

Parking Management Web Application

Document Overview

This document outlines the business requirements for the Parking Management Web Application. It details user roles, functional and non-functional requirements, API specifications, operational logic, and architectural decisions to ensure a scalable, user-friendly, and secure system for managing parking operations.

1. Abstract

The *Parking Management Application* is a scalable and intelligent platform designed to streamline parking operations across diverse geographical regions—be it malls, offices, hospitals, or public areas. It provides users with a seamless experience to locate, pre-book.

The system distinguishes itself by automatically allocating the most optimal and available parking slot based on proximity, vehicle type, and slot availability. It supports digital payments via an integrated wallet or UPI, enhancing convenience and reducing transaction time at exit points.

2. Roles and Responsibilities

Admin

- Manage parking slots for cars and bikes
- Configure pricing structures and monitor occupancy and revenue
- View in detail the booked parking slots

User

- Register and log into the platform
 - Manage wallet funds
 - Add vehicle details (one car and one bike per user)
 - Search and book parking slots instantly
 - View booking history and payment transactions
-

3. Functional Requirements

Admin Functionalities

- Organization registration and profile management
- Add/update/delete parking slots (car and bike)
- Define slot-specific pricing (hourly rate)

User Functionalities

- Registration and secure login
 - Wallet management: top-up, view balance, transaction history
 - Add or update vehicle numbers (pre-booking only)
 - Browse real-time slot availability by location
 - Auto-book slots for car or bike
 - Automatic billing based on parking duration
 - View booking and payment history
-

4. Non-Functional Requirements

- **Performance:** Low-latency, responsive UI with real-time updates
 - **Scalability:** Support for concurrent users and multiple organizations
 - **Security:** Data encryption, secure authentication, and transaction integrity
 - **Reliability:** High uptime and robust error handling
 - **Usability:** Clean, intuitive interface across devices
 - **Compliance:** Adherence to applicable data protection and financial regulations
-

5. Edge Cases and Special Conditions

- Each user can book only one slot at a time
- Slots are exclusive: only one user can occupy a slot at any given time
- Vehicle numbers must be unique across all users
- All bookings must be paid through the in-app wallet
- Vehicle details are editable only before a booking is made

- Slot booking is auto-confirmed upon click
 - Charges are time-based and automatically deducted from the wallet
 - Concurrent booking conflicts are resolved using timestamps (first-come, first-served)
-

6. Operational Logic

- Admin define and manage their own parking slot inventory and pricing
 - Upon successful booking, the slot status is marked as "occupied"
 - Charges accumulate hourly and are deducted when the user exits
 - Booking completion triggers final billing and wallet deduction
-

7. API Endpoints

Admin APIs

- POST /organization/register – Register organization
- POST /organization/login – Admin login
- POST /organization/slots – Add new parking slot
- PUT /organization/slots/{id} – Update parking slot details
- DELETE /organization/slots/{id} – Delete parking slot

User APIs

- POST /user/register – Register user
 - POST /user/login – User login
 - POST /user/vehicles – Add vehicle number
 - PUT /user/vehicles/{id} – Update vehicle number
 - GET /slots/available – Fetch available slots by location
 - POST /slots/book – Auto-book a parking slot
-

8. Software Architecture

- **Frontend:** Angular 17 with Bootstrap/TailwindCSS
 - **Backend:** Node.js with Express.js
 - **Database:** MongoDB
 - **Authentication:** JWT (JSON Web Token)
-

9. User Flow

Admin

1. Register and wait for approval
2. Upon activation, add slots and configure pricing
3. Monitor slot status and revenue metrics in real-time

User

1. Register and log in
 2. Add unique vehicle numbers
 3. View available slots by location
 4. Book instantly using auto-book
 5. Use slot and pay & exit
-

10. Deliverables

- Fully functional web application
- Responsive and accessible UI/UX
- Complete and well-documented source code
- User and technical documentation
- Admin and User dashboards
- API documentation
- Maintenance and support guidelines post-deployment