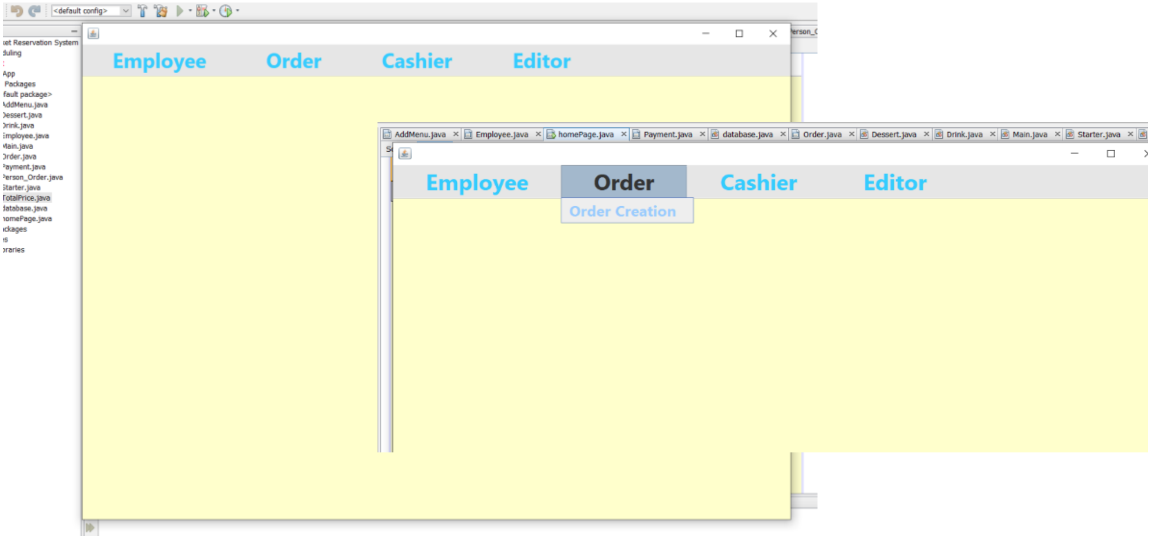
SOFTWARE ENGINEERING TOOLS REPORT

RESTAURANT APPLICATION

Arife Gül Yalçın

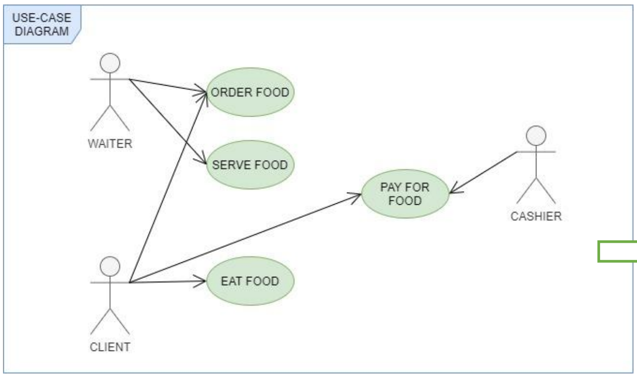
B1605.090054

First of all, I want to give information about our purpose in this application and what we want to see. In Restaurant Application;

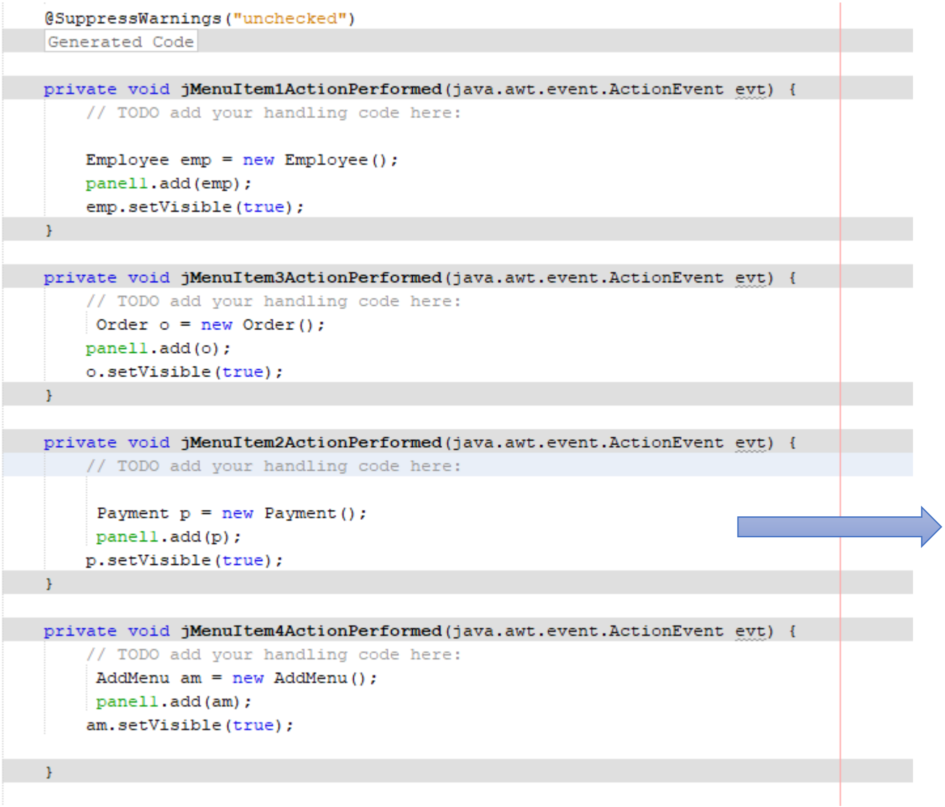


There are 4 different Menu and Menu Items;

- First, there is the option "Employee" and "AddEmployee" in it.
- Second, there is the option "Order" and "Order Creation" in it.
- Third, there is the option "Cashier" and "Payment" in it.
- Fourth, there is the option "Editor" and "Add Menu" in it.



Let's look at our Use-Case diagram that we prepared in the first part of our project. Here, with our 'AddEmployee' option, we can add people like Waiter, Cashier, Manager first.

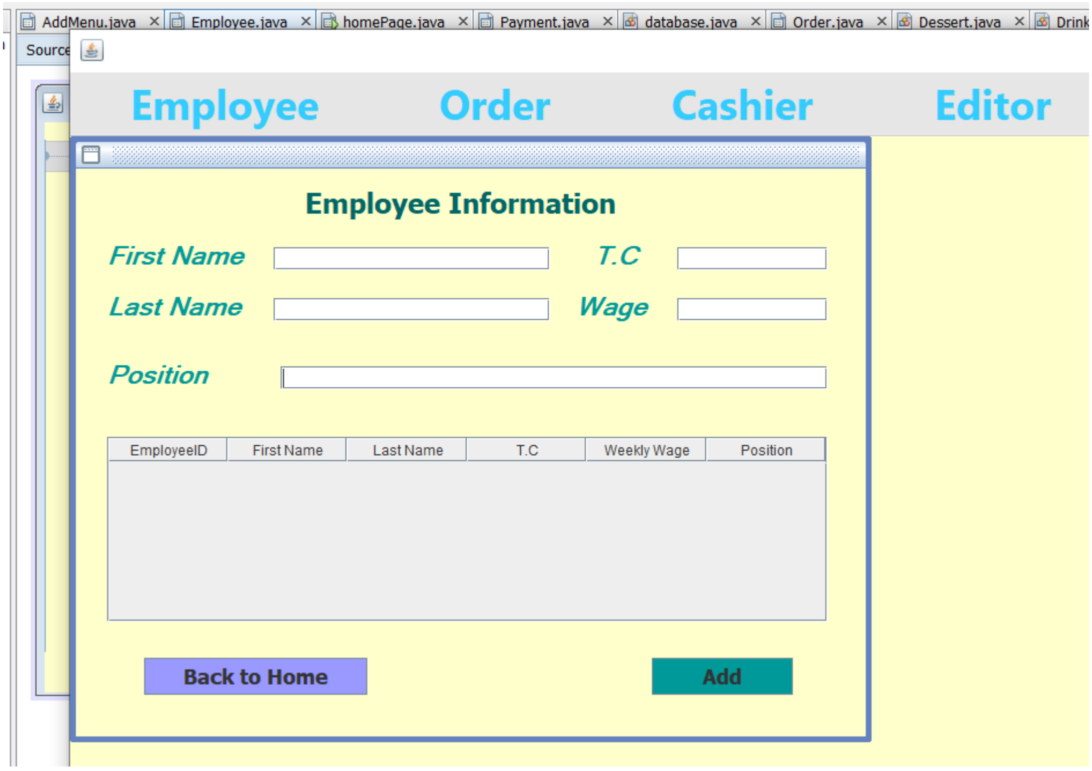


```
@SuppressWarnings("unchecked")
Generated Code

private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    Employee emp = new Employee();
    panel1.add(emp);
    emp.setVisible(true);
}

private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
     Order o = new Order();
    panel1.add(o);
    o.setVisible(true);
}

private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

     Payment p = new Payment();
     panel1.add(p);
    p.setVisible(true);
}

private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
     AddMenu am = new AddMenu();
     panel1.add(am);
    am.setVisible(true);

}
```

For example, let's see what happens when you click on Payment. Firstly, I create a new object(Payment). After that, I add this object to my panel1.(Panel name in the Home page)

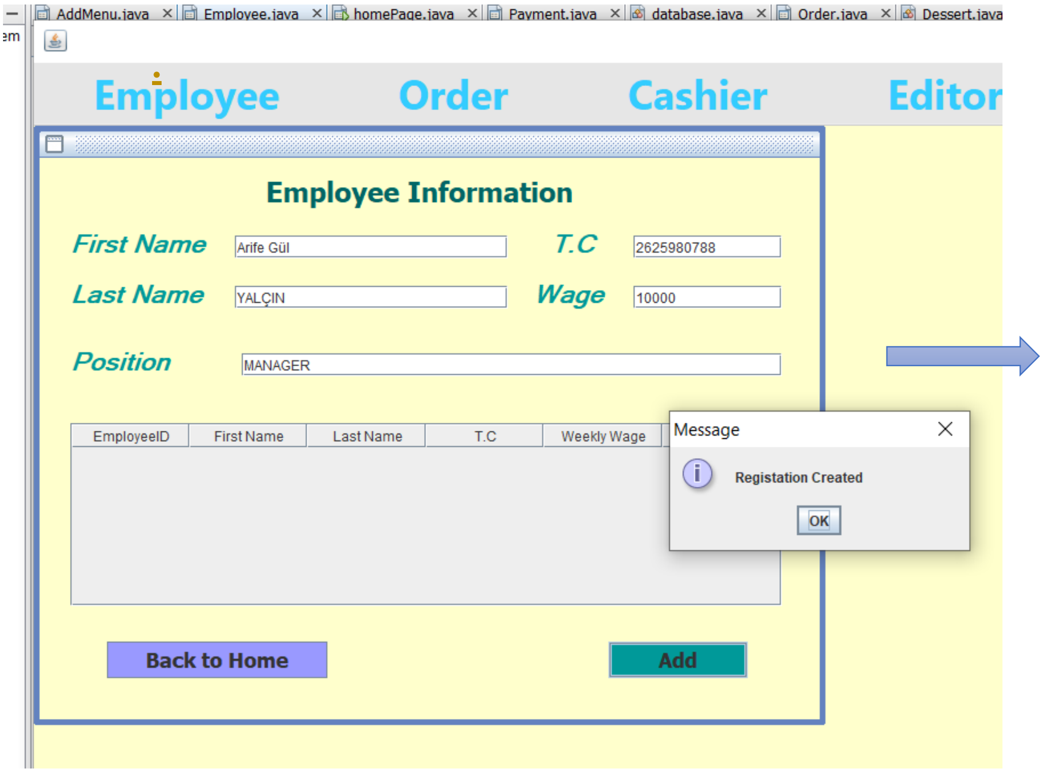
In this section, I showed how I have redirected to the pages that will open when I click on Menu Items.

FRAME 'ADD EMPLOYEE'



EmployeeID	First Name	Last Name	T.C	Weekly Wage	Position

How Does the Application Work?



Employee Information

First Name: Arife Gül

T.C: 2625980788

Last Name: YALÇIN

Wage: 10000

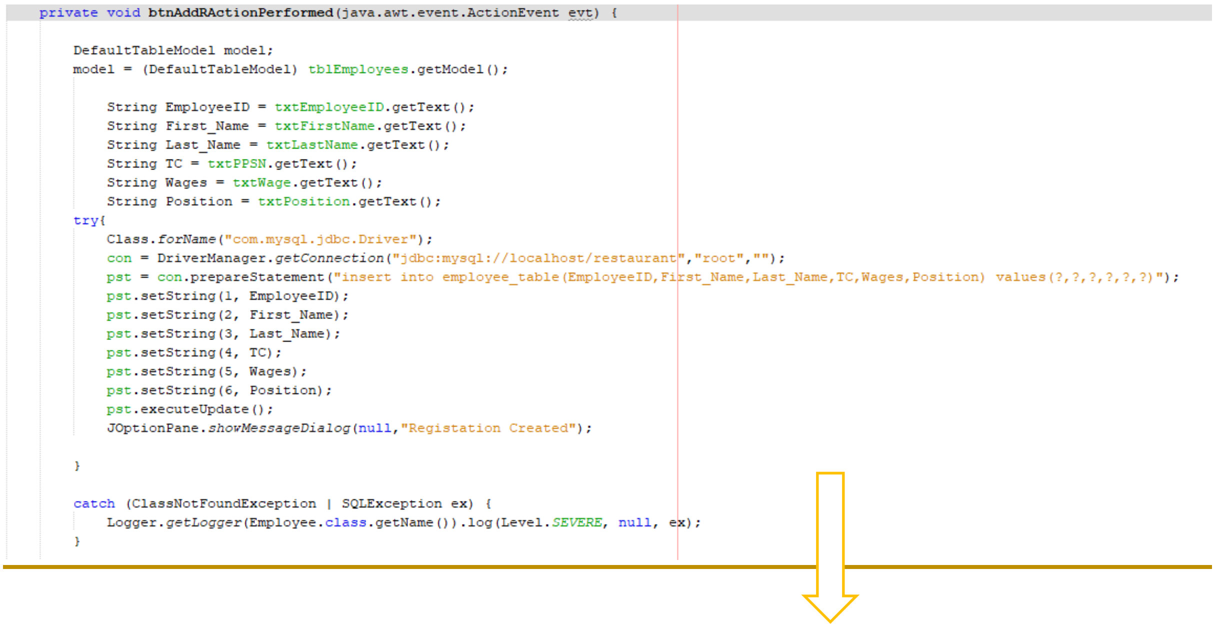
Position: MANAGER

EmployeeID	First Name	Last Name	T.C	Weekly Wage

Message

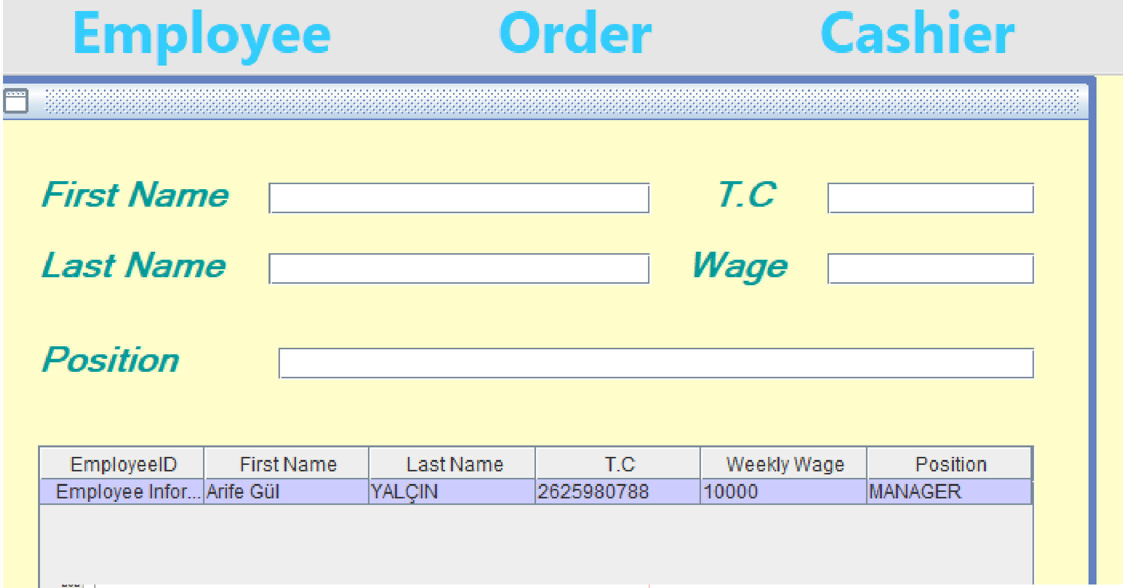
Registation Created

I enter the Employee information and click the Add button. 'Registration Created' message appears on the screen.

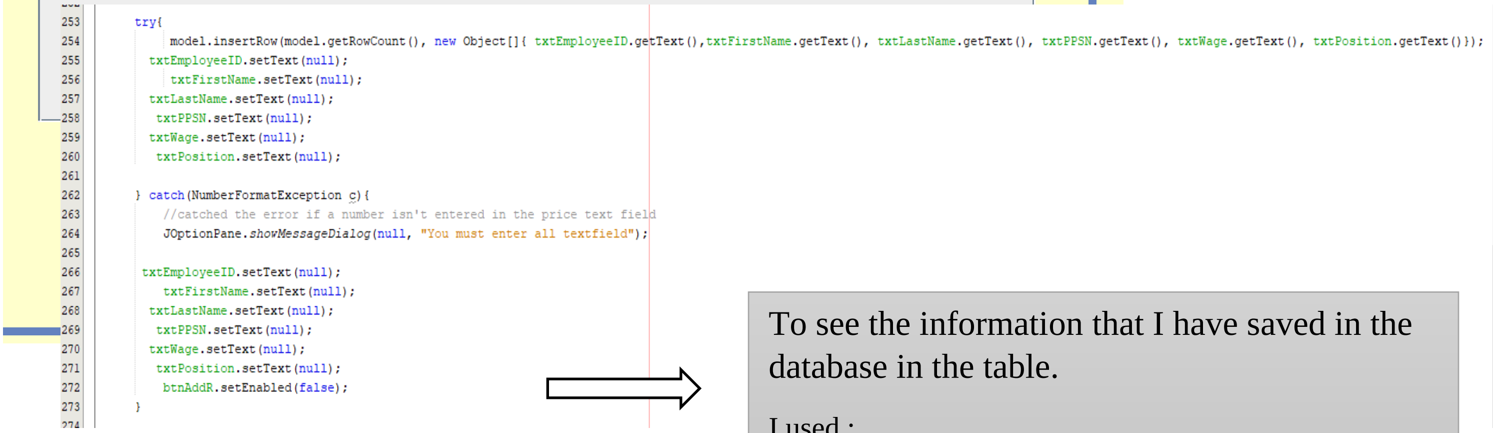


```
private void btnAddRActionPerformed(java.awt.event.ActionEvent evt) {

    DefaultTableModel model;
    model = (DefaultTableModel) tblEmployees.getModel();

        String EmployeeID = txtEmployeeID.getText();
        String First_Name = txtFirstName.getText();
        String Last_Name = txtLastName.getText();
        String TC = txtPPSN.getText();
        String Wages = txtWage.getText();
        String Position = txtPosition.getText();
    try{
        Class.forName("com.mysql.jdbc.Driver");
        con = DriverManager.getConnection("jdbc:mysql://localhost/restaurant","root","");
        pst = con.prepareStatement("insert into employee_table(EmployeeID,First_Name,Last_Name,TC,Wages,Position) values(?,?,?,?,?,?)");
        pst.setString(1, EmployeeID);
        pst.setString(2, First_Name);
        pst.setString(3, Last_Name);
        pst.setString(4, TC);
        pst.setString(5, Wages);
        pst.setString(6, Position);
        pst.executeUpdate();
        JOptionPane.showMessageDialog(null,"Registation Created");

    }

    catch (ClassNotFoundException | SQLException ex) {
        Logger.getLogger(Employee.class.getName()).log(Level.SEVERE, null, ex);
    }
```

In this section, the code of how the operation works when the 'Add' button is clicked. We connect to the **database**. With phpMyAdmin, we create separate tables for each. I used my 'employee_table' table in this section. I entered the information contained in it. Later, click the Add button. 'Registration Created' message appears on the panel.



EmployeeID	First Name	Last Name	T.C	Weekly Wage	Position
Employee Infor...	Arife Gül	YALÇIN	2625980788	10000	MANAGER



```
        try{
            model.insertRow(model.getRowCount(), new Object[]{ txtEmployeeID.getText(),txtFirstName.getText(), txtLastName.getText(), txtPPSN.getText(), txtWage.getText(), txtPosition.getText()});
          txtEmployeeID.setText(null);
             txtFirstName.setText(null);
          txtLastName.setText(null);
           txtPPSN.setText(null);
          txtWage.setText(null);
           txtPosition.setText(null);

        } catch(NumberFormatException c){
            //catched the error if a number isn't entered in the price text field
            JOptionPane.showMessageDialog(null, "You must enter all textfield");

         txtEmployeeID.setText(null);
            txtFirstName.setText(null);
          txtLastName.setText(null);
           txtPPSN.setText(null);
          txtWage.setText(null);
           txtPosition.setText(null);
            btnAddR.setEnabled(false);
        }
```

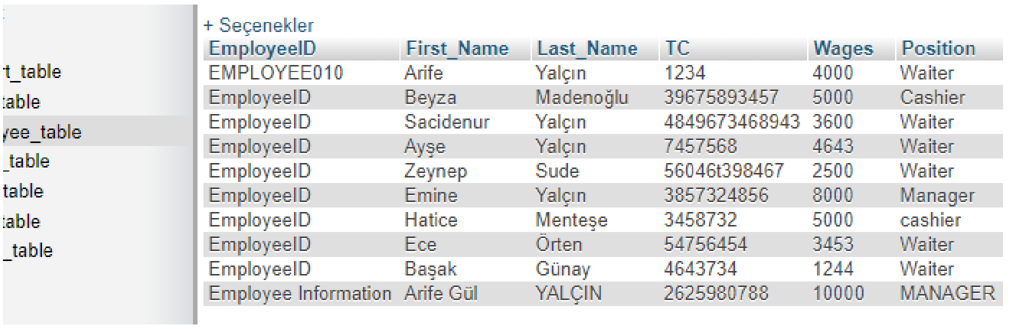
To see the information that I have saved in the database in the table.

I used ;

getText() ➔ Gets the information entered.

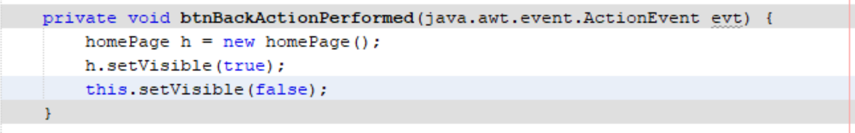
setText() ➔Paste the entered information into the table.

DATABASE 'ADD EMPLOYEE'



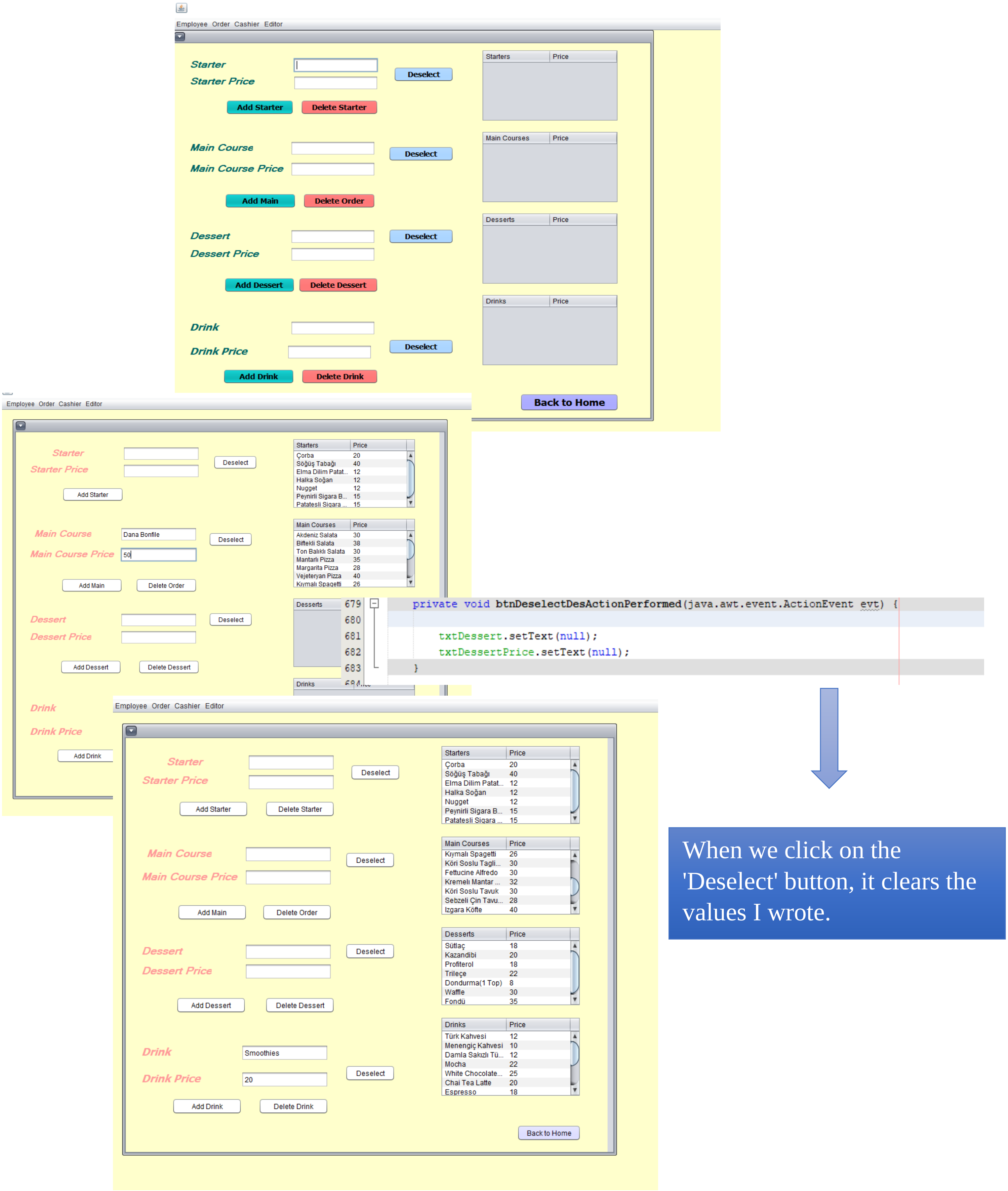
EmployeeID	First_Name	Last_Name	TC	Wages	Position
EMPLOYEE010	Arife	Yalçın	1234	4000	Waiter
EmployeeID	Beyza	Madenoğlu	39675893457	5000	Cashier
EmployeeID	Sacidenur	Yalçın	4849673468943	3600	Waiter
EmployeeID	Ayşe	Yalçın	7457568	4643	Waiter
EmployeeID	Zeynep	Sude	56046t398467	2500	Waiter
EmployeeID	Emine	Yalçın	3857324856	8000	Manager
EmployeeID	Hatice	Menteşe	3458732	5000	cashier
EmployeeID	Ece	Örten	54756454	3453	Waiter
EmployeeID	Başak	Günay	4643734	1244	Waiter
Employee Information	Arife Gül	YALÇIN	2625980788	10000	MANAGER

'BACK TO HOME' BUTTON



```
    private void btnBackActionPerformed(java.awt.event.ActionEvent evt) {
        homePage h = new homePage();
        h.setVisible(true);
        this.setVisible(false);
    }
```

FRAME ‘ADD MENU’



Main Course: Dana Bonfile

Main Course Price: 50

Starters	Price
Çorba	20
Söğüş Tabağı	40
Elma Dilim Patat...	12
Halka Soğan	12
Nugget	12
Peynirli Sigara B...	15
Patatesli Sigara ...	15

Main Courses	Price
Akdeniz Salata	30
Biftekli Salata	38
Ton Balıklı Salata	30
Mantarlı Pizza	35
Margarita Pizza	28
Vejeteryan Pizza	40
Kıymalı Spagetti	26

```
private void btnDeselectDesActionPerformed(java.awt.event.ActionEvent evt) {

    txtDessert.setText(null);
    txtDessertPrice.setText(null);
}
```

Drink: Smoothies

Drink Price: 20

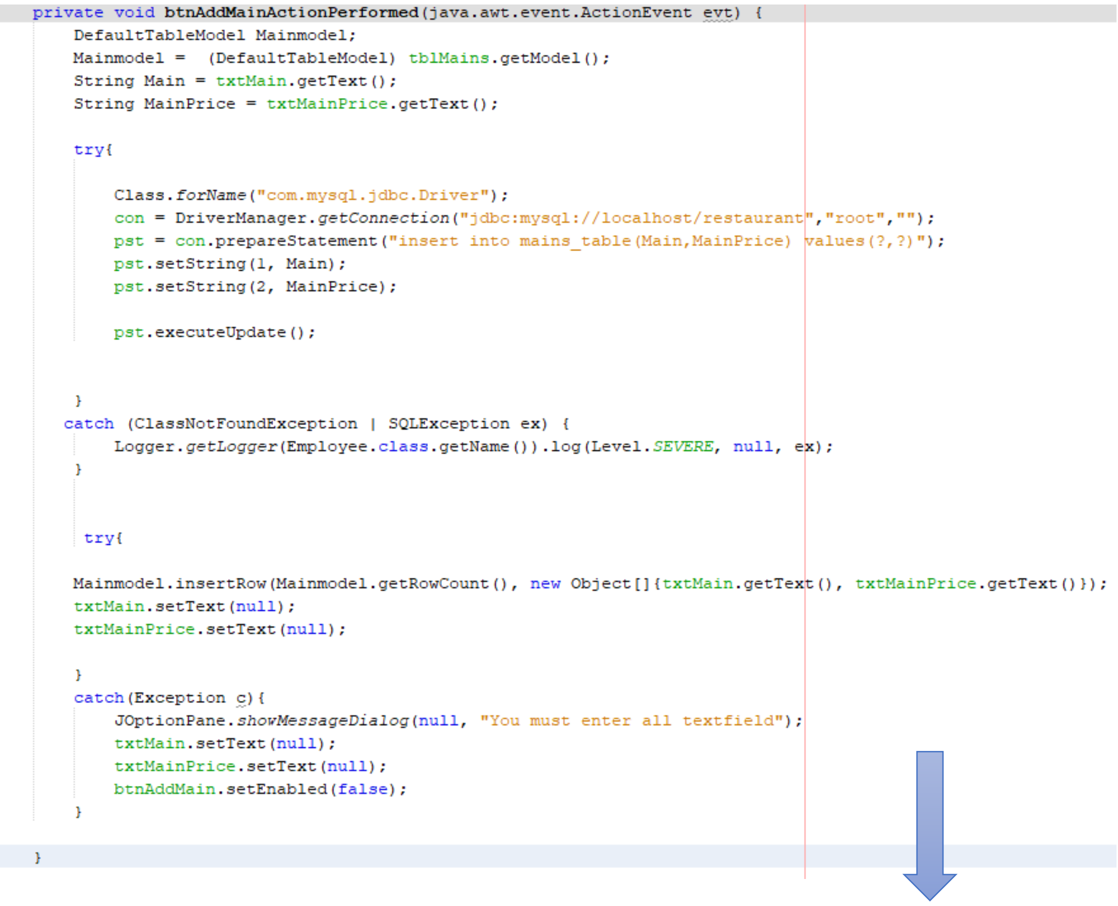
Starters	Price
Çorba	20
Söğüş Tabağı	40
Elma Dilim Patat...	12
Halka Soğan	12
Nugget	12
Peynirli Sigara B...	15
Patatesli Sigara ...	15

Main Courses	Price
Kıymalı Spagetti	26
Köri Soslu Tagli...	30
Fettucine Alfredo	30
Kremeli Mantar ...	32
Köri Soslu Tavuk	30
Sebzeli Çin Tavu...	28
Izgara Köfte	40

Desserts	Price
Sütlaç	18
Kazandibi	20
Profiterol	18
Trileçe	22
Dondurma(1 Top)	8
Waffle	30
Fondü	35

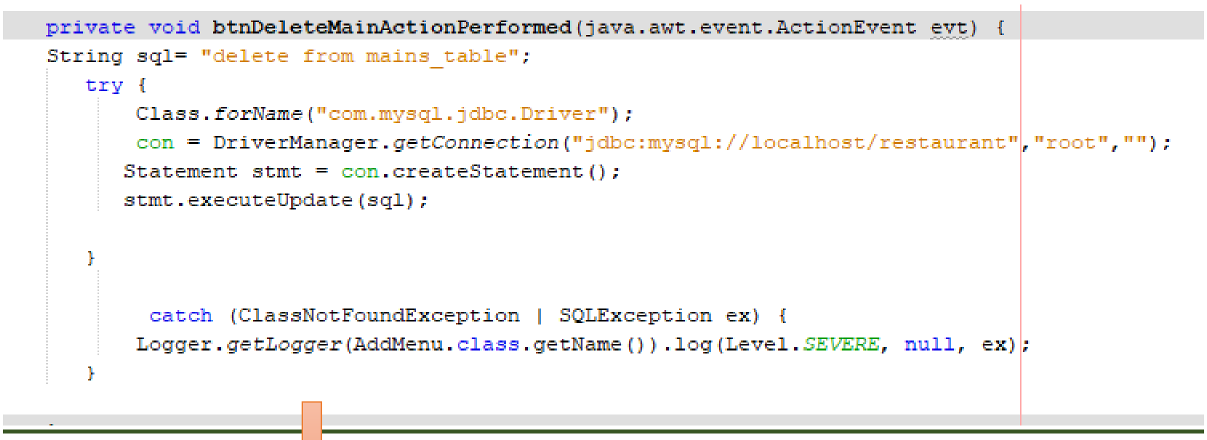
Drinks	Price
Türk Kahvesi	12
Menengiç Kahvesi	10
Damla Sakızlı Tü...	12
Mocha	22
White Chocolate...	25
Chai Tea Latte	20
Espresso	18

When we click on the 'Deselect' button, it clears the values I wrote.



```
private void btnAddMainActionPerformed(java.awt.event.ActionEvent evt) {
        DefaultTableModel Mainmodel;
        Mainmodel =  (DefaultTableModel) tblMains.getModel();
        String Main = txtMain.getText();
        String MainPrice = txtMainPrice.getText();

        try{

            Class.forName("com.mysql.jdbc.Driver");
            con = DriverManager.getConnection("jdbc:mysql://localhost/restaurant","root","");
            pst = con.prepareStatement("insert into mains_table(Main,MainPrice) values(?,?)");
            pst.setString(1, Main);
            pst.setString(2, MainPrice);

            pst.executeUpdate();


        }
       catch (ClassNotFoundException | SQLException ex) {
            Logger.getLogger(Employee.class.getName()).log(Level.SEVERE, null, ex);
        }


         try{

        Mainmodel.insertRow(Mainmodel.getRowCount(), new Object[]{txtMain.getText(), txtMainPrice.getText()});
        txtMain.setText(null);
        txtMainPrice.setText(null);

        }
        catch(Exception c){
            JOptionPane.showMessageDialog(null, "You must enter all textfield");
            txtMain.setText(null);
            txtMainPrice.setText(null);
            btnAddMain.setEnabled(false);
        }

    }
```

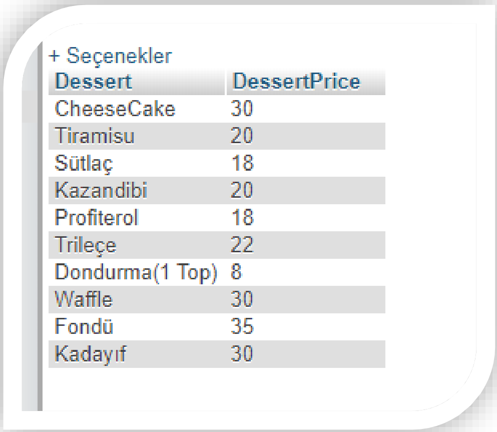
In this section, the code of how the operation works when the 'Add' button is clicked. We connect to the **database**. With phpMyAdmin, I used my 'mains_table' table in this section. I entered the information contained in it.



```
 private void btnDeleteMainActionPerformed(java.awt.event.ActionEvent evt) {
 String sql= "delete from mains_table";
     try {
         Class.forName("com.mysql.jdbc.Driver");
         con = DriverManager.getConnection("jdbc:mysql://localhost/restaurant","root","");
        Statement stmt = con.createStatement();
        stmt.executeUpdate(sql);


     }


          catch (ClassNotFoundException | SQLException ex) {
         Logger.getLogger(AddMenu.class.getName()).log(Level.SEVERE, null, ex);
     }
```

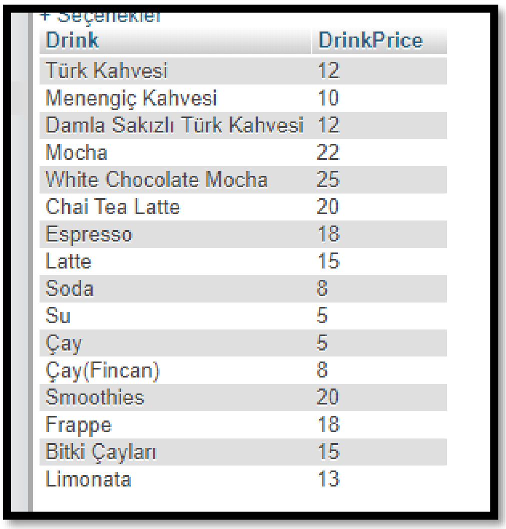
When the 'DELETE' button is clicked. We connect to the **database**. With phpMyAdmin, I used my 'mains_table' table in this section. Delete it the information.

DATABASE ‘ADD MENU’



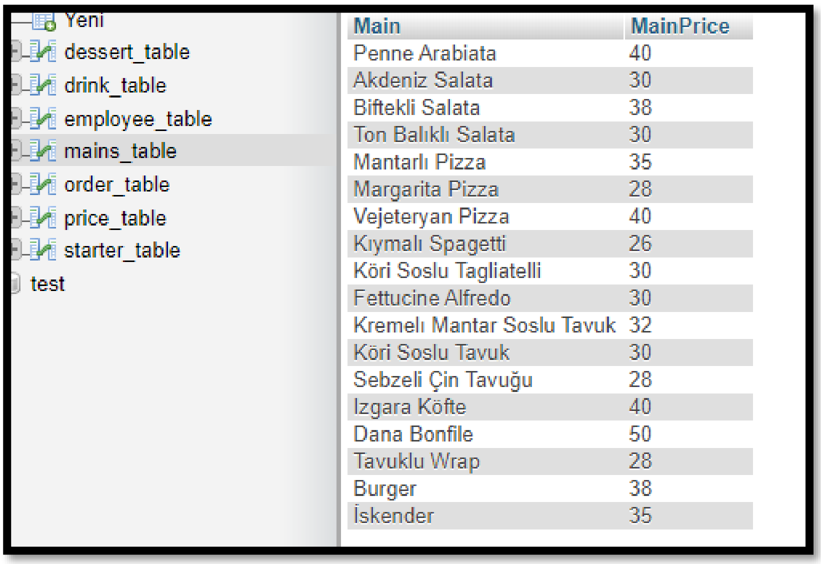
+ Seçenekler

Dessert	DessertPrice
CheeseCake	30
Tiramisu	20
Sütlaç	18
Kazandibi	20
Profiterol	18
Trileçe	22
Dondurma(1 Top)	8
Waffle	30
Fondü	35
Kadayıf	30

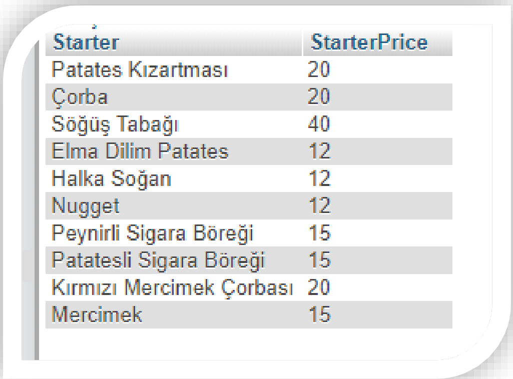


+ Seçenekler

Drink	DrinkPrice
Türk Kahvesi	12
Menengiç Kahvesi	10
Damla Sakızlı Türk Kahvesi	12
Mocha	22
White Chocolate Mocha	25
Chai Tea Latte	20
Espresso	18
Latte	15
Soda	8
Su	5
Çay	5
Çay(Fincan)	8
Smoothies	20
Frappe	18
Bitki Çayları	15
Limonata	13

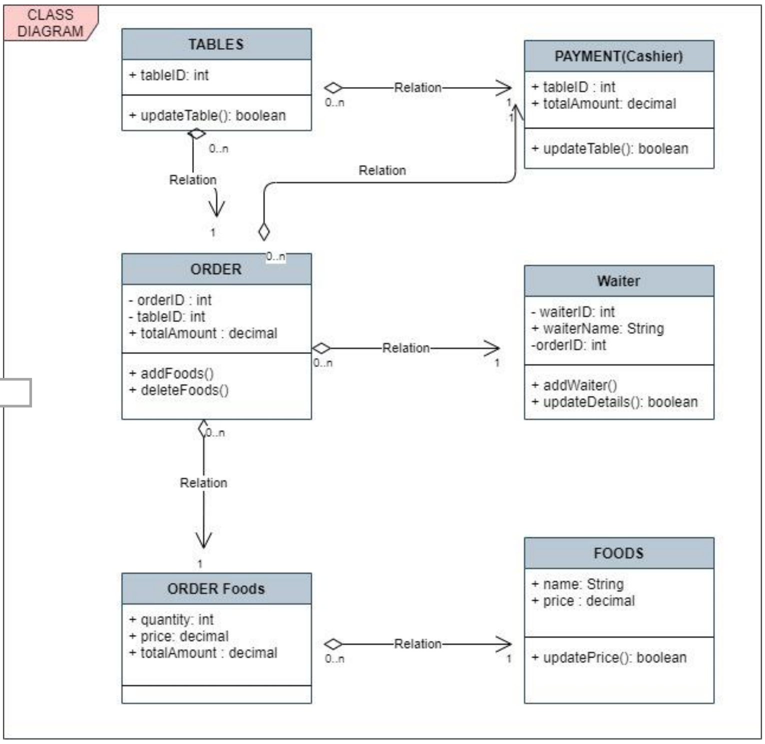


Main	MainPrice
Penne Arabiata	40
Akdeniz Salata	30
Biftekli Salata	38
Ton Balıklı Salata	30
Mantarlı Pizza	35
Margarita Pizza	28
Vejeteryan Pizza	40
Kıymalı Spagetti	26
Köri Soslu Tagliatelli	30
Fettucine Alfredo	30
Kremelı Mantar Soslu Tavuk	32
Köri Soslu Tavuk	30
Sebzeli Çin Tavuğu	28
Izgara Köfte	40
Dana Bonfile	50
Tavuklu Wrap	28
Burger	38
İskender	35

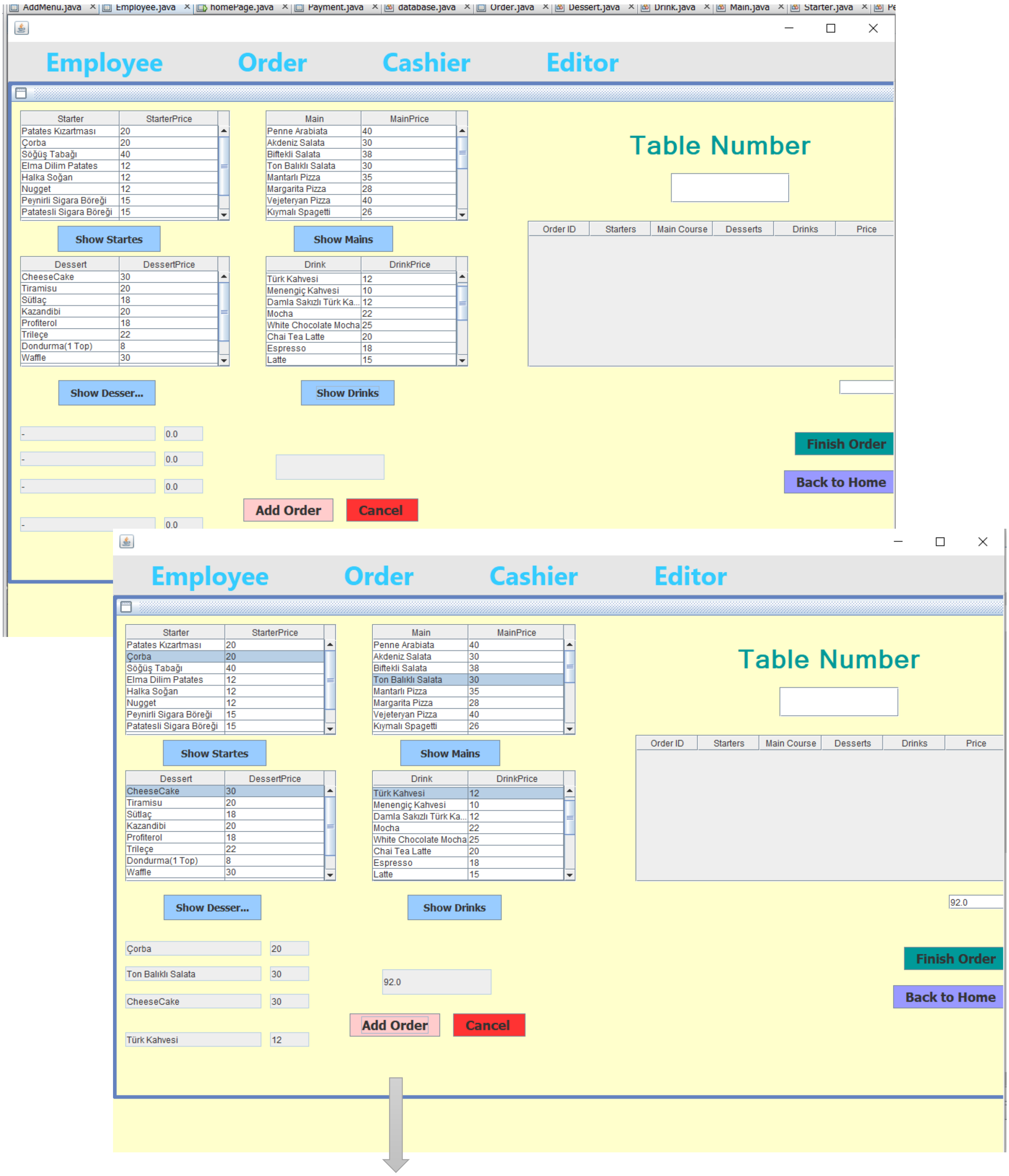


Starter	StarterPrice
Patates Kızartması	20
Çorba	20
Söğüş Tabağı	40
Elma Dilim Patates	12
Halka Soğan	12
Nugget	12
Peynirli Sigara Böreği	15
Patatesli Sigara Böreği	15
Kırmızı Mercimek Çorbası	20
Mercimek	15

Let's look at our Use-Case diagram that we prepared in the first part of our project. Here, with our 'Add Menu' option, we can add food like Starter, Main, Desert etc.



FRAME 'ORDER'



Starter	StarterPrice
Patates Kizartması	20
Çorba	20
Söğüş Tabağı	40
Elma Dilim Patates	12
Halka Soğan	12
Nugget	12
Peynirli Sigara Böreği	15
Patatesli Sigara Böreği	15

Main	MainPrice
Penne Arabiata	40
Akdeniz Salata	30
Biftekli Salata	38
Ton Balıklı Salata	30
Mantarlı Pizza	35
Margarita Pizza	28
Vejeteryan Pizza	40
Kıymalı Spagetti	26

Dessert	DessertPrice
CheeseCake	30
Tiramisu	20
Sütlaç	18
Kazandibi	20
Profiterol	18
Trileçe	22
Dondurma(1 Top)	8
Waffle	30

Drink	DrinkPrice
Türk Kahvesi	12
Menengiç Kahvesi	10
Damla Sakızlı Türk Ka...	12
Mocha	22
White Chocolate Mocha	25
Chai Tea Latte	20
Espresso	18
Latte	15

Order ID	Starters	Main Course	Desserts	Drinks	Price

-	0.0
-	0.0
-	0.0
-	0.0

Starter	StarterPrice
Patates Kizartması	20
Çorba	20
Söğüş Tabağı	40
Elma Dilim Patates	12
Halka Soğan	12
Nugget	12
Peynirli Sigara Böreği	15
Patatesli Sigara Böreği	15

Main	MainPrice
Penne Arabiata	40
Akdeniz Salata	30
Biftekli Salata	38
Ton Balıklı Salata	30
Mantarlı Pizza	35
Margarita Pizza	28
Vejeteryan Pizza	40
Kıymalı Spagetti	26

Dessert	DessertPrice
CheeseCake	30
Tiramisu	20
Sütlaç	18
Kazandibi	20
Profiterol	18
Trileçe	22
Dondurma(1 Top)	8
Waffle	30

Drink	DrinkPrice
Türk Kahvesi	12
Menengiç Kahvesi	10
Damla Sakızlı Türk Ka...	12
Mocha	22
White Chocolate Mocha	25
Chai Tea Latte	20
Espresso	18
Latte	15

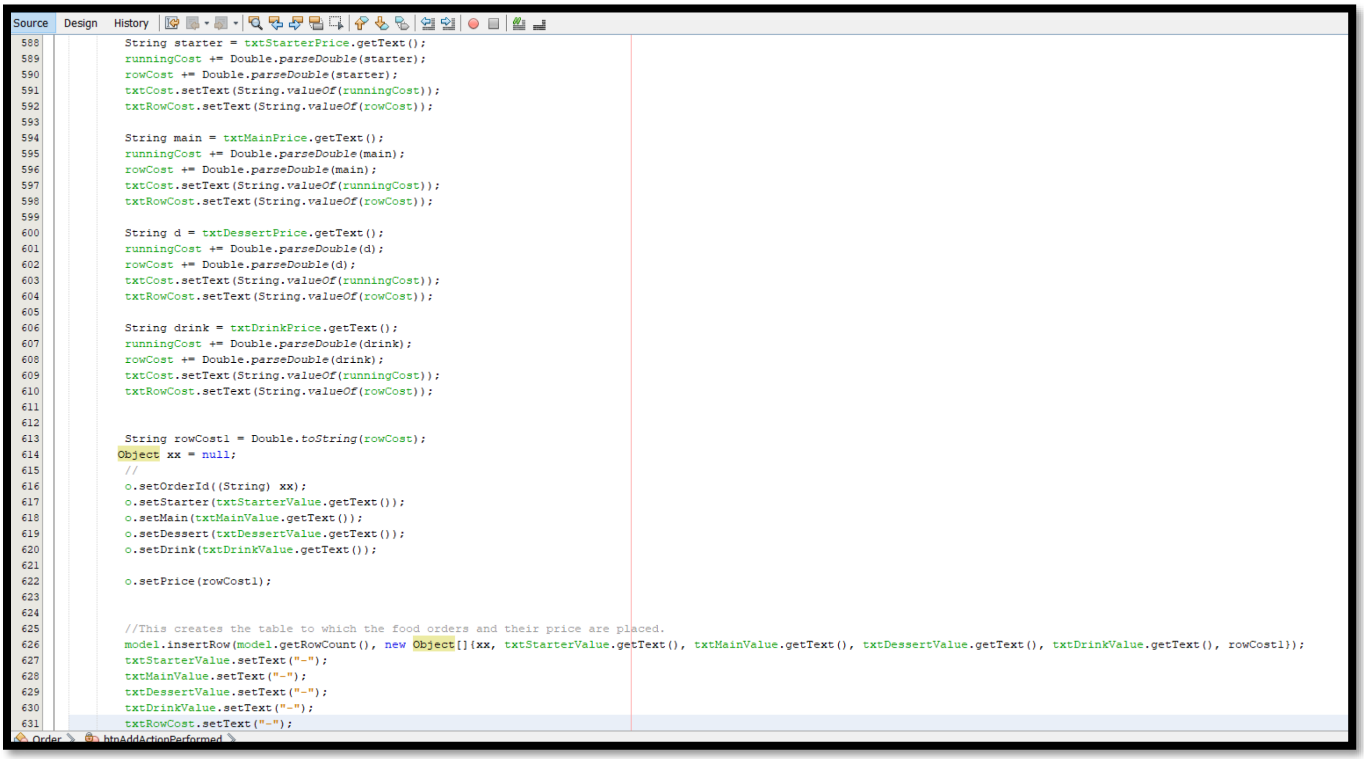
Order ID	Starters	Main Course	Desserts	Drinks	Price

92.0

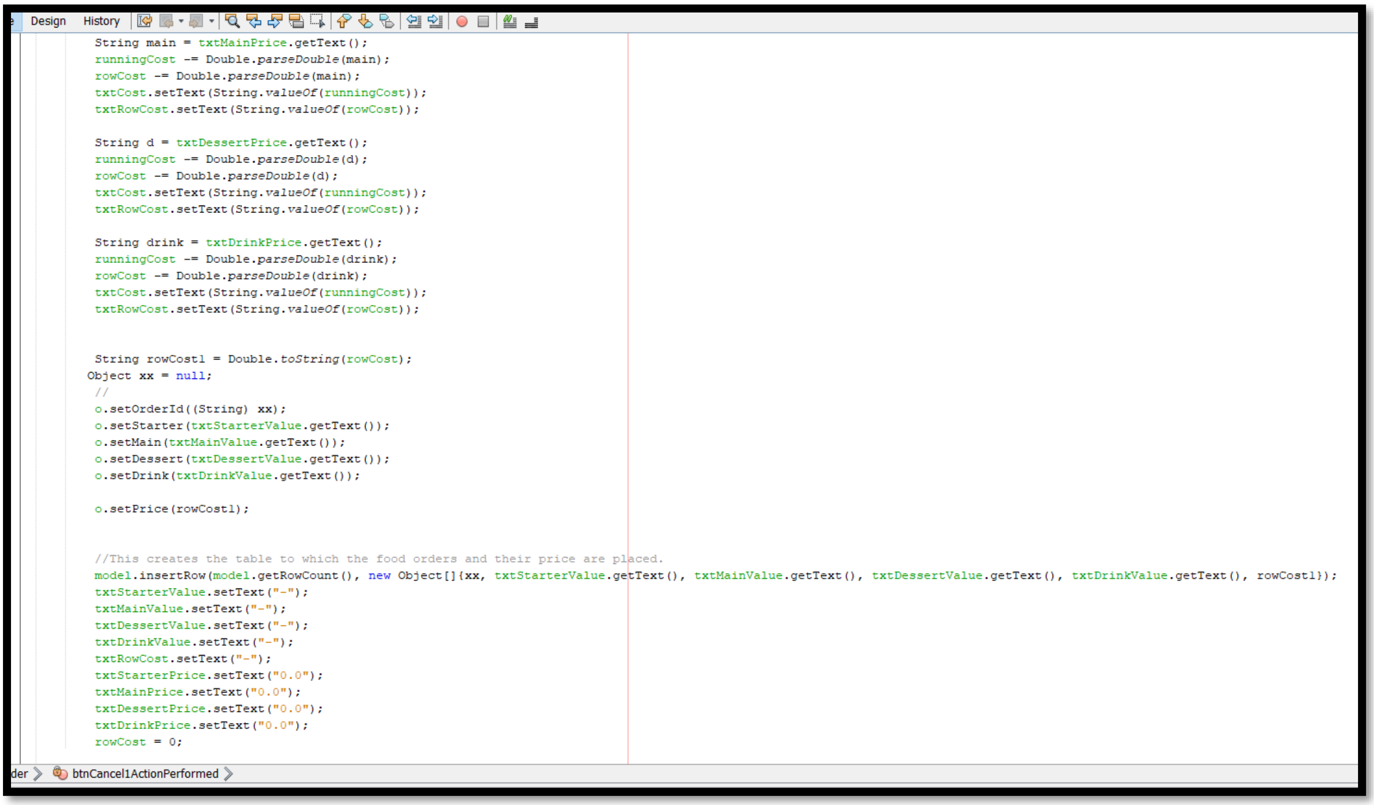
Çorba	20
Ton Balıklı Salata	30
CheeseCake	30
Türk Kahvesi	12

92.0

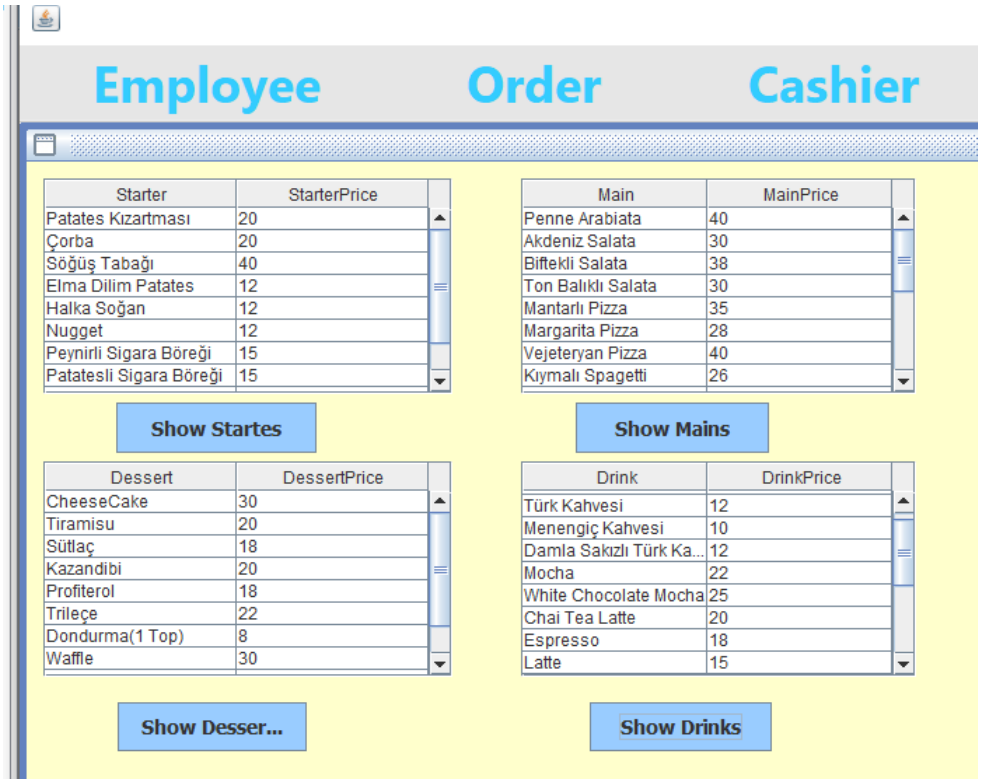
Click the 'Add Order' and the select values are calculated



```java
        String starter = txtStarterPrice.getText();
        runningCost += Double.parseDouble(starter);
        rowCost += Double.parseDouble(starter);
        txtCost.setText(String.valueOf(runningCost));
        txtRowCost.setText(String.valueOf(rowCost));

        String main = txtMainPrice.getText();
        runningCost += Double.parseDouble(main);
        rowCost += Double.parseDouble(main);
        txtCost.setText(String.valueOf(runningCost));
        txtRowCost.setText(String.valueOf(rowCost));

        String d = txtDessertPrice.getText();
        runningCost += Double.parseDouble(d);
        rowCost += Double.parseDouble(d);
        txtCost.setText(String.valueOf(runningCost));
        txtRowCost.setText(String.valueOf(rowCost));

        String drink = txtDrinkPrice.getText();
        runningCost += Double.parseDouble(drink);
        rowCost += Double.parseDouble(drink);
        txtCost.setText(String.valueOf(runningCost));
        txtRowCost.setText(String.valueOf(rowCost));


        String rowCost1 = Double.toString(rowCost);
       Object xx = null;
        //
        o.setOrderId((String) xx);
        o.setStarter(txtStarterValue.getText());
        o.setMain(txtMainValue.getText());
        o.setDessert(txtDessertValue.getText());
        o.setDrink(txtDrinkValue.getText());

        o.setPrice(rowCost1);


        //This creates the table to which the food orders and their price are placed.
        model.insertRow(model.getRowCount(), new Object[]{xx, txtStarterValue.getText(), txtMainValue.getText(), txtDessertValue.getText(), txtDrinkValue.getText(), rowCost1});
        txtStarterValue.setText("-");
        txtMainValue.setText("-");
        txtDessertValue.setText("-");
        txtDrinkValue.setText("-");
        txtRowCost.setText("-");
```



```java
        String main = txtMainPrice.getText();
        runningCost -= Double.parseDouble(main);
        rowCost -= Double.parseDouble(main);
        txtCost.setText(String.valueOf(runningCost));
        txtRowCost.setText(String.valueOf(rowCost));

        String d = txtDessertPrice.getText();
        runningCost -= Double.parseDouble(d);
        rowCost -= Double.parseDouble(d);
        txtCost.setText(String.valueOf(runningCost));
        txtRowCost.setText(String.valueOf(rowCost));

        String drink = txtDrinkPrice.getText();
        runningCost -= Double.parseDouble(drink);
        rowCost -= Double.parseDouble(drink);
        txtCost.setText(String.valueOf(runningCost));
        txtRowCost.setText(String.valueOf(rowCost));


        String rowCost1 = Double.toString(rowCost);
       Object xx = null;
        //
        o.setOrderId((String) xx);
        o.setStarter(txtStarterValue.getText());
        o.setMain(txtMainValue.getText());
        o.setDessert(txtDessertValue.getText());
        o.setDrink(txtDrinkValue.getText());

        o.setPrice(rowCost1);


        //This creates the table to which the food orders and their price are placed.
        model.insertRow(model.getRowCount(), new Object[]{xx, txtStarterValue.getText(), txtMainValue.getText(), txtDessertValue.getText(), txtDrinkValue.getText(), rowCost1});
        txtStarterValue.setText("-");
        txtMainValue.setText("-");
        txtDessertValue.setText("-");
        txtDrinkValue.setText("-");
        txtRowCost.setText("-");
        txtStarterPrice.setText("0.0");
        txtMainPrice.setText("0.0");
        txtDessertPrice.setText("0.0");
        txtDrinkPrice.setText("0.0");
        rowCost = 0;
```

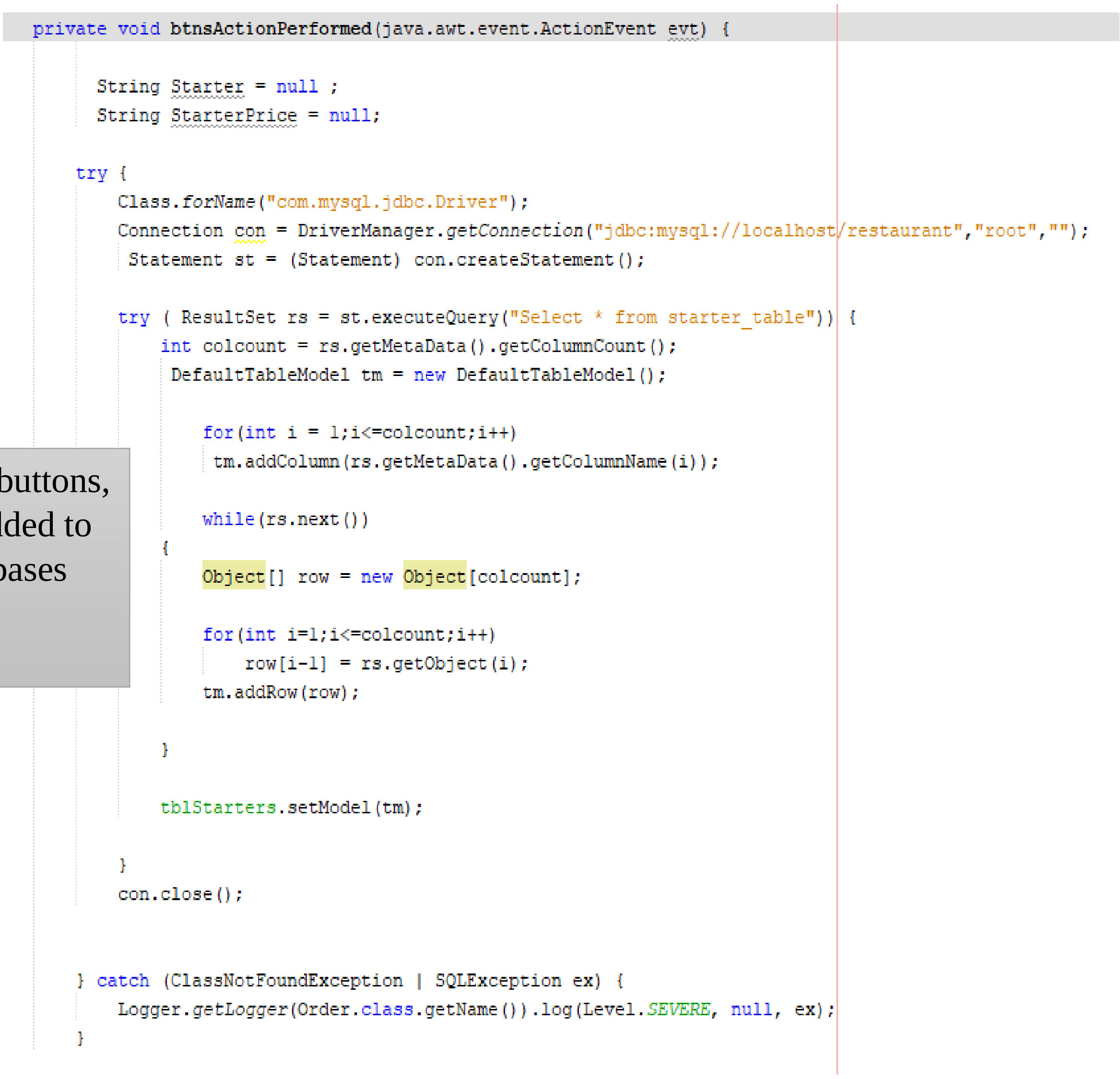


Starter	StarterPrice
Patates Kızartması	20
Çorba	20
Söğüş Tabağı	40
Elma Dilim Patates	12
Halka Soğan	12
Nugget	12
Peynirli Sigara Böreği	15
Patatesli Sigara Böreği	15

Main	MainPrice
Penne Arabiata	40
Akdeniz Salata	30
Biftekli Salata	38
Ton Balıklı Salata	30
Mantarlı Pizza	35
Margarita Pizza	28
Vejeteryan Pizza	40
Kıymalı Spagetti	26

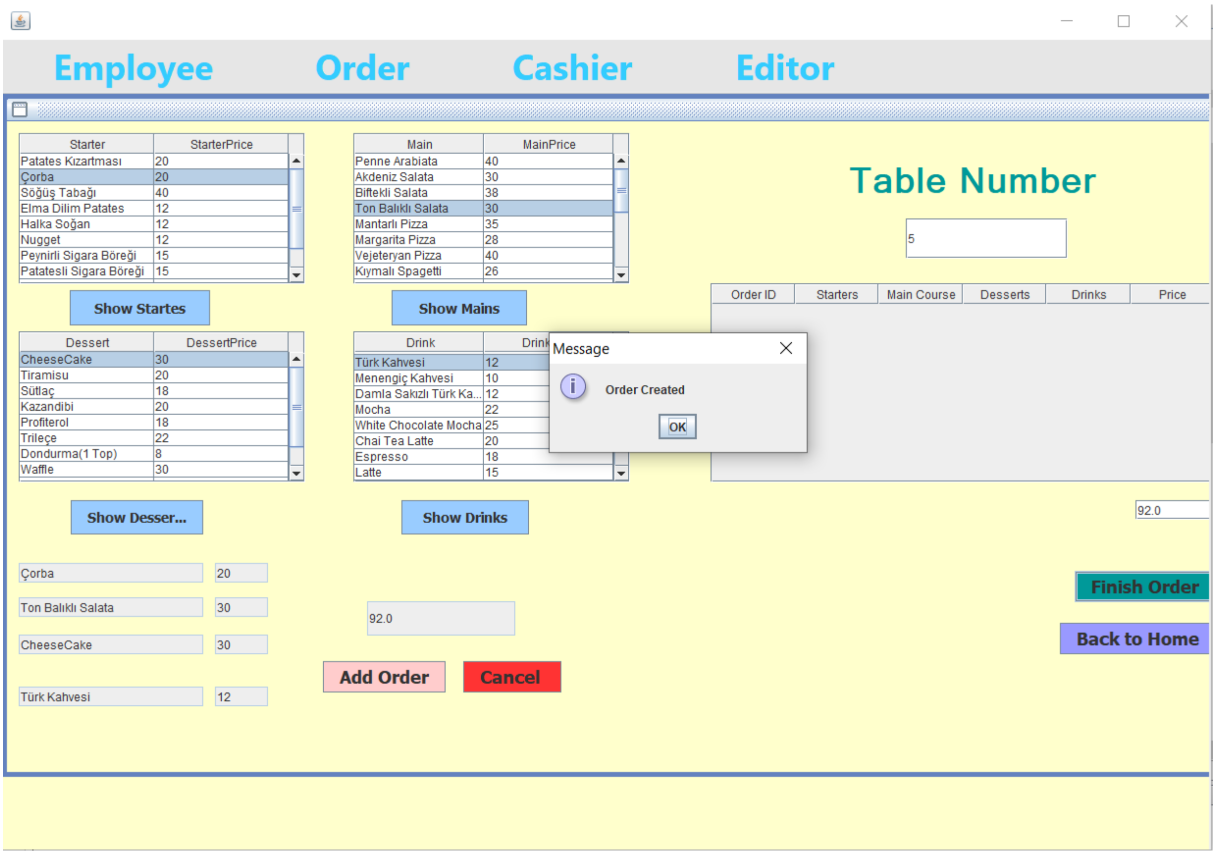
Dessert	DessertPrice
CheeseCake	30
Tiramisu	20
Sütlaç	18
Kazandibi	20
Profiterol	18
Trileçe	22
Dondurma(1 Top)	8
Waffle	30

Drink	DrinkPrice
Türk Kahvesi	12
Menengiç Kahvesi	10
Damla Sakızlı Türk Ka...	12
Mocha	22
White Chocolate Mocha	25
Chai Tea Latte	20
Espresso	18
Latte	15



```
private void btnsActionPerformed(java.awt.event.ActionEvent evt) {

    String Starter = null ;
    String StarterPrice = null;

  try {
      Class.forName("com.mysql.jdbc.Driver");
      Connection con = DriverManager.getConnection("jdbc:mysql://localhost/restaurant","root","");
       Statement st = (Statement) con.createStatement();

      try ( ResultSet rs = st.executeQuery("Select * from starter_table")) {
          int colcount = rs.getMetaData().getColumnCount();
           DefaultTableModel tm = new DefaultTableModel();

              for(int i = 1;i<=colcount;i++)
               tm.addColumn(rs.getMetaData().getColumnName(i));

              while(rs.next())
          {
              Object[] row = new Object[colcount];

              for(int i=1;i<=colcount;i++)
                  row[i-1] = rs.getObject(i);
              tm.addRow(row);

          }

          tblStarters.setModel(tm);

      }
      con.close();


  } catch (ClassNotFoundException | SQLException ex) {
      Logger.getLogger(Order.class.getName()).log(Level.SEVERE, null, ex);
  }
```

When clicked on Show buttons, it takes the data that I added to our menu from the databases where they are located.



Starter	StarterPrice
Patates Kızartması	20
Çorba	20
Söğüş Tabağı	40
Elma Dilim Patates	12
Halka Soğan	12
Nugget	12
Peynirli Sigara Böreği	15
Patatesli Sigara Böreği	15

Main	MainPrice
Penne Arabiata	40
Akdeniz Salata	30
Biftekli Salata	38
Ton Balıklı Salata	30
Mantarlı Pizza	35
Margarita Pizza	28
Vejeteryan Pizza	40
Kıymalı Spagetti	26

Dessert	DessertPrice
CheeseCake	30
Tiramisu	20
Sütlaç	18
Kazandibi	20
Profiterol	18
Trileçe	22
Dondurma(1 Top)	8
Waffle	30

Drink	Drink
Türk Kahvesi	12
Menengiç Kahvesi	10
Damla Sakızlı Türk Ka...	12
Mocha	22
White Chocolate Mocha	25
Chai Tea Latte	20
Espresso	18
Latte	15

Message

Order Created

Table Number

5

Order ID	Starters	Main Course	Desserts	Drinks	Price

92.0

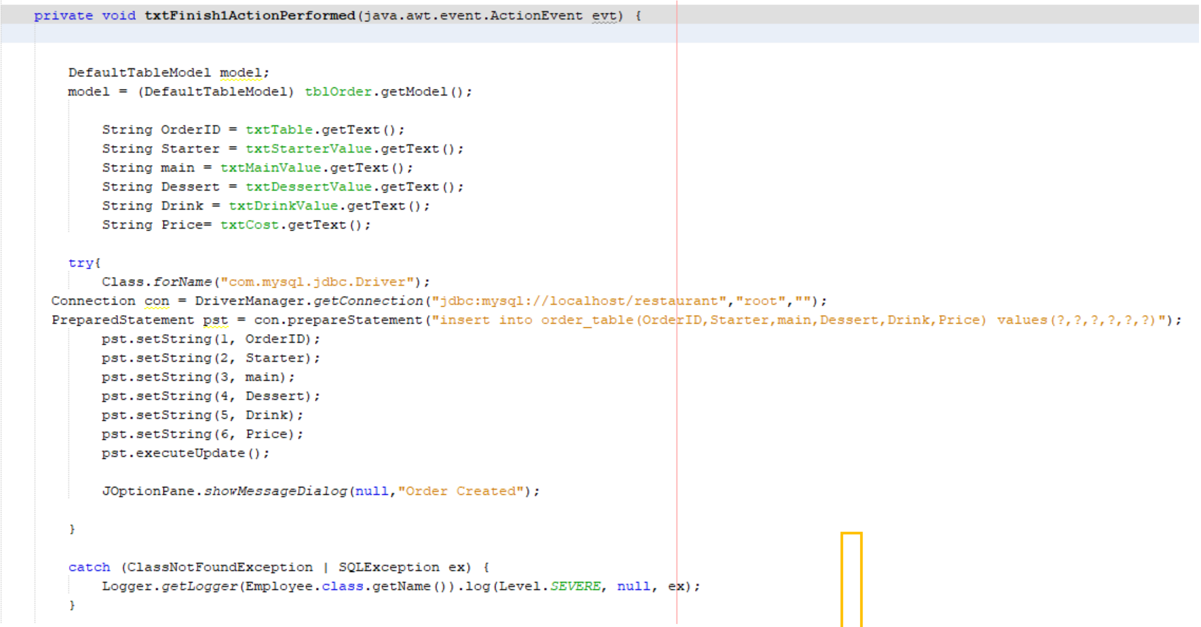
Çorba 20

Ton Balıklı Salata 30

CheeseCake 30

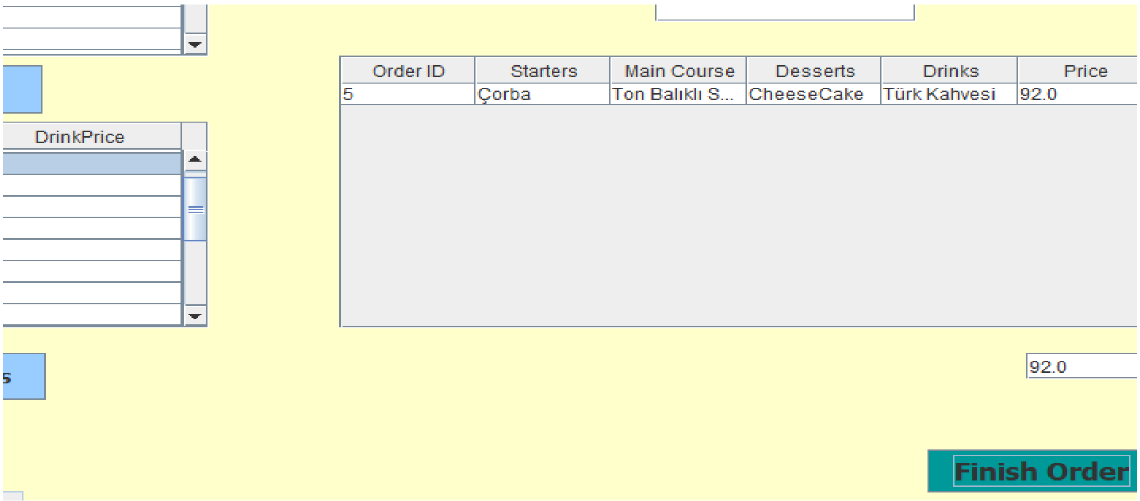
Türk Kahvesi 12

92.0

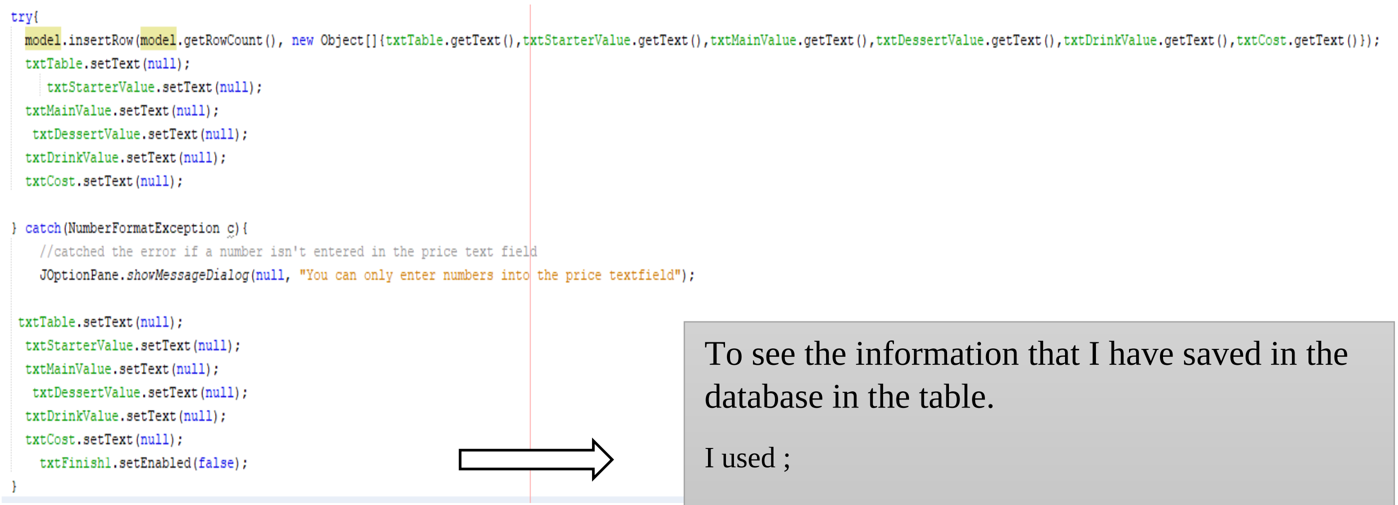


```
private void txtFinish1ActionPerformed(java.awt.event.ActionEvent evt) {

    DefaultTableModel model;
    model = (DefaultTableModel) tblOrder.getModel();

        String OrderID = txtTable.getText();
        String Starter = txtStarterValue.getText();
        String main = txtMainValue.getText();
        String Dessert = txtDessertValue.getText();
        String Drink = txtDrinkValue.getText();
        String Price= txtCost.getText();

    try{
        Class.forName("com.mysql.jdbc.Driver");
  Connection con = DriverManager.getConnection("jdbc:mysql://localhost/restaurant","root","");
  PreparedStatement pst = con.prepareStatement("insert into order_table(OrderID,Starter,main,Dessert,Drink,Price) values(?,?,?,?,?,?)");
        pst.setString(1, OrderID);
        pst.setString(2, Starter);
        pst.setString(3, main);
        pst.setString(4, Dessert);
        pst.setString(5, Drink);
        pst.setString(6, Price);
        pst.executeUpdate();

        JOptionPane.showMessageDialog(null,"Order Created");

    }

    catch (ClassNotFoundException | SQLException ex) {
        Logger.getLogger(Employee.class.getName()).log(Level.SEVERE, null, ex);
    }
```

When the 'Finish Order' button is clicked, 'Order Created' message appears on the panel and the created order is saved in the database. Then click OK and the order appears in the table.



Order ID	Starters	Main Course	Desserts	Drinks	Price
5	Çorba	Ton Balıklı S...	CheeseCake	Türk Kahvesi	92.0



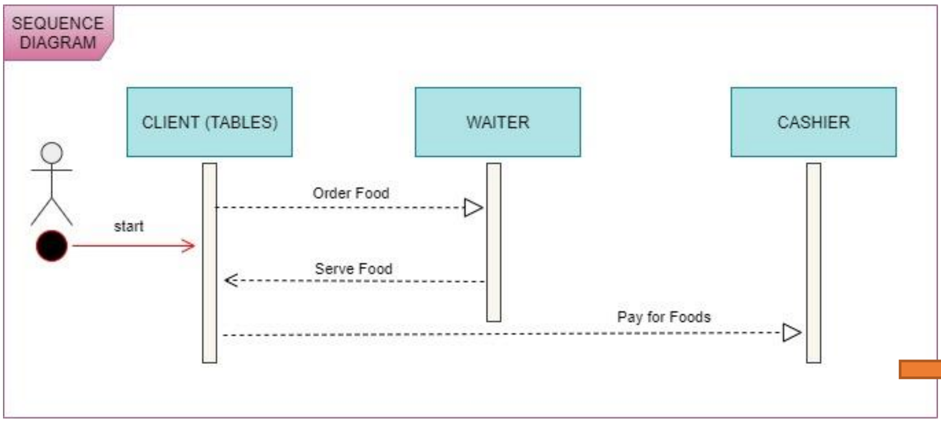
```
try{
  model.insertRow(model.getRowCount(), new Object[]{txtTable.getText(),txtStarterValue.getText(),txtMainValue.getText(),txtDessertValue.getText(),txtDrinkValue.getText(),txtCost.getText()});
  txtTable.setText(null);
     txtStarterValue.setText(null);
  txtMainValue.setText(null);
   txtDessertValue.setText(null);
  txtDrinkValue.setText(null);
  txtCost.setText(null);

} catch(NumberFormatException c){
    //catched the error if a number isn't entered in the price text field
    JOptionPane.showMessageDialog(null, "You can only enter numbers into the price textfield");

 txtTable.setText(null);
  txtStarterValue.setText(null);
  txtMainValue.setText(null);
   txtDessertValue.setText(null);
  txtDrinkValue.setText(null);
  txtCost.setText(null);
    txtFinish1.setEnabled(false);
}
```

To see the information that I have saved in the database in the table.

I used ;

getText() ➔ Gets the information entered.

setText() ➔Paste the entered information into the table.

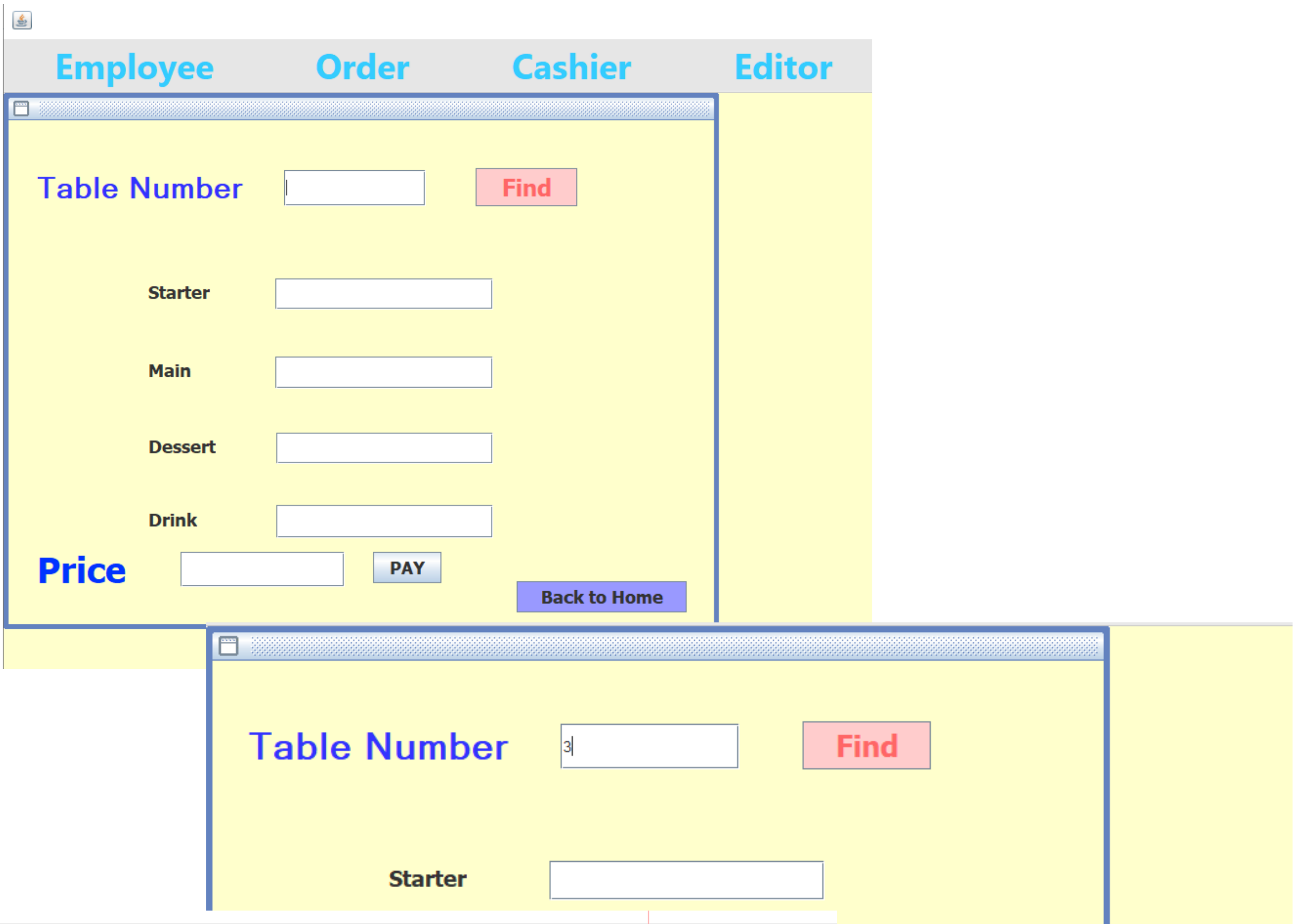


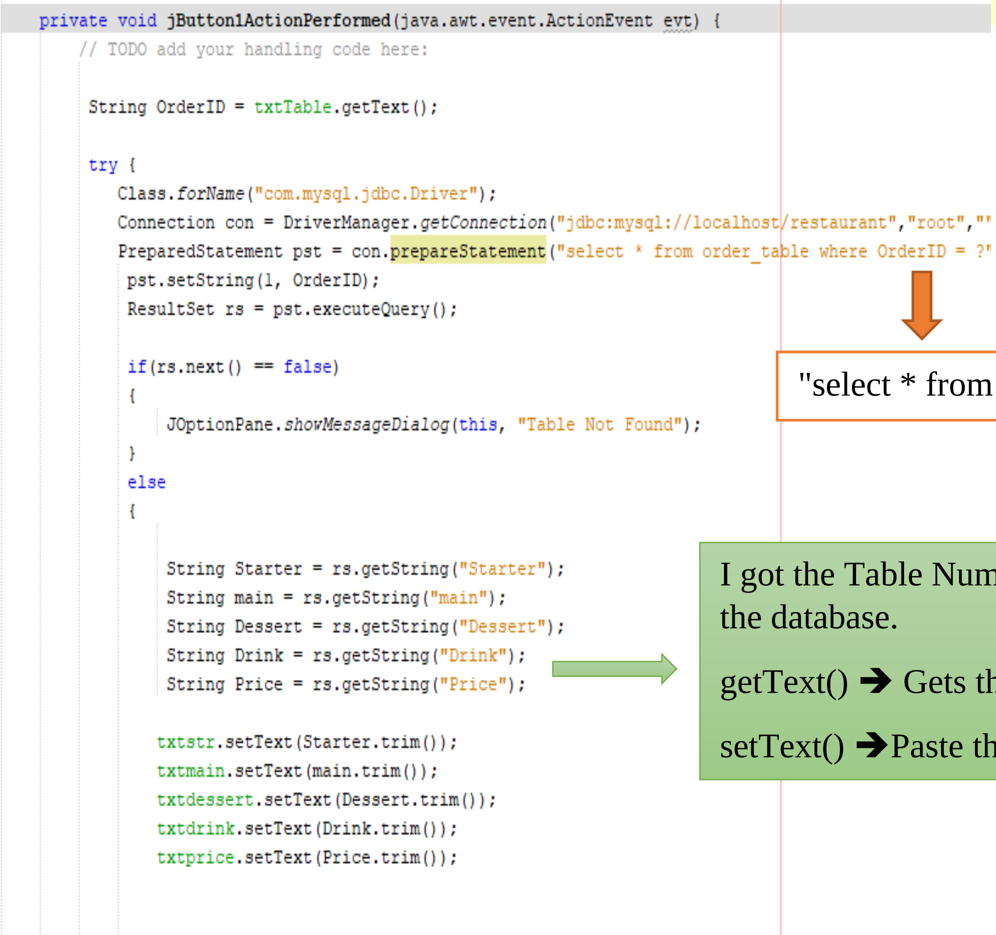
The last part;

➔Cashier

➔Payment

FRAME ‘PAYMENT’





```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

     String OrderID = txtTable.getText();

     try {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost/restaurant","root",""
        PreparedStatement pst = con.prepareStatement("select * from order_table where OrderID = ?"
         pst.setString(1, OrderID);
         ResultSet rs = pst.executeQuery();

         if(rs.next() == false)
         {
             JOptionPane.showMessageDialog(this, "Table Not Found");
         }
         else
         {

             String Starter = rs.getString("Starter");
             String main = rs.getString("main");
             String Dessert = rs.getString("Dessert");
             String Drink = rs.getString("Drink");
             String Price = rs.getString("Price");

           txtstr.setText(Starter.trim());
           txtmain.setText(main.trim());
           txtdessert.setText(Dessert.trim());
           txtdrink.setText(Drink.trim());
           txtprice.setText(Price.trim());
```

"select * from order_table where OrderID = ?"

I got the Table Number I wrote and got the data from the database.

getText() ➔ Gets the information entered.

setText() ➔Paste the entered information into the table.

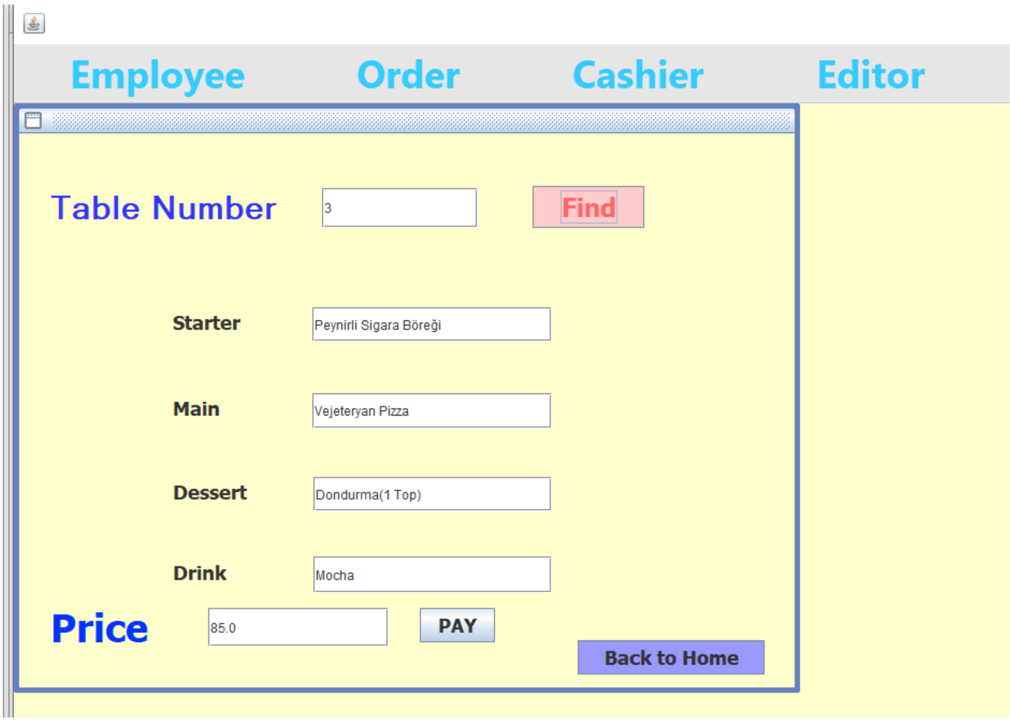


Table Number 3

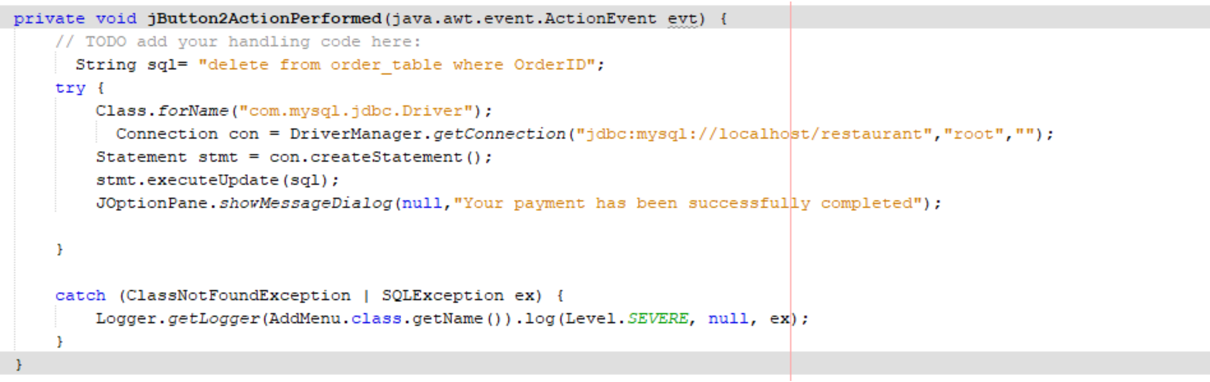
Starter Peynirli Sigara Böreği

Main Vejeteryan Pizza

Dessert Dondurma(1 Top)

Drink Mocha

Price 85.0



```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
     String sql= "delete from order_table where OrderID";
    try {
        Class.forName("com.mysql.jdbc.Driver");
          Connection con = DriverManager.getConnection("jdbc:mysql://localhost/restaurant","root","");
        Statement stmt = con.createStatement();
        stmt.executeUpdate(sql);
        JOptionPane.showMessageDialog(null,"Your payment has been successfully completed");

    }

    catch (ClassNotFoundException | SQLException ex) {
        Logger.getLogger(AddMenu.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

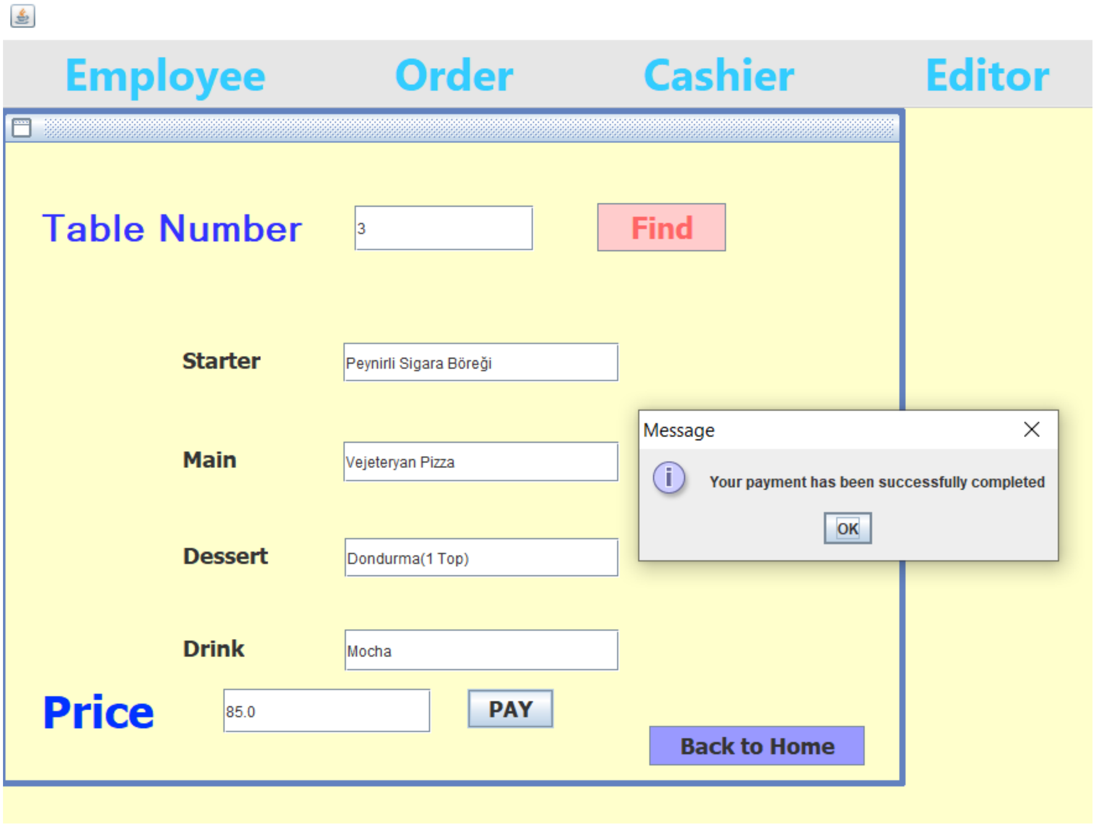


Table Number 3

Starter Peynirli Sigara Böreği

Main Vejeteryan Pizza

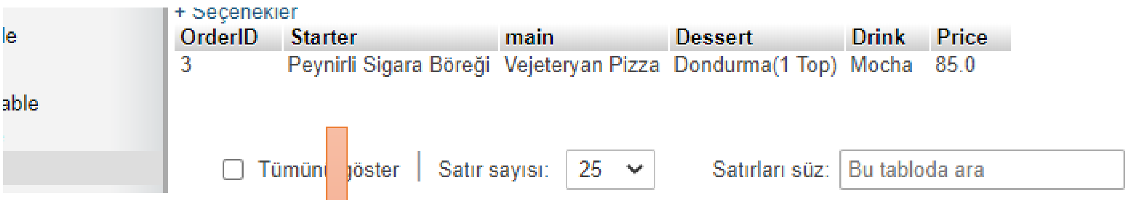
Dessert Dondurma(1 Top)

Drink Mocha

Price 85.0

Message

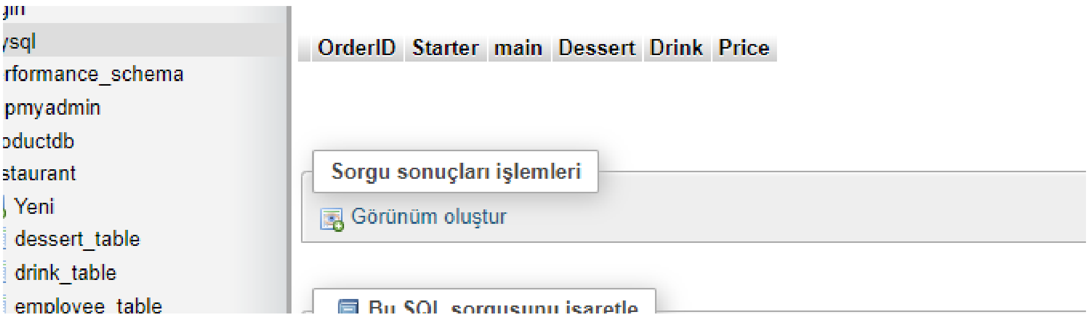
Your payment has been successfully completed



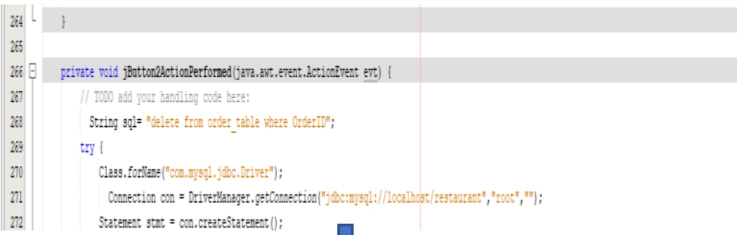
OrderID	Starter	main	Dessert	Drink	Price
3	Peynirli Sigara Böreği	Vejeteryan Pizza	Dondurma(1 Top)	Mocha	85.0

☐ Tümünü göster | Satır sayısı: 25 ⌄ Satırları süz: Bu tabloda ara

AFTER THE PAYMENT



OrderID	Starter	main	Dessert	Drink	Price



```
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
         String sql= "delete from order_table where OrderID";
        try {
          Class.forName("com.mysql.jdbc.Driver");
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost/restaurant","root","");
          Statement stmt = con.createStatement();
```

I wrote ;

"delete from order_table where OrderID"

It reaches the data in the order where the Table Number is located and shows that the table is empty after the payment.