



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Πρόγραμμα Μεταπτυχιακών Σπουδών
«Πληροφορική»**

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Εφαρμογή Android για παραγγελιοληψία γραμμένη σε Kotlin An Android Order Management application written in Kotlin
Όνοματεπώνυμο Φοιτητή	ΣΤΥΛΙΑΝΟΣ ΑΓΓΕΛΟΠΟΥΛΟΣ
Πατρώνυμο	ΠΑΝΑΓΙΩΤΗΣ
Αριθμός Μητρώου	ΜΠΠΛ20001
Επιβλέπων	Ευθύμιος Αλέπης, Αναπληρωτής Καθηγητής

Ημερομηνία Παράδοσης **Μάιος 2023**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

Ευθύμιος Αλέπης
Αναπληρωτής Καθηγητής

(υπογραφή)

Μαρία Βίρβου
Καθηγήτρια

(υπογραφή)

Ευάγγελος Σακκόπουλος
Αναπληρωτής Καθηγητής

Περιεχόμενα

Κατάλογος Εικόνων	5
Λίστα Διαγραμμάτων	6
Ευχαριστίες	7
Περίληψη	7
Abstract	8
1. Εισαγωγή	8
1.1 Χώρος του Προβλήματος	9
1.2 Στόχοι εργασίας	9
1.3 Τι είναι μια εφαρμογή παραγγελιοληψίας	9
1.4 Παρόμοιες Εφαρμογές	10
1.4.1 Prognosi.....	10
1.4.2 Inorder mobile.....	11
1.4.3 Cardlink APP	12
1.5 Σύγκριση Εφαρμογών.	13
2. Φορητές Έξυπνες Συσκευές	13
2.1 Έξυπνα Τηλέφωνα (Smartphones)	13
2.2 Υπολογιστής ταμπλέτα (Tablet)	16
2.3 Λειτουργικά Συστήματα για Κινητά (Mobile OS)	17
2.4 Λειτουργικά Συστήματα Android και iOS	19
2.4.1 Τα Αρνητικά του Λειτουργικού Android.....	20
2.4.2 Τα Αρνητικά του Λειτουργικού iOS.....	21
3. Εφαρμογές για το λειτουργικό Android.	22
3.1 Εργαλεία Ανάπτυξης Εφαρμογών σε Λειτουργικό Android.	23
3.2 Κύριοι τύποι Εφαρμογών για Κινητές Συσκευές.	24
3.2.1 Τα Κριτήρια για να Επιλέξουμε τον τύπο Εφαρμογής.	25
4. Λειτουργικό Σύστημα Android και Τεχνολογίες	26
4.1 Η Αρχιτεκτονική του Λειτουργικού Συστήματος Android	28
4.1.1 Android Linux Kernel.	29
4.1.2 Android Hardware Abstraction Layer (HAL).	29
4.1.3 Android Runtime (ART).....	30
4.1.4 Android Native Libraries.....	30
4.1.5 Android Java API Framework.....	31
4.1.6 Android System APP.	32
4.2 Κύκλος ζωής των δραστηριοτήτων (Activity Life Cycle).	33
4.3 Διεπαφή χρήστη (Android User Interface)	34
4.4 Android Material Design.	35

4.5 Γλώσσα προγραμματισμού Kotlin.....	36
4.6 Δικαιώματα πρόσβασης χρήστη του Android.....	36
4.7 RecyclerView & CardView του Android.	37
4.8 Google Firebase.	37
4.9 JSON.	38
4.10 Εργαλεία Ανάπτυξη.....	39
Στην ενότητα αυτήν θα δούμε τα εργαλεία που χρησιμοποιήσαμε για την ανάπτυξη της διπλωματικής εργασίας.....	39
4.10.1 Android Studio.....	39
4.10.2 GitHub.	39
4.10.3 SmartGit.	39
4.10.4 Vysor.....	40
5. Ανάλυση Λειτουργιών της Εφαρμογής.....	41
5.1 Ανάλυση Απαιτήσεων της Εφαρμογής.	41
5.2 Σχεδιασμός Εφαρμογής.	41
5.2.1 Δυνατότητες και Λειτουργίες των Χρηστών στην Εφαρμογή.	42
5.2.2 Σχεδιασμός Σύνδεσης Χρήστη στην Εφαρμογή.	42
5.2.3 Σχεδιασμός της Βάσης Δεδομένων στην Firebase.	43
6. Παρουσίαση της εφαρμογής Maitre APP.....	46
6.1 Οθόνη Σύνδεσης.	46
6.2 View Σερβιτόρου – Αρχική οθόνη.....	48
6.3 View Σερβιτόρου – Οθόνη Επιλογής Τραπεζιού.....	49
6.4 View Σερβιτόρου – Οθόνη Menu.....	50
6.5 View Κουζίνας – Αρχική οθόνη.....	51
6.6 View Κουζίνας – Οθόνη Orders.	52
6.7 View Κουζίνας – Οθόνη Catalogue.	53
7. Επεξήγηση Κώδικα.	56
7.1 Dark Mode.....	56
7.2 Καταχώριση Νέας Εγγραφής στην Βάση.	57
7.3 Επεξήγηση Λίστας (RecyclerView).....	58
8. Συμπεράσματα και Μελλοντικές Προτάσεις.	59
8.1 Συμπεράσματα.	59
8.2 Μελλοντικές Προτάσεις Βελτίωσης της Εφαρμογής.	60
9. Βιβλιογραφία.....	60

Κατάλογος Εικόνων

Εικόνα 1: Prognosi APP	10
Εικόνα 2: Inorder Mobile APP	11
Εικόνα 3: Cardlink APP.....	12
Εικόνα 4: IBM Simon mobile.....	14
Εικόνα 5: Ericsson T28.....	14
Εικόνα 6: Blackberry 850	15
Εικόνα 7: Apple iPhone 2007 (first iPhone).....	15
Εικόνα 8: Google Android Phone.....	15
Εικόνα 9: Tablets	16
Εικόνα 10: Operating System	17
Εικόνα 11: Λειτουργικά Συστήματα Android και iOS.....	19
Εικόνα 12: Εκδόσεις iOS μέχρι Γενάρη του 2023.....	21
Εικόνα 13: Εργαλεία Ανάπτυξης Εφαρμογών σε Λειτουργικό Android.....	23
Εικόνα 14: Τα θετικά και τα αρνητικά των διάφορων τύπων εφαρμογών	26
Εικόνα 15: Technologies και DEV Tools για το Android.....	27
Εικόνα 16: Android Operating System Layer Architecture.....	28
Εικόνα 17: Android Activity Life Cycle.....	33
Εικόνα 18: Android User Interface.....	34
Εικόνα 19: Η διαφορά πριν και μετά την χρήση του Metarial Design.....	35
Εικόνα 20: Παράδειγμα πρόσβασης δικαιώματα χρήστη.....	36
Εικόνα 21: Παράδειγμα RecyclerView και CardView.....	37
Εικόνα 22: Παράδειγμα JSON μηνύματος.	38
Εικόνα 23: Vysor σε Mac OS λειτουργικό.....	40
Εικόνα 24: Δομή πίνακα menu	44
Εικόνα 25: Δομή πίνακα orders	44
Εικόνα 26: Δομή πίνακα tables.....	45
Εικόνα 27: Firebase Storage	45
Εικόνα 28: Animation πριν την οθόνη σύνδεσης	46
Εικόνα 29: Οθόνη σύνδεσης - login screen	47
Εικόνα 30: Αρχική οθόνη Σερβιτόρου.....	48
Εικόνα 31: Οθόνη Επιλογής Τραπεζιού.	49
Εικόνα 32: Λίστα μετά την παραγγελία (αριστερά) - Λίστα πριν την παραγγελία (δεξιά).	50
Εικόνα 33: Αρχική οθόνη Κουζίνας.	51
Εικόνα 34: Οθόνη με τα τραπέζια που έχουν παραγγελιά (αριστερά) και οθόνη με την παραγγελία του τραπεζιού (δεξιά).	52
Εικόνα 35: Οθόνη με τον κατάλογο και το inventory.	53
Εικόνα 36: Οθόνη με τον κατάλογο (αριστερά) - Οθόνη μενού (δεξιά).	54
Εικόνα 37: Πιάτο ημέρας (αριστερά) - Πρόσθεση πιάτου ημέρας (δεξιά).....	55
Εικόνα 38: Dark mode xml code	56
Εικόνα 39: Παράδειγμα οθόνης με dark mode (αριστερά) και χωρίς (δεξιά).	57
Εικόνα 40: Καταχώριση νέας εγγραφής στην βάση του firebase.....	57
Εικόνα 41: Κώδικας RecyclerView.....	58

Λίστα Διαγραμμάτων

Διαγράμματα 1: Μερίδιο αγοράς MobileOS Q1 2018	18
Διαγράμματα 2: Εκδόσεις Android OS μέχρι Γενάρη του 2023	21
Διαγράμματα 3: Διαθέσιμες εφαρμογές στο Google Play	22
Διαγράμματα 4: Σύνδεση στην εφαρμογή	43

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον κ. Αλέπη για την ανάθεση της μεταπτυχιακής μου διατριβής καθώς και τον φίλο Χασάπη Παντελή για την επίβλεψη και την επιστημονικά άρτια καθοδήγηση του κατά τη διάρκεια του σχεδιασμού και της ανάπτυξης αυτής της εργασίας αλλά και τις καίριες συμβουλές που μου έδωσε για τη βελτίωση της εφαρμογής που αναπτύχθηκε. Την εταιρία στην οποία εργάζομαι Cardlink S.A. που με βοήθησε να φοιτησω στο μεταπτυχιακό πρόγραμμα.

Θερμές ευχαριστίες θα ήθελα να δώσω στην Λιλή Αγγελοπούλου που ήταν κοντά μου από την αρχή των ακαδημαϊκών μου σπουδών πριν από 14 χρόνια μέχρι και σήμερα.

Τέλος, ευχαριστίες οφείλω στην οικογένεια μου και στους φίλους μου, για την ηθική υποστήριξη και βοήθεια όλα τα χρόνια των σπουδών, όπως επίσης για την κατανόηση και την ενθάρρυνση κατά την διάρκεια εκπόνησης της μεταπτυχιακής διατριβής.

Περίληψη

Σκοπός της παρούσας μεταπτυχιακής διατριβής είναι η ανάπτυξη και παρουσίαση μιας εφαρμογής παραγγελιοληψίας σε λειτουργικό σύστημα Android για χρήση σε μαγαζιά εστίασης. Μέσω της εφαρμογής ο έμπορος θα μπορεί να πάρει παραγγελία από τον πελάτη και να την στείλει στην κουζίνα/bar για να ετοιμαστεί με την χρήση της έξυπνης συσκευής του (smart phone, tablet, Android POS, κτλπ).

Βασικά στοιχεία αυτής της εργασίας είναι η ανάπτυξη της εφαρμογής σε γλώσσα προγραμματισμού Kotlin που είναι η προτεινόμενη από την Google για χρήση σε λειτουργικά συστήματα Android, η δημιουργία μιας φιλικής προς στον χρήστη εφαρμογής μέσω διασαφών που να είναι ευχάριστη και εύκολη στον χρήστη της εφαρμογής και η χρήση τεχνολογιών σύννεφου (cloud) όπως είναι το firebase.

Στην παρούσα διπλωματική αναλύουμε τα στάδια για την δημιουργία της εφαρμογής παραγγελιοληψίας όπως την μελέτη των λειτουργικών απαιτήσεων, την αρχιτεκτονική του συστήματος, την υλοποίηση της εφαρμογής και κλείνουμε με την παρουσίαση συμπερασμάτων και των μελλοντικών προοπτικών της.

Abstract

The purpose of this master's thesis is the development and presentation of an ordering application in the Android operating system for use in restaurants. From the application, the merchant will be able to take an order from the customer and send it to the kitchen/bar to be prepared using his smart device (smart phone, tablet, Android POS, etc).

Key elements of this work are the development of the application in Kotlin programming language which is the one recommended by Google for use in Android operating systems, the creation of a user-friendly application that is pleasant and easy for the user of the application, and the use of cloud technologies such as firebase.

In this thesis we analyze the stages for the creation of the ordering application such as the study of the functional requirements, the system architecture, the implementation of the application and the presentation of conclusions and its future prospects.

1.Εισαγωγή

Εταιρίες εκμεταλλεούμενες την ραγδαία αύξηση των κινητών συσκευών, των έξυπνων συσκευών, υπολογιστή νέφους (cloud computing) και του διαδικτύου έχουν στραφεί στην δημιουργία εφαρμογών για επαγγελματική χρήση. Κάποια από τα επαγγέλματα αυτά είναι οι επαγγελματίες οδηγεί που χρησιμοποιούν εφαρμογές GPS στις έξυπνες συσκευές που διαθέτουν (smart phones, tablet, etc) και πολύ άλλοι κλάδοι και εργαζόμενοι.

Όσο αναφορά τα μαγαζιά εστίασης τα οποία θα αναφέρουμε, τα τελευταία χρονιά έχουν δημιουργική εφαρμογές για την παραγγελιοληψίας. Πολλές από τις εφαρμογές αυτές δεν είναι φιλικές προς των χρήστη, είτε αποτελούνται από δυο εφαρμογές μια για τον σερβιτόρο και άλλη για την κουζίνα/bar, είτε έχουν ελλείψεις όπως είναι το να κρατάνε inventory ή εύκολος και γρήγορος τρόπος του να προσθέτεις και να αφαιρείς νέους κωδικούς στο μενού (όπως είναι τα πιάτα ημέρας για παράδειγμα).

Στην παρούσα μεταπτυχιακή διατριβή θα παρουσιαστεί η ανάπτυξη της εφαρμογής παραγγελιοληψίας σε Android λειτουργικό.

Στο παρόν κεφάλαιο θα αναφέρουμε τον σκοπό εκπόνησης της διπλωματικής εργασίας, το χώρο προβλήματος αλλά και παρόμοιες εφαρμογές που υπάρχουν σήμερα στην Ελληνική αγορά.

1.1 Χώρος του Προβλήματος

Τα προηγούμενα χρόνια που δεν υπήρχε η τεχνολογική ανάπτυξη που υπάρχει τώρα και δεν ήταν προσβάσιμη για όλους τα μαγαζιά εστίασης δεν διέθεταν κάποιο πρόγραμμα παραγγελιοληψίας και ο σερβιτόρος έπρεπε να γράψει σε χαρτί την παραγγελία που θα του έδινε ο πελάτης και να πήγαινε στην κουζίνα/bar για να την παράδοση στον συνάδελφο του για να την ετοιμάσει.

Με την εξέλιξη των έξυπνων φορητών συσκευών δημιουργήθηκαν οι κατάλληλες εφαρμογές σε περιβάλλον Android/iOS όπως τις ξέρουμε σήμερα. Μέσω των εφαρμογών αυτών πλέον μπορεί ο σερβιτόρος μπορεί να στέλνει άμεσα την παραγγελία στην κουζίνα/bar γλιτώνοντας αρκετό χρόνο από το να το έκανε χωρίς την εφαρμογή, θα μπορεί να ενημερώνεται άμεσα για την διαθεσιμότητα των προϊόντων που έχει στον κατάλογο και να κρατάει της ανοιχτές παραγγελίες οργανωμένες στην συσκευή του και να ξέρει τι έχει παραγγείλει κάθε τραπέζι.

1.2 Στόχοι εργασίας

Αντικείμενο της παρούσας μεταπτυχιακής διατριβής αποτελεί η μελέτη και ανάπτυξη της εφαρμογής παραγγελιοληψίας για μαγαζιά εστίασης με λειτουργικό σύστημα Android με τους πλέον ενδεδειγμένους τρόπους σχεδίασης και συγχρόνων λειτουργιών. Και στόχος να βελτιώσουμε σε σχέση με τις υπάρχουσες. Το βασικό χαρακτηριστικό της εφαρμογής είναι να είναι εύχρηστη και φιλική στο τελικό χρήστη, να κάνει χρήση των κατάλληλων πόρων ώστε να μην επιβαρύνει τις επιδόσεις της συσκευής, να προσαρμόζεται σε κάθε τύπου συσκευή (Smart Phone/Tablet/Android POS) ανάλογα με τις διαστάσεις της οθόνης, να παρέχει τη κατάλληλη πληροφόρηση και να αξιοποιούνται νέες τεχνολογίες.

1.3 Τι είναι μια εφαρμογή παραγγελιοληψίας

Υπάρχουν διάφοροι τύποι εφαρμογών παραγγελιοληψίας που μπορούν να χρησιμοποιήσουν τα εστιατόρια για να εξυπηρετήσουν τους πελάτες τους.

Μερικά παραδείγματα περιλαμβάνουν:

- 1 Συστήματα διαδικτυακών παραγγελιών: Αυτά επιτρέπουν στους πελάτες να κάνουν παραγγελίες και να πραγματοποιούν πληρωμές μέσω του ιστότοπου (site).
- 2 Εφαρμογές παραγγελιών για κινητά (delivery app): Πρόκειται για εφαρμογές που οι πελάτες μπορούν να κατεβάσουν στα smartphone τους και να τις χρησιμοποιήσουν για να κάνουν παραγγελίες και να πραγματοποιήσουν πληρωμές απευθείας μέσω της εφαρμογής (eFood, Wolt, Box κ.λπ).
- 3 Συστήματα παραγγελιών περιπτέρου (Kiosk): Πρόκειται για τερματικά παραγγελιών αυτοεξυπηρέτησης που μπορούν να χρησιμοποιήσουν οι πελάτες για να υποβάλουν τις παραγγελίες τους και να πραγματοποιήσουν πληρωμές (McDonald).

- 4 Συστήματα παραγγελίας τραπέζιων: Πρόκειται για συστήματα που επιτρέπουν στους πελάτες να κάνουν παραγγελίες και να πραγματοποιούν πληρωμές απευθείας από το τραπέζι τους, συνήθως μέσω tablet ή άλλης συσκευής.
- 5 Συστήματα φωνητικών παραγγελιών: Πρόκειται για συστήματα που επιτρέπουν στους πελάτες να κάνουν παραγγελίες και να πραγματοποιούν πληρωμές μέσω φωνητικών εντολών.
- 6 Συστήματα παραγγελιών Chatbot: Πρόκειται για συστήματα που επιτρέπουν στους πελάτες να κάνουν παραγγελίες και να πραγματοποιούν πληρωμές μέσω αλληλεπίδρασης chatbot.

Αυτά είναι μερικά παραδείγματα και υπάρχουν διαφορετικοί τρόποι για να παραγγείλετε φαγητό από ένα εστιατόριο, αλλά οι πιο συνηθισμένοι είναι τα συστήματα παραγγελιών μέσω διαδικτύου και μέσω κινητού τηλεφώνου, τα οποία παρέχουν ευκολία και ευκολία στη χρήση στους πελάτες.

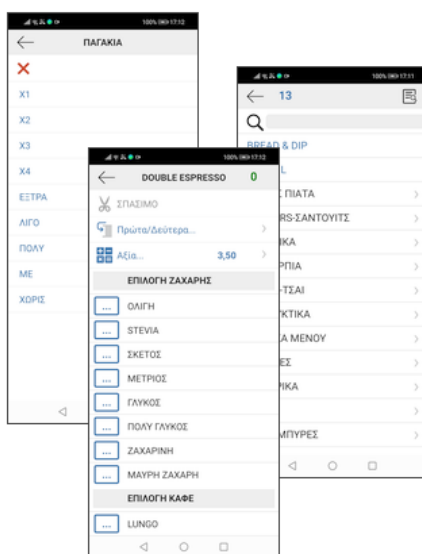
1.4 Παρόμοιες Εφαρμογές

Στο κεφάλαιο αυτό θα παρουσιαστούν αντίστοιχες εφαρμογές που κυκλοφορούν στην Ελληνική αγορά και θα αναφέρουμε τα χαρακτηριστικά τους.

1.4.1 Prognosi

Η Prognosi ιδρύθηκε το 2001 και από τότε προσφέρει ολοκληρωμένες λύσεις μηχανοργάνωσης για χώρους εστίασης και λιανικής.

Το Prognosi mobile app είναι πλήρως παραμετροποιήσιμο μενού σε δενδροειδή μορφή με βάθος και δυνατότητα διαμόρφωσης κάθε κουμπιού. Το μενού που θα εμφανίζεται μπορεί να διαφοροποιείται αυτόματα ανάλογα με τον τύπο παραγγελίας πχ όρθιοι ή τραπέζι, το τερματικό, την ημέρα ή και την ώρα αυτόματα.



Εικόνα 1: Prognosi APP

1.4.2 Inorder mobile

Η εφαρμογή Inorder mobile έχει υλοποιηθεί από την Wizcom. Η Wizcom από την δημιουργία της το 2003 δραστηριοποιείται στην ανάπτυξη λογισμικού και παροχής υπηρεσιών internet. Η δραστηριοποίηση της αυτή έχει διευρύνει την τεχνολογική της εμπειρία μέσα από την χρήση σύγχρονων πλατφόρμων λογισμικού, στην παροχή λύσεων αυτοματοποίησης διαδικασιών για την ανάπτυξη λογισμικού. Με γνώμονα την κάλυψη όλων των σταδίων ενός έργου που αφορούν την ανάλυση των απαιτήσεων, τον σχεδιασμό, την ανάπτυξη, τον έλεγχο και την πλήρη υποστήριξη.

Το Inorder mobile app προσφέρει:

- Διαχείριση παραγγελιών
- Παρακολούθηση ταμείων και βαρδιών
- Πολλαπλές καταστάσεις τραπεζιών (ανοιχτό, κλειστό, αναμονή λογαριασμού, εξοφλημένο, σε κράτηση)
- Τραπέζια
- Μεταφορά τραπεζιού σε άλλο χώρο και πόστο
Εκτυπώσεις παρασκευαστηρίων



Εικόνα 2: Inorder Mobile APP

1.4.3 Cardlink APP

Το Cardlink APP ανήκει στην Cardlink η οποία δραστηριοποιείται στον τομέα των ηλεκτρονικών πληρωμών και ειδικότερα στην αποδοχή και διαχείριση συναλλαγών με κάρτες, τόσο στο φυσικό όσο και το ηλεκτρονικό εμπόριο. Από τον Σεπτέμβριο του 2021, η Cardlink ανήκει στη Worldline, ηγέτη στην ευρωπαϊκή αγορά πληρωμών και υπηρεσιών διαχείρισης συναλλαγών και στην πρώτη τετράδα των παγκόσμιων παικτών.

Η Cardlink έχει επίσης λανσάρει μία σειρά από νέα προϊόντα όπως είναι το Cardlink e-grow για self-service δημιουργία e-shop και website με έτοιμα templates και το Cardlink smart που προσφέρει στις επιχειρήσεις τη δυνατότητα έκδοσης ηλεκτρονικής απόδειξης με απευθείας διασύνδεση με την Α.Α.Δ.Ε. και την πλατφόρμα myDATA και η web εφαρμογή Cardlink pulse, η οποία προσφέρει τη δυνατότητα στις επιχειρήσεις να αξιοποιούν τα δεδομένα που ήδη έχουν και να τα μετατρέπουν σε ευκαιρίες προκειμένου να βελτιώσουν την απόδοση της επιχείρησής τους.

Το Cardlink APP Είναι μια cloud εφαρμογή ασύρματης παραγγελιοληψίας για τον χώρο της εστίασης και συγκεκριμένα για εστιατόριο, καφετέρια, μπαρ ή αν προσφέρετε υπηρεσίες take-away και delivery. Το Cardlink APP διαθέτει έλεγχο στη λήψη και διαχείριση παραγγελιών, με real-time συντονισμό και οργάνωση των παραγγελιών, του προσωπικού και της αποθήκης.



Εικόνα 3: Cardlink APP

1.5 Σύγκριση Εφαρμογών.

Θα προσπαθήσουμε να συγκρίνουμε τις εφαρμογές που παρουσιάστηκαν. Αρχικά οι εφαρμογές ξεχωρίζουν λόγω των περισσότερων λειτουργιών όπως είναι η μεταφορά τραπέζιων, σύνδεση με εκτυπωτικές συσκευές και πληρωμή ξεχωριστά του λογαριασμού. Στα αρνητικά είναι η έλλειψη μοντέρνας σχεδίασης και του εύχρηστου περιβάλλοντος χρήσης.

Στην σχεδίαση μας ακολουθήσαμε τους κανόνες material design της Google με αποτέλεσμα να φτιάξουμε ένα μοντέρνο και εύχρηστο περιβάλλον για τον τελικό χρήστη. Έχουμε εφαρμόσει ένα έξυπνο σχεδιασμό που ανάλογα με τα credential του χρήστη (αν είναι σερβιτόρος ή δουλεύει στην κουζίνα/μπουφέ) του εμφανίζει διαφορετικό μενού και δυνατότητες, με αυτόν τον τρόπο προστατεύουμε τον χρήστη από λάθη και του κάνουμε ποιο εύχρηστη την εφαρμογή.

Πολύ σημαντικό είναι ο τρόπος που έχουμε σχεδιάσει την βάση στο firebase που μας επιτρέπει μελλοντικά να αναπτύξουμε λειτουργία που να αξιοποιεί τα δεδομένα που αποθηκεύει και να φτιάχνει στατιστικά που θα τα χρησιμοποιεί η επίχρηση.

2. Φορητές Έξυπνες Συσκευές

2.1 Έξυπνα Τηλέφωνα (Smartphones)

Το smartphone είναι ένας τύπος κινητού τηλεφώνου που έχει προηγμένες δυνατότητες πέρα από ένα παραδοσιακό κινητό τηλέφωνο. Τα smartphone λειτουργούν συνήθως σε λειτουργικό σύστημα κινητής τηλεφωνίας και προσφέρουν ένα ευρύ φάσμα λειτουργιών και λειτουργιών, όπως:

- **Συνδεσιμότητα στο διαδίκτυο:** Τα περισσότερα smartphone διαθέτουν ενσωματωμένη ασύρματη σύνδεση στο διαδίκτυο, επιτρέποντας στους χρήστες να περιηγούνται στον ιστό, να ελέγχουν email και να έχουν πρόσβαση στα μέσα κοινωνικής δικτύωσης.
- **Δυνατότητες πολυμέσων:** Τα smartphone διαθέτουν μεγάλες οθόνες υψηλής ανάλυσης και ισχυρούς επεξεργαστές που τους επιτρέπουν να παίζουν βίντεο, μουσική και παιχνίδια. Συχνά διαθέτουν κάμερες για λήψη φωτογραφιών και βίντεο, ενώ ορισμένα μοντέλα διαθέτουν ακόμη και πολλές κάμερες για προηγμένη φωτογραφία.
- **Εφαρμογές:** Τα smartphone επιτρέπουν στους χρήστες να κατεβάζουν και να χρησιμοποιούν ένα ευρύ φάσμα εφαρμογών ή «εφαρμογών» από καταστήματα εφαρμογών. Αυτές οι εφαρμογές μπορούν να χρησιμοποιηθούν για πλοήγηση, παραγωγικότητα, ψυχαγωγία και πολλούς άλλους σκοπούς.
- **Προσωπικοί βοηθοί:** Πολλά smartphone διαθέτουν εικονικούς προσωπικούς βοηθούς, όπως Siri, Google Assistant ή Alexa, που επιτρέπουν στους χρήστες να αλληλεπιδρούν με το τηλέφωνό τους χρησιμοποιώντας φωνητικές εντολές.
- **Πληρωμή μέσω κινητού:** Ορισμένα smartphone διαθέτουν ενσωματωμένες δυνατότητες πληρωμής μέσω κινητού τηλεφώνου, που επιτρέπουν στους χρήστες να πραγματοποιούν πληρωμές χρησιμοποιώντας το τηλέφωνό τους.
- **Προηγμένοι αισθητήρες:** Πολλά smartphone διαθέτουν μια ποικιλία προηγμένων αισθητήρων, όπως επιταχυνσιόμετρα, γυροσκόπια και GPS, που επιτρέπουν νέες και βελτιωμένες εμπειρίες χρήστη.

Συνολικά, τα smartphone είναι ισχυρές και ευέλικτες συσκευές που επιτρέπουν στους χρήστες να παραμένουν συνδεδεμένοι, να είναι παραγωγικοί και να διασκεδάζουν εν κινήσει.

Η ιστορία των smartphone χρονολογείται από τις αρχές της δεκαετίας του 1990, όταν αναπτύχθηκαν τα πρώτα πρωτότυπα αυτού που σήμερα γνωρίζουμε ως smartphone. Ακολουθεί μια σύντομη επισκόπηση των βασικών ορόσημων στην εξέλιξη των smartphone:

1993 - IBM Simon: Το IBM Simon ήταν το πρώτο smartphone που αναπτύχθηκε, αν και δεν κυκλοφορούσε ως smartphone εκείνη την εποχή. Ήταν μια μεγάλη, βαρετή συσκευή με οθόνη αφής και βασικές δυνατότητες επικοινωνίας και διαχείρισης προσωπικών πληροφοριών.



Εικόνα 4: IBM Simon mobile

1997 - Ericsson T28: Το T28 της Ericsson ήταν ένα από τα πρώτα εμπορικά διαθέσιμα smartphone. Ήταν ένα μεγάλο τηλέφωνο σε στυλ τούβλου με μονόχρωμη οθόνη, βασικές δυνατότητες περιήγησης στο web και υποστήριξη για email και φαξ.



Εικόνα 5: Ericsson T28

1999 - Blackberry 850: Το Blackberry 850 ήταν ένα από τα πρώτα smartphone που κέρδισαν ευρεία δημοτικότητα. Σχεδιάστηκε για επαγγελματίες χρήστες και διέθετε μια μικρή, μονόχρωμη οθόνη, ένα φυσικό πληκτρολόγιο και δυνατότητες email.



Εικόνα 6: BlackBerry 850

2007 - Apple iPhone: Το iPhone, που κυκλοφόρησε από την Apple το 2007, άλλαξε το παιχνίδι για την αγορά των smartphones. Με τον κομψό σχεδιασμό, τη μεγάλη οθόνη αφής και τις προηγμένες λειτουργίες του, έγινε γρήγορα ένα από τα πιο δημοφιλή smartphones στην αγορά.



Εικόνα 7: Apple iPhone 2007 (first iPhone)

2008 - Google Android: Το 2008, η Google κυκλοφόρησε το Android, ένα νέο λειτουργικό σύστημα για κινητά ανοιχτού κώδικα. Αυτό άνοιξε το δρόμο για μια νέα γενιά smartphones από διάφορους κατασκευαστές, συμπεριλαμβανομένων των Samsung, HTC και LG.



Εικόνα 8: Google Android Phone

Στα χρόνια από την κυκλοφορία του πρώτου iPhone, τα smartphones συνέχισαν να εξελίσσονται και να βελτιώνονται. Έχουν γίνει πιο ισχυρά, πιο συμπαγή και πιο ευέλικτα, με νέες δυνατότητες όπως κάμερες υψηλής ανάλυσης, προηγμένους αισθητήρες και βοηθούς φωνής που προστίθενται. Σήμερα, τα smartphones είναι ένα πανταχού παρόν μέρος της ζωής μας και συνεχίζουν να

εξελισσονται και να βελτιώνονται καθώς προχωρά η τεχνολογία. Είναι απαραίτητα για να παραμείνουμε συνδεδεμένοι, να διεξάγουμε επιχειρήσεις και να έχουμε πρόσβαση σε πληροφορίες και είναι πιθανό να διαδραματίσουν ακόμη πιο σημαντικό ρόλο στη ζωή μας τα επόμενα χρόνια.

2.2 Υπολογιστής ταμπλέτα (Tablet)

Το tablet είναι ένας τύπος φορητής υπολογιστικής συσκευής που έχει συνήθως μεγάλη οθόνη αφής και έχει σχεδιαστεί για χρήση με τα δάχτυλά ή με γραφίδα. Τα tablet είναι παρόμοια με τα smartphone ως προς τον παράγοντα μορφής και τη διεπαφή χρήστη, αλλά συνήθως έχουν μεγαλύτερες οθόνες και δεν μπορούν να πραγματοποιούν τηλεφωνικές κλήσεις.

Τα tablet είναι φορητά και μπορούν να χρησιμοποιηθούν για μια ποικιλία εργασιών, όπως περιήγηση στο διαδίκτυο, ανάγνωση ηλεκτρονικών βιβλίων, παρακολούθηση βίντεο, παιχνίδια και εργασίες παραγωγικότητας, όπως η επεξεργασία κειμένου και η διαχείριση υπολογιστικών φύλλων.

Γενικά τρέχουν σε λειτουργικά συστήματα όπως το Android, το iOS ή τα Windows και προσφέρουν πρόσβαση σε ένα ευρύ φάσμα εφαρμογών ή "εφαρμογών", που μπορούν να ληφθούν από τα καταστήματα εφαρμογών. Ορισμένα tablet διαθέτουν επίσης ένα φυσικό πληκτρολόγιο ή θήκη πληκτρολογίου που μπορεί να προσαρτηθεί στη συσκευή, διευκολύνοντας την πληκτρολόγηση για μεγαλύτερα χρονικά διαστήματα.

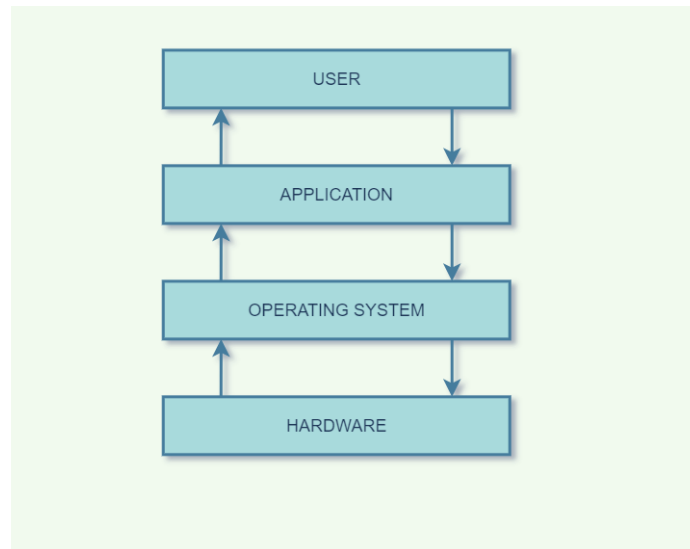
Τα tablet έχουν γίνει δημοφιλή επειδή προσφέρουν μια πιο φορητή και βολική εναλλακτική λύση σε σχέση με τους παραδοσιακούς φορητούς υπολογιστές ή τους επιτραπέζιους υπολογιστές. Είναι ιδιαίτερα χρήσιμα για εργασίες που απαιτούν μεγαλύτερη οθόνη ή μεγαλύτερη φορητότητα από ό,τι μπορεί να προσφέρει ένα smartphone, αλλά δεν απαιτούν την πλήρη ισχύ και τη λειτουργικότητα ενός παραδοσιακού υπολογιστή.



Εικόνα 9: Tablets

2.3 Λειτουργικά Συστήματα για Κινητά (Mobile OS)

Τα λειτουργικά συστήματα για κινητά (Mobile OS) είναι πλατφόρμες λογισμικού που τρέχουν σε κινητές συσκευές όπως smartphone, tablet και smartwatches. Διαχειρίζονται τους πόρους υλικού της συσκευής και παρέχουν μια διεπαφή για την αλληλεπίδραση των χρηστών με τη συσκευή και τις εφαρμογές της.



Εικόνα 10: Operating System

Μερικά από τα πιο δημοφιλή λειτουργικά συστήματα για κινητά είναι:

Android: Είναι ένα λειτουργικό σύστημα για κινητά που αναπτύχθηκε από την Google. Κυκλοφόρησε για πρώτη φορά το 2008 και αυτή τη στιγμή είναι το πιο ευρέως χρησιμοποιούμενο λειτουργικό σύστημα κινητής τηλεφωνίας στον κόσμο.

Το Android είναι γνωστό για την ευελιξία, τις επιλογές προσαρμογής και τη φύση ανοιχτού κώδικα, που επιτρέπει στους προγραμματιστές να τροποποιούν και να βελτιώνουν το λογισμικό. Μπορεί να χρησιμοποιηθεί σε ένα ευρύ φάσμα συσκευών από πολλούς κατασκευαστές, όπως smartphone, tablet, smartwatches και άλλες συσκευές Internet of Things (IoT).

Το λειτουργικό σύστημα Android ενημερώνεται τακτικά, με νέες δυνατότητες και βελτιώσεις ασφάλειας που προστίθενται με κάθε κυκλοφορία. Το Google Play Store είναι μια κεντρική τοποθεσία όπου οι χρήστες μπορούν να κάνουν λήψη και εγκατάσταση εφαρμογών τρίτων και η Google έχει εφαρμόσει διάφορα μέτρα ασφαλείας για να διασφαλίσει την ασφάλεια των χρηστών της.

iOS: Είναι ένα λειτουργικό σύστημα για κινητά που αναπτύχθηκε από την Apple Inc. Παρουσιάστηκε για πρώτη φορά το 2007 για το iPhone και έκτοτε χρησιμοποιείται σε άλλες συσκευές της Apple όπως το iPad, το iPod Touch και το Apple Watch.

Το iOS είναι γνωστό για τη φιλική προς τον χρήστη διεπαφή και το αυστηρά ελεγχόμενο οικοσύστημά του, που επιτρέπει στην Apple να διασφαλίζει υψηλό βαθμό ποιότητας και ασφάλειας στις συσκευές και το λογισμικό της. Το λειτουργικό σύστημα ενημερώνεται τακτικά, με νέες δυνατότητες και βελτιώσεις που προστίθενται σε κάθε έκδοση.

Το iOS ενσωματώνεται επίσης στενά με άλλες υπηρεσίες της Apple, όπως το iCloud, το iMessage και το App Store. Το App Store είναι μια κεντρική τοποθεσία όπου οι χρήστες μπορούν να

κάνουν λήψη και εγκατάσταση εφαρμογών τρίτων και η Apple επιμελείται προσεκτικά τις εφαρμογές που επιτρέπονται στην πλατφόρμα για να εξασφαλίσει ποιότητα και ασφάλεια.

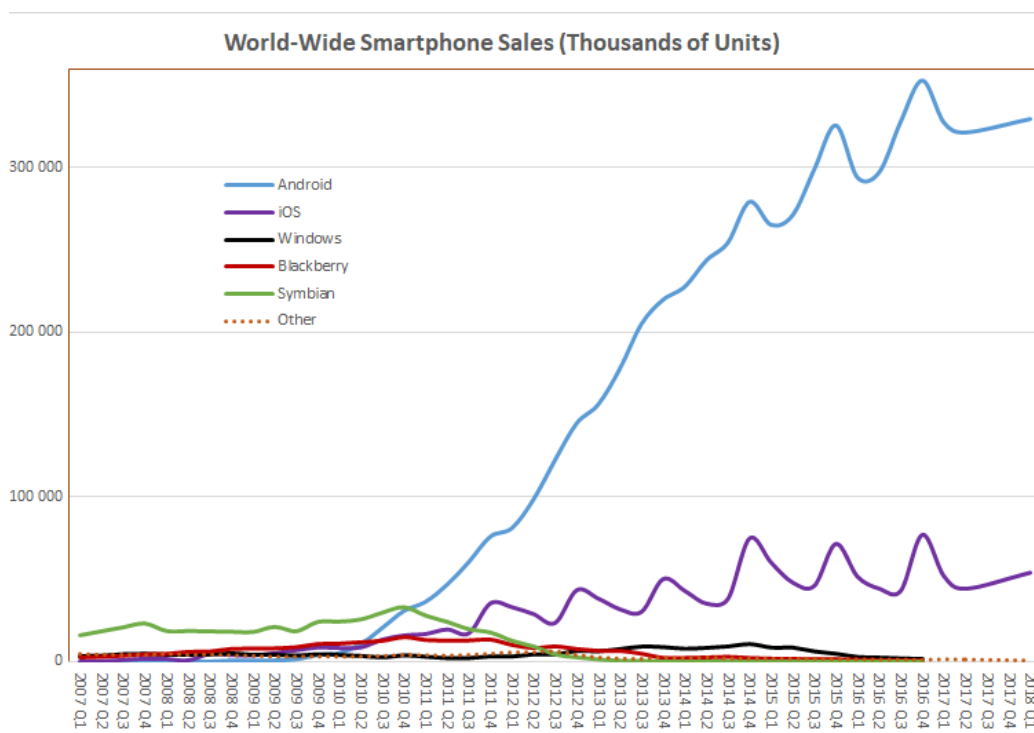
Windows Phone: Ήταν ένα λειτουργικό σύστημα για κινητά που αναπτύχθηκε από τη Microsoft. Παρουσιάστηκε το 2010 ως διάδοχος των Windows Mobile και σχεδιάστηκε για να ανταγωνιστεί το iOS και το Android στην αγορά των smartphone.

Το Windows Phone διέθετε μια χαρακτηριστική διεπαφή χρήστη "Metro", η οποία χρησιμοποιούσε μεγάλα, πολύχρωμα πλακίδια για να αναπαραστήσει εφαρμογές και άλλες λειτουργίες. Επίσης ενσωματώθηκε με άλλες υπηρεσίες της Microsoft, όπως το Office, το Xbox και το OneDrive.

Ενώ το Windows Phone έλαβε θετικές κριτικές για το σχεδιασμό και τη διεπαφή χρήστη του, δυσκολεύτηκε να κερδίσει την έλξη στην ανταγωνιστική αγορά των smartphone. Το μερίδιο αγοράς της παρέμεινε σχετικά μικρό, ιδιαίτερα εκτός των Ηνωμένων Πολιτειών, και αντιμετώπισε προκλήσεις στην προσέλκυση προγραμματιστών εφαρμογών στην πλατφόρμα.

Harmony OS: Είναι ένα λειτουργικό σύστημα για κινητά που αναπτύχθηκε από την Huawei, μια κινεζική εταιρεία τεχνολογίας. Ανακοινώθηκε για πρώτη φορά το 2019 ως εναλλακτική λύση στο λειτουργικό σύστημα Android της Google, το οποίο η Huawei δεν ήταν πλέον σε θέση να χρησιμοποιήσει λόγω των εμπορικών περιορισμών που επέβαλε η κυβέρνηση των Ηνωμένων Πολιτειών.



Το Harmony OS έχει σχεδιαστεί για να είναι ένα λειτουργικό σύστημα πολλαπλών πλατφορμών που μπορεί να χρησιμοποιηθεί σε διάφορες συσκευές, όπως smartphone, tablet, smart TV, wearables και άλλες συσκευές Internet of Things (IoT). Βασίζεται σε μια αρχιτεκτονική μικροπυρήνα, η οποία έχει σκοπό να βελτιώσει την ασφάλεια, την αξιοπιστία και την απόδοση του συστήματος.



Διαγράμματα 1: Μερίδιο αγοράς MobileOS Q1 2018

2.4 Λειτουργικά Συστήματα Android και iOS

Τα επικρατέστερα λειτουργικά συστήματα για έξυπνες κινητές συσκευές είναι το Android της Google και το iOS της Apple. Το Android που βρίσκετε στις περισσότερες μάρκες smartphones και το iOS που βρίσκετε στα iPhones της εταιρείας Apple.

Comparison Between Android & iOS		
Operating System		
Base OS	Linux	UNIX and OSX
Customizable	Yes	Limited, Needs Jailbreak for extensive use
Apps	1,000,000+ apps on the google	1,000,000+ apps in the iTunes store
Open Source	Yes	No
Supported Devices	Many devices like Oppo, Samsung, Vivo, MI, etc.	Only devices developed by Apple, like the iPhone, iPad and iPod Touch
Languages Suported	32	34
Default Net Browser	Google Chrome	Safari

Εικόνα 11: Λειτουργικά Συστήματα Android και iOS

Σε αυτήν την ενότητα θα αναφερθούμε στις βασικές διαφορές αυτών των δύο επικρατέστερων λειτουργικών συστημάτων καθώς και στα χαρακτηριστικά τους, αλλά πρώτα θα αναφερθούμε στο τι είναι λειτουργικό σύστημα μιας έξυπνης συσκευής. Ως λειτουργικό σύστημα αναφέρουμε την βάση για τις διάφορες λειτουργίες των έξυπνων συσκευών. Κάποιες από αυτές τις λειτουργίες είναι, η εμφάνιση των στοιχείων μέσα σε ένα έξυπνο κινητό, οι ρυθμίσεις του και ο τρόπος που κατεβάζει κανείς διάφορες εφαρμογές στα smartphones και στα iPhones της Apple.

Ο καθημερινός καταναλωτής πριν αγοράσει μια συσκευή smartphone που επιθυμεί, καλό θα είναι να δει κάποια χαρακτηριστικά των λειτουργικών συστημάτων που θα θέλει να αγοράσει, δηλαδή των Android και των iOS, για να έχει ολοκληρωμένη εικόνα και άποψη για την τεχνολογία της συσκευής που θα θέλει να αγοράσει. Ας τα πάρουμε όλα από την αρχή.

Το λειτουργικό σύστημα Android της Google είναι ένα λειτουργικό σύστημα ανοιχτού λογισμικού (open source). Δηλαδή, είναι ένας πηγαίος κώδικας που διατίθεται στην κοινότητα για να την αξιοποιήσει και να συνεισφέρει στην ανάπτυξη του. Επίσης, το λογισμικό ανοιχτού κώδικα επιτρέπει στον χρήστη να μπει στον πηγαίο κώδικα και να το τροποποιήσει στις δικές του ανάγκες.

Ο πυρήνας του λειτουργικού συστήματος Android είναι το λειτουργικό Linux. Αρχικά ξεκίνησε η ανάπτυξη του από την Google και μετά από την κοινότητα του Open Handset Alliance. Παρουσιάστηκε το 2007. Ως γλώσσα προγραμματισμού χαρακτηρίζετε η Java και η Kotlin.

Το Android δημιουργήθηκε στην πόλη Πόλο Άλτο της Καλιφόρνιας στην Αμερική το 2003 από τους Andy Rubin, Rich Miner και Nik Sears. Το αρχικό λογότυπο του Android από το 2007 έως και το 2014 ήταν σκέτο η λέξη Android και από το 2015 μέχρι και σήμερα έχει το λογότυπο του πράσινου robot. Από το 2008 είχε διατυπωθεί να γίνουν κάποιες ενημερώσεις του λογισμικού.

Αυτή η ιδέα πραγματοποιήθηκε το 2013 και ως και σήμερα έχουν αλλάξει πολλά στοιχεία τόσο στην ασφάλεια όσο και στην εμφάνιση τους.

Το iOS είναι ένα λειτουργικό σύστημα που υπάρχει μόνο στις συσκευές των iPhones της Apple. Πρώτο εμφανίστηκε το 2007 στις Ηνωμένες Πολιτείες της Αμερικής στην παρουσίασή του πρώτου iPhone. Ως γλώσσα προγραμματισμού του iOS, χαρακτηρίζετε η αντικειμενοστραφής γλώσσα Objective-C και Swift.

Το iOS σε αντίθεση με το Android είναι λειτουργικό σύστημα κλειστού κώδικα. Αυτό έχει σαν αποτέλεσμα να μην μπορεί κάποιος εκτός την Apple να το εξετάσει, ούτε και να μπορεί να πειράξει κάποιος τον πηγαίο κώδικα του λειτουργικού.

2.4.1 Τα Αρνητικά του Λειτουργικού Android

Κάποιες αρνητικές πτυχές του λειτουργικού συστήματος του Android είναι:

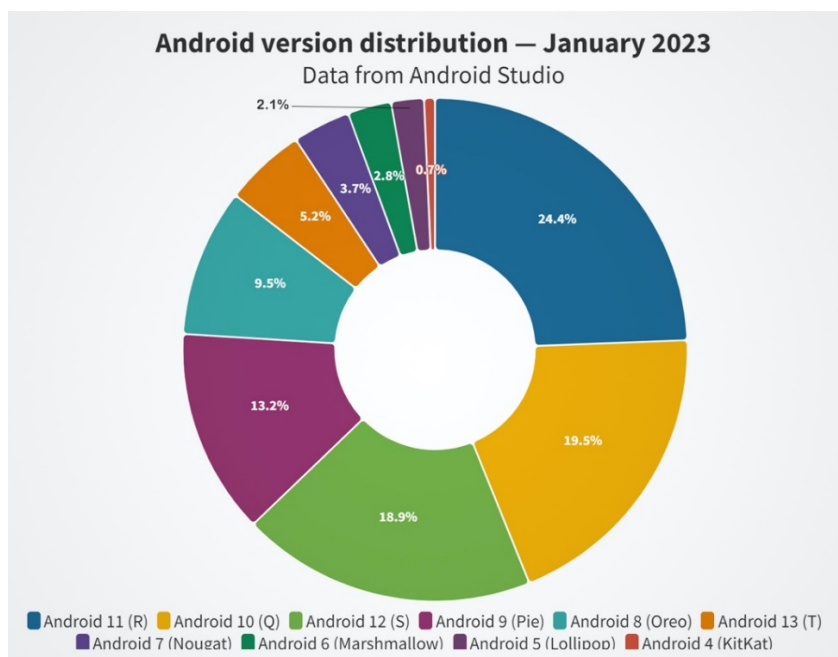
Fragmentation (Κατακερματισμός): Λόγω της φύσης ανοιχτού κώδικα του Android, μπορεί να υπάρξει σημαντικός κατακερματισμός στο οικοσύστημα, με διαφορετικούς κατασκευαστές και παρόχους να χρησιμοποιούν διαφορετικές εκδόσεις Android με διαφορετικούς βαθμούς προσαρμογής. Αυτό μπορεί να οδηγήσει σε ασυνέπεια στην εμπειρία χρήστη και καθυστερήσεις στις ενημερώσεις λογισμικού.

Security: Ενώ το Android έχει κάνει σημαντικά βήματα στη βελτίωση της ασφάλειάς του, εξακολουθεί να θεωρείται ότι είναι πιο ευάλωτο σε κακόβουλο λογισμικό και άλλες απειλές ασφαλείας σε σύγκριση με το iOS της Apple. Αυτό οφείλεται εν μέρει στον πιο ανοιχτό χαρακτήρα του οικοσυστήματος Android, το οποίο επιτρέπει μεγαλύτερη ευελιξία αλλά και το αφήνει πιο ευάλωτο σε κινδύνους ασφαλείας.

Bloatware: Ορισμένες συσκευές Android διαθέτουν προεγκατεστημένες εφαρμογές και υπηρεσίες από τον κατασκευαστή ή τον πάροχο, που συνήθως αναφέρονται ως "bloatware". Αυτές οι εφαρμογές μπορούν να καταλαμβάνουν χώρο αποθήκευσης και μπορεί να μην είναι χρήσιμες για τον χρήστη, αλλά δεν μπορούν να αφαιρεθούν εύκολα.

Ποιότητα εφαρμογών: Ενώ το Google Play έχει μια πιο ανοιχτή προσέγγιση για τις υποβολές εφαρμογών σε σύγκριση με το App Store της Apple, αυτό μπορεί επίσης να οδηγήσει σε μεγαλύτερη ποικιλομορφία στην ποιότητα των διαθέσιμων εφαρμογών. Ορισμένες εφαρμογές μπορεί να είναι κακοσχεδιασμένες ή να περιέχουν κακόβουλο λογισμικό και μπορεί να είναι δύσκολο για τους χρήστες να διακρίνουν μεταξύ καλών και κακών εφαρμογών.

Διάρκεια ζωής μπαταρίας: Λόγω των διαφορετικών επιλογών υλικού που διατίθενται για συσκευές Android, η διάρκεια ζωής της μπαταρίας μπορεί να ποικίλλει σημαντικά ανάλογα με τη συσκευή και τον τρόπο χρήσης της. Ορισμένες συσκευές Android ενδέχεται να έχουν μικρότερη διάρκεια ζωής της μπαταρίας σε σύγκριση με τις συσκευές iOS της Apple.



Διαγράμματα 2: Εκδόσεις Android OS μέχρι Γενάρη του 2023

2.4.2 Τα Αρνητικά του Λειτουργικού iOS

Κάποιες αρνητικές πτυχές του λειτουργικού συστήματος του iOS είναι:

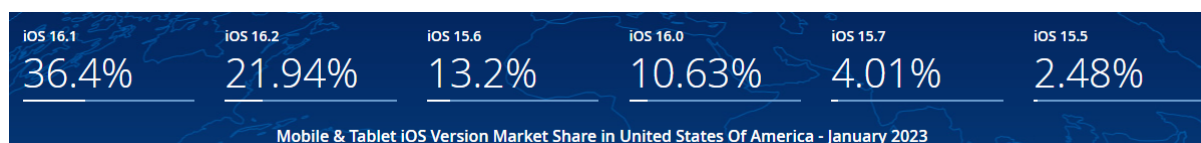
Περιορισμένες επιλογές προσαρμογής: Σε αντίθεση με το Android, το iOS δεν επιτρέπει μεγάλη προσαρμογή της διεπαφής και της αρχικής οθόνης. Οι χρήστες περιορίζονται στην τακτοποίηση εφαρμογών σε ένα πλέγμα και δεν μπορούν να αλλάξουν τις προεπιλεγμένες εφαρμογές για ορισμένες λειτουργίες με εφαρμογές τρίτων.

Χώρος αποθήκευσης: Τα iPhone δεν διαθέτουν επεκτάσιμες επιλογές αποθήκευσης. Οι χρήστες πρέπει να βασίζονται στη χωρητικότητα αποθήκευσης της συσκευής τους, η οποία μπορεί να είναι περιοριστική εάν έχουν μεγάλα αρχεία ή θέλουν να αποθηκεύσουν πολλά μέσα.

Υψηλές τιμές: Οι συσκευές Apple είναι γενικά πιο ακριβές από τις αντίστοιχες συσκευές Android, γεγονός που μπορεί να τις κάνει λιγότερο προσιτές σε ορισμένους καταναλωτές.

Περιορισμένη υποστήριξη για εφαρμογές τρίτων: Η Apple έχει αυστηρές οδηγίες για την ανάπτυξη και έγκριση εφαρμογών, γεγονός που μπορεί να δυσκολέψει ορισμένες εφαρμογές τρίτων να φτάσουν στο App Store. Επιπλέον, η Apple ενδέχεται να περιορίσει ορισμένες δυνατότητες ή λειτουργίες εφαρμογών τρίτων για να διατηρήσει τον έλεγχο της εμπειρίας χρήστη.

Κλειστό οικοσύστημα: Το κλειστό οικοσύστημα της Apple μπορεί να είναι περιοριστικό για χρήστες που θέλουν να χρησιμοποιήσουν συσκευές ή υπηρεσίες από άλλες εταιρείες. Για παράδειγμα, οι συσκευές Apple δεν υποστηρίζουν πολλές δημοφιλείς πλατφόρμες παιχνιδιών ή εικονικούς βοηθούς.



Εικόνα 12: Εκδόσεις iOS μέχρι Γενάρη του 2023

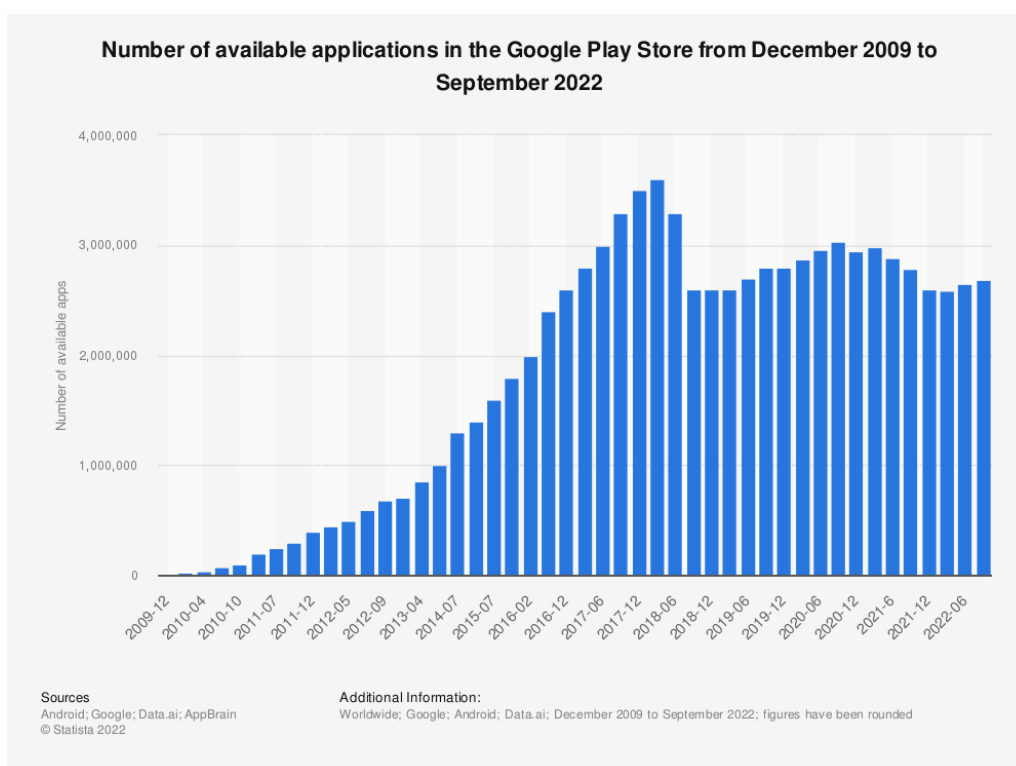
3. Εφαρμογές για το λειτουργικό Android.

Οι εφαρμογές Android είναι προγράμματα λογισμικού σχεδιασμένα να εκτελούνται σε λειτουργικά συστήματα Android, τα οποία χρησιμοποιούνται συνήθως σε κινητές συσκευές όπως smartphone και tablet. Οι εφαρμογές Android αναπτύσσονται χρησιμοποιώντας τη γλώσσα προγραμματισμού Java ή Kotlin και μπορούν να διανεμηθούν μέσω του Google Play Store ή άλλων καταστημάτων εφαρμογών τρίτων.

Υπάρχουν εκατομμύρια διαθέσιμες εφαρμογές Android, που καλύπτουν ένα ευρύ φάσμα κατηγοριών όπως παιχνίδια, μέσα κοινωνικής δικτύωσης, παραγωγικότητα, ψυχαγωγία και άλλα. Αυτές οι εφαρμογές μπορούν να ληφθούν και να εγκατασταθούν απευθείας σε μια συσκευή Android και πολλές από αυτές είναι δωρεάν για χρήση. Ορισμένες εφαρμογές προσφέρουν πρόσθετες λειτουργίες ή λειτουργικότητα μέσω αγορών εντός εφαρμογής ή συνδρομητικών υπηρεσιών.

Οι εφαρμογές Android μπορούν να έχουν πρόσβαση σε ένα ευρύ φάσμα λειτουργιών της συσκευής, όπως κάμερα, GPS και μικρόφωνο, κάτι που τους επιτρέπει να προσφέρουν πλούσιες και διαδραστικές εμπειρίες στους χρήστες. Οι προγραμματιστές μπορούν επίσης να χρησιμοποιήσουν διάφορα εργαλεία και πλαίσια για να δημιουργήσουν πολύπλοκες και εξελιγμένες εφαρμογές για συσκευές Android.

Συνολικά, οι εφαρμογές Android έχουν παίξει σημαντικό ρόλο στην ανάπτυξη των φορητών συσκευών, καθιστώντας τις πιο ευέλικτες, διασκεδαστικές και χρήσιμες για τους χρήστες.



Διαγράμματα 3: Διαθέσιμες εφαρμογές στο Google Play

3.1 Εργαλεία Ανάπτυξης Εφαρμογών σε Λειτουργικό Android.

Υπάρχουν διάφορα διαθέσιμα εργαλεία ανάπτυξης εφαρμογών Android που μπορούν να χρησιμοποιήσουν οι προγραμματιστές για τη δημιουργία εφαρμογών Android. Μερικά από τα πιο δημοφιλή εργαλεία ανάπτυξης εφαρμογών Android περιλαμβάνουν:

Android Studio: Το Android Studio είναι το επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για την ανάπτυξη εφαρμογών Android. Είναι ένα ισχυρό εργαλείο που προσφέρει μια σειρά από δυνατότητες όπως επεξεργασία κώδικα, εντοπισμό σφαλμάτων και ανάλυση απόδοσης.

Eclipse: Το Eclipse είναι ένα άλλο δημοφιλές IDE που χρησιμοποιείται για την ανάπτυξη εφαρμογών Android. Προσφέρει δυνατότητες όπως συμπλήρωση κώδικα, εντοπισμό σφαλμάτων και δοκιμή.

IntelliJ IDEA: Το IntelliJ IDEA είναι ένα δημοφιλές IDE για ανάπτυξη Java που προσφέρει επίσης υποστήριξη για ανάπτυξη εφαρμογών Android. Προσφέρει προηγμένα χαρακτηριστικά, όπως ανάλυση κώδικα και ανακατασκευή.

Visual Studio: Το Visual Studio είναι ένα δημοφιλές IDE για ανάπτυξη που βασίζεται σε Windows που προσφέρει επίσης υποστήριξη για ανάπτυξη εφαρμογών Android. Προσφέρει δυνατότητες όπως επεξεργασία κώδικα, εντοπισμό σφαλμάτων και δοκιμή.

Xamarin: Το Xamarin είναι ένα εργαλείο ανάπτυξης πολλαπλών πλατφορμών που επιτρέπει στους προγραμματιστές να γράφουν εφαρμογές Android χρησιμοποιώντας C# και .NET. Προσφέρει δυνατότητες όπως κοινή χρήση κώδικα και δοκιμές μεταξύ πλατφορμών.

Αυτά είναι μερικά μόνο παραδείγματα από τα πολλά εργαλεία ανάπτυξης εφαρμογών Android που είναι διαθέσιμα στους προγραμματιστές. Οι προγραμματιστές μπορούν να επιλέξουν το εργαλείο που ταιριάζει καλύτερα στις ανάγκες και τις προτιμήσεις τους, βάσει παραγόντων όπως η ευκολία χρήσης, οι δυνατότητες και το κόστος.



Εικόνα 13: Εργαλεία Ανάπτυξης Εφαρμογών σε Λειτουργικό Android

3.2 Κύριοι τύποι Εφαρμογών για Κινητές Συσκευές.

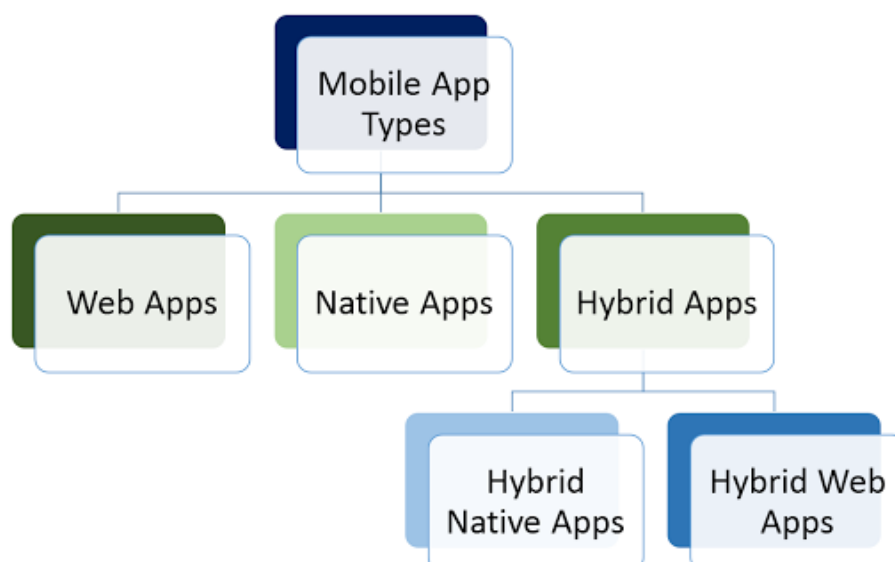
Υπάρχουν τρεις κύριοι τύποι εφαρμογών για κινητά και κάθε τύπος εφαρμογής για κινητά έχει τα δικά του πλεονεκτήματα και μειονεκτήματα και η επιλογή του τύπου που θα χρησιμοποιηθεί εξαρτάται από παράγοντες όπως οι απαιτήσεις, ο προϋπολογισμός και ο στόχος της εφαρμογής.

Αναλυτικά οι τύποι εφαρμογών είναι:

Native apps: Είναι εφαρμογές που έχουν αναπτυχθεί ειδικά για μια συγκεκριμένη πλατφόρμα για κινητά, όπως το iOS ή το Android. Αυτές οι εφαρμογές είναι γραμμένες στη γλώσσα προγραμματισμού της συγκεκριμένης πλατφόρμας, όπως το Swift για iOS ή Java για Android. Οι native εφαρμογές μπορούν να προσφέρουν υψηλότερο επίπεδο απόδοσης και καλύτερη εμπειρία χρήστη, επειδή είναι βελτιστοποιημένες για τη συγκεκριμένη πλατφόρμα και συσκευή.

Web apps: Οι εφαρμογές Ιστού (Web apps) είναι ουσιαστικά ιστότοποι βελτιστοποιημένοι για κινητά, στους οποίους έχετε πρόσβαση μέσω ενός προγράμματος περιήγησης για κινητά. Αυτές οι εφαρμογές είναι γραμμένες σε τεχνολογίες ανάπτυξης ιστού όπως HTML, CSS και JavaScript. Οι εφαρμογές Ιστού είναι ανεξάρτητες από την πλατφόρμα και μπορούν να εκτελεστούν σε οποιαδήποτε κινητή συσκευή με πρόγραμμα περιήγησης, αλλά ενδέχεται να μην προσφέρουν το ίδιο επίπεδο απόδοσης με τις εγγενείς εφαρμογές.

Hybrid apps: Οι υβριδικές εφαρμογές (Hybrid apps) είναι ένας συνδυασμός εγγενών εφαρμογών και εφαρμογών ιστού. Αυτές οι εφαρμογές έχουν κατασκευαστεί χρησιμοποιώντας τεχνολογίες ιστού, αλλά είναι τυλιγμένες σε ένα κέλυφος εγγενούς εφαρμογής που τους επιτρέπει να εγκατασταθούν και να εκτελούνται σε μια κινητή συσκευή όπως μια εγγενής εφαρμογή. Οι υβριδικές εφαρμογές (Hybrid apps) μπορούν να προσφέρουν τα πλεονεκτήματα τόσο των εγγενών εφαρμογών όσο και των εφαρμογών ιστού, όπως πρόσβαση σε λειτουργίες συσκευής και ανεξαρτησία πλατφόρμας.



Διαγράμματα 1: Τύποι Εφαρμογών

3.2.1 Τα Κριτήρια για να Επιλέξουμε τον τύπο Εφαρμογής.

Η επιλογή του σωστού τύπου εφαρμογής για κινητά για τις ανάγκες μας περιλαμβάνει διάφορους παράγοντες που θα πρέπει να εξετάσετε προσεκτικά.

Ακολουθούν ορισμένοι βασικοί παράγοντες που πρέπει να έχουμε κατά νου:

Εμπειρία χρήστη: Σκεφτείτε την εμπειρία χρήστη που θέλετε να παρέχετε στους χρήστες σας. Οι εγγενείς εφαρμογές προσφέρουν υψηλό επίπεδο απόδοσης και απρόσκοπτη εμπειρία χρήστη, αλλά είναι ακριβές στην ανάπτυξη. Οι εφαρμογές Ιστού προσφέρουν μεγαλύτερη ανεξαρτησία πλατφόρμας, αλλά ενδέχεται να μην παρέχουν το ίδιο επίπεδο εμπειρίας χρήστη.

Δυνατότητες: Εξετάστε τις δυνατότητες που θέλετε να προσφέρετε στην εφαρμογή σας. Οι εγγενείς εφαρμογές έχουν πρόσβαση στις λειτουργίες υλικού της συσκευής, όπως κάμερα, GPS και επιταχυνσιόμετρο, οι οποίες μπορεί να είναι σημαντικές για ορισμένους τύπους εφαρμογών. Οι εφαρμογές Ιστού εξακολουθούν να έχουν πρόσβαση σε ορισμένες λειτουργίες της συσκευής μέσω API, αλλά όχι τόσο όσο οι εγγενείς εφαρμογές.

Προϋπολογισμός: Εξετάστε τον προϋπολογισμό σας για την ανάπτυξη εφαρμογών. Οι εγγενείς εφαρμογές είναι γενικά πιο ακριβές στην ανάπτυξη από τις εφαρμογές ιστού, επειδή απαιτούν διαφορετικές βάσεις κώδικα για διαφορετικές πλατφόρμες. Οι υβριδικές εφαρμογές μπορούν να είναι μια οικονομικά αποδοτική λύση, εάν θέλετε να προσφέρετε μια εγγενή εμπειρία με μια ενιαία βάση κώδικα.

Target audience (Κοινό-στόχος): Λάβετε υπόψη το κοινό-στόχο σας και τις πλατφόρμες που χρησιμοποιούν. Εάν οι χρήστες σας βρίσκονται κυρίως σε μια συγκεκριμένη πλατφόρμα, μπορεί να έχει νόημα να αναπτύξετε μια εγγενή εφαρμογή για αυτήν την πλατφόρμα. Εάν οι χρήστες σας είναι καταναμημένοι σε πολλές πλατφόρμες, μια εφαρμογή ιστού ή υβριδική μπορεί να είναι μια καλύτερη επιλογή.

Λαμβάνοντας υπόψη αυτούς τους παράγοντες, μπορείτε να επιλέξετε τον τύπο εφαρμογής για κινητά που ταιριάζει καλύτερα στις ανάγκες σας και παρέχει την καλύτερη εμπειρία χρήστη για το κοινό-στόχο σας εντός των περιορισμών του προϋπολογισμού σας.

Mobile App Type	Pros	Cons
Native Apps	<ul style="list-style-type: none"> • Access to device-specific features • No internet connection needed • Impressive user experience on Android or iOS • The most responsive option • Efficiency driven by platform-specific features 	<ul style="list-style-type: none"> • Higher development costs • Maintenance can be complicated • Native apps tend to be less discoverable than hybrid or web apps
Web Apps	<ul style="list-style-type: none"> • Easy discoverability • Faster development time than native • Simple to maintain • Lower development costs • No need to distribute software to machines that use the application • Updates made to the application and immediately available 	<ul style="list-style-type: none"> • Slower performance than native • Unable to work offline • Not optimized for the given platform • Can't access platform features • Unable to access UI and match style like native apps
Hybrid Apps	<ul style="list-style-type: none"> • Less expensive to sell in app store • Simple to maintain • Lower development costs • Targets multiple platforms and reduces replication costs • Allows app to match look of native application and take advantage of the platform's UI style. 	<ul style="list-style-type: none"> • Slower performance than native • Can't access platform features due to webview restrictions • Not optimized for the given platform

Εικόνα 14: Τα θετικά και τα αρνητικά των διάφορων τύπων εφαρμογών

4. Λειτουργικό Σύστημα Android και Τεχνολογίες.

Το Android είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα για κινητά που αναπτύχθηκε από την Google, βασίζεται στον πυρήνα Linux και έχει σχεδιαστεί κυρίως για κινητές συσκευές με οθόνη αφής, όπως smartphone και tablet.

Το Android είναι χτισμένο πάνω από τη γλώσσα προγραμματισμού Java και χρησιμοποιεί μια εικονική μηχανή (VM) που ονομάζεται Dalvik Virtual Machine για την εκτέλεση κώδικα. Οι προγραμματιστές μπορούν να δημιουργήσουν εφαρμογές Android χρησιμοποιώντας το Android Software Development Kit (SDK), το οποίο περιλαμβάνει μια σειρά εργαλείων και βιβλιοθηκών για τη δημιουργία, τη δοκιμή και τον εντοπισμό σφαλμάτων εφαρμογών Android.

Το Android υποστηρίζει επίσης μια ποικιλία τεχνολογιών, όπως:

Material Design: Μια γλώσσα σχεδιασμού που αναπτύχθηκε από την Google που βοηθά τους προγραμματιστές να δημιουργούν οπτικά ελκυστικές και συνεπείς διεπαφές χρήστη.

Android Runtime (ART): Ένα περιβάλλον χρόνου εκτέλεσης εφαρμογών που χρησιμοποιείται για την εκτέλεση εφαρμογών Android.

Android Debug Bridge (ADB): Ένα εργαλείο γραμμής εντολών που χρησιμοποιείται για την επικοινωνία με μια συσκευή Android ή έναν εξομοιωτή.

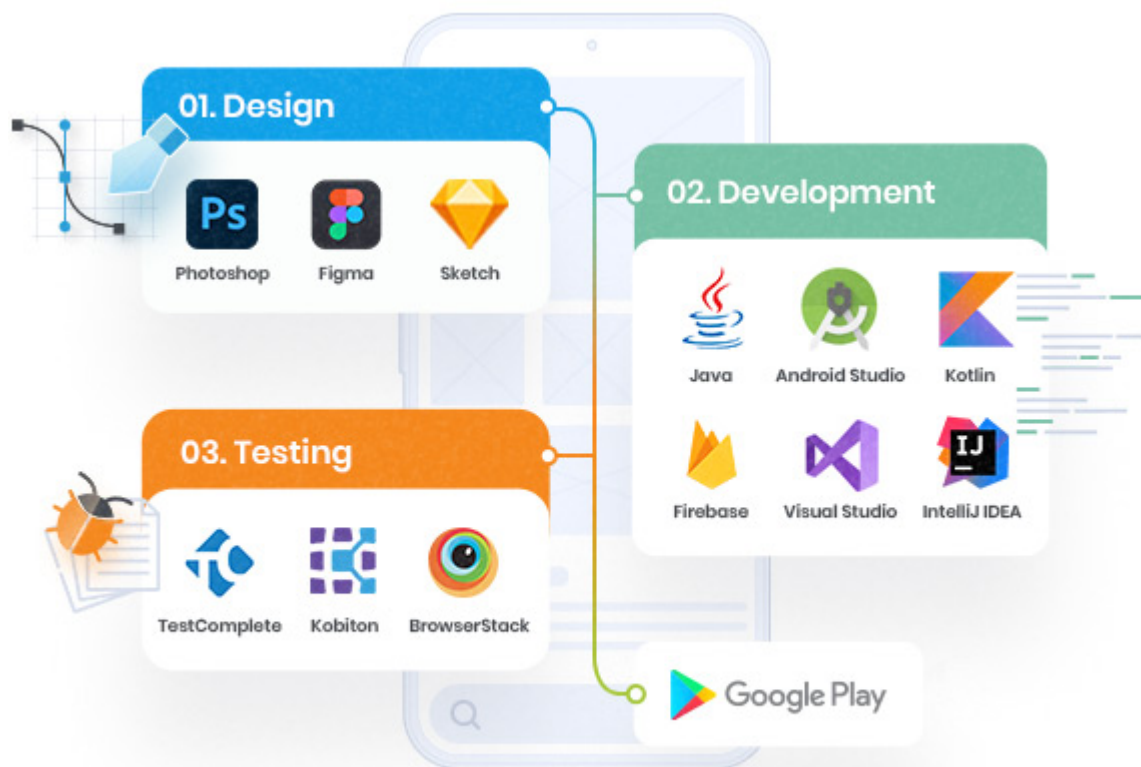
Android Studio: Το επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για την ανάπτυξη εφαρμογών Android.

Firebase: Μια πλατφόρμα ανάπτυξης εφαρμογών για κινητές συσκευές και ιστού που παρέχει μια σειρά εργαλείων και υπηρεσιών για τη δημιουργία εφαρμογών Android, όπως φιλοξενία, αποθήκευση, έλεγχο ταυτότητας και αναλυτικά στοιχεία.

Υπηρεσίες Google Play: Μια συλλογή από API και υπηρεσίες που παρέχονται από την Google που επιτρέπει στους προγραμματιστές να ενσωματώνουν διάφορες υπηρεσίες Google στις εφαρμογές τους Android, όπως Google Maps, το Google Drive και το Google Sign-In.

Kotlin: Μια σύγχρονη γλώσσα προγραμματισμού που υποστηρίζεται πλήρως από το Android και παρέχει μια σειρά λειτουργιών για τη βελτίωση της παραγωγικότητας των προγραμματιστών και της απόδοσης των εφαρμογών.

Συνολικά, το Android είναι ένα ισχυρό και ευέλικτο λειτουργικό σύστημα για κινητά που παρέχει στους προγραμματιστές μια σειρά τεχνολογιών και εργαλείων για τη δημιουργία εφαρμογών για κινητές συσκευές υψηλής ποιότητας και πλούσιες σε δυνατότητες.



Εικόνα 15: Technologies και DEV Tools για το Android

4.1 Η Αρχιτεκτονική του Λειτουργικού Συστήματος Android.

Το λειτουργικό σύστημα Android έχει πολυεπίπεδη αρχιτεκτονική που αποτελείται από τέσσερα κύρια επίπεδα.

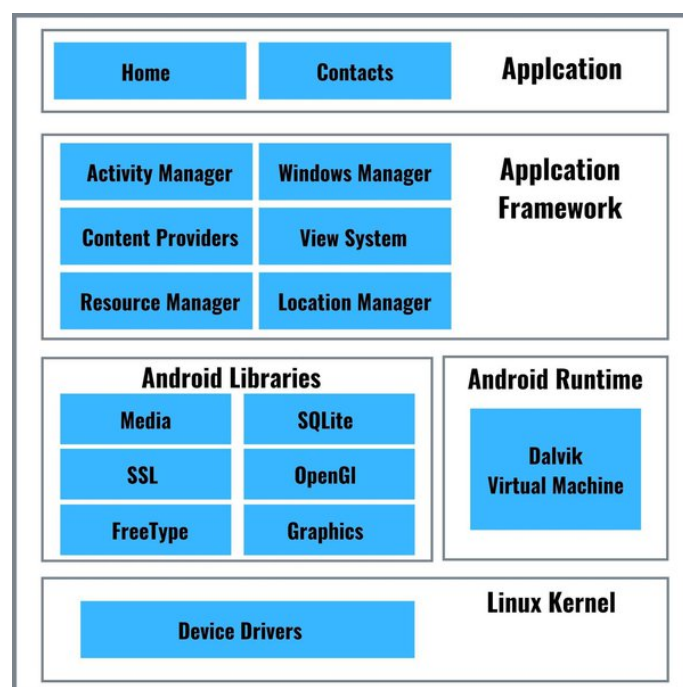
Linux kernel: Στο χαμηλότερο επίπεδο βρίσκεται ο πυρήνας Linux, ο οποίος παρέχει βασικές υπηρεσίες συστήματος όπως διαχείριση μνήμης, διαχείριση διεργασιών και προγράμματα οδήγησης συσκευών. Είναι υπεύθυνο για τη διαχείριση των πόρων υλικού της συσκευής, συμπεριλαμβανομένης της CPU, της μνήμης και διαφόρων συσκευών εισόδου/εξόδου.

Libraries: Στην κορυφή του πυρήνα υπάρχουν διάφορες βιβλιοθήκες, όπως το Android Runtime (ART) και το Native Development Kit (NDK). Το ART είναι υπεύθυνο για την εκτέλεση και τη διαχείριση εφαρμογών Android, ενώ το NDK παρέχει ένα σύνολο εργαλείων και βιβλιοθηκών για την ανάπτυξη εγγενών εφαρμογών Android σε γλώσσες όπως η C και η C++.

Application framework: Το Android παρέχει ένα σύνολο API και υπηρεσιών που μπορούν να χρησιμοποιήσουν οι προγραμματιστές για τη δημιουργία εφαρμογών Android. Αυτά τα API περιλαμβάνουν υποστήριξη για ανάπτυξη διεπαφής χρήστη, αναπαραγωγή πολυμέσων και διάφορες υπηρεσίες συστήματος, όπως υπηρεσίες και ειδοποιήσεις βάσει τοποθεσίας.

Applications: Το ανώτερο επίπεδο της αρχιτεκτονικής Android είναι το επίπεδο εφαρμογής, το οποίο περιλαμβάνει όλες τις εφαρμογές που εκτελούνται στη συσκευή. Αυτές οι εφαρμογές μπορούν να προεγκατασταθούν στη συσκευή ή να ληφθούν από το Google Play Store.

Η αρχιτεκτονική του Android έχει σχεδιαστεί για να είναι αρθρωτή και επεκτάσιμη, επιτρέποντας στους κατασκευαστές συσκευών να προσαρμόσουν το λειτουργικό σύστημα ώστε να ανταποκρίνεται στις συγκεκριμένες απαιτήσεις υλικού τους. Αυτή η ευελιξία έχει συμβάλει στην ευρεία υιοθέτηση του λειτουργικού συστήματος Android σε ένα ευρύ φάσμα συσκευών, από smartphone και tablet έως έξυπνες τηλεοράσεις και φορητές συσκευές.



Εικόνα 16: Android Operating System Layer Architecture

4.1.1 Android Linux Kernel.

Το λειτουργικό σύστημα Android είναι χτισμένο πάνω από τον πυρήνα του Linux, ο οποίος είναι πυρήνας λειτουργικού συστήματος ανοιχτού κώδικα που κυκλοφόρησε για πρώτη φορά το 1991. Ο πυρήνας Linux παρέχει τις βασικές υπηρεσίες του συστήματος και διαχειρίζεται τους πόρους του συστήματος όπως η CPU, η μνήμη και input/output operations.

Η ομάδα του Android έχει κάνει αρκετές τροποποιήσεις στον πυρήνα του Linux για την καλύτερη υποστήριξη φορητών συσκευών, όπως η προσθήκη υποστήριξης για διαχείριση ενέργειας, επεξεργασία γραφικών και πρωτόκολλα δικτύωσης όπως Bluetooth και Wi-Fi. Ο πυρήνας Linux είναι επίσης υπεύθυνος για τη διαχείριση των προγραμμάτων οδήγησης υλικού για τα διάφορα στοιχεία της κινητής συσκευής.

Η πλατφόρμα του Android χρησιμοποιεί μια τροποποιημένη έκδοση του πυρήνα Linux, γνωστή ως πυρήνα Android, η οποία περιλαμβάνει πρόσθετες λειτουργίες, όπως υποστήριξη για συγκεκριμένο υλικό Android και προγράμματα οδήγησης συσκευών. Αυτός ο πυρήνας περιλαμβάνει επίσης μια ειδική διεπαφή προγράμματος οδήγησης χώρου χρήστη που ονομάζεται πρόγραμμα οδήγησης Binder, η οποία χρησιμοποιείται για επικοινωνία μεταξύ των διεργασιών μεταξύ διαφορετικών εφαρμογών Android και στοιχείων συστήματος.

Συνολικά, ο πυρήνας Android Linux διαδραματίζει κρίσιμο ρόλο στην αρχιτεκτονική του λειτουργικού συστήματος Android, παρέχοντας τις βασικές υπηρεσίες συστήματος και διαχειρίζοντας τους πόρους υλικού που απαιτούνται για την υποστήριξη των διαφόρων εφαρμογών και λειτουργιών Android.

4.1.2 Android Hardware Abstraction Layer (HAL).

Το Android Hardware Abstraction Layer (HAL) είναι ένα επίπεδο λογισμικού στο λειτουργικό σύστημα του Android που αφαιρεί τις λεπτομέρειες για το υλικό της συσκευής από το πλαίσιο Android υψηλότερου επιπέδου.

Το HAL παρέχει μια τυποποιημένη διεπαφή για το πλαίσιο του Android για αλληλεπίδραση με διάφορα στοιχεία υλικού, όπως η κάμερα, το "ηχοσύστημα", το GPS και άλλα. Λειτουργεί ως γέφυρα μεταξύ του υλικού και των επιπέδων λογισμικού του λειτουργικού συστήματος Android. Αυτή η αφαίρεση επιτρέπει στους κατασκευαστές συσκευών να γράφουν τα δικά τους προγράμματα οδήγησης για συγκεκριμένο υλικό, διατηρώντας παράλληλα το βασικό πλαίσιο Android ανεξάρτητο από την πλατφόρμα υλικού.

Το HAL είναι γραμμένο σε C και C++ και συνήθως παρέχεται ως κοινόχρηστη βιβλιοθήκη από τον κατασκευαστή της συσκευής. Επικοινωνεί με τα προγράμματα οδήγησης συσκευών μέσω τυπικών διεπαφών πυρήνα Linux όπως ioctl(), mmap() και sysfs. Το HAL περιλαμβάνει επίσης ένα σύνολο τυποποιημένων διεπαφών που επιτρέπουν στο πλαίσιο Android να επικοινωνεί με τα στοιχεία υλικού με συνεπή και ανεξάρτητο από την πλατφόρμα τρόπο.

Το HAL είναι ένα ουσιαστικό μέρος της αρχιτεκτονικής Android, καθώς επιτρέπει στην πλατφόρμα να υποστηρίξει ένα ευρύ φάσμα συσκευών με ποικίλες διαμορφώσεις υλικού, διατηρώντας παράλληλα συνέπεια στην εμπειρία χρήστη.

4.1.3 Android Runtime (ART).

Android Runtime (ART) είναι ο διαχειριζόμενος χρόνος εκτέλεσης που χρησιμοποιείται από το λειτουργικό σύστημα Android για την εκτέλεση εφαρμογών. Το ART αντικαθιστά την εικονική μηχανή Dalvik, η οποία χρησιμοποιήθηκε σε προηγούμενες εκδόσεις του Android. Η ART είναι υπεύθυνη για τη διαχείριση της εκτέλεσης της εφαρμογής, της εκχώρησης μνήμης και της συλλογής σκουπιδιών.

Ένα από τα κύρια οφέλη του ART είναι η χρήση της συλλογής Ahead-of-Time (AOT). Όταν εγκαθίσταται μια εφαρμογή, το ART μεταγλωττίζει τον κώδικα εφαρμογής σε εγγενή κώδικα μηχανής, ο οποίος μπορεί να εκτελεστεί απευθείας από τον επεξεργαστή της συσκευής. Αυτό μπορεί να βελτιώσει την απόδοση της εφαρμογής, καθώς δεν υπάρχει ανάγκη δυναμικής μεταγλώττισης κώδικα κατά το χρόνο εκτέλεσης.

Το ART περιλαμβάνει επίσης μια σειρά από άλλες δυνατότητες που έχουν σχεδιαστεί για τη βελτίωση της απόδοσης και της αποτελεσματικότητας της εφαρμογής. Για παράδειγμα, περιλαμβάνει έναν μεταγλωττιστή just-in-time (JIT), ο οποίος μπορεί να μεταγλωττίσει δυναμικά κώδικα που εκτελείται συχνά για να βελτιώσει την απόδοση. Περιλαμβάνει επίσης έναν συλλέκτη σκουπιδιών, ο οποίος μπορεί να ανακτήσει αυτόματα τη μνήμη από αχρησιμοποίητα αντικείμενα για να βελτιώσει την απόδοση της μνήμης.

Συνολικά, το ART είναι ένα σημαντικό στοιχείο του λειτουργικού συστήματος Android και διαδραματίζει βασικό ρόλο στη διασφάλιση της αποτελεσματικής και ομαλής εκτέλεσης των εφαρμογών Android σε ένα ευρύ φάσμα συσκευών.

4.1.4 Android Native Libraries.

Android Native Libraries είναι συλλογές “προμεταγλωττισμένου” κώδικα που μπορούν να χρησιμοποιηθούν για την επέκταση της λειτουργικότητας των εφαρμογών Android. Αυτές οι βιβλιοθήκες είναι γραμμένες σε γλώσσες προγραμματισμού όπως η C και η C++ και χρησιμοποιούνται συχνά για την εκτέλεση εργασιών με υπολογιστική ένταση ή για την πρόσβαση σε λειτουργίες σε επίπεδο συστήματος που δεν είναι διαθέσιμες μέσω του Android SDK που βασίζεται σε Java.

Μερικά κοινά παραδείγματα native libraries στο Android περιλαμβάνουν τη βιβλιοθήκη OpenCV για εργασίες όρασης υπολογιστή, τη βιβλιοθήκη OpenSSL για ασφαλή επικοινωνία δικτύου και τη βιβλιοθήκη SQLite για τη διαχείριση βάσεων δεδομένων (Database).

Για να χρησιμοποιήσετε μια native library σε μια εφαρμογή Android, οι προγραμματιστές συνήθως περιλαμβάνουν τη βιβλιοθήκη ως “προμεταγλωττισμένο” δυαδικό αρχείο (.so) στο πακέτο APK της εφαρμογής και, στη συνέχεια, έχουν πρόσβαση στις λειτουργίες της χρησιμοποιώντας κλήσεις Java Native Interface (JNI). Η σωστή χρήση των εγγενών βιβλιοθηκών μπορεί να βελτιώσει σημαντικά την απόδοση και τη λειτουργικότητα των εφαρμογών Android.

4.1.5 Android Java API Framework.

Το Android Java API Framework παρέχει ένα σύνολο εργαλείων και βιβλιοθηκών που μπορούν να χρησιμοποιήσουν οι προγραμματιστές για να δημιουργήσουν εφαρμογές Android χρησιμοποιώντας τη γλώσσα προγραμματισμού Java. Αυτό το πλαίσιο περιλαμβάνει πολλά στοιχεία που επιτρέπουν στους προγραμματιστές να δημιουργούν διεπαφές χρήστη, να εργάζονται με δεδομένα, να διαχειρίζονται πόρους συστήματος και πολλά άλλα.

Ακολουθούν ορισμένα βασικά στοιχεία του Android Java API Framework:

- **Activities:** Αντιπροσωπεύει μια μεμονωμένη οθόνη σε μια εφαρμογή Android.
- **Views and Layouts:** Views είναι τα δομικά στοιχεία μιας διεπαφής χρήστη Android και Layouts καθορίζουν τον τρόπο οργάνωσης των προβολών στην οθόνη.
- **Intents:** Χρησιμοποιούνται για την αποστολή μηνυμάτων μεταξύ διαφορετικών τμημάτων μιας εφαρμογής ή μεταξύ διαφορετικών εφαρμογών.
- **Content Providers:** Επιτρέπουν στις εφαρμογές να μοιράζονται δεδομένα με άλλες εφαρμογές.
- **Services:** Επιτρέπουν στις εφαρμογές να εκτελούν εργασίες παρασκήνιου που δεν απαιτούν διεπαφή χρήστη.
- **Broadcast Receivers:** Ακούν system events ή app events και μπορούν να εκτελέσουν ενέργειες ως απόκριση.
- **Data Storage:** Το Android παρέχει διάφορους τρόπους αποθήκευσης και ανάκτησης δεδομένων από τις εφαρμογές, συμπεριλαμβανομένων των βάσεων δεδομένων SQLite, των SharedPreferences και της αποθήκευσης αρχείων.

Συνολικά, το Android Java API Framework παρέχει στους προγραμματιστές ένα ισχυρό σύνολο εργαλείων και βιβλιοθηκών που μπορούν να τους βοηθήσουν να δημιουργήσουν πλούσιες, διαδραστικές και ελκυστικές εφαρμογές Android.

4.1.6 Android System APP.

Android System Apps είναι προεγκατεστημένες εφαρμογές που συνοδεύονται από το λειτουργικό σύστημα Android. Αυτές οι εφαρμογές αποτελούν αναπόσπαστο μέρος του συστήματος και παρέχουν βασικές λειτουργίες για τη σωστή λειτουργία της συσκευής.

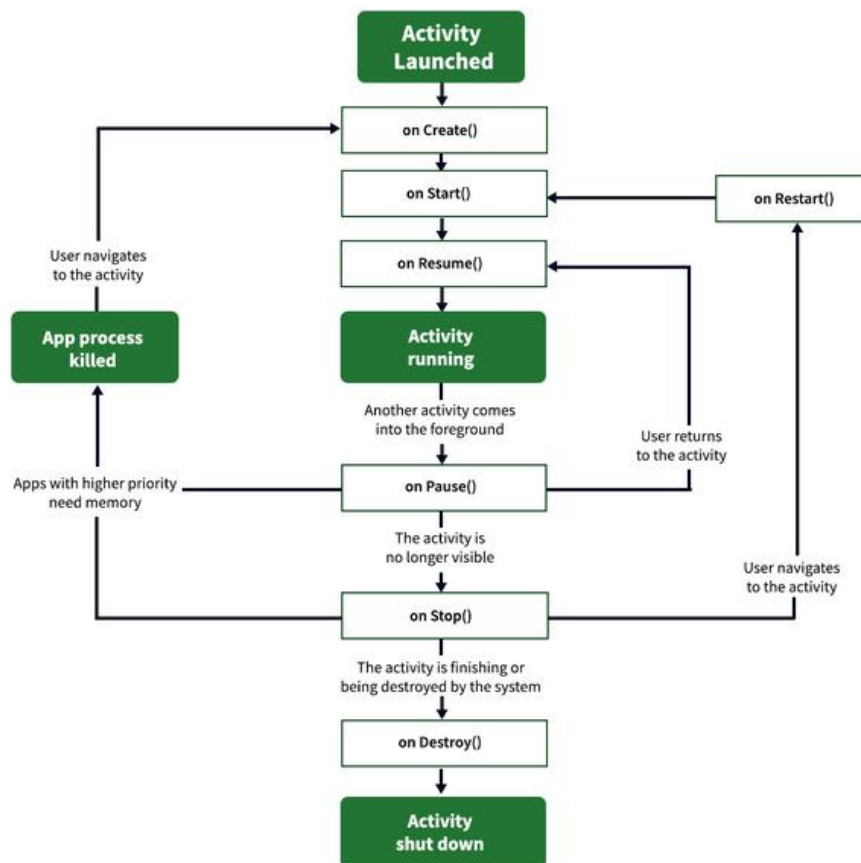
Μερικά παραδείγματα εφαρμογών συστήματος του Android περιλαμβάνουν:

- **Settings:** Αυτή η εφαρμογή επιτρέπει στους χρήστες να διαχειρίζονται τις ρυθμίσεις της συσκευής, όπως τις ασύρματες συνδέσεις, τις ρυθμίσεις οθόνης και την ασφάλεια της συσκευής.
- **Phone:** Αυτή η εφαρμογή επιτρέπει στους χρήστες να πραγματοποιούν και να λαμβάνουν τηλεφωνικές κλήσεις, καθώς και να διαχειρίζονται τις επαφές τους.
- **Messages:** Αυτή η εφαρμογή επιτρέπει στους χρήστες να στέλνουν και να λαμβάνουν μηνύματα SMS και MMS.
- **Camera:** Αυτή η εφαρμογή παρέχει πρόσβαση στην κάμερα της συσκευής και επιτρέπει στους χρήστες να τραβούν φωτογραφίες και να εγγράφουν βίντεο.
- **Gallery:** Αυτή η εφαρμογή επιτρέπει στους χρήστες να προβάλλουν και να διαχειρίζονται τις φωτογραφίες και τα βίντεό τους.
- **Clock:** Αυτή η εφαρμογή παρέχει πρόσβαση στις λειτουργίες ρολογιού και αφύπνισης της συσκευής.
- **Calendar:** Αυτή η εφαρμογή επιτρέπει στους χρήστες να διαχειρίζονται τα χρονοδιαγράμματα και τα ραντεβού τους.
- **Browser:** Αυτή η εφαρμογή παρέχει πρόσβαση στο διαδίκτυο και επιτρέπει στους χρήστες να περιηγούνται σε ιστότοπους.

Αυτά είναι μερικά μόνο παραδείγματα από τις πολλές εφαρμογές συστήματος του Android που είναι προεγκατεστημένες σε συσκευές Android.

4.2 Κύκλος ζωής των δραστηριοτήτων (Activity Life Cycle).

Στο Android, μια δραστηριότητα είναι μια ενιαία οθόνη που έχει σχεδιαστεί για να παρέχει στον χρήστη μια διεπαφή για την εκτέλεση μιας συγκεκριμένης εργασίας. Ο κύκλος ζωής δραστηριότητας αναφέρεται στις διαφορετικές καταστάσεις στις οποίες μπορεί να βρίσκεται μια δραστηριότητα κατά τη διάρκεια της ζωής της.



Activity Lifecycle in Android

Εικόνα 17: Android Activity Life Cycle

Ο κύκλος ζωής της δραστηριότητας αποτελείται από τις ακόλουθες μεθόδους:

onCreate(): Αυτή η μέθοδος καλείται όταν δημιουργείται για πρώτη φορά η δραστηριότητα. Χρησιμοποιείται για την προετοιμασία της δραστηριότητας και τη ρύθμιση των απαραίτητων πόρων.

onStart(): Αυτή η μέθοδος καλείται όταν η δραστηριότητα γίνει ορατή στον χρήστη. Προετοιμάζει τη δραστηριότητα για να μπει στο προσκήνιο και να αρχίσει να αλληλεπιδρά με τον χρήστη.

onResume(): Αυτή η μέθοδος καλείται όταν η δραστηριότητα βρίσκεται στο προσκήνιο και είναι έτοιμη να αλληλεπιδράσει με τον χρήστη. Χρησιμοποιείται για την εκκίνηση κινούμενων εικόνων, τη συνέχιση συνδέσεων δικτύου και την ενημέρωση στοιχείων διεπαφής χρήστη.

onPause(): Αυτή η μέθοδος καλείται όταν η δραστηριότητα εξακολουθεί να είναι μερικώς ορατή στον χρήστη, αλλά δεν έχει πλέον εστίαση. Χρησιμοποιείται για την απελευθέρωση πόρων που δεν χρειάζονται ενώ η δραστηριότητα είναι σε παύση.

onStop(): Αυτή η μέθοδος καλείται όταν η δραστηριότητα δεν είναι πλέον ορατή στον χρήστη. Χρησιμοποιείται για την απελευθέρωση τυχόν πόρων που δεν χρειάζονται πλέον.

onRestart(): Αυτή η μέθοδος καλείται όταν η δραστηριότητα πρόκειται να επανεκκινηθεί μετά τη διακοπή.

onDestroy(): Αυτή η μέθοδος καλείται όταν η δραστηριότητα πρόκειται να καταστραφεί. Χρησιμοποιείται για την απελευθέρωση τυχόν πόρων που εξακολουθούν να κατέχονται από τη δραστηριότητα.

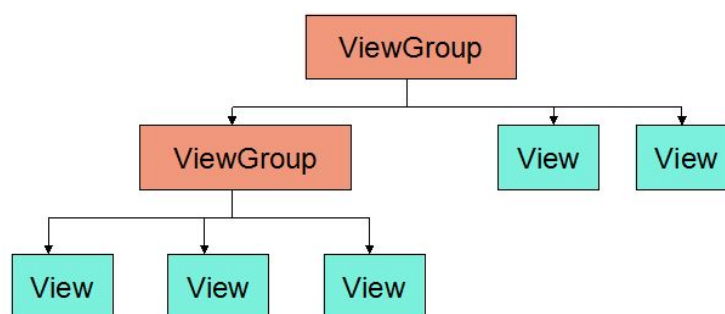
Κατανοώντας τον κύκλο ζωής της δραστηριότητας, οι προγραμματιστές μπορούν να διασφαλίσουν ότι οι εφαρμογές τους έχουν σχεδιαστεί για να χειρίζονται διαφορετικά σενάρια και είναι βελτιστοποιημένες για απόδοση και χρήση πόρων.

4.3 Διεπαφή χρήστη (Android User Interface).

Η διεπαφή χρήστη (User Interface - UI) μιας εφαρμογής Android αναφέρεται στα οπτικά και διαδραστικά στοιχεία που επιτρέπουν στους χρήστες να αλληλεπιδρούν με την εφαρμογή. Το πλαίσιο διεπαφής χρήστη Android παρέχει μια ποικιλία προκατασκευασμένων στοιχείων διεπαφής χρήστη, όπως κουμπιά, πεδία κειμένου, λίστες και εικόνες, τα οποία οι προγραμματιστές μπορούν να χρησιμοποιήσουν για να δημιουργήσουν τη διεπαφή χρήστη της εφαρμογής τους.

Το βασικό δομικό στοιχείο μιας διεπαφής χρήστη Android είναι μια δραστηριότητα, η οποία αντιπροσωπεύει μια ενιαία οθόνη με τη δική της διεπαφή χρήστη. Μέσα σε μια δραστηριότητα, οι προγραμματιστές μπορούν να χρησιμοποιήσουν διαχειριστές διάταξης, όπως `LinearLayout` ή `RelativeLayout`, για να τοποθετήσουν στοιχεία διεπαφής χρήστη στην οθόνη. Αυτά τα στοιχεία μπορούν να προσαρμοστούν με μια ποικιλία ιδιοτήτων, όπως χρώμα, μέγεθος και γραμματοσειρά.

Το Android παρέχει επίσης υποστήριξη για πιο προηγμένες λειτουργίες διεπαφής χρήστη, όπως κινούμενα σχέδια και προσαρμοσμένες προβολές. Τα κινούμενα σχέδια μπορούν να χρησιμοποιηθούν για την παροχή οπτικών σχολίων στους χρήστες, όπως όταν γίνεται κλικ σε ένα κουμπί ή όταν φορτώνονται δεδομένα. Οι προσαρμοσμένες προβολές επιτρέπουν στους προγραμματιστές να δημιουργήσουν τα δικά τους στοιχεία διεπαφής χρήστη που μπορούν να επαναχρησιμοποιηθούν σε μια εφαρμογή.



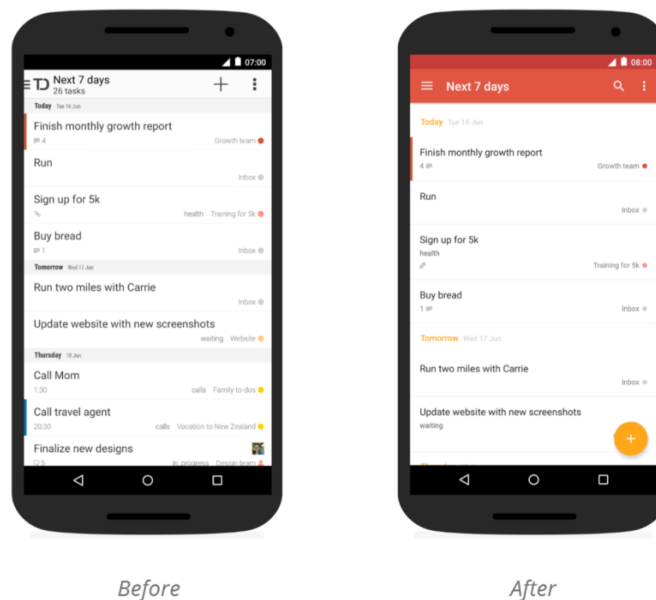
Εικόνα 18: Android User Interface

4.4 Android Material Design.

Το Material Design είναι μια σχεδιαστική γλώσσα που αναπτύχθηκε από την Google για τη δημιουργία ενός ενοποιημένου σχεδίου στον σχεδιασμό των εφαρμογών του Android. Βασίζεται σε ένα σύνολο κατευθυντήριων αρχών που έχουν ως στόχο τη βελτίωση της χρηστικότητας και της ευκολίας χρήσης των εφαρμογών σε κινητές συσκευές.

Μερικά βασικά στοιχεία του Material Design είναι η εναρμόνιση των χρωμάτων, η χρήση καθαρών γραμμών και απλών σχημάτων, η κίνηση και η απόκριση των στοιχείων σε διάφορες καταστάσεις και η χρήση σκιών για την προβολή διαφορετικών επιπέδων.

Οι προγραμματιστές μπορούν να χρησιμοποιήσουν τα στοιχεία του Material Design στις εφαρμογές τους με τη βοήθεια του Android Support Library ή του Material Components Library. Αυτό διευκολύνει την ανάπτυξη εφαρμογών που έχουν μια εναρμονισμένη και ελκυστική εμφάνιση στις κινητές συσκευές που χρησιμοποιούν το Android λειτουργικό σύστημα.



Εικόνα 19: Η διαφορά πριν και μετά την χρήση του Material Design.

4.5 Γλώσσα προγραμματισμού Kotlin.

Το Kotlin είναι μια γλώσσα προγραμματισμού που έχει σχεδιαστεί για την ανάπτυξη Android εφαρμογών και έχει δημιουργηθεί από την εταιρεία JetBrains. Έχει σχεδιαστεί ως μια γλώσσα αντικειμενοστραφούς προγραμματισμού (OOP) και συνδυάζει τα καλύτερα χαρακτηριστικά της Java με σύγχρονες γλώσσες προγραμματισμού, όπως η Scala.

Η Kotlin είναι σχεδιασμένη για να είναι πιο απλή και ευκολότερη στη χρήση από την Java, με μια πιο συμπαγή σύνταξη και λειτουργίες όπως null ασφάλεια, λειτουργία extension και λειτουργίες συνδυασμού (λ.χ. λειτουργία let).

Η Kotlin έχει αναδειχθεί ως μια από τις προτιμώμενες γλώσσες προγραμματισμού για την ανάπτυξη εφαρμογών Android, καθώς έχει βελτιώσει την παραγωγικότητα των προγραμματιστών και έχει επιταχύνει την ανάπτυξη εφαρμογών σε σχέση με την Java. Επιπλέον, η Google έχει ενσωματώσει επίσημα την υποστήριξη της Kotlin στο Android Studio, το επίσημο περιβάλλον ανάπτυξης εφαρμογών Android.

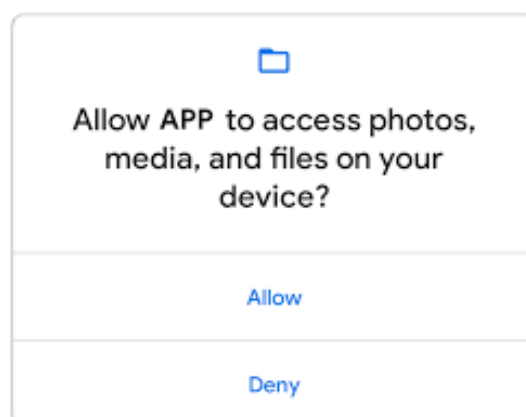
4.6 Δικαιώματα πρόσβασης χρήστη του Android.

Τα δικαιώματα του Android αφορούν τις άδειες που απαιτούνται από μια εφαρμογή για να έχει πρόσβαση σε διάφορες λειτουργίες ή πληροφορίες του συστήματος. Τα δικαιώματα αυτά ελέγχονται από το λειτουργικό σύστημα Android και η άδεια πρέπει να δοθεί από τον χρήστη κατά την εγκατάσταση της εφαρμογής.

Κάποια από τα συνηθέστερα δικαιώματα που απαιτούνται από μια εφαρμογή είναι:

- Δικαιώματα πρόσβασης στην κάμερα
- Δικαιώματα πρόσβασης στις επαφές του χρήστη
- Δικαιώματα πρόσβασης στο αρχείο συστήματος
- Δικαιώματα πρόσβασης στη θέση του χρήστη (GPS)
- Δικαιώματα πρόσβασης στο διαδίκτυο και τα δεδομένα κινητής τηλεφωνίας

Είναι σημαντικό να δίνεται προσοχή στα δικαιώματα που απαιτούνται από μια εφαρμογή και να εξετάζονται προσεκτικά πριν την εγκατάσταση της στη συσκευή μας.

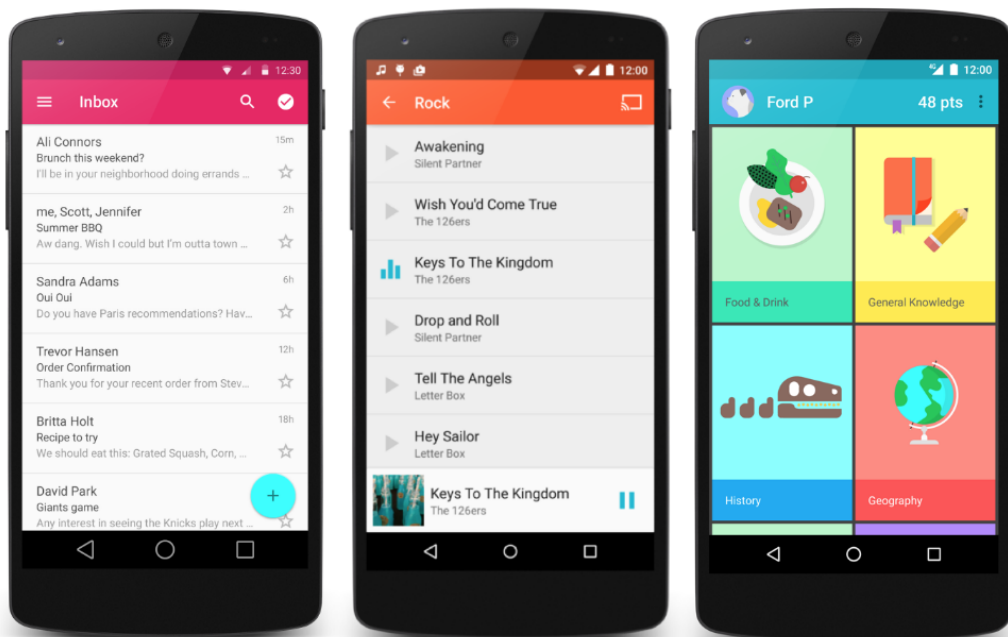


Εικόνα 20: Παράδειγμα πρόσβασης δικαιώματα χρήστη.

4.7 RecyclerView & CardView του Android.

Το RecyclerView είναι ένας σύγχρονος τρόπος για την εμφάνιση μιας λίστας στο Android. Επιτρέπει τη δυναμική δημιουργία λιστών με διάφορα στυλ και ευελιξία στη διαχείριση του περιεχομένου τους. Επίσης, παρέχει τη δυνατότητα επαναχρησιμοποίησης των κυττάρων της λίστας, κάνοντας τον κώδικα πιο επαναχρησιμοποιήσιμο και ευκολότερο στη διαχείριση.

Το CardView είναι ένα widget που επιτρέπει την εμφάνιση πλαισίων καρτών στο Android, παρέχοντας μια ευέλικτη μέθοδο για τη δημιουργία όμορφων και ευανάγνωστων διεπαφών χρήστη. Το CardView είναι ένας ευέλικτος τρόπος για την εμφάνιση περιεχομένου σε ένα πλαίσιο που μπορεί να είναι μονόχρωμο ή να έχει διαφορετικά στυλ, όπως αποχρώσεις χρώματος και σκιές, για να δημιουργήσει μια προσαρμοσμένη εμφάνιση στη διεπαφή του χρήστη.



Εικόνα 21: Παράδειγμα RecyclerView και CardView.

4.8 Google Firebase.

Το Google Firebase είναι μια πλατφόρμα υπηρεσιών cloud που παρέχεται από την Google για την ανάπτυξη εφαρμογών για διάφορες πλατφόρμες, συμπεριλαμβανομένου του Android. Το Firebase προσφέρει πολλές υπηρεσίες, όπως η αυθεντικοποίηση χρηστών, η αποθήκευση δεδομένων στο cloud, η ειδοποίηση push, η ανάλυση απόδοσης και πολλές άλλες.

Για το Android, το Firebase παρέχει SDK για την εύκολη ενσωμάτωση των υπηρεσιών του Firebase στις εφαρμογές. Με τη χρήση του Firebase, οι προγραμματιστές μπορούν να αναπτύξουν εφαρμογές με βελτιωμένες δυνατότητες απόδοσης, επεκτασιμότητας και ασφάλειας, καθώς οι υπηρεσίες του Firebase παρέχονται από το cloud.

Μπορούμε να χρησιμοποιήσουμε το Google Firebase σε οποιαδήποτε εφαρμογή για Android που θέλουμε να αναπτύξουμε και χρειάζεται λειτουργίες όπως αποθήκευση δεδομένων, εξουσιοδότηση χρηστών, αναλυτικά στατιστικά και πολλά άλλα. Μπορούμε να χρησιμοποιήσουμε το Firebase σε εφαρμογές που αναπτύσσουμε από την αρχή, αλλά μπορούμε επίσης να το ενσωματώσουμε σε υπάρχουσες εφαρμογές για να προσθέσουμε λειτουργικότητα και να βελτιώσουμε την εμπειρία των χρηστών.

4.9 JSON.

Το JSON (JavaScript Object Notation) είναι ένας δημοφιλής τρόπος ανταλλαγής δεδομένων μεταξύ εφαρμογών. Είναι ένα απλό και ελαφρύ μορφότυπο δεδομένων, παρόμοιο με το XML, που χρησιμοποιείται για τη μεταφορά δεδομένων από ένα σημείο σε ένα άλλο. Στο πλαίσιο του Android, το JSON χρησιμοποιείται συχνά σε συνδυασμό με τη βιβλιοθήκη GSON της Google για την ανάλυση και δημιουργία JSON αντικειμένων.

Το JSON συνήθως χρησιμοποιείται για τη μεταφορά δεδομένων μέσω του διαδικτύου, και μπορεί να χρησιμοποιηθεί για τη μεταφορά δεδομένων μεταξύ μιας εφαρμογής Android και ενός web server ή μιας άλλης εφαρμογής. Η σύνταξη του JSON είναι πολύ απλή και εύκολη στην κατανόηση, και προσφέρει μια αποδοτική και ασφαλή τρόπο ανταλλαγής δεδομένων.

```
{
  "product_name": "Android Phone",
  "product_id": "12345",
  "description": "A high-end Android phone with advanced features.",
  "price": 599.99,
  "availability": true,
  "specs": {
    "screen_size": "6.4 inches",
    "processor": "Qualcomm Snapdragon 888",
    "ram": "8 GB",
    "storage": "128 GB"
  },
  "reviews": [
    {
      "username": "john_doe",
      "rating": 4,
      "comment": "Great phone, but the battery life could be better."
    },
    {
      "username": "jane_smith",
      "rating": 5,
      "comment": "This is the best phone I've ever owned!"
    }
  ]
}
```

Εικόνα 22: Παράδειγμα JSON μηνύματος.

Σε αυτό το παράδειγμα, οι πληροφορίες για το προϊόν αποθηκεύονται σε ένα αντικείμενο JSON, το οποίο περιλαμβάνει το όνομα του προϊόντος, το ID του, μια περιγραφή, την τιμή του, τη διαθεσιμότητά του, τις προδιαγραφές και τις κριτικές του. Οι προδιαγραφές του προϊόντος αποθηκεύονται σε ένα εμφωλευμένο αντικείμενο JSON, ενώ οι κριτικές αποθηκεύονται σε έναν πίνακα JSON.

4.10 Εργαλεία Ανάπτυξη.

Στην ενότητα αυτήν θα δούμε τα εργαλεία που χρησιμοποιήσαμε για την ανάπτυξη της διπλωματικής εργασίας.

4.10.1 Android Studio.

Το Android Studio είναι μια ολοκληρωμένη περιβάλλον ανάπτυξης εφαρμογών για το λειτουργικό σύστημα Android. Βασίζεται στο IntelliJ IDEA από την JetBrains και είναι επίσημο προγραμματιστικό εργαλείο της Google για την ανάπτυξη εφαρμογών Android. Υποστηρίζει την ανάπτυξη εφαρμογών για το Android σε Java, Kotlin και C++ με τη χρήση του Android NDK. Μερικά από τα βασικά χαρακτηριστικά του Android Studio είναι η δυνατότητα σύνδεσης στο Firebase, η ενσωμάτωση του Android Emulator, η υποστήριξη του Android App Bundle, η δυνατότητα εκτέλεσης τεστ και η αυτόματη πρόταση κώδικα (auto-complete) και η αυτόματη διόρθωση (auto-correct) κώδικα.

4.10.2 GitHub.

Το GitHub είναι μια πλατφόρμα φιλοξενίας κώδικα που επιτρέπει στους προγραμματιστές να αποθηκεύουν, να διαχειρίζονται και να κοινοποιούν τον κώδικά τους στο Διαδίκτυο. Είναι επίσης ένα κοινότητα προγραμματιστών όπου μπορούν να συνεργαστούν, να μοιράζονται γνώσεις και να συνεισφέρουν σε ανοιχτού κώδικα έργα. Το GitHub παρέχει επίσης διάφορα εργαλεία διαχείρισης έργων, όπως παρακολούθηση προβλημάτων, συνεργατικές διαδικασίες, σύνδεση με συνεργάτες και πολλά άλλα.

4.10.3 SmartGit.

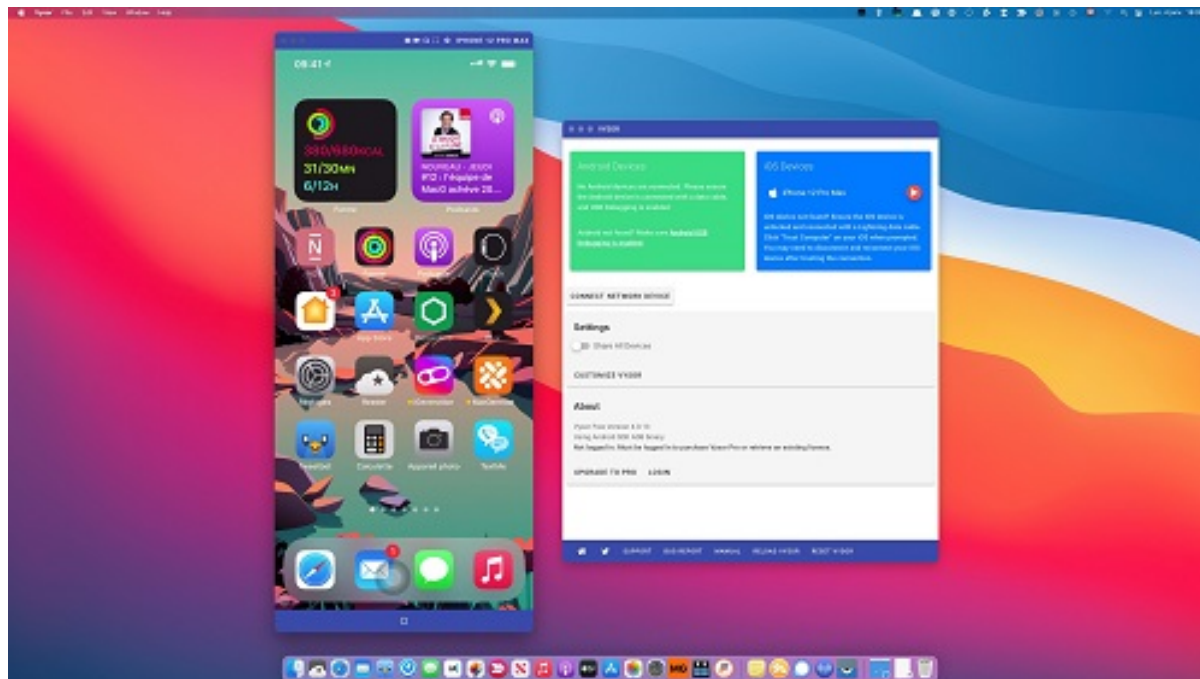
Το SmartGit είναι ένα εργαλείο διαχείρισης αποθετηρίων Git που προσφέρει μια ευέλικτη και εύχρηστη διεπαφή χρήστη για την εκτέλεση διάφορων εργασιών σε ένα αποθετήριο Git. Παρέχει δυνατότητες όπως η διαχείριση κλαδιών, η σύγκριση αλλαγών, η συγχώνευση κλαδιών και η αποθήκευση αλλαγών στο αποθετήριο.

Με το SmartGit, μπορείτε επίσης να δημιουργήσετε νέα αποθετήρια, να διαχειριστείτε διαφορετικά αποθετήρια μέσω μιας κεντρικής διεπαφής, να δημιουργήσετε και να εφαρμόσετε αλλαγές, να δείτε το ιστορικό των αλλαγών και να εκτελέσετε προσαρμοσμένα σενάρια σε διαφορετικά σημεία του κύκλου ζωής του αποθετηρίου. Επιπλέον, το SmartGit υποστηρίζει επίσης τη διαχείριση αποθετηρίων Subversion (SVN).

4.10.4 Vysor.

Το Vysor είναι ένα εργαλείο που σας επιτρέπει να ελέγχετε το Android smartphone ή tablet σας από τον υπολογιστή σας μέσω USB ή Wi-Fi σύνδεσης. Με το Vysor μπορείτε να εκτελέσετε το Android στην οθόνη του υπολογιστή σας, να χρησιμοποιήσετε το πληκτρολόγιο του υπολογιστή σας για την εισαγωγή κειμένου στο Android συσκευή σας, και να κάνετε αντιγραφή και επικόλληση από τον υπολογιστή σας στο Android σας.

Το Vysor είναι ιδιαίτερα χρήσιμο για ανάπτυξη εφαρμογών Android ή για χρήστες που θέλουν να έχουν μια μεγαλύτερη οθόνη για την προβολή του Android συστήματός τους.



Εικόνα 23: Vysor σε Mac OS λειτουργικό.

5. Ανάλυση Λειτουργιών της Εφαρμογής.

Σε αυτό το κεφάλαιο θα αναφερθούμε στην ανάλυση απαιτήσεων, τον σχεδιασμό και την βάση δεδομένων της εφαρμογής. Τέλος θα αναφερθούμε στον τρόπο υλοποίησης της εφαρμογής.

5.1 Ανάλυση Απαιτήσεων της Εφαρμογής.

Μέσω της εφαρμογής Maitre ο χρήστης θα μπορεί να συνδεθεί ως σερβιτόρος και να πάρει παραγγελίες από τους πελάτες ή να συνδεθεί σαν υπεύθυνος κουζίνας και του δίνετε η δυνατότητα να δει τις παραγγελίες, να περάσει πιάτα ημέρας στον κατάλογο αλλά ακόμα και να ενημερώσει το inventory που αυτό με την σειρά του ενημερώνει με τον κατάλογο του σερβιτόρου.

Οι απαιτήσεις της εφαρμογής ήταν:

- Φιλικό και εύχρηστο γραφικό περιβάλλον.
- Γρήγορη εφαρμογή.
- Αξιόπιστη εφαρμογή μέσω της επιδιόρθωσης των σφαλμάτων.
- Χρήση cloud για μεγαλύτερη ευελιξία.
- Υποστήριξη διαφορετικών συσκευών ανάλογα με τα τεχνικά χαρακτηριστικά, την έκδοση του λογισμικού και το μέγεθος της οθόνης.
- Χρήση σύγχρονων αρχιτεκτονικών και βιβλιοθηκών.
- Πρωσική εγγράφων(πιάτο ημέρας) στον κατάλογο.
- Χρήση material components της Google.
- Κατάλληλο optimization στα δεδομένα.
- Παρακολούθηση του inventory.

Η εφαρμογή έχει αναπτυχθεί ώστε να μπορεί να εκτελεστεί από συσκευές με έκδοση μεγαλύτερη του λειτουργικού συστήματος Android 6.0. Έχει χρησιμοποιηθεί το firebase για αποθήκευση των παραγγελιών στο cloud. Στο cloud αποθηκεύονται επίσης όλες οι εικόνες που χρησιμοποιεί η εφαρμογή. Έχει γίνει χρήση της λειτουργίας RecyclerView για την καλύτερη περιήγησή του χρήστη.

5.2 Σχεδιασμός Εφαρμογής.

Από την ανάλυση αποτίσεων που αναφέραμε στο κεφάλαιο 5.1 θα αναφέρουμε ενδεικτικά παραδείγματα από τον σχεδιασμό της εφαρμογής.

5.2.1 Δυνατότητες και Λειτουργίες των Χρηστών στην Εφαρμογή.

Στην ενότητα αυτή θα μιλήσουμε για της δυνατότητες και λειτουργίες που έχει ο χρήστης στην εφαρμογή αυτήν.

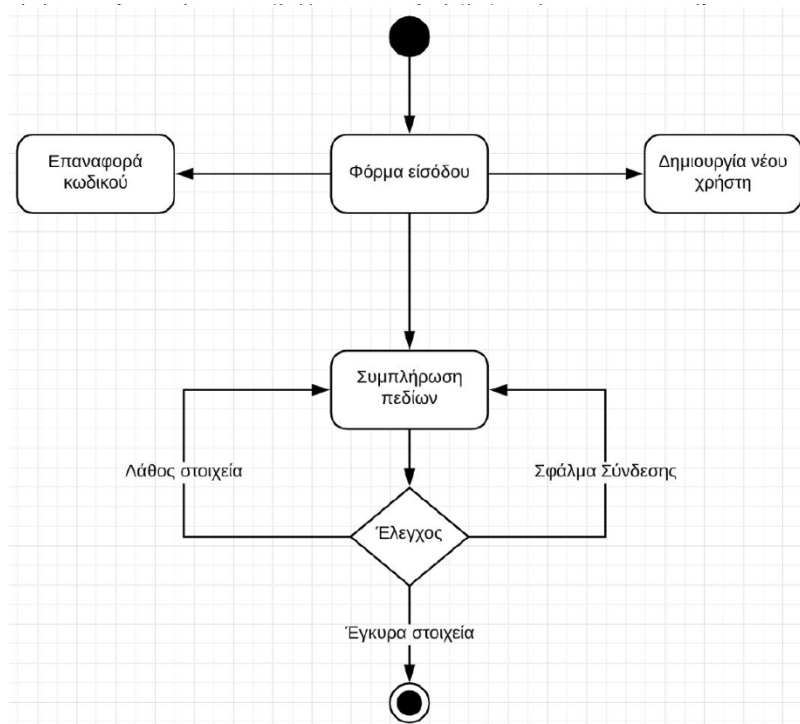
Οι λειτουργίες που είναι διαθέσιμες στην εφαρμογή είναι:

- Splash οθόνη καλωσορίσματος στην εφαρμογή.
- Οθόνη σύνδεσης στην εφαρμογή.
- Κεντρική οθόνη σερβιτόρου.
 - Επιλογή τραπεζιού
 - Πρόσθεση ή αφαίρεση αριθμό επιλογής προϊόντος
 - Αποστολή παραγγελίας στην κουζίνα με την χρήση firebase.
- Κεντρική οθόνη κουζίνας / μπουφέ.
 - Δυνατότητα προβολής παραγγελίας ανά τραπέζι και διαγραφή της.
 - Ενημέρωση μενού με το απόθεμα κάποιας επιλογής από το μενού
 - Πρόσθεση πιάτου ημέρας στον κατάλογο.
- Κουμπί αποσύνδεσης και αλλαγής χρήστη.

5.2.2 Σχεδιασμός Σύνδεσης Χρήστη στην Εφαρμογή.

Μετα από την ανάλυση αποτίσεων είδαμε πόσο αναγκαίο είναι η σύνδεση του χρήστη στην εφαρμογή να δοθεί προσοχή στον σχεδιασμό της διαδικασίας που θα ακολουθήσει έτσι ώστε να αποφευχθούν τυχόν λάθη.

Σε περίπτωση που κάποιο πεδίο είναι κενό τότε εμφανίζετε στον χρήστη ένα μήνυμα που τον ενημερώνει ποιο πεδίο έχει αφήσει κενό. Επίσης σε περίπτωση που κάτι συμπληρωθεί λάθος πάλι η εφαρμογή θα του εμφανίσει ένα μήνυμα που θα τον ενημερώσει ότι έχει συμπληρώσει λάθος τα στοιχεία. Όταν ο χρήστης συμπληρώσει τον κωδικό του, εμφανίζεται masked και σε περίπτωση που ο χρήστης θέλει να το δει για να επιβεβαιώσει ότι τον έχει γράψει σωστά τότε μπορεί να επιλέξει το “ματάκι” για να του εμφανιστεί ο κωδικός.



Διαγράμματα 4: Σύνδεση στην εφαρμογή

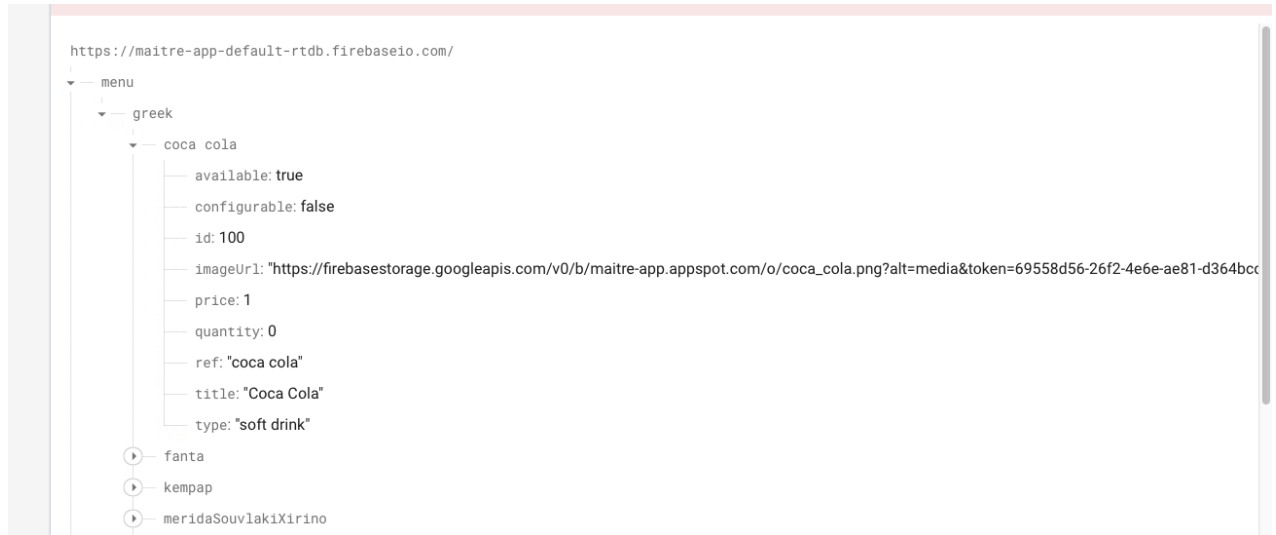
5.2.3 Σχεδιασμός της Βάσης Δεδομένων στην Firebase.

Για βάση δεδομένων επιλέχθηκε να χρησιμοποιήσουμε την Firebase της Google για τον λόγο ότι είναι μια cloud real time database η οποία μας επιτρέπει οι παραγγελίες που στέλνει ο σερβιτόρος να εμφανίζονται άμεσα στην εφαρμογή της κουζίνας/μπουφέ.

Στην real time database έχουμε τρεις πίνακες, το menu το οποίο εκεί είναι αποθηκευμένο το μενού του εστιατορίου που εμφανίζει το μενού στο view του σερβιτόρου, τον πίνακα orders που εμφανίζει την παραγγελία στο view της κουζίνας/μπουφές και τον πίνακα tables που εμφανίζει στο view του σερβιτόρου τα τραπέζια.

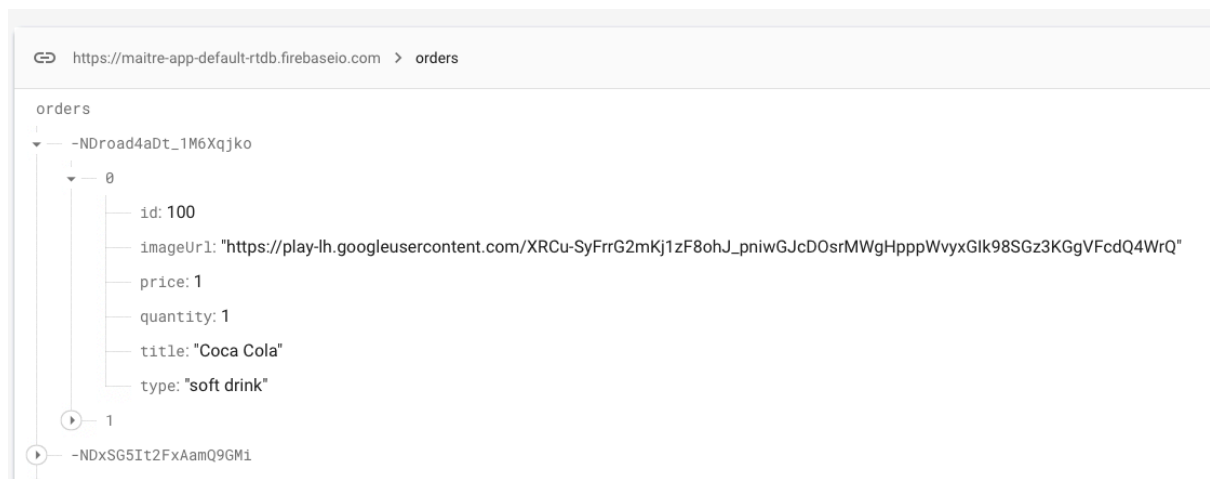
Στον πίνακα **menu** τα πεδία που χρησιμοποιούμε είναι:

- **available:** Το πεδίο αυτό παίρνει τις τιμές true/false ανάλογα αν είναι διαθέσιμο το πιάτο αυτό.
- **configurable:** Το πεδίο αυτό παίρνει τις τιμές true/false ανάλογα αν μπορεί η εγγραφή αυτή να αλλάξει τιμές.
- **id:** Κάθε εγγραφή στον πίνακα αυτόν έχει ένα μοναδικό id, το πεδίο αυτό είναι αριθμητικό.
- **imageUrl:** Στο πεδίο αυτό παίρνει σαν τιμή ένα url που έχει την εικόνα.
- **price:** Το πεδίο αυτό περιέχει την τιμή που εμφανίζετε στο μενού, το πεδίο αυτό είναι αριθμητικό.
- **quantity:** Το πεδίο αυτό περιέχει την ποσότητα που υπάρχει στο inventory για αυτή την εγγραφή, το πεδίο αυτό είναι αριθμητικό.
- **ref:** Το πεδίο αυτό περιέχει το όνομα της εγγραφής.
- **title:** Το πεδίο αυτό περιέχει τον τίτλο της εγγραφής που εμφανίζεται στην εφαρμογή.
- **type:** Το πεδίο αυτό περιγράφει τον τύπο της εγγραφής.



Εικόνα 24: Δομή πίνακα menu

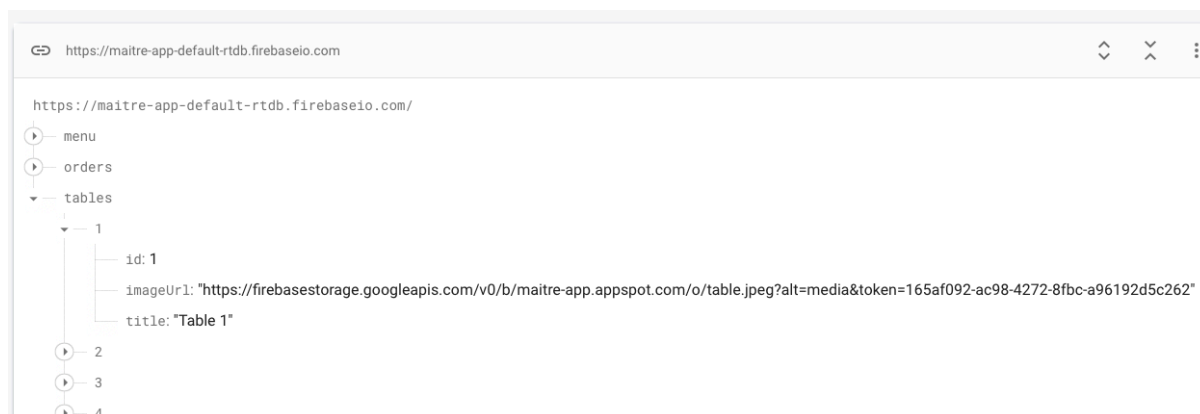
Στον πίνακα **orders** διαθέτη ένα πεδίο **id** που το πεδίο μέσα του περιέχει την πληροφορία για την παραγγελία.



Εικόνα 25: Δομή πίνακα orders

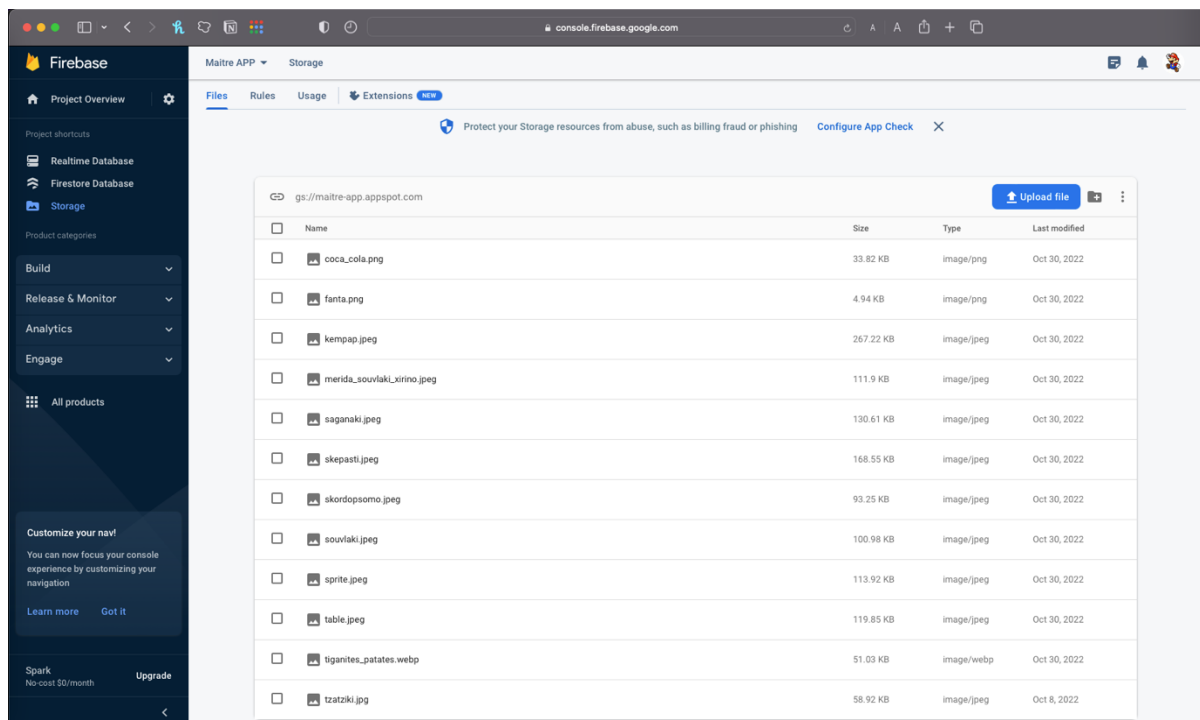
Στον πίνακα **tables** κάθε τραπέζι περιέχει τα έξι πεδία:

- **id**: Κάθε εγγραφή στον πίνακα αυτόν έχει ένα μοναδικό id, το πεδίο αυτό είναι αριθμητικό.
- **imageUrl**: Στο πεδίο αυτό παίρνει σαν τιμή ένα url που έχει την εικόνα.
- **title**: Το πεδίο αυτό περιέχει τον τίτλο της εγγραφής που εμφανίζεται στην εφαρμογή.



Εικόνα 26: Δομή πίνακα tables

Την Firebase την χρησιμοποιούμε επίσης σαν storage για εικόνες που χρησιμοποιούμε στον πίνακα **menu** και **tables** στα πεδία **imageUrl**.



Εικόνα 27: Firebase Storage

6. Παρουσίαση της εφαρμογής Maitre APP.

Σε αυτό το κεφάλαιο θα παρουσιάσουμε την λειτουργικότητα της εφαρμογής και θα δούμε τις δυνατότητες που έχει ο χρήστης. Βοηθητικά-θα παρουσιαστούν screenshots για μεγαλύτερη ευκολία στην κατανόηση, η παρουσίαση αυτή χρησιμεύει και ως user manual.

Σκοπός αυτής της διπλωματικής εργασίας είναι η παροχή μιας εφαρμογής που θα βελτίωση την αλληλοεπίδραση του χρήστη-με την εφαρμογή και η δυνατότητα που θα δόση στην επιχειρήσει να έχει μια απλή, εύχρηστη και πολύ γρήγορη εφαρμογή για παραγγελιοληψίας.

Η παρουσίαση θα αποτελείτε κυρίως από screenshots.

6.1 Οθόνη Σύνδεσης.


Με την είσοδο του χρήστη στην εφαρμογή θα δει ένα animation με το σήμα της εφαρμογής και μετα ακολουθεί η οθόνη σύνδεσης. Ανάλογα με τα centennial που έχει κάθε χρήστης έχει και διαφορετικά views μέσα στην εφαρμογή (σερβιτόρος ή κουζίνα/μπουφές).



Εικόνα 28: Animation πριν την οθόνη σύνδεσης

Login

User Name

Password 

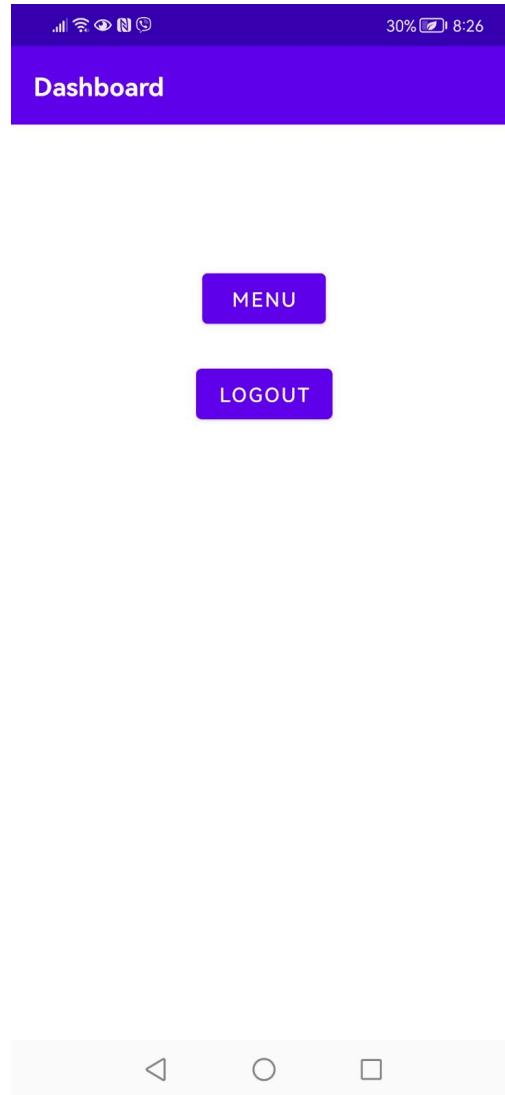
Reset Submit



Εικόνα 29: Οθόνη σύνδεσης - login screen

6.2 View Σερβιτόρου – Αρχική οθόνη.

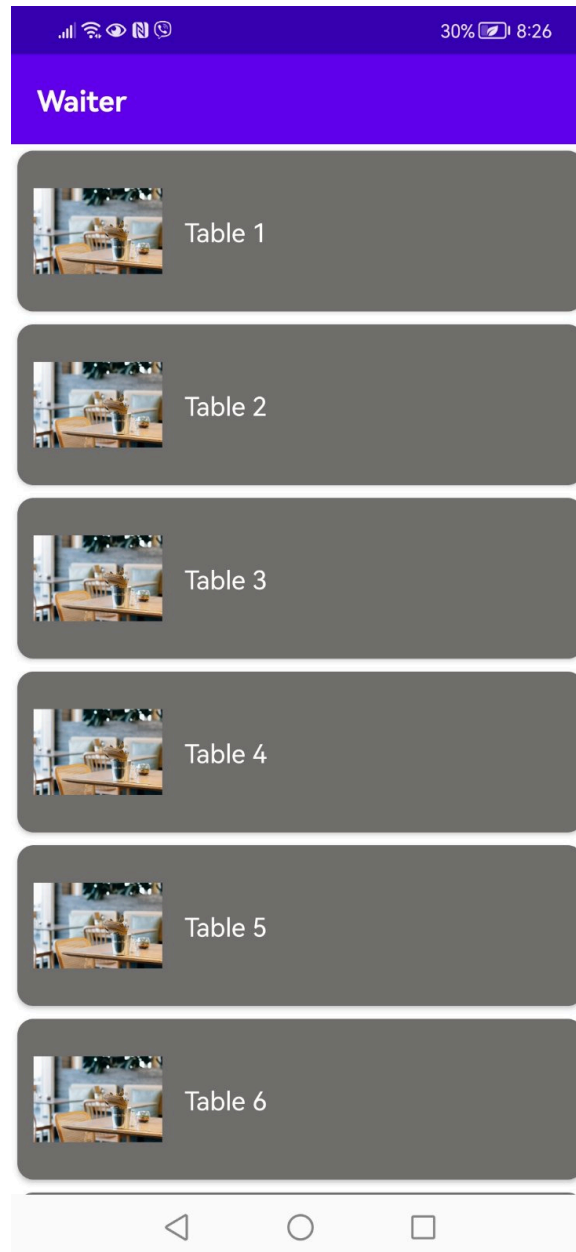
Στην αρχική οθόνη στο view του σερβιτόρου ο χρήστης έχει δύο επιλογές, με το κουμπί **LOGOUT** ο χρήστης πάει στην αρχική οθόνη σύνδεσης ενώ με το κουμπί **MENU** πάει στην οθόνη για να επιλέξει τραπέζι που θα πάρει την παραγγελία.



Εικόνα 30: Αρχική οθόνη Σερβιτόρου

6.3 View Σερβιτόρου – Οθόνη Επιλογής Τραπεζιού.

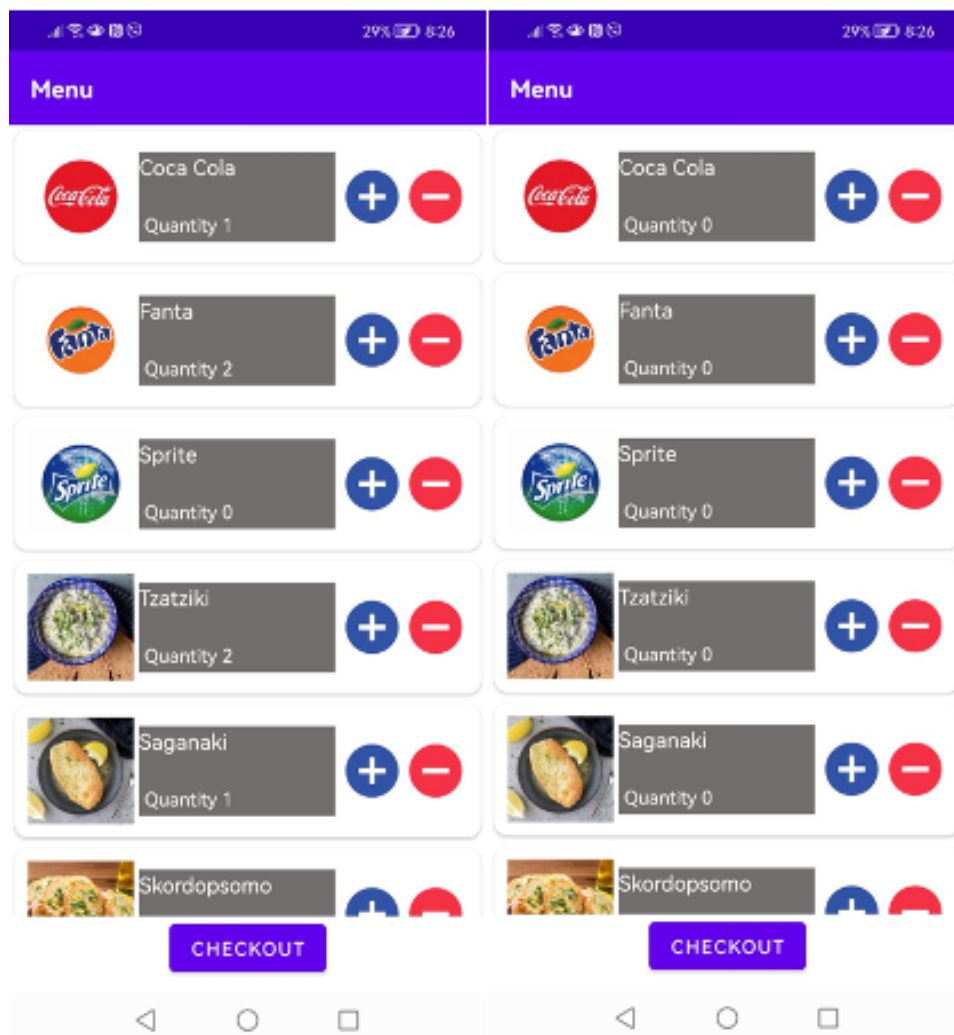
Ο σερβιτόρος εδώ έχει την δυνατότητα να επιλέξει το τραπέζι που θα εξυπηρετήσει. Στην οθόνη αυτήν χρησιμοποιούμε το RecyclerView οπότε ο χρήστης μπορεί να κάνει scroll για να δει και τα υπόλοιπα τραπέζια στην λίστα. Η λίστα αυτή έρχεται από την firebase και τον πίνακα **tables**. Είναι παραμετροποιήσιμος δίνοντας μας την δυνατότητα να προσθέσουμε ή να αφαιρέσουμε τραπέζια, επίσης μπορούμε να αλλάξουμε και την εικόνα.



Εικόνα 31: Οθόνη Επιλογής Τραπεζιού.

6.4 View Σερβιτόρου – Οθόνη Menu.

Ο σερβιτόρος εδώ έχει την δυνατότητα να πάρει την παραγγελία επιλέγοντας το πρωιών και την ποσότητα που ζήτησε ο πελάτης. Όταν ο σερβιτόρος τελειώσει με την παραγγελία πατάει το κουμπί checkout η παραγγελία στέλνεται στην firebase. Με το σύμβολο «+» προσθέτουμε στην παραγγελία την ποσότητα από το προϊόν που θέλουμε και το βλέπουμε εκεί που λέει **Quantity**. Πατώντας το σύμβολο «-» αφαιρούμε κατά ένα την ποσότητα του προϊόντος. Αν είναι μηδέν τότε το προϊόν δεν είναι στην λίστα.

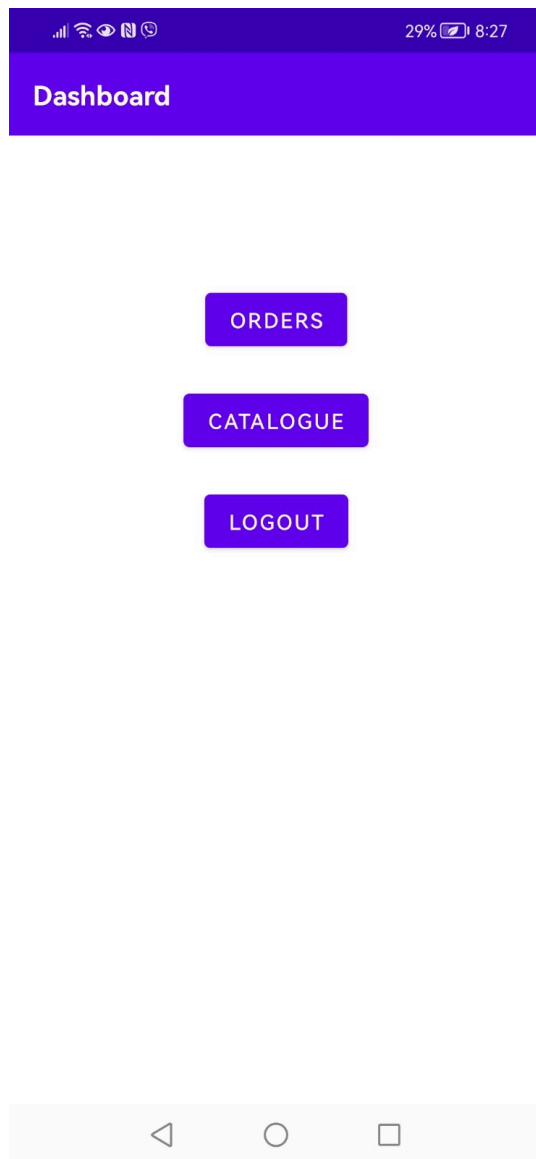


Εικόνα 32: Λίστα μετά την παραγγελία (αριστερά) - Λίστα πριν την παραγγελία (δεξιά).

Στην οθόνη αυτήν χρησιμοποιούμε το Recycler View οπότε ο χρήστης μπορεί να κάνει scroll και να δει όλα τα προϊόντα στην λίστα. Η λίστα αυτή έρχεται από την firebase και τον πίνακα **menu**. Είναι και παραμετροποιήσιμη δίνοντας μας την δυνατότητα να προσθέσουμε ή να αφαιρέσουμε προϊόντα στο μενού ή ακόμα και να αλλάξουμε κάποια από τις υπάρχοντες εγγραφές.

6.5 View Κουζίνας – Αρχική οθόνη.

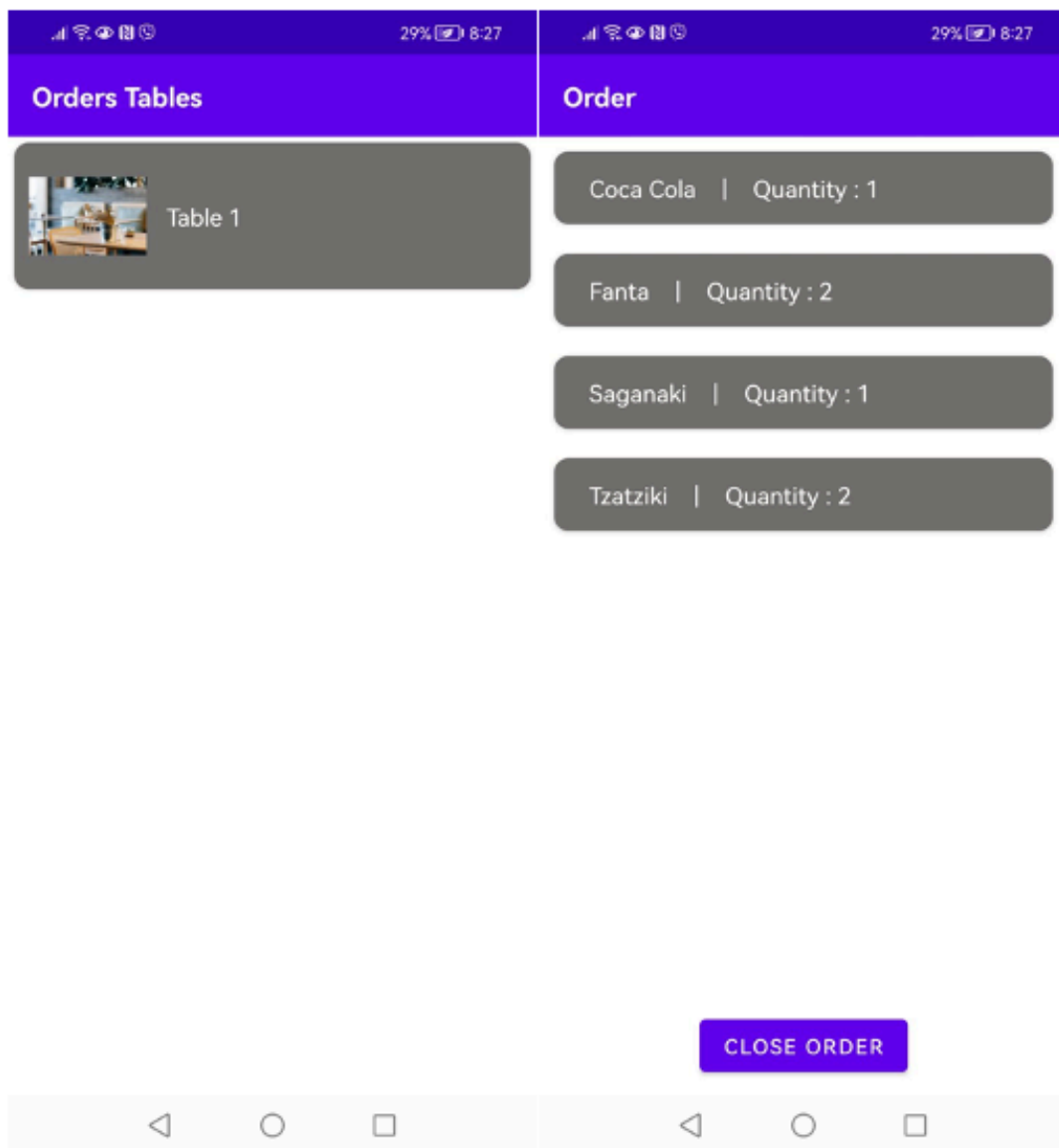
Στην αρχική οθόνη στο view της κουζίνας ο χρήστης έχει τρεις επιλογές, με το κουμπί **LOGOUT** ο χρήστης πάει στην αρχική οθόνη σύνδεσης ενώ με το κουμπί **ORDERS** πάει στην οθόνη για να δει την παραγγελία του τραπέζιου που του έχει στείλει ο σερβιτόρος και με το κουμπί **CATALOGUE** μπορεί να διαχειριστεί το inventory (δηλαδή να ανοίξει ή να κλείσει μια παραγγελία από το μενού) και να προσθέσει τα πιάτα ημέρας.



Εικόνα 33: Αρχική οθόνη Κουζίνας.

6.6 View Κουζίνας – Οθόνη Orders.

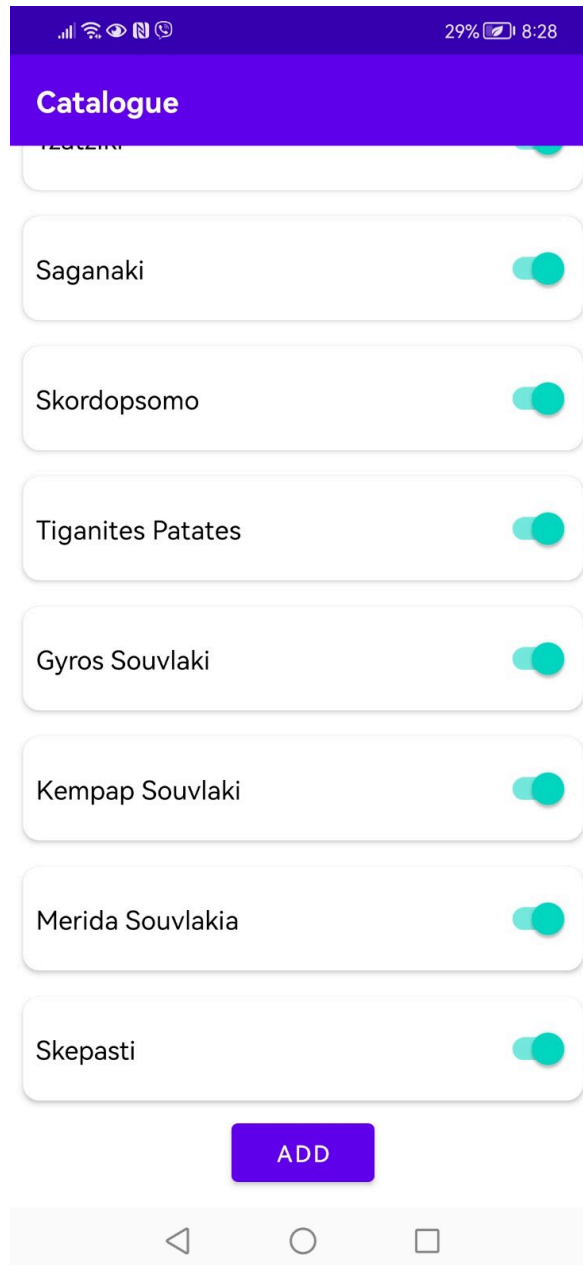
Η κουζίνα από την οθόνη orders βλέπει ποιο τραπέζι εκεί παραγγελία και επιλέγοντας το τραπέζι τον πάει σε μια δεύτερη οθόνη που μπορεί να δει την παραγγελία και την ποσότητα που παρήγγειλε ο πελάτης. Επιλέγοντας το κουμπί **CLOSE ORDER** η κουζίνα κλείνει την παραγγελία. Στις οθόνες αυτές χρησιμοποιούμε RecyclerView οπότε ο χρήστης μπορεί να κάνει scroll και να δει όλα τα προϊόντα στην λίστα. Η λίστα αυτή έρχεται από την firebase και τον πίνακα **orders**.



Εικόνα 34: Οθόνη με τα τραπέζια που έχουν παραγγελιά (αριστερά) και οθόνη με την παραγγελία του τραπεζιού (δεξιά).

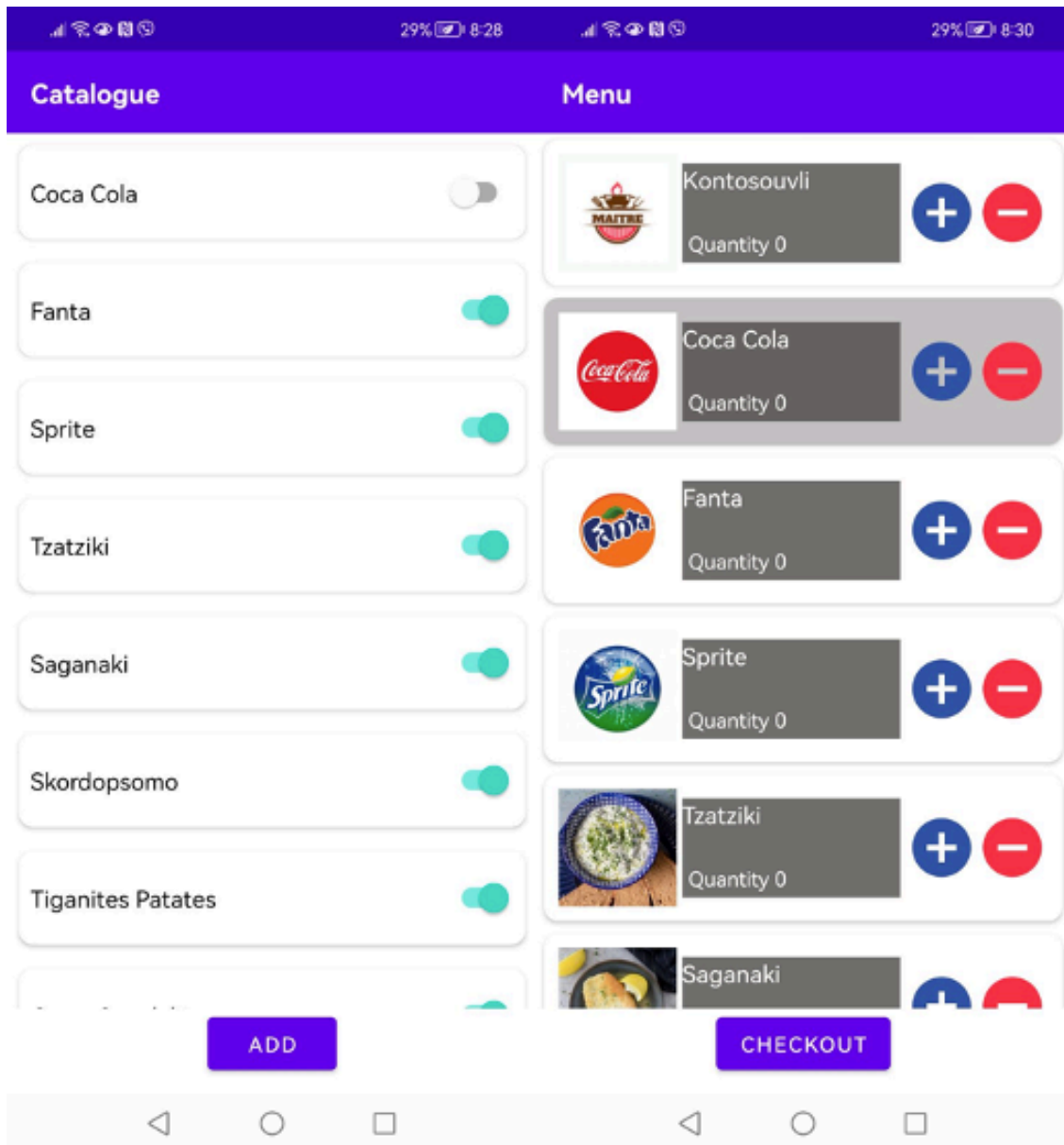
6.7 View Κουζίνας – Οθόνη Catalogue.

Αυτή η οθόνη περιέχει τον κατάλογο του μενού εμφανίζοντας την με την μορφή κάθετης λίστας με την χρήση του Recycler View οπότε ο χρήστης μπορεί να κάνει scroll και να δει όλα τα προϊόντα στην λίστα. Η λίστα αυτή έρχεται από την firebase και τον πίνακα **menu**.



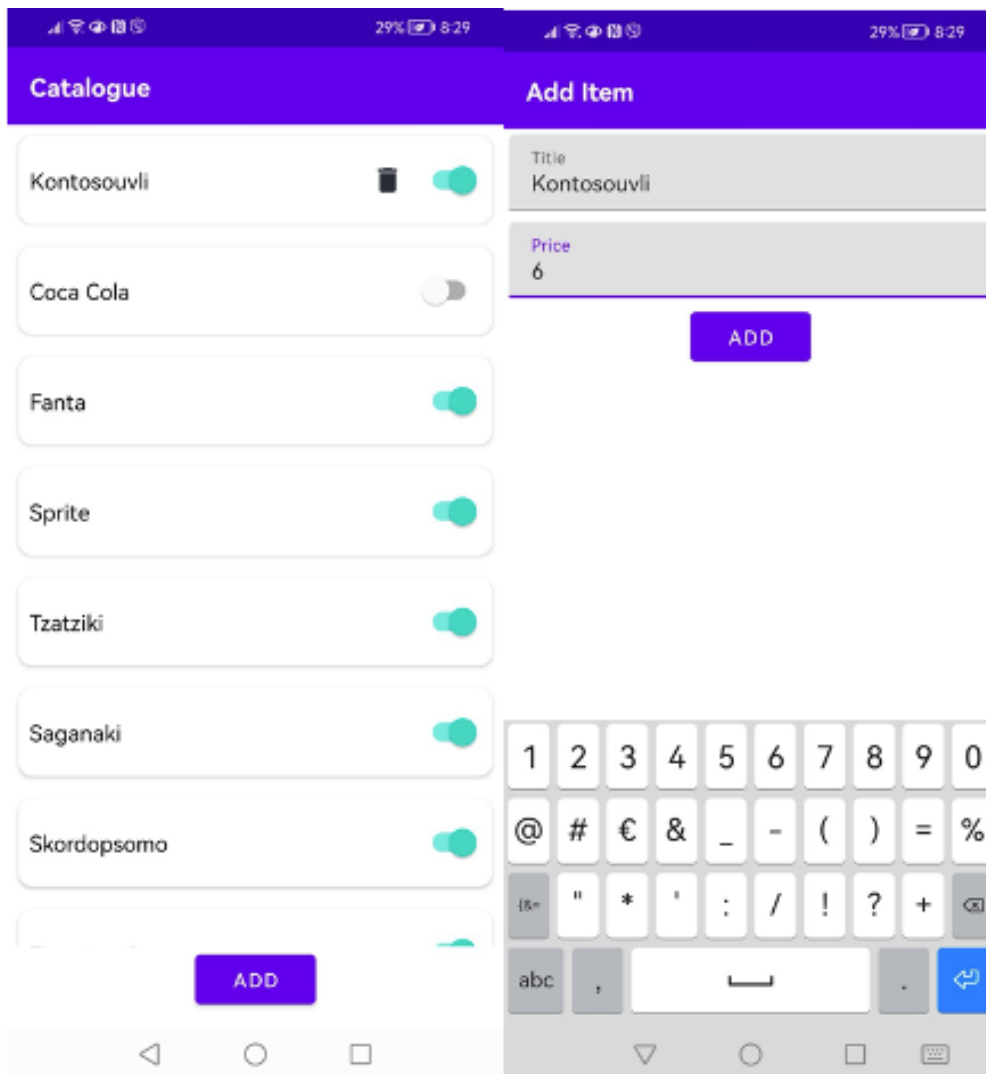
Εικόνα 35: Οθόνη με τον κατάλογο και το inventory.

Η κουζίνα έχει την δυνατότητα να “κλείσει” ένα προϊόν αν έχει τελειώσει. Ενημερώνει την firebase και τον πίνακα του **menu** έτσι ο σερβιτόρος δεν μπορεί να επιλέξει από το μενού το συγκεκριμένο προϊόν και του εμφανίζεται με γκριζο χρώμα.



Εικόνα 36: Οθόνη με τον κατάλογο (αριστερά) - Οθόνη μενού (δεξιά).

Επιλέγοντας το κουμπί **ADD** έχει την δυνατότητα η κουζίνα να προσθέσει ή να αφαιρέσει πιάτο ημέρας και να το “κλείσει”. Στο πιάτο ημέρας μπορεί να προσθέσει τίτλο και τιμή.



Εικόνα 37: Πιάτο ημέρας (αριστερά) - Πρόσθεση πιάτου ημέρας (δεξιά).

7. Επεξήγηση Κώδικα.

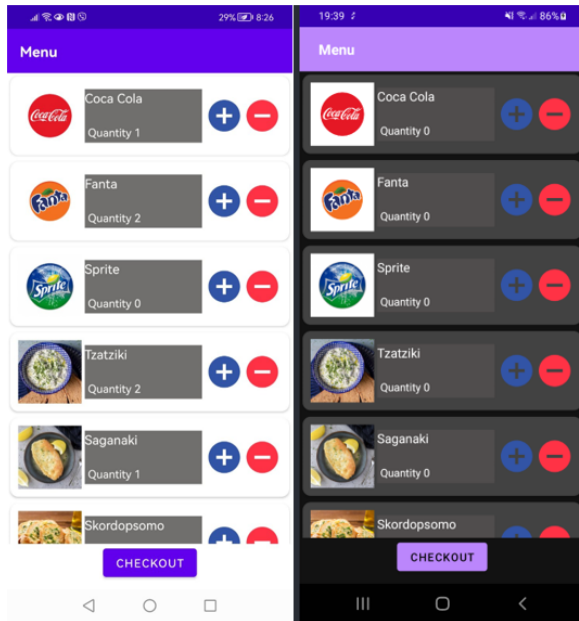
Στο κεφάλαιο αυτό θα δούμε και θα εξηγήσουμε σημαντικά σημεία κώδικα της εφαρμογής και τρόπο λειτουργίας του.

7.1 Dark Mode

Η ύπαρξη δύο διαφορετικών mode εμφάνισης στις εφαρμογές είναι ένα χαρακτηριστικό το οποίο έχει γίνει αρκετά δημοφιλές τα τελευταία χρόνια. Για την υλοποίηση-του απαιτούνται τρία στοιχεία, η δήλωση των χρωμάτων σαν styleable resources στο αρχείο attrs.xml, η δήλωση των δύο διαφορετικών modes στο αρχείο styles.xml και η κλήση μιας function σε κάθε activity. Η function, δεδομένο το ότι θα τρέχει σε κάθε activity είναι υλοποιημένη στο BaseActivity και τρέχει κάνοντας override την onCreate function. Ακολουθούν screenshots με τα xml αρχεία.

```
<resources xmlns:tools="http://schemas.android.com/tools">
  <!-- Base application theme. -->
  <style name="Theme.Maitre" parent="Theme.MaterialComponents.DayNight.NoActionBar">
    <!-- Primary brand color. -->
    <item name="colorPrimary">@color/purple_200</item>
    <item name="colorPrimaryVariant">@color/purple_700</item>
    <item name="colorOnPrimary">@color/black</item>
    <!-- Secondary brand color. -->
    <item name="colorSecondary">@color/teal_200</item>
    <item name="colorSecondaryVariant">@color/teal_200</item>
    <item name="colorOnSecondary">@color/black</item>
    <!-- Status bar color. -->
    <item name="android:statusBarColor" tools:targetApi="1">?attr/colorPrimaryVariant</item>
    <!-- Customize your theme here. -->
  </style>
</resources>
```

Εικόνα 38: Dark mode xml code



Εικόνα 39: Παράδειγμα οθόνης με dark mode (αριστερά) και χωρίς (δεξιά).

7.2 Καταχώριση Νέας Εγγραφής στην Βάση.

Επιλέγοντας το κουμπί **ADD** έχει την δυνατότητα η κουζίνα να προσθέσει ή να αφαιρέσει πιάτο ημέρας . Στο πιάτο ημέρας μπορεί να προσθέσει τίτλο και τιμή. Αυτό γίνεται κάνοντας insert μια νέα εγγραφή στον πίνακα menu της firebase. Έτσι με την χρήση της online βάσης του firebase η οθόνη του μενού που έχει ο σερβιτόρος ανανεώνεται αυτόματα.

Στον κώδικα πρώτα κοιτάμε αν έχει βάλει τίτλο και valid τιμή, δηλαδή ότι δεν είναι κενό ή μηδέν. Μόλις περάσει των έλεγχο γίνεται στον insert στην βάση.

```

addItem.setOnClickListener { @View
    if (title.text?.isEmpty() == true) {
        titleLayout.error = "Add a valid title"
        return@setOnClickListener
    } else {
        titleLayout.error = null
    }

    if (price.text?.isEmpty() == true || price.text.toString() == "0" || price.text.toString() == "0.0") {
        priceLayout.error = "Add a valid price"
        return@setOnClickListener
    } else {
        priceLayout.error = null
    }

    val itemRef = title.text.toString()
    val dataRef = Firebase.database.getReference( path: "menu/greek" )
        .child(itemRef)
    dataRef.setValue(
        MenuItem(
            id = 0,
            title = title.text.toString(),
            price = price.text.toString().toDouble(),
            type = "piatoImeras",
            ref = itemRef,
            isAvailable = true,
            isConfigurable = true
        )
    )
    finish()
}

```

Εικόνα 40: Καταχώριση νέας εγγραφής στην βάση του firebase.

7.3 Επεξήγηση Λίστας (RecyclerView).

Η χρήση ενός RecyclerView μας βοηθάει ώστε να απεικονίζουμε μεγάλο σύνολο δεδομένων το οποίο βελτιώνει την ταχύτητα και την απόκριση της διεπαφής (UI). Στην περίπτωση μας η πλειοψηφία των οθονών στην εφαρμογή μας είναι με την χρήση του RecyclerView.

```
class ItemViewHolder(view: View) : RecyclerView.ViewHolder(view) {  
  
    fun bind(  
        item: MenuItem,  
        onSwitchClicked: (MenuItem) -> Unit,  
        onDeleteClicked: (MenuItem) -> Unit  
    ) {  
  
        val textView = itemView.findViewById<TextView>(R.id.textItem)  
        val switchButton = itemView.findViewById<SwitchMaterial>(R.id.switchButton)  
        val delete = itemView.findViewById<ImageButton>(R.id.delete)  
  
        delete.visibility = if (item.isConfigurable) {  
            View.VISIBLE  
        } else {  
            View.GONE  
        }  
  
        textView.text = item.title  
        switchButton.isChecked = item.isAvailable  
        switchButton.setOnCheckedChangeListener { _, _ ->  
            item.isAvailable = !item.isAvailable  
            onSwitchClicked.invoke(item)  
        }  
        delete.setOnClickListener { it: View! ->  
            onDeleteClicked.invoke(item)  
        }  
    }  
}
```

Εικόνα 41: Κώδικας RecyclerView.

8. Συμπεράσματα και Μελλοντικές Προτάσεις.

Στο κεφάλαιο αυτό θα περιγράψουμε τα συμπεράσματα που προέκυψαν από την υλοποίηση της παρούσας διπλωματικής εργασίας. Θα αναφέρουμε επίσης και μελλοντικές επεκτάσεις που θα κάνουμε σε αυτήν την εφαρμογή.

8.1 Συμπεράσματα.

Έχοντας ολοκληρώσει την ανάπτυξη της διπλωματικής εργασίας μου, είμαι σε θέση να βγάλω διάφορα χρήσιμα συμπεράσματα. Στη παρούσα διπλωματική εργασία σχεδιάστηκε και υλοποιήθηκε μια Android εφαρμογή για έξυπνες συσκευές (τηλέφωνα, tablet, POS, κτλπ) που απευθύνετε σε επαγγελματίες του χώρου της εστίασης. Αρχικά η εφαρμογή καταφέρνει τον σκοπό της χωρίς να έχουν παρατηρηθεί bug ή κάποια καθυστέρηση στην απόδοση της. Η γρήγορη απόκριση της, το εύρος των δυνατοτήτων καθώς και η σχεδίαση της έχει αποσπάσει θετικά σχόλια από όποιον την έχει χρησιμοποιήσει που δουλεύουν στον χώρο της εστίασης είτε σαν σερβιτόροι είτε στην κουζίνα και λαμβάνουν παραγγελίες.

Από την έρευνα των απαιτήσεων, οι πρώτες γραμμές κώδικα και τον σχεδιασμό της εφαρμογής, μέχρι και την ολοκλήρωσή της μπόρεσε να μας δώσει μια συνολική εικόνα για την ανάπτυξη εφαρμογών σε Android λειτουργικό. Ο προγραμματισμός σε Android λειτουργικό είναι μια ευχάριστη πρόκληση που δίνει πολλές δυνατότητες και επιλογές στον προγραμματιστή, του δίνει την-αίσθηση της-δημιουργίας και τον αναγκάζει να χρησιμοποιήσει διαφορετικά πολύ ενδιαφέροντα εργαλεία. Από την άλλη όμως ο προγραμματιστής έρχεται αντιμέτωπος με αρκετά προβλήματα-που προκύπτουν κατά την ανάπτυξη. Συγκεκριμένα αν και φαινομενικά η συγκεκριμένη εφαρμογή δείχνει σχετικά απλή, στην πράξη υπήρξαν διάφορα τεχνικά προβλήματα.

Η μεγαλύτερη δυσκολία προέρχεται από το γεγονός ότι η Kotlin είναι μια σχετικά καινούργια γλώσσα προγραμματισμού με αποτέλεσμα το documentation να είναι-ελλιπές, τα σχετικά forum είναι περιορισμένα και τα διαθέσιμα libraries λίγα σχετικά με αυτά που είναι διαθέσιμα στην Java στην οποία αρχικά σχεδιάστηκε να λειτουργούν οι εφαρμογές του Android.

8.2 Μελλοντικές Προτάσεις Βελτίωσης της Εφαρμογής.

Στις μέρες μας που ανάπτυξη των έξυπνων συσκευών και των υπηρεσιών που προσφέρουν είναι ραγδαία. Με κάθε καινούργια έκδοση Android οι εφαρμογές έχουν επιπλέον δυνατότητες και για να ανταπεξέλθει στον ανταγωνισμό κάθε εφαρμογή πρέπει να αναβαθμίζεται και να βελτιώνεται. Έτσι και οι δυνατότητες της εφαρμογής που υλοποιήσαμε μπορούν να επεκταθούν μελλοντικά προς αρκετές κατευθύνσεις.

Μερικές ιδέες για μελλοντική επέκταση της εφαρμογής παρουσιάζονται παρακάτω:

- Μετάβαση σε αρχιτεκτονική Material Design 3, χρησιμοποιώντας έτσι την πλέον σύγχρονη αρχιτεκτονική την οποία προτείνει και η Google
- Διόρθωση οποιονδήποτε bug προκύψουν
- Βελτιστοποίηση των διεπαφών ως προς το γραφικό περιβάλλον (UI) και την εμπειρίαχρήσης (UX).
- Χρήση του Jetpack Compose για δημιουργία οθονών.
- Προσθήκη επιλογής γλώσσας.
- Προσθήκη Analytics που θα μπορεί ο χρήστης να βλέπει ανά εβδομάδα ποια πιάτα έχουν μεγαλύτερη ζήτηση και τη τζίρο ειχε.

9. Βιβλιογραφία

1. *Creating Dynamic UI Android Fragments*, Jim Wilson
2. *The Busy Coder's Guide to Android Development*, Mark L. Murphy
3. <https://www.maxmag.gr/tecnologia/android-vs-ios-i-aionia-machi-leitoyrgikon-systimaton/>
4. https://dev.to/creativetim_official/material-design-1-vs-material-design-2-he3?fbclid=IwAR1hmAOLxEeTGFgrhk5jrI23GNMsdLjKB_Hk8WnCSvSJEIDTguFJ7RDivKA
5. <https://guides.codepath.com/android/using-the-recyclerview>
6. [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))
7. https://www.diffen.com/difference/Android_vs_iOS
8. <https://www.mobiloud.com/blog/native-web-or-hybrid-apps/>
9. <https://en.wikipedia.org/wiki/IOS>