

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 – 2025/2026

Software Requirements Specification

For

فضل ظهر

Version 1

CS 411 – Software Engineering
Group 2

Subervised by:

Dr. Enas El-sayed El-sharawy

1 November 2025

#	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 Alghamdi	2230003568	Member

Table of Content

Revision History.....	4
Table of Table.....	5
Table of Figures.....	7
1. Introduction.....	8
1.1 Purpose.....	8
1.2 Document Conventions.....	8
1.3 Intended Audience and Reading Suggestions.....	9
1.4 Product Scope.....	10
1.5 Definitions, Acronyms, and Abbreviations	11
1.6 References.....	12
2. Overall Description.....	13
2.1 Product Perspective.....	13
2.2 Product functions.....	14
2.3 User Classes and Characteristics.....	16
2.4 Operating Environment.....	16
2.5 Design and Implementation Constraints.....	17
2.6 User Documentation.....	18
2.7 Assumptions and Dependencies.....	18
3. External Interface Requirements.....	19
3.1 User Interfaces.....	19
3.1.1 Common Interfaces.....	19
3.1.1.1 Sign In Interface.....	19
3.1.1.2 Identity Verification Interface.....	20
3.1.1.3 Forget My Password Interface.....	20
3.1.2 Renter Interfaces.....	20
3.1.2.1 Renter Homepage Interface.....	20
3.1.2.2 Car Search & Discovery Interface.....	21
3.1.2.3 Car Details Interface.....	21
3.1.2.4 Booking & Payment Interface.....	22
3.1.2.5 Booking History Interface.....	22
3.1.3 Car Owner Interfaces.....	22
3.1.3.1 Owner Dashboard Interface.....	22
3.1.3.2 Car Listing Management Interface.....	23
3.1.3.3 Booking Requests Interface.....	23
3.1.3.4 Earnings & Transaction History Interface.....	23
3.1.4 Admin Interfaces.....	23
3.1.4.1 Admin Dashboard Interface.....	23
3.1.4.2 User Management Interface.....	24

3.1.4.3 Content Moderation Interface.....	24
3.1.4.4 Reports & Analytics Interface.....	24
3.1.5 Rating & Review Interfaces.....	25
3.1.5.1 Rate Experience Interface.....	25
3.2 Hardware Interfaces.....	25
3.3 Software Interfaces.....	25
3.4 Communication Interface.....	27
4. System Features.....	29
4.1 System Features for renters.....	30
4.2 System Features for Owner.....	35
4.3 System Features for Admin.....	40
5. Non-functional requirement.....	43
5.1 Performance Requirements.....	43
5.2 Safety Requirements.....	43
5.3 Security Requirements.....	44
5.4 Software Quality Attributes.....	44
5.5 Business Rules.....	44
6. Other Requirements.....	45

Revision History

Name	Date	Reason For Changes	Version
All members	Oct 26, 2025	Prepared initial version	0.1
All members	Oct 29, 2025	Update section 3	0.2
All members	Oct 30, 2025	Update section 2	0.3
All members	Nov 2, 2025	Update section 4	0.4
All members	Nov 3, 2025	Update section 4	0.5
All members	Nov 4, 2025	Update section 5	0.6
All members	Nov 5, 2025	Update section 6	0.7
All members	Nov 6, 2025	Update section 1.5	0.8
All members	Nov 8, 2025	Update section 1.6	0.9
All members	Nov 10, 2025	Complete review - Final version	1.0

Table of Tables

Table 1- application features.....	10
Table 2 - Definitions, Acronyms, and Abbreviations.....	11
Table 3- sign in interface.....	19
Table 4 - Identity Verification interface.....	20
Table 5 - Forget My password interface	20
Table 6 - Car search and Discovery interface	21
Table 7 - Car details interface	21
Table 8 - Booking and payment interface	22
Table 9 - Car listing management interface.....	23
Table 10 - User management interface.....	24
Table 11 - Rate Experience interface.....	25
Table 12 – Register.....	30
Table 13 - Verify Identity.....	30
Table 14 - Update profile.....	31
Table 15 - Search Cars.....	31
Table 16 - View Car Details.....	31
Table 17 - Book car.....	32
Table 18 - Make Payment.....	32
Table 19 - Schedule booking.....	32
Table 20 - View Booking History.....	33
Table 21 - Cancel Booking.....	33
Table 22 - Send Feedback.....	33

Table 23 - Chat with Owner.....	34
Table 24 - Receive Notifications.....	34
Table 25 - Add Car Listing.....	35
Table 26 - Upload Car Photos.....	35
Table 27 - Set Price & Availability.....	36
Table 28 - Set Seasonal / Promo Rates.....	36
Table 29 - View Booking Requests.....	37
Table 30 - Approve / Reject Booking.....	37
Table 31 - Chat with Renter.....	38
Table 32 - Receive Payment.....	38
Table 33 - View Transaction History.....	39
Table 34 - Edit / Delete Listing.....	39
Table 35 - View Feedback.....	40
Table 36 - Monitor Activities.....	40
Table 37 - Manage Reviews & feedback.....	40
Table 38 - Manage Vehicle Categories.....	41
Table 39 - Edit Filter Options.....	41
Table 40 - Send Notifications & Announcements.....	41
Table 41 - Perform Security Audit.....	42
Table 42 - View Reports.....	42
Table 43 - Backup & Restore Data.....	42
Table 44 - Block Suspicious Account.....	43

Table of Figures

Figure 1 – Fadl Zahr system architecture.....	13
Figure 2 – Use cases.....	29

1. Introduction

This document is a Software Requirements Specification (SRS) for the “Fadl Zahr” system and will serve as the foundation for future improvements and updates. The “Fadl Zahr” system is designed to provide a safe and user-friendly platform for car Rental within the local community. Its main goal is to help reduce the number of unused private vehicles by allowing car owners to rent their cars with individuals who need temporary transportation. Through this application, users can easily list their cars, search for available vehicles, make reservations, and complete simulated payment transactions securely. The system mainly targets local residents who need a short-term transportation solution. It also encourages sustainability and efficient resource usage by promoting shared mobility among individuals. This section will describe some aspects related to the requirements of the “Fadl Zahr” system. The purpose and scope of this report will be defined, and all definitions, abbreviations, and acronyms used throughout the report will be listed for better understanding. In addition, all references used in this report will be mentioned.

1.1 Purpose

The objective of this document is to provide a complete and clear description of all software requirements for the “Fadl Zahr” application.

This SRS is intended to guide the software development lifecycle, ensuring that developers, testers, project supervisors, and future maintenance teams have a common understanding of the system requirements. It establishes a formal reference point for design, implementation, and verification activities, helping to reduce ambiguities and misinterpretations during development.

The document defines the boundaries of the system in the context of this academic project, clarifying what is included and what is excluded from the current prototype. Real commercial integrations, such as live payment gateways or external Absher connections, are outside the current scope. By clearly documenting functional and non-functional requirements.

This SRS serves as a basis for planning, risk assessment, validation, and traceability throughout the development process. Stakeholders can track changes, review updates, and verify that the final implementation meets the intended objectives. This SRS focuses on the core subsystems of the “Fadl zahr” system and provides a reference throughout the software lifecycle, supporting both academic evaluation and the establishment of professional software engineering practices.

1.2 Document Conventions

This document follows specific formatting and writing conventions to ensure consistency and readability across all sections.

- Font Style: *Times New Roman*
- Font Size: 14 pt for main titles, 13 pt for subsection titles, 12 pt for regular text
- Numbering Format: Hierarchical numbering (e.g., 1, 1.1, 1.1.1)
- Highlighting: Section titles appear in bold for clarity. Important technical terms may be emphasized when necessary.
- Priority: Higher-level requirement priorities are inherited by detailed sub-requirements unless stated otherwise.
- Standard Reference: The structure and terminology follow IEEE Std 830-1998 -IEEE Recommended Practice for Software Requirements Specifications.

1.3 Intended Audience and Reading Suggestions

This document is intended for the following stakeholders:

- **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/ Instructor:**

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

- **End 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.

- **Testing and Quality Assurance Team:**

Testers will use this document to develop test cases, verify that each requirement is met, and ensure the overall quality and reliability of the system.

- **Future Maintenance Team (if applicable):**

Developers or students who maintain or extend the system later can use this document as a reference to understand the system's functionality and design rationale.

Readers should begin with the **Introduction** and **Product Scope** sections to understand the system overview, followed by the detailed **Functional** and **Non-Functional Requirements** sections that describe specific system behaviors and constraints.

1.4 Product Scope

The “Fadl Zahr” system is a mobile car-rent application that connects verified car owners and renters. It allows owners to list their vehicles with full details, and enables users to search, book, and rent cars easily within their area. The system is designed to be secure, simple, and efficient, helping users make short-term car rental arrangements quickly. The application provides several useful features that enhance user experience and system efficiency.

Feature	Description
User Authentication	Secure login using a simulated Absher API for identity verification
Car Listing Management	Owners can add, edit, or remove car listings with images and details.
Search and booking	Renters can browse available cars and reserve them based on time and location
Payment Simulation	Mock payment process for academic testing purposes.
Rating and Feedback	Both owners and renters can rate and review each other after a transaction.
Admin Dashboard	Provides administrators with tools to manage users, monitor activities, and handle reports.

Table 1- application features

1.5 Definitions, Acronyms, and Abbreviations

Term / Acronym	Definition
SRS	Software Requirements Specification – A document that describes the software system's functions and constraints.
UI	User Interface – The visual part of the application that users interact with.
API	Application Programming Interface – A set of rules that allows different software systems to communicate with each other.
SAMA	Saudi Arabian Monetary Authority – The official body that regulates financial and electronic transactions in Saudi Arabia.
Absher	A Saudi government platform used for user identity verification and digital services.
OTP	One-Time Password – A temporary password used for secure login or verification processes.
DB	Database – A structured system used to store and manage application data.
GPS	Global Positioning System- Determining precise locations on Earth
IOS	iPhone Operating System- Running apps and managing the device.
HTTPS	Hypertext Transfer Protocol Secure-Encrypted website communication.
RESTful	Representational State Transfer- A design style for web APIs
SSL	Secure Sockets Layer- A protocol to encrypt data between server and browser.
TLS	Transport Layer Security- An improved version of SSL for secure data transfer.
LAN	Local Area Network- A network covering a small area like an office or building
MAN	Metropolitan Area Network- A network covering a city or large area
TCP/IP	Transmission Control Protocol / Internet Protocol- Basic protocols for internet data transfer.

Table 2 - Definitions, Acronyms, and Abbreviations

1.6 References

The following documents and web sources are referenced in this Software Requirements Specification (SRS). Each provides supporting information, standards, or background material relevant to the development and specification of the system.

1. **IEEE Computer Society.** (1998). *IEEE Std 830-1998 — IEEE Recommended Practice for Software Requirements Specifications*. Institute of Electrical and Electronics Engineers. Retrieved from <https://ieeexplore.ieee.org/document/720574>
2. **ScienceDirect.** (n.d.). *Software Interface*. Retrieved November 4, 2025, from <https://www.sciencedirect.com/topics/computer-science/software-interface>
3. **Perforce Software.** (n.d.). *How to Write a Software Requirements Specification (SRS) Document*. Retrieved November 4, 2025, from <https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document>
4. **Lumi Rental.** (2025). *Lumi – Home*. Retrieved November 7, 2025, from <https://lumirental.com/ar/>
5. **Academia.edu.** (n.d.). *Functional Approach to Software Requirements Specification*. Retrieved from <https://www.academia.edu/download/77651836/FuncApproach.pdf>
6. **IEEE Computer Society.** (2014). *IEEE Std 1016-2009 — IEEE Standard for Information Technology – Systems Design Description (SDD)*. Institute of Electrical and Electronics Engineers. Retrieved from <https://ieeexplore.ieee.org/document/5290795>
7. **ISO/IEC/IEEE 29148:2018.** (2018). *Systems and Software Engineering — Life Cycle Processes — Requirements Engineering*. International Organization for Standardization. Retrieved from <https://www.iso.org/standard/72026.html>

2. Overall description

This section will provide an overall description of the Fadl Zahr application and elaborates on two subsections which are product perspective and functions. In the product perspective subsection, a background of the functions and the constraints imposed on their operation will be provided and how the software interacts with other systems and products shall be identified. Product functions subsection gives a summary of major functions for each end-user to be performed by the software.

2.1 Product perspective

Fadl Zahr is an interactive transaction-based mobile application developed to create a secure and trustworthy peer-to-peer car-sharing platform that connects vehicle owners with potential renters. The system promotes efficient resource utilization by enabling car owners to generate income from their idle vehicles while providing renters with flexible and affordable transportation options. The end-users of the system are Car Owners, Renters, and Administrators. The system displays various interfaces which are designated for each type of end user. The Figure 1 below demonstrates the system architecture of Fadl Zahr which connects the primary user groups with the core application and external services.

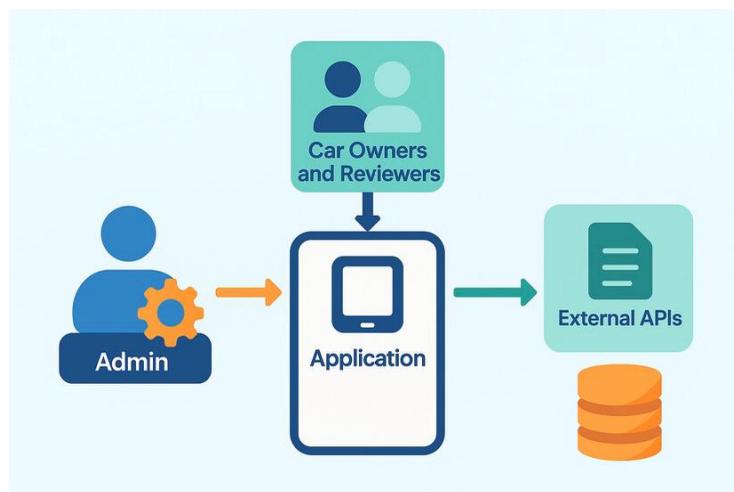


Figure 1 – Fadl Zahr system architecture

The Fadl Zahr application is directly connected to a cloud database which is efficiently used to store the data of the entire system, including user profiles, vehicle listings, booking records, and transaction history. Moreover, the system is connected to external APIs including the Absher platform for identity verification, payment gateways for processing financial transactions, and notification services for sending real-time alerts to users.

There are constraints imposed on the system's operations and database based on the function of each user. The Administrator has the privilege to completely access the database, therefore allowed to monitor user activities, manage vehicle categories, moderate reviews, and block suspicious accounts. The Car Owners can create and manage their vehicle listings, set pricing and availability, approve or reject booking requests, and track their earnings. The Renters can search for available vehicles, make booking requests, process secure payments, and manage their rental history.

2.2 Product functions

The "Fadl Zahr" app is a smartphone application that connects car owners with people who want to rent cars in a secure and easy-to-use way.

Here are the most important functions offered by the system:

1. User Registration and Identity Verification

- The system allows users, whether owners or renters, to register and log in securely.
- User identity is verified through the Absher system.
- Users can recover their password if they forget it.
- Allows users to update their personal information and profile picture after registration.
- The system detects and prevents duplicate accounts using the same ID.

2. Managing Listed Cars

- Owners can add their cars for rent, edit information, or delete them.
- They can enter details such as car type, license plate number, price, and availability
- Renters can search for cars using filters such as price, location, or type.
- The system allows owners to upload multiple car photos for better visibility.
- The system allows owners to set seasonal or promotional rates.

3. Booking and Renting

- Renters can easily book the car they want.
- A notification is sent to the owner with the booking request, and they approve it.
- The system keeps a record of bookings and does not allow duplicate bookings for the same car.
- Users can schedule future bookings and specify preferred time periods.
- Owners can communicate with renters within the app for further coordination.

4. Payment and Transactions

- The system offers a secure in-app payment feature.
- A receipt is displayed to the user after each payment transaction
- A refund is available in case of booking cancellation.
- Supports multiple payment methods such as digital wallets, credit and debit cards, and bank transfers.
- Transaction history is available for both renters and owners.
- Automatic invoice generation and tax calculation for each booking.

5. Feedback and Reviews

- The renter and the car owner are both fully informed about the scope of the service after its completion.
- The feedback feature helps foster trust and credibility among users.
- Administrators can review comments and ratings to ensure their authenticity and monitor service quality.
- Allows users, whether renter or owner, to respond to comments to clarify or acknowledge feedback.
- The system highlights top-rated cars to encourage service quality.

6. Administrator Control Panel

- Grants administrators permissions to monitor users and activities within the system.
- Allows administrators to review details, block suspicious accounts, and manage reviews.
- Allows management of application content, including vehicle categories and filtering options.

7. Alerts and Notifications

- The system sends notifications to confirm bookings, cancellations, or payment updates.
- Sends reminders about upcoming car bookings and the expiration of car listings.
- Administrators can send general or specific notifications to system users as needed.
- Push notifications for promotions, updates, and system maintenance.
- Alerts include reminders for required documents or license expiration

8. Data Protection and System Security

- The system uses data encryption technologies to protect user information.
- Access to the system is restricted to authorized users only after login and identity verification.
- The system follows strict privacy policies to protect user data and transaction records from any unauthorized use.
- Regular security reviews and vulnerability assessments are conducted.
- Role-based access control ensures that users only perform authorized actions.
- Data is backed up periodically, and logs of all user activity are maintained for security and data protection purposes.

2.3 User Classes and Characteristics

1- Car Owners

Characteristics:

- Primary users of the application are the ones who list their cars for rent.
- Technical expertise in the field may differ since some people using smartphones are more familiar with apps.
- Experienced with the processes involved in the ownership and rental of cars.
- Interested in maximizing the exposure of their listings for better-earning potential.

Key Functions Used:

- Managing listed cars: adding, editing, deleting cars.
- Handling bookings and payments.
- Respond to renter feedback and ratings.

2- Rental

Characteristics:

- Renters who need to rent cars for certain periods.
- usually non-technical, more concerned with functionality and ease of use.
- Looking for a variety of cars at different price points.

Important Functions Used:

- Searching and filtering available cars.
- Booking cars and making payments.
- Leaving feedback and reviewing their rental experiences.

3- Administrators

Characteristics:

- App staff or managers responsible for overseeing the system.
- Highly technical with deep system control access.
- Focused on protecting the integrity of the app, maintaining smooth operations, and preventing fraud.

Key Functions Used:

- Monitoring and managing of user accounts and activities.
- Overseeing car listings and feedback.
- Sending notifications and updates to users.

2.4 Operating Environment

Hardware Platform:

- Smartphones: iOS and Android devices.
- Devices must have at least the minimum technical requirements for running the app with ease, including memory and processing speed.

Operating System and Versions:

- iOS: Version 11.0 or higher.
- Android: Version 7.0 (Nougat) or higher.

Additional Software Components:

- Backend system: The cloud-based services, such as AWS or Google Cloud, are used for maintaining the users' data, payment, and transaction information.
- Secure payment gateway integration
- Absher system integration for identity verification.

2.5 Design and Implementation Constraints

- Devices with low processing capability may face app performance issues, especially with image-intensive tasks such as photo uploading of cars.

Regulatory Policies:

- Compliance with local data privacy and security regulations such as, but not limited to, GDPR, Saudi Arabian data protection laws.
- Payment processing should be PCI-DSS compliant for credit card transactions.

Technical Constraints:

- Support by a wide array of devices on numerous operating systems.
 - Optimize the app for performance by achieving fast load times, especially when filtering through cars or processing payments.
- Integration:
- Integration with external systems for identity verification, such as Absher.
 - Secure integration with the payment providers for the processing of each transaction.

Security Considerations:

- Encryption for user data is required in the application, such as SSL/TLS for data in transit.
- Role-based access control for administrators to prevent unauthorized access.
- This means that no two successive elements depend on each other.

2.6 User Documentation

1. User Manuals:

- Available digitally as PDF or app-based guide for iOS and Android users.
- Includes step-by-step procedures for registration, listing of cars, booking, and payment management.

2. On-Line Help:

- Contextual help in-app because sometimes users need step-by-step assistance for common issues.
- FAQs and troubleshooting tips are available within the app or on the official website.

3. Tutorials:

- Short in-app video tutorials, also available on YouTube, which demonstrate key functionalities. For example, how to list a car, how to book a car, etc.

4. Help Desk:

- Contact details for any support services through the application: this will include live chat and email.

2.7 Assumptions and Dependencies

- The users will have stable internet connections to ensure smooth functionality.
- The car owners are responsible for giving authentic information about the car.
- Renters follow the app's guidelines in booking and follow through on rentals.
- Users will have access to the required hardware (smartphones) with compatible OS versions.

Dependencies:

- The use of at least one third-party service for payment processing, such as Apple Pay.
- The Absher system for identity verification, which may define the way of user registration and verification.
- External services for the sending of SMS notifications or push notifications for bookings and alerts.
- Cloud service providers: host and data storage, maintenance, and preparatory measures for downtime.

3. External interface requirements

This section describes the requirements for all external interfaces of the car rental system, including user interfaces and communication interfaces with external systems.

3.1 User interfaces

The system provides distinct interface experiences for three types of users: Renters, Car Owners, and Administrators. All interfaces are designed to be responsive and accessible across web and mobile platforms.

3.1.1 Common Interfaces

Interfaces used by all user types for fundamental system interactions:

3.1.1.1 Sign in Interface

All users (Renter, Owner, Admin) will sign into the system using their credentials

Field	Format	Level	Input/output	Comment
User ID/Email	Text	Required	Input	Email formats for all users
Password	Encrypted Text	Required	Input	Minimum 8 characters with uppercase, lowercase, numbers, and special characters
User type	Dropdown	Required	Input	Select: Renter, Car Owner, or Admin

Table 3 – sign in interface

3.1.1.2 Identity Verification Interface

For Renters and Car Owners to verify identity through Absher integration

Field	Format	Level	Input/output	Comment
National ID	Number	Required	Input	10-digit Saudi National ID
Absher verification	API call	Required	Input/output	Redirect to Absher for biometric verification

Table 4 -Identity Verification interface

3.1.1.3 Forget My Password Interface

All users can reset their password through email verification

Field	Format	Level	Input/output	Comment
Email	Text	Required	Input	Must match registered email
OTP	Number	Required	Input	6-digit code sent to email
New password	Encrypted Text	Required	Input	Must meet password requirements

Table 5 – Forget My password interface

3.1.2 Renter Interfaces

3.1.2.1 Renter Homepage Interface

This interface serves as the primary landing page for the renter after a successful login. It is designed for quick access and discovery, providing a personalized overview of relevant options and information. Key elements and features include:

- Personalized Welcome & Quick Search Bar: A prominent search bar is pre-positioned for initiating new car searches, potentially with pre-filled popular locations or dates.
- Recommended Vehicles: A dynamically generated list of car suggestions based on the user's previous searches, browsing history, and booking preferences.
- Recent or Featured Listings: A section showcasing high-demand vehicles, special offers, or newly added cars in the user's preferred locations.

- Quick-Action Shortcuts: Clearly defined buttons or tiles for primary actions, such as:
 - "Search Cars"
 - "Manage My Bookings"
 - "View Booking History"
- Active Booking Alerts: Notifications or a summary card for any ongoing or upcoming bookings, providing essential details (e.g., car model, pickup date) and quick links to manage them.
- System-Wide Notifications: Alerts from the platform regarding new features, promotional discounts, or important updates about their account or past bookings.

3.1.2.2 Car Search & Discovery Interface

Advanced search with multiple filter options

Field	Format	Level	Input/output	Comment
Location	Text/GPS	Required	Input	City or use current location
Date Range	Date	Required	Input	Start and end dates for rental
Car Type	Dropdown	Optional	Input	Sedan, SUV, Luxury, etc.
Price Range	Slider	Optional	Input	Minimum to maximum price per day

Table 6 – Car search and Discovery interface

3.1.2.3 Car Details Interface

Comprehensive view of selected vehicles

Field	Format	Level	Input/output	Comment
Car Photos	Gallery	Required	Output	Multiple high-quality images
Owner info	Text	Required	Output	Name, rating, response rate
Pricing	Number	Required	Output	Daily rate, security deposit
Availability	Calendar	Required	Output	Available/Booked dates

Table 7 – Car details interface

3.1.2.4 Booking & Payment Interface

Multi-step booking and payment process

Field	Format	Level	Input/output	Comment
Rental Period	Date	Required	Output	Selected dates from search
Total cost	Number	Required	Output	Calculated based on duration
Payment Method	Dropdown	Required	Input	Credit Card, Apple Pay, Mada
Insurance Option	Checkbox	Optional	Input	Additional insurance coverage

Table 8 – Booking and payment interface

3.1.2.5 Booking History Interface

This interface provides the renter with a complete record of all past and current bookings. Key features include:

- A chronologically ordered list of bookings (Current, Upcoming, Completed).
- The status of each booking (Confirmed, Completed, Cancelled).
- Detailed information for each booking: Vehicle, Owner, Rental Period, Total Cost.
- Option to re-book a vehicle from history.
- Ability to submit a review and rating after a booking is completed.

3.1.3 Car Owner Interfaces

3.1.3.1 Owner Dashboard Interface

This is the owner's main control panel, providing an immediate overview of their business performance. It includes:

- A visual summary of earnings (Weekly, Monthly, Yearly).
- Count of active and upcoming bookings.
- Recent ratings and reviews from renters.
- Status of all listed vehicles (Active, Booked, Under Maintenance).
- Instant notifications for new booking requests requiring action.

3.1.3.2 Car Listing Management Interface

Add, edit, and manage vehicle listings.

Field	Format	Level	Input/output	Comment
Vehicle Details	Text	Required	Input	Make, model, year, features
Photos	Multiple Files	Required	Input	Up to 10 images per vehicle
Pricing	Number	Required	Input	Daily rate, seasonal pricing
Availability	Calendar	Required	Input	Block unavailable dates

Table 9 – Car listing management interface

3.1.3.3 Booking Requests Interface

This interface allows the owner to manage incoming booking requests for their vehicles. Features include:

- A list of all new booking requests with full details (Renter, Rental Dates, Total Cost).
- Clear options to accept or decline each booking request.
- Ability to communicate with the renter via an internal messaging system before confirmation.
- Display of response history and tracking of booking status after a decision is made.

3.1.3.4 Earnings & Transaction History Interface

A comprehensive financial interface enabling the owner to track their income. Key features are:

- A graph illustrating earnings over different time periods.
- A detailed breakdown of all transactions (Payments, System Commissions, Payouts).
- Ability to filter the history by date, vehicle, or transaction type.
- Options to download account statements or have them emailed.

3.1.4 Admin Interfaces

3.1.4.1 Admin Dashboard Interface

This is the central control panel for the administrator, designed to monitor system health and overall activity. It includes:

- Live statistics: number of active users, total bookings, total revenue.
- Charts and graphs to track growth and user activity.
- A list of critical alerts and recently reported issues.
- Quick-access shortcuts to key admin sections (User Management, Moderation, Reports).

3.1.4.2 User Management Interface

Manage all user accounts and permissions.

Field	Format	Level	Input/output	Comment
User Search	Text	Required	Input	Search by name, email, or ID
Account Status	Dropdown	Required	Input/Output	Active, Suspended, Banned
Verification Status	Text	Required	Output	Verified, Pending, Rejected

Table 10 – User management interface

3.1.4.3 Content Moderation Interface

This interface allows the administrator to monitor and manage system content to maintain quality and safety. Features include:

- A queue of reported content (reviews, car images, profile pictures) with reporting reasons.
- Tools to review and take action (Delete Content, Ignore Report, Hide).
- Ability to temporarily or permanently suspend abusive users.
- A section to review and approve new vehicles added by owners before they are publicly listed.

3.1.4.4 Reports & Analytics Interface

This interface provides the administrator with deep insights into system performance. It offers:

- Customizable reports on users, bookings, revenue, and owner performance.
- Advanced analytics such as occupancy rates, most demanded locations, and customer usage patterns.
- Comparisons between different time periods to assess growth and performance.
- Capability to export reports in multiple formats (PDF, Excel, CSV) for sharing or archiving.

3.1.5 Rating & Review Interfaces

3.1.5.1 Rate Experience Interface

Post-rental rating system for both renters and owners.

Field	Format	Level	Input/output	Comment
Star Rating	Number	Required	Input	1-5 stars with half increments
Written Review	Text	Optional	Input	Maximum 500 characters
Cleanliness	Number	Optional	Input	1-5 rating for vehicle condition
Communication	Number	Optional	Input	1-5 rating for responsiveness

Table 11 – Rate Experience interface

3.2 Hardware interfaces

The Fadl Zahr mobile application operates on both Android and iOS devices, developed using the Flutter framework.

Supported Devices:

- Android smartphones (Android 10 or higher)
- iOS smartphones (iOS 13 or higher)
- Laptops or PCs for the admin dashboard (via web browser)

Hardware Interactions:

- Camera: Used by the owner to capture photos of the vehicle before adding a new listing and after the rental period to verify the car's condition.
- Microphone: Enables in-app voice notes or calls between the renter and the owner to simplify communication.
- Internet Connection: Essential for browsing cars, booking, processing payments, and synchronizing real-time data.

3.3 Software interfaces

The Fadl Zahr system interacts with external APIs and cloud services to provide a secure and smooth experience for all users.

1) **Absher API (Identity Verification):**

Absher is a Saudi government platform used for official digital identity verification. In Fadl Zahr, the Absher API is integrated to verify the identity of users (owners and renters) before they can use sensitive features such as listing a car or making a payment. When a user registers, the app redirects them to Absher's secure portal where they log in with their national credentials. Once their identity is confirmed, Absher returns a unique verification token to the Fadl Zahr system. This ensures that only legitimate users participate in transactions, reducing fraud and building trust between owners and renters.

2) **Payment Gateway API:**

Handles online transactions securely between renters and owners using trusted services such as STC Pay or Mada. The gateway encrypts all financial data and confirms each transaction with banks before sending confirmation to the system.

3) **Insurance API:**

Allows the application to verify whether a car's insurance policy is valid and active before allowing it to be listed for rent.

4) **Firebase SDK and Backend Service:**

Firebase provides authentication, real-time database synchronization, and cloud storage. It handles user accounts, data backup, and push notifications.

5) **Database:**

All data (user profiles, cars, bookings, transactions, feedback) is stored in Firebase Firestore and updated instantly across devices.

6) **Framework & Version Control:**

Flutter Framework builds a cross-platform app for Android and iOS. GitHub is used for code version control and collaboration.

7) **Notification Service:**

Firebase Cloud Messaging (FCM) sends alerts about bookings, payments, and system updates instantly to users.

3.4 Communications interfaces

The communication interfaces of the Fadl Zahr system define how the mobile application, backend services, and third-party APIs exchange information securely and efficiently. All communications occur through encrypted internet protocols to ensure data privacy and reliability.

1) Internal Communication (App ↔ Server)

- The mobile application communicates with the backend server through HTTPS RESTful APIs.
- Every request (e.g., login, registration, booking, or payment) is sent from the client app to the server in JSON format.
- The server validates, processes, and returns a structured response to the app.
- The system uses token-based authentication (JWT) to ensure that each user session is secure and that only authorized requests are processed.

2) Communication with Absher API (for ID Verification)

- When a new user registers, the system sends an HTTPS request to the Absher API to initiate the identity verification process.
- The user is redirected to Absher's secure page to log in using their National ID and password.
- After successful verification, Absher returns a verification token (encrypted) that confirms the user's real identity.
- The backend stores this token temporarily to complete registration and then removes it for privacy protection.
- All communication with Absher uses TLS 1.3 encryption and follows the Saudi e-Government security standards.

3) Communication with Payment Gateway (for Transactions)

- The application integrates with a certified Payment Gateway API (e.g., STC Pay or Mada).
- When a renter makes a payment, the system sends an encrypted transaction request containing booking details and the amount due.
- The gateway communicates with the bank to authorize the transaction, then returns a confirmation code or error message.
- After successful payment, the system updates the renter's and owner's transaction records in the database.
- Sensitive payment data (like card numbers) never passes through the app - only secure tokens are exchanged.

4) Communication with Firebase Services

- Firebase Authentication: handles secure login and registration using encrypted tokens.
- Firebase Firestore Database ensures real-time synchronization of data (bookings, listings, feedback).
- Firebase Cloud Messaging (FCM): sends push notifications to users about booking status, payments, or messages.
- All Firebase communications occur via HTTPS with Google's global infrastructure and comply with data protection regulations.

5) Communication with Insurance API

- The system periodically checks with an external Insurance API to verify the validity of the car's insurance policy before allowing it to appear in the listings.
- If the insurance is expired or invalid, the system alerts the owner to update the policy before proceeding.

6) Communication with Admin Dashboard (Web Application)

- The admin dashboard is a web-based interface connected to the same backend as the mobile app.
- Admins use it to monitor bookings, payments, reports, and user activities.
- Communication between the admin web dashboard and the backend is also done via HTTPS RESTful APIs.

4. System features

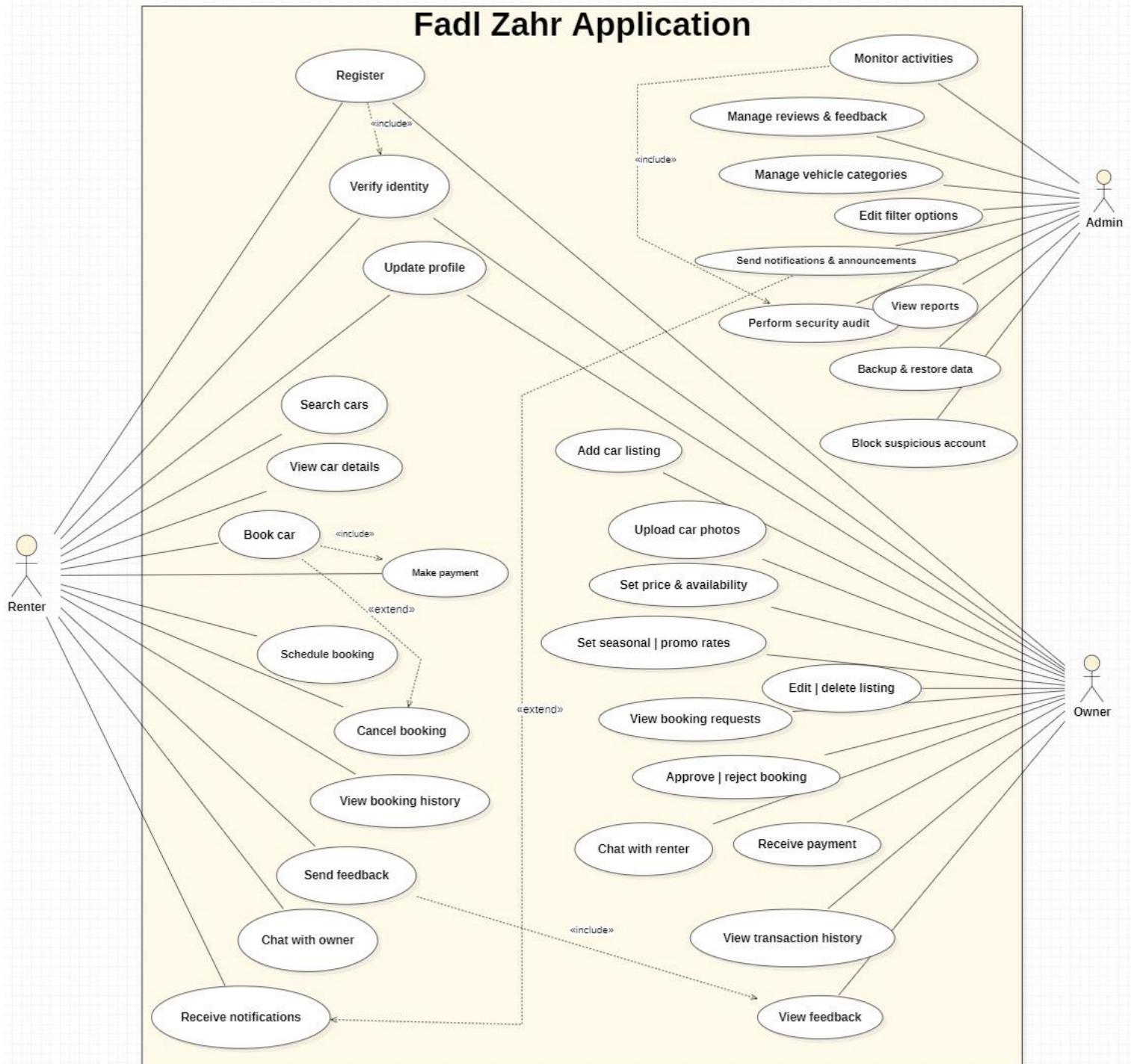


Figure 2 – Use cases

4.1 System Features for Renter

- Use Case: [Register](#)

Actors	Renter, Authentication System
Description	The renter creates a new account by entering personal details such as name, email, phone number, and password. The system validates the information, checks for duplicates, and creates a new record in the database. After successful registration, a verification link or code is sent to confirm the account before full access is granted.
Data	Name, email, phone number, password, verification code.
Stimulus	User selects “Register” and submits the registration form.
Response	A new account is created successfully, and a confirmation email or message is sent to verify the identity.
Comments	Email and phone number must be unique; password must meet security requirements.

Table 12 – Register

- Use Case: [Verify Identity](#)

Actors	Renter, Absher verification API
Description	After registration, the renter must verify their identity through Absher to ensure security and trust. The system connects with the Absher verification API and prompts the renter to provide their National ID and verification details. Once verified, the user's account status changes to “Verified”, allowing access to sensitive actions such as booking cars.
Data	National ID, date of birth, verification code from Absher.
Stimulus	User clicks “Verify Identity” and completes the Absher verification process.
Response	The system confirms verification and updates the user status to “Verified”.
Comments	Verification must be completed before the renter can book a car.

Table 13 - Verify Identity

- Use Case: [Update profile](#)

Actors	Renter
Description	The renter can update their account information at any time, including name, profile photo, phone number, and language preferences. The system validates all changes and updates the database instantly. If sensitive data such as email or phone number is changed, the system may require re-verification for security purposes.
Data	Display name, profile picture, phone number, notification preferences.
Stimulus	User navigates to “Profile” and selects “Edit”.
Response	Profile is updated successfully with a confirmation message.
Comments	Sensitive data may require re-verification.

Table 14 - Update profile

- Use Case: [Search Cars](#)

Actors	Renter, System database
Description	The renter searches for available cars using filters such as car type, location, price range, and availability dates. The system processes the query, retrieves matching results, and displays them in a list for the renter to choose from. Results can be sorted or refined according to user preferences.
Data	Location, car type, price range, date and time availability.
Stimulus	User enters search filters and clicks “Search”.
Response	System displays a list of matching cars.
Comments	Search results depend on availability and filter accuracy.

Table 15 - Search Cars

- Use Case: [View Car Details](#)

Actors	Renter
Description	When the renter selects a car from the search results, the system displays detailed information about the car such as specifications, photos, price, availability, and owner’s reviews. The renter can then decide to proceed with booking or contact the owner directly for more information.
Data	Car images, specifications, owner rating, price, and terms.
Stimulus	User clicks on a specific car from the search results.
Response	Car details are displayed with options to book or chat with the owner.
Comments	All car details are retrieved from the owner’s listing.

Table 16 - View Car Details

- Use Case: Book car

Actors	Renter, Owner, Payment Gateway
Description	The renter selects the desired car, sets the start and end dates, and confirms the booking. The system checks the car's availability, calculates the total cost, and reserves the car temporarily. A confirmation notification is sent to both renter and owner, and the renter is prompted to make payment to complete the booking.
Data	Booking dates, car ID, price, renter info, payment details.
Stimulus	User clicks "Book Car" and confirms booking details.
Response	Booking requests are created and sent to the owner for approval.
Comments	The renter must be verified before booking. Payment step is included.

Table 17 - Book car

- Use Case: Make Payment

Actors	Renter, Payment Gateway
Description	After booking, the renter proceeds to make a secure payment using the integrated gateway. The system connects the payment provider to process the transaction. If payment is successful, the booking status changes to "Confirmed", and both renter and owner receive notifications. In case of failure, the renter can retry using another method.
Data	Amount, card or Mada details, transaction ID, payment status.
Stimulus	User confirms the booking and selects a payment method.
Response	Payment is processed, and the booking status becomes "Confirmed".
Comments	Failed payments show an error message and allow retry.

Table 18 - Make Payment

- Use Case: Schedule booking

Actors	Renter
Description	The renter can plan future bookings or set up recurring ones, such as weekly rentals. The system verifies the car's availability for the chosen dates and saves the schedule. Before the booking date, reminders are automatically sent to both renter and owner.
Data	Preferred start/end dates, time, frequency, reminder options.
Stimulus	User selects "Schedule Booking" and sets future dates.
Response	System stores the schedule and sends a reminder before the booking date.
Comments	Extended use case of "Book Car"; verifies date availability.

Table 19 - Schedule booking

- Use Case: [View Booking History](#)

Actors	Renter
Description	The renter can view all past and current bookings, including details such as car name, duration, payment status, and total cost. The system retrieves this data from the database and displays it in an organized list with filtering options. The renter can also download receipts or submit feedback for completed trips.
Data	Booking records, transaction amounts, booking status.
Stimulus	User opens “Booking History” from the menu.
Response	System shows a list of completed and active bookings.
Comments	Users can view invoices or booking details from this list.

Table 20 - View Booking History

- Use Case: [Cancel Booking](#)

Actors	Renter, Owner
Description	The renter may cancel an existing booking before the start date according to the cancellation policy. The system checks eligibility for refund, updates the booking status, and notifies the owner. If applicable, the payment gateway initiates a refund transaction.
Data	Booking ID, reason, refund amount, cancellation date.
Stimulus	User clicks “Cancel Booking”.
Response	Booking status changes to “Cancelled” and both parties are notified.
Comments	Refund amount depends on the cancellation time.

Table 21 - Cancel Booking

- Use Case: [Send Feedback](#)

Actors	Renter, Review System, Owner
Description	After completing a trip, the renter submits a rating and reviews their experience. The renter can rate the owner and car from one to five stars and leave a written comment. The system records feedback and updates the owner’s average rating, which is visible to future users.
Data	Rating stars, written review, optional photo.
Stimulus	User clicks “Send Feedback” after a completed booking.
Response	System saves and displays the review under the owner’s profile.
Comments	Feedback can only be submitted after trip completion.

Table 22 - Send Feedback

- Use Case: Chat with Owner

Actors	Renter, Owner, Chat Service
Description	The renter communicates directly with the car owner through an in-app chat feature to discuss pickup time, location, or special requests. The system ensures real-time messaging, secure storage of chats, and instant notifications for new messages.
Data	Text messages, timestamps, optional attachments.
Stimulus	User clicks “Chat with Owner” and sends a message.
Response	Messages are delivered instantly with read receipts.
Comments	Offensive language is filtered, and users can report misuse.

Table 23 - Chat with Owner

- Use Case: Receive Notifications

Actors	Renter, Notification Service
Description	The renter receives automatic alerts about important events such as booking confirmations, payments, or chat messages. Notifications appear as push alerts or in-app messages. The renter can customize which notifications to receive and the preferred method.
Data	Notification type, title, content, timestamp, link.
Stimulus	System triggers notifications based on events (e.g., booking approved).
Response	User receives push or in-app notifications.
Comments	Users can customize notification preferences or disable them.

Table 24 - Receive Notifications

4.2 System Features for Owner

- Use Case: [Add Car Listing](#)

Actors	Owner, System
Description	The owner adds a new car listing by entering all relevant vehicle specifications and details. These include the car's make, model, year of manufacture, color, condition (new or used), mileage, fuel type, transmission type, number of seats, safety features, insurance status, and any existing scratches or damages. The owner may also include maintenance history and additional notes about the vehicle's condition. Once all required data is entered, the system saves the information and adds the new car listing to the available cars for renters to view and book.
Data	Car specifications (make, model, year, color, condition, mileage, fuel type, transmission type, number of seats, safety features, insurance status, damages, maintenance history, notes)
Stimulus	The owner selects “Add Car Listing” and fills out the car information form, then submits it to the system.
Response	The system validates the details provided, saves the new listing in the database, and confirms that the car has been successfully added to the available listings.
Comments	The owner must have a verified account to create a listing. The system should ensure all mandatory fields are completed and prevent duplicate listings.

Table 25 - Add Car Listing

- Use Case: [Upload Car Photos](#)

Actors	Owner, System
Description	The owner uploads one or more pictures of the car to make the listing more accurate and visually informative for renters. These images help clarify the car's condition, color, and overall appearance, allowing renters to better evaluate the vehicle before booking.
Data	Image files (JPEG, PNG), car listing ID
Stimulus	The owner selects “Upload Car Photos” for a specific car listing and chooses images from their device to upload.
Response	The system validates the image format and size, uploads the photos, and links them to the corresponding car listing. A confirmation message is then displayed to the owner.
Comments	The system must accept only supported image formats and enforce size limits. At least one clear photo of the car's exterior is required for each listing.

Table 26 - Upload Car Photos

- Use Case: Set Price & Availability

Actors	Owner, System
Description	The owner sets the rental price options for each car listing and defines its availability period. The owner specifies pricing details for different durations such as daily, weekly, monthly, or annual rates and may adjust prices based on demand or season. The system stores this information and displays the updated pricing and availability to renters when browsing cars.
Data	Rental price options (daily, weekly, monthly, annual), availability dates, car listing ID
Stimulus	The owner selects “Set Price & Availability” from the car listing options, enters the price values and availability dates, then saves the settings.
Response	The system validates the pricing inputs, updates the listing with the new prices and availability, and confirms successful update. The changes are immediately visible to renters.
Comments	The system should prevent overlapping availability periods and ensure that all price fields contain valid numeric values.

Table 27 - Set Price & Availability

- Use Case: Set Seasonal / Promo Rates

Actors	Owner, System
Description	The owner creates promotional or seasonal discount rates for specific periods. Discounts may be applied during holidays, special occasions, or peak seasons to attract more renters. The owner may also configure automatic price reductions for longer rental durations (e.g., weekly or monthly rentals). The system saves the updated promotional rates and displays them within the car listing for renters to view.
Data	Rental price options (daily, weekly, monthly, annual), availability dates, car listing ID
Stimulus	The owner selects “Set Seasonal / Promo Rates,” specifies the discount percentage and duration, and submits the promotional offer.
Response	The system validates the entered data, applies the discount to the selected car listing, and confirms successful update. Renters can immediately see the discounted rates on the listing.
Comments	The system should prevent overlapping promotions and automatically deactivate expired offers.

Table 28 - Set Seasonal / Promo Rates

- Use Case: View Booking Requests

Actors	Owner, System
Description	The owner views all booking requests submitted by renters for their cars. Each request includes rental details, requested rental dates, selected duration, and any special requests provided by the renter (e.g., delivery time, child seat, or specific conditions). The owner can review this information before approving or rejecting the request.
Data	Booking ID, renter details, car ID, rental dates, booking status, special requests
Stimulus	The owner selects “View Booking Requests” from the menu to review incoming requests for their car listings.
Response	The system retrieves and displays all booking requests with complete details and current statuses (pending, approved, or rejected).
Comments	The system should organize requests by date or car listing and clearly show any additional notes or special rental requests.

Table 29 - View Booking Requests

- Use Case: Approve / Reject Booking

Actors	Owner, System
Description	The owner reviews booking requests from renters, checking the requested dates, payment status, and any special requests included by the renter. Based on car availability and renter reliability, the owner either approves or rejects the booking. If approved, the system reserves the car and notifies the renter. If rejected, the system sends a message explaining the reason.
Data	Booking ID, renter details, car ID, booking dates, payment information, special requests, booking status
Stimulus	The owner opens the “Booking Requests” section, selects a pending request, and clicks either “Approve” or “Reject.”
Response	The system updates the booking status, notifies the renter via the app and email, and adjusts car availability accordingly.
Comments	Approved bookings are locked for the selected period. The system should prevent double-booking or overlapping reservations.

Table 30 - Approve / Reject Booking

- Use Case: Chat with Renter

Actors	Owner, System, Rent
Description	The owner communicates directly with the renter through an in-app chat feature to discuss booking details, pickup arrangements, or car conditions. Both parties can exchange messages before and during the rental period.
Data	Chat messages, timestamps, sender/receiver IDs, booking ID
Stimulus	The owner opens the “Chat” section and selects a conversation with a renter to send or view messages.
Response	The system sends and displays messages in real-time, notifies the receiver, and stores the chat history securely.
Comments	The chat feature supports only text and system-approved media. Conversations are saved for reference in case of disputes.

Table 31 - Chat with Renter

- Use Case: Receive Payment

Actors	Owner, System, Payment Gateway
Description	After a renter completes the booking and returns the car in good condition, the system automatically transfers the rental payment to the owner's account. The total amount is calculated after deducting service fees or penalties if applicable. The owner can then view the payment confirmation and related transaction details in their account dashboard.
Data	Booking ID, Payment ID, Owner ID, Amount, Service Fees, Transfer Date, Payment Status
Stimulus	Once the renter confirms the car return and the system verifies the booking completion, the payment process is initiated automatically. The system checks the payment record, deducts platform fees, and triggers the transfer to the owner's linked account or digital wallet.
Response	The system confirms successful payment transfer, updates the owner's balance, and sends a detailed transaction notification including the date, amount, and reference number.
Comments	Only the owner has access to view or track received payments. The system must ensure secure transfers and maintain detailed transaction logs for auditing and reporting purposes.

Table 32 - Receive Payment

- Use Case: [View Transaction History](#)

Actors	Owner, System
Description	The owner views a complete history of financial transactions related to their car listings. The system displays all received payments, pending transfers, and service fees for each booking. Owners can filter results by date, car, or renter.
Data	Transaction ID, car ID, renter ID, booking ID, date, amount, payment status
Stimulus	The owner selects “Transaction History” from their profile menu.
Response	The system retrieves and displays a list of transactions with details and filter options.
Comments	The system should allow exporting reports (PDF or CSV) and flag any failed or delayed transactions.

Table 33 - View Transaction History

- Use Case: [Edit / Delete Listing](#)

Actors	Owner, System
Description	The owner edits or removes existing car listings. The owner can update details such as price, availability, or car specifications if changes occur (e.g., maintenance or insurance renewal). When a listing is deleted, the system ensures no active bookings are linked before removing it.
Data	Car ID, car details (specifications, price, availability), status
Stimulus	The owner selects a listing and chooses either “Edit” or “Delete” from the menu.
Response	The system validates the updates, saves changes, or removes the listing and displays a confirmation message.
Comments	The system must prevent deleting listings with active bookings and update visibility immediately after any modification.

Table 34 - Edit / Delete Listing

- Use Case: [View Feedback](#)

Actors	Owner, System
Description	The owner views feedback and ratings submitted by renters after completing their bookings. Each review includes a rating score and a comment describing the renter's experience. The owner can analyze this feedback to improve service quality or vehicle condition.
Data	Feedback ID, renter ID, car ID, rating (1–5), comment, date
Stimulus	The owner selects "Feedback" from the menu to view reviews for their cars.
Response	The system retrieves and displays feedback sorted by date or rating.
Comments	The system should prevent owners from editing renter feedback but allow responses or reports of inappropriate comments.

Table 35 - View Feedback

4.3 System Features for Admin

- Use Case: [Monitor Activities](#)

Actors	Admin, System Logs
Description	The admin comprehensively monitors all real-time and historical user activities and system events across the platform. This includes tracking user logins, booking actions, search queries, payment attempts, and API calls to identify trends, diagnose issues, and detect suspicious behavior or security breaches.
Data	User actions, timestamps, IP addresses, system events, error logs.
Stimulus	Admin accesses the "Activity Monitor" dashboard.
Response	The system displays a live feed or log of recent activities.
Comments	Help in identifying unusual patterns or potential security issues.

Table 36 - Monitor Activities

- Use Case: [Manage Reviews & feedback](#)

Actors	Admin, Review System
Description	The admin moderates all user-generated reviews and ratings to maintain the platform's credibility and trustworthiness. This involves proactively screening new reviews and responding to user-reported content to remove or edit feedback that contains inappropriate language, false information, spam, or content that violates the platform's community guidelines.
Data	Review ID, rating, content, author, date, flags/reports.
Stimulus	Admin selects a review from the management panel and chooses an action (e.g., delete).
Response	The review is modified or removed from the platform.
Comments	Ensures the quality and appropriateness of user-generated content.

Table 37 - Manage Reviews & feedback

- Use Case: Manage Vehicle Categories

Actors	Admin, System Database
Description	The admin defines and maintains the entire taxonomy of vehicles available on the platform. This includes creating new categories (e.g., "Electric SUVs"), editing existing ones, or archiving outdated categories. The admin specifies the attributes for each category, which directly controls the filters available to renters and the data fields owners must fill when creating a listing.
Data	Category name, description, icon, specifications template.
Stimulus	Admin navigates to "Vehicle Categories" and makes changes.
Response	The system updates the available categories for car listings.
Comments	Changes affect how owners list cars and how renters filter searches.

Table 38 - Manage Vehicle Categories

- Use Case: Edit Filter Options

Actors	Admin, System Database
Description	The admin configures and customizes the search and filter parameters available to renters on the car discovery page. This involves activating or deactivating specific filters (e.g., "Number of Doors"), defining the range or set of allowed values (e.g., setting the maximum price slider to 1000 SAR), and controlling the display order of filters to optimize the user experience and relevance of search results.
Data	Filter types (e.g., price, car type, features), allowed values, order.
Stimulus	Admin accesses "Search Settings" and updates the filter options.
Response	The search interface is updated with the new filter set.
Comments	Improves the search experience and relevance of results.

Table 39 - Edit Filter Options

- Use Case: Send Notifications & Announcements

Actors	Admin, Notification Service
Description	The admin creates and broadcasts system-wide or targeted messages to communicate directly with users. This is used for critical platform updates, promotional campaigns, new feature announcements, security alerts, or targeted messages to specific user segments (e.g., all owners in Riyadh). Notifications can be scheduled for future delivery and can include deep links to relevant app sections.
Data	Target audience, notification title, message, schedule, deep link.
Stimulus	Admin composes a message and clicks "Send Notification".
Response	The notification is delivered to the selected users.
Comments	Used for promotions, policy updates, or important system alerts.

Table 40 - Send Notifications & Announcements

- Use Case: Perform Security Audit

Actors	Admin, Security Tools
Description	The admin initiates a comprehensive, system-wide security scan to proactively identify vulnerabilities, misconfigurations, and potential threats. The audit analyzes user authentication logs, database access patterns, API call histories, and system configurations to flag issues like brute-force attack attempts, unusual data exports, or outdated software components, generating a detailed report with actionable recommendations.
Data	Audit logs, login attempts, access patterns, security flags.
Stimulus	Admin initiates a "Security Audit" from the admin panel.
Response	A report is generated listing potential security issues and recommendations.
Comments	Should be performed regularly to maintain system integrity.

Table 41 - Perform Security Audit

- Use Case: View Reports

Actors	Admin, Reporting System
Description	The admin generates, views, and analyzes detailed analytical reports on all aspects of the platform's performance. This includes financial reports (revenue, commissions), operational reports (booking volume, most popular car models), user growth analytics (new sign-ups, user retention), and marketplace health metrics (owner-to-renter ratio, booking cancellation rates). Reports can be filtered by custom date ranges and exported for further analysis.
Data	Booking volume, revenue, user growth, popular car models, time.
Stimulus	Admin selects report type and date range, then clicks "Generate Report".
Response	The system displays or exports the requested report (charts, graphs, tables).
Comments	Reports are crucial for business intelligence and strategic decisions.

Table 42 - View Reports

- Use Case: Backup & Restore Data

Actors	Admin, Database, Cloud Storage
Description	The admin performs critical data management operations to ensure business continuity. This includes manually initiating a full system backup (safeguarding all user accounts, listings, bookings, and transaction records) and, in a disaster recovery scenario, restoring the entire platform or specific data subsets from a previous backup point to minimize data loss and downtime.
Data	Database dumps, user files, transaction records, backup timestamp.
Stimulus	Admin selects "Backup Now" or chooses a backup file to restore.
Response	The system confirms a successful backup or completes the data restoration process.
Comments	Critical for disaster recovery and data integrity.

Table 43 - Backup & Restore Data

- Use Case: Block Suspicious Account

Actors	Admin, User Account System
Description	The admin investigates and takes definitive action against user accounts that exhibit malicious behavior, violates terms of service, or are flagged as suspicious by the system's automated monitoring. This action immediately deactivates the account, terminates all active sessions, cancels any pending bookings, and prevents the user from logging in or accessing any platform services.
Data	User ID, reason for blocking, timestamp, admin notes.
Stimulus	Admin selects an account from the user list and clicks "Block Account".
Response	The account is deactivated, and the user is logged out and prevented from accessing the platform.
Comments	A notification should be sent to the user explaining the reason for the block.

Table 44 - Block Suspicious Account

5. Non-Functional Requirement

5.1 Performance Requirements

The Fadl Zahr system shall ensure a smooth and responsive user experience under normal operating conditions. All essential actions such as searching, booking, and payment processing shall be executed within two seconds on average. The application shall support at least one thousand concurrent users without performance degradation or crashes. Database queries and record retrieval operations shall be optimized to complete within one second for typical use cases. The overall system uptime shall remain above ninety-nine percent, excluding planned maintenance periods, guaranteeing continuous service availability.

5.2 Safety Requirements

To protect data integrity and prevent loss, the system shall perform automatic backups of all essential records such as user profiles, booking histories, and payment transactions every twenty-four hours. Backup files shall be securely stored in an encrypted cloud environment. In case of a network failure, hardware malfunction, or server outage, the system shall restore data to the most recent stable state. Any critical user actions such as deleting a car listing or cancelling a confirmed booking shall require user confirmation to avoid unintended data loss.

5.3 Security Requirements

All communication between clients and the server shall be secured using SSL/TLS encryption to prevent unauthorized data interception. user passwords shall be hashed and stored using a secure encryption algorithm such as SHA-256. access to the application's primary features such as listing or booking vehicles shall only be granted to users who have successfully completed identity verification through the Absher platform. role-based access control shall be implemented to restrict each user group including admin, owner, and renter to the actions they are authorized to perform. Administrative accounts shall employ multi-factor authentication to ensure an additional layer of protection against unauthorized access.

5.4 Software Quality Attributes

The Fadl Zahr system shall maintain high software quality standards through the following attributes.

Usability: The mobile interface shall be simple, intuitive, and consistent across android and iOS platforms to ensure ease of use for all users.

Reliability: The system will operate with stability and handle unexpected failures gracefully without loss of data.

Maintainability: The software code shall be well-documented, modular, and structured to support future modifications, updates, and debugging.

Scalability: The system should be designed to handle an increasing number of users, cars, and transactions while maintaining optimal performance

Portability: The application shall be compatible with android version 7.0 and above, and iOS version 11.0 and above, ensuring accessibility for users on multiple devices.

5.5 Business Rules

Only verified car owners are permitted to list their vehicles for rent. a fixed service commission of ten percent shall be automatically deducted from every completed booking transaction. refunds for cancelled bookings shall be processed within forty-eight hours after administrative approval. submitted ratings and reviews shall not be modified or deleted once published. system administrators shall have the authority to block or suspend user accounts and remove any content that violates ethical standards or application policies.

6. Other Requirements

Legal and Regulatory Requirements:

- The application must comply with Saudi Arabian laws and regulations related to electronic transactions and data privacy.
- Integration with the Absher system for user identity verification must meet official government security and data protection standards.
- All financial transactions must follow the Saudi Arabian Monetary Authority (SAMA) guidelines for secure and verified electronic payments.
- The system must include clearly stated Terms of Service and a Privacy Policy that users must agree to before accessing the application.

Environmental and Technical Constraints

- The mobile application must operate effectively on both Android and iOS platforms.
- It should remain functional under varying internet conditions, including limited or unstable network connectivity.
- The system should be compatible with at least the two latest major versions of Android and iOS operating systems.
- The backend infrastructure should support scalability to handle an increasing number of users and transactions.

Project Constraints:

- The project should be developed using a secure and modern framework such as Node.js, Django, or Spring Boot for the backend, and Flutter or React Native for the frontend.
- Development must be completed within the academic project timeline specified by the course.
- Only authorized and publicly available APIs should be used for external integrations.

Documentation and Maintenance:

- Comprehensive user and administrator documentation must be provided.
- The project source code and related documents should be managed using version control tools such as GitHub.
- A maintenance plan should be established to address bug fixes, software updates, and future enhancements after deployment.