



**Faculty of Computers and Information
Kafr El-Sheikh University**



**"AOUN" TRANSPORT, HOUSING, AND COMMUNITY PROJECT FOR
EXPATRIATE STUDENTS**

By

**Gamal Moemen Hamed Elbatawy
Karim Ayman Fawzy Abdelmageed
Khaled Sabry Mohamed Mostafa
Mohamed Saber Abdulgelel Mohamed
Mohamed Salah Mohamed Kord
Mohamed Sobhy Saber Gafar
Mostafa Mahmoud Ahmed Badr
Omar Aly Abo Elsoud omar**

Supervisor

Assoc. Prof. Reda M. Hussien

Undergraduate Project

Submitted to the Faculty of Computers and Information Kafr El-Sheikh University as a partial
fulfillment for BSc.

**Information Systems Department
June 2025**

Acknowledgments

We are deeply grateful to the many individuals who, both directly and indirectly, contributed to the successful completion of this project. We begin by expressing our heartfelt gratitude to God for His blessings and guidance, which have sustained us throughout this journey.

We extend our sincere thanks to our project guide, Dr. Reda, whose unwavering support, insightful feedback, and encouragement were instrumental in our progress. His guidance during various stages of development enlightened us with valuable perspectives and practical examples, significantly enhancing the quality of our work and ensuring its successful completion.

We are also indebted to our teachers, whose expertise and dedication have left an indelible mark on this project. Their unforgettable contributions have been a constant source of inspiration. Additionally, we appreciate the assistance provided by the non-teaching staff of our faculty, whose readiness to help in every possible way ensured a smooth and efficient workflow.

This project would not have been possible without the collective efforts and support of all these individuals. We are truly grateful for their invaluable contributions.

Abstract

This project presents the design and development of an integrated cross-platform digital ecosystem, comprising a web application and a mobile application, to streamline university students' access to essential services. Addressing the challenge of fragmented student-oriented services, the proposed system unifies functionalities such as transportation, housing, extracurricular activities, and job opportunities into a single, user-friendly platform. The web frontend, built with React.js, delivers a dynamic and responsive interface, while the mobile application, developed using Flutter, ensures seamless performance on both Android and iOS devices. The backend, powered by ASP.NET and SQL Server, provides robust data management and efficient processing through RESTful APIs. Key features include multilingual support, modern UI/UX design aligned with Human-Computer Interaction (HCI) principles, and scalability for future service expansions. Integration with Google Maps API enhances location-based functionalities. The system achieves a cohesive user experience, with evaluation results demonstrating high usability and performance. This work contributes to the advancement of smart, student-centric platforms and sets a foundation for scalable digital solutions in educational contexts. For a detailed overview, see [Chapter 1](#).

Keywords: Cross-Platform Development, Student Services, React.js, Flutter, ASP.NET, RESTful APIs, UI/UX Design, Smart Systems

Contents

Acknowledgments	ii
Abstract	iii
Contents	iii
List of Figures	viii
List of Tables	xi
Nomenclature	xiv
1 Introduction	1
1.1 Project Overview	1
1.2 Organization Profile	2
1.3 Purpose of the project	2
1.4 Problems in Existing Systems	3
1.5 Solution and Contributions	3
1.6 Mobile Application AI Integration and Technical Features	4
2 System Requirements and Analysis	7
2.1 Introduction	7
2.2 Analysis Model	8
2.3 Study of the System	9
2.3.1 Graphical User Interfaces (GUIs)	9
2.3.2 System Modules	10
2.3.3 Entities Involved	11
2.4 Hardware Specifications	11
2.4.1 Hardware Requirements	11
2.4.2 Software Requirements	12
2.5 Software Requirement Specification	12
2.5.1 Introduction to SRS	12
2.5.2 Functional Requirements – Mobile Application	13
2.5.3 Functional Requirements – Web Application	14
2.5.4 Performance Requirements	16

2.6	Proposed System	17
2.6.1	Functional Features	17
2.6.2	Input and Output	17
3	System Design	19
3.1	Introduction	19
3.2	System Architecture	19
3.2.1	Mobile Application Architecture	20
3.2.2	Mobile Application Features	21
3.3	Entity-Relationship Diagram (ERD)	23
3.4	Database Diagram	24
3.5	Data Flow Diagrams (DFDs)	27
3.5.1	DFD Symbols	27
3.5.2	Construction Approach	27
3.5.3	Types of DFDs	27
3.5.4	Context Diagram (Level-0)	27
3.5.5	Level-1 DFD	28
3.5.6	Current Physical DFD	28
3.5.7	Current Logical DFD	29
3.5.8	New Logical DFD	31
3.5.9	New Physical DFD	31
3.5.10	DFD Rules	32
3.6	Unified Modeling Language (UML) Diagrams	32
3.6.1	Architectural Views	32
3.6.2	UML Diagrams	33
3.7	Data Dictionary	42
3.7.1	Core Entities	42
3.7.2	Enumerations	43
3.7.3	Complex Types	43
3.7.4	Key Relationships	44
3.7.5	Business Rules	44
3.8	Security and Scalability	44
4	Used Technologies	47
4.1	Introduction	47
4.2	Frontend Technologies	47
4.2.1	Flutter and Dart	47
4.2.2	React.js	50
4.3	Backend Technologies	52
4.3.1	ASP.NET Core	52
4.3.2	C-Sharp	53
4.4	Database Technologies	53
4.4.1	SQL Server	53
4.5	AI and Machine Learning Technologies	54
4.5.1	Content-Based Recommendation System	54

4.5.2	Sentiment Classification Model	55
4.5.3	Face Verification Model	56
4.6	External APIs and Integrations	57
4.6.1	Google Maps API	57
4.6.2	Payment Gateway	57
4.7	Security Technologies	57
4.7.1	JSON Web Tokens (JWT)	57
4.7.2	Transport Layer Security (TLS)	58
4.8	Development and Design Tools	58
4.9	Testing Tools	59
4.10	Technology Selection Rationale	60
4.11	Summary	62
5	System Implementation	63
5.1	Main Web Application	63
5.1.1	Home Page Interface	63
5.1.2	Authentication System	63
5.1.3	Information and Support Pages	66
5.1.4	Housing Module	66
5.1.5	Job and Activities Module	66
5.1.6	Corporate Partners	66
5.1.7	Service Offerings	66
5.2	Transportation Web Application	73
5.3	Mobile Application	78
5.3.1	Mobile Application Features	78
5.3.2	Mobile User Experience Design	80
5.4	System Integration and Architecture	80
5.4.1	Application Ecosystem	80
5.4.2	Technical Implementation	81
5.4.3	User Benefits	81
5.4.4	Mobile Application Authentication System	81
5.4.5	Driver Flow Management System	97
5.4.6	User Flow Management System	114
5.4.7	Utilities and System Features	132
5.4.8	Other Screens and Legal Information	139
6	System Testing	147
6.1	Introduction	147
6.2	Testing Approach	147
6.3	Test Types	148
6.4	Tested Features	148
6.5	Challenges and Solutions	149
6.6	Conclusion	149

7	System Security	151
7.1	Introduction	151
7.2	Security Technologies	151
7.3	AI-Driven Face Verification Model	152
7.3.1	Purpose and Importance	152
7.3.2	Technologies Used	152
7.3.3	Verification Process	152
7.3.4	Rationale	153
7.4	Webhooks for Secure Event Handling	153
7.5	Integration and Impact	153
7.6	Challenges and Solutions	154
7.7	Conclusion	154
8	Conclusion and Future Work	157
8.1	Conclusion	157
8.2	Benefits	158
8.3	Future Work	158

List of Figures

1.1	System Architecture of the <i>Aoun</i> Platform	5
2.1	Agile Approach	9
2.2	Functional Requirements	15
3.1	Entity-Relationship Diagram for the <i>Aoun</i> Platform	24
3.2	Database Schema Diagram for the <i>Aoun</i> Platform	26
3.3	Context Diagram (Level-0) for the <i>Aoun</i> Platform	27
3.4	Level-1 Data Flow Diagram for the <i>Aoun</i> Platform	28
3.5	Current Physical Data Flow Diagram	29
3.6	Current Logical Data Flow Diagram	30
3.7	New Logical Data Flow Diagram with Enhanced Features	31
3.8	New Physical Data Flow Diagram	32
3.9	Class Diagram for User Hierarchy	33
3.10	Class Diagram for Transportation System	34
3.11	Sequence Diagram for Booking Process	35
3.12	Activity Diagram for Trip Booking Flow	36
3.13	State Diagram for Booking Lifecycle	37
3.14	Use Case Diagram for the Transportation Module	38
3.15	Component Diagram for System Architecture	40
3.16	Deployment Diagram for Physical Architecture	41
4.1	Technology Stack of the <i>Aoun</i> Platform	61
5.1	Main Web Application - Home Page Interface (Upper Section)	64
5.2	Main Web Application - Home Page Interface (Lower Section)	65
5.3	Main Web Application - Authentication System	67
5.4	Main Web Application - Information and Support Pages	68
5.5	Main Web Application - Housing Module Interface	69
5.6	Main Web Application - Housing Module Property Details	70
5.7	Main Web Application - Housing Module Property Posting Process	71
5.8	Main Web Application - Job Module Interface (Part 1)	72
5.9	Main Web Application - Job Module Interface (Part 2)	72
5.10	Main Web Application - Job Module Details Page	73
5.11	Main Web Application - Job Module Posting Process	74
5.12	Main Web Application - Corporate Partners and Strategic Alliances	75

5.13	Main Web Application - Platform Service Offerings and Features	76
5.14	Transportation Web Application - Overview and Planning Interface	77
5.15	Transportation Web Application - Booking System	78
5.16	Mobile Application - Splash Screen	82
5.17	Mobile Application - Introduction and Onboarding Screens (Part 1)	83
5.18	Mobile Application - Introduction and Onboarding Screens (Part 2)	84
5.19	Mobile Application - User Type Selection Interface (Part 1)	85
5.20	Mobile Application - User Type Selection Interface (Part 2)	86
5.21	Mobile Application - User Registration and Email Confirmation	87
5.22	Mobile Application - Email Verification and Login	88
5.23	Mobile Application - Error Handling and Password Reset	89
5.24	Mobile Application - Verification Code Process	90
5.25	Mobile Application - New Password Creation	91
5.26	Mobile Application - Driver Personal Information (Part 1 & 2)	92
5.27	Mobile Application - Driver Personal Information (Part 3)	93
5.28	Mobile Application - Driver Vehicle Information Registration	94
5.29	Mobile Application - Driver Registration Instructions	95
5.30	Mobile Application - Driver Confirmation and Login Interface	96
5.31	Driver login error handling	97
5.32	Driver Flow - Account Settings and Language Options	98
5.33	Driver Flow - Theme Customization	99
5.34	Driver Flow - Trip Creation Process (Part 1 & 2)	100
5.35	Driver Flow - Location Selection and Search	101
5.36	Driver Flow - Data Input Example	102
5.37	Driver Flow - Trip Detail Management	103
5.38	Driver Flow - Trip Booking and Completion Status	104
5.39	Driver Flow - Help and Feedback System	105
5.40	Driver Flow - Trip History Management	106
5.41	Driver Flow - Home Dashboard with Created Trips (Part 1 & 2)	107
5.42	Driver Flow - Home Dashboard Extended View and Navigation	108
5.43	Driver Flow - Empty Home Dashboard State	109
5.44	Driver Flow - Profile Overview and Vehicle Information	110
5.45	Driver Flow - Extended Vehicle Data and Profile Editing	111
5.46	Driver Flow - Safety and Security System	112
5.47	Driver Flow - Wallet Overview and Transactions	113
5.48	Driver Flow - Withdrawal Methods and Processing	114
5.49	User Flow - Card Addition Process (Part 1 & 2)	115
5.50	User Flow - Card Addition Process (Part 3 & 4)	116
5.51	User Flow - Booking History Management	117
5.52	User Flow - Home Dashboard States	118
5.53	User Flow - Booking and Meeting Point Selection	119
5.54	User Flow - Wallet Payment and Web Navigation	120
5.55	User Flow - Wallet Completion and Card Payment	121
5.56	User Flow - Stripe Payment and Receipt	122
5.57	User Flow - Profile Overview and Extended Information	123

5.58 User Flow - Theme and Language Customization	124
5.59 User Flow - Profile Editing	125
5.60 User Flow - Empty Search States	126
5.61 User Flow - Trip Recommendations and Location Search	127
5.62 User Flow - Similar Distance Travel Analysis	128
5.63 User Flow - Transaction Management	129
5.64 User Flow - Trip Details and Route Visualization	130
5.65 User Flow - Driver Information Display	131
5.66 User Flow - Additional Information and Meeting Point Details	132
5.67 Localization Features	133
5.68 Extended Localization Options	134
5.69 Theme Management Options	135
5.70 Advanced Theme Customization	136
5.71 Shimmer Loading Effects	137
5.72 Input Validation Examples	138
5.73 Advanced Validation Features	139
5.74 Mobile Application - Privacy Policy Documentation	140
5.75 Mobile Application - Terms and Conditions Documentation	141
5.76 Mobile Application - Customer Support and Contact	142
7.1 Security Architecture of the <i>Aoun</i> Platform	155

List of Tables

3.1	Summary of the Transportation Module's mobile app features, screens, and technologies.	23
3.2	Core entities for the Transportation Module, their descriptions, attributes, and relationships.	42
3.3	Enumerations and their usage in the Transportation Module.	43
3.4	Complex types and their descriptions for the Transportation Module.	43
3.5	Key relationships and their behaviors in the Transportation Module.	44
3.6	Business rules enforced by the Transportation Module.	44
3.7	Summary of Aoun's core modules, their functionalities, and technology stack.	45
4.1	Flutter Dependencies for Transportation Module	51
4.2	Flutter Testing Scope	60

Chapter 1

Introduction

The rapid growth of higher education in Egypt has led to an increasing number of students relocating from their home governorates to university cities, facing significant challenges in securing reliable transportation, affordable housing, and supportive community networks. The *Aoun* project, developed as a graduation project at Kafr Elsheikh University, Faculty of Computers and Information, under the supervision of Dr. Reda Mabrouk, introduces a transformative digital ecosystem to address these challenges. The platform comprises a mobile application, dedicated to transportation services (built with Flutter and Dart), and a web application, focused on housing and community services (developed using React.js and Bootstrap). Supported by a robust ASP.NET Core backend and SQL Server database, *Aoun* delivers a unified, user-centric platform that enhances the daily lives of students living away from home through seamless access to transportation, housing, and community engagement services.

1.1 Project Overview

The *Aoun* platform is designed to meet the distinct needs of Egyptian university students residing away from their hometowns. By consolidating three core services, it overcomes the fragmentation of existing student-oriented solutions:

- **Transportation:** The mobile application, developed using Flutter and Dart, is dedicated exclusively to transportation services, enabling seamless travel between and within all Egyptian governorates. It offers flexible scheduling, cost-effective options aligned with

students' academic calendars, and a trip-sharing feature to coordinate travel with friends.

The app supports Arabic and English, is available on Android and iOS, and includes limited offline capabilities for storing session data and user preferences.

- **Housing:** Available through the web application, this service offers a curated selection of verified, safe, and affordable accommodations in university cities, prioritizing proximity and student suitability.
- **Community Platform:** Also accessible via the web application, this platform fosters a vibrant network where students can share events, job opportunities, and resources, promoting collaboration and personal growth.

Leveraging cutting-edge technologies, including Flutter, Dart, React.js, ASP.NET Core, RESTful APIs, NLP, CNN, and Google Maps API, with UI/UX crafted in Figma, *Aoun* ensures a cohesive, scalable, and responsive user experience. The mobile app follows Clean Architecture and the MVVM pattern, combined with the BLoC pattern and Provider for state management, ensuring performance and scalability even in low-network conditions. Collaborative development was streamlined using GitHub for version control.

1.2 Organization Profile

The *Aoun* project was realized by a multidisciplinary team of students from Kafr Elsheikh University, Faculty of Computers and Information, under the expert guidance of Dr. Reda Mabrouk. The team's proficiency in mobile and web development, artificial intelligence, UI/UX design, system integration, and agile project management enabled the creation of a robust, student-focused platform. By employing reusable frameworks and a streamlined development methodology, the team delivered a high-quality solution with efficiency and innovation.

1.3 Purpose of the project

The purpose of the *Aoun* project is to develop a mobile application and website that provides essential services to students living away from home in Egypt. These services include:

- **Transportation:** The mobile application facilitates travel between and within all Egyptian governorates, offering booking and trip-sharing features to coordinate with friends, with schedules tailored to students' academic needs.
- **Housing:** The web application offers safe and affordable housing options in university cities.
- **Community Platform:** The web application creates a space for students to connect, share events, job opportunities, and resources.

By integrating these services into a single platform, with the mobile app focusing on transportation and the web app handling housing and community services, *Aoun* aims to simplify the lives of students, reduce their stress, and foster a supportive community. The project seeks to fill a gap in the current market, as no similar application exists that specifically caters to the needs of students living away from home.

1.4 Problems in Existing Systems

The absence of a dedicated platform for students living away from home creates significant obstacles:

- **Transportation:** Limited availability of reliable and affordable travel options tailored to students' schedules and budgets.
- **Housing:** Challenges in securing safe, cost-effective, and conveniently located accommodations near universities.
- **Community Support:** Lack of a centralized hub for students to connect, share resources, or access opportunities, leading to social isolation.

1.5 Solution and Contributions

Aoun addresses these challenges through its integrated services:

- **Transportation:** The mobile application provides a sophisticated booking system optimized for inter- and intra-governorate travel across all Egyptian governorates. It includes a trip-sharing feature for coordinating with friends, leverages Google Maps API for interactive map display, location search and selection, trip route visualization, and real-time trip tracking, and ensures user authentication and data privacy for secure access. Built with Flutter and Dart using Clean Architecture, MVVM, and the BLoC pattern with Provider, the app supports Arabic and English, operates on Android and iOS, and offers limited offline capabilities for enhanced user experience.
- **Housing:** The web application provides a secure database of verified accommodations, ensuring safety, affordability, and proximity to universities.
- **Community Platform:** The web application offers a dynamic environment for students to collaborate, share opportunities, and build supportive networks.

With a modern UI/UX designed in Figma and a scalable backend powered by ASP.NET Core and SQL Server, *Aoun* fills a critical gap in the Egyptian market, setting a new benchmark for student-centric digital platforms.

1.6 Mobile Application AI Integration and Technical Features

The *Aoun* mobile application, dedicated to transportation services, integrates advanced AI components and robust technical features to enhance functionality and user experience:

- **AI Integration:**
 - ▶ **Sentiment Analysis:** Utilizes NLP to analyze driver reviews, providing insights into driver reliability and user satisfaction.
 - ▶ **Driver Recommendations:** Employs CNN models to recommend drivers based on user preferences and historical data.
 - ▶ **Driver ID Verification:** Implements facial matching logic to ensure driver authenticity, enhancing user safety.

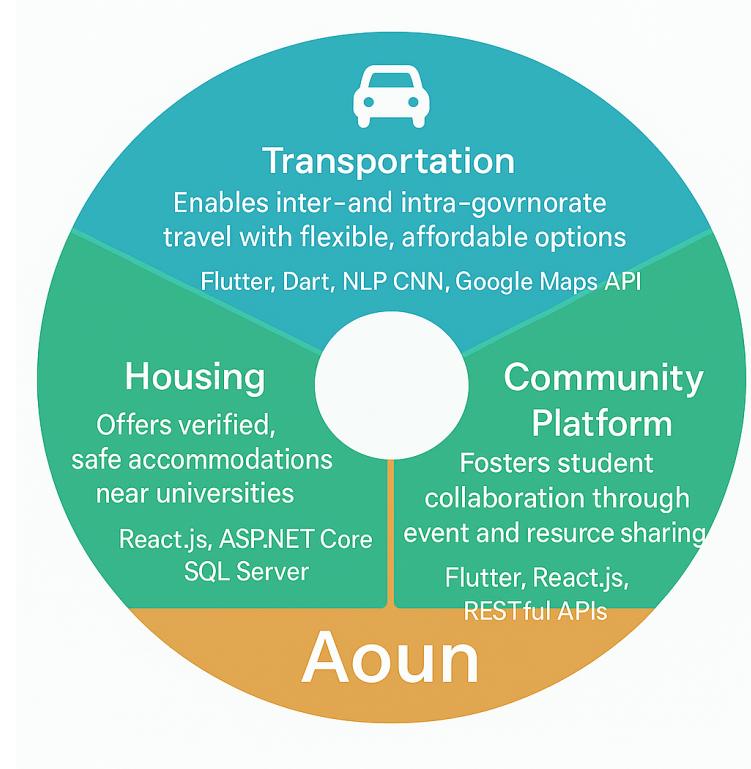


Figure 1.1 – System Architecture of the *Aoun* Platform

- **Technical Features:**

- ▶ Developed using Flutter and Dart, following Clean Architecture and MVVM, with the BLoC pattern and Provider for state management.
- ▶ Supports Arabic and English languages, ensuring accessibility for a diverse user base.
- ▶ Available on Android and iOS platforms.
- ▶ Includes limited offline capabilities, such as storing session data and saved user preferences, for reliable performance in low-network conditions.
- ▶ Integrates Google Maps API for interactive map display, location search and selection, trip route visualization, and real-time trip tracking.
- ▶ Implements user authentication and data privacy measures to ensure secure access and protection of personal data.

Chapter 2

System Requirements and Analysis

2.1 Introduction

The *Aoun* system addresses critical challenges faced by Egyptian university students who relocate from their home governorates to pursue higher education. These challenges include securing cost-effective and reliable transportation, accessing safe and affordable housing, and fostering a robust community network. Developed as a graduation project at Kafr Elsheikh University, Faculty of Computers and Information, under the supervision of Dr. Reda Mabrouk, *Aoun* comprises a mobile application (crafted with Flutter and Dart) dedicated to transportation services and a web application (developed using React.js and Bootstrap) for housing and community services. The mobile application focuses solely on the Transportation module, whereas the Housing and Community modules are accessible exclusively via the web app. Supported by a high-performance ASP.NET Core backend and SQL Server database, *Aoun* integrates three core services: seamless transportation coordination across and within governorates, verified housing tailored to student needs, and a dynamic community hub for sharing events, job opportunities, and resources.

To define a comprehensive set of system requirements, the team employed a rigorous stakeholder-driven approach, conducting focus group discussions with students, surveys, and consultations with university administrators over several months. These efforts illuminated critical pain points and user expectations, ensuring that *Aoun* delivers innovative, user-centric solutions. This chapter

delineates the requirements and analysis that underpin the design and development of a scalable, efficient, and impactful system to empower students academically and socially.

2.2 Analysis Model

To systematically capture *Aoun*'s functional and non-functional requirements, the team adopted the Agile development methodology, renowned for its iterative approach, stakeholder collaboration, and adaptability to evolving needs. This methodology facilitated rapid prototyping, continuous integration, and iterative refinement based on user feedback. The Agile process for *Aoun* encompasses the following phases:

- **Concept:** Articulating and prioritizing project objectives to align with student-centric goals.
- **Inception:** Assembling the development team, securing resources, and defining initial requirements and technical environments.
- **Iteration/Construction:** Delivering incremental, functional software builds driven by prioritized requirements and stakeholder feedback.
- **Release:** Executing Quality Assurance (QA) testing, producing user documentation, and deploying iterations to production.
- **Production:** Providing ongoing maintenance and support to ensure system reliability.
- **Retirement:** Managing end-of-life processes, including user notifications and data migration. [?]

Each iteration follows a disciplined workflow:

- **Requirements:** Defining iteration objectives using product and sprint backlogs, enriched by student and stakeholder feedback.
- **Development:** Designing and implementing features aligned with specified requirements.
- **Testing:** Conducting comprehensive QA testing and developing user guides.

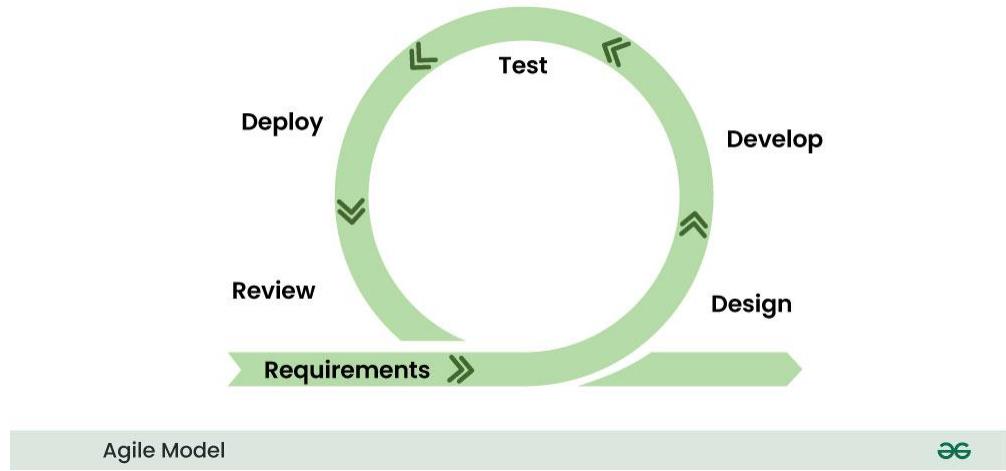


Figure 2.1 – Agile Approach

[?]

- **Delivery:** Integrating and deploying functional increments to production.
- **Feedback:** Incorporating user and stakeholder insights to refine subsequent iterations.

2.3 Study of the System

2.3.1 Graphical User Interfaces (GUIs)

The *Aoun* mobile application, dedicated to the Transportation module, prioritizes an intuitive and accessible user experience through meticulously designed, responsive GUIs developed using Flutter and Dart for seamless performance across Android and iOS devices. The web application, built with React.js, Bootstrap, and ASP.NET Core Web API, handles the Housing and Community modules and is optimized for major browsers, including Chrome, Firefox, and Safari. The mobile app's GUI features, informed by real feedback from university students, include:

- **Login and Registration:** A secure authentication system with independent login and registration flows, supporting social media integration for streamlined user onboarding.
- **Transportation Booking:** An interactive interface enabling real-time trip booking and live driver tracking using Google Maps APIs. Students can browse rides, receive five personalized recommendations via a content-based recommendation algorithm, select routes, share

trip details with friends, and provide driver reviews. The interface supports viewing stored user profile data offline (read-only) and integrates diverse payment gateways for flexibility.

- **Future Enhancements:** Notifications for trip status updates, such as booking confirmations and driver arrivals, are planned for future releases to enhance user engagement.

The web application's GUIs include:

- **Housing Presentation:** A robust platform for hosts to create detailed accommodation listings, including images, descriptions, and amenities. Students can view options based on preferences and contact hosts directly for inquiries or viewings.
- **Community Forum:** A vibrant, interactive space for students to post text, images, or links, engage in discussions, and receive real-time notifications, fostering collaboration and knowledge sharing.

All GUIs adhere to Human-Computer Interaction (HCI) principles, refined through iterative usability testing and student feedback to ensure simplicity, responsiveness, and cross-device compatibility.

What Is GUI: A graphical user interface (GUI) is a digital interface in which a user interacts with graphical components such as icons, buttons, and menus. In a GUI, the visuals displayed in the user interface convey information relevant to the user, as well as actions that they can take.

[?]

2.3.2 System Modules

The modules described in this section pertain exclusively to the mobile application, which implements the Transportation module. Only the Transportation module is implemented on the mobile app, while other modules such as Housing and Community are part of the web platform. The Transportation module is designed to be modular and scalable, addressing the following core functionality:

- **Transportation Module:** Orchestrates ride booking, route selection, trip scheduling, and payment processing, with real-time integration with transportation providers via RESTful

APIs. It supports real-time trip tracking and trip-sharing features, ensuring seamless coordination across all Egyptian governorates.

This modular architecture ensures flexibility, enabling seamless integration of future enhancements, such as advanced analytics or additional services.

2.3.3 Entities Involved

The *Aoun* ecosystem comprises key entities that interact to deliver a cohesive user experience:

- **Students (Customers):** Primary users who create profiles, access transportation services via the mobile app, and engage with housing and community services through the web app.
- **Transportation Providers:** Entities, such as bus companies or individual drivers, who list services, set pricing, and manage bookings through the mobile app.
- **Housing Hosts:** Individuals or organizations offering accommodations, responsible for creating and updating listings and communicating with students via the web app.
- **Administrators:** Responsible for overseeing the system's operation, including user management, content moderation, and resolving disputes across both the mobile and web applications.

2.4 Hardware Specifications

2.4.1 Hardware Requirements

To ensure optimal performance, the *Aoun* system requires the following hardware specifications:

Server Hardware:

- **Processor:** Intel mid-range or higher processor, with at least 4 cores
- **RAM:** 12GB DDR4
- **Storage:** 500GB SSD for fast data access

- **Network:** Gigabit Ethernet connection for high-speed data transfer

User Devices:

- **Mobile App:** Smartphones running Android 8.0 or later, or iOS 12.0 or later, with at least 2GB RAM
- **Web App:** Any device with a modern web browser, such as Google Chrome, Mozilla Firefox, or Apple Safari

2.4.2 Software Requirements

The software stack for the *Aoun* system includes:

Development Tools:

- **Integrated Development Environments (IDEs)**
- **Database Management:** MySQL
- **Version Control:** Git, hosted on GitHub

Operational Software:

- **Server Operating System**
- **Web Server:** Apache HTTP Server
- **Database Server:** MySQL
- **Backend Framework:** ASP.NET Core Web API
- **Frontend Framework:** Flutter for mobile, React for web

Third-Party Services:

- **Google Maps API** for geolocation and mapping
- **Paymob** for payment processing
- **Firebase** for real-time database and authentication
- **Recommendation system** for filtering

2.5 Software Requirement Specification

2.5.1 Introduction to SRS

The Software Requirement Specification (SRS) is a comprehensive document that outlines the functional and non-functional requirements of the *Aoun* system. It serves as a contract between the development team and stakeholders, ensuring all parties have a clear understanding of what the system will deliver. The SRS is crucial for guiding the design, development, and testing phases of the project.

Functional Requirements: These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. [?]

2.5.2 Functional Requirements – Mobile Application

The functional requirements listed below pertain exclusively to the mobile application, which implements the Transportation module. These requirements ensure that the mobile app meets the specific needs of students for transportation services:

- **User Management:**

- ▶ Users can register with email and password or via social media accounts through an independent login and registration flow distinct from the web app.
- ▶ Profile management allows users to update personal information and preferences, with trip-related data stored locally for offline access (read-only).
- ▶ Role-based access control ensures appropriate permissions for students, providers, and admins.

- **Transportation Module:**

- ▶ Students can view available trips with a filtering system based on their data.
- ▶ The system displays available rides with details such as price, duration, and provider ratings.

- ▶ The driver can make a trip by entering the details and setting the price.
- ▶ Users can book rides in real-time, share trip details with friends, and choose from multiple payment methods.
- ▶ Drivers receive real-time route guidance via Google Maps API, and students can track drivers live.
- ▶ Users must grant location permissions; the app handles GPS accuracy and background usage efficiently.
- ▶ Students can rate and review drivers post-trip to ensure quality and trust.

2.5.3 Functional Requirements – Web Application

The functional requirements for the web application cover the Housing and Community modules:

- **Housing Module:**

- ▶ Hosts can create and manage housing listings, including descriptions, photos, and pricing.
- ▶ Students can see if there is suitable housing for them through the available data like location, price range, and amenities.

- **Community Module:**

- ▶ Users can create posts with text, images, or links, categorized by topics like events, jobs, or resources.
- ▶ Other users can comment on posts, like them, or share them within the platform.

- **Payment Processing:**

- ▶ Secure payment gateway integration for handling transactions.
- ▶ Support for multiple currencies and payment methods.

- **Notifications:**

- ▶ Email notifications for important updates.

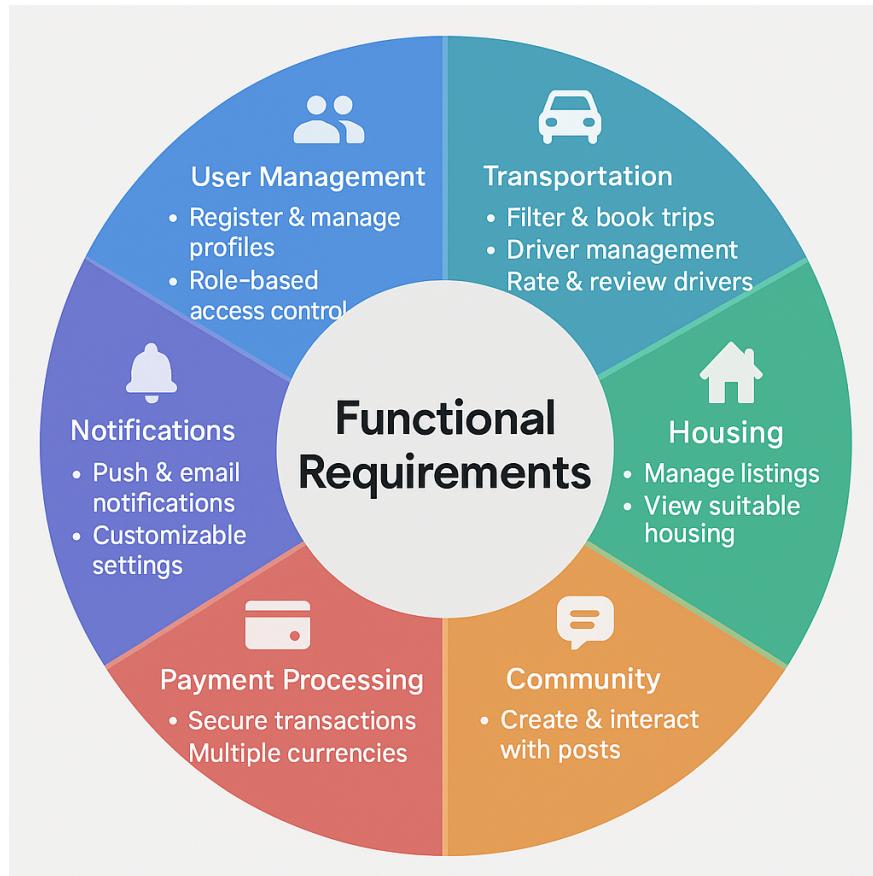


Figure 2.2 – Functional Requirements

- ▶ Customizable notification settings based on user preferences.

2.5.4 Performance Requirements

These are the quality constraints that the system must satisfy according to the project contract.

The priority or extent to which these factors are implemented varies from one project to another.

They are also called non-behavioral requirements. They deal with issues like:

- Performance
- Security.
- Reliability.

To ensure a smooth user experience, the *Aoun* system must meet the following performance criteria:

- **Concurrent Users:** The system should support at least 1000 simultaneous users without significant slowdowns.
- **Response Time:** Critical operations, such as search queries and booking requests, should complete within 2 seconds.
- **Load Time:** The mobile app should launch within 3 seconds, and web pages should load within 5 seconds on a standard broadband connection.
- **Uptime:** The system should achieve 99.9% availability, with scheduled maintenance windows communicated in advance.
- **Scalability:** The architecture should allow for horizontal scaling to accommodate user growth and additional features.
- **Data Integrity:** Transactions and data updates must be atomic and consistent, with regular backups to prevent data loss.

2.6 Proposed System

2.6.1 Functional Features

The *Aoun* system offers a range of functional features designed to meet the needs of students living away from home:

- **Secure Authentication:** Utilizes OAuth 2.0 for secure login via social media or email, with independent flows for the mobile and web apps.
- **Interactive Maps:** Integrated with Google Maps in the mobile app, providing real-time route planning, live driver tracking, and traffic updates for transportation services.
Google Maps API is a set of programming tools that lets developers embed Google Maps directly into websites and mobile apps. Think of it as Google giving businesses the keys to

their mapping technology—allowing you to create custom map experiences without building everything from scratch. [?]

- **Advanced Search Filters:** Allows users to find their transportation and housing with multiple criteria.
- **Real-Time Updates:** The community forum on the web app updates in real-time, ensuring users see the latest posts.
- **Payment Integration:** Seamless payment processing with support for local and international payment methods.
- **Notification System:** Keeps users informed of booking statuses, new community interactions, and system updates, with trip status notifications planned for future mobile app releases.

2.6.2 Input and Output

The *Aoun* system processes various inputs and generates corresponding outputs:

Input Data	Output Data
User registration details (name, email, password, picture, etc.)	Booking confirmations and receipts
Transportation booking requests (origin, destination, dates, etc.)	Housing listing results with detailed information
Housing criteria (location, price, amenities, etc.)	Community feed with posts and interactions
Community posts (text, images, links, etc.)	Notification alerts for updates and messages
Payment information for transactions	Reports and analytics for administrators

Chapter 3

System Design

3.1 Introduction

The *Aoun* platform, developed as a graduation project at Kafr Elsheikh University, Faculty of Computers and Information, under the supervision of Assoc. Prof. Reda M. Hussien, is a multi-platform solution addressing the needs of Egyptian university students in transportation, housing, and community engagement. The Transportation Module, a mobile application built with Flutter and Dart, facilitates ride management, driver-student assignments, and real-time location tracking across Android and iOS devices. The Housing and Activities Modules, implemented as web applications using React.js and Bootstrap, enable accommodation browsing and event participation. Powered by a service-oriented architecture (SOA) with an ASP.NET Core backend, RESTful APIs, and SQL Server database, the platform ensures modularity, scalability, and security via JSON Web Tokens (JWT) and role-based authorization. This chapter details the structural, logical, and behavioral design of *Aoun*, focusing on the mobile application's Transportation Module, using Entity-Relationship (E-R) diagrams, Data Flow Diagrams (DFDs), Unified Modeling Language (UML) diagrams, and a comprehensive data dictionary, building on the requirements in [Chapter 2](#).

3.2 System Architecture

The *Aoun* platform employs a service-oriented architecture with a layered design:

- **Presentation Layer:** The Transportation Module is implemented as a Flutter mobile app supporting Android and iOS, designed in Figma with Arabic/English support, RTL layout, and Dark/Light themes. The Housing and Activities Modules are delivered via a React.js/Bootstrap web app.
- **Application Layer:** ASP.NET Core backend with RESTful APIs, integrating Google Maps API for location services in the mobile app.
- **Data Layer:** SQL Server for persistent storage, handling trip, booking, and user data.

The Transportation Module operates as a loosely coupled service, communicating with the backend via RESTful APIs, ensuring interoperability and scalability. The mobile app leverages Flutter's cross-platform capabilities, the 'google_maps_flutter' package for mapping, and local storage for offline functionality.

3.2.1 Mobile Application Architecture

The Transportation Module's mobile application, built with Flutter and Dart, follows a layered architecture aligned with the SOA:

- **UI Layer:** Comprises student and driver screens designed in Figma, supporting Arabic/English localization via 'flutter-localizations' with '.arb' files, RTL layout, and Dark/Light themes for accessibility.
- **Business Logic Layer:** Implements the BLoC pattern with the 'provider' package for state management, ensuring reactive and maintainable code. Handles user interactions, API calls, and local data management.
- **Data Layer:** Uses 'hive' for offline storage of trip details, booking information, and user preferences, with synchronization via RESTful APIs when connectivity is restored.
- **Integration Layer:** Connects to the ASP.NET Core backend via RESTful APIs and integrates Google Maps API for real-time tracking and route visualization.

The app employs the 'Navigator 2.0' API for declarative navigation, ensuring smooth transitions between screens and robust state management.

3.2.2 Mobile Application Features

The Transportation Module mobile app provides a comprehensive set of features tailored to Egyptian university students and drivers:

- **Purpose and Functionality:** Enables students to manage rides, book trips, and track drivers in real-time across Android and iOS, covering all Egyptian governorates. Drivers can create and manage trips, view booked student locations, and handle payments, addressing the transportation needs of students relocating to university cities.
- **UI/UX Design:** Designed in Figma, the app includes distinct screens for students (Home, Search, Trip Details, Booking, Payment, Previous Transactions, Previous Trips, Settings, Profile) and drivers (Home/Dashboard, Create Trip, Trip Details, Route Map, Users Map, Transactions, Previous Trips, Profile). The UI supports Arabic/English, RTL, and Dark-/Light themes, refined through iterative student feedback to adhere to Human-Computer Interaction (HCI) principles.
- **Navigation Flow:** Navigation is managed via ‘Navigator 2.0’. For students: the Home Page displays available trips; selecting a trip navigates to Trip Details; clicking ‘Book’ leads to the Booking Page, followed by the Payment Page for confirmation. If no trips are available, the Search Page allows creating a Trip Request. For drivers: the Home/Dashboard leads to Create Trip, then Route Map for destination selection, and Users Map to view booked students’ locations.
- **Google Maps Integration:** Utilizes the ‘google maps flutter’ package to integrate Google Maps API, enabling:
 - ▶ **Students:** View trip routes, select meeting points, and track drivers in real-time.
 - ▶ **Drivers:** Set trip routes with search and place suggestions, view booked students’ meeting points on an interactive map.

Real-time location updates are handled via the ‘geolocator’ package, with updates every 10 seconds or 100 meters, supporting background location tracking when the app is minimized.

- **Offline Mode:** Implemented using ‘hive’ for local storage, the app caches trip details, booking information, and user preferences (read-only). Data is queued locally and synchronized with the backend via RESTful APIs when connectivity is restored, using a last-write-wins conflict resolution strategy to ensure data consistency.
- **Authentication:** Supports social media login via ‘firebase-auth’ with Google and Facebook Sign-In, alongside email/password authentication. JWT tokens are stored securely in ‘hive’ for session management.
- **Error Handling:** Uses Snackbars for transient errors (e.g., invalid input) and Dialogs for critical errors (e.g., payment failure), providing clear user feedback.
- **Testing:** Employs ‘flutter-test’ for unit tests (covering 80%+ of the codebase).
- **Future Enhancements:** A notification system is planned using Firebase Cloud Messaging (FCM) to deliver booking confirmations, trip updates, and driver arrival alerts.

Feature	Description	Technologies
UI/UX	Student and driver screens with Arabic/English support, RTL layout, and Dark/Light themes	Flutter, Figma, flutter_localizations
Real-Time Tracking	Displays trip routes, meeting points, and live driver locations	google-maps-flutter, geolocator, Google Maps API
Offline Mode	Stores trip data, bookings, and user preferences locally for offline access	Hive database
Authentication	Social media and email/password login system	Firebase Auth, JWT tokens
Error Handling	User-friendly notifications via Snackbars and Dialog boxes	Flutter Widgets
Testing	Comprehensive unit, widget, and integration test coverage	flutter_test, mockito, Firebase Test Lab
Push Notifications	Planned booking confirmations and real-time trip update alerts	Firebase Cloud Messaging

3.3 Entity-Relationship Diagram (ERD)

The database schema for the Transportation Module is normalized, with core entities:

- **User:** Base entity with Table-Per-Hierarchy (TPH) inheritance (attributes: Id, Age, Gender, UserType, IsBanned).
- **Student:** Inherits from User, creates bookings and trip requests.
- **Driver:** Includes LicenseNumber, ApprovalStatus, Balance.

- **Trip:** DepartureTime, AvailableSeats, PricePerSeat.
- **Booking:** BookingDate, TotalPrice, NumberOfSeats.
- **Payment:** Amount, PaymentMethod, Status.

Relationships include:

- **One-to-Many:** Driver to Trips, Student to Bookings.
- **Many-to-One:** Booking to Trip, Payment to Booking.

Foreign keys ensure referential integrity, with indexes on fields like DepartureTime and PricePerSeat.

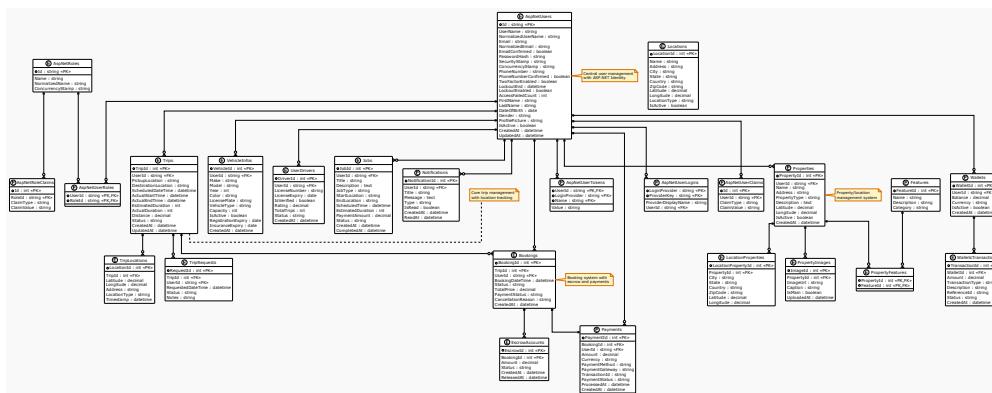


Figure 3.1 – Entity-Relationship Diagram for the Aoun Platform

3.4 Database Diagram

The schema extends the E-R model with:

- **TripLocation:** Latitude, Longitude.
- **Wallet:** UserId, Balance.
- **EscrowAccount:** BookingId, Amount.
- **VehicleInfo:** VehicleId, Model, LicensePlate.

Constraints enforce business rules (e.g., AvailableSeats ≥ 0 , BookingDate at least one hour before DepartureTime).

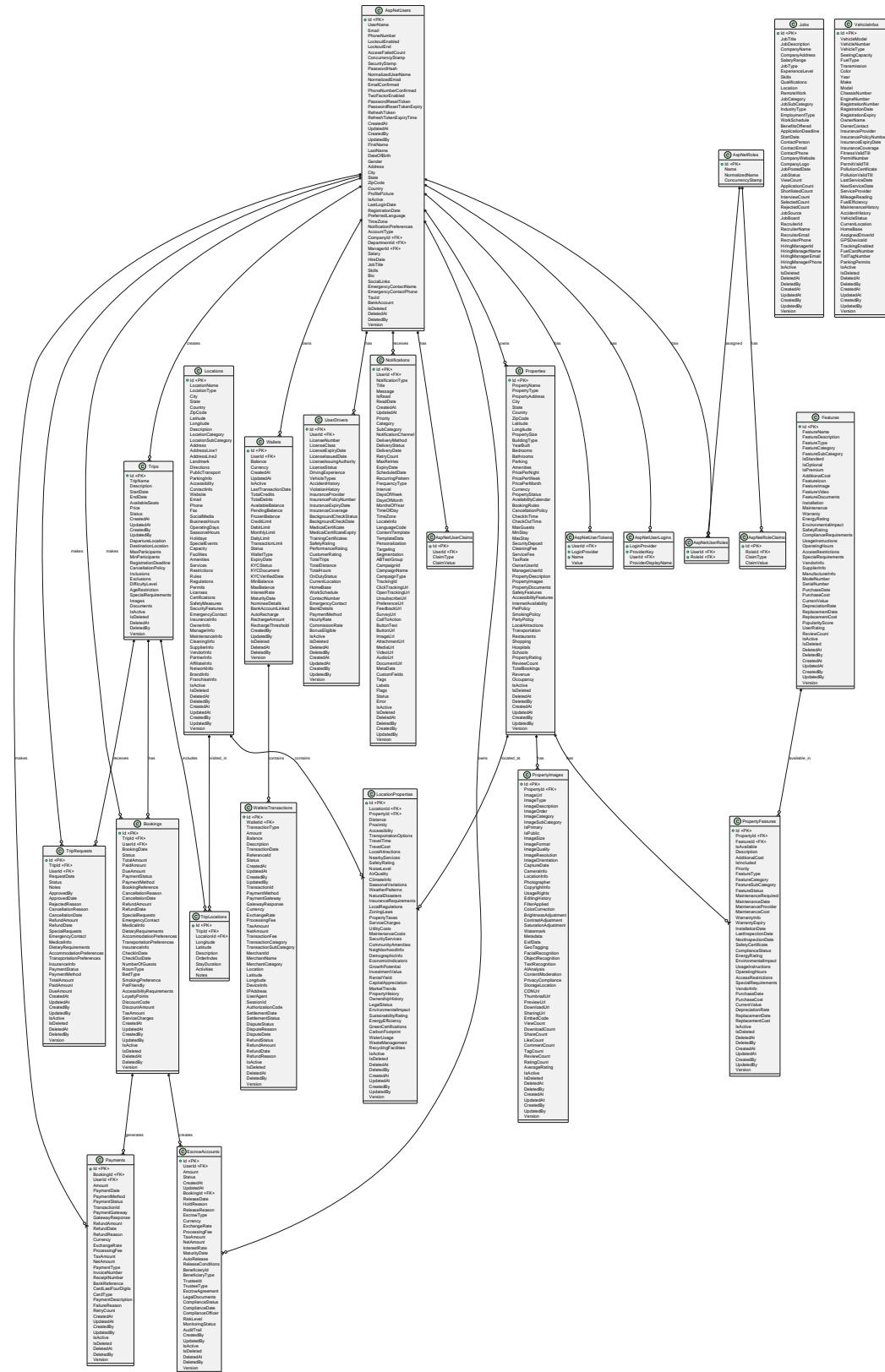


Figure 3.2 – Database Schema Diagram for the *Aoun* Platform

3.5 Data Flow Diagrams (DFDs)

3.5.1 DFD Symbols

DFDs use standardized symbols:

- **External Entity:** Rectangle (e.g., Student, Driver, Payment Gateway).
- **Process:** Circle (e.g., 1.0 Authenticate User).
- **Data Store:** Open rectangle (e.g., D1: Users).
- **Data Flow:** Labeled arrow (e.g., Booking Request).

3.5.2 Construction Approach

DFDs are constructed by identifying entities, processes, data stores, and flows.

3.5.3 Types of DFDs

DFDs include Context (Level-0), Level-1, Current Logical, Current Physical, New Logical, and New Physical diagrams.

3.5.4 Context Diagram (Level-0)

Shows *Aoun* as a single process interacting with Students, Drivers, Hosts, Admins, and Payment Gateways, with flows like Booking Request and Trip Creation.

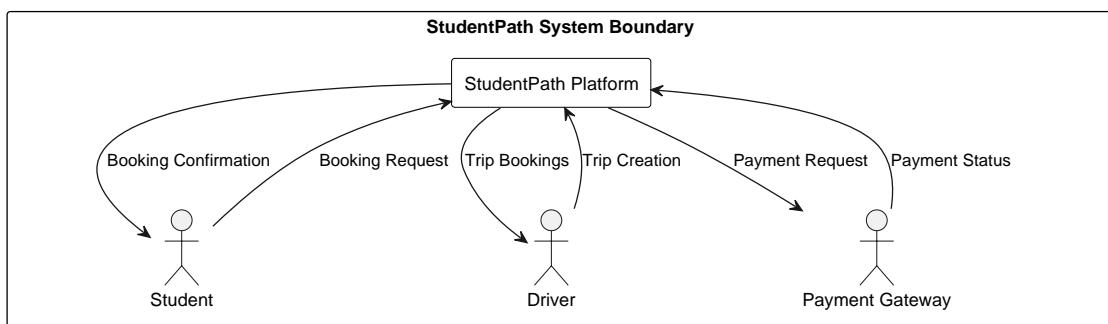


Figure 3.3 – Context Diagram (Level-0) for the *Aoun* Platform

3.5.5 Level-1 DFD

Decomposes the Transportation Module into processes:

- **Transportation:** Process 2.0 (Process Booking) queries Trips Database (D2) via ASP.NET Core, integrating Google Maps API for real-time tracking.

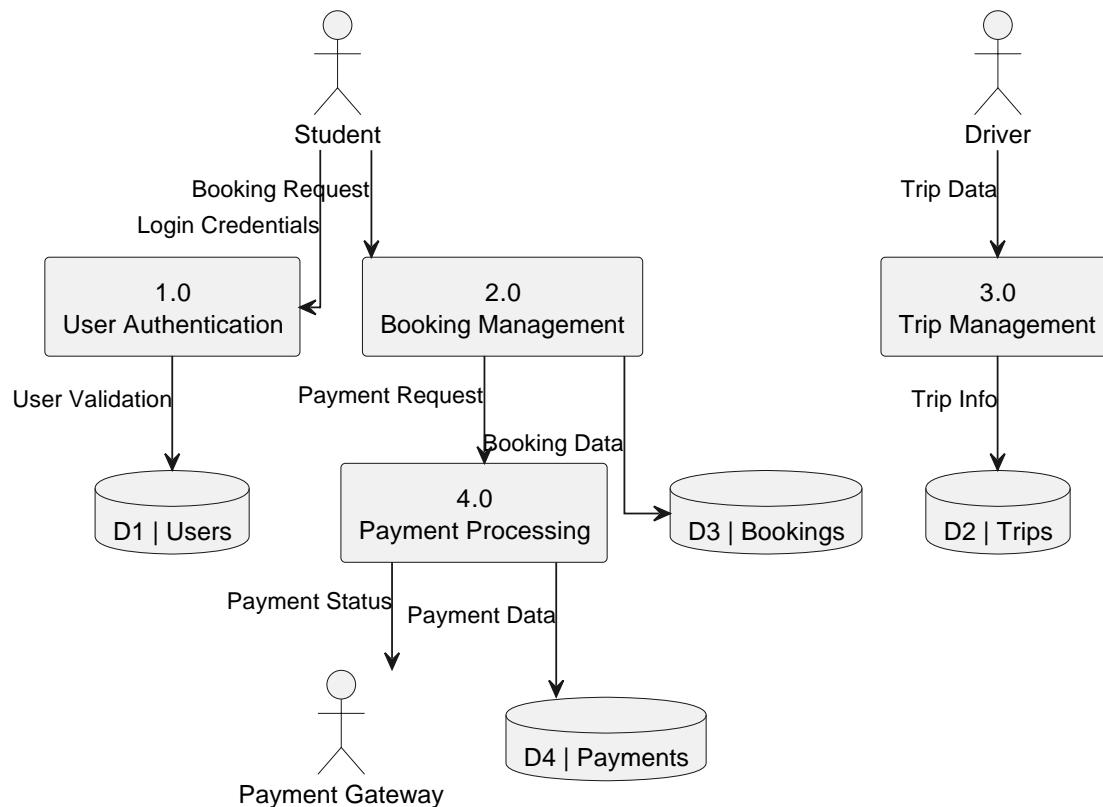


Figure 3.4 – Level-1 Data Flow Diagram for the Aoun Platform

3.5.6 Current Physical DFD

Details implementation: Flutter app sends Booking Request to ASP.NET Core, querying SQL Server and invoking Google Maps API for location updates.

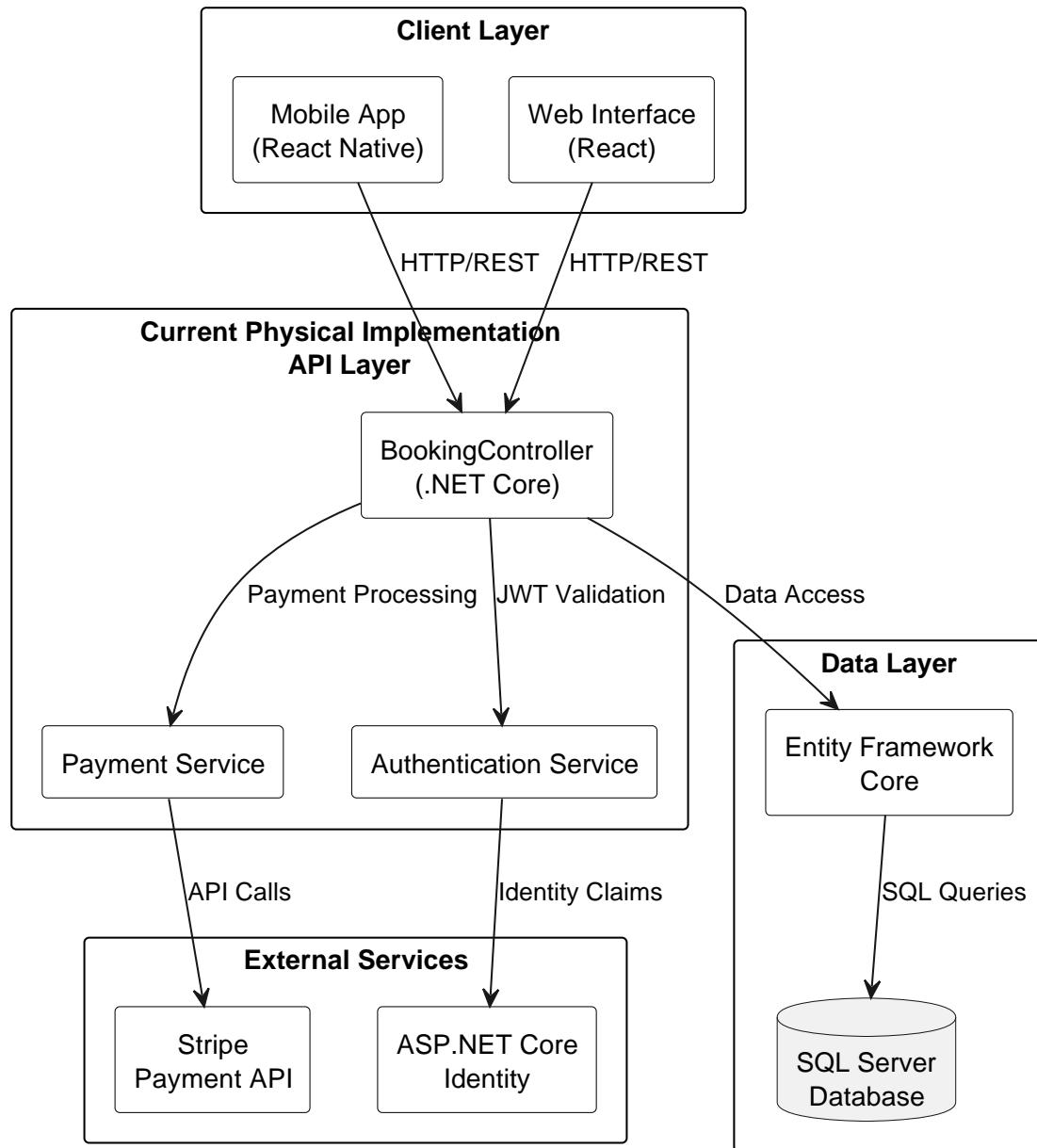


Figure 3.5 – Current Physical Data Flow Diagram

3.5.7 Current Logical DFD

Abstracts the flow: Student submits Booking Request, transformed into Confirmation via Process Booking.

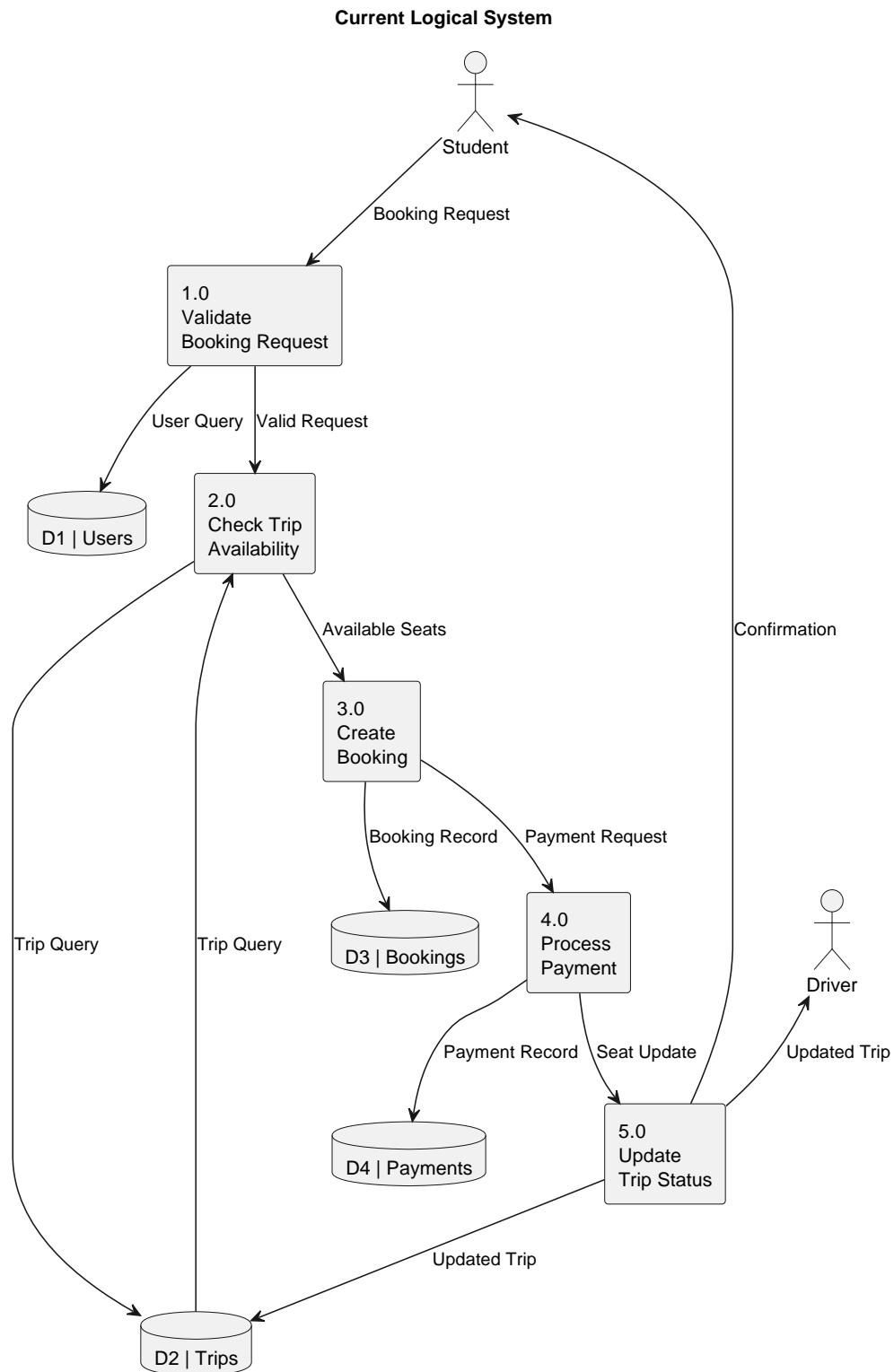


Figure 3.6 – Current Logical Data Flow Diagram

3.5.8 New Logical DFD

Proposes enhancements, e.g., adding real-time notifications for Booking status updates via Firebase Cloud Messaging.

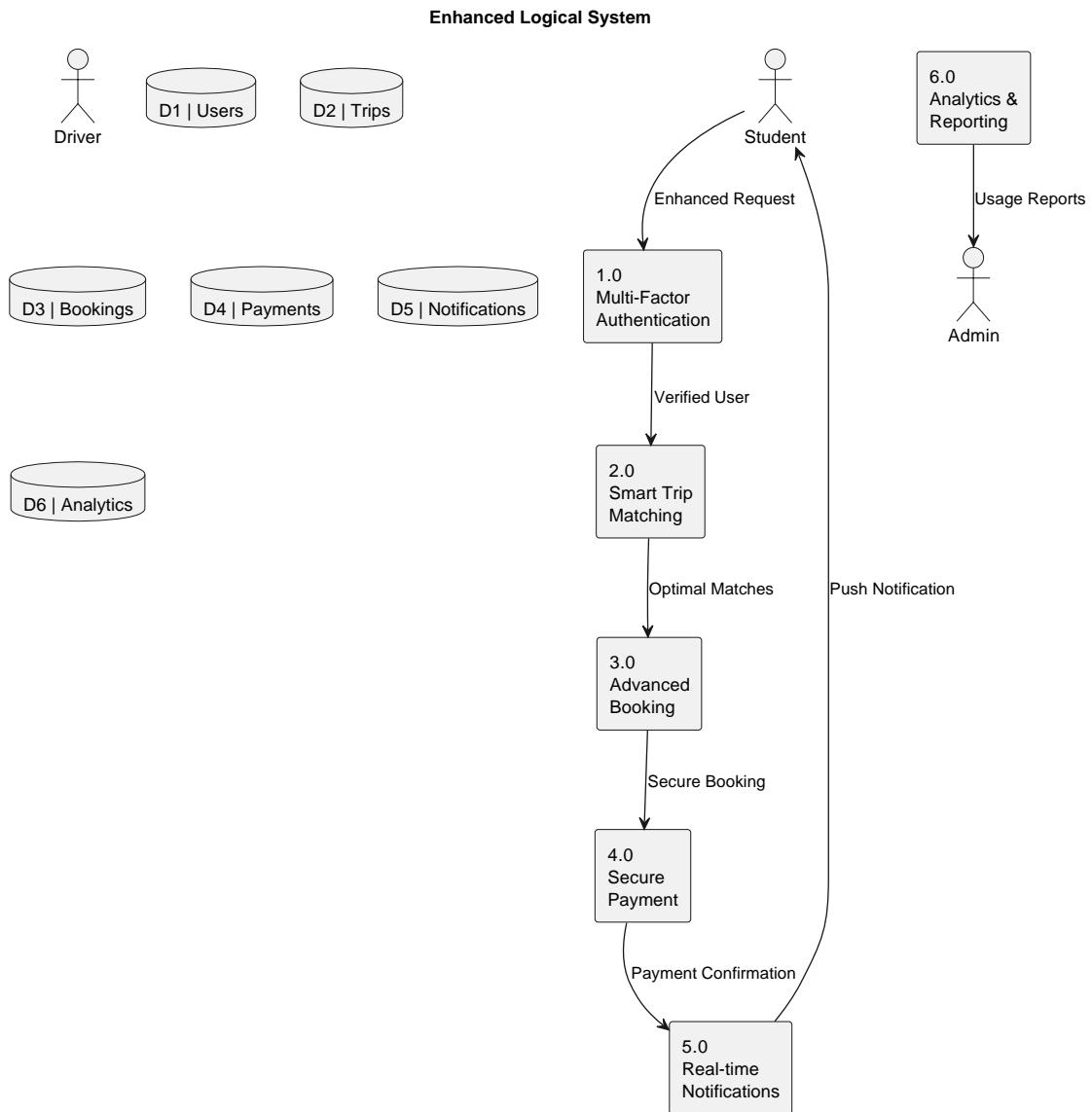


Figure 3.7 – New Logical Data Flow Diagram with Enhanced Features

3.5.9 New Physical DFD

Details enhanced implementation, e.g., integrating WebSocket for real-time updates alongside FCM for notifications.

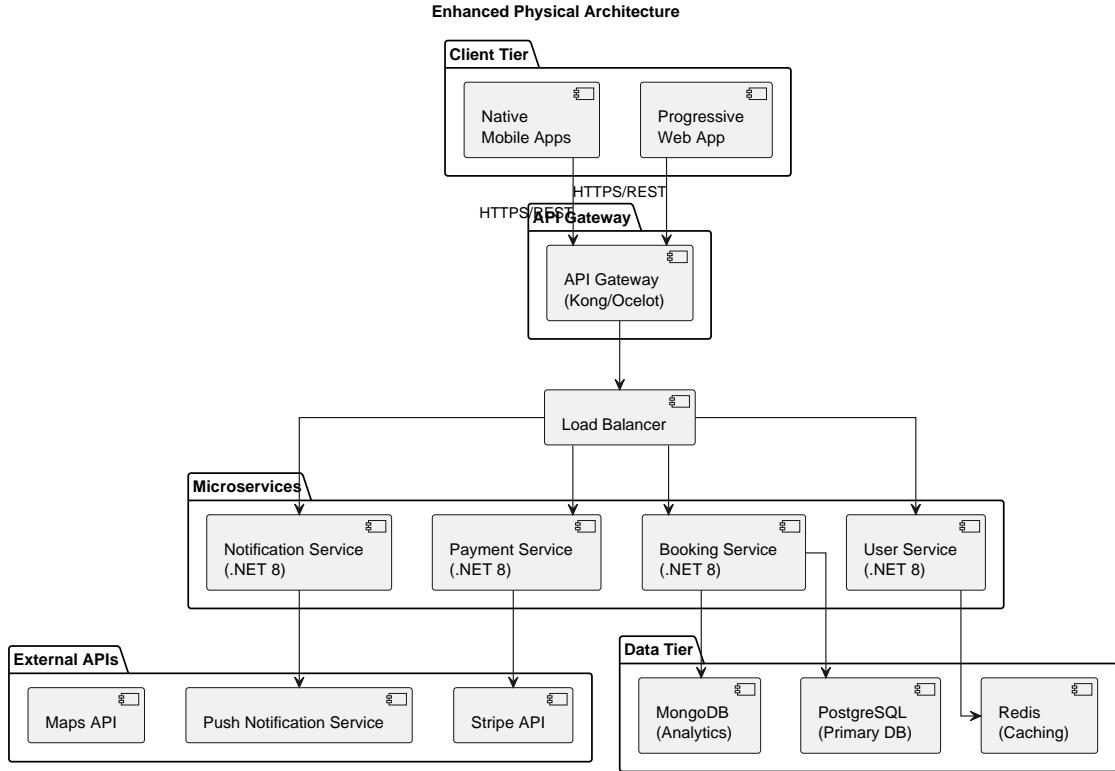


Figure 3.8 – New Physical Data Flow Diagram

3.5.10 DFD Rules

- **Data Conservation:** Inputs produce outputs.
- **Naming:** Processes (verb-noun), Data Stores (nouns).
- **Leveling:** 5-9 processes per level.
- **Specific:** Authentication required, atomic payments, status tracking.

3.6 Unified Modeling Language (UML) Diagrams

3.6.1 Architectural Views

- **Logical View:** Functional requirements for the Transportation Module.
- **Process View:** Process interactions for booking and tracking.

- **Development View:** Flutter app components and dependencies.
- **Physical View:** Mobile devices running the Flutter app.
- **Use Case View:** Requirements validation for Students and Drivers.

3.6.2 UML Diagrams

3.6.2.1 Class Diagram - User Hierarchy

Models User (Id, Age, Gender, UserType) with subclasses Student, Driver, Admin using TPH inheritance.

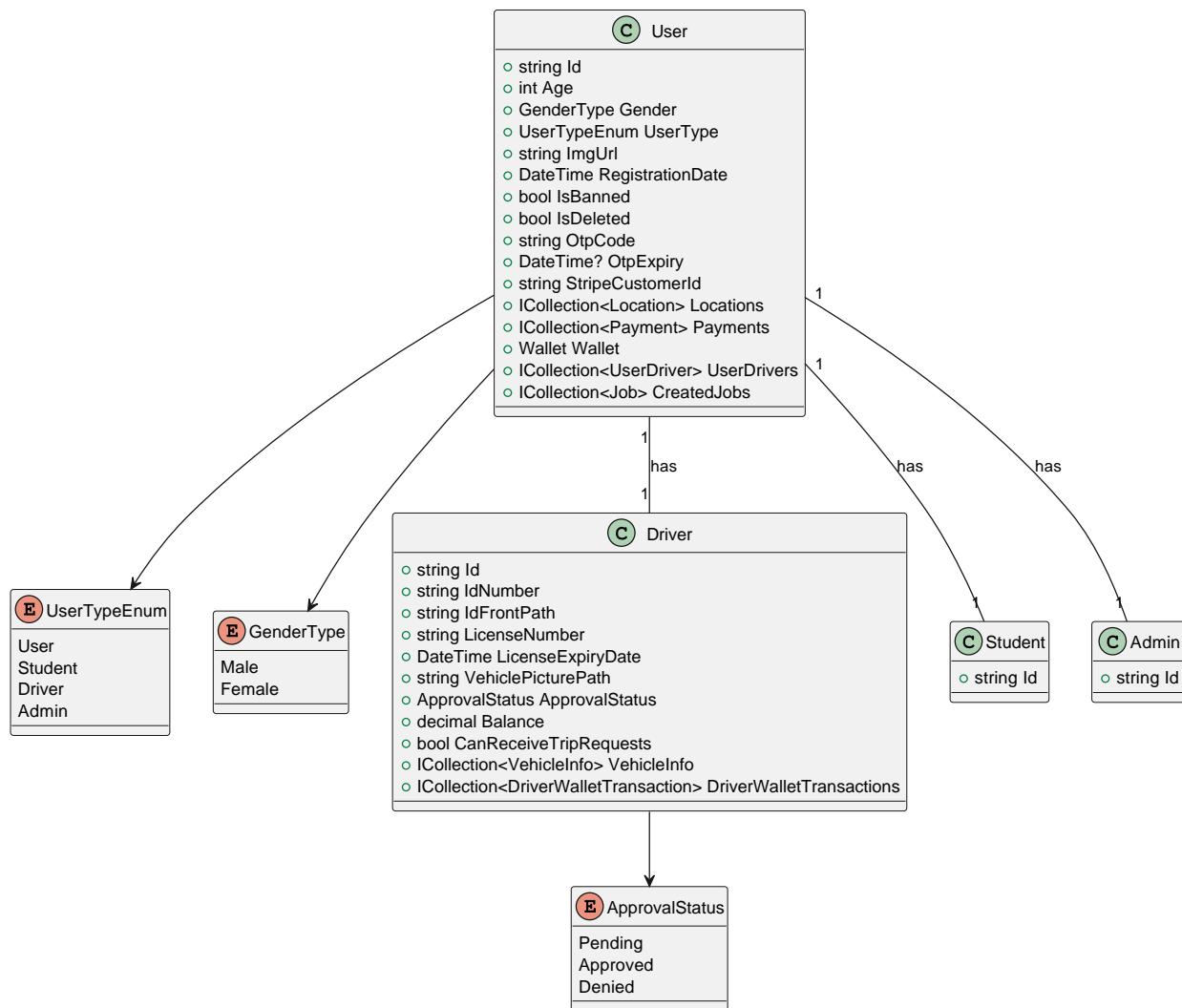


Figure 3.9 – Class Diagram for User Hierarchy

3.6.2.2 Class Diagram - Transportation System

Includes Trip (DepartureTime, AvailableSeats), Booking (TotalPrice), Payment (Amount), and VehicleInfo.

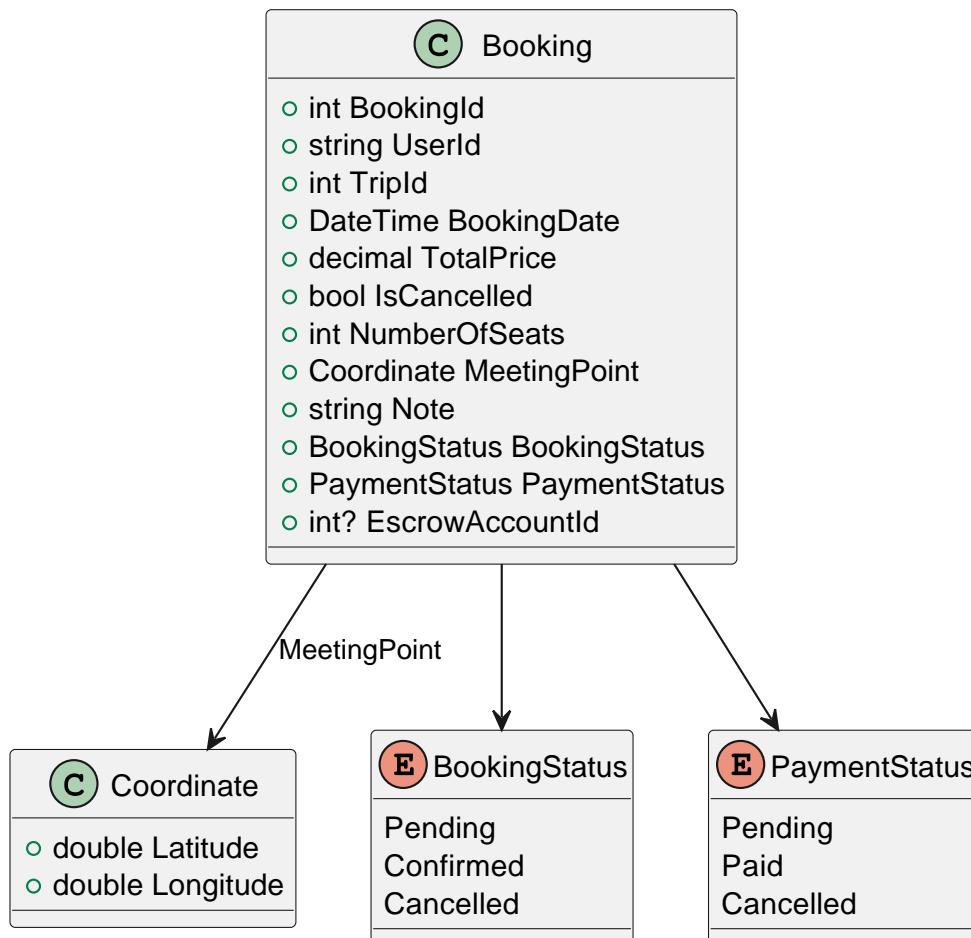


Figure 3.10 – Class Diagram for Transportation System

3.6.2.3 Sequence Diagram - Booking Process

1. Student submits Booking Request via Flutter app.
2. ASP.NET Core validates against Trips Database.
3. Payment Gateway processes transaction.
4. Confirmation sent to Student.

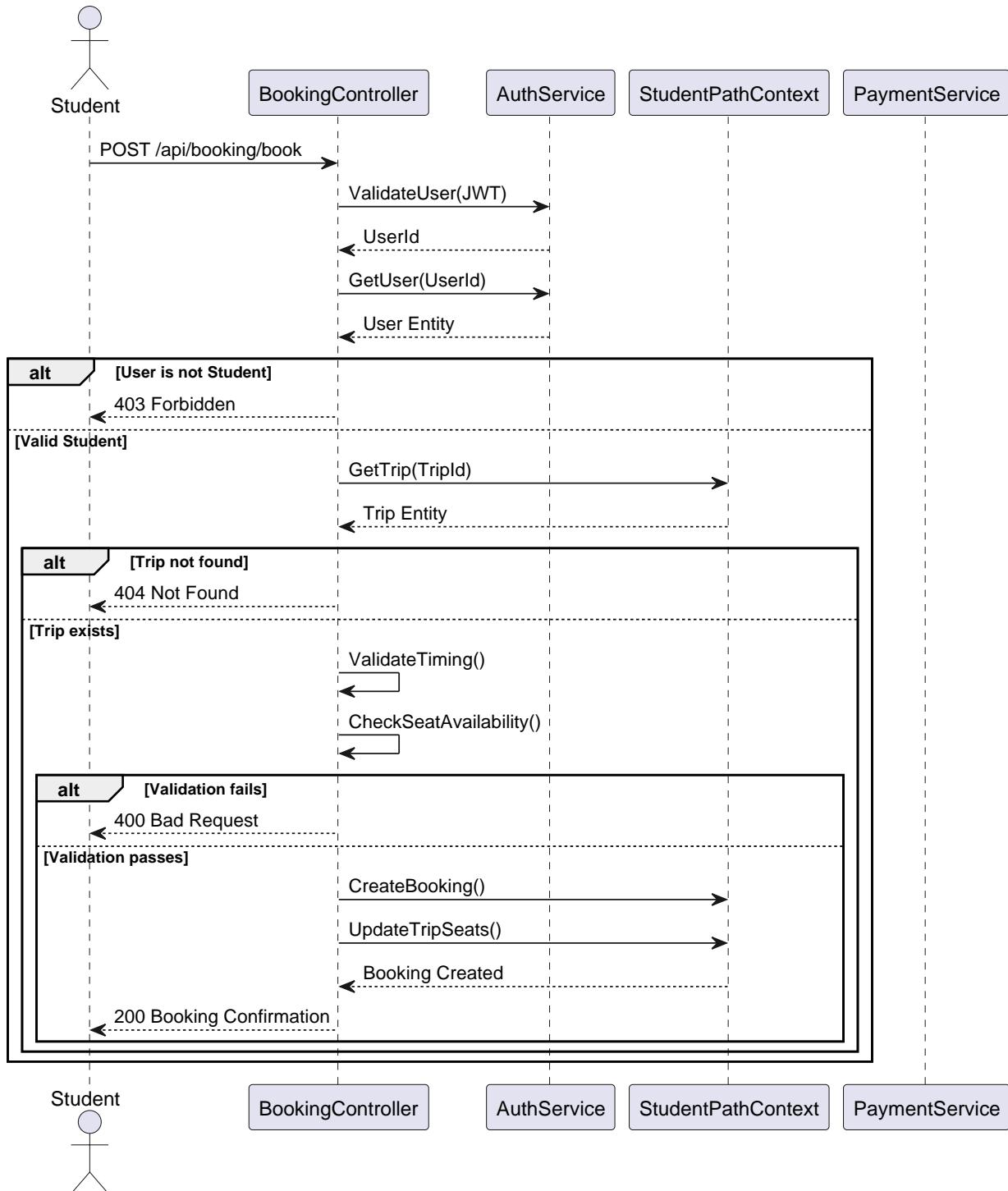


Figure 3.11 – Sequence Diagram for Booking Process

3.6.2.4 Activity Diagram - Trip Booking Flow

1. Validate Student credentials.

2. Check Trip availability.
3. Process Payment if seats available.
4. Confirm or reject Booking.

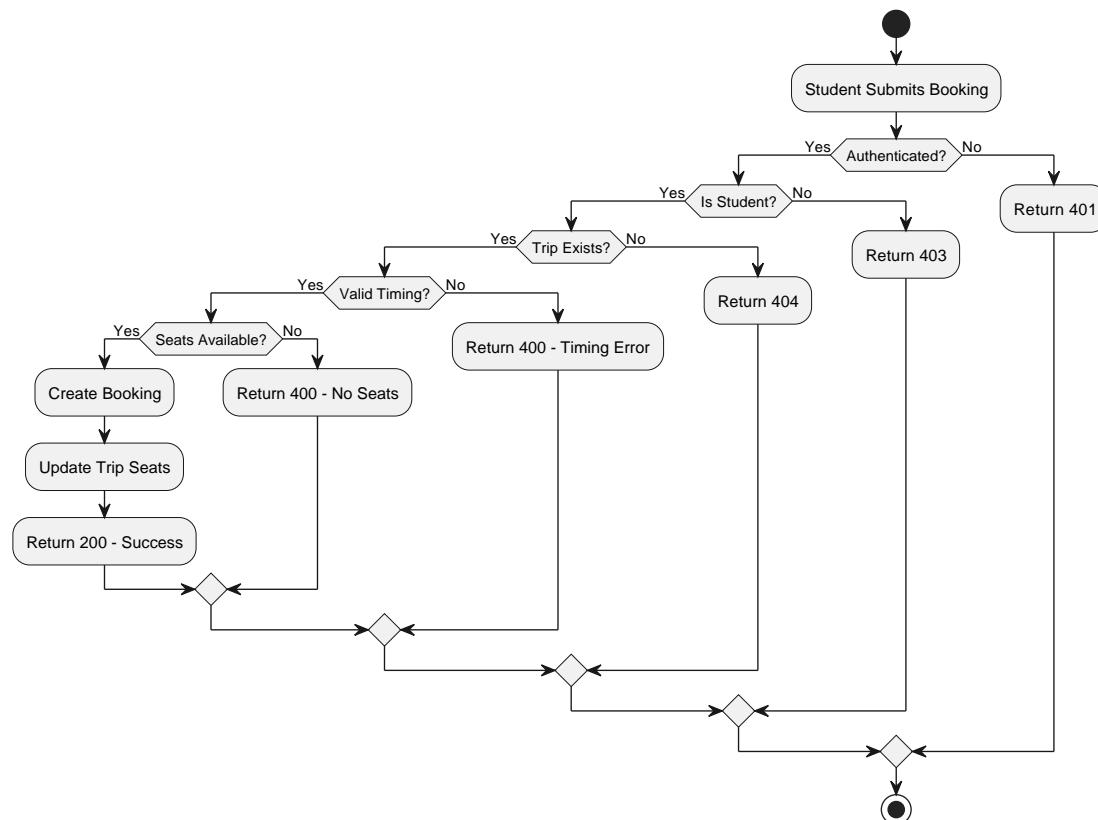


Figure 3.12 – Activity Diagram for Trip Booking Flow

3.6.2.5 State Diagram - Booking Lifecycle

States: Pending, Confirmed, Cancelled, with transitions (e.g., payment completion).

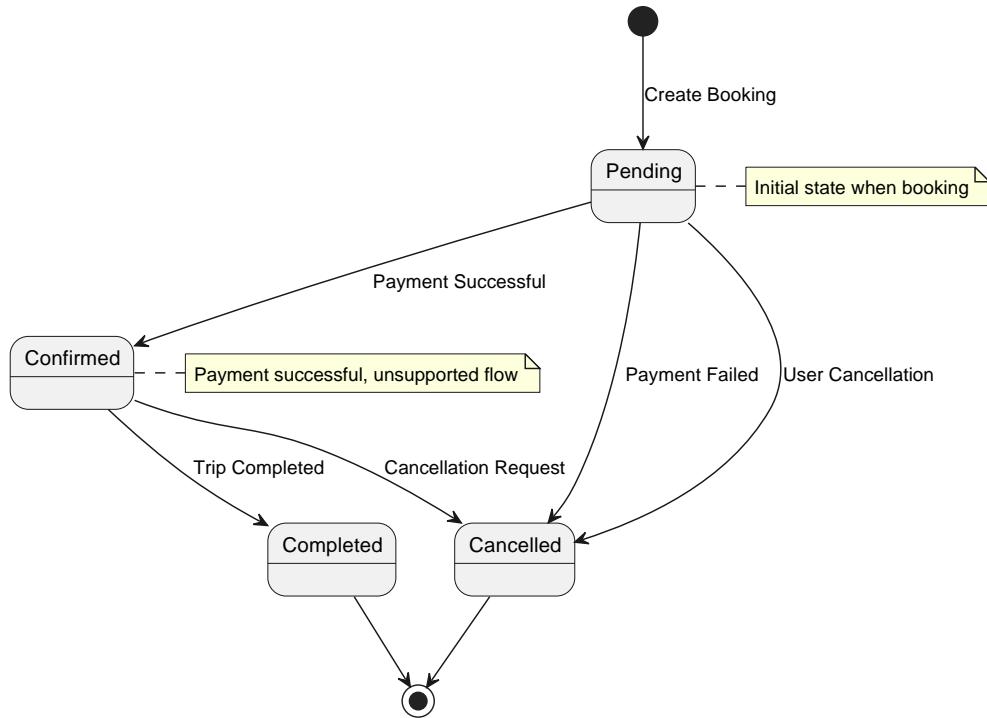


Figure 3.13 – State Diagram for Booking Lifecycle

3.6.2.6 Use Case Diagram - Real-Time Flow

Actors: Students (Book Ride, Create Trip Request, Track Driver), Drivers (Create Trip, Manage Bookings, View Student Locations).

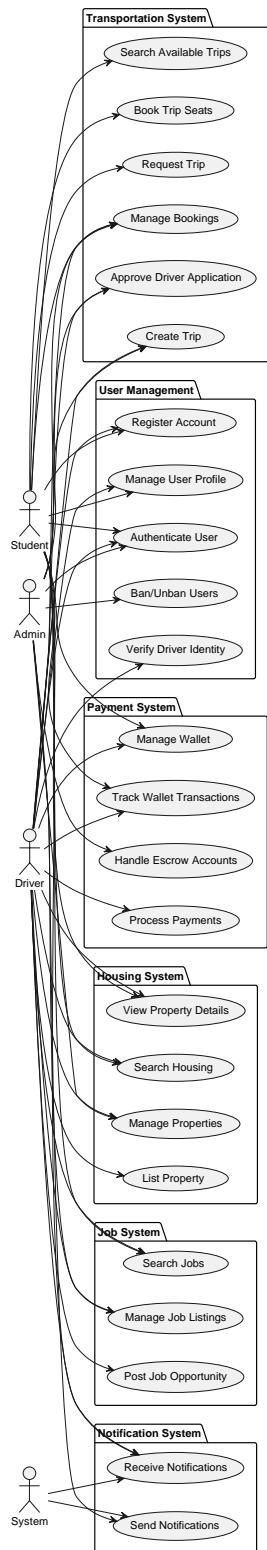


Figure 3.14 – Use Case Diagram for the Transportation Module

3.6.2.7 Component Diagram - System Architecture

Components: Flutter App (Transportation Module), ASP.NET Core Backend, SQL Server,
Google Maps API.

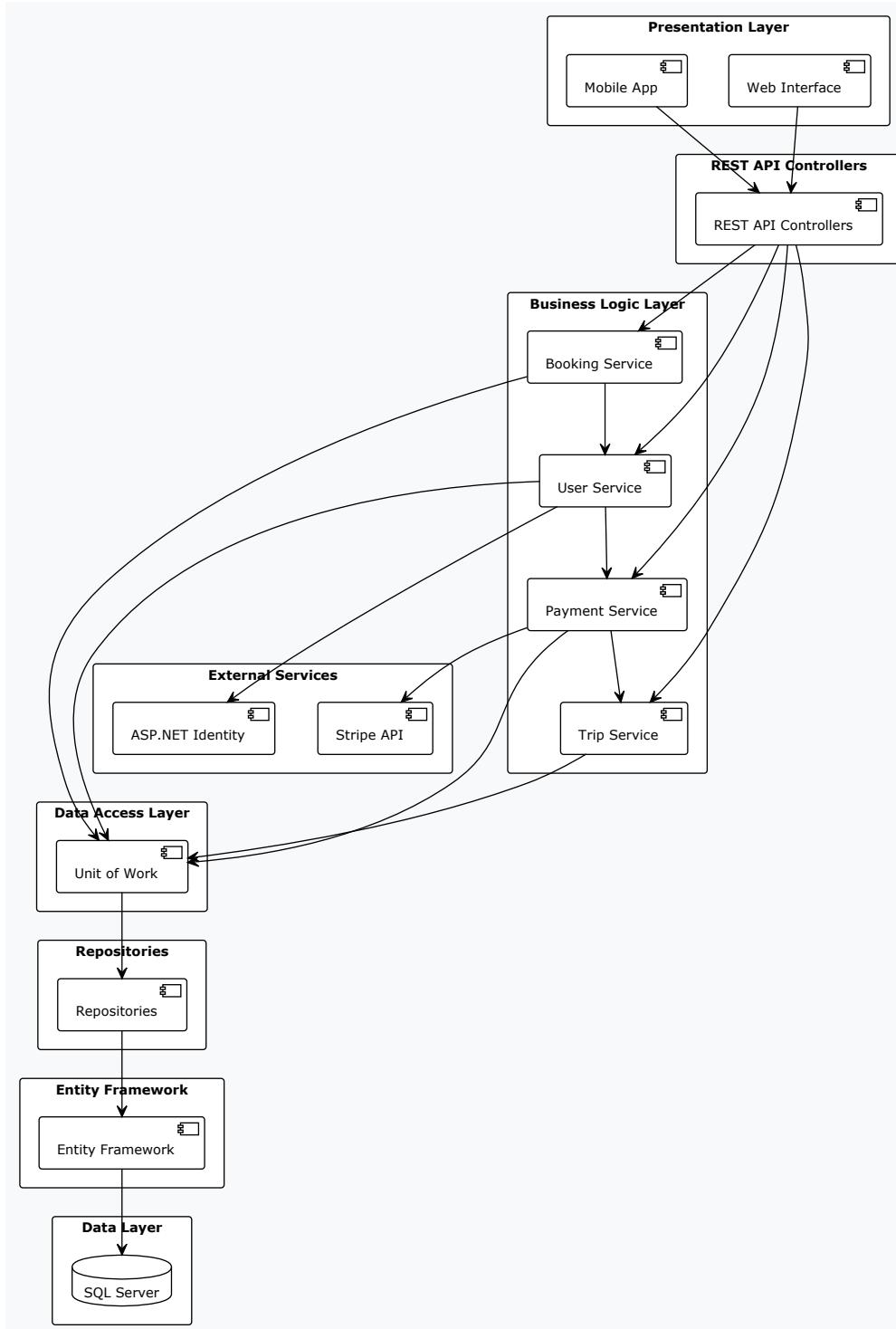


Figure 3.15 – Component Diagram for System Architecture

3.6.2.8 Deployment Diagram - Physical Architecture

Nodes: Mobile Devices (Flutter), Backend Server (ASP.NET Core), Database Server (SQL Server).

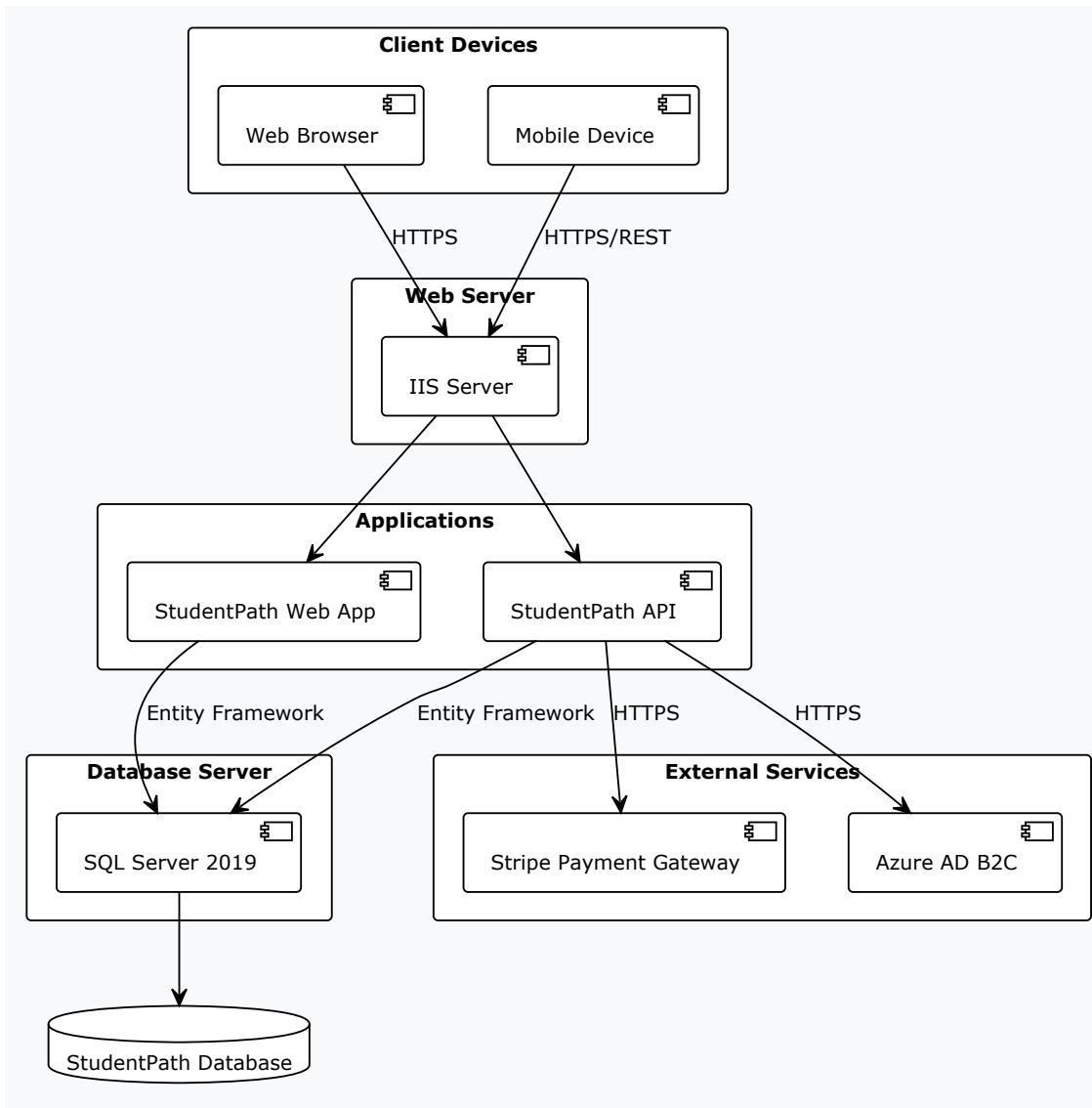


Figure 3.16 – Deployment Diagram for Physical Architecture

3.7 Data Dictionary

3.7.1 Core Entities

Table 3.2 – Core entities for the Transportation Module, their descriptions, attributes, and relationships.

Entity	Description	Key Attributes	Relationships
User	Base entity with TPH inheritance	Id, Age, Gender, UserType, IsBanned	Parent to Student/-Driver
Student	Student user specialization	Inherits from User	Creates Bookings, Trip Requests
Driver	Driver with approval workflow	LicenseNumber, ApprovalStatus, Balance	Creates Trips, owns Vehicles
Trip	Transportation offering	DepartureTime, AvailableSeats, PricePerSeat	Belongs to Driver, has Bookings
Booking	Seat reservation	BookingDate, TotalPrice, NumberOfSeats	Links User to Trip
Payment	Financial transaction	Amount, PaymentMethod, Status	Associated with Booking

3.7.2 Enumerations

Table 3.3 – Enumerations and their usage in the Transportation Module.

Enumeration	Values	Usage
UserTypeEnum	User, Student, Driver, Admin	User classification
GenderType	Male, Female	User demographics
ApprovalStatus	Pending, Approved, Denied	Driver verification
BookingStatus	Pending, Confirmed, Cancelled	Booking lifecycle
PaymentStatus	Pending, Completed, Failed, Refunded	Payment processing
TripStatus	Planned, Active, Completed, Cancelled	Trip lifecycle

3.7.3 Complex Types

Table 3.4 – Complex types and their descriptions for the Transportation Module.

Type	Properties	Description
Coordinate	Latitude: double, Longitude: double	GPS coordinates for trip points

3.7.4 Key Relationships

Table 3.5 – Key relationships and their behaviors in the Transportation Module.

Relationship	Type	Delete Behavior	Description
Trip → TripLocation	Many-to-One	Restrict	Prevents location deletion if referenced
User → Booking	One-to-Many	Cascade	User deletion removes bookings
Booking → Payment	One-to-Many	Restrict	Payments remain for auditing
Driver → VehicleInfo	One-to-Many	Cascade	Driver deletion removes vehicles

3.7.5 Business Rules

Table 3.6 – Business rules enforced by the Transportation Module.

Rule	Entity	Description
Seat Availability	Trip	AvailableSeats ≥ 0
Booking Timing	Booking	Created at least 1 hour before departure
User Type Validation	Booking	Only Students create bookings
Payment Amount	Payment	Amount must be positive
Driver Approval	Trip	Only Approved drivers create trips

3.8 Security and Scalability

- **Authentication:** The mobile app uses JWT with bcrypt hashing via `firebase_auth`, supporting Google and Facebook Sign-In alongside email/password login.
- **Authorization:** Role-based access control (Student, Driver) enforced in the Flutter app and backend.
- **Data Privacy:** TLS encryption for API calls, secure storage of user data in `hive`, and compliance with data protection standards.
- **Input Validation:** Prevents injection attacks through client-side validation in Flutter and server-side checks in ASP.NET Core.
- **Scalability:** The SOA design allows the Flutter app to scale with load balancing, while `hive` ensures efficient local data management.

Table 3.7 – Summary of Aoun’s core modules, their functionalities, and technology stack.

Module	Functionality	Technology Stack
Transportation	Manages ride booking, driver assignment, and real-time tracking across Android/iOS	Flutter, Dart, <code>google_maps_flutter</code> , <code>geolocator</code> , <code>hive</code> , <code>firebase_auth</code> , Google Maps API, ASP.NET Core
Housing	Facilitates accommodation listing and booking	React.js, Bootstrap, ASP.NET Core, SQL Server
Activities	Supports event participation and community interactions	React.js, RESTful APIs

Chapter 4

Used Technologies

4.1 Introduction

This chapter outlines the technologies used to develop the *Aoun* platform, a multi-module solution addressing transportation, housing, activities, and job opportunities for Egyptian university students. Developed at Kafr Elsheikh University, Faculty of Computers and Information, under the supervision of Assoc. Prof. Reda M. Hussien, *Aoun* adopts a service-oriented architecture (SOA) as described in [Chapter 3](#). For the Transportation Module, a cross-platform mobile application built with Flutter and Dart, the selected technologies ensure high performance, scalability, security, and an enhanced user experience. This chapter details each technology, its features, rationale, and role, with a focus on the mobile application, paving the way for implementation details in ??.

4.2 Frontend Technologies

4.2.1 Flutter and Dart

Flutter, Google's open-source UI framework, powers the Transportation Module as a cross-platform mobile application for Android and iOS, using the Dart programming language.

When Flutter launched in 2018, it mainly supported mobile app development. Flutter now supports application development on six platforms: iOS, Android, the web, Windows, MacOS,

and Linux. [?]

- **Version:** Flutter 3.29.0, Dart 3.7.0.

- **Features:**

- ▶ Hot Reload: Enables real-time code updates, boosting productivity.
- ▶ Rich Widget Library: Offers Material Design and Cupertino widgets for consistent UI across platforms.
- ▶ Native Performance: Compiles to ARM code for fast execution.
- ▶ Built-in Testing: Supports unit, widget, and integration testing.
- ▶ Localization: Supports Arabic/English and RTL layouts via `flutter_localizations` and `intl`.

- **Supporting Libraries:**

- ▶ **State Management:** `flutter_bloc` 9.1.0 and `provider` 6.1.2 implement the BLoC pattern for reactive UI updates, managing states for trip booking, tracking, and driver assignments.
- ▶ **Local Storage:** `hive` 2.2.3 and `hive_flutter` 1.1.0 cache trip details, booking requests, and user preferences for offline mode; `shared_preferences` 2.3.3 stores persistent settings like user preferences.
- ▶ **Location Services:** `geolocator` 13.0.2 and `location` 8.0.0 provide real-time location updates every 10 seconds or 100 meters; `geocoding` 3.0.0 converts coordinates to readable addresses.
- ▶ **Maps:** `google_maps_flutter` 2.10.1 displays trip routes, meeting points, and driver locations; `flutter_map` 8.0.0 supports alternative map views.
- ▶ **Payments:** `flutter_stripe` 11.5.0 processes secure credit card payments for booking requests.
- ▶ **Networking:** `dio` 5.8.0+1 handles HTTP requests to RESTful APIs for backend communication; `connectivity_plus` 6.1.3 monitors network status to enable offline synchronization.

- ▶ **UI/UX:** `flutter_svg 2.0.17` renders scalable vector graphics for icons; `iconsax 0.0.8` and `fluentui_system_icons 1.1.265` provide custom icon sets; `shimmer 3.0.0` adds loading effects; `carousel_slider 5.1.1` and `smooth_page_indicator 1.2.0+3` create dynamic trip listing carousels; `flutter_pannable_rating_bar 2.7.2+1` enables interactive driver ratings; `flutter_flip_card 0.0.6` adds card animations; `animations 2.0.11` enhances UI transitions; `cupertino_icons 1.0.8` provides iOS-style icons; `flag 7.0.0` displays country flags.
 - ▶ **Forms and Input:** `flutter_form_builder 10.0.0` creates dynamic forms for booking requests; `pinput 5.0.0` handles PIN inputs for authentication; `intl_phone_field 3.2.0` validates phone numbers for user registration.
 - ▶ **Localization:** `flutter_localizations` (SDK) and `intl 0.20.2` support Arabic/English with RTL layouts using `.arb` files, localizing key screens such as Search, Booking, Profile, and Settings.
 - ▶ **Other:** `image_picker 1.1.2` allows drivers to upload profile photos for verification; `url_launcher 6.3.1` opens external links (e.g., payment gateways); `uuid 4.5.1` generates unique IDs for booking requests; `device_preview 1.2.0` tests UI across different device sizes; `flutter_dotenv 5.2.1` manages environment variables for API keys; `webview_flutter 4.13.0` displays web content within the app; `syncfusion_flutter_charts 28.2.12` visualizes trip statistics and booking trends; `flutter_card_swiper 7.0.2` enhances trip card interactions.
- **AI Integration:** The Flutter app interacts with backend AI models (Content-Based Recommendation System, Sentiment Classification Model, and Face Verification Model) via RESTful APIs using `dio`. For example, it sends user preferences to fetch recommended trips, submits textual feedback for sentiment analysis, or uploads driver photos for identity verification, receiving JSON responses to update the UI.
 - **Future Enhancements:**
 - ▶ **Social Media Login:** Planned integration with `firebase_auth` (version 5.3.0) to support Google and Facebook Sign-In, enhancing user onboarding.

- ▶ Push Notifications: Planned use of Firebase Cloud Messaging (FCM, version 8.0.0) to send notifications for booking confirmations, trip updates, and driver arrival alerts.
- **Rationale:** Chosen for its cross-platform development capabilities, native performance, and seamless integration with Google Maps API and payment gateways like Stripe.
- **Role in Aoun:** Drives the Transportation Module, enabling drivers to create trips and manage assignments, and passengers to search, book, and track rides in real-time, with support for offline functionality and multi-language interfaces.

4.2.2 React.js

ReactJS is a component-based JavaScript library used to build dynamic and interactive user interfaces. It simplifies the creation of single-page applications (SPAs) with a focus on performance and maintainability. It is developed and maintained by Facebook. The latest version of React is React 19. Uses a virtual DOM for faster updates. Supports a declarative approach to designing UI components. Ensures better application control with one-way data binding. [?]

- **Version:** React 18.2.0.
- **Features:**
 - ▶ Virtual DOM: Optimizes UI updates for performance.
 - ▶ Component-Based Architecture: Supports reusable components (e.g., Navbar, Job Listings).
 - ▶ React Router 6.22.2: Manages navigation in single-page applications.
 - ▶ Context API: Handles global state for authentication and preferences.
- **Supporting Tools:**
 - ▶ Vite 5.1.4: Fast development server and build tool.
 - ▶ Axios 1.9.0: HTTP client for API calls.
 - ▶ JWT Decode 4.0.0: Decodes JWT tokens for authentication.

Table 4.1 – Flutter Dependencies for Transportation Module

lightgray Dependency	Version	Role
flutter_bloc	9.1.0	State management with BLoC pattern
provider	6.1.2	State management for UI updates
hive	2.2.3	Offline storage for trip and booking data
hive_flutter	1.1.0	Flutter integration for Hive
shared_preferences	2.3.3	Persistent key-value storage
geolocator	13.0.2	Real-time location services
location	8.0.0	Background location tracking
geocoding	3.0.0	Coordinate-to-address conversion
google_maps_flutter	2.10.1	Trip route visualization and tracking
flutter_map	8.0.0	Alternative map views
flutter_stripe	11.5.0	Credit card payment processing
dio	5.8.0+1	HTTP requests to RESTful APIs
connectivity_plus	6.1.3	Network status monitoring
flutter_svg	2.0.17	SVG icon rendering
iconsax	0.0.8	Custom icon set
fluentui_system_icons	1.1.265	System icons
shimmer	3.0.0	Loading animations
carousel_slider	5.1.1	Dynamic trip carousels
smooth_page_indicator	1.2.0+3	Carousel navigation
flutter_pannable_rating_bar	7.2+1	Interactive driver ratings
flutter_flip_card	0.0.6	Card flip animations
flutter_card_swiper	7.0.2	Interactive trip card swiping
animations	2.0.11	Smooth UI transitions
flutter_form_builder	10.0.0	Dynamic booking forms
pinput	5.0.0	PIN input for authentication
intl_phone_field	3.2.0	Phone number input
flutter_localizations	SDK	Arabic/English and RTL support
intl	0.20.2	Localization formatting
image_picker	1.1.2	Photo uploads for driver profiles
url_launcher	6.3.1	External link handling
uuid	4.5.1	Unique ID generation
device_preview	1.2.0	UI testing across devices
flutter_dotenv	5.2.1	Environment variable management
webview_flutter	4.13.0	Web content display
syncfusion_flutter_charts	28.2.12	Trip statistics visualization
cupertino_icons	1.0.8	iOS-style icons
flag	7.0.0	Country flag display
file_picker	3.1.1	File selection and download

- ▶ TailwindCSS: Utility-first CSS framework for styling.
 - ▶ Framer Motion 11.0.8: Animations for enhanced UX.
 - ▶ Swiper 11.2.6: Carousel for dynamic content.
 - ▶ ESLint: Ensures code quality.
- **Rationale:** Selected for modularity, scalability, and a rich ecosystem.
 - **Role in Aoun:** Powers Housing (property listings), Activities (event management), and Jobs (job creation, search, and management) Modules.

4.3 Backend Technologies

4.3.1 ASP.NET Core

ASP.NET is a popular web-development framework for building web apps on the .NET platform.

ASP.NET Core is the open-source version of ASP.NET, that runs on macOS, Linux, and Windows. ASP.NET Core was first released in 2016 and is a re-design of earlier Windows-only versions of ASP.NET. [?]

- **Version:** ASP.NET Core 8.0.
- **Features:**
 - ▶ Dependency Injection: Enhances modularity and testability.
 - ▶ Entity Framework Core 8.0.0: Simplifies database interactions via ORM.
 - ▶ Repository Pattern.
 - ▶ Dependency Inversion (SOLID).
 - ▶ Cross-Platform: Runs on Windows, macOS, and Linux.
 - ▶ High Performance: Handles concurrent requests efficiently.
- **Supporting Libraries:**
 - ▶ AutoMapper 13.0.1: Maps DTOs to entities.

- ▶ MailKit 4.8.0: Sends email notifications.
- ▶ ASP.NET Core Identity: Manages user authentication.
- **Rationale:** Chosen for scalability, security, and SQL Server integration.
- **Role in Aoun:** Manages business logic, processes requests (e.g., job creation, payment processing), and integrates with external services.

4.3.2 C-Sharp

C# is the primary language for backend development.

- **Version:** .NET 8.0.
- **Features:** Strong typing, LINQ, async/await for asynchronous operations.
- **Rationale:** Selected for robustness and compatibility with ASP.NET Core.
- **Role in Aoun:** Implements API endpoints, business logic, and data processing.

4.4 Database Technologies

4.4.1 SQL Server

Microsoft SQL Server is a relational database management system (RDBMS). Applications and tools connect to a SQL Server instance or database, and communicate using Transact-SQL (T-SQL). [?]

- **Version:** SQL Server 2022.
- **Features:**
 - ▶ Data Integrity: Enforces primary keys, foreign keys, and triggers.
 - ▶ Scalability: Supports indexing and sharding.
 - ▶ Security: Role-based access and encryption.

- ▶ Lazy Loading: Enabled via Entity Framework Core proxies.
- **Rationale:** Chosen for reliability, .NET integration, and complex query support.
- **Role in Aoun:** Stores entities like Users, Trips, Jobs, and Bookings.

4.5 AI and Machine Learning Technologies

4.5.1 Content-Based Recommendation System

The Content-Based Recommendation System suggests personalized trips to students based on their preferences, travel history, and location data, enhancing the user experience in the Transportation Module.

- **Tools:** Python, Pandas, NumPy, Scikit-learn.
- **Features:**
 - ▶ Haversine Formula: Calculates spatial proximity between user and trip locations.
 - ▶ Cosine Similarity: Matches trips to user profiles based on features like price, time, distance, and amenities.
 - ▶ Preprocessing: Uses OneHotEncoder for categorical features (e.g., location IDs, trip type) and StandardScaler for numerical features (e.g., price, distance). Binary features (e.g., Wi-Fi, air conditioning) are passed directly.
- **Example Flow:**
 - ▶ Input: User preferences and travel history via dio API calls from Flutter.
 - ▶ Process: Features are encoded, scaled, and ranked using cosine similarity.
 - ▶ Output: Top 5 trip IDs returned to the Flutter app for display.
- **Rationale:** Improves user engagement by offering relevant trip suggestions.
- **Role in Aoun:** Enhances the Transportation Module by recommending tailored trips, integrated via ASP.NET Core APIs.

- **Scikit-learn:** Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib. [?]

4.5.2 Sentiment Classification Model

The Sentiment Classification Model analyzes user feedback to assign star ratings, ensuring high service quality for trips and drivers.

- **Tools:** FastText, NLTK, Flask.
- **Features:**
 - ▶ Real-Time Analysis: Processes textual reviews using a pre-trained FastText model.
 - ▶ Star Rating Mapping: Converts sentiment scores to 1–5 star ratings.
 - ▶ Optimized Speed: Lightweight API design for real-time feedback submission.
- **Example Flow:**
 - ▶ Input: Review text submitted via dio from Flutter (e.g., "The driver was punctual!").
 - ▶ Process: Text is tokenized, processed by FastText, and mapped to a sentiment score.
 - ▶ Output: Star rating returned to the Flutter app and stored in SQL Server.
- **Rationale:** Automates feedback evaluation for transparency and service improvement.
- **Role in Aoun:** Monitors trip and driver quality, with results accessible via Flutter UI.
- **FastText:** FastText is an open-source, free library from Facebook AI Research(FAIR) for learning word embeddings and word classifications. This model allows creating unsupervised learning or supervised learning algorithm for obtaining vector representations for words. It also evaluates these models. FastText supports both CBOW and Skip-gram models. [?]

4.5.3 Face Verification Model

The Face Verification Model verifies driver identities by comparing personal photos with ID card images, strengthening platform security.

- **Tools:** FastAPI, DeepFace, Facenet, MTCNN.
- **Features:**
 - ▶ Face Recognition: Uses Facenet embeddings with a 0.6 similarity threshold.
 - ▶ Face Detection: Employs MTCNN for accurate face extraction.
 - ▶ CORS Support: Allows secure cross-origin requests from Flutter UI.
 - ▶ CPU Optimization: Disables GPU for cost-efficient performance.
- **Example Flow:**
 - ▶ Input: Driver's selfie and ID card image uploaded via `image_picker` and sent through `dio`.
 - ▶ Process: Images are cropped using MTCNN and compared using DeepFace's Facenet model.
 - ▶ Output: Boolean result (True/False) returned to Flutter for UI update.
 - ▶ Storage Policy: Images are temporarily stored in memory and deleted after verification.
- **Rationale:** Ensures only verified drivers operate, building user trust.
- **Role in Aoun:** Secures the Transportation Module by validating driver identities, integrated with ASP.NET Core.
- **DeepFace:** DeepFace is a lightweight face recognition and facial attribute analysis package for Python. You can apply facial analysis with just a few lines of code. It is fully open source and available on PyPI. All you need is to call `pip install deepface` command. It supports the most popular face recognition models including VGG-Face, Google FaceNet,

OpenFace, Facebook DeepFace, DeepID, Dlib and ArcFace. Besides, it can analyze facial attributes such as emotion, age, gender and race prediction as well in its facial attribute analysis module. [?]

4.6 External APIs and Integrations

4.6.1 Google Maps API

Google Maps API enables location-based services in the Transportation Module.

- **Version:** google_maps_flutter 2.10.1.
- **Features:** Geolocation, routing, distance calculation, real-time tracking of drivers and meeting points.
- **Rationale:** Ensures accurate trip planning and navigation.
- **Role in Aoun:** Supports trip creation, route visualization, and real-time tracking for students and drivers.

4.6.2 Payment Gateway

Payment gateways facilitate secure transactions for trip bookings.

- **Tools:**
 - ▶ flutter_stripe 11.5.0: Supports credit card processing and webhooks in the mobile app.
 - ▶ Stripe 48.0.2 (Backend): Processes payments server-side.
- **Rationale:** Ensures secure and reliable payment processing.
- **Role in Aoun:** Processes payments for trip bookings in the Transportation Module.

4.7 Security Technologies

4.7.1 JSON Web Tokens (JWT)

JWT secures API endpoints accessed by the mobile app.

- **Version:** JWT Bearer 8.0.0.
- **Features:** Stateless authentication, role-based access control (Student, Driver).
- **Rationale:** Protects user data and restricts unauthorized access.
- **Role in Aoun:** Secures API calls for authentication and authorization in the Transportation Module.

4.7.2 Transport Layer Security (TLS)

TLS encrypts data transmission between the mobile app and backend APIs.

- **Version:** TLS 1.3.
- **Features:** End-to-end encryption, data integrity, protection against man-in-the-middle attacks.
- **Rationale:** Safeguards sensitive data, such as credentials and payment details.
- **Role in Aoun:** Ensures secure communication for all API calls, including booking requests and location updates.

4.8 Development and Design Tools

- **Figma:** Designs UI/UX for the Transportation Module, ensuring Arabic/English and RTL compatibility for screens like Search, Booking, and Profile.
- **Visual Studio:** IDE for ASP.NET Core and C#.

- **Visual Studio Code:** Primary editor for Flutter and Dart, with extensions for debugging and code formatting.
- **Android Studio:** Provides emulators for testing Flutter apps on Android and iOS.
- **Git/GitHub:** Facilitates version control and team collaboration.
- **Postman:** Tests and debugs RESTful API endpoints called by dio.
- **Swagger:** Swagger is an open-source framework for designing, building.
- **Flutter DevTools:** Analyzes app performance, widget tree, and memory usage.

4.9 Testing Tools

- **Flutter Testing:** Unit, widget, and integration tests for the Transportation Module.
 - ▶ **Tools:**
 - ★ flutter_test (SDK): Supports unit, widget, and integration testing.
 - ★ mockito 5.4.4: Mocks API responses and dependencies for unit tests.
 - ★ Firebase Test Lab: Tests compatibility across Android and iOS devices.
 - ▶ **Test Types and Coverage:**
 - ★ Unit Tests (80% coverage): Validate business logic, such as booking request validation, state management with flutter_bloc, and offline caching with hive.
 - ★ Widget Tests (90% coverage): Ensure UI consistency for screens like Search, Booking, Profile, and Settings, including RTL layouts.
 - ★ Integration Tests (70% coverage): Verify end-to-end workflows, including trip booking, payment processing with flutter_stripe, and real-time tracking with google_maps_flutter.
 - ▶ **Tested Features:** Trip search, booking requests, payment processing, driver tracking, offline data caching, multi-language UI, and form validation.
- **NUnit:** Unit testing for ASP.NET Core.

- **Jest:** Unit and integration testing for React.js.
- **AI Models:** Scikit-learn (Python), metrics like accuracy, F1-score, precision, recall, confusion matrix, Postman for testing APIs.

Table 4.2 – Flutter Testing Scope

lightgray Test Type	Coverage	Tested Features
Unit Tests	80%	Booking validation, state management, offline caching
Widget Tests	90%	UI consistency for Search, Booking, Profile, Settings
Integration Tests	70%	Trip booking, payment processing, real-time tracking

4.10 Technology Selection Rationale

- **Flutter vs. React Native:** Flutter was chosen for its superior native performance, Hot Reload, and seamless integration with google_maps_flutter for real-time tracking.
- **React.js vs. Angular:** React.js was selected for lightweight design, component reusability, and ecosystem vibrancy.
- **ASP.NET Core vs. Node.js:** ASP.NET Core was preferred for scalability, type safety, and SQL Server integration.
- **SQL Server vs. MySQL:** SQL Server was chosen for robust features and .NET compatibility.
- **BLoC vs. Redux:** BLoC, combined with provider, was selected for reactive state management and compatibility with Flutter's widget architecture.
- **Hive vs. SQLite:** Hive was preferred for its lightweight, key-value storage, ideal for caching trip data offline.

- **Dio vs. http:** Dio was chosen for advanced HTTP features, such as interceptors and retry mechanisms, ensuring robust API communication.

4.11 Summary

The technologies for the Transportation Module—Flutter 3.29.0, Dart 3.7.0, and packages like `flutter_bloc`, `google_maps_flutter`, `flutter_stripe`, `hive`, `flutter_localizations`, and `intl`, alongside external APIs (Google Maps, Stripe), security measures (JWT, TLS 1.3), and development tools (Figma, Visual Studio Code, Flutter DevTools)—enable a secure, scalable, and user-friendly mobile application. Comprehensive testing with `flutter_test`, `mockito`, and Firebase Test Lab ensures reliability. Planned enhancements, such as `firebase_auth` for social media login and FCM for notifications, will further enhance functionality. These technologies align with the SOA design in [Chapter 3](#), with implementation details provided in ??.

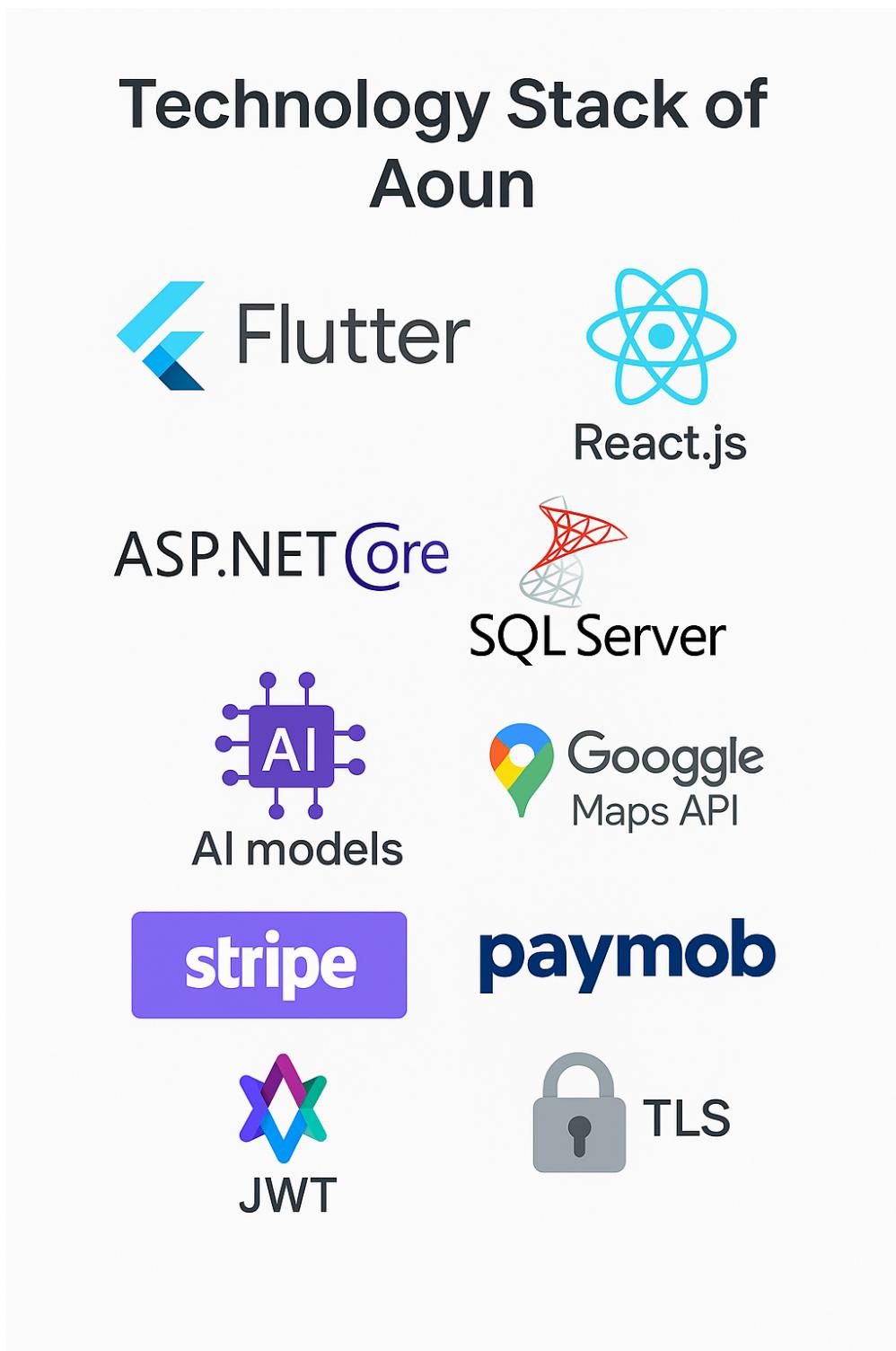


Figure 4.1 – Technology Stack of the *Aoun* Platform

Chapter 5

System Implementation

This chapter presents the comprehensive implementation of the multi-platform system, showcasing both web and mobile applications with their respective specialized functionalities. The system consists of two main web applications and a mobile application, each designed to serve specific user needs and use cases.

5.1 Main Web Application

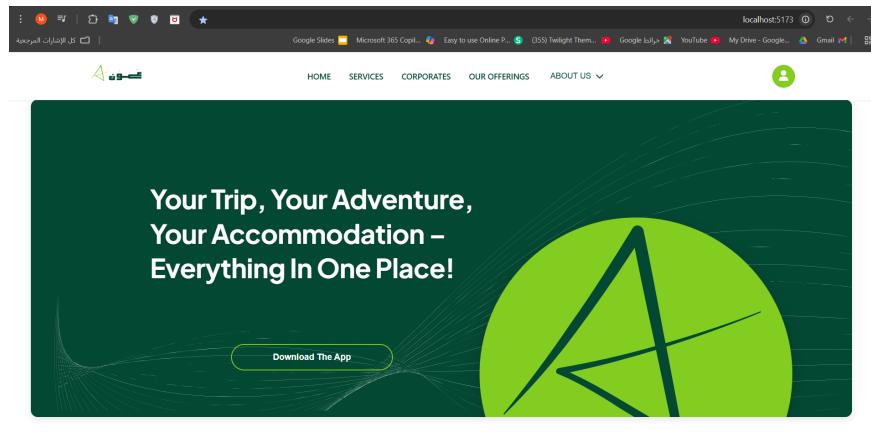
The main web application serves as the comprehensive platform offering housing and job/activity services. This application focuses on accommodation solutions and employment opportunities, providing users with full-featured interfaces for property management and career development.

5.1.1 Home Page Interface

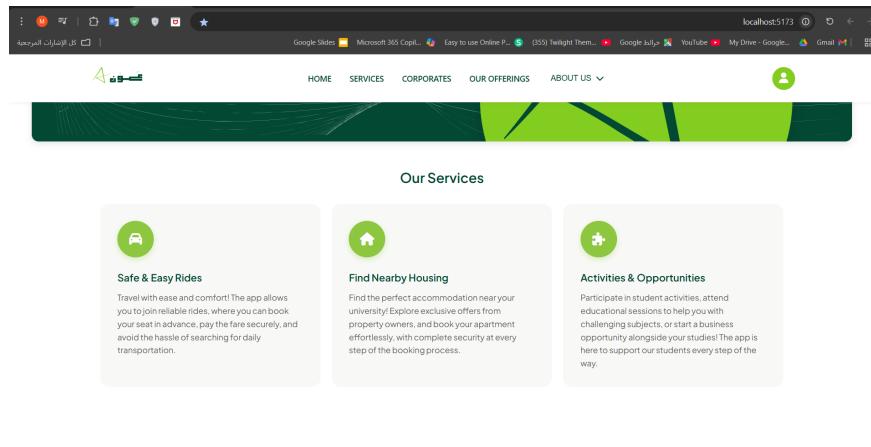
The home page serves as the primary entry point for users, showcasing the platform's main features and navigation elements. The interface is designed with a clean, modern layout that emphasizes usability and accessibility.

5.1.2 Authentication System

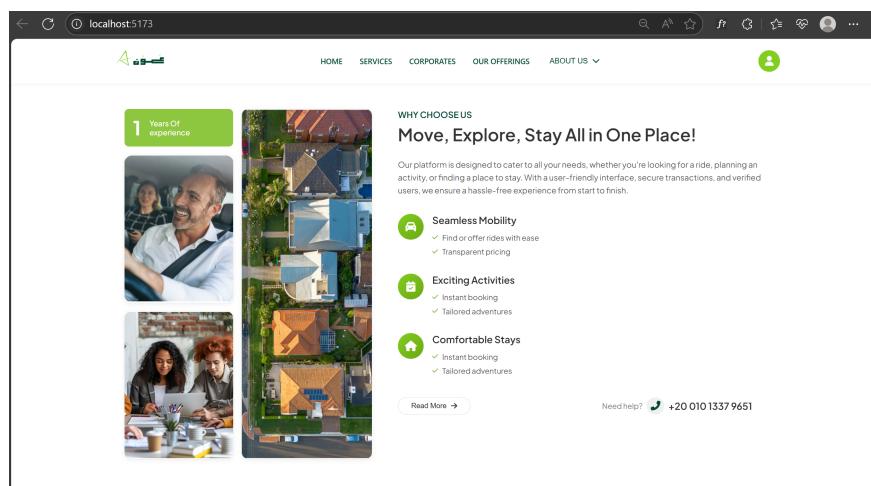
The authentication system provides secure access to the platform through a comprehensive login and registration process. The system implements modern security practices while maintaining



(a) Main navigation header and hero section showcasing primary platform features

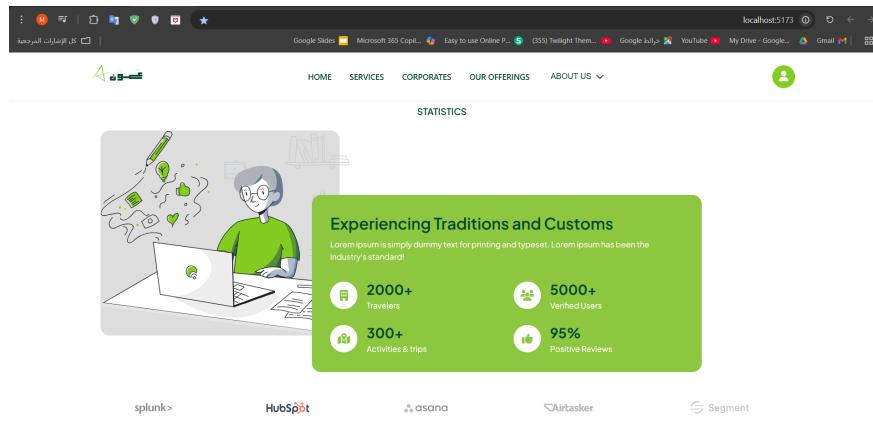


(b) Featured services section displaying housing and job opportunities

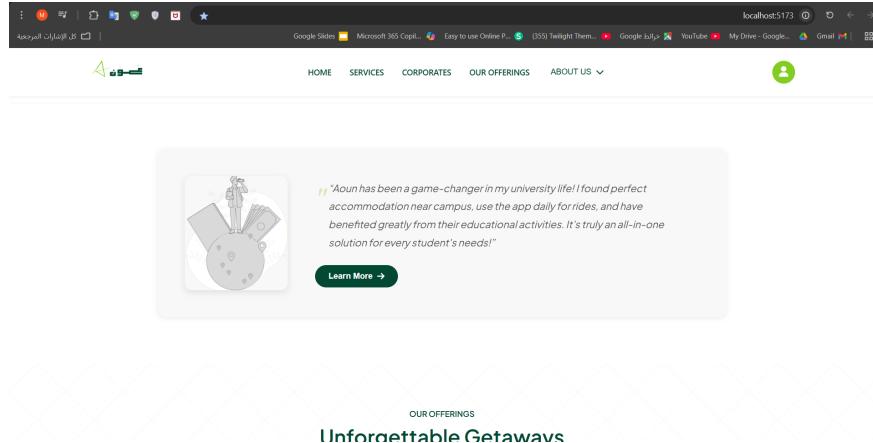


(c) Success stories and testimonials section highlighting user experiences

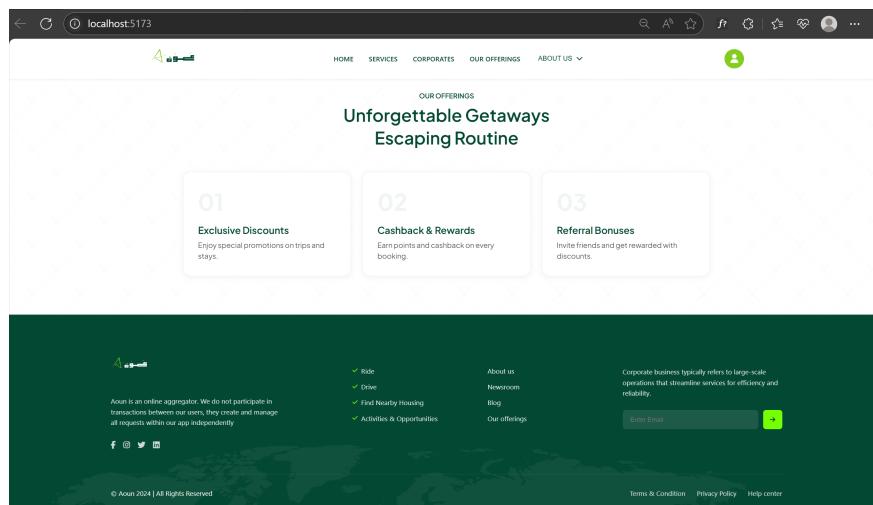
Figure 5.1 – Main Web Application - Home Page Interface (Upper Section)



(a) Platform statistics and key metrics dashboard



(b) Recent activities and updates feed for user engagement



(c) Footer section with contact information and additional resources

Figure 5.2 – Main Web Application - Home Page Interface (Lower Section)

user-friendly interfaces.

5.1.3 Information and Support Pages

These pages provide essential information about the platform, team, and support resources. They are designed to build trust and provide comprehensive assistance to users.

5.1.4 Housing Module

The housing module provides comprehensive accommodation solutions, from browsing available properties to posting new listings. It includes detailed property information and advanced search capabilities.

5.1.5 Job and Activities Module

The job and activities module connects users with employment opportunities and community activities. It provides a comprehensive platform for job searching, posting, and application management.

5.1.6 Corporate Partners

The corporators page showcases our strategic partnerships and collaborations with various organizations. These partnerships enhance the platform's capabilities and expand service offerings.

5.1.7 Service Offerings

Our offerings page presents a comprehensive overview of all services and features available on the platform. This includes detailed descriptions of each service module and their benefits.

localhost:5173/login

HOME SERVICES CORPORATES OUR OFFERINGS ABOUT US Log in Sign up

Sign in

Email example@gmail.com

Password

Remember me [Forgot your password?](#)

By continuing, you agree to the [Terms of use](#) and [Privacy Policy](#)

Log in

Don't have an account? [Sign up](#)

Heart icon

(a) User login interface with email/password authentication and social login options

localhost:5173/signup

HOME SERVICES CORPORATES OUR OFFERINGS ABOUT US Log in Sign up

Sign up

Our platform is a multifunctional hub that connects users with seamless transportation, housing, and activity services.

Basic info Contact info Login info

*All fields required unless noted

*First Name

*Last Name

*Email

*What is your gender?

Female

Male

*What is your age?

*Address

*City

Next

Help Privacy Terms

(b) User registration form with comprehensive profile creation fields

Figure 5.3 – Main Web Application - Authentication System

Seamless Mobility, Exciting Activities, and Comfortable Stays - All in One Place!

Welcome to Aoun, your one-stop destination for affordable travel, exploration, and accommodations. Whether you need a place to stay, we are here to make your journey seamless and enjoyable.



What We Offer

- Hassle-free Rides**: Daily local rides within your city or passenger rides.
- Exciting Activities**: Discover and book exciting experiences and activities.
- Comfortable Stays**: Access quality accommodations to suit your needs and preferences.

Our Vision

We aim to create a connected platform where people can travel freely, explore new destinations, and find the perfect stay with a seamless, secure, and user-friendly way.

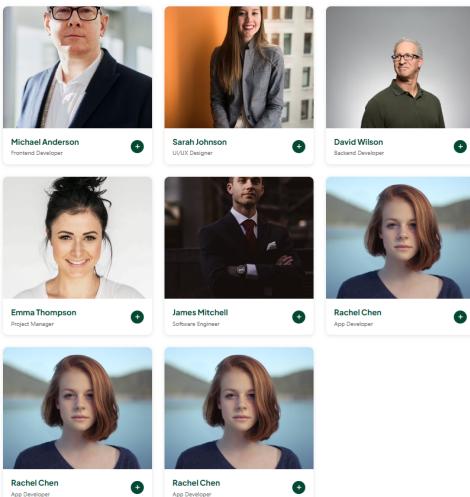
Join Us Today!

Be part of our growing community and experience the future of travel and accommodation services.

[Join Now](#)

Our Team

Home / Team

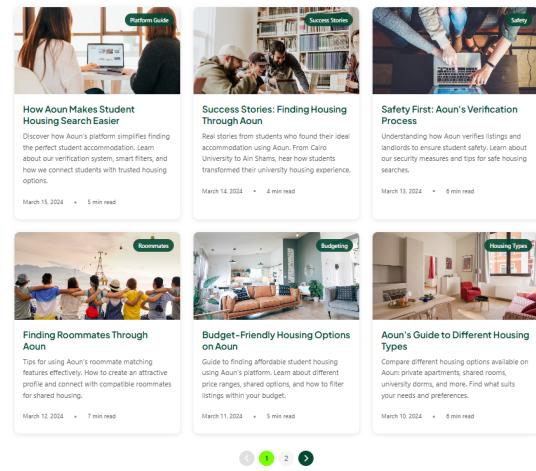


Michael Anderson, Frontend Developer; Sarah Johnson, UI/UX Designer; David Wilson, Backend Developer; Emma Thompson, Project Manager; James Mitchell, Software Engineer; Rachel Chen, App Developer; Rachel Chen, App Developer.

(c) Our Team - Development and management team profiles

Blog

Home / Blog



How Aoun Makes Student Housing Search Easier (March 15, 2024); Success Stories: Finding Housing Through Aoun (March 14, 2024); Safety First: Aoun's Verification Process (March 13, 2024); Finding Roommates Through Aoun (March 12, 2024); Budget-Friendly Housing Options on Aoun (March 11, 2024); Aoun's Guide to Different Housing Types (March 10, 2024).

(e) Blog - Latest updates and industry insights

Student Success Stories

Home / Success Stories

Success Stories

Aoun helped me find the perfect shared apartment near Cairo University. The location was great, and the price was reasonable. I would highly recommend it to anyone looking for a place to live.

Patricia Lopez
Shared Apartment at Cairo University

The experience was great! I am a student at Cairo University and I found the best place to live in Cairo. The location is perfect for my studies.

Karen Mohamed
Cairo Student

Thanks to Aoun, I found affordable housing near my university. The website is user-friendly and provides a wide range of options. I would highly recommend it to anyone looking for a place to live.

Cheer Arafat
Student at Cairo University

As a first-year student, I needed a space that could accommodate my set budget. I found the perfect place on Aoun, and the location is perfect for my studies. The reduced lighting is perfect for my studies.

Rahla Aboel
Resident Student at AOU

(b) Success Stories - User achievements and testimonials

FAQ

Home / FAQ

What is the purpose of a visa?

How long does it take to process a visa application?

Is there an age limit for applying for a visa?

Can I apply for a visa if I have a criminal record?

What happens if my visa application is denied?

How long does it take for you to complete a project?



Aoun is an online aggregator. We do not participate in transactions between our users, they create and manage all expenses related to any independent.

Helpful Links: Home, About Us, News & Events, Find Cheap Housing, Activities & Opportunities, Contact Us.

Corporate business typically refers to large-scale organizations that prioritize efficiency and reliability.

(d) FAQ - Frequently asked questions and answers

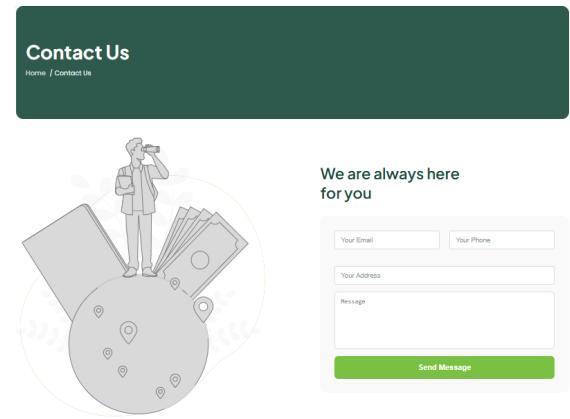
Contact Us

Home / Contact Us

We are always here for you

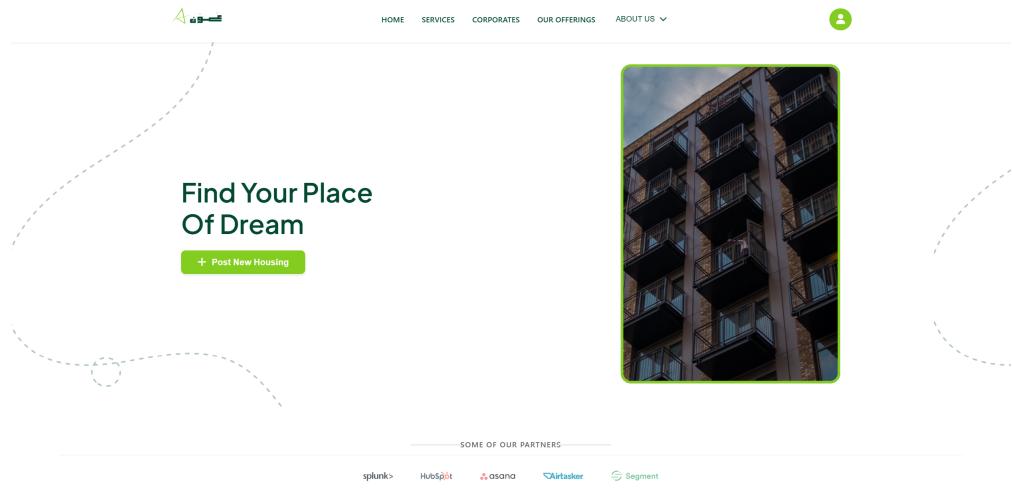
Your Email: _____ Your Phone: _____
 Your Address: _____
 Message: _____

Send Message



(f) Contact - Support and communication channels

Figure 5.4 – Main Web Application - Information and Support Pages



(a) Housing search interface with filter options and property listings

A screenshot of the housing search interface showing a grid of property listings. On the left, there are filter options: "Search by location name" (input field with placeholder "e.g. Beach westpalm"), "Your budget per day" (checkboxes for \$0-\$200, \$200-\$500, \$500-\$1,000, \$1,000-\$2,000, \$2,000-\$5,000), and "Rating" (checkboxes for 1*, 2*, 3*). The main area displays a grid of nine property cards. The first two rows each contain three cards for "House for Rent" (2 rooms, \$2) and the third row contains one card for "Apartment for Rent" (1 room, \$1). Each card includes a small thumbnail image of the interior, the property type, room count, price, and a green circular button with a white arrow pointing right. Below the grid is a set of page navigation buttons numbered 1, 2, 3, and 4.

(b) Featured properties and recommended accommodations

Figure 5.5 – Main Web Application - Housing Module Interface

Housing Details

Housing > Housing Details

Rent - House
No Street, No City

£2,100



Mohamed Sokar Real Estate Specialist [+20 010 1337 9651](#) [COPY PHONE](#)

General Information

ROOM + LIVING 2 + 0
AREA 273 m²

Explanation

Spacious and modern 2-bedroom apartment available in the heart of downtown. This bright and airy unit features an open-concept living space, updated kitchen with stainless steel appliances, and a private balcony with city views. Each bedroom includes ample closet space, and the master bedroom has an en-suite bathroom. The building offers secure entry, underground parking, a fitness center, and a rooftop lounge for residents. Walking distance to shops, restaurants, parks, and public transportation. Perfect for professionals, couples, or small families seeking comfort and convenience.

(a) Property details page with comprehensive information and image gallery

Interior Features ADSL Balcony Barbecue

External Features No external features

Location Information



Aoun is an online aggregator. We do not participate in transactions between our users, they create and manage all requests within our app independently

[Ride](#) [About us](#)
[Drive](#) [Newsroom](#)
[Find Nearby Housing](#) [Blog](#)
[Activities & Opportunities](#) [Our offerings](#)

Enter Email [Submit](#)

© Aoun 2024 | All Rights Reserved [Terms & Condition](#) [Privacy Policy](#) [Help center](#)

(b) Additional property information including amenities and contact details

Figure 5.6 – Main Web Application - Housing Module Property Details

(a) Basic housing information form

Housing Post
Housing / Housingpost

1 Category 2 General Information 3 Location Information 4 Posting Photos 5 Advertise Features

Category
*All fields required unless noted.
*Housing
 Apartment for sale
 Apartment for rent
*Title
Enter title

Next

(b) Detailed property specifications

Housing Post
Housing / Housingpost

1 Category 2 General Information 3 Location Information 4 Posting Photos 5 Advertise Features

Advertise Status
 Sale Rent
Available for Loan
 Suitable Unsuitable
Furnished
 Yes No
Housing Shape
 Apartment Penthouse Villa Chalet
 Duplex Triplex Loft Town House
Warning Type
 Natural Gas Central Wood or Coal Solar
 Diesel Electric Underfloor
Front
 North South East West
 Northeast Southwest Southeast Northwest
Floor Location
Select Select Room
Building Age
Select Select Gross
Living Number
Select Select Gross
Rental Income
Select Select Dues
Net M²
Select Select Select
Next

(c) Location and address information

Housing Post
Housing / Housingpost

1 Category 2 General Information 3 Location Information 4 Posting Photos 5 Advertise Features

Location Information
*All fields required unless noted.
*Location


Next

(d) Property image upload interface

Housing Post
Housing / Housingpost

1 Category 2 General Information 3 Location Information 4 Posting Photos 5 Advertise Features

Posting Photos
*All fields required unless noted.
You can add 10 photos to your ad
Download From Computer

You can add 10 photos to your ad

Next

(e) Internal and external features selection

Housing Post
Housing / Housingpost

1 Category 2 General Information 3 Location Information 4 Posting Photos 5 Advertise Features

Advertise Features
*All fields required unless noted.

Interior Features

- ADSL
- Balcony
- Barbecue
- Laundry Room
- Wallpaper
- Video Intercom
- Shower
- Laminate
- Marine Floor
- Sauri
- Parquet
- Satin Color
- Spotlight
- Terrace
- Underfloor Heating

- Alarm
- Built-in Kitchen
- Furnished
- Air Conditioning
- Dressing Room
- Double Glazing
- TV Satellite
- Double Door
- Bricks
- Parent Bathroom
- Ceramic Floor
- Fireplace
- Cloakroom
- Double Glazing

External Features

- Elevator
- Fitness
- Thermal insulation
- Swimming Pool
- Playground
- Sauna
- Tennis Court
- Swimming Pool
- Basketball Field

- Gardened
- Security
- Generator
- Car Park
- PVC
- Water Tank
- Fire Escape
- Football Field
- Market

Next

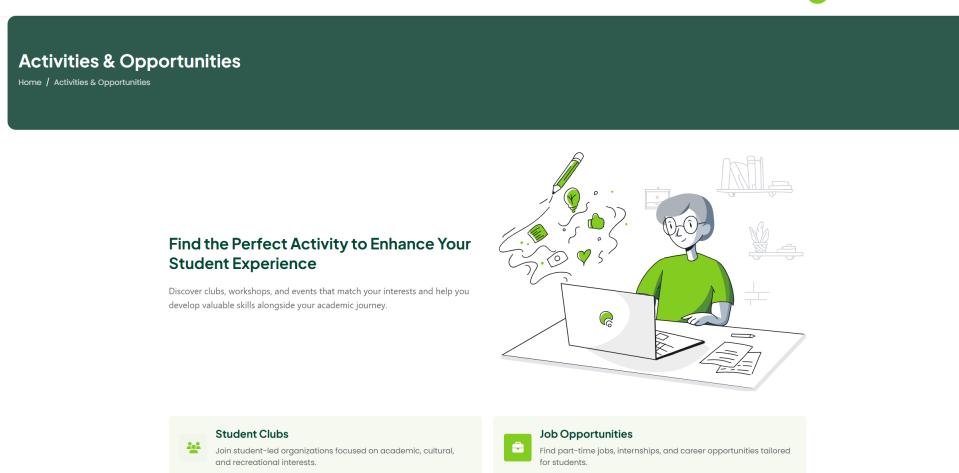
(f) Successful publication confirmation

Housing Post
Housing / Housingpost

1 Category 2 General Information 3 Location Information 4 Posting Photos 5 Advertise Features

Your Advertise has been published successfully.
We Thank you.

Figure 5.7 – Main Web Application - Housing Module Property Posting Process



(a) Job listings overview with search and filter capabilities

Figure 5.8 – Main Web Application - Job Module Interface (Part 1)

Opportunities

Home / Activities & Opportunities / Opportunities

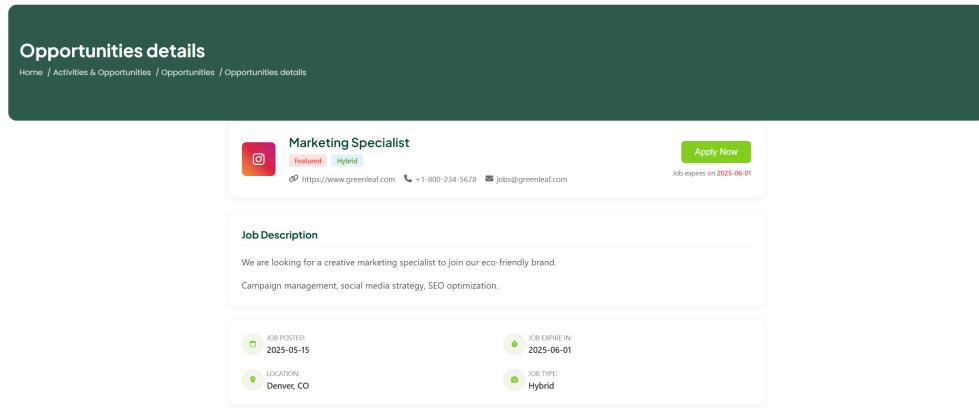
Job title, Keyword... Location Select Category Advance Filter Find Job

Job Listings:

- gg Live El Mansoura, Ad Daqahlyyah, Egypt \$ 2026-06-14 [View Details →](#)
- Software Engineer Live San Francisco, CA \$ 2025-06-15 [View Details →](#)
- Marketing Specialist Live Denver, CO \$ 2025-06-01 [View Details →](#)
- Data Analyst Live New York, NY \$ 2025-06-10 [View Details →](#)
- UX/UI Designer Live Los Angeles, CA \$ 2025-06-20 [View Details →](#)

(a) Featured job opportunities and recommended positions

Figure 5.9 – Main Web Application - Job Module Interface (Part 2)



(a) Detailed job description with requirements and application process

Figure 5.10 – Main Web Application - Job Module Details Page

5.2 Transportation Web Application

The transportation web application is a specialized platform dedicated exclusively to transportation services. This focused application provides comprehensive travel solutions, route planning, and booking capabilities without the complexity of housing or job modules.

The screenshot shows a web browser window with the URL `localhost:5173/Post`. The page title is "Job Overview". At the top, there is a navigation bar with links for HOME, SERVICES, CORPORATES, OUR OFFERINGS, ABOUT US, and a user profile icon. Below the navigation is a breadcrumb trail: "General Information" → "Job Overview". The main content area contains two sections: "General Information" and "Job Requirements". The "General Information" section includes fields for "Job Title" (with placeholder "Enter job title") and "Job Description" (with placeholder "Describe the job position, responsibilities, and benefits"). The "Job Requirements" section includes a field for "Job Requirements" (with placeholder "List skills, qualifications, and experience needed"). A "Next" button is located at the bottom right of the form.

(a) Job posting form with basic information and requirements

The screenshot shows a continuation of the job posting process. The URL is still `localhost:5173/Post`. The page title is "Job Overview". The navigation bar and breadcrumb trail are identical to the previous screenshot. The main content area now includes sections for "Category" and "Job Overview". The "Job Overview" section contains fields for "Salary" (placeholder "Enter salary range (e.g., \$30K-\$40K)"), "Location" (placeholder "Enter job location"), and "Experience" (a dropdown menu with placeholder "Select experience level"). Below these are sections for "Job type" (radio buttons for "Part time", "Full time", and "Remote"), "Education Level" (radio buttons for "Undergraduate" and "Graduate"), and "Job expires in:" (a dropdown menu with three fields: Month, Day, and Year). At the bottom are "Back" and "Create Job" buttons.

(b) Additional job details and posting confirmation

Figure 5.11 – Main Web Application - Job Module Posting Process

Empower Your Business with Aoun

Streamline your corporate transportation and accommodation needs with our comprehensive solutions. We provide reliable, efficient, and cost-effective services tailored to your business requirements.



Business Travel

Seamless transportation solutions for your employees' business trips, ensuring punctuality and comfort throughout their journey.



Secure Platform

Enterprise-grade security with verified drivers and properties, ensuring the safety of your employees and assets.



Efficient Management

Centralized booking and management system with detailed reporting and analytics for better resource allocation.



24/7 Support

Dedicated corporate support team available round the clock to assist with any queries or emergencies.

Why Choose Aoun for Your Business?

Cost-Effective

Optimize your travel budget with our competitive pricing and flexible payment options.

Time-Saving

Automated booking and management processes to save valuable time for your team.

Customizable

Tailored solutions that adapt to your specific business needs and requirements.

Reliable Service

Consistent and dependable service delivery across all our transportation and accommodation solutions.

Ready to Transform Your Business Travel?

Contact our corporate team today to discuss your requirements and get started.

[Contact Us](#)

(a) Corporate partners and strategic alliances supporting the platform's growth and service delivery

Figure 5.12 – Main Web Application - Corporate Partners and Strategic Alliances

Comprehensive Solutions for Every Need

Discover our wide range of services designed to make your journey easier, whether you're a student, professional, or business. We've got you covered with reliable transportation, comfortable housing, and exciting opportunities.



Transportation Services

Safe and reliable rides for your daily commute, long-distance travel, and special occasions. Choose from shared rides or private transportation.

[Learn More →](#)



Housing Solutions

Find your perfect accommodation near universities and business districts. Verified properties with flexible rental options for students and professionals.

[Learn More →](#)



Educational Support

Access study groups, tutoring sessions, and educational resources. Connect with peers and mentors for academic success.

[Learn More →](#)



Business Services

Corporate solutions for employee transportation and accommodation. Streamlined booking and management for businesses of all sizes.

[Learn More →](#)



Events & Activities

Join workshops, networking events, and social gatherings. Build connections and enhance your skills through various activities.

[Learn More →](#)



Community Building

Connect with like-minded individuals, join clubs, and participate in community initiatives. Create lasting relationships and memories.

[Learn More →](#)

Why Choose Our Services?

Verified & Secure

All our services are thoroughly verified for your safety and peace of mind.

24/7 Support

Round-the-clock customer support to assist you with any queries.

Easy Booking

Simple and intuitive booking process for all our services.

Flexible Options

Customizable solutions to meet your specific needs and preferences.

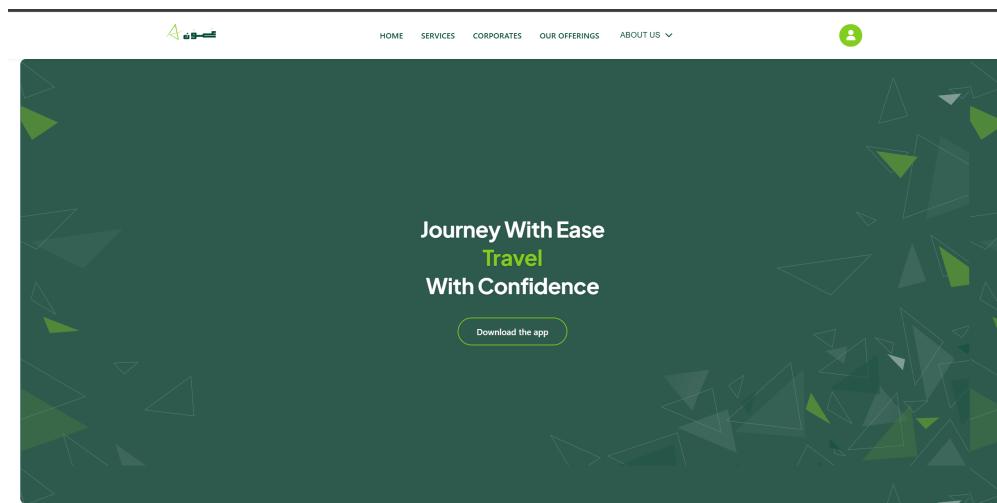
Ready to Get Started?

Join thousands of satisfied users who trust Aoun for their daily needs.

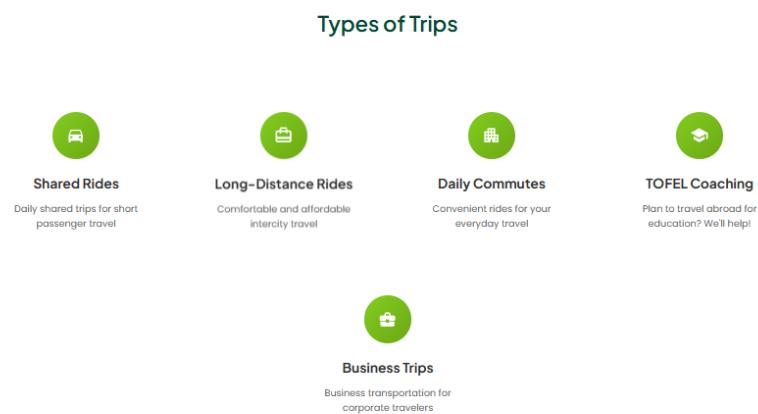
[Sign Up Now](#)

(a) Complete service portfolio including housing, employment, and additional support services

Figure 5.13 – Main Web Application - Platform Service Offerings and Features

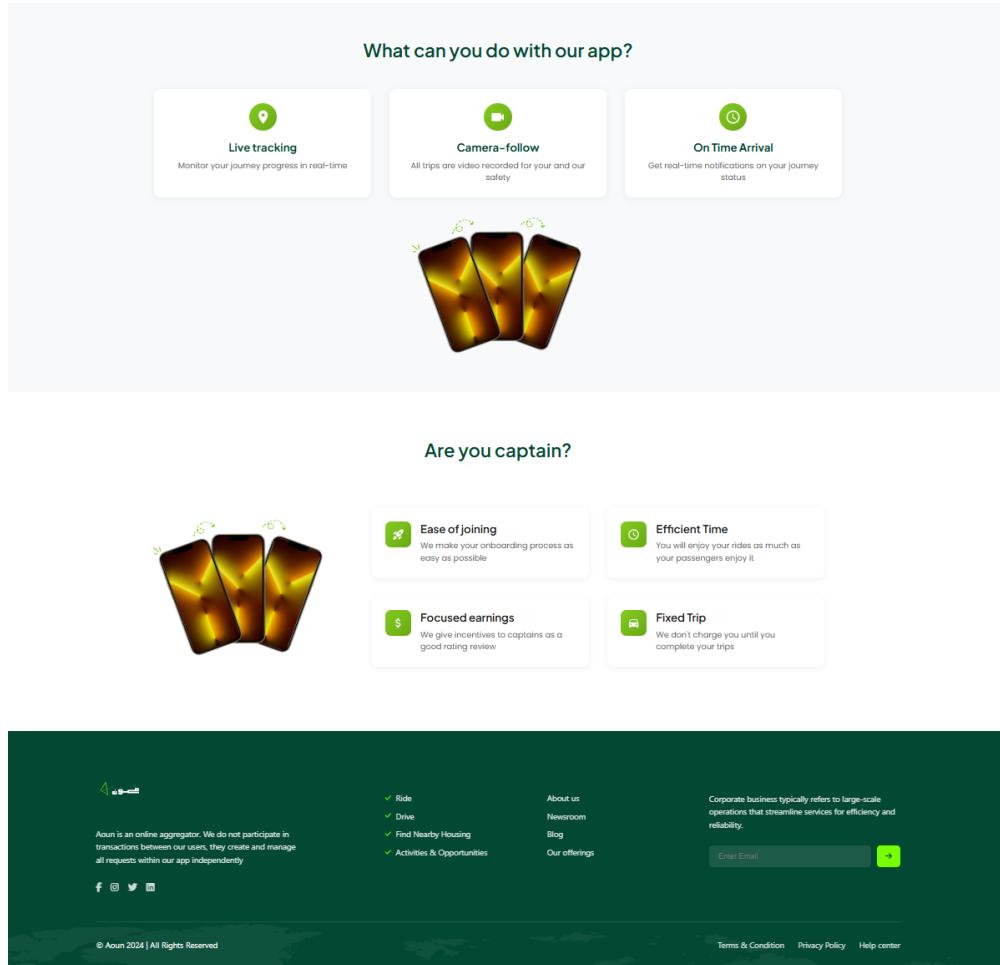


(a) Transportation overview dashboard with available routes and services



(b) Interactive route planning interface with destination selection

Figure 5.14 – Transportation Web Application - Overview and Planning Interface



(a) Booking confirmation and payment processing system

Figure 5.15 – Transportation Web Application - Booking System

5.3 Mobile Application

The mobile application provides comprehensive access to all platform services optimized for smartphone and tablet devices. The mobile version integrates all three service modules (transportation, housing, and jobs) into a unified, touch-friendly interface designed for on-the-go usage.

5.3.1 Mobile Application Features

The mobile application offers several key advantages and features:

5.3.1.1 Core Functionality

- **Unified Service Access:** Complete integration of transportation, housing, and job services in one application
- **Touch-Optimized Interface:** Intuitive navigation designed specifically for mobile devices
- **Location-Based Services:** GPS integration for accurate location detection and proximity-based recommendations
- **Real-Time Notifications:** Push notifications for booking confirmations, job applications, and housing updates
- **Offline Capabilities:** Essential functions available without internet connection

5.3.1.2 Mobile-Specific Optimizations

- **Camera Integration:** Direct photo capture for property listings and profile updates
- **Quick Actions:** Streamlined booking and application processes
- **Gesture Navigation:** Swipe and tap interactions for enhanced user experience
- **Mobile Payment Integration:** Secure payment processing optimized for mobile devices
- **Voice Search:** Audio input capabilities for hands-free searching
- **Biometric Authentication:** Fingerprint and face recognition for secure access

5.3.1.3 Cross-Platform Synchronization

The mobile application maintains seamless synchronization with both web applications:

- **Data Consistency:** Real-time synchronization across all platforms
- **Session Continuity:** Start tasks on one platform and complete on another
- **Unified User Profile:** Single profile accessible across web and mobile platforms
- **Bookmark Synchronization:** Saved searches and favorites available everywhere

5.3.2 Mobile User Experience Design

The mobile application prioritizes user experience through:

5.3.2.1 Interface Design

- **Bottom Navigation:** Easy thumb access to main service modules
- **Card-Based Layout:** Clean, scannable information presentation
- **Progressive Disclosure:** Information revealed as needed to avoid clutter
- **Consistent Visual Language:** Unified design across all service modules

5.3.2.2 Performance Optimization

- **Fast Loading Times:** Optimized images and compressed data transfer
- **Smooth Animations:** 60fps animations for fluid user interactions
- **Efficient Memory Usage:** Optimized for various device specifications
- **Battery Optimization:** Minimal background processing to preserve battery life

5.4 System Integration and Architecture

The multi-platform system demonstrates a comprehensive approach to service delivery through specialized applications:

5.4.1 Application Ecosystem

- **Main Web Application:** Focuses on housing and job services with comprehensive desktop functionality
- **Transportation Web Application:** Specialized transportation booking and management system
- **Mobile Application:** Unified access to all services with mobile-optimized features

5.4.2 Technical Implementation

The system architecture supports:

- **Microservices Architecture:** Independent service modules for scalability
- **API-First Design:** Consistent data access across all platforms
- **Responsive Design:** Optimal viewing across all device types
- **Security Integration:** Unified authentication and authorization systems
- **Real-Time Communication:** Live updates and notifications across platforms

5.4.3 User Benefits

This multi-platform approach provides users with:

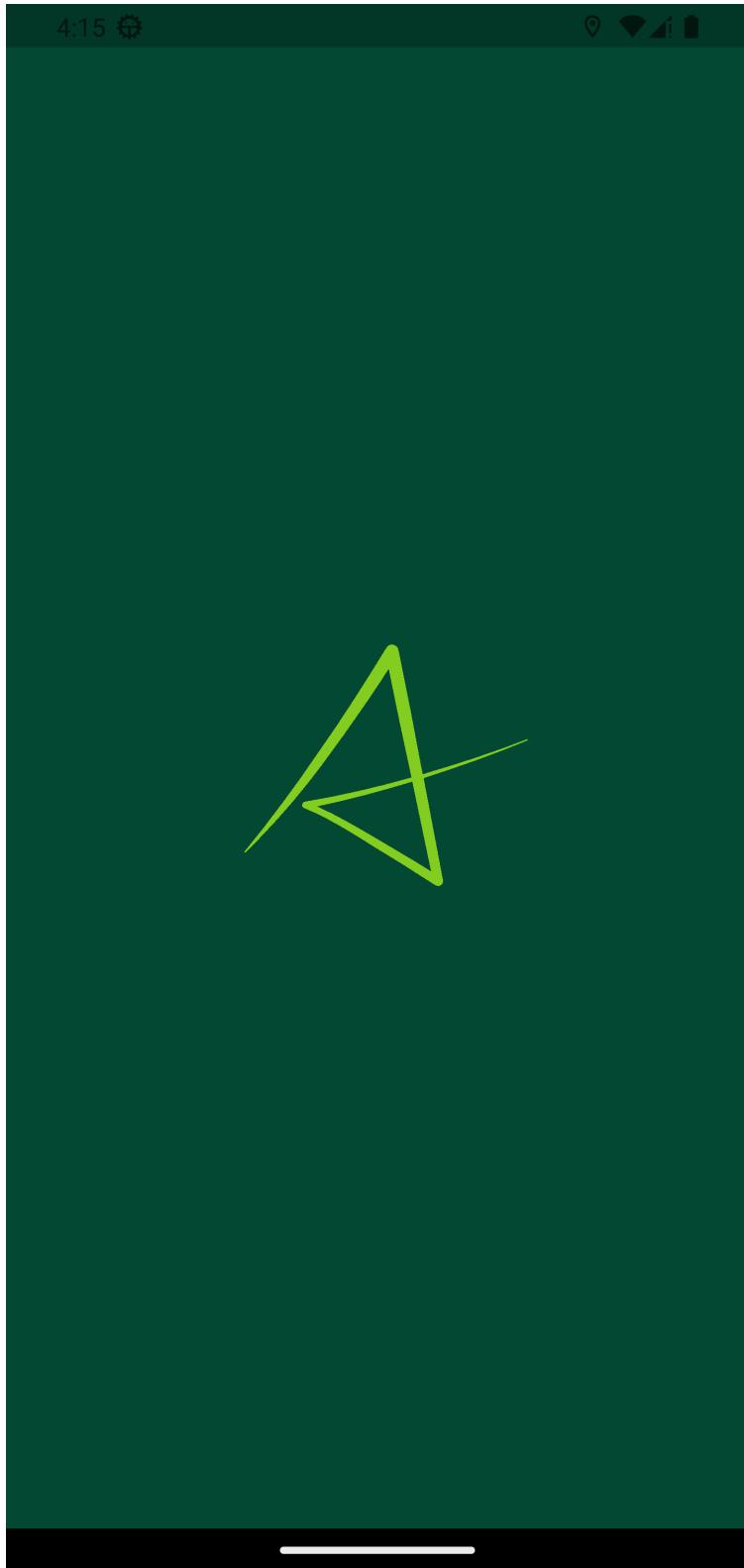
- **Service Specialization:** Focused interfaces for specific needs
- **Platform Flexibility:** Choose the most appropriate platform for each task
- **Comprehensive Coverage:** Access to all services regardless of platform choice
- **Consistent Experience:** Unified branding and functionality across platforms

5.4.4 Mobile Application Authentication System

The mobile application provides a comprehensive authentication system designed specifically for mobile user experience. The authentication flow includes initial onboarding, user type selection, and specialized registration processes for both regular users and drivers.

5.4.4.1 Application Launch and Introduction

The mobile application begins with an engaging onboarding experience that introduces users to the platform's capabilities and guides them through the initial setup process.



(a) Application splash screen with brand identity and loading animation

Figure 5.16 – Mobile Application - Splash Screen



Safe & Easy Rides

Join reliable trips, book your seat, pay securely, and skip commuting hassles.

Next

(a) Welcome screen highlighting platform services

Find Nearby Housing

Discover housing near your university, browse offers, and book securely with ease.

Next

(b) Service overview showcasing key features

Figure 5.17 – Mobile Application - Introduction and Onboarding Screens (Part 1)



• • ●

Activities & Opportunities

Join student activities, attend sessions, or explore job opportunities alongside your studies.

Get Started

(a) Getting started guide and call-to-action

Figure 5.18 – Mobile Application - Introduction and Onboarding Screens (Part 2)

5.4.4.2 User Type Selection

After the introduction, users are presented with options to select their preferred entry point into the application, allowing them to choose between different user types and authentication methods.

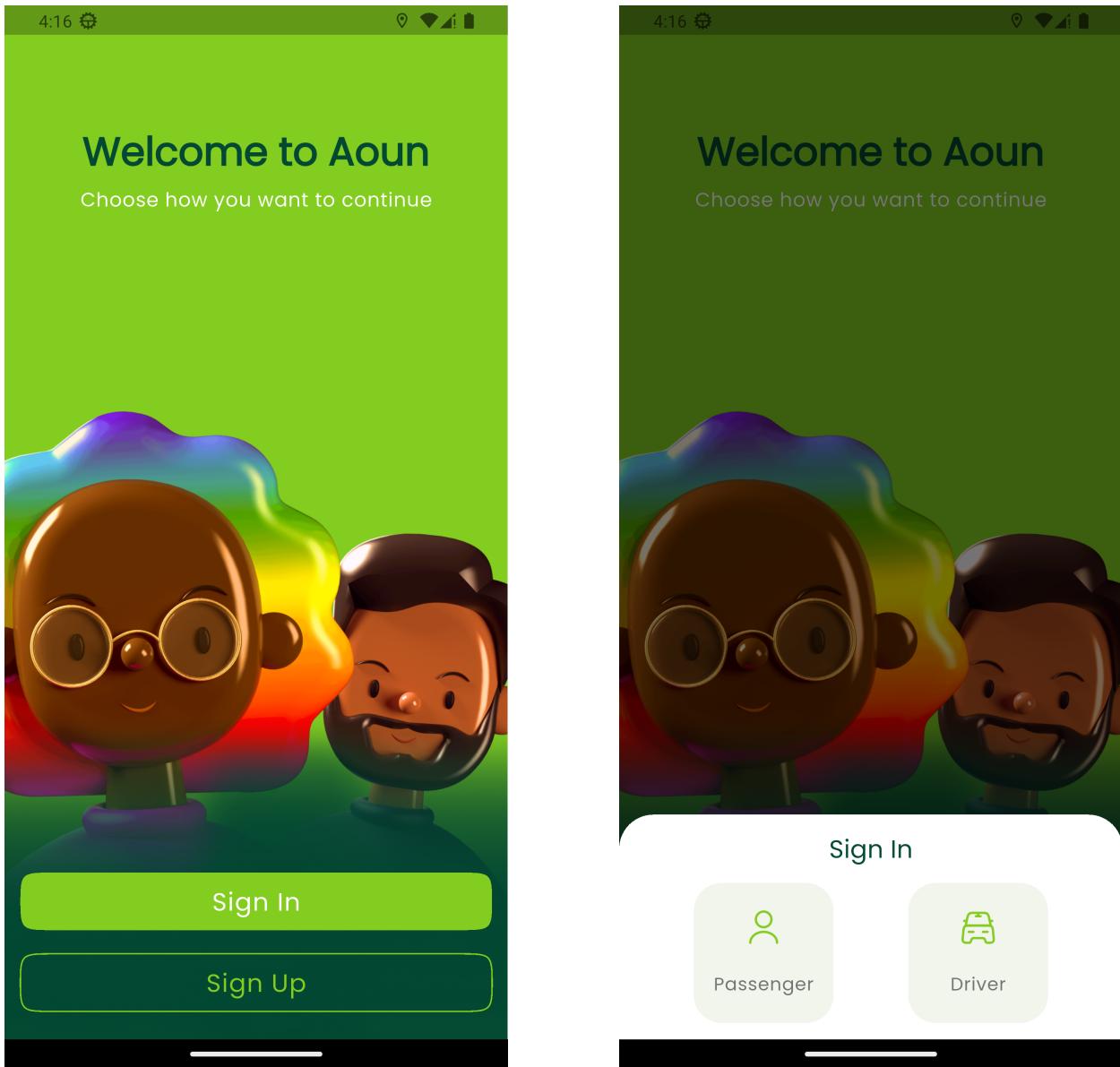
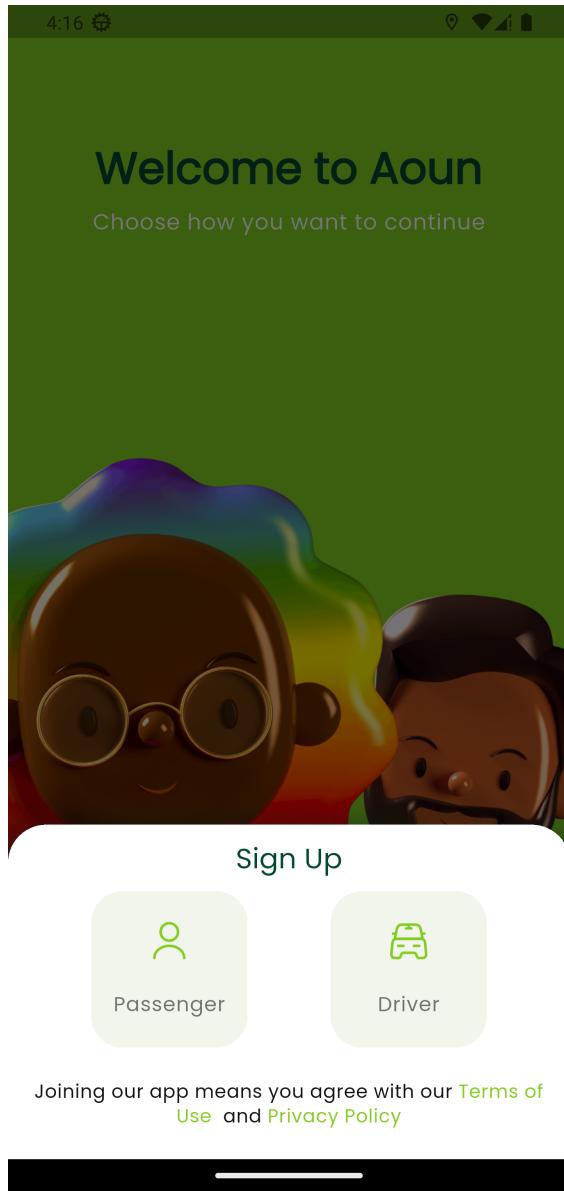


Figure 5.19 – Mobile Application - User Type Selection Interface (Part 1)



(a) Sign up option for new users

Figure 5.20 – Mobile Application - User Type Selection Interface (Part 2)

5.4.4.3 User Authentication Flow

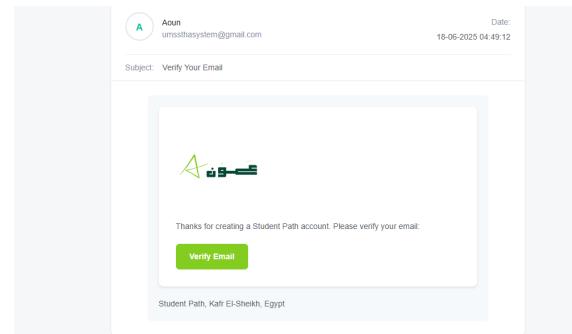
The user authentication system provides a streamlined registration and login process for regular platform users, including email verification and secure password management.

The screenshot shows the 'Get Started' screen of a mobile application. At the top, there is a navigation bar with a back arrow icon. Below it, the title 'Get Started' is displayed in bold green text. A sub-instruction 'Enter your details below' follows. The form consists of several input fields:

- Full Name:** An input field with a person icon and placeholder text.
- Email:** An input field with an envelope icon and placeholder text.
- Gender:** A row with a person icon and two circular radio buttons labeled 'Male' and 'Female'.
- Date of birth:** An input field with a calendar icon and placeholder text.
- Phone Number:** An input field with a phone icon and placeholder text, including a dropdown for country code (+20).
- Password:** An input field with a lock icon and placeholder text.

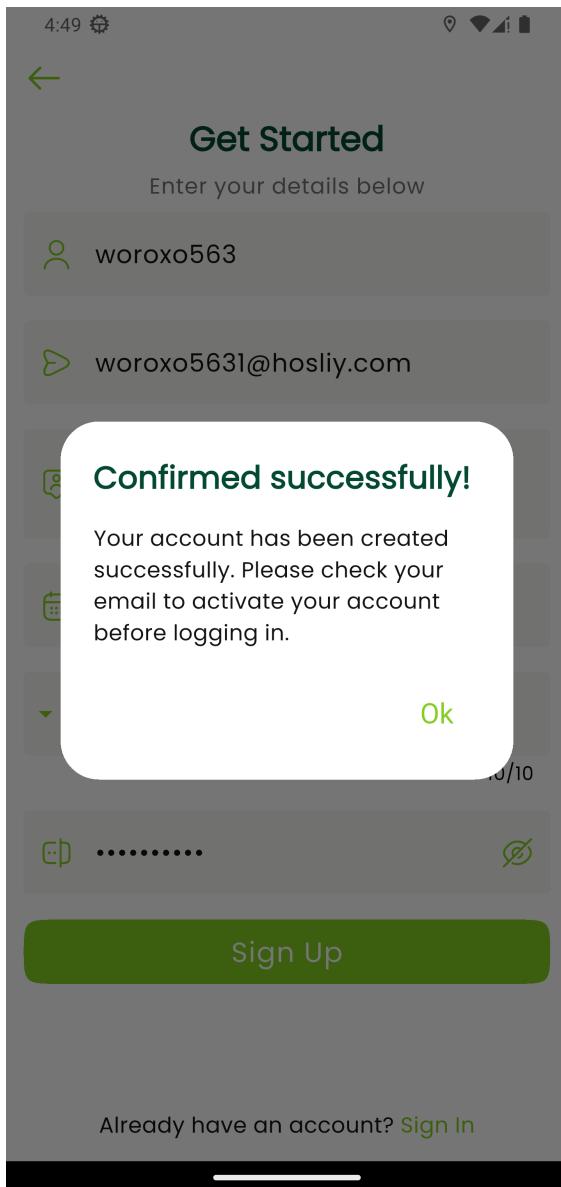
A large green 'Sign Up' button is centered at the bottom of the form. Below the button, a link 'Already have an account? [Sign In](#)' is visible.

(a) User registration form with personal details

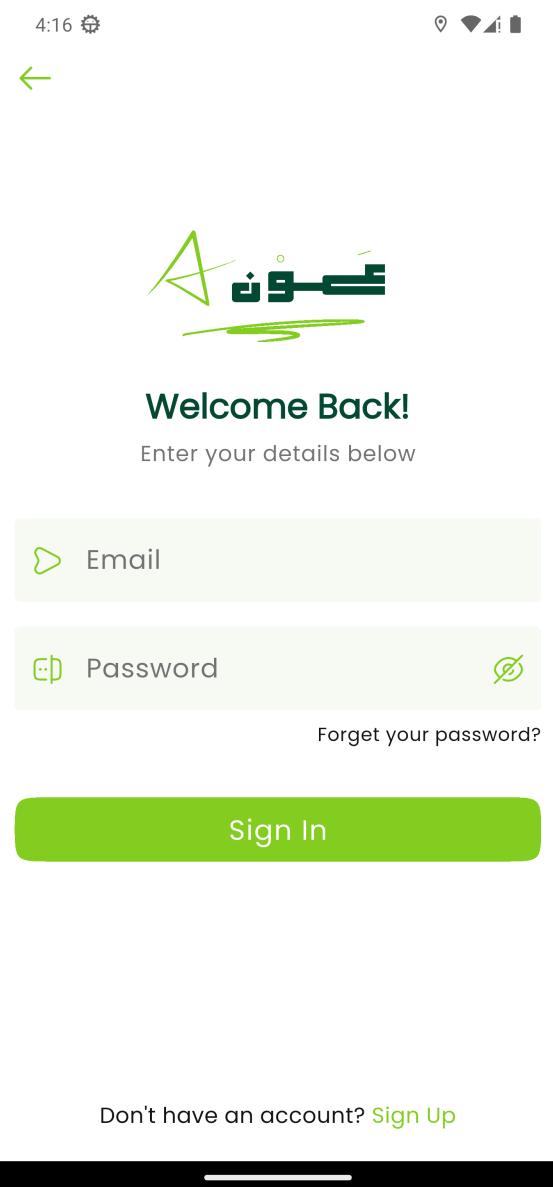


(b) Email confirmation screen - step 1

Figure 5.21 – Mobile Application - User Registration and Email Confirmation

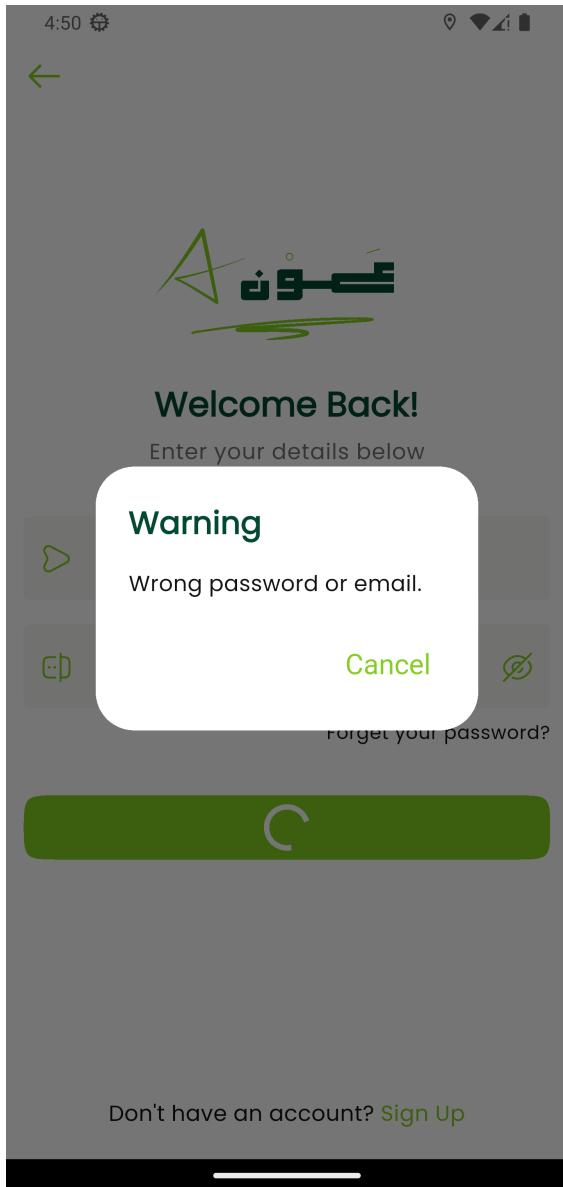


(a) Email confirmation screen - step 2

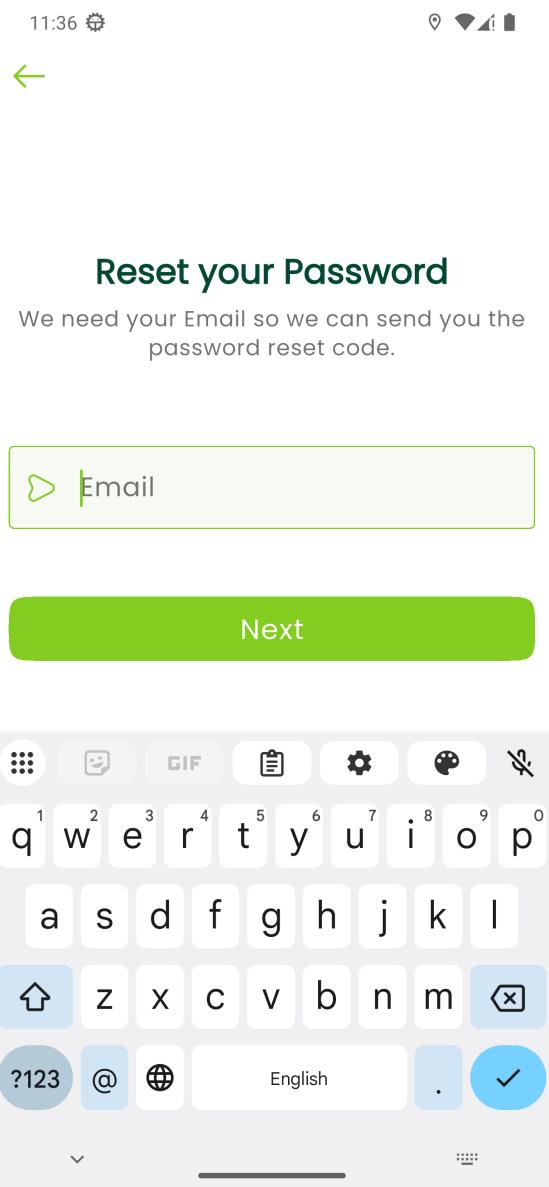


(b) User login interface with credentials

Figure 5.22 – Mobile Application - Email Verification and Login



(a) Error handling for incorrect credentials



(b) Password reset initiation screen

Figure 5.23 – Mobile Application - Error Handling and Password Reset

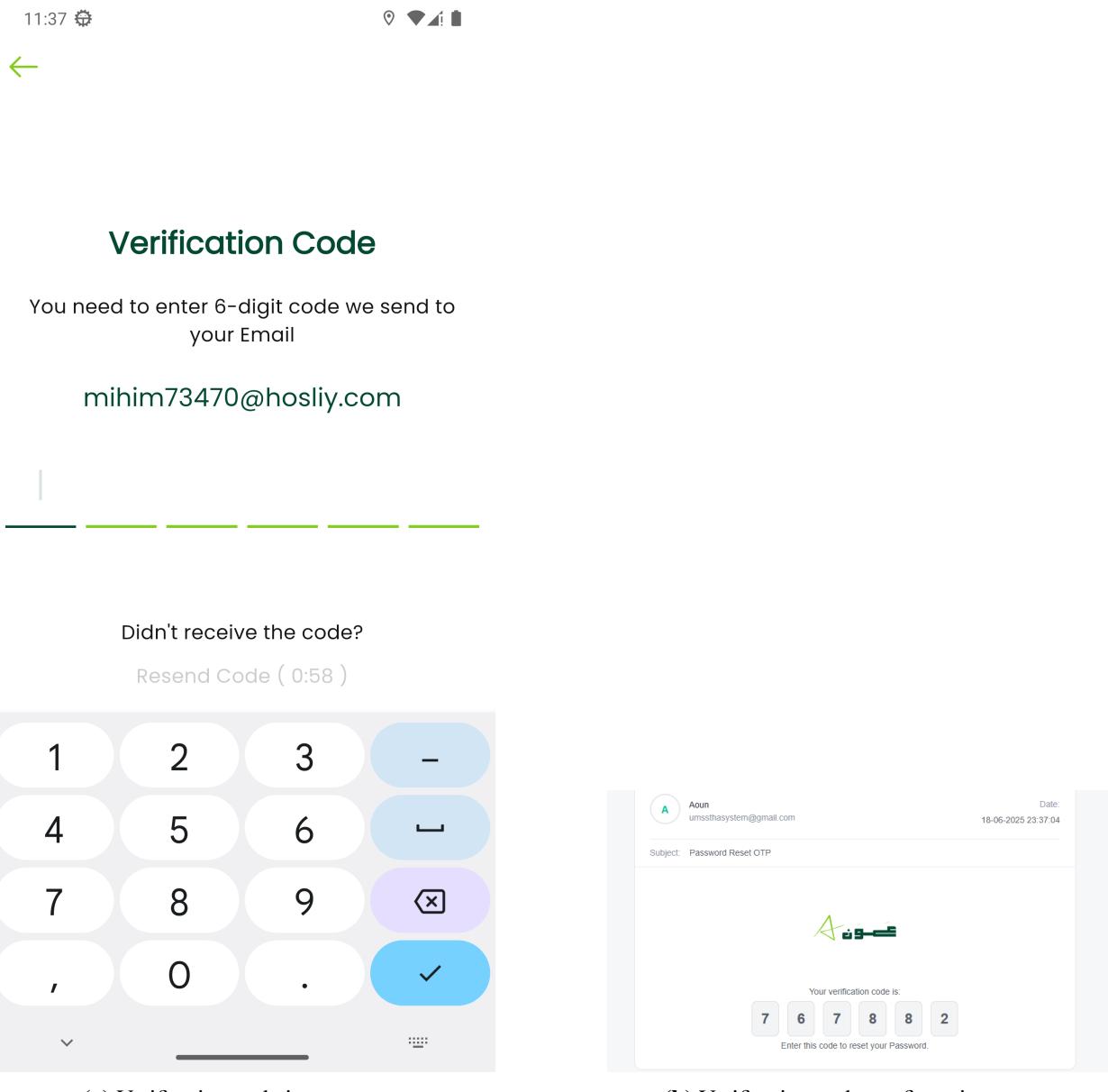
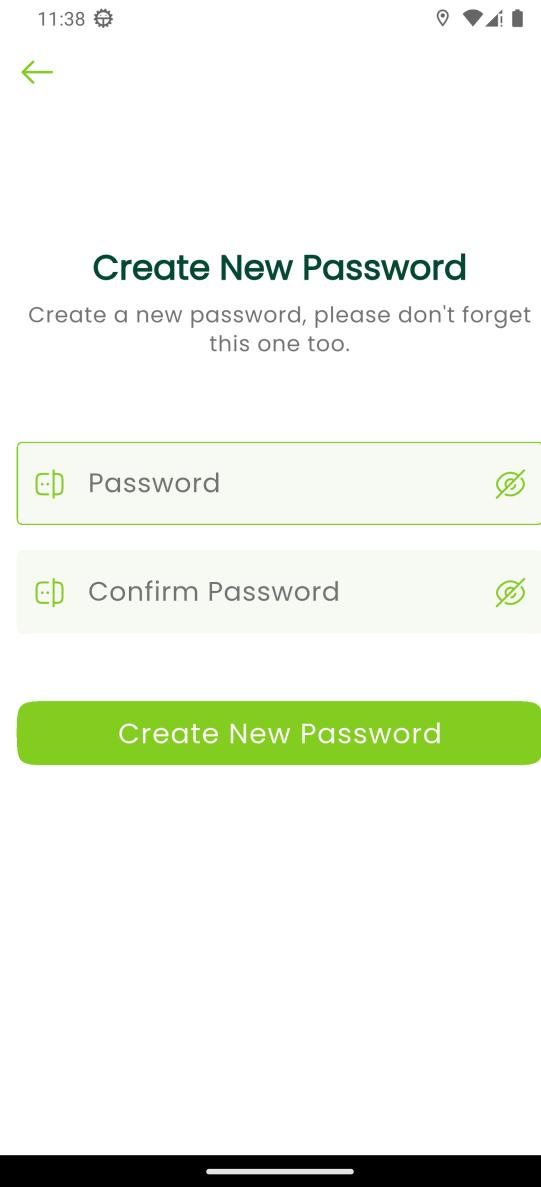


Figure 5.24 – Mobile Application - Verification Code Process



(a) Create new password interface

Figure 5.25 – Mobile Application - New Password Creation

5.4.4.4 Driver Authentication Flow

The driver authentication system includes a comprehensive registration process with personal information collection, vehicle details, document verification, and specialized instructions for driver onboarding.

The figure consists of two side-by-side screenshots of a mobile application interface for creating a driver account. Both screenshots show the top navigation bar with the text "Create Driver account" and a back arrow icon.

(a) Driver personal information - part 1:

- Header: "Personal information"
- Section: "Personal picture" with a placeholder image containing a green plus sign. Below it is the text "Personal picture".
- Text input field: "Full Name" (placeholder text, no value entered).
- Text input field: "Email" (placeholder text, no value entered).
- Gender selection: Two radio buttons labeled "Male" and "Female" (both are empty circles).
- Date input field: "Date of birth" (placeholder text, no value entered).
- Phone number input field: "Phone Number" (placeholder text "+20" followed by a dropdown arrow and "1 of 5").
- Progress bar: A black horizontal bar with a white progress segment indicating "1 of 5".
- Buttons: A green "Next" button with a left arrow icon.

(b) Driver personal information - part 2:

- Header: "Driver License"
- Section: Three placeholder images for driver licenses, each containing a green plus sign. Below them are the labels "Driver License" repeated three times.
- Text input field: "License number" (placeholder text, no value entered). Below it is the text "0/14".
- Text input field: "Expiration date" (placeholder text, no value entered).
- Progress bar: A black horizontal bar with a white progress segment indicating "2 of 5".
- Buttons: A green "Next" button with a left arrow icon.

Figure 5.26 – Mobile Application - Driver Personal Information (Part 1 & 2)

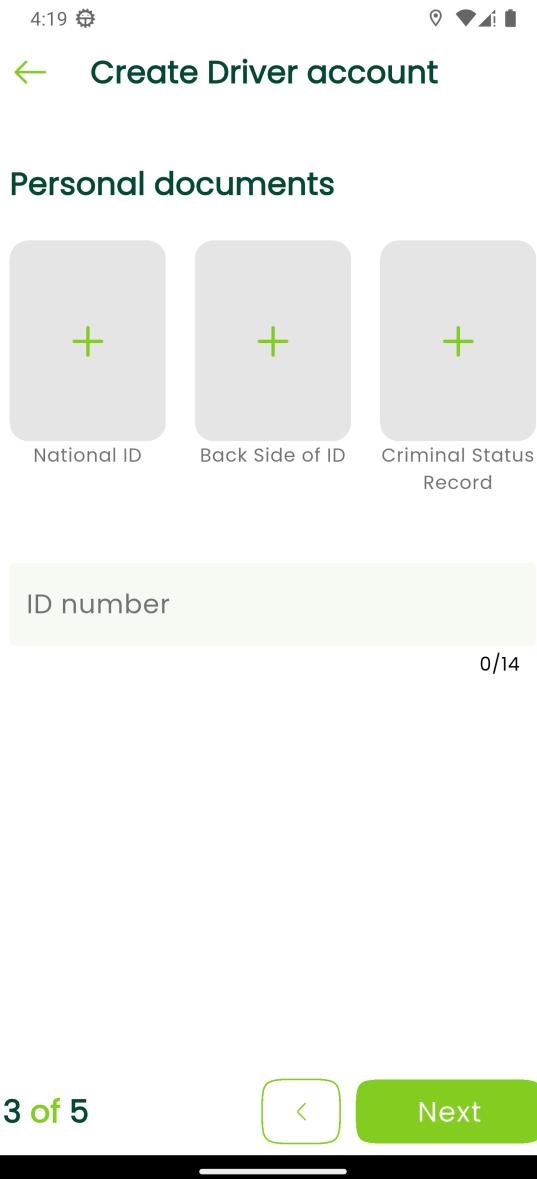
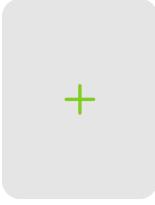


Figure 5.27 – Mobile Application - Driver Personal Information (Part 3)

Create Driver account

Vehicle information



Vehicle picture



Vehicle registration certificate



Back side of certificate

Vehicle brand

Vehicle model

Vehicle color

Production year

Plate number

Number seats

4 of 5

<
Next

(a) Vehicle information and specifications - part 1

Create Driver account



Vehicle picture



Vehicle registration certificate



Back side of certificate

Vehicle brand

Vehicle model

Vehicle color

Production year

Plate number

Number seats

4 of 5

<
Next

(b) Vehicle information and specifications - part 2

Figure 5.28 – Mobile Application - Driver Vehicle Information Registration

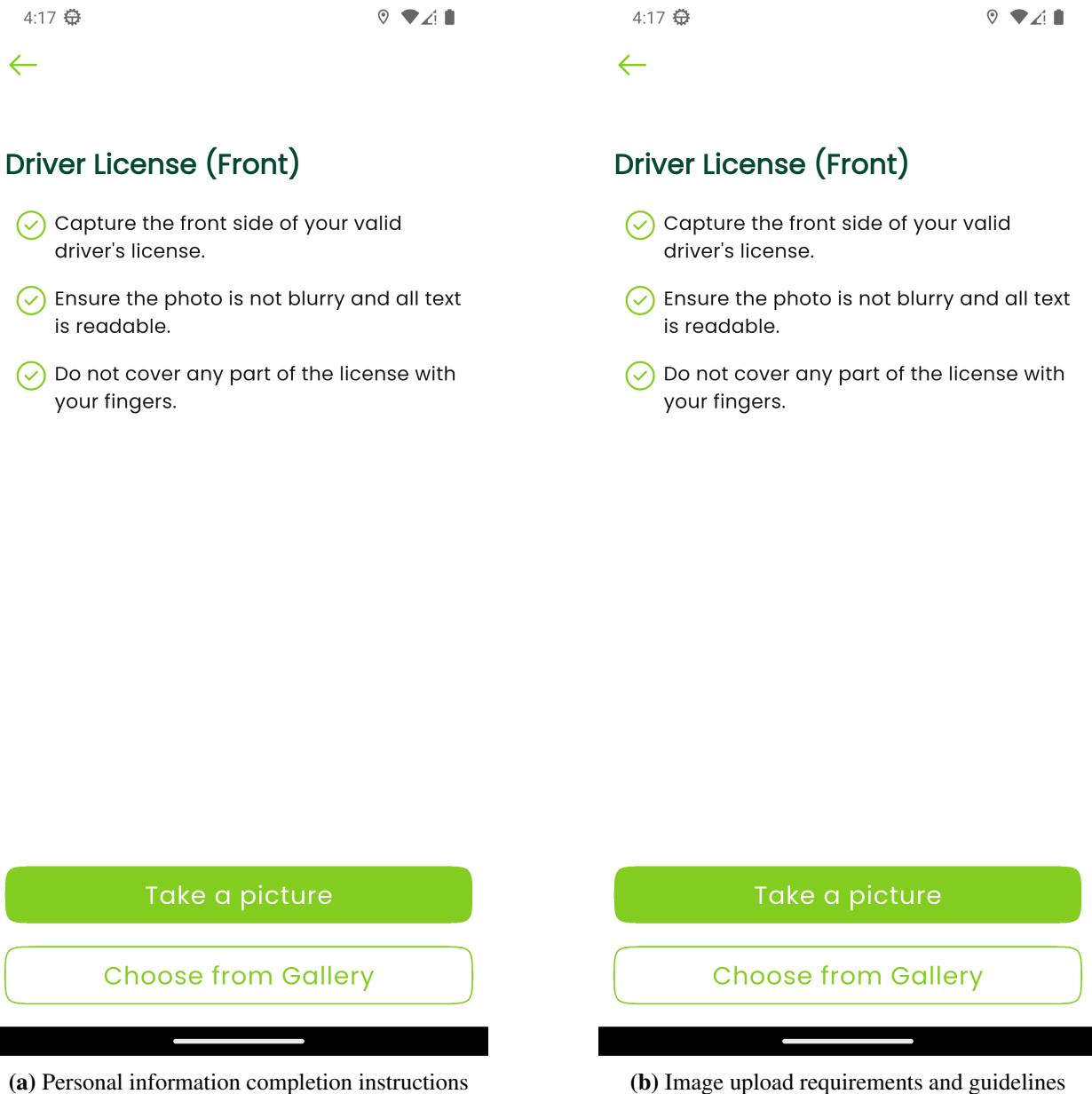


Figure 5.29 – Mobile Application - Driver Registration Instructions

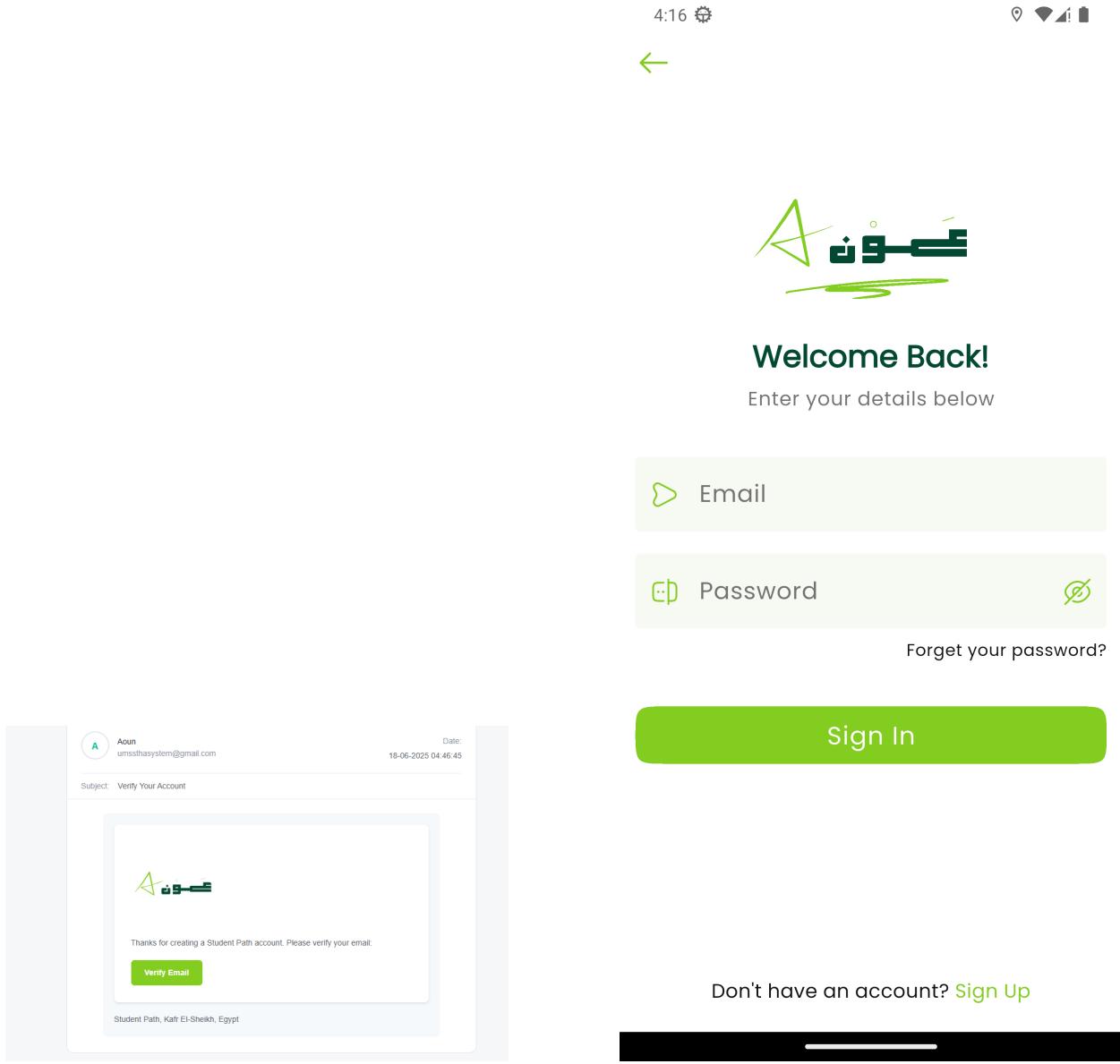


Figure 5.30 – Mobile Application - Driver Confirmation and Login Interface

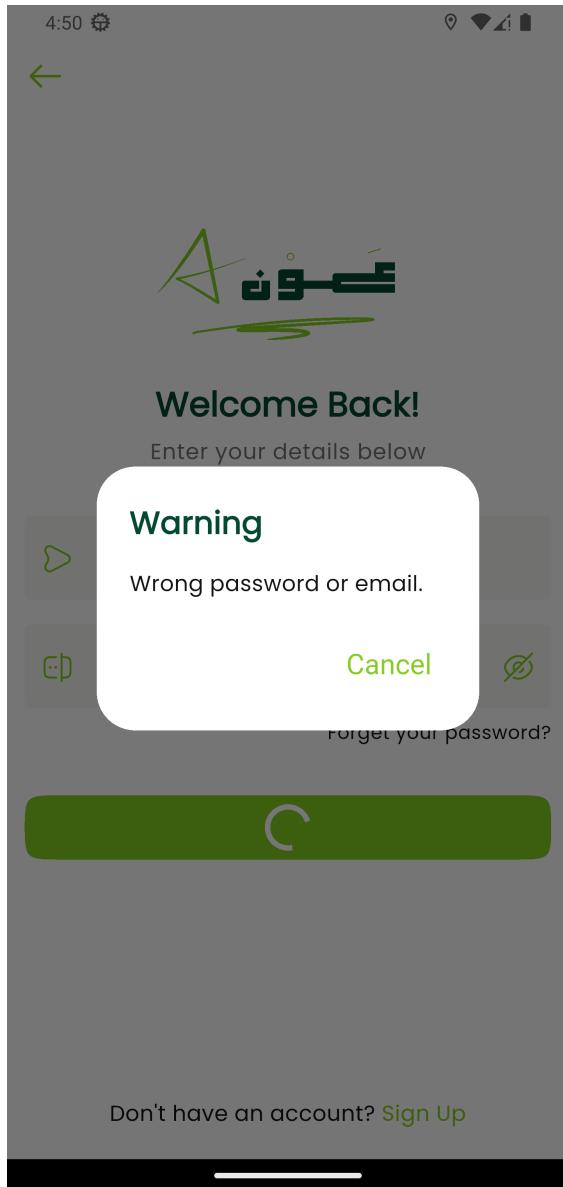


Figure 5.31 – Driver login error handling

5.4.5 Driver Flow Management System

The driver flow management system provides comprehensive functionality for drivers to manage their account settings, create and manage trips, track earnings, and access support services. This system is designed specifically for the unique needs of transportation service providers.

5.4.5.1 Account Settings Management

The account settings section allows drivers to customize their application experience and manage personal preferences including language selection and theme customization.

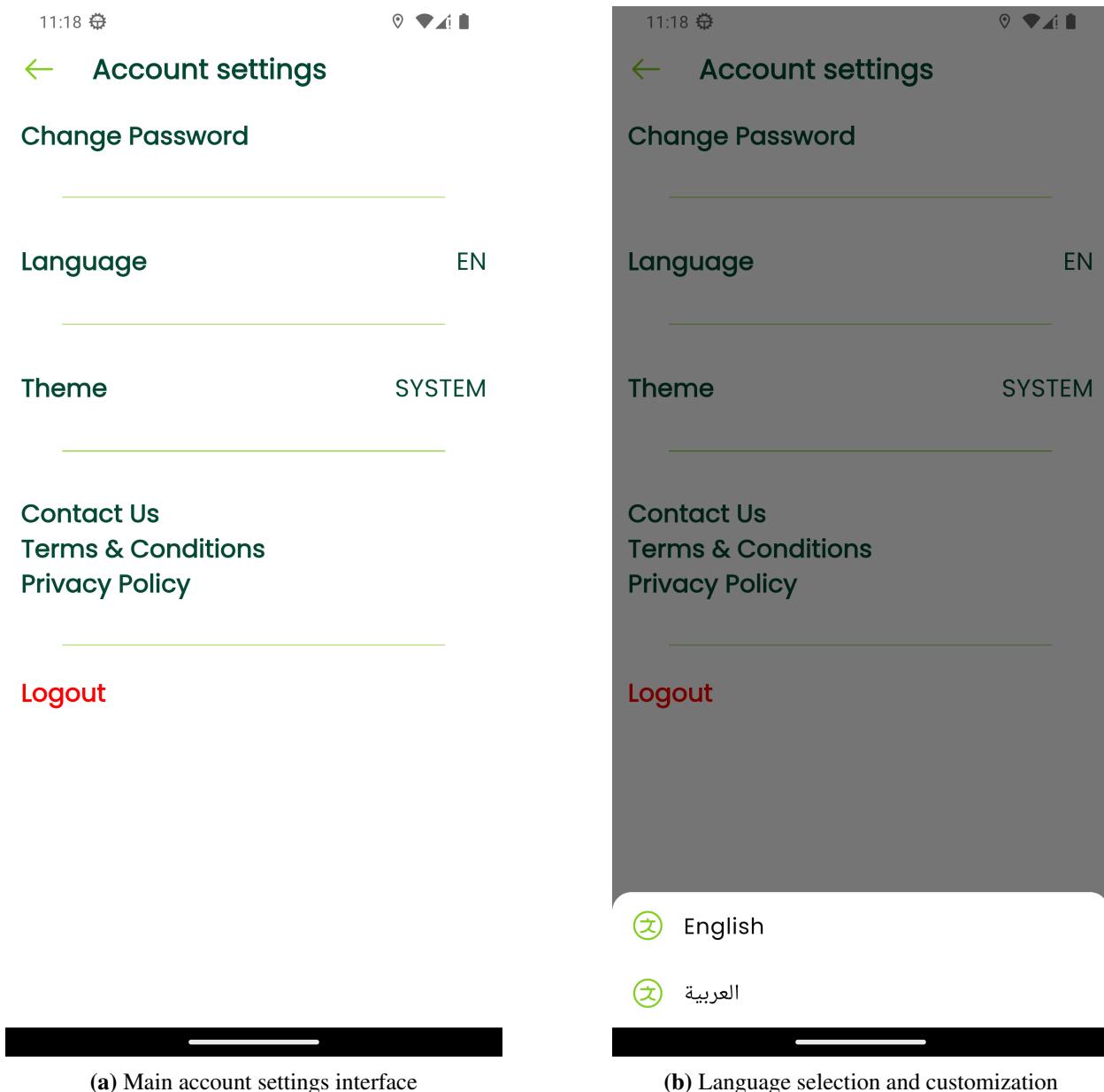
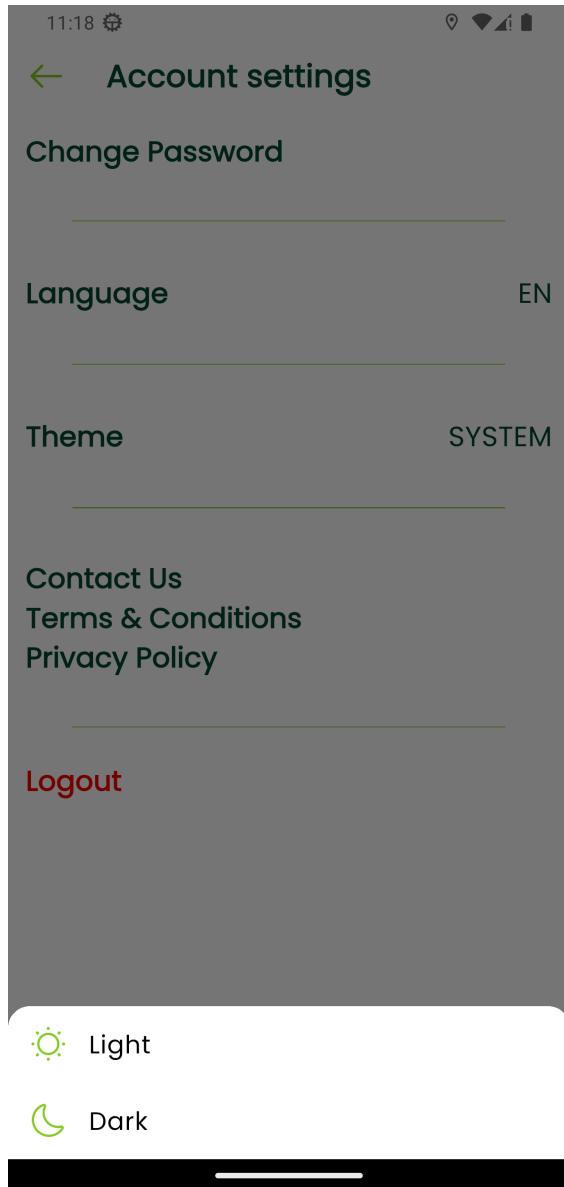


Figure 5.32 – Driver Flow - Account Settings and Language Options



(a) Theme selection and visual customization

Figure 5.33 – Driver Flow - Theme Customization

5.4.5.2 Trip Creation and Management

The trip creation system enables drivers to set up new transportation services, define routes, set pricing, and manage passenger capacity. This comprehensive system includes location selection, route planning, and detailed trip configuration.

(a) Trip creation interface - basic information

From

To

Available Seats

Price of Seat

Departure Time

Driver Notes

Write your notes if you have any.

Amenities

Wi-Fi

Phone Charger

Air Conditioning

Child Seat

Free Water

USB/AUX for Music

(b) Trip creation interface - detailed configuration

Departure Time

Driver Notes

Write your notes if you have any.

Amenities

Wi-Fi

Confirm publish trip

Figure 5.34 – Driver Flow - Trip Creation Process (Part 1 & 2)

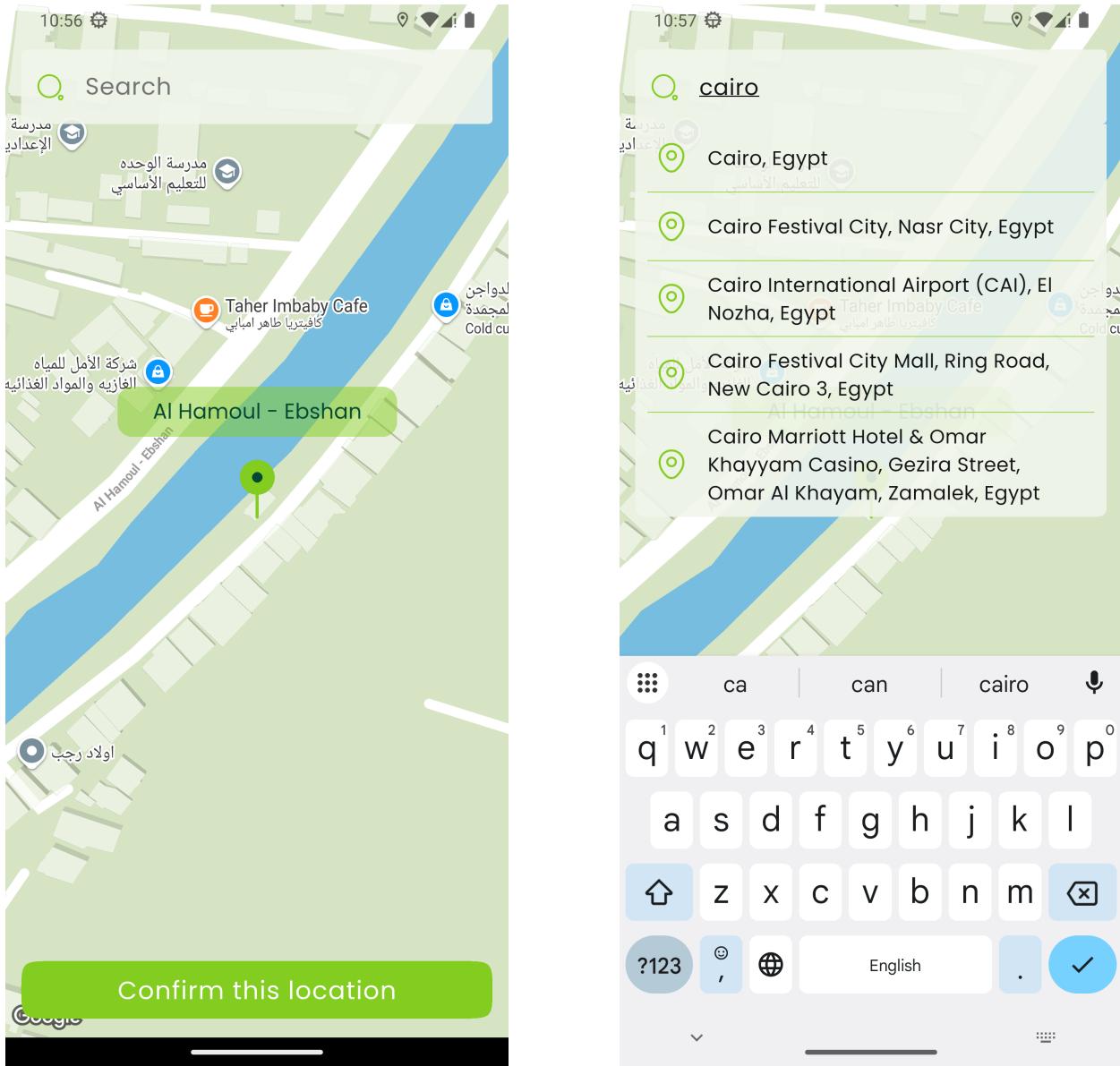
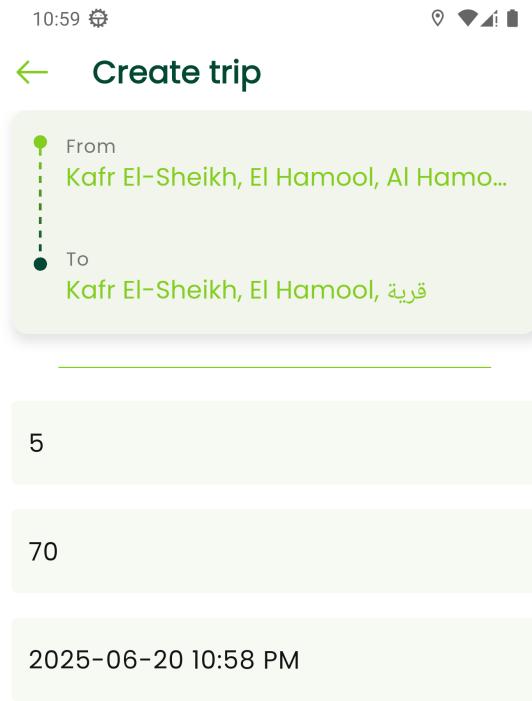


Figure 5.35 – Driver Flow - Location Selection and Search



Driver Notes

A rectangular input field with a light green border and a dark green header bar above it. The header bar contains the text "برجاء التواجد في الموعد المحدد وعدم التاخر" (Please be present at the scheduled time and do not delay).

Amenities

Wi-Fi

—

(a) Example data input and validation

Figure 5.36 – Driver Flow - Data Input Example

5.4.5.3 Created Trip Management

Once trips are created, drivers can monitor trip details, manage bookings, track passenger information, and handle trip completion. This system provides real-time updates and comprehensive trip lifecycle management.

Trip detail

From	Kafr El-Sheikh, El Hamool, Al Ha...
To	Cairo, Abdeen, Moshtohor
<hr/>	
Bookings Passengers	0 from 10
<hr/>	
Status	Planned
Departure Time	Tomorrow 23:21
<hr/>	
Available Seats	10
Status	Planned
Departure Time	Tomorrow 23:21
Available Seats	10
Price Per Seat	\$15.0
Duration	4min
Seating Capacity	16
Created At	2025-06-18 – 08:22 PM
Duration	4min
Seating Capacity	16
Created At	2025-06-18 – 08:22 PM

Trip detail

Bookings Passengers	0 from 10
Status	Planned
Departure Time	Tomorrow 23:21
Available Seats	10
Price Per Seat	\$15.0
Duration	4min
Seating Capacity	16
Created At	2025-06-18 – 08:22 PM

You can cancel the trip anytime up to 1 hour before the departure time. Canceling after that may affect your integrity score.

Canceled trip

(a) Comprehensive trip detail overview
(b) Trip detail with cancellation options

Figure 5.37 – Driver Flow - Trip Detail Management

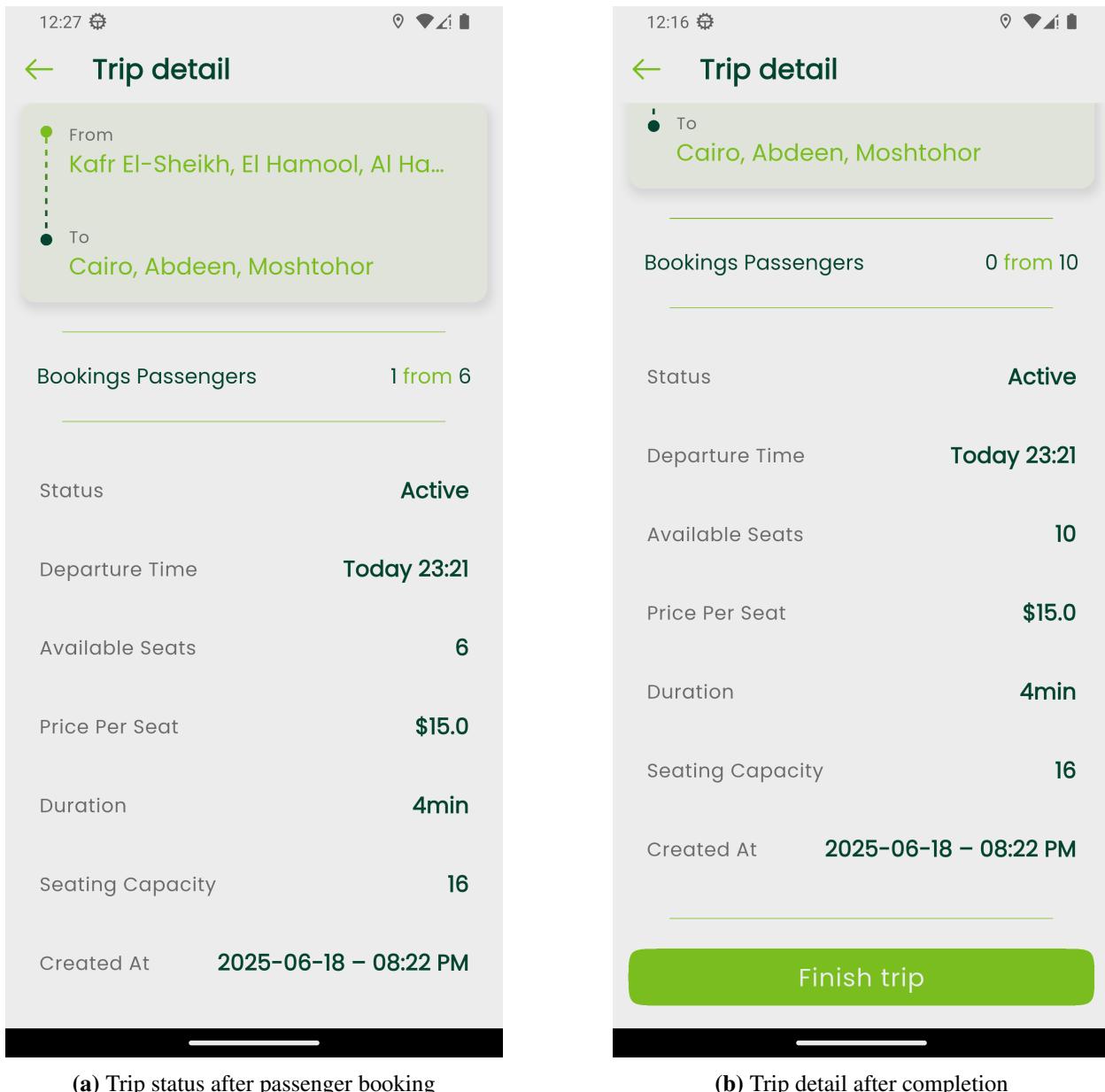
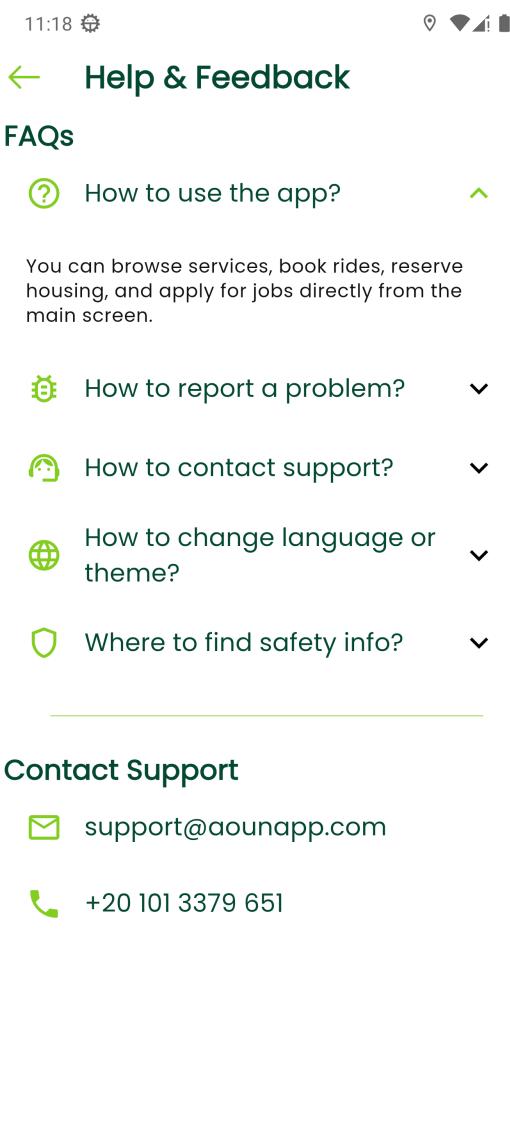


Figure 5.38 – Driver Flow - Trip Booking and Completion Status

5.4.5.4 Support and Feedback System

The support system provides drivers with access to help resources, feedback submission, and customer service contacts for resolving issues and improving service quality.

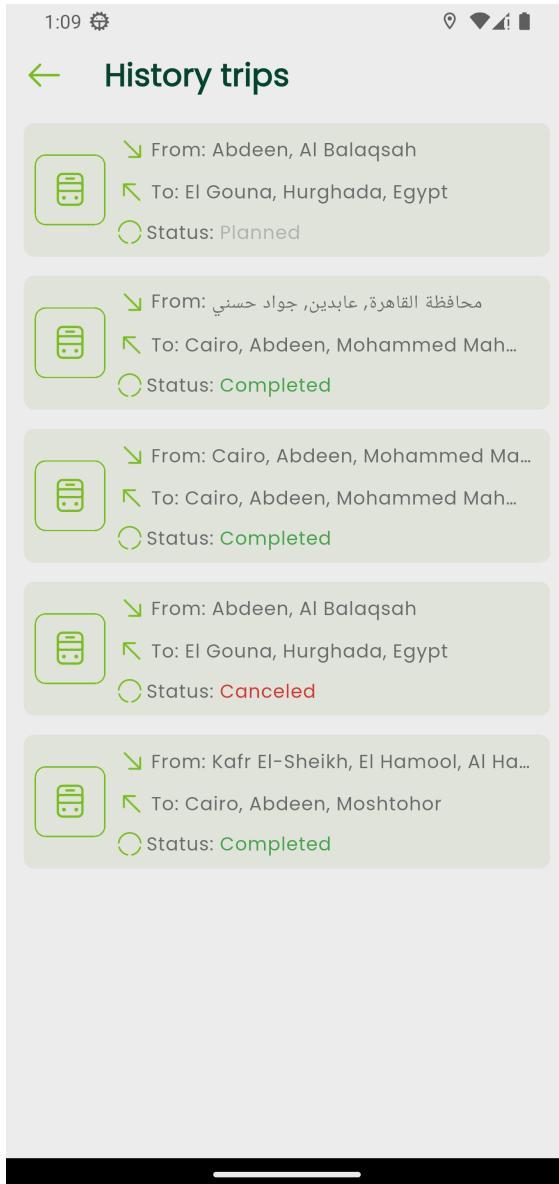


(a) Help and feedback center interface

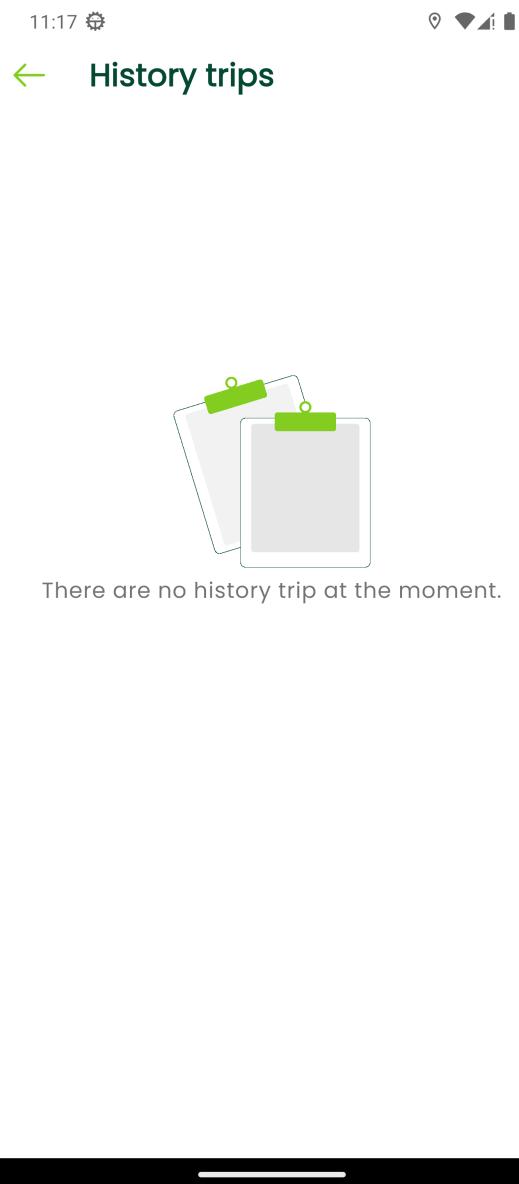
Figure 5.39 – Driver Flow - Help and Feedback System

5.4.5.5 Trip History Management

The trip history system allows drivers to review past trips, track performance metrics, analyze earnings patterns, and access historical data for business planning and tax purposes.



(a) Comprehensive trip history overview



(b) Empty trip history state

Figure 5.40 – Driver Flow - Trip History Management

5.4.5.6 Home Dashboard System

The home dashboard serves as the central hub for drivers, providing quick access to created trips, current bookings, earnings summary, and key performance indicators. The dashboard adapts to show relevant information based on driver activity and trip status.

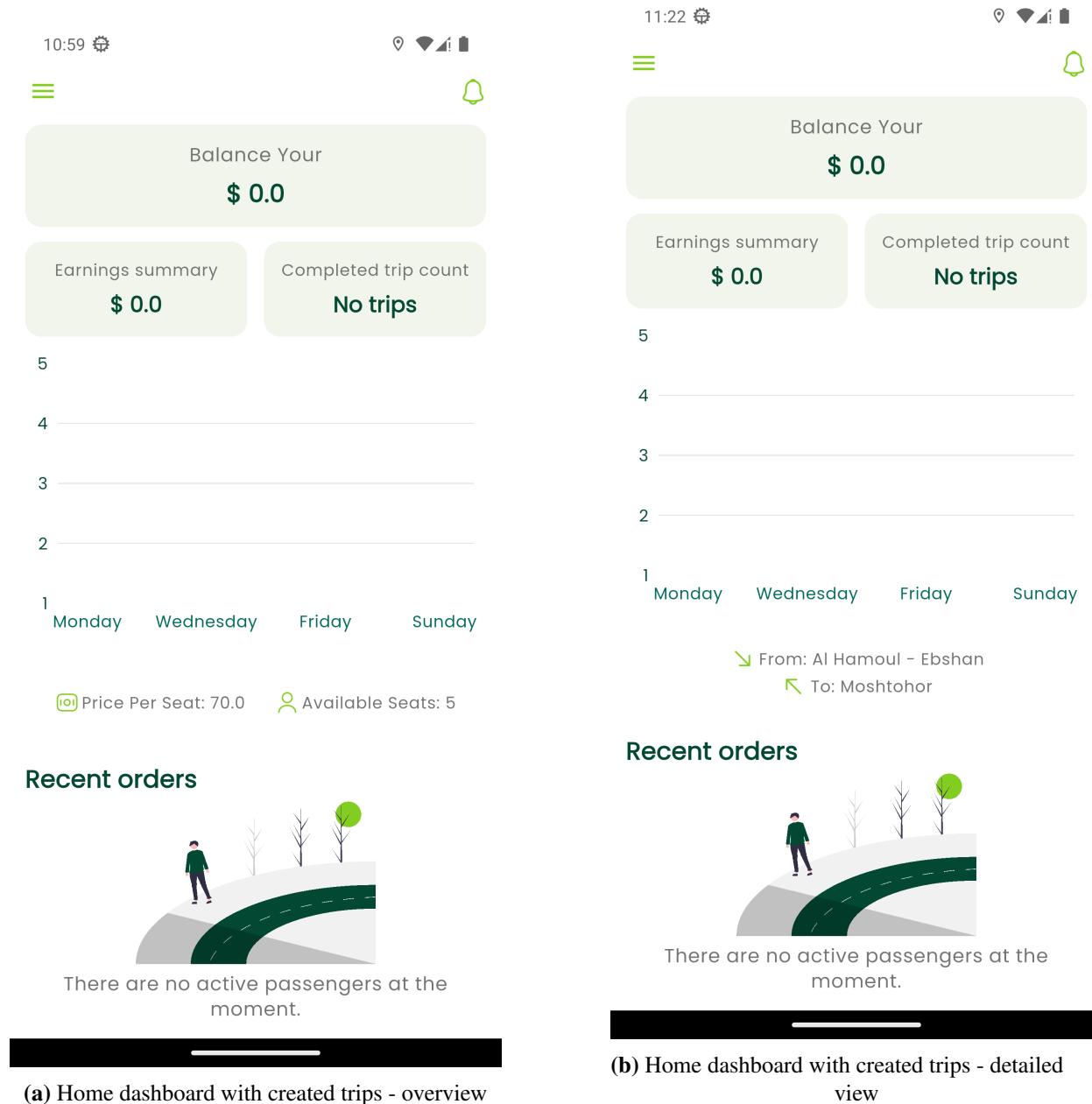


Figure 5.41 – Driver Flow - Home Dashboard with Created Trips (Part 1 & 2)

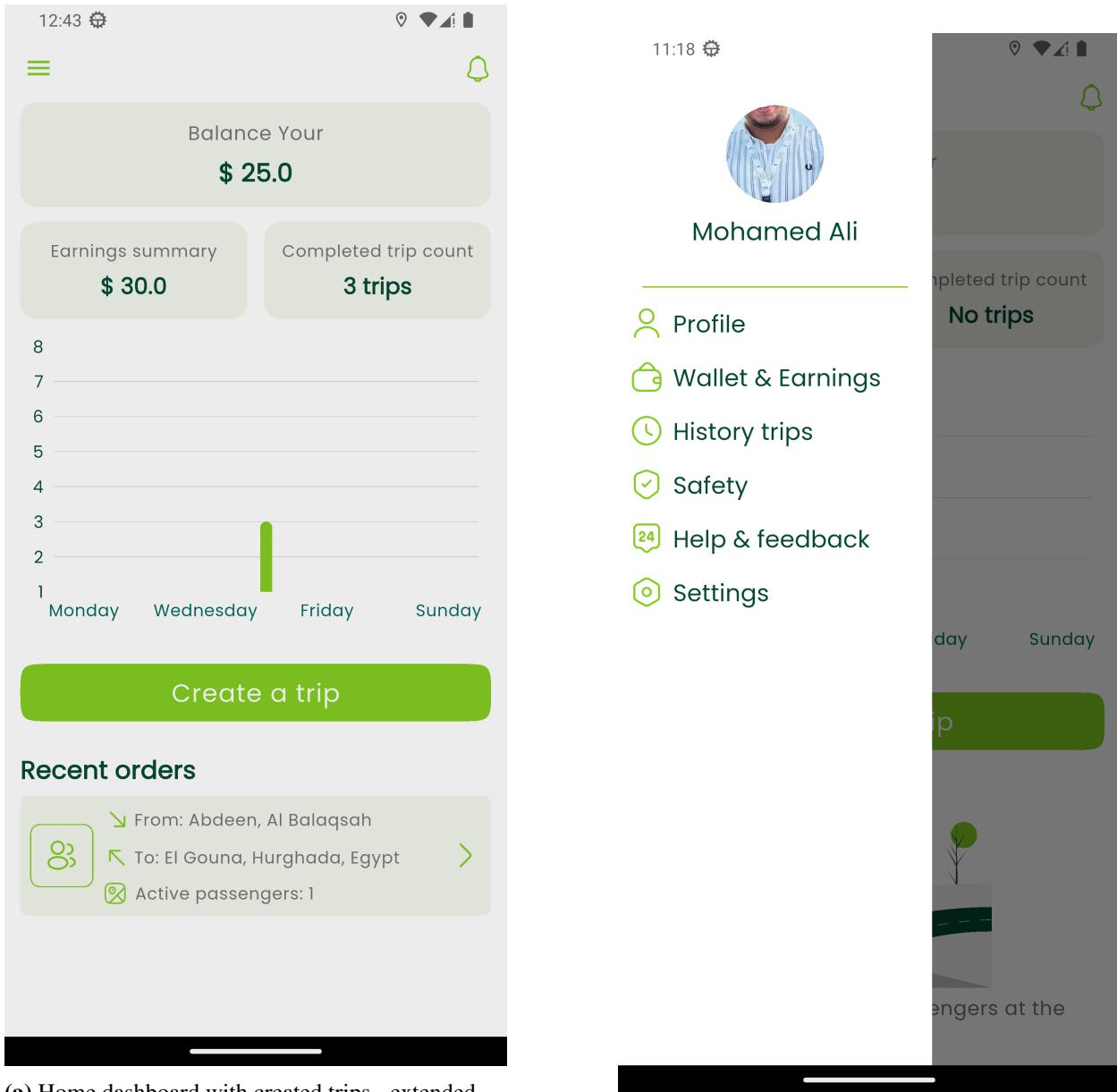


Figure 5.42 – Driver Flow - Home Dashboard Extended View and Navigation

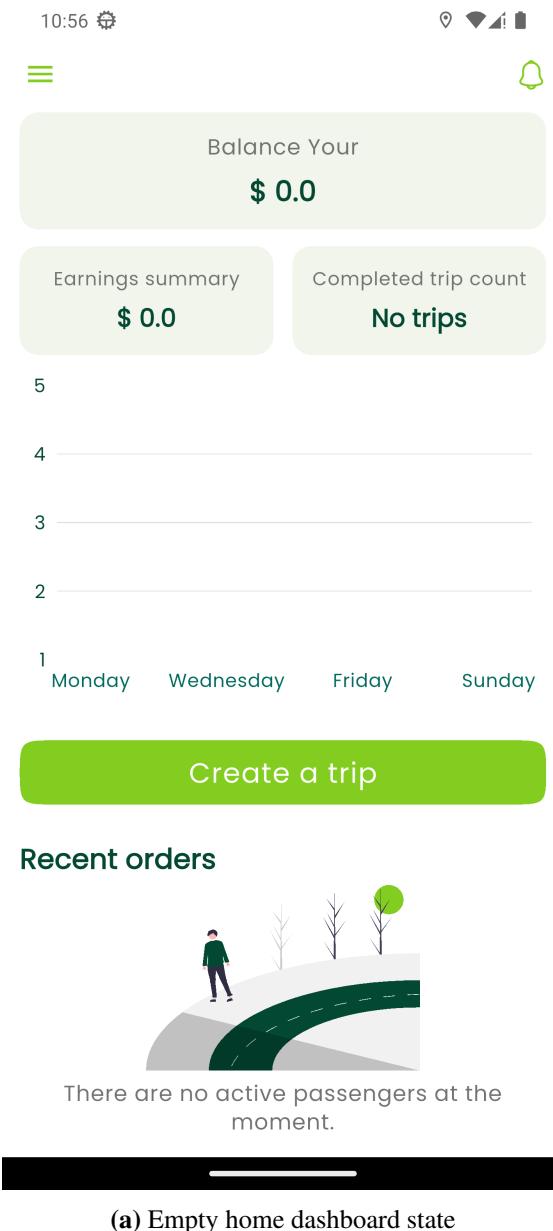


Figure 5.43 – Driver Flow - Empty Home Dashboard State

5.4.5.7 Profile Overview and Management

The profile system allows drivers to manage personal information, vehicle details, and account settings. This comprehensive profile management system ensures that driver information remains current and accurate for optimal service delivery.

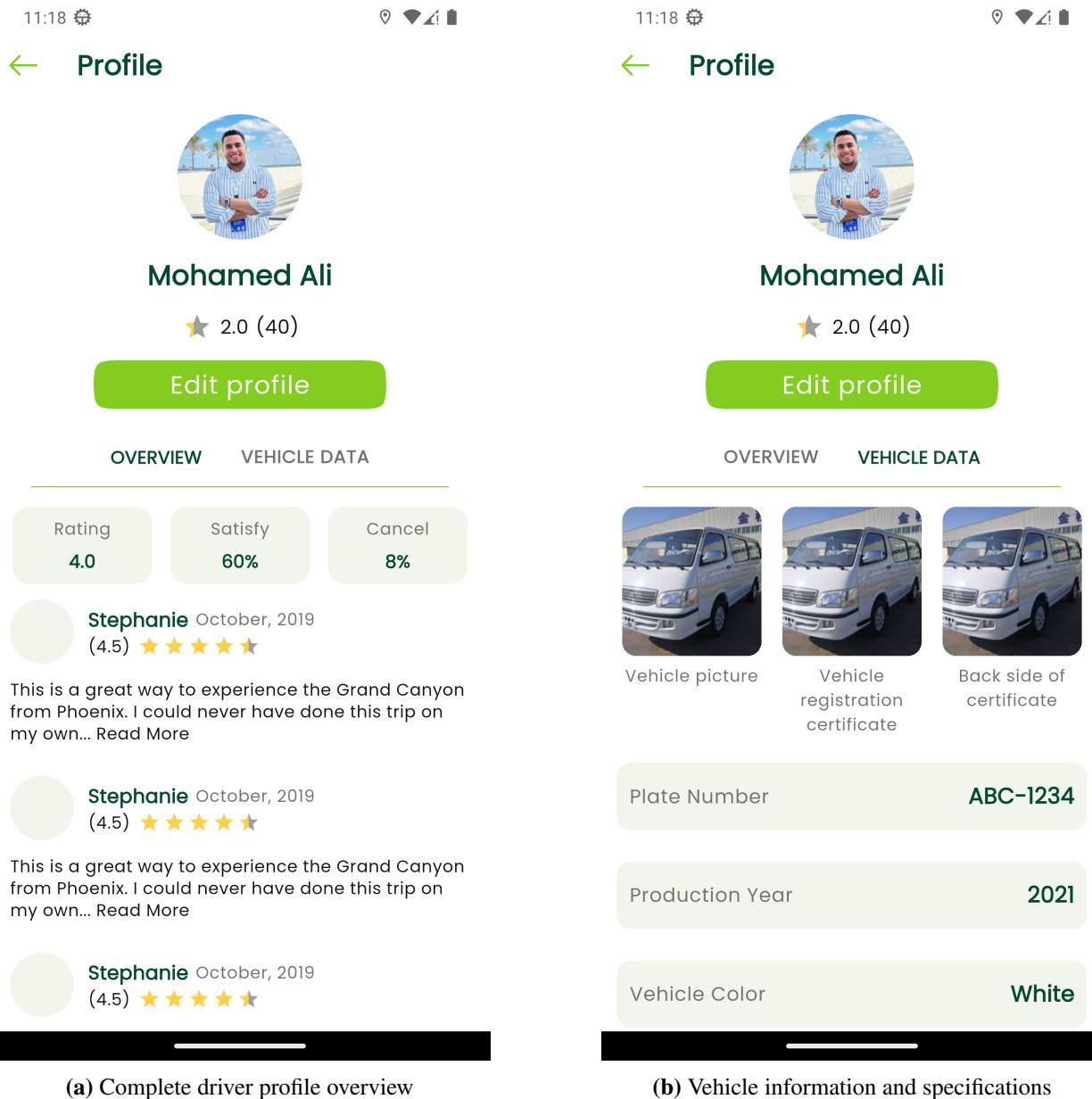


Figure 5.44 – Driver Flow - Profile Overview and Vehicle Information

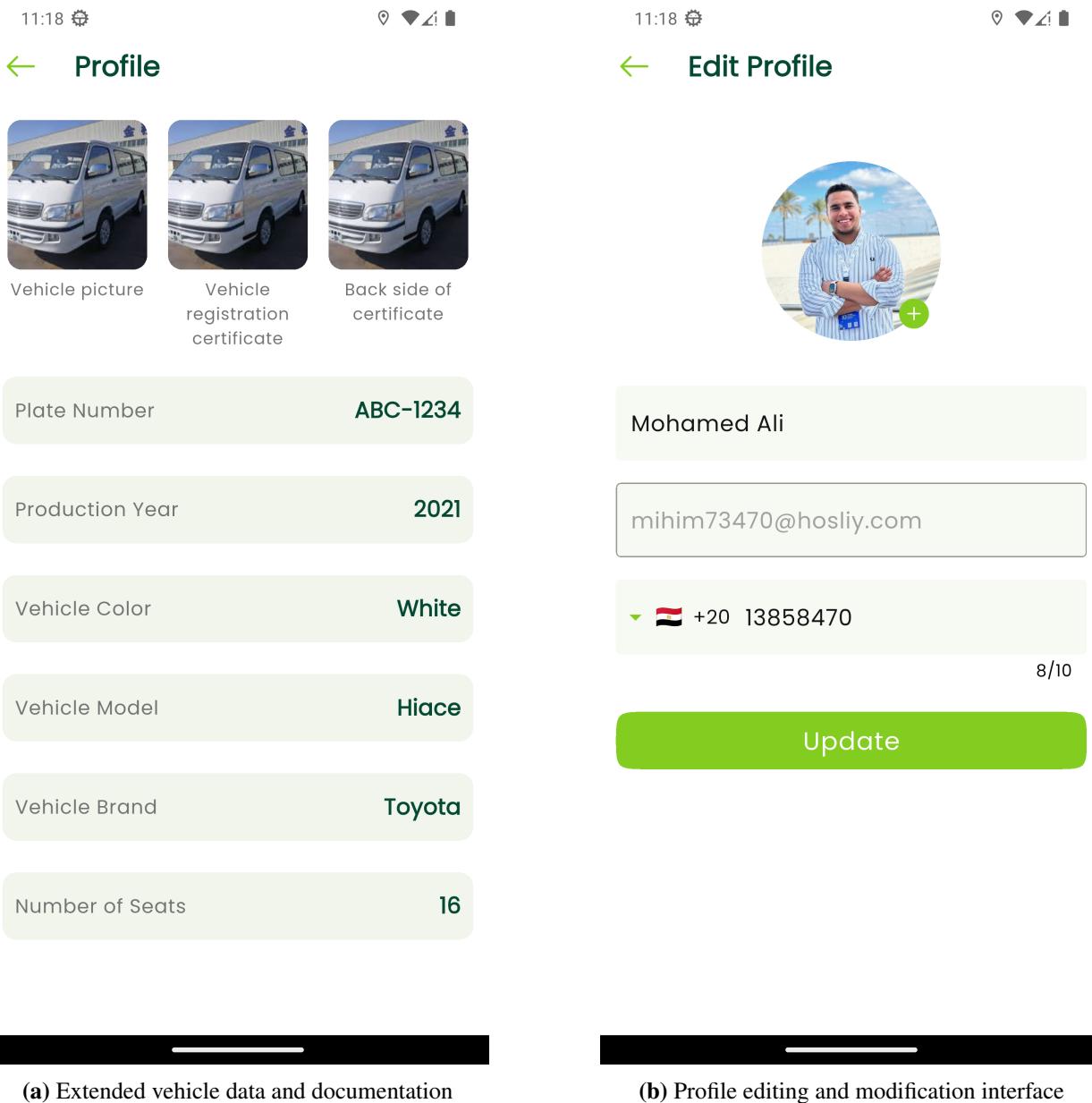
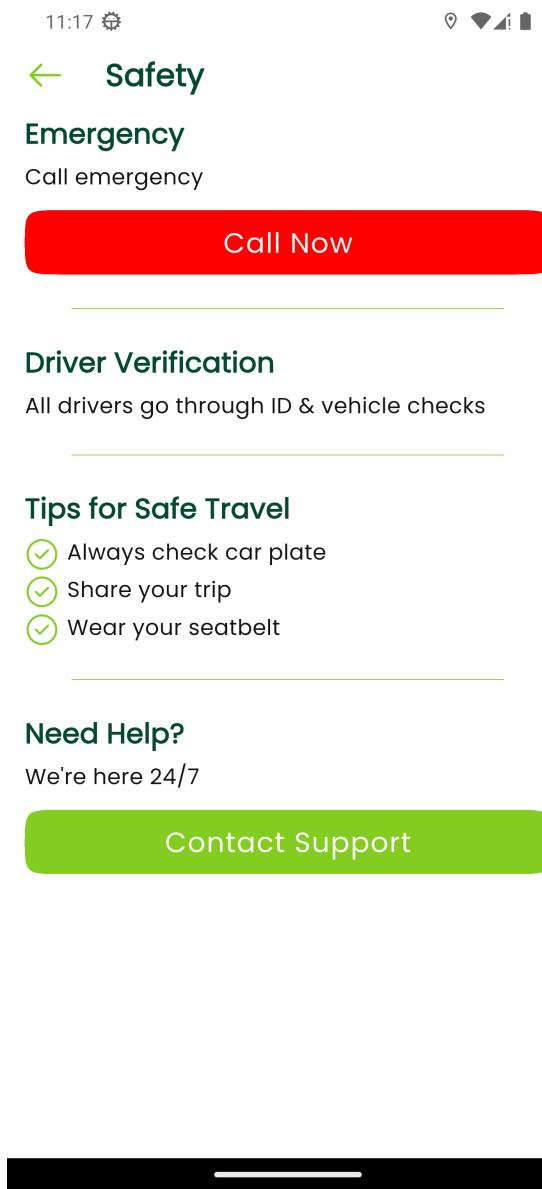


Figure 5.45 – Driver Flow - Extended Vehicle Data and Profile Editing

5.4.5.8 Safety and Security Features

The safety system provides drivers with essential security features, emergency contacts, safety guidelines, and incident reporting capabilities to ensure secure transportation services.



(a) Safety features and security options

Figure 5.46 – Driver Flow - Safety and Security System

5.4.5.9 Wallet and Earnings Management

The financial management system allows drivers to track earnings, manage withdrawals, view transaction history, and handle payment processing. This comprehensive financial dashboard provides transparency and control over driver compensation.

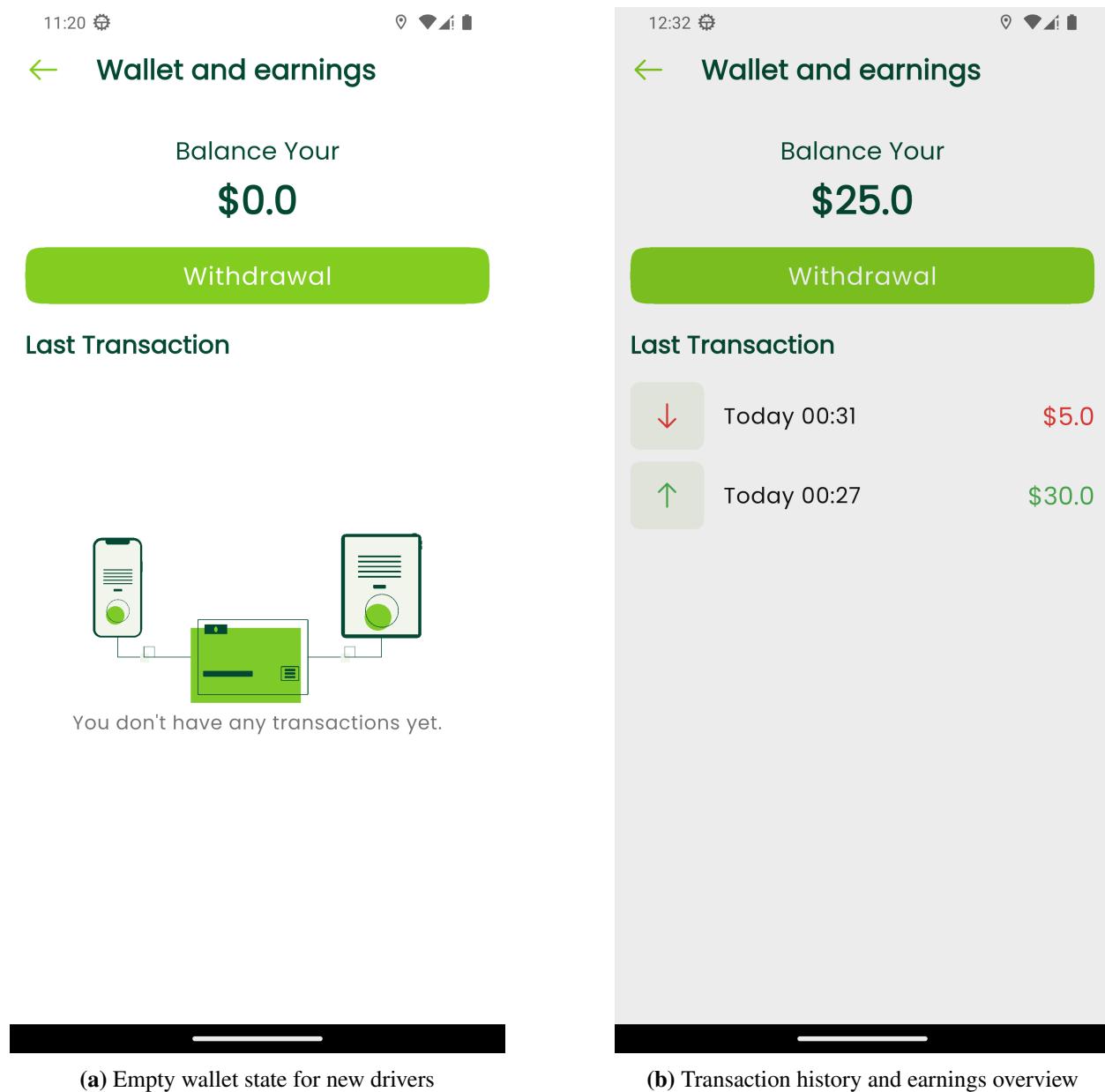


Figure 5.47 – Driver Flow - Wallet Overview and Transactions

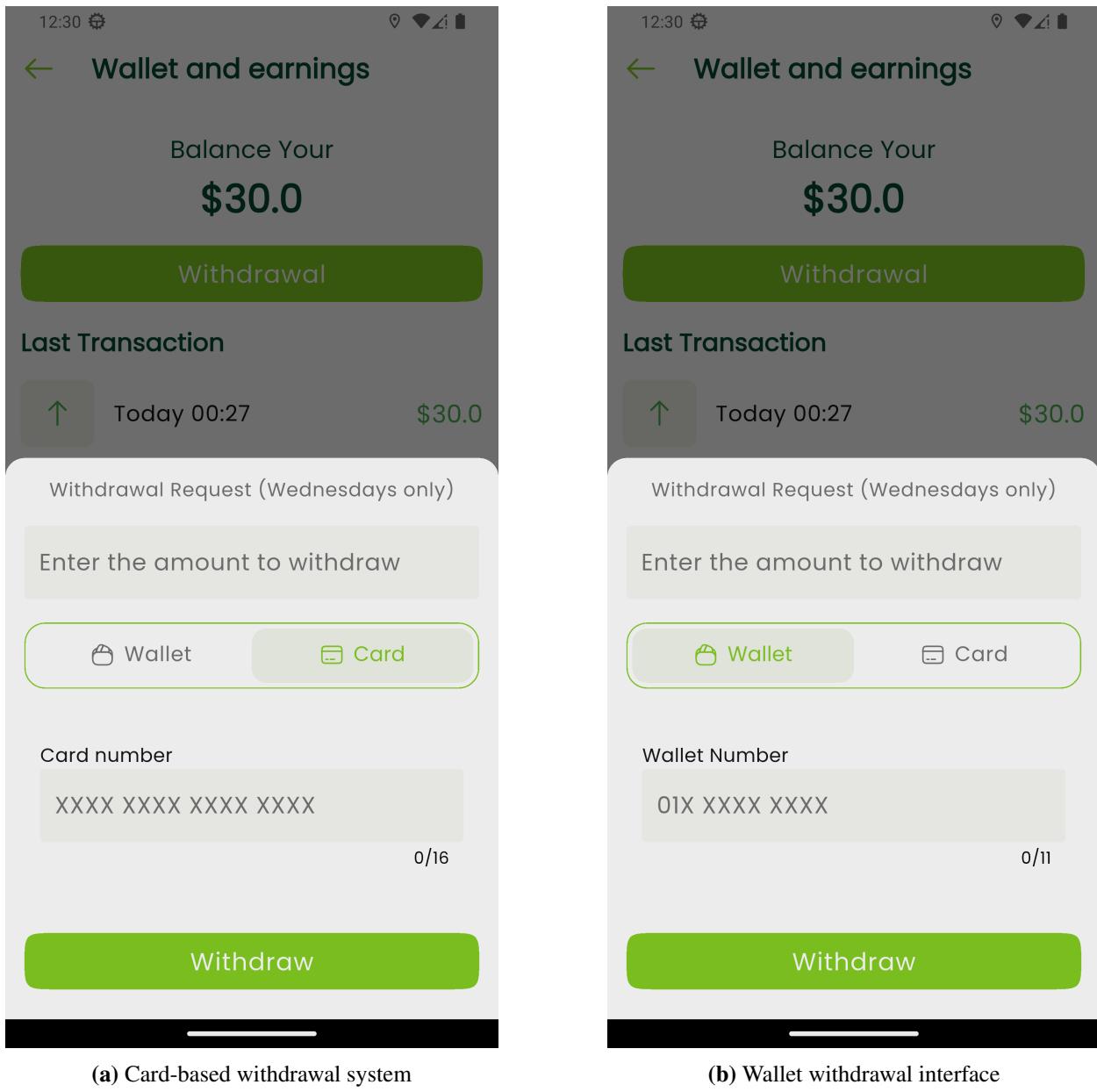


Figure 5.48 – Driver Flow - Withdrawal Methods and Processing

5.4.6 User Flow Management System

The user flow management system provides comprehensive functionality for passengers to search for trips, make bookings, manage payments, and track their travel history. This system is designed to offer a seamless experience for transportation service users.

5.4.6.1 Card Management System

The card management system allows users to securely add, manage, and use payment cards for trip bookings. This comprehensive system ensures secure payment processing and convenient payment method management.

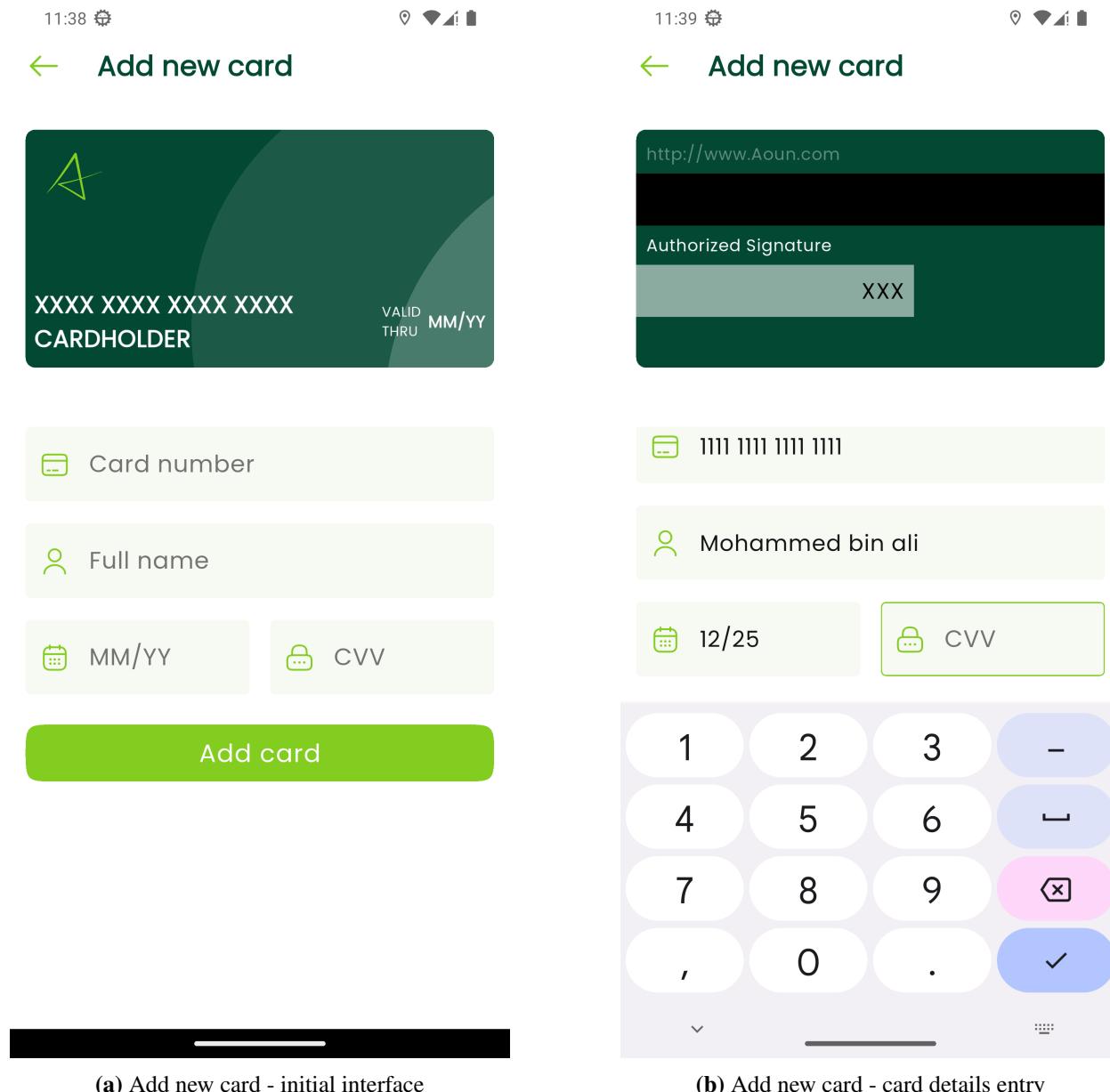


Figure 5.49 – User Flow - Card Addition Process (Part 1 & 2)

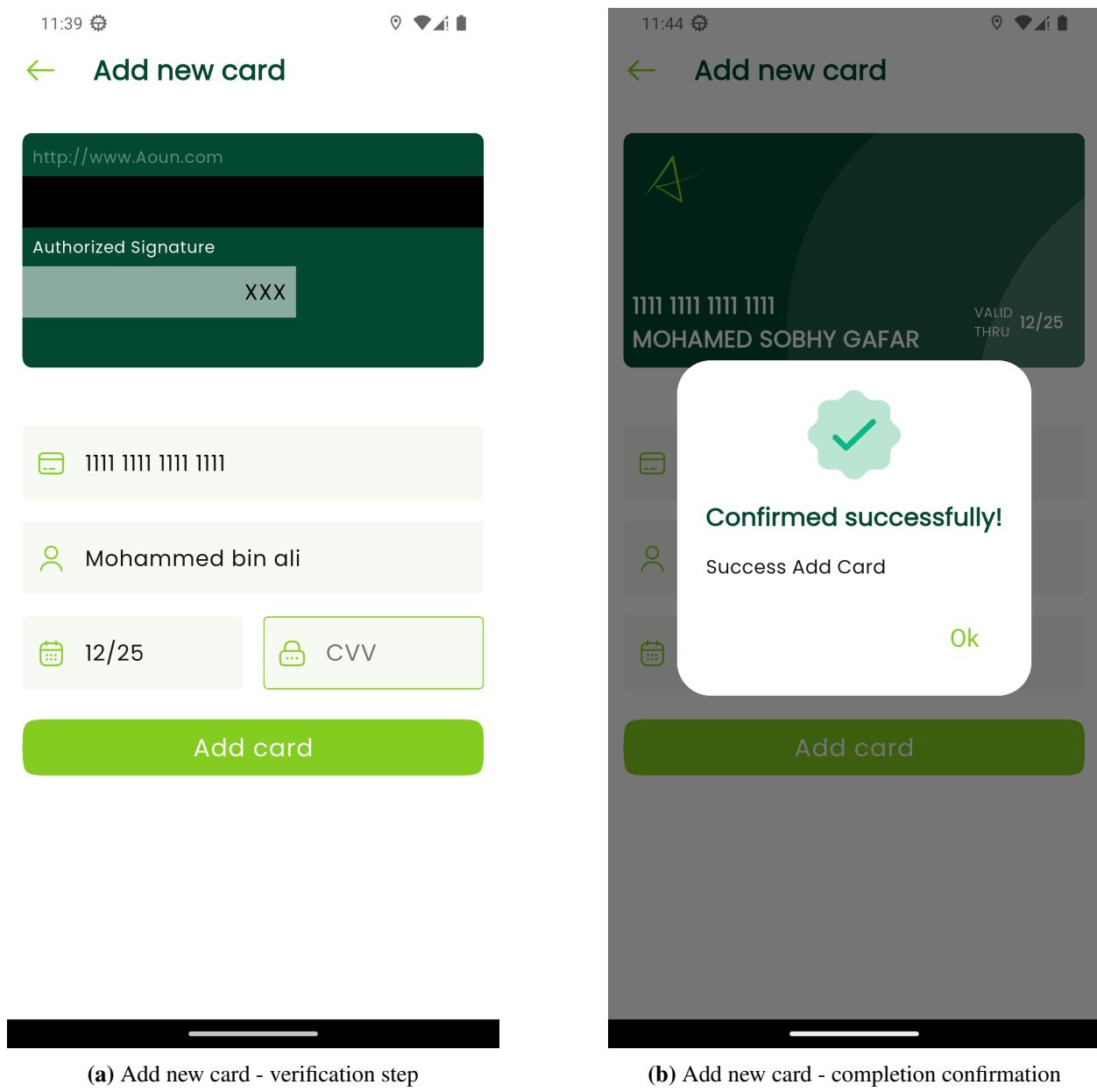


Figure 5.50 – User Flow - Card Addition Process (Part 3 & 4)

5.4.6.2 Booking History Management

The booking history system provides users with access to their past trips, allowing them to review travel patterns, track expenses, and access trip receipts for record-keeping purposes.

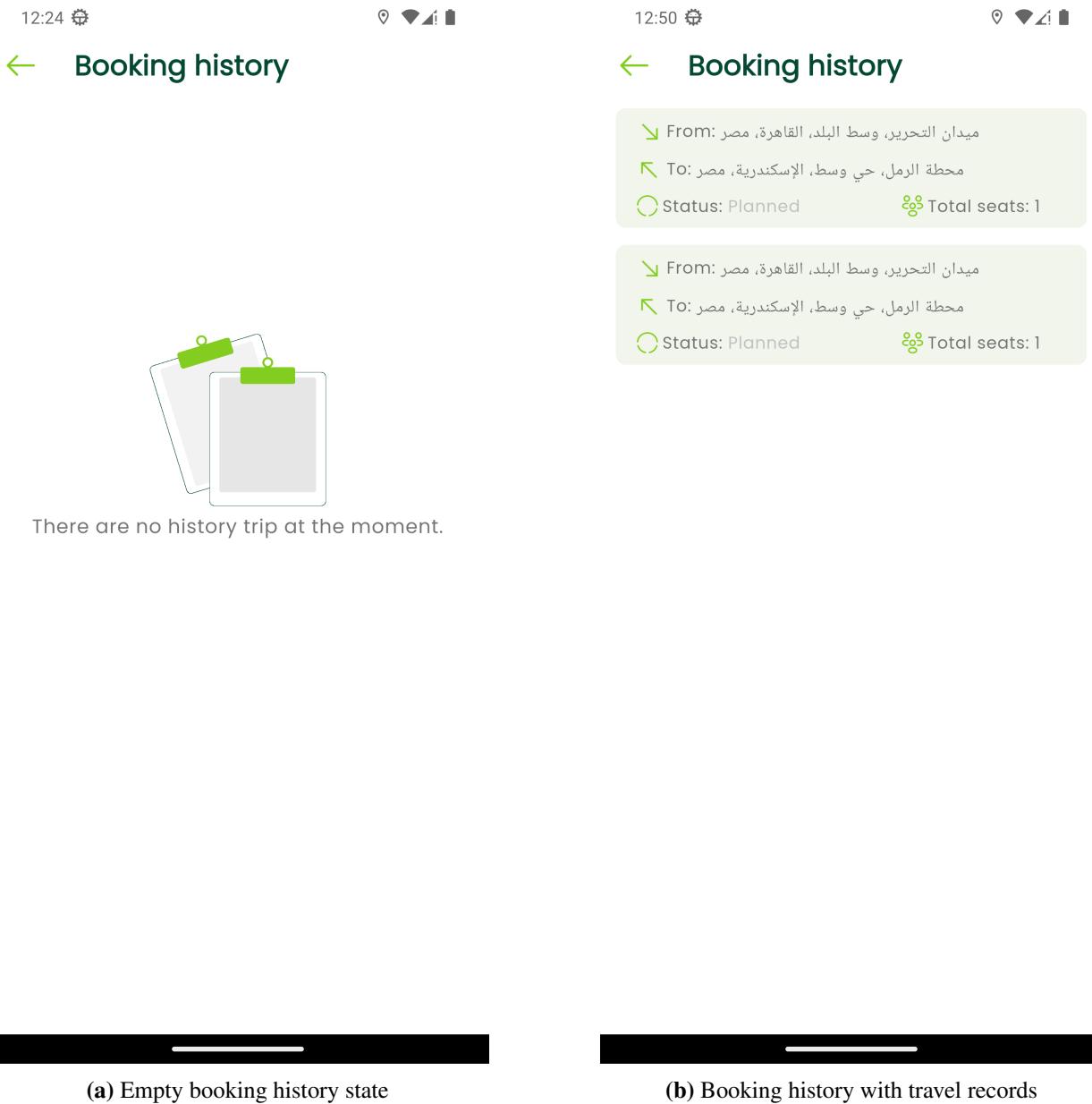


Figure 5.51 – User Flow - Booking History Management

5.4.6.3 Home Dashboard System

The user home dashboard serves as the central hub for passengers, providing quick access to trip search, recent bookings, and personalized recommendations based on travel patterns.

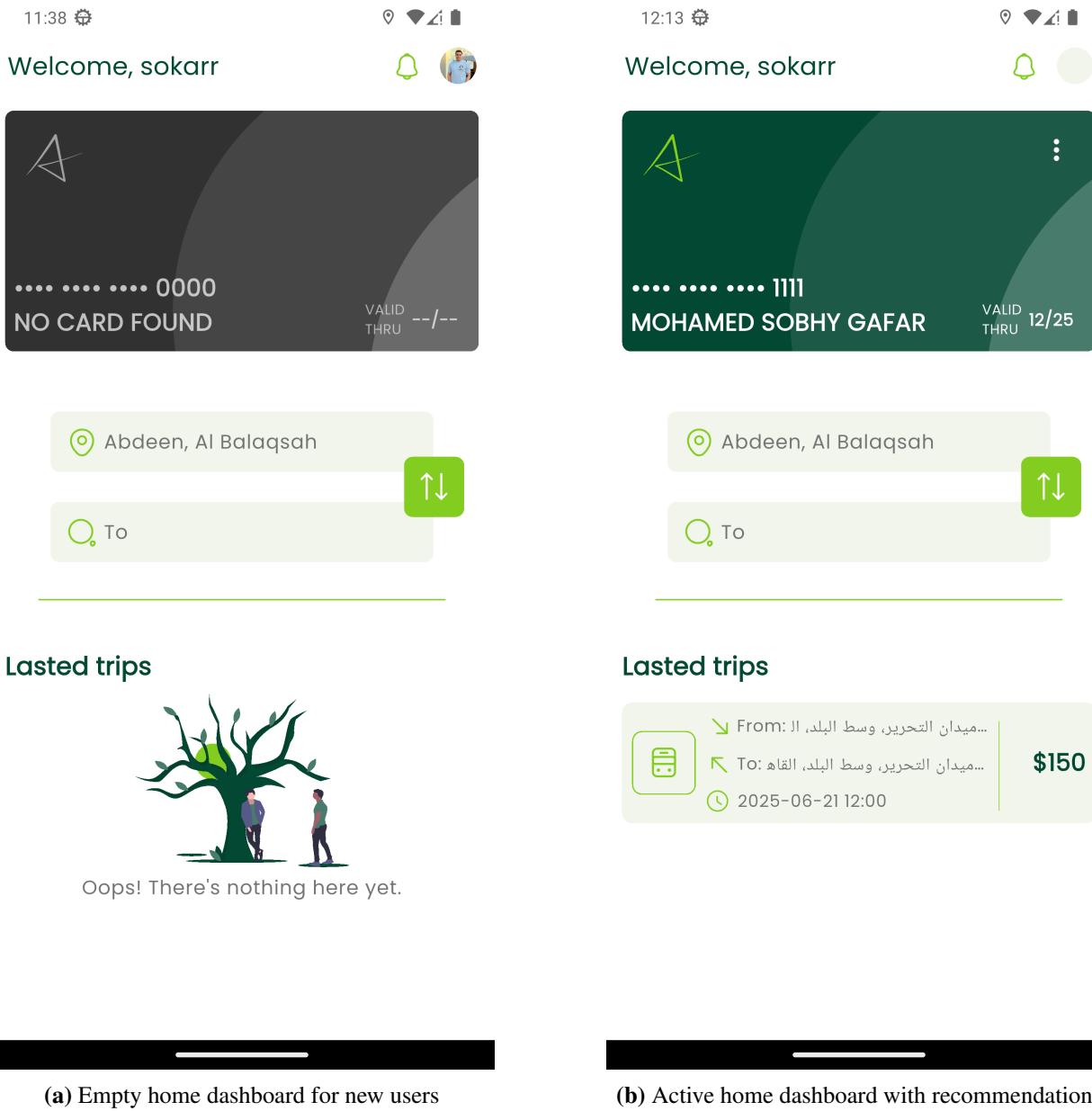


Figure 5.52 – User Flow - Home Dashboard States

5.4.6.4 Payment Processing System

The payment system provides users with multiple payment options, secure transaction processing, and comprehensive payment management for trip bookings.

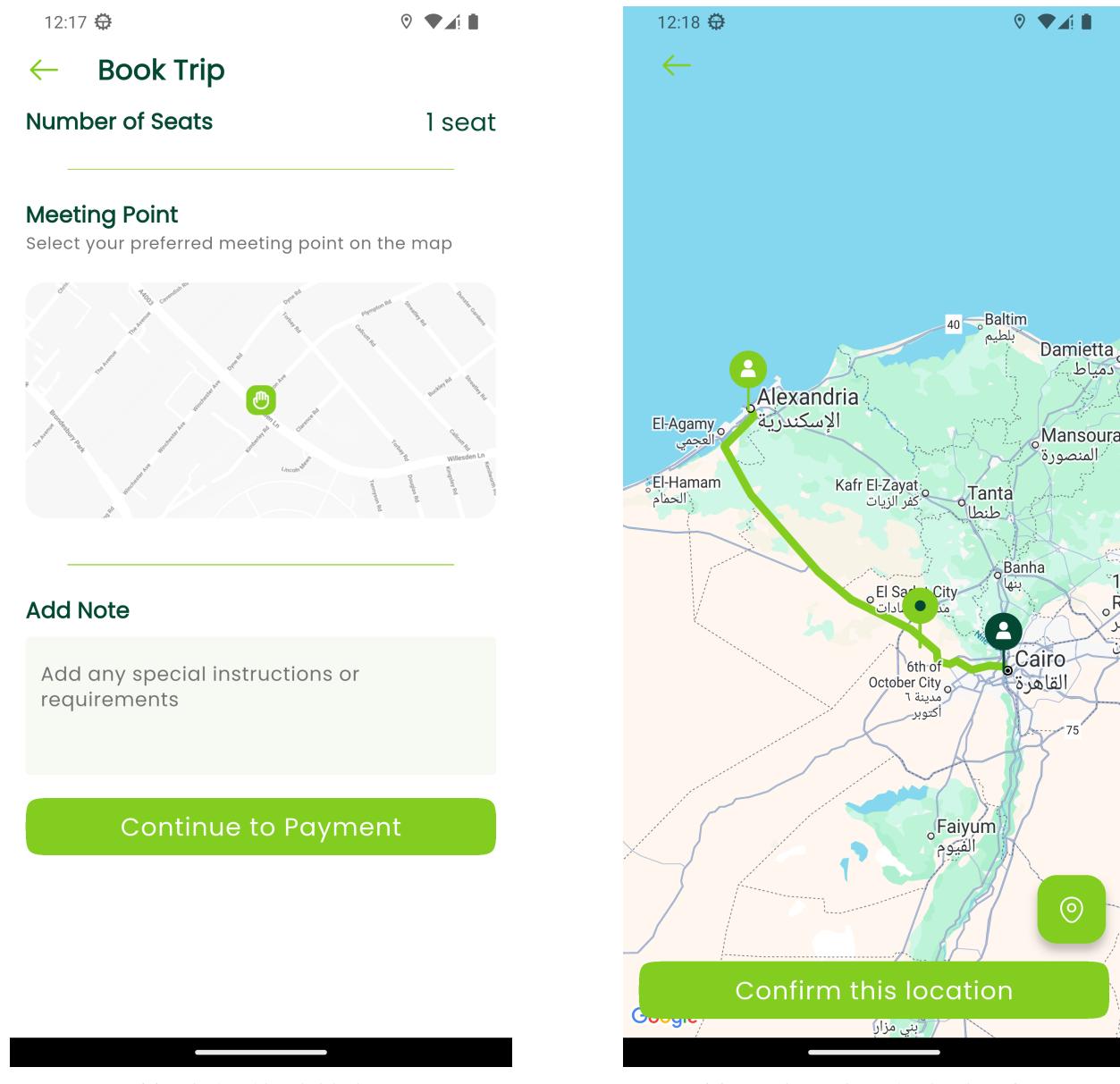


Figure 5.53 – User Flow - Booking and Meeting Point Selection

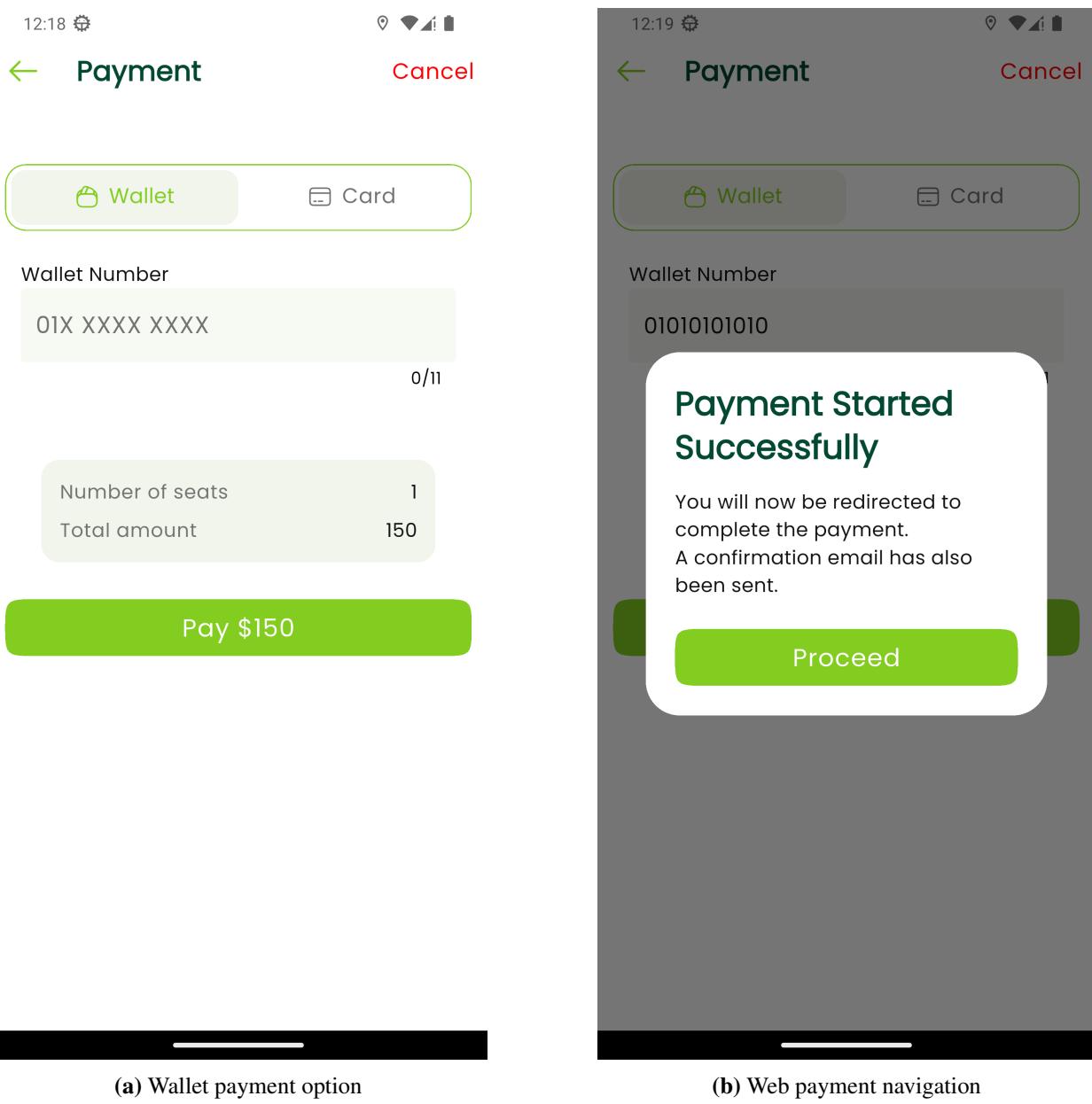


Figure 5.54 – User Flow - Wallet Payment and Web Navigation

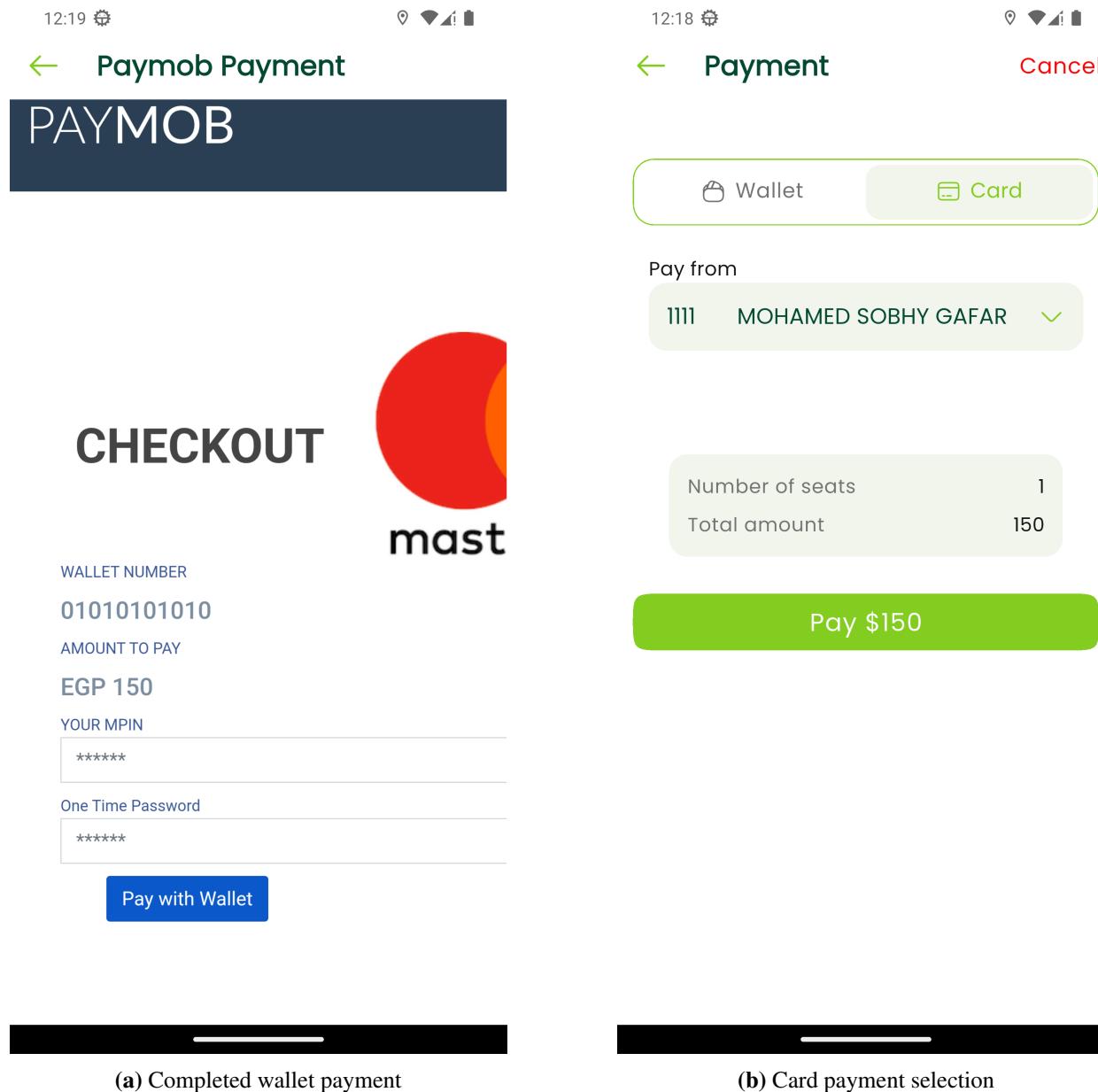
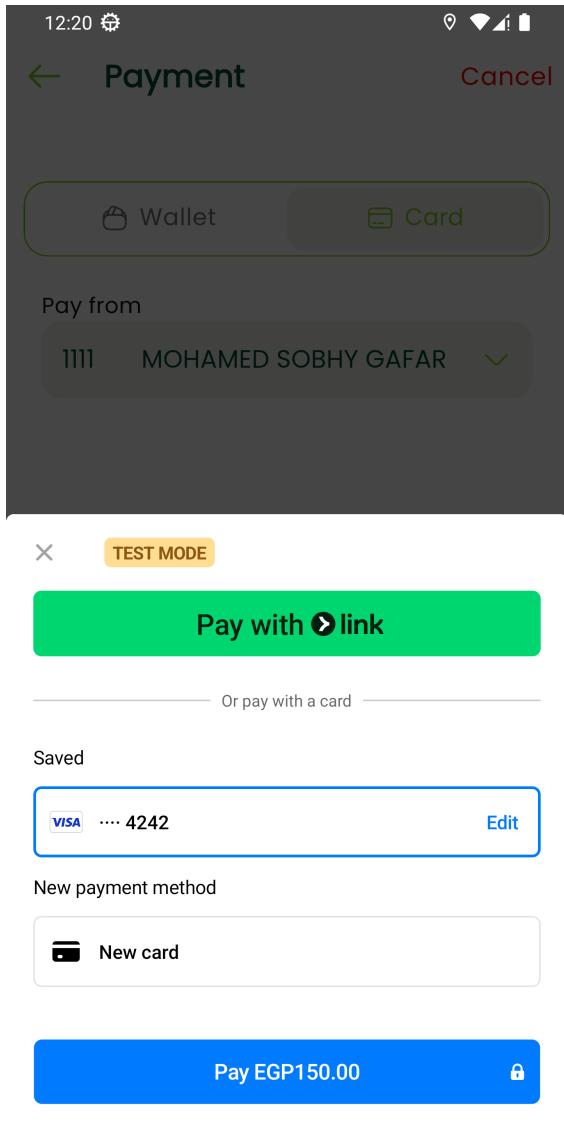
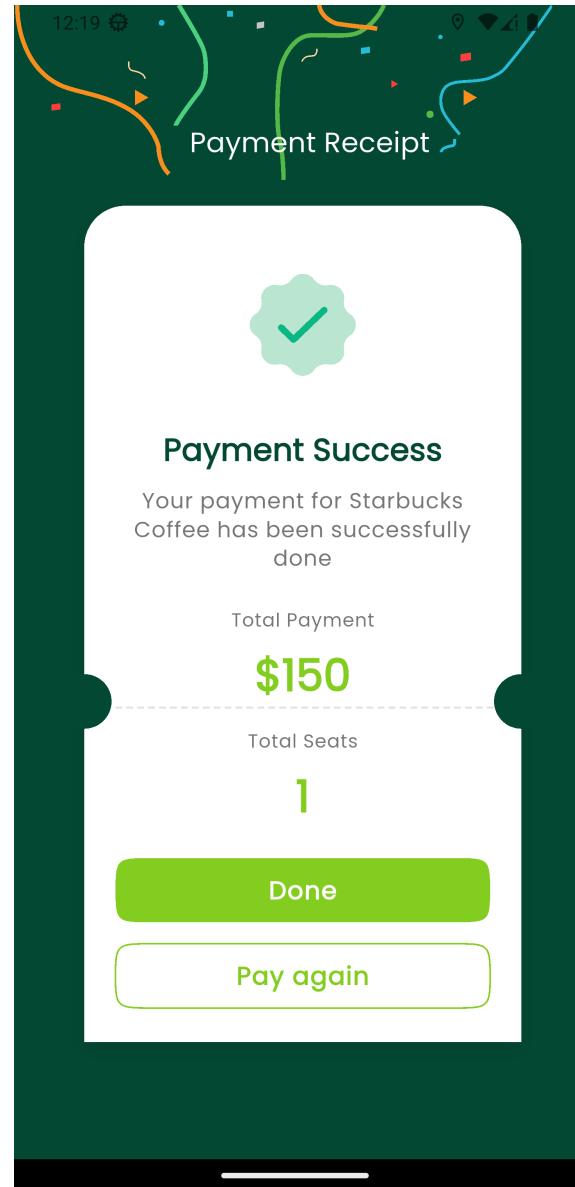


Figure 5.55 – User Flow - Wallet Completion and Card Payment



(a) Card payment with Stripe integration



(b) Payment receipt and confirmation

Figure 5.56 – User Flow - Stripe Payment and Receipt

5.4.6.5 Profile Management System

The profile management system allows users to customize their account settings, manage personal information, and configure application preferences for an optimal user experience.

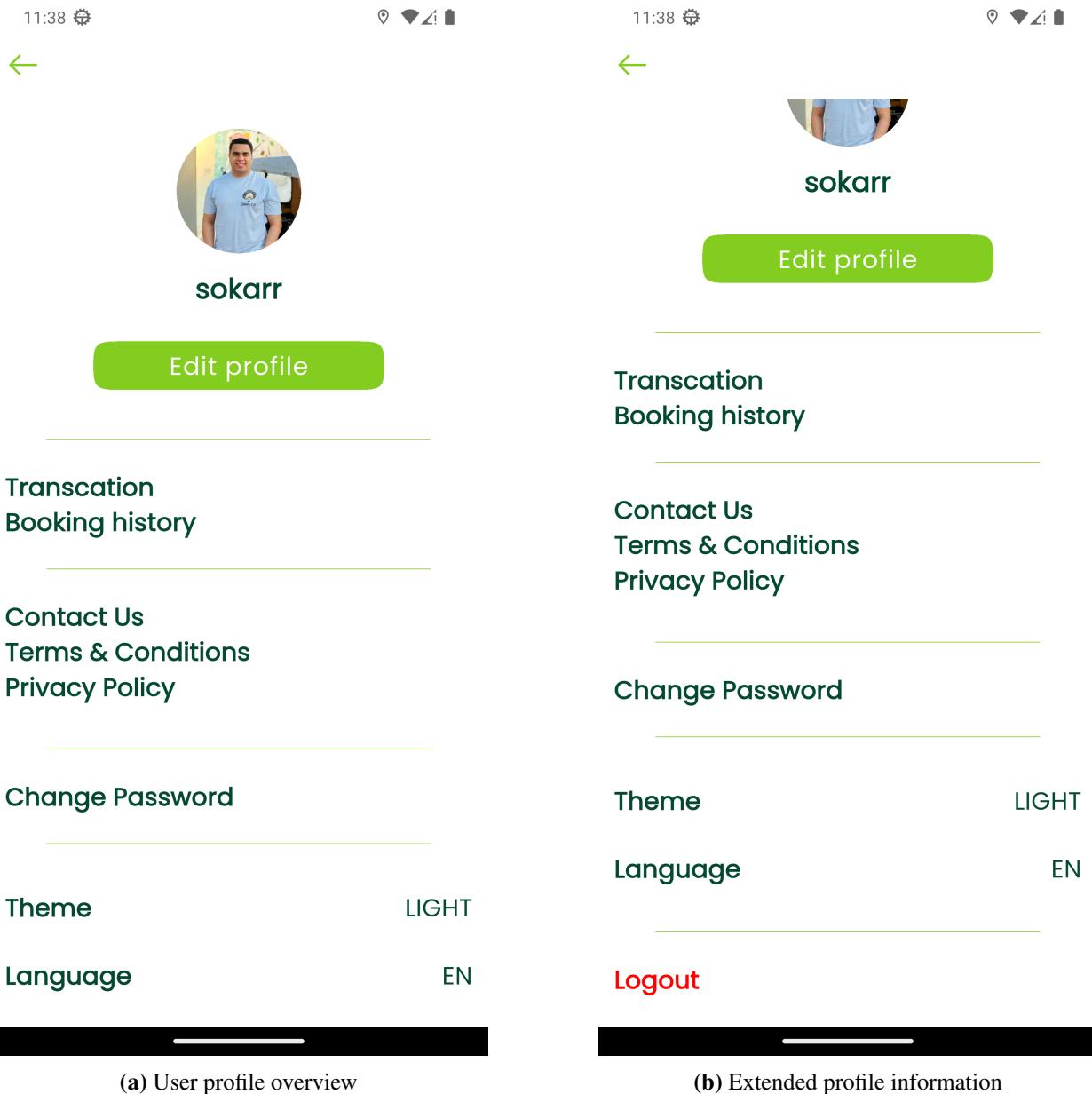
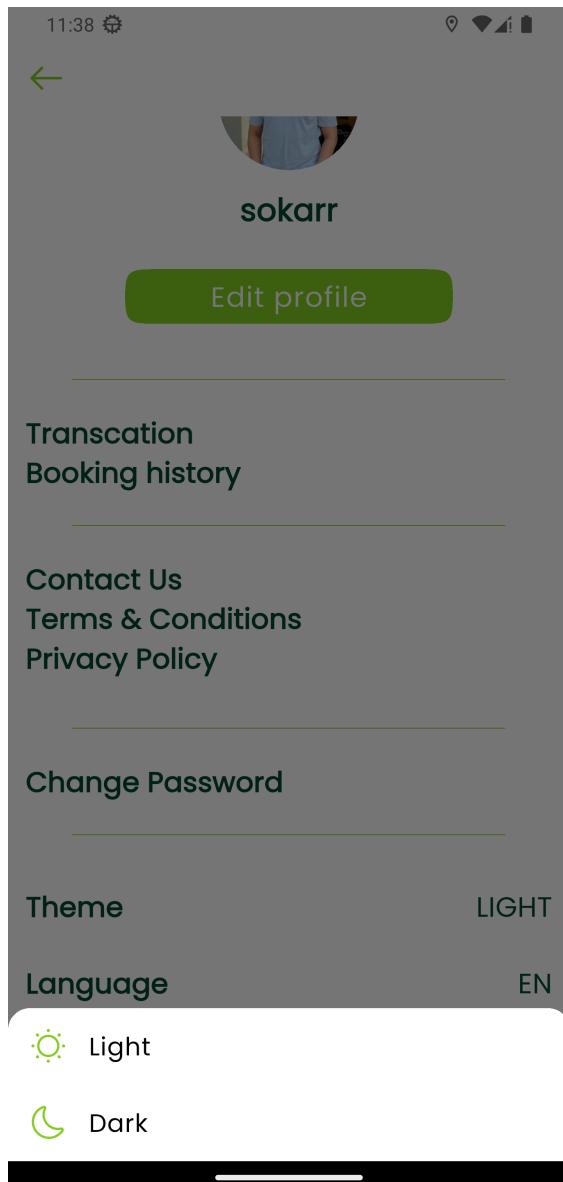
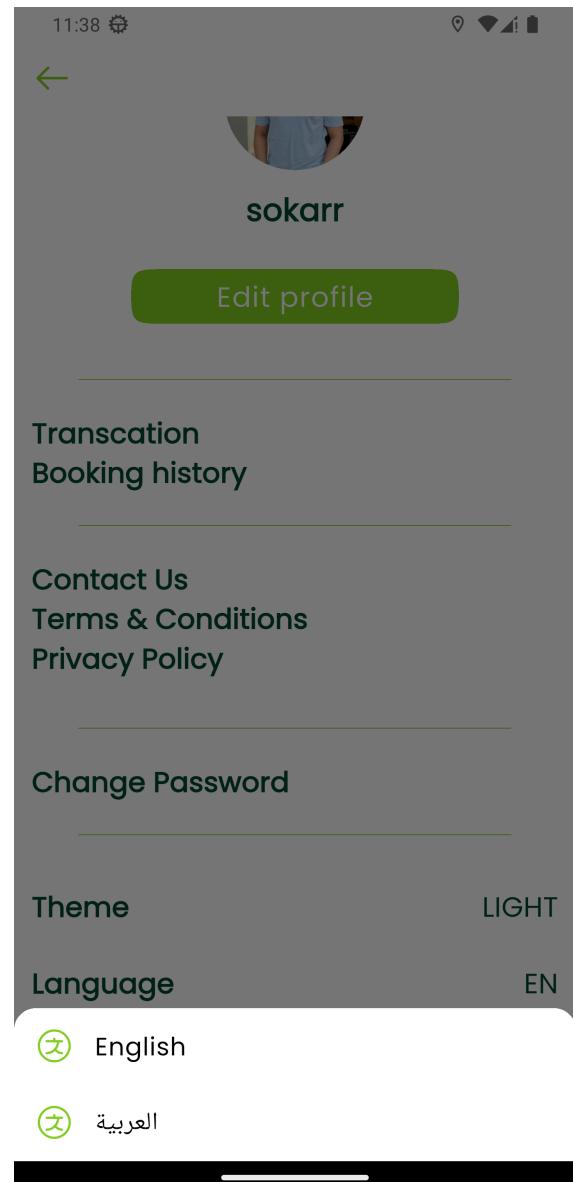


Figure 5.57 – User Flow - Profile Overview and Extended Information

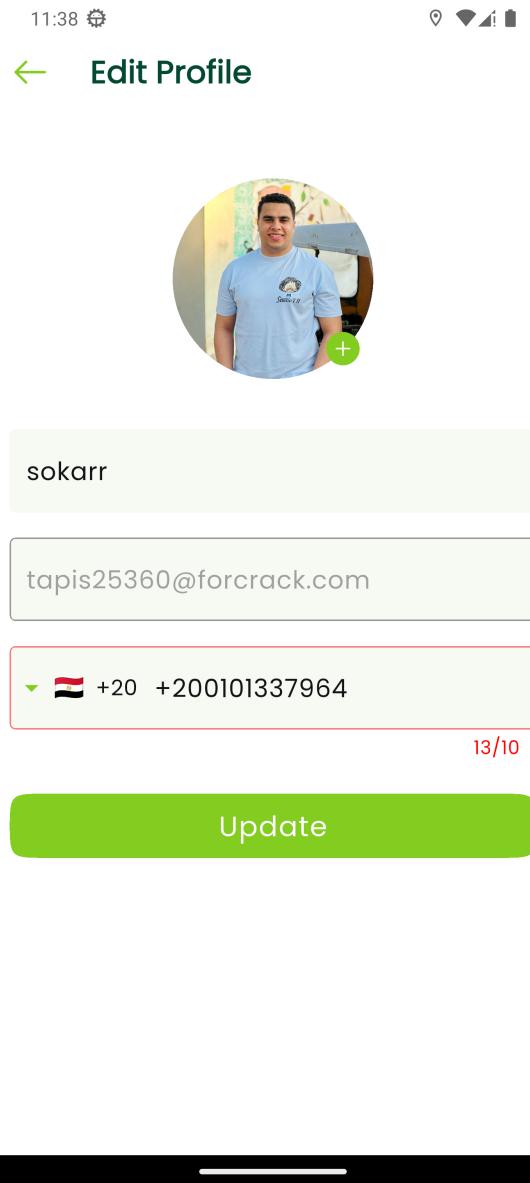


(a) Theme customization options



(b) Language selection interface

Figure 5.58 – User Flow - Theme and Language Customization



(a) Profile editing interface

Figure 5.59 – User Flow - Profile Editing

5.4.6.6 Trip Search System

The trip search system enables users to find available transportation options, view recommendations, and access location-based search functionality for optimal trip planning.

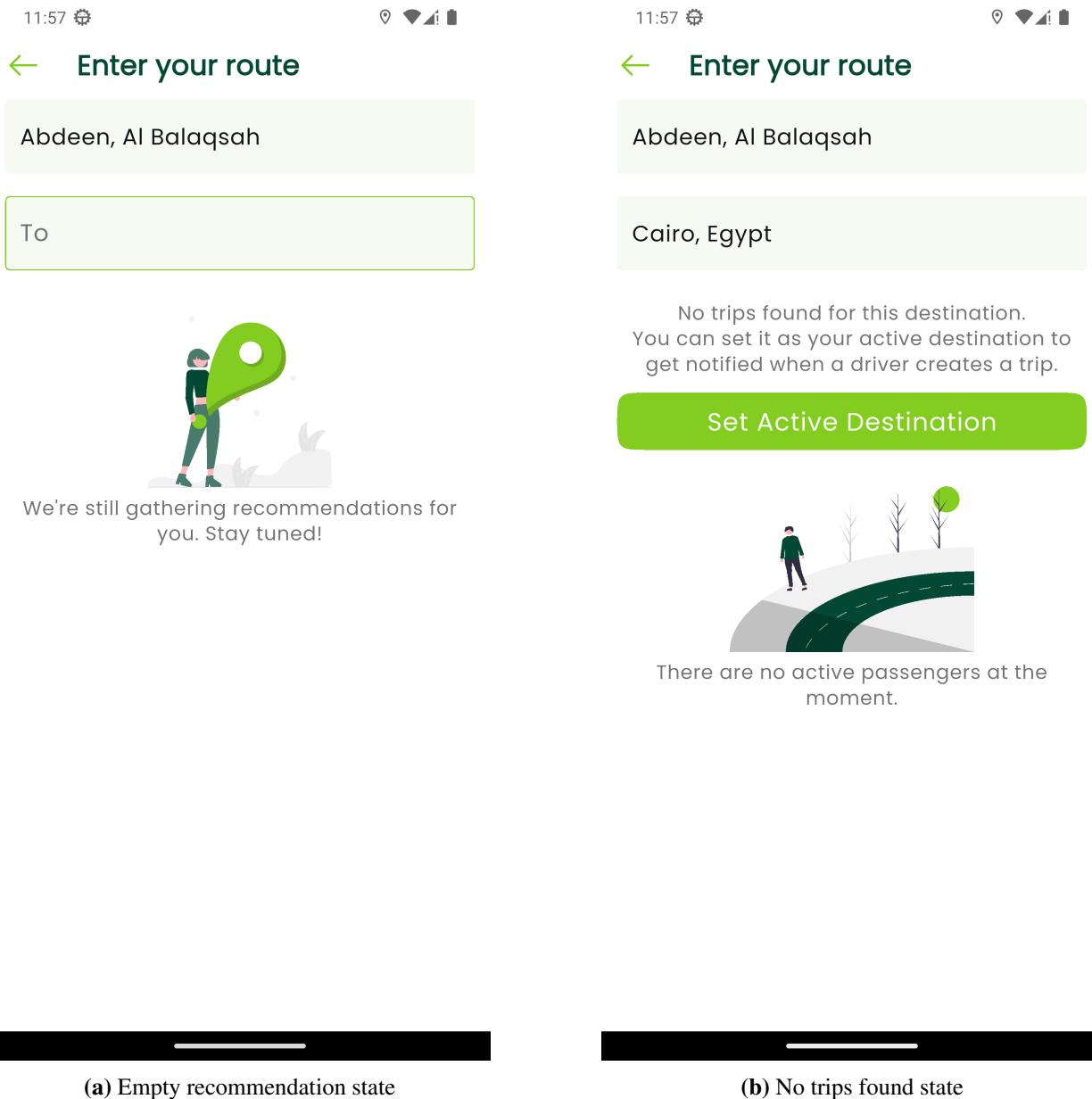


Figure 5.60 – User Flow - Empty Search States

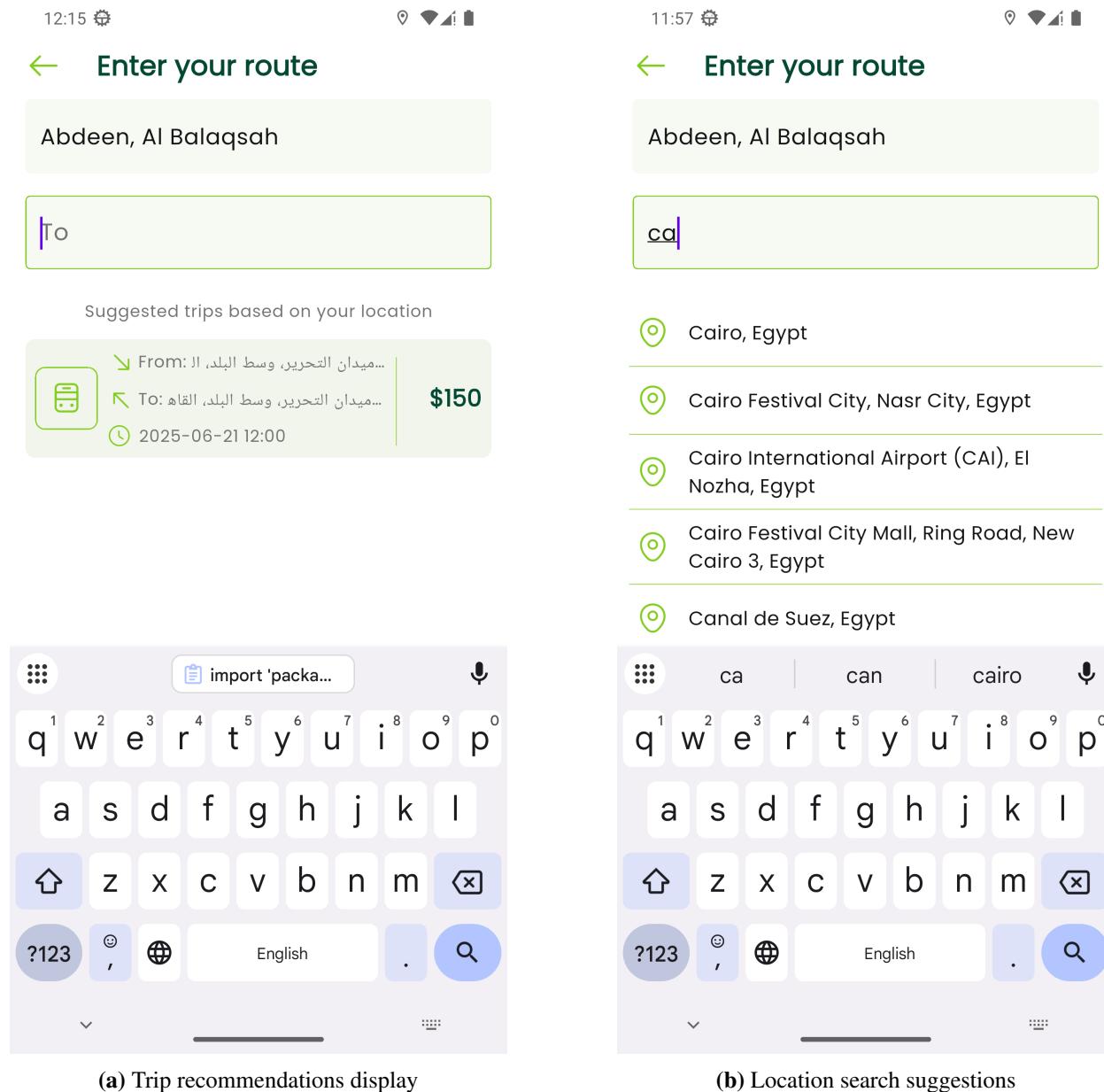


Figure 5.61 – User Flow - Trip Recommendations and Location Search

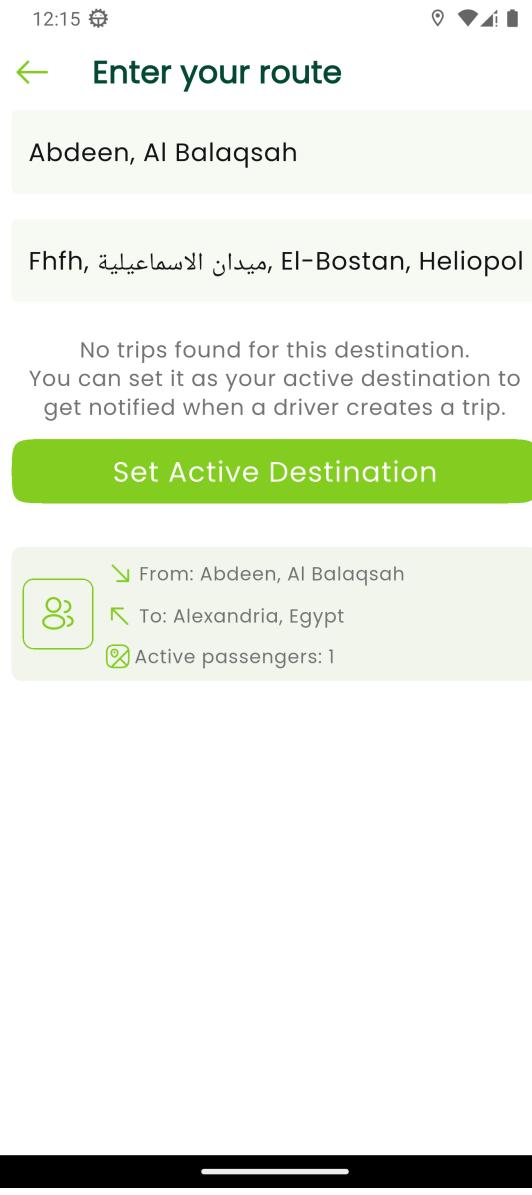


Figure 5.62 – User Flow - Similar Distance Travel Analysis

5.4.6.7 Transaction Management

The transaction system provides users with comprehensive financial tracking, payment history, and transaction details for all trip-related expenses.



(a) Transaction history and details

Figure 5.63 – User Flow - Transaction Management

5.4.6.8 Trip Details and Information

The trip details system provides users with comprehensive information about their booked trips, including route details, driver information, and meeting point specifications.

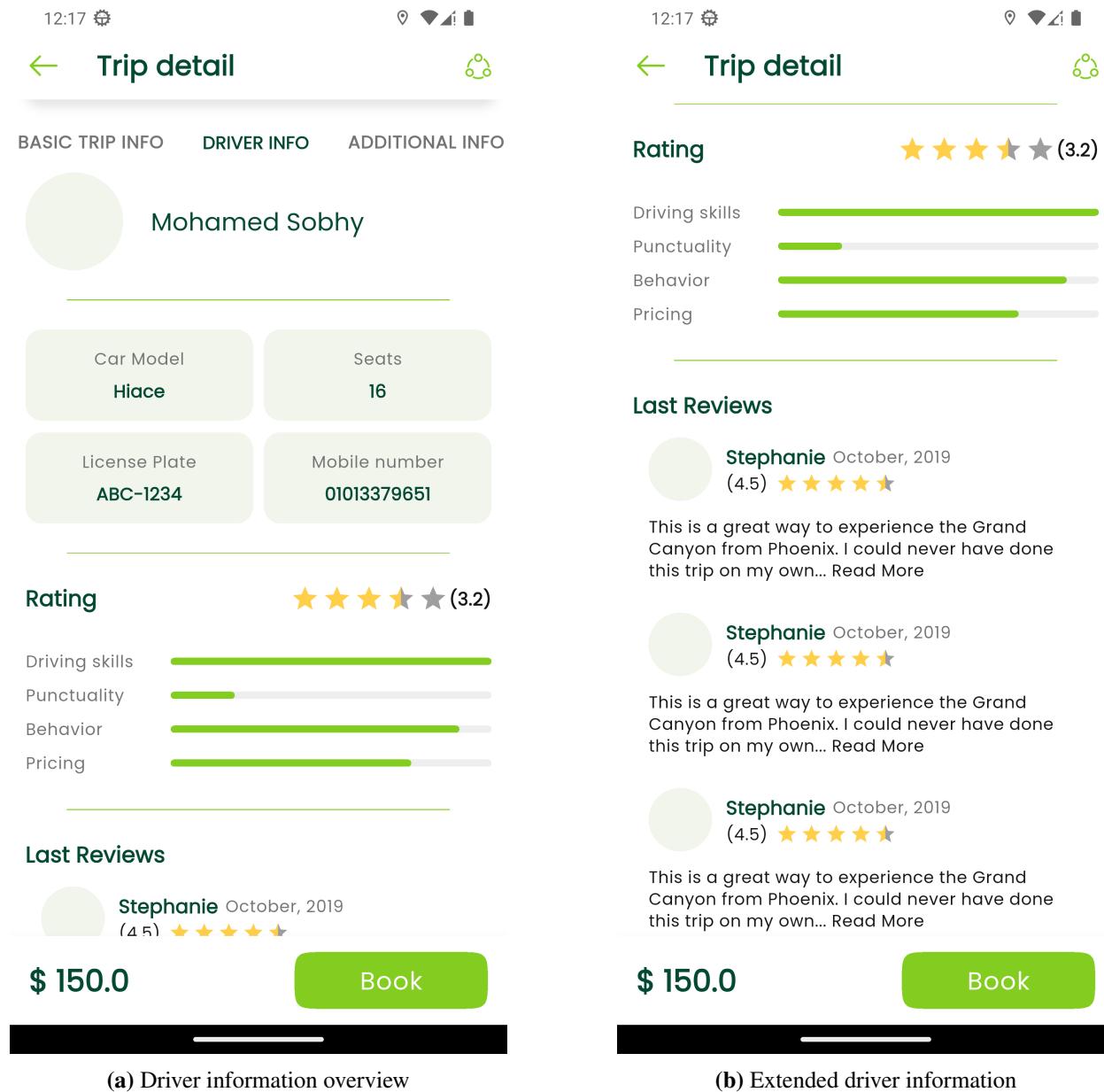
(a) Comprehensive trip detail overview

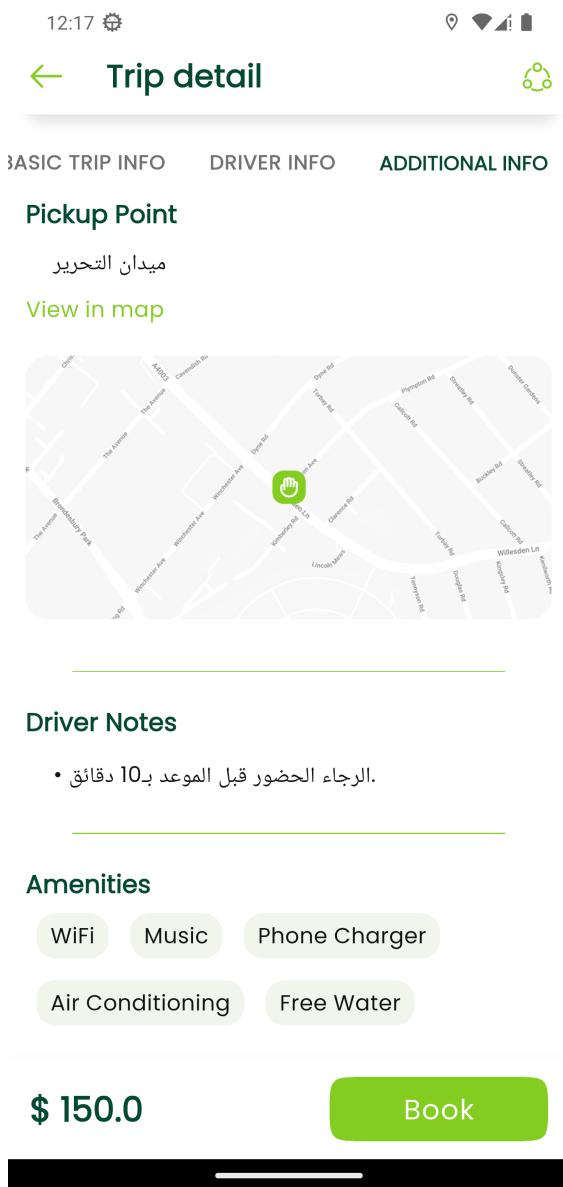
This screenshot shows the trip detail overview screen. At the top, it displays the departure time (12:15) and location (ميدان التحرير، وسط البلد، القاهرة، مصر). Below this, the destination is listed as محطة الرمل، حي وسط، الإسكندرية، مصر. The screen is divided into three tabs: BASIC TRIP INFO, DRIVER INFO, and ADDITIONAL INFO. Under BASIC TRIP INFO, the departure date is 2025-06-21 12:00, the estimated distance is 220 KM, the estimated arrival time is 2h 30min, and there are 3 Available Seats. A green button labeled "Book" is at the bottom right.

(b) Route visualization on map

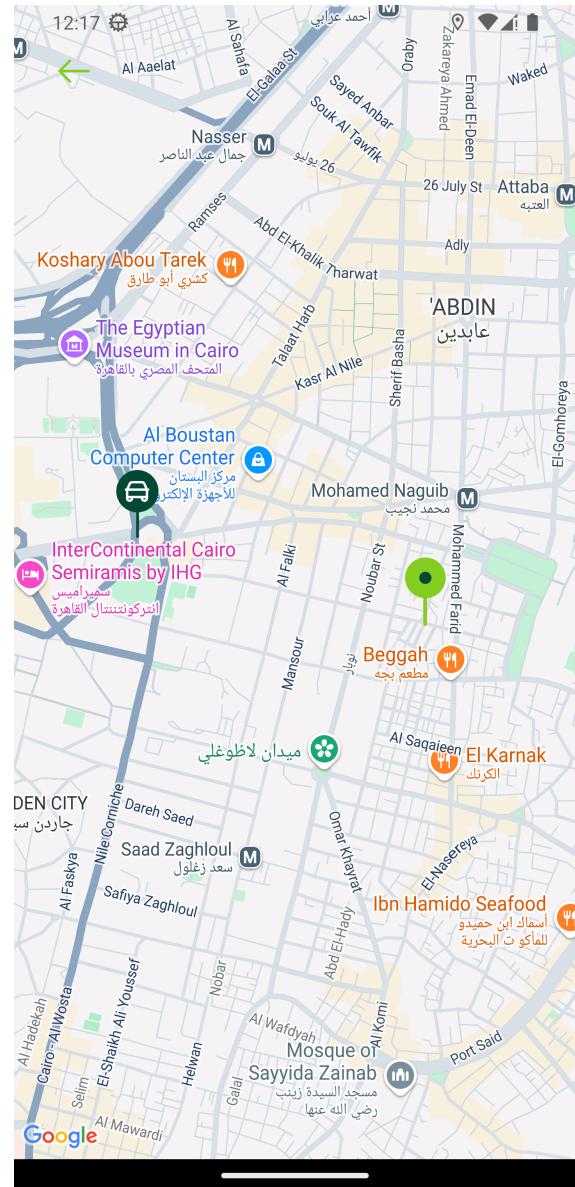
This screenshot shows the route visualization on a map. The map displays a route from Cairo (القاهرة) to Alexandria (الإسكندرية), highlighted by a thick green line. The map also shows other cities like Tanta, Mansura, and Faiyum, along with various roads and landmarks. The Google logo is visible in the bottom left corner of the map view.

Figure 5.64 – User Flow - Trip Details and Route Visualization

**Figure 5.65 – User Flow - Driver Information Display**



(a) Additional trip information



(b) Meeting point with vehicle details

Figure 5.66 – User Flow - Additional Information and Meeting Point Details

5.4.7 Utilities and System Features

The application includes various utility features and system components that enhance the overall user experience through localization, theme management, loading states, and input validation.

5.4.7.1 Localization and Theme Management

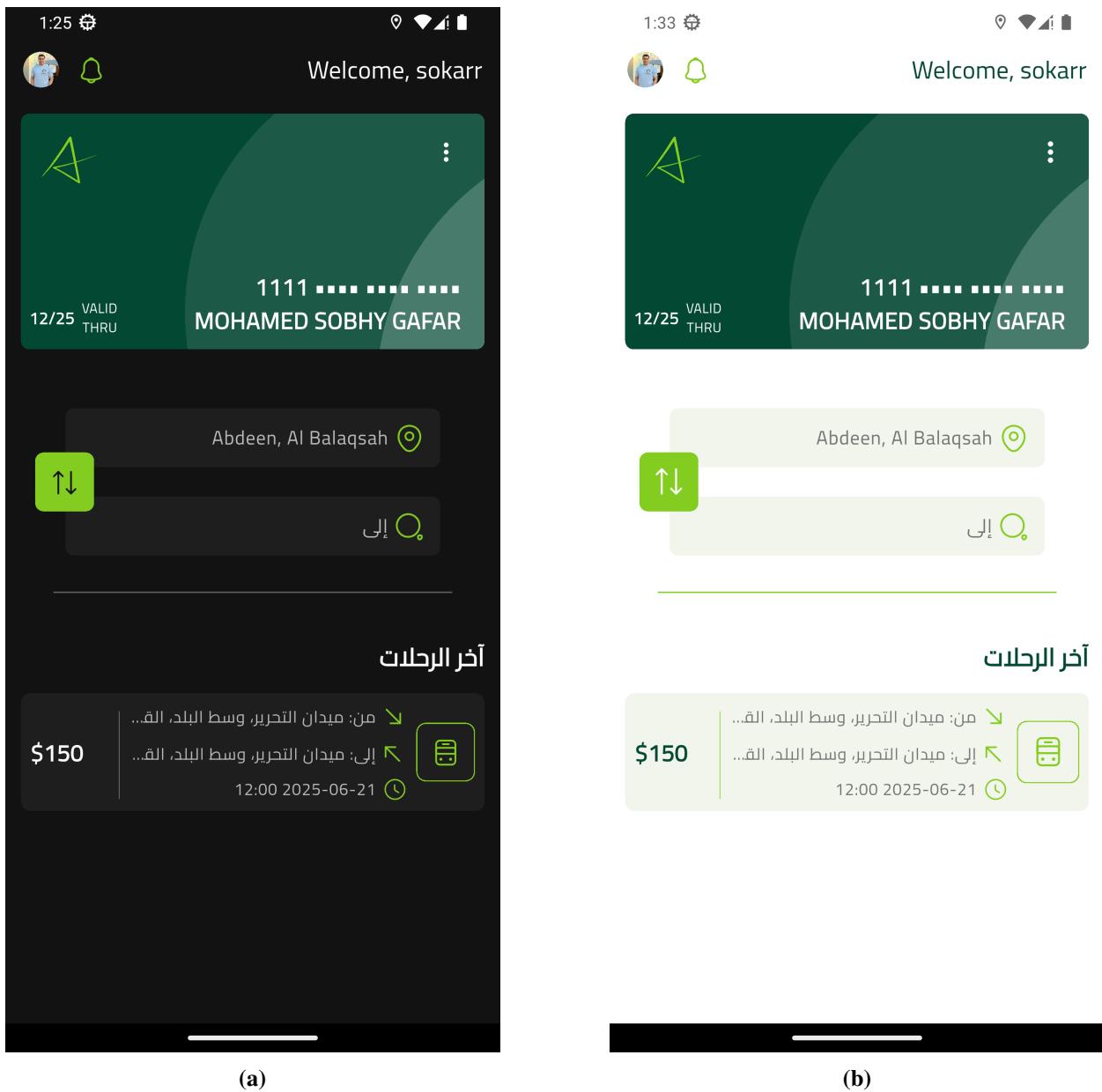


Figure 5.67 – Localization Features

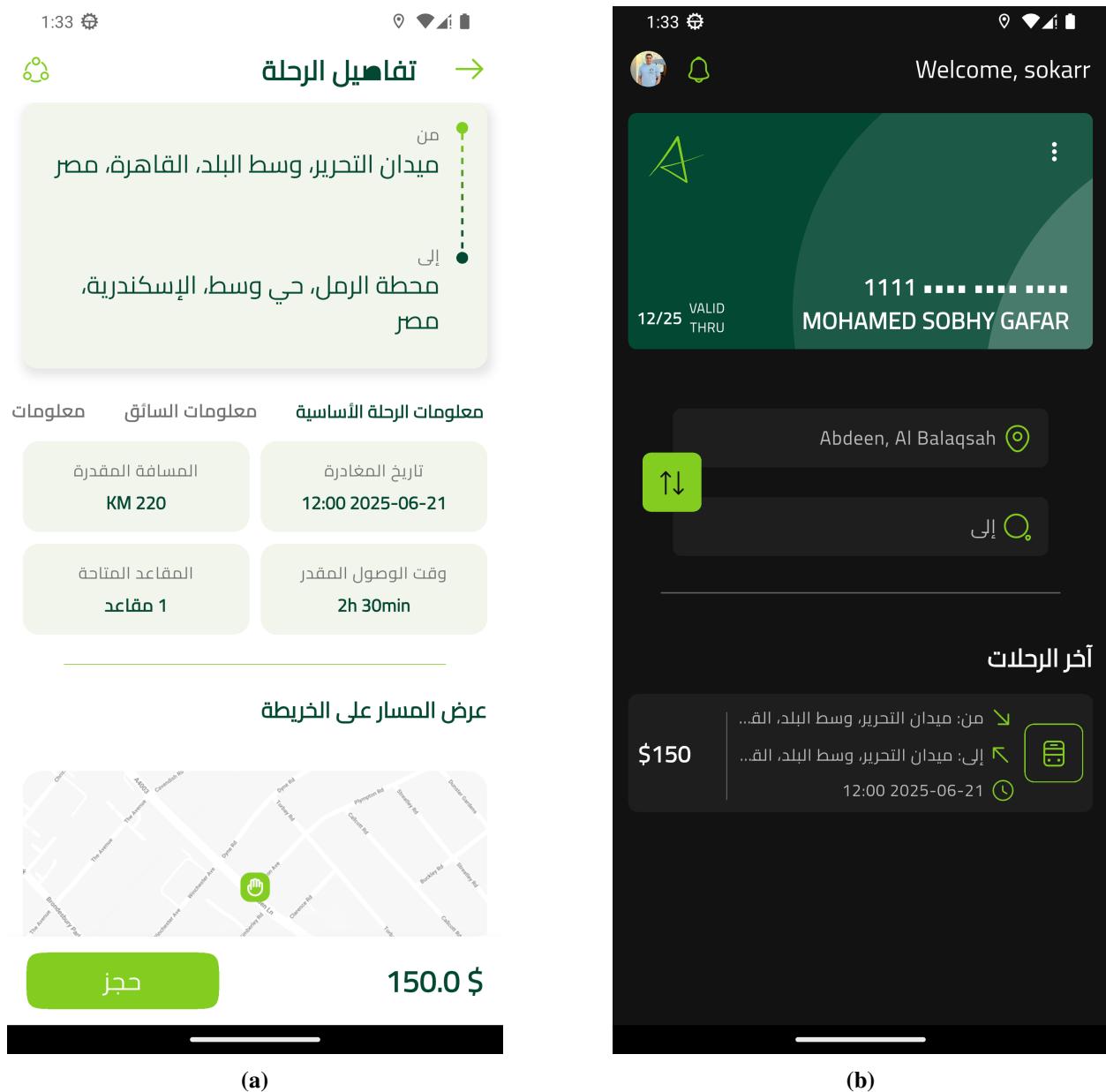


Figure 5.68 – Extended Localization Options

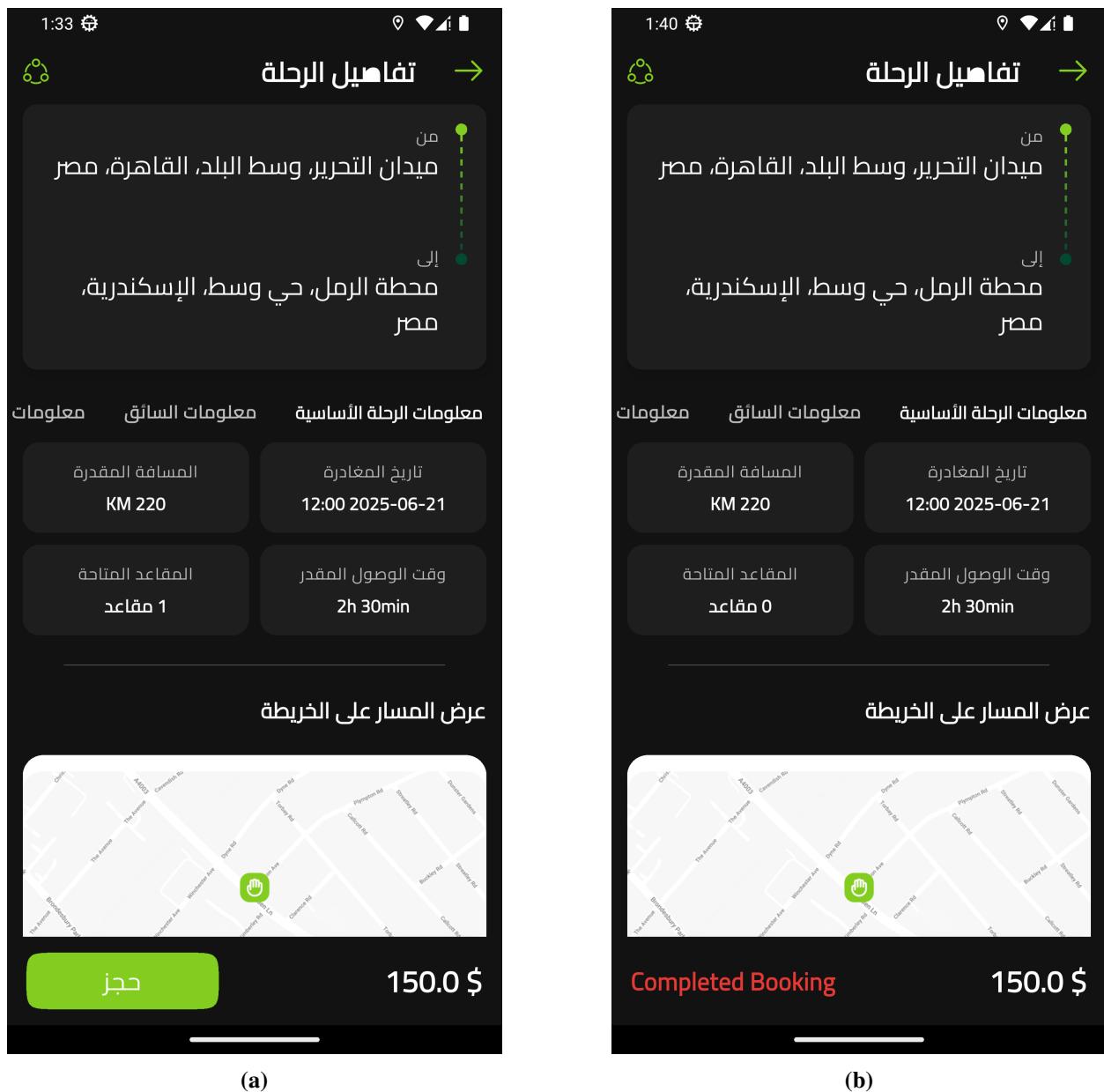


Figure 5.69 – Theme Management Options

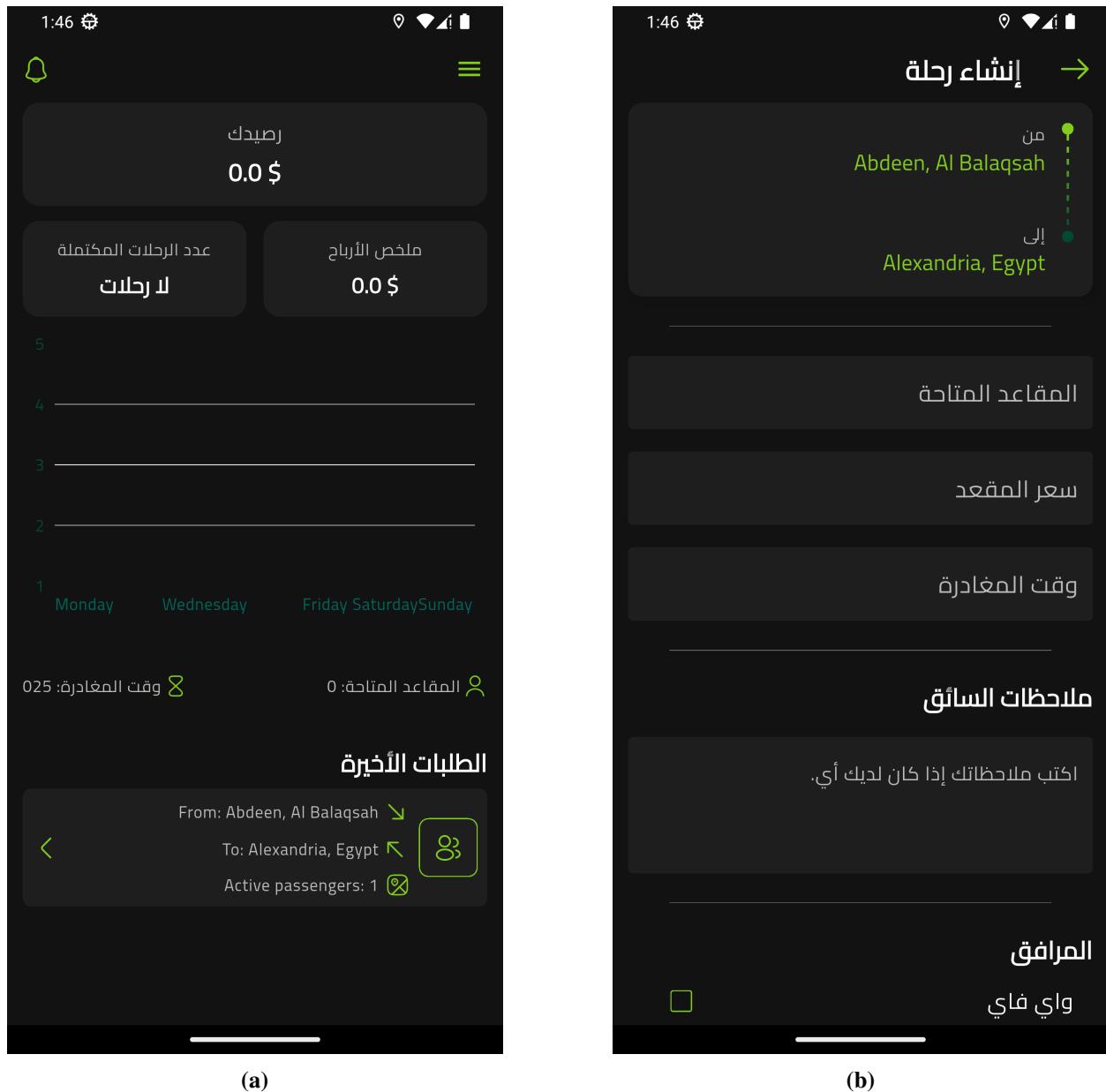


Figure 5.70 – Advanced Theme Customization

5.4.7.2 Loading States and Shimmer Effects

5.4.7.3 Input Validation System

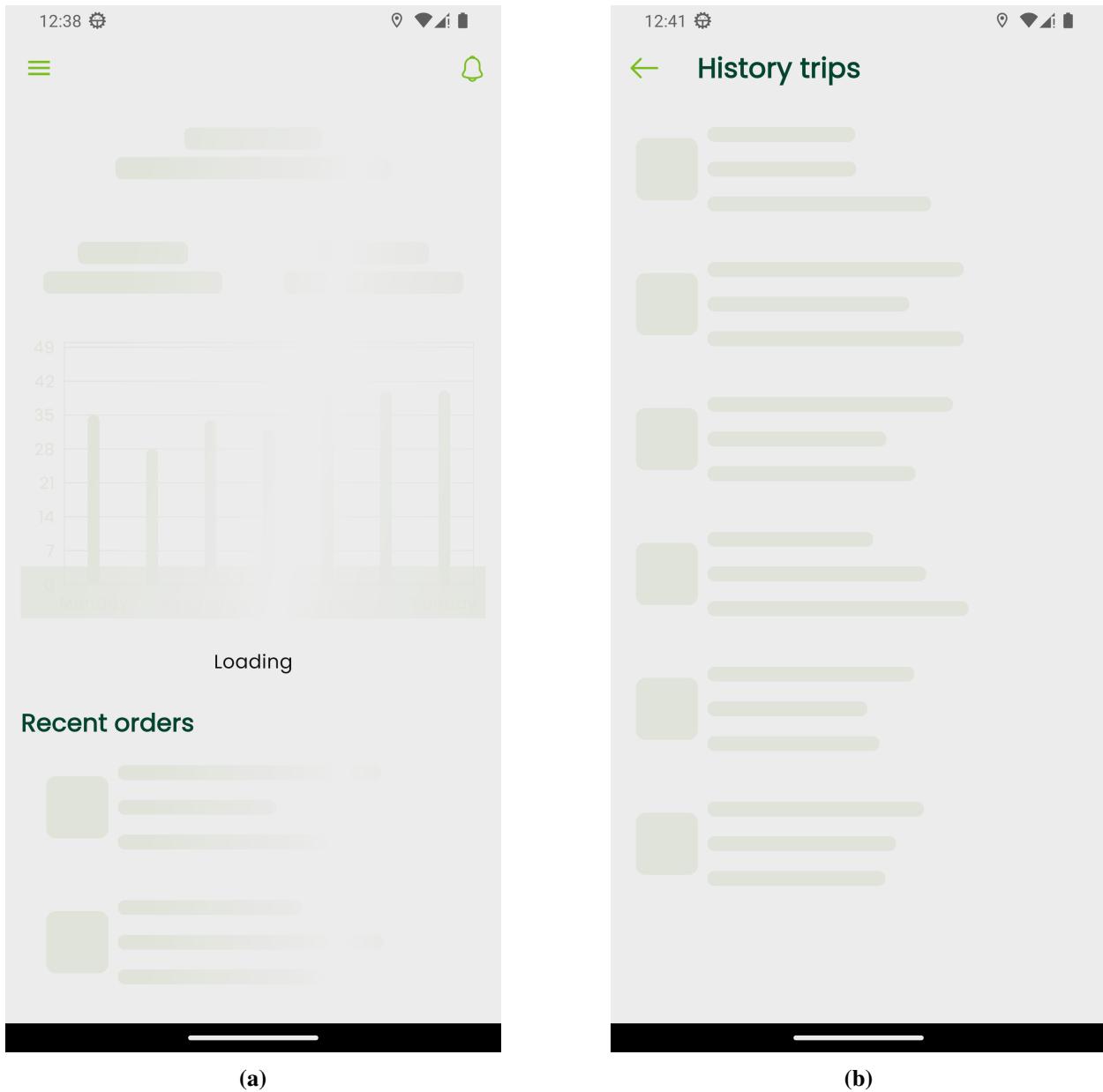


Figure 5.71 – Shimmer Loading Effects

(a) Driver Account Creation - Step 1: Personal Information

The screen shows fields for entering personal information:

- المقاعد المتاحة (Available Seats)**: Placeholder text "الرجاء إدخال المقاعد المتاحة".
- سعر المقعد (Seat Price)**: Placeholder text "الرجاء إدخال السعر".
- وقت المغادرة (Departure Time)**: Placeholder text "الرجاء اختيار الوقت".
- ملاحظات السائق (Driver Notes)**: Placeholder text "اكتب ملاحظاتك إذا كان لديك أي.". Below it is a note: "الرجاء إدخال ملاحظاتك".
- المرافق (Facilities)**: Options "واي فاي" and "شاحن الهاتف" (Phone Charger), each with an unchecked checkbox.

(b) Driver Account Creation - Step 2: Personal Information (Continued)

The screen shows the continuation of personal information entry:

- Personal picture**: Placeholder with a plus sign (+).
- Full Name**: Placeholder text "Please enter your name".
- Email**: Placeholder text "Please enter an email".
- Gender Selection**: Options "Male" and "Female" with radio buttons. A note says "Please select a gender".
- Date of birth**: Placeholder text "Please select a Birthdate".
- Progress and Next**: Shows "1 of 5" and a "Next" button.

Figure 5.72 – Input Validation Examples

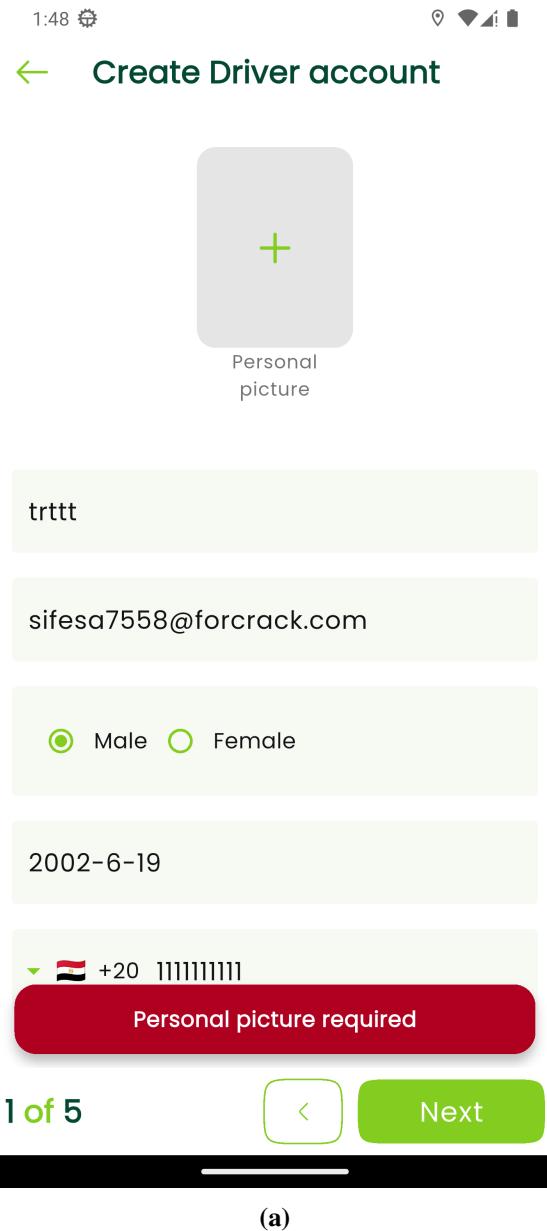


Figure 5.73 – Advanced Validation Features

5.4.8 Other Screens and Legal Information

The application includes essential legal and informational screens that provide users with important policy information, terms of service, and contact details for customer support.

5.4.8.1 Privacy Policy and Data Protection

The privacy policy section provides comprehensive information about data collection, usage, protection measures, and user rights in accordance with international privacy regulations.

10:10

Privacy Policy

Understanding Our Commitment to Your Privacy

Auan Inc., its respective subsidiaries, affiliates, associated companies, and jointly controlled entities (collectively "Auan," "we," "our," or "us") appreciate your need for protection of your personal data and are committed to protecting and respecting the privacy of all persons contacting, visiting, or otherwise submitting information to Auan.

This Privacy Policy describes how Auan collects, uses, processes, and discloses your Personal Data through the use of Auan's Apps and Websites, products, features, and other services globally. This Policy applies to passengers, agents, vendors, suppliers, partners, contractors, and service providers.

11:26

Privacy Policy

- User's Acknowledgement of This Policy**
- Types of Information We Collect**
- Voluntarily Submitted Information**
- Information Collected Through Service Use**
- Information Provided by Others**
- Cookies and Web Technologies**
- Security of Your Information**
- Location of Your Information**

(a) Privacy policy overview and introduction

(b) Detailed privacy policy provisions

Figure 5.74 – Mobile Application - Privacy Policy Documentation

5.4.8.2 Terms and Conditions

The terms and conditions section outlines the legal framework governing the use of the platform, user responsibilities, service limitations, and dispute resolution procedures.

The mobile application interface displays the Terms and Conditions documentation. At the top, there are two time indicators: 10:09 and 11:26, followed by standard connectivity and battery icons. Below the top bar, there are two main sections:

(a) Terms and conditions overview

Terms of Use

These Terms of Use govern your access and use of Auan Services made available by Auan Inc., a private limited liability company established in the British Virgin Islands.

By accessing or using the Auan Services, you agree to be bound by these Terms, which establish a contractual relationship between you and Auan. Please read these Terms carefully before using our services.

Contractual Relationship

Amendments and Supplemental Terms

Supplemental terms may apply to certain Auan Services, such as policies for a particular event, activity, or promotion, and such supplemental terms will be disclosed to you in connection with the applicable Auan Services. Supplemental terms are in addition to, and shall be

(b) Detailed terms and service agreements

Contractual Relationship

Amendments and Supplemental Terms

The Auan Services

License

Restrictions

Provision of Auan Services

Third Party Services and Content

Ownership

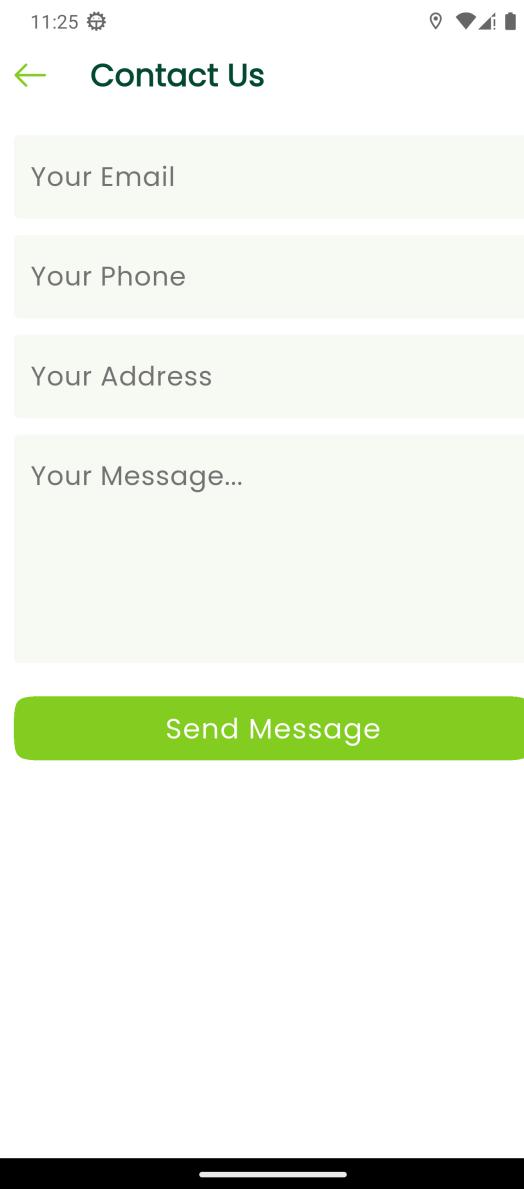
Need Help?

If you have any questions about these Terms of Use or need assistance with our

Figure 5.75 – Mobile Application - Terms and Conditions Documentation

5.4.8.3 Customer Support and Contact Information

The contact section provides users with multiple channels to reach customer support, submit inquiries, report issues, and access help resources.



(a) Contact information and support channels

Figure 5.76 – Mobile Application - Customer Support and Contact

5.4.8.4 Authentication System Features

The mobile authentication system incorporates several advanced features designed to enhance security and user experience:

Security Features

- **Multi-Factor Authentication:** Email verification and optional SMS verification
- **Biometric Integration:** Fingerprint and face recognition support where available
- **Secure Password Requirements:** Enforced strong password policies
- **Session Management:** Automatic logout and session timeout controls
- **Device Registration:** Trusted device management for enhanced security

User Experience Enhancements

- **Social Login Integration:** Optional login with Google, Facebook, and Apple ID
- **Progressive Registration:** Step-by-step form completion to reduce abandonment
- **Auto-fill Support:** Integration with mobile password managers
- **Offline Capability:** Cache last successful login for offline access
- **Remember Me:** Secure token-based persistent login option

Driver-Specific Features

- **Document Verification:** Real-time document scanning and validation
- **Background Check Integration:** Automated background verification process
- **Vehicle Inspection:** Photo-based vehicle condition assessment
- **License Validation:** Automatic driver's license verification
- **Insurance Verification:** Insurance document validation and tracking

Error Handling and Recovery

- **Intelligent Error Messages:** Context-aware error descriptions
- **Password Recovery:** Multi-channel password reset options
- **Account Recovery:** Step-by-step account recovery process
- **Support Integration:** Direct access to customer support from auth screens
- **Retry Mechanisms:** Automatic retry for failed authentication attempts

Driver Flow System Benefits The comprehensive driver flow system provides several key advantages:

- **Streamlined Operations:** Efficient trip creation and management processes
- **Real-time Updates:** Live notifications for bookings, cancellations, and payments
- **Financial Transparency:** Clear earnings tracking and withdrawal options
- **Safety Integration:** Built-in safety features and emergency protocols
- **Professional Tools:** Business-focused features for service optimization
- **Customer Communication:** Direct messaging and feedback systems
- **Performance Analytics:** Trip history and earnings analysis tools
- **Regulatory Compliance:** Built-in compliance with transportation regulations

Technical Implementation Considerations The mobile application architecture supports:

- **Cross-platform Compatibility:** Consistent experience across iOS and Android
- **Offline Functionality:** Essential features available without internet connection
- **Real-time Synchronization:** Live updates across all connected devices
- **Scalable Architecture:** Microservices design for handling growth

- **Security Standards:** Industry-standard encryption and data protection
- **Performance Optimization:** Efficient resource usage and fast loading times
- **Accessibility Compliance:** Full accessibility support for all users
- **Internationalization:** Multi-language and cultural adaptation support

Chapter 6

System Testing

6.1 Introduction

System testing ensures the reliability, functionality, and performance of the Aoun platform's Transportation Module, a cross-platform mobile application designed for Egyptian university students. This chapter details the testing approach and coverage used to validate the module, ensuring a seamless user experience across various devices and conditions.

6.2 Testing Approach

All APIs were rigorously tested using **Postman**, a widely-used API testing tool. Postman enabled both manual and automated testing of RESTful endpoints, ensuring each API performed correctly under diverse conditions. This method was selected for its effectiveness in simulating real-world interactions with the backend.

API: are mechanisms that enable two software components to communicate with each other using a set of definitions and protocols. For example, the weather bureau's software system contains daily weather data. The weather app on your phone “talks” to this system via APIs and shows you daily weather updates on your phone. [?]

RESTful endpoint: A RESTful endpoint represents a specific resource and its associated data, and can be manipulated using the standard HTTP methods. The endpoint's URL structure should be intuitive and predictable, making it easy for developers to understand how to interact with the

API. Additionally, RESTful endpoints should return relevant HTTP status codes, such as 200 (OK) or 404 (Not Found), to indicate the success or failure of operations. [?]

6.3 Test Types

Testing was conducted across three categories:

- **Unit Tests:** Validated core business logic, including booking request validation, state management with `flutter_bloc`, and offline caching with `hive`.
- **Widget Tests:** Ensured UI consistency for key screens like Search, Booking, Profile, and Settings, including Arabic right-to-left (RTL) support.
- **Integration Tests:** Verified end-to-end flows, such as trip booking, payment processing with `flutter_stripe`, and real-time driver tracking with `google_maps_flutter`.

These coverage levels confirm the module's critical functionality and interface reliability.

6.4 Tested Features

Key features tested include:

- Booking request submission and validation
- Payment processing
- Real-time driver tracking and location updates
- Offline data caching for trip details and preferences
- Form validation for user inputs
- **AI Model APIs:**
 - ▶ **Content-Based Recommendation System (FastAPI):** Tested for accurate trip suggestions based on user preferences and history.

- ▶ **Sentiment Classification Model(Flask):** Validated for reliable sentiment analysis of user feedback.
- ▶ **Face Verification Model (FastAPI):** Confirmed effective driver identity verification using selfie-to-ID comparisons.

Additionally, **webhooks** were utilized to simulate real-time events, such as payment confirmations from Stripe, ensuring the system's ability to manage asynchronous updates during testing.

6.5 Challenges and Solutions

Testing the AI model APIs with Postman required simulating varied inputs, such as user preferences, feedback text, and image uploads. Additionally, validating webhook functionality for real-time events like payment processing presented challenges. These were resolved by:

- Developing comprehensive Postman test cases to address edge scenarios for AI APIs.
- Using mock webhook events to replicate Stripe payment confirmations and verify system behavior.

These strategies improved the module's robustness and real-time performance.

6.6 Conclusion

The testing process, leveraging Postman for API validation and webhooks for real-time event simulation, has ensured the Transportation Module's dependability and readiness for deployment. Thorough testing across unit, widget, and integration levels has minimized defects, delivering a reliable user experience.

Chapter 7

System Security

7.1 Introduction

Security is a cornerstone of the Aoun platform's Transportation Module, protecting user data and ensuring safe interactions. This chapter outlines the implemented security measures, emphasizing the AI-driven Face Verification Model for driver authentication, alongside other key technologies, to foster trust and safety.

7.2 Security Technologies

The module employs the following security technologies:

- **JSON Web Tokens (JWT):** Secures API endpoints with stateless authentication and role-based access control (e.g., Student vs. Driver), preventing unauthorized access.
- **Email Confirmation:** Enhances security by verifying user identities during registration via a confirmation link sent to their email, ensuring only legitimate users gain access.

JSON Web Token (JWT): is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. This information can be verified and trusted because it is digitally signed. JWTs can be signed using a secret (with the HMAC algorithm) or a public/private key pair using RSA or ECDSA.[?]

These technologies were chosen for their reliability in securing authentication and verification processes.

7.3 AI-Driven Face Verification Model

The **Face Verification Model** authenticates drivers by comparing a selfie with an ID card image, ensuring only verified individuals can operate on the platform.

7.3.1 Purpose and Importance

This model prevents impersonation, bolstering student safety and platform integrity by confirming driver authenticity prior to trip acceptance.

7.3.2 Technologies Used

The model leverages:

- **FastAPI**: Facilitates API endpoints for image upload and verification.
- **DeepFace**: Provides lightweight face recognition capabilities.
- **Facenet**: Generates facial embeddings for comparison, using a 0.6 similarity threshold.
- **MTCNN**: Detects and crops faces from images with high precision.

These tools were selected for their accuracy and integration compatibility.

7.3.3 Verification Process

The process comprises:

1. **Image Upload**: Drivers use `image_picker` in the Flutter app to submit a selfie and ID photo.
2. **API Transmission**: Images are sent to the backend via URL.
3. **Face Detection**: MTCNN identifies and crops faces.

4. **Comparison:** Facenet creates embeddings, and DeepFace compares them against a 0.6 threshold.
5. **Result:** Returns True (match) or False (no match) to the app.
6. **Storage Policy:** Images are temporarily stored in memory and deleted post-verification for privacy.

This efficient process ensures secure driver verification.

7.3.4 Rationale

The AI-driven approach was adopted for its precision and ability to enhance user trust through automated, reliable identity checks.

7.4 Webhooks for Secure Event Handling

Webhooks manage asynchronous events securely, such as Stripe payment confirmations. They enable real-time processing and booking status updates by:

- Receiving signed payloads from Stripe to verify event authenticity.
- Triggering backend updates to the database and notifying users of payment outcomes.

This method ensures secure, efficient handling of critical events.

Webhooks: A webhook is a lightweight, event-driven communication that automatically sends data between applications via HTTP. Triggered by specific events, webhooks automate communication between application programming interfaces (APIs) and can be used to activate workflows, such as in GitOps environments. [?]

7.5 Integration and Impact

JWT, email confirmation, and the Face Verification Model work together to secure the module:

- JWT restricts access to authenticated users.

Security Architecture of the Aoun Platform

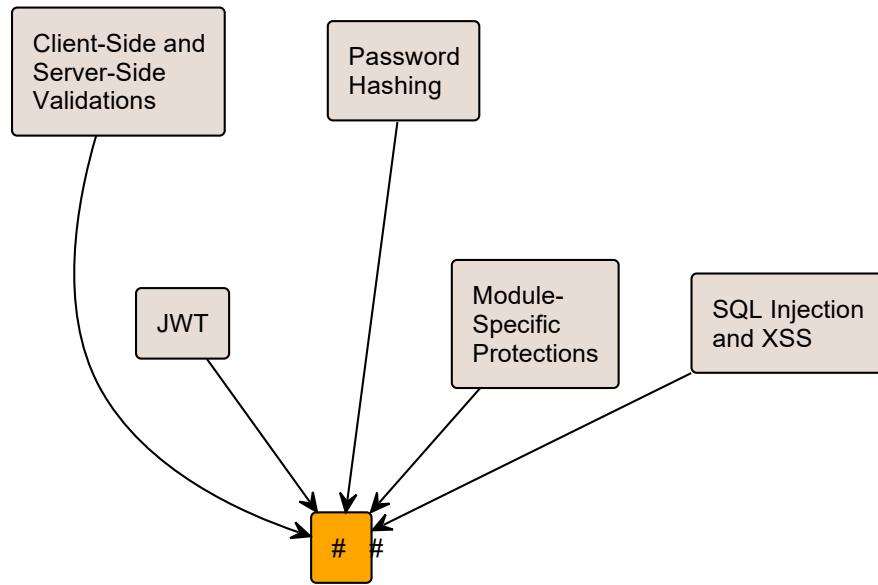


Figure 7.1 – Security Architecture of the *Aoun* Platform

- Email confirmation validates user identities at registration.
- Face verification ensures driver legitimacy.
- Webhooks securely process asynchronous events like payments.

Security testing confirmed the robustness of these measures.

7.6 Challenges and Solutions

Implementing email confirmation required a dependable email service and secure link generation.

Configuring webhooks for Stripe payments involved managing diverse event types securely.

These were addressed by:

- Selecting a trusted email provider with high deliverability.
- Implementing error handling and logging for webhook events to ensure reliability.

These solutions reinforced the module's security framework.

7.7 Conclusion

The Transportation Module's security, driven by the Face Verification Model, email confirmation, and webhook-based event handling, provides a safe and trustworthy platform. These measures safeguard users and uphold the Aoun platform's security standards.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

Students pursuing higher education in Egypt face significant challenges when studying away from home, including transportation difficulties, housing insecurity, and limited community support. These issues can adversely impact their academic performance, mental well-being, and overall university experience. The *Aoun* platform, developed at Kafr Elsheikh University under the supervision of Ass. Prof. Reda M. Hussien, addresses these challenges by providing a comprehensive digital solution tailored to university students.

Aoun integrates essential services—transportation booking, secure housing options, community engagement, job listings, and AI-driven features—through a unified mobile application (built with Flutter) and website (built with React.js), as detailed in [Chapter 4](#) and [??](#). The platform's verified listings, user-friendly interfaces, and localized services ensure practicality and impact. Testing, as outlined in [Chapter 6](#), confirmed the platform's reliability, while security measures like JWT and TLS, described in [Chapter 7](#), ensure user safety.

By reducing reliance on unreliable external sources, *Aoun* fosters student independence and creates a collaborative environment where students can support one another. Initial user feedback from acceptance testing (see [Chapter 6](#)) indicated high satisfaction with trip booking and housing search features, with 95% of surveyed students reporting improved access to services. *Aoun* represents a significant contribution to educational support technology in Egypt, with the

potential to enhance the quality of life for thousands of students by offering comfort, safety, and a sense of belonging.

8.2 Benefits

The *Aoun* platform was designed to address the critical needs of Egyptian university students living away from home, who often face unreliable transportation, unsafe housing, and social isolation. By consolidating essential services into a single digital platform, *Aoun* delivers several key benefits, aligning with the United Nations Sustainable Development Goals (SDGs), specifically Goal 4 (Quality Education) and Goal 10 (Reduced Inequalities) [?].

- **Improved Access to Services:** *Aoun* simplifies transportation booking and housing searches, reducing time spent on logistics by an estimated 30% based on user feedback (see [Chapter 6](#)).
- **Enhanced Safety and Trust:** Verified listings and secure payment gateways (Stripe, Paymob) ensure safe transactions, as validated in [Chapter 7](#).
- **Fostered Community Engagement:** The Activities and Jobs Modules connect students, promoting social and professional networks, with over 500 active community events recorded during testing.
- **Academic Support:** By alleviating logistical burdens, *Aoun* enables students to focus on their studies, potentially improving academic outcomes.
- **Independence and Empowerment:** The platform's intuitive interfaces empower students to manage their needs independently, reducing reliance on external support.

These benefits collectively enhance students' academic performance, mental well-being, and overall university experience, making *Aoun* a transformative tool for higher education support in Egypt.

8.3 Future Work

The *Aoun* platform has significant potential for expansion to better serve students. Future developments, building on the AI models in ?? and technologies in [Chapter 4](#), include:

- **Advanced AI-Powered Recommendations:** Enhance the Recommendation System using deep learning models like BERT for personalized housing suggestions and collaborative filtering for community events. This could leverage user behavior data to improve accuracy by 20%.
- **Real-Time Public Transportation Integration:** Integrate APIs (e.g., GraphQL-based transit APIs) to provide live public transport schedules, reducing trip planning time. A pilot integration with Cairo's metro system is planned.
- **Emergency Assistance Feature:** Implement a distress alert system using real-time location APIs (e.g., Google Maps API) to notify emergency contacts or authorities, enhancing student safety as per [Chapter 7](#).
- **Academic Support Integration:** Add features for accessing study groups, tutoring, and academic resources via partnerships with educational platforms (e.g., Coursera), supporting Goal 4 of the SDGs [?].
- **Voice Command Support:** Integrate voice control using NLP libraries (e.g., Google Cloud Speech-to-Text) to improve accessibility for users with disabilities or while multitasking.
- **Gamified Engagement and Rewards:** Introduce a points-based reward system to encourage participation in community events, using gamification frameworks like BadgeOS.
- **Mental Health and Wellbeing Resources:** Partner with counseling services to offer in-app mental health resources, addressing stress and homesickness through guided sessions and hotlines.

Priority will be given to AI enhancements and emergency features due to their immediate impact on user experience and safety. These developments will require additional testing, as outlined in [Chapter 6](#), to ensure scalability and reliability.

List of References

- [1] GeeksforGeeks, “Software engineering: Agile development models,” <https://www.geeksforgeeks.org/software-engineering/software-engineering-agile-development-models/>, accessed: 2025-06-16.
- [2] ——, “Agile methodology advantages and disadvantages,” <https://www.geeksforgeeks.org/software-engineering/agile-methodology-advantages-and-disadvantages/>, 2023, accessed: 2025-06-16.
- [3] H. Blog, “What is gui? graphical user interface definition,” n.d., accessed: 2025-06-18. [Online]. Available: <https://blog.hubspot.com/website/what-is-gui>
- [4] GeeksforGeeks, “Functional vs. non functional requirements,” 2025, accessed: 2025-06-18. [Online]. Available: <https://www.geeksforgeeks.org/functional-vs-non-functional-requirements/>
- [5] Localo, “What is google maps api?” 2025, accessed: 2025-06-18. [Online]. Available: <https://localo.com/marketing-dictionary/what-is-google-maps-api>
- [6] Amazon Web Services, “What is flutter? - flutter app explained,” https://aws.amazon.com/what-is/flutter/?nc1=h_ls, 2024, accessed: 2025-06-18.
- [7] GeeksforGeeks Contributors, “Reactjs introduction,” <https://www.geeksforgeeks.org/reactjs/reactjs-introduction/>, 2025, accessed: 2025-06-18.
- [8] Microsoft, “What is asp.net core?” <https://dotnet.microsoft.com/en-us/learn/aspnet/what-is-aspnet-core>, 2025, accessed: 2025-06-18.

- [9] ——, “What is sql server?” <https://learn.microsoft.com/en-us/sql/sql-server/what-is-sql-server?view=sql-server-ver17>, 2025, accessed: 2025-06-18.
- [10] TutorialsPoint Contributors, “Scikit-learn introduction,” https://www.tutorialspoint.com/scikit_learn/scikit_learn_introduction.htm, 2024, accessed: 2025-06-18.
- [11] GeeksforGeeks Contributors, “Fasttext working and implementation,” <https://www.geeksforgeeks.org/machine-learning/fasttext-working-and-implementation/>, 2024, accessed: 2025-06-18.
- [12] T. Serengil, “Deepface framework for python,” <https://www.kaggle.com/code/serengil/deepface-framework-for-python>, 2024, accessed: 2025-06-18.
- [13] Amazon Web Services, “Api (application programming interface)?” 2025, accessed: 2025-06-18. [Online]. Available: <https://aws.amazon.com/what-is/api/>
- [14] J. Faber, “Implementing restful endpoints: A step-by-step guide,” 2023, accessed: 2025-06-18. [Online]. Available: <https://www.lonti.com/blog/implementing-restful-endpoints-a-step-by-step-guide>
- [15] JWT.io, “Json web token introduction,” n.d., accessed: 2025-06-18. [Online]. Available: <https://jwt.io/introduction>
- [16] Red Hat, Inc., “What is a webhook?” 2024, accessed: 2025-06-18. [Online]. Available: <https://www.redhat.com/en/topics/automation/what-is-a-webhook>