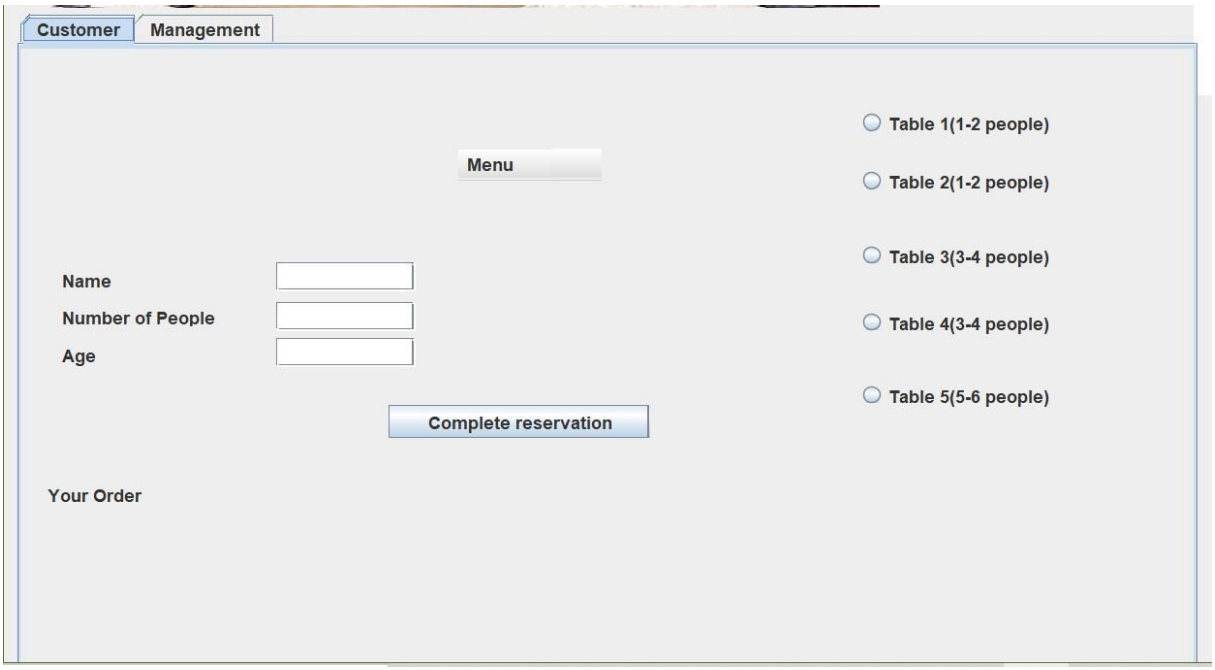
Esra Duran
71694 Comp132 -Term Project -2021Fall

Java Project Description

In this project, we have 2 different panels which are Customer and Manager Panels.

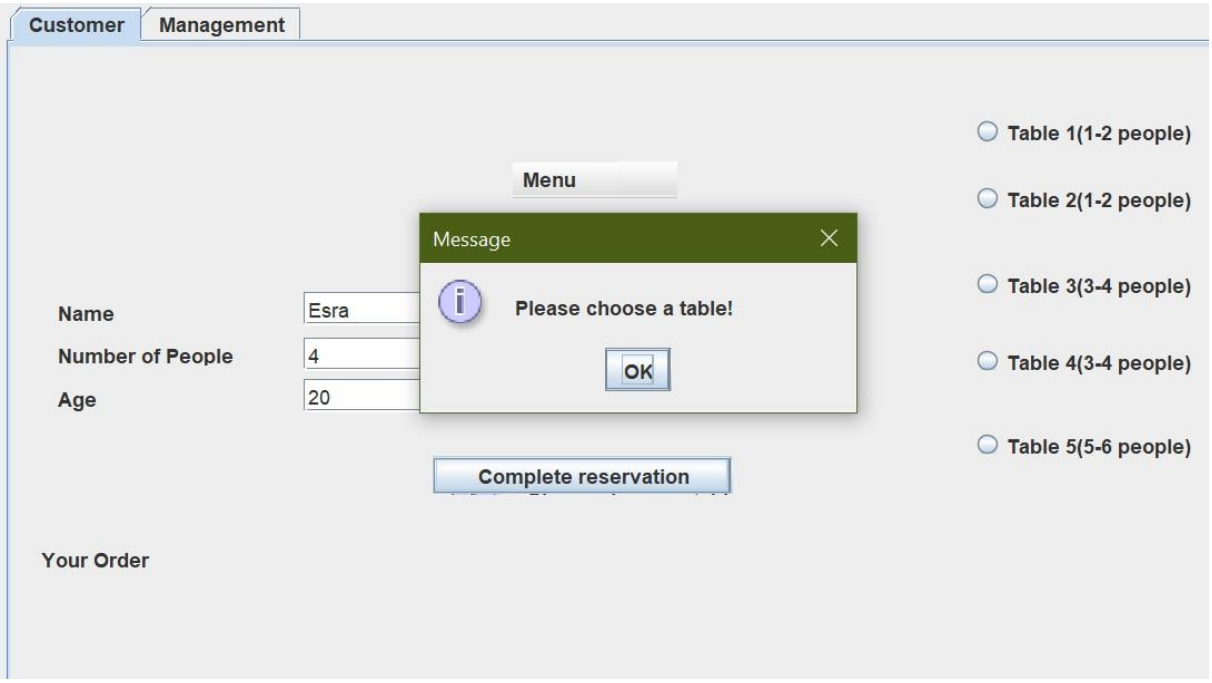
Customer Panel



There are 3 different blanks that we are getting inputs from the user. Also, we expect the user to select the proper table which are provided as JRadioButtons.

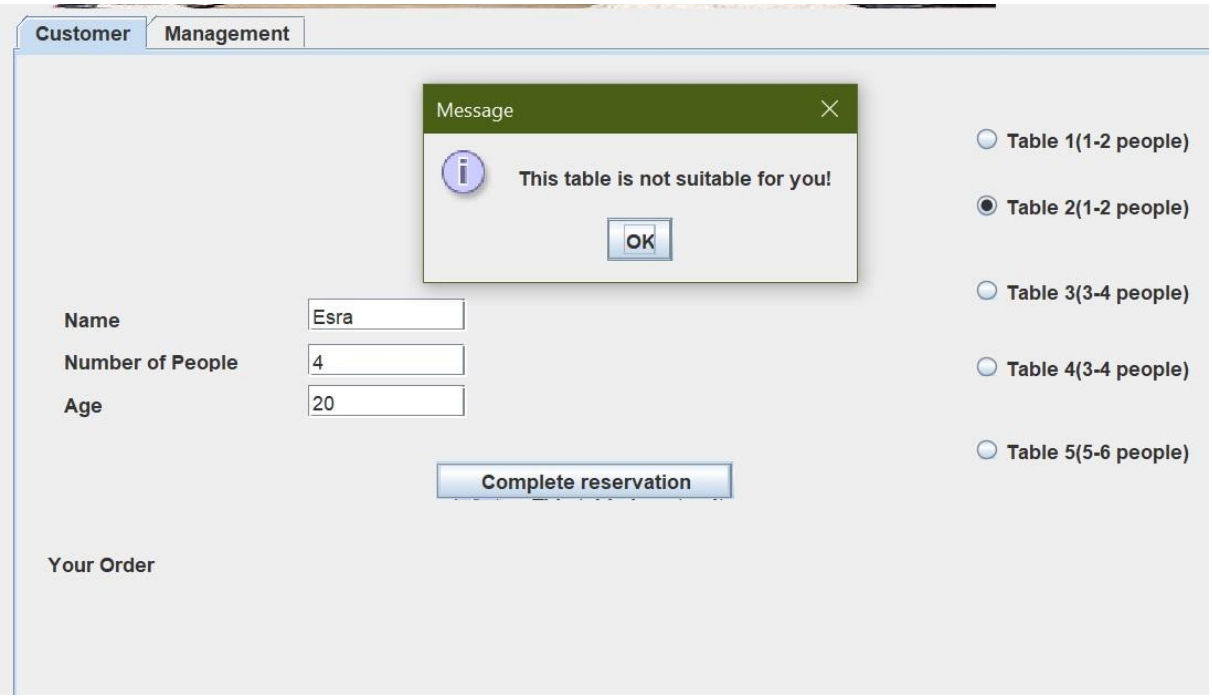
Errors

When user does not select a table;

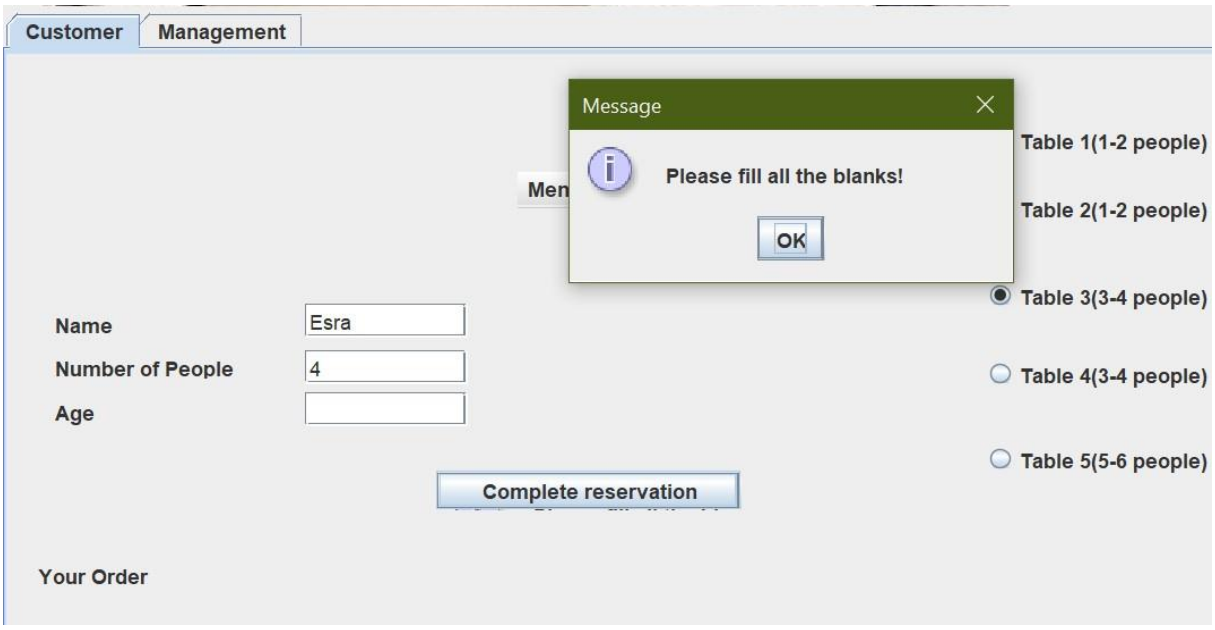


Esra Duran
71694 Comp132 -Term Project -2021Fall

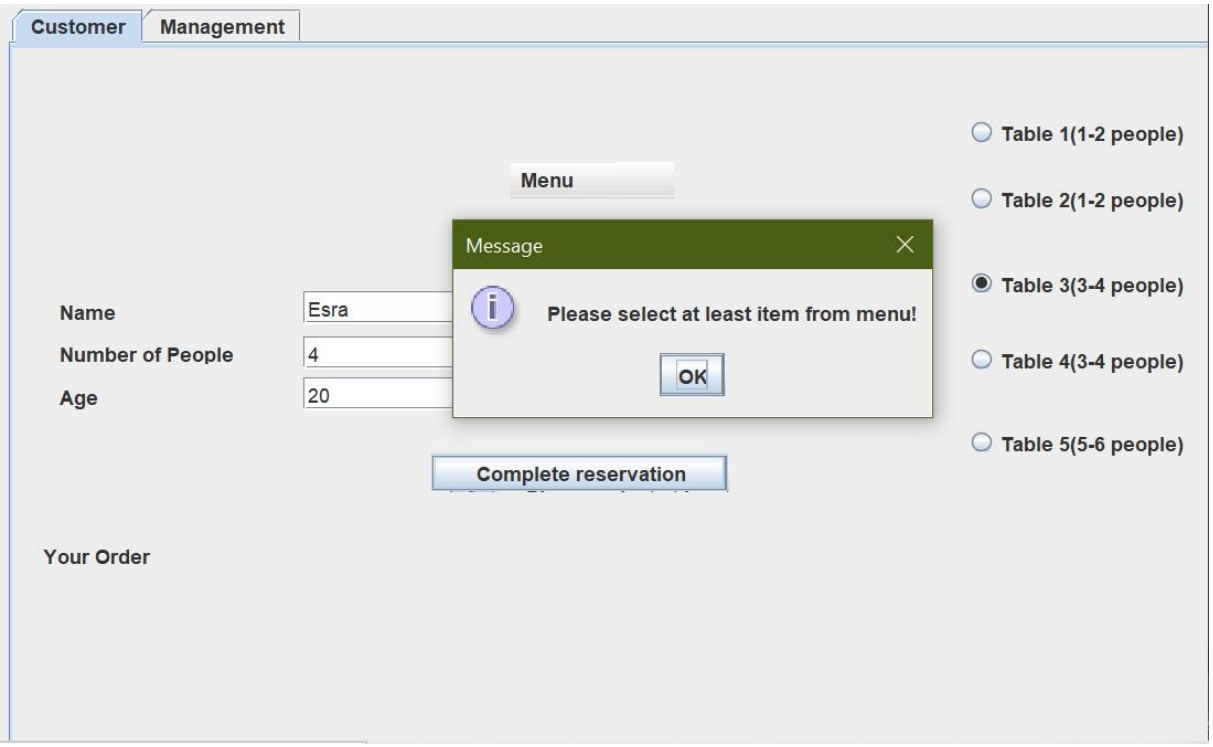
There are 5 tables in the restaurant. In order to select table1 or 2, the customer should enter the number of people as 1 or 2. Similarly, Table 5 only accepts the customers who are 5 or 6 people.



When user does not provide at least one information;.

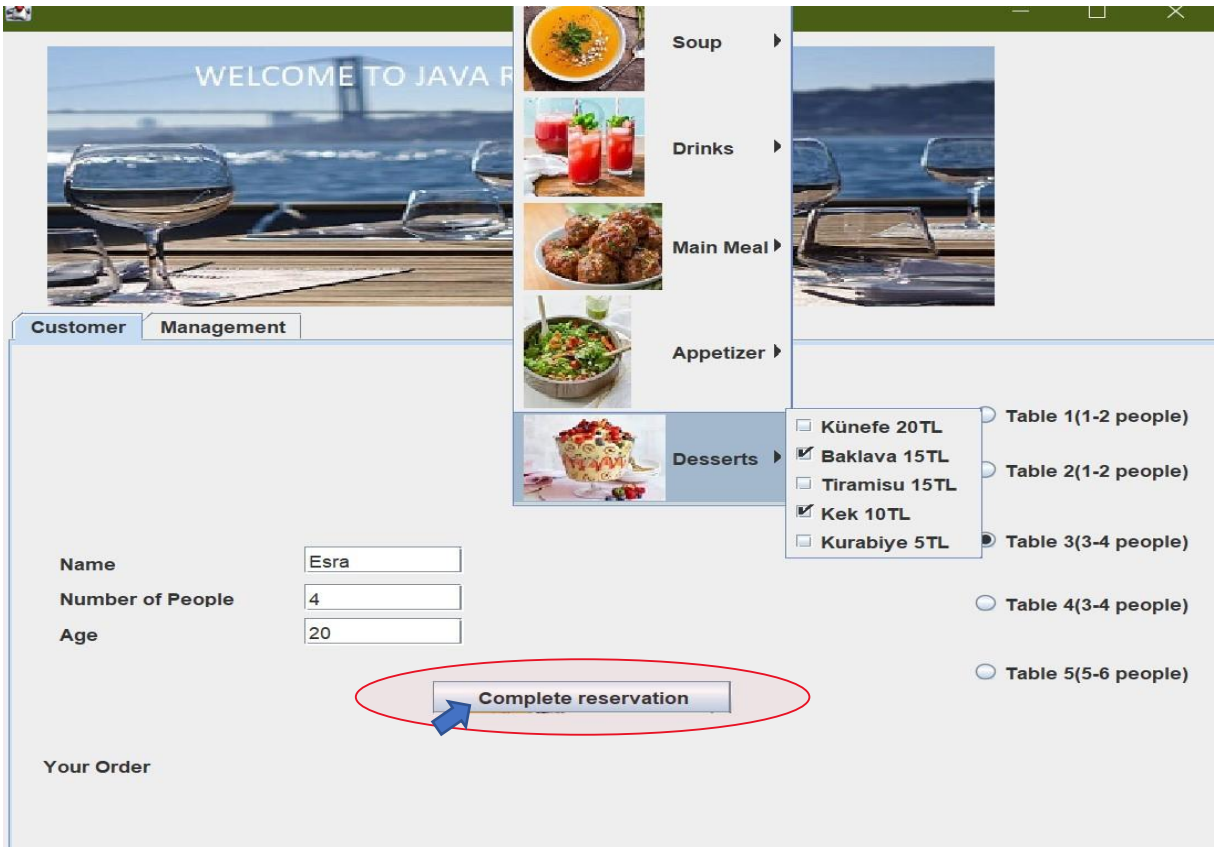


After we completed filling the blanks and selecting our table, we need to select item from the menu. If we press the “Complete reservation” without selecting any item we get that error;

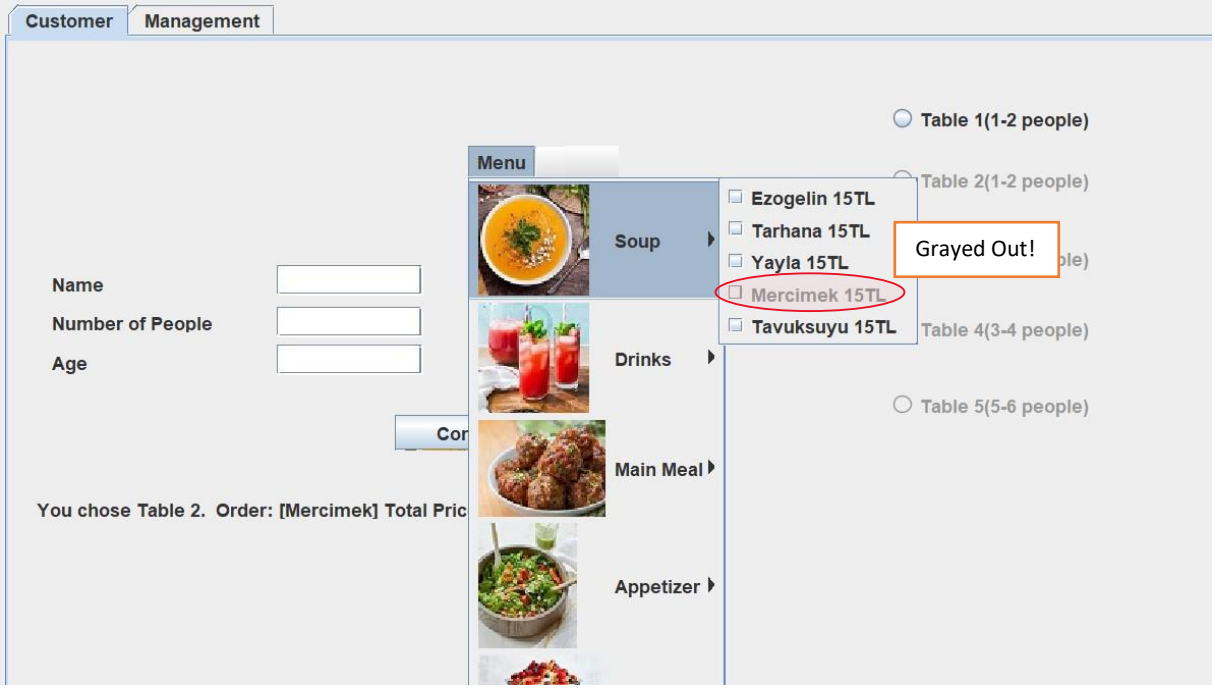


Selecting Item

When we click the “Menu”, we check the JCheckBoxMenuItems and then press the “Complete Reservation” button.☞

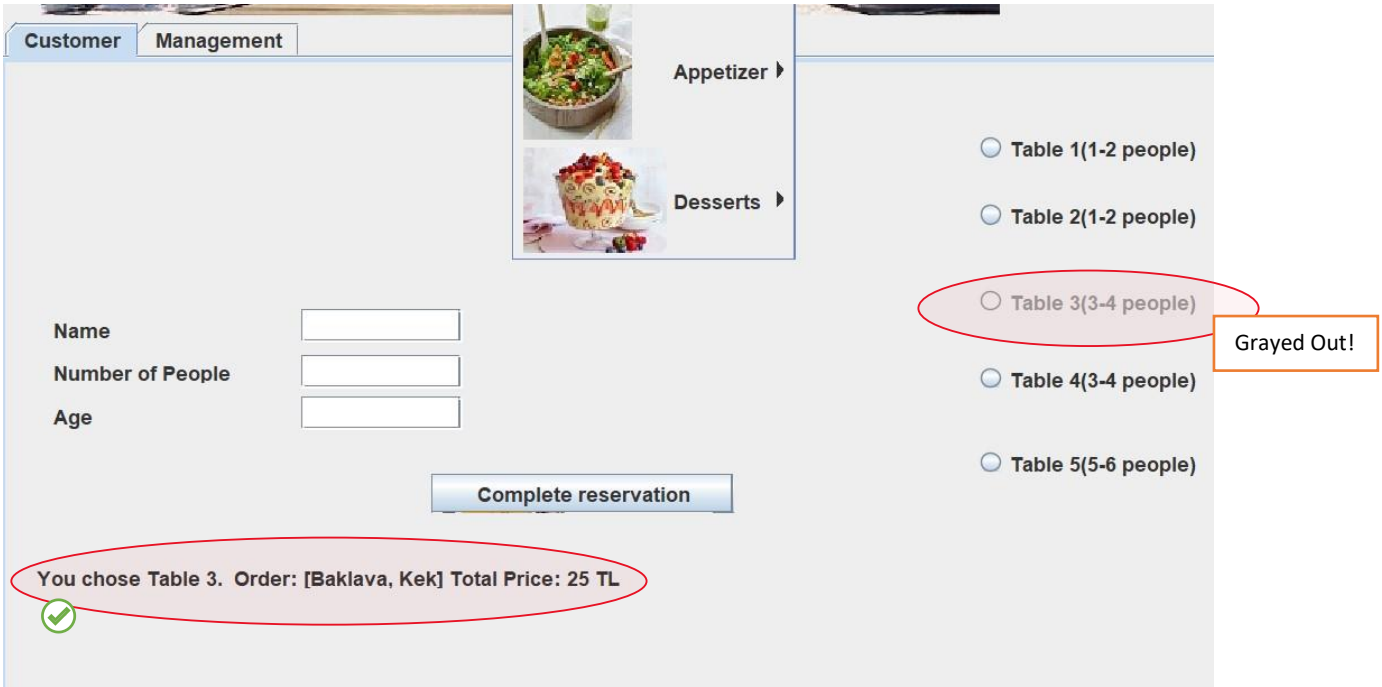


As seen below, when there is no stock left, the selections would be gray out and customers cannot select them.



Also, the selected tables would be disabled after the order is completed,

At the bottom, the customer can see their order and total prices.

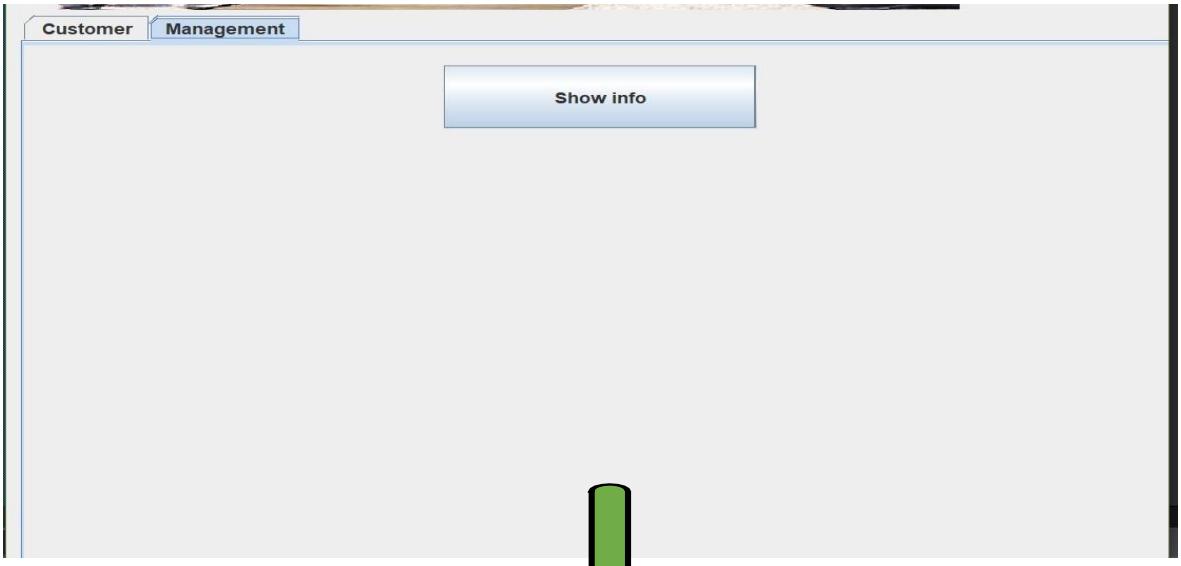


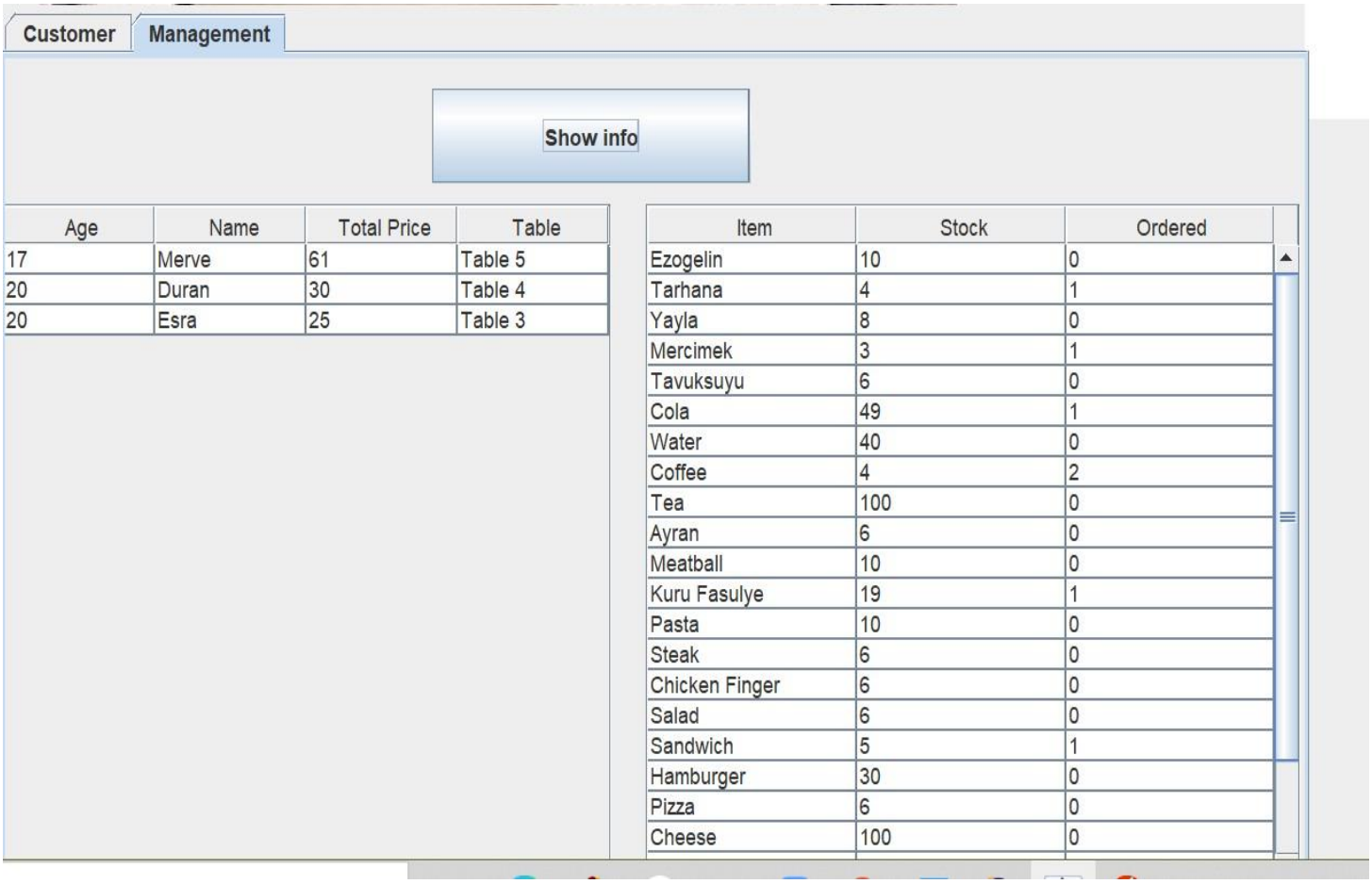
After creation of customer, the blanks and all the selections will be empty in order to register new customer.

Esra Duran
71694 Comp132 -Term Project -2021Fall

Manager Panel

The manager panel would be empty at first, when we click the “Show info” button, there appears a table with orders and also a table with stock information of items.



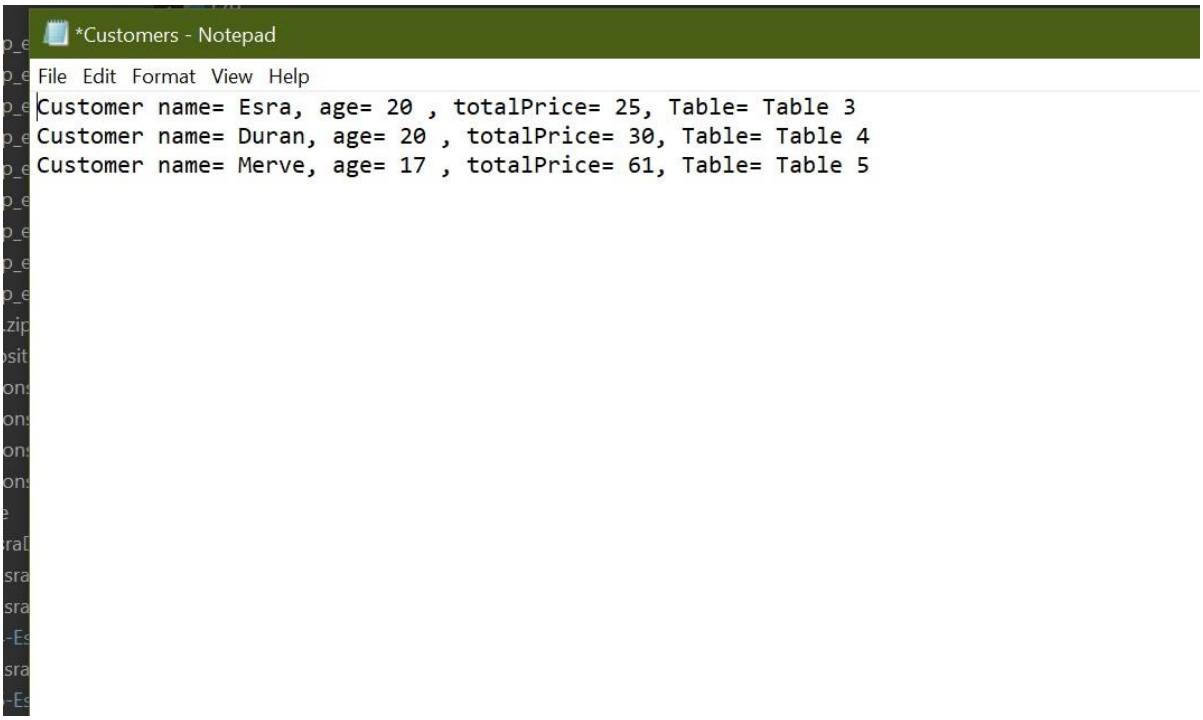


Age	Name	Total Price	Table
17	Merve	61	Table 5
20	Duran	30	Table 4
20	Esra	25	Table 3

Item	Stock	Ordered
Ezogelin	10	0
Tarhana	4	1
Yayla	8	0
Mercimek	3	1
Tavuksuyu	6	0
Cola	49	1
Water	40	0
Coffee	4	2
Tea	100	0
Ayran	6	0
Meatball	10	0
Kuru Fasulye	19	1
Pasta	10	0
Steak	6	0
Chicken Finger	6	0
Salad	6	0
Sandwich	5	1
Hamburger	30	0
Pizza	6	0
Cheese	100	0

Esra Duran
71694 Comp132 -Term Project -2021Fall

Note: Also, our codes creates a txt file that consists of each orders named ”Customers.txt”.



Esra Duran
71694 Comp132 -Term Project -2021Fall

PART 2

In this project there exists 3 classes.

Class Customer

```java
package src;

public class Customer {
    public String name;
    public int age;
    public int totalPrice;
    public String table;
    public Customer(String name, int age, int totalPrice, String table) {
        this.name = name;
        this.age = age;
        this.totalPrice = totalPrice;
        this.table=table;
    }
```

The customer class creates customer objects, however they are not be storage in the memory since each time we are creating the customer with the same name (Number of customers are not known).

We storage that customers information in an ArrayList in the Main class with the help of the Customer methods.

Class MenuItem

```java
package src;

import java.util.List;

public class MenuItem {
    public   int stock;
    public int price;
    public String name;
    public int decreasedstock;
    public MenuItem(int stock, int price,String name) {

        this.stock = stock;
        this.price = price;
        this.name=name;
        decreasedstock=stock;
    }

    public void decreaseStock() {
```

Each 25 items are objects of MenuItem and this class storages their attributes. There are stock and decreasedstock attributes that we will be using while identifying the number of order of each type of items.

Main Class

In main class, we create our project. First we start with creating objects from MenuItem class.

```
public void Project() throws Exception {

    List<String> orders= new ArrayList<>();


    ArrayList<MenuItem > menuitemList= new ArrayList<>();
    MenuItem Ezogelin = new MenuItem(10,15,"Ezogelin");
    menuitemList.add(Ezogelin);
    MenuItem Tarhana = new MenuItem(5,15,"Tarhana");
    menuitemList.add(Tarhana);
    MenuItem Yayla = new MenuItem(8,15,"Yayla");
    menuitemList.add(Yayla);
    MenuItem Mercimek= new MenuItem(4,15,"Mercimek");
    menuitemList.add(Mercimek);
    MenuItem Tavuksuyu = new MenuItem(6,15,"Tavuksuyu");
    menuitemList.add(Tavuksuyu);
```

Then, we created panels, menu and menuitems. As well as we inserted some images.

```
setResizable(true);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setBounds(100, 100, 1243, 726);
contentPane = new JPanel();
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
contentPane.setLayout(null);

JTabbedPane tabbedPane = new JTabbedPane(JTabbedPane.TOP);
tabbedPane.setBounds(10, 203, 812, 486);
contentPane.add(tabbedPane);

Image resim=new ImageIcon(Project.class.getResource("/imageRes.jpg")).getImage();
Image resim1=new ImageIcon(Project.class.getResource("/resim1.jpg")).getImage();
```

Esra Duran
71694 Comp132 -Term Project -2021Fall

We set the labels of CheckBoxMenuItems

```
JCheckBoxMenuItem soup1 = new JCheckBoxMenuItem("Ezogelin 15TL");
JCheckBoxMenuItem soup2 = new JCheckBoxMenuItem("Tarhana 15TL");
JCheckBoxMenuItem soup3 = new JCheckBoxMenuItem("Yayla 15TL");
JCheckBoxMenuItem soup4 = new JCheckBoxMenuItem("Mercimek 15TL");
JCheckBoxMenuItem soup5 = new JCheckBoxMenuItem("Tavuksuyu 15TL");
```

Then, we add ActionListener to the button" Complete Reservation". We check the conditions and display corresponding warnings.

```
Button.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        ArrayList<String> orderedmeals= new ArrayList<>();
        int totalprice=0;

        if (bg.getSelection()==null) {
            JOptionPane.showMessageDialog(null, "Please choose a table! ");
        }
        else if (Nametext.getText().isEmpty() || numtext.getText().isEmpty() || Agetext.getText().isEmpty()) {
            JOptionPane.showMessageDialog(null, "Please fill all the blanks! ");

        }
```

If all necessities are provided, we add the customer into list and set the selected table disabled.

```
if (selected!=0) {


        customerLabel.setText("You chose "+bg.getSelection().getActionCommand()
                        + ".  Order: "+orderedmeals.toString() + " Total Price: "+totalprice +" TL"
                        );

        Customer customer =new Customer(Nametext.getText(),Integer.parseInt(Agetext.getText())
                ,totalprice,bg.getSelection().getActionCommand());

        bg.getSelection().setEnabled(false);
        bg.clearSelection();
```

Esra Duran
71694 Comp132 -Term Project -2021Fall

We create a txt output file and append the information of new customer.

```
FileWriter fileWriter;
try {
    fileWriter = new FileWriter(file, true);
    BufferedWriter bWriter = new BufferedWriter(fileWriter);
    bWriter.write(customer.toString()+"\n");


     bWriter.close();
} catch (IOException e2) {

    e2.printStackTrace();
}
```

Then, we create new button named “Show info” in Management panel. We also call the method ActionListener for that button and creates ordered table for customers according to their ages (if same, descending order of total prices). Then, we generate the second table which shows the stocks and the number of ordered each type of foods.

```
orderButton.addActionListener(new ActionListener()    {

    @Override
    public void actionPerformed(ActionEvent e){

        Collections.sort(orders);

        for (int i=0;i<orders.size()-1;i++) {
            System.out.println();
            if (Integer.parseInt(orders.get(i).split("-")[0])==Integer.parse
                System.out.println("jer");
                if (Integer.parseInt(orders.get(i).split("-")[2]) < Integer.
                    String a=orders.get(i);
                    orders.set(i, orders.get(i+1));
                    orders.set(i+1, a);
                }
            }

        }
        String[][] data= new String[orders.size()][4];
        String[] colNames= {"Age","Name","Total Price","Table"};
        for(int i=0;i<orders.size();i++) {
            String[] a=orders.get(i).split("-");
            data[i]=a;


        }
        JTable jtable = new JTable(data,colNames);
```