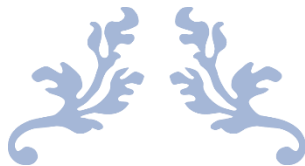




**Daffodil**  
*International*  
**University**



---

# PROJECT REPORT

---

## Parking Management System

**Course Code: CSE 222**

**Course Titel: Object Oriented Programming Lab**

<b>Submitted To:</b>  Md. Jubayar Alam Rafi Lecturer Department of CSE	<b>Submitted By:</b>	
	<b>Name</b>	<b>ID</b>
	Redwan Ahmed Rafi	232-15-374
	Md. Mahfujur Rahman	232-15-324
	Md. Ruhul Amin	232-15-727
	Sanzida Aktar Nupur	232-15-759
	Mst. Nafiza Nazneen	232-15-545
Section: 65_P Department of CSE		

**Submission Date: March 18, 2025**

## ACKNOWLEDGEMENT

We express our sincere gratitude to all those who guided and supported us throughout this project. First and foremost, we would like to extend our heartfelt thanks to our esteemed teacher, **Md. Jubayar Alam Rafi**, for their invaluable guidance, encouragement, and unwavering support. Their expertise, patience and dedication were instrumental in the successful completion of this endeavor.

We are truly grateful to the head of the department and our project guide for their enthusiastic guidance and assistance, which proved invaluable in shaping this work. Their precious time, valuable suggestions and constant encouragement were crucial in overcoming challenges and achieving our goals.

This project is a reflection of the collective efforts of many individuals. We would like to acknowledge and appreciate the contributions of our lab mates, lab staff, and everyone who directly or indirectly played a role in making this project a success.

Teamwork and collaboration are the cornerstone of any successful project and we are indebted to all those who contributed their time, efforts and intellectual inputs towards its realization.



# Table of Contents

- 1. Abstract**
- 2. Introduction**
- 3. System Requirements**
  - **Functional Requirements**
  - **Non-Functional Requirements**
- 4. System Design**
  - **UML Diagrams**
  - **System Architecture**
- 5. Implementation**
  - **Technologies Used**
  - **Key Modules and Code Structure**
  - **Multi-threading and Networking**
- 6. Testing and Error Handling**
- 7. Challenges and Solutions**
- 8. Conclusion and Future Enhancements**
- 9. References**

# Parking Management System

## **1. Abstract**

The Parking Management System is designed to efficiently manage parking slots and registered members. The system allows for the registration of members, parking and removing of cars, and viewing the list of parked cars and available slots. The primary objective of this system is to ensure that the parking space is utilized effectively by registered members. The key features include member registration, car parking and removal, and the ability to check available parking slots. The system also allows viewing the list of registered members. The solution is implemented using Java and Object-Oriented Programming principles such as classes, objects, inheritance, and polymorphism. The project aims to streamline parking management, avoid parking conflicts, and ensure a smooth experience for both the users and parking operators.

## **2. Introduction**

The Parking Management System was developed to address the problem of efficient parking space allocation and management. It simplifies the process for users to park and remove their cars while ensuring only registered members can use the parking slots. The system also keeps track of the available and occupied parking slots. Its relevance is seen in real-world applications like public parking lots, office buildings, and residential areas. The main objectives are to improve the management of parking spaces, register members, and manage parking and removal of cars.

### Importance and Real-World Relevance

- Eliminates manual tracking of parking slots.
- Reduces unauthorized parking issues.
- Enhances efficiency and convenience for users.

### Objectives

- Implement an automated parking system.
- Allow only registered members to park vehicles.

- Provide real-time tracking of available slots.
- Ensure secure and error-free operations.

### 3. System Requirements

#### Functional Requirements

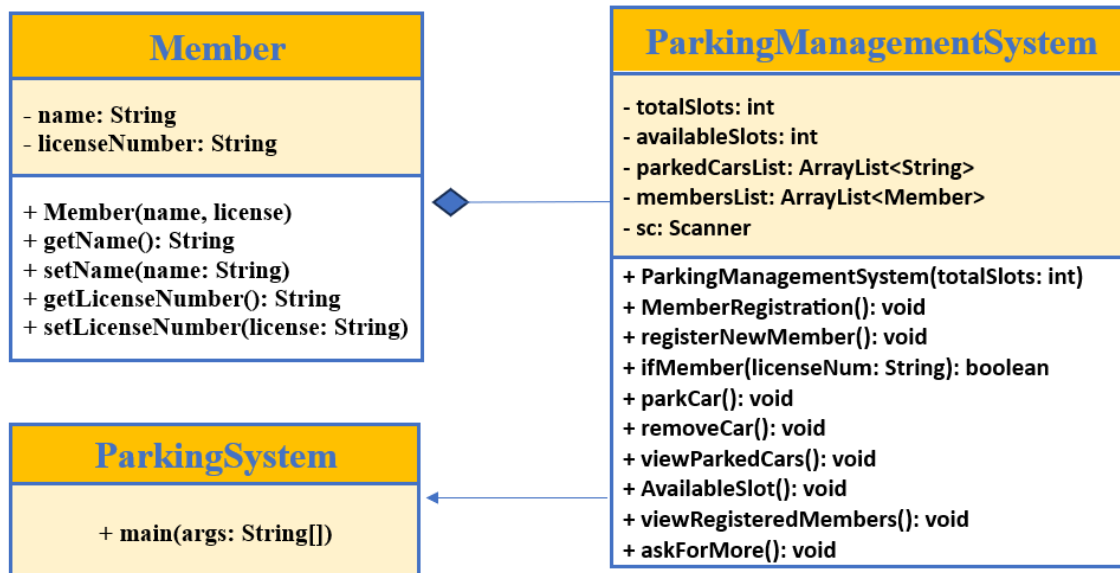
- **Member Registration:** Allows users to register with their name and license number.
- **Car Parking:** Users can park their cars if slots are available and they are registered members.
- **Car Removal:** Allows users to remove their car from the parking lot.
- **View Parked Cars:** Displays the list of currently parked cars.
- **Available Slots:** Displays the number of available parking slots.
- **View Registered Members:** Displays the list of all registered members.

#### Non-Functional Requirements

- **Performance:** The system should handle a reasonable number of parking slots and users efficiently.
- **Security:** Ensures only registered members can park or remove cars.
- **Scalability:** The system can accommodate more parking slots and users.
- **Usability:** The system should be easy to use with a simple interface and clear instructions.

## 4. System Design

### UML Diagrams



### System Architecture

The system follows a modular structure:

1. **Member Management:** Handles registration and authentication.
2. **Parking Management:** Manages parking slot allocation and tracking.
3. **User Interface:** Provides an interactive console-based interface.

## 5. Implementation

### Technologies Used

- **Programming Language:** Java
- **Development Tools:** NetBeans, Eclipse

- **Data Structures:** ArrayList for managing member and vehicle records

### Key Modules and Code Structure

- **Member Class:** Stores and manages user details.
- **ParkingManagementSystem Class:** Controls all parking operations.
- **ParkingSystem Class:** Acts as the main program driver.

### Multi-threading and Networking

The current implementation does not involve multi-threading or networking since it is a simple application that runs on a single machine. However, these features can be added for handling concurrent parking requests and remote access in a future version.

## 6. Testing and Error Handling

### Error Handling

- Prevents duplicate member registration.
- Ensures only registered members can park vehicles.
- Avoids duplicate car parking.
- Implements user input validation to prevent errors.

## 7. Challenges and Solutions

### Challenges

- Preventing unauthorized parking.
- Tracking available slots efficiently.
- Handling incorrect user inputs.

### Solutions

- Implemented member validation before allowing parking.
- Used ArrayLists for efficient slot tracking.
- Incorporated error-handling mechanisms.

## 8. Conclusion and Future Enhancements

### Conclusion

The Parking Management System successfully automates parking operations, providing an efficient and structured solution. By leveraging OOP principles, the system is modular, scalable, and easy to maintain.

### Future Enhancements

- **GUI Implementation:** Develop a user-friendly graphical interface.
- **Database Integration:** Store member and vehicle data persistently.
- **Multi-threading Support:** Allow concurrent operations for large-scale use.
- **Remote Access:** Extend functionality for mobile or web-based management.

## 9. References

- Java Documentation
- Object-Oriented Programming Concepts
- Online Java Tutorials