

**ASSIGNMENT**

**TECHNOLOGY PARK MALAYSIA**

**CT038-3.4-2-OODJ**

**OBJECT-ORIENTED DEVELOPMENT WITH JAVA**

**APD2F2209IT(FT)**

**HAND OUT DATE: 19TH SEPTEMBER 2022**

**HAND IN DATE: 9TH DECEMBER 2022**

**WEIGHTAGE: 50%**

****

**INSTRUCTIONS TO CADIDATES:**

1. **Submit your assignment at the administrative counter**
2. **Students are advised to underpin their answers with the use of references (cited using the Harvard Name System of Referencing)**
3. **Late submission will be awarded zero(0) 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 soft copy, the soft copy 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.**

Table of Contents

[1.0 Introduction 5](#_Toc122089496)

[1.1 Assumptions 5](#_Toc122089497)

[2.0 Output sample for admin user 6](#_Toc122089498)

[2.1 Login Page 6](#_Toc122089499)

[2.3 Car Management page 10](#_Toc122089500)

[Adding cars 11](#_Toc122089501)

[Update car 17](#_Toc122089502)

[Delete car 25](#_Toc122089503)

[2.4 Booking management 27](#_Toc122089504)

[2.5 Manage report 39](#_Toc122089505)

[3.0 Output example for Customer’s 43](#_Toc122089506)

[3.1 Registration Part 43](#_Toc122089507)

[3.2 Login Part 57](#_Toc122089508)

[3.3 Forgot Password Part 59](#_Toc122089509)

[3.4 Update Customer’s Personal Information Part 64](#_Toc122089510)

[3.5 Car Renting Page 76](#_Toc122089511)

[3.6 Rental Payment Page 81](#_Toc122089512)

[3.7 Check Receipt Feature 96](#_Toc122089513)

[3.8 Check Booking Feature 101](#_Toc122089514)

[**3.9 Cancel Booking** 102](#_Toc122089515)

[3.10 Submit Feedback 108](#_Toc122089516)

[3.11 Return Car 110](#_Toc122089517)

[4.0 Source codes 116](#_Toc122089518)

[4.1 Encapsulation 116](#_Toc122089519)

[4.2 Constructor and methods 117](#_Toc122089520)

[4.3 Abstractions 118](#_Toc122089521)

[4.4 Modularity 119](#_Toc122089522)

[4.5 High Cohesion 122](#_Toc122089523)

[4.7 Loose Coupling 125](#_Toc122089524)

[5.0 Diagrams 127](#_Toc122089525)

[5.1 Case diagram 127](#_Toc122089526)

[Diagram

Description automatically generated 127](#_Toc122089527)

[5.2 Class diagram 134](#_Toc122089528)

[6.0 Extra Features 135](#_Toc122089529)

[6.1 Forgot Password Feature 135](#_Toc122089530)

[6.2 Cancel Booking/Refund Feature 139](#_Toc122089531)

[6.3 Blacklist 144](#_Toc122089532)

[7.0 References 146](#_Toc122089533)

[1bestcsharp (n.d.) - How to Populate JTable From Txt File Text Using Java NetBeans 146](#_Toc122089534)

# Introduction

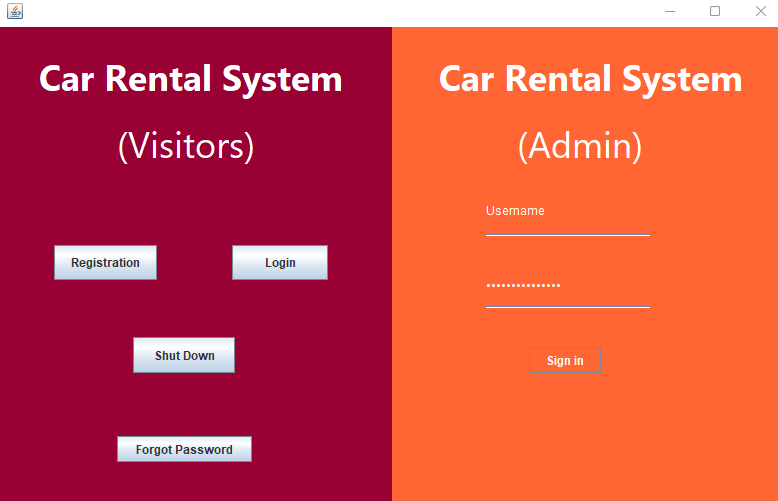
This report includes the design of a car management system with the implementation of object-oriented concept with Java language. There are only 2 users, which are admin and customers. Admin can manage cars’ information; handle customers’ booking and generate reports. While customers can register themselves as users, make car rental bookings, view their booking history.

## 1.1 Assumptions

1. Assume that admins can only log in to system using 1 account with username “admin” and password “123”.
2. Assume that blacklisted customer is not able to register another account.

# 2.0 Output sample for admin user

## 2.1 Login Page



*Figure 2.1.1 Login Page*

As shown in *figure 2.1.1*, this will be the first interface for the car rental system. On the left, there will be 3 buttons for customers to register, login and cancel to quit the system. On the right, there will be 2 text fields for admin to put in the fixed username and password to grand access to the admin menu.

Graphical user interface, application, Word

Description automatically generated

*Figure 2.1.2 Invalid admin login*

If there is any logins action with invalid username and password, there will be a message dialog to prompt admin to enter the correct login details as shown in *figure 2.1.2*.

* 1. Admin Menu

Graphical user interface, application

Description automatically generated

*Figure 2.2.1 Admin’s menu*

As shown in *figure 2.2.1* above, the admin menu has a table on the right and 2 buttons to show cars and orders. While on the left, there will be 3 buttons that direct the user to another page which are car management, booking management and report management pages.

Table

Description automatically generated

*Figure 2.2.2 Showing car records*

*Figure 2.2.2* above shows the table is filled with car records after the button “Show Car”. The table will display the column name of CarID, Plate number, Brand, Type, Status and price and align with the car records.

Graphical user interface

Description automatically generated with medium confidence

*Figure 2.2.3 Showing booking records*

As shown in the *figure* 2.2.3 above, the records of booking made by customers are shown in the table. The column is also labeled on top with each bookingID, duration, booking date, start time, status, cutomerID and car that align to each data respectively.

Graphical user interface, application, PowerPoint

Description automatically generated

*Figure 2.2.4. Showing customers’ feedback*

## 2.3 Car Management page

Graphical user interface

Description automatically generated

*Figure 2.3.1 Car management page*

In the car management page, there will be 2 sections which allow admin to add a new car into text file. In that section, there will be 4 text fields and 2 combo box with their own value to enter which are CarID, Plate Number, Brand, car type (Sedan, Hatchback, SUV), Status (Available, Unavailable), and price. While the section below will be a section for admin to update existing car record. It also has the same exact text field as add car section. On the right will consist of a table to display the car records. Below that, there will be a text field to search for any related word on the table. Other than that, the other 2 buttons are for deleting the selected car record and to import the data from text file to the table.

### Adding cars

Chart

Description automatically generated with low confidence

*Figure 2.3.2 Adding car record*

In the process of admin typing the information of the car in each text field, the text field will set to clear on click. Not only that, but the text field for cars’ plate numbers will also automatically capitalize the alphabets typed inside. Moreover, the text field for cars’ price will not allow admin to type anything other than numbers only.

Graphical user interface

Description automatically generated

*Figure 2.3.3 Car added successfully*

*Figure 2.3.3* above featuring a successful attempt of adding a new car record into the system.

Graphical user interface, application, PowerPoint

Description automatically generated

*Figure 2.3.4 error by just clicking*

This error shown in *figure 2.3.4* appears when the “add” button is clicked without any valid information is typed in any of the text field.

Graphical user interface, PowerPoint

Description automatically generated

*Figure 2.3.5 car id duplicated*

This error message will pop out when the new car’s ID is conflicted with the existing car id. As shown in *figure* 2.3.5 above, admin is trying to add a new car record with car ID “C01”. However, there is anohter car record with ID “C01” in the table.

Graphical user interface, application, PowerPoint

Description automatically generated

*Figure 2.3.6 error plate number duplicated*

Showing in *figure 2.3.6* above, the message shows plate number is duplicated while the user tries to add a new car record with a plate number that exited in the table.

Graphical user interface, PowerPoint

Description automatically generated

*Figure 2.3.7 car id error*

In *figure 2.3.7* above shows the error message of “Car ID is empty” while the car ID text field is inserted without any value.

Graphical user interface, application, PowerPoint

Description automatically generated

*Figure 2.3.8 car plate number box is empty*

This error message in *figure 2.3.8* is to remind admin user that the car plate number text field is empty when the add button is clicked.

Graphical user interface, application, PowerPoint

Description automatically generated

*Figure 2.3.9 car brand box is empty*

Error message “Car brand is empty message” is shown as shown in *figur2.3.9 e* to inform the text field for brand box is empty while trying to add a new car record.

Graphical user interface, PowerPoint

Description automatically generated

*Figure 2.3.10 price box is empty*

In *figure 2.3.10* above is showing an error message for admin user that the text field that collect admin’s input for car’s price is empty.

### Update car

Graphical user interface, application, PowerPoint

Description automatically generated

*Figure 2.3.11 Update section*

In this update section, admin will have to select a car record to update. After that, the selected record on the table will show its current car details on each text field respectively as shown in *figure* above. Then, admins will be able to edit from each text field according to their demand. However, only the car ID text field will not be editable.

Graphical user interface

Description automatically generated

*Figure 2.3.12 detail to be update*

This *figure 2.3.12* shows the car details that are modified and ready to be updated. From plate number “ABC1234” to “AFK6969”.

Graphical user interface, PowerPoint

Description automatically generated

*Figure 2.3.13 update successfully*

*Figure 2.3.13* above shows the record being updated successfully with the modified details.

Graphical user interface, PowerPoint

Description automatically generated

*Figure 2.3.14 update error*

In *figure 2.3.14* above shows the error message “Update Error!” when the “update” button is clicked while no record has been selected.

Graphical user interface, application, PowerPoint

Description automatically generated

*Figure 2.3.15 empty car plate number*

As shown in *figure 2.3.15* above, the error message “Car plate number is empty!” pops out when the text field for plate number is left empty.

*Graphical user interface, application, PowerPoint

Description automatically generated*

*Figure 2.3.16 car brand empty*

*Figure 2.3.16* is showing the error message “Car brand is empty” that appears when text field for car’s brand is left empty.

Graphical user interface, application, PowerPoint

Description automatically generated

*Figure 2.3.17 price empty*

In *figure 2.3.17* shows the error message “Price is empty!” when admin attempts to update with empty price text field.

Search

A picture containing graphical user interface

Description automatically generated

*Figure 2.3.18*

According to the *figure 2.3.18* above shows the existing car record showing in the table that can be searched by admin.

Graphical user interface

Description automatically generated

*Figure 2.3.19* Search carID

While the admin type “C03” in the search text field, the following record with “C03” in Car ID column is shown in the table in *figure 2.3.19*.

Graphical user interface, PowerPoint

Description automatically generated

*Figure 2.3.20* search brand

In *figure 2.3.20* above, admin searches for any car that has “BMW”. Then car record with “BMW” as the brand appears in the table on the right.

Graphical user interface

Description automatically generated with medium confidence

*Figure 2.3.21* Search car type

After “Hatchback” word is entered, the car records with “Hatchback” is shown in the table as shown in *figure 2.3.21* above.

Graphical user interface

Description automatically generated with medium confidence

*Figure 2.3.22* Search status

To search any car that is available, admin will have to type “Available” in the following search text field as shown in *figure 2*.3.22 above.

### Delete car

Graphical user interface

Description automatically generated

*Figure 2.3.22 Before Delete*

From the *figure 2.3.22* above, it is shown that the car record with car ID “7” is selected and ready to be deleted.

*Graphical user interface

Description automatically generated*

*Figure 2.3.23 Delete successful*

After pressing “Delete” button, the car deleted successfully as shown in *figure 2.3.23* above.

*Graphical user interface

Description automatically generated*

*Figure 2.3.24 error only 1 selected*

If admin attempts to select multiple record and delete at once, the system will pop a message “Please select single row for delete” to remind admin to delete 1 record at once.

## 2.4 Booking management

A picture containing graphical user interface

Description automatically generated

*Figure 2.4.1 Booking management page*

In the page for booking management, admin will be able to see a table in the middle where the list of the bookings and payments made by customers will be displayed by clicking the 2 buttons on the left which are “Show payments” and “Show bookings”. Then, below will be a combo box for admin to filter booking by the status and filter payments by the payment methods. For the booking status, the filter choices include “Approved”, “Rejected”, “Completed” and “Pending”. While for payments, there will be “DebitCredit Card”, “Online banking”, “E wallet”. After that, admin user will also have the ability to manage customers’ booking request by either approving or rejecting with 2 buttons below the table. In case admin have trouble looking for records in the table, there is also a search box to solve this difficulty.

Table

Description automatically generated

*Figure 2.4.2 Show all bookings*

According to the *figure* *2.4.2* above, all bookings records made by customers are shown in the table. The details of each booking include **booking id**, **duration** that count by days, the **date that booking** was made, **the date for car renta**l, **bookings status**, **customers’ id**, and the **car plate number.**

*Table

Description automatically generated*

*Figure 2.4.3 Approve booking*

As shown in *figure* above, the booking status of record with booking ID ‘2’ was “Pending” in *figure 2.4.3*. However, after the admin selects the record and approve, the status has turned “Approved”.

*Table

Description automatically generated*

*Figure 2.4.4 Reject booking*

In *figure 2.4.4,* the status of record with booking ID “3” has changed into “Rejected”. It was “Pending” before the button is clicked in *figure.*

*Graphical user interface

Description automatically generated*

*Figure 2.4.5 Approve error*

If admin attempts to approve any booking request with status other than “Pending”, the error message “Approve error!” will pop out as shown in *figure 2.4.4*.

*Graphical user interface

Description automatically generated*

*Figure 2.4.6 Reject error*

In *figure 2.4.6* above, the error message with “Reject error!” will also pop out if admin tries to reject booking request with status other than “Pending”.

*Graphical user interface, application, PowerPoint

Description automatically generated*

*Figure 2.4.7 Show Approved bookings*

In *figure 2.4.7* above, it shows the result after the “Show approved” is pressed. All records with “Approved” status are displayed in the table.

*Graphical user interface, PowerPoint

Description automatically generated*

*Figure 2.4.8 Show rejected bookings*

While “Rejected” option is in the filter combo box, the table will show booking records with “Rejected” Status as shown in *figure 2.4.8.*

*Graphical user interface, timeline, PowerPoint

Description automatically generated*

*Figure 2.4.9 Show pending bookings*

While “Pending” is selected in the filter box, those customers booking request with status in pending status as shown in *figure 2.4.9.*

*Graphical user interface, timeline, PowerPoint

Description automatically generated with medium confidence*

*Figure 2.4.10 Show completed bookings*

As shown in *figure 2.4.10*, if admin chose to filter with “Completed”, the booking request with completed status will be displayed in the table.

Timeline

Description automatically generated with medium confidence

*Figure 2.4.11 cancelled booking*

According to the *figure 2.4.11,* the table shows booking request with status “Cancelled” when admin picks “Cancelled” in the filter box. This means that the customer has cancelled the booking on their own.

Table

Description automatically generated

*Figure 2.4.12 show all payment*

*Figure 2.4.12* shows the changes after “Show payment” button is clicked. The “Approve” and “Reject button” are disabled. However, admin can still filter the payment with different payment methods.

*Graphical user interface, PowerPoint

Description automatically generated*

*Figure 2.4.13 filter Grab Pay*

In *figure 2.4.13*, the table shows the payment record that is made with “x” payment method.

*Graphical user interface, PowerPoint

Description automatically generated*

*Figure 2.4.14 filter payment DebitCredit Card*

As shown in *figure 2.4.14*, the result of filtering payment method “x” is displayed in the table

*Graphical user interface, PowerPoint

Description automatically generated*

*Figure 2.4.15 filter Online Banking*

*Figure 2.4.15* shows the filtering result of payment record that contains “x” payment method

**

*Figure 2.4.16 filter payment Card on Delivery*

*Figure 2.4.16* is the consequences of choosing payment method “x” in the filter box.

*Graphical user interface, PowerPoint

Description automatically generated*

*Figure 2.4.17 filter payment Tng E wallet*

In figure 2.4.17, the table shows the filter result of payment record with “TnG E-Wallet”.

## 2.5 Manage report

A picture containing text

Description automatically generated

*Figure 2.5.1 Report page*

In *figure* above shows the report page after admin clicked the “Manage report” button in admin menu page. In this page, there will be mainly 5 buttons. 3 of them will be for generating reports such as **monthly sales report**, **marketing report** and **vehicle report**. Another 2 will be clearing the text field that functions as a displaying space for reports and return to admin menu.

Graphical user interface

Description automatically generated

*Figure 2.5.2. Monthly sales report*

According to *figure* above, the “Monthly Sales Report” is clicked, and the information of the report is generated. In the meantime, the buttons on top that generate reports will be disabled. This report consists of the number of bookings approved, total sales amount, and total customer using different payment methods.

*Graphical user interface, text

Description automatically generated*

*Figure 2.5.3 Marketing report*

As shown above, *figure* is showing the result after “Marketing Report” button is clicked. The 3 buttons on top are disabled. In the text field, the marketing report is displayed without allowing admin user to edit. This report includes the number of members registered, number of different genders members and number of members with age below and above 30.

*A picture containing graphical user interface

Description automatically generated*

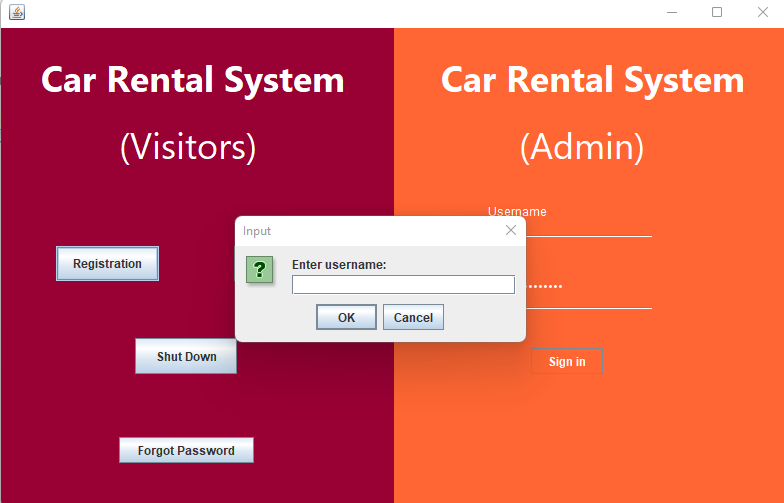
*Figure 2.5.4 Vehicle report*

In *figure* shown above, it represents the outcome of “Vehicle Report” button clicked. The text field that displays the report will not be editable which is same as the function above.

# 3.0 Output example for Customer’s

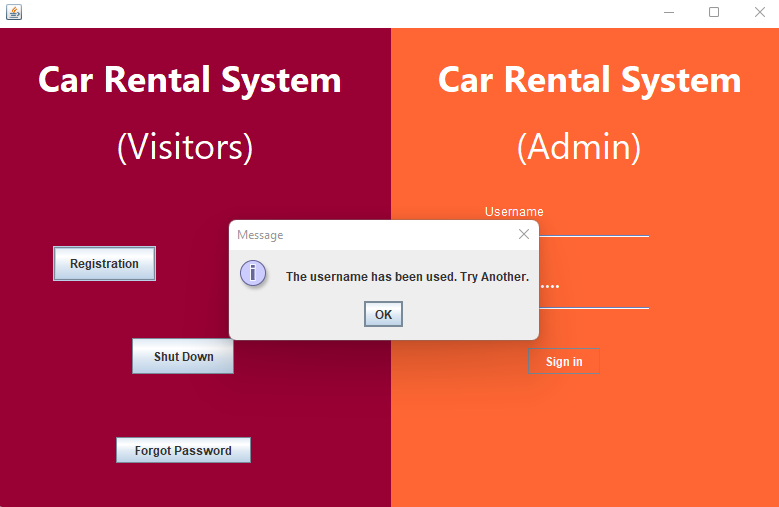
## 3.1 Registration Part

On the other hand, now it comes to the customer’s aspect of the car rental system. First will be the registration part in which customers must be register as a member to rent cars. Notice that the validation of each input of customers has been done accordingly, also empty input does not allow in this system.



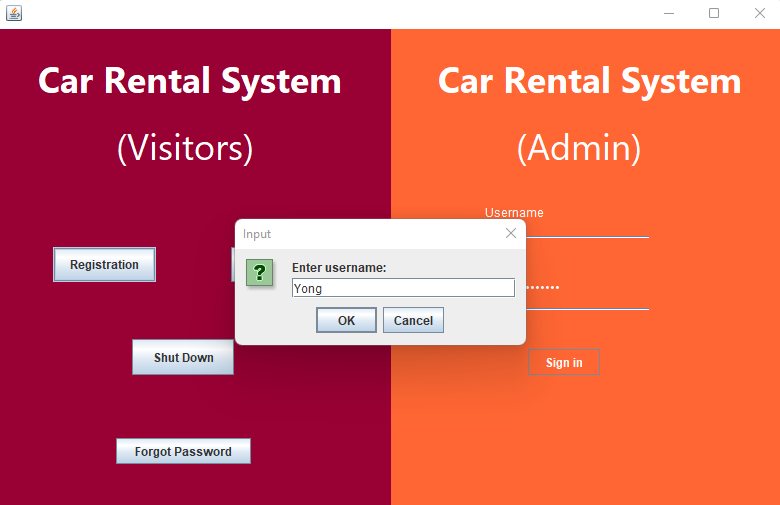
*Figure 3.1.1*

As figure above shown, after the “Registration” button was clicked an option panel will pop out and ask customers to input their username.

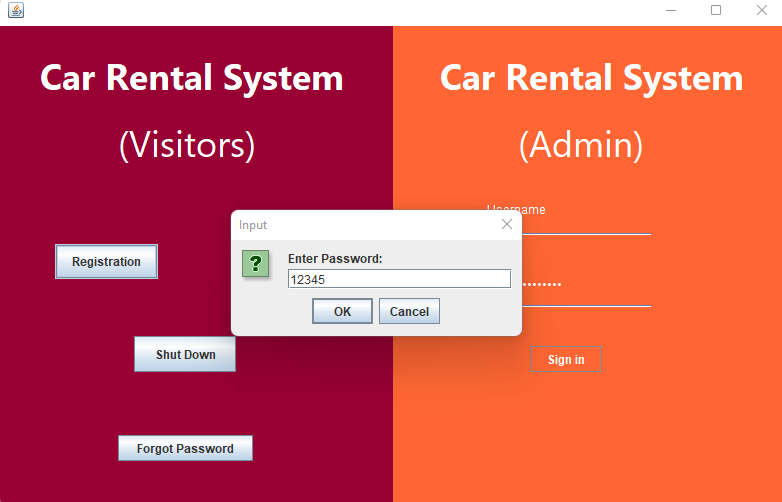


*Figure 3.1.2*

As figure above shown, if the username that customer entered was already existed inside the database of the car rental system, the system will inform customer a message and ask customer to try another.

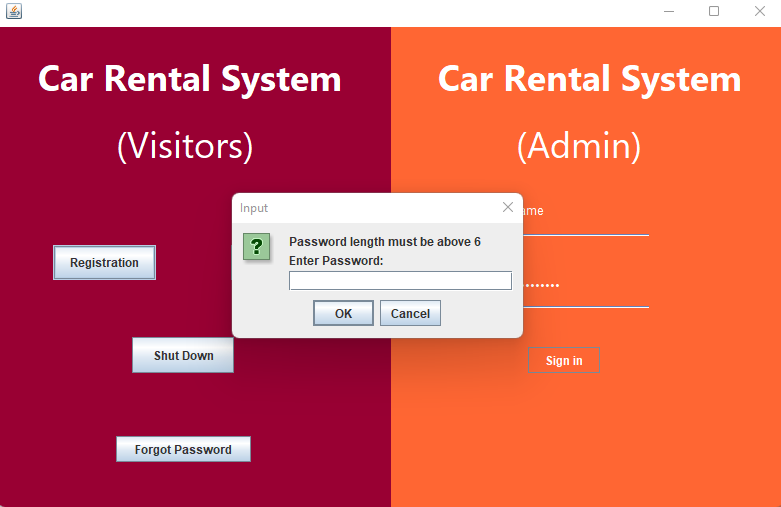


*Figure 3.1.3*



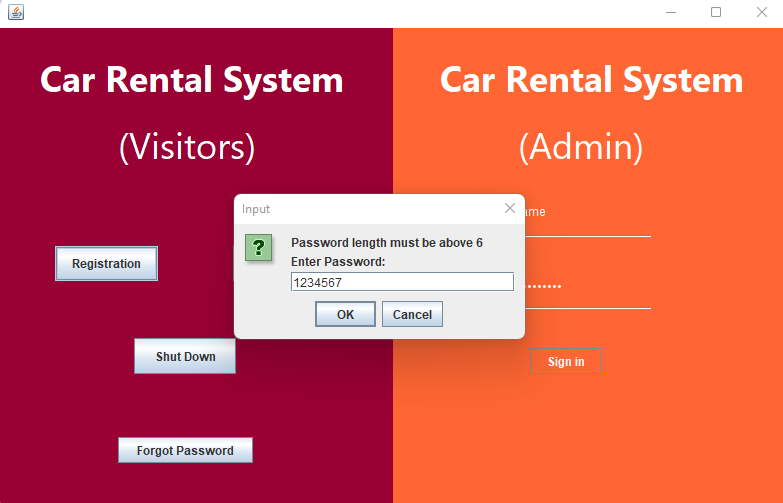
*Figure 3.1.4*

After the username has been input, the system will pop out another option pane to ask customer to input password.

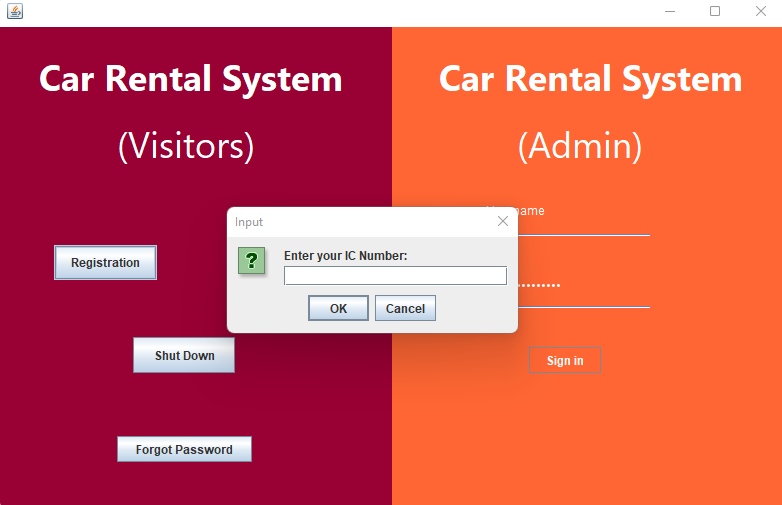


*Figure 3.1.5*

Notice that the validation of password has been made which length of password must exceed 6, or else system will ask customer to input the password again as figure shown.

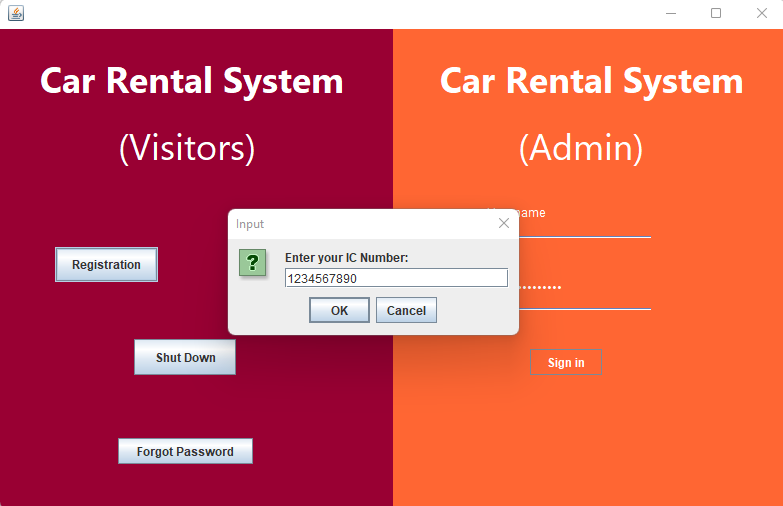


*Figure 3.1.6*



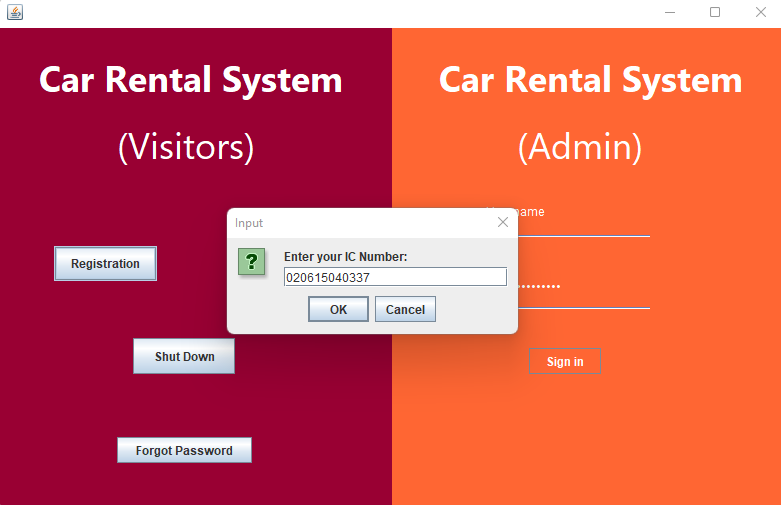
*Figure 3.1.7*

After a valid password “1234567” has been input, system will ask users to input IC number.



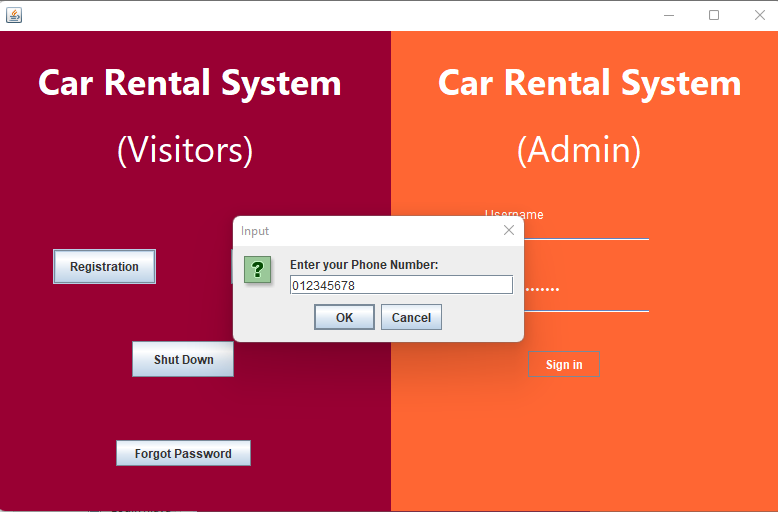
*Figure 3.1.8*

Notice that the format of IC number must be correct which is a 12-digits numbers. Length of inputs that does not equal to 12 such as figure shown “1234567890” will not be excepted and need to reenter again.

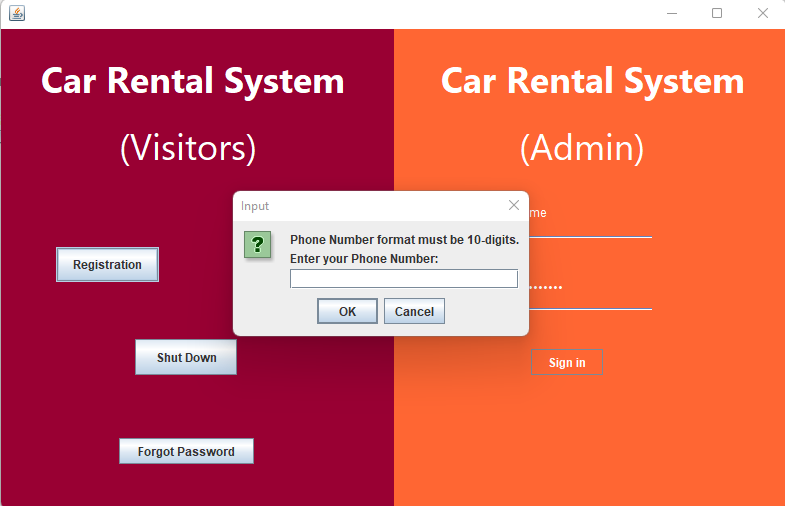


*Figure 3.1.9*

After a valid IC number “020615040337” was input, the system will proceed and ask users to input the phone number.

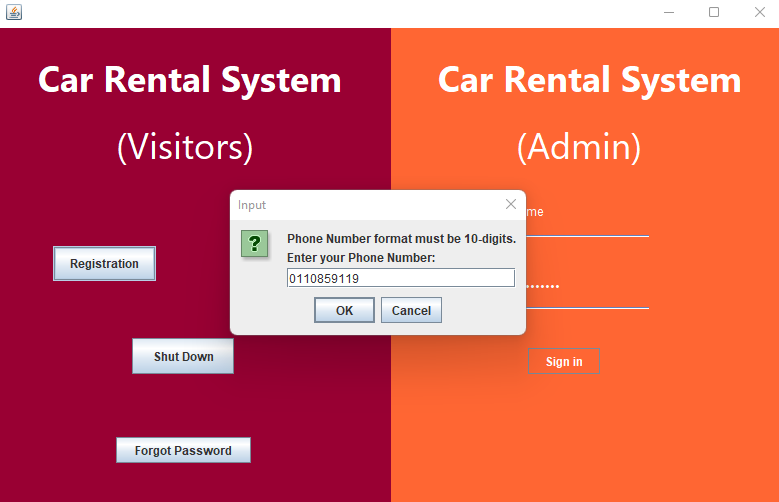


*Figure 3.1.10*

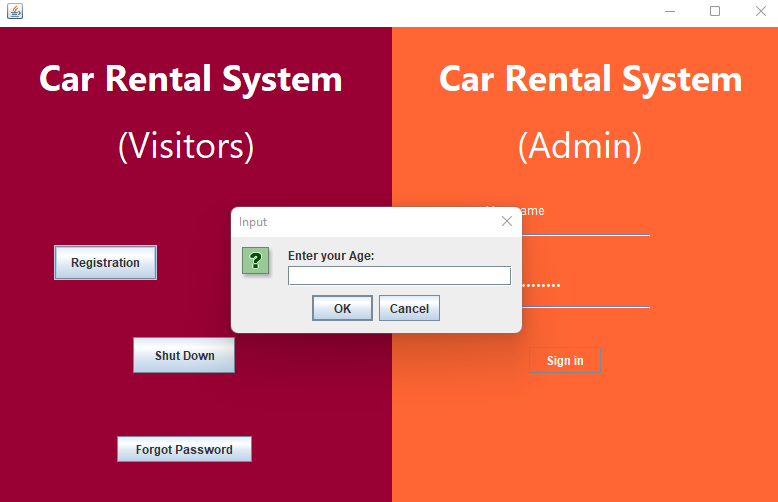


*Figure 3.1.11*

As figure above shown, after a phone number that having invalid format which the length is not 10-digits, the system will ask customer to reinput.

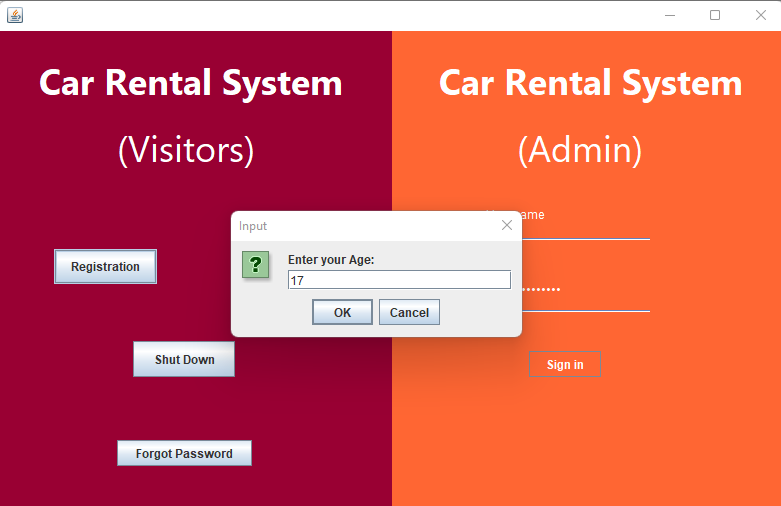


*Figure 3.1.12*

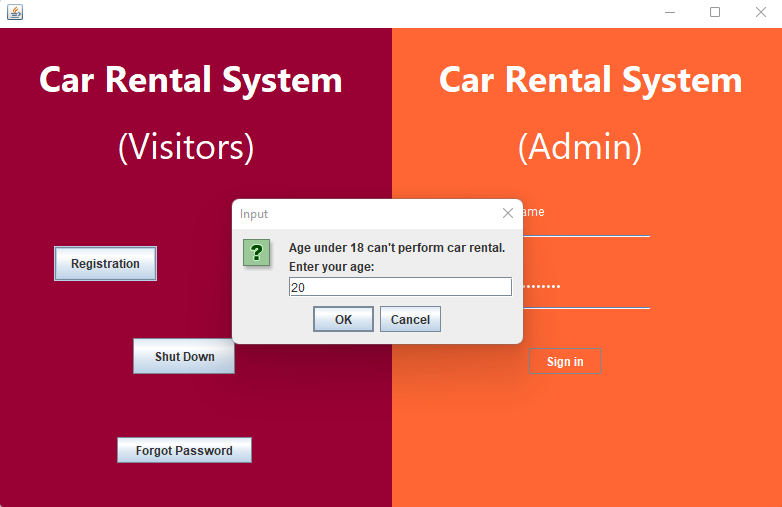


*Figure 3.1.13*

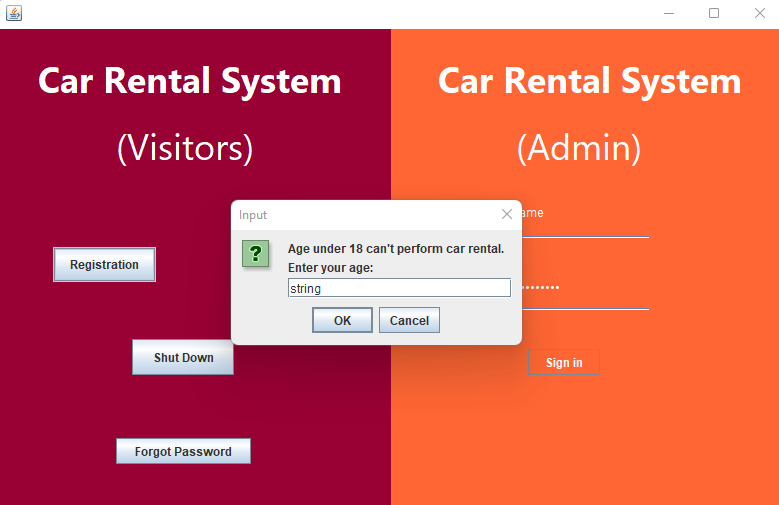
In this case, the customer entered a valid phone number “0110859119”, system then proceed to obtain age from the customer.



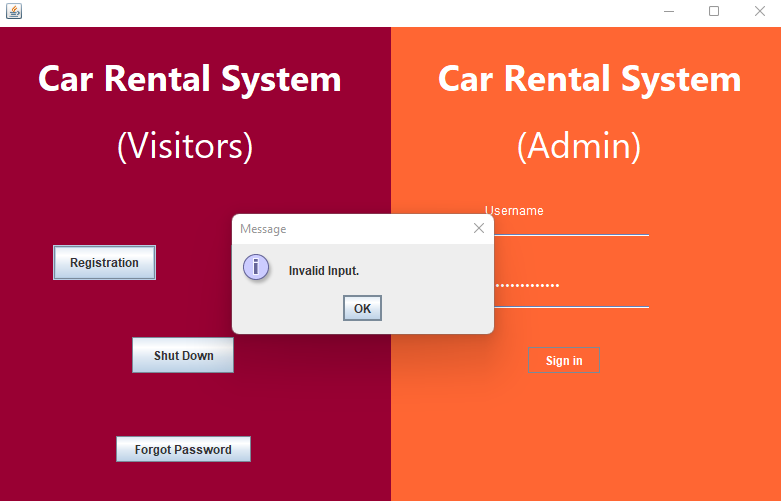
*Figure 3.1.14*



*Figure 3.1.15*

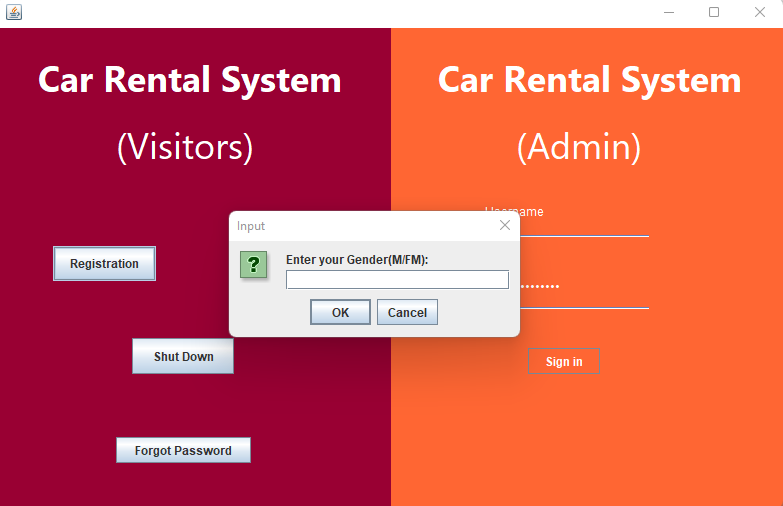


*Figure 3.1.16*

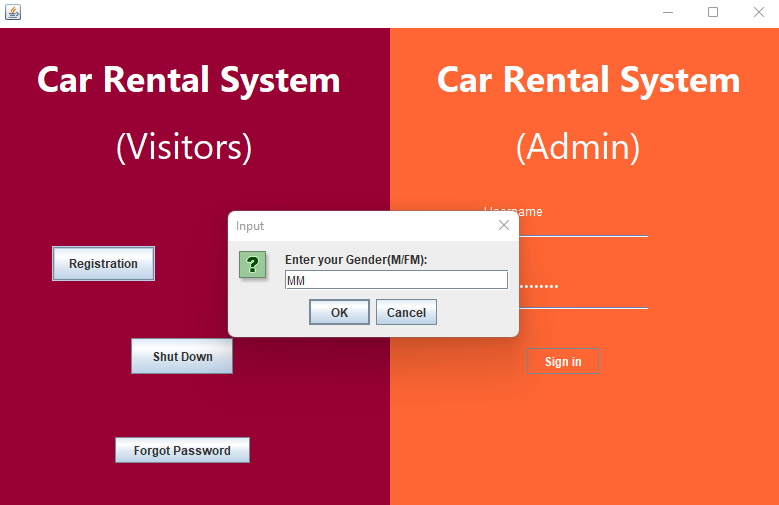


*Figure 3.1.17*

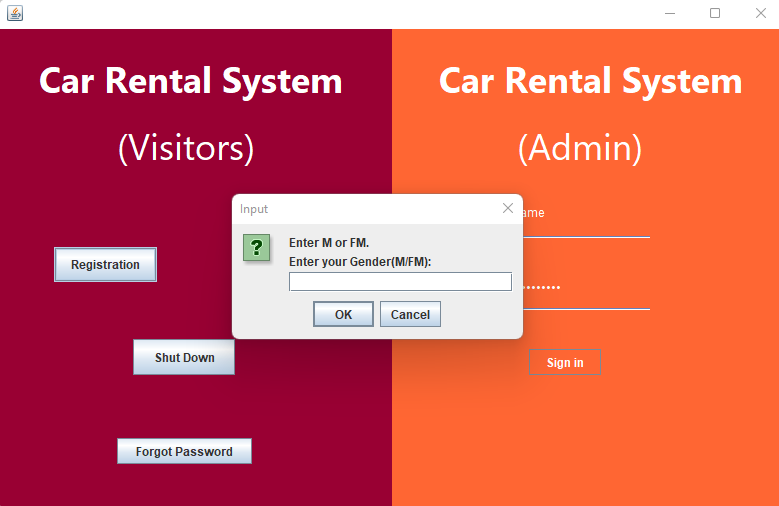
As figure above shown, age must be numeric and customer must exceed 18 years old in order to register as a member. In this case, the customer entered age of 20 so the system will proceed to the next step.



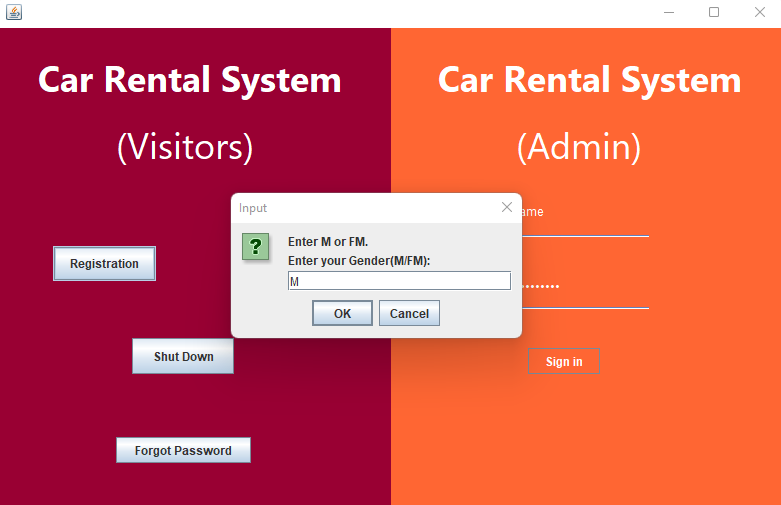
*Figure 3.1.18*



*Figure 3.1.19*

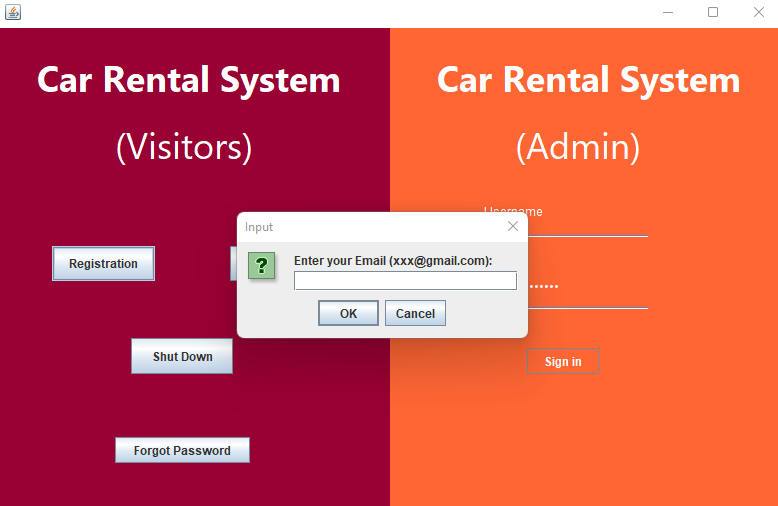


*Figure 3.1.20*



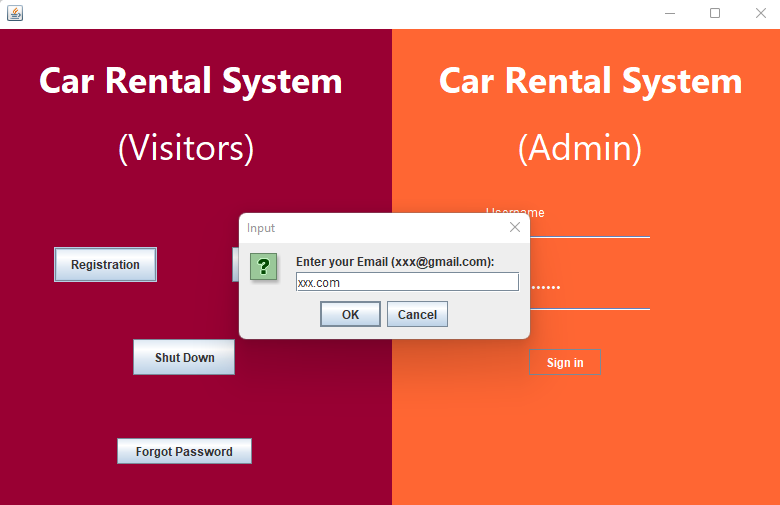
*Figure 3.1.21*

For the input customer’s gender part, customers can only enter either “M” or “FM”. Other than them, the system will ask customer to reenter again. In this case, the customer entered “MM”, so that the system rejected the input.

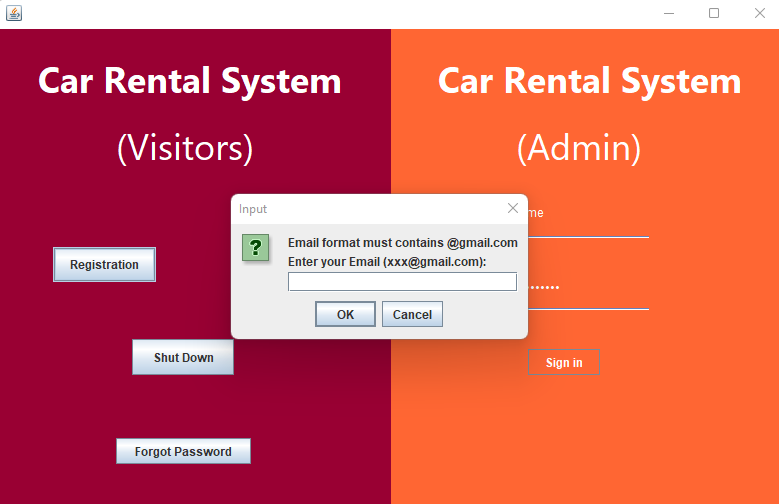


*Figure 3.1.22*

After input the valid gender “M”, next the customer will need to entered their email.

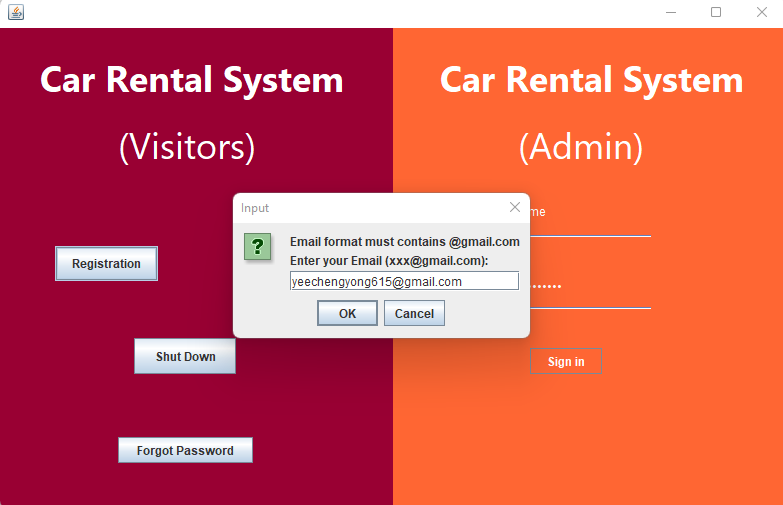


*Figure 3.1.23*

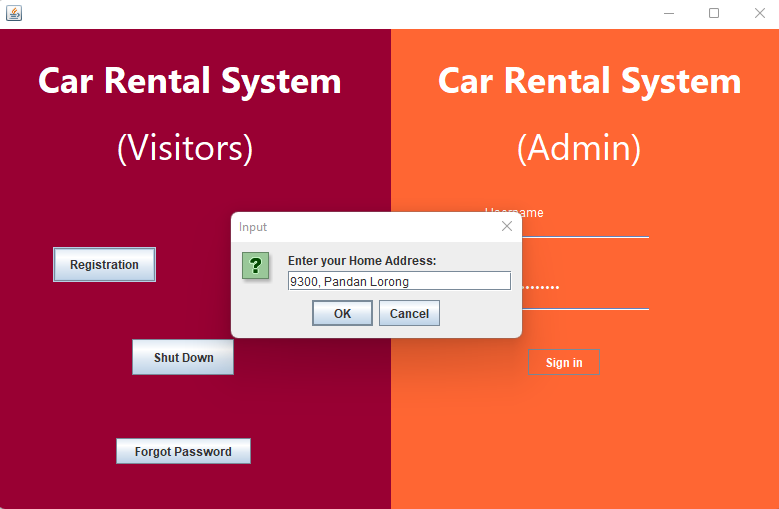


*Figure 3.1.24*

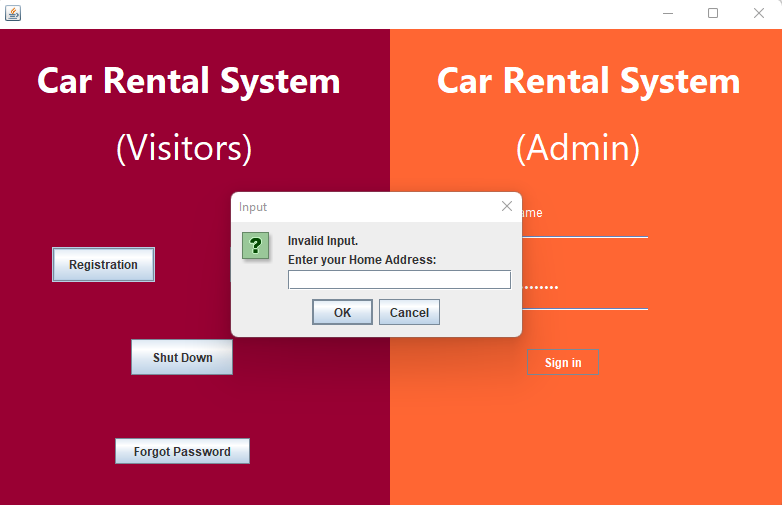
Notice that the format of email must be contains “[@gmail.com](http://@gmail.com) ”, or else system will reject the input and need to reinput again.



*Figure 3.1.25*



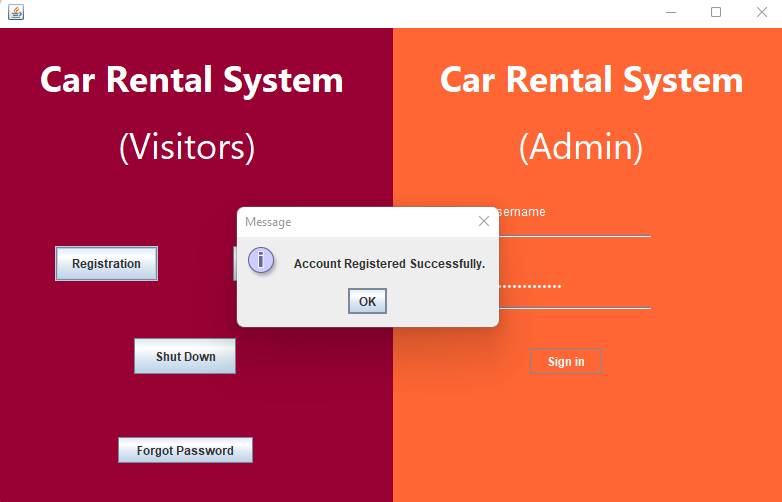
*Figure 3.1.26*



*Figure 3.1.27*

Notice that, If the customer didn’t enter a home address, the system will reject the input and ask customer to re-enter.

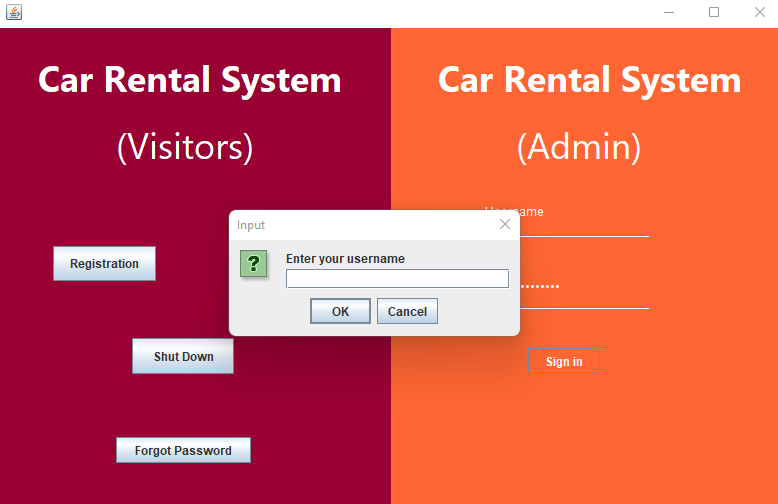
After the valid email “yeechengyong615@gmail.com” has been entered the system will ask customer to input the home address.



*Figure 3.1.28*

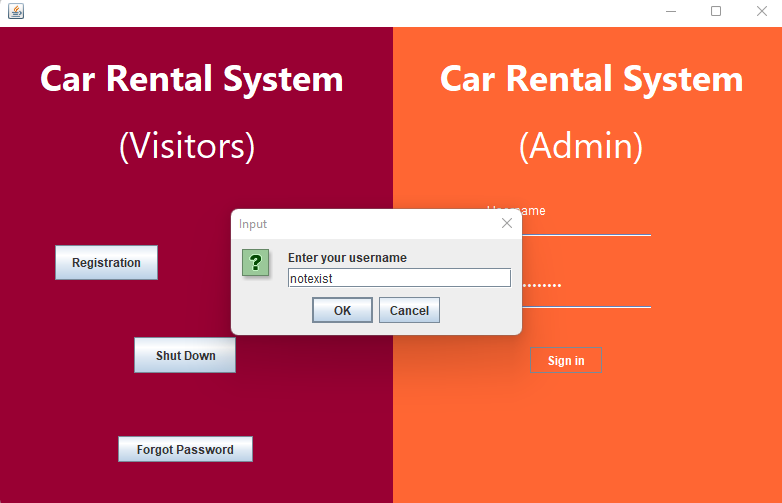
After the home address has been entered, the system will output a pane to inform customer that account registered successfully.

## 3.2 Login Part

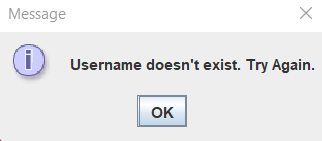


*Figure 3.2.1*

For the login part, after the customer clicked the login button, the system will ask customer to input their username first.

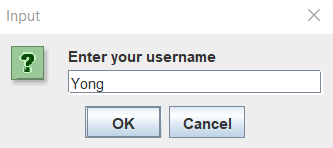


*Figure 3.2.2*

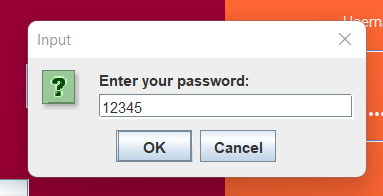


*Figure 3.2.3*

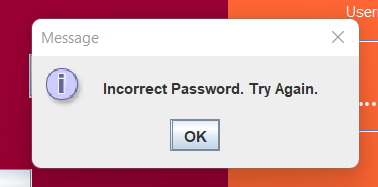
If the customer entered a username that does not exist in the customer database. The system will output message to inform the customer.



*Figure 3.2.4*



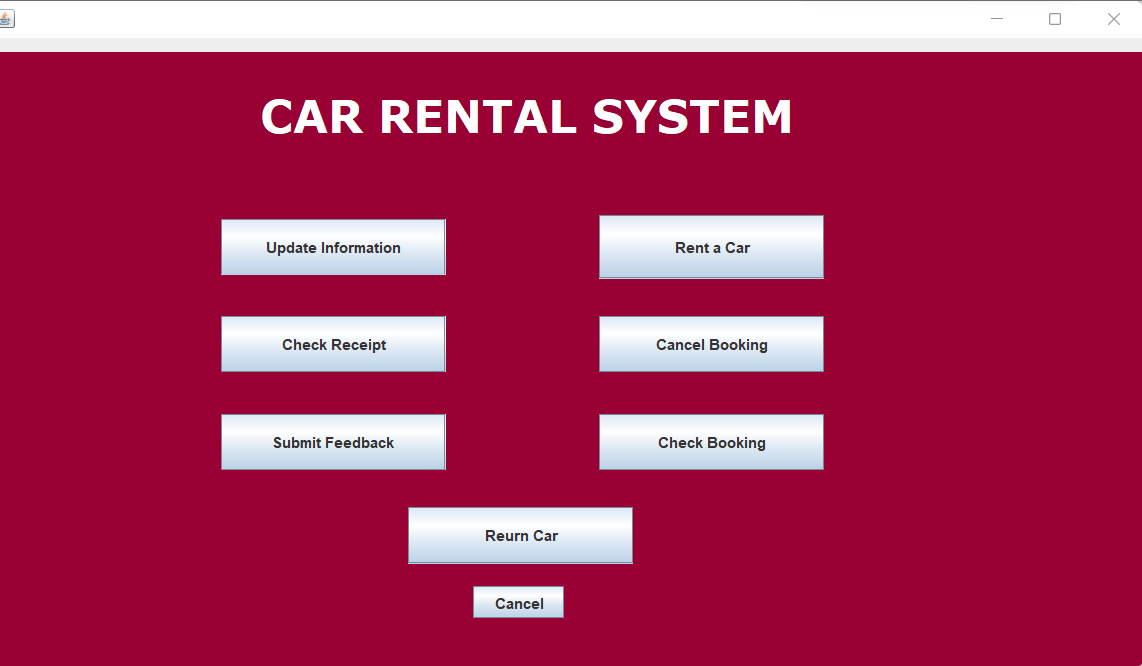
*Figure 3.2.5*



*Figure 3.2.6*

The figure shows that if the customer entered the existed username but incorrect password, the system will tell the customer to input again.

***Customer Main Menu***

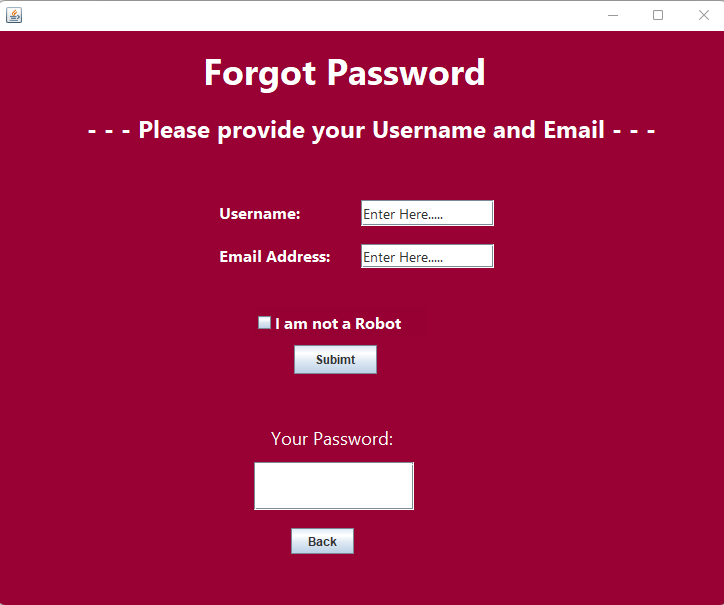


*Figure 3.2.7*

After the customer entered the username “Yong” and password “1234567” correctly, he/she will then enter the customer main menu page. There will be several main functions such as update information, rent car, check booking, check receipt, cancel booking, submit feedback and return car.

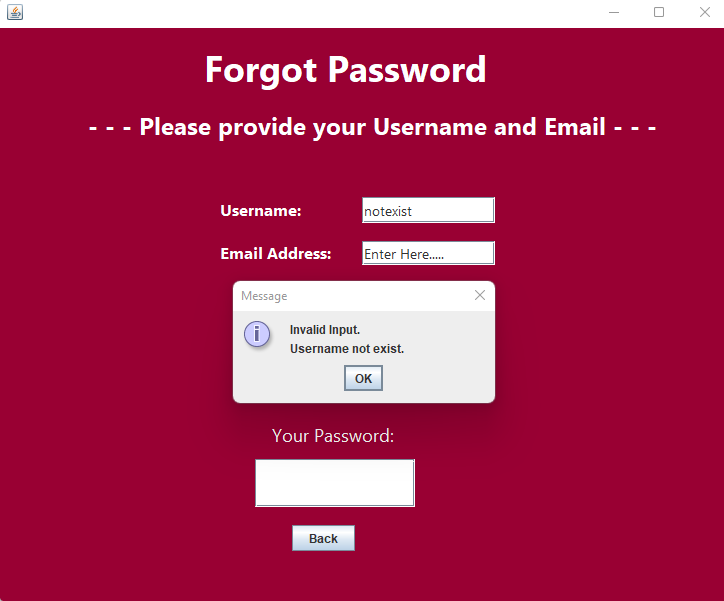
## 3.3 Forgot Password Part

There will a be an extra feature called “Forgot Password” which will be place on the register/login interface to assist the customer that having login issue on password.



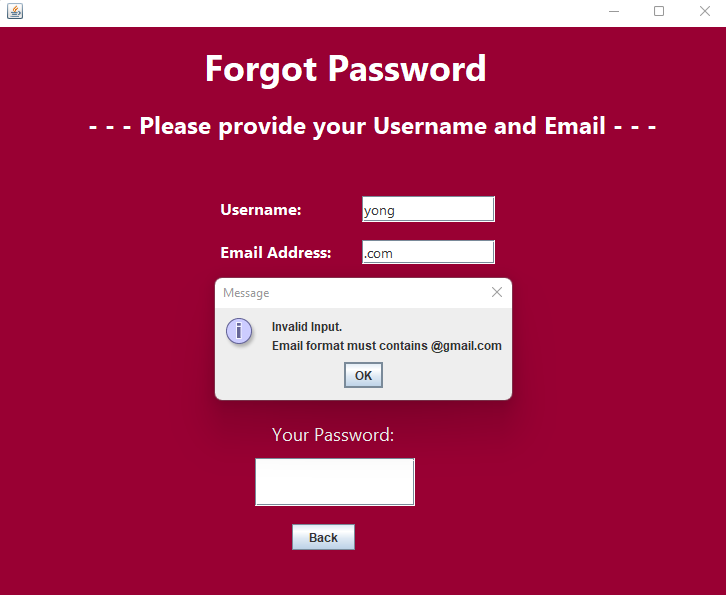
*Figure 3.3.1*

As the figure above shown, once the “Forgot Password” button was pressed, a forgot password page will pop out. There is several information required for the customer to enter such as username and email.



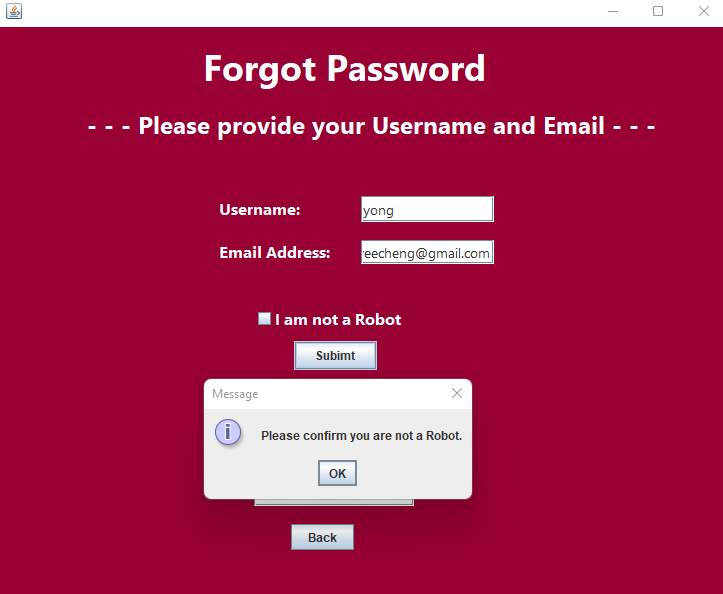
*Figure 3.3.2*

As illustrated in the figure above, if the username that the customer entered was not exist in the Car Rental System customer database, the system will ask the customer to re-enter.



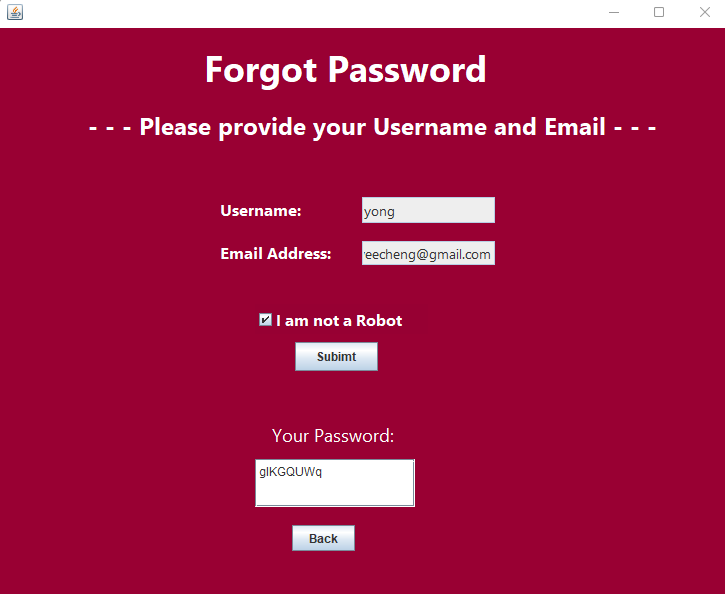
*Figure 3.3.3*

If the username was existed in the database but the email given was not in valid format, the system will still reject this input.



*Figure 3.3.4*

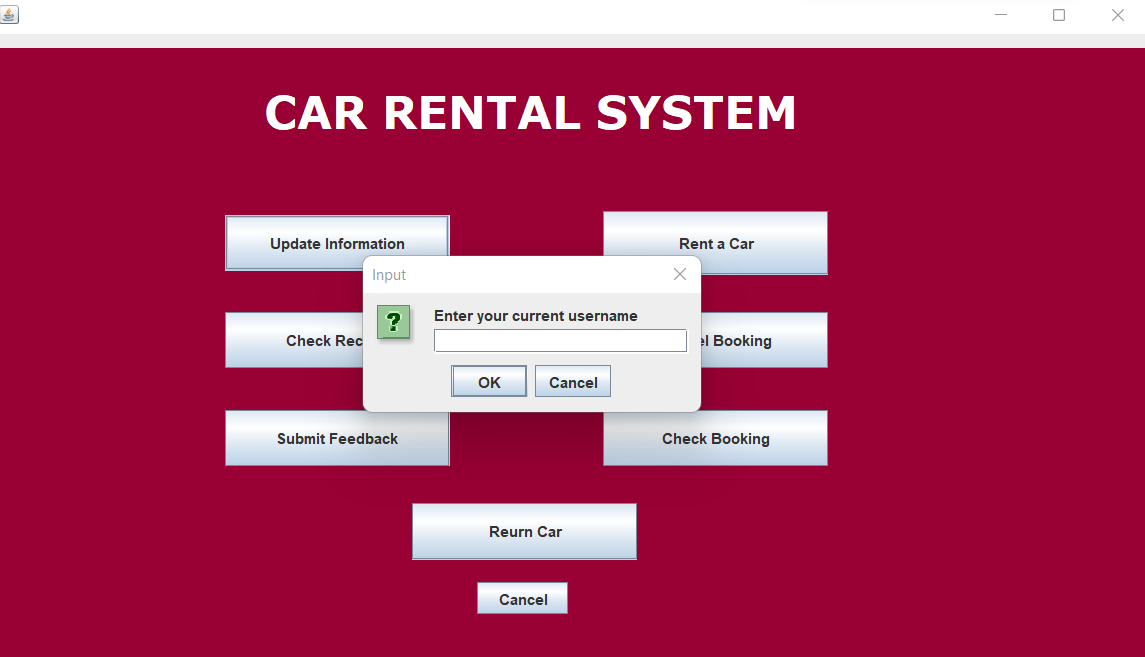
As the figure shown, if the username and email address was both entered correctly but the identity confirmation “I am not a Robot” click box was not ticked, the system will still reject and remind the customer to tick the button.



*Figure 3.3.5*

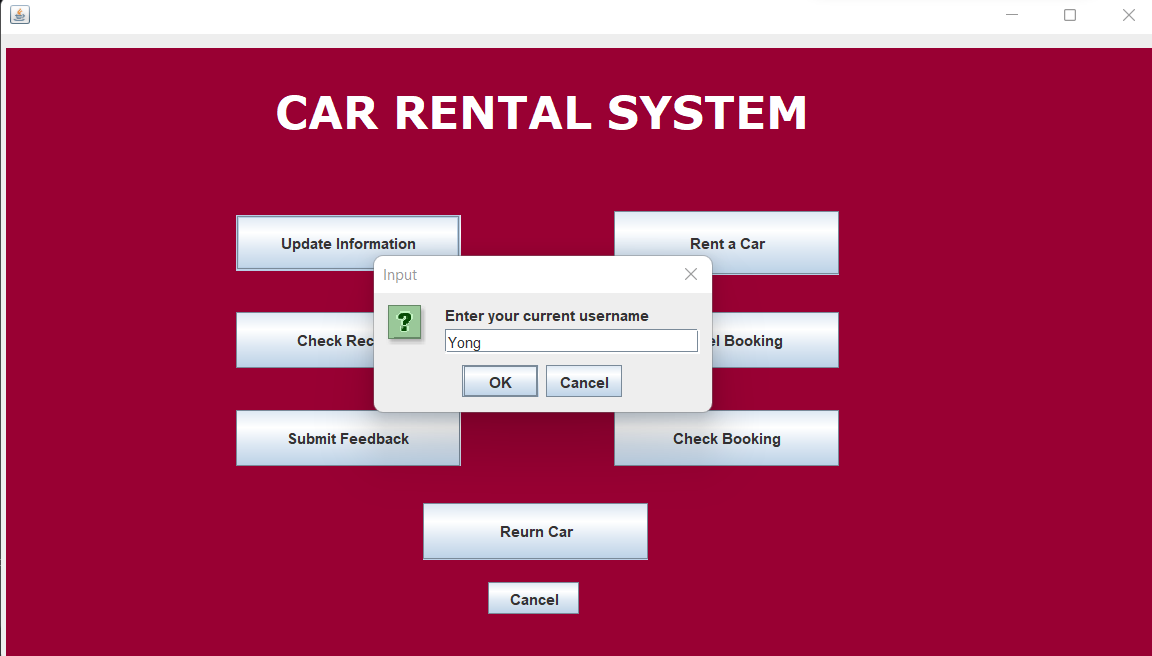
As illustrated in the figure above, if all necessary information was given, the “Submit” button can be clicked. After that a unique password will be generated for the customer to login to their account.

## 3.4 Update Customer’s Personal Information Part

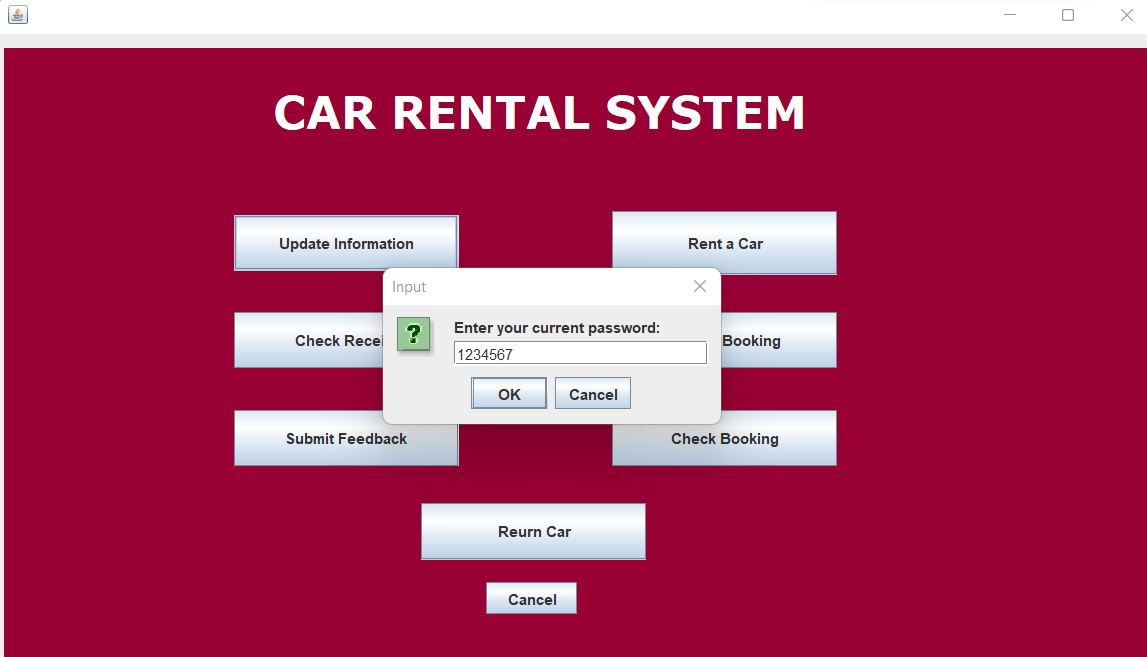


*Figure 3.4.1*

After the customer clicked on the “Update Information” button, the system will ask he/she to input his/her current username and password for identity confirmation.

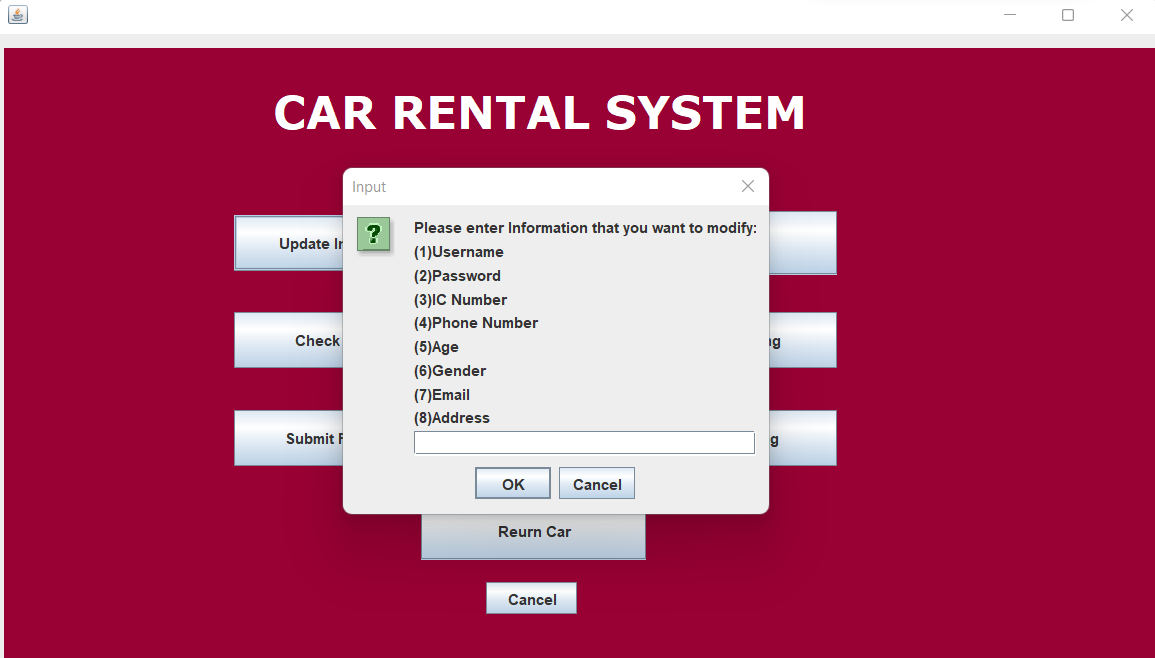


*Figure 3.4.2*



*Figure 3.4.3*

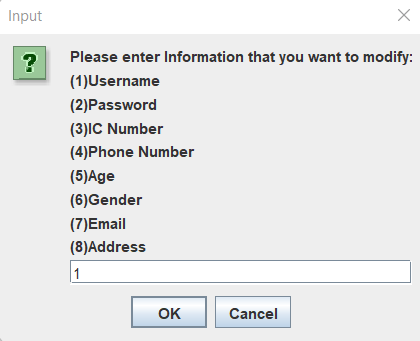
Notice that same validation as registration part will be made which input incorrect username and password will get rejected by the system.



*Figure 3.4.4*

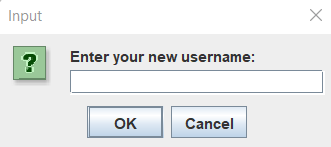
After the correct username and password have been entered, the client is then prompted to enter the option they wish to edit in a window that displays the options of the data that may be modified

**Update Username**

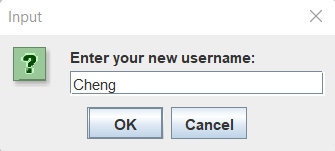


*Figure 3.4.5*

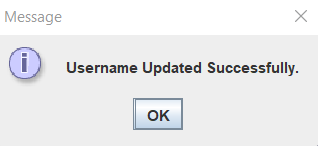
If the customer wanted to update their username, they may enter value “1”, then the system will ask the customer to input a new username.



*Figure 3.4.6*



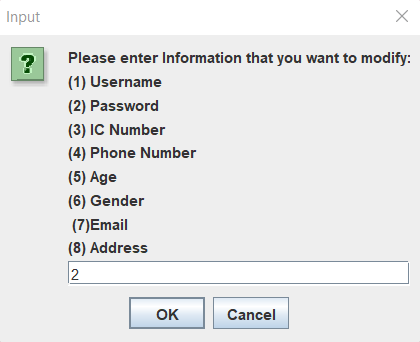
*Figure 3.4.7*



*Figure 3.4.8*

After a new username “Cheng” has been entered the system will output message “Username Updated Successfully”. Notice that all validation when updating customer’s personal details are same as customer registration part.

**Update Password**



*Figure 3.4.9*

If the customer wanted to modify their password, they will need to entered “2”.



*Figure 3.4.10*

After that a new password will require the customer to entered, in this scenario the new password will be “0206155”.

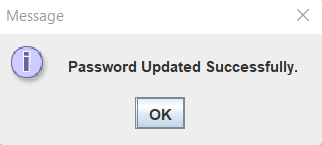
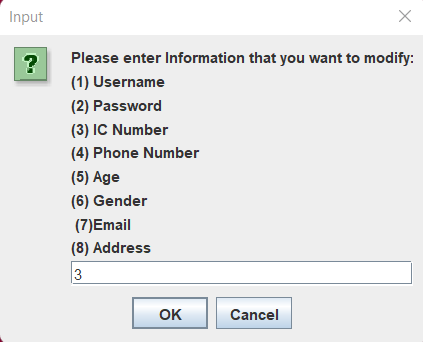


Figure 3.4.11

Finally, the system will inform customer the password has been modified.

**Update IC Number**



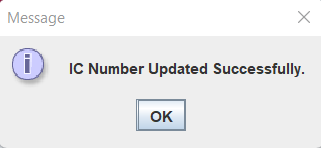
*Figure 3.4.12*

If the customer wanted to modify their IC Number, they will need to entered “3”.



*Figure 3.4.13*

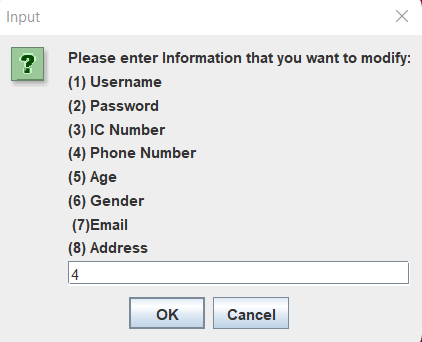
A new IC Number “020615040447” will be entered.



*Figure 3.4.14*

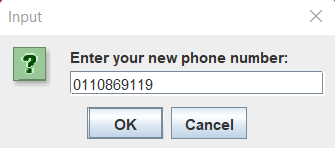
Lastly, the system will output the message to inform customer that the IC Number has been changed.

**Update Phone Number**



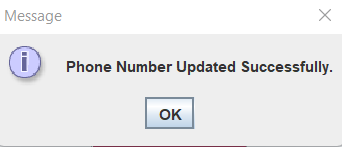
*Figure 3.4.15*

If the customer wanted to modify their IC Number, they will need to entered “4”.



*Figure 3.4.16*

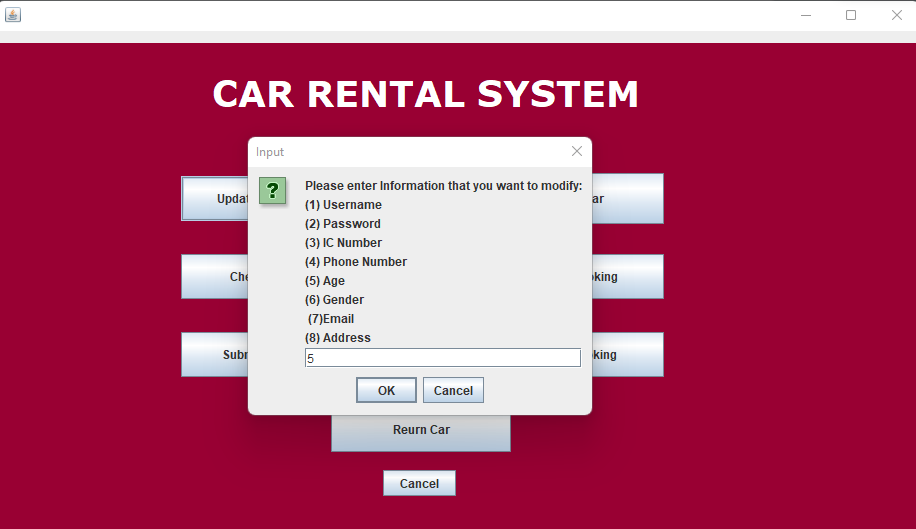
A new Phone Number “0110869119” will be entered.



*Figure 3.4.17*

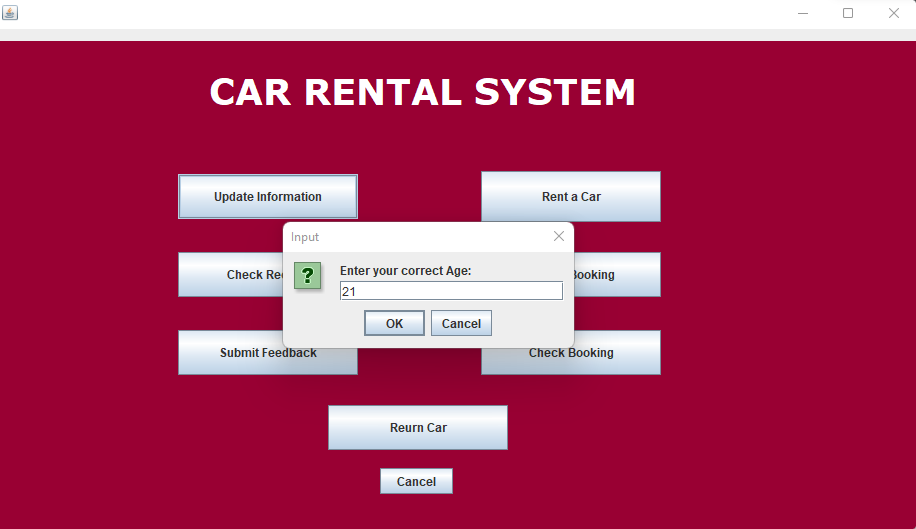
Lastly, the system will output the message to inform customer that the Phone Number has been changed.

**Update Age**



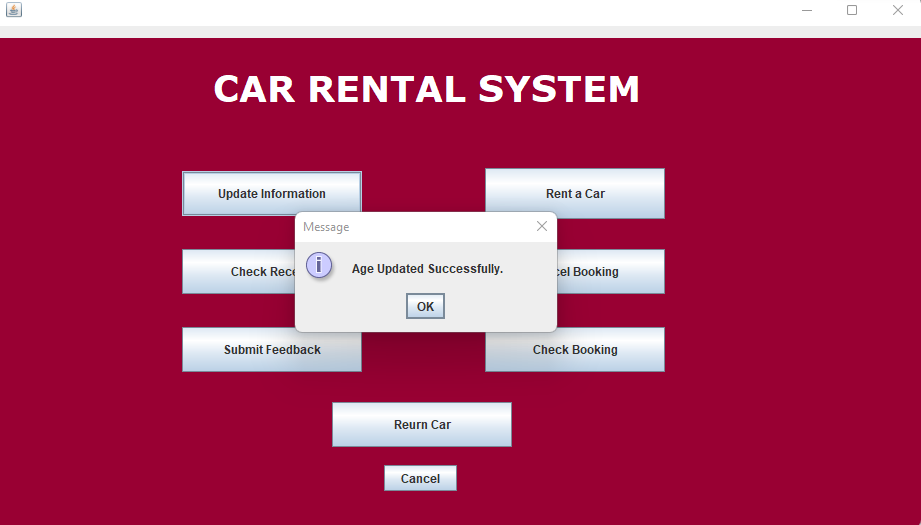
*Figure 3.4.18*

Next, if the customer would like to update his/her age, the value “5” must be entered.



*Figure 3.4.19*

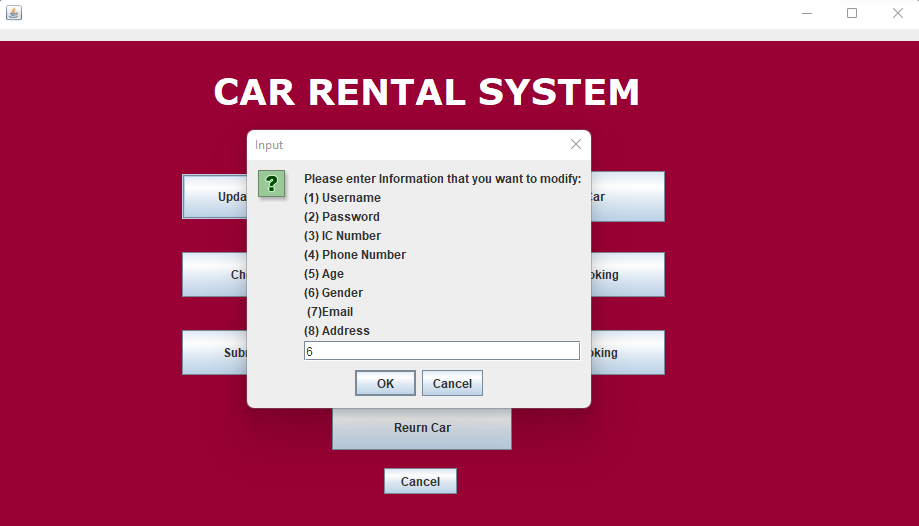
In this case, the age “21” will be entered.



*Figure 3.4.20*

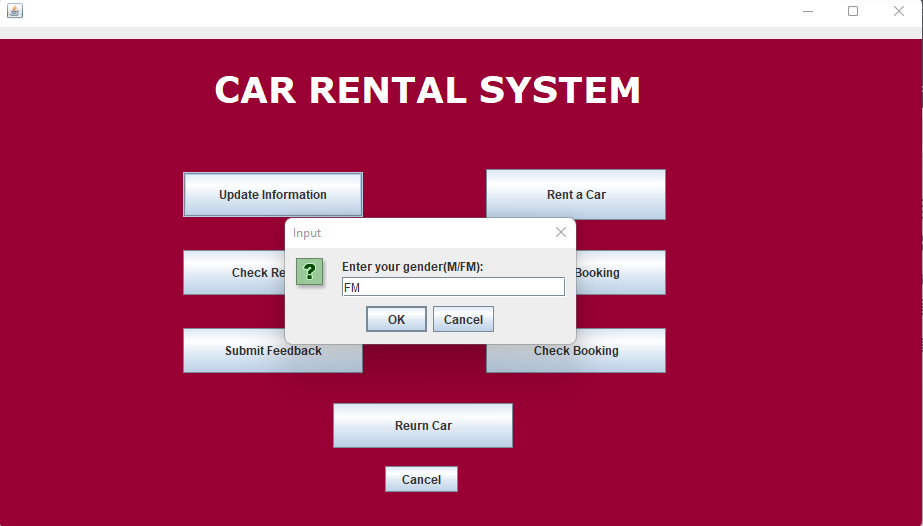
Finally, the system will output a message to inform customer that age has been updated from 20 to 21.

**Update Gender**



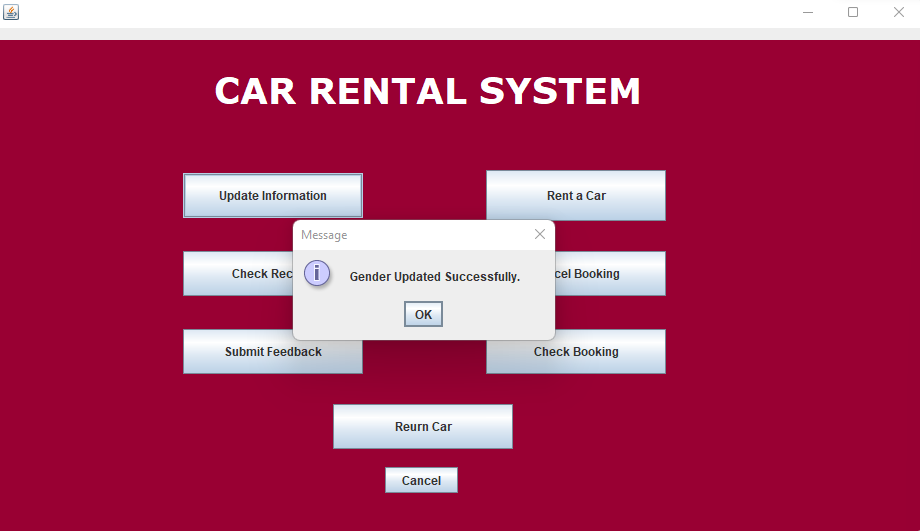
*Figure 3.4.21*

As the figure above shown, if customer would like to update his gender information, the value “6” must be input.



*Figure 3.4.22*

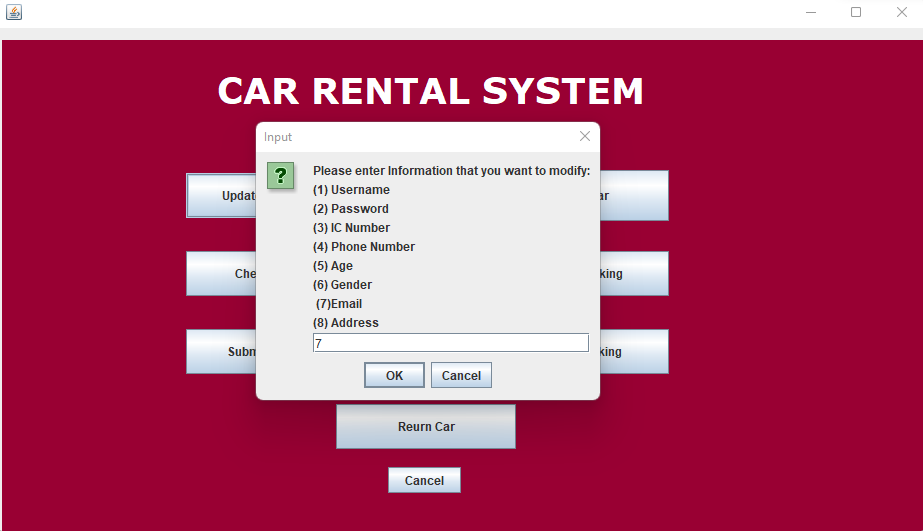
In this case, the customer input “FM”, to update the incorrect previous gender.



*Figure 3.4.23*

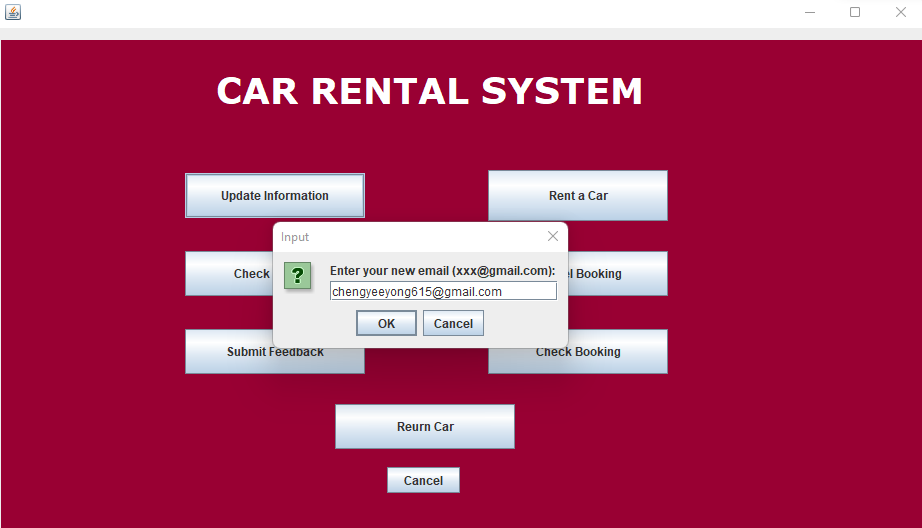
Finally, the message of gender has been successfully updated will be displayed.

**Update Email Address**



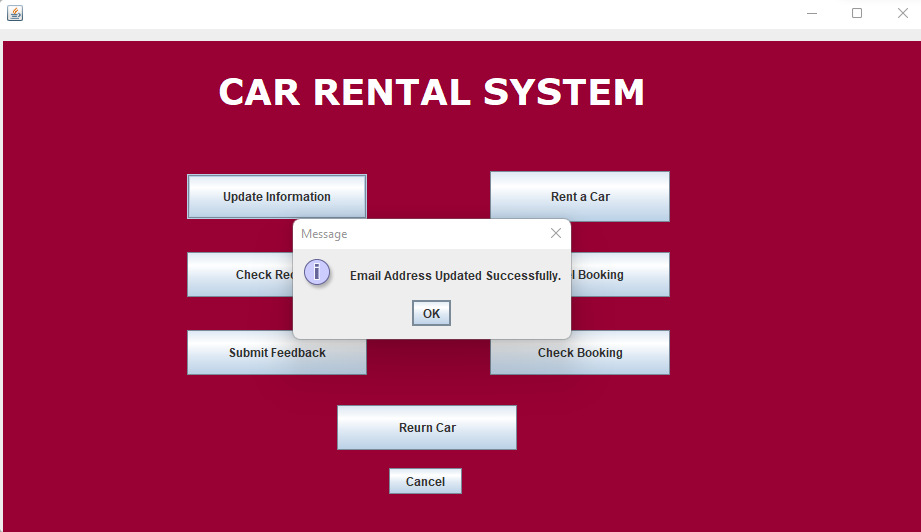
*Figure 3.4.24*

As figure above shows, if the customer wanted to modify their email address, the value “7” must be input.



*Figure 3.4.25*

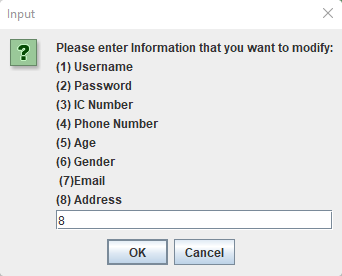
The email address “chengyeyong615@gmail.com” was entered into the system.



*Figure 3.4.26*

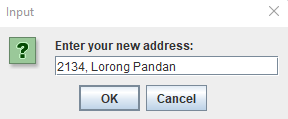
Finally, the email address was updated.

**Update Home Address**



*Figure 3.4.27*

As the figure above shown, the value “8” was entered to modify the home address.



*Figure 3.4.28*

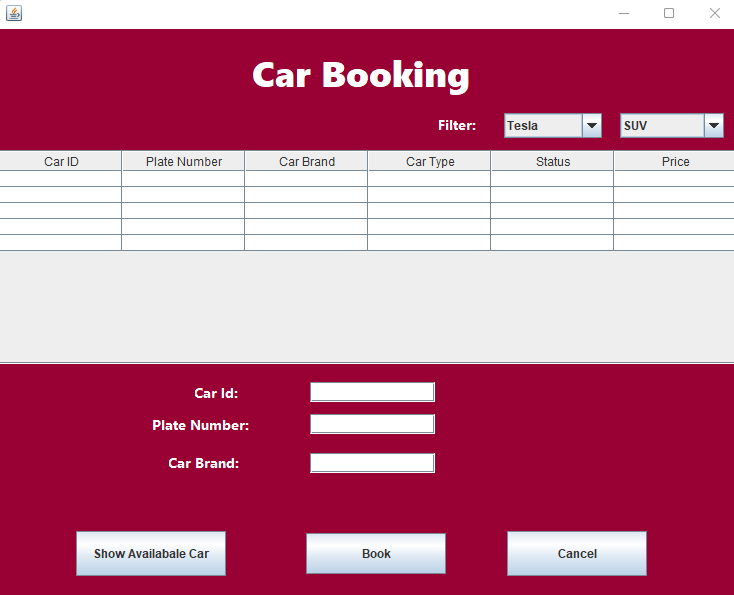
The system will ask customer to entered new home address.



*Figure 3.4.29*

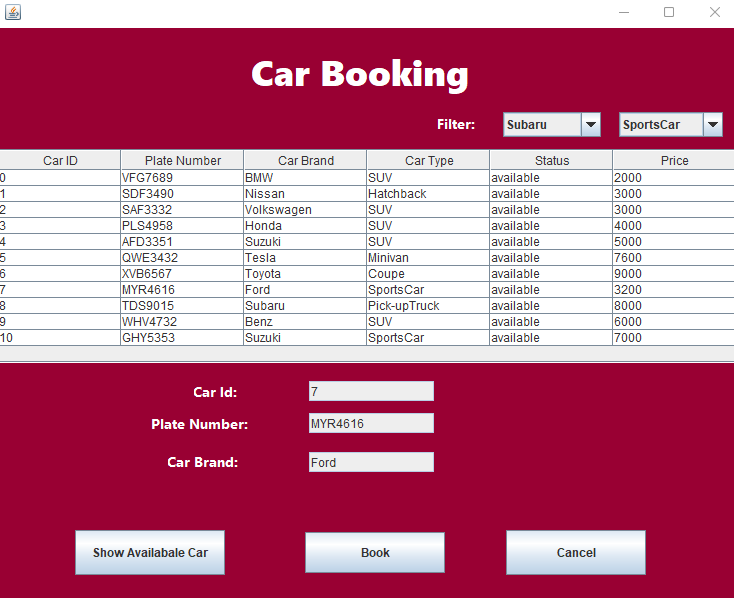
After the new address has been entered, the system will output the message to inform the customer that address has been modified.

## 3.5 Car Renting Page



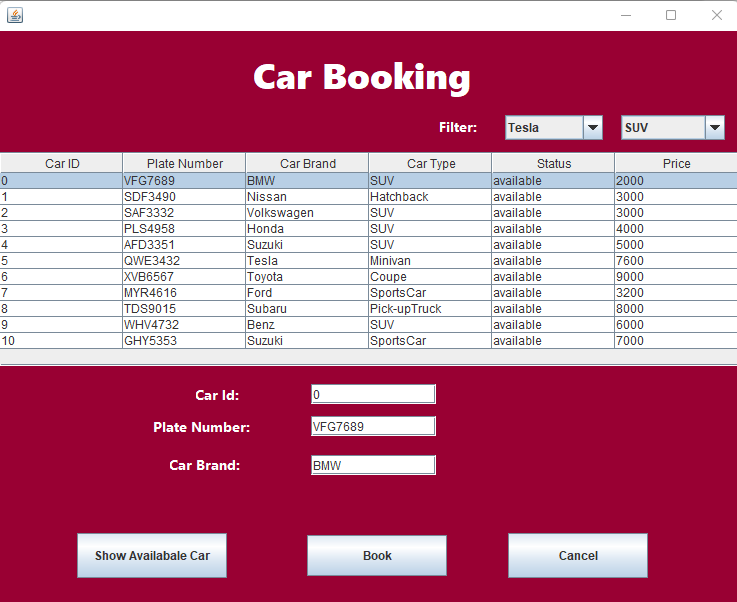
*Figure 3.5.1*

After customer clicked on the “Rent a Car” button on customer main menu. As figure above shown, the renting page will be pop out. The table in the middle of the page will first appeared to be empty.

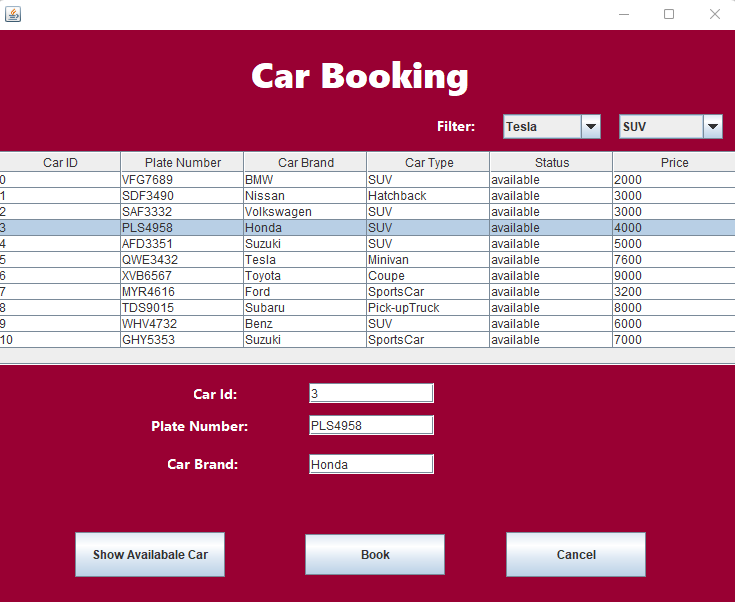


*Figure 3.5.2*

Customers is required to click on the button “Show Available Car”. After clicked it, cars that are available for rent will be display on the table as figure above shown.

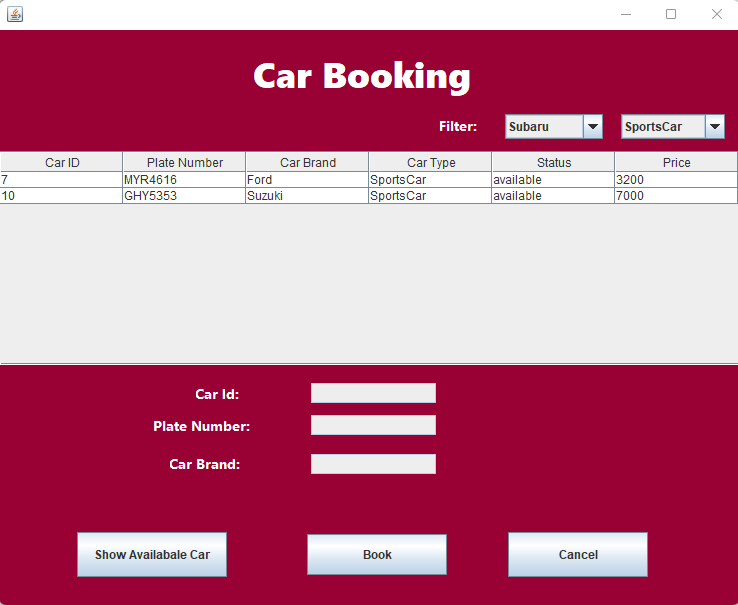


*Figure 3.5.3*



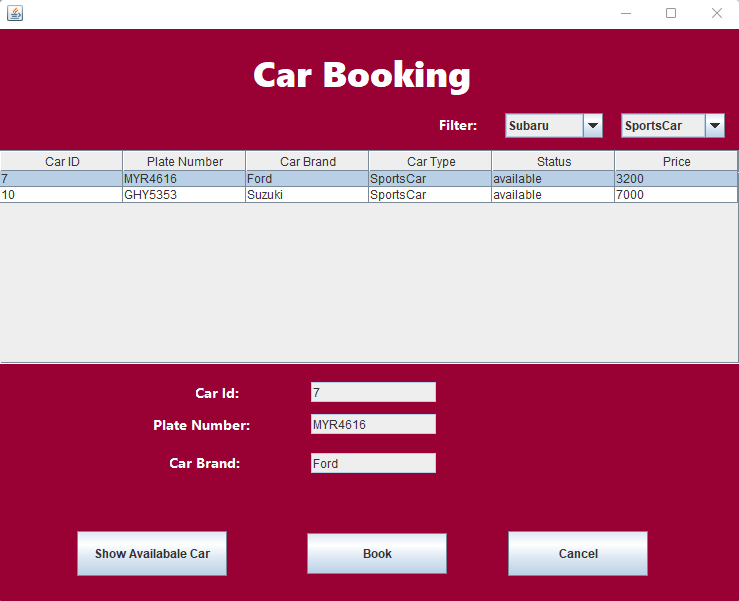
*Figure 3.5.4*

As illustrated in the pictures above, when a client clicks on any of the rows in the table that display a car, the information about that car appears in the car id, plate number, and car brand text fields above. This auto-filling features was brought convenience for customers which they no need to type themselves. However, notice that the text fields are uneditable for customers.



*Figure 3.5.5*

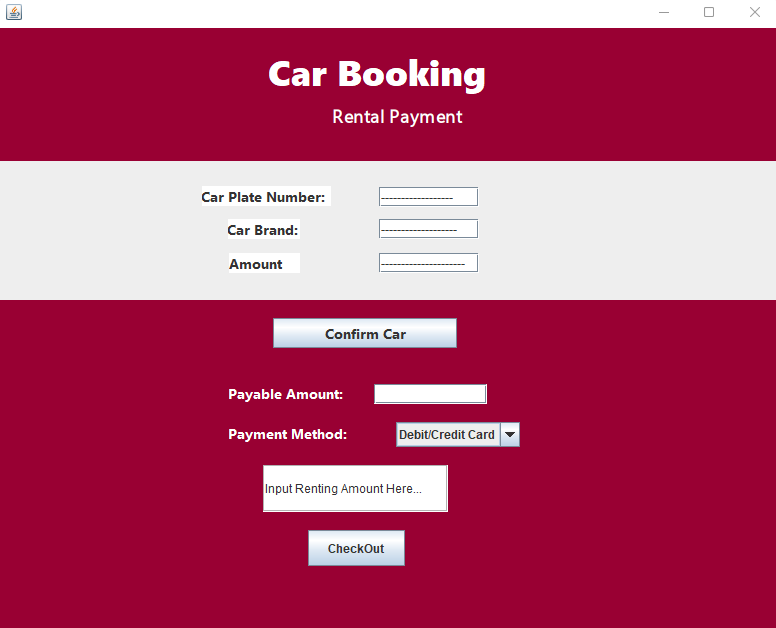
Notice that there will be a filter extra feature to view available car under conditions such as Car Type and Car Brand. In this case, the customer trying to filter with Subaru that are a sport car.



*Figure 3.5.6*

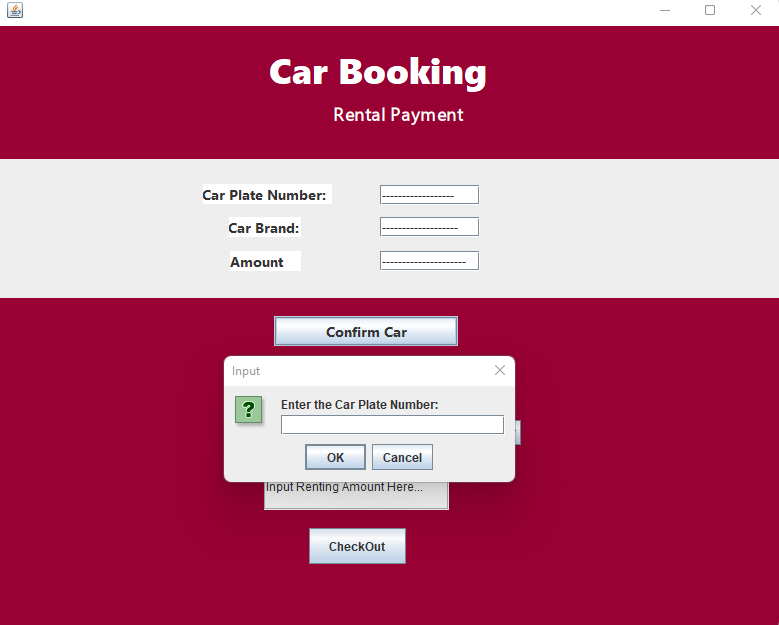
After that, the customer will choose a car that car id is “7”, plate number is “MYR4616” and car brand is Ford. At the end, clicked on “Book” Button.

## 3.6 Rental Payment Page



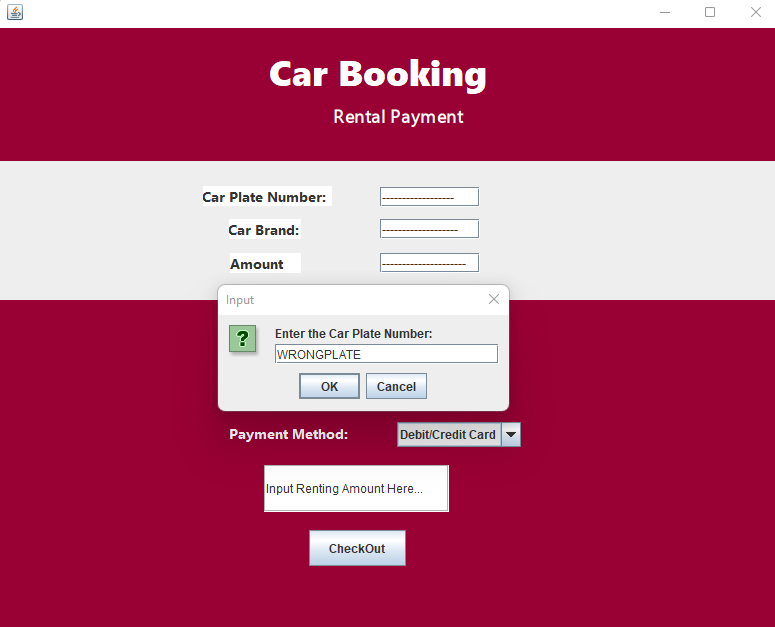
*Figure 3.6.1*

After the customer clicked the “Book” button, as figure above shown, the payment page will pop out and asked the customer to make payment.



*Figure 3.6.2*

As figure above shown, the customer must be clicked on the “Confirm Car” button to double confirm the car that wanted to rent. Notice that system will ask the customer to re-enter the car plate number of the car which customer are required to refer from the table of booking page.



*Figure 3.6.3*

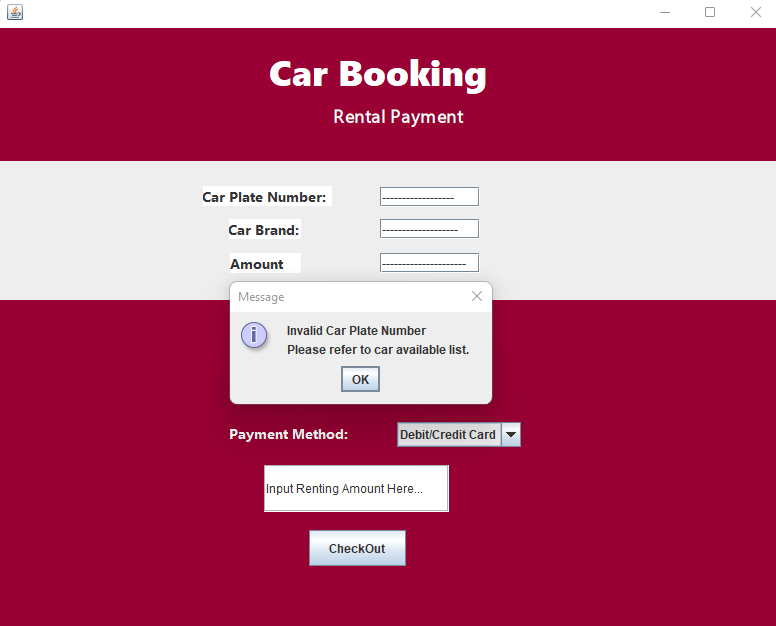
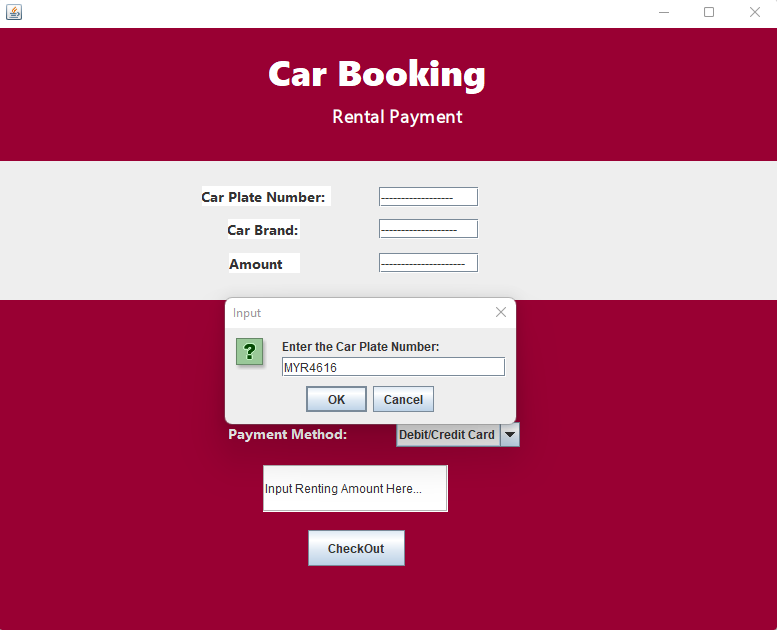
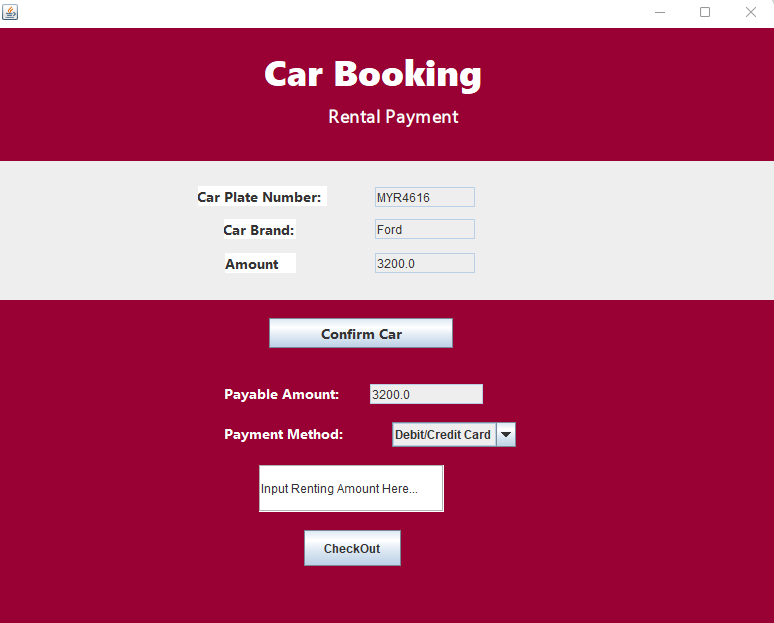


Figure 3.6.4

As figure above shows, notice that if the customer entered wrong plate number, the system will output a message to reject the input and ask customer to re-enter.



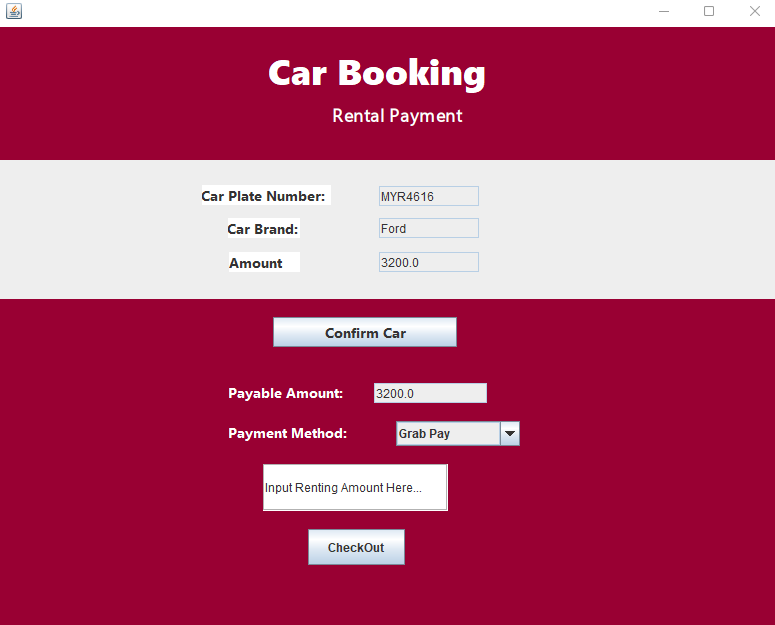
*Figure 3.6.5*



*Figure 3.6.6*

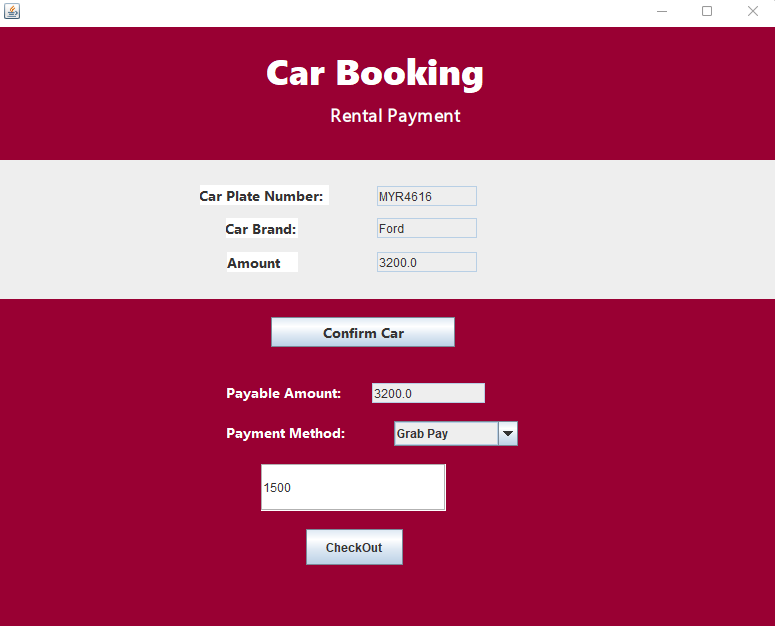
After the correct plate number “MYR4616” has been entered by the customer in figure. The information such as car plate number, car brand and car renting amount will be auto-filled into the form that located in the middle of the page.

The payable amount will be display below for the customer to enter the payment amount.



*Figure 3.6.7*

As figure above shown, the customer can choose their payment method by using the combo box under the payable amount label. In this case, the customer chose “Grab Pay” method.



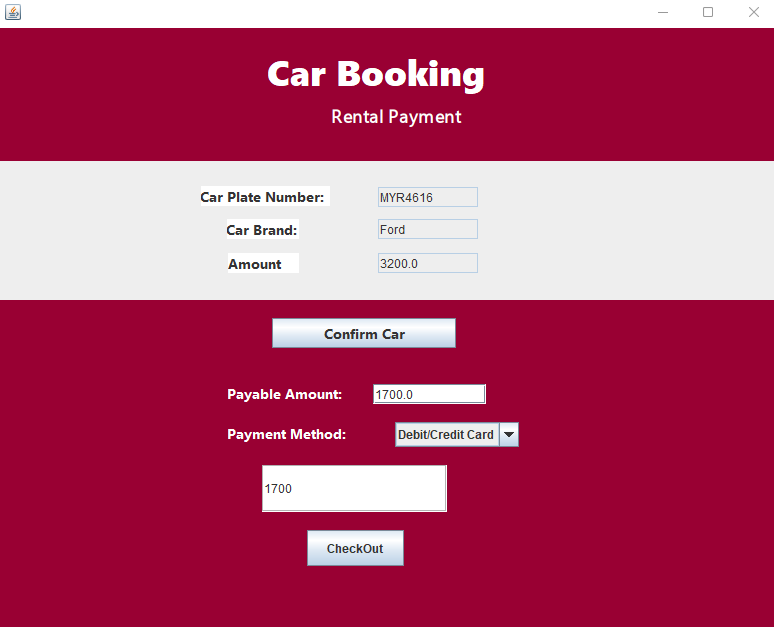
*Figure 3.6.8*

As the figure above indicates, after the method has been chosen, the system will require customer to enter rental amount which the amount must match with the payable amount.



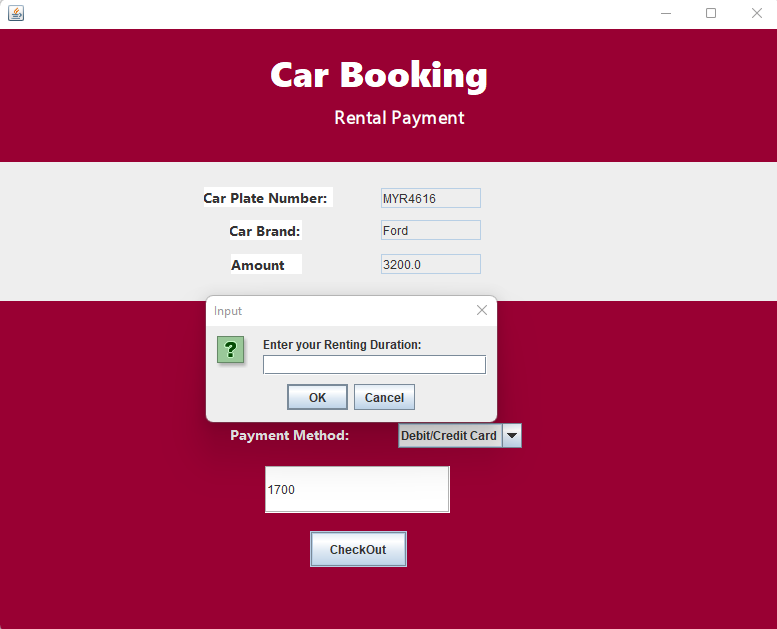
*Figure 3.6.9*

As the figure display, if the customer entered an unmatched amount such as insufficient amount “RM1500.00” compared to payable amount “RM3200”. After that the payable amount will change to “RM1700” which are the RM3200 – RM1500 = RM1700.



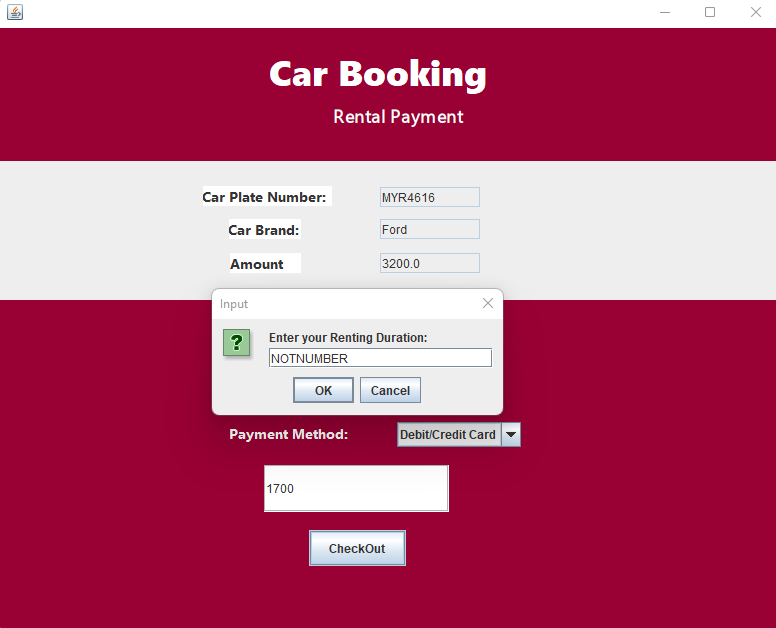
*Figure 3.6.10*

Next the customer will enter RM1700 as rental amount to match the payable amount. Finally, click on “CheckOut” button to submit the payment.

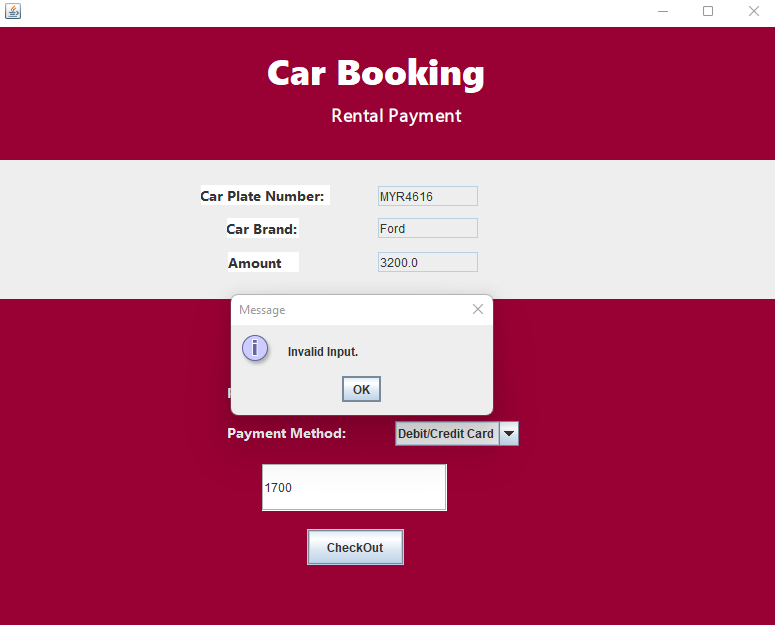


*Figure 3.6.11*

As figure above shown, after the customer clicked “CheckOut” button and the payable amount matched with the input rental amount, the system will ask customer to input renting duration.



*Figure 3.6.12*



*Figure 3.6.13*

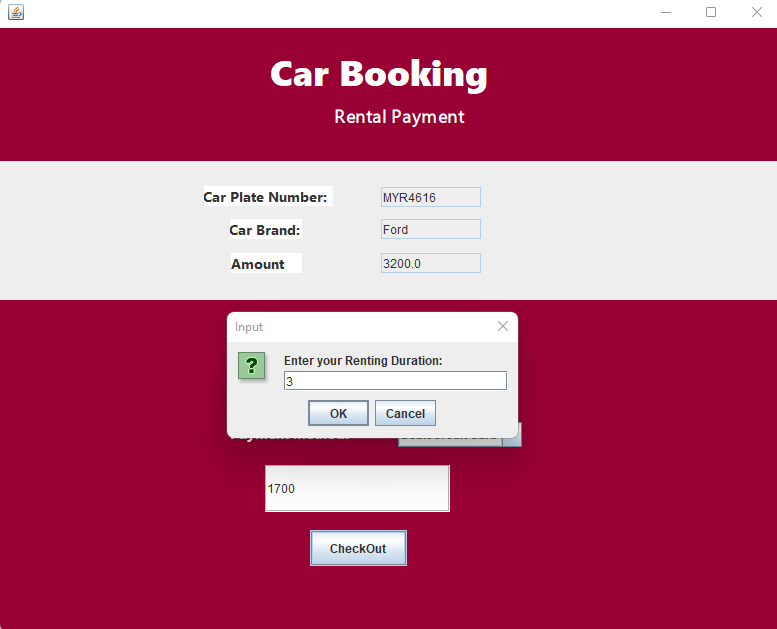
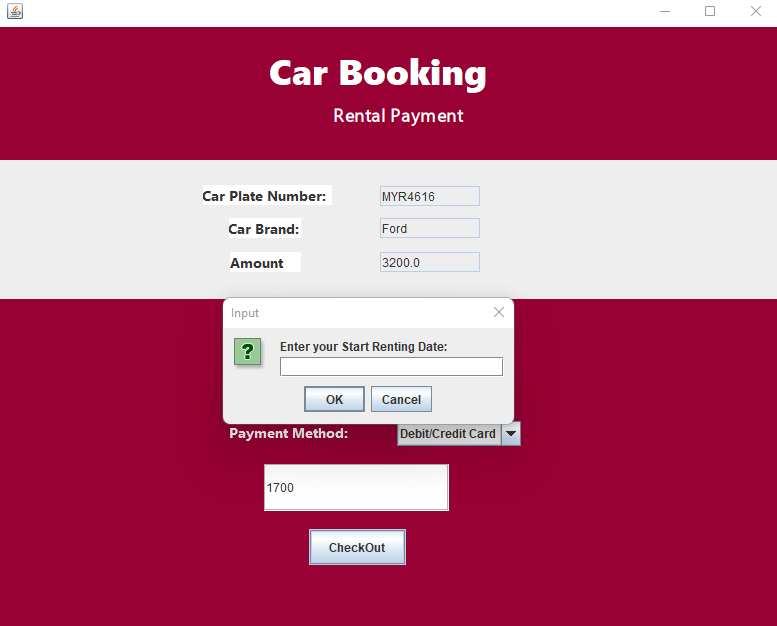


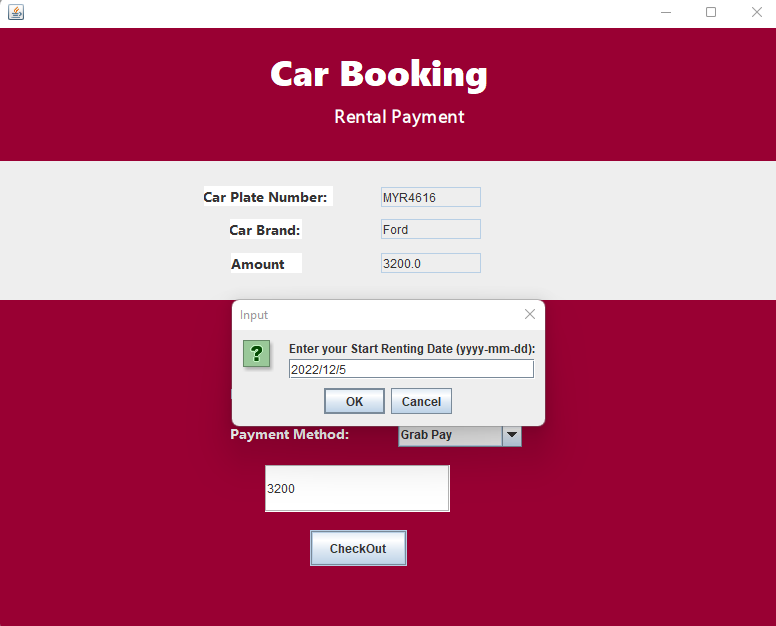
Figure 3.6.14

As figure and figure above shows, if the customer entered a value that is not a digit. The system will reject the input and the customer need to re-enter. In this case, the customer entered value “3” which represent 3 hours into the system.

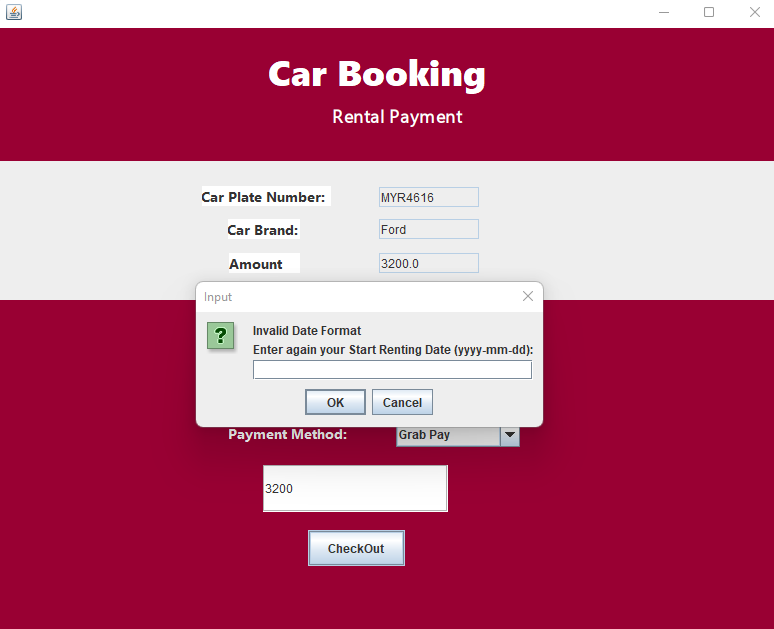


*Figure 3.6.15*

After input rental duration, the system will require customer to enter a start renting date.

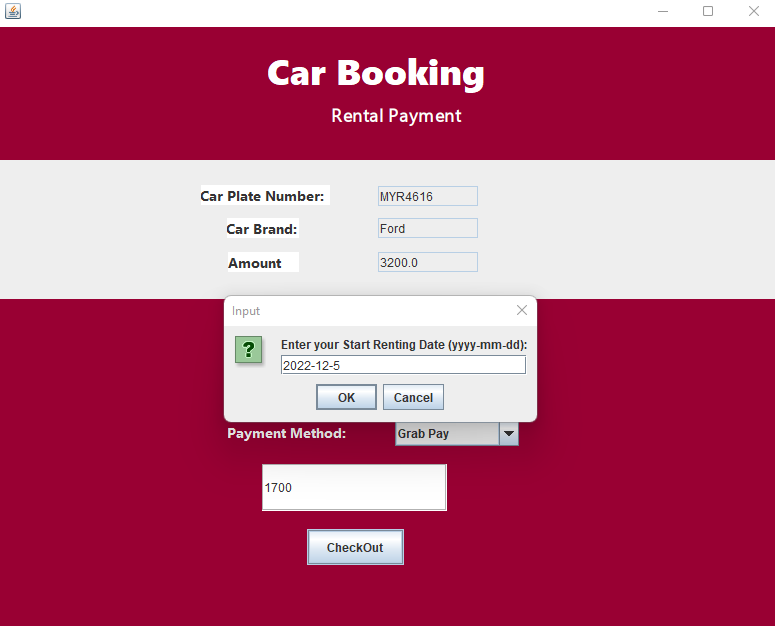


*Figure 3.6.16*



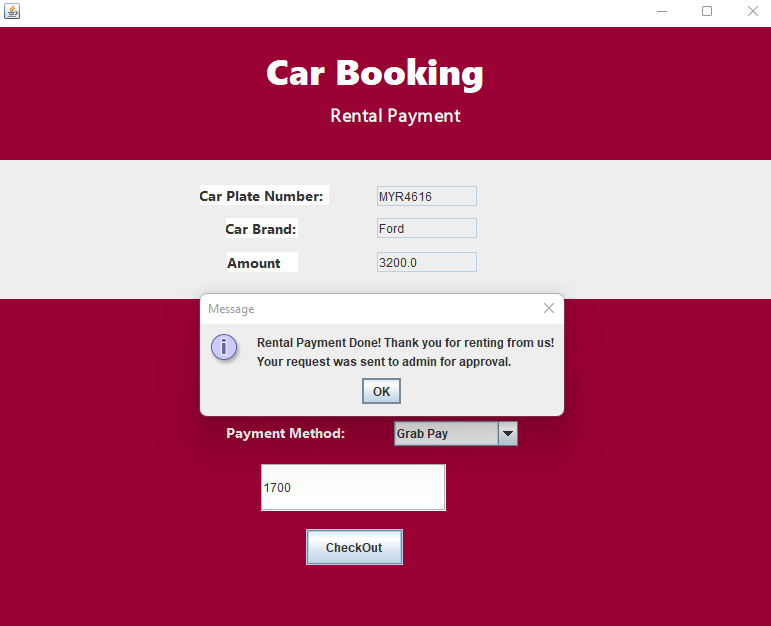
*Figure 3.6.17*

As the figure and figure above, if an invalid date format such as “2022/12/5”, the system will reject the input and ask the customer to input again.



*Figure 3.6.18*

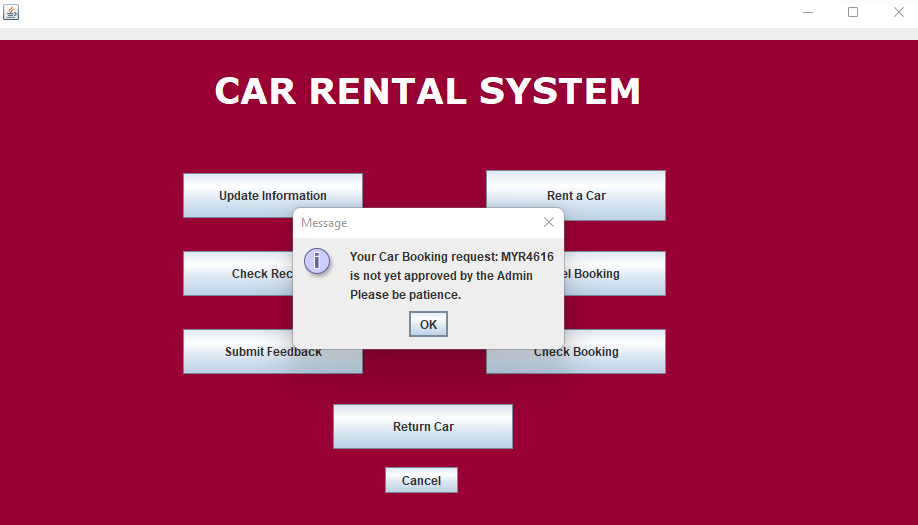
In this case, the customer will input a valid start date “2022-12-5”.



*Figure 3.6.19*

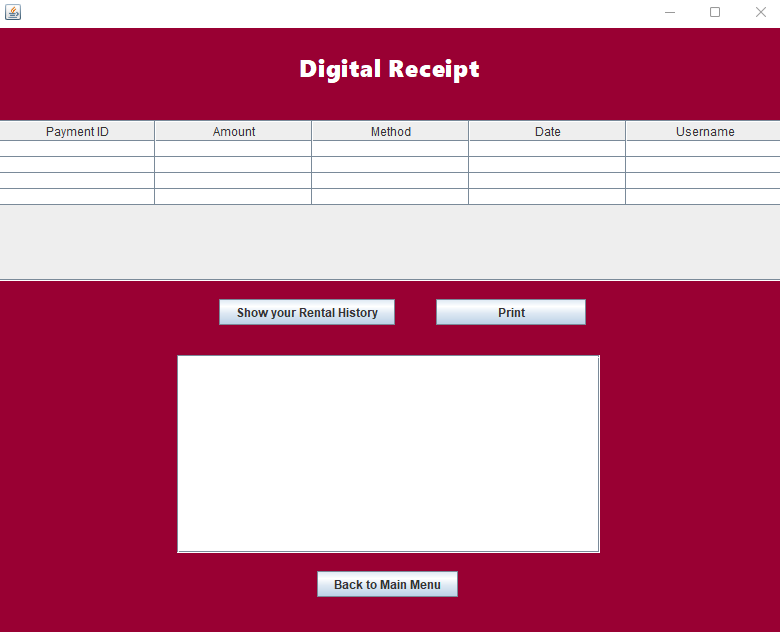
After the start date was entered, the system will output a message to inform customer that the rental payment was done and was sent for approval.

# 3.7 Check Receipt Feature



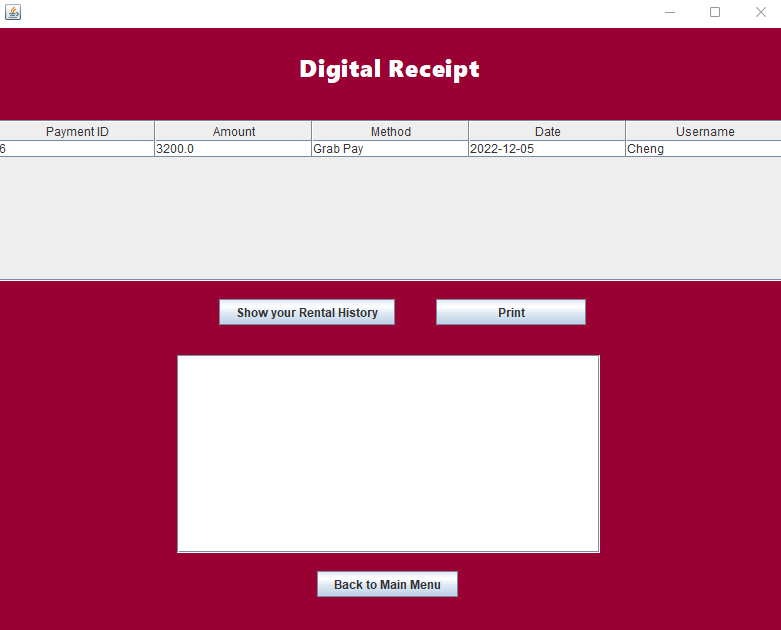
*Figure 3.7.1*

Notice that after a booking request was done, each time the customer login to the system, the notification of the booking status (approved/not yet approved) will pop out as figure above shown to inform the customer.



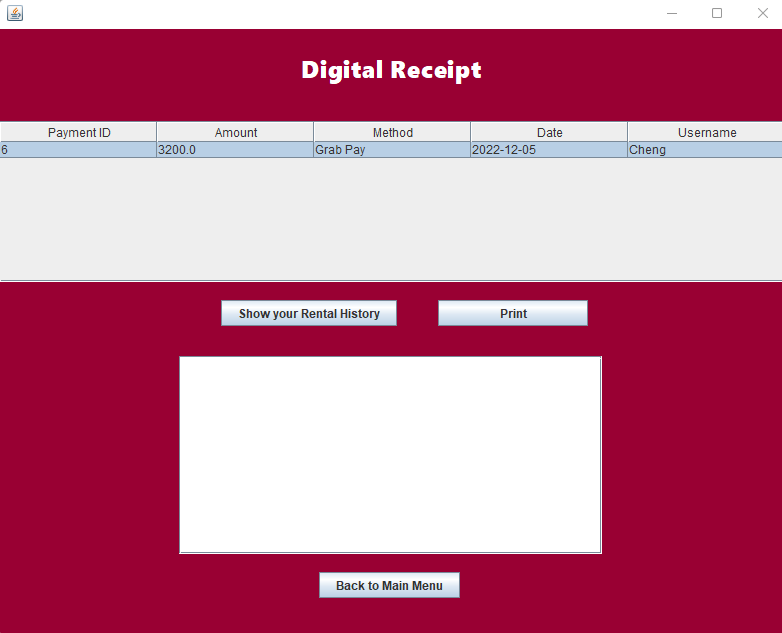
*Figure 3.7.2*

Now comes to the check receipt function, after the customer clicked on the “Check Receipt” button, a receipt page as figure above shown will pop out.



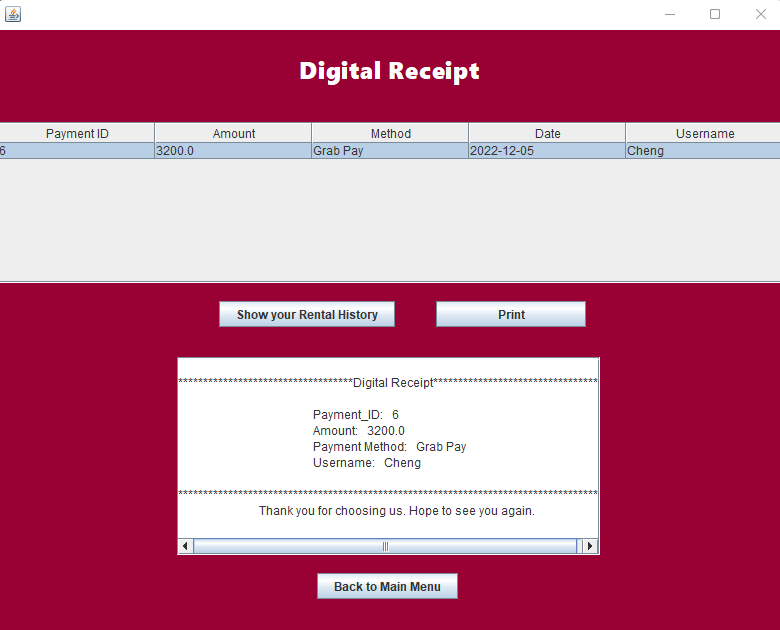
*Figure 3.7.3*

At first the table above will be empty; customer is required to click on the “Show your Rental History” button to show the rental payment history.



*Figure 3.7.4*

As figure above shown, after the history was shown, the customer may click on the rows of the specify payment to select it.



*Figure 3.7.5*

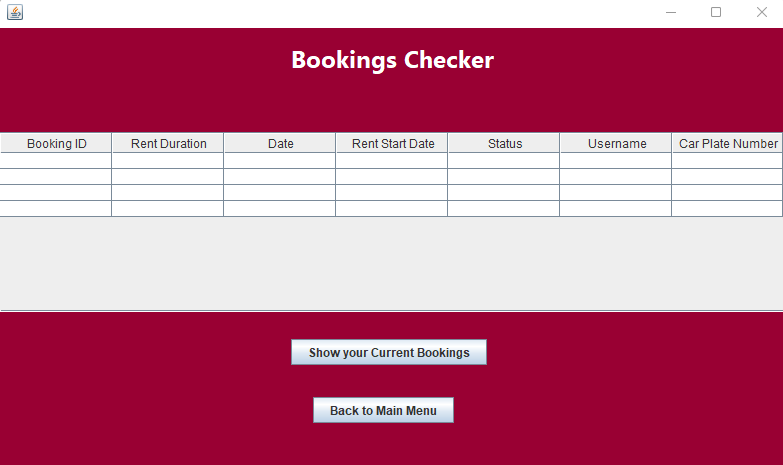
Following that, as seen by the customer above, after selecting a payment history and clicking on the print button, the receipt will be created under the buttons.



*Figure 3.7.6*

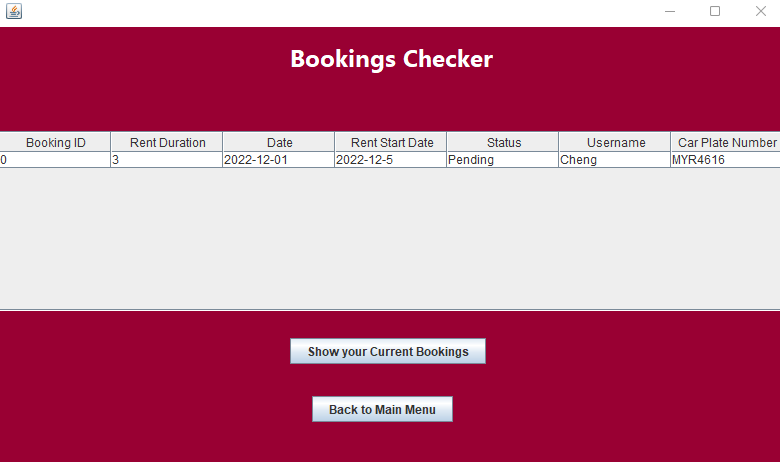
As the figure above indicated, notice that if the customer did not select any of the payment history inside the table and the “print” button was clicked, the system will pop out a message to inform customer that they are required to select from the table.

## 3.8 Check Booking Feature



*Figure 3.8.1*

As figure above shown, if the customer wants to check their booking details, the “Check Booking” button must be clicked, then a booking checker page will pop out.

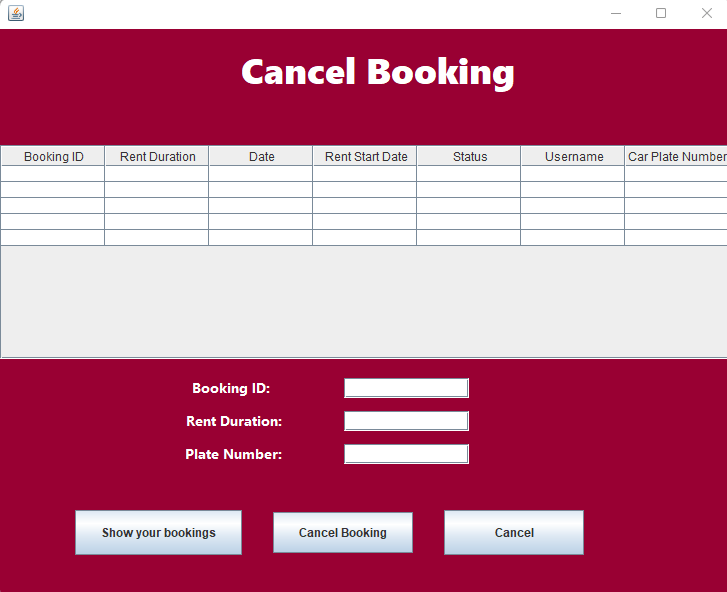


*Figure 3.8.2*

As seen in the picture above, the table will first be empty; however, after the client clicks on the "Show your Current Reservations" button, the bookings made by the customer will be displayed in it.

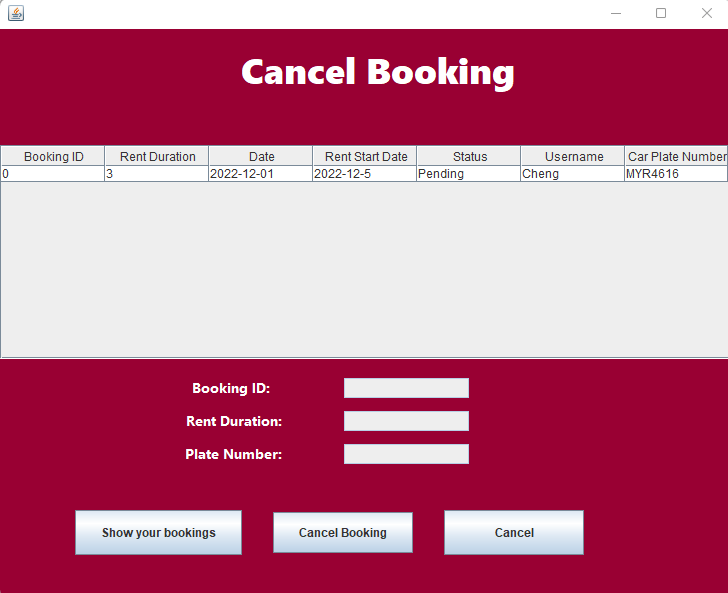
## **3.9 Cancel Booking**

Now comes to the cancel booking function, after done checking bookings, pressed “Back to Main Menu” to go back to customer main menu.



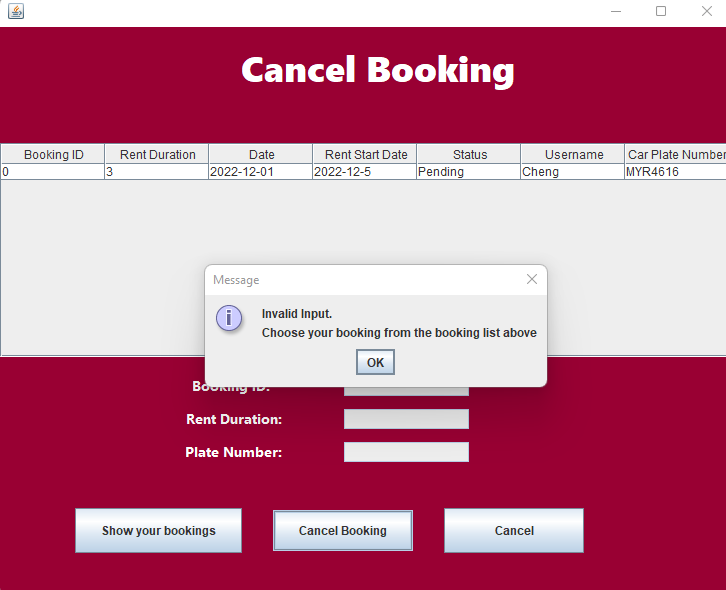
*Figure 3.9.1*

After the button “Cancel Booking” was pressed, a cancel booking page will pop out and the table that shows the bookings will remain empty at first.



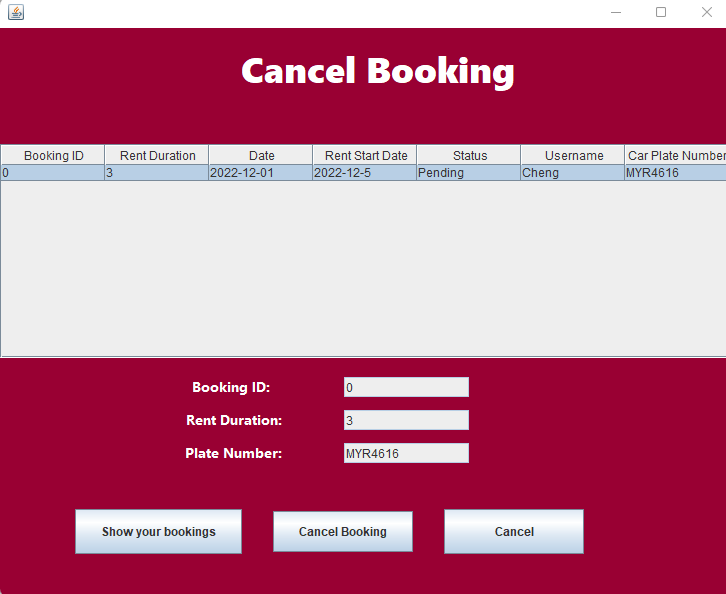
*Figure 3.9.2*

The system required customer to clicked on the “Show your bookings” button to show his/her pending bookings.



*Figure 3.9.3*

As the figure shown, if the customer didn’t select any of the booking and straight away initiate the “Cancel Booking” button, the system will display a message to ask customer to choose booking from above booking list.



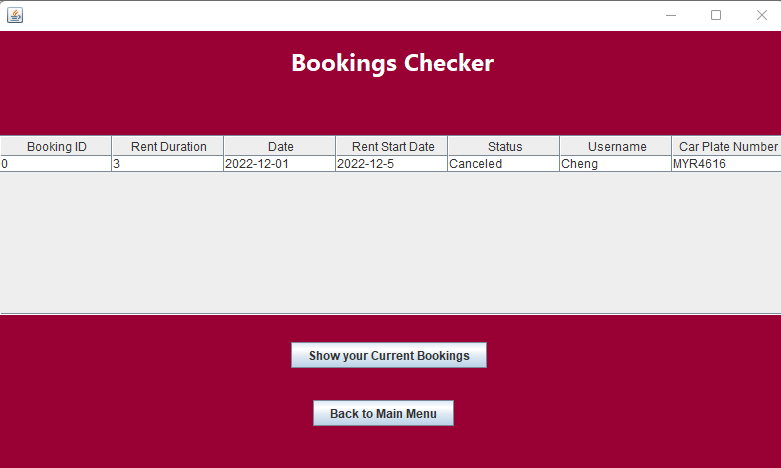
*Figure 3.9.4*

In this figure, the client selects the previous booking that he or she made, and the columns below are auto-filled with that booking's details.



*Figure 3.9.5*

After clicked on the “Cancel Booking”, a message that inform the customer the specify booking has been cancelled will be display and the refund will be received soon.

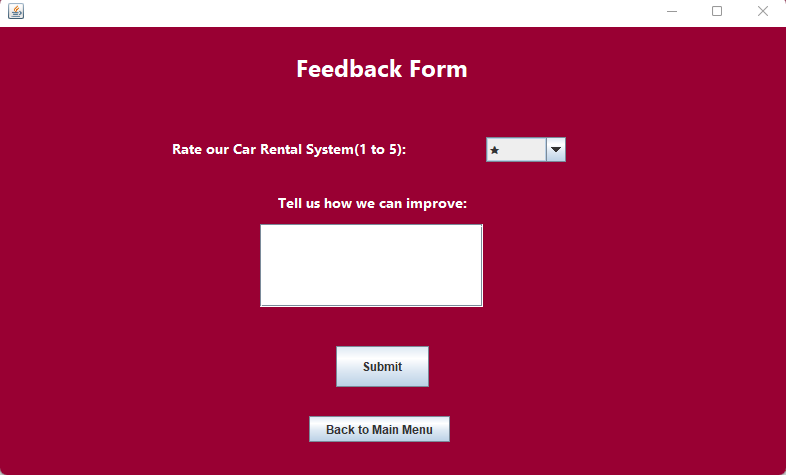


*Figure 3.9.6*

Finally, the result of the booking has been canceled can be viewed on the booking checker feature.

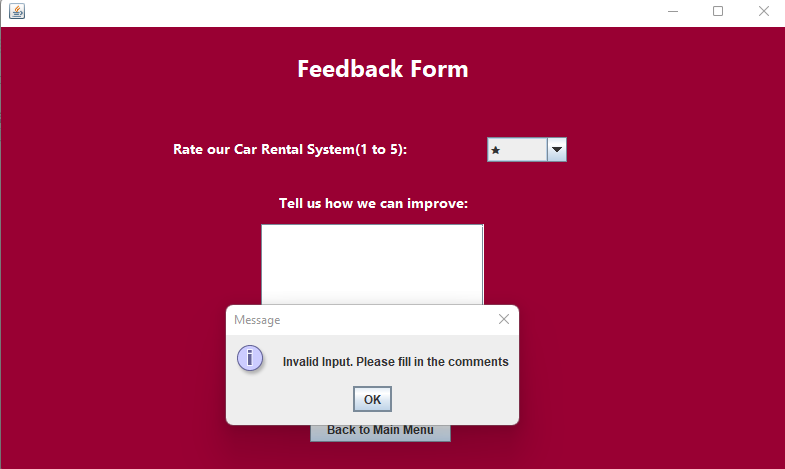
## 3.10 Submit Feedback

Now it came to the submit feedback feature, the “Back to Main Menu” button will be click to get back to the main menu.



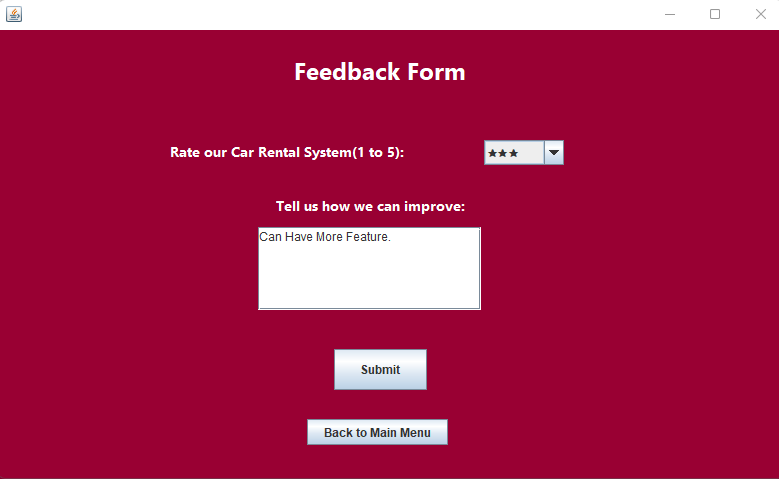
*Figure 3.10.1*

After clicking the “Submit Feedback Button” on the customer main menu, a feedback form will display as figure above shown.



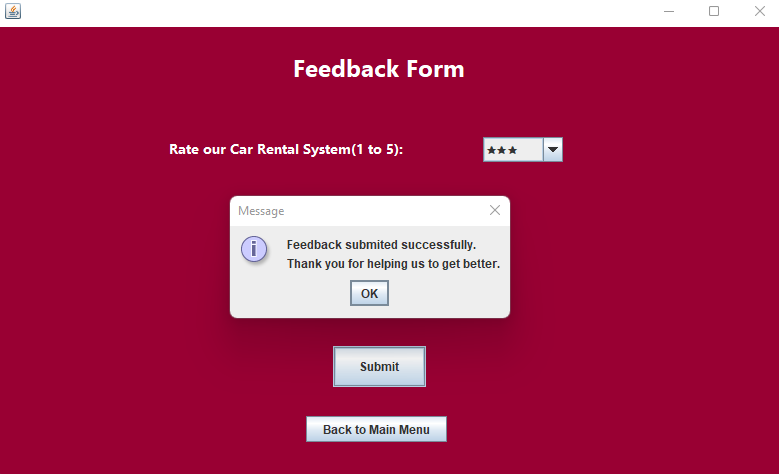
*Figure 3.10.2*

As illustrated in the graphic above, if the client did not fill out the comments and then pressed the "Submit" button, the system will display a message reminding the customer to do so first.



*Figure 3.10.3*

The customer is then asked to assess the car rental system on a scale of one to five stars. Also, some comments will need to fill in to submit the feedback form.

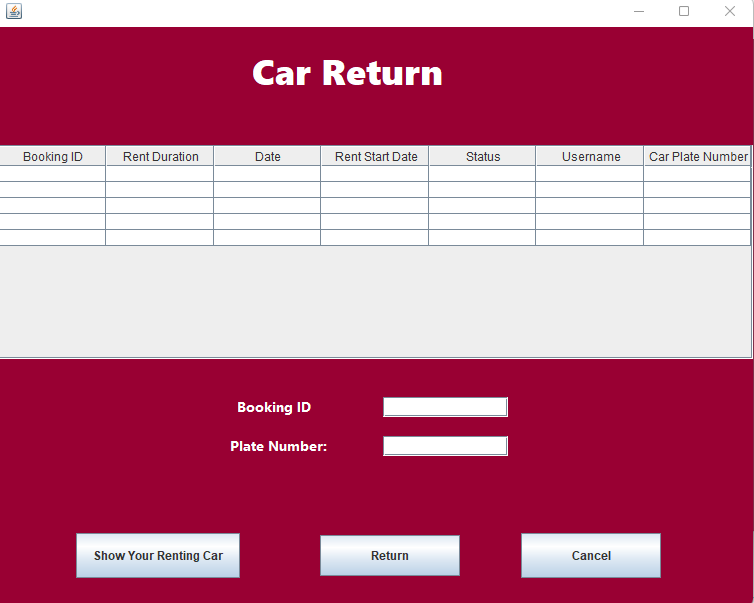


*Figure 3.10.4*

After entering the required information, the "Submit" button was pressed. Finally, a notification informing the customer that the feedback was successfully sent will be shown.

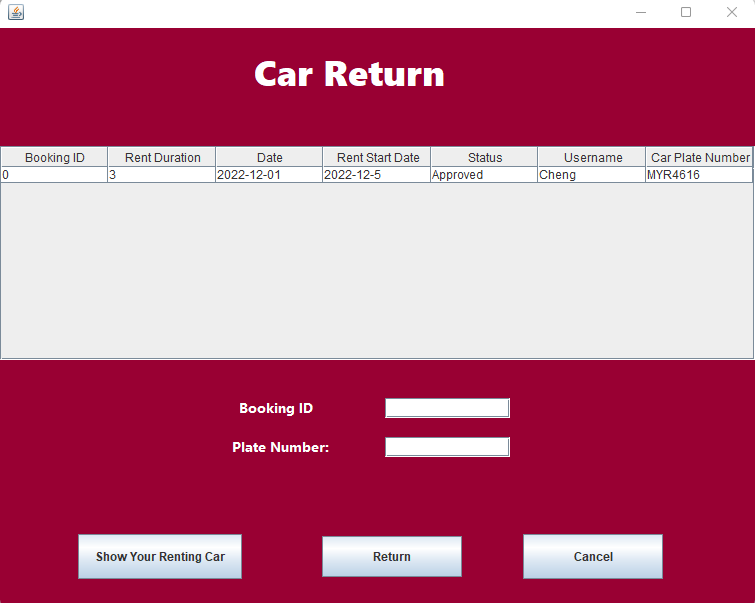
## 3.11 Return Car

Noticed that the return car function can only functionable when the booking request is already approved by the admin. After the customer rented the car, keep in mind that the car must be return on time.



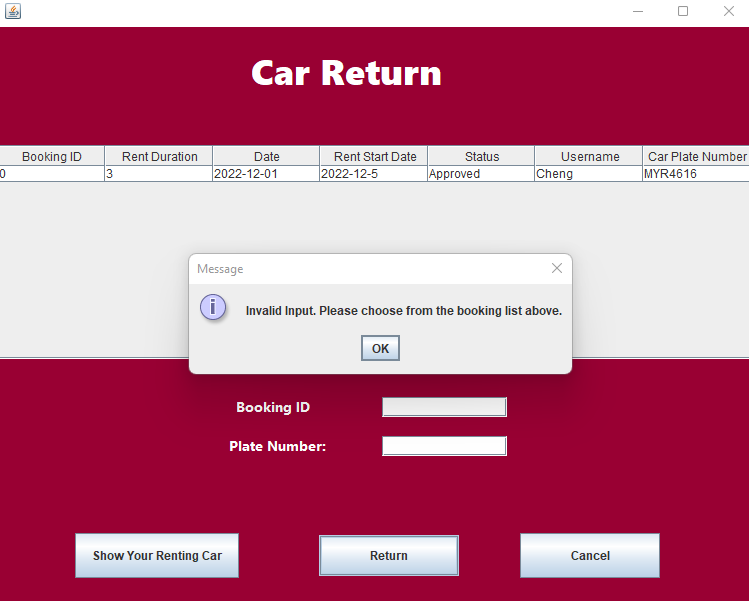
*Figure 3.11.1*

As figure above indicates, after the “Return Car” button on main menu was clicked, a return car page will display for the customer. Notice that the bookings list will remain empty at first.



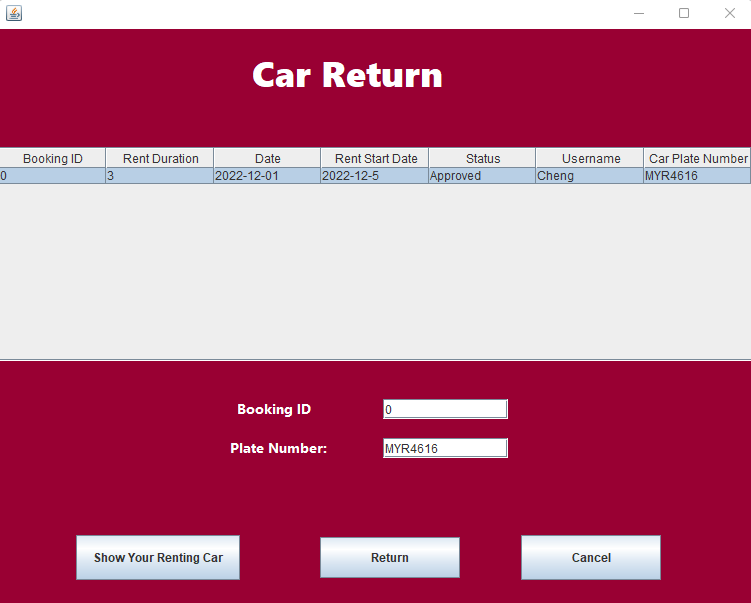
*Figure 3.11.2*

In this figure, the previous booking was approved by the admin and the car has been rented by the customer. After clicking the “Show Your Renting Car” button, the booking for car with plate number “MYR4616” will appeared on the list for the customer to choose.



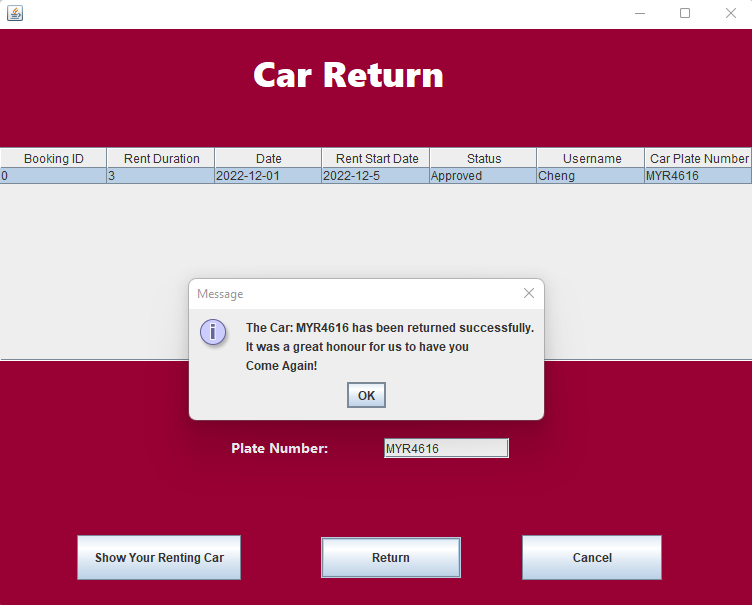
*Figure 3.11.3*

As the figure shown, if the customer did not select any of the booking from the booking list above, the system will proceed to remind the customer with a pop out message.



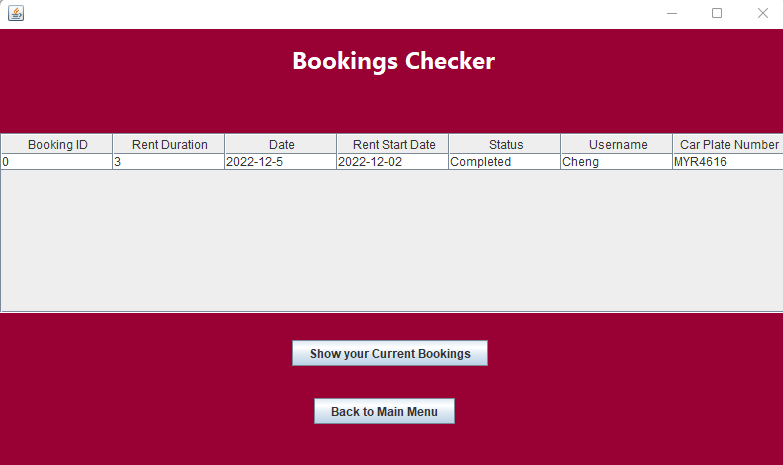
*Figure 3.11.4*

In this case, the booking for “MYR4616” car was selected and the information of this car was auto-fill in to the box above for verification purposes.



*Figure 3.11.5*

After the “Return” button was clicked, the return car procedure has been completed successfully, the system will also output a message to inform the customer and the system will go back to main menu.



*Figure 3.11.6*

As the figure shown, the result of the booking has been completed can be view by using booking checker feature.

# 4.0 Source codes

## 4.1 Encapsulation

Encapsulation, being one of the fundamental object-oriented programming is included in this car rental system. The definition of encapsulation is an action of combining data and functions that operate on that data into a single entity which is similar to a Java class. (Jansenn, 2022)

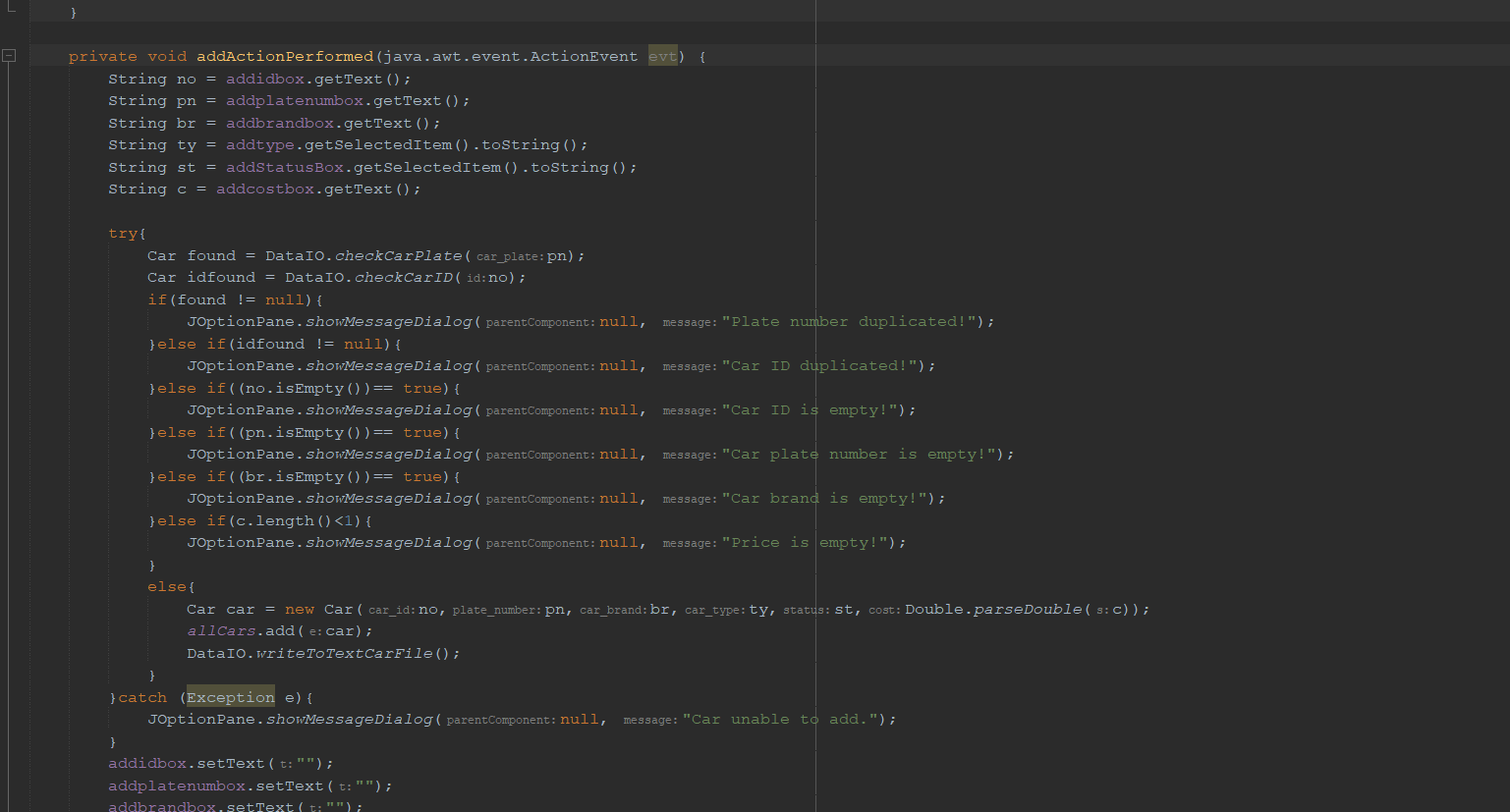
*Graphical user interface, text

Description automatically generated*

*Figure 4.1.1 Array list of objects*Text

Description automatically generated

*Figure 4.1.2 Writing array list into text file*



*Figure 4.1.3 Creating object*

*Text

Description automatically generated*

*Figure 4.1.4*

From the figure above, it shows the examples of encapsulation implemented in the car rental system. While the code in *figure* shows how data is collected from text fields and new object “car” is created from class Car. Then, the details of each car are created with variables that is declared as private in “Car” class as shown in *figure.* After that, the object is with each private attribute is then stored in array list of objects named “allcars”. The getter for each selected object is then used in “WriteToTextCarFile()” function for every car object stored in “allcars” with for loop. At last, every object will then be stored in the text file “Car.txt”.

## 4.2 Constructor and methods

In this car rental system, object-oriented concepts are implemented including constructors and methods. These 2 are similar to each other as both of them xxx. However, A constructor does not have a return type and it must have the same name as the class while a method may have a return type unless it is “void” and it can be named however the user desires. Moreover, method is known as explicitly while for constructor, it is known as implicitly. (Javin, 2021)

*Text

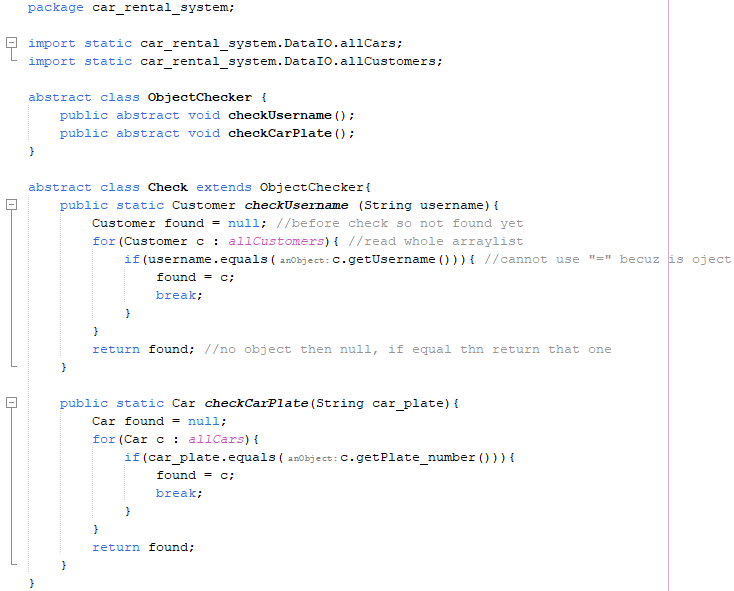
Description automatically generated*

*Figure 4.2.1*

From the *figure* above*,* it shows the class of car. In this class, there are constructor for object, setters and getters for each variable.

## 4.3 Abstractions

Similarly, in object-oriented programming, abstraction is the practice of concealing implementation details from the user so that just functionality is exposed. In other terms, the user will know what the thing does rather than how it does it.

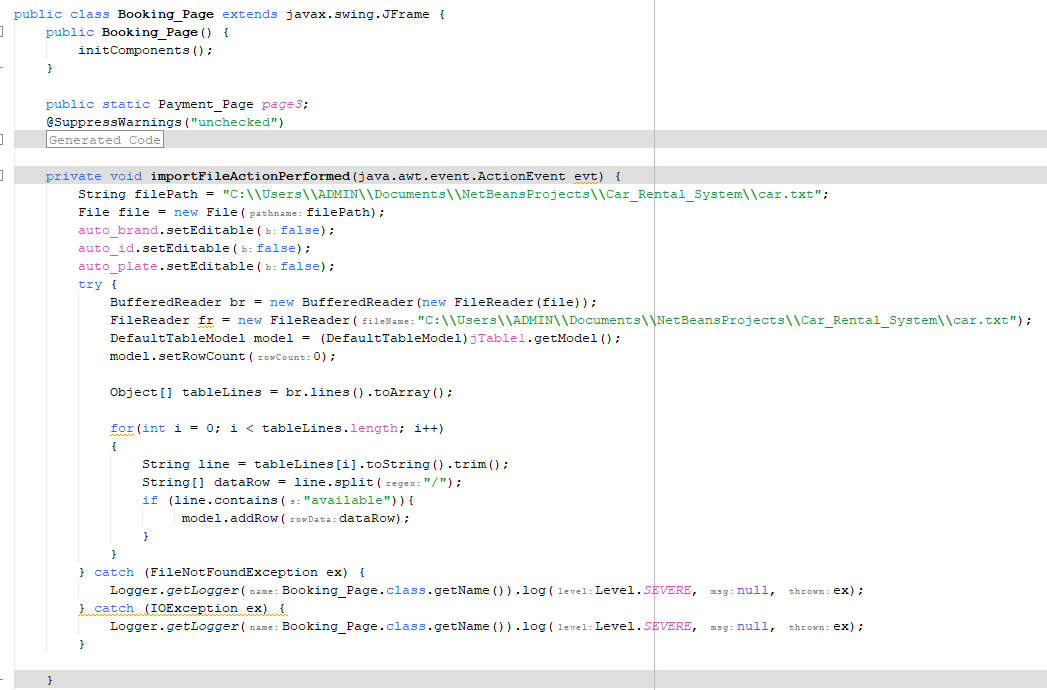


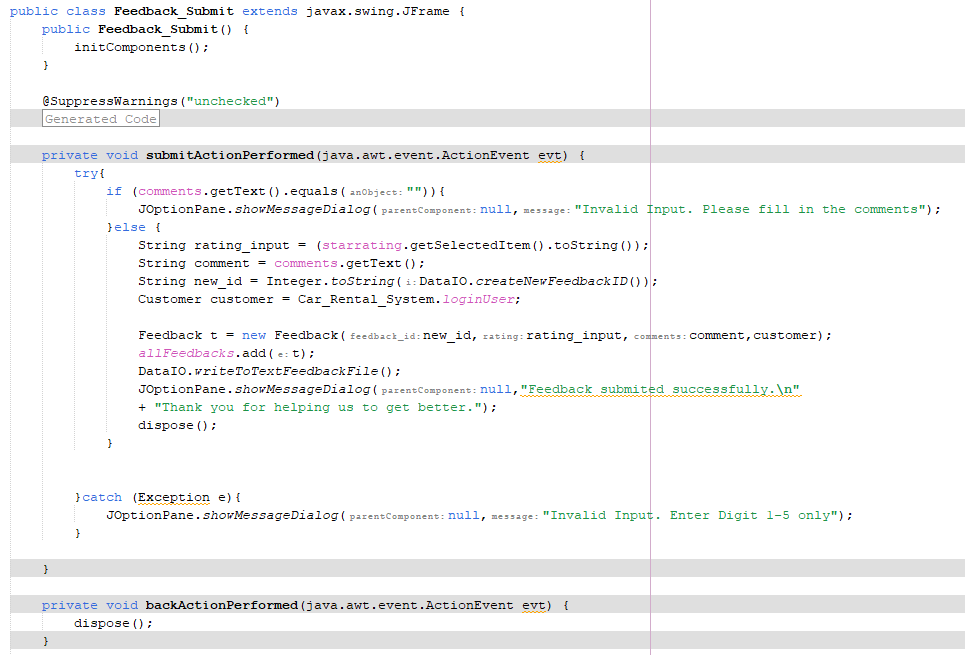
*Figure 4.3.1*

As indicated in the images above, an abstract class called "ObjectChecker" and abstract methods called "checkUsername" and "checkCarPlate" will be constructed to check the username of the Customer object and the vehicle plate of the Car object, respectively. The abstraction makes the code easy to adapt and implement, avoiding the necessity to write complex code several times. It also contributes to the long-term maintainability of the car rental system.

## 4.4 Modularity

In general, modularization refers to the process of constructing anything in various components or modules. The main motivation for modularization is to make things more manageable and to have more control over them (Xperti, 2021). Modularization in Java refers to the process of designing a Java program in separate modules rather than as a monolithic, single architecture. Also, it helps other to understand the complex system.

*Figure 4.4.1*

*Figure 4.4.2*

*Figure 4.4.3*

As seen in the images above, a large system project such as Car Rental System will be divided into numerous modules such as Booking Page class, Feedback submit class, Payment page class, and more, each of the classes will be divided into several functions to suit the system's requirements. Such modularity in Java will be advantageous for manipulating and modifying the system, as only the specified module needs to be modified, rather than the entire code.

## 4.5 High Cohesion

The principle of Object-Oriented coding is cohesion in Java. Cohesion is directly connected to ensuring that the goal for which a Java class is designed is focused and single. In other words, the more closely similar elements in a class are grouped together, the stronger the cohesiveness （JavaTPoint).

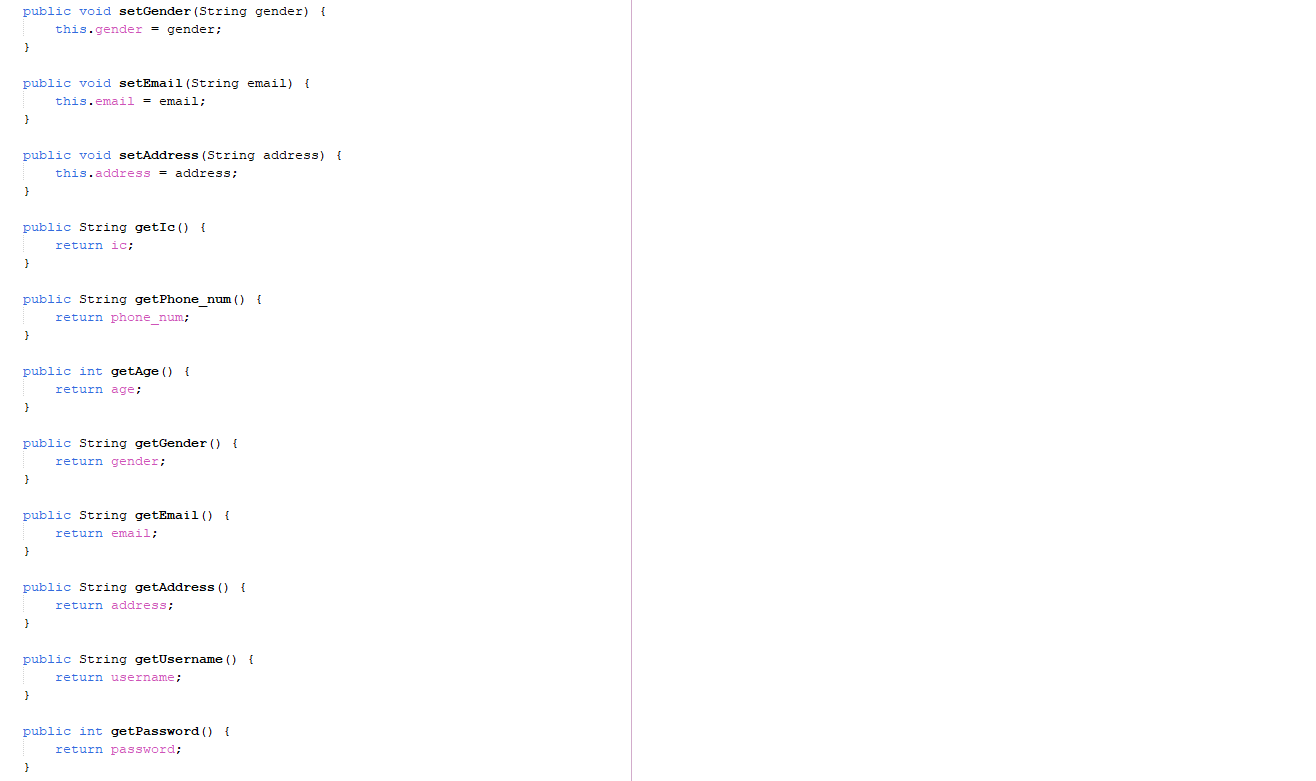
Cohesiveness is required since it is simple to operate in a highly cohesive system. The code is now simple to update and modify. Consider how books are organized at a library. In a library, we sort things into categories. There is a section for each of the several subjects. The difference between high and low cohesion is that high cohesiveness allows us to properly manage our facilities and Low cohesion leads to monolithic classes that are hard to manage, comprehend, and reuse.

Text

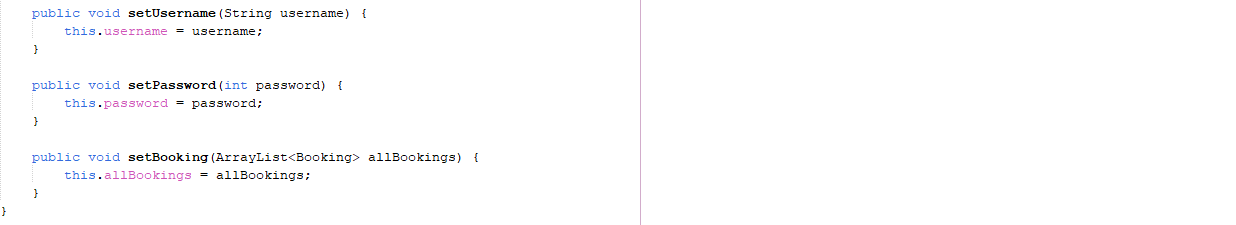
Description automatically generated*Figure 4.5.1*



*Figure 4.5.2*



*Figure 4.5.3*



*Figure 4.5.4*

As figures above shown, the Customer class only contains functions that related to customer, there are no redundant unrelated function in this Customer class. Therefore, the cohesion is high and brought convenience for programmers to manage and modify the class.

4.6 Hierarchy

Hierarchy is one of the inheritance types in object-oriented programming. One class can acquire all the characteristics and behaviors of another class through a process known as inheritance. The class that inherits the behaviors and attributes is known as the parent or base class. While the class that inherits it is named derived or child class.

Text

Description automatically generated*Figure 4.5.5*

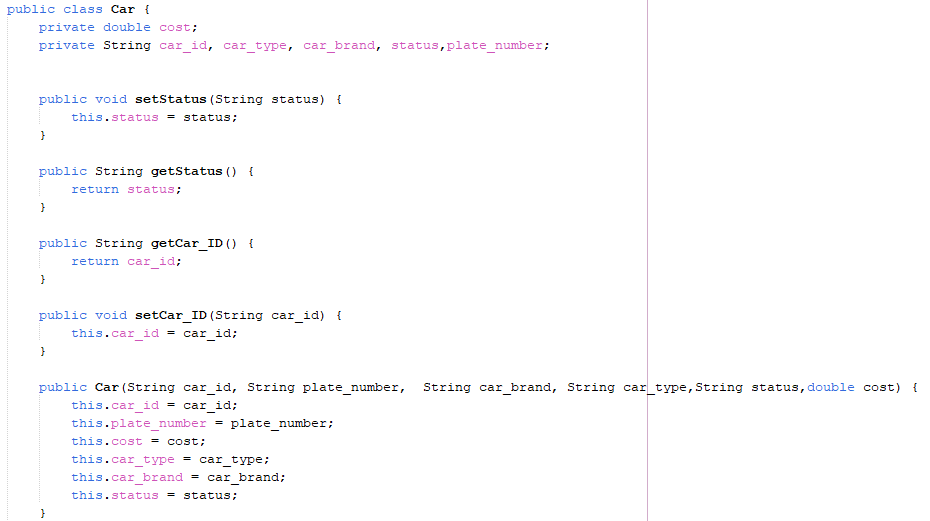
As shown in figure above, the children class “blacklist” inherits from the class “Customer”. The constructor of the class is carrying the attributes from class “Customer”.

## 4.7 Loose Coupling

To put it simply, loose coupling implies they are mainly autonomous. Class A and class B are considered to be loosely connected if the only knowledge that class A knows about class B is what class B has revealed through its interface (GeekforGeeks, 2021). For the loose coupling, A modification in one class does not need changes in the other, and the two classes can function independently.



*Figure 4.7.1*



*Figure 4.7.2*

As figures above shown, figure 1 is the class named “Customer” and figure 2 is a class named “Car”, these both classes are independent to each other which if modify one of the classes it does not affect the other class.

# 5.0 Diagrams

## 5.1 Case diagram

## Diagram Description automatically generated

|  |  |
| --- | --- |
| **Use Case** | Register Member |
| **Brief**  **Description** | This use case allows a new customer to register as a member of the system. |
| **Actors** | Customer |
| **Preconditions** | Customer provides personal information. |
| **Main Flow** | (a) The use case begins when a visitor activates the option to register account.  (b) The system requests that the visitor enters personal details to be register.  (c) The system searches in the database and displays a confirmation before an account was registered successfully. (ref. Alternative flow). |
| **Alternative**  **Flows** | (c) (i) The system searches through the database and found a same username in which case cannot having two same username account exist in database. |

|  |  |
| --- | --- |
| **Use Case** | Login Member |
| **Brief**  **Description** | This use case allows a customer to login to their account of the system. |
| **Actors** | Customer |
| **Preconditions** | Customer provides username and password. |
| **Main Flow** | (a) The use case begins when a visitor activates the option to login account.  (b) The system requests that the visitor enters username and password to be register.  (c) The system searches in the database and displays a confirmation before an account was login successfully. (ref. Alternative flow). |
| **Alternative**  **Flows** | (c) (i) The system searches through the database and found different username or password as the customer’s input in which case cannot login to the account using wrong username or password. |

|  |  |
| --- | --- |
| **Use Case** | Select Car |
| **Brief**  **Description** | This use case allows a customer to select car for booking. |
| **Actors** | Customer |
| **Preconditions** | Customer select a car to book from the car list. |
| **Main Flow** | (a) The use case begins when a visitor activates the option to select car.  (b) The system requests that the visitor select a car that they want to book.  (c) The system only displays cars that are available.  (d) The booking request will be sent to admin after the customer done selecting a car. |
| **Alternative**  **Flows** | - |

|  |  |
| --- | --- |
| **Use Case** | Check Booking |
| **Brief**  **Description** | This use case allows a customer to check their bookings. |
| **Actors** | Customer |
| **Preconditions** | Customers view their bookings using a bookings list. |
| **Main Flow** | (a) The use case begins when a visitor activates the option to view their bookings.  (b) The system searches in the database and displays all bookings of the specify customer. |
| **Alternative**  **Flows** | - |

|  |  |
| --- | --- |
| **Use Case** | Update Information |
| **Brief**  **Description** | This use case allows a customer to updates their personal information. |
| **Actors** | Customer |
| **Preconditions** | Customers update their personal information. |
| **Main Flow** | (a) The use case begins when a visitor activates the option to updates their personal information.  (b) The system searches in the database and displays all bookings of the specify customer.  (c) The system searches in the database and displays a confirmation before a personal info was modified successfully. (ref. Alternative flow). |
| **Alternative**  **Flows** | - (c)(i) The new information that customer input was invalid which the system will ask them to reinput another. |

|  |  |
| --- | --- |
| **Use Case** | Give Feedback |
| **Brief**  **Description** | This use case allows a customer to give feedback on the car rental system. |
| **Actors** | Customer |
| **Preconditions** | Customers submits feedback. |
| **Main Flow** | (a) The use case begins when a visitor activates the option to gives feedback.  (b) The system obtains the customer input and store them into the database. |
| **Alternative**  **Flows** | - |

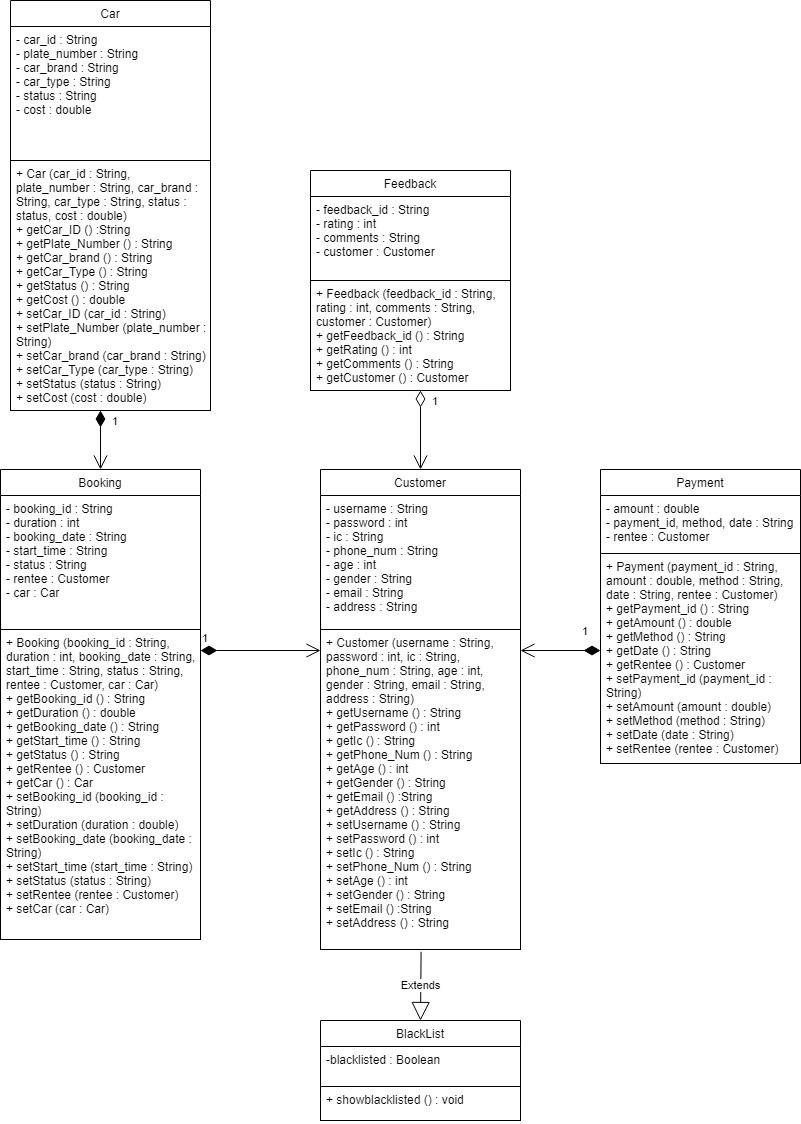
|  |  |
| --- | --- |
| **Use Case** | Manage Car Detail |
| **Brief**  **Description** | This use case give admin the ability to manage cars’ detail such as CarID,  Plate number, car brand, car type, car status, car price |
| **Actors** | Admin |
| **Preconditions** | Admins append, update, delete,search and view, car records with details. |
| **Main Flow** | (a) This use case starts when admin clicks on the button “manage car” in admin’s menu.  (b)The system will display the section for adding car records and update car records while also displaying the existing car records.  (c) In the table of displaying car records, admin is able to select a car to update and select a specific car to delete. |
| **Alternative**  **Flows** | (b)(i) The new car record will not be appended if there was invalid input or conflict with existing car plate number and car ID. |

|  |  |
| --- | --- |
| **Use Case** | Manage Customer Booking |
| **Brief**  **Description** | This use case allows the admin to manage customer’s bookings. |
| **Actors** | Admin |
| **Preconditions** | Admin can view, approved and reject customer’s bookings. |
| **Main Flow** | (a) The use case begins when a visitor activates the option to manage bookings.  (b) The system displayed a bookings list for the admin to choose a booking that he/she wanted to manage.  (b) The system obtains the admin’s input and updates the status of the specified booking. |
| **Alternative**  **Flows** | - |

|  |  |
| --- | --- |
| **Use Case** | Mange report |
| **Brief**  **Description** | This use case allows the admin to generate the reports of the car rental system. |
| **Actors** | Admin |
| **Preconditions** | Admin can generate monthly sales report, marketing report and vehicle report. |
| **Main Flow** | (a) The use case begins when a visitor activates the option to generate reports.  (b) The system displayed three options for the admin to choose which reports to generate.  (b) The system obtains the admin’s choices and generate the specify report. |
| **Alternative**  **Flows** | - |

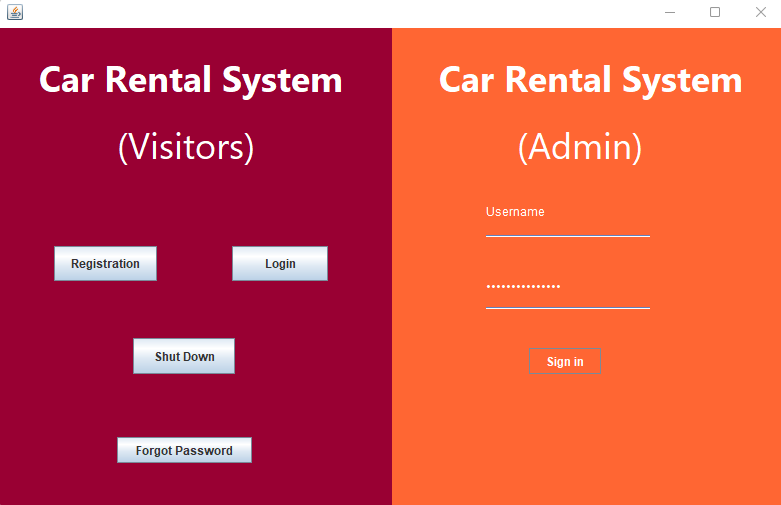
|  |  |
| --- | --- |
| **Use Case** | View customer feedback |
| **Brief**  **Description** | This use case allows admin to read customers’ feedback. |
| **Actors** | Admin |
| **Preconditions** | Admin car read customer feedback with different rating and comments. |
| **Main Flow** | 1. This use case is activated when admin clicked on “View feedback” button in admin’s menu. 2. Then system will read from existing feedback and display for admin |
| **Alternative**  **Flows** | - |

## 5.2 Class diagram



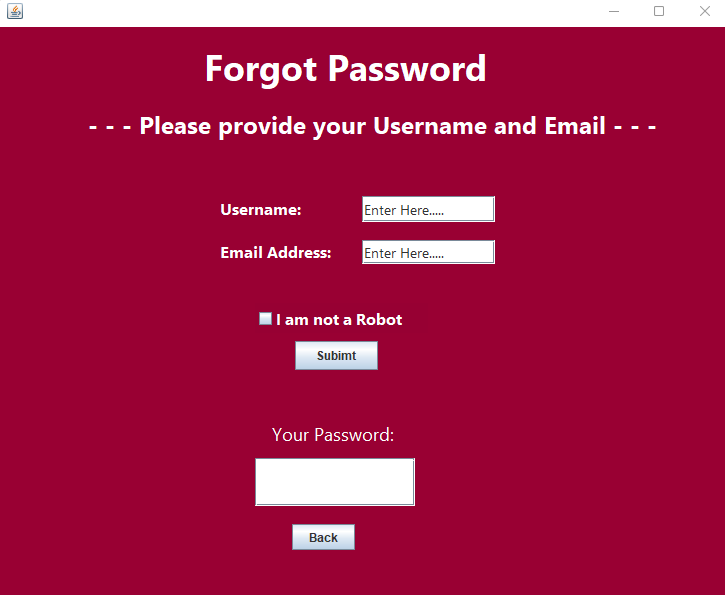
# 6.0 Extra Features

## 6.1 Forgot Password Feature



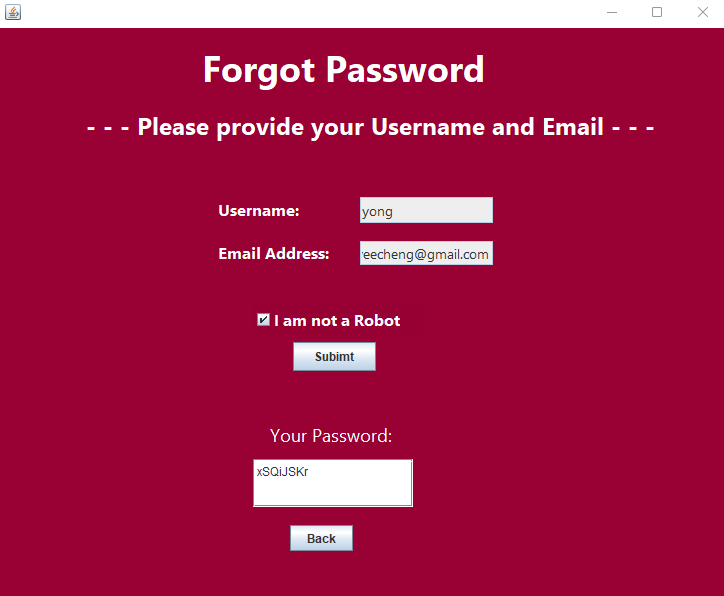
*Figure 6.1.1*

As figure above shows, there will be a “Forgot Password” button on the register/login interface which was design for the customers who having login issues with their password.



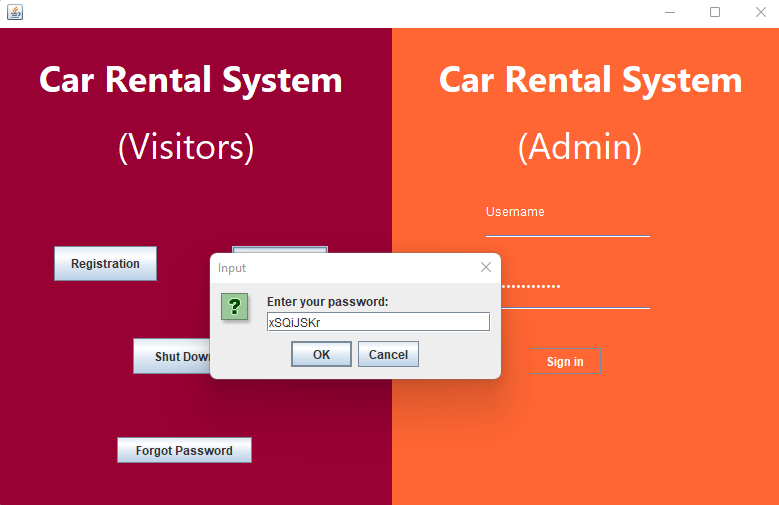
*Figure 6.1.2*

After the button was clicked, a forgot password interface will display to collect information from the customer. For identity authentication purposes, customer must provide their account username and email address to the system.

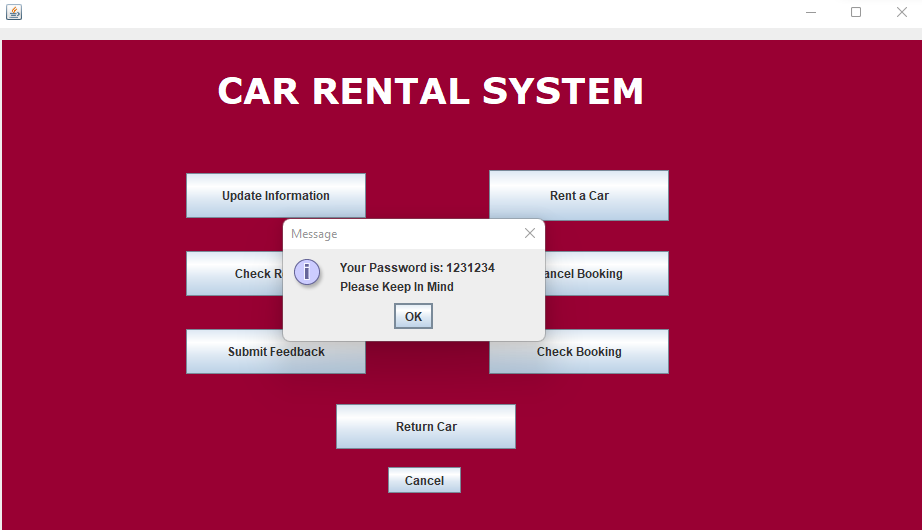


*Figure 6.1.3*

Once the username, email address was entered, and the “I am not a Robot” click box was confirm, clicking the “Submit” button will display a new unique password for the customer. Notice that the validation of the username, email address and click box has been made (refer to Flow of the System).



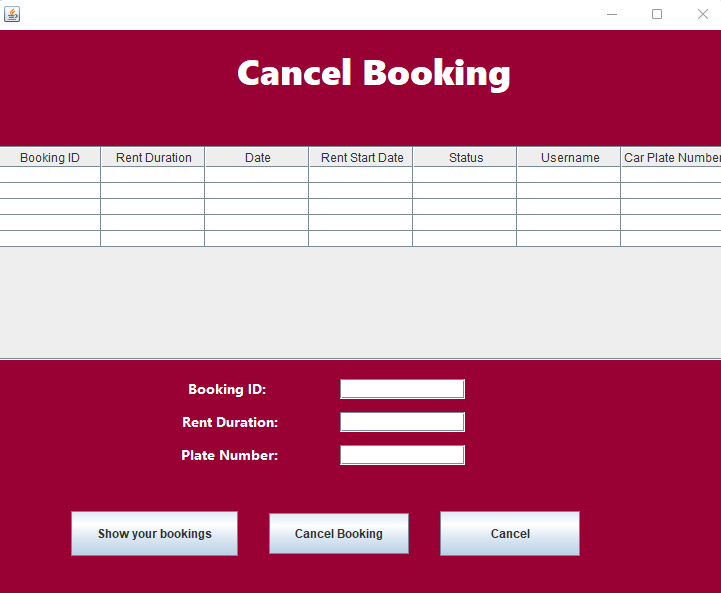
*Figure 6.1.4*



*Figure 6.1.5*

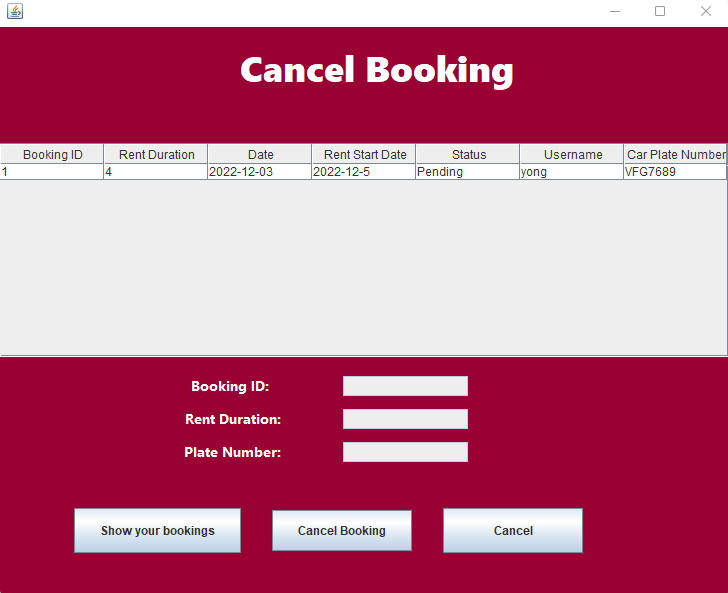
As the figures above shown, the unique password can be used to login to the customer account. After using that password to login, the system will print a notice displaying the customer's true password.

## 6.2 Cancel Booking/Refund Feature



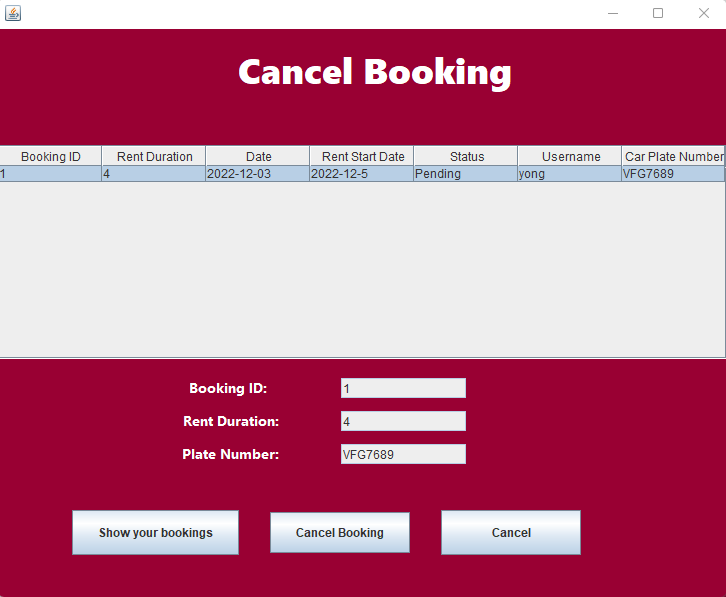
*Figure 6.2.1*

As the figure above shown, there will be an extra booking cancel feature for the customers who regret on their car bookings.



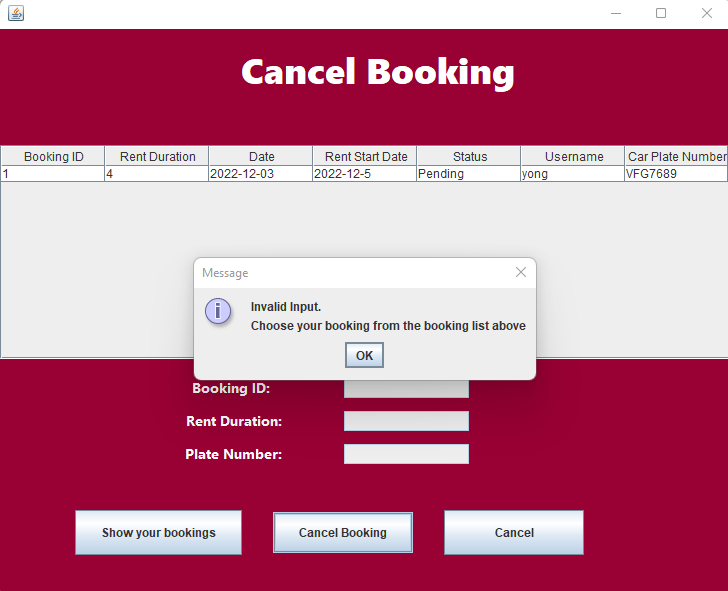
*Figure 6.2.2*

The booking list will only display bookings when the customer clicked on the “Show your bookings” button and only bookings that already approved by the admin will display on the list.



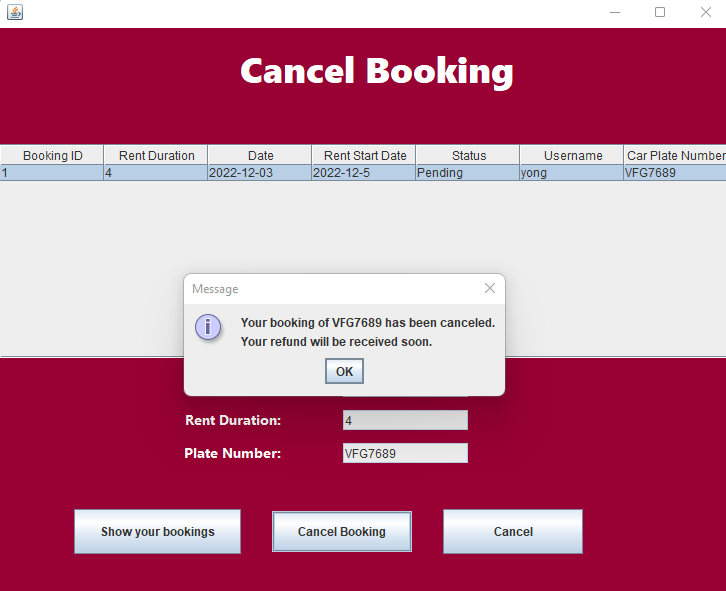
*Figure 6.2.3*

As the figure shows, after choosing the specify booking by clicked on the row of the list, the below booking information will auto-filled in.



*Figure 6.2.4*

A Warning message occurs if customers did not choose any booking and click “Cancel Booking” button.



*Figure 6.2.5*

After the customer chose and clicked on the cancel button, the booking has been officially canceled and the refund will be given to the customers shortly.

## 6.3 Blacklist

This extra feature is to prevent people from being hacked into their account. If the log-in attempt has reached a maximum, the customer will be blacklisted and will not be able to log in again.

Graphical user interface, application

Description automatically generated

*Figure 6.3.1 Attempt fail log in*

In *figure 6.2.1* above, shows the 6th attempt of user trying to login into an existing customer’s account.

Graphical user interface, application, PowerPoint

Description automatically generated

*Figure 6.3.2 not more than 5*

After that, the system will notify the user that the login attempts have reach the limit with message “The login attempt exceeded 5 times.” as shown in *figure 6.2.2* above.

Graphical user interface, application, PowerPoint

Description automatically generated

*Figure 6.3.3 Blacklisted*

Then, the system will inform the user that he or she has been backlisted as shown in in *figure 6.2.3*.

# 7.0 References

Singh, C. (2022, November 2). *Constructors in Java – A Complete Guide*. BeginnersBook. Retrieved from: <https://beginnersbook.com/2013/03/constructors-in-java/>

Javin, P. (2021, September 28). *Difference between Method and Constructor in Java and OOP?* Retrieved from Java67: <https://www.java67.com/2012/11/difference-between-method-vs-constructor-java.html>

Banu, A. (2022, June 23). *Hierarchical Inheritance in Java*. EDUCBA. Retrieved from: <https://www.educba.com/hierarchical-inheritance-in-java/>

Afshan Banu (2020, February 5) - Coupling in Java

Retrieved from: <https://www.geeksforgeeks.org/coupling-in-java/>

Asghar, B. A. (2022, March). *Encapsulation in Java | Explained*. Retrieved from: <https://linuxhint.com/encapsulation-java/>

Janssen, T. (2022, January 21). *OOP Concept for Beginners: What is Encapsulation*. Stackify. Retrieved from: <https://stackify.com/oop-concept-for-beginners-what-is-encapsulation/>

# 1bestcsharp (n.d.) - How to Populate JTable From Txt File Text Using Java NetBeans

<https://1bestcsharp.blogspot.com/2017/08/java-import-text-file-data-to-jtable.html>

Rohan Vats (2021, May 2)

Vats, R. (2022, November 17). *Loose Coupling vs Tight Coupling in Java: Difference Between Loose Coupling & Tight Coupling*. upGrad Blog. Retrieved from: <https://www.upgrad.com/blog/loose-coupling-vs-tight-coupling-in-java/>

GeeksforGeeks. (2022, July 5). *Cohesion in Java*. Retrieved from: [https://www.geeksforgeeks.org/cohesion-in- java/](https://www.geeksforgeeks.org/cohesion-in-%20java/)

<https://www.javatpoint.com/cohesion-in-java>

<https://www.tutorialspoint.com/java/java_abstraction.htm>

Vats, R. (2022, November 22). *Modularity in Java Explained With Step by Step Example [2023]*. upGrad Blog. Retrieved from: <https://www.upgrad.com/blog/modularity-in-java/>

Lalani, S. (2022, May 9). *Modularization in Java: What Is It?* Blogs. Retrieved from: <https://xperti.io/blogs/modularization-in-java-what-is-it/>