η εφαρμογή για το όταν ο χρήστης κάνει περιστροφή οθόνης. Αν θυμάστε σύμφωνα με τον κύκλο ζωής της εφαρμογής η εφαρμογή ξαναδμιουργείται σε κάθε περιστροφή της οθόνης γιαυτό θα πρέπει να το περιορίσουμε και αυτό επιτυγχάνεται με τον παρακάτω κώδικα.

```

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

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

```

Εικόνα 44: Μέθοδος για να μην έχουμε επαναδιμουργία της εφαρμογής σε κάθε περιστροφή οθόνης

Βεβαίως εφόσον έχουμε ελέγξει τα πάντα είμαστε έτοιμοι να φτιάξουμε την μέθοδο που θα μας επιτρέψει να φορτώσουμε τα δεδομένα σε μορφή json απο το ιντερνετ. Αυτό θα γίνει με την χρήση AsyncTask η οποία είναι μια μέθοδος ασύγχρονης διεργασίας για φόρτωση δεδομένων.

```

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

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

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

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

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

} // end async task

public void accessWebService() {...}

public void ListDrawer() {...}

```

Εικόνα 45: AsyncTask για φόρτωση δεδομένων

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

Βεβαίως μόλις φέρουμε τα δεδομένα απο τον σερβερ θα πρέπει να τα περάσουμε στην λίστα μας. Γιατο θα πρέπει να δημιουργήσουμε κάποιες κλάσεις και μεθόδους για βοήθεια.

```
public StockList(String stockCurrentName, String stockCurrentPrice, String stockImage, String[] restPrices) {
    this.stockCurrentName = stockCurrentName;
    this.stockCurrentPrice = stockCurrentPrice;
    this.stockImage = stockImage;
    this.restPrices = restPrices;
}
```

Εικόνα 46: Τι θα περιέχει το κάθε αντικείμενο της λίστας

Συγκεκριμένα θα περιέχει 3 Strings και ενα πίνακα απο Strings που θα είναι οι 15 τιμές σε μορφή String.

Όλα αυτά τα περνάμε στην λίστα μας η οποία είναι ενας δυναμικός πίνακας τύπου StockList.

```
public void ListDrawer() {
    customList = new ArrayList<StockList>();
    try {
        JSONObject jsonResponse = new JSONObject(jsonResult);
        JSONArray jsonMainNode = jsonResponse.optJSONArray("metoxes");
        for (int i = 0; i < jsonMainNode.length(); i++) {
            JSONObject jsonChildNode = jsonMainNode.getJSONObject(i);
            name = jsonChildNode.optString("name");
            price = jsonChildNode.optString("price");
            price1 = jsonChildNode.optString("price1");
            price2 = jsonChildNode.optString("price2");
            price3 = jsonChildNode.optString("price3");
            price4 = jsonChildNode.optString("price4");
            price5 = jsonChildNode.optString("price5");
            price6 = jsonChildNode.optString("price6");
            price7 = jsonChildNode.optString("price7");
            price8 = jsonChildNode.optString("price8");
            price9 = jsonChildNode.optString("price9");
            price10 = jsonChildNode.optString("price10");
            price11 = jsonChildNode.optString("price11");
            price12 = jsonChildNode.optString("price12");
            price13 = jsonChildNode.optString("price13");
            price14 = jsonChildNode.optString("price14");
            price15 = jsonChildNode.optString("price15");

            image = jsonChildNode.optString("image");

            justPrices = new String[]{price1, price2,
                price3, price4, price5, price6, price7, price8, price9,
                price10, price11, price12, price13, price14, price15};

            customList.add(new StockList(name, price, image, justPrices));
        }
    } catch (Exception e) {
        Intent intent1 = new Intent(ListLoaderActivity.this,
            RefreshActivity.class);
        startActivity(intent1);
        ListLoaderActivity.this.finish();
    }

    ArrayAdapter adapter = new MyStocksAdapter(ListLoaderActivity.this, R.layout.list_item, customList);
    adapter.notifyDataSetChanged();
    startList.setAdapter(adapter);
}
```



## Εικόνα 47: Μέθοδος για την δημιουργία της λίστας

Θα πρέπει όμως να φτιάξουμε το πώς θα περνιόνται τα δεδομένα στην λίστα και αυτό είναι δουλειά του αντάπτορα μας

```
public class MyStocksAdapter extends ArrayAdapter<StockList> {  
  
    public MyStocksAdapter(Context context, int resource, List<StockList> objects) {  
        super(context, resource, objects);  
    }  
  
    @Override  
    public View getView(int position, View convertView, ViewGroup parent) {  
  
        ViewHolderItems holder;  
        View stockView = null;  
        StockList current = null;  
  
        if (stockView == null) {  
            LayoutInflater li = LayoutInflater.from(getContext());  
            stockView = li.inflate(R.layout.list_item, parent, false);  
            current = getItem(position);  
  
            holder = new ViewHolderItems();  
            holder.stockViewName = (TextView) stockView.findViewById(R.id.stock_name);  
            holder.stockViewPrice = (TextView) stockView.findViewById(R.id.stock_price);  
            holder.stockViewImage = (ImageView) stockView.findViewById(R.id.imagestartinglist);  
            stockView.setTag(holder);  
        } else {  
            holder = (ViewHolderItems) stockView.getTag();  
        }  
  
        if (current != null) {  
            holder.stockViewName.setText(current.getStockCurrentName());  
            holder.stockViewPrice.setText(current.getStockCurrentPrice());  
            Ion.with(holder.stockViewImage).placeholder(R.drawable.ic_chat).error(R.drawable.ic_chat).load(current.getStockImage());  
        }  
  
        return stockView;  
    }  
  
    static class ViewHolderItems {  
        TextView stockViewName, stockViewPrice;  
        ImageView stockViewImage;  
    }  
}
```

## Εικόνα 48: Adapter Class

Και η τελευταία μέθοδος για την πρώτη οθόνη θα είναι η μέθοδος που θα δηλώσουμε τι θα γίνεται και ποιά δεδομένα θα περάσουμε στην επόμενη οθόνη.

```

private void registerCallClickBack() {
    startList.setOnItemClickListener((parent, viewClicked, position, id) -> {
        tv1 = (TextView) viewClicked.findViewById(R.id.stock_name);
        tv2 = (TextView) viewClicked.findViewById(R.id.stock_price);
        Intent intent = new Intent(ListLoaderActivity.this, StockItem.class);
        intent.putExtra("name", tv1.getText().toString());
        intent.putExtra("price", tv2.getText().toString());
        intent.putExtra("stockInfo", customList.get(position).getRestPrices());
        intent.putExtra("stockImage", customList.get(position).getStockImage());
        startActivity(intent);
        Log.v("hello", "intent successful");
        overridePendingTransition(R.anim.slide_in_right, R.anim.slide_out_right);
    });
    fab = (ImageButton) findViewById(R.id.add_button);

    startList.setOnScrollListener(new AbsListView.OnScrollListener() {
        @Override
        public void onScrollStateChanged(AbsListView view, int scrollState) {
            if(scrollState == SCROLL_STATE_FLING || scrollState == SCROLL_STATE_TOUCH_SCROLL){
                Ion.getDefault(ListLoaderActivity.this).cancelAll(ListLoaderActivity.this);
            }
        }

        @Override
        public void onScroll(AbsListView view, int firstVisibleItem, int visibleItemCount, int totalItemCount) {
        }
    });

    fab.setOnClickListener((v) -> { accessWebService(); });
}

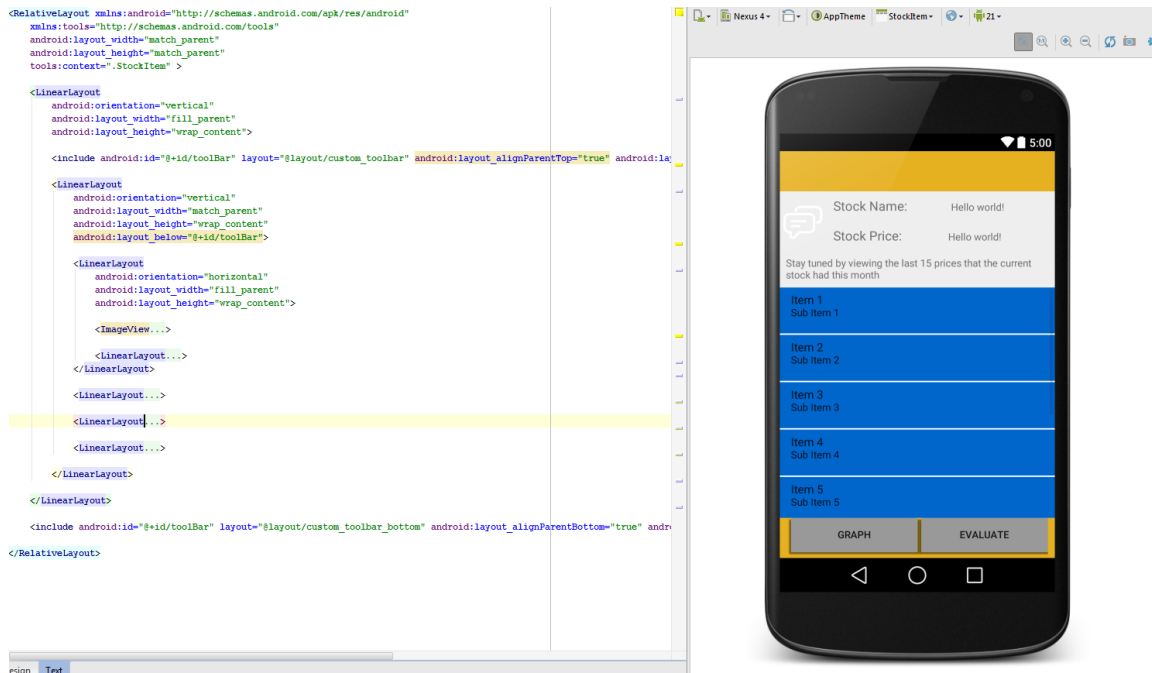
```

Εικόνα 50: Μέθοδος που μας μεταφέρει στην δευτερη οθόνη

Εδώ χρησιμοποιώντας τα Intent για τα οποία μιλήσαμε στο κεφάλαιο 2 θα μεταφέρουμε τον πίνακα απο Strings καθώς και το όνομα και την τωρινή τιμή της μετοχής που κλικάρουμε στην επόμενη οθόνη.

### 3.5 Διμιουργία εμφανισιακού περιεχομένου δευτερης οθόνης

Εδώ θα πρέπει να διμιουργήσουμε το περιεχόμενο της οθόνης που θα εμφανίζεται μόλις ο χρήστης κλικάρει πάνω σε μια μετοχή απο την λίστα μας. Θα χρειαστεί να σχεδιάσουμε το .xml layout μας ως εξής:



Εικόνα 51: Δημιουργία της δευτερης οθόνης

### 3.6 Δημιουργία κώδικα της δεύτερης οθόνης

Στην δευτερη οθόνη εφόσον έχουμε περάσει τα δεδομένα απο την πρώτη οθόνη απλώς τα τοποθετούμε στην λίστα μας και φτιάχνουμε μια μέθοδο για να κάνουν κατι τα κουμπιά όταν τα πατάει ο χρήστης. Το κουμπί στα αριστερά θα φτιάχνει ζωντανό γράφημα βασισμένο στις 15 τελευταίες τιμές της μετοχής και το κουμπί στα δεξιά θα κάνει την εκτίμηση της μετοχής και θα ενημερώνει τον χρήστη αν υπάρχει άνοδος ή κάθοδος.

```

private void populateBottomToolBar() {
    Button btngraph = (Button) findViewById(R.id.toolBar_button1);
    Button btnevaluate = (Button) findViewById(R.id.toolBar_button2);

    btngraph.setOnClickListener((v) -> {
        Intent intent = new Intent(StockItem.this, LiveGraph.class);
        intent.putExtra("stockprices", myItems);
        startActivity(intent);
    });

    btnevaluate.setOnClickListener((v) -> {
        /*Intent intent = new Intent(StockItem.this, EvaluationActivity.class);
        startActivity(intent);*/
        if(listaFake.isShown()){
            listaFake.setVisibility(View.INVISIBLE);
            LinearLayout ll = (LinearLayout) findViewById(R.id.layout_to_be_shown);
            TextView evalText = (TextView) findViewById(R.id.textViewEvaluation);
            for(int k=0;k<myItems.length;k++){
                sum = sum + Double.parseDouble(myItems[k]);
            }
            TextView evalName = (TextView) findViewById(R.id.textViewEvalName);
            ImageView evalImage = (ImageView) findViewById(R.id.imageViewEval);
            evalName.setText(name);
            avg = sum/myItems.length;
            if(avg > 10){
                evalText.setText(": " + String.valueOf(avg));
                evalText.setTextColor(Color.GREEN);
                evalImage.setImageResource(R.drawable.green);
            }else{
                evalText.setText(": " + String.valueOf(avg));
                evalText.setTextColor(Color.RED);
                evalImage.setImageResource(R.drawable.red);
            }

            ll.setVisibility(View.VISIBLE);
        }
    });
}

```

Εικόνα 52: Μέθοδος υλοποίησης των κουμπιών

Θα χρειαστεί να κάνουμε import την βιβλιοθήκη achartengine η οποία θα μας βοηθήσει για να κάνουμε το γράφημα μας περνώντας τα δεδομένα που θέλουμε σαν όρισμα. Ο κώδικας για την δημιουργία του γραφήματος φαίνεται παρακάτω.

```

public class LiveGraph extends ActionBarActivity {

    private static GraphicalView view;
    private LiveGraphClass live = new LiveGraphClass();
    private static Thread thread;
    static String[] myPrices;
    double point;

    @Override
} protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_live_graph);
    Toolbar tb = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(tb);
    Intent i = this.getIntent();
    myPrices = i.getStringArrayExtra("stockprices");
} thread = (Thread) run() -> {
    for (int i = 0; i < myPrices.length; i++) {
        try {
            Thread.sleep(800);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        Point p = new Point(i, Double.parseDouble(myPrices[i]));
        live.addNewPoints(p);
        view.repaint();
    }
};
thread.start();
}

    @Override
} protected void onStart() {
    // TODO Auto-generated method stub
    super.onStart();
    view = live.getView(this);
    setContentView(view);
}
}

```

Εικόνα 53: Δημιουργία Γραφήματος

Επίσης θα πρέπει να ρυθμίσουμε πως θα φαίνεται η οθόνη του γραφήματος

```

public class LiveGraphClass {

    private GraphicalView view;
    private TimeSeries dataset = new TimeSeries("");
    private XYMultipleSeriesDataset mDataset = new XYMultipleSeriesDataset();
    private XYSeriesRenderer renderer = new XYSeriesRenderer();
    private XYMultipleSeriesRenderer mRenderer = new XYMultipleSeriesRenderer();

}

    public LiveGraphClass(){
        mDataset.addSeries(dataset);
        mRenderer.setAntialiasing(true);
        mRenderer.setFitLegend(true);
        mRenderer.setYAxisMax(150);
        mRenderer.setXAxisMax(15);
        mRenderer.setYAxisMin(0);
        mRenderer.setXAxisMin(0);
        mRenderer.addSeriesRenderer(renderer);
        renderer.setColor(Color.WHITE);
        renderer.setPointStyle(PointStyle.SQUARE);
        renderer.setFillPoints(true);
        renderer.setChartValuesTextSize(13);
        renderer.setShowLegendItem(true);
        mRenderer.setExternalZoomEnabled(true);
        mRenderer.setZoomButtonsVisible(true);
        mRenderer.setBackgroundColor(Color.BLACK);
        mRenderer.setApplyBackgroundColor(true);
        mRenderer.setZoomEnabled(true);
        mRenderer.setYTitle("Prices");
        mRenderer.setXTitle("Day Values");
    }

    public GraphicalView getView(Context context){

        view = ChartFactory.getLineChartView(context, mDataset, mRenderer);
        return view;
    }

    public void addNewPoints(Point p) { dataset.add(p.getX(), p.getY()); }
}

```

Εικόνα 54: Πως θα φαίνεται το γράφημα μας

### 3.7 Διμιουργία Animation εναλλαγή οθονών

Με την βοήθεια του παρακάτω κώδικα διμιούργησα custom animations για εναλλαγή μεταξύ οθονών.

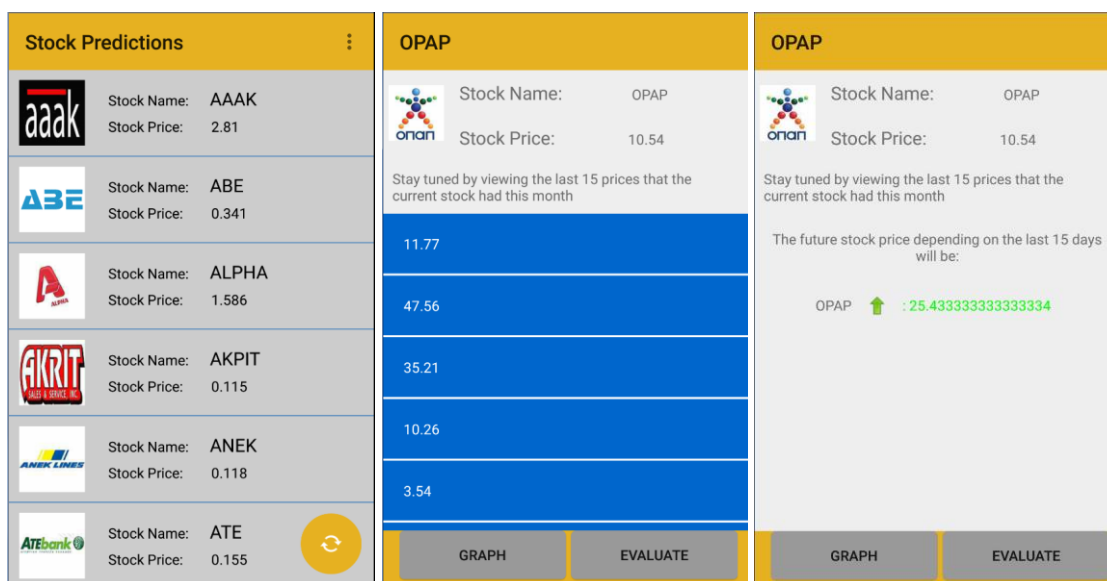
```
<?xml version="1.0" encoding="utf-8"?>
<translate xmlns:android="http://schemas.android.com/apk/res/android"
    android:fromXDelta="0%p"
    android:toXDelta="100%p"
    android:duration="@android:integer/config_longAnimTime"/>

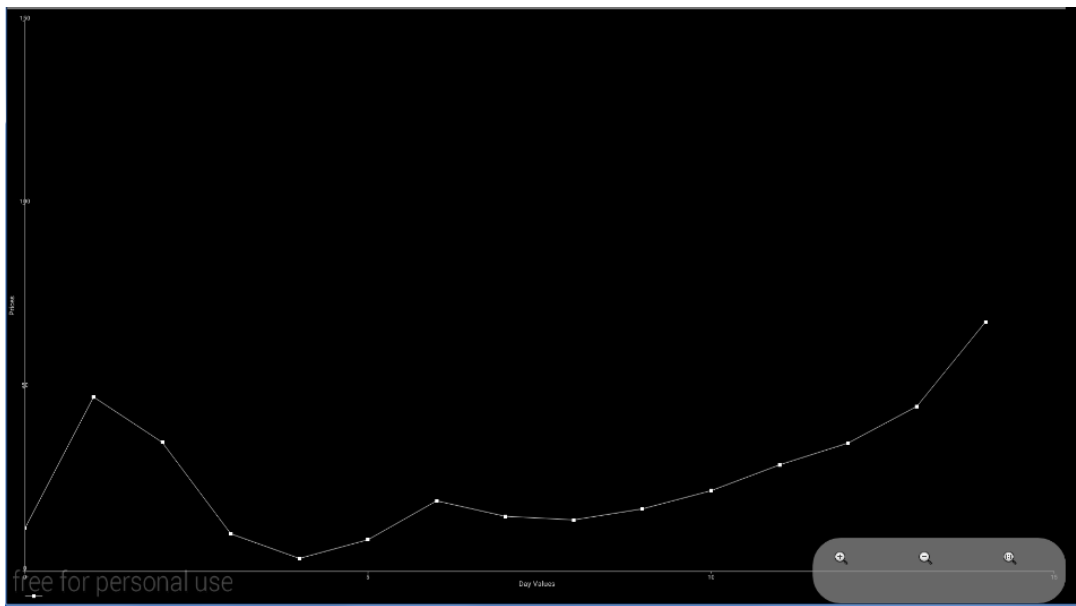
<translate xmlns:android="http://schemas.android.com/apk/res/android"
    android:fromXDelta="100%p"
    android:toXDelta="0%p"
    android:duration="@android:integer/config_longAnimTime"/>
```

Εικόνα 55: Custom Animations

### 3.8 Τελικό Αποτέλεσμα

Η εφαρμογή στον τελικό χρήστη θα φαίνεται κάπως έτσι:





## Κεφάλαιο 4

### 4.1 Μελλοντική Εξέλιξη

Αρχικός στόχος της εφαρμογής είναι η συνεργασία με κάποιο website που θα παρέχει ζωντανές τιμές όλο το 24ώρο. Θα δημιουργηθούν push notifications που θα ενημερώνουν τον χρήστη να βλέπει τις επιλεγμένες του μετοχές ανα πάσα ώρα και στιγμή. Επίσης θα γινόταν και ζωντανό ποντάρισμα στην τιμή της μετοχής μέσω Bitcoins ή Credit/Debit Card.

### 4.2 Συμπεράσματα

Η ανάπτυξη της εφαρμογής Android είναι μια διαδικασία ιδιαίτερα δύσκολη ειδικά όταν θα πρέπει να μάθεις πολλά πράγματα μόνος/μόνη εφόσον δεν συμπεριλαμβάνονται σε μάθημα της σχολής. Θα πρέπει να σου κεντρίσει τον ενδιαφέρον για να ασχοληθείς και θέλει τεράστια προσπάθεια και συμβουλές απο φόρουμ με εμπειρία στον χώρο. Υπήρχαν αρκετές δυσκολίες όσο αναφορά την φόρτωση δεδομένων απο πίνακα json. Παρόλα αυτά όμως στο τέλος βγήκε ενα αρκετά καλό αποτέλεσμα πιστεύω.



## Κεφάλαιο 5

### Βιβλιογραφία

1. J. Friesen, 2010, "Learn Java for Android Development", Apress
2. M. Murphy, 2011, "Android Programming Tutorials, 3rd Edition", CommonsWare
3. C. Hasenan, 2008, "Android Essentials", Firstpress
4. R. Meier, 2010, "Professional Android 2 Application Development", Wrox
5. Design Guidelines <http://www.developers.android.com/design/>
6. Android Studio [Download Page](#)
7. Android Training [Page](#)
8. AchartEngine [Demo an Download Page](#)
9. [Stack Overflow](#), [Android Enthusiasts](#)