SMART PARKING SYSTEM

A PROJECT REPORT

Submitted by

MADESH A(2303811710421088)

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–621112

NOVEMBER- 2024

K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM–621112

BONAFIDECERTIFICATE

Certified that this project report on **“SMART PARKING SYSTEM ”** is the bonafide work of **MADESH A (2303811710421088)** 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
Mrs.K.VALLI PRIYADHARSHINI, M.E.,(Ph.D.,)
SUPERVISOR
ASSISTANTP ROFESSOR

SIGNATURE

Mrs.K.Valli Priyadharshini, M.E.,(Ph.D.,),

SUPERVISOR

ASSISTANT PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology
(Autonomous)

Samayapuram–621112.

Submitted for the viva-voce examination held on 03.12.2024

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

INTERNAL EXAMINER

CGB1201-JAVA PROGRAMMING
Ms.R.ARUNA PRIYA, M.E.,
EXTERNAL EXAMINER
ASSISTANT PROFESSOR
8104-DSEC, PERAMBALUR.

EXTERNAL EXAMINER

DECLARATION

I declare that the project report on **“HELPDESK MANAGEMENT SYSTEM”** is the result of original work done by us and best of our knowledge, similar workhas 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.**

.

MADESH A

Place: Samayapuram

Date: 03/12/2024

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 adequateduration 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.K.VALLI PRIYADHARSHINI, M.E.,(Ph.D.,),** Department of **COMPUTER SCIENCE AND ENGINEERING,** for his incalculable suggestions, creativity, assistance and patiencewhich 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 change.

ABSTRACT

The Smart Parking System is an advanced Java-based application developed to revolutionize parking management by leveraging technology for real-time monitoring, dynamic pricing, and streamlined reservations. It continuously tracks parking space occupancy and adjusts prices based on demand, ensuring optimal utilization of resources while maximizing revenue. A key feature of the system is its reservation module, which allows users to book parking spaces for specific time slots, confirm their reservations, and update availability instantly. The system is built with robust logic to handle conflicts, prevent overbooking, and provide users with a seamless experience. Additionally, dynamic pricing incentivizes off-peak usage, reduces congestion, and enhances parking efficiency in high-demand urban areas. Scalable and user-friendly, the Smart Parking System is a modern solution tailored to address the increasing challenges of urban mobility and parking shortages, catering to both individual users and parking operators with enhanced convenience, transparency, and profitability.

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 is a Java-based application designed to optimize parking space management through real-time occupancy monitoring and dynamic pricing adjustments based on demand. It features a reservation module enabling users to book parking spaces for specific time slots, with functionalities for confirming reservations and updating space availability. By integrating data-driven algorithms for pricing and efficient booking management, the system ensures optimal utilization of parking resources, enhances user convenience, and maximizes operator revenue.	**PO1 -3** **PO2 -3** **PO3 -3** **PO5 -3** **PO6 -3** **PO8 -3** **PO9 -3** **PO10 -3** **PO11-3** **PO12 -3**	**PSO1 -3** **PSO2 -3** **PSO3 -3**

Note: 1- Low, 2-Medium, 3- High

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1. Objective

The objective of the Smart Parking System is to develop a Java-based application that enhances parking space management through real-time monitoring and dynamic pricing adjustments based on demand. The system aims to optimize parking resource utilization while providing users with a seamless experience by incorporating a reservation module for booking parking spaces for specific time slots. Additionally, the system ensures accurate availability updates, conflict-free reservation confirmations, and efficient handling of occupancy data to cater to the needs of both users and parking operators, ultimately improving convenience, reducing congestion, and maximizing revenue.

2. Overview

The Smart Parking System is a Java-based application designed to enhance parking management by integrating real-time monitoring, dynamic pricing, and a user-friendly reservation system. It tracks parking space occupancy in real time and uses intelligent algorithms to adjust pricing based on demand, optimizing resource utilization and revenue generation. The reservation module allows users to book parking spaces for specific time slots, confirm reservations, and update availability instantly, ensuring a seamless experience. By addressing challenges such as overbooking, inefficient pricing, and space shortages, this scalable solution provides convenience for users and supports operators in managing parking resources effectively, making it an ideal tool for modern urban infrastructure.

3. Java Programming Concepts

- **Encapsulation:** To organize data (parkingSlots, reservations) and operations (handleReservation, refreshParkingSlots). The class extends Frame, demonstrating inheritance for GUI components.
- **GUI Programming with AWT:** Components like Button, Label, List, and TextField create the user interface. Event handling using ActionListener allows the application to respond to user actions like button clicks.
- **Collections Framework:** HashMap is used to store parking slot availability and reservation details. It demonstrates key-value mapping and dynamic updates.
- **Role-Based Access Control:** Conditional logic differentiates between Admin and User roles, restricting or enabling specific functionalities such as adding/removing slots.
- **Iteration:** A for loop initializes parking slots dynamically, and a list is updated iteratively to display their status.
- **String Manipulation:** String splitting (split(“ – “)) is used to extract slot names from the selected list item.
- **Event-Driven Programming:** The application operates based on user-triggered events, such as logging in or reserving a slot.
- **Dynamic GUI Updates:** Components like the parking slot list and admin buttons are updated in real-time to reflect user actions and system changes.

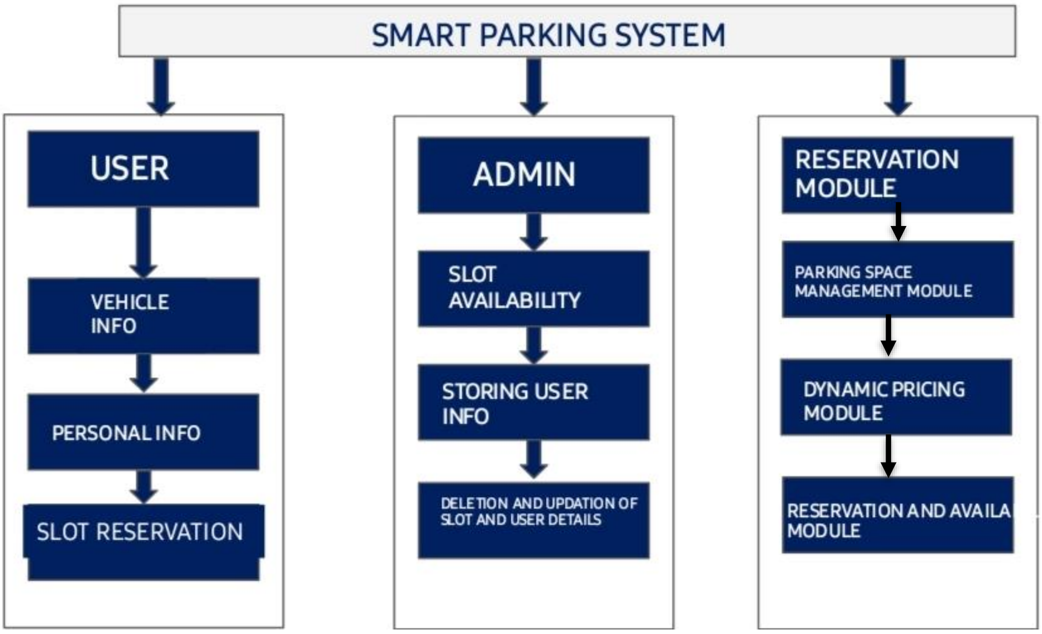
CHAPTER 2

PROJECT METHODOLOGY

1. Proposed Work

The proposed work for enhancing the Smart Parking System includes modernizing the UI with tools like Swing or JavaFX, integrating a database for persistent storage, and implementing a robust authentication system with encrypted passwords and user registration. Features like dynamic pricing, timed reservations, and admin-controlled customizations for pricing and reports can improve functionality. Mobile and web compatibility using frameworks like Spring Boot and ReactJS can broaden accessibility, while integrating IoT sensors for real-time slot detection enhances automation. Additional improvements include payment gateway integration, a notification system for alerts, multilingual support, slot search and filtering, and scalability enhancements to handle larger parking systems efficiently. These upgrades aim to create a feature-rich, user-friendly, and scalable parking management solution.

2. Block Diagram



CHAPTER 3
MODULE DESCRIPTION

1. Authentication Module

The Module manages user login by validating credentials and distinguishing between Admin and User roles. Admins gain access to advanced features such as slot management and system adjustments, while Users are limited to slot reservation functionalities. By implementing role-based access control, the module ensures secure and personalized system interaction, restricting sensitive operations to authorized personnel while providing a user-friendly login experience.

2. Parking Slot Management Module

The Module handles the initialization, addition, removal, and dynamic status updates of parking slots. Admins can add new slots or remove unused ones, while the system tracks the real-time availability of slots as "Available" or "Reserved." It ensures efficient use of parking spaces and adapts to changing requirements, allowing flexible parking management while maintaining accurate slot information.

3. Reservation Module

The Reservation Module allows users to select available slots, specify a time slot and price, and book them. It updates the slot status dynamically and prevents double bookings by maintaining accurate reservations data. This module enhances system efficiency by managing parking space utilization and streamlining the booking process, ensuring that users can reserve slots easily and without conflicts.

4. GUI Module

The Module provides a graphical interface using AWT components like Label, Button, TextField, and List to display parking slots, reservation options, and admin controls. The GUI dynamically updates based on user actions, such as changes in slot status or role-based features. It ensures a seamless and intuitive user experience, enabling both admins and users to interact effectively with the system.

5. Admin Dashboard Module

The admin dashboard offers exclusive functionalities like adding/removing slots, adjusting prices, and managing system configurations. By centralizing administrative operations, it provides a comprehensive control panel for efficiently adapting the parking system to changing needs. This module enhances system oversight and simplifies management tasks for admins, ensuring smooth system operation.

CHAPTER 4

CONCLUSION AND FUTURE SCOPE

1. CONCLUSION

The Smart Parking System provides a comprehensive solution for efficient parking space management, catering to both users and administrators. With essential features like user authentication, dynamic parking slot updates, and seamless reservations, it streamlines the parking process and optimizes the utilization of available spaces. The system's modular architecture allows for easy future enhancements, such as the integration of payment systems, real-time slot occupancy sensors, and database management for persistent data storage. The administrative dashboard empowers admins with control over slot management, pricing adjustments, and system configurations, ensuring smooth operation. The GUI ensures a user-friendly experience, making it simple for users to reserve spots and interact with the system. Overall, this system not only enhances parking efficiency but also improves customer satisfaction by providing a flexible, scalable, and secure parking solution. As future improvements are implemented, the system can be adapted to handle larger parking infrastructures, integrate advanced technologies, and provide richer functionality, ultimately contributing to smarter, more sustainable urban parking management.

2. FUTURE SCOPE

The Smart Parking System has the potential for several future improvements to further enhance its functionality and user experience. Key upgrades include the integration of IoT technology for real-time parking slot monitoring, allowing for automatic updates and better management of available spaces. Dynamic pricing models based on demand, time of day, or special events could optimize revenue and improve space utilization. The addition of diverse payment options, including mobile wallets, UPI, and contactless payments, will streamline the payment process for user.

Furthermore, the development of mobile and web applications would make the system more accessible and convenient for users on the go. Transitioning to a database-driven approach will ensure data persistence, scalability, and efficient management of larger datasets. Features like user profiles, loyalty programs, and personalized recommendations would enhance customer satisfaction, while advanced reporting and analytics will empower administrators with valuable insights for data-driven decision-making. Additionally, multilingual support, AI-powered predictive analytics for demand forecasting, and the introduction of electric vehicle (EV) charging stations would broaden the system’s functionality, making it adaptable to evolving parking needs. With these improvements, the system could evolve into a comprehensive, smart, and sustainable parking solution that addresses the growing demands of urban infrastructure.

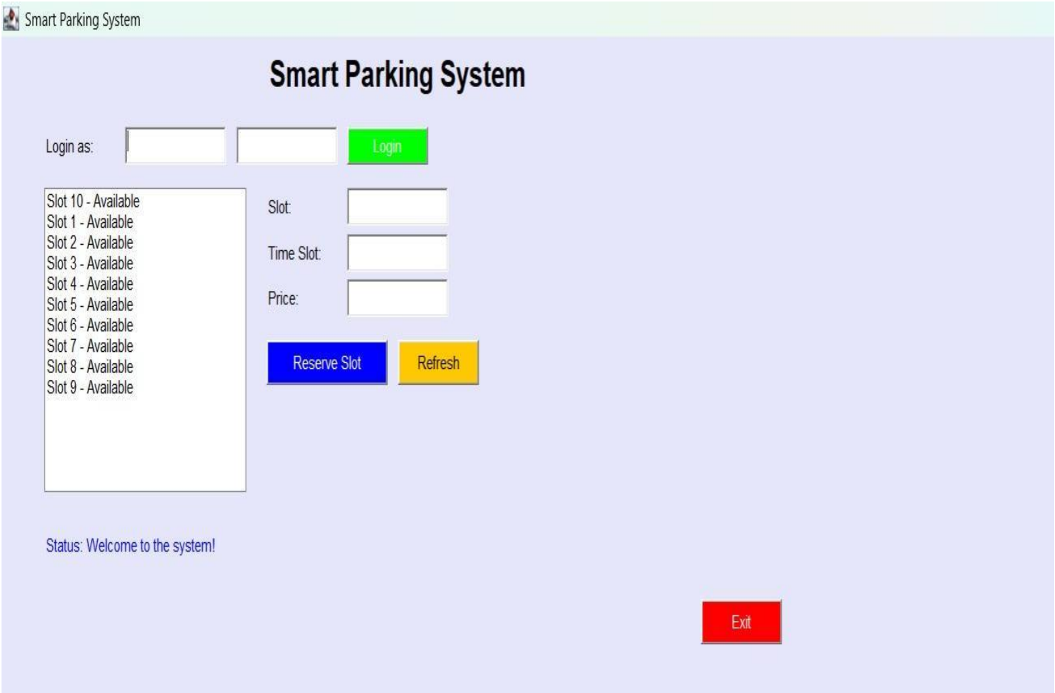
APPENDIX A

```
import java.awt.*;
import java.awt.event.*;
import java.util.HashMap;

public class SmartParkingSystem extends Frame {
    private Label titleLabel, statusLabel, reservationLabel, roleLabel, slotLabel,
timeSlotLabel, priceLabel;
    private Button reserveButton, refreshButton, exitButton, addSlotButton,
removeSlotButton, adjustPriceButton, loginButton;
    private List parkingSlotsList;
    private TextField timeSlotField, priceField, slotNameField, usernameField,
passwordField;
    private HashMap<String, Boolean> parkingSlots; // true if available
    private HashMap<String, String> reservations; // stores reservations
    private boolean isAdmin;

    public SmartParkingSystem() {
        // Initialize data
        parkingSlots = new HashMap<>();
        reservations = new HashMap<>();
        for (int i = 1; i <= 10; i++) {
            parkingSlots.put(“Slot “ + i, true);
        }
        // Set up Frame
        setLayout(null);
        setSize(800, 500);
        setTitle(“Smart Parking System”);
        setBackground(new Color(230, 230, 250));
        // Title Label
        titleLabel = new Label(“Smart Parking System”);
        titleLabel.setFont(new Font(“Arial”, Font.BOLD, 24));
        titleLabel.setBounds(250, 40, 300, 30);
        titleLabel.setAlignment(Label.CENTER);
        add(titleLabel);
```

```
// Role Label
roleLabel = new Label(“Login as:”);
roleLabel.setBounds(50, 90, 80, 25);
add(roleLabel);

usernameField = new TextField();
usernameField.setBounds(130, 90, 100, 25);
add(usernameField);

passwordField = new TextField();
passwordField.setBounds(240, 90, 100, 25);
passwordField.setEchoChar(‘*’);
add(passwordField);

loginButton = new Button(“Login”);
loginButton.setBounds(350, 90, 80, 25);
loginButton.setBackground(Color.GREEN);
loginButton.setForeground(Color.WHITE);
add(loginButton);

// Parking Slots List
parkingSlotsList = new List();
parkingSlotsList.setBounds(50, 130, 200, 200);
refreshParkingSlots();
add(parkingSlotsList);

// Slot Reservation Section
slotLabel = new Label(“Slot:”);
slotLabel.setBounds(270, 130, 80, 25);
add(slotLabel);
timeSlotLabel = new Label(“Time Slot:”);
timeSlotLabel.setBounds(270, 160, 80, 25);
add(timeSlotLabel);

priceLabel = new Label(“Price:”);
priceLabel.setBounds(270, 190, 80, 25);
add(priceLabel);

slotNameField = new TextField();
slotNameField.setBounds(350, 130, 100, 25);
add(slotNameField);
```

```
  timeSlotField = new TextField();
  timeSlotField.setBounds(350, 160, 100, 25);
  add(timeSlotField);

  priceField = new TextField();
  priceField.setBounds(350, 190, 100, 25);
  add(priceField);

  reserveButton = new Button(“Reserve Slot”);
  reserveButton.setBounds(270, 230, 120, 30);
  reserveButton.setBackground(Color.BLUE);
  reserveButton.setForeground(Color.WHITE);
  add(reserveButton);

  refreshButton = new Button(“Refresh”);
  refreshButton.setBounds(400, 230, 80, 30);
  refreshButton.setBackground(Color.ORANGE);
  add(refreshButton);

  exitButton = new Button(“Exit”);
  exitButton.setBounds(700, 400, 80, 30);
  exitButton.setBackground(Color.RED);
  exitButton.setForeground(Color.WHITE);
  add(exitButton);
// Admin Buttons
  addSlotButton = new Button(“Add Slot”);
  addSlotButton.setBounds(500, 130, 120, 30);
  addSlotButton.setBackground(Color.GREEN);
  addSlotButton.setVisible(false);
  add(addSlotButton);

  removeSlotButton = new Button(“Remove Slot”);
  removeSlotButton.setBounds(500, 170, 120, 30);
   removeSlotButton.setBackground(Color.RED);
  removeSlotButton.setVisible(false);
  add(removeSlotButton);

  adjustPriceButton = new Button(“Adjust Price”);
  adjustPriceButton.setBounds(500, 210, 120, 30);
adjustPriceButton.setBackground(Color.MAGENTA);
        adjustPriceButton.setVisible(false);
        add(adjustPriceButton);

        // Status Label
        statusLabel = new Label(“Status: Welcome to the system!”);
        statusLabel.setBounds(50, 350, 400, 30);
        statusLabel.setForeground(Color.BLUE);
        add(statusLabel);

        // Event Listeners
        loginButton.addActionListener(e -> handleLogin());
        reserveButton.addActionListener(e -> handleReservation());
        refreshButton.addActionListener(e -> refreshParkingSlots());
        exitButton.addActionListener(e -> System.exit(0));
        addSlotButton.addActionListener(e -> addParkingSlot());
        removeSlotButton.addActionListener(e -> removeParkingSlot());
        adjustPriceButton.addActionListener(e -> adjustSlotPrice());
    }

    private void handleLogin() {
        String username = usernameField.getText();
        String password = passwordField.getText();
        if (username.equals(“admin”) && password.equals(“admin123”)) {
            isAdmin = true;
            addSlotButton.setVisible(true);
            removeSlotButton.setVisible(true);
            adjustPriceButton.setVisible(true);
            statusLabel.setText(“Status: Logged in as Admin”);
        } else if (username.equals(“user”) && password.equals(“user123”))
{
            isAdmin = false;
            addSlotButton.setVisible(false);
            removeSlotButton.setVisible(false);
            adjustPriceButton.setVisible(false);
            statusLabel.setText(“Status: Logged in as User”);
        } else {
            statusLabel.setText(“Status: Invalid login credentials!”);
        }
    }
    private void handleReservation() {
        String selectedSlot = parkingSlotsList.getSelectedItem();
        String timeSlot = timeSlotField.getText();
        String price = priceField.getText();

        if (selectedSlot == null || timeSlot.isEmpty() || price.isEmpty()) {
            statusLabel.setText(“Status: Please fill in all fields!”);
            return;
        }

        if (!parkingSlots.get(selectedSlot)) {
            statusLabel.setText(“Status: Slot already reserved!”);
            return;
        }

        parkingSlots.put(selectedSlot, false);
        reservations.put(selectedSlot, “Reserved for “ + timeSlot + “ at $” +
price);
        statusLabel.setText(“Status: Slot reserved successfully!”);
        refreshParkingSlots();
    }

    private void refreshParkingSlots() {
        parkingSlotsList.removeAll();
        for (String slot : parkingSlots.keySet()) {
            String status = parkingSlots.get(slot) ? “Available” : “Reserved”;
            parkingSlotsList.add(slot + “ – “ + status);
        }
    }
    private void addParkingSlot() {
        String newSlot = slotNameField.getText();
        if (newSlot.isEmpty() || parkingSlots.containsKey(newSlot)) {
            statusLabel.setText(“Status: Invalid slot name or already exists!”);
            return;
        }

        parkingSlots.put(newSlot, true);
        statusLabel.setText(“Status: Slot added successfully!”);
        refreshParkingSlots();
    }
    private void removeParkingSlot() {
        String selectedSlot = parkingSlotsList.getSelectedItem();
        if (selectedSlot == null) {
            statusLabel.setText(“Status: Please select a slot to remove!”);
            return;
        }

        String slotName = selectedSlot.split(“ – “)[0];
        parkingSlots.remove(slotName);
        reservations.remove(slotName);
        statusLabel.setText(“Status: Slot removed successfully!”);
        refreshParkingSlots();
    }

    private void adjustSlotPrice() {
        statusLabel.setText(“Status: Adjusting price is not implemented!”);
    }

    public static void main(String[] args) {
        SmartParkingSystem app = new SmartParkingSystem();
        app.setVisible(true);
    }
}
```

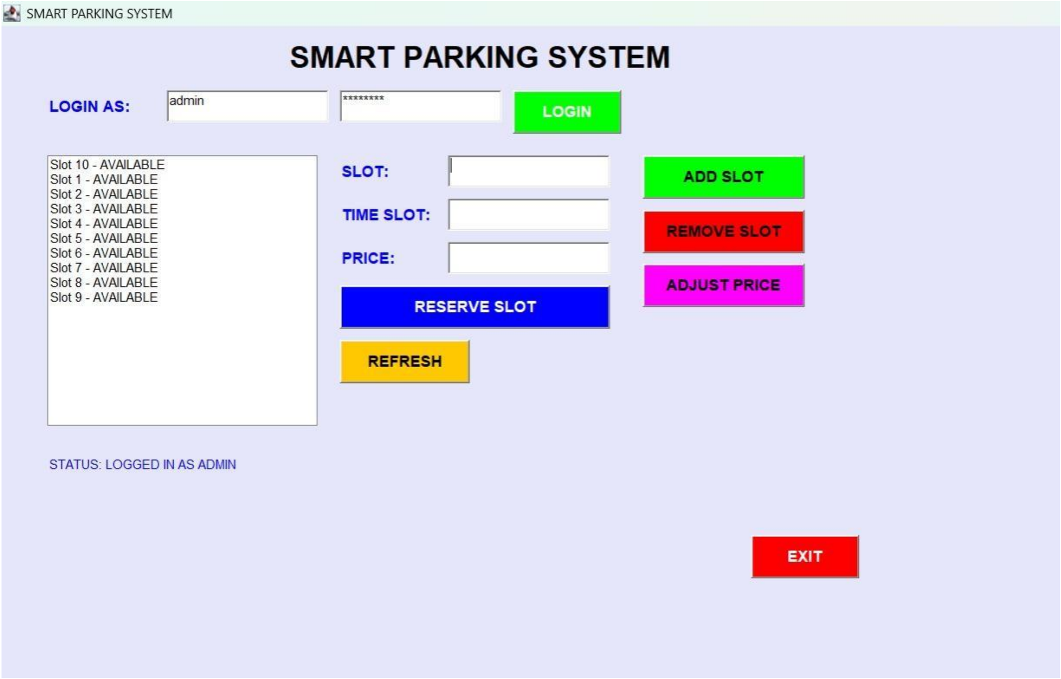
APPENDIX B



Slot 10 - Available
Slot 1 - Available
Slot 2 - Available
Slot 3 - Available
Slot 4 - Available
Slot 5 - Available
Slot 6 - Available
Slot 7 - Available
Slot 8 - Available
Slot 9 - Available

Status: Welcome to the system!

ADMIN LOGIN

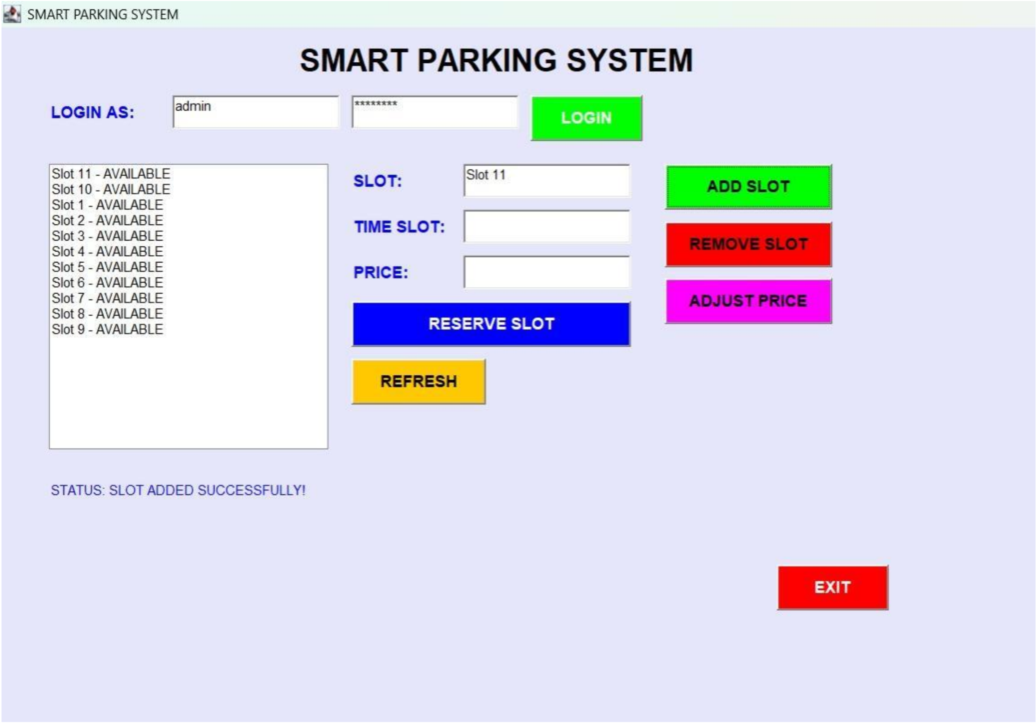


LOGIN AS: admin ********

Slot 10 - AVAILABLE
Slot 1 - AVAILABLE
Slot 2 - AVAILABLE
Slot 3 - AVAILABLE
Slot 4 - AVAILABLE
Slot 5 - AVAILABLE
Slot 6 - AVAILABLE
Slot 7 - AVAILABLE
Slot 8 - AVAILABLE
Slot 9 - AVAILABLE

STATUS: LOGGED IN AS ADMIN

SLOT ADDED



SMART PARKING SYSTEM

SMART PARKING SYSTEM

LOGIN AS: admin ********

Slot 11 - AVAILABLE
Slot 10 - AVAILABLE
Slot 1 - AVAILABLE
Slot 2 - AVAILABLE
Slot 3 - AVAILABLE
Slot 4 - AVAILABLE
Slot 5 - AVAILABLE
Slot 6 - AVAILABLE
Slot 7 - AVAILABLE
Slot 8 - AVAILABLE
Slot 9 - AVAILABLE

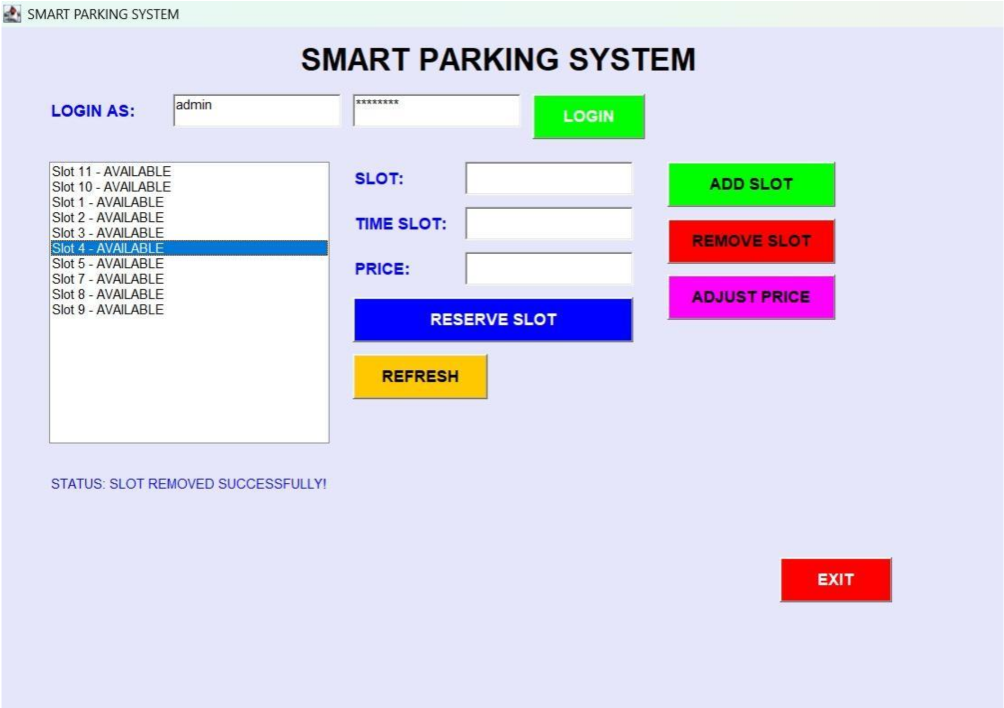
SLOT: Slot 11

TIME SLOT:

PRICE:

STATUS: SLOT ADDED SUCCESSFULLY!

SLOT REMOVED



SMART PARKING SYSTEM

SMART PARKING SYSTEM

LOGIN AS: admin ********

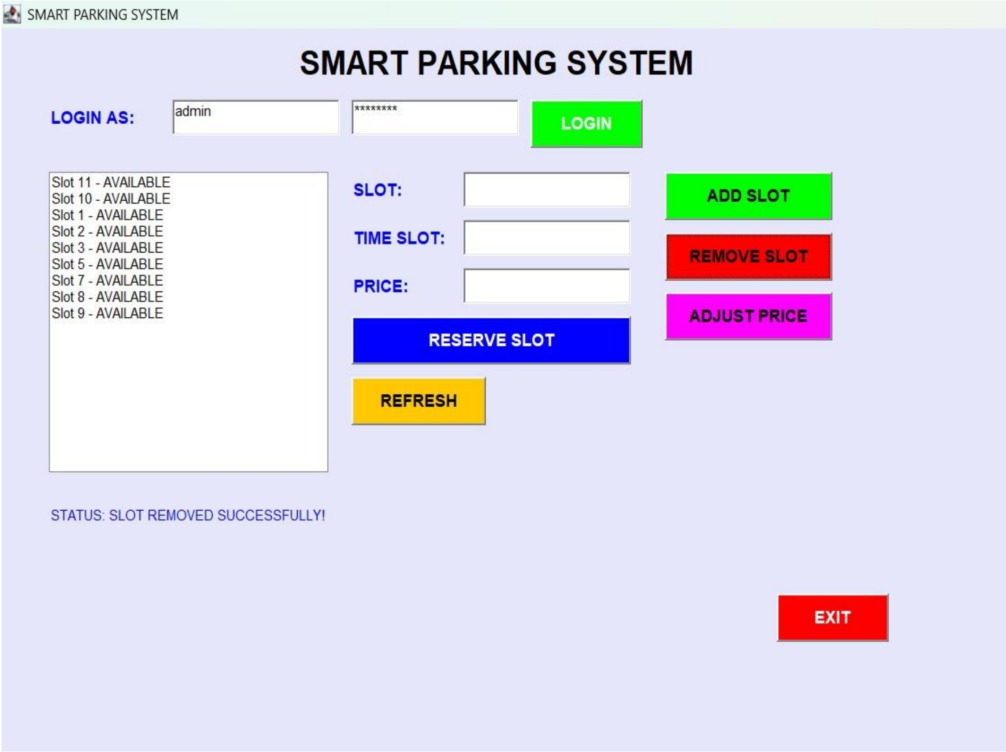
Slot 11 - AVAILABLE
Slot 10 - AVAILABLE
Slot 1 - AVAILABLE
Slot 2 - AVAILABLE
Slot 3 - AVAILABLE
Slot 4 - AVAILABLE
Slot 5 - AVAILABLE
Slot 7 - AVAILABLE
Slot 8 - AVAILABLE
Slot 9 - AVAILABLE

SLOT:

TIME SLOT:

PRICE:

STATUS: SLOT REMOVED SUCCESSFULLY!



SMART PARKING SYSTEM

LOGIN AS: admin ********

Slot 11 - AVAILABLE
Slot 10 - AVAILABLE
Slot 1 - AVAILABLE
Slot 2 - AVAILABLE
Slot 3 - AVAILABLE
Slot 5 - AVAILABLE
Slot 7 - AVAILABLE
Slot 8 - AVAILABLE
Slot 9 - AVAILABLE

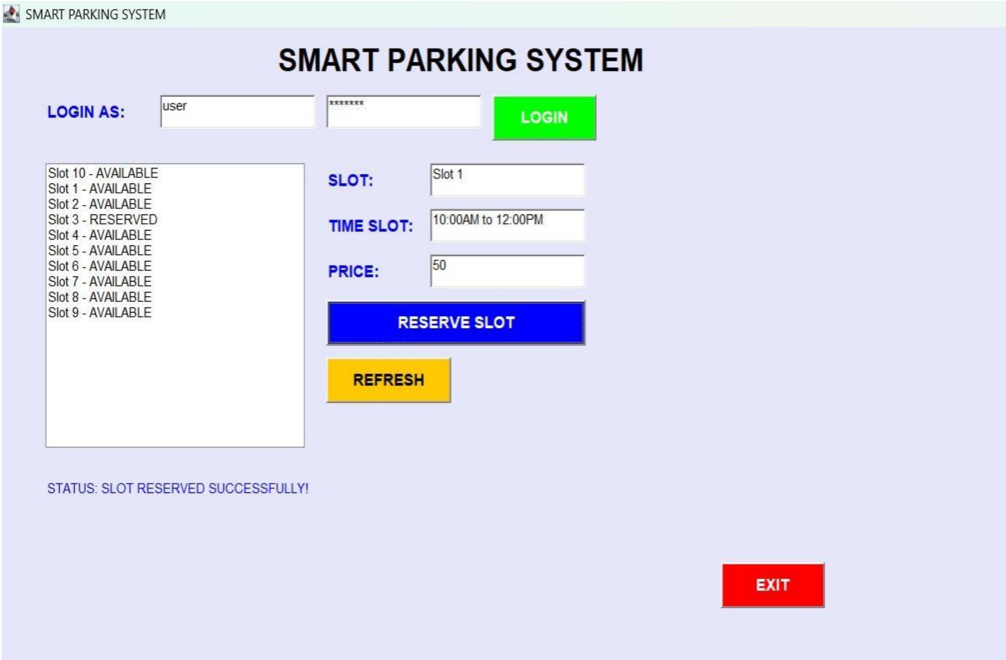
SLOT:

TIME SLOT:

PRICE:

STATUS: SLOT REMOVED SUCCESSFULLY!

RESERVE SLOT



SMART PARKING SYSTEM

LOGIN AS: user *******

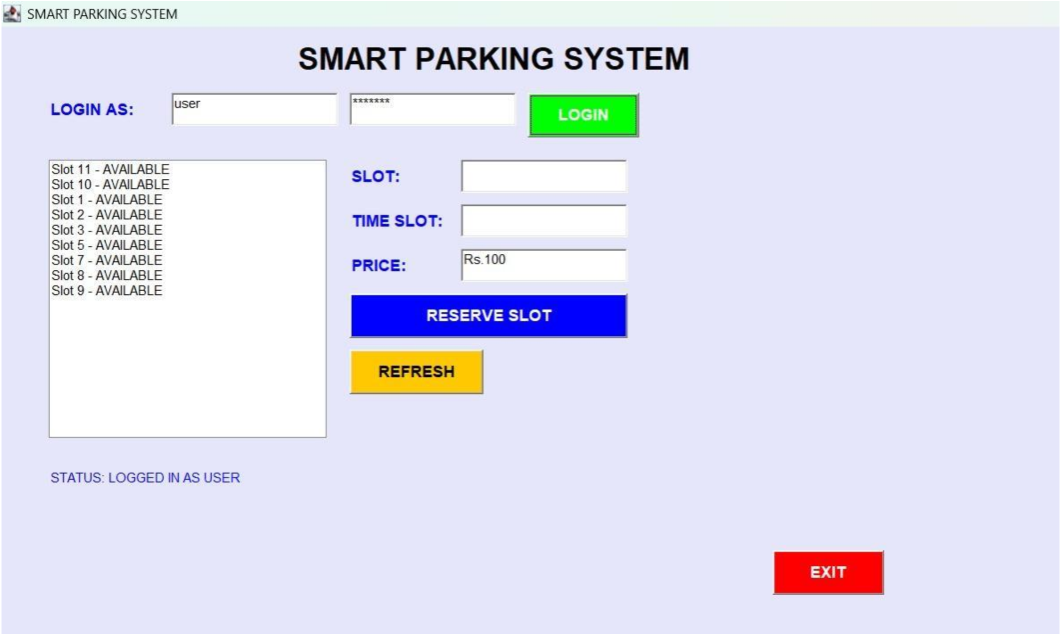
Slot 10 - AVAILABLE
Slot 1 - AVAILABLE
Slot 2 - AVAILABLE
Slot 3 - RESERVED
Slot 4 - AVAILABLE
Slot 5 - AVAILABLE
Slot 6 - AVAILABLE
Slot 7 - AVAILABLE
Slot 8 - AVAILABLE
Slot 9 - AVAILABLE

SLOT: Slot 1

TIME SLOT: 10:00AM to 12:00PM

PRICE: 50

STATUS: SLOT RESERVED SUCCESSFULLY!



SMART PARKING SYSTEM

SMART PARKING SYSTEM

LOGIN AS: user *******

Slot 11 - AVAILABLE
Slot 10 - AVAILABLE
Slot 1 - AVAILABLE
Slot 2 - AVAILABLE
Slot 3 - AVAILABLE
Slot 5 - AVAILABLE
Slot 7 - AVAILABLE
Slot 8 - AVAILABLE
Slot 9 - AVAILABLE

SLOT:

TIME SLOT:

PRICE: Rs.100

STATUS: LOGGED IN AS USER

REFERENCES

WEBSITES

- https://www.geeksforgeeks.org/java-awt-introduction/
- https://www.tutorialspoint.com/awt/index.htm
- https://www.codeproject.com/

BOOK

- "Core Java Volume I – Fundamentals" by Cay S. Horstmann and Gary Comell
- "Java: The Complete Reference" by Herbert Schildt