

**VIETNAM NATIONAL UNIVERSITY - HO CHI MINH
CITY**

INTERNATIONAL UNIVERSITY

**SCHOOL OF COMPUTER SCIENCE AND
ENGINEERING**



**PRINCIPLES OF DATABASE
MANAGEMENT**

**IT079IU
FINAL REPORT**

Course by Assoc. Prof. Nguyen Thi Thuy Loan

TOPIC: Food delivery application

BY GROUP 09 – MEMBER LIST

1. Nguyễn Quốc Tuấn	ITITI22177	Team Leader
2. Trần Lưu Hồng Phương	ITITI22198	Team Member
3. Lương Quang Huy	ITITI22076	Team Member
4. Nguyễn Quách Dịch Thịnh	ITCSI22280	Team Member
5. Nguyễn Phúc Đạt	ITITI22028	Team Member
6. Bùi Nguyễn Thảo Vân	ITITI22218	Team Member

Table of contents

Chapter 1: INTRODUCTION.....	6
1. Abstract	6
2. Objective	6
3. Technologies	6
Chapter 2: TASK & TIMELINE.....	7
1. Contribution	7
2. Project Timeline	7
Chapter 3: PROJECT ANALYSIS.....	9
1. Design Analysis	9
2. Database Design.....	11
2.1 Database Creation	11
2.2 Tables Creation	12
2.3 Sample data insertion.....	14
3. SQL Usage	16
3.1 Login.....	16
3.2 Sign-up.....	16
3.3 Password Reset	17
3.4 Restaurant Side	17
3.5 User Side.....	18
Chapter 4: Features and Functionalities.....	21
1. User Management	21
1.1 Login	21
1.2 Sign-up.....	21
1.3 Password Reset	22
2. Functionalities.....	22
2.1 Restaurant Side	22
2.2 User Side.....	24
Chapter 5: Security Measures	31

1. Data Protection:	31
2. SQL Injection Prevention:	31
Chapter 6: Conclusions	33
1. Achieved Goal	33
2. Future Work	33
- UI Improvement	33
- Real-Time Delivery Tracking.....	33
- Estimated Time of Arrival (ETA) Calculation	33
- Advanced Food Sorting Functionality	33
3. Conclusion	34
Chapter 7: References	35

Table of Figures

Figure 1. Entity Relationship Diagram	9
Figure 2. Schema design created by using SQL Server	10
Figure 3. Login panel	21
Figure 4. Sign-up panel	21
Figure 5. Password reset panel.	22
Figure 6. Listing Food and Price	23
Figure 7. Add food to menu.	23
Figure 8. Assign Delivery Person to Order Panel	24
Figure 9. User search for Food	25
Figure 10. Users view the Restaurant list.	25
Figure 11. User view Restaurant and Menu	26
Figure 12. Add to Cart Panel	26
Figure 13. Add to Cart Panel	27
Figure 14. Cart Panel	27
Figure 15. Removing Food from Cart Panel	28
Figure 16. Payment Panel	29
Figure 17. Choose payment options	29
Figure 18. Password Hash	31
Figure 19. Prepared Statement Implementation	32

Table of tables

Table 1. Contribution	7
Table 2. Project timeline	8
Table 3. NF Form	10
Table 4. Database attributes	12

Chapter 1: INTRODUCTION

1. Abstract

This is a project for Principles of Database Management course. The Food Delivery project is an initiative to develop a practical mobile application for ordering and delivering food. In response to the increasing demand for convenient food services, our project aims to create a user-friendly platform connecting restaurants, delivery personnel, and customers. By focusing on simplicity and real-world usability, we aim to provide an efficient and enjoyable way for people to order their favorite meals using their smartphones or computers. This report outlines the project's key objectives, including database design, system integration, user experience enhancement, real-time tracking, and payment security.

2. Objective

- **Database Optimization:** Design and implement a database system that adheres to the Third Normal Form (3NF) to ensure data integrity, reduce redundancy, and improve data efficiency.
- **Seamless Frontend-Backend Integration:** Develop a robust connection between the frontend user interface and the database backend, ensuring smooth data communication and interaction between users and the system.
- **User Registration and Authentication:** Create user-friendly sign-up and login functionalities for both customers and restaurant owners, prioritizing security and ease of use.
- **Streamlined Restaurant Menu Management:** Develop a user-friendly interface for restaurant owners to efficiently list their food items and menus on the platform, allowing for easy updates and modifications.
- **Efficient Food Search and Ordering:** Implement a powerful search feature that enables customers to browse and find food items from various restaurants easily. Additionally, provide a seamless ordering process that allows customers to place orders effortlessly.

3. Technologies

- Microsoft SQL Server Management Studio
- Java
- JetBrains IntelliJ IDEA GUI
- JDBC driver

Chapter 2: TASK & TIMELINE

1. Contribution

Name	Contribution
Nguyễn Quốc Tuấn	20%
Trần Lưu Hồng Phương	15%
Lương Quang Huy	15%
Nguyễn Quách Dịch Thịnh	15%
Nguyễn Phúc Đạt	20%
Bùi Nguyễn Thảo Vân	15%

Table 1. Contribution

2. Project Timeline

Stage	Task	Member	Week
Planning	Researching information about Food Delivery	All	1
	Using food delivery app for references	All	2
	Choosing features and technology	All	3
			4
Database Design	Specify information which is necessary to be stored	Tuan, Dat	5
	Design relational model	Tuan, Dat	
	Design ERD	Dat	
	Review and adjusting	Tuan, Dat	
Implementations	Setting up database, relationships, keys in SQL	Phuong, Huy	6
	Collecting and inserting sample data	Phuong, Huy	
	Simulating functions by implement SQL queries	All	7
	Developing login and register function for User and Restaurant Owner	Tuan	8

	Developing resetting function for User and Restaurant Owner	Tuan	
	Developing list food in menu for Restaurant Owner	Phuong	
	Developing search food function for User	Phuong	
	Developing view restaurants for User	Huy, Thinh	
	Developing order food from selected restaurant for User	Huy, Thinh	
	Developing show cart and payment function for User	Van	
	Merging functions into one app	Dat	9
Testing	Reviewing Code, detecting bug and fixing	Dat, Thinh	10
	Stabilizing and modifying app	Dat, Thinh	11
Presentation	Final report	All	12
	Presentation slides	Thinh, Huy	

Table 2. Project timeline

Chapter 3: PROJECT ANALYSIS

1. Design Analysis

The project aims to make a food delivery app that can make a connection between customers and restaurant owners. The system will be built with Microsoft SQL Server and Java. We use JDBC driver for database connections and external libraries for the UI and features.

The app has functions including register account, login account for customers and restaurant owner. Customers can search and access the restaurant's menu, add food to cart and make an order. The restaurant can list their foods and menu.

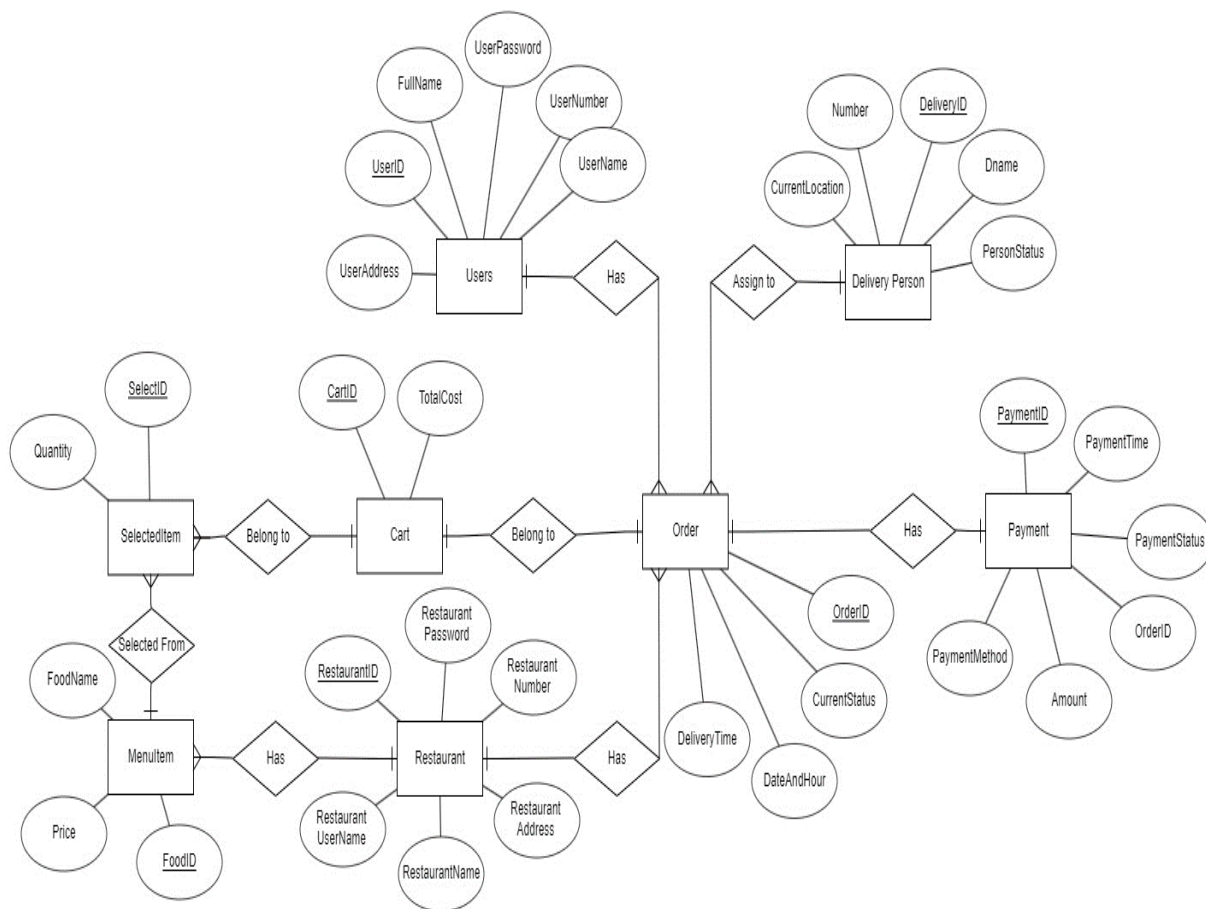


Figure 1. Entity Relationship Diagram

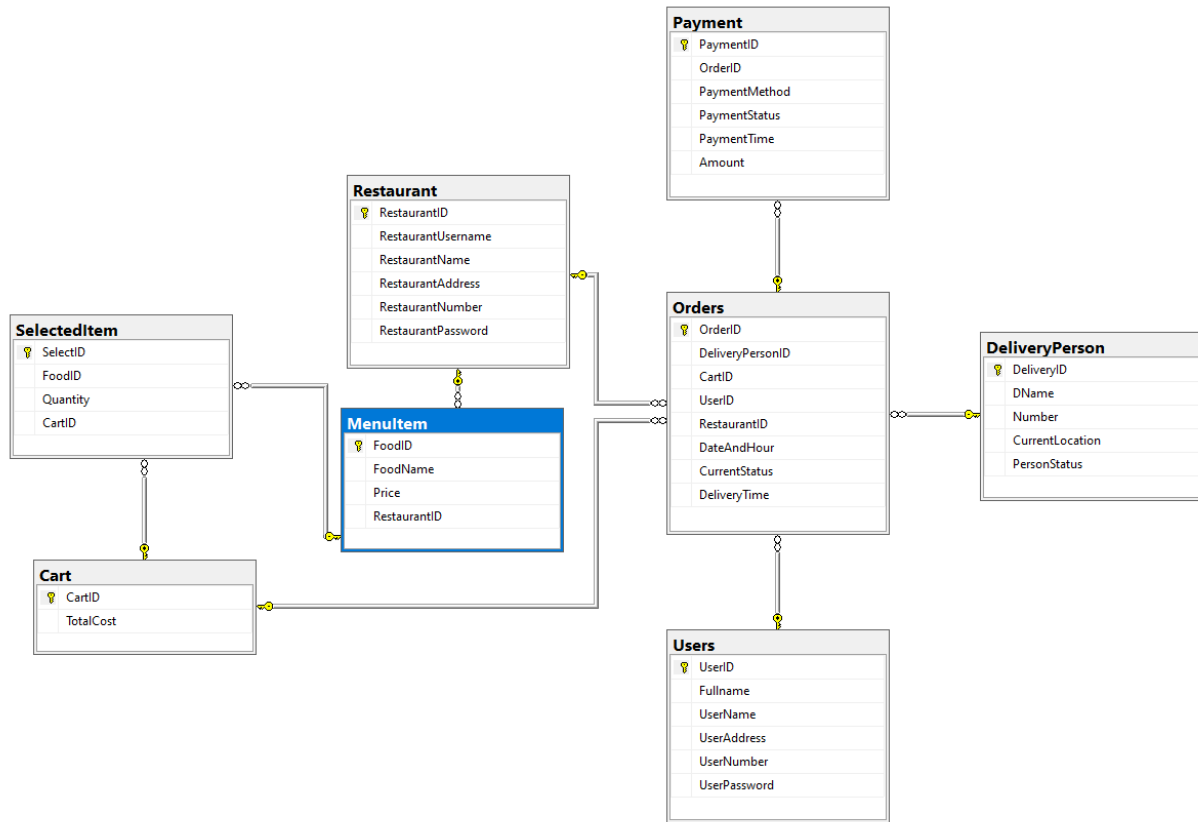


Figure 2. Schema design created by using SQL Server

For the design above, our database's highest normalization form is 3 NF. (geeksforgeeks, 2023)

Normal Form	Description
1 N.F	The database does not have any multivalued tuples.
2 N.F	All the non-key attributes depend on the primary key.
3 N.F	There are no transitive dependencies between non-key attributes.

Table 3. NF Form

2. Database Design

2.1 Database Creation

Table	Attributes
Users	UserID INT IDENTITY(1,1) PRIMARY KEY, Fullname VARCHAR(255), UserName VARCHAR(255) unique, UserAddress VARCHAR(255) NOT NULL, UserNumber VARCHAR(20) NOT NULL, UserPassword VARCHAR(255) NOT NULL
Restaurants	RestaurantID INT IDENTITY(1,1) PRIMARY KEY, RestaurantUsername VARCHAR(255) unique, RestaurantName VARCHAR(255) NOT NULL, RestaurantAddress VARCHAR(255) NOT NULL, RestaurantNumber VARCHAR(20) NOT NULL, RestaurantPassword VARCHAR(255) NOT NULL,
Payment	PaymentID INT IDENTITY(1,1) PRIMARY KEY, OrderID INT , PaymentMethod VARCHAR(20) NOT NULL, PaymentStatus VARCHAR(20) NOT NULL, PaymentTime VARCHAR(255) NOT NULL, Amount FLOAT NOT NULL
MenuItem	FoodID INT IDENTITY(1,1) PRIMARY KEY, FoodName VARCHAR(255) NOT NULL, Price FLOAT NOT NULL, RestaurantID INT NOT NULL
DeliveryPerson	DeliveryID INT IDENTITY(1,1) PRIMARY KEY, DName VARCHAR(255) NOT NULL, Number VARCHAR(20) NOT NULL, CurrentLocation VARCHAR(255) NOT NULL, PersonStatus VARCHAR(10) NOT NULL
SelectedItem	SelectID INT IDENTITY(1,1) PRIMARY KEY, FoodID INT NOT NULL, Quantity INT NOT NULL, CartID INT NOT NULL
Cart	CartID INT IDENTITY(1,1) PRIMARY KEY, TotalCost FLOAT);
Orders	OrderID INT IDENTITY(1,1) PRIMARY KEY, DeliveryPersonID INT ,

	CartID INT NOT NULL, UserID INT NOT NULL, RestaurantID INT , DateAndHour VARCHAR(255) , CurrentStatus VARCHAR(20) , DeliveryTime VARCHAR(255)
--	--

Table 4. Database attributes

2.2 Tables Creation

Users Table:

```
CREATE TABLE Users (
  UserID INT IDENTITY(1,1) PRIMARY KEY,
  Fullname VARCHAR(255),
  UserName VARCHAR(255) unique,
  UserAddress VARCHAR(255) NOT NULL,
  UserNumber VARCHAR(20) NOT NULL,
  UserPassword VARCHAR(255) NOT NULL
);
```

Restaurants Table:

```
CREATE TABLE Restaurant (
  RestaurantID INT IDENTITY(1,1) PRIMARY KEY ,
  RestaurantUsername VARCHAR(255) unique,
  RestaurantName VARCHAR(255) NOT NULL,
  RestaurantAddress VARCHAR(255) NOT NULL,
  RestaurantNumber VARCHAR(20) NOT NULL,
  RestaurantPassword VARCHAR(255) NOT NULL,
);
```

Payment Table:

```
CREATE TABLE Payment (
  PaymentID INT IDENTITY(1,1) PRIMARY KEY,
  OrderID INT ,
  PaymentMethod VARCHAR(20) NOT NULL,
  PaymentStatus VARCHAR(20) NOT NULL,
  PaymentTime VARCHAR(255) NOT NULL,
```

```
Amount FLOAT NOT NULL  
);
```

MenuItem Table:

```
CREATE TABLE MenuItem (  
FoodID INT IDENTITY(1,1) PRIMARY KEY,  
FoodName VARCHAR(255) NOT NULL,  
Price FLOAT NOT NULL,  
RestaurantID INT NOT NULL  
);
```

DeliveryPerson Table:

```
CREATE TABLE DeliveryPerson (  
DeliveryID INT IDENTITY(1,1) PRIMARY KEY,  
DName VARCHAR(255) NOT NULL,  
Number VARCHAR(20) NOT NULL,  
CurrentLocation VARCHAR(255) NOT NULL,  
PersonStatus VARCHAR(10) NOT NULL  
);
```

SelectedItem Table:

```
CREATE TABLE SelectedItem (  
SelectID INT IDENTITY(1,1) PRIMARY KEY,  
FoodID INT NOT NULL,  
Quantity INT NOT NULL,  
CartID INT NOT NULL  
);
```

Cart Table:

```
CREATE TABLE Cart (  
CartID INT IDENTITY(1,1) PRIMARY KEY,  
TotalCost FLOAT  
);
```

Orders Table:

```
CREATE TABLE Orders (  
  OrderID INT IDENTITY(1,1) PRIMARY KEY,  
  DeliveryPersonID INT ,  
  CartID INT NOT NULL,  
  UserID INT NOT NULL,  
  RestaurantID INT ,  
  DateAndHour VARCHAR(255) ,  
  CurrentStatus VARCHAR(20) ,  
  DeliveryTime VARCHAR(255)  
);
```

2.3 Sample data insertion

In general, the data inserted into the database is sample data that comes from self-generated data for testing purposes.

Users Table:

```
INSERT INTO Users (Fullname, UserName, UserAddress,  
  UserNumber, UserPassword)  
VALUES  
(  
'John Doe', 'johndoe', '123 Main St, Townsville', '1234567890', '  
ef92b778bafef771e89245b89ecbc08a44a4e166c06659911881f383d  
4473e94f',  
'Jane Smith', 'janesmith', '789 Park Ave, Townsville',  
'0987654321',  
'89e01536ac207279409d4de1e5253e01f4a1769e696db0d6062ca9b  
8f56767c8'  
);
```

Restaurants Table:

```
INSERT INTO Restaurant (RestaurantUsername, RestaurantName,  
  RestaurantAddress, RestaurantNumber, RestaurantPassword)  
VALUES  
(  
'burgerplace', 'Burger Place', '456 Side St, Townsville',  
'0987654321', '  
e0e6097a6f8af07daf5fc7244336ba37133713a8fc7345c36d667dfa5  
13fabaa',  
'
```

```
('pizzacorner', 'Pizza Corner', '321 Hill Rd, Townsville',  
'1230987654',  
'934d553924cd730793934748884a2b8efc42c2b76bb20a1ae7a8958  
583aa0d19');
```

Payment:

```
INSERT INTO Payment (OrderID, PaymentMethod,  
PaymentStatus, PaymentTime, Amount)  
VALUES  
(1, 'Credit Card', 'Completed', '2023-12-05 11:50:00', 16.00),  
(2, 'PayPal', 'Pending', '2023-12-06 17:50:00', 20.00);
```

MenuItem Table:

```
INSERT INTO MenuItem (FoodName, Price, RestaurantID)  
VALUES  
( 'Cheeseburger', 8, 1),  
( 'Veggie Pizza', 10, 2);
```

DeliveryPerson Table:

```
INSERT INTO DeliveryPerson (DName, Number,  
CurrentLocation, PersonStatus)  
VALUES  
( 'Alice Smith', '9876543210', 'Central Hub, Townsville',  
'Available'),  
( 'Bob Johnson', '8765432109', 'North Hub, Townsville',  
'Available');
```

SelectedItem Table:

```
INSERT INTO SelectedItem (FoodID, Quantity, CartID)  
VALUES  
(1, 2, 1),  
(2, 2, 2);
```

Cart Table:

```
INSERT INTO Cart (TotalCost)
VALUES
(16.00),
(20.00);
```

Orders Table:

```
INSERT INTO Orders (DeliveryPersonID, CartID, UserID,
RestaurantID, DateAndHour, CurrentStatus, DeliveryTime)
VALUES
(1, 1, 1, 1, '2023-12-05 12:00:00', 'Preparing', '2023-12-05
13:00:00'),
(2, 2, 2, 2, '2023-12-06 18:00:00', 'Delivered', '2023-12-06
19:00:00');
```

3. SQL Usage

3.1 Login

- User:

```
SELECT UserID FROM Users WHERE UserName = ? AND UserPassword = ?
```

- Restaurant:

```
SELECT RestaurantID FROM Restaurant WHERE RestaurantUserName = ? AND
RestaurantPassword = ?
```

3.2 Sign-up

- User:

```
INSERT INTO Users (UserName, Fullname, UserAddress, UserNumber, UserPassword)
VALUES (''+username+ ''',''+fullname+ ''',''+address+ ''',''+number+ ''',''+password+ '')
```

- Restaurant:


```
INSERT INTO Restaurant (RestaurantUserName, RestaurantName, RestaurantAddress,
RestaurantNumber, RestaurantPassword) VALUES ('" + Rusername+ "', '" + Rname+ "', '"
+ Raddress+ "', '" + Rnumber+ "', '" + Rpassword+ "')
```

3.3 Password Reset

- User:

```
UPDATE Users SET UserPassword = 'hashedPassword '
WHERE UserName = 'username '
```

- Restaurant:

```
UPDATE Restaurant SET RestaurantPassword = 'hashedPassword '
WHERE RestaurantUsername = 'username'
```

3.4 Restaurant Side

3.4.1 Listing Food and Price

The following query is used for the owner of the restaurant to show their menu

```
SELECT M.FoodName, R.RestaurantName, M.Price
FROM MenuItem M, Restaurant R
ON M.RestaurantID = R.RestaurantID
WHERE R.RestaurantID = ?
```

The following query is used to for the restaurant to add new food into their menu.

```
INSERT INTO MenuItem (FoodName, Price, RestaurantID) VALUES (?, ?, ?);
```

3.4.2 Assigning Delivery Person

```
SELECT OrderID
```

```
FROM Orders
WHERE CurrentStatus = 'Pending' AND RestaurantID = 'SessionData.getInstance().getId()' ;
```

This query shows all the delivery person that are available to be assigned.

```
SELECT DName, DeliveryID
FROM DeliveryPerson
WHERE PersonStatus = 'Available'
```

This query assigns the delivery person to the chosen order.

```
UPDATE Orders
SET DeliveryPersonID = ? , DeliveryTime = ? , CurrentStatus = 'Preparing'
WHERE OrderID = ?;
```

This query set delivery person's status to delivering state.

```
UPDATE Deliveryrson
SET PersonStatus = 'Shipping'
WHERE DeliveryID = ?;
```

3.5 User Side

3.5.1 Searching Food

The following query is used to fetch information of the food items from the MenuItem tables that are related to the food name that the user searched for. It shows the food name and corresponding price.

```
SELECT M.FoodName, R.RestaurantName, M.Price
FROM MenuItem M
JOIN Restaurant R ON M.RestaurantID = R.RestaurantID
WHERE M.FoodName LIKE ?;
```

3.5.2 Viewing Restaurants

```
SELECT RestaurantID, RestaurantName FROM Restaurant
```

3.5.3 Viewing and Adding Food to Cart

```
SELECT FoodName, Price FROM MenuItem WHERE RestaurantID = ?
```

3.5.4 Delivery Information Confirmation

The function of the following query is to get delivery information of recipient. The collected information will be displayed on screen for user's confirmation before placing order.

```
SELECT FullName, UserAddress, UserNumber FROM Users WHERE UserID = ?;
```

3.5.5 Payment of User's Order:

The following query's function is to insert Payment information into Database.

```
INSERT INTO Payment(OrderID, PaymentMethod, PaymentStatus, PaymentTime, Amount)
VALUES (?, ?, 'Completed', ?, ?);
```

3.5.6 Inserting User's Selected Items

The function of the following query is to insert user's selected food to SelectedItem table.

```
INSERT INTO SelectedItem (FoodID, Quantity, CartID) VALUES ((SELECT FoodID FROM
MenuItem WHERE FoodName = ?), ?, ?);
```

3.5.7 Making orders

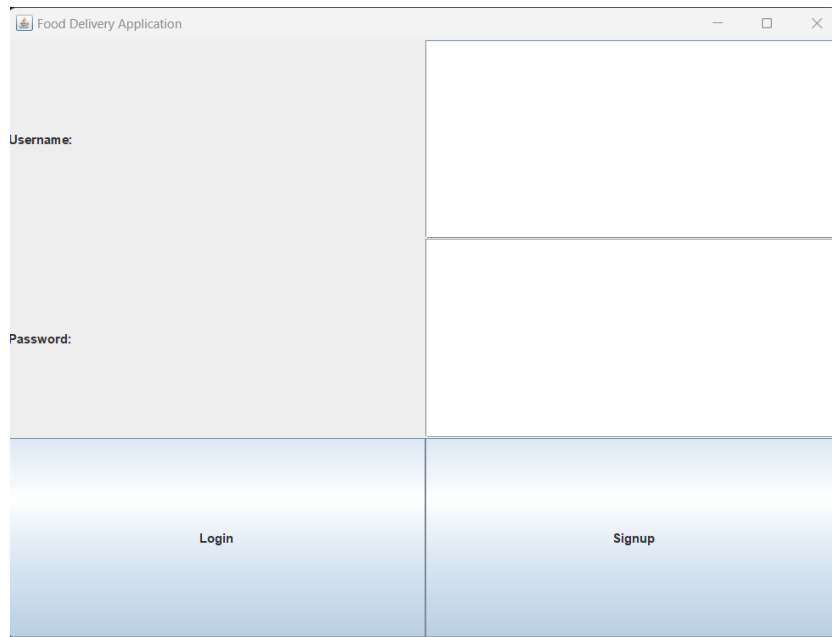
This query inserts order items into the "SelectedItem" table. It iterates through the itemQuantities map, which contains the quantities of various food items selected by the user. For each item, it gets the FoodID using a separate query and inserts a new record into the "SelectedItem" table.

```
INSERT INTO Orders(CartID, UserID, RestaurantID, DateAndHour, CurrentStatus)\n" +  
    "VALUES (?, ?, ?, ?, 'Pending')
```

Chapter 4: Features and Functionalities

1. User Management

1.1 Login

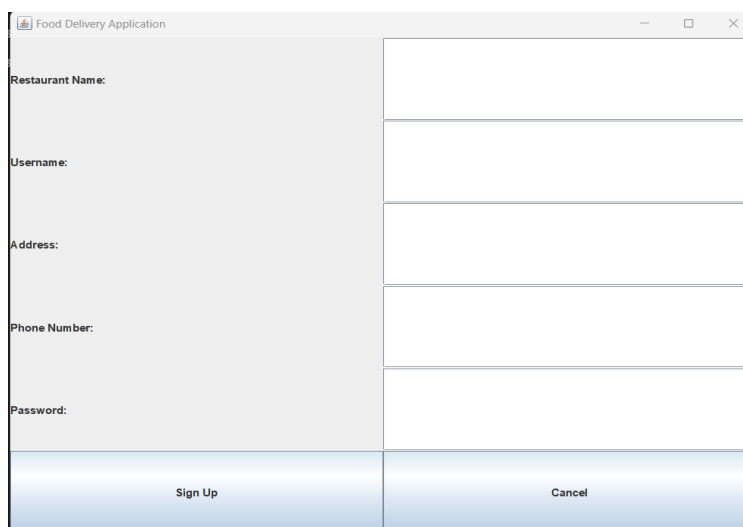


The screenshot shows a window titled "Food Delivery Application". The login panel is divided into two main sections. The left section, with a light gray background, contains labels for "Username:" and "Password:". The right section has two white input fields corresponding to these labels. At the bottom, there are two blue buttons: "Login" on the left and "Signup" on the right.

Figure 3. Login panel

We will take the user's input, then use SELECT query (3.3.1) to authenticate the user.

1.2 Sign-up



The screenshot shows a window titled "Food Delivery Application". The sign-up panel is divided into two main sections. The left section, with a light gray background, contains labels for "Restaurant Name:", "Username:", "Address:", "Phone Number:", and "Password:". The right section has five white input fields corresponding to these labels. At the bottom, there are two blue buttons: "Sign Up" on the left and "Cancel" on the right.

Figure 4. Sign-up panel

The program will take the user's input then use INSERT query (3.3.2) to register a new account.

1.3 Password Reset

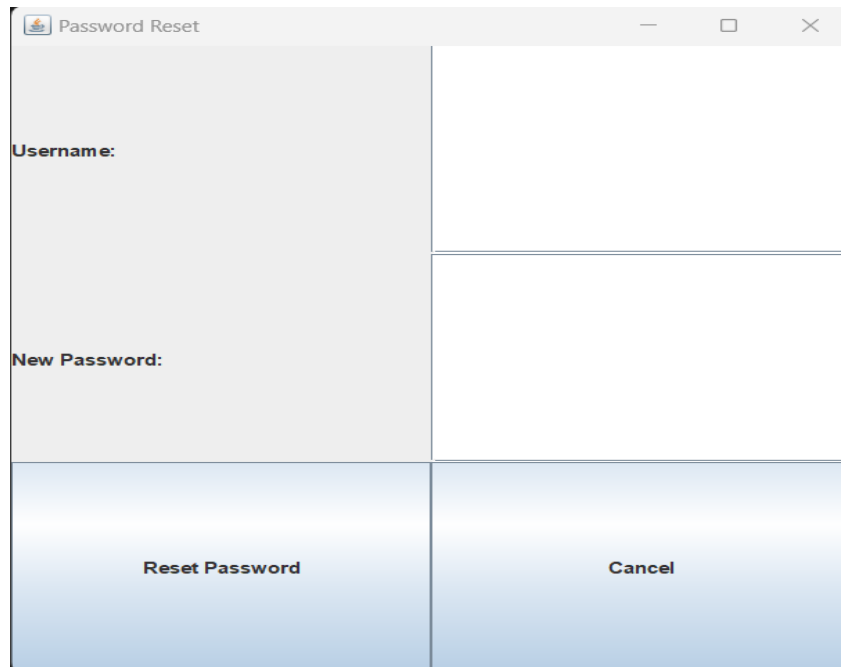
A screenshot of a 'Password Reset' dialog box. The window has a title bar with a small icon and the text 'Password Reset'. The main area is divided into three sections. The top section is a light gray rectangle containing the label 'Username:'. The middle section is a white rectangle containing the label 'New Password:'. The bottom section is a blue gradient rectangle divided into two buttons: 'Reset Password' on the left and 'Cancel' on the right.

Figure 5. Password reset panel.

The program will take username and new password, then use UPDATE query (3.3.3) to update the new password referred to the username.

2. Functionalities

2.1 Restaurant Side

2.1.1 Listing Food and Price

The program will search for and display details of specific menu items within a restaurant by providing the restaurantID as the restaurant owner has resgistered.

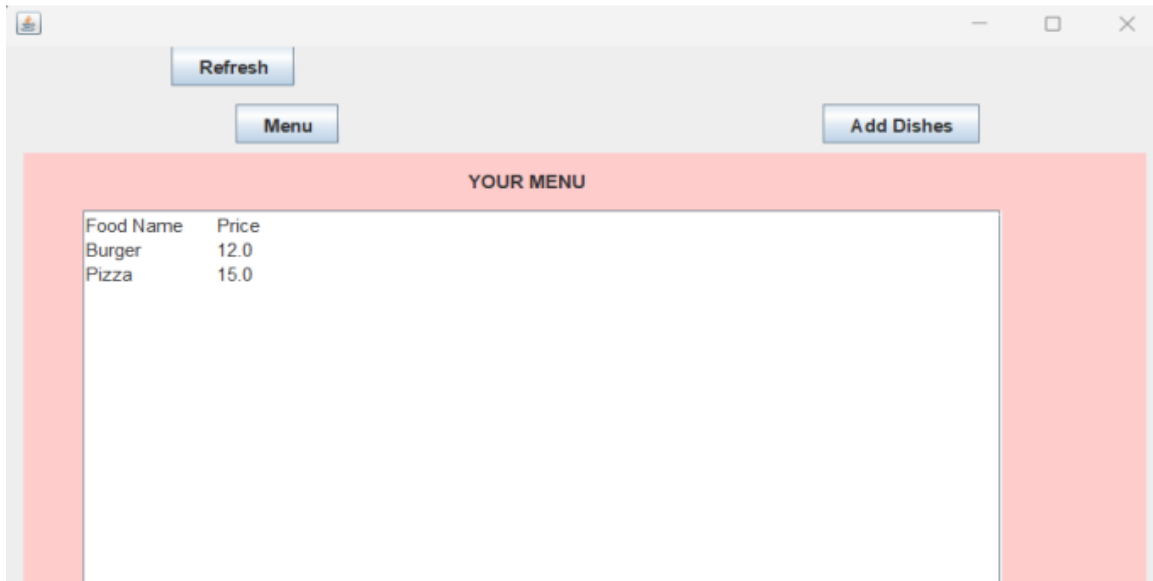


Figure 6. Listing Food and Price

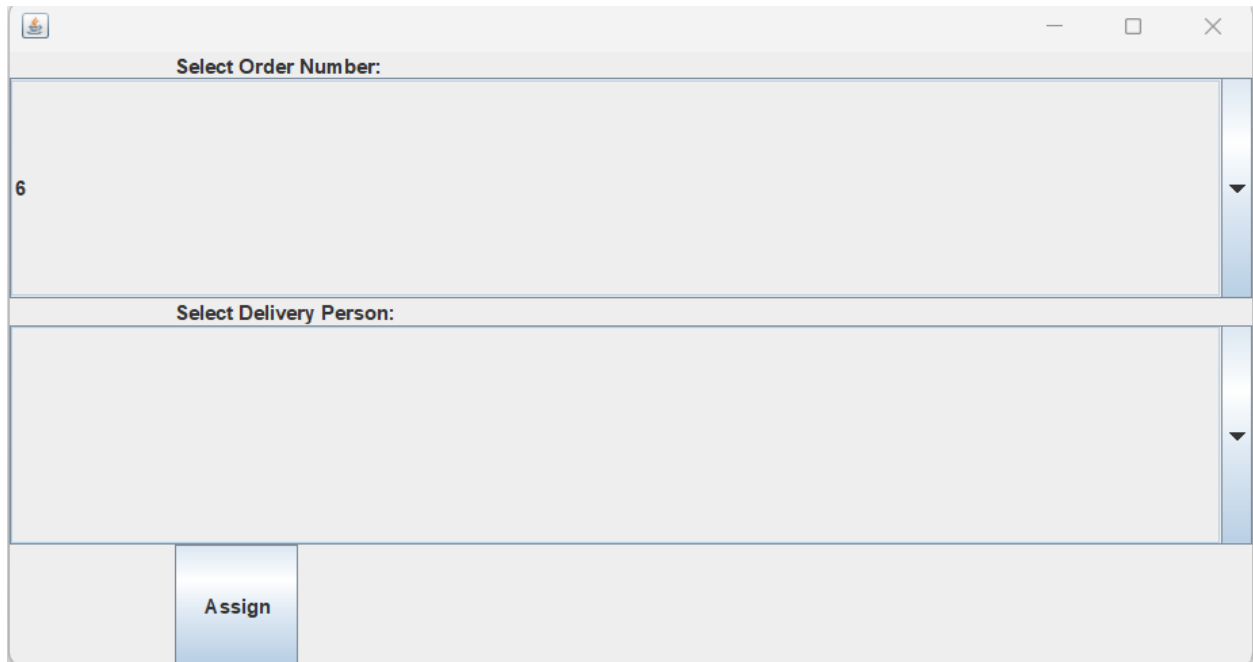
2.1.2 Add Food

The feature is designed to add additional dishes to the restaurant's menu. By providing the food name and price to the restaurantId as same for the previous one, the menu will be updated after clicking refresh button.

A screenshot of a web application window. At the top, there are three buttons: 'Refresh', 'Menu', and 'Add Dishes'. Below these buttons is a cyan-bordered box containing the text 'RESTAURANT'S NEW FOOD'. Inside this box is a form with two input fields: 'Food Name:' and 'Price:'. Below the input fields is an 'ADD' button.

Figure 7. Add food to menu.

2.1.3 Assigning Delivery Person



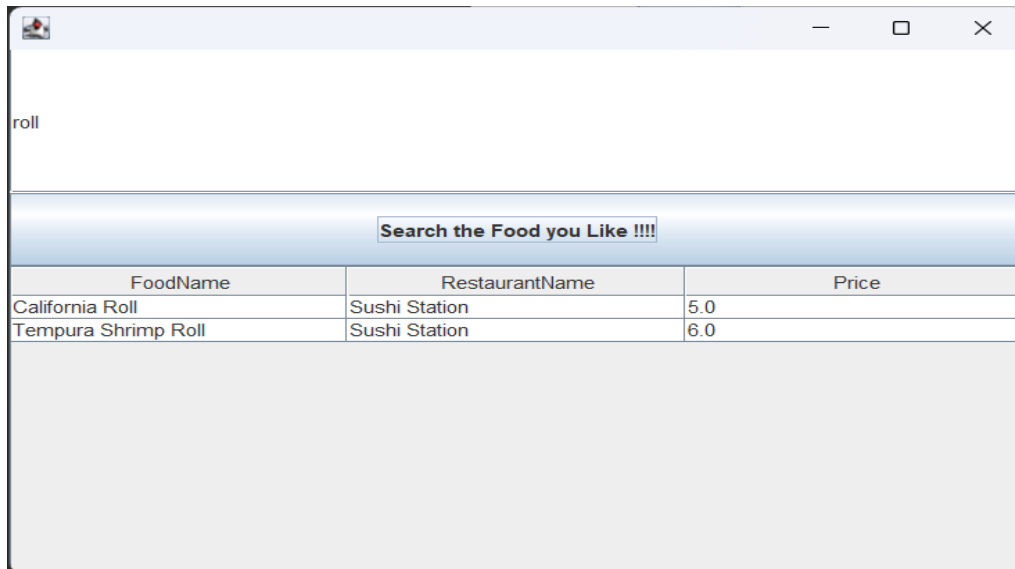
The image shows a software window with a title bar containing a small icon and standard window controls (minimize, maximize, close). The window is divided into three main sections. The top section is labeled 'Select Order Number:' and contains a list box with the number '6' visible. The middle section is labeled 'Select Delivery Person:' and contains an empty list box. The bottom section contains a single button labeled 'Assign'.

Figure 8. Assign Delivery Person to Order Panel

The program will a list of orders that need a delivery person and a list of delivery person that are available. After pressing the Assign button, the system will change the status of the delivery person into “Unavailable” and the order into “Shipping”.

2.2 User Side

2.2.1 Searching Food



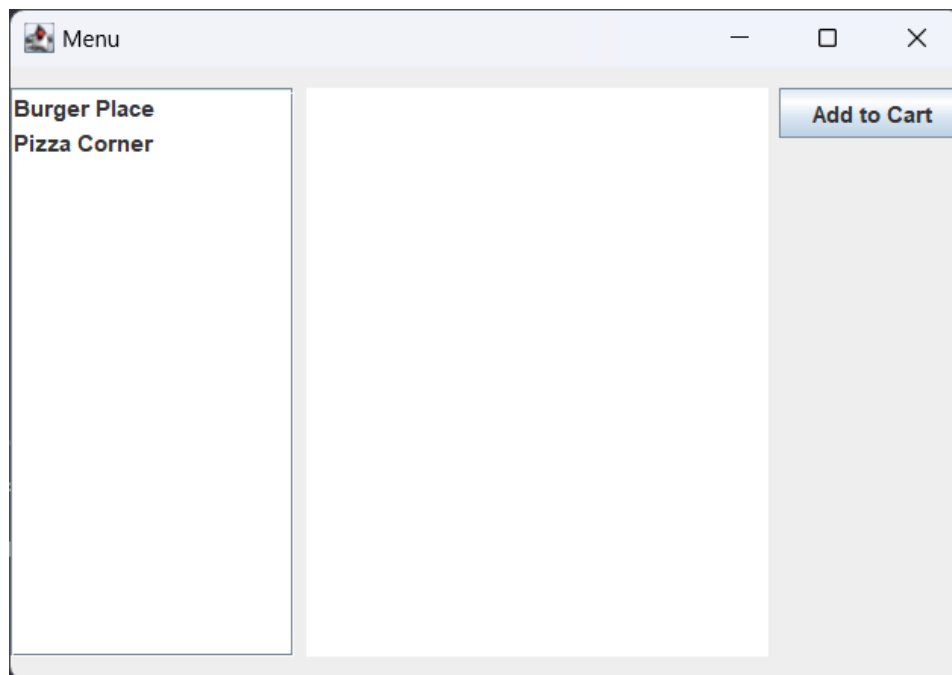
roll

Search the Food you Like !!!!

FoodName	RestaurantName	Price
California Roll	Sushi Station	5.0
Tempura Shrimp Roll	Sushi Station	6.0

Figure 9. User search for Food

2.2.2 Viewing Restaurants



Menu

Burger Place
Pizza Corner

Add to Cart

Figure 10. Users view the Restaurant list.

Upon initialization, the application shows the restaurant lists.

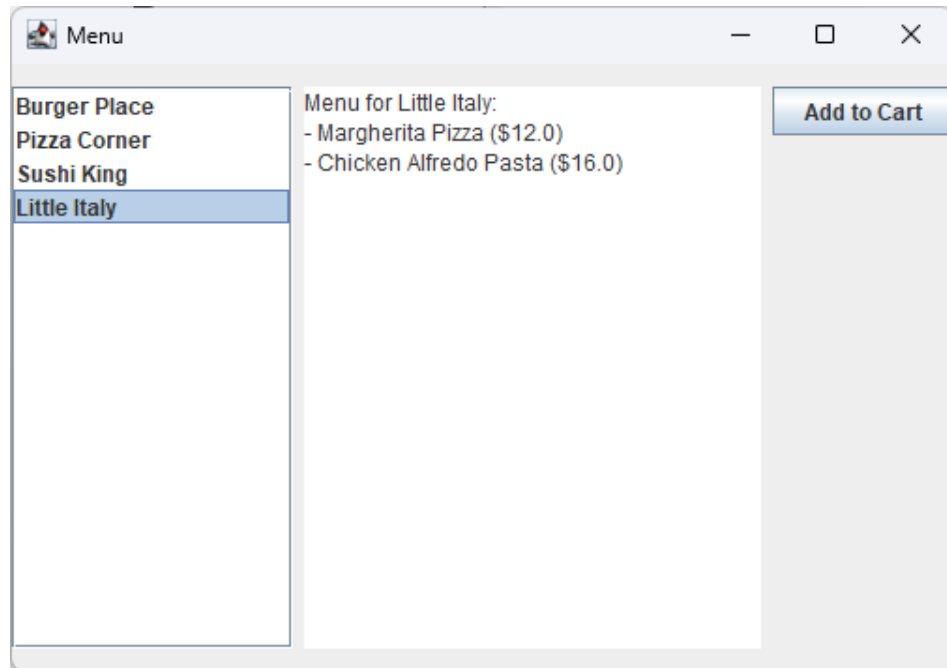


Figure 11. User view Restaurant and Menu

When a restaurant is selected, the application executes a SQL code and then shows us the menu and the price of the foods.

2.2.3 Adding Food to Cart and View Cart

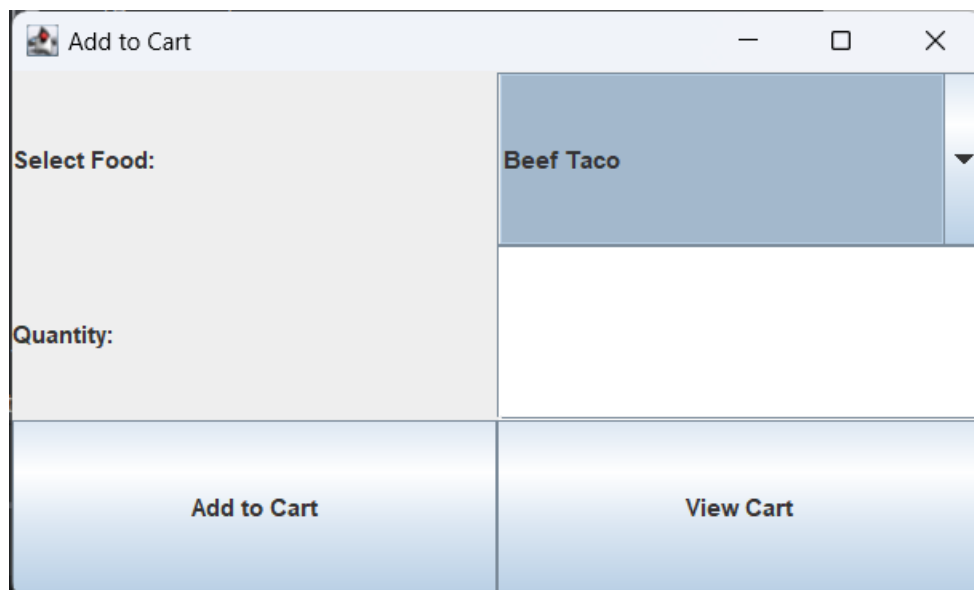


Figure 12. Add to Cart Panel

After pressing the add to cart button, we can see the select food drop down menu that show us all the food the restaurant offers and we can select them.

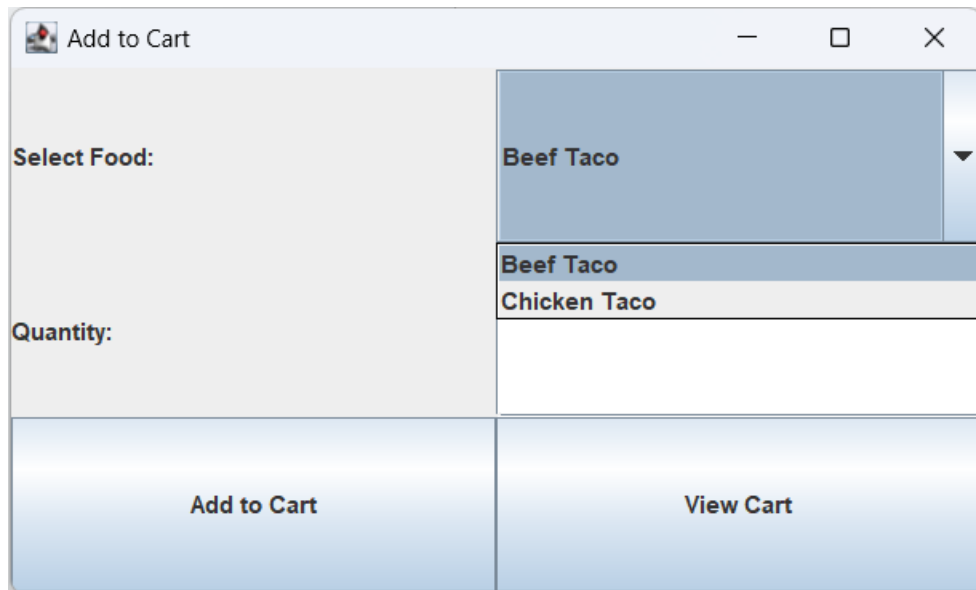


Figure 13. Add to Cart Panel

Drop down menu that show us all the food.

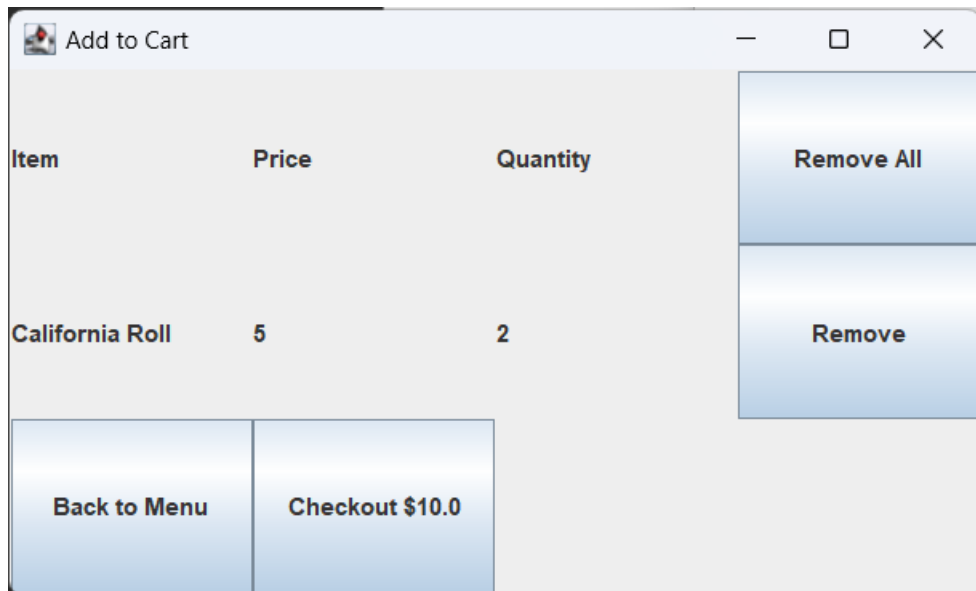


Figure 14. Cart Panel

After adding food to Cart and press 'View Cart', the Cart panel appears.

2.2.4 Removing Food from Cart

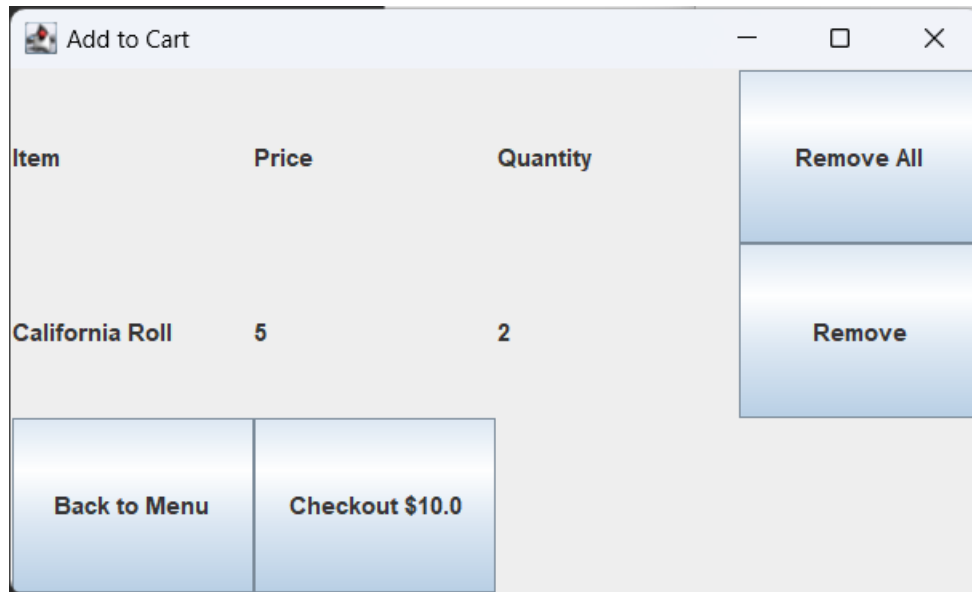
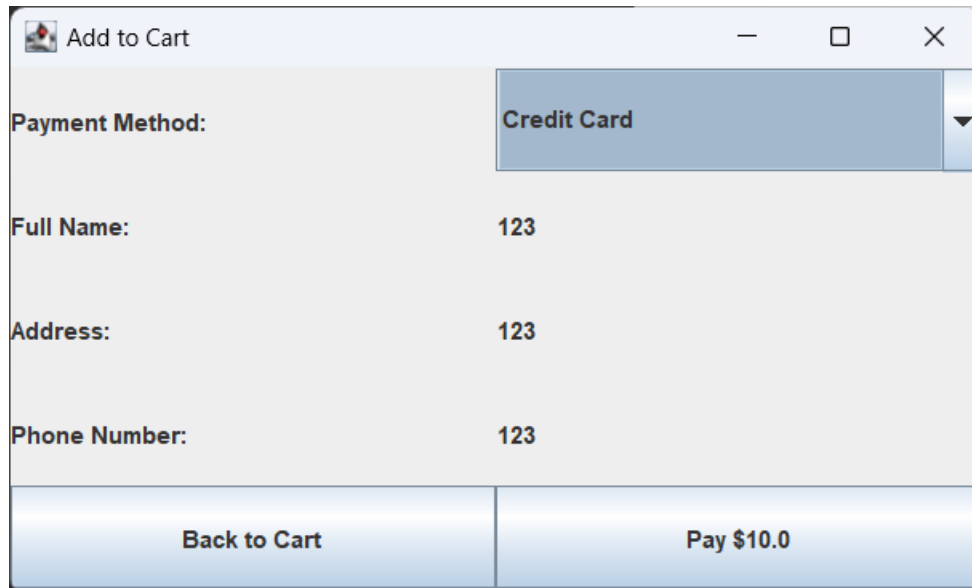


Figure 15. Removing Food from Cart Panel

In Cart Panel, first, users can view selected foods, including names, quantities, and prices of a whole order. Second, they can remove items. For instance, if a customer presses the remove button, one item will be removed from their cart no matter their quantity; or if they press remove all, all the items will be immediately discarded.

To add more items or modify items' quantities, customers can press 'Back to menu' and view 'Menu frame' to select more items. After choosing all items they wanted to order, customers can press 'Checkout', with the total price of the order on it.

2.2.5 Payment

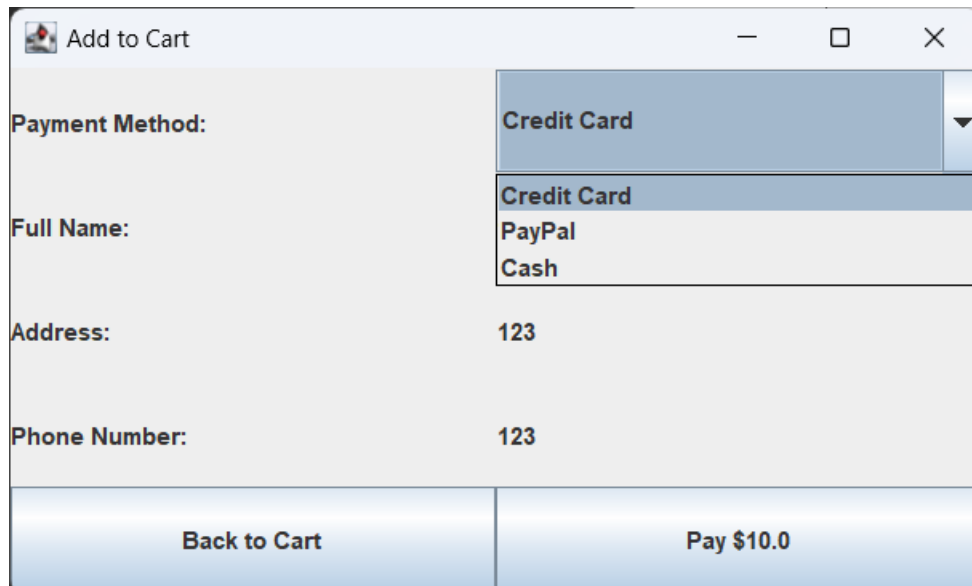


The 'Add to Cart' window displays a payment form. It includes a 'Payment Method' dropdown menu set to 'Credit Card'. Below this are three text input fields: 'Full Name', 'Address', and 'Phone Number', each containing the placeholder text '123'. At the bottom, there are two buttons: 'Back to Cart' and 'Pay \$10.0'.

Payment Method:	Credit Card
Full Name:	123
Address:	123
Phone Number:	123
<div>Back to CartPay \$10.0</div>	

Figure 16. Payment Panel

After pressing 'Checkout', the payment frame will appear. Customer's information that is selected from the 'Users' table, including name, phone number, and address, is shown in this frame to let them check their delivery information before assigning the order to a delivery man. If customers want to check their cart, they can get back to 'Cart' frame by clicking 'Back to cart', and to complete the order, they can press the 'Pay' button.



The 'Add to Cart' window shows the 'Payment Method' dropdown menu open, displaying three options: 'Credit Card', 'PayPal', and 'Cash'. The other fields ('Full Name', 'Address', 'Phone Number') and the bottom buttons ('Back to Cart', 'Pay \$10.0') remain the same as in Figure 16.

Payment Method:	Credit Card PayPal Cash
Full Name:	123
Address:	123
Phone Number:	123
<div>Back to CartPay \$10.0</div>	

Figure 17. Choose payment options

Our application offers various types of payment so that customers will be able to choose their payment method such as cash, PayPal or credit card in a combo box

Chapter 5: Security Measures

1. Data Protection:

```
public class PasswordHash {  
    6 usages  
    public String passwordHash(String password) throws NoSuchAlgorithmException{  
  
        MessageDigest md = MessageDigest.getInstance( algorithm: "SHA-256");  
        byte[] messageDigest = md.digest(password.getBytes());  
        BigInteger bigInt =new BigInteger( signum: 1,messageDigest);  
  
        return bigInt.toString( radix: 16);  
    }  
}
```

Figure 18. Password Hash

We use SHA 256 algorithm to encrypt the password so that the password is secure to be stored in the database.

2. SQL Injection Prevention:

A prepared statement is a database query that is precompiled and stored in a form that can be reused with different parameter values. Prepared statements separate SQL code from user-provided data, eliminating the risk of SQL injection attacks. When using prepared statements, input values are treated as data rather than executable SQL code.

```

private boolean authenticateUser(String username, String password) {
    try {
        ConnectDB newConn = new ConnectDB();
        String sql = "SELECT UserID FROM Users WHERE UserName = ? AND UserPassword = ?";
        newConn.preparedStmt = newConn.getConn().prepareStatement(sql);
        newConn.preparedStmt.setString( parameterIndex: 1, username);
        newConn.preparedStmt.setString( parameterIndex: 2, password);

        ResultSet rs = newConn.preparedStmt.executeQuery();
        if (rs.next()) {
            int userID = rs.getInt( columnLabel: "UserID");
            SessionData.getInstance().setId(userID);
            System.out.println("User ID set: " + userID);
            return true;
        }
        newConn.closeConnection();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return false;
}

```

Figure 19. Prepared Statement Implementation

Chapter 6: Conclusions

1. Achieved Goal

In summary, our team has successfully achieved the goal of creating a food delivery application that connects customers with restaurant owners. This app features functionalities such as easy account creation and login for both customers and restaurant owners. Customers have the convenience of searching for restaurants, browsing through menus, and adding their preferred meals to the cart before placing an order. Meanwhile, restaurant owners can list their dishes and menu, enhancing their visibility and operational efficiency.

We have put in place strong security measures to safeguard user information and prevent unauthorized access, SQL injection, making our platform both safe and reliable. In essence, this project has led to the development of a highly effective and secure food delivery system that excels in ease of use, functionality, and safety.

2. Future Work

- **UI Improvement:** The user interface (UI) is the first point of interaction for our users. It's vital to ensure that it is intuitive, responsive, and aesthetically pleasing. We plan to revamp the UI to make it more user-friendly and engaging, with an emphasis on ease of navigation and visual appeal.
- **Real-Time Delivery Tracking:** Integrating an API for real-time tracking of the delivery person's location is a key enhancement. This feature will provide users with the ability to see the exact location of their order in real-time, enhancing transparency and trust in our service.
- **Estimated Time of Arrival (ETA) Calculation:** Implementing a system to accurately calculate and display the ETA of orders will significantly improve user satisfaction. This feature will consider several factors such as distance, traffic conditions, and preparation time to provide users with a reliable estimate of when their order will arrive.
- **Advanced Food Sorting Functionality:** To streamline the process of finding and ordering food, we propose the development of an advanced sorting feature. This will include categorization of menu items into groups such as main dishes, side dishes, desserts, and extra items. Such a feature will simplify the user's decision-making process and enhance the overall ordering experience.

3. Conclusion

In conclusion, we are excited about the prospects of our food delivery application. Our commitment is to continuously improve and make the app more user-friendly and efficient. We are grateful for the invaluable guidance received from our mentors, which has been instrumental in the success of this project. As we move forward, we look forward to the positive impact our application will have in the food service industry and the enhanced experience it will offer to customers and restaurant owners alike.

Chapter 7: References

- GeeksforGeeks. (2022, April 29). *SHA 256 hash in Java*.
<https://www.geeksforgeeks.org/sha-256-hash-in-java/>
- *Servlet login with prepared statement*. (n.d.). Stack Overflow.
<https://stackoverflow.com/questions/27356075/servlet-login-with-prepared-statement>
- GeeksforGeeks. (2023, November 6). *Normal forms in DBMS*.
https://www.geeksforgeeks.org/normal-forms-in-dbms/?ref=header_search
- *SQL Syntax*. (n.d.).
https://www.w3schools.com/sql/sql_syntax.asp
- What is Entity Relationship Diagram (ERD)? (n.d.).
<https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/>