

```
import javax.swing.*;
import java.awt.*;
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.util.ArrayList;

class Car {
    private String model;
    private int seats;
    private boolean isAvailable;
    private String availableDate;

    public Car(String model, int seats, String availableDate) {
        this.model = model;
        this.seats = seats;
        this.isAvailable = true;
        this.availableDate = availableDate;
    }

    public String getModel() {
        return model;
    }

    public int getSeats() {
        return seats;
    }

    public boolean isAvailable() {
        return isAvailable;
    }

    public String getAvailableDate() {
        return availableDate;
    }

    public void rentCar(int rentalDays) {
        if (isAvailable) {
            isAvailable = false;

            // Calculate return date
            LocalDate startDate = LocalDate.parse(availableDate,
            DateTimeFormatter.ofPattern("yyyy-MM-dd"));
            LocalDate returnDate = startDate.plusDays(rentalDays);
```

```

        JOptionPane.showMessageDialog(null, "You have successfully rented the car: " + model
+
        "\nRental start date: " + availableDate +
        "\nNumber of days to return the car: " + rentalDays +
        "\nExpected return date: " + returnDate);
    } else {
        JOptionPane.showMessageDialog(null, "Sorry, the car is not available.");
    }
}

public void returnCar() {
    if (!isAvailable) {
        isAvailable = true;
        JOptionPane.showMessageDialog(null, "You have successfully returned the car: " +
model);
    } else {
        JOptionPane.showMessageDialog(null, "The car is already available.");
    }
}

public void displayCarDetails(JTextArea textArea) {
    textArea.append("Model: " + model + ", Seats: " + seats + ", Available: " +
        (isAvailable ? "Yes" : "No") + ", Date: " + availableDate + "\n");
}
}

public class CarRentalSystemGUI {
    private JFrame frame;
    private ArrayList<Car> carList;
    private JTextArea textArea;

    public CarRentalSystemGUI() {
        carList = new ArrayList<>();
        initializeCars();
        createGUI();
    }

    private void initializeCars() {
        carList.add(new Car("Toyota Corolla", 5, "2024-12-01"));
        carList.add(new Car("Honda Civic", 5, "2024-12-05"));
        carList.add(new Car("Ford Explorer", 7, "2024-12-10"));
    }

    private void createGUI() {

```

```

frame = new JFrame("Car Rental System");
frame.setSize(600, 500);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

JPanel panel = new JPanel();
panel.setLayout(new BorderLayout());

textArea = new JTextArea();
textArea.setEditable(false);
JScrollPane scrollPane = new JScrollPane(textArea);

JPanel buttonPanel = new JPanel();
buttonPanel.setLayout(new GridLayout(4, 1, 10, 10));

JButton viewCarsButton = new JButton("View Available Cars");
JButton rentCarButton = new JButton("Rent a Car");
JButton returnCarButton = new JButton("Return a Car");
JButton exitButton = new JButton("Exit");

viewCarsButton.addActionListener(e -> displayAvailableCars());
rentCarButton.addActionListener(e -> rentCar());
returnCarButton.addActionListener(e -> returnCar());
exitButton.addActionListener(e -> System.exit(0));

buttonPanel.add(viewCarsButton);
buttonPanel.add(rentCarButton);
buttonPanel.add(returnCarButton);
buttonPanel.add(exitButton);

panel.add(scrollPane, BorderLayout.CENTER);
panel.add(buttonPanel, BorderLayout.EAST);

frame.add(panel);
frame.setVisible(true);
}

private void displayAvailableCars() {
    textArea.setText("Available Cars:\n");
    for (Car car : carList) {
        car.displayCarDetails(textArea);
    }
}

private void rentCar() {

```

```

        String requiredModel = JOptionPane.showInputDialog(frame, "Enter the car model you
want to rent:");
        String requiredSeatsStr = JOptionPane.showInputDialog(frame, "Enter the number of seats
you need:");
        int requiredSeats = Integer.parseInt(requiredSeatsStr);

        boolean carFound = false;
        for (Car car : carList) {
            if (car.getModel().equalsIgnoreCase(requiredModel) && car.getSeats() >= requiredSeats
&& car.isAvailable()) {
                carFound = true;
                String confirmation = JOptionPane.showInputDialog(frame, "Do you want to rent this
car? (yes/no)");
                if (confirmation != null && confirmation.equalsIgnoreCase("yes")) {
                    String availableDate = JOptionPane.showInputDialog(frame, "Enter the available
date for rental (yyyy-MM-dd):");

                    if
(LocalDate.parse(availableDate).isBefore(LocalDate.parse(car.getAvailableDate())) {
                        JOptionPane.showMessageDialog(frame, "The car is not available on the
chosen date.");
                        return;
                    }

                    String rentalDaysStr = JOptionPane.showInputDialog(frame, "Enter the number of
days you want to rent the car:");
                    int rentalDays = Integer.parseInt(rentalDaysStr);
                    car.rentCar(rentalDays);
                    return;
                }
            }
        }
        if (!carFound) {
            JOptionPane.showMessageDialog(frame, "No car found matching your model and seat
requirements.");
        }
    }

    private void returnCar() {
        String model = JOptionPane.showInputDialog(frame, "Enter the car model you want to
return:");

        for (Car car : carList) {
            if (car.getModel().equalsIgnoreCase(model)) {

```

```
        car.returnCar();
        return;
    }
}
JOptionPane.showMessageDialog(frame, "Car model not found.");
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(CarRentalSystemGUI::new);
}
}
```