RentMyRide

Introduction:

1.Purpose: The purpose of this Software Requirements Specification (SRS) document is to provide a comprehensive overview of the Bikes & Scooters Rental System, its features, functionality, and the requirements necessary for its development. This document serves as a critical reference for all stakeholders involved in the project, including developers, testers, project managers, and clients.

2. Scope: The vehicles Rental System is designed to offer a convenient and efficient solution for renting different types of vehicles to users within a specified geographical area. It aims to provide an easy-to-use platform that allows users to browse, reserve, and rent vehicles for short-term use, promoting sustainable urban transportation options.

This document outlines the system's scope, including the key features, user classes, and the high-level architecture. It also describes the external interfaces, non-functional requirements, and other relevant information necessary for the successful development and deployment of the system.

3. Document Conventions: Throughout this document, we will use the following conventions:

Requirement ID: Requirements are uniquely identified using an alphanumeric code, e.g., "REQ-001."

Priority: Requirements are categorized as "Critical," "High," "Medium," or "Low" based on their importance and impact on the system.

NFR: Non-functional requirements are marked with "NFR" to distinguish them from functional requirements.

User: Refers to any user of the Bikes & Scooters Rental System, including customers, administrators, and support staff.

Administrator: Refers to users with administrative privileges responsible for managing the system.

4. Intended Audience:

This document is intended for the following audience:

Development Team: Developers, architects, and engineers responsible for designing, implementing, and testing the system.

Project Managers: Those responsible for project planning, scheduling, and resource allocation. Quality Assurance Team: Testers and QA professionals who will validate and verify that the system meets the specified requirements.

Clients and Stakeholders: The individuals or organizations funding and overseeing the project.

Overall Description:

1. Product Perspective : The Bikes & Scooters Rental System is a web-based platform designed to facilitate the rental of bicycles and scooters within a specified urban or suburban area. The system is an independent application that interfaces with various external components, including GPS services, payment gateways, and a fleet management system.

2. System Interfaces: The Bikes & Scooters Rental System interacts with the following external components:

GPS Services: The system uses GPS data to track the location of

available bicycles and scooters, enabling users to locate and reserve them.

Payment Gateway: Integration with a secure payment gateway allows users to make online payments for rentals.
Fleet Management System: Real-time information about the status, availability, and maintenance needs of the fleet is obtained through this interface.

User Interfaces : The system provides user interfaces accessible via web browsers and mobile devices. These interfaces are designed to be intuitive and user-friendly, allowing customers to browse available vehicles, make reservations, and complete rental transactions.

Hardware Interfaces: The Bikes & Scooters Rental System relies on the hardware components of users' devices, such as smartphones, tablets, and computers, to access the web-based platform. Additionally, it may integrate with hardware used for fleet tracking and management.

Software Interfaces: The system interfaces with external software components, including the payment gateway, GPS services, and the fleet management system, through well-defined APIs and communication protocols.

Product Functions: The Vehicles Rental System provides the following key functions:

1.User Registration and Authentication: Users can create accounts, log in, and manage their profiles.

2.Vehicle Search and Reservation: Users can search for available bicycles and scooters in their vicinity, view details, and reserve them.
3.Rental Transactions: Users can initiate, manage, and complete rental transactions, including payment processing.

4.Real-Time Tracking: The system offers real-time tracking of rented

vehicles, enabling users to locate them during their rental.

5.Maintenance Alerts: The system generates maintenance alerts based on the status and condition of the fleet.

6.Administrative Functions: Administrators can manage user accounts, monitor system health, and access reporting tools.

User Classes and Characteristics: The Bikes & Scooters Rental System is designed to serve the following user classes:

1.Customers: These are the primary users of the system, individuals who register, search for, and rent bicycles and scooters.

2.Administrators: Administrators have privileged access to the system, allowing them to manage user accounts, monitor system performance, and resolve issues.

3.Support Staff: Support staff assists users with inquiries, issues, and emergencies through a customer support interface.

3. Functional Requirements:

User Registration and Authentication:

1. User Registration: The system shall allow new users to register by providing their personal information, including name, email address, and password. User registration shall require email verification to activate the

account.

2. User Login: Registered users shall be able to log in using their email and password. Users shall have the option to log in through third-party authentication methods, such as social media accounts (e.g., Google or Facebook).

Profile Management:

1 User Profile: Registered users shall have access to a user profile page where they can edit their personal information, including name, contact details, and profile picture. Users shall be able to set their notification preferences and communication settings.

Find Nearby Riders:

1. Location Tracking: The system shall access the user's device location to identify nearby riders and available bicycles or scooters. Location data shall be updated in real-time.

2. Rider Search: Users shall have the ability to search for nearby riders based on criteria such as distance, vehicle type, and availability.

Real-Time Ride Tracking :

1. Ride Initiation: Users shall be able to start a ride by selecting an available bicycle or scooter. The system shall record the ride's start time

and location.

2. Real-Time GPS Tracking: During a ride, the system shall track the user's location and display it on a map interface. Users shall have the option to share their real-time ride progress with friends or contacts.

Fare Estimation:

1. Fare Calculation: The system shall calculate the fare for a ride based on factors such as distance traveled, ride duration, and vehicle type. Fare estimates shall be provided to users before they confirm a ride.

In-App Transaction:

1. Ride Payment : Users shall be able to make payments for their rides directly through the application. The system shall support various payment methods, including credit/debit cards, mobile wallets, and digital payment platforms.

2. Payment Confirmation: After completing a ride, users shall receive a payment confirmation with a detailed receipt.

Payment Methods:

1. Payment Options: The system shall offer multiple payment methods, including credit/debit cards, PayPal, and in-app wallet payments. Users shall have the ability to add, update, or remove payment methods from their profiles.

Ride History

1. Ride Records: The system shall maintain a comprehensive history of

users' past rides, including ride start and end times, locations, distance, fare details, and payment records. Users shall have access to their ride history for reference.

Performance Requirements

1. Response Time: The system shall respond to user interactions (e.g., ride requests, payment processing) within 5 seconds on average under normal load conditions.

2. Availability : The system shall maintain an availability rate of at least 99.9%, with planned maintenance windows communicated in advance.

3. Scalability : The system architecture shall support scaling to accommodate up to 10,000 concurrent users during peak usage periods without significant degradation in performance. These functional requirements outline the specific capabilities and behaviors expected from the Bikes & Scooters Rental System in relation to user registration, authentication, profile management, ride tracking, payment, and other key features. Tailor these requirements further to meet your project's unique needs and priorities

Non-functional Requirements:

1. Performance: The system shall respond to user interactions within five seconds on average under normal load conditions.
Throughput: The system should support a minimum of 1000 concurrent users performing typical actions simultaneously. The system architecture must be scalable to accommodate up to 10,000 concurrent users during peak usage periods without significant degradation in performance.

2. Security: All data transmitted over the network, including user credentials and payment information, shall be encrypted using industry-standard encryption algorithms.

3. Reliability: The system shall maintain an availability rate of at least 99.9%, with planned maintenance windows communicated in advance.

4. Usability: The user interface shall be designed to be intuitive and user-friendly to ensure a positive user experience.

5. Compatibility: Cross-Platform Compatibility: The system shall be compatible with major web browsers (e.g., Chrome, Firefox, Safari) and mobile platforms (iOS and Android).

6 Compliance: Legal and Regulatory Compliance: The system shall adhere to all applicable data protection and transportation regulations.

7 Data Privacy: The system shall implement stringent data privacy measures to protect user information and ensure compliance with data privacy laws.

8 Performance Monitoring: The system shall include real-time performance monitoring tools to proactively identify and address performance issues.

9. Customer Support: Customer Support: A 24/7 customer support channel shall be available to assist users with inquiries, issues, and emergencies.

External Interface Requirements:

1. User Interfaces: Describe the user interfaces the system will have, including their design, layout, and usability considerations.
Specify any hardware or software requirements for user interfaces, such as supported browsers or devices.

2. Hardware Interfaces: Detail any hardware components the system will interact with, including specifications, protocols, and connection methods. Specify any compatibility requirements with external hardware devices.

3.Software Interfaces: List and describe any external software systems, databases, or APIs that the Bikes & Scooters Rental System will need to integrate with.
Include information about data formats, protocols, and communication methods.

4 Communication Interfaces: Describe how the Bikes & Scooters Rental System will communicate with external systems or services. Specify communication protocols, encryption requirements, and security measures.

Budget:

Available budget:Not yet Decided.

Development cost:25000-30000 per month

Hosting cost:1000-1500 taka per month

Quality Assurance Testing:15000-20000 taka

Maintenance Cost:3000-4000 taka per year

Total Cost:56000 taka

Timeline:

Month 1-2: Planning and Research

- Project scope and objectives.
- Research the ride-sharing market and competition.
- Identify target audience and user personas.
- Create a detailed project plan and budget.
- Secure initial funding if required.

Month 3-4: Design Phase

- Design the app's user interface (UI) and user experience (UX).
- Create wireframes and prototypes for testing.
- Develop branding, including the app logo and color scheme.
- Start building the app's database structure.

Month 5-6: Development

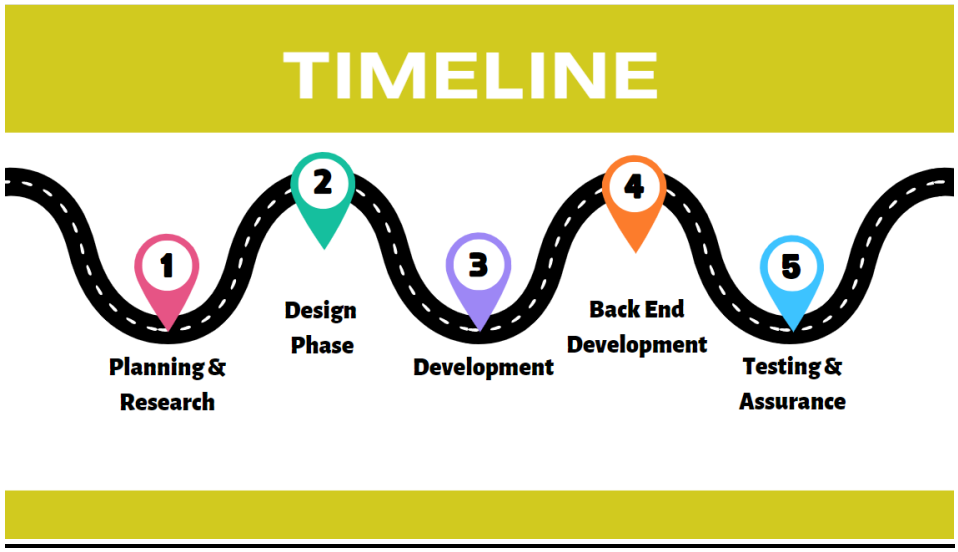
- Begin front-end development of the app.
- Implement core features, such as user registration and login.
- Develop the ride matching and booking system.
- Build the driver and passenger interfaces.
- Conduct regular testing and debugging.

Month 7-8: Back-End Development

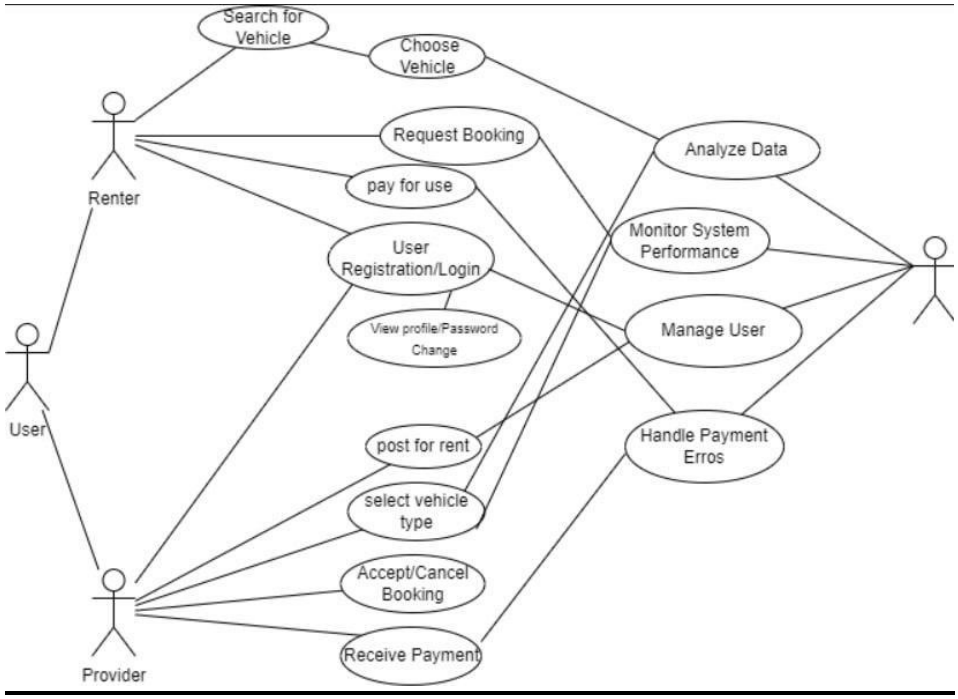
- Develop the server-side and database components.
- Implement payment processing and security features.
- Set up the server infrastructure and hosting.
- Conduct load testing to ensure scalability.

Month 9-10: Testing and Quality Assurance

- Perform thorough testing of all app functionalities.
- Address and fix any bugs or issues.
- Conduct beta testing with a select group of users.
- Gather feedback and make necessary improvements.



Case Diagram:



UML **Diagram:**

Formatted: Font color: Auto, Highlight