Lab Exam

In this exam you are expected to develop a Java application which views and manages customers of a company. Your program will keep and operate on certain information of customers. A graphical user interface (GUI) is mandatory for this application. There will be 4 main functionalities and each has its own screen: Table (includes a filter), Insertion, Update and Delete.

Table

With usage of a table the data will be available for display. Each information field is a column and each customer is a row in this table. To complete this task **you have to use *javax.swing.JTable*, you cannot use another GUI component.**

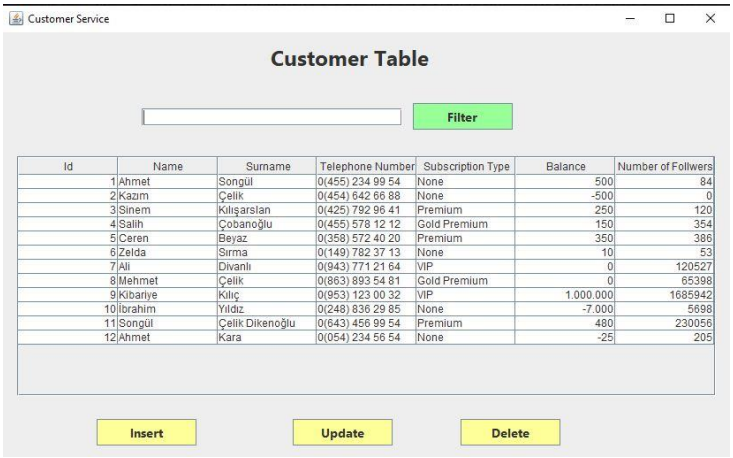

Figure 1: Table GUI

Filter:

Your table screen must include a filtering feature. It must take a value and filter the list of customers according to it. If there is any field of a customer which **contains** given filter value then it must stay on the table, otherwise the customer mustn't be displayed on table.

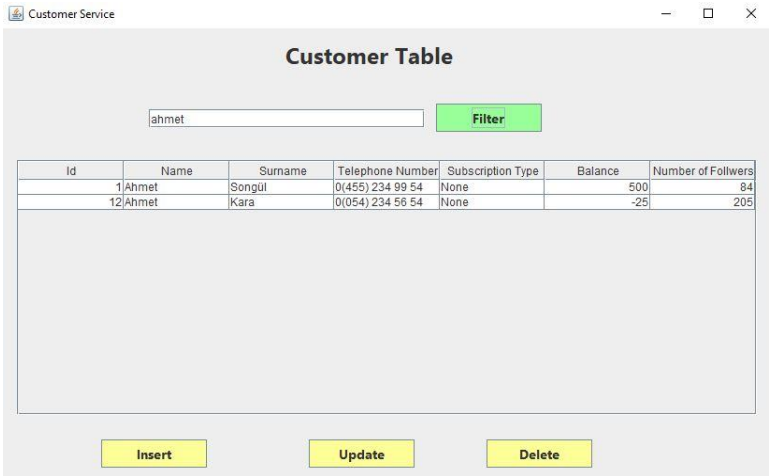

Figure 2: Table filter

Insertion

The second screen will be insertion of a customer. For this screen the values for all fields are expected to be filled. When insert button clicked, a customer with filled values will be inserted into the list and can be seen in table screen. **Subscription type component have to be a javax.swing.JComboBox and can only take certain values.** In this screen you can assume all fields will be filled by user, none will remain empty. Also, user will not mistake types for certain fields, for example balance field is of double type and user will not enter alphabetical characters and will only enter valid double numbers to that text field. Implementing these non-mandatory checks will not earn you extra points.

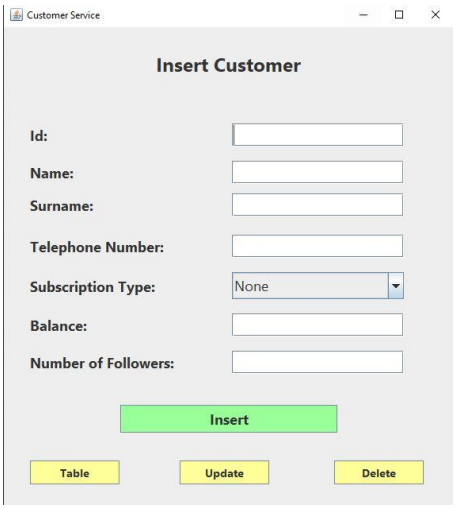

Figure 3: Insertion GUI

There is also a mandatory check for insertion. The duplicate id check has to be implemented on your application. Since id is a unique value, an id cannot belong to more than one customer. Thus, during insertion if an id which already exists is entered, then an error has to be given and insertion must not be completed. If there is no duplicate id error, then the insertion will be successful and a popup will appear which notifies the user.

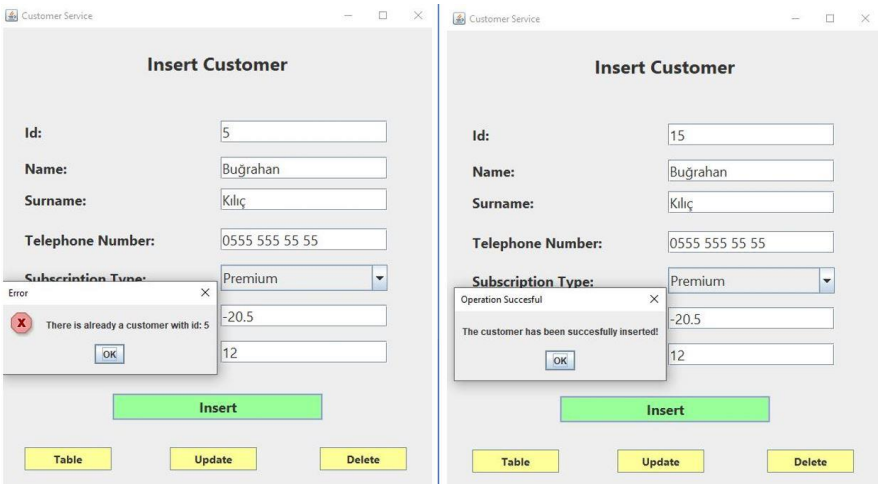

Figure 4: Insertion duplicate id error and successful operation popup

Update

The third functionality will be update. Update operation on GUI must be done in 2 steps: first get the customer to be updated by id, then fill the areas of the customer and update button will save new information on the customer. After an update, newly saved data must be seen in table screen. **Subscription type component have to be a javax.swing.JComboBox and can only take certain values.** In this screen you can assume all fields will be filled by user, none will remain empty. Also, user will not mistake types for certain fields, for example balance field is of double type and user will not enter alphabetical characters and will only enter valid double numbers. Implementing these non-mandatory checks will not earn you extra points.

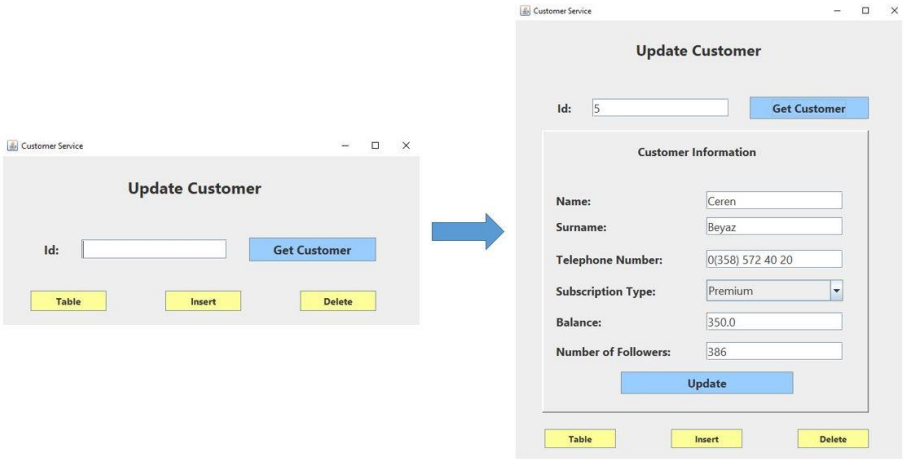

Figure 5: Update screen

Since getting the customer to update works by id, if there is no customer with given id an error popup will be displayed. If there are no errors and operation is successful a success message will be displayed.

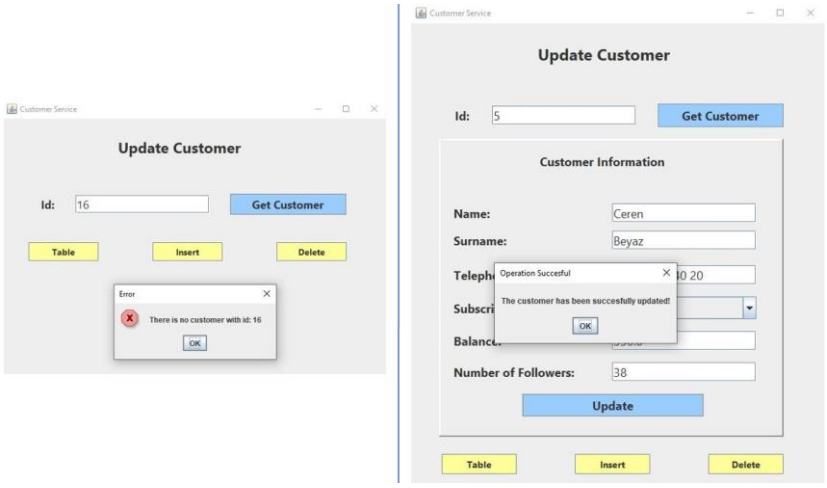

Figure 6: Update id error and success popups

Delete

The final functionality is delete. Similar to update screen it also works in two steps: get the customer by id and display his/her info then delete it. The difference is that the field values must not be editable like in update screen.

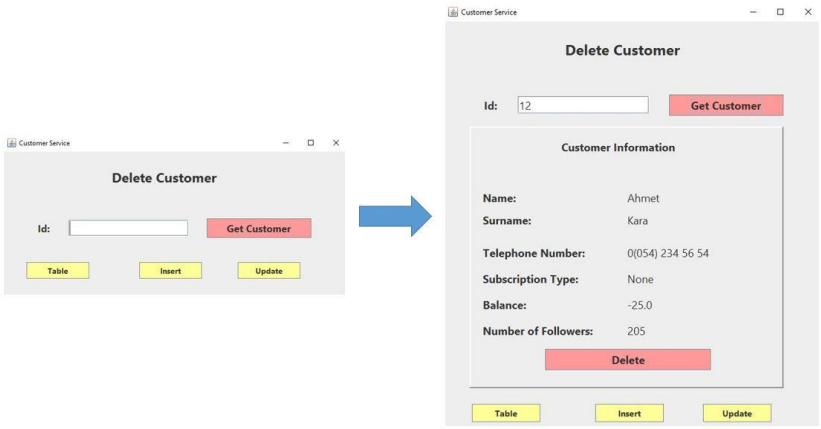

Figure 7: Delete screen

Since getting the customer to delete works by id, if there is no customer with given id an error popup will be displayed. If there are no errors and operation is successful a success message will be displayed.

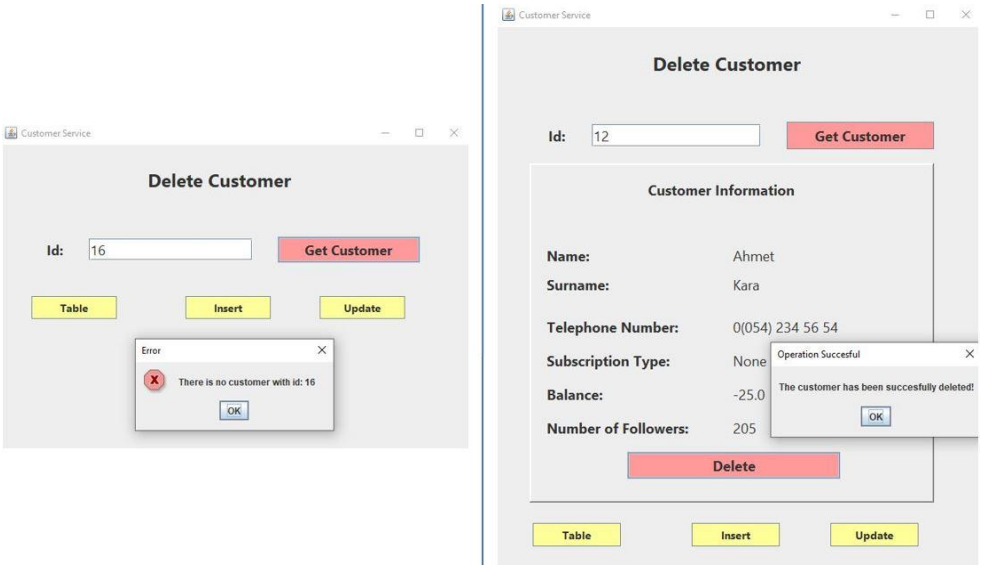

Figure 8: Delete id error and success popups

Constraints and Conditions

- As previously mentioned, certain places must have certain GUI components: Editable subscription type as JCombobox, customer table as JTable.
- Subscription Type (String) values are: *None*, *Premium*, *Gold Premium* and *VIP*.
- Only one screen must be visible at a time.
- In all 4 screens, as seen in all the figures there are 3 yellow buttons at the bottom. They provide access from one screen to other three screens anytime. When clicked, the current screen will close itself and the clicked screen will open. These buttons are mandatory to be implemented.
- Initial customer data is given in the **Model.java**, so you must include it as a class in your own project as well and put it into use when the program is first run.

Hint

You can check out last three lab videos.

Grading

Table Screen (25p)

- Insert – Update – Delete buttons (yellow buttons) (5p)
- Filter (20p)

Insertion Screen (25p)

- Table – Update – Delete buttons (yellow buttons) (5p)
- Insertion operation (10p)
- Operation successful message popup, as in *Figure 4* (5p)
- Duplicate id error popup, as in *Figure 4* (5p)

Update Screen (25p)

- Table – Insert – Delete buttons (yellow buttons) (5p)
- Update operation (10p)
- Operation successful message popup, as in *Figure 6* (5p)
- "There is no customer with id" error popup, as in *Figure 6* (5p)

Delete Screen (25p)

- Table – Insert – Update buttons (yellow buttons) (5p)
- Delete operation (10p)
- Operation successful message popup, as in *Figure 8* (5p)
- "There is no customer with id" error popup, as in *Figure 8* (5p)

Important Points

- All features are tested on the table. **If the table itself will not work, it would not be possible to check any feature, so you would get zero points (0p).** Before implementing the other features, you must first complete the table which displays the customer list. You can easily access the lab videos which can show you how to implement the GUI tables from the course page.
- Subscription Type GUI component in Insertion and Update screens has to be JCombobox. **If you implement these screens without using JCombobox, you would get minus ten points (-10p) on each feature (Insertion and Update), minus twenty points (-20p) total.**