**ASSIGNMENT 1 FRONT SHEET**

|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification** | **BTEC Level 5 HND Diploma in Computing** | | |
| **Unit number and title** |  | | |
| **Submission date** | 24/06/2022 | **Date Received 1st submission** |  |
| **Re-submission Date** |  | **Date Received 2nd submission** |  |
| **Student Name** | Bùi Hương Linh | **Student ID** | GBH200662 |
| **Class** | GCH1002 | **Assessor name** | Đinh Đức Mạnh |
| **Student declaration**  I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice. | | | |
|  |  | **Student’s signature** | Linh |

**Grading grid**

|  |
| --- |
| Grade (0-10) |
|  |

|  |  |  |
| --- | --- | --- |
| **❒ Summative Feedback: ❒ Resubmission Feedback:** | | |
| **Grade:** | **Assessor Signature:** | **Date:** |
| **IV Signature:** | | |

Table of Contents

[I. Introduction 4](#_Toc106982230)

[II. Requirement 4](#_Toc106982231)

[III. UI design 4](#_Toc106982232)

[IV. Implementation 6](#_Toc106982233)

[1. Explain program structure 6](#_Toc106982234)

[1.1. Register.java 7](#_Toc106982235)

[1.2. Login.java 9](#_Toc106982236)

[1.3. TableFrm.java 11](#_Toc106982237)

[2. Explain classes 13](#_Toc106982238)

[2.1. Class Customer (Customer.java) 13](#_Toc106982239)

[2.2. Class AccountModel(Accountmodel.java) 17](#_Toc106982240)

[2.3. Class MyModel(MyModel.java) 19](#_Toc106982241)

[3. Explain important algorithms 21](#_Toc106982242)

[3.1. Register 21](#_Toc106982243)

[3.2. Login 22](#_Toc106982244)

[3.3. TableFrm 25](#_Toc106982245)

[4. Explain how to handle errors 30](#_Toc106982246)

[V. Test 32](#_Toc106982247)

[VI. Result 36](#_Toc106982248)

[VII. Conclusion 45](#_Toc106982249)

1. Introduction

Currently, the use of hotels for relaxation as well as entertainment is increasingly popular. This leads to manual customer management which makes it difficult, delayed and out of control of the right quantity. Therefore, I decided to create an application to support customer management when booking in java language. This application helps me to know the exact customer information and control the price as well as the number of rooms, the date that the customer wants to stay.

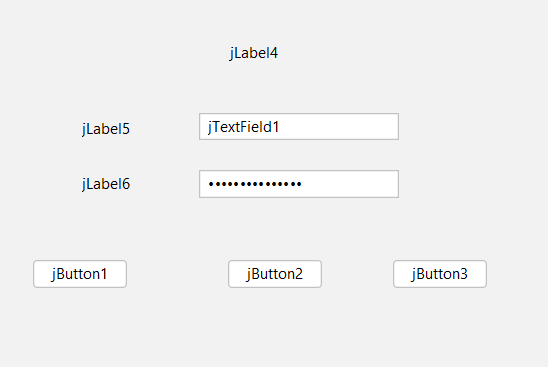
1. Requirement

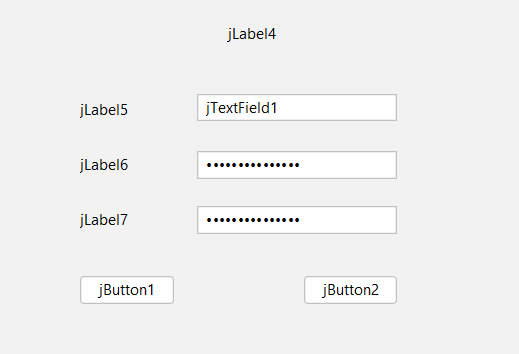
The java program that I created helped me solve some of the following problems:

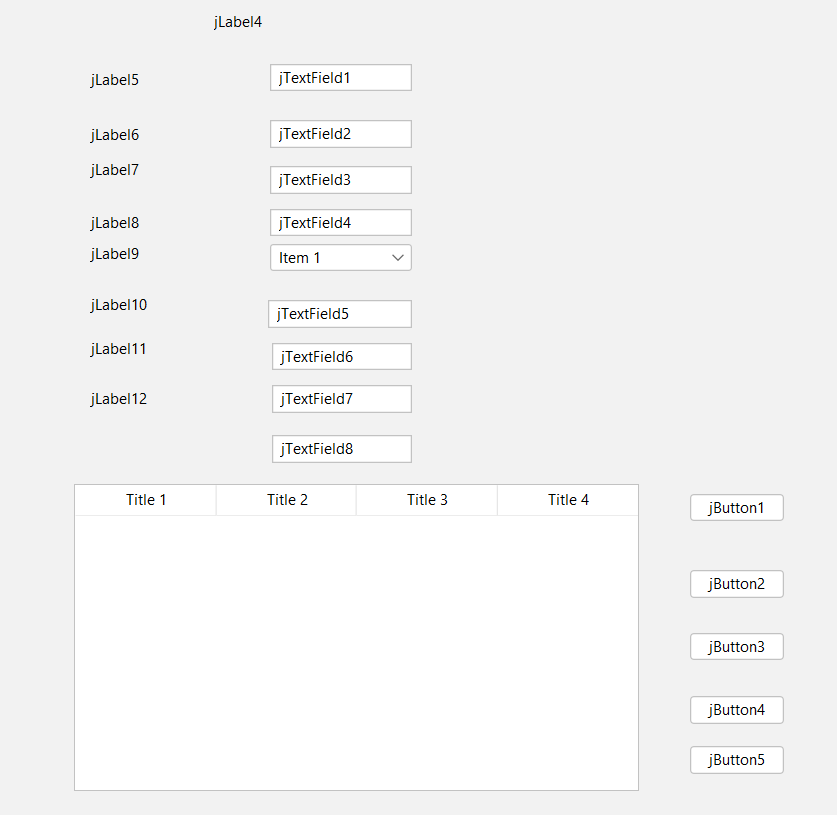
* I can insert, edit, delete, update, insert new and search them easily and understandably.
* I can find the customer information I need through the search function.
* Register a new user for the system.
* Save file to computer.
* Read file from computer.

1. UI design

The basic user interface of the program will have labels, text fields and combo boxes for the user to enter the appropriate data. In addition, the user can interact with the entered data through buttons. Finally, display a table of the entries' results.

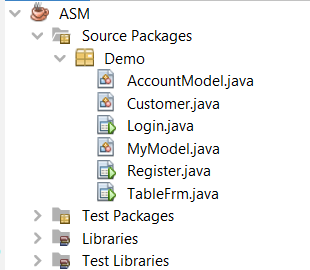






1. Implementation
2. Explain program structure

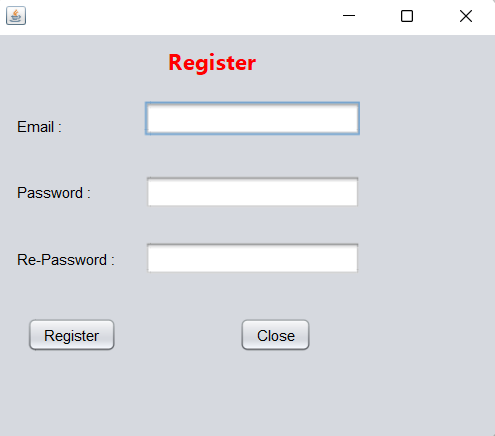
I create a java application, which consists of 6 main parts: 3 classes named Customer.java, MyModel.java and AccountModel.java and 3 Jframes named TableFrm.java, Login.java and Register.java.

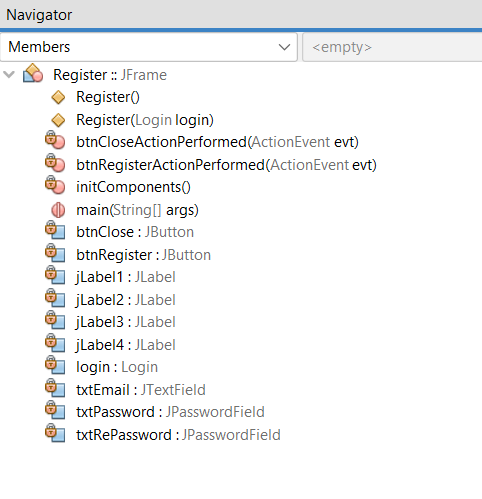


Then the class I created is used to declare customer variables, mymodel, accountmodel with methods. Then I will use them to populate the Jframe I created. I'll develop functions for my software using it.

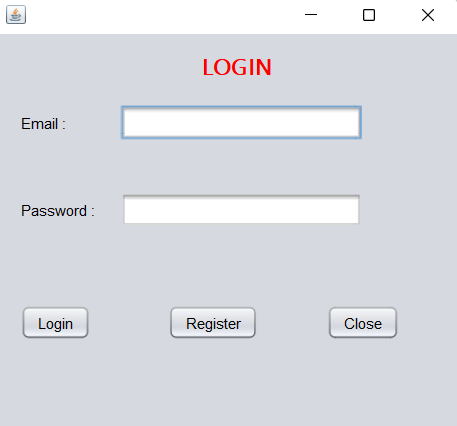
I based my program interface design on the wireframe I had already constructed, giving each piece a variable name (button, text field, label, table, combo box,... )

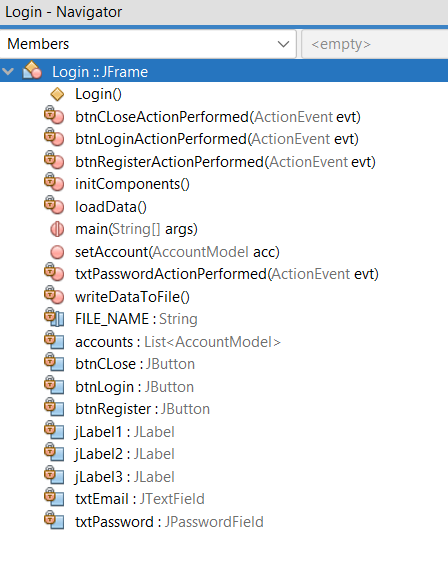
* 1. Register.java



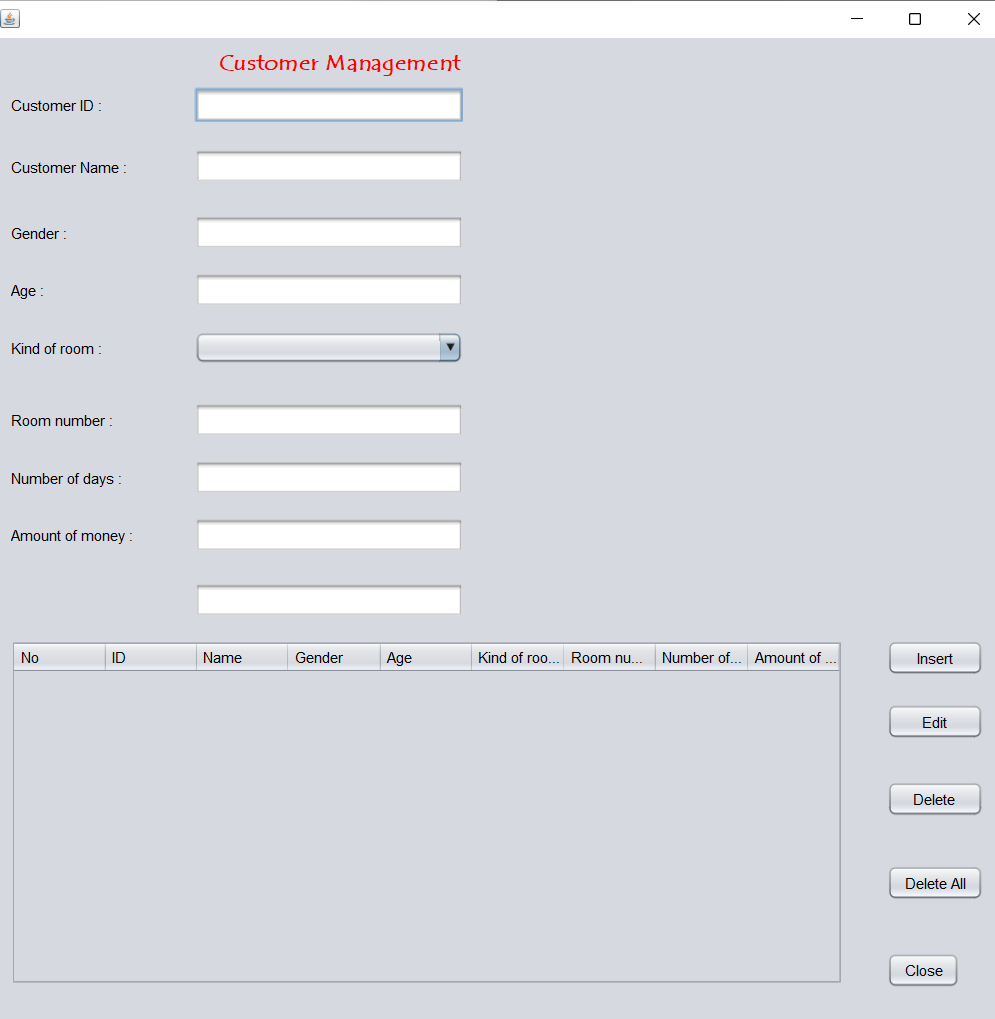


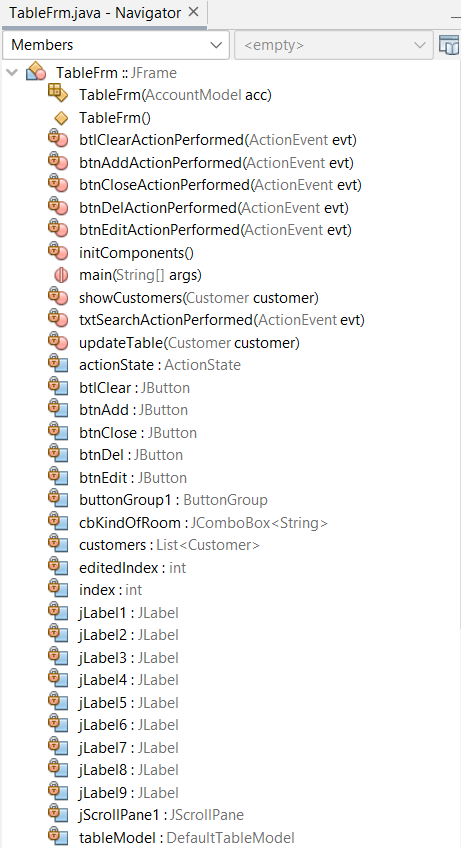
* 1. Login.java

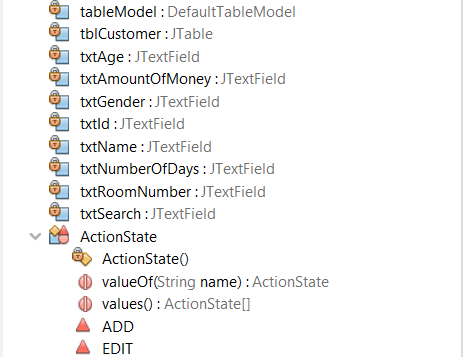




* 1. TableFrm.java

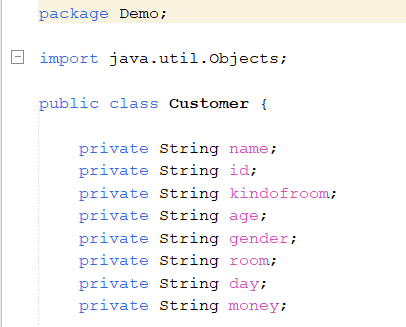




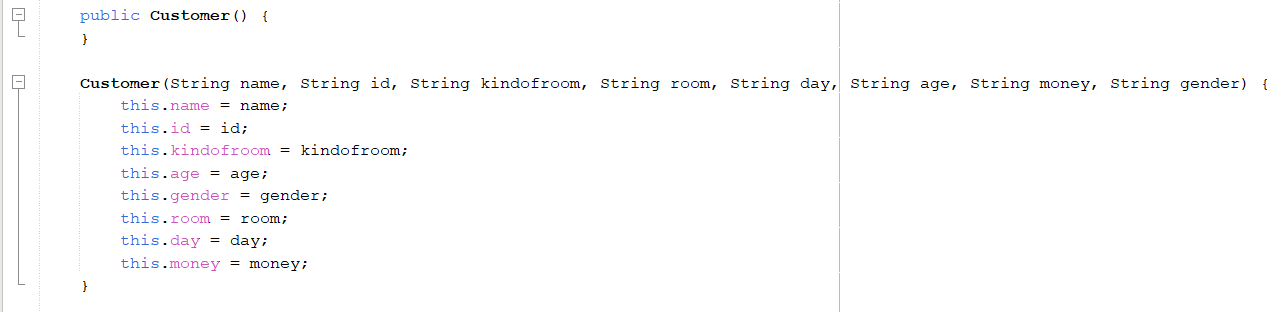


1. Explain classes
   1. Class Customer (Customer.java)

* Import the necessary libraries to use the defined variables.
* In this section, I have declared all local variables with respective types (string, integer, float...) as private because when declaring private can only be accessed within the declared class itself there.

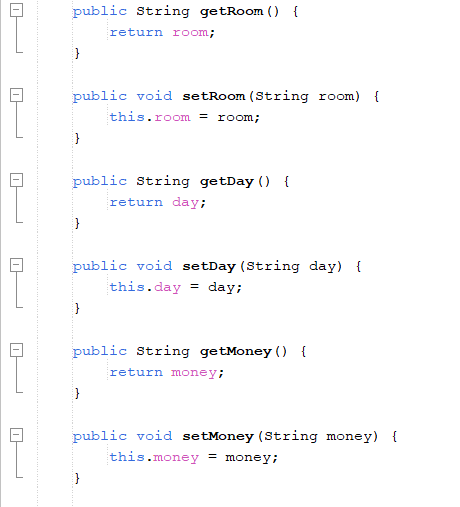


* I'll initialize the methods first. I use the aforementioned declarations to initialize the objects. I was able to call other equivalent methods to initialize the logic inside the objects as well as initialize all the values for the internal properties thanks to this.

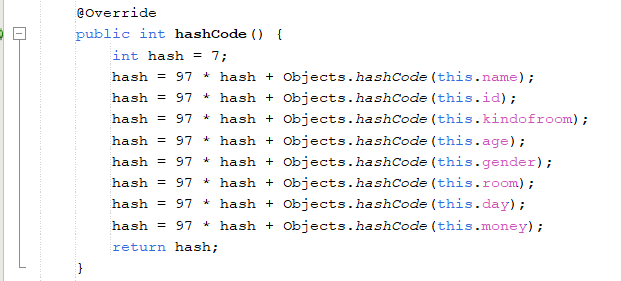


* I applied Java encapsulation using getter() and setter() methods, once accessibility was defined for the private properties I created above, and those properties were given view and edit.



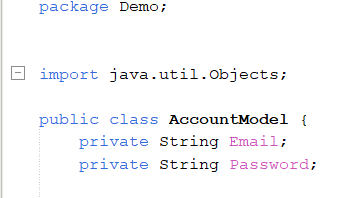


* To this class, I'm using inheritance in Java. I can use the hashCode() function to get the hashCode value of the aforementioned initialized components. Additionally, I compare two objects semantically using the equals() method.





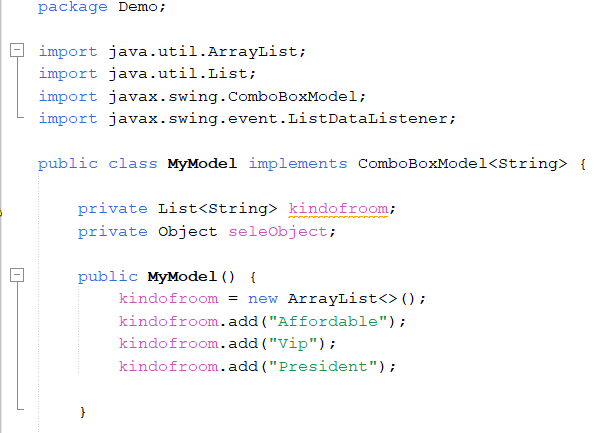
* 1. Class AccountModel(Accountmodel.java)
* Import the necessary libraries to use the defined variables.
* In this section, I have declared all local variables with respective types (string, integer, float...) as private because when declaring private can only be accessed within the declared class itself there.
* Similar to the explanation of the customer.java .







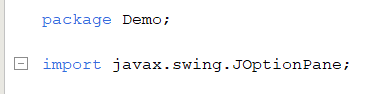
* 1. Class MyModel(MyModel.java)
* Similar to the explanation of the customer.java .
* Import all necessary libraries. Create a class mymodel that inherits comboboxmodel of type string.

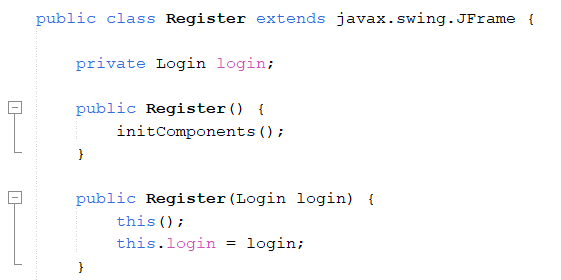




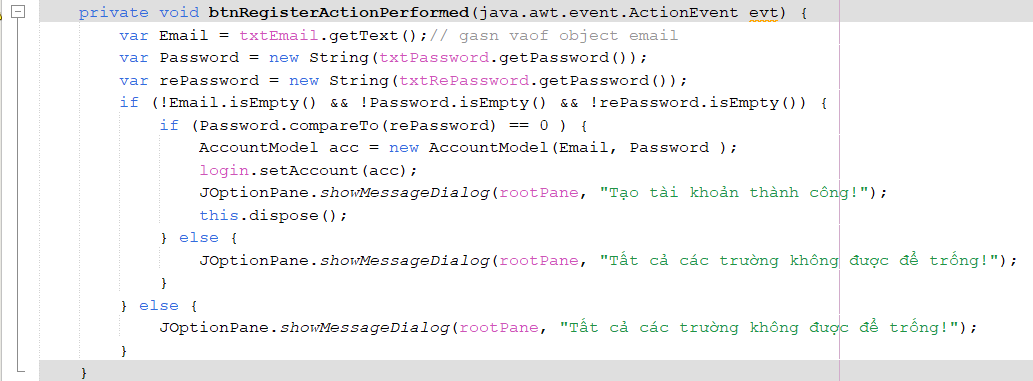
1. Explain important algorithms
   1. Register

* Import all necessary libraries



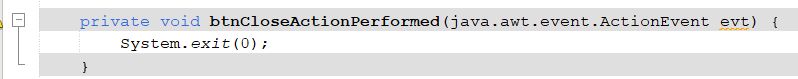


* Get data from users; if entered correctly, it will display the message "Account creation successful," but if incorrectly entered or left blank, the error message "All cases cannot be blank" will be shown. The information is then put into a list using setAccount.

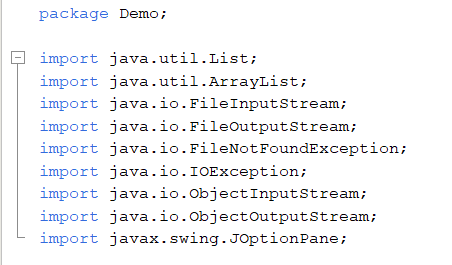


* Exit

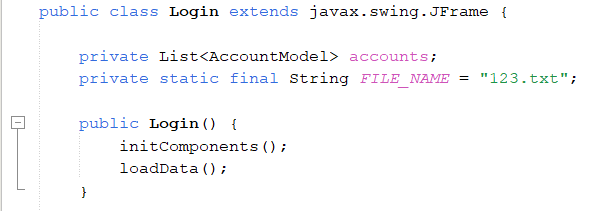
Exit system



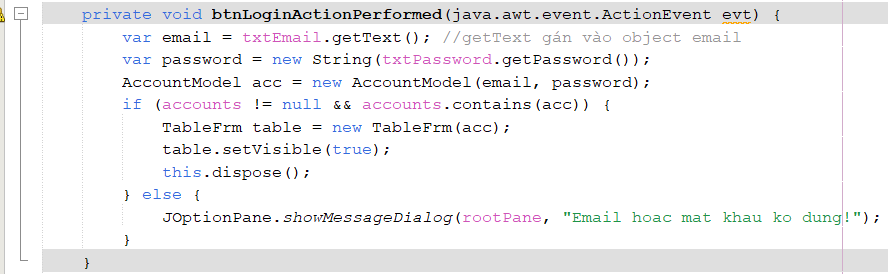
* 1. Login
* Import all necessary libraries.



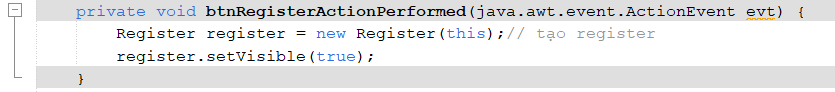
* Create a list containing the ‘AccountModel’ assigned to ‘accounts’. Create a file-name = "123.txt' of type string.



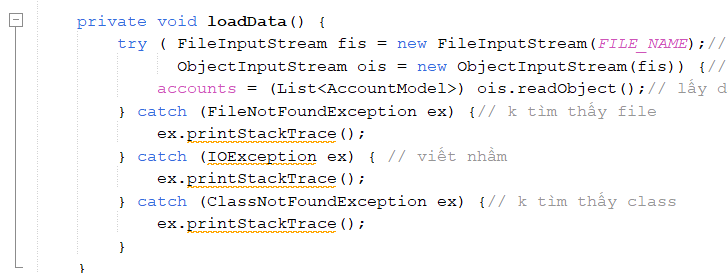
* Get the value from the UI variable and assign it to an intermediate variable. Create an 'AccountModel' containing the email and password assigned to an intermediate variable. If 'account ! = null' and 'account.contains(acc)' will bring up the table 'TableFrm'. Otherwise, an error 'Email or password is incorrect' will be displayed.



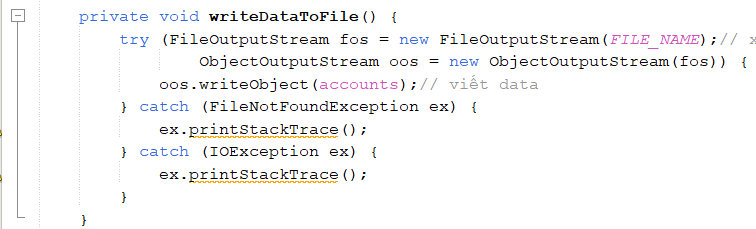
* If the customer does not have an account, when logging in, the customer will be taken to the registration interface before logging into the account.



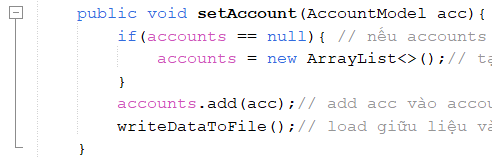
* Create a file containing the 'File\_Name' created above for the variable 'fis'. Create object of 'fi's for variable 'ois'. Read an object in the list that has 'AccountModel' assigned to 'accounts'. There are 3 cases: FileNotFoundException, IOException, ClassNotFoundException.



* Create file output stream containing 'File\_Name' create above for variable 'fos'. Creates an ObjectOutputStream that writes to the specified 'fos' assigned to 'oos'. Write an object to the stream. 2 cases appear: FileNotFoundException, IOException.

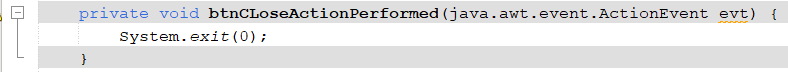


* If 'accounts == null' create an 'ArrayList' assigned to 'accounts'. Then add the object to 'accounts'. Finally load the data into the file.

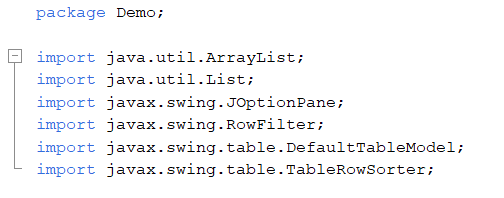


* Exit

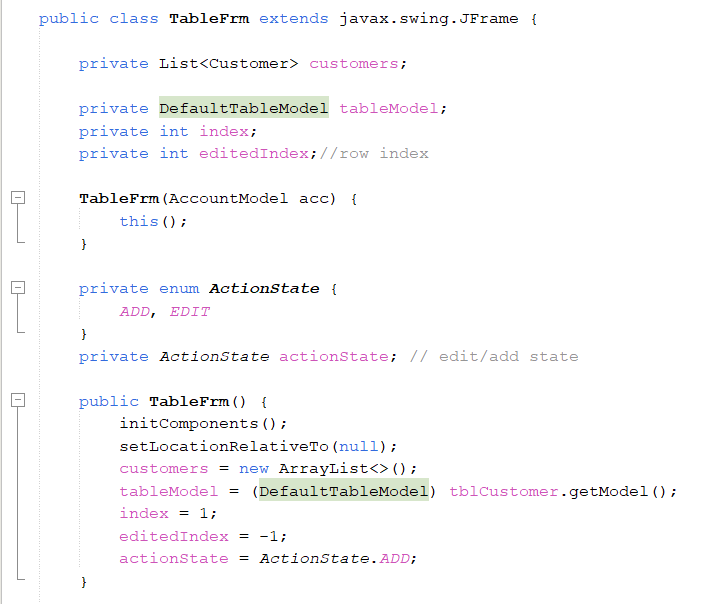
Exit system



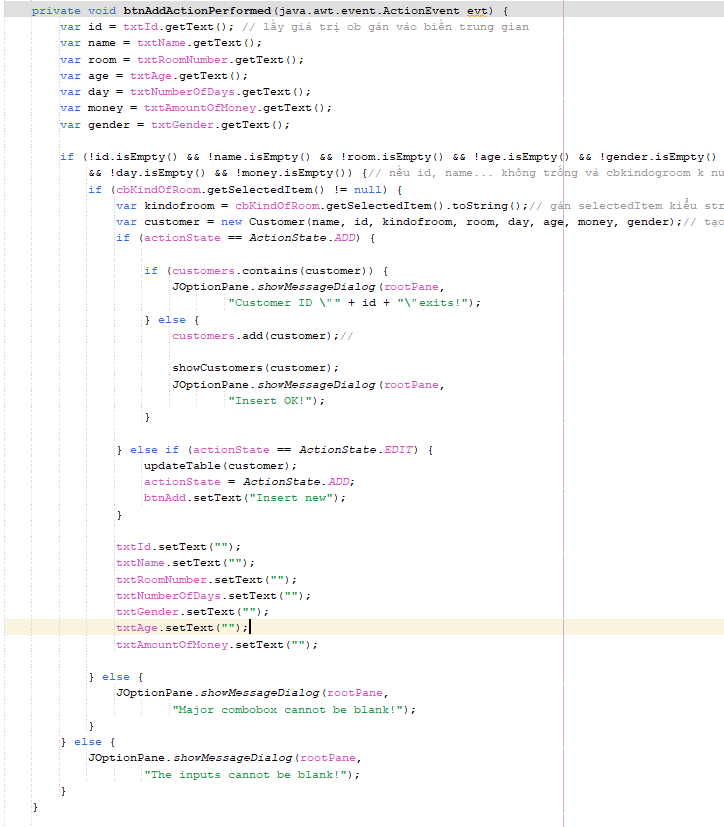
* 1. TableFrm
* I built my program on JFrame. I've first imported the libraries I'll be using in my program.



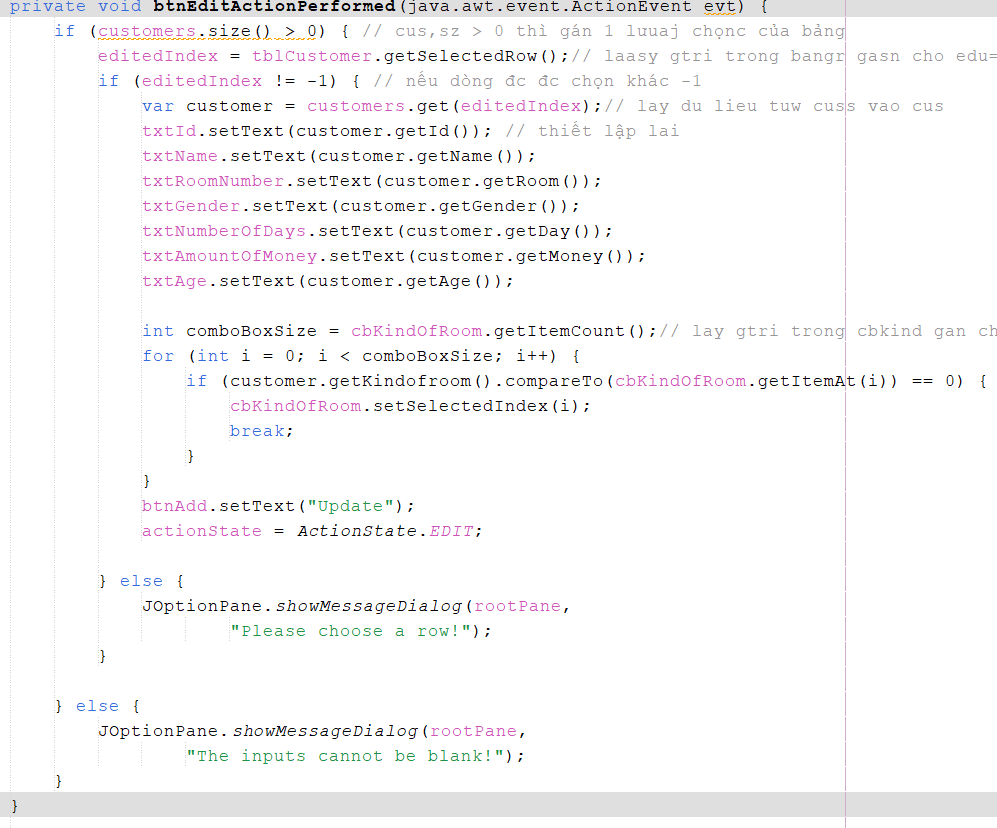
* I constructed the AccountModel.java class and have declared properties, a new List named customer that is connected from it, a memory variable called model to use. Two integer variables, index and editIndex, are utilized to show and interact with information in the information display function.



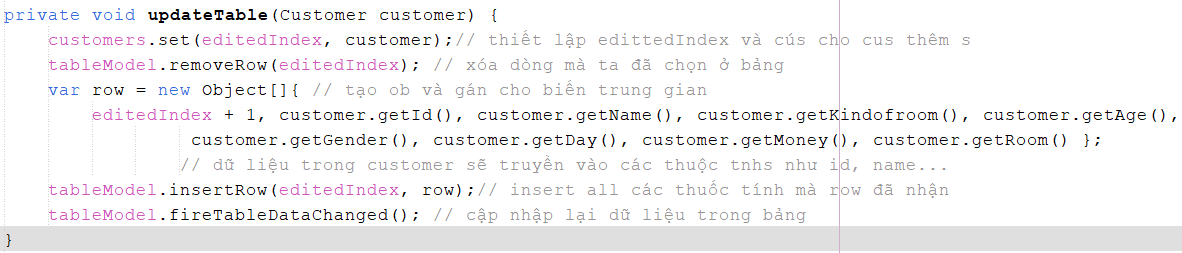
* Get value from UI assign to intermediate variable of type var. If id, name, gender, age,... are not empty and 'cbKindOfRoom' is not null. Get 'selectedItem' of type String to an intermediate variable of type var. Create 'Customer' containing id, name, room, day, gender, .... assign it to a variable of type var. If add is correct, it will display " Insert oki '. Otherwise, it will display the message 'The inputs cannot be blank'



* If customers.size > 0 then assign a selection of the table. Get the value in the table assign to 'editedindex'. If selected line is different from '-1' then get data from 'customers' assign it to 'customer' in type var and reset. done we use for loop vs 'i=0; i<comboBoSize; i++'. if 2 ob =0 then will set the selected item in 'cbKindOfRoom' and 'add' will change to 'update'. Otherwise, the error message 'Please choose a row' will appear. Else the error message 'The inputs cannot the blank' appears.



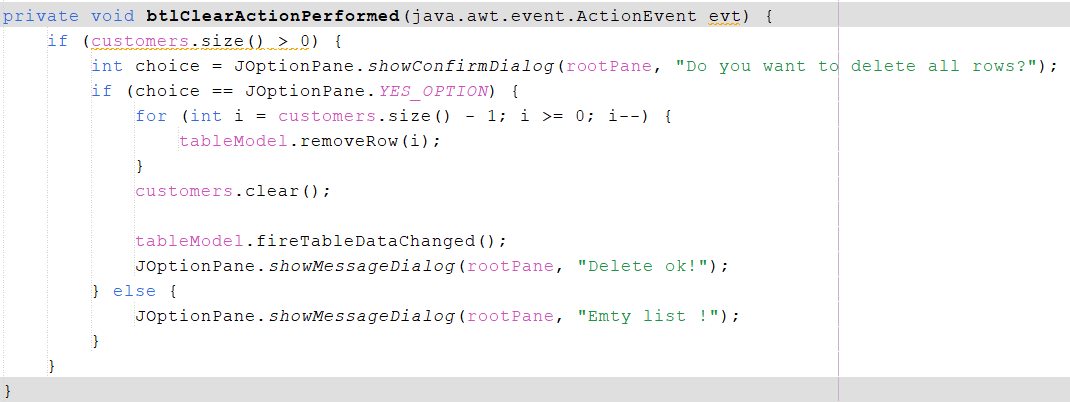
* Set 'edittedIndex and customer' for 'customers'. done, delete the row that we have selected in the table. Create an object and assign it to a variable of type var. data in 'customer' will pass in attributes such as: id, name, age, room, day.... Finish, inserting all the properties that the newly created variable has received into the 'tableModel' table. finally update the data in the table 'tableModel'.



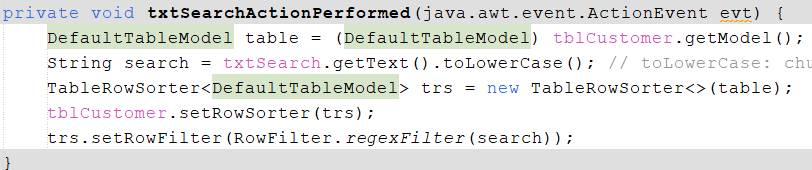
* When the user selects a row in the table, if the selected row is different from '-1' then delete appears the message 'Do you want to delete' and when press yes, the message 'Delete ok' will appear. otherwise, the error message 'Please choose a row to delete' appears. otherwise the error message appears 'Empty list'.



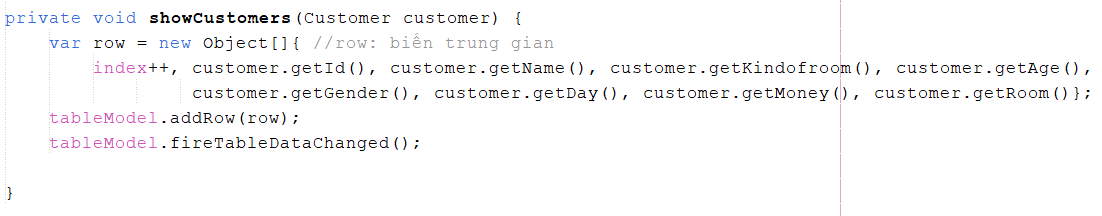
* When the user wants to delete all, the message 'Do you want to delete all rows' appears and when pressing yes, the message 'Delete ok' will appear. Else the error message appears 'Empty list'.



* At the Search function I have initialized the event for it. get Model from table assign to variable. Get value from UI variable convert from chuooix to lowercase assign to variable 'search' of type string. create a 'TableRowSorter' contains 'table' assigned to 'trs'. Then set up the table.

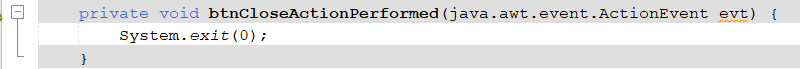


* Create an object and assign it to a variable of type var. data in 'customer' will pass in attributes such as: id, name, age, room, day.... Finish, inserting all the properties that the newly created variable has received into the 'tableModel' table. finally update the data in the table 'tableModel'.



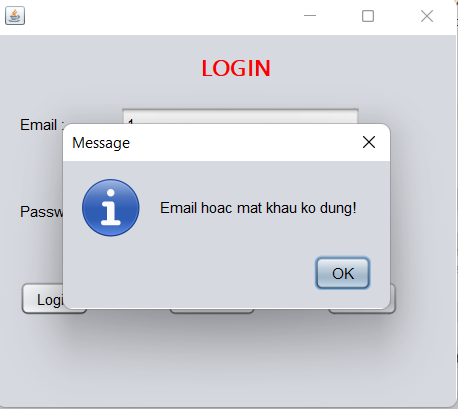
* Exit

Exit system

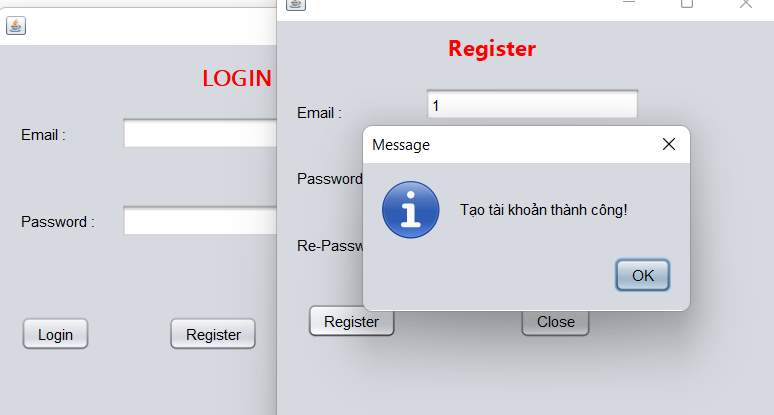


1. Explain how to handle errors

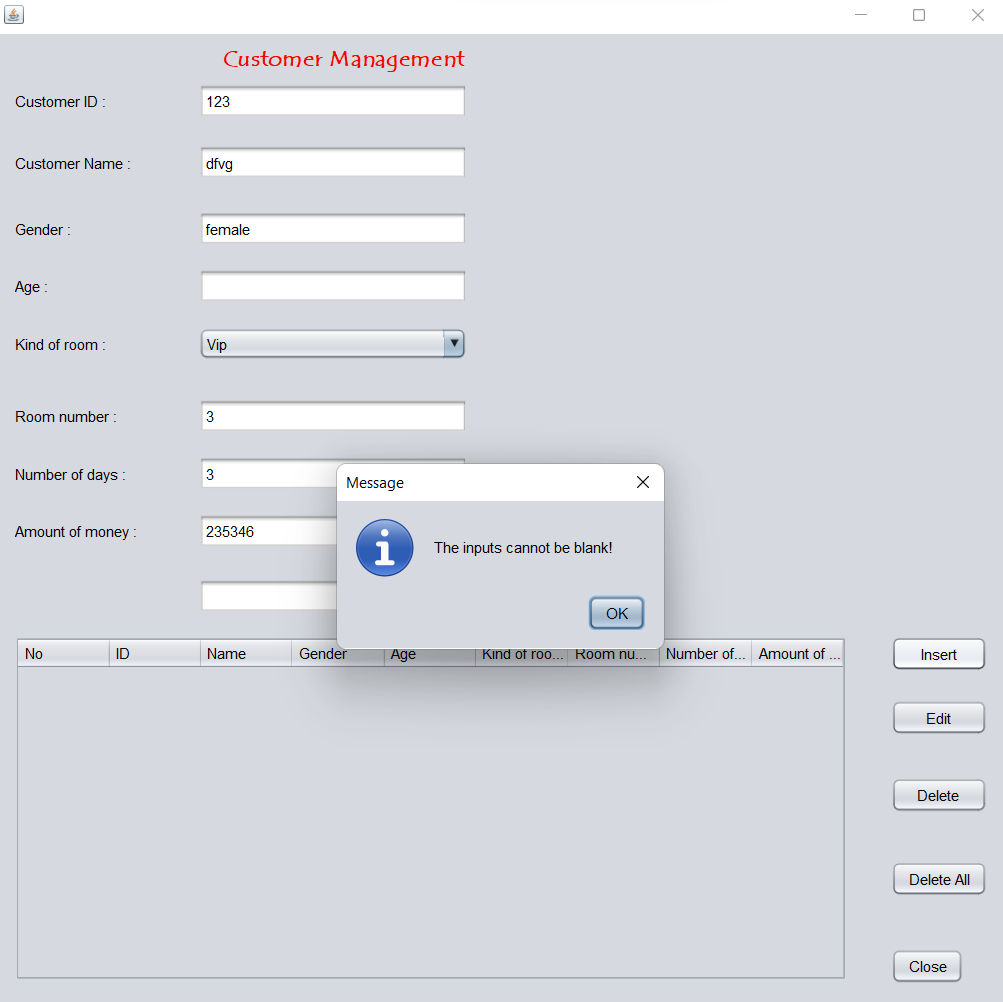
* When a user logs into the account the following error may occur:



* To fix the error the user needs to register a new account.



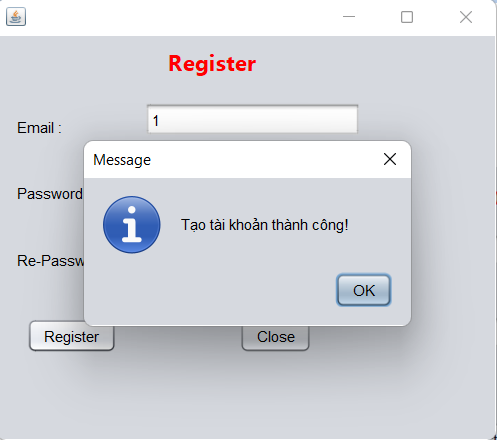
* User forgot to enter age

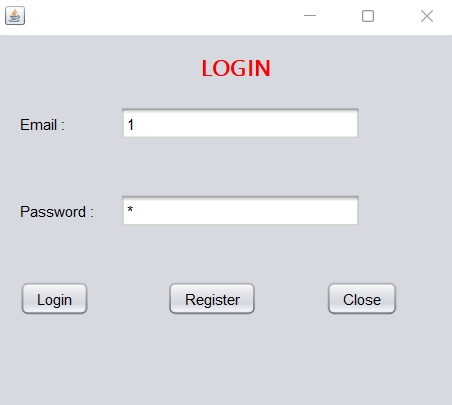


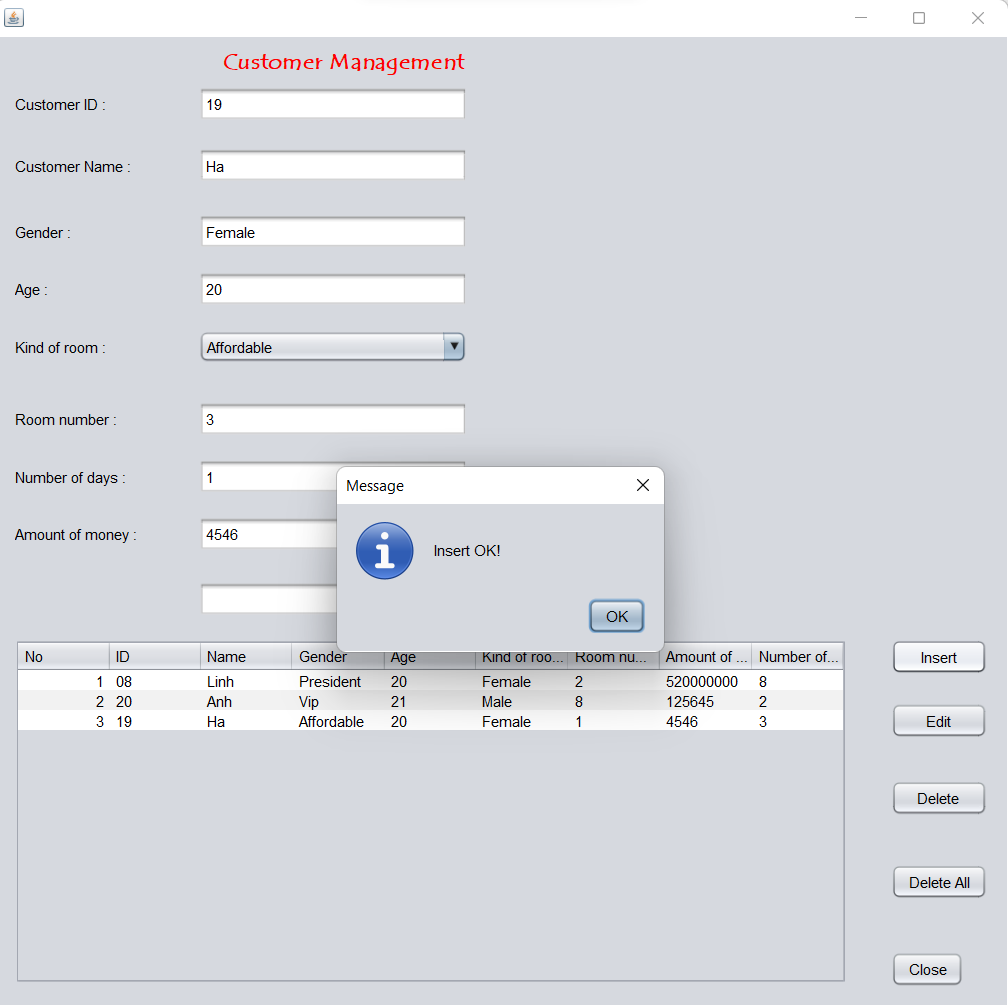
1. Test

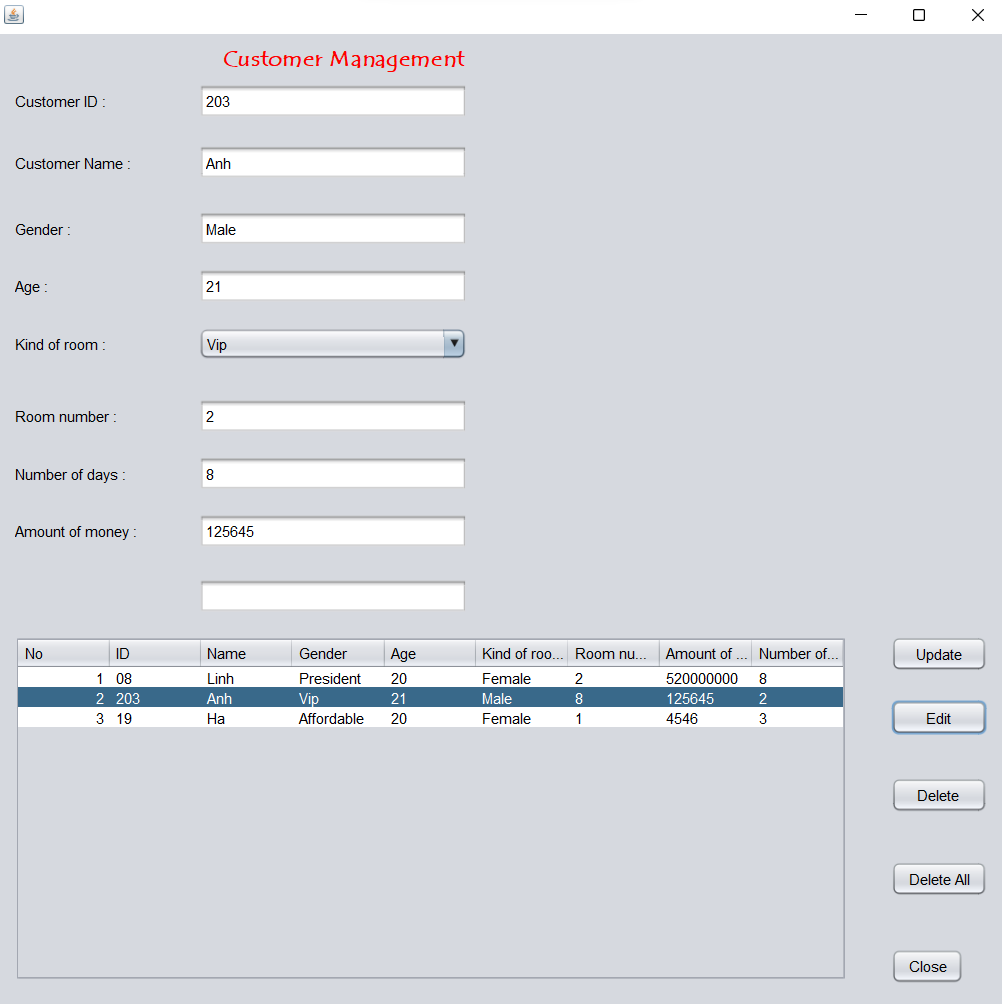
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test | Test Description | Test Steps | Test Data | Expected Results | Actural Results | Pass/False |
| 1 | Register | 1/ Enter  2/ Login | - Email: 1  - Password: \*  - Re-Password: \* | Successfully registered an account. | Successfull | Pass |
| 2 | Login | 1/ Register  2/ Login | - Email: 1  - Password: \* | Successfully logged into the account. | Successfull | Pass |
| 3 | Insert | 1/ Enter  2/ Add customer | - Customer ID: 08  - Customer Name: Linh  - Gender: Female  - Age: 20  - Kind of room: President  - Room number: 8  - Number of days: 2  - Amount of money: 520.000.000 | Add customer | Successfull | Pass |
| 4 | Edit | 1/ Enter  2/ Edit customer  3/ Update customer | - Customer ID: 28  - Customer Name: Anh  - Gender: Male  - Age: 21  - Kind of room: Vip  - Room number: 3  - Number of days: 1  - Amount of money: 5.200.000 | Edit id: 20 -> id: 28  and Update successfully. | Successfull | Pass |
| 5 | Insert new | 1/ Enter  2/ Insert new customer | - Customer ID: 20  - Customer Name: Anh  - Gender: Male  - Age: 21  - Kind of room: Vip  - Room number: 3  - Number of days: 1  - Amount of money: 5.200.000 | Successfully inserted new more customers. | Successfull | Pass |
| 6 | Delete | 1/ Enter  2/ Delete customer | - Customer ID: 08  - Customer Name: Linh  - Gender: Female  - Age: 20  - Kind of room: President  - Room number: 8  - Number of days: 2  - Amount of money: 520.000.000 | Customers with attributes like: id, name, gender, age..... have been removed from the table. | Successfull | Pass |
| 7 | Delete All | 1/ Enter  2/ Clear | - Customer ID: 08  - Customer Name: Linh  - Gender: Female  - Age: 20  - Kind of room: President  - Room number: 8  - Number of days: 2  - Amount of money: 520.000.000  - Customer ID: 20  - Customer Name: Anh  - Gender: Male  - Age: 21  - Kind of room: Vip  - Room number: 3  - Number of days: 1  - Amount of money: 5.200.000 | All customers have been cleared. | Successfull | Pass |
| 8 | Search | 1/ Enter ID  2/ Search | - Customer ID: 08  - Customer Name: Linh  - Gender: Female  - Age: 20  - Kind of room: President  - Room number: 8  - Number of days: 2  - Amount of money: 520.000.000 | Show ID, name, gender, age, kind of room, room number,… searched. | Successfull | Pass |
| 9 | Close | 1/ Enter  2/ Exit | 0 | Exit program | Successfull | Pass |

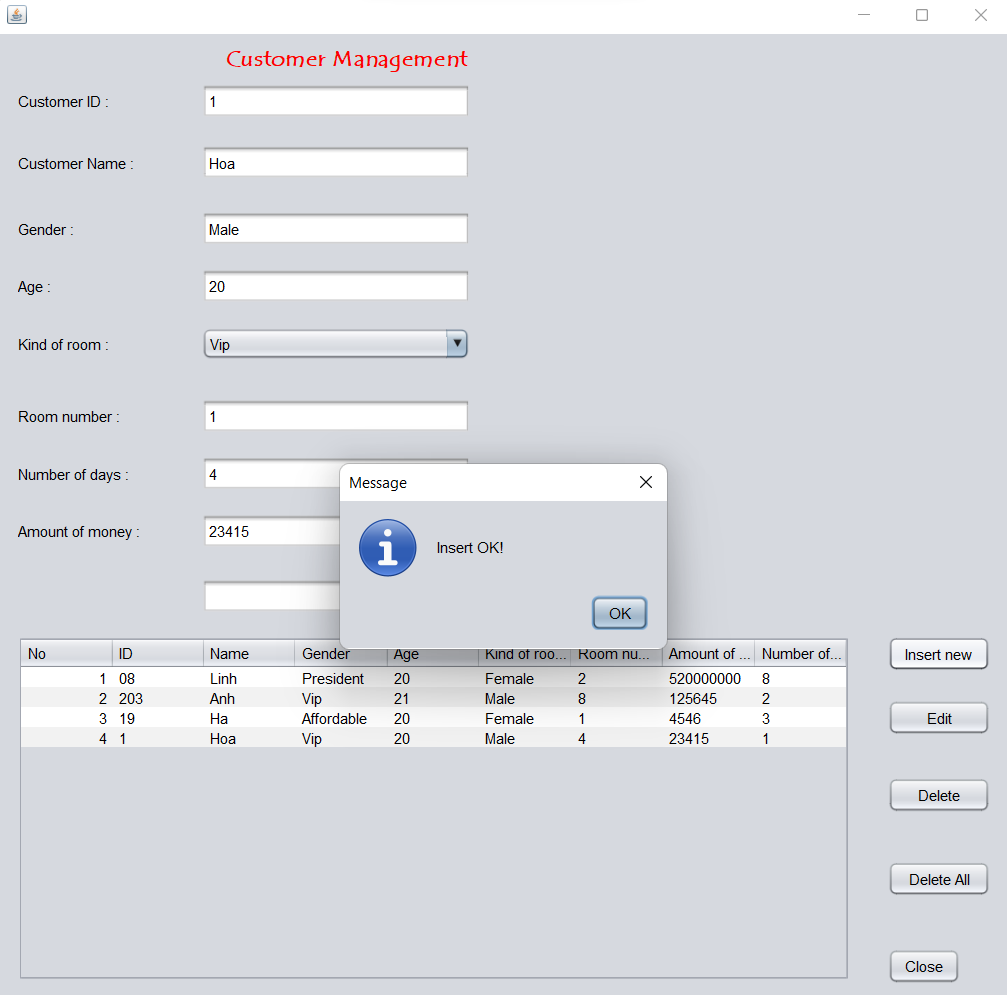
1. Result

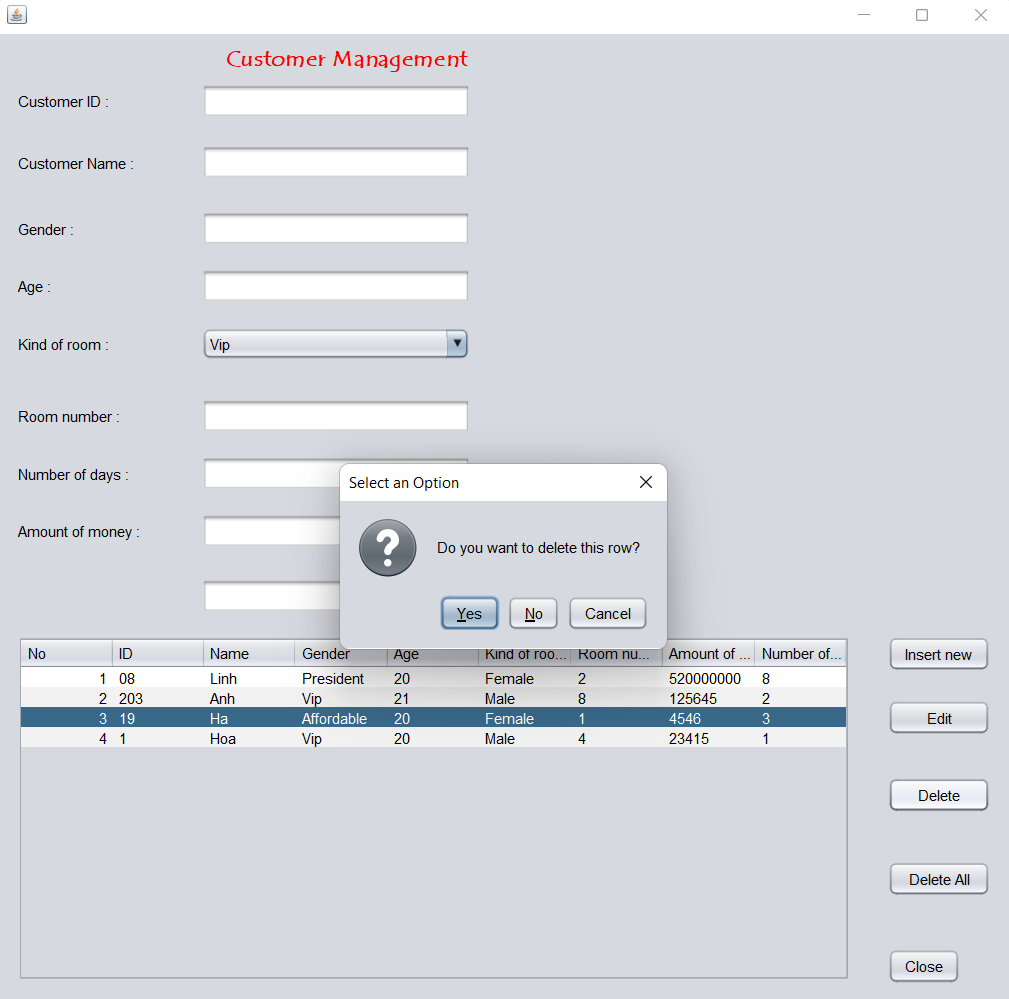


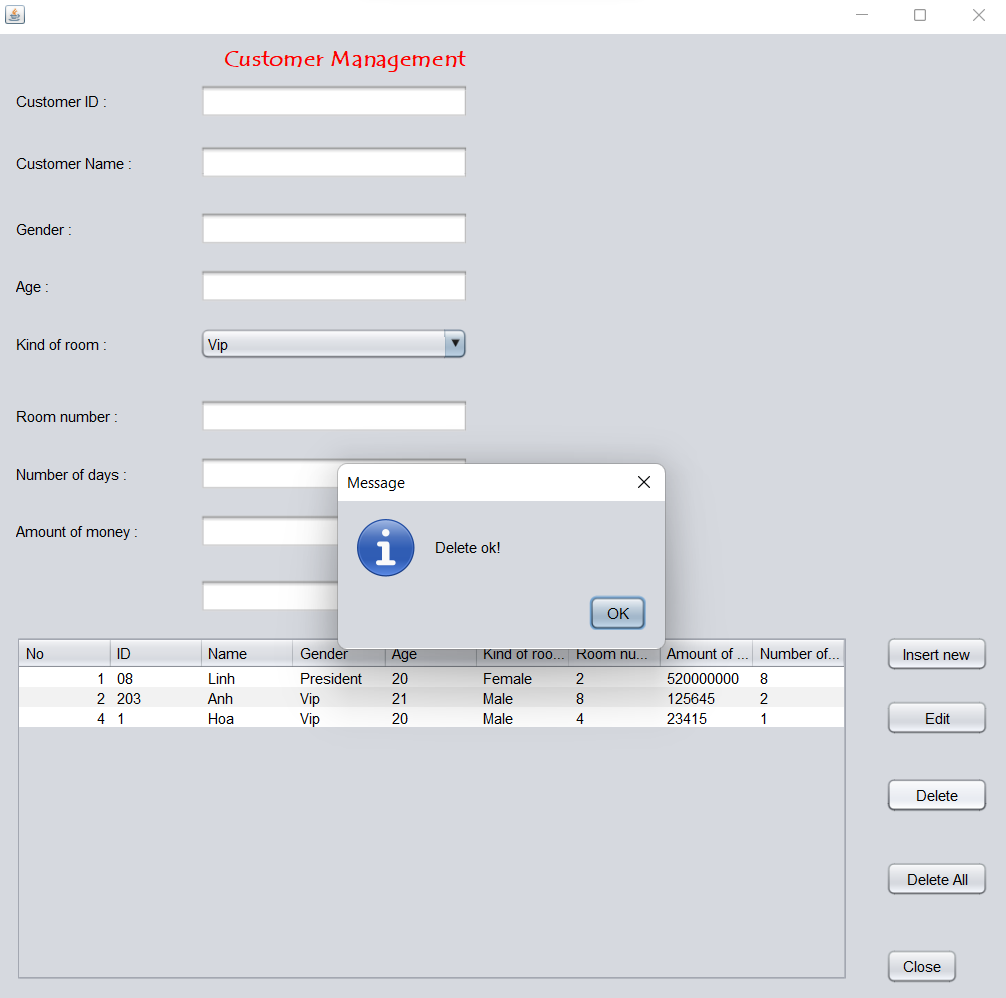


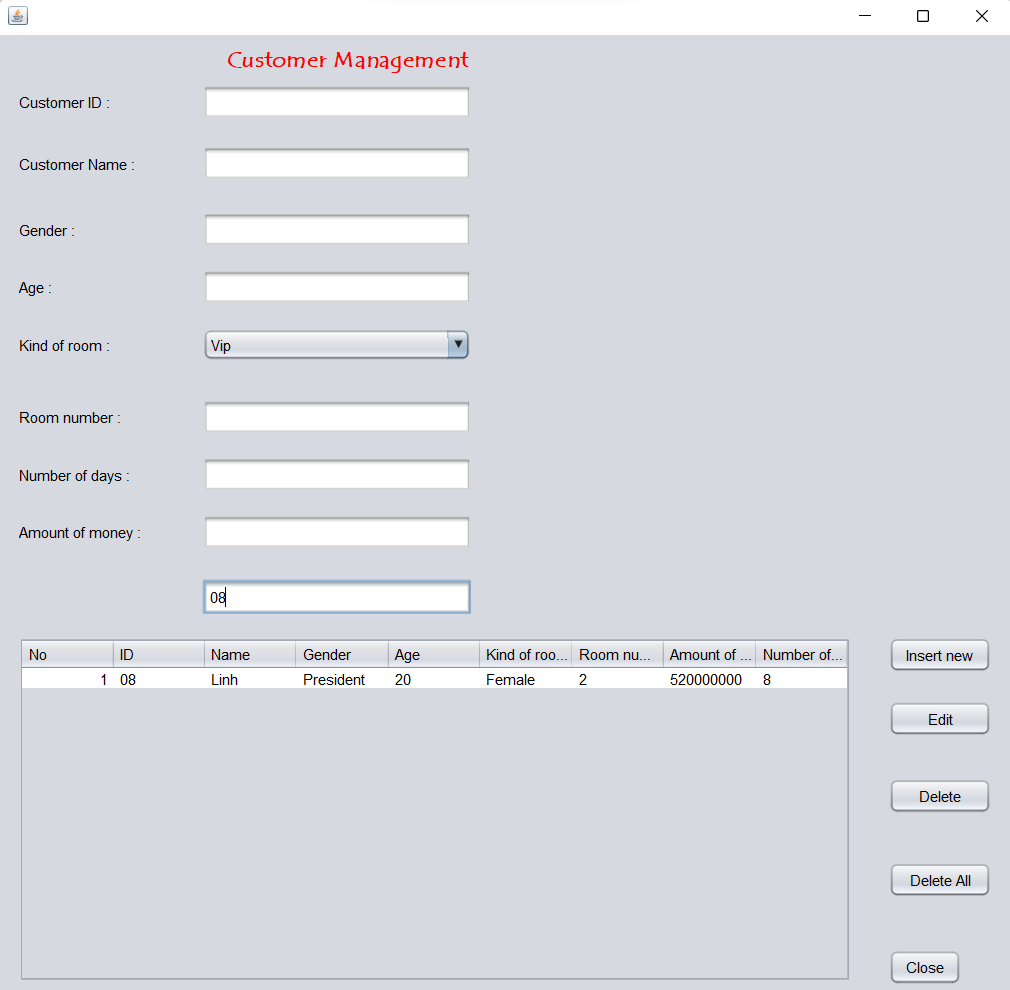


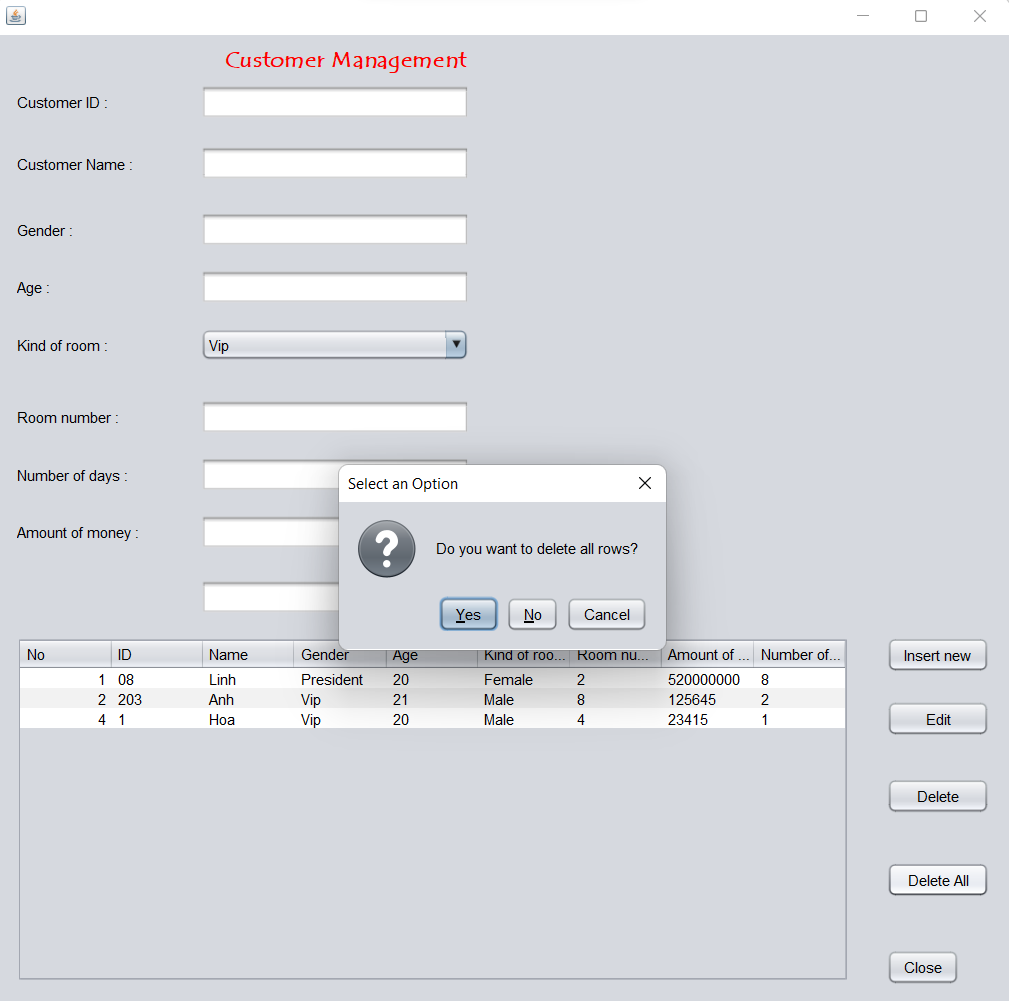


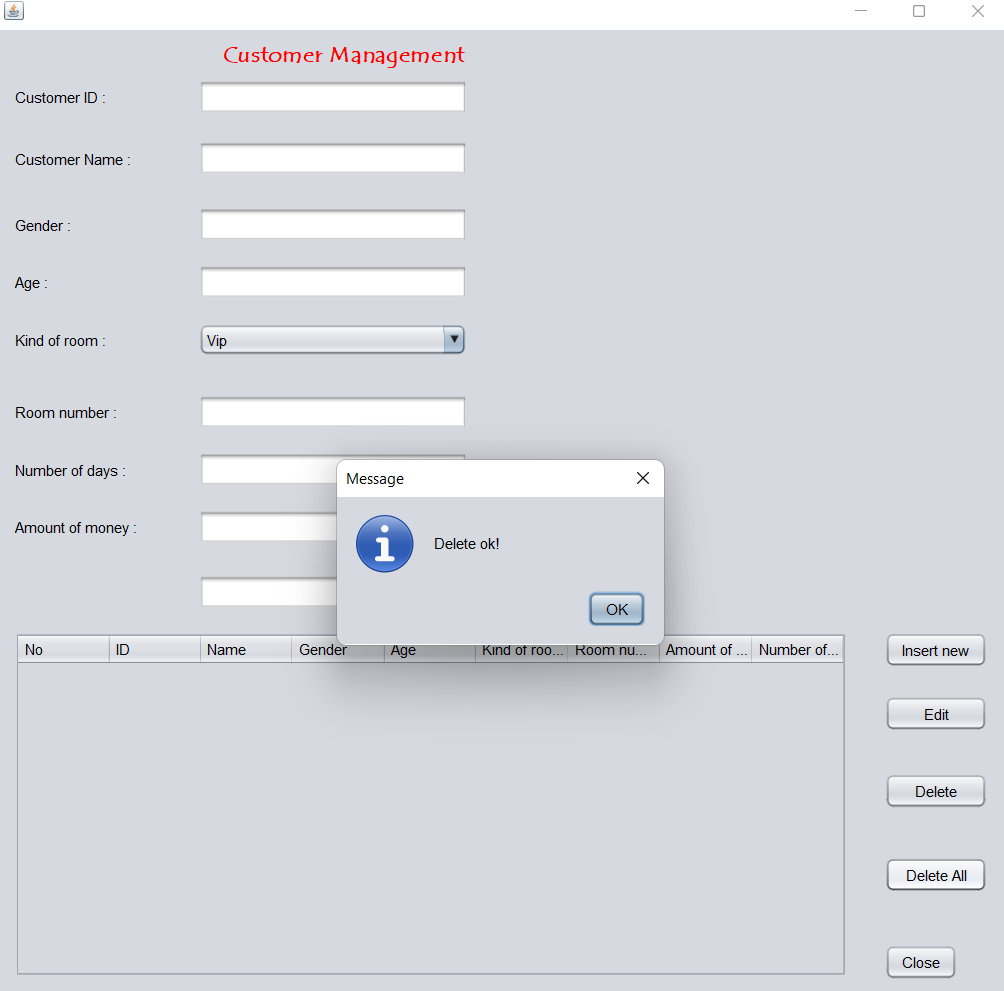












1. Conclusion

My program is complete and can help users manage customer information. This program has the functions of adding, updating, editing and deleting information. There is also a search function to help users control information more easily. However, there are still some limitations that need to be overcome such as: the interface is not attractive to viewers, lack of many functions, ... If given the opportunity, I will make the program to manage more elements and better interface.