

CAR RENTAL SYSTEM

GROUP 5

ERFAN SYABIL
AMIRUL HANI
SARANYA
ISWARY



CONTENT

Key topics discussed in this presentation :

- 1 **SECTION A : PROJECT DISCUSSION**
- 2 **SECTION B: CLASS DIAGRAMS**
- 3 **SECTION C: SOURCE CODE AND USER MANUAL**



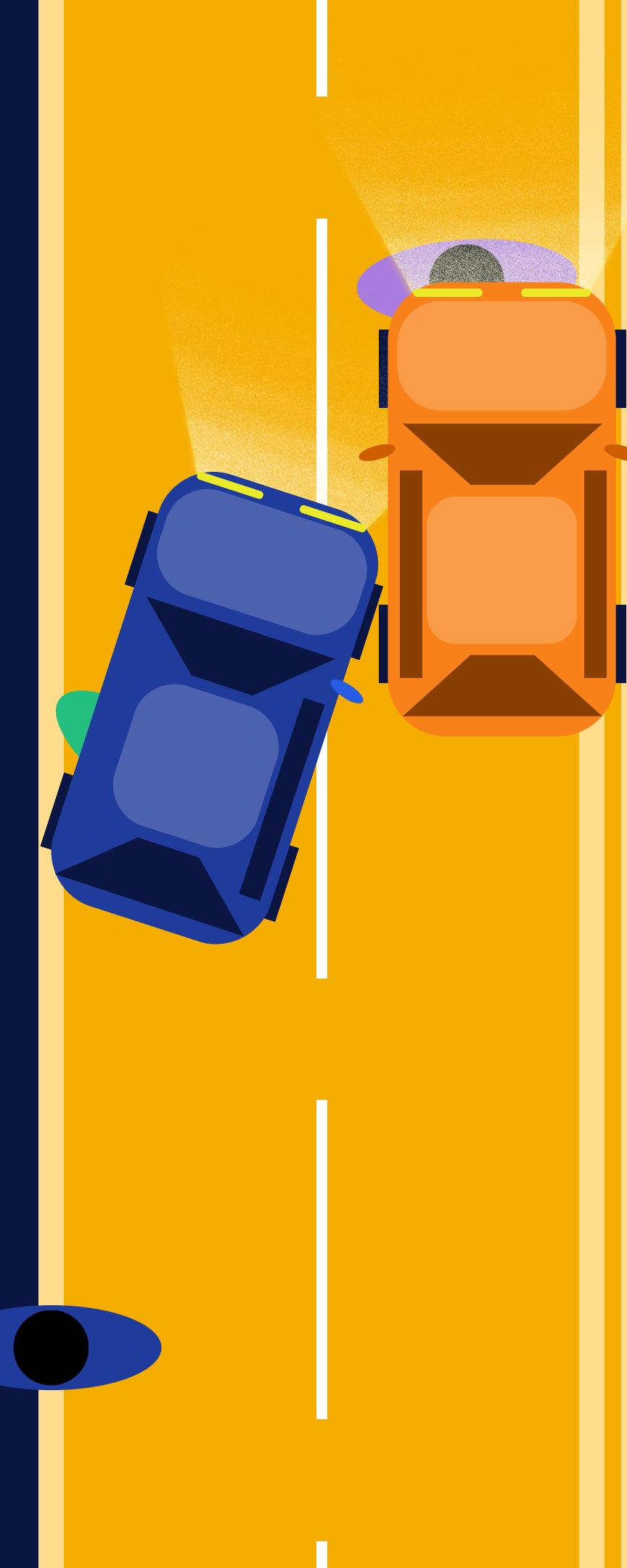


PROJECT DESCRIPTION

The goal of the Car Rental System is to streamline the rental process for suppliers and make renting cars simple and convenient for users. It guarantees correct documentation and makes it simple for customers to reserve cars. This technology makes it possible to schedule, reserve, and rent a range of vehicles. Essentially, the goal of the car rental system is to make renting a car easy and accessible for consumers while also assisting rental companies in running efficiently.

OBJECTIVES

The objective of the Car Rental System is to ease the car rental process for customers. The system aims to enhance the overall experience of renting a car by providing a user-friendly and simple interface, ensuring accurate documentation and facilitating efficient management for rental companies.





SCOPE

The Car Rental System's scope includes various functions that help in renting cars efficiently and user-friendly. This system is designed to satisfy the needs of customers. Below is the description of the scope:

1. Customer registration

- Users have the option to register customers and view their details under customer information.

2. Car rental confirmation and car rental history

- Users can select cars and specify pickup and return location.
- Confirmation details and rental history can be previewed.

3. Pickup address and return location address

- Users can view details about pickup and return location.

4. System usage steps:

- View available car and rental prices
- Add customer details
- View all customer details
- Rent a car
- View rental history
- Exit



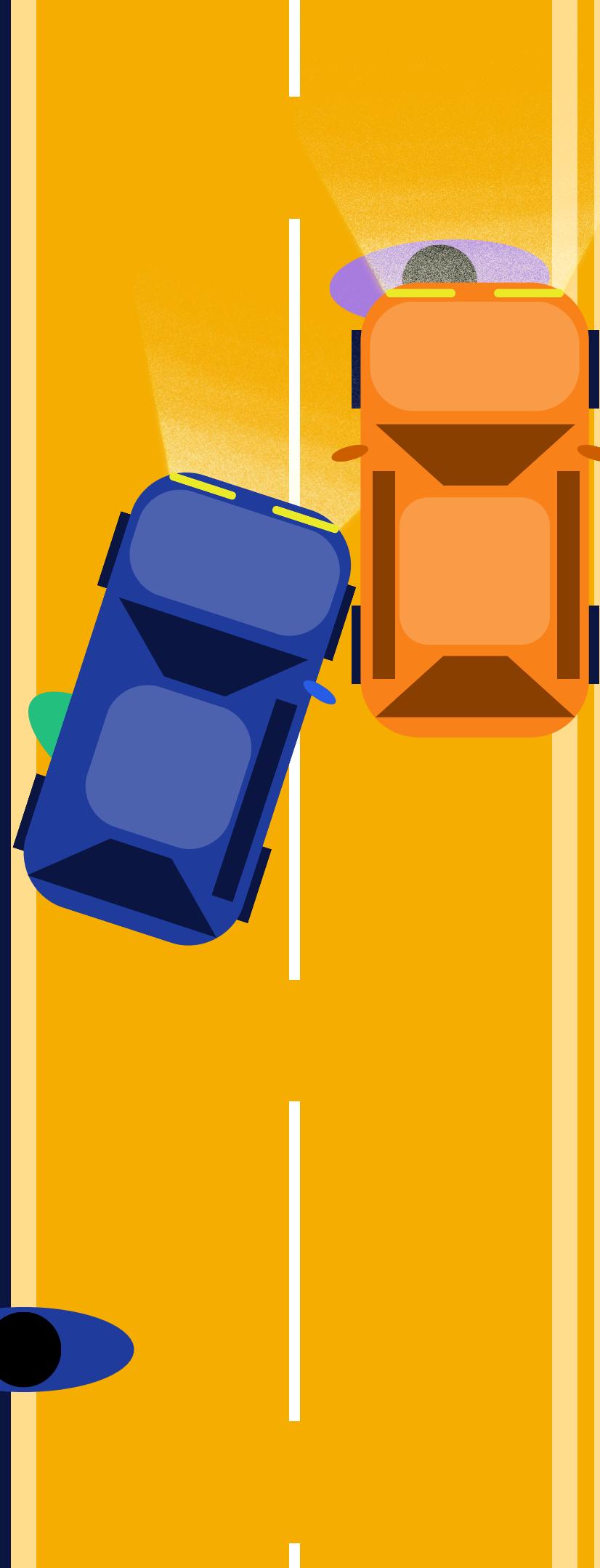
WORKFLOW



1. **Begin**
2. Initialize RentalSystem object: rentalSystem
3. Initialize Customer object: customer (set to null initially)

4. Loop until the user chooses to exit:
 - a. Display the Car Rental System Menu:
 1. View available vehicles and rental prices
 2. Add user details
 3. View all customer details
 4. Rent a car
 5. View rental history
 6. Exit
 - b. Prompt the user to enter their choice (1-6).
 - c. Read the user's choice.

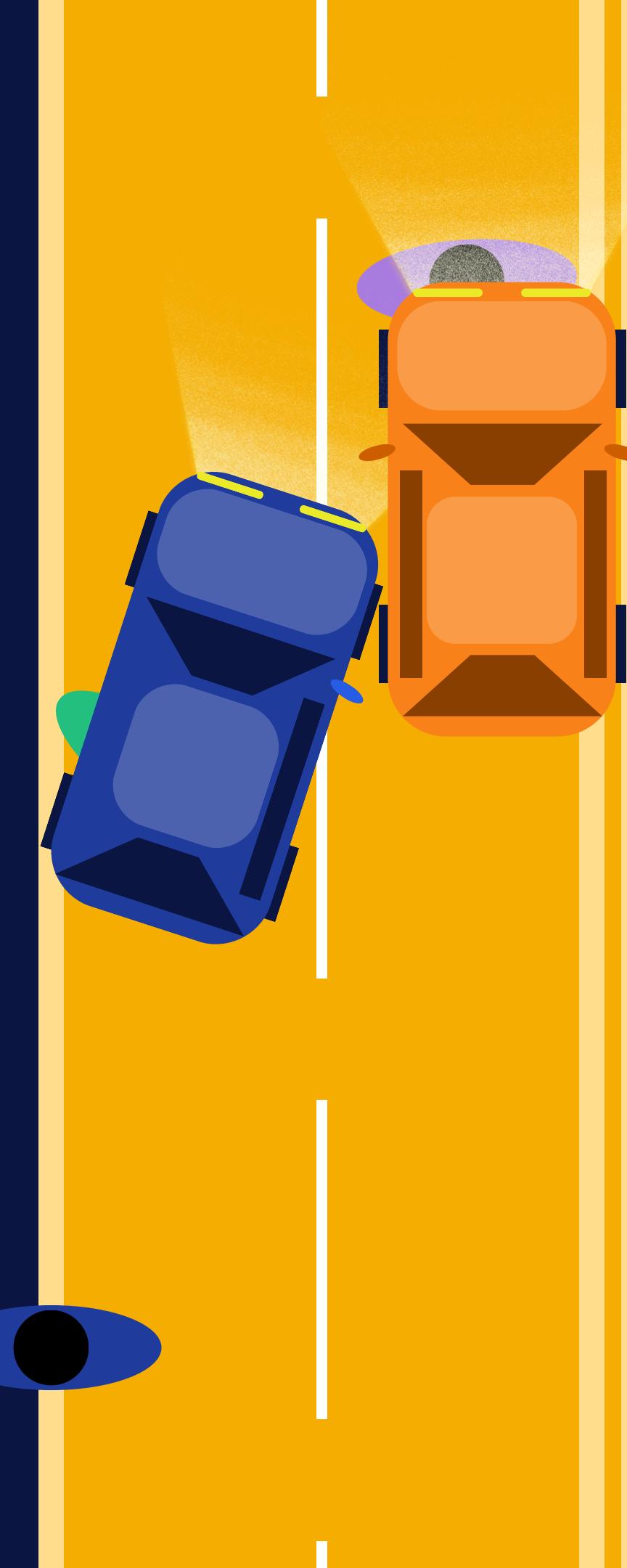
 - d. Switch based on the user's choice:
 - Case 1:





WORKFLOW

- Call rentalSystem.displayAvailableCar()
- Case 2:
 - Prompt the user to enter customer details (name, licenseNumber, address, email, phoneNumber).
 - Create a new Customer object with the entered details.
 - Call rentalSystem.addUserDetails(customer)
- Case 3:
 - Call rentalSystem.printAllCustomers()
- Case 4:
 - IF rentalSystem.hasCustomerDetails() // user details (customer) exist:
 - Call rentalSystem.printAllCustomers().
 - Prompt user to enter index of customer
 - IF customerIndex >= 1 && customerIndex <= rentalSystem.getCustomers().size()



WORKFLOW



- Get selected customer based on index
 - Call rentalSystem.displayPickupLocations()
 - Prompt user to enter index of pickup location
 - Get selected pickup location
 - Call rentalSystem.displayAvailableCar()
 - Prompt user to enter index of vehicle to reserve
 - Get selected car based on index
 - Call rentalSystem.displayReturnLocations()
 - Prompt user to enter index of return location
 - Get selected return location
 - Schedule an appointment by calling
rentalSystem.scheduleAppointment()
 - Prompt user to enter rental duration in days
 - Get entered rental duration
 - Call rentalSystem.rentVehicle(customer, selectedCar,
appointment, pickupLocation, returnLocation, rentalDays)
 - ELSE Print “Invalid customer index.”
- ELSE Print “Please add user details first.”

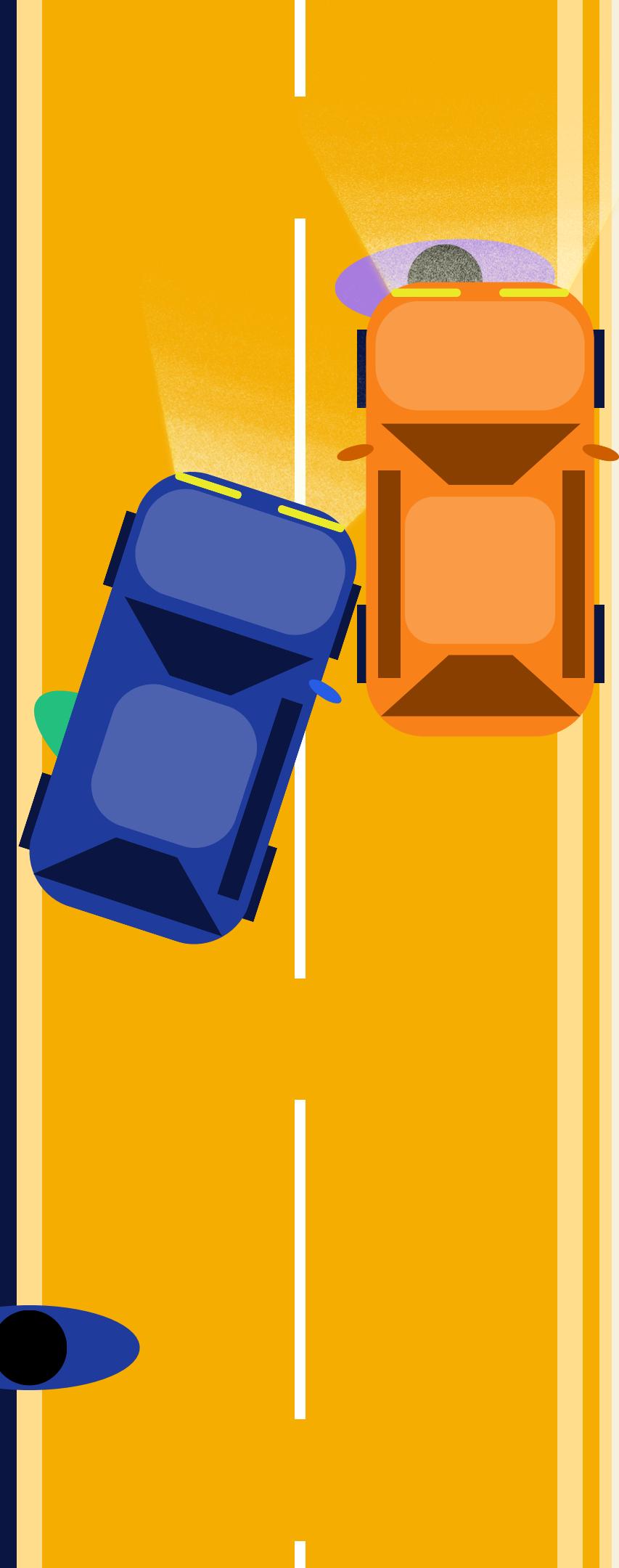


WORKFLOW



- Case 5:
 - Call rentalSystem.displayRentals()
 - Case 6:
 - Print message "Exiting Rental System. Thank you!".
 - Default:
 - Print message "Invalid choice. Please enter a number between 1 and 6".
6. Handle exceptions if they occur
7. Close scanner by calling rentalSystem.closeScanner()

End





OO CONCEPTS

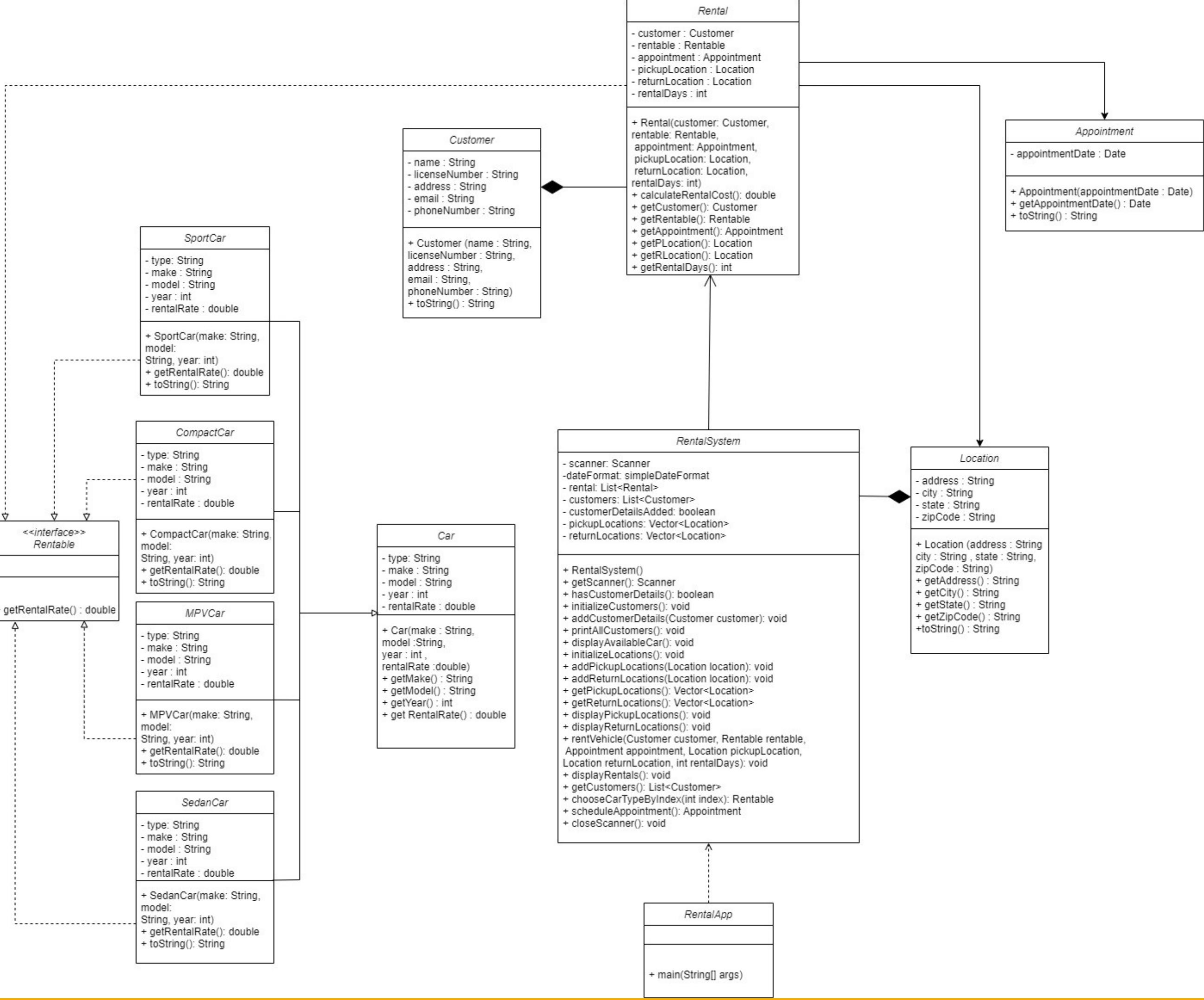
CLASS AND OBJECT	<p>Car, SportCar, Customer, Appointment , CompactCar , Location, MPVCar, Rentable, Rental, RentalApp, RentalSystem and SedanCar.</p>
INHERITANCE	<p>SportCar, MPVCar, SedanCar, and CompactCar extend the Car class</p>
ABSTRACTION	<p>Abstract classes (Car) and interfaces (Rentable) are used to abstract common properties and behaviors, allowing for code reusability and the definition of a common interface for rental rates.</p>
ENCAPSULATION	<p>Data hiding and encapsulation are achieved by making the fields of classes (Car, Customer, etc.) private and providing public methods to access and manipulate them. This protects the internal state of objects.</p>



OO CONCEPTS

COMPOSITION	<p>The system uses composition to build complex objects. Rental object includes instances of Customer, Rentable (vehicle), Appointment, and Location.</p>
POLYMORPHISM	<p>Through method overriding. For example, the <code>getRentalRate</code> method is overridden in each vehicle class (<code>SportCar</code>, <code>MPVCar</code>, <code>SedanCar</code> and <code>CompactCar</code>) to provide different rental rates.</p>
INTERFACE	<p>Rentable interface defines a common method (<code>getRentalRate</code>) that is implemented by vehicle classes (<code>SportCar</code>, <code>MPVCar</code>, etc.) . This allows for a consistent way to retrieve rental rates.</p>

UML DIAGRAM





SECTION C

Source Code and User Manual

RentalSystem.java

```
J RentalSystem.java X
source_code > J RentalSystem.java > ...
1 //AMIRUL HANI BIN SYAFRIZON
2 //B23CS0025
3 // RentalSystem.java
4
5 import java.util.ArrayList;
6 import java.util.Date;
7 import java.util.List;
8 import java.util.Scanner;
9 import java.util.Vector;
10 import java.text.*;
11
12 public class RentalSystem {
13     //variable
14     private Scanner scanner;
15     private SimpleDateFormat dateFormat;
16     private List<Rental> rental;
17     private List<Customer> customers;
18     private boolean customerDetailsAdded = false;
19     private Vector<Location> pickupLocations;
20     private Vector<Location> returnLocations;
21
22     //constructor
23     public RentalSystem() {
24         scanner = new Scanner(System.in);
25         customerDetailsAdded = false;
26         customers = new ArrayList<>(); // Initialize the customers list
27         rental = new ArrayList<>();
28         dateFormat = new SimpleDateFormat(pattern:"yyyy-MM-dd HH:mm:ss");
29         pickupLocations = new Vector<>();
30         returnLocations = new Vector<>();
31         initializeLocations(); // Initialize locations when the system is created
32         initializeCustomers(); // Initialize customers when the system is created
33     }
34
35     //scanner method for input
36     public Scanner getScanner() {
37         return scanner;
38     }
39 }
```



SECTION C

Source Code and User Manual

RentalSystem.java

```
72 }
73
74 //Display all available Car by retrieving the list in chooseCarTypeByIndex() method
75 public void displayAvailableCar() {
76     System.out.println("\n--- Available Car and Rental Prices ---");
77     System.out.println("\n Type" + "\t\t" + "Make" + "\t" + "Model" + "\t" + " Year" + "\t" + "Rental Price");
78     for (int i = 1; i <= 4; i++) {
79         Rentable car = chooseCarTypeByIndex(1);
80         System.out.println(i + ". " + car.toString() + "\t" + "RM" + car.getRentalRate() + " per day");
81     }
82 }
83
84 //Initialize pickup and return location
85 private void initializeLocations() {
86     //Pickup Location
87     pickupLocations.add(new Location(address:"123 BP 11", city:"Puchong", state:"Selangor", zipCode:"47120"));
88     pickupLocations.add(new Location(address:"456 NP 32", city:"Kulai", state:"Johor", zipCode:"81310"));
89     pickupLocations.add(new Location(address:"789 Villa", city:"Kuantan", state:"Pahang", zipCode:"43543"));
90     pickupLocations.add(new Location(address:"102 Elm St", city:"Gombak", state:"Kuala Lumpur", zipCode:"91232"));
91     pickupLocations.add(new Location(address:"879 Oak St", city:"Ipoh", state:"Perak", zipCode:"30000"));
92     pickupLocations.add(new Location(address:"321 Pine St", city:"Jeli", state:"Kelantan", zipCode:"54321"));
93
94     //Return Location
95     returnLocations.add(new Location(address:"123 BP 11", city:"Puchong", state:"Selangor", zipCode:"47120"));
96     returnLocations.add(new Location(address:"456 NP 32", city:"Kulai", state:"Johor", zipCode:"81310"));
97     returnLocations.add(new Location(address:"879 Villa", city:"Kuantan", state:"Pahang", zipCode:"43543"));
98     returnLocations.add(new Location(address:"102 Elm St", city:"Gombak", state:"Kuala Lumpur", zipCode:"91232"));
99     returnLocations.add(new Location(address:"879 Oak St", city:"Ipoh", state:"Perak", zipCode:"30000"));
100    returnLocations.add(new Location(address:"321 Pine St", city:"Jeli", state:"Kelantan", zipCode:"54321"));
101 }
102
103 //Add pickup location into the vector location
104 public void addPickupLocation(Location location) {
105     pickupLocations.add(location);
```



SECTION C

Source Code and User Manual

RentalSystem.java

```
186
187
188     //Add return location into the vector location
189     public void addReturnLocation(Location location) {
190         returnLocations.add(location);
191     }
192
193     //Retrieve pickup location object
194     public Vector<Location> getPickupLocations() {
195         return pickupLocations;
196     }
197
198     //Retrieve return location object
199     public Vector<Location> getReturnLocations() {
200         return returnLocations;
201     }
202
203     //Display pickup location
204     public void displayPickupLocations() {
205         System.out.println("----- Pickup Locations -----");
206         int i = 1;
207         for (Location pickuplocation : pickupLocations) {
208             System.out.println(i + ". " + pickuplocation.getAddress() + ", " + pickuplocation.getCity() + ", " + pickuplocation.getState() + ", " + pickuplocation.getZipCode());
209             i++;
210         }
211     }
212
213     //Display return location
214     public void displayReturnLocations() {
215         System.out.println("----- Return Locations -----");
216     }
```

```
196     int i = 1;
197     for (Location returnlocation : returnLocations) {
198         System.out.println(i + ". " + returnlocation.getAddress() + ", " + returnlocation.getCity() + ", " + returnlocation.getState() + ", " + returnlocation.getZipCode());
199         i++;
200     }
201
202
203     //Rent a vehicle and after that produce output containing info about customer, car rented, pickup time, pickup and return location, duration and total.
204     public void rentVehicle(Customer customer, Rentable rentable, Appointment appointment, Location pickuplocation, Location returnlocation, int rentalDays) {
205         Rental rental = new Rental(customer, rentable, appointment, pickuplocation, returnlocation, rentalDays);
206         this.rental.add(rental);
207
208         System.out.println("----- Rental -----");
209         System.out.println("Rental Successful for: " + customer.getName());
210
211         System.out.println("Name: " + customer.getName());
212         System.out.println("License Number: " + customer.getLicenseNumber());
213         System.out.println("Address: " + customer.getAddress());
214         System.out.println("Email: " + customer.getEmail());
215         System.out.println("Phone Number: " + customer.getPhoneNumber());
216
217         System.out.println("\nCar rented: " + rentable);
218         System.out.println("Pickup date: " + rental.getAppointment().getAppointmentDate());
219         System.out.println("Pickup location: " + rental.getLocation().toString());
220         System.out.println("Return location: " + rental.getLocation().toString());
221         System.out.println("Duration: " + rental.getRentalDays() + " day(s)");
222         System.out.printf(format("Total: %.2f", rental.calculateRentalCost()));
223         System.out.println("\n-----");
224     }
225
226
227     //To display rental history
228     public void displayRentals() {
229         System.out.println("----- Rental History -----");
230         for (Rental rental : rentals) {
231             System.out.println("Name: " + rental.getCustomer().getName());
232             System.out.println("License Number: " + rental.getCustomer().getLicenseNumber());
233             System.out.println("Address: " + rental.getCustomer().getAddress());
234             System.out.println("Email: " + rental.getCustomer().getEmail());
235             System.out.println("Phone Number: " + rental.getCustomer().getPhoneNumber());
236
237             System.out.println("\nCar rented: " + rental.getRentable());
238             System.out.println("Pickup date: " + rental.getAppointment().getAppointmentDate());
239             System.out.println("Pickup location: " + rental.getLocation().toString());
240             System.out.println("Return location: " + rental.getLocation().toString());
241         }
242     }
243 }
```



SECTION C

Source Code and User Manual

RentalSystem.java

```
181     System.out.println("Duration: " + rental.getRentalDays() + " day(s)");
182     System.out.printf(format:"\nTotal: RM%.2f", rental.calculateRentalCost());
183     System.out.println(x:"\n-----");
184 }
185 }
186
187 //Close scanner to prevent resource leak
188 public void closeScanner() {
189     scanner.close();
190 }
191
192 //Initialize car into arrayList and return the list index
193 public Rentable chooseCarTypeByIndex(int carIndex) {
194     // Choose a vehicle based on index
195     List<Rentable> availableCar = new ArrayList<>();
196     availableCar.add(new CompactCar(type:"Compact",make:"Perodua", model:"Axia", year:2022));
197     availableCar.add(new SportCar(type:"Sports", make:"Ford", model:"Mustang", year:2022));
198     availableCar.add(new SedanCar(type:"Sedan", make:"Proton", model:"Saga", year:2022));
199     availableCar.add(new MPVCar(type:"MPV\lt", make:"Honda", model:"BRV", year:2022));
200
201     if (carIndex >= 1 && carIndex <= availableCar.size()) {
202         return availableCar.get(carIndex - 1);
203     } else {
204         throw new IllegalArgumentException(s:"Invalid vehicle index.");
205     }
206 }
207
208 //Set appointment using appropriate format of date and time
209 public Appointment scheduleAppointment() {
210     System.out.print(s:"Enter appointment date (yyyy-MM-dd HH:mm:ss): ");
211     String dateString = scanner.nextLine();
212
213     try {
214         Date appointmentDate = dateFormat.parse(dateString);
215         return new Appointment(appointmentDate);
216     } catch (ParseException e) {
217         System.out.println(x:"Invalid date format. Defaulting to current date and time.");
218         return new Appointment(new Date());
219     }
220 }
221
222 //Retrieve customer object.
223 public List<Customer> getCustomers() {
224     return customers;
225 }
```



SECTION C

Source Code and User Manual

RentalApp.java



SECTION C

Source Code and User Manual

RentalApp.java

```
RentalApp.java X
source_code > RentalApp.java > ...
74 rentalSystem.displayPickupLocations();
75 System.out.print(s:"Enter the index of the pickup location: ");
76 int pickupIndex = rentalSystem.getScanner().nextInt();
77 rentalSystem.getScanner().nextLine();
78
79 Location pickupLocation = rentalSystem.getPickupLocations().get(pickupIndex - 1);
80
81
82 //choose car to rent
83 rentalSystem.displayAvailableCar();
84 System.out.print(s:"Enter the car index to reserve (1-4): ");
85 int carIndex = rentalSystem.getScanner().nextInt();
86 rentalSystem.getScanner().nextLine();
87 Rentable selectedCar = rentalSystem.chooseCarTypeByIndex(carIndex);
88
89 // Choose return location
90 rentalSystem.displayReturnLocations();
91 System.out.print(s:"Enter the index of the return location: ");
92 int returnIndex = rentalSystem.getScanner().nextInt();
93 rentalSystem.getScanner().nextLine();
94
95 Location returnLocation = rentalSystem.getReturnLocations().get(returnIndex - 1);
96
97 Appointment appointment = rentalSystem.scheduleAppointment();
98
99 System.out.print(s:"Enter the duration of rental in days: ");
100 int rentalDays = rentalSystem.getScanner().nextInt();
101 rentalSystem.getScanner().nextLine();
102
103 rentalSystem.rentVehicle(customer, selectedCar, appointment, pickupLocation, returnLocation, rentalDays);
104 } else {
105     System.out.println(x:"invalid customer index.");
106 }
107 } else {
108     System.out.println(x:"Please add user details first.");
109 }
110 break;
```

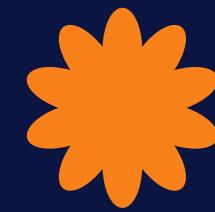


SECTION C

Source Code and User Manual

Rental.java

```
RentalApp.java x
source_code > RentalApp.java > ...
111 |           case 5:
112 |               // View rental history
113 |               rentalSystem.displayRentals();
114 |               break;
115 |           case 6:
116 |               System.out.println("Exiting Rental System. Thank you!");
117 |               break;
118 |           default:
119 |               System.out.println("Invalid choice. Please enter a number between 1 and 5.");
120 |
121     } while (choice != 6);
122
123 } catch (Exception e) {
124     System.out.println("An error occurred: " + e.getMessage());
125 } finally {
126     // Close the scanner
127     rentalSystem.closeScanner();
128 }
129
130 }
131
```



SECTION C

Source Code and User Manual

Rental.java

```
Rental.java X
source_code > J Rental.java > ...
1 //SARANYA A/P JAYARAMA REDDY (833CC381)
2
3 public class Rental {
4     private Customer customer;
5     private Rentable rentable;
6     private Appointment appointment;
7     private Location pickuplocation;
8     private Location returnlocation;
9     private int rentalDays;
10
11     public Rental(Customer customer, Rentable rentable, Appointment appointment, Location pickuplocation, Location returnlocation, int rentalDays){
12         this.customer = customer;
13         this.rentable = rentable;
14         this.appointment = appointment;
15         this.pickuplocation = pickuplocation;
16         this.returnlocation = returnlocation;
17         this.rentalDays = rentalDays;
18     }
19
20     //Method to calculate the total rental
21     public double calculateRentalCost(){
22         return rentable.getRentalRate() * rentalDays;
23     }
24
25     //Getter method
26     public Customer getCustomer() {
27         return customer;
28     }
29
30     public Rentable getRentable() {
31         return rentable;
32     }
33
34     public Appointment getAppointment() {
35         return appointment;
36     }
37
38     public Location getPickupLocation(){
39         return pickuplocation;
40     }
41
42     public Location getReturnLocation(){
43         return returnlocation;
44     }
45
46     public int getRentalDays(){
47         return rentalDays;
48     }
49 }
```



SECTION C

Source Code and User Manual

Rentalable.java

```
J Rentable.java X
source_code > J Rentable.java > •○ Rentable
1 //SARANYA A/P JAYARAMA REDDY (B22EC3013)
2
3 public interface Rentable {
4     double getRentalRate();
5 }
6
```



SECTION C

Source Code and User Manual

Location.java

```
J Location.java X
source_code > J Location.java > ...
1 //ISWARY:022EC3004
2
3 public class Location {
4     private String address;
5     private String city;
6     private String state;
7     private String zipCode;
8
9     public Location(String address, String city, String state, String zipCode) {
10        this.address = address;
11        this.city = city;
12        this.state = state;
13        this.zipCode = zipCode;
14    }
15
16    // Getter methods
17    public String getAddress() {
18        return address;
19    }
20
21    public String getCity() {
22        return city;
23    }
24
25    public String getState() {
26        return state;
27    }
28
29    public String getZipCode() {
30        return zipCode;
31    }
32
33    //toString method to get location as a String
34    public String toString(){
35        return getAddress() + ", " + getCity() + ", " + getState() + ", " + getZipCode();
36    }
37 }
```



SECTION C

Source Code and User Manual

Appoinment.java

```
J Appointment.java •  
source_code > J Appointment.java > ...  
1 //Iswary:B22EC3004  
2  
3 import java.util.Date;  
4  
5 public class Appointment {  
6     private Date appointmentDate;  
7  
8     public Appointment(Date appointmentDate) {  
9         this.appointmentDate = appointmentDate;  
10    }  
11  
12    public Date getAppointmentDate() {  
13        return appointmentDate;  
14    }  
15 }
```



SECTION C

Source Code and User Manual

Customer.java

```
J Customer.java X
source_code > J Customer.java > ...
1 //SARANYA A/P JAYARAMA REDDY (B22EC3013)
2
3 public class Customer {
4     private String name;
5     private String licenseNumber;
6     private String address;
7     private String email;
8     private String phoneNumber;
9
10    public Customer(String name, String licenseNumber, String address, String email, String phoneNumber) {
11        this.name = name;
12        this.licenseNumber = licenseNumber;
13        this.address = address;
14        this.email = email;
15        this.phoneNumber = phoneNumber;
16    }
17
18    //Getter method
19    public String getName(){
20        return name;
21    }
22
23    public String getLicenseNumber(){
24        return licenseNumber;
25    }
26
27    public String getAddress(){
28        return address;
29    }
30
31    public String getEmail(){
32        return email;
33    }
34
35    public String getPhoneNumber(){
36        return phoneNumber;
37    }
38 }
```



SECTION C

Source Code and User Manual

Car.java

```
J Carjava ●
source_code > J Carjava > ...
1 //MUHAMMAD ERFAN SYABIL BIN ESA
2 //B23CS0055
3 // Car.java
4 //Superclass
5 public abstract class Car implements Rentable {
6     protected String type;
7     protected String make;
8     protected String model;
9     protected int year;
10    protected double rentalRate;
11
12    public Car(String type, String make, String model, int year, double rentalRate) {
13        this.type = type;
14        this.make = make;
15        this.model = model;
16        this.year = year;
17        this.rentalRate = rentalRate;
18    }
19
20    //Getter method
21    public String getType(){
22        return type;
23    }
24
25    public String getMake(){
26        return make;
27    }
28
29    public String getModel(){
30        return model;
31    }
32
33    public int getYear(){
34        return year;
35    }
36
37    public abstract double getRentalRate();
38
39
```



SECTION C

Source Code and User Manual

CompactCar.java

```
CompactCar.java
source_code > CompactCar.java > ...
1 //MUHAMMAD ERFAN SYABIL BIN ESA
2 //B2BCS0055
3 //Compact.java
4 //Subclass to Car
5 class CompactCar extends Car{ //Shows Inheritance
6     public CompactCar(String type, String make, String model, int year) {
7         super(type, make, model, year, rentalRate:50.00); //use super keyword to because of inheritance
8     }
9
10    public double getRentalRate(){
11        return 50.00; //Rate per day
12    }
13
14    //toString() method is used to get the whole car name, make, model and year
15    public String toString() {
16        return getType() + "\t" + getMake() + "\t" + getModel() + "\t" + "(" + getYear() + ")";
17    }
18
19 }
```



SECTION C

Source Code and User Manual

MPVCar.java

```
MPVCar.java X
source_code > MPVCar.java > ...
1 //MUHAMMAD ERFAN SYABIL BIN ESA
2 //B23CS0055
3 //MPVCar.java
4 //Subclass to Car
5 class MPVCar extends Car { //Shows Inheritance
6     public MPVCar(String type, String make, String model, int year) {
7         super(type, make, model, year, rentalRate:100.00); //use super keyword to because of inheritance
8     }
9
10    public double getRentalRate(){
11        return 100.00; //Rate per day
12    }
13
14    //toString() method is used to get the whole car name, make, model and year
15    public String toString(){
16        return getType() + "\t" + getMake() + "\t" + getModel() + "\t" + "(" + getYear() + ")";
17    }
18 }
```



SECTION C

Source Code and User Manual

SedanCar.java

```
J SedanCar.java X
source_code > J SedanCar.java > ...
1
2 //MUHAMMAD ERFAN SYABIL BIN ESA
3 //B23CS0055
4 //Sedan.java
5 //Subclass to Car
6 class SedanCar extends Car { //Shows Inheritance
7     public SedanCar(String type, String make, String model, int year) {
8         super(type, make, model, year, rentalRate:80.00); //use super keyword to because of inheritance
9     }
10
11    public double getRentalRate(){
12        return 80.00; //Rate per day
13    }
14
15    //toString() method is used to get the whole car name, make, model and year
16    public String toString() {
17        return getType() + "\t" + getMake() + "\t" + getModel() + "\t" + "(" + getYear() + ")";
18    }
19 }
```



SECTION C

Source Code and User Manual

SportCar.java

```
J SportCar.java X
source code > J SportCar.java > ...
1 //MUHAMMAD ERFAN SYABIL BIN ESA
2 //B23CS0055
3 // Sports.java
4 //Subclass to Car
5 class SportCar extends Car{ //Shows Inheritance
6     public SportCar(String type, String make, String model, int year) {
7         super(type, make, model, year, rentalRate:300.00); //use super keyword to because of inheritance
8     }
9
10    public double getRentalRate(){
11        return 300.00; //Rate per day
12    }
13
14    //toString() method is used to get the whole car name, make, model and year
15    public String toString() {
16        return getType() + "\t" + getMake() + "\t" + getModel() + "\t" + "(" + getYear() + ")";
17    }
18 }
```

USER MANUAL

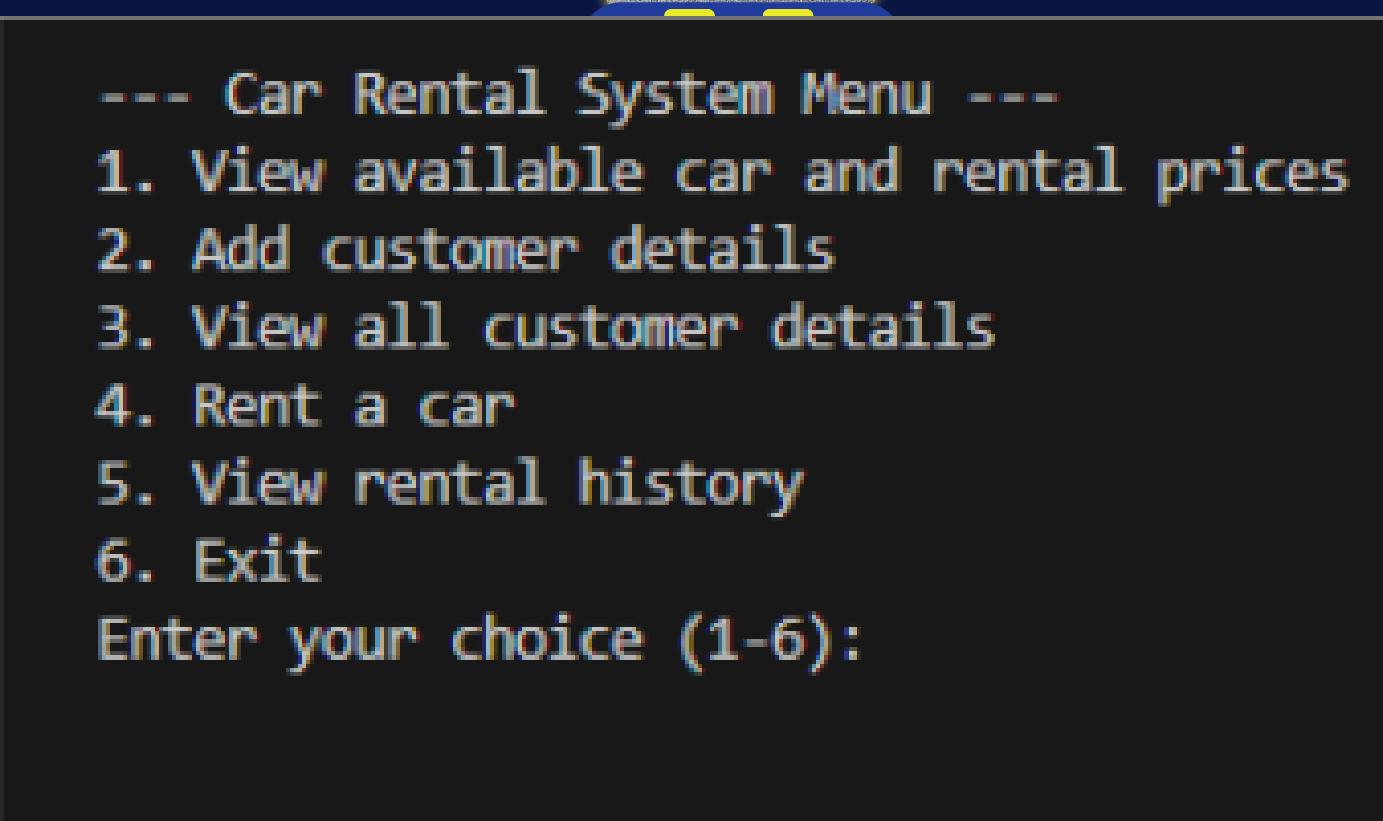
Main Menu

User greeted with the simple main menu.

It has 6 choices and the purposes of those choices are:

1. User can view the available cars for rent and their daily rental prices.
2. The user can add a customer detail such as their name, license number, address, email, and phone number.
3. The user can view all the multiple customer details.
4. This is the essential part of the system where the user can rent a vehicle.
5. User can track their rental history with this.
6. Exit the program.

User must enter one of the choices based on their needs.



--- Car Rental System Menu ---

1. View available car and rental prices
2. Add customer details
3. View all customer details
4. Rent a car
5. View rental history
6. Exit

Enter your choice (1-6):

USER MANUAL

Error (User entered none of the choices)

If a user entered a choice number other than 1 to 6 it will display:

“Invalid choice. Please enter a number between 1 to 6”

It will loop back to the main menu.

```
--- Car Rental System Menu ---
1. View available car and rental prices
2. Add customer details
3. View all customer details
4. Rent a car
5. View rental history
6. Exit
Enter your choice (1-6): 7
Invalid choice. Please enter a number between 1 and 6.

--- Car Rental System Menu ---
1. View available car and rental prices
2. Add customer details
3. View all customer details
4. Rent a car
5. View rental history
6. Exit
Enter your choice (1-6):
```

USER MANUAL

Display available car:

If the user enters 1 as one of their choices, it will display a table of all the available cars based on type, make, model, year and their rental price per day.

Enter your choice (1-6): 1

--- Available Car and Rental Prices ---

Type	Make	Model	Year	Rental Price
1. Compact	Perodua	Axia	(2022)	RM50.0 per day
2. Sports	Ford	Mustang	(2022)	RM300.0 per day
3. Sedan	Proton	Saga	(2022)	RM80.0 per day
4. MPV	Honda	BRV	(2022)	RM100.0 per day

--- Car Rental System Menu ---

1. View available car and rental prices
2. Add customer details
3. View all customer details
4. Rent a car
5. View rental history
6. Exit

Enter your choice (1-6): 1

USER MANUAL

Add customer details:

```
--- Car Rental System Menu ---
1. View available car and rental prices
2. Add customer details
3. View all customer details
4. Rent a car
5. View rental history
6. Exit
Enter your choice (1-6): 2

Enter customer name: Amirah
Enter customer license number: 020813060296
Enter customer address: Dungun, Terengganu
Enter customer email: nuramierah@gmail.com
Enter customer phone number: 0179213496

Customer details added:
Name: Amirah
License Number: 020813060296
Address: Dungun, Terengganu
Email: nuramierah@gmail.com
Phone No: 0179213496
```

If the user enters 2 as one of their choices, it will allow the user to enter customer details. Start with the customer name, license number, address, email and phone number.

After that, it will print back the output of the information that the user entered.



USER MANUAL

```
--- Car Rental System Menu ---
1. View available car and rental prices
2. Add customer details
3. View all customer details
4. Rent a car
5. View rental history
6. Exit
Enter your choice (1-6): 3

--- All Customer Information ---

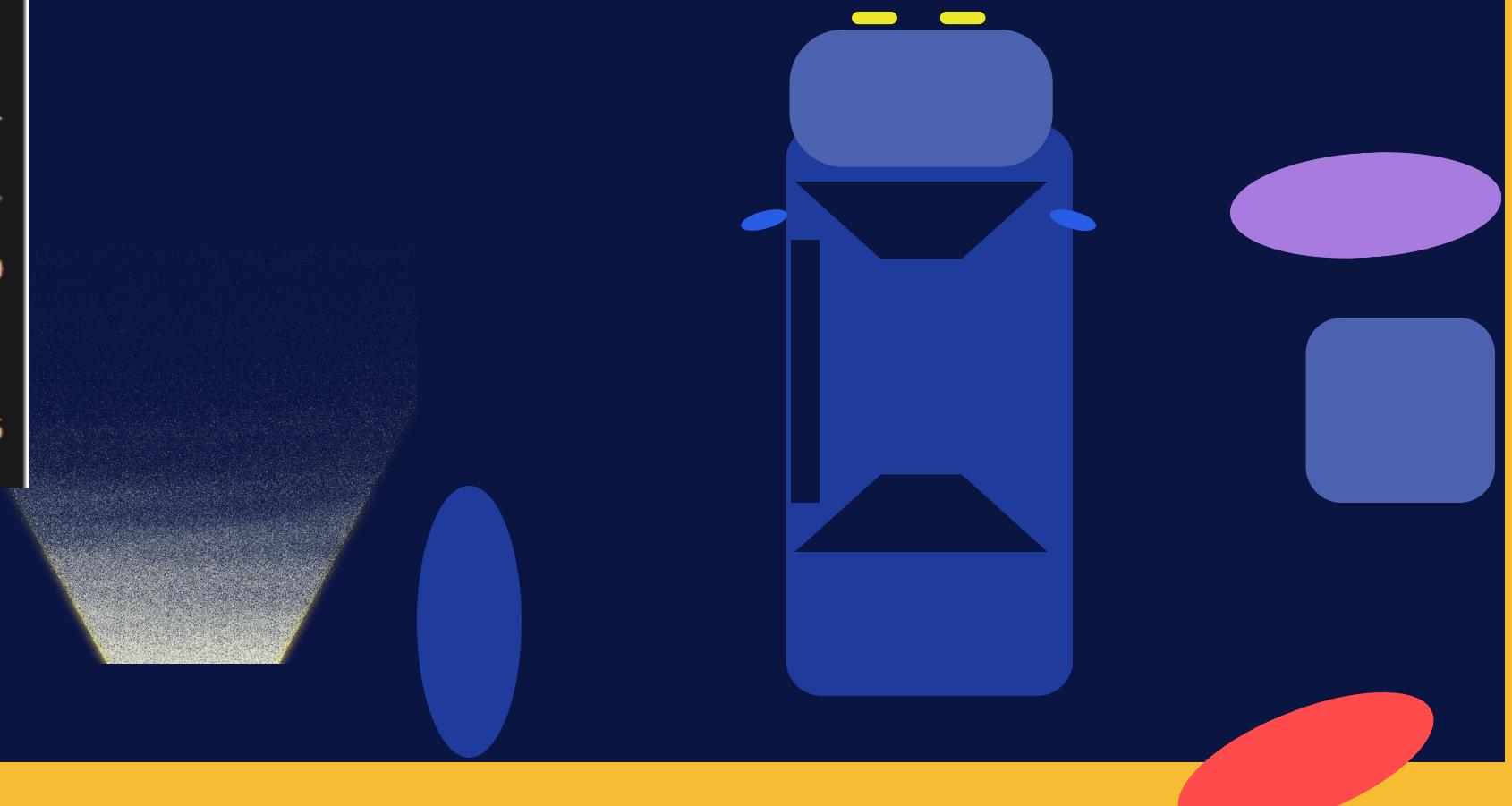


|    | Name         | License Number | Address              | Email                        | Phone No   |
|----|--------------|----------------|----------------------|------------------------------|------------|
| 1. | Erfan Syabil | 021026060451   | Puchong, Selangor    | muhderfan2610@gmail.com      | 0198525011 |
| 2. | Amirul Hani  | 020814112378   | Gombak, Kuala Lumpur | amirulhani02@gmail.com       | 0173223121 |
| 3. | Saranya Red  | 010515113278   | Ipoh, Perak          | saranyaJayarama@yahoo.com.my | 0178324320 |
| 4. | Iswary Aish  | 010715442781   | Kuantan, Pahang      | aish@email.com.my            | 0132454430 |
| 5. | Amirah       | 020813060296   | Dungun, Terengganu   | nuramierah@gmail.com         | 0179213496 |


```

Display all of the customers information

If the user enters 3 as one of their choices, it will print out a list of customers, allowing the user to view all customer details.



USER MANUAL

```
--- Car Rental System Menu ---
1. View available car and rental prices
2. Add customer details
3. View all customer details
4. Rent a car
5. View rental history
6. Exit
Enter your choice (1-6): 4

--- All Customer Information ---
-----

| Name            | License Number | Address              | Email                        | Phone No   |
|-----------------|----------------|----------------------|------------------------------|------------|
| 1. Erfan Syabil | 021026060451   | Puchong, Selangor    | muhderfan2610@gmail.com      | 0198525011 |
| 2. Amirul Hani  | 020814112378   | Gombak, Kuala Lumpur | amirulhani02@gmail.com       | 0173223121 |
| 3. Saranya Red  | 010515113278   | Ipoh, Perak          | saranyaJayarama@yahoo.com.my | 0178324320 |
| 4. Iswary Aish  | 010715442781   | Kuantan, Pahang      | aish@email.com.my            | 0132454430 |
| 5. Amira        | 020813060296   | Dungun, Terengganu   | nuramierah@gmail.com         | 0179213496 |

-----
```

Enter the index of the customer to make a rental for: 2

--- Pickup Locations ---

```
1. 123 BP 11, Puchong, Selangor, 47120
2. 456 NP 32, Kulai, Johor, 81310
3. 789 Villa, Kuantan, Pahang, 43543
4. 102 Elm St, Gombak, Kuala Lumpur, 91232
5. 879 Oak St, Ipoh, Perak, 30000
6. 321 Pine St, Jeli, Kelantan, 54321
```

Enter the index of the pickup location: 4

Rent a car

Choice 4 allows users to rent a car.

-Users must choose a customer whom they want to rent a car by inputting the index of the customer list.

-User must choose a pickup location by entering a number based on the list shown.

-Choose a car, user must enter their preferred choices by the number of the available car list. A return location must be chosen.

-User must enter their appointment date and time to pick up their car.

-Enter a number for their rental duration.

Output will be shown, informing the user their rental is successful and printing out their rental information along with the total rental cost

USER MANUAL

--- Available Car and Rental Prices ---

Type	Make	Model	Year	Rental Price
1. Compact	Perodua	Axia	(2022)	RM50.0 per day
2. Sports	Ford	Mustang	(2022)	RM300.0 per day
3. Sedan	Proton	Saga	(2022)	RM80.0 per day
4. MPV	Honda	BRV	(2022)	RM100.0 per day

Enter the car index to rent (1-4): 3

--- Return Locations ---

1. 123 BP 11, Puchong, Selangor, 47120
2. 456 NP 32, Kulai, Johor, 81310
3. 879 Villa, Kuantan, Pahang, 43543
4. 102 Elm St, Gombak, Kuala Lumpur, 91232
5. 879 Oak St, Ipoh, Perak, 30000
6. 321 Pine St, Jeli, Kelantan, 54321

Enter the index of the return location: 4

Enter appointment date (yyyy-MM-dd HH:mm:ss): 2024-02-01 16:00:00

Enter the duration of rental in days: 3

Rental Successful for: Amirul Hani

Name: Amirul Hani
License Number: 020814112378
Address: Gombak, Kuala Lumpur
Email: amirulhani02@gmail.com
Phone Number: 0173223121

Car rented: Sedan Proton Saga (2022)

Pickup date: Thu Feb 01 16:00:00 MYT 2024

Pickup location: 102 Elm St, Gombak, Kuala Lumpur, 91232

Return location: 102 Elm St, Gombak, Kuala Lumpur, 91232

Duration: 3 day(s)

Total: RM240.00

Rent a car

Choice 4 allows users to rent a car.

-Users must choose a customer whom they want to rent a car by inputting the index of the customer list.

-User must choose a pickup location by entering a number based on the list shown.

-Choose a car, user must enter their preferred choices by the number of the available car list. A return location must be chosen.

-User must enter their appointment date and time to pick up their car.

-Enter a number for their rental duration.

Output will be shown, informing the user their rental is successful and printing out their rental information along with the total rental cost

USER MANUAL

```
--- Car Rental System Menu ---
1. View available car and rental prices
2. Add customer details
3. View all customer details
4. Rent a car
5. View rental history
6. Exit
Enter your choice (1-6): 5
```

```
--- Rental History ---
```

Name: Amirul Hani
License Number: 020814112378
Address: Gombak, Kuala Lumpur
Email: amirulhani02@gmail.com
Phone Number: 0173223121

Car rented: Sedan Proton Saga (2022)
Pickup date: Thu Feb 01 16:00:00 MYT 2024
Pickup location: 102 Elm St, Gombak, Kuala Lumpur, 91232
Return location: 102 Elm St, Gombak, Kuala Lumpur, 91232
Duration: 3 day(s)

Total: RM240.00

View Rental History

User can see their rental history by pressing 5. Output will be shown containing the customer information and their rental information.

USER MANUAL

Exit

User press press 6 to exit

---- Car Rental System Menu ----

1. View available car and rental prices
2. Add customer details
3. View all customer details
4. Rent a car
5. View rental history
6. Exit

Enter your choice (1-6): 6

Exiting Rental System. Thank you!