



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Ανάπτυξη Android εφαρμογής για τουριστικό οδηγό  
και προμήθειας προϊόντων και υπηρεσιών μέσω  
ηλεκτρονικού καταστήματος»



Του φοιτητή  
Μαχαλιώτη Χριστοφόρου  
Αρ. Μητρώου: 185224

Επιβλέπων  
Τσιακμάκης Κυριάκος  
Προσωπικό Ε.ΔΙ.Π.

ΙΟΥΝΙΟΣ 2024

Ανάπτυξη Android εφαρμογής για τουριστικό οδηγό και προμήθειας προϊόντων  
και υπηρεσιών μέσω ηλεκτρονικού καταστήματος  
22332

Φοιτητής: Μαχαλιώτης Χριστοφόρος

Εισηγητής: Τσιακμάκης Κυριάκος

Ημερομηνία ανάληψης Δ.Ε. 31-10-2022

Ημερομηνία περάτωσης Δ.Ε. 27-10-2023

*Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.*

*Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Μαχαλιώτη Χριστοφόρου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιοδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.*

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

## Πρόλογος

Στην εποχή της τεχνολογίας που διανύουμε, όλες οι επιχειρήσεις προσπαθούν να αξιοποιήσουν κάθε μορφή τεχνολογίας για να μεγιστοποιήσουν τα κέρδη τους. Ένας αποτελεσματικός τρόπος για να επιτευχθεί αυτό είναι η δημιουργία διαφόρων εφαρμογών για κινητά τηλέφωνα και η ενθάρρυνση της χρήσης αυτών των εφαρμογών από τους πελάτες τους. Ο στόχος αυτής της εργασίας είναι η ανάπτυξη μιας Android εφαρμογής που θα υποστηρίξει τις δραστηριότητες της Riverview Project Κοιν.Σ.Επ. και θα προωθήσει την προστασία και ανάδειξη της φυσικής και πολιτιστικής κληρονομιάς της περιοχής της Αιτωλοακαρνανίας. Μέσω αυτής της εφαρμογής, επιδιώκεται η ενίσχυση της επικοινωνίας μεταξύ της επιχείρησης και της τοπικής κοινότητας, η αύξηση της ευαισθητοποίησης σχετικά με τα έργα και τις πρωτοβουλίες της Riverview Project Κοιν.Σ.Επ., καθώς και η προώθηση της βιώσιμης ανάπτυξης στην περιοχή.

## Περίληψη

Η παρούσα εργασία αναλύει εκτενώς την ανάπτυξη και τις λειτουργίες της εφαρμογής "River View Project". Στο πρώτο κεφάλαιο της εργασίας παρουσιάζεται η σημασία του έργου στο πλαίσιο της κοινωνικής οικονομίας, ενώ εξετάζονται τα κίνητρα και οι λόγοι που ώθησαν στην ανάπτυξη της εφαρμογής. Έπειτα, παρουσιάζεται η υλοποίηση του "River View Project", δηλαδή περιγράφονται λεπτομερώς οι αιτίες, που οδήγησαν στην ανάπτυξη της εφαρμογής και η συνεισφορά της στον τοπικό τουρισμό και την οικονομία. Στο δεύτερο κεφάλαιο της εργασίας εξετάζονται οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής. Συγκεκριμένα, παρουσιάζονται η χρήση του Android Studio, του Kotlin, του Git, του Firebase καθώς οι αρχιτεκτονικές προσεγγίσεις που εφαρμόστηκαν, όπως η MVVM και η Clean Architecture. Στο τρίτο μέρος της εργασίας, γίνεται λεπτομερής ανάλυση της λειτουργικότητας της εφαρμογής και περιλαμβάνονται θέματα όπως ο χειρισμός των προτιμήσεων του χρήστη, η αποθήκευση δεδομένων, η διαχείριση εξαρτήσεων και η ενσωμάτωση υπηρεσιών Firebase, όπως η αναλυτική αναφορά, η ανίχνευση σφαλμάτων και οι ειδοποιήσεις Push. Επιπροσθέτως, γίνεται μια λεπτομερής ανάλυση του κώδικα της εφαρμογής, και περιγράφονται όλες οι συνιστώσες της εφαρμογής, με τη χρήση στιγμιότυπων (screenshots), όπως το Gradle, το Android Manifest και τα στοιχεία της Jetpack Compose που χρησιμοποιήθηκαν για τη δημιουργία του χρήσιμου περιεχομένου της εφαρμογής. Στο τέταρτο και τελευταίο μέρος της εργασίας, γίνεται η τελική παρουσίαση της εφαρμογής και επεξηγείται βήμα προς βήμα η λειτουργικότητά της.

## Abstract

This paper extensively analyzes the development and operations of the "River View Project" application. In the first chapter of the work, the importance of the project in the context of the social economy is presented, while the motivations and reasons that prompted the development of the application are examined. Then, the implementation of the "River View Project" is presented, i.e. the reasons that led to the development of the application and its contribution to local tourism and the economy are described in detail. In the second chapter of the work, the technologies used for the implementation of the application are examined. In particular, the use of Android Studio, Kotlin, Git, Firebase as well as the architectural approaches applied, such as MVVM and Clean Architecture, are presented. In the third part of the paper, the application's functionality is analyzed in detail and includes topics such as handling user preferences, data storage, dependency management, and integrating Firebase services such as reporting, error detection, and push notifications. In addition, a detailed analysis of the application code is performed, and all components of the application are described, using screenshots, such as Gradle, the Android Manifest, and the Jetpack Compose components used to create the useful content of the application . In the fourth and last part of the work, the final presentation of the application is made and its functionality is explained step by step.

## Περιεχόμενα

Πρόλογος.....	iii
Περίληψη.....	iv
Abstract.....	v
Περιεχόμενα.....	vi
Κατάλογος Εικόνων.....	ix
Συντομογραφίες.....	x
Κεφάλαιο 1ο: Εισαγωγή .....	1
1.1 Εισαγωγή.....	1
1.2 Κοινωνική Οικονομία .....	1
1.3 River View Project .....	1
1.4 Κίνητρο .....	2
1.5 Συνεισφορά .....	3
Κεφάλαιο 2ο: Τεχνολογίες για το Android Εφαρμογή .....	4
2.1 Εισαγωγή.....	4
2.2 Android .....	4
2.2.1 Android Studio .....	4
2.2.2 Android SDK.....	4
2.2.3 Android Gradle .....	5
2.3 Kotlin .....	5
2.4 Git .....	5
2.4.1 GitLab .....	7
2.5 Android Coroutines .....	8
2.6 Αρχιτεκτονικές (Architecture Patterns) .....	9
2.7 MVVM.....	9
2.8 Clean Architecture.....	11
2.8.1 Frameworks και Drivers .....	11
2.8.2 Interface Adapters .....	12
2.8.3 Use Cases .....	13
2.8.4 Entities.....	13

2.9 Clean Architecture με MVVM.....	13
2.10 Shared Preferences.....	13
2.11 Datastore .....	14
2.12 Hilt .....	14
2.13 Firebase .....	15
2.13.1 Analytics.....	15
2.13.2 Crashlytics .....	16
2.13.3 Push Notifications .....	16
Κεφάλαιο 3ο: Ανάλυση Κώδικα της Εφαρμογής.....	17
3.1 Gradle.....	17
3.2 Android Manifest .....	19
3.3 Activity.....	20
3.4 Fragment .....	21
3.5 Navigation Components.....	22
3.6 ComposeView .....	23
3.7 Compose .....	24
3.7.1 Διαφορές μεταξύ Jetpack Compose και XML UI toolkit στο android .....	24
3.7.2 Στήλη (column) .....	25
3.7.3 Σειρά (row).....	26
3.7.4 LazyColumn .....	26
3.7.5 LazyRow .....	27
3.7.6 Διαφορές μεταξύ Column και Row με LazyColumn και LazyRow.....	28
3.7.7 Compose Box .....	28
3.7.8 Launched Effect .....	29
3.7.9 Grids στο Jetpack Compose .....	30
3.7.10 Pager στο Jetpack Compose .....	31
3.7.11 Modifiers στο Jetpack Compose .....	32
3.7.12 States στο Jetpack Compose .....	33
3.8 Βιβλιοθήκες (Libraries) .....	34
3.8.1 Material Design .....	35

3.8.2 Material Ui .....	36
3.8.3 Timber .....	37
3.8.4 AndroidX Navigation .....	37
3.8.5 Google Maps Android API .....	38
3.8.6 Accompanist .....	39
3.8.7 Accompanist Map Compose.....	40
3.8.10 Compose UI.....	41
Κεφάλαιο 4ο: Παρουσίαση Εφαρμογής .....	45
4.1 Android εφαρμογή .....	45
4.2 Desktop App.....	56
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	59
Παραρτήματα.....	61

## Κατάλογος Εικόνων

Εικόνα 2.1: Συγχώνευση πολλών branches σε ένα .....	6
Εικόνα 2.2: Αποθήκευση αλλαγής δεδομένων σε μια βασική έκδοση κάθε αρχείου .....	6
Εικόνα 2.3: Αποθήκευση ως στιγμιότυπα .....	17
Εικόνα 2.4 : Σχήμα MVVM .....	10
Εικόνα 2.5 : Σχεδιάγραμμα Clean Architecture .....	12
Εικόνα 3.1: Ο κύκλος ζωής ενός activity .....	22
Εικόνα 3.2: Το nav_graph της εφαρμογής.....	23
Εικόνα 4.1: Οθόνη Εισόδου .....	46
Εικόνα 4.2: Οθόνη Εγγραφής χρήστη .....	48
Εικόνα 4.3:Κεντρική οθόνη .....	49
Εικόνα 4.4:Κεντρική οθόνη .....	49
Εικόνα 4.5: Οθόνη Ανακοινώσεων.....	51
Εικόνα 4.6: Οθόνη Δράσεων .....	51
Εικόνα 4.7: Οθόνη τουριστικού οδηγού.....	52
Εικόνα 4.8: Οθόνη τοπίου .....	53
Εικόνα 4.9: Οθόνη market.....	54
Εικόνα 4.10: Εμφάνιση καλαθιού με αντικείμενα .....	55
Εικόνα 4.11 : Εμφάνιση άδειου καλαθιού.....	55
Εικόνα 4.12: Οθόνη επικοινωνίας .....	56
Εικόνα 4.13: Οθόνη Desktop app για ανέβασμα event .....	58
Εικόνα 4.14: Οθόνη Desktop app για ανέβασμα ανακοίνωσης .....	58
Εικόνα 4.16: Οθόνη Desktop app για ανέβασμα market item .....	59
Εικόνα 4.17: Οθόνη Desktop app για ανέβασμα δράσης.....	59

Συντομογραφίες

## **Συντομογραφίες**

ΔΠΑΕ Διεθνές Πανεπιστήμιο Ελλάδος

Π.Ε. Πτυχιακή Εργασία

## **Κεφάλαιο 1ο: Εισαγωγή**

### **1.1 Εισαγωγή**

Πρώτα-πρώτα είναι σημαντικό να γίνει μια σύντομη εισαγωγή για την ιστορία της εφαρμογής, ώστε να γίνει αντιληπτός ο λόγος και τα κίνητρα, που οδήγησαν στον σχεδιασμό της, και εν συνεχεία στην υλοποίησή της κάνοντας την ένα ένα αυτόνομο και ολοκληρωμένο εργαλείο όχι μόνο για τους τουρίστες αλλά και για τις τοπικές επιχειρήσεις.

Ωστόσο, για να γίνει μια εμβάθυνση, δηλαδή για να κατευθυνθεί η εργασία από το γενικό σε κάτι ειδικό είναι σημαντικό να οριστεί αρχικά η κοινωνική οικονομία, που συσχετίζεται άμεσα με την εφαρμογή River View Project.

### **1.2 Κοινωνική Οικονομία**

“Η Κοινωνική Οικονομία” αφορά τις οικονομικές δραστηριότητες που αναλαμβάνονται από επιχειρήσεις και οργανισμούς, κυρίως από συνεταιρισμούς, κοινωνίες αλληλοβοήθειας και σωματεία (ενώσεις), των οποίων η ίδρυση και λειτουργία στοχεύουν: α) στην παροχή υπηρεσιών προς τα μέλη τους ή την κοινωνία, παρά την επιδίωξη κέρδους, β) στην ανεξάρτητη διοίκηση, γ) στην εφαρμογή δημοκρατικής διαδικασίας λήψης αποφάσεων, δ) και στην απόδοση προτεραιότητας για άτομα και για εργασία έναντι του κεφαλαίου κατά τη διανομή του εισοδήματος” (Defourny, 2001). Σύμφωνα με το άρθρο 2 παράγραφος 1 του ισχύοντος νόμου (Ν. 4430/2016) για την Κοινωνική και Αλληλέγγυα Οικονομία και ανάπτυξη των φορέων της, ως «Κοινωνική και Αλληλέγγυα Οικονομία» ορίζεται το σύνολο των οικονομικών δραστηριοτήτων που στηρίζονται σε μία εναλλακτική μορφή οργάνωσης των σχέσεων παραγωγής, διανομής, κατανάλωσης και επανεπένδυσης, βασισμένη στις αρχές της δημοκρατίας, της ισότητας, της αλληλεγγύης, της συνεργασίας, καθώς και του σεβασμού στον άνθρωπο και το περιβάλλον.

### **1.3 River View Project**

Η Riverview Project Κοιν.Σ.Επ. είναι μια πρόσφατα ιδρυόμενη κοινωνική συνεταιριστική επιχείρηση με έδρα το Δήμο Αργινίου. Στοχεύει στην προστασία και ανάδειξη της φυσικής και πολιτιστικής κληρονομιάς της ευρύτερης περιοχής του Αργινίου και της Αιτωλοακαρνανίας, εστιάζοντας στον ποταμό Αχελώο, όπως και στην ενίσχυση της κοινωνικής συνοχής, της επιχειρηματικότητας και της καινοτομίας, προωθώντας την τοπική ανάπτυξη. Μέσω αυτών των στόχων, η επιχείρηση διαμορφώνει προοπτικές ανάπτυξης εναλλακτικών μορφών τουρισμού, και ενισχύει την τοπική οικονομία.

Οι πρωτοβουλίες της επικεντρώνονται κυρίως σε μουσικές εκδηλώσεις, εκθέσεις τέχνης, προβολή τοπικών προϊόντων και παραγωγών, αθλητικές δραστηριότητες, επιμορφωτικές

ημερίδες, εθελοντικές δράσεις και θεματικά workshops. Στα σχέδιά της περιλαμβάνονται επίσης, η δημιουργία νέων θέσεων εργασίας μέσω της ανάπτυξης εγκαταστάσεων για φαγητό, διαμονή και αναψυχή, οι οποίες θα χρηματοδοτηθούν μέσω δωρεών, επιχορηγήσεων και εσόδων από τις δράσεις της επιχείρησης.

#### 1.4 Κίνητρο

Η δημιουργία μιας εφαρμογής για την Riverview Project προσφέρει μια μοναδική ευκαιρία για προσωπική και επαγγελματική ανάπτυξη, καθώς και την ευκαιρία να κάνει κάποιος τη διαφορά στη ζωή των ανθρώπων της τοπικής κοινότητας. Γίνεται λοιπόν αντιληπτό, ότι πίσω από μια τέτοια δημιουργία, κρύβονται ποικίλοι λόγοι, κίνητρα αλλά και αξίες που οδηγούν ένα άτομο ή μια ομάδα να προσφέρει τις γνώσεις και τις δεξιότητες τους για την υλοποίησή της. Ειδικότερα:

**Ενδιαφέρον για το περιβάλλον και την πολιτιστική κληρονομιά:** Πολλοί άνθρωποι εκδηλώνουν ενδιαφέρον για την προστασία και ανάδειξη της φυσικής και πολιτιστικής κληρονομιάς. Συνεργαζόμενοι με την Riverview Project Κοιν.Σ.Επ., μπορούν να συμβάλουν στη διατήρηση της πολιτιστικής κληρονομιάς και του φυσικού πλούτου της περιοχής, γεγονός που αποτελεί προσωπική επιδίωξη για όλους.

**Κοινωνική συνεισφορά:** Η συμβολή στη βελτίωση των συνθηκών ζωής στην τοπική κοινωνία μέσω της δημιουργίας εφαρμογής η οποία υποστηρίζει την κοινωνική συνοχή, την επιχειρηματικότητα και την καινοτομία αποτελεί μια ευκαιρία για να προσφέρει κάποιος θετικό αντίκτυπο στη βελτίωση της ποιότητας ζωής των κατοίκων.

**Προσωπική ικανοποίηση:** Η ευκαιρία να συνεργαστεί κάποιος με μια κοινωνική συνεταιριστική επιχείρηση και να δει τα αποτελέσματα της εργασίας του να έχουν άμεση και θετική επίδραση στην κοινότητα προσφέρει μια αίσθηση προσωπικής ικανοποίησης και ολοκλήρωσης.

**Δίκτυο και ευκαιρίες:** Η εμπλοκή με την Riverview Project Κοιν.Σ.Επ. μπορεί να προσφέρει την ευκαιρία σε κάποιον να διευρύνει το δίκτυό του, να έρθει σε επαφή με ανθρώπους με παρόμοια ενδιαφέροντα και καλλιεργηθεί πνευματικά μέσω της ανταλλαγής απόψεων.

**Επαγγελματική ανάπτυξη:** Η δημιουργία μιας εφαρμογής για την Riverview Project Κοιν.Σ.Επ. μπορεί να προσφέρει μία πολύτιμη και μοναδική εμπειρία καθώς επίσης και την ανάπτυξη δεξιοτήτων σε όποιον αναλάβει το έργο, βελτιώνοντας το βιογραφικό του και ανοίγοντας νέες επαγγελματικές προοπτικές.<sup>2</sup>

**Αίσθημα υποχρέωσης:** Άτομα που συνδέονται με την περιοχή του Αγρινίου και της Αιτωλοακαρνανίας μπορεί να αισθάνονται υποχρέωση να βοηθήσουν στην ανάπτυξη της περιοχής τους και να συνεισφέρουν στην ευημερία της κοινότητάς τους.

**Αγάπη για τη τεχνολογία:** Άτομα με πάθος για την τεχνολογία και τη δημιουργία εφαρμογών μπορεί να βρίσκουν κίνητρο στην ίδια τη διαδικασία δημιουργίας, ειδικά όταν γνωρίζουν ότι το έργο τους θα συμβάλει σε κάτι μεγαλύτερο και θετικότερο.

## 1.5 Συνεισφορά

Η δημιουργία μιας εφαρμογής για τη Riverview Project Κοιν.Σ.Επ. αποτελεί μια ευκαιρία να επιτευχθούν οι στόχοι της επιχείρησης με πιο αποτελεσματικό και οργανωμένο τρόπο. Καθώς η επιχείρηση έχει πρόσφατα ιδρυθεί και δραστηριοποιείται στον τομέα της προστασίας και ανάδειξης της φυσικής και πολιτιστικής κληρονομιάς, η εφαρμογή αυτή μπορεί να προσφέρει πολυάριθμα οφέλη. Συγκεκριμένα, η εφαρμογή μπορεί να λειτουργήσει ως κεντρικό σημείο πληροφόρησης για τις πρωτοβουλίες της Riverview Project Κοιν.Σ.Επ. Οι ενδιαφερόμενοι πολίτες και επισκέπτες μπορούν να ενημερώνονται για τις μουσικές εκδηλώσεις, εκθέσεις τέχνης, προβολή τοπικών προϊόντων και παραγωγών, αθλητικές δραστηριότητες, επιμορφωτικές ημερίδες, εθελοντικές δράσεις και θεματικά workshops που διοργανώνει η επιχείρηση. Επίσης, η εφαρμογή έχει τη δυνατότητα να προωθήσει την οικονομική τοπική ανάπτυξη, προβάλλοντας τα προϊόντα και τις υπηρεσίες που προσφέρονται από την περιοχή. Κατ' επέκταση με την παρουσίαση αυτών των τοπικών προϊόντων μέσω της εφαρμογής, η Riverview Project Κοιν.Σ.Επ. μπορεί να προσελκύσει περισσότερους τουρίστες και επενδυτές, δημιουργώντας νέες ευκαιρίες απασχόλησης. Γίνεται λοιπόν αντιληπτό, ότι μια τέτοια εφαρμογή αποτελεί ένα εργαλείο για την προώθηση της κοινωνικής συνοχής και της επιχειρηματικότητας, αφού μέσω της εφαρμογής, οι χρήστες μπορούν να ενημερώνονται για τις δράσεις της επιχείρησης, να λαμβάνουν μέρος σε εθελοντικές δραστηριότητες και να συμμετέχουν σε workshops που προωθούν την καινοτομία. Τέλος, η εφαρμογή μπορεί να προσφέρει στους επισκέπτες και τους πολίτες τη δυνατότητα να συμβάλλουν οικονομικά στην επιχείρηση μέσω δωρεών και χορηγιών. Η δημιουργία μιας εφαρμογής που παρέχει ευκολία στη διαδικασία δωρεών μπορεί να αυξήσει τα έσοδα της επιχείρησης και να χρηματοδοτήσει την ανάπτυξη εγκαταστάσεων για φαγητό, διαμονή και αναψυχή. Συνολικά, η δημιουργία μιας εφαρμογής για τη Riverview Project Κοιν.Σ.Επ. αποτελεί ευκαιρία να υλοποιηθούν οι στόχοι της επιχείρησης, προωθώντας την τοπική ανάπτυξη, την κοινωνική συνοχή και την καινοτομία.

## Κεφάλαιο 2ο: Τεχνολογίες για το Android Εφαρμογή

### 2.1 Εισαγωγή

Έχοντας κάνει λοιπόν μία γενική εισαγωγή αναφορικά με την ιστορία της εφαρμογής και τα οφέλη της δημιουργίας της, ανοίγει ο δρόμος για την εμβάθυνση στις τεχνολογίες που είναι απαραίτητες για την δημιουργία μιας Android εφαρμογής σαν το RiverView project. Σε αυτό το κεφάλαιο λοιπόν, θα εξεταστούν λεπτομερώς όχι μόνο οι τεχνολογίες αλλά και τα εργαλεία, οι βιβλιοθήκες και οι αρχιτεκτονικές που εφαρμόστηκαν και συνέβαλαν για την υλοποίηση της Android εφαρμογής.

### 2.2 Android

Το Android είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα, εμπνευσμένο από το λειτουργικό σύστημα Linux και είναι γνωστό κυρίως από την εφαρμογή του σε διάφορες κινητές συσκευές. Αναπτύχθηκε από την Android Inc. και συστήθηκε στο κοινό το 2007, ενώ στην συνέχεια αγοράστηκε από την Google. Οι χρήσεις του όμως, δεν σταματούν εκεί, καθώς είναι προσβάσιμο σε τάμπλετ, τηλεοράσεις, ρολόι και σε αυτοκίνητα.

#### 2.2.1 Android Studio

Το Android स्टούντιο είναι ένα ολοκληρωμένο προγραμματιστικό περιβάλλον ( IDE ) για ανάπτυξη εφαρμογών Android. Ανακοινώθηκε 16 Μαΐου 2013 στο συνέδριο Google I/O, με την πρώτη σταθερή έκδοση να είναι διαθέσιμη τον Δεκέμβριο του 2014[1]. Είναι βασισμένο στο λογισμικό του η JetBrains' IntelliJ IDEA και είναι διαθέσιμο για Linux, MacOS, Windows και ChromeOS λειτουργικά. Ακόμη, περιέχει το ειδικό εργαλείο για έλεγχο εκδόσεων εντός του κιτ, γεγονός που βοηθάει στην διαχείριση των εκδόσεων.

#### 2.2.2 Android SDK

Το Android SDK αποτελεί ένα σύνολο εργαλείων, το οποίο περιέχει πολλά πακέτα που απαιτούνται για την ανάπτυξη εφαρμογών Android και παρέχεται από την Google. Περιλαμβάνει διάφορα εργαλεία και βιβλιοθήκες τα οποία είναι απαραίτητα για την δημιουργία, δόκιμή και την εκτέλεση των εφαρμογών[2], καθώς επίσης διαθέτει και διάφορα API δ τα οποία δίνουν την δυνατότητα στους προγραμματιστές να αξιοποιούν διάφορες λειτουργίες του συστήματος, όπως πρόσβαση σε διάφορες συσκευές πήχη φακό ακουστικό κάμερα αισθητήρα και τα λοιπά αλλά και την διαχείριση του δικτύου. Την τελευταία έκδοση που κυκλοφορεί είναι το as DK 34 το οποίο αντιστοιχεί στο Android 15. Τέλος, πρέπει να σημειωθεί ότι είναι συμβατό με Linux, MacOS, Windows και ChromeOS.

### 2.2.3 Android Gradle

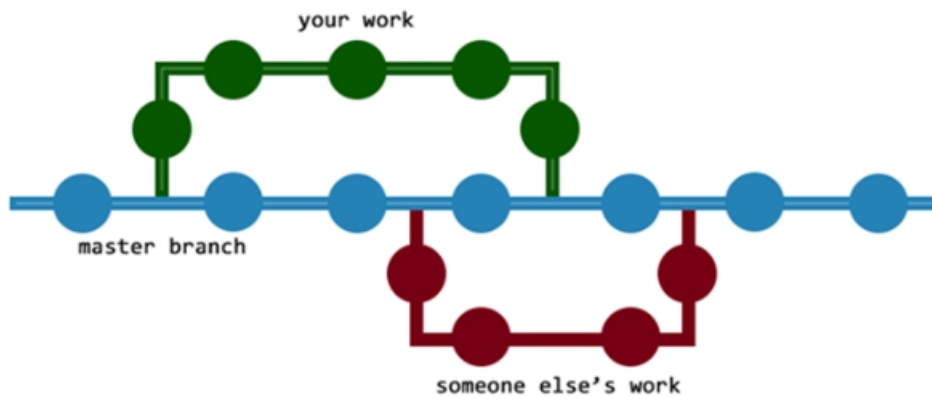
Το Android Gradle είναι ένα σημαντικό εργαλείο ανάπτυξης εφαρμογών Android [4]. Στο Android στούντιο το gradle είναι προεγκατεστημένο και χρησιμοποιείται στην δημιουργία χρησιμοποιείται στην δημιουργία μπιλντ των project, ως ένα σύστημα δημιουργίας (build) των project, ως ένα σύστημα δημιουργίας (build system). Παρέχει εργαλεία για την οργάνωση του κώδικα, τη διαχείριση εξαρτήσεων, διαχείριση βιβλιοθηκών κ.α και τα συγκεντρώνει σε ένα APK. Είναι υπεύθυνο για την δοκιμή την ανάπτυξη και την μετατροπή του κώδικα σε αρχεία .dex και τελικώς στην εκτέλεση της εφαρμογής στην συσκευή.

### 2.3 Kotlin

Η Kotlin είναι μια αντικειμενοστρεφής γλώσσα προγραμματισμού που αναπτύχθηκε από την JetBrains και κυκλοφόρησε το 2011. Το 2012 αναπτύσσεται ως γλώσσα ανοικτού κώδικα, με την πρώτη επίσημη έκδοση 1.0 να δημοσιεύεται τον Φεβρουάριο του 2016. Στη συνέχεια τον Μάιο του 2017 γίνεται η επίσημη προτεινόμενη γλώσσα για την ανάπτυξη Android εφαρμογών. Η Κότλιν τρέχει σε εικονική μηχανή java (JVM) και είναι απολύτως συμβατή με την Java. Εκτός από το JVM στο οποίο βρίσκει περισσότερη χρήση, αξίζει να σημειώσουμε, ότι μεταγλωττίζεται και σε Javascript ,για web frontend εφαρμογες, αλλά και σε LLVM, για native iOS εφαρμογες. Η Kotlin επηρεάστηκε από διάφορες γλώσσες προγραμματισμού και προσέλαβε στοιχεία από αυτές για την ανάπτυξή της, οι οποίες είναι η Java, η Scala, η C# και η Groovy. Τέλος, μέσω του Android Studio υπάρχει η δυνατότητα μετατροπής κώδικα kotlin σε κώδικα java και το αντίστροφο, καθώς επίσης μπορεί να υποστηρίξει κώδικα και απο τις δύο γλώσσες ταυτόχρονα.

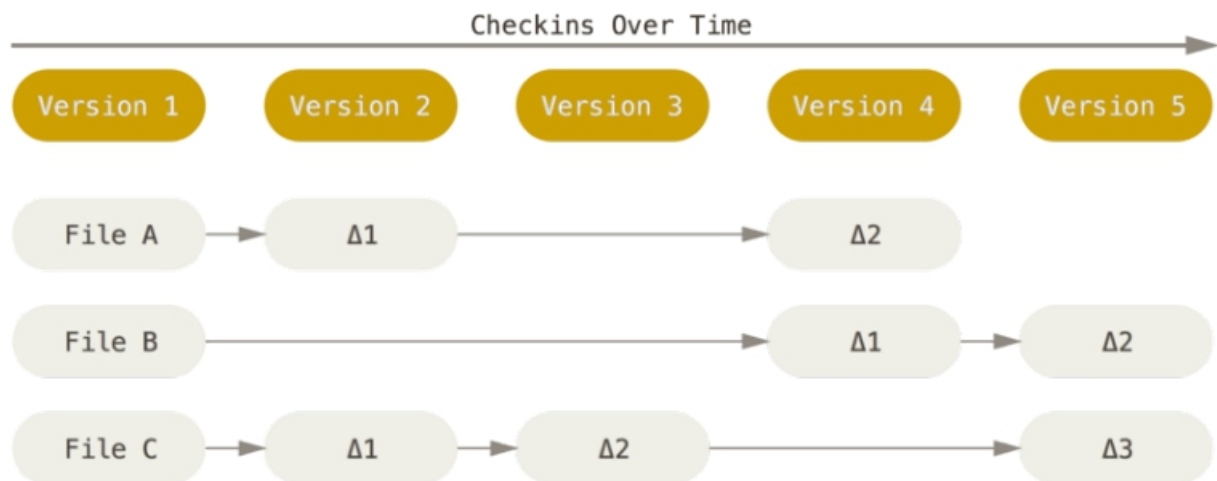
### 2.4 Git

Το Git είναι ένα από τα πιο δημοφιλή συστήματα διαχείρισης εκδόσεων (Version Control System - VCS) και έχει γίνει αναπόσπαστο εργαλείο για προγραμματιστές και ομάδες ανάπτυξης[4]. Ο στόχος του Git είναι να διευκολύνει τη συνεργασία μεταξύ προγραμματιστών, τη διαχείριση αλλαγών στον κώδικα και την παρακολούθηση της εξέλιξης του έργου. Με το Git, οι προγραμματιστές μπορούν να δημιουργούν παρακλάδια (branches) του κώδικα για την ανάπτυξη νέων χαρακτηριστικών, να παρακολουθούν τις αλλαγές που έχουν γίνει και να τις συγχωνεύει από διάφορες πηγές σε έναν κλάδο (branch). Σημαντικό κρίνεται επίσης να αναφερθεί, ότι δίνει τη δυνατότητα για επιστροφή σε συγκεκριμένη παλιότερη έκδοση αν αυτό χρειαστεί.



Εικόνα 2.1: Συγχώνευση πολλών branches σε ένα

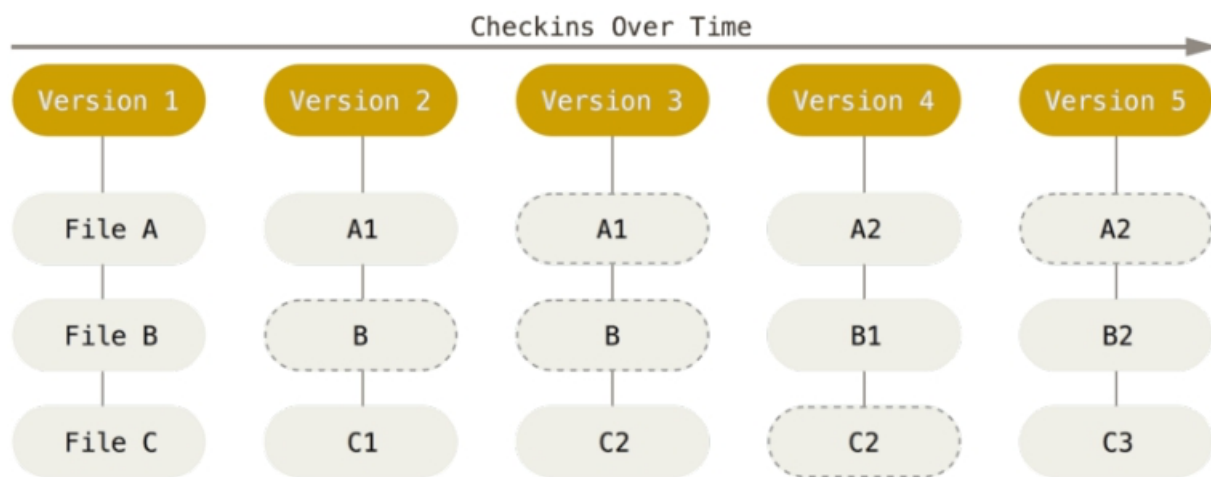
Η κύρια διαφορά μεταξύ του Git και οποιουδήποτε άλλου VCS, εντοπίζεται στον τρόπο με τον οποίο το Git σκέφτεται για τα δεδομένα του. Γενικά, τα υπόλοιπα συστήματα αποθηκεύουν πληροφορίες ως μια λίστα αλλαγών που βασίζονται σε αρχεία. Αυτά τα συστήματα (CVS, Subversion, Perforce κ.τ.λ) θεωρούν τις πληροφορίες που αποθηκεύουν ως ένα σύνολο αρχείων και τις αλλαγές που γίνονται σε κάθε αρχείο με την πάροδο του χρόνου. Αυτό ονομάζεται συνήθως δέλτα έλεγχος έκδοσης (delta-based version control).



Εικόνα 2.2: Αποθήκευση αλλαγής δεδομένων σε μια βασική έκδοση κάθε αρχείου

Η λογική αποθήκευσης δεδομένων του Git αποκλίνει από τα παραπάνω συστήματα διαχείρισης εκδόσεων. Αντί να παρακολουθεί μεμονωμένες αλλαγές σε αρχεία, το Git

αντιμετωπίζει τα δεδομένα του ως μια σειρά στιγμιότυπων ενός συστήματος αρχείων. Κάθε φορά που γίνεται ένα commit στο Git, ουσιαστικά δημιουργείτε ένα στιγμιότυπο της τρέχουσας κατάστασης του έργου σας. Το Git καταγράφει τις αλλαγές που έχουν γίνει στα αρχεία από τον τελευταίο στιγμιότυπο και τις αποθηκεύει ως ένα νέο commit. Για να εξοικονομήσει χώρο αποθήκευσης, εάν ένα αρχείο δεν έχει αλλάξει, το Git δεν το αποθηκεύει ξανά, αλλά δημιουργεί έναν σύνδεσμο προς το προηγούμενο αντίστοιχο αρχείο. Με αυτόν τον τρόπο, το Git αντιμετωπίζει τα δεδομένα του ως μια ροή στιγμιότυπων, διατηρώντας μια συνεχή ιστορική πορεία του έργου.



Εικόνα 2.3: Αποθήκευση ως στιγμιότυπα

### 2.4.1 GitLab

Το Git και το GitLab, αν και συνδέονται στενά μεταξύ τους, είναι δύο διαφορετικά πράγματα. Το GitLab είναι μια υπηρεσία φιλοξενίας (hosting service)[5], που βασίζεται στο cloud και παρέχει μια εξαιρετική πλατφόρμα για τη διαχείριση του Git. Παρέχει τις βασικές του υπηρεσίες δωρεάν, ενώ προηγμένες λειτουργίες επαγγελματικού και επιχειρηματικού επιπέδου είναι διαθέσιμες με εμπορική συνδρομή. Οι δωρεάν λογαριασμοί στο GitLab είναι δημοφιλείς για τη φιλοξενία project ανοιχτού κώδικα, παρέχοντας ένα κέντρο κοινότητας για προγραμματιστές και χρήστες που επιθυμούν να συνεισφέρουν, να συνεργαστούν και να αξιολογήσουν τον κώδικα. Με τις λειτουργίες ανίχνευσης σφαλμάτων, διαχείρισης προτεραιοτήτων και αυτοματοποιημένων διαδικασιών ελέγχου ποιότητας, το GitLab προσφέρει ένα πλήρες εργαλείο διαχείρισης κώδικα για κάθε μέγεθος ομάδας ανάπτυξης.

## 2.5 Android Coroutines

Οι Android Coroutines αποτελούν ένα σημαντικό design pattern στην Kotlin και, συνεπώς, στο Android, που επιτρέπει την εκτέλεση κώδικα ασύγχρονα[6]. Εισήχθησαν στην Kotlin από την έκδοση 1.3 και βασίζονται σε καθιερωμένες έννοιες από άλλες γλώσσες προγραμματισμού.

Θέτοντας ένα εισαγωγικό πλαίσιο, πριν την ανάλυση, οι coroutines σαν ένας γενικός ορισμός αποτελεί ένας τρόπος γραφής ενός παλιού και ασύγχρονου κώδικα με έναν σύγχρονο και πιο απλό τρόπο. Ουσιαστικά, ένας κώδικας ο οποίος εκτελείται ασύγχρονα μπορεί να γραφτεί με έναν όμοιο τρόπο που θα γραφόταν ένας σύγχρονος κώδικας, χωρίς επιβραδύνεται η διεπαφή χρήστη ή να κολλάει σε ένθετες επανακλήσεις, πράγμα που διευκολύνει την διατήρηση του. Έτσι, η σύνταξη εφαρμογών, που εκτελούν ασύγχρονες διεργασίες, γίνεται πιο εύκολη και σαφής.

Οι coroutines στα Android συμβάλλουν στην διαχείριση εργασιών οι οποίες θα μπορούσαν να αποκλείσουν το κύριο νήμα κάνοντας την εφαρμογή μη ανταποκρίσιμη. Σαν κύρια λειτουργία η χρήση coroutine στο Android είναι να κάνουν εύκολη την ανάγνωση ενός ασύγχρονου κώδικα κατά την σύνταξή του. Επίσης, δεν ασκούν μεγάλη επιρροή στην απόδοση μιας Android εφαρμογής συγκριτικά με άλλες λύσεις, επομένως αποτελούν μια ελκυστική επιλογή για την εκτέλεση ασύγχρονων εργασιών στο Android. Για να δηλωθεί μια coroutine στο Android, είναι σημαντικό πρώτα να συμπεριληφθεί η βιβλιοθήκη υποστήριξης AndroidX στην εφαρμογή και να προστεθεί η εξάρτηση στο build.gradle αρχείο.

Το επόμενο βήμα, είναι δημιουργηθεί μια συνάρτηση που να εκτελεί την ασύγχρονη εργασία επισημαίνοντας της παράλληλα με την λέξη-κλειδί “suspend”, ώστε η κλήση της συνάρτησης να είναι εφικτή όταν είναι απαραίτητο, από την coroutine και την αναστολή. Το πρόγραμμα εκκίνησης δημιουργίας της coroutine, είναι μια μέθοδος του CoroutineScope που ξεκινά μια coroutine η οποία θα εκτελέσει την λειτουργία αναστολής με ασύγχρονο τρόπο και δεν χρειάζεται να επιστρέψει μια τιμή στον καλούντα, όπως την ενημέρωση διεπαφής χρήστη.

Υπάρχουν πολλοί τύποι CoroutineScope οι οποίοι μπορούν να δημιουργήσουν coroutines στο Android . Οι πιο κοινοί είναι:

**GlobalScope:** Αποτελεί τύπος coroutine που λειτουργεί από ένα διαφορετικό στοιχείο Android όπως μια δραστηριότητα και χρησιμοποιείται για εργασίες που δεν αφορούν στοιχεία διεπαφής χρήστη και για coroutines που δεν εξαρτώνται από τη διάρκεια ζωής ενός στοιχείου Android.

**viewModelScope:** Αποτελεί επέκταση του ViewModel που δημιουργεί coroutine που διαγράφονται όταν το ViewModel καταστραφεί. Χρησιμοποιείται για διεργασίες που αφορούν την προβολή ανεξαρτήτως από τη διάρκεια ζωής μιας δραστηριότητας.

**lifecycleScope:** Είναι μια επέκταση του LifecycleOwner (όπως μια Activity ή Fragment) η οποία διαμορφώνει coroutines οι οποίες ακυρώνονται όταν το LifecycleOwner καταστραφεί. Σε αντίθεση με τα άλλα, η χρήση του αφορά διεργασίες που εξαρτώνται από τη διάρκεια ζωής ενός συγκεκριμένου στοιχείου Android.

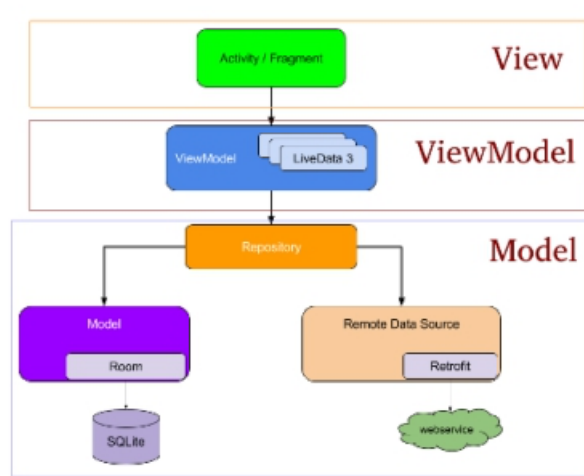
Τελικό βήμα για την δημιουργία μιας coroutine είναι η χρήση “try-catch” για την διαχείριση ζητημάτων που μπορεί να δημιουργηθούν κατά την εκτέλεση της λειτουργίας αναστολής (suspend)

## 2.6 Αρχιτεκτονικές (Architecture Patterns)

Η χρήση μιας κατάλληλης αρχιτεκτονικής σε κάθε έργο αποτελεί σημαντικό παράγοντα για την οργάνωση του κώδικα και την αποφυγή του "spaghetti code", δηλαδή του μπερδεμένου κώδικα[7]. Τα πλεονεκτήματα αυτών των αρχιτεκτονικών δεν περιορίζονται μόνο σε αυτό, αλλά επεκτείνονται και στην γενικότερη ευελιξία που προσφέρουν στο λογισμικό και τον κώδικα κατά την εφαρμογή τους. Η Clean Architecture, σε συνδυασμό με το MVVM (Model View ViewModel), αποτελεί ένα δημοφιλές παράδειγμα αρχιτεκτονικής που παρέχει οργάνωση και δομή στον κώδικα. Η Clean Architecture διαχωρίζει τον κώδικα σε επίπεδα με διαφορετικές ευθύνες και εξασφαλίζει την επαναχρησιμοποίηση, τη διαχείριση των εξαρτήσεων και τη δομή του κώδικα. Από την άλλη πλευρά, το MVVM παρέχει ένα πρότυπο για τη διαχείριση του UI (User Interface) και τη διασύνδεση με τα δεδομένα, επιτρέποντας την αποφυγή της σύγχυσης μεταξύ της λογικής εφαρμογής και της παρουσίασης των δεδομένων. Με τη συνδυασμένη χρήση αυτών των δύο αρχιτεκτονικών, επιτυγχάνεται η ανάπτυξη λογισμικού που είναι ευανάγνωστο, ευέλικτο και ευαίσθητο στις μελλοντικές αλλαγές και αναβαθμίσεις.

## 2.7 MVVM

Κατά την ανάπτυξη μιας εφαρμογής, ο προγραμματιστής καλείται να ξεπεράσει κάποια προβλήματα τα οποία μπορεί να εμφανιστούν στον κώδικα του[8]. Συνήθως, όταν μία εφαρμογή είναι πολύπλοκη και έχει γραφτεί αρκετός κώδικας, πρέπει αυτός να είναι συγκεκριμένος και τοποθετημένος έτσι ώστε να είναι εύκολος προς την συντήρηση του καθώς και ευανάγνωστος. Για να το πετύχουμε αυτό, αναγκαίο κρίνεται να χρησιμοποιείται κάποια αρχιτεκτονική. Μία αρχιτεκτονική η οποία έχει τοποθετηθεί στις πιο δημοφιλείς και αναγνωρίσιμες αρχιτεκτονικές, είναι η αρχιτεκτονική MVM. Το μοντέλο αυτό οργανώνει τον κώδικα, διαχωρίζοντας την εφαρμογή σε τρία κομμάτια. Αρχικά το model, στη συνέχεια το view και τέλος το ViewModel.



Εικόνα 2.4 : Σχήμα MVVM

Αναλυτικότερα, το κομμάτι του model αποτελεί ένα καίριο σημείο για την αρχιτεκτονική, καθώς είναι υπεύθυνο για την ενσωμάτωση των δεδομένων από διάφορες πηγές όπως βάσεις δεδομένων, αλλά και για την ενσωμάτωση της επιχειρησιακής λογικής business logic την οποία θα ακολουθήσει η εφαρμογή. Με το να κρατηθεί το model ξεχωριστά από τα άλλα στάδια της εφαρμογής, επιτυγχάνεται η ανεξαρτησία των δεδομένων από την διεπαφή του χρήστη. Δηλαδή, οποιαδήποτε αλλαγή γίνει σε αυτά, δεν θα επηρεαστεί αυτό που βλέπει ο χρήστης (το UI).

Στη συνέχεια, το κομμάτι του view είναι υπεύθυνο για την διεπαφή του χρήστη. Στην ουσία είναι αυτό που βλέπει ο χρήστης, άρα γενικότερα το σημείο επαφής του χρήστη με την εφαρμογή. Στην αρχιτεκτονική του MVM, το κομμάτι του view, θα πρέπει να μην είναι φορτωμένο με δεδομένα, αλλά να περιορίζεται στην παρουσίαση τους, κάνοντας πιο εύκολη την συντήρηση του κώδικα.

Τελευταίο κομμάτι της αρχιτεκτονικής, αποτελεί το ViewModel. Το ViewModel είναι ο συνδετικός κρίκος ανάμεσα στο model και των δύο. Ο λόγος ύπαρξης του παραπάνω, είναι να παρέχει δεδομένα και λειτουργικότητα στο view, αλλά και να επικοινωνεί και με το μοντέλο. Με τον τρόπο αυτό, ο κώδικας του view διατηρείται πιο ελαφρύς, χωρίς να πρέπει να διαχειριστεί και δεδομένα, ενώ σε επίπεδο προγραμματιστικό, ο κώδικας είναι ευανάγνωστος και η συντήρηση του γίνεται πιο αποδοτική.

## 2.8 Clean Architecture

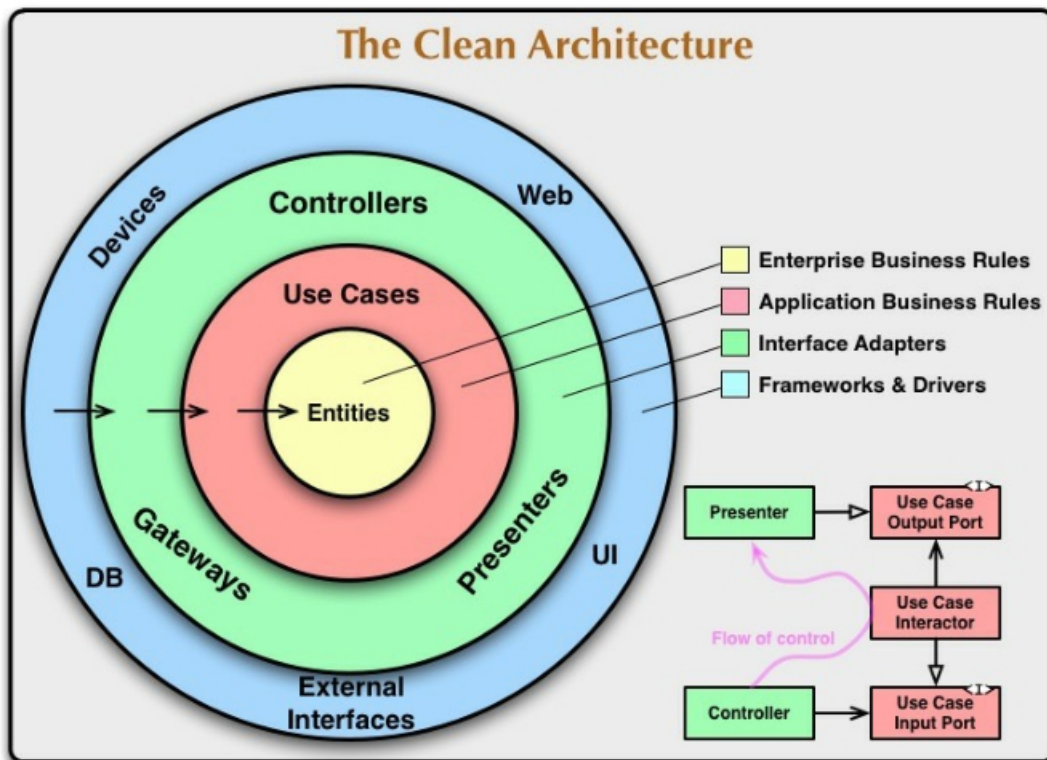
Η clean architecture είναι μία αρχιτεκτονική, της οποίας το χαρακτηριστικό είναι να διαχωρίζει τα μέρη μιας εφαρμογής ή ενός έργου σε επίπεδα[9]. Η γραφική αναπαράσταση της αρχιτεκτονικής απαρτίζεται από πολλούς ομόκεντρους κύκλους, που καθένας απ αυτών αντιστοιχεί σε κάποιο στάδιο. Η παραπάνω κύκλοι είναι σχηματικοί. Αυτό σημαίνει ότι δεν υπάρχει κάποιος κανόνας που να υπαγορεύει πόσοι θα είναι αυτοί στο σύνολο τους. Υπάρχουν περιπτώσεις που μπορεί να είναι περισσότεροι ή λιγότεροι. Πάντα όμως θα ισχύει ο κανόνας της εξάρτησης που θα αναπτύξουμε παρακάτω. Ο στόχος που έχει η clean architecture, είναι και αυτός που την κάνει πολύ σημαντική. Έχει τη δυνατότητα να οργανώνει τον κώδικα με αποτέλεσμα να ενσωματώνει την επιχειρηματική λογική, ενώ παράλληλα να διατηρεί τον κώδικα “καθαρό”.

Με τη χρήση της clean architecture, ο προγραμματιστής απολαμβάνει και κάποια προνόμια σχετικά με την εξέλιξη του έργου του. Αρχικά, με την παραπάνω αρχιτεκτονική δίνεται η δυνατότητα να απομονωθεί επιχειρηματική λογική, ώστε να προβούμε σε τμηματικές δοκιμές του έργου. Με σκοπό να συντηρήσουμε τον κώδικα ποιοτικό και αξιόπιστο. Ένα ακόμα προνόμιο, αποτελεί ο διαχωρισμός σε τρίμματα λέιζερ του έργου. Με την παραπάνω δυνατότητα η διαχείριση του κώδικα γίνεται πιο εύκολη ενώ παράλληλα η τροποποίηση και συντήρηση του, καθίσταται πιο εύκολη. Συνεχίζοντας, προσφέρει μία ευελιξία όσο αφορά τα εξωτερικά εργαλεία τα οποία χρησιμοποιεί εφαρμογή. Δηλαδή η οποιαδήποτε αλλαγή φραγμένες ή άλλων βιβλιοθηκών που κρίνεται απαραίτητη αφού τον προγραμματιστή, δεν επηρεάζει τον κώδικα. Τέλος, συνεισφέρει στην στην εργασία μεταξύ των προγραμματιστών, καθώς μπορούν να διορθώνουν ή να προσθέτουν κώδικα ταυτόχρονα σε διαφορετικά σημεία χωρίς αυτό να επηρεάζει τον άλλον.

Κρίσιμο για την αρχιτεκτονική θεωρείται ένα ακόμα σημείο. Η αρχή της εξάρτησης (The Dependency Rule). Η παραπάνω αρχή θέτει τον τρόπο με τον οποίο το έργο θα διαχειρίζεται διάφορα στοιχεία. Βάση του παραπάνω, ισχύει ότι κάθε κύκλος μπορεί να αναφέρεται μόνο σε κάποιον από τους εσωτερικούς κύκλους. Από την άλλη, κανένας εσωτερικός κύκλος δεν μπορεί να γνωρίζει το οτιδήποτε για κάποιον κύκλο ο οποίος είναι εξωτερικά από αυτόν. Πρακτικά, για εσωτερική κύκλοι εξαρτώνται από τους εξωτερικούς κύκλους. Με την τήρηση αυτής της αρχής οι αλλαγές σε οποιοδήποτε επίπεδο, δεν έχουν επίδραση στα υπόλοιπα.

### 2.8.1 Frameworks και Drivers

Το επίπεδο των frameworks και drivers απαρτίζεται από εξωτερικές εξαρτήσεις που χρησιμοποιούνται στην εφαρμογή. Αυτά μπορεί να είναι λογισμικά ή εργαλεία όπως για παράδειγμα κάποια βάση δεδομένων ή το UI της εφαρμογής. Πρακτικά απομονώνουμε αυτά



Εικόνα 2.5 : Σχεδιάγραμμα Clean Architecture

τα στοιχεία από τον πυρήνα της εφαρμογής, έτσι ώστε σε περίπτωση που αλλάξουν να μην επιφέρουν κάποιο πρόβλημα. Γεγονός που προσδίδει σταθερότητα στην εφαρμογή.

## 2.8.2 Interface Adapters

Σε αυτό το επίπεδο, γίνεται ο μετασχηματισμός των δεδομένων (mapping) από τη μορφή που είναι πιο κατάλληλη για τις επιχειρηματικές οντότητες (entities) και τα σενάρια χρήσης (use cases) σε μια μορφή που εξυπηρετεί τις εξωτερικές λειτουργίες, όπως μια βάση δεδομένων.

Συγκεκριμένα θα περιλαμβάνει data source τα οποία θα μετατρέπουν τα δεδομένα ανάλογα με τις ανάγκες των εξωτερικών λειτουργιών, όπως μία βάση δεδομένων. Με τον τρόπο αυτό επιτυγχάνεται το γεγονός ότι ο πυρήνας παραμένει ανεξάρτητος από τις λεπτομέρειες υλοποίησης των εξωτερικών στρωμάτων. Έτσι με τον τρόπο αυτό δεν χρειάζεται κανένας κώδικας που αφορά τον πυρήνα του έργου να έχει εξάρτηση από τις λεπτομέρειες της εξωτερικής υλοποίησης.

### 2.8.3 Use Cases

Σε επίπεδο αυτό φιλόξενη επιχειρηματικούς κανόνες που αφορούν την εφαρμογή, οι οποίοι απορρέουν από την επιχειρηματική λογική. Αναλαμβάνει την εκτέλεση όλων των περιπτώσεων χρήσης (use cases) . Με τη σειρά τους αυτά κάνουν μόνον, για να πετύχουν τους στόχους τους, κάνουν χρήση των επιχειρηματικών κανόνων. Ενώ παράλληλα οργανώνουν την ροή δεδομένων από και προς τις οντότητες. Στο επίπεδο αυτό δεν αναμένεται να επηρεαστεί από κάποια οντότητα ή κάποια εξωτερική λειτουργία, όπως για παράδειγμα μία βάση δεδομένων , στο UI της εφαρμογής κτλ. Από την άλλη όμως, οποιαδήποτε αλλαγή στην λειτουργία της εφαρμογής επηρεάσει τα use cases, αυτομάτως θα επηρεάσει και τον κώδικα σε αυτό το επίπεδο.

### 2.8.4 Entities

Στην clean architecture, οι οντότητες αποτελούν τον πυρήνα του συστήματος. Εκφράζουν τις επιχειρηματικές έννοιες και κανόνες τους οποίους θα πρέπει να ακολουθεί η εφαρμογή. Συγκεκριμένα απαρτίζουν την μοντελοποίηση των πραγματικών εννοιών που έχουν βαρύτητα για την επιχείρηση. Στοιχεύουν στην αναπαράσταση της επιχειρηματική λογική, ενώ είναι ανεξάρτητες με τις λεπτομέρειες υλοποίησης. Είναι ανθεκτικές έλεγες στην εξωτερικών παραγόντων, ενώ παράλληλα θα πρέπει να περιμένω σταθερές ανεξάρτητα από τις μεταβολές που γίνονται στην εφαρμογή.

## 2.9 Clean Architecture με MVVM

Ο συνδυασμός της clean architecture και του MVVM ερχέται να καλύψει κάποια ζητήματα τα οποία απορρέουν από το MVVM. Όπως αυτό της διαχώρισης του επιχειρησιακού μέρους του project. Τα επίπεδα της clean architecture αντιστοιχίζονται πλήρως με αυτά του MVVM, εξασφαλίζοντας έτσι την διαχωρισμένη δομή και την κατανομή των εργασιών. Το ViewModel, αντιπροσωπεύοντας το MVVM, καθιστάται ως ο συνδετικός κρίκος ανάμεσα στο επίπεδο τομέα και της διεπαφής του χρήστη, σύμφωνα πάντα με τις αρχές της clean architecture. Προσφέροντας ευκολία στην διαχείριση της εφαρμογής, καθώς και στην ανάπτυξη της.

### 2.10 Shared Preferences

Το SharedPreferences αποτελεί ένα βασικό εργαλείο στην ανάπτυξη εφαρμογών για το Android, καθώς παρέχει έναν εύκολο τρόπο αποθήκευσης και ανάγνωσης απλών δεδομένων ρυθμίσεων και προτιμήσεων του χρήστη[10]. Είναι κατάλληλο για περιπτώσεις όπου οι

πληροφορίες που αποθηκεύονται αφορούν τον τρόπο με τον οποίο ο χρήστης προτιμά να διαμορφώνει την εφαρμογή, όπως το θέμα της διεπαφής χρήστη ή η προτίμηση για την εμφάνιση περιεχομένου σε μια λίστα ή ένα πλέγμα. Το SharedPreferences αντανακλά αμέσως τις αλλαγές που πραγματοποιούνται από τους χρήστες, αλλά είναι σημαντικό να σημειωθεί ότι αυτές οι αλλαγές ενδέχεται να μην διατηρηθούν μόνιμα σε περιπτώσεις όπως σφάλματα εφαρμογής ή τερματισμός. Επομένως, δεν συνιστάται για την αποθήκευση ευαίσθητων δεδομένων όπως οι ρυθμίσεις ασφαλείας ή απορρήτου, καθώς οι αλλαγές ενδέχεται να χαθούν χωρίς προειδοποίηση. Για τη διατήρηση δεδομένων υψηλού επιπέδου και την αντιμετώπιση τέτοιων σεναρίων, είναι προτιμότερο να χρησιμοποιηθούν λύσεις όπως το Room ή το DataStore, τα οποία παρέχουν περισσότερες δυνατότητες διαχείρισης και συνεπώς αυξημένη αξιοπιστία.

## 2.11 Datastore

Το Jetpack DataStore είναι μια λύση αποθήκευσης δεδομένων που σας επιτρέπει να αποθηκεύετε ζεύγη κλειδιών-τιμών ή ηλεκτρολογημένα αντικείμενα με buffer πρωτοκόλλου[11]. Το DataStore χρησιμοποιεί τις κορουτίνες Kotlin και το Flow για την αποθήκευση δεδομένων ασύγχρονα, σταθερά και συναλλακτικά.

Αντικαθιστά το παραδοσιακό SharedPreferences, προσφέροντας βελτιωμένη ασφάλεια, απόδοση και ευελιξία. Η κύρια διαφορά του Preferences DataStore από το SharedPreferences είναι ότι χρησιμοποιεί έναν ασύγχρονο, μη-μπλοκαρισμένο μηχανισμό αποθήκευσης δεδομένων, ο οποίος επιτρέπει τη διαχείριση μεγάλων ογκωδών δεδομένων χωρίς να δημιουργεί απροσδιόριστη καθυστέρηση στην απόκριση της εφαρμογής. Το DataStore παρέχει επίσης ενσωματωμένη υποστήριξη για την ασφαλή ανάγνωση και εγγραφή δεδομένων, καθώς και δυνατότητες διαχείρισης των διαφόρων τύπων δεδομένων, όπως μοναδικές τιμές, λίστες και πίνακες. Επιπλέον, η δυνατότητα ασύγχρονης αποθήκευσης επιτρέπει την εκτέλεση εργασιών αποθήκευσης δεδομένων χωρίς να διακόπτεται η κύρια διεπαφή χρήστη, βελτιώνοντας την εμπειρία του χρήστη και την απόδοση της εφαρμογής. Συνολικά, το Jetpack Preferences DataStore προσφέρει μια εξελιγμένη και αξιόπιστη λύση για τη διαχείριση προτιμήσεων και δεδομένων στο Android, επιτρέποντας στους προγραμματιστές να δημιουργήσουν αξιόπιστες και αποδοτικές εφαρμογές.

## 2.12 Hilt

Το Hilt είναι ένα εργαλείο που παρέχεται από το Android Jetpack για τη διαχείριση των εξαρτήσεων (dependency injection) στις εφαρμογές Android[12]. Η χρήση του Hilt σας επιτρέπει να καθορίσετε τον τρόπο που οι εξαρτήσεις ενσωματώνονται στα στοιχεία της εφαρμογής σας, παρέχοντας προβλέψιμες και εύκολα διαχειρίσιμες συνθήκες.

Με τη χρήση του Hilt, μπορείτε να ανακουφίσετε τη διαδικασία εγκατάστασης εξαρτήσεων, να διαχειριστείτε τον κύκλο ζωής τους, και να ελέγξετε την πρόσβαση σε κοινόχρηστα αντικείμενα (shared objects) μέσω της dependency injection. Αυτό επιτυγχάνεται με τη χρήση των συνόλων (annotated classes) και των εργοστασίων (factories) που παρέχει το Hilt. Το Hilt βασίζεται στο Dagger, ένα άλλο εργαλείο διαχείρισης εξαρτήσεων για τη Java και την πλατφόρμα Android. Το Hilt παρέχει μια πιο απλοποιημένη και πιο εύχρηστη προσέγγιση από το Dagger.

## 2.13 Firebase

Το Google Firebase είναι μία πλατφόρμα ανάπτυξης έργων που προσφέρει ένα ευρύ φάσμα εργαλείων για την δημιουργία την ανάπτυξη διαχείριση εφαρμογών για διάφορες πλατφόρμες, όπως το Android, το iOS, το web και πολλά άλλα[13]. Για την κάλυψη του backend κομμάτι της εφαρμογής, χρησιμοποιήθηκε η ενλόγω βάση δεδομένων.

Οι λειτουργίες του είναι πολλές. Ειδικότερα, κάποιες από αυτές, που χρησιμοποιεί και η εφαρμογή, είναι το διευθυντικό ποίηση των χρηστών (authentication), η αποθήκευση δεδομένων σε cloud, διάφορα push notifications δημιουργώντας καμπάνιες, διάφορες αναλύσεις σχετικά με την εφαρμογή (Analytics), διάφορες αναλύσεις σφαλμάτων μέσα στην εφαρμογή (Crashlytics) κτλ.

Με τη χρήση του Firebase, οι προγραμματιστές μπορούν να επωφεληθούν από τη δύναμη του cloud για την ανάπτυξη γρήγορων, αξιόπιστων και ασφαλών εφαρμογών, ενώ ταυτόχρονα μειώνουν τον χρόνο ανάπτυξης και την ανάγκη για υποδομές διαχείρισης. Η προσαρμοστικότητα και η ευκολία χρήσης του Firebase το καθιστούν μια δημοφιλή επιλογή για τους προγραμματιστές, που εργάζονται σε εφαρμογές διαφορετικών κλιμάκων και συνθηκών ανάπτυξης.

### 2.13.1 Analytics

Το Firebase Analytics είναι εργαλείο αναλυτικής δεδομένων, που παρέχει λεπτομερείς πληροφορίες σχετικά με τη χρήση και τη συμπεριφορά των χρηστών στην εφαρμογή[14]. Με το Firebase Analytics, υπάρχει η δυνατότητα παρακολούθησης των χρηστών, όπως τις ενέργειες τους σε εφαρμογές, τις αλληλεπιδράσεις με συγκεκριμένα στοιχεία της εφαρμογής, τον ρυθμός μετατροπών και πολλά άλλα. Αυτές οι πληροφορίες μπορούν να χρησιμοποιηθούν, για να κατανοηθούν καλύτερα οι προτιμήσεις και οι συνήθειες των χρηστών, με σκοπό να βελτιωθεί η εμπειρία τους προσαρμόζοντας την εφαρμογή, για να ανταποκρίνεται καλύτερα στις ανάγκες τους. Με τη δυνατότητα παρακολούθησης πολλαπλών μετρήσεων, όπως εγκαταστάσεις εφαρμογών, συμβάντα χρηστών, ανατροφοδοτήσεις,

κατανοήσεις κοινού και πολλά άλλα, το Firebase Analytics είναι ένα ανεκτίμητο εργαλείο για την ανάπτυξη, τη βελτίωση και την επιτυχία της εφαρμογής σας.

### 2.13.2 Crashlytics

Το Firebase Crashlytics είναι ένα εργαλείο ανάλυσης και εύρεσης σφαλμάτων για ανάπτυξη εφαρμογών. Το Crashlytics παρέχει λεπτομερείς πληροφορίες σχετικά με τα σφάλματα και τις εξαιρέσεις που συμβαίνουν στην εφαρμογή[15], συμπεριλαμβανομένων στοιχείων, όπως η αιτία του σφάλματος, η συχνότητα εμφάνισης, οι πληροφορίες σχετικά με τη συσκευή και το λειτουργικό σύστημα, και άλλα σχετικά δεδομένα που βοηθούν στην εντοπισμό και την αντιμετώπιση των προβλημάτων.

Ένα από τα κύρια πλεονεκτήματα του Firebase Crashlytics είναι η άμεση και αυτόματη αποστολή αναφορών σφαλμάτων από τις εφαρμογές προς το Firebase, χωρίς την ανάγκη για επεμβάσεις από τον χρήστη. Αυτό, επιτρέπει στους προγραμματιστές να λαμβάνουν άμεσα ειδοποιήσεις για σφάλματα και να αντιμετωπίζουν τα προβλήματα πριν αυτά επηρεάσουν την εμπειρία των χρηστών. Επιπλέον, το Crashlytics παρέχει δυνατότητες για εξατομικευμένες αναφορές σφαλμάτων, στατιστικά στοιχεία και αναλύσεις, καθώς και δυνατότητες σύγκρισης των εκδόσεων της εφαρμογής για την παρακολούθηση της πρόοδος της ανάπτυξης. Με αυτόν τον τρόπο, το Firebase Crashlytics είναι ένα απαραίτητο εργαλείο για τη διασφάλιση της σταθερότητας και της ποιότητας των εφαρμογών.

### 2.13.3 Push Notifications

Σημαντικό για μία εφαρμογή κρίνεται και η αλληλεπίδραση της εφαρμογής με τον χρήστη. Στον τομέα αυτό η firebase δίνει ένα δυνατό εργαλείο, τα push notifications. Πρόκειται για ειδοποιήσεις που έρχονται στην εφαρμογή, οι οποίες δημιουργούνται από διάφορες καμπάνιες που φτιάχνει ο διαχειριστής της εφαρμογής μέσω της firebase[16]. Με τον τρόπο αυτό στην παρούσα εφαρμογή μπορεί ο χρήστης να ενημερώνεται για διάφορα events, για προσθήκη προϊόντων που είναι προς αγορά, για κάποια νέα ανακοίνωση και για οτιδήποτε άλλο κριθεί απαραίτητο από τον διαχειριστή. Μέσω του Firebase, η δημιουργία και η διαχείριση των ειδοποιήσεων είναι εύκολη, χρησιμοποιώντας το Firebase Cloud Messaging (FCM). Αυτό επιτρέπει την αποστολή ειδοποιήσεων σε μεμονωμένους χρήστες ή σε ομάδες χρηστών, βάσει κριτηρίων όπως η τοποθεσία, ο τύπος συσκευής ή άλλα προσαρμοσμένα χαρακτηριστικά. Δυνατή είναι επίσης η ανάλυση δεδομένων του Firebase, για να μετρηθεί η απόδοση των ειδοποιήσεων και να βελτιωθεί η στρατηγική επικοινωνίας της επιχείρησης με τους χρήστες. Γίνεται αντιληπτό ότι λόγω της ευελιξίας του Firebase, είναι δυνατόν να δημιουργηθούν οι εξατομικευμένες ειδοποιήσεις, που είναι προσαρμοσμένες στις ανάγκες των χρηστών αλλά και της εφαρμογής.

## Κεφάλαιο 3ο: Ανάλυση Κώδικα της Εφαρμογής

### 3.1 Gradle

Το Android Gradle είναι ένα ισχυρό εργαλείο που παρέχει πολλές δυνατότητες προσαρμογής και επέκτασης των διαδικασιών build στο Android Studio[17]. Οι προσθήκες στο Android Gradle μπορούν να περιλαμβάνουν:

- 1) **Επεκτάσεις (Plugins):** Υπάρχει δυνατότητα προσθήκης plugins, που παρέχουν επιπλέον λειτουργικότητα στο Gradle. Για παράδειγμα στο παρακάτω στιγμιότυπο του gradle της εφαρμογής, έχει γίνει εισαγωγή του plugin της firebase, του hilt, της kotlin κ.α. με σκοπό την ενσωμάτωση των λειτουργιών τους στην εφαρμογή.

```
plugins {  
    id 'com.android.application'  
    id 'org.jetbrains.kotlin.android'  
    id 'androidx.navigation.safeargs.kotlin'  
}  
plugins {  
    id 'kotlin-kapt'  
    id 'dagger.hilt.android.plugin'  
    id 'com.google.gms.google-services'  
    id("com.google.firebase.crashlytics")  
}
```

Παράρτημα 3.1: Διάφορα plugins που υπάρχουν στο gradle

- 2) **Εξαρτήσεις (Dependencies):** Μπορείτε να προσθέσετε εξαρτήσεις στο αρχείο Gradle για να χρησιμοποιήσετε εξωτερικές βιβλιοθήκες ή πακέτα κώδικα στην εφαρμογή σας. Αυτό μπορεί να γίνει εύκολα, αρκεί με την προσθήκη της εντολής implementation συνοδευόμενη από το πακέτο της βιβλιοθήκης και την επιθυμητή έκδοση. Αυτό επιτρέπει τη χρήση τρίτων βιβλιοθηκών όπως το Retrofit για τις δικτυακές επικοινωνίες, των Google Maps για την εισαγωγή χαρτών, του datastore το οποίο βοηθάει στη διαχείριση και την αποθήκευση δεδομένων στην εφαρμογή και πολλων άλλων ακόμα. Το στιγμιότυπο που ακολουθεί περιέχει κάποια εξαρτήσεις από αυτές που περιέχει η εφαρμογή.

```

implementation("androidx.datastore:datastore-preferences:1.0.0")
implementation("androidx.datastore:datastore-preferences-rxjava3:1.0.0")
implementation("androidx.datastore:datastore-preferences-core:1.0.0")
implementation("androidx.datastore:datastore-preferences:1.0.0-alpha08")
implementation('com.google.firebase:firebase-messaging-ktx:23.2.1')
implementation(platform("com.google.firebase:firebase-bom:32.3.1"))
implementation("com.google.firebase:firebase-firestore-ktx")
implementation("com.google.firebase:firebase-auth-ktx")
implementation("com.google.firebase:firebase-messaging-ktx")
implementation("com.google.firebase:firebase-crashlytics-ktx")

```

Παράρτημα 3.2: Διάφορες εξαρτήσεις που υπάρχουν στο gradle

3) **Ρυθμίσεις Εκδόσεων:** Μέσω του Gradle, μπορούμε να διενεργήσουμε διάφορες παραμετροποιήσεις σχετικά με τη διαδικασία κατασκευής (build) του προγράμματος μας. Αυτές οι παραμετροποιήσεις περιλαμβάνουν την καθορισμένη ελάχιστη, μέγιστη και επιθυμητή έκδοση του SDK στην οποία θα λειτουργεί η εφαρμογή μας. Επιπλέον, μπορούμε να προσδιορίσουμε την έκδοση της Java που θα χρησιμοποιείται. Αυτές οι παραμετροποιήσεις επιτρέπουν στους προγραμματιστές να προσαρμόσουν τη διαδικασία κατασκευής του έργου τους σύμφωνα με τις ανάγκες και τις προδιαγραφές της εφαρμογής τους.

```

android {
    namespace 'com.example.riverviewproject'
    compileSdk 33
    defaultConfig {
        applicationId "com.example.riverviewproject"
        minSdk 24
        targetSdk 33
    }
}

```

Παράρτημα 3.3: Ρύθμιση της έκδοσης του SDK

### 3.2 Android Manifest

Το αρχείο AndroidManifest.xml περιέχει τις καθοριστικές πληροφορίες που χρειάζονται για την εφαρμογή σας στο λειτουργικό σύστημα Android[18]. Ας εξετάσουμε πιο αναλυτικά τα περιεχόμενα του:

- **Άδειες (Permissions):** Στις γραμμές <uses-permission> καθορίζονται οι άδειες που απαιτούνται για την εφαρμογή. Στην εν λόγω εφαρμογή, ζητείται η πρόσβαση στο διαδίκτυο και στην κατάσταση του δικτύου, τα οποία κρίνονται συχνά απαραίτητα για εφαρμογές που χρησιμοποιούν δίκτυα.
- **Εφαρμογή (Application):** Στις γραμμές <application> περιλαμβάνει πληροφορίες για την εφαρμογή όπως το όνομα, το εικονίδιο, το θέμα, και άλλες ρυθμίσεις. Εδώ ορίζεται επίσης η κύρια δραστηριότητα της εφαρμογής, η οποία είναι η MainActivity στο παράδειγμά σας. Η MainActivity είναι η πρώτη οθόνη που εμφανίζεται όταν ο χρήστης εκκινεί την εφαρμογή.
- **Μεταδεδομένα (Meta-data):** Οι γραμμές <meta-data> επιτρέπουν την προσθήκη επιπλέον πληροφοριών για την εφαρμογή. Στην εφαρμογή δίνεται το κλειδί API για τους χάρτες Google και η έκδοση των υπηρεσιών Google Play. Αυτές οι πληροφορίες είναι σημαντικές για την επικοινωνία της εφαρμογής με άλλες υπηρεσίες ή προϊόντα.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools">

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

  <application
    android:name=".RiverViewProject"
    android:allowBackup="false"
    android:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:screenOrientation="portrait"
    android:supportsRtl="true" android:theme="@style/
Theme.RiverViewProject" tools:targetApi="31">
```

Παράρτημα 3.4: Το Android Manifest της εφαρμογής 1/2

```

<activity
    android:name=".MainActivity"
    android:exported="true"
    android:screenOrientation="portrait"
    tools:ignore="LockedOrientationActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />
<meta-data
    android:name="com.google.android.maps.v2.API_KEY"
    android:value="AIzaSyCR9cPnKrZ7pxQ9Nax_ehwhOzlorHuNxLM" />
</application>
</manifest>

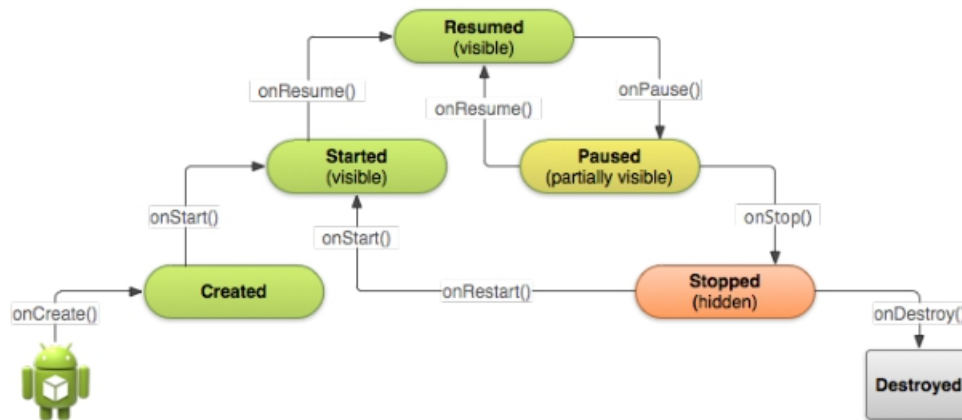
```

Παράρτημα 3.5: Το Android Manifest της εφαρμογής 2/2

Το `AndroidManifest.xml` είναι βασικό για τη διαχείριση των ρυθμίσεων και την επιτυχή λειτουργία της εφαρμογής. Κρίνεται σημαντικό να υπάρχει συνεχής ενημέρωση του αρχείου και να προστίθονται οι απαραίτητες ρυθμίσεις και δηλώσεις, ώστε η εφαρμογή να είναι άρτια αναπτυγμένη.

### 3.3 Activity

Τα `Activities` συνιστούν ουσιαστικό κομμάτι μιας `Android` εφαρμογής, καθώς αποτελούν τον πυρήνα της διαδικασίας εκκίνησης και εναλλαγής των οθονών στην πλατφόρμα. Αντίθετα με άλλα προγραμματιστικά παραδείγματα, όπου οι εφαρμογές ξεκινούν μέσω μιας κύριας μεθόδου (`main()`), στο `Android` η εκκίνηση της εφαρμογής ξεκινά με ένα `Activity`[19]. Μία εφαρμογή μπορεί να περιλαμβάνει πολλά `Activities`, αλλά το αρχικό `Activity` ορίζεται στο `AndroidManifest`. Η παρούσα εφαρμογή διαθέτει μόνο ένα `activity`, το οποίο είναι το `main activity`. Το σύστημα του `Android` εκκινεί την κλάση `Activity`, καλώντας συγκεκριμένες μεθόδους που αντιστοιχούν σε στάδια του κύκλου ζωής της. Στο σχήμα μου ακολουθεί αναπαρίστανται ακριβώς η κύκλος ζωής ενός `Activity`.



Εικόνα 3.1: Ο κύκλος ζωής ενός activity

### 3.4 Fragment

Τα Fragments αναπαριστούν επαναχρησιμοποιήσιμα τμήματα της διεπαφής χρήστη της εφαρμογής, έχοντας τη δυνατότητα να ορίζουν και να διαχειρίζονται τη δική τους διάταξη. Κάθε Fragment έχει τον δικό του κύκλο ζωής και μπορεί να αντιδράσει στα δικά του συμβάντα[20]. Τα Fragments δεν είναι αυτοτελή, αλλά πρέπει να ενσωματώνονται εντός μιας δραστηριότητας ή άλλου τμήματος της διεπαφής. Με αυτόν τον τρόπο, τα Fragments επιτρέπουν τη δημιουργία ευέλικτων και διαισθητικών διεπαφών χρήστη, ενισχύοντας την εμπειρία του χρήστη και επιτρέποντας την αποτελεσματική διαχείριση της διεπαφής σε διαφορετικές οθόνες και συσκευές.

```
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent" xmlns:tools="http://
schemas.android.com/tools" android:background="@color/white"
    tools:context=".ui.home.HomeFragment">
<androidx.compose.ui.platform.ComposeView
    android:id="@+id/compose_view1"
    android:layout_width="0dp"
    android:layout_height="0dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>;
```

Παράρτημα 3.6: Παράδειγμα παραδείγμα Fragment

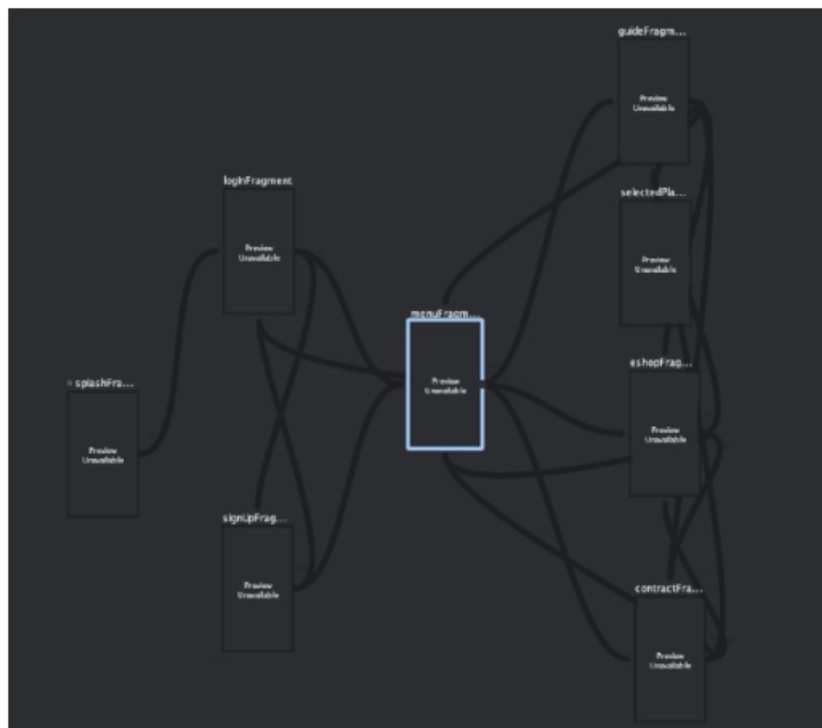
### 3.5 Navigation Components

Το Navigation Component είναι ένα ισχυρό εργαλείο που παρέχεται από το Jetpack της Google για τη διαχείριση της πλοήγησης σε εφαρμογές Android[21]. Αποτελεί μια προηγμένη λύση που επιτρέπει τη διαχείριση των μεταβάσεων μεταξύ διαφορετικών οθονών και προβολών με έναν δηλωτικό και συνεπή τρόπο.

Με το Navigation Component, οι προγραμματιστές μπορούν να οργανώσουν τη δομή πλοήγησης της εφαρμογής τους σε ένα κεντρικό μέρος, το αρχείο navigation graph, όπου καθορίζονται οι σχέσεις μεταξύ των διάφορων προορισμών. Αυτό επιτρέπει την ευκολότερη διαχείριση και ανάπτυξη της πλοήγησης, καθώς επίσης και την αυτόματη δημιουργία των απαιτούμενων μεταβάσεων και των UI στοιχείων για την πλοήγηση.

Η δομή του Navigation Component στηρίζεται σε δύο βασικά στοιχεία: το navigation graph και το NavHostFragment.

Το navigation graph είναι ένα αρχείο XML που καθορίζει τον τρόπο πλοήγησης στην εφαρμογή. Περιλαμβάνει τους προορισμούς (destinations) της εφαρμογής, όπως fragment, activities και άλλα navigation graphs, καθώς και τις σχέσεις και τις μεταβάσεις μεταξύ αυτών.



Εικόνα 3.2: Το nav\_graph της εφαρμογής

Το NavHostFragment είναι ένα ειδικό fragment που χρησιμοποιείται ως περιβάλλον για την πλοήγηση στην εφαρμογή. Αντί να φορτώνει άμεσα τα fragment, το NavHostFragment φιλοξενεί έναν NavHost, ο οποίος αναλαμβάνει τη διαχείριση των μεταβάσεων μεταξύ των προορισμών που ορίζονται στο navigation graph.

Κατά την εκκίνηση της εφαρμογής, ο Navigation Component φορτώνει το navigation graph και χρησιμοποιεί το NavHostFragment για να παρουσιάσει τον αρχικό προορισμό στην οθόνη. Καθώς ο χρήστης πλοηγείται στην εφαρμογή, ο Navigation Component διαχειρίζεται δυναμικά τις μεταβάσεις μεταξύ των προορισμών, επιτρέποντας έτσι τη δημιουργία πολύπλοκων δομών πλοήγησης με ελάχιστον κώδικα και αυξάνοντας τη συντηρησιμότητα της εφαρμογής.

### 3.6 ComposeView

Το ComposeView αντιπροσωπεύει ένα από τα κύρια στοιχεία της σύγχρονης ανάπτυξης εφαρμογών για το Android. Είναι δημοφιλές πλέον εργαλείο, που προσφέρει έναν διαδραστικό τρόπο για τη δημιουργία των χρήσιμων διεπαφών χρήστη (UI) με τη χρήση δηλωτικού κώδικα. Η χρήση του ComposeView επιτρέπει στους προγραμματιστές να δημιουργήσουν UI στοιχεία χρησιμοποιώντας απλή, συνολική σύνταξη Kotlin, απαλλάσσοντας τους από την ανάγκη γραφής XML. Με το ComposeView, οι προγραμματιστές μπορούν να δημιουργήσουν επικοινωνιακά UI στοιχεία που ανταποκρίνονται δυναμικά στις αλλαγές των δεδομένων ή των ενεργειών του χρήστη. Το ComposeView προσφέρει μια πληθώρα από ενσωματωμένα UI στοιχεία, όπως κουμπιά, καρτέλες, λίστες και πολλά άλλα, και είναι εύκολο στη χρήση μέσω της ενσωματωμένης υποστήριξης στην πλατφόρμα Android Studio. Με τον συνδυασμό της ευκολίας χρήσης και της ισχυρής λειτουργικότητας, το ComposeView αποτελεί μια εξαιρετική επιλογή για τη δημιουργία εντυπωσιακών και αποδοτικών εφαρμογών Android.

```
<androidx.compose.ui.platform.ComposeView
    android:id="@+id/compose_view1"
    android:layout_width="0dp"
    android:layout_height="0dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

Παράρτημα 3.7: Χρήση Compose View

### 3.7 Compose

Jetpack Compose είναι ένα σύγχρονο toolkit UI ανάπτυξης που ανακοινώθηκε από την Google το 2019. Το Compose σαν εργαλείο βοηθάει τους προγραμματιστές να φτιάξουν εφαρμογές Android με πιο βολικά και ευπροσάρμοστα γραφικά[22]. Ουσιαστικά, ο κάθε προγραμματιστής μπορεί να δημιουργήσει UI με την χρήση κώδικα Kotlin, ενάντι των αρχείων διάταξης XML. Με το Compose, υπάρχει δυνατότητα ορισμού των στοιχείων διεπαφής χρήστη ως συναρτήσεις συντεθειμένες από τους προγραμματιστές, οι οποίες μπορούν να συνδυαστούν και να δημιουργήσουν πολύπλοκα γραφικά. Η διαχείριση των αλλαγών που ορίζονται από τον χρήστη διευκολύνεται, ενώ ταυτοχρόνων παρέχεται μια μεγάλη γκάμα εργαλείων για κινούμενα σχέδια και οπτικά εφέ.

#### Οφέλη από την χρήση Compose

- Συνοπτικός και ευανάγνωστος κώδικας
- Αποτελεσματικότητα και αποδοτικότητα
- Ενσωματωμένη υποστήριξη για το Υλικό Σχεδιασμού (Material Design)
- Διευκόλυνση στην εφαρμογή λειτουργιών προσβασιμότητας

#### Δυνατότητες προσβασιμότητας

Οι λειτουργίες προσβασιμότητας στο Compose είναι τα εργαλεία αλλά και οι τεχνικές που παρέχουν την ευκαιρία στους προγραμματιστές να δημιουργήσουν εύχρηστες και βολικές διεπαφές χρήστη, ιδιαίτερα για τα άτομα με αναπηρίες. Αυτή η δυνατότητα, συμβάλει στο να διαμορφωθούν περιεκτικές και προσβάσιμες εφαρμογές για όλο το κοινό, ανεξαρτήτως των ικανοτήτων ή περιορισμών που μπορεί να έχουν.

#### 3.7.1 Διαφορές μεταξύ Jetpack Compose και XML UI toolkit στο android

Η κύρια διαφορά μεταξύ του Compose και XML UI toolkit στο Android εντοπίζεται στον τρόπο με τον οποίο ορίζουν και δημιουργούν τις διεπαφές χρήστη. Πιο συγκεκριμένα:

##### XML UI toolkit:

- Χρησιμοποιεί τη γλώσσα σήμανσης XML, για να ορίσει τα στοιχεία διεπαφής χρήστη και τις ιδιότητές τους.
- Απαιτεί ένα ξεχωριστό αρχείο διάταξης XML για κάθε οθόνη ή διάταξη.
- Περιλαμβάνει σε μεγάλο βαθμό κώδικα boilerplate για τον διαχειρισμό συμβάντων του UI όπως και των αλλαγών κατάστασης.

- Παρέχει έναν οπτικό επεξεργαστή (π.χ. Android Studio) για να βοηθήσει στον σχεδιασμό των διατάξεων.

### Jetpack Compose

- Χρησιμοποιεί κώδικα Kotlin ώστε να ορίσει τα στοιχεία διεπαφής χρήστη και τις ιδιότητές τους ως συναρτήσεις που μπορούν να συνθέσουν.
- Η δημιουργία διεπαφής χρήστη και ο σχεδιασμός διατάξεων γίνεται ευκολότερος
- Περιορίζει τον κώδικα boilerplate και παρέχοντας με αυτόν τον τρόπο έναν πιο συνοπτικό και ευανάγνωστο κώδικα.
- Παρέχει ένα εργαλείο προεπισκόπησης (π.χ. Android Studio) για βοήθεια με το σχεδιασμό διάταξης.
- Αύξηση αποδοτικότητας και αποτελεσματικότητας συγκριτικά με το toolkit XML UI.

Ανακεφαλαιώνοντας, γίνεται αντιληπτό ότι το Jetpack Compose έχει την δυνατότητα να προσφέρει μια εκσυγχρονισμένη προσέγγιση στη δημιουργία UI του Android, διευκολύνοντας την δημιουργία δυναμικών και ανταποκρινόμενων UI, με αποτέλεσμα να αποκτά μια δημοφιλή θέση αλλά και επιλογή για τους προγραμματιστές Android UI.

#### 3.7.2 Στήλη (column)

Η Στήλη (column) είναι ένα βασικό στοιχείο στο Jetpack Compose, που χρησιμεύει ως συνθετική λειτουργία που οργανώνει τα παιδιά της με κατακόρυφο τρόπο, παρόμοια με την κατακόρυφη διάταξη σε μια παραδοσιακή Γραμμική διάταξη (LinearLayout) Android[23]. Ουσιαστικά, μια Στήλη (column) λειτουργεί ως ένα κατακόρυφο δοχείο (container) για τη συγκράτηση άλλων συνθετικών στοιχείων, απλοποιώντας τη δημιουργία κατακόρυφα δομημένων διεπαφών χρήστη. Μια αξιοσημείωτη ιδιότητα της Στήλης είναι η κατακόρυφη διάταξη, η οποία διαχειρίζεται η παράμετρος Arrangement.

```
Column(modifier = modifier.padding(horizontal =
SpacingHalf_8dp).verticalScroll(rememberScrollState())) {
    HeaderComponent(modifier = Modifier.padding(top = SpacingQuarter_4dp),
onBasketClick = {
        onCloseButton()
    })
    Spacer(modifier = Modifier.fillMaxWidth(0.1 f).aspectRatio(6 f))
    HorizontalPagerWithIndicators(photoUiState = photoUiState)
}
```

Παράρτημα 3.8: Παράδειγμα Column

Αυτή η παράμετρος υπαγορεύει την ευθυγράμμιση και την απόσταση μεταξύ των θυγατρικών συνθετών, επιτρέποντας διάφορες διαμορφώσεις κατακόρυφων διατάξεων εντός της Στήλης.

### 3.7.3 Σειρά (row)

Στο Jetpack Compose, μια Σειρά (row) είναι μια συνάρτηση με δυνατότητα σύνθεσης που οργανώνει τα θυγατρικά της στοιχεία σε οριζόντια κατεύθυνση, παρόμοια με μια οριζόντια Γραμμική διάταξη (LinearLayout) στο παραδοσιακό Android[24]. Ουσιαστικά, μια σειρά λειτουργεί ως οριζόντιο δοχείο (container) για τη συγκράτηση άλλων συνθετικών στοιχείων, καθιστώντας εύκολη τη δημιουργία οριζόντια δομημένων διεπαφών χρήστη. Όπως μια Στήλη, μια Σειρά (row) προσφέρει μια ιδιότητα οριζόντιας διάταξης που ελέγχει την ευθυγράμμιση και την απόσταση μεταξύ των θυγατρικών συνθέσιμων στοιχείων της. Αυτό επιτρέπει διάφορες διαμορφώσεις οριζόντιων διατάξεων εντός της Σειράς για να ταιριάζουν σε διαφορετικές ανάγκες διάταξης.

```
Row(modifier = modifier.fillMaxWidth().aspectRatio(7
f).background(Color.White).clip(RoundedCornerShape(SpacingDefault_16dp)),
verticalAlignment = Alignment.CenterVertically, horizontalArrangement =
Arrangement.SpaceEvenly) {

    Image(painter = painterResource(id = R.drawable.home),
contentDescription = null, modifier = Modifier)

    Image(painter = painterResource(id = R.drawable.baseline_mode_of_travel_24),
contentDescription = null, modifier = Modifier.clickable(interactionSource =
interactionSource, indication = null) {

        onGuideClick()

    })
})
```

Παράρτημα 3.9: Παράδειγμα row

### 3.7.4 LazyColumn

Ένα βασικό χαρακτηριστικό στο Jetpack Compose είναι η εισαγωγή των LazyColumns[25]. Η Τεμπέλικη LazyColumn είναι μια συνθετική συνάρτηση που χειρίζεται αποτελεσματικά μεγάλες λίστες στοιχείων αποδίδοντας μόνο τα στοιχεία που είναι ορατά αυτήν τη στιγμή στην οθόνη. Αυτή η προσέγγιση εξοικονομεί μνήμη και βελτιώνει την απόδοση ανακυκλώνοντας και επαναχρησιμοποιώντας σύνθετα υλικά καθώς οι χρήστες περιηγούνται στη λίστα.

Η LazyColumn απλοποιεί τη δημιουργία δυναμικών λιστών με δυνατότητα κύλισης στη διεπαφή μιας εφαρμογής. Οι προγραμματιστές μπορούν εύκολα να ενσωματώσουν σύνθετες λίστες με διάφορους τύπους περιεχομένου και στυλ, διατηρώντας παράλληλα την ομαλή κύλιση και τη βέλτιστη χρήση της μνήμης, προσφέροντας έναν απλό και αποτελεσματικό τρόπο διαχείρισης μεγάλων λιστών, οι lazy στήλες ενισχύουν τη συνολική εμπειρία ανάπτυξης και την ικανοποίηση του τελικού χρήστη.

### 3.7.5 LazyRow

Η Τεμπέλικη Σειρά (LazyRow) είναι μια συνάρτηση με δυνατότητα σύνθεσης στο Jetpack Compose που οργανώνει τα θυγατρικά της συνθέσιμα σε μια οριζόντια λίστα με αποτελεσματικό τρόπο. Λειτουργεί παρόμοια με μια οριζόντια κυλιόμενη λίστα (scrolling list) ή καρουζέλ και συνθέτει και αποδίδει μόνο τα στοιχεία που είναι ορατά αυτήν τη στιγμή στην οθόνη[26]. Αυτή η προσέγγιση συμβάλλει στη βελτίωση της απόδοσης όταν αντιμετωπίζει η εφαρμογή μεγάλα σύνολα δεδομένων, εξοικονομώντας μνήμη και επεξεργαστική ισχύ. Καθώς οι χρήστες πραγματοποιούν κύλιση στη λίστα (scroll through the list), Η Τεμπέλικη Σειρά (LazyRow) ανακυκλώνει και επαναχρησιμοποιεί τα συνθέσιμα στοιχεία, διασφαλίζοντας ότι μόνο τα στοιχεία στην ορατή περιοχή είναι ενεργά και αποδίδονται. Η Τεμπέλικη Σειρά (LazyRow) προσφέρει επιλογές προσαρμογής, όπως οριζόντια διάταξη που ελέγχει την οριζόντια απόσταση μεταξύ των θυγατρικών συνθετών και κατακόρυφη ευθυγράμμιση όπου υπαγορεύει την ευθυγράμμιση των θυγατρικών συνθετικών (composables) κατακόρυφα εντός της Τεμπέλικης Σειράς (LazyRow). Αυτές οι ιδιότητες επιτρέπουν στους προγραμματιστές να προσαρμόσουν με ακρίβεια τη διάταξη και την εμφάνιση της οριζόντιας λίστας ώστε να ταιριάζει στις σχεδιαστικές απαιτήσεις τους. Συνολικά, η Τεμπέλικη Σειρά (LazyRow) είναι ένα ισχυρό εργαλείο για την αποτελεσματική δημιουργία οριζόντιων λιστών κύλισης στο Jetpack Compose.

```

LazyRow {
    items(announcementUiState.announcementUiItem.size) {
        val item = announcementUiState.announcementUiItem[it]
        AnnouncementContentItem(announcementItem = item, onClick = {
            onAnnouncementClick(item)
        })
    }
}

```

Παράρτημα 3.10 : Παράδειγμα Lazy Row

### 3.7.6 Διαφορές μεταξύ Column και Row με LazyColumn και LazyRow

Η Στήλη και Σειρά χρησιμοποιείται όταν υπάρχει ανάγκη για εμφάνιση λίστας στοιχείων που είναι όλα ορατά στην οθόνη ταυτόχρονα, για παράδειγμα, μια στήλη μπορεί να εμφανίσει μια λίστα μηνυμάτων σε μια εφαρμογή συνομιλίας, ενώ μια γραμμή μπορεί να οργανώσει ένα σύνολο κουμπιών σε ένα μενού ρυθμίσεων. Η Τεμπέλικη Στήλη και Τεμπέλικη Σειρά χρησιμοποιούνται για την εμφάνιση λίστας στοιχείων που μπορεί να μην είναι όλα ορατά στην οθόνη ταυτόχρονα, όπως για παράδειγμα μια παρουσίαση λίστας άρθρων ειδήσεων σε μία εφαρμογή που έχει δημιουργηθεί για αυτό το σκοπό, η χρήση του LazyRow είναι απαραίτητη. Η βασική διαφορά μεταξύ Column και Row έναντι LazyColumn και LazyRow είναι ότι το Column και το Row αποδίδουν όλα τα στοιχεία στη λίστα, ακόμη και εκείνα εκτός οθόνης, γεγονός που μπορεί να οδηγήσει σε προκλήσεις απόδοσης με μεγάλες λίστες. Αντίθετα, το LazyColumn και το LazyRow αποδίδουν μόνο τα αντικείμενα που προβάλλονται αυτήν τη στιγμή, προσφέροντας σημαντικές βελτιώσεις απόδοσης, ιδιαίτερα με εκτεταμένες λίστες.

### 3.7.7 Compose Box

Το Compose Box, είναι ένα κομμάτι του Jetpack Compose, που χρησιμοποιείται με σκοπό την διαμόρφωση των στοιχείων διεπαφής χρήστη στις Android εφαρμογές. Αναλυτικότερα, τα «κουτιά» στο Compose (Compose Box) αποτελούν πλαίσια, που επιτρέπουν στον προγραμματιστή να τοποθετήσουν μέσα σε αυτό το πλαίσιο στοιχεία διεπαφής χρήστη σε οποιοδήποτε σημείο επιθυμούν εντός των ορίων του πλαισίου.<sup>1</sup> Πρακτικά, οι προγραμματιστές προσθέτουν ένα οποιοδήποτε στοιχείο διεπαφής χρήστη (πχ κείμενο, κουμπιά, εικόνες) μέσα στο πλαίσιο (Box) και μπορούν να το μετακινήσουν δίνοντας την εντολή που θα το καθοδηγήσει σε ποιο σημείο πρέπει να μπει μέσα στο πλαίσιο. Με αυτόν τον τρόπο, γίνεται ένας σαφής και ακριβής έλεγχος της διάταξης των στοιχείων διεπαφής χρήστη. Συνήθως, χρησιμοποιείται συνδυαστικά με άλλα στοιχεία διάταξης, ώστε να σχηματιστούν σύνθετα σχέδια διεπαφής χρήστη σε εφαρμογές Jetpack Compose. Οι προγραμματιστές, μπορούν να ορίσουν τις παραμέτρους, μεταξύ άλλων της στοίχισης (δηλαδή, επιλογές τύπου «πάνω, κάτω, αρχή, τέλος, κέντρο», ώστε να παρέχεται ο πλήρης έλεγχος τοποθέτησης των στοιχείων) ή του χρώματος φόντου, ώστε να διαμορφώσουν την εμφάνιση και την θέση του στοιχείου διεπαφής χρήστη, μέσα στο πλαίσιο. Είναι σημαντικό να αναφερθεί ότι τα Compose Boxes μπορούν να τοποθετηθούν το ένα μέσα στο άλλο, δημιουργώντας με αυτόν τον τρόπο πιο σύνθετα σχέδια. Επίσης, σημειώνεται μια σχέση αλληλεπίδρασης, καθώς εάν υπάρχει κάποια αλλαγή στην κατάσταση της εφαρμογής, αυτό

αντικατοπτρίζεται αυτομάτως στο UI.

Ανακεφαλαιώνοντας, το Compose Box είναι ένα σημαντικό στοιχείο του Jetpack Compose, καθώς προσφέρει στους προγραμματιστές την δυνατότητα να σχηματίσουν μοντέρνες και σύγχρονες διατάξεις διεπαφών χρήστη στις Android εφαρμογές.

### 3.7.8 Launched Effect

Το LaunchedEffect στο Jetpack Compose είναι ένα εργαλείο που ασχολείται με τον χειρισμό παρενεργειών που μπορεί να προκύψουν μετά την σύνθεση συναρτήσεων. Πιο συγκεκριμένα, επιτρέπει στους προγραμματιστές να διαχειρίζονται χρονοβόρες εργασίες και δεδομένα από εξωτερικές πηγές και να ενημερώνουν την διεπαφή χρήστη με βάση τα αποτελέσματα τους. Πρέπει να σημειωθεί ότι το LaunchedEffect είναι βασισμένο σε κορουτίνες, δηλαδή νήματα που έχουν την δυνατότητα να κάνουν διεργασίες ασυγχρόνως. Με αυτόν τον τρόπο, η ταυτόχρονη λειτουργία διευκολύνεται, καθώς οι διεργασίες που γίνονται στο παρασκήνιο δεν αφαιρούν το βασικό νήμα της διεπαφής χρήστη, με την ανταπόκριση της εφαρμογής να διατηρείται σταθερά.. Επίσης, το Launched Effect εξασφαλίζει την ομαλή λειτουργία σε κάθε τομέα, με επίγνωση του κύκλου ζωής της εφαρμογής ακυρώνοντας την κατάλληλη στιγμή τις εργασίες όταν παύουν να είναι απαραίτητες. Πρακτικά, τα βήματα που ακολουθεί είναι τα εξής:

1. Ξεκινά μια κορουτίνα στο παρασκήνιο για να εκτελέσει την επιθυμητή παρενέργεια.
2. Διασφαλίζει ότι η διεργασία αυτή αρχίζει μόνο όταν εμφανίζεται η σύνθεση.
3. Ακυρώνει αυτόματα την κορουτίνα όταν αφαιρεθεί το συντεθειμένο από την σύνθεση.
4. Επιτρέπει να ρυθμιστούν οι εξαρτήσεις για τον έλεγχο της κορουτίνας με βάση τις αλλαγές στα δεδομένα ή την κατάσταση.<sup>2</sup>

Από αυτή τη διαδικασία προκύπτουν πολλά οφέλη. Αναλυτικότερα, χωρίζει τις διεργασίες σε φόντο, κάνοντας με αυτόν τον τρόπο τις συνθέσιμες λειτουργίες του χρήστη καθαρές. Όπως αναφέρθηκε και προηγουμένως, ξεκινά και σταματά αυτομάτως τις διεργασίες με βάση τον κύκλο ζωής της εφαρμογής, αποτρέποντας με αυτόν τον τρόπο τις διαρροές ή της περιττές λειτουργίες. Ουσιαστικά, το LaunchedEffect δίνει την δυνατότητα καθορισμού εξαρτήσεων από συγκεκριμένα δεδομένα. Επομένως, όταν αλλάζουν οι εξαρτήσεις, πρακτικά η διεργασία «ξανατρέχει» από την αρχή, εξασφαλίζοντας ότι η διεπαφή χρήστη ενημερώνεται με τα καινούρια δεδομένα, μειώνοντας σε μεγάλο βαθμό ταυτόχρονα την μην απαραίτητη εργασία. Τέλος, αυξάνεται η αποτελεσματικότητα και η αποδοτικότητα, καθώς οι διεργασίες εκτελούνται ξανά μόνο όταν αλλάζουν οι εξαρτήσεις τους.

### 3.7.9 Grids στο Jetpack Compose

Στο Jetpack Compose το grid (=πλέγμα) βοηθάει στην οργάνωση των στοιχείων διάταξης στην οθόνη. Σαν έναν «πίνακα»[27], παρέχει την δυνατότητα στους προγραμματιστές να τοποθετούν τα στοιχεία σε στήλες και σειρές διευκολύνοντας τον σχεδιασμό της διεπαφής χρήστη. Το Grid, επιτρέπει να οριστούν μέσα σε ένα πλέγμα οι αριθμοί στηλών, γραμμών, αλλά και το περιεχόμενο που θα παρέχει κάθε κελί. Επίσης, υπάρχει δυνατότητα καθορισμού χαρακτηριστικών τύπου, στοίχιση και απόσταση κελιών. Ως επί το πλείστον, τα πλέγματα (grids), χρησιμοποιούνται για την εμφάνιση εικόνων ή κειμένων με οργανωμένο τρόπο, δημιουργώντας έναν δομημένο τρόπο παρουσίασης των δεδομένων, και δίνουν την δυνατότητα στους χρήστες να «επικοινωνούν» με το περιεχόμενο. Για την δημιουργία διατάξεων πλεγμάτων υπάρχουν δύο τύποι Grid που μπορούν να χρησιμοποιηθούν:

- **LazyVerticalGrid:** Λειτουργεί για την εμφάνιση στοιχείων σε ένα κοντέινερ με δυνατότητα κύλισης κάθετα, ενώ για να χρησιμοποιηθεί πρέπει να οριστούν οι αριθμοί των στηλών(column), καθώς και ένα σταθερό ύψος ώστε να υπάρχει πλήρης ευθυγράμμιση των στοιχείων.
- **LazyVerticalStaggeredGrid:** Αντιθέτως, αυτή η διάταξη δίνει την δυνατότητα στα στοιχεία να μεταβάλλουν το ύψος, μήκος ή πλάτος, ανεξαρτήτως της τοποθέτησής τους. Ως αποτέλεσμα, δημιουργείται μια οπτικά ελκυστική διάταξη.
- **LazyHorizontalGrid:** Έχει ακριβώς την ίδια λειτουργία με το LazyVerticalGrid αλλά κατά μήκος του οριζοντίου άξονα. Για να χρησιμοποιηθεί πρέπει να οριστούν οι αριθμοί των σειρών (rows) 3
- **LazyHorizontalStaggeredGrid:** Ομοίως με το LazyHorizontalGrid τακτοποιεί τα πλέγματα οριζοντίως, με την διαφορά ότι το μήκος ή πλάτος των στηλών είναι μεταβλητό, με αποτέλεσμα να δημιουργεί ένα «κλιμακωτό» (=staggered) εφέ.

```

LazyVerticalStaggeredGrid(modifier = modifier.fillMaxSize(), columns =
StaggeredGridCells.Fixed(2), verticalItemSpacing = SpacingDefault_16dp,
horizontalArrangement = Arrangement.spacedBy(SpacingHalf_8dp), content = {
    items(marketUiState.eventUiItems.size) {
        EshopItem(modifier = Modifier, marketUiItem = marketUiState.eventUiItems[it],
addclick = {
            addToShoppingList(marketUiState.eventUiItems[it].description,
marketUiState.eventUiItems[it].imageUrl, marketUiState.eventUiItems[it].price)
        })
    }
})

```

Παράρτημα 3.11: Παράδειγμα LazyVerticalStaggeredGrid

### 3.7.10 Pager στο Jetpack Compose

Το Pager στο Jetpack Compose είναι ένα ακόμη στοιχείο διάταξης που επιτρέπει στους προγραμματιστές να προσθέσουν πολλές σελίδες (ViewPager)[28]. Για την μετάβαση μια σελίδας προς μια άλλη υπάρχει η δυνατότητα με την μετακίνηση του δαχτύλου προς τα δεξιά, αριστερά, πάνω ή κάτω, αναλόγως με τον τρόπο τον οποίο έχουν ρυθμιστεί οι σελίδες (πχ καρουζέλ εικόνων) . Αυτό το στοιχείο διάταξης αφενός διευκολύνει τον χρήστη, καθώς προσφέρει μια ομαλή εμπειρία.

Υπάρχουν δύο βασικοί τύποι ViewPager: Το HorizontalPager, και το VerticalPager. Ξεκινώντας από το πρώτο, το HorizontalPager αποτελεί το στοιχείο διάταξης που επιτρέπει επιτρέπει την οριζόντια κύλιση προς τα αριστερά και προς τα δεξιά, είτε από τον χρήστη είτε μέσω προγραμματισμού. Αντιθέτως το VerticalPager είναι το συνθετικό στοιχείο, το οποίο επιτρέπει την κατακόρυφη κύλιση προς τα πάνω και προς τα κάτω, είτε από το χρήστη είτε μέσω προγραμματισμού.

Σε αυτούς τους δύο τύπους υπάρχουν πολλά ορίσματα, που διαμορφώνουν την εμφάνιση των σελιδών. Ενδεικτικά:

- **pageCount:** Είναι το μόνο υποχρεωτικό όρισμα που πρέπει να τοποθετηθεί και χρησιμοποιείται για τον υπολογισμό συνολικού αριθμού σελίδων/αντικειμένων που θα εμφανίσει αυτή η σελίδα.
- **pageSize:** καθορίζει το μέγεθος μιας σελίδας μέσα στο Pager.
- **PageSpacing:** καθορίζει την απόσταση που πρέπει να υπάρχει ανάμεσα στις σελίδες
- **contentPadding:** καθορίζει τον τρόπο με τον οποίο οι σελίδες ευθυγραμμίζονται μέσα στο πρόγραμμα.
- **beyondBoundsPageCount:** καθορίζει πόσες σελίδες φορτώνονται πέρα από την τρέχουσα ορατή.
- **state:** Επιτρέπει την διαχείριση και την παρακολούθηση του ViewPager. Χρησιμοποιείται για τον έλεγχο των σελιδών.
- **flingBehavior:** ορίζει πώς πρέπει να συμπεριφέρονται οι χειρονομίες κύλισης.
- **reverseLayout:** αντιστρέφει τη διάταξη και την κατεύθυνση της κύλισης, με αποτέλεσμα η πρώτη σελίδα να εμφανίζεται σαν τελευταία και να χρειάζεται να γίνει κύλιση προς την αρχή της σελίδας, αντί προς το τέλος, όπως γίνεται κατά κανόνα.
- **key:** Είναι το μοναδικό κλειδί για το αντικείμενο, και μπορεί να χρησιμοποιηθεί για την διατήρηση της θέσης κύλισης ασχέτως εάν γίνει προσθήκη ή αφαίρεση στοιχείων.

```

HorizontalPager(pageCount = photoUiState.photosUiItem.size, state = pagerState)
{
    AsyncImage(
        model = photoUiState.photosUiItem[it].imageUrl,
        modifier = Modifier
            .padding(SpacingQuarter_4dp)
            .clip(RoundedCornerShape(SpacingHalf_8dp)),
        contentScale = ContentScale.Crop,
        contentDescription = ""
    )
}

```

Παράρτημα 3.12 : Παράδειγμα pager

### 3.7.11 Modifiers στο Jetpack Compose

Οι modifiers(=τροποποιητές) στο Jetpack Compose επιτρέπουν την τροποποίηση ενός στοιχείου διεπαφής χρήστη[29]. Αναλυτικότερα, οι τροποποιητές εφαρμόζονται στα στοιχεία διεπαφής χρήστη ώστε να αλλάξουν την εμφάνιση και συμπεριφορά τους, να προσθέσουν πληροφορίες (πχ ετικέτες προσβασιμότητας), όπως και σύνθετες αλληλεπιδράσεις (πχ να κάνουν ένα στοιχείο να έχει την δυνατότητα «κλικ», σύρσιμο, μεγένθυση κλπ) .Το Jetpack Compose παρέχει μια ποικιλία ενσωματωμένων Modifier που χρησιμοποιούνται ακριβώς για την εκτέλεση των προαναφερθεισών λειτουργιών, με τους πιο κοινούς να αφορούν:

- **Χώρος:** Προσαρμόζει τον χώρο σε ένα στοιχείο διεπαφής χρήστη.
- **Φόντο:** Προσθέτει ένα χρώμα φόντου σε ένα στοιχείο διεπαφής χρήστη.
- **Με δυνατότητα «κλικ» :** Δίνει την δυνατότητα κλικ σε ένα στοιχείο διεπαφής
- **Πλάτος:** Καθορίζει το πλάτος ενός στοιχείου διεπαφής χρήστη.
- **Ύψος:** Καθορίζει το ύψος του στοιχείου διεπαφής χρήστη.
- **Βάρος:** Καθορίζει το βάρος ενός στοιχείου διεπαφής χρήστη σε μια διάταξη Composable.
- **FillMaxWidth:** Κάνει ένα στοιχείο διεπαφής χρήστη να γεμίζει το διαθέσιμο πλάτος σε μια διάταξη Composable.

- **fillMaxHeight:** Κάνει ένα στοιχείο διεπαφής χρήστη να γεμίζει το διαθέσιμο ύψος σε μια διάταξη Composable.

### 3.7.12 States στο Jetpack Compose

Όπως αναφέραμε, το Jetpack Compose είναι μια μοντέρνα εργαλειοθήκη στοιχείων διεπαφής χρήστη με σκοπό την δημιουργία Android εφαρμογών[30]. Βασικό κομμάτι του Jetpack Compose είναι η διαχείριση της κατάστασης (State management) το οποίο δίνει την δυνατότητα στους προγραμματιστές να διαχειρίζονται τα δεδομένα με έναν αποτελεσματικό τρόπο. Ο,τιδήποτε σχετικό με τα States διεπαφής χρήστη, αναφέρονται δύο διαφορετικοί ορισμοί:

**Screen UI state:** Ουσιαστικά αφορά αυτά που χρειάζεται να εμφανιστούν στην οθόνη. Για παράδειγμα, εάν υπάρχει μια λίστα με δεδομένα API και παρουσιάζεται στον χρήστη, η λίστα αυτή γίνεται μέρος της κατάστασης διεπαφής χρήστη. Επομένως, εάν αλλάζουν τα δεδομένα της λίστας, πρέπει να ενημερώνεται και η οθόνη. Φυσικά, πίσω από αυτήν την λογική δεν υπάρχει κάποια σχετικότητα με τα στοιχεία διεπαφής χρήστη, γι' αυτόν τον λόγο ονομάζουμε την διαδικασία ως μια κατάσταση διεπαφής χρήστη. Στο Jetpack Compose το Screen UI state καθορίζεται από μια σειρά εφαρμογής επιχειρηματικών κανόνων και αναλαμβάνεται από ένα ViewModel, το οποίο συμβάλλει στην κεντρική διαχείριση αλλά και στην οργάνωση, καθώς ξεχωρίζει τα λειτουργικά κομμάτια της εφαρμογής Android.

**UI element state:** Η διαχείριση του στοιχείου διεπαφής χρήστη γίνεται εξωτερικά από το συνθέσιμο και μπορεί να μετακινηθεί στον καλούντα συνθέτη ή τον κάτοχο κατάστασης. Σε αντίθεση με το Android Views, στο οποίο η διαχείριση μιας κατάστασης γίνεται εσωτερικά, το Jetpack Compose θεωρεί μια κατάσταση διαφορετική από την σύνθεση. Το UI element state, δηλαδή η κατάσταση στοιχείου διεπαφής χρήστη αφορά τα χαρακτηριστικά των στοιχείων διεπαφής χρήστη σχετικά με την εμφάνιση και την συμπεριφορά. Για παράδειγμα, είναι δυνατή η απόκρυψη ή εμφάνιση ενός στοιχείου διεπαφής χρήστη, ο ορισμός συγκεκριμένης γραμματοσειράς, μεγέθους γραμματοσειράς ή χρώματος γραμματοσειράς. Η κατάσταση αποθηκεύεται χωριστά από το στοιχείο διεπαφής χρήστη και μπορεί να τοποθετηθεί σε έναν περιβάλλοντα κωδικό ή σε έναν συγκεκριμένο κάτοχο κατάστασης. Με αυτόν τον τρόπο αυξάνεται ο έλεγχος λειτουργείας και εμφάνισης στοιχείων διεπαφής χρήστη παρέχοντας μεγαλύτερη ευελιξία.

#### Τύποι State:

**MutableState:** Εκπροσωπεί μια τιμή που μπορεί να μεταβληθεί όσο περνάει ο καιρός. Η χρήση των αντικειμένων MutableState αφορά κυρίως την συντήρηση των μεταβλητών, που ο

τιμές τους ενδεχομένως να αλλάξουν και να και να ενεργοποιώντας με αυτόν τον τρόπο τον ανασχηματισμό της διεπαφής χρήστη όταν τροποποιηθούν.

**DerivedState:** Αυτή είναι μια κατάσταση η οποία είναι προϊόν άλλων καταστάσεων και δεδομένων εντός της συνθετικής συνάρτησης. Υπολογίζεται αναλόγως των τρέχουσων τιμών των άλλων καταστάσεων και δεν διατηρεί τη δική του τιμή άμεσα.<sup>5</sup>

**State Hoisting:** Ουσιαστικά, η κατάσταση ενός στοιχείου διεπαφής χρήστη μετακινείται σε έναν υψηλότερο επίπεδο, απλοποιώντας με αυτόν τον τρόπο το ίδιο το στοιχείο. Αυτό γίνεται με την αντικατάσταση της κατάστασης με δύο πράγματα: με την τιμή που πρέπει να εμφανιστεί και με τον τρόπο που θα αλλάξει την τιμή. Με την ανύψωση(State Hoisting), εξασφαλίζεται η μοναδικότητα της κατάστασης αλλά και η δυνατότητα να μοιραστεί σε πολλά διαφορετικά στοιχεία, διευκολύνοντας την διαχείριση της κατάστασης.

### 3.8 Βιβλιοθήκες (Libraries)

Μια βιβλιοθήκη είναι ένα σύνολο κώδικα που έχει γραφεί σε προγενέστερο χρόνο και οι προγραμματιστές μπορούν να αξιοποιήσουν για να εξορθολογίσουν τις εργασίες τους και να βελτιώσουν την αποτελεσματικότητά τους. Αυτός ο επαναχρησιμοποιήσιμος κώδικας έχει σχεδιαστεί συχνά για να αντιμετωπίζει συγκεκριμένα κοινά προβλήματα, καθιστώντας ευκολότερη την εφαρμογή τυπικής λειτουργικότητας σε διάφορα έργα. Μια βιβλιοθήκη αποτελείται συνήθως από πολλαπλά προ-κωδικοποιημένα στοιχεία που προσφέρουν λύσεις σε διαφορετικές πτυχές ενός δεδομένου προβλήματος ή λειτουργικότητας.

Οι προγραμματιστές χρησιμοποιούν βιβλιοθήκες για να επιταχύνουν την ανάπτυξη εφαρμογών και ιστότοπων, καθώς κάθε βιβλιοθήκη έχει σχεδιαστεί για να προσφέρει λύσεις για συγκεκριμένες λειτουργίες όπως έλεγχος ταυτότητας χρήστη, επικοινωνία διακομιστή, διεπαφές χρήστη, διαχείριση δεδομένων, αλγόριθμοι, κινούμενα σχέδια και άλλα. Όταν εργάζονται σε μια συγκεκριμένη δυνατότητα, οι προγραμματιστές συχνά αναζητούν βιβλιοθήκες που μπορούν να τους βοηθήσουν να δημιουργήσουν το στοιχείο πιο αποτελεσματικά ή να λύσουν ένα δύσκολο πρόβλημα. Στη συνέχεια, επιλέγουν τα σχετικά στοιχεία από μια ενιαία βιβλιοθήκη για να διασφαλίσουν ότι η εφαρμογή τους διατηρεί τη συνέπεια και τη συνοχή.

Επιπλέον, οι βιβλιοθήκες προσφέρουν στους προγραμματιστές μια νέα προοπτική για τη δουλειά τους. Καθώς κάθε κωδικοποιητής έχει τη δική του προσέγγιση, η αναφορά σε βιβλιοθήκες μπορεί να παρέχει πληροφορίες για διαφορετικές μεθόδους κωδικοποίησης και να εμπνεύσει εναλλακτικούς τρόπους αντιμετώπισης ενός προβλήματος.

### 3.8.1 Material Design

Το Google Material, γνωστό και ως Material Design, είναι ένα σύστημα σχεδιασμού που αναπτύχθηκε από την Google[31]. Προσφέρει ολοκληρωμένες οδηγίες για τη δημιουργία ψηφιακών εμπειριών, συμπεριλαμβανομένων διεπαφών χρήστη για εφαρμογές ιστού και για κινητές συσκευές. Το Material Design προσπαθεί να παρέχει μια συνεπή και διαισθητική εμπειρία χρήστη σε διαφορετικές συσκευές και πλατφόρμες. Τα βασικά στοιχεία του Υλικού Σχεδιασμού περιλαμβάνουν:

- **Visual Hierarchy:** Επικεντρώνεται σε σαφείς, οργανωμένες διατάξεις για να βοηθά τους χρήστες να πλοηγούνται εύκολα στις διεπαφές και να τις κατανοούν γρήγορα.
- **Τυπογραφία:** Το σύστημα σχεδίασης περιγράφει συγκεκριμένες κατευθυντήριες γραμμές για τα στυλ γραμματοσειράς, τα μεγέθη και τα κενά για να διασφαλίσει την αναγνωσιμότητα και την αισθητική.
- **Παλέτα χρωμάτων:** Ο σχεδιασμός υλικού παρέχει μια επιλογή από πρωτεύοντα και δευτερεύοντα χρώματα για τη δημιουργία οπτικά συνεκτικών διεπαφών.
- **Στοιχεία:** Διαθέτει μια βιβλιοθήκη στοιχείων διεπαφής χρήστη, όπως κουμπιά, κάρτες και παράθυρα διαλόγου, σύμφωνα με τις αρχές του Material Design.
- **Κίνηση και κινούμενα σχέδια:** Ο σχεδιασμός υλικού ενσωματώνει ομαλές μεταβάσεις και κινούμενα σχέδια για τη βελτίωση της αλληλεπίδρασης των χρηστών και την παροχή σχολίων.
- **Ανύψωση και σκιές:** Το σύστημα χρησιμοποιεί εφέ βάθους και σκιάς για να επικοινωνήσει την ιεραρχία και να προσθέσει μια αίσθηση χώρου.

Η Google προσφέρει Material Design για διάφορες πλατφόρμες όπως Android, iOS και web. Ακολουθώντας τις οδηγίες σχεδιασμού υλικού, οι προγραμματιστές μπορούν να δημιουργήσουν οπτικά συνεπείς και φιλικές προς το χρήστη διεπαφές που ευθυγραμμίζονται με τα πρότυπα σχεδιασμού της Google.

Μετά το 2018, η Google άρχισε να ανανεώνει τις περισσότερες από τις εφαρμογές της ακολουθώντας ένα ενημερωμένο σύνολο αρχών και κατευθυντήριων γραμμών γνωστών ως "Material Design 2". Αυτή η επανάληψη δίνει μεγαλύτερη έμφαση στην προσαρμογή των βασικών στοιχείων Σχεδιασμού Υλικού ώστε να ταιριάζουν καλύτερα με την επωνυμία των προϊόντων στα οποία χρησιμοποιούνται. Οι ενημερωμένες οδηγίες υπογραμμίζουν τη σημασία του λευκού χώρου, των στρογγυλεμένων γωνιών, των πολύχρωμων εικονιδίων και των κάτω γραμμών πλοήγησης.

Στο Google I/O τον Μάιο του 2021, η Google εισήγαγε μια νέα προσέγγιση σχεδίασης στο Android 12 που ονομάζεται "Material You" (επίσης γνωστή ως "Material Design 3"), η οποία δίνει προτεραιότητα σε περισσότερα κινούμενα σχέδια, μεγαλύτερα κουμπιά και τη δυνατότητα δημιουργίας προσαρμοσμένων θεμάτων διεπαφής χρήστη με βάση την ταπετσαρία του χρήστη. Το Material You εφαρμόστηκε σταδιακά σε διάφορες εφαρμογές Google σε παλαιότερες εκδόσεις Android τους επόμενους μήνες και έγινε κεντρικό χαρακτηριστικό των σειρών smartphone Pixel 6 και Pixel 6 Pro.

### 3.8.2 Material Ui

Το Material UI είναι μια βιβλιοθήκη στοιχείων React ανοιχτού κώδικα που ζωντανεύει το Material Design της Google. Προσφέρει ένα εκτεταμένο σύνολο προκατασκευασμένων εξαρτημάτων έτοιμα για άμεση χρήση στην παραγωγή και παρέχει μια σειρά από επιλογές προσαρμογής που επιτρέπουν στο προγραμματιστή να "στρώσει" το προσαρμοσμένο σύστημα σχεδίασης πάνω από τα εξαρτήματά του. Το Material UI υποστηρίζει Material Design 2.

Τα προτερήματα του Material UI είναι:

- Γρηγορότερη υλοποίηση: Με περισσότερους από 2.500 συνεισφέροντες ανοιχτού κώδικα που αφιέρωσαν τον χρόνο τους σε αυτά τα στοιχεία, ο προγραμματιστής μπορεί να συγκεντρωθεί στη βασική επιχειρηματική λογική χωρίς να χρειάζεται να αναδημιουργήσει στοιχεία διεπαφής χρήστη από την αρχή. Το Material UI καλύπτει αυτές τις ανάγκες διεπαφής.
- Όμορφο από προεπιλογή: Το Material UI εφαρμόζει σχολαστικά το Material Design με τα υψηλότερα πρότυπα αισθητικής και χρηστικότητας. Ενώ τηρεί πιστά τις προδιαγραφές σχεδιασμού, αποκλίνει επίσης όπου είναι απαραίτητο για να προσφέρει πολλαπλές επιλογές υψηλής ποιότητας.
- Προσαρμοστικότητα: Η διεπαφή χρήστη Material UI παρέχει ένα ευρύ φάσμα δυνατοτήτων προσαρμογής που επιτρέπουν την προσαρμογή στοιχείων στις ανάγκες του εκάστοτε προγραμματιστή.
- Συνεργασία μεταξύ ομάδων: Η φιλική προς το χρήστη εμπειρία προγραμματιστή του Material UI μειώνει την καμπύλη εκμάθησης για προγραμματιστές και σχεδιαστές back-end, επιτρέποντας καλύτερη ομαδική εργασία. Τα kit σχεδίασης εξορθολογίζουν τις ροές εργασίας και ενισχύουν τη συνέπεια μεταξύ των ομάδων.
- Αξιόπιστο από χιλιάδες οργανισμούς: Το Material UI διαθέτει τη μεγαλύτερη κοινότητα διεπαφής χρήστη στο οικοσύστημα React, με ιστορία που χρονολογείται από το 2014. Η

μακροζωία και η υποστήριξη της κοινότητας, συμπεριλαμβανομένων πόρων όπως το Stack Overflow, διασφαλίζουν ένα αξιόπιστο δίκτυο.

### 3.8.3 Timber

Ένα πολύ γνωστό πρόγραμμα καταγραφής για εφαρμογές Android είναι η βιβλιοθήκη Timber, η οποία ξεχωρίζει μεταξύ άλλων, καθώς παρέχει έναν όχι μόνο εύκολο αλλά και αποτελεσματικό τρόπο καταγραφής μηνυμάτων σε έναν κώδικα μιας εφαρμογής. Ο εντοπισμός σφαλμάτων κατά την ανάπτυξη μιας εφαρμογής γίνεται πιο εύκολα, κάνοντας το οποιοδήποτε πρόβλημα να αντιμετωπίζεται γρήγορα. Ένα από τα κυριότερα οφέλη της βιβλιοθήκης αυτής, αφορά την προσθήκη ετικετών καθώς προσθέτει αυτομάτως πληροφορίες κλάσης και μεθόδου σε δηλώσεις καταγραφής, ώστε να συμβάλει στον εντοπισμό προέλευσης ενός μηνύματος καταγραφής. Επιπλέον, η συμπεριφορά καταγραφής προσαρμίζεται με εύκολο τρόπο αναλόγως τα διαφορετικά σενάρια στα οποία μπορεί να υπάρξει. Για παράδειγμα, υπάρχει δυνατότητα ενεργοποίησης/ απενεργοποίησης συνδέσεων σε κάποιες εκδόσεις της εφαρμογής ή καταγραφής σφαλμάτων στα Crashlytics για εκδόσεις παραγωγής. Είναι σημαντικό να αναφερθεί πως η καταγραφή κατηγοριοποιείται σε ποικίλα επίπεδα, που δείχνουν την σοβαρότητα ενός καταγεγραμμένου μηνύματος. Συγκεκριμένα:

**VERBOSE:** Με χαμηλή προτεραιότητα, συνήθως χρησιμοποιείται για έναν ακριβή εντοπισμό σφαλμάτων.

**DEBUG:** Η χρήση του αφορά τον εντοπισμό πληροφοριών αλλά και σφαλμάτων.

**INFO:** Δίνει πληροφορίες αναφορικά με το πώς λειτουργεί η εφαρμογή.

**WARN:** Παρέχει πιθανά ζητήματα τα οποία ενδεχομένως να προκαλέσουν πρόβλημα.

**ERROR:** Αφορά την καταγραφή λαθών που ίσως δημιουργήσουν δυσλειτουργία στην εφαρμογή.

**ASSERT:** Καταγράφει κρίσιμα θέματα τα οποία δεν πρέπει ποτέ να προκύψουν.

### 3.8.4 AndroidX Navigation

Ένα ακόμα κομμάτι της βιβλιοθήκης Android Jetpack το οποίο συμβάλλει στην διαμόρφωση της πλοήγησης μιας εφαρμογής είναι το AndroidX Navigation, το οποίο δημιουργεί ένα πλαίσιο για πλοήγηση με έναν προσεκτικό και συνεπή τρόπο. Το AndroidX Navigation παρέχει την δυνατότητα ορισμού γραφήματος πλοήγησης σε μια εφαρμογή, που παρουσιάζει σε οπτικό επίπεδο την ροή των οθονών καθώς και τις ενδεχόμενες διαδρομές που μπορεί να υπάρξουν μεταξύ τους. Με αυτόν τον τρόπο, η μεταφορά δεδομένων μεταξύ προορισμών, ο έλεγχος συνδέσμων εις βάθος και η εξασφάλιση της σωστής διαχείρισης στοίβας αλλά και η γενικότερα η διαχείριση όλων των εργασιών που αφορούν την πλοήγηση γίνονται πιο εύκολες. Με απλά λόγια, το AndroidX Navigation προσφέρει έναν πιο απλό τρόπο

εφαρμογής μοτίβων πλοήγησης στις εφαρμογές Android, αυξάνοντας με αυτόν τον τρόπο την αποδοτικότητα τους.

### 3.8.5 Google Maps Android API

Το Google Maps Android API είναι μια βιβλιοθήκη που επιτρέπει στους προγραμματιστές να ενσωματώνουν τους Χάρτες Google στις εφαρμογές τους Android, παρέχοντας πλούσιες και διαδραστικές δυνατότητες χαρτογράφησης[32]. Ως μέρος των υπηρεσιών Google Play, αυτή η βιβλιοθήκη προσφέρει μια σειρά από λειτουργίες που μπορούν να χρησιμοποιηθούν για τη βελτίωση εφαρμογών με λειτουργικότητα που βασίζεται σε χάρτες.

Ένα από τα κύρια χαρακτηριστικά του API, που επιτρέπει στους προγραμματιστές να ενσωματώσουν ένα `MapView` στη διάταξη μιας εφαρμογής για να εμφανίζουν διαφορετικούς τύπους χαρτών, όπως τυπικούς, δορυφορικούς, εδάφους και υβριδικούς χάρτες είναι το `Map Display`. Οι χρήστες μπορούν να αλληλοεπιδράσουν με αυτούς τους χάρτες μέσω μετατόπισης, μεγέθυνσης και περιστροφής της προβολής.

Ένα άλλο βασικό χαρακτηριστικό του API είναι οι δείκτες και οι επικαλύψεις, οι οποίες επιτρέπουν στους προγραμματιστές να προσθέτουν διάφορα στοιχεία στον χάρτη, όπως δείκτες, πολύγραμμες, πολύγωνα, κύκλους και επικαλύψεις εδάφους για να αντιπροσωπεύουν δεδομένα ή σημεία ενδιαφέροντος. Αυτό μπορεί να είναι ιδιαίτερα χρήσιμο για εφαρμογές που απαιτούν οπτική αναπαράσταση τοποθεσιών ή μονοπατιών.

Το API υποστηρίζει δυνατότητες γεωκωδικοποίησης και αντίστροφης γεωκωδικοποίησης, οι οποίες επιτρέπουν στις εφαρμογές να μετατρέπουν διευθύνσεις σε γεωγραφικές συντεταγμένες και αντίστροφα. Αυτή η δυνατότητα είναι πολύτιμη για εφαρμογές που βασίζονται σε τοποθεσία που πρέπει να μετατρέψουν τα δεδομένα ή τα δεδομένα χρήστη σε χρήσιμες πληροφορίες χάρτη.

Για εφαρμογές που σχετίζονται με την πλοήγηση, το API μπορεί να παρέχει οδηγίες και διαδρομές μεταξύ δύο ή περισσότερων σημείων στον χάρτη, προσφέροντας καθοδήγηση πλοήγησης και επισημάνσεις διαδρομής. Επιπλέον, το API μπορεί να εμφανίσει την τρέχουσα τοποθεσία του χρήστη στον χάρτη, επιτρέποντας στις εφαρμογές να παρακολουθούν την κίνηση σε πραγματικό χρόνο.

Η βιβλιοθήκη προσφέρει επίσης διάφορες επιλογές για στυλ και προσαρμογή. Οι προγραμματιστές μπορούν να προσαρμόσουν την εμφάνιση του χάρτη χρησιμοποιώντας διαφορετικά στυλ χάρτη και να ελέγξουν την εμφάνιση χαρακτηριστικών όπως ετικέτες και σημεία ενδιαφέροντος. Αυτό το επίπεδο προσαρμογής επιτρέπει μια πιο προσαρμοσμένη εμπειρία χρήστη και επωνυμία.

Για να χρησιμοποιηθεί το Google Maps Android API, οι προγραμματιστές πρέπει να αποκτήσουν ένα κλειδί API και να συμπεριλάβουν τις απαραίτητες εξαρτήσεις στο έργο τους. Συνολικά, η βιβλιοθήκη απλοποιεί τη διαδικασία προσθήκης προηγμένων χαρακτηριστικών χαρτογράφησης σε εφαρμογές Android και παρέχει πρόσβαση στα εκτεταμένα δεδομένα και υπηρεσίες χαρτογράφησης της Google. Αυτό επιτρέπει στους προγραμματιστές να δημιουργούν πιο δυναμικές, διαδραστικές και βασισμένες στην τοποθεσία εφαρμογές που βελτιώνουν την εμπειρία και την αφοσίωση των χρηστών.

### 3.8.6 Accompanist

Το Accompanist είναι μια συλλογή βιβλιοθηκών που έχουν σχεδιαστεί για να συμπληρώνουν το Jetpack Compose παρέχοντας χαρακτηριστικά και λειτουργικότητα που συνήθως απαιτούνται από προγραμματιστές που δεν περιλαμβάνονται ακόμη στις βασικές βιβλιοθήκες Jetpack Compose AndroidX. Δημιουργημένο και συντηρημένο από την ομάδα Jetpack Compose της Google, το Accompanist λειτουργεί ως πειραματικός χώρος για τη δοκιμή νέων Compose API και τη συλλογή σχολίων από προγραμματιστές για να καθοδηγήσει τη μελλοντική ανάπτυξη του Jetpack Compose.

Οι βιβλιοθήκες στο Accompanist στοχεύουν να γεφυρώσουν το χάσμα μεταξύ του Jetpack Compose και των υπάρχοντων API ή βιβλιοθηκών Android, προσφέροντας στους προγραμματιστές έναν τρόπο να ενσωματώσουν τις συμβατές με το Compose εκδόσεις κοινών εργαλείων και στοιχείων στα έργα τους. Ο στόχος του Accompanist είναι τελικά να μεταφέρει τις βιβλιοθήκες του στο επίσημο κιτ εργαλείων Jetpack Compose μόλις φτάσουν σε ωριμότητα και σταθερότητα. Μόλις μεταφερθούν στο upstream, οι βιβλιοθήκες θα καταργηθούν και θα αφαιρεθούν από το Accompanist.

Το Accompanist προσφέρει ένα ευρύ φάσμα βιβλιοθηκών για διάφορες περιπτώσεις χρήσης, συμπεριλαμβανομένων πειραματικών και καινοτόμων λειτουργιών που μπορεί να μην είναι ακόμη σταθερές, αλλά επιτρέπουν στους προγραμματιστές να εξερευνήσουν νέους τρόπους δημιουργίας στοιχείων και λειτουργιών διεπαφής χρήστη στο Compose. Αυτές οι βιβλιοθήκες περιλαμβάνουν Διατάξεις ροής, Pager, Navigation Transitions, Swipe Refresh και άλλα. Επιπλέον, ορισμένες βιβλιοθήκες Accompanist χρησιμεύουν ως γέφυρα μεταξύ του Jetpack Compose και των υπάρχοντων API Android, παρέχοντας εκδόσεις κοινών εργαλείων και στοιχείων συμβατές με το Compose.

Ως έργο ανοιχτού κώδικα, το Accompanist προσκαλεί κοινοτικές συνεισφορές και προτάσεις για βελτιώσεις. Οι προγραμματιστές μπορούν να συμπεριλάβουν τις βιβλιοθήκες Accompanist στα έργα τους ως εξαρτήσεις, όπως και κάθε άλλη βιβλιοθήκη. Ωστόσο, καθώς το Accompanist περιέχει πειραματικές δυνατότητες, οι προγραμματιστές θα πρέπει να

παρακολουθούν τις ενημερώσεις και τις αλλαγές για να διασφαλίσουν τη συμβατότητα με τα έργα τους.

Συμπερασματικά, το Accompanist διαδραματίζει κρίσιμο ρόλο στην εξέλιξη του Jetpack Compose δίνοντας τη δυνατότητα στους προγραμματιστές να έχουν πρόσβαση σε νέες δυνατότητες και πειραματικά API, να συλλέγουν σχόλια και να συμβάλλουν στη συνεχή βελτίωση του οικοσυστήματος Compose. Καθώς το έργο προχωρά, το Accompanist στοχεύει να ενσωματώσει τις προσφορές του στην επίσημη εργαλειοθήκη, ανοίγοντας το δρόμο για μια πιο ισχυρή και ολοκληρωμένη βιβλιοθήκη Jetpack Compose.

### 3.8.7 Accompanist Map Compose

Για να ενσωματώσουν τους Χάρτες Google με τις εφαρμογές Jetpack Compose, οι προγραμματιστές χρησιμοποιούν τη βιβλιοθήκη Accompanist Maps Compose. Αυτή η βιβλιοθήκη χρησιμεύει ως γέφυρα μεταξύ του API των Χαρτών Google Android και του Jetpack Compose, παρέχοντας έναν απλό τρόπο ενσωμάτωσης χαρτών σε στοιχεία Compose. Το Accompanist Maps Compose με τους Χάρτες Google χρησιμοποιείται με τους εξής τρόπους:

Πρώτες ρυθμίσεις: Για να ξεκινήσουν οι προγραμματιστές πρέπει να συμπεριλάβουν στο έργο τους τα απαραίτητα dependencies για το Accompanist Maps Compose. Επιπλέον, πρέπει να αποκτήσουν ένα κλειδί API από την Google για τη χρήση των υπηρεσιών των Χαρτών Google.

Δημιουργία MapView: Το Accompanist Maps Compose προσφέρει μια δυνατότητα σύνθεσης MapView που μπορεί να ενσωματωθεί απευθείας στη διάταξη διεπαφής χρήστη. Αυτό επιτρέπει στους προγραμματιστές να ενσωματώνουν απρόσκοπτα τους Χάρτες Google στις εφαρμογές τους Compose.

Προσαρμογή του χάρτη: Οι προγραμματιστές μπορούν να προσαρμόσουν την εμφάνιση και τη συμπεριφορά του χάρτη καθορίζοντας ιδιότητες όπως τον τύπο χάρτη, την αρχική θέση της κάμερας και το στυλ χάρτη. Το Accompanist Maps Compose παρέχει μια σειρά επιλογών για την προσαρμογή της εμφάνισης και της αίσθησης του χάρτη σύμφωνα με τις ανάγκες σχεδίασης και λειτουργικότητας της εφαρμογής.

Χειρισμός αλληλεπιδράσεων χαρτών: Το Accompanist Maps Compose επιτρέπει στους προγραμματιστές να χειρίζονται διάφορες αλληλεπιδράσεις χαρτών, συμπεριλαμβανομένης της μεγέθυνσης, της μετατόπισης και του αγγίγματος στον χάρτη. Οι ακροατές συμβάντων μπορούν να συνδεθούν για να ανταποκριθούν σε αυτές τις αλληλεπιδράσεις, όπως η εκτέλεση ενεργειών όταν ένας χρήστης κάνει κλικ σε μια συγκεκριμένη περιοχή του χάρτη.

**Προσθήκη στοιχείων χάρτη:** Η βιβλιοθήκη υποστηρίζει την προσθήκη δεικτών, πολύγραμμών, πολυγώνων και άλλων στοιχείων χάρτη στον χάρτη. Αυτά μπορούν να χρησιμοποιηθούν για την αναπαράσταση τοποθεσιών, μονοπατιών ή περιοχών ενδιαφέροντος εντός της εφαρμογής.

**Εργασία με Ζωντανά Δεδομένα:** Το `Accompanist Maps Compose` ενσωματώνεται ομαλά με τις λειτουργίες διαχείρισης κατάστασης και ζωντανών δεδομένων του `Jetpack Compose`. Οι προγραμματιστές μπορούν να ενημερώνουν τον χάρτη δυναμικά με βάση τις αλλαγές στα δεδομένα, όπως η μετακίνηση δεικτών ή η προσαρμογή των προβολών χάρτη ως απόκριση στις ενέργειες των χρηστών.

**Χειρισμός συμβάντων:** Οι προγραμματιστές μπορούν να ακούν και να χειρίζονται συμβάντα από τους Χάρτες Google, όπως κλικ δεικτών ή κλικ σε χάρτη, χρησιμοποιώντας προγράμματα ακρόασης συμβάντων στο `MapView composable`.

**Συνδυασμός με άλλα στοιχεία Σύνθεσης:** Το `Accompanist Maps Compose` μπορεί να χρησιμοποιηθεί σε συνδυασμό με άλλα στοιχεία `Compose` στην εφαρμογή, επιτρέποντας πλούσιες και διαδραστικές εμπειρίες χρήστη που συνδυάζουν χάρτες με άλλα στοιχεία διεπαφής χρήστη.

Χρησιμοποιώντας το `Accompanist Maps Compose`, οι προγραμματιστές μπορούν να ενσωματώσουν απρόσκοπτα τους Χάρτες Google στις εφαρμογές τους `Jetpack Compose`, ενώ εκμεταλλεύονται το δηλωτικό, αντιδραστικό μοντέλο προγραμματισμού του `Compose`. Αυτός ο συνδυασμός παρέχει έναν σύγχρονο και αποτελεσματικό τρόπο δημιουργίας χαρακτηριστικών και διεπαφών χρήστη που βασίζονται σε χάρτες.

### 3.8.10 Compose UI

Το `Compose UI` είναι μια σύγχρονη, δηλωτική εργαλειοθήκη διεπαφής χρήστη που επιτρέπει στους προγραμματιστές να δημιουργούν πλούσιες και διαδραστικές διεπαφές χρήστη σε εφαρμογές Android χρησιμοποιώντας το `Jetpack Compose`. Αντιπροσωπεύει μια μετατόπιση από τα παραδοσιακά μοντέλα επιτακτικού προγραμματισμού σε ένα δηλωτικό παράδειγμα, εκσυγχρονίζοντας την ανάπτυξη διεπαφής χρήστη και διευκολύνοντας τη δημιουργία πολύπλοκων, δυναμικών και αποκρινόμενων εφαρμογών. Το `Compose UI` απλοποιεί τη διαδικασία δημιουργίας και ενημέρωσης διεπαφής χρήστη, επιτρέποντας στους προγραμματιστές να περιγράφουν πώς θα πρέπει να είναι το UI σε κάθε δεδομένη κατάσταση και η εργαλειοθήκη φροντίζει για την υποκείμενη απόδοση και τις αλληλεπιδράσεις. Για τους προγραμματιστές τα προτερήματα ποικίλουν

Το `Compose UI` προσφέρει πολλά προτερήματα για τους προγραμματιστές του. Κάποια από αυτά είναι:

**Απλοποιημένος κώδικας:** Η διεπαφή χρήστη Compose απλοποιεί τη βάση κώδικα χρησιμοποιώντας μια δηλωτική προσέγγιση, η οποία επιτρέπει στους προγραμματιστές να εστιάζουν στην περιγραφή της επιθυμητής κατάστασης της διεπαφής χρήστη αντί να διαχειρίζονται την τρέχουσα κατάστασή της. Αυτό έχει ως αποτέλεσμα καθαρότερο και πιο ευανάγνωστο κώδικα.

**Λιγότερο Boilerplate:** Ο παραδοσιακός προγραμματισμός διεπαφής χρήστη συχνά περιλαμβάνει πολύ κώδικα boilerplate για δέσμευση προβολής και διαχείριση κατάστασης. Το Compose μειώνει αυτό παρέχοντας άμεση πρόσβαση στις δυνατότητες διαχείρισης κατάστασης και εξαλείφοντας την ανάγκη για διατάξεις που βασίζονται σε XML.

**Αντιδραστικός προγραμματισμός:** Το Compose χρησιμοποιεί ένα αντιδραστικό μοντέλο προγραμματισμού που ενημερώνει αυτόματα τη διεπαφή χρήστη όταν αλλάζει η κατάσταση. Αυτό μειώνει την ανάγκη για μη αυτόματες ενημερώσεις διεπαφής χρήστη και συγχρονισμό.

**Επαναχρησιμοποιήσιμα στοιχεία:** Οι προγραμματιστές μπορούν να δημιουργήσουν επαναχρησιμοποιήσιμες συναρτήσεις για στοιχεία διεπαφής χρήστη. Αυτά τα στοιχεία μπορούν εύκολα να επαναχρησιμοποιηθούν σε διάφορα μέρη μιας εφαρμογής, βελτιώνοντας την αρθρωτή δομή κώδικα και μειώνοντας την αντιγραφή.

**Απρόσκοπτη κινούμενη εικόνα:** Το Compose περιλαμβάνει ενσωματωμένη υποστήριξη για κινούμενα σχέδια, μεταβάσεις και εφέ κίνησης, καθιστώντας εύκολη τη δημιουργία οπτικά ελκυστικών και ελκυστικών διεπαφής χρήστη.

**Διαλειτουργικότητα:** Ενώ το Compose αντιπροσωπεύει έναν νέο τρόπο δημιουργίας διεπαφής χρήστη, εξακολουθεί να μπορεί να διαλειτουργεί με τις υπάρχουσες προβολές Android και άλλα πλαίσια διεπαφής χρήστη. Αυτό επιτρέπει στους προγραμματιστές να μετεγκαταστήσουν σταδιακά υπάρχοντα έργα στο Compose.

**Υποστήριξη εργαλείων:** Το Jetpack Compose έχει ισχυρή υποστήριξη από εργαλεία ανάπτυξης Android όπως το Android Studio, συμπεριλαμβανομένων των ζωντανών προεπισκοπήσεων και ενός ενσωματωμένου επιθεωρητή διάταξης

Παρά τα πλεονεκτήματά του, το Compose UI παρουσιάζει επίσης ορισμένες προκλήσεις:

**Καμπύλη εκμάθησης:** Για προγραμματιστές που είναι συνηθισμένοι στην παραδοσιακή ανάπτυξη Android, η μετάβαση σε ένα δηλωτικό παράδειγμα μπορεί να πάρει χρόνο για να μάθουν και να προσαρμοστούν.

**Σταθερότητα API:** Το Compose είναι μια σχετικά νέα τεχνολογία και τα API του ενδέχεται να αλλάξουν καθώς ωριμάζει. Οι προγραμματιστές πρέπει να γνωρίζουν πιθανές σημαντικές αλλαγές στις νέες εκδόσεις.

**Θέματα απόδοσης:** Ενώ το Compose γενικά προσφέρει καλή απόδοση, οι προγραμματιστές πρέπει να προσέχουν την ανασύνθεση και να αποφεύγουν τις υπερβολικές ανασυνθέσεις που μπορούν να επηρεάσουν την απόδοση της εφαρμογής.

**Συμβατότητα βιβλιοθήκης:** Ορισμένες υπάρχουσες βιβλιοθήκες Android ενδέχεται να μην είναι ακόμη πλήρως συμβατές με το Compose, απαιτώντας από τους προγραμματιστές να βρουν λύσεις ή να χρησιμοποιήσουν βιβλιοθήκες Accompanist για λειτουργίες που λείπουν.

**Εντοπισμός σφαλμάτων (Debugging):** Οι εφαρμογές Σύνταξης μπορεί να διαφέρουν από τις παραδοσιακές εφαρμογές Android λόγω της αντιδραστικής φύσης τους και της χρήσης απόδοσης βάσει κατάστασης.

Η εφαρμογή του Compose UI στον κώδικα ενός προγραμματιστή περιλαμβάνει τα ακόλουθα βήματα:

1. **Ρύθμιση του έργου:** Οι προγραμματιστές θα πρέπει να διασφαλίσουν ότι διαθέτουν την πιο πρόσφατη έκδοση του Android Studio και να ρυθμίσουν το έργο τους ώστε να χρησιμοποιούν το Jetpack Compose. Αυτό περιλαμβάνει την ενημέρωση της διαμόρφωσης Gradle ώστε να περιλαμβάνει τις απαραίτητες εξαρτήσεις Compose.
2. **Δημιουργία Composables:** Τα θεμελιώδη δομικά στοιχεία του Compose UI είναι συναρτήσεις με δυνατότητα σύνθεσης, οι οποίες καθορίζουν τα στοιχεία του UI. Οι προγραμματιστές γράφουν αυτές τις λειτουργίες για να δημιουργήσουν διαφορετικά μέρη της διεπαφής χρήστη, χρησιμοποιώντας το DSL (Γλώσσα συγκεκριμένης περιοχής) του Compose για τη δημιουργία διατάξεων.
3. **Διαχείριση κατάστασης:** Οι δυνατότητες διαχείρισης κατάστασης του Compose επιτρέπουν στους προγραμματιστές να διαχειρίζονται την κατάσταση διεπαφής χρήστη με δομημένο τρόπο. Μπορούν να χρησιμοποιήσουν αντικείμενα State ή MutableState για να παρακολουθούν την κατάσταση της διεπαφής χρήστη και να ενημερώνουν αυτόματα τη διεπαφή χρήστη όταν αλλάζει η κατάσταση.
4. **Πλοήγηση και δρομολόγηση:** Για εφαρμογές με πολλές οθόνες, οι προγραμματιστές μπορούν να χρησιμοποιήσουν τις βιβλιοθήκες πλοήγησης του Compose για να χειριστούν την πλοήγηση μεταξύ διαφορετικών οθονών. Αυτό απλοποιεί τη λογική πλοήγησης και υποστηρίζει τη χρήση συνθετικών λειτουργιών για οθόνες.
5. **Χειρισμός συμβάντων:** Ο χειρισμός συμβάντων στο Compose περιλαμβάνει τη διαβίβαση χειριστών συμβάντων ως παραμέτρων σε συναρτήσεις που μπορούν να συνθέτουν. Αυτό επιτρέπει στους προγραμματιστές να χειρίζονται τις αλληλεπιδράσεις των χρηστών και να ανταποκρίνονται σε συμβάντα διεπαφής χρήστη με απλό τρόπο.
6. **Διαλειτουργικότητα με υπάρχον περιβάλλον χρήστη:** Το Compose μπορεί να συνυπάρχει με υπάρχοντα στοιχεία διεπαφής χρήστη Android, επιτρέποντας στους

προγραμματιστές να ενσωματώνουν στοιχεία Compose σε παραδοσιακές εφαρμογές Android και αντίστροφα. Αυτό μπορεί να είναι χρήσιμο κατά τη διαδικασία μετεγκατάστασης ή κατά την εργασία με υπάρχουσες βάσεις κωδικών.

7. **Χρήση βιβλιοθηκών Accompanist:** Οι προγραμματιστές μπορούν να χρησιμοποιήσουν τις βιβλιοθήκες Accompanist για να αποκτήσουν πρόσβαση σε λειτουργίες που δεν αποτελούν ακόμη μέρος της βασικής βιβλιοθήκης Jetpack Compose, όπως το Swipe Refresh, οι μεταβάσεις πλοήγησης και η ενσωμάτωση με τους Χάρτες Google.
8. **Προεπισκόπηση και δοκιμή:** Η δυνατότητα ζωντανής προεπισκόπησης του Android Studio επιτρέπει στους προγραμματιστές να βλέπουν τη διεπαφή χρήστη σε πραγματικό χρόνο καθώς κωδικοποιούν. Οι προγραμματιστές θα πρέπει επίσης να γράφουν δοκιμές για τις συνθέσιμες λειτουργίες τους για να διασφαλίσουν ότι συμπεριφέρονται όπως αναμένεται.

Εν κατακλείδι, το Compose UI προσφέρει μια φρέσκια και σύγχρονη προσέγγιση στην ανάπτυξη διεπαφής χρήστη σε εφαρμογές Android. Απλοποιεί τον κώδικα, ενθαρρύνει ένα αντιδραστικό μοντέλο προγραμματισμού και υποστηρίζει τη δημιουργία δυναμικών και ελκυστικών διεπαφών χρήστη. Αν και υπάρχουν προκλήσεις και μια καμπύλη μάθησης, τα πλεονεκτήματα του Compose το καθιστούν μια πολύτιμη εργαλειοθήκη για προγραμματιστές που στοχεύουν να δημιουργήσουν εφαρμογές Android αιχμής. Με συνεχείς βελτιώσεις και ένα αυξανόμενο οικοσύστημα βιβλιοθηκών, το Compose UI αντιπροσωπεύει το μέλλον της ανάπτυξης διεπαφής χρήστη στην πλατφόρμα Android.

## Κεφάλαιο 4ο: Παρουσίαση Εφαρμογής

### 4.1 Android εφαρμογή



Εικόνα 4.1: Οθόνη Εισόδου

Η ανωτέρω εικόνα δείχνει το κομμάτι της σύνδεσης του χρήστη στην εφαρμογή. Αρχικά φαίνεται το λογότυπο της εφαρμογής και έπειτα παρουσιάζονται η λέξη 'Είσοδος' ακολουθούμενη από δύο ευδιάκριτα πεδία, το πεδίο email και το πεδίο κωδικού πρόσβασης. Αυτά είναι τυπικά στοιχεία μιας οθόνης σύνδεσης, που χρησιμοποιούνται για τον έλεγχο ταυτότητας του χρήστη. Το πεδίο email επιτρέπει στον χρήστη να εισάγει την καταχωρισμένη διεύθυνση email του, χρησιμεύοντας ως μοναδικό αναγνωριστικό για τον λογαριασμό. Το πεδίο κωδικού πρόσβασης χρησιμοποιείται για την εισαγωγή του κωδικού πρόσβασης του λογαριασμού και συνήθως καλύπτεται για τη διατήρηση του απορρήτου του χρήστη. Κάτω από τα πεδία email και κωδικού πρόσβασης, υπάρχει ένα κουμπί με την ένδειξη "Σύνδεση". Αυτό το κουμπί χρησιμεύει ως το έναυσμα ενεργειών για τη διαδικασία ελέγχου ταυτότητας. Μόλις οι χρήστες συμπληρώσουν τα διαπιστευτήριά τους και κάνουν κλικ στο κουμπί σύνδεσης, η εφαρμογή επιχειρεί να επαληθεύσει το παρεχόμενο email και τον κωδικό πρόσβασης στη βάση δεδομένων των χρηστών της. Αμέσως επόμενο είναι το κουμπί 'Εγγραφή' που συνάδει με την εγγραφή νέου χρήστη. Το κουμπί εγγραφής είναι ένα ουσιαστικό μέρος της διεπαφής σύνδεσης, παρέχοντας στους χρήστες μια διαδρομή για να δημιουργήσουν έναν νέο λογαριασμό εάν δεν έχουν ήδη. Όταν πατηθεί, το κουμπί καθοδηγεί τον χρήστη σε μια φόρμα εγγραφής όπου μπορεί να εισάγει τα απαραίτητα στοιχεία όπως

όνομα, email και κωδικό πρόσβασης. Αυτό επιτρέπει στους νέους χρήστες να δημιουργήσουν έναν λογαριασμό στο σύστημα και να έχουν πρόσβαση στις δυνατότητές του. Η παρουσία ενός κουμπιού εγγραφής ενθαρρύνει την απόκτηση χρηστών παρέχοντας έναν απλό και προσιτό τρόπο συμμετοχής στην πλατφόρμα. Έπειτα υπάρχει η επιλογή "Ξέχασα τον κωδικό μου!" και βρίσκεται από κάτω από το κουμπί εγγραφής. Αυτή η δυνατότητα είναι ζωτικής σημασίας για την ανάκτηση λογαριασμού χρήστη.

Η εικόνα περιλαμβάνει επίσης το κουμπί "Σύνδεση με Google", ένα παράδειγμα επιλογής σύνδεσης μέσω κοινωνικής δικτύωσης. Αυτό παρέχει μια εναλλακτική μέθοδο σύνδεσης, η οποία επιτρέπει στους χρήστες να ελέγχουν την ταυτότητα χρησιμοποιώντας τον λογαριασμό που έχουν ήδη από τη Google.

Κάνοντας κλικ σε αυτό το κουμπί, οι χρήστες ανακατευθύνονται στη σελίδα ελέγχου ταυτότητας της Google, όπου μπορούν να συνδεθούν με τα διαπιστευτήριά Google.

Μετά από επιτυχή έλεγχο ταυτότητας, ο χρήστης ανακατευθύνεται πίσω στην εφαρμογή και τα στοιχεία του λογαριασμού Google χρησιμοποιούνται για τη σύνδεσή του.

Λειτουργικότητα:

Η κύρια λειτουργία της διεπαφής σύνδεσης είναι ο έλεγχος ταυτότητας του χρήστη. Τα πεδία email και κωδικού πρόσβασης είναι τα κύρια μέσα με τα οποία οι χρήστες εισάγουν τα διαπιστευτήριά τους για να αποκτήσουν πρόσβαση στην εφαρμογή.

Μόλις ο χρήστης κάνει κλικ στο κουμπί σύνδεσης, το σύστημα επαληθεύει τα παρεχόμενα διαπιστευτήρια έναντι των αποθηκευμένων δεδομένων. Εάν τα διαπιστευτήρια ταιριάζουν με έναν υπάρχοντα λογαριασμό, ο χρήστης έχει πρόσβαση.

Το κουμπί ανάκτησης κωδικού παίζει καθοριστικό ρόλο στην ανάκτηση λογαριασμού, επιτρέποντας στους χρήστες να ανακτήσουν την πρόσβαση στους λογαριασμούς τους σε περίπτωση που ξεχάσουν τον κωδικό πρόσβασής τους. Όταν οι χρήστες επιλέγουν αυτήν την επιλογή, καθοδηγούνται σε μια διαδικασία ανάκτησης κωδικού πρόσβασης, η οποία περιλαμβάνει την αποστολή συνδέσμου επαναφοράς κωδικού πρόσβασης στο email τους το οποίο υλοποιείται από το firebase που έχει αναφερθεί σε προηγούμενη ενότητα.

Αυτή η δυνατότητα βελτιώνει την εμπειρία του χρήστη παρέχοντας ένα δίχτυ ασφαλείας για όσους μπορεί να δυσκολεύονται να θυμηθούν τα διαπιστευτήριά τους. Το κουμπί "Σύνδεση με Google" αποτελεί παράδειγμα της δυνατότητας σύνδεσης μέσω κοινωνικής δικτύωσης, η οποία παρέχει στους χρήστες μια βολική εναλλακτική λύση στον παραδοσιακό έλεγχο ταυτότητας με κωδικό email.

Χρησιμοποιώντας τον λογαριασμό της Google για να συνδεθούν, οι χρήστες μπορούν να βελτιστοποιήσουν τη διαδικασία σύνδεσης και να αποφύγουν την ανάγκη να θυμούνται πρόσθετα διαπιστευτήρια.

Συμπερασματικά η διεπαφή σύνδεσης που περιγράφεται στην εικόνα συνδυάζει τα βασικά στοιχεία ενός ασφαλούς και φιλικού προς το χρήστη συστήματος ελέγχου ταυτότητας. Τα πεδία email και κωδικού πρόσβασης επιτρέπουν στους χρήστες να εισάγουν τα διαπιστευτήριά τους, ενώ το κουμπί σύνδεσης ενεργοποιεί τη διαδικασία ελέγχου ταυτότητας. Η επιλογή "Ξέχασα τον κωδικό μου!" βελτιώνει τη χρηστικότητα προσφέροντας έναν μηχανισμό ανάκτησης λογαριασμού και το κουμπί "Σύνδεση με Google" προσφέρει μια απρόσκοπτη εναλλακτική μέθοδο σύνδεσης. Μαζί, αυτά τα στοιχεία σχηματίζουν μια συνεκτική και αποτελεσματική διεπαφή σύνδεσης που απευθύνεται σε ένα ευρύ φάσμα χρηστών και βελτιώνει τη συνολική εμπειρία τους.



Εικόνα 4.2: Οθόνη Εγγραφής χρήστη

Στην ανωτέρω εικόνα βλέπουμε τη διαδικασία εγγραφής. Ακριβώς όπως η διεπαφή σύνδεσης, η οθόνη εγγραφής περιλαμβάνει πεδία για το email και τον κωδικό πρόσβασης του χρήστη. Αυτά τα πεδία χρησιμοποιούνται για τους νέους χρήστες για την εισαγωγή των επιθυμητών διαπιστευτηρίων για τον νέο τους λογαριασμό. Το πεδίο email χρησιμεύει ως το κύριο αναγνωριστικό για τον λογαριασμό του χρήστη, ενώ το πεδίο κωδικού πρόσβασης είναι όπου ο χρήστης ορίζει έναν ασφαλή κωδικό πρόσβασης για τον λογαριασμό του.

Το κύριο κουμπί ενέργειας σε αυτήν τη διεπαφή φέρει την ένδειξη "Εγγραφή" ή "Εγγραφή". Όταν οι χρήστες συμπληρώσουν τα στοιχεία τους και κάνουν κλικ σε αυτό το κουμπί, η εφαρμογή δημιουργεί έναν νέο λογαριασμό χρήστη με τις παρεχόμενες πληροφορίες.

Το κουμπί εγγραφή ενεργοποιεί τη διαδικασία δημιουργίας λογαριασμού, ελέγχοντας τις παρεχόμενες πληροφορίες για την εγκυρότητά τους και πιθανώς στέλνοντας ένα email επαλήθευσης.

### Λειτουργικότητα:

Η διαδικασία εγγραφής ξεκινά όταν ο χρήστης κάνει κλικ στο κουμπί εγγραφή αφού εισάγει το email και τον κωδικό πρόσβασής του. Η εφαρμογή επαληθεύει ότι το email δεν χρησιμοποιείται ήδη και ότι ο κωδικός πρόσβασης πληροί τις απαιτήσεις ασφαλείας.

Μόλις δημιουργηθεί ο λογαριασμός, ο χρήστης μπορεί να λάβει ένα μήνυμα ηλεκτρονικού ταχυδρομείου ή μήνυμα επιβεβαίωσης για την επαλήθευση του λογαριασμού του.

Η διεπαφή εγγραφής είναι ζωτικής σημασίας για την απόκτηση και την ανάπτυξη χρηστών, επιτρέποντας στους νέους χρήστες να δημιουργήσουν εύκολα έναν λογαριασμό και να αρχίσουν να χρησιμοποιούν την εφαρμογή, παρέχοντας μια σαφή και φιλική προς τον χρήστη διαδικασία, η διεπαφή ενθαρρύνει νέους χρήστες να συμμετάσχουν και συμβάλλει στη συνολική επιτυχία της πλατφόρμας.



Εικόνα 4.3:Κεντρική οθόνη



Εικόνα 4.4:Κεντρική οθόνη

Στην παραπάνω εικόνα διακρίνουμε την πρώτη οθόνη που βλέπει ο χρήστης αφού κάνει τη σύνδεση του. Αρχικά, ξεκινώντας την ανάλυση αυτού του σημείου της εφαρμογής από πάνω προς τα κάτω, διακρίνουμε πάνω δεξιά ένα κουμπί από το οποίο ο χρήστης μπορεί να εξέλθει από την εφαρμογή. Πατώντας αυτό το κουμπί γίνεται αποσύνδεση του χρήστη και η εφαρμογή γυρνάει στην οθόνη που παραπέμπει στην είσοδο.

Συνεχίζοντας βλέπουμε έναν `page air` ο οποίος διαθέτει φωτογραφίες οι οποίες αφορούν το `project`. Είτε είναι από δράσεις, είτε από τον οργανισμό, είτε από ανθρώπους που συμμετέχουν στις εκδηλώσεις. Για την αλλαγή των εικόνων έχει ρυθμιστεί ένας χρονομετρητής ο οποίος είναι υπεύθυνος μετά το πέρας κάποιων δευτερολέπτων να δείχνει την επόμενη φωτογραφία. Επίσης, Ο χρήστης μπορεί να αλλάζει τις φωτογραφίες κάνοντας `swar` δεξιά ή αριστερά.

Παρακάτω παρατηρούμε μια `lazy row`. Στο σημείο αυτό υπάρχουν συγκεντρωτικά τα `events` του οργανισμού. Τα `events` τοποθετούνται ανάλογα με την χρονική σειρά που έχουν γίνει. Περιλαμβάνει γεγονότα που έχουν γίνει οι που έχουν δρομολογηθεί να γίνει. Λειτουργεί σαν `remember` για το χρήστη, δημιουργώντας έτσι κάτι σαν ημερολόγιο δράσεων.

Ένα ακόμα σημαντικό σημείο της εφαρμογής και συγκεκριμένα της οθόνης αυτής βρίσκεται λίγο πιο κάτω. Εκεί παρατηρούμε επίσης μία `lazy row` η οποία παρουσιάζει τις ανακοινώσεις της `riverview`. Επίσης αυτές τοποθετούνται ανάλογα με την χρονική σειρά με την οποία δημοσιεύτηκαν. Ενώ με το πάτημα πάνω σε μία από αυτές δημιουργείται ένα `pop up` παράθυρο το οποίο θα δούμε και θα σχολιάσουμε παρακάτω. Ο χρήστης θα μπορεί να κυλήσει προς τα αριστερά για να δει περισσότερες ανακοινώσεις. Στο σημείο αυτό αξίζει να σημειώσουμε ότι για κάθε ανακοίνωση η οποία προστίθεται, ο χρήστης λαμβάνει ένα `push notification` το οποίο τον ενημερώνει για την ύπαρξη μιας καινούργιες ανακοινώσεις. Δίνοντάς του έτσι το κίνητρο να εισέλθει στην εφαρμογή πατώντας πάνω στην ειδοποίηση, ώστε να διαβάσει την ανακοίνωση. το οποίο τον ενημερώνει για την ύπαρξη μιας καινούργιες ανακοινώσεις. Δίνοντάς του έτσι το κίνητρο να εισέλθει στην εφαρμογή πατώντας πάνω στην ειδοποίηση, ώστε να διαβάσει την ανακοίνωση.

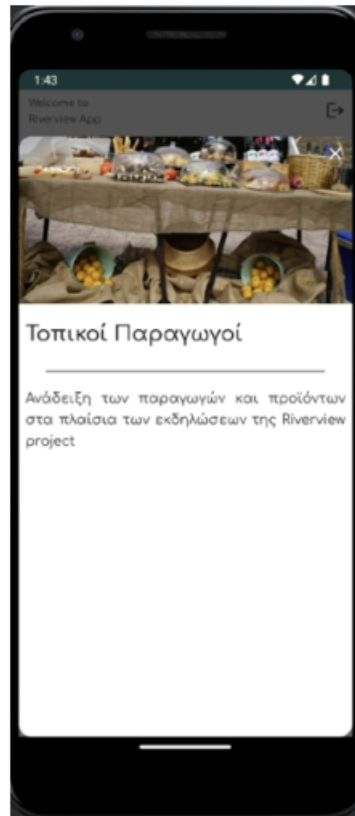
Κλείνοντας την οθόνη αυτή, υπάρχει ένα τελευταίο κομμάτι που αφορά τις δράσεις τις οποίες διοργανώνει η `riverview`. Ομοίως με το παραπάνω κομμάτι, με την επιλογή κάποιες δράσεις ανοίγει ένα παράθυρο πάνω από την προηγούμενη οθόνη. Εκεί υπάρχει κάποια φωτογραφία της δράσης και από κάτω μία περιγραφή αυτής.

Στο κάτω μέρος της οθόνης, όπως θα δούμε και σε όλες τις άλλες οθόνες, υπάρχει μπάρα με την οποία ο χρήστης θα μπορεί να μεταφέρεται από οθόνη σε οθόνη. Με τη σειρά από δεξιά προς τα αριστερά, αρχικά βλέπουμε την πρώτη επιλογή που είναι Με τη σειρά από δεξιά προς τα αριστερά, αρχικά βλέπουμε την πρώτη επιλογή που είναι κεντρική οθόνη της εφαρμογής. Την οποία την έχουμε αναλύσει πιο πάνω. Στη συνέχεια υπάρχει η επιλογή της οθόνης που αναφέρεται στον τουριστικό οδηγό. Η τρίτη κατα σειρά είναι η οθόνη που αναφέρεται στο

κατάστημα της εφαρμογής. Ενώ στο τέλος υπάρχει μία οθόνη , η οποία διαθέτει διάφορες πληροφορίες για την Riverview project. Επιπρόσθετα, ανάλογα με την οθόνη στην οποία βρίσκεται ο χρήστης, η αντίστοιχη επιλογή στην κάτω μπάρα χρωματίζεται ώστε να ξέρει ο χρήστης που βρίσκεται.

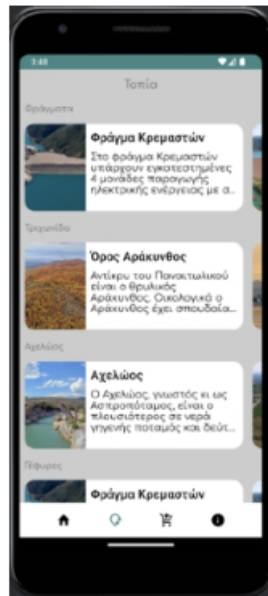


Εικόνα 4.5: Οθόνη Ανακοινώσεων



Εικόνα 4.6: Οθόνη Δράσεων

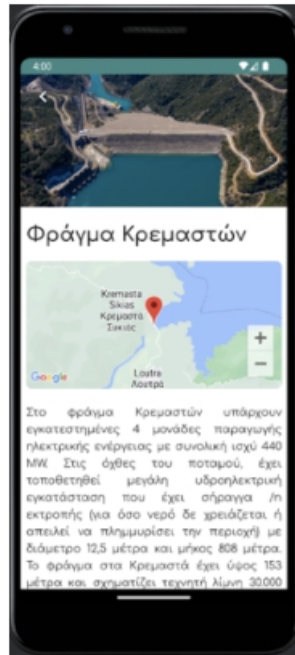
Στις παραπάνω εικόνες, βλέπουμε το παράθυρο που ανοίγει πατώντας σε μία ανακοίνωση ή σε μία δράση. Λειτουργικά αυτά τα δύο παράθυρα είναι το ίδιο, μόνο που το καθένα από αυτά αναφέρεται στο λόγο για τον οποίο δημιουργήθηκαν. Με το X πάνω δεξιά μπορούμε να κλείσουμε το παράθυρο, ενώ κάνοντας scroll μπορούμε να δούμε όλα τα περιεχόμενα.



Εικόνα 4.7: Οθόνη τουριστικού οδηγού

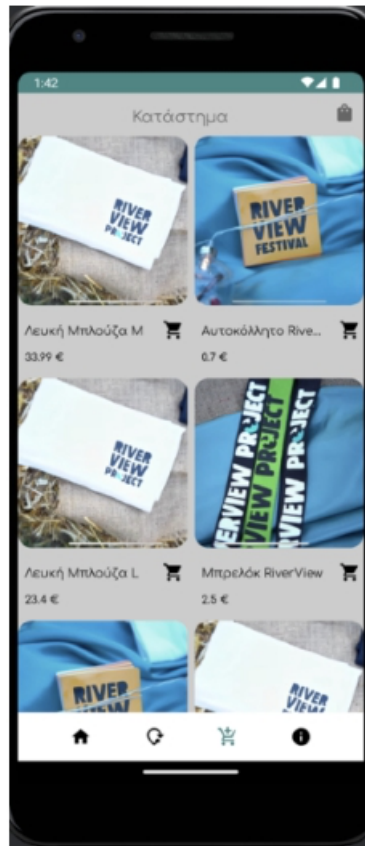
Συνεχίζοντας στην δεύτερη επιλογή που δίνεται στην μπάρα που υπάρχει στο κάτω μέρος της εφαρμογής, συναντάμε μία καρτέλα που έχει ως στόχο την ανάδειξη της φυσικής ομορφιάς του τόπου στον οποίο αναφέρεται η οργάνωση αυτή. Η επαφή με το περιβάλλον, η ανάδειξη του φυσικού πλούτου της περιοχής και η ευαισθητοποίηση των πολιτών με τη φύση αποτελεί ένα κύριο μέλημα της εν λόγω εταιρείας. Η παραπάνω εικόνα αναφέρεται στην οθόνη αυτή και δείχνει πως φαίνεται στο χρήστη.

Ξεκινώντας, για την δημιουργία της οθόνης, χωρίσαμε το σύνολο των τοπίων που είχαμε στη διάθεσή μας σε κατηγορίες ανάλογα με το τι υπάρχει στο τοπίο αυτό. Δημιουργήσαμε έτσι τέσσερις κατηγορίες και μέσα σε αυτές βάλουμε τα αντίστοιχα δεδομένα. Με το πάτημα σε ένα τοπίο από αυτά ανοίγει μία άλλη οθόνη η οποία είναι υπεύθυνη για την παρουσίαση των λεπτομερειών του εκάστοτε τοπίου. Η για την δημιουργία των κατηγοριών αυτών χρησιμοποιήθηκαν τέσσερα διαφορετικά lazy row τα οποία κάνοντας swap Ο χρήστης μπορεί να δει το σύνολό τους.



Εικόνα 4.8: Οθόνη τοπίου

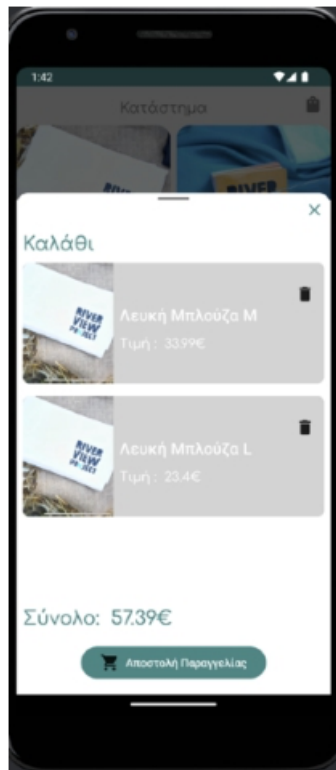
Κατεπέκταση της λειτουργικότητας της προηγούμενης οθόνης, η παραπάνω εικόνα παρουσιάζει την οθόνη που εμφανίζεται στο χρήστη, αφού αυτός επιλέξει κάποιο τοπίο. Όπως μπορούμε να διακρίνουμε αρχικά έχει τοποθετηθεί εικόνα του τοπίου το οποίο οφείλεται σε ένα image. Πάνω αριστερά στο image έχει τοποθετηθεί ένα εικονίδιο με το οποίο ο χρήστης μπορεί να πάει πίσω την οθόνη των τοπίων. Το πιο σημαντικό στην οθόνη αυτή βρίσκεται λίγο πιο κάτω. Με τη χρήση χαρτών της Google έχει εισαχθεί ένας χάρτης στον οποίον βρίσκεται ένα Πιν το οποίο αναφέρεται στην ακριβή τοποθεσία του τοπίου. Κάνοντας ταπ πάνω στο στίγμα αυτό, ο χρήστης μπορεί εύκολα να δεχθεί οδηγίες πως μπορεί να φτάσει εκεί. Με τον τρόπο αυτό, τα διάφορα γνωστά και μη τοπία, είναι προσεγγίσιμα για τον οποιοδήποτε. Στη συνέχεια υπάρχει μία περιγραφή του σημείου, την οποία ο χρήστης μπορεί να διαβάσει και να λάβει κάποιες πληροφορίες σχετικά με αυτό που θα επισκεφτεί.



Εικόνα 4.9: Οθόνη market

Το επόμενο κομμάτι της εφαρμογής, είναι αυτό του καταστήματος. Η οθόνη αυτή παρουσιάζεται στην παραπάνω εικόνα και κρίνεται πολύ σημαντική τόσο ως προς την ανάδειξη τοπικών προϊόντων όσο και για την οικονομική της ανάπτυξη του project, με σκοπό την ανάπτυξη του μέσα από δράσεις, εκδηλώσεις και οτιδήποτε άλλο κριθεί απαραίτητο. Στο σημείο αυτό, μπορεί ο κάθε παραγωγός να προωθήσει τα προϊόντα του. Ακόμη μπορεί το ίδιο το project να προωθήσει κάποια πράγματα διαφημιστικά γι' αυτό όπως για παράδειγμα διάφορα ρούχα που αναφέρονται στο project και πολλά άλλα. Πατώντας στο καλάθι που έχει δίπλα από την περιγραφή του προϊόντος ο χρήστης μπορεί να το βάλει στο καλάθι του.

Το καλάθι του χρήστη βρίσκεται στο πάνω δεξιά μέρος της οθόνης. Πατώντας σε αυτό ο χρήστης μπορεί να εισέλθει σε αυτό και να δει τα προϊόντα τα οποία έχει επιλέξει και είναι ως προς αγορά. Με το πάτημα του καλαθιού, ανοίγει ένα παράθυρο πάνω από την εφαρμογή.



Εικόνα 4.10: Εμφάνιση καλαθιού με αντικείμενα



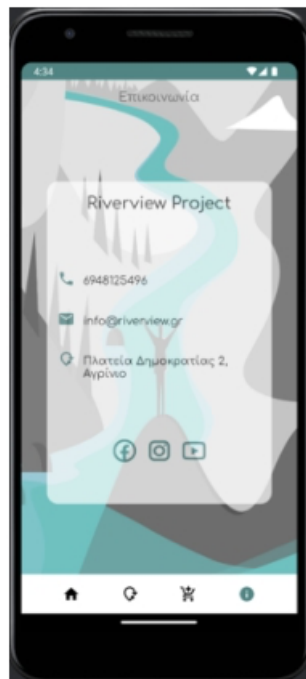
Εικόνα 4.11 : Εμφάνιση άδειου καλαθιού

Ανοίγοντας λοιπόν το καλάθι βλέπουμε ότι αν ο χρήστης δεν έχει προσθέσει κανένα προϊόν δεν μπορεί να προχωρήσει και στην καταχώριση της παραγγελίας. Ο χρήστης θα πρέπει να έχει τουλάχιστον ένα αντικείμενο ώστε να κάνει κάποια παραγγελία. Στο κάθε προϊόν υπάρχει και δυνατότητα του να σβήσει ο χρήστης το προϊόν που δεν χρειάζεται. Στο κάτω μέρος υπάρχει η επιλογή της αποστολής της παραγγελίας. Με το που πάτησε ο χρήστης το κουμπί αυτό, η παραγγελία πηγαίνει στη βάση δεδομένων με την οποία επικοινωνεί η εφαρμογή. Ο διαχειριστής της εφαρμογής διαθέτει μία ακόμη απλή εφαρμογή στην οποία μπορεί να βλέπει όλες τις παραγγελίες τις οποίες έχει. Εκεί μπορεί επίσης να έχει πρόσβαση στο ποιος την στέλνει την παραγγελία. Και να έρχεται σε επαφή με εκείνη το χρήστη για την επιβεβαίωση και την αποστολή της. Διατίθεται μόνο η παραγγελία με αντικαταβολή, έτσι εφαρμογή δεν έχει κάποιον τρόπο ώστε να μπορεί ο χρήστης να πληρώνει εξ αποστάσεως. Με την αποστολή της παραγγελίας τα πράγματα που χρήζει στο καλάθι, διαγράφονται και η λίστα είναι και πάλι κενή.

Με το κουμπί πάνω δεξιά ο χρήστης μπορεί να εξέλθει από το καλάθι χωρίς όμως να χάσει κάποια από τα πράγματα τα οποία έχει εισάγει σε αυτό. Με τον τρόπο αυτό οχ χρήστης μπορεί να προσθέσει κάποιο προϊόν το οποίο θέλει να προμηθευτεί και είχε ξεχάσει να το εισάγει.

Στο σημείο αυτό κρίνεται και απαραίτητο η έγγραφη και κατεπέκταση είσοδος του χρήστη στην εφαρμογή. Καθώς άμα δεν υπήρχε η σύνδεση του χρήστη στην εφαρμογή δεν θα μπορούσε Ο διαχειριστής της εφαρμογής να δει ποιος έκανε την παραγγελία και τα στοιχεία του. Συγκεκριμένα θα έπρεπε να φορτωθεί η εφαρμογή μία ακόμα οθόνη στην οποία ο χρήστης θα έπρεπε να βάλει τα στοιχεία του. Με τον τρόπο αυτό γίνονται όλα αυτόματα και ο χρήστης κάνει απλά μία εγγραφή.

Για την επίτευξη του καλαθιού του χρήστη χρησιμοποιήθηκε η τοπική βάση data store. Με τον τρόπο αυτό η εφαρμογή διατηρεί κάποια δεδομένα στη μνήμη της. Αυτά τα δεδομένα δεν είναι μεγάλα και προτιμήθηκε αυτός ο τρόπος αποπεράτωσης της λειτουργίας ως την εφαρμογή να είναι γρήγορη και απολύτως λειτουργική.



Εικόνα 4.12: Οθόνη επικοινωνίας

Το τελευταίο και εξίσου σημαντικό στάδιο της εφαρμογής είναι αυτό της επικοινωνίας. Κρίνεται απαραίτητο καθώς ο χρήστης γίνεται γνώστης του μέρους και του τρόπου με τον οποίο μπορεί να έρθει σε επαφή με την εταιρεία. Αρχικά, στο σημείο αυτό βλέπουμε ότι υπάρχει ένα κινητό τηλέφωνο το οποίο αντιστοιχεί στην οργάνωση. Πατώντας πάνω στον αριθμό κινητού, αυτόματα ανοίγει η εφαρμογή των κλήσεων του κινητού, έχοντας ήδη συμπληρωμένο τον αριθμό για να καλέσει ο χρήστης.

Στην συνέχεια, υπάρχει το e-mail του project. Στο e-mail αυτό μπορεί ο χρήστης να πατήσει και να κατευθυνθεί σε κάποια εφαρμογή ηλεκτρονικού ταχυδρομείου με σκοπό την αποστολή

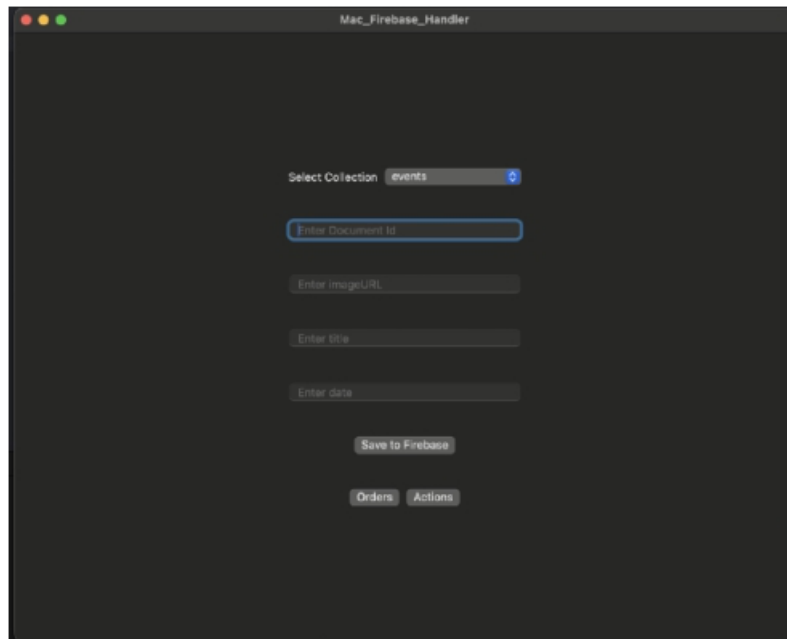
κάποιο mail προς την οργάνωση. Αξίζει να σημειωθεί επίσης ότι το e-mail θα είναι συμπληρωμένο στο πεδίο του παραλήπτη.

Ακόμη, στο τρίτο κατά σειρά πεδίο βρίσκεται το γεωγραφικό στίγμα στο οποίο η οργάνωση έχει τα γραφεία της και θα μπορεί ο χρήστης να βρεθεί αυτοπροσώπως και να συζητήσει το θέμα που τον απασχολεί. Πατώντας πάνω σε αυτό, θα μεταφερθεί στους χάρτες της Google και με το στίγμα που θα υπάρχει εκεί, μπορεί να ζητήσει οδηγίες για την μετακίνηση του στο σημείο αυτό.

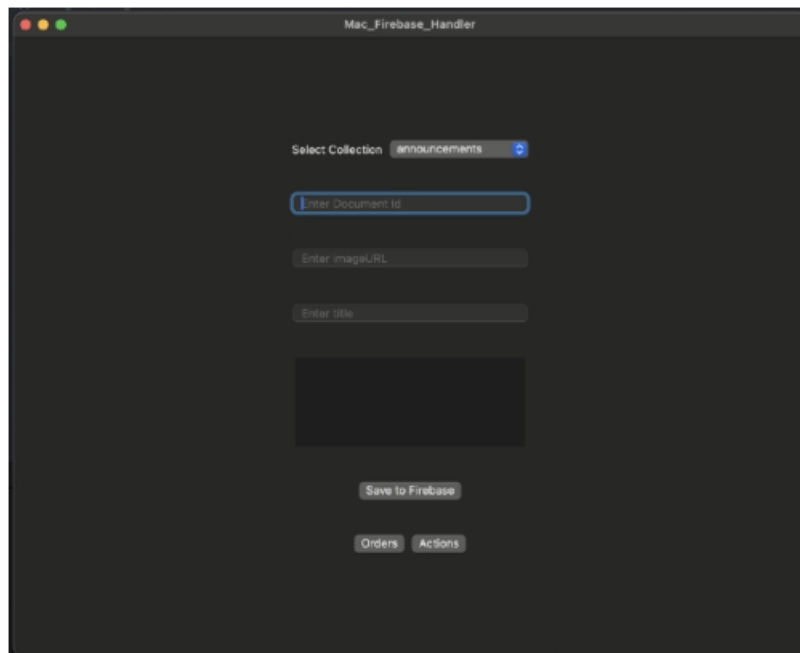
Όπως είναι γνωστό, Ο κύριος τρόπος για την προώθηση πολλών πραγμάτων είναι αυτός των social media. Έτσι και το project αυτό διαθέτει λογαριασμούς σε πολλά από αυτά. Το γεγονός αυτό κρίνει απαραίτητο στην εφαρμογή να υπάρχει και αναφορά σε αυτά. Το τελευταίο και πολύ σημαντικό της οθόνης αυτής είναι αυτό των social media. Στο κάτω μέρος της οθόνης βλέπουμε τρία εικονίδια τα οποία αναφέρονται στο facebook, ίνσταγκραμ και YouTube αντίστοιχα. Πατώντας πάνω στο πρώτο εικονίδιο, από τα αριστερά προς τα δεξιά, θα γίνει η ανακατεύθυνση του χρήστη στην σελίδα του οργανισμού στο facebook. Πατώντας στο δεύτερο κατά σειρά εικονίδιο, θα γίνει ανακατεύθυνση του στη σελίδα του οργανισμού στο in στα Γκραμ και τέλος με το πάτημα του τρίτου και τελευταίου εικονιδίου, ο χρήστης θα μεταφερθεί στην σελίδα της εταιρείας στο YouTube. Με τον τρόπο αυτό, διατηρούμε ενεργό τον χρήστη με οτιδήποτε γύρω από την εταιρεία αυτή. Ενώ ταυτόχρονα υπάρχει μία συχνή επικοινωνία μεταξύ του χρήστη και της Riverview.

## 4.2 Desktop App

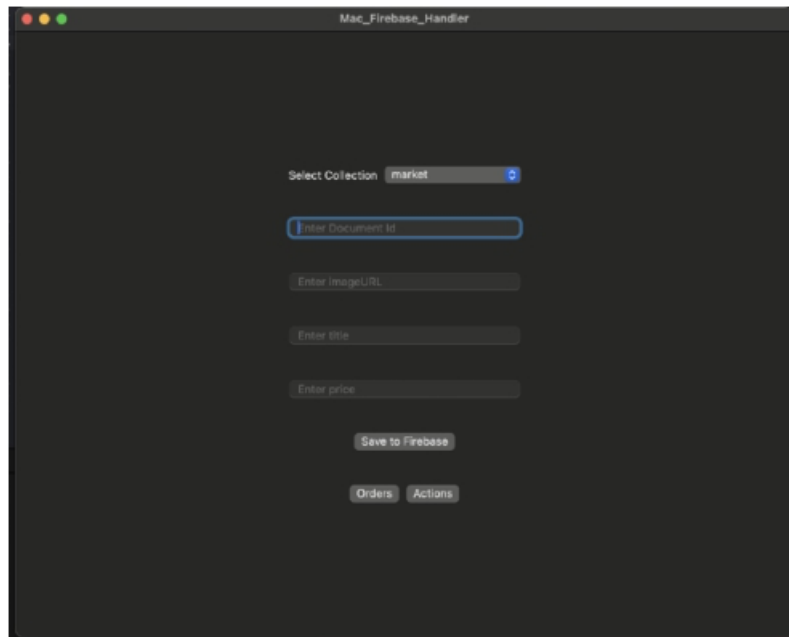
Για την ενημέρωση της εφαρμογής, έχει αναπτυχθεί μία desktop εφαρμογή στην οποία ο διαχειριστής μπορεί να ανεβάζει ανακοινώσεις, δράσης, φωτογραφίες, προϊόντα προς πώληση, να βλέπει τις παραγγελίες κτλ. Με τον τρόπο αυτό γίνεται Η εφαρμογή πιο φιλική και στον χρήστη αλλά και στον διαχειριστή. Παρακάτω ακολουθούν κάποια στιγμιότυπα οθόνης τα οποία παρουσιάζουν το app αυτό. Συγκεντρώθηκαν έχουμε μία ολοκληρωμένη εφαρμογή η οποία είναι πλήρως λειτουργική.



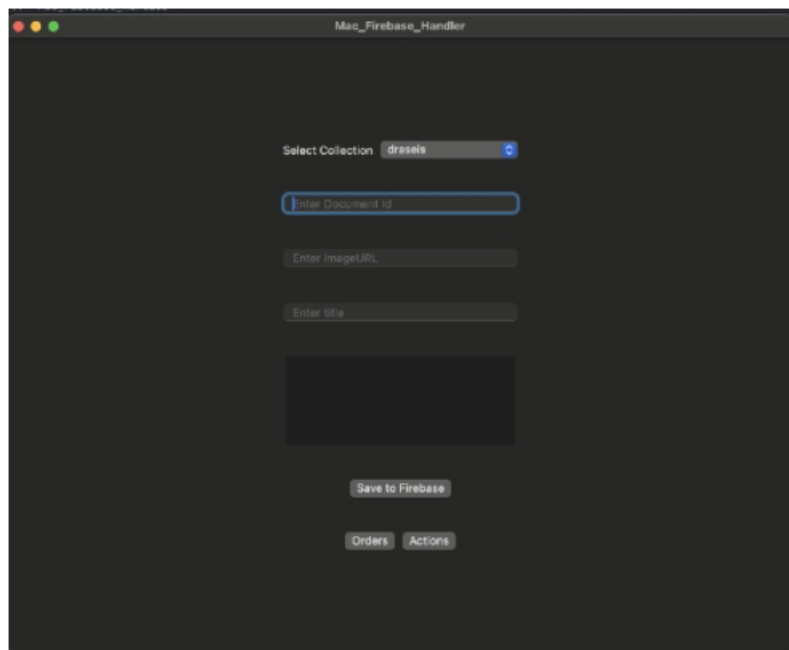
Εικόνα 4.13: Οθόνη Desktop app για ανέβασμα event



Εικόνα 4.14: Οθόνη Desktop app για ανέβασμα ανακοίνωσης



Εικόνα 4.16: Οθόνη Desktop app για ανέβασμα market item



Εικόνα 4.17: Οθόνη Desktop app για ανέβασμα δράσης

**ΒΙΒΛΙΟΓΡΑΦΙΑ**

- [1] Google, Android Developers, <https://developer.android.com/studio/intro>.
- [2] Google, Android Developers, Android SDK <https://developer.android.com/tools>
- [3] Google, Android Developers , <https://developer.android.com/tools/releases/platforms>
- [4] Gradle <https://developer.android.com/build/releases/gradle-plugin>
- [4] <https://git-scm.com/book/en/v2/Getting-Started-What-is-Git?>
- [5] <https://docs.gitlab.com/>
- [6] Google, Android Developers , Android Coroutines <https://developer.android.com/kotlin/coroutines>
- [7] Google,Android Developers, Architecture Patterns <https://developer.android.com/topic/architecture#recommended-app-arch>
- [8] [https://www.google.gr/books/edition/Clean\\_Android\\_Architecture/4QtxEAAAQBAJ?hl=el&gbpv=1&dq=Clean+Architecture+%C2%B5%CE%B5+MVVM+android+dev&printsec=frontcover](https://www.google.gr/books/edition/Clean_Android_Architecture/4QtxEAAAQBAJ?hl=el&gbpv=1&dq=Clean+Architecture+%C2%B5%CE%B5+MVVM+android+dev&printsec=frontcover)
- [9] Clean Architecture [https://www.google.gr/books/edition/Clean\\_Architecture\\_for\\_Android/jGmSEAAAQBAJ?hl=el&gbpv=1&dq=clean+architecture+android+dev&printsec=frontcover](https://www.google.gr/books/edition/Clean_Architecture_for_Android/jGmSEAAAQBAJ?hl=el&gbpv=1&dq=clean+architecture+android+dev&printsec=frontcover)
- [10] Google,Android Developers, Shared Preferences <https://developer.android.com/training/data-storage/shared-preferences>
- [11]Google,Android Developers, Datastore <https://developer.android.com/topic/libraries/architecture/datastore>
- [12]Google,Android Developers, hilt <https://developer.android.com/training/dependency-injection/hilt-android>
- [13] Google,Android Developers, Firebase <https://developer.android.com/studio/write/firebase>
- [14] Google,Android Developers, Analytics <https://developer.android.com/topic/google-play-instant/guides/analytics>
- [15] Google,Android Developers, Crashlytics <https://developer.android.com/studio/debug/app-quality-insights>
- [16] Google,Android Developers, Push Notifications <https://firebase.google.com/docs/cloud-messaging/android/receive>
- [17]Gradle <https://developer.android.com/build/releases/gradle-plugin>

- [18] Google,Android Developers, Android Manifest <https://developer.android.com/guide/topics/manifest/manifest-intro>
- [19] Google,Android Developers, Activity <https://developer.android.com/guide/components/activities/intro-activities>
- [20] Google,Android Developers, Fragment <https://developer.android.com/reference/android/app/Fragment>
- [21] Google,Android Developers, Navigation Components <https://developer.android.com/guide/navigation>
- [22] Google,Android Developers, Compose <https://developer.android.com/develop/ui/compose/documentation>
- [23] Google,Android Developers, Column <https://developer.android.com/develop/ui/compose/layouts/basics>
- [24] Google,Android Developers, row <https://developer.android.com/develop/ui/compose/layouts/basics>
- [25] Google,Android Developers, LazyColumn <https://developer.android.com/develop/ui/compose/lists>
- [26] Google,Android Developers, LazyRow <https://developer.android.com/develop/ui/compose/lists>
- [27] Google,Android Developers, Grids στο Jetpack Compose LazyRow <https://developer.android.com/develop/ui/compose/lists>
- [28] Google,Android Developers, Pager <https://developer.android.com/develop/ui/compose/layouts/pager>
- [29] Google,Android Developers, Modifiers στο Jetpack Compose <https://developer.android.com/develop/ui/compose/modifiers>
- [30] Google,Android Developers, States στο Jetpack Compose <https://developer.android.com/develop/ui/compose/state>
- [31] Google,Android Developers, Material Design <https://developer.android.com/develop/ui/compose/designsystems/material3>
- [32] Google,Android Developers, Google Maps Android API <https://developer.android.com/develop/ui/compose/libraries>

## Παραρτήματα

Παράρτημα 3.1: Διάφορα plugins που υπάρχουν στο gradle.....	18
Παράρτημα 3.2: Διάφορες εξαρτήσεις που υπάρχουν στο gradle.....	19
Παράρτημα 3.3: Ρύθμιση της έκδοσης του SDK .....	19
Παράρτημα 3.4: Το Android Manifest της εφαρμογής 1/2 .....	20
Παράρτημα 3.5: Το Android Manifest της εφαρμογής 2/2 .....	21
Παράρτημα 3.6: Παράδειγμα παράδειγμα Fragment .....	22
Παράρτημα 3.7: Χρήση Compose View .....	24
Παράρτημα 3.8: Παράδειγμα Column .....	26
Παράρτημα 3.9: Παράδειγμα row .....	27
Παράρτημα 3.10 : Παράδειγμα Lazy Row.....	28
Παράρτημα 3.11: Παράδειγμα LazyVerticalStaggeredGrid .....	31
Παράρτημα 3.12 : Παράδειγμα pager .....	33

## Επίλογος/Συμπεράσματα

Μια εφαρμογή Android φτιαγμένη ακριβώς για μια ιστοσελίδα σαν το RiverView Project βοηθάει στην ενίσχυση μιας κοινότητας και την προώθηση του τουρισμού αλλά και των επιχειρήσεων, στην περιοχή του Αγρινίου. Αναλυτικότερα, η εφαρμογή αυτή έχει την δυνατότητα να παρέχει στους χρήστες της, πρόσβαση σε εκδηλώσεις και δραστηριότητες, αλλά και τους παρακινεί να γίνουν μέλη αυτών των δράσεων. Εκτός από την βελτίωση της προσβασιμότητας, η οποία βοηθάει στην άμεση ενημέρωση των χρηστών, διευκολύνεται επίσης η επικοινωνία, που συμβάλλει στην οργάνωση μιας συνδεδεμένης κοινότητας. Η οικονομία της κοινωνίας του Αγρινίου ενισχύεται δραστικά, μέσω της προώθησης τουριστικών επιχειρήσεων (ξενοδοχείων, μαγαζιών με τοπικά προϊόντα, εστίασης κλπ.) και οι χρήστες έχουν την ικανότητα να μπορούν να χειρίζονται με ευκολία τις δραστηριότητες, αλλά και τις προσωπικές τους προτιμήσεις. Οι προσδοκίες για το μέλλον είναι ελπιδοφόρες για την δημιουργία της ίδια εφαρμογής για iOS, ώστε να συμπεριληφθούν και οι χρήστες των κινητών με λογισμικό iOS. Ανακεφαλαιώνοντας, η εφαρμογή είναι συντεθειμένη ειδικά για να πετύχει τους στόχους του Riverview Project, καθώς παρέχει ένα δυνατό εργαλείο, που προωθεί το κοινωνικό αλλά και συλλογικό συμφέρον της κοινωνίας, και ενισχύει τον ίδιο οργανισμό που την χρησιμοποιεί.