



**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη εφαρμογών σε κινητές συσκευές

Φίλιππος Σίμπφ

Επιβλέπων καθηγητής: Τζήμας Γιάννης

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Πάτρα, Ημερομηνία

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Ονοματεπώνυμο, Υπογραφή
2. Ονοματεπώνυμο, Υπογραφή
3. Ονοματεπώνυμο, Υπογραφή

Ευχαριστίες

Ευχαριστώ όσους μου έδωσαν τα ερεθίσματα που με βοήθησαν και με βοηθούν στην προσωπική εξέλιξη των ικανοτήτων , της αντίληψης, των συναισθημάτων μου, είτε αυτοί είναι καθηγητές και συμφοιτητές που πορευθήκαν μαζί μου τα χρόνια των σπουδών μου, είτε είναι φίλοι και οικογένεια που πορεύονται μαζί μου πάντοτέ.

Επίσης, θέλω να ευχαριστήσω ξεχωριστά τον καθηγητή μου κ. Γιάννη Τζήμα για την επίβλεψη της παρούσας πτυχιακής εργασίας και τον φίλο Στέφανο Κόκκαλη για τίς πολύτιμες συμβουλές του.

Ιδικές Ευχαριστίες στην οικογένεια μου που με στηρίζει και με υπομένει όλα αυτά τα χρόνια με κάθε δυνατό μέσο.

Περιεχόμενα

Αφιέρωση.....	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Ευχαριστίες.....	2
Περιεχόμενα	3
Λίστα Σχημάτων	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Λίστα Πινάκων	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Λίστα Εικόνων.....	5
1 Εισαγωγή.....	10
1.1 Κινητές συσκευές.....	10
1.2 Εφαρμογές κινητών συσκευών.....	13
2 Ανάπτυξη εφαρμογών για κινητές συσκευές.....	16
2.1 Native εφαρμογές (Native apps)	17
2.2 Ανεξαρτήτως πλατφόρμας εφαρμογές (cross-platform apps).....	19
2.2.1 Progressive Web apps.....	20
2.2.2 Hybrid apps	21
2.2.3 Interpreted apps	23
2.2.4 Model-driven software development.....	25
2.2.5 Cross-compiled apps.....	26
3 Επικρατέστερες τεχνολογίες.....	28
4 Η εφαρμογή μου (G.D.T.W. / Go Discover The World)	34
4.1 Εισαγωγή.....	34
4.2 Λειτουργικότητα	34
4.2.1 Login/Register	35
4.2.2 Χάρτης.....	36
4.2.3 NPC dialog	42
4.3 Λίγα λόγια για το Android.....	45
4.4 Περιβάλλον ανάπτυξης - IDE (Android Studio).....	46
4.4.1 Project structure	47
4.4.2 Διεπαφή χρήστη.....	49
4.4.3 Gradle build system	50

4.5	Βάση δεδομένων	50
4.5.1	XAMPP server.....	50
4.5.2	phpMyAdmin	51
4.5.3	MYSQL/PHP connection.....	54
4.6	APIS - application programming interface.....	58
4.6.1	Πώς λειτουργούν τα APIS.....	58
4.6.2	Google Maps Platform	59
4.6.3	Maps SDK for Android.....	62
4.6.4	The Directions API.....	67
4.7	Λοιπές τεχνολογίες	68
4.7.1	Shared Preferences	68
4.7.2	Volley.....	68
4.7.3	Alert Dialog.....	69
4.7.4	Array Adapter.....	70
4.7.5	JSON	71
4.7.6	Connectivity Manager	72
4.8	Μελλοντικές επεκτάσεις.....	73
5	Λίστα RFCs.....	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
6	Βιβλιογραφία	74

Λίστα Εικόνων

Εικόνα 1: xiaomi smartphone (<i>mistore-greece.gr</i>)	11
Εικόνα 2: xiaomi tablet (<i>mistore-greece.gr</i>)	12
Εικόνα 3: smartphone-phablet-tablet comparison [3]	13
Εικόνα 4: Smartphone with apps	13
Εικόνα 5: mobile app marketplaces, Apple App Store and Google Play Store	14
Εικόνα 6: Smartphone users worldwide 2016 to 2021 [14]	15
Εικόνα 7: Most popular apps in 2020 [15]	16
Εικόνα 8: Native Mobile App [10]	18
Εικόνα 9: Native applications functionality illustration[5]	18
Εικόνα 10: Progressive web applications functionality illustration[5]	20
Εικόνα 11: Overview of the Progressive Web Apps approach build workflow. [4]	21
Εικόνα 12: Hybrid applications functionality illustration[5]	22
Εικόνα 13: Overview of the Hybrid Apps approach build workflow. [4]	23
Εικόνα 14: Interpreted applications functionality illustration[5]	24
Εικόνα 15: Overview of the Interpreted Apps approach build workflow. [4]	25
Εικόνα 16: Overview of the Model-Driven Apps approach build workflow. [4]	26
Εικόνα 17: Cross-Compiled applications functionality illustration[5]	27
Εικόνα 18: Overview of the Cross-Compiled Apps approach build workflow. [4]	28
Εικόνα 19: Smartphone shipment share worldwide Q1'18-Q2'21, by vendor. [14]	29
Εικόνα 20: Mobile Operating System Market Share Worldwide Sept 2020 - Sept 2021 [15]	30
Εικόνα 21 : Number of apps available in leading app stores as of 1st quarter 2021 [14]	31
Εικόνα 22: Cross-platform mobile frameworks used by software developers worldwide from 2019 to 2021 [14]	32
Εικόνα 23: React Native and Flutter's users based on GitHub stars [17]	33
Εικόνα 24: React Native and Flutter's users based on GitHub stars (chart) [17]	33
Εικόνα 25: App launch icon	35
Εικόνα 26: GDTW Login	35
Εικόνα 27: GDTW Register	36
Εικόνα 28: GDTW Map	37
Εικόνα 29: GDTW New person/place	38
Εικόνα 30: GDTW Low button pressed	39
Εικόνα 31: GDTW My places button pressed	40
Εικόνα 32: GDTW Delete NPC	41
Εικόνα 33: GDTW Clear my places	42
Εικόνα 34: GDTW Talk to	42
Εικόνα 35: NPC dialog	43
Εικόνα 36: GDTW NPC places map	44
Εικόνα 37: GDTW NPC places	44
Εικόνα 38: Android Studio [19]	47
Εικόνα 39: Android project view[19]	48
Εικόνα 40: Κύριο παράθυρο του Android Studio[19]	49

Εικόνα 41: XAMPP control panel	51
Εικόνα 42: XAMPP control panel, start Apache and MySQL.....	52
Εικόνα 43: phpMyAdmin, create new database.....	52
Εικόνα 44: phpMyAdmin, create new table	53
Εικόνα 45: phpMyAdmin, create database columns	54
Εικόνα 46: Google Cloud Console, project selector page [22].....	60
Εικόνα 47: Google Cloud Console, Maps API Library page [22].....	61
Εικόνα 48: Google Cloud Console, Credentials page [22].....	62
Εικόνα 49: Android Studio, add Google Maps Activity	63
Εικόνα 50: Android Studio, google_maps_api.xml	64
Εικόνα 51: Android Studio, google map	67

Πρόλογος

Η παρούσα διπλωματική εργασία εκπονήθηκε στα πλαίσια του Προπτυχιακού προγράμματος σπουδών του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών πληροφορικής του Πανεπιστημίου Πελοποννήσου. Σκοπός της είναι η καταγραφή των διαφορετικών αρχιτεκτονικών προσεγγίσεων στην ανάπτυξη εφαρμογών σε κινητές συσκευές και τεχνολογιών που χρησιμοποιούνται, όπως και μία καταγραφή της αγοράς του σήμερα.

Επιπλέον στα πλαίσια της διπλωματικής αναπτύχθηκε μία native Android εφαρμογή για κινητές συσκευές με την γλώσσα προγραμματισμού Java ως αναφορά στην native αρχιτεκτονική προσέγγιση.

Περίληψη

Στην παρούσα Διπλωματική εργασία παρουσιάζονται οι διαφορετικές αρχιτεκτονικές προσεγγίσεις αλλά και τεχνολογίες που χρησιμοποιούνται για την ανάπτυξη εφαρμογών σε κινητές συσκευές. Αναλύονται Native και Cross Platform (ανεξαρτήτως πλατφόρμας) προσεγγίσεις που περιλαμβάνουν τις native εφαρμογές, τα Progressive Web Apps, τα Hybrid Apps, τα Interpreted Apps και τα Cross Compiled Apps. Ακόμα γίνεται μία καταγραφή της αγοράς για smartphone, αγορές εφαρμογών για κινητές συσκευές, λειτουργικά συστήματα που χρησιμοποιούν οι συσκευές αυτές και τεχνολογίες ανάπτυξης που επικρατούν.

Επιπρόσθετα παρουσιάζεται μία native Android εφαρμογή που αναπτύχθηκε στα πλαίσια της εργασίας με μία περιγραφή των τεχνολογιών που χρησιμοποιήθηκαν για αυτόν τον σκοπό. Αναφέρονται οι συσκευές που υποστηρίζουν την εφαρμογή που αναπτύχθηκε, η λειτουργικότητα της εφαρμογής, ο τρόπος και οι τεχνολογίες για την σύνδεση της εφαρμογής με μία βάση δεδομένων, βιβλιοθήκες κώδικα, και λεπτομερής περιγραφή των εργαλείων που χρησιμοποιήθηκαν.

Λέξεις κλειδιά: Native, Cross platform, mobile app, mobile device, architecture, development approach

Abstract

This thesis presents the different architectural approaches and technologies used to develop applications on mobile devices. Native and Cross Platform approaches are analyzed that include native applications, Progressive Web Apps, Hybrid Apps, Interpreted Apps and Cross Compiled Apps. There is also a record of the market for smartphones, mobile application markets, operating systems used by these devices and development technologies that prevail.

In addition, a native Android application developed as part of the work that is presented with a description of the technologies used for this purpose. List the devices that support the application developed, the functionality of the application, technologies to connect the application to a database, code libraries, and a detailed description of the tools used.

Keywords: Native, Cross platform, mobile app, mobile device, architecture, development approach

1 Εισαγωγή

1.1 Κινητές συσκευές

Smartphone

Smartphone ονομάζουμε το κινητό τηλέφωνο το οποίο μπορεί να προσφέρει λειτουργίες ενός υπολογιστή κι όχι μόνο τις επικοινωνιακές δυνατότητες που προσφέρει ένα συμβατικό κινητό τηλέφωνο. Τα σύγχρονα smartphone προσφέρουν πολλαπλές λειτουργίες ενός υπολογιστή πάνω στο τηλέφωνο, έχουν πρόσβαση στο internet, κι ένα λειτουργικό σύστημα ικανό να εγκατασταθούν εφαρμογές σε αυτό, που θα διευρύνουν ακόμη περισσότερο τις λειτουργίες του.

Ενδεικτικά κάποιες από τις λειτουργίες του και τα χαρακτηριστικά του είναι η υποστήριξη 3G/4G σύνδεσης, δυνατότητα αναπαραγωγής πολυμέσων, GPS (Παγκόσμιο Σύστημα Εντοπισμού) ή A-GPS (Υποβοηθούμενο - Παγκόσμιο Σύστημα Εντοπισμού), ψηφιακή πυξίδα, κάμερα με δυνατότητα βιντεοληψίας, Bluetooth, λειτουργίες αφής, accelerometer (Επιταχυνσιόμετρο), γυροσκόπιο κ.α. ανάλογα την συσκευή.

Τα ποιο διαδεδομένα λειτουργικά συστήματα (Operating Systems) που χρησιμοποιούν τα smartphones είναι :

- Apple iOS
- RIM – Blackberry
- Google Android
- Microsoft Windows Phone
- Nokia Symbian

Τέτοια λειτουργικά συστήματα μπορούν να προσαρμοστούν με πολλά διαφορετικά μοντέλα τηλεφώνου. Επίσης, κάθε συσκευή μπορεί να έχει πολλαπλά λειτουργικά συστήματα ή εκδόσεις λειτουργικών συστημάτων εγκατεστημένες κατά την διάρκεια ζωής του.[1][2]



Εικόνα 1: xiaomi smartphone (mistore-greece.gr)

Tablet

Οι υπολογιστές tablet, που ονομάζονται επίσης tablet, είναι φορητές συσκευές που ταξινομούνται μεταξύ φορητών υπολογιστών και έξυπνων τηλεφώνων (smartphone). Το tablet λειτουργεί κυρίως αγγίζοντας την οθόνη, δεν τοποθετείται φυσικό πληκτρολόγιο αλλά μπορούν να συνδεθούν σε ασύρματο ή πληκτρολόγιο USB, χρησιμοποιεί ένα εικονικό πληκτρολόγιο στην οθόνη και συχνά ένα παθητικό στυλό γραφίδας.

Κάποιες τυπικές λειτουργίες των υπολογιστών tablet είναι η σύνδεση στο διαδίκτυο (χρησιμοποιώντας 2G, 3G, 4G ή WiFi), πιθανές λειτουργίες κινητού τηλεφώνου (μηνύματα, κλήσεις, βιντεοκλήσεις), δορυφορική πλοήγηση GPS, λειτουργίες κάμερας με δυνατότητα βιντεοληψίας, προβολή και επεξεργασία φωτογραφιών και βίντεο, ανάγνωση ηλεκτρονικού βιβλίου (συμπεριλαμβανομένων ηλεκτρονικών εκδόσεων περιοδικών), λήψη εφαρμογών (παιχνίδια, εκπαίδευση, βοηθητικά προγράμματα), δυνατότητα αναπαραγωγής πολυμέσων, Bluetooth, λειτουργίες αφής, accelerometer (Επιταχυνσιόμετρο), γυροσκόπιο κ.α. ανάλογα την συσκευή. [1][2]

Τα ποια διαδεδομένα λειτουργικά συστήματα (Operating Systems) που χρησιμοποιούν τα tablet είναι :

- Apple iOS
- Google Android
- Microsoft Windows RT



Εικόνα 2: xiaomi tablet (mistore-greece.gr)

Phablet

Τα phablets μπορούν να θεωρηθούν κάτι ενδιάμεσο σε smartphone και υπολογιστή tablet, παρουσιάζουν χαρακτηριστικά όμοια με εκείνα ενός smartphone όμως συχνά απουσιάζει η δυνατότητα τηλεφωνικών κλήσεων και αποστολής SMS. Το μέγεθος είναι συνήθως μεγαλύτερο από εκείνο των smartphones και μικρότερο από αυτό των tablet. Οι υπόλοιπες δυνατότητές τους είναι ίδιες με εκείνες ενός smartphone. [2][3]

Τα ποια διαδεδομένα λειτουργικά συστήματα (Operating Systems) που χρησιμοποιούν τα phablets παρομοίως με τα smartphones είναι :

- Apple iOS
- RIM – Blackberry
- Google Android
- Microsoft Windows Phone
- Nokia Symbian



Εικόνα 3: smartphone-phablet-tablet comparison [3]

1.2 Εφαρμογές κινητών συσκευών

Μια εφαρμογή για κινητά (mobile application), η οποία συνήθως αναφέρεται ως app, είναι ένας τύπος λογισμικού εφαρμογών που έχει σχεδιαστεί για να λειτουργεί σε κινητή συσκευή, όπως smartphone ή υπολογιστή tablet. Οι εφαρμογές για κινητά συχνά χρησιμεύουν για να παρέχουν στους χρήστες παρόμοιες υπηρεσίες με αυτές που έχουν πρόσβαση από υπολογιστές. Οι εφαρμογές είναι γενικά μικρές, μεμονωμένες μονάδες λογισμικού με περιορισμένη λειτουργία. Αυτή η χρήση λογισμικού εφαρμογών συστήθηκε αρχικά από την Apple Inc. και το App Store της[11].



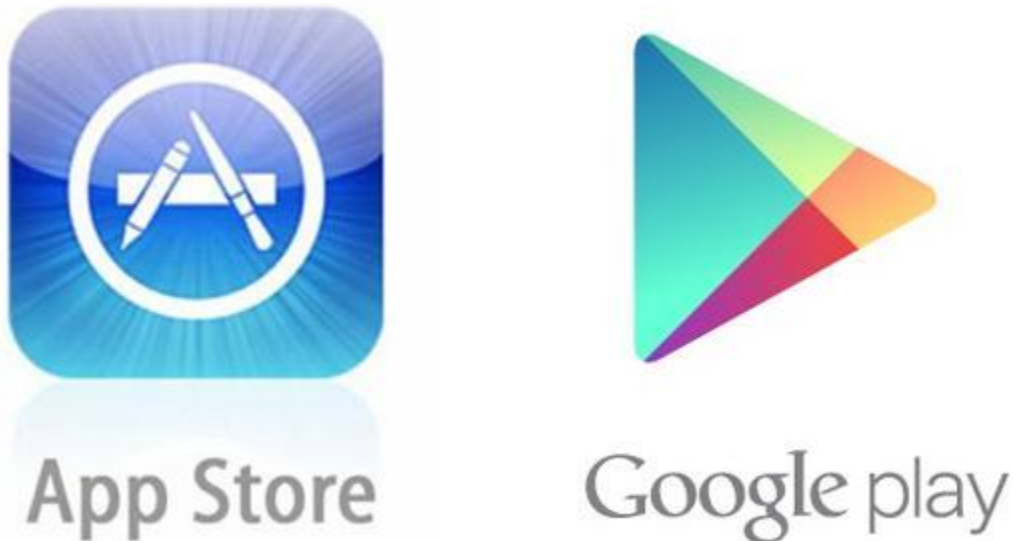
Εικόνα 4: Smartphone with apps

Οι εφαρμογές για κινητά απέχουν από τα ολοκληρωμένα συστήματα λογισμικού που βρίσκονται γενικά σε υπολογιστές. Αντ' αυτού, κάθε εφαρμογή παρέχει περιορισμένες και μεμονωμένες λειτουργίες, όπως ένα παιχνίδι, μία αριθμομηχανή ή περιήγηση στον ιστό για κινητά. Παρόλο που οι εφαρμογές μπορεί να απέφυγαν την εκτέλεση πολλαπλών εργασιών λόγω των περιορισμένων πόρων υλικού των

πρώτων φορητών συσκευών, η ιδιαιτερότητά τους αποτελεί πλέον μέρος της επιθυμητότητάς τους, επειδή επιτρέπουν στους καταναλωτές να επιλέγουν τι είναι σε θέση να κάνουν οι συσκευές τους.

Οι απλούστερες εφαρμογές για κινητά λαμβάνουν εφαρμογές που βασίζονται σε υπολογιστή και τις μεταφέρουν σε φορητή συσκευή. Καθώς οι εφαρμογές για κινητά γίνονται πιο «στιβαρές», αυτή η τεχνική κάπου χάνει. Μια πιο εξελιγμένη προσέγγιση περιλαμβάνει την ανάπτυξη ειδικά για το κινητό περιβάλλον, αξιοποιώντας τόσο τους περιορισμούς όσο και τα πλεονεκτήματά του. Για παράδειγμα, οι εφαρμογές που χρησιμοποιούν λειτουργίες βάσει τοποθεσίας είναι εγγενώς χτισμένες από την αρχή με το βλέμμα στο κινητό δεδομένου ότι ο χρήστης δεν είναι συνδεδεμένος με μια τοποθεσία, όπως στον υπολογιστή[11].

Ως συνέπεια της κυκλοφορίας smartphones με δυνατότητα εγκατάστασης εφαρμογών, εφαρμογές λογισμικού για κινητά (apps) απολαμβάνουν τεράστια δημοτικότητα τόσο από τους developers όσο και από τους τελικούς χρήστες. Αυτό μπορεί να διαπιστωθεί ιδιαίτερα από την εκπληκτική αύξηση της λήψη εφαρμογών και από τις πωλήσεις των δύο κορυφαίων αγορών εφαρμογών για κινητά, το Google Play Store και το Apple App Store[7].



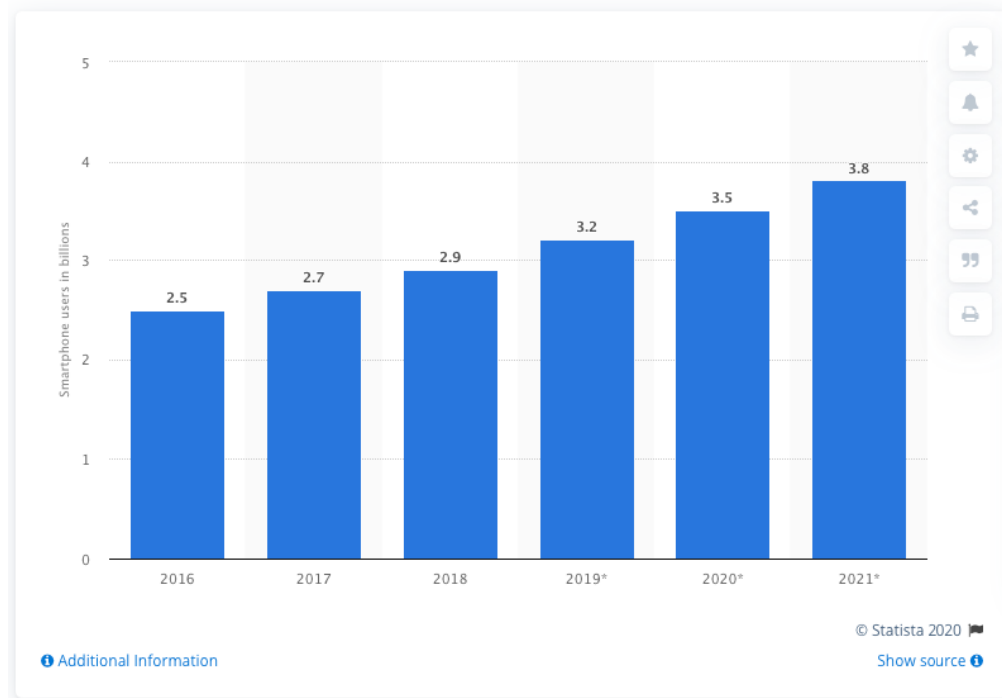
Εικόνα 5: mobile app marketplaces, Apple App Store and Google Play Store

Τα επόμενα έτη, το διπλοπώλιο και των δύο λειτουργικών συστημάτων αντιπροσωπεύει περισσότερα από το 99% των πωλήσεων smartphone το 2018 (Statista Inc 2018a). Τα δύο οικοσυστήματα συνδυαστικά δημιουργούν εκτιμώμενα έσοδα 58,6 δισεκατομμυρίων δολαρίων (USD) μόνο το 2017, αύξηση 30% σε σύγκριση με το προηγούμενο έτος. Περισσότερα από 5,8 εκατομμύρια ειδικά για smartphone εφαρμογές λογισμικού, είναι διαθέσιμες σε όλες τις πολυάριθμες αγορές εφαρμογών, ως μέρος της συνολικής εκτιμώμενης οικονομίας εφαρμογών που κυμαίνεται στα 6,3 τρισεκατομμύρια δολάρια.

Η κυκλοφορία εφαρμογών είναι μια εν εξελίξει διαδικασία, που θολώνει τις γραμμές μεταξύ smartphone και tablet, ηλεκτρονικών ειδών ευρείας κατανάλωσης, Internet-of-things (IoT) ακόμα και

αυτοκίνητων. Αυτό οφείλεται στην πανταχού παρούσα διαθεσιμότητα των smartphone και στον τεράστιο αριθμό χρηστών smartphone. Σύμφωνα με πρόσφατες αναφορές, 3,8 δισεκατομμύρια μοναδικοί χρήστες εκτιμάται ότι έχουν πρόσβαση σε ένα έξυπνο τηλέφωνο και αναμένεται περαιτέρω αύξηση των χρηστών (Statista Inc 2016). Για να είναι μέρος της οικονομίας εφαρμογών (οι ιδιώτες και εταιρείες), απαιτείται να είναι παρών δηλαδή να συμμετέχουν στα καταστήματα εφαρμογών για κινητά (mobile app stores) με την μορφή μίας εφαρμογής για κινητά (app)[7].

Number of smartphone users worldwide from 2016 to 2021 (in billions)



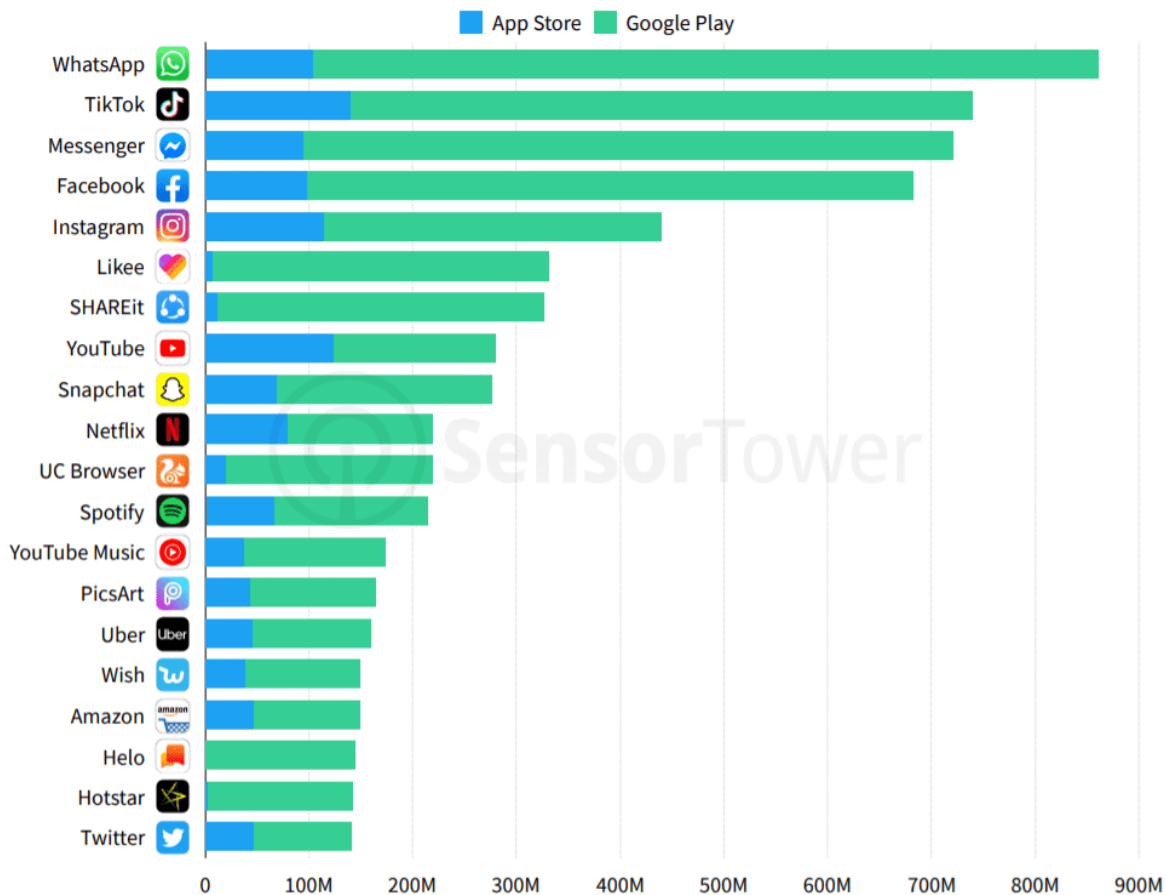
Εικόνα 6: Smartphone users worldwide 2016 to 2021 [14]

Υπάρχουν διάφοροι τύποι εφαρμογών που διατίθενται αυτήν τη στιγμή.

- **Εφαρμογές τυχερού παιχνιδιού:** Το ισοδύναμο των βιντεοπαιχνιδιών στον υπολογιστή, είναι από τους πιο δημοφιλείς τύπους εφαρμογών. Αποτελούν το ένα τρίτο όλων των λήψεων εφαρμογών και τα τρία τέταρτα όλων των καταναλωτικών δαπανών.
- **Εφαρμογές παραγωγικότητας:** Αυτές εστιάζουν στη βελτίωση της αποτελεσματικότητας των επιχειρήσεων διευκολύνοντας διάφορες εργασίες, όπως αποστολή μηνυμάτων ηλεκτρονικού ταχυδρομείου, παρακολούθηση της προόδου εργασίας, κράτηση ξενοδοχείων και πολλά άλλα.
- **Εφαρμογές τρόπου ζωής και ψυχαγωγίας:** Όλο και πιο δημοφιλείς, αυτές περιλαμβάνουν πολλές πτυχές του προσωπικού τρόπου ζωής και της κοινωνικοποίησης, όπως γνωριμίες,

επικοινωνία στα κοινωνικά μέσα, καθώς και κοινή χρήση (και παρακολούθηση) βίντεο. Μερικές από τις πιο ευρέως γνωστές εφαρμογές όπως το Netflix, το Facebook ή το TikTok ανήκουν σε αυτήν την κατηγορία.

Άλλοι τύποι εφαρμογών περιλαμβάνουν εφαρμογές κινητού εμπορίου (M-commerce) που χρησιμοποιούνται για την αγορά αγαθών στο διαδίκτυο, όπως το Amazon ή το eBay, εφαρμογές ταξιδιού που βοηθούν έναν ταξιδιώτη με πολλούς τρόπους (κράτηση περιηγήσεων και εισιτηρίων, εύρεση δρόμων μέσω χαρτών και γεωγραφικής τοποθεσίας, ημερολόγια ταξιδιού κ.λπ. .), και εφαρμογές κοινής ωφέλειας όπως εφαρμογές υγείας και σαρωτές γραμμωτού κώδικα [12].



Εικόνα 7: Most popular apps in 2020 [15]

2 Ανάπτυξη εφαρμογών για κινητές συσκευές

Η Ανάπτυξη εφαρμογών για κινητές συσκευές έρχεται με κάποιες προκλήσεις για τους προγραμματιστές. Οι πολλαπλές πλατφόρμες για κινητές συσκευές είναι μία από τις πιο δύσκολες πτυχές της ανάπτυξης κινητών συσκευών. Δεδομένου ότι οι κινητές συσκευές και οι πλατφόρμες κινούνται προς τον κατακερματισμό, η διαδικασία ανάπτυξης δεν μπορεί να αξιοποιήσει πληροφορίες και γνώσεις από μια πλατφόρμα σε άλλη πλατφόρμα. Όταν η «ίδια» εφαρμογή αναπτύσσεται για πολλές πλατφόρμες, οι προγραμματιστές αντιμετωπίζουν αυτήν την εφαρμογή για κινητά για κάθε πλατφόρμα ξεχωριστά και χειροκίνητα ώστε να βεβαιωθεί ότι η λειτουργικότητα διατηρείται σε πολλές πλατφόρμες και συσκευές.

- *Κατακερματισμός μεταξύ πλατφορμών:* Κάθε πλατφόρμα για κινητές συσκευές διαφέρει ανάλογα με τη διεπαφή χρήστη, την εμπειρία χρήστη, την αλληλεπίδραση ανθρώπου - υπολογιστή, προσδοκίες χρηστών, γλώσσες προγραμματισμού, API/SDK και υποστηριζόμενα εργαλεία.
- *Κατακερματισμός εντός της ίδιας πλατφόρμας:* Στην ίδια πλατφόρμα, υπάρχουν διάφορες συσκευές με διαφορετικές ιδιότητες, όπως μνήμη, ταχύτητα CPU και γραφική ανάλυση. Υπάρχει επίσης πιθανός κατακερματισμός στις εκδόσεις λειτουργικών συστημάτων. Ένα διάσημο παράδειγμα είναι ο κατακερματισμός σε συσκευές Android με διαφορετικά μεγέθη, αναλύσεις οθόνης και πολλές διαφορετικές εκδόσεις του λειτουργικού συστήματος Android.

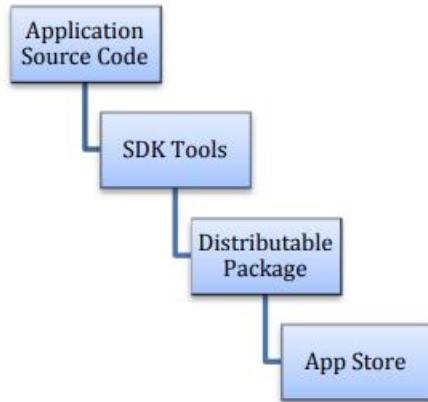
Επίσης, η δημιουργία ενός επαναχρησιμοποιήσιμου σχεδιασμού διεπαφής χρήστη για την εφαρμογή είναι μια αντιστάθμιση μεταξύ συνέπειας και τήρησης των προτύπων κάθε πλατφόρμας. Οι προγραμματιστές κινητών συσκευών χρειάζονται εργαλεία ανάλυσης που υποστηρίζονται από την κάθε πλατφόρμα για να παρακολουθούν τις εφαρμογές τους.

Ακόμα, οι δοκιμές είναι μια τεράστια πρόκληση προς το παρόν. Οι περισσότεροι προγραμματιστές δοκιμάζουν χειροκίνητα τις εφαρμογές. Πλέον υπάρχει μεγαλύτερη υποστήριξη για Unit testing εντός της κοινότητας κινητών τεχνολογιών, ωστόσο τα τρέχοντα πλαίσια δοκιμών δεν παρέχουν το ίδιο επίπεδο υποστήριξης για διαφορετικές πλατφόρμες.

Επιπλέον, οι περισσότεροι αναπτυσσόμενοι φορείς πιστεύουν ότι τα τρέχοντα εργαλεία δοκιμών είναι αδύναμα και έχουν περιορισμένη υποστήριξη για σημαντικά χαρακτηριστικά των δοκιμών σε κινητά όπως η κινητικότητα (π.χ. αλλαγή της συνδεσιμότητας δικτύου), υπηρεσίες τοποθεσίας, αισθητήρες ή διαφορετικές χειρονομίες χρηστών και διαφορετικοί τρόποι εισόδου τις πληροφορίες. Τέλος, φαίνεται ότι εξομοιωτές παρουσιάζουν έλλειψη αρκετών πραγματικών χαρακτηριστικών των φορητών συσκευών, γεγονός που κάνει ανάλυση και δοκιμές, ακόμη περισσότερο, προκλητικό[13].

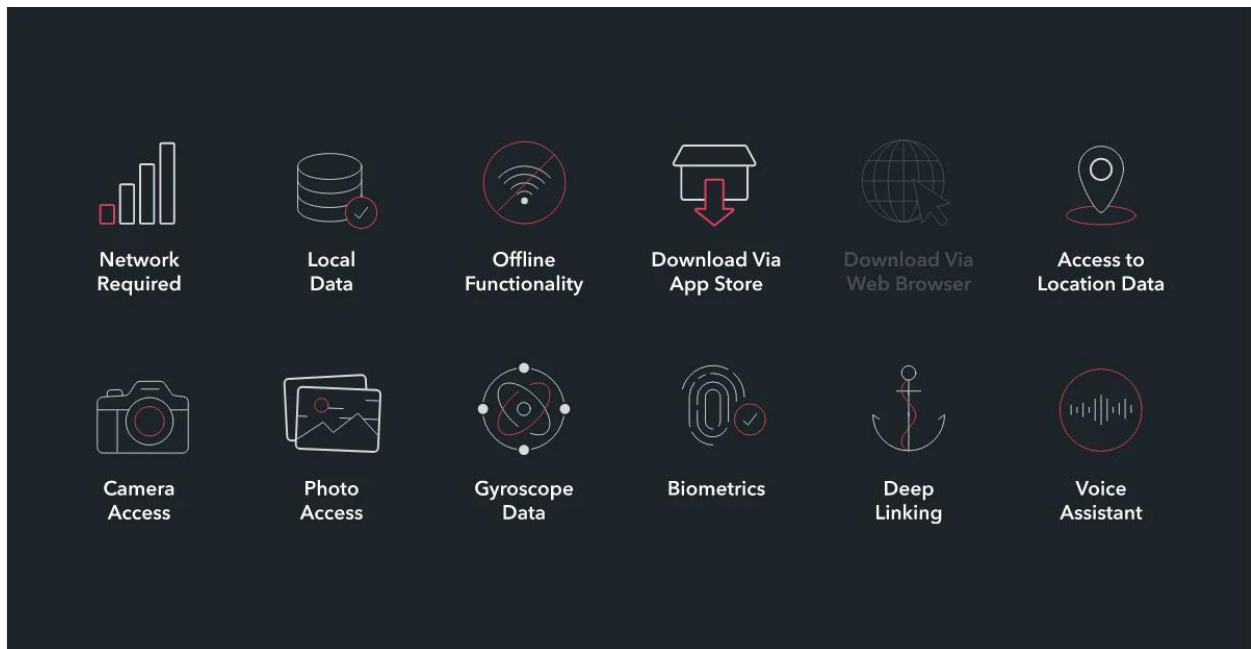
2.1 Native εφαρμογές (Native apps)

Παραδοσιακά, εφαρμογές για κινητά smartphone έχουν αναπτυχθεί χρησιμοποιώντας εργαλεία σχεδιασμένο για ετερογενή ανάπτυξη συγκεκριμένης πλατφόρμας. Αυτός ο τύπος ανάπτυξης αναφέρεται συνήθως ως προσέγγιση της εγγενούς ανάπτυξης (native app development), η οποία γίνεται με τη χρήση εγγενούς περιβάλλοντος ανάπτυξης, kit ανάπτυξης λογισμικού (SDK) και γλώσσες προγραμματισμού που είναι εγγενείς στη πλατφόρμα προορισμού[8].



Εικόνα 8: Native Mobile App [10]

Οι εγγενείς εφαρμογές μπορούν να έχουν ελεύθερη πρόσβαση σε όλα τα API και τις ενσωματωμένες λειτουργίες της συσκευής, όπως ο τηλεφωνητής, η κάμερα, τα SMS και οι υπηρεσίες τοποθεσίας. Ενώ η εγγενής ανάπτυξη εφαρμογών για κινητές συσκευές απαιτεί βαθιά γνώση της συγκεκριμένης πλατφόρμας στην οποία ανήκει η εκάστοτε συσκευή. Αυτή η στρατηγική (native) παρέχει εμπειρία χρήστη υψηλότερης ποιότητας από άλλες μεθόδους ανάπτυξης εφαρμογών για κινητά [10]. Οι εγγενείς εφαρμογές είναι συνήθως γραμμένες στο περιβάλλον Xcode με χρήση Swift/Objective-C για ανάπτυξη iOS και στο περιβάλλον Android Studio με χρήση Java/Kotlin για Android[8].



Εικόνα 9: Native applications functionality illustration[5]

Με πολλαπλές μεγάλες πλατφόρμες στην αγορά, συγκεκριμένα, αυτήν της iOS της Apple, του Android της Google, και των Windows 10 Mobile της Microsoft και ενώ η κινητή τεχνολογία αναπτύσσεται

ραγδαία ένα από τα πιο σημαντικά μειονεκτήματα της ανάπτυξης εγγενών εφαρμογών είναι ότι ο πηγαίος κώδικας που γράφτηκε για μια πλατφόρμα για κινητά δεν μπορεί να χρησιμοποιηθεί σε άλλη. Έτσι η ανάπτυξη native εφαρμογών υποφέρει από αναποτελεσματικότητα χρόνου και κόστους, λόγω διπλών προσπαθειών για ανάπτυξη και συντήρηση εγγενών εφαρμογών για κάθε πλατφόρμα. Παρ' όλα αυτά, από τις περισσότερες από 5,6 εκατομμύρια εφαρμογές διατίθεται σε όλα τα καταστήματα εφαρμογών, η απόλυτη πλειοψηφία αναπτύχθηκαν χρησιμοποιώντας αυτήν την ακριβή προσέγγιση ανάπτυξης[10].

Αυτό συμβαίνει για διάφορους λόγους όπως ότι οι εγγενείς εφαρμογές για κινητά αλληλοεπιδρούν άμεσα με εγγενή API χωρίς να εξαρτώνται από το ενδιάμεσο λογισμικό, όπως plugins και WebViews. Δεδομένου ότι υπάρχουν λιγότερες εξαρτήσεις, οι εγγενείς εφαρμογές για κινητά είναι πιο γρήγορες από τις cross-platform εφαρμογές. Αυτό είναι ιδιαίτερα σημαντικό για εφαρμογές με επίκεντρο την απόδοση, όπως παιχνίδια και εφαρμογές με βαριά γραφικά.

Καθώς οι εγγενείς εφαρμογές για κινητά αναπτύσσονται με χρήση εγγενών SDK (kit ανάπτυξης λογισμικού), οι διεπαφές χρήστη τους φαίνονται ταιριαστές με την πλατφόρμα στην οποία ανήκουν. Αυτό εξασφαλίζει καλύτερη εμπειρία χρήστη, καθώς δεν υπάρχουν αποκλίσεις μεταξύ του λειτουργικού συστήματος και του σχεδιασμού της εφαρμογής.

Οι εγγενείς εφαρμογές για κινητά μπορούν να έχουν άμεση πρόσβαση στις πιο πρόσφατες λειτουργίες iOS ή Android. Καθώς οι cross-platform τεχνολογίες δεν μπορούν να χρησιμοποιήσουν απευθείας εγγενή API, οι εφαρμογές cross-platform πρέπει να περιμένουν έως ότου υπάρξει ένα plugin που να υποστηρίζει τη νέα λειτουργία.

Λόγω της αρχιτεκτονικής τους, οι εγγενείς εφαρμογές για κινητά συμμορφώνονται καλύτερα με τις οδηγίες του καταστήματος εφαρμογών. Το 2017, η Apple περιόρισε τις οδηγίες υποβολής της. Έκτοτε, έχουν αρχίσει να απορρίπτονται εφαρμογές που βασίζονται πάρα πολύ σε WebViews, όπως το Ionic View που επέτρεψε στους προγραμματιστές να δοκιμάσουν τις Ionic εφαρμογές τους. Καθώς είναι πιθανό ότι τα καταστήματα εφαρμογών θα συνεχίσουν να πατάνε τις υβριδικές εφαρμογές, οι εγγενείς εφαρμογές για κινητά είναι μια επένδυση πιο μελλοντική[9].

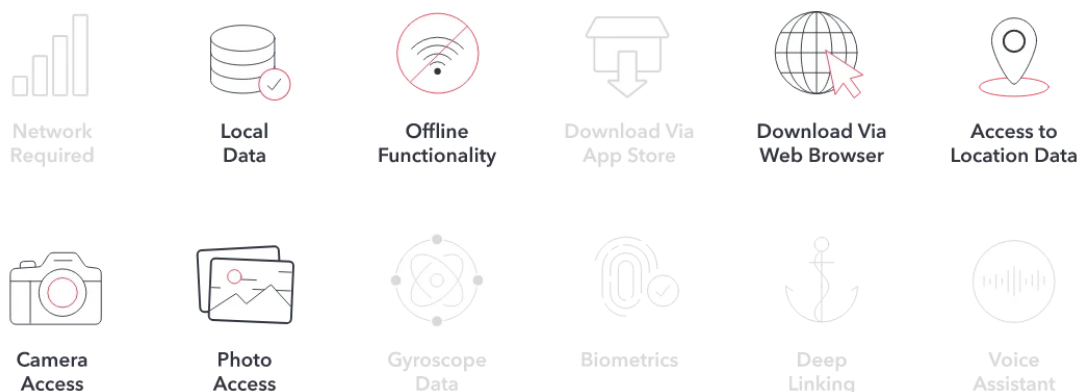
2.2 Ανεξαρτήτως πλατφόρμας εφαρμογές (cross-platform apps)

Λόγω του κόστους και των απαιτήσεων γνώσεων που σχετίζονται με τη native προσέγγιση ανάπτυξης εφαρμογών, είναι διαθέσιμες πολλές εναλλακτικές προσεγγίσεις. Αυτές αναφέρονται συνήθως ως cross-platform mobile development, ένας όρος-ομπρέλα που καλύπτει ένα ευρύ φάσμα εννοιολογικών προσεγγίσεων ανάπτυξης και τεχνικά πλαίσια για την απλοποίηση της δημιουργίας εφαρμογών. Συνήθως, μια μεμονωμένη βάση κώδικα μπορεί να χρησιμοποιηθεί για τον καθορισμό ή τη δημιουργία εφαρμογών που μπορούν να αναπτυχθούν σε διάφορες πλατφόρμες με ελάχιστες έως καθόλου τροποποιήσεις για την εκάστοτε πλατφόρμα, αν και το επίπεδο κοινής χρήσης κώδικα σε κάθε πλατφόρμα διαφέρει μεταξύ frameworks και προσεγγίσεων ανάπτυξης. Ωστόσο, ενώ τα frameworks ανάπτυξης μεταξύ πλατφορμών μπορούν να βοηθήσουν στην ανάπτυξη εφαρμογών που εκτελούνται σε πολλές πλατφόρμες, ορισμένες μελέτες περιπτώσεων δείχνουν ότι εταιρείες και επαγγελματίες του κλάδου αξιοποιούν frameworks πολλαπλών πλατφορμών για την ανάπτυξη εφαρμογών μιας πλατφόρμας. Μια τέτοια πρόσφατη μελέτη περίπτωσης περιγράφει την διαδικτυακή πλατφόρμα

τέχνης «Artsy» να χρησιμοποιεί το framework React Native του Facebook για να γράφει εφαρμογές για κινητά χρησιμοποιώντας JavaScript, καθώς εκεί η εσωτερική τεχνική τους ικανότητα ήταν ισχυρή σε αυτήν την τεχνολογία. Μια ακόμη πιο πρόσφατη μελέτη από την online ταξιδιωτική πλατφόρμα «Townске» περιγράφει την ίδια κατάσταση: ανάπτυξη μιας πλατφόρμας χρησιμοποιώντας cross-platform framework , με την εμπειρία των προγραμματιστών και την εσωτερική γνώση της εταιρίας σε JavaScript να είναι βασικοί λόγοι επιλογής του έναντι της εγγενούς προσέγγισης [7].

2.2.1 Progressive Web apps

Τα Progressive Web Apps --ή για συντομία PWA-- αν και εξακολουθούν να στερούνται ορισμένων εγγενών (Native) χαρακτηριστικών άλλων προσεγγίσεων cross-platform (π.χ. πρόσβαση για τις περισσότερες, αν όχι για όλες τις δυνατότητες της συσκευής), κερδίζουν όλο και περισσότερο δημοτικότητα μεταξύ των επαγγελματιών από το 2016 .



Εικόνα 10: Progressive web applications functionality illustration[5]

Στον πυρήνα του, ένα PWA είναι μία εφαρμογή Ιστού με βελτιωμένες δυνατότητες.

Ενώ φιλοξενείται από κάποιον διακομιστή web (web server) και οι χρήστες πρόσβαση αποκτούν πρόσβαση με της διεύθυνσης URL του ιστότοπου μέσω ενός προγράμματος περιήγησης, στόχος αυτής της νέας προσέγγισης είναι να επιτρέψει εφαρμογές ιστού να μοιάζουν και να έχουν την αίσθηση μίας συνηθισμένης Native ή cross-platform εφαρμογής.

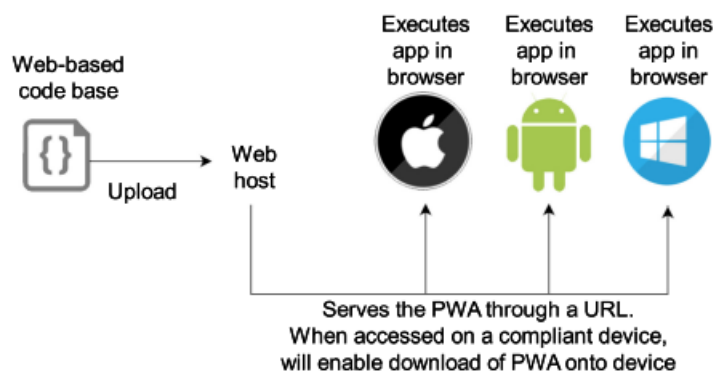
Λόγω του ότι είναι βασισμένα στο web, η διεπαφή χρηστών των PWA μπορεί να σχεδιαστεί έτσι ώστε να μοιάζει και να συμπεριφέρεται σαν μία Native εφαρμογή χρησιμοποιώντας τις ίδιες μεθόδους όπως στην υβριδική προσέγγιση (κεφάλαιο 2.2.2), συγκεκριμένα μέσω της χρήσης HTML και CSS για δομή και στυλ.

Κατά την πρόσβαση σε έναν ιστότοπο με δυνατότητα PWA, ένα banner θα ζητήσει από τον χρήστη να την εγκατάσταση της ιστοσελίδας στο τηλέφωνό τους. Αυτό θα κατεβάσει όλα τα απαραίτητα στοιχεία, συμπεριλαμβανομένων αρχείων JavaScript, HTML και CSS, εικόνες και γραμματοσειρές που επιτρέπουν τη χρήση του ιστότοπου εκτός σύνδεσης. Το πλέον εκτός σύνδεσης PWA έχει επίσης προστεθεί στην αρχική οθόνη του χρήστη παρόμοια με οποιαδήποτε εφαρμογή που έχει ληφθεί από καταστήματα εφαρμογών.

Κατά την εκκίνηση του PWA από την αρχική οθόνη, ένα απλοποιημένο παράθυρο του προγράμματος περιήγησης θα ανοίξει τα στοιχεία που έχετε κατεβάσει προηγουμένως και θα διαβάσει ένα αρχείο manifest για να διαμορφώσει μια οθόνη εκτόξευσης, εικονίδια και χρώματα. Σε αυτό το πλαίσιο, απλοποιημένο παράθυρο του προγράμματος περιήγησης σημαίνει ότι δεν εμφανίζεται καμία γραμμή διευθύνσεων, εικονίδιο ρυθμίσεων ή κάτι παρόμοιο, μόνο το περιεχόμενο του ιστότοπου που σημαίνει ότι αν ο χρήστης δεν γνωρίζει τις έννοιες του PWA, δεν θα γνωρίζει ότι η εφαρμογή εκτελείται εντός ενός προγράμματος περιήγησης.

Αυτό αυξάνει σημαντικά την αίσθηση της Native εφαρμογής στην χρήση ενός PWA σε σύγκριση με έναν κανονικό ιστότοπο που περιηγείται με παραδοσιακό τρόπο.

Η επέκταση μιας κανονικής εφαρμογής ιστού σε PWA συνεπάγεται την ενσωμάτωση μίας υπηρεσίας εξυπηρέτησης (service worker), ένα αρχείο manifest που βασίζεται σε JSON και μια δέσμη στατικών στοιχείων διεπαφής χρήστη που δεν εξαρτώνται από το δυναμικό περιεχόμενο που αναφέρεται ως Application Shell. Ο σκοπός του service worker είναι ο έλεγχος και η διαχείριση του κύκλου ζωής της εφαρμογής, της επιχειρησιακής λογικής (business logic) για το συγχρονισμό δεδομένων και τον χειρισμό των ειδοποιήσεων push, όλα μέσω ενός script αρχείου γραμμένο σε JavaScript που είναι σε θέση να εκτελεστεί στο υπόβαθρο [4].



Εικόνα 11: Overview of the Progressive Web Apps approach build workflow. [4]

2.2.2 Hybrid apps

Αυτή η προσέγγιση είναι ένα μείγμα, εξ ου και το όνομα υβριδικό, τόσο Native όσο και Web τεχνολογιών. Όπου ο πυρήνας της εφαρμογής γράφεται χρησιμοποιώντας τεχνολογίες ιστού (HTML, CSS και JavaScript), οι οποίες στη συνέχεια ενσωματώνονται σε μια εγγενή εφαρμογή (Native app). Με

τη χρήση πρόσθετων, αυτές οι εφαρμογές μπορούν να έχουν πλήρη πρόσβαση στις δυνατότητες της κινητής συσκευής.



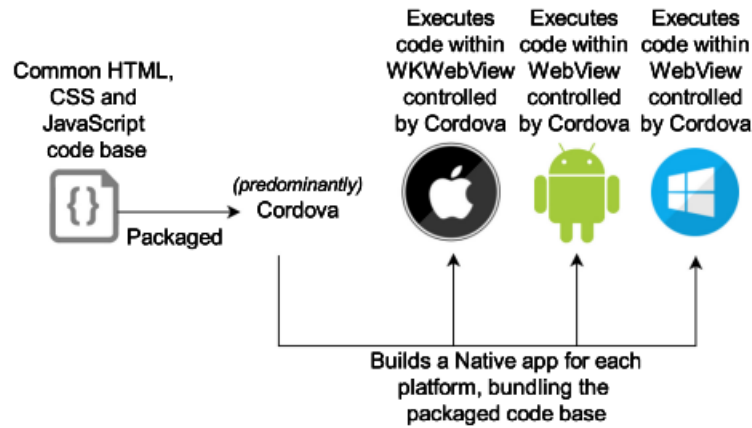
Εικόνα 12: Hybrid applications functionality illustration[5]

Για να κατανοήσουμε καλύτερα αυτήν την προσέγγιση, ας αναλύσουμε πώς ταιριάζουν όλα μαζί. Η καρδιά μιας υβριδικής εφαρμογής για κινητά εξακολουθεί να είναι μια εφαρμογή που είναι γραμμένη με HTML, CSS και JavaScript. Ωστόσο, αντί να εμφανίζεται η εφαρμογή στο πρόγραμμα περιήγησης του χρήστη, εκτελείται από μια native εφαρμογή και το δικό της ενσωματωμένο πρόγραμμα περιήγησης, το οποίο είναι ουσιαστικά αόρατο για τον χρήστη.

Για παράδειγμα, μια εφαρμογή iOS θα χρησιμοποιούσε το WKWebView για να εμφανίσει την εφαρμογή μας, ενώ στο Android θα χρησιμοποιούσε το στοιχείο WebView για να κάνει την ίδια λειτουργία. Αυτός ο κώδικας στη συνέχεια ενσωματώνεται σε ένα φυσικό περιτύλιγμα εφαρμογών χρησιμοποιώντας μια τεχνολογία όπως το Apache Cordova (επίσης γνωστό ως PhoneGap) ή το Ionic's Capacitor. Αυτές οι τεχνολογίες δημιουργούν μια native εφαρμογή κελύφους που είναι απλώς το στοιχείο προβολής ιστού της πλατφόρμας στο οποίο θα φορτώσει την εφαρμογή Ιστού σας. Αυτό δίνει τη δυνατότητα να δημιουργηθούν και να δημοσιευτούν πραγματικές native εφαρμογές που μπορούν να υποβληθούν σε κάθε ένα από τα καταστήματα εφαρμογών της πλατφόρμας προς πώληση.

Επιπλέον, τόσο η Cordova όσο και ο Capacitor διαθέτουν ένα σύστημα προσθηκών που σας επιτρέπει να επεκτείνετε πέρα από τους περιορισμούς του «προγράμματος περιήγησης» και να αποκτήσετε πρόσβαση στην πλήρη σειρά δυνατοτήτων της κινητής συσκευής ενός χρήστη. Έτσι, εάν θέλετε να χρησιμοποιήσετε το TouchID σε μια συσκευή iOS ως επιλογή σύνδεσης ή θέλετε να συνδεθείτε σε μια συσκευή Bluetooth, αυτό μπορεί να γίνει εύκολα εγκαθιστώντας ένα πρόσθετο. Αυτά τα πρόσθετα δημιουργούνται από ένα ευρύ φάσμα προγραμματιστών και πολλά υποστηρίζονται ενεργά. Το Ionic προσφέρει ακόμη και ένα πλήρες οικοσύστημα υποστηριζόμενων plugins ως μέρος της τεχνολογίας Enterprise. Έτσι, οι περιορισμοί μιας εφαρμογής μόνο web ξεπερνιούνται εύκολα, επιτρέποντας στην εφαρμογή σας να έχει ισοτιμία με τις εγγενείς εφαρμογές στις δυνατότητές τους. Ωστόσο, υπάρχουν

ορισμένα μειονεκτήματα σε αυτήν την επιλογή. Η βιβλιοθήκη UI πρέπει να δημιουργηθεί ξανά. Εδώ εμπλέκονται τεχνολογίες όπως το Ionic, το NativeScript, το Xamarin και άλλες. Όλες αυτές οι επιλογές παρέχουν ισχυρά συστατικά UI που μοιάζουν με τα αντίστοιχα των native εφαρμογών, δίνοντας μια πλήρη σειρά δομικών στοιχείων για την εφαρμογή [6]

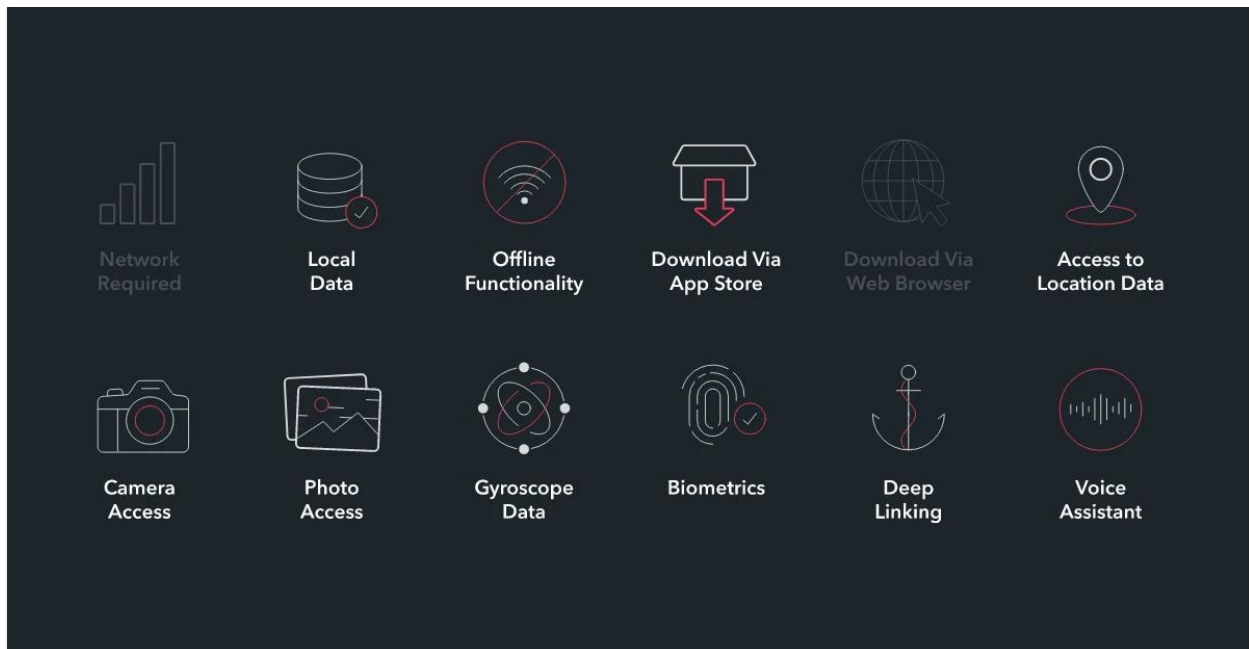


Εικόνα 13: Overview of the Hybrid Apps approach build workflow. [4]

2.2.3 Interpreted apps

Παρόμοια με την υβριδική προσέγγιση που περιεγράφηκε παραπάνω, έτσι και για την ερμηνευμένη προσέγγιση (interpreted approach) υπάρχουν frameworks που επιτρέπουν στους προγραμματιστές να δημιουργούν τις εφαρμογές τους χρησιμοποιώντας τη γλώσσα JavaScript, αν και η JavaScript δεν είναι έμφυτη γλώσσα της ανάπτυξης εφαρμογών με ερμηνευμένη προσέγγιση (interpreted apps).

Οι εφαρμογές που αναπτύσσονται με την προσέγγιση Interpreted διαφέρουν θεμελιωδώς από τις εφαρμογές Hybrid, καθώς οι εφαρμογές Interpreted δεν βασίζονται σε ένα στοιχείο WebView για την απόδοση ενός ομαδοποιημένου ιστότοπου. Αντ' αυτού, οι ερμηνευμένες εφαρμογές μπορούν να αποδώσουν στην οθόνη τα πραγματικά στοιχεία (components) native διεπαφών χρήστη, και όχι στοιχεία που βασίζονται σε HTML και CSS. Αυτό γίνεται εφικτό με τη χρήση ενός διερμηνευτή JavaScript στην εκάστοτε συσκευή, εξ ου και η ονομασία της προσέγγισης.

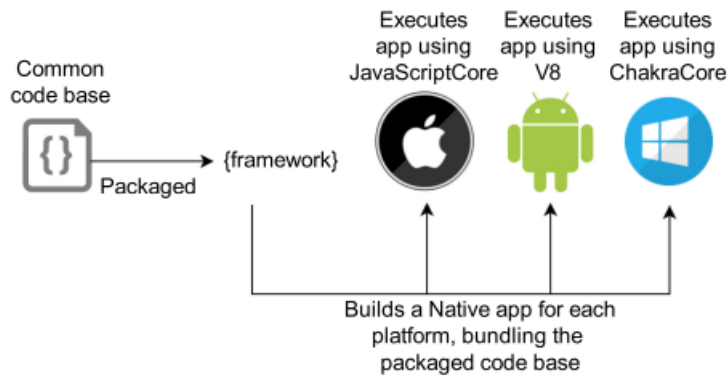


Εικόνα 14: Interpreted applications functionality illustration[5]

Όσον αφορά τους διερμηνείς κώδικα, το JavaScriptCore είναι ο προεπιλεγμένος διερμηνέας σε συσκευές iOS. Στις συσκευές Android, ο διερμηνέας που χρησιμοποιείται διαφέρει μεταξύ των frameworks, με το V8 JavaScript engine να χρησιμοποιείται συχνά. Ορισμένα frameworks της προσέγγισης Interpreted, όπως το Titanium Appcelerator, συσχετίζονται λανθασμένα με την προσέγγιση Cross-compiled. Ενώ και οι δύο προσεγγίσεις δημιουργούν εγγενείς διεπαφές χρήστη, η ερμηνευμένη προσέγγιση δεν μεταφράζει το codebase σε Native byte κώδικα, που είναι το πώς λειτουργεί η προσέγγιση Cross-compiled.

Για επικοινωνία μεταξύ του επιπέδου JavaScript και του επιπέδου native κώδικα, το οποίο έχει πρόσβαση στα εγγενές χαρακτηριστικά της συσκευής, η ερμηνευμένη προσέγγιση χρησιμοποιεί επίσης την τεχνική γεφύρωσης (Foreign Function Interfaces), παρόμοια με το πώς η υβριδική προσέγγιση επιτρέπει μια τέτοια πρόσβαση.

Μία ενδιαφέρων Σημείωση είναι ότι στα frameworks της ερμηνευμένης προσέγγισης δεν έχει ακόμη προσδιοριστεί μία κοινή βάση όπως για παράδειγμα στην υβριδική προσέγγιση. Έτσι, plugins ή modules για την εμπλουτισμό των δυνατοτήτων της JavaScript από τα εγγενή περιβάλλοντα που ανήκουν σε ένα framework ερμηνευμένης προσέγγισης δεν θα λειτουργούσαν σε άλλο framework της ίδιας προσέγγισης λόγω διαφορών στην εφαρμογή και γεφύρωση των APIs. Παραδείγματα τέτοιας μη λειτουργικότητας περιλαμβάνουν plugins σε frameworks όπως το React Native του Facebook και το NativeScript του Telerik. Ενώ και τα δύο ανήκουν την ίδια προσέγγιση ανάπτυξης, τα frameworks APIs είναι τόσο διαφορετικής φύσης που ένα plugin ή module που αναπτύχθηκε για το ένα framework δεν μπορεί να λειτουργήσει προς το παρόν στο άλλο, ενώ στην υβριδική ανάπτυξη, όλα τα frameworks που χτίζονται πάνω από το Cordova θα μπορούσαν θεωρητικά να χρησιμοποιούν τα ίδια Cordova modules [4].

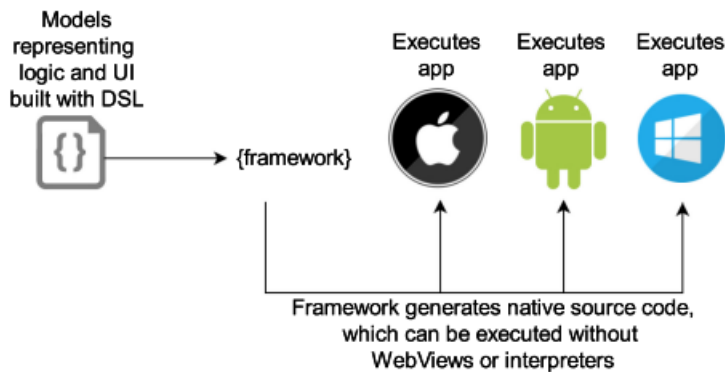


Εικόνα 15: Overview of the Interpreted Apps approach build workflow. [4]

2.2.4 Model-driven software development

Η model-driven προσέγγιση έχει χρησιμοποιηθεί για πολλά χρόνια στη ανάπτυξη λογισμικού με σκοπό τη διαχείριση της μεταβλητότητας. Επικεντρώνεται στο μοντέλο ως αφηρημένη αναπαράσταση (ενδεχομένως μέρους) ενός συστήματος από το οποίο παράγεται πραγματικό. Στο πλαίσιο της φορητής υπολογιστικής, αυτή η προσέγγιση επιτρέπει την ανάπτυξη εφαρμογών cross-platform χρησιμοποιώντας υψηλότερο βαθμό αφαιρετικότητας από τον πηγαίο κώδικα με τη χρήση γλωσσών ειδικού πεδίου (domain specific languages - DSLs) ή γλωσσών μοντελοποίησης γενικής χρήσης όπως η UML.

Στη συνέχεια, οι γεννήτριες κώδικα (μία ανά πλατφόρμα-στόχο) μετατρέπουν το αγνωστικιστικό μοντέλο πλατφόρμας σε συγκεκριμένο πηγαίο κώδικα πλατφόρμας, ο οποίος μπορεί στη συνέχεια να μεταγλωττιστεί και να κάνει build σε κάθε κινητή πλατφόρμα που υποστηρίζεται από το framework. Οι εφαρμογές που προκύπτουν μπορούν επομένως να εκμεταλλευτούν το πλήρες δυναμικό της πλατφόρμας όπως είναι - ιδανικά - δεν διακρίνονται από τις εγγενείς εφαρμογές. Στα εμπορικά frameworks αυτής της προσέγγισης περιλαμβάνονται το WebRatio Mobile, BiznessApps και Bubble (WebRatio Srl 2015; Bizness Εφαρμογές 2019; Bubble Group 2019), ενώ στον ακαδημαϊκό χώρο τα frameworks που είναι πιο διαδεδομένα είναι για παράδειγμα το MD2, MAML, και το applause (applause 2015). Ενώ υπάρχουν πολλά frameworks τόσο για την βιομηχανία όσο και για τον ακαδημαϊκό χώρο, σπάνια χρησιμοποιείτε η προσέγγιση ανάπτυξης που βασίζεται στο μοντέλο από επαγγελματίες, η οποίοι συνήθως ασχολούνται περισσότερο με τις υβριδικές και ερμηνευμένες προσεγγίσεις ανάπτυξης [7].



Εικόνα 16: Overview of the Model-Driven Apps approach build workflow. [4]

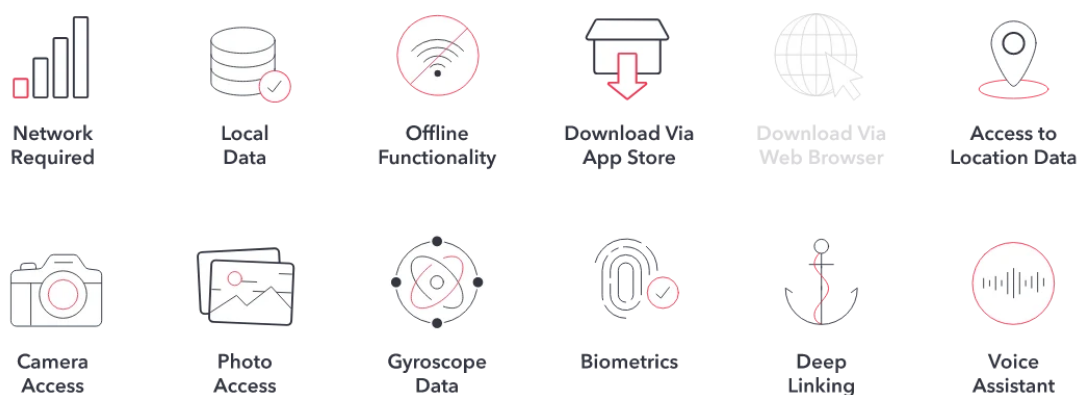
2.2.5 Cross-compiled apps

Οι εφαρμογές διασταυρούμενης σύνταξης δημιουργούνται χρησιμοποιώντας την ίδια βασική τεχνολογία ιστού με τις hybrid και interpreted εφαρμογές.

Η βασική διαφορά μεταξύ της προσέγγισης Cross-compiled και των προαναφερθεισών προσεγγίσεων είναι ότι λόγω μεταγλωττιστή, frameworks και εργαλείων ανάπτυξης η προσέγγιση Cross-compiled δεν βασίζεται σε στοιχεία WebView ή σε διερμηνείς στη συσκευή για την απόδοση της διασύνδεσης χρήστη ή την πρόσβαση σε δυνατότητες της πλατφόρμας και της συσκευής.

Αντίθετα, μια γλώσσα όπως η C# (Xamarin) μεταγλωττίζεται σε εκτελέσιμο κώδικα Native byte για στοχευμένες πλατφόρμες. Έτσι, το στρώμα γέφυρας γνωστό από τις ερμηνευμένες και υβριδικές προσεγγίσεις δεν υφίσταται στις εφαρμογές Cross compiled. Ούτε η χρήση ούτε η πρόσβαση στις δυνατότητες των εγγενών συσκευών ελέγχεται από ένα τέτοιο επίπεδο, αλλά μάλλον εκτίθεται στον προγραμματιστή της εφαρμογής μέσω του kit ανάπτυξης λογισμικού (SDK), το οποίο αντιστοιχεί ανάλογα με τη λειτουργικότητα στα SDK των υποκείμενων πλατφορμών.

Άλλη μία θετική συνέπεια της μεταγλώττισης του κώδικα σε Native byte είναι ότι η διεπαφές χρήστη που δημιουργούνται αποδίδονται ως στοιχεία εγγενών διεπαφών [4].

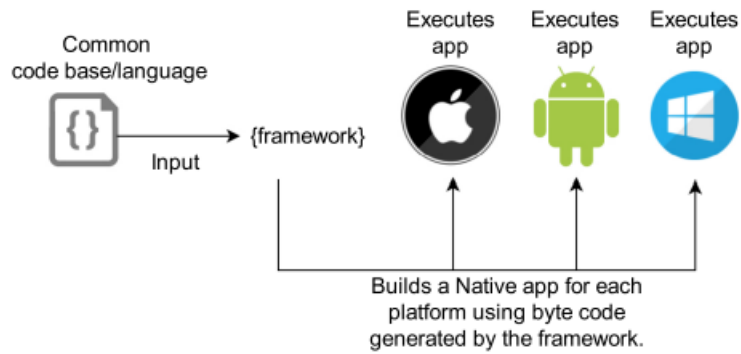


Εικόνα 17: Cross-Compiled applications functionality illustration[5]

Δεδομένου ότι αυτές οι εφαρμογές διασταύρωσης εκτελούνται επίσης εγγενώς στη συσκευή, απολαμβάνουν πλήρη πρόσβαση σε εγγενή API. Κατά συνέπεια, οι cross-compiled εφαρμογές είναι επιτυχημένες τόσο για εφαρμογές καταναλωτών όσο και για επιχειρησιακές εφαρμογές. Οι εφαρμογές διασταυρούμενης μεταγλώττισης παρέχουν πρόσβαση σε API συσκευών και συνεπώς παρέχουν μια πληρέστερη και πλουσιότερη εμπειρία εφαρμογών που είναι συχνά πιο αποτελεσματική από τα PWA ή τις υβριδικές εφαρμογές.

Το Flutter (ένας πρόσφατος συμμετέχων στο τοπίο ανάπτυξης εφαρμογών) συγκεντρώνει ολοένα και μεγαλύτερη προσοχή. Όπως και οι PWA εφαρμογές, έτσι και η προσέγγιση εφαρμογών του Flutter (Cross-Compiled approach) απολαμβάνουν την υποστήριξη και την επένδυση της Google. Αυτές οι εφαρμογές (με χρήση Flutter) έχουν δημιουργηθεί χρησιμοποιώντας μια νέα γλώσσα, την Dart, η οποία υποστηρίζεται επίσης από την Google. Οι εφαρμογές Flutter εκτελούνται εγγενώς σε ένα κέλυφος Flutter.

Η React Native, που ανήκει στο Facebook, υποβάλλεται επίσης σε έναν ακόμη κύκλο επανεπένδυσης και επανεργασίας. Οι πρώτες υποσχέσεις υποδηλώνουν ότι θα επιτύχει απόδοση σε σχεδόν εγγενή επίπεδα. Αυτές οι επενδύσεις στέλνουν ένα ηχηρό και σαφές μήνυμα: το μέλλον οδεύει προς την ανάπτυξη cross-platform εφαρμογών [5].



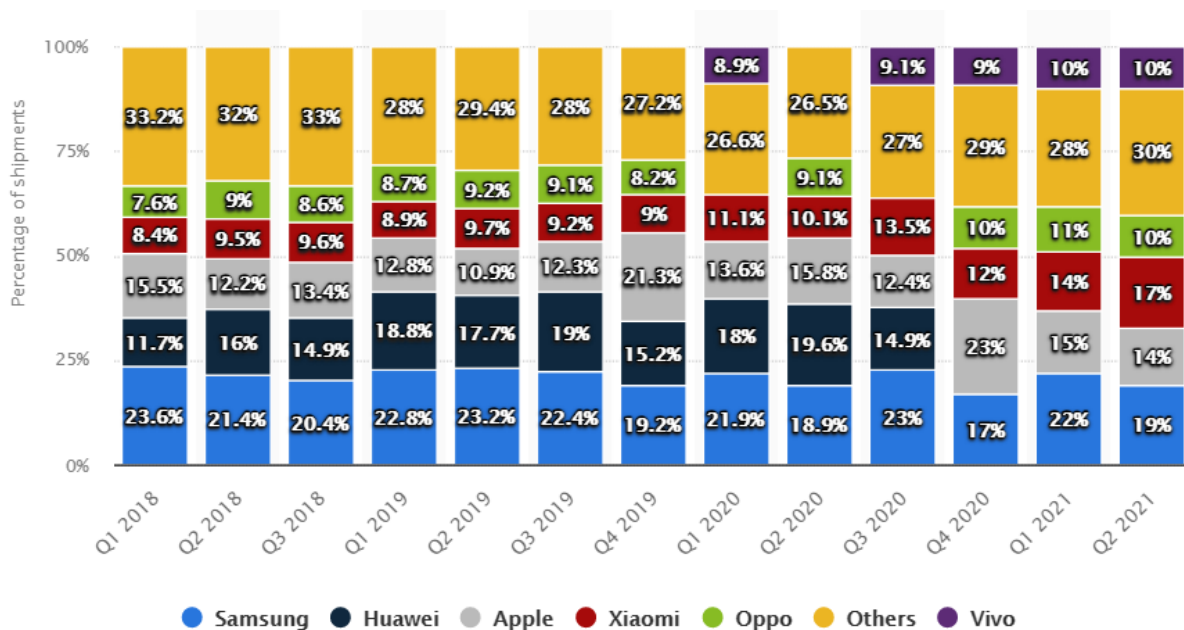
Εικόνα 18: Overview of the Cross-Compiled Apps approach build workflow. [4]

3 Επικρατέστερες τεχνολογίες

Σήμερα η αγορά, ειδικά όταν μιλάμε για το κομμάτι της τεχνολογίας αλλάζει με πολύ γρήγορους ρυθμούς με κάποιες εταιρίες να ηγούνται αυτής. Σε αυτό το κεφάλαιο θα γίνει μία αναφορά στην κατάσταση την αγοράς όπως έχει τώρα και μία παρουσίαση των επικρατέστερων εταιριών και τεχνολογιών.

Smartphone

Η Χίαομι ξεπέρασε την Apple το δεύτερο τρίμηνο του 2021, για να γίνει ο δεύτερος μεγαλύτερος πωλητής smartphone ως προς το μερίδιο αγοράς. Αυτή είναι η υψηλότερη θέση που κατείχε ποτέ η Χίαομι, καθώς συνεχίζουν να γεμίζουν το κενό που άφησαν οι πτωτικές πωλήσεις της Huawei. Όπως φαίνεται και στο γράφημα παρακάτω την πρώτη θέση σε πωλήσεις smartphone κατέχει η Samsung με ποσοστό 19% έναντι των ανταγωνιστών για το 2021.[14]



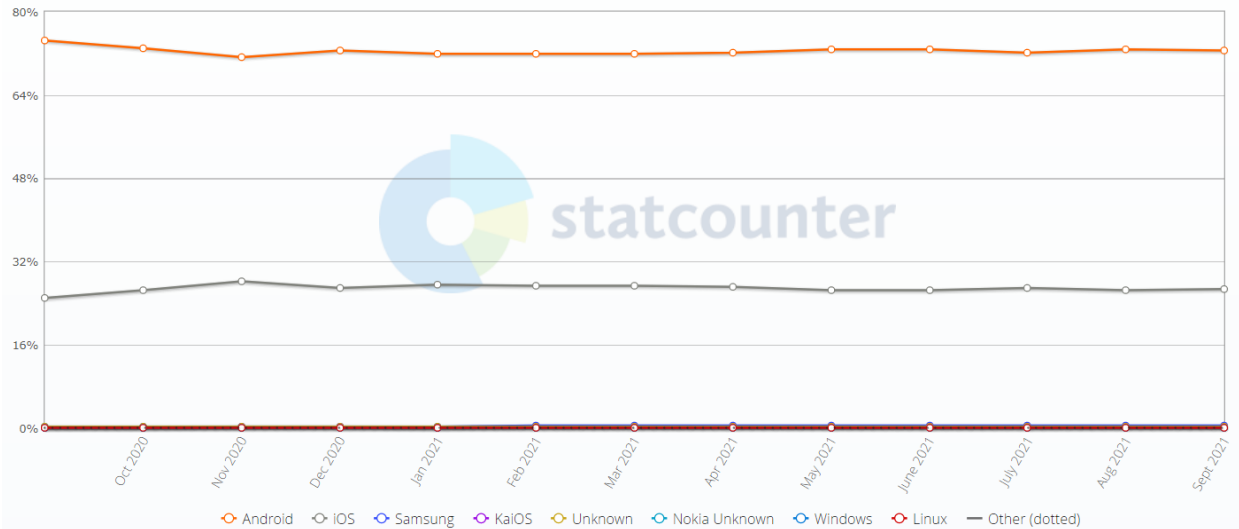
Εικόνα 19: Smartphone shipment share worldwide Q1'18-Q2'21, by vendor. [14]

Λειτουργικά συστήματα για κινητές συσκευές

Το Android διατήρησε τη θέση του ως το κορυφαίο λειτουργικό σύστημα κινητής τηλεφωνίας παγκοσμίως τον Ιούνιο του 2021, ελέγχοντας την αγορά λειτουργικών συστημάτων κινητής τηλεφωνίας με μερίδιο κοντά στο 73 τοις εκατό. Το Android της Google και το iOS της Apple κατέχουν από κοινού πάνω από το 99 τοις εκατό του παγκόσμιου μεριδίου αγοράς. Και τα δύο που παρουσιάστηκαν το 2007, το Android της Google και το iOS της Apple έχουν προχωρήσει το λογισμικό τους όλα αυτά τα χρόνια και παρέχουν τακτικές ενημερώσεις στα λειτουργικά τους συστήματα.

Η Google προηγουμένως ονόμασε τις εκδόσεις Android για γλυκά και γλυκά - το ένατο λειτουργικό τους σύστημα ονομάστηκε Pie και χρησιμοποιήθηκε από περισσότερο από το 31 % των χρηστών κινητών Android παγκοσμίως. Ωστόσο, σε παράβαση της παράδοσης, η τελευταία έκδοση της Google ονομάζεται απλά Android 10.

Τον Σεπτέμβριο του 2019, η Apple κυκλοφόρησε τη δωδέκατη έκδοση του λειτουργικού της συστήματος (iOS 12) και λειτουργούσε στο 88 % των κινητών συσκευών της Apple από τον Αύγουστο του 2019. Το φθινόπωρο του 2019, η Apple κυκλοφόρησε το iOS 13 μαζί με το iPadOS - ένα νέο λειτουργικό σύστημα για κινητά που αναπτύχθηκε ειδικά για iPad και από τον Ιούνιο του 2020 χρησιμοποιήθηκε στο 81 τοις εκατό όλων των συσκευών της Apple [14].

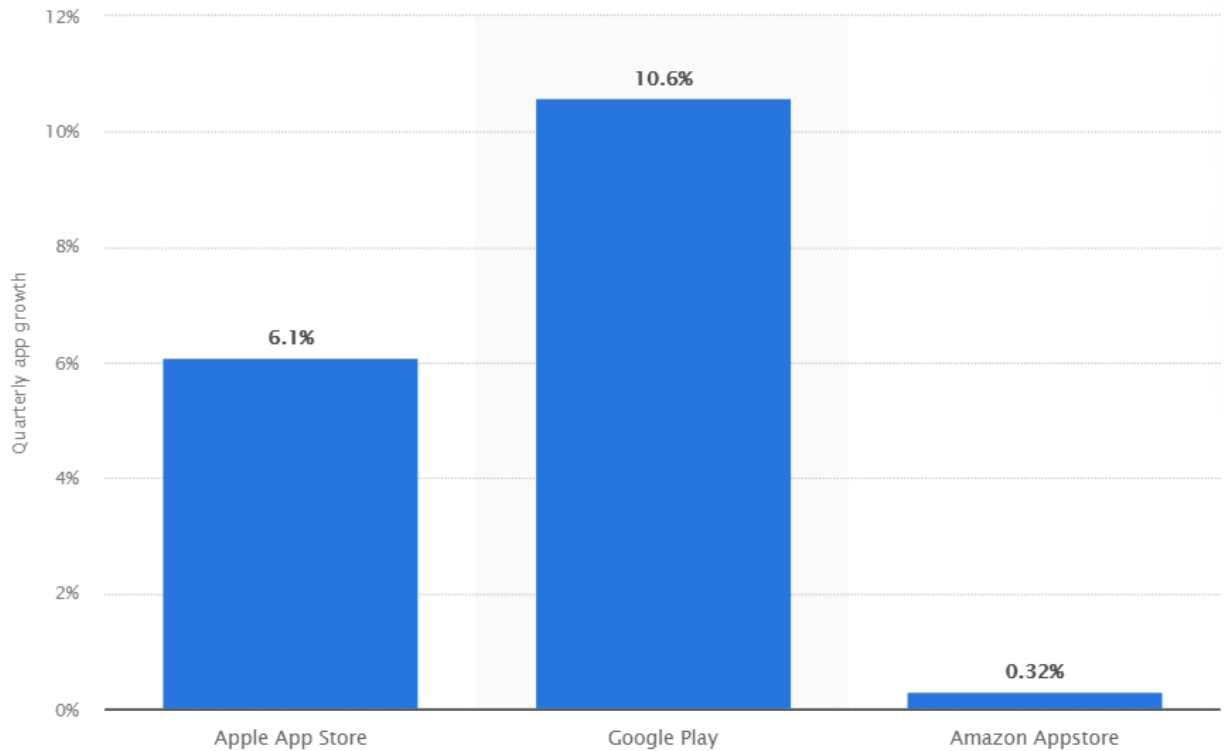


Εικόνα 20: Mobile Operating System Market Share Worldwide Sept 2020 - Sept 2021 [15]

Αγορές εφαρμογών για κινητές συσκευές

Από το πρώτο τρίμηνο του 2021, οι χρήστες Android είχαν τη δυνατότητα να επιλέξουν μεταξύ 3,48 εκατομμυρίων εφαρμογών, καθιστώντας το Google Play το κατάστημα εφαρμογών με τον μεγαλύτερο αριθμό διαθέσιμων εφαρμογών. Το Apple App Store ήταν το δεύτερο μεγαλύτερο κατάστημα εφαρμογών με περίπου 2,22 εκατομμύρια διαθέσιμες εφαρμογές για iOS. Ενώ ο ακριβής αριθμός εφαρμογών μπορεί να κυμαίνεται καθώς η Apple και η Google αφαιρούν τακτικά περιεχόμενο χαμηλής ποιότητας από τα καταστήματα εφαρμογών τους, ο αριθμός των εφαρμογών αυξάνεται σταθερά με τα χρόνια.

Αυτό το γράφημα παρουσιάζει την αύξηση των διαθέσιμων εφαρμογών για κινητά στο Amazon Appstore, το Apple App Store και το Google Play σε όλο τον κόσμο κατά το πρώτο τρίμηνο του 2021. Κατά την πιο πρόσφατη περίοδο μέτρησης, η ανάπτυξη εφαρμογών για κινητά στο Google Play αυξήθηκε σχεδόν κατά 11 τοις εκατό σε σύγκριση με την προηγούμενο τρίμηνο [14].



Εικόνα 21 : Number of apps available in leading app stores as of 1st quarter 2021 [14]

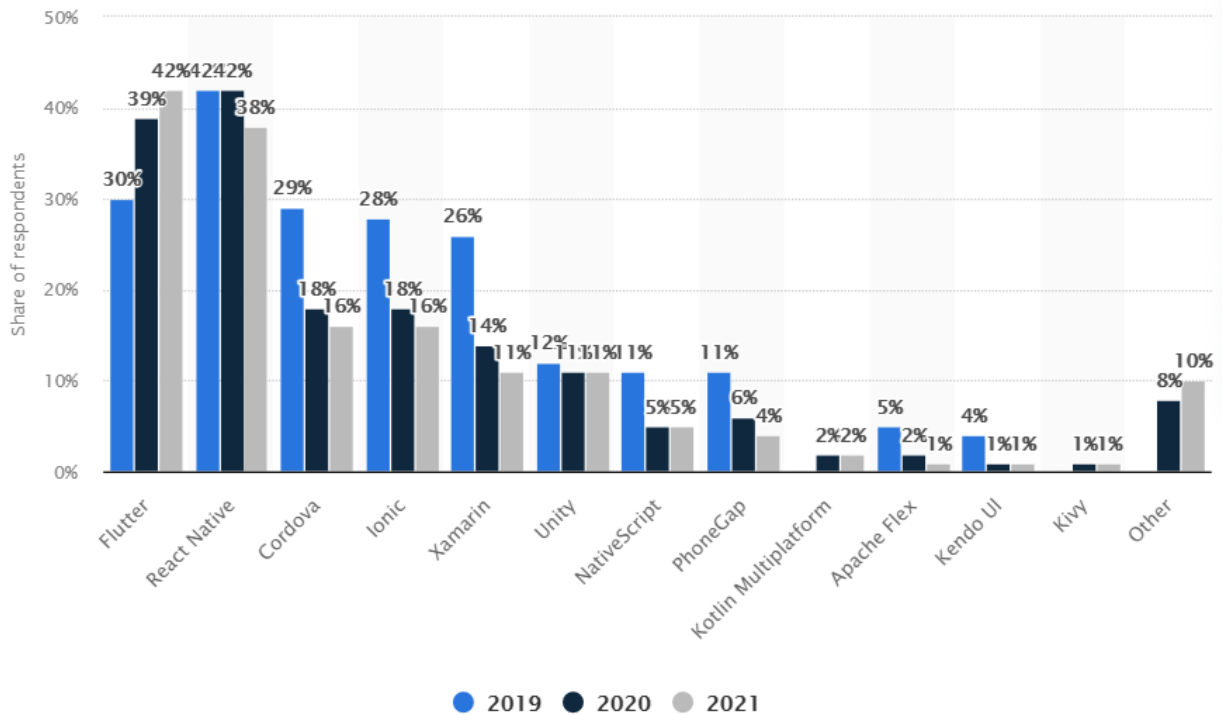
Τεχνολογίες ανάπτυξης εφαρμογών για κινητές συσκευές

Όπως προαναφέρθηκε από τις περισσότερες από 5,6 εκατομμύρια εφαρμογές που διατίθεται σε όλα τα καταστήματα εφαρμογών, η απόλυτη πλειοψηφία αναπτύχθηκαν χρησιμοποιώντας native τεχνολογίες[10].

Τα τελευταία χρόνια η χρήση cross-platform τεχνολογιών έχει αυξηθεί σε μεγάλο ποσοστό με το Flutter να είναι το πιο δημοφιλές framework πλατφορμών για κινητά που χρησιμοποιείται από προγραμματιστές παγκόσμια, σύμφωνα με έρευνα προγραμματιστών του 2021.

Με βάση την έρευνα, το 42 % των προγραμματιστών λογισμικού χρησιμοποίησε το Flutter. Συνολικά, περίπου το ένα τρίτο των προγραμματιστών κινητών χρησιμοποιεί τεχνολογίες cross-platform[14].

Παρόλο που είναι το νεότερο cross-platform framework, το Flutter δημιουργεί τη δική του ξεχωριστή εικόνα στη σημερινή αγορά. Είναι ένα framework ανοιχτού κώδικα για πολλές πλατφόρμες που κυκλοφόρησε για πρώτη φορά από την Google τον Μάιο του 2017. Ένα ξεχωριστό χαρακτηριστικό που κάνει το Flutter διαφορετικό είναι ότι χρησιμοποιεί γλώσσα προγραμματισμού Dart αντί για JavaScript.



Εικόνα 22: Cross-platform mobile frameworks used by software developers worldwide from 2019 to 2021 [14]

Στην δεύτερη θέση ανέρχεται η React Native με ποσοστό 38% έχοντας χάσει έδαφος σε σχέση με τα προηγούμενα έτη όπου κατείχε την πρώτη θέση με ποσοστό 42%.

Η React Native χρησιμοποιείται ευρέως για iOS και Android, καθώς είναι μια γνωστή οντότητα στην ανάπτυξη εφαρμογών μεταξύ πλατφορμών. Δημιουργήθηκε από το Facebook το 2015 και έκτοτε, η κοινότητα των προγραμματιστών του αυξάνεται με αλματώδεις ρυθμούς. Το framework είναι βασισμένο σε JavaScript και δίνει μια φυσική αίσθηση για πλατφόρμες Android και iOS.

Καθώς είναι χτισμένο σε JavaScript, γίνεται πιο εύκολο για τους προγραμματιστές να δημιουργήσουν διαισθητικές διεπαφές χρήστη χρησιμοποιώντας το React Native. Παρά το γεγονός ότι βασίζεται σε JavaScript. Το React Native δεν χρησιμοποιεί τεχνολογία HTML ή WebView ή CSS. Αντί αυτών, τα συστατικά της πλατφόρμας περιλαμβάνουν JSX και polyfills παρόμοια με CSS [16].

Σύμφωνα με στοιχεία που παρέχει το GitHub Το Flutter έχει σταθερή ροή 30K προγραμματιστών κάθε χρόνο και η React Native έχει 12-15 χιλιάδες νέους χρήστες κάθε χρόνο. Με βάση την τάση των τελευταίων τριών ετών, το Flutter έχει 3 φορές περισσότερους χρήστες από το React Native - 75k έναντι 25k [17].

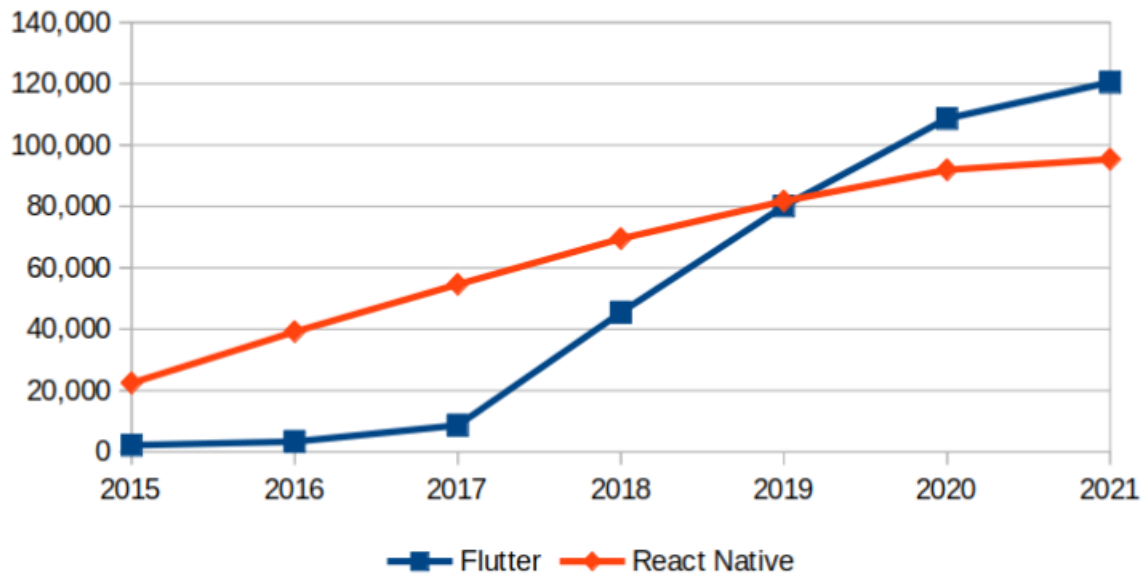
Year	Flutter	React Native
2015	2,200	22,468
2016	3,367	39,175
2017	8,665	54,625
2018	45,487	69,523
2019	80,135	81,732
2020	108,655	91,924
2021	120,510	95,391

Εικόνα 23: React Native and Flutter users based on GitHub stars [17]

Οι λεπτομέρειες που παρέχονται παρακάτω βασίζονται στα δεδομένα που συλλέγονται χρησιμοποιώντας το GitHub GraphQL API:

Από τα 120k αστέρια (GitHub stars) για το Flutter

- Τα 20k αστέρια έχουν δοθεί από χρήστες React Native.
- Το 24k αστέρια έχουν δοθεί από χρήστες Vue.
- Το 23k αστέρια έχουν δοθεί από χρήστες React.
- Συνολικά τα 41k αστέρια έχουν δοθεί από χρήστες της JavaScript



Εικόνα 24: React Native and Flutter users based on GitHub stars (chart) [17]

4 Η εφαρμογή μου (G.D.T.W. / Go Discover The World)

4.1 Εισαγωγή

Η εφαρμογή Go Discover The World, GDTW για συντομία είναι μία native εφαρμογή Android βασισμένη πάνω στην τεχνολογία Google Maps της Google. Κύριος σκοπός της εφαρμογής είναι προσφέρει στον χρήστη πληροφορίες για το μέρος που βρίσκετε την προκειμένη στιγμή βασισμένες σε συστάσεις άλλων χρηστών με ευχάριστο και διαδραστικό τρόπο, αλλά και να παρέχει ο ίδιος τις δικές του συστάσεις σε άλλους χρήστες.

Ανοίγοντας την εφαρμογή ο εκάστοτε χρήστης θα μπορεί:

- Να δημιουργεί **NPC** πάνω στον χάρτη. Ως NPC εδώ ορίζονται εικονικά <ανθρωπάκια> που βρίσκονται καρφισωμένα πάνω στον χάρτη Google Map.
- Να περνάει σαν πληροφορία **διαλόγους** (απαντήσεις) και **μέρη** (σημεία ενδιαφέροντος) στο κάθε NPC κατηγοριοποιημένα με βάση το τί πρεσβεύει το κάθε <μέρος>.
- Να δημιουργεί **μέρη** < σημεία ενδιαφέροντος > καρφισωμένα πάνω στον χάρτη Google Map κατηγοριοποιημένα με βάση το τί πρεσβεύει το κάθε <μέρος>, τα οποία θα μπορεί εκ των υστέρων να περάσει σαν πληροφορία στα NPC που έχει δημιουργήσει.
- Να μπαίνει σε **εικονικό διάλογο** με NPC ώστε να πάρει τις πληροφορίες αυτές μέσα από ένα μενού διαλόγου (σύμφωνα με τις κατηγορίες), πλησιάζοντας το σημείο του χάρτη όπου βρίσκονται τα NPC αυτά.
- Να αποθηκεύει σημεία ενδιαφέροντος για μελλοντική προβολή.
- Να ενημερώνει την εφαρμογή αν οι πληροφορίες που πείρε από κάποιο NPC ήταν χρήσιμες ή όχι.

Το GDTW χωρίζει τα NPC σε τρεις κατηγορίες (HIGH, MEDIUM, LOW) με βάση το feedback των χρηστών, ώστε να μπορεί ο χρήστης να φιλτράρει ποια NPC θα εμφανίζονται πάνω στον χάρτη. Τα NPC με πολύ αρνητικό feedback διαγράφονται αυτόματα.

4.2 Λειτουργικότητα

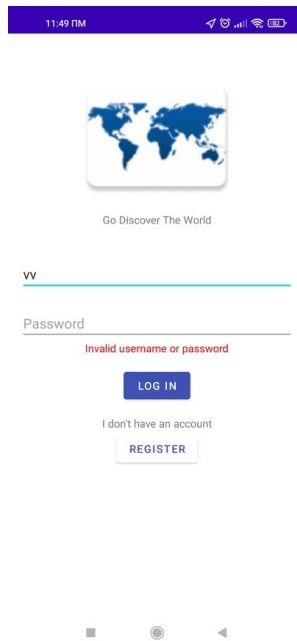
Σε αυτήν την υποενότητα παρουσιάζετε το Activity flow της εφαρμογής όπως θα το βλέπει ο χρήστης πατώντας το εικονίδιο εκκίνησης.



Εικόνα 25: App launch icon

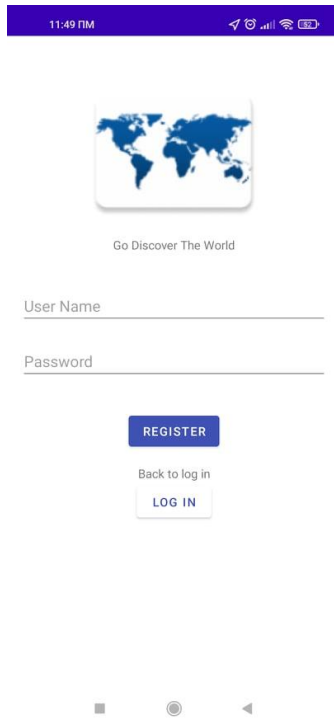
4.2.1 Login/Register

Μόλις ανοίξει η εφαρμογή ζητείτε από τον χρήστη να κάνει **login** το username και password του, Προβάλλοντας και τα κατάλληλα μηνύματα σε περίπτωση που ο χρήστης εισάγει λάθος στοιχεία.



Εικόνα 26: GDTW Login

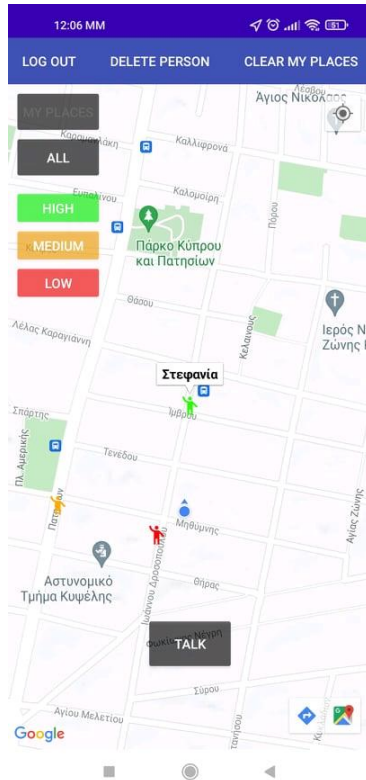
Σε περίπτωση που ο χρήστης δεν έχει λογαριασμό πατώντας το κουμπί <Register> μπορεί να δημιουργήσει έναν όπου θα του αποδοθεί και ένα μοναδικό **user id** (κρυφό από τον χρήστη).



Εικόνα 27: GDTW Register

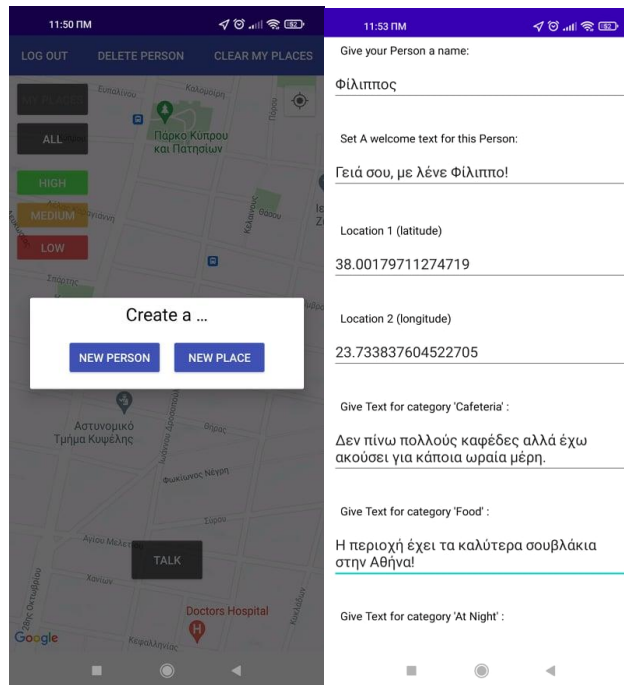
4.2.2 Χάρτης

Η οθόνη του χάρτη είναι το κύριο περιβάλλον όπου ο χρήστης μπορεί να αναζητάει τα NPC.



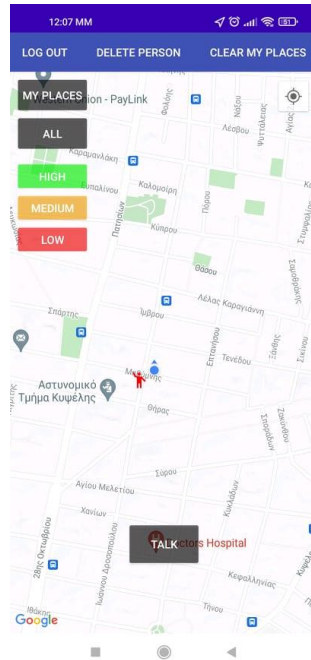
Εικόνα 28: GDTW Map

Κάνοντας LongClick πάνω στον χάρτη ο χρήστης μπορεί να δημιουργήσει ένα καινούριο μέρος η NPC στο σημείο που πάτησε.



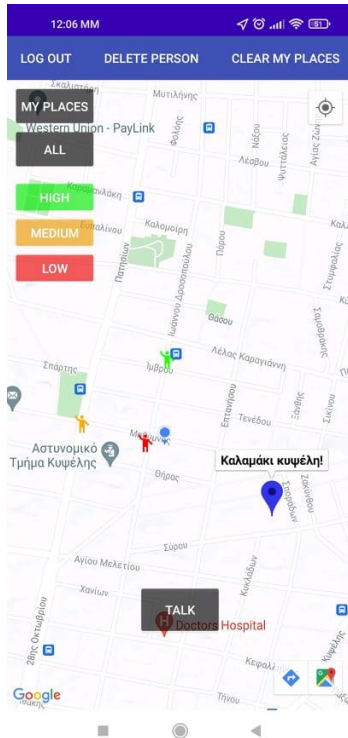
Εικόνα 29: GDTW New person/place

Πατώντας ένα από τα κουμπιά (ALL, HIGH, MEDIUM, LOW) ο χρήστης μπορεί να επιλέξει πια NPC εμφανίζονται στον χάρτη.



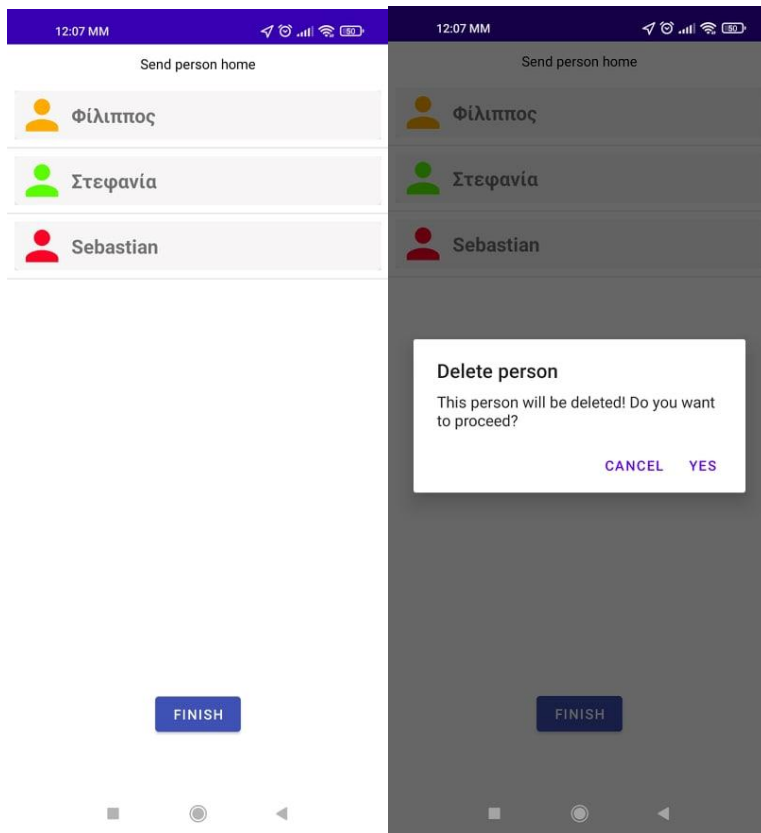
Εικόνα 30: GDTW Low button pressed

Πατώντας το κουμπί MY PLACES εμφανίζονται τα αποθηκευμένα μέρη του χρήστη που έχει πάρει σαν πληροφορία από κάποιον διάλογο με NPC.



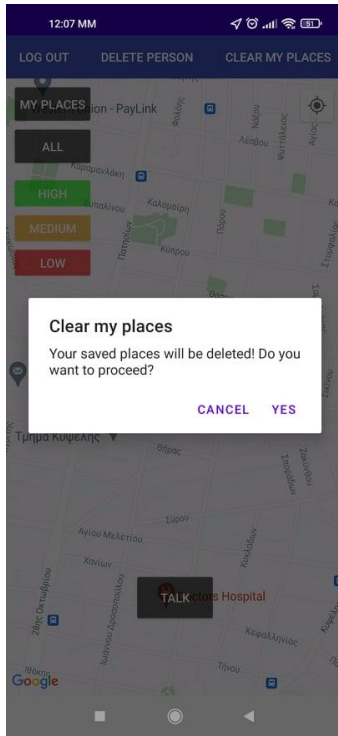
Εικόνα 31: GDTW My places button pressed

Πατώντας το DELETE PERSON ο χρήστης θα βρεθεί στην οθόνη διαγραφής των NPC που έχει φτιάξει.



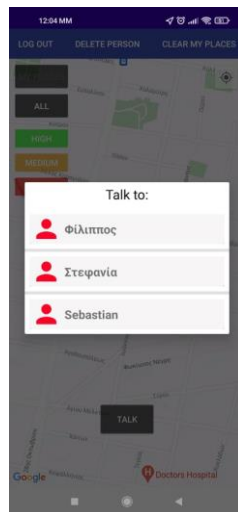
Εικόνα 32: GDTW Delete NPC

Αν Πατηθεί το κουμπί CLEAR MY PLACES ο χρήστης μπορεί να διαγράψει τα αποθηκευμένα μέρη.



Εικόνα 33: GDTW Clear my places

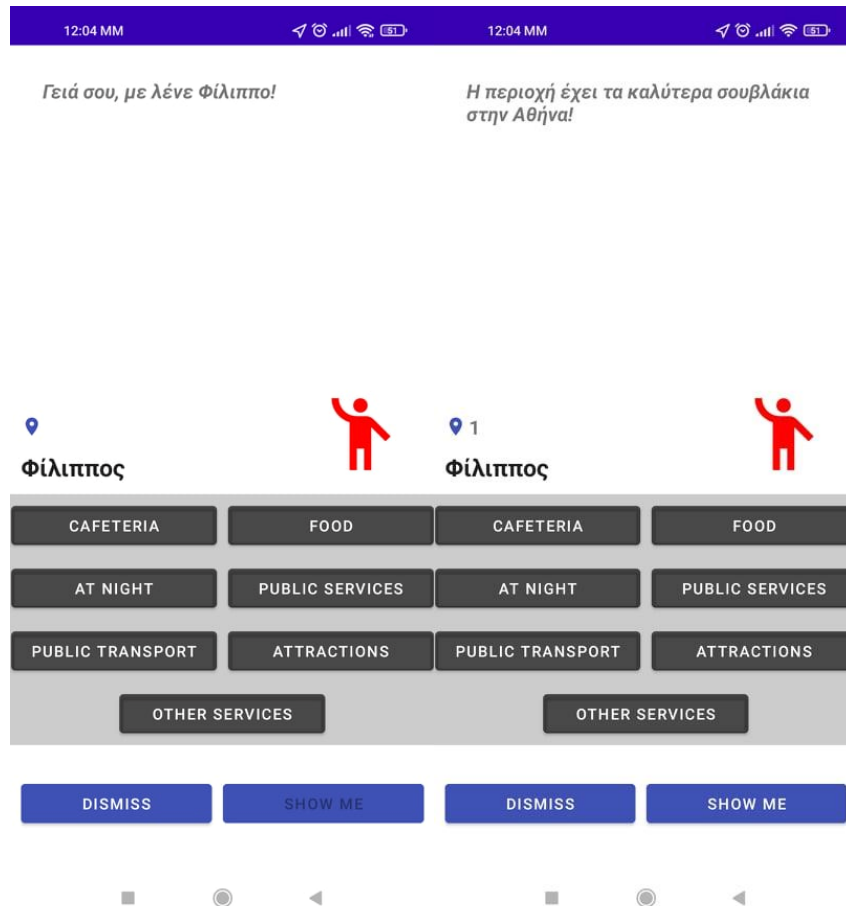
Τέλος όταν ο χρήστης βρεθεί κοντά σε NPC πατώντας το TALK εμφανίζετε μία λίστα των NPC που είναι σε talk range (εδώ το talk range είναι μεγάλο για λόγους επιδείξεις) και από εκεί μπορεί να επιλέξει με ποιο NPC θέλει να μπει σε διάλογο.



Εικόνα 34: GDTW Talk to

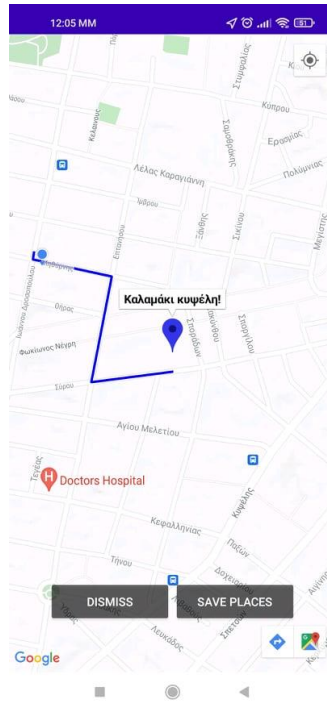
4.2.3 NPC dialog

Εδώ ο χρήστης βρίσκεται σε διάλογο με κάποιον NPC όπου θα λάβει απαντήσεις ανάλογα την κατηγορία που θα επιλεχτεί.



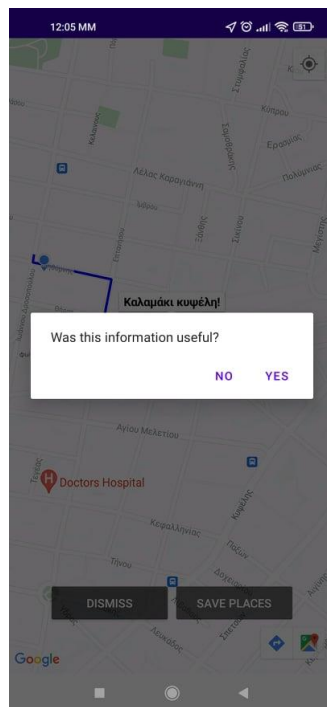
Εικόνα 35: NPC dialog

Για να δει κανείς τα προτεινόμενα μέρη αρκεί να πατήσει το κουμπί SHOW ME



Εικόνα 36: GDTW NPC places map

Πατώντας το SAVE PLACES τα μέρη αποθηκεύονται για μελλοντική προβολή και με το κουμπί DISMISS προκύπτει ο παρακάτω διάλογος για το feedback ου χρήστη πριν βρεθεί ξανά στην οθόνη του χάρτη.



Εικόνα 37: GDTW NPC places

4.3 Λίγα λόγια για το Android

Το Android είναι ένα λειτουργικό σύστημα για κινητά που υπάρχει εδώ και σχεδόν 15 χρόνια. Θα το βρείτε κυρίως ως το βασικό λειτουργικό σύστημα τηλεφώνων και tablet σε όλο τον κόσμο. Το Android ανήκει στον γίγαντα αναζήτησης Google. Ωστόσο, το σύστημα είναι ανοιχτού κώδικα (open source), γεγονός που το καθιστά ελεύθερα προσβάσιμο σε οποιονδήποτε, ακόμη και για εμπορική χρήση. Αυτό κάνει το Android πολύ διαφορετικό από τα iOS, macOS και Windows της Microsoft της Apple, που είναι όλες πλατφόρμες κλειστού κώδικα. Το Android είναι μακράν το πιο δημοφιλές λειτουργικό σύστημα στον κόσμο. Σύμφωνα με εκτιμήσεις, λειτουργεί σε 2,5 δισεκατομμύρια ενεργές συσκευές σε όλο τον κόσμο, με πάνω από τρία δισεκατομμύρια χρήστες — ή περίπου το 39% του συνολικού παγκόσμιου πληθυσμού. Αυτό ξεπερνά το iOS της Apple με ένα σημαντικό περιθώριο και μάλιστα βρίσκεται στην κορυφή των Windows της Microsoft, που είναι το δεύτερο πιο δημοφιλές λειτουργικό σύστημα παγκοσμίως.

Όταν οι περισσότεροι άνθρωποι σκέφτονται το Android, σκέφτονται τα τηλέφωνα. Αν και είναι αλήθεια ότι οι περισσότερες συσκευές Android είναι smartphone, υπάρχουν πολλές άλλες συσκευές εκεί έξω με Android. Τα tablet είναι η πιο προφανής δευτερεύουσα συσκευή Android. Εξάλλου, είναι απλά μεγάλα τηλέφωνα, από πολλές απόψεις. Το Android εμφανίζεται και σε smartwatches. Εάν διαθέτετε ένα ρολόι που λειτουργεί με Wear OS, αυτό είναι ένα λειτουργικό σύστημα που βασίζεται σε Android.

Τι είναι ένα λειτουργικό σύστημα που βασίζεται σε Android;

Ένα λειτουργικό σύστημα που βασίζεται σε Android είναι όταν κάποιος παίρνει το Android και το τροποποιεί για να το κάνει κάτι διαφορετικό, αλλά εξακολουθεί να βασίζεται στον ίδιο βασικό κώδικα.

Υπάρχει επίσης μια πλατφόρμα τηλεόρασης, που ονομάζεται Android TV. Δεν μπορούμε να ξεχάσουμε το Android Automotive, το οποίο είναι λογισμικό που βασίζεται σε Android και τροφοδοτεί τα οχήματα. Τέλος, υπάρχουν και άλλα λειτουργικά συστήματα εκεί έξω που δεν βασίζονται στο Android αλλά υποστηρίζουν την εκτέλεση εφαρμογών Android. Οι πρόσφατες εκδόσεις του Chrome OS το επιτρέπουν. Αυτό σημαίνει ότι σχεδόν όλα τα Chromebook στην αγορά υποστηρίζουν εφαρμογές Android. Από τα τέλη του 2021, τα Windows 11 θα υποστηρίζουν και αυτά εφαρμογές Android.

Είτε το πιστεύετε είτε όχι, το Android ξεκίνησε ως λογισμικό για κάμερες. Ο Andy Rubin και η ομάδα του ανέπτυξαν το Android ξεκινώντας το 2003 χρησιμοποιώντας τον βασικό κώδικα από το Linux, ένα άλλο λειτουργικό σύστημα ανοιχτού κώδικα. Η ιδέα ήταν να φτιάξουμε ένα γενικό λειτουργικό σύστημα που θα μπορούσαν να χρησιμοποιήσουν όλες οι εταιρείες κάμερας. Ωστόσο, κατά τη μεταγενέστερη ανάπτυξη του Android, ο Rubin συνειδητοποίησε ότι τα smartphones ήταν το μέλλον. Αντ' αυτού αποφάσισε να ανανεώσει το Android ως λειτουργικό σύστημα smartphone. Όταν κάτι είναι ανοιχτού κώδικα σημαίνει ότι ο κάτοχος των πνευματικών δικαιωμάτων επιτρέπει τη χρήση του για οποιονδήποτε σκοπό, χωρίς καμία ανάγκη οικονομικής αμοιβής. Όπως αναφέρθηκε προηγουμένως, ο βασικός κώδικας του Android βασίζεται σε λογισμικό ανοιχτού κώδικα που ονομάζεται Linux. Αυτό σημαίνει ότι το Android, εξ ορισμού, πρέπει επίσης να είναι ανοιχτού κώδικα.

Μέχρι σήμερα, υπήρξαν 18 εκδόσεις Android, με 11 σημαντικές εκδόσεις. Η πιο πρόσφατη σταθερή έκδοση του λειτουργικού συστήματος είναι το Android 11. Αναμένουμε ότι το Android 12 θα κυκλοφορήσει αργότερα το 2021, στα τέλη Αυγούστου ή στις αρχές Σεπτεμβρίου, γεγονός που θα το κάνει την 19η έκδοση και τη 12η σημαντική έκδοση. Αρχικά, η Google ονόμασε το Android από τις «γλυκίες λιχουδιές». Κάθε όνομα <λιχουδιάς> έβγαινε με αλφαβητική σειρά, ξεκινώντας με το γράμμα "C" το 2009. Ωστόσο, η Google εγκατέλειψε αυτήν την τάση το 2019 με την κυκλοφορία του Android 10.

Εδώ είναι όλες οι κύριες εκδόσεις Android:

- 2009 — Cupcake (v. 1.5)
- 2009 — Donut (v. 1.6)
- 2009 — Eclair (vs. 2.0, 2.0.1, and 2.1)
- 2010 — Froyo (vs. 2.2 through 2.2.3)
- 2010 — Gingerbread (vs. 2.3 through 2.3.7)
- 2011 — Honeycomb (vs. 3.0 through 3.2.6)
- 2011 — Ice Cream Sandwich (vs. 4.0 through 4.0.4)
- 2012 — Jelly Bean (vs. 4.1 through 4.3.1)
- 2013 — KitKat (vs. 4.4 through 4.4W.2)
- 2014 — Lollipop (vs. 5.0 through 5.1.1)
- 2015 — Marshmallow (vs. 6.0 through 6.0.1)
- 2016 — Nougat (vs. 7.0 through 7.1.2)
- 2017 — Oreo (vs. 8.0 and 8.1)
- 2018 — Pie (v. 9.0)
- 2019 — Android 10
- 2020 — Android 11

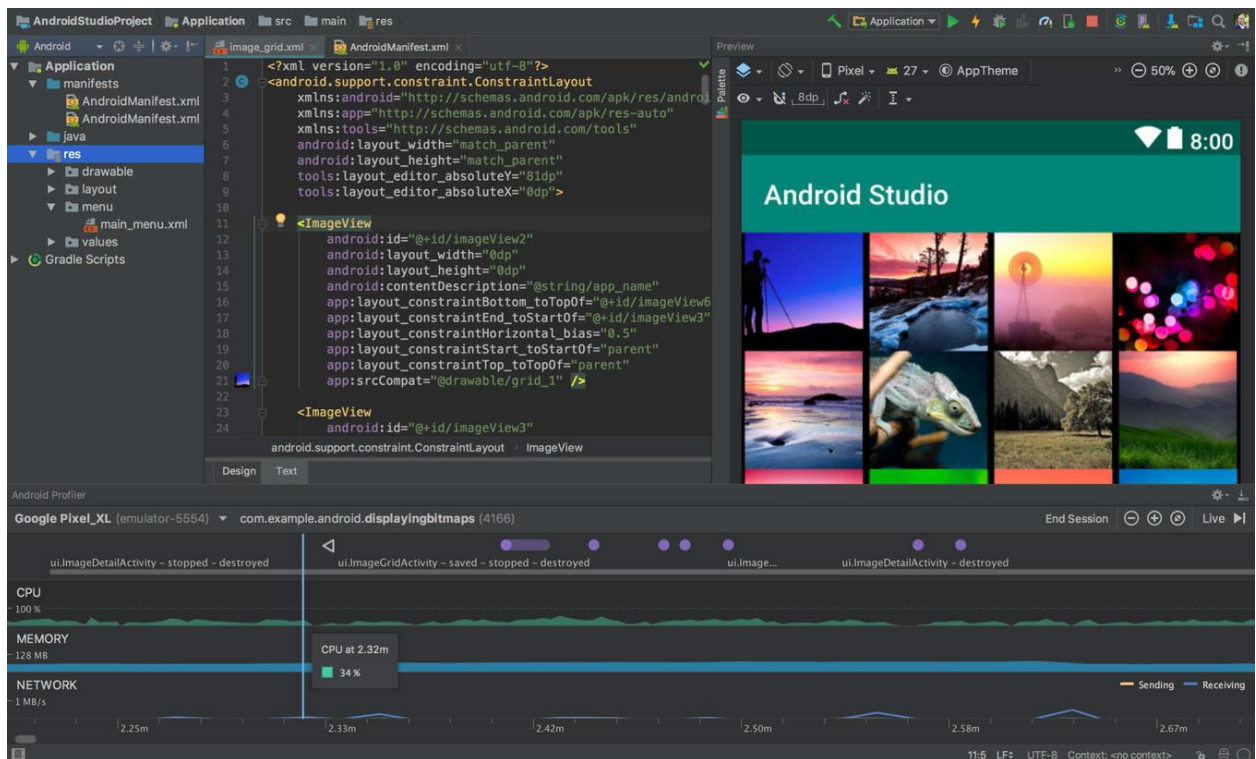
Το Android 11 εισήγαγε πολλές νέες δυνατότητες που σχετίζονται με την ανταλλαγή μηνυμάτων, το απόρρητο, την ασφάλεια και την προσβασιμότητα. Με το Android 12, αναμένουμε από την Google να ανανεώσει την εμφάνιση και την αίσθηση του λειτουργικού συστήματος. Θα υπάρχουν επίσης ακόμη περισσότερες δυνατότητες και έλεγχοι που σχετίζονται με το απόρρητο και την ασφάλεια.[18]

4.4 Περιβάλλον ανάπτυξης - IDE (Android Studio)

Το Android Studio είναι το επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για την ανάπτυξη εφαρμογών Android, βασισμένο στο IntelliJ IDEA . Εκτός από τα ισχυρά εργαλεία επεξεργασίας κώδικα και εργαλεία προγραμματιστών της IntelliJ, το Android Studio προσφέρει ακόμη περισσότερες δυνατότητες που ενισχύουν την παραγωγικότητά κατά τη δημιουργία εφαρμογών Android, όπως:

- Ένα ευέλικτο σύστημα κατασκευής που βασίζεται σε Gradle.
- Ένας γρήγορος και πλούσιος σε χαρακτηριστικά εξομοιωτής.

- Ένα ενοποιημένο περιβάλλον όπου μπορεί να γίνει ανάπτυξη εφαρμογών για όλες τις συσκευές Android.
- Προώθηση αλλαγών κώδικα και πόρων στην εφαρμογή που εκτελείται χωρίς να γίνετε επανεκκίνηση της εφαρμογής.
- Πρότυπα κώδικα και ενσωμάτωση GitHub για να σας βοηθήσουν να δημιουργήσετε κοινές λειτουργίες εφαρμογής και να εισαγάγετε δείγμα κώδικα.
- Εκτεταμένα εργαλεία και πλαίσια δοκιμών.
- Εργαλεία Lint για να εντοπίσετε την απόδοση, τη χρηστικότητα, τη συμβατότητα έκδοσης και άλλα προβλήματα.
- Υποστήριξη C++ και NDK.
- Ενσωματωμένη υποστήριξη για την πλατφόρμα Google Cloud, που καθιστά εύκολη την ενσωμάτωση του Google Cloud Messaging και του App Engine



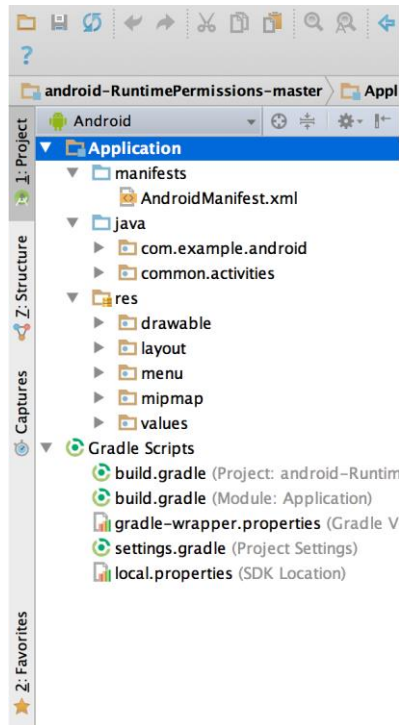
Εικόνα 38: Android Studio [19]

4.4.1 Project structure

Κάθε project στο Android Studio περιέχει μία ή περισσότερες λειτουργικές μονάδες με αρχεία πηγαίου κώδικα και αρχεία πόρων. Οι τύποι modules περιλαμβάνουν:

- Android app modules
- Library modules
- Google App Engine modules

Από προεπιλογή, το Android Studio εμφανίζει τα αρχεία του έργου σας στο Android project view, όπως φαίνεται στην εικόνα 30.



Εικόνα 39: Android project view[19]

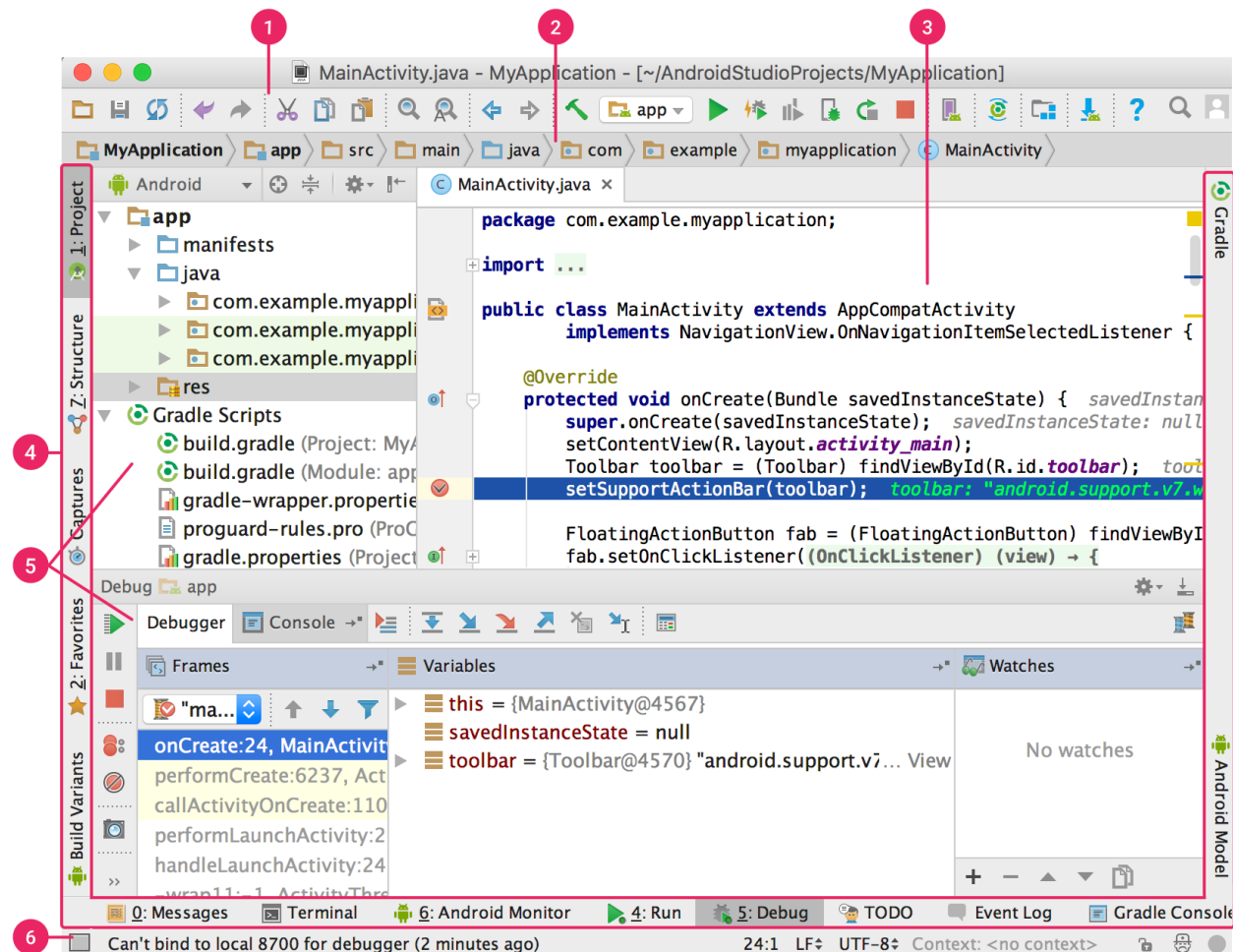
Αυτή η προβολή οργανώνεται ανά λειτουργικές μονάδες για να παρέχει γρήγορη πρόσβαση στα βασικά αρχεία πηγής του έργου σας. Όλα τα αρχεία έκδοσης είναι ορατά στο ανώτατο επίπεδο κάτω από το Gradle Scripts και κάθε λειτουργική μονάδα εφαρμογής (app module) περιέχει τους ακόλουθους φακέλους:

- manifests: Περιέχει το αρχείο AndroidManifest.xml.
- java: Περιέχει τα αρχεία πηγαίου κώδικα Java, συμπεριλαμβανομένου του κώδικα δοκιμής JUnit.
- res: Περιέχει όλους τους μη κωδικούς πόρους, όπως διατάξεις XML, συμβολοσειρές διεπαφής χρήστη και εικόνες bitmap.

Η δομή του έργου Android στο δίσκο διαφέρει από αυτήν την ισοπεδωμένη αναπαράσταση. Για να δείτε την πραγματική δομή του αρχείου του project, επιλέξτε Project από το αναπτυσσόμενο μενού Project (στην εικόνα 18, εμφανίζεται ως Android).[19]

4.4.2 Διεπαφή χρήστη

Το κύριο παράθυρο του Android Studio αποτελείται από πολλές λογικές περιοχές που προσδιορίζονται στην εικόνα 31.



Εικόνα 40: Κύριο παράθυρο του Android Studio[19]

1. Η **γραμμή εργαλείων** σας επιτρέπει να πραγματοποιείτε ένα ευρύ φάσμα ενεργειών, συμπεριλαμβανομένης της εκτέλεσης της εφαρμογής σας και της εκκίνησης εργαλείων Android.
2. Η **γραμμή πλοήγησης** σας βοηθά να πλοηγηθείτε στο έργο σας και να ανοίξετε αρχεία για επεξεργασία. Παρέχει μια πιο συμπαγή άποψη της δομής που είναι ορατή στο παράθυρο Project.
3. Το **παράθυρο του προγράμματος επεξεργασίας** είναι το σημείο όπου δημιουργείτε και τροποποιείτε τον κώδικα. Ανάλογα με τον τρέχοντα τύπο αρχείου, ο επεξεργαστής μπορεί να

αλλάξει. Για παράδειγμα, κατά την προβολή ενός αρχείου διάταξης, ο επεξεργαστής εμφανίζει τον Επεξεργαστή διάταξης.

4. Η **γραμμή παραθύρου εργαλείων** τρέχει γύρω από το εξωτερικό του παραθύρου IDE και περιέχει τα κουμπιά που σας επιτρέπουν να επεκτείνετε ή να συμπτύξετε μεμονωμένα παράθυρα εργαλείων.
5. Τα **παράθυρα εργαλείων** σας δίνουν πρόσβαση σε συγκεκριμένες εργασίες όπως διαχείριση έργου, αναζήτηση, έλεγχος έκδοσης και άλλα. Μπορείτε να τα επεκτείνετε και να τα καταρρίψετε.
6. Η **γραμμή κατάστασης** εμφανίζει την κατάσταση του έργου σας και του ίδιου του IDE, καθώς και τυχόν προειδοποιήσεις ή μηνύματα.[19]

4.4.3 Gradle build system

Το Android Studio χρησιμοποιεί το Gradle ως τη βάση του συστήματος κατασκευής, με περισσότερες δυνατότητες ειδικά για το Android που παρέχονται από την προσθήκη Android για το Gradle. Αυτό το σύστημα κατασκευής εκτελείται ως ενσωματωμένο εργαλείο από το μενού Android Studio και ανεξάρτητα από τη γραμμή εντολών. Μπορείτε να χρησιμοποιήσετε τις δυνατότητες του συστήματος κατασκευής για να κάνετε τα εξής:

- Προσαρμόστε, διαμορφώστε και επεκτείνετε τη διαδικασία κατασκευής.
- Δημιουργήστε πολλά APK για την εφαρμογή σας, με διαφορετικές δυνατότητες, χρησιμοποιώντας το ίδιο έργο και λειτουργικές μονάδες.
- Επαναχρησιμοποίηση κώδικα και πόρων σε σύνολο πηγών.

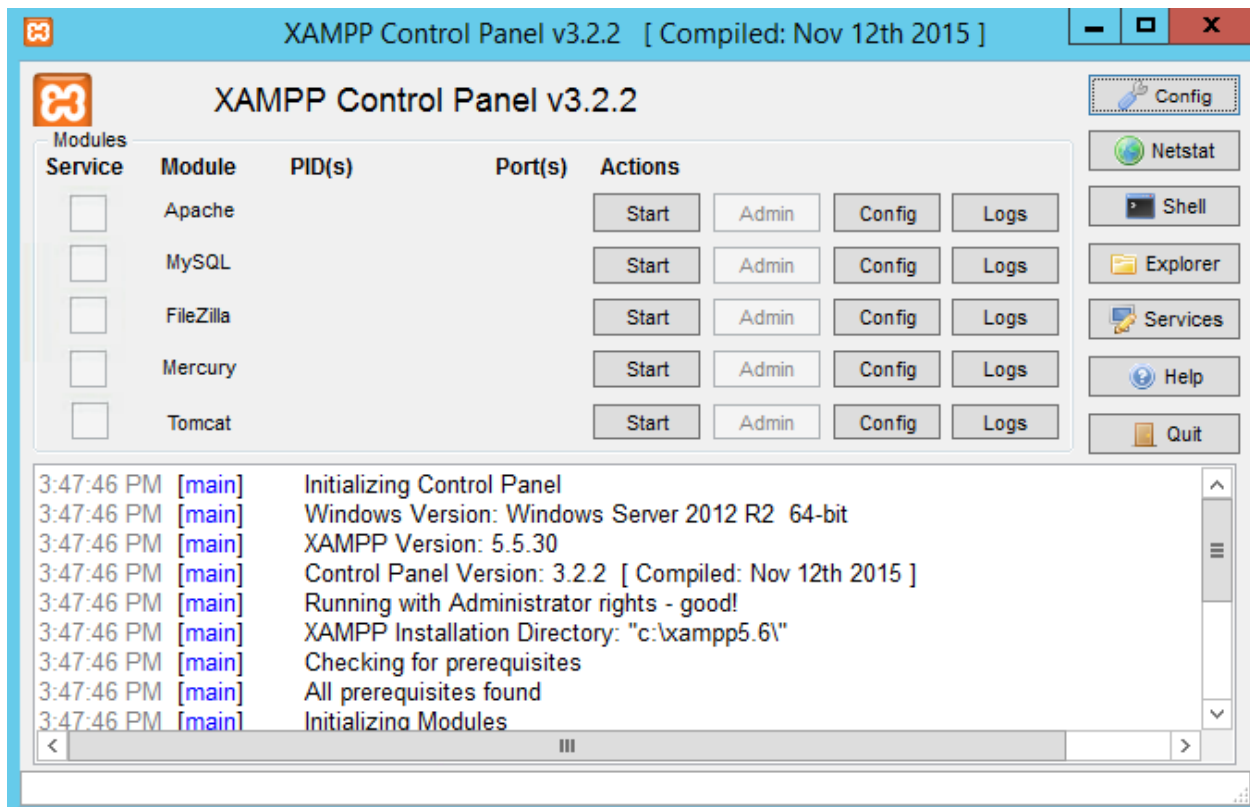
Χρησιμοποιώντας την ευελιξία του Gradle, μπορείτε να τα επιτύχετε όλα αυτά χωρίς να τροποποιήσετε τα βασικά αρχεία πηγής της εφαρμογής σας. Τα αρχεία κατασκευής του Android Studio ονομάζονται `build.gradle`. Είναι αρχεία απλού κειμένου που χρησιμοποιούν τη σύνταξη Groovy για τη διαμόρφωση της κατασκευής με στοιχεία που παρέχονται από την προσθήκη Android για Gradle. Κάθε Project έχει ένα αρχείο κατασκευής ανώτατου επιπέδου για ολόκληρο το project και ξεχωριστά αρχεία κατασκευής σε επίπεδο ενότητας για κάθε λειτουργική μονάδα. Όταν εισάγετε ένα υπάρχον έργο, το Android Studio δημιουργεί αυτόματα τα απαραίτητα αρχεία κατασκευής.[19]

4.5 Βάση δεδομένων

4.5.1 XAMPP server

Το XAMPP είναι μια συντομογραφία όπου το X σημαίνει Cross-Platform, το A σημαίνει Apache, το M σημαίνει MySQL και το Ps σημαίνει PHP και Perl, αντίστοιχα.

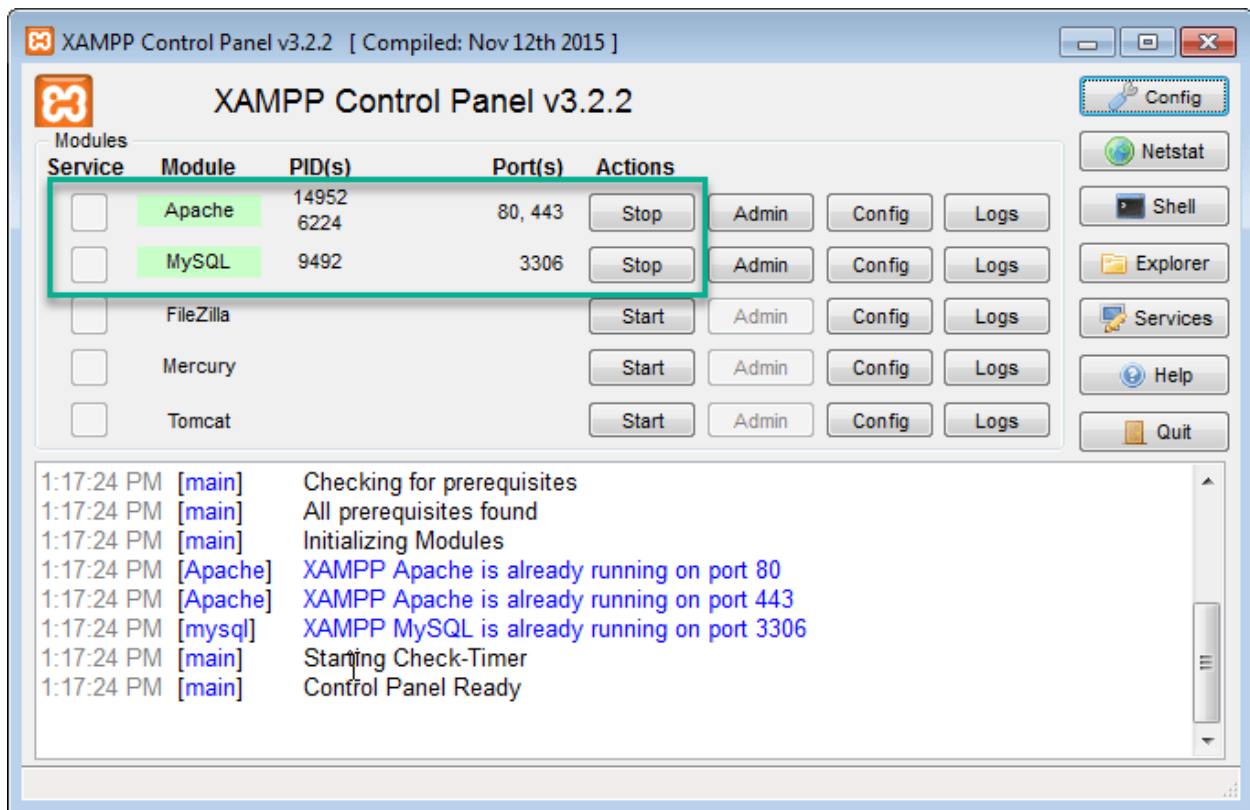
Είναι ένα πακέτο λύσεων ιστού ανοιχτού κώδικα που περιλαμβάνει διανομή Apache για πολλούς διακομιστές και εκτελέσιμα στοιχεία γραμμής εντολών μαζί με λειτουργικές μονάδες όπως διακομιστής Apache, MariaDB, PHP και Perl. Το XAMPP βοηθά έναν τοπικό κεντρικό υπολογιστή ή διακομιστή να δοκιμάσει τον ιστότοπο και τους πελάτες του μέσω υπολογιστών και φορητών υπολογιστών πριν τον αποδεσμεύσει στον κύριο διακομιστή. Είναι μια πλατφόρμα που παρέχει ένα κατάλληλο περιβάλλον για δοκιμή και επαλήθευση της λειτουργίας έργων που βασίζονται σε Apache, Perl, MySQL βάση δεδομένων και PHP μέσω του συστήματος του ίδιου του host. Μεταξύ αυτών των τεχνολογιών, η Perl είναι μια γλώσσα προγραμματισμού που χρησιμοποιείται για την ανάπτυξη Ιστού, η PHP είναι μια γλώσσα δέσμης ενεργειών υποστήριξης και η MariaDB είναι η πιο έντονα χρησιμοποιούμενη βάση δεδομένων που αναπτύχθηκε από τη MySQL. Η λεπτομερής περιγραφή αυτών των στοιχείων δίνεται παρακάτω.[23]



Εικόνα 41: XAMPP control panel

4.5.2 phpMyAdmin

Για να αποκτήσετε πρόσβαση στο phpMyAdmin από το XAMPP, θα χρειαστεί να βεβαιωθείτε ότι έχετε το Apache και το MySQL που εκτελούνται στον πίνακα ελέγχου XAMPP, κάνοντας κλικ στα κουμπιά έναρξης στη στήλη Ενέργειες:



Εικόνα 42: XAMPP control panel, start Apache and MySQL

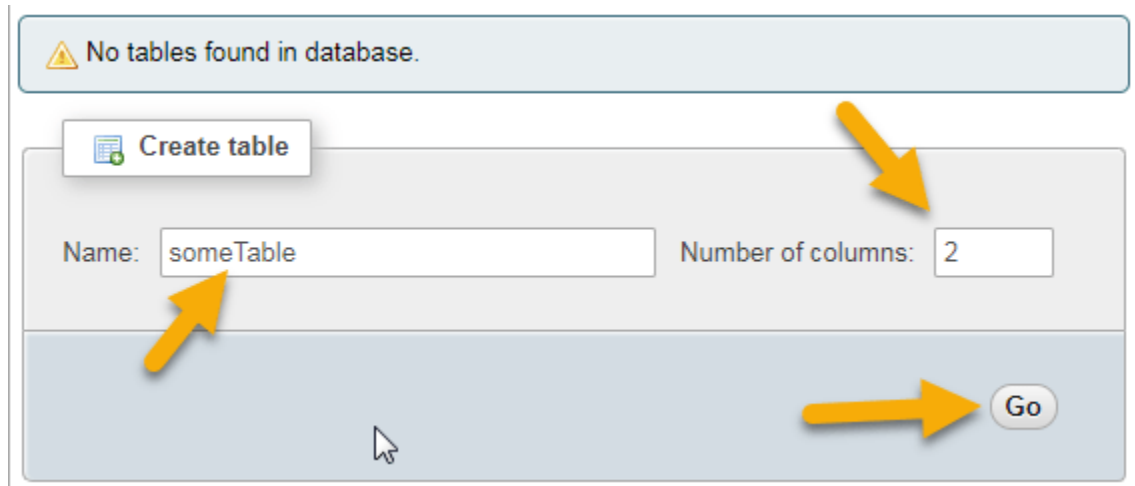
Έπειτα αποκτήσετε πρόσβαση στο phpMyAdmin πληκτρολογώντας <http://localhost/phpmyadmin/> στο πρόγραμμα περιήγησής σας. Την πρώτη φορά που θα αποκτήσετε πρόσβαση, θα χρειαστεί να συνδεθείτε χρησιμοποιώντας το "root" ως όνομα χρήστη και δεν θα υπάρχει κωδικός πρόσβασης.

Αφού συνδεθείτε θα θελήσετε να δημιουργήσετε μια νέα βάση δεδομένων κάνοντας κλικ εδώ:



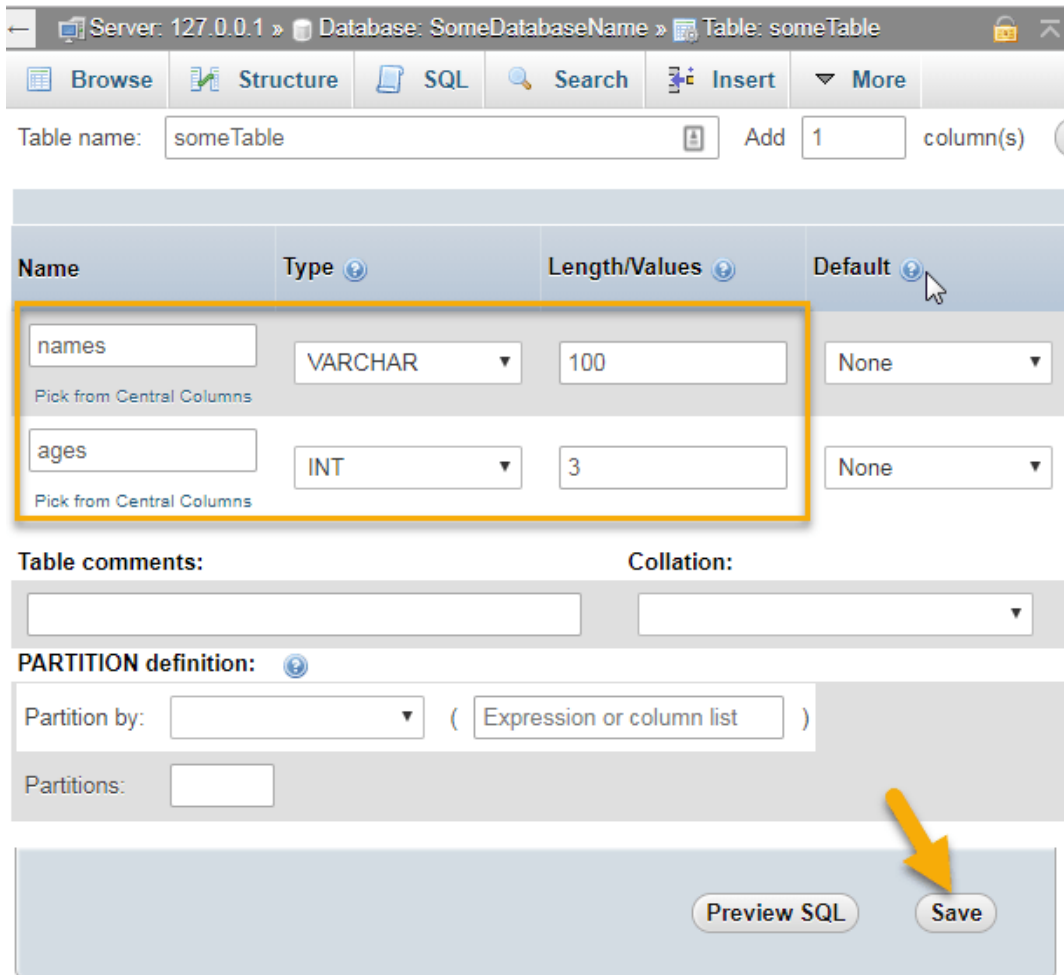
Εικόνα 43: phpMyAdmin, create new database

Στη συνέχεια θα σας ζητηθεί να δημιουργήσετε έναν πίνακα με όσες στήλες θέλετε. Μόλις το αποφασίσετε, κάντε ξανά κλικ στο «Μετάβαση»:



Εικόνα 44: phpMyAdmin, create new table

Αυτό θα απαιτήσει να σχεδιάσετε λίγο εκ των προτέρων τη βάση δεδομένων σας. Θα πρέπει να γνωρίζετε ακριβώς τι θα αποθηκεύσετε σε αυτό.



Εικόνα 45: phpMyAdmin, create database columns

4.5.3 MYSQL/PHP connection

Η MYSQL χρησιμοποιείται ως βάση δεδομένων στον webserver και η PHP χρησιμοποιείται για την ανάκτηση δεδομένων από τη βάση δεδομένων. Η εφαρμογή μας θα επικοινωνήσει με τη σελίδα PHP με τις απαραίτητες παραμέτρους και η PHP θα επικοινωνήσει με τη βάση δεδομένων MYSQL και θα ανακτήσει το αποτέλεσμα μετά θα μας επιστρέψει τα αποτελέσματα.

Για την σύνδεση στην βάση δεδομένων:

```
<?php

class DbConnect{

    private $con;

    function __construct(){
    }

    function connect(){
        include_once dirname(__FILE__).'/Constants.php';
        $this->con = new mysqli(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);

        if(mysqli_connect_errno()){
            echo "Failed to connect with database".mysqli_connect_err();
        }

        return $this->con;
    }

}

?>
```

CRUD operations (some examples):

Δημιουργία ενός καινούριου NPC:

```
function createNpc($npcname, $npclat, $npclng, $userid,$npcwelcome){

    $stmt = $this->con->prepare("INSERT INTO npcs (npcname, npclat, npclng,
userid,npcwelcome) VALUES (?, ?, ?, ?, ?)");
    $stmt->bind_param("sddis", $npcname, $npclat, $npclng,
$userid,$npcwelcome);
    if($stmt->execute())
        $last_id = $stmt->insert_id;
    return $last_id;
    return false;
}
```


Για να πάρουμε τα NPC από την βάση:

```
public function getAllNpcs(){
    $stmt = $this->con->prepare("SELECT npcname, npclat, npclng,
userid,npcid,npcwelcome,npcllevel FROM npcs");
    $stmt->execute();
    $stmt->bind_result($npcname, $npclat, $npclng,
$userid,$npcid,$npcwelcome,$npclevel);

    $results = array();

    while($stmt->fetch()){
        $result = array();
        $result['npcid'] = $npcid;
        $result['userid'] = $userid;
        $result['npclng'] = $npclng;
        $result['npclat'] = $npclat;
        $result['npcname'] = $npcname;
        $result['npcwelcome'] = $npcwelcome;
        $result['npclevel'] = $npclevel;

        array_push($results, $result);
    }
    return $results;
}
```

Διαγραφή NPC:

```
public function deleteNpc($npcid){
    $stmt = $this->con->prepare("DELETE FROM npcs WHERE npcid = ?
");
    $stmt->bind_param("i", $npcid);
    if($stmt->execute())
        return true;

    return false;
}
```

Και τέλος για τον απαραίτητο έλεγχο για το αν υπάρχουν οι απαιτούμενες μεταβλητές και τα μηνύματα απόκτησης σε μορφή JSON:

```
case 'createnpc':
    //first check the parameters required for this request are
    available or not
    isTheseParametersAvailable(array('npcname', 'npclat',
    'npclng', 'userid', 'npcwelcome'));

    $db = new DbOperation();

    $result = $db->createNpc(
    $_POST['npcname'],
    $_POST['npclat'],
    $_POST['npclng'],
    $_POST['userid'],
    $_POST['npcwelcome']
    );

    if($result){
    $response['error'] = false;
    $response['message'] = 'npc addedd successfully';
    $response['results'] = $result;
    }else{

    $response['error'] = true;
    $response['message'] = 'Some error occurred please try
again';
    }

    break;
```

Από την μεριά του Android χρησιμοποιώντας την βιβλιοθήκη Volley:

```
public static void createNpc(String npcName, String npcWelcome, String
npcLat, String npcLng, int userId, VolleyCallback callback) {

    StringRequest stringRequest = new StringRequest(Request.Method.POST,
Constants.URL_NPC_CREATE,
    response -> {
        try {
            //create json object and check if error
            JSONObject jsonResponse = new JSONObject(response);
            if (jsonResponse != null &&
```

```

!jsonResponse.getBoolean("error")) {
    callback.onSuccess(jsonResponse);
} else {
    Log.d("getNPCs Message: ",
jsonResponse.getString("message"));
}
} catch (
    JSONException e) {
    e.printStackTrace();
}
},
error -> {
    Log.e("getNPCs ErrorMessage: ", error.getMessage());
    callback.onError(error.getMessage());
}) {
@Override
protected Map<String, String> getParams() throws AuthFailureError {
    Map<String, String> params = new HashMap<>();
    params.put("npcname", npcName);
    params.put("npcwelcome", npcWelcome);
    params.put("npclat", npcLat);
    params.put("npclng", npcLng);
    params.put("userid", String.valueOf(userId));
    return params;
}
};
VolleyRequestHandler.addToRequestQueue(stringRequest);
}

```

4.6 APIS - application programming interface

Μια διεπαφή προγραμματισμού εφαρμογών ή API, επιτρέπει στις εταιρείες να ανοίγουν τα δεδομένα και τη λειτουργικότητα των εφαρμογών τους σε τρίτους προγραμματιστές, επιχειρηματικούς συνεργάτες και εσωτερικά τμήματα των εταιρειών τους. Αυτό επιτρέπει στις υπηρεσίες και τα προϊόντα να επικοινωνούν μεταξύ τους και να αξιοποιούν τα δεδομένα και τη λειτουργικότητα του άλλου μέσω μιας τεκμηριωμένης διεπαφής. Οι προγραμματιστές δεν χρειάζεται να γνωρίζουν πώς υλοποιείται ένα API. χρησιμοποιούν απλώς τη διεπαφή για να επικοινωνούν με προϊόντα και υπηρεσίες. Η χρήση API έχει αυξηθεί την τελευταία δεκαετία, σε βαθμό που πολλές από τις πιο δημοφιλείς εφαρμογές σήμερα δεν θα ήταν δυνατές χωρίς τα API.

4.6.1 Πώς λειτουργούν τα APIS

Ένα API είναι ένα σύνολο καθορισμένων κανόνων που εξηγούν πώς οι υπολογιστές ή οι εφαρμογές επικοινωνούν μεταξύ τους. Τα API βρίσκονται μεταξύ μιας εφαρμογής και του διακομιστή web, λειτουργώντας ως ενδιάμεσο επίπεδο που επεξεργάζεται τη μεταφορά δεδομένων μεταξύ συστημάτων.

Πώς λειτουργεί ένα API:

1. **Μια εφαρμογή πελάτη ξεκινά μια κλήση API για την ανάκτηση πληροφοριών**—γνωστή και ως αίτημα. Αυτό το αίτημα υποβάλλεται σε επεξεργασία από μια εφαρμογή στον διακομιστή ιστού μέσω του Uniform Resource Identifier (URI) του API και περιλαμβάνει ένα ρήμα αιτήματος, κεφαλίδες και μερικές φορές ένα σώμα αιτήματος.
2. **Μετά τη λήψη ενός έγκυρου αιτήματος**, το API πραγματοποιεί μια κλήση στο εξωτερικό πρόγραμμα ή διακομιστή ιστού.
3. **Ο διακομιστής στέλνει μια απάντηση** στο API με τις ζητούμενες πληροφορίες.
4. **Το API μεταφέρει τα δεδομένα** στην αρχική αίτηση αίτησης.

Ενώ η μεταφορά δεδομένων θα διαφέρει ανάλογα με την υπηρεσία ιστού που χρησιμοποιείται, αυτή η διαδικασία αιτημάτων και απαντήσεων γίνεται μέσω ενός API. Μια διεπαφή χρήστη έχει σχεδιαστεί για χρήση από ανθρώπους, τα API έχουν σχεδιαστεί για χρήση από υπολογιστή ή εφαρμογή.

Τα API προσφέρουν ασφάλεια από το σχεδιασμό, επειδή η θέση τους ως μεσάζων διευκολύνει την αφαίρεση της λειτουργικότητας μεταξύ δύο συστημάτων - το τελικό σημείο API αποσυνδέει την καταναλωτική εφαρμογή από την υποδομή που παρέχει την υπηρεσία. Οι κλήσεις API περιλαμβάνουν συνήθως διαπιστευτήρια εξουσιοδότησης για τη μείωση του κινδύνου επιθέσεων στον διακομιστή και μια πύλη API μπορεί να περιορίσει την πρόσβαση για να ελαχιστοποιήσει τις απειλές ασφαλείας. Επίσης, κατά τη διάρκεια της ανταλλαγής, οι κεφαλίδες HTTP, τα cookie ή οι παράμετροι συμβολοσειράς ερωτήματος παρέχουν πρόσθετα επίπεδα ασφαλείας στα δεδομένα.

Για παράδειγμα, εξετάστε ένα API που προσφέρεται από μια υπηρεσία επεξεργασίας πληρωμών. Οι πελάτες μπορούν να εισάγουν τα στοιχεία της κάρτας τους στο μπροστινό μέρος μιας εφαρμογής για ένα κατάστημα ηλεκτρονικού εμπορίου. Ο επεξεργαστής πληρωμών δεν απαιτεί πρόσβαση στον τραπεζικό λογαριασμό του χρήστη. Το API δημιουργεί ένα μοναδικό διακριτικό για αυτήν τη συναλλαγή και το περιλαμβάνει στην κλήση API προς τον διακομιστή. Αυτό εξασφαλίζει υψηλότερο επίπεδο Main libraries

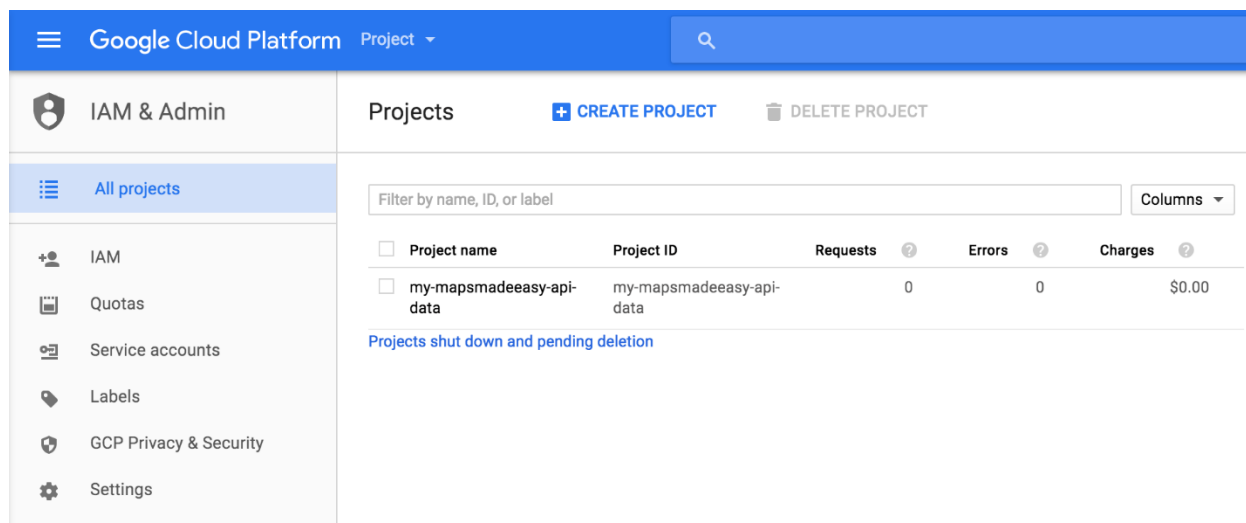
4.6.2 Google Maps Platform

Η Πλατφόρμα Χαρτών Google είναι ένα σύνολο API και SDK που επιτρέπει στους προγραμματιστές να ενσωματώνουν τους Χάρτες Google σε εφαρμογές για κινητά και ιστοσελίδες ή να ανακτούν δεδομένα από τους Χάρτες Google.

Βήματα για να ξεκινήσετε να χρησιμοποιείτε οποιοδήποτε από τα API και SDK στην Πλατφόρμα Χαρτών Google:

1. Set up your Project:

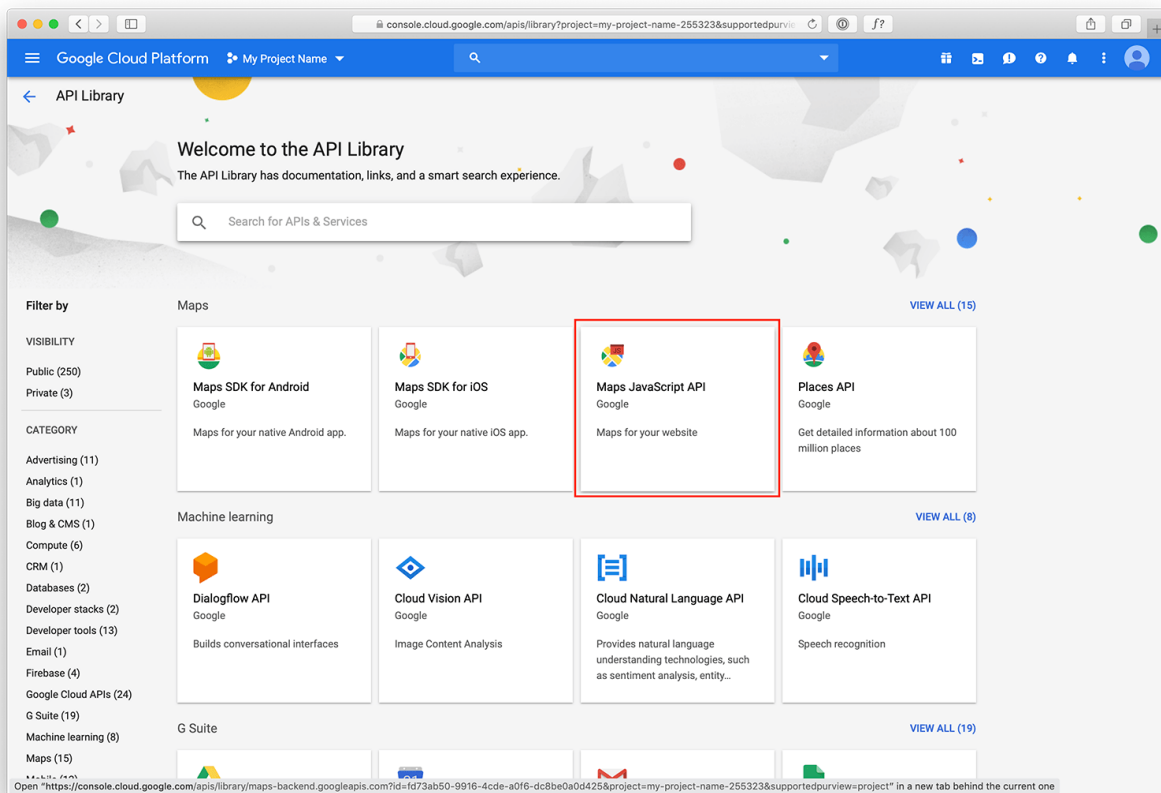
- 1.1. Στο Google Cloud Console, στη σελίδα επιλογή έργου, κάντε κλικ στην επιλογή Δημιουργία έργου για να ξεκινήσετε τη δημιουργία ενός νέου έργου Cloud.
- 1.2. Βεβαιωθείτε ότι η χρέωση είναι ενεργοποιημένη για το έργο σας στο Cloud.[21]



Εικόνα 46: Google Cloud Console, project selector page [22]

2. Ενεργοποιήστε τα API ή τα SDK που σκοπεύετε να χρησιμοποιήσετε με το έργο σας:

- 2.1. Δείτε τα API και SDK της Πλατφόρμας Χαρτών Google που μπορείτε να ενεργοποιήσετε μεταβαίνοντας στη σελίδα Βιβλιοθήκη API Χαρτών στο Cloud Console.
- 2.2. Μεταβείτε στη σελίδα Βιβλιοθήκη API Χαρτών Κάντε κλικ στο API ή το SDK που θέλετε να ενεργοποιήσετε.
 - 2.2.1. Εάν το κουμπί λέει ΕΝΕΡΓΟΠΟΙΗΣΗ, κάντε κλικ στο κουμπί για να ενεργοποιήσετε το API ή το SDK.
 - 2.2.2. Εάν το κουμπί λέει ΔΙΑΧΕΙΡΙΣΗ, το API ή το SDK είναι ήδη ενεργοποιημένο και δεν χρειάζεται να κάνετε τίποτα περαιτέρω.
 - 2.2.3. Κάνοντας κλικ σε οποιοδήποτε κουμπί θα εμφανιστεί ο πίνακας εργαλείων για το API ή το SDK. (Κάντε κλικ στο κουμπί ΑΠΕΝΕΡΓΟΠΟΙΗΣΗ για να καταργήσετε το API ή το SDK από αυτό το έργο.)[21]



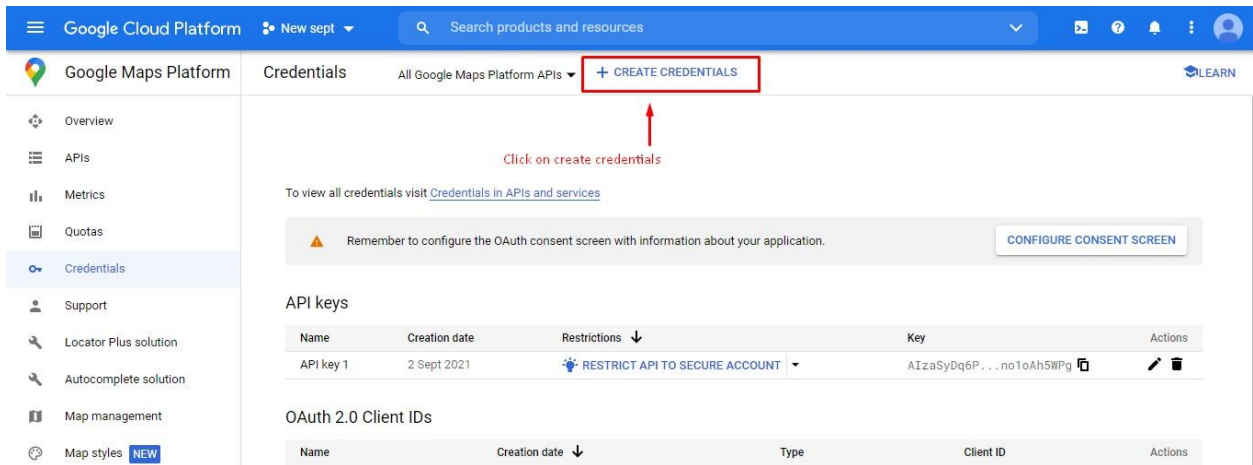
Εικόνα 47: Google Cloud Console, Maps API Library page [22]

3. Δημιουργήστε ένα κλειδί API:

3.1. Μεταβείτε στη σελίδα Πλατφόρμα Χαρτών Google > Διαπιστευτήρια.

3.2. Μεταβείτε στη σελίδα Διαπιστευτήρια Στη σελίδα Διαπιστευτήρια, κάντε κλικ στην επιλογή Δημιουργία διαπιστευτηρίων > Κλειδί API. Το παράθυρο διαλόγου δημιουργίας κλειδιού API εμφανίζει το νέο κλειδί API που δημιουργήσατε.

3.3. Κάντε κλικ στο Κλείσιμο. Το νέο κλειδί API παρατίθεται στη σελίδα Διαπιστευτήρια κάτω από τα κλειδιά API. (Θυμηθείτε να περιορίσετε το κλειδί API πριν το χρησιμοποιήσετε στην παραγωγή.)[21]



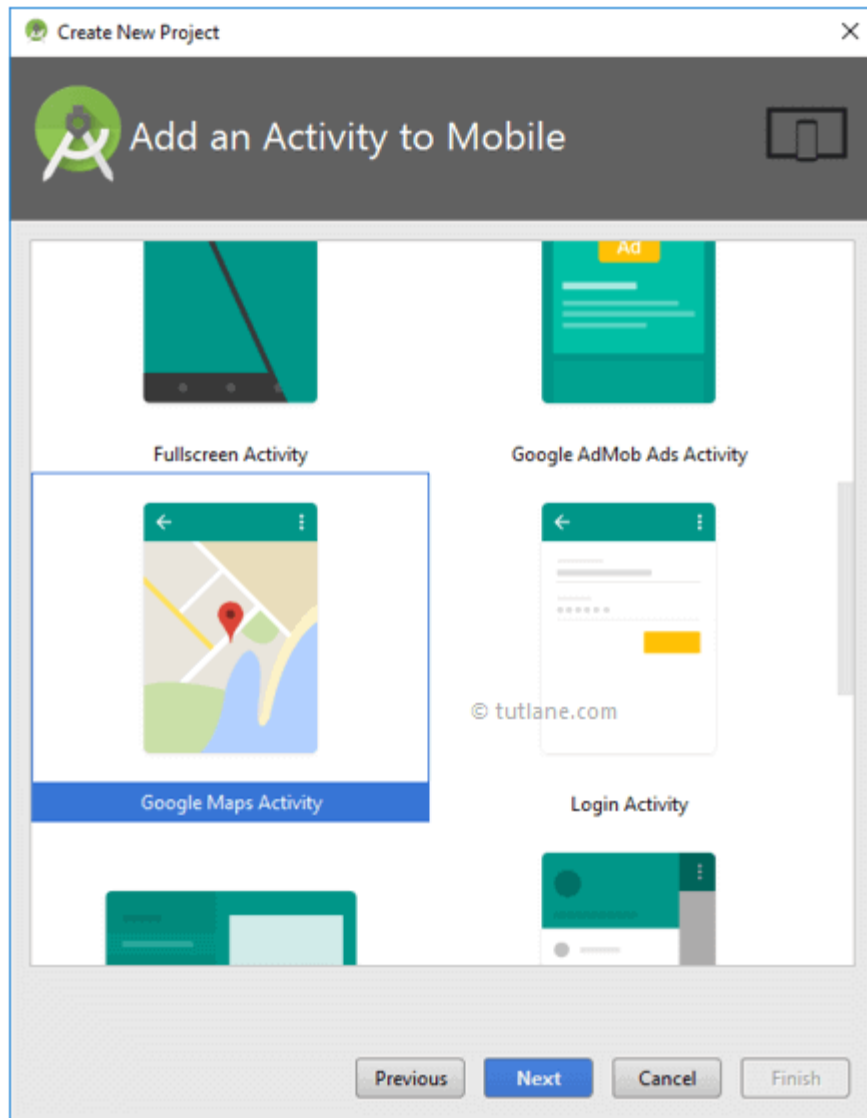
Εικόνα 48: Google Cloud Console, Credentials page [22]

4.6.3 Maps SDK for Android

Με το SDK Χαρτών για Android, προσθέστε χάρτες στην εφαρμογή σας Android, συμπεριλαμβανομένων των εφαρμογών Wear OS χρησιμοποιώντας δεδομένα Χαρτών Google, εμφανίσεις χαρτών και αποκρίσεις με κινήσεις χάρτη. Μπορείτε επίσης να παρέχετε πρόσθετες πληροφορίες για τοποθεσίες του χάρτη και να υποστηρίξετε την αλληλεπίδραση των χρηστών προσθέτοντας δείκτες, πολύγωνα και επικαλύψεις στον χάρτη σας. Το SDK υποστηρίζει και τις δύο γλώσσες προγραμματισμού Kotlin και Java και παρέχει πρόσθετες βιβλιοθήκες και επεκτάσεις για προηγμένες δυνατότητες και τεχνικές προγραμματισμού. [21]

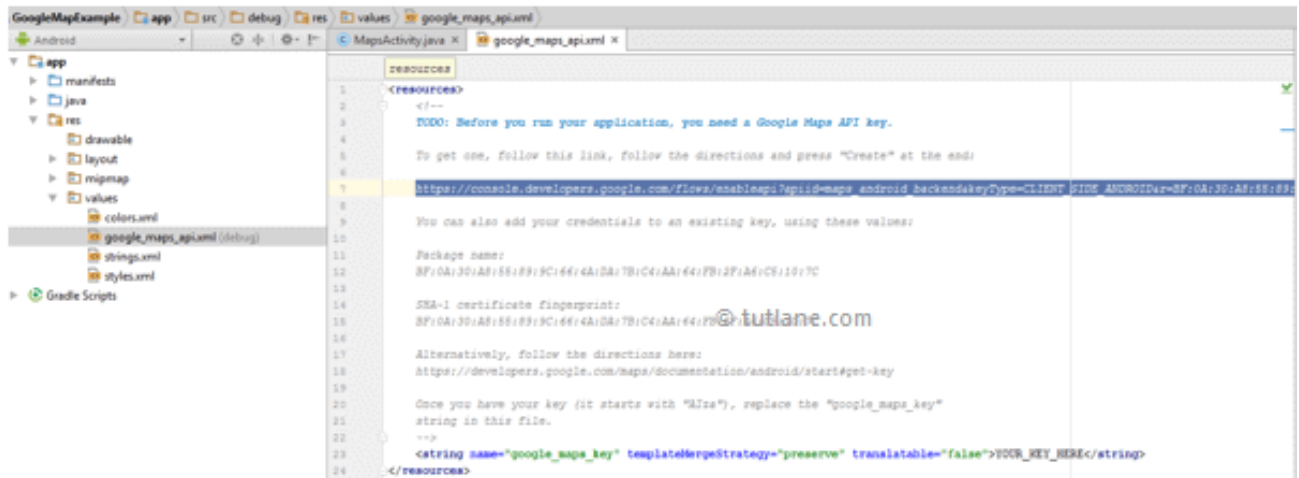
4.6.3.1 Ενσωμάτωση χαρτών σε ένα project.

Για να ενσωματώσει κανείς έναν χάρτη της Google στο project του δεν χρειάζεται παρά να δημιουργήσει ένα καινούριο **Google Maps Activity**.



Εικόνα 49: Android Studio, add Google Maps Activity

Μόλις δημιουργηθεί το project, το Android Studio θα ανοίξει τα αρχεία **google_maps_api.xml** και **MapsActivity.java** στο πρόγραμμα επεξεργασίας. Το αρχείο **google_maps_api.xml** θα περιέχει οδηγίες για τη δημιουργία ενός κλειδιού API Χαρτών Google για πρόσβαση στους διακομιστές των Χαρτών Google. Αντιγράψτε τον σύνδεσμο που παρέχεται στο αρχείο **google_maps_api.xml** όπως φαίνεται παρακάτω.[21]



Εικόνα 50: Android Studio, google_maps_api.xml

Τώρα αντιγράψτε το **κλειδί API**, επιστρέψτε στο android studio και επικολλήστε το κλειδί API στο στοιχείο <string> στο αρχείο **google_maps_api.xml** όπως φαίνεται παρακάτω.

```
<string name="google_maps_key" templateMergeStrategy="preserve" translatable="false">AIzaSyCKPTaBv41DKqr9qxMPWOQAsqp0Q4NHMER</string>
```

Activity_maps.xml

Το αρχείο XML (activity_maps.xml) που καθορίζει τη διάταξη της εφαρμογής βρίσκεται στο res/layout/ περιέχει τον ακόλουθο κώδικα.

```
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.tutlane.googlemapexample.MainActivity" />
```

AndroidManifest.xml

Το αρχείο δήλωσης της εφαρμογής μας (AndroidManifest.xml) θα περιέχει τον κώδικα όπως φαίνεται παρακάτω με το απαιτούμενο κλειδί API Χαρτών Google και τα δικαιώματα. <?xml version="1.0" encoding="utf-8"?>

```

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.tutlane.googlemapexample">

    <uses-
permission android:name="android.permission.ACCESS_FINE_LOCATION" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="@string/google_maps_key" />
        <activity
            android:name=".MapsActivity"
            android:label="@string/title_activity_maps">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER"
/>>
            </intent-filter>
        </activity>
    </application>
</manifest>

```

MapsActivity.java

Το αρχείο Java (MapsActivity.java) που ορίζει τη δραστηριότητα των χαρτών θα περιέχει τον ακόλουθο κώδικα.

```

package com.tutlane.googlemapexample;
import android.support.v4.app.FragmentActivity;
import android.os.Bundle;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

public
class MapsActivity extends FragmentActivity implements OnMapReadyCallback

```

```

{
    private GoogleMap mMap;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        // Obtain the SupportMapFragment and get notified when the map is
        ready to be used.
        SupportMapFragment mapFragment = (SupportMapFragment)
        getSupportFragmentManager()
            .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
    }
    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
        // Add a marker in hyderabad and move the camera
        LatLng hyderabad = new LatLng(17, 78);
        mMap.addMarker(new MarkerOptions().position(hyderabad).title("Tut
lane in India"));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(hyderabad));
    }
}

```

Όταν εκτελούμε το παραπάνω παράδειγμα χρησιμοποιώντας μια εικονική συσκευή android (AVD) θα έχουμε ένα αποτέλεσμα όπως φαίνεται παρακάτω.



Εικόνα 51: Android Studio, google map

4.6.4 The Directions API

Το Directions API είναι μια υπηρεσία ιστού που χρησιμοποιεί ένα αίτημα HTTP για να επιστρέψει οδηγίες με μορφή JSON ή XML μεταξύ τοποθεσιών. Μπορείτε να λάβετε οδηγίες για διάφορους τρόπους μεταφοράς, όπως συγκοινωνία, οδήγηση, περπάτημα ή ποδήλατο. Το API επιστρέφει τις πιο αποτελεσματικές διαδρομές κατά τον υπολογισμό των οδηγιών. Ο χρόνος ταξιδιού είναι ο κύριος παράγοντας που έχει βελτιστοποιηθεί, αλλά το API μπορεί επίσης να λαμβάνει υπόψη άλλους παράγοντες όπως η απόσταση, ο αριθμός στροφών και πολλοί άλλοι όταν αποφασίζει ποια διαδρομή είναι η πιο αποτελεσματική.[21]

4.6.4.1 Δημιουργία των αιτημάτων στο Directions API

Ένα αίτημα Directions API έχει την ακόλουθη μορφή:

<https://maps.googleapis.com/maps/api/directions/outputFormat?parameters>

όπου outputFormat μπορεί να είναι μία από τις ακόλουθες τιμές:

- json (συνιστάται) υποδεικνύει έξοδο σε σημειογραφία αντικειμένου JavaScript (JSON)
- Το xml υποδεικνύει την έξοδο ως XML[21]

4.7 Λοιπές τεχνολογίες

4.7.1 Shared Preferences

Το Android παρέχει πολλούς τρόπους αποθήκευσης δεδομένων μιας εφαρμογής. Ένας από αυτούς τους τρόπους ονομάζεται Shared Preferences. Οι Κοινόχρηστες Προτιμήσεις σας επιτρέπουν να αποθηκεύετε και να ανακτάτε δεδομένα με τη μορφή ζεύγους κλειδιών, τιμών.

Για να χρησιμοποιήσετε κοινόχρηστες προτιμήσεις, πρέπει να καλέσετε μια μέθοδο `getSharedPreferences()` που επιστρέφει μια παρουσία `SharedPreferences` που δείχνει το αρχείο που περιέχει τις τιμές των προτιμήσεων.[19]

```
public void userLogin(int id, String username) {  
  
    SharedPreferences sharedPreferences =  
mCtx.getSharedPreferences(SHARED_PREF_NAME, Context.MODE_PRIVATE);  
    SharedPreferences.Editor editor = sharedPreferences.edit();  
    editor.putInt(KEY_USER_ID, id);  
    editor.putString(KEY_USERNAME, username);  
    editor.apply();  
}
```

4.7.2 Volley

Το Volley είναι μια βιβλιοθήκη HTTP που κάνει τη δικτύωση για εφαρμογές Android ευκολότερη και το πιο σημαντικό, ταχύτερη. Το Volley είναι διαθέσιμο στο GitHub.

Το Volley προσφέρει τα ακόλουθα οφέλη:

- Αυτόματος προγραμματισμός αιτημάτων δικτύου.
- Πολλαπλές ταυτόχρονες συνδέσεις δικτύου.
- Διαφανής αποθήκευση απόκρισης δίσκου και μνήμης με τυπική συνοχή κρυφής μνήμης HTTP.
- Υποστήριξη για ιεράρχηση αιτημάτων.
- API αιτήματος ακύρωσης. Μπορείτε να ακυρώσετε ένα μόνο αίτημα ή μπορείτε να ορίσετε μπλοκ ή εύρος αιτημάτων για ακύρωση.
- Ευκολία προσαρμογής, για παράδειγμα, για επανάληψη και επιστροφή.

- Ισχυρή παραγγελία που διευκολύνει τη σωστή συμπλήρωση του UI σας με δεδομένα που λαμβάνονται ασύγχρονα από το δίκτυο.
- Εργαλεία εντοπισμού σφαλμάτων και ανίχνευσης.

Το Volley υπερέχει σε λειτουργίες τύπου RPC που χρησιμοποιούνται για τη συμπλήρωση μιας διεπαφής χρήστη, όπως η ανάκτηση μιας σελίδας αποτελεσμάτων αναζήτησης ως δομημένα δεδομένα. Ενσωματώνεται εύκολα με οποιοδήποτε πρωτόκολλο και βγαίνει από το κουτί με υποστήριξη για ακατέργαστες συμβολοσειρές, εικόνες και JSON. Παρέχοντας ενσωματωμένη υποστήριξη για τις λειτουργίες που χρειάζεστε, το Volley σας απαλλάσσει από τη σύνταξη κώδικα boilerplate και σας επιτρέπει να επικεντρωθείτε στη λογική που είναι συγκεκριμένη για την εφαρμογή σας.

Το Volley δεν είναι κατάλληλο για μεγάλες λειτουργίες λήψης ή ροής, καθώς το Volley διατηρεί όλες τις αποκρίσεις στη μνήμη κατά την ανάλυση.

Η βασική βιβλιοθήκη Volley έχει αναπτυχθεί στο GitHub και περιέχει τον κύριο αγωγό αποστολής αιτημάτων καθώς και ένα σύνολο κοινών βοηθητικών προγραμμάτων, που είναι διαθέσιμα στην "εργαλειοθήκη" του Volley. Ο ευκολότερος τρόπος για να προσθέσετε το Volley στο έργο σας είναι να προσθέσετε την ακόλουθη εξάρτηση στο αρχείο build.gradle της εφαρμογής σας:[19]

```
dependencies {
    ...
    implementation 'com.android.volley:volley:1.2.1'
}
```

παράδειγμα ενός Volley Request φαίνεται στο κεφάλαιο 4.5.3

4.7.3 Alert Dialog

Ένα Dialog είναι ένα μικρό παράθυρο που προτρέπει τον χρήστη να αποφασίσει ή να εισαγάγει πρόσθετες πληροφορίες.

Μερικές φορές στην αίτησή σας, εάν θέλετε να ρωτήσετε τον χρήστη σχετικά με τη λήψη μιας απόφασης μεταξύ ναι ή όχι ως απάντηση σε οποιαδήποτε συγκεκριμένη ενέργεια που έχει κάνει ο χρήστης, παραμένοντας στην ίδια δραστηριότητα και χωρίς να αλλάξετε την οθόνη, μπορείτε να χρησιμοποιήσετε το Alert Dialog.

Για να δημιουργήσετε ένα Alert Dialog, πρέπει να δημιουργήσετε ένα αντικείμενο του AlertDialog.Builder που είναι μια εσωτερική κλάση του AlertDialog. Η σύνταξή του δίνεται παρακάτω.[23]

```
AlertDialog.Builder alertDialogBuilder = new
AlertDialog.Builder(this);
```

Τώρα πρέπει να ορίσετε το θετικό (ναι) ή το αρνητικό (όχι) κουμπί χρησιμοποιώντας το αντικείμενο της κλάσης AlertDialogBuilder. Η σύνταξή του είναι:

```
AlertDialogBuilder.setPositiveButton(CharSequence text,
    DialogInterface.OnClickListener listener)
AlertDialogBuilder.setNegativeButton(CharSequence text,
    DialogInterface.OnClickListener listener)
```

Εκτός από αυτό, μπορείτε να χρησιμοποιήσετε άλλες λειτουργίες που παρέχονται από την κλάση builder για να προσαρμόσετε το παράθυρο διαλόγου ειδοποίησης.[24]

```
//delete npc dialog
public void deleteNPCDialog(int npcId,int position){
    new AlertDialog.Builder(this)
        .setTitle("Delete person")
        .setMessage("This person will be deleted! Do you want to
proceed?")
        .setPositiveButton("Yes", (paramDialogInterface, paramInt) -> {
            // delete npc
            VolleyRequests.deleteNPC(npcId, userId);
            //delete places of npc
            VolleyRequests.deletePlaces(npcId);
            // delete texts of npc
            deleteTexts(npcId);
            // remove item from list
            myNPCList.remove(position);
            npcAdapter.notifyDataSetChanged();
        })
        .setNegativeButton("Cancel",null)
        .show();
}
```

4.7.4 Array Adapter

Μπορείτε να χρησιμοποιήσετε αυτόν τον Adapter για να παρέχετε προβολές για ένα View, Επιστρέφει μια προβολή για κάθε αντικείμενο σε μια συλλογή αντικειμένων δεδομένων που παρέχετε και μπορεί να χρησιμοποιηθεί με γραφικά στοιχεία διεπαφής χρήστη που βασίζονται σε λίστα, όπως το ListView ή το Spinner.

Από προεπιλογή, ο ArrayAdapter δημιουργεί μια προβολή καλώντας το Object#toString() σε κάθε αντικείμενο δεδομένων στη συλλογή που παρέχετε και τοποθετεί το αποτέλεσμα σε ένα TextView.

Μπορείτε επίσης να προσαρμόσετε τον τύπο προβολής που χρησιμοποιείται για το αντικείμενο δεδομένων στη συλλογή. Για να προσαρμόσετε τον τύπο προβολής που χρησιμοποιείται για το αντικείμενο δεδομένων, παρακάμψτε το `getView(int, android.view.View, android.view.ViewGroup)` και διογκώστε έναν πόρο προβολής.[19]

```
public class NpcAdapter extends ArrayAdapter<Npc> {
    public NpcAdapter(Context context, List<Npc> users) {
        super(context, 0, users);
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        // Get the data item for this position
        Npc npc = getItem(position);
        // Check if an existing view is being reused, otherwise inflate the
view
        if (convertView == null) {
            convertView =
LayoutInflater.from(getContext()).inflate(R.layout.npc_list, parent, false);
        }
        // Lookup view for data population
        TextView textViewNpcName = (TextView)
convertView.findViewById(R.id.npcNameTextView);
        ImageView npcImageView = (ImageView)
convertView.findViewById(R.id.npcImageView);

        // Populate the data into the template view using the data object
        textViewNpcName.setText(npc.getNpcname());

        if(npc.getNpclevel() <= 30) {
            npcImageView.setImageResource(R.drawable.person_red);
        }
        else if (npc.getNpclevel() > 30 && npc.getNpclevel() <80){
            npcImageView.setImageResource(R.drawable.person_orange);
        }
        else {
            npcImageView.setImageResource(R.drawable.person_green);
        }

        // Return the completed view to render on screen
        return convertView;
    }
}
```

4.7.5 JSON

Το JSON (JavaScript Object Notation) είναι μια ελαφριά μορφή ανταλλαγής δεδομένων. Είναι εύκολο για τους ανθρώπους να διαβάζουν και να γράφουν. Είναι εύκολο για τις μηχανές να αναλύουν και να

δημιουργούν. Βασίζεται σε ένα υποσύνολο του προτύπου γλώσσας προγραμματισμού JavaScript ECMA-262 3rd Edition - Δεκέμβριος 1999. Το JSON είναι μια μορφή κειμένου που είναι εντελώς ανεξάρτητη από τη γλώσσα, αλλά χρησιμοποιεί συμβάσεις που είναι γνωστές στους προγραμματιστές της οικογένειας γλωσσών C, συμπεριλαμβανομένης της C, C++, C#, Java, JavaScript, Perl, Python και πολλά άλλα. Αυτές οι ιδιότητες καθιστούν την JSON ιδανική γλώσσα ανταλλαγής δεδομένων.

Το JSON είναι χτισμένο σε δύο δομές:

- Μια συλλογή από ζεύγη ονόματος/τιμών. Σε διάφορες γλώσσες, αυτό πραγματοποιείται ως αντικείμενο, εγγραφή, δομή, λεξικό, πίνακας κατακερματισμού, λίστα με πλήκτρα ή συσχετικός πίνακας.
- Μια ταξινομημένη λίστα τιμών.

Στις περισσότερες γλώσσες, αυτό πραγματοποιείται ως πίνακας, διάνυσμα, λίστα ή ακολουθία. Αυτές είναι καθολικές δομές δεδομένων. Σχεδόν όλες οι σύγχρονες γλώσσες προγραμματισμού τις υποστηρίζουν με τη μία ή την άλλη μορφή.[25]

4.7.6 Connectivity Manager

Κλάση που απαντά σε ερωτήματα σχετικά με την κατάσταση της συνδεσιμότητας δικτύου. Επίσης, ειδοποιεί τις εφαρμογές όταν αλλάζει η συνδεσιμότητα δικτύου.

Οι κύριες αρμοδιότητες αυτής της τάξης είναι:

- Παρακολούθηση συνδέσεων δικτύου (Wi-Fi, GPRS, UMTS, κ.λπ.)
- Αποστολή προθέσεων εκπομπής όταν αλλάζει η συνδεσιμότητα δικτύου
- Παρέχετε ένα API που επιτρέπει στις εφαρμογές να υποβάλλουν ερωτήματα για τη κατάσταση των διαθέσιμων δικτύων
- Παρέχετε ένα API που επιτρέπει στις εφαρμογές να ζητούν και να επιλέγουν δίκτυα για την κίνηση δεδομένων τους

```
// network Availability
private boolean isNetworkAvailable() {
    ConnectivityManager connectivityManager
        = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo activeNetworkInfo =
connectivityManager.getActiveNetworkInfo();
    return activeNetworkInfo != null && activeNetworkInfo.isConnected();
}
```

4.8 Μελλοντικές επεκτάσεις

Στην παρούσα πτυχιακή εργασία μελλοντικές επεκτάσεις εντοπίζονται κυρίως στην εφαρμογή που υλοποιήθηκε. Οι επεκτάσεις αυτές αφορούν τόσο το λειτουργικό/τεχνικό μέρος όσο και τις υπηρεσίες που προσφέρονται.

Λειτουργικές και τεχνικές επεκτάσεις.

Η προσθήκη μίας εσωτερικής βάσης δεδομένων όπως αυτή της SQLite DB θα επιτρέψει την offline λειτουργία της εφαρμογής. Αυτό μπορεί να επιτευχθεί με τον συγχρονισμό των δύο βάσεων δεδομένων (εσωτερικής βάσης SQLite και εξωτερικής βάσης phpMyAdmin) όποτε υπάρχει διαθέσιμο δίκτυο.

Επίσης, η καταγραφή της λειτουργικότητας και εμπειρίας χρήσης που προσφέρει η εφαρμογή θα βοηθήσει στο να αξιολογηθούν τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της και να αντικατασταθούν εάν κριθεί απαραίτητο.

Επεκτάσεις στις υπηρεσίες που παρέχονται.

Πιθανές υπηρεσίες που μπορούν να προστεθούν στην παρούσα εφαρμογή:

- Μετάφραση της εφαρμογής σε πολλές γλώσσες.
- Επέκταση του προφίλ χρήστη , για παράδειγμα με φωτογραφία προφίλ, user trust level.
- Προσθήκη mini games όπως το να μαζεύει κανείς αντικείμενα σε κάποια διαδρομή ή να ολοκληρώνει μια διαδρομή σε κάποιον χρόνο με επιβράβευση εικονίδια για τα NPC του.
- Περεταίρω ανάπτυξη των NPC , εμπλουτισμός των χαρακτηριστικών τους, δυνατότητα επεξεργασίας τους.
- Κοινή βιβλιοθήκη για τα μέρη και για προτάσεις, ώστε να μπορεί ο χρήστης να επιλέξει και μέσα από μέρη άλλων χρηστών και αποφυγή των πολλών διπλότυπων προτάσεων.

Πριν από τον εμπλουτισμό της εφαρμογής θα πρέπει να αξιολογηθούν οι υπάρχουσες υπηρεσίες από τους χρήστες. Οι απόψεις των χρηστών θα βοηθήσουν στην περαιτέρω ανάπτυξη της εφαρμογής με εμπλουτισμό νέων υπηρεσιών που δίνουν λύση σε ανάγκες που δεν έγιναν αντιληπτές.

5 Βιβλιογραφία

[1] Mobile Computing: Principles, Devices and Operating Systems Masoud Nosrati * - Ronak Karimi - Hojat Allah Hasanvand (2012)

[2] Mobile Device Management: Mobility Evaluation in Small and Medium-Sized Enterprises 1st ed. 2016 Edition by Markus Pierer (Author)

[3] <https://whitehatsdesign.com/mobile-ux/>

[4] Andreas Biørn-Hansen, Tor-Morten Grønli, and Gheorghita Ghinea. 2018. A Survey and Taxonomy of Core Concepts and Research Challenges in Cross-Platform Mobile Development. ACM Comput. Surv. 51, 5, Article 108 (November 2018), 29 pages. <https://doi.org/10.1145/3241739>

[5] Making the move to mobile. How to determine the best technology stack and architecture for modern app development Christopher Wilkinson, Ed Price 02.03.21 (<https://www.devbridge.com/white-papers/mobile-application-development/progressive-web-applications/#body>)

[6] <https://ionic.io/resources/articles/what-is-hybrid-app-development#h-what-is-a-hybrid-mobile-app>

[7] An empirical investigation of performance overhead in cross-platform mobile development frameworks Andreas Biørn-Hansen^{1,2} · Christoph Rieger³ · Tor-Morten Grønli¹ · Tim A. Majchrzak⁴ · Gheorghita Ghinea^{1,2} © The Author(s) 2020

[8] An Empirical Study of Cross-Platform Mobile Development in Industry Andreas Biørn-Hansen, ^{1,2} Tor-Morten Grønli, ¹ Gheorghita Ghinea, ^{1,2} and Sahel Alouneh ³

[9] Understanding native app development - what you need to know in 2019 By Anna Monus | Posted Mar 19, 2019

[10] Choosing a Mobile Application Development Approach Phyomintun Stamford International University, Thailand phyomintun@stamford.edu

[11] <https://www.techopedia.com/definition/2953/mobile-application-mobile-app>

[12] <https://codium.one/index.php/en/blog/25-mobile-app-industry-statistics-and-facts-to-know-in-2020>

[13] A. Ahmad, K. Li, C. Feng, S. M. Asim, A. Yousif and S. Ge, "An Empirical Study of Investigating Mobile Applications Development Challenges," in *IEEE Access*, vol. 6, pp. 17711-17728, 2018, doi: 10.1109/ACCESS.2018.2818724.

[14] <https://www.statica.com>

[15] <https://gs.statcounter.com>

[16] Best 10 Cross-Platform App Frameworks to Consider in 2021 Krunal Shah
MAR 08, 2021

[17] <https://www.GitHub.com>

[18] <https://www.androidauthority.com>

[19] <https://developer.android.com>

[20] <https://www.ibm.com/cloud/learn/api>

[21] <https://developers.google.com>

[22] <https://console.cloud.google.com>

[23] <https://www.javatpoint.com>

[24] <https://www.tutorialspoint.com>

[25] <https://www.json.org/json-en.html>