

**PIMPRI CHINCHWAD COLLEGE OF ENGINEERING, PUNE**

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**MINI PROJECT REPORT**

on

**“Parking management System”**

Submitted by:

Name of Student: Satvik Patil,Devashish Rahate

Roll No.: 122B1F097,122B1F105

Class & Division: Final Year B

Academic Year: 2025–26

## TABLE OF CONTENTS

Sr. No.	Content	Page No.
1	Introduction	3
2	Objectives	3
3	System Requirements	3
4	System Design (UML Diagrams)	3-9
5	Implementation	9
6	Output Screens	9-14
7	Conclusion	15
8	References	15

## 1. INTRODUCTION

The **Parking Management System** is a Java-based application designed to efficiently manage vehicle parking, entry/exit records, and staff/admin interactions using Object-Oriented Programming (OOP) principles. The project addresses the need for a streamlined, user-friendly system to handle parking operations, motivated by the inefficiency and errors in manual parking management. It applies OOP concepts such as **inheritance, encapsulation, association, and composition** to model real-world parking activities effectively, including vehicle check-in, check-out, revenue tracking, and administrative control.

## 2. OBJECTIVES

- To implement core OOP concepts in Java.
- To develop an efficient and user-friendly application.
- To understand the use of abstraction and inheritance in solving real-world problems like parking management and parking space handling.

## 3. SYSTEM REQUIREMENTS

Hardware:

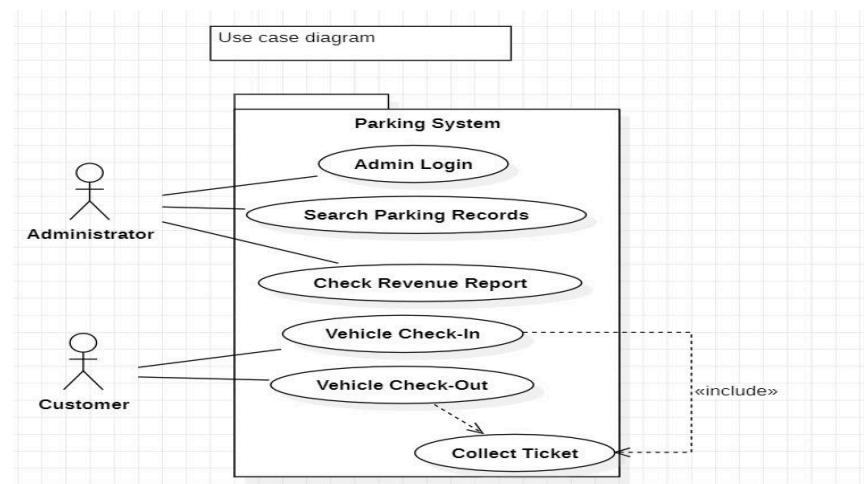
- Processor: Intel i3 or higher
- RAM: 4 GB or higher

Software:

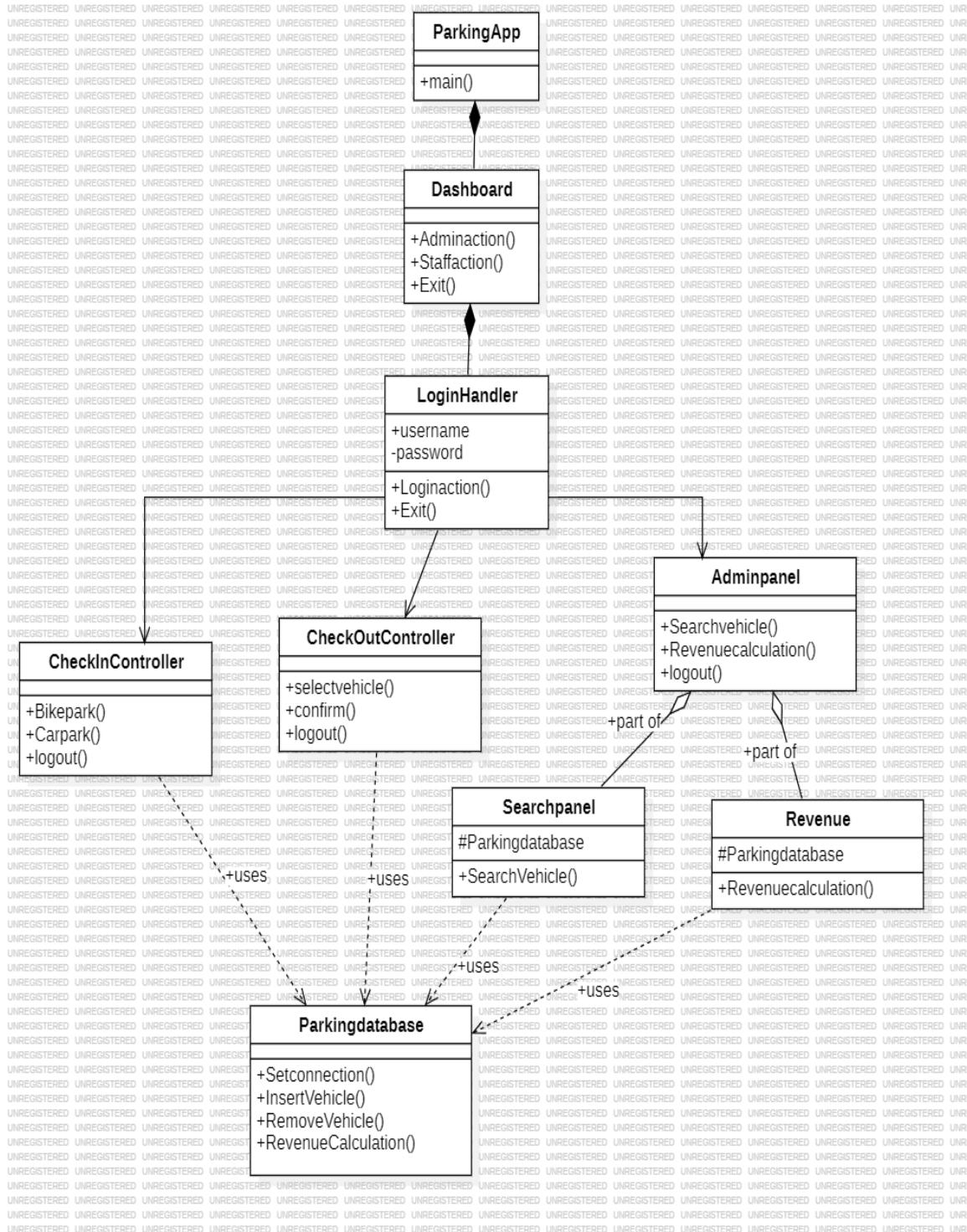
- Java JDK 17 or above
- IDE: Eclipse / IntelliJ IDEA / NetBeans
- OS: Windows / Linux / macOS

## 4. System Design (UML Diagrams)

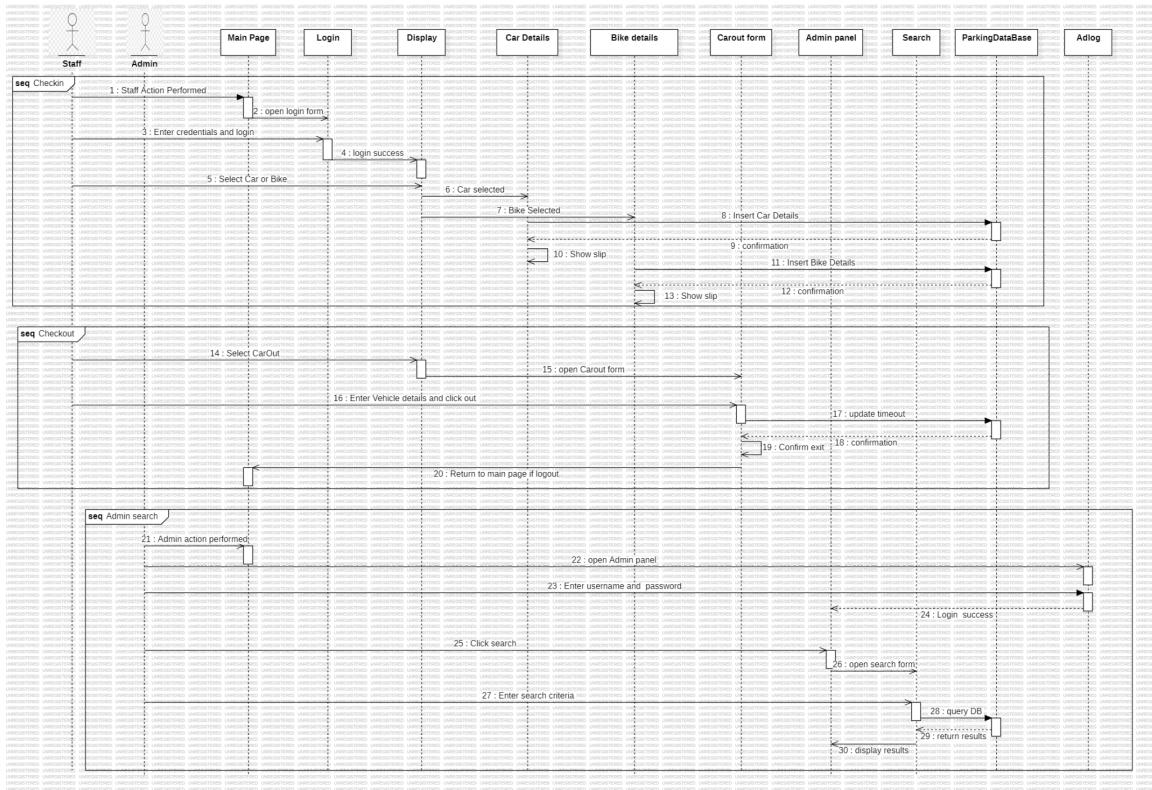
- Use Case Diagram



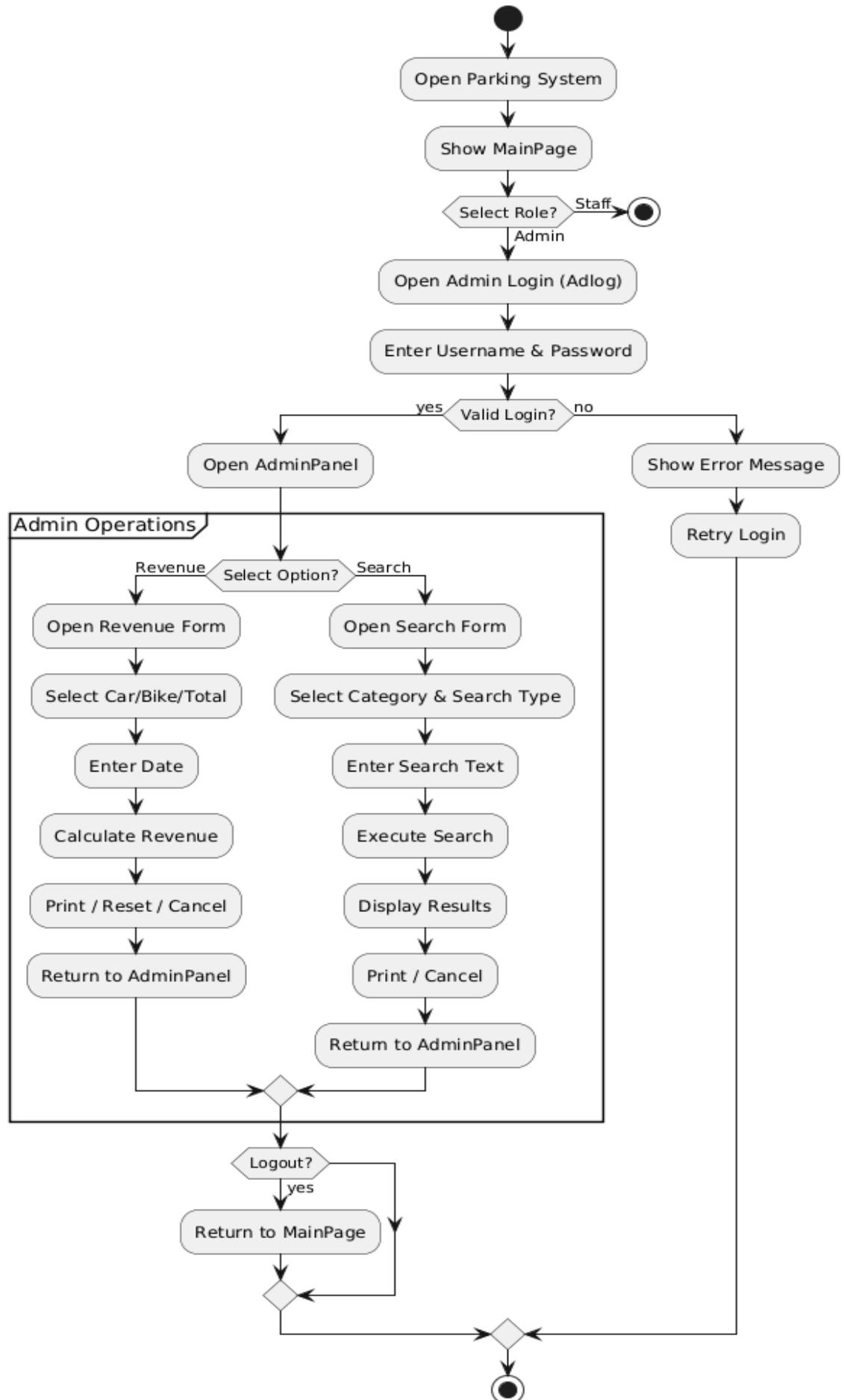
## • Class Diagram

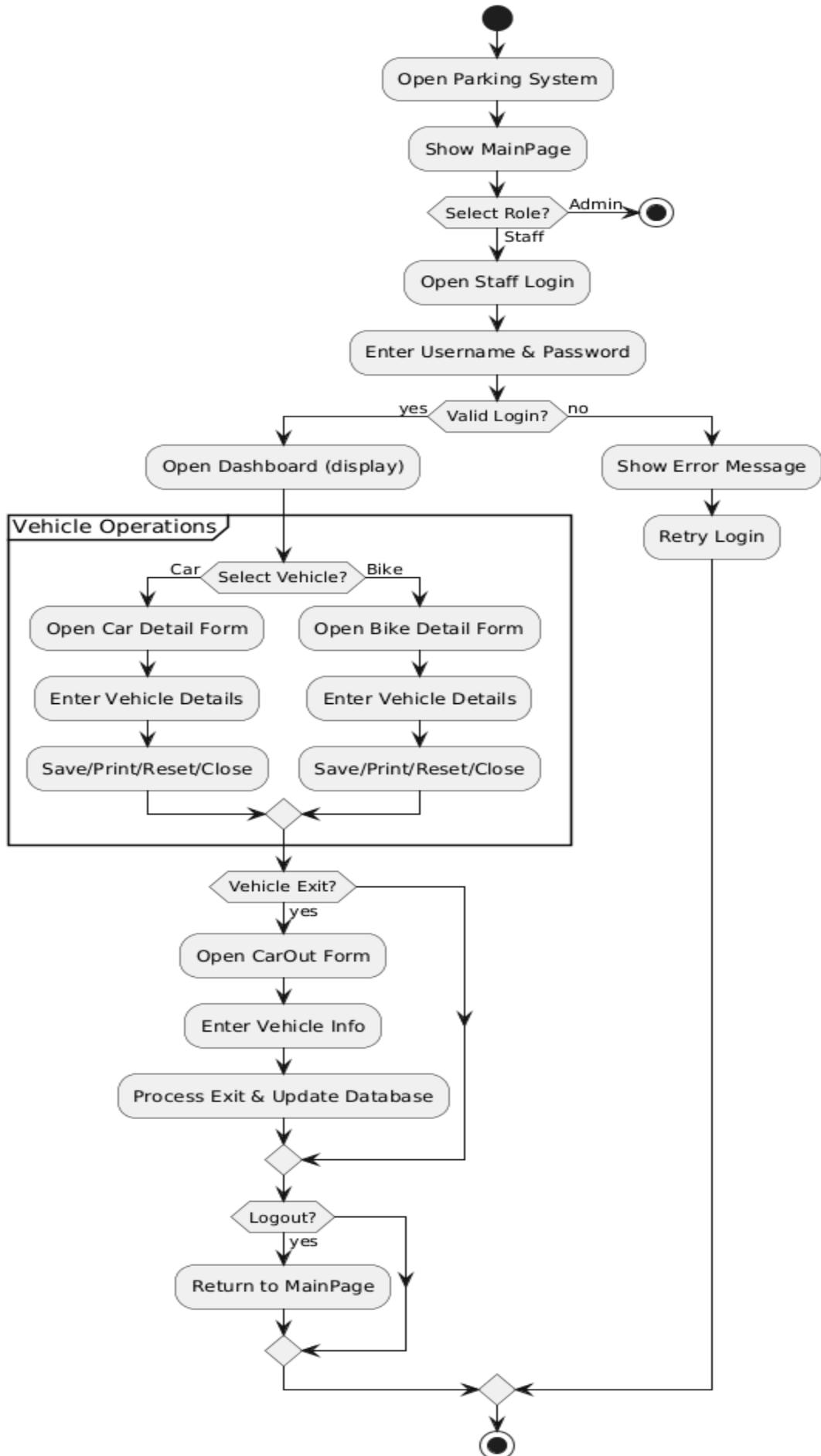


## • Sequence Diagram

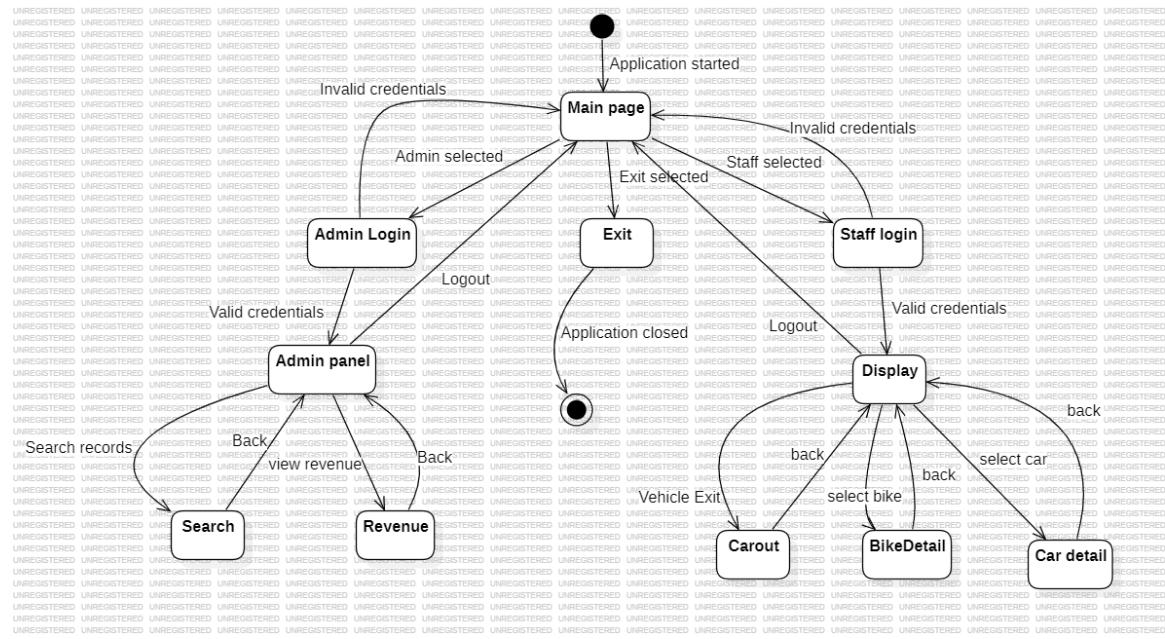


## • Activity Diagram

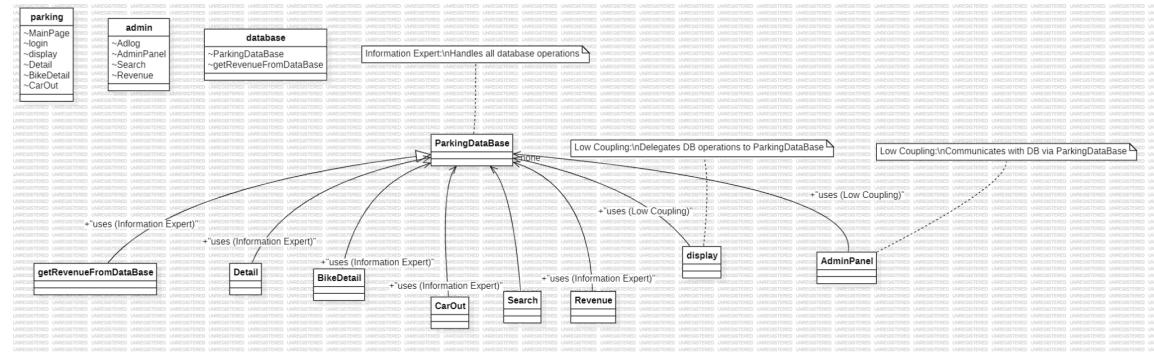




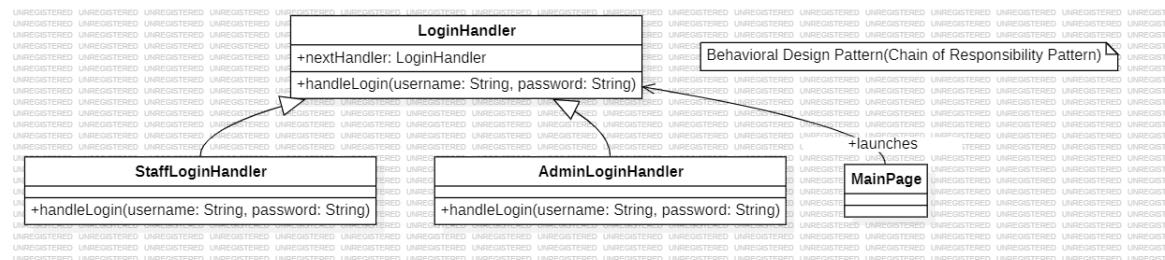
## -state diagram-

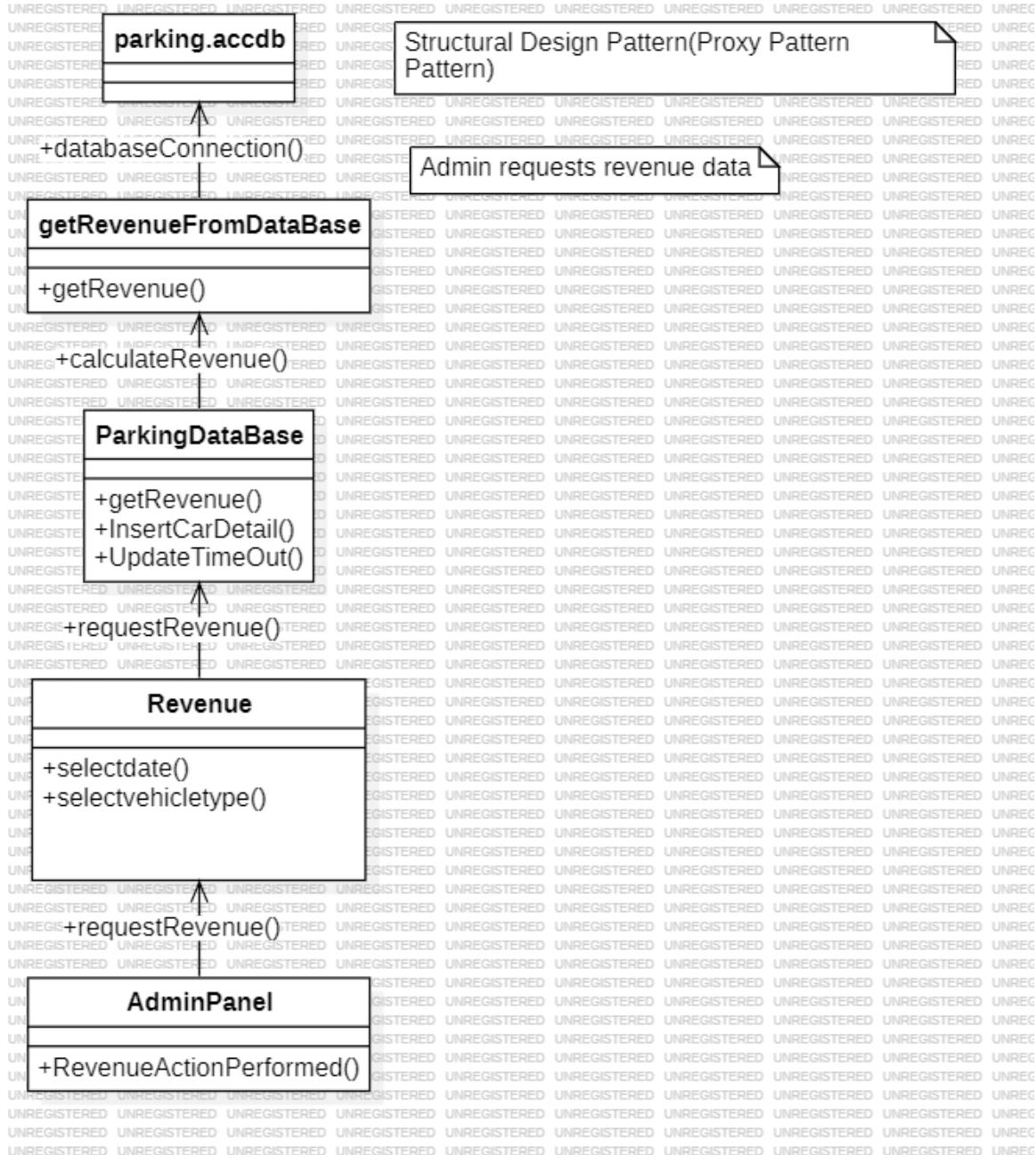


## -Grasp Diagram-



## -GoF Diagram-





**Use Case Diagram:** Shows the interactions between **Staff** and **Admin** actors and the main system functions (Vehicle Entry, Vehicle Exit, Revenue Report, Search Records).

**Class Diagram:** Depicts the structure using **classes** (e.g., **ParkingDataBase**, **Detail**, **CarOut**, **AdminPanel**), showing their attributes, methods, and relationships.

**Sequence Diagram:** Illustrates the **step-by-step process** of a critical operation, like **Processing Vehicle Entry**, showing message flow between forms and the database helper.

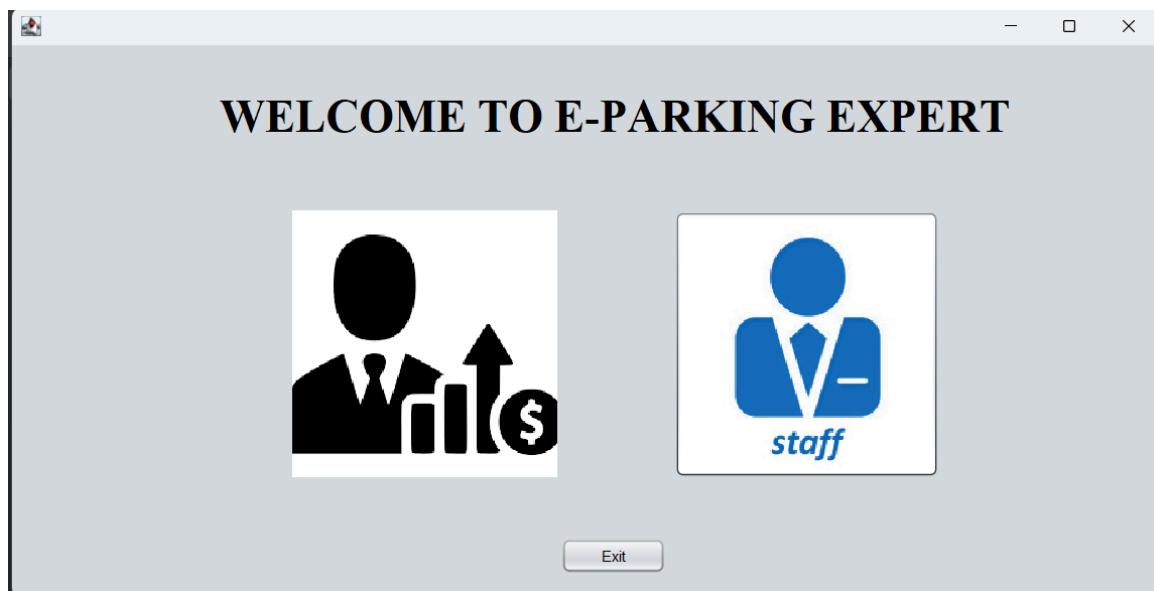
**Activity Diagram:** Maps the complete **flow of control** for both **Staff operations** (Login \$\rightarrow\$ Parking Management) and **Admin operations**

## 5. IMPLEMENTATION

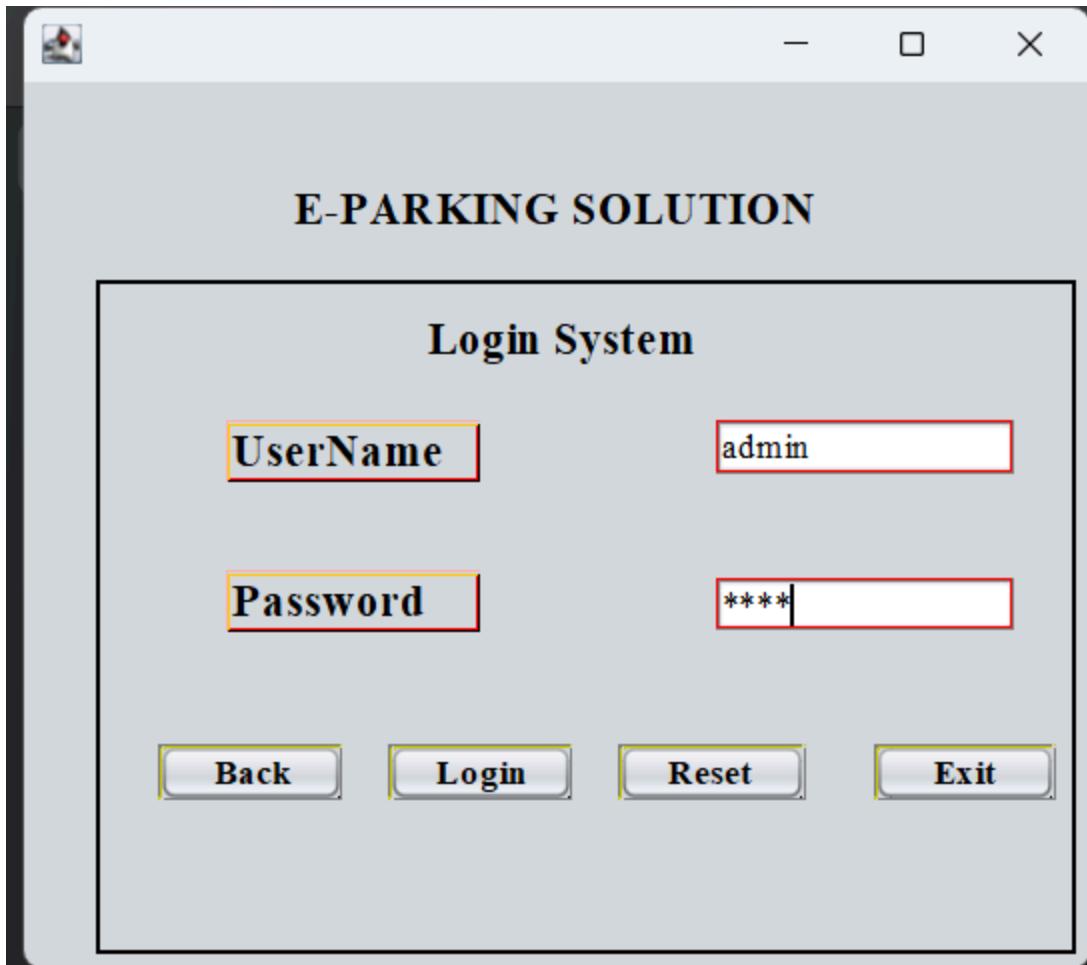
The project is implemented using Java, leveraging OOP principles:

## 6. OUTPUT SCREENS

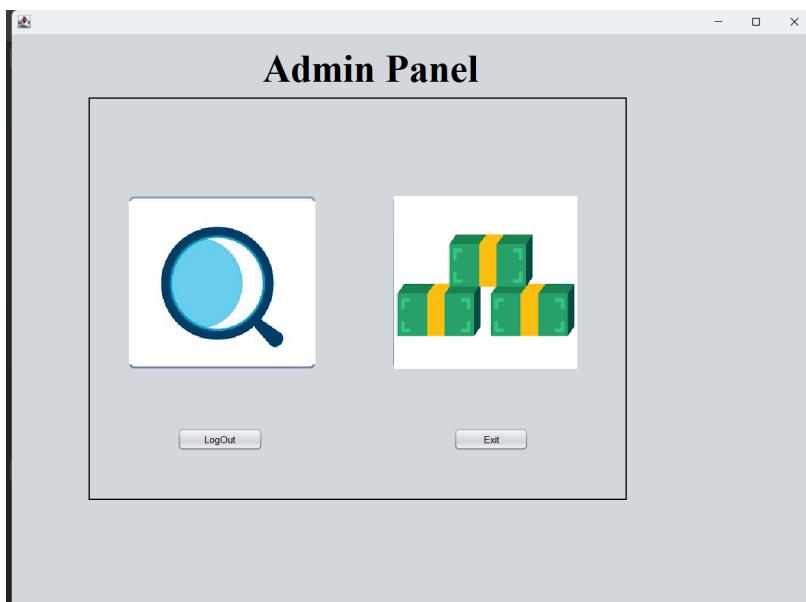
**Launch Screen:**



**Login Page:**



**Admin Panel:**



### **Admin Search:**

The screenshot shows a Windows-style application window titled "Admin Search". At the top, there are three dropdown menus: "Date" (set to "03-11-2025"), "Bike" (set to "Search"), and a third dropdown menu currently showing "Select", "RegNo", "ID", and "Date". Below these are two buttons: "Search" and "Print". The main area contains a table with the following data:

ID	RegNo	ParkingDate	TimeIn	TimeOut	Position
8	mh1997	03-11-2025	02:58:18 am	03:01:28 am	8
9	mh1997	03-11-2025	02:58:27 am		8
10	mh1997	03-11-2025	02:58:31 am		8
11	mh1997	03-11-2025	02:58:39 am		8

### **Admin Revenue Check:**

The screenshot shows a Windows-style application window titled "Admin Revenue Check". It features a date input field set to "26-05-2019", a dropdown menu set to "Car", and four buttons at the bottom: "Check", "Print", "Cancel", and "Reset". The main area displays the following text output:

\*\*\*\*\*Car Revenue\*\*\*\*\*  
Date:26-05-2019  
Total Cars Parked: 7  
Parking Charges for a Car: 30  
Total Revenue Collected: 210

**Check IN Panel:**

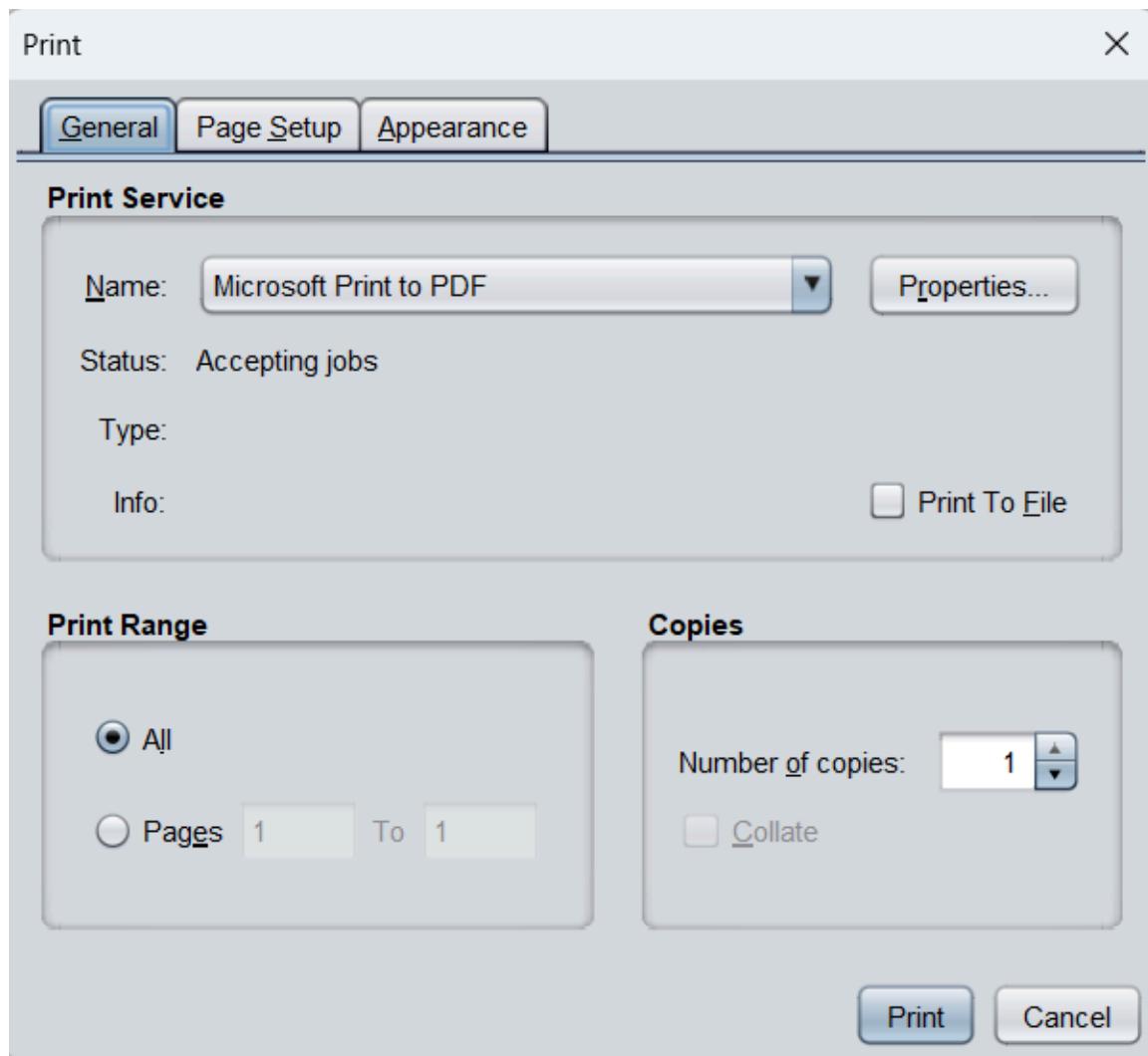


**Ticket Booking:**

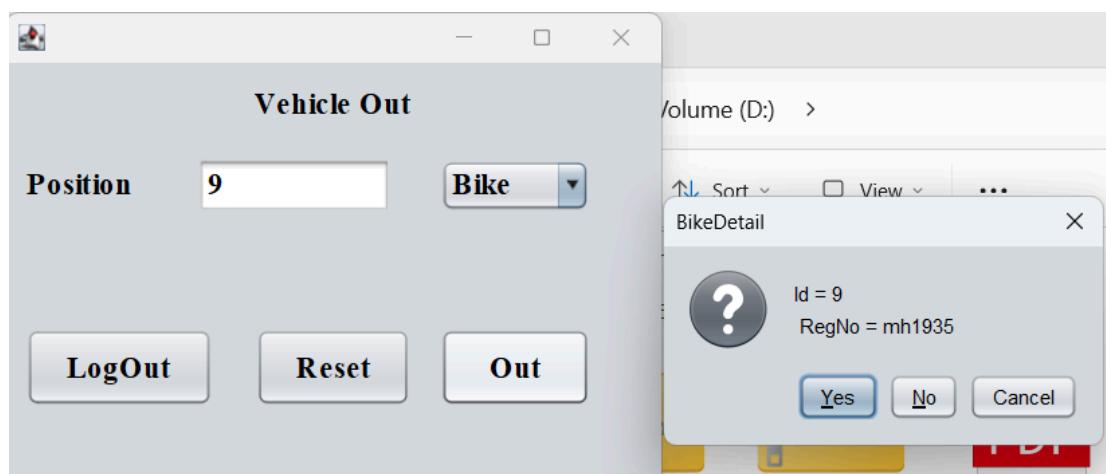
The window has a title bar "Detail". It contains four input fields: "ID" (value 9), "Reg #" (value mh1935), "Position" (value 9), and "Ticket" (value 30). To the right of these fields is a box labeled "\*\*\*\*\*Parking Slip\*\*\*\*\*" containing the following text:  
ID: 9  
RegNo: mh1935  
Position: 9  
Parking Charges: 30

At the bottom are four buttons: "Save", "Print", "Reset", and "Close".

**Print Ticket:**



**Check Out:**



## **7. CONCLUSION**

The Parking Management System project provided hands-on experience in applying OOP concepts like encapsulation, inheritance, and association in Java. It successfully demonstrated vehicle entry management, exit processing, revenue calculation, and user role management for both staff and admin. Future enhancements could include a more advanced GUI, real-time parking availability display, or integration with automated payment and ticketing systems.

## **8. REFERENCES**

- **Java Documentation (JDK 17)**
- **UML Standards by Object Management Group (OMG)**
- **GitHub Repository Guidelines**