



American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

Friendly Ride Share

A Software Quality and Testing Project Submitted

By

| Semester: Fall_23_24 | | | Section: | Group No: |
|----------------------|-----------------|------------|--------------------------------|-----------------|
| SN | Student Name | Student ID | Individual Contribution (in %) | Total Marks: 50 |
| | | | | Earned Marks: |
| 39 | SANDIP MISRA | 21-44506-1 | 25% | |
| 36 | PRANTO SEN | 20-44040-2 | 25% | |
| 22 | ARGHYAJIT SADHU | 20-42867-1 | 25% | |
| 20 | SAIFUL ISLAM | 20-42667-1 | 25% | |

The project will be Evaluated for the following Course Outcomes

| EVALUATION CRITERIA | Total Marks (50) | |
|---|------------------|--|
| | | |
| Revision History, Test Plan Identifier, Reference Materials, Problem Background, Solutions | [10 Marks] | |
| Requirements Specification (System feature, Quality Attributes, System Interface, Project Requirements) | [10 Marks] | |
| Item Not to be tested, Testing approach (Testing levels, tools, meetings), Test cases | [10 Marks] | |
| Item pass/fail criteria, Test deliverables, Staffing and Training, Responsibilities, Scheduling, Risk | [10 Marks] | |
| Approval, Format, Submission, and Defense | [10 Marks] | |

Software Test Plan

for

Friendly Ride Share

Version 1.0 approved

Prepared by Sandip Misra, Saiful Islam, Arghyajit Sadhu, Pranto Sen

**American International University-Bangladesh (AIUB) Department of
Computer Science**

06 December 2023

Table of Contents

| | |
|---|-----------|
| Revision History | 3 |
| 1. TEST PLAN IDENTIFIER: TP_2023_FRS_V_1.0 | 4 |
| 2. REFERENCE MATERIALS | 4 |
| 3. INTRODUCTION | 4 |
| 3.1 Background to the Problem..... | 5 |
| 3.2 Solution to the Problem..... | 5 |
| 4. REQUIREMENT SPECIFICATION | 6 |
| 4.1 System Features | 8 |
| 4.2 System Quality Attributes..... | 10 |
| 4.3 System Interface..... | 12 |
| 4.4 Project Requirements | 14 |
| 5. FEATURES NOT TO BE TESTED | 14 |
| 6. TESTING APPROACH | 16 |
| 6.1 Testing Levels..... | 16 |
| 6.2 Test Tools..... | 16 |
| 6.3 Meetings..... | 17 |
| 7. TEST CASES/TEST ITEMS | 18 |
| 8. ITEM PASS/FAIL CRITERIA | 23 |
| 9. TEST DELIVERABLES | 23 |
| 10. STAFFING AND TRAINING NEEDS | 23 |
| 11. RESPONSIBILITIES | 24 |
| 12. TESTING SCHEDULE | 25 |
| 13. PLANNING RISKS AND CONTINGENCIES | 25 |
| 14. APPROVALS | 25 |

Revision History

| Revision | Date | Updated by | Update Comments |
|----------|------------|-----------------|-----------------|
| 0.1 | 2024.12.07 | Saiful Islam | First Draft |
| 0.2 | 2024.12.09 | Sandip Misra | Second Draft |
| 0.3 | 2024.12.10 | Arghyajit Sadhu | Third Draft |
| 0.4 | 2024.12.11 | Saiful Islam | Fourth Draft |
| 0.5 | 2024.12.13 | Pranto Sen | Fifth Draft |
| 0.6 | 2024.12.15 | Arghyajit Sadhu | Sixth Draft |
| 0.7 | 2024.12.19 | Sandip Misra | Seventh Draft |

1. TEST PLAN IDENTIFIER: [TP_2023_FRS_V_1.0](#)

2. REFERENCE MATERIALS

Pathao: Pathao is a prominent ridesharing and delivery service in Bangladesh, offering users a convenient platform to book rides and courier services through a mobile app. Similar to our Friendly Ride Share project, Pathao connects riders with drivers, providing a seamless experience for users to navigate through traffic, save time, and reach their destinations efficiently. Pathao incorporates features such as real-time tracking, transparent pricing, and a cashless payment system to enhance the overall ride-sharing experience. (<https://pathao.com/>)

Uber: Uber is a globally recognized ride-sharing platform that operates in numerous countries, providing users with a reliable means of transportation. Our Friendly Ride Share project draws inspiration from Uber's user-friendly interface, where passengers can easily request rides, view driver details, and track their journey in real-time. Additionally, the integration of a secure payment gateway in Friendly Ride Share aligns with Uber's emphasis on cashless transactions for added convenience. (<https://www.uber.com/>)

DriveGo: DriveGo is a ride-sharing service that caters to university students in specific regions. Our Friendly Ride Share project shares similarities with DriveGo in its focus on university students, ensuring a niche and tailored experience for users within a particular demographic. The concept of connecting students through a ride-sharing app, as seen in DriveGo, aligns with our project's objective of facilitating seamless transportation for students within the same university community. (<https://drivego.in/>)

BRTA Ride-Sharing Rules: The "Friendly Ride Share" project strictly adheres to the "Ride Sharing Service Rules – 2017" set by the Bangladesh Road Transport Authority (BRTA). All vehicles participating in the ride-sharing service within Dhaka, Narayanganj, Gazipur, Manikganj, Narsinghdi, and Munshiganj districts possess valid enlistment certificates issued by the BRTA. This compliance ensures that the service operates in accordance with the regulatory framework and contributes to maintaining proper social distancing during commuting. (<https://brta.gov.bd/site/page/a7c23d2f-3971-4e84-923a-94e36d75ddff/Ride-Sharing-Service-Guideline-2017>)

Government ICP Policy Alignment: In line with the Government's Information and Communication Technology (ICT) policy, the "Friendly Ride Share" project aligns its technological infrastructure with the prescribed standards. The Information and Communication Policy (ICP) guidelines set forth by the government are integrated into the platform's development, ensuring the secure and ethical use of technology for providing reliable ride-sharing services to users.

Digital Security Act: The "Friendly Ride Share" project places a paramount focus on user data security, in accordance with the Digital Security Act of Bangladesh. The platform implements robust measures to safeguard user information, ensuring confidentiality, integrity, and availability. Compliance with the Digital Security Act reflects the commitment to protecting

user privacy, preventing unauthorized access, and maintaining the highest standards of digital security throughout the ride-sharing experience. (<http://bdlaws.minlaw.gov.bd/>)

3. INTRODUCTION

3.1 Background to the Problem

University students often face difficulties finding reliable and convenient transportation within their campus environment. While ride-sharing platforms like Uber, Pathao, and DriveGo offer a general solution, they fall short in addressing the specific needs of students within a confined campus setting.

Problems:

- **Limited availability:** Ride-sharing drivers may not be readily available within campus boundaries, particularly during peak hours.
- **Unfamiliarity:** Students new to the university may be unfamiliar with safe off-campus locations or optimal routes, leading to unreliable choices.
- **Safety concerns:** The anonymity of ride-sharing drivers and potential off-campus travel can raise safety anxieties, especially at night.
- **Traffic congestion:** Over-reliance on individual vehicles within campus grounds contributes to traffic jams, hindering efficient transportation.

Root Causes:

- **Lack of dedicated services:** Existing ride-sharing platforms are not designed specifically for the needs of university students and may not operate within campus boundaries.
- **Limited information:** Students lack readily available information about safe routes and locations, leading to inefficient travel choices.
- **Anonymity and safety:** Concerns regarding the anonymity of drivers and potential for off-campus travel raise safety anxieties for students.
- **Car-centric culture:** A reliance on individual car usage within campus grounds contributes to traffic congestion and inefficiency.

Impact:

These challenges significantly impact students' daily lives, affecting their:

- **Accessibility:** Students with limited personal transportation options may face difficulties accessing educational resources and facilities.

- Time management: Inefficient travel can lead to time constraints, impacting academic performance and participation in campus activities.
- Social engagement: Limited transportation options can hinder students' ability to participate in social events and build connections within the campus community.
- Safety: Safety concerns associated with anonymous drivers and off-campus travel can create anxiety and limit student mobility.

Overall, the lack of a dedicated and user-friendly solution for student transportation within university campuses presents a significant barrier to academic success, social engagement, and overall well-being.

3.2 Solution to the Problem

Addressing the identified transportation problems and fostering a robust network among university students, the "Friendly Ride Share" project proposes a dedicated ride-sharing platform designed specifically for the campus community. This platform will offer several features to enhance student experience:

- Campus-focused service: Operates exclusively within campus grounds and surrounding approved areas, ensuring familiarity and accessibility.
- Verified community: Only registered students from Aiub, Brac, NSU, UIU, and other collaborating universities can join as drivers and riders, fostering a safe and trustworthy environment.
- Real-time tracking: Provides real-time location tracking and ride status updates, enhancing transparency and reducing safety concerns.
- Route optimization: Suggests optimal routes within campus, saving time and maximizing efficiency.
- Offline functionality: Allows users to access basic features and request rides even without internet connectivity, ensuring reliability.
- Inter-university networking: Facilitates collaboration and connections between students from various universities, fostering a broader university community.

By promoting ridesharing and encouraging students to utilize fewer cars, the "Friendly Ride Share" project aims to:

- Reduce traffic congestion: By optimizing car usage, the project can significantly lower traffic volume within campus grounds, improving overall transportation efficiency.
- Enhance safety: A verified community of students and real-time tracking features contribute to a safe and secure ride-sharing experience.

- Strengthen student connections: The platform facilitates inter-university networking, fostering new friendships and collaborations among students from different institutions.
- Promote environmental sustainability: Reducing individual car usage and promoting carpooling contributes to a more sustainable and eco-friendly campus environment.

The "Friendly Ride Share" project has the potential to revolutionize student transportation within university campuses, leading to a safer, more efficient, and interconnected campus community.

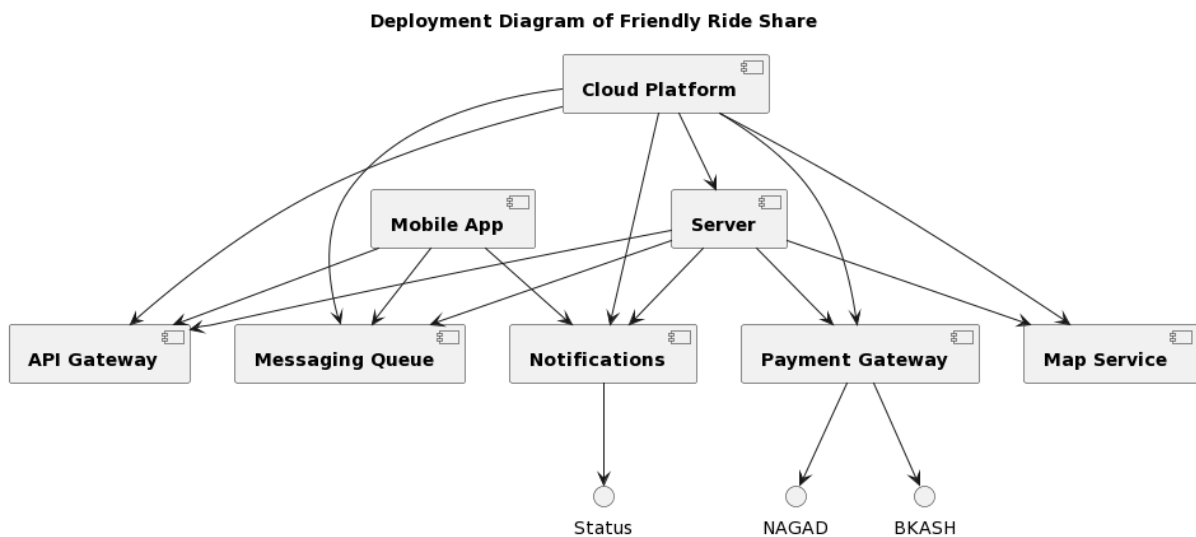


Figure 01: Deployment Diagram of Friendly Ride share

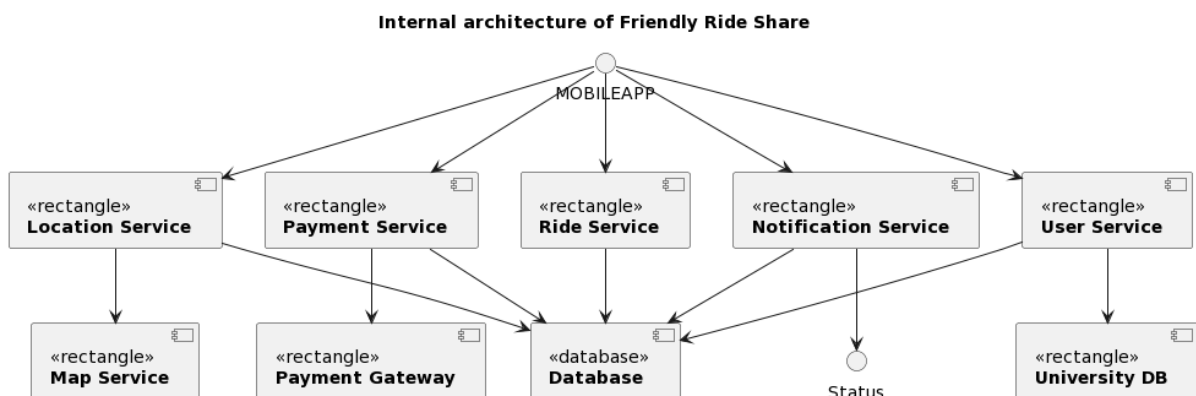


Figure 02: Internal Architecture of Friendly Rideshare

4. REQUIREMENT SPECIFICATION

4.1 System Features

1. Customer Login

Functional Requirements

1. The software shall allow users to log in with their university ID and password.
2. The login credentials (university ID and password) will be verified with database records.
3. If the login is successful, the user will be notified.
4. If invalid credentials are provided, the user will receive a notification by email.
5. If the number of login attempts exceeds its limit (3 times), the system shall block the user account login for one hour.

Priority Level: High

Precondition: User is registered.

Cross reference: N/A

2. Customer Sign Up

Functional Requirements

1. The system shall allow users to sign up for an account.
2. Users will have to fill out a form requesting their information and credentials, and any required credentials (Email, Phone) will be verified by tokens.
3. If an account with the same email or phone already exists, the user will be notified.

Priority Level: High

Precondition: User must have a valid university ID and phone number.

Cross-references: N/A

3. Browse Ride Sharers/Riders

Functional Requirements

1. The system shall allow users to view the map showing nearby ride sharers/riders.
2. Users can navigate to the map page and see the location of available ride sharers/riders within a 10 km range.

Priority Level: Medium

Precondition: None

Cross reference: N/A

4. View Ride Sharer/Rider Details

Functional Requirements

1. Users can view details of any individual ride sharer/rider, including their university, name, and nothing more.
2. The user can click on a ride sharer/rider to view details.

Priority Level: Medium
Precondition: None
Cross reference: 3.1, 3.2

5. Connect Ride Sharer and Rider

Functional Requirements

1. Users should be able to connect with a nearby ride sharer/rider via Bluetooth automatically.
2. If ride sharer and rider are within 10 feet for more than 5 minutes, the system will establish a connection for ride sharing without tapping the phone.
3. The ride sharer's and rider's Bluetooth should remain turned on for an untouchable experience of the app.

Priority Level: High
Precondition: Ride sharer and rider must be within 10 feet.
Cross reference: 1.3, 2.1

6. Confirm Ride

Functional Requirements

1. Ride confirmation should happen automatically if ride sharer and rider are connected for more than 5 minutes within a 10 feet range.
2. No manual confirmation or tapping on the phone is required.
3. After the ride is completed and the rider reaches the university, the system should process the required amount to the ride sharer's account.

Priority Level: High
Precondition: Ride sharer and rider must be connected for more than 5 minutes.
Cross reference: 1.1, 1.2, 2.2, 5.1, 5.2

7. Payment

Functional Requirements

1. Automated Payment Processing: The system shall automatically process the payment once the ride is confirmed and the rider reaches the university or the destination.
2. Payment Confirmation Notification: Both the ride sharer and the rider shall receive a real-time notification confirming the successful processing of the payment.
3. Payment Gateway Integration: The platform shall integrate secure payment gateways (e.g., BKASH, NAGAD) to facilitate seamless and reliable financial transactions.
4. Payment Receipt Generation: A digital receipt for the completed ride and the corresponding payment transaction shall be generated and made available to both the ride sharer and the rider within the mobile app.

Priority Level: High

Precondition: Ride confirmation is successfully completed.

Cross reference: 1.1, 1.2, 2.2, 5.1, 5.2, 6.1, 6.3

8. Notification

Functional Requirements

1. Real-time Ride Status Updates: The system shall provide real-time notifications to both the ride sharer and the rider regarding the status of the ride, including confirmation, ride progress, and completion.
2. Payment Confirmation Notification: Upon successful payment processing, both the ride sharer and the rider shall receive instant notifications confirming the completion of the financial transaction.
3. Low Balance Alert: In the event of low or insufficient balance in the user's account (e.g., for ride sharer's payment), the system shall send an alert notification to the affected user.
4. Ride Queue Notification: Ride sharers shall receive notifications when there are riders in their vicinity who have requested rides, allowing for proactive engagement.

Priority Level: High

Precondition: Ride confirmation and payment processing are successfully completed.

Cross reference: 1.1, 1.2, 2.2, 5.1, 5.2, 6.1, 6.3, 7.1, 7.2

9. Queue Ride

Functional Requirements

1. Queue Request: Riders can queue for a ride in advance, providing flexibility for missed rides or future trips.
2. Queue Notification: Ride sharers receive notifications about queued rides, allowing timely response.
3. Queue Seat Availability: If all seats in the ride sharer's vehicle are occupied, the queue ride option is disabled.
4. Queue Priority: Queued rides are processed based on the order received and ride sharer availability.
5. User Interaction: Riders can manage queued rides, view status, and cancel queued requests if needed.

Priority Level: Medium

Precondition: Ride sharer vehicle has available seats.

Cross reference: 6, 8

4.2 System Quality Attributes

QA1: Usability: This system should allow users to navigate to each page within a maximum of 10 seconds. All buttons should provide a response to the user from 2 to 5 seconds. After completing a ride, the user should receive a confirmation notification within 1 to 2 minutes.

Priority Level: High
Precondition: N/A
Cross-references: QA2, QA5

QA2: Availability: The system should be available for 24 hours a day and all 7 days of the week, with a 30-minute maximum downtime per day for server reset, database backup, and data integrity check.

Priority Level: High
Precondition: N/A
Cross-references: QA5, QA8

QA3: Security: All user data should be protected and kept private. User passwords should be saved after being hashed using a high complexity hashing algorithm.

Priority Level: High
Precondition: N/A
Cross-references: QA7

QA4: Scalability: The project is intended to be released globally; therefore, the system needs to be at least 80% to handle an increasing number of users, requests, and data.

Priority Level: Medium
Precondition: N/A
Cross-references: QA1, QA3

QA5: Efficiency: The server should send a response within a minimum of 2 seconds and a maximum of 10 seconds. At least 25% of the processing capacity and RAM available to the program must be idle.

Priority Level: Low
Precondition: N/A
Cross-references: QA1

QA6: Flexibility: The software should be maintainability to 10% change and can produce additional outputs using existing data in the database.

Priority Level: High
Precondition: N/A
Cross-reference: QA8

QA7: Robustness: The software should be able to handle unexpected situations and recover from errors or failures 2-3 seconds and smoothly.

Priority Level: Low
Precondition: N/A
Cross Reference: QA6

4.3 System Interface

The Friendly Ride Share project's user interface and user experience (UI/UX) have been meticulously designed using Figma, ensuring a visually appealing and intuitive platform. The design focuses on seamless navigation, accessibility, and an engaging experience for both riders and ride sharers.

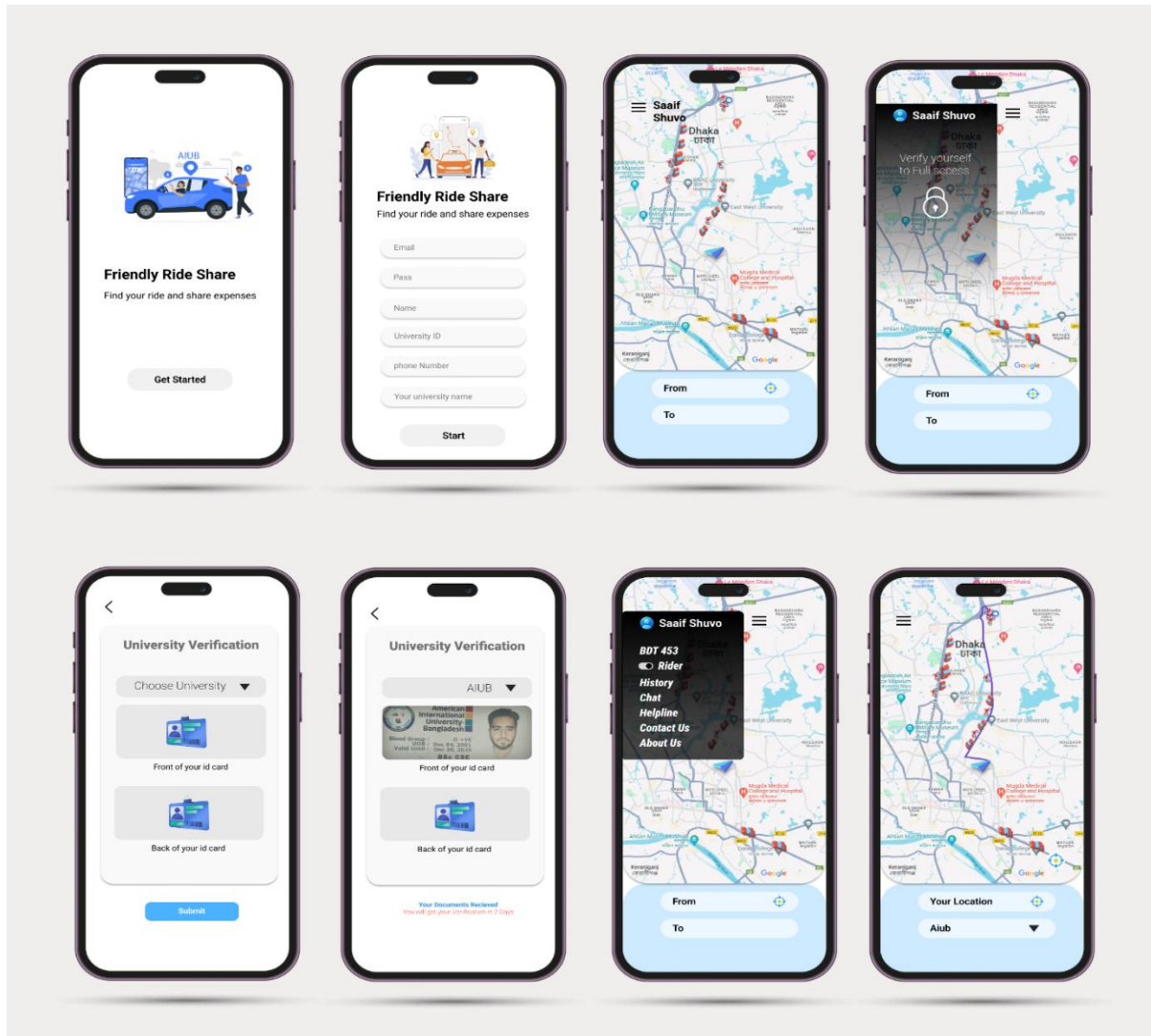


Figure 03: System Interface

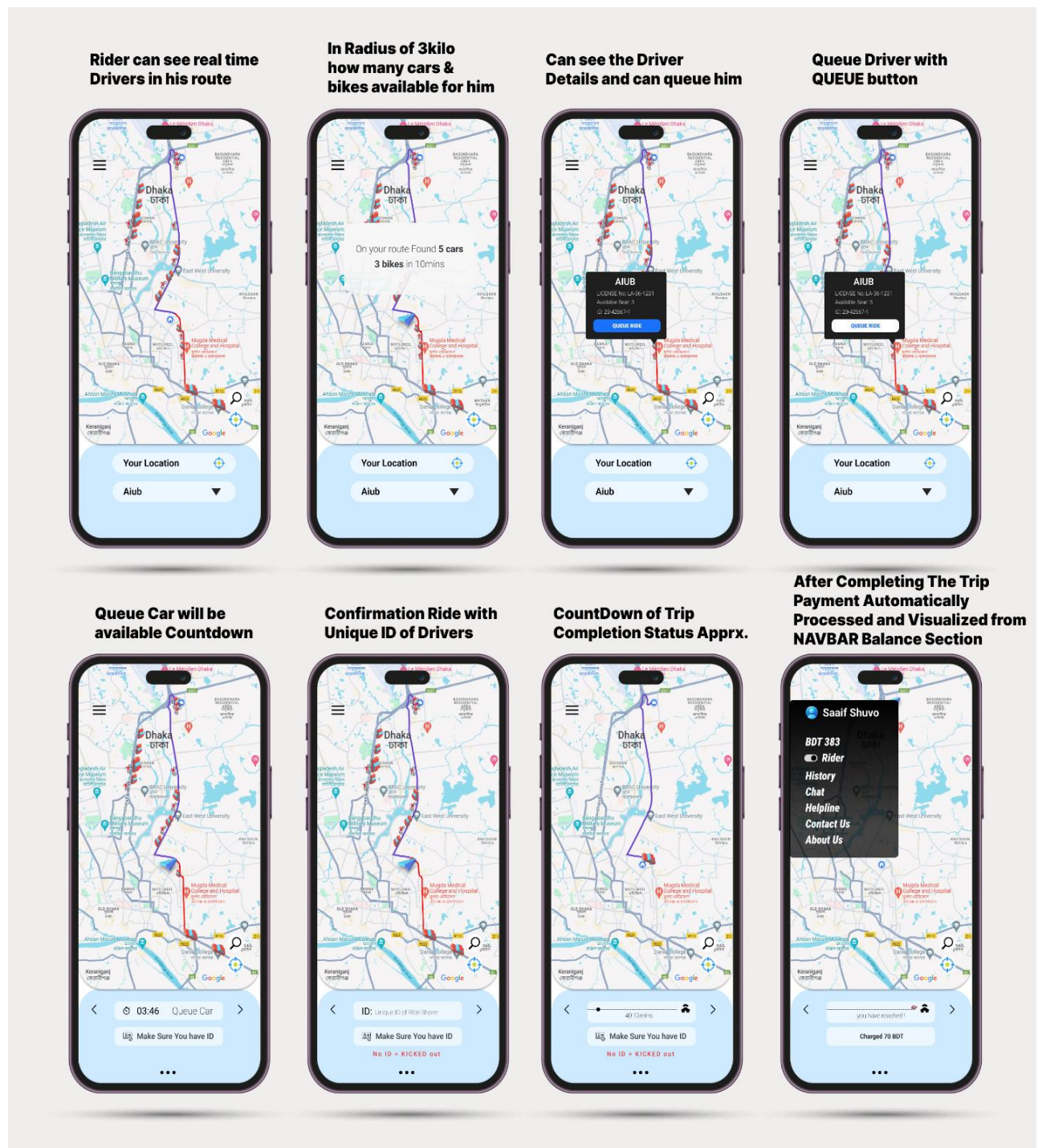


Figure 04: System Interface

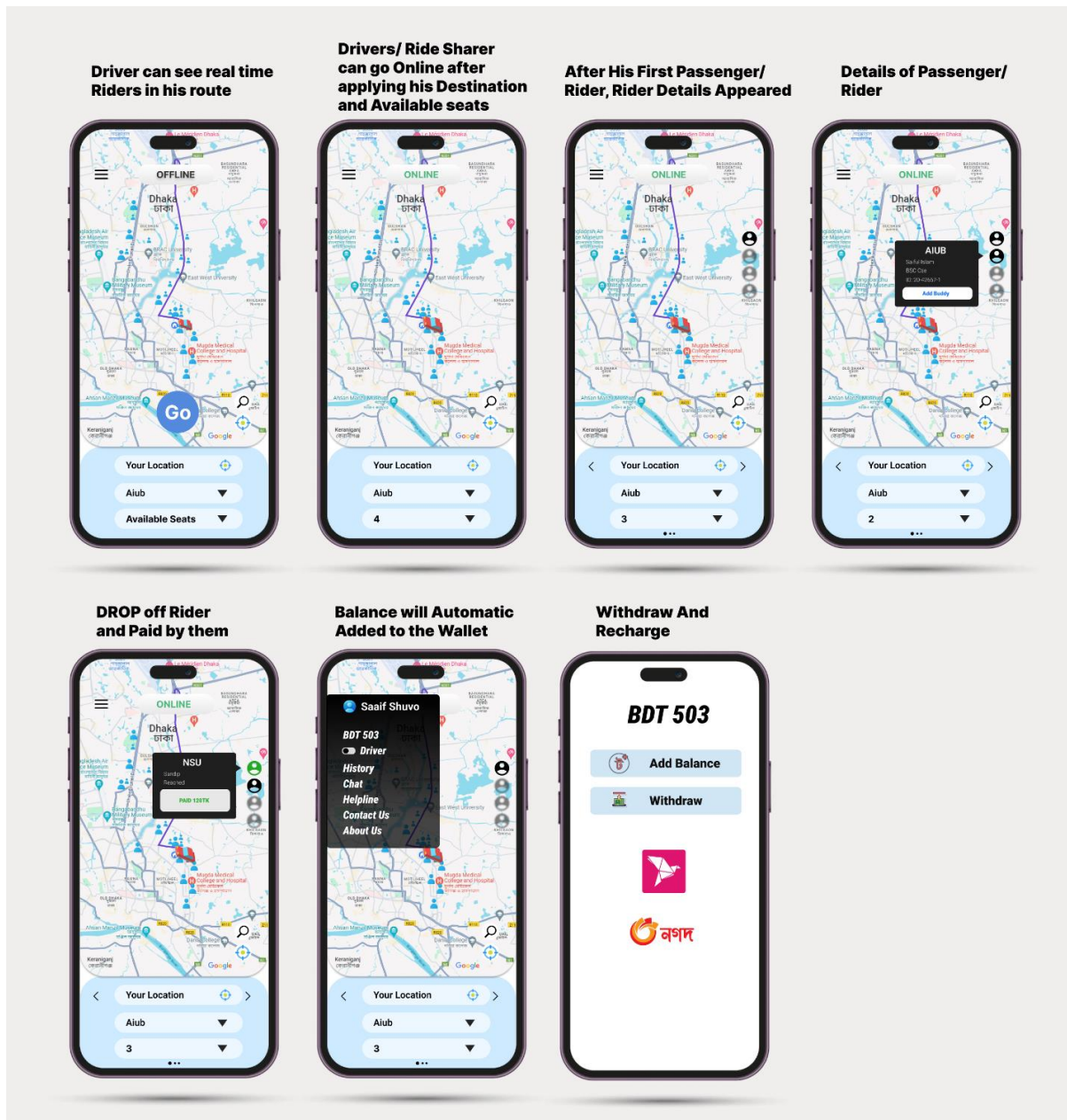


Figure 05: System Interface

4.4 Project Requirements

Tool: The test tools to be used is Selenium IDE. Selenium IDE (Integrated Development Environment) is primarily a record/run tool that a test case developer uses to develop Selenium Test cases. Selenium IDE is an easy-to-use tool from the Selenium Test Suite and can even be used by someone new to developing automated test cases for their web applications. The system developer needs PHP, JavaScript to develop Backend Application and run it on Apache server and to run PHPMyAdmin SQL server for MySQL database, CSS to develop Frontend Application with any type of code editor (Visual studio).

Budget: The project requires the development of a software solution within a budget of 19 lakh taka. The software must meet the specified requirements and deliverables, and the rest will be determined through consultation with the stakeholders.

Time: The project timeline is 9 months. Adequate planning, execution, and completion will be crucial, with frequent progress updates to ensure that the project remains on course.

Maintenance and support: The system must be designed to be easy to maintain and supported to ensure that any issues can be quickly and efficiently resolved.

4.3.1 Resources:

Human Resource:

- Project Manager: 1 person
- Software Developer: 5 persons
- Software Tester: 3 persons

Hardware/Networking Device:

- Personal Computer: 6 desktops and 2 laptops
- Router: 1
- Repeater: 1
- Server: 1
- Switch: 1

Reusable Components:

- Sub-system or Module
- Database Management System

Effort Estimation:

Let's, assume our project is an organic project, where $\text{Coefficient}_{\langle \text{Effort Factor} \rangle} = 2.4$, $P = 1.05$, $T = 0.38$ according to the COCOMO model

The estimated size of the software product in Source Lines of Code is 10000.

We know that $= (\text{SLOC}/1000) = (10,000/1000) = 10$.

Estimation of Development Effort, $PM = \text{Coefficient}_{\langle \text{Effort Factor} \rangle} * (\text{SLOC}/1000)^P$

$$= 2.4 \times (10)^{1.05}$$

$$= 26.928$$

$$= 4308.48 \text{ Person-Hours}$$

Estimation of Development Time, $DM = 2.50 * (PM)^T$

$$= 2.50 \times (26.928)^{0.38}$$

$$= 8.738 \text{ Months}$$

= 9 months (Aprox.)

Required number of people = $ST = PM/DM$

$$= 26.928/8.738 = 3.082$$

Let's assume the average salary of each employee in our company is TK 10,000 and they work 8 hours each day.

Therefore, Rate of per hour = $10,000 \div (3.082 \times 8)$

$$= \text{TK } 405.58 \text{ per hour}$$

Cost = Effort \times Rate = 4308.48 Person-Hours \times TK 405.58 per hour

$$= 1747433.318 \text{ Taka}$$

Suppose other necessary cost will be 1, 00,000 taka

Total cost = $(1747433.318 + 1, 00,000)$ taka

$$= 1847433.318 \text{ taka.}$$

$$= 1900000.000 \text{ taka. (Aprox.)}$$

5. FEATURES NOT TO BE TESTED

- **User interface design:** While the functionality of the application may be tested, the specific design of the user interface may not be a focus of testing efforts.
- **Network latency:** If the application relies on an internet connection to function, testing for network latency or internet connectivity issues may not be a primary focus.
- **Mobile device hardware:** The application may be designed to run on a specific type of mobile device (e.g., iPhone, Android), but the hardware itself is not within the scope of this project.
- **All Feature will test.**

6. TESTING APPROACH

6.1 Testing Levels

UNIT Testing: During our unit testing, tests will be conducted on individual software modules, and any errors will be checked. This testing will be carried out by the developer and approved by the development team leader. The goal will be to verify that each unit of code functions as intended. Unit testing is a white box technique where the software's internal structure, design, and coding will be tested to verify input-output flow and improve design, usability, and security.

Integration Testing: After unit testing, integration testing will be performed. Unit testing uses modules for testing purposes, and these modules will be combined and tested in integration testing. The software will be developed with a number of software modules that will be coded by different coders or programmers. The goal of integration testing will be

to check the correctness of communication among all the modules. The Bottom-Up technique will be used in our software.

System Testing: After completing integration testing, system testing will be proceeded with, where the fully integrated system will be thoroughly tested to ensure it meets all the specified requirements. This testing will be done using the "Black Box Testing" technique, as it will focus on the system's overall functionality without examining the internal code or structure. The goal will be to verify that the system functions as a complete, integrated unit and meets all the requirements.

Acceptance Testing: Then, Acceptance testing will be conducted, which is a testing technique performed to determine whether the software system has met the requirement specifications. The main purpose of this test will be to evaluate the system's compliance with the business requirements and verify if it has met the required criteria for delivery to end users.

6.2 Test Tools

The testing tools Selenium IDE will be used when our testers test the whole system. Selenium IDE (Integrated Development Environment) is primarily a record/run tool that a test case developer uses to develop Selenium Test cases. Selenium IDE is an easy-to-use tool from the Selenium Test Suite and can even be used by someone new to developing automated test cases for their web applications. In Unit testing the system developer needs PHP, JavaScript to develop Backend Application and run it on Apache server and to run PHPMyAdmin SQL server for MySQL database, CSS/NodeJs to develop Frontend Application with NextJs and any type of code editor (Visual studio). UsabilityHub will be used when our testers test the user acceptance testing. UsabilityHub is a remote user testing and research platform that provides valuable feedback and allows making data-driven decisions. It offers tools designed to identify bugs and issues during user acceptance testing.

6.3 Meetings

We will meet multiple times to present our project accurately. To monitor our progress and quickly identify error trends and bugs/issues, we will communicate via MS Teams and Google Meet three times. During our meetings, the test team leader will also meet with other project testers to assess overall progress. In these meetings, we will resolve our project's bugs and defects, ensuring its accuracy and a defect-free outcome. It will be one week since the first draft. Additional meetings could be held in the future if there is a crisis.

Table 1: Meetings details

| Objective | Meeting Criteria | Meeting Date |
|---|---|--------------|
| ○ Analysis | ○ Functions Analysis ○ Analysis the working process | 10-11-2024 |
| ○ Projects Progress | ○ How far the work of the project has been completed | 28-11-2024 |
| ○ Error/Defects checking. ○ Revise the full system | ○ Checks Error/defect ○ Run the system. ○ Revise the whole system | 17-12-2024 |

7. TEST CASES/TEST ITEMS

Table 2: Test Case for Customer Login

| | | | | |
|--|---|--|----------------|--------------------|
| Project Name: Friendly Ride Share | | Test Designed by: Sandip Misra | | |
| Test Case ID: FR_1 | | Test Designed date: 25/11/2024 | | |
| Test Priority (Low, Medium, High): High | | Test Executed by: | | |
| Module Name: Customer Login | | Test Execution date: | | |
| Test Title: Verify login with valid username and password | | | | |
| Description: Test website login page | | | | |
| Precondition (If any): Customer must have valid username and password | | | | |
| Test Steps | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1. Go to the website. 2. Enter username. 3. Enter password. 4. Click Login | Username: 21445061 Password: 506 | User should login into the application | | |
| Post Condition: User is validated with database and successfully login to account. The account session details are logged in the database. | | | | |

Table 3: Test Case for Customer Sign Up

| | | | | | |
|--|--|---|--|----------------|--------------------|
| Project Name: Friendly Ride Share | | | Test Designed by: Sandip Misra | | |
| Test Case ID: FR_2 | | | Test Designed date: 25/11/2024 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Customer Sign Up | | | Test Execution date: | | |
| Test Title: Verify Sign UP with valid information | | | | | |
| Description: Test the Sign-Up page | | | | | |
| Precondition (If any): User must have valid information. | | | | | |
| Test Steps | | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1. Go to the Sign-up page. 2. Enter valid Information. 3. Click submit | | Name: Sandip Mobile No: 0171212156 Institute: AIUB Username: 21445061 Password: 506 | User should sign up into the application | | |
| Post Condition: User is validated with database and successfully login to account. The account session details are logged in the database. | | | | | |

Table 4: Test Case for Browse Ride sharers / Riders.

| | | | | |
|---|-----------------|---|--------------------------------|--------------------|
| Project Name: Friendly Ride Share | | | Test Designed by: Sandip Misra | |
| Test Case ID: FR_3 | | | Test Designed date: 26/11/2024 | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: | |
| Module Name: Browse Ride sharers / Riders | | | Test Execution date: | |
| Test Title: Find ride sharers or riders | | | | |
| Description: Test the system of browse riders / ride sharers | | | | |
| Precondition (If any): Database should consist of all riders/ ride sharers data. | | | | |
| Test Steps | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1. Go for browse riders or ride sharers. 2. Select location. 3. Click browse | Location: Kuril | Bring in all the riders and make it visible | | |
| Post Condition: User is validated with database and successfully login to account. The account session details are logged in the database. Database should consist of all riders / ride sharers data. | | | | |

Table 5: Test Case for View Ride sharers / Riders Details

| | | | | |
|---|---|---|--------------------------------|--------------------|
| Project Name: Friendly Ride Share | | | Test Designed by: Sandip Misra | |
| Test Case ID: FR_4 | | | Test Designed date: 27/11/2024 | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: | |
| Module Name: View Ride sharers / Riders Details | | | Test Execution date: | |
| Test Title: Test the Ride sharers / Riders’ detail’s view | | | | |
| Description: Test website login page | | | | |
| Precondition (If any): Database should consist of all user details. | | | | |
| Test Steps | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1. Select Riders / Ride sharers. 2. Click view | Validate the display of personal information. | Bring in all the data and make it visible | | |
| Post Condition: Database should consist of all user details. | | | | |

Table 6: Test Case for Connect Ride sharers & Riders.

| | | | | |
|---|-------------------------------------|---------------------------------|--------------------------------|--------------------|
| Project Name: Friendly Ride Share | | | Test Designed by: Sandip Misra | |
| Test Case ID: FR_5 | | | Test Designed date: 03/12/2024 | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | |
| Module Name: Connect Ride sharers & Riders | | | Test Execution date: | |
| Test Title: Test the connect Ride sharers & Rider’s system. | | | | |
| Description: Test the connection between Ride sharers & Rider’s system. | | | | |
| Precondition (If any): User must have login and database should consist of all user details. | | | | |
| Test Steps | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1. Select riders. 2. Click connect. | Rider / Ride sharers Name: SaifulSS | User should connect the riders. | | |
| Post Condition: User is validated with database and successfully login to account. The account session details are logged into the database. Database should consist of all user details. | | | | |

Table 7: Test Case for Confirm Ride

| | | | | |
|--|---------------------|-------------------------------|--------------------------------|--------------------|
| Project Name: Friendly Ride Share | | | Test Designed by: Sandip Misra | |
| Test Case ID: FR_6 | | | Test Designed date: 04/12/2024 | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | |
| Module Name: Confirm Ride | | | Test Execution date: | |
| Test Title: Test the Confirm Ride. | | | | |
| Description: Test website Confirm Ride system. | | | | |
| Precondition (If any): User must have login and database should consist of all user details also select ride details. | | | | |
| Test Steps | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1. Check the Ride Details 2. Click Confirm | Valid ride request. | User should confirm the ride. | | |
| Post Condition: User is validated with database and successfully login to account. The account session details are logged into the database. Database should consist of all user details also select ride details. | | | | |

Table 8: Test Case for Payment

| | | | | |
|--|--------------------------------------|----------------------------------|--------------------------------|--------------------|
| Project Name: Friendly Ride Share | | | Test Designed by: Sandip Misra | |
| Test Case ID: FR_7 | | | Test Designed date: 04/12/2024 | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | |
| Module Name: Payment | | | Test Execution date: | |
| Test Title: Test the payment. | | | | |
| Description: Test website payment system. | | | | |
| Precondition (If any): User must have login and database should consist of all user details also select ride details. | | | | |
| Test Steps | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1. Select the payment method. 2. Enter amount. 3. Click Confirm | Payment method: Bkash Amount: 200 | User should confirm the Payment. | | |
| Post Condition: User is validated with database and successfully login to account. The account session details are logged into the database. Database should consist of all user details also select ride details. | | | | |

Table 9: Test Case for Notifications

| | | | | |
|--|-------------------------------------|-----------------------------------|--------------------------------|--------------------|
| Project Name: Friendly Ride Share | | | Test Designed by: Sandip Misra | |
| Test Case ID: FR_8 | | | Test Designed date: 05/12/2023 | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | |
| Module Name: Notifications | | | Test Execution date: | |
| Test Title: Test the notifications. | | | | |
| Description: Test website notification system. | | | | |
| Precondition (If any): User must have login and database should consist of all user details also select ride details. | | | | |
| Test Steps | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1. Press Login 2. Confirm Login 3. Click the notification button. | Username: 21445061 Password: 506 | User should see all notification. | | |
| Post Condition: User is validated with database and successfully login to account. The account session details are logged into the database. Database should consist of all user details also select ride details. | | | | |

Table 10: Test Case for usability

| | | | | |
|--|-------------------------------------|---------------------------------------|--------------------------------|--------------------|
| Project Name: Friendly Ride Share | | | Test Designed by: Sandip Misra | |
| Test Case ID: FR_9 | | | Test Designed date: 05/12/2024 | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | |
| Module Name: Usability | | | Test Execution date: | |
| Test Title: Test the usability. | | | | |
| Description: Test website usability. | | | | |
| Precondition (If any): User must have login. | | | | |
| Test Steps | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1. Login to the system. 2. Use this apps. | Username: 21445061 Password: 506 | Response to the user from 2-5 second. | | |
| Post Condition: User is validated with database and successfully login to account. The account session details are logged into the database. | | | | |

Table 11: Test Case for Availability

| | | | | |
|---|-------------------------------------|---|--------------------------------|--------------------|
| Project Name: Friendly Ride Share | | | Test Designed by: Sandip Misra | |
| Test Case ID: FR_10 | | | Test Designed date: 06/12/2024 | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | |
| Module Name: Availability | | | Test Execution date: | |
| Test Title: Test the availability. | | | | |
| Description: Test website availability. | | | | |
| Precondition (If any): Valid access to the system's ride allocation module. Availability of both ride requests and available drivers in the system | | | | |
| Test Steps | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1. Login to the system. 2. Use this apps | Username: 21445061 Password: 506 | System should be available for 24 hours/days with 30 minutes down time per day. | | |
| Post Condition: Ensure that rides are allocated accurately and promptly. Review system logs for any errors or inconsistencies during ride allocation. | | | | |

Table 12: Test Case for Security

| | | | | |
|---|-----------------------------|--|--------------------------------|--------------------|
| Project Name: Friendly Ride Share | | | Test Designed by: Sandip Misra | |
| Test Case ID: FR_11 | | | Test Designed date: 06/12/2024 | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | |
| Module Name: Security | | | Test Execution date: | |
| Test Title: Test the Security. | | | | |
| Description: Test website Security. | | | | |
| Precondition (If any): Access to the system's user authentication module. Test environment set up with valid test user credentials. | | | | |
| Test Steps | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1. Input SQL injection code in the login fields. | Valid user ID and password. | System detects and rejects SQL injection attempts, maintaining system integrity and security | | |
| Post Condition: Ensure that the system returns to a secure state after testing. Log out from the system to clear user session data. | | | | |

8. ITEM PASS/FAIL CRITERIA

Outlining the PASS/FAIL criteria for the tests contained in this project is the main objective of this part. Here conducted total number of 6 tests cases. If a system or unit receives a score between 93% and 96%, it will be regarded to have met the pass criterion, and if it receives a score below 80%, it will have failed. To assess how trustworthy and user-satisfying project is, employed this metric.

9. TEST DELIVERABLES

1. Test plan: A document that outlines the testing approach, test cases, and schedules for testing.
2. Test cases: Documents that describe individual test scenarios and the expected results. Such as unit testing. Integration testing, system testing, acceptance testing.
3. Test scripts: Automated scripts that execute test cases.
4. Test results: Reports that show the results of individual tests and overall testing progress.
5. Defect reports: Documents that report defects or issues discovered during testing.
6. Test summary reports: A document that summarizes the testing process and results, including pass/fail rates, defect counts, and testing metrics.
7. Test environment setup documentation: A document that describes the configuration of the testing environment, including hardware, software, and network configurations.

8. Test completion report: A document that describes the overall results of testing and provides a summary of the testing process, including any issues encountered and recommendations for future testing.

10. STAFFING AND TRAINING NEEDS

Staffing Needs:

- Test Manager: A test manager who will be responsible for overseeing the testing activities, managing the testing team, and communicating the progress and results of testing to the project stakeholders.
- Test Engineers: Test engineers who will be responsible for executing the test cases, documenting defects, and reporting on the results of the testing activities.
- Automation Engineer: An automation engineer who will be responsible for developing and maintaining the automated test scripts used to execute the test cases.

Training Needs:

- Testing Methodologies: All testing team members need to be trained in the various testing methodologies, including manual and automated testing, to ensure that they can execute the test cases effectively.
- Communication Skills: The test manager and quality analyst should be trained in effective communication skills to ensure that they can communicate the progress and results of the testing activities to the project stakeholders effectively.

11. RESPONSIBILITIES

Table 13: Responsibilities

| | Test Manager | Project Manager | Dev Team | Test Team | Client |
|---|--------------|-----------------|----------|-----------|--------|
| Acceptance test Documentation & Execution | ✓ | ✓ | | ✓ | ✓ |
| System/Integration test Documentation & Execution | ✓ | | ✓ | ✓ | |
| Unit test documentation & execution | ✓ | | ✓ | ✓ | |
| System Design Review | ✓ | ✓ | ✓ | ✓ | ✓ |
| Detail Design Review | ✓ | ✓ | ✓ | ✓ | |
| Test procedure and rules | ✓ | ✓ | ✓ | ✓ | |
| Screen & Report prototypes | | | ✓ | ✓ | ✓ |
| Change control and regression testing | ✓ | ✓ | ✓ | ✓ | ✓ |

12. TESTING SCHEDULE

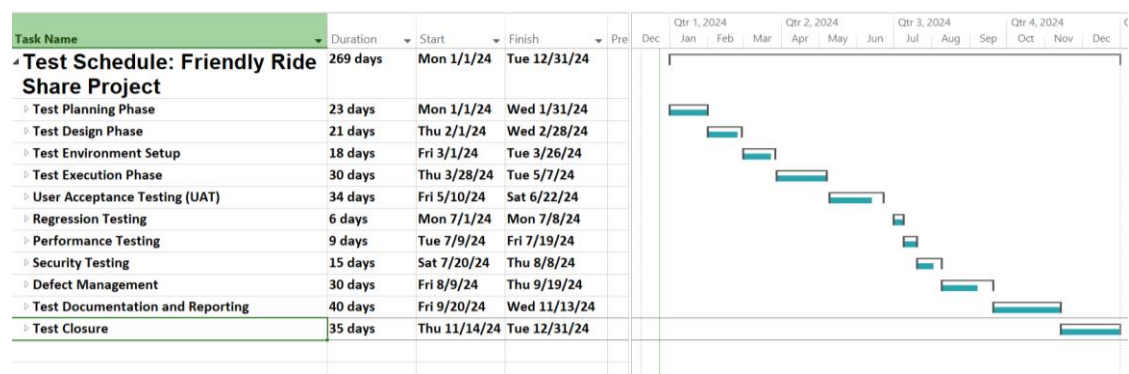


Figure 06: Testing Schedule

13. PLANNING RISKS AND CONTINGENCIES

Table 14: Risk Mitigation Plan for testing

| S/N | Risk Description | Probability | Impact | Mitigation Plan |
|-----|--------------------------|-------------|--|--|
| 1 | Changes in requirements | 20% | Quality of the testing | Informed the specific changes to the testing team |
| 2 | Communication breakdowns | 20% | Misunderstandings, delays, or errors in testing. | Establish regular communication channels, make rules clear, and have a backup plan |
| 3 | Resource constraints | 15% | Slow down testing work. | Create a backup plan to focus on important testing and use automation. |
| 4 | Technical issues | 15% | Disrupt testing, causing delays and data loss. | Plan for quick data recovery, use backup systems, monitor networks proactively, and have a quick response plan for crashes |
| 5 | Exceeding budget | 30% | Can harm the trust of the client and the relationship. | Take some extra money from client for safety. |

14. APROVALS

Table 15: Approval

| No | Name | Designation | Signature | Approval |
|----|-----------------|------------------|-----------|----------|
| 1 | Saiful Islam | Project Manager | Saiful | Approved |
| 2 | Arghyajit Sadhu | Business Analyst | Arghyajit | Approved |
| 3 | Pranto Sen | Team Manager | Pranto | Approved |
| 4 | Sandip Misra | Test Manager | Sandip | Approved |