**Software Requirements Specification (SRS) Templates**

**1. Introduction5**

**1.1 Purpose55change**

The purpose of this document is to define the software requirements for [Project Name], serving as a guideline for the development team.

**1.2 Scopechange**

This SRS document covers the functional and non-functional requirements for [Project Name].

**1.3 Definitions, Acronyms, and Abbreviations**

* **SRS**: Software Requirements Specification
* [Add any specific terms or acronyms relevant to your project]

**1.4 References**

[Include any documents, standards, or external references relevant to this SRS.]

**2. Overall Description**

The Software Requirements Specification (SRS) document for the taxi app outlines the fundamental aspects of the software system, including its context within the broader ecosystem, high-level features, user classes and characteristics, operating environment, design and implementation constraints, as well as assumptions and dependencies.

**2.1 Product Perspective**

The taxi app is a standalone software system designed to facilitate seamless transportation services for users. It operates within a larger transportation ecosystem, which includes users, drivers, administrators, and external interfaces such as payment gateways and map services. The app interacts with these external components to provide users with a reliable and efficient taxi booking experience.

**2.2 Product Features**

The taxi app offers a range of high-level features and functionalities, including but not limited to:

* **Ride Booking:** Users can book rides by specifying pickup and drop-off locations and choose from various vehicle options.
* **Real-Time Tracking:** Real-time tracking of drivers' locations allows users to monitor the progress of their rides.
* **Driver Management:** Drivers can register, accept ride requests, and manage their availability.
* **Payment Processing:** Secure payment processing supports various payment methods for fare settlement.
* **Rating and Feedback:** Users can rate drivers and provide feedback after each ride.
* **User and Driver Support:** Comprehensive support for users and drivers via chat, phone, and in-app help center.
* **Safety Features:** Emergency services and safety measures for users and drivers in case of critical situations.
* **Localization:** Multi-language and currency support to cater to a diverse user base.

**2.3 User Classes and Characteristics**

The taxi app caters to three primary user classes:

* **Customers:** Users who seek transportation services. They create ride requests, interact with drivers, and pay for services.
* **Drivers:** Individuals who offer transportation services. They accept ride requests, provide rides, and earn through the platform.
* **Administrators:** System administrators responsible for managing users, drivers, rides, and overseeing the platform's operations.

Each user class possesses unique characteristics, requirements, and roles within the application.

**2.4 Operating Environment**

The taxi app operates in a dynamic environment that includes:

* **Mobile Platforms:** Android and iOS for user and driver mobile applications.
* **Server Infrastructure:** Backend servers to handle ride requests, manage user profiles, and process payments.
* **Network Connectivity:** Reliable internet connectivity for real-time communication between users, drivers, and servers.

**2.5 Design and Implementation Constraints**

The design and implementation of the taxi app are subject to specific constraints, including technology choices, budget limitations, and compliance with local transportation regulations. Additionally, the app must ensure data privacy and security in accordance with relevant laws and regulations.

**2.6 Assumptions and Dependencies**

Assumptions made during the development process include assumptions about user behavior, system performance, and third-party service availability. The taxi app depends on external services for features such as map integration, payment processing, and communication services.

This SRS document serves as a comprehensive guide for the development, testing, and maintenance of the taxi app, providing stakeholders with a clear understanding of its scope, features, and operational requirements.

**3. System Features**

**3.1 Feature 1 Title**

**3.1.1 Description**

[Detailed description of the feature.]

**3.1.2 Requirements**

[List functional and non-functional requirements related to this feature.]

**3.2 Feature 2 Title**

[Repeat the structure for each major feature.]

**4. External Interface Requirements**

**4.1 User Interfaces**

[List the user interfaces and their requirements.]

**4.2 Hardware Interfaces**

[List any hardware interfaces and their requirements.]

**4.3 Software Interfaces**

[List any software interfaces and their requirements.]

**5. Non-functional Requirements**

**5.1 Performance Requirements**

[List performance-related requirements such as response times, throughput, etc.]

**5.2 Security Requirements**

[List security-related requirements.]

**5.3 Availability Requirements**

[List availability and uptime requirements.]

**5.4 Scalability and Capacity**

[List requirements related to scalability and capacity.]

**6. Other Requirements**

[Include any other requirements not covered in previous sections.]

**7. Appendix A: Use Cases**

[Include detailed use case diagrams and descriptions if applicable.]

**8. Appendix B: Glossary**

[List and define any domain-specific terms or acronyms.]