

Kingdom of Saudi Arabia
Ministry of Education
Imam Abdulrahman bin Faisal University
College of Science and Humanities-Jubail
Computer science Department



Software Engineering
Term 1 – 2024/2025

Software Design Specifications

For



Version 1

CS 411 – Software Engineering
Group 2

Subervised by:

Dr. Enas El-sayed El-sharawy

15 November 2025

This Software Design Specification was prepared and provided as a deliverable for [Software Engineering, CS 411, 2025-2026], and it will be used by [CS students].

This document is based in part on the IEEE Recommended Practice for Software Design Descriptions.

#	Name	ID	Role
1	Jood Mutlaq Abaqmi	2230000478	Leader
2	Nada Ali Alshamsi	2230001749	Member
3	Jana Fahad Alharbi	2230004859	Member
4	Zahraa Mohammed Alqattan	2230006222	Member
5	Batool Sadiq Alsharqi	2230000960	Member
6	Kadijah Yasser Al bader	2230006906	Member
7	Roaa abbas Alhaddad	2230000387	Member
8	Shaden Salman Alghamdi	2230003568	Member

Table of Content

Revision History	5
1. Introduction.....	8
1.1 Purpose.....	8
1.2 Scope	8
1.3 Definitions, Acronyms, and Abbreviations.....	9
1.4 References	10
2. System overview.....	10
3. Design Considerations	11
3.1 Assumptions and Dependencies.....	11
3.2 General Constraints	12
4. User Interface Design	13
4.1 Overview of User Interface.....	13
4.2 Interface Design Rules	15
4.3 Screen Images	16
4.3.1 Homepage Interface.....	16
4.4 Screen Objects and Actions.....	17
4.5 Other Interfaces	23
4.5.1 Error Messages	23
4.5.1.1 Sign Up Error	23
4.5.1.2 User Type Selection Error.....	24
4.5.1.3 Incorrect OTP Error Message.....	25
4.5.1.4 No Results Found.....	26
4.5.1.5 This car cannot be rebooked now.....	27
4.5.1.6 This Field required.....	28
4.5.1.7 Invalid Plate number format.....	29
4.5.1.8 Action Failed.....	30
4.5.1.9 Unable to load Earnings.....	31
4.5.1.10 Approval Requires.....	32
4.5.1.11 Please select a rating before submitting.....	33
5. System Architecture	34
5.1 Architectural Design Approach.....	34
5.2 Architectural Design	34
5.3 Subsystem Architecture	35
6. Data Design.....	38
6.1 Data Description	38
6.2 Data Dictionary	42
6.3 Database Description	44
7. Component Design.....	46
7.1 Common Functions.....	46
7.1.1 User Sign In.....	46
7.1.2 Identity Verification.....	46

7.1.3 Password Recovery.....	47
7.2 Renter Functions.....	47
7.2.1 Search Available Vehicle.....	47
7.2.2 Create Booking Request.....	48
7.2.3 Process Payment.....	48
7.3 Owner Functions.....	49
7.3.1 Add Vehicle Listing.....	49
7.3.2 Manage Booking Requests.....	49
7.3.3 Update Vehicle Availability	50
7.4 Admin Functions.....	50
7.4.1 User Management.....	51
7.4.2 System Monitoring.....	51
7.4.3 Content Moderation.....	52
7.4.4 Financial Reporting.....	53
8. Detailed System Design	54
8.1 Classification.....	54
8.2 Definition	57
8.3 Responsibilities	57
8.4 Constraints	58
8.5 Composition	61
8.6 Uses/Interactions	61
8.7 Resources	67
8.8 Processing	68
8.9 Interface/Exports	69
8.10 Detailed Subsystem Design	69
8.11 Detailed System Interfaces.....	70
8.11.1 Common Interfaces.....	70
8.11.1.1 Sign\Login Interface.....	70
8.11.1.2 Identity Verification Interface.....	71
8.11.1.3 Forget My Password Interface.....	72
8.11.2 Renter Interfaces.....	73
8.11.2.1 Renter Homepage Interface.....	73
8.11.2.2 Car Search & Discovery Interface.....	74
8.11.2.3 Car Details Interface.....	75
8.11.2.4 Booking & Payment Interface.....	76
8.11.2.5 Booking History Interface.....	77
8.11.3 Owner Interfaces.....	78
8.11.3.1 Owner Dashboard Interface.....	78
8.11.3.2 Car Listing Management Interface.....	79
8.11.3.3 Booking Requests Interface.....	80
8.11.3.4 Earnings & Transaction History Interface.....	81
8.11.4 Admin Interfaces.....	82
8.11.4.1 Admin Dashboard.....	82
8.11.4.2 User Management Interface.....	83
8.11.4.3 Content Moderation Interface.....	84
8.11.4.4 Report & Analytics Interface.....	85

8.11.5 Rating & Review Interfaces.....	86
8.11.5.1 Rate Experience Interface.....	86
9. Other Design Features.....	87
10. Requirements Traceability Matrix	87

Revision History

Name	Date	Reason For Changes	Version
All members	Nov 11, 2025	Prepared initial version	0.1
All members	Nov 14, 2025	Updated section 1	0.2
All members	Nov 16, 2025	Updated section 3	0.3
All members	Nov 17, 2025	Updated section 5	0.4
All members	Nov 18, 2025	Updated section 6	0.5
All members	Nov 19, 2025	Updated section 7	0.6
All members	Nov 20, 2025	Updated section 8.11	0.7
All members	Nov 23, 2025	Update section 1.3	0.8
All members	Nov 23, 2025	Update section 1.4	0.9
All members	Nov 24, 2025	Complete review - Final version	1.0

Table of Tables

Table 1 - Definition, Acronyms and Abbreviations.....	10
Table 2 - End-user characteristics.....	12
Table 3 - Screen Objects and actions.....	18
Table 4 - Data Description.....	39
Table 5 - Data Dictionary.....	43
Table 6 - Renter Component.....	55
Table 7 - Owner Component.....	56
Table 8 - Admin Component.....	57
Table 9 – Definition.....	58
Table 10 – Responsibilities.....	58
Table 11 - User Authentication Constraints.....	59
Table 12 - Car Browsing & Booking Component Constraint.....	59
Table 13 - Car Listing & Booking Management constraints.....	60
Table 14 - System Supervision constraint.....	61
Table 15 - System-Level Interfaces constraints.....	61
Table 16 - Composition.....	62
Table 17 - system resources.....	68
Table 18 - possible race conditions.....	68
Table 19 - potential deadlock.....	68
Table 20 – Requirements Traceability Matrix.....	88

1. Introduction

This Software Design Specification (SDS) document describes the complete design of the “Fadl Zahr” mobile application. It translates the software requirements defined in the Software Requirements Specification (SRS) into an architectural and detailed design suitable for implementation. This document explains how the system will be structured, how its components interact, and how data will flow throughout the application. It follows the IEEE recommended structure for software design documentation.

The document includes the purpose and scope of the SDS, definitions and acronyms, references to related documents, and a full overview of the system's architecture, interface design, data structure, subsystem definitions, and component-level design. It is intended to serve as a blueprint for system developers, testers, and maintainers.

1.1 Purpose

design for the “Fadl Zahr” system based on the functional and non-functional requirements defined in the SRS. It aims to guide developers in implementing the system accurately, ensuring that all design decisions support the system’s goals, maintain system integrity, and allow for efficient development and future enhancements. This document also supports testers by mapping design elements to the requirements, enabling effective validation and verification

1.2 Scope

This SDS describes the full design of the “Fadl Zahr” car-rental mobile application (Version 1.0), which runs on Android and iOS platforms and includes an online administration dashboard. It covers the system architecture, data design, user interface design, subsystem interactions, and component-level behavior. The document defines how the system will be built and how each major module will contribute to system functionality.

Intended Audience:

The Development Team: By following the requirements described in this document, the team can design and implement the system according to the defined goals and functionality

The Project Supervisor: The supervisor will ensure that the project aligns with its objectives and timeline, provide feedback, and suggest modifications when needed.

The Users (Clients): Users will review whether the system meets their needs and expectations, confirm that the features function as planned, and suggest improvements for better usability.

1.3 Definitions, Acronyms, and Abbreviations

Term / Acronym	Definition
SDS	Software Design Specifications. A document that describes the architecture, components, interfaces, and other characteristics of a software system, serving as a blueprint for development.
SRS	Software Requirements Specification. A document that describes the functional and non-functional requirements of a software system, serving as a foundation for design and development.
UI	User Interface. The means by which a user interacts with a software application, including screens, pages, and visual elements like buttons and icons.
API	Application Programming Interface. A set of defined rules and protocols that allow different software applications to communicate with each other.
OTP	One-Time Password. A dynamically generated, single-use code sent to a user for authentication or verification purposes.
JWT	JSON Web Token. An open standard for securely transmitting information between parties as a compact, self-contained JSON object.
Firebase	A Backend-as-a-Service (BaaS) platform provided by Google, used for developing mobile and web applications, offering services like authentication, a real-time database, and cloud storage.

Absher	A Saudi governmental electronic platform that provides various services to citizens, residents, and businesses. It is integrated into the system for official identity verification.
Mada	The national debit card scheme and electronic payment network of the Kingdom of Saudi Arabia.
STC Pay	A digital wallet and electronic payment service offered by STC Group in Saudi Arabia.
IBAN	International Bank Account Number. An internationally agreed system of identifying bank accounts across national borders to facilitate the processing of cross-border transactions.
PDF	Portable Document Format. A file format developed by Adobe to present documents, including text formatting and images, in a manner independent of application software, hardware, and operating systems.
CSV	Portable Document Format. A file format developed by Adobe to present documents, including text formatting and images, in a manner independent of application software, hardware, and operating systems.
JSON	JavaScript Object Notation. A lightweight, text-based, language-independent data interchange format that is easy for humans to read and write and for machines to parse and generate.
FCM	Firebase Cloud Messaging. A cross-platform messaging solution provided by Google that allows for reliable and efficient delivery of messages and notifications at no cost.

ORM	Object-Relational Mapping. A programming technique for converting data between incompatible type systems, such as between objects in an application and tables in a relational database.
Admin	Administrator. A user role with the highest level of privileges, responsible for system monitoring, user management, content moderation, and overall platform governance.
Owner	Car Owner. A user role representing an individual who lists their personal vehicle(s) for rent on the platform.
Renter	A user role representing an individual who searches for books, and rents vehicles listed on the platform.
System	The entire "Fadl Zahr" software ecosystem, including the mobile applications, web dashboard, backend servers, databases, and integrated third-party services.
User	A general term for any individual or system actor that interacts with the "Fadl Zahr" application, typically categorized as a Renter, Owner, or Admin.
Functional Requirement	A statement of what a software system must do, describing its behavior and functions (e.g., "The system shall allow users to reset their password").
Non-Functional Requirement	A criterion that specifies the quality attributes of a system, such as performance, security, usability, and reliability, rather than its specific behaviors.

Module	A self-contained unit of software that handles a specific functionality and can be independently developed and tested (e.g., Payment Module, Booking Management Module).
Interface	A shared boundary across which two or more separate components of a system exchange information. This can be a UI for users or an API for systems.
Database	An organized collection of structured data, typically stored electronically in a computer system, managed by a Database Management System (DBMS).
UML	Unified Modeling Language. A standardized, general-purpose modeling language in the field of software engineering used to visualize the design of a system.
Actor	An entity, whether a user or an external system, that interacts with the system to achieve a goal. In "Fadl Zahr," primary actors are Renter, Owner, and Admin.
Use case	A description of a sequence of interactions between an actor and the system to achieve a specific goal. It defines the system's behavior under various conditions.
Client	The front-end part of the application that runs on the user's device (e.g., the mobile app or web browser) and requests services from the server.
Server	The back-end part of the application that resides on a remote computer, providing resources, data, and services to the client upon request.

Deployment	The process of making a software application available for use, typically by installing, configuring, and running it on a server or distributing it through an app store.
Software Architecture	The fundamental structures of a software system, including its components, their relationships, and the principles and guidelines governing its design and evolution.

Table 1- Definition, Acronyms and Abbreviations

1.4 References

- [1] Lumi Rental. Lumi – Home. Website. Retrieved November 7, 2025, from lumirental.com.
- [2] Pressman, R. S., & Maxim, B. R. Software Engineering: A Practitioner’s Approach. 9th Edition. McGraw-Hill Education, 2020.
- [3] ScienceDirect. “Software Interface.” Retrieved November 4, 2025, from sciencedirect.com.
- [4] Saudi Data & Artificial Intelligence Authority (SDAIA). Data Protection Law (PDPL). Retrieved November 24, 2025, from sdaia.gov.sa.
- [5] Saudi Central Bank (SAMA). Implementing Regulations for the Law of Payments and Payment Services. 2020. Retrieved November 24, 2025, from sama.gov.sa.
- [6] Saudi Data & Artificial Intelligence Authority (SDAIA). Regulations and Policies. Retrieved November 24, 2025, from sdaia.gov.sa.
- [7] Al-Fedaghi, S. Modeling the Realization and Execution of Functions and Functional Requirements. arXiv, 2020. Retrieved November 24, 2025, from arxiv.org/abs/2004.00078.

2. System overview

The Fadl Zahr application is a mobile-based car rental platform that seamlessly connects car owners with individuals seeking to rent vehicles. The system provides a secure, user-friendly environment that supports identity verification, digital payments, car listing management, and booking operations. The application aims to simplify the rental process by offering real-time communication, transparent pricing, and an automated workflow that enhances trust between renters and car owners.

The system includes the following major components:

- User authentication and identity verification through Absher.
- A car listing module enabling owners to add, modify, or remove vehicles.
- A booking and transaction workflow allowing renters to search, reserve, and pay securely.
- A communication and feedback system to improve user experience and service quality.
- An administrative control panel for platform governance, content management, and security monitoring.
- A notification service ensuring timely updates about bookings, payments, and system alerts.

The app ensures data protection, applies strict privacy and security protocols, and supports automated invoice generation, tax calculation, and role-based access control.

3. Design Considerations

This section outlines the most important factors, challenges, and considerations that must be upheld during the development of the “Fadl Zahr” car rental system. It covers assumptions about which the system is based, dependencies affecting workflow, and constraints that limit options for design.

3.1 Assumptions and Dependencies

The following are assumptions and dependencies pertinent to the development of the system, covering topics ranging from operating environment considerations to user characteristics:

- Related Software or Hardware:
 - The system relies on smartphones running either Android operating systems, version 7.0 and above, or iOS, version 11.0 and above. These devices must also be reasonably resourced in terms of memory and processing power to handle features related to car searching and booking.
 - Firebase is used for user authentication and real-time database management.
 - The Absher API is used for Identity Verification to confirm any user's application while guaranteeing security.
 - Payment services will be executed using external APIs like STC Pay or Mada for online transactions securely.
- Operating System:

- Designed and optimized for Android and iOS mobile devices, the application will work on both.
- End-user characteristics:
- Car owners and renters are expected to have basic mobile and internet usage skills. Renters are often non-technical and are mainly concerned with the ease of booking and payment. Car owners are familiar with vehicle rental processes.

Role	Age Range	Gender	Education Level	Tech Skills	Region
Admin	25–50	Male/Female	Bachelor's Degree	Excellent	Saudi Arabia
Owner	25–60	Male/Female	High School or Above	Good/Excellent	Saudi Arabia
Renter	18–50	Male/Female	Below High School	Fair/Good	Saudi Arabia

Table 2 - End-user characteristics

- Possible and/or Probable Changes in Functionality:
 - This includes UI updates, the addition of new payment methods, or expansion of vehicle categories, if needed, as the project is rolled out and based on feedback from users or market trends.

3.2 General Constraints

When designing the system, the following constraints should be considered:

- Hardware or Software Environment:
 - The application will run on mobile platforms and will require constant access to the internet. For the application to function well, especially for real-time processing of information (such as car booking and payment transactions), stable internet access is required.
- End-User Environment:
 - The system must be flexible enough to accommodate users from all walks of life. The interface should be simple and easy to use by both car owners and renters, depending on their level of

technological knowledge. The inclusion of multi-language support will also improve usage among Saudi Arabian residents.

- **Availability or Volatility of Resources:**

- Third-party service dependency: The system is dependent on third-party services like Absher for identity verification and the payment gateway. Any issue or change in these services could interfere with the functionality of the system. Additionally, security must be ensured for all data stored within the system to prevent loss or unauthorized alteration.

- **Standards Compliance:**

- The system must comply with local regulations concerning data protection and privacy. In Saudi Arabia, this includes adhering to personal data protection laws and ensuring that all transactions comply with SAMA guidelines regarding payment systems.

- **Interoperability Requirements:**

- Integrations with external services such as Absher and payment gateways should be seamless. Appropriate, secure protocols for data exchange are required.

- **Security Requirements:**

- Users' information, including personal and financial data, should always be encrypted for secure storage. Role-based access control will ensure that only authorized users can access sensitive functionalities.

- **Memory and Other Capacity Limitations:**

- The system should be optimized to run efficiently on devices with varying amounts of memory. It is critical that the application performs well even on devices with low amounts of storage.

- **Performance Requirements:**

- The system should, on average, respond to user queries (i.e., car search and bookings) within 2 seconds. This should be maintained even when up to 1,000 users are concurrently accessing the system.

- Verification and Validation Requirements:
- Comprehensive testing will be performed to ensure that all core functions bookings, payments, and identity verification function as expected and meet the needs of users.

4. User Interface Design

4.1 Overview of User Interface

The Fadl Zahr app is easy and friendly to use and designed for effortless communication between owners, renters, and the administering. When users first use the site will see the main homepage. The main interface shows two options “Renter” and “Car Owner/Admin”. Every choice sends the user to the login or access screen that suits them.

When the user clicks on the button “Car Owner/Admin”, the user is taken to the Sign In interface, which is common for car owners and administrators. From here, the users can log in to access their dashboards. Users can recover accounts using the “Forgot My Password” option without any difficulty. If the user chooses Renter, he is transferred to the Renter Interfaces, which has options for Sign Up (new user) and Sign In (returning user).

Different types of users have various privileges, services and authorizations. Thus, the interface changes:

Renter:

- Renter Homepage: Allows users to quickly search for listings, manage booked properties and access recommendations.
- Car Search & Discovery: Renters can search for cars using filters based on location, date, car type, and price.
- Car details: contain all the information about the car which includes the photos of the car, owner details, price and availability.
- To confirm booking details: select a method of payment and make payment can happen in different steps on the interface.
- Renters can view past and present booking history, along with rebooking and review leaving options.
- Chat with Owner: In-app messaging system that allows chat with owner directly.
- My Profile: gives renters access to update their personal information, order history, and sign out.

Car Owner:

- The Dashboard for Owners provides an overview of the earnings, ongoing bookings and recent reviews.

- Vehicle listing management allows the owner to add, modify or remove car listings along with adding images and pricing and availability.
- Booking Requests: Shows you incoming booking requests with options to approve or reject them.
- Earnings & Transaction History: Shows a summary of how much has been earned, and the transactions.
- Talk to the Renter: Communicate with your renters before or during the rental.

Administrator:

- Admin Dashboard: Centralized control panel for monitoring system activity, user accounts, and platform performance.
- User Management: Meticulously managing user accounts and verification status and permissions.
- Tools to review and moderate all user-generated content like reviews or car images.
- Reports & Analytics: You can use these reports for analyzing are use of system, revenue and user behavior.

4.2 Interface Design Rules

The user interface designs of Fadl Zahr follow Nielsen and Molich's 10 User Interface Design Guidelines to ensure a consistent and efficient experience for the user.

1. Visibility of System Status: Users should always be informed about what the system is doing via appropriate feedback (e.g., loading, verification).
2. System Should Correspond with Real World Contexts The interface should use familiar concepts, language and conventions e.g., calendar represents availability, star ratings are used for reviews.
3. User Control and Freedom: Users can easily execute an undo or a backward (e.g., back buttons, cancellation option while booking).
4. Consistency and Standards: Every interface should utilize consistent terminology, icons and layout patterns to avoid confusion.
5. Error Prevention: The system should be designed to minimize errors by clearly displaying labels, constraining options, and confirming critical actions.
6. Recognition Rather Than Recall: Users should reference any car rental information to avoid memorizing important information like date, type etc.
7. Adjustability and ease of utilization: The interface should provide shortcuts, searching filters, and personalized recommendations for both novice as well as expert users.

8. Aesthetic and Minimalist Design: User Interfaces (UI's) should be clean and tidy, only displaying what is needed.
9. Assist users in understanding, problem solving, and fixing mistakes. It's important that your error messages are clear, polite and suggest a solution. For example, "invalid date range please select a valid rental period" is a good error message.
10. Help and Documentation: Users should be able to get contextual help, tooltips and a user guide as needed.

4.3 Screen Images

4.3.1 Homepage Interface

This is the first screen shown to users when they open the *Fadl Zahr* application. It introduces the app and gives a quick overview of its purpose. The interface includes:

- App logo at the top for brand recognition.
- Welcome headline: "Let's Start a New Experience with Car Rental" to attract the user's attention.
- Background image of a car to reflect the app's service.
- Short description explaining that the app provides an easy and smooth car-rental experience.
- Progress indicator showing the user is on the first onboarding step.
- Get Started button that takes the user to the next interface

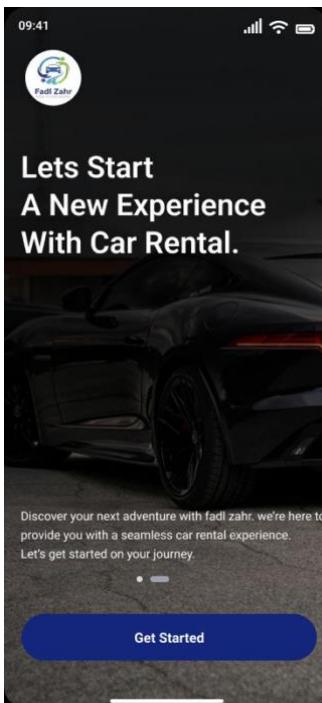


Figure 1 - Home page Interface

4.4 Screen Objects and Actions

Object	Type	Action
Main Homepage Interface		
App Logo	Image	Displays the application's branding at the top of the screen.
Intro Title	Text	Shows the main welcome message ("Let's Start A New Experience with Car Rental").
Background Image	Image	Provides a visual background related to car rental.
Description Text	Text	Gives a short introduction about the app and its purpose.
Onboarding Indicator	Indicator	Shows the user their current onboarding step.
Get Started Button	Button	Navigates the user to the next screen of the application.
Common Interface		
Sign Up Interface		
App Logo	Image	Displays the application's branding at the top of the screen.
Page Title (Sign Up)	Text	Shows the title of the screen and informs the user they are in the registration page.
Email / National ID Field	Input Text	Allow the user to enter their email or national ID.
Password Field	Input Text	Allow the user to enter a password for account creation.
Password Visibility Icon	Icon/Button	Toggles between showing and hiding the typed password.
User Type Dropdown	Dropdown Menu	Let's the user select whether they are a renter, car owner, or admin.
Sign In Button	Button	Create a new account and proceed to the next step.
Login Button	Button	Redirects the user to the login page if they already have an account.
Social Login (Apple)	Button	Allows the user to sign up using Apple credentials.
Social Login (Google)	Button	Allow the user to sign up using Google credentials.

“Already have an account?” Text	Text	Provides navigation awareness for the user.
Login Redirect Link	Text/Button	Sends the user to the login interface.
Log in Interface		
Email / National ID Field	Input Field	Allows the user to enter their email or national ID for login.
Password Field	Input Field	Allows the user to type their password.
Password Visibility Icon	Icon/Button	Shows or hides the entered password.
User Type Selector	Dropdown	Let's the user choose between Renter, Car Owner, or Admin.
Log in Button	Button	Submits the login information and proceeds to the next screen.
Forget Password Link	Text/Button	Redirects the user to the “Forget Password” process.
Apple Login Button	Button	Logs in using Apple account.
Google Login Button	Button	Logs in using Google account.
Sign Up Link	Text/Button	Takes the user to the Sign-Up interface.
Reset Your Password Interface		
Email Field	Input Field	Users enter the email associated with their account.
Return to Log in Link	Text/Button	Take the user back to the login screen.
Continue Button	Button	Proceeds to the verification code screen.
Create a New Account Link	Text/Button	Redirects user to the sign-up page.
Return to Log in Link	Text/Button	Take the user back to the login screen.
Enter Verification Code Interface		
OTP Input Boxes	Input Fields	User enters the 6-digit verification code.
Continue Button	Button	Confirms the verification code.
Resend OTP Link	Text/Button	Sends a new verification code.
Numeric Keypad	Buttons	Allows users to enter digits manually.
Delete Key	Button	Removes the last entered digit.
OTP Input Boxes	Input Fields	User enters the 6-digit verification code.
Reset Password Interface		
New Password Field	Input Field	User enters a new password.
Confirm Password Field	Input Field	User re-enters password for confirmation.
Continue Button	Button	Saves the new password and completes the process.
Cancel Button	Button	Cancel the reset and returns to the previous screen.

Renter Homepage Interface		
Back Icon	Button	Returns to the previous screen.
Notifications Icon	Button	Open the notifications page.
Search Bar	Input Field	Allows the user to search for a car.
Filter Icon	Button	Opens the car filter options.
View All (Recommended)	Text/Button	Shows all recommended cars.
Favorite Icon on Car Card	Button	Saves the car to favorites.
Book Now Button	Button	Goes to the car details/booking page.
Navbar – Home	Button	Navigates to the home screen.
Navbar – Search	Button	Opens the search interface.
Navbar – Messages	Button	Opens the messages page.
Navbar – Notifications	Button	Open the notifications page.
Navbar – Profile	Button	Opens the user profile.
Screen Objects and Actions		
Search 1		
Back Icon	Button	Returns to the previous screen.
More Options Icon	Button	Opens additional options menu.
Search Bar	Input Field	Allows the user to search for a car.
Filter Icon	Button	Opens the filter interface.
Categories (All, Ferrari, Tesla, BMW, Lamborghini)	Buttons	Filters cars by selected brand.
View All (Popular Cars)	Text/Button	Shows all popular cars.
Car Card	Button	Opens car details.
Favorite Icon	Button	Saves the car to favorites.
Navbar (Home, Search, Messages, Notifications, Profile)	Buttons	Navigates between main sections.
Search 2		
Close Icon	Button	Exits the filters panel.
Car Type Options (All, Regular, Luxury)	Buttons	Selects the desired car type.
Price Range Slider	Slider	Adjusts minimum and maximum price.
Rental Time	Buttons	Selects rental duration.

Options (Hour, Day, Weekly, Monthly)		
Pick-up & Drop Date Field	Button	Opens the date calendar.
Car Location Field	Input Field	Allows entering car location.
Color Options	Buttons	Selects preferred car color.
Seating Capacity (2, 4, 6, 8)	Buttons	Selects number of seats.
Fuel Type (Electric, Petrol, Diesel, Hybrid)	Buttons	Filters cars by fuel type.
Clear All	Button	Resets all filters.
show Cars Button	how Cars Button	how Cars Button
Car Details Interface		
Back Icon	Button	Returns to the previous screen.
More Options Icon	Button	Opens the additional options menu.
Favorite Icon	Button	Adds the car to favorites.
Image Slider Dots	Buttons	Switches between car images.
Owner Contact (Call Icon)	Button	Initiates a call to the car owner.
Owner Contact (Chat Icon)	Button	Opens messaging with the owner.
Availability Calendar	Calendar	Selects available rental dates.
Book Now Button	Button	Proceeds to the booking process.
Booking Details Interface		
Back Icon	Button	Returns to the previous screen.
More Options Icon	Button	Opens additional options menu.
Offers Section	Button	Opens available offers.
Payment Method Selection	Radio Button	Selects a saved credit/debit card.
Add New Card	Button	Adds a new payment method.
Insurance Checkbox	Checkbox	Adds additional insurance coverage.
View Detailed Bill	Text/Button	Opens the detailed bill summary.
Proceed to Pay	Button	Completes payment and confirms booking.
Booking History Interface		
Back Icon	Button	Returns to the previous screen.
More Options Icon	Button	Opens additional options.

History Tabs (Current / Upcoming / Completed)	Tabs	Switches between booking categories.
Booking Card	Button	Opens booking details (if supported).
Re-Book Link	Text/Button	Repeats the same booking again.
Write a Review Link	Text/Button	Opens the review submission page.
Navbar (Home, Search, Messages, Notifications, Profile)	Buttons	Navigates between main app sections.
Owner Dashboard Interface		
Back Icon	Button	Returns to previous screen.
More Options Icon	Button	Opens additional owner options.
Earnings Tabs (Week / Month / Year)	Tabs	Changes the earnings view.
Recent Rating Section	Button	Opens full rating list (if tapped).
Notification Box	Button	Opens the notifications page.
Dashboard Tabs (Active, Booked, Under Maintenance)	Tabs	Filters booking categories.
Navbar (Home, Search, Messages, Notifications, Profile)	Buttons	Moves between main app pages.
Content Moderation Interface		
Search	Text Field	Allows the admin to search for reported content.
Moderate	Button	Opens moderation tools / activates filter mode.
Content (Row)	Text	Displays the description of the flagged content.
Author	Text	Displays the username of the content owner.
Status	Label (Tag)	Shows moderation status: Pending / Approved / Rejected.
Review	Button	Opens detailed review screen for this report.
Reject	Button	Allows the admin to reject content or mark the report invalid.

Content Moderation Interface		
Search	Text Field	Allows the admin to search for a specific report.
Moderate	Button	Opens moderation/filters for reports.
Report Name (Column)	Text	Shows available report types (User Activity, Payments...).
Last Generated	Text	Shows the last generated date for each report.
Downloads	Text/Icon	Displays number of downloads or "-" if none.
Download Icon	Icon/Button	Downloads the report file in PDF/Excel format.
View	Button	Opens a detailed view of the selected report.
Monthly Active Users Chart	Chart	Displays analytics visualization for user activity across months.

Table 3 - Screen Objects and action

4.5 Other Interfaces

An error message is a notification that appears when something goes wrong in the system or application

4.5.1 Error Messages

4.5.1.1 Sign Up Error

This error message appears when the user enters Invalid Email or a password or National-ID that does not meet the system's required security rules. The system requires The National-ID to be 10-Digit, the password to be at least 8 characters long, and it must include:

- Uppercase letters
- Lowercase letters
- Numbers
- Special characters

If any of these conditions are missing, the system displays this error to notify the user that their password is too weak or incomplete. The user must correct their password and try again.

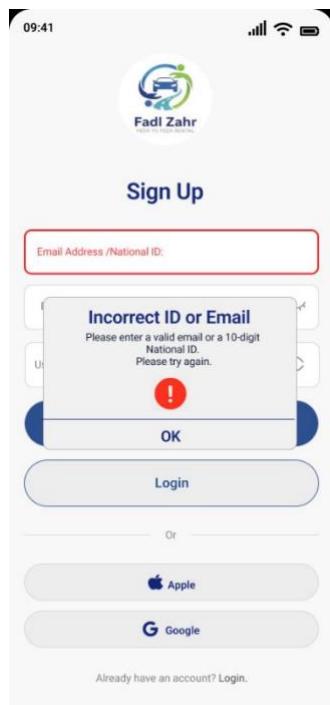


Figure 2 – Sign Up error 1

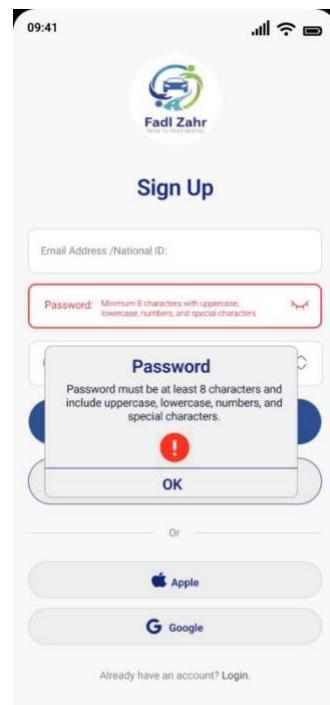


Figure 3 - Sign Up error 2

4.5.1.2 User Type Selection Error

This error message appears when the user tries to sign up without selecting a **user type** from the dropdown menu. The system requires the user to choose one of the three available roles:

- Renter
- Car Owner
- Admin

If the user does not select a role, the system cannot continue the registration process. The error message notifies the user to choose the appropriate user type before proceeding.

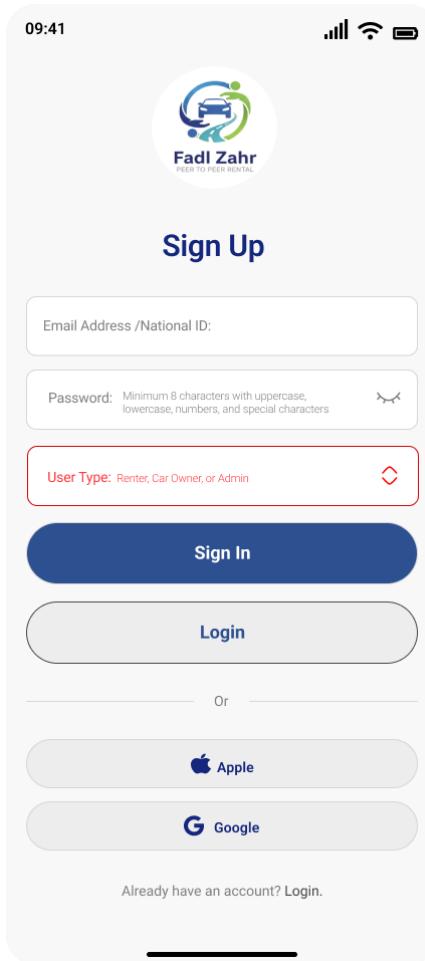


Figure 4 - User Type selection Error

4.5.1.3 Incorrect OTP Error Message

This error message appears when the user enters an invalid or incorrect verification code (OTP). The system checks whether the 6-digit code matches the one that was sent to the user's email. If the numbers do not match, are incomplete, or contain invalid digits, the system rejects the input and displays this message. The user must re-enter the correct 6-digit OTP or request a new code.



Figure 5 - Incorrect OTP

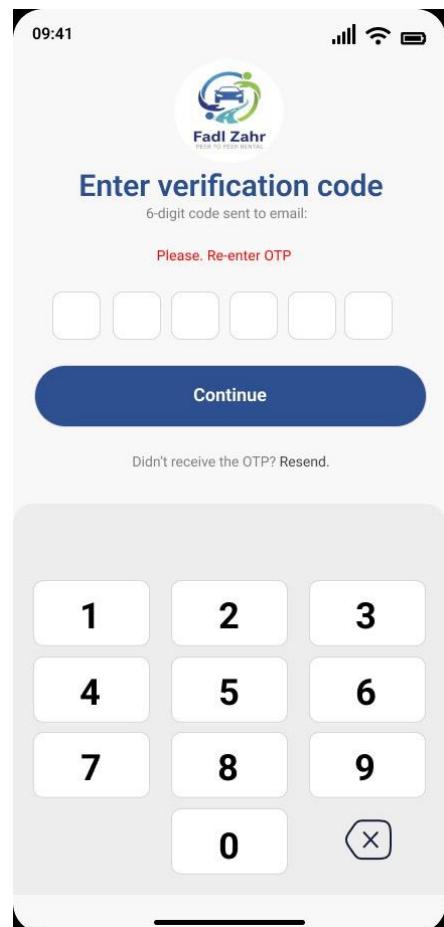


Figure 6 - Re-enter OTP

4.5.1.4 No Results Found

This error message appears when the system cannot find any cars that match the user's search keywords or selected filters. It indicates that the current search criteria are too specific, incorrect, or do not match any available cars in the system. The user is advised to adjust their search terms or modify the selected filters to get different results.

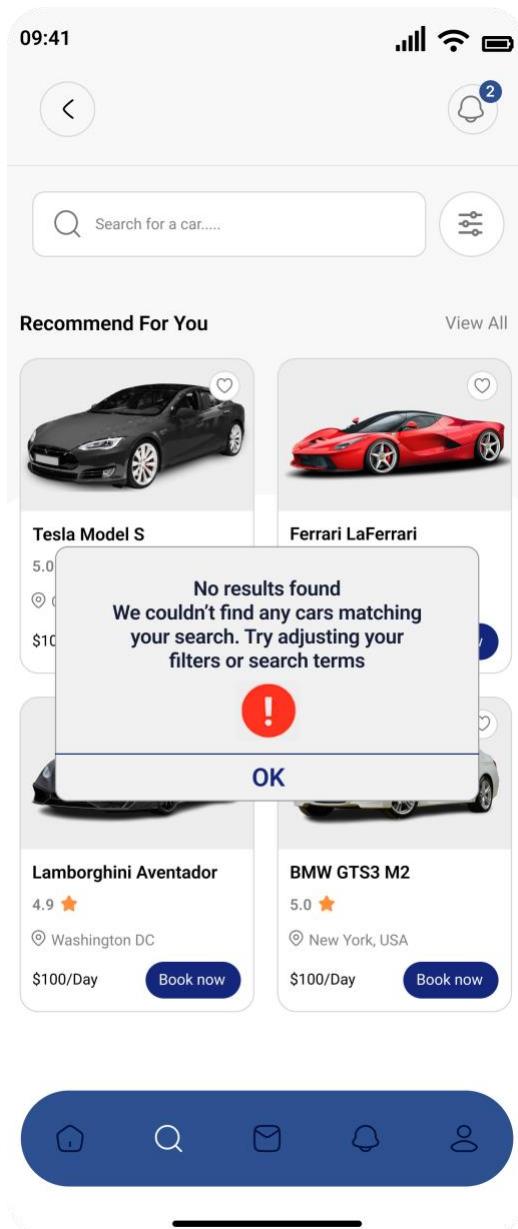


Figure 7 - No result Found Error

4.5.1.5 This car cannot be rebooked now

This error message appears when the user tries to rebook a car that is **currently unavailable**. The unavailability might be due to:

- The car already being booked by another user.
- The owner temporarily disabling bookings for this vehicle.
- The selected dates are not available.

The system displays this message to inform the user that the car cannot be booked again right now, and they must choose a different vehicle

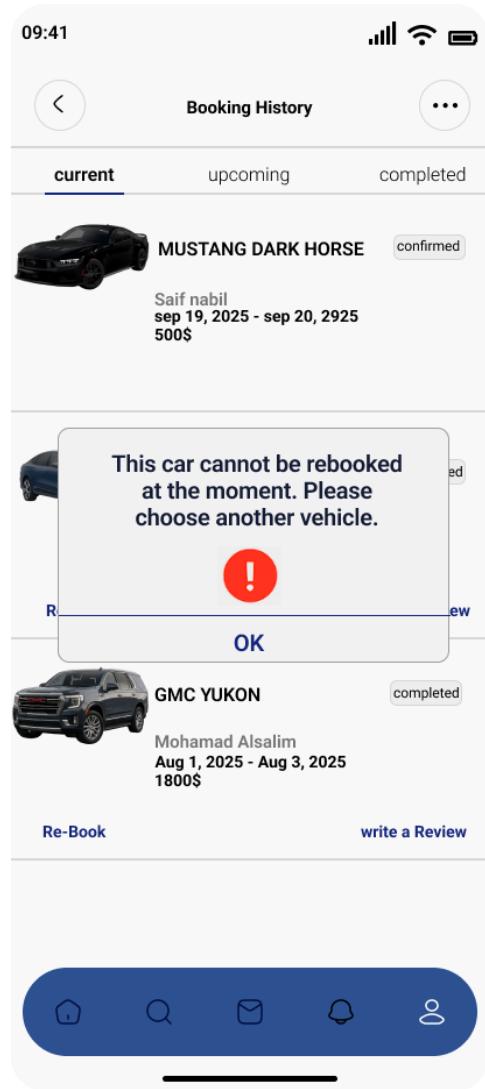


Figure 8 - Car Rebooked Error

4.5.1.6 This field is required

This error occurs because the system has detected that one or more mandatory fields in the Edit Car Listing interface were left incomplete. When the user attempts to submit the form without providing all required information such as the car's make, model, year, plate number, or pricing details the system triggers a validation check.

Since the validation fails, the application displays the alert message “This field is required” to indicate that essential data is missing. The highlighted input fields in red serve as visual indicators, guiding the user to the specific areas that must be completed. The update process cannot proceed until all required fields are properly filled.

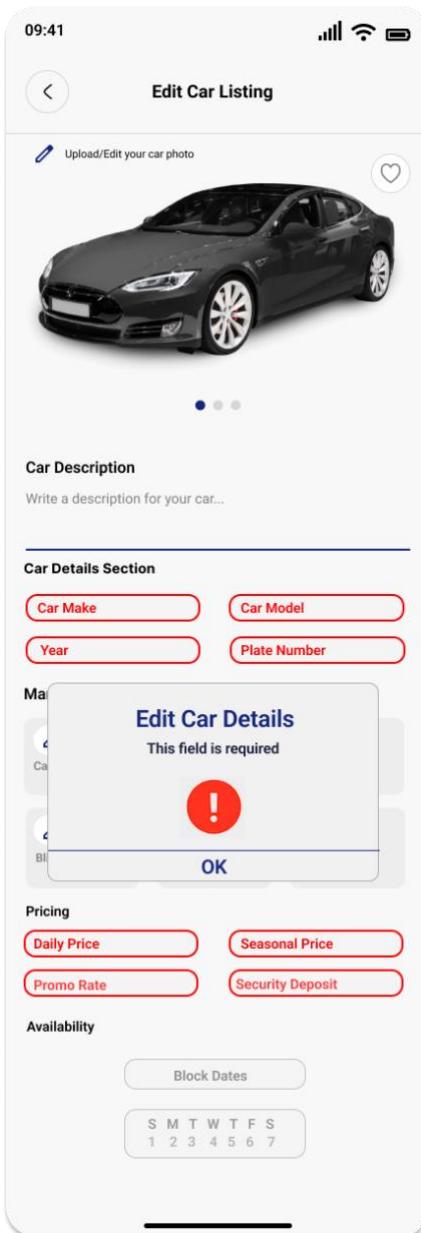


Figure 9 - Edit Car Details Error

4.5.1.7 Invalid plate number format

This error appears because the plate number entered does not match the required format. The system checks the plate number structure, and when the input is invalid whether the pattern, characters, or length is incorrect it stops the update and shows “Invalid plate number format.” The red highlight indicates exactly which field needs to be corrected before saving.

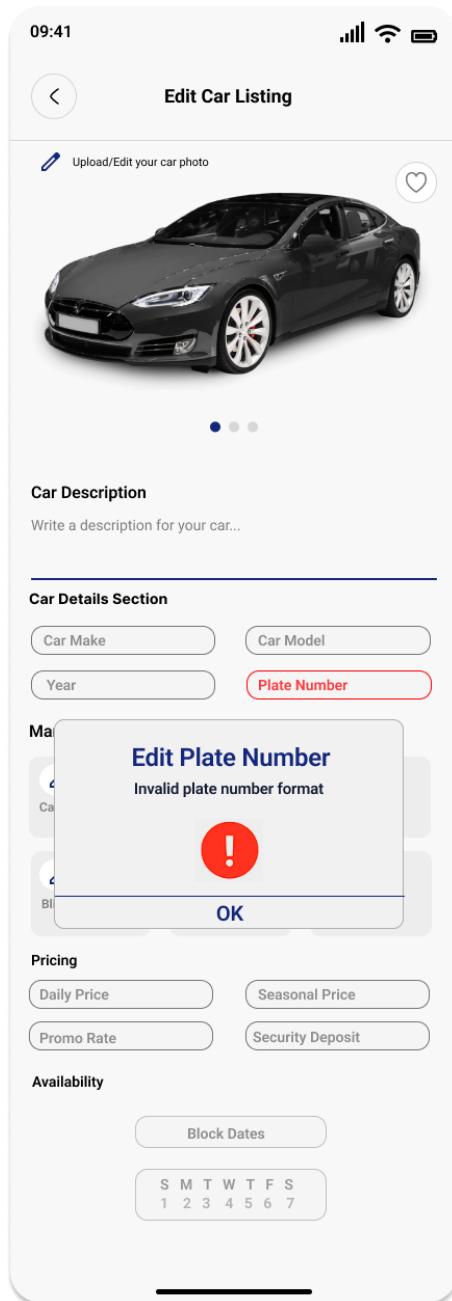


Figure 10 - Invalid Plate Number Error

4.5.1.8 Action failed

This error appears when the owner tries to accept or decline a booking request while the system is still missing some required data for that action. It means the system could not complete the operation because a necessary piece of information such as booking details, dates, or user info was not fully loaded or validated. So, the message “Action failed. Missing required information” indicates that the request cannot be processed until all needed data is correctly available.

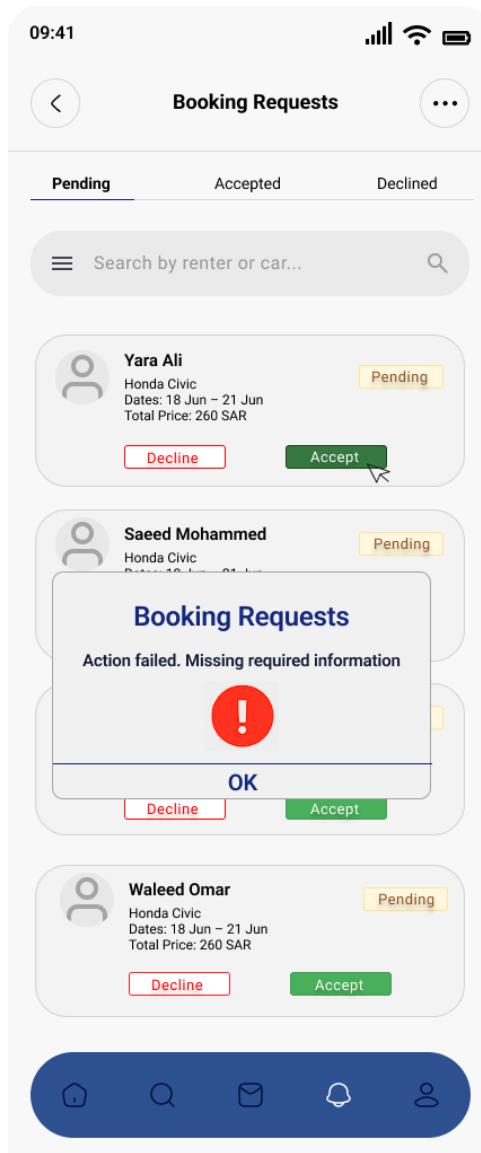


Figure 11 - Booking Request Action Error

4.5.1.9 Unable to load earnings. Try again later

This error appears when the system tries to retrieve the earnings data for a specific user, but the request fails usually because the server didn't respond, the data couldn't be fetched, or there was a temporary connection issue. The message “Unable to load earnings. Try again later” indicates that the earnings information is currently unavailable, and the system is asking the admin to retry once the data can be successfully loaded. It's a temporary failure related to data loading, not a mistake from the user.

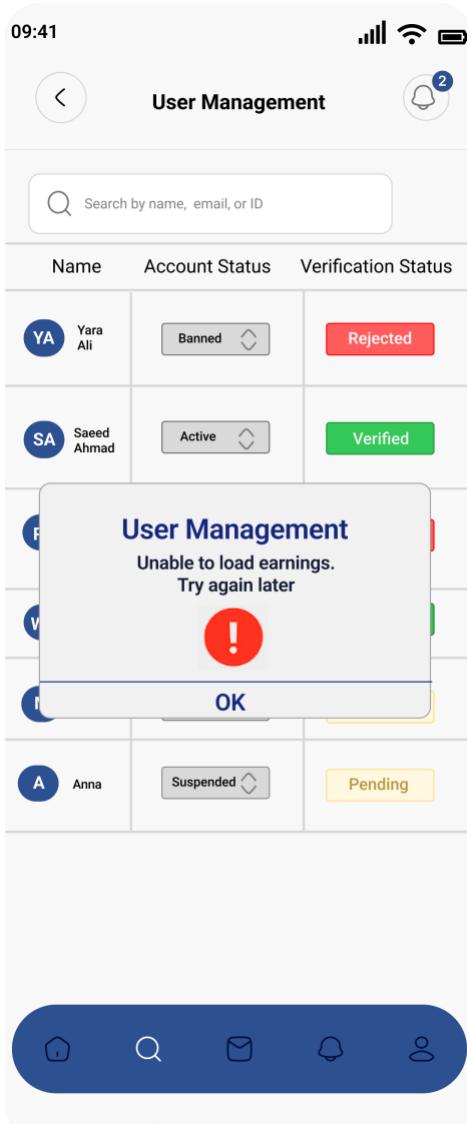


Figure 12 - User Management Data Retrieval Error

4.5.1.10 Approval Requires Explanation

This error appears because the system requires the moderator to write an explanation before approving any content. When the moderator clicks “Approve” without entering a justification, the system blocks the action and displays this warning. The purpose is to ensure accountability and maintain a clear moderation record.

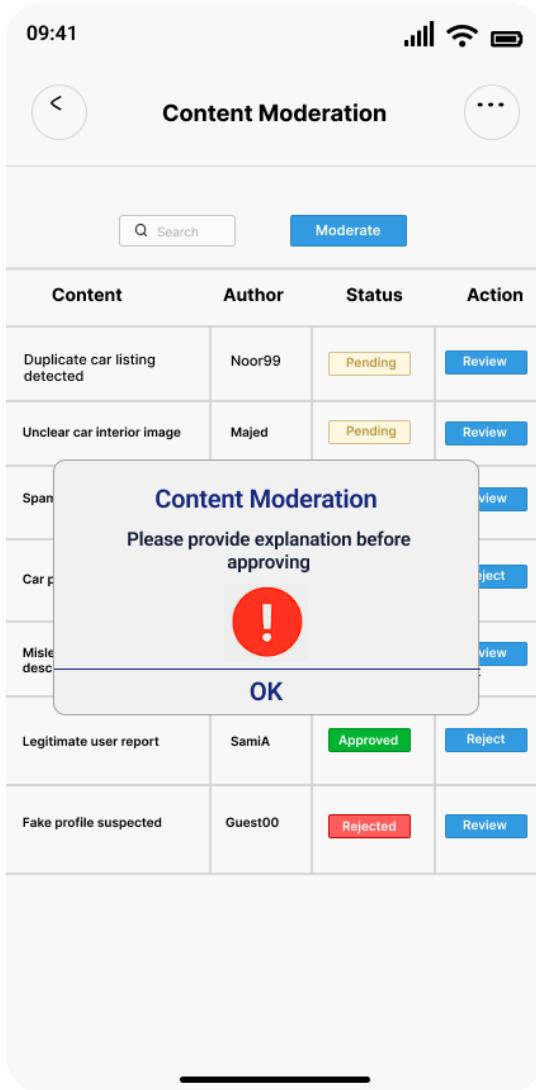


Figure 13 - Content Moderation Approval Validation Error

4.5.1.11 Please select a rating before submitting

This error appears because the user did not select the main rating (Overall Rating) at the top. Even if the other categories (Car, Communication) are selected, the system requires the primary rating before submitting.

So the app shows the message:

“Please select a rating before submitting.”

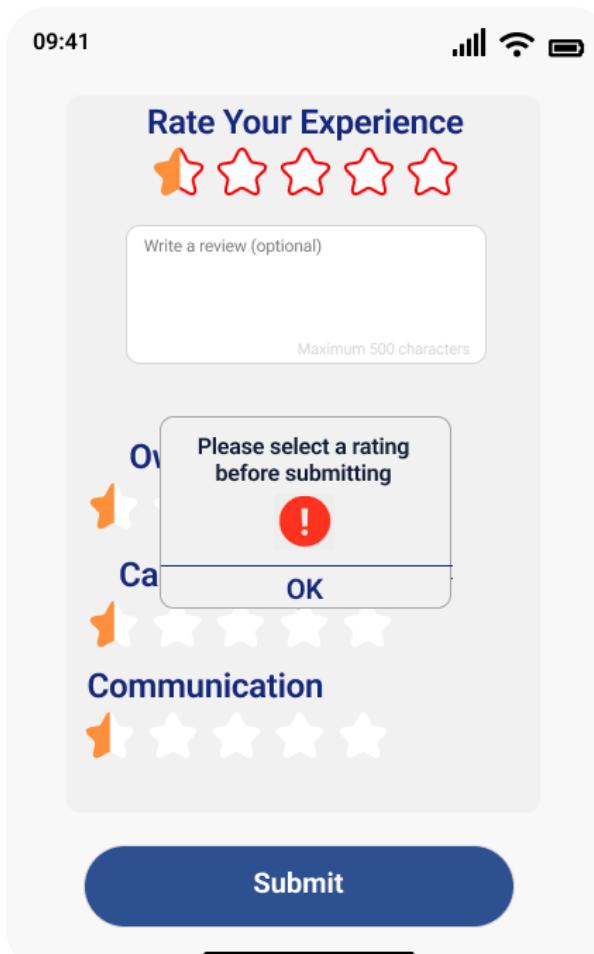


Figure 14 - Rating Submission Validation Error

5. System Architecture

In this section, an abstract overview of the functionalities and responsibilities of the "Fadl Zahr" system is provided. The decomposition of the system into components and subsystems, along with their assigned responsibilities, is addressed in the Architectural Design Approach, Architectural Design, and Subsystem Architecture subsections.

5.1 Architectural Design Approach

The optimal architectural design approach for the "Fadl Zahr" system is the Layered (3-Tier) Architecture. This approach is highly suitable for client-server systems like "Fadl Zahr," where each layer is dedicated to a specific set of concerns. This separation ensures that changes or future updates are confined to their respective layers, making the system highly maintainable and adaptable. Furthermore, the layered approach provides a robust multi-level security model, which is critical for "Fadl Zahr" as it handles sensitive user data, including personal identification, financial transactions, and vehicle details. This design inherently supports the high security and data protection requirements outlined in the SRS.

5.2 Architectural Design

Architectural Design The "Fadl Zahr" system's layered architecture consists of four distinct layers: Presentation Layer, Business Layer, Persistence Layer, and Data Layer.

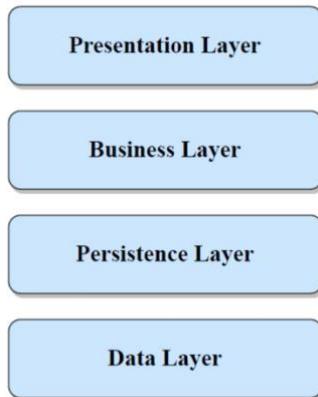


Figure 15- Fadl Zahr System Layered Approach

- Presentation (UI) Layer: This is the top layer, responsible for all end-user interaction. It comprises the Flutter-based mobile application for Renters and Car Owners, and a web-based dashboard for Administrators. Its primary functions are to display information, capture user input, and communicate with the Business Layer.
- Business Layer: This layer contains the core application logic, services, and API endpoints. It processes requests from the Presentation Layer, orchestrates workflows (e.g., booking,

payment), enforces business rules, and handles validation and exception handling. It acts as a bridge between Presentation and Persistence layers.

- Persistence Layer: This layer is responsible for all data access and abstraction. It translates operations from the Business Layer into database queries, managing the Object-Relational Mapping (ORM). It ensures secure and permission-based access to the data stored in the Data Layer. Firebase SDKs will be primarily used for this interaction.
- Data Layer: This is the foundational layer, comprising the system's databases and external services. It includes:
 - Firebase Firestore: The primary cloud database for storing user profiles, vehicle listings, bookings, and transactions.
 - Firebase Authentication: Manages user accounts and secure login.
 - External Servers/APIs: Integration points with the Absher API for identity verification, payment gateways (e.g., STC Pay, Mada) for transactions, and Firebase Cloud Messaging (FCM) for push notifications.

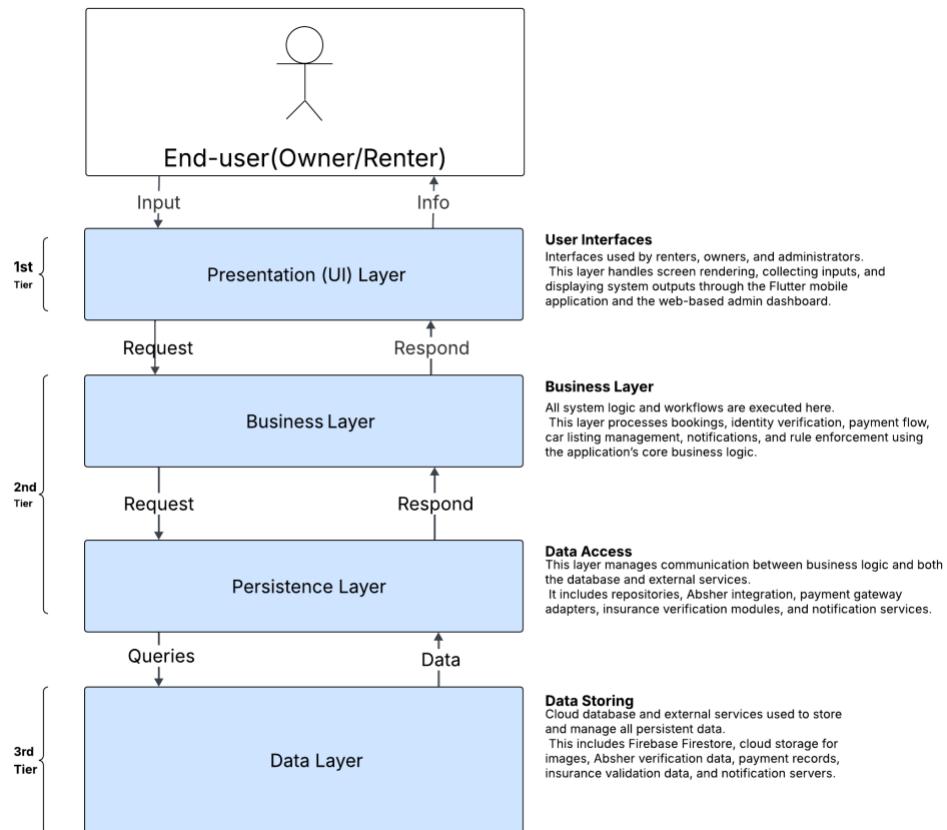


Figure 16- Fadl Zahr System architecture Design

5.2 Subsystem Architecture

In this subsection, the architectural design is decomposed into its major functional subsystems. The data flow for key processes within these subsystems is illustrated using Level 0 Data Flow Diagrams (DFDs) to demonstrate the logical architecture and how data is processed and stored.

5.3.1 General View of System

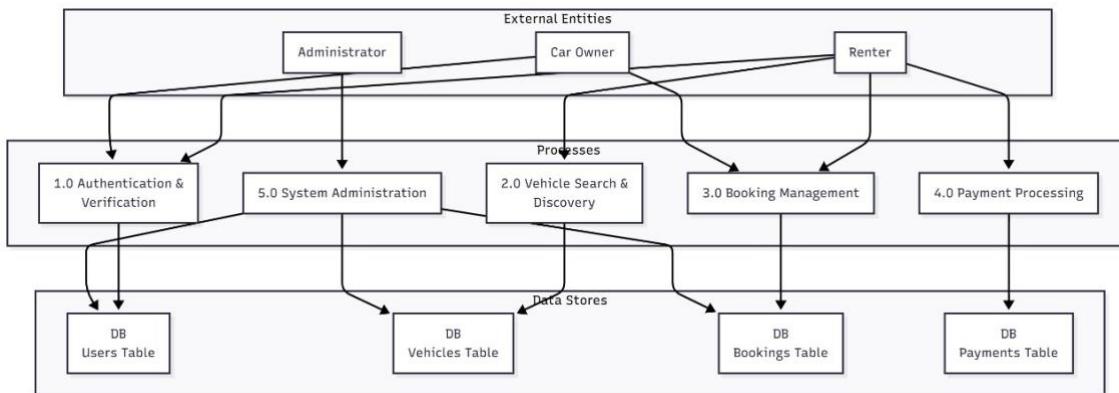


Figure 17 - General View of System

5.3.2 Renter Subsystem

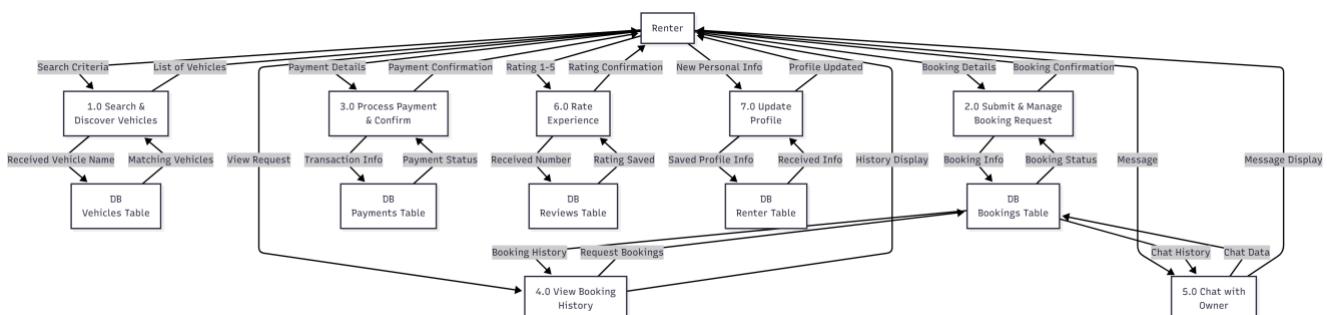


Figure 18 - Renter Subsystem

5.3.3 Car Owner Subsystem

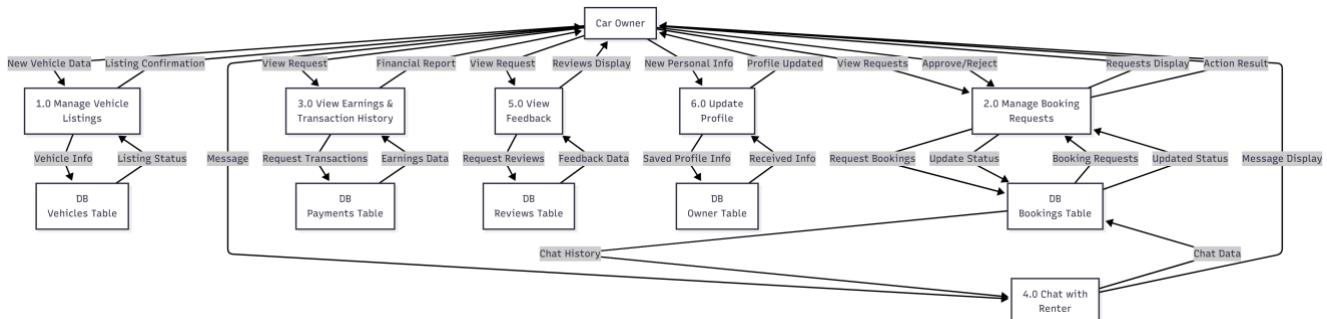


Figure 19 – Car Owner Subsystem

5.3.4 Admin Subsystem

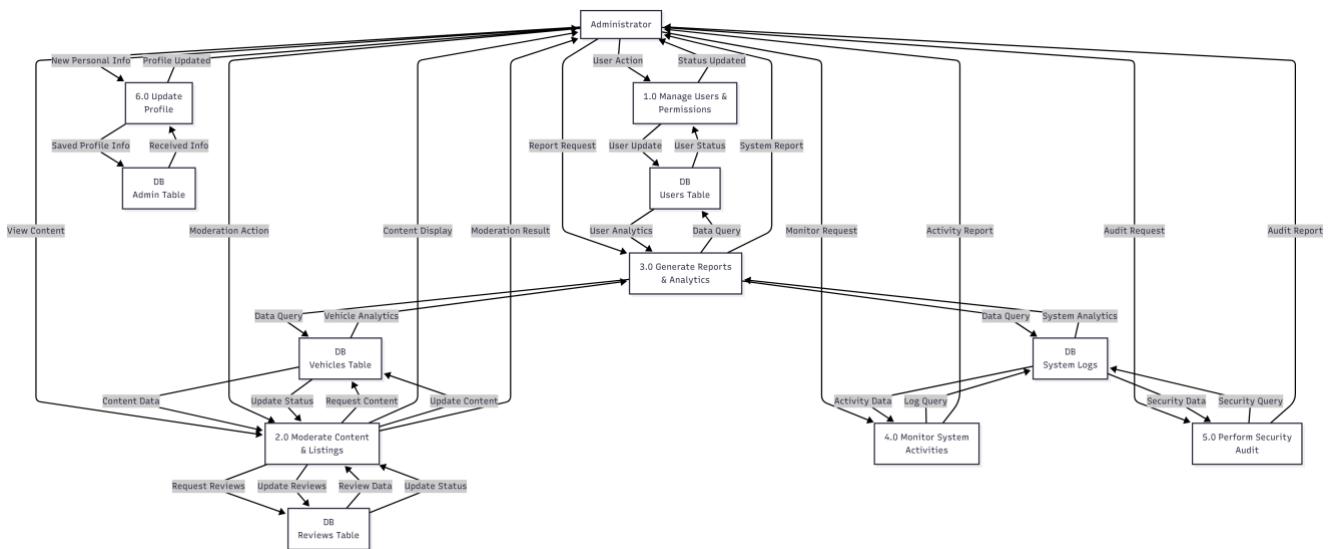


Figure 20 - Admin Subsystem

6. Data Design

In this section, a detailed description of how Fadl Zahr system information transformed into data structure including data, data types, necessary fields, and the application's database. Additionally, a list of the application's entities will be provided. ER diagrams and ER mapping are used as a UML representation of the database entities.

6.1 Data Description

The Fadl Zahr system manages data with separate entities for each user type (Admin, Owner, Renter) along with their related data for vehicles, bookings, payments, and reviews.

Entity	Field	Type	PK	FK	Constraint
Admin	AdminID	VARCHAR (10)	✓		NOT NULL, UNIQUE
	FullName	VARCHAR (50)			NOT NULL
	Email	VARCHAR (70)			NOT NULL, UNIQUE
	PhoneNumber	VARCHAR (10)			NOT NULL
	NationalID	VARCHAR (10)			NOT NULL, UNIQUE
	PasswordHash	VARCHAR(255)			NOT NULL
	Role	VARCHAR (20)			DEFAULT 'Admin'
	CreatedDate	DATETIME			NOT NULL
Owner	OwnerID	VARCHAR (10)	✓		NOT NULL, UNIQUE
	FullName	VARCHAR(50)			NOT NULL
	Email	VARCHAR(70)			NOT NULL, UNIQUE
	PhoneNumber	VARCHAR(10)			NOT NULL
	NationalID	VARCHAR(10)			NOT NULL, UNIQUE
	PasswordHash	VARCHAR(255)			NOT NULL

	IsVerified	BOOLEAN			DEFAULT FALSE
	IBAN	VARCHAR(24)			NULL
	BankName	VARCHAR(30)			NULL
	TotalEarnings	DECIMAL(12,2)			DEFAULT 0.00
Renter	RatingAvg	DECIMAL (3,2)			DEFAULT 0.00
	RenterID	VARCHAR(10)	✓		NOT NULL, UNIQUE
	FullName	VARCHAR(50)			NOT NULL
	Email	VARCHAR(70)			NOT NULL, UNIQUE
	PhoneNumber	VARCHAR(10)			NOT NULL
	NationalID	VARCHAR(10)			NOT NULL, UNIQUE
	PasswordHash	VARCHAR(255)			NOT NULL
	IsVerified	BOOLEAN			DEFAULT FALSE
	Address	TEXT			NULL
	DrivingLicenseNo	VARCHAR(20)			NULL
	LicenseExpiry	DATE			NULL
	RatingAvg	DECIMAL(3,2)			DEFAULT 0.00
	VehicleID	VARCHAR(10)	✓		NOT NULL, UNIQUE
	OwnerID	VARCHAR(10)		✓	NOT NULL
	Make	VARCHAR(30)			NOT NULL

Vehicle	Model	VARCHAR(30)			NOT NULL
	Year	INT			NOT NULL
	Color	VARCHAR(20)			NOT NULL
	LicensePlate	VARCHAR(10)			NOT NULL, UNIQUE
	FuelType	VARCHAR(20)			NOT NULL
	Transmission	VARCHAR(20)			NOT NULL
	Seats	INT			NOT NULL
	DailyPrice	DECIMAL(10,2)			NOT NULL
	InsuranceExpiry	DATE			NOT NULL
	IsAvailable	BOOLEAN			DEFAULT TRUE
Booking	Location	VARCHAR(100)			NOT NULL
	BookingID	VARCHAR(10)	✓		NOT NULL, UNIQUE
	RenterID	VARCHAR(10)		✓	NOT NULL
	VehicleID	VARCHAR(10)		✓	NOT NULL
	StartDate	DATE			NOT NULL
	EndDate	DATE			NOT NULL
	TotalAmount	DECIMAL(10,2)			NOT NULL
	Status	VARCHAR(20)			DEFAULT 'Pending'
	PickupLocation	VARCHAR(100)			NULL
	SpecialRequests	TEXT			NULL
	PaymentID	VARCHAR(10)	✓		NOT NULL, UNIQUE

Payment					
	BookingID	VARCHAR(10)		✓	NOT NULL
	Amount	DECIMAL(10,2)			NOT NULL
	PaymentMethod	VARCHAR (20)			NOT NULL
	Status	VARCHAR(20)			DEFAULT 'Pending'
	CommissionAmount	DECIMAL(10,2)			NOT NULL
	OwnerAmount	DECIMAL(10,2)			NOT NULL
Owner_Renter Review	ReviewID	VARCHAR(10)	✓		NOT NULL, UNIQUE
	BookingID	VARCHAR(10)		✓	NOT NULL
	ReviewerType	VARCHAR(10)			NOT NULL
	ReviewedOwnerID	VARCHAR(10)		✓	NULL
	ReviewedRenterID	VARCHAR(10)		✓	NULL
	Rating	INT			CHECK (1– 5)
	Comment	TEXT			NULL

Table 4 - Data Description

6.2 Data Dictionary

Entity	Field	Description
Admin	AdminID	Unique identifier for each administrator
	Role	Administrative role (e.g., SuperAdmin, Support)
	CreatedDate	Date when admin account was created
Owner	OwnerID	Unique identifier for each vehicle owner
	IBAN	Owner's bank account number for payments
	BankName	Name of owner's bank
	RatingAvg	Average rating received from renters
	TotalEarnings	Total earnings from all completed bookings
Renter	RenterID	Unique identifier for each renter
	Address	Renter's residential address
	DrivingLicenseNo	Renter's driving license number
	LicenseExpiry	Expiry date of driving license
	RatingAvg	Average rating received from owners
Vehicle	VehicleID	Unique identifier for each vehicle
	OwnerID	Reference to the vehicle owner
	FuelType	Type of fuel (Petrol, Diesel, Electric, Hybrid)
	Transmission	Transmission type (Automatic,

		Manual)
	Seats	Number of seats in the vehicle
	InsuranceExpiry	Expiry date of vehicle insurance
	Location	Current location of the vehicle
Booking	BookingID	Unique identifier for each booking
	RenterID	Reference to the renter who made the booking
	VehicleID	Reference to the booked vehicle
	PickupLocation	Preferred pickup location for the vehicle
	SpecialRequests	Any special requirements from the renter
Payment	PaymentID	Unique identifier for each payment
	CommissionAmount	Platform commission deducted from payment
	OwnerAmount	Net amount transferred to owner after commission
Owner_Renter Review	ReviewID	Unique identifier for each review
	ReviewerType	Type of reviewer ('Owner' or 'Renter')
	ReviewedOwnerID	Reference to the owner being reviewed
	ReviewedRenterID	Reference to the renter being reviewed
	Rating	Numeric rating from 1 to 5

Table 5 - Data Dictionary

6.3 Database Description

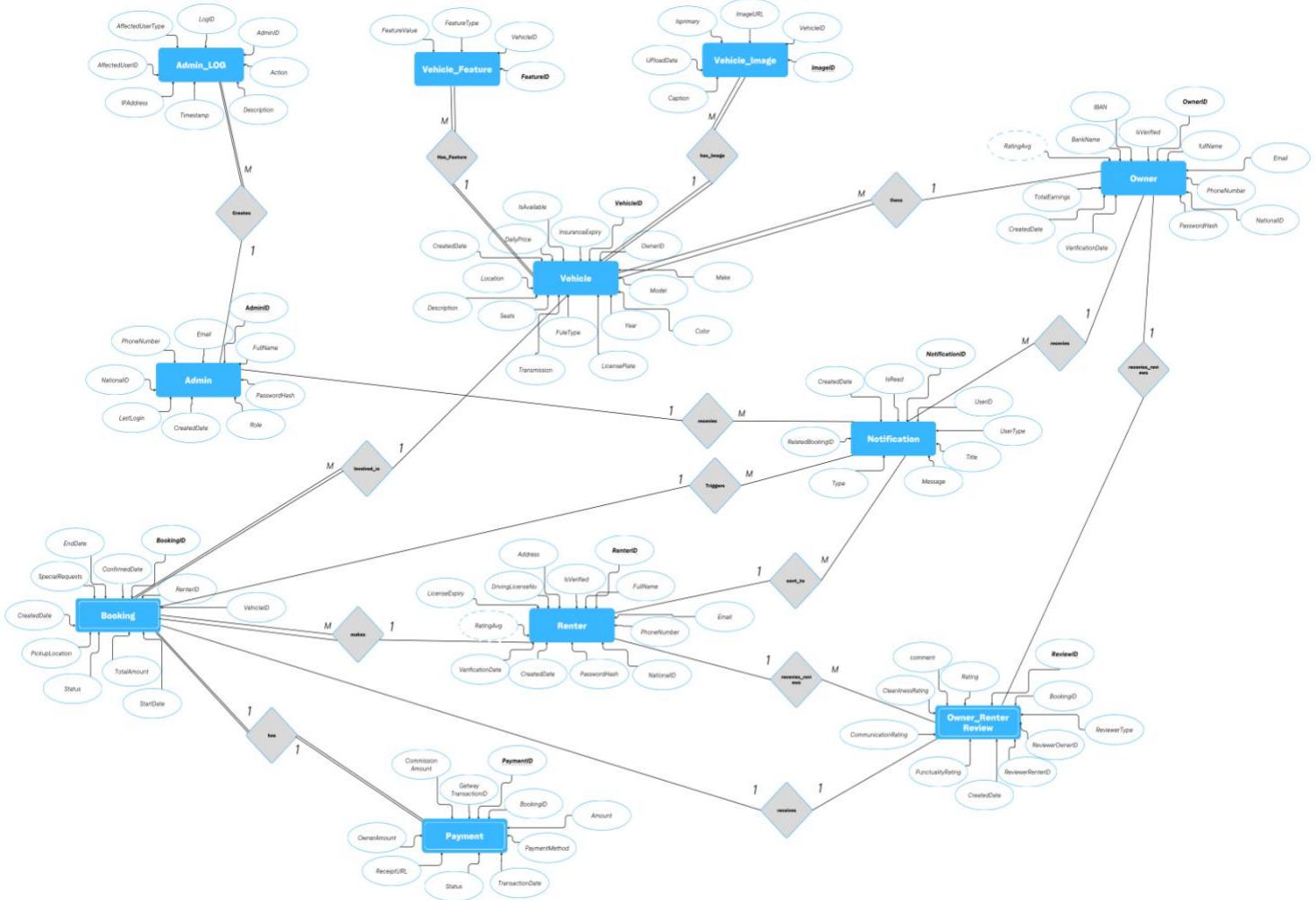


Figure 21 - Fadl Zahr ER Diagram



Figure 22 - Fadl Zahr Relational Schema

7. Component Design

7.1 Common Functions

7.1.1 User Sign In

```
UserSignIn( ){  
  
INPUT: email, password, userType  
BEGIN  
    DETERMINE which collection to search based on userType:  
        IF userType = "admin" THEN search admins collection  
        IF userType = "owner" THEN search owners collection  
        IF userType = "renter" THEN search renters collection  
  
    SEARCH Firestore for user document with matching email  
  
    IF user found AND password matches THEN  
        GENERATE JWT token containing userID and userType  
        UPDATE user's lastLogin timestamp  
        RETURN success response with token and user profile  
    ELSE  
        RETURN error message "Invalid email or password"  
    END IF  
}
```

7.1.2 Identity Verification

```
VerifyIdentity( ){  
  
INPUT: nationalID, userID, userType  
BEGIN  
    CALL Absher API with nationalID and user credentials  
    WAIT for biometric verification process  
  
    IF Absher returns verification success THEN  
        UPDATE user document in appropriate collection:  
            SET IsVerified = true  
            SET verificationDate = current timestamp  
        LOG verification success in audit trail  
        RETURN "Verification completed successfully"  
    ELSE  
        LOG verification failure with reason  
        RETURN "Verification failed: " + error reason  
    END IF }
```

7.1.3 Password Recovery

```
RecoverPassword( ){  
  
    INPUT: email, userType  
    BEGIN  
        SEARCH for user in appropriate collection using email  
  
        IF user found THEN  
            GENERATE 6-digit OTP code  
            SET OTP expiry time to 10 minutes from now  
            STORE OTP hash in user document temporarily  
            SEND OTP to user's email or phone  
            RETURN "Verification code sent to your registered email"  
        ELSE  
            RETURN "No account found with this email address"  
        END IF  
    }  
  
}
```

7.2 Renter Functions

7.2.1 Search Available Vehicles

```
SearchVehicles( ){  
  
    INPUT: location, startDate, endDate, filters (priceRange, carType, features)  
    BEGIN  
        QUERY vehicles collection WHERE:  
            IsAvailable = true  
            location is within search radius  
            vehicle is not booked for the selected dates  
  
        APPLY additional filters:  
            IF priceRange specified THEN filter by DailyPrice  
            IF carType specified THEN filter by Make/Model/Type  
            IF features specified THEN filter by VehicleFeatures  
  
        CALCULATE distance from user location for each vehicle  
        SORT results by:  
            closest distance first  
            then by highest owner rating  
            then by lowest price  
  
        RETURN paginated list of matching vehicles with details  
    }  
  
}
```

7.2.2 Create Booking Request

CreateBooking(){

INPUT: renterID, vehicleID, startDate, endDate, specialRequests

BEGIN

 CHECK if renter is verified

 IF renter not verified THEN RETURN "Please verify your identity first"

 CHECK vehicle availability for the selected dates

 IF vehicle not available THEN RETURN "Vehicle not available for selected dates"

 CALCULATE total amount:

 rentalDays = difference between endDate and startDate

 totalAmount = rentalDays × vehicle.DailyPrice

 CREATE new booking document:

 SET Status = "Pending"

 SET RenterID = renterID

 SET VehicleID = vehicleID

 SET TotalAmount = totalAmount

 SET SpecialRequests = specialRequests

 SEND notification to vehicle owner about new booking request

 RETURN "Booking request sent to owner. Waiting for approval."

}

7.2.3 Process Payment

ProcessPayment(){

INPUT: bookingID, paymentMethod, paymentDetails

BEGIN

 RETRIEVE booking details

 IF booking Status ≠ "Approved" THEN RETURN "Booking not approved yet"

 PROCESS payment through payment gateway:

 IF paymentMethod = "Credit Card" THEN process card payment

 IF paymentMethod = "Mada" THEN process Mada payment

 IF paymentMethod = "Apple Pay" THEN process Apple Pay

 IF payment successful THEN

 CALCULATE commission (10% of total amount)

 CALCULATE ownerAmount (total amount - commission)

 CREATE payment record:

 SET Status = "Completed"

 SET CommissionAmount = commission

 SET OwnerAmount = ownerAmount

```
UPDATE booking Status = "Confirmed"
SEND confirmation notifications to renter and owner
RETURN "Payment successful. Booking confirmed."
ELSE
    RETURN "Payment failed: " + gateway error message
END IF
}
```

7.3 Owner Functions

7.3.1 Add Vehicle Listing

```
AddVehicleListing( ){
    INPUT: ownerID, vehicleDetails, imageFiles
    BEGIN
        VERIFY owner is authenticated and verified
        IF owner not verified THEN RETURN "Please verify your identity first"

        UPLOAD vehicle images to Firebase Storage
        GET download URLs for uploaded images

        CREATE new vehicle document:
            GENERATE unique VehicleID
            SET OwnerID = ownerID
            SET all vehicle specifications from vehicleDetails
            SET IsAvailable = true
            SET createdDate = current timestamp
            STORE image URLs in VehicleImages array

        RETURN "Vehicle listed successfully. Now visible to renters."
}
```

7.3.2 Manage Booking Requests

```
ManageBookingRequests( ){
    INPUT: ownerID, action, bookingID, rejectionReason
    BEGIN
        RETRIEVE booking details and associated vehicle
        VERIFY that vehicle belongs to the owner

        IF action = "approve" THEN
            CHECK vehicle availability for booking dates
}
```

```
IF available THEN
    UPDATE booking Status = "Approved"
    SEND notification to renter: "Booking approved"
    RETURN "Booking approved successfully"
ELSE
    RETURN "Vehicle no longer available for these dates"
END IF

ELSE IF action = "reject" THEN
    UPDATE booking Status = "Rejected"
    SET rejectionReason = provided reason
    SEND notification to renter: "Booking rejected: " + reason
    RETURN "Booking rejected"
END IF
}
```

7.3.3 Update Vehicle Availability

```
UpdateVehicleAvailability( ){
    INPUT: ownerID, vehicleID, newAvailability
    BEGIN
        VERIFY vehicle belongs to owner
        IF not owner's vehicle THEN RETURN "Access denied"

        UPDATE vehicle document:
            SET IsAvailable = newAvailability.available
            SET blockedDates = newAvailability.blockedDates
            IF newAvailability.dailyPrice provided THEN update DailyPrice

        IF vehicle made unavailable THEN
            CANCEL all pending bookings for this vehicle
            NOTIFY affected renters about cancellation
        END IF

        RETURN "Vehicle availability updated successfully"
}
```

7.4 Admin Functions

7.4.1 User Management

```
ManageUser( ){
    INPUT: adminID, action, targetUserID, userType, reason
    BEGIN
```

VERIFY admin has necessary permissions

IF action = "suspend" THEN
 UPDATE user document in appropriate collection:
 SET accountStatus = "Suspended"
 SET suspensionReason = reason
 SET suspendedBy = adminID
 SET suspensionDate = current timestamp

 CANCEL all active bookings for this user
 NOTIFY user about suspension with reason
 RETURN "User suspended successfully"

ELSE IF action = "verify" THEN
 UPDATE user document:
 SET IsVerified = true
 SET verifiedBy = adminID
 SET verificationDate = current timestamp

 NOTIFY user about manual verification
 RETURN "User verified manually"

ELSE IF action = "delete" THEN
 ANONYMIZE user personal data for privacy
 ARCHIVE user records for legal compliance
 REMOVE user from active collections
 RETURN "User data anonymized and archived"

END IF

}

7.4.2 System Monitoring

MonitorSystem(){

INPUT: adminID, timeframe (today, week, month), metrics
BEGIN

GATHER statistics from all collections:
 FROM owners: count new registrations, total owners
 FROM renters: count new registrations, total renters
 FROM vehicles: count active listings, new listings
 FROM bookings: count total, by status, conversion rate
 FROM payments: total revenue, commission, owner payouts

CALCULATE key performance indicators:
 bookingConversionRate = (confirmed bookings / total bookings) × 100
 platformRevenue = sum of all commissions
 activeUserRate = (users with activity / total users) × 100

```
    GENERATE visual charts and trend analysis
    FLAG any unusual patterns or suspicious activities

    RETURN dashboard with all metrics and visualizations
}
```

7.4.3 Content Moderation

```
ModerateContent( ){

INPUT: adminID, contentID, contentType, moderationAction, reason
BEGIN

    IF contentType = "vehicle_listing" THEN
        RETRIEVE vehicle details and images
        IF moderationAction = "approve" THEN
            SET listingStatus = "Active"
            NOTIFY owner: "Your vehicle listing is now live"
        ELSE IF moderationAction = "reject" THEN
            SET listingStatus = "Rejected"
            NOTIFY owner: "Listing rejected: " + reason
        END IF

    ELSE IF contentType = "review" THEN
        RETRIEVE review content
        IF moderationAction = "remove" THEN
            DELETE review content
            NOTIFY author: "Review removed: " + reason
        ELSE IF moderationAction = "edit" THEN
            REMOVE inappropriate language from review
            KEEP rating and safe content
            NOTIFY author about edits made
        END IF

    ELSE IF contentType = "user_profile" THEN
        RETRIEVE user profile information
        REMOVE any inappropriate content
        NOTIFY user about profile cleanup
    END IF

    LOG moderation action in admin logs
    RETURN "Content moderation completed"
}
```

7.4.4 Financial Reporting

GenerateFinancialReport(){

INPUT: adminID, reportType, startDate, endDate

BEGIN

 QUERY payments collection within date range

 GROUP payments by:

 payment method

 booking status

 date intervals (daily, weekly, monthly)

 CALCULATE financial metrics:

 totalRevenue = sum of all commissions

 totalTransactionVolume = sum of all payment amounts

 ownerPayouts = sum of all ownerAmount values

 averageCommissionPerBooking = totalRevenue / number of bookings

 BREAKDOWN by payment method:

 madaVolume, creditCardVolume, applePayVolume

 percentage distribution of payment methods

 GENERATE exportable formats:

 PDF report with charts and tables

 CSV file for spreadsheet analysis

 JSON data for API consumption

 RETURN financial report with all breakdowns and exports

}

8. Detailed System Design

This section provides a detailed description of the main components of the *Fadl Zahr* application based on the system's use cases. Each component is described in terms of its classification, definition, responsibilities, constraints, and composition. The goal is to show how the system requirements are transformed into implementable modules.

8.1 Classification

Renter Components:

Component	Classification	Component Definition and Responsibilities
Register	Function	Allow renters to create a new account
Verify Identity	Function	Confirms renter identity using (simulation) ID verification
Update Profile	Function	Allows renter to update personal details
Search for Cars	Function	Enables renters to search using filters such as price, model, and location
View Car Details	Display	Displays full car information including images, price, and availability
Book Car	Function	Submits booking request to owner
Make Payment	Function	Completes rental payment through simulated payment gateway
Schedule Booking	Function	Allows selecting rental start/end dates
Cancel Booking	Function	Allows renter to cancel booking request before approval
View Booking History	Display	Shows past and active bookings

Send Feedback	Function	Allows the renter to submit a rating and feedback, which is stored and shown to the owner and admin.
Chat with Owner	Function	Enables communication with the car owner
Receive Notifications	Function	Receives booking updates and payment confirmations

Table 6 - Renter Component

Owner Components:

Component	Classification	Component Definition and Responsibilities
Add Car Listing	Function	Allows owner to list a new car with full details
Upload Car Photos	Function	Enables uploading multiple images for the car
Set Price & Availability	Function	Sets daily rate and available dates for the vehicle
Edit / Delete Listing	Function	Updates or removes a car listing
Approve / Reject Booking	Function	Allows owner to accept or decline booking
Chat With Renter	Function	Enables direct conversation with renters
Receive Payment	Function	Confirms received payment
View Transaction History	Display	Shows all completed payments
View Feedback	Display	Displays renter reviews and ratings
View Booking Requests	Display	Shows incoming booking requests

Table 7 - Owner Component

Admin Component:

Component	Classification	Component Definition and Responsibilities
Monitor Activities	Function	Tracks system usage and user actions
Manage Reviews & Feedback	Function	Oversees and moderates customer feedback
Manage Vehicle Categories	Function	Edits and organizes car categories in the system
Edit Filter Options	Function	Adjust system filters (price ranges, models, types)
Send Announcements	Function	Allows the admin to broadcast system-wide announcements and important notifications to renters and owners regarding updates, warnings, or system events
Perform Security Audit	Function	Reviews system logs and checks security issues.
View Reports	Display	Displays analytical and statistical reports
Backup & Restore Data	Function	Saves and restores system data
Block Suspicious Account	Function	Restricts or locks suspicious users

Table 8 - Admin Component

8.2 Definition

Component	Definition
Renter Interface	Handles all renter functions including registration, identity verification, searching cars, booking, payment, and feedback
Owner Interface	Allows owners to list cars, manage bookings, view payments, and communicate with renters
Admin Panel	Enables administrative monitoring, security auditing, account control, and system configuration
Booking Management Module	Responsible for booking creation, modification, validation, and status updates
Payment Module	Processes rental payments using a simulated gateway and updates payment records
Notification System	Sends alerts to users regarding bookings, payments, and actions

Table 9 - Definition

8.3 Responsibilities

Component	Responsibilities
Renter Interface	Manage account creation, car searching, booking, payment, and renter feedback
Owner Interface	Manage car listings, handle booking requests, and track payments
Admin Panel	Ensure system stability, manage categories, security, feedback moderation, and user monitoring
Booking Management Module	Validate availability, prevent double bookings, update booking status

Payment Module	Process charges, calculating totals, confirm payment transactions
Notification System	Deliver real-time notifications to renters, owners, and admins

Table 10 - Responsibilities

8.4 Constraints

In this subsection, each component and its sub-components along with constraints, pre-conditions, and post-conditions will be illustrated through the following tables.

Common Interface – User Authentication

User Authentication (Login, Register, Password Reset)

Component	Constraints	Pre-Condition	Post-Condition
User Login	The user must have an existing account	Correct email/ID and password must be entered	User is directed to the homepage matching their role (Renter / Owner / Admin)
Registering New User	Email must be valid and not registered before	Required fields must be filled (Name, Email, Password)	A new user account is created and stored in the database
Reset Password	User must have an existing account	User must provide a valid email	A password reset link/code is sent and new password updated in the system

Table 11 - User Authentication Constraints

Renter Interfaces

Renter – Car Browsing & Booking Component Constraint

Component	Constraints	Pre-Condition	Post-Condition
Search Cars	Car listings must exist	User must be logged in	System displays a filtered list

			of available cars
View Car Details	Cannot view a deleted listing	A car must be selected	Display full details: price, availability, photos, owner profile
Book Car	The selected car must be available	User must have an active account	A pending booking request is created and sent to the owner
Cancel Booking	Booking must not be previously completed	Booking must exist	Booking is marked as canceled and removed from owner pending list
Send Feedback	User must have completed a trip	Booking must be completed	Feedback is stored and shown to the owner and admin
Chat With Owner	The trip request must exist	Renter selects a car/booking	Chat session is created for communication

Table 12 - Car Browsing & Booking Component Constraint

Owner Interfaces

Owner – Car Listing & Booking Management

Component	Constraints	Pre-Condition	Post-Condition
Add Car Listing	All fields must be filled	Owner must be logged in	A new listing is added to the marketplace
Upload Car Photos	Photo format must be valid	A car listing must exist	Photos are saved and linked to the listing
Set Price & Availability	Listing must belong to the owner	Owner selects listing	Updated pricing/availability is saved
View Booking Requests	Requests must exist	Owner must have at least one listing	Owner views all pending booking requests
Approve / Reject Booking	Request must be pending	A booking request must be selected	Status is updated → renter receives notification
Edit / Delete Listing	Listing cannot be booked at the moment	Listing must exist	Listing is modified or deleted successfully
Chat With Renter	Booking must exist	Select renter from booking	Chat session continues or starts

--	--	--	--

Table 13 - Car Listing & Booking Management constraints

Admin Interfaces

Admin – System Supervision

Component	Constraints	Pre-Condition	Post-Condition
Monitor Activities	None	Admin must be logged in	Admin views system-wide logs and activities
Manage Vehicle Categories	Category name must be unique	Admin selects category tab	Category added / edited / deleted
Edit Filter Options	None	Admin accesses filter page	Filters updated and saved
Block Suspicious Account	Account must exist	Admin selects user profile	Account is blocked and cannot log in
Backup & Restore Data	Admin privileges required	Admin selects backup/restore function	System backup created or restored
View Reports	None	Admin selects report type	System generates the requested report

Table 14 - System Supervision constraint

System-Level Interfaces (Shared Components)

Component	Constraints	Pre-Condition	Post-Condition
Notification System	User must have a valid ID	Notification triggered by event	User receives in-app or email notification
Payment Processing	Valid payment method required	User must select a booking to pay for	Payment is processed and stored in database
Review Display Engine	Feedback must exist	User selects car/owner profile	Feedback list rendered on screen

Table 15 - System-Level Interfaces constraints

8.5 Composition

Component	Subcomponents	Description
Renter Interface	Registration Form – Car Search Engine – Booking Form – Payment Page – Feedback Module – Chat Module	Supports full renter workflow from signup to booking and feedback
Owner Interface	Car Listing Form – Availability Manager – Booking Requests Page – Earnings Dashboard – Chat Module	Enables full control of listings and communication
Admin Panel	Review Manager – Category Manager – Audit Log – Reports Dashboard – Data Backup Tool	Enables administrative and security operations
Booking Management Module	Date Validator – Status Manager – Conflict Checker	Manages all booking actions and checks availability
Payment Module	Payment Processor – Fee Calculator – Receipt Generator	Handles simulated payment operations
Notification System	Email Sender – In-App Alerts – Status Update Triggers	Sends real-time updates to all user types

Table 16 - Composition

8.6 Uses/Interactions

In the following subsection, key interactions between system components are illustrated using Sequence Diagrams. Although the system includes seventeen interfaces, ten core interfaces were selected for modeling, as they represent the most essential and impactful user and system interactions.

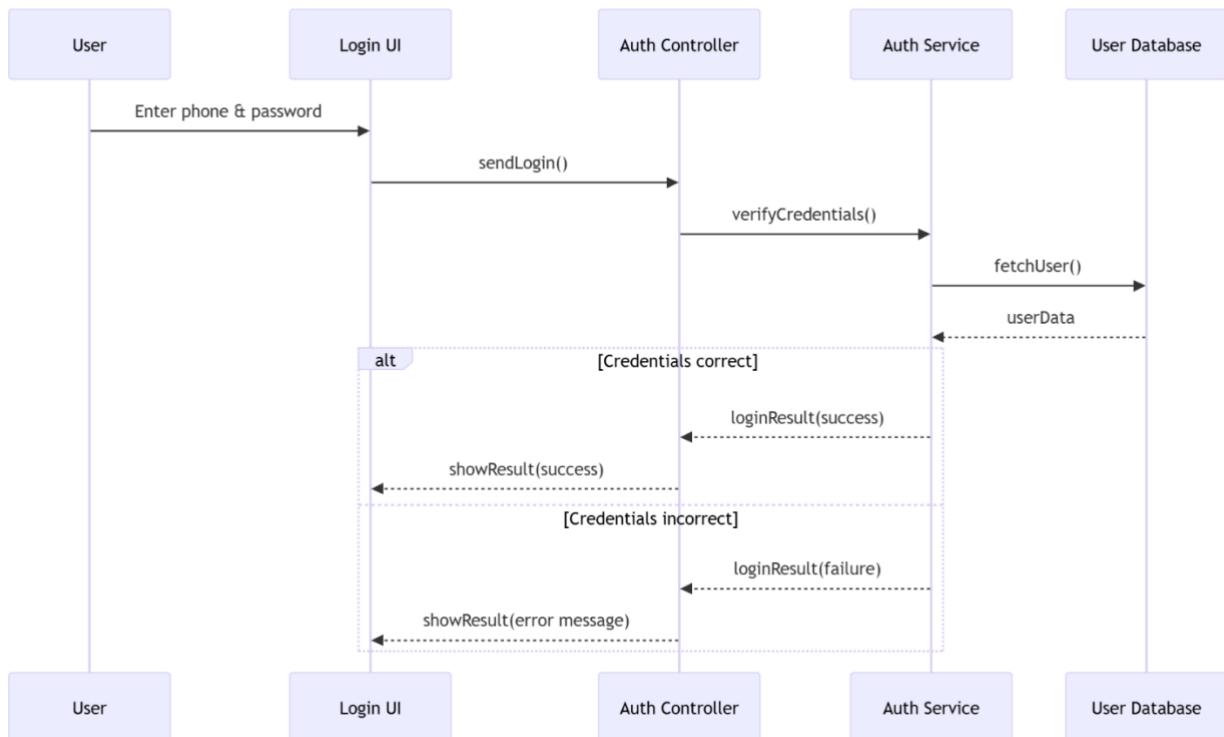


Figure 23 -User sign in

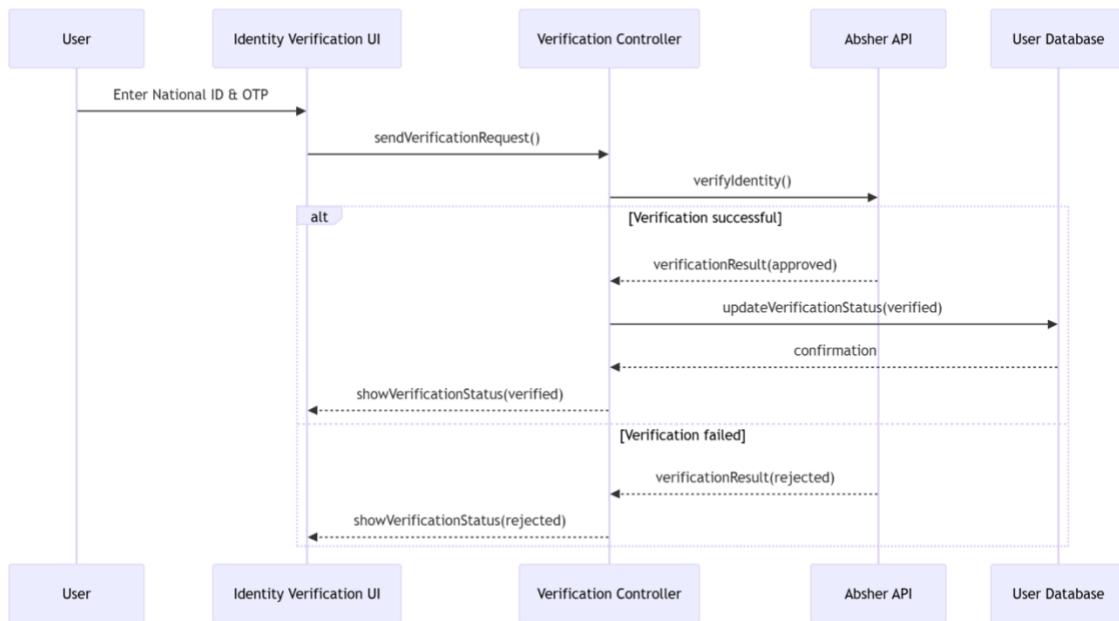


Figure 24 – User login

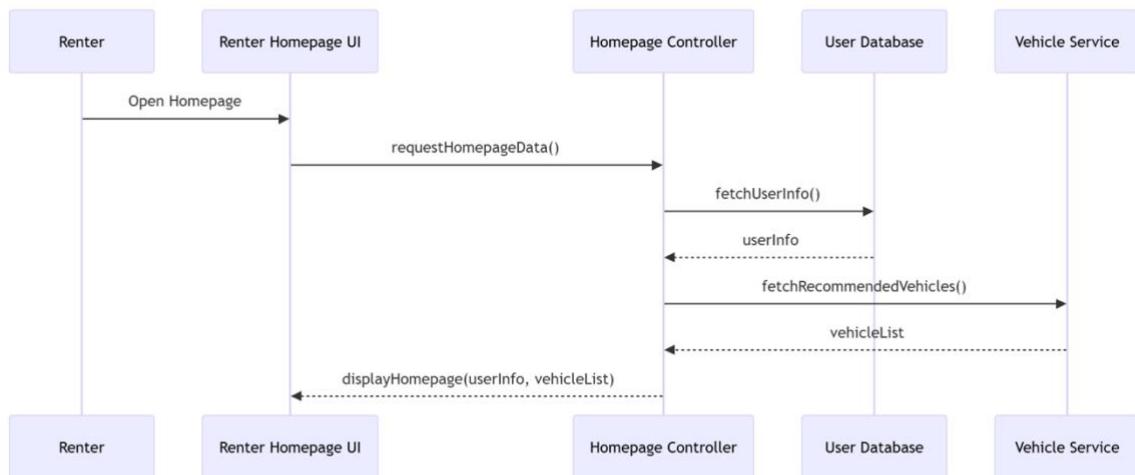


Figure 25-Renter home page

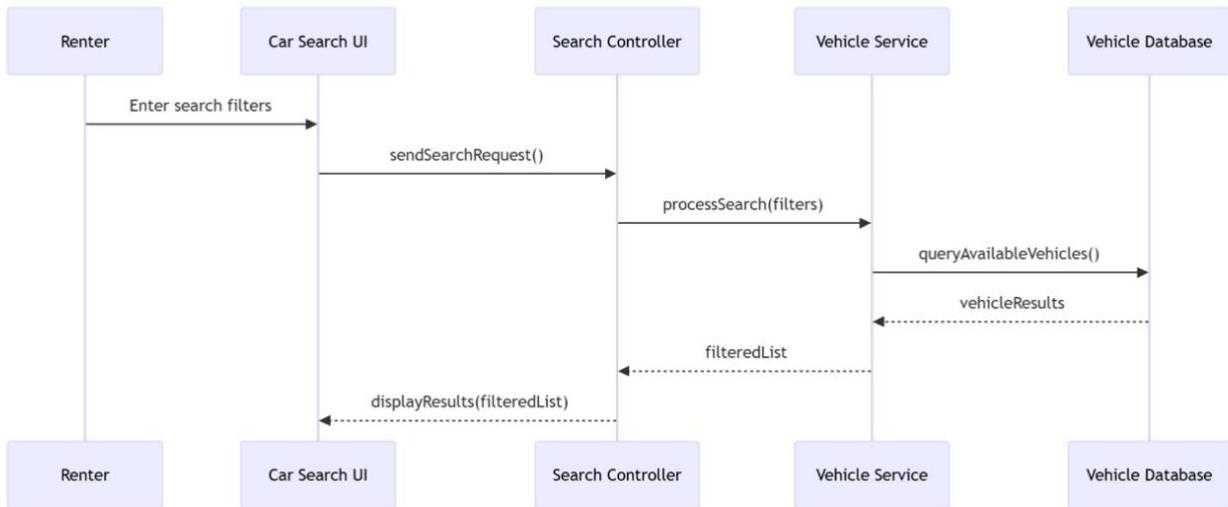


Figure 26 -Renter Search

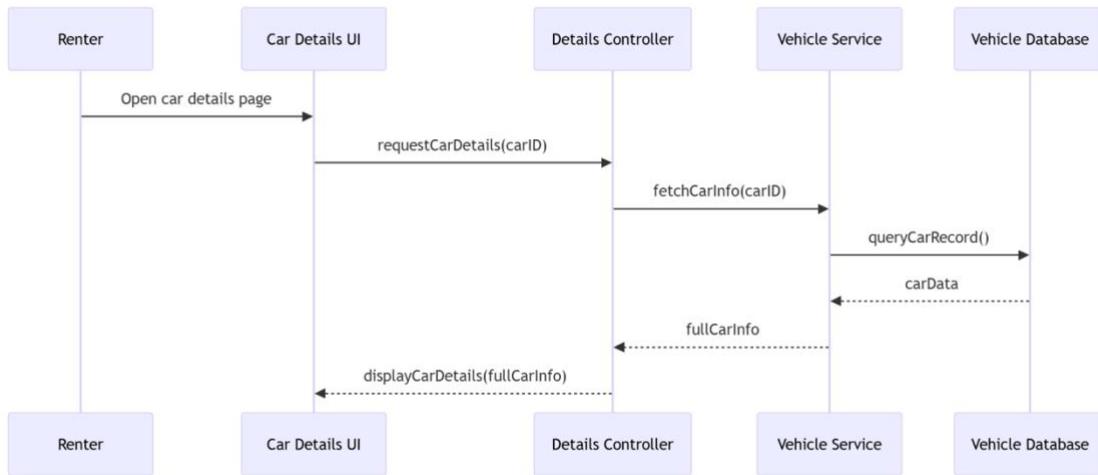


Figure 27 -Renter car

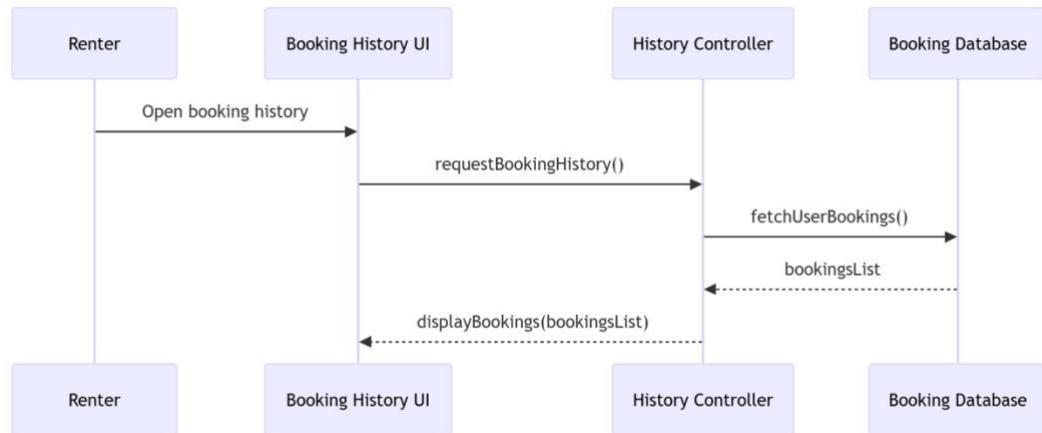


Figure 28 -Renter Booking

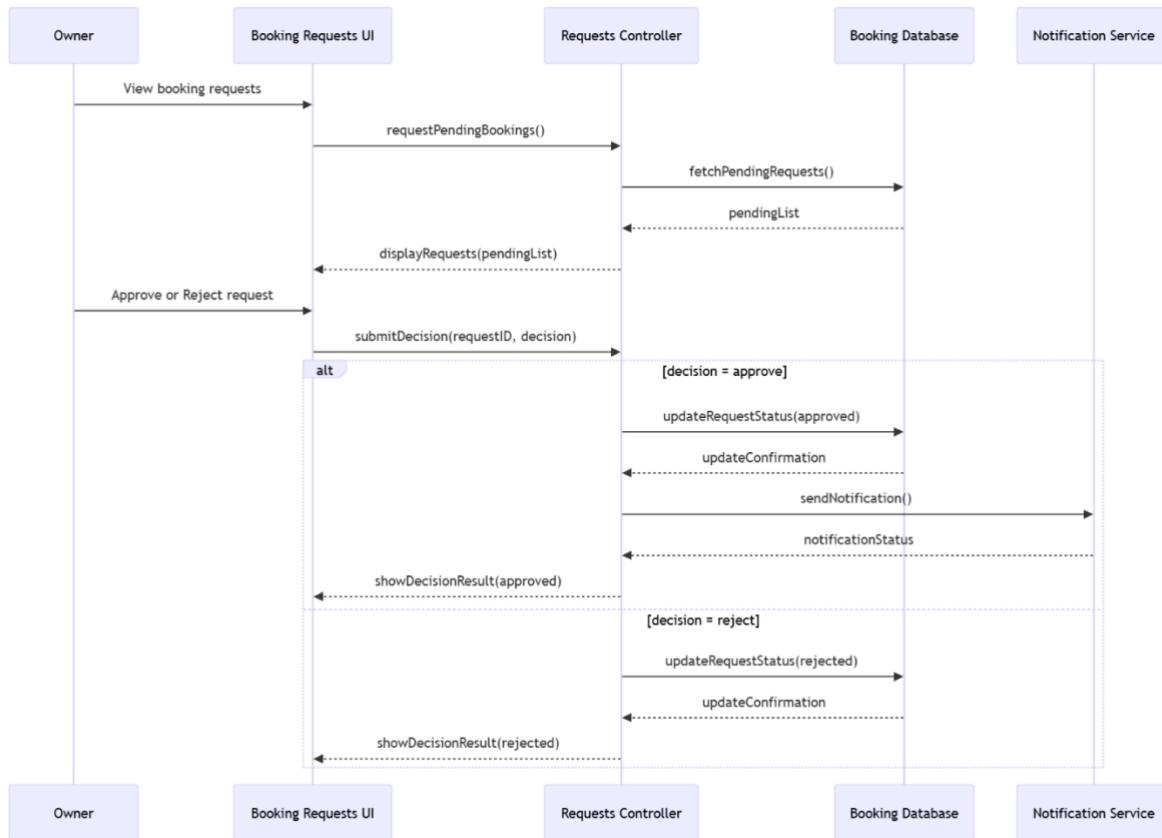


Figure 29 -Owner View

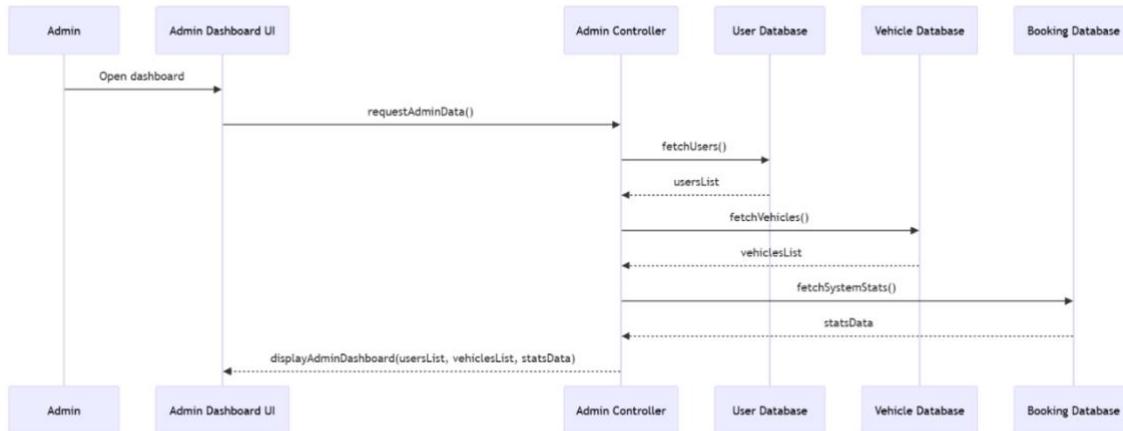


Figure 30 -Admin Dashboard

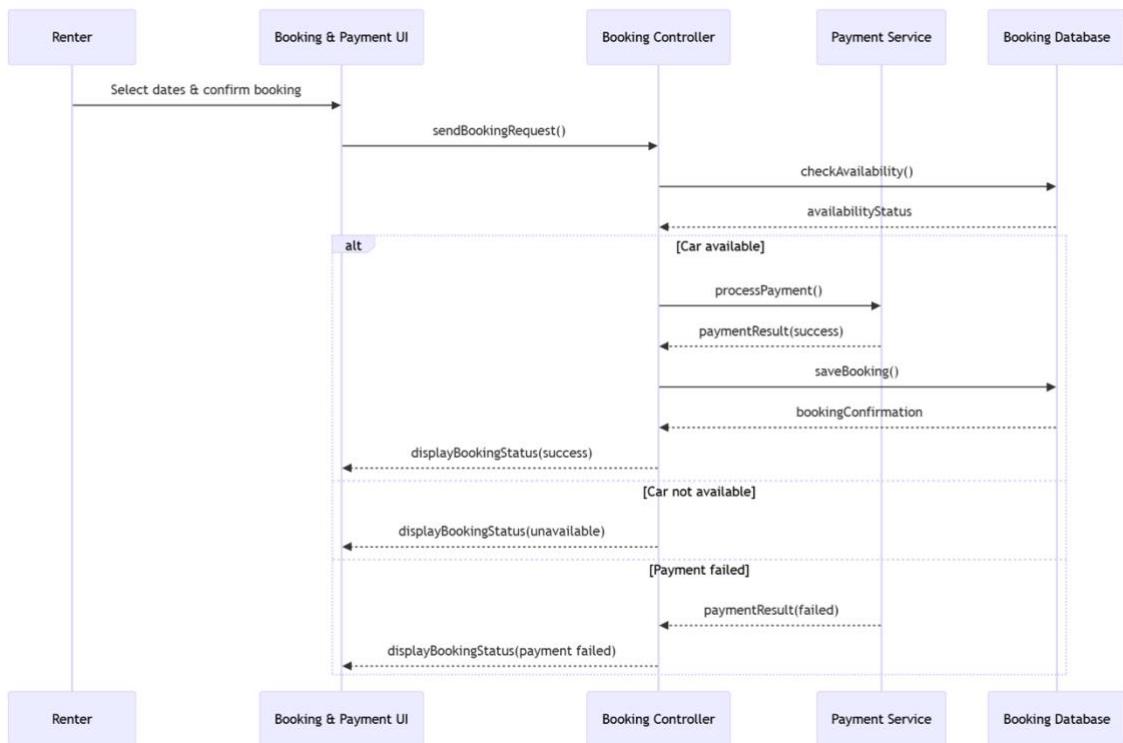


Figure 31 -Renter Select

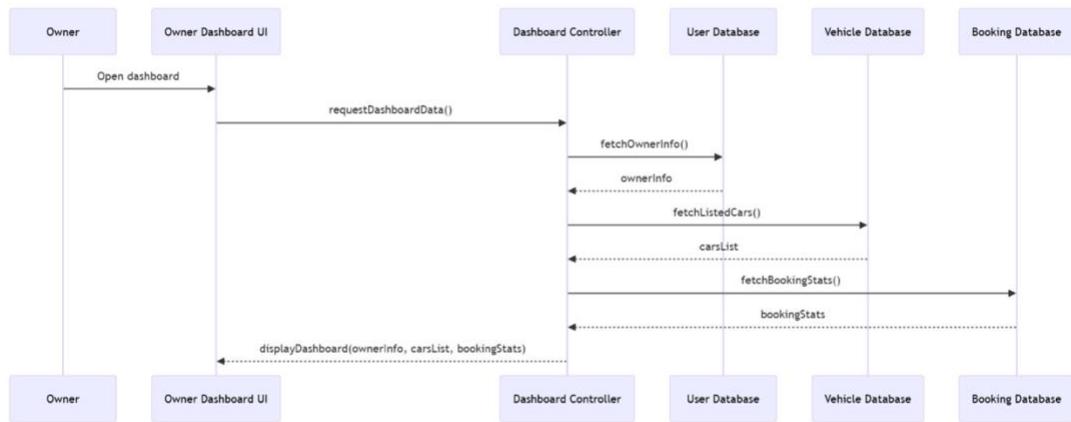


Figure 32 -Owner Dashboard

8.7 Resources

This subsection provides a brief explanation of the tables included in this section. The first table presents the system resources used by the Fadl Zahr application. The second table outlines possible race conditions that may occur during system operation and how they are handled. The third table summarizes potential deadlock situations, along with the solutions implemented to avoid or resolve them.

Type	Resource
Database	Firebase Firestore (user accounts, cars, bookings)
Storage	150–200 MB mobile storage for cached data and images
Operating System	Android 8.0+ / iOS 12.0
Server Environment	Node.js backend for API and verification processes
Network Requirement	Stable Wi-Fi or 4G/5G (minimum 2 Mbps)
Third-Party Services	Absher API (identity verification), Payment Gateway API

Table 17 - system resources

Race Condition	Solution
Two renters attempt to book the same car at the same time.	The system locks the booking record and confirms the earliest request. The second request receives a “Car already booked” response.
Owner updates car details while a renter is viewing the page.	The interface refreshes data automatically when new updates are detected.
Admin modifies user/car data while owner performs an action.	Use Firebase transactions to ensure atomic updates and prevent partial overwrites.
Payment confirmation interrupted due to unstable internet.	Payment service retries confirmation automatically once connection is restored.

Table 18 - possible race conditions

Deadlock Situation	Description	Solution
Outdated app version	User cannot access updated booking or verification features.	Force an update prompt: “Please update the app to continue.”
No internet connection	System cannot retrieve cars, bookings, or dashboard data.	Show “No connection” message and retry automatically when network returns.

Payment session freeze	Gateway waiting for confirmation after user closes the app.	Auto-cancel session after 60 seconds and notify user.
Car booked simultaneously by two users	Both users attempt to confirm at the final step.	Backend enforces a booking lock to prevent double-booking.
Missing permissions	App cannot access camera/location needed for features.	Show message instructing the user to enable permissions.

Table 19 – potential deadlock

8.8 Processing

This subsection describes how the Fadl Zahr subsystems process inputs, generate outputs, and handle exceptional cases. The processing behavior is derived from the system's interaction and sequence diagrams presented earlier in this document.

Each subsystem follows a structured processing flow:

- **Input Handling:**

Inputs from the user interfaces or external APIs are validated to ensure correctness and consistency before any operation begins.

- **Core Processing:**

The controllers coordinate the internal workflow, triggering the appropriate services to execute tasks such as verifying identities, retrieving vehicle data, creating bookings, processing payments, and updating system records.

- **Output Generation:**

Once processing is completed, the system returns structured outputs back to the interface or commits the results to the database, depending on the operation.

- **Exception and Error Handling:**

Abnormal conditions such as invalid user data, conflicting booking actions, failed payments, or unavailable network resources are captured and handled gracefully. Clear feedback messages are returned to guide the user toward corrective actions.

8.9 Interface/Exports

This subsection summarizes the interfaces and exports interactions that connect the main subsystems of the Fadl Zahr application. Each subsystem communicates through well-defined inputs, outputs, and operational conditions to ensure consistent and reliable coordination across the entire system.

The user interfaces interact with controllers that manage the system's internal logic, while backend services exchange data with the database and external APIs such as Absher and the payment gateway. These interfaces define how information is transferred, validated, and returned, ensuring that all booking, verification, and management operations follow the expected constraints and behaviors.

The interaction flows illustrated in the earlier sequence diagrams demonstrate how each interface collaborates with other components, showing a clear mapping of requests, processing steps, and resulting outputs across the system.

8.10 Detailed Subsystem Design

This subsection describes the internal structure and behavior of the primary subsystems within the Fadl Zahr application. Each subsystem is designed using a layered approach that separates responsibilities across controllers, services, and database components. This structure ensures clear communication between modules and supports maintainability and scalability as the system grows.

Controllers handle user and system requests, validate inputs, and coordinate the required operations. Services perform the core logic of the application, including identity verification, vehicle filtering, booking creation, payment processing, and status updates. Database components manage all persistent data such as user accounts, vehicle listings, booking records, and system logs.

The internal workflows of these subsystems are illustrated in the sequence diagrams provided earlier. These diagrams demonstrate how data flows between interfaces, backend components, and external APIs, showing the step-by-step execution of each operation. This design ensures consistent processing behavior, smooth integration between modules, and reliable execution of the system's functionalities.

8.11 Detailed System Interfaces

8.11.1 common interfaces

8.11.1.1 Sign\Login in Interface

- A field for Email / National ID.
- A Password field with a visibility toggle.
- A User type selector (Renter, Car Owner, or Admin).
- A main Log in button to enter the system.
- A link for Forgot Password.
- Optional sign-in methods using Apple or Google.
- A link to create a new account if the user doesn't have one

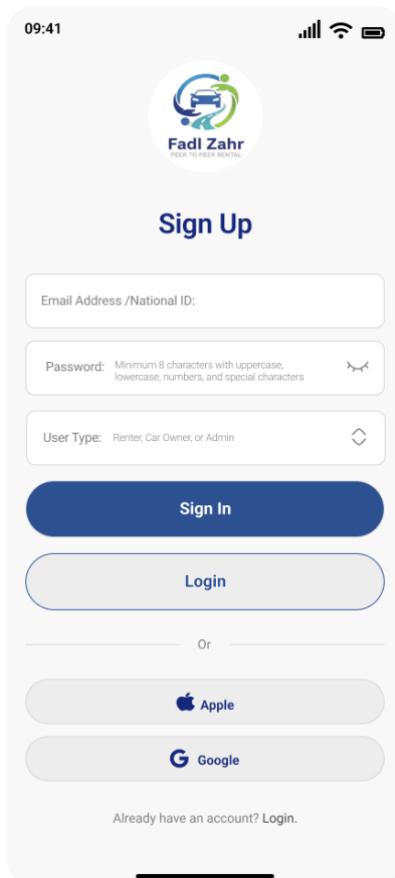


Figure 33 - Sign-up Interface

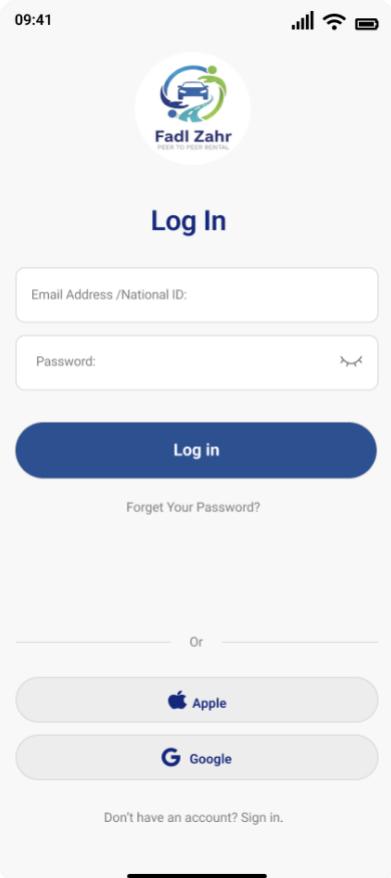


Figure 34 - Log-in Interface

8.11.1.2 Identity Verification Interface

This screen is used to verify the user's identity. It includes a field to enter the Saudi National ID, followed by a button to Verify with Absher. The Absher logo is shown to indicate the verification method. A Back to Sign in option is also provided for users who want to return to the login page

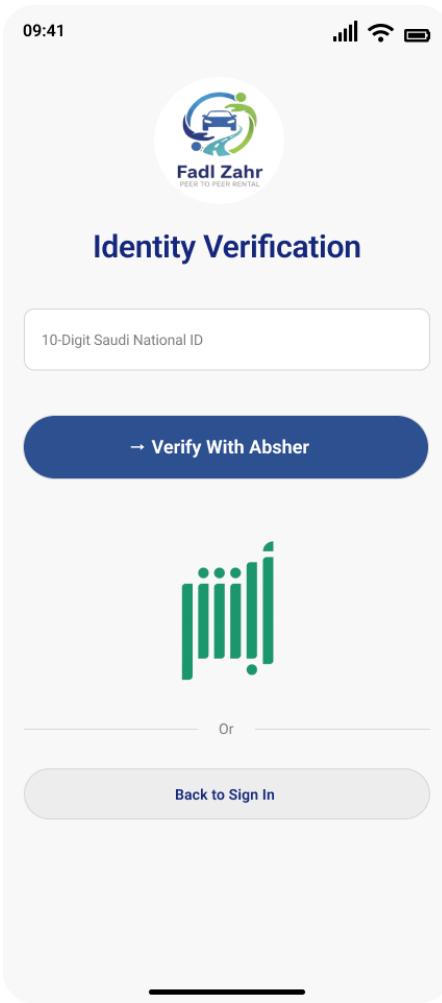


Figure 35 - Identity verification interface

8.11.1.3 Forget My Password Interface

This section includes the steps used to help the user reset their password.

1. Reset Password Request Screen

This screen allows the user to enter the email associated with their account to receive a password-reset link. It also includes a Return to Log in option and a Continue button to proceed.

2. Verification Code Screen

After submitting the email, the user is asked to enter a 6-digit verification code sent to their email. A Continue button is provided to confirm the code, along with an option to resend the OTP if needed.

3. New Password Screen *Figure- New password. Figure- Verification code*

In this step, the user can create a new password by entering it twice for confirmation. The screen includes a Continue button to save the new password and a Cancel option to go back

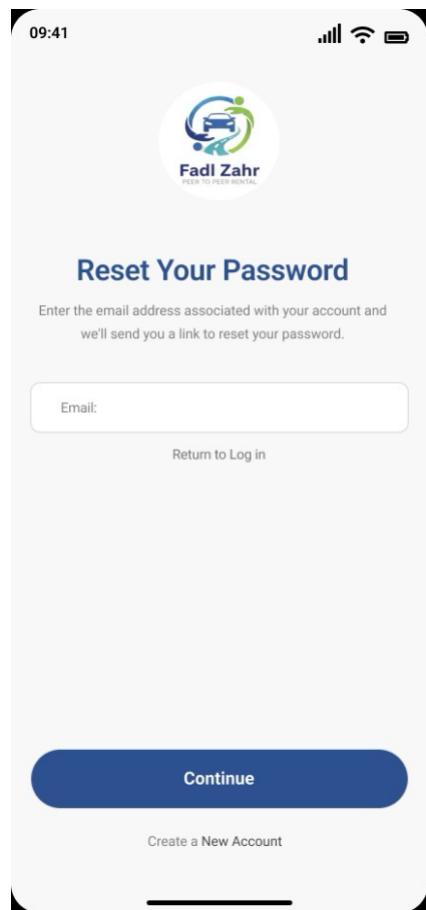


Figure 36- Request Reset

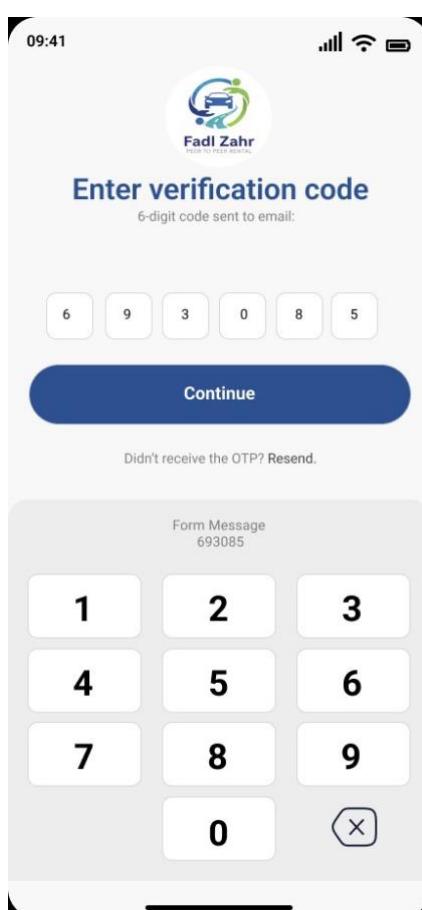


Figure 37- Verification code

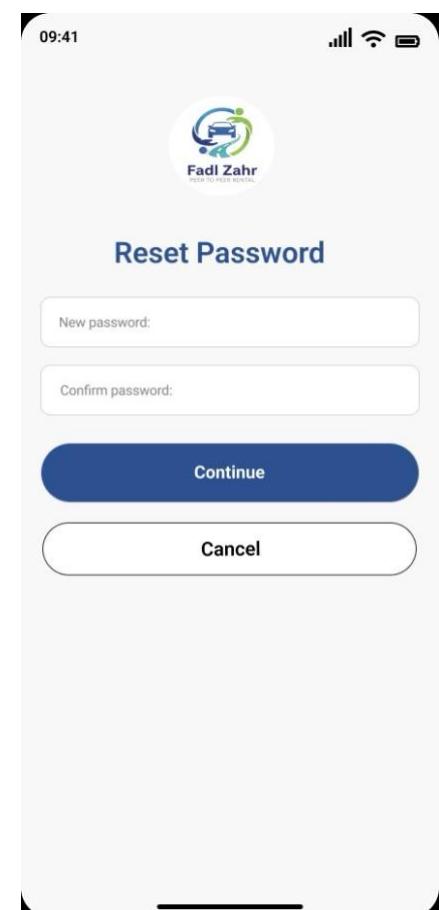


Figure 38- New password

8.11.2 Renter Interfaces

8.11.2.1 Renter Homepage Interface

The renter homepage is the main screen where users can browse available cars and explore options recommended for them.

- At the top of the interface, a search bar is provided to allow the renter to quickly look for a specific car or model. Beside it, notification and favorite icons appear to give the user easy access to saved cars and new alerts.
- Below, the screen displays a “Recommend for You” section that shows different car cards based on the user’s preferences. Each card includes the car’s name, rating, location, price per day, and a button that allows the renter to book the car directly.
- At the bottom of the interface, a navigation bar is available to switch between the home page, messages, reservations, and the user profile

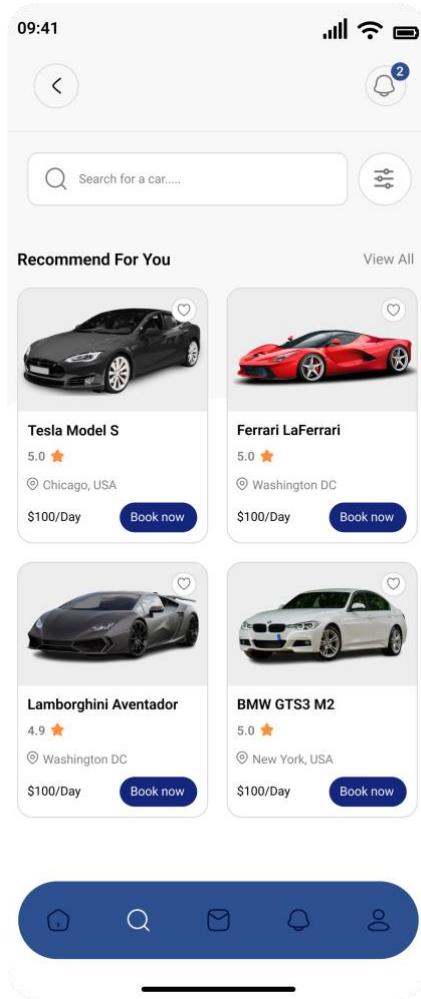


Figure 39 - Renter homepage interface

8.11.2.2 Car Search & Discovery Interface

The car search and discovery interface helps the user explore available vehicles using different criteria. The screen begins with a search bar that allows the user to type any car name or model, followed by filter options that make the search more specific. When the user opens the filter panel, they can choose the type of car they are looking for, adjust the price range, select the rental duration, and pick both the start and end dates from the calendar. The filter page also lets the user specify the car location, preferred colors, seating capacity, and fuel type. Once the user sets their preferences, the system displays a list of matching cars with their information such as rating, seats, price per day, and location. This interface helps the renter quickly find the most suitable car based on their needs and makes the browsing experience smoother and more personalized.

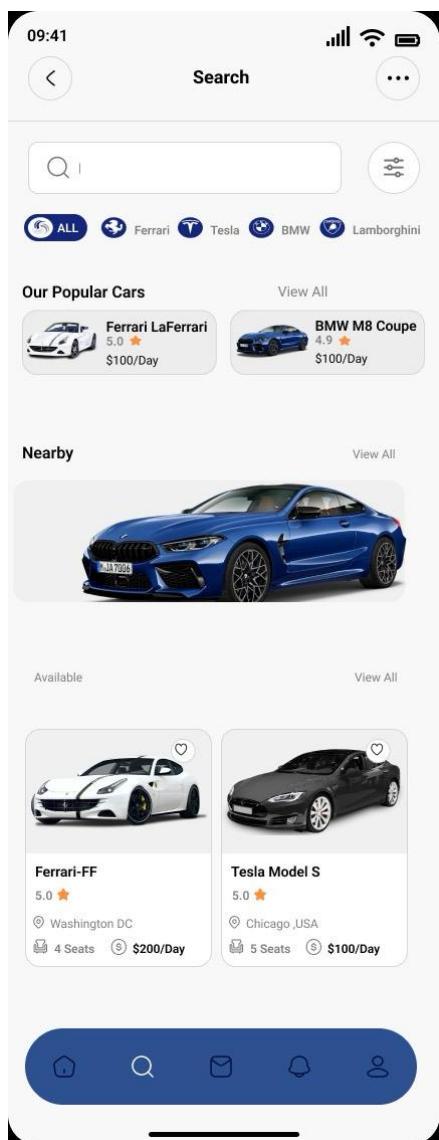


Figure 40 - Car search

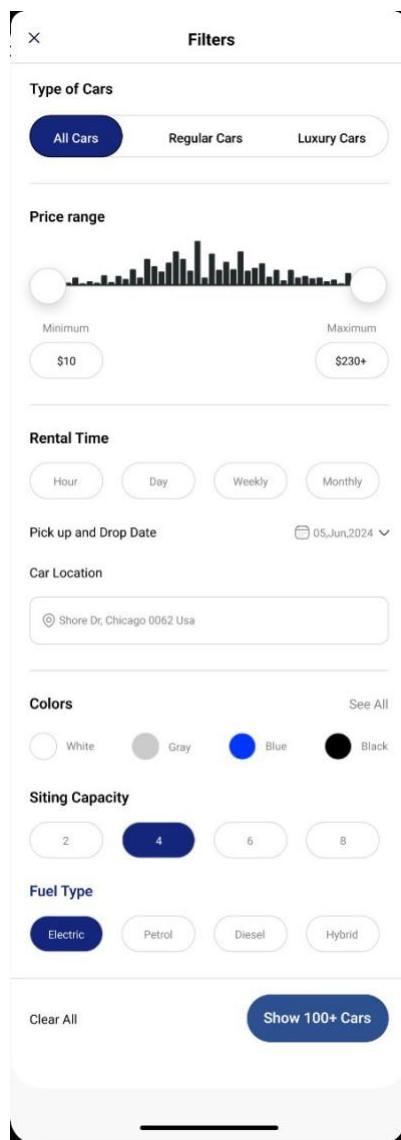


Figure 41 - Car filter

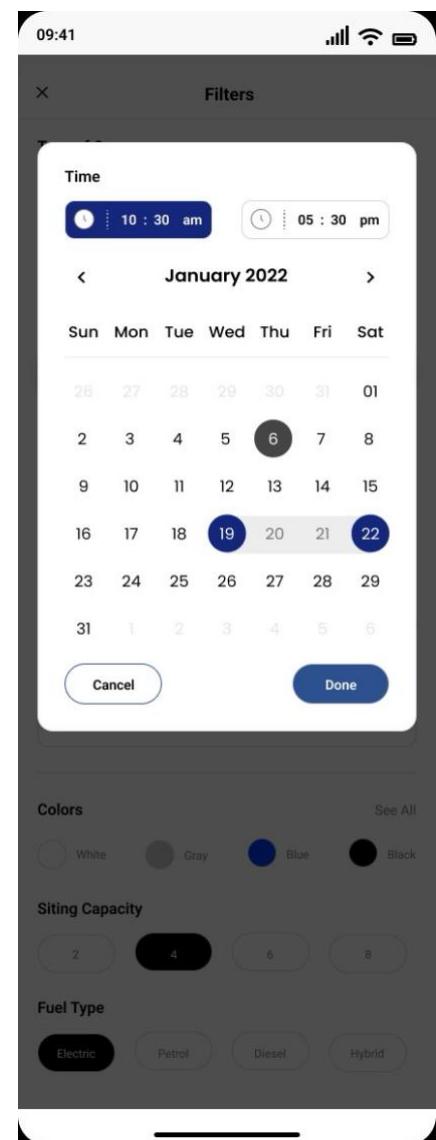


Figure 42 - Date Calendar

8.11.2.3 Car Details Interface

The car details interface provides the user with a complete view of the selected vehicle.

- At the top of the screen, images of the car are displayed to help the user see the exterior clearly. Below the images, the car's name, rating, and a short description appear, followed by the owner's information, including their name, profile picture, and response rate. The interface also presents the main features of the car, such as seating capacity, engine power, maximum speed, driving assistance options, and fuel details. Pricing information is shown in separate labels, including the daily price, seasonal price, promo rate, and security deposit.
- At the bottom of the page, the user can check availability through a calendar that shows the free and booked dates for the car. After reviewing all the details, the user can proceed with the rental by clicking the “Book Now” button

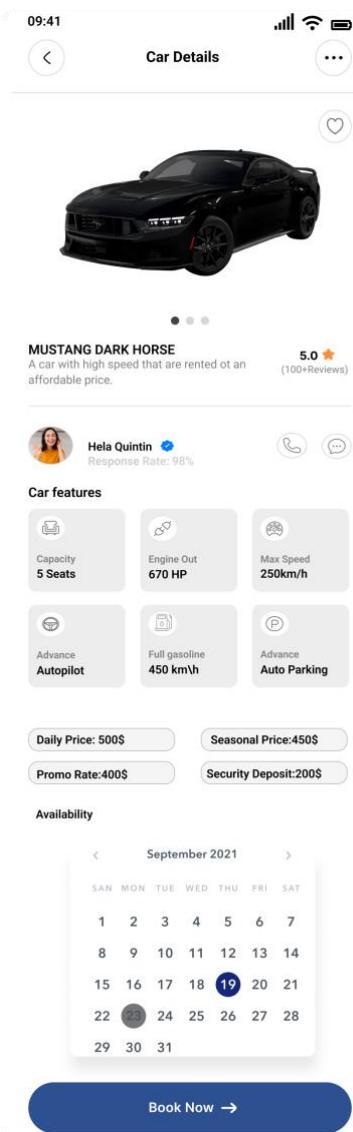


Figure 43 - car Details Interface

8.11.2.4 Booking & Payment Interface

The booking and payment interface provides the user with a clear summary of their rental before completing the payment.

- At the top of the screen, the car name, image, and location are displayed, followed by the selected rental period showing the start and end dates. The total amount is presented so the user can review the full cost before proceeding. Below the summary, the interface offers access to available discounts through an offers section. The user can then choose a preferred payment method from a list of saved credit and debit cards, including options such as Visa, Apple Pay, and Mada, with the ability to add a new card if needed. An optional toggle for additional insurance coverage is also available.
- At the bottom, the interface shows the final price and provides a button to proceed to the payment confirmation screen.

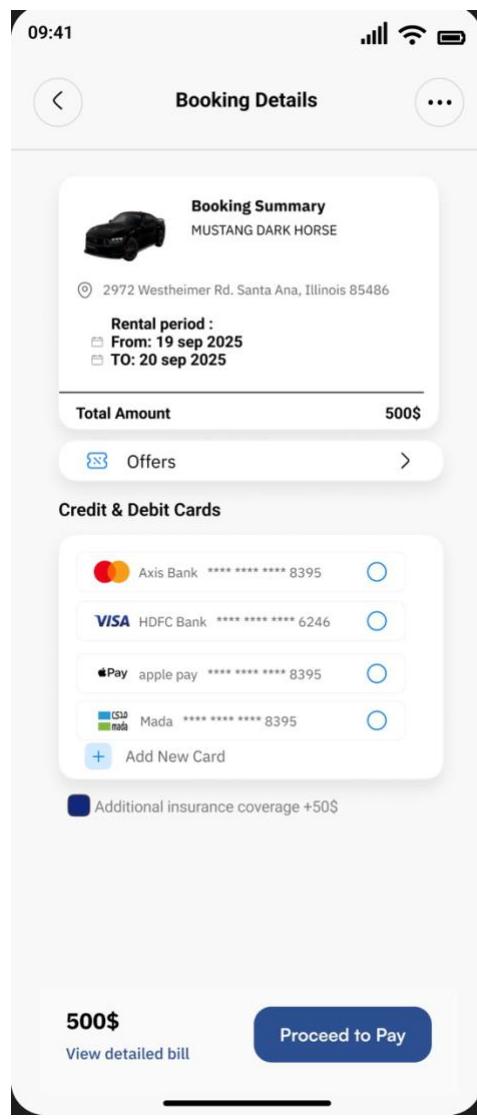


Figure 44 - Booking and payment interface

8.11.2.5 Booking History Interface

The booking history interface shows the renter all their past and current reservations in an organized way. At the top of the screen, three tabs are available current, upcoming, and completed allowing the user to switch easily between different booking categories. Each reservation card displays the car image, name, booking dates, renter or owner name, price, and the status of the booking such as “confirmed” or “completed”. The interface also provides quick actions under each reservation, including the option to re-book the same car or write a review for completed rentals. This layout allows the user to keep track of their rental activity and manage past or future trips smoothly.

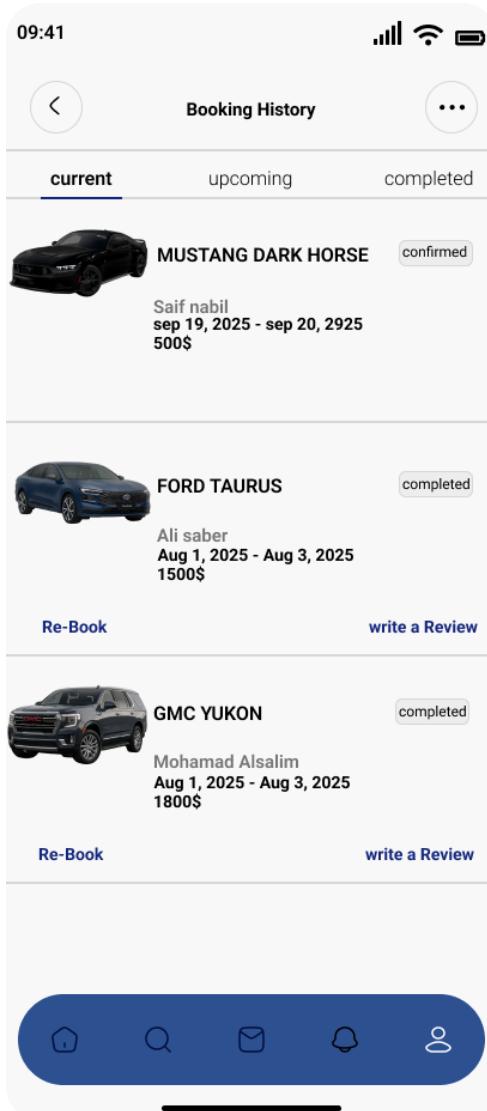


Figure 45 - Booking history interface of the application

8.11.3 Owner Interfaces

8.11.3.1 Owner Dashboard Interface

The owner dashboard interface gives the car owner a clear overview of their performance and booking activity.

- At the top of the screen, a graph displays the owner's earnings, with the ability to switch between weekly, monthly, or yearly views. Below the graph, the dashboard shows the number of active bookings and upcoming bookings, allowing the owner to quickly monitor their current workload. The interface also highlights recent ratings left by renters, showing their names, profile pictures, rating scores, and comments. This section helps the owner understand user satisfaction and service quality.
- At the bottom, the notifications area displays new alerts related to bookings or system updates, making it easier for the owner to stay informed about any changes.

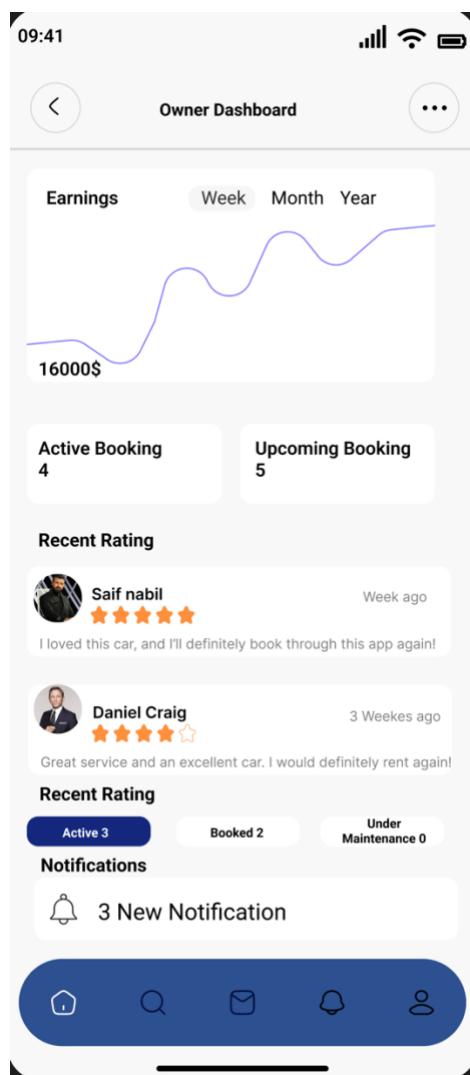


Figure 46 - Owner Dashboard Interface

8.11.3.2 Car Listing Management Interface

This interface allows car owners to fully edit their vehicle listing.

- The top section contains the car image with an option to upload or modify the photo. A favorite icon enables marking the listing for quick access. Below the image, the owner can add or update the car description. The Car Details Section provides fields for the car's make, model, year, and plate number. Additional editable feature cards let owners specify or modify technical details such as capacity, engine type, maximum speed, and charging or safety features. The Pricing section allows setting daily, seasonal, promotional rates, and required security deposits. Lastly, the Availability area enables owners to block specific dates or days when the car cannot be rented.

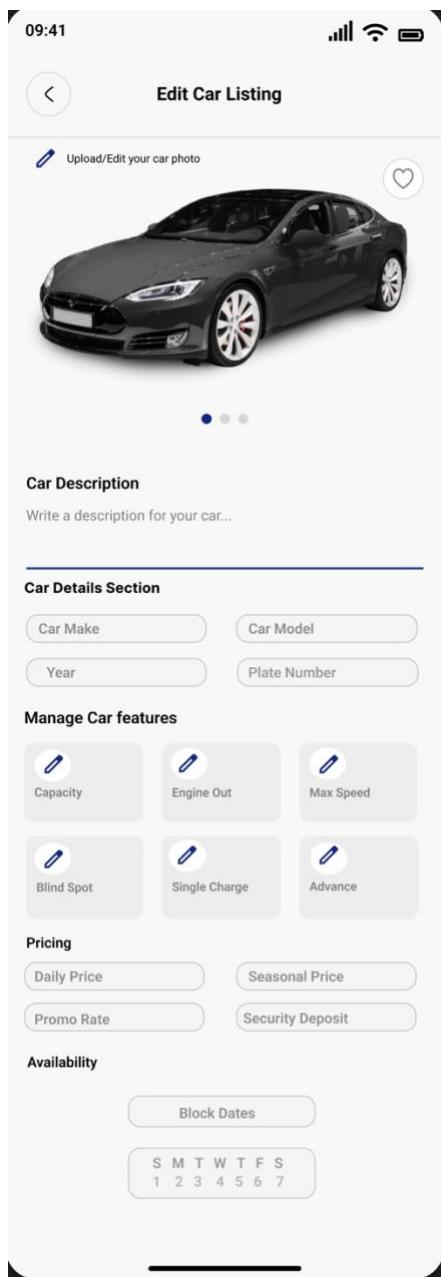


Figure 47 - Car Listing Management Interface

8.11.3.3 Booking Requests Interface

This interface displays all incoming booking requests submitted to the car owner in a clear and organized format.

- At the top, the owner can switch between different request categories such as Pending, Accepted, and Declined, making it easy to manage ongoing and past requests. A search bar allows filtering by renter name or car model. Each request card shows the renter's name, the car being booked, the booking dates, and the total price, along with an action tag indicating the current status.
- The owner can immediately approve or decline any request using the buttons provided.
- A navigation bar at the bottom gives quick access to other main sections of the app.

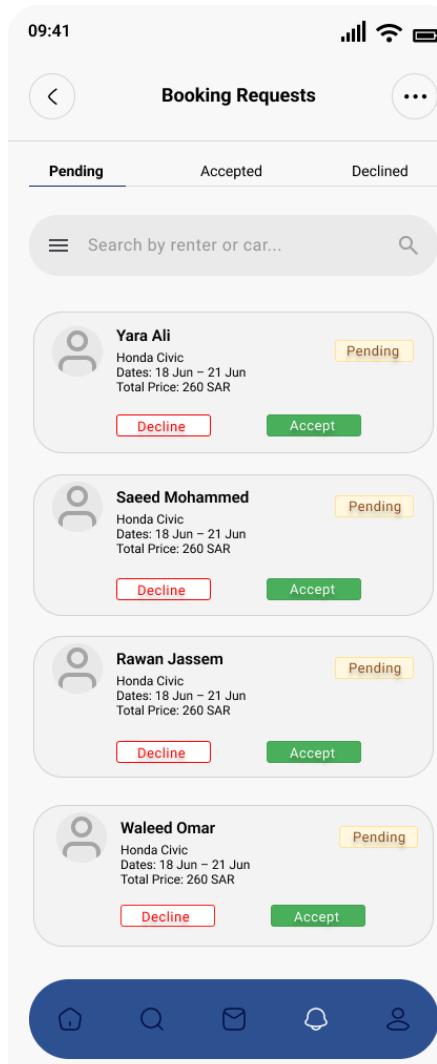


Figure 48 - Booking request Interface

8.11.3.4 Earnings & Transaction History Interface

This interface is designed to help car owners track their financial activity within the platform. The total earnings amount is displayed at the top, followed by tabs allowing owners to view transactions by day, week, month, or all-time. Each transaction includes details such as the payment type, car name, date and time, payment method, and the amount added or refunded. Positive amounts appear as earnings, while negative amounts represent refunds. This organized view helps owners understand their revenue flow and easily review past payments.

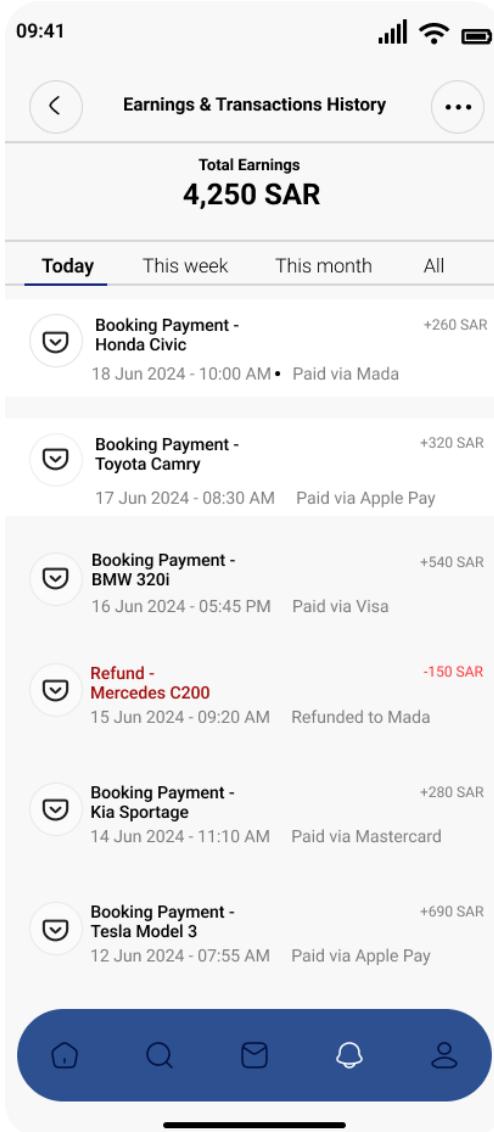


Figure 49 - Earnings & Transaction History Interface

8.11.4 Admin Interfaces

8.11.4.1 Admin Dashboard Interface

The dashboard provides the admin with a real-time overview of system activity. It begins with summary cards showing the total number of users, active cars, and current bookings. Below this section, a System Overview area displays daily statistics related to new bookings, issues, or listings. A graph illustrates booking trends over the past seven days, helping track usage patterns. At the bottom, the Recent Activities section logs actions performed by users or the system, such as new listings, reports, cancellations, or automated checks. This dashboard supports fast decision-making and efficient system monitoring.

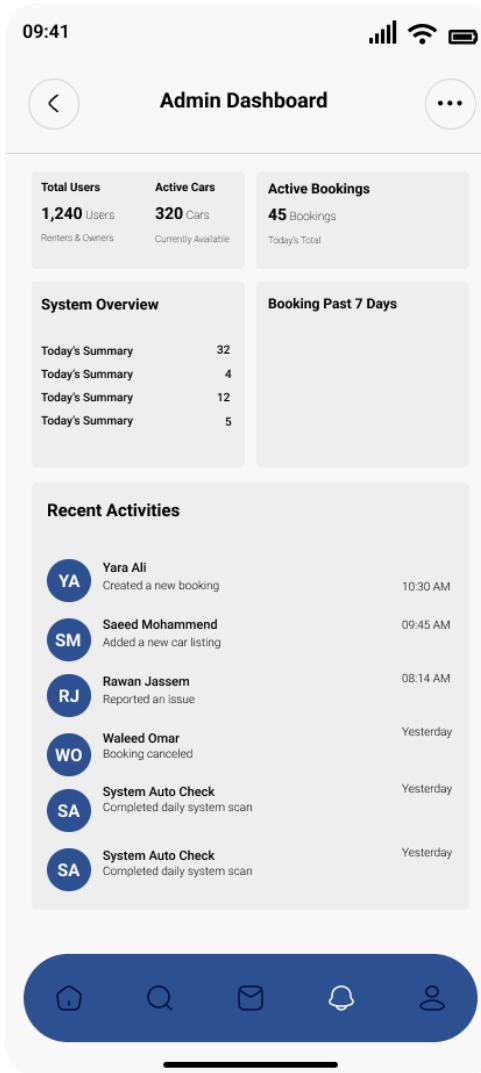


Figure 50 – Admin Dashboard Interface

8.11.4.2 User Management Interface

This interface gives administrators complete control over user accounts. A search bar at the top allows searching by name, email, or user ID. The table displays each user's profile initials, their full name, account status (such as Active, Suspended, or Banned), and their verification status (Verified, Pending, or Rejected). Admins can change the account status using the dropdown menu next to each user. This interface ensures that the platform maintains user authenticity, prevents misuse, and allows quick action when issues arise.

Name	Account Status	Verification Status
YA Yara Ali	Banned	Rejected
SA Saeed Ahmad	Active	Verified
RJ Rawan Jassem	Banned	Rejected
WO Waleed Omar	Active	Verified
M Maichel	Suspended	Pending
A Anna	Suspended	Pending

Figure 51 – User Management Interface

8.11.4.3 Content Moderation Interface

This screen is dedicated to the moderation of user-generated content on the platform, such as car listings and profiles. Moderators are presented with a queue of flagged items, detailing the content issue (e.g., "Duplicate listing", "Unclear image", "Fake profile"), the author, and the current status. The primary action for each item is "Review," allowing a moderator to inspect and make a final decision to approve or reject the content, ensuring platform quality and safety.

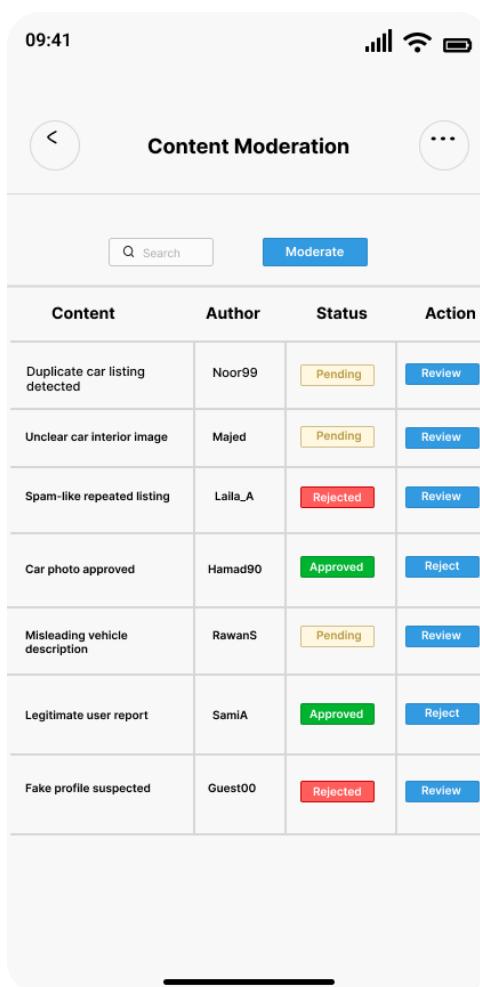


Figure 52- Content Moderation Interface

8.11.4.4 Reports & Analytics Interface

This interface enables admins to generate, view, and manage platform reports. A searchable table lists various report types such as User Activity, Vehicle Listings, Payments, and Ratings. Each report entry shows the last generated date, download count, and a View button for accessing or exporting the report. Below the reports section, the Analytics area displays visual charts, such as monthly active users, providing valuable insights into platform performance and user behavior. This interface helps administrators make data-driven decisions and monitor system growth.

The screenshot shows a mobile-style interface titled "Reports & Analytics". At the top, there is a header bar with the time (09:41), signal strength, and battery level. Below the header is a navigation bar with a back arrow, the title "Reports & Analytics", and a three-dot menu icon. Underneath is a search bar with a magnifying glass icon and the word "Search", followed by a blue "Moderate" button. The main content area is divided into two sections: "Reports" and "Analytics". The "Reports" section contains a table with four columns: Report, Last Generated, Downloads, and a "View" button. The table has five rows, each corresponding to a report type: User Activity, Vehicle Listings, Payments, and Ratings. The "Analytics" section shows a small portion of a chart with months Feb, Mar, Apr, and May at the bottom.

Report	Last Generated	Downloads	
User Activity	Jan 15, 2024	-	<button>View</button>
Vehicle Listings	Dec 1, 2023	↓ 35	<button>View</button>
Payments	Nov 20, 2023	↓ 42	<button>View</button>
Ratings	Dec 28, 2023	↓ 12	<button>View</button>

Figure 53 - Reports & Analytics Interface

8.11.5 Rating & Review Interfaces

8.11.5.1 Rate Experience Interface

This interface appears after a renter completes a booking, allowing them to evaluate their experience. The renter can select an overall star rating and write an optional text review of up to 500 characters. The interface also contains separate rating sections for the Owner, Car, and Communication, each using a star rating system. Once submitted, the ratings contribute to the car and owner's public score. This interface supports transparency and helps maintain quality across the platform

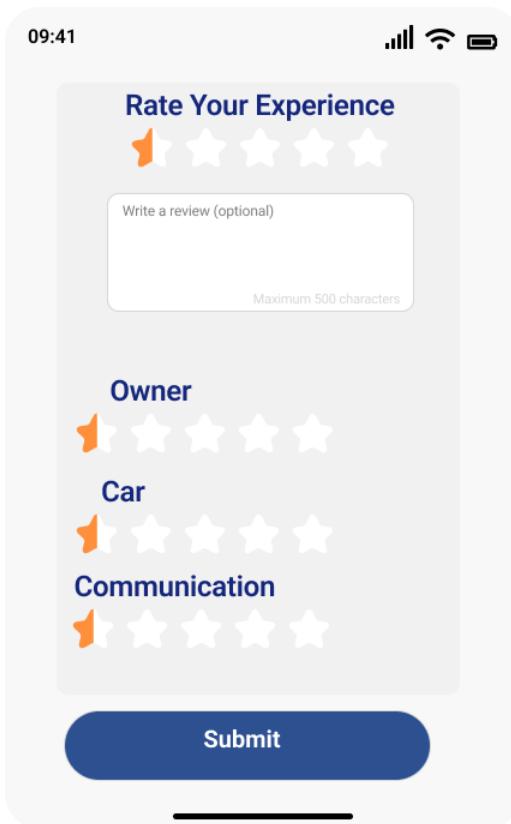


Figure 54 – Rate Experience Interface

9. Other Design Features

All the design features of the “Fadl Zahr” system have been illustrated in a detailed manner in the above Sections and Subsections.

10. Requirements Traceability Matrix

Association ID in SRS		Technical Assumption(s) / Customer Need(s)	Functional Requirement	User	Association ID in SDS	System Component(s)
3.1.1 Common Functionalities	3.1.1.1	Users need secure access to the system using valid credentials.	User Login	Renter / Owner / Admin	7.1.1	Sign-In Interface / Authentication Service
	3.1.1.2	User identity must be verified before accessing renting features.	Identity Verification (Abshar Integration)	Renter / Owner	7.1.2	Identity Verification Module / Abshar Adapter
3.1.2 Renter Functionalities	3.1.1.2	Renters need a starting point to access main services	Renter Home page Display	Renter	8.11.2.1	Renter Homepage Interface

			easily.					
	3.1. 2.2	Renter s need to search for availa ble cars based on filters.	Search Availa ble Vehicl es	Rente r	7.2. 1	Search Availab le Vehicle Module		
	3.1. 2.3	Renter s must view full car infor matio n before booki ng.	View Vehicl e Detail s	Rente r	8.11 .2.3	Car Details Interface		
3.1.2 Renter Func tionalities	3.1. 2.5	Users need to revie w previ ous booki ngs for refer ence.	Boo king Hist ory	Re nte r	8.11 .2.5	Booking History Interface		
3.1.3 Owner Func tionalities	3.1.3.3			Owners need to accept or reject booking requests efficientl y.	Manage Booking Requests	Owner	8.11.3. 3	Booking Requests Interface
	3.1.3.1			Owners require a dashboar d to manage listings and view activities.	Owner Dashboa rd	Owner	8.11.3. 1	Owner Dashboard Interface

	3.1.4.1	Admin needs visibility over system activity, users, and reports.	Admin Dashboard	Admin	8.11.4.1	Admin Dashboard Interface
--	---------	--	-----------------	-------	----------	---------------------------

Table 20 – Requirements Traceability Matrix