Software Requirements Specification (SRS)

Project Title: MyShuttle – Smart Shuttle Booking App

Prepared by:

- Elangovan

- Gayathri

- Teja

1.Introduction

1.1 Overview

This document is organized into three main sections, followed by appendices. Section 1 provides an introduction, scope, and overview of the MyShuttle application. Section 2 offers a high-level description of the product, its users, constraints, and dependencies. Section 3 details all specific requirements, including external interfaces, functions, performance, database, and other software attributes. The appendices provide supporting information, including diagrams.

1.2 Purpose:

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed description of the functional and non-functional requirements for the MyShuttle project. MyShuttle is a transportation application designed to facilitate users with shuttle services through an easy-to-use mobile platform. It allows users to book rides, track shuttles in real-time, manage payment options, and communicate with the shuttle providers.

1.3 Scope:

MyShuttle aims to provide a user-friendly shuttle service solution that operates on mobile devices (iOS and Android). The application will enable users to:

- Book shuttle rides.

- View real-time shuttle locations and estimated arrival times.

- Manage user accounts, profiles, and preferences.

- Process payments securely.

- Receive notifications regarding shuttle status and promotions.

1.4 Definitions, Acronyms, and Abbreviations:

Shuttle: A vehicle that provides transport between designated points, often on a fixed route or schedule.

- User: A person who books or uses the shuttle service.

- Admin: A person who manages the shuttle service operations.

- Driver: A person who drives the shuttle.

- App: MyShuttle mobile application for booking and tracking shuttle services.

1.5 References

- Industry standards for mobile application design and user experience.

- API documentation for payment gateway (e.g., Stripe, PayPal).

- Real-time location services API (e.g., Google Maps API).

2.Overall Description

2.1 Product Perspective

The MyShuttle application is a self-contained system that will consist of three main components: a User App, an Admin Panel, and a Driver App.

- User App: For users to book and track shuttle rides.

- Admin Panel: For administrators to manage shuttle schedules, driver assignments, and user data.

- Driver App: For drivers to receive ride assignments, track routes, and communicate with passengers and admins.

  The system will operate in a cloud-based environment (e.g., AWS or Google Cloud) to handle database management, user authentication, payment processing, and real-time location tracking. The mobile applications will be compatible with both Android and iOS operating systems.

2.2 Product Features

The key features of MyShuttle include:

- User Registration and Authentication: Users can create accounts, log in, and securely access their profiles.

- Ride Booking: Users can input pickup and drop-off locations, select a shuttle route, and confirm bookings.

- Real-Time Shuttle Tracking: Users can track the shuttle's real-time location and estimated time of arrival.

- Payment Integration: Secure payment processing for bookings.

- Notifications: Push notifications for ride confirmations, updates, delays, and promotions.

- Admin Dashboard: Admins can monitor shuttle availability, assign drivers, and manage

customer interactions.

2.3 User Classes and Characteristics

- End Users: Individuals who require shuttle services. They are likely to have smartphones and a basic understanding of mobile applications.

- Drivers: Individuals operating the shuttles. They will use the Driver App to receive ride assignments and communicate with users.

- Administrators: Employees managing shuttle operations. They will use the Admin Dashboard for oversight, scheduling, and management.

2.4 Operating Environment

- Mobile Application: Compatible with both Android and iOS operating systems.

- Backend Services: Cloud-based infrastructure (e.g., AWS or Google Cloud) for handling database management, user authentication, payment processing, and real-time location tracking.

- Driver App: Android and iOS versions for driver support.

2.5 Constraints

- Real-time Location Tracking: The application will depend on mobile devices' GPS capabilities and internet connectivity.

- Payment Gateway: The payment process is subject to the limitations and rules of third- party payment processors (Stripe, PayPal).

- Regulatory Compliance: The system must adhere to transportation laws and payment security regulations (e.g., GDPR, PCI DSS).

2.6 Assumptions and Dependencies

- Users will have internet access and mobile devices with GPS capabilities.

- The application will require regular updates for maintaining compatibility with new OS versions and APIs.

- Payment processing services will be available without interruptions.

2.6 Apportioning of Requirements

The initial release will include all core functions listed in Section 2.2. Advanced features such as a "Frequent Rider Program" or detailed "Campaign Management" analytics may be delayed until a future version (v2.0).

3.System Requirements

3.1 External Interface Requirements

- **User Interfaces:** The User App will feature a modern, mobile-first design with an intuitive layout for login, booking, tracking, and payment. The Admin Panel will be a web-based interface with data visualizations and dashboards.

- **Hardware Interfaces:** The application will use the mobile device's GPS for location tracking.

- **Software Interfaces:** The system will integrate with payment systems like Stripe and PayPal for transactions and use the Google Maps API for mapping and tracking.

3.2 Functions

3.2.1 User Registration and Authentication

- **Description:** Users can create a profile and authenticate using email, social media, or phone numbers.

- **Requirements:**

- FR1: Users must be able to register via email, phone number, or social login.

- FR2: The system shall validate and store user credentials securely.

- FR3: The system must provide a password reset option.

3.2.2 Ride Booking

- **Description:** Users can input ride details, select a shuttle, and confirm the booking.

- **Requirements:**

  - FR1: Users can enter pickup and drop-off addresses.

  - FR2: The system shall display available shuttles with estimated arrival times.

  - FR3: Users can confirm the booking, and a driver will be assigned.

3.2.3 Real-Time Shuttle Tracking

- **Description:** The app will display the real-time location of the assigned shuttle and its ETA.

- **Requirements:**

  - FR1: Users must be able to track the shuttle's location on a map.

  - FR2: The app shall update the ETA based on the shuttle's location and traffic data.

3.2.4 Payment Integration

- **Description:** Allows users to securely pay for services via integrated third-party payment systems.

- **Requirements:**

  - FR1: Users must be able to save payment information securely.

  - FR2: The app shall support credit/debit cards and digital wallet payments.

  - FR3: Users shall receive a receipt upon successful payment.

3.2.5 Notifications

- **Description:** Push notifications inform users, drivers, and admins about ride status and other updates.

- **Requirements:**

  - FR1: Users shall receive notifications about shuttle arrivals, delays, and promotions.

- FR2: Admins and drivers shall receive notifications about assignments and ride completions.

3.2.6 Admin Dashboard

- **Description:** Allows administrators to manage users, view reports, and assign drivers.

- **Requirements:**

  - FR1: Admins must be able to view ride schedules and manage driver availability.

  - FR2: Admins can track payment records and issue refunds.

  - FR3: Admins can update shuttle routes and manage fleet maintenance schedules.

3.3 Performance Requirements

- The application must load within 2 seconds on an average 4G network connection.

- Real-time shuttle tracking updates must be reflected on the user interface within 30 seconds of the shuttle's location change.

3.4 Logical Database Requirements

The system shall maintain a database to store information related to users, drivers, shuttles, and rides.

- **Data Entities:** User Profile (User ID, name, contact info, encrypted payment details), Driver Profile (Driver ID, name), Ride History (Ride ID, route, fare), and Shuttle Fleet (Shuttle ID, capacity, current location).

- **Relationships:** Each ride must be linked to a User Profile and Driver Profile.

- **Integrity Constraints:** A User ID must be unique. A driver cannot be assigned to more than one active ride at the same time.

- **Data Retention:** Ride history must be retained for a minimum of 24 months.

3.5 Design Constraints

- **Real-time Location Tracking:** The application will depend on mobile devices' GPS capabilities and internet connectivity.

- **Payment Gateway:** The payment process is subject to the limitations and rules of third-party payment processors (e.g., Stripe, PayPal).

- **Regulatory Compliance:** The system must adhere to regional transportation laws and data protection standards like GDPR.

3.6 Software System Attributes

- **3.6.1 Security:** All user data must be encrypted in transit and at rest using secure protocols (e.g., SSL/TLS). The system must implement role-based access control (RBAC).

- **3.6.2 Availability:** The application should be available 24/7 with a target uptime of 99.9%. Backup and disaster recovery procedures must be in place.

- **3.6.3 Usability:** The application must be easy to navigate with a minimal learning curve. The UI should be responsive and adapt to various screen sizes.

- **3.6.4 Maintainability:** The software should be developed with modularity to allow for ease of maintenance and updates.

- **3.6.5 Compatibility:** The mobile app must be compatible with the latest and previous major versions of iOS and Android.

4.External Interface Requirements

4.1 User Interfaces

- User Interface (UI): A modern, mobile-first design with an intuitive layout. The application will include screens for login, booking, tracking, and payment processing.

- Admin Interface (UI): A web-based interface with data visualizations and dashboards to manage shuttles, users, and drivers.

4.2 Hardware Interfaces

- The application will use the mobile device's GPS for tracking shuttles.

- A third-party payment gateway will integrate for processing payments securely.

4.3 Software Interfaces

- Integration with payment systems such as Stripe and PayPal for processing transactions.

- Google Maps API for shuttle tracking and mapping.

5. Non-Functional Requirements

5.1 Performance Requirements

- The application must load within 2 seconds on average.

- Real-time shuttle tracking updates should be reflected on the user interface within 30 seconds.

5.2 Security Requirements

- All user data (including personal and payment information) must be encrypted using secure protocols (e.g., SSL/TLS).

- The application must implement role-based access control (RBAC) to ensure that only authorized users can access certain features.

5.3 Availability

- The application should be available 24/7 with minimal downtime.

- Backup and disaster recovery procedures must be in place.

5.4 Usability

- The application must be easy to navigate, with a minimal learning curve for new users.

- The user interface should be responsive and adapt to various screen sizes (smartphones and tablets).

5.5 Compatibility

- The mobile app should be compatible with the latest iOS and Android versions.

5.6 Legal and Regulatory Requirements

- The application must comply with local and international data protection regulations (e.g., GDPR).

- The shuttle service must follow regional transportation laws and safety standards.
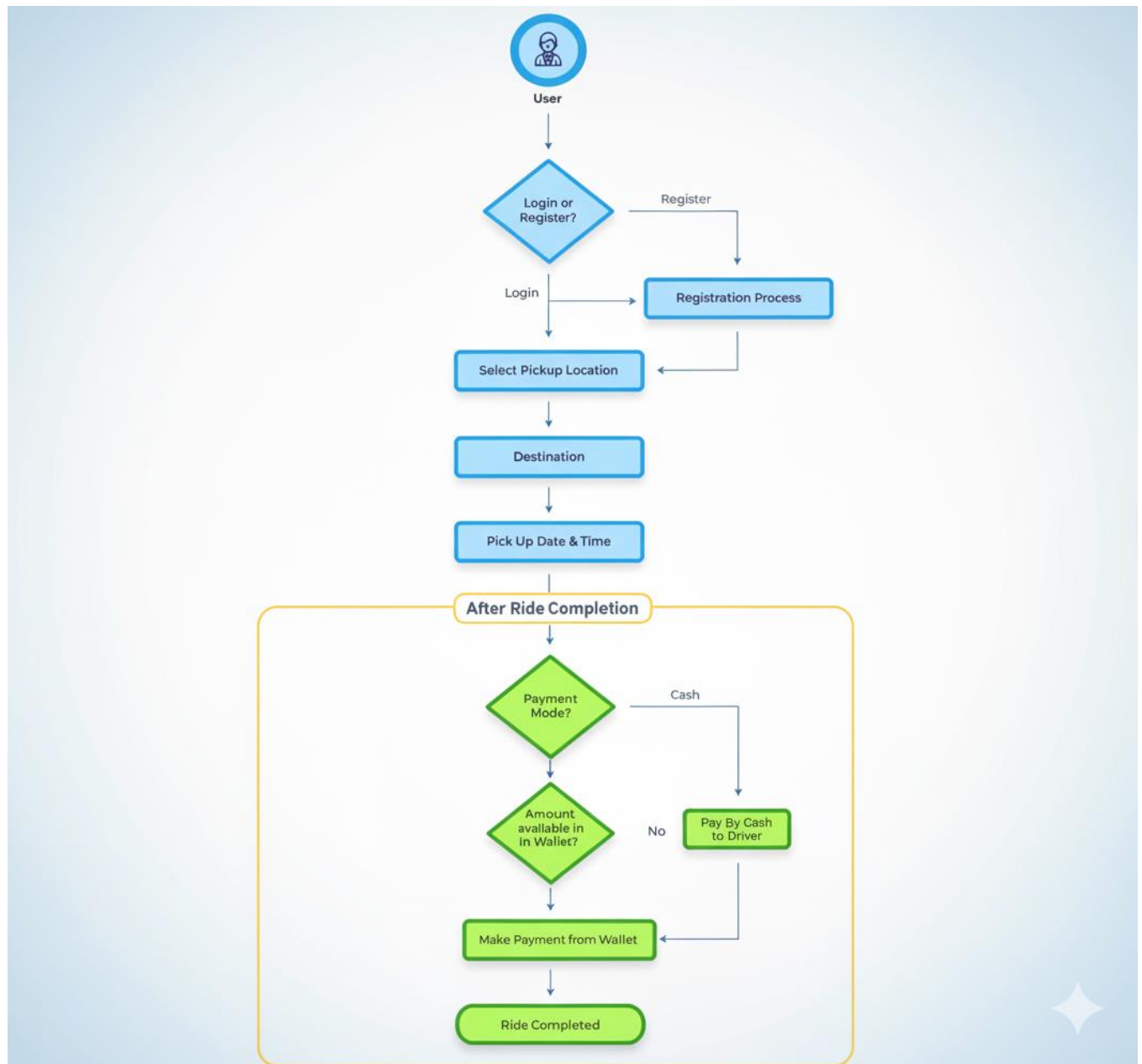
6.Appendices

6.1 Glossary

- Shuttle: A vehicle that operates on a scheduled route for transportation.

- User Profile: A user's personal information and preferences stored in the system.

6.2 Diagrams

- Flowchart for the ride booking process.

- Architecture diagram for the mobile application.

FLOWCHART:

MINDMAP:

Center: My Shuttle

User Management
- User Registration
- User Login
- Profile Management
- Authentication & Authorization

Ride Booking
- Select Pickup and Drop off Locations
- Choose Date & Time
- Check Avaliability
- Seat Selection
- Fare Calculation
- Booking Confirmation
- Cancallation & Rescheduling

Payment System
- Credit Cord Payment
- Payment Gateway Integration
- Promo Codes and Discounts
- Refund Prccessing
- Payment History

Notification System
- Push Notifications
- SMS Notifications
- Email Notfications
- Booking Confirmation Notifications
- Reminder Notifications
- Delay/Update Notffications

Admin Portal
- User Management
- Driver Management
- Ride and Booking Monitoring
- System Analytics and Reporting
- Campaign Management

Driver Management
- Campaign Creation
- Target Urer Sugments
- Campaign Analytics
- Social Medis Integration
- Engagement Tracking

Driver Management
- Driver Registration & Login
- Fare Histery
- Ride Assignment
- Rating and Feedback
- Earnings Tracking

Frequent Rider Program
- Points Accunulation
- Reward Tiers
- Redemption Process
- Program Eligiblity
- Loyalty Benelits