

Ride Sharing Platform

Group Members

19sw102 (Alima khan)
19sw104 (Tooba Arain)
19sw112 (Amrat kumar)

2-10-2023

—

SOFTWARE QUALITY ENGINEERING

—

DR NAEEM AHMED

Project Initiation:

Introduction: We're making an app that helps you book electric cars and share rides. It's good for the environment and makes getting around town easier. You can use it on your computer or phone, and our team is making sure it works perfectly. Let's make city travel cleaner and more convenient!

Project overview: Our ride-sharing project is on a mission to transform urban transportation. We're introducing an innovative platform designed to make getting around the city more convenient, sustainable, and secure. Key features include hassle-free ride booking, encouraging eco-friendly carpooling, and ensuring safe and seamless payment processing. With this, we aim to address the challenges of urban mobility, offering a greener and more efficient way to travel in cities .

Defining Scope:

Scope Statement: Our project's scope encompasses the boundaries and constraints that guide its development. It includes:

1. Developing web and mobile applications to cater to a diverse user base.
2. Adhering strictly to the defined Software Development Life Cycle (SDLC) phases.
3. Ensuring compliance with transportation regulations and data privacy laws.
4. Promoting eco-friendliness and sustainable transportation solutions as core values.

Objective and Goals

Certainly, here are the objectives and goals presented in a tabular form for clarity:

Objectives	Descriptions
Objective 1 - Promoting Eco-friendliness	Encourage eco-friendliness and sustainable transportation by promoting carpooling and reducing individual vehicle usage.
Objective 2 - Web and Mobile Applications	Develop web and mobile applications to accommodate a diverse user base, ensuring accessibility and user convenience.
Objective 3 - Strict SDLC Adherence	Adhere rigorously to the defined Software Development Life Cycle (SDLC) phases to maintain systematic and efficient project execution

Planning: In this phase, the team planned project timeline and Role and Responsibilities: Estimated Project Timeline (CEP – A3):

Phase 1: Project Initiation (1 week): Define project scope, objectives, and Goals.

Phase 2: Requirements Gathering (1 week): Identify and document functional and non-functional requirements. Review and validate requirements.

Phase 3: Database Design and Backend Development (2 weeks)

Design the database schema for user information, vehicle details, payments, etc.

Develop backend systems for user registration, authentication, and payment processing. Implement database constraints.

Phase 4: Frontend Development (1 weeks):

Develop web and mobile applications for ride booking, carpooling, and user profiles.

Implement user interfaces for navigation.

Phase 5: Integration of Renewable Energy (2 weeks)

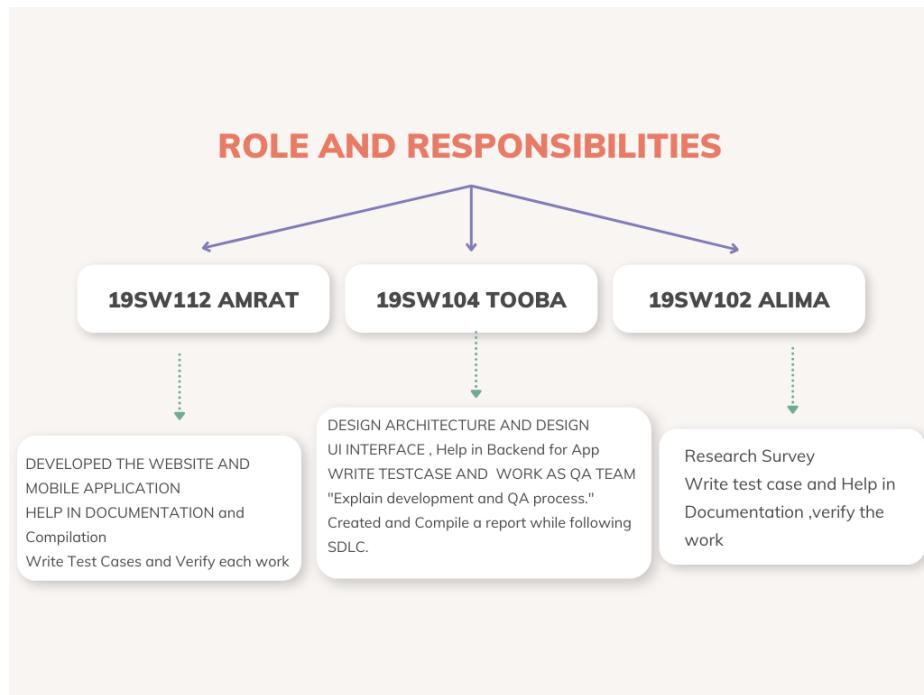
Survey renewable energy sources like solar and wind power. Assess feasibility and costs of integrating renewable energy into charging infrastructure.

Phase 6: Quality Assurance and Testing (3 weeks)

Design and execute test cases for all platform functionalities. Conduct unit testing, integration testing. Identify and resolve bugs and issues.

Phase 7: Deployment (4 days)

Prepare the platform for deployment to production servers. Perform final testing and verification.



Requirements Gathering: Project Design Requirements (CEP – A1, CEP – A2):

Functional Requirements: Project CEP – A2): Identify and functional requirements for the ride-sharing platform

Design document

Requirements the functional ride-sharing platform

(CEP – A1, and non-

1. User Registration:

- Users can create accounts with their first name, last name, email, and password.
- Account creation requires email verification.

2. Ride Booking:

- Users can view ride details, history, including vehicle type, driver information, and price.
- Users can book rides for a specific date and time.

3. Carpooling:

- Users can join existing carpools or create new ones.
- Carpooling functionality optimizes routes to reduce carbon emissions and traffic congestion.
- Users can invite others to join their carpools.

4. Payment Processing:

- Users can make payments for booked rides.
- Payment processing is secure and supports various payment methods (credit/debit cards, mobile wallets).
- Invoices are generated and sent to users after each ride.

Non-Functional Requirements:

1. Performance:

- The platform should handle a high volume of simultaneous users without significant performance degradation.
- Response times for critical actions (e.g., ride booking) should be within seconds.

2. Security:

- User data, including passwords, must be securely stored and transmitted using encryption.
- • Payment transactions must be protected with robust security measures.

3. Scalability:

- The system should be designed to scale horizontally to accommodate a growing user base.
- Scalability should not impact performance negatively.

4. Usability:

- The user interface should be intuitive and user-friendly on both web and mobile applications.

5. Reliability:

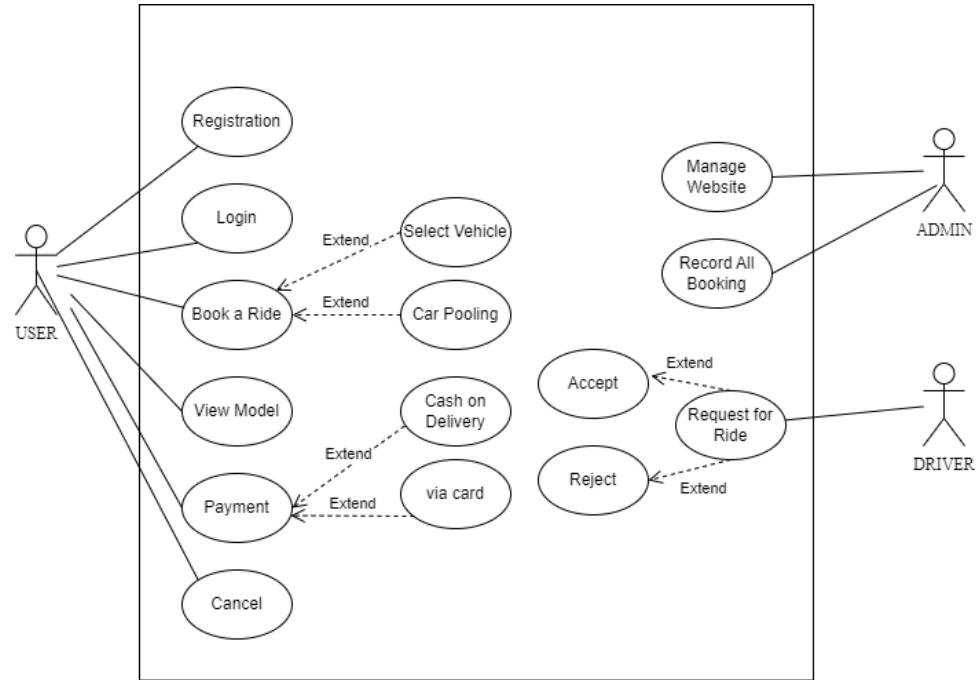
- The platform should have high uptime and be available 24/7, with minimal downtime for maintenance.

6 Eco-Friendliness:

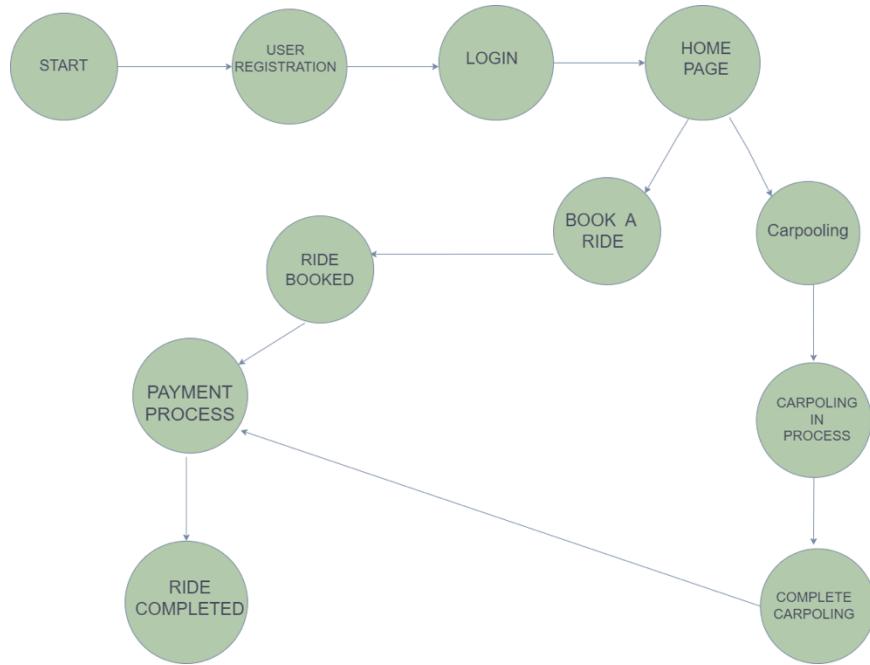
- The platform should prioritize eco-friendliness by promoting shared rides and integrating renewable energy

Designing and Architecture:

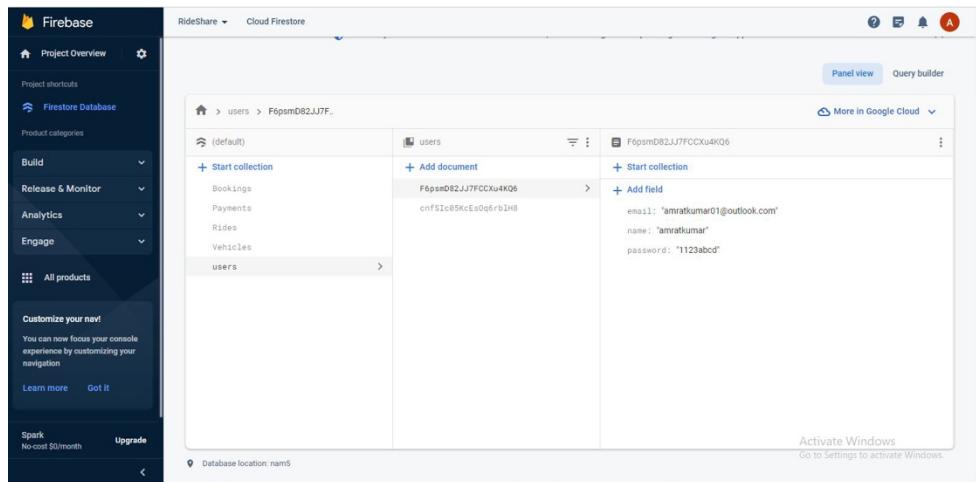
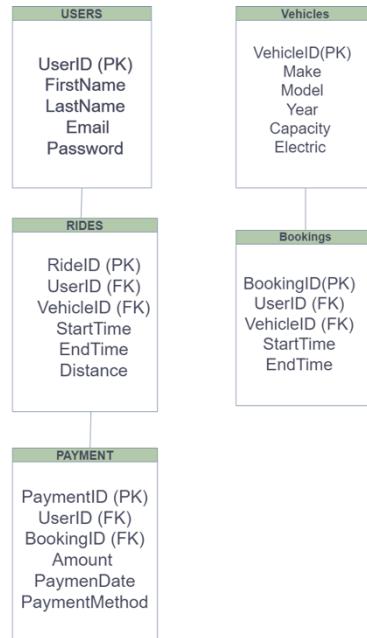
Use Case Diagram:



State Transition Diagram:



Database Design: Database Design – A3, CEP – A9:



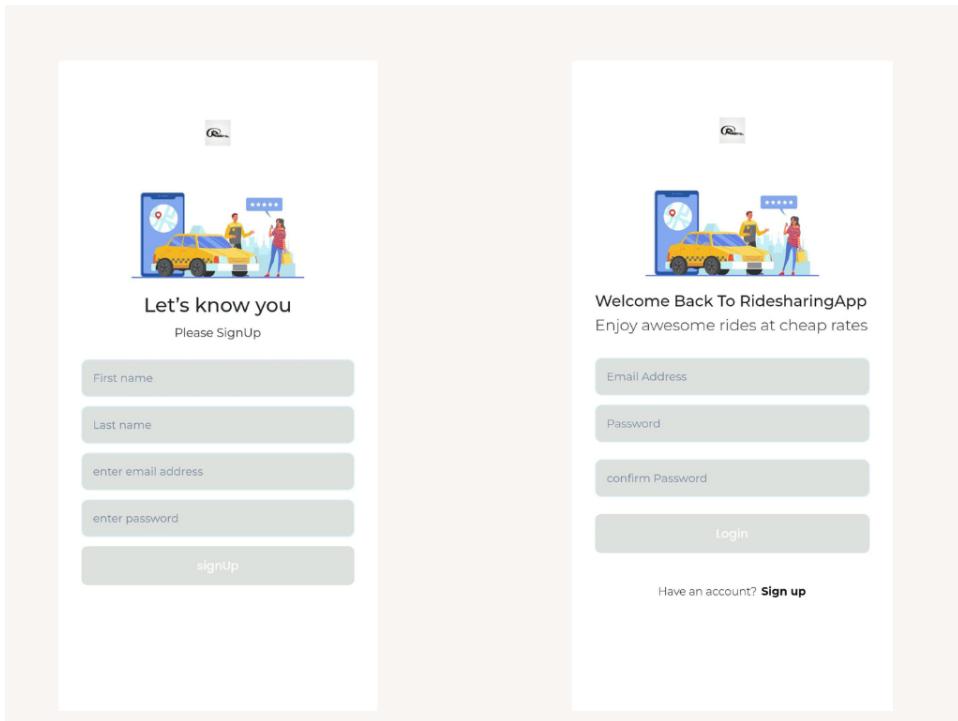
The screenshot shows the Firebase Project Overview on the left and the Cloud Firestore interface on the right. The left sidebar has 'Project Overview' at the top, followed by 'Project shortcuts' and 'Firestore Database'. Under 'Firestore Database', there are sections for 'Build', 'Release & Monitor', 'Analytics', 'Engage', and 'All products'. A 'Customize your nav!' section allows users to focus their console experience by customizing navigation. At the bottom, there are 'Learn more' and 'Got it' buttons, and a 'Spark' plan with a 'No-cost \$0/month' option and an 'Upgrade' button.

The main area shows the 'RideShare' database. In the 'Cloud Firestore' section, the 'Rides' collection is selected. The left sidebar under 'Rides' shows 'Bookings', 'Payments', 'Rides' (selected), 'Vehicles', and 'users'. The right panel displays a document named 'GaNmcASNtVbXt70gSep' with fields: 'CurrentLocation: "hyderabad"', 'Destination: "Karachi"', 'RideID: "112"', and 'email: "amrakumar01@outlook.com"'. There are buttons for '+ Start collection', '+ Add field', and '+ Add document'.

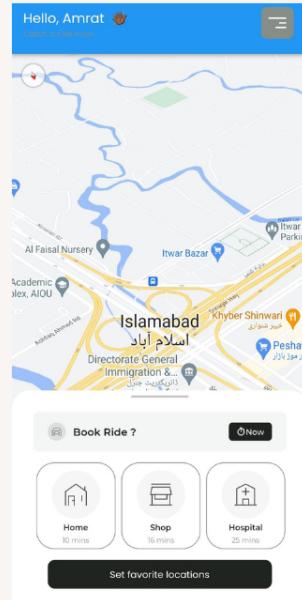
This screenshot shows the same Firebase Project Overview and Cloud Firestore interface as the previous one, but with a different collection selected. The 'Payments' collection is now selected in the left sidebar under 'Payments'. The right panel shows a document named 'IUM7F1lQUWtEqw6Sqab' with fields: 'Amount: "13,300"', 'AmountDate: "30-9-2023"', 'PaymentId: "01"', and 'email: "amrakumar01@outlook.com"'. There are buttons for '+ Start collection', '+ Add field', and '+ Add document'.

This screenshot shows the same Firebase Project Overview and Cloud Firestore interface, with the 'Vehicles' collection selected in the left sidebar under 'Vehicles'. The right panel shows a document named 'q7JZv1buURVuMVD03PQ' with fields: 'model: "Abscr123"', 'vehicle id: "11"', and 'year: "2007"'. There are buttons for '+ Start collection', '+ Add field', and '+ Add document'.

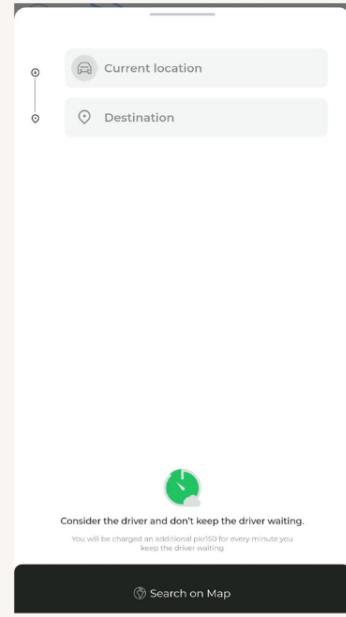
Interface:



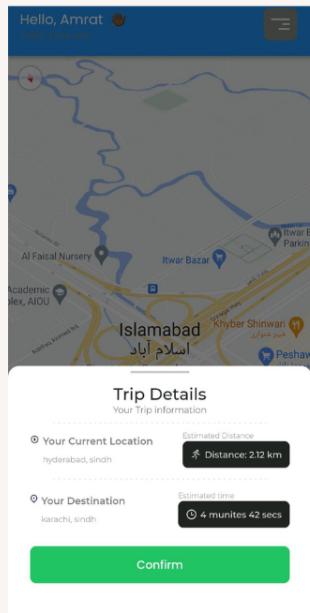
Home Screen



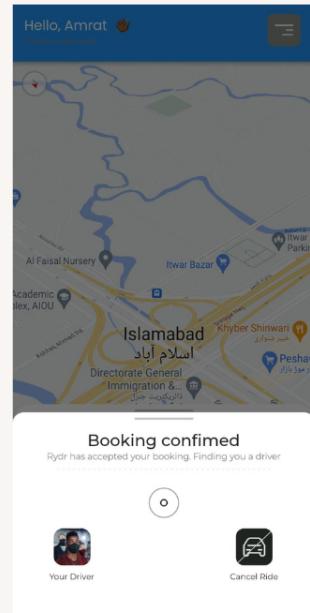
Ride Booking



Trip Detail



Booking confirmed



Add Payment Method

How would you like to pay for your trips?

Pay with Cash

Pay with Debit or Credit Card

Payment

rider Balance
pkr:13,300

Payment Methods

Pay with Cash

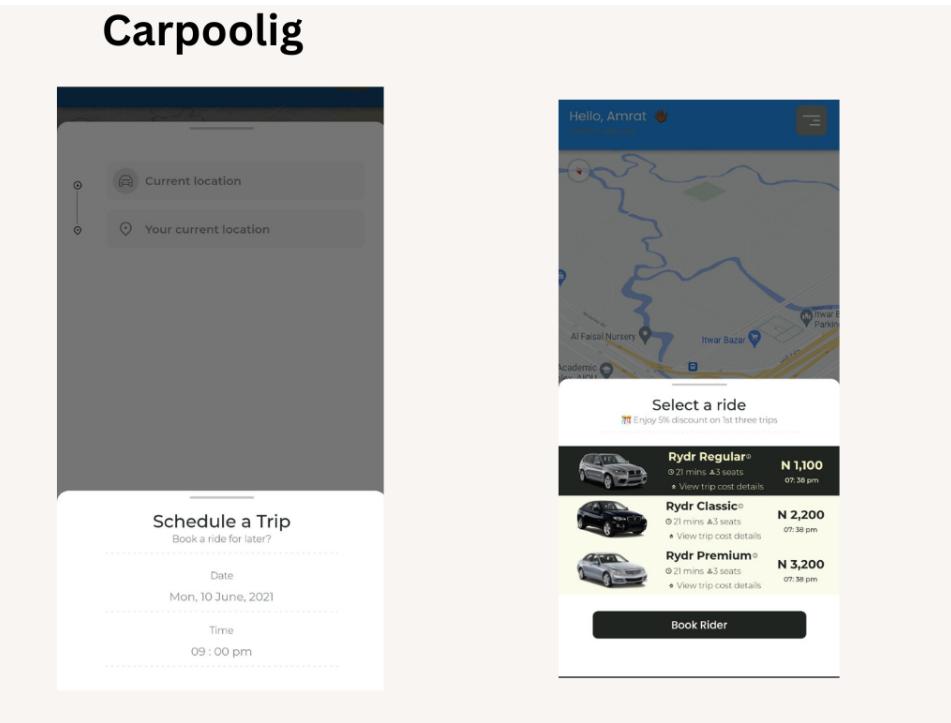
Add Payment Method

Apply Coupon

Enter Coupon Code

Apply Coupon Code

Carpoolig



WEB INTERFACE:

The image shows a screenshot of the 'Ride Booking Prototype' web application. The main form is titled 'Book a Ride' and contains the following fields:

- From: Rydrlocal
- To: Jhore
- Date: 01/06/2021
- Carpooling:
 - Offer a Carpool
 - Carpool Destination: Search
- Payment:
 - Payment Method: Cash on Delivery
 - Book and Pay Book Now

Below the main form, there are two sections: 'Carpooling' and 'Payment Processing'. The 'Payment Processing' section shows a progress bar with the message 'Processing Payment...'. At the bottom right of the page, there is a note: 'Activate Windows 10 Go to Settings to activate Windows.'

The image displays two screenshots of a web-based ride booking application. The top screenshot shows the 'Book a Ride' page with fields for 'From' (Hyderabad), 'To' (Lahore), 'Date' (2023-09-30), and 'Carpooling' options. A modal window is open, stating 'Payment will be collected on delivery. Thank you!' with an 'OK' button. Below the main form, there's a 'Carpooling' section showing a carpool from Hyderabad to Lahore on 2023-09-30 and a 'Payment Processing' section indicating a total amount of \$10.00. The bottom screenshot is identical to the top one, showing the same form and payment processing details.

Implementation: WORKING PROTOTYPE FOLDER

TESTING: Test Case Identification (CEP – A1, CEP – A7, CEP – A8):

1-User Registration:

Test Case ID	1
Created By	QA Team
Tester's Name	Amrat
S#	1
Step #	Prerequisites: Access to Mobile App
Test Case Description	User Registration and Login
Reviewed By	Amrat kumar
Version	0.1
Date Tested	2-10-2023
Test Data	User Registration Information
	User ID: testuser123
	Password: TestPass123
Step #	User Login
Test Data	User Login Information
	User ID: testuser123
	Password: TestPass123

--	--

Test Scenario: Verify User Registration Functionality

Step #	Description	Expected Result	Actual Result	Pass/Fail Description
1	Open the app	The app is launched successfully.	The app is launched successfully.	Pass - The app opens as expected.
2	Select "Register"	The registration page opens, allowing the user to input their registration details.	The registration page opens, providing fields for registration.	Pass - The registration page is accessible.
3	Fill in registration details	The user can enter valid registration details (e.g., name, email, password).	Valid registration details can be entered without errors.	Pass - Registration details can be entered.
4	Submit registration form	After submission, the user is successfully registered and can proceed to log in.	After submission, the user is registered and can log in with the provided credentials.	Fail - Registration is not successful.

Test Scenario: Verify user login functionality.

Step #	Description	Expected Result	Actual Result	Pass/Fail Description
1	Open the app	The app is launched successfully.	The app is launched successfully.	Pass - The app opens as expected.
2	Enter User email	The user can input a valid email	A valid email can be entered.	Pass - Email input is valid.
3	Enter Password	The user can input a valid password.	A valid password can be entered.	Pass - Password input is valid.
4	Click "Login"	After clicking "Login," the user is logged in and can access their account.	After clicking "Login," the user is successfully logged in and can access their account.	Fail - Login is not successful.

2-Ride booking Testcase:

Test Case ID	1
Created By	QA Team
Tester's Name	Tooba
S#	1
Step #	Prerequisites: Access to Mobile App
Test Case Description	Verify Booking Functionality
Reviewed By	Amrat kumar
Version	0.1
Date Tested	2-10-2023
Test Data	User credentials
	User ID: testuser123
	Password: TestPass123

Test scenario: verify booking ride functionality:

Step #	Description	Expected Result	Actual Result	Pass/Fail Description
1	Open the app	The app is launched successfully.	The app is launched successfully.	Pass - The app opens as expected.
2	Enter User ID	The user can input a valid User ID.	A valid User ID can be entered.	Pass - User ID input is valid.
3	Enter Password	The user can input a valid password.	A valid password can be entered.	Pass - Password input is valid.
4	Click "Login"	After clicking "Login," the user is logged in and can access their account.	After clicking "Login," the user is successfully logged in and can access their account.	Pass - Login is successful.
5	Set pick-up location	The user can select a pick-up location on the map.	A pick-up location can be selected on the map.	Pass - Pick-up location can be set.
6	Set drop-off location	The user can select a drop-off location on the map.	A drop-off location can be selected on the map.	Pass - Drop-off location can be set.

Step #	Description	Expected Result	Actual Result	Pass/Fail Description
7	Choose a vehicle type	The user can select a vehicle type (e.g., standard, luxury).	A vehicle type (e.g., standard, luxury) can be chosen.	Pass - Vehicle type can be selected.
8	Confirm the booking	After confirming the booking, the ride is successfully booked, and the user receives a ride confirmation.	After confirmation, the ride is successfully booked, and a ride confirmation is received.	Pass - Ride booking is successful.

3-Payment processing Testcase:

Test Case ID	3
Created By	QA Team
Tester's Name	Alima
S#	1
Step #	Prerequisites: Access to Mobile App
Test Case Description	Cash and Card Payment Processing
Reviewed By	Amrat kumar
Version	0.1
Date Tested	2-10-2023
Test Data	Payment Information
	Payment Method: Cash
	Payment Method: Credit Card

Test Case ID	3
	Card Number: 1234-5678-9012-3456
	Expiry Date: 12/25
	CVV: 123

Test Scenario: Verify Payment Processing Functionality

Step #	Description	Expected Result	Actual Result	Pass/Fail Description
1	Open the app	The app is launched successfully.	The app is launched successfully.	Pass - The app opens as expected.
2	Book a ride	The user can book a ride without errors.	A ride can be booked without errors.	Pass - Ride booking is successful.
3	Select payment method	The user can choose a payment method (e.g., credit card,cash) for the ride.	A payment method can be selected (e.g., credit card) for the ride.	Pass - Payment method selection is successful.
4	Confirm booking	After confirming the booking, the payment is processed, and the user	After confirmation, the payment is processed, and a ride confirmation is received.	Pass - Ride booking and payment

Step #	Description	Expected Result	Actual Result	Pass/Fail Description
		receives a ride confirmation.		processing are successful.

"QA" into the context while explaining the steps of the software development process FOR DEVELOPING RIDE SHARING APP:

Step 1 - Requirements Gathering:

In the initial phase of the software development process, the QA team collaborates with stakeholders to gather detailed project requirements, including functional, non-functional, and regulatory requirements. As part of this QA process, the team checks and verifies these requirements to ensure they are clear, complete, and feasible. This step is crucial to lay the foundation for the entire project.

Step 2 - Planning:

Following the requirements gathering, the QA team proceeds to the planning phase. Here, they create a comprehensive test plan that outlines the scope, objectives, and strategies for testing the software. The QA team also identifies the necessary resources, tools, and schedules to execute the testing process effectively. During this

phase, the QA team ensures that the testing plan aligns with the verified requirements from the previous step.

Step 3 - Design:

In the design phase, the QA team reviews design documents and architectural plans to identify any potential quality issues or gaps in the design that could impact the successful realization of the requirements. They may suggest design improvements or adjustments to ensure the software's testability and reliability while adhering to the verified requirements.

Step 4 - Implementation (Coding):

During the coding phase, we act as development teams and write the actual code based on the design specifications. In this step, the QA team ensures that the code aligns with the verified requirements, implementing the functionality as specified. The QA team also verifies that coding standards and best practices are followed to maintain code quality and facilitate testing.

Step 5 - Testing:

The testing phase is where the QA team executes test cases to validate that the software meets the specified requirements. Various types of testing, including functional, integration, performance, security, usability, and regression testing, are performed by the QA team to verify the software's quality. The QA team also identifies and reports defects, ensuring that any deviations from the verified requirements are addressed promptly.

Step 6 - Deployment:

Once the software passes testing and quality checks, it proceeds to deployment. The QA team verifies that the deployment process aligns with the requirements and ensures that the software is installed and configured correctly in the production environment.

Step 7 - Maintenance and Monitoring:

Even after deployment, the QA team continues to monitor the software in the production environment to detect any issues that may arise. They ensure that the software maintains its quality and reliability as it operates in a real-world setting. Any post-release issues are addressed promptly by the QA team to keep the software aligned with the verified requirements.

By following these steps, we QA Team ensures that the software development process is closely aligned with the assigned requirements, and the final product meets both project objectives and user expectations.