

## CS 321 PROJECT MILESTONE (3): Complete Submission

Project Group Number (in Blackboard): **Group 2** Section Number: **6F9**

Project Title: **Oregano's**



|   | Name                      | ID          |
|---|---------------------------|-------------|
| 1 | (Leader) Aseel Alqahatani | 2200003677  |
| 2 | Ghaida Alshammary         | 2200005622  |
| 3 | Fay Alsadlan              | 2200005122  |
| 4 | Thuraya Alzubaidi         | 22000006254 |
| 5 | Sara Alzahrani            | 2200005923  |
| 6 | Dana Alsharari            | 2200004812  |
| 7 | Layan Faisal Alanazi      | 2200003089  |

Instructor: **Ms. Abrar Alotaibi**

## Milestone 1:

## **Proposal of Application is: Italian Restaurant**

**Description:** Our Restaurant system will include an Italian cuisine menu. The restaurant application will provide various services. It will allow the user to browse and select from the menu items from various categories and make new orders. The managers will have the authority to manage the staff. Also, the employees to receive and control the orders.

**Logical analysis & Business:** The restaurant system has two users:

- 1- Managers: will be able to add new employee, search for specific employee, update existing staff records and delete employees.
- 2- Employee: will be able to add, delete and update orders.

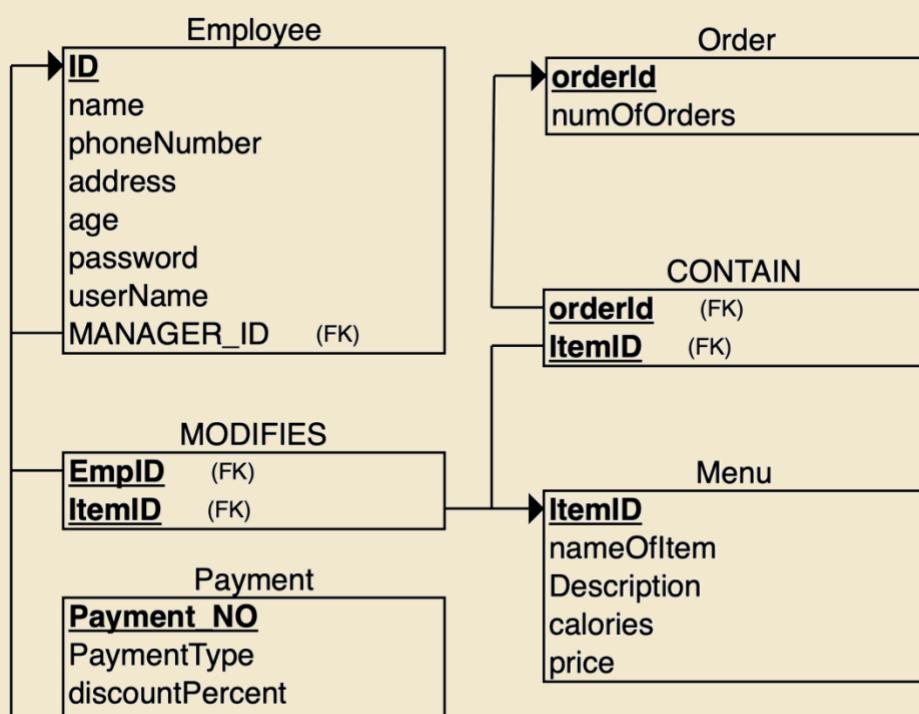
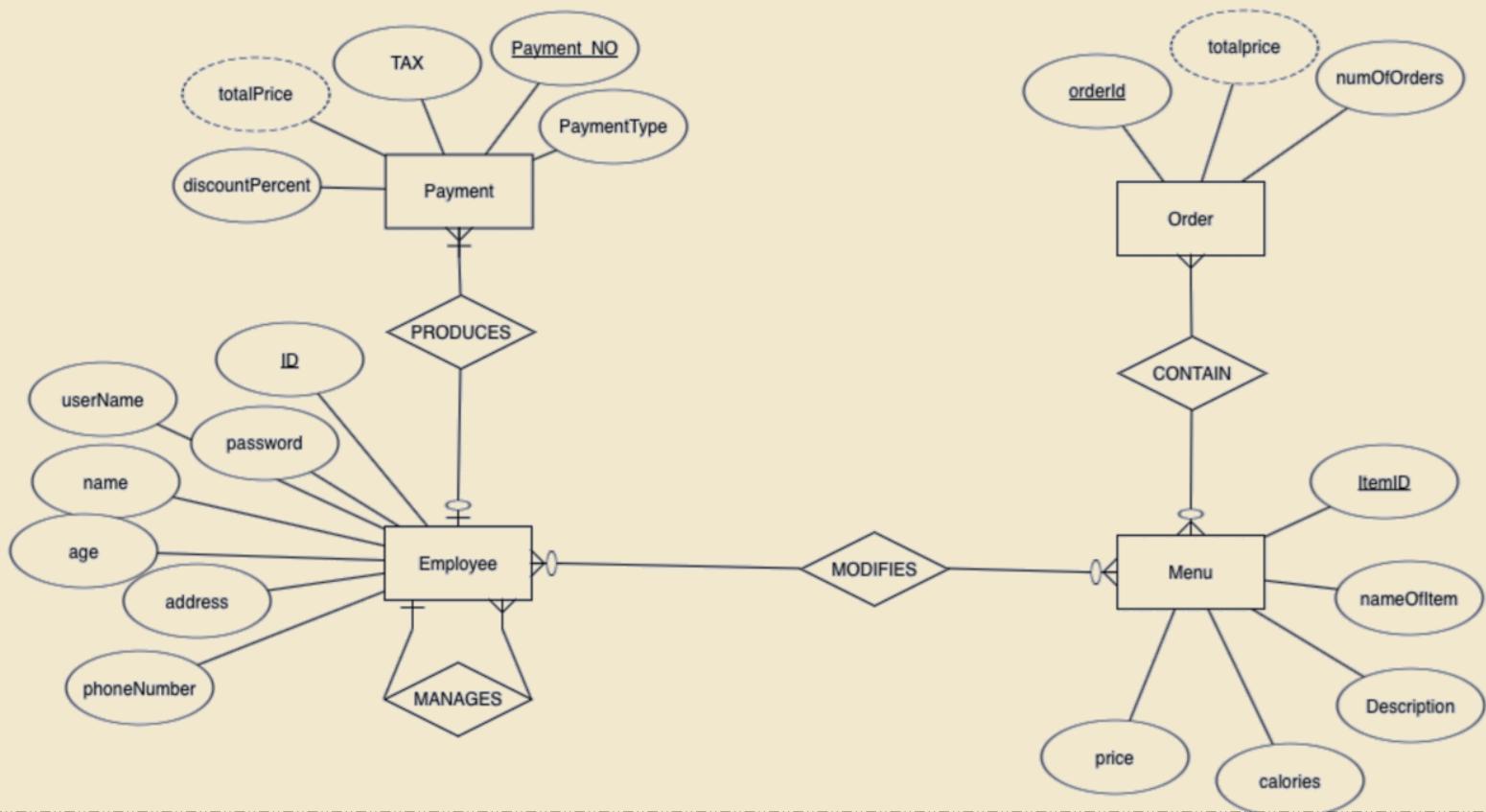
| Function / Application Aspect  | Classes Management | Data Export/Import | Professional Appearance | Error Handling | Data Access Tools and Techniques |
|--|--------------------|--------------------|-------------------------|----------------|----------------------------------|
| User's Log-in<br>(Different types of users: managers, employees, & customer) | ✓                  |                    | ✓                       | ✓              | ✓                                |
| Employees make orders  |                    |                    |                         | ✓              |                                  |
| Managers manage employees (hair/fair)  |                    | ✓                  |                         |                |                                  |
| Employee record/view orders  | ✓                  |                    |                         |                | ✓                                |
| Employee can update the menu (add/delete)                                    | ✓                  |                    |                         |                |                                  |

## Milestone 2:



# Database Design:

ER diagram & Schema



mapped tables filled with data

### -Employees table

| ID        | name    | phoneNumber | address   | age | password | userName | MANAGER_ID |
|-----------|---------|-------------|-----------|-----|----------|----------|------------|
| ► 1001221 | Lamya   | 0554456789  | Al Nuzha  | 36  | 5678     | lamaya2  | NULL       |
| 1004433   | Aseel   | 0552233880  | Al Rakah  | 30  | 1234     | aseel2   | NULL       |
| 1502321   | Fay     | 0561304668  | Al Shati  | 22  | 1234     | fay1     | 1001221    |
| 1503821   | Dana    | 0552231828  | Al Rakah  | 24  | 9101     | dana3    | 1004433    |
| 1503822   | Thuraya | 0552231828  | Al rawada | 24  | 9101     | thuraya  | 1001221    |
| 1504542   | Sara    | 0543322120  | Al Wahan  | 23  | 5678     | sara2    | 1001221    |

### -menu table

| ITEMID | nameOfItem           | Description                                | calories | price |
|--------|----------------------|--|----------|-------|
| ► 1    | Pepsi                | Pepsi                                      | 150      | 10    |
| 2      | Water                | Sparkling water                            | 0        | 18    |
| 3      | Sprite               | Sprite                                     | 145      | 25    |
| 4      | Passion Fruit Mojito | Mojito with passion fruit syrup            | 163      | 35    |
| 5      | Orange Juice         | Freshly Squeezed                           | 87       | 35    |
| 6      | Spaghetti            | Spaghetti with pomodoro sauce              | 897      | 67    |
| 7      | Burrata Salad        | with basil and fresh cherry tomato         | 477      | 69    |
| 9      | Risotto              | Risotto with creamy cheese and mushroom    | 989      | 79    |
| 10     | Alfredo Pasta        | Penne pasta with our special alfredo sauce | 1200     | 85    |
| 11     | Pizza                | Grilled chicken Pizza                      | 900      | 73    |
| 12     | Mixed sea food pa... | Mixed sea food pasta                       | 1000     | 97    |

### -payment table

| Payment_NO | PaymentType | discountPercent | TAX | EmpID   |
|------------|-------------|-----------------|-----|---------|
| ► 1        | cash        | 0               | 15  | 1502321 |
| 2          | credit      | 15              | 15  | 1503821 |
| 3          | cash        | 20              | 15  | 1503822 |

### -Orders table

| orderId | numOfOrders |
|---------|-------------|
| ► 1     | 2           |
| 2       | 3           |
| 3       | 1           |
| 4       | 2           |
| 5       | 3           |
| 6       | 3           |
| 7       | 2           |
| 8       | 2           |
| 9       | 3           |

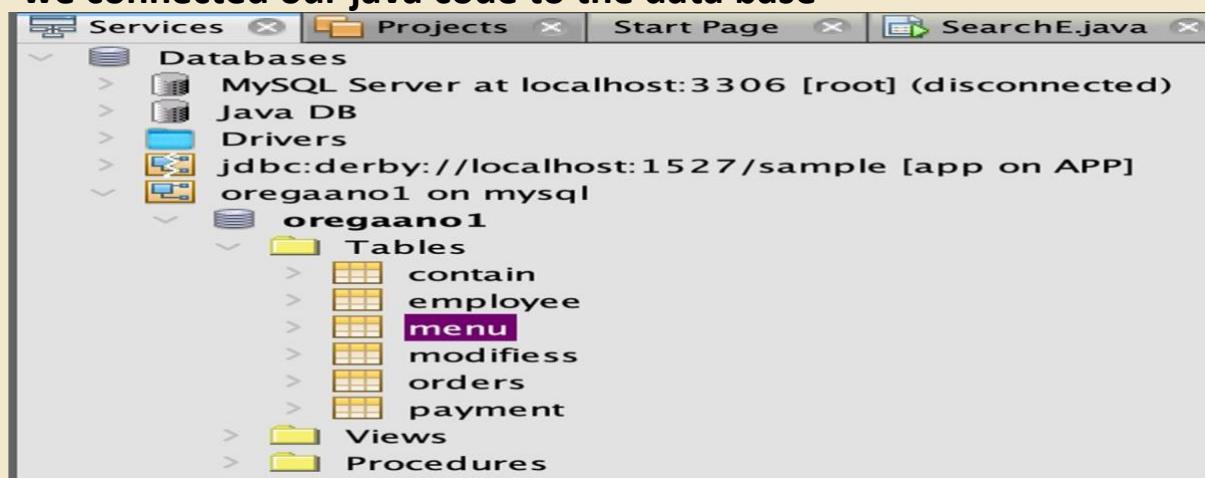
### - Modifies table

| EmpID   | ItemID |
|---------|--------|
| 1503821 | 2      |
| 1504542 | 3      |
| 1504542 | 10     |
| NULL    | NULL   |

### -contain table

| orderId | ItemID |
|---------|--------|
| ► 1     | 1      |
| 8       | 1      |
| 9       | 1      |
| 7       | 2      |
| 8       | 2      |
| 2       | 3      |
| 1       | 4      |
| 2       | 6      |
| 7       | 7      |
| 9       | 9      |
| 9       | 11     |

### -we connected our java code to the data base





Tables with full specification of data type for all fields, constraints and keys:

### **Employee:**

provides information about the employees in the restaurant who can produce payments, modify menu or manage other employees.

- ID int **PK**
  - name varchar
  - phoneNumber varchar
  - address varchar
  - age int
  - password varchar
  - username varchar
  - MANAGER\_ID int (Optional)
- **constraint managerID foreign key (MANAGER\_ID) references Employee(ID).**

### **CONTAIN:**

provides information about the orderId and what menu items it contains.

- orderId int
  - itemID int
- **primary key (orderId,itemID)**
- **constraint order foreign key (orderId) references Order(orderId).**
- **constraint item foreign key (itemID) references Menu(itemID).**

### **Payment:**

provides information about the payments that are produced by employees. totalPrice is a derived attribute that is calculated using discountPercent.

- Payment\_NO int **PK**
  - PaymentType varchar
  - discountPercent int
  - TAX int
  - EmpID int
- **constraint paymentEmpID foreign key (EmpID) references Employee(ID).**

### **Menu:**

provides information about menu items (dishes & beverages).

- itemID int **PK**
- nameOfItem varchar
- description varchar
- calories double
- price double

### **Modifies:**

provides information containing the ID of the employees who modified menu items and the itemID of these menu items.

- EmpID int
- itemID int
- primary key (EmpID, itemID)
- constraint modify foreign key (EmpID) references Employee(ID).
- constraint modifyItem foreign key (itemID) references Menu(itemID).

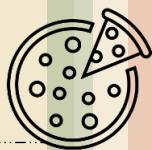
### **Order:**

provides information about the order and how many items it contains.

- orderId int PK
- numOfOrders int

### **Relationship between entities:**

- **CONTAIN:** between MENU and ORDER:  
Each order can contain many menu items, and each menu items can be added in many orders.  
(N:M)
- **MODIFIES:** between EMPLOYEE and MENU:  
Each employee can modify different menu items, and each menu item can be modified by different employees.
- **PRODUCES:** between EMPLOYEE and PAYMENT:  
Each employee can produce many payments, but each payment can be produced by only one employee.
- **MANAGES:** recursive relationship between EMPLOYEES:  
Each employee can manage many employees, but each employee is managed by only one employee.



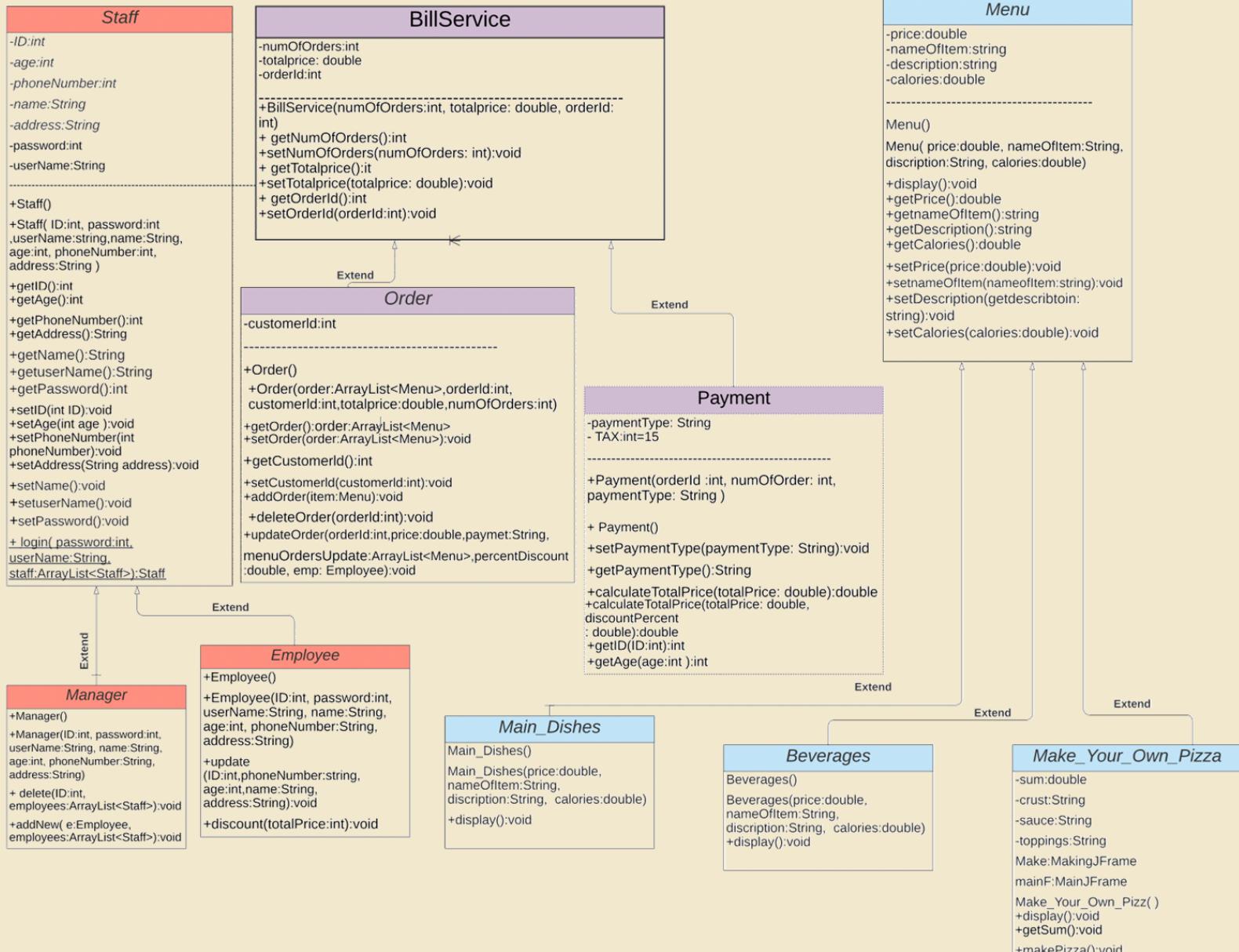
|   | User Role | Privileges  |
|---|-----------|---|
| 1 | Manager   | Add/delete employees, update and view employees info. |

2

Employee

Add/ delete menu items, add/delete/update orders.

## User's Roles



## Class Hierarchy:

## **Milestone 3:**

**Work Division:**

- We divided the tasks such that each student is responsible of a particular subtask, a subclass for example. Some students were working on the java code. Meanwhile, other students were working on the database design. However, we shared our knowledge together and worked collaboratively.

**Difficulties & how problems were solved:**

- It was difficult for us to find a suitable time for all the group members to join in, especially that it was difficult to conduct a physical meeting during the vacation.

**Solution:** We used virtual methods instead, such as WhatsApp & Zoom to help us maintain the work progress.

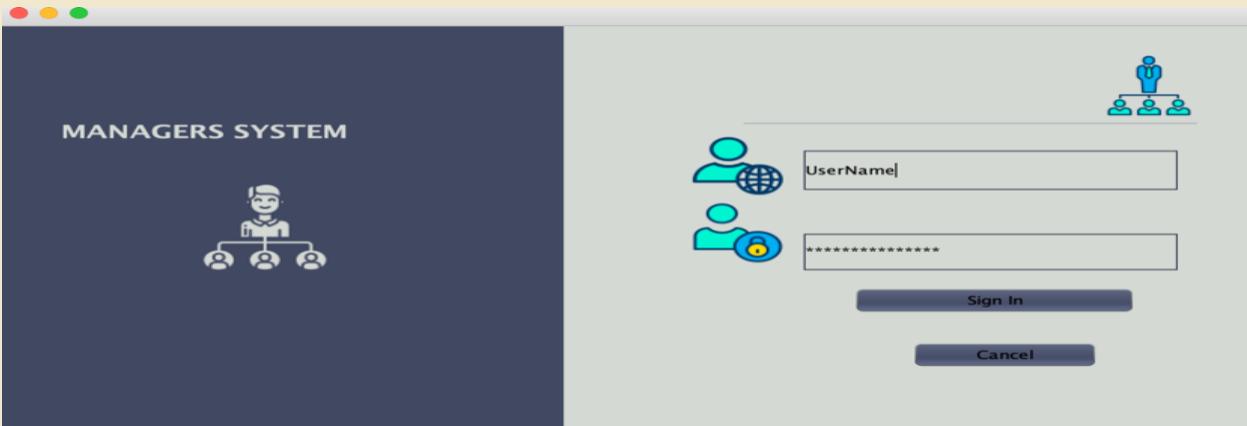
- Trying to use the sql database in a device other than where it has been created was difficult, and we spent some time trying to figure a way to solve this issue. We faced some difficulties while trying to connect the database as well.

**Solution:** At the end, we wrote the sql code again in the same device where we applied the connection, and it worked.

## Part1: Home Page:

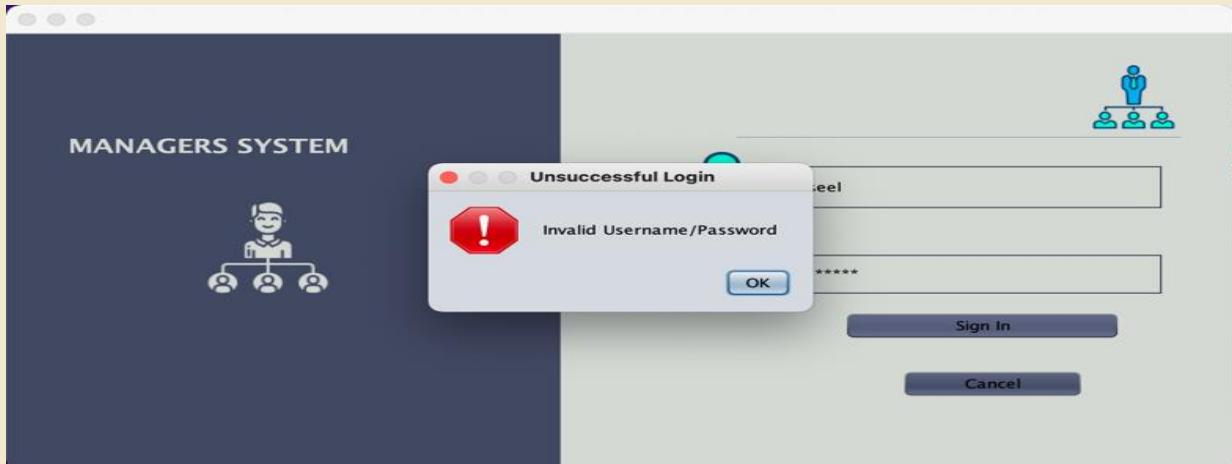


-When the user enters M it will display the manager login page

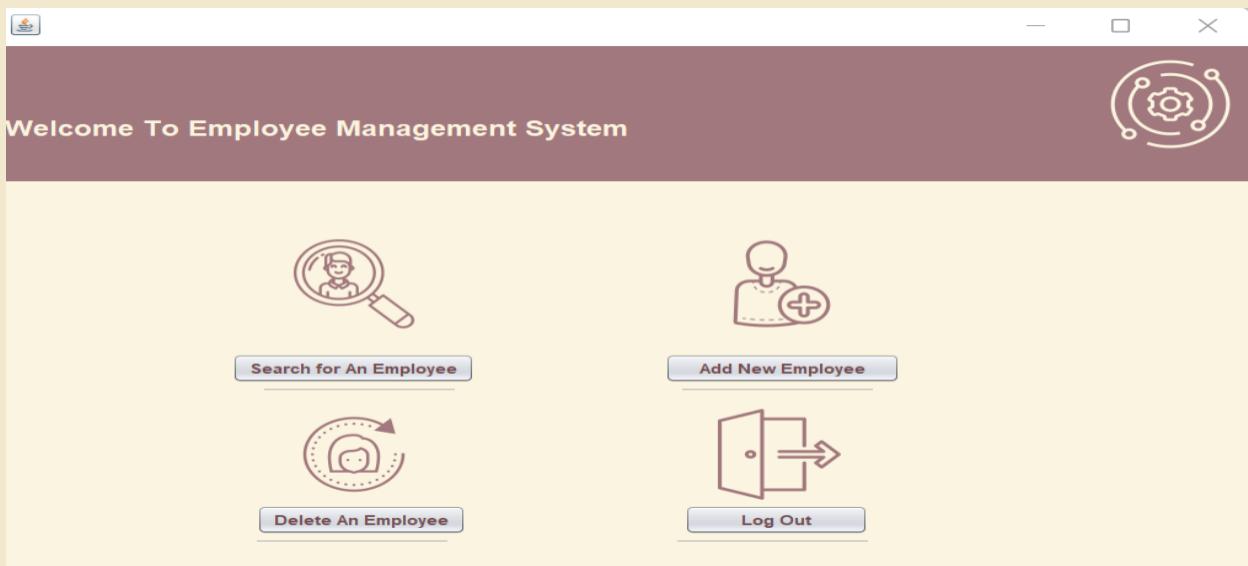


-If the user clicks cancel button, it will go back to home page

-If the user enters incorrect manager username and password and clicks the submit button it will display unsuccessful login message



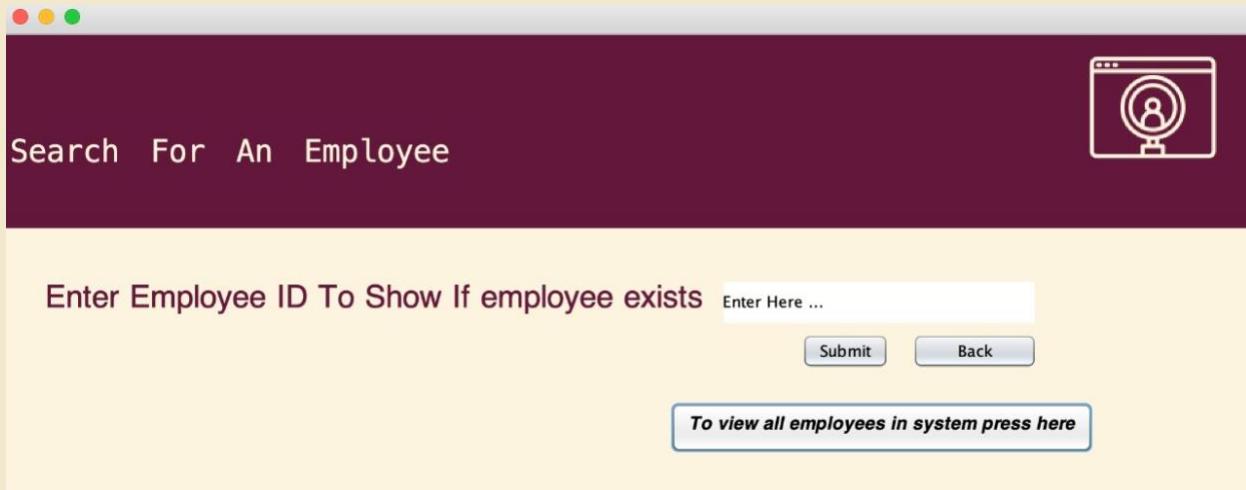
-If the user enters the correct manager username and password and clicks submit button it will display successful message and displays the employee management system interface



Here we have 4 cases:

-Logout button will exit the employee management system interface and display the home page

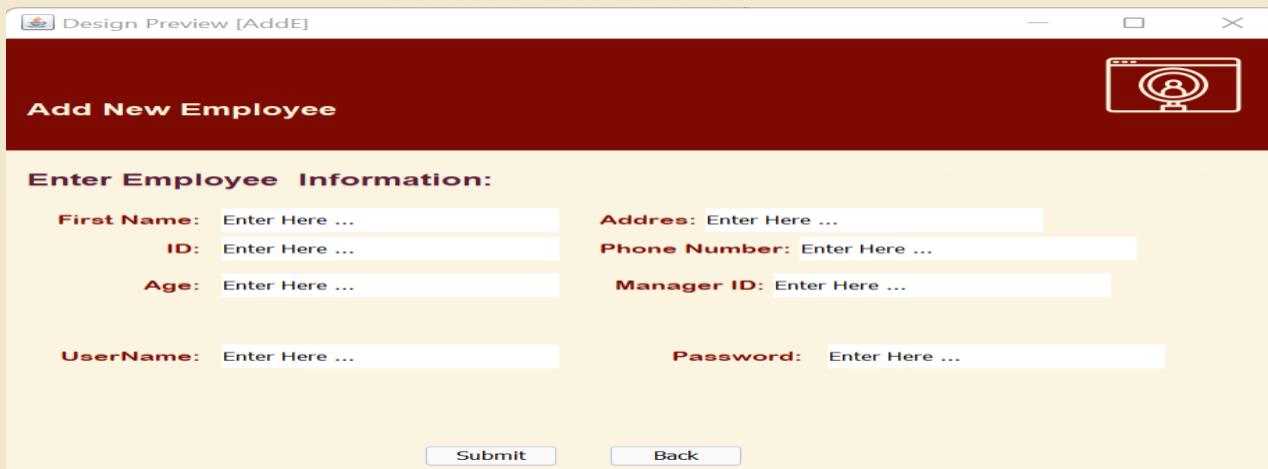
-Search for employee button: After selecting Search for an employee, this window will appear -by clicking "to view all employee in system press here" button it will create a file to show all the employees



-After entering employee ID, and clicking submit, it will check if this employee exists in the employee table in the database and it will display successful message



-Add new employee button: After selecting add new employee, it will show this window.



**-data base befor adding**

| # | ID      | name    | phoneNumber | address   | age | password | userName | MANAGER_ID |
|---|---------|---------|-------------|-----------|-----|----------|----------|------------|
| 1 | 1001221 | Lamya   | 0554456789  | Al Nuzha  | 36  | 5678     | lamaya2  | <NULL>     |
| 2 | 1004433 | Aseel   | 0552233880  | Al Rakah  | 30  | 1234     | aseel2   | <NULL>     |
| 3 | 1502321 | Fay     | 0561304668  | Al Shati  | 22  | 1234     | fay1     | 1001       |
| 4 | 1503821 | Dana    | 0552231828  | Al Rakah  | 24  | 9101     | dana3    | 1004       |
| 5 | 1503822 | Thuraya | 0552231828  | Al rawada | 24  | 9101     | thuraya  | 1001       |
| 6 | 1504542 | Sara    | 0543322120  | Al Wahan  | 23  | 5678     | sara2    | 1001       |
|   |         |         |             |           |     |          |          |            |

**-After entering the values for all fields, and clicking Submit, the Employee will be added to the Employees table in the database and a confirmation message will be displayed.**

Add New Employee

Enter Employee Information:

First Name: Layan

ID: 152039

Age: 27

UserName: Layan27

Password: 1234

Message

Adding new Employee to the system is successful.

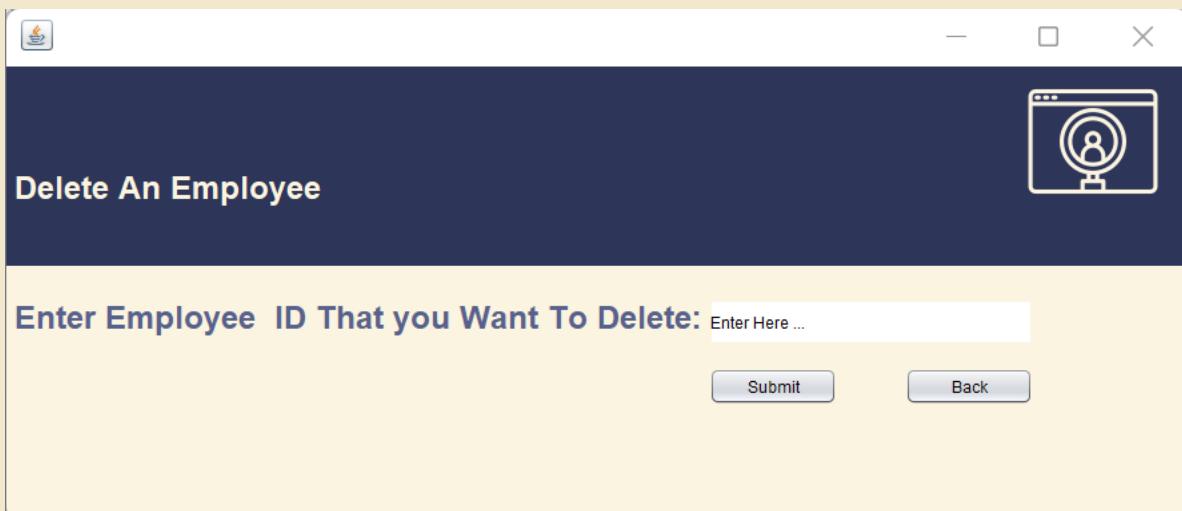
OK

Submit Back

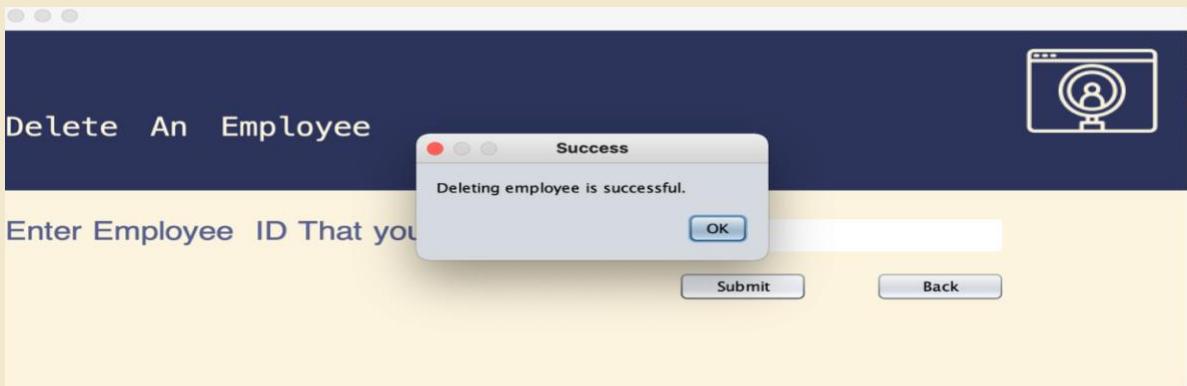
**-data base after adding**

| # | name            | phoneNumber | address   | age | password | userName | MANAGER_ID |
|---|-----------------|-------------|-----------|-----|----------|----------|------------|
| 1 | 152039 Layan    | 0553947732  | Riyadh    | 27  | 1234     | Layan27  | 100443     |
| 2 | 1001221 Lamya   | 0554456789  | Al Nuzha  | 36  | 5678     | lamaya2  | <NULL>     |
| 3 | 1004433 Aseel   | 0552233880  | Al Rakah  | 30  | 1234     | aseel2   | <NULL>     |
| 4 | 1502321 Fay     | 0561304668  | Al Shati  | 22  | 1234     | fay1     | 1001221    |
| 5 | 1503821 Dana    | 0552231828  | Al Rakah  | 24  | 9101     | dana3    | 1004433    |
| 6 | 1503822 Thuraya | 0552231828  | Al rawada | 24  | 9101     | thuraya  | 1001221    |
| 7 | 1504542 Sara    | 0543322120  | Al Wahan  | 23  | 5678     | sara2    | 1001221    |

**-Delete an employee button:** After selecting Delete an employee, this window will show asking for ID.



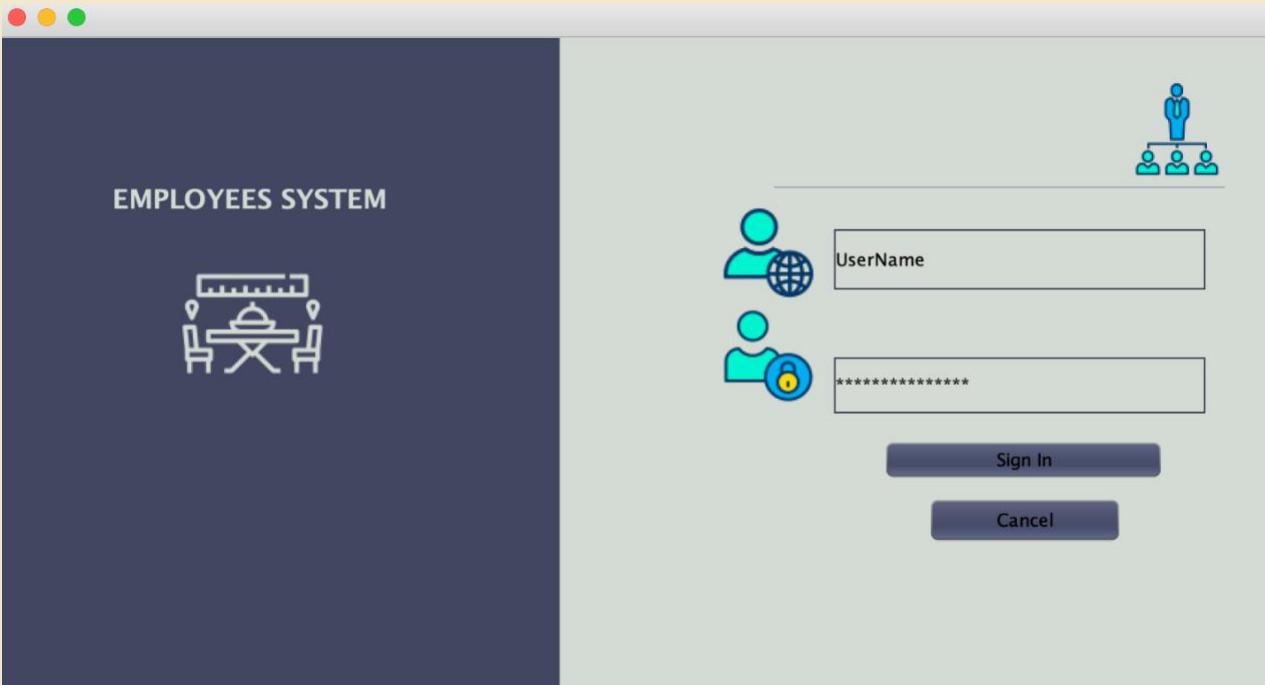
**-After entering the ID of the selected employee and clicking submit, the employee will be deleted from the employee table in the database and a successful message will be displayed**



## Part 2: Home page

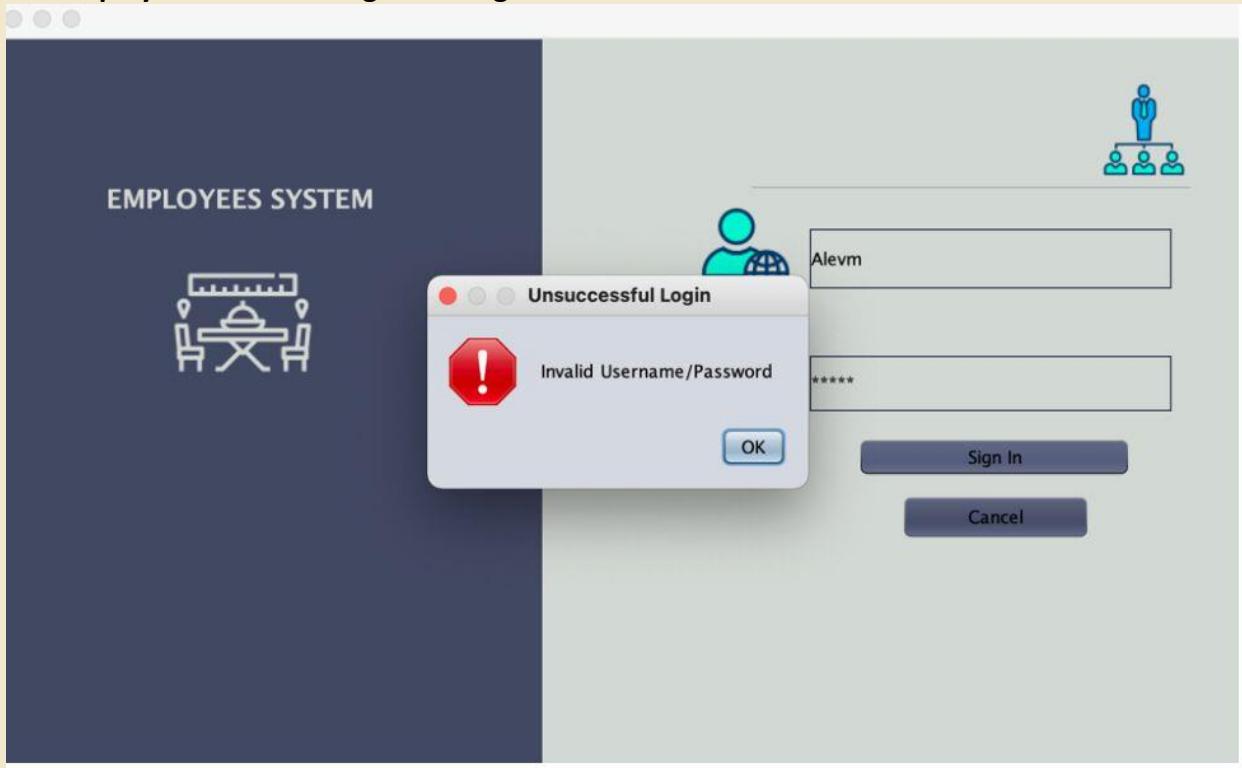


-When the user enters E it will display the employee login page:

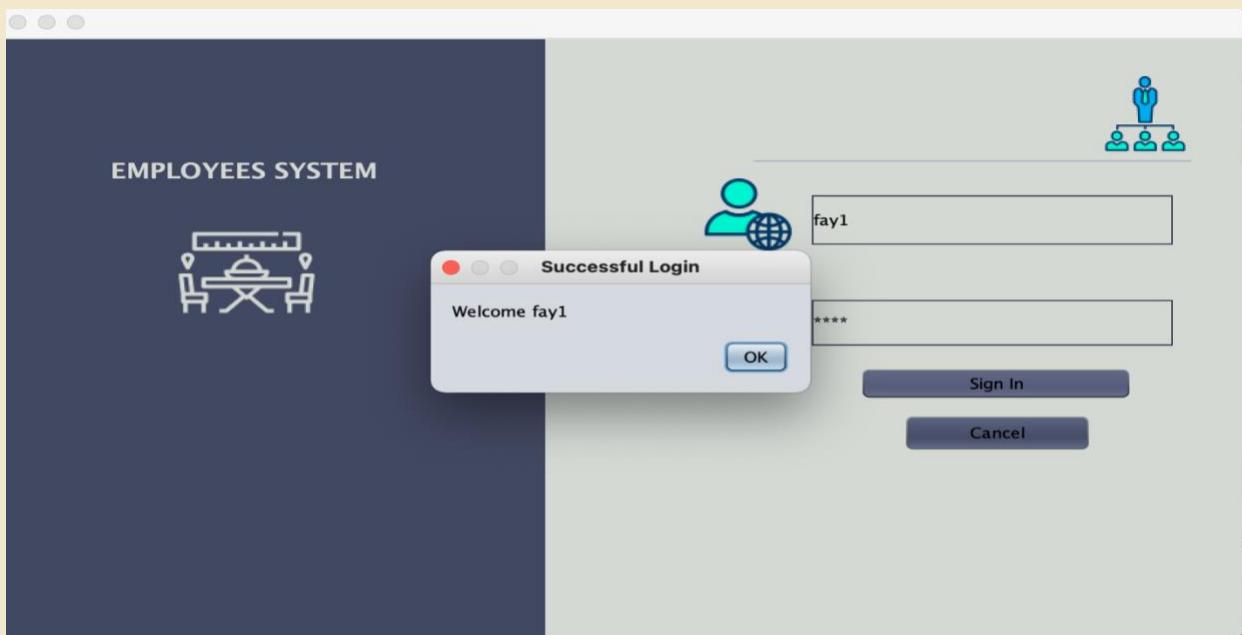


-If the user clicks cancel button, it will go back to home page

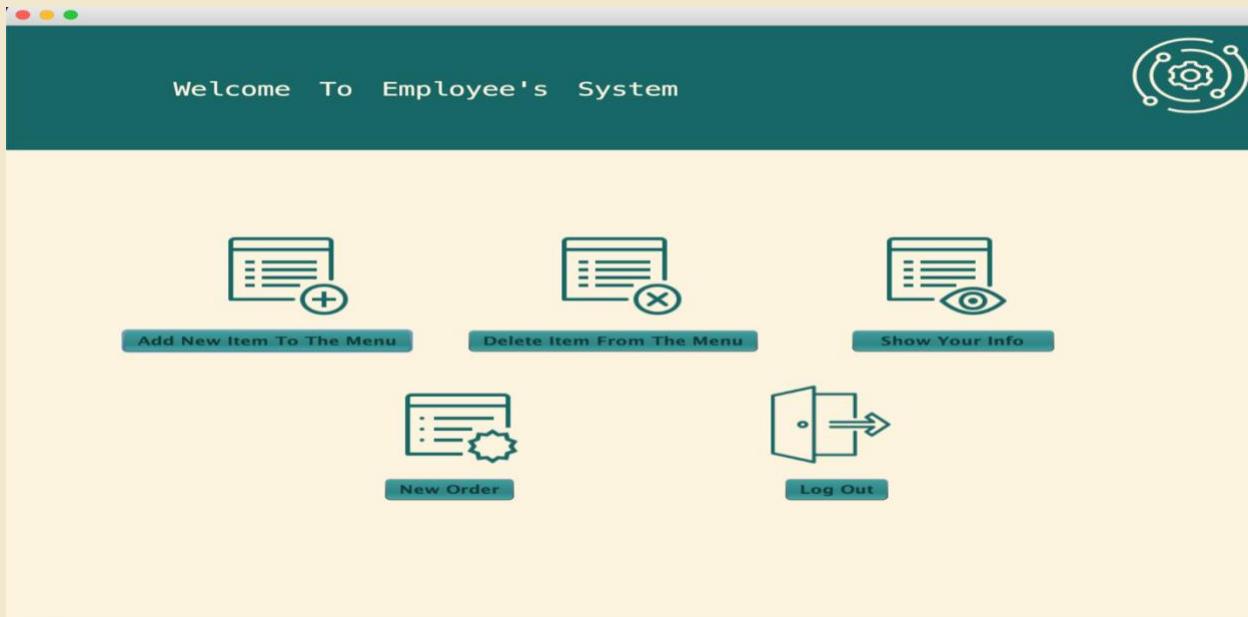
-If the user enters incorrect employee username and password and clicks the submit button it will display unsuccessful login message



-If the user enters the correct employee username and password and clicks submit button it will display successful message and displays the customer management system interface



## INTERFACE FOR EMPLOYEE SYSTEM



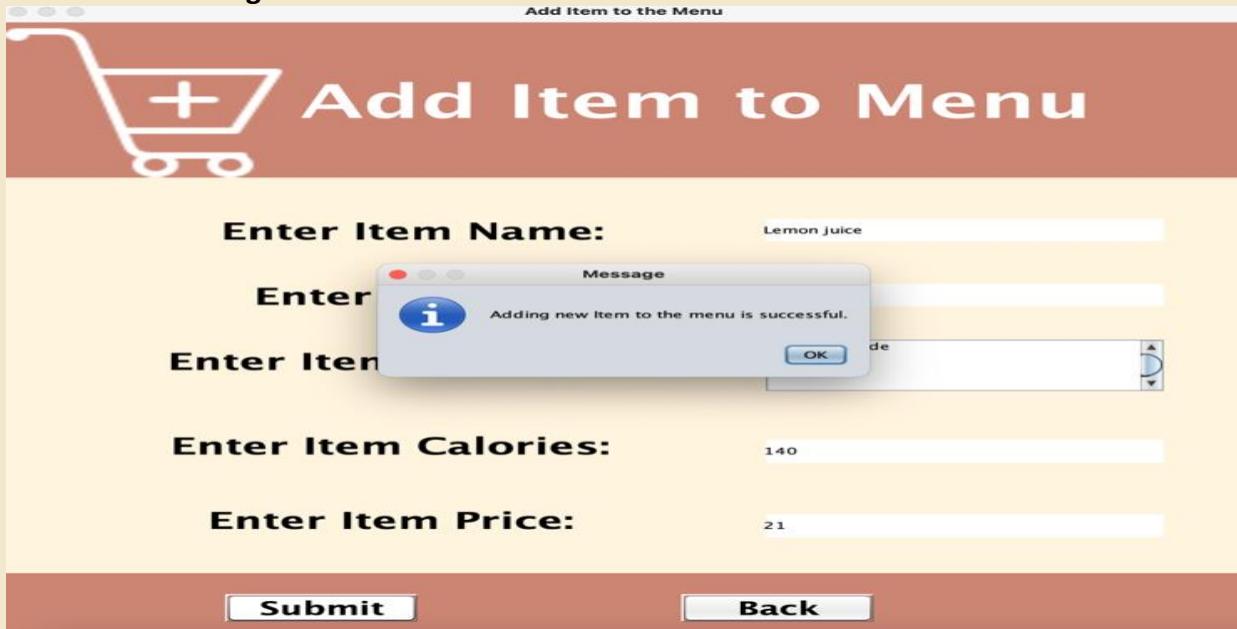
Here we have 5 cases:

- 1-Logout button: will exit the customer management interface and go back to home page
- 2-Add new item to the menu: After selecting add new item, this window will appear.

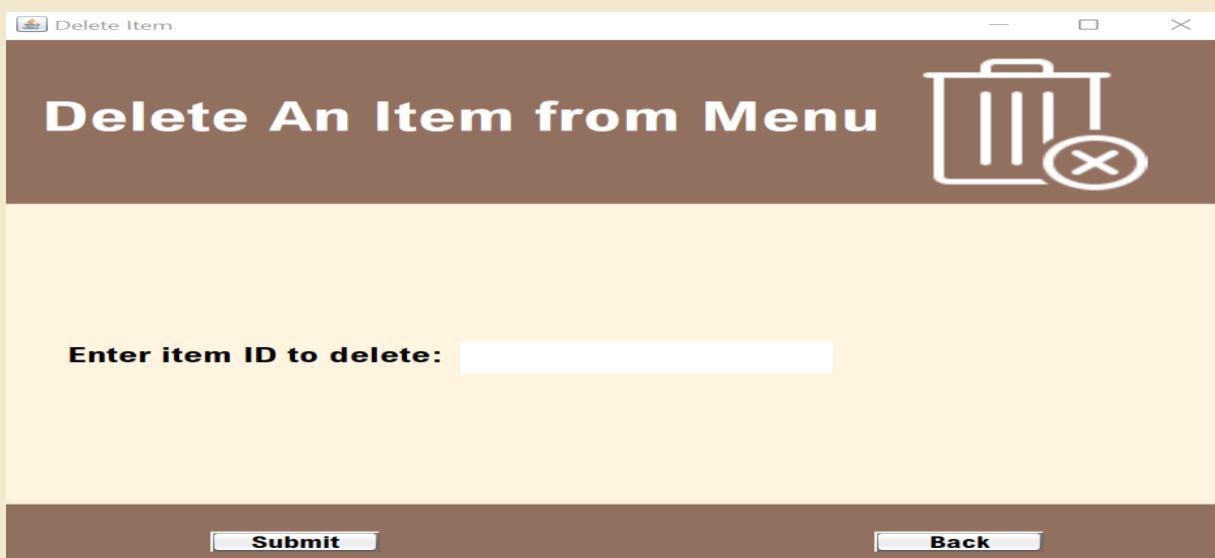
A screenshot of a window titled "Add Item to the Menu". The window has a brown header bar with a shopping cart icon and the title. The main body of the window contains five input fields: "Enter Item Name:" with a text input field, "Enter Item ID:" with a text input field, "Enterd Item Description:" with a text input field containing a scroll bar, "Enter Item Calories:" with a text input field, and "Enter Item Price:" with a text input field. At the bottom of the window are two buttons: "Submit" and "Back".

-After entering the required fields and clicking submit, new item will be added to the menu table in the database and successful message will be displayed.

- Back button will go back to EMPLOYEE SYSTEM interface

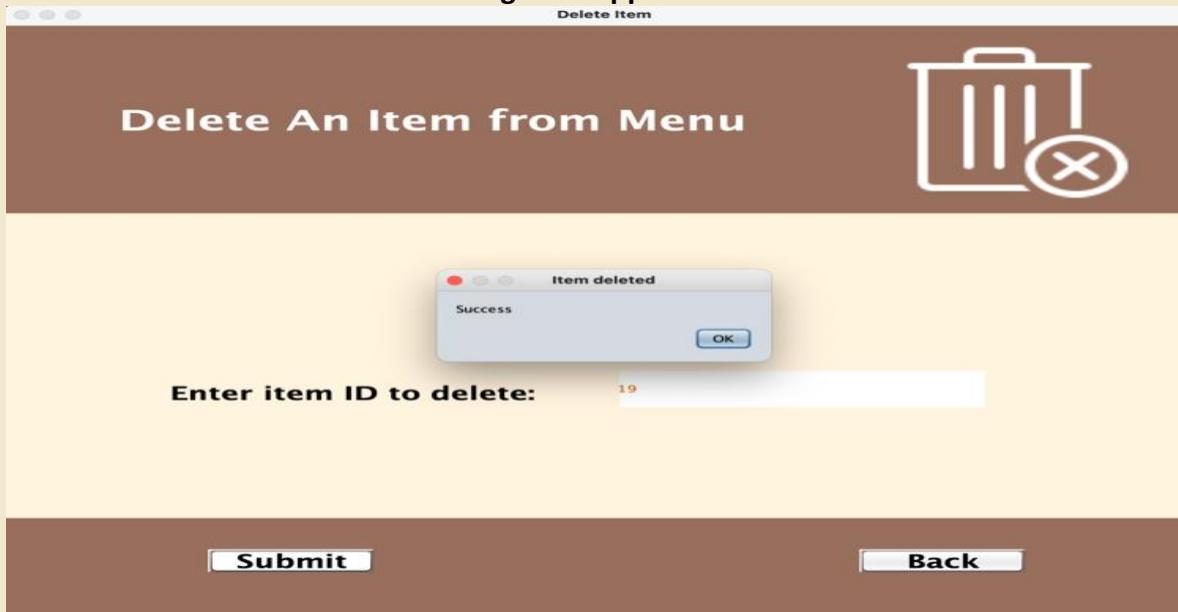


3-Delete item from the menu button: After selecting delete item, this window will appear.

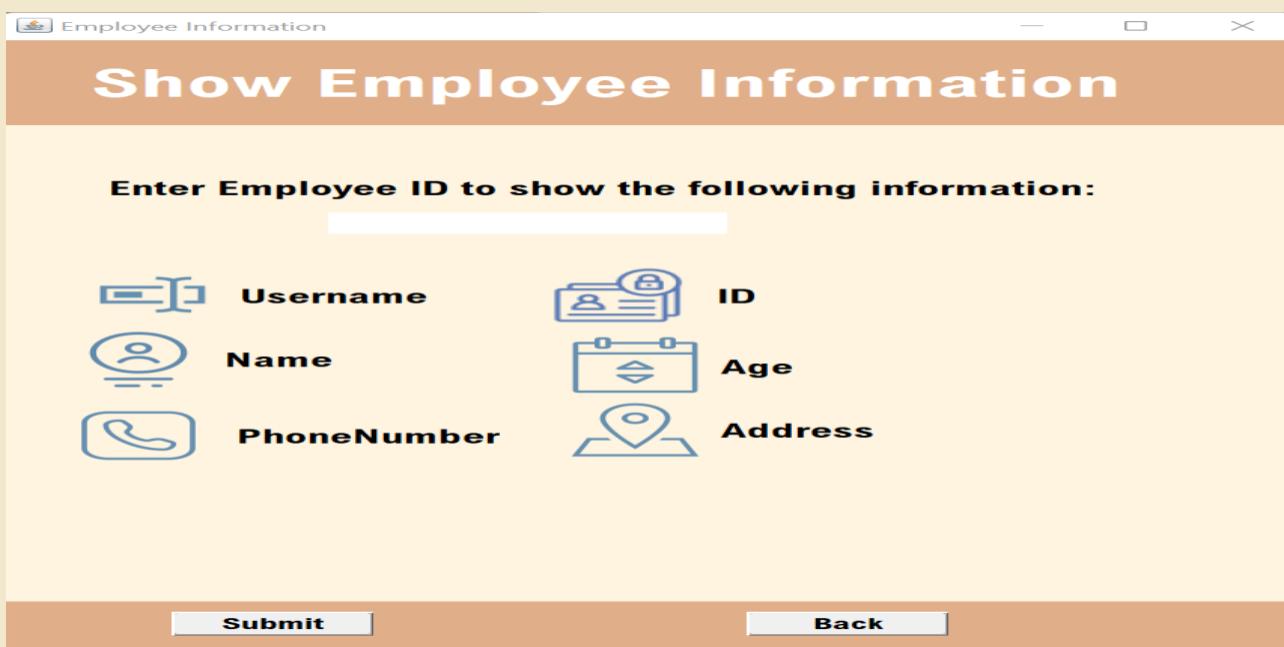


- Back button will go back to customer management system interface

-After entering the item ID and clicking submit, the item will be deleted from the menu table in the database and successful message will appear.

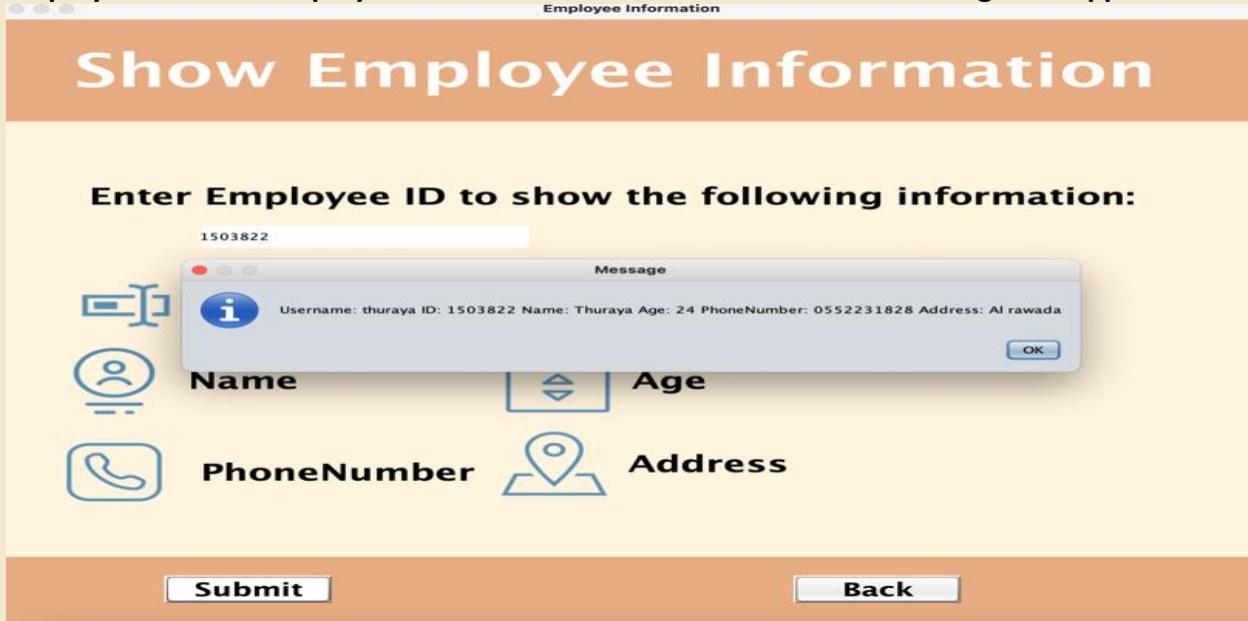


4-Show your info button: After selecting show your info, this window will appear



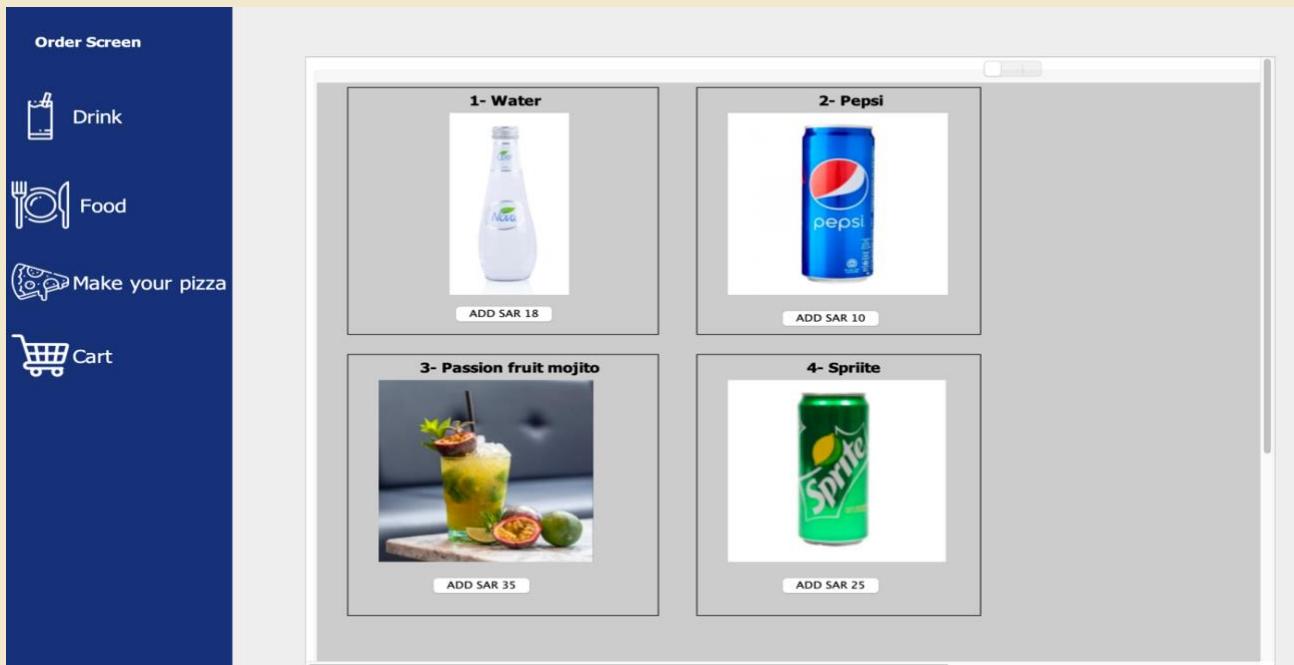
- Back button will go back to customer management system interface

-when you enter the employee ID and click submit it will show all information about the employee from the employee table in the database and successful message will appear.



5-new order: after selecting new order a window will appear which contain :

1-Drinks list:



## 2-food list

Order Screen

The interface shows a sidebar with icons for Drink, Food, Make your pizza, and Cart. The main area displays six food items in a grid:

- 1- Pizza: ADD SAR 73
- 2- Risotto: ADD SAR 79
- 3- Alfredo pasta: ADD SAR 85
- 4- Spaghetti: ADD SAR 67
- 5- Burrata salad: ADD SAR 69
- 6- Mixed sea food pasta: ADD SAR 97

## 3- The order cart:

-Here all the requests requested by the customer are added and the total price and the payment method wither its cash or credit card

Order Screen

The interface shows a sidebar with icons for Drink, Food, Make your pizza, and Cart. The main area displays an order summary table and payment options:

| Item | QTY | Price |
|------|-----|-------|
|      |     |       |

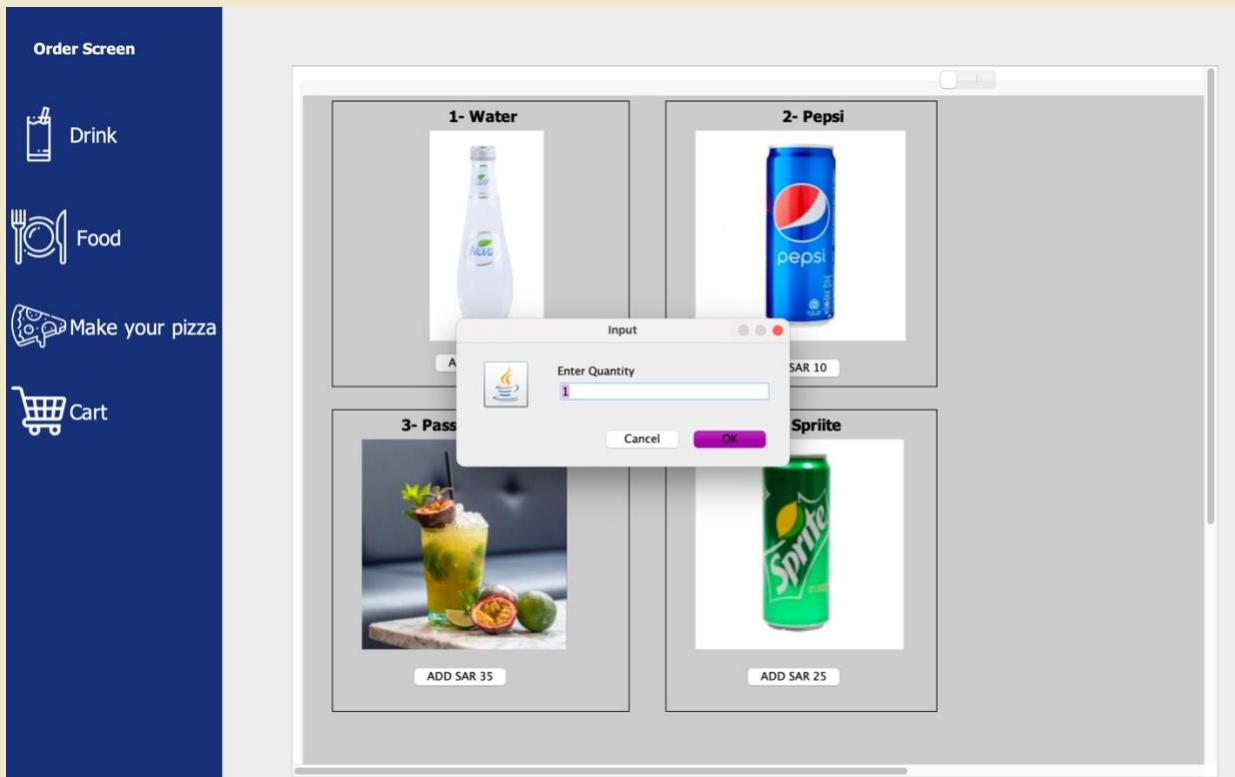
Total: 0.0

Any other requests?

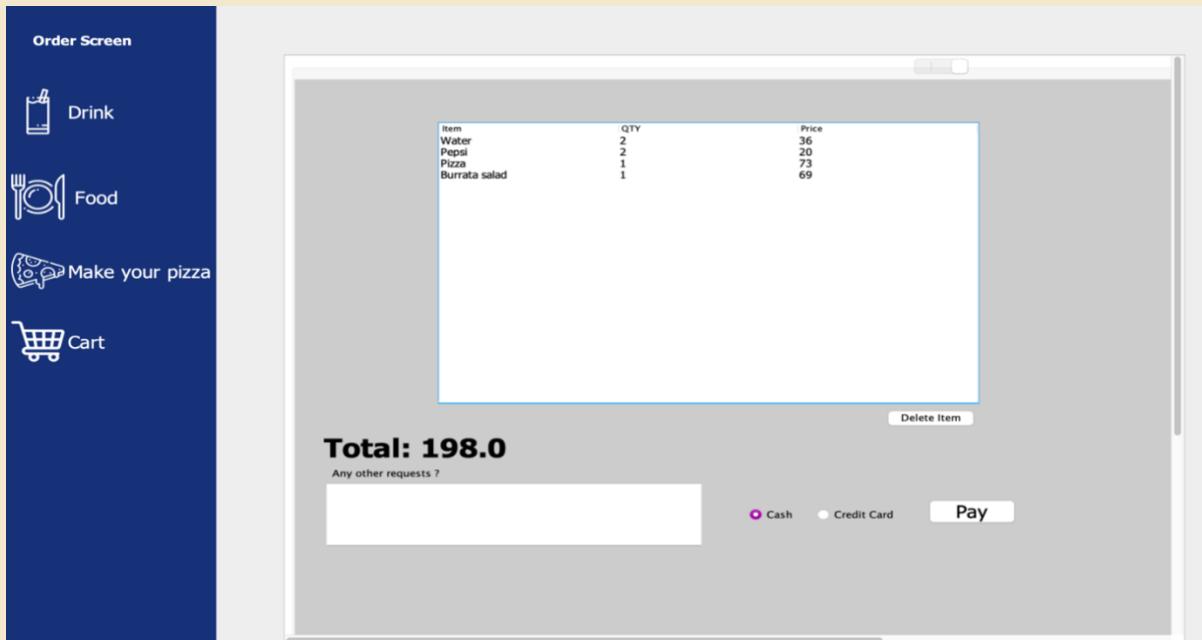
Cash    Credit Card

Pay

-Adding an order to the cart and specify the quantity: Determine the quantity that the customer wants from each item

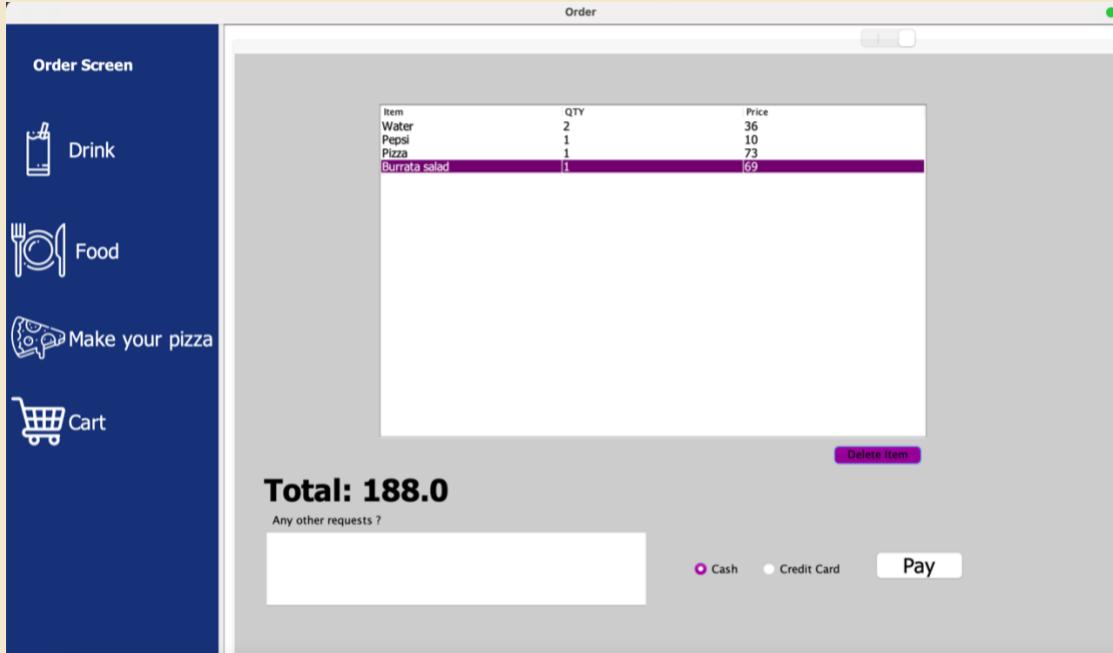


-If the customer wanted to View the selected orders by clicking on cart it will show the item that selected and the total and the payment method



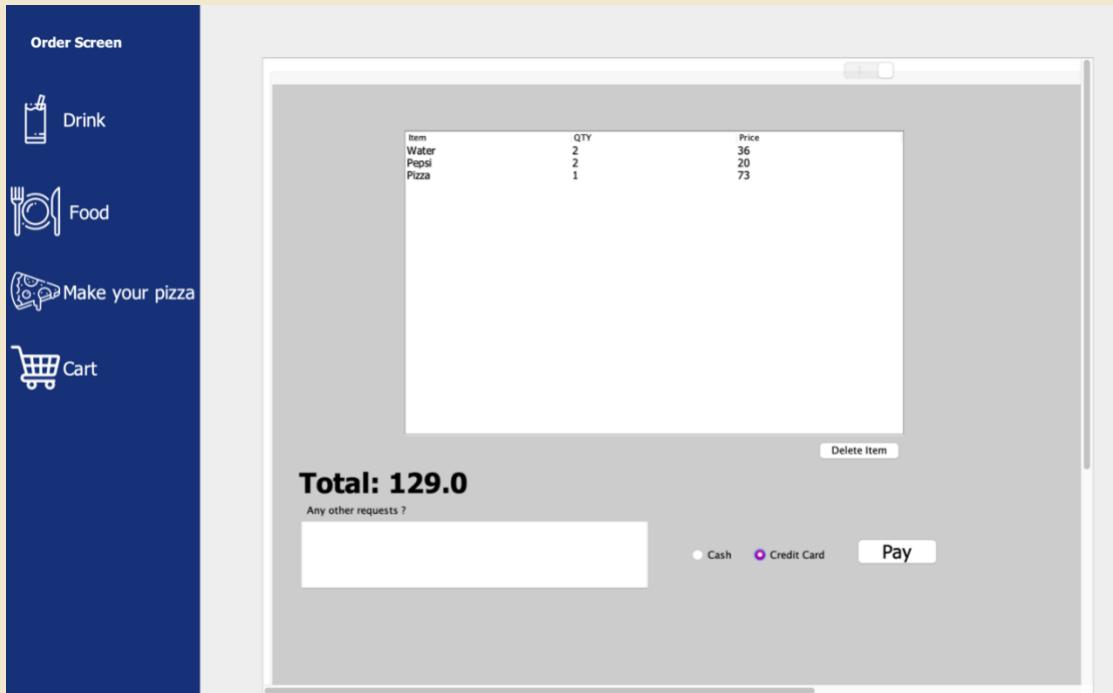
-If the customer wants to delete one item or all

-Either select one item from the list and press the delete item button, or select the entire list and press the delete item button

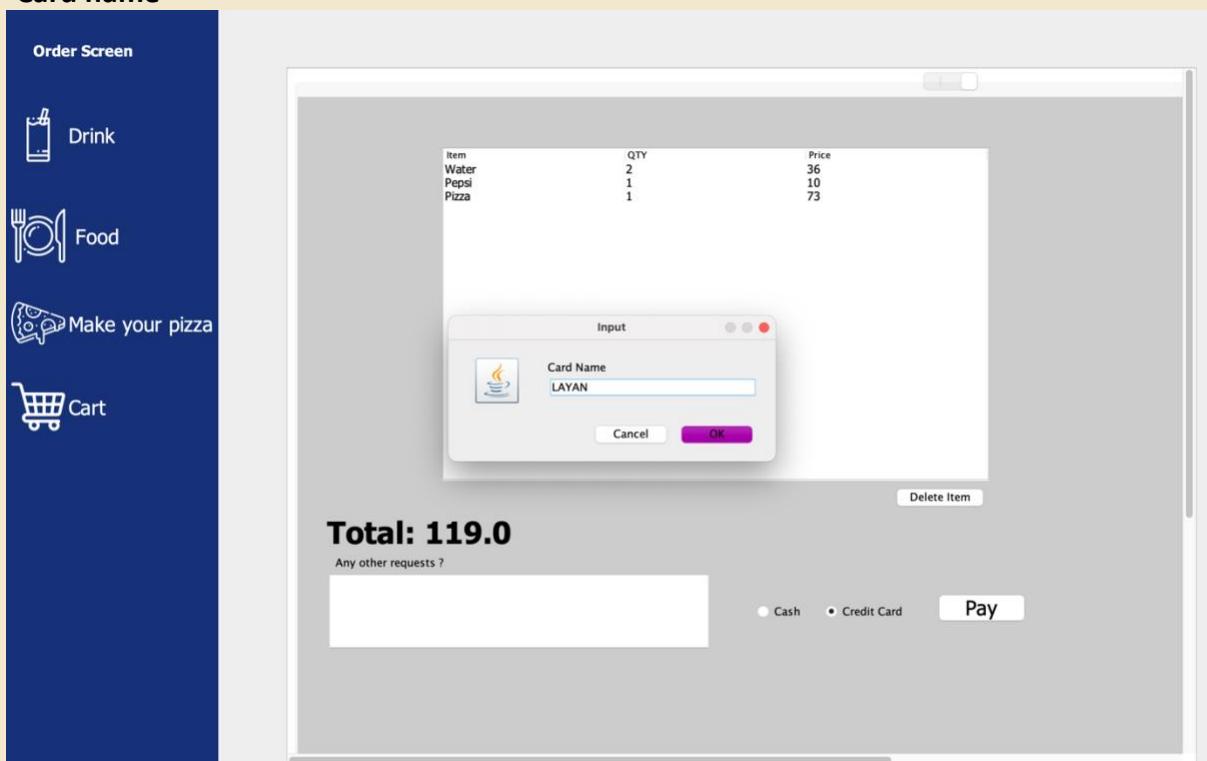


- after deleting an item, the customer can add more orders by clicking drinks or food and adding item again

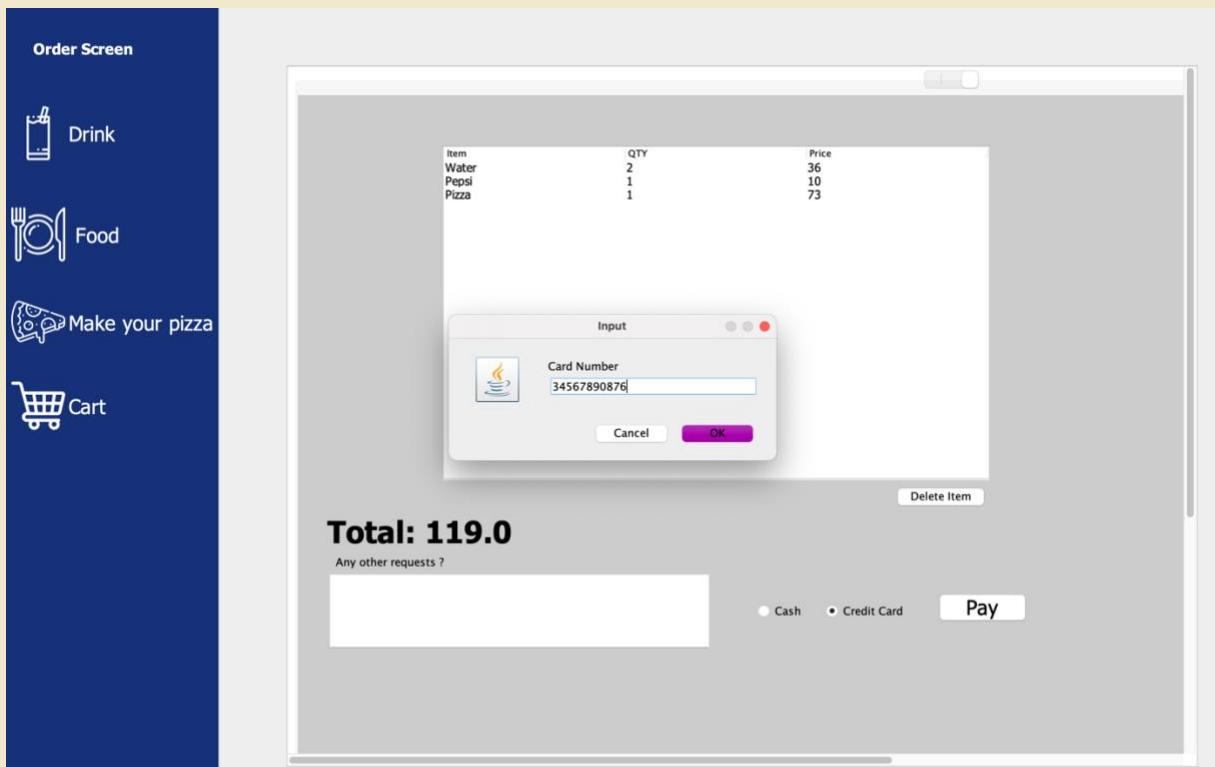
-If coatomer finished with the order, he should choose the payment method, if its credit card he should write its information



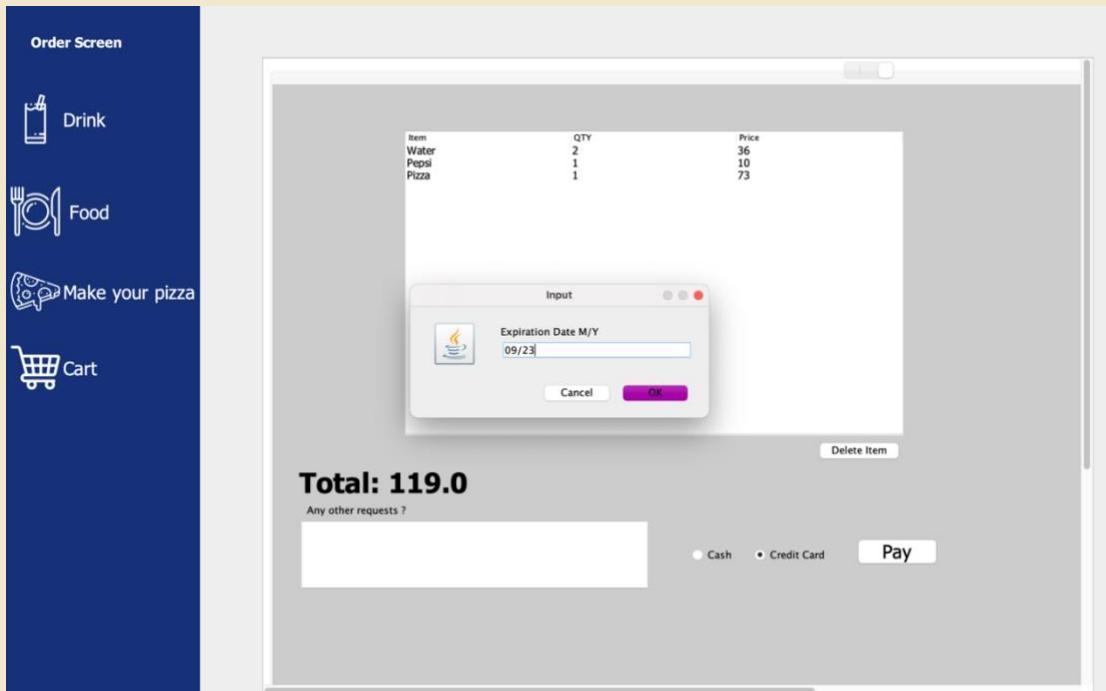
### -Card name



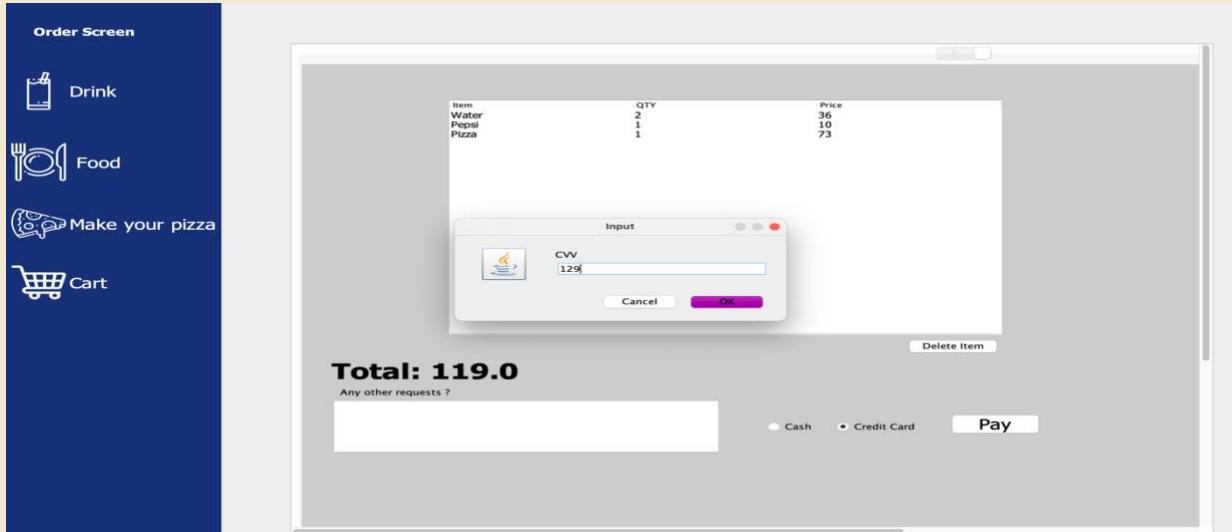
### -Card number



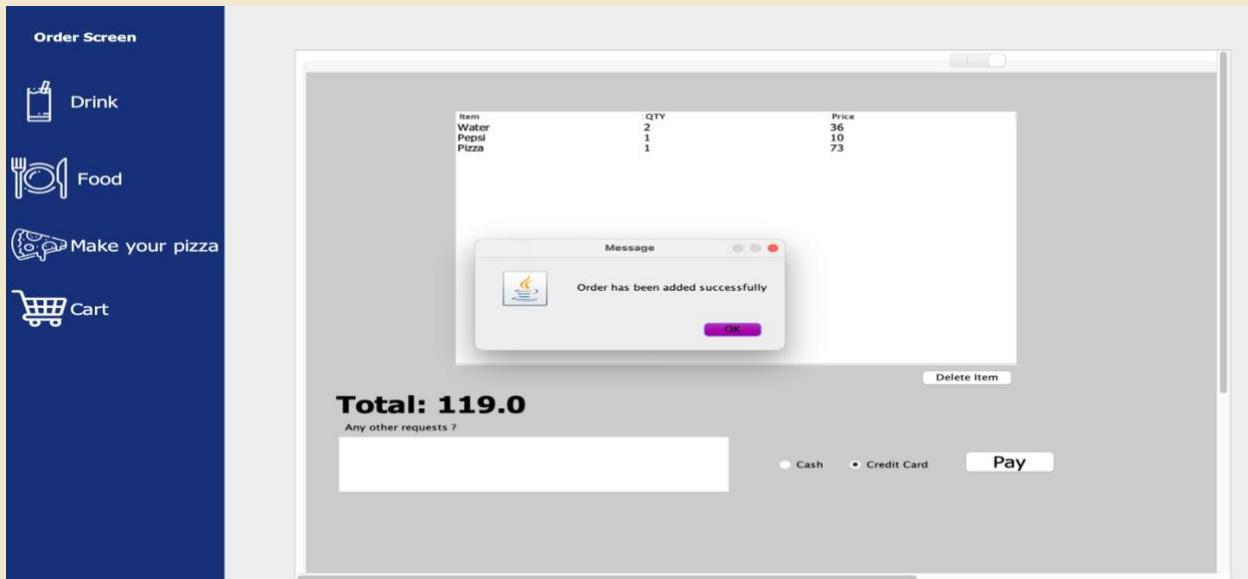
-Expiration date month and year



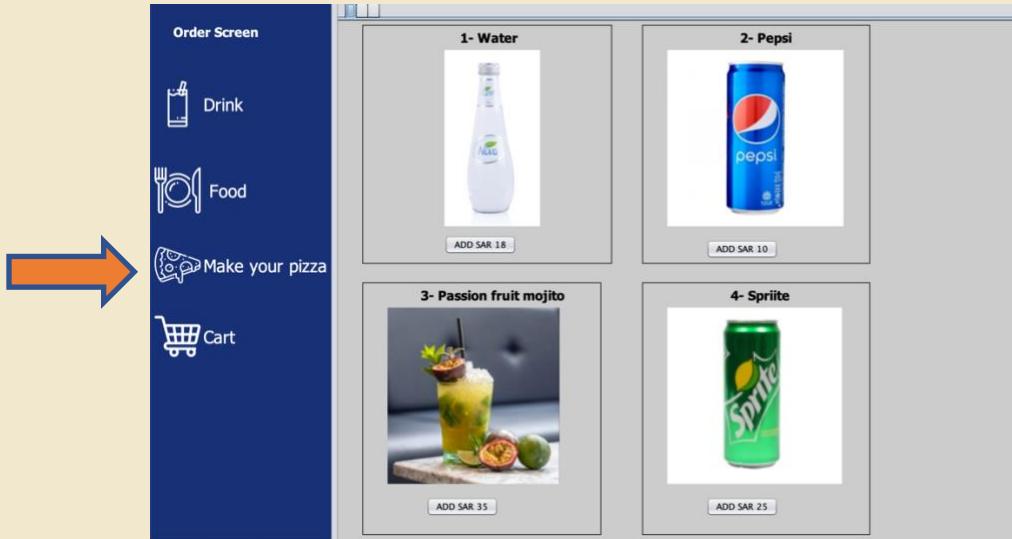
-the CVV the code in the back of the card



-After the customer done with the credit card information this message will appear to show that the order added and paid successfully



4-To start making your own Pizza click this in the menu :



-You have the option to cancel and no further effects will be done :



-Start button will bring another window for making Pizza

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Making_Frame jf2 = new Making_Frame();
    jf2.show();
    dispose();
```

-Cancel will dispose the current one

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    dispose();
}
```

-You start customizing your Pizza, you can navigate panels through the buttons or from the top



**choose your**

- 1**  
Crust
- 2**  
Sause
- 3**  
Toppings
- 4**  
your pizza
- 5**  
receipt

**choose your crust ...**

Thin       Thick       Stuffed

**choose your sause...**

normal tomato       spicy tomato       arigano sause

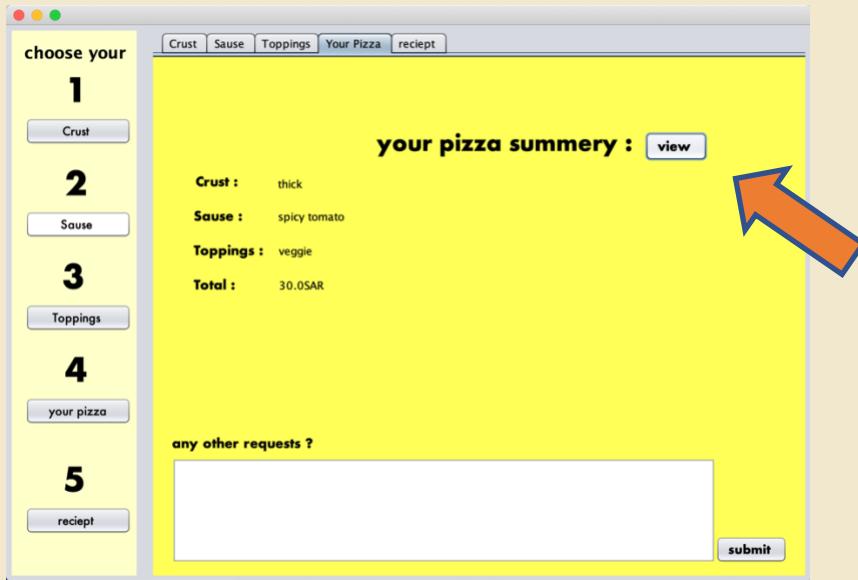
**choose your toppings...**

A dropdown menu showing toppings:

- classic
- classic
- vegie
- hawaiin**
- arabic

super cheesy      ?      super cheesy

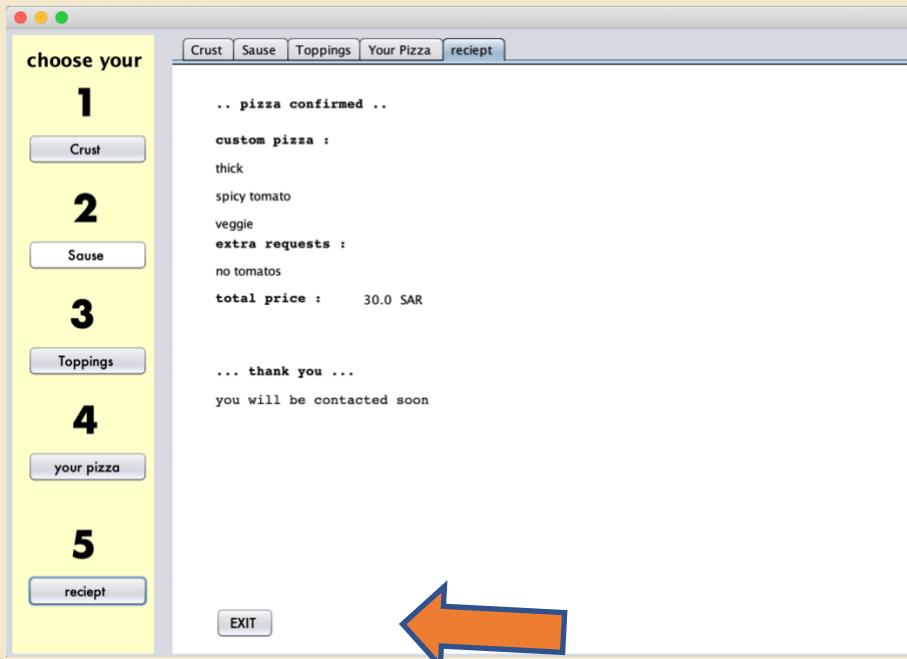
-clicking view you will let you see your order and here you have the chance to go back and edit it or add extra requests



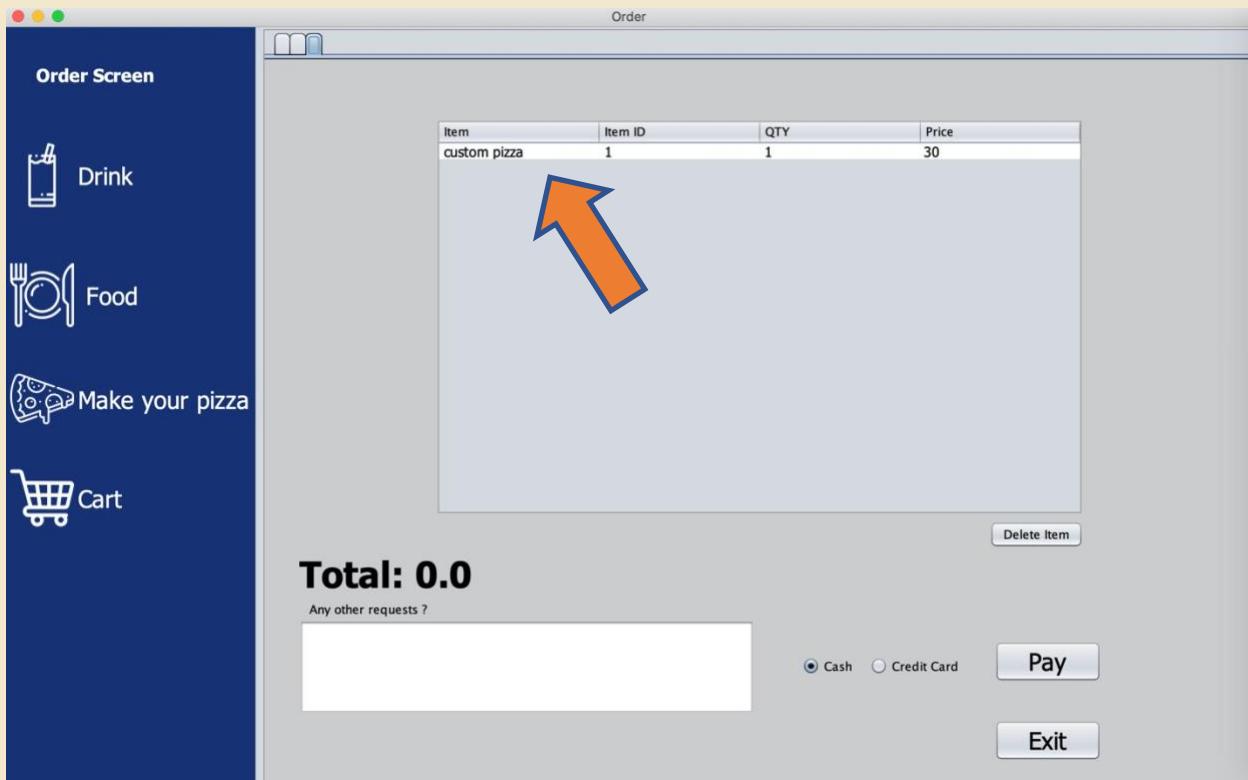
-By clicking “submit” the button will change to “submitted” your bill is generated and the custom pizza is added to your cart in the previous menu window



#### -Your generated Bill



#### -Custom pizza added to the cart



-To make a button action effect a different Jframe and take the data to a different window we did the following :

-We created an instance in the constructor of the menu table and a model and assigned them to "this" or the current model and instance

```
/*
public OrderGUI() {

    initComponents();
    model = new DefaultTableModel();
    jTable1.setModel(model);

    model.addColumn("Item");
    model.addColumn("Item ID");
    model.addColumn("QTY");
    model.addColumn("Price");
    jLabel25.setText("Total: 0.0");
    instance = this;
    mdl = model;
}
```

-In the second jFrame we made the action performed method effect the other jFrame.instance we created and had access to the methods in the other jFrame

```
private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) { // TODO add your handling code here:  
    try{  
  
        OrderGUI.instance.mdl.addRow(new Object[]{"custom pizza", "1","","","30"});  
  
    }  
    catch(NullPointerException e ){  
        e.printStackTrace();  
    }  
    catch(RuntimeException e ){  
        e.printStackTrace();  
    }  
    catch(Exception e ){  
        e.printStackTrace();  
    }  
    dispose();  
  
}
```