

CSCE 5430 (Spring 2025)

Vehicle Parking Management System (VPMS)

Team Alpha

Table of Contents

Table of Contents.....	2
Introduction.....	3
System Overview.....	3
System Structure Diagram.....	3
System Requirements	6
Functional Requirements.....	6
Non-Functional Requirements.....	7
Implementation Plan	8
Member Contribution Table	9

Introduction

System Overview

The VPMS is a web-based application designed to automate parking space allocation, reservation operations, and real-time payment processing. It is structured into three main subsystems:

- **Frontend Subsystem:** Provides an intuitive user interface for vehicle owners and administrators.
- **Backend Subsystem:** Handles business logic, database operations, and system processing using Laravel.
- **Database Subsystem:** Stores user data, parking slot information, booking history, and payment details in a MySQL database.

System Structure Diagram

Below is the high-level system architecture diagram:

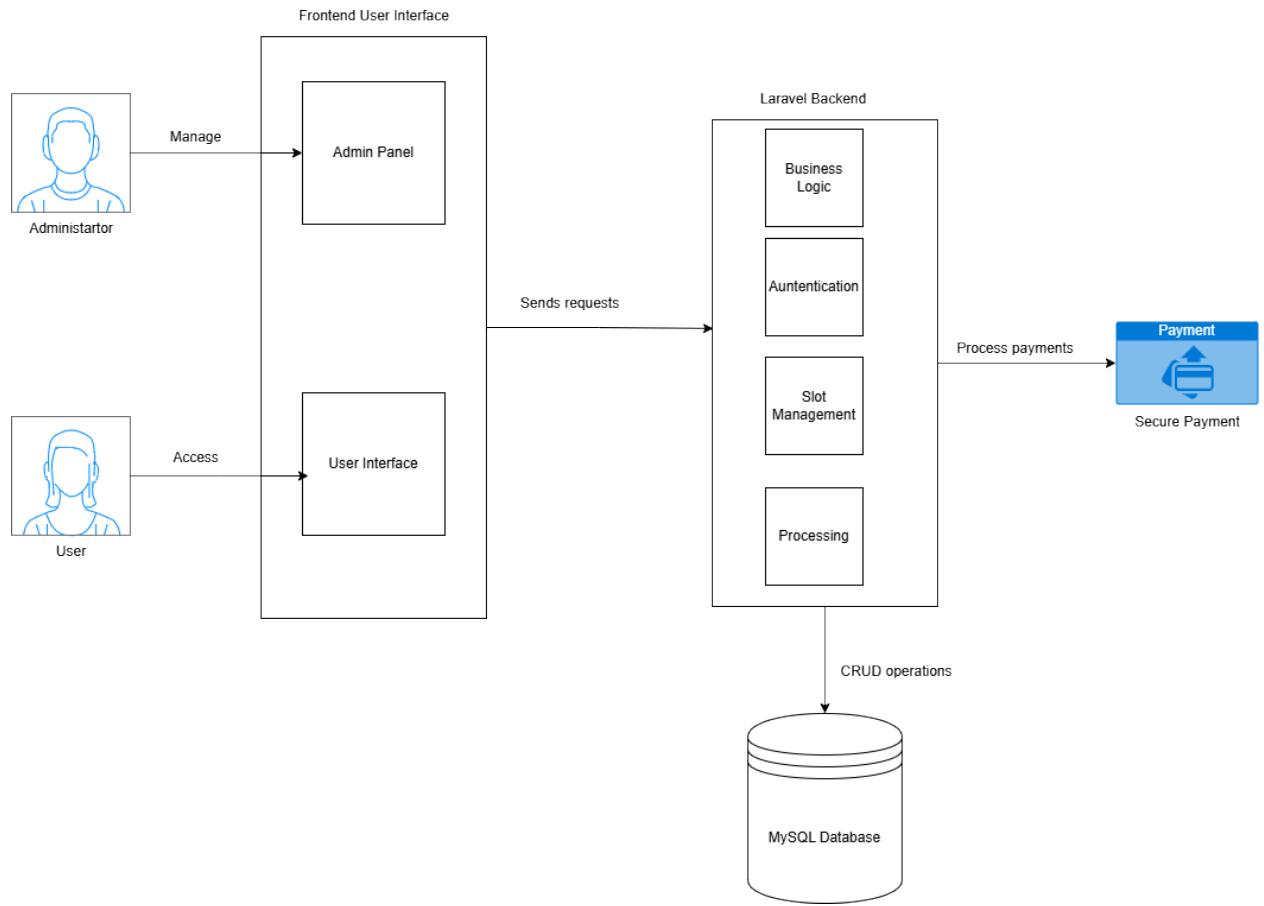


Figure 1: Vehicle Parking Management System Architecture

1. Frontend Subsystem:

- Developed using HTML, Tailwind CSS, and JavaScript.
- Provides user registration, booking, feedback, and real-time slot availability functionalities.

2. Backend Subsystem:

- Built using Laravel (PHP Framework).
- Implements business logic, handles authentication, booking management, and overstay fine calculations.

3. Database Subsystem:

- MySQL database for storing user profiles, parking slots, booking history, and payment transactions.

4. Third-Party Payment Gateway:

- Ensures secure payment processing for parking reservations.

System Requirements

Functional Requirements

1. User Registration & Authentication:

- Users can register, log in, and reset logins.

2. Management of Users

- The admins will be able to add, delete, or edit user profiles.

3. Parking Slot Management:

- Real-time tracking and management of parking slots.

4. Online Booking & Reservation:

- Users can book parking slots in advance.

5. Secure Payment Integration:

- Integration with payment gateways for secure transactions.

6. Vehicle Registration

- Users can register their vehicles using License Plate Details.

7. Overstay Fine Calculation:

- Automatic detection of overstays and fine calculation.

8. Subscription Plans:

- Support for recurring subscriptions for frequent users.

9. Parking History & Reports:

- Users can view their parking history and generate reports.

10. Management of Staff Roles & Permissions

- Admin to manage roles and different levels of permissions.

11. User Feedback & Ratings:

- Users can rate their parking experiences.

12. Automated Ticketing System

- Users can print a ticket for payment verification.

13. Multi-Section Parking Support

- Support for multi-level and underground parking arrangements.

14. Slot Availability Checking

- A real-time check on available parking slots using an intuitive interface

15. Collision Avoidance System

- Prevents booking conflicts for the same slot.

Non-Functional Requirements

- 1. Performance** - The system should handle concurrent users.
- 2. Scalability** - Designed to support additional features.
- 3. Security** - Implements secure authentication and data encryption.
- 4. Usability** - Intuitive UI for seamless user experience.
- 5. Reliability** - 99.9% uptime with robust error handling.

Implementation Plan

The project will be implemented in three incremental development phases:

Phase	Criticality	Deliverables	Requirements
Phase 1: Core Functionality	High	Functional user authentication, slot booking, and payment system.	1. User Registration & Authentication 2. Management of Users 3. Parking Slot Management 4. Online Booking & Reservation 5. Secure Payment Integration 6. Vehicle Registration
Phase 2: Enhanced Features	Medium	Subscription management, overstay fine calculations, and reporting.	7. Overstay Fine Calculation 8. Subscription Plans 9. Parking History & Reports 10. Management of Staff Roles & Permissions.
	Low		11. User Feedback & Ratings.

Phase 3: Advanced Features		Feedback system, multi-section parking, and collision avoidance.	12. Automated Ticketing System
			13. Multi-Section Parking Support
			14. Slot Availability Checking.
			15. Collision Avoidance System

Member Contribution Table

Member Name	Contribution Description	Overall Contribution (%)	Note
Jaswanth Nalluri	Wrote system structure and helped in preparation of implementation plan	15%	Team Lead
Anurag Lakkavathula	Drafted functional and non-functional requirements	15%	
Abhishek Darsha	Defined interfaces and created the system architecture diagram	14%	
Lakshmi Deepika Yadagiri	Organized feature prioritization for development phases	13%	

Sai Sri Naga Sashank Pasupuleti	Drafted non-functional requirements and usability aspects	12%	
Jishna Sathvik Vucha	Created database schema and backend requirement details	10%	
Dhiresh Venkateshwarlu Katuri	Documented and updated the meeting minutes	10%	
Nikhil Chowdary Kollipati	Set up and repository structure and updating github read me and Note-deliverable-2.txt file	11%	