SMART PARKING SYSTEM

A PROJECT REPORT

Submitted by

VIJAYESWARAN V S (23038117104241177)

in partial fulfillment of requirements for the award of the course

CGB1201 - JAVA PROGRAMMING

In

COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY
(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

NOVEMBER- 2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on **"SMART PARKING SYSTEM"** is the bonafide work of **VIJAYESWARAN V S (2303811710421177)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

CGB1201-JAVA PROGRAMMING
Dr.A.DELPHIN CAROLINA RANI, M.E.,Ph.D.,
HEAD OF THE DEPARTMENT
PROFESSOR

SIGNATURE

Dr.A.Delphin Carolina Rani, M.E.,Ph.D.,

HEAD OF THE DEPARTMENT

PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram–621112.

CGB1201-JAVA PROGRAMMING
Mr.MALARMANNAN A, M.E.,
ASSISTANTP ROFESSOR

SIGNATURE

Mr. A. Malarmannan, M.E.,

SUPERVISOR

ASSISTANTP ROFESSOR

Department of CSE

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram–621112.

Submitted for the viva-voce examination held on…..06-12-2024…..

CGB1201-JAVA PROGRAMMING
Mr.M.SARAVANAN, M.E.,
INTERNAL EXAMINER
ASSISTANT PROFESSOR

INTERNAL EXAMINER

CGB1201-JAVA PROGRAMMING
Mr.R. KARTHIK, M.E.,
EXTERNAL EXAMINER
ASSISTANT PROFESSOR
8138-SCE, TRICHY.

EXTERNAL EXAMINER

DECLARATION

I declare that the project report on **"SMART PARKING SYSTEM"** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **CGB1201 - JAVA PROGRAMMING.**

.

Signature

VIJAYESWARAN V S

Place: Samayapuram

Date: 06/12/2024

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution "**K.Ramakrishnan College of Technology (Autonomous)**", for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. A. DELPHIN CAROLINA RANI, M.E.,Ph.D.,** Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **Mrs. A.MALARMANNAN , M.E.,** Department of **COMPUTER SCIENCE AND ENGINEERING,** for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To serve the society by offering top-notch technical education on par with global standards

MISSION OF THE INSTITUTION

- Be a center of excellence for technical education in emerging technologies by exceeding the needs of the industry and society.
- Be an institute with world class research facilities
- Be an institute nurturing talent and enhancing the competency of students to transform them as all-round personality respecting moral and ethical values

VISION OF DEPARTMENT

To be a center of eminence in creating competent software professionals with research and innovative skills.

MISSION OF DEPARTMENT

M1: Industry Specific: To nurture students in working with various hardware and software platforms inclined with the best practices of industry.

M2: Research: To prepare students for research-oriented activities.

M3: Society: To empower students with the required skills to solve complex technological problems of society.

PROGRAM EDUCATIONAL OBJECTIVES 1. PEO1: Domain Knowledge

To produce graduates who have strong foundation of knowledge and skills in the field of Computer Science and Engineering.

2. PEO2: Employability Skills and Research

To produce graduates who are employable in industries/public sector/research organizations or work as an entrepreneur.

3. PEO3: Ethics and Values

To develop leadership skills and ethically collaborate with society to tackle real-world challenges.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO 1: Domain Knowledge

To analyze, design and develop computing solutions by applying foundational concepts of Computer Science and Engineering.

PSO 2: Quality Software

To apply software engineering principles and practices for developing quality software for scientific and business applications.

PSO 3: Innovation Ideas

To adapt to emerging Information and Communication Technologies (ICT) to innovate ideas and solutions to existing/novel problems

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological

ABSTRACT

The Smart Parking System is a technological innovation designed to addressurban parking challenges by leveraging IoT, sensors, and advanced data analytics. It aims to reduce the time and effort spent searching for parking spaces while improving traffic flow and minimizing environmental impacts. The system uses real-time data from sensors installed in parking spaces to monitor availability, updating users through mobile or web applications.

Drivers can locate, reserve, and pay for parking spaces seamlessly, enhancing convenience and user experience. The system incorporates automated payment gateways, allowing cashless transactions and reducing the need for manual intervention. Dynamic pricing mechanisms can also be integrated to manage high-demand areas effectively.

By providing real-time data to municipal authorities, the system supports efficient parking management, identifying underutilized spaces and forecasting demand patterns. It also helps reduce congestion, lower vehicle emissions, and promote sustainable urban mobility.

The Smart Parking System is scalable and can be implemented in various settings, including commercial complexes, residential areas, and public parking facilities. Its adoption represents a significant step toward smarter, more sustainable cities, benefiting both users and urban planners by improving accessibility and resource utilization.

ABSTRACT WITH POs AND PSOs MAPPING

CO 5 : BUILD JAVA APPLICATIONS FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
The **Smart Parking System** leverages IoT and sensors to detect parking space availability in real-time, providing updates via mobile or web apps. It enables drivers to locate, reserve, and pay for spaces seamlessly, reducing search time and traffic congestion. Features like cashless payments, dynamic pricing, and navigation aids enhance convenience and efficiency. Scalable for malls, IT hubs, and public facilities, it promotes sustainable urban mobility by optimizing resource utilization and reducing emissions.	**PO1 -3** **PO2 -3** **PO3 -3** **PO4 -3** **PO5 -3** **PO6 -3** **PO7 -3** **PO8 -3** **PO9 -3** **PO10 -3** **PO11-3** **PO12 -3**	**PSO1 -3** **PSO2 -3** **PSO3 -3**

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Objective

The primary objective of a Smart Parking System is to optimize the utilization of parking spaces and enhance the overall parking experience for users. By leveraging advanced technologies such as IoT, sensors, and data analytics, the system aims to reduce the time drivers spend searching for parking, thereby decreasing traffic congestion and lowering vehicle emissions. It provides real-time updates on parking availability, allowing users to locate, reserve, and pay for spots conveniently through mobile or web platforms. Additionally, the system supports dynamic pricing to manage demand in high-traffic areas and offers valuable data insights for city planners to improve resource management. Scalable and adaptable, the Smart Parking System is designed to integrate with diverse urban infrastructures, contributing to smarter, more sustainable cities while improving accessibility, security, and convenience for all stakeholders. The aim of implementing Parking Management System is to reduce time and increase efficiency of the current Parking Management System. In overpopulated cosmopolitan zones, parking strategies must be wellimplemented for management of vehicles.

1.2 Overview

The Smart Parking System is a modern solution designed to address the growing challenges of parking in urban environments. By integrating technologies such as IoT, cloud computing, and data analytics, it offers a streamlined approach to parking management. Sensors installed in parking spaces detect real-time availability, while mobile applications guide users to vacant spots, allowing for reservations, navigation, and cashless payments. This system significantly reduces the time spent searching for parking, alleviates traffic congestion, and minimizes environmental impacts caused by idling vehicles. Beyond user convenience, it provides valuable data for authorities to optimize parking resources, forecast demand, and implement dynamic pricing. The Smart Parking System represents a critical step toward sustainable urban development, improving efficiency, accessibility, and the overall urban experience.

1.3 Java Programming Concepts

To build a Smart Parking System using Java, several programming concepts are essential. Here's an explanation of the key Java concepts used in designing such a system, without diving into code:

1. **Object-Oriented Programming (OOP)**

Encapsulation:

Encapsulation is about bundling related data and methods together. For a Smart Parking System, you can create classes like ParkingSpace, Reservation, and User. Each class contains attributes (e.g., a parking space's ID and status) and methods (e.g., to reserve or release a parking space). This keeps the data secure and only allows controlled access.

Inheritance: If there are different types of parking spaces (e.g., compact, electric, or large), you can use inheritance to define a general ParkingSpace class and let specific types inherit from it. This avoids redundancy and simplifies maintenance.

Polymorphism: Through polymorphism, you can handle various parking space types or payment methods uniformly. For instance, a method to calculate parking fees could work differently depending on whether the space is reserved for a car or a bike.

2. **Collections Framework**

Java's Collections Framework provides dynamic data structures to store and manage data efficiently.

- List: Use a list to manage all parking spaces, allowing easy addition and iteration.
- Map: A map can link parking space IDs to reservations or user details for quick lookup.
- PriorityQueue: A priority queue can be used to allocate parking spaces based on certain criteria, like proximity or size.

3. **Exception Handling**

Exception handling ensures the system can manage errors gracefully. For example:

If a user tries to reserve a parking space that is already occupied, the system can throw a meaningful error instead of crashing.Custom exceptions can be created for specific issues, such as invalid user input or payment failures.

4. **Multithreading and Concurrency**

In a real-world scenario, multiple users may try to reserve parking spaces simultaneously.mechanisms help handle this:

Synchronized Methods: Ensure that only one user can book a specific parking space at a time.

ExecutorService: Java's thread pool allows managing multiple tasks efficiently, such as processing

reservations and updating space availability.

5. Abstraction

Abstraction hides the internal workings of complex processes. For example:

A PaymentProcessor interface can abstract the logic of different payment methods (e.g., card, mobile wallet, or cash).

The system's user interface can interact with backend logic without exposing how parking space allocation works internally.

6. Database Integration

Databases are essential for persisting data, such as parking space details and user reservations.Java can connect to a database using JDBC (Java Database Connectivity). The database stores and,retrieves information like available spaces, reserved spaces, and payment history, ensuring the system remains consistent even if it restarts.

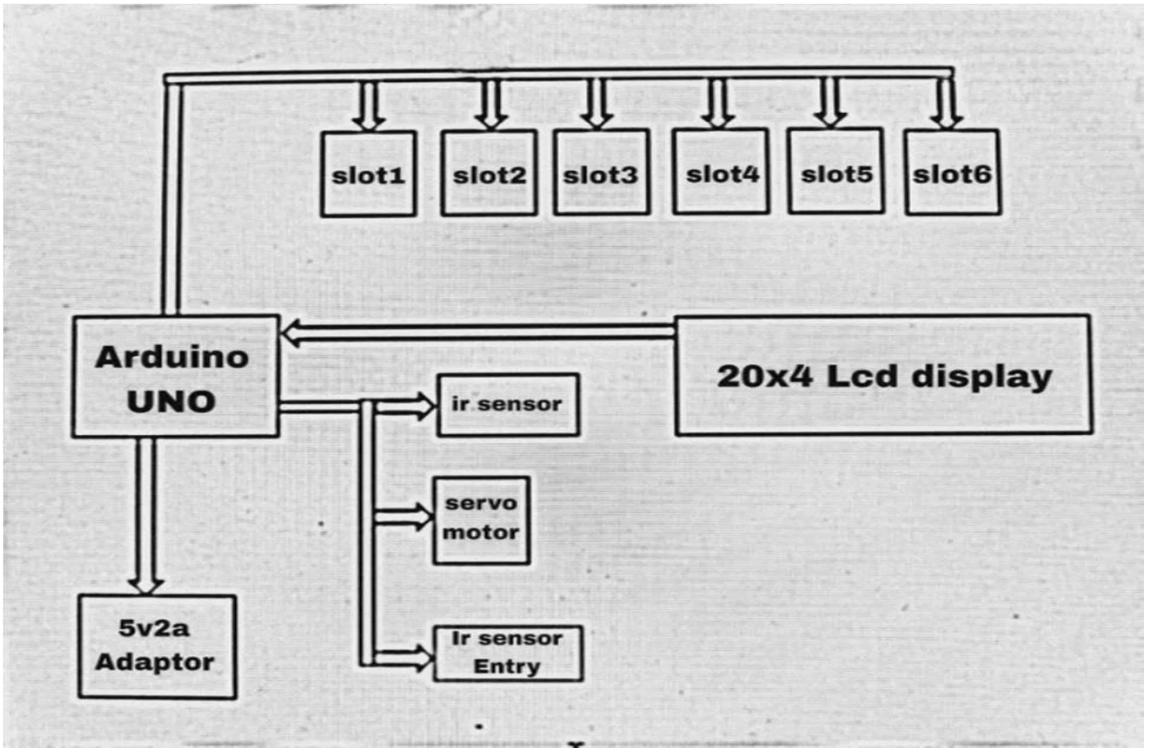
CHAPTER 2

PROJECT METHODOLOGY

2.1 Proposed Work

The proposed work for the Smart Parking System aims to develop an efficient, user-friendly, and automated solution for managing parking spaces. The system will leverage advanced Java programming concepts to provide seamless functionalities, including real-time space availability updates, user reservations, and payment processing. It will employ object-oriented principles to model parking spaces, users, and reservations, ensuring modularity and scalability. A database will be integrated for persistent storage of parking details, user data, and transaction records. The system will incorporate multithreading to handle concurrent user requests effectively and provide real-time notifications for available or occupied spaces using server-client communication. Additionally, a graphical user interface (GUI) will enable users to interact with the system easily, allowing them to view available spaces, reserve slots, and make payments. The ultimate goal is to optimize parking space utilization, reduce user wait times, and enhance overall parking management efficiency.

2.2 Block Diagram



CHAPTER 3

MODULE DESCRIPTION

3.1 User Management Module

- **Purpose:**

To manage user accounts, authentication, and personalized features.Functionality:

- **Registration and Login:**

Users can create accounts or log in through secure methods (email, phone number, or social media login). This allows the system to track their parking history, preferences, and payment details.

- **Profile Management:**

Users can update their profiles, including vehicle information (e.g., license plate number, vehicle type), preferences (e.g., favorite parking locations), and contact information.

- **Authentication:**

Ensures that only authorized users can access their personal parking information, making the system secure.Roles and Permissions: Allows for different types of users, such as customers and administrators, each with specific permissions to access and manage certain features (e.g., admins can view all reservations, while customes can only see their own).

- **Benefits:**

Secure and personalized experience for users.Tracks parking history and payment preferences for faster transactions.

Space Allocation:

Allocates a parking space to a user based on availability, user preferences (e.g., proximity, vehicle size), and rules (e.g., reserved or first-come-first-serve). Dynamic Availability Updates: As vehicles park or leave, the system continuously updates the availability status in real-time to inform other users.

- **Benefits:**

Maximizes parking space utilization by efficiently managing available spots.Ensures users are directed to the most suitable space based on their needs.

3.3 Reservation and Booking Module

- **Purpose:** To allow users to book parking spaces in advance, ensuring guaranteed availability

when they arrive.

- **Booking Process:** Users can search for available parking spaces, view prices, and reserve a space for a specific time. This can be done via the mobile app or website interface.
- **Timed Reservations:** The system allows users to specify the duration for which they need the parking space, helping to optimize space allocation.
- **Reservation History:** Tracks users' past bookings, helping them find and reuse their preferred parking spots.
- **Cancellation and Modifications:** Users can modify or cancel their reservation before arrival, ensuring that parking spaces are freed up for others if needed.
- **Benefits:**Guarantees users a parking space, reducing the frustration of finding an available spot.Reduces congestion and improves parking efficiency by ensuring spaces are used when they are needed.

3.4 Payment Processing Module

- **Purpose:** To securely handle transactions for parking reservations and usage.

Functionality:

- **Multiple Payment Methods:** The module supports various payment options, such as credit/debit cards, mobile wallets (e.g., PayPal, Google Pay), or even direct bank transfers. This allows users to pay for parking in the most convenient way.

- **Real-Time Pricing:** Calculates parking fees dynamically based on factors like time of day, duration of stay, and space type (e.g., premium spaces may cost more).

- **Secure Transactions:** Ensures that payments are securely processed, using encryption to protect sensitive user data.

- **Transaction History:** Provides users with a detailed receipt for each payment and allows them to view past transactions.
- **Benefits:**Facilitates easy and quick payments, enhancing user experience.

Ensures that the payment process is secure and efficient, reducing the chances of fraud.

3.5 Admin Dashboard Module

- **Purpose:** To give administrators full control over the parking system, monitoring activities, and managing users.

Functionality:

- **Space Management:** Admins can manually update space availability, add new parking spaces, and mark spaces as out of service if necessary.
- **User Monitoring:** Administrators can view all user accounts, reservations, and payment history, providing insights into user behavior and system performance.
- **Analytics and Reports:** Generates reports on parking space usage, revenue generation, peak times, and occupancy rates. This helps administrators optimize operations and pricing strategies.
- **Configuration Settings:** Admins can modify parking rates, set rules for reservations (e.g., maximum duration, early booking discounts), and configure the system's notifications.
- **Issue Resolution:** Administrators can address user complaints, parking disputes, and payment issues through the dashboard.

CHAPTER 5

CONCLUSION & FUTURE SCOPE

4.1 CONCLUSION

The Smart Parking System offers a modern solution to a common urban problem—finding and managing parking spaces efficiently. By integrating technology into the parking process, this system not only helps drivers locate available parking spots in real-time but also provides a more organized and optimized approach to managingparking resources.

In the context of our Java-based Smart Parking System, the system allows for:

1. Efficient Spot Management: Parking spots are dynamically tracked, ensuring that vehicles are parked in available spaces, minimizing the time spent searching for a spot.

2. Real-time Parking Status: The system provides real-time updates on the status of each parking spot (available or occupied), enabling better space utilization.

3. Fee Calculation: The system includes the capability to calculate parking fees based on the duration of the car's stay, promoting transparency and accountability.

4. Scalability and Extensibility: The system can be extended to include more sophisticated features such as sensor integration, mobile app support, and dynamic pricing models, making it adaptable for larger-scale deployments like multi-story parking lots or city-wide parking systems.

By automating and streamlining parking management, such systems contribute to:

- **Reduced Traffic Congestion:** Drivers spend less time searching for parking, which helps to reduce traffic and fuel consumption.
- **Better Resource Management:** Parking lots can be more effectively managed, maximizing space utilization and reducing operational costs

4.2 FUTURE SCOPE

The future scope of the Smart Parking System includes:

1. AI Integration: Predictive analytics for forecasting parking demand.
2. EV Charging Support: Integration with electric vehicle charging stations.
3. Blockchain Technology: Secure and transparent payment systems.
4. Smart City Connectivity: Integration with urban mobility solutions like public transport systems.
5. Advanced Automation: Use of robotics for valet parking and enhanced space management.

These advancements will further enhance efficiency, sustainability, and user convenience, aligning with smart city goals.

APPENDIX(SOURCE CODE)

```
 import java.util.Scanner;

class ParkingSpot {
    int spotId;
    boolean isOccupied;
    String vehicleNumber;

    public ParkingSpot(int spotId) {
        this.spotId = spotId;
        this.isOccupied = false;
        this.vehicleNumber = "";
    }

    public void parkCar(String vehicleNumber) {
        this.isOccupied = true;
        this.vehicleNumber = vehicleNumber;
    }

    public void removeCar() {
        this.isOccupied = false;
        this.vehicleNumber = "";
    }

    public String getStatus() {
        return isOccupied ? "Occupied by " + vehicleNumber : "Available";
    }
}

class ParkingLot {
    ParkingSpot[] spots;
    int totalSpots;

    public ParkingLot(int totalSpots) {
        this.totalSpots = totalSpots;
        this.spots = new ParkingSpot[totalSpots];
        for (int i = 0; i < totalSpots; i++) {
            spots[i] = new ParkingSpot(i + 1); // Parking spots numbered from 1
        }
    }

    public boolean parkCar(String vehicleNumber) {
        for (ParkingSpot spot : spots) {
            if (!spot.isOccupied) {
                spot.parkCar(vehicleNumber);
                System.out.println("Car parked at spot " + spot.spotId);
                return true;
            }
        }
```

```
        System.out.println("No available parking spots.");
        return false;
    }

    public boolean removeCar(String vehicleNumber) {
        for (ParkingSpot spot : spots) {
            if (spot.isOccupied && spot.vehicleNumber.equals(vehicleNumber)) {
                spot.removeCar();
                System.out.println("Car with number " + vehicleNumber + " removed from spot " + spot.spotId);
                return true;
            }
        }
        System.out.println("Car not found.");
        return false;
    }

    public void displayParkingLotStatus() {
        System.out.println("\nParking Lot Status:");
        for (ParkingSpot spot : spots) {
            System.out.println("Spot " + spot.spotId + ": " + spot.getStatus());
        }
    }

    public void calculateParkingFee(String vehicleNumber, int hoursParked) {
        final double RATE_PER_HOUR = 5.0; // Charge rate per hour
        double totalFee = hoursParked * RATE_PER_HOUR;
        System.out.println("Parking fee for vehicle " + vehicleNumber + ": $" + totalFee);
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ParkingLot parkingLot = new ParkingLot(10); // Create a parking lot with 10 spots

        while (true) {
            System.out.println("\n--- Smart Parking System ---");
            System.out.println("1. Park Car");
            System.out.println("2. Remove Car");
            System.out.println("3. Display Parking Lot Status");
            System.out.println("4. Calculate Parking Fee");
            System.out.println("5. Exit");
            System.out.print("Choose an option: ");

            int choice = scanner.nextInt();
            scanner.nextLine();  // Consume newline

            switch (choice) {
                case 1:
                    System.out.print("Enter vehicle number to park: ");
```

```
                String vehicleNumber = scanner.nextLine();
                parkingLot.parkCar(vehicleNumber);
                break;
            case 2:
                System.out.print("Enter vehicle number to remove: ");
                vehicleNumber = scanner.nextLine();
                parkingLot.removeCar(vehicleNumber);
                break;
            case 3:
                parkingLot.displayParkingLotStatus();
                break;
            case 4:
                System.out.print("Enter vehicle number: ");
                vehicleNumber = scanner.nextLine();



                System.out.print("Enter number of hours parked: ");
                int hoursParked = scanner.nextInt();
                parkingLot.calculateParkingFee(vehicleNumber, hoursParked);
                break;
            case 5:
                System.out.println("Exiting Smart Parking System.");
                scanner.close();
                return;
            default:
                System.out.println("Invalid option, please try again.");
        }
    }
  }
}
```

APPENDIX B
(SCREENSHOTS)

Output

```
--- Smart Parking System ---
1. Park Car
2. Remove Car
3. Display Parking Lot Status
4. Calculate Parking Fee
5. Exit
Choose an option: 2
Enter vehicle number to remove: AM2321
Car not found.

--- Smart Parking System ---
1. Park Car
2. Remove Car
3. Display Parking Lot Status
4. Calculate Parking Fee
5. Exit
Choose an option: 1
Enter vehicle number to park: AM2321
Car parked at spot 1

--- Smart Parking System ---
1. Park Car
2. Remove Car
3. Display Parking Lot Status
4. Calculate Parking Fee
5. Exit
```

REFERENCES

1. Ariponnammal, S. and Natarajan, S. (1994) SMART PARKING SYSTEM - X Asx', Pramana - Journal of Physics Vol.42, No.1, pp.421-425.
2. Barnard, R.W. and Kellogg, C. (1980) 'Applications SMART PARKING SYSTEM Theory', Michigan Mach, J.,
   Vol.27, pp.81-94.
3. Shin, K.G. and Mckay, N.D. (1984) 'Open Loop Minimum Time Control of M SMART PARKING SYSTEM Proc. Amer.Contr. Conf.,
   San Diego, CA, pp. 1231-1236.