VEHICLE PARKING MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

AUSTIN POWERS NEWTON [Reg No:RA2211030010057]

YOGESH K R [Reg No: RA2211030010064]

KAVISH DHIMAN [Reg No: RA2211030010067]

Under the Guidance of

DR. DEEPA NATESAN

Assistant Professor, Department of Networking and communications

In partial fulfilment of the requirements for the degree of

BACHELOR OF TECHNOLOGY IN

COMPUTER SCIENCE AND ENGINEERING

with a specialization in CYBER SECURITY

DEPARTMENT OF NETWORKING AND COMMUNICATIONS

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR – 603 203

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that this B.Tech project report titled "**VEHICLE PARKING SYSTEM**" is the bonafide work of, Mr. Austin Powers Newton [Reg. No.RA2211030010057] and Mr. Yogesh K. R.[Reg. No.: RA221030010064], Mr. Kavish Dhiman [Reg. No.: RA221030010067] who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on thebasis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

DR. DEEPA NATESAN
Assistant Professor
Department of Networking and Communications

DR. ANNAPURANI K.
HEAD OF THE DEPARTMENT
Department of Networking and Communications

SIGNATURE OF INTERNAL EXAMINER

SIGNATURE OF EXTERNAL EXAMINER

Department of Networking and Communications

SRM Institute of Science and Technology

Own Work Declaration Form

Degree/ Course : B.Tech in Computer Science and Engineering with specialization in Cyber Security

Student Names : Austin Powers Newton, Yogesh K R, Kavish Dhiman

Registration Number: RA2211030010057, RA2211030010064,RA22110300100067

Title of Work : **VEHICLE PARKING SYTEM**

We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism, as listed in the University Website, Regulations, and the Education Committee guidelines.

We confirm that all the work contained in this assessment is our own except where indicated, and that we have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc.)
- Given the sources of all pictures, data etc. that are not my own

- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:
I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.
If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

ACKNOWLEDGEMENT

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr. T.V.Gopal**, for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman**, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the projectwork.

We are incredibly grateful to our Head of the Department, **Dr. Annapurani Panaiyappan .K,** Professor, Department of Networking and Communications, SRM Institute of Science and Technology,for her suggestions and encouragement at all the stages of the project work.

We register our immeasurable thanks to our Faculty Advisor, **Dr. Rajaram V**, Assistant Professor, Department of Networking and Communications, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to our guide, **Dr. Deepa Natesan**, Assistant Professor, Department of Networking and Communications, SRM Institute of Science and Technology,for providing us with an opportunity to pursue our project under her mentorship. She providedus with the freedom and support to explore the research topics of our interest. Her passion forsolving problems and making a difference in the world has always been inspiring.

We sincerely thank the Networking and Communications Department staff and students, SRM Institute of Science and Technology, for their help during our project. Finally, we wouldlike to thank parents, family members, and friends for their unconditional love, constant support, and encouragement.

Austin Powers Newton [RA2211030010057]

Yogesh K R [RA2211030010064]

Kavish Dhiman [RA2211030010067]

TABLE OF CONTENTS

1.ABSTRACT

This code implements a Parking Management System, featuring a graphical user interface with functionalities for parking, exiting, and information display. The system effectively utilizes time by incorporating a timer mechanism for real-time updates and providing a user-friendly interface for seamless interactions. However, the code requires further development in accurately calculating the hours a vehicle has been parked, a critical factor for precise fee calculations. In terms of space utilization, the system employs a dictionary structure to represent parking slots, allowing for efficient tracking of slot occupancy. This code serves as a foundational prototype, demonstrating potential for optimization in both time and space utilization. Further development and practical testing are recommended to refine algorithms and incorporate additional features, advancing the Parking Management System's overall efficiency and usability.

2. INTRODUCTION

The code presented herein represents the foundation of a Parking Management System, a critical component in addressing the complexities of urban parking. This system is designed to streamline parking operations through a graphical user interface, offering functionalities for parking, vehicle exit, information display, and payment processing. The primary focus of this report is to evaluate the system's effectiveness in terms of time and space utilization. Efficient time management is crucial for providing real-time updates and ensuring a seamless user experience. Simultaneously, the utilization of parking space is a key consideration, necessitating an examination of algorithms that allocate and track parking slots. As urban areas grapple with the ever-increasing demand for parking solutions, the assessment of this code aims to shed light on its capacity to optimize both temporal and spatial resources within the context of a Parking Management System.

3. FEATURES OF THE VEHICLE PARKING SYSTEM

The Parking Management System features several key functionalities aimed at optimizing the allocation of parking slots, time management, and overall space utilization.

1. Dynamic Slot Allocation:

The system dynamically assigns parking slots to incoming vehicles based on availability, reducing the time vehicles spend searching for a spot upon entry.

It utilizes an iterative process to identify the first available slot, ensuring an efficient and timely allocation process for incoming vehicles.

Dynamic slot allocation optimizes the use of parking space by minimizing empty slots and preventing congestion in specific areas of the parking lot.

2. Real-Time Updates:

The system incorporates a timer mechanism to provide users with continuous real-time updates on the current system time, enhancing the overall user experience.

Users benefit from accurate and timely information, enabling them to plan their parking transactions and manage their time effectively within the parking facility.

3. Payment Processing:

The inclusion of a payment functionality streamlines transactions, allowing users to pay for their parking within the system.

Efficient payment processing not only contributes to a seamless user experience but also enhances overall time utilization for both users and parking attendants.

4. Structured Parking Slots:

The system organizes parking slots in a structured manner, utilizing a dictionary that represents each slot.

This structured approach facilitates quick and efficient tracking of slot occupancy, aiding in effective space utilization and ensuring a systematic allocation of parking resources.

5. Information Display:

The system provides a feature to display information about the current status of parking slots, indicating which slots are occupied and which are available.

Users can make informed decisions about where to park based on real-time information, contributing to optimized space utilization within the parking environment.

4. CODE

PYTHON:

```
import tkinter as tk
from tkinter import messagebox, simpledialog
import random
from datetime import datetime


class ParkingManagementSystem(tk.Tk):
    def __init__(self):
        super().__init__()
        self.title("Parking Management System")
        self.geometry("400x300")
        self.parkingSlots = {i: None for i in range(1, 201)}
        self.timeLabel = tk.Label(self)
        self.statusLabel = tk.Label(self)
        self.parkButton = tk.Button(self, text="Park Car", command=self.park_car)
        self.exitButton = tk.Button(self, text="Exit", command=self.exit_car)
        self.infoButton = tk.Button(self, text="Information",
command=self.show_info)
        self.payButton = tk.Button(self, text="Pay", command=self.pay)
        self.vehicleNumberEntry = tk.Entry(self)
        self.vehicleNumberEntry.pack()
        self.parkButton.pack()
        self.exitButton.pack()
        self.infoButton.pack()
```

```
        self.payButton.pack()
        self.timeLabel.pack()
        self.statusLabel.pack()
        self.update_time()
        self.timer = self.after(1000, self.update_time)

    def park_car(self):
        vehicleNumber = self.vehicleNumberEntry.get()
        slotNumber = self.find_available_slot()
        if slotNumber != -1:
            self.parkingSlots[slotNumber] = vehicleNumber
            messagebox.showinfo("", f"Vehicle parked in slot {slotNumber}")
        else:
            messagebox.showinfo("", "No available slots")
        self.vehicleNumberEntry.delete(0, "end")

    def exit_car(self):
        vehicleNumber = simpledialog.askstring("Enter Vehicle Number", "Enter
Vehicle Number:")
        slotNumber = self.find_slot_by_vehicle_number(vehicleNumber)
        if slotNumber != -1:
            self.parkingSlots[slotNumber] = None
            hoursParked = self.calculate_hours_parked(slotNumber)
            amountToPay = self.calculate_amount_to_pay(hoursParked)
            messagebox.showinfo("", f"Vehicle {vehicleNumber} exited from slot
{slotNumber}\nAmount to pay: {amountToPay}")
```

```
        else:
            messagebox.showinfo("", "Vehicle not found")

    def show_info(self):
        info = ""
        for slotNumber, vehicleNumber in self.parkingSlots.items():
            if vehicleNumber is not None:
                info += f"Slot {slotNumber}: {vehicleNumber}\n"
            else:
                info += f"Slot {slotNumber}: Empty\n"
        messagebox.showinfo("", info)

    def pay(self):
        counterNumber = random.randint(1, 3)
        messagebox.showinfo("", f"Please go to counter {counterNumber} for
payment")

    def find_available_slot(self):
        for slotNumber, vehicleNumber in self.parkingSlots.items():
            if vehicleNumber is None:
                return slotNumber
        return -1

    def find_slot_by_vehicle_number(self, vehicleNumber):
        for slotNumber, parkedVehicleNumber in self.parkingSlots.items():
            if vehicleNumber == parkedVehicleNumber:
```

```
            return slotNumber
        return -1

    def calculate_hours_parked(self, slotNumber):
        return 0  # Implement your logic for calculating hours parked

    def calculate_amount_to_pay(self, hoursParked):
        amount = 30
        if hoursParked > 1:
            amount += (hoursParked - 1) * 10
        return amount

    def update_time(self):
        current_time = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        self.timeLabel.config(text=f"Current Time: {current_time}")
        self.timer = self.after(1000, self.update_time)

if __name__ == "__main__":
    parking_system = ParkingManagementSystem()
    parking_system.mainloop()
```

JAVA:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.HashMap;
import java.util.Map;
import java.util.Random;

public class ParkingManagementSystem extends JFrame {
    private Map<Integer, String> parkingSlots;
    private JLabel timeLabel;
    private JLabel statusLabel;
    private JButton parkButton;
    private JButton exitButton;
    private JButton infoButton;
    private JButton payButton;

    public ParkingManagementSystem() {
        setTitle("Parking Management System");
        setSize(400, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        setLayout(new FlowLayout());

        parkingSlots = new HashMap<>();
        for (int i = 1; i <= 200; i++) {
            parkingSlots.put(i, null);
        }

        timeLabel = new JLabel();
        statusLabel = new JLabel();

        parkButton = new JButton("Park Car");
        exitButton = new JButton("Exit");
        infoButton = new JButton("Information");
        payButton = new JButton("Pay");

        parkButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String vehicleNumber = JOptionPane.showInputDialog("Enter Vehicle
Number:");
                int slotNumber = findAvailableSlot();
                if (slotNumber != -1) {
                    parkingSlots.put(slotNumber, vehicleNumber);
                    JOptionPane.showMessageDialog(null, "Vehicle parked in slot " +
slotNumber);
                } else {
```

```
                JOptionPane.showMessageDialog(null, "No available slots");
            }
        }
    });

    exitButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            String vehicleNumber = JOptionPane.showInputDialog("Enter Vehicle Number:");
            int slotNumber = findSlotByVehicleNumber(vehicleNumber);
            if (slotNumber != -1) {
                parkingSlots.put(slotNumber, null);
                int hoursParked = calculateHoursParked(slotNumber);
                int amountToPay = calculateAmountToPay(hoursParked);
                JOptionPane.showMessageDialog(null, "Vehicle " + vehicleNumber + " exited from slot " + slotNumber + "\nAmount to pay: " + amountToPay);
            } else {
                JOptionPane.showMessageDialog(null, "Vehicle not found");
            }
        }
    });

    infoButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
```

```
            StringBuilder sb = new StringBuilder();
            for (int slotNumber : parkingSlots.keySet()) {
                String vehicleNumber = parkingSlots.get(slotNumber);
                if (vehicleNumber != null) {
                    sb.append("Slot ").append(slotNumber).append(": 
").append(vehicleNumber).append("\n");
                } else {
                    sb.append("Slot ").append(slotNumber).append(": Empty\n");
                }
            }
            JOptionPane.showMessageDialog(null, sb.toString());
        }
    });

    payButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            Random random = new Random();
            int counterNumber = random.nextInt(3) + 1;
            JOptionPane.showMessageDialog(null, "Please go to counter " + 
counterNumber + " for payment");
        }
    });

    add(timeLabel);
    add(statusLabel);
```

```
    add(parkButton);
    add(exitButton);
    add(infoButton);
    add(payButton);

    Timer timer = new Timer(1000, new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            updateTime();
        }
    });
    timer.start();
}

private int findAvailableSlot() {
    for (int slotNumber : parkingSlots.keySet()) {
        if (parkingSlots.get(slotNumber) == null) {
            return slotNumber;
        }
    }
    return -1;
}

private int findSlotByVehicleNumber(String vehicleNumber) {
    for (int slotNumber : parkingSlots.keySet()) {
```

```
            if (vehicleNumber.equals(parkingSlots.get(slotNumber))) {
                return slotNumber;
            }
        }
        return -1;
    }

    private int calculateHoursParked(int slotNumber) {
        // Calculate hours parked based on current time and time parked
        return 0;
    }

    private int calculateAmountToPay(int hoursParked) {
        int amount = 30; // Rate for the first hour
        if (hoursParked > 1) {
            amount += (hoursParked - 1) * 10; // Additional rate for each additional
hour
        }
        return amount;
    }

    private void updateTime() {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss");
        String currentTime = sdf.format(new Date());
        timeLabel.setText("Current Time: " + currentTime);
```

```
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                new ParkingManagementSystem().setVisible(true);
            }
        });
    }
}
```

5. OUTPUT

Figure 1: Parking of the Car: The following window, displays options on parking the vehicle with its number on the ascending order of slots.

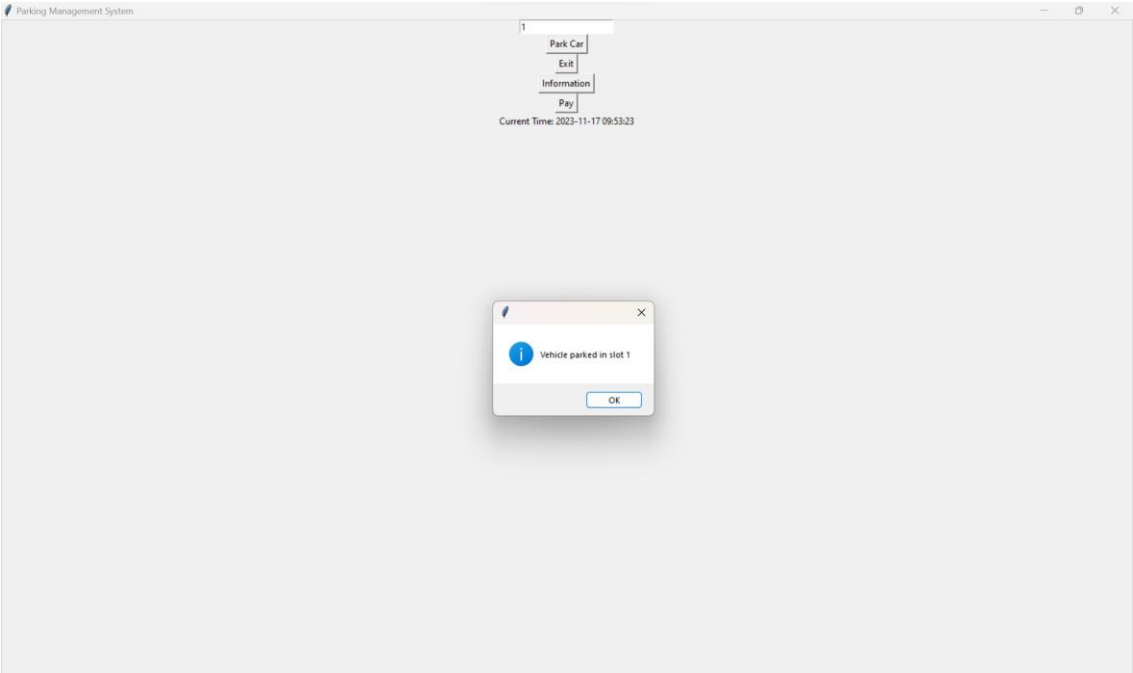


Figure 2: Parking the Vehicle: The Second Vehicle is parked in slot 2.

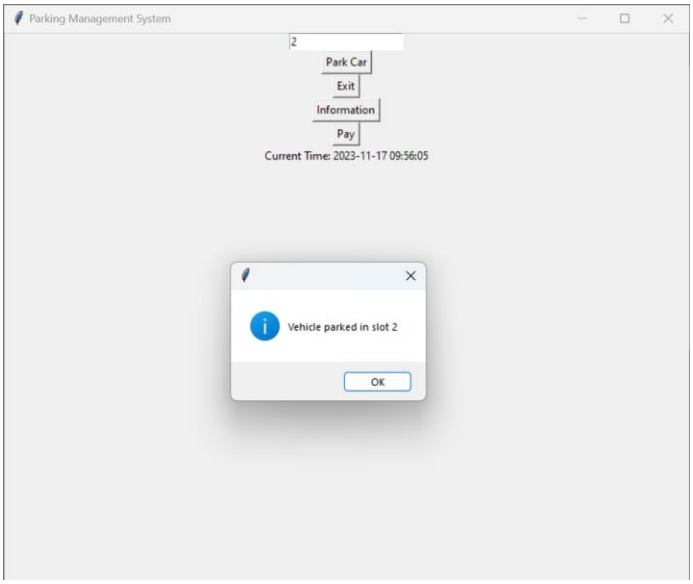


Figure 3: Information Table: A total of 200 slots are created and among them the first two vehicles are parked in slots 1 and 2.

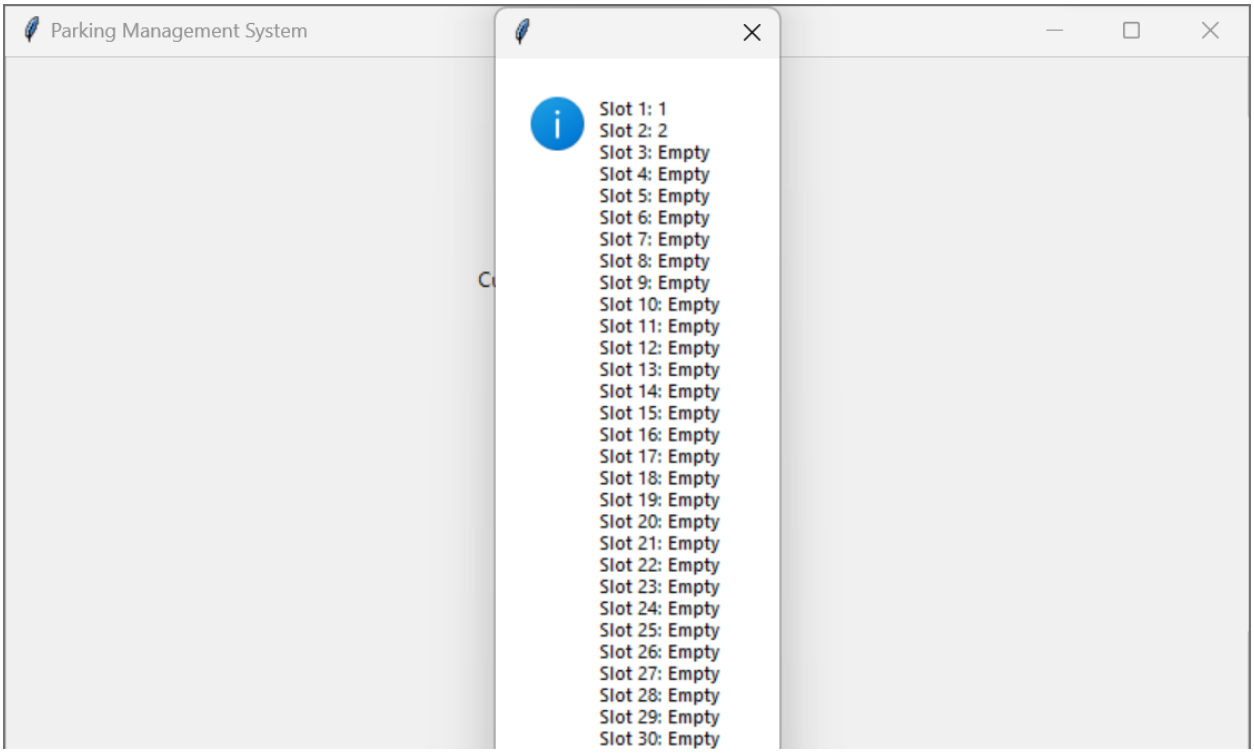


Figure 4: Exit the Vehicle: On attempting to exit the vehicle, the vehicle to be exited is asked with respect to the vehicle number.

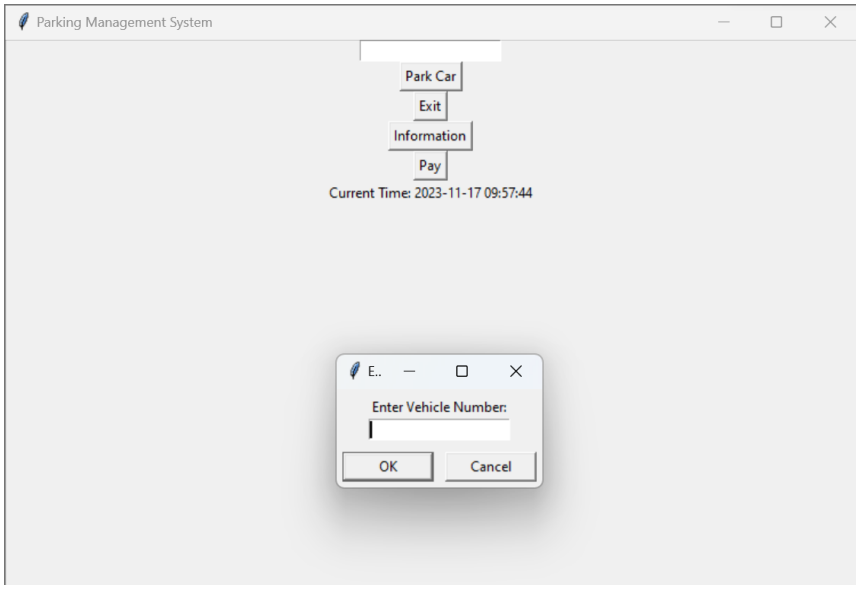


Figure 5: Exited Information: The vehicle exited is informed and also its payment with respect to the time duration of parked hours is provided

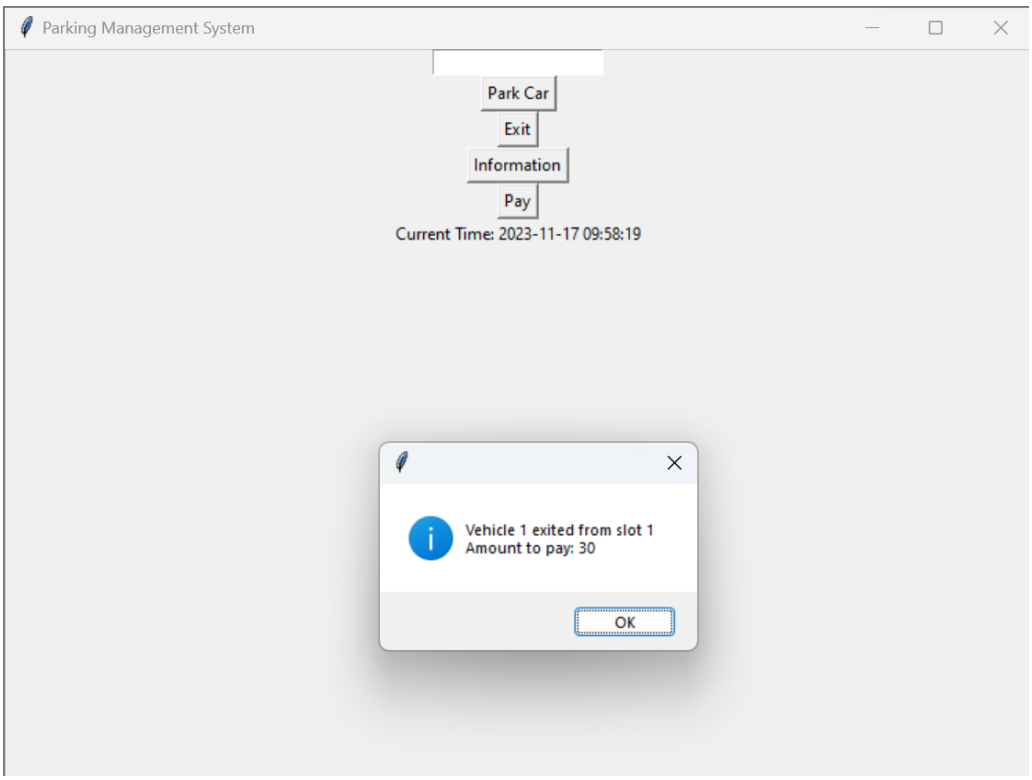
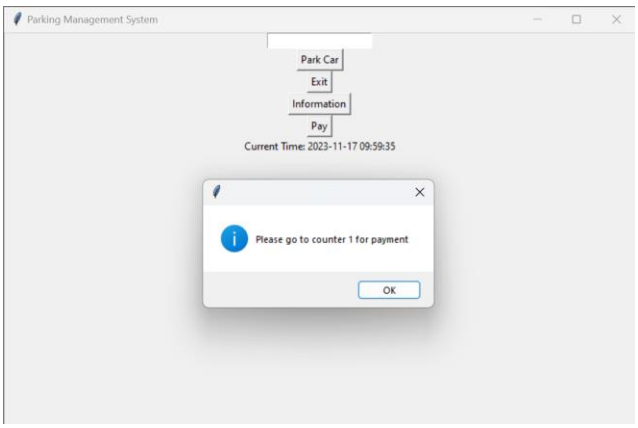


Figure 6: Payment Counter Allocation: To pay the fees of the parked hours, the counter is allotted dynamically to the vehicle owner so that, they can go directly and pay it without any confusion or time complication.



6. FUTURE SCOPE

The Parking Management System presented in the code holds significant potential for real-life applications with several future scopes that can contribute to improved parking management and overall urban mobility. Here are key aspects of its future scope:

1. **Smart City Integration:**

    - Integration with smart city initiatives: The system can be integrated into broader smart city frameworks, allowing for seamless coordination with traffic management systems, public transportation, and real-time data analytics. This integration can lead to more comprehensive urban planning and enhanced traffic flow.

2. **IoT Integration for Sensor-Based Parking:**

    - Sensor-based parking: Future iterations of the system can incorporate Internet of Things (IoT) devices and sensors to provide real-time data on parking space availability. This enables drivers to access live information about available parking spots, reducing congestion and enhancing user convenience.

3. **Predictive Analytics for Parking Demand:**

    - Predictive analytics: Implementing predictive analytics algorithms can help forecast parking demand based on historical data, events, or even weather conditions. This proactive approach allows for better resource allocation, ensuring sufficient parking availability during peak times.

4. **Mobile Application for User Convenience:**

    - Mobile applications: Developing a mobile application interface for the Parking Management System can offer users greater convenience. Users can reserve parking slots in advance, receive navigation assistance, and make payments seamlessly through their smartphones, contributing to an enhanced user experience.

5. **Integration with Electric Vehicle Charging Stations:**

    - Electric vehicle infrastructure: As the demand for electric vehicles (EVs) rises, the system can evolve to integrate with EV charging stations. This expansion supports the growing need for electric vehicle infrastructure and encourages sustainable transportation practices.

6. **Enhanced Security Features:**

    - Security enhancements: Future developments could incorporate advanced security features, such as license plate recognition and CCTV surveillance, to ensure a secure parking environment. This not only enhances user safety but also contributes to the overall security of the parking facility.

7. **User Behavior Analysis:**

    - User behavior analysis: Analyzing user patterns and preferences within the parking system can provide valuable insights for urban planners. Understanding user behavior can inform the development of policies and infrastructure improvements to better meet the evolving needs of the community.

7. CONCLUSION

In conclusion, the presented Parking Management System project lays the groundwork for an efficient and user-friendly solution to urban parking challenges. The system's current features, including dynamic slot allocation, real-time updates, payment processing, structured parking slots, and information display, demonstrate its potential to enhance both time and space utilization within a parking environment. While the system serves as a functional prototype, its future scope holds promise for integration with smart city initiatives, sensor-based parking, predictive analytics, mobile applications, sustainable infrastructure, and advanced security features.

The success of the Parking Management System lies in its adaptability to real-life urban scenarios, contributing to optimized traffic flow, reduced congestion, and improved user experiences. As urban areas continue to grow and evolve, the scalability and versatility of the system position it as a valuable tool for urban planners and administrators. By embracing future developments and addressing emerging challenges, the Parking Management System has the potential to revolutionize the way cities approach parking management, offering a glimpse into a more efficient, connected, and sustainable urban mobility landscape.

8. REFERENCES

- 1] Rad F. et al, “A Smart Hybrid System for Parking Space Reservation in VANET”, Journal of Advances in Computer Engineering and Technology, 2017.
- [2] Correa A. et al, “Autonomous Car Parking System through a co operative vehicular positioning network”, Sensors, vol. 17, issue 4, 2017.
- [3] Sun G., Yu M., Liao D. and Chang V., “Analytical Exploration of Energy Savings for Parked Vehicles to Enhance VANET Connectivity”, IEEE Transactions on Intelligent Transportation Systems, vol. 99, pp. 1-13, 2018.
- [4] Aliedani A. and Loke S. W., “Vehicular Co-operation to Overcome the Car Park Challenge”, Proceedings of 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, ACM, pp. 539-540, 2017.
- [5] Jaya J. et. al. “Identifying the Available Parking Area by the
- Assistance of Parked Vehicle”, International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol 2, 2017.
- [6] Burak Kizilkaya et al, “An Intelligent Car Park Management System: Hierarchical Placement Algorithm based on Nearest Location”, 32nd

International Conference on Advanced Information Networking and Applications Workshops, 2018.

- [7] Fatih Can Akinci and Murat Karakaya, “Parking Space Occupancy Detection using Deep Learning Methods”, IEEE, 2018.
- [8] Jan Silar, Jiri Ruzicka et al, “Smart Parking in the Smart City Application”, Smart Cities Symposium Prague, 2018.
- [9] Jasti Satyakrishna and Raj Kumar Sagar, “Analysis of Smart City Transportation using IOT” , Proceedings of the Second International Conference on Inventive Systems and control (ICISC), pp. 268-273, 2018.
- [10] Kin Sum Liu, Jie Gao et al, “On-Street Parking Guidance with Real Time Sensing Data for Smart Cities” ,IEEE, 2018.
- [11] Shladover S. E., “Connected and Automated Vehicle Systems: Introduction and Overview”, Journal of Intelligent Transportation Systems, vol. 22, issue 3, pp.190-200, 2018.
- [12] Hua L. C. et al, “Social Networking based co-operation
- Mechanisms in Vehicular Ad-hoc Network—A Survey”, Vehicular Communications, 2017.
- [13] Aliedani A. et al, “Cooperative Car Parking using vehicle-to-vehicle communication: An Agent based analysis”, Computers, Environment and Urban Systems, 2018.

- [14] Li Chunhai, Simin Wang, Xumin Huang, Xiaohuan Li, Rong Yu, and Feng Zhao, “Parked Vehicular Computing for Energy-Efficient Internet of Vehicles: A Contract Theoretic Approach”, IEEE Internet of Things Journal, 2018.
- [15] Tong, Richard Chia Tsing, Robert Victor Welland, John Hayes Ludwig, John Palmer Cordell and Samuel James McKelvie, “Method and system for providing interactive parking management via Artificial Intelligence Analytic services using Cloud Network”, U.S. Patent Application 15/853,593, filed July 19, 2018.
- [16] Tong, Richard Chia Tsing, Robert Victor Welland, John Hayes Ludwig, John Palmer Cordell, and Samuel James McKelvie, “Method and System for Providing Artificial Intelligence Analytic (AIA) Services Using Operator Fingerprints and Cloud Data”, U.S. Patent Application 15/852,346, filed June 28, 2018.
- [17] Mahmud S. A., et al “A Survey of Intelligent Car Parking System”, Journal of Applied Research and Technology, Vol.11, pp. 714-726, 2016.
- [18] Demirli, Kudret et al, “Autonomous Parallel Parking of a car-like mobile robot by a neuro-fuzzy sensor-based controller”, Fuzzy Sets and Systems Vol.160, pp. 2876-2891, 2009.

- [19] Hongwei Wang and Wenbo He, “A Reservation-based Smart Parking System”, IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Shanghai, pp. 690-695, 2011.
- [20] Y. Geng et al, “New Smart Parking System based on Resource Allocation and Reservations”, IEEE Transactions on Intelligent Transportation Systems, vol. 14, pp. 1129-1139, 2013