TUNIS BUSINESS SCHOOL
UNIVERSITY OF TUNIS

جـــامعة تـــونـــس
Depuis 1958

Republic Of Tunisia
Ministry Of Higher Education And Scientific Research
University Of Tunis

Tunis Business School

Travee: A Trip Planner Application

Final Report

Submitted in Partial Fulfillment of the Requirements for the Advanced Programming, IT 370

Supervised By:

Prof. Najet Boughanmi

Written By:

Saja Moussa

Junior, Majoring in Information Technology & Minoring in Business Analysis

May, 2025

Contents

List of Figures

Chapter 1

Abstraction

Travee is an intelligent mobile solution aimed at enhancing the way individuals approach trip coordination. It empowers users to discover flight options that align with their financial preferences, travel timeframe, and geographical context. Beyond transportation, the application dynamically curates destination-specific experiences, offering engaging activity ideas tailored to each journey.

By leveraging real-time data and external service integrations, Travee delivers an adaptive and responsive planning environment. It serves a wide range of users seeking convenience, personalization, and a streamlined interface to oversee all aspects of their travel in one place.

Chapter 2

Introduction

2.1 Purpose of the Application

In the era of digital mobility, planning travel efficiently is more essential than ever. *Travee* is a Kotlin-based Android application developed to assist users in organizing their travel experiences with ease and intelligence. The app helps users discover affordable flight options based on their budget, length of stay, and departure location, and complements this by suggesting relevant activities at their destination. Through an adaptive design and intelligent backend logic, Travee simplifies the entire trip planning process into one integrated platform.

2.2 Problem Statement

Travelers often face the challenge of piecing together flights, activities, and logistics from multiple disconnected sources. This fragmentation leads to unnecessary complexity, especially when considering budget constraints, time limitations, and the need for personalized recommendations. Travee addresses these issues by providing a unified platform that aggregates flight data and activity suggestions in a tailored and user-friendly manner.

2.3 Target Audience

The application targets tech-savvy travelers, students, professionals, and frequent flyers who prefer personalized and consolidated travel planning. Whether preparing a vacation or a work trip, users benefit from automated suggestions, centralized trip data, and a simplified way to manage their entire travel experience on one interface.

2.4 Key Features

The core functionalities of the Travee application include:

- User registration and secure authentication.
- Trip creation, modification, and saving to a persistent favorites list.
- Integration with the **TravelPayouts/Aviasales API** to fetch real-time flight pricing, departure/return times, airline codes, airport information, and booking links.
- Use of the **Groq API** for AI-generated activity suggestions tailored to the user's selected destination and trip duration.
- Manual data mapping for countries, airports, and airline names using hardcoded datasets for improved user readability.
- A clean and interactive UI built with Jetpack Compose to provide a responsive user experience.

2.5 Project Objectives

This project aims to build a modular, scalable, and intelligent travel planning assistant using Kotlin and Jetpack Compose. The application integrates real-time external APIs and internal logic to deliver a personalized experience for each user. The emphasis is on usability, performance, and future extensibility to accommodate additional features such as accommodation booking, reviews, and multilingual support.

Chapter 3

Technologies Used

This section outlines the primary technologies and tools employed in the development of the *Travee* application. Each technology was chosen based on its capabilities, compatibility with Android development, and relevance to the application's objectives.

3.1 Kotlin

Kotlin is the main programming language used for building the Travee application. It is officially supported by Google for Android development and offers concise syntax, improved safety features, and full interoperability with Java. Kotlin was selected for its modern features and its compatibility with Jetpack Compose.

3.2 Jetpack Compose

Jetpack Compose is Android's modern toolkit for building native UI. It simplifies UI development by using a declarative approach, allowing developers to build dynamic and responsive interfaces with less boilerplate code. Travee's user interface is entirely built using Jetpack Compose, ensuring smooth user interaction and maintainable code structure.

3.3 Firebase

Firebase is used for user authentication and cloud-based data management in the application. It provides secure sign-in methods and enables persistent storage of user trips, preferences, and favorites. Firebase also facilitates real-time database updates and integrates easily with Android projects.

3.4 TravelPayouts (Aviasales API)

The Aviasales API, provided by TravelPayouts, is used to fetch real-time flight data including pricing, departure and return times, airlines, and booking links. It allows the application to dynamically generate flight suggestions based on user preferences such as budget, dates, and locations.

3.5 Groq API

The Groq API is leveraged to generate activity suggestions based on the destination selected by the user. Using advanced natural language processing capabilities, the API helps deliver personalized and engaging travel experiences by proposing things to do that align with the user's interests and itinerary.

Chapter 4

System Architecture

4.1 Overview

This section provides a comprehensive view of the architectural design and internal structure of the *Travee* application. It covers the key components, data flow, user interaction, and system behavior.

4.2 Use Case Diagram

The use case diagram illustrates the main interactions between the user and the system. It provides a high-level view of the application's core functionalities such as user authentication, searching for flights, saving trips, and viewing activity suggestions.

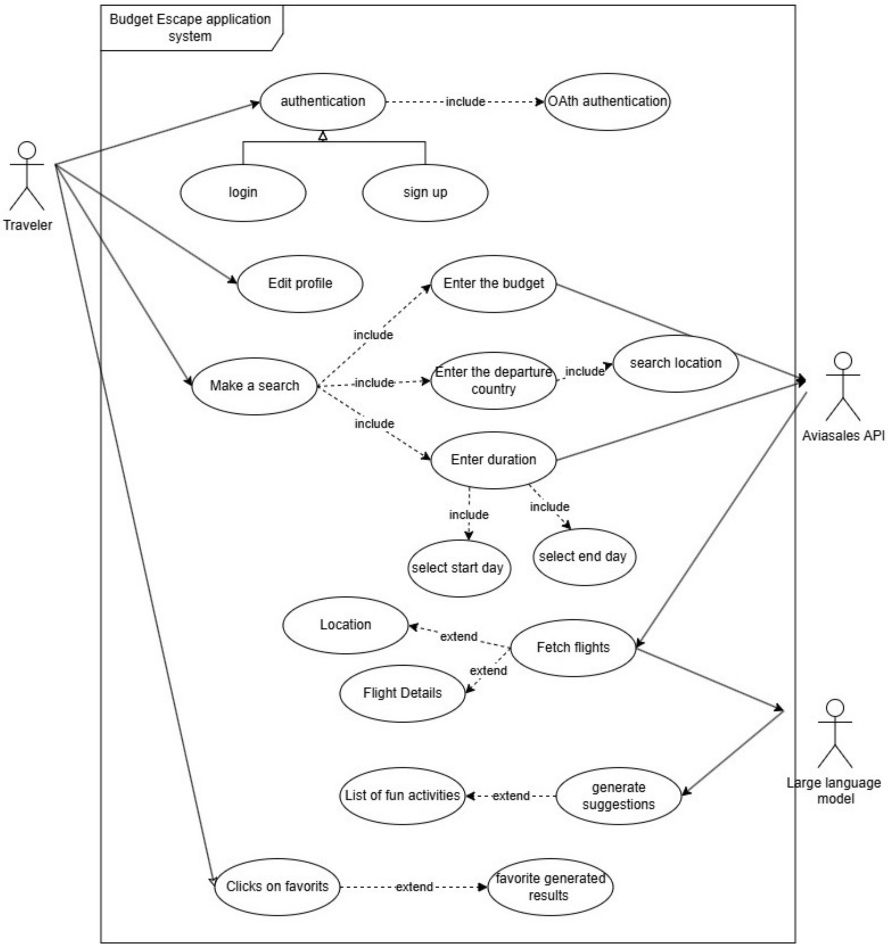


Figure 4.1: Use-Case Diagram for Trip Planner

4.3 Class Diagram

The class diagram describes the object-oriented structure of the application. It shows the main classes, their attributes, methods, and relationships such as associations.

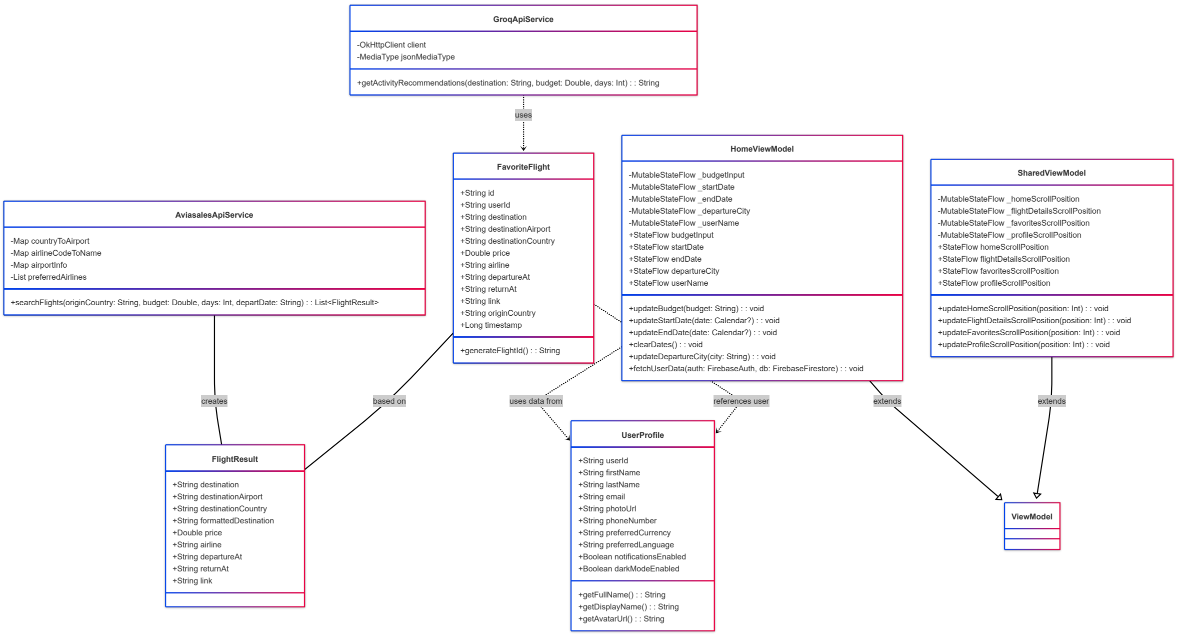
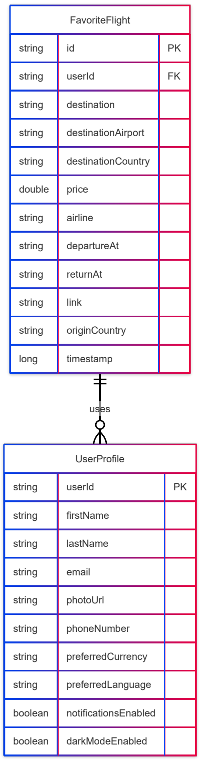

GroqApiService
-OkHttpClient client
-MediaType jsonMediaType
+getActivityRecommendations(destination: String, budget: Double, days: Int) : : String
uses
FavoriteFlight
+String id
+String userId
+String destination
+String destinationAirport
+String destinationCountry
+Double price
+String airline
+String departureAt
+String returnAt
+String link
+String originCountry
+Long timestamp
+generateFlightId() : : String
HomeViewModel
-MutableStateFlow _budgetInput
-MutableStateFlow _startDate
-MutableStateFlow _endDate
-MutableStateFlow _departureCity
-MutableStateFlow _userName
+StateFlow budgetInput
+StateFlow startDate
+StateFlow endDate
+StateFlow departureCity
+StateFlow userName
+updateBudget(budget: String) : : void
+updateStartDate(date: Calendar?) : : void
+updateEndDate(date: Calendar?) : : void
+clearDates() : : void
+updateDepartureCity(city: String) : : void
+fetchUserData(auth: FirebaseAuth, db: FirebaseFirestore) : : void
SharedViewModel
-MutableStateFlow _homeScrollPosition
-MutableStateFlow _flightDetailsScrollPosition
-MutableStateFlow _favoritesScrollPosition
-MutableStateFlow _profileScrollPosition
+StateFlow homeScrollPosition
+StateFlow flightDetailsScrollPosition
+StateFlow favoritesScrollPosition
+StateFlow profileScrollPosition
+updateHomeScrollPosition(position: Int) : : void
+updateFlightDetailsScrollPosition(position: Int) : : void
+updateFavoritesScrollPosition(position: Int) : : void
+updateProfileScrollPosition(position: Int) : : void
AviasalesApiService
-Map countryToAirport
-Map airlineCodeToName
-Map airportInfo
-List preferredAirlines
+searchFlights(originCountry: String, budget: Double, days: Int, departDate: String) : : List<FlightResult>
creates
based on
uses data from
references user
extends
extends
FlightResult
+String destination
+String destinationAirport
+String destinationCountry
+String formattedDestination
+Double price
+String airline
+String departureAt
+String returnAt
+String link
UserProfile
+String userId
+String firstName
+String lastName
+String email
+String photoUrl
+String phoneNumber
+String preferredCurrency
+String preferredLanguage
+Boolean notificationsEnabled
+Boolean darkModeEnabled
+getFullName() : : String
+getDisplayName() : : String
+getAvatarUrl() : : String
ViewModel

Figure 4.2: Class Diagram of the System

4.4 Database Schema / ER Diagram

The database schema diagram shows how data is organized and related within Firebase and/or any local storage system. It outlines entities such as users, trips, saved flights, and preferences.



FavoriteFlight		
string	id	PK
string	userId	FK
string	destination	
string	destinationAirport	
string	destinationCountry	
double	price	
string	airline	
string	departureAt	
string	returnAt	
string	link	
string	originCountry	
long	timestamp	

UserProfile		
string	userId	PK
string	firstName	
string	lastName	
string	email	
string	photoUrl	
string	phoneNumber	
string	preferredCurrency	
string	preferredLanguage	
boolean	notificationsEnabled	
boolean	darkModeEnabled	

Figure 4.3: Database (ER) Diagram

Chapter 5

Security Considerations

5.1 Overview

Security is a core aspect of the *Travee* application, especially since it involves handling user accounts and personal travel details. To ensure safe access and secure data storage, Firebase Authentication is used as the primary mechanism for managing user sign-in and session handling.

5.2 Email and Password Authentication

The application implements secure user authentication via email and password, using Firebase Authentication. Firebase handles the underlying security aspects such as password hashing, salting, and secure storage.

- **Password Policy:** All passwords must be a minimum of 6 characters, enforcing basic password strength to mitigate trivial brute-force attempts.
- **Credential Protection:** User credentials are never stored or processed directly within the application, and all authentication is managed via Firebase's secure infrastructure.

5.3 Token-Based Session Management

Upon successful authentication, Firebase issues a secure ID token for the user session. This token is used to validate the user's identity and authorize access to protected features of the app.

- **Secure Token Storage:** The ID token is stored locally on the user's device using Android's shared preferences or secure storage mechanisms.
- **Auto Sign-In:** On app restart, the token is retrieved and validated to maintain user session without re-login, enhancing usability.
- **Token Expiration:** Firebase tokens are short-lived and automatically refreshed to maintain security and prevent misuse.

5.4 Security Benefits

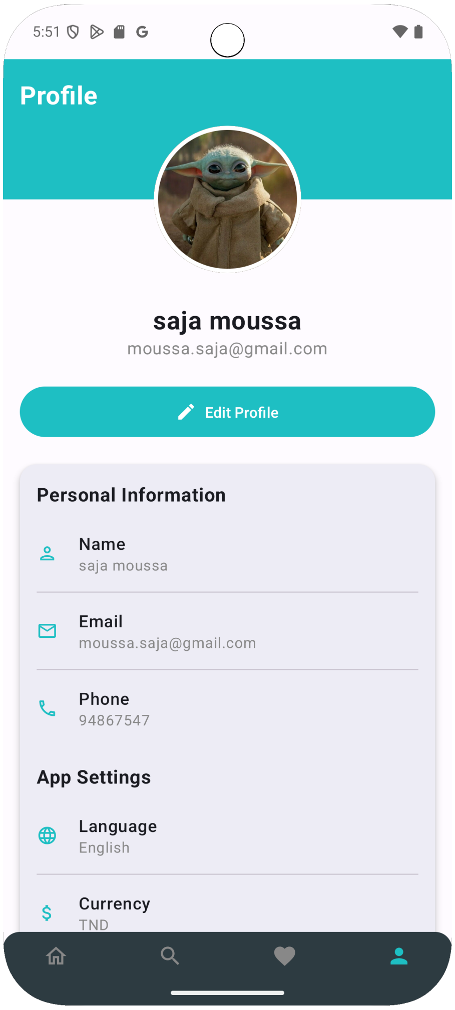
- **Backend as a Service (BaaS) Security:** Firebase provides industry-standard security practices such as SSL encryption, token lifecycle management, and real-time monitoring.
- **Minimal Attack Surface:** By offloading sensitive authentication processes to Firebase, the application reduces the risk of vulnerabilities associated with custom login implementations.
- **Session Integrity:** Token-based validation ensures that only authenticated users can access and modify their trip data.

Chapter 6

UI Design

Figure 6.1: Application Logo

Light and Dark Theme



5:51

Profile

saja moussa

moussa.saja@gmail.com

Edit Profile

Personal Information

Name
saja moussa

Email
moussa.saja@gmail.com

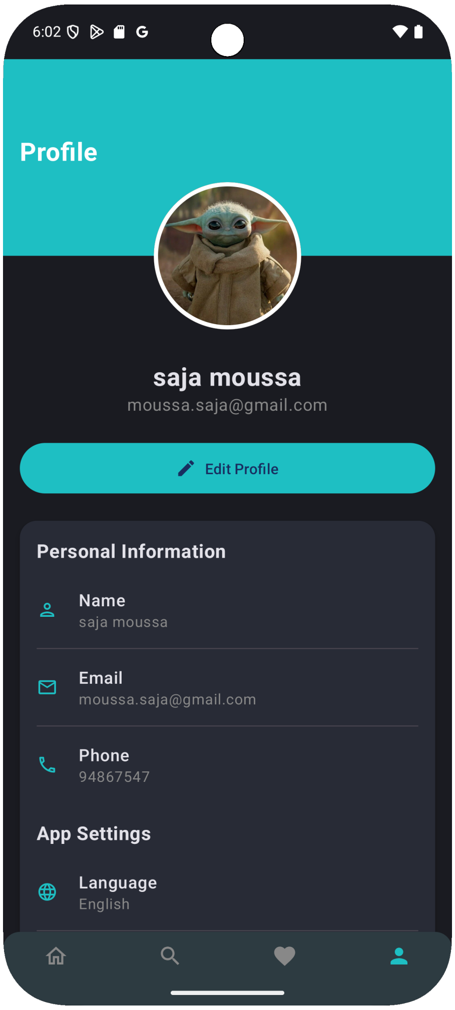
Phone
94867547

App Settings

Language
English

Currency
TND

Figure 6.2: Light Theme



6:02

Profile

saja moussa

moussa.saja@gmail.com

Edit Profile

Personal Information

Name
saja moussa

Email
moussa.saja@gmail.com

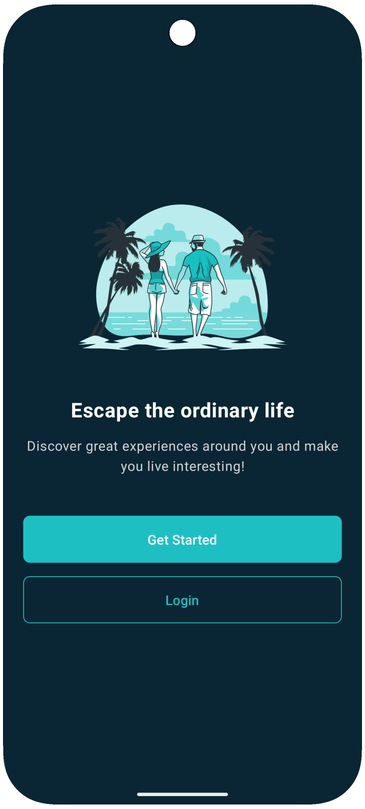
Phone
94867547

App Settings

Language
English

Figure 6.3: Dark Theme

Interface Prototypes



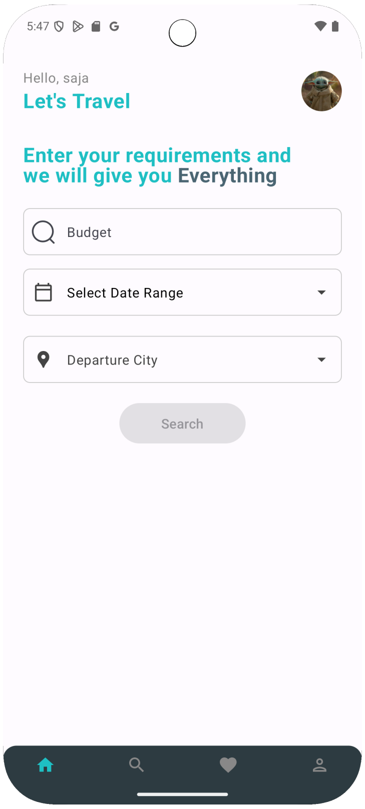
Escape the ordinary life

Discover great experiences around you and make you live interesting!

Get Started

Login

Figure 6.4: Welcome Page



5:47

Hello, saja

Let's Travel

Enter your requirements and we will give you Everything

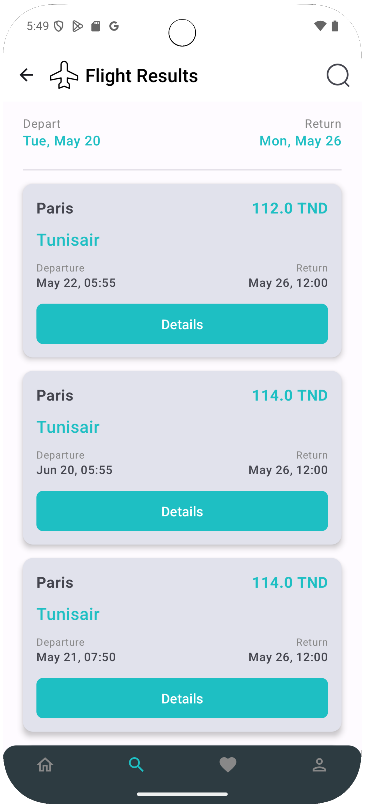
Budget

Select Date Range

Departure City

Search

Figure 6.5: Home Page



5:49

Flight Results

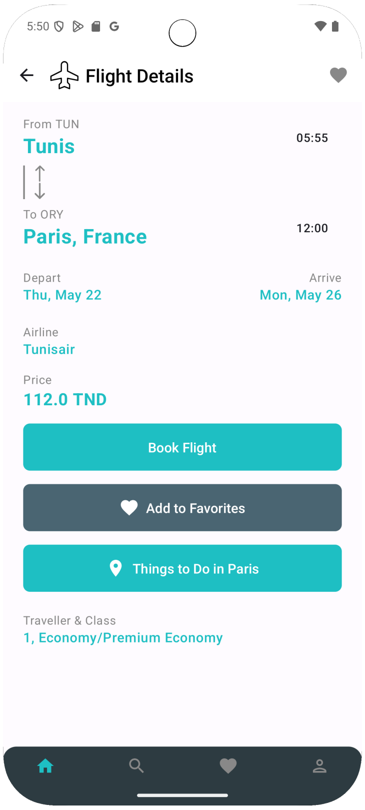
Depart Tue, May 20 — Return Mon, May 26

Paris 112.0 TND
Tunisair
Departure May 22, 05:55 — Return May 26, 12:00
Details

Paris 114.0 TND
Tunisair
Departure Jun 20, 05:55 — Return May 26, 12:00
Details

Paris 114.0 TND
Tunisair
Departure May 21, 07:50 — Return May 26, 12:00
Details

Figure 6.6: Flights Result Page



5:50

Flight Details

From TUN
Tunis 05:55

To ORY
Paris, France 12:00

Depart Thu, May 22 — Arrive Mon, May 26

Airline
Tunisair

Price
112.0 TND

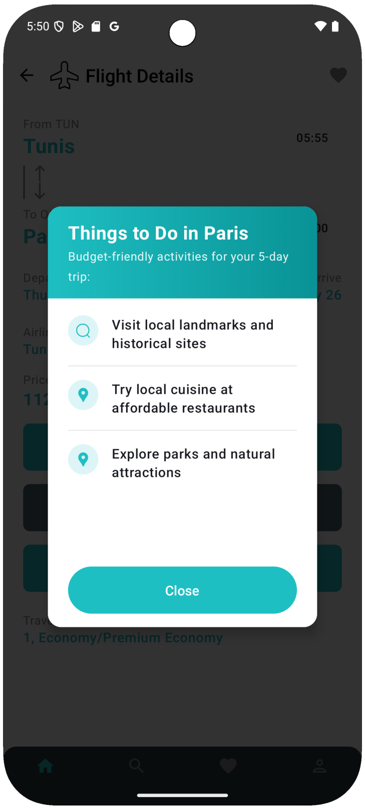
Book Flight

Add to Favorites

Things to Do in Paris

Traveller & Class
1, Economy/Premium Economy

Figure 6.7: Flight Details Page



5:50

Flight Details

Things to Do in Paris

Budget-friendly activities for your 5-day trip:

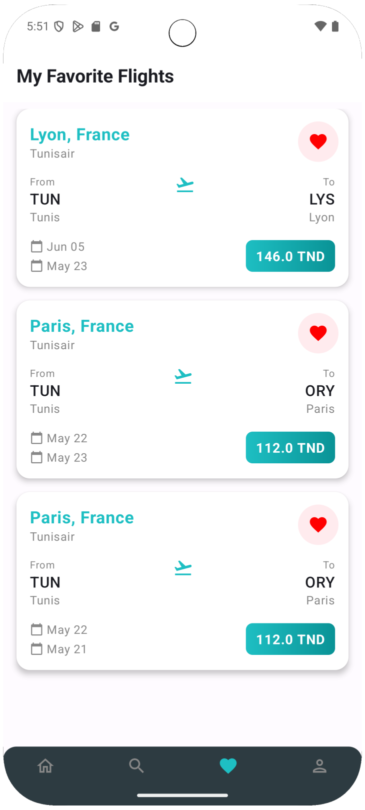
Visit local landmarks and historical sites

Try local cuisine at affordable restaurants

Explore parks and natural attractions

Close

Figure 6.8: Suggested Activities



5:51

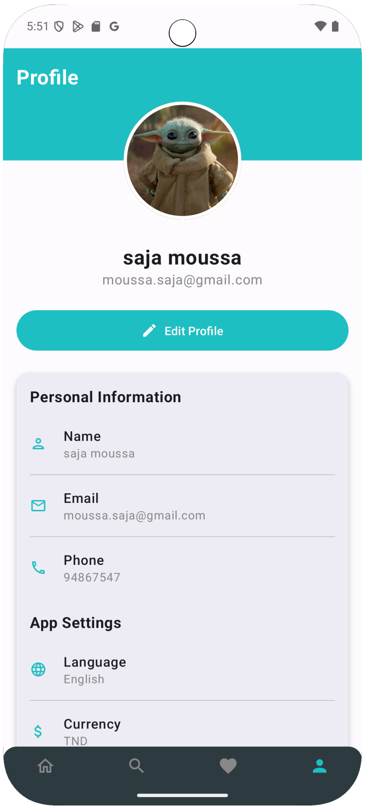
My Favorite Flights

Lyon, France
Tunisair
From TUN Tunis — To LYS Lyon
Jun 05
May 23
146.0 TND

Paris, France
Tunisair
From TUN Tunis — To ORY Paris
May 22
May 23
112.0 TND

Paris, France
Tunisair
From TUN Tunis — To ORY Paris
May 22
May 21
112.0 TND

Figure 6.9: Favorites Page



5:51

Profile

saja moussa

moussa.saja@gmail.com

Edit Profile

Personal Information

Name
saja moussa

Email
moussa.saja@gmail.com

Phone
94867547

App Settings

Language
English

Currency
TND

Figure 6.10: Profile Page

Chapter 7

Conclusion

The development of the *Travee* application aimed to streamline the travel planning process through an intelligent and user-centric mobile solution. By combining real-time flight data from the TravelPayouts API with activity suggestions generated via the Groq API, users are empowered to explore, plan, and organize trips more effectively.

The integration of Kotlin with Jetpack Compose enabled the creation of a responsive and visually appealing user interface, while Firebase offered secure authentication and persistent data storage. Throughout the project, emphasis was placed on modularity, scalability, and user experience.

This project not only provided practical experience in Android development and API integration but also highlighted the importance of delivering value-driven features to end users. In future versions, the application could be extended to include hotel bookings, map navigation, user reviews, and social sharing features, enhancing its utility as a complete travel assistant.

Appendix A

Appendix

A.1 Project Repository

The complete source code for the *Travee* application is available at:

- GitHub Repository: `https://github.com/sajaa45/Travee`