Universidad Distrital Francisco Jose de Caldas

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Technical Report: Parking Management System for Vehicle Entry/Exit and Receipt Generation

Andrés Felipe Mateus Saavedra - 20201020119
Anderson Jefrey López Jiménez - 20162020424
Juan Esteban Oviedo Sandoval – 20192020064

School of Engineering
Bogotá DC. October 2025

Contents

List of Figures

1 Abstract

Parking management is a recurring operational challenge in urban and corporate environments. Manual systems often generate inefficiencies, data inaccuracies, and delays in vehicle control. This report presents the design and implementation of a low-cost digital system that automates the registration of vehicle entry and exit, receipt generation, and space availability management. Developed using React, Flask, PostgreSQL, and Docker, the system aims to improve operational efficiency, reduce human error, and facilitate real-time tracking of vehicles. The document details the methodologies, design architecture, testing strategies, and future improvements proposed. The solution demonstrates potential for scalability and easy deployment in small and medium-sized parking facilities.

2 Introduction

Parking management is a critical component of mobility and space optimization in cities and organizations. Traditional methods that rely on manual entry logs often lead to data inconsistency, time loss, and user dissatisfaction. Furthermore, the lack of real-time monitoring complicates billing and reduces the overall efficiency of parking operations.

To address these issues, this project proposes the development of a low-cost, digital Parking Management System capable of handling entry and exit records, receipt generation, and real-time space tracking. The proposed system provides an efficient and scalable alternative that small parking facilities can implement without major financial barriers.

This report documents the process, design, implementation, and evaluation plan of the system. It also reflects on the technical decisions made during the development phase and proposes future improvements to ensure broader applicability.

3 Literature Review

Several studies have explored the use of technology to optimize parking operations. Chien et al. (2020) proposed a Bluetooth-based parking management system to track vehicle positions at a low cost, demonstrating that automation improves traceability and reduces human dependency.[1]

Similarly, WebbyLab (2024) presented a smart parking approach using IoT sensors for real-time space monitoring and payment integration.[2]

These studies highlight that the combination of low-cost technologies and real-time data processing significantly enhances parking efficiency. However, most existing systems rely on expensive hardware or complex infrastructure. This project bridges that gap by using software-centric, open-source tools to create an accessible, scalable solution.

4 Background

Urban areas face increasing pressure on parking facilities due to population growth and vehicle proliferation. Traditional parking systems rely heavily on manual data entry, which increases human error and reduces throughput. Modern digital systems use database-driven solutions and user interfaces to automate processes, offering improvements in accuracy, time, and resource management.

The proposed system leverages client-server architecture and modular design principles, ensuring adaptability to various environments — from small private lots to large institutional parking spaces.

5 Objectives

5.1 General Objective

To design and implement a low-cost, scalable digital system that automates vehicle entry and exit registration, receipt generation, and real-time space management for parking facilities.

5.2 Specific Objectives

- Develop a modular system architecture with separate frontend, backend, and database components.
- Implement an intuitive user interface using React for parking attendants.

  Build a reliable backend using Flask capable of handling data
- requests and processing business logic.
- Store and manage vehicle and receipt data in a PostgreSQL database.
- Test system performance, scalability, and usability through controlled simulations.

6 Scope

The project focuses on the software layer of parking management — registration, data handling, and receipt generation. It excludes hardware components such as sensors, cameras, or gate control systems, which could be integrated later. The system targets small to medium parking facilities and aims to provide a foundation that can be scaled up with minimal cost.

7 Methodology

The methodology applied in this project followed a structured and iterative engineering approach that ensured coherence between design, implementation, and validation. Each

phase was guided by principles of modularity, scalability, and cost efficiency, in alignment with the project's objective to deliver a low-cost, reliable parking management system.

7.1 System Design

The system follows a modular client-server design with three primary layers:

- **Frontend (React)**: Interactive interface for vehicle registration and monitoring.
- **Backend (Flask)**: Handles business logic, receipt generation, and API requests.
- **Database (PostgreSQL)**: Stores persistent data on vehicles, slots, and receipts.

7.2 Development Tools

- React for frontend design and UI interactivity.
- Flask for backend logic and REST API development.
- PostgreSQL for data persistence.
- Docker and Docker Compose for containerization and deployment consistency.
- GitHub Actions for continuous integration and delivery (CI/CD).

7.3 User Flow

1. The attendant logs into the system.
2. Vehicle entry is recorded (license plate, time).
3. Real-time availability updates on the dashboard.
4. Upon exit, the system calculates duration and generates a receipt.
5. Data is stored automatically in the database.

8 Results

The system's design and early-stage implementation produced a series of architectural and conceptual diagrams that validate its structure, workflow, and usability. Although full real-world testing is pending, these diagrams serve as evidence of a coherent and well-organized design process that supports scalability, maintainability, and user-centered functionality.

8.1 User Story Mapping

The User Story Map outlines the interaction flow from the parking attendant's perspective. It decomposes the system into essential user tasks — login, vehicle registration, space monitoring, and receipt generation. This representation ensures that each system feature aligns with user needs and operational priorities. By prioritizing these stories, the development team maintained focus on usability and functional relevance during design.

Figure 1: User Story Mapping for the Parking Management System

8.2 Class Diagram

The Class Diagram defines the main entities: Vehicle, User, Register, Fee, Slot, and Area. It shows the relationships between these objects and their responsibilities, confirming a robust object-oriented design. This structure enables modularity, simplifies data management, and ensures that future extensions (such as additional user roles or pricing models) can be implemented without major architectural changes.

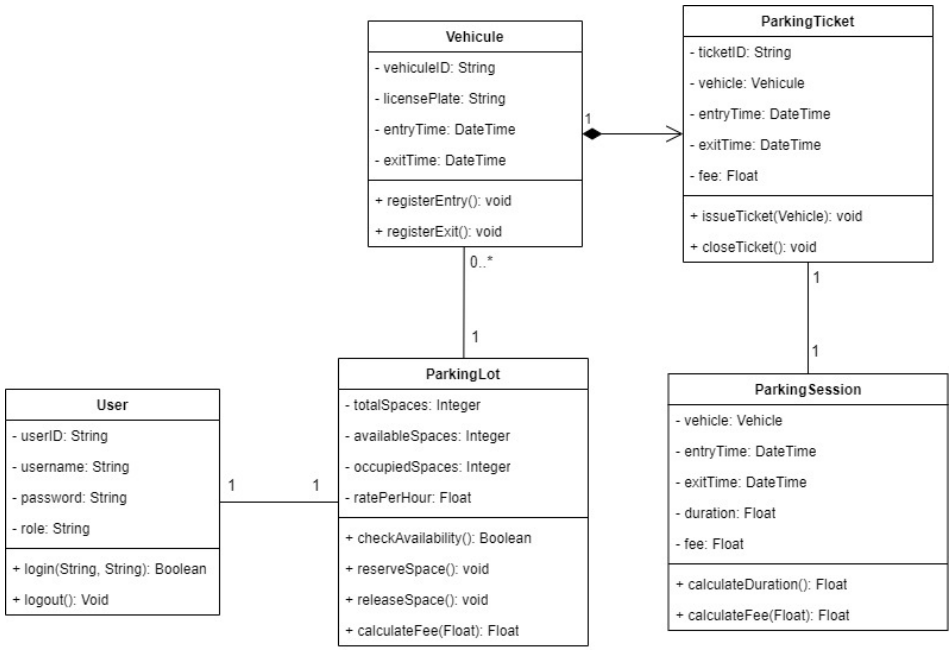

Figure 2: Class Diagram for the Parking Management System

8.3 System Architecture Diagram

The System Architecture Diagram illustrates a three-layer client-server model, composed of the React-based frontend, Flask backend API, and PostgreSQL database.

The diagram clarifies the data flow: the frontend sends HTTP requests to the backend, which processes logic and stores or retrieves data from the database. This separation of concerns guarantees clear communication between modules and supports future scalability in distributed environments.

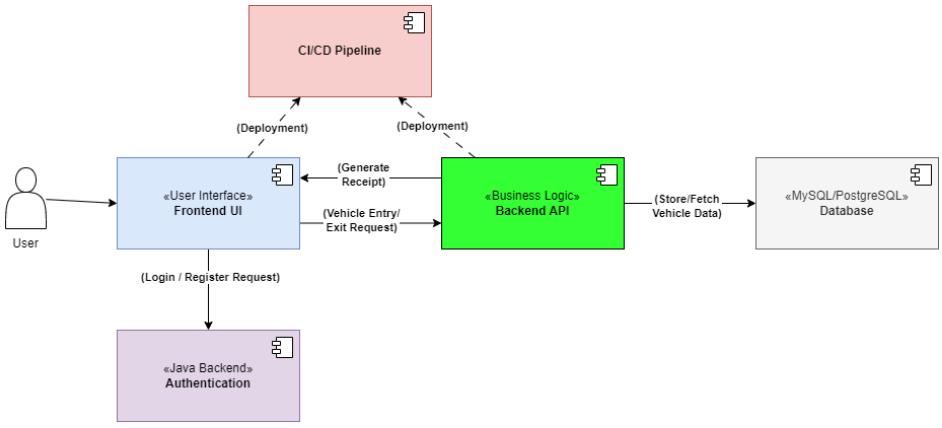

Figure 3: System Architecture Diagram for the Parking Management System

9 Discussion

The modular architecture and use of containerized technologies ensure the system's scalability and portability. While commercial parking systems often depend on proprietary hardware, this solution emphasizes open-source software and low operational cost.

Key findings suggest that a properly designed database and a responsive web interface can provide sufficient functionality for small to mid-scale parking operations without high investment. The proposed testing plan (unit, integration, and usability testing) ensures continuous improvement and system reliability.

10 Conclusion

The Parking Management System offers a cost-effective, automated alternative to traditional parking processes. Its modular design, based on open-source technologies, ensures scalability, maintainability, and adaptability to various environments. Future work will integrate payment processing, sensor data, and a mobile application for attendants and users. By focusing on simplicity and efficiency, this project demonstrates the feasibility of low-cost technological solutions in urban management systems.

References

[1] C. F. Chien, Y. C. Ding, C. H. Wei, and C. H. Wei, "A low-cost on-street parking management system based on bluetooth beacons," *Sensors*, vol. 20, no. 6, p. 1701, 2020. DOI: 10.3390/s20061701. [Online]. Available: https://doi.org/10.3390/s20061701.

[2] WebbyLab. "Smart parking system using iot: How to create your own." Accessed: 2025-10-24. [Online]. Available: https://webbylab.com/blog/smart-parking-system-using-iot/.