



# **ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**

## **ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**

### **ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ**

#### **ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Εφαρμογή Android για την αναγνώριση και καταγραφή  
προϊόντων μέσω barcode**

**Χανιώτης Αλέξανδρος  
ΑΜ: 711141024**

**Εισηγητής: Ιωάννης Βογιατζής, Καθηγητής**

**Δεκέμβριος 2020**

**(Κενό φύλλο)**

## ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Χανιώτης Αλέξανδρος του Γεωργίου, με αριθμό μητρώου 711141024 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών



**(Κενό φύλλο)**

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Θα ήθελα, μέσα από αυτές τις λίγες γραμμές, να ευχαριστήσω όλους όσους με βοήθησαν και με στήριξαν στην εκπόνηση και ολοκλήρωση της διπλωματικής μου εργασίας. Η παρούσα διπλωματική εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο, όπως αυτό της ανάπτυξης εφαρμογής για έξυπνα κινητά τηλέφωνα Android. Η εκπόνηση της αποτέλεσε μια πρόκληση για μένα διότι είναι η βασική προϋπόθεση για την ολοκλήρωση του κύκλου σπουδών μου στο Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών της σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής.

Πρώτα από όλα θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στην οικογένεια μου για την αμείωτη υποστήριξη και συμπαράσταση αλλά και στους καθηγητές μου, κύριο Γεώργιο Μελετίου και κύριο Ιωάννη Βογιατζή, για το ενδιαφέρον, τις υποδείξεις, και τη καθοδήγησή τους κατά τη συγγραφή της εργασίας. Ακόμη, απευθύνω θερμές ευχαριστίες σε όλους τους καθηγητές που συνεργαστήκαμε σε όλα αυτά τα χρόνια της ακαδημαϊκής μου ζωής, για τις γνώσεις και τις εμπειρίες που μου μετέδωσαν και που με διαμόρφωσαν στον άνθρωπο που είμαι σήμερα.

**(Κενό φύλλο)**

## ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική εργασία με τίτλο «Εφαρμογή Android για την αναγνώριση και καταγραφή προϊόντων μέσω barcode» αναπτύχθηκε στα πλαίσια της υποχρεωτικής εκπόνησης διπλωματικής εργασίας από κάθε φοιτητή κατά τη διάρκεια του 8ου (ή μεταγενέστερου) εξαμήνου σπουδών όπως ορίζεται από το πρόγραμμα σπουδών του τμήματος Μηχανικών Πληροφορικής και Υπολογιστών.

Η εφαρμογή μέσω της ικανότητας της να σαρώνει γραμμωτούς κώδικες, αποσκοπεί στην ταχύτερη καταγραφή του προϊόντος καθώς και των χαρακτηριστικών του σε διαχειρίσιμα αρχεία. Ο χρήστης όταν πραγματοποιεί σάρωση ενός γραμμωτού κώδικα, η εφαρμογή συνδέεται σε μια βάση δεδομένων, και όλες οι πληροφορίες του προϊόντος που είναι συνδεδεμένες με τον συγκεκριμένο κωδικό, επιστρέφονται και εμφανίζονται στην οθόνη του χρήστη. Ο ίδιος μπορεί ακόμη να καταγράψει όλα τα προϊόντα που έχει καταγράψει και να τα εξάγει σε διαχειρίσιμο αρχείο excel ή ακόμα και να ενημερώσει παλαιά αρχεία excel με νέα πλέον καταγεγραμμένα προϊόντα. Επιπλέον στον χρήστη δίνεται η ευκαιρία να αποστείλει τον κωδικό που αναγράφεται στο γραμμωτό κώδικα του προϊόντος σε ένα μηχανογραφικό σύστημα, να μοιραστεί τα αρχεία των καταγεγραμμένων προϊόντων μέσω διαφόρων πλατφορμών/εφαρμογών και να αναζητήσει αυτόματα το προϊόν μέσω του κωδικού που σαρώθηκε, στην μηχανή αναζήτησης Google.

Οι γλώσσες προγραμματισμού και οι τεχνολογίες που χρησιμοποιήθηκαν είναι : Java, SQL, php, xml, JSON.

**ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ:** Εφαρμογή για Κινητά Τηλέφωνα, Λειτουργικό Σύστημα Android, Barcodes, Απομακρυσμένη Σύνδεση με Εξυπηρετητή

**ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ:** Android, Barcode, Σάρωση, Αναγνώριση, Καταγραφή, Εφαρμογή

## **ABSTRACT**

The present thesis entitled "Android application for product identification and registration via barcode" was developed in the context of the mandatory thesis preparation by each student during the 8th (or later) semester as defined by the curriculum of the Computer Engineering Department.

The application is intended to make the procedure of product identification faster and safer, due to its main key feature of scanning, decoding barcodes, fetching and recording all of the products characteristics in manageable files. When a user scans a product, the application connects to a remote database, and all the product information that is linked to the specific product code is retrieved from the database and shown to the user. Moreover, he can even record all the products that he scans and export them to an excel file or even update old excel files with newly recorded products. In addition, the user is given the opportunity to send the code of the product to a computer system, to share the files of scanned products and to automatically search product code in the Google search engine.

The programming languages and technologies that were used during this application are: Java, SQL, php, xml, JSON.

**SUBJECT AREA:** Application for Mobile Phones, Android Operating System, Barcodes, Remote Connection to Server

**KEYWORDS:** Android, Barcode, Scan, Decode, Record, Application



## Περιεχόμενα

<b>ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ.....</b>	<b>11</b>
<b>1 Εισαγωγή.....</b>	<b>13</b>
1.1 Αντικείμενο της Διπλωματικής Εργασίας .....	13
1.2 Δομή της Διπλωματικής Εργασίας.....	13
<b>2 Εισαγωγή στο Android .....</b>	<b>15</b>
2.1 Τι εστί Android και πού χρησιμοποιείται .....	15
2.2 Σύγκριση του Android με άλλα λειτουργικά συστήματα για smartphones.....	16
2.2.1 Γιατί Ανάπτυξη σε Android;.....	16
2.3 Οι εκδόσεις του Android .....	18
2.3.1 Εκδόσεις Android 1.0 έως 1.1.....	20
2.3.2 Έκδοση Android 1.5: Cupcake.....	20
2.3.3 Έκδοση Android 1.6: Donut .....	20
2.3.4 Εκδόσεις Android 2.0 έως 2.1: Eclair .....	21
2.3.5 Έκδοση Android 2.2: Froyo .....	21
2.3.6 Έκδοση Android 2.3: Gingerbread .....	22
2.3.7 Εκδόσεις Android 3.0 έως 3.2: Honeycomb .....	22
2.3.8 Έκδοση Android 4.0: Ice Cream Sandwich .....	23
2.3.9 Εκδόσεις Android 4.1 έως 4.3: Jelly Bean.....	23
2.3.10 Έκδοση Android 4.4: KitKat .....	24
2.3.11 Εκδόσεις Android 5.0 και 5.1: Lollipop .....	24
2.3.12 Έκδοση Android 6.0: Marshmallow .....	25
2.3.13 Εκδόσεις Android 7.0 και 7.1: Nougat .....	25
2.3.14 Έκδοση Android 8.0 και 8.1: Oreo .....	26
2.3.15 Έκδοση Android 9: Pie .....	26
2.3.16 Έκδοση Android 10 .....	27
2.3.17 Έκδοση Android 11 .....	28
<b>3 Τι περιλαμβάνει μια εφαρμογή Android. ....</b>	<b>30</b>
3.1 Αρχεία στο Android. ....	30
3.1.1 Το αρχείο AndroidManifest.xml.....	30
3.1.2 Ο φάκελος src (source).....	30
3.1.3 Ο φάκελος res .....	31
3.1.4 Ο φάκελος bin.....	31
3.1.5 Ο φάκελος gen .....	31
3.2 Βασικά Δομικά Μέρη μιας Εφαρμογής Android .....	31
3.2.1 Δραστηριότητες .....	31
3.2.2 Υπηρεσίες .....	33

3.2.3 Πάροχοι Περιεχομένου .....	34
3.2.4 Παραλήπτες Εκπομπών .....	34
<b>4 Προγραμματιστικά Εργαλεία .....</b>	<b>35</b>
4.1 Εισαγωγή .....	37
4.2 Android Studio .....	37
4.3 Android SDK .....	41
4.4 Figma.....	42
<b>5 Γλώσσες Προγραμματισμού που χρησιμοποιήθηκαν.....</b>	<b>45</b>
5.1 Java .....	45
5.1.1 Εισαγωγή .....	45
5.1.2 Ιστορία.....	45
5.1.3 Από την Oak στη Java.....	45
5.1.4 Η εξαγορά από την Oracle και το μέλλον της Java.....	46
5.1.5 Τα χαρακτηριστικά της Java .....	46
5.1.5.1 Η εικονική μηχανή της Java .....	46
5.1.5.2 Ο συλλέκτης απορριμμάτων (Garbage Collector) .....	47
5.1.5.3 Επιδόσεις .....	47
5.1.6 Εργαλεία ανάπτυξης .....	48
5.1.7 Ολοκληρωμένο περιβάλλον ανάπτυξης (IDE).....	48
5.2 SQL (Structured Query Language).....	48
5.2.1 Η ιστορία της SQL .....	48
5.2.2 DML (Data Manipulation Language) DDL (Data Definition Language) DCL (Data Control Language) .....	48
5.2.3 DML Γλώσσα Διαχείρισης Δεδομένων .....	49
5.3 Php .....	50
5.3.1 Ιστορία της PHP .....	50
5.3.2 Τι είναι η PHP .....	50
5.3.3 Πλεονεκτήματα της PHP .....	51
5.3.4 Που Χρησιμοποιείται ένα PHP script.....	52
5.3.5 Τι μπορεί να κάνει η PHP.....	53
5.3.6 Συναρτήσεις.....	53
5.3.6.1 GET.....	53
5.3.6.2 POST.....	53
5.4 Xml .....	53
5.4.1 Ανατομία Διατάξεων Android XML .....	54
<b>6 Ανάπτυξη Συστήματος .....</b>	<b>57</b>
6.1 Φάσεις Ανάπτυξης Συστήματος.....	57
6.1.1 Αρχική Ανάλυση .....	59

6.1.2 Ανάλυση Συστήματος .....	60
6.1.3 Σχεδιασμός Συστήματος .....	60
6.1.3.1 Σχεδίαση Αρχιτεκτονικής.....	61
6.1.3.2 Διεπαφή Χρήστη (UI) .....	63
6.2 Υλοποίηση.....	64
6.2.1 Υλοποίηση Βάσης Δεδομένων.....	64
6.2.2 Υλοποίηση Εφαρμογής.....	67
6.2.2.1 Κλάση Product.....	71
6.2.2.2 Κλάση Company.....	72
6.2.2.3 Manifest.....	73
6.2.2.4 MainActivity.Java .....	74
6.2.2.5 ScanActivity.Java .....	75
6.2.2.6 BarCodeScanFragment.Java.....	76
6.2.2.7 TypeFragment.Java .....	77
6.2.2.8 SettingsActivity.Java.....	78
6.2.2.9 AboutActivity.....	79
6.2.2.7 CompanyFragment.Java .....	80
6.2.2.8 Κλάση FetchDetailsTask.....	81
6.2.3 Σύνδεση της Εφαρμογής με τη Βάση Δεδομένων .....	82
6.2.4 Έλεγχος.....	85
6.2.5 Διανομή .....	85
6.3 Λειτουργία της Εφαρμογής.....	85
6.3.1 Η Δραστηριότητα MainActivity .....	85
6.3.2 Η Δραστηριότητα ScanActivity.....	86
6.3.2.1 Fragment BarCodeScanFragment .....	87
6.3.2.2 Fragment TypeFragment .....	88
6.3.3 Η Δραστηριότητα ResultsActivity .....	89
6.3.4 Η Δραστηριότητα BrowseFilesActivity.....	94
6.3.5 Η Δραστηριότητα SettingsActivity .....	97
6.3.6 Η Δραστηριότητα AboutActivity.....	99
<b>Βιβλιογραφία .....</b>	<b>100</b>

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Το λογότυπο του Android και η εξέλιξή του .....	15
Εικόνα 2: Μεριδία λειτουργικών συστημάτων στην παγκόσμια αγορά από statcounter...	16
Εικόνα 3: Εικονίδια των εκδόσεων του Android .....	18
Εικόνα 4: Χρονολογική σειρά των εκδόσεων του Android .....	19
Εικόνα 5: Το περιβάλλον ανάπτυξης Android Studio .....	35
Εικόνα 6: Εικονική συσκευή με λειτουργικό σύστημα Android.....	36
Εικόνα 7: Αρχική οθόνη Android Studio .....	38
Εικόνα 8: Οθόνη εισαγωγής των αρχικών πληροφοριών της εφαρμογής.....	39
Εικόνα 9: Επιλογή αρχικής Activity .....	40
Εικόνα 10: Κύρια διεπαφή χρήστη .....	41
Εικόνα 11: SDK Manager .....	42
Εικόνα 12: Λογότυπο Figma .....	43
Εικόνα 13: Περιβάλλον χρήστη Figma .....	43
Εικόνα 14: Φάσεις Ανάπτυξης .....	57
Εικόνα 15: Χρονοπρογραμματισμός Έργου.....	58
Εικόνα 16: Διάγραμμα Ροής Φάσεων .....	59
Εικόνα 17: Διάγραμμα ροής εφαρμογής. ....	61
Εικόνα 18: Διάγραμμα βάσης δεδομένων .....	62
Εικόνα 19: Σχεδίαση layouts μέσω Figma .....	63
Εικόνα 20: Αρχική είσοδος στην πλατφόρμα .....	64
Εικόνα 21: Κουμπί δημιουργίας νέας ιστοσελίδας.....	65
Εικόνα 22: Αρχικό μενού μετά την δημιουργία της ιστοσελίδας.....	65
Εικόνα 23: Κουμπί «Manage Website».....	66
Εικόνα 24: Επιλογή rhpMyAdmin.....	66
Εικόνα 25: Πίνακες Βάσης Δεδομένων .....	66
Εικόνα 26: Πίνακας «Company».....	67
Εικόνα 27: Πίνακας «Gtin» .....	67
Εικόνα 28: Όψη Android Studio .....	68
Εικόνα 29: Κύρια διεπαφή χρήστη Android Studio.....	69
Εικόνα 30: Βοηθητικό αρχείο σύνδεσης ιστοσελίδας με συγκεκριμένη βάση δεδομένων 82	
Εικόνα 31: Αρχείο μεταφοράς δεδομένων μεταξύ εφαρμογής και βάσης δεδομένων .....	83
Εικόνα 32: Η δραστηριότητα MainActivity .....	86
Εικόνα 33: Fragment BarCodeScanFragment .....	87
Εικόνα 34: Fragment TypeFragment .....	88
Εικόνα 35: Η δραστηριότητα ResultActivity με τιμή «MODE_DEFAULT» .....	89
Εικόνα 36: Η δραστηριότητα ResultActivity με τιμή «MODE_CHECK_IN» και «MODE_CHECK_OUT».....	90
Εικόνα 37: Dialog επιλογής ποσότητας προς εισαγωγή ή/και αφαίρεση από το αρχείο ..	91
Εικόνα 38: Snackbar επιτυχημένης εισαγωγής και αποτυχημένης εξαγωγής προϊόντος 92	
Εικόνα 39: Αναζήτηση μέσω Google .....	93
Εικόνα 40: Περιηγητής αρχείων με «MODE_EXPLORER» και «MODE_CHECK_IN» ....	95
Εικόνα 41: Περιηγητής αρχείων με τιμή «MODE_CHECK_OUT» .....	95
Εικόνα 42: Διαθέσιμες επιλογές κατά την επιλογή αρχείου .....	96
Εικόνα 43: Η δραστηριότητα SettingsActivity .....	97
Εικόνα 44: Ορισμός φακέλου αποθήκευσης αρχείων.....	98
Εικόνα 45: Η δραστηριότητα AboutActivity .....	99

Εφαρμογή Android για την αναγνώριση και καταγραφή προϊόντων μέσω barcode

# 1 Εισαγωγή

## 1.1 Αντικείμενο της Διπλωματικής Εργασίας

Βασικός στόχος της διπλωματικής εργασίας είναι η υλοποίηση της εφαρμογής Barcode Scanner, σε λειτουργικό σύστημα Android. Πρόκειται για μία εφαρμογή με την οποία ο χρήστης έχει τη δυνατότητα για:

- Σάρωση και αποκωδικοποίηση γραμμωτού κώδικα
- Σύνδεση με οποιαδήποτε επιθυμητή βάση δεδομένων
- Λήψη δεδομένων και πληροφοριών για το προϊόν που αναγνώστηκε
- Αναζήτηση κωδικού προϊόντος σε μηχανή αναζήτησης
- Σύνδεση σε επιθυμητό server για την αυτόματη αποστολή του αποκωδικοποιημένου κωδικού
- Δημιουργία φακέλων και αρχείων excel στα αρχεία του συστήματος της συσκευής
- Αποθήκευση των ληφθέντων δεδομένων του αναγνωσμένου προϊόντος σε αρχείο excel
- Πρόσθεση και αφαίρεση ποσότητας του αναγνωσμένου προϊόντος προς και από αρχεία excel
- Εύκολη διαχείριση/διαμοιρασμός των αρχείων excel μέσω των εγκατεστημένων πλατφορμών της συσκευής
- Εύκολη περιήγηση και αναζήτηση στα αρχεία του συστήματος της κινητής συσκευής
- Η εφαρμογή είναι πλήρως μεταφρασμένη για χρήση σε συσκευή με εγκατεστημένες την Ελληνική και Αγγλική γλώσσα. Προεπιλεγμένη είναι η Αγγλική γλώσσα.

Η εφαρμογή δύναται να λειτουργήσει σε οποιαδήποτε συσκευή που φέρει λειτουργικό Android με έκδοση μεγαλύτερη ή ίση της έκδοσης 5.0 Lollipop.

Ακόμη, στα πλαίσια της παρούσας διπλωματικής, αναλύονται γλώσσες προγραμματισμού, τεχνολογίες και εργαλεία για τον προγραμματισμό σε λειτουργικό Android, καθώς και η αρχιτεκτονική του συστήματος με τα κυριότερα τμήματά της.

## 1.2 Δομή της Διπλωματικής Εργασίας

Η παρούσα διπλωματική εργασία δομείται από 6 κεφάλαια:

Κεφάλαιο 1 – Εισαγωγή: Περικλείει το αντικείμενο της εργασίας και τη συνοπτική δομή αυτής.

Κεφάλαιο 2 – Εισαγωγή στο Android: Εισάγει το χρήστη στο λογισμικό Android, μαζί με μία γενικότερη αναφορά στα υπόλοιπα λειτουργικά συστήματα που χρησιμοποιούνται στα smartphones σήμερα και μια ιστορική αναδρομή στις εκδόσεις του Android.

Κεφάλαιο 3 – Τι περιλαμβάνει μια εφαρμογή Android: Παρουσιάζει στο χρήστη τα βασικότερα δομικά μέρη, στοιχεία και αρχεία μιας εφαρμογής Android.

Κεφάλαιο 4 – Προγραμματιστικά Εργαλεία: Περιλαμβάνει την ανάλυση των εργαλείων που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής.

Κεφάλαιο 5 – Γλώσσες προγραμματισμού που χρησιμοποιήθηκαν: Παρουσιάζει αναλυτικά τα στοιχεία κάθε γλώσσας που χρησιμοποιήθηκε για την υλοποίηση του συστήματος.

Κεφάλαιο 6 – Ανάπτυξη Συστήματος: Περιέχει τις φάσεις ανάπτυξης, την υλοποίηση του συστήματος και την λειτουργία της εφαρμογής.

## 2 Εισαγωγή στο Android

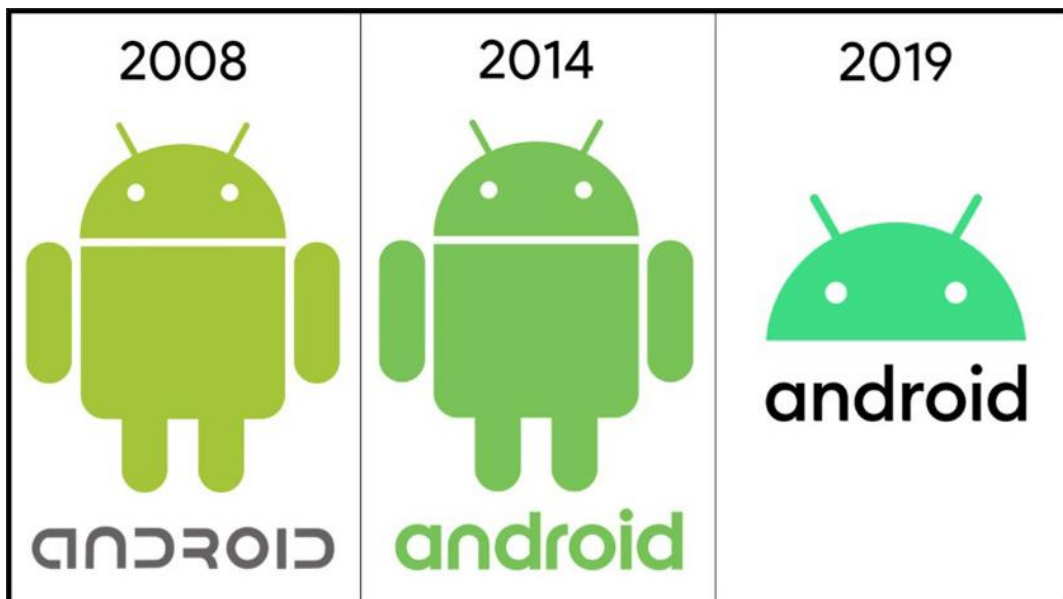
### 2.1 Τι εστί Android και πού χρησιμοποιείται

Το Android είναι ένα λειτουργικό σύστημα για κινητά τηλέφωνα και όχι μόνο. Είναι βασισμένο στο λειτουργικό Linux και αναπτύσσεται από την Google.

Πρόκειται για λογισμικό ανοιχτού κώδικα (Open Source Software - OSS), το οποίο σημαίνει ότι ο πηγαίος κώδικάς του διατίθεται ελεύθερα σε όσους ζητούν να τον εξετάσουν ή να τον τροποποιήσουν για να τον αξιοποιήσουν σε άλλες εφαρμογές. Η γλώσσα προγραμματισμού των εφαρμογών Android είναι η Java ενώ το 2017 προσδέθηκε και η γλώσσα Kotlin ως διαθέσιμη γλώσσα προγραμματισμού εφαρμογών android.

Το Android είναι σχεδιασμένο ώστε να λειτουργεί κυρίως με συσκευές που διαθέτουν οθόνη αφής, όπως τα smartphones και τα tablets. Ωστόσο τα τελευταία χρόνια, χρησιμοποιείται ακόμη και σε κονσόλες παιχνιδιών, ηλεκτρονικούς υπολογιστές, ψηφιακές φωτογραφικές μηχανές, έξυπνα ψυγεία, τηλεοράσεις, ρολόγια, ακόμα και αυτοκίνητα.

Το λογότυπο του Android έχει αλλάξει 2 φορές από την πρώτη εμφάνισή του, όπως παρουσιάζεται στην παρακάτω εικόνα.

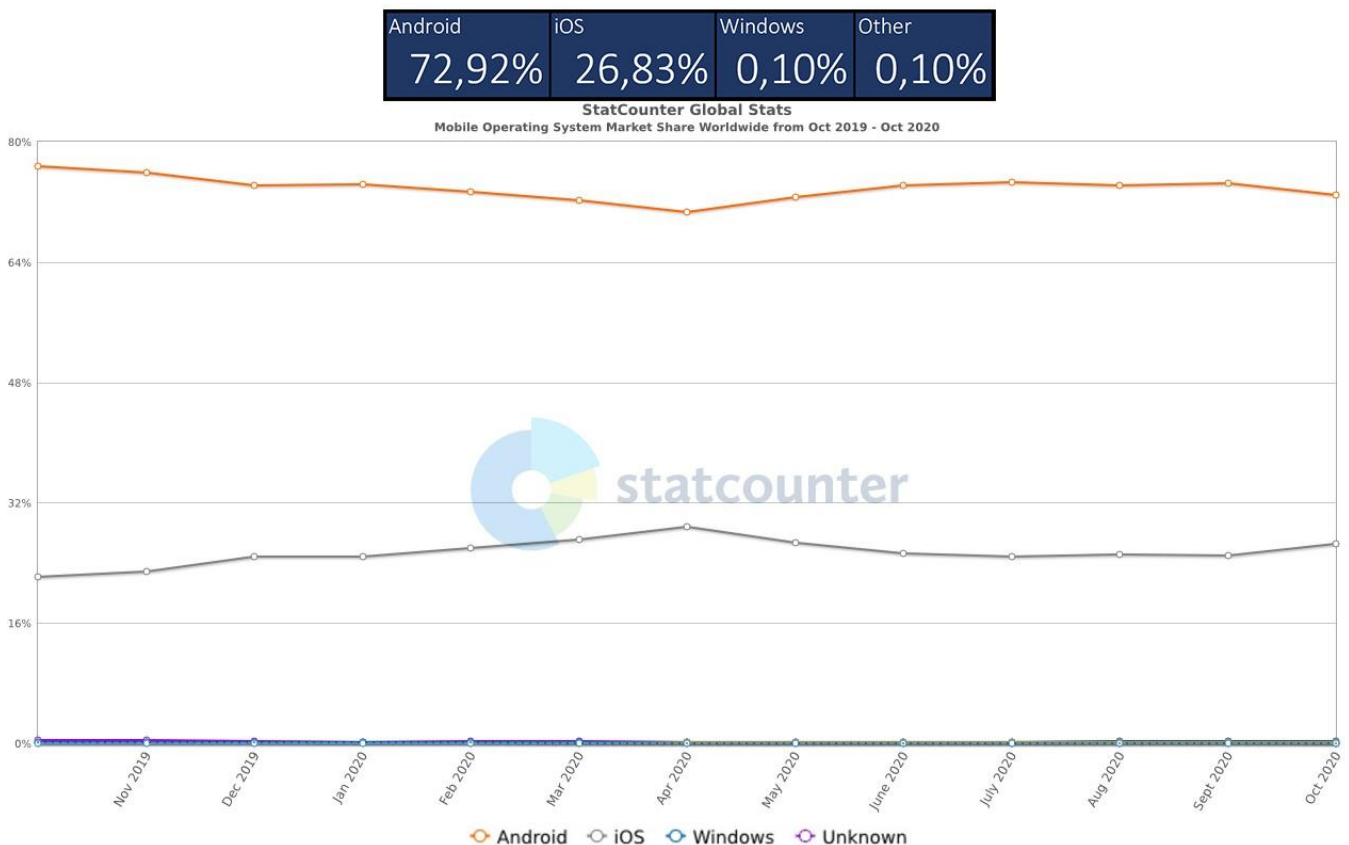


Εικόνα 1: Το λογότυπο του Android και η εξέλιξή του



## 2.2 Σύγκριση του Android με άλλα λειτουργικά συστήματα για smartphones

Αυτή τη στιγμή στην παγκόσμια αγορά χρησιμοποιούνται κατά κύριο λόγο τα εξής λειτουργικά συστήματα στα smartphones: Android (της Google), iOS (της Apple), Windows Phone (της Microsoft) και κάποια άλλα λειτουργικά συστήματα άγνωστα στο ευρύ κοινό. Συγκεκριμένα το μεγαλύτερο μέρος της παγκόσμιας αγοράς κατέχει το Android, με το iOS να ακολουθεί, και τελευταία να έρχονται λειτουργικά συστήματα όπως Windows OS, KaiOS και άλλα άγνωστα συστήματα.



Εικόνα 2: Μεριδία λειτουργικών συστημάτων στην παγκόσμια αγορά από [statcounter.com](https://www.statcounter.com)

### 2.2.1 Γιατί Ανάπτυξη σε Android;

Το Android είναι μια ολοκληρωμένη, ανοιχτή και ελεύθερη πλατφόρμα για κινητά τηλέφωνα που περιλαμβάνει ένα λειτουργικό σύστημα, το απαραίτητο ενδιάμεσο λογισμικό, βιβλιοθήκες και βασικές εφαρμογές. Επιπλέον, παρέχει στους προγραμματιστές όλα τα εργαλεία και APIs για να αρχίσουν να αναπτύσσουν λογισμικό για την πλατφόρμα Android χρησιμοποιώντας τη γλώσσα προγραμματισμού Java. Τα κύρια χαρακτηριστικά του είναι τα παρακάτω:

## **1) Λειτουργικότητα και Ευελιξία**

Το Android εκμεταλλεύεται πλήρως τις δυνατότητες και τα χαρακτηριστικά μιας συσκευής και επιτρέπει την ανάπτυξη εφαρμογών λογισμικού βάσει των χαρακτηριστικών αυτών. Για παράδειγμα, οι προγραμματιστές εφαρμογών μπορούν να υλοποιήσουν εφαρμογές οι οποίες χρησιμοποιούν οποιαδήποτε από τις λειτουργίες της συσκευής όπως η κάμερα, η αποστολή μηνυμάτων, οι τηλεφωνικές κλήσεις, η λήψη φωτογραφιών, το GPS κτλ. Επιπλέον, η πλατφόρμα ανάπτυξης Android δίνει την δυνατότητα στις εφαρμογές να τρέχουν στο παρασκήνιο, και ταυτόχρονα να τρέχει στη συσκευή κάποια άλλη εφαρμογή χωρίς να επηρεαστεί η απόδοσή τους. Το Android είναι μια πλατφόρμα ανοικτού κώδικα, πράγμα που σημαίνει ότι μπορεί εύκολα να επεκταθεί και να τροποποιηθεί για να συμβαδίζει και να υιοθετεί τις τελευταίες τεχνολογίες και εξελίξεις. Το γεγονός ότι και η πηγή της πλατφόρμας είναι ανοικτή διασφαλίζει ότι η ανάπτυξη το Android θα έχει συνεχή πρόοδο και θα εξελίσσεται αφού υπάρχουν αρκετοί επαγγελματίες και ερασιτέχνες προγραμματιστές που ασχολούνται με τη δημιουργία ελευθέρων για χρήση εφαρμογών, βιβλιοθηκών και εργαλείων.

## **2) Πλήρη Παραμετροποίηση**

Μεταξύ των εφαρμογών οι οποίες είναι ενσωματωμένες στη συσκευή από τις εφαρμογές που δημιουργούνται και προστίθενται από τρίτους προγραμματιστές. Ακόμα και οι εφαρμογές από τρίτους μπορούν να έχουν ίδια πρόσβαση σε όλες τις κύριες λειτουργίες της συσκευής κάτι που επιτρέπει στους χρήστες να απολαμβάνουν ένα ευρύ φάσμα εφαρμογών Android που μπορούν να χρησιμοποιηθούν για σχεδόν απεριόριστους σκοπούς. Με συσκευές κατασκευασμένες στην πλατφόρμα Android, οι χρήστες έχουν τη δυνατότητα να προσαρμόσουν ανάλογα με τις ανάγκες και τις απαιτήσεις τους, πλήρως τη συσκευή τους. Εφαρμογές ακόμα και οι βασικές λειτουργίες μπορεί να τροποποιηθούν ή να αντικατασταθούν πλήρως από άλλες. Για παράδειγμα, ο χρήστης μπορεί να χρησιμοποιήσει την επιθυμητή του εφαρμογή για την πρόσβαση στο διαδίκτυο, ή να χρησιμοποιεί διαφορετική εφαρμογή από αυτή του κατασκευαστή, για την προβολή των εικόνων.

## **3) Δραστικότητα**

Οι προγραμματιστές εφαρμογών Android διαθέτοντας αρκετές επιλογές βιβλιοθηκών και εργαλείων, μπορούν να δημιουργήσουν πολύπλοκες εφαρμογές με σχεδόν απεριόριστη λειτουργικότητα. Για παράδειγμα, μια εφαρμογή μπορεί να επικοινωνήσει με server για την μετάδοση δεδομένων από τη συσκευή μέσω διαδικτύου όπως για παράδειγμα ο συγχρονισμός των συναντήσεων του ημερολογίου, τις φωτογραφίες του χρήστη, την τρέχουσα τοποθεσία. Ακόμη μια εφαρμογή μπορεί να λάβει οποιαδήποτε πληροφορία, και να εμφανιστεί στην οθόνη της συσκευής.

## **4) Απλότητα στην ανάπτυξη εφαρμογών**

Η πλατφόρμα παρέχει στο καθένα που ασχολείται με την ανάπτυξη εφαρμογών τη δυνατότητα χρήσης μια μεγάλης ποικιλίας από βιβλιοθήκες και χρήσιμα εργαλεία που μπορούν να χρησιμοποιηθούν για τη δημιουργία ενός πιο εξελιγμένου και πολύπλοκου λογισμικού. Αυτή η ολοκληρωμένη δέσμη από έτοιμα εργαλεία αυξάνει σημαντικά την

παραγωγικότητα των προγραμματιστών Android εφαρμογών αφού τους βοηθά να δημιουργήσουν εκπληκτικά πλούσιο λογισμικό ταχύτερα και με λιγότερα λάθη.

## 2.3 Οι εκδόσεις του Android

Το Android είναι ένα λειτουργικό σύστημα βασισμένο στο Linux. Η ανάπτυξη του ξεκίνησε το 2003 από την Android, Inc., η οποία τελικά το 2005 αγοράστηκε από την Google. Το λογισμικό ανάπτυξης εφαρμογών (SDK) καθώς και η beta έκδοση κυκλοφόρησε τον Νοέμβρη του 2007.

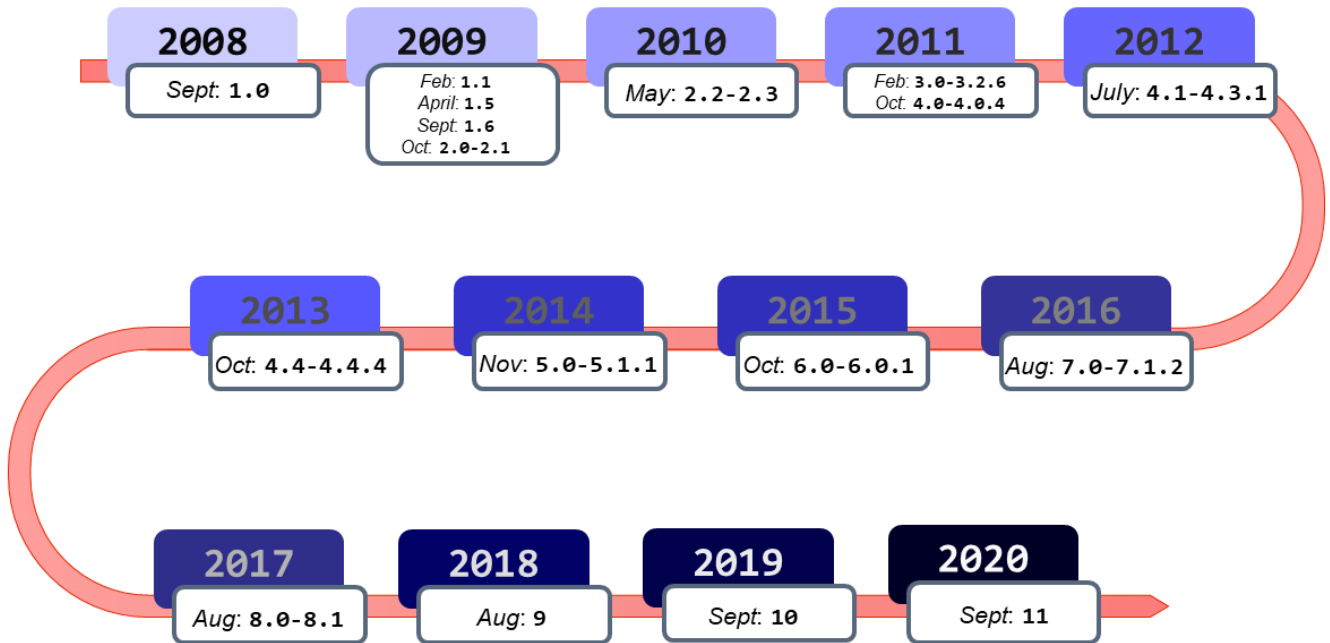
Ένα χρόνο αγνότερα από την πρώτη κυκλοφορία του Android, παρουσιάστηκε από την HTC, το πρώτο εμπορικό smartphone, το HTC Dream, το οποίο βασίστηκε στην έκδοση 1.0.

Ο διαχειριστής του έργου, Ryan Gibson, συνέλαβε τη χρήση ονομασίας με θέμα τη ζαχαροπλαστική, ξεκινώντας από το Android 1.5 Cupcake. Η Google ανακοίνωσε τον Αύγουστο του 2019 ότι τερματίζουν το θεματικό σχέδιο ζαχαροπλαστικής και επισημάνθηκε η χρήση αριθμών για μελλοντικές εκδόσεις. Η πρώτη έκδοση χωρίς κωδική ονομασία ήταν το Android 10, το οποίο κυκλοφόρησε ένα μήνα αργότερα.



Εικόνα 3: Εικονίδια των εκδόσεων του Android

Ας δούμε τις κυριότερες εκδόσεις Android που κυκλοφόρησαν από το 2008 μέχρι σήμερα με χρονολογική σειρά.



Εικόνα 4: Χρονολογική σειρά των εκδόσεων του Android

### 2.3.1 Εκδόσεις Android 1.0 έως 1.1

Το Android έκανε το επίσημο δημόσιο ντεμπούτο του το 2008 με το Android 1.0, μια κυκλοφορία τόσο αρχαία που δεν είχε καν ένα κωδικό όνομα.

Τα πράγματα ήταν αρκετά βασικά τότε, αλλά το λογισμικό περιλάμβανε ένα σύνολο πρώιμων εφαρμογών Google όπως το Gmail, οι Χάρτες, το Ημερολόγιο και το YouTube, όλες ενσωματωμένες στο λειτουργικό σύστημα σε αντίθεση με το πιο εύκολα ανανεώσιμο μοντέλο αυτόνομων εφαρμογών που έχουμε σήμερα.

Χαρακτηριστικά:

- Χάρτες Google, Περιηγητής Ιστού, Ημερολόγιο
- Scroll προς τα κάτω τη γραμμή ειδοποιήσεων
- Ενσωμάτωση Gmail, Επαφές και Συγχρονισμός Google.
- Ασύρματη υποστήριξη - Wi-Fi και Bluetooth.
- Εμφάνιση λεπτομερειών και κριτικών για τοποθεσίες
- Προσθήκη Αποθήκευσης συνημμένου στο μήνυμα

### 2.3.2 Έκδοση Android 1.5: Cupcake

Με την κυκλοφορία του Android 1.5 Cupcake στις αρχές του 2009, γεννήθηκε η παράδοση των ονομάτων έκδοσης Android. Το Cupcake εισήγαγε πολλές βελτιώσεις στη διεπαφή Android, συμπεριλαμβανομένου του πρώτου πληκτρολογίου στην οθόνη, κάτι που θα ήταν απαραίτητο καθώς τα τηλέφωνα απομακρύνθηκαν από το πανταχού παρόν μοντέλο φυσικού πληκτρολογίου.

Το Cupcake έφερε επίσης το framework για widget εφαρμογών, τα οποία γρήγορα θα μετατραπούν σε ένα από τα πιο διακριτικά στοιχεία του Android.

Χαρακτηριστικά:

- Πληκτρολόγιο οθόνης και λειτουργία αναζήτησης
- Μεταφόρτωση βίντεο και εικόνων
- Αντιγραφή και επικόλληση εγγραφών βίντεο
- Υποστήριξη για μορφές MPEG4 και 3GP

### 2.3.3 Έκδοση Android 1.6: Donut

Το Android 1.6, Donut, κυκλοφόρησε στον κόσμο το φθινόπωρο του 2009. Μια από τις πιο σημαντικές προσθήκες ήταν η δυνατότητα λειτουργίας του λειτουργικού συστήματος σε μια ποικιλία διαφορετικών μεγεθών οθόνης και αναλύσεων, ένας παράγοντας που θα ήταν κρίσιμος στα επόμενα χρόνια. Πρόσθεσε επίσης υποστήριξη για δίκτυα CDMA, τα οποία θα διαδραματίσουν βασικό ρόλο στην επικείμενη έκρηξη του Android.

- Widget για χειρισμό Wi-Fi, Bluetooth, GPS κ.λπ.
- Λειτουργίες γρήγορης εναλλαγής γκαλερί και κάμερας
- Ανάλυσης οθόνης WVGA

- Τεχνολογική υποστήριξη για CDMA / EVDO, 802.1x, VPN και μηχανή μετατροπής κειμένου σε ομιλία
- Βελτίωση ταχύτητας κάμερας
- Βελτίωση ταχύτητας αναζήτησης εφαρμογών
- Πλαίσιο γρήγορης αναζήτησης

### 2.3.4 Εκδόσεις Android 2.0 έως 2.1: Eclair

Διατηρώντας τον πρωτοποριακό ρυθμό κυκλοφορίας των πρώτων ετών του Android, το Android 2.0 Eclair, εμφανίστηκε μόλις έξι εβδομάδες μετά το Donut. Η ενημέρωση «point-on», που ονομάζεται επίσης Eclair, κυκλοφόρησε μερικούς μήνες αργότερα.

Το πιο μετασχηματιστικό στοιχείο της κυκλοφορίας ήταν η προσθήκη φωνητικής καθοδήγησης πλοήγησης και πληροφοριών κίνησης σε πραγματικό χρόνο, κάτι που δεν είχε γίνει προηγουμένως (και εξακολουθεί να είναι ουσιαστικά απaráμιλλο) στον κόσμο των smartphone. Πέρα από την πλοήγηση, το Eclair έφερε τις ζωντανές ταπετσαρίες καθώς και την πρώτη λειτουργία speech-to-text της πλατφόρμας. Ακόμη εισήγαγε το τσίμπημα-ζουμ που ήταν τότε στοιχείο αποκλειστικά του iOS. Αυτή η κίνηση θεωρείται ως η σπίθα που πυροδότησε τον μακροχρόνιο «θερμοπυρηνικό πόλεμο» της Apple εναντίον της Google.

Χαρακτηριστικά:

- Ενημέρωση διεπαφής χρήστη
- Υποστήριξη Bluetooth 2.1
- Βελτίωση των Χαρτών Google
- Μικρές αλλαγές API
- Υποστήριξη ζωντανών και κινούμενων ταπετσαριών
- Δυνατότητα προσθήκης φωτογραφίας επαφής και επιλογής κλήσης, μηνύματος ή email

### 2.3.5 Έκδοση Android 2.2: Froyo

Μόλις τέσσερις μήνες μετά την άφιξη του Android 2.1, η Google παρουσίασε το Android 2.2, το Froyo, το οποίο ενσωμάτωσε σε μεγάλο βαθμό βελτιώσεις στην απόδοση.

Το Froyo έφερε την υποστήριξη του Flash στο πρόγραμμα περιήγησης ιστού, μια επιλογή που ήταν σημαντική τόσο λόγω της εκτεταμένης χρήσης του Flash εκείνη την εποχή όσο και λόγω της ανυπόμονης στάσης της Apple να το υποστηρίξει στις δικές του κινητές συσκευές.

Χαρακτηριστικά:

- Υποστήριξη για κινούμενες εικόνες GIF και πολλές γλώσσες πληκτρολογίου
- Βελτιώσεις ταχύτητας και απόδοσης
- Υποστήριξη μεταφόρτωσης αρχείων μέσω προγράμματος περιήγησης ιστού
- Υποστήριξη αριθμητικών και αλφαριθμητικών κωδικών πρόσβασης για ενίσχυση της ασφάλειας
- Αυξημένη συμβατότητα με συστήματα αυτοκινήτου και ακουστικά
- Λειτουργία Wi-Fi Hotspot

### 2.3.6 Έκδοση Android 2.3: Gingerbread

Η πρώτη πραγματική οπτική ταυτότητα του Android άρχισε να επικεντρώνεται στην κυκλοφορία του 2010. Το λαμπερό πράσινο ήταν από καιρό το χρώμα της μασκότ ρομπότ του Android και με το Gingerbread, έγινε αναπόσπαστο μέρος της εμφάνισης του λειτουργικού συστήματος. Το μαύρο και πράσινο υπήρχε στην πλειονότητα της διεπαφής του χρήστη καθώς το Android ξεκίνησε την αργή πορεία προς τον διακριτικό σχεδιασμό.

Χαρακτηριστικά:

- Βελτίωση δυνατότητας αντιγραφής και επικόλλησης
- Ανανέωση διεπαφής χρήστη
- Υποστήριξη για μορφή βίντεο VP8 και WebM
- Υποστήριξη βιντεοκλήσεων και πλατφορμών κοινωνικής δικτύωσης
- Βελτίωση απόδοσης πληκτρολογίου με ταχύτερη πληκτρολόγηση

### 2.3.7 Εκδόσεις Android 3.0 έως 3.2: Honeycomb

Η περίοδος του Honeycomb το 2011 ήταν μια περίεργη εποχή για το Android. Το Android 3.0 ήρθε στον κόσμο ως έκδοση μόνο για tablet για να συνοδεύσει την κυκλοφορία του Motorola Xoom και μέσω των επόμενων ενημερώσεων 3.1 και 3.2, παρέμεινε μια οντότητα αποκλειστική για tablet και κλειστού κώδικα.

Υπό την καθοδήγηση του νεοαφιχθέντος επικεφαλής σχεδιασμού Matias Duarte, το Honeycomb παρουσίασε ένα δραματικά επανασχεδιασμένο περιβάλλον εργασίας χρήστη για Android. Είχε ένα διαστημικό «ολογραφικό» σχέδιο που ανταλλάσσει το εμπορικό σήμα της πλατφόρμας με πράσινο χρώμα και έδωσε έμφαση στο να αξιοποιήσει στο έπακρο το χώρο της οθόνης ενός tablet.

Ενώ η ιδέα μιας διεπαφής για tablet δεν κράτησε πολύ, πολλές από τις ιδέες της Honeycomb έθεσαν τα θεμέλια για το Android που γνωρίζουμε σήμερα. Το λογισμικό ήταν το πρώτο που χρησιμοποίησε κουμπιά στην οθόνη για τις κύριες εντολές πλοήγησης του Android.

Χαρακτηριστικά:

- Βελτιώσεις σε Gmail, Επαφές, Κάμερα και Συλλογή
- Υποστήριξη για κωδικούς πρόσβασης με πολύπλοκους χαρακτήρες
- Κρυπτογράφηση χώρου αποθήκευσης και ενημερωμένο περιβάλλον χρήστη μορφής 3D
- Υποστήριξη πολυεπεξεργαστών και πρόσφατων εφαρμογών για εύκολη εργασία μεταξύ πολλαπλών εφαρμογών
- Συγχρονισμός πολυμέσων από κάρτα SD
- Γραμμή συστήματος για κατάσταση συσκευής και ειδοποιήσεις
- Google eBooks
- Υποστήριξη του Adobe Flash στο πρόγραμμα περιήγησης
- Περισσότερη υποστήριξη αισθητήρων
- Βελτίωση ταχύτητα συνδέσεων Wi-Fi
- Επανασχεδιασμένο πληκτρολόγιο

### 2.3.8 Έκδοση Android 4.0: Ice Cream Sandwich

Με το Honeycomb να λειτουργεί ως γέφυρα από το παλιό στο νέο, το Ice Cream Sandwich, κυκλοφόρησε επίσης το 2011 και χρησίμευσε ως επίσημη είσοδος της πλατφόρμας στην εποχή του μοντέρνου σχεδιασμού. Η κυκλοφορία βελτίωσε τις οπτικές έννοιες που εισήχθησαν με το Honeycomb και επανένωσαν τα tablet και τα τηλέφωνα με ένα ενιαίο, εννοποιημένο UI.

Το ICS έριξε μεγάλο μέρος της «ολογραφικής» εμφάνισης της Honeycomb, αλλά κράτησε τη χρήση του μπλε ως highlight σε όλο το σύστημα. Ακόμη μετέφερε βασικά στοιχεία συστήματος, όπως κουμπιά στην οθόνη και εμφάνιση εφαρμογών σε μορφή τύπου κάρτας για την εναλλαγή τους.

Το Android 4.0 έκανε επίσης το swiping μια πιο αναπόσπαστη μέθοδο περιήγησης στο λειτουργικό σύστημα, με την τότε επαναστατική αίσθηση να απομακρύνει πράγματα όπως ειδοποιήσεις και πρόσφατες εφαρμογές. Επίσης ξεκίνησε την αργή διαδικασία της δημιουργίας ενός τυποποιημένου σχεδιαστικού πλαισίου, γνωστού ως «Holo», σε όλο το λειτουργικό σύστημα και στο οικοσύστημα εφαρμογών του Android.

Χαρακτηριστικά:

- Δυνατότητα ορθογραφικού ελέγχου
- Direct Wi-Fi
- Ξεκλείδωμα με αναγνώριση προσώπου.
- Εμφάνιση εφαρμογών σε μορφή τύπου κάρτας για εναλλαγή μεταξύ αυτών
- Βελτιωμένη εγγραφή βίντεο με υψηλή ανάλυση
- Καλύτερη απόδοση της κάμερας
- Δυνατότητα ανοίγματος έως 16 καρτελών στο πρόγραμμα περιήγησης ιστού

### 2.3.9 Εκδόσεις Android 4.1 έως 4.3: Jelly Bean

Διανεμημένες σε τρεις εκπληκτικές εκδόσεις Android, οι εκδόσεις Jelly Bean του 2012 και του 2013 πήραν το νέο ICS ως βάση και σημείωσαν σημαντική πρόοδο στην τελειοποίηση και την ανάπτυξη. Οι κυκλοφορίες πρόσθεσαν αρκετή ισορροπία στο λειτουργικό σύστημα και προχώρησαν πολύ στο να κάνουν το Android πιο ελκυστικό για τον μέσο χρήστη.

Πέρα από τις αλλαγές στην διεπαφή του χρήστη, το Jelly Bean έφερε το Google Now, ένα θεαματικό βοηθητικό προγνωστικό πρόγραμμα που δυστυχώς από τότε μετατράπηκε σε πρόγραμμα ροής ειδήσεων. Μας έδωσε επεκτάσιμες και διαδραστικές ειδοποιήσεις, ένα διευρυμένο σύστημα φωνητικής αναζήτησης και ένα πιο προηγμένο σύστημα για την εμφάνιση αποτελεσμάτων αναζήτησης.

Επιπλέον άρχισε να υποστηρίζεται η υποστήριξη πολλαπλών χρηστών, αν και σε tablet μόνο σε αυτό το σημείο, και μια πρώτη έκδοση του πίνακα γρήγορων ρυθμίσεων του. Το Jelly Bean εισήγαγε ένα σύστημα για τοποθέτηση widget στην οθόνη κλειδώματος που παρά όπως τις πολλές λειτουργίες, με την πάροδο των ετών, εξαφανίστηκε.

Χαρακτηριστικά:



- Φωνητική αναζήτηση και πληκτρολόγηση
- Πανόραμα
- Project Butter
- Επεκτάσιμες ειδοποιήσεις
- Daydream ως screensaver
- Υποστήριξη ήχου USB
- Βελτιωμένη εφαρμογή κάμερας
- Βελτιώσεις ασφαλείας
- Νέες κινήσεις gestures και δυνατότητες προσβασιμότητας
- Πολλαπλοί λογαριασμοί χρηστών (Μόνο για tablet)
- Υποστήριξη ανάλυσης 4k
- Υποστήριξη Bluetooth με χαμηλή ενέργεια
- Η Google εμφανίζει σχετικό περιεχόμενο βάσει του ιστορικού αναζήτησης
- Εγγενής υποστήριξη emoji

### 2.3.10 Έκδοση Android 4.4: KitKat

Η κυκλοφορία του KitKat στα τέλη του 2013 σηματοδότησε το τέλος της σκουρόχρωμης εποχής του Android, καθώς τα μαύρα του Gingerbread και τα μπλε της Honeycomb τελικά έφυγαν από το λειτουργικό σύστημα. Ελαφρύτερα φόντα και πιο ουδέτερα χρώματα πήραν τη θέση τους, με μια διαφανή γραμμή κατάστασης και λευκά εικονίδια που δίνουν στο λειτουργικό σύστημα μια πιο σύγχρονη εμφάνιση.

Το Android 4.4 είδε επίσης την πρώτη έκδοση της υποστήριξης «OK, Google», αλλά στο KitKat, η προτροπή ενεργοποίησης ανοιχτής ακρόασης λειτουργούσε μόνο όταν η οθόνη ήταν ήδη ενεργοποιημένη και ο χρήστης βρισκόταν είτε στην αρχική οθόνη είτε μέσα στην εφαρμογή Google.

Χαρακτηριστικά:

- Εγγραφή οθόνης
- Υποστήριξη GPS
- Ευφύτερο αναγνωριστικό καλούντος
- Υποστήριξη μουσικής εκτός σύνδεσης
- Ενημερώσεις διεπαφής χρήστη για λειτουργία αφύπνισης και χαρτών Google
- Το KitKat διαθέτει τη δυνατότητα «OK Google» που επιτρέπει την πρόσβαση του χρήστη στο Google χωρίς να αγγίξει την συσκευή.

### 2.3.11 Εκδόσεις Android 5.0 και 5.1: Lollipop

Η Google ανακάλυψε ουσιαστικά και πάλι το Android με την έκδοση Android 5.0 Lollipop το φθινόπωρο του 2014. Το Lollipop κυκλοφόρησε το πρότυπο σχεδίασης υλικών που εξακολουθεί να υπάρχει σήμερα, το οποίο έφερε μια εντελώς νέα εμφάνιση που επεκτάθηκε σε όλο το Android και τις εφαρμογές της.

Η ιδέα που βασίζεται σε «κάρτες» και που είχε διασκορπιστεί σε όλο το Android έγινε ένα βασικό μοτίβο διεπαφής χρήστη, που θα άλλαζε την εμφάνιση όλων των στοιχείων.

Το Lollipop εισήγαγε πολλές νέες λειτουργίες στο Android, συμπεριλαμβανομένου του φωνητικού ελέγχου hands-free μέσω της εντολής «OK, Google», υποστήριξη για πολλούς χρήστες σε τηλέφωνα και λειτουργία προτεραιότητας για καλύτερη διαχείριση ειδοποιήσεων. Δυστυχώς, άλλαξε τόσο πολύ που εισήγαγε επίσης πολλά ενοχλητικά σφάλματα, πολλά από τα οποία δεν θα εξολοθρευτούν πλήρως μέχρι την κυκλοφορία του επόμενου έτους 5.1.

Χαρακτηριστικά:

- Υποστήριξη ART
- Καλύτερη προστασία συσκευών
- Οι ειδοποιήσεις μπορούν να αφαιρεθούν από την οθόνη κλειδώματος
- Καλύτερη και βελτιωμένη διεπαφή χρήστη
- Ενσωματωμένη λειτουργία εξοικονόμησης μπαταρίας
- Ανανεωμένη γραμμή πλοήγησης
- Υποστήριξη για πολλαπλές κάρτες sim
- Υψηλή ευκρίνεια της φωνητικής κλήσης.

### 2.3.12 Έκδοση Android 6.0: Marshmallow

Το 2015, το Marshmallow ήταν μια αρκετά μικρή έκδοση Android, μια που έμοιαζε περισσότερο με μια ενημέρωση 0.1. Τότε ξεκίνησε η τάση της Google να κυκλοφορεί μια σημαντική έκδοση Android ανά έτος και αυτή η έκδοση να λαμβάνει πάντα έναν ολόκληρο αριθμό.

Το πιο ελκυστικό στοιχείο του Marshmallow ήταν ένα χαρακτηριστικό αναζήτησης οθόνης που ονομάζεται Now On Tap, κάτι που, είχε πολλές δυνατότητες που δεν είχαν αξιοποιηθεί πλήρως. Η Google δεν τελειοποίησε ποτέ αρκετά το σύστημα και κατέληξε να αποσύρει ήσυχα την επωνυμία της και να το απομακρύνει τον επόμενο χρόνο.

Το Android 6.0 παρουσίασε ορισμένα πράγματα με διαρκή αντίκτυπο, συμπεριλαμβανομένων πιο λεπτομερών αδειών εφαρμογής, υποστήριξη για αναγνώστες δακτυλικών αποτυπωμάτων και υποστήριξη για USB-C.

Χαρακτηριστικά:

- Υποστήριξη για αναγνώστες δακτυλικών αποτυπωμάτων
- Υποστήριξη USB τύπου C
- Λειτουργία πολλαπλών παραθύρων
- «Λειτουργία Ύπνου» για εξοικονόμηση ζωής μπαταρίας
- Εκκαθάριση αδειών συστήματος
- Προσαρμοσμένες καρτέλες google και βελτιωμένη αντιγραφή-επικόλληση

### 2.3.13 Εκδόσεις Android 7.0 και 7.1: Nougat

Οι κυκλοφορίες Android Nougat της Google για το 2016 παρείχαν στο Android μια λειτουργία διαχωρισμού οθόνης, ένα νέο σύστημα για την οργάνωση ειδοποιήσεων ανά εφαρμογή και μια λειτουργία εξοικονόμησης δεδομένων. Το Nougat πρόσθεσε επίσης

μερικές μικρότερες αλλά πολύ σημαντικές δυνατότητες, όπως μια συντόμευση τύπου Alt-Tab για εναλλαγή μεταξύ εφαρμογών.

Ίσως το πιο σημαντικό μεταξύ των βελτιώσεων του Nougat, ήταν η κυκλοφορία του Google Assistant, που συνοδεύτηκε από την ανακοίνωση του πρώτου πλήρως κατασκευασμένου τηλεφώνου της Google, του Pixel, περίπου δύο μήνες μετά το ντεμπούτο του Nougat. Το Google Assistant θα συνεχίσει να γίνεται ένα κρίσιμο στοιχείο του Android και των περισσότερων άλλων προϊόντων Google και είναι αναμφισβήτητη η κύρια προσπάθεια της εταιρείας σήμερα.

Χαρακτηριστικά:

- Λειτουργία πολλαπλών εργασιών και split-screen
- Βελτιώσεις διαχειριστή αποθήκευσης
- Γρήγορη εναλλαγή ρυθμίσεων
- Καλύτερη εφαρμογή ρυθμίσεων
- Ενσωματωμένη απάντηση σε μηνύματα και ειδοποιήσεις χωρίς άνοιγμα εφαρμογών

### 2.3.14 Έκδοση Android 8.0 και 8.1: Oreo

Το Android Oreo πρόσθεσε μια ποικιλία στοιχείων στην πλατφόρμα, όπως μια εγγενή λειτουργία PiP (picture in picture) που επιτρέπει την προβολή βίντεο σε ένα αιωρούμενο παράθυρο και η επιλογή αναβολής ειδοποιήσεων.

Η κυκλοφορία του 2017 περιελάμβανε επίσης μερικά αξιοσημείωτα στοιχεία που προωθούσαν τον στόχο της Google να ευθυγραμμίσει το Android και το Chrome OS και να βελτιώσει την εμπειρία χρήσης εφαρμογών Android σε Chromebook. Ήταν η πρώτη έκδοση Android που παρουσίασε το Project Treble, μια φιλόδοξη προσπάθεια να δημιουργήσει μια αρθρωτή βάση για τον κώδικα του Android με την ελπίδα να διευκολύνει τους κατασκευαστές συσκευών να παρέχουν έγκαιρες ενημερώσεις λογισμικού στις εφαρμογές τους.

Χαρακτηριστικά:

- Αυτόματη συμπλήρωση κωδικού πρόσβασης
- Αυτόματη ενεργοποίηση Wi-Fi
- Δυνατότητα λήψης γραμματοσειρών
- Υποστήριξη πολλαπλών οθονών
- Υποστήριξη Picture-in-Picture
- Υποστήριξη Google Play και νέο στυλ emoji
- Προσαρμοστικά εικονίδια και επιλογή έξυπνου κειμένου

### 2.3.15 Έκδοση Android 9: Pie

Το Android Pie, γνωστό και ως Android 9, εισήλθε στον κόσμο Android τον Αύγουστο του 2018. Η πρώτη αλλαγή του Pie που γίνεται αντιληπτή από το χρήστη ήταν το σύστημα πλοήγησης με κουμπιά και gestures, το οποίο πηρέ τη θέση των παραδοσιακών πλήκτρων.

Το Pie περιελάμβανε επίσης μερικές αξιοσημείωτες λειτουργίες, όπως ένα καθολικό σύστημα προτεινόμενης απάντησης για ειδοποιήσεις ανταλλαγής μηνυμάτων, έναν νέο πίνακα ελέγχου ευεξίας του χρήστη και πιο έξυπνα συστήματα διαχείρισης ενέργειας και φωτεινότητας οθόνης. Φυσικά, δεν έλειπαν μικρότερες αλλά εξίσου σημαντικές προσθήκες, συμπεριλαμβανομένου ενός εξυπνότερου τρόπου χειρισμού σημείων πρόσβασης Wi-Fi, της λειτουργίας εξοικονόμησης μπαταρίας και μια ποικιλία βελτιώσεων σχετικά με το απόρρητο και την ασφάλεια.

Χαρακτηριστικά:

- Ενισχυτής ήχου
- Συμβατότητα τεχνητής νοημοσύνης (AI)
- Προσαρμοσμένη μπαταρία και φωτεινότητα με περιορισμούς στο φόντο
- Υποστήριξη πολλαπλών φωτογραφικών μηχανών με συμβατότητα εξωτερικής κάμερας
- Νέο πλήκτρο συντόμευσης για στιγμιότυπο οθόνης
- Ευκολότερη περιστροφή οθόνης
- Βελτιώσεις έντασης και ήχου
- Δυνατότητα σκοτεινής λειτουργίας
- HDR, HD ήχος, πολλές ταυτόχρονες συνδέσεις Bluetooth
- Βελτιωμένες δυνατότητες ασφαλείας για επιπλέον προστασία
- Ψηφιακή ευεξία με χρονοδιακόπτες εφαρμογών
- Δημιουργία αντιγράφων ασφαλείας Android και βελτιώσεις απορρήτου

### 2.3.16 Έκδοση Android 10

Η Google κυκλοφόρησε το Android 10 τον Σεπτέμβριο του 2019, την πρώτη έκδοση Android που αφαιρείται από το όνομα η κωδική ονομασία με θέμα κάποιο επιδόρπιο. Το λογισμικό φέρνει μια εντελώς επανασχεδιασμένη διεπαφή για Gestures, αφαιρώντας το πλήκτρο Πίσω, βασισμένο πλήρως στην κίνηση του δάχτυλου του χρήστη για τη πλοήγηση στο σύστημα.

Το λογισμικό έχει πολλές άλλες μικρές αλλά σημαντικές βελτιώσεις, συμπεριλαμβανομένου ενός ενημερωμένου συστήματος δικαιωμάτων που δίνει περισσότερο έλεγχο στον χρήστη για το πώς και πότε οι εφαρμογές έχουν πρόσβαση σε δεδομένα τοποθεσίας, καθώς και ένα εκτεταμένο σύστημα για την προστασία μοναδικών αναγνωριστικών των συσκευών.

Πέρα από αυτό, το Android 10 περιλαμβάνει ένα σκοτεινό θέμα σε όλο το σύστημα, μια νέα λειτουργία εστίασης που επιτρέπει στον χρήστη να περιορίσει τους περισπασμούς από συγκεκριμένες εφαρμογές με το πάτημα ενός κουμπιού. Παρουσιάζει επίσης μια νέα λειτουργία ζωντανών λεζάντων που επιτρέπει στον χρήστη τη δημιουργία οπτικοακουστικών υποτίτλων για οποιοδήποτε μέσο αναπαραγωγής στο τηλέφωνό.

Χαρακτηριστικά:

- Υποστήριξη για αναδιπλούμενα smartphone με ευέλικτες και ελαστικές οθόνες
- Σκοτεινή λειτουργία
- Ενισχυτής ήχου με πιο καθαρό ήχο
- Προτάσεις έξυπνης απάντησης για όλες τις εφαρμογές ανταλλαγής μηνυμάτων
- Ζωντανή λεζάντα κατά την αναπαραγωγή πολυμέσων

- Αναίρεση κατάργησης εφαρμογής
- Καλύτερος έλεγχος ειδοποιήσεων με πολλές επιλογές

### 2.3.17 Έκδοση Android 11

Το Android 11, που κυκλοφόρησε στις αρχές Σεπτεμβρίου 2020, είναι μια αρκετά σημαντική ενημέρωση Android τόσο στην διεπαφή του χρήστη όσο και τον πυρήνα του συστήματος. Οι πιο σημαντικές αλλαγές της έκδοσης περιστρέφονται γύρω από το απόρρητο. Η ενημέρωση βασίζεται στο διευρυμένο σύστημα αδειών που εισήχθη στο Android 10 και προσθέτει τη δυνατότητα των χρηστών να παραχωρούν σε εφαρμογές συγκεκριμένες άδειες μόνο σε περιορισμένη βάση μίας χρήσης.

Ακόμη η συγκεκριμένη έκδοση ωθεί την άδεια τοποθεσίας ακόμη πιο βαθιά στο σύστημα και καθιστά πιο δύσκολο για τις εφαρμογές να ζητήσουν την συγκεκριμένη άδεια και συνεπώς λιγότερο πιθανό για τους χρήστες να την ενεργοποιήσουν ακούσια. Ακόμη εισάγει μια νέα δυνατότητα στην οποία οι εφαρμογές που δεν έχουν χρησιμοποιηθεί για λίγους μήνες, θα ανακαλέσουν αυτόματα τα δικαιώματά τους.

Πέρα από αυτό, το Android 11 αφαιρεί την ικανότητα μιας εφαρμογής να βλέπει ποιες άλλες εφαρμογές είναι εγκατεστημένες στη συσκευή, κάτι που ήταν πραγματικά χρήσιμο μέχρι αυτήν την κυκλοφορία, και περιορίζει τους τρόπους με τους οποίους οι εφαρμογές μπορούν να αλληλοεπιδράσουν με τον τοπικό χώρο αποθήκευσης, προκειμένου να προστατευθούν καλύτερα οι πληροφορίες.

Το Android 11 βελτιώνει την περιοχή ειδοποιήσεων του συστήματος για να τονίσει και να απλοποιήσει τις ειδοποιήσεις που σχετίζονται με τη συζήτηση. Εισάγει ένα νέο βελτιωμένο πρόγραμμα αναπαραγωγής πολυμέσων που περιέχει στοιχεία ελέγχου για όλες τις εφαρμογές αναπαραγωγής ήχου και βίντεο σε ένα μόνο χώρο. Ακόμη το Android 11 σηματοδοτεί το πολυαναμενόμενο ντεμπούτο του Bubbles, ένα νέο είδος συστήματος πολλαπλών εργασιών που συζητήθηκε για πρώτη φορά το 2019. Με εφαρμογές που υποστηρίζουν το Bubbles, το σύστημα επιτρέπει στο χρήστη να μεταφέρει συνομιλίες σε κινούμενα παράθυρα που εμφανίζονται πάνω από οτιδήποτε άλλο, και μπορούν να συμπυκνωθούν σε μικρές φυσαλίδες που παραμένουν εύκολα προσβάσιμες για επέκταση.

Το Android 11 έχει πολλές άλλες μικρές αλλά σημαντικές βελτιώσεις, συμπεριλαμβανομένης μιας νέας ενότητας ιστορικού ειδοποιήσεων, μιας λειτουργίας εγγενούς εγγραφής της οθόνης και ενός αυτοματοποιημένου συστήματος προγραμματισμού για το Dark Theme σε όλο το σύστημα.

Χαρακτηριστικά:

- Εγγενής εγγραφή οθόνης
- Σίγαση ειδοποιήσεων κατά τη διάρκεια του βίντεο
- Αυξήση ευαισθησίας αφής
- Ιστορικό ειδοποιήσεων
- Αυτόματη ανάκληση δικαιωμάτων εφαρμογής
- Ανανεωμένες συντομεύσεις μενού και στιγμιότυπου οθόνης
- Νέα λειτουργία επιλογής κειμένου από τη μία εφαρμογή στην άλλη
- Αναίρεση πρόσφατων εκκαθαρισμένων εφαρμογών
- Η λειτουργία πτήσης δεν απενεργοποιεί πια το Bluetooth

- Καρφίτσωμα εφαρμογής στο μενού κοινής χρήσης
- Βελτιωμένες αποχρώσεις ειδοποιήσεων συνομιλίας
- Συννεφάκια συνομιλίας και σκοτεινή λειτουργία με γνώμονα το περιβάλλον
- Βελτιωμένα δικαιώματα μιας χρήσης

## 3 Τι περιλαμβάνει μια εφαρμογή Android.

Μια εφαρμογή Android είναι γραμμένη στην γλώσσα προγραμματισμού Java. Τα εργαλεία Android SDK, που μας βοηθούν για την δημιουργία της εφαρμογής, συγκεντρώνουν διάφορα αρχεία και φακέλους. Όλα αυτά τα δεδομένα, που είναι δομημένα σε μορφή project, εφόσον γίνουν build και compile μέσω των εργαλείων που αναφέραμε, μας δίνουν το αρχείο .apk. Το αρχείο αυτό είναι ουσιαστικά η εφαρμογή που έχουμε δημιουργήσει και μπορούμε να εγκαταστήσουμε στο Android κινητό μας.

### 3.1 Αρχεία στο Android.

#### 3.1.1 Το αρχείο AndroidManifest.xml

Το αρχείο AndroidManifest.xml αποτελεί το βασικό συστατικό κάθε εφαρμογής και πρέπει να τοποθετείται πάντα στον ριζικό κατάλογο (root directory), με ακριβώς αυτό το όνομα. Το αρχείο αυτό, παρουσιάζει βασικές πληροφορίες σχετικά με την εφαρμογή μας, πληροφορίες τις οποίες το λειτουργικό σύστημα πρέπει να γνωρίζει για να μπορέσει να τρέξει οποιοδήποτε κώδικα της εφαρμογής. Ο προγραμματιστής στο αρχείο αυτό καταχωρεί τις σημαντικότερες πληροφορίες της εφαρμογής για χρήση από το λειτουργικό σύστημα. Μεταξύ άλλων το αρχείο AndroidManifest.xml κάνει τα εξής:

- Το όνομα του πακέτου της εφαρμογής. Το όνομα του πακέτου χρησιμεύει ως ένα μοναδικό αναγνωριστικό για την εφαρμογή.
- Το όνομα της εφαρμογής που φαίνεται στον χρήστη ή τη μεταβλητή του αρχείου strings.xml που περιέχει το όνομα της εφαρμογής.
- Όλα τα συστατικά στοιχεία(δραστηριότητες, υπηρεσίες, πάροχοι περιεχομένου, παραλήπτες εκπομπών) της εφαρμογής.
- Η έκδοση των APIs που χρησιμοποιούνται.
- Ο αριθμός έκδοσης της εφαρμογής ή τη μεταβλητή του αρχείου strings.xml που δείχνει τον αριθμό έκδοσης.
- Οι άδειες που χρειάζονται για να εκτελεστούν ορισμένες λειτουργίες της εφαρμογής.

#### 3.1.2 Ο φάκελος src (source)

Ο φάκελος src συμπεριλαμβάνει τα αρχεία Java που αντιστοιχούν σε όλες τις δραστηριότητες, υπηρεσίες, παρόχους περιεχομένου, βοηθητικά αρχεία, κλπ. της εφαρμογής. Επομένως αποτελεί τον μοναδικό φάκελο στο project στον οποίο αποθηκεύονται αρχεία κώδικα και πακέτα τα οποία περιέχουν αρχεία Java.

### 3.1.3 Ο φάκελος res

Στο φάκελο res περιέχονται όλα τα αρχεία εικόνας, κειμένου, xml layout, κλπ. τα οποία χρησιμοποιούνται από τις δραστηριότητες. Φυσικά, δεν βρίσκονται όλα τα αρχεία πόρων, σε έναν αποκλειστικά φάκελο, αλλά είναι χωρισμένα και ταξινομημένα σε υποφακέλους ανάλογα με το είδος τους. Συνηθισμένοι υποφάκελοι του κύριου φακέλου res, είναι ο φάκελος drawable ο οποίος περιέχει τα αρχεία εικόνας (.png, .jpg, .gif, .svg, .xml) τα οποία χρησιμοποιεί η εφαρμογή, ο φάκελος layout ο οποίος περιέχει όλα τα αρχεία xml τα οποία ορίζουν τα διάφορα layouts που υπάρχουν στην εφαρμογή, και τέλος ο φάκελος values στον οποίο αποθηκεύονται τιμές για διάφορες μεταβλητές της εφαρμογής.

### 3.1.4 Ο φάκελος bin

Ο φάκελος bin παράγεται από τον compiler. Περιέχει όλα τα αρχεία .class για να παράγει το αρχείο .apk, το οποίο είναι το εκτελέσιμο αρχείο στο λειτουργικό σύστημα Android. Σε αυτό το φάκελο μπορούμε να βρούμε το αρχείο .apk αφού κάνουμε build και compile το project μας.

### 3.1.5 Ο φάκελος gen

Δημιουργείται αυτόματα και περιέχει τις κλάσεις Java που μας δίνουν διάφορες πληροφορίες για την εφαρμογή μας. Δεν πρέπει να επεξεργαζόμαστε τα περιεχόμενα του φακέλου αυτού.

## 3.2 Βασικά Δομικά Μέρη μιας Εφαρμογής Android

Η κάθε εφαρμογή Android περιέχει κάποια απαραίτητα δομικά στοιχεία. Κάθε στοιχείο είναι ένα διαφορετικό σημείο από το οποίο το σύστημα μπορεί να εισέλθει στην εφαρμογή μας. Υπάρχουν τέσσερις διαφορετικοί τύποι δομικών στοιχείων: δραστηριότητες (activities), υπηρεσίες (services), πάροχοι περιεχομένου (content providers) και παραλήπτες εκπομπών (broadcast receivers). Κάθε τύπος εξυπηρετεί ένα συγκεκριμένο σκοπό και έχει ένα ξεχωριστό κύκλο ζωής που καθορίζει το πώς το στοιχείο δημιουργείται και καταστρέφεται.

### 3.2.1 Δραστηριότητες

Πρόκειται ίσως για το κύριο δομικό στοιχείο μιας εφαρμογής. Μια δραστηριότητα αντιπροσωπεύει ένα στιγμιότυπο της οθόνης με μια διεπαφή χρήση. Για παράδειγμα, μία εφαρμογή ηλεκτρονικού ταχυδρομείου μπορεί να έχει μία δραστηριότητα που εμφανίζει μια λίστα των μηνυμάτων ηλεκτρονικού ταχυδρομείου, μια άλλη δραστηριότητα για να συνθέσει ένα μήνυμα ηλεκτρονικού ταχυδρομείου, καθώς και μια τρίτη δραστηριότητα για την ανάγνωση ενός ηλεκτρονικού μηνύματος. Αν και οι δραστηριότητες δουλεύουν μαζί

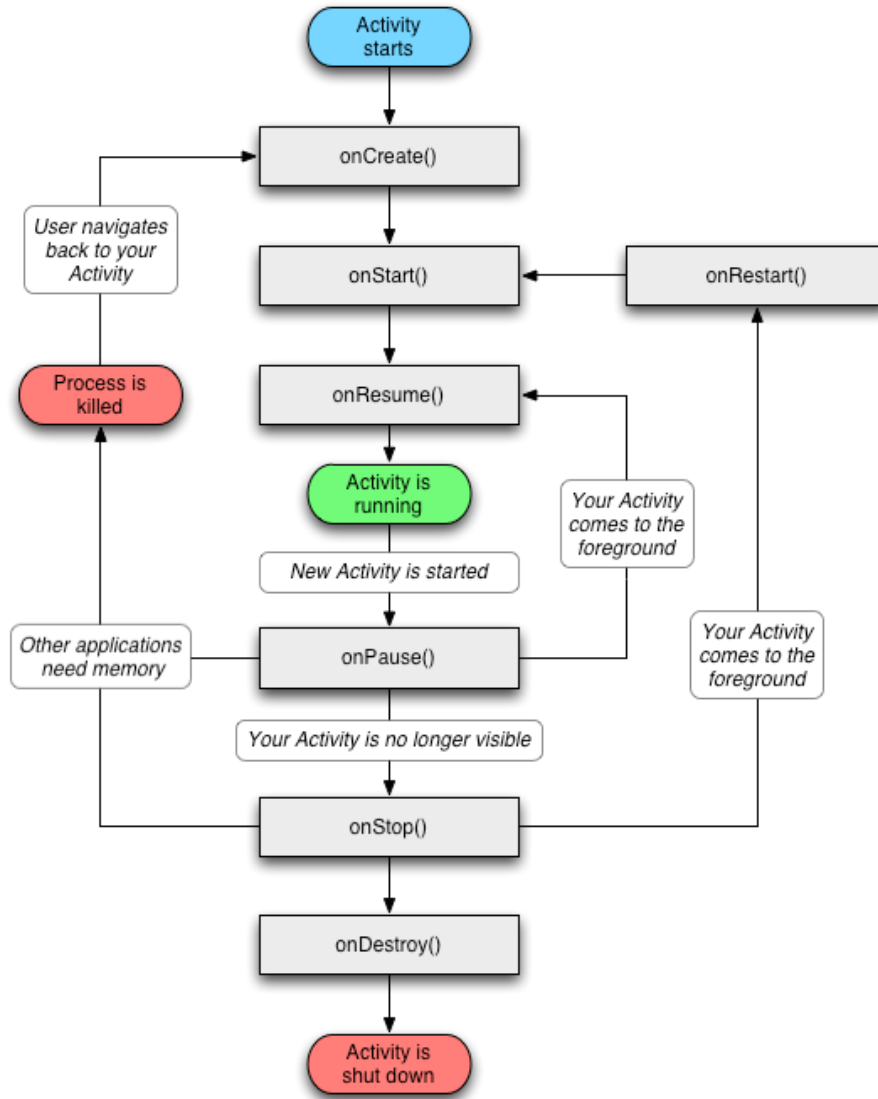


για να δημιουργήσουν μια ομαλή εμπειρία στο χρήστη της εφαρμογής, κάθε μια είναι ανεξάρτητη από τις άλλες. Ως εκ τούτου, μια διαφορετική εφαρμογή μπορεί να ξεκινήσει οποιαδήποτε από τις δραστηριότητες αυτές. Για παράδειγμα, μια εφαρμογή της κάμερας μπορεί να ξεκινήσει τη δραστηριότητα που συνθέτει ένα νέο ηλεκτρονικό μήνυμα, έτσι ώστε ο χρήστης να μοιράσει μια εικόνα που έχει βγάλει.

Κάθε δραστηριότητα υλοποιείται ως μια υποκλάση της κλάσης Activity και υλοποιεί τη μέθοδο onCreate() η οποία καλείται τη στιγμή που ξεκινάει η δραστηριότητα. Επίσης, μέσα στη μέθοδο αυτή πρέπει να δημιουργηθεί η διεπαφή χρήστη η οποία δηλώνεται με την εντολή setContentView() και η findViewById() για να ανακτήσει τα διάφορα widgets προγραμματιστικά. Για να ξεκινήσουμε μια δραστηριότητα MyActivity μέσα από μία άλλη δραστηριότητα χρησιμοποιούμε τις παρακάτω γραμμές κώδικα:

```
Intent intent = new Intent (this, MyActivity.class);  
startActivity(intent);
```

Ο τερματισμός μιας δραστηριότητας μπορεί να επιτευχθεί με την κλήση της μεθόδου finish(). Το Android έχει την ικανότητα να διαχειριστεί από μόνο του τον κύκλο ζωής των δραστηριοτήτων, οπότε δεν χρειάζεται να τις τερματίσουμε εμείς. Ο κύκλος ζωής και οι αντίστοιχες μέθοδοι των δραστηριοτήτων φαίνονται παρακάτω.



Εικόνα 5: Κύκλος ζωής δραστηριοτήτων

### 3.2.2 Υπηρεσίες

Μια υπηρεσία είναι ένα στοιχείο που εκτελείται στο παρασκήνιο χωρίς άμεση αλληλεπίδραση με τον χρήστη. Καθώς η υπηρεσία δεν έχει διεπαφή χρήστη, δεν δεσμεύεται από τον κύκλο ζωής μιας δραστηριότητας. Οι υπηρεσίες χρησιμοποιούνται για επαναλαμβανόμενες και δυνητικά μακροχρόνιες λειτουργίες, δηλαδή, λήψεις μέσω διαδικτύου, έλεγχος για νέα δεδομένα, επεξεργασία δεδομένων, ενημέρωση παρόχων περιεχομένου και παρόμοιες άλλες λειτουργίες. Οι υπηρεσίες εκτελούνται με υψηλότερη προτεραιότητα από τις ανενεργές δραστηριότητες και, ως εκ τούτου, είναι λιγότερο πιθανό το σύστημα Android να τις τερματίσει. Οι υπηρεσίες μπορούν επίσης να ρυθμιστούν για επανεκκίνηση εάν τερματιστούν από το σύστημα Android μόλις διατεθούν πάλι οι επαρκείς πόροι από το σύστημα.

Κάθε υπηρεσία υλοποιείται ως μια υποκλάση της κλάσης Service και έχει δύο βασικές μεθόδους. Η πρώτη μέθοδος είναι η `startService()` και καλείται από κάποια δραστηριότητα για να ξεκινήσει την υπηρεσία η οποία ακόμα και αν η δραστηριότητα που

τερματιστεί, η συγκεκριμένη υπηρεσία συνεχίζει να εκτελείται. Για το λόγο αυτό οι υπηρεσίες δεν επιστρέφουν κάποιο αποτέλεσμα στις δραστηριότητες που τις κάλεσαν αλλά εκτελούν κάποια λειτουργία όπως η αναπαραγωγή ενός τραγουδιού. Η μέθοδος `bindService()` καταφέρνει να ενώσει την υπηρεσία με κάποιο άλλο στοιχείο της εφαρμογής, όπως με μία δραστηριότητα, έτσι ώστε να υπάρχει αλληλεπίδραση μεταξύ των στοιχείων αυτών. Επομένως εάν τερματιστεί η δραστηριότητα τερματίζεται και η υπηρεσία. Για τον τερματισμό των υπηρεσιών, θα πρέπει να πραγματοποιηθεί η κλήση της μεθόδου `stopSelf()`, αφού το Android δεν διαχειρίζεται τον κύκλο ζωής των υπηρεσιών.

### 3.2.3 Πάροχοι Περιεχομένου

Ένας πάροχος περιεχομένου διαχειρίζεται ένα κοινό σύνολο δεδομένων μιας εφαρμογής. Τα δεδομένα μπορούν να αποθηκευτούν στο σύστημα αρχείων, σε μια βάση δεδομένων SQLite, στο διαδίκτυο ή σε οποιαδήποτε άλλη θέση αποθήκευσης στην οποία έχει πρόσβαση η εφαρμογή. Μέσω του παρόχου περιεχομένου, άλλες εφαρμογές μπορούν να προσπελάσουν ή ακόμα και να τροποποιήσουν τα δεδομένα (αν ο πάροχος περιεχομένου το επιτρέπει). Για παράδειγμα, το σύστημα Android παρέχει μια υπηρεσία παροχής περιεχομένου που διαχειρίζεται τις επαφές του χρήστη. Ως εκ τούτου, οποιαδήποτε εφαρμογή έχει τα κατάλληλα δικαιώματα μπορεί να μέσω του παρόχου περιεχομένου να διαβάσει και να γράψει πληροφορίες σχετικά με ένα συγκεκριμένο πρόσωπο.

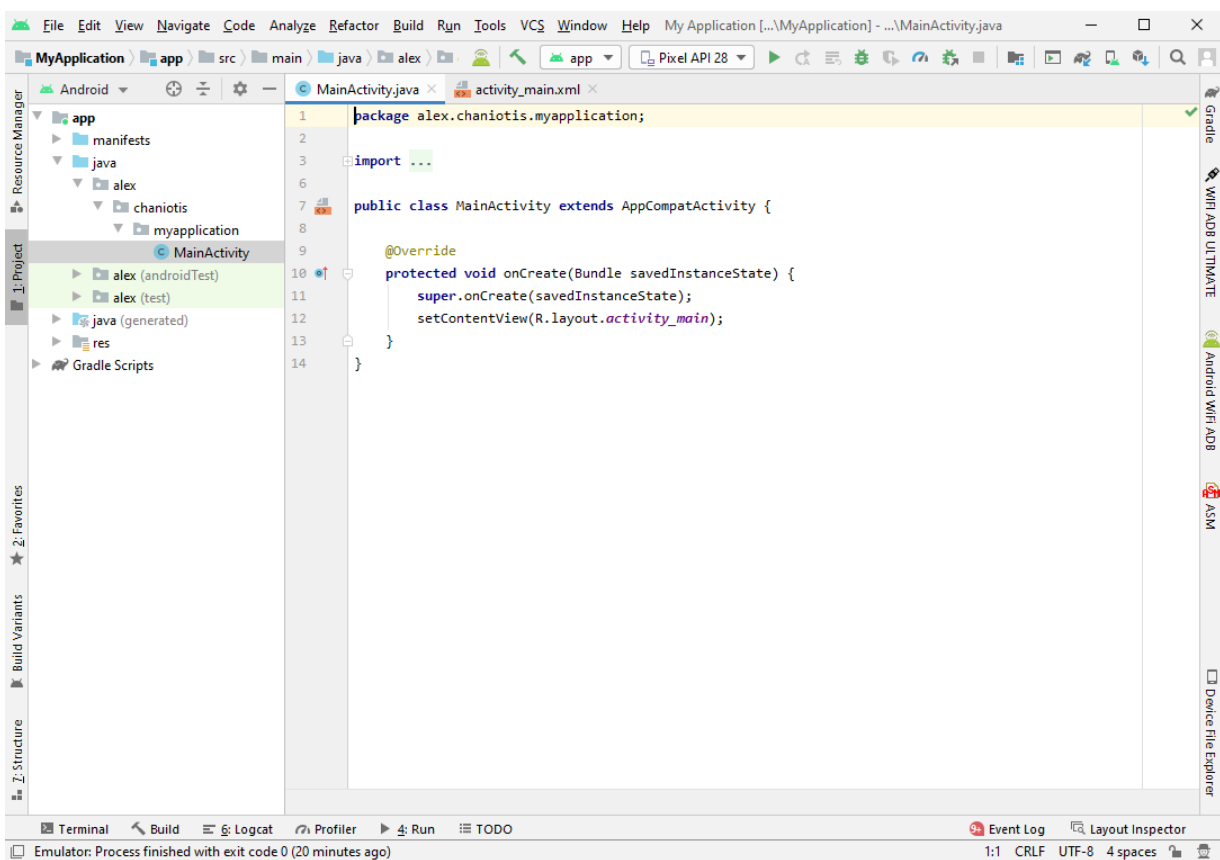
Κάθε πάροχος περιεχομένου υλοποιείται ως μια υποκλάση της κλάσης `ContentProvider` και πρέπει να υλοποιεί στο εσωτερικό του ένα τυποποιημένο σύνολο από APIs που επιτρέπουν σε άλλες εφαρμογές να πραγματοποιούν συναλλαγές.

### 3.2.4 Παραλήπτες Εκπομπών

Ένας παραλήπτης εκπομπών είναι ένα στοιχείο που ανταποκρίνεται σε όλο το σύστημα μετάδοσης ανακοινώσεων. Πολλές εκπομπές προέρχονται από το σύστημα, για παράδειγμα, μία εκπομπή που ανακοινώνει ότι η οθόνη έχει σβήσει, ότι η μπαταρία είναι χαμηλή, ή ότι μια φωτογραφία τραβήχτηκε. Οι εκπομπές μπορεί να προέρχονται κι από εφαρμογές. Για παράδειγμα, για να ενημερώσει άλλες εφαρμογές ότι έχει κατεβάσει κάποια δεδομένα και είναι διαθέσιμα να τα χρησιμοποιήσουν. Αν και οι παραλήπτες εκπομπών δεν εμφανίζουν διεπαφές χρήστη, μπορεί να δημιουργήσουν μια ειδοποίηση στη γραμμή κατάστασης για να ειδοποιήσουν το χρήστη όταν συμβεί μια εκπομπή. Πιο συχνά όμως, ένας παραλήπτης εκπομπών είναι απλά μια «πύλη» προς άλλα στοιχεία και έχει ως στόχο να κάνει ένα πολύ ελάχιστο ποσό της εργασίας. Για παράδειγμα, θα μπορούσε να ξεκινήσει μια υπηρεσία για να εκτελέσει κάποιες εργασίες που βασίζονται στο γεγονός. Κάθε παραλήπτης εκπομπών υλοποιείται ως μια υποκλάση της κλάσης `BroadcastReceiver` και κάθε εκπομπή παραδίδεται ως ένα αντικείμενο της κλάσης `Intent`.

## 4 Προγραμματιστικά Εργαλεία

Η γλώσσα προγραμματισμού που χρησιμοποιείται για την κατασκευή εφαρμογών Android είναι η Java, η οποία έχει πλήρη πρόσβαση στο API του Android. Συνεπώς, βασική προϋπόθεση για την ανάπτυξη εφαρμογών Android είναι η εγκατάσταση του Java Development Kit (JDK). Επίσης, είναι βολικό να χρησιμοποιήσουμε ένα περιβάλλον ανάπτυξης (Integrated Development Environment, IDE) για τη συγγραφή του κώδικα και την εκτέλεση του προγράμματός μας. Αρχικά, το περιβάλλον ανάπτυξης που υποστήριζε η Google ήταν το Eclipse χρησιμοποιώντας το plugin Android Development Tools (ADT). Τον Δεκέμβριο του 2014, η ίδια η Google κυκλοφόρησε το Android Studio, το οποίο βασίζεται στο IntelliJ IDEA, και το κατοχύρωσε ως το κύριο περιβάλλον για την ανάπτυξη εφαρμογών Android.

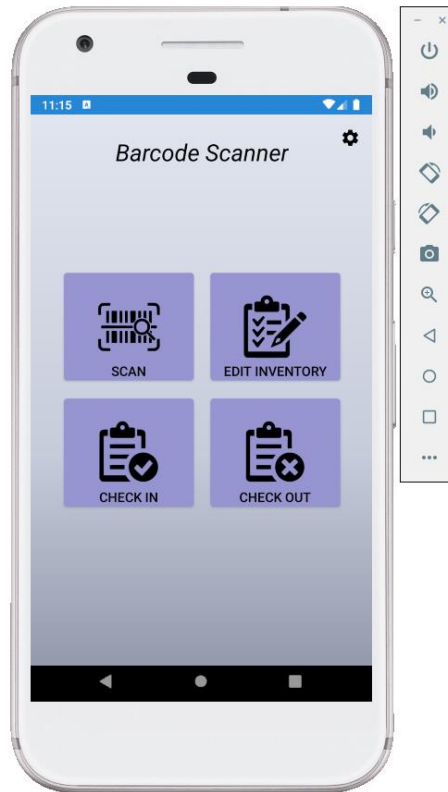


Εικόνα 6: Το περιβάλλον ανάπτυξης Android Studio

Το βασικότερο εργαλείο για την ανάπτυξη Android εφαρμογών είναι το Android SDK (Software Development Kit) το οποίο μας παρέχει τα επιπλέον εργαλεία ώστε να μπορούμε να γράψουμε κώδικα συγκεκριμένα για την κατασκευή εφαρμογής Android. Συγκεκριμένα, το SDK περιλαμβάνει μια πλήρη σειρά από εργαλεία ανάπτυξης, συμπεριλαμβανομένων ενός προγράμματος εντοπισμού σφαλμάτων, βιβλιοθηκών λογισμικού, ενός εξομοιωτή βασισμένου στο QEMU, εγγράφων, δειγμάτων κώδικα, και παραδειγμάτων.

## Εφαρμογή Android για την αναγνώριση και καταγραφή προϊόντων μέσω barcode

Για να τρέξουμε την Android εφαρμογή μας, είτε θα την εγκαταστήσουμε σε ένα smartphone ή θα χρησιμοποιήσουμε κάποιον εξομοιωτή κινητού τηλεφώνου. Το Android SDK περιλαμβάνει έναν εξομοιωτή κινητών συσκευών, δηλαδή μια εικονική φορητή συσκευή που τρέχει στον υπολογιστή. Ο εξομοιωτής επιτρέπει την ανάπτυξη και τη δοκιμή εφαρμογών του Android χωρίς τη χρήση φυσικής συσκευής.



Εικόνα 7: Εικονική συσκευή με λειτουργικό σύστημα Android.

## 4.1 Εισαγωγή

Τα κύρια και απαραίτητα εργαλεία λογισμικού για την ανάπτυξη μιας Android εφαρμογής είναι μια πλατφόρμα για την ανάπτυξη εφαρμογών, στην περίπτωση μας το Android Studio, οι βιβλιοθήκες οι οποίες επιτρέπουν την ανάπτυξη της εφαρμογής (Android SDK) και μια πλατφόρμα δοκιμής της εφαρμογής είτε αυτή είναι φυσική (smartphone), είτε είναι εικονική.

## 4.2 Android Studio

Το Android Studio είναι ένα ολοκληρωμένο προγραμματιστικό περιβάλλον (IDE) για ανάπτυξη εφαρμογών στην πλατφόρμα Android. Ανακοινώθηκε στις 16 Μαΐου 2013 στο συνέδριο Google I/O και είναι διαθέσιμο ελεύθερα με την άδεια Apache License 2.0.

Το Android Studio ήταν διαθέσιμο σε πρώιμο στάδιο, για προεπισκόπηση, ξεκινώντας από την έκδοση 0.1 τον Μάιο του 2013, έπειτα ξεκίνησε το δοκιμαστικό στάδιο από την έκδοση 0.8 που βγήκε τον Ιούνιο του 2014. Η πρώτη σταθερή έκδοση βγήκε το Δεκέμβριο του 2014, με την έκδοση 1.0.

Βασισμένο στο λογισμικό της JetBrains' IntelliJ IDEA, το Android Studio σχεδιάστηκε αποκλειστικά για προγραμματισμό Android. Είναι διαθέσιμο για Windows, Mac OS X και Linux, και αντικατέστησε τα Eclipse Android Development Tools (ADT) ως το κύριο IDE της Google για ανάπτυξη εφαρμογών Android.

Τα βασικά χαρακτηριστικά του Android Studio είναι τα ακόλουθα:

- Gradle based build automation είναι δηλαδή ένα εργαλείο το οποίο αυτοματοποιεί τα builds του προγράμματος.
- ProGuard, το οποίο βελτιστοποιεί τον Java κώδικα και επίσης προσφέρει τη δυνατότητα «ηλεκτρονικής υπογραφής» των εφαρμογών.
- Υποστηρίζει το μετασχηματισμό κώδικα αποκλειστικά για Android
- Εργαλεία Lint τα οποία βοηθούν στη βελτίωση της απόδοσης, συμβατότητας και άλλων προβλημάτων.
- Βοηθούς για την εύκολη δημιουργία σχεδίων τόσο του κορμού όσο και των διαφόρων συστατικών μιας εφαρμογής.
- Editor με πολλές δυνατότητες για τη δημιουργία του γραφικού περιβάλλοντος της εφαρμογής με δυνατότητα drag and drop και προεπισκόπησης του γραφικού περιβάλλοντος σε διάφορους τύπους οθονών.
- Υποστήριξη δημιουργίας εφαρμογών για smartwatches.
- Ενσωματωμένη υποστήριξη για Google Cloud Platform, ενεργοποιώντας έτσι τη δυνατότητα αποστολής μηνυμάτων από την Google όπως για παράδειγμα την λειτουργία των ενημερώσεων.

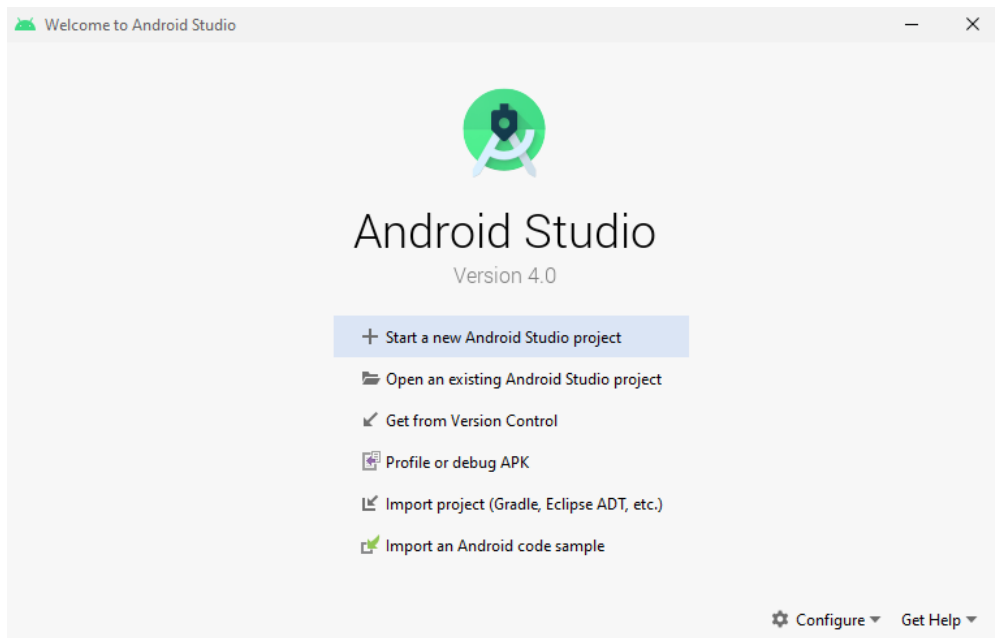
- Ενσωματωμένη υποστήριξη άμεσης δοκιμής της εφαρμογής σε virtual device, γεγονός που μείωσε κατά πολύ το χρόνο που απαιτείται για να γίνει το build, έλεγχος και debug της εφαρμογής.

Στη συνέχεια θα δώσουμε τον σύνδεσμό για τη λήψη του Android Studio (η εγκατάσταση γίνεται με τη χρήση του εκτελέσιμου αρχείου και ακολουθώντας τις οδηγίες που εμπεριέχονται) και θα παρουσιάσουμε με τη χρήση εικόνων τη διαδικασία αρχικοποίησης μιας καινούριας εφαρμογής.

Για την λήψη του Android Studio κάντε κλικ στον σύνδεσμο [ΕΔΩ](#).

Η διαδικασία αρχικοποίησης μια νέας εφαρμογής είναι αυτή όπως παρουσιάζεται παρακάτω:

### 1. Ανοίγουμε το Android Studio

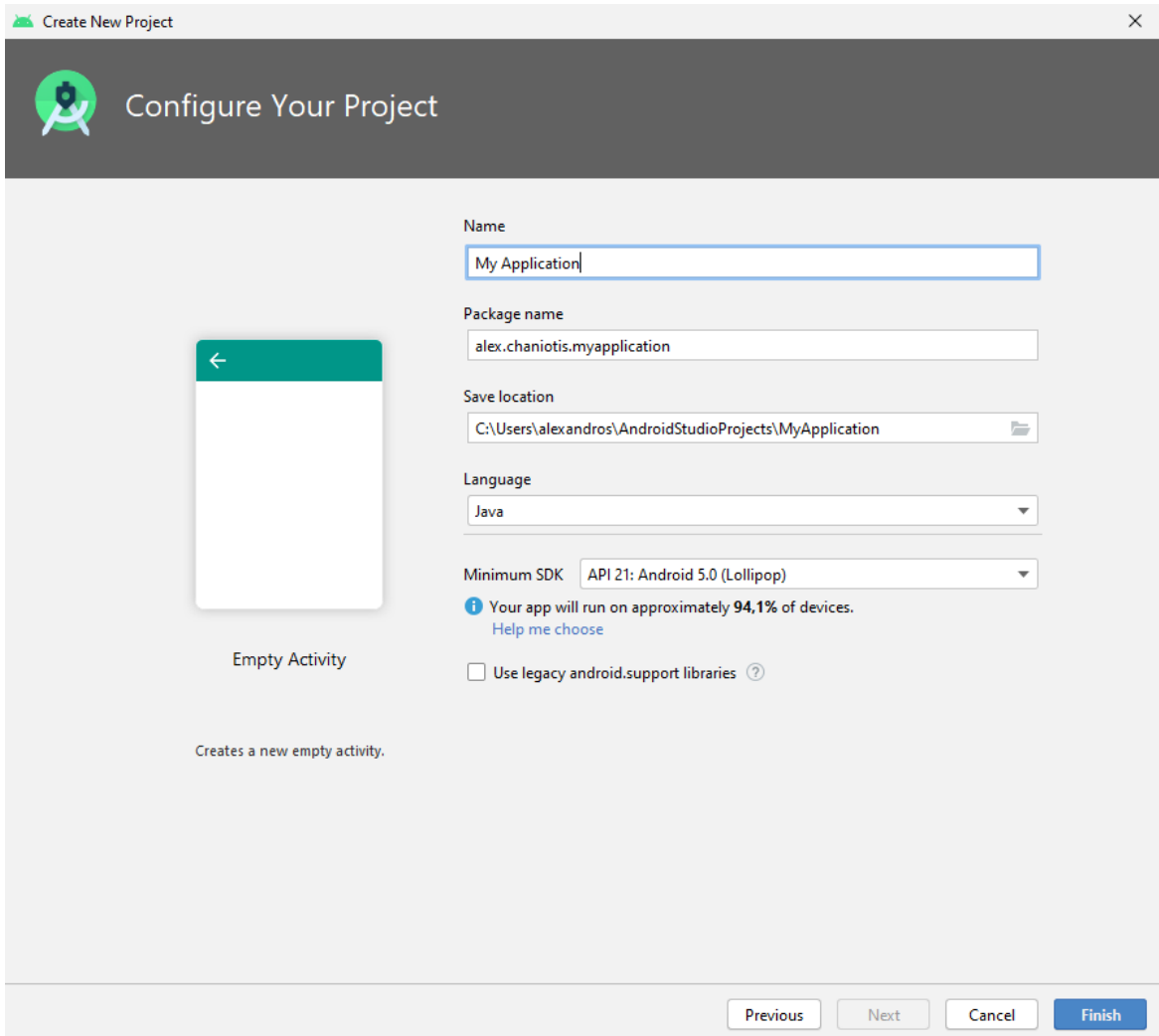


Εικόνα 8: Αρχική οθόνη Android Studio

Στην οθόνη αυτή εμφανίζονται αυτόματα αφενός τα ήδη υπάρχοντα projects (αριστερά) και αφετέρου κάποιες βασικές επιλογές όπως η δυνατότητα εκκίνησης καινούριου project.

## 2. Προχωρούμε στη δημιουργία καινούριας εφαρμογής.

Αφού επιλέξουμε τη δημιουργία καινούριου project, εμφανίζεται η οθόνη στην οποία δίνουμε το όνομα της εφαρμογής και του βασικού της πακέτου και ορίζουμε τη διαδρομή της τοποθεσίας στο δίσκο στην οποία αποθηκεύονται τα αρχεία της εφαρμογής. Ακόμη μας ζητείται να επιλέξουμε την παλαιότερη έκδοση λογισμικού Android που θα υποστηρίξει η εφαρμογή μας.

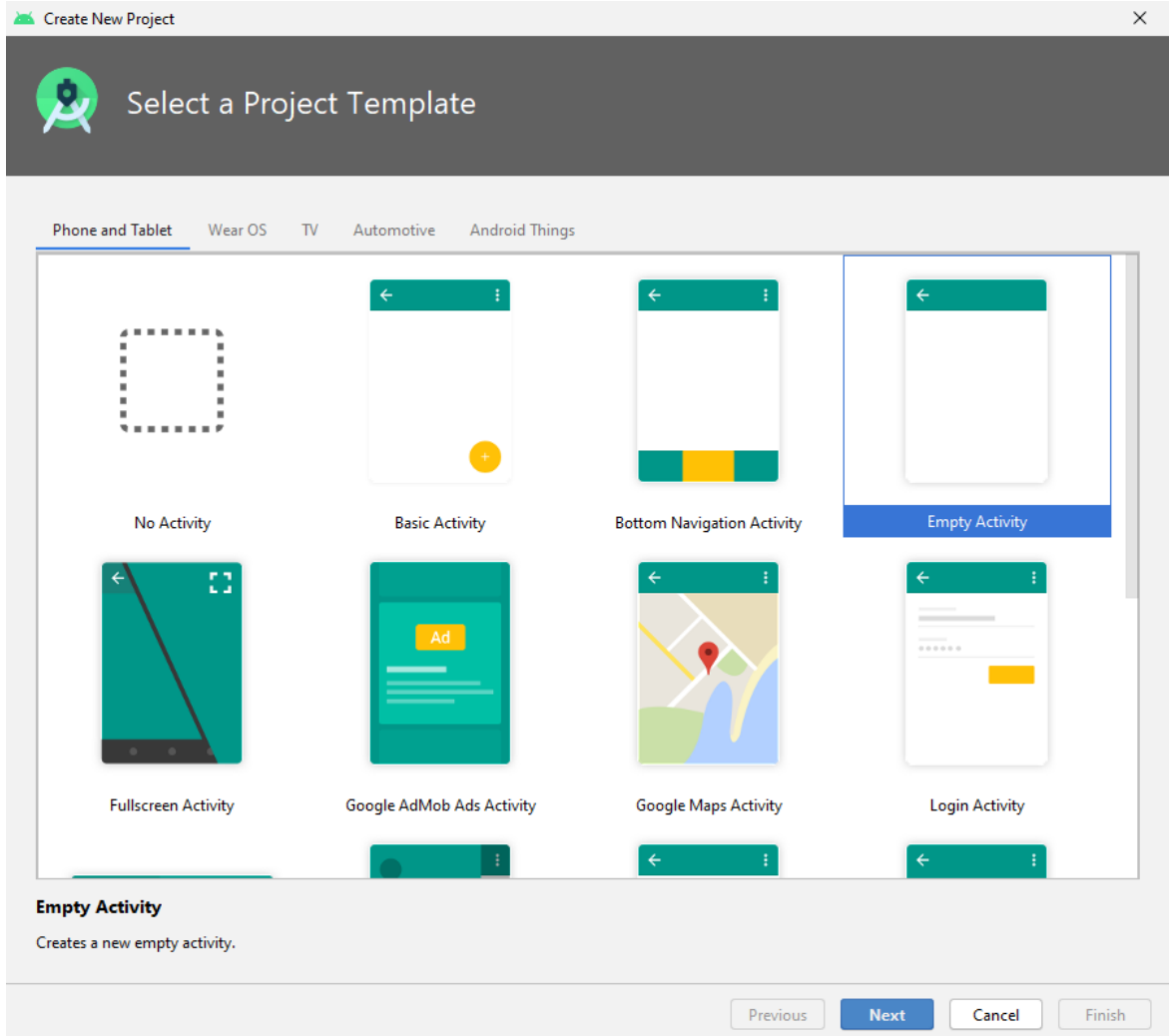


Εικόνα 9: Οθόνη εισαγωγής των αρχικών πληροφοριών της εφαρμογής



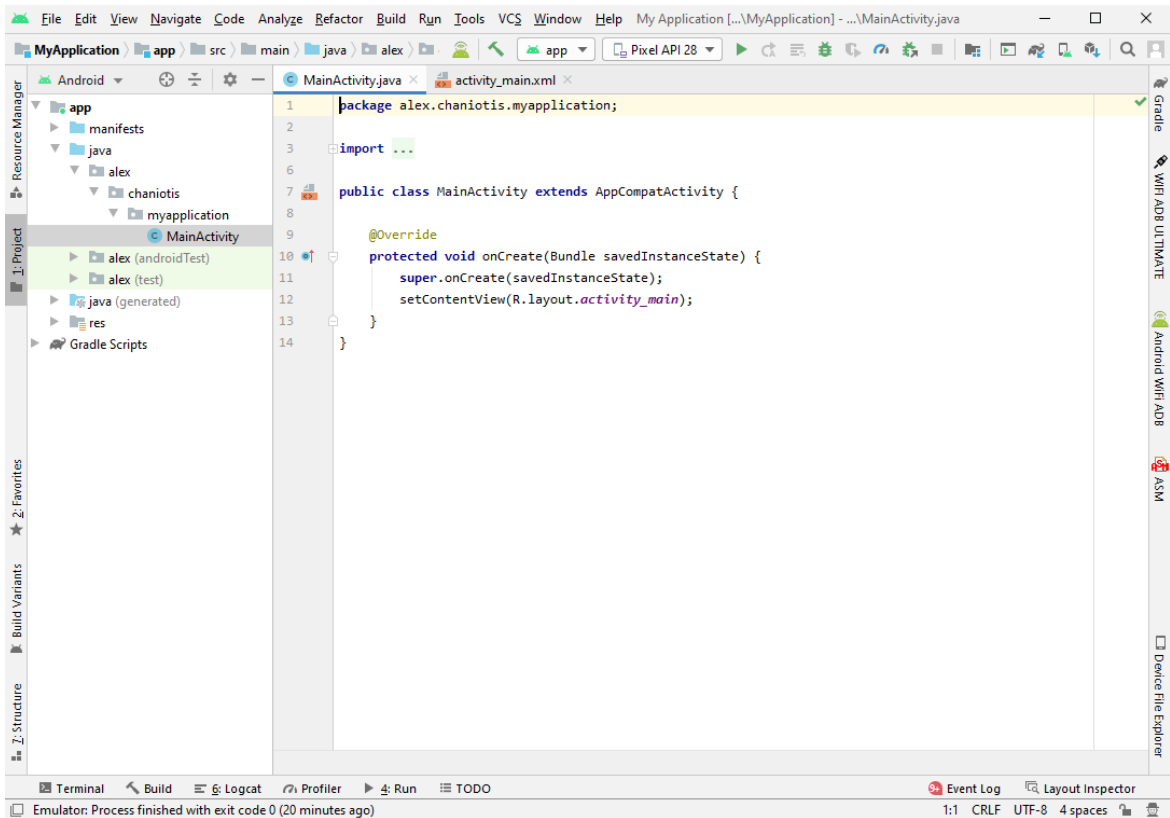
#### 4. Επιλογή είδους αρχικής Activity

Στη συνέχεια πρέπει να επιλέξουμε το είδος της αρχικής Activity της εφαρμογής. Το Android Studio έχει ενσωματωμένες πολλές επιλογές. Στο στάδιο αυτό σημασία έχει αν θέλουμε μια κενή Activity δηλαδή να ξεκινήσουμε την εφαρμογή από την αρχή, χωρίς κομμάτια έτοιμου κώδικα ή να εισάγει το πρόγραμμα κάποιες έτοιμες ρυθμίσεις.



Εικόνα 10: Επιλογή αρχικής Activity

## 6. Κυρίως γραφικό περιβάλλον της πλατφόρμας



Εικόνα 11: Κύρια διεπαφή χρήστη

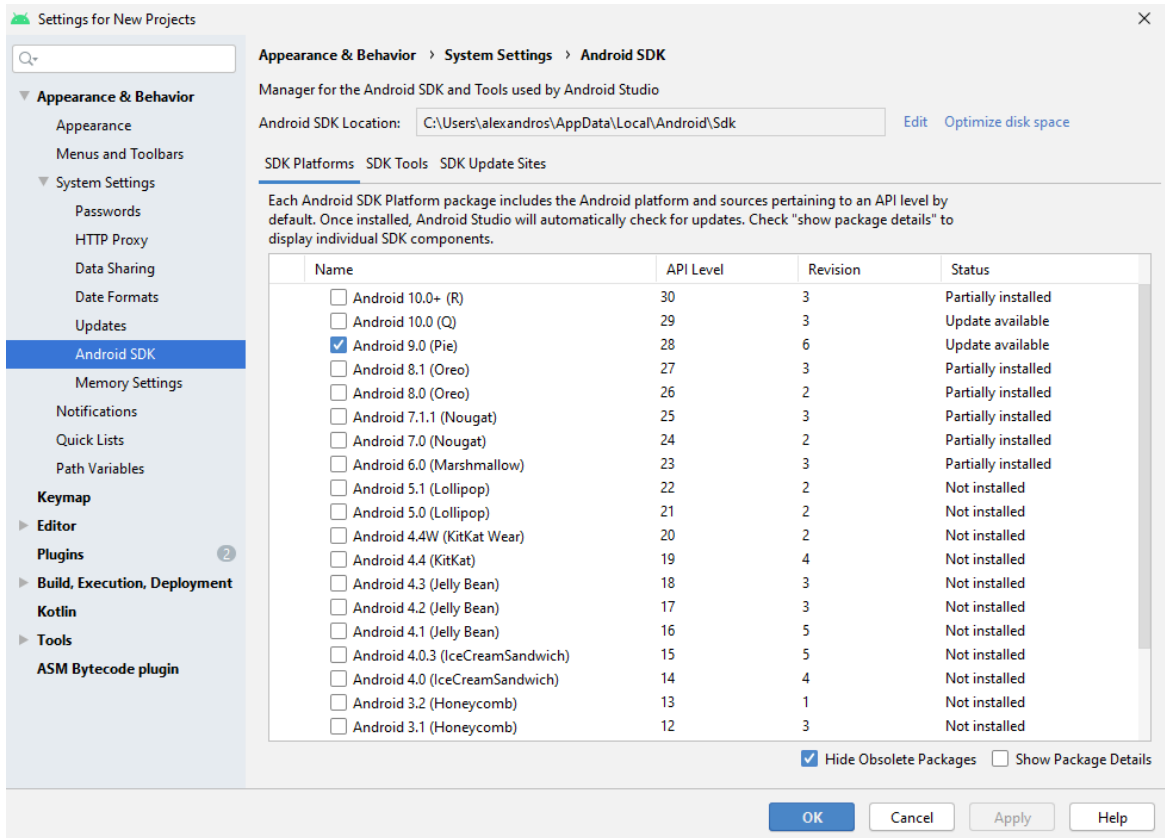
Μετά το τέλος των αρχικοποιήσεων εμφανίζεται το κυρίως γραφικό περιβάλλον στο οποίο εμφανίζεται ο κώδικας ο οποίος γράφουμε αλλά και η δομή του project μας με τα επιμέρους αρχεία.

### 4.3 Android SDK

Τα βασικά εργαλεία ανάπτυξης της εφαρμογής ονομάζονται Android SDK (Software Development Kit) και παρέχονται από την Google. Αυτό περιλαμβάνει τις βιβλιοθήκες Java του Android που θα χρησιμοποιήσει η εφαρμογή, τον Emulator για την δοκιμαστική εκτέλεσή της με τη χρήση εικονικής συσκευής, τον Compiler-Builder για τη «μετάφραση» του κώδικα και μία πληθώρα διαφορετικών βιβλιοθηκών.

Το SDK είναι ενσωματωμένο στο Android Studio, το οποίο βασίζεται στο IntelliJ IDEA και αναπτύσσεται ενεργά από τη Google.

Μας δίνεται έτσι η δυνατότητα να εγκαταστήσουμε περισσότερες της μιας εκδόσεις του λειτουργικού και να χειριζόμαστε όποια από αυτές επιθυμούμε κάθε φορά. Για την ανάπτυξη της εφαρμογής της παρούσας εργασίας επιλέξαμε την έκδοση Android 9.0 . Στην επόμενη εικόνα εμφανίζεται το γραφικό περιβάλλον εγκατάστασης των απαραίτητων βιβλιοθηκών.



Εικόνα 12: SDK Manager

## 4.4 Figma

Η διεπαφή χρήστη (UI) είναι ένα πολύ σημαντικό μέρος οποιασδήποτε εφαρμογής και το Figma έχει το δικό της ισχυρό και μοναδικό σύνολο από εργαλεία για σχεδιασμό στοιχείων του UI όπως για παράδειγμα διανύσματα.

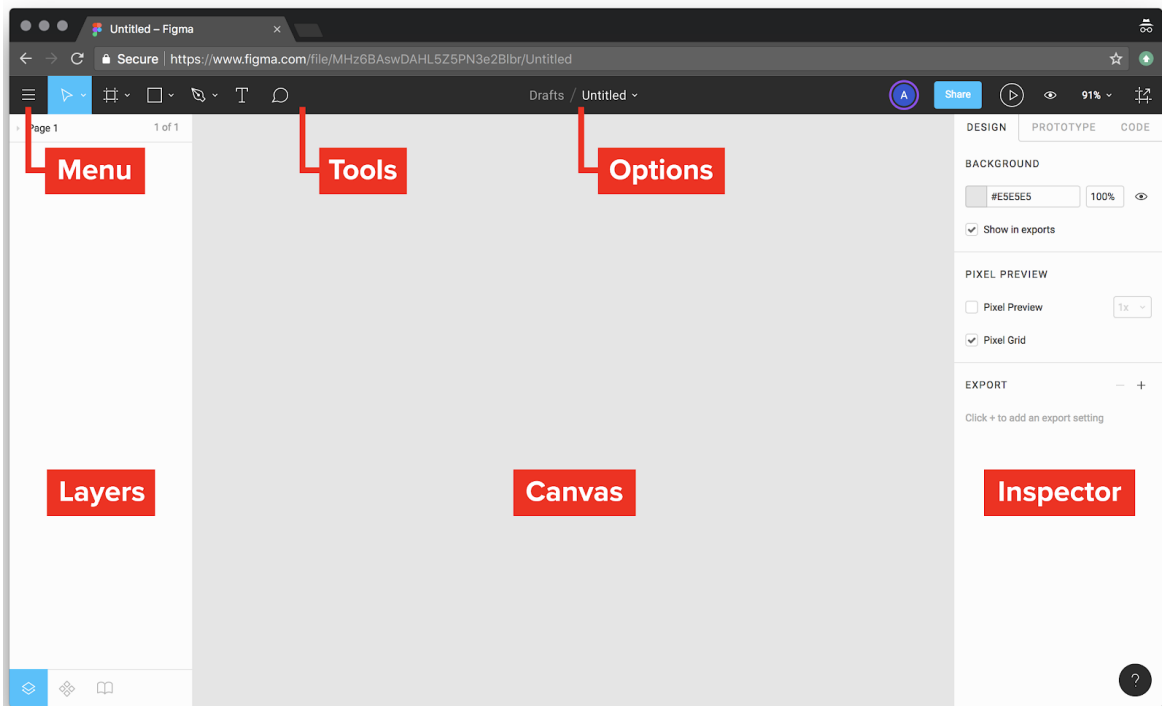
Το Figma είναι ένα διανυσματικό πρόγραμμα επεξεργασίας γραφικών που βασίζεται κυρίως στο web, με πρόσθετες λειτουργίες εκτός σύνδεσης που επιτρέπονται από εφαρμογές επιφάνειας εργασίας για macOS και Windows. Οι συνοδευτικές εφαρμογές Figma Mirror για Android και iOS επιτρέπουν την προβολή πρωτοτύπων Figma σε κινητές συσκευές. Ακόμη δίνει τη δυνατότητα σε προγραμματιστές και σχεδιαστές να μπορούν να εργάζονται σε πραγματικό χρόνο σε κοινά projects μέσω του cloud συγχρονισμού που διαθέτει. Αυτό είναι ένα πολύ βολικό εργαλείο για μεγάλες εταιρείες με πολλούς σχεδιαστές και προγραμματιστές, αλλά στην δικιά μας περίπτωση, θα χρησιμοποιηθεί μόνο για τον σχεδιασμό των αρχικών Activities, ώστε να έχουμε μια κατεύθυνση ως προς το UI πριν ξεκινήσουμε την ανάπτυξη της εφαρμογής μέσω του Android Studio.



Εικόνα 13: Λογότυπο Figma

Το ξεκίνημα στο Figma είναι τόσο απλό όσο το [www.figma.com](https://www.figma.com), κάνοντας κλικ στο «Εγγραφή» και εισάγοντας τα στοιχεία σας. Μόλις το κάνετε αυτό, το Figma θα ανοίξει με μια οθόνη έναρξης όπως αυτή. Κάντε κλικ στο «Νέο αρχείο» και θα ξεκινήσουμε!

Παρακάτω εμφανίζονται και αναλύονται οι κύριοι τομείς της διεπαφής χρήστη του Figma.



Εικόνα 14: Περιβάλλον χρήση Figma

#### **Μενού:**

Σε αντίθεση με τις άλλες εφαρμογές σχεδίασης, το μενού και οι διάφορες ρυθμίσεις του Figma μπορούν να βρεθούν κάνοντας κλικ στο κουμπί που βρίσκεται στην επάνω αριστερή γωνία της οθόνης.

#### **Tools:**

Στην γραμμή εργαλείων ο χρήστης αποκτά γρήγορη πρόσβαση στα εργαλεία που χρησιμοποιούνται πιο συχνά όπως τα πλαίσια, τα σχήματα, το κείμενο και άλλα εργαλεία.

**Options:**

Αυτή η περιοχή εμφανίζει επιπλέον επιλογές για οποιοδήποτε εργαλείο έχει επιλεγθεί από τον χρήστη. Όταν δεν έχει επιλεγεί αντικείμενο, όπως φαίνεται στην εικόνα 14 παραπάνω, το Figma εμφανίζει το όνομα του αρχείου. Όταν έχει επιλεγεί ένα αντικείμενο, εμφανίζονται οι αντίστοιχες επιλογές ανάλογα το επιλεγμένο αρχείο.

**Layers:**

Όπου αναφέρεται κάθε στοιχείο στο αρχείο, οργανωμένο σε πλαίσια και ομάδες.

**Canvas:**

Στον καμβά ο χρήστης μπορεί να δημιουργήσει ελεύθερα και να ελέγξει όλη την εργασία του.

**Inspector:**

Ο Inspector εμφανίζει πληροφορίες και ρυθμίσεις για οποιοδήποτε αντικείμενο από το σύνολο της εργασίας του έχει επιλέξει ο χρήστης.

## 5 Γλώσσες Προγραμματισμού που χρησιμοποιήθηκαν

Οι γλώσσες προγραμματισμού είναι εργαλεία και επιλέγουμε διαφορετικά εργαλεία για την επίλυση διαφορετικού τύπου προβλημάτων. Κάθε γλώσσα προγραμματισμού στοχεύει στην επίλυση ενός συγκεκριμένου προβλήματος. Περισσότερες από μία γλώσσες μπορεί να είναι σε θέση να επιλύσουν το ίδιο πρόβλημα, αλλά κάθε γλώσσα θα προσεγγίσει διαφορετικά την λύση του προβλήματος, καθιστώντας έτσι κάποια γλώσσα πιο κατάλληλη από άλλη. Για την ανάπτυξη της του συστήματος υποστήριξης της εφαρμογής χρησιμοποιήθηκε η JAVA και για τον σχεδιασμό των layout της εφαρμογής ή αλλιώς της διεπαφής του χρήστη χρησιμοποιήθηκε η XML. Όσον αφορά την υλοποίηση της βάσης δεδομένων και την εισαγωγή δεδομένων σε αυτήν, χρησιμοποιήθηκε η γλώσσα SQL, και για την σύνδεση της με την ιστοσελίδα ώστε να επιτευχθεί η μεταφορά δεδομένων με την εφαρμογή, χρησιμοποιήθηκε η PHP.

### 5.1 Java

#### 5.1.1 Εισαγωγή

Η Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού που σχεδιάστηκε από την εταιρεία πληροφορικής Sun Microsystems.

#### 5.1.2 Ιστορία

Στις αρχές του 1991, η Sun αναζητούσε το κατάλληλο εργαλείο για να αποτελέσει την πλατφόρμα ανάπτυξης λογισμικού σε μικροσυσκευές (έξυπνες οικιακές συσκευές έως πολύπλοκα συστήματα παραγωγής γραφικών). Τα εργαλεία της εποχής ήταν γλώσσες όπως η C++ και η C. Μετά από διάφορους πειραματισμούς προέκυψε το συμπέρασμα ότι οι υπάρχουσες γλώσσες δεν μπορούσαν να καλύψουν τις ανάγκες τους. Ο «πατέρας» της Java, James Gosling, που εργαζόταν εκείνη την εποχή για την Sun, έκανε ήδη πειραματισμούς πάνω στη C++ και είχε παρουσιάσει κατά καιρούς κάποιες πειραματικές γλώσσες (C++ ++, που μετέπειτα ονομάστηκε C#) ως πρότυπα για το νέο εργαλείο που αναζητούσαν στην Sun. Τελικά μετά από λίγο καιρό κατέληξαν με μια πρόταση για το επιτελείο της εταιρίας, η οποία ήταν η γλώσσα Oak. Το όνομά της το πήρε από το ομώνυμο δένδρο (βελανιδιά) το οποίο ο Gosling είχε έξω από το γραφείο του και έβλεπε κάθε μέρα.

#### 5.1.3 Από την Oak στη Java

Η Oak ήταν μία γλώσσα που διατηρούσε μεγάλη συγγένεια με την C++. Παρόλα αυτά είχε πολύ πιο έντονο αντικειμενοστραφή χαρακτήρα σε σχέση με την C++ και χαρακτηριζόταν για την απλότητα της. Σύντομα οι υπεύθυνοι ανάπτυξης της νέας γλώσσας ανακάλυψαν ότι το όνομα Oak ήταν ήδη κατοχυρωμένο οπότε κατά την διάρκεια μιας εκ των πολλών συναντήσεων σε κάποιο τοπικό καφέ αποφάσισαν να μετονομάσουν το νέο τους δημιουργήμα σε Java που εκτός των άλλων ήταν το όνομα της αγαπημένης ποικιλίας καφέ για τους δημιουργούς της. Η επίσημη εμφάνιση της Java αλλά και του HotJava (πλοηγός με υποστήριξη Java) στη βιομηχανία της πληροφορικής έγινε το Μάρτιο του 1995 όταν η Sun την ανακοίνωσε στο συνέδριο Sun World 1995. Ο πρώτος μεταγλωττιστής (compiler) της ήταν γραμμένος στη γλώσσα C από τον James Gosling.

Το 1994, ο A. Van Hoff ξαναγράφει τον μεταγλωττιστή της γλώσσας σε Java, ενώ το Δεκέμβριο του 1995 πρώτες οι IBM, Borland, Mitsubishi Electronics, Sybase και Symantec ανακοινώνουν σχέδια να χρησιμοποιήσουν τη Java για την δημιουργία λογισμικού. Από εκεί και πέρα η Java ακολουθεί μία ανοδική πορεία και είναι πλέον μία από τις πιο δημοφιλείς γλώσσες στον χώρο της πληροφορικής. Στις 13 Νοεμβρίου του 2006 η Java έγινε πλέον μια γλώσσα ανοιχτού κώδικα (GPL) όσον αφορά το μεταγλωττιστή (Javac) και το πακέτο ανάπτυξης (JDK, Java Development Kit).

#### **5.1.4 Η εξαγορά από την Oracle και το μέλλον της Java**

Στις 27 Απριλίου 2010 η εταιρία λογισμικού Oracle Corporation ανακοίνωσε ότι μετά από πολύμηνες συζητήσεις ήρθε σε συμφωνία για την εξαγορά της Sun Microsystems και των τεχνολογιών (πνευματικά δικαιώματα/ πατέντες) που η δεύτερη είχε στην κατοχή της ή δημιουργήσει. Η συγκεκριμένη συμφωνία θεωρείται σημαντική για το μέλλον της Java και του γενικότερου οικοσυστήματος τεχνολογιών γύρω από αυτή μιας και ο έμμεσος έλεγχος της τεχνολογίας και η εξέλιξη της περνάει σε άλλα χέρια.

#### **5.1.5 Τα χαρακτηριστικά της Java**

Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα. Για να επιτευχθεί όμως αυτό χρειαζόταν κάποιος τρόπος έτσι ώστε τα προγράμματα γραμμένα σε Java να μπορούν να είναι «κατανοητά» από κάθε υπολογιστή ανεξάρτητα του είδους επεξεργαστή (Intel x86, IBM, Sun SPARC, Motorola) αλλά και λειτουργικού συστήματος (Windows, Unix, Linux, BSD, MacOS). Ο λόγος είναι ότι κάθε κεντρική μονάδα επεξεργασίας κατανοεί διαφορετικό κώδικα μηχανής. Ο συμβολικός κώδικας (assembly) που μεταφράζεται και εκτελείται σε Windows είναι διαφορετικός από αυτόν που μεταφράζεται και εκτελείται σε έναν υπολογιστή Macintosh. Η λύση δόθηκε με την ανάπτυξη της Εικονικής Μηχανής (Virtual Machine ή VM ή EM στα ελληνικά).

##### **5.1.5.1 Η εικονική μηχανή της Java**

Αφού γραφεί κάποιο πρόγραμμα σε Java, στη συνέχεια μεταγλωττίζεται μέσω του μεταγλωττιστή Javac, ο οποίος παράγει έναν αριθμό από αρχεία .class (κώδικας byte ή bytecode). Ο κώδικας byte είναι η μορφή που παίρνει ο πηγαίος κώδικας της Java όταν μεταγλωττιστεί. Όταν πρόκειται να εκτελεστεί η εφαρμογή σε ένα μηχάνημα, το Java Virtual Machine που πρέπει να είναι εγκατεστημένο σε αυτό θα αναλάβει να διαβάσει τα αρχεία .class. Στη συνέχεια τα μεταφράζει σε γλώσσα μηχανής που να υποστηρίζεται από το λειτουργικό σύστημα και τον επεξεργαστή, έτσι ώστε να εκτελεστεί (αυτό συμβαίνει με την παραδοσιακή Εικονική Μηχανή (Virtual Machine). Πιο σύγχρονες εφαρμογές της εικονικής Μηχανής μπορούν και μεταγλωττίζουν εκ των προτέρων τμήματα bytecode απευθείας σε κώδικα μηχανής (εγγενή κώδικα ή native code) με αποτέλεσμα να βελτιώνεται η ταχύτητα). Χωρίς αυτό δε θα ήταν δυνατή η εκτέλεση λογισμικού γραμμένου σε Java. Η JVM είναι λογισμικό που εξαρτάται από την πλατφόρμα, δηλαδή για κάθε είδος λειτουργικού συστήματος και αρχιτεκτονικής

επεξεργαστή υπάρχει διαφορετική έκδοση του. Έτσι υπάρχουν διαφορετικές JVM για Windows, Linux, Unix, Macintosh, κινητά τηλέφωνα, παιχνιδιομηχανές κλπ. Οτιδήποτε θέλει να κάνει ο προγραμματιστής (ή ο χρήστης) γίνεται μέσω της εικονικής μηχανής. Αυτό βοηθάει στο να υπάρχει μεγαλύτερη ασφάλεια στο σύστημα γιατί η εικονική μηχανή είναι υπεύθυνη για την επικοινωνία χρήστη - υπολογιστή. Ο προγραμματιστής δεν μπορεί να γράψει κώδικα ο οποίος θα έχει καταστροφικά αποτελέσματα για τον υπολογιστή γιατί η εικονική μηχανή θα τον ανιχνεύσει και δε θα επιτρέψει να εκτελεστεί. Από την άλλη μεριά ούτε ο χρήστης μπορεί να κατεβάσει «κακό» κώδικα από το δίκτυο και να τον εκτελέσει. Αυτό είναι ιδιαίτερα χρήσιμο για μεγάλα κατανεμημένα συστήματα όπου πολλοί χρήστες χρησιμοποιούν το ίδιο πρόγραμμα συγχρόνως.

### 5.1.5.2 Ο συλλέκτης απορριμμάτων (Garbage Collector)

Ακόμα μία ιδέα που βρίσκεται πίσω από τη Java είναι η ύπαρξη του συλλέκτη απορριμμάτων (Garbage Collector). Συλλογή απορριμμάτων είναι μία κοινή ονομασία που χρησιμοποιείται στον τομέα της πληροφορικής για να δηλώσει την ελευθέρωση τμημάτων μνήμης από δεδομένα που δε χρειάζονται και δε χρησιμοποιούνται άλλο. Αυτή η απελευθέρωση μνήμης στη Java είναι αυτόματη και γίνεται μέσω του συλλέκτη απορριμμάτων. Υπεύθυνη για αυτό είναι και πάλι η εικονική μηχανή η οποία μόλις «καταλάβει» ότι ο σωρός (heap) της μνήμης (στη Java η συντριπτική πλειοψηφία των αντικειμένων αποθηκεύονται στο σωρό σε αντίθεση με τη C++ όπου αποθηκεύονται κυρίως στη στοίβα) κοντεύει να γεμίσει ενεργοποιεί το συλλέκτη απορριμμάτων. Έτσι ο προγραμματιστής δε χρειάζεται να ανησυχεί για το πότε και αν θα ελευθερώσει ένα συγκεκριμένο τμήμα της μνήμης, ούτε και για σφάλματα δεικτών. Αυτό είναι ιδιαίτερα σημαντικό γιατί είναι κοινά τα σφάλματα προγραμμάτων που οφείλονται σε λανθασμένο χειρισμό της μνήμης.

### 5.1.5.3 Επιδόσεις

Παρόλο που η εικονική μηχανή προσφέρει όλα αυτά (και όχι μόνο) τα πλεονεκτήματα, η Java αρχικά ήταν πιο αργή σε σχέση με άλλες προγραμματιστικές γλώσσες υψηλού επιπέδου (high-level) όπως η C και η C++. Εμπειρικές μετρήσεις στο παρελθόν είχαν δείξει ότι η C++ μπορούσε να είναι αρκετές φορές γρηγορότερη από την Java. Ωστόσο γίνονται προσπάθειες από τη Sun για τη βελτιστοποίηση της εικονικής μηχανής, ενώ υπάρχουν και άλλες υλοποιήσεις της εικονικής μηχανής από διάφορες εταιρίες (όπως της IBM), οι οποίες μπορεί σε κάποια σημεία να προσφέρουν καλύτερα και σε κάποια άλλα χειρότερα αποτελέσματα. Επιπλέον με την καθιέρωση των μεταγλωττιστών JIT (Just In Time), οι οποίοι μετατρέπουν τον κώδικα byte απευθείας σε γλώσσα μηχανής, η διαφορά ταχύτητας από τη C++ έχει μικρύνει κατά πολύ.

Οι τελευταίες εκδόσεις του Javac με τη χρήση της τεχνολογίας Hot Spot έχουν καταφέρει αξιόλογες επιδόσεις που πλησιάζουν ή και ξεπερνούν σε μερικές περιπτώσεις τον εγγενή κώδικα.



### 5.1.6 Εργαλεία ανάπτυξης

Όλα τα εργαλεία που χρειάζεται κάποιος για να γράψει Java προγράμματα έρχονται δωρεάν, από το περιβάλλον ανάπτυξης μέχρι εργαλεία build όπως το Apache Ant και βιβλιοθήκες, ενώ υπάρχουν πολλές διαφορετικές υλοποιήσεις της Εικονικής Μηχανής και του μεταγλωττιστή (πχ the GNU Compiler for Java) της Java. Πολλά εργαλεία και τεχνολογίες σε Java μπορούν να βρεθούν στο Apache Software Foundation αλλά και στο Jakarta Project.

### 5.1.7 Ολοκληρωμένο περιβάλλον ανάπτυξης (IDE)

Για να γράψει κάποιος κώδικα Java δε χρειάζεται τίποτα άλλο παρά έναν επεξεργαστή κειμένου, όπως το Σημειωματάριο (Notepad) των Windows ή ο vi (γνωστός στο χώρο του Unix). Βέβαια ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) βοηθάει πολύ, ιδιαίτερα στον εντοπισμό σφαλμάτων (debugging). Υπάρχουν αρκετά διαθέσιμα, ενώ πολλά από αυτά έρχονται δωρεάν.

## 5.2 SQL (Structured Query Language)

### 5.2.1 Η ιστορία της SQL

Η SQL βασίστηκε σε ιδέες που προτάθηκαν στις αρχές του 1970 από τον ερευνητή Tedd Codd των εργαστηρίων IBM San Josi Research Laboratories. Δημιουργήθηκε με σκοπό να παρέχει μια ημιφυσική γλώσσα για το IBM System Relational database system. Αρχικά οι σχεσιακές βάσεις δεδομένων είχαν πολλούς και πιο αποδοτικούς ανταγωνιστές όπως συστήματα διαχείρισης δεδομένων τα οποία βασίζονταν σε δικτυακά μοντέλα δεδομένων. Ωστόσο, όλο και πιο αποδοτικά σχεσιακά συστήματα άρχισαν να εμφανίζονται στα τέλη του 1980 κι αυτό οδήγησε στην επικράτηση των σχεσιακών βάσεων δεδομένων και της SQL.

### 5.2.2 DML (Data Manipulation Language) DDL (Data Definition Language) DCL (Data Control Language)

Η δομημένη γλώσσα ερωταποκρίσεων SQL (Structured Query Language, SQL) χρησιμοποιείται για τη διαχείριση των δεδομένων μιας βάσης δεδομένων. Η διαχείριση περιλαμβάνει τόσο τη δημιουργία και μεταβολή των πινάκων της εφαρμογής, όσο και την καταχώρηση και ανάκτηση δεδομένων με βάση συγκεκριμένα κριτήρια επιλογής. Μια τυπική γλώσσα SQL περιλαμβάνει τις επόμενες δομικές μονάδες:

- Γλώσσα ορισμού δεδομένων (Data Definition Language, DDL)  
Η γλώσσα αυτή περιλαμβάνει εντολές που μας επιτρέπουν να υλοποιήσουμε πίνακες, σχέσεις ανάμεσα σε πίνακες και γενικά όλη τη δομή μιας βάσης δεδομένων.
- Γλώσσα χειρισμού δεδομένων (Data Manipulation Language, DML)  
Η γλώσσα αυτή επιτρέπει τη διαχείριση των δεδομένων της εφαρμογής, όπως την εισαγωγή, διαγραφή, ανάκτηση και τροποποίηση δεδομένων.
- Ορισμός όψεων της βάσης (View Definition)

Επιτρέπει τη δημιουργία όψεων της βάσης δεδομένων οι οποίες ορίζονται ως εικονικοί πίνακες (virtual tables) οι οποίοι περιέχουν δεδομένα από ένα ή περισσότερους πίνακες της βάσης.

- Ορισμός εξουσιοδοτήσεων (Authorization)

Επιτρέπει τη δημιουργία ομάδων χρηστών και την απόδοση διαφορετικών δικαιωμάτων πρόσβασης σε κάθε έναν από αυτούς, προκειμένου η κάθε ομάδα χρηστών, να διαχειρίζεται μόνο τα δικά της δεδομένα.

- Διαχείριση ακεραιότητας (Integrity)

Επιτρέπει το λεπτομερή έλεγχο των δεδομένων που καταχωρούνται στη βάση, έτσι ώστε να μην παραβιάζονται οι κανόνες ακεραιότητας (integrity constraints) που έχουμε ορίσει και οι οποίοι όταν τηρούνται απομακρύνουν τον κίνδυνο καταχώρησης ασυνεπών δεδομένων (inconsistent data).

- Η γλώσσα διαχείρισης δεδομένων (Data Control Language, DCL) χειρίζεται τις εξουσιοδοτήσεις των δεδομένων.

### 5.2.3 DML Γλώσσα Διαχείρισης Δεδομένων

Η γλώσσα χειρισμού δεδομένων (Data Manipulation Language DML), επιτρέπει την διαχείριση των δεδομένων των πινάκων της βάσης, και πιο συγκεκριμένα, την εισαγωγή, διαγραφή και τροποποίηση των εγγραφών των πινάκων. Επιπλέον, έχουμε την δυνατότητα να ανακτήσουμε από τους πίνακες, δεδομένα τα οποία πληρούν κάποια κριτήρια. Η πραγματοποίηση αυτών των διαδικασιών, γίνεται χρησιμοποιώντας τις εντολές INSERT, DELETE, UPDATE και SELECT.

- Η εντολή SELECT

Η δύναμή της SQL είναι η εντολή SELECT. Χρησιμοποιείται για να ανακαλεί δεδομένα κάνοντας ερωτήματα στην βάση δεδομένων, επιλέγοντας γραμμές από έναν πίνακα, που ταιριάζουν με συγκεκριμένα κριτήρια.

Η σύνταξη της SELECT: SELECT <πεδία> FROM <πίνακες> WHERE <κριτήριο>

- Η εντολή INSERT, εισάγει μια ή και περισσότερες νέες γραμμές.

Η σύνταξη της INSERT: INSERT INTO EXTABLE VALUES (4,300,103);

Με αυτή την εντολή εισάγαγέ μια νέα γραμμή στον πίνακα EXTABLE. Όμως πρέπει να γνωρίζουμε τις στήλες του EXTABLE και την σειρά τους πριν τρέξουμε την παραπάνω εντολή.

- Η εντολή UPDATE, ενημερώνει τα δεδομένα σε μια ή περισσότερες σειρές.

Η σύνταξη της UPDATE: UPDATE EXTABLE SET EURO = EURO \* 10 WHERE AA = 3;

Η εντολή αλλάζει την 3η συναλλαγή.

- Η εντολή DELETE, διαγράφει μια ή περισσότερες γραμμές.

Σίγουρα αυτή η εντολή πρέπει να χρησιμοποιείται με προσοχή.

Η σύνταξη της DELETE: DELETE \* FROM EXTABLE WHERE ID = 104;

## 5.3 Php

Η PHP είναι μια γλώσσα script από την πλευρά του διακομιστή, σχεδιασμένη ειδικά για το Web. Μέσα σε μια HTML σελίδα μπορεί να ενσωματωθεί ο PHP κώδικας, που θα εκτελείται κάθε φορά που κάποιος θα επισκέπτεται τη σελίδα. Ο PHP κώδικας μεταφράζεται στο Web διακομιστή και δημιουργεί HTML ή άλλη έξοδο που θα δει ο επισκέπτης.

### 5.3.1 Ιστορία της PHP

Η ιστορία της PHP ξεκινά από το 1995, όταν ένας φοιτητής, ο Rasmus Lerdorf δημιούργησε, χρησιμοποιώντας τη γλώσσα προγραμματισμού Perl, ένα απλό script με όνομα php.cgi, για προσωπική χρήση. Το script αυτό είχε σαν σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημειώμα. Αργότερα αυτό το script το διέθεσε και σε φίλους του, οι οποίοι άρχισαν να του ζητούν να προσθέσει περισσότερες δυνατότητες. Η γλώσσα τότε ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter. Το 1997 η PHP/FI έφθασε στην έκδοση 2.0, βασιζόμενη αυτή τη φορά στη γλώσσα C και αριθμώντας περισσότερους από 50.000 ιστότοπους που τη χρησιμοποιούσαν, ενώ αργότερα την ίδια χρονιά οι Andi Gutmans και Zeev Suraski ξαναέγραψαν τη γλώσσα από την αρχή, βασιζόμενοι όμως αρκετά στην PHP/FI 2.0. Έτσι η PHP έφθασε στην έκδοση 3.0 η οποία θύμιζε περισσότερο τη σημερινή μορφή της. Στη συνέχεια, οι Zeev και Andi δημιούργησαν την εταιρεία Zend (από τα αρχικά των ονομάτων τους), η οποία συνεχίζει μέχρι και σήμερα την ανάπτυξη και εξέλιξη της γλώσσας PHP. Ακολούθησε το 1998 η έκδοση 4 της PHP, τον Ιούλιο του 2004 διατέθηκε η έκδοση 5 και το 2008 κυκλοφόρησε η έκδοση 6. Η επόμενη έκδοση (7) διατέθηκε το 2014, ενώ πριν λίγες εβδομάδες κυκλοφόρησε η έκδοση 8. Οι περισσότεροι προγραμματιστές και συστήματα σήμερα χρησιμοποιούν την έκδοση 7.

### 5.3.2 Τι είναι η PHP

Η PHP είναι μια γλώσσα προγραμματισμού που σχεδιάστηκε για τη δημιουργία δυναμικών σελίδων στο διαδίκτυο και είναι επισήμως γνωστή ως: HyperText preprocessor. Είναι μια server-side (εκτελείται στον διακομιστή) scripting γλώσσα που γράφεται συνήθως πλαισιωμένη από HTML, για μορφοποίηση των αποτελεσμάτων. Αντίθετα από μια συνηθισμένη HTML σελίδα, η σελίδα PHP δεν στέλνεται άμεσα σε έναν πελάτη (client), αντί αυτού πρώτα αναλύεται και μετά αποστέλλεται το παραγόμενο αποτέλεσμα. Τα στοιχεία HTML στον πηγαίο κώδικα μένουν ως έχουν, αλλά ο PHP κώδικας ερμηνεύεται και εκτελείται. Ο κώδικας PHP μπορεί να θέσει ερωτήματα σε βάσεις δεδομένων, να δημιουργήσει εικόνες, να διαβάσει και να γράψει αρχεία, να συνδεθεί με απομακρυσμένους υπολογιστές, κ.ο.κ. Σε γενικές γραμμές οι δυνατότητες που μας δίνει είναι απεριόριστες.

### 5.3.3 Πλεονεκτήματα της PHP

Κάποιοι από τους βασικούς ανταγωνιστές της PHP είναι ο Perl, Microsoft Active Server Pages (ASP), Java Server Pages (JSP) και Allaire Cold Fusion.

Σε σύγκριση με αυτά τα προϊόντα, η PHP έχει πολλά πλεονεκτήματα όπως :

1. Υψηλή απόδοση
2. Διασυνδέσεις με πολλά διαφορετικά συστήματα βάσεων δεδομένων
3. Ενσωματωμένες βιβλιοθήκες για πολλές συνηθισμένες Web διαδικασίες
4. Χαμηλό κόστος
5. Ευκολία μάθησης και χρήσης
6. Διαθεσιμότητα του κώδικα προέλευσης

Ακολουθεί μια πιο λεπτομερή περιγραφή αυτών των πλεονεκτημάτων.

#### 1. Απόδοση

Η PHP είναι πολύ αποτελεσματική. Με ένα φθινό διακομιστή μπορούν να εξυπηρετηθούν εκατομμύρια επισκέψεων καθημερινά. Οι δοκιμές που δημοσιεύθηκαν από την Zend Technologies ( <http://www.zend.com>), δείχνουν ότι η PHP ξεπερνά τους ανταγωνιστές της.

#### 2. Ολοκλήρωση με Βάσεις δεδομένων

Η PHP έχει εγγενείς συνδέσεις για πολλά συστήματα βάσεων δεδομένων. Εκτός από την MySQL, μπορεί κάποιος να συνδεθεί κατευθείαν με τις βάσεις δεδομένων PostgreSQL , mSQL , Oracle , dbm , filePro , Informix , InterBase , Sybase, μεταξύ άλλων. Χρησιμοποιώντας το Open Database Connectivity Standard (ODBC) μπορεί να συνδεθεί σε οποιαδήποτε βάση δεδομένων παρέχει ένα πρόγραμμα οδήγησης ODBC .

Αυτό περιλαμβάνει και τα προϊόντα της Microsoft.

#### 3. Ενσωματωμένες Βιβλιοθήκες

Επειδή η PHP σχεδιάστηκε για να χρησιμοποιείται στο Web, έχει πολλές ενσωματωμένες βιβλιοθήκες που εκτελούν πολλές χρήσιμες λειτουργίες σχετικές με το Web. Μπορεί κάποιος να δημιουργήσει εικόνες GIF δυναμικά, να συνδεθεί με άλλες υπηρεσίες δικτύων, να στείλει ηλεκτρονικό ταχυδρομείο, να δουλέψει με cookies και να δημιουργήσει PDF έγγραφα : όλα αυτά με λίγες γραμμές κώδικα.

#### 4. Κόστος

Η PHP είναι δωρεάν. Μπορεί κάποιος να λήψη της τελευταίας έκδοσης από [ΕΛΩ](#), χωρίς χρέωση.

### 5.3.4 Που Χρησιμοποιείται ένα PHP script.

Υπάρχουν τρεις κύριοι τομείς που χρησιμοποιείται ένα PHP script.

- Server-side scripting.

Αυτό είναι το πιο παραδοσιακό και το κύριο πεδίο για την PHP. Απαιτούνται τρία πράγματα για να δουλέψει αυτό. Τον PHP μεταγλωττιστή, ένα webserver (εξυπηρετητή σελίδων) και ένα web browser (φυλλομετρητή). Πρέπει κάποιος να 'τρέξει' τον webserver, με μια συνδεδεμένη εγκατάσταση της PHP. Μπορεί να προσπελάσει τα αποτελέσματα του PHP προγράμματος με ένα web browser, βλέποντας την σελίδα PHP μέσα από τον server.

- Command line scripting.

Μπορεί κάποιος να φτιάξει ένα PHP script για να το «τρέξει» χωρίς server ή browser. Απαιτείται μόνο ο PHP μεταγλωττιστής για να χρησιμοποιηθεί με αυτό τον τρόπο. Αυτός ο τύπος είναι ιδανικός για script που εκτελούνται συχνά με τη χρήση της cron (σε Unix ή Linux) ή με τον Task Scheduler (στα Windows). Αυτά τα script μπορούν επίσης να χρησιμοποιηθούν για απλές εργασίες επεξεργασίας κειμένου.

- Εγγραφή client-side GUI εφαρμογών (Γραφικά περιβάλλοντα χρηστών).

Η PHP ίσως να μην είναι η πιο καλή γλώσσα για να γράψει κανείς παραθυρικές εφαρμογές, αλλά αν γνωρίζει την PHP πολύ καλά και θέλει να χρησιμοποιήσει κάποια προχωρημένα χαρακτηριστικά της PHP στις client-side εφαρμογές, μπορεί επίσης να χρησιμοποιήσει το PHP-GTK για αυτού του είδους τα προγράμματα. Έχει επίσης τη δυνατότητα να γράφει cross-platform εφαρμογές με αυτό τον τρόπο. Το PHP-GTK είναι μια επέκταση της PHP και δεν συμπεριλαμβάνεται στην κύρια διανομή. Αν κάποιος ενδιαφέρεται για το PHP-GTK, θα πρέπει να επισκεφθεί την δική του ιστοσελίδα.

Επίσης, η PHP μπορεί να χρησιμοποιηθεί σε όλα τα κύρια λειτουργικά συστήματα, συμπεριλαμβανομένου του Linux, πολλών εκδοχών του Unix (HP-UX, Solaris και OpenBSD), Microsoft Windows, Mac OS X, RISC OS και πιθανώς σε άλλα. Η PHP υποστηρίζει επίσης τους Apache, Microsoft Internet Information Server, Personal Web Server, Netscape και iPlanet servers, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd, και πολλούς άλλους webserver. Για την πλειοψηφία των server η PHP έχει ένα module, για τους υπόλοιπους η PHP μπορεί να λειτουργήσει ως ένας CGI επεξεργαστής. Έτσι με την PHP, υπάρχει η ελευθερία επιλογής ενός λειτουργικού συστήματος και ενός web server. Επιπλέον, μπορεί κάποιος ελεύθερα να χρησιμοποιήσει συναρτησιακό (procedural) ή αντικειμενοστραφή (object oriented) προγραμματισμό ή μια ανάμειξη τους. Αν και η παρούσα έκδοση δεν υποστηρίζει όλα τα πρότυπα χαρακτηριστικά, μεγάλες βιβλιοθήκες κώδικα και μεγάλες εφαρμογές (συμπεριλαμβανομένης και της βιβλιοθήκης PEAR) είναι γραμμένες μόνο με αντικειμενοστραφή κώδικα. Με την PHP δεν είναι κάποιος περιορισμένος να εξάγει HTML. Οι δυνατότητες της PHP συμπεριλαμβάνουν την εξαγωγή εικόνων, αρχείων PDF, ακόμη και ταινίες Flash (χρησιμοποιώντας τα libswf και Ming) παράγονται αμέσως. Μπορεί επίσης να πραγματοποιηθεί εύκολα οποιοδήποτε κείμενο όπως XHTML και

οποιοδήποτε άλλο XML αρχείο. Η PHP μπορεί να δημιουργεί αυτόματα αυτά τα αρχεία και να τα αποθηκεύει στο σύστημα αρχείων, αντί να τα εκτυπώνει, αποτελώντας έτσι μια server-side cache για το δυναμικό σας περιεχόμενο. Ένα από τα πιο δυνατά και σημαντικά χαρακτηριστικά της PHP είναι η υποστήριξη που έχει για ένα μεγάλο σύνολο βάσεων δεδομένων. Η συγγραφή μιας σελίδας που υποστηρίζει βάσεις δεδομένων είναι εξαιρετικά απλή.

### 5.3.5 Τι μπορεί να κάνει η PHP.

Οτιδήποτε. Η PHP επικεντρώνεται κυρίως στο server-side scripting, έτσι μπορεί να κάνει τα ίδια πράγματα με οποιοδήποτε άλλο CGI πρόγραμμα, όπως να μαζέψει δεδομένα, να παράγει δυναμικό περιεχόμενο σελίδων, ή να στείλει και να πάρει cookies.

### 5.3.6 Συναρτήσεις.

Στην PHP, για να «μμεταφερθούν» οι τιμές που έχουν οι μεταβλητές μας από μία σελίδα σε μία άλλη, πρέπει να αποθηκευτούν προσωρινά. Αυτό γίνεται με 2 μεθόδους την GET και την POST.

#### 5.3.6.1 GET

Όταν χρησιμοποιείται η μέθοδος GET τα δεδομένα είναι σε μορφή text και είναι ορατά από τον καθένα στο URL που καλεί το script. Επιπλέον η ποσότητα των δεδομένων που μπορούν να μεταδοθούν με τη χρήση της μεθόδου GET είναι περιορισμένη. Έτσι για να περάσουμε μια μεταβλητή που θα μας δίνει το όνομα κάποιου γίνεται ως εξής: (page.php?name=Alex).

#### 5.3.6.2 POST

Όταν χρησιμοποιείται η μέθοδος POST τα δεδομένα μεταδίδονται με ξεχωριστή σύνδεση στο script και δεν είναι ορατά στο URL που καλεί το script. Επιπλέον η ποσότητα των δεδομένων που μπορούν να μεταδοθούν με τη χρήση της μεθόδου POST. Τα δεδομένα στέλνονται κυρίως από φόρμες, όπου μέσα στο \$\_POST[""] βάζουμε το όνομα (name) του πεδίου που μας στέλνει η φόρμα. Οι πληροφορίες που στέλνονται από την φόρμα μας με την μέθοδο POST είναι «αόρατες», και δεν έχουν όρια στο μέγεθος των πληροφοριών που μπορούν να στείλουν.

## 5.4 Xml

Οι διατάξεις του Android είναι γραμμένες στη γλώσσα eXtensible Markup Language, επίσης γνωστή και ως XML. Όπως η HTML (ή HyperText Markup Language), η XML είναι

επίσης μια γλώσσα σήμανσης. Δημιουργήθηκε ως τυπικός τρόπος κωδικοποίησης δεδομένων σε εφαρμογές που βασίζονται στο Διαδίκτυο. Ωστόσο, σε αντίθεση με την HTML, η XML είναι ευαίσθητη στα πεζά και κεφαλαία γράμματα, απαιτεί κάθε tag να κλείνει σωστά και διατηρεί τα κενά διαστήματα.

Οι διατάξεις Android XML αποτελούν επίσης μέρος μιας μεγαλύτερης ομπρέλας αρχείων και στοιχείων Android που ονομάζονται πόροι. Οι πόροι είναι τα πρόσθετα αρχεία και το στατικό περιεχόμενο που χρειάζεται μια εφαρμογή, όπως κινούμενα σχέδια, χρωματικοί συνδυασμοί, διατάξεις, διατάξεις μενού.

### 5.4.1 Ανατομία Διατάξεων Android XML

Κάθε αρχείο διάταξης πρέπει να περιέχει ένα (και μόνο ένα!) ριζικό στοιχείο. Τα Linear Layouts, τα Relative Layouts και Frame Layouts μπορούν να είναι ριζικά στοιχεία. Άλλες διατάξεις ενδέχεται να μην είναι. Όλα τα άλλα στοιχεία XML θα βρίσκονται μέσα σε αυτό το ριζικό αντικείμενο.

Ένα στοιχείο View είναι απλώς ένα αντικείμενο από την ενσωματωμένη κλάση View του Android. Αντιπροσωπεύει μια ορθογώνια περιοχή της οθόνης και είναι υπεύθυνη για την εμφάνιση πληροφοριών ή περιεχομένου, καθώς και για το χειρισμό συμβάντων. Το κείμενο, οι εικόνες και τα κουμπιά είναι όλα τύπου View στο Android.

Η Ομάδα Προβολών (ViewGroup) είναι μια υποκατηγορία του View και ουσιαστικά είναι ένα «αόρατο κιβώτιο» που συγκρατεί πολλές προβολές View ή ομάδες προβολών (ViewGroups) μαζί και καθορίζει τις ιδιότητες διάταξής τους.

#### Ριζικές Προβολές (Root Views)

- Η Γραμμική Διάταξη (Linear Layout) ευθυγραμμίζει τα περιεχόμενά της σε μία μόνο κατεύθυνση, κάθετη ή οριζόντια.
- Η Σχετική Διάταξη (Relative Layout) εμφανίζει το περιεχόμενο του παιδιού της σε θέσεις σχετικές με τον γονέα. (δηλαδή: παράταξη ενός στοιχείου μέχρι το πάνω άκρο ενός γονέα, κεντράρισμα του εντός ενός γονέα κ.λπ.)
- Η Διάταξη Πλαισίου (Frame Layout) είναι ένας χώρος σε μια οθόνη που μπορεί να εμφανίσει μόνο μία προβολή. Για αυτόν τον λόγο, οι Διατάξεις πλαισίου θα πρέπει να χρησιμοποιούνται με προσοχή. Συνήθως περιέχουν στοιχεία όπως Fragments

#### Ομάδες Προβολής (ViewGroups)

Άλλες κοινά χρησιμοποιούμενες Ομάδες Προβολής ViewGroups έχουν ως εξής. Αυτές μπορεί να μην είναι το ριζικό στοιχείο μιας διάταξης, αλλά μπορεί να βρίσκονται εντός του ριζικού στοιχείου:

- Η Προβολή Λίστας (List View) εμφανίζει μια λίστα με αντικείμενα με δυνατότητα κύλισης.
- Η Προβολή Πλέγματος (Grid View) εμφανίζει στοιχεία σε ένα δισδιάστατο, κυλιόμενο πλέγμα.
- Η Διάταξη Πίνακα (Table Layout) ομαδοποιεί τις προβολές σε σειρές και στήλες.

## Χαρακτηριστικά Διάταξης

Κάθε τύπος διάταξης έχει χαρακτηριστικά που καθορίζουν τον τρόπο εμφάνισης των στοιχείων του. Υπάρχουν τόσο κοινά χαρακτηριστικά που μοιράζονται όλες οι διατάξεις όσο και συγκεκριμένα χαρακτηριστικά για ορισμένους από τους τύπους διατάξεων που αναφέρονται παραπάνω. Τα ακόλουθα είναι χαρακτηριστικά που ισχύουν για όλες τις διατάξεις:

- **Android:id**: Ένα μοναδικό αναγνωριστικό που αντιστοιχεί στην προβολή.
- **Android:layout\_width**: Το πλάτος της διάταξης. Απαιτείται για κάθε προβολή (view).
- **Android:layout\_height**: Το ύψος της διάταξης. Απαιτείται για κάθε προβολή (view).
- **Android:layout\_marginTop**: Επιπλέον χώρος στην κορυφή της διάταξης.
- **Android:layout\_marginBottom**: Επιπλέον χώρος στο κάτω μέρος της διάταξης.
- **Android:layout\_marginLeft**: Επιπλέον χώρος στα αριστερά της διάταξης.
- **Android:layout\_marginRight**: Επιπλέον χώρος στα δεξιά της διάταξης.
- **Android:layout\_weight**: Καθορίζει πόσος επιπλέον χώρος στη διάταξη πρέπει να διατεθεί στην προβολή.
- **Android:paddingLeft**: Επένδυση χώρου στα αριστερά της προβολής.
- **Android:paddingRight**: Επένδυση χώρου στα δεξιά της προβολής.
- **Android:paddingTop**: Επένδυση χώρου στην κορυφή της προβολής.
- **Android:paddingBottom**: Επένδυση στο κάτω μέρος της προβολής.

## Σχετικό Μέγεθος

Οι ιδιότητες ύψους και πλάτους μπορούν να ρυθμιστούν σε συγκεκριμένες μετρήσεις, αλλά οι ακόλουθοι ορισμοί είναι περισσότερο κοινοί:

- **Android:layout\_width=wrap\_content**: Ορίζει το πλάτος της προβολής σε οποιοδήποτε μέγεθος απαιτείται από τα περιεχόμενά της. Αυτό μπορεί επίσης να χρησιμοποιηθεί με το ύψος height.
- **Android:layout\_width=match\_parent**: Ορίζει το πλάτος της προβολής στο πλάτος του γονέα της. Αυτό μπορεί επίσης να χρησιμοποιηθεί με το ύψος height.

## Ρύθμιση Ιδιοτήτων και Αναγνωριστικών IDs

Όπως και στην HTML, μπορούμε επίσης να προσθέσουμε χαρακτηριστικά id σε στοιχεία XML. Συχνά αυτό είναι απαραίτητο αφού έπειτα βοηθάει στον εντοπισμό και την αλληλεπίδραση μιας συγκεκριμένης προβολής.

Η εκχώρηση ενός χαρακτηριστικού id είναι κάπως έτσι:

```
Android:id="@+id/search_button"
```



Εδώ, το τμήμα `@+id/` υποδεικνύει ότι το όνομα που ακολουθεί είναι μια συμβολοσειρά που περιέχει το αναγνωριστικό αυτού του στοιχείου και ότι είναι ένας νέος πόρος που πρέπει να δημιουργηθεί και να προστεθεί στους πόρους μας.

Για παράδειγμα, η ακόλουθη XML θα δημιουργήσει ένα Κουμπί με το αναγνωριστικό `search_button`:

```
<Button Android:id="@+id/search_button"
    Android:layout_width="wrap_content"
    Android:layout_height="wrap_content"
    Android:text="@string/my_button_text"/>
```

### Στόχευση Προβολών (Views) ανά Αναγνωριστικό ID

Για να στοχεύσουμε μεμονωμένα στοιχεία XML και να αλληλοεπιδράσουμε μαζί τους θα πρέπει να αναζητηθεί το συγκεκριμένο ID της προβολής (View) όπως φαίνεται στο παρακάτω παράδειγμα:

```
Button myButton = (Button) findViewById(R.id.search_button);
```

Η μέθοδος `findViewById()` μπορεί να εντοπίσει τόσο τις προβολές Views όσο και τις ριζικές προβολές αρκεί να χρησιμοποιήσουμε ξανά το ίδιο χαρακτηριστικό `id` που εκχωρήσαμε στην XML για να εντοπίσουμε αργότερα αυτό το στοιχείο στη Java.

### Σχετικά με τις Διατάξεις Περιορισμού

Οι νεότερες εκδόσεις του Android Studio ενδέχεται να δημιουργήσουν προεπιλεγμένες διατάξεις που ονομάζονται Διατάξεις Περιορισμού Constraint Layouts αντί για Γραμμικές Διατάξεις Linear Layouts ή Σχετικές Διατάξεις Relative Layouts. Τα Constraint Layouts είναι μια νεότερη μορφή διάταξης που είναι πιο ισχυρή, αλλά και πιο περίπλοκη για να δουλέψουν οι αρχάριοι.

Λάβετε υπόψη ότι απλά και μόνο επειδή υπάρχει μια νεότερη επιλογή, δυνατότητα ή έκδοση ενός εργαλείου ή μιας τεχνολογίας, δεν σημαίνει ότι όλες οι εταιρείες και τα προϊόντα μεταβαίνουν αμέσως στη νεότερη επιλογή. Ειδικά εάν οι παλαιότερες εκδόσεις ή δυνατότητες εξακολουθούν να εξυπηρετούν με επιτυχία τις ανάγκες τους. Αυτό ισχύει τόσο για τις διατάξεις Android όσο και για οποιαδήποτε άλλη τεχνολογία ανάπτυξης ιστού.

## 6 Ανάπτυξη Συστήματος

### 6.1 Φάσεις Ανάπτυξης Συστήματος

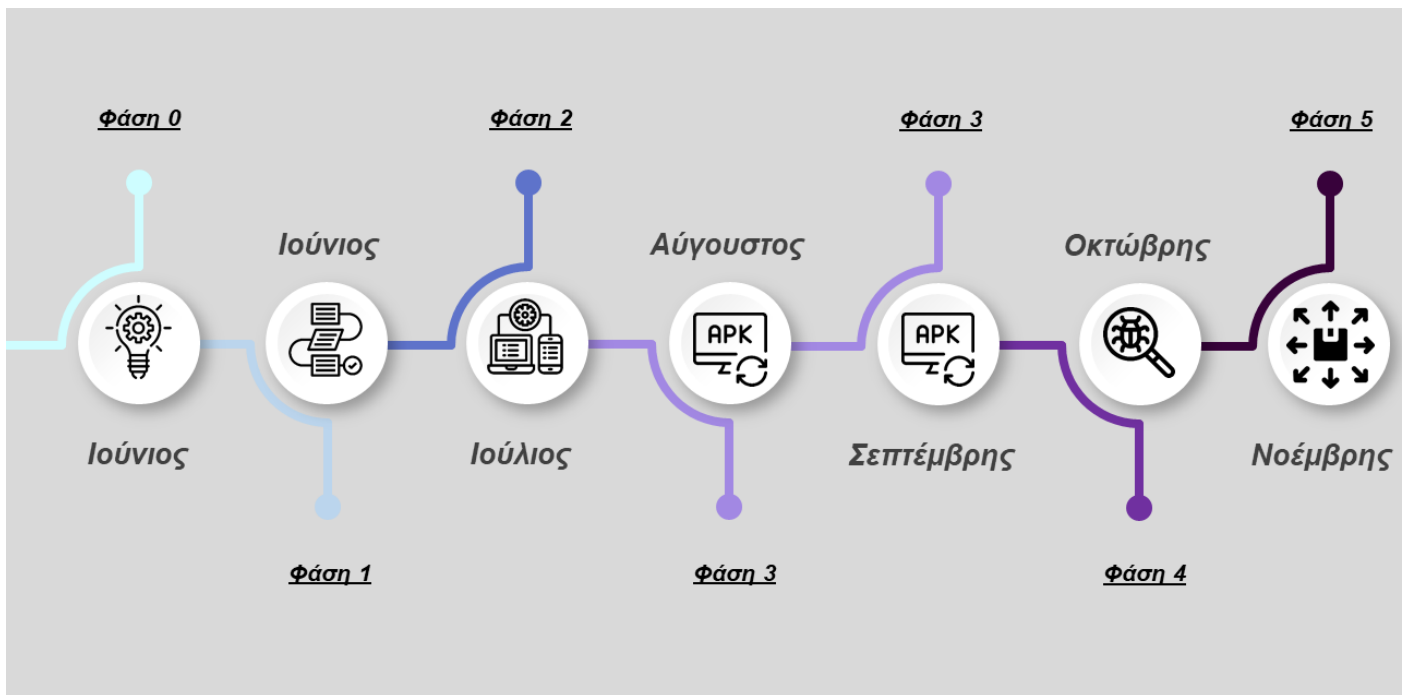
Στο κεφάλαιο αναλύονται οι φάσεις σχεδίασης, ανάπτυξης και υλοποίησης της εφαρμογής. Γενικά, πριν ξεκινήσει η ανάπτυξη οποιασδήποτε εφαρμογής, πρέπει να πραγματοποιείται η ανάλυση απαιτήσεων χρήστη και ικανοποίηση απαιτήσεων της εφαρμογής. Μόνο μετά από αυτά λαμβάνει χώρα η σχεδίαση και εν συνεχεία η υλοποίηση της τελικής εφαρμογής. Τέλος, έχουμε τη φάση του τελικού ελέγχου και αποσφαλμάτωσης του συστήματος.

Στην συνέχεια αναλύθηκαν οι φάσεις ανάπτυξης του συστήματος και ο χρονοπρογραμματισμός του έργου.

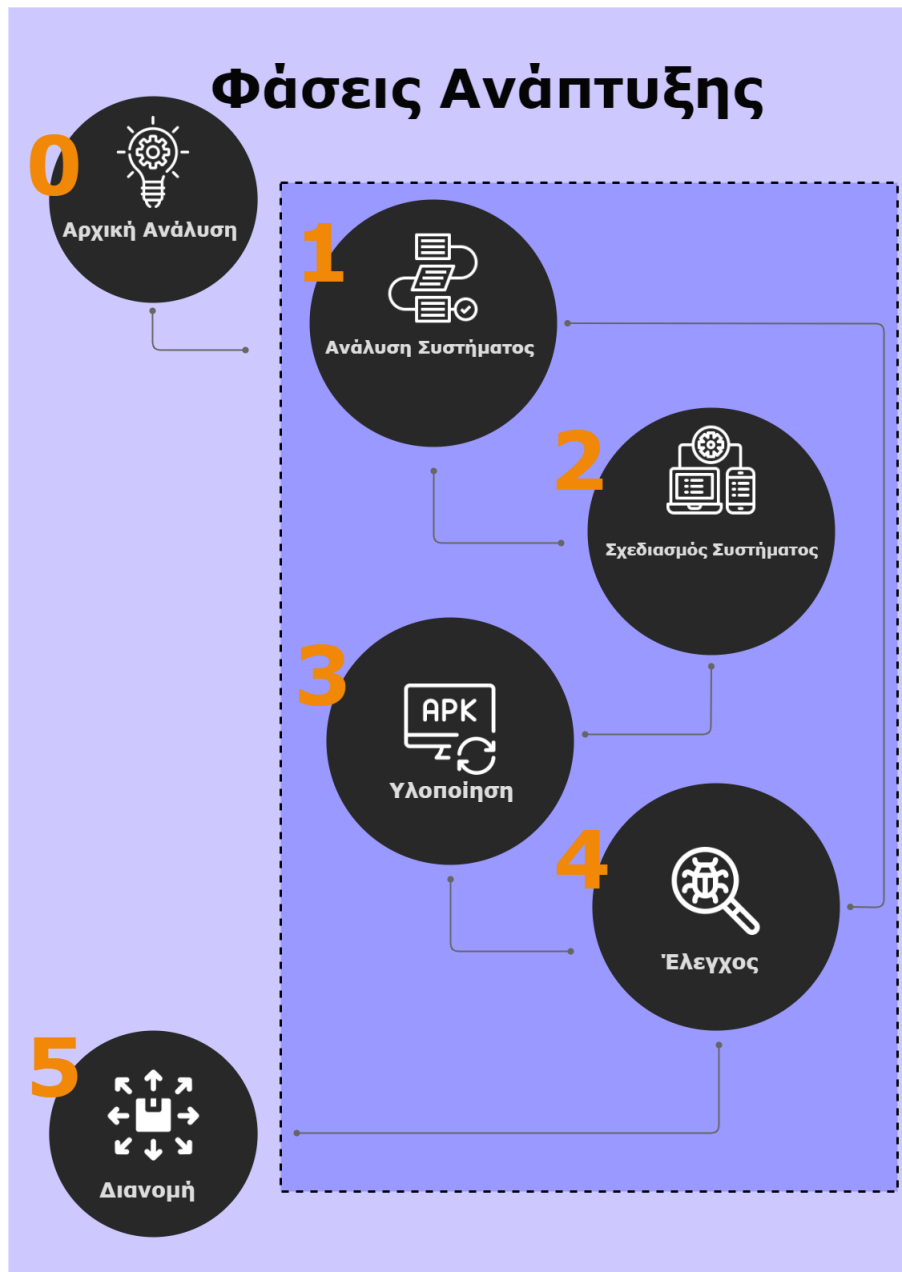
Φάσεις Ανάπτυξης	
Φάση 0 (Αρχική Ανάλυση):	Καταγραφή βασικών ιδεών
	Χρονοπρογραμματισμός και πλάνο εξέλιξης Συστήματος
Φάση 1 (Ανάλυση Συστήματος):	Ανάλυση Απαιτήσεων Συσσκευής
	Ανάλυση Απαιτήσεων Εφαρμογής
	Ανάλυση Απαιτήσεων Χρήστη
Φάση 2 (Σχεδιασμός Συστήματος):	Σχεδιασμός Αρχιτεκτονικής Συστήματος
	Σχεδιασμός Διεπαφής Χρήστη
	Συγκέντρωση απαραίτητων βιβλιοθηκών και υλικού
	Επεξεργασία απαραίτητων βιβλιοθηκών και υλικού
Φάση 3 (Υλοποίηση):	Υλοποίηση Εφαρμογής
	Υλοποίηση Βάσης Δεδομένων
	Υλοποίηση Σύνδεσης Συστήματος
Φάση 4 (Έλεγχος):	Έλεγχος λειτουργίας Εφαρμογής
	Έλεγχος λειτουργίας Βάσης Δεδομένων
	Έλεγχος καθολικής λειτουργίας Συστήματος
	Εφαρμογή διορθώσεων
Φάση 5 (Διανομή):	Διανομή Συστήματος στους Χρήστες

Εικόνα 15: Φάσεις Ανάπτυξης

## Χρονοπρογραμματισμός Έργου



Εικόνα 16: Χρονοπρογραμματισμός Έργου



Εικόνα 17: Διάγραμμα Ροής Φάσεων

### 6.1.1 Αρχική Ανάλυση

Η πρώτη φάση ανάπτυξης του συστήματός μας, αποτελεί την σύλληψη της γενικής ιδέας του συστήματος. Έτσι με την αναγνώριση του αρχικού προβλήματος μπορούμε πιο εύκολα να διευρύνουμε τις ιδέες μας και να εστιάσουμε στους τρόπους με τους οποίους το πρόβλημα θα μειωθεί και ιδανικά θα εξαλειφθεί. Ακόμη η παρούσα φάση περιλαμβάνει και το αρχικό σχεδιάγραμμα των φάσεων που είναι απαραίτητες, καθώς και των διαδικασιών της κάθε φάσης, που θα πρέπει να πραγματοποιηθούν για την ολοκλήρωση της υλοποίησης του συστήματος.

### 6.1.2 Ανάλυση Συστήματος

Στην δεύτερη φάση ανάπτυξης παρουσιάζεται η ανάλυση του συστήματος καθώς και μια αρχική προσέγγιση των βημάτων υλοποίησης της. Αποτελεί μια από τις σημαντικότερες ώστε να προχωρήσουμε ορθά στη σχεδίαση και στην υλοποίηση του τελικού συστήματος.

Στην παρούσα φάση της εφαρμογής θα καθοριστούν οι κύριοι στόχοι του συστήματος της καθώς και θα συγκεντρωθούν οι απαιτήσεις χρηστών. Παρακάτω ακολουθούν βοηθητικοί πίνακες που θα συντελέσουν στην περαιτέρω οργάνωση της δομής του έργου.

Ανάλυση Απαιτήσεων Χρήστη	
Ηλικία	> 18
Γλώσσα	Αγγλικά, Ελληνικά
Εμπειρία με Smartphones	ΝΑΙ
Εμπειρία με Παρόμοιες Εφαρμογές	ΌΧΙ
Προγραμματιστικές Γνώσεις	ΌΧΙ

Ανάλυση Απαιτήσεων Εφαρμογής	
Ικανότητα σάρωσης γραμμωτού κώδικα	ΝΑΙ
Ικανότητα αποθήκευσης δεδομένων σε αρχείο	ΝΑΙ
Ικανότητα διαχείρισης αρχείων	ΝΑΙ
Σύνδεση με Βάση δεδομένων	ΝΑΙ
Δυνατότητα αλλαγής γλώσσας	ΝΑΙ

Ανάλυση Απαιτήσεων Συσκευής	
Smartphone	ΝΑΙ
Έκδοση Android	> 5.0
Κάμερα	ΝΑΙ
Πρόσβαση στο διαδίκτυο	ΝΑΙ
Διαθέσιμο αποθηκευτικό χώρο	> 35MB

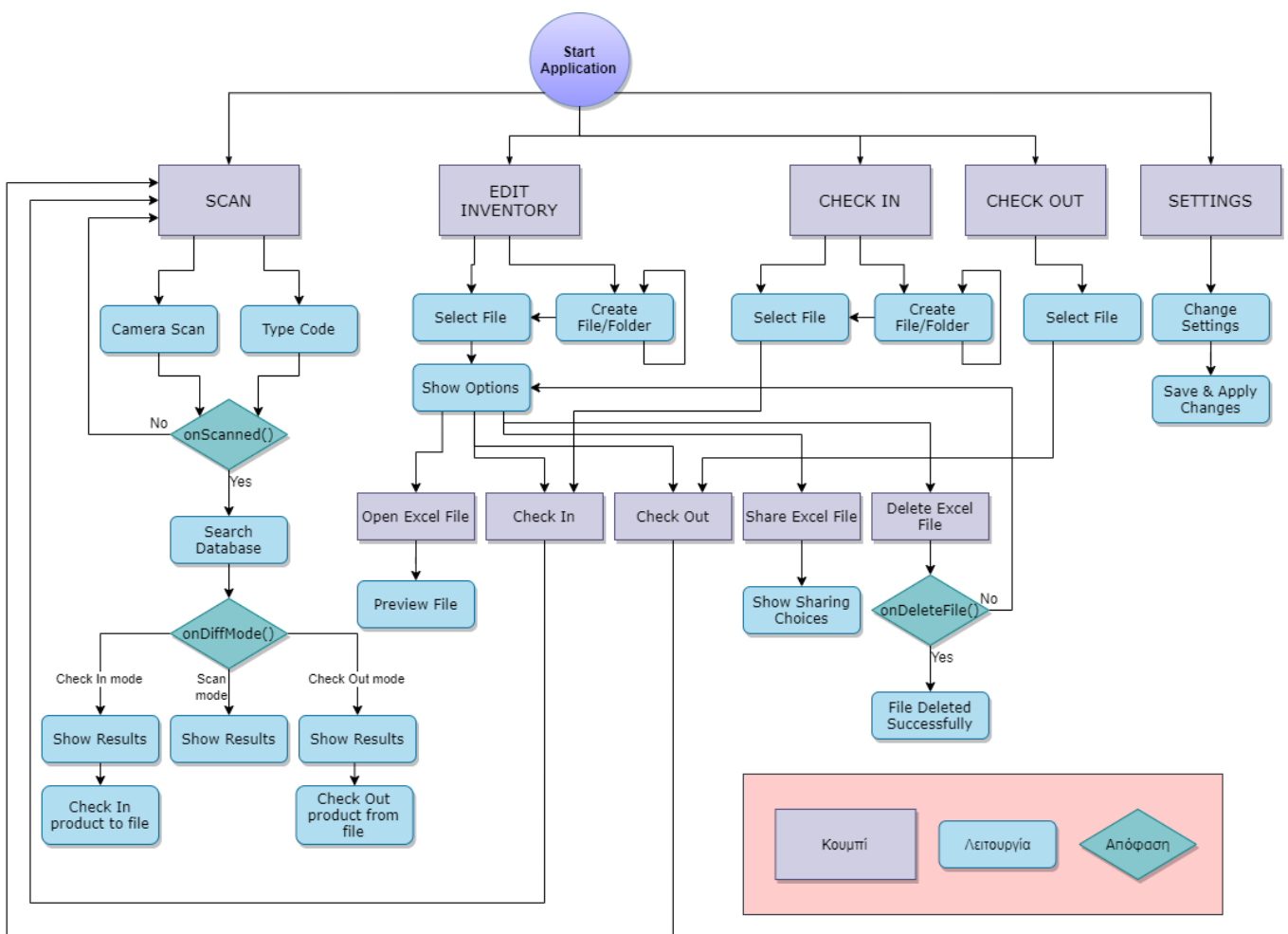
### 6.1.3 Σχεδιασμός Συστήματος

Η φάση του σχεδιασμού του Συστήματος αποτελεί τη τρίτη φάση ανάπτυξης του Συστήματος και περιλαμβάνει την σχεδίαση της αρχιτεκτονικής και την σχεδίαση της διεπαφής του χρήστη. Οι γενικοί στόχοι και οι αρχές που καθορίστηκαν στη φάση της ανάλυσης μετατρέπονται σε μια ολοκληρωμένη αναλυτική περιγραφή της εφαρμογής.

### 6.1.3.1 Σχεδίαση Αρχιτεκτονικής

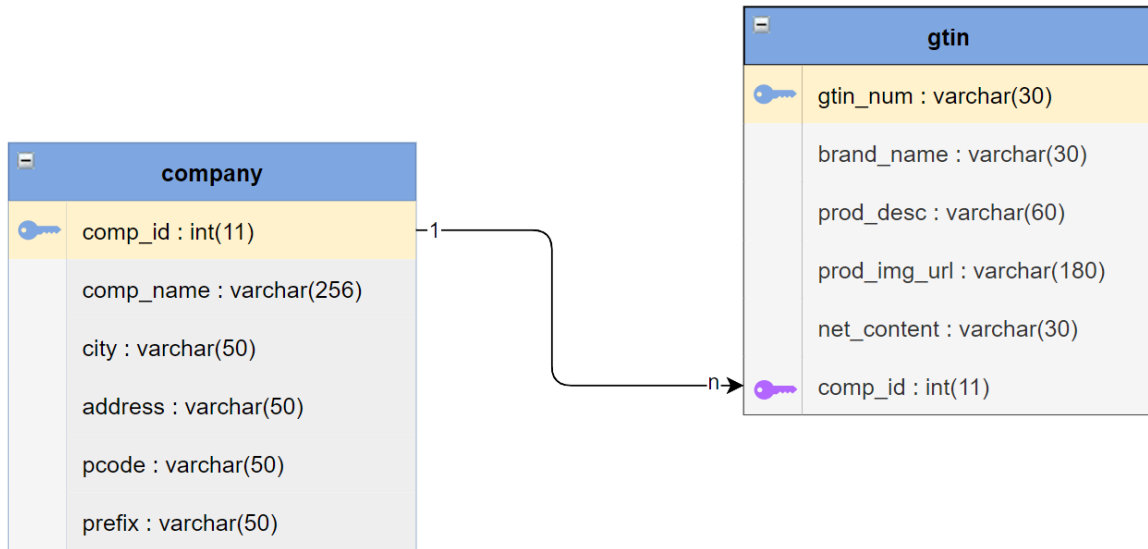
Στην φάση της σχεδίασης της αρχιτεκτονικής, υλοποιείται η βάση δεδομένων του συστήματος καθώς και το διάγραμμα λειτουργίας της εφαρμογής, αφού πρώτα λάβουμε υπόψιν τις απαιτήσεις που αναλύθηκαν στην πρώτη φάση. Η ανάπτυξη της βάσης δεδομένων πραγματοποιήθηκε με τη γλώσσα προγραμματισμού SQL ενώ η ανάπτυξη της εφαρμογής πραγματοποιήθηκε κατά κόρον με τη γλώσσα προγραμματισμού Java. Για τη διαγραμματική απεικόνιση τόσο της βάσης δεδομένων και της δομής της εφαρμογής, χρησιμοποιήθηκε το εργαλείο δημιουργίας διαγραμμάτων [Draw.io](http://Draw.io).

Για την ευκολότερη κατανόηση της δομής της εφαρμογής και της βάσης δεδομένων βλέπουμε παρακάτω τις αντίστοιχες εικόνες:



Εικόνα 18: Διάγραμμα ροής εφαρμογής.

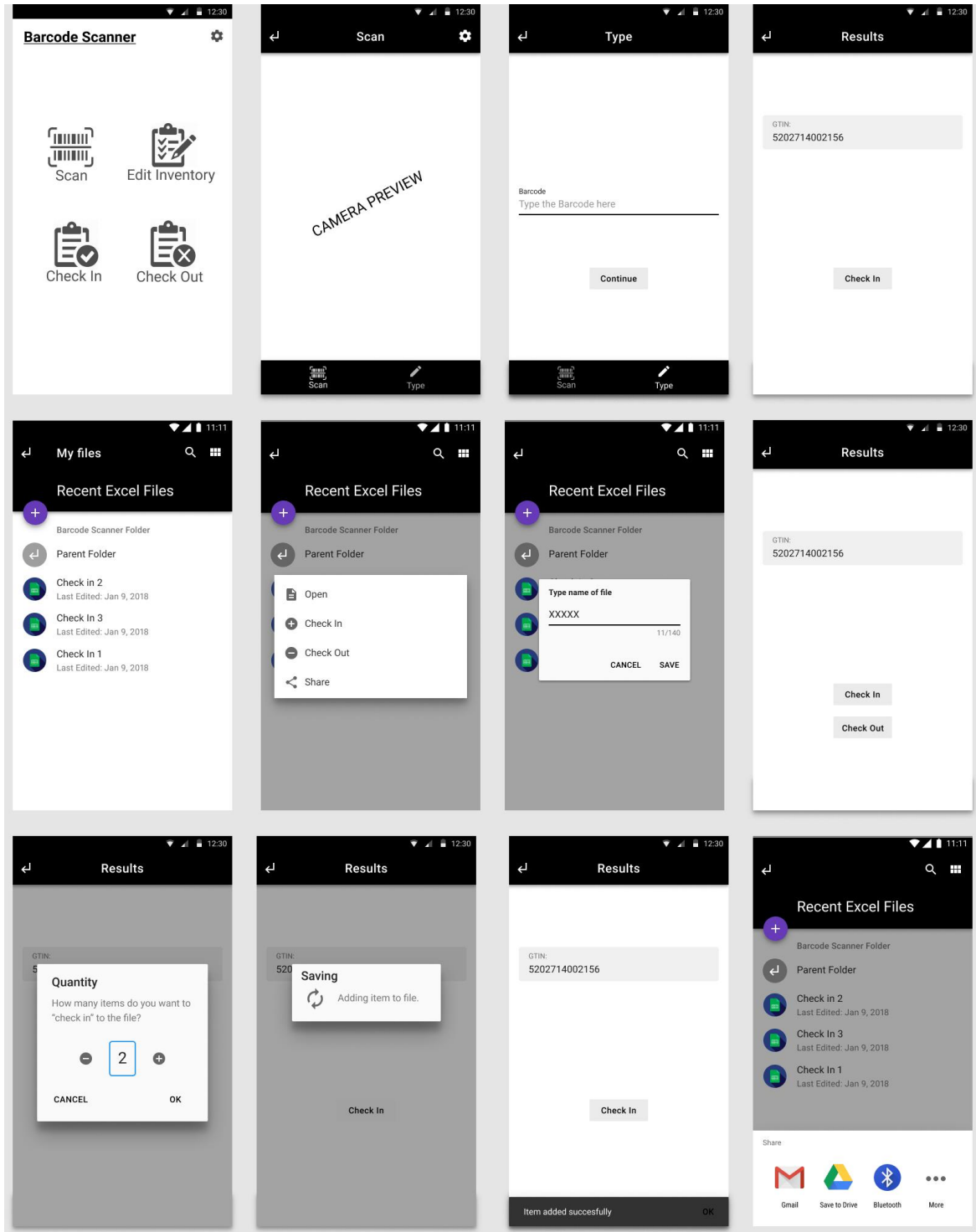
## Εφαρμογή Android για την αναγνώριση και καταγραφή προϊόντων μέσω barcode



Εικόνα 19: Διάγραμμα βάσης δεδομένων

### 6.1.3.2 Διεπαφή Χρήστη (UI)

Για την διαμόρφωση της διεπαφής του χρήστη, χρησιμοποιήθηκε το διανυσματικό πρόγραμμα επεξεργασίας γραφικών Figma. Με στόχο την απλοποίηση του γραφικού περιβάλλοντος, για την ευκολότερη χρήση της εφαρμογής από τον χρήστη, το αρχικό μενού και τα υπόλοιπα activities, διαμορφώθηκαν όπως φαίνονται στις παρακάτω εικόνες.



Εικόνα 20: Σχεδίαση layouts μέσω Figma

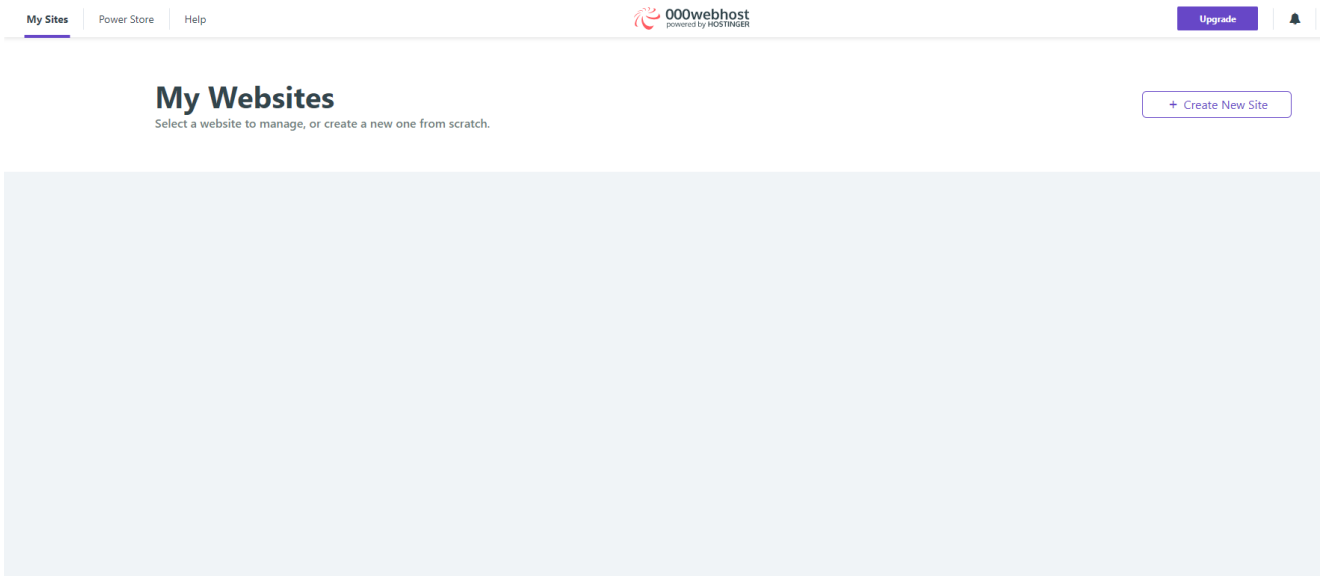


## 6.2 Υλοποίηση

Η φάση της υλοποίησης αποτελεί την τέταρτη φάση στην οποία εισαγόμαστε. Στα προηγούμενα κεφάλαια παρουσιάστηκαν η φάση της ανάλυσης και της σχεδίασης του συστήματος. Στη φάση αυτή θα βασιστούμε στα δεδομένα που προέκυψαν από τις τρεις προηγούμενες φάσεις, ώστε να προχωρήσουμε στην υλοποίηση της εφαρμογής. Ο ολοκληρωμένος κώδικας της εφαρμογής παρουσιάζεται στο τέλος της παρούσας διπλωματικής εργασίας.

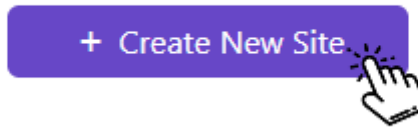
### 6.2.1 Υλοποίηση Βάσης Δεδομένων

Για την ανάπτυξη και υλοποίηση της βάσης δεδομένων, χρησιμοποιήθηκε η ιστοσελίδα [000webhost.com](http://000webhost.com) η οποία παρέχει δωρεάν φιλοξενία ιστοσελίδων. Η συγκεκριμένη ιστοσελίδα μέσα στις υπηρεσίες που παρέχει περιλαμβάνεται και η δημιουργία και τροποποίηση βάσης δεδομένων. Για να φτιάξουμε τη βάση δεδομένων μας ακολουθούμε τις παρακάτω εικόνες. Η πλατφόρμα επεξεργασίας της βάσης δεδομένων είναι το rhpMyAdmin. Εκεί, βασισμένοι στην προηγούμενη φάση όπου αναλύσαμε το σύστημα, δημιουργήσα δύο πίνακες καθώς και τα απαραίτητα στοιχεία τους. Στις παρακάτω εικόνες φαίνονται τα βήματα που ακολουθούμε για τη δημιουργία νέας βάσης δεδομένων μέσω της πλατφόρμας που αναφέρθηκε.



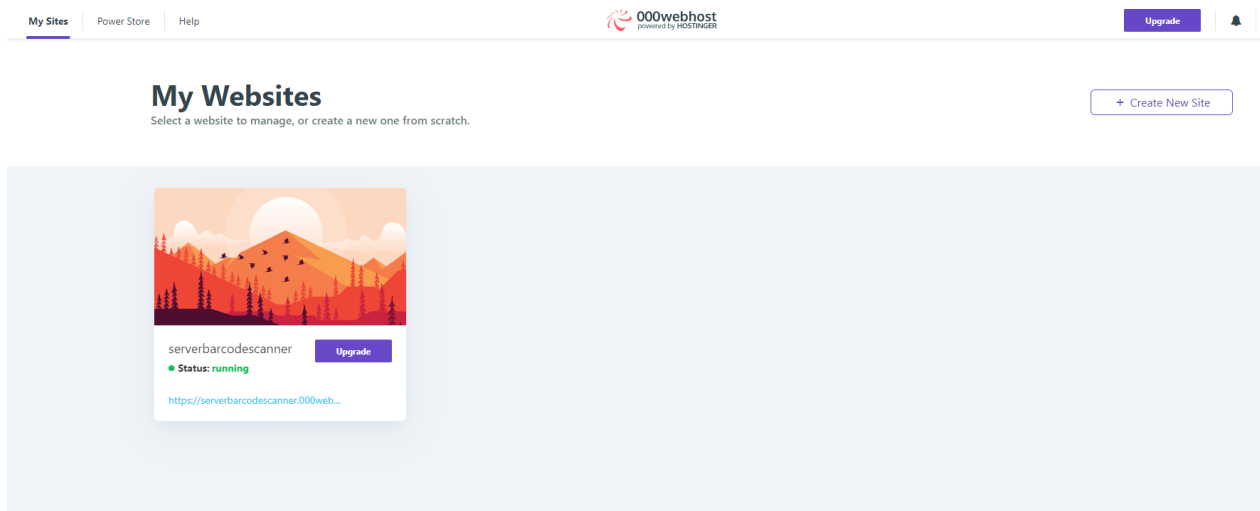
Εικόνα 21: Αρχική είσοδος στην πλατφόρμα

Η δημιουργία νέας ιστοσελίδας είναι απαραίτητη, ώστε να μέσω του URL της, να έχουμε πρόσβαση απομακρυσμένα, στη βάση δεδομένων μας. Για την δημιουργία της ιστοσελίδας μας, επιλέγουμε το κουμπί με την επιγραφή «Create New Site», όπως φαίνεται στη παρακάτω εικόνα.

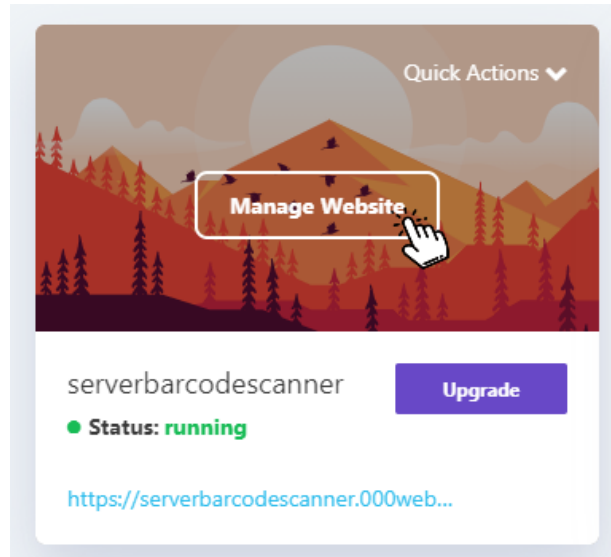


Εικόνα 22: Κουμπί δημιουργίας νέας ιστοσελίδας

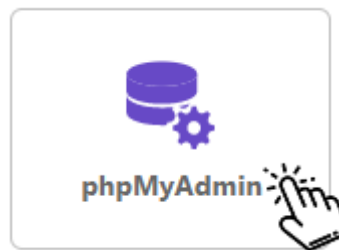
Αφού κάνουμε κλικ στο παραπάνω κουμπί, πληκτρολογούμε τις πληροφορίες της νέας ιστοσελίδας και επιλέγουμε το κουμπί «Create». Εάν ολοκληρώθηκε ορθά η δημιουργία της ιστοσελίδας, θα εμφανιστεί στο αρχικό μενού της πλατφόρμας. Εφόσον εμφανιστεί, επιλέγουμε την επιλογή «Manage Website» και έπειτα επιλέγουμε «rhpMyAdmin», όπως φαίνεται στις παρακάτω εικόνες.



Εικόνα 23: Αρχικό μενού μετά την δημιουργία της ιστοσελίδας



Εικόνα 24: Κουμπί «Manage Website»





Εικόνα 25: Επιλογή phpMyAdmin

Αφού κάνουμε αριστερό κλικ στο κουμπί «phpMyAdmin», επιλέγουμε «New Database» και πληκτρολογούμε τις απαραίτητες πληροφορίες της βάσης δεδομένων μας.



Με την ολοκλήρωση της δημιουργίας της βάσης δεδομένων, δημιουργούμε τους πίνακες, όπως αναλύθηκαν στην προηγούμενη φάση. Έπειτα από την δημιουργία των πινάκων και των στοιχείων τους, οι πίνακες θα εμφανίζονται στο portal του phpMyAdmin, όπως φαίνονται στις παρακάτω εικόνες.

Filters		Containing the word: <input type="text"/>				
Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> company	★ Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	32.0 KiB	-
<input type="checkbox"/> gtin	★ Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	32.0 KiB	-
<b>2 tables</b>	<b>Sum</b>	<b>8</b>	<b>InnoDB</b>	<b>utf8_unicode_ci</b>	<b>64.0 KiB</b>	<b>0 B</b>

Εικόνα 26: Πίνακες Βάσης Δεδομένων

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	comp_id 	int(11)			No	None		AUTO_INCREMENT
2	comp_name 	varchar(256)	latin1_swedish_ci		No	None		
3	city	varchar(50)	latin1_swedish_ci		No	None		
4	address	varchar(50)	latin1_swedish_ci		No	None		
5	pcode	varchar(50)	latin1_swedish_ci		No	None		
6	prefix	varchar(50)	latin1_swedish_ci		No	None		

Εικόνα 27: Πίνακας «Company»

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	gtin_num 	varchar(30)	latin1_swedish_ci		No	None		
2	brand_name	varchar(30)	latin1_swedish_ci		No	None		
3	prod_desc	varchar(60)	latin1_swedish_ci		No	None		
4	prod_img_url	varchar(90)	latin1_swedish_ci		No	None		
5	gpc	varchar(30)	latin1_swedish_ci		No	None		
6	net_content	varchar(30)	latin1_swedish_ci		No	None		
7	country_of_sale	varchar(30)	latin1_swedish_ci		No	None		
8	comp_id 	int(11)			No	None		

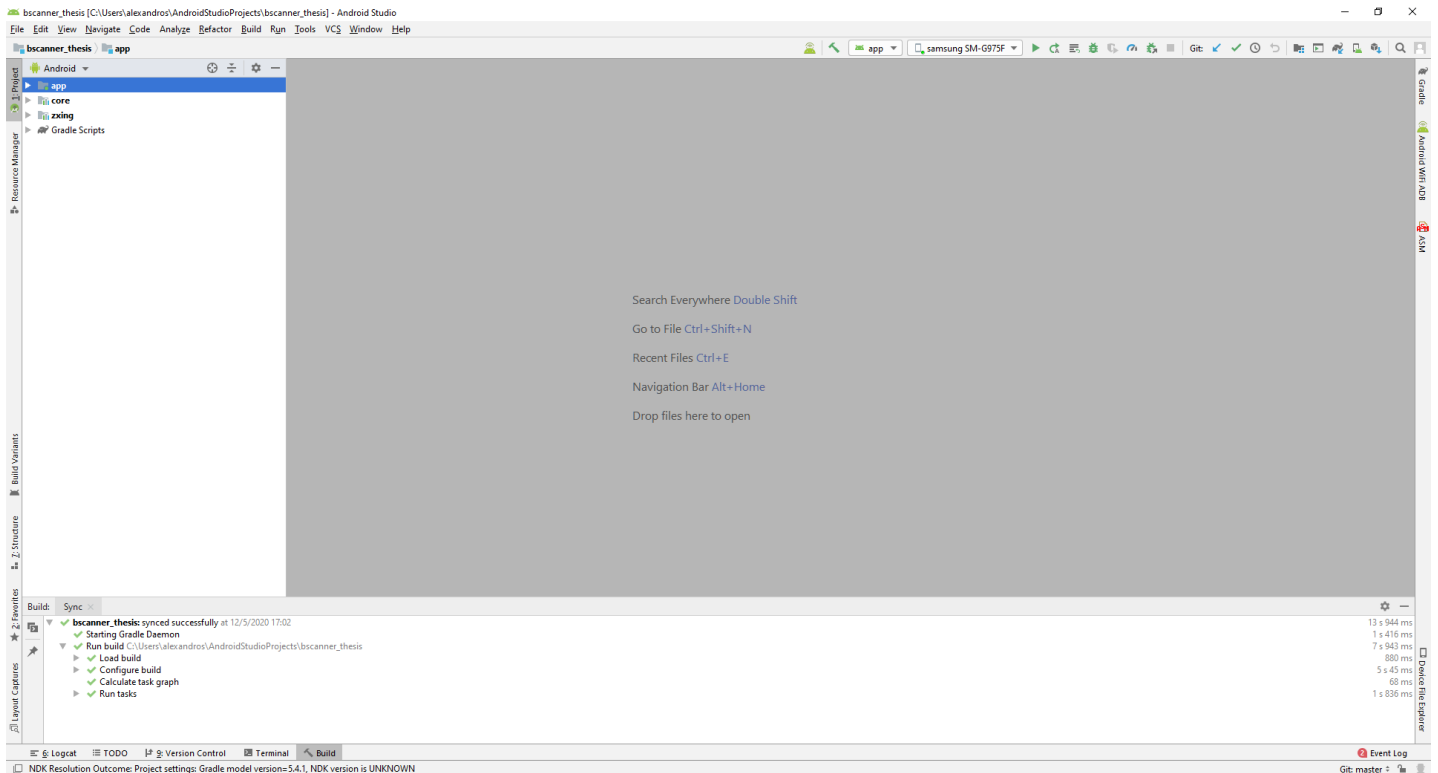
Εικόνα 28: Πίνακας «Gtin»

## 6.2.2 Υλοποίηση Εφαρμογής

Αφού έχουμε ολοκληρώσει την δημιουργία της ιστοσελίδας και της βάσης δεδομένων, θα προχωρήσουμε στην ανάπτυξη και υλοποίηση της εφαρμογής.

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκε ένας υπολογιστής με λειτουργικό σύστημα Windows 10. Σε αυτόν τον υπολογιστή εγκαταστάθηκε το περιβάλλον ανάπτυξης λογισμικού της Google, Android Studio. Για την δημιουργία μιας νέας εφαρμογής ακολουθούμε τα βήματα που εμφανίζονται στο παράθυρο του Android Studio. Το συγκεκριμένο πρόγραμμα στο αρχικό μενού μας δίνει τη δυνατότητα, πέρα από τη δημιουργία ενός νέου project, να εισάγουμε έναν ήδη υπάρχων φάκελο project, να συνδέσουμε λογαριασμό GitHub και άλλες επιλογές. Εμείς επιλέγουμε την δημιουργία νέου project, με όνομα Barcode Scanner και ελάχιστη υποστηριζόμενη έκδοση Android, την έκδοση 5.0 Lollipop.

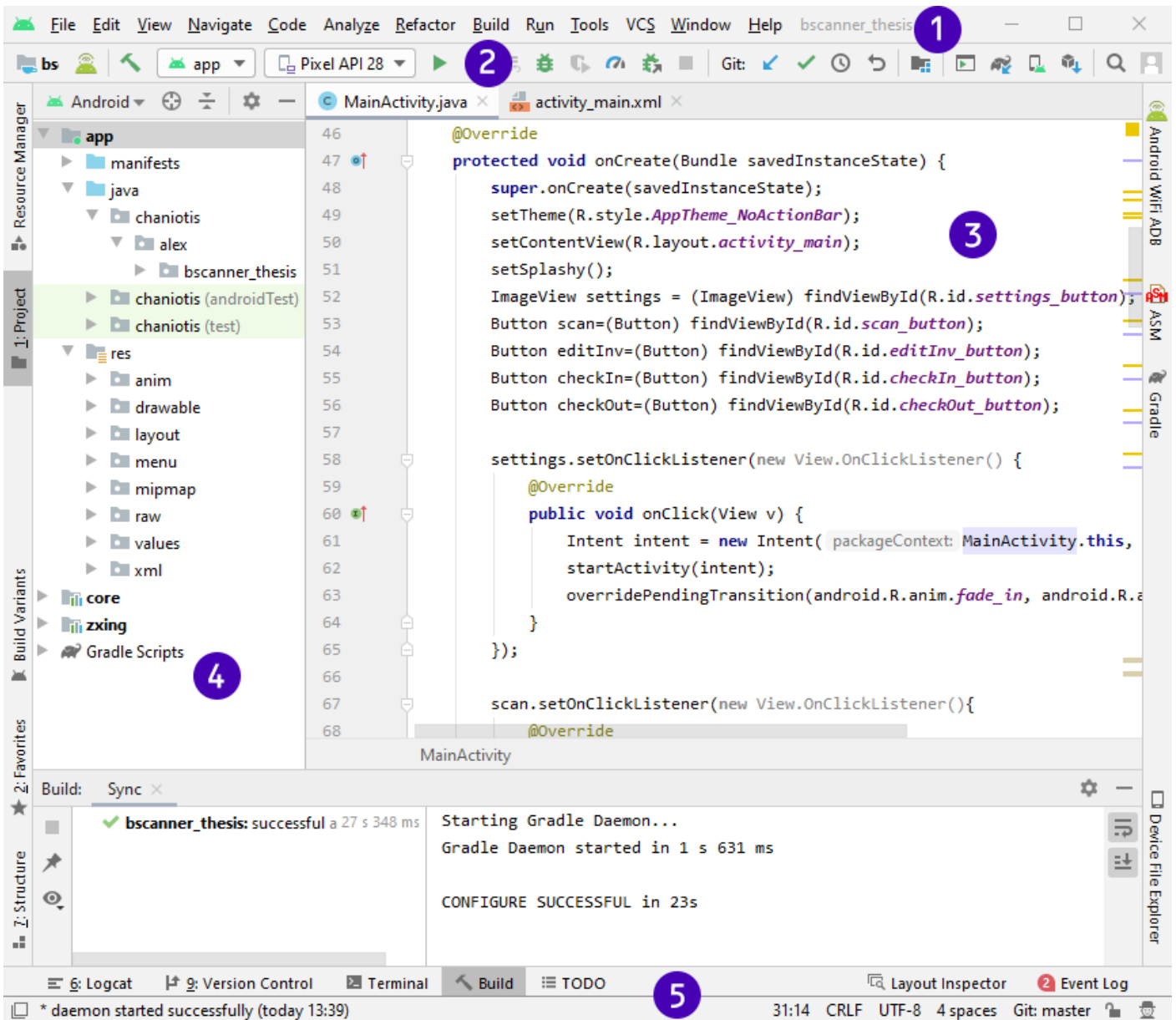
## Εφαρμογή Android για την αναγνώριση και καταγραφή προϊόντων μέσω barcode



Εικόνα 29: Όψη Android Studio

Το γραφικό περιβάλλον του Android Studio, απαρτίζεται κυρίως από 4 κομμάτια. Αριστερά εμφανίζεται η δομή του project όπου μέσω αυτής της προβολής βλέπουμε μια πεπλατυσμένη μορφή της δομής του έργου μας, που παρέχει γρήγορη πρόσβαση στα βασικά αρχεία προέλευσης των έργων Android. Κάτω εμφανίζονται τυχόν ERRORS έχουν προκύψει κατά το Build ή κατά το τρέξιμο της εφαρμογής. Στο πάνω μέρος βρίσκεται το μενού με τις βασικές επιλογές σχετικά με το Android Studio και επίσης το μενού με τις επιλογές σχετικά με το Build και Run της εφαρμογής σε ψηφιακή ή πραγματική κινητή συσκευή. Στο κέντρο της οθόνης βρίσκεται ο χώρος όπου εμφανίζεται ο τροποποιήσιμος κώδικας των αρχείων καθώς και μια εικονική μορφή των layout αρχείων κατά τη διάρκεια της επεξεργασίας τους. Παρακάτω ακολουθεί σχετική εικόνα για την ανάλυση των δομικών μερών του περιβάλλοντος του Android Studio.

## Εφαρμογή Android για την αναγνώριση και καταγραφή προϊόντων μέσω barcode



Εικόνα 30: Κύρια διεπαφή χρήση Android Studio

1. **Menu Bar** - Η γραμμή μενού περιέχει μια σειρά από επιλογές για την εκτέλεση εργασιών εντός του περιβάλλοντος του Android Studio.
2. **Toolbar** - Η γραμμή εργαλείων επιτρέπει την εκτέλεση ένα ευρύ φάσμα ενεργειών, όπως η εκτέλεση της εφαρμογής σας, η εκκίνηση του εργαλείου Debugger και άλλων χρήσιμων εργαλείων.
3. **Window Editor** - Ο χώρος επεξεργασίας εμφανίζει το περιεχόμενο του αρχείου στο οποίο εργάζεται ο προγραμματιστής. Ωστόσο, η μορφή που εμφανίζεται κάθε αρχείο σε αυτό το παράθυρο εξαρτάται εξ'ολοκλήρου από το είδος του κώδικα που περιέχει. Κατά την επεξεργασία κώδικα, για παράδειγμα, θα εμφανιστεί ο επεξεργαστής κώδικα. Κατά την επεξεργασία ενός αρχείου διάταξης διεπαφής χρήστη, θα εμφανιστεί το εργαλείο επεξεργασίας διάταξης διεπαφής χρήστη. Όταν είναι ανοιχτά πολλά αρχεία, κάθε αρχείο αντιπροσωπεύεται από μια καρτέλα που βρίσκεται στο πάνω άκρο του χώρου επεξεργασίας, όπως φαίνεται στην παραπάνω εικόνα.

4. **Project Window** - Το παράθυρο αυτό παρέχει μια ιεραρχική επισκόπηση της δομής αρχείων του έργου που επιτρέπει την πλοήγηση σε συγκεκριμένα αρχεία και φακέλους. Η γραμμή εργαλείων μπορεί να χρησιμοποιηθεί για την προβολή του έργου με διάφορους τρόπους. Η προεπιλεγμένη ρύθμιση είναι η προβολή Android που είναι και η λειτουργία που χρησιμοποιήθηκε για την ανάπτυξη της συγκεκριμένης εφαρμογής.
5. **Status Bar** - Η γραμμή κατάστασης εμφανίζει ενημερωτικά μηνύματα σχετικά με το έργο και τις δραστηριότητες του Android Studio μαζί με το κουμπί μενού εργαλείων που βρίσκεται στην αριστερή γωνία. Τοποθετώντας το δείκτη πάνω από τα στοιχεία στη γραμμή κατάστασης θα βρείτε μια περιγραφή αυτού του πεδίου. Πολλά πεδία είναι διαδραστικά, επιτρέποντας στον χρήστη να κάνει κλικ για να εκτελέσει εργασίες ή να λάβει πιο λεπτομερείς πληροφορίες κατάστασης.

### 6.2.2.1 Κλάση Product

```
package chaniotis.alex.bsscanner_thesis.app;

public class Product {
    private static String Product_GTIN = "Record Not Found!";
    private static String Product_Brand_Name= "Record Not Found!";
    private static String Product_Description= "Record Not Found!";
    private static String Product_Image_URL= "Record Not Found!";
    private static String Product_NetWeight= "Record Not Found!";

    public static String getProduct_GTIN() {
        return Product_GTIN;
    }

    public static String getProduct_Brand_Name() {
        return Product_Brand_Name;
    }

    public static String getProduct_Description() {
        return Product_Description;
    }

    public static String getProduct_Image_URL() {
        return Product_Image_URL;
    }

    public static String getProduct_NetWeight() {
        return Product_NetWeight;
    }

    public static void setProduct_GTIN(String product_GTIN) {
        Product_GTIN = product_GTIN;
    }

    public static void setProduct_Brand_Name(String product_Brand_Name) {
        Product_Brand_Name = product_Brand_Name;
    }

    public static void setProduct_Description(String product_Description) {
        Product_Description = product_Description;
    }

    public static void setProduct_Image_URL(String product_Image_URL) {
        Product_Image_URL = product_Image_URL;
    }

    public static void setProduct_NetWeight(String product_NetWeight) {
        Product_NetWeight = product_NetWeight;
    }
}
```



### 6.2.2.2 Κλάση Company

```
package chaniotis.alex.bsscanner_thesis.app;

public class Company {
    private static String Company_name = "Record Not Found!";
    private static String Company_city= "Record Not Found!";
    private static String Company_address= "Record Not Found!";
    private static String Company_pcode= "Record Not Found!";
    //setters
    public static void setCompany_name(String company_name) {
        Company_name = company_name;
    }

    public static void setCompany_city(String company_city) {
        Company_city = company_city;
    }

    public static void setCompany_address(String company_address) {
        Company_address = company_address;
    }

    public static void setCompany_pcode(String company_pcode) {
        Company_pcode = company_pcode;
    }

    //getters
    public static String getCompany_name() {
        return Company_name;
    }

    public static String getCompany_city() {
        return Company_city;
    }

    public static String getCompany_address() {
        return Company_address;
    }

    public static String getCompany_pcode() {
        return Company_pcode;
    }
}
```

### 6.2.2.3 Manifest

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

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

    <application
        android:name="chaniotis.alex.bsscanner_thesis.app.MyApplication"
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name="chaniotis.alex.bsscanner_thesis.ui.results.ResultActivity"
            android:screenOrientation="portrait"
            android:theme="@style/AppTheme.NoActionBar"></activity>

        <provider
            android:name="android.support.v4.content.FileProvider"
            android:authorities="${applicationId}.provider"
            android:exported="false"
            android:grantUriPermissions="true">
            <meta-data
                android:name="android.support.FILE_PROVIDER_PATHS"
                android:resource="@xml/provider_paths" />
        </provider>

        <activity
            android:name="chaniotis.alex.bsscanner_thesis.ui.files.BrowseFilesActivity"
            android:screenOrientation="portrait"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name="chaniotis.alex.bsscanner_thesis.ui.scan.ScanActivity"
            android:screenOrientation="portrait"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name="chaniotis.alex.bsscanner_thesis.ui.home.MainActivity"
            android:screenOrientation="portrait"
            android:theme="@style/AppTheme.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name="chaniotis.alex.bsscanner_thesis.ui.settings.SettingsActivity"
            android:screenOrientation="portrait"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name="chaniotis.alex.bsscanner_thesis.ui.settings.AboutActivity"
            android:screenOrientation="portrait"
            android:theme="@style/AppTheme.NoActionBar" />
    </application>
</manifest>
```

## 6.2.2.4 MainActivity.Java

```

package chaniotis.alex.bsscanner_thesis.ui.home;
import ...

public class MainActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setTheme(R.style.AppTheme_NoActionBar);
        setContentView(R.layout.activity_main);
        setSplashy();
        ImageView settings = (ImageView) findViewById(R.id.settings_button);
        Button scan=(Button) findViewById(R.id.scan_button);
        Button editInv=(Button) findViewById(R.id.editInv_button);
        Button checkIn=(Button) findViewById(R.id.checkIn_button);
        Button checkOut=(Button) findViewById(R.id.checkOut_button);

        settings.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, SettingsActivity.class);
                startActivity(intent);
                overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
            }
        });

        scan.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(MainActivity.this, ScanActivity.class);
                intent.putExtra(Constants.ACTIVITY_MODE, Constants.SCAN_MODE_NONE);
                startActivity(intent);
                overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
            }
        });

        editInv.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(MainActivity.this, BrowseFilesActivity.class);
                intent.putExtra(Constants.ACTIVITY_MODE, Constants.MODE_EXPLORER);
                startActivity(intent);
                overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
            }
        });

        checkIn.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(MainActivity.this, BrowseFilesActivity.class);
                intent.putExtra(Constants.ACTIVITY_MODE, Constants.MODE_CHECK_IN);
                startActivity(intent);
                overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
            }
        });

        checkOut.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(MainActivity.this, BrowseFilesActivity.class);
                intent.putExtra(Constants.ACTIVITY_MODE, Constants.MODE_CHECK_OUT);
                startActivity(intent);
                overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
            }
        });
    }

    private void setSplashy(){
        new Splashy(this)
            .setLogo(R.drawable.ic_launcher_foreground)
            .setTitle("Barcode Scanner")
            .setTitleColor(R.color.black)
            .setLogoScaleType(FIT_XY)
            .setBackgroundColor(R.color.gradient_color2)
            .setAnimation(Splashy.Animation.GROW_LOGO_FROM_CENTER, 800)
            .setFullScreen(false)
            .show();
    }
}

```

### 6.2.2.5 ScanActivity.Java

```

ScanActivity

package chaniotis.alex.bsscanner_thesis.ui.scan;
import ...

public class ScanActivity {
    private String mFilePath;
    boolean frag=true;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_scan);
        mViewBinding = getViewDataBinding();
        mViewModel.setNavigator(this);
        requestPermissions();
        getIntentData();
        setUp();
    }

    @Override
    public void onAttachFragment(Fragment fragment) {
        if (fragment instanceof BarCodeScanFragment) {
            ((BarCodeScanFragment) fragment).setListener(this);
            frag=true;
        }
        if (fragment instanceof TypeFragment) {
            ((TypeFragment) fragment).setListener(this);
            frag=false;
        }
    }

    private void setUp() {
        scanScreenBottomNavigationView
            .setOnNavigationItemSelectedListener(menuItem -> {
                scanScreenTitle.setText(getString(R.string.scan_tab_title));
                switch (menuItem.getItemId()) {
                    case R.id.scan_screen_scan_tab: {
                        if (frag!=true) {
                            scanScreenTitle.setText(getString(R.string.scan_tab_title));
                            loadFragment(BarCodeScanFragment.newInstance());
                        }
                    }
                    return true;
                    case R.id.scan_screen_type_tab: {
                        if (frag!=false) {
                            scanScreenTitle.setText(getString(R.string.type_tab_title));
                            loadFragment(TypeFragment.newInstance());
                        }
                    }
                    return true;
                }
            });
    }

    @Override
    public void onContinue(String gtin, String format) {
        Intent intent = new Intent(ScanActivity.this, ResultActivity.class);
        intent.putExtra(Constants.getKeyGtin(), gtin);
        intent.putExtra(Constants.getKeyFormat(), format);
        intent.putExtra(Constants.FILE_PATH, mFilePath);
        intent.putExtra(Constants.ACTIVITY_MODE, mActivityMode);
        startActivity(intent);
        overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
    }

    //this method uses Dexter runtime permission library to check for required permissions
    private void requestPermissions() {
        Dexter.withActivity(this).withPermission(Manifest.permission.CAMERA)
            .withListener(new PermissionListener() {
                @Override
                public void onPermissionGranted(PermissionGrantedResponse response) { //granted
                    loadFragment(BarCodeScanFragment.newInstance());
                }

                @Override
                public void onPermissionDenied(PermissionDeniedResponse response) { //denied
                    if (response.isPermanentlyDenied()) {
                        showSettingsDialog();
                        return;
                    }
                    showPermissionRationale();
                }
            })
            .onSameThread().check();
    }
}

```

### 6.2.2.6 BarCodeScanFragment.Java

```
package chaniotis.alex.bsscanner_thesis.ui.scan.fragment.barcode;
import ...

public class BarCodeScanFragment {

    public static BarCodeScanFragment newInstance() {
        Bundle args = new Bundle();
        BarCodeScanFragment fragment = new BarCodeScanFragment();
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public void onCreateView(@NonNull View view, Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        mViewBinding = getViewDataBinding();
        setUp();
    }

    private void setUp() {
        FullScannerFragment barcodeReader = new FullScannerFragment();
        barcodeReader.setListener(this);
        FragmentTransaction transaction = getFragmentManager().beginTransaction();
        transaction.add(R.id.barcode_container, barcodeReader);
        transaction.commit();
    }

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        this.mViewModel.setNavigator(this);
    }

    @Override
    public void onScanned(Result rawResult) {
        SharedPreferences prefs = PreferenceManager.getDefaultSharedPreferences(getContext());
        if(prefs.getBoolean("key_notifications", true)){
            final MediaPlayer beep = MediaPlayer.create(getContext(), R.raw.beep);
            beep.start();
        }
        if(prefs.getBoolean("key_vibrate", true)){
            Vibrator v = (Vibrator) getActivity().getSystemService(Context.VIBRATOR_SERVICE);
            if(v.hasVibrator())v.vibrate(250);
        }
        if (mListener != null) {
            mListener.onContinue(rawResult.getText(),rawResult.getBarcodeFormat().toString());
        }
    }
}
```

### 6.2.2.7 TypeFragment.Java

```

package chaniotis.alex.bsscanner_thesis.ui.scan.fragment.type;
import ...

public class TypeFragment {

    public static TypeFragment newInstance() {
        Bundle args = new Bundle();
        TypeFragment fragment = new TypeFragment();
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public void onCreateView(@NonNull View view, Bundle savedInstanceState) {
        super.onCreateView(view, savedInstanceState);
        setUp();
    }

    private void hideKeyboard() {
        InputMethodManager imm =
        (InputMethodManager)this.getContext().getSystemService(Context.INPUT_METHOD_SERVICE);
        imm.hideSoftInputFromWindow(barcodeEditText.getWindowToken(), 0);
    }

    private void setUp() {
        barcodeEditText.setOnEditorActionListener(new TextView.OnEditorActionListener() {
            @Override
            public boolean onEditorAction(TextView v, int actionId, KeyEvent event) {
                boolean handled = false;
                if (actionId == EditorInfo.IME_ACTION_NEXT) { //next button on bottom right keyboard
                    onContinue();
                    handled = true;
                }
                return handled;
            }
        });

        barcodeEditText.setOnFocusChangeListener((view, isFocused) -> {
            if (isFocused) {
                barcodeEditText.setHint(R.string.barcode_here);
            }
            if (!isFocused) {
                hideKeyboard();
            }
        });

        Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                barcodeEditText.requestFocus();
            }
        }, 300);
        continueButton.setOnClickListener(view -> {
            onContinue();
        });
    }

    private void onContinue(){
        boolean isValid = true;
        String gtin = barcodeEditText.getText().toString().trim();
        if(gtin.length() == 0 || gtin.length() == 8 || gtin.length() == 12 || gtin.length() ==
        13 || gtin.length() == 14) {
            if (gtin.length() == 0) {
                isValid = false;
                barcodeEditText.setError(getString(R.string.empty_gtin));
                barcodeEditText.requestFocus();
            } else if (gtin.length() < MIN_GTIN_LENGTH) {
                isValid = false;
                barcodeEditText.setError(getString(R.string.gtin_short, MIN_GTIN_LENGTH));
                barcodeEditText.requestFocus();
            } else if (gtin.length() > MAX_GTIN_LENGTH) {
                isValid = false;
                barcodeEditText.setError(getString(R.string.gtin_large, MAX_GTIN_LENGTH));
                barcodeEditText.requestFocus();
            }
            if (isValid && mListener != null) {
                mListener.onContinue(gtin, null);
                hideKeyboard();
            }
        }
        else{
            barcodeEditText.setError(getString(R.string.gtin_length));
            barcodeEditText.requestFocus();
        }
    }
}

```

## 6.2.2.8 SettingsActivity.java

```

package chaniotis.alex.bsscanner_thesis.ui.settings;
import ...

public class SettingsActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // load settings fragment
        getFragmentManager().beginTransaction().replace(android.R.id.content, new
MainPreferenceFragment()).commit();
    }

    public static class MainPreferenceFragment extends PreferenceFragment {
        @Override
        public void onCreate(final Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            addPreferencesFromResource(R.xml.pref_main);
            // storage EditText change listener
            bindPreferenceSummaryToValue(findPreference(getString(R.string.key_storage_name)));
            //ip editttext listener
            bindPreferenceSummaryToValue(findPreference(getString(R.string.key_server_ip)));
            //port editttext listener
            bindPreferenceSummaryToValue(findPreference(getString(R.string.key_server_port)));
            //about on click listener
            Preference AboutPref = findPreference(getString(R.string.key_about));
            AboutPref.setOnPreferenceClickListener(new Preference.OnPreferenceClickListener() {
                public boolean onPreferenceClick(Preference preference) {
                    Intent intent = new Intent(getActivity(), AboutActivity.class);
                    startActivity(intent);
                    return true;
                }
            });
        }
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        if (item.getItemId() == android.R.id.home) {
            onBackPressed();
        }
        return super.onOptionsItemSelected(item);
    }

    private static void bindPreferenceSummaryToValue(Preference preference) {
        preference.setOnPreferenceChangeListener(sBindPreferenceSummaryToValueListener);
        sBindPreferenceSummaryToValueListener.onPreferenceChange(preference,
            PreferenceManager
                .getDefaultSharedPreferences(preference.getContext())
                .getString(preference.getKey(), ""));
    }

    //A preference value change listener that updates the preference's summary to reflect its new
value.
    private static Preference.OnPreferenceChangeListener sBindPreferenceSummaryToValueListener = new
Preference.OnPreferenceChangeListener() {
        @Override
        public boolean onPreferenceChange(Preference preference, Object newValue) {
            String stringValue = newValue.toString();

            if (preference instanceof ListPreference) {
                // For list preferences, look up the correct display value in the preference's
'entries' list.
                ListPreference listPreference = (ListPreference) preference;
                int index = listPreference.findIndexOfValue(stringValue);
                // Set the summary to reflect the new value.
                preference.setSummary(
                    index >= 0
                        ? listPreference.getEntries()[index]
                        : null);
            } else if (preference instanceof EditTextPreference) {
                if (preference.getKey().equals("key_storage_name")) {
                    // update the changed storage name to summary
                    preference.setSummary(stringValue);
                } else if (preference.getKey().equals("key_server_ip")) {
                    // update the changed ip address to summary
                    preference.setSummary(stringValue);
                } else if (preference.getKey().equals("key_server_port")) {
                    // update the changed port address to summary
                    preference.setSummary(stringValue);
                }
            } else if (preference instanceof SwitchPreference)
                preference.setSummary(stringValue);
            return true;
        }
    };
}

```

### 6.2.2.9 AboutActivity

```
package chaniotis.alex.bsscanner_thesis.ui.settings;

import android.os.Bundle;
import android.app.Activity;
import android.view.View;
import android.widget.ImageView;

import chaniotis.alex.bsscanner_thesis.R;

public class AboutActivity extends Activity {

    ImageView softBackBtn;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_about);
        softBackBtn=(ImageView) findViewById(R.id.screen_back);
        softBackBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
                overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
            }
        });
    }

    @Override
    public void onBackPressed() {
        finish();
        overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);
    }
}
```



### 6.2.2.7 CompanyFragment.Java

```

package chaniotis.alex.bsscanner_thesis.ui.results;
import ...

public class CompanyFragment extends Fragment {

    TextView companyTextView,addressTextView,pcodeTextView,cityTextView;
    TextView gtinTextView,brandNameTextView,productDescTextView,netContentTextView;
    LinearLayout parentLayout;
    private ActivityResultBinding mViewBinding;
    private AlertDialogBinding binding;
    private ProgressDialog pDialog;
    private String
        compName,compAddress,compCity,compPcode,gtinNum,brandName,productDesc,productImgUrl,netWeight,gtin_co
de;

    private String gtin_code;
    ImageView image;
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        return inflater.inflate(R.layout.company_details_frag, container, false);
    }

    @Override
    public void onViewCreated(View view, Bundle savedInstanceState){
        gtin_code=((ResultActivity)this.getActivity()).getVar_Gtin();
        deleteProductDetails();
        companyTextView = (TextView) getView().findViewById(R.id.companyDetails_textView);
        addressTextView = (TextView) getView().findViewById(R.id.addressDetails_textView);
        pcodeTextView = (TextView) getView().findViewById(R.id.pcodeDetails_textView);
        cityTextView = (TextView) getView().findViewById(R.id.cityDetails_textView);
        gtinTextView = (TextView) getView().findViewById(R.id.gtin_textView);
        brandNameTextView = (TextView) getView().findViewById(R.id.brandname_textView);
        productDescTextView = (TextView) getView().findViewById(R.id.proddesc_textView);
        image =(ImageView) getView().findViewById(R.id.imageView);
        netContentTextView = (TextView) getView().findViewById(R.id.netweight_textView);
    }

    /**
     * Fetches values from the server
     */
    private class FetchDetailsTask extends AsyncTask<String, String, String> {
        ...
        ...
        ...
    }

    private void deleteCompanyDetails(){
        Company.setCompany_name(EMPTY);
        Company.setCompany_address(EMPTY);
        Company.setCompany_city(EMPTY);
        Company.setCompany_pcode(EMPTY);
    }

    private void deleteProductDetails(){
        Product.setProduct_GTIN(EMPTY);
        Product.setProduct_Brand_Name(EMPTY);
        Product.setProduct_Description(EMPTY);
        Product.setProduct_Image_URL(EMPTY);
        Product.setProduct_NetWeight(EMPTY);
    }

    public void setClassesValues(String gtin,String brand, String desc, String url, String weight, String
        comp, String address, String city, String pcode){
        Product.setProduct_GTIN(gtin);
        Product.setProduct_Brand_Name(brand);
        Product.setProduct_Description(desc);
        Product.setProduct_Image_URL(url);
        Product.setProduct_NetWeight(weight);
        Company.setCompany_name(comp);
        Company.setCompany_address(address);
        Company.setCompany_city(city);
        Company.setCompany_pcode(pcode);
    }

    public void setClassesValuesNotFound(String gtin){
        Product.setProduct_GTIN(gtin);
        Product.setProduct_Brand_Name(RECORD_NOT_FOUND);
        Product.setProduct_Description(RECORD_NOT_FOUND);
        Product.setProduct_Image_URL(RECORD_NOT_FOUND);
        Product.setProduct_NetWeight(RECORD_NOT_FOUND);
        Company.setCompany_name(RECORD_NOT_FOUND);
        Company.setCompany_address(RECORD_NOT_FOUND);
        Company.setCompany_city(RECORD_NOT_FOUND);
        Company.setCompany_pcode(RECORD_NOT_FOUND);
    }
}

```

### 6.2.2.8 Κλάση FetchDetailsTask

```

/**
 * Fetches values from the server
 */
private class FetchDetailsTask extends AsyncTask<String, String, String> {

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        progressDialog = new ProgressDialog(getActivity());
        progressDialog.setMessage("Connecting to Database.\nPlease wait...");
        progressDialog.setIndeterminate(false);
        progressDialog.setCancelable(false);
        progressDialog.show();
    }

    @Override
    protected String doInBackground(String... params) {
        HttpJsonParser httpJsonParser = new HttpJsonParser();
        Map<String, String> httpParams = new HashMap<>();
        httpParams.put(Constants.getKeyGtin(), gtin_code);
        Log.d("scanning with", ""+gtin_code);
        JSONObject jsonObject = httpJsonParser.makeHttpRequest(getBaseUrl(), "GET", httpParams);
        try {
            int status = jsonObject.getInt(getKeyStatus());
            JSONObject result;
            String message;
            if (status == 1) {
                result = jsonObject.getJSONObject(getKeyData());

                gtinNum=result.getString(getKeyGtinNum());
                brandName=result.getString(getKeyBrandName());
                productDesc=result.getString(getKeyProdDesc());
                productImgUrl=result.getString(getKeyKeyImgUrl());
                netWeight= result.getString(getKeyNetContent());
                compName = result.getString(getKeyCompName());
                compAddress=result.getString(getKeyAddress());
                compCity=result.getString(getKeyCity());
                compPcode=result.getString(getKeyPcode());

                setClassesValues(gtinNum, brandName, productDesc, productImgUrl, netWeight, compName, compAddress, compCity,
                    compPcode);
            }
            if(Company.getCompany_name().isEmpty()) {
                setClassesValuesNotFound(gtin_code);
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
        return null;
    }

    protected void onPostExecute(String result) {
        progressDialog.dismiss();
        getActivity().runOnUiThread(new Runnable() {
            public void run() {

                gtinTextView.setText(Product.getProduct_GTIN());
                brandNameTextView.setText(Product.getProduct_Brand_Name());
                productDescTextView.setText(Product.getProduct_Description());
                if(Product.getProduct_Image_URL().equals(RECORD_NOT_FOUND))
                    image.setImageResource(R.drawable.ic_unavailable_image);
                else Picasso.get().load(Product.getProduct_Image_URL()).into(image);
                netContentTextView.setText(Product.getProduct_NetWeight());

                companyTextView.setText(Company.getCompany_name());
                addressTextView.setText(Company.getCompany_address());
                cityTextView.setText(Company.getCompany_city());
                pcodeTextView.setText(Company.getCompany_pcode());
            }
        });
    }
}

```

### 6.2.3 Σύνδεση της Εφαρμογής με τη Βάση Δεδομένων

Για τη σύνδεση της εφαρμογής με τη βάση δεδομένων χρησιμοποιήσαμε scripts γραμμένα σε γλώσσα php. Η γλώσσα προγραμματισμού php σχεδιάστηκε με στόχο την ανάπτυξη και τη διαχείριση ιστοσελίδων αλλά στην εφαρμογή μας χρησιμοποιείται για την σύνδεση με τη βάση δεδομένων αλλά και για να παρέχει πρόσβαση στα δεδομένα αυτής, ενσωματώνοντας εντολές SQL στον κώδικα php. Για την επιτυχή σύνδεση της ιστοσελίδας με τη βάσης δεδομένων, είναι απαραίτητη η δημιουργία βοηθητικού αρχείου που θα τοποθετηθεί στον αποθηκευτικό χώρο της ιστοσελίδας μας. Στην ουσία το αρχείο αυτό ορίζει στην ιστοσελίδα τα στοιχεία της βάσης δεδομένων στην οποία θα γίνει η αναζήτηση. Το βοηθητικό αυτό αρχείο είναι το παρακάτω.

#### db\_connect.php

```

db_connect.php x
<?php
define('DB_USER', "id8619460_admin@_αδκ"); // db user
define('DB_PASSWORD', "Xz8d_270E"); // db password (mention your db password here)
define('DB_DATABASE', "id8619460_bscanner_thesis"); // database name
define('DB_SERVER', "localhost"); // db server

$con = mysqli_connect(DB_SERVER,DB_USER,DB_PASSWORD,DB_DATABASE);

// Check connection
if(mysqli_connect_errno())
{
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
}
?>

```

Εικόνα 31: Βοηθητικό αρχείο σύνδεσης ιστοσελίδας με συγκεκριμένη βάση δεδομένων

Αφού δημιουργηθεί και τοποθετηθεί το αρχείο της σύνδεσης στο root φάκελο της ιστοσελίδας μας, απομένει να προστεθεί το αρχείο το οποίο θα περιέχει τον κώδικα SQL, που στην ουσία θα είναι υπεύθυνο για την μεταφορά των δεδομένων μεταξύ της εφαρμογής και της βάσης δεδομένων. Για την επίτευξη της μεταφοράς δεδομένων το αρχείο αυτό θα πρέπει να τρέχει τον κώδικα του παραπάνω αρχείου (θα γίνει include στο νέο αρχείο), και τον απαραίτητο κώδικα αναζήτησης στη βάση δεδομένων. Ο κώδικας φαίνεται στην παρακάτω εικόνα.

## get\_comp.php

```

get_comp.php
<?php
include 'db/db_connect.php';
$resArray = array();
$response = array();
$response["status"] = 0;
$response["message"] = "No products found";
if(isset($_GET['gtin_num'])){
    $var_gtin = $_GET['gtin_num'];
    $query="SELECT gtin.gtin_num, gtin.brand_name, gtin.prod_desc, gtin.prod_img_url,
            gtin.net_content, company.comp_name, company.city, company.address, company.
            pcode FROM company INNER JOIN gtin ON company.comp_id= gtin.comp_id WHERE gtin
            .gtin_num =? LIMIT 1";
    if($stmt = $con->prepare($query)){
        $stmt->bind_param("s",$var_gtin);
        $stmt->execute();
        $stmt->bind_result($gtinNum,$brandName,$prodDesc,$prodImgUrl,$netContent,$compName
            ,$compCity,$compAddress,$compCode);
        if($stmt->fetch()){
            $resArray["gtin_num"] = $gtinNum;
            $resArray["brand_name"] = $brandName;
            $resArray["prod_desc"] = $prodDesc;
            $resArray["prod_img_url"] = $prodImgUrl;
            $resArray["net_content"] = $netContent;
            $resArray["comp_name"] = $compName;
            $resArray["city"] = $compCity;
            $resArray["address"] = $compAddress;
            $resArray["pcode"] = $compCode;
            $response["status"] = 1;
            $response["data"] = $resArray;
        }else{
            $response["status"] = 0;
            $response["message"] = "Movie not found";
        }
        $stmt->close();
    }else{
        $response["status"] = 0;
        $response["message"] = mysqli_error($con);
    }
}else{
    $response["status"] = 0;
    $response["message"] = "missing parameter gtin_num";
}
echo json_encode($response);
?>

```

Εικόνα 32: Αρχείο μεταφοράς δεδομένων μεταξύ εφαρμογής και βάσης δεδομένων

Αφού υλοποιήσουμε με επιτυχία τα δυο παραπάνω αρχεία, μας απομένει να προσθέσουμε τον απαραίτητο κώδικα για την σύνδεσης της εφαρμογής με την ιστοσελίδα και τη βάση δεδομένων.

Στο αρχείο με τις σταθερές μεταβλητές της εφαρμογής, θα προσθέσουμε την παρακάτω γραμμή κώδικα:

```

private static final String BASE_URL=
"https://serverbarcodescanner.000webhostapp.com/get_comp.php";

```

Αφού ορίσουμε τον σύνδεσμο της ιστοσελίδας μας στις σταθερές μεταβλητές, όταν έρθει η στιγμή να κάνουμε αναζήτηση στη βάση δεδομένων βάσει κωδικού προϊόντος θα πρέπει να προσθέσουμε το παρακάτω κομμάτι κώδικα.

```
HttpJsonParser httpJsonParser = new HttpJsonParser();
Map<String, String> httpParams = new HashMap<>();
httpParams.put(Constants.getKeyGtin(), gtin_code); // gtin_code=κωδικός προϊόντος
JSONObject jsonObject=
httpJsonParser.makeHttpRequest(getBaseUrl(), "GET", httpParams);
```

Οι επιστρεφόμενες τιμές θα έχουν JSON μορφή όπως φαίνεται στο παράδειγμα παρακάτω:

```
{
  "status": 1,
  "data": [
    {
      "gtin_num": "5202714001259",
      "brand_name": "BRAND NAME A",
      "prod_desc": "Product A",
      "net_content": "450g",
      "prod_img_url": "www.exampleurl.com/prodA.jpg",
      "comp_name": "Company A",
      "city": "City A",
      "address": "Address A",
      "pcode": "12242",
      "status": "1"
    }
  ]
}
```

Για την αποθήκευση των τιμών αυτών θα χρησιμοποιήσουμε τον παρακάτω κώδικα για την εκχώρηση τους στην μεταβλητή `result`.

```
JSONObject result;
result = jsonObject.getJSONObject(getKeyData());
```

## 6.2.4 Έλεγχος

Η φάση αυτή ακολουθεί μετά από τη φάση της υλοποίησης και σε περίπτωση που εντοπιστούν παραλείψεις ή δυσλειτουργίες, επιστρέφουμε στη φάση της υλοποίησης ή ακόμα και στη φάση της ανάλυσης, για επανασχεδιασμό του συστήματος και έλεγχο εκ νέου. Σκοπός μας είναι πάντοτε η αποδοτικότερη τελική λειτουργία του συστήματος.

Στη συγκεκριμένη εφαρμογή ο έλεγχος έγινε αρχικά με απλά σενάρια χρησιμοποιώντας την εφαρμογή σε μια μόνο συσκευή η οποία ήταν συνδεδεμένη στο Android Studio, ώστε να παρακολουθούμε την εξέλιξη των δεδομένων και των τυχόν σφαλμάτων.

Για τον έλεγχο της σύνδεσης, της ορθής λειτουργίας μεταξύ της εφαρμογής και της βάσης δεδομένων και για τυχόν σφάλματα ή αστοχίες, έκανα εισαγωγή στη βάση δεδομένων, dummy δεδομένα και χρησιμοποίησα τον κωδικό 5201234567894 ως κωδικό προϊόντος λιανικής.

Εν συνεχεία η εφαρμογή ελέγχθηκε σε πραγματικές συνθήκες και διαφορετικές συσκευές(διαφορετικών εταιρειών) όπου και λειτούργησε ορθά χωρίς να παρουσιαστούν σφάλματα.

## 6.2.5 Διανομή

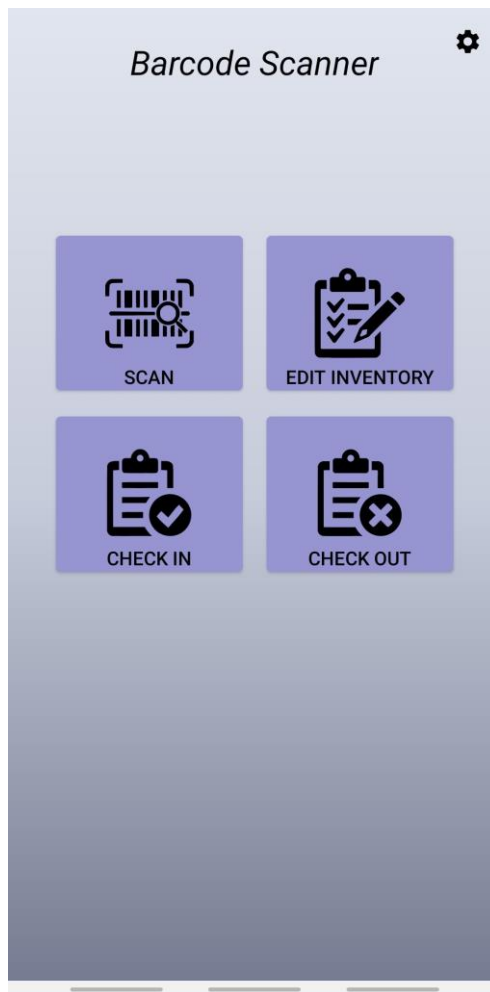
Η παρούσα εφαρμογή καθώς και το ολοκληρωμένο σύστημα, δε θα διατίθεται κάπου προς λήψη ή πώληση όπως για παράδειγμα Google Play, δημόσιο repository στο GitHub ή υπηρεσία εταιρείας λογισμικού. Η εφαρμογή και η βάση δεδομένων σχεδιάστηκε και υλοποιήθηκε μόνο στα πλαίσια της διπλωματικής εργασίας.

## 6.3 Λειτουργία της Εφαρμογής

Στην ενότητα αυτή θα δώσουμε μια αναλυτική περιγραφή της εφαρμογής Barcode Scanner καθώς και του τρόπου που μπορεί να την αξιοποιήσει ένας χρήστης. Η εφαρμογή αποτελείται από μια αλληλουχία δραστηριοτήτων τις οποίες παρουσιάζουμε παρακάτω.

### 6.3.1 Η Δραστηριότητα MainActivity

Η δραστηριότητα MainActivity είναι η αρχική οθόνη της εφαρμογής. Εμφανίζει τις βασικές επιλογές και λειτουργίες της εφαρμογής. Στην διεπαφή του χρήστη εμφανίζονται 4 κεντρικά κουμπιά με τις υποδείξεις «Scan» «Edit Inventory» «Check In» και «Check Out». Εάν ο χρήστης επιλέξει το κουμπί με την επιγραφή «Scan», τότε μεταβαίνει στην δραστηριότητά ScanActivity. Εάν επιλέξει ένα από τα υπόλοιπα τρία κουμπιά, θα μεταβεί στο «BrowserFilesActivity» με τη μεταβλητή «ActivityMode» να έχει τιμές «MODE\_EXPLORER», «MODE\_CHECK\_IN» και «MODE\_CHECK\_OUT» αντίστοιχα, αναλόγως την επιλογή του χρήστη.



Εικόνα 33: Η δραστηριότητα *MainActivity*.

### 6.3.2 Η Δραστηριότητα *ScanActivity*

Η δραστηριότητα «*ScanActivity*» έχει σκοπό την ανάγνωση του γραμμικού κώδικα. Η δραστηριότητά αυτή επιτρέπει στον χρήστη δυο διαφορετικούς τρόπους ανάγνωσης του γραμμικού κώδικα. Ο πρώτος τρόπος είναι με τη χρήση της κάμερας ενώ ο δεύτερος τρόπος είναι με τη πληκτρολόγηση του κωδικού που αναγράφεται κάτω από τις κάθετες γραμμές του συμβόλου. Και στις δύο περιπτώσεις, μετά την επιτυχή ανάγνωση του κωδικού, ο χρήστης παραπέμπεται στην δραστηριότητα «*ResultActivity*» για την προβολή των αποτελεσμάτων. Οι δύο λειτουργίες ανάγνωσης του κωδικού γίνονται φόρτωση ως *Fragments* στο *main\_layout*, ανάλογα με την επιλογή που θα κάνει ο χρήστης στην μπάρα πλοήγησης που βρίσκεται στο κάτω μέρος της οθόνης. Ακόμη μια εργασία που εκτελείται στην δραστηριότητα *ScanActivity*, είναι η αίτηση δικαιωμάτων από το σύστημα για χρήση της κάμερας της συσκευής.

### 6.3.2.1 Fragment BarCodeScanFragment

Το fragment «BarCodeScanFragment» έχει σκοπό την σάρωση του γραμμωτού κώδικα που φέρεται στην συσκευασία του προϊόντος. Αφού αποκτηθούν τα απαραίτητα δικαιώματα από το σύστημα, η εφαρμογή με τη χρήση της βιβλιοθήκης σάρωσης γραμμωτών κωδικών, εντοπίζει το σύμβολο που βρίσκεται εντός του πλαισίου που έχει οριστεί στο layout, και εντέλει αποκωδικοποιεί τον γραμμωτό κώδικα σε αριθμό GTIN. Με την επιτυχή ανάγνωση και αποκωδικοποίηση του συμβόλου ο χρήστης μεταφέρεται στη δραστηριότητα ResultsActivity, όπου του εμφανίζονται πληροφορίες σχετικά με τον κωδικό που μόλις αποκωδικοποιήθηκε.

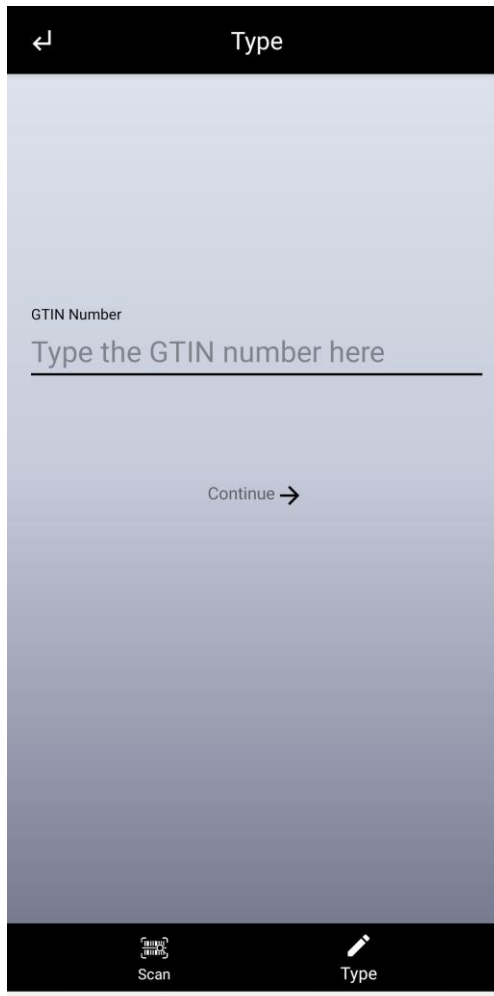


Εικόνα 34: Fragment BarCodeScanFragment



### 6.3.2.2 Fragment TypeFragment

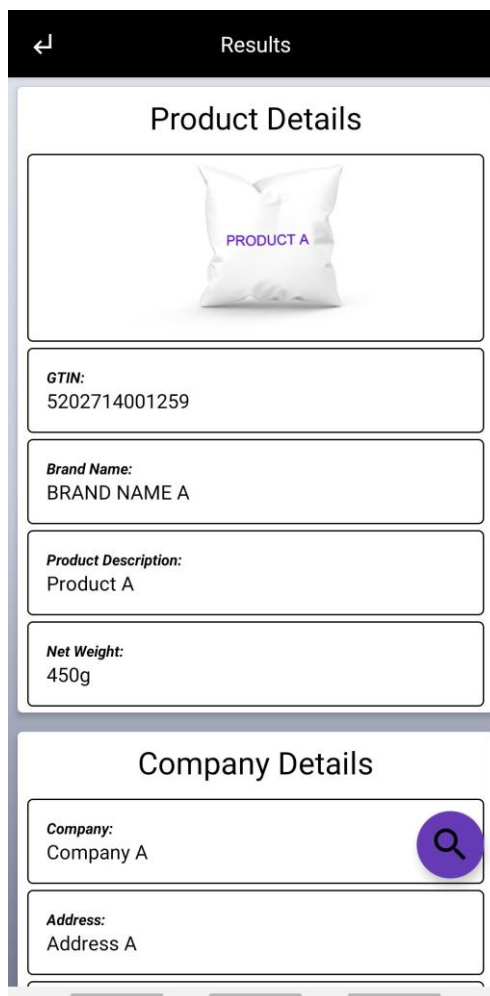
Το fragment «TypeFragment» παρέχεται στον χρήστη ως δεύτερη επιλογή, σε περίπτωση που δεν είναι εφικτή η σάρωση του γραμμωτού κώδικα. Έτσι ο χρήστης πληκτρολογεί τον κωδικό που αναγράφεται κάτω από τις κάθετες γραμμές του συμβόλου ή τον κωδικό που επιθυμεί και στη συνέχεια με το πάτημα του πλήκτρου «Continue», μεταφέρεται στη δραστηριότητα ResultsActivity, όπου εκεί εμφανίζονται πληροφορίες σχετικά με τον συγκεκριμένο κωδικό.



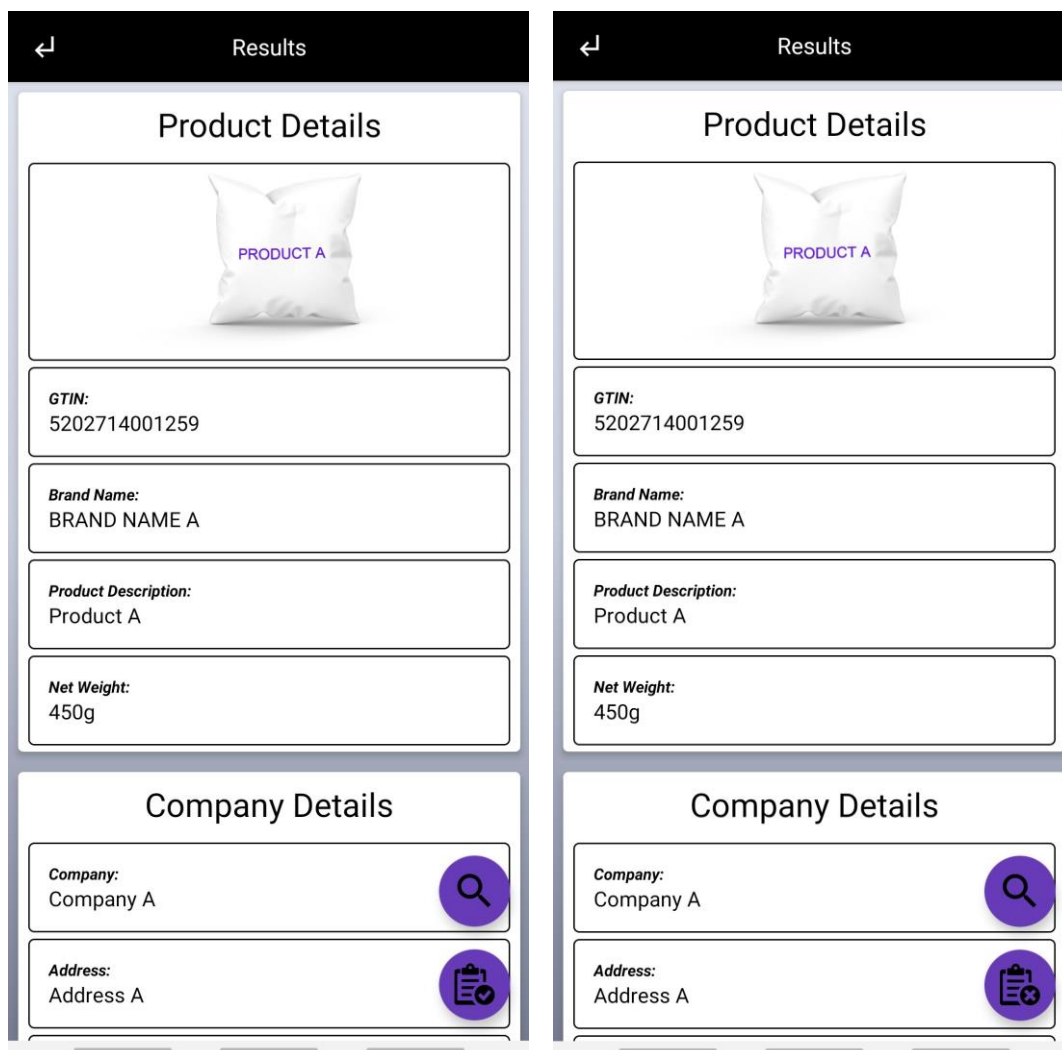
Εικόνα 35: Fragment TypeFragment

### 6.3.3 Η Δραστηριότητα ResultsActivity

Η δραστηριότητα ResultsActivity έχει σκοπό την εμφάνιση των αποτελεσμάτων, έπειτα από την επιτυχή σάρωση του γραμμωτού κώδικα ή την ορθή πληκτρολόγηση του κωδικού του προϊόντος. Στην περίπτωση που ο χρήστης έχει πραγματοποιήσει ανάγνωση κωδικού μέσω του πλήκτρου «Scan» στο αρχικό μενού, τότε στο ResultsActivity θα εφαρμοστεί το «MODE\_DEFAULT», δηλαδή τα κουμπιά για «Check In» και «Check Out» του κωδικού από ή σε αρχείο, θα είναι κρυφά. Αντίστοιχα, εάν πραγματοποιηθεί ανάγνωση του κωδικού κατά τη παραλαβή ή αποστολή προϊόντων (πλήκτρα αρχικής «Check In», «Check Out»), τότε η μεταβλητή «ActivityMode» θα έχει τιμές «MODE\_CHECK\_IN» ή «MODE\_CHECK\_OUT», που σημαίνει ότι τα κουμπιά για «Check In» ή «Check Out» θα ενεργοποιηθούν.

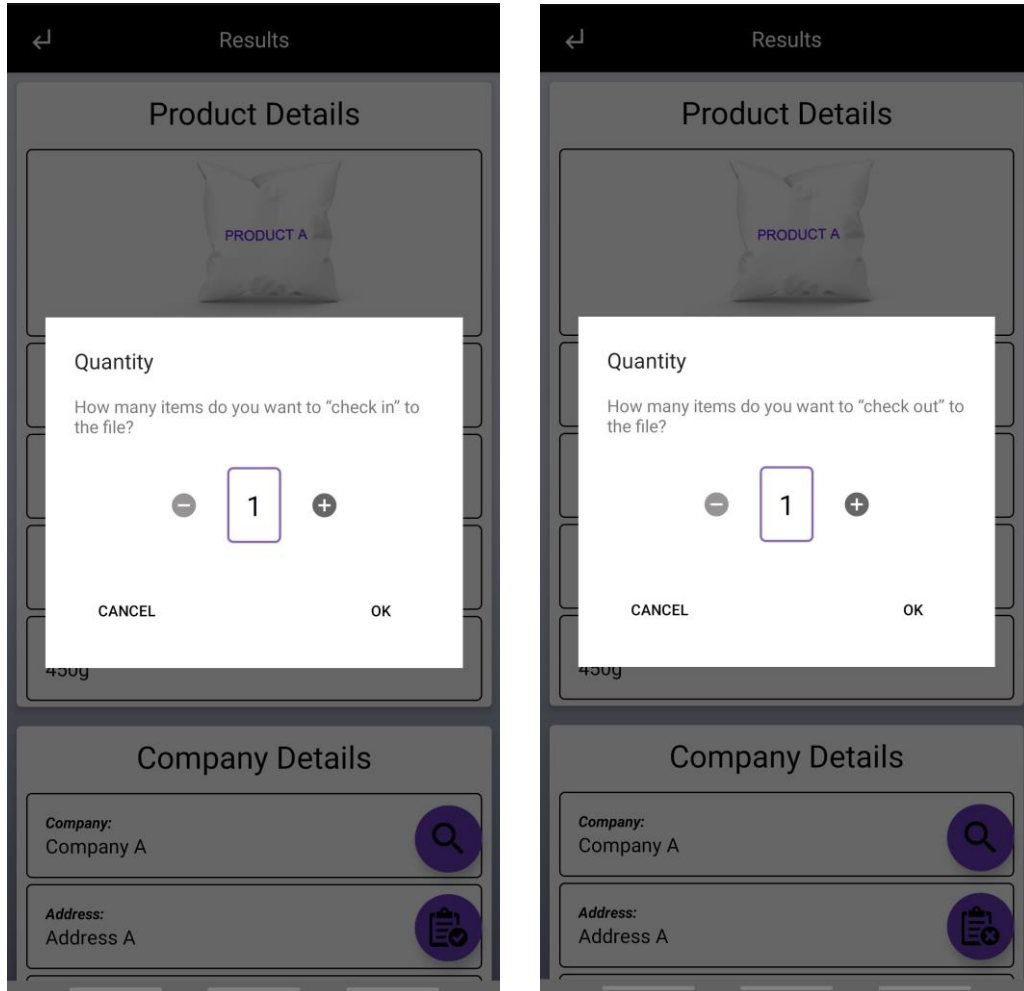


Εικόνα 36: Η δραστηριότητα ResultActivity με τιμή «MODE\_DEFAULT»



Εικόνα 37: Η δραστηριότητα ResultActivity με τιμή «MODE\_CHECK\_IN» και «MODE\_CHECK\_OUT»

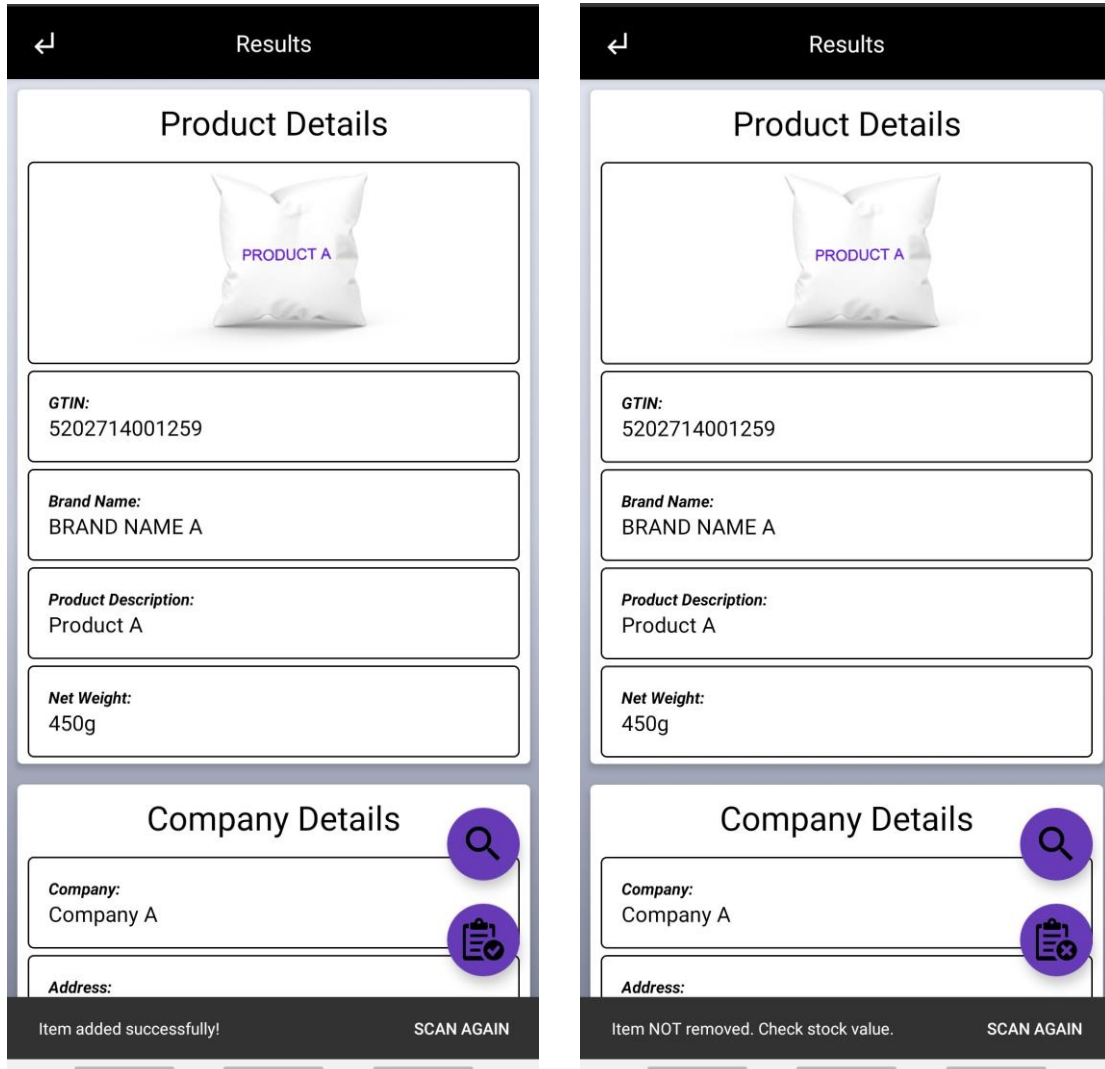
Όταν ο χρήστης επιλέξει το πλήκτρο εισαγωγής ή αφαίρεσης κωδικού προϊόντος από το επιλεγμένο αρχείο, τότε εμφανίζεται ένα Dialog. Αυτό το Dialog επιτρέπει στον χρήστη να επιλέξει το πλήθος της ποσότητας που επιθυμεί, προς εισαγωγή ή αφαίρεση από το επιλεγμένο αρχείο.



Εικόνα 38: Dialog επιλογής ποσότητας προς εισαγωγή ή/και αφαίρεση από το αρχείο

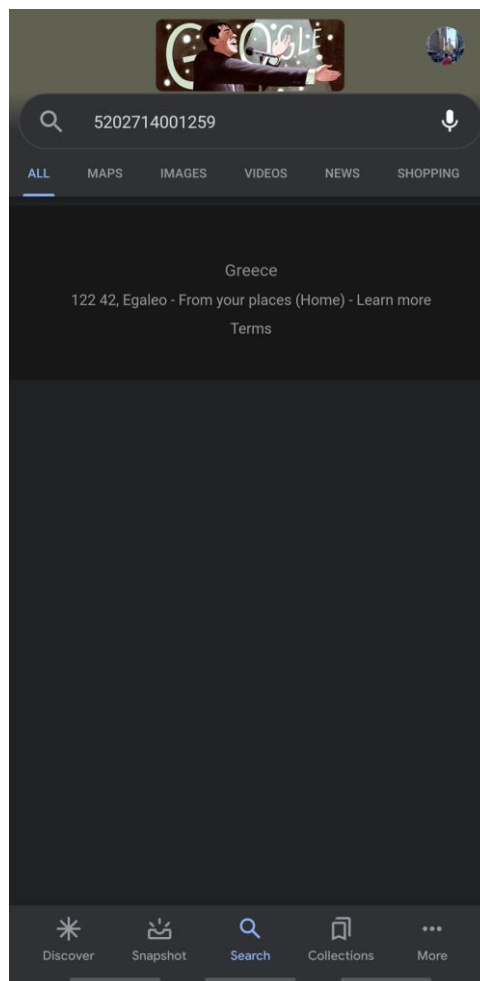
## Εφαρμογή Android για την αναγνώριση και καταγραφή προϊόντων μέσω barcode

Με την επιτυχημένη ή αποτυχημένη εισαγωγή ή αφαίρεση προϊόντος από το επιλεγμένο αρχείο, στο κάτω μέρος της οθόνης εμφανίζεται ένα Snackbar που ειδοποιεί τον χρήστη σχετικά με το αποτέλεσμα της ενέργειας.



Εικόνα 39: Snackbar επιτυχημένης εισαγωγής και αποτυχημένης εξαγωγής προϊόντος

Ακόμη μια λειτουργία που παρέχεται στον χρήστη μέσω της συγκεκριμένης Activity, είναι η αναζήτηση του αναγνωσμένου κωδικού στην μηχανή αναζήτησης Google. Ο χρήστης, πατώντας το FloatingActionButton που φέρει μορφή μεγεθυντικού φακού, μεταφέρεται στην εφαρμογή της Google, όπου και του εμφανίζονται αποτελέσματα σχετικά με τον συγκεκριμένο κωδικό. Σημειώνεται ότι η λειτουργία της αναζήτησης μέσω Google, θα πρέπει πρώτα να είναι ενεργοποιημένη από τις ρυθμίσεις της εφαρμογής.



Εικόνα 40: Αναζήτηση μέσω Google

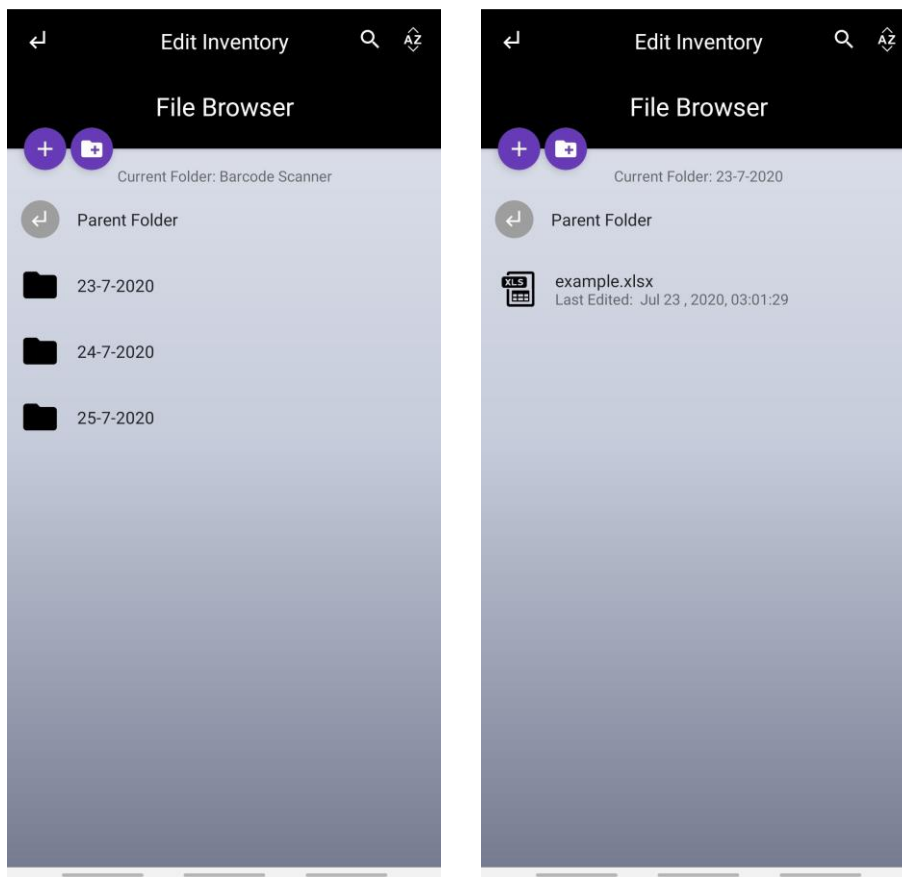
### 6.3.4 Η Δραστηριότητα `BrowseFilesActivity`

Η δραστηριότητα `BrowseFilesActivity` είναι υπεύθυνη για την διαχείριση των αρχείων excel που ο χρήστης είναι σε θέση να δημιουργήσει, τροποποιήσει, μοιραστεί και διαγράψει από το αποθηκευτικό χώρο της συσκευής. Η συγκεκριμένη δραστηριότητα εμφανίζεται στον χρήστη πατώντας τα πλήκτρα του αρχικού μενού «Edit Inventory», «Check In» και «Check Out».

Σε κάθε μια λειτουργία, η δραστηριότητα `BrowseFilesActivity` έχει ελάχιστα διαφορετικό Layout. Το Layout της κάθε λειτουργίας εξαρτάται όπως και στην προηγούμενη δραστηριότητα που αναλύθηκε, στην τιμή της μεταβλητής «*ActivityMode*». Εάν ο χρήστης επιλέξει το πλήκτρο «Edit Inventory», τότε θα μεταβεί στην δραστηριότητα `BrowseFilesActivity`, με την μεταβλητή «*ActivityMode*» να έχει τιμή «`MODE_EXPLORER`». Αντίστοιχα, εάν ο χρήστης επιλέξει το πλήκτρο «Check In» ή «Check Out» τότε η μεταβλητή θα έχει την τιμή «`MODE_CHECK_IN`» ή «`MODE_CHECK_OUT`».

Στην περίπτωση που η μεταβλητή «*ActivityMode*» έχει την τιμή «`MODE_EXPLORER`» ή «`MODE_CHECK_IN`» τότε το Layout θα παρέχει στον χρήστη τη δυνατότητα δημιουργίας νέου αρχείου και νέου φακέλου αφού πρώτα περιηγηθεί στην τοποθεσία που επιθυμεί. Αντιθέτως όταν η μεταβλητή «*ActivityMode*» έχει την τιμή «`MODE_CHECK_OUT`», τα πλήκτρα δημιουργίας νέου φακέλου ή/και νέου αρχείου, δεν εμφανίζονται στον χρήστη, αφού δεν θα ήταν εφικτή η αφαίρεση κωδικού προϊόντος από αρχείο που μόλις έχει δημιουργηθεί.

## Εφαρμογή Android για την αναγνώριση και καταγραφή προϊόντων μέσω barcode



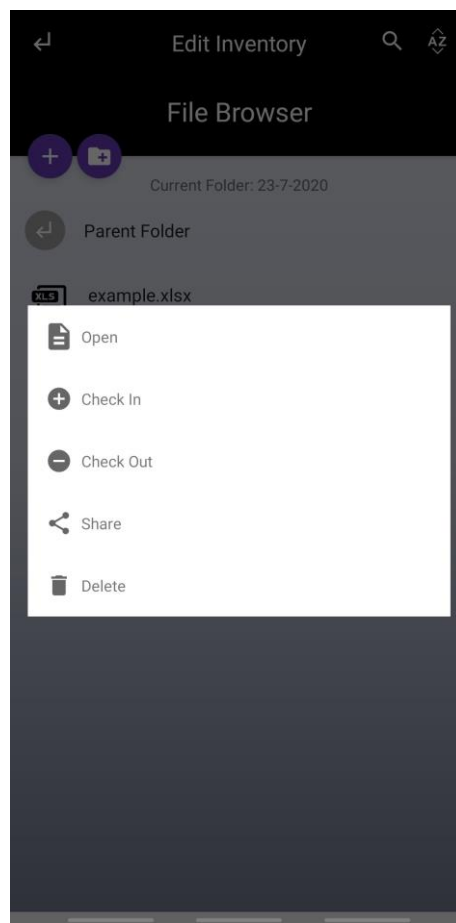
Εικόνα 41: Περιηγητής αρχείων με «MODE\_EXPLORER» και «MODE\_CHECK\_IN»



Εικόνα 42: Περιηγητής αρχείων με τιμή «MODE\_CHECK\_OUT»



Ακόμη, η συγκεκριμένη δραστηριότητα στην περίπτωση που η μεταβλητή «ActivityMode» έχει την τιμή «MODE\_EXPLORER», εάν ο χρήστης επιλέξει κάποιο αρχείο, τότε θα εμφανιστεί ένα Dialog το οποίο περιέχει τις επιλογές «Open», «Check In», «Check Out», «Share» και «Delete». Εάν ο χρήστης επιλέξει το πλήκτρο «Open», θα του εμφανιστούν οι διαθέσιμες εφαρμογές του συστήματος για άνοιγμα και τροποποίηση του συγκεκριμένου αρχείου. Επιπλέον εάν επιλέξει το πλήκτρο «Check In» ή το πλήκτρο «Check Out», θα μεταβεί στην δραστηριότητα «Scan» όπως ακριβώς θα συνέβαινε εάν επέλεγε το πλήκτρο «Check In» ή «Check Out» από το αρχικό μενού. Επιλέγοντας το κουμπί «Share», εμφανίζονται στον χρήστη οι διαθέσιμες εφαρμογές και πλατφόρμες του συστήματος για την κοινοποίηση του αρχείου. Τέλος, το πλήκτρο «Delete» δίνει τη δυνατότητα στον χρήστη για μόνιμη διαγραφή του αρχείου από τον χώρο αποθήκευσης.

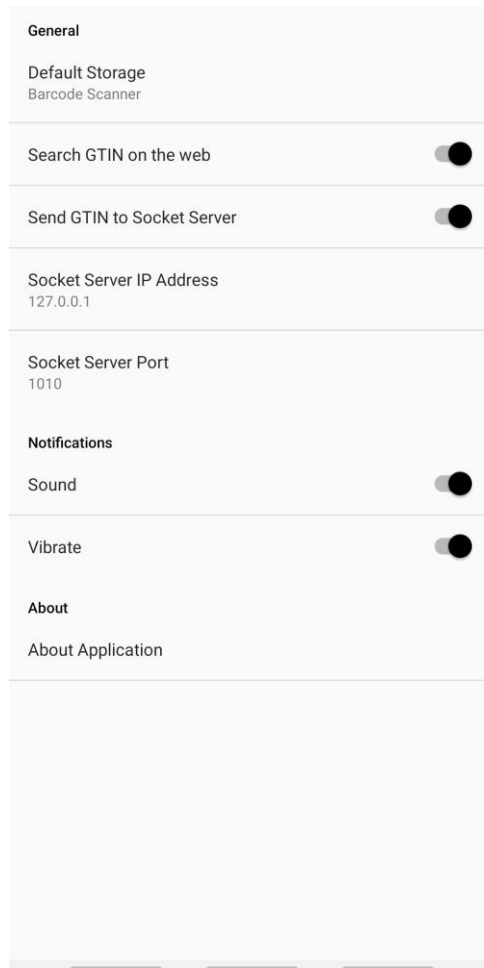


Εικόνα 43: Διαθέσιμες επιλογές κατά την επιλογή αρχείου

### 6.3.5 Η Δραστηριότητα SettingsActivity

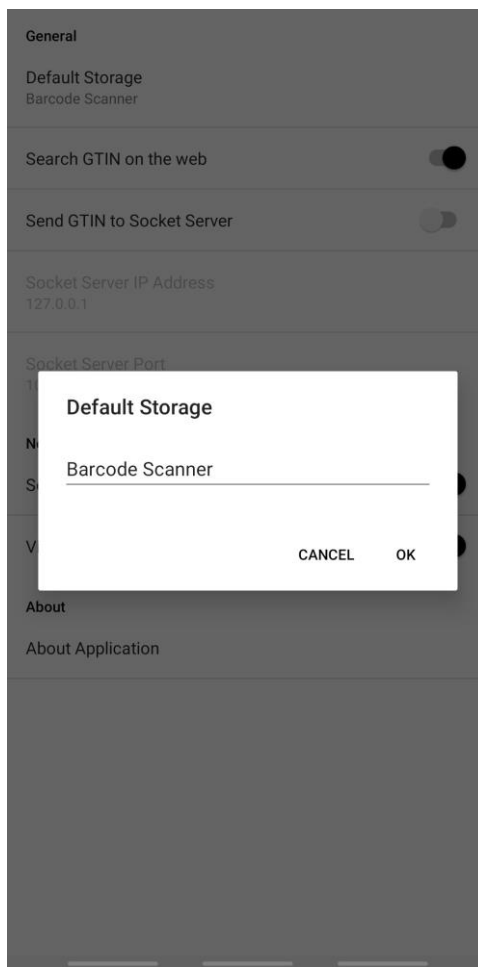
Η δραστηριότητα SettingsActivity είναι υπεύθυνη για τον ορισμό και την τροποποίηση των βασικών ρυθμίσεων της εφαρμογής. Οι ρυθμίσεις αυτές χωρίζονται σε τρεις ομάδες.

Στην πρώτη ομάδα ρυθμίσεων περιλαμβάνεται ο ορισμός του ονόματος του φακέλου στον οποίο θα αποθηκεύονται τα αρχεία που τροποποιούνται, η επιλογή για την εμφάνιση του πλήκτρου αναζήτησης στη μηχανή αναζήτησης Google κατά την εμφάνιση των αποτελεσμάτων και η επιλογή για αποστολή κωδικού προϊόντος σε socket server, αφού πρώτα οριστούν η διεύθυνση IP και η θύρα. Στην δεύτερη ομάδα ρυθμίσεων περιλαμβάνονται οι επιλογές για ενεργοποίηση αναπαραγωγής ήχου και αναπαραγωγής δόνησης κατά την επιτυχή σάρωση γραμμωτού κώδικα. Στην τελευταία ομάδα ρυθμίσεων βρίσκεται ένα πλήκτρο που μεταβαίνει τον χρήστη στην δραστηριότητα AboutActivity, την οποίας το περιεχόμενο αναδύεται παρακάτω, στο επόμενο κεφάλαιο.



Εικόνα 44: Η δραστηριότητα SettingsActivity

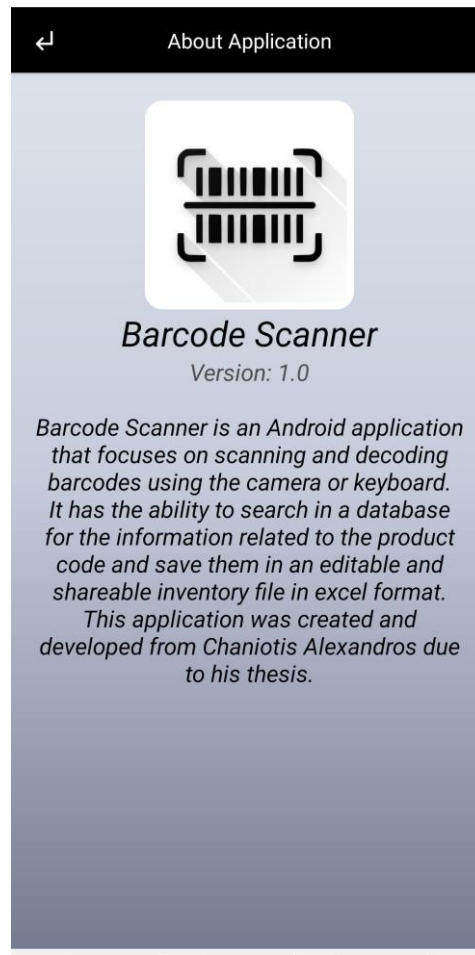
## Εφαρμογή Android για την αναγνώριση και καταγραφή προϊόντων μέσω barcode



Εικόνα 45: Ορισμός φακέλου αποθήκευσης αρχείων

### 6.3.6 Η Δραστηριότητα AboutActivity

Η δραστηριότητα AboutActivity έχει ως σκοπό την εμφάνιση ενός στατικού Layout, με μοναδικό περιεχόμενο το όνομα και εικονίδιο της εφαρμογής και ένα σύντομο κείμενο. Στο κείμενο αυτό, αναφέρονται εν συντομία οι λειτουργίες της εφαρμογής, το όνομα του developer, καθώς και ο αριθμός της έκδοσης της εφαρμογής.



Εικόνα 46: Η δραστηριότητα AboutActivity

## Βιβλιογραφία

- (2018). Ανάκτηση από DesignLab: <https://trydesignlab.com/figma-101-course/introduction-to-figma/>
- (2018). Ανάκτηση από Software Engineer Insider: <https://www.softwareengineerinsider.com/programming-languages/php.html>
- (2018). Ανάκτηση από FreeJavaGuide: <https://www.freejavaguide.com/history.html>
- (2019). Ανάκτηση από Techotopia: [https://www.techotopia.com/index.php/A\\_Tour\\_of\\_the\\_Android\\_Studio\\_User\\_Interface](https://www.techotopia.com/index.php/A_Tour_of_the_Android_Studio_User_Interface)
- (2019). Ανάκτηση από Oreilly: <https://www.oreilly.com/library/view/php-in-a/0596100671/ch01s01.html>
- (2019). Ανάκτηση από PHP: <https://www.php.net/manual/en/history.php.php>
- (2020). Ανάκτηση από Computer Hope: <https://www.computerhope.com/issues/ch000569.htm>
- Android version history.* (2020). Ανάκτηση από Wikipedia: [https://en.wikipedia.org/wiki/Android\\_version\\_history](https://en.wikipedia.org/wiki/Android_version_history)
- Deitel, P., Harvey, D., & Abbey, D. (2014). *Android Προγραμματισμός, 2η Έκδοση.*
- HAROLD, E. R. (2016). *ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΔΙΚΤΥΑΚΩΝ ΕΦΑΡΜΟΓΩΝ ΜΕ JAVA.*
- History of Java.* (2018). Ανάκτηση από Javatpoint: <https://www.javatpoint.com/history-of-java>
- Java (programming language).* (2020). Ανάκτηση από Wikipedia: [https://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))
- Petrovan, B. (2019). Ανάκτηση από Android Authority: <https://www.Androidauthority.com/Android-brand-refresh-overdue-1021462/>
- Raphael, J. (2020). Ανάκτηση από Computer World: <https://www.computerworld.com/article/3235946/Android-versions-a-living-history-from-1-0-to-today.html>
- SQL. (2020). Ανάκτηση από Wikipedia: <https://en.wikipedia.org/wiki/SQL>
- Tamada, R. (2017). Ανάκτηση από Android Hive: <https://www.Androidhive.info/2012/05/how-to-connect-Android-with-php-mysql/>
- Team, A. S. (2020). Ανάκτηση από Android Developers: <https://developer.Android.com/studio/intro>
- Team, A. S. (2020). Ανάκτηση από <https://developer.Android.com/guide/slices/getting-started>
- William, J. (2020). Ανάκτηση από Temok: <https://www.temok.com/blog/Android-version-list/>
- Wolfe, J. (2018). Ανάκτηση από Medium: <https://medium.com/@johnwolfe820/the-history-of-php-ffb920ba4555>
- ΜΑΙΡΗ, Γ. (2019). *ΣΥΣΤΗΜΑΤΑ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ.*