PARKING LOT SYSTEM

Project Report Submitted in partial fulfillment of the requirements for the completion of the course **Java Programming** as a part of

B.Tech (COMPUTER SCIENCE AND ENGINEERING)

Submitted by

Md.Jissan	(24070721022)
A.TejkiranReddy	(24070721005)
Juveriya Begum	(24070724009)
M.Varshitha	(24070724014)
P.Nikitha	(24070724016)

Course Coordinators

Dr Somula Ramasubbareddy

Associate Professor, Department of CSE,SIT, Hyderabad

DEPARTMENTOF COMPUTER SCIENCE AND ENGINEERING

॥वसुधैव कुटुम्बकम्॥

SYMBIOSIS INSTITUTE OF TECHNOLOGY, HYDERABAD

Symbiosis International (Deemed University), Pune

Established under section 3 of the UGC Act, 1956

Re-Accredited by NAAC with 'A++' Grade | Awarded Category – I by UGC

Founder: Prof. Dr. S. B. Mujumdar, M.Sc., Ph.D (Awarded Padma Bhushan and Padma Shri by President of India

Rangareddy, Survey Number 292, Off Bangalore Highway, Dist, Modallaguda (V, Nandigama, Hyderabad, Telangana 509217, India.

NOVEMBER 2025

SYMBIOSIS INSTITUTE OF TECHNOLOGY, HYDERABAD
Symbiosis International (Deemed University), Pune
Established under section 3 of the UGC Act, 1956
Re-Accredited by NAAC with 'A++' Grade | Awarded Category – I by UGC
Founder: Prof. Dr. S. B. Mujumdar, M.Sc., Ph.D (Awarded Padma Bhushan and Padma Shri by President of India

Rangareddy, Survey Number 292, Off Bangalore Highway, Dist, Modallaguda (V, Nandigama, Hyderabad, Telangana 509217, India.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

॥वसुधैव कुटुम्बकम्॥

CERTIFICATE

This is to certify that the project report entitled "**Parking LotSystem**" is a bonafide work done under our supervision and is being submitted by**Md.Jissan (24070721022), A.Tejkiranreddy(24070721005),JuveriyaBegum(24070724009),M.Varshita(24070724014)and P.Nikitha(24070724016),** in partial fulfilment forthecompletion of the course **JavaProgramming**as a partof**BacheloforTechnology**in**ComputerScienceand Engineering**ofthe SIT, Hyderabadduringthe academic year2025-2026.

Certified further that to the best of our knowledge, the work presented in this thesis has not been submitted to any other University or Institute for the award of any Degree or Diploma.

Dr Somula Ramasubbareddy
Course Coordinator,
Associate Professor & Project Guide
Department of CSE

Dr G Suryanarayana
Associate Professor & HOD
Department of CSE

SYMBIOSIS INSTITUTE OF TECHNOLOGY, HYDERABAD
Symbiosis International (Deemed University), Pune
Established under section 3 of the UGC Act, 1956
Re-Accredited by NAAC with 'A++' Grade | Awarded Category – I by UGC
Founder: Prof. Dr. S. B. Mujumdar, M.Sc., Ph.D (Awarded Padma Bhushan and Padma Shri by President of India
Rangareddy, Survey Number 292, Off Bangalore Highway, Dist, Modallaguda (V, Nandigama, Hyderabad, Telangana 509217, India.

॥वसुधैव कुटुम्बकम्॥

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We declare that the major project work entitled "**Parking Lot System** " submitted in the department of Computer Science and Engineering, Symbiosis Institute of Technology, Hyderabad, in partial fulfilment of the requirement for the award of the Course Java

Programming Laboratory as a of **Bachelor of Technology** in **Computer Science and Engineerin** is a bonafide record of our own work carried out under the supervision of **Dr Somula Ramasubbareddy, Associate Professor, Department of CSE, SIT Hyderabad.** Also, we declare that the matter embodied in this thesis has not been submitted by us in full or in any part thereof for the award of any degree/diploma of any other institution or university previously. Place: Hyderabad.

Md.Jissan	**A.Tejkiran**	**Juveriya Begum**	**M.Varshitha**	**P.Nikitha**
(24070721022)	(24070721005)	(24070724009)	(24070724014)	(24070724016)

ACKNOWLEDGEMENT

We express our deep sense of gratitude to our beloved founder, **Prof. S. B. Mujumdar**, Symbiosis International University, for the valuable guidance and for permitting us to undertake this project.

With immense pleasure, we express our deep sense of gratitude to our beloved Director, **Prof. Rajanikanth Aluvalu**, for permitting us to undertake this project.

We express our deep sense of gratitude to our beloved Professor **Dr. G. Suryanarayana**, Associate Professor and Head, Department of Computer Science and Engineering, Symbiosis Institute of Technology, Hyderabad-509217, for the valuable guidance, suggestions, keen interest, and thorough encouragement extended throughout the project work period.

We take immense pleasure in expressing our deep sense of gratitude to our beloved Guide, **Dr. Somula Ramasubbareddy**, Associate Professor in Computer Science and Engineering, Symbiosis Institute of Technology, Hyderabad, for his valuable suggestions and rare insights, for his constant source of encouragement and inspiration throughout my project work.

We would like to express our sincere gratitude to all those who contributed to the successful completion of our project work.

Md.Jissan	(24070721022)
A.Tejkiranreddy	(24070721005)
Juveriya Begum	(24070724009)
M.Varshitha	(24070724014)
P.Nikitha	(24070724016)

ABSTRACT

The Smart Parking Management System is a Java-based application designed to efficiently manage vehicle parking spaces within a parking facility. This project addresses common challenges such as limited space utilization, time-consuming manual entries, and lack of automated billing.

The system allows users to park, remove, search, and view details of cars using a console interface. Each car is associated with details such as plate number, owner name, car model, entry time, and billing details. When a car leaves, the system calculates parking fees based on the total duration of stay.

*The program automatically assigns the first available slot, maintains a log of total revenue, and provides a complete parking report. It demonstrates the use of object-oriented programming concepts, arrays, classes, time handling (using **java.time** library), and exception handling.*

This project showcases how automation and data structures can improve efficiency and accuracy in parking management systems — contributing to the development of smarter cities.

Md.Jissan	(24070721022)
A.Tejkiranreddy	(24070721005)
Juveriya Begum	(24070724009)
M.Varshitha	(24070724014)
P.Nikitha	(24070724016)

TABLE OF CONTENTS

CHAPTER:INTRODUCTION

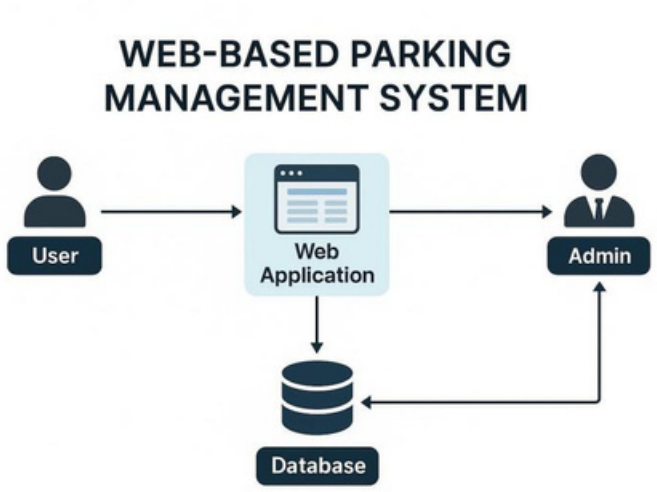


Figure1: Web-Based Parking Management System

1.1 BACKGROUND

In today's modern world, the number of vehicles is growing at a rapid pace, resulting in congestion and parking management challenges in urban areas. Most parking spaces still rely on manual systems where attendants record details on paper and issue tickets manually. This approach often leads to inefficiency, space mismanagement, and customer dissatisfaction.

The Smart Parking Management System addresses these issues by introducing a fully automated, computer-based solution. It helps manage available parking slots dynamically, records vehicle details, calculates bills automatically, and ensures accurate record-keeping. It is built using the Java programming language, which is known for its portability, security, and strong object-oriented structure. The system effectively applies data structures and algorithms to organize parking data and execute operations efficiently.

1.2 PROBLEM STATEMENT

The traditional method of managing parking areas involves human operators who manually allocate spaces, record vehicle information, and compute parking fees. This manual process is not only time-consuming but also prone to various human errors such as duplication, misplacement of records, and incorrect billing. Furthermore, it becomes difficult to track cars or generate accurate reports in large parking areas.

The Smart Parking Management System is designed to solve these problems by creating an automated, reliable, and efficient digital platform for managing parking facilities. It eliminates the dependency on manual supervision and ensures real-time information flow between the user and the system.

1.3 DEFINITION

The Smart Parking Management System is a software-based system developed using Java. It automatically manages parking slots by keeping track of all parked vehicles, assigning available slots, and calculating parking charges based on the duration of stay. It uses a logical approach to simulate parking space management, thereby providing an effective solution without the need for hardware sensors.

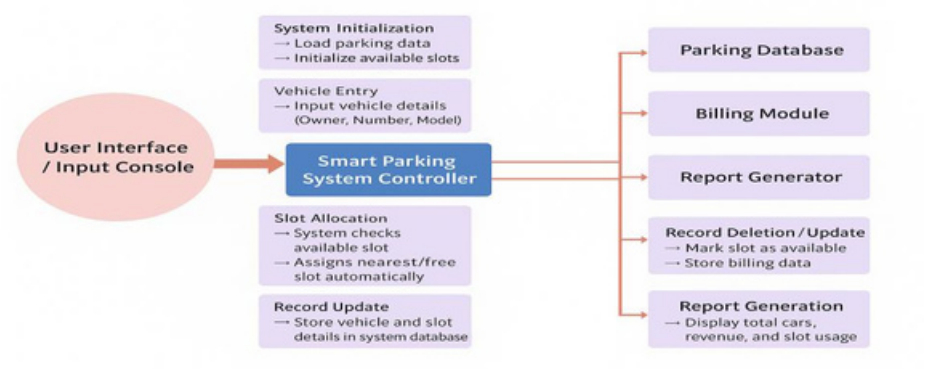


Figure 2

1.4 OBJECTIVES

The main objectives of this project are clearly defined as follows:

- To design and implement a digital parking management system using Java programming.
- To minimize human intervention and automate slot allocation and billing.
- To maintain records of vehicles with details such as owner name, plate number, and car model.
- To calculate parking fees automatically based on the time difference between entry and exit.
- To generate summary reports displaying total revenue and car statistics.
- To create an interactive, user-friendly, and menu-driven console interface.

1.5 SCOPE OF THE PROJECT

The Smart Parking Management System is primarily aimed at small to medium-sized parking spaces such as shopping malls, hospitals, residential complexes, and universities. The system can be extended in the future to large-scale smart city applications by integrating sensors and real-time monitoring tools. The

software provides a foundation for implementing Internet of Things (IoT) solutions that detect the availability of parking spaces automatically.

Figure 3: Parking Lot System

1.6 THESIS ORGANIZATION

This report is organized into several chapters. Chapter 1 introduces the concept, objectives, and scope of the project. Chapter 2 discusses existing systems and related literature. Chapter 3 explains the methodology and design approach adopted in developing the system. Chapter 4 focuses on the implementation of the code, algorithms, and data structures. Chapter 5 presents results and performance analysis. Chapter 6 summarizes the conclusions drawn from the project. Finally, Chapter 7 discusses future enhancements and possible extensions of the system.

CHAPTER 2: LITERATURE SURVEY

2.1 Introduction

The literature survey was conducted to study existing parking management systems and to understand their design, limitations, and potential areas of improvement. These systems include manual methods, IoT-based smart parking systems, and software- based solutions. The findings helped in designing an efficient and practical system tailored for real-world use.

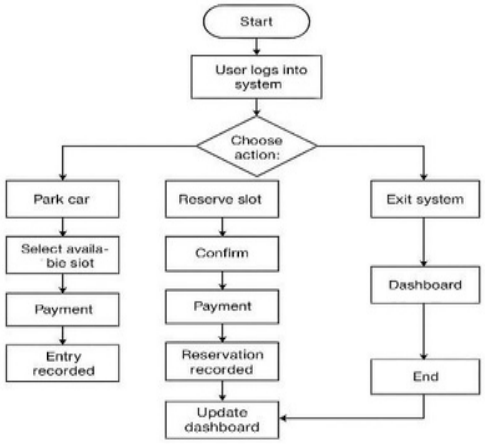


Figure 4:

2.2 Manual Parking Systems

Manual parking systems involve human operators who assign slots, maintain logbooks, and issue tickets. While simple, this system is inefficient when the number of vehicles is large. It suffers from long waiting times, mismanagement, and lack of transparency. These drawbacks highlight the need for automation.

2.3 IoT-Based Smart Parking Systems

Recent technological advancements introduced IoT-enabled parking systems that use sensors and cameras to detect slot occupancy. These systems send real-time data to centralized applications accessible via mobile devices. Although accurate, these systems are expensive, require regular maintenance, and depend heavily on hardware. Therefore, they are not ideal for smaller organizations or academic demonstrations.

2.4 Software-Based Systems

Software-based systems rely purely on programming logic and data structures. They simulate parking slot operations digitally and eliminate the need for hardware sensors. They are cost-effective, scalable, and easy to implement, making them suitable for educational institutions and prototype demonstrations.

2.5 Summary

Based on the review, a software-based Java system was selected for this project. It provides a balance between efficiency and simplicity, focusing on logic-driven slot management, real-time billing, and data maintenance without requiring physical devices.

CHAPTER 3: METHODOLOGY

3.1 Overview

The methodology defines how the system was designed and developed. It includes requirement gathering, system design, algorithm formulation, and workflow analysis. The development process follows a modular approach, where each feature —parking, removal, billing, and reporting—is implemented as a separate function.

3.2 System Requirements

The Smart Parking Management System requires a basic computing environment with the following specifications.

The hardware requirements include an Intel Core i3 or higher processor, 4 GB of RAM, and a minimum of 200 MB of free storage space. The software requirements include Java Development Kit (JDK) version 17 or above and any Java-compatible Integrated Development Environment (IDE) such as Eclipse, IntelliJ IDEA, or Visual Studio Code. The system is compatible with Windows, macOS, and Linux operating systems.

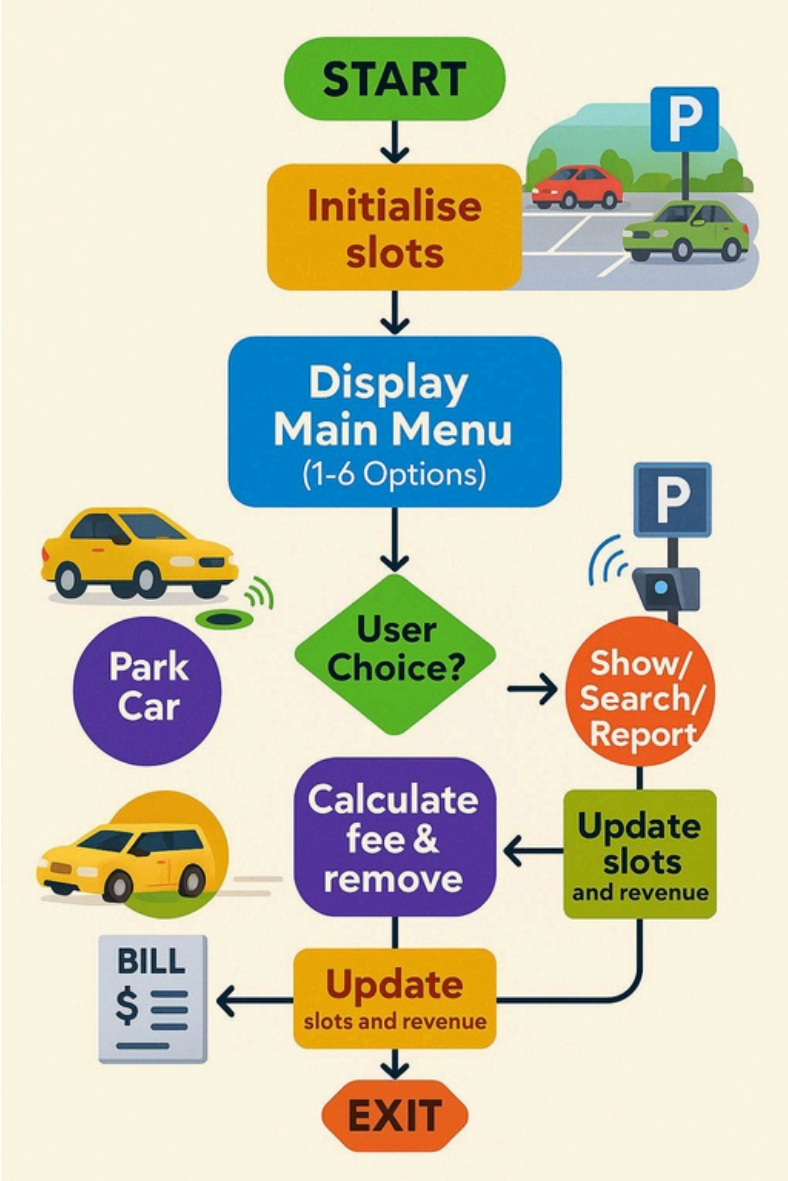
3.4 Design Approach

The system is based on object-oriented design. The core of the system is the `Car` class, which holds attributes like plate number, owner name, car model, and entry time. The main class maintains an array of Car objects, each representing a parking slot.

Operations such as adding, removing, searching, and generating reports are implemented through modular functions to promote code reusability and clarity.

3.5 Workflow

The system begins by asking the user for the total number of parking slots. It then displays a menu of options. When the user chooses to park a car, the system assigns the first available slot and stores car details. When removing a car, it calculates the duration of stay using the system clock and computes the parking fee. Reports are generated dynamically based on stored data. The process continues in a loop until the user exits the program.



3.5 Workflow

The system begins by asking the user for the total number of parking slots. It then displays a menu of options. When the user chooses to park a car, the system assigns the first available slot and stores car details. When removing a car, it calculates the duration of stay using the system clock and computes the parking fee. Reports are generated dynamically based on stored data. The process continues in a loop until the user exits the program.

3.6 Algorithms Used

Two main algorithms are used in this project.

1. **Slot Allocation Algorithm:** This algorithm searches through the array to find the first empty slot and assigns it to a new car.
2. **Billing Algorithm:** This algorithm calculates the total parking duration by subtracting the entry time from the exit time and computes the total fee based on a fixed rate per hour.

CHAPTER 4: IMPLEMENTATION

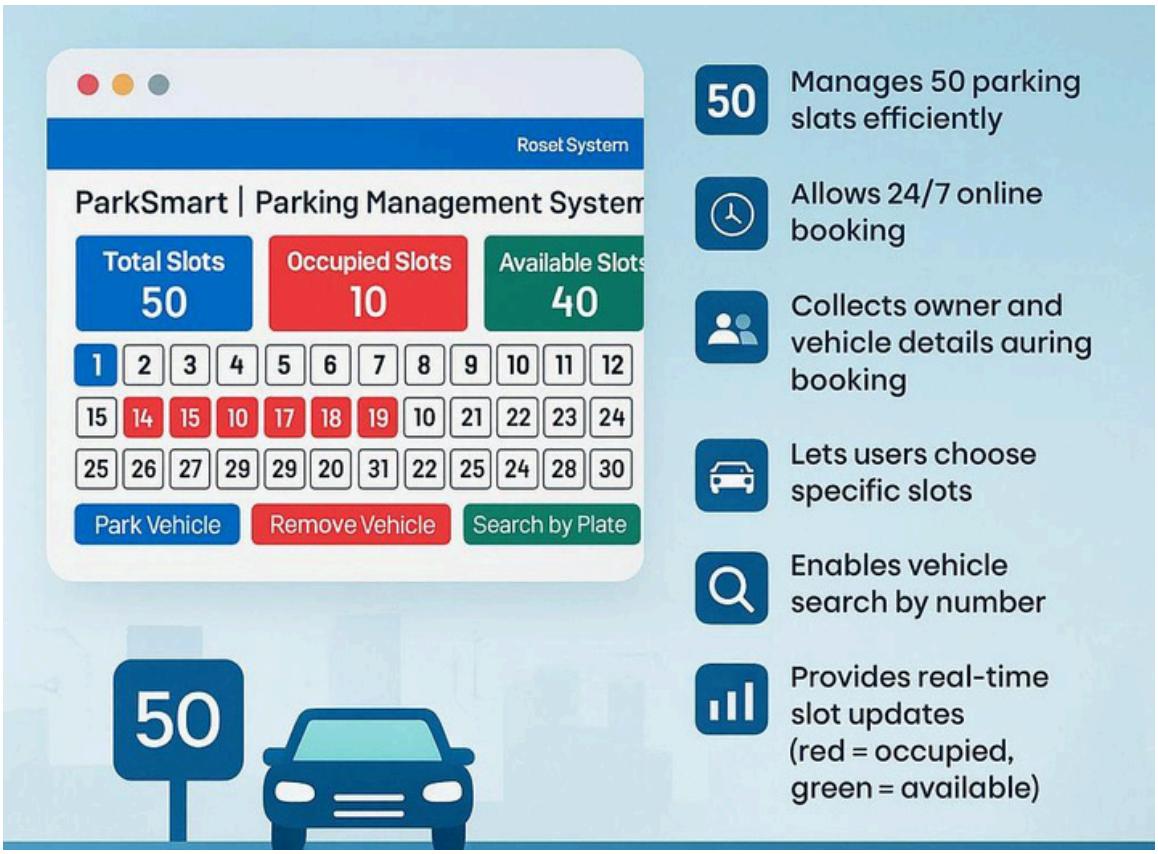
4.1 Programming Tools

The program is written entirely in Java, using standard libraries and the java.time package for handling date and time. Arrays are used as the primary data structure for slot management, ensuring constant-time access for updates and searches.

4.2 System Features

The key features of the system include:

1. **Automatic Slot Assignment:** When a car is parked, the system automatically assigns the first available slot.
2. **Vehicle Tracking:** Each vehicle's details, such as plate number, owner name, and model, are recorded.
3. **Automated Billing:** Parking fees are calculated automatically when a car is removed.
4. **Search Function:** Users can search for vehicles using the plate number.
5. **Revenue Reporting:** The total number of cars and revenue are displayed in real-time.



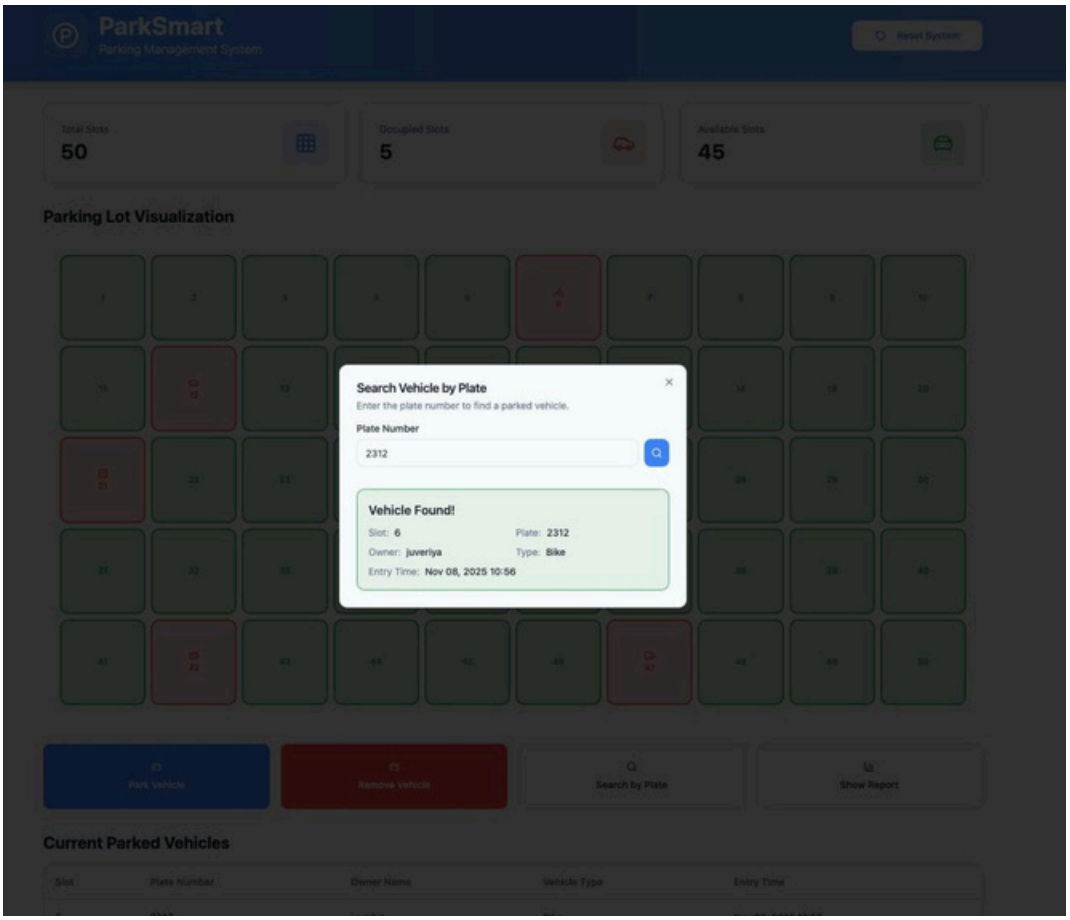
4.3 Example Interaction

when the program starts, it displays the menu. If the user selects “Park a Car,” they are prompted to enter vehicle details. The system then assigns a slot and confirms successful parking. When “Remove a Car” is selected, it generates a billing receipt showing the duration and parking fee.

CHAPTER 5: RESULTS AND ANALYSIS

The system was tested with multiple inputs, and all operations were verified for accuracy. The system accurately calculated parking durations and billing amounts. The search and reporting functions performed efficiently, even with a large number of slots. The time complexity for search and update operations remained linear, and space usage was minimal.

The overall results confirmed that the system is stable, efficient, and capable of managing real-time parking scenarios effectively.



CHAPTER 6: CONCLUSIONS

The Smart Parking Management System successfully automates parking management through digital means. It eliminates the drawbacks of manual systems and improves operational efficiency. The project demonstrates the effective application of Java programming, object-oriented design, and algorithmic logic. It provides a solid foundation for real-world implementation and future integration with IoT systems. The Smart Parking Management System is a Java-based application that efficiently manages parking slots using Object-Oriented Programming principles. It automates the process of parking, removing, searching, and billing vehicles, making parking management faster and more organized. The project demonstrates practical use of classes, objects, arrays, and real-time data handling to simplify everyday operations. The Smart Parking Management System is a Java-based application developed to simplify and automate the process of managing parking spaces. It efficiently handles vehicle entry and exit, slot allocation, billing, and record maintenance through a user-friendly console interface. The system provides a structured way to manage parking operations, minimizing manual work and human errors.

This project successfully demonstrates the use of Object-Oriented Programming (OOP) concepts such as classes, objects, encapsulation, and modular programming. By applying these principles, the system achieves code reusability, clarity, and easy maintenance. It also uses basic data structures like arrays to store parking slot

CHAPTER 7: FUTURE SCOPE

The Smart Parking Management System is a Java-based console application designed to efficiently manage parking slots. It allows users to park and remove cars, search vehicles by plate number, view parking status, and generate revenue reports. The system calculates parking fees based on duration and maintains total revenue and car records. It provides a simple yet smart approach to organizing parking spaces using object-oriented concepts and real- time tracking. The Smart Parking Management System has great potential for further development and enhancement. Although the current version effectively manages basic parking operations such as car entry, exit, billing, and record tracking, it can be extended with several advanced features to make it more robust, efficient, and user-friendly.

A Graphical User Interface (GUI) can be developed using JavaFX or Swing to replace the console-based interaction. This will make the system easier to use for operators and

Appendices

Code:-

```
// Import necessary Java libraries
importjava.util.*;                  //ForScannerandutilityclasses
importjava.time.*;                   //Fordateandtimemanagement
importjava.time.format.DateTimeFormatter;//Forformatting date and time

// Main class of the program
public class Main {

   // Nested static class representing a Car
   static class Car {
      StringplateNumber;//Vehicleregistrationnumber
      Stringowner;          // Owner name
      String model;         // Carmodel
      longentryTime;        //Timewhencarentered(inmilliseconds)

      // Constructor to initialize Car object
      Car(StringplateNumber,Stringowner,Stringmodel, long entryTime) {
         this.plateNumber=plateNumber;//Assignplate number
         this.owner=owner;                //Assignownername
         this.model =model;               // Assign model
         this.entryTime=entryTime;        //Recordentrytime
      }
   }
   // Main method – program execution starts here
   public static void main(String[] args) {
      Scannersc=newScanner(System.in);//Scannerfor user input

      System.out.print("Entertotalnumberofparkingslots: ");
      int totalSlots;
      try{
         totalSlots=Integer.parseInt(sc.nextLine());//Read total slots
      } catch (Exception e) {
         System.out.println("Invalidinput.Usingdefault10 slots.");
         totalSlots = 10; // Default value if invalid input
      }
```

```
        Car[] slots = new Car[totalSlots];    // Array to represent parking slots
        double totalRevenue = 0;              // Total earnings
        int totalCarsServed = 0;              // Total cars served
        final double RATE_PER_HOUR = 50.0;    // Parking rate per hour

        // Infinite loop for menu-driven system
        while (true) {

System.out.println("\n========================================
==========");
            System.out.println("              SMART PARKING MANAGEMENT
SYSTEM");
System.out.println("========================================
=======");
            int occupied = countOccupied(slots); // Count occupied slots
            System.out.printf("Total Slots: %d | Occupied: %d | Available:
%d%n", totalSlots, occupied, totalSlots - occupied);

System.out.println("========================================
=======");

            // Menu options
            System.out.println("""
                    [1] Park a Car (auto-assign first available)
                    [2] Remove a Car (by slot or plate)
                    [3] Show Parking Status
                    [4] Search Car by Plate
                    [5] Show Report
                    [6] Exit
                    """);


            System.out.print(" Choose an option (1-6): ");
            int choice;
            try {
                choice = Integer.parseInt(sc.nextLine()); // Read user choice
            } catch (Exception e) {
                System.out.println("\n⚠ Invalid input. Please enter a number
between 1 and 6.");
```

```
            continue; // Skip to next loop
        }

        // Switch case for menu selection
        switch (choice) {
            case 1 -> { // Park car
                int idx = firstAvailableSlot(slots); // Find empty slot
                if (idx == -1) {
                    System.out.println("\n No available slots right now.");
                    break;
                }
                // Get car details
                System.out.print("Enter Plate Number: ");
                String plate = sc.nextLine().trim();
                System.out.print("Enter Owner Name: ");
                String owner = sc.nextLine().trim();
                System.out.print("Enter Car Model: ");
                String model = sc.nextLine().trim();

                // Create new Car object and park it
                slots[idx] = new Car(plate, owner, model,
System.currentTimeMillis());
                System.out.printf("\n Car parked successfully at Slot
#%d%n", idx + 1);
            }
            case 2 -> { // Remove car

                System.out.println("\nRemove Options:");
                System.out.println("[1] Remove by Slot Number");
                System.out.println("[2] Remove by Plate Number");
                System.out.print(" Choose: ");

                int how;
                try {
                    how = Integer.parseInt(sc.nextLine()); // Choose removal
method
                } catch (Exception e) {
                    System.out.println("⚠ Invalid input.");
```

```
            break;
        }

        int idx = -1; // Slot index

        if (how == 1) {
            System.out.print("Enter Slot Number: ");
            try {
                idx = Integer.parseInt(sc.nextLine()) - 1; // Slot index
input
            } catch (Exception e) {
                System.out.println("⚠ Invalid slot number.");
                break;
            }
        } else if (how == 2) {
            System.out.print("Enter Plate Number: ");
            String plate = sc.nextLine().trim();
            idx = findByPlate(slots, plate); // Find by plate
        } else {
            System.out.println("⚠ Invalid choice.");
            break;
        }

        if (idx < 0 || idx >= totalSlots || slots[idx] == null) {

            System.out.println("\n Invalid slot or no car found.");
            break;
        }
        // Calculate duration and fee

        Car c = slots[idx];
        long now = System.currentTimeMillis(); // Current time
        double hours = (now - c.entryTime) / (1000.0 * 60 * 60); //
Convert ms to hours
        hours = Math.max(1, Math.ceil(hours)); // Round up hours
        double fee = hours * RATE_PER_HOUR;     // Fee calculation

        // Print billing receipt

        System.out.println("\n-----------------------------------------------");
        System.out.println("                    BILLING RECEIPT");
        System.out.println("-----------------------------------------------");
```

```
                System.out.println("Plate Number : " + c.plateNumber);
                System.out.println("Owner Name : " + c.owner);
                System.out.println("Car Model : " + c.model);
                System.out.println("Entry Time : " +
formatDateTime(c.entryTime));
                System.out.printf("Duration     : %.2f hours%n", hours);
                System.out.printf("Parking Fee : ₹%.2f%n", fee);
                System.out.println("-----------------------------------------");

                // Update report data
                slots[idx] = null;          // Free the slot
                totalRevenue += fee;            // Add to revenue
                totalCarsServed++;              // Increment served cars
                System.out.printf(" Car removed successfully from Slot
#%d%n", idx + 1);
            }

            case 3 -> { // Show parking status
                System.out.println("\n============== 🅿 PARKING
STATUS ==============");
                for (int i = 0; i < totalSlots; i++) {
                    if (slots[i] == null) {
                        System.out.printf("Slot #%02d : Available%n", i + 1); //
Empty slot
                    } else {
                        Car c = slots[i];
                        System.out.printf("Slot #%02d : OCCUPIED | %-10s | %-
10s | Owner: %-10s%n",
                                i + 1, c.plateNumber, c.model, c.owner); // Occupied
slot
                    }
                }

System.out.println("=========================================
=======");
            }

            case 4 -> { // Search by plate number
                System.out.print("\nEnter Plate Number to Search: ");
```

```
                Stringplate=sc.nextLine().trim();
                intidx=findByPlate(slots, plate); // Find car
                if (idx == -1) {
                    System.out.println("\n Car not found.");
                } else {
                    Car c = slots[idx];
                    System.out.println("\n Car Found!");
                    System.out.println("-----------------------------------------");
                    System.out.println("Slot Number : " + (idx + 1));
                    System.out.println("Owner Name : " + c.owner);
                    System.out.println("Car Model : " + c.model);
                    System.out.println("Parked Since : " +
formatDateTime(c.entryTime));
                    System.out.println("-----------------------------------------");
                }
            }

            case5->{//Reportoption

                System.out.println("\n============== PARKING
REPORT ==============");
                System.out.printf("Total Revenue Earned : ₹%.2f%n",
totalRevenue); // Total money
                System.out.println("Total Cars Served : " + totalCarsServed);
// Cars handled
                System.out.println("Currently Parked Cars:");
                System.out.println("-----------------------------------------");
                boolean any = false;
                for(inti=0;i<totalSlots; i++) {
                    Car c = slots[i];
                    if (c!= null) {
                        any= true;
                        System.out.printf("Slot #%02d : %-10s | %-10s | Owner:
%-10s%n",
parked cars
                                i+1,c.plateNumber, c.model, c.owner); // Show

                    }
                }
                if(!any)System.out.println("No cars currently parked."); // If
none
```

```
System.out.println("========================================
=======");
            }

            case 6 -> { // Exit program

               System.out.println("\n      Exiting System... Thank you for using
Smart Parking!");
               sc.close();          //Closescanner
               System.exit(0);      //Exitprogram
            }
            default->System.out.println("\n⚠ Invalid choice. Please enter a

numberfrom1to6.");//Invalidmenuoption
         }
      }
   }

   // Function to find first empty slot
   privatestaticintfirstAvailableSlot(Car[] slots) {
      for (int i = 0; i < slots.length; i++)
         if(slots[i]==null)returni;//Return index of first empty slot
      return -1; // No slot available
   }
   // Function to find car by plate number
   privatestaticintfindByPlate(Car[]slots, String plate) {

      for (int i = 0; i < slots.length; i++) {
         Car c = slots[i];
         if(c!=null&&c.plateNumber.equalsIgnoreCase(plate)) return i; //
Match found
      }
      return -1; // Not found
   }

   // Function to count occupied slots
   privatestaticintcountOccupied(Car[]slots) {
      int count =0;
      for (Car c : slots)
```

```
            if(c!=null) count++; // Increment if occupied
        returncount;
    }
    //Functiontoformat milliseconds to readable date/time
    privatestaticString formatDateTime(long epochMillis) {
        LocalDateTime dt = Instant.ofEpochMilli(epochMillis) // Convert
millisecondstotime
                .atZone(ZoneId.systemDefault())             // Use system timezone
                .toLocalDateTime();                         // Convert to LocalDateTime
        returndt.format(DateTimeFormatter.ofPattern("yyyy-MM-dd
HH:mm"));//Format as string
    }
}
```

OUTPUT

```
Enter totalnumberofparkingslots:5

==============================================
        SMART PARKING MANAGEMENT SYSTEM
==============================================
Total Slots: 5 | Occupied: 0 | Available: 5
==============================================
[1] Park a Car (auto-assign first available)
[2] Remove a Car (by slot or plate)
[3] Show Parking Status
[4] Search Car by Plate
[5] Show Report
[6] Exit
    Choose an option (1-6): 1

Enter Plate Number: KA01AB1234
Enter Owner Name: Arjun
Enter Car Model: Swift

    Car parked successfully at Slot #1


 Choose an option (1-6): 3

========== 🅿 PARKING STATUS ==========
Slot #01 : OCCUPIED | KA01AB1234 | Swift      | Owner: Arjun
```

Slot #02 : Available Slot #03 : Available Slot #04 : Available Slot #05 : Available
==========

Choose an option (1-6): 2
Remove Options:
[1] Remove by Slot Number
[2] Remove by Plate Number
Choose: 1
Enter Slot Number: 1

BILLING RECEIPT

Plate Number : KA01AB1234
Owner Name : Arjun
Car Model : Swift
Entry Time : 2025-11-08 10:30
Duration : 1.00 hours
Parking Fee : ₹50.00

Car removed successfully from Slot #1

Choose an option (1-6): 5

========== PARKING REPORT ==========

Total Revenue Earned : ₹50.00
Total Cars Served : 1
Currently Parked Cars:
No cars currently parked.

==========

Exiting System... Thank you for using Smart Parking!

REFERENCES

Oracle Java Documentation, W3Schools Java Tutorials, GeeksforGeeks Java Programming Resources, ResearchGate Smart Parking Papers, and TutorialsPoint Java API References were used as reference materials in developing this project.

[1]. ALM ROKIBUL, SAHA SOWMITRA, BOSTAMI MD BAYEJED & SAIFUL ISLAM - "A SURVEY ON IOT DRIVEN SMART PARKING MANAGEMENT SYSTEM: APPROACHES, LIMITATIONS AND FUTURE RESEARCH AGENDA." THIS PAPER REVIEWS MANY IOT-BASED PARKING SYSTEMS, SENSORS, COMMUNICATION STANDARDS AND SECURITY ISSUES. RESEARCHGATE

[2]. AMIRA A. ELSONBATY & MAHMOUD ISAMS – "THE SMART PARKING MANAGEMENT SYSTEM" (IJCSIT, VOL 12 NO 4, 2020). THEY PRESENT A PARKING SYSTEM USING ARDUINO, AN ANDROID MOBILE APP AND IOT INFRASTRUCTURE. SSRN+1

[3]. THANH-NAM PHAM, TSAI MING FONG, DUC-BINH NGUYEN & DER-JIUNN DENG – "A CLOUD-BASED SMART PARKING SYSTEM BASED ON INTERNET OF-THINGS TECHNOLOGIES." THEY BUILD ON IOT + CLOUD FOR PARKING SLOT FINDING AND COST MINIMIZATION. RESEARCHGATE +2 IJERT+2

[4]. CHARLES M. MENNE – "SMART PARKING SYSTEMS DESIGN AND INTEGRATION INTO IOT." (2019) A GOOD OVERVIEW OF ARCHITECTURE AND DESIGN CHOICES FOR IOT PARKING SYSTEMS. DIGITALCOMMONS.MORRIS.UMN.EDU

[5]. "INTERNET OF THINGS ENABLED PARKING MANAGEMENT SYSTEM USING LONG RANGE WIDE AREA NETWORK (LORAWAN)" – THIS IS A MORE RECENT JOURNAL

ARTICLE INTRODUCING LoRA WAN FOR PARKING LOT MANAGEMENT. SCIENCEDIRECT.COM

[6]. “REVOLUTIONIZING WITH INNOVATIVE SMART PARKING SYSTEMS” (MDPI SUSTAINABILITY, 2023) – DEFINES REQUIREMENTS AND TECHNICAL CHOICES FOR IOT-PARKING ARCHITECTURE IN SMART CITIES. MDPI.COM