**Software Requirements Specification (SRS) Document**

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to provide a detailed overview of the requirements for the Parking Management System. It outlines the functionalities, constraints, and interfaces of the system to be developed.

## 1.2 Scope

The Parking Management System is designed to automate and manage parking facilities efficiently. It aims to provide users with a convenient way to find available parking spots, reserve parking slots, and facilitate payment processes. The system will cater to both administrators managing parking facilities and users seeking parking spaces.

## 1.3 Definitions, Acronyms, and Abbreviations

1. SRS: Software Requirements Specification
2. CMS: Content Management System
3. API: Application Programming Interface
4. UI: User Interface
5. DBMS: Database Management System

## 1.4 References

IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications.

# 2. Overall Description

## 2.1 Product Perspective

The Parking Management System will serve as a standalone application accessible via web browsers and mobile devices. It will interact with external systems such as payment gateways for processing transactions and databases for storing parking-related data.

## 2.2 Product Features

### 2.2.1 User Features

1. **User Registration and Login**: Users can register accounts and login securely to access system functionalities.
2. **Parking Spot Availability:** Users can view real-time availability of parking spots.
3. **Parking Reservation:** Users can reserve parking spots in advance.
4. **Payment Processing:** Users can make payments for parking reservations securely.
5. **Notification System:** Users receive notifications regarding reservation confirmations, reminders, and payment receipts.

### 2.2.2 Administrator Features

1. **Parking Lot Management:** Administrators can manage parking lot information, including capacity and pricing.
2. **User Management:** Administrators can manage user accounts, view reservations, and generate reports.
3. **Revenue Tracking:** Administrators can track revenue generated from parking reservations.

## 2.3 User Classes and Characteristics

* **User:** Individuals seeking parking spaces.
* **Administrator:** Personnel responsible for managing parking facilities.

## 2.4 Operating Environment

The system will operate on web servers running Next.js and utilize a MySQL database for data storage. It will be accessible via modern web browsers and mobile devices with internet connectivity.

# 3. System Features

## 3.1 User Management

### 3.1.1 Description

Users should be able to register, login, and manage their accounts.

### 3.1.2 Inputs

1. User registration details (name, email, password, etc.).
2. Login credentials (username/email, password).

### 3.1.3 Outputs

1. Confirmation messages for successful registration and login.
2. Error messages for invalid credentials or registration details.

## 3.2 Parking Spot Availability

### 3.2.1 Description

Users should be able to view available parking spots in real-time.

### 3.2.2 Inputs

1. Location selection.
2. Date and time preferences.

### 3.2.3 Outputs

1. List of available parking spots.
2. Visual representation of parking lot layout with available spots highlighted.

## 3.3 Parking Reservation

## 3.3.1 Description

Users should be able to reserve parking spots for a specified duration.

### 3.3.2 Inputs

1. Selected parking spot.
2. Reservation duration.
3. Payment details.

### 3.3.3 Outputs

1. Confirmation of reservation.
2. Reservation details and receipt.

# 4. External Interface Requirements

## 4.1 User Interfaces

The system will provide user-friendly interfaces for both web and mobile platforms.

## 4.2 Hardware Interfaces

The system will require internet-connected devices for access.

## 4.3 Software Interfaces

1. Payment Gateway API: Integration with payment gateway services for payment processing.
2. Google Maps API: Integration for location services and map visualization.

## 4.4 Communication Interfaces

The system will communicate with users via email for notifications and alerts.

# 5. Non-functional Requirements

## 5.1 Performance Requirements

1. The system should respond to user actions within 2 seconds.
2. The system should handle a minimum of 1000 concurrent users.

## 5.2 Security Requirements

1. User passwords should be securely hashed and stored.
2. Secure socket layer (SSL) encryption should be used for data transmission.

## 5.3 Reliability Requirements

1. The system should have a downtime of less than 1% per month.

# 6. Other Requirements

## 6.1 Legal Requirements

* Compliance with data protection regulations (e.g., GDPR, CCPA).

## 6.2 Documentation Requirements

* User manuals for both users and administrators.
* Technical documentation for developers.

# Appendix A: Glossary

1. Reservation: The act of reserving a parking spot for a specified duration.
2. Payment Gateway: A service provider that facilitates online transactions.
3. API: Application Programming Interface, a set of rules and protocols for building and interacting with software applications.
4. SSL: Secure Socket Layer, a standard security technology for establishing an encrypted link between a web server and a browser.

This concludes the Software Requirements Specification (SRS) for the Parking Management System.