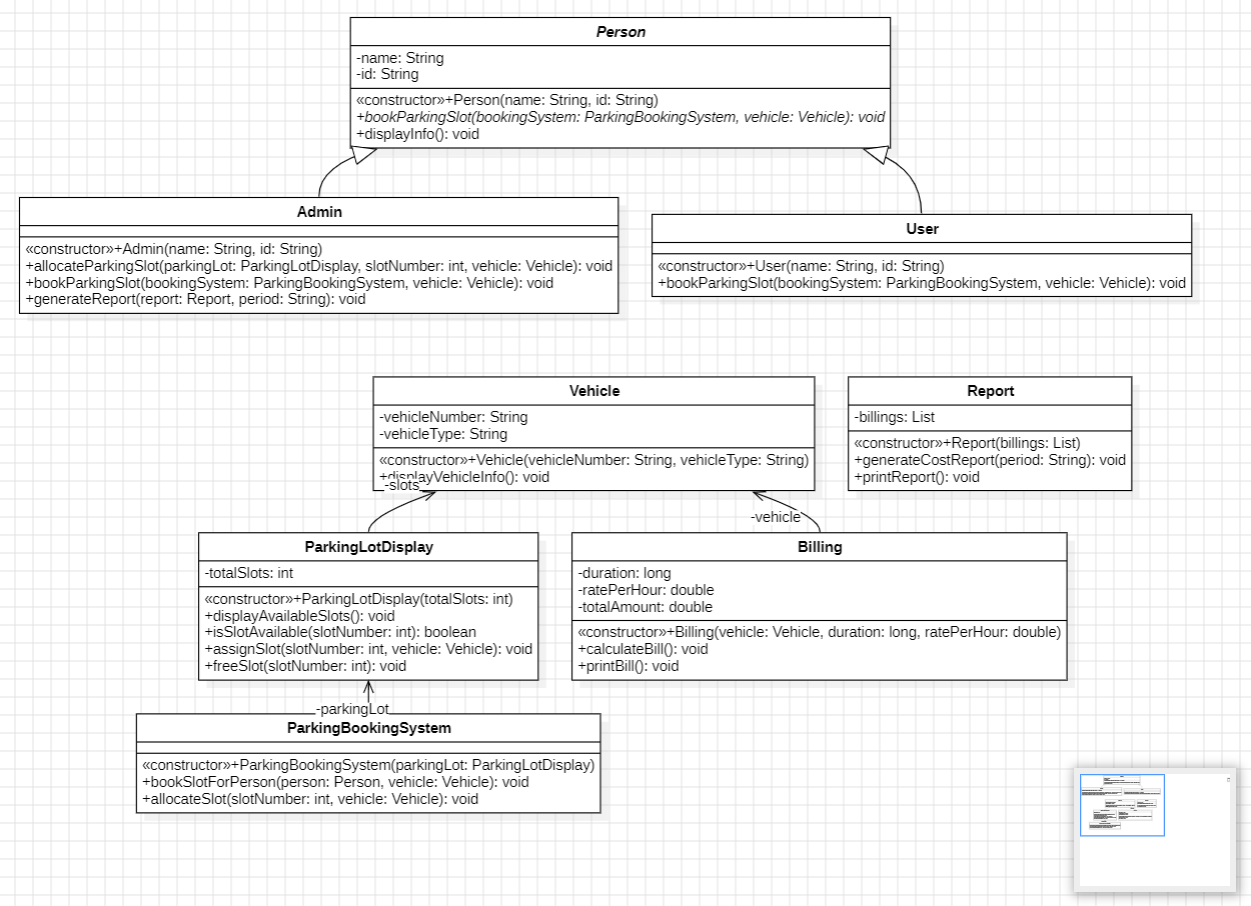
**12.09.2024**

**ASSIGNMENT-4**

**1.Appropriate class Diagram(modules wise if applicable) for the chosen problem.** 

**2. code generated from the class diagram for a module/s and its implementation with sample input output (for your code)**

**CODE:**

*//code generated from class diagram*

public abstract class Person {

    private String name;

    private String id;

    public Person(String name, String id) {

        this.name = name;

        this.id = id;

    }

    public abstract void bookParkingSlot(ParkingBookingSystem bookingSystem, Vehicle vehicle);

    public void displayInfo();

    // Getters and Setters

}

public class Admin extends Person {

    public Admin(String name, String id) {

        super(name, id);

    }

    public void allocateParkingSlot(ParkingLotDisplay parkingLot, int slotNumber, Vehicle vehicle);

    @Override

    public void bookParkingSlot(ParkingBookingSystem bookingSystem, Vehicle vehicle);

    public void generateReport(Report report, String period);

}

public class User extends Person {

    public User(String name, String id) {

        super(name, id);

    }

    @Override

    public void bookParkingSlot(ParkingBookingSystem bookingSystem, Vehicle vehicle);

}

public class ParkingLotDisplay {

    private int totalSlots;

    private Vehicle[] slots;

    public ParkingLotDisplay(int totalSlots) {

        this.totalSlots = totalSlots;

    }

    public void displayAvailableSlots();

    public boolean isSlotAvailable(int slotNumber);

    public void assignSlot(int slotNumber, Vehicle vehicle);

    public void freeSlot(int slotNumber);

}

public class ParkingBookingSystem {

    private ParkingLotDisplay parkingLot;

    public ParkingBookingSystem(ParkingLotDisplay parkingLot) {

        this.parkingLot = parkingLot;

    }

    public void bookSlotForPerson(Person person, Vehicle vehicle);

    public void allocateSlot(int slotNumber, Vehicle vehicle);

}

public class Vehicle {

    private String vehicleNumber;

    private String vehicleType;

    public Vehicle(String vehicleNumber, String vehicleType) {

        this.vehicleNumber = vehicleNumber;

        this.vehicleType = vehicleType;

    }

    public void displayVehicleInfo();

    // Getters and Setters

}

public class Billing {

    private Vehicle vehicle;

    private long duration;

    private double ratePerHour;

    private double totalAmount;

    public Billing(Vehicle vehicle, long duration, double ratePerHour) {

        this.vehicle = vehicle;

        this.duration = duration;

        this.ratePerHour = ratePerHour;

    }

    public void calculateBill();

    public void printBill();

    // Getters and Setters

}

public class Report {

    private List<Billing> billings;

    public Report(List<Billing> billings) {

        this.billings = billings;

    }

    public void generateCostReport(String period);

    public void printReport();

}

**WORKING MODEL:**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class ParkingManagementGUI extends JFrame {

    private JPanel slotPanel;

    private int totalSlots;

    public ParkingManagementGUI() {

        setTitle("Admin Parking Management");

        setSize(500, 500);

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        String input = JOptionPane.showInputDialog(this, "Enter the number of parking slots:");

        try {

            totalSlots = Integer.parseInt(input);

        } catch (NumberFormatException e) {

            JOptionPane.showMessageDialog(this, "Invalid input. Setting default to 10 slots.");

            totalSlots = 10;

        }

        slotPanel = new JPanel();

        slotPanel.setLayout(new GridLayout(0, 4));

        add(slotPanel, BorderLayout.CENTER);

        createParkingSlots(totalSlots);

        setVisible(true);

    }

    private void createParkingSlots(int totalSlots) {

        for (int i = 1; i <= totalSlots; i++) {

            JButton slotButton = new JButton("Slot " + i);

            slotButton.setBackground(Color.GREEN);

            slotButton.setActionCommand(String.valueOf(i));

            slotButton.addActionListener(new SlotButtonListener());

            slotPanel.add(slotButton);

        }

    }

    private class SlotButtonListener implements ActionListener {

        @Override

        public void actionPerformed(ActionEvent e) {

            String slotId = e.getActionCommand();

            JOptionPane.showMessageDialog(ParkingManagementGUI.this, "You clicked Slot " + slotId);

        }

    }

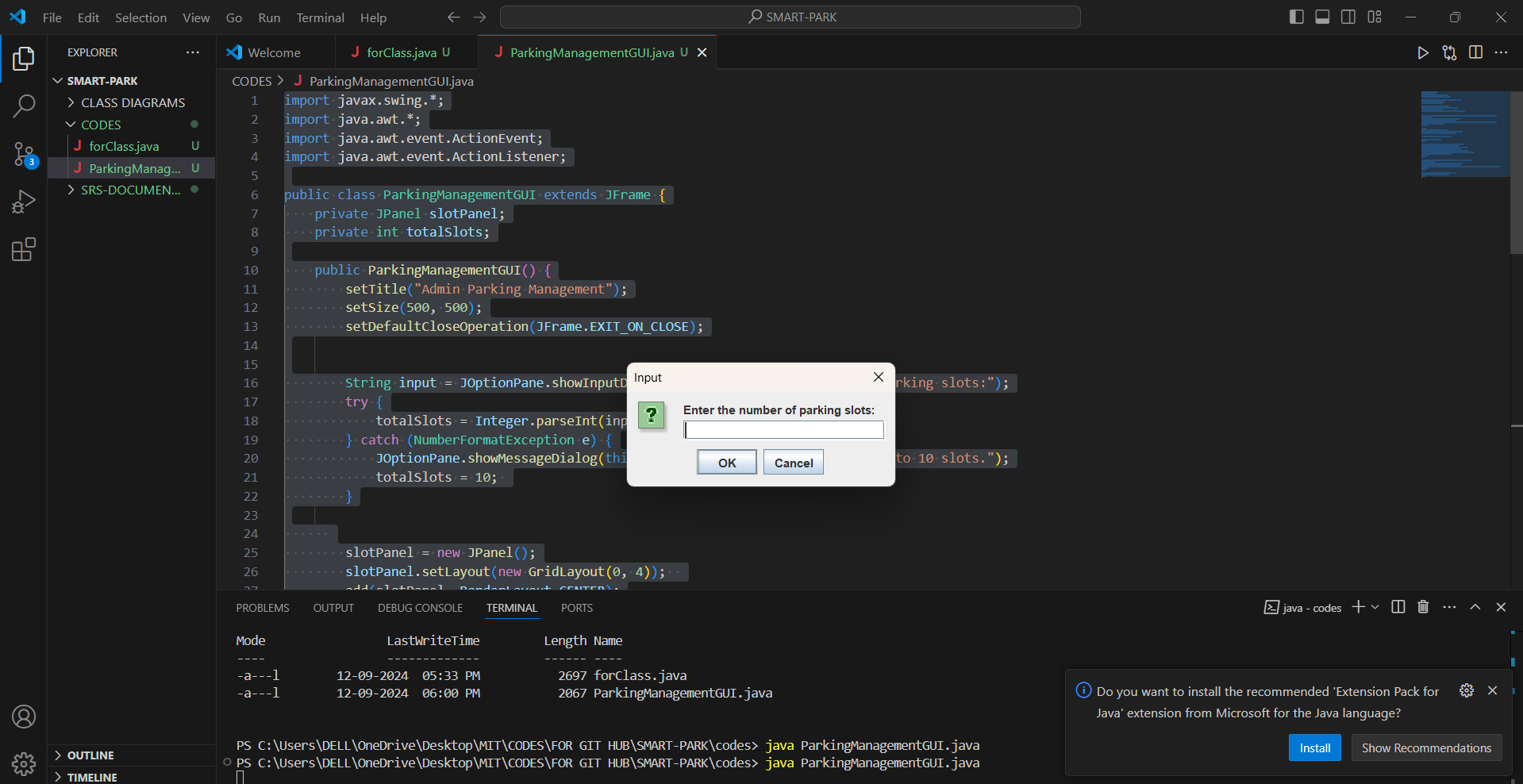
    public static void main(String[] args) {

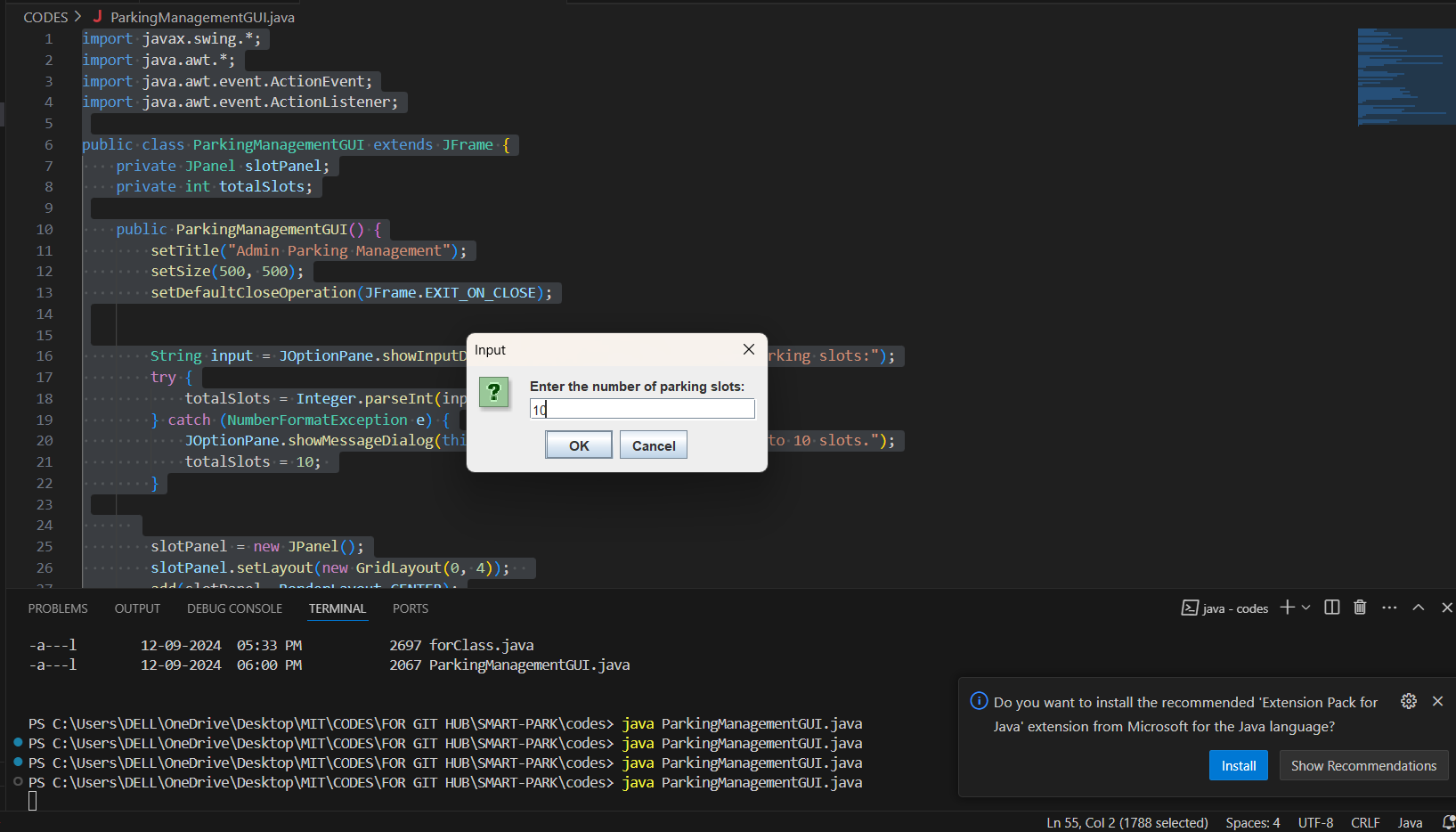
        new ParkingManagementGUI();

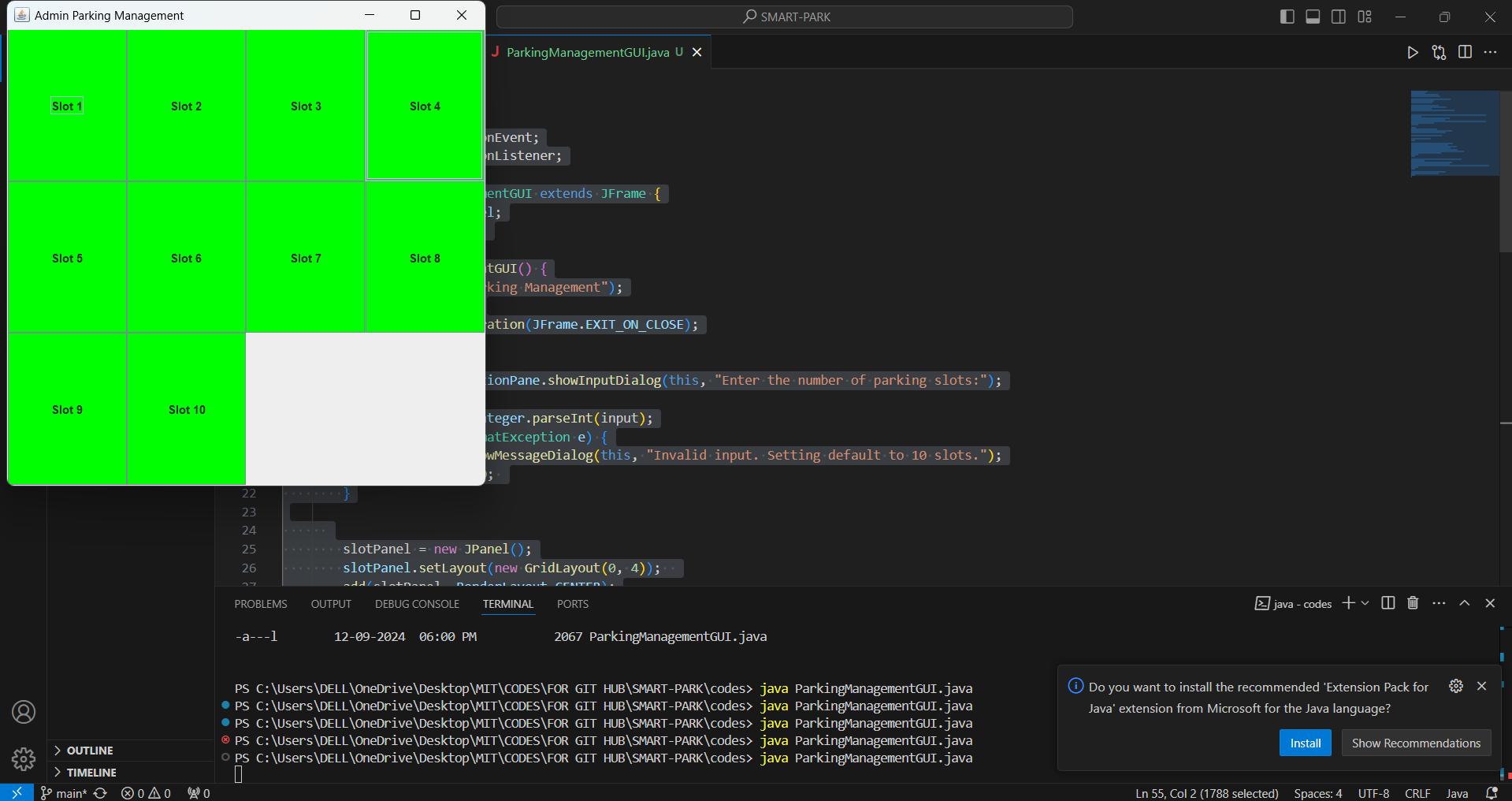
    }

}

**‘I made a clickable gui for admin to create slots!’**



****

****