GROUP ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

CT038-3.5-2-OODJ

OBJECT-ORIENTED DEVELOPMENT WITH JAVA

HAND OUT DATE: 19th SEPTEMBER 2022

HAND IN DATE: 9th DECEMBER 2022

WEIGHTAGE: 50%

INSTRUCTION TO CANDIDATES:

1 Submit your assignment at the administrative counter

2 Students are advised to underpin their answers with the use of reference (cited using Harvard Name System of Referencing)

3 Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld

4 Cases of plagiarism will be penalized

5 The assignment should be bound in an appropriate style (comb bound or stapled).

6 Where the assignment should be submitted in both hardcopy and softcopy, the softcopy of the written assignment and source code (where appropriate) should be on a CD in an envelope/ CD cover and attached to the hardcopy.

7 You must obtain 50% overall to pass this module.

Topic: Car Rental System

Name	TP Number
Loh Wan Ning	TP065926
Tey Xin Ying	TP066247

Table of Contents

1.0 Introduction

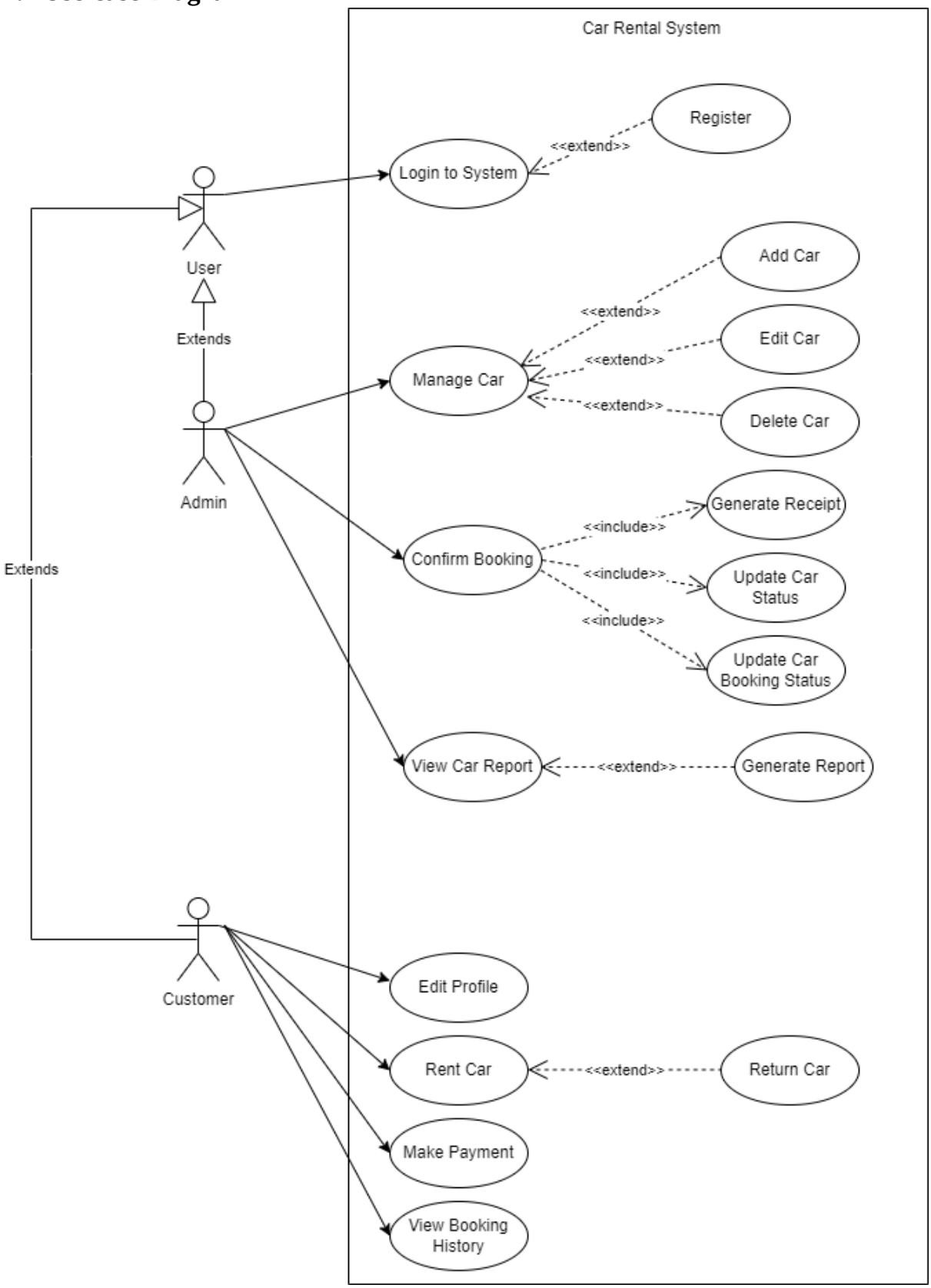
Java programming language is the most secure programming language among other programming languages. It is easy to implement, read, write, compile, and debug. (*Advantages of Java*, 2020) Therefore, Java Programming language will be used throughout this assignment. There are total of two roles in this assignment which is admin and customer. Car Rental System allows admin and customers to handle all information required to rent a car. The advantage of this system is all process and transaction can be done remotely and it is 7 days and 24 hours availability. In this assignment, we will design and implement a system and the target users are admin and customer. Admin can add, edit car information, and handle customer's booking through online. Besides, customer can register, update, rent and return car and check booking history details from the system.

Assumption

1. Assume that this system is only used in Malaysia.
2. Assume the admin of this system sharing one username and password.
3. Assume all the transactions in this system are in MYR.
4. Customer ID, Order ID and Receipt ID will automatically give by the system increment to avoid duplication.
5. Admin do not have the authorization to change the profile of customer and admin.
6. Customer only able to change the profile of themselves.
7. Every new booking and new payment must be confirmed by admin, then only customer can proceed to the next step.
8. Sales report is generated based on the car plate.
9. Assume that rental overdue must be paid offline.

2.0 UML Diagrams

2.1 Use-case Diagram



2.1.1 Use Case Specification

Login

Use Case	Login to System
Brief Description	Allow users to login to the system
Actors	User
Preconditions	Input username and password, pressing the “login” button
Main Flow	a) User type in the username and password then press on the login button. b) System checks the username and password in the User text file. c) If the file exists, system will then check the role of the user, if the role is “Admin”, the system will display admin page. However, if the role is “Customer” the system will then display customer page.
Alternate Flow	a) If system does not find matched data, the system will prompt wrong password. b) Users need to check the input username and password. c) Users need to register to the system before login.

Register

Use Case	Register to the system
Brief Description	Allow users to register to the system
Actors	User
Preconditions	Input the Username, Password, Name, Gender, Contact Number, Email Address, Address
Main Flow	a) User input personal details including username, password, name, gender, contact number, email address and address. b) The system will then record the details and write in the User text file. c) The details of the user will then exist in the system and the user able to login to the system by the username and password.
Alternate Flow	a) If user left the text field blank, the system will prompt a message to ask the user input all field. b) The system will not accept alphabet in contact text field.

Admin

Use Case	Manage Car
Brief Description	This use case allows admin to manage a car information
Actors	Admin
Preconditions	Admin login and select the add/edit car menu option
Main Flow	a) The use case begins when admin select the option to manage car information. b) The system displays the function in the add/edit car menu

Use Case	Add Car
Brief Description	This use case allows admin to add a new car into the system
Actors	Admin
Preconditions	Admin input the new car information into the system.
Main Flow	a) The use case begins when admin input the new car details: car plate, car model, price etc in the system. b) Click the "Add" button to add the new car to the system. c) "New Car Added" message will be shown.
Alternative Flow	a i) If the price is not entered in integers, the system will display message "Please enter integers! a ii) If the text field is left empty, the system will display message "Please Enter Full Details"

Use Case	Edit Car
Brief Description	This use case allows admin to edit existing car information
Actors	Admin
Preconditions	Admin select the row that wants to be edit
Main Flow	a) The use case begins when admin select the row where is going to be edited b) Admin modify the new details: car model, price etc using text field. c) Click the Edit button to edit the car information. d) The table will display the newest car information.
Alternative Flow	a i) Car details will not be displayed in text fields as no row is selected b i) The existing car information will remain the same if is not modified by the admin.

Use Case	Delete Car
Brief Description	This use case allows admin to delete existing car record
Actors	Admin
Preconditions	Admin select the row that wants to be deleted
Main Flow	a) The use case begins when the admin selects a row to delete the car record. b) Click the Delete button to remove the car record. c) The table will display the newest car record.
Alternative Flow	a i) Car details will not be displayed in text fields as no row is selected

Use Case	Confirm Booking
Brief Description	This use case allows admin to confirm the customer booking
Actors	Admin
Preconditions	Admin select the Customer Booking Confirmation menu
Main Flow	a) The use case begins when admin select the Customer Booking Confirmation button on the admin menu. b) The system then displays the options for admin. c) The admin chooses the booking to be confirmed.

Use Case	Generate Receipt
Brief Description	This use case allows admin to generate a receipt id for every confirmed booking.
Actors	Admin
Preconditions	Admin click the Send button and confirms the customer booking.
Main Flow	a) The use case begins when admin confirmed the customer booking. b) Receipt ID is then auto incremented when new booking is confirmed. a) The receipt details will be stored in text file.
Alternative Flow	b i) Receipt ID will not increment as no booking is confirmed.

Use Case	Update Booking Status
Brief Description	This use case allows admin to update booking status
Actors	Admin
Preconditions	Admin confirm the booking that needs to be updated
Main Flow	a) The use case begins when admin search for Customer ID b) Admin then select the order made by the customer. c) The order details will be displayed in the system. d) Admin then confirm a pending customer booking. e) The status of the booking will be modified according to its status.
Alternative Flow	a i) A message “CustomerID not found” will be displayed if the customer does not exist in the system. d i) Status will not be modified if the booking is confirmed.

Use Case	View Car Report
Brief Description	This use case allows admin to generate car report
Actors	Admin
Preconditions	Admin select the Generate Report options in the admin menu.
Main Flow	a) The use case begins when admin select the Generate Report button on the admin menu. b) The system then displays the options for admin. c) The admin chooses the car plate for viewing reports.

Use Case	Generate Car Report
Brief Description	This use case allows admin to generate a specific car report
Actors	Admin
Preconditions	Admin select the car plate and date to be generated for the report.
Main Flow	a) The use case begins when admin select the car plate number. b) Admin then select the date range to be calculated. c) The Confirm button is clicked. d) The sales of the select car plate will be display in the system. e) Admin then click the Export button to export the report to pdf format.
Alternative Flow	d i) No sales details is displayed as no orders for the car is made within the date range. e i) If no pdf is required, admin view the sales report by just looking at the system.

Customer

Use Case	Edit Profile
Brief Description	Allow customer to change the personal profile details
Actors	Customer
Preconditions	Customer change the password, contact number, email address and address
Main Flow	a) Customer change personal information such as password, contact number, email address and address. b) The changed button is clicked, if the text field is empty, the system will then prompt the user to fill in the text field.
Alternate Flow	a) Customer can just ignore if no personal details need to be edit.

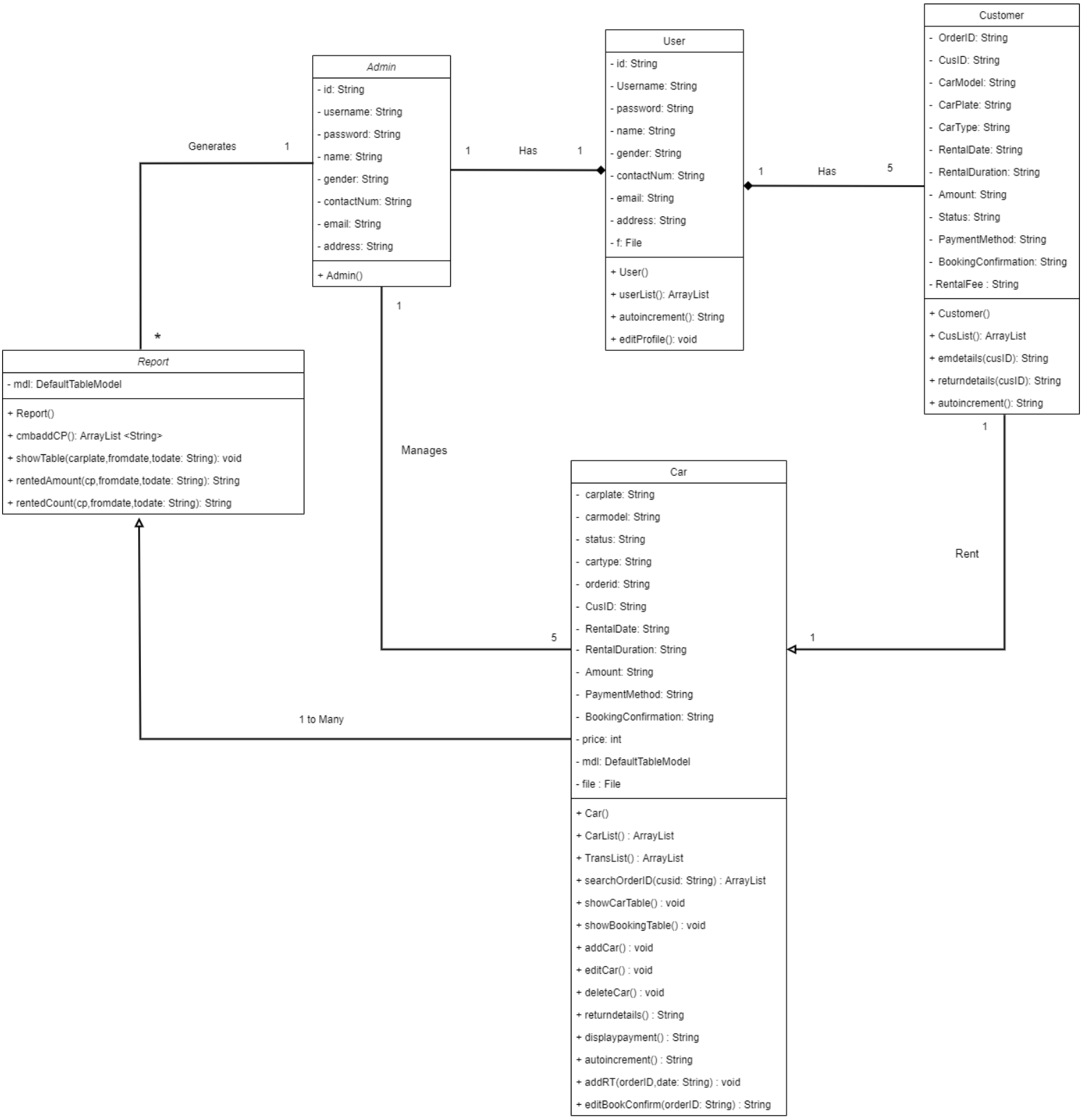
Use Case	Car Booking
Brief Description	Allow customer to book car
Actors	Customer
Preconditions	Customer able to choose and book the car in the system
Main Flow	a) Customer can choose the car in combo box and insert the duration. b) The button search is clicked to check the car status. c) If the car status is available, customer need to select the particular row to book the car by pressing the button submit.
Alternate Flow	a) If customer did not select the row in JTable, the system will prompt a message to ask user to select. b) If no car show in JTable, the customer should choose another car.

Use Case	Make Payment
Brief Description	Allow customer to make payment
Actors	Customer
Preconditions	Customer need to pay before renting a car
Main Flow	a) Customer type in the order ID, the system will show the total of the payment. b) Customer need to pay for that amount.
Alternate Flow	a) Customer will not able to rent the car if payment is unsuccessful.

Use Case	Return Car
Brief Description	Allow customer return the car after renting
Actors	Customer
Preconditions	Customer click on the return button to return the car
Main Flow	a) Customer select the car rented then click on the button return. b) The system will then change the car status from occupied to available.
Alternate Flow	a) If customer did not return the car, the status of the car will always in occupied and other customer will not able to rent that car.

Use Case	View Booking History
Brief Description	Allow customer to view the booking history
Actors	Customer
Preconditions	Customer able to view and check the previous booking history
Main Flow	a) Customer can type in the Customer ID to check the history.
Alternate Flow	a) If wrong user ID typed, the system will show blank result.

2.2 Class Diagram



3.0 Description and justification of Code with OOP Concepts

There is a total of five main object-oriented concepts in Java which is Class/ Object, Inheritance, Polymorphism, Abstraction and Encapsulation. Object-oriented is faster and easier to be execute and it is the method that work without jeopardising security.

3.1 Class/ Object

The core concepts if object-oriented programming is classes and objects. In object-oriented programming, objects are made up of both data and methods. A class is a collection of items with similar characteristics. It serves as a model or blueprint from which things can be made. It cannot be bodily. (*Object in Java | Class in Java - Javatpoint*, n.d.) The declaration always has the same name as the java file and the keyword "class", the name, modifiers, and body should also be included.

```
public class User {

    private static String id;
    private static String Username;
    private static String password;
    private static String name;
    private static String gender;
    private static String contactNum;
    private static String email;
    private static String address;
```

Figure 3.1.1

Built-in String function in Java. ID, username, contact number, email and address is unique for each customer. The others are counted as state objects.

```
public User(String id, String u, String p, String n, String g, String c, String e, String a) {
    this.id = id;
    Username = u;
    password = p;
    name = n;
    gender = g;
    contactNum = c;
    email = e;
    address = a;

}
```

Figure 3.1.2

Object is also known as member of the java class. Each object consists of a state, a behaviour and an identity. Fields variables hold an object's state whereas methods functions show the object's behaviour. The identity provides a unique name to the object and enables object to interact together. Runtime objects are produces from templates which also referred to as classes. (Techopedia, 2022)

```
public static ArrayList userList() throws FileNotFoundException, IOException {
    File f = new File( pathname: "\\C:\\Users\\txy20\\Desktop\\uni\\Degree Year 2 (Sem1)\\Object Oriented Development with Java\\Assignment\\CarRentalSystem\\src\\ca
    BufferedReader br = new BufferedReader(new FileReader( file: f));
    ArrayList<String> userlist = new ArrayList<String>();
    String lineread = null;
    while ((lineread = br.readLine()) != null) {
        userlist.add( e: lineread);
    }
    br.close();
    return userlist;
}
```

Figure 3.1.3

```
public static String autoincrement() throws FileNotFoundException, IOException {
    String newid = "";
    ArrayList<String> userlist = new ArrayList<>();
    userlist = User.userList();

    String lastid = userlist.get(userlist.size() - 1);
    String[] iddata = lastid.strip().split( regex: ",");
    String oldid = iddata[0];
    String id = oldid.substring( beginIndex: 3, endIndex: 7);
    int oldintid = Integer.parseInt( s: id);
    int nextid = oldintid + 1;
    String strnextid = Integer.toString( i: nextid);
```

Figure 3.1.4

User defined classes can also use to declare an abject. Figure above shows the value of object and able to be called and obtained through the interaction of objects and class. Userlist is created in User class and able to be called in Customer booking class. The details of the userlist can use in other class.

3.2 Inheritance

Inheritance allows one class to inherit the characteristics of other class like methods and fields. (GeeksforGeeks, 2022) Inheritance is a usability code, the fields and methods can be reused and used in other class. Superclass also known as parent class that contains the features inherited while subclass only inherit the method and fields in superclass. However, subclass cannot work if superclass disabled. The keyword “extends” need to be used.

Figure 3.1.5

Figure 3.1.6

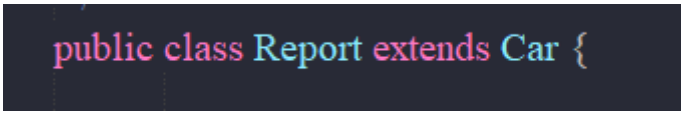


Figure 3.1.7

The keyword “extends” is used to inherit the class. The figure above shows the Report class inherit Car class.

3.3 Encapsulation

Encapsulation is more flexibility and easy to used. It can hide bundle of data into single unit and stops the code get access by unwanted data or from outside class. All the variables can be encapsulated by declaring the variable with the keyword “private”.

```
public class User {

    private static String id;
    private static String Username;
    private static String password;
    private static String name;
    private static String gender;
    private static String contactNum;
    private static String email;
    private static String address;
```

```
public String getId() {
    return id;
}

public void setId(String id) {
    this.id = id;
}

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;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}
```

```
public String getGender() {
    return gender;
}

public void setGender(String gender) {
    this.gender = gender;
}

public String getContactNum() {
    return contactNum;
}

public void setContactNum(String contactNum) {
    this.contactNum = contactNum;
}

public String getEmail() {
    return email;
}

public void setEmail(String email) {
    this.email = email;
}

public String getAddress() {
    return address;
}

public void setAddress(String address) {
    this.address = address;
}
```

The getter and setter method are used to access the value in private variable. The set method is used to set the value while get method used to return the variable value. The setter method allows variables in read-only status whereas the getter method allows write-only status.

3.4 Polymorphism

```
addMouseMotionListener(new MouseMotionAdapter() {
    @Override
    public void mouseMoved(MouseEvent me) {
        if(checkMouseOver( mouse: me.getPoint())) {
            if(cursorChange == true) {
                setCursor(new Cursor( type: Cursor.HAND_CURSOR));
            }
        }
    }
```

Polymorphisms apply in classes to enable the implementation of the interfaces that use in particular class. Overloading means lots of functions with same name but different parameters that can be overloaded. The keyword "@Override" indicates the appliance of concept.

3.5 Static

Static used to declare the class variable or method that belongs to the class.

```
public class MainPage extends javax.swing.JFrame {

    private static String username;
    private static String password;
    private static String id;
```

Static usually created and used inside the class to declare variable or method. Static can always access by the private members and method of other class. Static is used to remove restriction on member class. (JanbaskTraining,2018)

3.6 File Handling and Exception

```
private void btnSubmitActionPerformed(java.awt.event.ActionEvent evt) {
    if (!txtUsername.getText().isEmpty() && !txtPassword.getText().isEmpty() && !txtName.getText().isEmpty() &&
            !txtContact.getText().isEmpty() && !txtEmail.getText().isEmpty() && !txtAddress.getText().isEmpty()) {
        try {
            File f = new File(
                    "C:\\Users\\txy20\\Desktop\\uni\\Degree Year 2 (Sem1)\\Object Oriented Development with Java\\Assignment\\CarRentalSystem"
                            + "\\src\\carrentalsystem\\User.txt");
            if (!f.exists()) {
                f.createNewFile();
            }
```

```
public static ArrayList CusBookList() throws FileNotFoundException, IOException {
    File f = new File( pathname: "C:\\APU\\Year 2 Sem 1\\Objecct Oriented Java\\CarRentalSystem (2)\\CarRentalSystem\\src\\carrentalsystem\\Cusbooking.txt");
    BufferedReader br = new BufferedReader(new FileReader( file: f));
    ArrayList<String> cusbooklist = new ArrayList<String>();
    String lineread = null;
    while ((lineread = br.readLine()) != null) {
        cusbooklist.add( e: lineread);
    }
    br.close();
    return cusbooklist;
}

public static ArrayList CusList() throws FileNotFoundException, IOException {
    File f = new File( pathname: "C:\\APU\\Year 2 Sem 1\\Objecct Oriented Java\\CarRentalSystem (2)\\CarRentalSystem\\src\\carrentalsystem\\User.txt");
    BufferedReader br = new BufferedReader(new FileReader( file: f));
    ArrayList<String> cuslist = new ArrayList<String>();
    String lineread = null;
    while ((lineread = br.readLine()) != null) {
        cuslist.add( e: lineread);
    }
    br.close();
    return cuslist;
```

File handling methods is important to ensure the program does not crash with other text file. The createNewFile() method has been import in the system to ensure the continuity of the execution of the program. The system will generate a new file to write data if the text file does not exist. Otherwise, the system will write the data in existing file.

4.0 Input/Output of The Program

Login Page

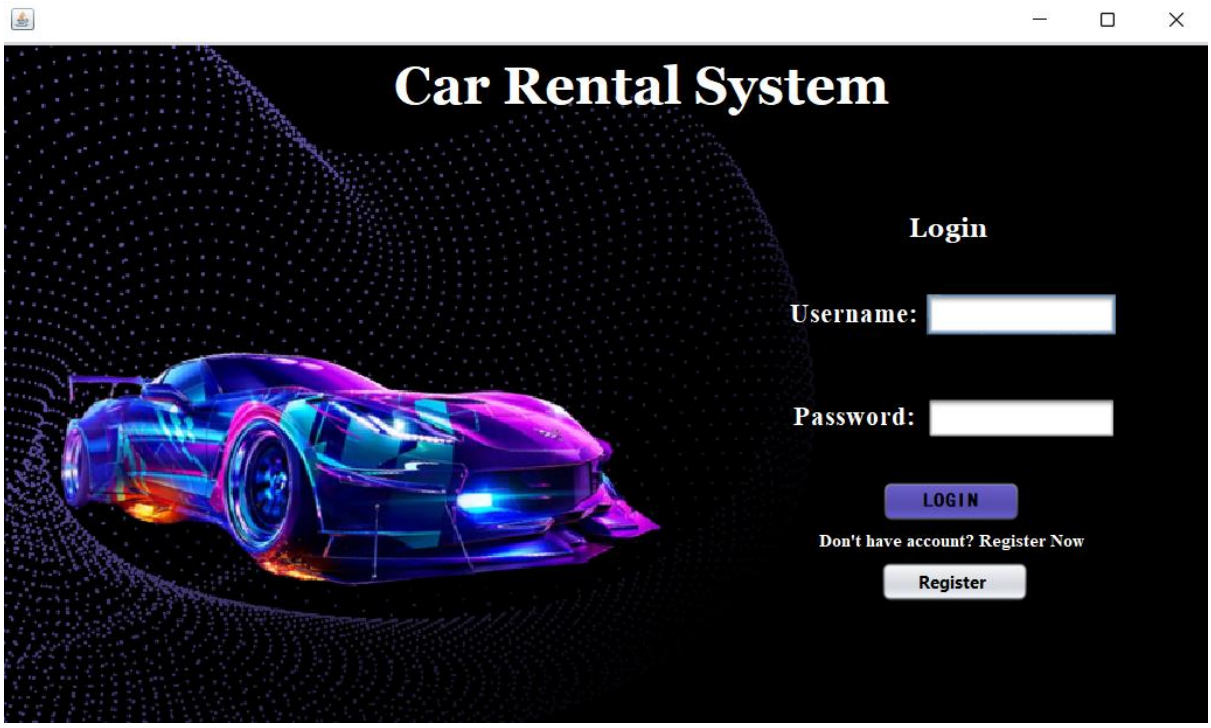


Figure 4.1.1

Figure 4.1.1 shows the login page of the car rental system and user need to type in the username and password to enter the system.

Register

Figure 4.1.2

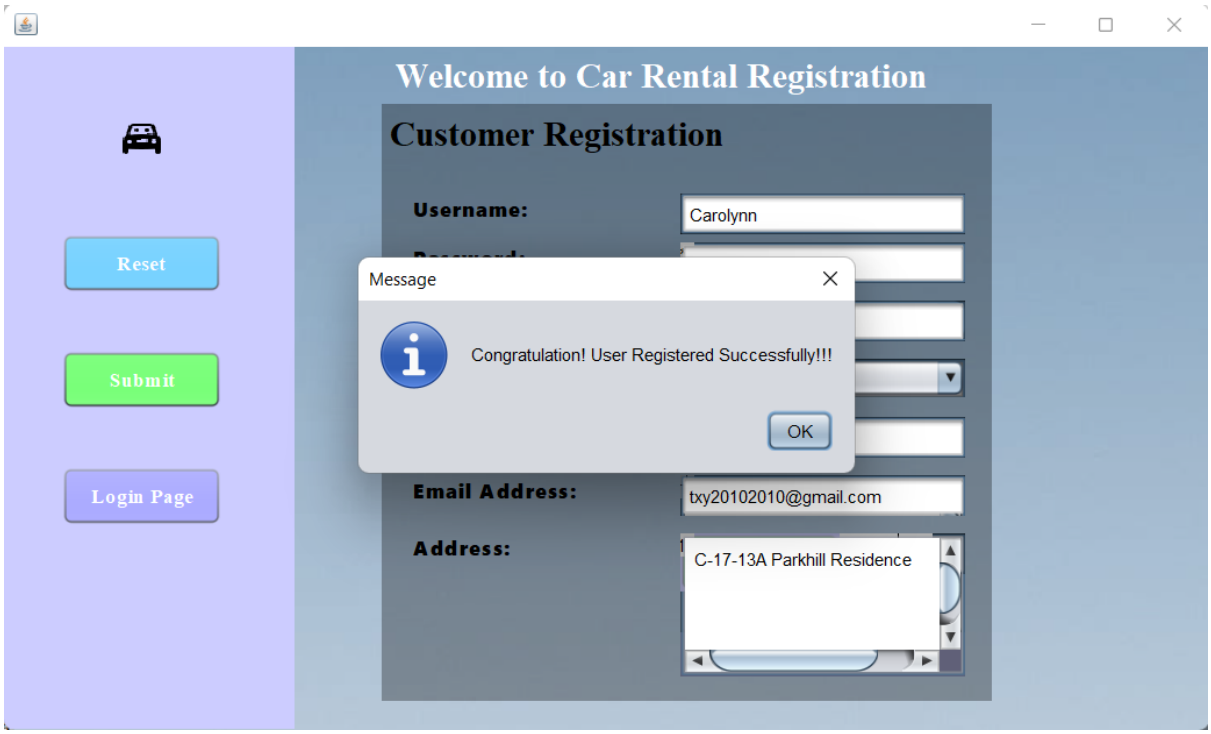


Figure 4.1.3

If user do not have an account. Click the register button to register an account by filling personal details. If the text field is empty, the system will prompt a message to ask user to fill in the details. When all details have been filled, a message show that the user has successfully registered.

Customer

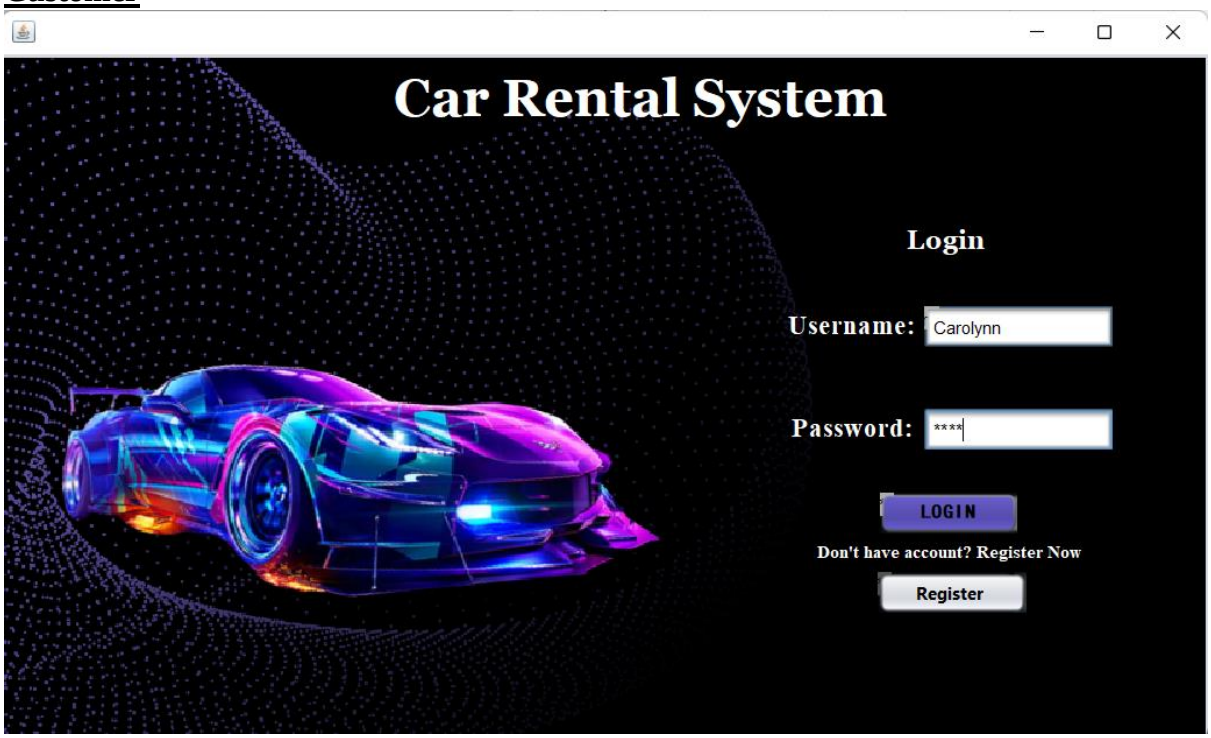


Figure 4.1.4

User can now login to the system as customer by entering the username and password.

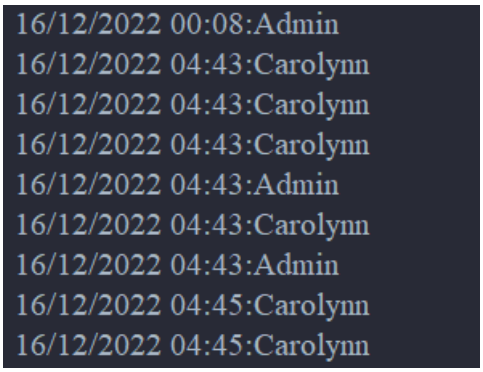


Figure 4.1.5

System login record for authentication and authorisation process. The record will be marked down and show in text field.

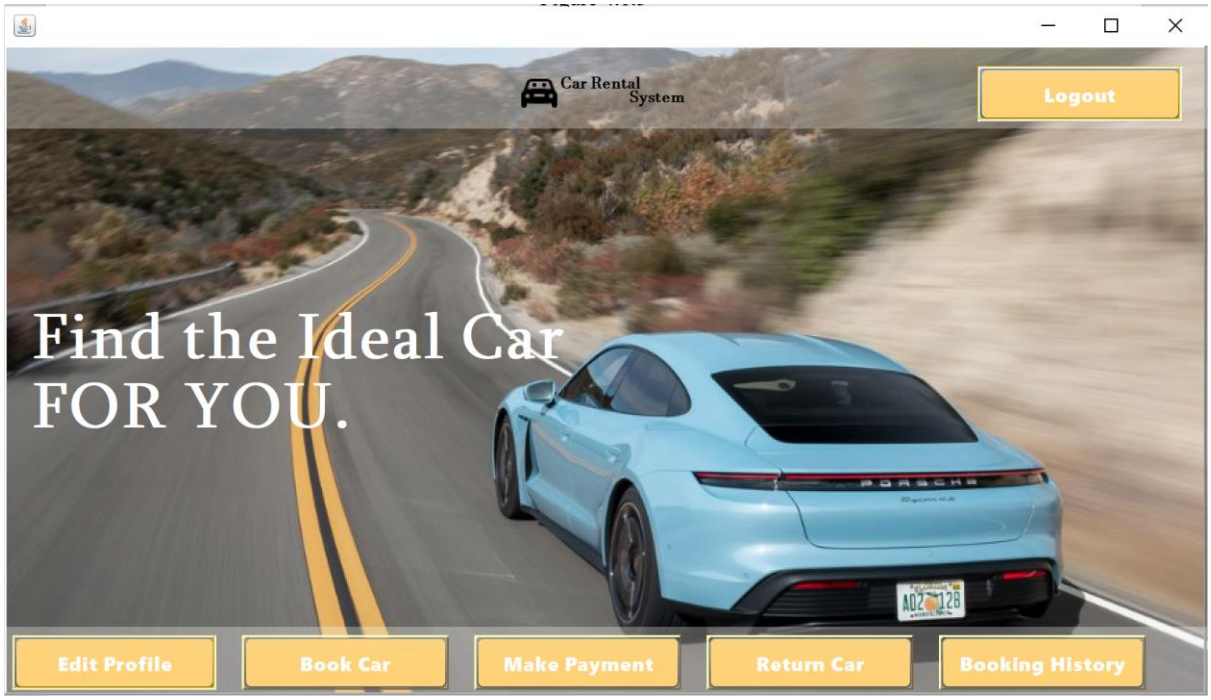


Figure 4.1.6

The main page of the car rental system has been shown in Figure 4.16.

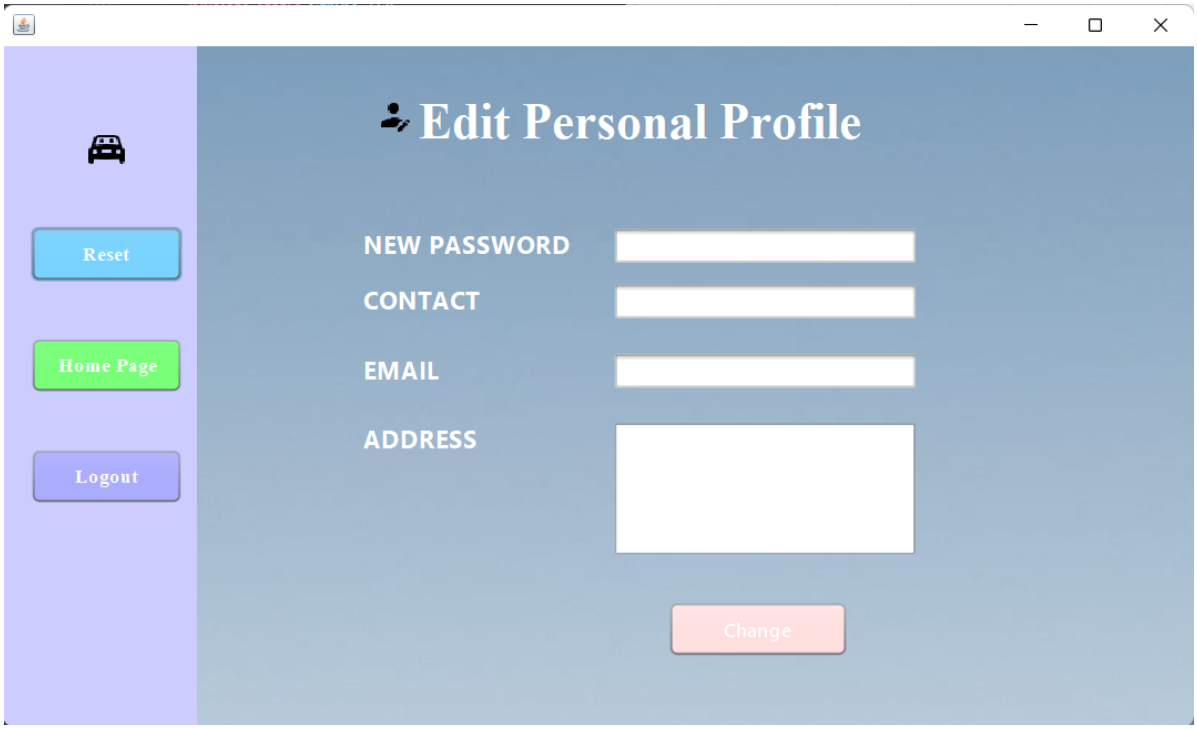


Figure 4.1.7

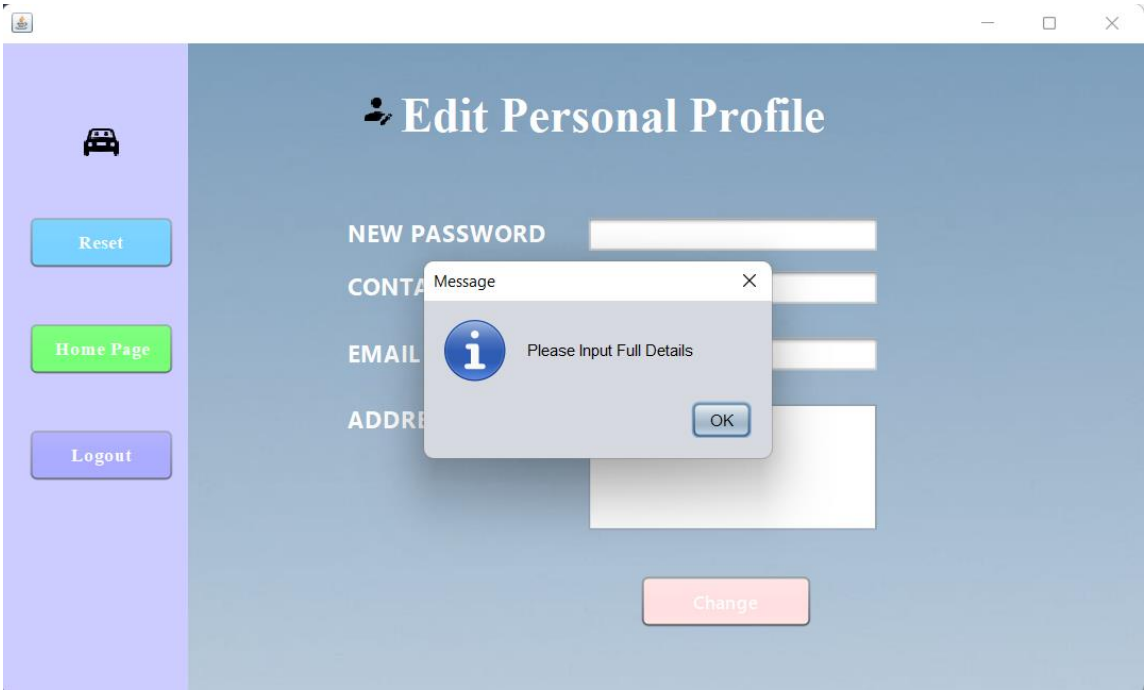


Figure 4.1.8

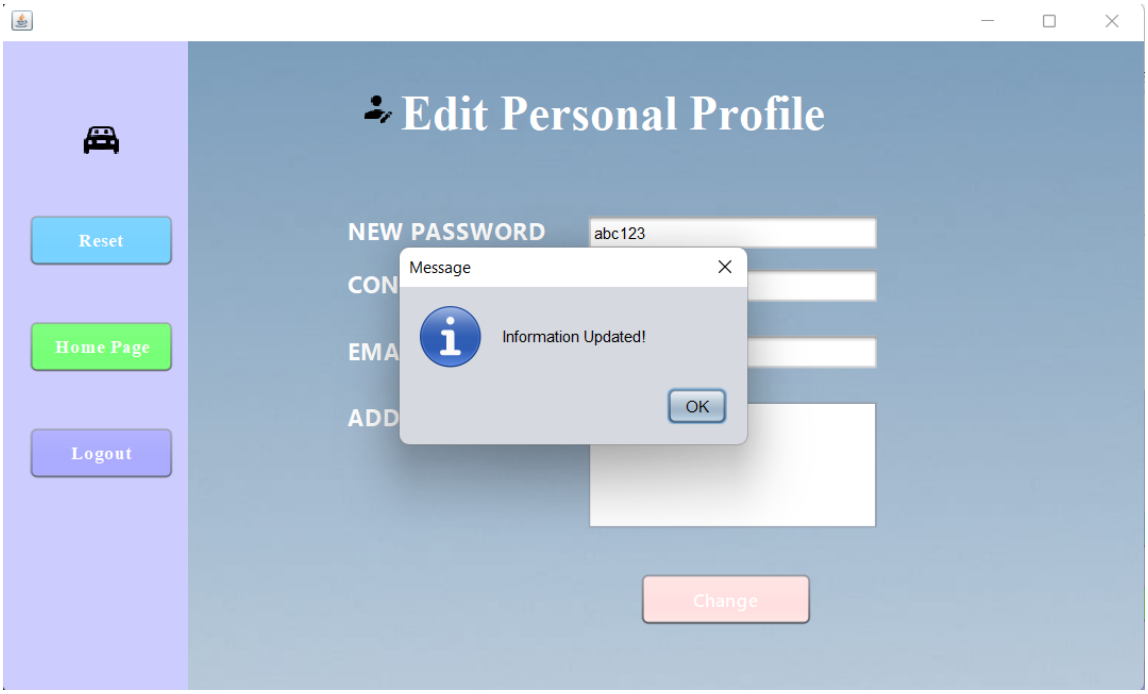


Figure 4.1.9

Figure 4.1.7, Figure 4.1.8 and Figure 4.1.9 shows that customer able to edit personal profile such as password, contact number, email and address in the system. The text field of contact is set to only accept numbers to prevent customer type in alphabet. If one of the text fields is empty, the system will prompt a message and customer required to fill in all details before update. When all information has been updated, the system will prompt a message and all value is already stored in particular text field.

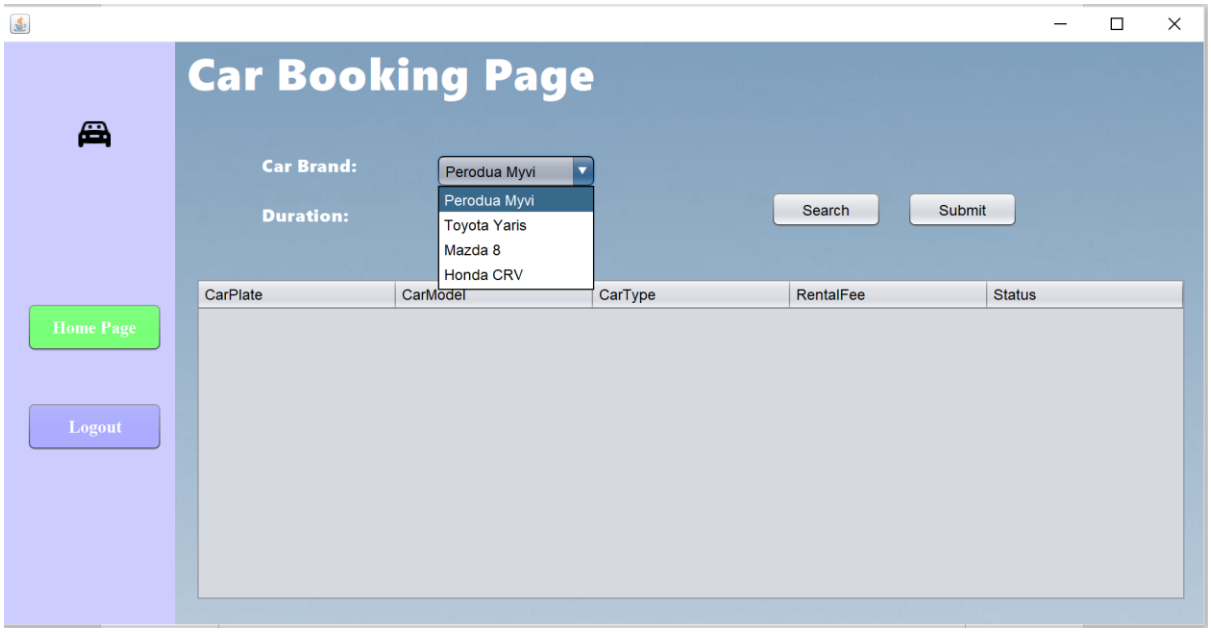


Figure 4.1.10

When customer enter the car booking page, they need to choose the car brand and input the rental duration in the system.

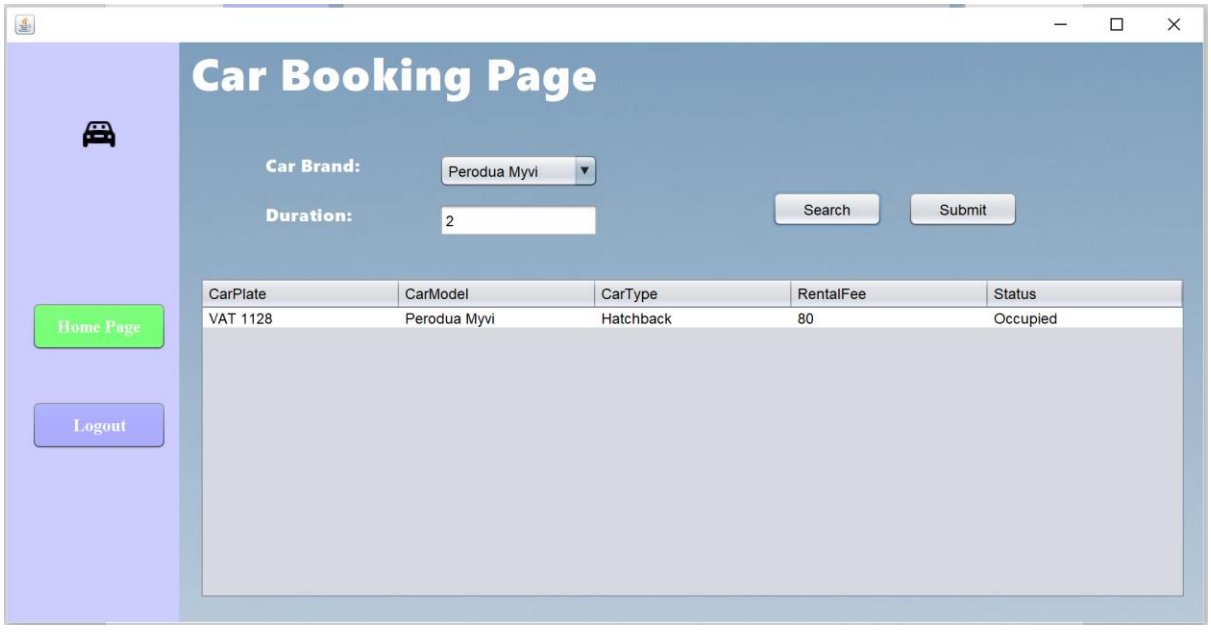


Figure 4.1.11

After the search button clicked, the system will then show the car details in JTable.

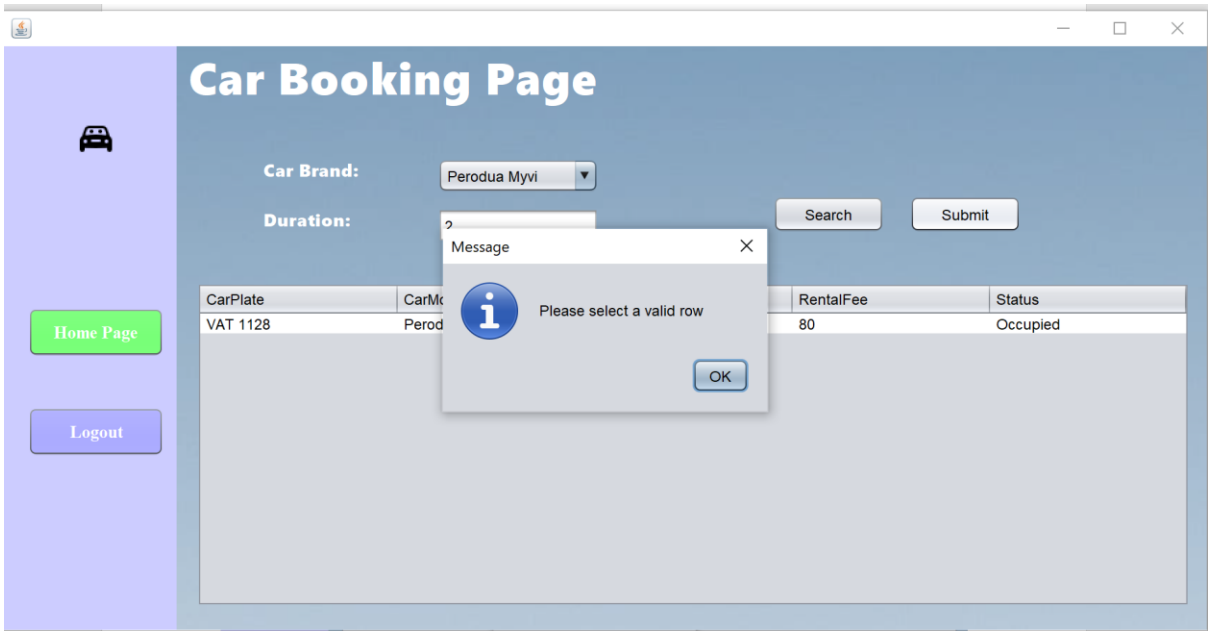


Figure 4.1.12

Before click on the submit button, customer need to manually select the row if not the system will prompt a message to guide customer to choose the row.

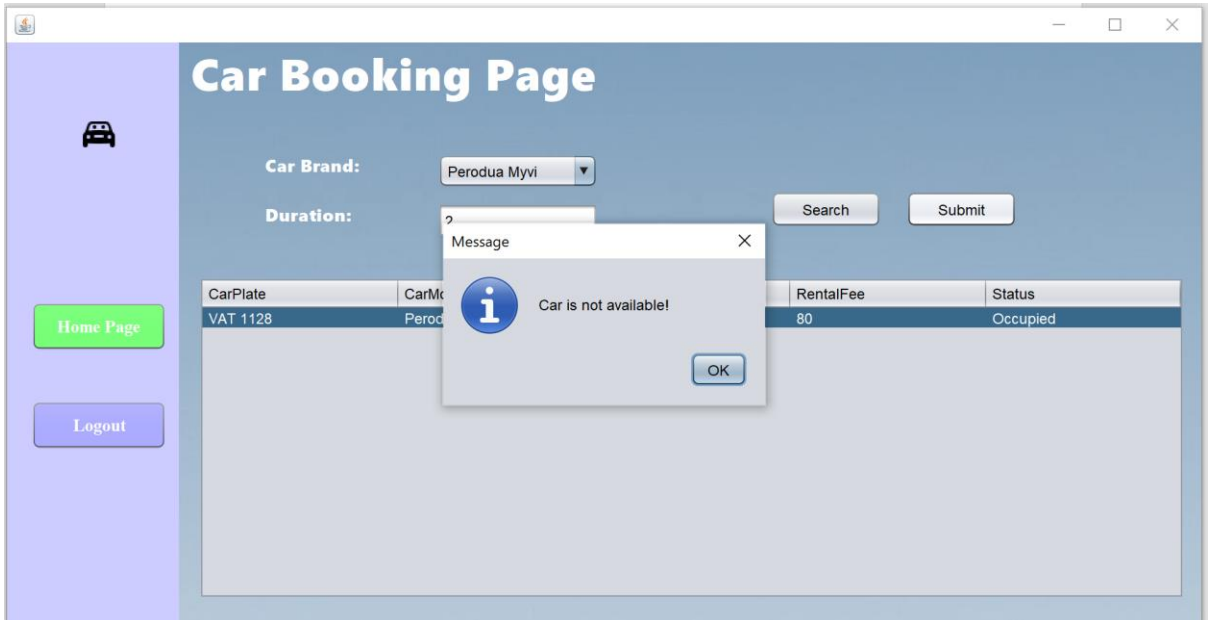


Figure 4.1.13

Customer are not allowed to choose the car with status occupied. The system will prompt a message to ask customer choose again if customer select the car with occupied status.

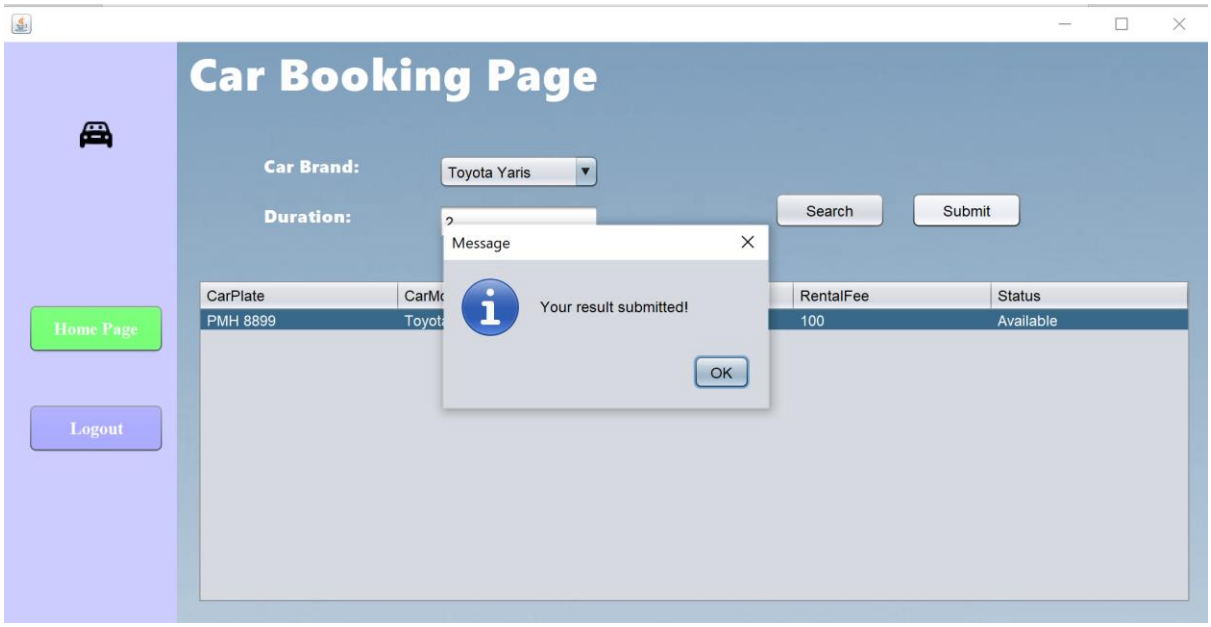


Figure 4.1.14

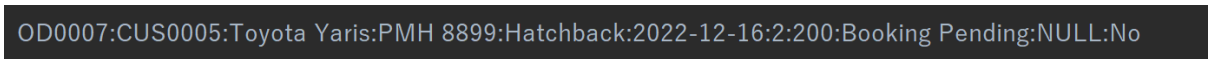


Figure 4.1.15

Figure 4.1.16

When the result is submitted, the system will prompt a message. The text file of Customer booking will then show the order id, customer id, car brand, car plate, car type, rental date, rental duration, rental fee, booking status, payment method and booking confirmation. The payment status will remain null until the payment steps. The car information text file will change the status of car from available to occupied.

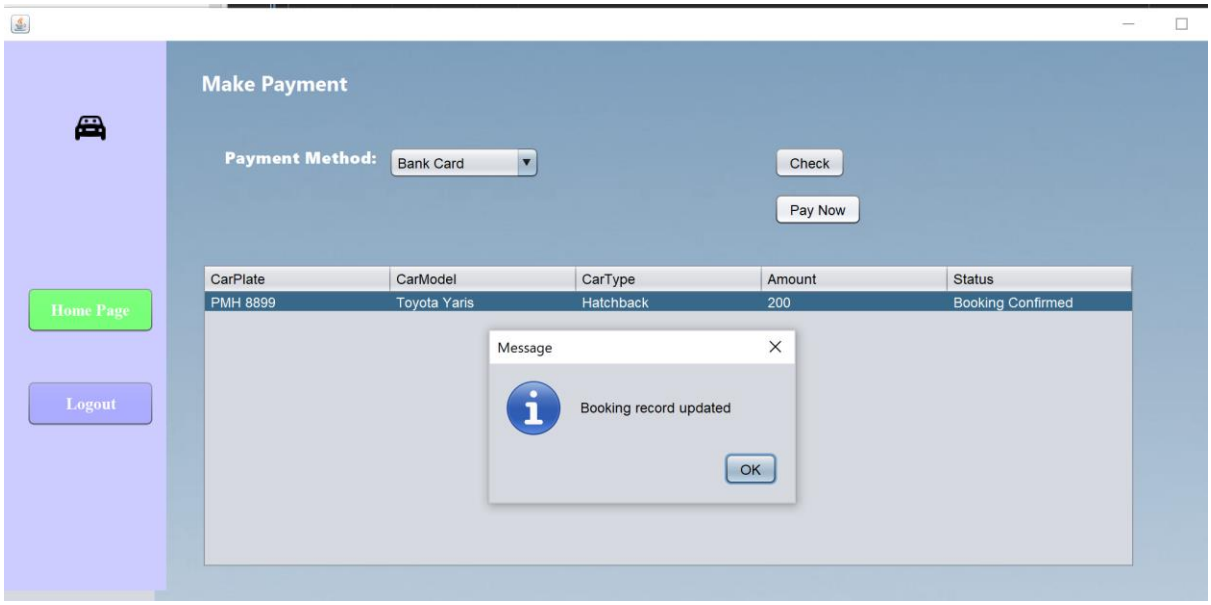


Figure 4.1.17

```
OD0007:CUS0005:Toyota Yaris:PMH 8899:Hatchback:2022-12-16:2:200:In Rent:Bank Card:Yes
```

Figure 4.1.18

Customer able to check the payment status by clicking on the button check. The details of the car and payment fees will show in JTable, After the button pay now clicked, the system will show a message “Booking record updated”. The customer booking text file will append the status of car from “Booking Pending” to “In Rent”. Besides, the payment method will also change from null to “Bank Card”.

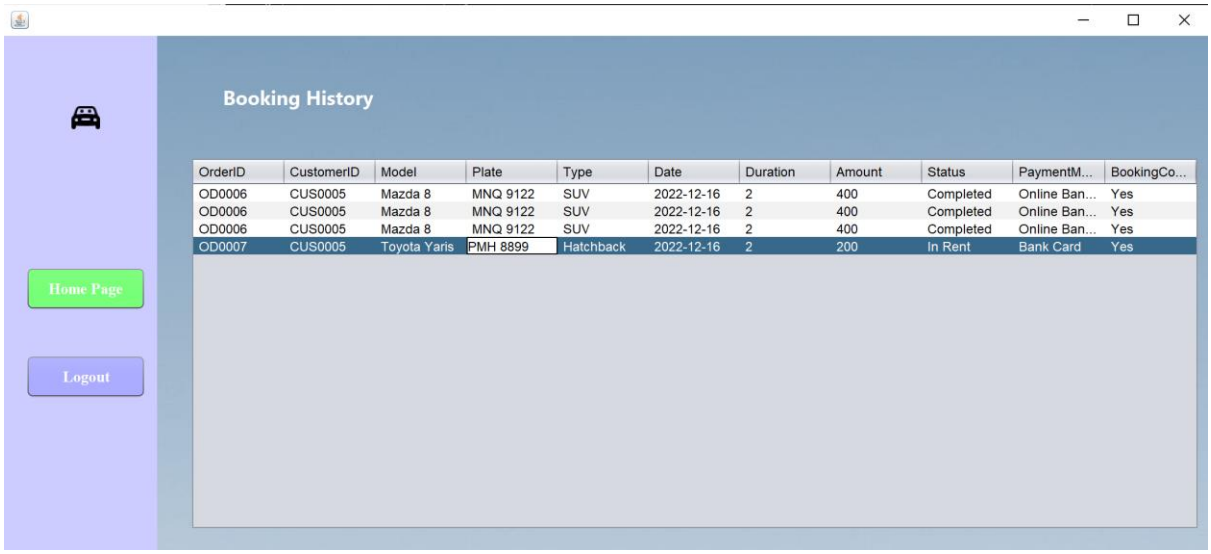


Figure 4.1.19

Customer able to check the booking history in booking history page. The system will read the customer's username and show all booking history of that particular customer.

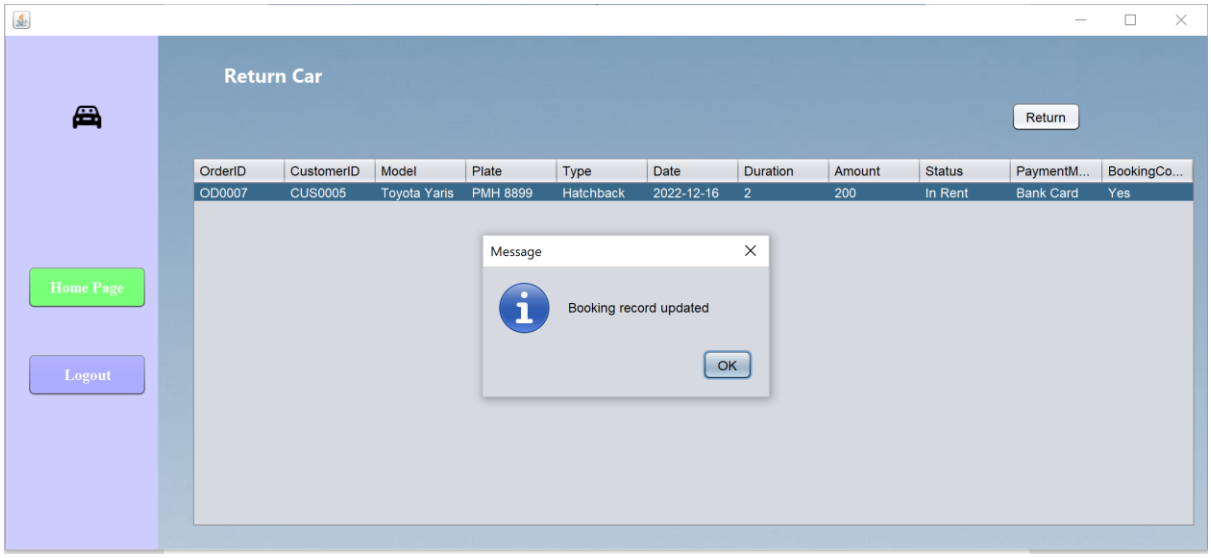


Figure 4.1.20

```
OD0007:CUS0005:Toyota Yaris:PMH 8899:Hatchback:2022-12-16:2:200:Completed:Bank Card:Yes
```

Figure 4.1.21

```
PMH 8899:Toyota Yaris:Hatchback:100:Available
```

Figure 4.1.22

Customer need to return the car by using the car return page. When the row and return button is being clicked, the system will show a message booking record updated. The customer booking text file will append the status from “In Rent” to “Completed. The car information text file will also change the status from “Occupied” to “Available”.

Admin

Admin: Login

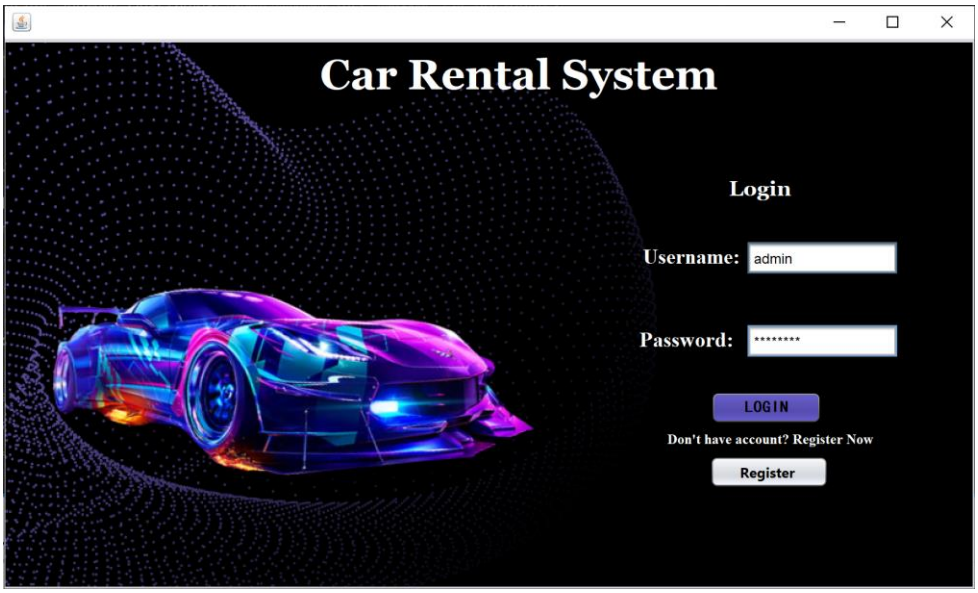


Figure 4.1.23

When the username: “admin” and password: “admin123” is entered in the login page, then system will allow the user to login into the system as the role: admin.

Admin: Main Menu

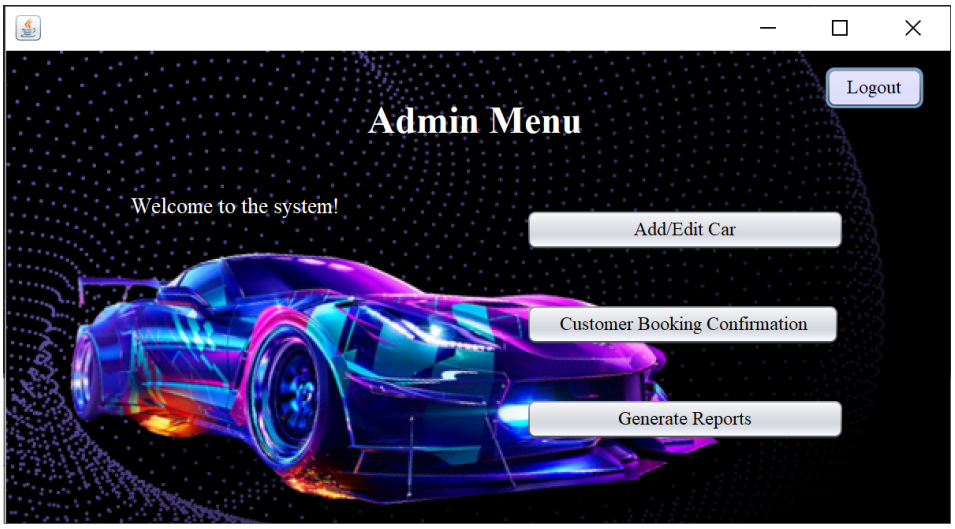


Figure 4.1.24

After the admin successfully login into the system, there will be three options for the admin. The admin is able to add/edit car, customer booking confirmation and generate reports.

Admin: Add/Edit Car

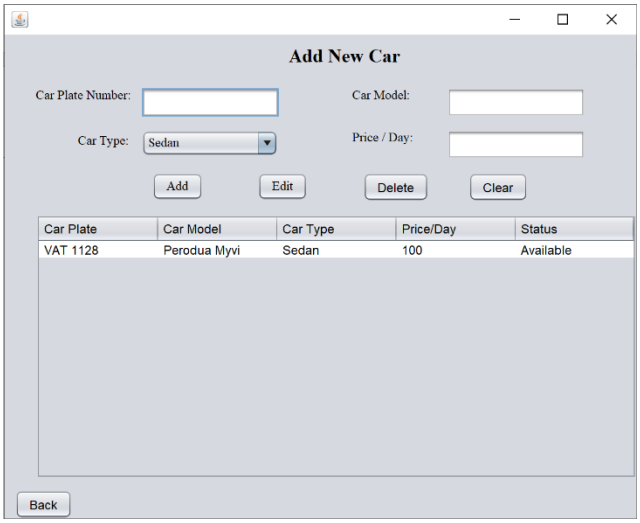


Figure 4.1.25

After the admin select the "Add/Edit" options, the JFrame above will appear and allows admin to add, edit or delete cars.

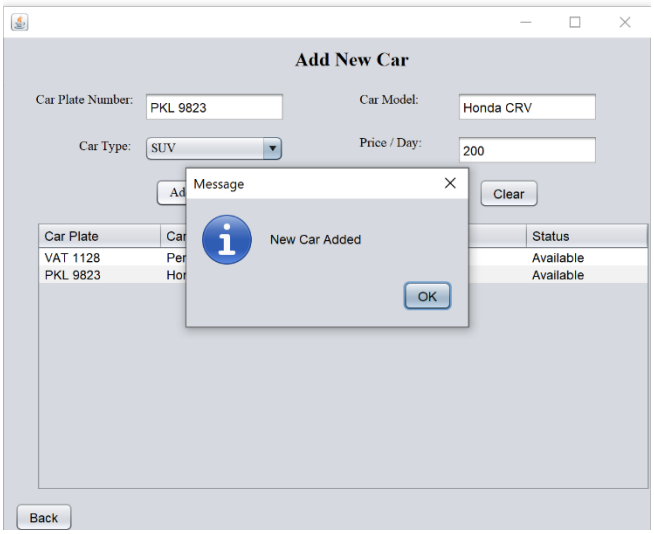


Figure 4.1.26

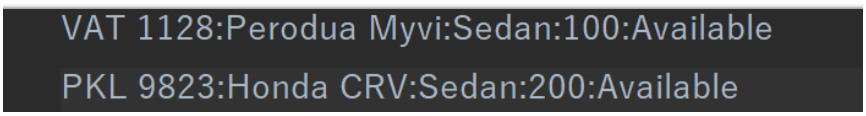


Figure 4.1.27

If admin wants to add a new car to the system, he/she should enter the full details of the car. After the “Add” button is clicked, the new car will be added to the system and a JOptionPane will be displayed showing the message “New Car Added”.

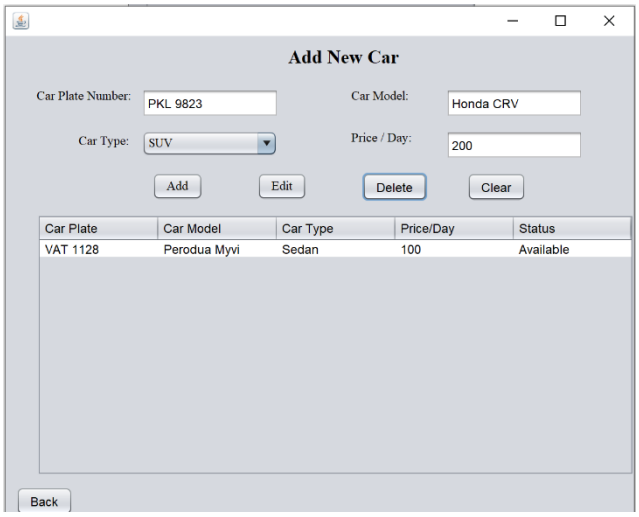


Figure 4.1.28

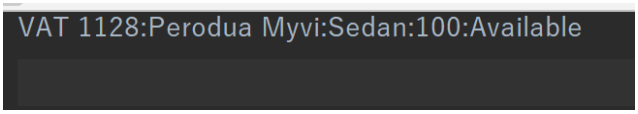


Figure 4.1.29

If one row is selected and the “Delete” button is clicked, the selected car information will be deleted.

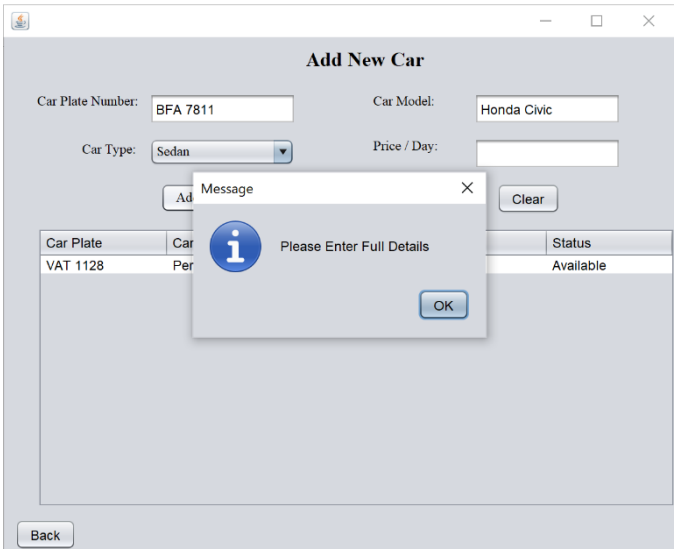


Figure 4.1.30

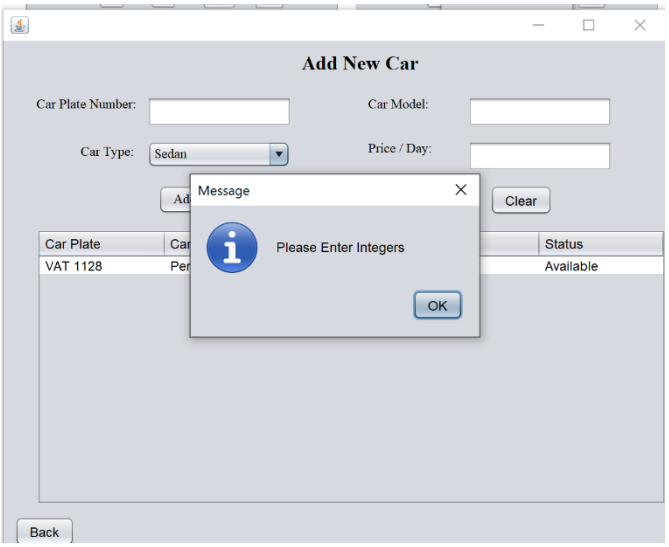


Figure 4.1.31

The figures above show the validation during the input of the car information. If the full details are not filled in by the admin, a message “Please Enter Full Details” will be displayed. While the price text field is entered with characters, a message “Please Enter Integers” will be displayed.

Admin: Booking Confirmation

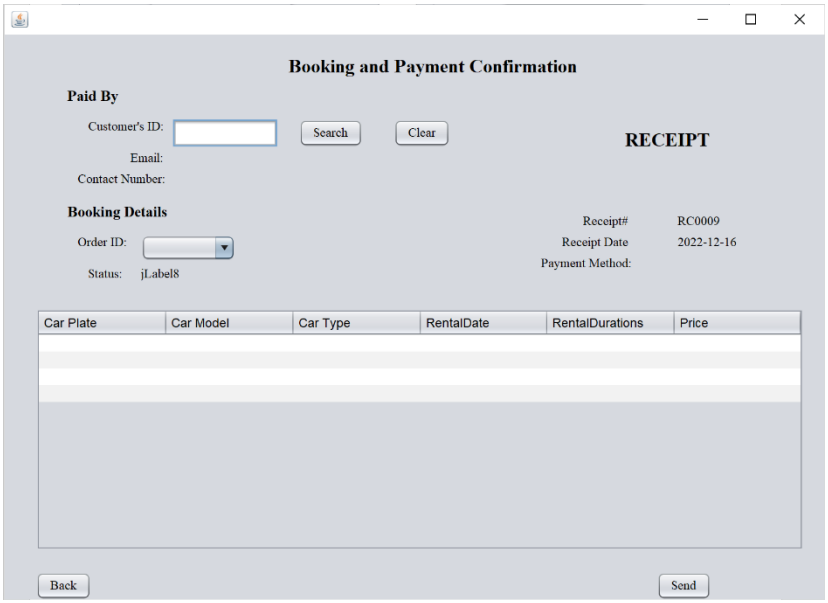


Figure 4.1.32

When the option “Customer Booking Confirmation” is selected, the Jframe above will be shown.

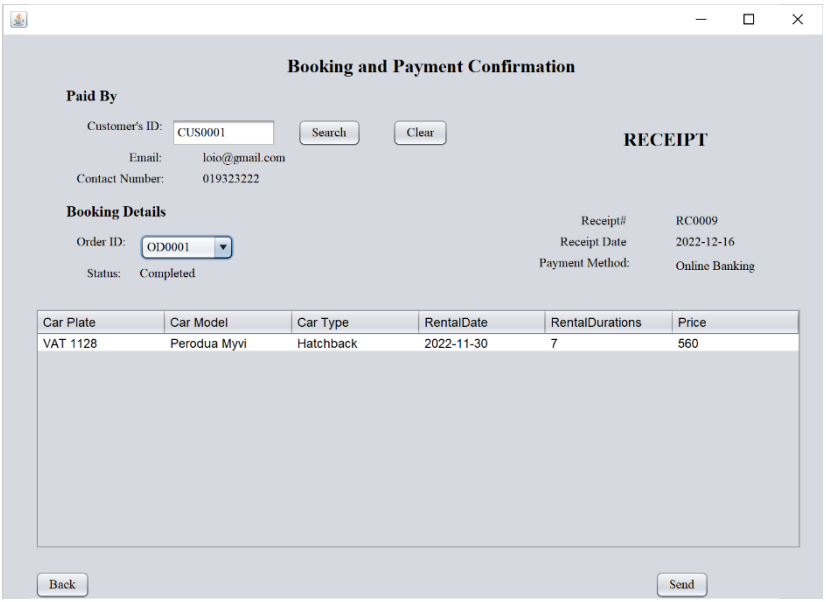


Figure 4.1.33

Firstly, the admin should type the Customer ID in the textfield: Customer’s ID. While the “Search” button is clicked, the details of the booking will be displayed in the system and the order ID of the customer can be selected in the combo box.

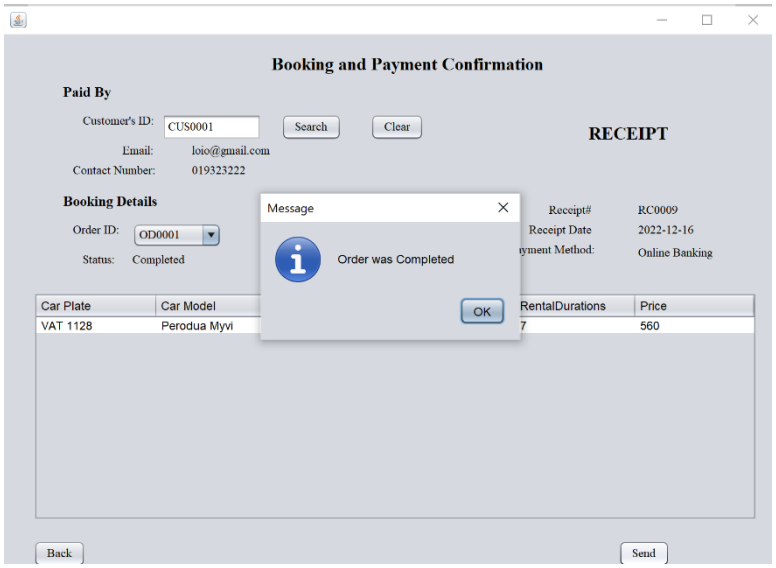


Figure 4.1.34

If the status of the booking is “Completed / In Rent”, a message “Order was Completed” will be displayed. This indicates that the order was already confirmed by the admin so no data will be changed.

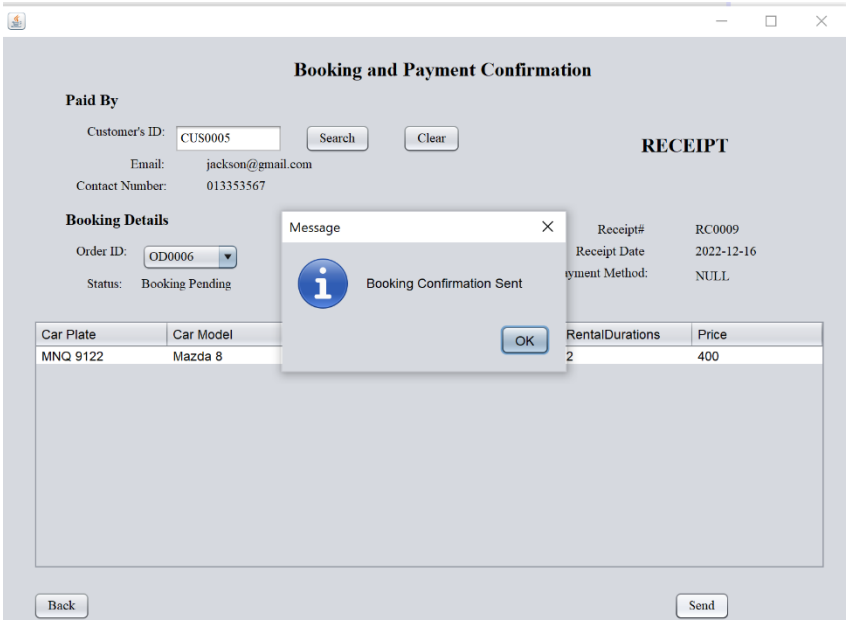


Figure 4.1.35

Else if the status of the booking is “Booking Pending/Payment Pending/Car Returned”, a message “Booking Confirmation Sent” will be displayed. This indicates that the order is confirmed by the admin and the next step can be taken by the customer.

```
OD0001:CUS0001:Perodua Myvi:VAT 1128:Hatchback:2022-11-30:7:560:Completed:Online Banking:Yes
OD0002:CUS0002:Toyota Yaris:PMH 8899:Hatchback:2022-12-01:3:300:Completed:Online Banking:Yes
OD0003:CUS0003:Mazda 8:MNQ 9122:SUV:2022-12-04:2:400:Completed:Online Banking:Yes
OD0004:CUS0004:Honda CRV:PKL 9203:SUV:2022-12-05:3:600:In Rent:Online Banking:Yes
OD0006:CUS0005:Mazda 8:MNQ 9122:SUV:2022-12-05:2:400:Booking Confirmed:NULL:Yes
OD0007:CUS0001:Perodua Myvi:VAT1128:Hatchback:2022-12-07:1:80:In Rent:Online Banking:Yes
```

Figure 4.1.36

The figure above shows the data stored in the CusBooking.txt. This is for the Booking Confirmation Jframe.

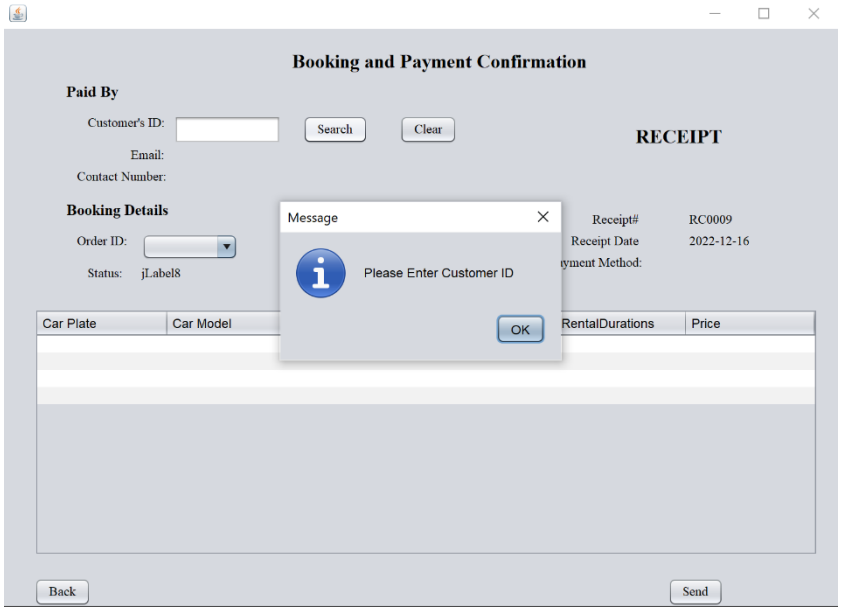


Figure 4.1.37

If the “Search” button is clicked while there is no input in the Customer ID, a message will be shown “Please Enter Customer ID”. This is to warn the admin that it is a must to enter Customer ID to get the booking details of the customer.

Admin: Generate Report

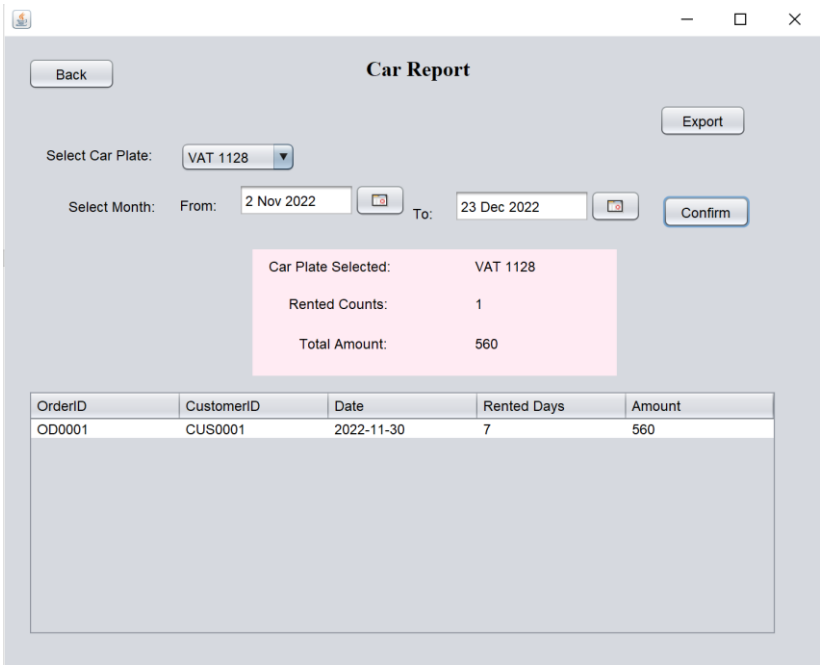


Figure 4.1.38

The Jframe above will be displayed if the admin selects the "Generate Report" button. First, the admin will choose the desired car plate from the combo box. Next, the start date and end date must be selected to print the report within the date range. The details will be displayed in the JTable, the car plate selected, rented counts and total amount will be displayed in the system.

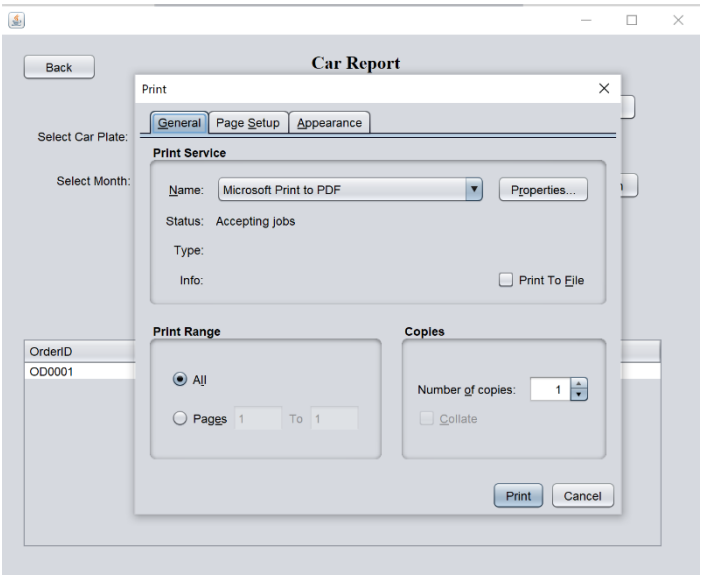


Figure 4.1.39

When admin clicks the "Export" button, the window displayed in the figure above will prompt the admin to choose the preferred file format to be printed.

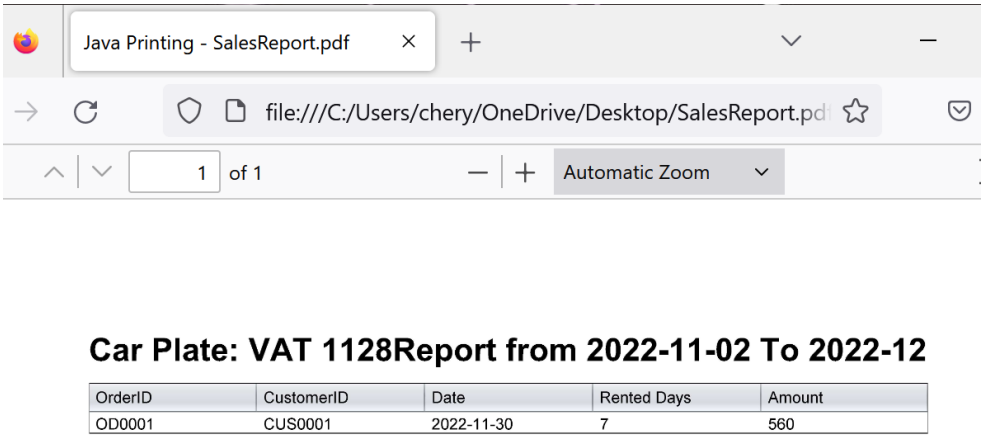


Figure 4.1.40

If the admin chose to export the sales report to pdf, the details will be shown in the PDF file as shown in the figure above.

5.0 Conclusion

In conclusion, the car rental system has successfully coded and built by Java Programming Language in Netbeans IDLE. Use case diagram and class diagram has been created and applied in this assignment to illustrate and define the function of the car rental system. This car rental system has achieved all the requirements that stated in the assignment documentation with other special features. GUI has been implemented in this system to make the features easier more user friendly.

6.0 References

Advantages of Java. (2020). Retrieved December 16, 2022, from https://www.ibm.com/docs/en/aix/7.1?topic=monitoring-advantages-java

GeeksforGeeks. (2022, November 21). *Classes and Objects in Java*. Retrieved December 16, 2022, from https://www.geeksforgeeks.org/classes-objects-java/

GeeksforGeeks. (2022, November 21). *Classes and Objects in Java*. Retrieved December 16, 2022, from https://www.geeksforgeeks.org/classes-objects-java/

JanbaskTraining. (2018, November 29). *What is static class in Java? List of static nested classes in Java*. https://www.janbasktraining.com/blog/what-is-static-class-in-java/

Techopedia. (2022b, September 1). *Java Object*. Techopedia.com. Retrieved December 16, 2022, from https://www.techopedia.com/definition/24339/java-object