

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 Test Plan

For



Version 1

CS 411 – Software Engineering
Group 2

Subervised by:

Dr. Enas El-sayed El-sharawy

1 December 2025

This Software Test Plan 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 Test Documentation.

#	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
Table of Tables.....	6
1. Introduction.....	7
1.1 Objectives.....	7
1.2 Testing Strategy.....	7
1.3 Scope.....	9
1.4 References.....	9
1.5 Definitions and Acronyms.....	10
2. Test Items.....	11
2.1 Program Modules.....	11
2.2 Job Control Procedures.....	11
2.3 User Procedures.....	11
2.4 Operator Procedures.....	12
3. Features To Be Tested.....	12
4. Features Not to Be Tested.....	15
5. Approach.....	16
5.1 Component Testing.....	16
5.1.1 Common Functionalities.....	16
5.1.1.1 Login Test Cases.....	16
5.1.1.2 Sign Up Test Cases.....	21
5.1.1.3 Identity Verification Test Cases.....	25
5.1.1.4 Forget Password Test Cases.....	28
5.1.2 Renter.....	32
5.1.2.1 Search for cars Test Cases.....	32
5.1.2.2 Filter car Test Cases.....	34
5.1.2.3 View car details Test Cases.....	37
5.1.2.4 Create Booking Test Cases.....	40
5.1.2.5 Cancel Booking Test Cases.....	43
5.1.2.6 Booking history Test Cases.....	46
5.1.2.7 Ratings & Reviews Test Cases.....	48
5.1.2.8 Profile Management Test Cases.....	50
5.1.2.9 Notifications Test Cases.....	52
5.1.3 Owner.....	55
5.1.3.1 Add Car Listings Test Cases.....	55
5.1.3.2 Edit Car Listings Test Cases.....	58
5.1.3.3 Deactivate / Delete Car.....	62
5.1.3.4 Update Availability Calendar Test Cases.....	63
5.1.3.5 Approve or Reject Bookings Test Cases.....	65
5.1.3.6 Owner Dashboard & Earning Test Cases.....	66
5.1.3.7 Withdraw Earnings Test Cases.....	68
5.1.3.8 Owner Notification Test Cases.....	69

5.1.4 Admin.....	70
5.1.4.1 Manage User Test Cases.....	70
5.1.4.2 Manage Cars Test Cases.....	72
5.1.4.3 Manage Bookings Test Cases.....	74
5.1.4.4 Complaints & Report Test Cases.....	75
5.1.4.5 System Dashboard & analytic.....	77
5.2 Integration Testing.....	79
5.3 Conversion Testing.....	81
5.4 Job Stream Testing.....	82
5.5 Interface Testing.....	83
5.6 Security Testing.....	83
5.7 Recovery Testing.....	84
5.8 Performance Testing.....	85
5.9 Regression Testing.....	86
5.10 Acceptance Testing.....	87
5.11 Beta Testing.....	87
6. Pass / Fail Criteria.....	88
6.1 Suspension Criteria.....	88
6.2 Resumption Criteria.....	89
6.3 Approval Criteria.....	89
7. Testing Process.....	89
7.1 Test Deliverables.....	89
7.2 Testing Tasks.....	91
7.3 Responsibilities.....	93
7.4 Resources.....	93
7.5 Schedule.....	94
8. Environmental Requirements.....	95
8.1 Hardware.....	95
8.2 Software.....	95
8.3 Security.....	95
8.4 Tools.....	96
8.5 Publications.....	96
8.6 Risks and Assumptions.....	97
9. Chang Management Procedures.....	97
10. Plan Approvals.....	98

Revision History

Name	Date	Reason For Changes	Version
All members	Dec 1, 2025	Prepared initial version	0.1
All members	Dec 1, 2025	Updated section 1	0.2
All members	Dec 4, 2025	Updated section 3	0.3
All members	Dec 6, 2025	Updated section 5	0.4
All members	Dec 6, 2025	Updated section 6	0.5
All members	Dec 7, 2025	Updated section 7	0.6
All members	Dec 7, 2025	Updated section 8	0.7
All members	Dec 7, 2025	Update section 1	0.8
All members	Dec 8, 2025	Update section 5	0.9
All members	Dec 8, 2025	Complete review - Final version	1.0

Table of Tables

Table 1- Testing Strategy	7
Table 2- Definitions and Acronyms.....	10
Table 3- Features to be tested.....	12
Table 4- Features not to be tested.....	15
Table 5-21 Login Test Cases.....	16-20
Table 22-40 Sign Up Test Cases	21-25
Table 41-51 Identity Verification Test Cases.....	25-28
Table 52-66 Forgot Password Test Cases.....	28-32
Table 67-75 Search for cars Test Cases.....	32-34
Table 76-87 Filter car Test Cases.....	34-37
Table 88-98 View car details Test Cases.....	37-40
Table 99-111 Create Booking Test Cases.....	40-43
Table 112-121 Cancel Booking Test Cases.....	43-46
Table 122-131 Booking History Test Cases.....	46-48
Table 132-138 Ratings & Reviews Test Cases.....	48-50
Table 139-144 Profile Management Test Cases.....	50-51
Table 145-155 Notifications Test Cases.....	52-54
Table 156-169 Add Car Listings Test Cases.....	55-58
Table 170-183 Edit Car Listings Test Cases.....	58-62
Table 184-189 Deactivate / Delete Car.....	62-63
Table 190-194 Update Availability Calendar Test Cases.....	63-64
Table 195-201 Approve or Reject Bookings Test Cases.....	65-66
Table 202-206 Owner Dashboard & Earning Test Cases.....	66-67
Table 207-211 Withdraw Earnings Test Cases.....	68-69
Table 212-215 Owner Notification Test Cases.....	69-70
Table 216-225 Manage User Test Cases.....	70-72
Table 226-232 Manage Cars Test Cases.....	72-74
Table 233-237 Manage Bookings Test Cases.....	74-75
Table 238-244 Complaints & Report Test Cases.....	75-77
Table 245-250 System Dashboard & analytic.....	77-78
Table 251- Login Module ↔ User Roles.....	79
Table 252- Owner Module ↔ Car Listing Module.....	79
Table 253- Availability Calendar ↔ Booking Module.....	79
Table 254- Renter Search ↔ Availability ↔ Car Details.....	80
Table 255- Booking Request ↔ Owner Approval ↔ Renter Notification.....	80
Table 256- Payment Module ↔ Booking Module ↔ Earnings Module.....	80
Table 257- Earnings Module ↔ Withdrawal Module ↔ Admin Module.....	80
Table 258- Ratings Module ↔ Completed Booking.....	81
Table 259- Notification System ↔ All Modules.....	81
Table 260- Admin Actions ↔ User/Car/Booking Modules.....	81
Table 261- Responsibilities.....	93
Table 262- Schedule.....	94
Table 263- plan approvals.....	98

1. Introduction

The Introduction section of the Software Test Plan (STP) establishes an overview of the project, defines the overall testing philosophy, lists testing deliverables, and outlines how the STP will be maintained throughout the system development life cycle. It also identifies the reference documents, definitions, and acronyms related to the testing activities. This STP serves as the high-level document guiding all testing operations for the system, ensuring clarity, consistency, and structured execution of verification and validation processes.

1.1 Objectives

The objective of this Software Test Plan is to define the overall direction for all testing activities, including the scope, approach, resources, and schedule. This plan ensures that the system is validated and verified against the functional and non-functional requirements defined in the SDS and SRS.

The STP outlines the key testing deliverables, the main work activities for preparing and executing tests, the responsibilities of the testing team, and the major milestones aligned with the project timeline. It also identifies the necessary resources, environments, and estimated effort needed to complete testing effectively.

1.2 Testing Strategy

Section	Component
Introduction	Includes the adopted Testing Strategy for the FADL ZAHR system, outlining the objectives, testing scope, and overall quality goals.
Test Items	The test items of the software will be identified here. This includes all modules, features, and components that are subject to testing.
Features to Be Tested	The software features and all the different combinations of software functionalities that will be tested, including Authentication, Renter features, Owner features, Admin features, Payments, Notifications, and Non-functional requirements.
Features Not to Be Tested	The software features and combinations of features that will not be tested in this cycle, along with the reasons for exclusion (e.g., out-of-scope features, future enhancements, unavailable integrations).

Approach	The overall approaches used to test the system are described in this section. The approaches included in this testing plan are: <ul style="list-style-type: none"> • Component Testing • Integration Testing • Conversion Testing • Job Stream Testing • Interface Testing • Security Testing • Recovery Testing • Performance Testing • Regression Testing • Acceptance Testing • Beta Testing
Pass / Fail Criteria	The criteria used to determine whether each test item has passed or failed. This includes expected outputs, acceptance conditions, error-handling requirements, and threshold values for performance or reliability.
Testing Process	The methods, sequence, and execution flow used to perform the testing activities. Includes preparation, test execution, defect reporting, retesting, and validation processes.
Environmental Requirement	The necessary and desired properties of the test environment, including hardware, software, network conditions, mobile devices, browsers, and backend configurations.
Change Management Procedures	The STP change management process will be identified here. This includes how modifications to the test plan, test cases, and requirements will be requested, reviewed, and approved.
Plan Approvals	The formal approvals of team members, instructors, or stakeholders responsible for validating the designed test plan.

Table 1- Testing strategy

1.3 Scope

The scope of this Software Test Plan covers how testing activities will be planned, updated, and controlled throughout the development life cycle. Since testing is dependent on the progress of the system, this plan will be refined whenever new requirements, design changes, or updates occur. All revisions to STP will follow the project's management and version-control procedures to ensure accuracy and consistency. The STP applies to all levels of testing including unit, integration, system, and acceptance testing and outlines how updates will be communicated and distributed to the project team. Testing will continue throughout development to verify that each part of the system meets its defined requirements.

1.4 References

This section provides a complete list of all documents and other sources referenced in this Software Test Plan:

Project Documents:

- Project Plan: Software Engineering Project Plan for FADL ZAHR System
- Software Requirements Specification (SRS) for FADL ZAHR
- Software Design Specification (SDS) for FADL ZAHR

Standards and Guidelines:

- International Organization for Standardization & International Electrotechnical Commission. (2013). ISO/IEC 29119-1:2013: Software and systems engineering - Software testing - Part 1: Concepts and definitions.
- IEEE Computer Society. (2008). IEEE Standard for Software and System Test Documentation (IEEE 829-2008).
- ISTQB Foundation Level Syllabus, International Software Testing Qualifications Board.
- SWEBOK – Software Engineering Body of Knowledge (V3.0), IEEE Computer Society.

Academic References:

- Sommerville, Ian. Software Engineering, 10th Edition. Pearson Education, 2016.

Organization Policies:

- Imam Abdulrahman bin Faisal University Academic Guidelines
- Computer Science Department Project Standards

1.5 Definitions and Acronyms

Term / Acronyms	Definition
STP	Software Test Plan – The document outlining the testing strategy, scope, resources, and schedule for the FADL ZAHR system
SRS	Software Requirements Specification – A document that describes the functional and non-functional requirements of the system.
SDS	Software Design Specification – A document detailing the system architecture, modules, and design decisions.
UI	User Interface – The visual and interactive part of the application through which users interact with the system.
API	Application Programming Interface – A set of protocols and tools for building software and enabling communication between different systems.
Firebase	A backend platform developed by Google used for database management, authentication, and real-time data synchronization in the FADL ZAHR system.
Absher	A Saudi governmental platform used for identity verification and citizen services; integrated into the system for user verification.
STC Pay / Mada	Payment gateways used in Saudi Arabia for processing online transactions.
OTP	One-Time Password – A temporary code sent via SMS or email for verification purposes, such as password reset or login.
IBAN	International Bank Account Number – A standardized international numbering system for identifying bank accounts, used for earnings withdrawal.
CRUD	Create, Read, Update, delete – The four basic operations performed on database records.
UAT	User Acceptance Testing – The final phase of testing where real users validate the system against business requirements.
KPI	Key Performance Indicator – A measurable value that demonstrates how effectively the system is achieving key objectives.
RTM	Requirements Traceability Matrix – A document that maps requirements to test cases to ensure full coverage.
Admin	Administrator – A user role with full access to manage users, cars, bookings, and system settings.
Renter	A user who rents cars from owners through the platform.
Owner	A user who lists and manages their cars for rent on the platform.
Booking	A reservation made by a renter for a specific car and date range.
Pending	A status indicating that an action (e.g, booking, car listing) is awaiting approval or confirmation.
Verified	A user status confirming that identity verification has been successfully completed.

Table 2- Definitions and Acronyms

2. Test Items

The system will be tested across all its main items, including the user interfaces, backend modules, and database operations. Each item will be checked to ensure that it behaves as intended and supports the workflows described in the SRS and SDS.

2.1 Program Modules

The program modules of the system will be tested individually to confirm that each part functions as expected. user interface modules will be checked by running the main interactions on every screen, including login, identity verification, car search, car details, booking and payment, booking history, owner dashboard, booking requests, and the admin dashboard. backend modules will be tested by verifying the behavior of authentication, vehicle management, booking operations, payment handling, and notification services. database modules will be tested by validating data retrieval, updates, and consistency across all related tables. Each module will be tested on its own first, and later in combination with the modules it interacts with.

2.2 Job Control Procedures

The testing process for the system will follow three main activities: Unit Testing, Component Testing, and System Testing. Each unit (method, class, or service) will first be isolated and tested individually to ensure the correctness of its internal logic. After that, several related units belonging to the same module such as the Sign-In module, Car Search module, or Booking module will be integrated and tested together as combined components.

Finally, all modules of the system, including user interfaces, database operations, and external service integrations (Absher verification and Payment Gateway), will be fully integrated and tested as a complete system to confirm that the entire application functions smoothly as a whole.

2.3 User Procedures

All user procedures will be tried out in a way that reflects how the system is used. The goal is simply to make sure that every user type can move through their tasks smoothly and that nothing unexpected gets in the way. for renters, this means checking that the basic flow works in (completing identity verification, searching for a car, opening the details page, submitting a booking request, completing payment, and later going back to view their booking history). for owners, the procedures focus on the actions they rely on (adding a vehicle, updating its information, viewing booking requests, responding to those requests, and checking their earnings). for the admin, the tested procedures include (opening the admin dashboard, monitoring what's happening in the system, managing user accounts, and reviewing the reports available to them).

2.4 Operator Procedures

operators will prepare the testing environment by setting up the required services, confirming database accessibility, and making sure that the identity verification and payment integrations are active. They will reset or load test data when needed, verify that all system modules can be reached, and monitor logs during the execution of the tests. Operators will also keep the environment stable by restarting services if necessary and ensuring that no conflicts or interruptions occur while tests are running.

3. Features To Be Tested

User	Features	Design Specification
Common	Log In	A test to verify the sign-in credentials of all users (Admin/Owner/Renter) and ensure only valid accounts can access the system.
	Sign Up	A test to verify the registration process, ensuring required data is entered correctly and that email/National ID are unique.
	Identity Verification	A test to ensure that the identity verification process is completed correctly and securely.
	Forget My Password	A test to verify that users can reset their passwords using a valid email/identifier and that a new password is stored correctly.
Renter	Search for Cars	A test to ensure that renters can search for cars based on keywords, location, date, and availability correctly.
	Filter Car Results	A test to verify that car filters (price, type, transmission, etc.) work properly and display only matching cars.
	View Car Details	A test to ensure that complete and correct car details, owner info, pricing, and availability are displayed.

Owner	Create Booking	A test to ensure that renters can create bookings with valid dates and that bookings are stored and reflected in availability.
	Cancel Booking	A test to verify that renters can cancel bookings and that booking status and availability are updated correctly.
	Booking History	A test to ensure that past and current bookings are listed accurately with correct details for each renter.
	Ratings & Reviews	A test to verify that renters can rate and review cars/owners and that these ratings are saved and displayed correctly.
	Profile Management	A test to ensure renters can view and update their personal information and password correctly in the system.
	Notifications	A test to verify that renters receive correct notifications (booking status, reminders, updates) through the system.
	Add Car Listings	A test to ensure that owners can add new car listings with full details and that these cars appear correctly in searches.
	Edit Car Listings	A test to verify that owners can modify existing car details and that changes are reflected throughout the system.
	Deactivate / Delete Car	A test to ensure that owners can deactivate or delete car listings and that removed cars are not rentable anymore.
		A test to verify that owners can update car availability dates and

Admin	Update Availability Calendar	that search/booking respects these changes.
	Approve or Reject Bookings	A test to ensure that owners can approve or reject booking requests and that the booking status updates correctly.
	Owner Dashboard & Earnings	A test to verify that owners can view their earnings, booking statistics, and car performance correctly.
	Withdraw Earnings	A test to ensure that owners can request withdrawals and that the withdrawal amount and status are handled correctly.
	Notifications	A test to verify that owners receive correct notifications about new bookings, cancellations, and payment events.
User	Manage Users	A test to ensure that the admin can view, add, update, and disable user accounts without affecting other functionalities.
	Manage Cars	A test to verify that the admin can review, approve, or remove car listings and that changes are reflected to users.
	Manage Bookings	A test to ensure that the admin can view and override bookings (e.g., resolve conflicts) and that booking states remain consistent.
	Complaints & Reports	A test to verify that the admin can view, track, and resolve complaints/reports submitted by users.
	System Dashboard & Analytics	A test to ensure that the admin can view accurate system statistics, usage metrics, and

		reports for monitoring and decision making.
--	--	---

Table 3 – Features to be tested

4. Features Not to Be Tested

All essential system functionalities have been included in the testing process to ensure reliability and proper performance. However, some features do not require testing because they do not affect system behavior, data integrity, or core workflows. These features are listed below.

Features Not to Be Tested	Reasons
Absher System Integration (Real Environment)	It is not necessary to test since real-environment Absher integration is outside the current scope and relies on external government systems not available for test simulation.
Advanced Analytics and Reporting Dashboards	These dashboards are future enhancements and do not affect current transactional flow; therefore, testing is not required at this stage.
Future Marketing or Loyalty Modules	These modules are not part of the current version and have no effect on system operations, so testing them is unnecessary.
Non-Core Features (Extra Languages, Web Version)	These are optional add-ons that do not impact the core mobile application functionality and can be tested in later phases.
Hardware-Specific Behavior Beyond Basic GPS Usage	The system only depends on standard GPS functionality; testing advanced or device-specific hardware behavior is not required as it does not impact system reliability.

Table 4 – Features not to be tested

5. Approach

5.1 Component Testing

Component testing will be carried out on all individual modules of the system before integrating them. Each component including Authentication, Identity Verification, Car Listing, Booking, Payment, Ratings & Reviews, and Admin Controls will be isolated and tested to ensure that:

- Inputs and outputs behave as expected
- Internal logic executes correctly
- Data is stored and retrieved from Firebase properly
- External APIs (Absher, STC Pay/Mada) respond correctly when called

Both positive and negative test cases will be applied to verify robustness and error handling.

5.1.1 Common Functionalities

5.1.1.1 Login Test Cases

Field	Description
Test ID	Login_001
Prerequisite	Valid user account exists with stored ID and password.
Test Procedure	Enter correct ID and correct password, then click Login.
Expected Result	User is successfully logged in and redirected to the correct homepage/dashboard according to their role.
Actual Result	User is successfully logged in and redirected to the correct homepage/dashboard according to their role.
Verified (Yes/No)	Yes

Table 5 - Correct ID and Correct Password

Field	Description
Test ID	Login_002
Prerequisite	Valid user account exists.
Test Procedure	Enter correct ID and incorrect password, then click Login.
Expected Result	Login fails and an error message appears: “The ID or password you entered is incorrect. Please try again.”
Actual Result	Login fails and an error message appears: “The ID or password you entered is incorrect. Please try again.”
Verified (Yes/No)	Yes

Table 6 - Correct ID and Incorrect Password

Field	Description
Test ID	Login_003
Prerequisite	None.
Test Procedure	Leave ID field empty, enter correct password, then click Login.
Expected Result	Login is blocked and a validation message appears: "Please enter your ID."
Actual Result	Login is blocked and a validation message appears: "Please enter your ID."
Verified (Yes/No)	Yes

Table 7 - Empty ID and Correct Password

Field	Description
Test ID	Login_004
Prerequisite	None.
Test Procedure	Enter correct ID, leave password field empty, then click Login.
Expected Result	Login is blocked and a validation message appears: "Please enter your password."
Actual Result	Login is blocked and a validation message appears: "Please enter your password."
Verified (Yes/No)	Yes

Table 8- Correct ID and Empty Password

Field	Description
Test ID	Login_005
Prerequisite	None.
Test Procedure	Enter incorrect ID and correct password, then click Login.
Expected Result	Login fails and an error message appears: "The ID or password you entered is incorrect. Please try again."
Actual Result	Login fails and an error message appears: "The ID or password you entered is incorrect. Please try again."
Verified (Yes/No)	Yes

Table 9 - incorrect ID and correct password

Field	Description
Test ID	Login_006
Prerequisite	Stored password is case-sensitive and not all caps.
Test Procedure	Enter correct ID and the same password but in all capital letters, then click Login.
Expected Result	Login fails and an error message appears: "The ID or password you entered is incorrect. Please try again."
Actual Result	Login fails and an error message appears: "The ID or password you entered is incorrect. Please try again."
Verified (Yes/No)	Yes

Table 10 - Correct ID and capitalized password

Field	Description
Test ID	Login_007
Prerequisite	None.
Test Procedure	Leave ID and password fields empty, then click Login.
Expected Result	Login is blocked and validation messages appear for both fields, e.g., “Please enter your ID” and “Please enter your password.”
Actual Result	Login is blocked and validation messages appear for both fields, e.g., “Please enter your ID” and “Please enter your password.”
Verified (Yes/No)	Yes

Table 11 – both ID and password empty

Field	Description
Test ID	Login_008
Prerequisite	Account is configured to accept email or ID in the same field; valid email is stored.
Test Procedure	Enter registered email address in ID field and correct password, then click on Login.
Expected Result	User is successfully logged in and redirected to the correct homepage/dashboard.
Actual Result	User is successfully logged in and redirected to the correct homepage/dashboard.
Verified (Yes/No)	Yes

Table 12- login using email

Field	Description
Test ID	Login_009
Prerequisite	Email format validation is implemented.
Test Procedure	Enter invalid email format (e.g., user@wrong) in ID field and any password, then click Login.
Expected Result	Login is blocked and a validation message appears, e.g., “Please enter a valid email or ID.”
Actual Result	Login is blocked and a validation message appears, e.g., “Please enter a valid email or ID.”
Verified (Yes/No)	Yes

Table 13 - invalid email format

Field	Description
Test ID	Login_010
Prerequisite	Valid ID exists (e.g., 2230001749).
Test Procedure	Type the correct ID with spaces before and after it (e.g., " 2230001749 "), enter correct password, click Login.
Expected Result	System trims spaces, recognizes the ID correctly, and logs the user in successfully.
Actual Result	System trims spaces, recognizes the ID correctly, and logs the user in successfully.
Verified (Yes/No)	Yes

Table 14 - leading and trailing spaces in ID

Field	Description
Test ID	Login_011
Prerequisite	Valid account exists.
Test Procedure	Enter correct ID, and password with extra spaces around it (e.g., " pass1234 "), click Login.
Expected Result	If spaces are not allowed/trimmed, login fails with "incorrect password" error.
Actual Result	Login fails with "incorrect password" error because of extra spaces.
Verified (Yes/No)	Yes

Table 15 - leading and trailing spaces in password

Field	Description
Test ID	Login_012
Prerequisite	User account exists but identity is not verified.
Test Procedure	Enter correct ID and password for an unverified user, then click Login.
Expected Result	Login is blocked or redirected; message appears: "Please verify your identity before accessing the system," with a button to go to verification screen.
Actual Result	Login is blocked or redirected; message appears: "Please verify your identity before accessing the system," with a button to go to verification screen.
Verified (Yes/No)	Yes

Table 16 - unverified account attempt to login

Field	Description
Test ID	Login_013
Prerequisite	User account exists with status Suspended/Blocked in database.
Test Procedure	Enter correct ID and password of suspended user, then click Login.
Expected Result	Login fails; message appears: "Your account is suspended. Please contact support or the administrator."
Actual Result	Login fails; message appears: "Your account is suspended. Please contact support or the administrator."
Verified (Yes/No)	Yes

Table 17 - Suspended / Blocked User Attempts Login

Field	Description
Test ID	Login_014
Prerequisite	Same credentials cannot be used across all roles; user has a specific role (e.g., Owner).
Test Procedure	Enter correct ID and correct password but choose the wrong user type from the dropdown (e.g., select Admin instead of Owner), then click Login.
Expected Result	Login fails and error message appears: "User type does not match this account."
Actual Result	Login fails and error message appears: "User type does not match this account."
Verified (Yes/No)	Yes

Table 18 - Wrong User Type Selected (Renter vs Owner vs Admin)

Field	Description
Test ID	Login_015
Prerequisite	Account exists; system has rule to temporarily block after several failed attempts.
Test Procedure	Enter correct ID and incorrect password 5 times in a row, then attempt a 6th login with correct password.
Expected Result	After the defined threshold, system temporarily locks login for that account or device and shows message: “Too many failed attempts. Please try again later or reset your password.” The 6th attempt is also blocked.
Actual Result	System temporarily locks login and shows message: “Too many failed attempts. Please try again later or reset your password.”
Verified (Yes/No)	Yes

Table 19 - Multiple Failed Attempts (Rate Limiting)

Field	Description
Test ID	Login_016
Prerequisite	Device is offline (Wi-Fi/data turned off).
Test Procedure	Enter correct ID and password, then click Login while there is no internet connection.
Expected Result	Login fails; message appears: “No internet connection. Please check your network and try again.” No request is sent to server.
Actual Result	Login fails; message appears: “No internet connection. Please check your network and try again.”
Verified (Yes/No)	Yes

Table 20 - No Internet Connection While Logging In

Field	Description
Test ID	Login_017
Prerequisite	Password field has a visibility (eye) icon.
Test Procedure	Type a password while it is hidden (●●●●). Tap the eye icon to show, then tap again to hide.
Expected Result	First tap changes field to plain text so the user can see the password; second tap returns it to hidden dots; the value itself does not change.
Actual Result	First tap changes field to plain text; second tap hides it again; password value remains unchanged.
Verified (Yes/No)	Yes

Table 21 - show/hide password icon

5.1.1.2 Sign Up Test Cases

Field	Description
Test ID	SignUp_001
Prerequisite	Email/phone not previously registered.
Test Procedure	Enter valid name, valid phone/email, strong password, matching confirm password, select Renter, click Sign Up.
Expected Result	Account is created successfully; user is redirected to verification or home screen.
Actual Result	Account is created successfully; user is redirected to verification or home screen.
Verified (Yes/No)	Yes

Table 22- correct data for renter sign up

Field	Description
Test ID	SignUp_002
Prerequisite	Email/phone not previously registered.
Test Procedure	Enter correct data and select Owner, then click Sign Up.
Expected Result	Owner account is created; user is redirected to owner onboarding.
Actual Result	Owner account is created; user is redirected to owner onboarding.
Verified (Yes/No)	Yes

Table 23- correct data for owner sign up

Field	Description
Test ID	SignUp_003
Prerequisite	None.
Test Procedure	Enter correct data but use different passwords, click Sign Up.
Expected Result	Error: “Passwords do not match.”
Actual Result	Error: “Passwords do not match.”
Verified (Yes/No)	Yes

Table 24- password and confirm password does not match

Field	Description
Test ID	SignUp_004
Prerequisite	Password policy enforced (min length, numbers, special chars).
Test Procedure	Enter password that does not meet security rules (e.g., “12345”), click Sign Up.
Expected Result	Error: “Password does not meet security requirements.”
Actual Result	Error: “Password does not meet security requirements.”
Verified (Yes/No)	Yes

Table 25-week password

Field	Description
Test ID	SignUp_005
Prerequisite	Email or phone already registered.
Test Procedure	Enter account information using an email/phone already in system.
Expected Result	Error: “This email/phone is already registered.”
Actual Result	Error: “This email/phone is already registered.”
Verified (Yes/No)	Yes

Table 26- existing email or phone number

Field	Description
Test ID	SignUp_006
Prerequisite	Email validation enabled.
Test Procedure	Enter invalid email format (e.g., test@wrong) and click Sign Up.
Expected Result	Error: “Please enter a valid email address.”
Actual Result	Error: “Please enter a valid email address.”
Verified (Yes/No)	Yes

Table 27-invalid email format

Field	Description
Test ID	SignUp_007
Prerequisite	None.
Test Procedure	Leave one or more required fields empty (name, phone, password...), click Sign Up.
Expected Result	Error showing which fields are missing.
Actual Result	Error showing which fields are missing.
Verified (Yes/No)	Yes

Table 28-missing required fields

Field	Description
Test ID	SignUp_008
Prerequisite	Age requirement exists.
Test Procedure	Enter valid details but age < 18, click Sign Up.
Expected Result	Error: “You must be at least 18 years old to register.”
Actual Result	Error: “You must be at least 18 years old to register.”
Verified (Yes/No)	Yes

Table 29- Under age renter

Field	Description
Test ID	SignUp_009
Prerequisite	Checkbox exists.
Test Procedure	Enter correct data but do not check Terms & Conditions, click Sign Up.
Expected Result	Error: “You must agree to the Terms & Conditions.”
Actual Result	Error: “You must agree to the Terms & Conditions.”
Verified (Yes/No)	Yes

Table 30 - terms and conditions not accepted

Field	Description
Test ID	SignUp_010
Prerequisite	None.
Test Procedure	Enter values with spaces before/after (e.g., “ Nada ”), click Sign Up.
Expected Result	System trims spaces and processes data normally; account is created.
Actual Result	System trims spaces and processes data normally; account is created.
Verified (Yes/No)	Yes

Table 31-leading and trailing spaces in fields

Field	Description
Test ID	SignUp_011
Prerequisite	Phone validation active.
Test Procedure	Enter a phone number with invalid length or characters (e.g., “056abc123”), then click Sign Up.
Expected Result	Error: “Please enter a valid phone number.”
Actual Result	Error: “Please enter a valid phone number.”
Verified (Yes/No)	Yes

Table 32-invalid phone number format

Field	Description
Test ID	SignUp_012
Prerequisite	Phone exists in database.
Test Procedure	Enter correct details but use an existing phone number, then click Sign Up.
Expected Result	Error: “This phone number is already registered.”
Actual Result	Error: “This phone number is already registered.”
Verified (Yes/No)	Yes

Table 33-phone number already registered

Field	Description
Test ID	SignUp_013
Prerequisite	Name field has character validation.
Test Procedure	Enter name such as “N🌟da” and click Sign Up.
Expected Result	Error: “Name contains invalid characters.”
Actual Result	Error: “Name contains invalid characters.”
Verified (Yes/No)	Yes

Table 34-using emoji or invalid character in name

Field	Description
Test ID	SignUp_014
Prerequisite	Character limit defined (example: 50 chars).
Test Procedure	Enter name exceeding the allowed length (e.g., 100 characters) and click Sign Up.
Expected Result	Error: “Name is too long.”
Actual Result	Error: “Name is too long.”
Verified (Yes/No)	Yes

Table 35-using very long name, exceeding max length

Field	Description
Test ID	SignUp_015
Prerequisite	Password must include letters and symbols.
Test Procedure	Enter password like “12345678”, then click Sign Up.
Expected Result	Error: “Password must include letters or special characters.”
Actual Result	Error: “Password must include letters or special characters.”
Verified (Yes/No)	Yes

Table 36-entering password only numbers

Field	Description
Test ID	SignUp_016
Prerequisite	Password requires numbers and symbols.
Test Procedure	Enter password like “passwordonly”, then click Sign Up.
Expected Result	Error: “Password must include numbers or special characters.”
Actual Result	Error: “Password must include numbers or special characters.”
Verified (Yes/No)	Yes

Table 37-entering password only letters

Field	Description
Test ID	SignUp_017
Prerequisite	Special character requirement enabled.
Test Procedure	Enter password without any special character, e.g., “Password12”, then click Sign Up.
Expected Result	Error: “Password must include at least one special character.”
Actual Result	Error: “Password must include at least one special character.”
Verified (Yes/No)	Yes

Table 38-enter entering password without special character

Field	Description
Test ID	SignUp_018
Prerequisite	User type selection required (Renter / Owner).
Test Procedure	Enter all data correctly but do not select user type, then click Sign Up.
Expected Result	Error: “Please select a user type.”
Actual Result	Error: “Please select a user type.”
Verified (Yes/No)	Yes

Table 39-try to sign up without choosing user type

Field	Description
Test ID	SignUp_019
Prerequisite	Internet disabled.
Test Procedure	Enter all fields correctly, then click Sign Up while device is offline.
Expected Result	Error: “No internet connection. Please check your network.”
Actual Result	Error: “No internet connection. Please check your network.”
Verified (Yes/No)	Yes

Table 40-Internet connection lost during sign-up

5.1.1.3 Identity Verification Test Cases

Field	Description
Test ID	Verify_001
Prerequisite	User account exists and is not verified.
Test Procedure	Enter valid National ID + full name + date of birth and submit.
Expected Result	Verification succeeds and user status becomes Verified.
Actual Result	Verification succeeds and user status becomes Verified.
Verified (Yes/No)	Yes

Table 41-valid national ID and correct data

Field	Description
Test ID	Verify_002
Prerequisite	None
Test Procedure	Enter an ID containing letters or wrong length (e.g., 12 digits).
Expected Result	Error: “Invalid National ID format.”
Actual Result	Error: “Invalid National ID format.”
Verified	Yes

Table 42- valid national ID format

Field	Description
Test ID	Verify_003
Prerequisite	National ID exists in government database.
Test Procedure	Enter correct ID but different name.
Expected Result	Verification fails: “Data does not match records.”
Actual Result	Verification fails: “Data does not match records.”
Verified	Yes

Table 43- correct national ID but wrong name

Field	Description
Test ID	Verify_004
Prerequisite	Government data exists.
Test Procedure	Enter correct ID + correct name + wrong DOB.
Expected Result	Error: “Date of birth does not match.”
Actual Result	Error: “Date of birth does not match.”
Verified	Yes

Table 44-correct national idea, but wrong date of birth

Field	Description
Test ID	Verify_005
Prerequisite	Device disconnected.
Test Procedure	Enter all valid data and press submit while offline.
Expected Result	Error: “No internet connection.”
Actual Result	Error: “No internet connection.”
Verified	Yes

Table 45- verifying while off-line

Field	Description
Test ID	Verify_006
Prerequisite	Slow server or simulated delay.
Test Procedure	Submit verification and wait until timeout.
Expected Result	Error: “Verification service not responding. Try again.”
Actual Result	Error: “Verification service not responding. Try again.”
Verified	Yes

Table 46-server timeout

Field	Description
Test ID	Verify_007
Prerequisite	Account is already verified.
Test Procedure	Open verification screen and submit again.
Expected Result	System says: “You are already verified.”
Actual Result	System says: “You are already verified.”
Verified	Yes

Table 47-already verified user tries again

Field	Description
Test ID	Verify_008
Prerequisite	On verification screen.
Test Procedure	User presses back before submitting.
Expected Result	No verification occurs; user returns back.
Actual Result	No verification occurs; user returns back.
Verified	Yes

Table 48-cancel verification process

Field	Description
Test ID	Verify_09
Prerequisite	ID expired in system.
Test Procedure	Enter valid but expired ID.
Expected Result	Error: “Your ID is expired.”
Actual Result	Error: “Your ID is expired.”
Verified	Yes

Table 49-verifying with expired national ID

Field	Description
Test ID	Verify_010
Prerequisite	Force API error 500.
Test Procedure	Submit valid info.
Expected Result	Error: “Verification failed. Try again later.”
Actual Result	Error: “Verification failed. Try again later.”
Verified	Yes

Table 50-verification ABI returns unknown error

Field	Description
Test ID	Verify_011
Prerequisite	None
Test Procedure	Enter ID with spaces: " 1234567890 ".
Expected Result	Spaces trimmed automatically; verification proceeds normally.
Actual Result	Spaces trimmed automatically; verification proceeds normally.
Verified	Yes

Table 51-national ID with extra spaces

5.1.1.4 Forgot Password Test Cases

Field	Description
Test ID	FP_001
Prerequisite	Email exists in database.
Test Procedure	Enter registered email and click Send Reset Link.
Expected Result	Reset link/OTP is sent successfully.
Actual Result	Reset link/OTP is sent successfully.
Verified	Yes

Table 52 -valid email request

Field	Description
Test ID	FP_002
Prerequisite	Email not in database.
Test Procedure	Enter unregistered email and click Send Reset Link.
Expected Result	Error: “Email not found.”
Actual Result	Error: “Email not found.”
Verified	Yes

Table 53-unregistered email

Field	Description
Test ID	FP_003
Prerequisite	None
Test Procedure	Enter invalid email format (e.g., user@wrong) and click Send Reset Link.
Expected Result	Error: “Please enter a valid email.”
Actual Result	Error: “Please enter a valid email.”
Verified	Yes

Table 54 -invalid email format

Field	Description
Test ID	FP_004
Prerequisite	None
Test Procedure	Leave the email field empty and submit.
Expected Result	Error: “Email is required.”
Actual Result	Error: “Email is required.”
Verified	Yes

Table 55-empty email field

Field	Description
Test ID	FP_005
Prerequisite	Phone number exists in system.
Test Procedure	Enter a valid phone number and submit.
Expected Result	Reset OTP is sent via SMS.
Actual Result	Reset OTP is sent via SMS.
Verified	Yes

Table 56-valid phone number

Field	Description
Test ID	FP_006
Prerequisite	Phone not in database.
Test Procedure	Enter unregistered phone number.
Expected Result	Error: “Phone number not found.”
Actual Result	Error: “Phone number not found.”
Verified	Yes

Table 57-unregistered phone number

Field	Description
Test ID	FP_007
Prerequisite	Validation active.
Test Procedure	Enter invalid phone number (letters, short/long number).
Expected Result	Error: “Invalid phone number format.”
Actual Result	Error: “Invalid phone number format.”
Verified	Yes

Table 58-invalid phone format

Field	Description
Test ID	FP_008
Prerequisite	Device offline.
Test Procedure	Enter valid email then click submit while offline.
Expected Result	Error: “No internet connection.”
Actual Result	Error: “No internet connection.”
Verified	Yes

Table 59-Internet disconnected during request

Field	Description
Test ID	FP_009
Prerequisite	Server intentionally slowed.
Test Procedure	Request reset with slow connection.
Expected Result	Error: “Request timed out. Try again.”
Actual Result	Error: “Request timed out. Try again.”
Verified	Yes

Table 60-server timeout

Field	Description
Test ID	FP_010
Prerequisite	Valid email exists.
Test Procedure	Submit email and open inbox.
Expected Result	Reset link appears within user’s inbox.
Actual Result	Reset link appears within user’s inbox.
Verified	Yes

Table 61-reset link generated successfully

Field	Description
Test ID	FP_011
Prerequisite	Old expired reset link.
Test Procedure	Click an expired password reset link.
Expected Result	Error: “Reset link expired.”
Actual Result	Error: “Reset link expired.”
Verified	Yes

Table 62-reset link expired

Field	Description
Test ID	FP_012
Prerequisite	Link must be single use.
Test Procedure	Use reset link once successfully, then click it again.
Expected Result	Error: “This link has already been used.”
Actual Result	Error: “This link has already been used.”
Verified	Yes

Table 63-recently link used twice

Field	Description
Test ID	FP_013
Prerequisite	User opened reset page with valid token.
Test Procedure	Enter strong new password + matching confirmation.
Expected Result	Password updated successfully; user can log in with new password.
Actual Result	Password updated successfully; user can log in with new password.
Verified	Yes

Table 64-valid reset password entry

Field	Description
Test ID	FP_014
Prerequisite	Valid reset session.
Test Procedure	Enter password and another different confirm password.
Expected Result	Error: “Passwords do not match.”
Actual Result	Error: “Passwords do not match.”
Verified	Yes

Table 65-no matching new password

Field	Description
Test ID	FP_015
Prerequisite	Password strength rules applied.
Test Procedure	Enter weak new password like “12345”.
Expected Result	Error: “Password is too weak.”
Actual Result	Error: “Password is too weak.”
Verified	Yes

Table 66-week new password

5.1.2 Renter

5.1.2.1 Search for cars Test Cases

Field	Description
Test ID	Search_001
Prerequisite	Cars available in selected city and date range.
Test Procedure	Select city “Riyadh”, choose valid start/end dates, click Search.
Expected Result	List of available cars is displayed.
Actual Result	List of available cars is displayed.
Verified	Yes

Table 67-search with valid city and dates

Field	Description
Test ID	Search_002
Prerequisite	Cars exist in that city.
Test Procedure	Select city only → press Search.
Expected Result	System shows cars in that city or prompts users to pick dates depending on design.
Actual Result	System shows cars in that city.
Verified	Yes

Table 68-search with city only

Field	Description
Test ID	Search_003
Prerequisite	None
Test Procedure	Choose End date earlier than Start date, click Search.
Expected Result	Error: “Invalid date range.”
Actual Result	Error: “Invalid date range.”
Verified	Yes

Table 69-search with date range

Field	Description
Test ID	Search_004
Prerequisite	No cars available match criteria.
Test Procedure	Search for city/dates with no cars.
Expected Result	“No cars available” message appears.
Actual Result	“No cars available” message appears.
Verified	Yes

Table 70 -search where no cars are available

Field	Description
Test ID	Search_005
Prerequisite	Input validation active.
Test Procedure	Enter an extremely long string in city field.
Expected Result	Error or input restricted to valid character limit.
Actual Result	Input restricted to valid character limit.
Verified	Yes

Table 71 -searched with very long city name

Field	Description
Test ID	Search_006
Prerequisite	City names are case-insensitive.
Test Procedure	Enter “riyadh” (lowercase).
Expected Result	Search recognized as “Riyadh”. Results displayed normally.
Actual Result	Search recognized as “Riyadh”. Results displayed normally.
Verified	Yes

Table 72-search using lowercase city name

Field	Description
Test ID	Search_007
Prerequisite	Validation enabled.
Test Procedure	Enter “Riy@dh!!”.
Expected Result	Error: “Invalid city name.”
Actual Result	Error: “Invalid city name.”
Verified	Yes

Table 73-search with a special character in city field

Field	Description
Test ID	Search_008
Prerequisite	Device offline.
Test Procedure	Try to search while disconnected.
Expected Result	Error: “No internet connection.”
Actual Result	Error: “No internet connection.”
Verified	Yes

Table 74 -search while Internet off

Field	Description
Test ID	Search_09
Prerequisite	Server simulated downtime.
Test Procedure	Enter valid criteria and click Search.
Expected Result	Error: “Service unavailable. Try again later.”
Actual Result	Error: “Service unavailable. Try again later.”
Verified	Yes

Table 75 -search why a server is down

5.1.2.2 Filter car Test Cases

Field	Description
Test ID	Filter_001
Prerequisite	Search results include cars with different prices.
Test Procedure	Set min price and max price and apply filter.
Expected Result	Only cars whose price per day is within the selected range are displayed.
Actual Result	Only cars whose price per day is within the selected range are displayed.
Verified (Yes/No)	Yes

Table 76 -apply minimum at maximum price filter

Field	Description
Test ID	Filter_002
Prerequisite	No cars exist in selected price range.
Test Procedure	Set min/max price such that no car matches.
Expected Result	“No cars found” message shown.
Actual Result	“No cars found” message shown.
Verified (Yes/No)	Yes

Table 77-price filter return no result

Field	Description
Test ID	Filter_003
Prerequisite	Search results include multiple brands.
Test Procedure	Select brand “Toyota” and apply filter.
Expected Result	Only Toyota cars appear in the list.
Actual Result	Only Toyota cars appear in the list.
Verified (Yes/No)	Yes

Table 78-filter by car brand

Field	Description
Test ID	Filter_004
Prerequisite	Results include multiple car types.
Test Procedure	Select SUV in car type filter.
Expected Result	Only SUVs are shown.
Actual Result	Only SUVs are shown.
Verified (Yes/No)	Yes

Table 79-filter by car type

Field	Description
Test ID	Filter_005
Prerequisite	Some cars have ratings.
Test Procedure	Set rating filter to 4+ stars.
Expected Result	Only cars with rating ≥ 4.0 are shown.
Actual Result	Only cars with rating ≥ 4.0 are shown.
Verified (Yes/No)	Yes

Table 80-filter by rating

Field	Description
Test ID	Filter_006
Prerequisite	Data set supports combined conditions.
Test Procedure	Apply brand “Hyundai”, price 150–300, rating 4+.
Expected Result	Only cars that satisfy all selected filters are displayed.
Actual Result	Only cars that satisfy all selected filters are displayed.
Verified (Yes/No)	Yes

Table 81-combine multiple filters

Field	Description
Test ID	Filter_007
Prerequisite	One or more filters are active.
Test Procedure	Tap “Clear Filters” button.
Expected Result	All filters are reset, and full unfiltered search results are shown again.
Actual Result	All filters are reset, and full unfiltered search results are shown again.
Verified (Yes/No)	Yes

Table 82 - clear all filters

Field	Description
Test ID	Filter_008
Prerequisite	No search results (empty list).
Test Procedure	Open filter panel and set any filter.
Expected Result	Still no results; message remains “No cars found”, app does not crash.
Actual Result	Still no results; message remains “No cars found”.
Verified (Yes/No)	Yes

Table 83-apply filter without any search result

Field	Description
Test ID	Filter_009
Prerequisite	Several filtered cars exist.
Test Procedure	Apply filter, then scroll down through results.
Expected Result	List scrolls smoothly; no data duplication or missing items.
Actual Result	List scrolls smoothly; no data duplication or missing items.
Verified (Yes/No)	Yes

Table 84 -scroll filtered result set

Field	Description
Test ID	Filter_010
Prerequisite	Some cars manual, some automatic.
Test Procedure	Choose Automatic in transmission filter.
Expected Result	Only automatic cars appear.
Actual Result	Only automatic cars appear.
Verified (Yes/No)	Yes

Table 85 -filter by transmission type

Field	Description
Test ID	Filter_011
Prerequisite	Mixed fuel types in results.
Test Procedure	Choose Electric in fuel type filter.
Expected Result	Only electric cars are displayed.
Actual Result	Only electric cars are displayed.
Verified (Yes/No)	Yes

Table 86 -filter by fuel type

Field	Description
Test ID	Filter_012
Prerequisite	Search results already loaded.
Test Procedure	Open Filters, change nothing, tap Apply.
Expected Result	Same results as before, no change in list.
Actual Result	Same results as before, no change in list.
Verified (Yes/No)	Yes

Table 87 -filter with no criteria selected

5.1.2.3 View car details Test Cases

Field	Description
Test ID	CarDetails_001
Prerequisite	Car appears in search results.
Test Procedure	Tap on a car card in search results.
Expected Result	Car details page opens showing images, price, specs, owner info.
Actual Result	Car details page opens showing images, price, specs, owner info.
Verified (Yes/No)	Yes

Table 88-open car details successfully

Field	Description
Test ID	CarDetails_002
Prerequisite	Car has multiple images.
Test Procedure	Swipe left/right through the car images.
Expected Result	User can view all images smoothly without errors.
Actual Result	User can view all images smoothly without errors.
Verified (Yes/No)	Yes

Table 89-swipe through car image

Field	Description
Test ID	CarDetails_003
Prerequisite	Fullscreen image viewer is supported.
Test Procedure	Tap on any image.
Expected Result	Image opens in fullscreen with zoom controls.
Actual Result	Image opens in fullscreen with zoom controls.
Verified (Yes/No)	Yes

Table 90-tap image to open full screen view

Field	Description
Test ID	CarDetails_004
Prerequisite	Car has no images uploaded.
Test Procedure	Open details page.
Expected Result	Placeholder image or “No images available” shown.
Actual Result	Placeholder image or “No images available” shown.
Verified (Yes/No)	Yes

Table 91 -card with missing image

Field	Description
Test ID	CarDetails_005
Prerequisite	Specs stored in database.
Test Procedure	Scroll down to specifications section.
Expected Result	Display correct year, model, transmission, fuel type, mileage, etc.
Actual Result	Display correct year, model, transmission, fuel type, mileage, etc.
Verified (Yes/No)	Yes

Table 92 -view cars specification details

Field	Description
Test ID	CarDetails_006
Prerequisite	Car has a price value in database.
Test Procedure	View car details.
Expected Result	Price shown matches database value.
Actual Result	Price shown matches database value.
Verified (Yes/No)	Yes

Table 93-display correct price per day

Field	Description
Test ID	CarDetails_007
Prerequisite	Availability data exists.
Test Procedure	Open calendar in details page.
Expected Result	Unavailable dates are marked; user cannot select them.
Actual Result	Unavailable dates are marked; user cannot select them.
Verified (Yes/No)	Yes

Table 94 -check availability calendar

Field	Description
Test ID	CarDetails_008
Prerequisite	Owner profile linked.
Test Procedure	Scroll to owner information section.
Expected Result	Shows owner name, rating, and response rate.
Actual Result	Shows owner name, rating, and response rate.
Verified (Yes/No)	Yes

Table 95 -display owner information

Field	Description
Test ID	CarDetails_09
Prerequisite	No reviews submitted for car.
Test Procedure	Scroll to reviews section.
Expected Result	Message: “No reviews yet.”
Actual Result	Message: “No reviews yet.”
Verified (Yes/No)	Yes

Table 96 -car with no reviews

Field	Description
Test ID	CarDetails_010
Prerequisite	Device disconnected.
Test Procedure	Tap on a car while offline.
Expected Result	Error: “No internet connection.”
Actual Result	Error: “No internet connection.”
Verified (Yes/No)	Yes

Table 97 -attempt to view details with no Internet

Field	Description
Test ID	CarDetails_011
Prerequisite	Owner deleted or deactivated car.
Test Procedure	Tap on a car that is no longer available.
Expected Result	Error: “This car is no longer available.”
Actual Result	Error: “This car is no longer available.”
Verified (Yes/No)	Yes

Table 98 -attempt to view car removed by owner

5.1.2.4 Create Booking Test Cases

Field	Description
Test ID	Booking_001
Prerequisite	Renter logged in; car available for selected dates; renter verified if required.
Test Procedure	From car details, select valid start and end date within availability and tap “Book Now”.
Expected Result	Booking is created with status “Pending” or “Awaiting payment/approval” and confirmation screen displayed.
Actual Result	Booking is created with status “Pending” and confirmation screen displayed.
Verified (Yes/No)	Yes

Table 99 -create booking with validate and available car

Field	Description
Test ID	Booking_002
Prerequisite	Car details screen open.
Test Procedure	Tap “Book Now” without choosing dates.
Expected Result	Validation error: “Please select start and end date.” Booking not created.
Actual Result	Validation error: “Please select start and end date.” Booking not created.
Verified (Yes/No)	Yes

Table 100 -attempt to create booking without selecting dates

Field	Description
Test ID	Booking_003
Prerequisite	None.
Test Procedure	Select end date earlier than start date and press Book.
Expected Result	Error: “Invalid date range.” Booking not created.
Actual Result	Error: “Invalid date range.” Booking not created.
Verified (Yes/No)	Yes

Table 101-start date after end date

Field	Description
Test ID	Booking_004
Prerequisite	Car already has confirmed booking for part of selected period.
Test Procedure	Select dates that overlap with an existing confirmed booking, then press Book.
Expected Result	Error: “Car is not available for selected dates.”
Actual Result	Error: “Car is not available for selected dates.”
Verified (Yes/No)	Yes

Table 102- booking overlaps with existing confirmed booking for same car

Field	Description
Test ID	Booking_005
Prerequisite	Renter has already a pending or confirmed booking for this car in the same date range.
Test Procedure	Create another booking request for same car & overlapping dates.
Expected Result	System blocks duplicate booking with message: “You already have a booking for this period.”
Actual Result	System blocks duplicate booking with message: “You already have a booking for this period.”
Verified (Yes/No)	Yes

Table 103 -renter tries to book same card twice for overlapping periods

Field	Description
Test ID	Booking_006
Prerequisite	Car flagged as “cannot be rebooked now”.
Test Procedure	Attempt to create booking for that car for any date.
Expected Result	Error: “This car cannot be rebooked now,” booking is not created.
Actual Result	Error: “This car cannot be rebooked now,” booking is not created.
Verified (Yes/No)	Yes

Table 104 -booking when car is marked as not rebookable

Field	Description
Test ID	Booking_007
Prerequisite	Minimum rental policy exists (e.g., 2 days).
Test Procedure	Select rental period shorter than minimum (e.g., 1 day) and attempt booking.
Expected Result	Error: “Minimum rental period is 2 days.”
Actual Result	Error: “Minimum rental period is 2 days.”
Verified (Yes/No)	Yes

Table 105 -booking with minimum rental period violation

Field	Description
Test ID	Booking_008
Prerequisite	Maximum rental period defined (e.g., 30 days).
Test Procedure	Select rental period longer than allowed and press Book.
Expected Result	Error: “Maximum rental period is 30 days.”
Actual Result	Error: “Maximum rental period is 30 days.”
Verified (Yes/No)	Yes

Table 106 -booking with maximum rental period violation

Field	Description
Test ID	Booking_009
Prerequisite	User logged out.
Test Procedure	From car details, select dates and tap Book Now.
Expected Result	System redirects to login/sign-up with message: “Please log in to continue booking.”
Actual Result	System redirects to login/sign-up with message: “Please log in to continue booking.”
Verified (Yes/No)	Yes

Table 107 -attempt booking while not logged in

Field	Description
Test ID	Booking_010
Prerequisite	User logged in but identity not verified.
Test Procedure	Select valid dates and attempt to book.
Expected Result	Booking blocked with message: “You must verify your identity before booking,” redirect to verification screen.
Actual Result	Booking blocked with message: “You must verify your identity before booking,” redirect to verification screen.
Verified (Yes/No)	Yes

Table 108 -attempt booking with unverified identity

Field	Description
Test ID	Booking_011
Prerequisite	System supports “booking then pay” flow.
Test Procedure	Select dates and press Book; do not pay yet.
Expected Result	Booking created with status “Awaiting Payment”; user is taken to payment screen.
Actual Result	Booking created with status “Awaiting Payment”; user is taken to payment screen.
Verified (Yes/No)	Yes

Table 109 -booking created, but payment required

Field	Description
Test ID	Booking_012
Prerequisite	Start booking with valid data, then disable internet quickly.
Test Procedure	Tap Book Now, then lose connection before server responds.
Expected Result	Error: “Network error / Cannot complete booking.” No booking stored or booking is only created if server confirms, never in partial/inconsistent state.
Actual Result	Error: “Network error / Cannot complete booking.” No partial/inconsistent booking stored.
Verified (Yes/No)	Yes

Table 110-network loss after pressing book

Field	Description
Test ID	Booking_013
Prerequisite	Same request triggered twice quickly.
Test Procedure	Double-tap Book Now rapidly with same data.
Expected Result	System prevents duplicate bookings; only one booking record is created.
Actual Result	System prevents duplicate bookings; only one booking record is created.
Verified (Yes/No)	Yes

Table 111-booking created twice by double tapping book now

5.1.2.5 Cancel Booking Test Cases

Field	Description
Test ID	Cancel_001
Prerequisite	Booking exists with status “Pending”.
Test Procedure	Open booking → tap Cancel Booking.
Expected Result	Booking is cancelled; status becomes “Cancelled by Renter”.
Actual Result	Booking is cancelled; status becomes “Cancelled by Renter”.
Verified	Yes

Table 112- cancel pending booking

Field	Description
Test ID	Cancel_002
Prerequisite	Booking is confirmed; cancellation allowed before rental start.
Test Procedure	Tap Cancel Booking before start date.
Expected Result	Booking cancelled successfully; refund (if applicable) processed.
Actual Result	Booking cancelled successfully; refund processed.
Verified	Yes

Table 113-cancel confirmed booking before start date

Field	Description
Test ID	Cancel_003
Prerequisite	It's the rental start date.
Test Procedure	Try canceling booking.
Expected Result	Error: "Booking cannot be cancelled after start date."
Actual Result	Error: "Booking cannot be cancelled after start date."
Verified	Yes

Table 114-cancel booking on start date

Field	Description
Test ID	Cancel_004
Prerequisite	Booking already has status "Cancelled".
Test Procedure	Tap Cancel again.
Expected Result	Message: "This booking is already cancelled."
Actual Result	Message: "This booking is already cancelled."
Verified	Yes

Table 115-cancel already canceled booking

Field	Description
Test ID	Cancel_005
Prerequisite	Booking completed or expired.
Test Procedure	Try cancelling.
Expected Result	Error: "Completed bookings cannot be cancelled."
Actual Result	Error: "Completed bookings cannot be cancelled."
Verified	Yes

Table 116-attempt to cancel complete booking

Field	Description
Test ID	Cancel_006
Prerequisite	Booking available to cancel.
Test Procedure	Turn off internet → tap Cancel.
Expected Result	Error: “No internet connection.”
Actual Result	Error: “No internet connection.”
Verified	Yes

Table 117- cancel while off-line

Field	Description
Test ID	Cancel_007
Prerequisite	Server slow.
Test Procedure	Tap cancel during delay.
Expected Result	Error: “Unable to process cancellation. Try again.”
Actual Result	Error: “Unable to process cancellation. Try again.”
Verified	Yes

Table 118-cancel booking during server time out

Field	Description
Test ID	Cancel_008
Prerequisite	Cancellation policy includes refund rules.
Test Procedure	Cancel a booking that qualifies for partial refund.
Expected Result	Refund breakdown shown correctly (amount refunded, fees kept).
Actual Result	Refund breakdown shown correctly.
Verified	Yes

Table 119-refund calculation display after cancel

Field	Description
Test ID	Cancel_009
Prerequisite	Booking includes insurance.
Test Procedure	Tap cancel.
Expected Result	System applies correct refund on rental + insurance based on policy.
Actual Result	System applies correct refund on rental + insurance.
Verified	Yes

Table 120-cancel booking with insurance Add on

Field	Description
Test ID	Cancel_010
Prerequisite	Booking available to cancel.
Test Procedure	Double-tap cancel quickly.
Expected Result	Only one cancellation request is processed; no duplicate actions.
Actual Result	Only one cancellation processed.
Verified	Yes

Table 121-double tap cancel button

5.1.2.6 Booking History Test Cases

Field	Description
Test ID	History_001
Prerequisite	User has completed bookings.
Test Procedure	Open Booking History tab.
Expected Result	List of completed bookings shown with car, dates, price.
Actual Result	Completed bookings shown clearly.
Verified	Yes

Table 122-view completed bookings

Field	Description
Test ID	History_002
Prerequisite	User has future bookings.
Test Procedure	Tap Upcoming section.
Expected Result	Shows all future bookings with status and dates.
Actual Result	Shows all future bookings with correct info.
Verified	Yes

Table 123-view upcoming bookings

Field	Description
Test ID	History_003
Prerequisite	User has never booked.
Test Procedure	Open booking history.
Expected Result	Message: “No bookings found.”
Actual Result	Message: “No bookings found.”
Verified	Yes

Table 124-no booking history

Field	Description
Test ID	History_004
Prerequisite	Booking exists.
Test Procedure	Tap a booking card.
Expected Result	Booking details open with correct info.
Actual Result	Booking details page opens.
Verified	Yes

Table 125-tap booking to view detail details

Field	Description
Test ID	History_005
Prerequisite	20+ bookings exist.
Test Procedure	Scroll up/down history list.
Expected Result	Smooth scrolling: items load correctly.
Actual Result	Smooth scrolling: items load correctly.
Verified	Yes

Table 126-scroll through long history list

Field	Description
Test ID	History_006
Prerequisite	User has multiple bookings.
Test Procedure	Filter history by month/year.
Expected Result	Correct bookings displayed for that period.
Actual Result	Correct filtered list displayed.
Verified	Yes

Table 127-filtering booking history by date

Field	Description
Test ID	History_007
Prerequisite	Temporarily disconnect internet.
Test Procedure	Open history → fail → reconnect → retry.
Expected Result	History reloads successfully.
Actual Result	History reloads successfully.
Verified	Yes

Table 128 -retry loading history after network failure

Field	Description
Test ID	History_008
Prerequisite	User has many bookings (multiple pages).
Test Procedure	Scroll to bottom to load next page.
Expected Result	Additional bookings load correctly without duplication.
Actual Result	Additional bookings load correctly without duplication.
Verified	Yes

Table 129 -booking history paginated

Field	Description
Test ID	History_009
Prerequisite	Multiple bookings with different cars.
Test Procedure	Type a car name into history search bar.
Expected Result	Show only bookings that match typed name.
Actual Result	Shows only matching bookings.
Verified	Yes

Table 130 -booking history research by Car name

Field	Description
Test ID	History_010
Prerequisite	At least one cancelled booking exists.
Test Procedure	Apply “Cancelled” filter.
Expected Result	Only cancelled bookings appear.
Actual Result	Only cancelled bookings appear.
Verified	Yes

Table 131 -view canceled bookings separately

5.1.2.7 Ratings & Reviews Test Cases

Field	Description
Test ID	Rating_001
Prerequisite	Booking completed; rating pending.
Test Procedure	Submit rating (e.g., 5 stars) and comment.
Expected Result	Rating saved; appears under car reviews.
Actual Result	Rating saved; appears correctly.
Verified	Yes

Table 132-submit Rating after completed booking

Field	Description
Test ID	Rating_002
Prerequisite	Booking completed.
Test Procedure	Submit only star rating.
Expected Result	Rating accepted.
Actual Result	Rating accepted.
Verified	Yes

Table 133-submit Rating without comment

Field	Description
Test ID	Rating_003
Prerequisite	Active booking not yet completed.
Test Procedure	Tap “Rate now”.
Expected Result	Error: “You can rate only after completing the booking.”
Actual Result	Error shown correctly.
Verified	Yes

Table 134-try to rate before booking complete

Field	Description
Test ID	Rating_004
Prerequisite	Renter already submitted a review.
Test Procedure	Tap Edit Review → change rating or comment → save.
Expected Result	Updated review saved and displayed in the reviews list.
Actual Result	Updated review saved and displayed correctly.
Verified	Yes

Table 135-edit rating after submission

Field	Description
Test ID	Rating_005
Prerequisite	Renter has a review for that car.
Test Procedure	Tap Delete Review.
Expected Result	Review removed from system; no longer appears under ratings.
Actual Result	Review removed and not shown.
Verified	Yes

Table 136- delete review

Field	Description
Test ID	Rating_006
Prerequisite	Max review length (e.g., 500 chars).
Test Procedure	Enter comment at maximum allowed length → submit.
Expected Result	Review accepted, no errors.
Actual Result	Review accepted, no errors.
Verified	Yes

Table 137-submit rating with maximum text length

Field	Description
Test ID	Rating_007
Prerequisite	Review field cannot be empty.
Test Procedure	Enter spaces only and submit.
Expected Result	Error: “Please write a valid review.”
Actual Result	Error: “Please write a valid review.”
Verified	Yes

Table 138-attempt to submit review with only spaces

5.1.2.8 Profile Management Test Cases

Field	Description
Test ID	Profile_001
Prerequisite	User logged in.
Test Procedure	Edit name/phone and save.
Expected Result	Profile updated successfully.
Actual Result	Profile updated successfully.
Verified	Yes

Table 139-update profile information

Field	Description
Test ID	Profile_002
Prerequisite	User logged in.
Test Procedure	Upload new profile picture from gallery.
Expected Result	New profile picture saved and displayed.
Actual Result	New profile picture saved and displayed.
Verified	Yes

Table 140-change profile picture successfully

Field	Description
Test ID	Profile_003
Prerequisite	Profile picture exists.
Test Procedure	Tap Remove Photo.
Expected Result	Profile photo removed and replaced with default avatar.
Actual Result	Photo removed and replaced with default avatar.
Verified	Yes

Table 141- Remove profile picture

Field	Description
Test ID	Profile_004
Prerequisite	User knows current password.
Test Procedure	Enter correct current password → new password → confirm new password.
Expected Result	Password updated successfully.
Actual Result	Password updated successfully.
Verified	Yes

Table 142-change password successfully

Field	Description
Test ID	Profile_005
Prerequisite	Wrong current password provided.
Test Procedure	Enter incorrect current password.
Expected Result	Error: “Current password is incorrect.”
Actual Result	Error displayed correctly.
Verified	Yes

Table 143-incorrect current password when changing password

Field	Description
Test ID	Profile_006
Prerequisite	OTP system enabled.
Test Procedure	Enter new number → receive OTP → enter OTP → submit.
Expected Result	Number updated successfully.
Actual Result	Number updated successfully.
Verified	Yes

Table 144 -update phone number with OTP verification

5.1.2.9 Notifications Test Cases

Field	Description
Test ID	Notif_001
Prerequisite	Booking created.
Test Procedure	Wait for notification.
Expected Result	Notification appears with correct message.
Actual Result	Notification appears correctly.
Verified	Yes

Table 145-receive booking confirmation notification

Field	Description
Test ID	Notif_002
Prerequisite	Renter has a pending booking; owner approves it.
Test Procedure	Owner approves the renter's booking from owner side.
Expected Result	Renter receives a notification: "Your booking has been approved" with car name and dates.
Actual Result	Renter receives a notification: "Your booking has been approved" with car name and dates.
Verified (Yes/No)	Yes

Table 146-receive booking approval notification

Field	Description
Test ID	Notif_003
Prerequisite	Renter has a pending booking; owner rejects it.
Test Procedure	Owner rejects the booking request with a reason.
Expected Result	Renter receives a notification: "Your booking was rejected" and, if configured, shows the reason.
Actual Result	Renter receives a notification: "Your booking was rejected" with the reason.
Verified (Yes/No)	Yes

Table 147-receive booking rejection notification

Field	Description
Test ID	Notif_004
Prerequisite	Booking has status "Awaiting Payment" with a deadline.
Test Procedure	Wait until reminder time (e.g., 1 hour before payment deadline).
Expected Result	Renter receives notification reminding them to complete payment before deadline.
Actual Result	Renter receives notification reminding them to complete payment before deadline.
Verified (Yes/No)	Yes

Table 148-payment reminder notification

Field	Description
Test ID	Notif_005
Prerequisite	Confirmed booking starting soon (e.g., tomorrow).
Test Procedure	Wait until configured reminder time (e.g., 24 hours before start).
Expected Result	Renter receives notification: “Your trip starts tomorrow” with car and pickup info.
Actual Result	Renter receives notification: “Your trip starts tomorrow” with car and pickup info.
Verified (Yes/No)	Yes

Table 149-upcoming trip reminder

Field	Description
Test ID	Notif_006
Prerequisite	Confirmed booking exists.
Test Procedure	Owner or admin cancels the booking from their side.
Expected Result	Renter receives notification that the booking has been cancelled plus refund status if any.
Actual Result	Renter receives notification that the booking has been cancelled plus refund status.
Verified (Yes/No)	Yes

Table 150-booking cancellation notification

Field	Description
Test ID	Notif_007
Prerequisite	Chat enabled between renter and owner.
Test Procedure	Owner sends a new message to renter.
Expected Result	Renter receives notification: “New message from [Owner Name]” and tapping it opens the chat.
Actual Result	Renter receives notification: “New message from [Owner Name]” and tapping it opens the chat.
Verified (Yes/No)	Yes

Table 151-new message/chat notification from owner

Field	Description
Test ID	Notif_008
Prerequisite	Any booking-related notification exists (confirmation, approval...).
Test Procedure	Tap on the notification from notification bar.
Expected Result	App opens on the correct booking details screen related to that notification.
Actual Result	App opens on the correct booking details screen related to that notification.
Verified (Yes/No)	Yes

Table 152-tap on booking notification opens correct screen

Field	Description
Test ID	Notif_009
Prerequisite	App is open (foreground).
Test Procedure	Trigger any notification event (e.g., owner sends message).
Expected Result	In-app banner/message appears (instead of OS push) without breaking current screen.
Actual Result	In-app banner/message appears without breaking current screen.
Verified (Yes/No)	Yes

Table 153-notification when app in foreground

Field	Description
Test ID	Notif_010
Prerequisite	App is installed and user logged in; app is in background.
Test Procedure	Trigger booking approval or message while app is backgrounded.
Expected Result	System push notification appears in device notification bar.
Actual Result	System push notification appears in device notification bar.
Verified (Yes/No)	Yes

Table 154-notification when app is in background

Field	Description
Test ID	Notif_011
Prerequisite	User has disabled notifications for the app in OS settings.
Test Procedure	Trigger a notification (booking approved/message).
Expected Result	No OS push appears; if app design supports it, notification may still appear inside the app only when opened.
Actual Result	No OS push appears; notification visible only inside the app if supported.
Verified (Yes/No)	Yes

Table 155-notification with disabled push permission

5.1.3 Owner

5.1.3.1 Add Car Listings Test Cases

Field	Description
Test ID	AddCar_001
Prerequisite	Owner is logged in; all required fields known.
Test Procedure	Fill in all fields correctly → upload photos → save car.
Expected Result	Car listing created successfully; status becomes “Pending Verification” or “Active”.
Actual Result	Car listing created successfully; status becomes “Pending Verification”.
Verified	Yes

Table 156 -Add Car with All Valid Information

Field	Description
Test ID	AddCar_002
Prerequisite	Required fields include model, year, price per day.
Test Procedure	Leave required fields empty → attempt to save.
Expected Result	Validation errors highlight missing fields.
Actual Result	Validation errors highlight missing fields.
Verified	Yes

Table 157- Add Car Without Required Fields

Field	Description
Test ID	AddCar_003
Prerequisite	System accepts only JPG, PNG.
Test Procedure	Attempt to upload file with invalid extension (e.g., .pdf).
Expected Result	Error: “Unsupported file type.”
Actual Result	Error: “Unsupported file type.”
Verified	Yes

Table 158- Upload Invalid Image Format

Field	Description
Test ID	AddCar_004
Prerequisite	Photos required.
Test Procedure	Fill data but skip image upload.
Expected Result	Error: “Please upload at least one image.”
Actual Result	Error: “Please upload at least one image.”
Verified	Yes

Table 159- Add Car Without Photos

Field	Description
Test ID	AddCar_005
Prerequisite	Maximum 10 images allowed (example).
Test Procedure	Upload 10 images → save.
Expected Result	Listing accepts all photos and saves correctly.
Actual Result	Listing accepts all photos and saves correctly.
Verified	Yes

Table 160- Add Car with Maximum Photo Limit

Field	Description
Test ID	AddCar_006
Prerequisite	Limit is 10 photos.
Test Procedure	Try uploading 11 images.
Expected Result	Error: “Maximum photos allowed: 10.”
Actual Result	Error displayed.
Verified	Yes

Table 161- Exceed Photo Limit

Field	Description
Test ID	AddCar_007
Prerequisite	Car year must be ≤ current year.
Test Procedure	Enter next year as car model year.
Expected Result	Error: “Invalid car manufacturing year.”
Actual Result	Error: “Invalid car manufacturing year.”
Verified	Yes

Table 162-Enter Invalid Year (Future Year)

Field	Description
Test ID	AddCar_008
Prerequisite	Minimum daily price rule exists.
Test Procedure	Enter too-low price (e.g., 1 SAR).
Expected Result	Error: “Price per day is too low.”
Actual Result	Error displayed.
Verified	Yes

Table 163- Enter Price Below Minimum Allowed

Field	Description
Test ID	AddCar_009
Prerequisite	Description character limit (e.g., 500 characters).
Test Procedure	Enter description exceeding limit.
Expected Result	System prevents typing or shows an error.
Actual Result	System prevents exceeding length.
Verified	Yes

Table 164-Enter Extremely Long Description

Field	Description
Test ID	AddCar_010
Prerequisite	No internet.
Test Procedure	Fill all data → tap save.
Expected Result	Error: “No internet connection.”
Actual Result	Error displayed.
Verified	Yes

Table 165 -Add Car While Offline

Field	Description
Test ID	AddCar_011
Prerequisite	Slow connection simulated.
Test Procedure	Upload large images.
Expected Result	Loading indicator shown; upload completes without corruption.
Actual Result	Loading indicator shown; upload completes correctly.
Verified	Yes

Table 166 -Slow Network While Uploading Images

Field	Description
Test ID	AddCar_012
Prerequisite	VIN or plate already exists in database.
Test Procedure	Add car with same VIN/plate.
Expected Result	Error: “This car is already registered.”
Actual Result	Error appears correctly.
Verified	Yes

Table 167 - Duplicate Car Detection (Same VIN/Plate)

Field	Description
Test ID	AddCar_013
Prerequisite	Plate format validation exists.
Test Procedure	Enter illegal plate number format.
Expected Result	Error: “Invalid license plate format.”
Actual Result	Error displayed.
Verified	Yes

Table 168 - Enter Invalid License Plate Format

Field	Description
Test ID	AddCar_014
Prerequisite	Optional attributes exist.
Test Procedure	Add car with several optional features selected.
Expected Result	Listing saves all optional features; displayed in car details.
Actual Result	Listing saves and displays optional features.
Verified	Yes

Table 169 - Add Car with Optional Features (e.g., GPS, Sunroof)

5.1.3.2 Edit Car Listing Test Cases

Field	Description
Test ID	EditCar_001
Prerequisite	Owner has at least one active car listing.
Test Procedure	Open car listing → change price per day to a valid value → save changes.
Expected Result	New price is saved and shown correctly in car details and search results.
Actual Result	New price is saved and shown correctly in car details and search results.
Verified (Yes/No)	Yes

Table 170 - Edit Car Price Successfully

Field	Description
Test ID	EditCar_002
Prerequisite	Car listing exists.
Test Procedure	Update car description text and save.
Expected Result	Description updated and displayed correctly on the car details page.
Actual Result	Description updated and displayed correctly on the car details page.
Verified	Yes

Table 171- Edit Car Description Successfully

Field	Description
Test ID	EditCar_003
Prerequisite	Existing car listing.
Test Procedure	Delete required value (e.g., price or model) and tap Save.
Expected Result	Validation error: “This field is required” and changes are not saved.
Actual Result	Validation error: “This field is required” and changes are not saved.
Verified	Yes

Table 172 - Attempt to Save Empty Required Field While Editing

Field	Description
Test ID	EditCar_004
Prerequisite	Car listing exists.
Test Procedure	Change year to a year greater than current year and save.
Expected Result	Error: “Invalid car manufacturing year.”
Actual Result	Error: “Invalid car manufacturing year.”
Verified	Yes

Table 173 - Edit Car Year to Invalid Future Year

Field	Description
Test ID	EditCar_005
Prerequisite	Listing exists; price should be numeric.
Test Procedure	Enter letters instead of numbers in the price field and save.
Expected Result	Error: “Please enter a valid numeric price.”
Actual Result	Error: “Please enter a valid numeric price.”
Verified	Yes

Table 174 - Edit with Invalid Price Format (Letters)

Field	Description
Test ID	EditCar_006
Prerequisite	Minimum price rule exists.
Test Procedure	Change price to below minimum and save.
Expected Result	Error: "Price per day is too low."
Actual Result	Error: "Price per day is too low."
Verified	Yes

Table 175 - Reduce Price Below Minimum Allowed

Field	Description
Test ID	EditCar_007
Prerequisite	Max price configured.
Test Procedure	Enter very high price exceeding limit and save.
Expected Result	Error: "Price per day exceeds the allowed limit."
Actual Result	Error: "Price per day exceeds the allowed limit."
Verified	Yes

Table 176 - Increase Price Above Maximum Allowed

Field	Description
Test ID	EditCar_008
Prerequisite	Car listing with existing photos.
Test Procedure	Add one more valid photo and save listing.
Expected Result	New photo is uploaded and appears in car details gallery.
Actual Result	New photo is uploaded and appears in car details gallery.
Verified	Yes

Table 177 - Change Car Photos (Add New Photo)

Field	Description
Test ID	EditCar_009
Prerequisite	Listing with multiple images.
Test Procedure	Delete one of the photos and save.
Expected Result	Deleted photos are removed from gallery; other photos remain.
Actual Result	Deleted photos are removed; other photos remain.
Verified	Yes

Table 178 - Remove One of the Existing Photos

Field	Description
Test ID	EditCar_010
Prerequisite	At least one image required.
Test Procedure	Remove all photos and attempt to save.
Expected Result	Error: “At least one image is required.”
Actual Result	Error: “At least one image is required.”
Verified	Yes

Table 179 - Try to Remove All Photos (Photos Required)

Field	Description
Test ID	EditCar_011
Prerequisite	Listing exists.
Test Procedure	Change city from “Dammam” to “Riyadh” and save.
Expected Result	Cars appear under new city in search; location updated in detail.
Actual Result	Cars appear under new city in search; location updated in detail.
Verified	Yes

Table 180 - Edit Car Location (City)

Field	Description
Test ID	EditCar_012
Prerequisite	Car type field exists.
Test Procedure	Change car type and save.
Expected Result	Car now appears under the new type in filters and details.
Actual Result	Car appears under the new type in filters and details.
Verified	Yes

Table 181 - Edit Car Type/Category (e.g., Sedan to SUV)

Field	Description
Test ID	EditCar_013
Prerequisite	Admin modifies or locks the listing at same time (simulated conflict).
Test Procedure	Owner edits and saves listing during that period.
Expected Result	System either blocks saving with message “Listing updated elsewhere. Please reload” or merges correctly based on design.
Actual Result	System warns and forces owner to reload before saving.
Verified	Yes

Table 182 - Edit While Another User (Admin) Is Modifying Listing (Conflict)

Field	Description
Test ID	EditCar_014
Prerequisite	Device offline.
Test Procedure	Change some fields and tap Save.
Expected Result	Error: “No internet connection. Changes not saved.”
Actual Result	Error: “No internet connection. Changes not saved.”
Verified	Yes

Table 183- Edit Listing While Offline

5.1.3.3 Deactivate / Delete Car

Field	Description
Test ID	DeactivateCar_001
Prerequisite	Car listing is currently active.
Test Procedure	Tap Deactivate Listing from the car management page.
Expected Result	Car status changes to “Inactive”; car no longer appears in renter searches.
Actual Result	Car status changed to “Inactive”; hidden from search.
Verified	Yes

Table 184 -Deactivate Car Listing Successfully

Field	Description
Test ID	DeactivateCar_002
Prerequisite	Car listing is inactive.
Test Procedure	Tap Activate Listing to restore availability.
Expected Result	Car status becomes “Active” and visible to renters in search.
Actual Result	Car status becomes “Active” and visible.
Verified	Yes

Table 185 -Reactivate Car Listing Successfully

Field	Description
Test ID	DeactivateCar_003
Prerequisite	Car has an active or ongoing booking.
Test Procedure	Attempt to deactivate listing.
Expected Result	Error: “Car cannot be deactivated while a booking is active.”
Actual Result	Error: “Car cannot be deactivated while a booking is active.”
Verified	Yes

Table 186 - Deactivate Car with Ongoing Booking (Not Allowed)

Field	Description
Test ID	DeactivateCar_004
Prerequisite	Car has future bookings but not yet started.
Test Procedure	Tap deactivate.
Expected Result	Warning: “Car has upcoming bookings. Deactivation will not affect existing reservations.” → Allow or cancel action.
Actual Result	Warning displayed and deactivation allowed safely.
Verified	Yes

Table 187 - Deactivate Car with Upcoming Bookings (Warning Required)

Field	Description
Test ID	DeactivateCar_005
Prerequisite	Car has no past or future bookings.
Test Procedure	Tap Delete Listing → confirm deletion.
Expected Result	Car listing is removed permanently from the system.
Actual Result	Car listing removed permanently.
Verified	Yes

Table 188 - Delete Car Listing Permanently (No Bookings)

Field	Description
Test ID	DeactivateCar_006
Prerequisite	Car has past bookings in history.
Test Procedure	Tap delete.
Expected Result	Error: “This car cannot be deleted because it has booking history.”
Actual Result	Error displayed.
Verified	Yes

Table 189 - Attempt to Delete Car with Booking History (Not Allowed)

5.1.3.4 Update Availability Calendar Test Cases

Field	Description
Test ID	Availability_001
Prerequisite	Car listing is active; no booking on selected day.
Test Procedure	Tap a specific date → mark as unavailable → save.
Expected Result	Selected date changes to “Unavailable” and renters cannot book it.
Actual Result	Selected date marked unavailable successfully.
Verified	Yes

Table 190 - Block a Single Day Successfully

Field	Description
Test ID	Availability_002
Prerequisite	Car listing active.
Test Procedure	Select a range (e.g., 3 days) → mark unavailable → save.
Expected Result	All days in range marked unavailable.
Actual Result	All days in range marked unavailable.
Verified	Yes

Table 191 - Block a Range of Days

Field	Description
Test ID	Availability_003
Prerequisite	At least one blocked day exists.
Test Procedure	Select blocked date → mark available → save.
Expected Result	Day becomes available; renters can now book.
Actual Result	Day becomes available.
Verified	Yes

Table 192 - Unblock a Previously Blocked Day

Field	Description
Test ID	Availability_004
Prerequisite	Booking confirmed on selected date.
Test Procedure	Try to block that date.
Expected Result	Error: “Date cannot be blocked due to an existing booking.”
Actual Result	Error displayed.
Verified	Yes

Table 193- Attempt to Block Date with Confirmed Booking (Not Allowed)

Field	Description
Test ID	Availability_005
Prerequisite	Example: booking on Day 2 of a 3-day range.
Test Procedure	Try blocking Day 1–3.
Expected Result	Error: “Selected range includes booked days.”
Actual Result	Error displayed correctly.
Verified	Yes

Table 194 - Attempt to Block Partial Range with Bookings in Middle

5.1.3.5 Approve or Reject Bookings Test Cases

Field	Description
Test ID	ManageBooking_001
Prerequisite	Booking exists with status Pending for owner's car.
Test Procedure	Owner opens booking details → taps Approve → confirms.
Expected Result	Booking status changes to Approved/Confirmed; renter receives approval notification; car dates are blocked in availability.
Actual Result	Booking status changes to Approved/Confirmed; renter is notified; dates blocked.
Verified (Yes/No)	Yes

Table 195 -Approve Pending Booking Successfully

Field	Description
Test ID	ManageBooking_002
Prerequisite	Booking in Pending status.
Test Procedure	Owner taps Reject → enters a reason → confirms.
Expected Result	Booking status changes to Rejected by Owner; renter receives rejection notification including reason.
Actual Result	Booking status changes to Rejected by Owner; renter notified with reason.
Verified	Yes

Table 196- Reject Pending Booking with Reason

Field	Description
Test ID	ManageBooking_003
Prerequisite	Booking pending; approval message is optional.
Test Procedure	Tap Approve and confirm without typing any message.
Expected Result	Booking approved successfully; default notification text sent to renter.
Actual Result	Booking approved successfully; default notification sent.
Verified	Yes

Table 197 - Approve Booking Without Entering Optional Message

Field	Description
Test ID	ManageBooking_004
Prerequisite	Booking status already Approved.
Test Procedure	Owner opens booking and taps Approve again.
Expected Result	System prevents duplicate action with message: "This booking is already approved."
Actual Result	System prevents duplicate action with message: "This booking is already approved."
Verified	Yes

Table 198 - Try to Approve Already Approved Booking

Field	Description
Test ID	ManageBooking_005
Prerequisite	Booking status is Rejected.
Test Procedure	Owner taps Reject again.
Expected Result	System shows: "This booking is already rejected." No status change.
Actual Result	System shows: "This booking is already rejected." No status change.
Verified	Yes

Table 199-Try to Reject Already Rejected Booking

Field	Description
Test ID	ManageBooking_006
Prerequisite	Booking is pending but rental start date is already in the past.
Test Procedure	Owner taps Approve.
Expected Result	Error: "This booking has expired and cannot be approved."
Actual Result	Error: "This booking has expired and cannot be approved."
Verified	Yes

Table 200 -Try to Approve Expired Booking (Start Date Passed)

Field	Description
Test ID	ManageBooking_007
Prerequisite	Another booking for same car is already approved for overlapping dates.
Test Procedure	Owner attempts to approve new overlapping booking.
Expected Result	System prevents approval and shows: "Car is already booked for these dates."
Actual Result	System prevents approval and shows: "Car is already booked for these dates."
Verified	Yes

Table 201 - Approve Booking That Overlaps Another Approved Booking

5.1.3.6 Owner Dashboard & Earnings Test Cases

Field	Description
Test ID	Earnings_001
Prerequisite	Owner has completed bookings with generated earnings.
Test Procedure	Open Owner Dashboard → Earnings tab.
Expected Result	Total earnings displayed accurately based on completed bookings.
Actual Result	Total earnings displayed accurately.
Verified	Yes

Table 202-View Total Earnings Successfully

Field	Description
Test ID	Earnings_002
Prerequisite	Bookings completed but funds still in “Pending release” state.
Test Procedure	Navigate to Pending Earnings section.
Expected Result	Shows pending earnings separate from available balance.
Actual Result	Pending earnings shown correctly.
Verified	Yes

Table 203 -View Pending Earnings (Awaiting Release)

Field	Description
Test ID	Earnings_003
Prerequisite	Owner has earnings across multiple months.
Test Procedure	Apply filter “Last 30 days”.
Expected Result	Dashboard updates and displays earnings for selected period only.
Actual Result	Dashboard updates correctly for the chosen period.
Verified	Yes

Table 204 -View Earnings for Specific Time Period (Filter)

Field	Description
Test ID	Earnings_004
Prerequisite	Owner has never completed a booking.
Test Procedure	Open earnings page.
Expected Result	Show message: “No earnings yet.”
Actual Result	“No earnings yet” message displayed.
Verified	Yes

Table 205-No Earnings Yet (New Owner)

Field	Description
Test ID	Earnings_005
Prerequisite	Owners have multiple cars with different earnings.
Test Procedure	View “Earnings by Car” chart/table.
Expected Result	Each car shows its own earnings total.
Actual Result	Earnings displayed per car correctly.
Verified	Yes

Table 206-Earnings Breakdown Per Car

5.1.3.7 Withdraw Earnings Test Cases

Field	Description
Test ID	Withdraw_001
Prerequisite	Owner has sufficient balance available for withdrawal.
Test Procedure	Enter withdrawal amount → tap Withdraw → confirm.
Expected Result	Withdrawal requests created; balance decreases accordingly.
Actual Result	Withdrawal request created and balance updated.
Verified	Yes

Table 207 -Successful Withdrawal Request

Field	Description
Test ID	Withdraw_002
Prerequisite	Available balance is lower than requested amount.
Test Procedure	Enter amount greater than balance.
Expected Result	Error: “Insufficient balance.”
Actual Result	Error displayed.
Verified	Yes

Table 208-Withdrawal Amount Exceeds Available Balance

Field	Description
Test ID	Withdraw_003
Prerequisite	Minimum withdrawal amount exists (example: 100 SAR).
Test Procedure	Enter amount below minimum.
Expected Result	Error: “Minimum withdrawal amount is 100 SAR.”
Actual Result	Error shown correctly.
Verified	Yes

Table 209 -Minimum Withdrawal Amount Validation

Field	Description
Test ID	Withdraw_004
Prerequisite	IBAN must follow correct pattern.
Test Procedure	Enter incorrect IBAN → attempt withdrawal.
Expected Result	Error: “Invalid IBAN format.”
Actual Result	Error displayed.
Verified	Yes

Table 210- Enter Invalid IBAN Format

Field	Description
Test ID	Withdraw_005
Prerequisite	Withdrawal requests require admin approval.
Test Procedure	Submit withdrawal request.
Expected Result	Status becomes “Pending Approval”; balance locked until approved.
Actual Result	Status becomes “Pending Approval”; balance locked.
Verified	Yes

Table 211 -Withdrawal Request Waiting for Admin Approval

5.1.3.8 Owner Notifications Test Cases

Field	Description
Test ID	OwnerNotif_001
Prerequisite	Renter submits a booking for owner’s car.
Test Procedure	Owner receives new notification.
Expected Result	Notification shows renter name, dates, and car info.
Actual Result	Notification displayed correctly.
Verified	Yes

Table 212 -Notification for New Booking Request

Field	Description
Test ID	OwnerNotif_002
Prerequisite	Owner approves booking.
Test Procedure	Booking status updates.
Expected Result	Owner sees confirmation notification: “Booking approved successfully.”
Actual Result	Notification displayed.
Verified	Yes

Table 213 -Notification After Booking Approval Confirmation

Field	Description
Test ID	OwnerNotif_003
Prerequisite	Active or upcoming booking exists.
Test Procedure	Renter cancels booking.
Expected Result	Owner notified: “Renter canceled the booking.”
Actual Result	Notification received correctly.
Verified	Yes

Table 214 -Notification When Renter Cancels Booking

Field	Description
Test ID	OwnerNotif_004
Prerequisite	Booking completed and earnings released.
Test Procedure	System releases earnings.
Expected Result	Owner gets notification of amount released.
Actual Result	Notification shown correctly.
Verified	Yes

Table 215-Notification for Earnings Released

5.1.4 Admin

5.1.4.1 Manage User Test Cases

Field	Description
Test ID	AdminUser_001
Prerequisite	Admin logged in; system has multiple users (renters and owners).
Test Procedure	Go to Admin Panel → Users.
Expected Result	List of all users is displayed with ID, name, role, status.
Actual Result	List of all users is displayed with ID, name, role, status.
Verified (Yes/No)	Yes

Table 216 -View List of All Users

Field	Description
Test ID	AdminUser_002
Prerequisite	A user with known email exists.
Test Procedure	In search bar, type exact email → press search.
Expected Result	Matching user appears in search result table.
Actual Result	Matching user appears in search result table.
Verified	Yes

Table 217 -Search User by Email

Field	Description
Test ID	AdminUser_003
Prerequisite	Multiple users whose names contain similar substring.
Test Procedure	Type partial name (e.g., “Na”) and search.
Expected Result	All users with matching substring in name are shown.
Actual Result	All users with matching substring in name are shown.
Verified	Yes

Table 218 -Search User by Partial Name

Field	Description
Test ID	AdminUser_004
Prerequisite	Users of multiple roles exist.
Test Procedure	Choose role filter “Owner”.
Expected Result	Only owner accounts appear in the list.
Actual Result	Only owner accounts appear in the list.
Verified	Yes

Table 219 - Filter Users by Role (Renter / Owner / Admin)

Field	Description
Test ID	AdminUser_005
Prerequisite	Users with various statuses.
Test Procedure	Apply filter “Suspended”.
Expected Result	Only suspended users are displayed.
Actual Result	Only suspended users are displayed.
Verified	Yes

Table 220 -Filter Users by Status (Active / Suspended / Pending)

Field	Description
Test ID	AdminUser_006
Prerequisite	User exists.
Test Procedure	Click on a user row → open details.
Expected Result	Full user details displayed (name, email, phone, role, verification status, created date, history).
Actual Result	Full user details displayed correctly.
Verified	Yes

Table 221 -View Detailed User Profile

Field	Description
Test ID	AdminUser_007
Prerequisite	User status = Active.
Test Procedure	Open user → click Suspend → confirm.
Expected Result	User status changes to Suspended; user cannot log in anymore.
Actual Result	User status changes to Suspended; users cannot log in.
Verified	Yes

Table 222 -Suspend Active User Successfully

Field	Description
Test ID	AdminUser_008
Prerequisite	User status = Suspended.
Test Procedure	Try to suspend again.
Expected Result	System shows: “User is already suspended.” No additional change.
Actual Result	System shows: “User is already suspended.”
Verified	Yes

Table 223 -Suspend Already Suspended User

Field	Description
Test ID	AdminUser_009
Prerequisite	User status = Suspended.
Test Procedure	Click Activate → confirm.
Expected Result	User status changes to Active; users can log in again.
Actual Result	User status changes to Active; users can log in.
Verified	Yes

Table 224 -Reactivate Suspended User Successfully

Field	Description
Test ID	AdminUser_010
Prerequisite	User has active or upcoming booking.
Test Procedure	Admin suspends user.
Expected Result	System shows warning: “User has active bookings”; depending on policy either block or allow with note; bookings handled according to rules.
Actual Result	Warning shown; suspension follows defined policy.
Verified	Yes

Table 225 - Suspend User While They Have Active Booking

5.1.4.2 Manage Cars Test Cases

Field	Description
Test ID	AdminCar_001
Prerequisite	One or more cars with status “Pending Verification”.
Test Procedure	Go to Admin Panel → Cars → Pending.
Expected Result	List of all unverified cars with owner, model, plate, submitted date.
Actual Result	List of all unverified cars appears correctly.
Verified	Yes

Table 226-View All Cars Awaiting Verification

Field	Description
Test ID	AdminCar_002
Prerequisite	Pending cars exists.
Test Procedure	Click on a car in pending list.
Expected Result	Admin sees full details: photos, documents, specs, owner info, car status.
Actual Result	Full car details displayed correctly.
Verified	Yes

Table 227 -Open Car Details for Review

Field	Description
Test ID	AdminCar_003
Prerequisite	Car is in Pending Verification.
Test Procedure	Admin reviews car → clicks Approve.
Expected Result	Car status changes to Active; owner receives approval notification; car appears in renter search.
Actual Result	Car status changes to Active; owner notified; search updated.
Verified	Yes

Table 228 -Approve Car Listing Successfully

Field	Description
Test ID	AdminCar_004
Prerequisite	Pending car listing.
Test Procedure	Click Reject → type reason (e.g., "Photos unclear") → confirm.
Expected Result	Car status becomes Rejected; owner notified with the rejection reason; car is not visible in search.
Actual Result	Status becomes Rejected; owner notification sent with reason.
Verified	Yes

Table 229-Reject Car Listing With Reason

Field	Description
Test ID	AdminCar_005
Prerequisite	System requires reason to reject.
Test Procedure	Click Reject → do not type reason → confirm.
Expected Result	Validation error: "Please enter a rejection reason."
Actual Result	Validation error shown.
Verified	Yes

Table 230 - Reject Without Reason When Reason Is Mandatory

Field	Description
Test ID	AdminCar_006
Prerequisite	Car status = Active.
Test Procedure	Admin opens car details → click Suspend.
Expected Result	Car status becomes Suspended by Admin; car disappears from renter search; owner notified.
Actual Result	Car status becomes Suspended; not shown to renters; owner notified.
Verified	Yes

Table 231-Suspend Active Car Listing (Policy Violation)

Field	Description
Test ID	AdminCar_007
Prerequisite	Car has no bookings.
Test Procedure	Click Delete → confirm.
Expected Result	Car is permanently deleted from system.
Actual Result	Car permanently deleted.
Verified	Yes

Table 232 -Delete Car Listing (No Booking History) by Admin

5.1.4.3 Manage Bookings Test Cases

Field	Description
Test ID	AdminBooking_001
Prerequisite	Multiple bookings exist.
Test Procedure	Go to Admin Panel → Bookings.
Expected Result	All bookings displayed with status, renter, owner, car, dates, amount.
Actual Result	All bookings displayed with correct details.
Verified	Yes

Table 233 -View All Bookings

Field	Description
Test ID	AdminBooking_002
Prerequisite	Booking ID known.
Test Procedure	Enter booking ID in search bar.
Expected Result	Matching booking appears.
Actual Result	Matching booking appears.
Verified	Yes

Table 234 -Search Booking by Booking ID

Field	Description
Test ID	AdminBooking_003
Prerequisite	Bookings in all statuses exist.
Test Procedure	Select filter “Approved”.
Expected Result	Only approved bookings displayed.
Actual Result	Only approved bookings displayed.
Verified	Yes

Table 235 -Filter Bookings by Status (Pending / Approved / Completed / Cancelled)

Field	Description
Test ID	AdminBooking_004
Prerequisite	Bookings spread across months.
Test Procedure	Enter date from X to Y → apply filter.
Expected Result	Only bookings within that date range appear.
Actual Result	Results correctly filtered.
Verified	Yes

Table 236 -Filter Bookings by Date Range

Field	Description
Test ID	AdminBooking_005
Prerequisite	Booking exists.
Test Procedure	Click a booking row.
Expected Result	Full booking details shown: renter, owner, price, dates, payment, messages, history.
Actual Result	Full details displayed.
Verified	Yes

Table 237 -Open Booking Details

5.1.4.4 Complaints & Report Test Cases

Field	Description
Test ID	AdminReport_001
Prerequisite	At least one complaint exists in the system.
Test Procedure	Go to Admin Panel → Complaints / Reports.
Expected Result	List of complaints displayed with ID, reporter, type, status, date.
Actual Result	List of complaints displayed with ID, reporter, type, status, date.
Verified (Yes/No)	Yes

Table 238 -View List of All Complaints

Field	Description
Test ID	AdminReport_002
Prerequisite	Complaints exist in different statuses.
Test Procedure	Apply status filter “Open”.
Expected Result	Only open complaints are displayed.
Actual Result	Only open complaints are displayed.
Verified	Yes

Table 239 -Filter Complaints by Status (Open / In Progress / Resolved)

Field	Description
Test ID	AdminReport_003
Prerequisite	Complaints with various types exist.
Test Procedure	Select type filter “Car”.
Expected Result	Only complaints related to cars are listed.
Actual Result	Only car-related complaints are listed.
Verified	Yes

Table 240 -Filter Complaints by Type (Car / Booking / User)

Field	Description
Test ID	AdminReport_004
Prerequisite	At least one complaint is stored with details.
Test Procedure	Click a complaint row.
Expected Result	Details displayed: description, reporter, target (user/car/booking), attachments, timestamps.
Actual Result	All complaint details displayed correctly.
Verified	Yes

Table 241 -View Detailed Information About a Complaint

Field	Description
Test ID	AdminReport_005
Prerequisite	Complaint has an image/file attached.
Test Procedure	Click attachment thumbnail.
Expected Result	Image/file opens correctly and is viewable.
Actual Result	Attachment opens and is viewable.
Verified	Yes

Table 242 -Open Attached Evidence (Images / Files)

Field	Description
Test ID	AdminReport_006
Prerequisite	Complaint currently in “Open” state.
Test Procedure	Admin selects status “In Progress” and saves.
Expected Result	Status updated to “In Progress”; update logged in complaint history.
Actual Result	Status updated and logged.
Verified	Yes

Table 243 -Change Complaint Status to “In Progress”

Field	Description
Test ID	AdminReport_007
Prerequisite	Complaint in “In Progress” or “Open”.
Test Procedure	Change status to “Resolved” and add resolution note.
Expected Result	Complaint becomes “Resolved”; resolution note stored and visible; reporter notified.
Actual Result	Complaint resolved; note stored; reporter notified.
Verified	Yes

Table 244 -Resolve Complaint with Note

5.1.4.5 System Dashboard & analytics

Field	Description
Test ID	AdminDash_001
Prerequisite	Admin logged in; data exists.
Test Procedure	Open Admin → Dashboard.
Expected Result	KPIs displayed: total users, total cars, total bookings, total revenue.
Actual Result	KPIs displayed correctly.
Verified	Yes

Table 245 -View Main Admin Dashboard

Field	Description
Test ID	AdminDash_002
Prerequisite	Bookings data available.
Test Procedure	Scroll to bookings chart.
Expected Result	Chart shows number of bookings per day for selected period.
Actual Result	Chart displayed with correct values.
Verified	Yes

Table 246-View Bookings per Day Chart

Field	Description
Test ID	AdminDash_003
Prerequisite	Data across many months.
Test Procedure	Set date filter to “Last 7 days”.
Expected Result	All charts and KPIs refresh for last 7 days only.
Actual Result	Dashboard refreshes correctly.
Verified	Yes

Table 247 -Filter Dashboard by Date Range

Field	Description
Test ID	AdminDash_004
Prerequisite	Platform has data from multiple cities.
Test Procedure	Select “Riyadh” as city filter.
Expected Result	Stats show only data relevant to Riyadh.
Actual Result	Dashboard displays Riyadh-only stats.
Verified	Yes

Table 248 -Filter Dashboard by City

Field	Description
Test ID	AdminDash_005
Prerequisite	Completed bookings with known sum.
Test Procedure	Compare total revenue KPI vs manual sum of bookings.
Expected Result	Total earnings KPI equals actual calculated value.
Actual Result	KPI equals calculated total.
Verified	Yes

Table 249 -Total Earnings Figure Matches Sum of Completed Bookings

Field	Description
Test ID	AdminDash_007
Prerequisite	Cars with different booking counts.
Test Procedure	View “Top Cars” widget.
Expected Result	Cars sorted in descending order by number of bookings.
Actual Result	“Top Cars” list sorted correctly.
Verified	Yes

Table 250-Top Cars by Bookings Widget

5.2 Integration Testing

This section describes the integration testing performed to ensure that all modules in the system work correctly together. The purpose is to verify data flow, module communication, and system behavior when multiple components interact.

5.2.1 Integration Test Objectives

- Ensure modules exchange data correctly.
- Validate interactions between Renter, Owner, Admin, Booking, and Payment modules.
- Ensure notifications and status changes are synchronized across all users.
- Confirm system handles errors and conflicts across modules.

5.2.2 Integration Test

Field	Description
Test ID	INT_001
Modules Involved	Login Module, User Module (Renter/Owner/Admin)
Test Procedure	User logs in with valid credentials.
Expected Result	System loads the correct dashboard based on role.
Actual Result	Correct dashboard loaded based on role.
Verified (Yes/No)	Yes

Table 251- Login Module ↔ User Roles

Field	Description
Test ID	INT_002
Modules Involved	Owner Module, Car Listing Module
Test Procedure	Owner adds or edits car details.
Expected Result	Car updates appear immediately in renter search results.
Actual Result	Car listing updated and visible to renter.
Verified	Yes

Table 252 -Owner Module ↔ Car Listing Module

Field	Description
Test ID	INT_003
Modules Involved	Availability Calendar, Booking Module
Test Procedure	Owner blocks specific dates in calendar.
Expected Result	Renters cannot book on blocked dates; dates hidden from search.
Actual Result	Blocked dates not available for renter.
Verified	Yes

Table 253 -Availability Calendar ↔ Booking Module

Field	Description
Test ID	INT_004
Modules Involved	Search Module, Car Listing, Availability Calendar
Test Procedure	Renter searches for cars with selected date range.
Expected Result	Only cars available for those dates appear.
Actual Result	Search results match availability.
Verified	Yes

Table 254 -Renter Search ↔ Availability ↔ Car Details

Field	Description
Test ID	INT_005
Modules Involved	Booking Module, Owner Module, Notification System
Test Procedure	Renter submits booking → owner approves/rejects.
Expected Result	Booking status updates and renter receives notification.
Actual Result	Status updated and renter notified.
Verified	Yes

Table 255-Booking Request ↔ Owner Approval ↔ Renter Notification

Field	Description
Test ID	INT_006
Modules Involved	Payment Gateway, Booking Module, Earnings Module
Test Procedure	Renter pays for booking.
Expected Result	Payment recorded; booking confirmed; owner earnings updated.
Actual Result	Payment recorded and earnings updated.
Verified	Yes

Table 256-Payment Module ↔ Booking Module ↔ Earnings Module

Field	Description
Test ID	INT_007
Modules Involved	Earnings Module, Withdrawal Module, Admin Panel
Test Procedure	Owner requests withdrawal → Admin approves/rejects.
Expected Result	Owner balance updates and owner receive notification.
Actual Result	Balance updated and notification received.
Verified	Yes

Table 257-Earnings Module ↔ Withdrawal Module ↔ Admin Module

Field	Description
Test ID	INT_008
Modules Involved	Ratings & Reviews, Booking Module
Test Procedure	Renter submits ratings after completing booking.
Expected Result	Car and owner rating updated.
Actual Result	Ratings updated correctly.
Verified	Yes

Table 258 -Ratings Module ↔ Completed Booking

Field	Description
Test ID	INT_009
Modules Involved	Notification System, Booking, Owner, Renter, Admin
Test Procedure	Trigger events (booking approval, payment, suspension).
Expected Result	Correct notifications sent to relevant users.
Actual Result	Notifications sent successfully.
Verified	Yes

Table 259 -Notification System ↔ All Modules

Field	Description
Test ID	INT_010
Modules Involved	Admin Module, User Module, Car Listing, Booking Module
Test Procedure	Admin suspends users, rejects car, or force-cancels booking.
Expected Result	Status changes reflect across all related modules.
Actual Result	Status updates synced across system.
Verified	Yes

Table 260-Admin Actions ↔ User/Car/Booking Modules

5.3 Conversion Testing

Conversion Testing ensures that all existing or pre-stored data is correctly migrated, transformed, or imported into the FADL ZAHR system without loss, duplication, or corruption. This includes user accounts, car listings, booking history, payment records, and verification documents (when applicable).

Objectives

- Verify that imported data appears correctly in its corresponding module (users, cars, bookings).
- Ensure old data formats are converted to the new structure without errors.
- Validate that no records are duplicated or missing after conversion.
- Confirm that important identifiers (User ID, Car ID, Booking ID) remain consistent.

Test Activities

- Import user data → verify fields display correctly.
- Import car listings → check for completeness and accuracy.
- Convert historical booking data → ensure bookings appear in reports and dashboards.
- Validate payment transaction consistency after migration.
- Ensure relationships (e.g., user-to-car, booking-to-user) remain intact.

Expected Results

All converted data loads correctly, remains accessible through relevant modules, and maintains full integrity.

5.4 Job Stream Testing

Job Stream Testing ensures that all scheduled background processes execute correctly and at the correct time within the system.

These automated processes include notifications, cleanup operations, scheduled reports, availability updates, earnings release, and withdrawal processing.

Objectives

- Validate that all scheduled jobs run automatically at required intervals.
- Ensure each job completes successfully without manual intervention.
- Confirm that job outputs (notifications, earnings release, reminders) are delivered correctly.
- Verify that incomplete or failed jobs are logged and can be retried.

Examples of Scheduled Jobs

- Daily notifications for upcoming bookings.
- Automatic release of owner earnings after booking completion.
- Clearing expired booking requests.
- Sending scheduled reminders for profile completion or document expiry.
- Weekly system reports for admins.

Expected Results

All job streams execute at the correct time, produce accurate results, and do not interfere with live system operations.

5.5 Interface Testing

Interface Testing ensures that all system interfaces (internal and external) communicate correctly and return expected output. This involves API calls, UI-to-backend communication, and any integration with third-party services.

Objectives

- Validate correct API request and response structures.
- Ensure that each UI screen retrieves data from the correct backend service.
- Confirm error handling when interfaces fail or return invalid data.
- Verify external service connections (payment gateways, notifications, identity verification).

Interfaces Tested

- Frontend ↔ Backend APIs (authentication, search, booking, payments).
- System ↔ Payment Gateway (charge, refund, settlement).
- System ↔ Notification Service (push, in-app alerts).
- System ↔ Database (CRUD operations).
- Admin Panel ↔ System APIs (user, car, booking management).

Expected Results

All interfaces respond correctly, handle errors gracefully, and maintain consistent data across modules.

5.6 Security Testing

Security Testing validates protection against unauthorized access, data breaches, and misuse. It ensures that the FADL ZAHR platform meets security standards and protects user information.

Objectives

- Ensure only authorized users access restricted features.
- Validate password rules, encryption, and secure login.
- Confirm proper handling of sensitive data (IDs, payment data).
- Test protection against common security threats.

Security Areas Tested

- Authentication & Authorization
 - Strong password enforcement
 - Role-based access control (Admin / Renter / Owner)
- Data Protection
 - Encryption of passwords
 - Secure transport (HTTPS)
- Session Management
 - Auto logout on inactivity
 - Prevent session hijacking
- Threat Prevention
 - SQL Injection
 - Cross-Site Scripting (XSS)
 - Cross-Site Request Forgery (CSRF)
- Account Security
 - Login attempt limits
 - Suspended accounts cannot log in
 - Email/SMS verification (if applicable)

Expected Results

System remains secure under all tests and prevents unauthorized access or manipulation.

5.7 Recovery Testing

Recovery Testing checks the system's ability to recover from failures, errors, crashes, or unexpected shutdowns with minimal data loss and safe restoration of services.

Objectives

- Validate system recovery after server crash or power failure.
- Ensure no data corruption occurs during unexpected shutdowns.
- Confirm recovery procedures for database rollback or restore.
- Test that interrupted processes (payments, bookings) complete correctly after recovery.

Recovery Scenarios Tested

- System crashes during booking creation → booking should not be duplicated or lost.
- Power failure during payment → system should ensure payment is verified safely.
- Server restart → services should resume normally.
- Interrupted upload (car photos, documents) → system should resume or notify users.

Expected Results

System recovers quickly, maintains data integrity, and ensures all components return to a stable and operational state.

5.8 Performance Testing

Performance Testing ensures that the FADL ZAHR system delivers acceptable speed, responsiveness, and stability under various load conditions. The purpose is to confirm the system can handle expected user volume and data processing demands.

Objectives

- Measure response time for critical operations (login, search, booking, payment).
- Validate system behavior under normal, peak, and stress load conditions.
- Ensure database queries and APIs return results within acceptable time limits.
- Verify scalability and stability during prolonged usage.
- Identify performance bottlenecks in APIs, database, or UI components.

Performance Tests Conducted

- Load Testing:
 - Simulate multiple users searching, booking, and browsing at the same time.
- Stress Testing:
 - Increase load beyond system capacity to observe failure points.
- Spike Testing:
 - Sudden increase in traffic (e.g., promotions or holiday rush).
- Endurance Testing:
 - System tested for long durations to detect memory leaks or performance degradation.

- Database Performance:
 - Test indexing, query speed, and handling large datasets.

Expected Results

- System remains stable with no crashes.
- All pages load within acceptable time (under 3 seconds on average).
- Database responds efficiently even under high traffic.
- No data loss or corruption during peak loads.

5.9 Regression Testing

Regression Testing ensures that new updates, fixes, or feature enhancements do not break existing functionality.

Objectives

- Re-test previously validated features after changes are implemented.
- Ensure core modules (Login, Booking, Car Listing, Payments, Notifications) still operate correctly.
- Verify that bug fixes do not introduce new issues.
- Maintain system integrity after deployment.

Regression Tests Performed

- Re-tested booking workflow from start to finish.
- Verified car listing add/edit/deactivate still functions.
- Confirmed notifications still trigger correctly after updates.
- Re-executed login, signup, and password flows.
- Validated earnings and withdrawal features after backend improvements.
- Checked admin controls (suspend user, reject car, etc.) after new releases.

Expected Results

- All previous working functionalities remain stable and error-free.
- No new bugs were introduced due to code changes.
- System continues to behave consistently across modules.

5.10 Acceptance Testing

Acceptance Testing ensures that the final system meets client expectations, business requirements, and user needs before deployment.

Objectives

- Validate the system against all functional and non-functional requirements.
- Ensure complete workflows behave as expected for Renter, Owner, and Admin users.
- Obtain client approval for production release.
- Confirm that all mandatory features are implemented and tested.

Acceptance Test Activities

- Test Renter Journey
 - Search → Booking → Payment → Trip completion → Rating
- Test Owner journey
 - Add car → Approve bookings → Earnings → Withdrawal
- Test Admin journey
 - Manage users, cars, bookings, and complaints
- Validate UI and navigation for smooth user experience
- Check notifications, security, and performance requirements
- Conduct User Acceptance Testing (UAT) with stakeholders

Expected Results

- All acceptance criteria satisfied.
- No critical or major defects remain unresolved.
- Client signs off and approves the system for release.

5.11 Beta Testing

Beta Testing is performed by a selected group of real users before full system release. It helps identify usability issues, unexpected behaviors, and edge cases in real-life environments.

Objectives

- Gather user feedback on system usability, clarity, and performance.
- Detect issues not found during internal testing.
- Validate the system under real-world conditions (different devices, networks, user behaviors).
- Improve system reliability before public launch.

Beta Testing Activities

- Provide early-access version to selected renters and owners.
- Collect feedback regarding:
 - Ease of booking
 - Car listing usability
 - App responsiveness
 - Notification timing and clarity
 - Payment and earnings flow
- Log all issues and suggestions reported by beta testers.
- Prioritize fixes for critical and high-impact issues.

Expected Results

- Enhanced user experience based on tester feedback.
- Identification and resolution of remaining usability or functional issues.
- Increased confidence in system readiness for full deployment.

6. Pass / Fail Criteria

This section defines the criteria used to determine whether each part of the *Fadl Zahr* system has successfully passed the testing phase. Since the application involves sensitive operations such as identity verification, online payments, and car booking management, it is important that all functions operate accurately and consistently. A feature is considered “passed” when it performs its expected behavior without errors and produces stable results across different scenarios. Any major malfunction that affects security, booking flow, or data accuracy will be marked as a failure and must be resolved before moving to the next phase of testing.

6.1 Suspension Criteria

Testing is temporarily suspended when certain conditions prevent the process from continuing. Some examples of situations that may cause testing to stop in the *Fadl Zahr* system include:

- Failure or unavailability of the identity verification service (Absher), which makes registration and verification tests unable to run.
- Interruption of payment services such as STC Pay or Mada, preventing payment-related test cases from executing.
- The system being unable to read from or write data to Firebase.
- Receiving different results for the same inputs, indicating an issue in the system logic.
- Technical issues in the testing environment, such as internet outages or unexpected updates.
- A major error that blocks essential steps like logging in or creating a booking.

When any of these conditions occur, testing is paused until the issue is resolved

6.2 Resumption Criteria

Testing is resumed once the issue that caused the suspension has been fixed. Testing can continue when:

- The development team has resolved the problem and verified that the feature works correctly.
- Absher or payment services return to normal operation.
- The connection to the database becomes stable again.
- The testing environment is confirmed to be functioning properly.
- The test results become consistent, and no unexpected behavior appears.

After these conditions are met, testing continues from the point where it was stopped.

6.3 Approval Criteria

A test is considered “approved” when the actual results match the expected outcomes defined in the test plan. This means:

- All system functions operate without errors or abnormal behavior.
- Payment operations are completed successfully and properly recorded in the system.
- No double booking occurs for the same vehicle within the same time period.
- Identity verification (OTP and Absher) works correctly.
- Data displayed in the user interfaces is accurate and complete.

When these conditions are met, the test results are officially approved

7. Testing Process

This section describes all tasks required to plan, design, execute, and close the testing activities for the Fadl Zahr system.

7.1 Test Deliverables

1. **Software Test Plan (STP):** A document describing the testing scope, strategy, schedule, resources, and responsibilities.
2. **Test Design Specifications Documentation:** outlines the test conditions and test scenarios for each system feature, including:

- User authentication and Absher identity verification
- Car listing management (add, edit, delete, upload photos, pricing)
- Car search and discovery
- Booking management (requests, approval, cancellation, history)
- Payment simulation
- Rating and review module
- Admin dashboard and management tools

3. **Test Cases Document:** A detailed set of test cases describing:

- Test input
- Preconditions
- Execution steps
- Expected output
- Actual outputs
- Pass/Fail results

4. **Test Data:** The data required to execute the test cases, including:

- Test user accounts (owner, renter, admin)
- Sample car listings
- Booking samples
- Payment test data

5. **Defect/Bug Reports Documentation:** of defects encountered during testing, including severity level and resolution status.

6. **Test Execution Reports:** Reports showing test progress, test case results, coverage, and outstanding issues.

7. **Regression Test Report:** A document summarizing results after retesting previously working features impacted by changes.

8. **Final Test Summary Report:** A complete evaluation of the testing phase, including achieved results, system stability, and readiness for acceptance.

9. **Requirements Traceability Matrix (RTM):** A matrix mapping each SRS requirement to its corresponding test case to ensure full coverage.

10. **Test Scripts:** Automated or semi-automated scripts for UI or API testing.

7.2 Testing Tasks

7.2.1 Test Planning Task

These tasks are performed before test execution begins:

1. Review the SRS, SPMP, Proposal, and design documents to fully understand system requirements.
2. Define the scope of testing, the test types (component, integration, interface, performance, security, acceptance), and test objectives.
3. Identify features to be tested and features not to be tested.
4. Select testing tools (manual testing as the primary approach unless automation is introduced).
5. Prepare test schedule and allocate responsibilities among team members.
6. Set acceptance criteria based on the success measures defined in the Proposal, such as error-free booking, secure authentication, and a successful end-to-end rental flow.

7.2.2 Test Design Tasks

These tasks involve preparing detailed testing materials:

1. Develop Test Design Specifications for all system features, including:
 - Authentication & Absher Verification
 - Car Listing Management
 - Car Search & Discovery
 - Booking & Scheduling
 - Payment Simulation
 - Rating and Review
 - Admin Management
2. Write detailed test cases with inputs, steps, and expected results.
3. Prepare test data (test accounts, sample listings, booking data).

4. Create the Requirements Traceability Matrix (RTM) mapping requirements to test cases.
5. Identify preconditions, constraints, and dependencies for each test.

7.2.3 Test Execution Tasks

These tasks occur during the actual testing process:

1. Execute component-level tests for individual features (add car, login, payment, ...).
2. Conduct integration testing to ensure modules work together, such as:
 - Login → User dashboard
 - Search → Booking → Payment
 - Owner approval → Renter notification
3. Perform interface testing for all user types (Owner, Renter, Admin).
4. Execute security tests focusing on authentication, identity verification, and data protection.
5. Conduct performance testing to evaluate system responsiveness (search speed, booking flow).
6. Execute acceptance testing based on real user scenarios and project success criteria.
7. Log all discovered defects in the Bug Report with severity classification.
 - All tests will be executed according to the prepared test cases and RTM.

7.2.4 Defect Management Tasks

Tasks related to tracking and resolving system issues:

1. Document defects with ID, steps to reproduce, severity, and status.
2. Communicate defects to the development team for correction.
3. Re-test resolved issues (Regression Testing).
4. Update defect records until they are confirmed closed.
5. Produce a summary of defects by category and severity.

7.2.5 Test Closure Tasks

These tasks are performed once all planned tests are executed:

1. Prepare the Final Test Summary Report.
2. Evaluate whether test objectives and acceptance criteria were fully met.
3. Confirm that all critical defects have been resolved or justified.

4. Document lessons learned and recommendations for future releases.
5. Obtain final approval from the supervisor and project team.

7.3 Responsibilities

Role	Responsibilities
Team Leader	<ul style="list-style-type: none"> • Oversees the entire testing process. • Ensures test activities follow schedule and quality standards. • Assigns test tasks to team members. • Reviews test documentation and approves test reports
Testers – All Team Members	<ul style="list-style-type: none"> • Execute test cases and record results. • Identify, document, and report defects. • Participate in regression testing. • Validate that each requirement is fully tested.
Developers	<ul style="list-style-type: none"> • Fix defects reported during testing. • Provide updated builds for re-testing. • Clarify system behavior when needed.
Project Supervisor	<ul style="list-style-type: none"> • Reviews final testing outcomes. • Confirms system readiness for acceptance. • Provides approval on the final test summary report.

Table 261 – Responsibilities

7.4 Resources

7.4.1 Hardware Resources

The following hardware resources are required for testing:

- Smartphones (Android and iOS devices) for mobile app testing.
- Laptops for running test tools, documentation, and reporting.
- Stable internet connection to support cloud services (Firebase, APIs).

7.4.2 Software Resources

The following software and tools will be used:

- Firebase (database & authentication backend).
- Development IDEs: Android Studio and/or Xcode
- Testing Tools:
 - Postman for API testing
 - Chrome DevTools for debugging
 - Selenium/Appium for automated testing

- Version Control: GitHub repository.
- Communication Tools: WhatsApp, Trello.
- All tools will be configured before execution to ensure consistent testing results across devices.

7.4.3 Data Resources

- Data required for test execution includes:
- Test user accounts (Renter, Owner, Admin)
- Sample car listings with different details (photos, prices, availability)
- Booking data for testing scheduling, approval, and cancellation
- Payment test data for simulated transactions
- Admin data for moderation and management tasks

7.5 Schedule

Testing will follow the timeline specified in the SPMP, where the allocated time for testing is 2 weeks, followed by acceptance and final evaluation.

Week	Activity
Week 1	<ul style="list-style-type: none">• Prepare Test Design Specifications• Develop test cases and test data• Build Requirements Traceability Matrix (RTM)
Week 2	<ul style="list-style-type: none">• Execute component testing• Conduct integration and interface testing
Week 3	<ul style="list-style-type: none">• Perform security and performance testing• Execute acceptance testing• Log and track defects
Week 4	<ul style="list-style-type: none">• Conduct regression testing after fixes• Prepare Final Test Summary Report• Supervisor review and approval

Table 262 - Schedule

8. Environmental Requirements

8.1 Hardware

Hardware requirements that are necessary for testing the environment are as follows:

- **Devices:** These are the systems on which the tests will be conducted, and they need to have the recommended hardware specifications in terms of RAM, storage, and central processing units to run an application. The devices must support testing of the application on Android operating system, version 7.0 and above, and iOS, version 11.0 and above.
- **Network:** A stable internet connection is necessary since the application needs to engage in real-time data exchange; for example, authentication of identity through Absher and payment processing.
- **Device Testing:** The application should be tested on devices that are compatible with the operating systems mentioned above, ensuring its compatibility on multiple devices.

8.2 Software

The software needed to carry out these test procedures includes:

- **Operating Systems (OS):** The application needs to be tested on Android devices (version 7.0 or higher) and iOS devices (version 11.0 or higher).
- **Development Environment:** Development on the Android platform should be done by using tools like Android Studio, while on the iOS platform, it should be developed on Xcode.
- **Testing Frameworks:** The testing of the functional parts of mobile applications should use automated testing tools like Selenium and Appium.
- **External APIs:** The application must be integrated with external APIs like Absher for identifying any individual and Firebase for managing users and data in real time.

8.3 Security

The security requirements for the testing environment are as follows:

- **Access Control:** There is a need for sensitive data protection through access control strategies dependent on different user roles - car owners, renters, and administrators.

- **Data Encryption:** Techniques like encryption by SSL/TLS are important in enforcing the safety of communication between client and server and protecting the personal and financial data of users.
- **Security Vulnerability Assessments:** Regular security reviews and penetration testing must be done in order to identify and fix any potential vulnerabilities before the final release.
- **Backup Procedures:** In the test period, periodic data backups should be performed to avoid the loss of important data.

8.4 Tools

The following tools and techniques shall be utilized in the testing phase:

- **Test Management Tools:** The test cases, defects, and overall progress are tracked using tools such as Jira and TestRail.
- **Automated Testing Tools:** The application should perform automated functional testing using tools like Selenium or Appium.
- **Defect Tracking Tools:** Defects should be logged using tools such as Bugzilla or JIRA, together with the tracking of their resolution.
- **Version control:** GitHub should be used to manage the source code and track changes during the testing phase.

8.5 Publications

The following documents are required to support the testing activities:

- **Software Requirements Specification (SRS):** Understand functional and non-functional requirements, test accordingly.
- **Test Plan:** The detailed test plan needs to be prepared based on test cases, methodologies, and required resources.
- **Manuals:** User manuals will help in understanding the expected user experience and facilitate the creation of user-focused test scenarios.

- **Release Notes:** Release notes should be included for tracking changes during the development and testing phases.

8.6 Risks and Assumptions

- **Availability of Test Items:** Testing might be delayed due to unavailability or downtime of parts of the system under test, such as third-party APIs like Absher and payment gateways.
- **Availability of Test Resources:** The test team must have required devices and software, and all the test accounts must be prepared prior to the beginning of the testing.
- **Time Constraints:** Testing should be finished within the allowed time, which is around 1 week.
- **External Dependencies:** Third-party integrations like Absher and payment gateways may add risks if their APIs go through changes or if there are any compatibility issues.
- **Contingency Plans:** Extra time slots for the testing phase, availability of all resources, and focus on the most critical tests are some of the contingency plans if there are issues regarding testing.

9. Change Management Procedures

Any future updates or modifications to the Software Test Plan (STP) of Fadl Zahr must follow a structured and documented change management process. All change requests whether initiated by a team member, the project advisor, or the client must first be submitted to the development and testing team.

The proposed change will then be reviewed in a team meeting, where all members evaluate its necessity, its impact on the testing process, and its alignment with the project requirements. A change will only be approved when all team members and the supervisor agree to implement it. Once approved, the change will be added to the STP, and the version number will be updated accordingly.

If any member objects, or if the change negatively affects the testing schedule or the system's stability, the request will be rejected, and no modifications will be made. All accepted and rejected changes will be documented for reference.

10. Plan Approvals

#	Name	Signature	Date
1	Jood Mutlaq Abaqmi		
2	Nada Ali Alshamsi		
3	Jana Fahad Alharbi		
4	Zahraa Mohammed Alqattan		Des 6,2025
5	Batool Sadiq Alsharqi		
6	Kadijah Yasser Al bader		
7	Roaa abbas Alhaddad		
8	Shaden Alghamdi		

Table 263 – plan approvals