

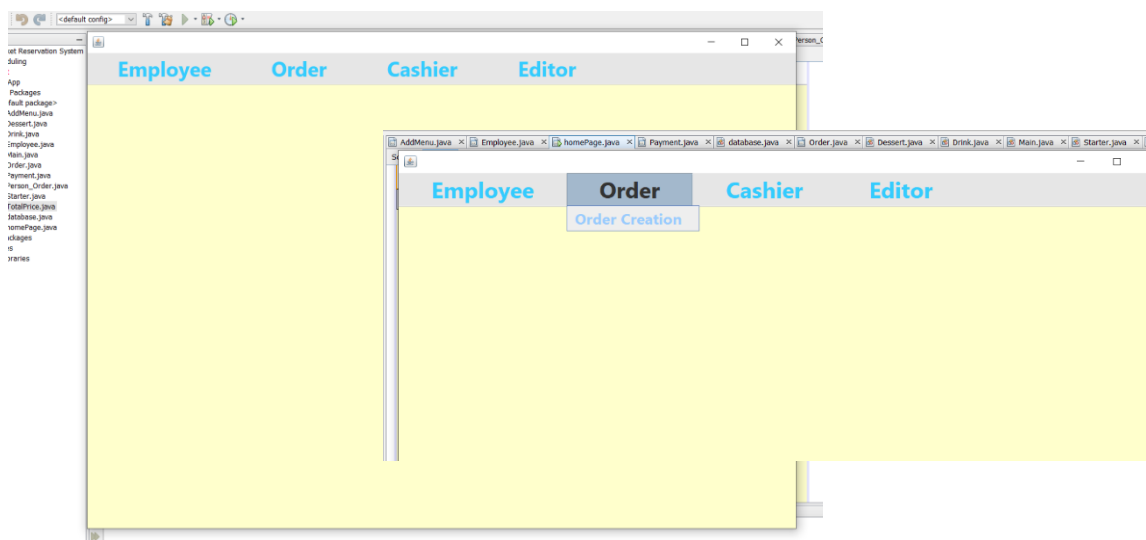
# SOFTWARE ENGINEERING TOOLS REPORT

## RESTAURANT APPLICATION

*Arife Gül Yalçın*

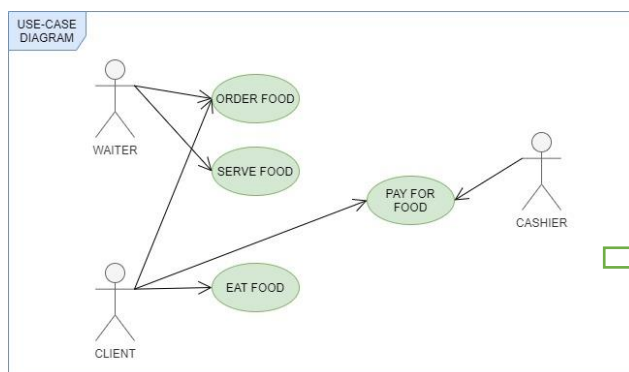
*B1605.090054*

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();
    panell.add(emp);
    emp.setVisible(true);
}

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

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

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

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

The screenshot shows a Java Swing application window titled "Employee Information". The window has a yellow background and a blue border. At the top, there is a header bar with four tabs: "Employee", "Order", "Cashier", and "Editor". The "Employee" tab is currently selected. Below the header, the form contains several input fields and a table.

**Employee Information**

First Name  T.C

Last Name  Wage

Position

EmployeeID	First Name	Last Name	T.C	Weekly Wage	Position

At the bottom of the form, there are two buttons: "Back to Home" and "Add".

## 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
------------	------------	-----------	-----	-------------

Back to Home Add

Message: Registration Created OK

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

```
private void btnAddActionPerformed(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,"Registration 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.

Employee

Order

Cashier

First Name

T.C

Last Name

Wage

Position

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

```

253 try{
254     model.insertRow(model.getRowCount(), new Object[] { txtEmployeeID.getText(),txtFirstName.getText(), txtLastName.getText(), txtPPSN.getText(), txtWage.getText(), txtPosition.getText()});
255     txtEmployeeID.setText(null);
256     txtFirstName.setText(null);
257     txtLastName.setText(null);
258     txtPPSN.setText(null);
259     txtWage.setText(null);
260     txtPosition.setText(null);
261 } catch (NumberFormatException g){
262     //caught the error if a number isn't entered in the price text field
263     JOptionPane.showMessageDialog(null, "You must enter all textfield");
264 }
265
266 txtEmployeeID.setText(null);
267 txtFirstName.setText(null);
268 txtLastName.setText(null);
269 txtPPSN.setText(null);
270 txtWage.setText(null);
271 txtPosition.setText(null);
272 btnAddR.setEnabled(false);
273
274

```

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'

	+ Seçenekler					
	EmployeeID	First_Name	Last_Name	TC	Wages	Position
t_table	EMPLOYEE010	Arife	Yalçın	1234	4000	Waiter
table	EmployeeID	Beyza	Madenoglu	39675893457	5000	Cashier
ye_table	EmployeeID	Sacidenur	Yalçın	4849673468943	3600	Waiter
_table	EmployeeID	Ayşe	Yalçın	7457568	4643	Waiter
_table	EmployeeID	Zeynep	Sude	56046t398467	2500	Waiter
table	EmployeeID	Emine	Yalçın	3857324856	8000	Manager
table	EmployeeID	Hatice	Menteşe	3458732	5000	cashier
_table	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'

Employee Order

Cashier

Editor

Starters

Starter

Starter Price

Add Starter

Delete Starter

Deselect

Main Courses

Main Course

Main Course Price

Add Main

Delete Order

Deselect

Desserts

Dessert

Dessert Price

Add Dessert

Delete Dessert

Deselect

Drinks

Drink

Drink Price

Add Drink

Delete Drink

Deselect

Starters

Price

Main Courses

Price

Desserts

Price

Drinks

Price

Back to Home

The screenshot displays the 'Employee Order Cashier Editor' application. The interface is divided into several sections for adding items to an order:

- Starter Section:** Includes a text input for the starter name, a numeric input for the price, and a 'Deselect' button. Below these is an 'Add Starter' button.
- Main Course Section:** Includes a text input for the main course name, a numeric input for the price (containing '50'), and a 'Deselect' button. Below these are 'Add Main' and 'Delete Order' buttons.
- Dessert Section:** Includes a text input for the dessert name, a numeric input for the price, and a 'Deselect' button. Below these are 'Add Dessert' and 'Delete Dessert' buttons.
- Drink Section:** Includes a text input for the drink name, a numeric input for the price, and a 'Deselect' button. Below these are 'Add Drink' and 'Delete Starter' buttons.

On the right side, there are four tables listing available items and their prices:

- Starters Table:**

Starters	Price
Çorba	20
Soğuş Tabagi	40
Elma Dilim Patat..	12
Halka Soğan	12
Nugget	12
Peynirli Sigara B...	15
Patatesli Sigara...	15
- Main Courses Table:**

Main Courses	Price
Akdeniz Salata	30
Bitfelli Salata	38
Ton Balıklı Salata	30
Manitarlı Pizza	35
Margarita Pizza	28
Vejeteryan Pizza	40
Kıymalı Spagetti	26
- Desserts Table:**

Desserts	Price
719	
680	
681	
682	
683	
- Drinks Table:**

Drinks	Price
684	

At the bottom, there is a status bar with the text 'Employee Order Cashier Editor' and a small inset window showing a partial view of the same interface.

**Employee Order Cashier Editor**

---

Starter

Deselect

Starter Price

Add Starter

Delete Starter

Main Course

Deselect

Main Course Price

Add Main

Delete Order

Dessert

Deselect

Dessert Price

Add Dessert

Delete Dessert

Drink

Smoothies

Deselect

Drink Price

20

Add Drink

Delete Drink

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

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

Desserts	Price
Sutlaç	18
Kazandibi	20
Profiterol	18
Tritleş	22
Condurma(1 Top)	8
Waffle	30
Fondü	35

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

Back to Home

```
private void btnDeselectDesActionPerformed(java.awt.event.ActionEvent evt) {  
  
    txtDessert.setText(null);  
    txtDessertPrice.setText(null);  
  
}
```

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 e){
        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'

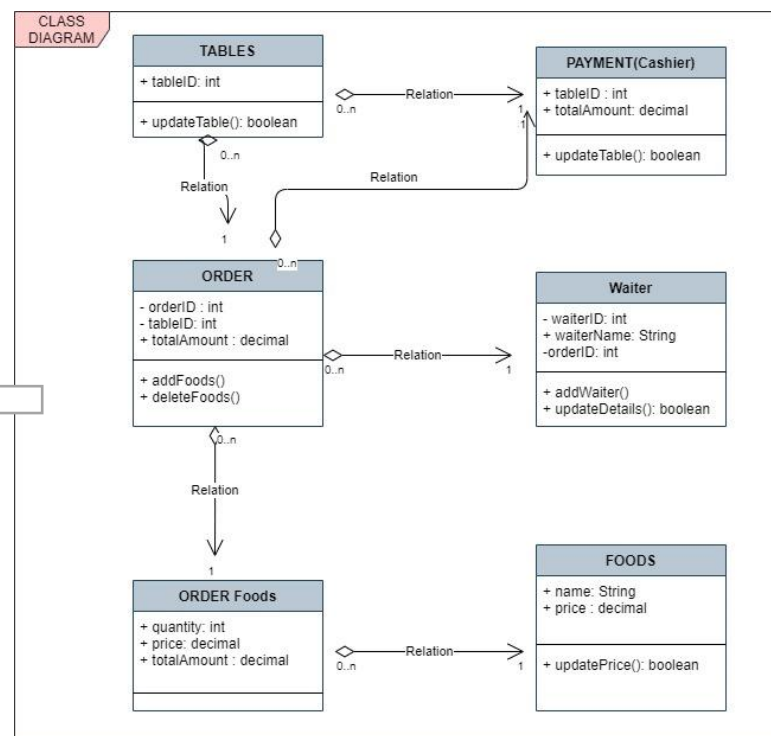
+ Seçenekler	
Dessert	DessertPrice
CheeseCake	30
Tiramisu	20
Sütlac	18
Kazandibi	20
Profiterol	18
Trileçe	22
Dondurma(1 Top)	8
Waffle	30
Fondü	35
Kadayıf	30

+ 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

Yeni	
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
Kremeli 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	
Starter	StarterPrice
Patates Kızartması	20
Çorba	20
Söğüş Tabagi	40
Elma Dilim Patates	12
Halka Soğan	12
Nugget	12
Peynirli Sığara Böreği	15
Patatesli Sığara 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'

Employee Order Cashier Editor

Starter	StarterPrice	Main	MainPrice
Patates Kızartması	20	Penne Arabiata	40
Çorba	20	Akdeniz Salata	30
Soğuş Tabağı	40	Bitkili Salata	38
Elma Dilim Patates	12	Ton Balıklı Salata	30
Halka Soğan	12	Mantarlı Pizza	35
Nugget	12	Margarita Pizza	28
Peynirli Sigara Böreği	15	Vejeteryan Pizza	40
Patatesli Sigara Böreği	15	Kıymalı Spagetti	26

Show Startes Show Mains

Dessert	DessertPrice	Drink	DrinkPrice
CheeseCake	30	Türk Kahvesi	12
Tiramisu	20	Menengiç Kahvesi	10
Sütlüç	18	Damla Sakızlı Türk Ka...	12
Kazandibi	20	Mocha	22
Profiterol	18	White Chocolate Mocha	25
Trileçe	22	Chai Tea Latte	20
Dondurma(1 Top)	8	Espresso	18
Waffle	30	Latte	15

Show Desser... Show Drinks

Table Number

Order ID	Starters	Main Course	Desserts	Drinks	Price
----------	----------	-------------	----------	--------	-------

0.0 0.0 0.0 0.0

Add Order Cancel

Finish Order Back to Home

Employee Order Cashier Editor

Starter	StarterPrice	Main	MainPrice
Patates Kızartması	20	Penne Arabiata	40
Çorba	20	Akdeniz Salata	30
Soğuş Tabağı	40	Bitkili Salata	38
Elma Dilim Patates	12	Ton Balıklı Salata	30
Halka Soğan	12	Mantarlı Pizza	35
Nugget	12	Margarita Pizza	28
Peynirli Sigara Böreği	15	Vejeteryan Pizza	40
Patatesli Sigara Böreği	15	Kıymalı Spagetti	26

Show Startes Show Mains

Dessert	DessertPrice	Drink	DrinkPrice
CheeseCake	30	Türk Kahvesi	12
Tiramisu	20	Menengiç Kahvesi	10
Sütlüç	18	Damla Sakızlı Türk Ka...	12
Kazandibi	20	Mocha	22
Profiterol	18	White Chocolate Mocha	25
Trileçe	22	Chai Tea Latte	20
Dondurma(1 Top)	8	Espresso	18
Waffle	30	Latte	15

Show Desser... Show Drinks

Table Number

Order ID	Starters	Main Course	Desserts	Drinks	Price
----------	----------	-------------	----------	--------	-------

92.0

Çorba 20 Ton Balıklı Salata 30 CheeseCake 30 Türk Kahvesi 12

92.0

Add Order Cancel

Finish Order Back to Home

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



```
Source Design History
588 String starter = txtStarterPrice.getText();
589 runningCost += Double.parseDouble(starter);
590 rowCost += Double.parseDouble(starter);
591 txtCost.setText(String.valueOf(runningCost));
592 txtRowCost.setText(String.valueOf(rowCost));
593
594 String main = txtMainPrice.getText();
595 runningCost += Double.parseDouble(main);
596 rowCost += Double.parseDouble(main);
597 txtCost.setText(String.valueOf(runningCost));
598 txtRowCost.setText(String.valueOf(rowCost));
599
600 String d = txtDessertPrice.getText();
601 runningCost += Double.parseDouble(d);
602 rowCost += Double.parseDouble(d);
603 txtCost.setText(String.valueOf(runningCost));
604 txtRowCost.setText(String.valueOf(rowCost));
605
606 String drink = txtDrinkPrice.getText();
607 runningCost += Double.parseDouble(drink);
608 rowCost += Double.parseDouble(drink);
609 txtCost.setText(String.valueOf(runningCost));
610 txtRowCost.setText(String.valueOf(rowCost));
611
612
613 String rowCost1 = Double.toString(rowCost);
614 Object xx = null;
615 //
616 o.setOrderId((String) xx);
617 o.setStarter(txtStarterValue.getText());
618 o.setMain(txtMainValue.getText());
619 o.setDessert(txtDessertValue.getText());
620 o.setDrink(txtDrinkValue.getText());
621
622 o.setPrice(rowCost1);
623
624
625 //This creates the table to which the food orders and their price are placed.
626 model.insertRow(model.getRowCount(), new Object[]{xx, txtStarterValue.getText(), txtMainValue.getText(), txtDessertValue.getText(), txtDrinkValue.getText(), rowCost1});
627 txtStarterValue.setText("-");
628 txtMainValue.setText("-");
629 txtDessertValue.setText("-");
630 txtDrinkValue.setText("-");
631 txtRowCost.setText("-");
Order btnAddActionPerformed
```

```
Design History
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;
Order btnCancelActionPerformed
```

Employee

Order

Cashier

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

Show Startes

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

Show Desser...

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

Show Mains

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

Show Drinks

```
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.

Employee
Order
Cashier
Editor

Starter	StarterPrice
Patates Kızartması	20
Çorba	20
Soguş Tabagi	40
Elma Dilim Patates	12
Halika Soğan	12
Nugget	12
Peynirli Sigara Boregi	15
Patatesli Sigara Boregi	15

Show Startes

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

Show Desser...

Main	MainPrice
Penne Arabiata	40
Akdeniz Salata	30
Biftekli Salata	38
Ton Balikli Salata	30
Mantarli Pizza	35
Margarita Pizza	28
Vejeteryan Pizza	40
Kıymali Spagetti	26

Show Mains

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

Show Drinks

Table Number

Order ID	Starters	Main Course	Desserts	Drinks	Price
					92.0

20
 30
 30
 12

Add Order
Cancel

Finish Order
Back to Home

Message
Order Created
OK

```

private void txtFinishActionPerformed(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

DrinkPrice

5

92.0

**Finish Order**

```
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 g){
    //caught 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");

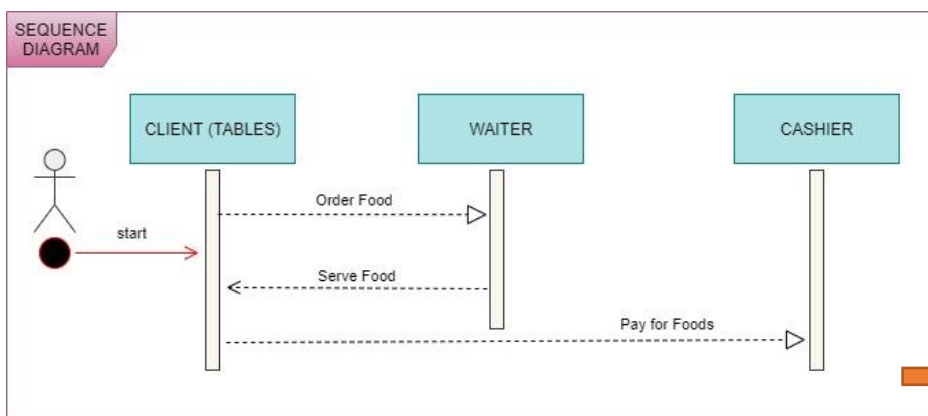
    txtTable.setText(null);
    txtStarterValue.setText(null);
    txtMainValue.setText(null);
    txtDessertValue.setText(null);
    txtDrinkValue.setText(null);
    txtCost.setText(null);
    txtFinish.setEnabled(false);
}
```

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'

The image shows two states of a Java Swing window titled 'PAYMENT'. The window has a title bar with a close button and a menu bar with 'Employee', 'Order', 'Cashier', and 'Editor' menus. The main content area is yellow. In the top state, there are input fields for 'Table Number', 'Starter', 'Main', 'Dessert', 'Drink', and 'Price'. There are also buttons for 'Find', 'PAY', and 'Back to Home'. In the bottom state, the 'Table Number' field contains the value '3', and the 'Starter' field is populated with data from the database. The 'Find' button is highlighted in red.

```
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.

EmployeeOrderCashierEditor

Table Number

3

Find

Starter

Peynirli Sigara Böreği

Main

Vejeteryan Pizza

Dessert

Dondurma(1 Top)

Drink

Mocha

Price

85.0

PAY

Back to Home

```
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);  
    }  
}
```

EmployeeOrderCashierEditor

Table Number

3

Find

Starter

Peynirli Sigara Böreği

Main

Vejeteryan Pizza

Dessert

Dondurma(1 Top)

Drink

Mocha

Price

85.0

PAY

Back to Home

Message

Your payment has been successfully completed

OK

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:

## AFTER THE PAYMENT

performance\_schema

pmyadmin

productdb

restaurant

Yeni

dessert\_table

drink\_table

employee table

OrderID Starter main Dessert Drink Price

Sorgu sonuçları işlemleri

Görünüm oluştur

Bu SQL sorgusunu işaretle

```

264 }
265
266 private void jButtonActionPerformed(java.awt.event.ActionEvent evt) {
267     // TODO add your handling code here:
268     String sql= "delete from order_table where OrderID";
269     try {
270         Class.forName("com.mysql.jdbc.Driver");
271         Connection con = DriverManager.getConnection("jdbc:mysql://localhost/restaurant","root","");
272         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.