VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

AAT PROJECT REPORT

on

SMART PARKING SYSTEM

Submitted by

Dhanush K (1BM24CS404)
K L Srujan (1BM24CS408)
Chethan N (1BM23CS075)
Chethan T M (1BM23CS076)

Under the Guidance of

Sonika Sharma D

Assistant Professor, BMSCE

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B. M. S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019

Aug 2025 to Dec 2025

B. M. S. College of Engineering,
Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the project work entitled "**Smart Parking System**" carried out by who are **Dhanush K (1BM24CS404), K L Srujan (1BM24CS408) Chethan N (1BM23CS075) and Chethan T M (1BM23CS076)** Bonafide students of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visveswaraya Technological University, Belgaum during the year 2025. The project report has been approved as it satisfies the academic requirements in respect of **AAT Object Oriented Modelling (23CS5PCOOM)** work prescribed for the said degree.

Index Sheet

SMART PARKING SYSTEM

1. Introduction

In today's rapidly growing urban areas, parking has become a major challenge due to the increasing number of vehicles. A Smart Parking System is a technology-driven solution designed to automate and simplify the process of finding and managing parking spaces.

This project aims to develop a Smart Parking System using software engineering principles to make parking more efficient, reduce traffic congestion, and save time for drivers. The system uses sensors, mobile or web applications, and real-time data to help users locate available parking spots easily. It also enables parking administrators to manage spaces effectively.

By applying software development life cycle (SDLC) models, design principles, and modern technologies, this project demonstrates how software engineering can solve real-world problems in urban transportation.

1.1 Problem Statement

Urban areas struggle with increasing traffic and limited parking space, making it difficult for drivers to locate available parking efficiently. Manual or traditional parking systems lead to congestion, wasted fuel, and user frustration.

A Smart Parking System is required to automate the detection of available parking spots, guide users through real-time applications, and help administrators manage parking efficiently.

2. Software Requirements Specification (SRS)

1. Introduction

1.1 Purpose

The purpose of the Smart Parking System is to automate the monitoring and allocation of parking spaces, provide real-time availability to users, and enable administrators to manage parking efficiently.

1.2 Document Conventions

This document follows the IEEE 830-1998 SRS guidelines. Requirements are categorized as functional, interface, and non-functional.

1.3 Intended Audience

This SRS is intended for developers, project managers, parking authorities, administrators, and students working on smart city applications.

1.4 Project Scope

The system detects empty parking slots, displays real-time availability, assists users in reserving or navigating to slots, and supports admin-level monitoring. It enhances urban mobility by reducing congestion and saving time.

1.5 References

IEEE 830-1998 SRS Standard

IoT & Sensor Technology Documentation

Mobile/Web UI Design Guidelines

2. Overall Description

2.1 Product Perspective

The system integrates sensors, a database, and a mobile/web interface. Sensors detect vehicle presence, while the application displays available slots and updates the system in real time.

2.2 Product Functions

Key functions include slot detection, real-time availability display, user reservation, parking history, admin monitoring, and automated alerts for full/empty zones.

2.3 User Classes

Drivers: View and reserve parking slots

Parking Administrators: Manage slots and monitor occupancy

System Operators: Maintain sensors, database, and system health

2.4 Operating Environment

Runs on mobile/web applications connected to a cloud database. Requires IoT sensors, Wi-Fi/LAN, and server connectivity.

2.5 Design and Implementation Constraints

Requires real-time sensor data, stable network connection, secure authentication, and compliance with smart city infrastructure guidelines.

2.6 User Documentation

User manual, admin guide, and mobile/web tutorial will be provided.

2.7 Assumptions and Dependencies

Assumes functioning sensors, uninterrupted internet, and user devices with app/browser access.

3. Specific Requirements

3.1 Functional Requirements

- Detect parking occupancy using sensors
- Display available and occupied spots in real time
- Allow users to reserve a slot (if reservation mode available)
- Provide navigation to the selected parking space
- Enable admins to add, update, or disable parking slots
- Generate daily/weekly/monthly parking usage reports
- Notify users when parking is full or reservation is expiring

3.2 External Interface Requirements

User Interface

- Mobile/Web dashboard showing slot availability
- Admin panel for monitoring and management
- Notification system for updates and alerts

Hardware Interfaces

- IoT sensors (IR, ultrasonic, RFID)
- Gate barriers (optional)
- User devices (mobile, tablets, desktop)

Software Interfaces

- Cloud database (SQL/NoSQL)
- Sensor communication protocols (MQTT/HTTP)
- Map/navigation API

Communication Interfaces

- Internet/Wi-Fi for real-time updates
- Bluetooth or local network (optional features)

3.3 Non-Functional Requirements

- **Performance:** Slot updates should reflect within 1 second.
- **Reliability:** System must work continuously with accurate sensor data.
- **Security:** User data and admin access must be securely authenticated.
- **Scalability:** System must support expanding parking slots and locations.
- **Availability:** System should be accessible 24/7.
- **Maintainability:** Easy to update sensors, software modules, and UI.

4. Appendices

4.1 Acronyms

SPS – Smart Parking System

IoT – Internet of Things

UI – User Interface

4.2 Glossary

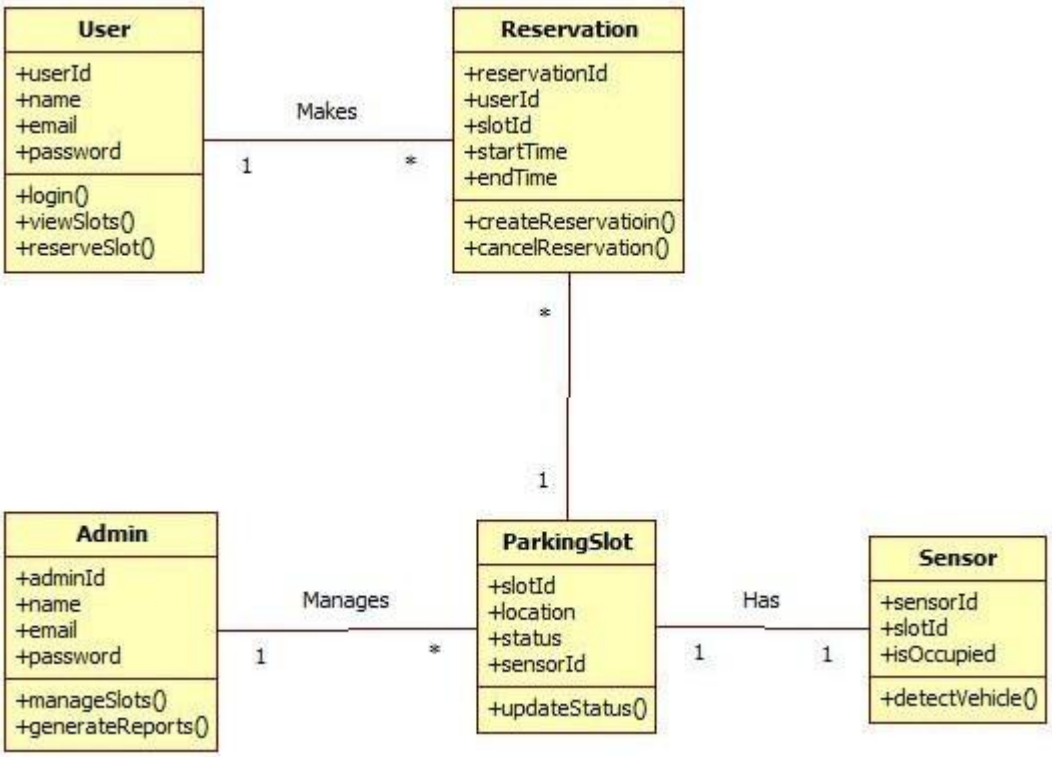
Parking Slot: A designated vehicle space

Reservation: Booking a slot before arrival

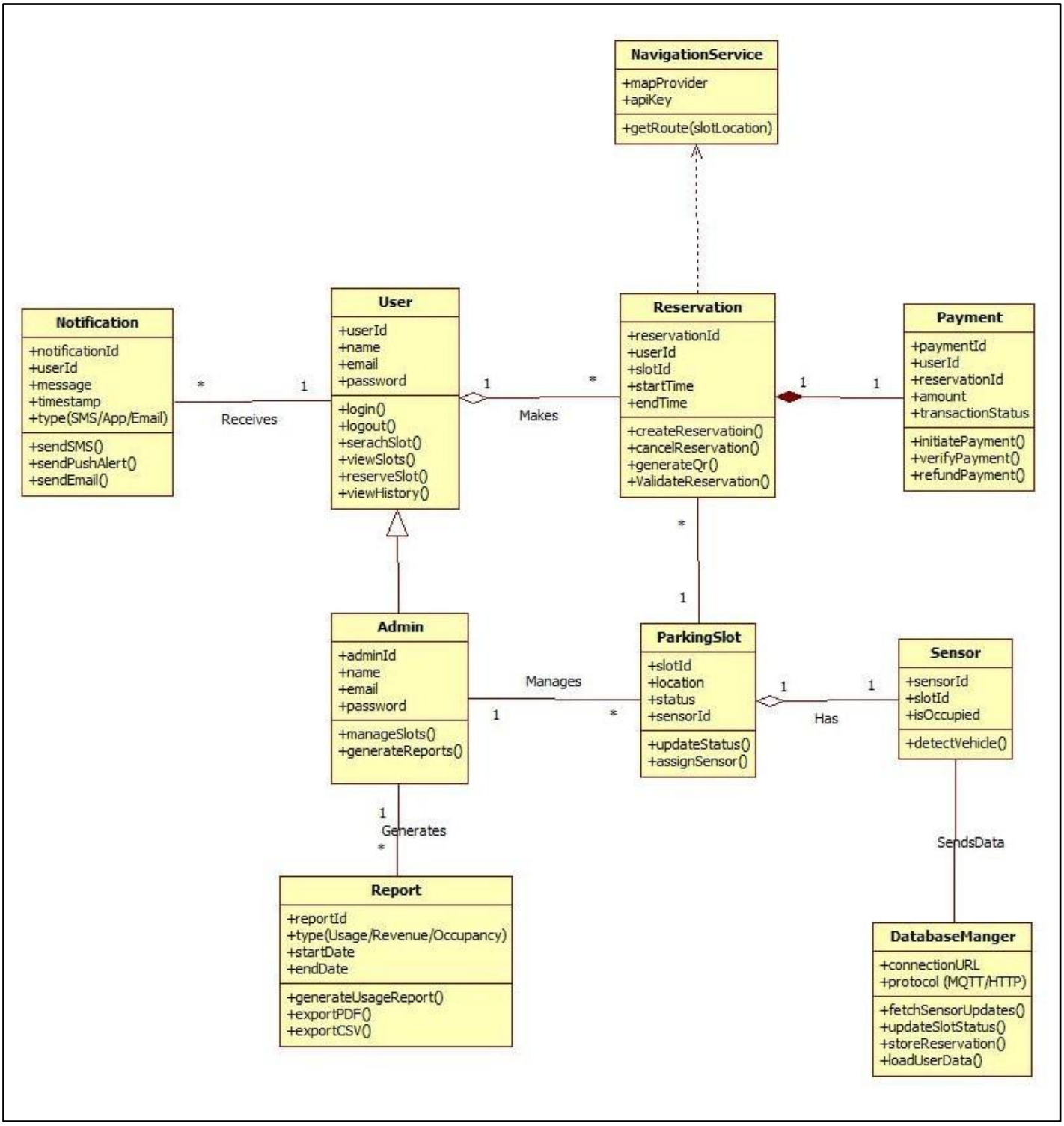
Occupancy Detection: Identifying whether a slot is filled or empty

Class Diagram

1. Simple Class Diagram

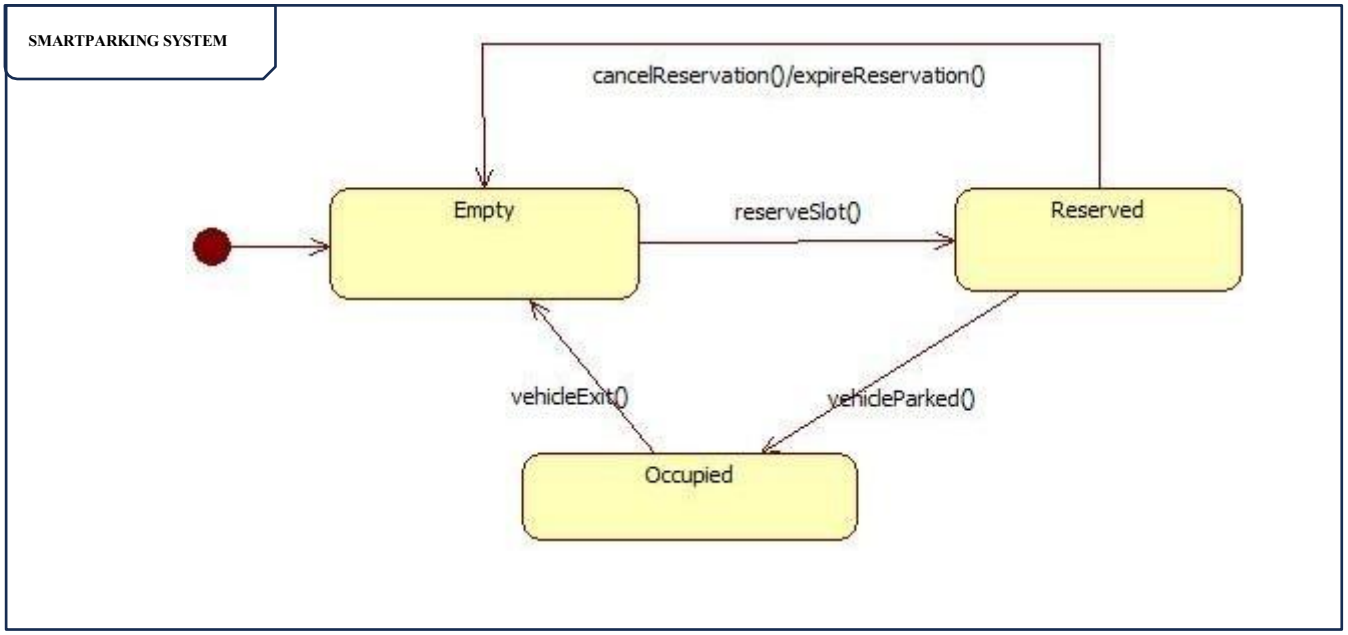


2. Advance Class diagram

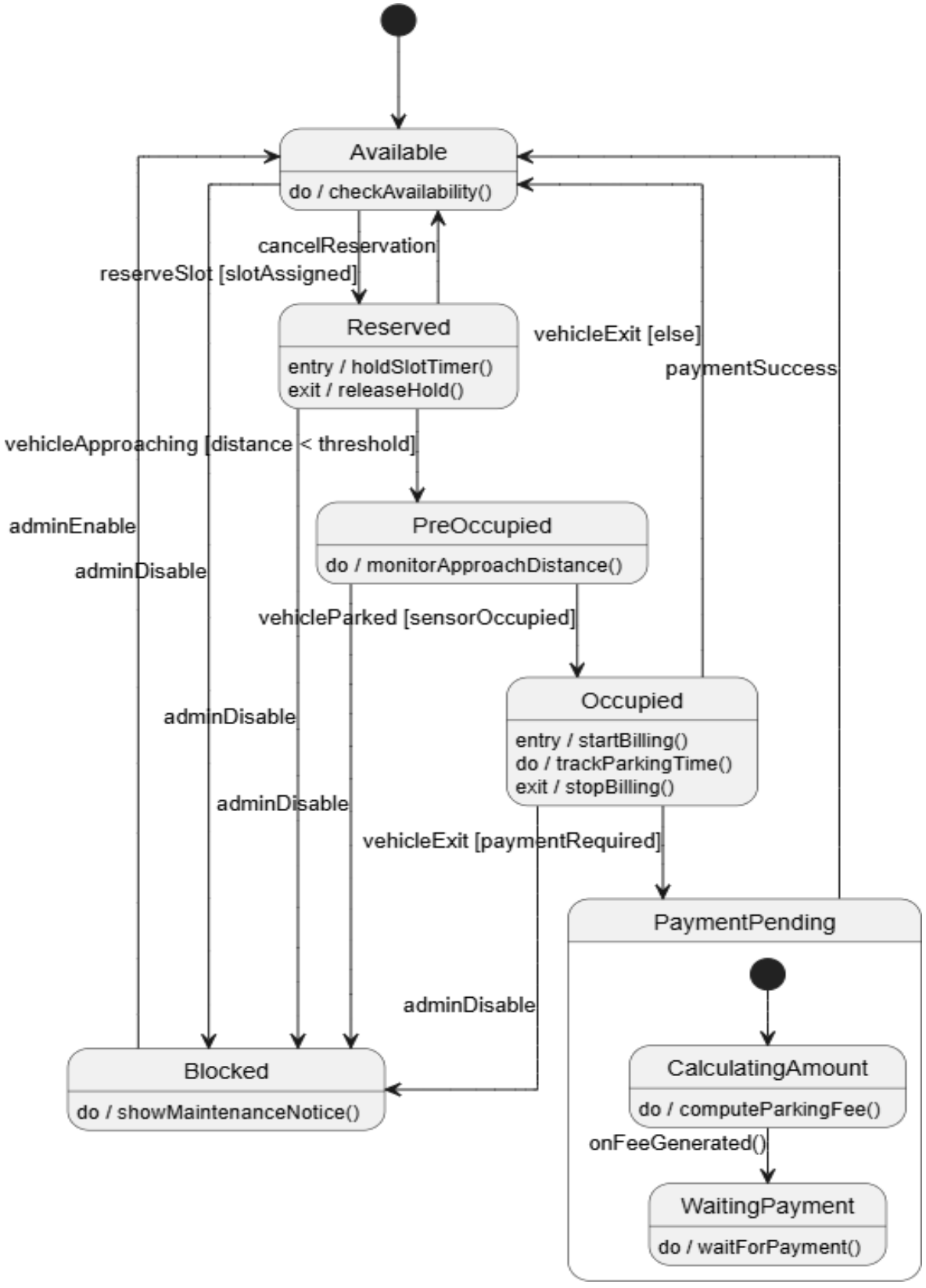


State Diagram

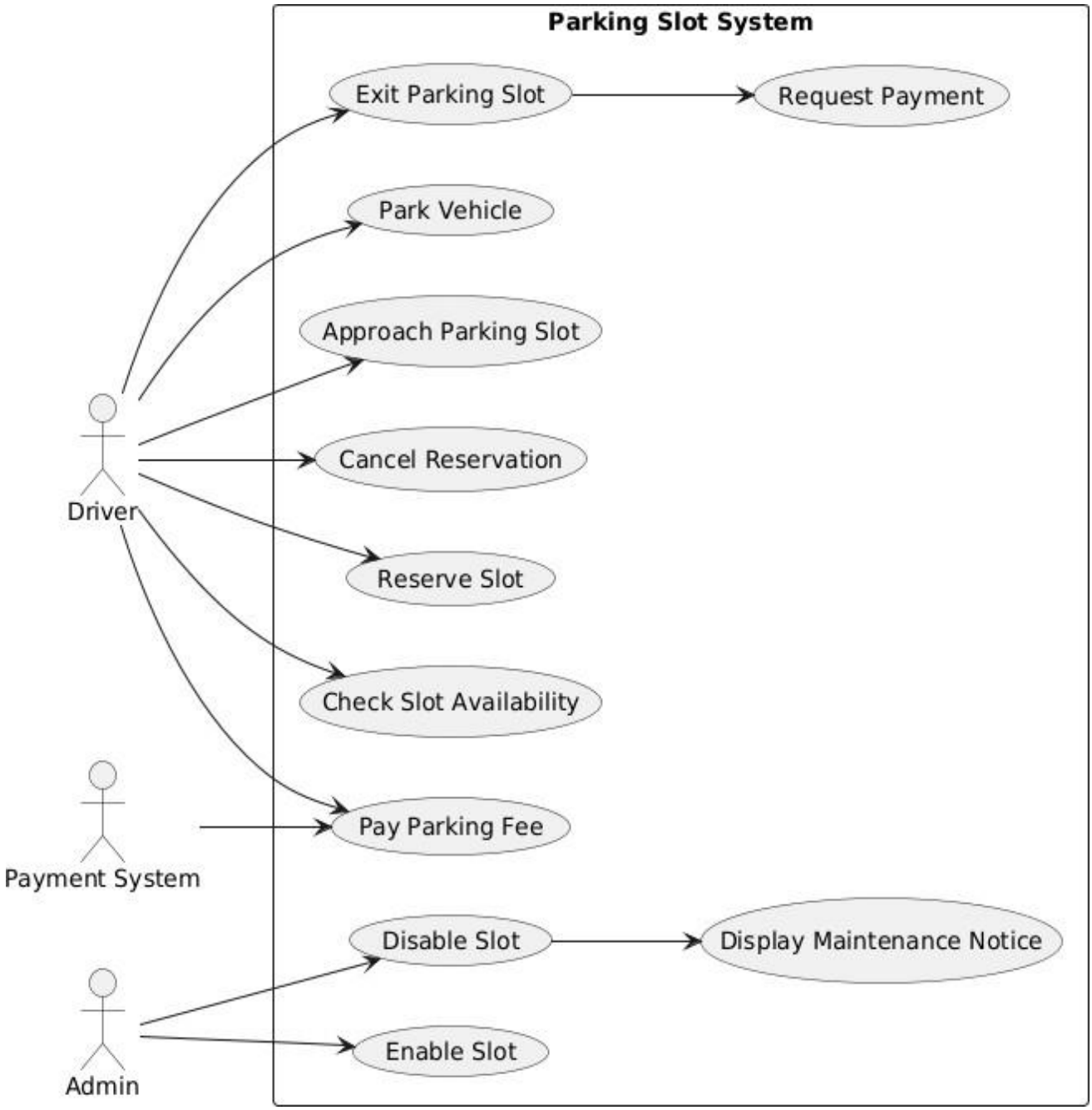
1. Simple State Diagram



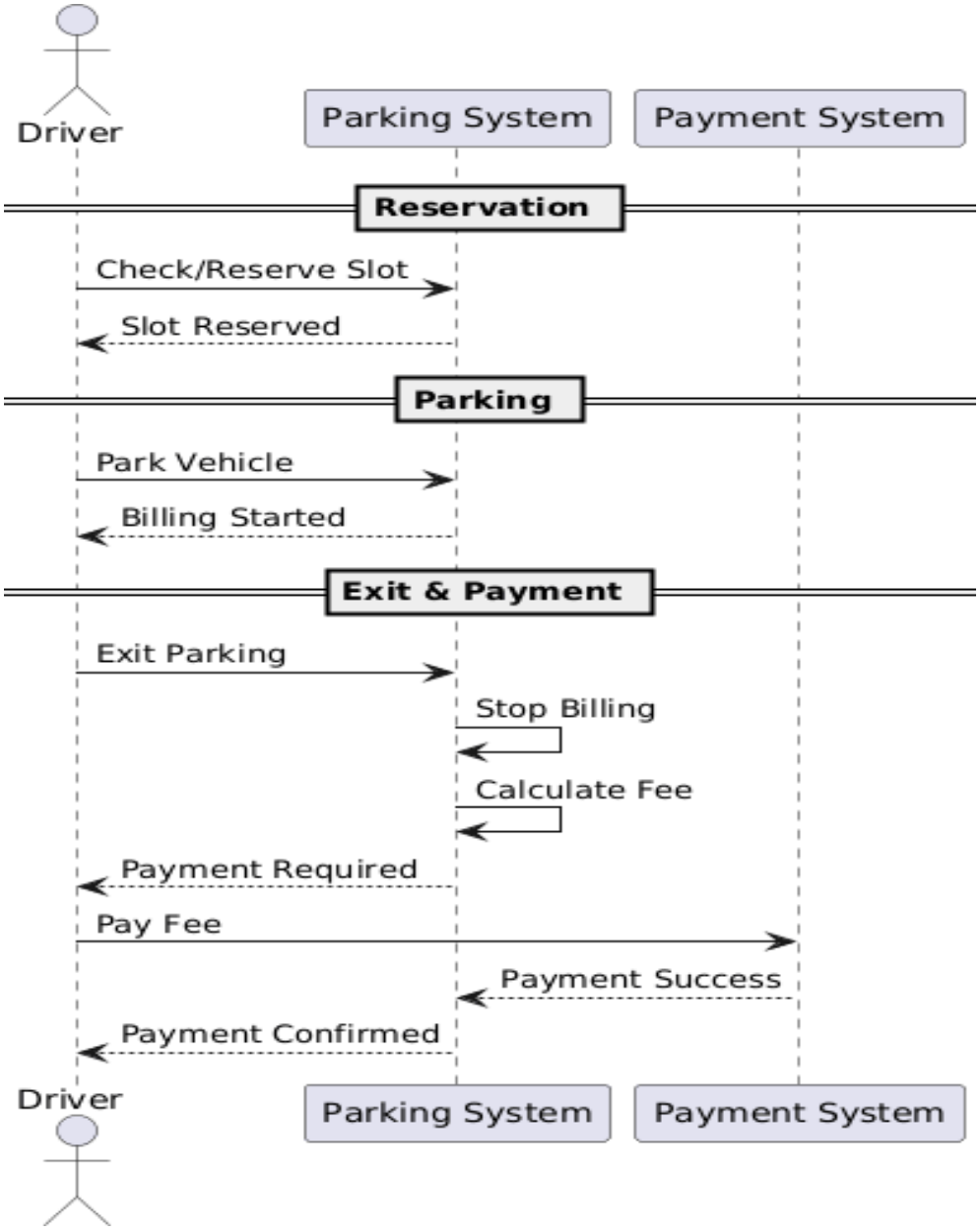
2. Advanced State Diagram



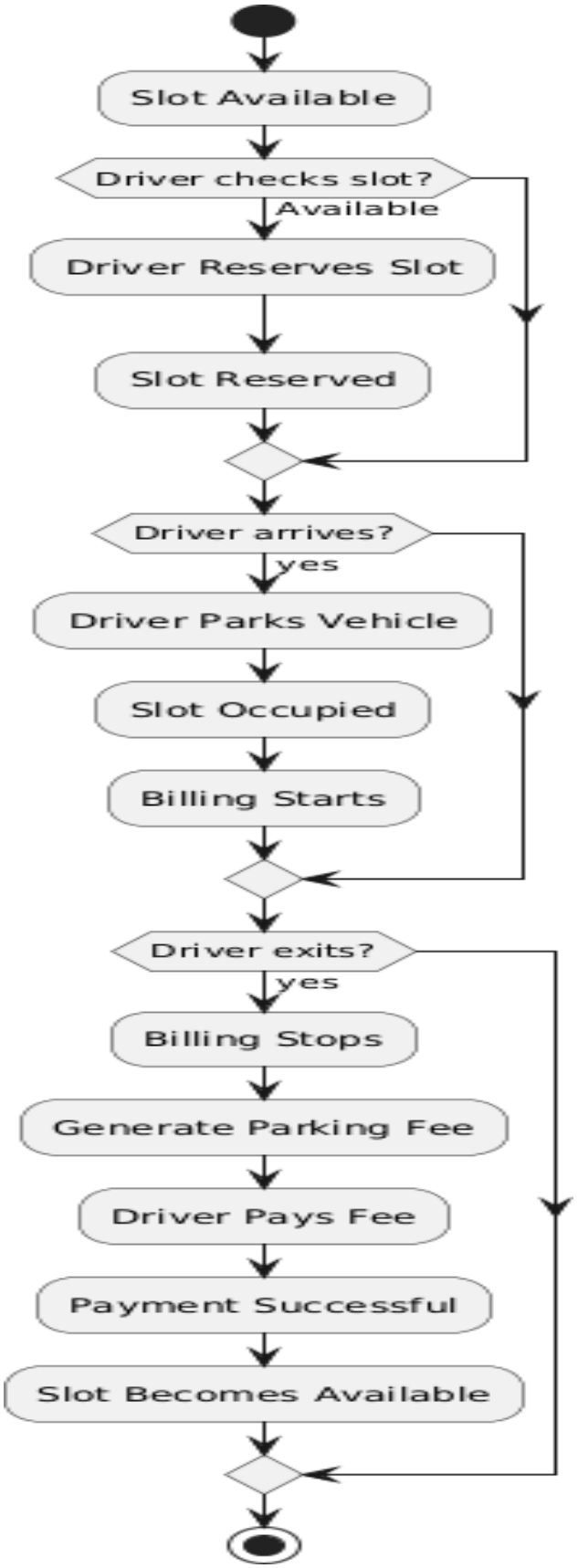
Use Case Diagram



Sequence Diagram



I. Activity Diagram


Slot Available
Driver checks slot?
Available
Driver Reserves Slot
Slot Reserved
Driver arrives?
yes
Driver Parks Vehicle
Slot Occupied
Billing Starts
Driver exits?
yes
Billing Stops
Generate Parking Fee
Driver Pays Fee
Payment Successful
Slot Becomes Available

Conclusion

The Parking Slot Management System successfully demonstrates how automation and smart system design can simplify everyday parking operations. By integrating features such as slot availability checking, reservation, parking detection, billing, and digital payment, the system enhances both efficiency and user convenience. The state diagram, use-case diagram, activity diagram, and sequence diagram together provide a clear visual understanding of how the system behaves and how different components interact. Each diagram plays an important role in analysing the system flow—from the moment a slot becomes free, through reservation and parking, to fee calculation and payment confirmation.

The project highlights the importance of well-structured system behaviour and shows how smooth transitions between various operational states can reduce confusion and manual workload. Furthermore, the design ensures that users experience a seamless flow, while administrators can maintain and manage slots easily. Overall, the system not only improves operational accuracy but also contributes to better time management, reduced congestion, and a more organized parking environment. This foundation also opens opportunities for future enhancements such as mobile app integration, IoT-based sensors, and real-time navigation support.