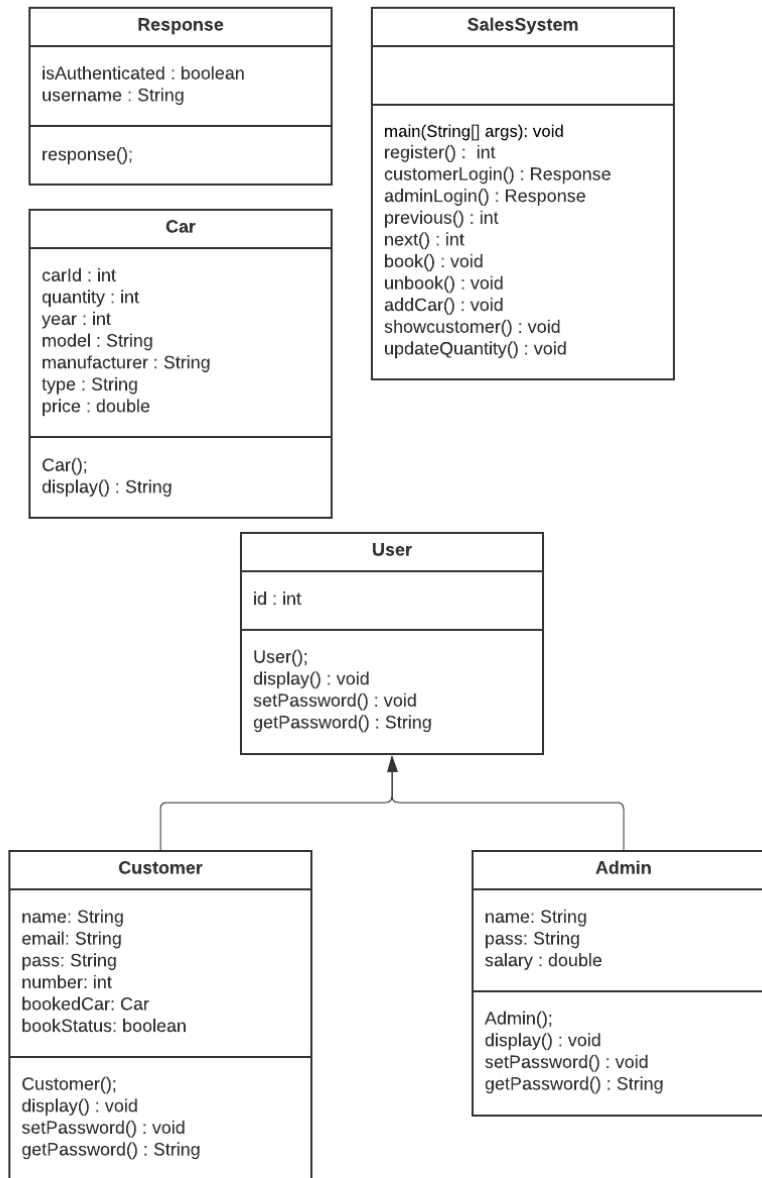


Project Topic:Car Sales System

Details:

Project Completed Using JOptionPane

1. Design Part: It will show the class diagrams



2. Implementation Part: The complete code:

```
public interface CarInfo {
    String display();
}

public class Car implements CarInfo {
    int carId, quantity, year;
    String model, manufacturer, type;
    double price;
    Car(){}
    Car(int carId, String manufacturer, int year, String model, String type, double price, int
quantity){
        super();// Constructor
        this.carId=carId;
        this.model=model;
        this.year=year;
        this.manufacturer=manufacturer;
        this.type=type;
        this.price=price;
        this.quantity=quantity;
    }
    public String display() {
        return ("Id: "+this.carId+"\nManufacturer: "
+this.manufacturer+"\nYear: "+this.year+"\nModel: "+this.model+"\nType: "+this.type
+"\nPrice: "+this.price+"\nCars in Stock: "+this.quantity);
    }
}
```

```

public abstract class User {
    int id;
    User(){}
    User(int id){ //Constructor
        this.id=id;
    }
    public abstract void display();
    public void setPassword(String password) {
    }
    public abstract String getPassword();
}

```

```

import java.util.*;
import javax.swing.*;
public class SalesSystem {
    public static void main(String args[]) {
        ArrayList<Car> vehicle=new ArrayList<>();
        vehicle.add(new Car(23,"Mercedes",2022,"Maybach","Sedan",285950,10));
        vehicle.add(new Car(24,"BMW",2022,"E3","Sedan",3980019,6));
        vehicle.add(new Car(25,"Lamborghini",2022,"Urus","SUV",6000000,0));
        vehicle.add(new Car(25,"Koenigsegg",2020,"Agera R","Sports",2000000,2));
        vehicle.add(new Car(25,"Pagani",2020,"Huayra","Sports",1400000,5));

        ArrayList<Customer> customer=new ArrayList<>();
        ArrayList<Admin> admin=new ArrayList<>();
        admin.add(new Admin("IDExist",23,"ggnore",9000000));
        admin.add(new Admin("Urusai",24,"ggez",1900203));
        admin.add(new Admin("BVD",29,"mfrahman",99999999));

        JOptionPane.showMessageDialog(null, "Welcome");
        int choice=0;
        String[] firstOptions = {"Login","Register","Exit"};
        while(true) {

```

```
choice = JOptionPane.showOptionDialog(null, "Select One", "Car Sales",  
    2, 1, null, firstOptions, firstOptions[0]); // showing options  
  
switch(choice) {  
case 0: // login  
String[] loginOptions = {"Customer", "Admin"}; //Choose login option  
between Customer or Admin  
String loginInput="";  
loginInput = (String) JOptionPane.showInputDialog(null, "Select  
One", "Car Sales",  
JOptionPane.QUESTION_MESSAGE,null,loginOptions,loginOptions[0]);  
if(loginInput=="Customer") { // Customer Login  
Response clogin = customerLogin(customer);  
if (clogin.isAuthenticated) {  
JOptionPane.showMessageDialog(null, "Login  
Successful!",null,JOptionPane.INFORMATION_MESSAGE);  
String[] customerOptions = {"View","Browse","Logout"}; //MENU  
while(true) {  
int customerInput =JOptionPane.showOptionDialog(null, "Select:  
","Logged in as: "+clogin.username ,  
JOptionPane.DEFAULT_OPTION,1, null,  
customerOptions, null); // showing menu options  
switch(customerInput) {  
case 0: // view case  
Customer found= new Customer();  
for(Customer user:customer) {  
if(user.name.equals(clogin.username)) {  
found=user;  
break;  
}  
}  
String[] viewOptions = {"Menu","Unbook"};  
if(found.bookedCar.model!=null) }
```

```

        while(true) {
            int viewInput =
OptionPane.showOptionDialog(null, "Your info"+
            "\nCar Model Chosen: "+found.bookedCar.model+
            "\nCar Type: "+found.bookedCar.type+
            "\nPrice to Pay: "+found.bookedCar.price,
            "Logged in as: "+clogin.username,
OptionPane.DEFAULT_OPTION, 1, null,
            viewOptions, null);
            switch(viewInput) {
                case 0:
                    break;
                case 1:

unbook(vehicle,found.bookedCar.model,clogin.username,customer);
                    break;

            } //switch for view break
            if(viewInput==0) { // out of while
                break;
            }
        }
    }
    else {
        JOptionPane.showMessageDialog(null, "You haven't
booked a Car.Nothing to View",
            "",
OptionPane.ERROR_MESSAGE);
    }
    break; //view case break;
    case 1: // Browse case
        String[] browseOptions =
{"Menu", "Previous", "Next", "Book"};

```

```

        int counter=0;
        while(true) {
            int browseInput = JOptionPane.showOptionDialog(null,
vehicle.get(counter).display(),
                                "Logged in as: "+clogin.username,
JOptionPane.DEFAULT_OPTION, 1, null,
                                browseOptions, null);
            switch(browseInput) { // browse switch case starts
                case 0:
                    //menu;
                    break;
                case 1:
                    counter = previous(counter);
                    break;
                case 2:
                    counter=next(counter,vehicle.size());
                    break;
                case 3:
                    book(vehicle,counter,clogin.username,customer);
                    break;
            }
            if(browseInput==0) { // out of browse
                break;
            } // breaking while
        } // while not back to menu
        break; // Browse case ends
    case 2: // Logout Case
        break;
} //switch for view, browse,logout ends
if(customerInput==2) {
    break;
}
}
}

```

```

        } // check done for authentication
        else {
            JOptionPane.showMessageDialog(null, "Wrong Password!\nPlease
try again", "", JOptionPane.ERROR_MESSAGE);
        } // Customer case ends
    }
    else if(loginInput=="Admin") {
        Response alogin = adminLogin(admin);
        if (alogin.isAuthenticated) {
            String[] adminOptions = {"Browse", "Add
Car", "Customers", "Profile", "Logout"}; // ADMIN MENU
            while(true) {
                int adminInput = JOptionPane.showOptionDialog(null,
"Select: ", "Logged in as: "
+alogin.username, JOptionPane.DEFAULT_OPTION, 1, null,
adminOptions, null);

                switch(adminInput) {
                    case 0: // Browse Case
                        String[] browseChoices =
{"Menu", "Previous", "Next", "Update Quantity"};
                        int counter=0;
                        while(true) {
                            int browseChoose = JOptionPane.showOptionDialog(null,
vehicle.get(counter).display(),
                                "Logged in as: "+alogin.username,
JOptionPane.DEFAULT_OPTION, 1, null,
                                browseChoices, null);
                            switch(browseChoose) { // browse switch case starts
                                case 0:
                                    break;
                                case 1:
                                    counter = previous(counter);
                                    break;

```



```

        case 2:
            counter=next(counter,vehicle.size());
            break;
        case 3:
            int quan=0;
            try {
                quan=
Integer.parseInt(JOptionPane.showInputDialog("Enter New Quantity"));
            }
            catch(Exception e) {
                JOptionPane.showMessageDialog(null,
                    "Quantity Should Be
Integer",null,
JOptionPane.WARNING_MESSAGE);
            }
            updateQuantity(counter,quan,vehicle);
            break;
    }
    if(browseChoose==0) {
        break;
    }
}
break; // browse breaks
case 1: // Add car case
    addCar(vehicle);
    break;
case 2: // See Customers
    showCustomer(customer);
    break;
case 3:
    for(Admin employee:admin) {
        if(employee.name.equals(alogin.username)) {

```

```

                                employee.display();
                                }
                                }
                                break;
                                case 4: // Admin Logout
                                break;
                                } // Admin Switch ends
                                if (adminInput==4) {
                                break;
                                } // goes back to Menu
                                }
                                } // Admin Verification
                                else {
                                JOptionPane.showMessageDialog(null,"Wrong Password!\nPlease
try again","",JOptionPane.ERROR_MESSAGE);
                                }
                                } // admin case ends
                                else if(loginInput==null) {
                                continue;
                                }
                                break; // case login break
                                case 1: //register
                                register(customer);
                                break;
                                case 2: //exit
                                System.exit(0);
                                break;
                                default:
                                System.exit(0);
                                break;
                                }
                                } // main while ends

```

```

} // main method ends

public static int register(ArrayList<Customer> customer) {
    JTextField name=new JTextField();
    JTextField email=new JTextField();
    JTextField number=new JTextField();
    JTextField id=new JTextField();
    JTextField pass=new JTextField();
    Object[] fields= {
        "Enter Name:",name,
        "Give email:",email,
        "Enter ID:",id,
        "Give a phone number:",number,
        "Enter a Password:",pass
    };
    int registerCheck=1;
    try {
        registerCheck= JOptionPane.showConfirmDialog(null,fields,"Registraion",
            JOptionPane.OK_CANCEL_OPTION);
        if (registerCheck ==2 ){
            return registerCheck;
        }
        if (name.getText().equals("") || email.getText().equals("") || id.getText().equals("") ||
            number.getText().equals("") || pass.getText().equals("")) {
            JOptionPane.showMessageDialog(null, "No Field can be Empty!",null,
                JOptionPane.ERROR_MESSAGE);
            return 0;
        }
        customer.add(new Customer(name.getText(),email.getText(),Integer.parseInt(id.getText()),
            Integer.parseInt(number.getText())));
        customer.get(customer.size()-1).setPassword(pass.getText());
        customer.get(customer.size()-1).display();
    }
    catch(NumberFormatException e) {

```

```

        JOptionPane.showMessageDialog(null, "Phone Number and Id should be Integers",
            null, JOptionPane.WARNING_MESSAGE);
    }
    return registerCheck;
}

    public static int addCar(ArrayList<Car> vehicle) {
        JTextField model= new JTextField();
        JTextField carId= new JTextField();
        JTextField manufacturer= new JTextField();
        JTextField year= new JTextField();
        JTextField type= new JTextField();
        JTextField price= new JTextField();
        JTextField quantity= new JTextField();
        Object[] fields= {
            "Car ID:",carId,
            "Manufacturer:",manufacturer,
            "Car Release Year:",year,
            "Enter Car Model:",model,
            "Car Type:",type,
            "Car Price:",price,
            "Cars In Stock:",quantity
        };
        int addCheck=1;
        try {
            addCheck=JOptionPane.showConfirmDialog(null,fields,"Adding Car",
                JOptionPane.OK_CANCEL_OPTION);
            if (addCheck ==2){
                return addCheck;
            }
        }
        if (carId.getText().equals("") || manufacturer.getText().equals("") || year.getText().equals("") ||
            model.getText().equals("") || type.getText().equals("") || price.getText().equals("") ||
            quantity.getText().equals("")) {

```

```

JOptionPane.showMessageDialog(null, "No Field can be
Empty!",null,JOptionPane.ERROR_MESSAGE);
return 0;
}

        vehicle.add(new Car(Integer.parseInt(carId.getText()),model.getText(),
                                Integer.parseInt(year.getText()), manufacturer.getText(),
                                type.getText(),Double.parseDouble(price.getText()),
                                Integer.parseInt(quantity.getText())));
JOptionPane.showMessageDialog(null, vehicle.get(vehicle.size()-1).display());
    }
    catch(NumberFormatException e) {
        JOptionPane.showMessageDialog(null,
            "CarID,Quantity and Price cannot be Characters Or
Empty",null,
            JOptionPane.WARNING_MESSAGE);
    }
    return addCheck;
}

public static Response customerLogin(ArrayList<Customer> people) {
    boolean isAuthenticated = false;
    JTextField userField = new JTextField(); //new field for username input
    JPasswordField passField = new JPasswordField(); //new field for password input
    JOptionPane.showOptionDialog(null, new Object[] {"Enter name and password",
userField, passField},
        "Login", JOptionPane.OK_CANCEL_OPTION,
JOptionPane.QUESTION_MESSAGE, null, null, null);
    String passText = new String(passField.getPassword());
    for(Customer users:people) {
        if(users.name.equals(userField.getText()) &&
            users.getPassword().equals(passText)) {
            isAuthenticated=true;
            return new Response(isAuthenticated,userField.getText());
        }
    }
}

```

```

    }

    return new Response(isAuthenticated,userField.getText());
}

public static Response adminLogin(ArrayList<Admin> admin) {
    boolean isAuthenticated = false;

    JTextField userField = new JTextField(); //new field for admin input
    JPasswordField passField = new JPasswordField(); //new field for admin password input
    JOptionPane.showOptionDialog(null, new Object[] {"Enter name and password",
userField, passField},

        "Login", JOptionPane.OK_CANCEL_OPTION,
JOptionPane.QUESTION_MESSAGE, null, null, null);

    String passText = new String(passField.getPassword());
    for(Admin users:admin) {
        if(users.name.equals(userField.getText()) &&
            users.getPassword().equals(passText)) {
            isAuthenticated=true;
            return new Response(isAuthenticated,userField.getText());
        }
    }

    return new Response(isAuthenticated,userField.getText());
}

public static int previous(int counter) {
    if (counter==0) {
        return counter;
    }else {
        counter--;
    }

    return counter;
}
}

```

```

public static int next(int counter,int size){
    if (counter==size-1) {
        return counter;
    }
    else {
        counter++;
    }

    return counter;
}

public static void updateQuantity(int num,int quan,ArrayList<Car>vehicle) {
    vehicle.get(num).quantity+=quan;
}

public static void book(ArrayList<Car> vehicle,int counter,String uname,
    ArrayList<Customer> customer) {
    boolean add=false;
    for(Customer user:customer) {
        if(user.name.equals(uname)) {
            if(!user.bookStatus) { // if needed
                if(vehicle.get(counter).quantity>=1) {
                    user.bookedCar.carId = vehicle.get(counter).carId;
                    user.bookedCar.manufacturer =
vehicle.get(counter).manufacturer;

                    user.bookedCar.year = vehicle.get(counter).year;
                    user.bookedCar.model =
vehicle.get(counter).model;

                    user.bookedCar.type = vehicle.get(counter).type;
                    user.bookedCar.price = vehicle.get(counter).price;
                    vehicle.get(counter).quantity--;
                    user.bookStatus=true;
                    JOptionPane.showMessageDialog(null,"Car is
Booked",

```

```

        "" ,
JOptionPane.INFORMATION_MESSAGE);

        add=true;
    }

    else { //no cars of this model available
        JOptionPane.showMessageDialog(null,"This Car is
Not in Stock",

        "" ,
JOptionPane.INFORMATION_MESSAGE);
    }

    }

    else { // if bookStatus true prints error message
        JOptionPane.showMessageDialog(null,"Sorry,You've Already
Booked A Car",

        "" ,JOptionPane.ERROR_MESSAGE);
    }

    }

    if(add) {
        break;
    }

    }

}

public static void unbook(ArrayList<Car> vehicle,String carModel,String uname,
    ArrayList<Customer> customer) {
    boolean remove=false;
    for(Customer user:customer) {
        if(user.name.equals(uname)) {
            if(user.bookStatus) {
                for(Car c: vehicle) {
                    if(c.model.equals(carModel)) {
                        c.quantity++;
                        break;

```



```

        }

        }

        user.bookedCar.model="";
        user.bookedCar.type="";
        user.bookedCar.price=0;
        user.bookStatus=false;
        JOptionPane.showMessageDialog(null,"Car is Unbooked",
            "", JOptionPane.INFORMATION_MESSAGE);
        remove=true;
    }
    else {
        JOptionPane.showMessageDialog(null,"You haven't booked
a Car. Cannot Unbook",
            "", JOptionPane.ERROR_MESSAGE);
    }
}

if(remove) {
}

}

}

}

public static void showCustomer(ArrayList<Customer> customer) {
String bookedUser="";
for (Customer user:customer) {
if (user.bookStatus==true) {
bookedUser =bookedUser + user.name+" "+user.number+" "+user.bookedCar.manufacturer+
    " "+user.bookedCar.model+
    " "+user.bookedCar.price+"\n";
}
}

if (bookedUser.isEmpty()) {
JOptionPane.showMessageDialog(null,"No cars have been booked","",
JOptionPane.DEFAULT_OPTION);

```

```
}else {  
JOptionPane.showMessageDialog(null, bookedUser,null, JOptionPane.DEFAULT_OPTION);  
}  
}  
}
```

```
public class Response {  
    boolean isAuthenticated=false;  
    String username;  
  
    Response(boolean isAuthenticated, String username){  
        this.isAuthenticated=isAuthenticated;  
        this.username=username;  
    }  
}
```

```
import javax.swing.JOptionPane;  
public class Customer extends User{  
    private String password;  
    String name,email;  
    int number;  
    Car bookedCar=new Car();  
    boolean bookStatus=false;  
  
    Customer(String name,String email,int id,int number){ //Constructor  
        super(id);  
        this.name=name;  
        this.email=email;  
        this.id=id;  
        this.number=number;  
    }  
}
```

```

    Customer(){
        public void setPassword(String password) {
            this.password=password;
        }
        public String getPassword() {
            return this.password;
        }
        public void display() {
            JOptionPane.showMessageDialog(null,"Name of customer: "+name+"\n"+"Your
email: "+email+"\n"
                                +"Your ID: "+id+"\n"+"Your phone number: "+number+"\n"+
                                "Your new password:
"+password,"Info",JOptionPane.PLAIN_MESSAGE);
        }
    }
}

```

```

import javax.swing.JOptionPane;
public class Admin extends User{
    String name;
    private String password;
    double salary;
    Admin(String name,int id,String password,double salary){ //Constructor
        super(id);
        this.name=name;
        this.password=password;
        this.salary=salary;
    }
    public void setPassword(String password) {
        this.password=password;
    }
    public String getPassword() {

```

```

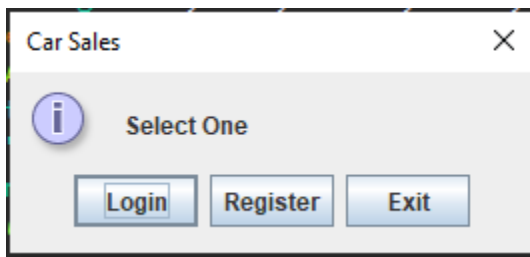
        return this.password;
    }

    public void display() {
        JOptionPane.showMessageDialog(null, "Your Name: "+name+"\n"+"Your Salary: "
+salary, "Info", JOptionPane.PLAIN_MESSAGE);
    }
}

```

3. Result: Screenshots of the inputs and outputs:

Start Menu




Customer Registration

Info ×

Name of customer: JamesBond
Your email: jamesbond900@email.com
Your ID: 7
Your phone number: 171282123
Your new password: forgetit


Car Sales ×

 Select One

Customer
Customer
Admin


Customer Logging in

Login ×

 Enter name and password

JamesBond
.....

Logged in as: JamesBond ×

 Select:

Customer Browsing Cars

Logged in as: JamesBond



Id: 23

Manufacturer: Mercedes

Year: 2022

Model: Maybach

Type: Sedan

Price: 285950.0

Cars in Stock: 10

Menu

Previous

Next

Book

Booking Car

Logged in as: JamesBond



Id: 24

Manufacturer: BMW

Year: 2022

Model: E3

Type: Sedan

Price: 3980019.0

Cars in Stock: 6

Menu

Previous

Next

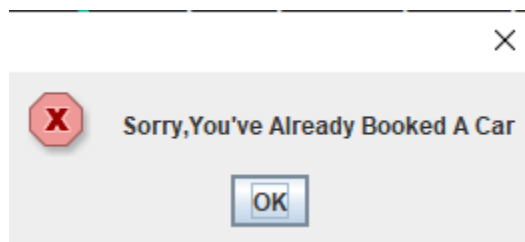
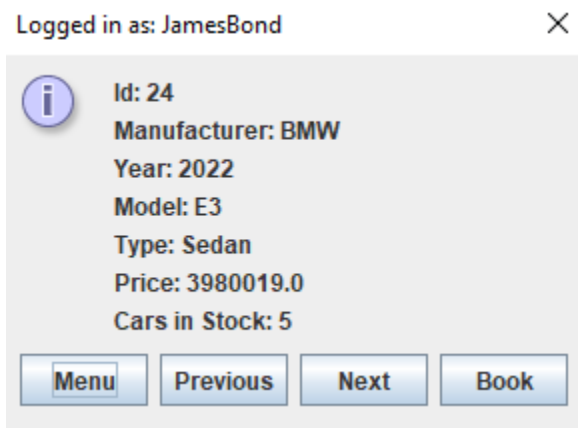
Book



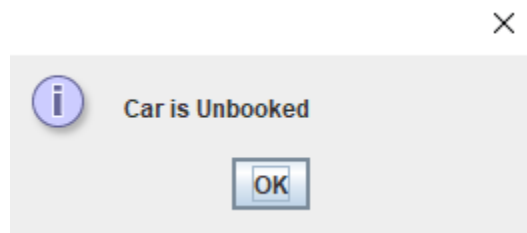
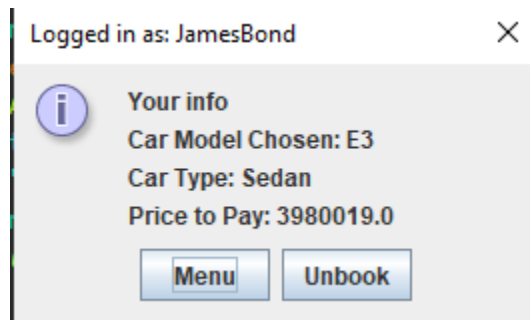
Car is Booked

OK

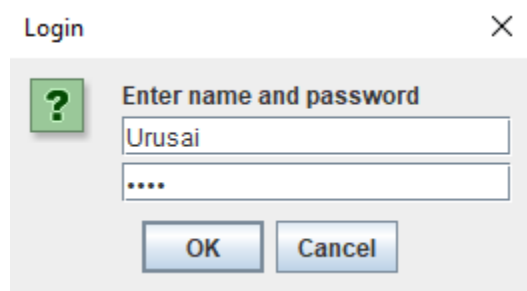
If Book Again



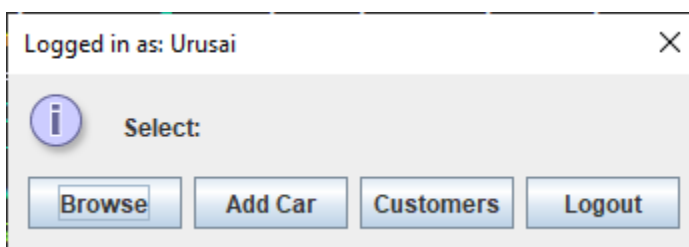
Unbook Car



Logging in as Admin:




Admin chooses From Admin Menu



Admin Browsing Cars

Logged in as: IDExist ×


 **Id: 23**
Manufacturer: Mercedes
Year: 2022
Model: Maybach
Type: Sedan
Price: 285950.0
Cars in Stock: 10

Admin Increasing Stock

Logged in as: IDExist ×

 **Id: 25**
Manufacturer: Lamborghini
Year: 2022
Model: Urus
Type: SUV
Price: 6000000.0
Cars in Stock: 0

Input ×

 **Enter New Quantity**

Logged in as: IDExist



Id: 25

Manufacturer: Lamborghini

Year: 2022

Model: Urus

Type: SUV

Price: 6000000.0

Cars in Stock: 5

Menu

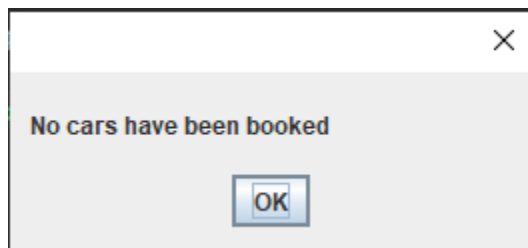
Previous

Next

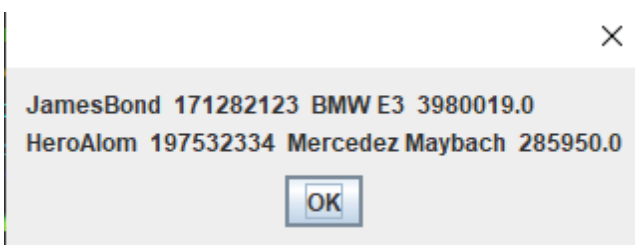
Update Quantity

Viewing Customers

No Customers




Yes Customers



Admin Adding Car

Adding Cars ×

 **Car ID:**

Manufacturer:

Car Release Year:


Enter Car Model:

Car Type:

Car Price:


Cars In Stock:

Message ×

 **Id: 892**
Manufacturer: Koenigsegg
Year: 2020
Model: Gemerah
Type: Sports Salon
Price: 1700000.0
Cars in Stock: 2

Customer Viewing New Car

Logged in as: JamesBond ×




Id: 892
Manufacturer: Koenigsegg
Year: 2020
Model: Gemerah
Type: Sports Salon
Price: 1700000.0
Cars in Stock: 2

Exception Handling

Wrong Input

Registraion ×



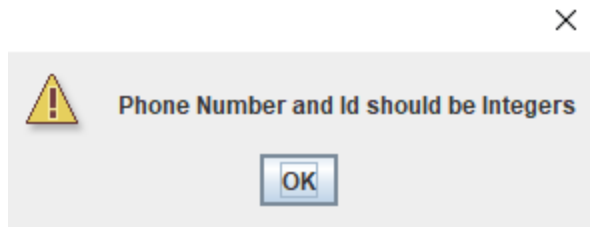
Enter Name:

Give email:

Enter ID:

Give a phone number:

Enter a Password:



Admin adding car with blank fields

Adding Car

?

Car ID:

Manufacturer:

Car Release Year:

Enter Car Model:

Car Type:

Car Price:


Cars In Stock:

OK Cancel



Adding Cars with Wrong Format As Inputs

Adding Car ×

 **Car ID:**

Manufacturer:

Car Release Year:


Enter Car Model:

Car Type:

Car Price:

Cars In Stock:

×

 CarID,Quantity and Price cannot be Characters Or Empty


Admin wrong input in update quantity

Logged in as: IDExist ×




Id: 25
Manufacturer: Lamborghini
Year: 2022
Model: Urus
Type: SUV
Price: 6000000.0
Cars in Stock: 0

Input ×



Enter New Quantity

×



Quantity Should Be Integer