



UML DIAGRAM OF CAR SALES MANAGEMENT

Project Description

Introduction:

The "Car Sales Management System" is a Java-based software application where customers can buy cars. The Car Sales Management System will contain also Admin and Dealer features. This application should be able to add new cars, remove any existing cars, update car details, show all car information, and make customers payments. It should also be able to add or remove any users.

Project Explanation:

What is it?

Our project is about Car Sales Management System. We made this project with some features. These are,

1. Admin
2. Customer
3. Dealer
4. Exit

In the console, if we run this project, the output will be shown.

How does this work?

- **For admin:** If we press 1 (as an admin) then there will be a login option.

There will be two options username and password. We need to fill these up.

After login as an admin, there will be 4 more options. These are “1. Manage users”, “2. Manage cars”, “3. Manage payments”, and “0. Exit”. Now if we press 1, then there will be more features like add and remove admin, dealer,

and customer, including “Show all users”, and “Exit” feature. If we press 2, then there will be features like add car, remove car, update car details, view all cars, and exit. If we press 3, then there will be features like view all payments, total balance, and exit. Now if we press 0, then we will be back on the main menu.

- **For Customer:** If we press 2 (as a customer) then there will be an option which will ask the customer to enter his ID. After entering ID, there will be some features like “1. Buy car”, “2. View all orders”, “3. Remove from order”, “4. Make payment”, “5. View payment details”, and “0. Exit”. Now if we enter 1 then there will be three more options and these are “View all cars”, “Search car”, and “Exit”. In “View all cars” option there will be all cars which are available on the store and there will be another option to order if customers want otherwise the customer can back to the previous menu by entering 0. Now in “Search car” option there will be some more options by which a customer can search car. It can be by car ID, model, brand, or year. Now if we press 2 then we can see our order list. If we press 3 then we can remove our order by entering the car ID. If we press 4 then we can see our total bill and can confirm payment by pressing 1. Now if we press 5 then we can view our payment details and lastly by pressing 0 we will be back on the main menu.

- **For Dealer:** If we press 3 (as a dealer) then there will be an option which will ask the dealer to enter his ID. After entering ID, there will be three features like “1. Add car”, “2. Remove car”, and “0. Exit”. If we press 1, then we need to enter car ID, brand, model, car type, fuel type etc. After successfully adding the car, we need to enter dealer ID. If we press 2, then we need to enter car ID for removing car. Now if we press 0, we will be back on the main menu.

- **Exit:** If we press 4, our program will be off.

Who is it for?

- **Car Dealerships:** They can use this system to manage their car sales business.
- **Customers:** Anyone looking to buy a car can use it to find and buy their dream car.

All Features:

1. **Admin:** Can login and enter the admin menu.

```
Enter as:  
1.Admin  
2.Customer  
3.Dealer  
0.Exit  
Enter- 1  
  
Enter Username: admin  
Enter Password: admin  
  
login successful  
  
Welcome to Admin menu
```

2. **Manage Users:** Can add or remove admin, dealer and customer. Also, can show all users.

```
1.Manage Users  
2.Manage Cars  
3.Manage payments  
0.Exit  
Enter- 1
```

```
1.Add Admin  
2.Remove Admin  
3.Add Dealer  
4.Remove Dealer  
5.Add Customer  
6.Remove Customer  
7.Show all Users  
0.Exit
```

3. **Manage Cars:** Can add, remove, update cars and view all cars.

```
1.Manage Users  
2.Manage Cars  
3.Manage payments  
0.Exit  
Enter- 2
```

```
1.Add Cars  
2.Remove Car  
3.Veiw All Cars  
4.Update Car Info  
0.Exit
```

4. **Manage Payments:** Can view all payments and check total balance.

```
1.Manage Users  
2.Manage Cars  
3.Manage payments  
0.Exit  
Enter- 3
```

```
1.View All Payments  
2.Total Balance  
0.Exit
```

5. **Customer:** Can enter by giving ID.

```
Enter as:  
1.Admin  
2.Customer  
3.Dealer  
0.Exit  
Enter- 2
```

```
Enter your customer ID: 11
```

```
Hello! Muqaddisa  
Welcome to our store!
```

6. **Buy Car:** Can buy car by viewing all cars or by searching.

```
1.Buy Car
2.View All Orders
3.Remove from Order
4.make payment
5.View Payment Details
0.Exit
Enter- 1
```

```
1.View All Cars
2.Search Car
0.Exit
```

7. View all Car: Display all available cars.

```
1.View All Cars
2.Search Car
0.Exit
Enter- 1

All Available cars:
Car details:
Car Id: 1
Brand: Ferrari
Model: X
Color: Red
Car Type: sports
Fuel Type: diesel
Year: 2022
Price: 10000.0
Available: true
Dealer Info: Dealer details:
Dealer Id: 101
Name: Abrar
Address: Barishal
E-mail: coolboiabrar@gmail.com
Phone No.: 017232323**
Number of Cars: 3

Car details:
Car Id: 2
Brand: Ferrari
Model: Y
Color: Blue
Car Type: sports
Fuel Type: diesel
Year: 2023
Price: 9500.0
Available: true
Dealer Info: Dealer details:
Dealer Id: 101
Name: Abrar
Address: Barishal
E-mail: coolboiabrar@gmail.com
Phone No.: 017232323**
Number of Cars: 3
```

8. Search Car: Can search car by car ID, model, brand, color, car type etc.

```
1.View All Cars  
2.Search Car  
0.Exit  
Enter- 2
```

```
Search Car by  
1.car id  
2.model  
3.brand  
4.year  
0.Exit
```

9. **View all Orders:** Can see all ordered cars.

```
1.Buy Car  
2.View All Orders  
3.Remove from Order  
4.make payment  
5.View Payment Details  
0.Exit  
Enter- 2  
All ordered cars are:
```

```
Hello! Muqaddisa  
Welcome to our store!
```

10. **Remove from Order:** Can remove any car by entering car ID.

```
1.Buy Car  
2.View All Orders  
3.Remove from Order  
4.make payment  
5.View Payment Details  
0.Exit  
Enter- 3
```

```
Enter the CarID to remove from order: 1
```

```
Succesfully removed from the OrderList
```

11. **Make Payment:** Can see total bill and make payment by press 1.

```
1.Buy Car
2.View All Orders
3.Remove from Order
4.make payment
5.View Payment Details
0.Exit
Enter- 4

Total bill: 0.0$
Press 1 to confirm payment: 1

Payment Successful
```

12. **View Payment Details:** Display the payment details of customers.

```
1.Buy Car
2.View All Orders
3.Remove from Order
4.make payment
5.View Payment Details
0.Exit
Enter- 5

payment details:
Payment:
Payment Id: 1
Payment Type: Cash
Amount: 0.0
Payment Details: Payment Successful.paid by Customer ID: 11
Payment Date: Sun Sep 17 07:28:23 BDT 2023
```

13. **Dealer:** Can enter by giving ID and can add or remove cars.

```
1.Admin  
2.Customer  
3.Dealer  
0.Exit  
Enter- 3
```

```
Enter your Dealer ID: 101
```

```
Hello! Abrar  
Welcome !
```

```
1.Add Car  
2.Remove Car  
0.Exit
```

14. **Add Car:** Can add cars by entering car ID, model, brand, color, price etc.

```
1.Add Car  
2.Remove Car  
0.Exit  
Enter- 1  
Enter CarID: 2  
Enter Brand name: name1  
Enter Model: ghsf  
Enter Color: red  
Enter Car Type: petrol  
Enter Fuel Type: jeis  
Enter Year: 3922  
Enter Price: 37283
```

```
Sucessfully added
```

15. **Remove Car:** Can remove car by entering car ID.

```
1.Add Car  
2.Remove Car  
0.Exit  
Enter- 2  
  
Enter Car ID to remove: 1  
  
Successfully removed
```

16. **Exit:** Ending the program.

```
1.Admin  
2.Customer  
3.Dealer  
0.Exit  
Enter- 0  
PS C:\Users\user\Desktop\project> █
```

Limitations:

The project is about the Car Sales Management System. We have added a few features which were “Admin login”, “Manage users”, “Manage cars”, “Manage payments”, “Add car”, “Remove car”, “Buy car”, “View all car”, “Search car”, “Remove order”, “View order”, “Make payment”, “View payment”, and “Exit”. We could add more features like “Test drive” where customer could schedule their time for driving test, “Calculate loan” which will estimate loan payments for a selected car, “Backup Data” which will backup of the system’s data, and “Restore Data” which will restore data from previous backup. But it was hard for us to do in the console especially backup and restore data.

Conclusion:

The "Car Sales System" project represents a significant technological advancement in the automotive industry. It empowers car dealerships to operate more efficiently and offers customers a more accessible and enjoyable car-buying experience. By continually adapting to changing needs and addressing potential challenges, this project has the potential to transform the way cars are bought and sold, benefiting both businesses and consumers in the process.

Source Code:

Class Admin:

```
package project;

public class Admin extends User {

    private String username;
    private String password;

    public Admin() {
    }

    public Admin(String username, String password, String name, String
address, String email, String phone) {
        super(name, address, email, phone);
        this.username = username;
        this.password = password;
    }
}
```

```

public String getUsername() {
    return username;
}

public void setUsername(String username) {
    this.username = username;
}

public String getPassword() {
    return password;
}

public void setPassword(String password) {
    this.password = password;
}

@Override
public String toString() {
    String s;
    s = "Admin details:\n" + "Name: " + getName() + "\nAddress: " +
getAddress() + "\nE-mail: " + getEmail() + "\nPhone No.: " + getPhone();
    return s;
}
}

```

Class Car:

```

package project;

public class Car {

    private int carID;
    private String brand;
    private String model;
    private String color;
    private String carType;
    private String fuelType;
    private int year;
}

```

```
private double price;
private boolean available;

Dealer dealerInfo;
Customer customerInfo;

Car() {

}

Car(int carID, String brand, String model, String color, String carType,
String fuelType, int year, double price) {
    this.carID = carID;
    this.brand = brand;
    this.model = model;
    this.color = color;
    this.carType = carType;
    this.fuelType = fuelType;
    this.year = year;
    this.price = price;
    this.available = true;
}

Car(int carID, String brand, String model, String color, String carType,
String fuelType, int year, double price, Dealer dealer) {
    this.carID = carID;
    this.brand = brand;
    this.model = model;
    this.color = color;
    this.carType = carType;
    this.fuelType = fuelType;
    this.year = year;
    this.price = price;
    this.available = true;
    this.dealerInfo = dealer;
    dealer.carList.add(new Car(carID, brand, model, color, carType,
fuelType, year, price));
}

// getter setters
public int getCarID() {
    return carID;
```

```
}

public void setCarID(int carID) {
    this.carID = carID;
}

public String getBrand() {
    return brand;
}

public void setBrand(String brand) {
    this.brand = brand;
}

public String getModel() {
    return model;
}

public void setModel(String model) {
    this.model = model;
}

public String getColor() {
    return color;
}

public void setColor(String color) {
    this.color = color;
}

public String getCarType() {
    return carType;
}

public void setCarType(String carType) {
    this.carType = carType;
}
```

```
public String getFuelType() {
    return fuelType;
}

public void setFuelType(String fuelType) {
    this.fuelType = fuelType;
}

public int getYear() {
    return year;
}

public void setYear(int year) {
    this.year = year;
}

public double getPrice() {
    return price;
}

public void setPrice(double price) {
    this.price = price;
}

public boolean isAvailable() {
    return available;
}

public void setAvailable(boolean available) {
    this.available = available;
}

public Dealer getDealerInfo() {
    return dealerInfo;
}

public void setDealerInfo(Dealer dealerInfo) {
    this.dealerInfo = dealerInfo;
}
```

```

public Customer getCustomerInfo() {
    return customerInfo;
}

public void setCustomerInfo(Customer customerInfo) {
    this.customerInfo = customerInfo;
}

public String toString1() {
    return "Car details: \n" + "Car Id: " + getCarID() + "\nBrand: " +
getBrand() + "\nModel: " + getModel() + "\nColor: " + getColor() + "\nCar
Type: " + getCarType() + "\nFuel Type: " + getFuelType() + "\nYear: " +
getYear() + "\nPrice: " + getPrice() + "\nAvailable: " + isAvailable() +
"\nDealer Info: " + getDealerInfo();
}
public String toString() {
    return "Car details: \n" + "Car Id: " + getCarID() + "\nBrand: " +
getBrand() + "\nModel: " + getModel() + "\nColor: " + getColor() + "\nCar
Type: " + getCarType() + "\nFuel Type: " + getFuelType() + "\nYear: " +
getYear() + "\nPrice: " + getPrice() + "\nAvailable: " + isAvailable();
}
}

```

Class Customer:

```

package project;

import java.util.ArrayList;

public class Customer extends User {

    private int cusID;
    private boolean hasMadeOrder = false;
    private int numberOfWorkers = 0;
    private Payment paymentdetails;
    private boolean paymentStatus = false;
    ArrayList<Car> carList = new ArrayList<>();
}

```

```
Customer() {  
  
}  
  
Customer(int cusID, String name, String address, String email, String  
phone) {  
    super(name, address, email, phone);  
    this.cusID = cusID;  
}  
  
public int getCusID() {  
    return cusID;  
}  
  
public void setCusID(int cusID) {  
    this.cusID = cusID;  
}  
  
public boolean hasMadeOrder() {  
    return hasMadeOrder;  
}  
  
public void setHasMadeOrder(boolean hasMadeOrder) {  
    this.hasMadeOrder = hasMadeOrder;  
}  
  
public int getNumberOfOrder() {  
    return numberofOrder;  
}  
  
public Payment getPaymentdetails() {  
    return paymentdetails;  
}  
  
public void setPaymentdetails(Payment paymentdetails) {  
    this.paymentdetails = paymentdetails;  
    this.paymentStatus = true;  
}
```

```
public boolean isPaymentStatus() {
    return paymentStatus;
}

public void setPaymentStatus(boolean paymentStatus) {
    this.paymentStatus = paymentStatus;
}

void buyCar(Car car) {
    carList.add(car);
    numberOfWorkOrder++;
}

void cancelOrder(Car car) {
    carList.remove(car);
    numberOfWorkOrder--;
}

void cancelAllOrders() {
    carList = null;
}

void printAllOrderedCars() {
    System.out.println("All ordered cars are:");
    for (int i = 0; i < carList.size(); i++) {
        System.out.println(carList.get(i));
        System.out.println("");
    }
}

public String toString() {
    return "Customer Details: \n" + "Customer Id: " + getCusID() +
"\nName: " + getName() + "\nAddress: " + getAddress() + "\nE-mail: " +
getEmail() + "\nPhone No.: " + getPhone() + "\nHas Made Order: " +
hasMadeOrder() + "\nNumber of Order: " + getNumberOfOrder();
}

public String toString1() {
    return "Customer Details: \n" + "Customer Id: " + getCusID() +
"\nName: " + getName() + "\nAddress: " + getAddress() + "\nE-mail: " +
getEmail() + "\nPhone No.: " + getPhone() + "\nHas Made Order: " +
```

```
hasMadeOrder() + "\nNumber of Order: " + getNumberOfOrder() + "\nCar List: " +
carList;
}
}
```

Class Dealer:

```
package project;

import java.util.*;

public class Dealer extends User {

    private int DeID;
    ArrayList<Car> carList = new ArrayList<>();
    private int numberofCars = this.carList.size();

    Dealer() {

    }

    Dealer(int DeID, String name, String address, String email, String phone)
    {
        super(name, address, email, phone);
        this.DeID = DeID;
    }

    public int getDeID() {
        return DeID;
    }

    public void setDeID(int DeID) {
        this.DeID = DeID;
    }

    public int getNumberOfCars() {
        return numberofCars;
    }
}
```

```

    }

    void addCar(Car car) {
        carList.add(car);
        numberOfCars++;
    }

    void removeCar(Car car) {
        carList.remove(car);
        numberOfCars--;
    }

    void printAllCars() {
        System.out.println("All cars by this dealer are :");
        for (int i = 0; i < carList.size(); i++) {
            System.out.println(carList.get(i));
            System.out.println("");
        }
    }

    @Override
    public String toString() {
        return "Dealer details: \n" + "Dealer Id: " + getDeID() + "\nName: " +
        getName() + "\nAddress: " + getAddress() + "\nE-mail: " + getEmail() +
        "\nPhone No.: " + getPhone() + "\nNumber of Cars: " + carList.size();
    }

    public String toString1() {
        return "Dealer details: \n" + "Dealer Id: " + getDeID() + "\nName: " +
        getName() + "\nAddress: " + getAddress() + "\nE-mail: " + getEmail() +
        "\nPhone No.: " + getPhone() + "\nNumber of Cars: " + carList.size() + "\nCar
List: " + carList;
    }
}

```

Class Payment:

```

package project;
import java.util.*;

public class Payment {
    private static int paymentID=0;

```

```
private String paymentType;
private double amount;
private String paymentDetails;
private Date paymentDate;
private Customer customerinfo;

Payment(){
    paymentID=++paymentID;
    paymentType="Cash";
    paymentDetails=" Payment Successful ";
    paymentDate=new Date();
}
Payment(double amount,Customer cus){
    this.amount=amount;
    this.paymentID=++paymentID;
    paymentType="Cash";
    customerinfo=cus;
    paymentDetails=" Payment Successful.paid by Customer ID:
"+customerinfo.getCusID();
    paymentDate=new Date();

}
Payment(String paymentType,double amount,String paymentDetails){
    this.paymentID=++paymentID;
    this.paymentType=paymentType;
    this.amount=amount;
    this.paymentDetails=paymentDetails;
    paymentDate=new Date();
}

public static int getPaymentID() {
    return paymentID;
}

public String getPaymentType() {
    return paymentType;
}

public void setPaymentType(String paymentType) {
    this.paymentType = paymentType;
}

public double getAmount() {
    return amount;
}
```

```

    }

    public void setAmount(double amount) {
        this.amount = amount;
    }

    public String getPaymentDetails() {
        return paymentDetails;
    }

    public void setPaymentDetails(String paymentDetails) {
        this.paymentDetails = paymentDetails;
    }

    public Date getPaymentDate() {
        return paymentDate;
    }

    public Customer getCustomerinfo() {
        return customerinfo;
    }

    public void setCustomerinfo(Customer customerinfo) {
        this.customerinfo = customerinfo;
    }

    @Override
    public String toString() {
        return "Payment: \n" + "Payment Id: " + getPaymentID() + "\nPayment
Type: " + getPaymentType() + "\nAmount: " + getAmount() + "\nPayment Details:
" + getPaymentDetails() + "\nPayment Date: " + getPaymentDate();
    }
}

```

Class Person:

```
package project;
```

```
interface Person {
```

```
public String getName();

public void setName(String name);

public String getAddress();

public void setAddress(String address);

public String getEmail();

public void setEmail(String email);

public String getPhone();

public void setPhone(String phone);

public String toString();
}
```

Class User:

```
package project;

import java.util.*;

public abstract class User implements Person {

    private String name;
    private String address;
    private String email;
    private String phone;
```

```
User() {  
  
}  
  
User(String name, String address, String email, String phone) {  
    this.name = name;  
    this.address = address;  
    this.email = email;  
    this.phone = phone;  
}  
  
public String getName() {  
    return name;  
}  
  
public void setName(String name) {  
    this.name = name;  
}  
  
public String getAddress() {  
    return address;  
}  
  
public void setAddress(String address) {  
    this.address = address;  
}  
  
public String getEmail() {  
    return email;  
}  
  
public void setEmail(String email) {  
    this.email = email;  
}  
  
public String getPhone() {  
    return phone;  
}
```

```

    public void setPhone(String phone) {
        this.phone = phone;
    }

    public abstract String toString();
}

```

Class Main:

```

package project;

import java.util.*;
import java.io.*;

public class Main {

    public static void main(String[] args) throws Exception {
        Scanner inp = new Scanner(System.in);
        ArrayList<Admin> AdminList = new ArrayList<>();
        ArrayList<Customer> CustomerList = new ArrayList<>();
        ArrayList<Dealer> DealerList = new ArrayList<>();
        ArrayList<Car> CarList = new ArrayList<>();
        ArrayList<Payment> PayList = new ArrayList<>();
        try {

            //test admin
            Admin admin1 = new Admin("admin", "admin", "Saif", "Banasree",
"2022-3-60-045@std.ewubd.edu", "018392289**");
            Admin admin2 = new Admin("sristy", "1234", "Sristy", "Banasree",
"2022-3-60-311@std.ewubd.edu", "0183324329**");
            Admin admin3 = new Admin("efte", "123456", "Efte", "Banasree",
"2022-3-60-042@std.ewubd.edu", "0183324289**");
            AdminList.add(admin1);
            AdminList.add(admin2);
            AdminList.add(admin3);

            //test customer
        }
    }
}

```

```

        Customer cus1 = new Customer(11, "Muqaddisa", "Narayanganj",
"2022-3-60-317@std.ewubd.edu", "01919426415");

        CustomerList.add(cus1);

        //test Dealer
        Dealer dealer1 = new Dealer(101, "Abrar", "Barishal",
"coolboiabrar@gmail.com", "017232323**");
        Dealer dealer2 = new Dealer(102, "Redwan", "gazipur",
"Homieboiredwan@gmail.com", "01342432323**");
        Dealer dealer3 = new Dealer(103, "shahmul", "kishorganj",
"chocolateboishahmul@gmail.com", "013422323**");
        Dealer dealer4 = new Dealer(104, "Ekra", "polligram",
"onlyimo@gmail.com", "013232323**");
        DealerList.add(dealer1);
        DealerList.add(dealer2);
        DealerList.add(dealer3);
        DealerList.add(dealer4);

        //test Cars
        Car testcar1 = new Car(1, "Ferrari", "X", "Red", "sports",
"diesel", 2022, 10000, dealer1);
        Car testcar2 = new Car(2, "Ferrari", "Y", "Blue", "sports",
"diesel", 2023, 9500, dealer1);
        Car testcar3 = new Car(3, "Honda ", "Z", "Black", "family_car",
"octane", 2021, 2500, dealer2);
        Car testcar4 = new Car(4, "Tesla", "X", "Black", "coupe",
"electricity", 2022, 5000, dealer2);
        Car testcar5 = new Car(5, "Toyota", "Z", "blue", "sports",
"diesel", 2023, 9500, dealer3);
        Car testcar6 = new Car(6, "McLaren ", "Y", "blue", "sports",
"diesel", 2023, 9500, dealer3);
        Car testcar7 = new Car(7, "BMW", "Y", "White", "Antique", "petrol",
", 2023, 6000, dealer4);
        Car testcar8 = new Car(8, "BMW", "Z", "White", "Limousine",
"petrol ", 2023, 9500, dealer4);
        Car testcar9 = new Car(9, "Honda", "X", "blue", "sports",
"Octane", 2023, 9500, dealer1);
        Car testcar10 = new Car(10, "Range_Rover", "Y", "Grey", "luxury",
"diesel", 2023, 8500, dealer3);
        CarList.add(testcar1);
        CarList.add(testcar2);
        CarList.add(testcar3);
        CarList.add(testcar4);
        CarList.add(testcar5);

```



```

        System.out.println("\n1.Manage Users\n2.Manage
Cars\n3.Manage payments\n0.Exit");
        System.out.print("Enter- ");
        int x1 = inp.nextInt();
        if (x1 == 0) {
            break;
        } else if (x1 == 1) {
            while (true) {
                System.out.println("\n1.Add
Admin\n2.Remove Admin\n3.Add Dealer\n4.Remove Dealer\n5.Add Customer\n6.Remove
Customer\n7.Show all Users\n0.Exit");
                System.out.print("Enter- ");
                int x2 = inp.nextInt();
                if (x2 == 0) {
                    break;
                } else if (x2 == 1) {
                    System.out.print("Enter Username :");
                    String un = inp.next();
                    System.out.print("Enter Password :");
                    String pas = inp.next();
                    System.out.print("Enter Name :");
                    String fname = inp.next();
                    System.out.print("Enter Address :");
                    String aadd = inp.next();
                    System.out.print("Enter Email :");
                    String amail = inp.next();
                    System.out.print("Enter Phone :");
                    String pn = inp.next();

                    Admin ad = new Admin(un, pas, fname,
aadd, amail, pn);
                    AdminList.add(ad);
                    System.out.println("\nSuccessfully
added\n");
                } else if (x2 == 2) {
                    System.out.print("\nEnter username of
admin to remove: ");
                    String remAd = inp.next();
                    int test = 1;
                    for (int i = 0; i < AdminList.size();
i++) {
                        if
(AdminList.get(i).getUsername().equals(remAd)) {
                            AdminList.remove(AdminList.get(i));
                            test = 0;
                        }
                    }
                    if (test == 1)
                        System.out.println("User not found");
                }
            }
        }
    }
}

```

```

System.out.println("\nSuccessfully removed\n");
                                break;
                            }
                        }
                    if (test == 1) {
                        System.out.println("\nNo Admin
found\n");
                    }
                } else if (x2 == 3) {

System.out.print("Enter Dealer ID :");
int Did = inp.nextInt();
System.out.print("Enter Name :");
String dname = inp.next();
System.out.print("Enter Address :");
String dadd = inp.next();
System.out.print("Enter Email :");
String dmail = inp.next();
System.out.print("Enter Phone :");
String pn = inp.next();

Dealer ad = new Dealer(Did, dname,
dadd, dmail, pn);
DealerList.add(ad);
System.out.println("\nSuccessfully
added\n");
} else if (x2 == 4) {
System.out.print("\nEnter Dealer ID to
remove: ");
int Did = inp.nextInt();
int test = 1;
for (int i = 0; i < DealerList.size();
i++) {
if (DealerList.get(i).getDeID() ==
Did) {

DealerList.remove(DealerList.get(i));
test = 0;
}
System.out.println("\nSuccessfully removed\n");
break;
}
if (test == 1) {
}
}

```

```

        System.out.println("\nNo Dealer
found\n");
    }
} else if (x2 == 5) {
    System.out.print("Enter Customer ID
:");
    int Cid = inp.nextInt();
    System.out.print("Enter Name :");
    String cname = inp.next();
    System.out.print("Enter Address :");
    String cadd = inp.next();
    System.out.print("Enter Email :");
    String cmail = inp.next();
    System.out.print("Enter Phone :");
    String pn = inp.next();

Customer cus = new Customer(Cid,
cname, cadd, cmail, pn);
CustomerList.add(cus);
System.out.println("\nSucessfully
added\n");
} else if (x2 == 6) {
    System.out.print("\nEnter Customer ID
to remove: ");
    int Cid = inp.nextInt();
    int test = 1;
    for (int i = 0; i <
CustomerList.size(); i++) {
        if (CustomerList.get(i).getCusID()
== Cid) {
            CustomerList.remove(CustomerList.get(i));
            test = 0;
        }
    }
    System.out.println("\nSucessfully removed\n");
    break;
}
if (test == 1) {
    System.out.println("\nNo Customer
found\n");
}
} else if (x2 == 7) {
    System.out.println("\nAll admins:
\n");
    for (int i = 0; i < AdminList.size();
i++) {

```

```

System.out.println(AdminList.get(i));
                                System.out.println("");
}
System.out.println("\nAll Dealers:
\n");
for (int i = 0; i < DealerList.size();
i++) {

System.out.println(DealerList.get(i));
                                DealerList.get(i).printAllCars();
                                System.out.println("");
}
System.out.println("\nAll Customers:
\n");
for (int i = 0; i <
CustomerList.size(); i++) {

System.out.println(CustomerList.get(i));
                                System.out.println("");
}
}

}
} else if (x1 == 2) {
    while (true) {
        System.out.println("\n1.Add Cars\n2.Remove
Car\n3.Veiw All Cars\n4.Update Car Info\n0.Exit");
        System.out.print("Enter- ");
        int x3 = inp.nextInt();
        if (x3 == 0) {
            break;
        }
        if (x3 == 1) {
            System.out.print("Enter CarID: ");
            int cid = inp.nextInt();
            System.out.print("Enter Brand name:
");

String brand = inp.next();
System.out.print("Enter Model: ");
String model = inp.next();
System.out.print("Enter Color: ");
String color = inp.next();
System.out.print("Enter Car Type: ");
String ctype = inp.next();
System.out.print("Enter Fuel Type: ");
String ftype = inp.next();

```

```

        System.out.print("Enter Year: ");
        int year = inp.nextInt();
        System.out.print("Enter Price: ");
        double price = inp.nextDouble();

        Car carob = new Car(cid, brand, model,
color, ctype, ftype, year, price);
        CarList.add(carob);
        System.out.println("\nSucessfully
added\n");

    } else if (x3 == 2) {
        System.out.print("\nEnter Car ID to
remove: ");
        int Cid = inp.nextInt();
        int test = 1;
        for (int i = 0; i < CarList.size();
i++) {
            if (CarList.get(i).getCarID() ==
Cid) {

                CarList.remove(CarList.get(i));
                test = 0;
            }
        }
        System.out.println("\nSucessfully removed\n");
        break;
    }
    if (test == 1) {
        System.out.println("\nNo Car
found\n");
    }

} else if (x3 == 3) {
    System.out.println("\nAll cars info:
");
    for (int i = 0; i < CarList.size();
i++) {

        System.out.println(CarList.get(i));
        System.out.println("");
    }
}
else if (x3 == 4) {
    System.out.println("\nUpdate Car
Price");
    System.out.print("\nEnter the Car ID
to update: ");
}

```

```

        int Cid = inp.nextInt();
        int test = 1;
        for (int i = 0; i < CarList.size();
i++) {
            if (CarList.get(i).getCarID() ==
Cid) {
                System.out.print("\nEnter the
new Price: ");
                double pr = inp.nextDouble();
                CarList.get(i).setPrice(pr);
                test = 0;
            }
            System.out.println("\nSuccessfully Updated\n");
            break;
        }
    }
    if (test == 1) {
        System.out.println("\nNo Car
found\n");
    }
}

} else if (x1 == 3) {
    while (true) {
        System.out.println("\n1.View All
Payments\n2.Total Balance\n0.Exit");
        System.out.print("Enter- ");
        int x4 = inp.nextInt();
        if (x4 == 0) {
            break;
        } else if (x4 == 1) {
            int a = Payment.getPaymentID();
            if (a != 0) {
                for (int i = 0; i <
PayList.size(); i++) {
                    System.out.println(PayList.get(i));
                    System.out.println("");
                }
            } else {
                System.out.println("\nNo payment
information found\n");
            }
        } else if (x4 == 2) {
            double totalbalance = 0;

```



```

        System.out.println("\n1.View All
Cars\n2.Search Car\n0.Exit");
        System.out.print("Enter- ");
        int x2 = inp.nextInt();
        if (x2 == 0) {
            breaker = 1;
            break;
        }
        if (x2 == 1) {
            System.out.println("\nAll
Available cars: ");
            for (int i = 0; i <
CarList.size(); i++) {
                System.out.println(CarList.get(i));
                System.out.println("");
            }
            System.out.print("\nEnter the
CarID to add to order (or enter 0 to exit): ");
            int carid = inp.nextInt();
            if (carid != 0) {
                Car car1;
                int found = 0;
                for (int i = 0; i <
CarList.size(); i++) {
                    if
(CarList.get(i).getCarID() == carid) {
                        car1 = CarList.get(i);
                        cus.buyCar(car1);

                System.out.println("\nSuccessfully Added to OrderList\n");
                        found = 1;
                        breaker = 1;
                        break;
                    }
                }
                if (found == 0) {
                    breaker = 1;
                    System.out.println("\nCar
not found\n");
                }
            }
        } else if (carid == 0) {
            breaker = 1;
        }
    } else if (x2 == 2) {

```

```

        System.out.println("\nSearch Car
by \n1.car id\n2.model\n3.brand\n4.year\n0.Exit");
        System.out.print("Enter- ");
        int x3 = inp.nextInt();
        if (x3 == 0) {
            break;
        } else if (x3 == 1) {
            int found = 0;
            System.out.print("Enter car
ID: ");
            int carid = inp.nextInt();
            for (int i = 0; i <
CarList.size(); i++) {
                if
(CarList.get(i).getCarID() == carid) {
                    System.out.println(CarList.get(i));
                    found = 1;
                }
            }
            if (found == 0) {
                breaker = 1;
                System.out.println("\nCar
not found\n");
            }
        }

    } else if (x3 == 2) {
        int found = 0;
        System.out.print("Enter car
Model: ");
        String carm = inp.next();
        for (int i = 0; i <
CarList.size(); i++) {
            if
(CarList.get(i).getModel().equals(carm)) {
                System.out.println(CarList.get(i));
                found = 1;
            }
        }
        if (found == 0) {
            breaker = 1;
            System.out.println("\nCar
not found\n");
        } else if (found == 1) {
            System.out.print("\nEnter
the CarID to add to order (or enter 0 to exit): ");

```



```

        } else if (found == 1) {
            System.out.print("\nEnter
the CarID to add to order (or enter 0 to exit): ");
            int carid = inp.nextInt();
            if (carid != 0) {
                Car car1;
                int found1 = 0;
                for (int i = 0; i <
CarList.size(); i++) {
                    if
(CarList.get(i).getCarID() == carid) {
                        car1 =
CarList.get(i);

                        cus.buyCar(car1);

                        System.out.println("\nSuccessfully Added to OrderList\n");
                        found1 = 1;
                        breaker = 1;
                        break;
                    }
                }
                if (found1 == 0) {
                    breaker = 1;
                }
            }
        }

        } else if (carid == 0) {
            breaker = 1;
        }
    }
}

} else if (x3 == 4) {
    int found = 0;
    System.out.print("Enter year:
");

    int cary = inp.nextInt();
    for (int i = 0; i <
CarList.size(); i++) {
        if
(CarList.get(i).getYear() == cary) {
            System.out.println(CarList.get(i));
            found = 1;
        }
    }
    if (found == 0) {

```

```

                    breaker = 1;
                    System.out.println("\nCar
not found\n");
                } else if (found == 1) {
                    System.out.print("\nEnter
the CarID to add to order (or enter 0 to exit): ");
                    int carid = inp.nextInt();
                    if (carid != 0) {
                        Car car1;
                        int found1 = 0;
                        for (int i = 0; i <
CarList.size(); i++) {
                            if
(CarList.get(i).getCarID() == carid) {
                                car1 =
CarList.get(i);
                                cus.buyCar(car1);

                                System.out.println("\nSuccessfully Added to OrderList\n");
                                found1 = 1;
                                breaker = 1;
                                break;
                            }
                        }
                        if (found1 == 0) {
                            breaker = 1;
                            System.out.println("\nCar not found\n");
                        }
                    } else if (carid == 0) {
                        breaker = 1;
                    }
                }
            }
        }
    }
}
} else if (x1 == 2) {
    cus.printAllOrderedCars();
} else if (x1 == 3) {
    System.out.print("\nEnter the CarID to
remove from order: ");
    int carid = inp.nextInt();
    if (carid != 0) {

```

```

        Car car1;
        int found = 0;
        for (int i = 0; i < CarList.size();
i++) {
            if (CarList.get(i).getCarID() ==
carid) {
                car1 = CarList.get(i);
                cus.cancelOrder(car1);

System.out.println("\nSuccessfully removed from the OrderList\n");
                found = 1;
                breaker = 1;
                break;
            }
        }
        if (found == 0) {
            breaker = 1;
            System.out.println("\nCar not
found\n");
        }

    } else if (carid == 0) {
        breaker = 1;
    }
} else if (x1 == 4) {
    double total = 0;
    for (int i = 0; i < cus.carList.size();
i++) {
        total = total +
cus.carList.get(i).getPrice();
    }
    System.out.println("\nTotal bill: " +
total + "$");
    System.out.print("Press 1 to confirm
payment: ");
    int c = inp.nextInt();
    if (c == 1) {
        Payment ps = new Payment(total, cus);
        cus.setPaymentdetails(ps);
        PayList.add(ps);
        System.out.println("\nPayment
Successful\n");
    } else {
        System.out.println("\nPayment
failed\n");
    }
} else if (x1 == 5) {

```

```

        System.out.println("\npayment details:");
        if (cus.isPaymentStatus()) {

System.out.println(cus.getPaymentdetails());

    } else {
        System.out.println("\nNo Payment
information found.\n");
    }

}

} else {
    System.out.println("No Customer found\n");
    break;
}

}

}

} else if (x == 3) {

int breaker = 0;
while (true && breaker == 0) {

    System.out.print("\nEnter your Dealer ID: ");
    int did = inp.nextInt();
    Dealer dealer = null;
    if (dealcheck(DealerList, did)) {

        for (int i = 0; i < DealerList.size(); i++) {
            if (DealerList.get(i).getDeID() == did) {
                dealer = DealerList.get(i);
            }
        }
        System.out.println("\nHello! " + dealer.getName()
+ "\nWelcome !");
        System.out.println("\n1.Add Car\n2.Remove
Car\n0.Exit");
        System.out.print("Enter- ");
        int x1 = inp.nextInt();
        if (x1 == 0) {

```

```

        break;
    }
    if (x1 == 1) {
        System.out.print("Enter CarID: ");
        int cid = inp.nextInt();
        System.out.print("Enter Brand name: ");
        String brand = inp.next();
        System.out.print("Enter Model: ");
        String model = inp.next();
        System.out.print("Enter Color: ");
        String color = inp.next();
        System.out.print("Enter Car Type: ");
        String ctype = inp.next();
        System.out.print("Enter Fuel Type: ");
        String ftype = inp.next();
        System.out.print("Enter Year: ");
        int year = inp.nextInt();
        System.out.print("Enter Price: ");
        double price = inp.nextDouble();

        Car carob = new Car(cid, brand, model, color,
ctype, ftype, year, price, dealer);

        CarList.add(carob);
        dealer.carList.add(carob);
        System.out.println("\nSuccessfully added\n");
    }
    if (x1 == 2) {
        System.out.print("\nEnter Car ID to remove:
");

        int Cid = inp.nextInt();
        int test = 1;
        for (int i = 0; i < dealer.carList.size();
i++) {
            if (dealer.carList.get(i).getCarID() ==
Cid) {

                dealer.carList.remove(dealer.carList.get(i));

                CustomerList.remove(dealer.carList.get(i));
                test = 0;
                System.out.println("\nSuccessfully
removed\n");
                break;
            }
        }
    }
}

```



```

        }
    } catch (Exception e) {
        System.out.println(e);
    }
    File file4 = new File(path + "/All Admins.txt");
    try (PrintWriter w = new PrintWriter(file4)) {
        for (int i = 0; i < AdminList.size(); i++) {
            w.println(AdminList.get(i));
            w.println("");
            w.println("");
        }
    } catch (Exception e) {
        System.out.println(e);
    }
    File file3 = new File(path + "/All Payments.txt");
    try (PrintWriter w = new PrintWriter(file3)) {

        int a = Payment.getPaymentID();
        if (a != 0) {
            for (int i = 0; i < PayList.size(); i++) {
                w.println(PayList.get(i));
                w.println("");
                w.println("");
            }
        } else {
            w.println("\nNo payment information found\n");
        }

        double totalbalance = 0;
        for (int i = 0; i < PayList.size(); i++) {
            totalbalance = totalbalance + PayList.get(i).getAmount();

        }
        w.println("Total balance of the Company is:" + totalbalance +
        "$");

    } catch (Exception e) {
        System.out.println(e);
    }
}

```

```
}

    public static boolean logincheck(ArrayList<Admin> adminList, String un,
String pass) {
        for (int i = 0; i < adminList.size(); i++) {
            if (adminList.get(i).getUsername().equals(un)) {
                if (adminList.get(i).getPassword().equals(pass)) {
                    return true;
                }
            }
        }
        return false;
    }

    public static boolean cuscheck(ArrayList<Customer> cusList, int id) {
        for (int i = 0; i < cusList.size(); i++) {
            if (cusList.get(i).getCusID() == id) {
                return true;
            }
        }
        return false;
    }

    public static boolean dealcheck(ArrayList<Dealer> dList, int id) {
        for (int i = 0; i < dList.size(); i++) {
            if (dList.get(i).getDeID() == id) {
                return true;
            }
        }
        return false;
    }
}
```