



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
Random Traveller



Του φοιτητή
Βασίλειου Σαμολαδά
Αρ. Μητρώου: 144270

Επιβλέπων
Κωσταντίνος Γουλιάννα
Βαθμίδα Καθηγητής

Ημερομηνία 01/06/2025

Τίτλος Δ.Ε Random Traveller - Επιλογές ταξιδιών με αεροπλάνο σύμφωνα με κάποιες προϋποθέσεις

Κωδικός Δ.Ε. 21382

Όνοματεπώνυμο φοιτητή/τών Βασίλειος Σαμολαδάς

Όνοματεπώνυμο εισηγητή Γουλιάνας Κωνσταντίνος

Ημερομηνία ανάληψης Δ.Ε. 16-10-2021

Ημερομηνία περάτωσης Δ.Ε. 01/06/2025

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 Σαμολαδά Βασίλη που την εκπόνησε/αν. 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 οποιονδήποτε 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.

Περίληψη

Στη σύγχρονη εποχή, πολλοί άνθρωποι αναζητούν την περιπέτεια και την ελευθερία του ταξιδιού χωρίς να δεσμεύονται από έναν συγκεκριμένο προορισμό. Η πτυχιακή αυτή εργασία παρουσιάζει μια

καινοτόμο εφαρμογή που ανταποκρίνεται σε αυτή την επιθυμία, προσφέροντας στους χρήστες τη δυνατότητα να ανακαλύψουν μοναδικές ταξιδιωτικές εμπειρίες με βάση τον προϋπολογισμό τους. Χωρίς την ανάγκη για εκτεταμένο σχεδιασμό, οι χρήστες μπορούν απλά να εισάγουν το σημείο αναχώρησης, τις διαθέσιμες ημέρες και το ποσό που επιθυμούν να δαπανήσουν, και η εφαρμογή θα τους προτείνει προορισμούς που ταιριάζουν στις ανάγκες τους. Αυτή η προσέγγιση όχι μόνο διευκολύνει την οργάνωση ταξιδιών, αλλά και εμπνέει τους ταξιδιώτες να εξερευνήσουν νέους ορίζοντες με ευκολία και οικονομία. Για όλους αυτούς τους λόγους, η παρούσα πτυχιακή εργασία επικεντρώνεται στην ανάπτυξη μιας εφαρμογής για κινητές συσκευές, ειδικά σχεδιασμένης για την πλατφόρμα Android, που θα προσφέρει στους χρήστες την ευκαιρία να ανακαλύψουν νέους προορισμούς με βάση τις προσωπικές τους προτιμήσεις και τον προϋπολογισμό τους.

«Random Traveller»

«Vasileios Samoladas»

Abstract

The current thesis aims to develop an Android application called Random Traveller. This application is designed to cater to travelers who wish to explore destinations without a predetermined plan. The application allows users to input key details such as their departure airport, the number of days available for travel, and their budget for airfare. Based on this information, the application provides a list of potential travel destinations, ranked from the most affordable to the most expensive options. This innovative application empowers users to discover new destinations and plan trips that align with their preferences and budget, offering a flexible and inspiring travel experience.

Ευχαριστίες

Η παρούσα πτυχιακή εργασία εκπονήθηκε στο Διεθνές Πανεπιστήμιο της Ελλάδος, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων, κατά το έτος 2025. Θα ήθελα να εκφράσω την ειλικρινή μου ευγνωμοσύνη προς τον καθηγητή μου, κ. Γουλιάνα Κωνσταντίνο, για την ευκαιρία που μου προσέφερε να υλοποιήσω αυτή την πτυχιακή εργασία, καθώς και για την αδιάκοπη υποστήριξή του καθ' όλη τη διάρκεια αυτής της διαδικασίας.

Περιεχόμενα

Περίληψη	3
Abstract	4
Ευχαριστίες	5
Περιεχόμενα	6
Συντομογραφίες	8
Κεφάλαιο 1ο: Πλατφόρμες Κρατήσεων Αεροπορικών Πτήσεων (OTA)	1
1.1 Εισαγωγή	1
1.1.1 Το πρόβλημα	1
1.1.2 Η σημασία της λύσης	1
1.1.3 Το κοινό-στόχος (Target Audience)	1
1.1.4 Το όραμα της εφαρμογής	1
1.1.5 Παράδειγμα Χρήσης (Use Case)	1
1.2 Ανάλυση Υφιστάμενων OTA Πλατφορμών	2
1.2.1 Skyscanner	2
1.2.2 Kiwi.com	3
1.2.3 eDreams ODIGEO	4
1.2.4 CheapOair	5
1.2.5 Kayak	6
1.2.6 Trip.com	7
1.3 Διαφοροποίηση και αναγκαιότητα της εφαρμογής Random Traveller σε σχέση με τις υφιστάμενες OTA πλατφόρμες	8
Κεφάλαιο 2ο: Λειτουργικό σύστημα Android	10
2.1 Εισαγωγή	10
2.2 Τι είναι το Android	10
2.3 Λογισμικού ανοιχτού κώδικα	11
2.4 Η Αρχιτεκτονική του Android	11
2.5 Επίλογος	12
Κεφάλαιο 3ο: Είδη Android εφαρμογών	13
3.1 Εισαγωγή	13
3.2 Εφαρμογές στο προσκήνιο	13
3.3 Υβριδικές εφαρμογές	13
3.4 Widgets	13
3.5 Εφαρμογές Android Wear (Smartwatches)	13
3.6 Εφαρμογές Android TV	13
Κεφάλαιο 4ο: Κύκλος ζωής εφαρμογών Android	14
4.1 Εισαγωγή	14
4.2 Καταστάσεις	14
4.3 Χρήση στην εφαρμογή Random Traveller	15
4.4 Σύνοψη και Συμπεράσματα	15
Κεφάλαιο 5ο: Συστατικά στοιχεία εφαρμογών Android	17
5.1 Εισαγωγή	17

5.2 Manifest	17
5.3 Activity	18
5.4 Intents	18
5.5 Fragments	18
5.6 Threads	19
5.7 AsyncTask	19
5.8 Layouts	20
5.9 Dialogs	21
5.10 Context	21
Κεφάλαιο 6ο: Τεχνολογίες που χρησιμοποιήθηκαν	23
6.1 Εισαγωγή	23
6.2 Google Firebase	23
6.2.1 Εισαγωγή	23
6.2.2 Ιστορία	24
6.2.3 Προσφερόμενες λειτουργίες	24
6.2.4 Προσθέτοντας το Firebase σε εφαρμογή	25
6.2.5 Firebase Authentication	25
6.2.6 Gradle	28
6.2.7 Kotlin	30
6.3 Compose Navigation	33
6.4 GraphQL	35
6.5 Apollo Client	36
6.6 OkHttp	36
6.7 Kiwi's API	38
6.8 Dependency Injection with Hilt	38
6.9 Εργαλεία UX/UI και Σχεδιαστική Προσέγγιση	40
6.9.1 Εισαγωγή	40
6.9.2 Επιλογή Figma	40
6.9.3 Αρχές Σχεδιασμού	40
6.9.4 Παραδείγματα Σχεδίασης	40
6.9.5 Συμπεράσματα	41
Κεφάλαιο 7ο: Ροή εφαρμογής	42
7.1 Εισαγωγή	42
7.2 Εγγραφή	42
7.3 Αρχική οθόνη (Οθόνη αναζήτησης)	46
7.4 Αυτόματη Συμπλήρωση Αεροδρομίου	47
7.5 Τελευταία αναζήτηση	49
7.6 Οθόνη Αναμονής Αποτελεσμάτων	50
7.7 Οθόνη Αποτελεσμάτων Πτήσεων	52
7.8 Οθόνη Αποθηκευμένων Αναζητήσεων	53
7.9 Πλευρικό μενού πλοήγησης	56
7.10 Οθόνη Μετάβασης στην Kiwi.com	58
ΒΙΒΛΙΟΓΡΑΦΙΑ	60

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
Π.Ε.	Πτυχιακή Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
OTA	Online Travel Agency
API	Application Programming Interface
GDS	Global Distribution System
UI	User Interface
SDK	Software Development Kit
IDE	Integrated Development Environment Studio
JVM	Java Virtual Machine
XML	eXtensible Markup Language
GPS	Global Positioning System
JSON	JavaScript Object Notation
CRUD	Create, Read, Update, Delete
CPU	Central Processing Unit

Κεφάλαιο 1ο: Πλατφόρμες Κρατήσεων Αεροπορικών Πτήσεων (ΟΤΑ)

1.1 Εισαγωγή

1.1.1 Το πρόβλημα

Οι Online Travel Agencies (OTAs) στον τομέα των αεροπορικών πτήσεων λειτουργούν ως ψηφιακοί ενδιάμεσοι ανάμεσα στον τελικό καταναλωτή και τις αεροπορικές εταιρείες. Χρησιμοποιούν APIs ή συνδέσεις με Global Distribution Systems (GDS) όπως τα Amadeus, Sabre και Travelport, για να αντλούν σε πραγματικό χρόνο πληροφορίες για διαθεσιμότητα, τιμές, κανονισμούς εισιτηρίων και φόρους.

Παρά τις εξελιγμένες δυνατότητες αναζήτησης και τις πληθώρα επιλογών, οι σύγχρονες ΟΤΑ πλατφόρμες προϋποθέτουν από τον χρήστη να γνωρίζει συγκεκριμένο προορισμό, ημερομηνίες και συχνά απαιτούν πολύπλοκο φιλτράρισμα. Αυτό καθιστά τη διαδικασία αποτρεπτική για χρήστες που αναζητούν αυθόρμητα και οικονομικά ταξίδια χωρίς συγκεκριμένο σχεδιασμό. Η πληροφοριακή υπερφόρτωση οδηγεί συχνά σε αβεβαιότητα και εγκατάλειψη της κράτησης.

1.1.2 Η σημασία της λύσης

Η εφαρμογή Random Traveller απλοποιεί ριζικά αυτή τη διαδικασία. Απευθύνεται κυρίως σε χρήστες που διαθέτουν περιορισμένο προϋπολογισμό και χρόνο, προσφέροντάς τους ταξιδιωτικές επιλογές χωρίς να απαιτείται συγκεκριμένος προορισμός ή ημερομηνία. Έτσι, επιτρέπει σε περισσότερους ανθρώπους να αξιοποιούν ευκαιρίες που διαφορετικά θα αγνοούσαν, συμβάλλοντας στην ενίσχυση του short-break τουρισμού και ενισχύοντας το στοιχείο της έκπληξης και του αυθορμητισμού.

1.1.3 Το κοινό-στόχος (Target Audience)

Η εφαρμογή απευθύνεται σε:- Φοιτητές ή νέους επαγγελματίες με περιορισμένο budget,- Ταξιδιώτες που επιθυμούν αποδράσεις τελευταίας στιγμής (last-minute deals),- Χρήστες που δεν έχουν συγκεκριμένο προορισμό στο μυαλό τους και αναζητούν έμπνευση.

1.1.4 Το όραμα της εφαρμογής

Το Random Traveller στοχεύει να αποτελέσει ένα καινοτόμο εργαλείο που εμπνέει νέες ταξιδιωτικές εμπειρίες, χωρίς τις παραδοσιακές δεσμεύσεις του προγραμματισμού. Το όραμα είναι η δημιουργία μιας εφαρμογής που κάνει το ταξίδι προσβάσιμο, ευχάριστο και αυθόρμητο, βασισμένο μόνο σε τρία στοιχεία: σημείο αναχώρησης, αριθμός ημερών και προϋπολογισμός.

1.1.5 Παράδειγμα Χρήσης (Use Case)

Φανταστείτε έναν χρήστη από τη Θεσσαλονίκη, ο οποίος έχει στη διάθεσή του ένα τριήμερο και προϋπολογισμό 150€. Χωρίς να χρειάζεται να επιλέξει συγκεκριμένο προορισμό ή να περάσει από δεκάδες φίλτρα, ανοίγει την εφαρμογή Random Traveller, εισάγει απλώς το αεροδρόμιο αναχώρησης, τις διαθέσιμες ημερομηνίες και το ποσό που μπορεί να ξοδέψει. Μέσα σε δευτερόλεπτα, η εφαρμογή του παρουσιάζει προτάσεις για αυθεντικές, οικονομικές αποδράσεις – όπως π.χ. μια πτήση προς Σόφια, ένα ταξίδι στην Κωνσταντινούπολη ή μια βόλτα στο Βουκουρέστι. Ο χρήστης μπορεί να

εμπνευστεί από μέρη που ίσως δεν είχε σκεφτεί καν, εξοικονομώντας χρόνο και χρήμα και μετατρέποντας ένα απλό τριήμερο σε αξέχαστη εμπειρία.

1.2 Ανάλυση Υφιστάμενων OTA Πλατφορμών

Για να αναδειχθεί η καινοτομία και η αξία της εφαρμογής Random Traveller, είναι κρίσιμο να προηγηθεί μια ανάλυση των κυριότερων ανταγωνιστικών πλατφορμών κρατήσεων πτήσεων (OTA) και μετα-μηχανών αναζήτησης. Οι πλατφόρμες αυτές, όπως οι Skyscanner, Kiwi.com, eDreams, Kayak και Trip.com, διαμορφώνουν το υφιστάμενο τοπίο των ψηφιακών κρατήσεων και καθορίζουν τις προσδοκίες των χρηστών. Μέσω της παρουσίασης των βασικών χαρακτηριστικών, της τεχνολογίας και της εμπειρίας χρήστη που προσφέρει η κάθε μία, επιδιώκεται να τεθούν οι βάσεις για τη σύγκριση και να αναδειχθούν οι αδυναμίες που καλύπτει η εφαρμογή Random Traveller.

1.2.1 Skyscanner

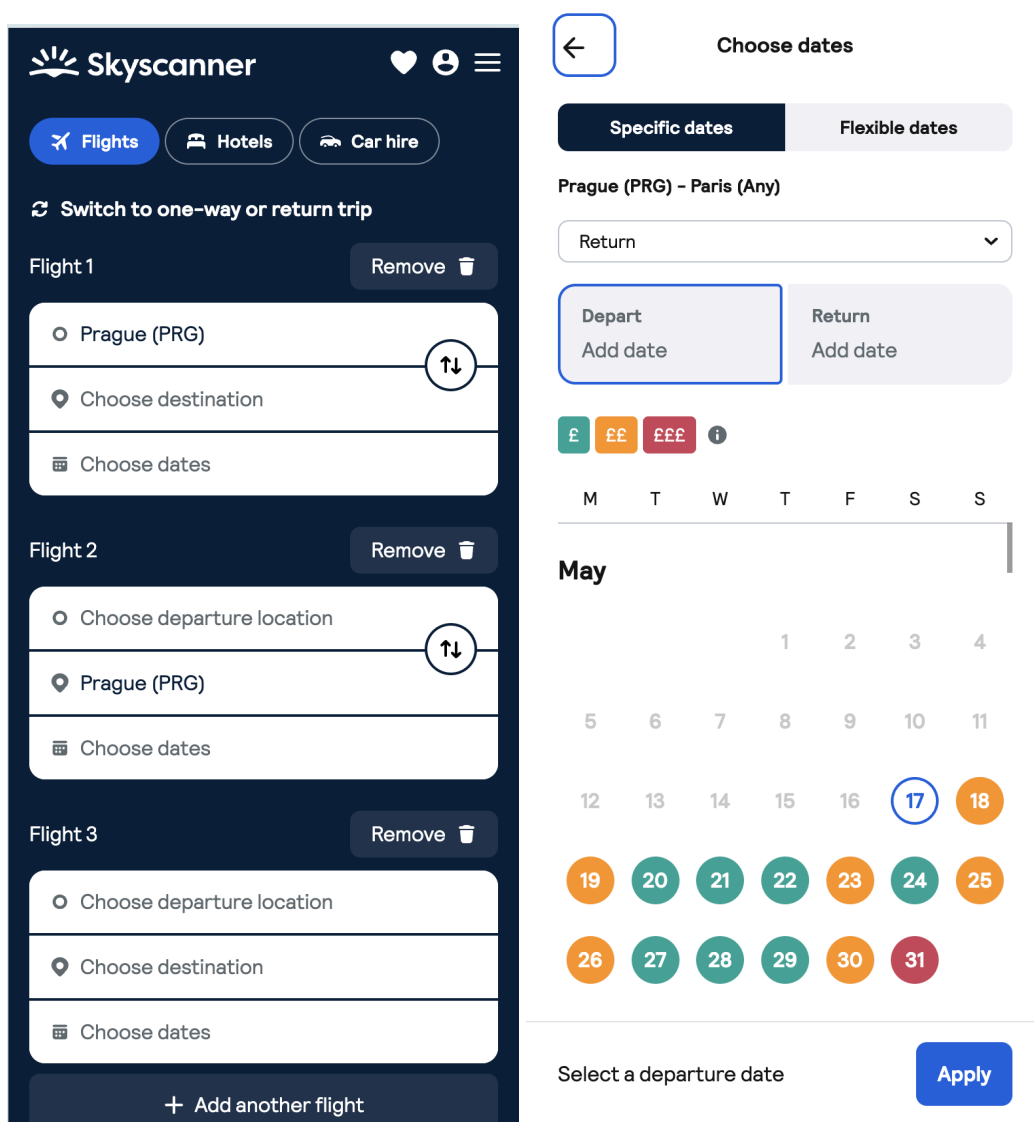
Το Skyscanner είναι μετα-μηχανή αναζήτησης πτήσεων, που δεν εκτελεί ο ίδιος κρατήσεις αλλά συγκρίνει αποτελέσματα από OTA, GDS και απευθείας συνεργάτες (αεροπορικές). Χρησιμοποιεί cache δεδομένων για ταχύτητα αλλά προσφέρει και real-time τιμολόγηση μέσω συνδεδεμένων API.

Πώς λειτουργεί:

- Η μηχανή του Skyscanner λαμβάνει και ενοποιεί δεδομένα από διαφορετικούς παρόχους.
- Οι χρήστες μπορούν να φιλτράρουν βάσει αριθμού στάσεων, χρόνου, αεροπορικής, κ.λπ.
- Η κράτηση ολοκληρώνεται μέσω ανακατεύθυνσης ή εντός της εφαρμογής με συνεργάτες που υποστηρίζουν “direct booking”.

Τι μπορεί να κάνει:

- Παρουσιάζει συνδυαστικά δρομολόγια (π.χ. Ryanair + Aegean).
- Ειδοποιεί με push notifications για αλλαγές τιμών.
- Υποστηρίζει πολυ-πόλεις και open-jaw κρατήσεις.



Οθόνη αναζήτησης και επιλογής ημερομηνιών στο Skyscanner

1.2.2 Kiwi.com

Η Kiwi.com χρησιμοποιεί εξελιγμένο αλγόριθμο που δημιουργεί “εικονικά δρομολόγια” (virtual interlining), συνδυάζοντας πτήσεις διαφορετικών αεροπορικών, ακόμα κι αν δεν υπάρχει συμφωνία συνεργασίας μεταξύ τους. Η Kiwi εγγυάται τη σύνδεση με τη λεγόμενη “Kiwi Guarantee”.

Πώς λειτουργεί:

- Αντλεί δεδομένα από GDS και αεροπορικές.
- Ο αλγόριθμος συνθέτει δρομολόγια που δεν υπάρχουν σε παραδοσιακά συστήματα.
- Δημιουργεί ένα ενιαίο PNR και αναλαμβάνει υποστήριξη σε περίπτωση καθυστερήσεων.

Τι μπορεί να κάνει:

- Συνδυάζει πτήσεις low-cost με legacy carriers.
- Υποστηρίζει επαναπρογραμματισμό μέσω υποστήριξης 24/7.
- Παρέχει χάρτη οπτικής αναζήτησης με πτήσεις ανά προορισμό.

The screenshot shows the Kiwi.com search interface. On the left, the search criteria are: One-way, 1 adult, 0 children, departure from Lisbon, arrival in Thessaloniki, and dates from Fri, 23 May to No return. A 'Search' button is visible. Below the search bar, there is a promotional banner for 'Live Boarding Pass' and a confirmation dialog for a trip from Thessaloniki to Bucharest on Sat, 16 Aug.

On the right, a calendar view shows the departure date selection process. The departure date is set to Fri, 23 May. The calendar displays prices for various dates in May and June 2025. The price for the selected date (23 May) is 195 €. The calendar also shows prices for other dates, such as 195 € for 22 May and 227 € for 24 May.

Mon	Tue	Wed	Thu	Fri	Sat	Sun
May 2025						
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	Today 17	18 195 €
19 217 €	20 185 €	21 111 €	22 195 €	23 195 €	24 227 €	25 239 €
26 127 €	27 134 €	28 111 €	29 167 €	30 195 €	31 202 €	
June 2025						
						1 133 €
2 117 €	3 172 €	4 167 €	5 195 €	6 130 €	7 163 €	8 164 €
9 184 €	10 138 €	11 167 €	12 100 €	13 128 €	14 140 €	15 155 €
16 185 €	17 94 €	18 154 €	19 167 €	20 167 €	21 154 €	22 135 €
23 195 €	24 167 €	25 100 €	26 167 €	27 167 €	28 163 €	29 163 €

Οθόνη αναζήτησης και επιλογής ημερομηνίας στην πλατφόρμα Kiwi.com

1.2.3 eDreams ODIGEO

Η eDreams διαχειρίζεται κρατήσεις πτήσεων με έμφαση στην εμπειρία χρήστη από κινητές συσκευές. Η πλατφόρμα διαθέτει προηγμένες δυνατότητες όπως check-in, boarding pass και ειδοποιήσεις μέσα από την εφαρμογή της.

Πώς λειτουργεί:

- Συνδέεται με GDS και APIs αεροπορικών.
- Οι κρατήσεις γίνονται εντός εφαρμογής, με email επιβεβαίωσης και πλήρη διαχείριση κράτησης.
- Υποστηρίζει πληρωμές με κάρτες, ψηφιακά πορτοφόλια και εκπωτικά κουπόνια.

Τι μπορεί να κάνει:

- Δημιουργία σύνθετων δρομολογίων (multi-city).
- Ειδοποιήσεις για καθυστερήσεις ή gate changes.
- Σύστημα loyalty (Prime) με εκπτώσεις.

Αρχική σελίδα και αποτελέσματα αναζήτησης πτήσεων στην πλατφόρμα eDreams

1.2.4 CheapOair

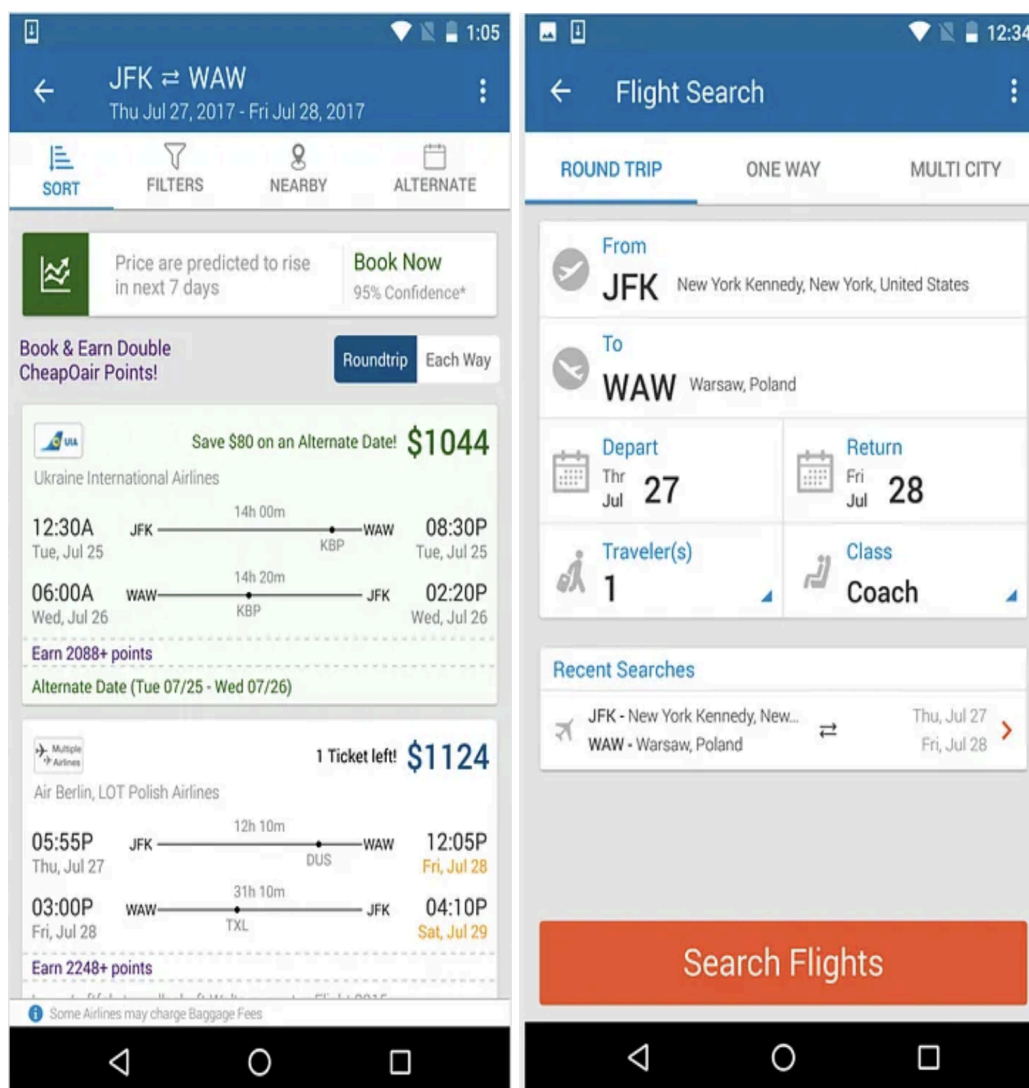
Η CheapOair εξειδικεύεται στη σύγκριση ναύλων για πτήσεις, με έμφαση στις προσφορές και τους «συγκαλυμμένους» (opaque) ναύλους. Συχνά προσφέρει χαμηλότερες τιμές μέσω ιδιόκτητων συμφωνιών με αεροπορικές.

Πώς λειτουργεί:

- Διαθέτει πρόσβαση σε ειδικά συμβόλαια ναύλων.
- Οι κρατήσεις γίνονται μέσω του δικού της συστήματος.
- Έχει δυνατότητα εξυπηρέτησης μέσω τηλεφώνου, live chat ή mobile app.

Τι μπορεί να κάνει:

- Εμφάνιση εναλλακτικών αεροδρομίων κοντά στον προορισμό.
- Ειδικές προσφορές για συνδυασμούς επιστροφής με διαφορετική αεροπορική.
- Δυνατότητα κράτησης μέσω call center.



Οθόνη αναζήτησης και αποτελεσμάτων πτήσεων στην εφαρμογή CheapOair

1.2.5 Kayak

Το Kayak είναι μετα-αναζήτηση OTA που συγκρίνει διαθέσιμα αποτελέσματα από διαφορετικές πηγές και δείχνει τις πιο συμφέρουσες επιλογές σε πραγματικό χρόνο. Δεν ολοκληρώνει κράτηση αλλά παραπέμπει σε τρίτες υπηρεσίες.

Πώς λειτουργεί:

- Χρησιμοποιεί crawling και API σύνδεση με συνεργάτες.
- Παρουσιάζει πολλαπλές επιλογές από Expedia, eDreams, αεροπορικές εταιρείες, κ.λπ.
- Παρέχει διαδραστικά εργαλεία σύγκρισης.

Τι μπορεί να κάνει:

- Χρήση εργαλείου “Price Forecast” (πότε να κάνεις κράτηση).
- Αναζήτηση πτήσεων με ανοιχτές ημερομηνίες.
- Προβολή χάρτη με εύρος περιοχών.

The image shows two screenshots from the Kayak mobile application. The left screenshot displays search results for the route DNR - LON, with filters set to 'Best'. It shows several flight options from Ryanair, with the cheapest being £11. The right screenshot shows search results for the route DNR - STN, with filters set to 'Cheapest'. It shows three flight options: Ryanair for £11, KAYAK for £14, and Kiwi.com for £27. Below the flight options, there is a summary for the Ryanair flight on Thursday, 10 Oct, showing a 1h 10m direct flight from DNR to STN.

Αναζήτηση και σύγκριση πτήσεων την εφαρμογή Kayak

1.2.6 Trip.com

Η Trip.com υποστηρίζει μεγάλο εύρος κρατήσεων πτήσεων σε παγκόσμια κλίμακα και συνεργάζεται με παραδοσιακές αλλά και low-cost αεροπορικές. Έχει ανεπτυγμένο σύστημα ειδοποιήσεων και διαχείρισης κράτησης εντός εφαρμογής.

Πώς λειτουργεί:

- Συνδέεται με GDS και άμεσους παρόχους.
- Ο χρήστης μπορεί να διαχειριστεί την κράτησή του πλήρως από την εφαρμογή.
- Παρέχει ειδοποιήσεις πτήσης σε πραγματικό χρόνο.

Τι μπορεί να κάνει:

- Ειδοποιεί για αλλαγές πτήσεων, καθυστερήσεις, αλλαγές πύλης.
- Παρέχει υποστήριξη σε πολλές γλώσσες.
- Ενσωματώνει loyalty σύστημα και δικά της rewards.

The image displays two screenshots of the Trip.com website. The left screenshot shows the homepage with a search bar and promotional banners. The right screenshot shows the flight search results page for a round-trip from Hong Kong to Seoul.

Αναζήτηση και προβολή αποτελεσμάτων πτήσεων στην πλατφόρμα Trip.com

1.3 Διαφοροποίηση και αναγκαιότητα της εφαρμογής Random Traveller σε σχέση με τις υφιστάμενες OTA πλατφόρμες

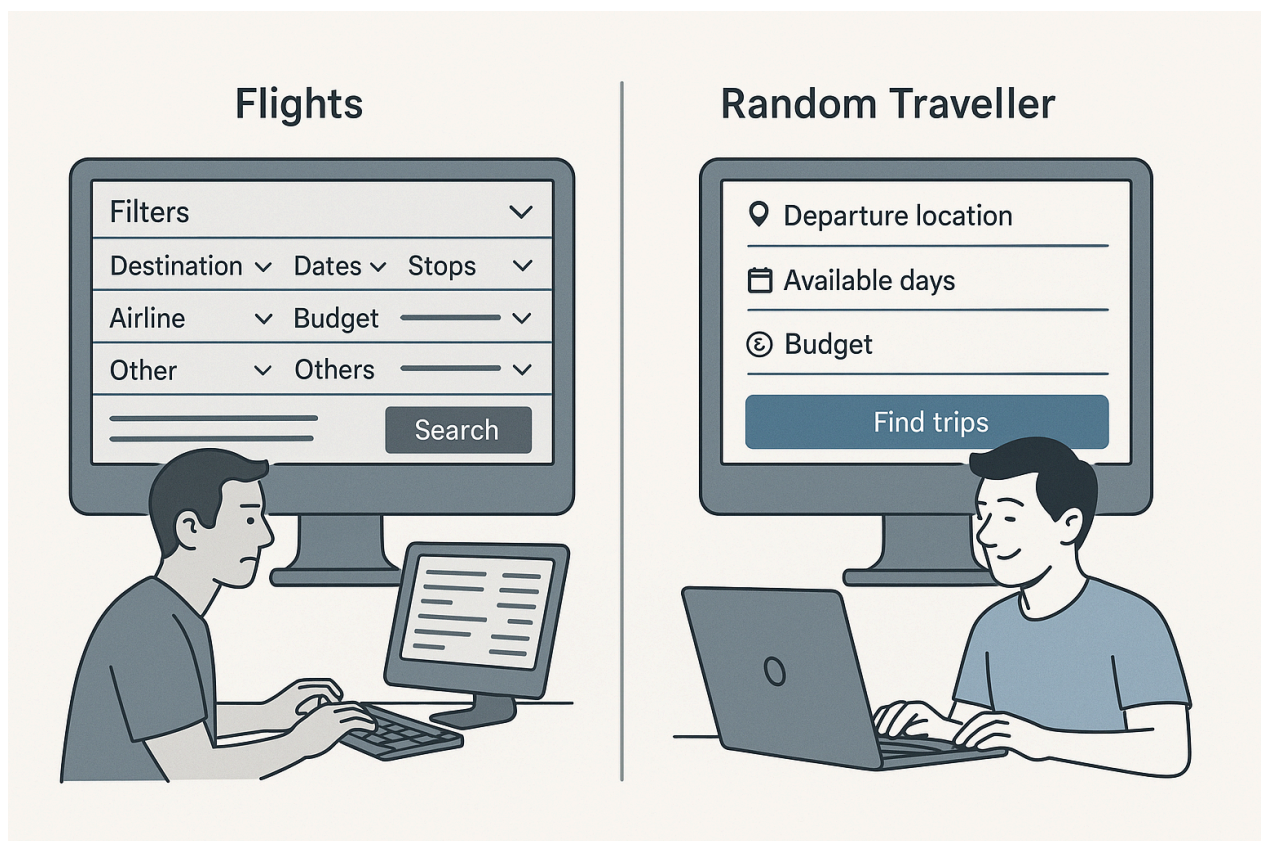
Σε αντίθεση με τις γνωστές πλατφόρμες κρατήσεων, όπως το Skyscanner, το Kiwi.com, το eDreams και άλλες, οι οποίες προσφέρουν πλούσιες δυνατότητες αναζήτησης και σύγκρισης πτήσεων, η πλοήγησή τους συχνά απαιτεί πολυάριθμα φίλτρα, εξειδικευμένες παραμέτρους και γνώση συγκεκριμένων προορισμών ή ημερομηνιών. Ο χρήστης καλείται να λάβει πολλές αποφάσεις πριν καν ξεκινήσει την αναζήτηση, κάτι που καθιστά την εμπειρία πιο σύνθετη και ενίοτε αποθαρρυντική, ειδικά για όσους δεν έχουν σταθερά ταξιδιωτικά σχέδια.

Η εφαρμογή Random Traveller έρχεται να καλύψει ακριβώς αυτό το κενό: έχει σχεδιαστεί αποκλειστικά για ταξιδιώτες περιορισμένου προϋπολογισμού (budget travelers), οι οποίοι αναζητούν ευκαιρίες για φθηνά, σύντομα και αυθόρμητα ταξίδια χωρίς την ανάγκη μακροπρόθεσμου σχεδιασμού. Η λειτουργία της βασίζεται σε μια εξαιρετικά απλή λογική: ο χρήστης εισάγει μόνο τρία

βασικά στοιχεία — σημείο αναχώρησης, διαθέσιμες ημέρες και προϋπολογισμό — και η εφαρμογή του προτείνει άμεσα πιθανούς προορισμούς που πληρούν τα κριτήρια αυτά.

Η καρδιά της εφαρμογής είναι ένας ειδικά σχεδιασμένος αλγόριθμος, ο οποίος ταξινομεί και φιλτράρει τις προτεινόμενες πτήσεις με βάση δύο βασικά κριτήρια: το κόστος και τη διάρκεια του ταξιδιού. Με αυτόν τον τρόπο, απορρίπτονται οι προορισμοί που, παρόλο που μπορεί να είναι οικονομικά προσιτοί, απαιτούν υπερβολικά μεγάλο ταξιδιωτικό χρόνο ή πολλές ενδιάμεσες στάσεις. Αντί να κατακλύζεται ο χρήστης από δεκάδες επιλογές, βλέπει μόνο τις καλύτερες δυνατές προτάσεις που μεγιστοποιούν την αξία του ταξιδιού σε σχέση με το κόστος και το διαθέσιμο χρόνο. Αυτό εξαλείφει τον αποπροσανατολισμό που συχνά προκαλείται από την υπερπληθώρα πληροφοριών στις κλασικές OTA πλατφόρμες και μειώνει σημαντικά το χρόνο λήψης απόφασης.

Παράλληλα, η απλοποιημένη και στοχευμένη εμπειρία χρήσης καθιστά το Random Traveller ιδανικό εργαλείο έμπνευσης για χρήστες που επιθυμούν να ταξιδέψουν άμεσα και έξυπνα, χωρίς να σπαταλούν χρόνο σε σύνθετα μενού ή να απαιτείται προηγούμενη γνώση προορισμών. Για όλους αυτούς τους λόγους, η εφαρμογή Random Traveller δεν αποτελεί απλώς μια ακόμα εναλλακτική στην αγορά, αλλά μια απαραίτητη προσθήκη στο οικοσύστημα των ταξιδιωτικών εφαρμογών, προσφέροντας μια εμπειρία σχεδιασμένη ειδικά για τις ανάγκες του σύγχρονου, αυθόρμητου και οικονομικά συνειδητοποιημένου ταξιδιώτη.



Σύγκριση εμπειρίας χρήστη μεταξύ παραδοσιακής OTA πλατφόρμας και της εφαρμογής Random Traveller

Κεφάλαιο 2ο: Λειτουργικό σύστημα Android

2.1 Εισαγωγή

Σε αυτό το κεφάλαιο αναλύεται η λειτουργική βάση της εφαρμογής «Random Traveller», δηλαδή το λειτουργικό σύστημα Android. Η επιλογή του Android ως πλατφόρμα υλοποίησης δεν αποτελεί απλώς τεχνική απόφαση, αλλά είναι το αποτέλεσμα αξιολόγησης πολλών παραμέτρων: της ευρείας διάδοσής του, της προσβασιμότητάς του, της ποικιλίας συσκευών που το υποστηρίζουν και της δυναμικής του ανάπτυξης τα τελευταία χρόνια.

Το Android χρησιμοποιείται σήμερα σε περισσότερες από 3 δισεκατομμύρια ενεργές συσκευές παγκοσμίως. Αυτό σημαίνει ότι αποτελεί ένα κρίσιμο μέσο διάδοσης οποιασδήποτε εφαρμογής στο ευρύ κοινό. Από τη στιγμή που η εφαρμογή μας απευθύνεται σε άτομα που ενδιαφέρονται να ανακαλύψουν τυχαίους προορισμούς και να εμπνευστούν νέες ταξιδιωτικές εμπειρίες, η χρήση μιας καθολικά αναγνωρισμένης πλατφόρμας όπως το Android αυξάνει σημαντικά τις πιθανότητες επιτυχίας και υιοθέτησης της εφαρμογής.

Επιπλέον, το Android δίνει τη δυνατότητα για συνεχή αναβάθμιση και προσαρμογή στις τεχνολογικές εξελίξεις. Η ίδια η Google, μέσω του Android ecosystem, προάγει τις βέλτιστες πρακτικές σχεδίασης διεπαφής και εμπειρίας χρήστη (User Experience – UX). Η προσβασιμότητα, η συμβατότητα με διαφορετικά μεγέθη οθόνης και η υποστήριξη πολλαπλών γλωσσών είναι χαρακτηριστικά ενσωματωμένα στο λειτουργικό σύστημα.

Για την ομάδα ανάπτυξης, το Android προσφέρει ένα πλούσιο σύνολο εργαλείων που βοηθούν στην υλοποίηση εφαρμογών με υψηλό επίπεδο λειτουργικότητας. Παρόλο που δεν θα αναλύσουμε εδώ τα τεχνικά εργαλεία αυτά, είναι σημαντικό να αναγνωρίσουμε ότι αποτελούν σημαντικό πλεονέκτημα για την υλοποίηση και συντήρηση μιας εφαρμογής με προοπτική εξέλιξης.

Η επιλογή του Android συνεπώς δεν βασίζεται μόνο στις δυνατότητες του ως λειτουργικό σύστημα, αλλά και στο οικοσύστημα που το περιβάλλει: κοινότητα προγραμματιστών, διαθέσιμη τεκμηρίωση, συνεχής υποστήριξη και ανοιχτό περιβάλλον ανάπτυξης. Όλοι αυτοί οι παράγοντες καθιστούν το Android την πιο προσιτή, αξιόπιστη και εξελίξιμη λύση για την ανάπτυξη της συγκεκριμένης εφαρμογής.

2.2 Τι είναι το Android

Το Android είναι ένα λειτουργικό σύστημα που έχει σχεδιαστεί με σκοπό την υποστήριξη εφαρμογών σε έξυπνες φορητές συσκευές. Υποστηρίζει τη δημιουργία περιβαλλόντων διεπαφής χρήστη (UI), διαχείριση δεδομένων, επικοινωνία με το διαδίκτυο, πρόσβαση σε ενσωματωμένες λειτουργίες όπως κάμερα, GPS, μικρόφωνο και αισθητήρες κίνησης. Επιτρέπει επίσης την εκτέλεση πολλαπλών εφαρμογών ταυτόχρονα, διατηρώντας ομαλή τη λειτουργία της συσκευής.

Το Android βασίζεται σε τροποποιημένο πυρήνα Linux, γεγονός που του προσδίδει σταθερότητα και δυνατότητες χαμηλού επιπέδου ελέγχου πόρων. Παράλληλα, το γραφικό περιβάλλον που παρέχει επιτρέπει τη δημιουργία εφαρμογών που είναι εύχρηστες και αισθητικά ευχάριστες για τον τελικό χρήστη.

2.3 Λογισμικού ανοιχτού κώδικα

Το Android αποτελεί μια πλατφόρμα ανοιχτού κώδικα που προσφέρεται δωρεάν για χρήση. Το λειτουργικό σύστημα είναι προστατευμένο με την άδεια GNU General Public License v2, η οποία απαιτεί οι τροποποιήσεις από τρίτους να παραμένουν εντός των προτύπων ανοιχτού κώδικα. Αυτό επιτρέπει σε οποιονδήποτε να αντιγράψει τον αρχικό κώδικα, να τον προσαρμόσει και να κυκλοφορήσει τη δική του έκδοση με πρόσθετα χαρακτηριστικά, υπό την προϋπόθεση ότι θα συνεχίσει να τηρεί τις άδειες ανοιχτού κώδικα.

Το περιβάλλον ανάπτυξης του Android διατίθεται με την άδεια Apache Software License, που επιτρέπει την προσθήκη κλειστού κώδικα. Έτσι, μια εταιρεία μπορεί να αξιοποιήσει την πλατφόρμα και να ενσωματώσει επιπλέον λειτουργίες χωρίς να χρειάζεται να διαθέσει το τελικό προϊόν ως ανοιχτού κώδικα. Αυτό είναι συνηθισμένο, καθώς οι εταιρείες επιδιώκουν να ξεχωρίσουν από τους ανταγωνιστές τους.

Παρά τις αλλαγές, το λειτουργικό σύστημα παραμένει Android. Οι προσαρμογές αυτές γίνονται σε ανώτερο επίπεδο και συνήθως εστιάζουν στη βελτίωση της διεπαφής χρήστη (UI).

2.4 Η Αρχιτεκτονική του Android

Εξερευνώντας την αρχιτεκτονική του Android, από κάτω προς τα πάνω, θα βρούμε στο πρώτο στρώμα έναν Linux Kernel. Ο πυρήνας αυτός, είναι υπεύθυνος για τους οδηγούς (drivers) του υλικού (Wi-Fi, USB, κάμερα, οθόνη, ηχείο κ.α) το οποίο απαρτίζει το μηχάνημα στο οποίο τρέχει. Ο πυρήνας, εκτός από υλικό, είναι υπεύθυνος και για την κοινόχρηστη μνήμη καθώς και για την διαχείριση ενέργειας της συσκευής.

Αναλύοντας την αρχιτεκτονική του Android από τη βάση προς την κορυφή, το πρώτο επίπεδο αποτελείται από τον πυρήνα Linux. Αυτός ο πυρήνας είναι υπεύθυνος για τους οδηγούς (drivers) του υλικού, όπως το Wi-Fi, το USB, η κάμερα, η οθόνη και το ηχείο, που συνθέτουν τη συσκευή. Εκτός από τη διαχείριση του υλικού, ο πυρήνας φροντίζει για την κοινόχρηστη μνήμη και τη διαχείριση ενέργειας της συσκευής.

Αμέσως επάνω βρίσκεται το Hardware Abstraction Layer, το οποίο περιλαμβάνει διάφορες βιβλιοθήκες του Android που υλοποιούν λειτουργίες σχετικές με το υλικό, όπως η οθόνη, η κάμερα, το Bluetooth και οι αισθητήρες. Όταν γίνεται μια κλήση για τη χρήση ενός κομματιού υλικού, το Android φορτώνει την αντίστοιχη βιβλιοθήκη για να επιτρέψει τη χρήση του.

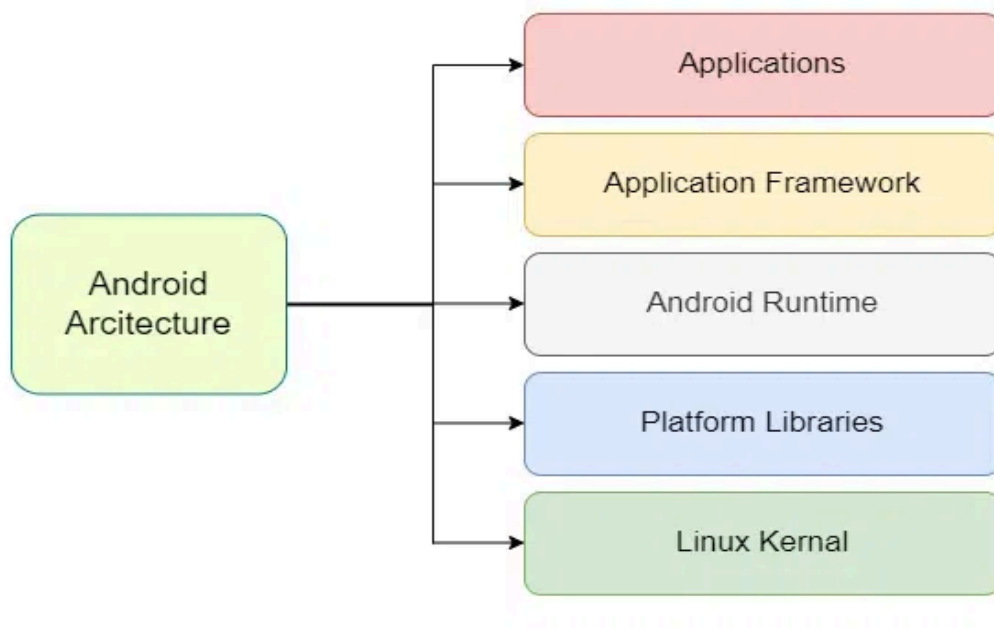
Στο επόμενο επίπεδο βρίσκονται οι native βιβλιοθήκες C/C++, που επιτρέπουν στους προγραμματιστές να έχουν πρόσβαση σε φυσικά συστατικά της συσκευής, όπως οι αισθητήρες.

Το Android Runtime, το οποίο είναι παράλληλο με τις native βιβλιοθήκες C/C++, περιλαμβάνει το Dalvik Virtual Machine, μια ειδική Java Virtual Machine σχεδιασμένη για το Android. Το Dalvik VM χρησιμοποιεί τις βασικές βιβλιοθήκες του πυρήνα Linux, όπως η διαχείριση μνήμης, και επιτρέπει σε κάθε εφαρμογή Android να εκτελείται σε δική της διεργασία μέσα στην εικονική μηχανή Dalvik.

Το Java API Framework βρίσκεται στο επόμενο επίπεδο και προσφέρει υπηρεσίες υψηλού επιπέδου μέσω κλάσεων Java, που μπορούν να χρησιμοποιηθούν από προγραμματιστές εφαρμογών. Περιλαμβάνει υπηρεσίες όπως:

- Resource Manager: Παρέχει πρόσβαση σε μη κωδικοποιημένους πόρους, όπως γραφικά και αρχεία διάταξης.

- Content Provider: Διευκολύνει την ανταλλαγή πληροφοριών μεταξύ εφαρμογών.
- Activity Manager: Διαχειρίζεται τον κύκλο ζωής των δραστηριοτήτων και την πλοήγηση μεταξύ τους.
- Notification Manager: Επιτρέπει την εμφάνιση ειδοποιήσεων στη γραμμή κατάστασης.
- Τέλος, το Android συνοδεύεται από ένα σύνολο βασικών εφαρμογών, όπως κλήσεις, μηνύματα (SMS), ημερολόγια και επαφές, που αποτελούν το ανώτατο επίπεδο της αρχιτεκτονικής του Android.



Διάγραμμα αρχιτεκτονικής του Android

2.5 Επίλογος

Σε αυτό το κεφάλαιο, προσφέρθηκε μια εκτενής εισαγωγή στο λειτουργικό σύστημα Android, με στόχο να εξοπλίσει τον αναγνώστη με μια σαφή και ολοκληρωμένη κατανόηση των βασικών του αρχών. Αυτή η θεμελιώδης γνώση είναι απαραίτητη για την καλύτερη κατανόηση των πιο σύνθετων και εξειδικευμένων θεμάτων που θα αναλυθούν στα επόμενα κεφάλαια, επιτρέποντας στον αναγνώστη να εκτιμήσει πλήρως τις δυνατότητες και τις εφαρμογές του Android στον σύγχρονο τεχνολογικό κόσμο.

Κεφάλαιο 3ο: Είδη Android εφαρμογών

3.1 Εισαγωγή

Σε αυτό το κεφάλαιο, θα παρουσιαστούν συνοπτικά οι διάφοροι τύποι εφαρμογών που μπορούν να αναπτυχθούν για το Android. Αυτή η γνώση είναι ουσιώδης για οποιονδήποτε επιθυμεί να εισέλθει στον τομέα του προγραμματισμού εφαρμογών για την πλατφόρμα Android, καθώς παρέχει μια βασική κατανόηση των δυνατοτήτων και των επιλογών που προσφέρει το οικοσύστημα του Android.

3.2 Εφαρμογές στο προσκήνιο

Αυτός είναι ο πιο κοινός τύπος εφαρμογών. Ο κύκλος ζωής τους ξεκινά όταν ο χρήστης τις ανοίγει, εμφανίζοντάς τες στην οθόνη της συσκευής. Όταν ο χρήστης μεταβεί σε άλλη εφαρμογή ή επιστρέψει στην αρχική οθόνη, η εφαρμογή δεν κλείνει αλλά παραμένει ενεργή στο παρασκήνιο σε κατάσταση αναστολής. Η συμπεριφορά της εφαρμογής κατά τη διάρκεια αυτής της αναστολής και μέχρι να επανέλθει στο προσκήνιο καθορίζεται από τον προγραμματιστή, ο οποίος μπορεί να διαμορφώσει πώς θα αντιδράσει η εφαρμογή σε αυτές τις αλλαγές κατάστασης.

3.3 Υβριδικές εφαρμογές

Αυτές οι εφαρμογές λειτουργούν τόσο στο προσκήνιο όσο και στο παρασκήνιο. Απαιτούν αλληλεπίδραση με τον χρήστη, ενώ ταυτόχρονα πρέπει να είναι σε θέση να ανταποκρίνονται σε γεγονότα όταν εκτελούνται στο παρασκήνιο. Σε τέτοιες εφαρμογές, ο προγραμματιστής πρέπει να διαμορφώσει τη λειτουργία τους έτσι ώστε να μπορούν να λαμβάνουν μηνύματα ενώ βρίσκονται στο παρασκήνιο και να ειδοποιούν τον χρήστη όταν είναι απαραίτητο για κάποιο σημαντικό γεγονός.

3.4 Widgets

Τα widgets αποτελούν έναν τύπο εφαρμογών που λειτουργούν ως διαδραστικά γραφικά στοιχεία στην αρχική οθόνη του χρήστη. Σκοπός τους είναι να παρέχουν λειτουργίες ή πληροφορίες χωρίς να απαιτείται η πλήρης εκκίνηση της εφαρμογής από τον χρήστη. Ένα χαρακτηριστικό παράδειγμα widget είναι το ρολόι που εμφανίζεται στην αρχική οθόνη, προσφέροντας άμεση πρόσβαση στην ώρα χωρίς την ανάγκη ανοίγματος μιας ξεχωριστής εφαρμογής.

3.5 Εφαρμογές Android Wear (Smartwatches)

Οι εφαρμογές Android Wear επεκτείνουν την εμπειρία Android σε smartwatches και άλλες φορητές συσκευές, προσφέροντας γρήγορες αλληλεπιδράσεις και ειδοποιήσεις, υποστήριξη φωνητικών εντολών, αξιοποίηση αισθητήρων όπως καρδιακού παλμού και GPS, ανάπτυξη με Android SDK & Wear OS API, δυνατότητα αυτόνομης λειτουργίας ή συνοδευτικής εφαρμογής, και εφαρμογές για fitness, μουσική, πλοήγηση και έξυπνο σπίτι.

3.6 Εφαρμογές Android TV

Οι εφαρμογές Android Auto επεκτείνουν τη λειτουργικότητα του Android στο αυτοκίνητο, προσφέροντας μια ασφαλή και βολική εμπειρία οδήγησης. Σχεδιασμένες για να ελαχιστοποιούν τους περισπασμούς, επιτρέπουν στους οδηγούς να χρησιμοποιούν βασικές λειτουργίες όπως πλοήγηση, μουσική, τηλεφωνικές κλήσεις και μηνύματα μέσω φωνητικών εντολών ή απλοποιημένων διεπαφών, συμβάλλοντας στην ασφαλέστερη οδήγηση.

Κεφάλαιο 4ο: Κύκλος ζωής εφαρμογών Android

4.1 Εισαγωγή

Ο κύκλος ζωής των εφαρμογών Android περιγράφει τη διαδρομή που ακολουθεί μια εφαρμογή από τη στιγμή που ξεκινά να εκτελείται μέχρι να τερματιστεί πλήρως. Η κατανόηση αυτού του κύκλου είναι κρίσιμη για τη σωστή διαχείριση των πόρων της συσκευής, τη διατήρηση της κατάστασης της εφαρμογής και την προσφορά σταθερής και αξιόπιστης εμπειρίας στον χρήστη. Ένα από τα βασικά χαρακτηριστικά του Android είναι η διαχείριση της μνήμης και των διεργασιών με τρόπο δυναμικό, ανάλογα με τη χρήση και τη διαθεσιμότητα πόρων στη συσκευή.

Η γνώση του κύκλου ζωής επιτρέπει στους δημιουργούς εφαρμογών να σχεδιάσουν στρατηγικές για αποθήκευση δεδομένων, ανανέωση περιεχομένου και εξοικονόμηση ενέργειας. Αυτό είναι ιδιαίτερα σημαντικό για εφαρμογές όπως η «Random Traveller», που στηρίζονται σε συνεχή ροή πληροφορίας και προσωποποιημένη εμπειρία.

Αντί να περιγράψουμε το τεχνικό κομμάτι του κύκλου ζωής μέσω κώδικα, όπως συνηθίζεται, θα χρησιμοποιήσουμε ψευδοκώδικα και περιγραφικές επεξηγήσεις για να αναλύσουμε τη λειτουργία του, προσαρμόζοντας την παρουσίαση στις οδηγίες της παρούσας πτυχιακής.

4.2 Καταστάσεις

Μια εφαρμογή Android περνάει από διαφορετικά στάδια (καταστάσεις) κατά την εκτέλεσή της. Οι σημαντικότερες είναι:

Created: Η εφαρμογή δημιουργείται για πρώτη φορά. Αρχικοποιούνται βασικά δεδομένα και πόροι, όπως μεταβλητές, δομές δεδομένων και το περιβάλλον χρήστη (UI). Είναι το στάδιο στο οποίο τίθενται τα θεμέλια για την εκκίνηση της εφαρμογής.

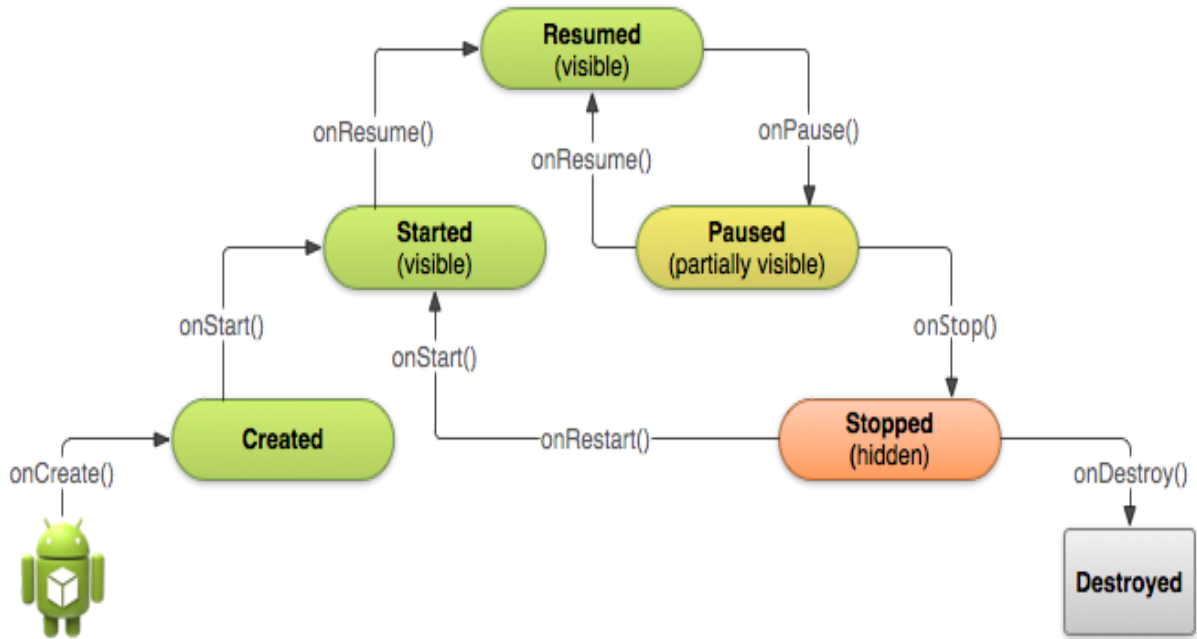
Started: Η εφαρμογή είναι πλέον ορατή στον χρήστη, ωστόσο δεν είναι ακόμη «ενεργή» για είσοδο δεδομένων από τον χρήστη. Είναι η φάση κατά την οποία το σύστημα την εμφανίζει στην οθόνη αλλά δεν έχει αποκτήσει ακόμη εστίαση.

Resumed: Η εφαρμογή είναι πλήρως ενεργή και έτοιμη για αλληλεπίδραση με τον χρήστη. Σε αυτή την κατάσταση ο χρήστης μπορεί να αγγίξει κουμπιά, να εισάγει δεδομένα, να πραγματοποιήσει αναζητήσεις κ.λπ.

Paused: Η εφαρμογή δεν είναι ενεργή, καθώς ο χρήστης μεταβαίνει προσωρινά σε άλλη εφαρμογή ή δέχεται κάποια ειδοποίηση. Ωστόσο, η εφαρμογή δεν έχει κλείσει και παραμένει στη μνήμη.

Stopped: Η εφαρμογή δεν είναι πλέον ορατή και μπορεί να απελευθερώσει πόρους. Αυτό συνήθως γίνεται όταν ο χρήστης κλείνει την εφαρμογή ή ανοίγει μια άλλη για μεγάλο χρονικό διάστημα.

Destroyed: Η εφαρμογή τερματίζεται οριστικά και διαγράφονται όλα τα προσωρινά ή αποθηκευμένα δεδομένα από τη μνήμη RAM. Είναι η φάση του πλήρους καθαρισμού.



[Απεικόνιση](#) κύκλου ζωής Android Activity

4.3 Χρήση στην εφαρμογή Random Traveller

Όταν ο χρήστης ανοίγει για πρώτη φορά την εφαρμογή, εκτελείται η **onCreate()** όπου αρχικοποιούνται όλα τα δεδομένα, συνδέεται η βάση δεδομένων Firebase και φορτώνεται το περιβάλλον χρήστη.

Στο **onStart()** και **onResume()** ελέγχεται αν υπάρχει ήδη κάποια αναζήτηση αποθηκευμένη στη βάση και αν ναι, προτείνεται ξανά στον χρήστη, ενισχύοντας τη συνεχή και προσωποποιημένη εμπειρία.

Όταν ο χρήστης μεταβεί προσωρινά σε άλλη εφαρμογή, ενεργοποιείται η **onPause()** που αποθηκεύει την πρόοδο και τα προσωρινά δεδομένα σε local storage ή cloud για να αποφευχθεί απώλεια πληροφορίας.

Αν ο χρήστης αφήσει την εφαρμογή κλειστή για μεγάλο χρονικό διάστημα, η **onStop()** φροντίζει να ελευθερωθούν μη απαραίτητοι πόροι όπως ζωντανές συνδέσεις με API, ακροατές ειδοποιήσεων ή προσωρινές δομές στη μνήμη.

Τέλος, όταν η εφαρμογή κλείσει οριστικά, η **onDestroy()** φροντίζει για το τελικό καθάρισμα μνήμης και την αποσύνδεση των υπηρεσιών.

Με αυτό τον τρόπο, η εφαρμογή «Random Traveller» αξιοποιεί στο έπακρο τον κύκλο ζωής του Android ώστε να είναι αποτελεσματική, ελαφριά στη χρήση και φιλική προς τον τελικό χρήστη. Ο σχεδιασμός αυτός προσδίδει στην εφαρμογή μια αίσθηση επαγγελματισμού και σταθερότητας, που αποτελεί κρίσιμο σημείο διαφοροποίησης σε σχέση με πολλές αντίστοιχες εφαρμογές.

4.4 Σύνοψη και Συμπεράσματα

Ο κύκλος ζωής των εφαρμογών Android αποτελεί βασική αρχή για τη σωστή ανάπτυξη και διαχείριση mobile εφαρμογών. Μέσα από την κατανόηση και εφαρμογή των σταδίων αυτού του κύκλου,

διασφαλίζεται η βελτιστοποίηση της απόδοσης της εφαρμογής, η αποδοτική χρήση των πόρων της συσκευής και η θετική εμπειρία χρήστη.

Η παρούσα εργασία ανέδειξε πώς κάθε στάδιο του κύκλου ζωής αξιοποιείται στρατηγικά στην εφαρμογή «Random Traveller», ώστε να διασφαλιστεί η αποθήκευση της προόδου, η ανάκτηση των δεδομένων και η αποτροπή περιττής κατανάλωσης πόρων. Η υιοθέτηση αυτού του προτύπου σχεδιασμού ενισχύει τη σταθερότητα, την επεκτασιμότητα και τη συνολική ποιότητα της εφαρμογής.

Επιπλέον, η κατανόηση του κύκλου ζωής συνέβαλε ουσιαστικά στη δημιουργία μιας ροής εργασιών εντός της εφαρμογής που ευθυγραμμίζεται με τις προσδοκίες του τελικού χρήστη. Η διατήρηση της κατάστασης της εφαρμογής, η ανάκτηση προηγούμενων επιλογών και η αποτροπή ανεπιθύμητων σφαλμάτων λόγω κακής διαχείρισης μνήμης είναι αποτελέσματα που προέκυψαν απευθείας από τον σχεδιασμό με βάση το μοντέλο κύκλου ζωής.

Τέλος, ο κύκλος ζωής συνδέεται άμεσα και με άλλες τεχνολογικές επιλογές, όπως η χρήση του Firebase για αποθήκευση κατάστασης και ελέγχους συγχρονισμού, γεγονός που δείχνει πως η θεωρία και η πράξη συνεργάζονται στενά στη σύγχρονη ανάπτυξη εφαρμογών. Αυτό υπογραμμίζει τη σημασία της κατανόησης των θεμελιωδών αρχών του Android, όχι μόνο για λειτουργικούς λόγους, αλλά και για την επίτευξη υψηλής χρηστικότητας και εμπειρίας χρήστη.

Συνοψίζοντας, η εφαρμογή του κύκλου ζωής σε πραγματικό σενάριο όπως η «Random Traveller» αποδεικνύει τη θεωρητική σημασία του μοντέλου αλλά και την πρακτική του αξία. Κάθε στάδιο, από τη δημιουργία έως τον τερματισμό, λειτουργεί ως κρίκος μιας αλυσίδας που οδηγεί σε εφαρμογές υψηλών προδιαγραφών, ικανές να ανταποκριθούν στις προσδοκίες του σύγχρονου χρήστη.

Κεφάλαιο 5ο: Συστατικά στοιχεία εφαρμογών Android

5.1 Εισαγωγή

Στο παρόν κεφάλαιο αναλύονται τα κύρια συστατικά στοιχεία που συγκροτούν μία εφαρμογή Android. Η κατανόηση αυτών των στοιχείων είναι θεμελιώδης για την ανάπτυξη και τη δομή οποιασδήποτε εφαρμογής, καθώς προσδιορίζουν τον τρόπο αλληλεπίδρασης του χρήστη, τη διαχείριση της πληροφορίας, την εκτέλεση διεργασιών στο παρασκήνιο και τη γενικότερη συμπεριφορά της εφαρμογής.

Τα συστατικά αυτά λειτουργούν ως τα βασικά δομικά στοιχεία του οικοσυστήματος Android και είναι απαραίτητα για τον ορισμό της αρχιτεκτονικής μιας εφαρμογής. Κάθε ένα εξυπηρετεί έναν συγκεκριμένο ρόλο – από τη διαχείριση του γραφικού περιβάλλοντος, μέχρι την ασύγχρονη επεξεργασία δεδομένων και την επικοινωνία μεταξύ διαφορετικών ενοτήτων της εφαρμογής. Η βαθιά κατανόηση αυτών των στοιχείων αποτελεί προϋπόθεση για την επίτευξη σταθερότητας, επεκτασιμότητας και αποδοτικότητας κατά την ανάπτυξη εφαρμογών.

Ιδιαίτερα στη σύγχρονη εποχή όπου η πολυπλοκότητα των mobile εφαρμογών αυξάνεται, η σωστή διαχείριση και αξιοποίηση των συστατικών στοιχείων προσφέρει τη δυνατότητα υλοποίησης λειτουργιών υψηλού επιπέδου, ενώ παράλληλα εξασφαλίζει καλή εμπειρία χρήστη και χαμηλή κατανάλωση πόρων. Μέσω των επόμενων ενοτήτων θα εξετάσουμε ένα προς ένα αυτά τα στοιχεία, με στόχο να αναδειχθεί η συμβολή τους στη συνολική λειτουργικότητα της εφαρμογής "Random Traveller".

5.2 Manifest

Το αρχείο "AndroidManifest.xml" αποτελεί το σημείο έναρξης για κάθε εφαρμογή Android. Εδώ δηλώνονται όλα τα βασικά στοιχεία της εφαρμογής, όπως τα Activities, τα Permissions, τα Services, τα Broadcast Receivers και άλλες κρίσιμες παραμέτρους.

Το αρχείο αυτό παίζει έναν καθοριστικό ρόλο στη διαμόρφωση της συμπεριφοράς της εφαρμογής, καθώς προσδιορίζει την κεντρική Activity, τις απαιτούμενες άδειες για πρόσβαση σε πόρους του συστήματος (όπως κάμερα, διαδίκτυο, τοποθεσία), καθώς και τα εξωτερικά APIs με τα οποία αλληλεπιδρά. Επιπλέον, επιτρέπει στον προγραμματιστή να δηλώσει ποια Activities μπορούν να είναι προσβάσιμες από άλλες εφαρμογές ή από το ίδιο το σύστημα.

Χρήση στην εφαρμογή: Το Manifest της εφαρμογής "Random Traveller" δηλώνει ως αρχική την Activity "SearchScreen". Επιπλέον, καθορίζονται οι άδειες πρόσβασης στο διαδίκτυο, ώστε να μπορεί η εφαρμογή να επικοινωνεί με το API της Kiwi για αναζήτηση πτήσεων. Το αρχείο περιλαμβάνει ακόμη δηλώσεις για τις βιβλιοθήκες Firebase και Hilt, οι οποίες ενσωματώνονται στο έργο μέσω εξαρτήσεων.

5.3 Activity

Η Activity αποτελεί μία από τις πιο θεμελιώδεις μονάδες μιας εφαρμογής Android. Αντιπροσωπεύει μία οθόνη με την οποία αλληλεπιδρά ο χρήστης. Κάθε Activity έχει τον δικό της κύκλο ζωής και επιτρέπει στον χρήστη να εκτελέσει συγκεκριμένες ενέργειες.

Στη σύγχρονη ανάπτυξη Android, πολλές παραδοσιακές Activities έχουν αντικατασταθεί με Jetpack Compose Screens, ωστόσο η θεμελιώδης έννοια παραμένει ίδια. Κάθε βασικό λειτουργικό κομμάτι της εφαρμογής (π.χ. αρχική σελίδα, αποτελέσματα πτήσεων, λεπτομέρειες κράτησης) υλοποιείται ως Activity ή Compose οθόνη.

Χρήση στην εφαρμογή: Η κύρια Activity της εφαρμογής είναι η "SearchActivity", στην οποία ο χρήστης εισάγει το budget, αεροδρόμιο και τις επιθυμητές ημέρες ταξιδιού. Από αυτή την Activity ξεκινά η αναζήτηση και δρομολογείται η μετάβαση στην επόμενη οθόνη.

Ψευδοκώδικας:

```
Δημιουργία Activity "SearchScreen"  
Εισαγωγή στοιχείων: budget, ημέρες, αεροδρόμιο  
Ενεργοποίηση button αναζήτησης
```

5.4 Intents

Τα Intents είναι βασικά εργαλεία επικοινωνίας μεταξύ των διαφόρων Activities ή ακόμη και μεταξύ διαφορετικών εφαρμογών. Ένα Intent μπορεί να μεταφέρει πληροφορίες, να ενεργοποιήσει άλλες εφαρμογές ή υπηρεσίες ή να αλλάξει Activity μέσα στην ίδια εφαρμογή.

Χρήση στην εφαρμογή: Όταν ο χρήστης ολοκληρώσει την αναζήτηση, ένα Intent δημιουργείται για να μεταφέρει τα δεδομένα στην οθόνη των αποτελεσμάτων. Το Intent περιλαμβάνει παραμέτρους όπως το budget, τον αριθμό των ημερών και τους πιθανούς προορισμούς.

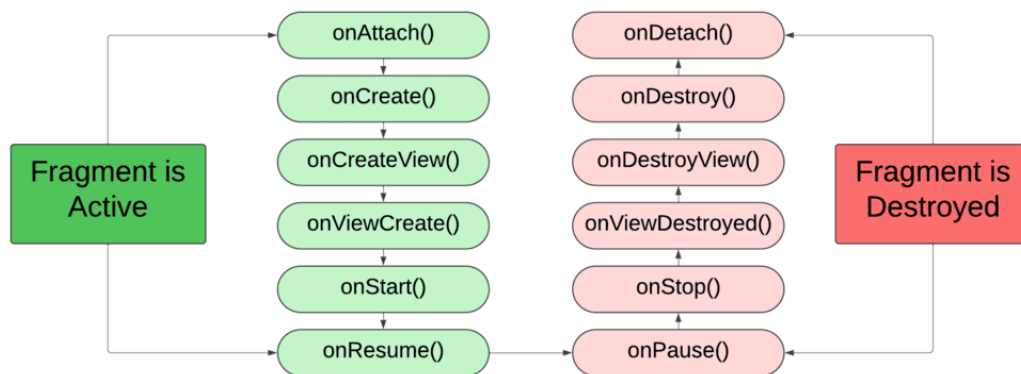
Ψευδοκώδικας:

```
intent = νέο Intent -> ResultsActivity  
intent.βάλει("προορισμοί", λίστα_πτήσεων)  
ξεκίνησε Activity: ResultsActivity
```

5.5 Fragments

Τα Fragments είναι επαναχρησιμοποιήσιμα τμήματα του περιβάλλοντος χρήστη τα οποία ανήκουν σε μία Activity. Επιτρέπουν τη διαίρεση μιας Activity σε μικρότερες, διαχειρίσιμες ενότητες. Ένα Fragment έχει δικό του κύκλο ζωής, και μπορεί να προστεθεί, να αφαιρεθεί ή να αντικατασταθεί δυναμικά.

Χρήση στην εφαρμογή: Αν και η εφαρμογή μας στηρίζεται κυρίως στο Jetpack Compose και δεν χρησιμοποιεί παραδοσιακά Fragments, η ιδέα της δομημένης και επαναχρησιμοποιούμενης διεπαφής υλοποιείται μέσα από ξεχωριστά Compose components. Σε παλιότερες υλοποιήσεις, η αναζήτηση ή το παιχνίδι μενού πλοήγησης θα μπορούσαν να έχουν μορφή Fragment.



Κύκλος ζωής ενός Fragment σε Android εφαρμογή

5.6 Threads

Τα Threads είναι μια βασική έννοια στον προγραμματισμό εφαρμογών Android και αποτελούν τον τρόπο με τον οποίο μπορεί να εκτελείται ταυτόχρονα περισσότερος από ένας κώδικας μέσα στην ίδια εφαρμογή. Στο λειτουργικό σύστημα Android, κάθε εφαρμογή εκτελείται εξ ορισμού σε ένα κύριο thread, το οποίο είναι υπεύθυνο για τη διαχείριση του περιβάλλοντος χρήστη (UI thread).

Ωστόσο, η εκτέλεση εντατικών εργασιών στο κύριο thread, όπως η πρόσβαση σε δίκτυο ή η επεξεργασία δεδομένων, μπορεί να προκαλέσει καθυστερήσεις και να κάνει την εφαρμογή να φαίνεται ότι "παγώνει". Για τον λόγο αυτό, χρησιμοποιούνται επιπλέον Threads ώστε τέτοιου είδους εργασίες να εκτελούνται στο παρασκήνιο, διατηρώντας την εφαρμογή ομαλή και ανταποκρίσιμη.

Χρήση στην εφαρμογή: Η εφαρμογή "Random Traveller" κάνει χρήση Threads για να διαχειριστεί ασύγχρονες κλήσεις σε APIs, όπως η ανάκτηση δεδομένων από την Kiwi. Αυτό αποτρέπει την εμφάνιση σφαλμάτων τύπου ANR (Application Not Responding) και επιτρέπει στο UI να παραμένει ενεργό κατά τη διάρκεια της αναζήτησης πτήσεων.

5.7 AsyncTask

Η κλάση AsyncTask ήταν ένα από τα πρώτα εργαλεία στο Android για την εκτέλεση ασύγχρονων εργασιών. Αν και έχει πλέον αποσυρθεί επίσημα υπέρ νεότερων βιβλιοθηκών όπως τα Kotlin Coroutines ή το WorkManager, αποτελεί σημαντικό παράδειγμα κατανόησης της βασικής αρχής του διαχωρισμού των διεργασιών σε background και UI thread.

Μια AsyncTask αποτελείται από τρία βασικά στάδια:

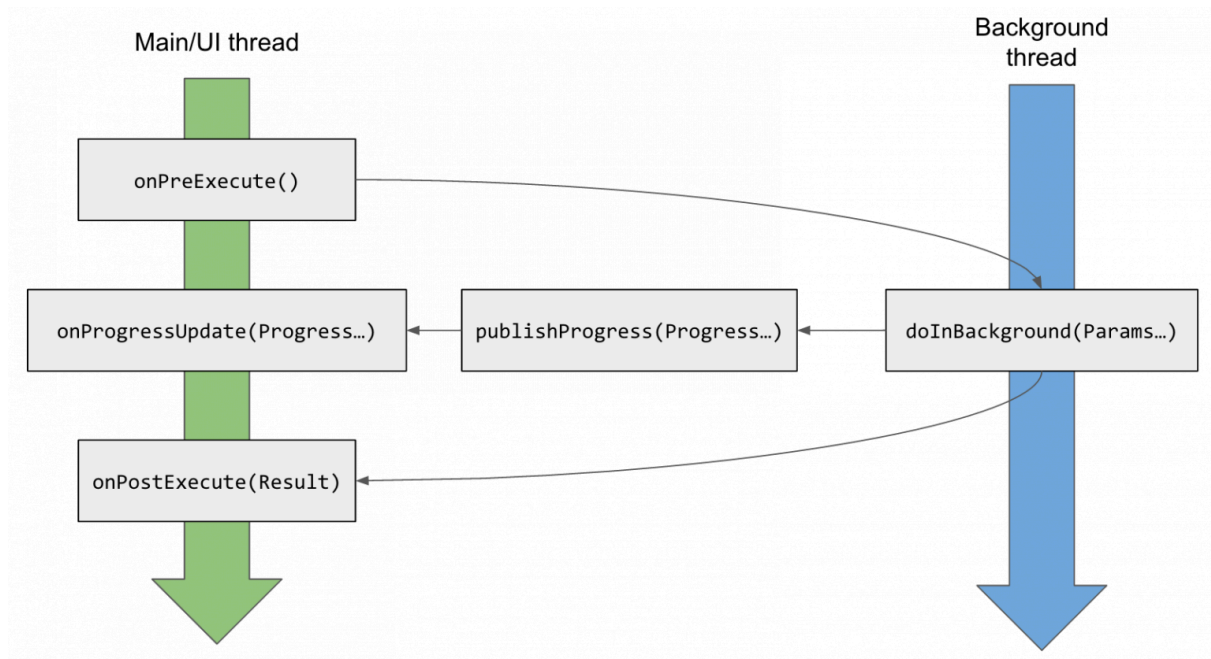
- `onPreExecute()`: Εκτελείται στο UI thread πριν ξεκινήσει η ασύγχρονη διεργασία.
- `doInBackground()`: Εκτελείται σε ξεχωριστό thread και περιλαμβάνει την κύρια εργασία (π.χ. δικτυακή κλήση).
- `onPostExecute()`: Εκτελείται στο UI thread με τα αποτελέσματα του background task.

Χρήση στην εφαρμογή: Αν και η εφαρμογή μας χρησιμοποιεί πιο σύγχρονες τεχνικές, η λογική της AsyncTask παραμένει επίκαιρη, καθώς οι λειτουργίες αναζήτησης εκτελούνται ασύγχρονα ώστε να διατηρείται ομαλή η διεπαφή.

Ψευδοκώδικας:

```

Κάλεσε API -> στο παρασκήνιο
Περίμενε απάντηση
Όταν έρθει η απάντηση -> εμφάνισε δεδομένα στο UI
    
```



Ροή εκτέλεσης της κλάσης AsyncTask – Επικοινωνία μεταξύ UI και background thread μέσω βασικών μεθόδων

5.8 Layouts

Τα Layouts καθορίζουν τον τρόπο με τον οποίο διατάσσονται τα γραφικά στοιχεία (κουμπιά, κείμενα, εικόνες κ.λπ.) στην οθόνη μιας εφαρμογής. Στο Android υπάρχουν διάφοροι τύποι Layouts όπως LinearLayout, RelativeLayout και ConstraintLayout. Κάθε ένα από αυτά τα είδη Layout έχει τα δικά του χαρακτηριστικά και περιορισμούς και επιλέγεται ανάλογα με τις απαιτήσεις σχεδιασμού της κάθε οθόνης.

Με την εξέλιξη του Android, η Google εισήγαγε το Jetpack Compose, ένα μοντέρνο εργαλείο για τον σχεδιασμό UI με δηλωτικό τρόπο. Αντί να χρησιμοποιούνται XML αρχεία, ο προγραμματιστής γράφει κώδικα Kotlin για να καθορίσει απευθείας το πώς πρέπει να εμφανίζεται το περιβάλλον χρήστη, ενισχύοντας την αναγνωσιμότητα, την επαναχρησιμοποίηση και τη δυνατότητα μεταβολής του περιεχομένου δυναμικά.

Η διάταξη των στοιχείων στην οθόνη παίζει πολύ σημαντικό ρόλο στην εμπειρία του χρήστη. Η καλή τοποθέτηση, η ισορροπία στην κατανομή του χώρου, καθώς και η δυνατότητα κλιμάκωσης του περιεχομένου στις διαφορετικές διαστάσεις οθονών (π.χ. smartphones, tablets) είναι απαραίτητα για την αποδοτική χρήση της εφαρμογής.

Χρήση στην εφαρμογή: Η εφαρμογή "Random Traveller" χρησιμοποιεί Compose Layouts για τη δημιουργία μοντέρνων και εύχρηστων οθονών. Για παράδειγμα, η κεντρική οθόνη περιλαμβάνει πεδία εισαγωγής για budget και αριθμό ημερών, τα οποία τοποθετούνται σε Column layout, ενώ το κουμπί αναζήτησης τοποθετείται σε εμφανές και εύκολα προσβάσιμο σημείο. Τα αποτελέσματα της αναζήτησης εμφανίζονται επίσης με χρήση Jetpack Compose, ώστε να επιτυγχάνεται ομοιομορφία και γρήγορη ανανέωση περιεχομένου.

Επιπλέον, χρησιμοποιείται δυναμική αλλαγή Layout ώστε να προσαρμόζεται η οθόνη όταν η συσκευή περιστρέφεται ή όταν ο χρήστης αλληλεπιδρά με συγκεκριμένα φίλτρα. Έτσι, βελτιώνεται σημαντικά η εμπειρία πλοήγησης, καθώς η διάταξη είναι φιλική και παραμένει κατανοητή ακόμα και σε πιο σύνθετες περιπτώσεις.

Η σωστή χρήση Layouts επιτρέπει όχι μόνο την αισθητική παρουσίαση των δεδομένων, αλλά και την προσαρμογή της διεπαφής σε διαφορετικές διαστάσεις οθονών και συσκευών, αυξάνοντας τη λειτουργικότητα και την προσβασιμότητα της εφαρμογής.

5.9 Dialogs

Τα Dialogs είναι μικρά παράθυρα που εμφανίζονται πάνω από το κύριο περιεχόμενο της οθόνης και χρησιμοποιούνται συνήθως για να τραβήξουν την προσοχή του χρήστη, να του παρέχουν επιπλέον πληροφορίες ή να τον κατευθύνουν ώστε να λάβει μία απόφαση. Μπορούν να περιλαμβάνουν προειδοποιητικά μηνύματα, επιβεβαιώσεις ενεργειών ή επιλογές μεταξύ πολλαπλών επιλογών.

Υπάρχουν διάφοροι τύποι Dialogs στο Android, όπως:

AlertDialog: Παρουσιάζει τίτλο, μήνυμα και κουμπιά ενεργειών (π.χ. "OK" ή "Cancel").

ProgressDialog: Δείχνει την πρόοδο κάποιας λειτουργίας, αν και πλέον αντικαθίσταται από άλλα components.

Custom Dialogs: Επιτρέπουν την εμφάνιση προσαρμοσμένων layouts για πιο σύνθετες λειτουργίες.

Η σωστή χρήση των Dialogs συμβάλλει στη βελτίωση της εμπειρίας χρήστη, καθώς επιτρέπει στο σύστημα να διαχειρίζεται κρίσιμες αλληλεπιδράσεις, χωρίς να διακόπτεται εντελώς η κύρια ροή της εφαρμογής. Είναι απαραίτητο να αποφεύγεται η υπερβολική χρήση τους, καθώς μπορεί να προκαλέσει σύγχυση ή να φανεί ενοχλητική στον χρήστη.

Χρήση στην εφαρμογή: Στην εφαρμογή "Random Traveller", τα Dialogs χρησιμοποιούνται κυρίως για την επιβεβαίωση ενεργειών του χρήστη, όπως η αποθήκευση μιας αναζήτησης ή η διαγραφή αποθηκευμένων αποτελεσμάτων. Επιπλέον, εμφανίζονται μηνύματα όταν δεν υπάρχει διαθέσιμη σύνδεση στο διαδίκτυο ή όταν υπάρχει σφάλμα κατά την ανάκτηση δεδομένων από την API.

Τα Dialogs, όταν σχεδιάζονται προσεκτικά, ενισχύουν την αίσθηση ασφάλειας του χρήστη, παρέχοντάς του ξεκάθαρη πληροφόρηση για την κατάσταση της εφαρμογής και καθοδήγηση για τα επόμενα βήματα. Αυτή η διαδραστική εμπειρία είναι κρίσιμη για την οικοδόμηση εμπιστοσύνης και την ενίσχυση της χρηστικότητας της εφαρμογής.

5.10 Context

Η έννοια του Context αποτελεί έναν από τους βασικότερους πυλώνες στην ανάπτυξη εφαρμογών Android, καθώς αποτελεί τη "γέφυρα" μεταξύ του συστήματος και της εφαρμογής. Το Context παρέχει

πρόσβαση σε πόρους του συστήματος, αρχεία, βάσεις δεδομένων, υπηρεσίες, και γενικά σε κάθε λειτουργικό στοιχείο που χρειάζεται η εφαρμογή για να εκτελέσει τις ενέργειές της.

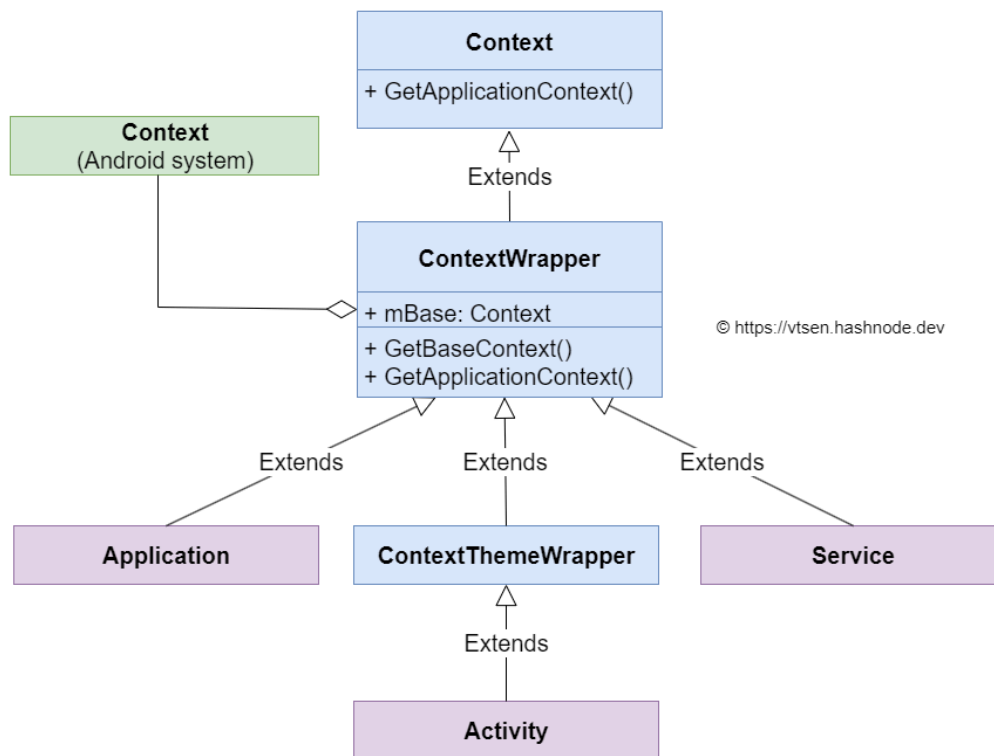
Υπάρχουν διάφοροι τύποι Context στο Android:

- **Application Context:** Ισχύει για όλη τη διάρκεια ζωής της εφαρμογής και χρησιμοποιείται κυρίως για global λειτουργίες.
- **Activity Context:** Συνδέεται με την εκάστοτε Activity και είναι κατάλληλο όταν θέλουμε να εμφανίσουμε διαλόγους, να ξεκινήσουμε νέες οθόνες ή να αποκτήσουμε πρόσβαση σε view components.

Η ορθή χρήση του Context είναι κρίσιμη. Για παράδειγμα, αν διατηρηθεί αναφορά σε Context μιας Activity μετά την καταστροφή της, μπορεί να προκληθεί διαρροή μνήμης (memory leak). Για τον λόγο αυτό, είναι σημαντικό να γίνεται διάκριση μεταξύ των περιπτώσεων όπου χρειάζεται το ένα ή το άλλο.

Χρήση στην εφαρμογή: Στην εφαρμογή "Random Traveller", το Context αξιοποιείται για την εμφάνιση Toasts (μηνυμάτων ειδοποίησης), την πρόσβαση σε Shared Preferences (προτιμήσεις χρήστη), τη δημιουργία Dialogs και την πλοήγηση μεταξύ των οθονών. Επιπλέον, αποτελεί βασικό παράγοντα στην επικοινωνία με εξωτερικές βιβλιοθήκες, όπως το Firebase και η Compose Navigation, όπου το Context είναι αναγκαίο για την αρχικοποίηση και διαχείριση πόρων.

Επομένως, το Context είναι κάτι παραπάνω από μια απλή έννοια – είναι ο κεντρικός μηχανισμός με τον οποίο η εφαρμογή συνδέεται με το Android λειτουργικό σύστημα. Η κατανόησή του είναι απαραίτητη για κάθε προγραμματιστή Android, και η σωστή του χρήση οδηγεί σε σταθερότερες, ασφαλέστερες και αποδοτικότερες εφαρμογές.



© <https://vtsen.hashnode.dev>

Διάγραμμα κλάσης Context

Κεφάλαιο 6ο: Τεχνολογίες που χρησιμοποιήθηκαν

6.1 Εισαγωγή

Στην ανάπτυξη της εφαρμογής "Random Traveller" αξιοποιήθηκε ένα σύνολο από σύγχρονες τεχνολογίες και εργαλεία, τα οποία επιλέχθηκαν με βάση τη λειτουργικότητα, την ευχρηστία και την ευκολία ενσωμάτωσής τους με την πλατφόρμα Android. Αυτές οι τεχνολογίες, αν και ποικίλουν ως προς τη χρήση και τον ρόλο τους, συνεργάζονται αρμονικά, υποστηρίζοντας τόσο την frontend όσο και την backend πλευρά της εφαρμογής.

Η χρήση αυτών των εργαλείων επιτρέπει:

- Τη δημιουργία δυναμικών και responsive περιβαλλόντων χρήστη.
- Τη διαχείριση χρηστών και την ασφάλεια των δεδομένων τους.
- Τη σύνδεση με εξωτερικά APIs για την ανάκτηση πληροφοριών πτήσεων.
- Την παρακολούθηση συμπεριφοράς χρήστη και τη συλλογή στατιστικών στοιχείων.

Η παρούσα ενότητα παρουσιάζει τις τεχνολογίες που χρησιμοποιήθηκαν, περιγράφοντας με σαφήνεια τον ρόλο της κάθε μιας στην ανάπτυξη της εφαρμογής, χωρίς να γίνεται χρήση κώδικα, αλλά μέσα από θεωρητική και λειτουργική προσέγγιση.

6.2 Google Firebase

6.2.1 Εισαγωγή

Το back-end της εφαρμογής βασίστηκε εξ ολοκλήρου στην πλατφόρμα Firebase της Google, μια ολοκληρωμένη λύση για την ανάπτυξη σύγχρονων εφαρμογών. Η επιλογή της Firebase επιτρέπει στον προγραμματιστή να επικεντρωθεί στην ανάπτυξη των βασικών λειτουργιών και του "ωφέλιμου" κώδικα της εφαρμογής, απαλλάσσοντάς τον από την πολυπλοκότητα της διαχείρισης και συντήρησης των servers που υποστηρίζουν τις διάφορες λειτουργίες.

Η Firebase υποστηρίζει την ανάπτυξη εφαρμογών για πολλαπλές πλατφόρμες, συμπεριλαμβανομένων των Android, iOS, Web, Unity και C++. Προσφέρει μια πληθώρα ενσωματωμένων λειτουργιών, απαραίτητων για τις περισσότερες σύγχρονες εφαρμογές, όπως:

Cloud Firestore: Μια ευέλικτη και επεκτάσιμη βάση δεδομένων NoSQL για αποθήκευση και συγχρονισμό δεδομένων σε πραγματικό χρόνο. (Αντικαθιστά την παλαιότερη "Realtime Database" σε νεότερα projects, αν και η Realtime Database παραμένει διαθέσιμη).

Authentication: Απλοποιημένη διαχείριση ταυτοποίησης χρηστών με διάφορους παρόχους (Google, Facebook, email/password, κ.λπ.).

- Cloud Storage: Αποθηκευτικός χώρος για αρχεία, όπως εικόνες και βίντεο.
- Cloud Functions: Εκτέλεση κώδικα back-end σε ένα ασφαλές και επεκτάσιμο περιβάλλον.
- Hosting: Γρήγορη και ασφαλής φιλοξενία για web εφαρμογές και στατικό περιεχόμενο.
- Machine Learning: Εργαλεία για την ενσωμάτωση λειτουργιών μηχανικής μάθησης στην εφαρμογή.
- Cloud Messaging (FCM): Αξιόπιστη αποστολή push notifications σε συσκευές χρηστών.

Η Firebase συνεχώς εξελίσσεται, προσθέτοντας νέες λειτουργίες και βελτιώσεις, καθιστώντας την μια εξαιρετική επιλογή για την ανάπτυξη σύγχρονων και επεκτάσιμων εφαρμογών.

6.2.2 Ιστορία

Η ιστορία της Firebase ξεκινά με την Envolv, μια startup που ιδρύθηκε το 2011 από τους James Tamplin και Andrew Lee. Αρχικά, η Envolv προσέφερε ένα API για την ενσωμάτωση λειτουργιών chat σε ιστοσελίδες.

Όμως, οι Tamplin και Lee παρατήρησαν κάτι ενδιαφέρον: οι προγραμματιστές χρησιμοποιούσαν το API τους όχι μόνο για chat, αλλά και για να μεταφέρουν και να συγχρονίζουν δεδομένα εφαρμογών σε πραγματικό χρόνο, όπως την κατάσταση ενός παιχνιδιού μεταξύ χρηστών.

Αυτό τους οδήγησε σε μια σημαντική απόφαση: να διαχωρίσουν το σύστημα chat από την υποκείμενη αρχιτεκτονική πραγματικού χρόνου. Έτσι, τον Απρίλιο του 2012, ιδρύθηκε η Firebase ως μια ανεξάρτητη εταιρεία, με στόχο να προσφέρει μια ολοκληρωμένη πλατφόρμα για την ανάπτυξη εφαρμογών πραγματικού χρόνου.

Η Firebase κατάφερε να εξασφαλίσει χρηματοδότηση από διάφορους επενδυτές και, τελικά, εξαγοράστηκε από την Google το 2014. Από τότε, η Google έχει επενδύσει σημαντικά στην ανάπτυξη της Firebase, προσθέτοντας συνεχώς νέες λειτουργίες και δυνατότητες, μετατρέποντάς την σε μια από τις πιο δημοφιλείς και ισχυρές πλατφόρμες για την ανάπτυξη σύγχρονων εφαρμογών. Σήμερα, η Firebase προσφέρει μια ευρεία γκάμα υπηρεσιών, από βάσεις δεδομένων και ταυτοποίηση χρηστών μέχρι φιλοξενία και λειτουργίες cloud, καλύπτοντας σχεδόν όλες τις ανάγκες ενός σύγχρονου προγραμματιστή.

6.2.3 Προσφερόμενες λειτουργίες

Οι υπηρεσίες της Firebase καλύπτουν ένα ευρύ φάσμα αναγκών και κατηγοριοποιούνται σε τρεις τομείς:

Ανάπτυξη Εφαρμογών: Εργαλεία για κατασκευή & επέκταση εφαρμογών.

- Cloud Firestore
- Firebase ML (Machine Learning Kit)
- Cloud Functions
- Authentication
- Hosting
- Cloud Storage
- Realtime Database (Legacy)
- Βελτίωση Ποιότητας Εφαρμογής: Εργαλεία για παρακολούθηση, εντοπισμό προβλημάτων & βελτίωση απόδοσης.
- Crashlytics
- Performance Monitoring
- Test Lab

Ανάπτυξη Επιχειρήσεων: Εργαλεία για αύξηση αφοσίωσης χρηστών, προώθηση & αύξηση εσόδων

- In-App Messaging
- Google Analytics
- Predictions

- A/B Testing
- Cloud Messaging (FCM)
- Remote Config
- Dynamic Links
- App Indexing (Deprecated)

Η Firebase προσφέρει ένα πλούσιο σύνολο εργαλείων που επιταχύνουν την ανάπτυξη εφαρμογών, επιτρέποντας στους προγραμματιστές να δημιουργήσουν πιο γρήγορα και αποτελεσματικά εφαρμογές υψηλής ποιότητας. Η ευκολία χρήσης και η ολοκληρωμένη φύση της πλατφόρμας την καθιστούν εξαιρετική επιλογή για προγραμματιστές όλων των επιπέδων.

6.2.4 Προσθέτοντας το Firebase σε εφαρμογή

Η ενσωμάτωση της Firebase σε Android εφαρμογές είναι απλή, χάρη στην καλή τεκμηρίωση της Google.

Προϋποθέσεις:

Android 4.0+ (Ice Cream Sandwich).

Google Play services (ελέγξτε την ελάχιστη έκδοση στην τεκμηρίωση Firebase για τις λειτουργίες που χρησιμοποιείτε).

Συνιστάται η τελευταία έκδοση του Android Studio.

Τρόποι Ενσωμάτωσης:

- Firebase Assistant (Android Studio): Ο ευκολότερος τρόπος. Συνδέει/δημιουργεί εφαρμογή με τις κατάλληλες ρυθμίσεις. Οι νεότερες εκδόσεις προσφέρουν περισσότερες δυνατότητες.
- Χειροκίνητα: Για παλαιότερες εκδόσεις ή σύνθετες ρυθμίσεις. Ακολουθήστε τις οδηγίες στην τεκμηρίωση Firebase.

Για απόλυτη ακρίβεια, ανατρέξτε πάντα στην επίσημη τεκμηρίωση της Firebase και του Android Studio.

6.2.5 Firebase Authentication

6.2.5.1 Εισαγωγή

Η υπηρεσία Firebase Authentication προσφέρει μια απλοποιημένη, ευέλικτη και ασφαλή προσέγγιση για τη διαχείριση ταυτοποίησης χρηστών σε εφαρμογές για κινητά και web. Στην περίπτωση της εφαρμογής "Random Traveller", χρησιμοποιείται αποκλειστικά για την εγγραφή και την είσοδο χρηστών, προκειμένου να αποκτήσουν πρόσβαση στις προσωποποιημένες λειτουργίες της εφαρμογής, όπως η αποθήκευση αναζητήσεων ή η προβολή προηγούμενων ταξιδιών.

Λειτουργία της Firebase Authentication (θεωρητική προσέγγιση)

Η ταυτοποίηση αποτελεί βασικό πυλώνα ασφάλειας σε κάθε εφαρμογή. Το Firebase Authentication υποστηρίζει πολλαπλές μεθόδους σύνδεσης, όπως:

- Email και Κωδικό πρόσβασης
- Σύνδεση με κοινωνικά δίκτυα (π.χ. Google, Facebook, Twitter)

- Επιβεβαίωση ταυτότητας μέσω κινητού τηλεφώνου Αυτές οι μέθοδοι καθιστούν εύκολη την υλοποίηση ενός ενιαίου, ασφαλούς συστήματος σύνδεσης χρηστών, χωρίς να απαιτείται δημιουργία σύνθετης υποδομής από την πλευρά του προγραμματιστή.

6.2.5.2 Εμπειρία Χρήστη & Πλεονεκτήματα

- Χάρη στη Firebase Authentication, η εφαρμογή μπορεί να:
- Παρέχει γρήγορη και ασφαλή είσοδο στους χρήστες
- Διατηρεί ενεργή την κατάσταση σύνδεσης (session)
- Ενσωματώνει πρακτικές βελτιστοποίησης εμπειρίας, όπως έξυπνη συμπλήρωση στοιχείων

6.2.5.3 Εγγραφή Χρήστη

Η δημιουργία νέου λογαριασμού χρήστη με email και κωδικό πρόσβασης στο Firebase Authentication γίνεται εύκολα μέσω του κατάλληλου API. Η διαδικασία αυτή περιλαμβάνει τη λήψη των στοιχείων από τον χρήστη, την επαλήθευση εγκυρότητας και την καταχώρισή τους στη βάση δεδομένων του Firebase.

Όταν ο χρήστης πατάει το κουμπί "Εγγραφή":
 Έλεγχος εγκυρότητας email και κωδικού
 Αν τα στοιχεία είναι έγκυρα:
 Δημιουργία νέου λογαριασμού με `createUser(email, password)`
 Εμφάνιση επιβεβαίωσης εγγραφής και μετάβαση στην αρχική οθόνη
 Αλλιώς:
 Εμφάνιση κατάλληλου μηνύματος λάθους

Ψευδοκώδικας Εγγραφής

Σημείωση: Η Firebase δεν επιβάλλει συγκεκριμένους κανόνες για την πολυπλοκότητα των κωδικών πρόσβασης. Είναι σημαντικό η εφαρμογή να υλοποιεί επιπλέον ελέγχους για να διασφαλίσει την ισχυρότητα των κωδικών.

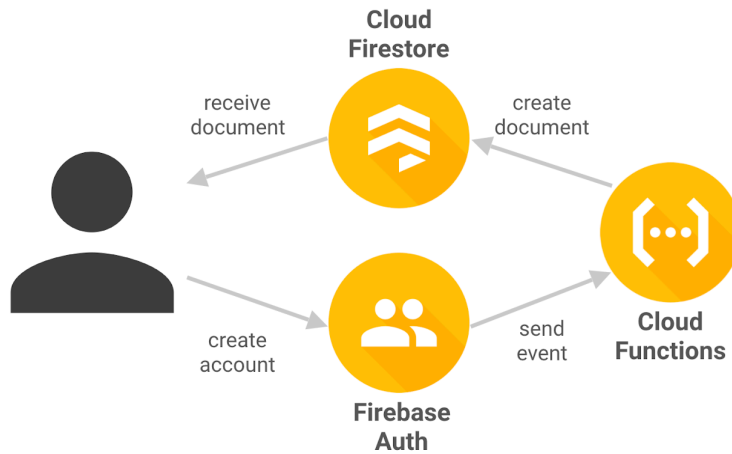
6.2.5.4 Είσοδος Χρήστη

Η είσοδος πραγματοποιείται με χρήση της μεθόδου `signInWithEmailAndPassword`, όπου ο χρήστης εισάγει τα διαπιστευτήριά του.

Όταν ο χρήστης πατάει το κουμπί "Είσοδος":
 Αν τα στοιχεία είναι σωστά:
 Εκτέλεση `signIn(email, password)`
 Αν η είσοδος είναι επιτυχής:
 Μετάβαση στην αρχική οθόνη
 Αλλιώς:
 Εμφάνιση μηνύματος σφάλματος (π.χ. λάθος στοιχεία)

Ψευδοκώδικας Εισόδου

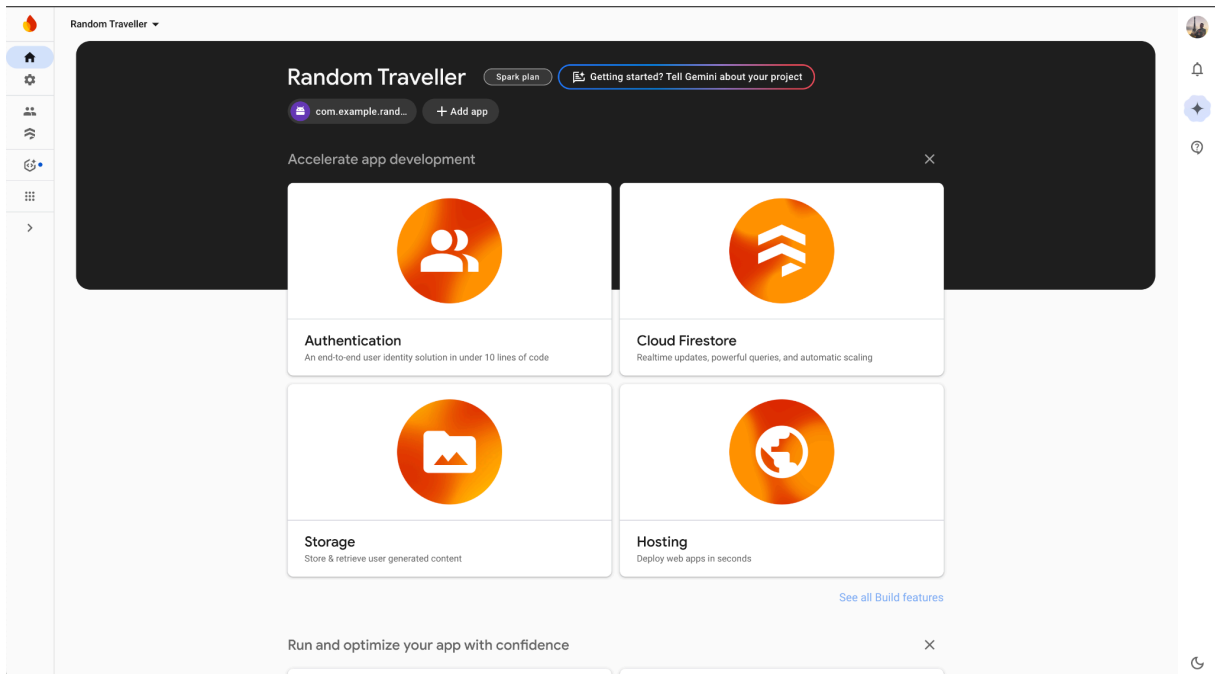
Η μέθοδος `getCurrentUser()` επιτρέπει την ανάκτηση των στοιχείων του τρέχοντος χρήστη μετά την είσοδο. Αυτή η πληροφορία μπορεί να χρησιμοποιηθεί για την προσωποποίηση της εμπειρίας χρήστη εντός της εφαρμογής.



Firestore authentication diagram

6.2.5.5 FirebaseUI

Το FirebaseUI είναι μια βιβλιοθήκη που απλοποιεί τη δημιουργία διεπαφής χρήστη για την είσοδο και εγγραφή χρηστών μέσω του Firebase Authentication.



Γραφικό περιβάλλον διαχείρισης Firebase

FirestoreUI

- Υποστήριξη πολλαπλών παρόχων
- Διαχείριση λογαριασμών (εγγραφή, ανάκτηση κωδικού)

- Σύνδεση λογαριασμών από διαφορετικούς παρόχους
- Ενσωμάτωση Smart Lock για αυτόματη σύνδεση

Κατά την εκκίνηση της εφαρμογής:
 Δημιουργία `Intent` με επιλογές εισόδου μέσω `FirebaseUI`
 Εκκίνηση του `Intent`

Όταν επιστρέφει το αποτέλεσμα:

Αν επιτυχής:

Είσοδος χρήστη στην εφαρμογή

Αλλιώς:

Εμφάνιση κατάλληλου μηνύματος λάθους

Ψευδοκώδικας `FirebaseUI` Ροής

6.2.5.6 Συμπεράσματα

Η χρήση της `Firebase Authentication` στην εφαρμογή "Random Traveller" εστιάζει αποκλειστικά στην ασφαλή και φιλική προς τον χρήστη διαδικασία σύνδεσης και εγγραφής. Χάρη στην υποδομή της `Google`, η υπηρεσία εξασφαλίζει αξιοπιστία, υποστήριξη πολλαπλών μεθόδων αυθεντικοποίησης και απλοποιεί σημαντικά την υλοποίηση λειτουργιών ασφάλειας, επιτρέποντας στην ομάδα ανάπτυξης να εστιάσει στη βελτίωση της εμπειρίας χρήσης και της λειτουργικότητας της εφαρμογής.

6.2.6 Gradle

6.2.6.1 Εισαγωγή

Το `Gradle` είναι ένα σύγχρονο εργαλείο αυτοματισμού για την κατασκευή λογισμικού (`build automation`), το οποίο χρησιμοποιείται ευρέως στην ανάπτυξη `Android` εφαρμογών. Πρόκειται για ένα σύστημα το οποίο βοηθά στην οργάνωση και την εκτέλεση διαδικασιών όπως η μεταγλώττιση του κώδικα (`compilation`), η διαχείριση εξαρτήσεων (`dependencies`), η συσκευασία (`packaging`), η δοκιμή και η εγκατάσταση εφαρμογών.

Στο πλαίσιο της εφαρμογής "Random Traveller", το `Gradle` χρησιμοποιήθηκε για την παραμετροποίηση του έργου, την ενσωμάτωση βιβλιοθηκών (όπως το `Firebase Authentication`), αλλά και τη ρύθμιση διαδικασιών `build`, διασφαλίζοντας την ομαλή και αποδοτική ανάπτυξη.

6.2.6.2 Χαρακτηριστικά Gradle

Το `Gradle` βασίζεται σε `Groovy` (ή `Kotlin DSL`) και επιτρέπει στον προγραμματιστή να ορίσει με λεπτομέρεια τη διαδικασία κατασκευής μιας εφαρμογής. Κάποια από τα βασικά χαρακτηριστικά του είναι:

- Υποστήριξη πολλαπλών `modules` (π.χ. `app`, `libraries`)
- Κατασκευή έργων με συγκεκριμένο τρόπο ανάλογα με το περιβάλλον (`debug`, `release`)
- Επαναχρησιμοποίηση σεναρίων και εύκολη επέκταση με `plugins`
- Καλή διαχείριση εξαρτήσεων, με αυτόματη λήψη και ανανέωση βιβλιοθηκών

6.2.6.3 Δομή Gradle στο Android:

Σε κάθε Android έργο υπάρχουν δύο βασικά αρχεία ρυθμίσεων Gradle:

1. Ένα αρχείο στο επίπεδο του έργου (project-level), όπου δηλώνονται γενικές ρυθμίσεις, όπως τα repositories για την ανάκτηση βιβλιοθηκών.
2. Ένα αρχείο στο επίπεδο της εφαρμογής (module-level), όπου ορίζονται οι εξαρτήσεις, η έκδοση SDK που χρησιμοποιείται, το μοναδικό αναγνωριστικό της εφαρμογής, και άλλες κρίσιμες παραμέτρους κατασκευής.

6.2.6.4 Περιγραφική παρουσίαση ρυθμίσεων Gradle:

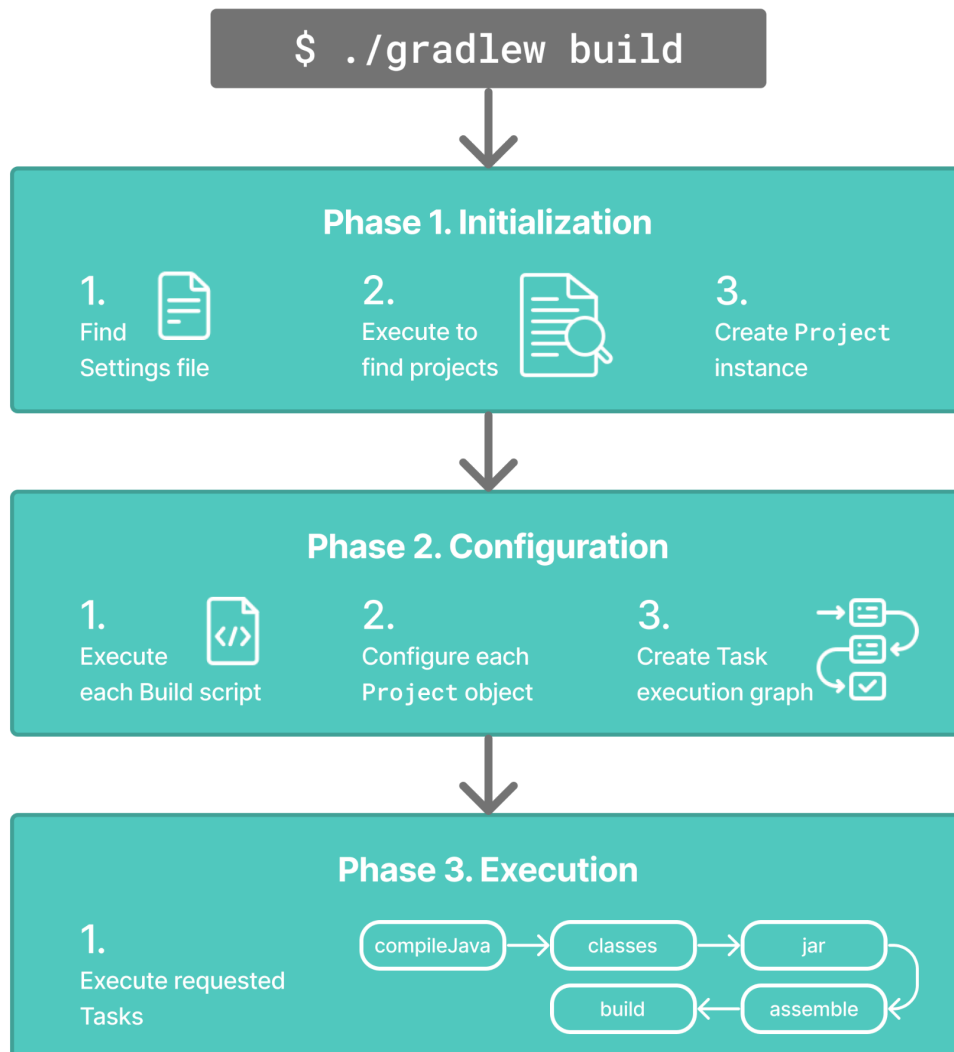
- Ορίζεται η έκδοση SDK με την οποία θα γίνει η μεταγλώττιση.
- Καθορίζονται οι ελάχιστες και οι προτεινόμενες εκδόσεις Android που θα υποστηρίζει η εφαρμογή.
- Δηλώνεται το μοναδικό όνομα της εφαρμογής στο Play Store.
- Ενεργοποιούνται ή απενεργοποιούνται επιλογές για συμπίεση κώδικα και ασφάλεια, όπως η χρήση ProGuard.
- Δηλώνονται οι βιβλιοθήκες που θα χρησιμοποιηθούν, όπως το Firebase Authentication, το AndroidX και άλλα εργαλεία διεπαφής ή λογικής της εφαρμογής.

6.2.6.5 Οφέλη χρήσης Gradle:

- Αυτοματοποίηση: Ελαχιστοποιεί τις χειροκίνητες ενέργειες κατά την κατασκευή του project.
- Αναπαραγωγιμότητα: Κάθε build μπορεί να επαναληφθεί με ακρίβεια.
- Ευκολία ενσωμάτωσης εργαλείων όπως το Firebase, το Kotlin, βιβλιοθήκες UI, εργαλεία δοκιμών, κ.λπ.
- Επέκταση με Plugins: Για CI/CD, δοκιμές, ανάλυση ποιότητας κώδικα και άλλες χρήσιμες λειτουργίες.

6.2.6.6 Συμπεράσματα

Το Gradle αποτελεί αναπόσπαστο κομμάτι της ανάπτυξης Android εφαρμογών. Στην περίπτωση του "Random Traveller", συνέβαλε καθοριστικά στην οργάνωση της ανάπτυξης, την ενσωμάτωση εξαρτήσεων, καθώς και στη δημιουργία ευέλικτων και επεκτάσιμων διαδικασιών build, διευκολύνοντας τόσο την ανάπτυξη όσο και τη συντήρηση της εφαρμογής.



Τα τρία βασικά στάδια εκτέλεσης του Gradle build – Initialization, Configuration και Execution

6.2.7 Kotlin

6.2.7.1 Εισαγωγή

Η γλώσσα προγραμματισμού Kotlin έχει καθιερωθεί τα τελευταία χρόνια ως η προτεινόμενη γλώσσα για την ανάπτυξη Android εφαρμογών, αποτελώντας πλέον επίσημα υποστηριζόμενη από την Google. Σχεδιάστηκε με σκοπό να είναι σύγχρονη, λιτή, ασφαλής από σφάλματα και πλήρως συμβατή με τον υπάρχοντα κώδικα σε Java. Η εφαρμογή "Random Traveller" αναπτύχθηκε χρησιμοποιώντας Kotlin, εκμεταλλευόμενη τα πλεονεκτήματά της τόσο στην αναγνωσιμότητα όσο και στην ευκολία συντήρησης του κώδικα.

6.2.7.2 Χαρακτηριστικά της Kotlin

Συνοπτικότητα: Ο κώδικας σε Kotlin είναι πιο περιεκτικός σε σχέση με την Java. Λιγότερες γραμμές κώδικα σημαίνουν λιγότερες πιθανότητες σφάλματος και μεγαλύτερη ευκολία στην ανάγνωση.

- Ασφάλεια τύπων (Type Safety): Ενσωματώνει ενδείξεις null safety, μειώνοντας σημαντικά τα runtime σφάλματα λόγω NullPointerException.
- Συμβατότητα με Java: Επιτρέπει τη σταδιακή μετάβαση από Java σε Kotlin, καθώς μπορούν να συνυπάρχουν στο ίδιο έργο.
- Λειτουργικός προγραμματισμός: Υποστηρίζει λειτουργικές έννοιες όπως lambdas, higher-order functions, map/filter/reduce.
- Επέκταση συναρτήσεων (Extension functions): Δίνει τη δυνατότητα επέκτασης λειτουργικότητας σε υπάρχουσες κλάσεις χωρίς να χρειάζεται η αλλαγή τους.

6.2.7.3 Kotlin Coroutines for Asynchronous Programming

Οι Kotlin Coroutines χρησιμοποιήθηκαν στην εφαρμογή "Random Traveller" για την εκτέλεση ασύγχρονων εργασιών, όπως δικτυακά αιτήματα και λειτουργίες βάσης δεδομένων. Η χρήση coroutines επιτρέπει την αναστολή και επανεκκίνηση διεργασιών χωρίς να μπλοκάρουν το κύριο νήμα (UI thread), με αποτέλεσμα την ομαλή εμπειρία χρήστη.

Πλεονεκτήματα:

- Απλοποιούν τον ασύγχρονο κώδικα κάνοντάς τον πιο ευανάγνωστο.
- Αποφεύγουν προβλήματα που προκαλούνται από callbacks και threads.
- Επιτρέπουν τη χρήση δομημένου concurrency, διατηρώντας ελεγχόμενες και ακυρώσιμες διεργασίες.
- Μειώνουν την πολυπλοκότητα του κώδικα όταν απαιτείται αναμονή για αποτελέσματα (π.χ. από API).

Χρήση coroutines στην εφαρμογή:

- Για την ανάκτηση δεδομένων από εξωτερικά APIs (όπως πτήσεις).
- Για επικοινωνία με Firebase Authentication.
- Για την αποθήκευση και ανάκτηση δεδομένων από τοπική ή cloud βάση δεδομένων.

```

Όταν ο χρήστης πατά "Αναζήτηση Προορισμού"
Ξεκινά coroutine
  Κάνε αίτημα στο API πτήσεων
  Αναμονή απάντησης
  Αν απάντηση επιτυχής
    -> Εμφάνισε προτεινόμενο προορισμό στην οθόνη
    -> Αποθήκευσε τα αποτελέσματα στην τοπική βάση
  Αλλιώς
    -> Εμφάνισε μήνυμα σφάλματος στον χρήστη
Τερματισμός coroutine

```

Ψευδοκώδικας αναζήτησης προορισμού

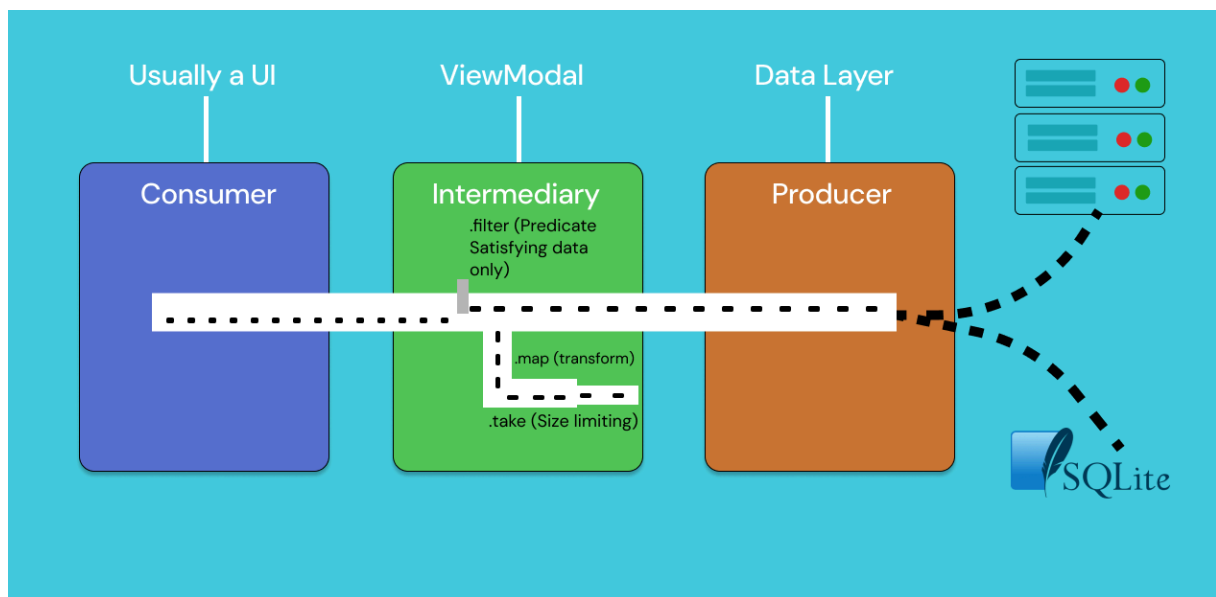
6.2.7.4 Kotlin Flow for UI Reactivity

Για τη διαχείριση των δεδομένων και την άμεση ανταπόκριση της διεπαφής χρήστη (UI) σε αλλαγές, στην εφαρμογή "Random Traveller" αξιοποιήθηκε το Kotlin Flow. Το Flow είναι μια βιβλιοθήκη της Kotlin που επιτρέπει την παρακολούθηση και την αντιδραστική ροή δεδομένων, επιτρέποντας στη

διεπαφή χρήστη να ενημερώνεται αυτόματα, χωρίς την ανάγκη άμεσης παρέμβασης από τον προγραμματιστή κάθε φορά που αλλάζουν τα δεδομένα.

Πλεονεκτήματα:

- Παρέχει υποστήριξη για streams δεδομένων που μπορούν να εκπέμπουν πολλά στοιχεία με την πάροδο του χρόνου.
- Υποστηρίζει την backpressure, διασφαλίζοντας ότι το UI δεν υπερφορτώνεται όταν οι ροές δεδομένων είναι έντονες ή συνεχείς.
- Δουλεύει άψογα με το lifecycle των Android components, μειώνοντας τον κίνδυνο memory leaks και σφαλμάτων συγχρονισμού.
- Επιτρέπει τη σύνθεση πολλών ροών (flows) για σύνθετη λογική, χωρίς την ανάγκη πολύπλοκου και μπλεγμένου κώδικα.



Ροή δεδομένων με Kotlin Flow για UI Reactivity – Από το Data Layer στο UI μέσω ViewModel

Χρήση Kotlin Flow στην εφαρμογή:

- Παρακολούθηση των αποθηκευμένων αναζητήσεων του χρήστη σε πραγματικό χρόνο.
- Άμεση ενημέρωση του UI όταν ο χρήστης πραγματοποιεί ή διαγράφει μια αναζήτηση.
- Ενημέρωση των αποτελεσμάτων πτήσεων με βάση φίλτρα ή μεταβλητές που ορίζει ο χρήστης χωρίς επανεκκίνηση της αναζήτησης.

```
Κατά την αρχικοποίηση της εφαρμογής
Εκκίνηση παρακολούθησης όλων των flows που σχετίζονται με το UI
-> SavedSearches Flow
-> Filters Flow
-> UI Visibility Flow
```

```
Για κάθε ροή που αλλάζει
-> Ενημέρωσε τα αντίστοιχα UI components
-> Καταγραφή αλλαγής (logging)
-> Έλεγχος εάν απαιτείται επαναφόρτωση εξωτερικών δεδομένων
```

Ψευδοκώδικα: Αντιδραστική ροή δεδομένων στην εφαρμογή

6.3 Compose Navigation

Για την αποτελεσματική διαχείριση της πλοήγησης ανάμεσα στις οθόνες και τα composables της εφαρμογής "Random Traveller", χρησιμοποιήθηκε η βιβλιοθήκη Jetpack Compose Navigation. Αυτή η βιβλιοθήκη επιτρέπει την υλοποίηση ενός declarative μοντέλου πλοήγησης, το οποίο είναι πλήρως εναρμονισμένο με τη φιλοσοφία του Jetpack Compose.

Με τη βοήθεια του Compose Navigation:

- Η πλοήγηση γίνεται με βάση έναν NavHost που περιέχει όλα τα δυνατά destinations της εφαρμογής.
- Κάθε οθόνη (ή Composable) δηλώνεται ως route με συγκεκριμένες παραμέτρους και τοποθετείται σε ένα κοινό σχήμα πλοήγησης.
- Οι μεταβάσεις μεταξύ των οθονών καθορίζονται με τρόπο σαφή και προβλέψιμο, αποφεύγοντας την περιττή πολυπλοκότητα του imperative navigation.

Πλεονεκτήματα:

- Απλοποιεί τη διαχείριση πλοήγησης σε Compose εφαρμογές.
- Ορίζει ξεκάθαρα τις διαθέσιμες διαδρομές της εφαρμογής και τις μεταβάσεις μεταξύ αυτών.
- Επιτρέπει την αποθήκευση και ανάκτηση της κατάστασης πλοήγησης (back stack), προσφέροντας συνεκτική εμπειρία χρήσης.
- Υποστηρίζει παραμέτρους στις διαδρομές (arguments) και δυναμικά URLs, βελτιώνοντας την ευελιξία της εφαρμογής.

```
Κατά την εκκίνηση της εφαρμογής
-> Ορίζεται ο NavController
-> Δημιουργείται ο NavHost με τις εξής διαδρομές:
  - home: Εμφάνιση κύριας σελίδας
  - search: Εμφάνιση σελίδας αναζήτησης
  - results: Εμφάνιση αποτελεσμάτων με δεδομένα
  - details/{id}: Σελίδα λεπτομερειών με δυναμικό id

Όταν ο χρήστης πατήσει "Αναζήτηση"
-> Εκτελείται πλοήγηση προς route "search"

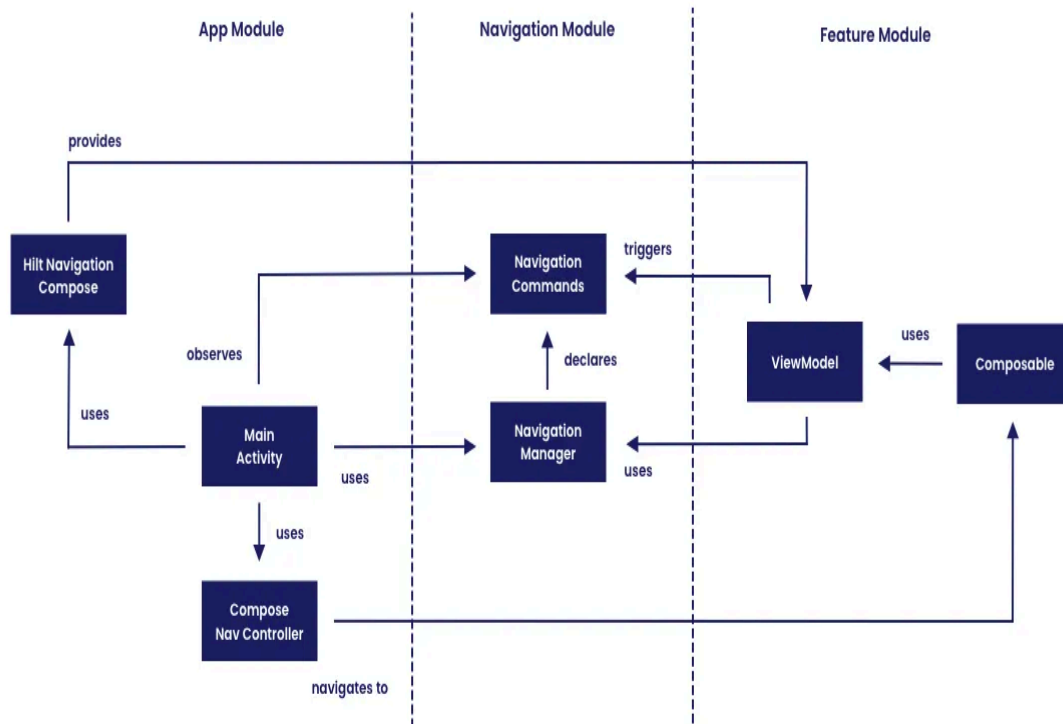
Όταν ολοκληρωθεί αναζήτηση
-> Πλοήγηση σε "results" και αποστολή φίλτρων ως arguments

Όταν επιλεγεί ένα αποτέλεσμα
-> Πλοήγηση σε "details/{id}" με id του επιλεγμένου προορισμού
```

Ψευδοκώδικα: Πλοήγηση με Compose Navigation στην εφαρμογή

Η χρήση του Jetpack Compose Navigation απλοποίησε σημαντικά τον χειρισμό πλοήγησης στην εφαρμογή, ενισχύοντας παράλληλα τη συντηρησιμότητα και την ευκολία επέκτασης της αρχιτεκτονικής.

Συγκεκριμένα, η declarative προσέγγιση του συστήματος πλοήγησης επιτρέπει στον προγραμματιστή να διαχειρίζεται όλες τις πιθανές διαδρομές και τις καταστάσεις της εφαρμογής με έναν οργανωμένο και επεκτάσιμο τρόπο. Η απουσία boilerplate κώδικα και η δυνατότητα απευθείας ορισμού των συνθηκών πλοήγησης μέσα στα composables οδηγούν σε πιο καθαρό και δομημένο κώδικα. Επιπλέον, το γεγονός ότι οι διαδρομές ορίζονται με σαφή τρόπο μειώνει την πιθανότητα σφαλμάτων και βοηθά στον ευκολότερο εντοπισμό και επίλυση προβλημάτων κατά τη διάρκεια της ανάπτυξης και της συντήρησης της εφαρμογής.



Αρχιτεκτονική Modular Navigation

6.4 GraphQL

Για τον καθορισμό του τρόπου με τον οποίο η εφαρμογή "Random Traveller" επικοινωνεί με τον διακομιστή και ανακτά τα απαιτούμενα δεδομένα, επιλέχθηκε η χρήση της γλώσσας ερωτημάτων GraphQL. Σε αντίθεση με τα παραδοσιακά REST APIs, όπου οι διαδρομές του backend είναι προκαθορισμένες και επιστρέφουν σταθερές δομές δεδομένων, το GraphQL εισάγει μια δυναμική προσέγγιση, επιτρέποντας στον client (δηλαδή την εφαρμογή) να ορίζει με ακρίβεια ποια πεδία και πόρους χρειάζεται.

Αυτό το χαρακτηριστικό είναι ιδιαίτερα ωφέλιμο, καθώς μειώνει φαινόμενα όπως το overfetching (λήψη περισσότερων δεδομένων από τα απαραίτητα) και το underfetching (λήψη ελλιπών δεδομένων που απαιτούν επιπλέον αιτήματα), οδηγώντας σε πιο στοχευμένη και αποδοτική επικοινωνία με τον server. Το αποτέλεσμα είναι ταχύτεροι χρόνοι απόκρισης, μειωμένη κατανάλωση εύρους ζώνης και γενικά πιο ομαλή εμπειρία χρήσης.

Επιπλέον, το GraphQL διαθέτει ενσωματωμένο σύστημα τύπων (type system), το οποίο παρέχει αυστηρή τυποποίηση και προβλέψιμη δομή στα δεδομένα που ανταλλάσσονται. Αυτή η δυνατότητα ενισχύει σημαντικά την αξιοπιστία και την ασφάλεια της εφαρμογής, καθώς επιτρέπει την ανίχνευση λαθών κατά το στάδιο ανάπτυξης (compile-time validation), μειώνοντας τις πιθανότητες αποτυχίας κατά την εκτέλεση.

Συνοπτικά, η χρήση του GraphQL στην "Random Traveller":

- Δίνει τον έλεγχο στον client: Η εφαρμογή ζητά μόνο τα απαραίτητα δεδομένα.
- Μειώνει την περιττή μεταφορά πληροφορίας: Εξοικονομεί bandwidth και επιταχύνει τις λειτουργίες.
- Προσφέρει ευελιξία και επεκτασιμότητα: Νέες λειτουργίες μπορούν να προστεθούν χωρίς να απαιτούνται σημαντικές αλλαγές στον backend.
- Ενισχύει την αξιοπιστία: Χάρη στο στατικό type system και τα έγκυρα σχήματα ερωτημάτων.

Αυτή η επιλογή αποδείχθηκε καθοριστική για τη σταθερότητα, την απόδοση και την επεκτασιμότητα της εφαρμογής.

6.5 Apollo Client

Για την απλοποίηση της αλληλεπίδρασης με το GraphQL backend και τη διαχείριση των δεδομένων που ανακτώνται, χρησιμοποιήθηκε το Apollo Client. Πρόκειται για μια εξειδικευμένη βιβλιοθήκη που λειτουργεί ως ενδιάμεσος ανάμεσα στην εφαρμογή και το backend, επιτρέποντας την αποστολή GraphQL ερωτημάτων και την αποτελεσματική διαχείριση των απαντήσεων.

Η αρχιτεκτονική του είναι cache-aware, δηλαδή έχει τη δυνατότητα να αποθηκεύει τα δεδομένα τοπικά στη συσκευή. Αυτό σημαίνει πως δεν χρειάζεται να αποστέλλεται το ίδιο αίτημα κάθε φορά που απαιτείται κάποιο δεδομένο, μειώνοντας τον φόρτο του backend και την κατανάλωση bandwidth, ενώ παράλληλα επιταχύνεται η απόκριση της εφαρμογής.

Ένα ακόμη σημαντικό χαρακτηριστικό του Apollo Client είναι η αυτοματοποιημένη διαχείριση της κατάστασης: όταν αλλάξουν τα δεδομένα (είτε μέσω νέου αιτήματος είτε λόγω αλληλεπίδρασης του χρήστη), το UI ενημερώνεται αυτόματα χωρίς να απαιτείται πρόσθετη λογική.

Επιπλέον, παρέχει ενσωματωμένα εργαλεία για τη διαχείριση σφαλμάτων, διευκολύνοντας τον προγραμματιστή στον εντοπισμό και την επίλυση προβλημάτων κατά την επικοινωνία με το backend, ενισχύοντας τη σταθερότητα και τη συντηρησιμότητα της εφαρμογής.

Παράδειγμα λειτουργίας Apollo Client στην εφαρμογή "Random Traveller":

Κατά την αναζήτηση προορισμού από τον χρήστη:

- Η εφαρμογή στέλνει ένα αίτημα GraphQL μέσω του Apollo Client, ζητώντας συγκεκριμένα δεδομένα όπως **τιμή εισιτηρίων**.
- Ο Apollo Client λαμβάνει τα δεδομένα από το backend και τα παραδίδει απευθείας στην οθόνη αποτελεσμάτων της εφαρμογής.
- Αν ο χρήστης τροποποιήσει κάποιο φίλτρο, ο Apollo Client αποστέλλει νέο αίτημα με τα ενημερωμένα κριτήρια και η UI ανανεώνεται αυτόματα με τα νέα αποτελέσματα.

Αυτό το παράδειγμα αναδεικνύει την απλότητα και την αποτελεσματικότητα που προσφέρει ο Apollo Client στην ανάκτηση και διαχείριση των δεδομένων, χωρίς την ανάγκη σύνθετου χειρισμού από τον προγραμματιστή.

6.6 OkHttp

Για την πραγματοποίηση των δικτυακών αιτημάτων προς τον διακομιστή της εφαρμογής "Random Traveller", χρησιμοποιήθηκε η βιβλιοθήκη OkHttp. Πρόκειται για μία ισχυρή και αξιόπιστη λύση που

επιτρέπει στην εφαρμογή να επικοινωνεί με τον απομακρυσμένο server με ασφάλεια και συνέπεια, ακόμη και σε συνθήκες περιορισμένης συνδεσιμότητας.

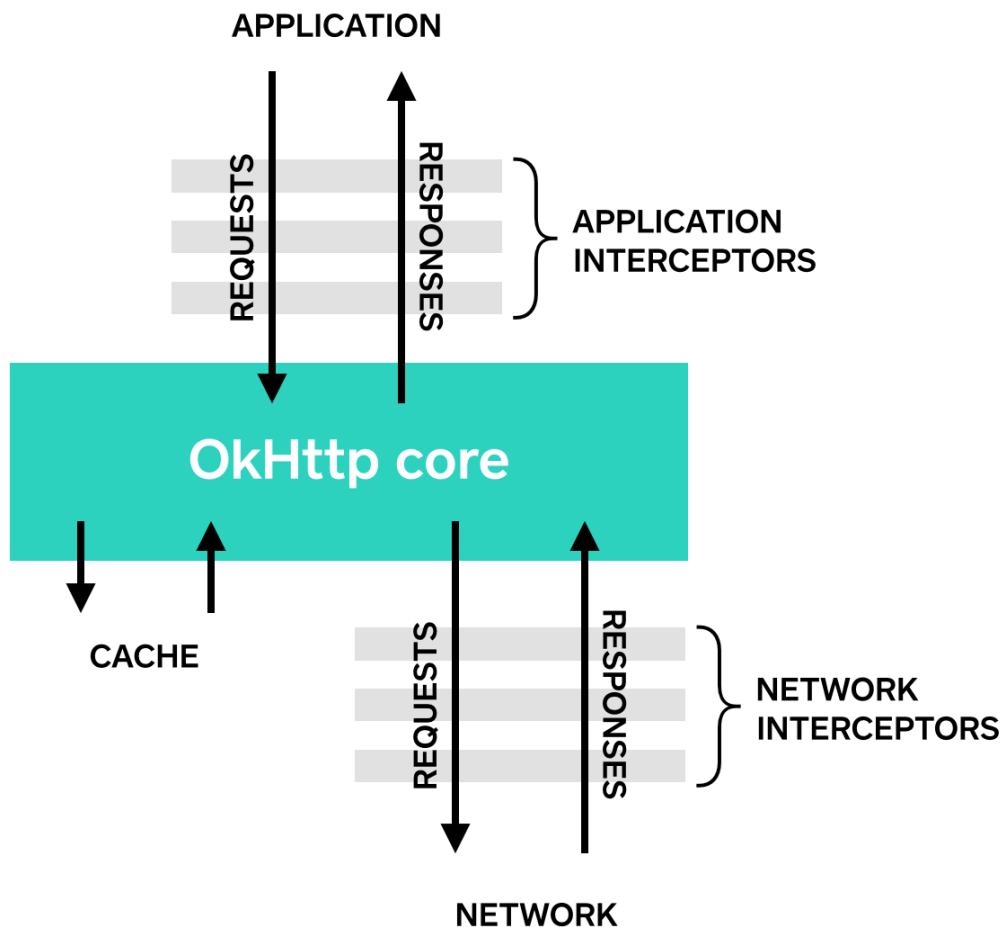
Η χρήση του OkHttp διασφαλίζει ότι τα αιτήματα αποστέλλονται γρήγορα και με ασφάλεια, υποστηρίζοντας τα σύγχρονα πρωτόκολλα επικοινωνίας (όπως HTTP/2) και τεχνολογίες όπως WebSocket, όπου χρειάζεται. Χάρη στην αποτελεσματική του σχεδίαση, η βιβλιοθήκη μπορεί να προσαρμοστεί σε διαφορετικές συνθήκες δικτύου, αποφεύγοντας την απώλεια δεδομένων ή την ανάγκη για χειροκίνητη επανάληψη αιτημάτων.

Παράδειγμα χρήσης στην εφαρμογή:

Όταν ο χρήστης της εφαρμογής εισάγει τα ταξιδιωτικά του κριτήρια και ξεκινά μια αναζήτηση:

- Η εφαρμογή χρησιμοποιεί το OkHttp για να δημιουργήσει ένα HTTP αίτημα που αποστέλλεται στον server.
- Ο server επεξεργάζεται το αίτημα και επιστρέφει μια απάντηση με τις διαθέσιμες πτήσεις.
- Το OkHttp λαμβάνει την απάντηση και τη διαβιβάζει γρήγορα στο σύστημα διαχείρισης δεδομένων της εφαρμογής, ώστε τα αποτελέσματα να εμφανιστούν άμεσα στην οθόνη του χρήστη.

Η αξιοπιστία του OkHttp καθιστά τη διαδικασία απρόσκοπτη και συμβάλλει στην ομαλή εμπειρία πλοήγησης, ανεξαρτήτως της ποιότητας σύνδεσης του χρήστη.



Αρχιτεκτονική του OkHttp

6.7 Kiwi's API

Για την απόκτηση πληροφοριών σχετικών με πτήσεις, όπως δρομολόγια, τιμές, διαθεσιμότητα και χρονική διάρκεια, η εφαρμογή "Random Traveller" βασίστηκε στο API της πλατφόρμας Kiwi.com. Το API αυτό λειτουργεί ως γέφυρα επικοινωνίας μεταξύ της εφαρμογής και της βάσης δεδομένων της Kiwi, επιτρέποντας την ανταλλαγή πληροφορίας σε πραγματικό χρόνο.

Η χρήση του Kiwi's API ήταν καθοριστική για τη λειτουργικότητα της εφαρμογής, καθώς χωρίς αυτό δεν θα ήταν δυνατή η δυναμική παρουσίαση των διαθέσιμων επιλογών πτήσεων. Το API παρείχε τη δυνατότητα ανάκτησης δεδομένων από έναν μεγάλο αριθμό αεροπορικών εταιρειών και ταξιδιωτικών πρακτόρων, επιτρέποντας στους χρήστες της εφαρμογής να συγκρίνουν πολλαπλές εναλλακτικές και να λαμβάνουν εξατομικευμένες προτάσεις με βάση το σημείο αναχώρησης, τον προϋπολογισμό και τη διάρκεια ταξιδιού.

Παράδειγμα χρήσης στην εφαρμογή:

Όταν ένας χρήστης της εφαρμογής εισάγει την τοποθεσία αναχώρησης και έναν επιθυμητό προϋπολογισμό:

- Η εφαρμογή δημιουργεί ένα ερώτημα βασισμένο σε αυτά τα κριτήρια και το αποστέλλει στο Kiwi API.
- Το API επιστρέφει δεδομένα για διαθέσιμους προορισμούς που ταιριάζουν με τα κριτήρια του χρήστη.
- Τα δεδομένα που λαμβάνονται ενσωματώνονται στο περιβάλλον της εφαρμογής, επιτρέποντας την εμφάνιση των αποτελεσμάτων πτήσεων με τρόπο οπτικά ευχάριστο και λειτουργικό.

Αυτή η ενσωμάτωση προσέφερε στους χρήστες μια πλούσια και λειτουργική εμπειρία αναζήτησης, με αξιόπιστες πληροφορίες και ενημερώσεις σε σχεδόν πραγματικό χρόνο. Επιπλέον, χάρη στην πληρότητα των δεδομένων του Kiwi API, η εφαρμογή απέκτησε πρόσβαση σε πτήσεις που δεν προσφέρονται εύκολα από άλλες υπηρεσίες.

6.8 Dependency Injection with Hilt

Στο πλαίσιο της αρχιτεκτονικής της εφαρμογής Random Traveller, κρίθηκε απαραίτητη η υιοθέτηση μιας μεθόδου που θα επιτρέψει την αποτελεσματική διαχείριση των εξαρτήσεων μεταξύ των επιμέρους στοιχείων. Για τον σκοπό αυτό αξιοποιήθηκε το Hilt, ένα εργαλείο dependency injection σχεδιασμένο ειδικά για το Android οικοσύστημα.

Η έννοια του dependency injection αναφέρεται στη διαδικασία κατά την οποία οι απαραίτητες εξαρτήσεις (όπως αντικείμενα, κλάσεις ή λειτουργίες) εισάγονται αυτόματα σε ένα στοιχείο του λογισμικού, αντί να δημιουργούνται εσωτερικά από αυτό. Αυτή η πρακτική ενισχύει την ανεξαρτησία των στοιχείων, διευκολύνει τις δοκιμές και βελτιώνει τη δυνατότητα επαναχρησιμοποίησης του κώδικα.

Το Hilt έπαιξε καθοριστικό ρόλο στην εφαρμογή Random Traveller για τα εξής:

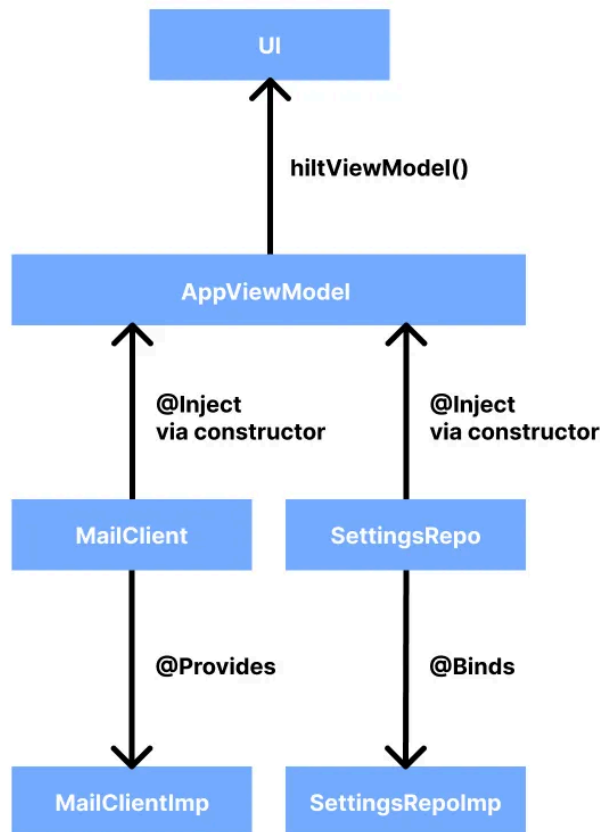
- Κεντρική διαχείριση εξαρτήσεων: Όλες οι κοινές λειτουργίες, όπως το repository για την ανάκτηση των δεδομένων, οι network clients, ή ακόμα και τα view models, μπορούν να οριστούν ως εξαρτήσεις και να διατίθενται αυτόματα όπου χρειάζονται μέσα στην εφαρμογή.

- Δομημένη αρχιτεκτονική: Το Hilt βοηθά στη δημιουργία μιας καθαρής αρχιτεκτονικής με σαφώς διαχωρισμένα επίπεδα (data, domain, presentation). Έτσι, κάθε μέρος της εφαρμογής γνωρίζει μόνο ό,τι χρειάζεται, χωρίς να υπάρχει άμεση εξάρτηση από το υπόλοιπο σύστημα.
- Ευκολία συντήρησης και επεκτασιμότητας: Με την αυτόματη διαχείριση της δημιουργίας αντικειμένων και των κύκλων ζωής τους, μειώνεται σημαντικά η πιθανότητα σφαλμάτων, ενώ διευκολύνεται η προσθήκη νέων λειτουργιών στο μέλλον.

Παράδειγμα χρήσης στην εφαρμογή:

Όταν η εφαρμογή χρειάζεται να φορτώσει προτάσεις ταξιδιών βασισμένες στο προφίλ του χρήστη, το σύστημα εξαρτήσεων εξασφαλίζει ότι όλα τα απαραίτητα στοιχεία (όπως το view model και η υπηρεσία αναζήτησης) είναι ήδη διαθέσιμα και έτοιμα προς χρήση, χωρίς ο προγραμματιστής να χρειαστεί να τα δημιουργήσει χειροκίνητα.

Με αυτόν τον τρόπο, το Hilt προσφέρει όχι μόνο ευκολία και αυτοματοποίηση, αλλά και μια σταθερή βάση για τη μακροχρόνια βιωσιμότητα και εξέλιξη της εφαρμογής.



Παράδειγμα Dependency Injection με Hilt

6.9 Εργαλεία UX/UI και Σχεδιαστική Προσέγγιση

6.9.1 Εισαγωγή

Στο πλαίσιο της ανάπτυξης της εφαρμογής "Random Traveller", ιδιαίτερη έμφαση δόθηκε στον σχεδιασμό της διεπαφής χρήστη (User Interface - UI) και της εμπειρίας χρήστη (User Experience - UX). Η φιλοσοφία του σχεδιασμού επικεντρώθηκε στην απλότητα, την ευκολία πλοήγησης και τη φιλικότητα προς τον τελικό χρήστη. Για την υποστήριξη αυτής της διαδικασίας, χρησιμοποιήθηκαν εξειδικευμένα εργαλεία σχεδιασμού, με κυριότερο το Figma, το οποίο αξιοποιήθηκε σε όλα τα στάδια σχεδίασης της εφαρμογής.

6.9.2 Επιλογή Figma

Το Figma αποτελεί ένα από τα πιο σύγχρονα εργαλεία σχεδιασμού διεπαφών, προσφέροντας δυνατότητες συνεργατικής εργασίας σε πραγματικό χρόνο, δημιουργία προτύπων (prototypes), καθώς και οργάνωση των σχεδίων σε αρχεία και frames. Χρησιμοποιήθηκε για:

Σχεδίαση των οθονών της εφαρμογής (π.χ. φόρμα αναζήτησης, αποτελέσματα, αποθηκευμένες αναζητήσεις).

- Επιλογή χρωματικής παλέτας και τυπογραφίας.
- Τοποθέτηση στοιχείων με βάση τη λογική του responsive design.
- Προεπισκόπηση της εμπειρίας χρήστη μέσω διαδραστικών πρωτοτύπων.

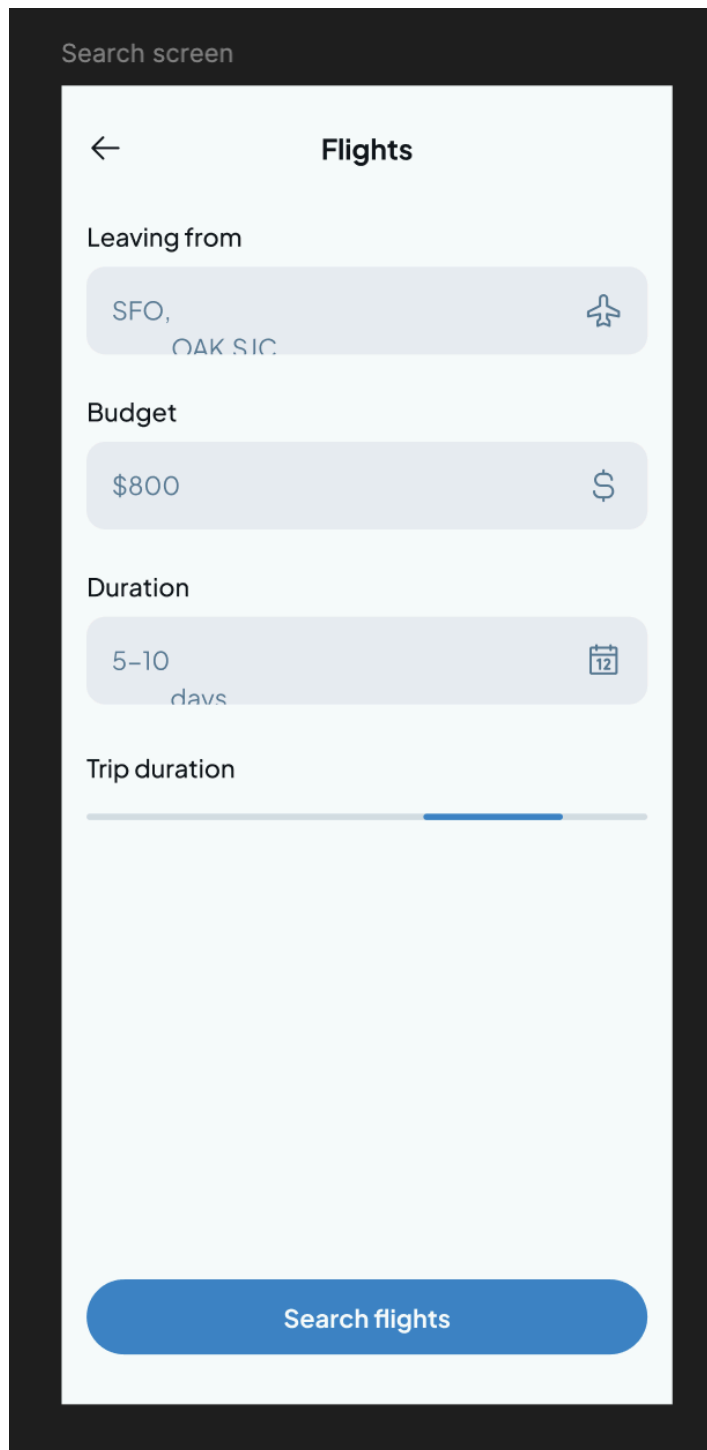
6.9.3 Αρχές Σχεδιασμού

Κατά τον σχεδιασμό της διεπαφής, τηρήθηκαν βασικές αρχές του UX/UI design:

- **Απλότητα (Simplicity):** Μείωση της πολυπλοκότητας μέσω περιορισμένου αριθμού επιλογών ανά οθόνη.
- **Συνοχή (Consistency):** Επαναχρησιμοποίηση των ίδιων στοιχείων πλοήγησης και κουμπιών σε όλες τις οθόνες.
- **Ιεράρχηση (Hierarchy):** Σαφής ιεράρχηση πληροφορίας με βάση την προτεραιότητα του κάθε στοιχείου.
- **Αναγνωσιμότητα (Readability):** Επιλογή ευανάγνωστης γραμματοσειράς και κατάλληλων μεγεθών κειμένου.
- **Ανταπόκριση (Responsiveness):** Προσαρμοστικότητα των οθονών σε διάφορες αναλύσεις και συσκευές Android.

6.9.4 Παραδείγματα Σχεδίασης

- Αρχική Οθόνη (SearchScreen): Πεδία εισαγωγής για αεροδρόμιο, budget και ημέρες, καθώς και κουμπί αναζήτησης.
- Οθόνη Αποτελεσμάτων: Παρουσίαση προτεινόμενων προορισμών με σύντομες πληροφορίες κόστους και διάρκειας.
- Αποθηκευμένες Αναζητήσεις: Καρτέλες με προηγούμενες αναζητήσεις του χρήστη για εύκολη επαναφορά.



Mockup της Αρχικής Οθόνης (SearchScreen) στο Figma

6.9.5 Συμπεράσματα

Η χρήση του Figma και η έμφαση στον καλό UX/UI σχεδιασμό ενίσχυσαν τη συνολική ποιότητα της εφαρμογής "Random Traveller". Το τελικό αποτέλεσμα είναι μια σύγχρονη, ελκυστική και λειτουργική διεπαφή που εξυπηρετεί άμεσα τον σκοπό της εφαρμογής: τη γρήγορη και αυθόρμητη αναζήτηση ταξιδιωτικών προορισμών με βάση βασικά προσωπικά κριτήρια.

Κεφάλαιο 7ο: Ροή εφαρμογής

7.1 Εισαγωγή

Η εφαρμογή Random Traveller έχει σχεδιαστεί με στόχο να προσφέρει μια εύχρηστη, σαφή και ευχάριστη εμπειρία στον χρήστη, δίνοντας έμφαση στη λειτουργικότητα, την απόκριση και την εξατομίκευση. Ο σχεδιασμός της βασίζεται στη δυνατότητα εξατομικευμένης πλοήγησης και αποθήκευσης δεδομένων αναζήτησης για χρήστες που επιλέγουν να εγγραφούν, χωρίς ωστόσο να περιορίζει τη βασική λειτουργικότητα της εφαρμογής.

Κάθε χρήστης μπορεί να πραγματοποιήσει αναζήτηση ταξιδιωτικών προορισμών μέσω προκαθορισμένων φίλτρων, όπως το διαθέσιμο χρηματικό ποσό, η χρονική διάρκεια του ταξιδιού και η απόσταση από το σημείο αναχώρησης. Αυτά τα φίλτρα επιτρέπουν την παροχή στοχευμένων προτάσεων προορισμών, βασισμένων στις ανάγκες και τις προτιμήσεις του εκάστοτε χρήστη.

Η δυνατότητα δημιουργίας λογαριασμού, αν και δεν είναι υποχρεωτική για την αναζήτηση προορισμών, παρέχει επιπρόσθετα οφέλη. Οι εγγεγραμμένοι χρήστες έχουν τη δυνατότητα να αποθηκεύουν τις αναζητήσεις τους, να επανέρχονται εύκολα σε παλαιότερα αποτελέσματα και να δημιουργούν ένα προσωποποιημένο ιστορικό αναζητήσεων. Αυτό όχι μόνο εξοικονομεί χρόνο σε μελλοντικές αναζητήσεις, αλλά ενισχύει και την εμπειρία χρήσης, προσφέροντας μία αίσθηση συνέχειας και οικειότητας στην πλοήγηση εντός της εφαρμογής.

Η προσέγγιση αυτή ενισχύει τον σκοπό της εφαρμογής να προσαρμόζεται ευέλικτα στις ανάγκες του χρήστη, ανεξάρτητα από το επίπεδο δέσμευσής του. Με τον τρόπο αυτό, η Random Traveller διατηρεί μια ευρεία προσβασιμότητα, ενώ παράλληλα ενθαρρύνει την αλληλεπίδραση και τη δέσμευση με τη μορφή επιλογών που προσθέτουν αξία στην εμπειρία πλοήγησης.

7.2 Εγγραφή

Η διαδικασία εγγραφής (sign up) και σύνδεσης (sign in) των χρηστών στην εφαρμογή υλοποιείται με χρήση της υπηρεσίας Firebase Authentication, προσφέροντας έναν ασφαλή και ελαφρύ μηχανισμό ταυτοποίησης.

Όταν ο χρήστης ανοίγει την εφαρμογή, του παρέχεται η επιλογή να προχωρήσει στη δημιουργία λογαριασμού. Επιλέγοντας "Εγγραφή", μεταφέρεται σε ειδική οθόνη όπου καλείται να εισάγει μία διεύθυνση email και έναν επιθυμητό κωδικό πρόσβασης. Η εφαρμογή, μέσω του Firebase, ελέγχει αν τα στοιχεία είναι έγκυρα και αν ο λογαριασμός μπορεί να δημιουργηθεί. Σε περίπτωση επιτυχίας, η δημιουργία λογαριασμού ολοκληρώνεται και ο χρήστης συνδέεται αυτόματα στο σύστημα, χωρίς να απαιτείται πρόσθετη επιβεβαίωση (π.χ. μέσω email).

Αντίστοιχα, για τη σύνδεση, ο χρήστης επιλέγει "Είσοδος", μεταφέρεται στην αντίστοιχη φόρμα και εισάγει τα διαπιστευτήριά του (email και κωδικό). Το Firebase αναλαμβάνει τον έλεγχο ταυτοποίησης, επιστρέφοντας επιβεβαίωση σε περίπτωση σωστών στοιχείων ή κατάλληλα μηνύματα σφάλματος αν τα δεδομένα δεν είναι έγκυρα.

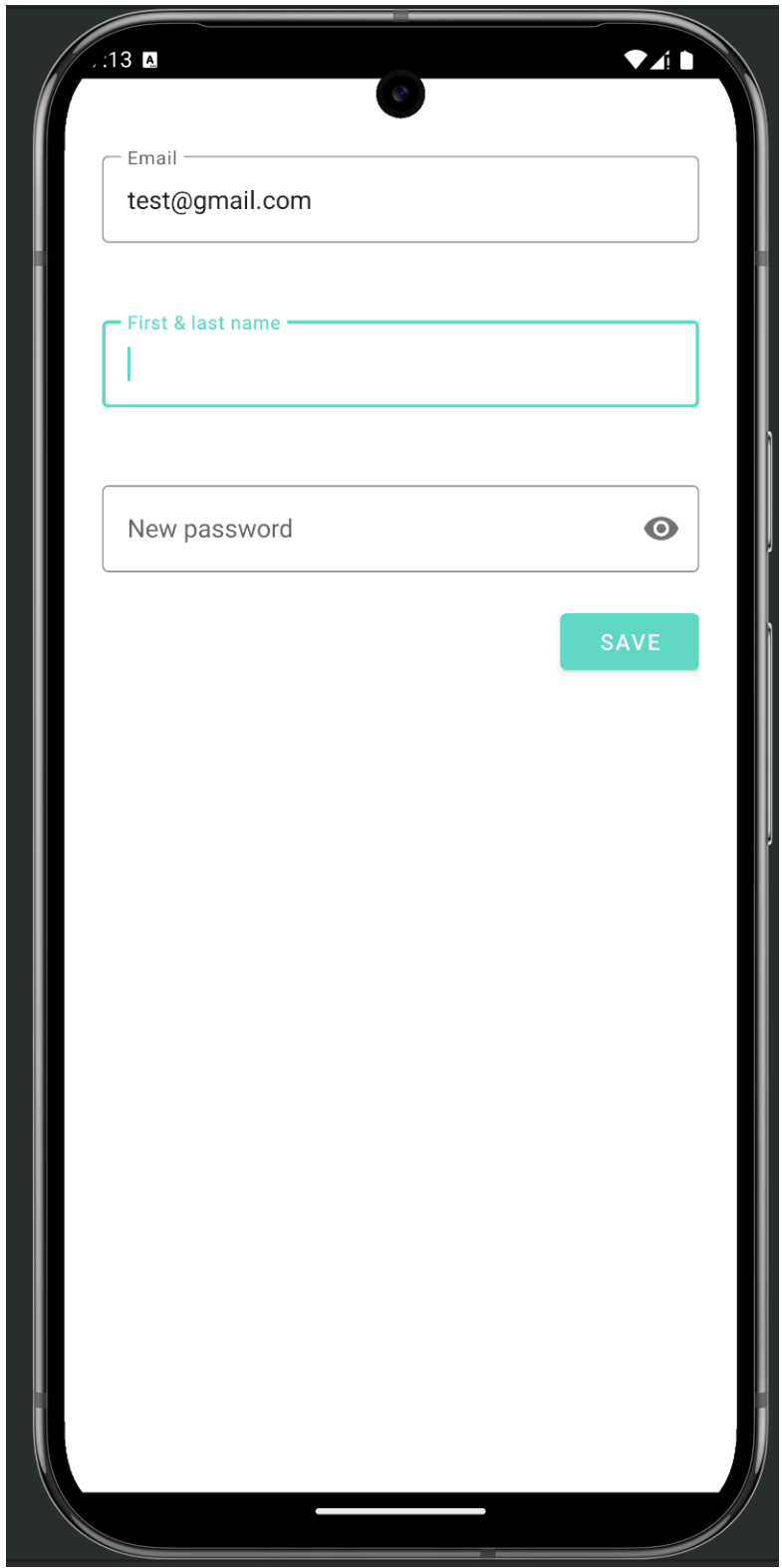
Η όλη διαδικασία είναι γρήγορη και ενσωματώνεται ομαλά στη ροή της εφαρμογής, χωρίς να αποσπά ή να κουράζει τον χρήστη. Επιπλέον, μόλις ένας χρήστης συνδεθεί επιτυχώς, παραμένει σε κατάσταση σύνδεσης και δεν χρειάζεται να επαναλαμβάνει τη διαδικασία κάθε φορά που ανοίγει την εφαρμογή, χάρη στη διαχείριση περιόδου σύνδεσης που προσφέρει το Firebase.



Αρχική οθόνη εκκίνησης της εφαρμογής Random Traveller



Οθόνη επιλογής μεθόδου σύνδεσης



Επιλογή τρόπου σύνδεσης και φόρμα εισαγωγής στοιχείων χρήστη

7.3 Αρχική οθόνη (Οθόνη αναζήτησης)

Μετά την επιτυχή είσοδο στην εφαρμογή, ο χρήστης οδηγείται στην κεντρική οθόνη αναζήτησης. Εκεί του ζητείται να εισαγάγει τρία βασικά στοιχεία: το αεροδρόμιο αναχώρησης (π.χ. Θεσσαλονίκη - SKG), το διαθέσιμο ποσό για το ταξίδι (το οποίο περιλαμβάνει εισιτήρια με επιστροφή), και τη διάρκεια του ταξιδιού σε ημέρες.

Μόλις ο χρήστης συμπληρώσει τα πεδία και πατήσει το κουμπί «Αναζήτηση», η εφαρμογή δημιουργεί ένα ερώτημα και το αποστέλλει στο backend μέσω GraphQL.

Ψευδοκώδικας λειτουργίας

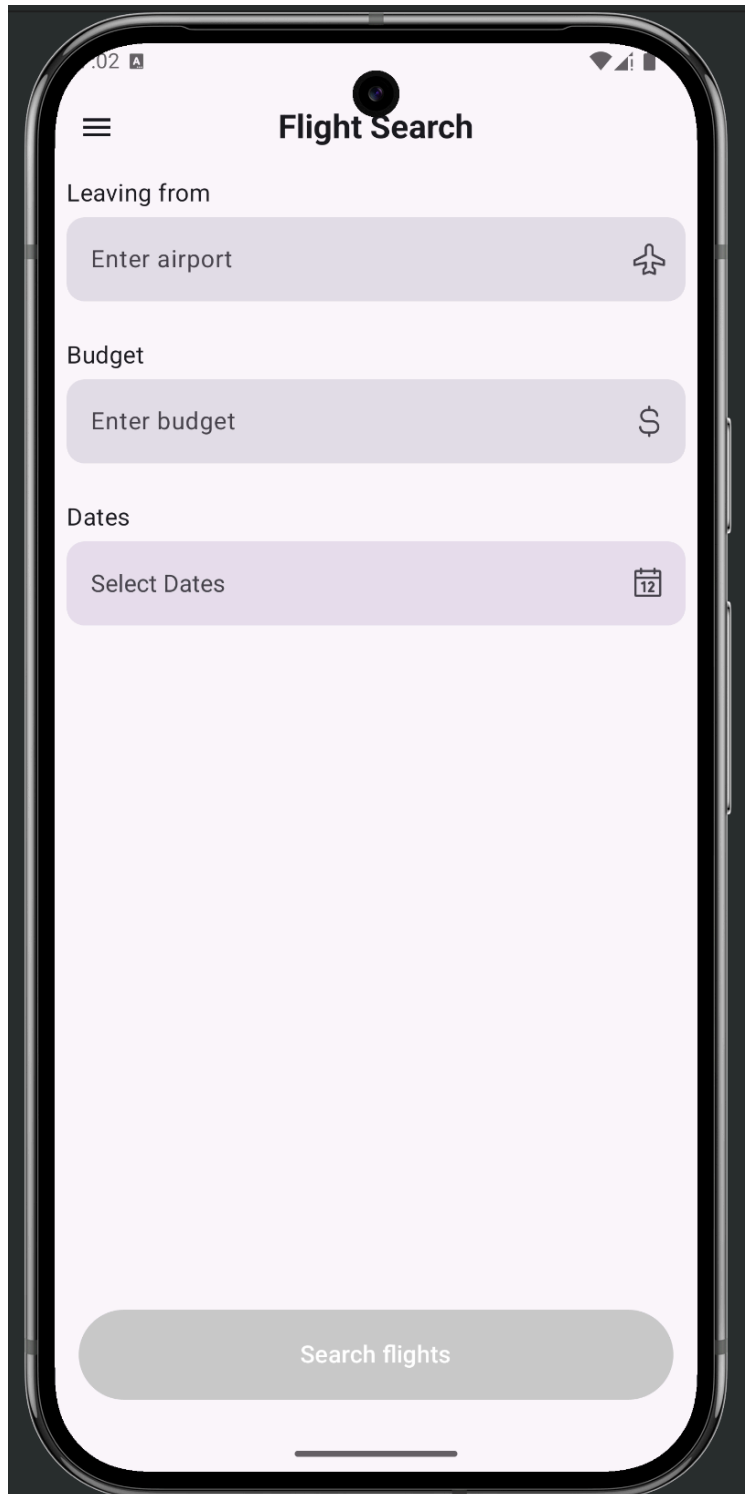
```
Όταν ο χρήστης πατήσει "Αναζήτηση": -> Εισάγονται τα στοιχεία αναχώρησης, προϋπολογισμού και διάρκειας -> Δημιουργείται GraphQL query με βάση τα παραπάνω -> Αποστέλλεται αίτημα στο backend
```

Παράδειγμα λειτουργίας

Ο χρήστης εισάγει:

- Αναχώρηση: Θεσσαλονίκη (SKG)
- Προϋπολογισμός: 150€
- Διάρκεια: 4 ημέρες

Η εφαρμογή δημιουργεί ένα ερώτημα και το αποστέλλει στον server για αναζήτηση πιθανών προορισμών με βάση αυτά τα δεδομένα.



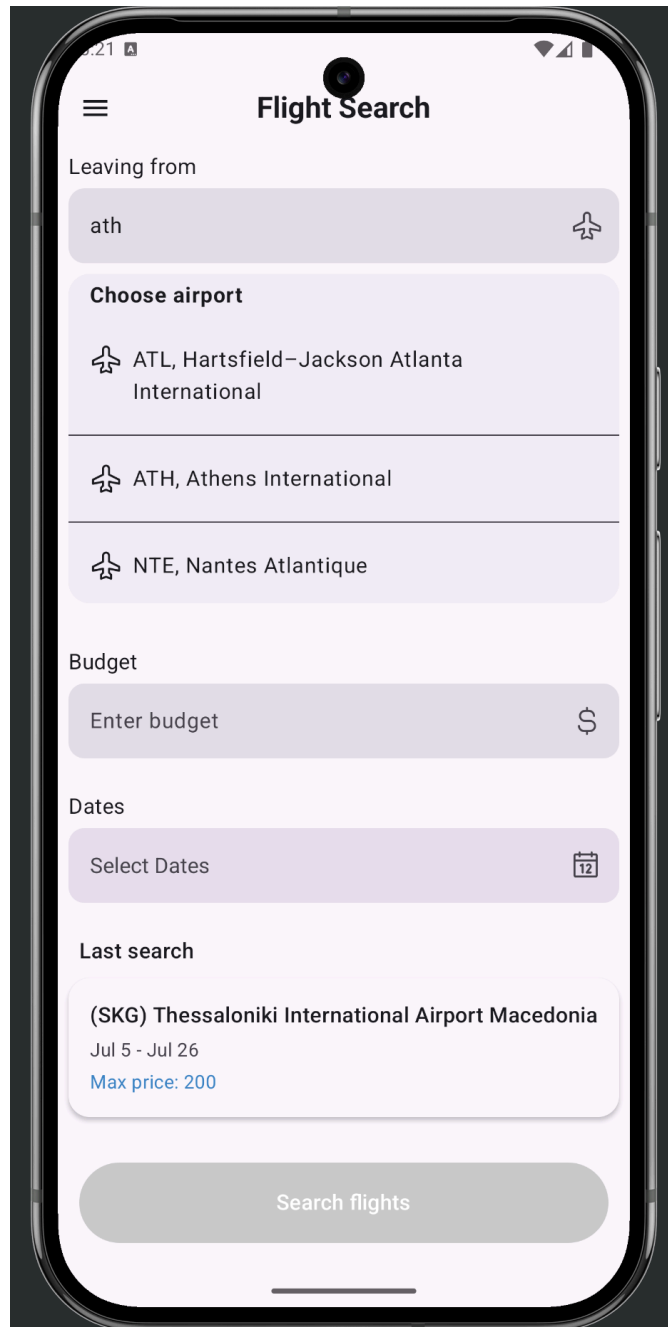
Φόρμα αναζήτησης πτήσεων στην εφαρμογή

7.4 Αυτόματη Συμπλήρωση Αεροδρομίου

Για τη διευκόλυνση και επιτάχυνση της διαδικασίας αναζήτησης, η εφαρμογή ενσωματώνει λειτουργία αυτόματης συμπλήρωσης του πεδίου αναχώρησης. Καθώς ο χρήστης αρχίζει να πληκτρολογεί το όνομα ή τον κωδικό ενός αεροδρομίου, και συγκεκριμένα από την πληκτρολόγηση των πρώτων τριών χαρακτήρων και μετά, ενεργοποιείται ένα δυναμικό φίλτρο προτάσεων.

Το σύστημα αναζητά σε πραγματικό χρόνο πιθανές αντιστοιχίες από μια προκαθορισμένη λίστα αεροδρομίων και εμφανίζει αυτόματα σχετικές επιλογές, όπως π.χ. "Thessaloniki - SKG" ή "Athens - ATH". Αυτή η δυνατότητα όχι μόνο μειώνει τον χρόνο συμπλήρωσης, αλλά επίσης περιορίζει δραστικά τα ορθογραφικά λάθη ή τις ασάφειες που θα μπορούσαν να οδηγήσουν σε λανθασμένη ή αποτυχημένη αναζήτηση.

Η λειτουργία αυτόματης συμπλήρωσης ενισχύει σημαντικά την εμπειρία χρήστη, καθώς προσφέρει αμεσότητα, προβλεψιμότητα και καθοδήγηση. Επιπλέον, καθιστά την εφαρμογή πιο προσβάσιμη σε χρήστες που δεν γνωρίζουν επακριβώς τα ονόματα ή τους διεθνείς κωδικούς των αεροδρομίων.



Οθόνη αναζήτησης πτήτης με ανοιχτό το πεδίο αεροδρομίου αναχώρησης με αυτόματη συμπλήρωση

7.5 Τελευταία αναζήτηση

Στο κάτω μέρος της αρχικής οθόνης της εφαρμογής προβάλλεται η τελευταία αναζήτηση που πραγματοποίησε ο χρήστης. Η εμφάνιση αυτής της ενότητας λειτουργεί ως λειτουργική συντόμευση, επιτρέποντας την άμεση επαναχρησιμοποίηση προηγούμενων κριτηρίων αναζήτησης, χωρίς να χρειάζεται εκ νέου εισαγωγή στοιχείων.

Η εφαρμογή αποθηκεύει το τελευταίο query τοπικά ή στο cloud (μέσω Firebase Firestore), ανάλογα με το αν ο χρήστης είναι αυθεντικοποιημένος. Στη συνέχεια, κατά την είσοδο στην αρχική οθόνη, γίνεται αυτόματη ανάκτηση των στοιχείων αυτών ώστε να εμφανιστούν έτοιμα προς χρήση.

Περιλαμβανόμενα στοιχεία:

- Το αεροδρόμιο αναχώρησης (π.χ. SKG – Θεσσαλονίκη)
- Ο προϋπολογισμός που ορίστηκε στην προηγούμενη αναζήτηση
- Η επιλεγμένη διάρκεια ταξιδιού σε ημέρες

Ψευδοκώδικας λειτουργίας

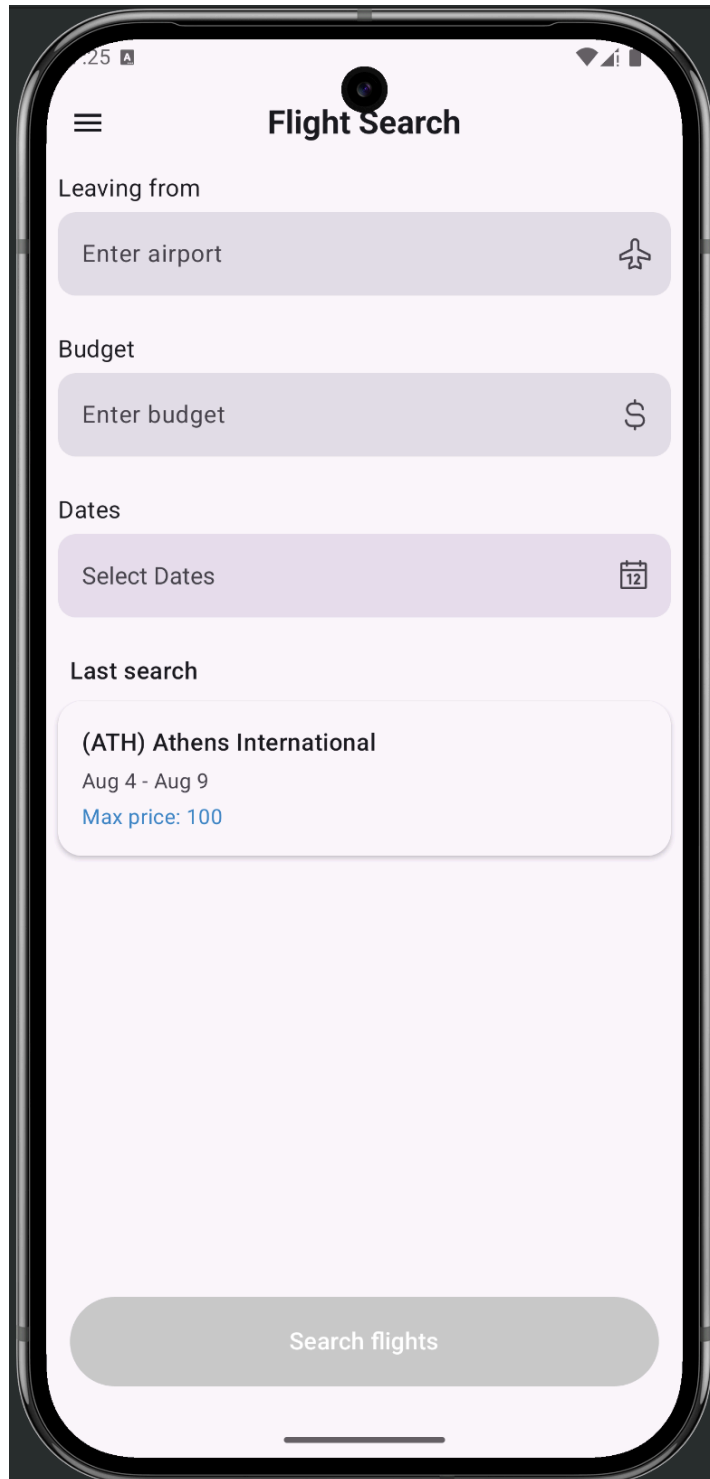
```
Κατά το άνοιγμα της αρχικής οθόνης:  
Αν υπάρχει συνδεδεμένος χρήστης:  
  -> Ζήτα το τελευταίο query από το Firestore  
Αλλιώς:  
  -> Δες αν υπάρχουν αποθηκευμένα δεδομένα τοπικά  
  
Αν βρεθούν δεδομένα:  
  -> Προετοίμασε UI στοιχείο "Τελευταία αναζήτηση"  
  -> Εμφάνισε τα πεδία (αναχώρηση, προϋπολογισμός, διάρκεια)  
  
Όταν ο χρήστης πατήσει "Χρήση τελευταίας αναζήτησης":  
  -> Αυτόματη συμπλήρωση της φόρμας με τα αποθηκευμένα δεδομένα
```

Παράδειγμα χρήσης

Ο χρήστης, σε προηγούμενη συνεδρία, έκανε αναζήτηση με τα εξής στοιχεία:

- Αναχώρηση: Θεσσαλονίκη (SKG)
- Προϋπολογισμός: 200€
- Η επιλεγμένη διάρκεια ταξιδιού Aug 4 - Aug 9

Την επόμενη φορά που ανοίγει την εφαρμογή, βλέπει στο κάτω μέρος της αρχικής οθόνης μια κάρτα με τα παραπάνω στοιχεία. Πατώντας σε αυτή την κάρτα, τα πεδία της φόρμας συμπληρώνονται αυτόματα και μπορεί να προχωρήσει σε νέα αναζήτηση άμεσα.



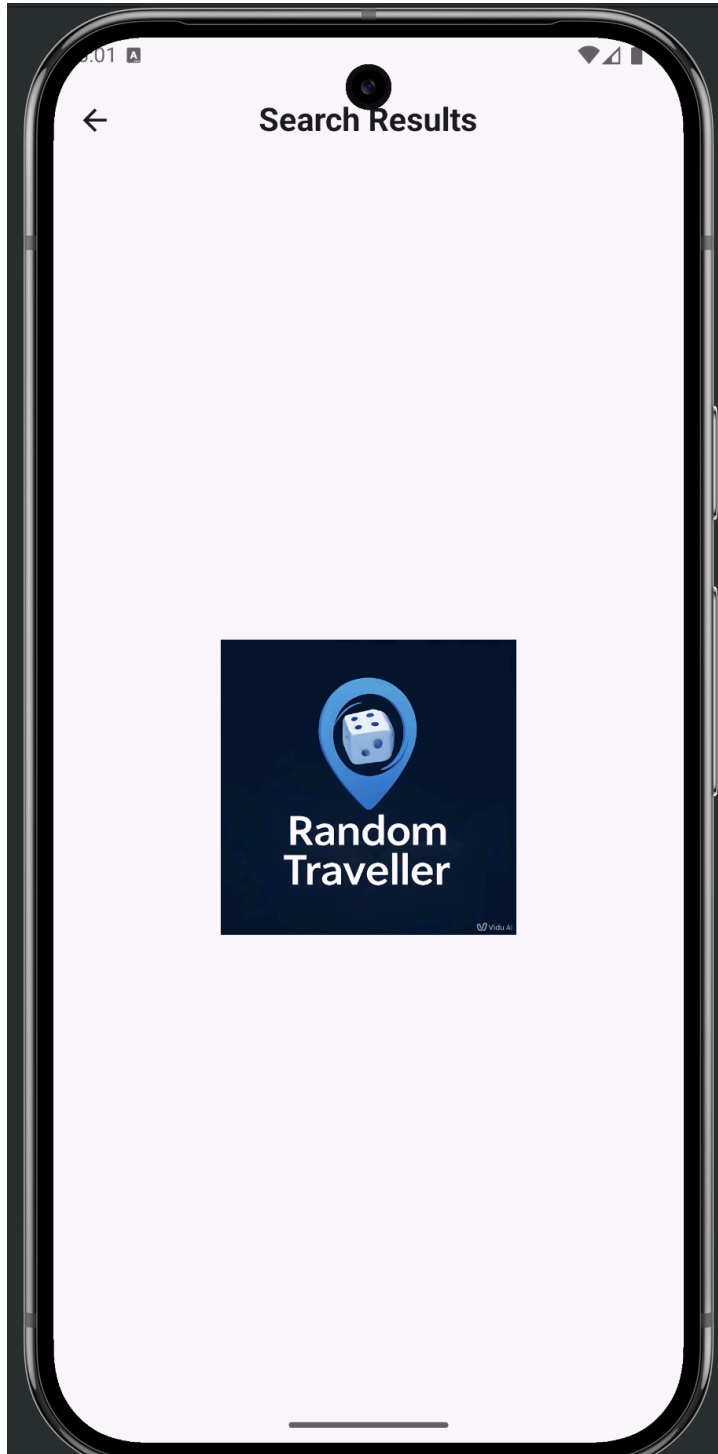
Οθόνη με Banner τελευταίας αναζήτησης

7.6 Οθόνη Αναμονής Αποτελεσμάτων

Κατά την αναζήτηση διαθέσιμων προορισμών με βάση τα κριτήρια του χρήστη (προέλευση, budget και αριθμός ημερών), η εφαρμογή "Random Traveller" παρουσιάζει μια ενδιάμεση οθόνη φόρτωσης. Η συγκεκριμένη οθόνη εμφανίζεται ενώ η εφαρμογή επικοινωνεί με εξωτερικές υπηρεσίες και επεξεργάζεται τα δεδομένα.

Για να αποφευχθεί η αίσθηση καθυστέρησης και να διατηρηθεί η εμπειρία χρήστη ευχάριστη, επιλέχθηκε η χρήση ενός διαδραστικού animation με το λογότυπο της εφαρμογής. Η ελαφριά κίνηση ή αλλαγή χρώματος του εικονιδίου δίνει στον χρήστη την αίσθηση ότι η εφαρμογή εργάζεται «ζωντανά», μειώνοντας την πιθανότητα εγκατάλειψης της διαδικασίας λόγω αναμονής.

Η συγκεκριμένη επιλογή ακολουθεί τις βέλτιστες πρακτικές του UI/UX σχεδιασμού, σύμφωνα με τις οποίες η παροχή οπτικής πληροφορίας κατά τη διάρκεια της φόρτωσης όχι μόνο ενισχύει την αντίληψη ταχύτητας, αλλά και συμβάλλει στη διατήρηση της προσοχής του χρήστη.



Οθόνη Αναμονής Αποτελεσμάτων

7.7 Οθόνη Αποτελεσμάτων Πτήσεων

Η παρουσίαση των αποτελεσμάτων δεν βασίζεται απλώς σε μια μηχανική παράθεση όλων των διαθέσιμων πτήσεων που επιστρέφει το εξωτερικό API, αλλά αξιοποιεί έναν στοχευμένο αλγόριθμο φιλτραρίσματος, σχεδιασμένο ώστε να αναδεικνύει τις πιο συμφέρουσες και σχετικές επιλογές για κάθε προορισμό.

1. Ο αλγόριθμος που υλοποιείται στο FindCheapestFlightsPerDestinationUseCase ακολουθεί την εξής λογική:
2. Ανάκτηση δεδομένων από την πλατφόρμα Kiwi, με χρήση των παραμέτρων που όρισε ο χρήστης (αναχώρηση, εύρος ημερομηνιών, προϋπολογισμός).
3. Ομαδοποίηση των αποτελεσμάτων ανά προορισμό, με ταυτόχρονο φιλτράρισμα εγγραφών που δεν περιλαμβάνουν έγκυρο όνομα πόλης.
4. Εντοπισμός της φθηνότερης επιλογής εντός κάθε ομάδας πτήσεων, με βάση την τιμή του εισιτηρίου, διατηρώντας τις υπόλοιπες παραμέτρους σταθερές.
5. Σύνθεση τελικής λίστας, η οποία περιλαμβάνει μία επιλεγμένη πτήση ανά προορισμό, προσφέροντας έτσι στον χρήστη μια καθαρή, αποδοτική και εξειδικευμένη προβολή επιλογών.

Η συγκεκριμένη προσέγγιση διασφαλίζει ότι ο χρήστης δεν κατακλύζεται από περιττές πληροφορίες, ενώ παράλληλα εστιάζει στις πιο οικονομικά και χρονικά αποδοτικές λύσεις, διευκολύνοντας τη διαδικασία επιλογής και ενισχύοντας τη συνολική εμπειρία πλοήγησης.

Ψευδοκώδικας λειτουργίας

```
Ανάκτησε όλες τις πτήσεις από το API με βάση τα φίλτρα
Αν δεν επιστραφούν αποτελέσματα:
  -> Επιστροφή κενής λίστας

Ομαδοποίησε τις πτήσεις ανά τελική πόλη προορισμού
Αφαίρεσε προορισμούς χωρίς έγκυρο όνομα

Για κάθε προορισμό:
  -> Εντόπισε την προσφορά με τη χαμηλότερη τιμή

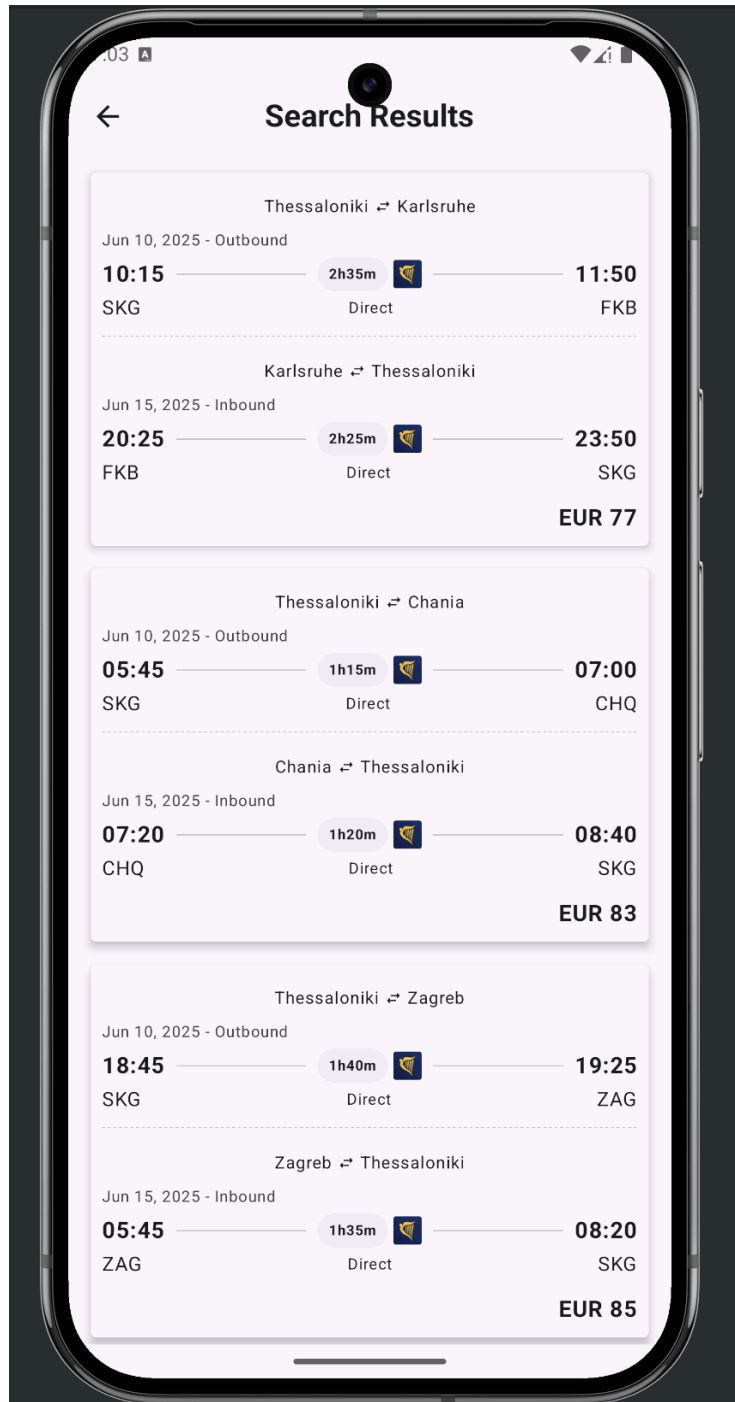
Επιστροφή λίστας με τις φθηνότερες πτήσεις ανά προορισμό
```

Παράδειγμα χρήσης

Ο χρήστης ορίζει αναχώρηση από Θεσσαλονίκη (SKG) με:

- Προϋπολογισμό: 180€
- Διάρκεια ταξιδιού: 4 ημέρες

Το σύστημα ανακτά δεκάδες διαθέσιμες πτήσεις για διαφορετικούς προορισμούς. Αντί να εμφανιστούν όλες, η εφαρμογή μέσω του παραπάνω αλγορίθμου εμφανίζει μόνο μία επιλεγμένη πτήση ανά πόλη, αυτή με το χαμηλότερο κόστος. Ο χρήστης έχει έτσι στη διάθεσή του μία στοχευμένη και περιεκτική λίστα από πραγματικά συμφέρουσες προτάσεις.



Οθόνη εμφάνισης αποτελεσμάτων αναζήτησης πτήσεων

7.8 Οθόνη Αποθηκευμένων Αναζητήσεων

Η οθόνη Saved Searches παρέχει στον χρήστη πρόσβαση σε όλες τις πτήσεις που έχει αποθηκεύσει σε προηγούμενες συνεδρίες. Κάθε αποθηκευμένη αναζήτηση εμφανίζεται σε ξεχωριστή κάρτα, η οποία περιλαμβάνει τα βασικά κριτήρια με τα οποία πραγματοποιήθηκε η αρχική αναζήτηση.

Η διεπαφή είναι σχεδιασμένη με στόχο την ευχρηστία και την καθαρή οπτική παρουσίαση: οι κάρτες διαθέτουν στρογγυλεμένες γωνίες και χρησιμοποιούν απαλή χρωματική παλέτα ώστε να ενισχύουν την αναγνωσιμότητα.

Ο χρήστης έχει τη δυνατότητα να επαναχρησιμοποιήσει μια αποθηκευμένη αναζήτηση αγγίζοντας την αντίστοιχη κάρτα ή να τη διαγράψει εύκολα με ένα slide gesture προς τα αριστερά. Η ενέργεια αυτή αφαιρεί τη σχετική εγγραφή τόσο από την οθόνη όσο και από τη βάση δεδομένων, επιτρέποντας την εύκολη διαχείριση των παλαιών αναζητήσεων.

Ψευδοκώδικας λειτουργίας

Κατά το άνοιγμα της οθόνης Saved Searches:

-> Αν ο χρήστης είναι συνδεδεμένος:

Ζήτα τις αποθηκευμένες αναζητήσεις από το Firestore

Εμφάνισε κάθε αναζήτηση ως κάρτα με:

- Προορισμό (όνομα + κωδικός)
- Ημέρες ταξιδιού
- Προϋπολογισμό

Όταν ο χρήστης κάνει slide μία κάρτα προς τα αριστερά:

-> Προβάλλεται επιλογή διαγραφής

-> Αν επιβεβαιωθεί η ενέργεια:

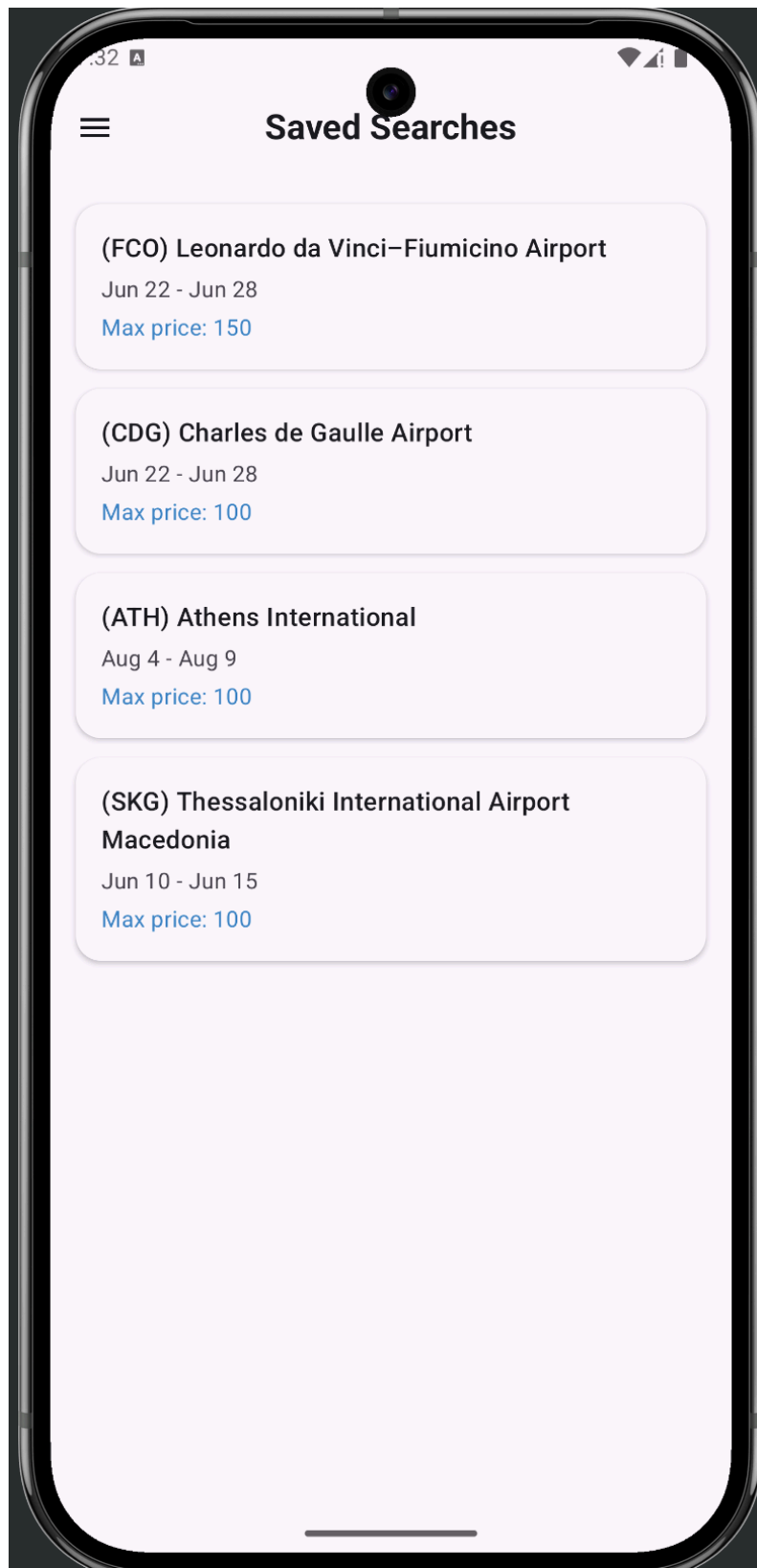
Στείλε εντολή διαγραφής της αναζήτησης στο Firestore

Αφαίρεσε την κάρτα από το UI

Ο χρήστης έχει αποθηκεύσει μια αναζήτηση με τα εξής στοιχεία:

- Προορισμός: Ρώμη (FCO)
- Διάρκεια: 3 ημέρες
- Προϋπολογισμός: 120€

Κατά την είσοδό του στην οθόνη Saved Searches, εμφανίζεται μια κάρτα με αυτά τα στοιχεία. Μπορεί να πατήσει την κάρτα για να επαναφορτώσει την ίδια αναζήτηση ή να την διαγράψει κάνοντας slide προς τα αριστερά και επιβεβαιώνοντας τη διαγραφή.



Οθόνη αποθηκευμένων αναζητήσεων

7.9 Πλευρικό μενού πλοήγησης

Η εφαρμογή "Random Traveller" διαθέτει ένα πλευρικό μενού πλοήγησης (navigation drawer), το οποίο λειτουργεί ως ένας από τους βασικότερους μηχανισμούς αλληλεπίδρασης με τον χρήστη. Το μενού αυτό παρέχει άμεση και γρήγορη πρόσβαση στις βασικές λειτουργίες της εφαρμογής, προσφέροντας μια σαφή και οργανωμένη δομή πλοήγησης. Ενεργοποιείται μέσω του γνωστού εικονιδίου "hamburger", το οποίο βρίσκεται στην πάνω αριστερή γωνία της οθόνης, και ανοίγει με μια ομαλή, συρόμενη κίνηση από τα αριστερά προς τα δεξιά.

Η επιλογή αυτής της μορφής πλοήγησης δεν είναι τυχαία. Το navigation drawer θεωρείται μία από τις πιο αποδοτικές μεθόδους πλοήγησης σε εφαρμογές Android, καθώς επιτρέπει την συγκέντρωση πολλαπλών λειτουργιών σε ένα κρυφό αλλά άμεσα προσβάσιμο σημείο.

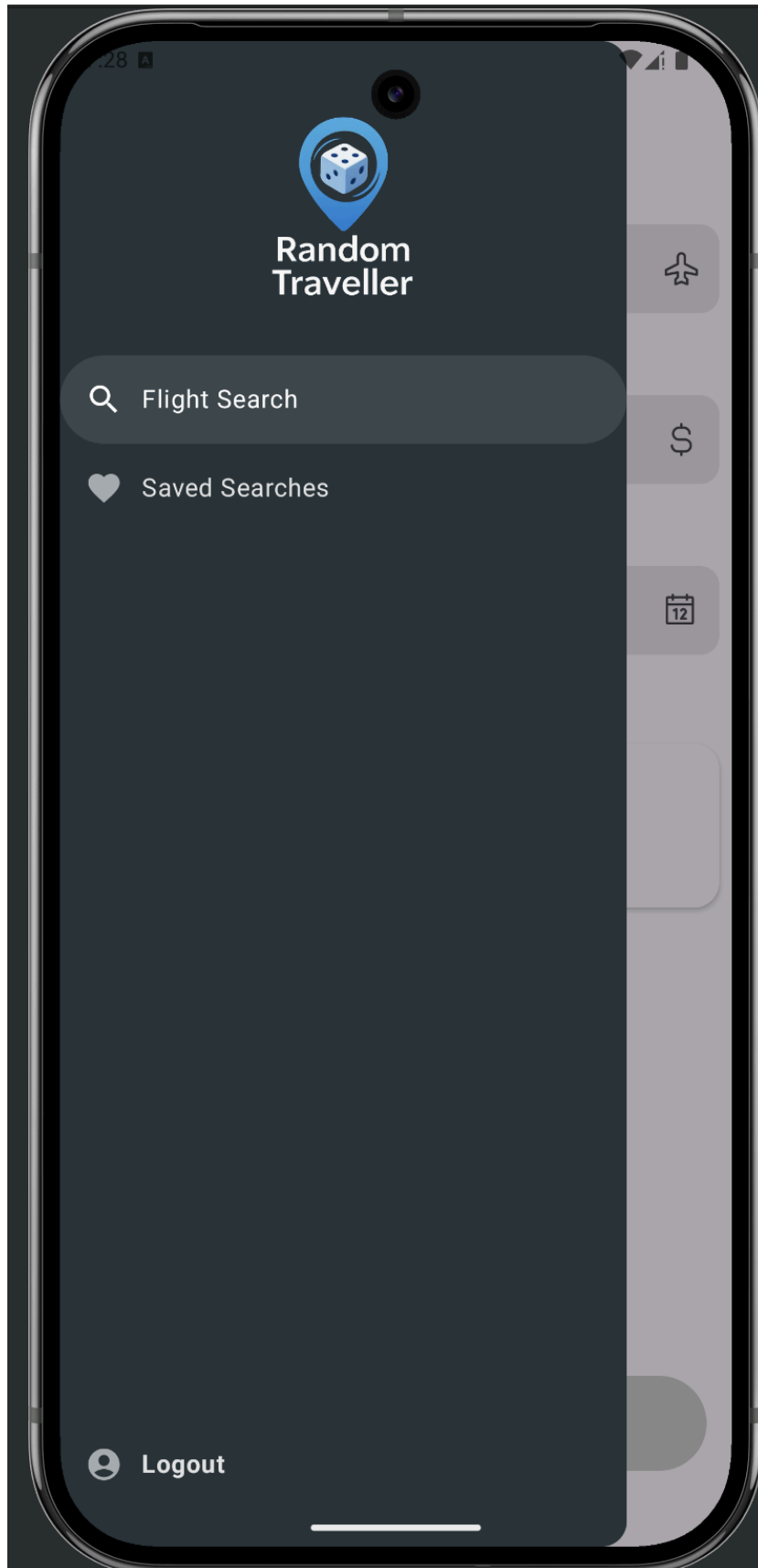
Το περιεχόμενο του μενού έχει διαρθρωθεί με τέτοιο τρόπο ώστε να ανταποκρίνεται στις συχνότερες ανάγκες των χρηστών. Οι επιλογές που προσφέρονται είναι οι εξής:

- Αρχική Οθόνη Αναζήτησης: Επιτρέπει στον χρήστη να επιστρέψει γρήγορα στη φόρμα εισαγωγής κριτηρίων ταξιδιού, χωρίς να χρειάζεται να πλοηγηθεί προς τα πίσω με πολλαπλά taps.
- Αποθηκευμένες Αναζητήσεις: Παρέχει άμεση πρόσβαση στο ιστορικό των αναζητήσεων που έχει αποθηκεύσει ο χρήστης. Η δυνατότητα αυτή είναι ιδιαίτερα χρήσιμη για χρήστες που πραγματοποιούν συχνές αναζητήσεις και επιθυμούν να επιστρέφουν σε προηγούμενες επιλογές χωρίς να εισάγουν ξανά όλα τα δεδομένα.
- Αποσύνδεση: Μέσω αυτής της επιλογής, ο χρήστης μπορεί να τερματίσει τη συνεδρία του. Η εφαρμογή φροντίζει να διαγράψει το authentication token για λόγους ασφαλείας και να επιστρέψει τον χρήστη στην οθόνη σύνδεσης.

Η σχεδίαση του μενού έχει δομηθεί με βάση τις αρχές της ευχρηστίας και της καθαρότητας. Κάθε επιλογή συνοδεύεται από αντίστοιχο εικονίδιο, το οποίο βοηθά στην αναγνώριση της λειτουργίας με μια ματιά. Επιπλέον, γίνεται χρήση κατάλληλων χρωματικών αντιθέσεων ώστε να διασφαλίζεται η αναγνωσιμότητα ακόμη και σε συσκευές με χαμηλή φωτεινότητα ή σε συνθήκες έντονου φωτισμού.

Ένα ακόμη σημαντικό χαρακτηριστικό του navigation drawer είναι ότι προσαρμόζεται οπτικά με βάση το θέμα της εφαρμογής και το μέγεθος οθόνης της συσκευής. Έτσι, διατηρείται μια συνεπής αισθητική εμπειρία σε κινητά τηλέφωνα αλλά και tablets, ανεξαρτήτως αν βρίσκονται σε κατακόρυφη ή οριζόντια λειτουργία. Το μενού αυτό αποδεικνύεται εξαιρετικά χρήσιμο, ειδικά για επαναλαμβανόμενους χρήστες που επιθυμούν να πλοηγηθούν γρήγορα ανάμεσα σε λειτουργίες χωρίς να ακολουθήσουν πολύπλοκες διαδρομές. Η παρουσία του ενισχύει τη συνολική αίσθηση προβλεψιμότητας και ευκολίας, δύο παράγοντες που παίζουν καθοριστικό ρόλο στη διατήρηση και ικανοποίηση των χρηστών μιας εφαρμογής.

Με βάση όλα τα παραπάνω, είναι σαφές ότι το πλευρικό μενού της εφαρμογής δεν αποτελεί απλώς ένα λειτουργικό εργαλείο, αλλά έναν κρίσιμο κόμβο στη δομή της πλοήγησης και της εμπειρίας χρήστη.



Πλευρικό μενού πλοήγηση

7.10 Οθόνη Μετάβασης στην Kiwi.com

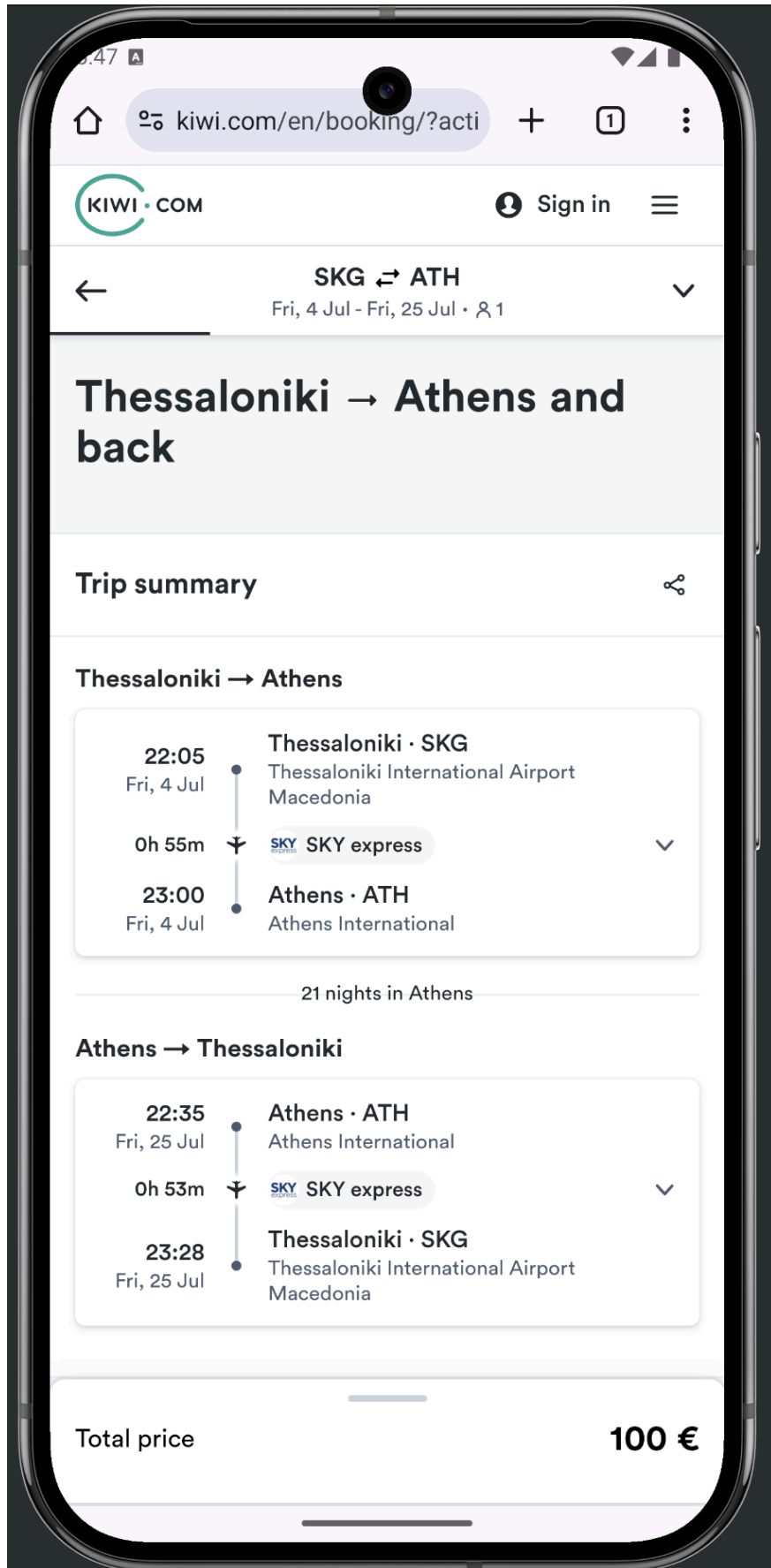
Στην τελευταία φάση της εμπειρίας του χρήστη, όταν έχει ήδη επιλέξει έναν προτεινόμενο ταξιδιωτικό προορισμό, η εφαρμογή "Random Traveller" τον μεταφέρει αυτόματα στην εξωτερική πλατφόρμα κρατήσεων Kiwi.com. Η μετάβαση αυτή γίνεται με τρόπο άμεσο, χωρίς ενδιάμεσες οθόνες ή επιβεβαιώσεις, και ανοίγει απευθείας στον προεπιλεγμένο browser της συσκευής.

Αυτό που καθιστά τη διαδικασία ξεχωριστή είναι το γεγονός ότι ο σύνδεσμος δεν οδηγεί απλώς στη γενική αρχική σελίδα της Kiwi, αλλά τοποθετεί αμέσως τον χρήστη στο τελικό στάδιο της κράτησης. Εκεί εμφανίζονται όλα τα απαραίτητα στοιχεία του ταξιδιού, τα οποία έχουν ήδη καθοριστεί μέσα από την εφαρμογή: αναχώρηση, επιστροφή, ημερομηνίες, ώρες, διάρκεια, αεροπορική εταιρεία, και συνολική τιμή. Ο χρήστης μπορεί, χωρίς επιπλέον ενέργειες, να προχωρήσει στην ολοκλήρωση της κράτησης.

Η αμεσότητα αυτής της εμπειρίας μειώνει τις πιθανότητες εγκατάλειψης της διαδικασίας και ενισχύει την αίσθηση αποτελεσματικότητας. Το γεγονός ότι ο χρήστης αποφεύγει νέα αναζήτηση ή συμπλήρωση στοιχείων ενισχύει σημαντικά τη χρηστικότητα της εφαρμογής και καθιστά τη ροή ευχάριστη και γρήγορη.

Η δυνατότητα αυτή δεν περιορίζεται απλώς σε ένα τεχνικό χαρακτηριστικό, αλλά αποτελεί βασικό στοιχείο ολοκλήρωσης της ταξιδιωτικής εμπειρίας: από την έμπνευση και την αναζήτηση μέχρι την κράτηση, όλα υλοποιούνται σε λίγα μόνο βήματα. Η εφαρμογή αποκτά έτσι μια ουσιαστική και πρακτική διάσταση, ιδιαίτερα χρήσιμη για αυθόρμητους ταξιδιώτες που θέλουν να δράσουν άμεσα.

Επιπλέον, η σύνδεση με μία αξιόπιστη και διεθνώς αναγνωρισμένη πλατφόρμα όπως η Kiwi προσδίδει κύρος και εμπιστοσύνη στην εφαρμογή. Ο χρήστης γνωρίζει πως μπορεί να βασιστεί σε μια αξιόπιστη υπηρεσία για την ολοκλήρωση της κράτησης, κάτι που ενισχύει την εμπιστοσύνη του στην εφαρμογή συνολικά. Το γεγονός ότι η μετάβαση πραγματοποιείται απρόσκοπτα και χωρίς περιττές παρεμβολές συμβάλλει στη δημιουργία μιας θετικής εντύπωσης που ενισχύει την πιθανότητα επαναχρησιμοποίησης της εφαρμογής στο μέλλον.



Οθονη πρωτου βήματος τις διαδικασίας κράτησης της πτήσης στην πλατφορμα της Kiwi.com

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] R. Meier and I. Lake, Professional Android, 4th ed., Indianapolis, IN: Wiley, 2018.
- [2] D. Griffiths and D. Griffiths, Head First Android Development: A Brain-Friendly Guide, 2nd ed., Sebastopol, CA: O'Reilly Media, 2017.
- [3] M. Gargenta and M. Nakamura, Learning Android, 2nd ed., Sebastopol, CA: O'Reilly Media, 2014.
- [4] Firebase, “Firebase Documentation,” Google, [Online]. Available: <https://firebase.google.com/docs>
- [5] Android Developers, “Developer Guides,” Google, [Online]. Available: <https://developer.android.com/get-started/overview>
- [6] Kiwi.com, “Kiwi API Documentation,” [Online]. Available: <https://docs.kiwi.com>
- [7] D. Stevenson, “Patterns for security with Firebase: offload client work to Cloud Functions,” Available: <https://medium.com/firebase-developers/patterns-for-security-with-firebase-offload-client-work-to-cloud-functions-7c420710f07>
- [8] Google Developer Experts, “Modular Navigation with Jetpack Compose,” Medium, [Online]. Available: <https://medium.com/google-developer-experts/modular-navigation-with-jetpack-compose-fda9f6b2bef7>
- [9] T. Repcik, “Dependency Injection with Hilt in Android Development,” Medium, [Online]. Available: <https://tomas-repcik.medium.com/dependency-injection-with-hilt-in-android-development-e23fc636d65c>
- [10] GameDev Academy, “Android Networking Tutorial With AsyncTask,” [Online]. Available: <https://gamedevacademy.org/android-networking-tutorial-with-async-task/>