Software requirement specification (SRS) document template

Project name: Smart car parking system

Date:10/02/2025

Version:

By: Harish Honrao
Shreya Deshmukh
Shreya Agrawal
Srishti Khandagale

Table of contents

1 Introduction

The purpose of this document is to define the Software Requirements Specification (SRS) for the Smart Parking System Web Application. This web-based system aims to streamline parking slot management by allowing users to check availability, book slots, and make online payments. The system will enhance convenience, reduce manual effort, and provide a seamless parking experience.

1.1 Product scope

The Smart Parking System Web App is designed to provide an efficient, user-friendly, and automated parking slot management solution. This web-based system will allow users to check parking availability, book slots, track check-in/out times, and make online payments seamlessly.
Benefits

- Convenience: Users can book parking slots in advance, reducing waiting time.
- Cost Transparency: A clear pricing model ensures users know their charges beforehand.
- Efficient Management: Admins can track parking usage and payments effortlessly.
- Digital Payments: QR-based UPI payment integration eliminates the need for cash transactions.

Objectives

- Develop a web-based system that allows users to book and manage parking slots.
- Implement an authentication system with signup and login functionality.
- Enable users to check real-time availability of parking slots.
- Automate check-in and check-out time tracking for accurate billing.
- Integrate a pricing model.
- Provide a secure online payment system using QR code-based UPI transactions.
- Send email notifications for booking confirmation and payment receipts.
- Create an admin panel for managing users, bookings, and payments.

Goals

- Develop a fully functional Smart Parking System Web App with Django as the backend.
- Ensure a simple and user-friendly UI for easy slot booking.
- Implement a secure and scalable database to store user and booking information.
- Ensure seamless online payments with proper transaction tracking.
- Deliver a reliable, automated, and efficient parking management solution.

1.2 Product value

The Smart Parking System Web App enhances parking management by providing an automated, efficient, and user-friendly solution. It allows users to check real-time slot availability, make advance bookings, and complete secure digital payments using QR-based UPI transactions. The system ensures transparent pricing with an automated billing structure, where the first hour is free, followed by ₹100 per hour. By eliminating manual record-keeping, it reduces operational workload for parking operators while improving revenue tracking and customer satisfaction. Additionally, the system offers email notifications for booking confirmations and payment receipts, ensuring a smooth user experience. The web app is scalable, making it suitable for different parking facilities, and supports environmental sustainability by reducing fuel wastage through efficient slot management.

1.3 Intended audience

- Parking Lot Operators/Administrators: Personnel responsible for managing parking facilities, tracking slot availability, handling bookings, and overseeing payments.
- Business Owners & Investors: Stakeholders interested in implementing a smart parking system to improve customer service, optimize revenue, and scale operations.
- Developers & System Administrators: Technical teams involved in building, deploying, and maintaining the web application, ensuring functionality and security.

1.4 Intended use

For parking operators, the system provides an admin panel to monitor slot occupancy, manage user bookings, and track payments efficiently. The web app also integrates email notifications for booking confirmations and payment receipts, reducing manual errors and improving the overall user experience. Designed to be scalable, the system can be implemented across multiple locations, making it suitable for businesses, shopping malls, and public parking facilities.

1.5 General description

The Smart Parking System Web App is a web-based platform designed to automate and simplify parking management. The system enables users to check parking slot availability, make reservations, and complete secure digital payments using QR-based UPI transactions

The web app will provide the following core functionalities:

- User Authentication – Secure sign-up and login for users.
- Parking Slot Management – Real-time tracking of available and occupied slots.
- Slot Reservation – Users can book parking slots in advance.
- Check-in & Check-out Tracking – Automatic time recording for billing.
- Pricing & Payment System – Charges calculated based on duration, with QR-based UPI payments.
- Email Notifications – Automated emails for booking confirmations and payment receipts.

- Admin Dashboard – Parking operators can manage slots, bookings, and payment records.

2 Functional requirements

- **User Authentication** – Users can sign up, log in, and manage their accounts.
- **Parking Slot Availability** – Users can view real-time slot availability.
- **Slot Reservation** – Users can select and book a parking slot.
- **Check-in & Check-out** – System records entry and exit times for billing.
- **Online Payment** – QR-based UPI payment integration for cashless transactions.
- **Email Notifications** – Sends booking confirmations and payment receipts.
- **Admin Dashboard** – Admins can manage slots, users, bookings, and payments.
- **Booking History** – Users can view past reservations and payments.
- **Automatic Slot Release** – Frees up the slot after user checkout.

3 External interface requirements

3.1 User interface requirements

- The system should have a responsive and user-friendly web interface accessible from desktops, tablets, and smartphones.
- The UI should display real-time parking slot availability in a clear and structured format.
- Users should be able to sign up, log in, book slots, make payments, and check booking history with minimal steps.
- The admin dashboard should allow easy management of slots, bookings, and payments.
- The system should provide QR codes for UPI payments and display email confirmation messages after booking and payment.

3.2 Hardware interface requirements

- The system should be accessible from any device with a web browser (PCs, laptops, tablets, smartphones).
- A server is required to host the web application and database.
- A printer (optional) can be used for printing receipts, if needed.

3.3 Software interface requirements

- Backend Framework: Django (Python) for handling authentication, database, and business logic.
- Frontend Technologies: HTML, CSS, JavaScript for a responsive UI.
- Database: SQLite, PostgreSQL, or MySQL for storing user and booking details.
- Payment Integration: UPI-based QR code payments.
- Email Service: SMTP or third-party services (e.g., SendGrid) for sending notifications.
- Web Browser Compatibility: Chrome, Firefox, Edge, Safari, etc.

3.4 Communication interface requirements

- The system should support secure HTTPS communication between the client and server.
- Users should receive email notifications for booking confirmation and payments.
- The system should allow real-time updates on slot availability using AJAX or WebSocket.
- Admins should be able to send manual notifications to users if needed.

4 Non-functional requirements

4.1 Security

- User authentication using secure login (username & password hashing).
- HTTPS encryption to protect data transmission.
- Role-based access control (RBAC) to restrict admin and user privileges.
- Secure payment processing via UPI QR code transactions.
- Automatic session timeout for inactive users to prevent unauthorized access

4.2 Capacity

- The system should handle at least 100 concurrent users efficiently.
- The database should be able to store user details, booking records, and payment history for up to 5 years.

4.3 Compatibility

- Compatible with all major web browsers (Chrome, Firefox, Edge, Safari).
- Should work on desktop, tablet, and mobile devices with responsive design.
- Compatible with UPI payment systems for seamless transactions

4.4 Reliability

- The system should have 99.5% uptime, ensuring continuous availability.
- Automatic database backups should be scheduled daily to prevent data loss.
- The system should be able to recover from failures within 5 minutes.

4.5 Scalability

- Should support expansion to multiple parking locations without major changes.
- The system should allow adding more parking slots dynamically as needed.
- The backend should be optimized for future integration with IoT sensors.

4.6 Maintainability

- The codebase should follow modular architecture for easy updates.
- Detailed documentation should be provided for developers and administrators.

- Logs should be maintained to track system errors and user activity for troubleshooting.

4.7 Usability

- User-Friendly Interface – The system will have an intuitive and responsive design for easy navigation across devices.
- Minimal User Effort – Users can complete tasks like booking a slot and making payments within 3 clicks.
- Accessible Design – The UI will follow standard accessibility guidelines to accommodate all users.

4.8 Other

- Performance – The system should respond within 2 seconds for all major user actions.
- Legal Compliance – Must comply with data protection laws and digital payment regulations.
- Environmental Impact – Supports paperless transactions and reduces fuel wastage by minimizing search time for parking.

5 Definitions and acronyms

UI	User Interface
UX	User Experience
HTTP	Hypertext Transfer Protocol
UPI	Unified Payments Interface
RBAC	Role-Based Access Control
DBMS	Database Management System
SMTP	Simple Mail Transfer Protocol
QR Code	Quick Response Code