

# ΠΡΟΗΓΜΕΝΑ ΘΕΜΑΤΑ ΛΟΓΙΣΜΙΚΟΥ ΥΠΗΡΕΣΙΩΝ ΙΣΤΟΥ.



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ  

---

UNIVERSITY OF PIRAEUS

Πανεπιστήμιο Πειραιώς.

Τμήμα Πληροφορικής.

ΠΜΣ «Προηγμένα Συστήματα Πληροφορικής –  
Ανάπτυξη Λογισμικού και Τεχνητής Νοημοσύνης».

Απαλλακτική Εργασία Εαρινού Εξαμήνου 2023.

24.09.2023

Τίτλος Εργασίας: IF\_shop Java Android Application.

Διδάσκων: Σακκόπουλος Ευάγγελος.

Όνομα: Ασημακόπουλος Γεώργιος.

Αριθμός Μητρώου: ΜΠΣΠ20004.

MS Teams e-mail: [mpsp20004@unipi.gr](mailto:mpsp20004@unipi.gr)

# Πίνακας περιεχομένων

Κεφάλαιο 1: Εισαγωγή .....	2
1.1 Συνοπτική περιγραφή.....	2
1.2 Βασικές λειτουργίες.....	2
Κεφάλαιο 2: Προγράμματα & τεχνολογίες.....	3
2.1 Cloud Design Patterns .....	3
2.1.1 Γενικές πληροφορίες .....	3
2.1.2 Authentication and Authorization .....	4
2.1.3 Fallback .....	5
2.1.4 Data Synchronization .....	6
2.1.5 Υλοποίηση στην εφαρμογή.....	7
2.2 Η γλώσσα προγραμματισμού Java.....	10
2.3 Android .....	11
2.4 Android Studio .....	12
2.5 Firebase Database .....	13
2.6 MySQL Workbench .....	14
Κεφάλαιο 3: Η εφαρμογή.....	15
3.1 Λήψη .....	15
3.2 Εκτέλεση .....	16
3.3 Εικόνες.....	17
3.4 Βίντεο παρουσίασης.....	22
Βιβλιογραφία .....	23

# Κεφάλαιο 1: Εισαγωγή

## 1.1 Συνοπτική περιγραφή

Η κύρια λειτουργία της εφαρμογής, είναι η δημιουργία μίας λίστας από ψώνια από τον χρήστη. Ο χρήστης, μπορεί να δηλώσει το προϊόν, όπου θέλει, όπως και την ποσότητα του. Στην συνέχεια, μπορεί επίσης να δει με τι καιρό θα ψωνίσει (θερμοκρασία). Η εφαρμογή «διαβάζει» τις τρέχουσες συντεταγμένες του χρήστη και του δίνει την θερμοκρασία, στην τοποθεσία όπου βρίσκεται.

## 1.2 Βασικές λειτουργίες

Οι βασικές λειτουργίες της εφαρμογής, είναι οι παρακάτω:

- ✓ Σύνδεση υπάρχων χρήστη ή εγγραφή νέου (ταυτοποίηση μέσω της βάσης δεδομένων ή εγγραφή νέου στοιχείου στην βάση).
- ✓ Επίσης, υπάρχει και εναλλακτική σύνδεση ως διαχειριστής. Ο διαχειριστής, μπορεί να «διαγράψει» έναν χρήστη μέσω του μέιλ του (ο χρήστης διαγράφεται και από την βάση ταυτόχρονα).
- ✓ Ανάκτηση κωδικού υπάρχων χρήστη μέσω μέιλ, αν έχει ξεχάσει τον κωδικό του, στην εφαρμογή.
- ✓ Αλλαγή και αποθήκευση βασικών στοιχείων του χρήστη (όνομα, ηλικία, χώρα). Ενημέρωση γίνεται επίσης και στην βάση δεδομένων.
- ✓ Δημιουργία λίστας από ψώνια. Ο χρήστης μπορεί να δηλώσει το προϊόν και την ποσότητα του, και στην συνέχεια, μπορεί να δει όλα αυτά τα προϊόντα σε μορφή λίστας.
- ✓ Ο χρήστης, μπορεί να δει με τι καιρό (θερμοκρασία) θα πραγματοποιήσει τα ψώνια του. Η εφαρμογή «διαβάζει» τις τρέχουσες συντεταγμένες της τοποθεσίας του χρήστη και «δίνει» τον καιρό στο σημείο αυτό.
- ✓ Έξοδος χρήστη (και διαχειριστή) από την εφαρμογή.

## Κεφάλαιο 2: Προγράμματα & τεχνολογίες

Στο κεφάλαιο αυτό, αρχικά θα αναλυθούν ένα προς ένα τα cloud design patterns που χρησιμοποιήθηκαν στην εφαρμογή και πως αυτά υλοποιήθηκαν. Επίσης, παρατίθενται λίγα λόγια για κάθε πρόγραμμα και τεχνολογία (γλώσσες προγραμματισμού και frameworks) που χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής.

### 2.1 Cloud Design Patterns

#### 2.1.1 Γενικές πληροφορίες

Τα cloud design patterns, αποτελούν βέλτιστες αρχιτεκτονικές πρακτικές που βοηθούν τους προγραμματιστές να σχεδιάσουν και να δημιουργήσουν αξιόπιστες, επεκτάσιμες και ανθεκτικές εφαρμογές που βασίζονται στο cloud. Αν και αυτές οι τεχνικές δεν είναι αποκλειστικά για Java ή Android, μπορούν να χρησιμοποιηθούν σε εφαρμογές Java Android όταν αυτές οι εφαρμογές αλληλεπιδρούν με υπηρεσίες cloud. Στην παρακάτω λίστα, παρατίθενται μερικά κοινά cloud design patterns, τα οποία μπορούν να εφαρμοστούν σε τέτοιου τύπου εφαρμογές:

1. Microservices.
2. Serverless.
3. Caching.
4. Load Balancing.
5. Database Replication.
6. Queue-Based Messaging.
7. Retry and Circuit Breaker.
8. Authentication and Authorization.
9. Monitoring and Logging.
10. Scalability and Elasticity.

Κατά την υλοποίηση αυτών των cloud design patterns σε μία Java Android εφαρμογή, είναι σημαντικό να λάβουμε υπόψη τις συγκεκριμένες απαιτήσεις και περιορισμούς της εφαρμογής μας και να επιλέξουμε τα μοτίβα που ανταποκρίνονται καλύτερα στις τρέχουσες απαιτήσεις. Επιπλέον, θα χρειαστεί να επιλεγούν κατάλληλες υπηρεσίες cloud και βιβλιοθήκες που να «ταιριάζουν» με τα επιλεγμένα μοτίβα και να «ενσωματώνονται» στην αρχιτεκτονική της εφαρμογής.

## 2.1.2 Authentication and Authorization

Είναι ένα από τα θεμελιώδη μοτίβα σχεδιασμού cloud που διασφαλίζουν την ασφάλεια και τον έλεγχο πρόσβασης των πόρων σε μία εφαρμογή η οποία βασίζεται σε μία τέτοια τεχνική. Αυτά τα μοτίβα είναι απαραίτητα για την προστασία ευαίσθητων δεδομένων και λειτουργιών. Ας εμβαθύνουμε στις λεπτομέρειες του κάθε μοτίβου:

- **Authentication:** επαληθεύει την ταυτότητα των χρηστών ή των υπηρεσιών που προσπαθούν να αποκτήσουν πρόσβαση σε ένα σύστημα. Διασφαλίζει ότι οι χρήστες είναι αυτοί που ισχυρίζονται ότι είναι πριν από την παραχώρηση πρόσβασης. Η υλοποίηση μπορεί να γίνει με ποικίλους τρόπους. Κάποιοι από αυτούς είναι οι εξής: Username & password, MFA(Multi-Factor Authentication), Single Sign-On(SSO), OAuth and OpenID Connect και τέλος API κλειδιά.
- **Authorization:** καθορίζει ποιες ενέργειες ή πόρους επιτρέπεται να έχει πρόσβαση σε ένα πιστοποιημένο χρήστη ή υπηρεσία. Καθορίζει επίσης τα δικαιώματα και τις πολιτικές ελέγχου πρόσβασης σε μία εφαρμογή. Η υλοποίηση μπορεί να γίνει με ποικίλους τρόπους. Κάποιοι από αυτούς είναι οι εξής: Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC), Policy-Based Authorization και τέλος Claims-Based Authorization.

### **Βέλτιστες τεχνικές:**

- ✓ Χρήση ισχυρών αλγορίθμων κρυπτογράφησης και κατακερματισμού για την προστασία των αποθηκευμένων κωδικών πρόσβασης.
- ✓ Εφαρμογή μηχανισμών περιορισμού ρυθμού και κλειδώματος λογαριασμού για να αποτραπούν επιθέσεις στην εφαρμογή.
- ✓ Έλεγχος και τακτική ενημέρωση των μεθόδων ελέγχου ταυτότητας για να είναι η εφαρμογή «μπροστά» από τις σύγχρονες απειλές ασφάλειας.
- ✓ Εφαρμογή της αρχής του ελάχιστου προνομίου, το οποίο εξασφαλίζει ότι οι χρήστες ή οι υπηρεσίες θα πρέπει να έχουν το ελάχιστο επίπεδο πρόσβασης που απαιτείται για την εκτέλεση των καθηκόντων τους.
- ✓ Έλεγχος και τακτική ενημέρωση των πολιτικών ελέγχου πρόσβασης.
- ✓ Χρήση αρχείων καταγραφής ελέγχου και παρακολούθηση της πρόσβασης και τον εντοπισμό μη εξουσιοδοτημένων δραστηριοτήτων.

### 2.1.3 Fallback

Είναι γνωστό και ως Circuit Breaker pattern, αποτελεί ένα μοτίβο ανθεκτικότητας που συμβάλλει στην βελτίωση της διαθεσιμότητας και της ανοχής σφαλμάτων των καταναμημένων συστημάτων, συμπεριλαμβανομένων των εφαρμογών που βασίζονται στο cloud. Χρησιμοποιείται κυρίως για το χειρισμό και την ανάκτηση από αστοχίες σε απομακρυσμένες υπηρεσίες ή εξαρτήσεις πόρων.

#### **Βασικός σκοπός:**

Αντιμετωπίζει το πρόβλημα των κλιμακωτών αστοχιών και των μη ανταποκρινόμενων εξαρτήσεων σε ένα καταναμημένο σύστημα. Σκοπός του είναι να αποτρέψει το σύστημα από το να υποβάλλει συνεχώς αιτήματα σε μία αποτυχημένη υπηρεσία ή πόρο, γεγονός που θα μπορούσε να οδηγήσει σε υποβαθμισμένη απόδοση ή αστοχίες σε όλο το σύστημα. Αντίθετα, εισάγει έναν μηχανισμό για να ανιχνεύει τότε μία εξάρτηση βρίσκεται σε κατάσταση αποτυχίας και προσωρινά «σπάει» την σύνδεση με αυτήν την εξάρτηση, ανακατευθύνοντας αιτήματα σε εναλλακτικούς πόρους.

#### **Ιδιότητες:**

Κάποιες από τις βασικές ιδιότητες του Fallback cloud design pattern, συνοψίζονται παρακάτω:

- ✓ Closed State.
- ✓ Open State.
- ✓ Fallback mechanism.
- ✓ Reset and Half-Open State.

#### **Πλεονεκτήματα:**

- **Fault Tolerance:** το μοτίβο επαναφοράς ενισχύει την ανοχή σφαλμάτων του συστήματος αποτρέποντας τις αστοχίες σε ένα μέρος της εφαρμογής από το να επιταχυνθούν και να επηρεάσουν ολόκληρο το σύστημα.
- **Improved Responsiveness:** εντοπίζοντας και απομονώνοντας γρήγορα τις εξαρτήσεις που αποτυγχάνουν, το μοτίβο συμβάλλει στη διατήρηση της ανταπόκρισης και της διαθεσιμότητας της εφαρμογής.
- **Graceful Degradation:** οι εφαρμογές μπορούν να παρέχουν μια υποβάθμιση της λειτουργικότητας όταν μια εξάρτηση δεν είναι διαθέσιμη, διασφαλίζοντας ότι οι χρήστες αντιμετωπίζουν μειωμένη ποιότητα υπηρεσιών αντί για πλήρη αποτυχία.

## 2.1.4 Data Synchronization

Είναι ένα μοτίβο σχεδιασμού cloud που αντιμετωπίζει την πρόκληση της διατήρησης της συνέπειας και της συνοχής των δεδομένων στα κατακευκταμένα συστήματα, ιδιαίτερα σε εφαρμογές που βασίζονται σε cloud. Αυτό το μοτίβο διασφαλίζει ότι τα δεδομένα σε διαφορετικές τοποθεσίες ή μεταξύ διαφόρων στοιχείων παραμένουν ενημερωμένα και ευθυγραμμισμένα μεταξύ τους. Ακολουθούν οι βασικές λεπτομέρειες σχετικά με το μοτίβο σχεδίασης συγχρονισμού δεδομένων:

### **Βασικός σκοπός:**

Το μοτίβο αυτό χρησιμοποιείται κυρίως για να διατηρεί τα δεδομένα συνεπή και συγχρονισμένα σε πολλαπλές βάσεις δεδομένων (τοπικές και μη) ή υπηρεσίες σε μία κατακευκταμένη εφαρμογή. Οι πρωταρχικοί του στόχοι είναι:

- ✓ **Συνέπεια δεδομένων:** βεβαίωση ότι τα δεδομένα παραμένουν συνεκτικά και ενημερωμένα σε όλα τα σχετικά μέρη της εφαρμογής.
- ✓ **Επίλυση διενέξεων:** χειρισμός διενέξεων που μπορεί να προκύψουν όταν γίνεται ενημέρωση δεδομένων ταυτόχρονα από διαφορετικές πηγές.
- ✓ **Επεκτασιμότητα:** ενεργοποίηση της εφαρμογής, έτσι ώστε να κλιμακώνεται οριζόντια, διατηρώντας παράλληλα την συνοχή των δεδομένων.

### **Ιδιότητες:**

Κάποιες από τις βασικές ιδιότητες του Data Synchronization cloud design pattern, συνοψίζονται παρακάτω:

- ✓ Master Data Store.
- ✓ Replica Data Stores.
- ✓ Synchronization Mechanism.
- ✓ Conflict Resolution Logic.

### **Πεδία εφαρμογής:**

Αυτό το μοτίβο σχεδίασης, βρίσκει εφαρμογή στα παρακάτω σενάρια:

- ✓ **Geographically Distributed Applications:** όταν μία εφαρμογή έχει χρήστες ή στοιχεία σε διαφορετικές περιοχές ή κέντρα δεδομένων, ο συγχρονισμός δεδομένων διασφαλίζει ότι τα δεδομένα είναι προσβάσιμα με μικρή καθυστέρηση.
- ✓ **Multi-Data Center Deployments:** σε περιβάλλοντα cloud, όπου μια εφαρμογή αναπτύσσεται σε πολλά κέντρα δεδομένων ή ζώνες διαθεσιμότητας, ο συγχρονισμός δεδομένων είναι ζωτικής σημασίας για την ανάκτηση από καταστροφές και την υψηλή διαθεσιμότητα.

- ✓ **Collaborative Editing:** οι εφαρμογές που υποστηρίζουν τη συλλογική επεξεργασία ή τη συνεργασία σε πραγματικό χρόνο, όπως τα Έγγραφα Google ή τα εργαλεία διαχείρισης έργων, απαιτούν συγχρονισμένα δεδομένα από πολλούς χρήστες.
- ✓ **Κινητές εφαρμογές:** οι εφαρμογές για κινητά συχνά λειτουργούν εκτός σύνδεσης και πρέπει να συγχρονίσουν δεδομένα με τον διακομιστή όταν αποκαθίσταται η συνδεσιμότητα.

### 2.1.5 Υλοποίηση στην εφαρμογή

Στην υπό-ενότητα αυτή, θα αναλυθεί κυρίως ο τρόπος πού έγινε η υλοποίηση των μοτίβων αυτών (σε ποια αρχεία και ποιος ο σκοπός), μέσα στην τρέχουσα εφαρμογή:

#### **Authentication and Authorization:**

Ο σκοπός του μοτίβου αυτού στην εφαρμογή, είναι ο διαχωρισμός των σελίδων ανάλογα με τον ρόλο πού έχει ο χρήστης στην εφαρμογή, δηλαδή αν είναι απλός χρήστης ή διαχειριστής. Αν πραγματοποιηθεί σύνδεση ως απλός χρήστης, η εφαρμογή μας εμφανίζει την κεντρική σελίδα, διαφορετικά, αν γίνει σύνδεση με ρόλο διαχειριστή (συγκεκριμένο μέιλ), εμφανίζεται άλλη σελίδα. Με γνώμονα την βάση δεδομένων της εφαρμογής, γίνεται η επιλογή αυτή, αφού πρώτα ταυτοποιηθεί ο χρήστης, πού κάνει την σύνδεση. Η υλοποίηση του μοτίβου αυτού, έγινε στα παρακάτω αρχεία:

- **LoginActivity.java**

Μέσω της Firebase Database γίνεται αρχικά ταυτοποίηση του χρήστη, μέσω της μεθόδου `loginUser()`, στην συνέχεια, πραγματοποιείται ο διαχωρισμός και ανάλογα με το μέιλ, η εμφάνιση της κατάλληλης σελίδας, μέσω της μεθόδου `showProperActivity(FirebaseAuth mAuth)`. Η τελευταία μέθοδος εμφανίζει είτε την σελίδα `AdminActivity.java` ή την `MainActivity.java`.



### **Fallback:**

Ο βασικός σκοπός του μοτίβου αυτού, είναι να παρέχει στον χρήστη, την θερμοκρασία της τρέχουσας τοποθεσίας του. Η εφαρμογή πρέπει να έχει πρόσβαση στο διαδίκτυο, όπως επίσης υπάρχει και ερώτηση προς τον χρήστη, για την χρήση της τοποθεσίας του (αν το επιτρέπει ή όχι). Το μοτίβο αυτό υλοποιήθηκε με την βοήθεια ενός API κλειδιού και την «κλήση» σε μία σελίδα, όπου μας δίνει καιρικές πληροφορίες, σύμφωνα με το γεωγραφικό μήκος και πλάτος της τρέχουσας τοποθεσίας του χρήστη. Η υλοποίηση του έγινε στα παρακάτω αρχεία:

- **AndroidManifest.xml**  
Στο αρχείο αυτό, έγινε η προσθήκη των δικαιωμάτων της εφαρμογής για την χρήση του διαδικτύου, αλλά και της τοποθεσίας του χρήστη.
- **WeatherService.java**  
Στον φάκελο `networking` της εφαρμογής υλοποιείται το interface αυτό, το οποίο περιέχει μία μέθοδο, όπου δέχεται τις συντεταγμένες του χρήστη, όπως και το API κλειδί για την ανάκτηση των καιρικών δεδομένων.
- **RetrofitClient.java**  
Στον φάκελο `networking`, υλοποιείται η κλάση αυτή, η οποία κάνει κλήση στην αντίστοιχη σελίδα, έτσι ώστε μέσω του API κλειδιού και του URL, να λάβει τα καιρικά δεδομένα.
- **WeatherResponse.java & WeatherMain.java**  
Αυτές οι δύο κλάσεις, υλοποιούνται μέσα στον φάκελο `models` της εφαρμογής και αποθηκεύουν τα σχετικά καιρικά δεδομένα.
- **WeatherActivity.java**  
Τέλος, το αρχείο αυτό καλεί την μέθοδο για την λήψη της θερμοκρασίας, σύμφωνα με την τοποθεσία του χρήστη. Διαβάζει, το τρέχων γεωγραφικό μήκος και πλάτος και εμφανίζει την θερμοκρασία στον χρήστη, αφού αυτός ενεργοποιήσει το αντίστοιχο κουμπί και επιτρέψει τα απαραίτητα δικαιώματα.

### Data Synchronization:

Ο σκοπός του μοτίβου αυτού είναι ο συγχρονισμός των δεδομένων, που υπάρχουν σε μία απομακρυσμένη βάση δεδομένων σε σχέση με μία τοπική βάση. Και οι μορφές βάσεων χρησιμοποιούνται από την ίδια εφαρμογή. Στην περίπτωση μας, οι βάσεις δεδομένων αποθηκεύουν τα προϊόντα, που έχει προσθέσει ο κάθε χρήστης στο καλάθι του. Στο παράδειγμα αυτό, τον ρόλο της απομακρυσμένης βάσης, την έχει η Firebase Database, και τον ρόλο της τοπικής βάσης, την έχει η MySQL. Η εφαρμογή προσθέτει δεδομένα και στις δύο βάσεις ταυτόχρονα και επίσης ελέγχει αν έχουν τα ίδια στοιχεία και ανάλογα ενημερώνει. Η υλοποίηση του μοτίβου αυτού, έγινε στα παρακάτω αρχεία:

- `LocalDBHelper.java`

Η κλάση αυτή, η οποία βρίσκεται μέσα στην φάκελο `local_DB`, μας βοηθά να αποθηκεύσουμε ένα νέο προϊόν στην τοπική βάση ή να ανακτήσουμε τα δεδομένα από αυτήν σε μορφή λίστας από προϊόντα, σύμφωνα με εκάστοτε χρήστη.

- `SyncDBService.java`

Η κλάση αυτή, η οποία βρίσκεται στον φάκελο `service`, παίρνει τις λίστες των προϊόντων και από τις δύο βάσεις δεδομένων (απομακρυσμένη και τοπική), κάνει την σύγκριση ανάμεσα σε όλα τα ID, και όποιο δεν υπάρχει σε μία από τις, το ανεβάζει, έτσι ώστε να επιτευχθεί ο συγχρονισμός.

## 2.2 Η γλώσσα προγραμματισμού Java

Η Java είναι μία γλώσσα προγραμματισμού και πλατφόρμα (περιβάλλον που τρέχει προγράμματα). Είναι μία υψηλού επιπέδου, ισχυρή, αντικειμενοστραφής και ασφαλής γλώσσα προγραμματισμού. Αναπτύχθηκε από την Sun Microsystems (όπου τώρα είναι θυγατρική της Oracle) το έτος 1995. Ο James Gosling έχει μείνει γνωστός ως ο «πατέρας» της γλώσσας. Πριν από το όνομα Java, υπήρχε το Oak. Λόγω του ότι το δεύτερο ήταν ήδη καταχωρημένο από άλλη εταιρεία, ο James Gosling και η ομάδα του άλλαξαν το όνομα της γλώσσας από Oak σε Java.

### Εφαρμογές Java

Σύμφωνα με την Sun, 3 δισεκατομμύρια συσκευές «τρέχουν» Java. Υπάρχουν πάρα πολλές συσκευές που αυτήν την στιγμή εκτελούν την γλώσσα. Μερικές από αυτές είναι:

1. Εφαρμογές υπολογιστών όπως acrobat reader, media players, antivirus και λοιπά.
2. Διαδικτυακές εφαρμογές.
3. Εταιρικές εφαρμογές, όπως τραπεζικές εφαρμογές.
4. Κινητά τηλέφωνα.
5. Ενσωματωμένα συστήματα.
6. Έξυπνες κάρτες.
7. Ρομποτική.
8. Βίντεο παιχνίδια, και άλλα...



**Εικόνα :** Το επίσημο λογότυπο της Java.

## 2.3 Android

Το Android είναι ένα λειτουργικό σύστημα κινητών βασισμένο σε μία τροποποιημένη έκδοση του Linux πυρήνα, αλλά και σε άλλα λογισμικά ανοικτού-κώδικα, το οποίο σχεδιάστηκε κυρίως για κινητές συσκευές αφής όπως smartphones και tablets. Αναπτύχθηκε αρχικά από την Google και αργότερα από την OHA (Open Handset Alliance, κοινοπραξία πολλών εταιρειών). Η γλώσσα προγραμματισμού Java χρησιμοποιείται κυρίως για την σύνταξη του κώδικα Android, βέβαια υπάρχει και η δυνατότητα χρήσης και άλλων παρόμοιων γλωσσών. Ο βασικός στόχος ενός android πρότζεκτ είναι η δημιουργία ενός επιτυχημένου προϊόντος αληθινού-κόσμου, το οποίο θα βελτιώσει την εμπειρία των τελικών χρηστών στα κινητά τηλέφωνα. Υπάρχουν πάρα πολλά κωδικά-ονόματα για το Android όπως Lollipop, KitKat, Jelly Bean, Donut και άλλα.

Για την δημιουργία της παρούσας εφαρμογής, έγινε χρήση τού:

- Android Studio Flamingo | 2022.2.1 Patch 2 (May 2023).

### Χαρακτηριστικά

Τα κύρια χαρακτηριστικά του λειτουργικού συστήματος Android παρατίθενται παρακάτω:

1. Είναι σύστημα ανοικτού-κώδικα.
2. Ο οποιοσδήποτε μπορεί να μορφοποιήσει την Android πλατφόρμα.
3. Υπάρχει μία τεράστια πληθώρα εφαρμογών για κινητές συσκευές, όπου μπορεί ο τελικός χρήστης να διαλέξει και να χρησιμοποιήσει.
4. Προσφέρει πολλές χρήσιμες και ενδιαφέρουσες λειτουργίες, όπως πρόβλεψη καιρού, άνοιγμα οθονών, υπηρεσίες μηνυμάτων (SMS & MMS), αποθήκευση βάσης δεδομένων, συνδεσιμότητα (Bluetooth, Wi-Fi), internet, media και άλλα.
5. Υποστηρίζεται από έξυπνα τηλέφωνα, tablets, υπολογιστές, έξυπνες τηλεοράσεις και άλλα.



Εικόνα: Το επίσημο λογότυπο του Android.

## 2.4 Android Studio

Το Android Studio είναι το επίσημο Ολοκληρωμένο Περιβάλλον Ανάπτυξης (Integrated Development Environment, IDE) για την δημιουργία και ανάπτυξη android εφαρμογών. Ανακοινώθηκε την 16η Μαΐου 2013 από την Google ως το επίσημο IDE για την ανάπτυξη android εφαρμογών. Η πρώτη σταθερή έκδοση, δημοσιεύθηκε τον Δεκέμβριο του 2014, ξεκινώντας την αρίθμηση από την έκδοση 1.0. Από την 7η Μαΐου 2019, η Kotlin είναι η γλώσσα προτίμησης για την ανάπτυξη των εφαρμογών, εκτός από αυτήν το Android Studio υποστηρίζει και άλλες γλώσσες προγραμματισμού.



**Εικόνα:** Το επίσημο λογότυπο του Android Studio.

### Χαρακτηριστικά

Το πρόγραμμα προσφέρει ακόμη περισσότερες δυνατότητες που ενισχύουν την παραγωγικότητα, κατά την δημιουργία εφαρμογών android, όπως:

- Διαθέτει ένα ευέλικτο σύστημα κατασκευής που βασίζεται σε Gradle.
- Παρέχει έναν γρήγορο και πλούσιο σε δυνατότητες εξομοιωτή για δοκιμές εφαρμογών.
- Διαθέτει ένα ενοποιημένο περιβάλλον, όπου μπορούμε να αναπτύξουμε εφαρμογές για όλες τις συσκευές που υποστηρίζουν το λειτουργικό.
- Μπορούμε να κάνουμε αλλαγές στον κώδικα της εφαρμογής και να την εκτελέσουμε, χωρίς να χρειαστεί επανεκκίνησή της.
- Παρέχει εκτεταμένα εργαλεία και πλαίσια δοκιμών.
- Υποστηρίζει C++ και NDK.
- Παρέχει ενσωματωμένη υποστήριξη για την πλατφόρμα Google Cloud.
- Μπορεί να χρησιμοποιηθεί από όλα τα λειτουργικά συστήματα (Windows, macOS, Linux).

## 2.5 Firebase Database

Η Firebase (βάση δεδομένων) είναι ένα προϊόν της Google που βοηθά τους προγραμματιστές να δημιουργούν, να διαχειρίζονται και να αναπτύσσουν τις εφαρμογές τους εύκολα. Βοηθά τους προγραμματιστές να δημιουργήσουν τις εφαρμογές τους πιο γρήγορα και με πιο ασφαλή τρόπο. Δεν απαιτείται προγραμματισμός από την πλευρά της βάσης Firebase, γεγονός που την καθιστά εύκολη ως προς την χρήση των δυνατοτήτων της και πιο αποτελεσματική. Παρέχει υπηρεσίες σε android, iOS, web και unity. Παρέχει αποθήκευση cloud. Χρησιμοποιεί τύπου NoSQL βάση δεδομένων για την αποθήκευση δεδομένων. Πρακτικά, το τελευταίο σημαίνει ότι η συγκεκριμένη βάση δεν κάνει χρήση πινάκων για την αποθήκευση και ανάκτηση δεδομένων, αλλά αρχείων τύπου JSON (JavaScript Object Notation).



**Εικόνα:** Το επίσημο λογότυπο της Firebase Database.

### Χαρακτηριστικά

Μερικά από τα βασικά χαρακτηριστικά της Firebase, συνοψίζονται παρακάτω:

- Βάση δεδομένων πραγματικού χρόνου: βάση δεδομένων τύπου NoSQL που βασίζεται σε cloud τεχνολογία και διαχειρίζεται τα δεδομένα με απίστευτη ταχύτητα σε χιλιοστά του δευτερολέπτου. Πιο απλά, μπορεί να θεωρηθεί ως ένα μεγάλο αρχείο JSON.
- Αυθεντικοποίηση: η υπηρεσία αυτή παρέχει εύχρηστες βιβλιοθήκες διεπαφής χρήστη και SDK για τον έλεγχο ταυτότητας των χρηστών στην εφαρμογή μας. Χειρίζεται ακόμη και εργασίες όπως η συγχώνευση λογαριασμών, η οποία αν γίνει χειροκίνητα είναι δύσκολη και χρονοβόρα.
- Hosting: η βάση παρέχει φιλοξενία εφαρμογών με ταχύτητα και ασφάλεια. Μπορεί να χρησιμοποιηθεί για την φιλοξενία στατικών ή δυναμικών σελίδων. Έχει την δυνατότητα φιλοξενίας μίας εφαρμογής με μία και μόνο εντολή.
- Παρακολούθηση απόδοσης: αυτή η υπηρεσία δίνει μία εικόνα για τα χαρακτηριστικά απόδοσης των εφαρμογών.
- Google Analytics: είναι μία δωρεάν υπηρεσία μέτρησης εφαρμογών που παρέχεται από την Google και παρέχει πληροφορίες σχετικά με τη χρήση της εφαρμογής και τους χρήστες.

## 2.6 MySQL Workbench

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Το πρόγραμμα τρέχει έναν εξυπηρετητή (server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων. Ανήκει και χρηματοδοτείται από μία και μοναδική κερδοσκοπική εταιρία, τη σουηδική MySQL AB, η οποία σήμερα ανήκει στην Oracle. Η MySQL είναι δημοφιλής βάση δεδομένων για διαδικτυακά προγράμματα και ιστοσελίδες. Χρησιμοποιείται σε κάποιες από τις πιο διαδεδομένες διαδικτυακές υπηρεσίες, όπως το Flickr, το YouTube, η Wikipedia, το Google, το Facebook και το Twitter. Το MySQL Workbench είναι ένα ενοποιημένο οπτικό εργαλείο για αρχιτέκτονες βάσεων δεδομένων, προγραμματιστές και DBA. Το MySQL Workbench παρέχει μοντελοποίηση δεδομένων, ανάπτυξη SQL και ολοκληρωμένα εργαλεία διαχείρισης για διαμόρφωση διακομιστή, διαχείριση χρήστη, δημιουργία αντιγράφων ασφαλείας και πολλά άλλα. Το MySQL Workbench είναι διαθέσιμο σε Windows, Linux και Mac OS X.



**Εικόνα :** Το επίσημο λογότυπο του MySQL Workbench.

## Κεφάλαιο 3: Η εφαρμογή

### 3.1 Λήψη

Μπορούμε να κάνουμε λήψη της εφαρμογής από το repository που έχει δημιουργηθεί για τον σκοπό αυτό, στο github, ακολουθώντας τα παρακάτω βήματα:

1. Μεταβαίνουμε στην παρακάτω σελίδα (remote repository του πρότζεκτ):  
[https://github.com/GeorgeCR7/IF\\_shop\\_android\\_app](https://github.com/GeorgeCR7/IF_shop_android_app)
2. Κάνουμε κλικ στο μπλε κουμπί πάνω δεξιά <>Code, εκεί έχουμε δύο επιλογές, είτε κάνουμε λήψη του πρότζεκτ ως συμπιεσμένο αρχείο (Download ZIP), είτε μέσω του git, κάνουμε λήψη του πρότζεκτ.
3. Αν ακολουθήσουμε την δεύτερη μέθοδο, πρέπει αρχικά, να έχουμε το git στον υπολογιστή μας. Αυτό μπορούμε να το ελέγξουμε, ανοίγοντας ένα τερματικό και δίνοντας την εντολή:  
`$ git --version`
4. Αν το τερματικό μας επιστρέφει την έκδοση του git (πχ git version 2.37.3), δεν κάνουμε κάτι, διαφορετικά κάνουμε εγκατάσταση του git στον υπολογιστή μας, από την σελίδα [αυτή](#).
5. Κάνουμε αντιγραφή τον HTTPS σύνδεσμο που έχουμε από το βήμα 2, για το repository αυτό.
6. Τέλος, μεταβαίνουμε τοπικά στον φάκελο που θέλουμε να κατεβάσουμε το πρότζεκτ, ανοίγουμε τερματικό σε αυτόν και δίνουμε την εντολή:  
`$ git clone <repository_HTTPS_url>`



## 3.2 Εκτέλεση

Για να «εκτελέσουμε» την εφαρμογή στον δικό μας υπολογιστή, πρέπει αρχικά να έχουμε κάνει λήψη του κώδικα (παραπάνω παράγραφος), στην συνέχεια, ακολουθούμε τα παρακάτω βήματα:

1. Ανοίγουμε το πρόγραμμα Android Studio στον υπολογιστή μας, αν δεν το έχουμε, μπορούμε να το κατεβάσουμε από [εδώ](#) και το κάνουμε εγκατάσταση σύμφωνα με τις [οδηγίες](#).
2. Στο κεντρικό παράθυρο του προγράμματος, κάνουμε κλικ στο κουμπί Open, και στο μικρό παράθυρο που ανοίγει, επιλέγουμε τον φάκελο του πρότζεκτ, όπου κατεβάσαμε από παραπάνω και το ανοίγουμε.
3. Κάτω από την βασική μπάρα/μενού του προγράμματος (αριστερά από το πράσινο κουμπί εκτέλεσης της εφαρμογής) δημιουργούμε μία εικονική συσκευή (Device Manager > Create Device). Στο πρότζεκτ αυτό έγινε χρήση της εικονικής συσκευής Pixel 4 API 30 Android 11.0 Google Play.
4. Στην συνέχεια, κάνουμε κλικ στο πράσινο κουμπί εκτέλεσης της εφαρμογής (ή διαφορετικά Shift + F10).

### 3.3 Εικόνες

Στην ενότητα αυτή, θα παρουσιαστούν μία προς μία, με την μορφή εικόνων, οι σελίδες της εφαρμογής:



**Εικόνα:** Η σελίδα σύνδεσης του χρήστη (απλού ή διαχειριστή).

The image shows a registration form titled "Εγγραφή" (Registration) in bold black text. Below the title are three input fields with placeholder text: "Email...", "Κωδικός..." (Password...), and "Κωδικός ξανά..." (Password again...). Below the input fields are two buttons: a purple button labeled "Εγγραφή" (Register) and a purple button labeled "Ήδη χρήστης; Σύνδεση εδώ!" (Already a user? Log in here!).

**Εικόνα:** Η σελίδα εγγραφής νέου χρήστη στην εφαρμογή.

The image shows a form titled "Το email σου, στην εφαρμογή..." (Your email in the app...). Below the title is an input field with placeholder text "Email...". Below the input field is a purple button labeled "Ανάκτηση κωδικού" (Retrieve password). At the bottom of the form is a red button labeled "Πίσω" (Back).

**Εικόνα:** Η σελίδα ανάκτησης του λογαριασμού.

## Σελίδα διαχειριστή

Email χρήστη προς διαγραφή...

Διαγραφή

Εξοδος

**Εικόνα:** Η σελίδα του διαχειριστή.

# IF\_shop

Κυριακή, 17.09.2023

Κάνε αρχικά την

λίστα

με τα ψώνια σου!...

...και δεξ εδώ με τι

καιρό

θα ψωνίσεις!

Προφίλ

Εξοδος

**Εικόνα:** Η κεντρική σελίδα της εφαρμογής.

**E-mail**  
1@1.gr

**Ημερομηνία δημιουργίας λογαριασμού**  
26.08.2023

**Όνοματεπώνυμο**  
Δεν υπάρχει όνομα

**Ηλικία**  
Δεν υπάρχει ηλικία

**Χώρα**  
Δεν υπάρχει χώρα

**Αποθήκευση**

**Πίσω**

**Εικόνα:** Η σελίδα όπου ο χρήστης διαχειρίζεται το προφίλ του.

**Επιλογή προϊόντος**

Ποσότητα  
1

**Προσθήκη**

**Δες εδώ την**  
λίστα

**Πίσω**

**Εικόνα:** Η σελίδα, όπου ο χρήστης, κάνει προσθήκη προϊόντος στην λίστα του.



**Εικόνα:** Η σελίδα όπου αναφέρεται με τι καιρό θα ψωνίσει ο χρήστης.

### 3.4 Βίντεο παρουσίασης

Στον παρακάτω σύνδεσμο, υπάρχει ένα βίντεο παρουσίασης της εφαρμογής, από τον Emulator (Pixel 4 API 30) μέσω του προγράμματος Android Studio:

<https://drive.google.com/file/d/1XtoiwWoznEJ9AVCwapYmjb72SIbRnUz2/view?usp=sharing>

Η καταγραφή του βίντεο έγινε με το πρόγραμμα OBS Studio.

## Βιβλιογραφία

Στην παρακάτω λίστα, παρατίθενται όλες οι σελίδες από τις οποίες αντλήθηκαν πληροφορίες για την δημιουργία της παρούσας εργασίας:

- Java (Programming language) – Wikipedia.  
Σελίδα: [https://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))
- MySQL – Wikipedia.  
Σελίδα: <https://el.wikipedia.org/wiki/MySQL>
- MySQL :: MySQL Workbench Official Website.  
Σελίδα: <https://www.mysql.com/products/workbench/>
- Android (operating system) – Wikipedia. (Ενημέρωση: 15.11.2022).  
Σελίδα: [https://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))
- What is android – javatpoint.  
Σελίδα: <https://www.javatpoint.com/android-what-where-and-why>
- Meet Android Studio – Android Developers.  
Σελίδα: <https://developer.android.com/studio/intro>
- Android Studio – javatpoint.  
Σελίδα: <https://www.javatpoint.com/android-studio>
- Firebase official website.  
Σελίδα: <https://firebase.google.com/>
- Firebase – Introduction – GeeksforGeeks. (Ενημέρωση: 15.07.2021).  
Σελίδα: <https://www.geeksforgeeks.org/firebase-introduction/>
- OBS Studio (Open Broadcaster Software).  
Σελίδα: <https://obsproject.com/>
- Git SCM.  
Σελίδα: <https://git-scm.com/>
- GitHub: Let's build from here.  
Σελίδα: <https://github.com/>
- Lightshot – screenshot tool for Mac & Win.  
Σελίδα: <https://app.prntscr.com/en/index.html>
- OpenWeather API – Official website.  
Σελίδα: <https://openweathermap.org/api>