

**Talha Ahmad 22i-0760**

**Omer Mustafa 22i-1180**

**Ibrahim Shahid 22i-0873**

**CS-F**

**Final Project Documentation**

**App Name: Food Ninja**

Github repo: [https://github.com/randomefool/Food\\_Ninja](https://github.com/randomefool/Food_Ninja)

(master branch)

Github documentation repo:

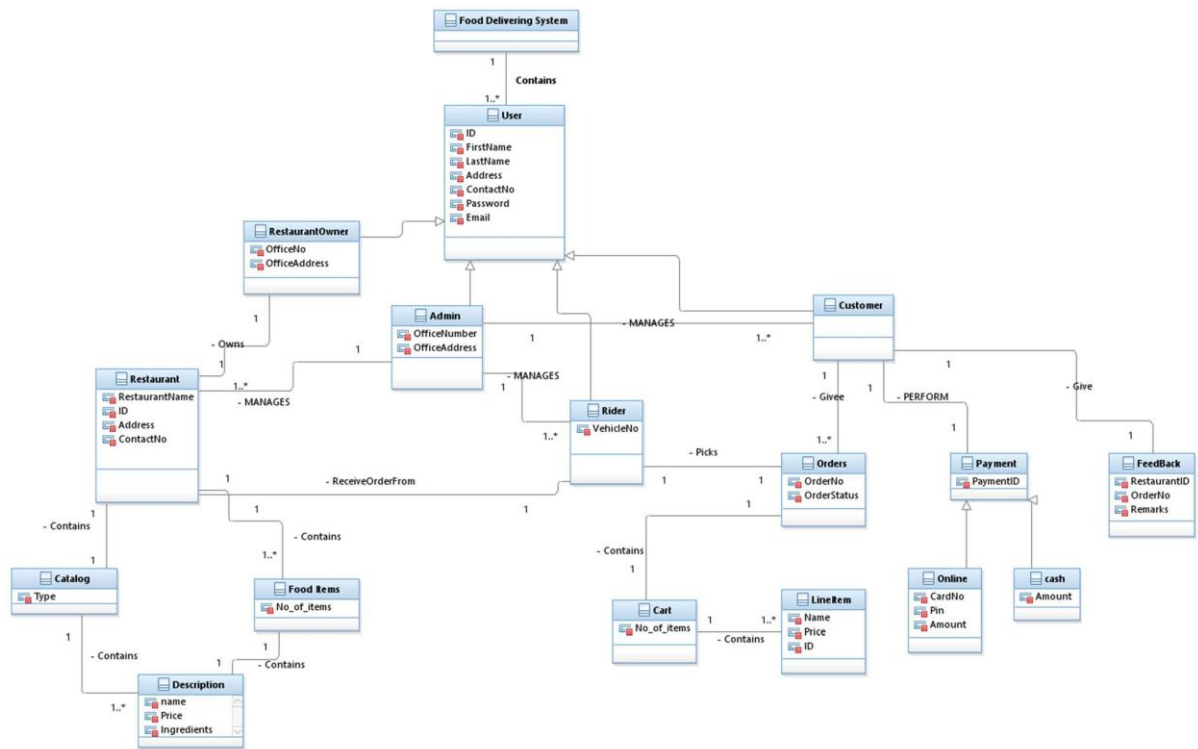
[https://github.com/randomefool/Food\\_Ninja/blob/main/i220760\\_i221180\\_i220873\\_final\\_project\\_documentation.pdf](https://github.com/randomefool/Food_Ninja/blob/main/i220760_i221180_i220873_final_project_documentation.pdf)

Github presentation repo:

[https://github.com/randomefool/Food\\_Ninja/blob/main/i220760\\_i221180\\_i220873\\_Presentation.pptx](https://github.com/randomefool/Food_Ninja/blob/main/i220760_i221180_i220873_Presentation.pptx)

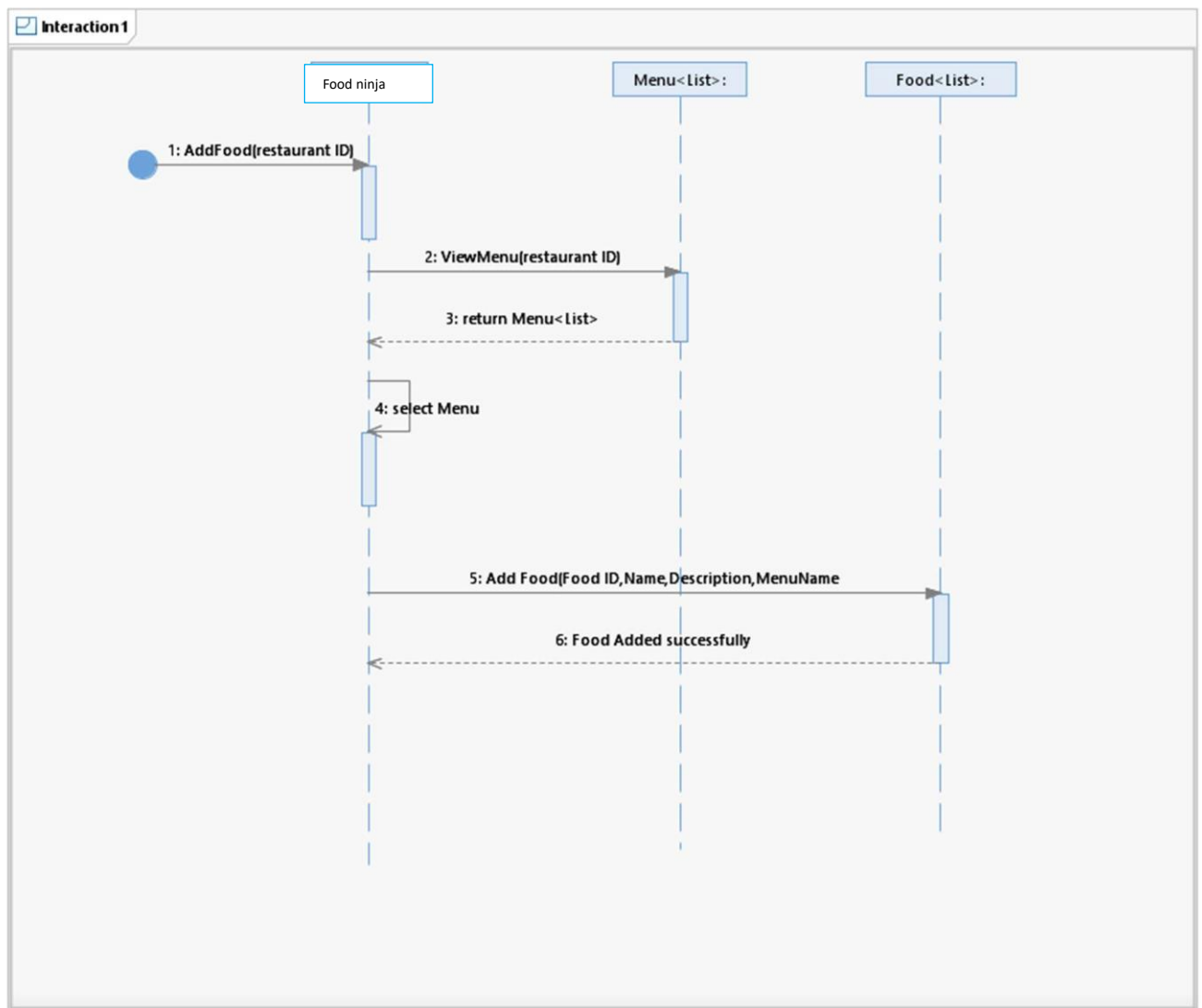
# **Domain diagram**



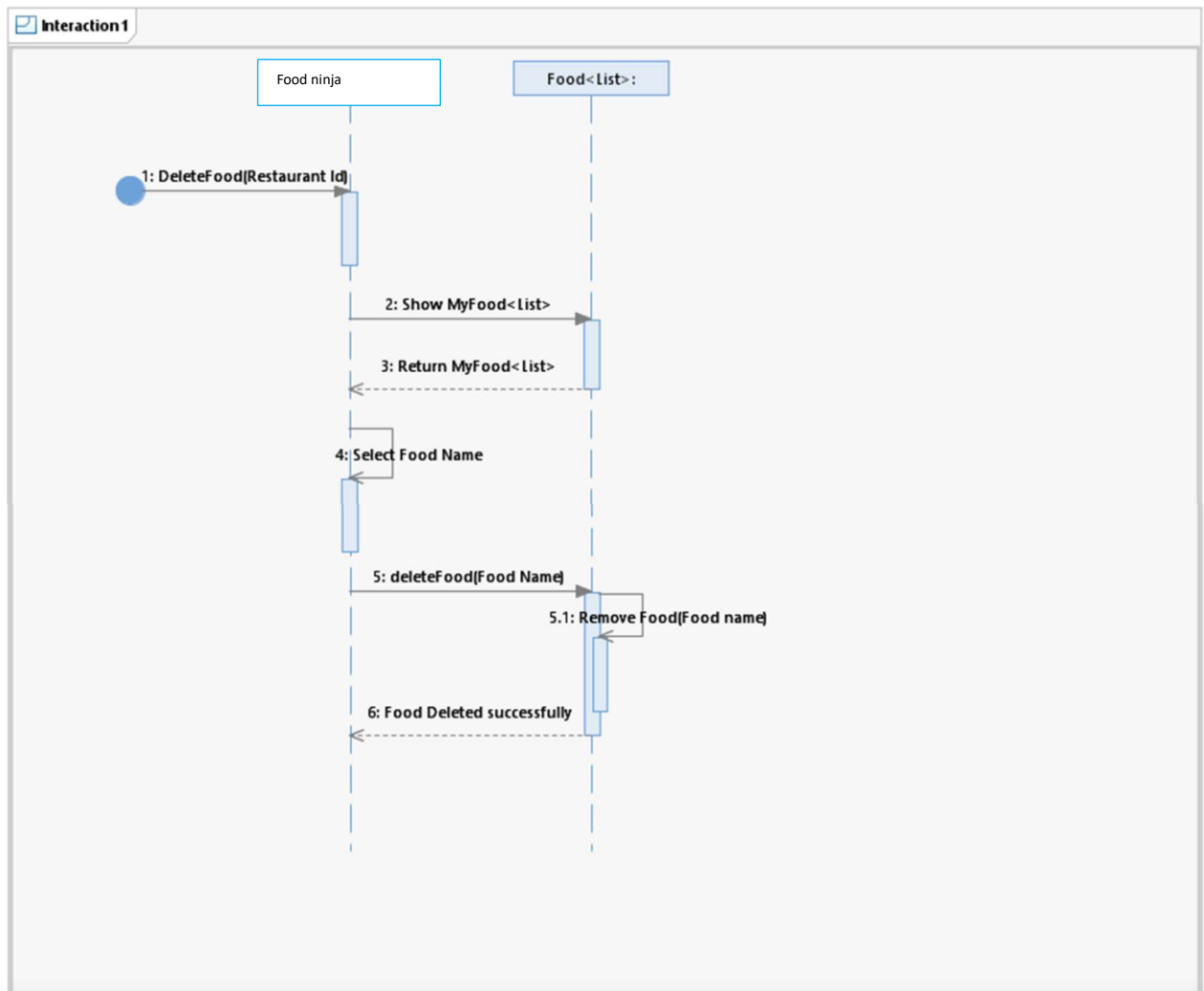


# Sequence diagram

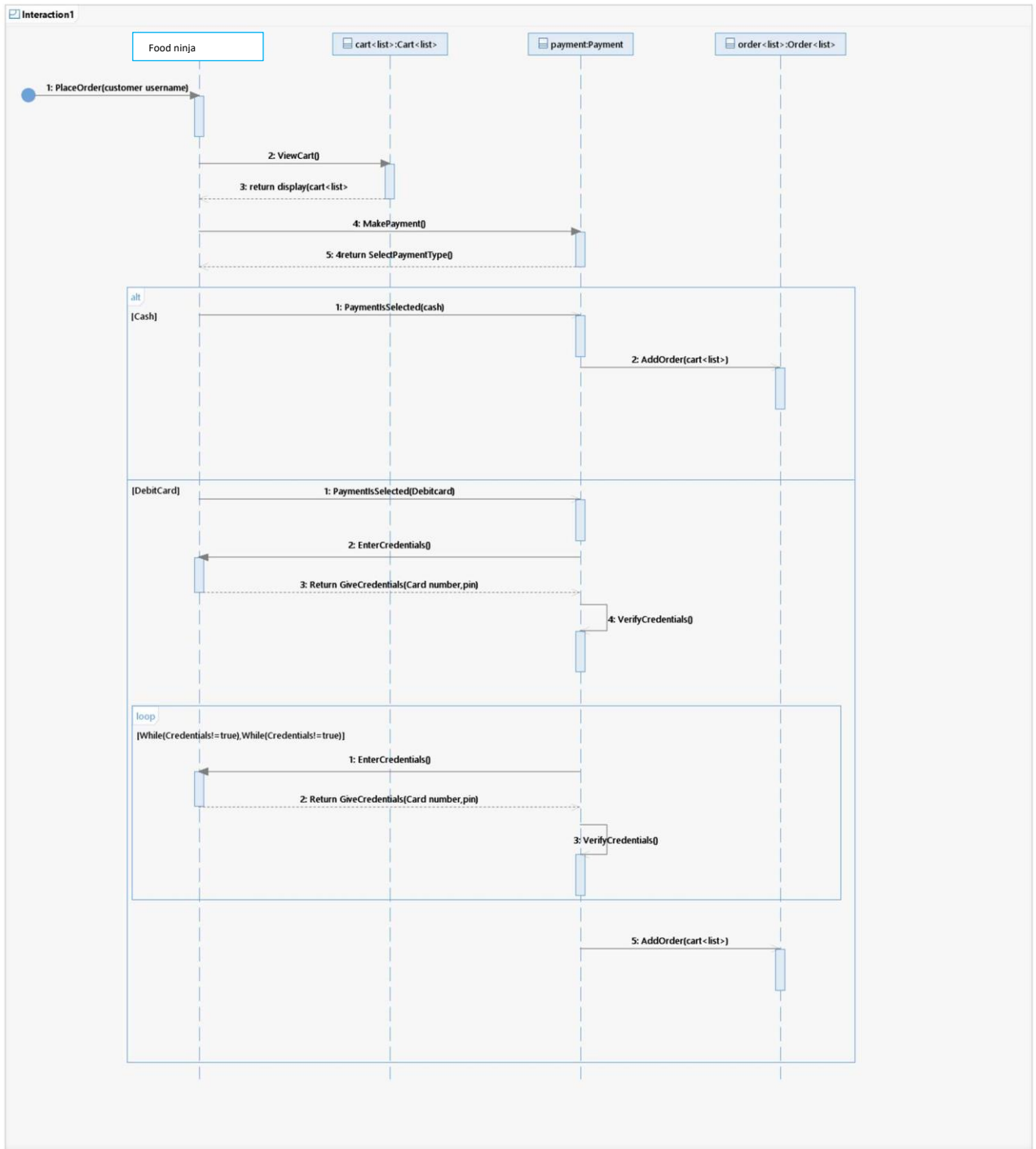
**Add Food**



## Delete Food

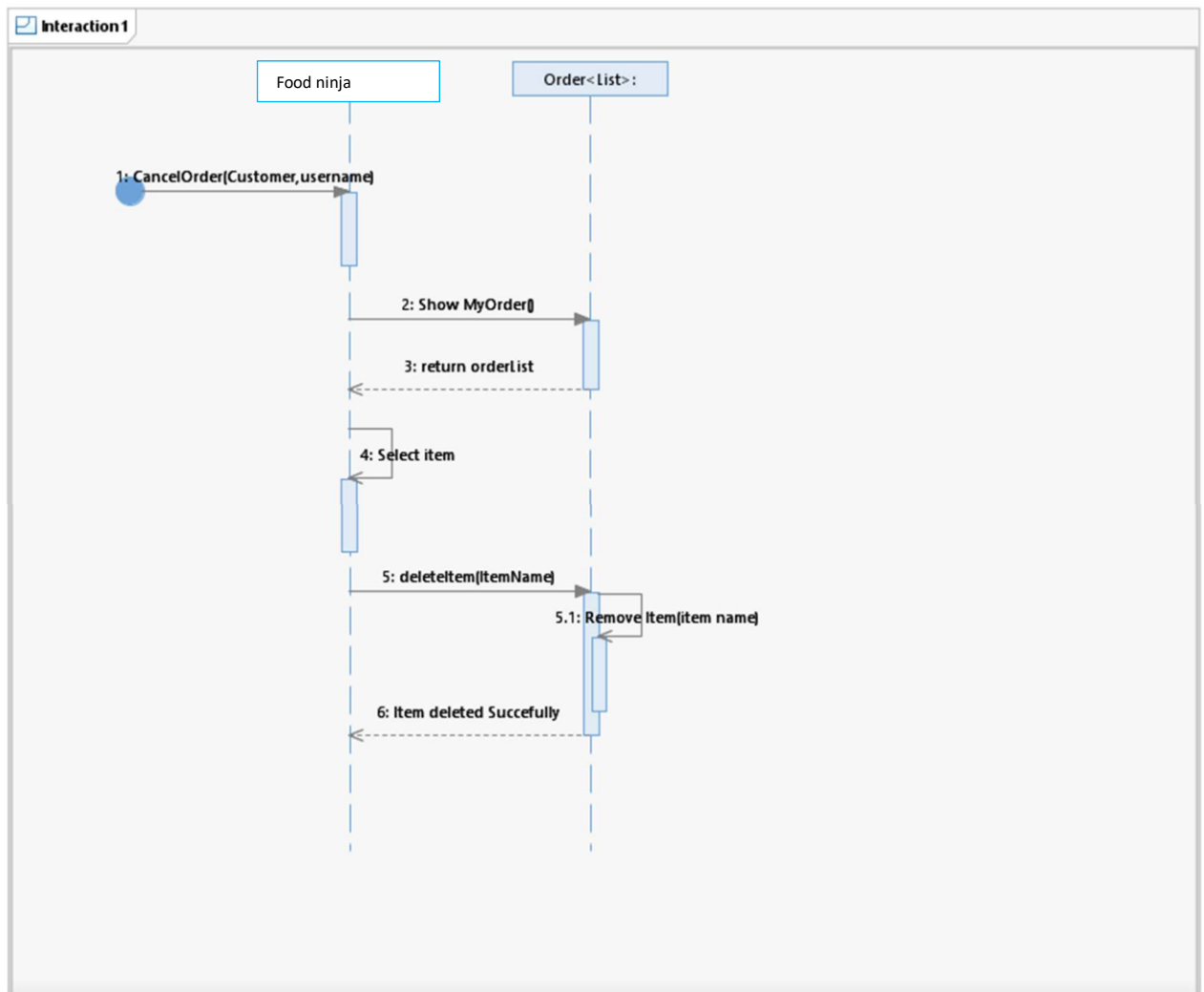


# Place Order

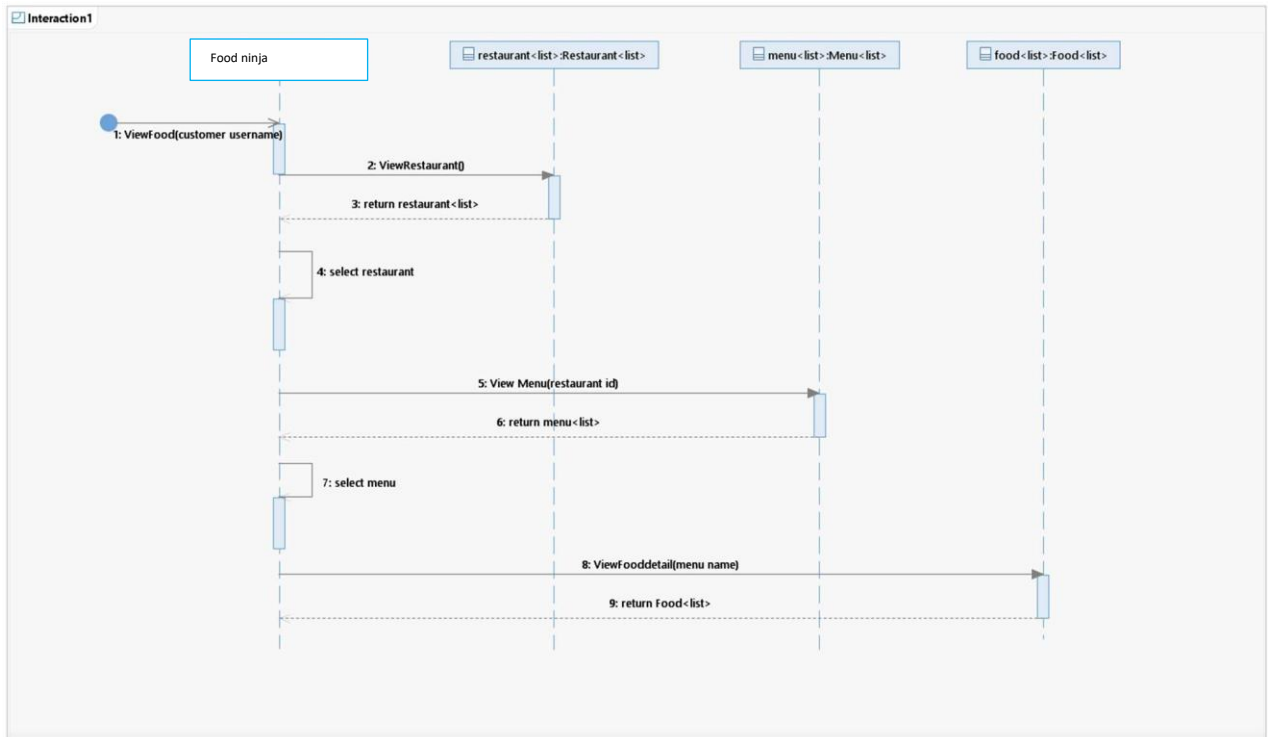




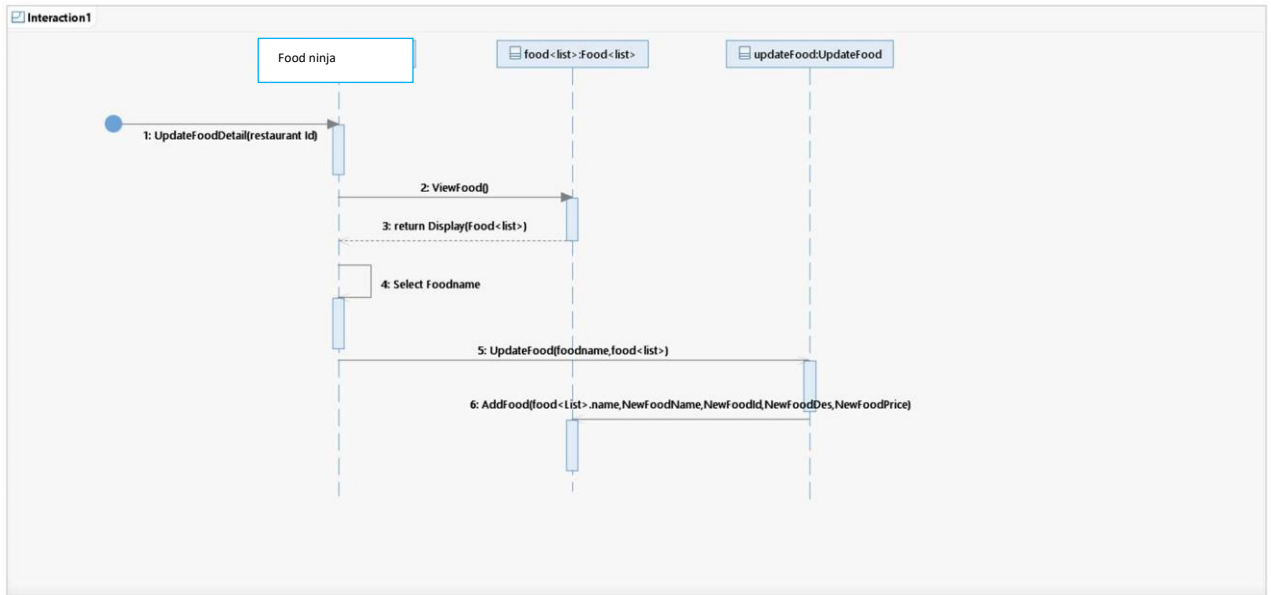
# Cancel Order



# View Food

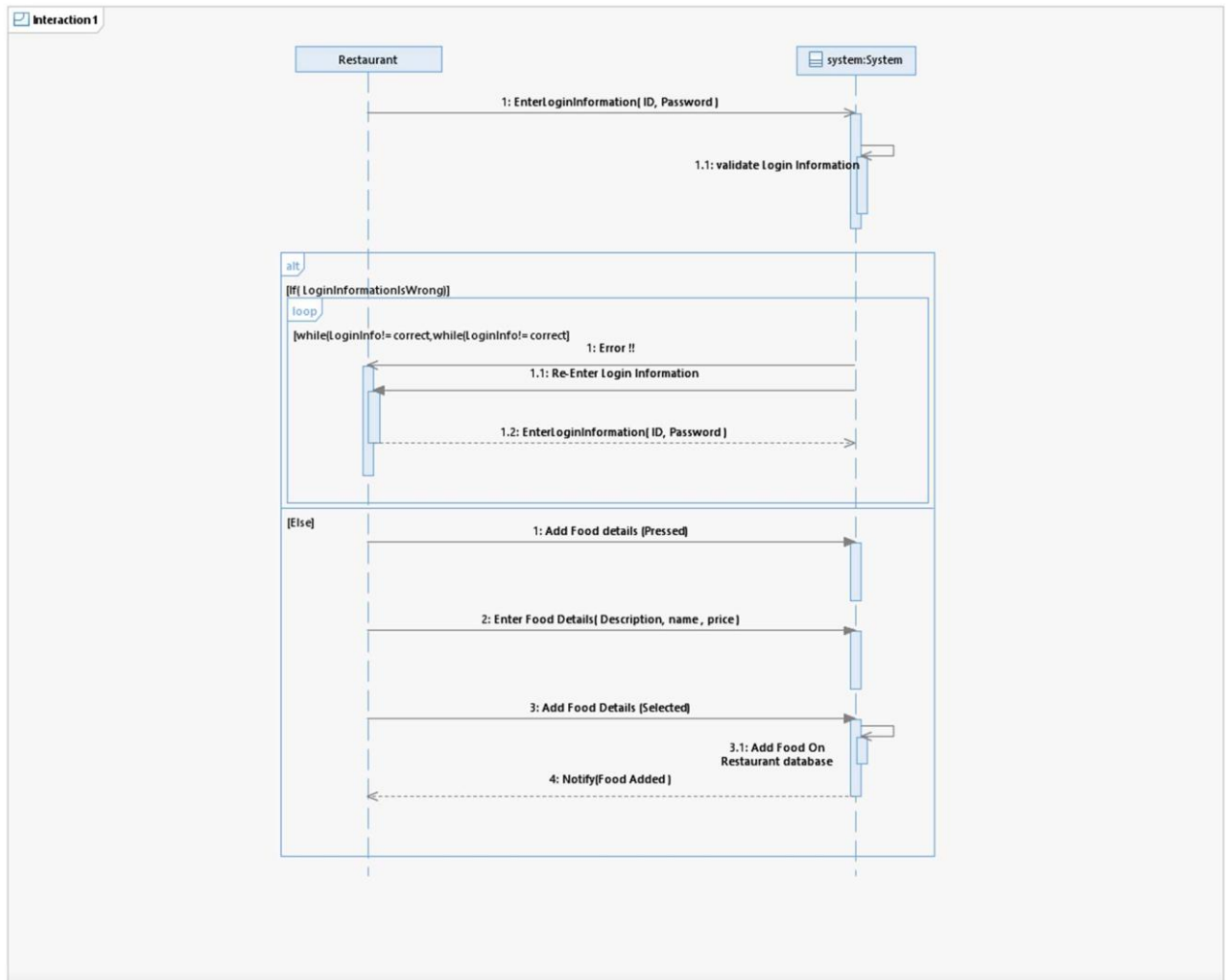


# Update Food

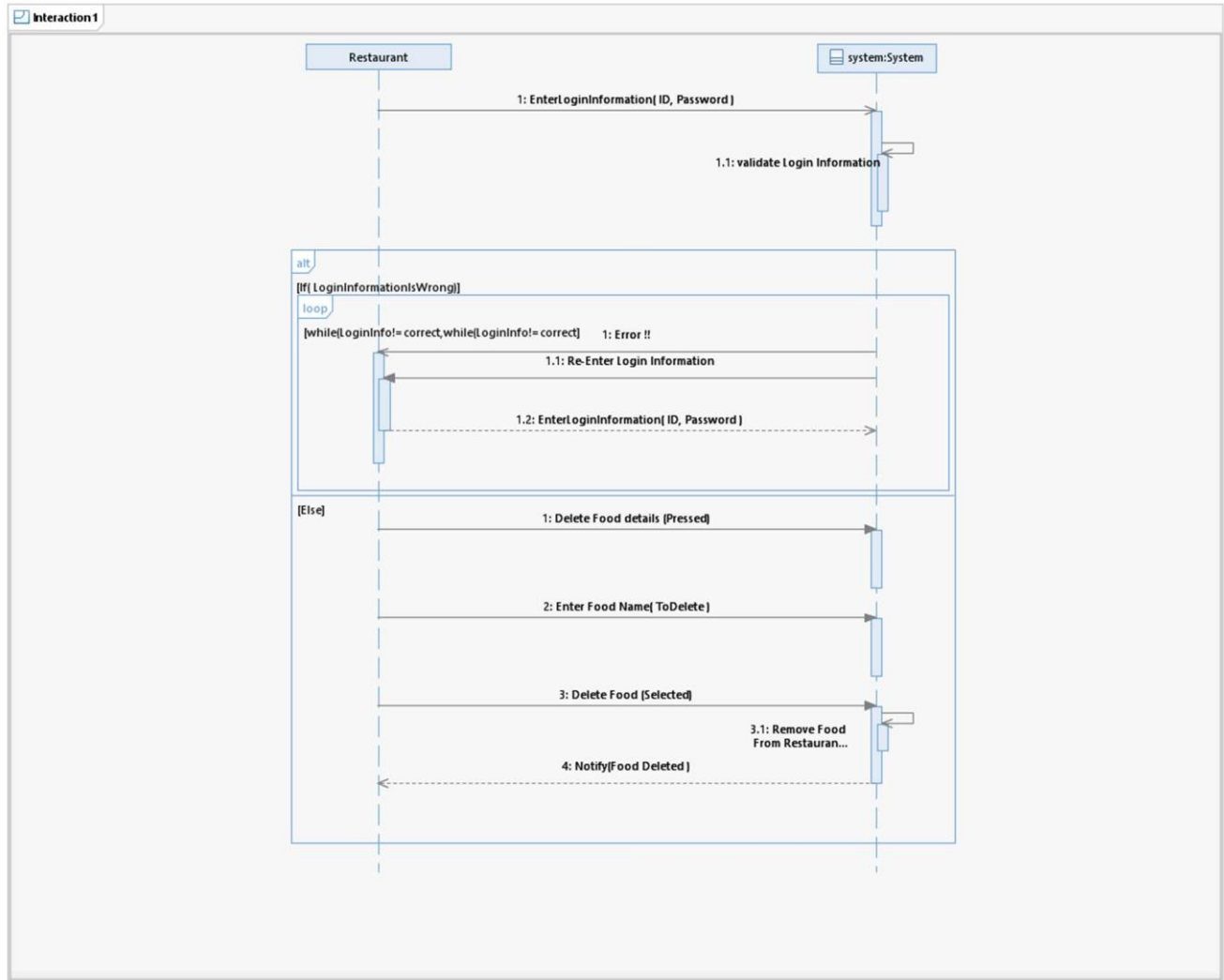


# **System Sequence Diagrams**

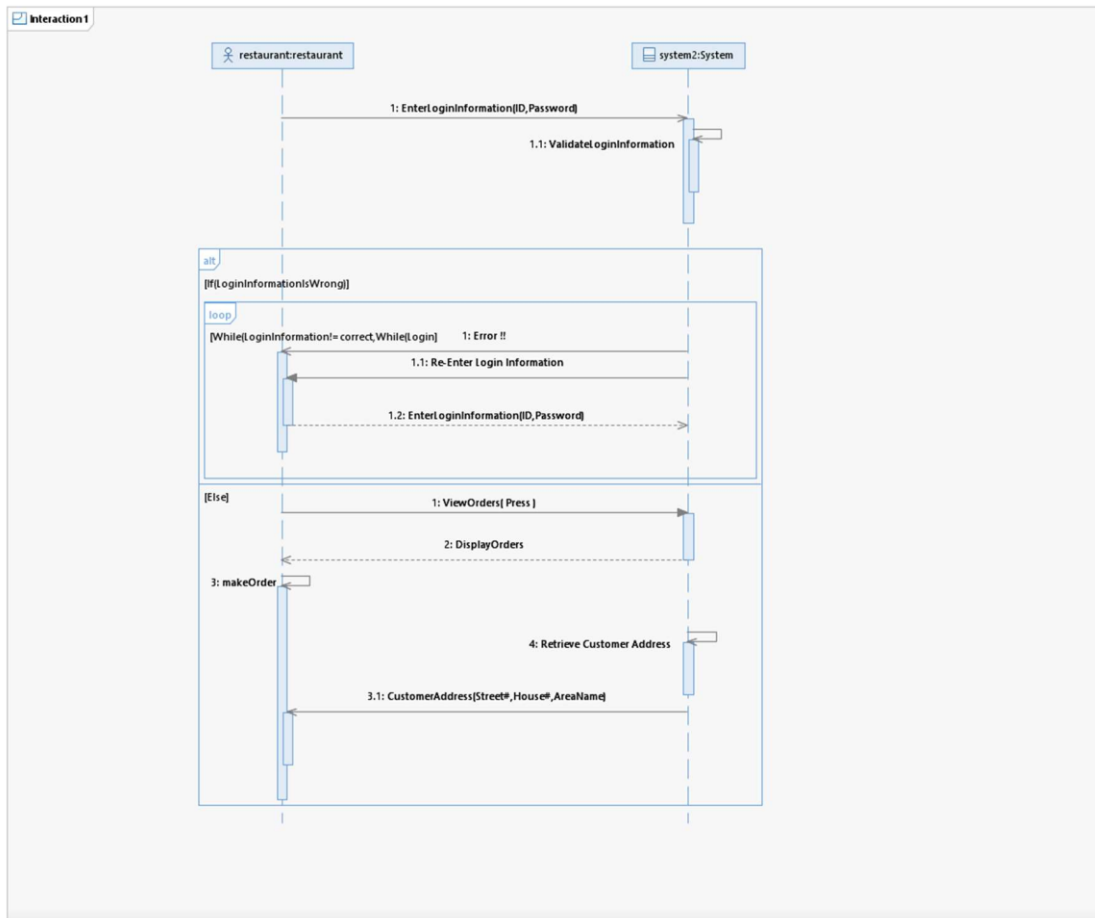
# Add Food Details



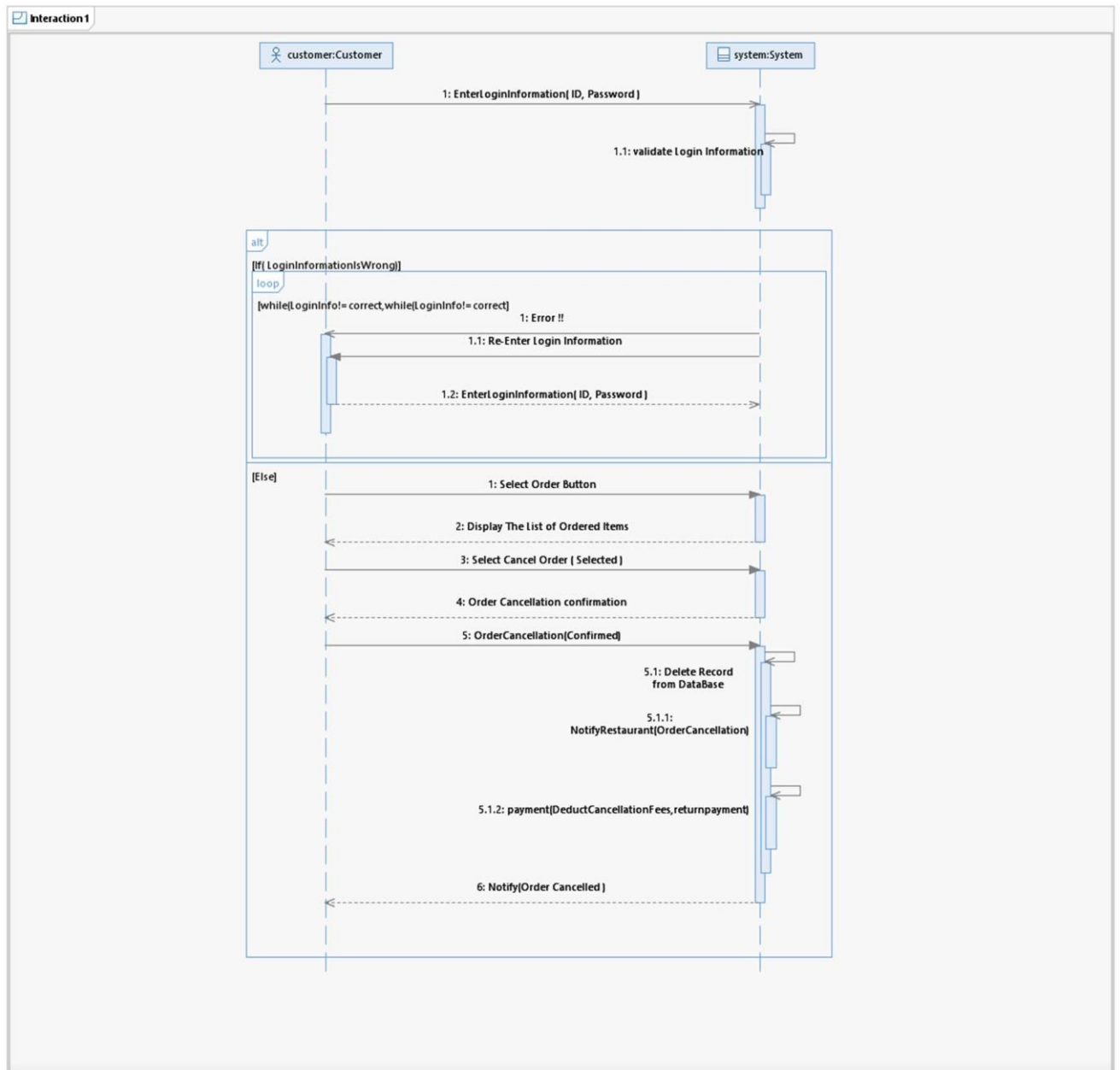
# Delete Food



# Check Food Order

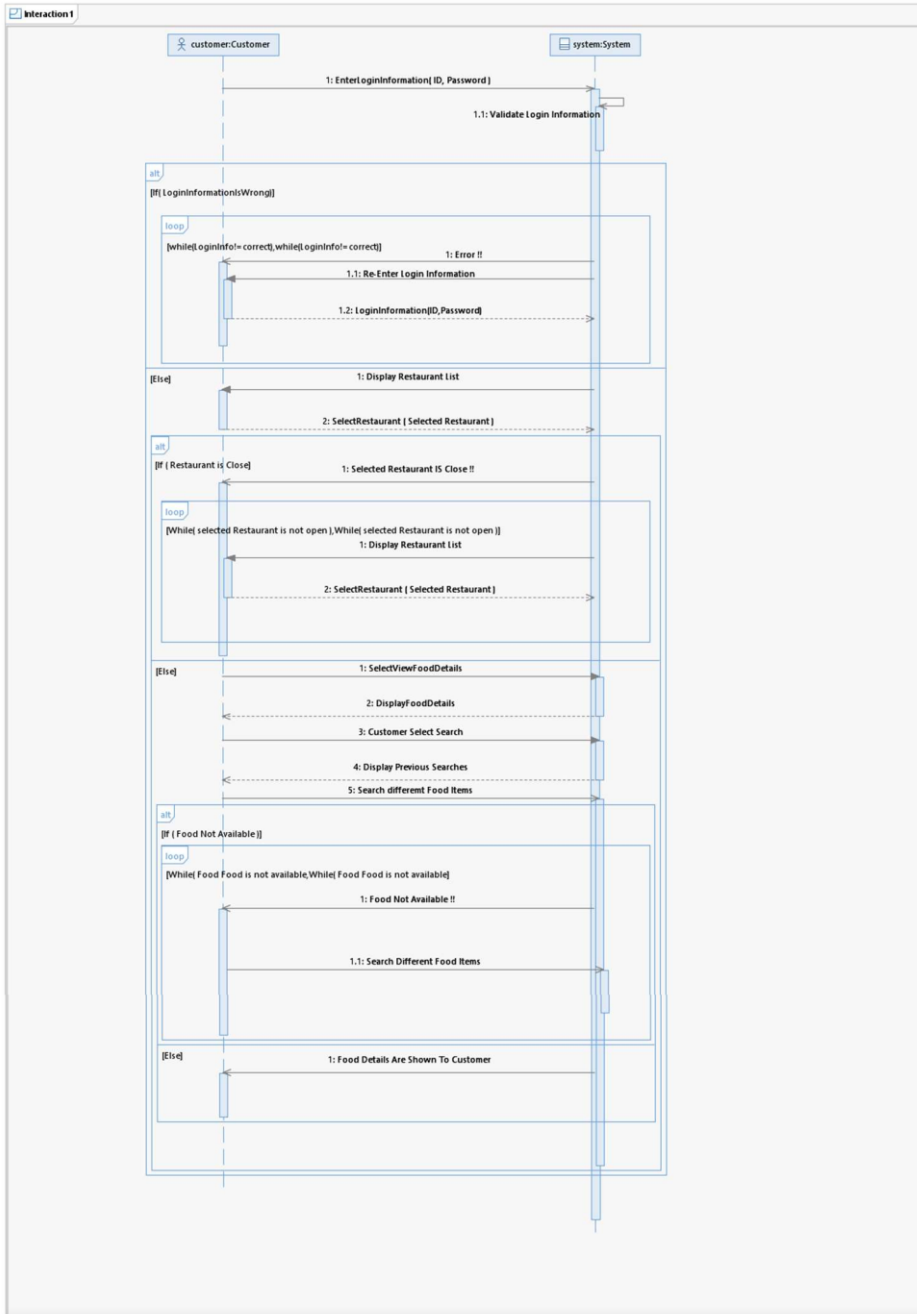


# Cancel Order



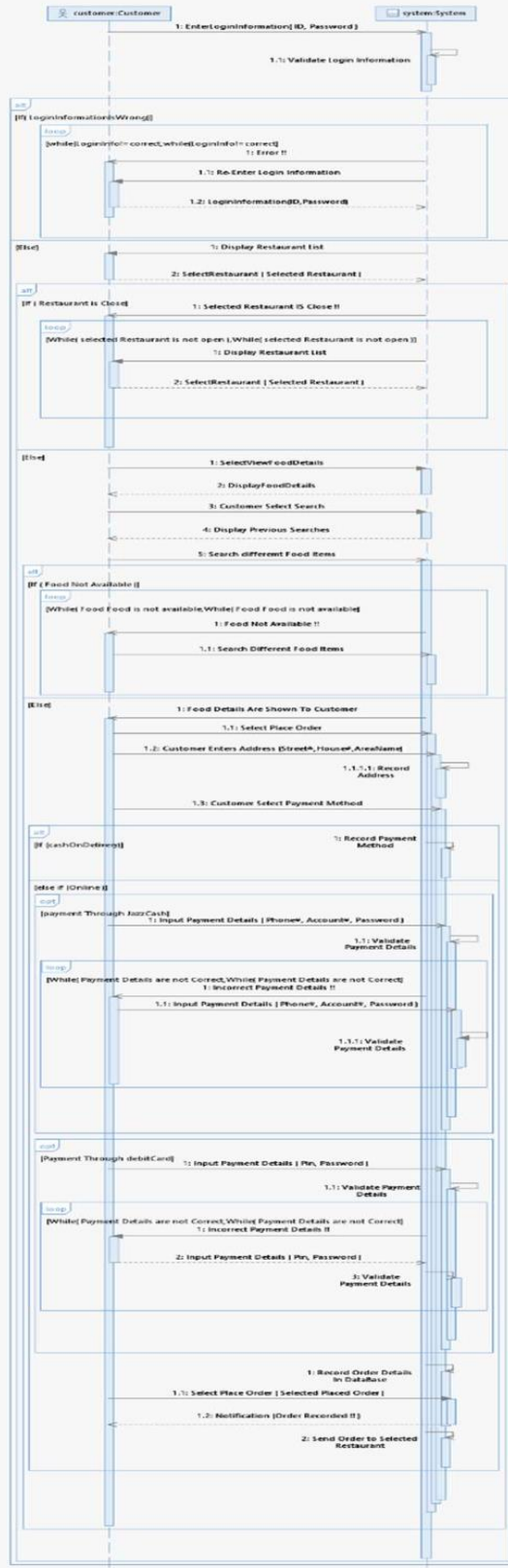


# View Food Details

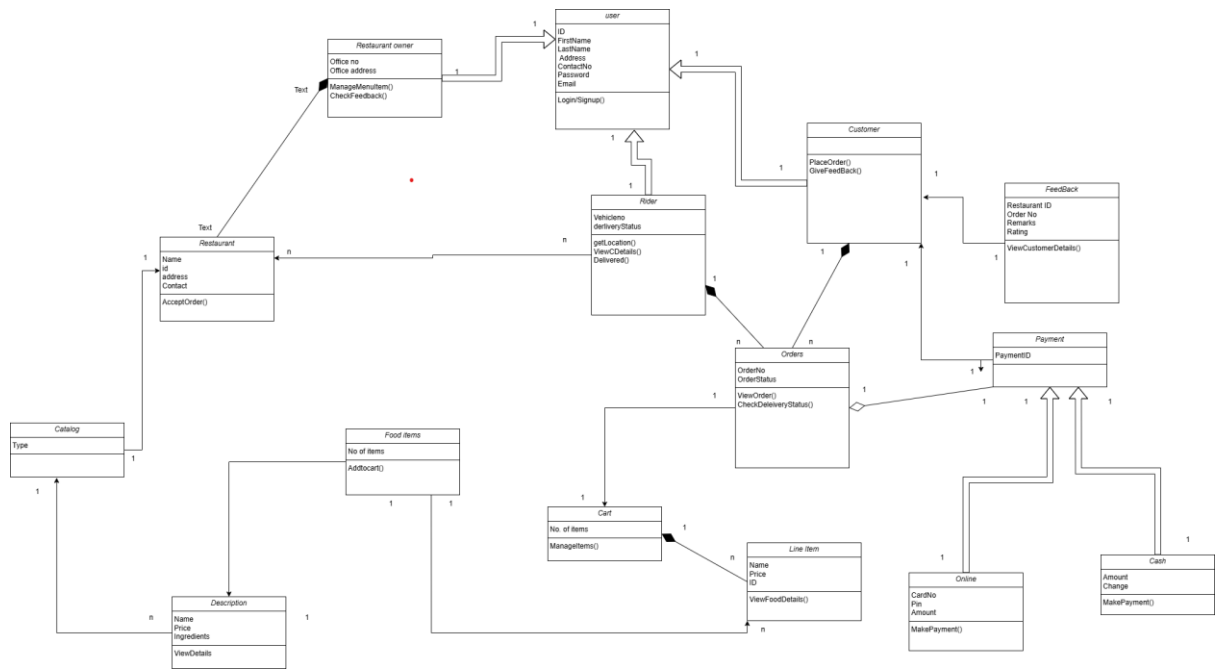


## **Place Order**

# Interaction 1



# **Class Diagram**



About Food Ninja: Food Ninja is designed to serve the urban population, particularly working professionals and students. It is supposed to be an affordable competitor to Food Panda and Cheetah.

**The primary objectives of Food Ninja is to:**

- Provide users with a convenient and efficient way to order food from local restaurants.
- Browse menus and place orders
- Rate meals and delivery experiences
- Avoid long waiting lines at restaurant.
- Featuring restaurants with discounts for cheaper meals.

**Aim:**

- It targets users in metropolitan areas with active restaurant and food delivery services.
- Working professionals, students and individuals with a busy lifestyle.
- Cheap and hygienic food at your doorstep.

# **Step by Step working of the App**

Login:



FOOD NINJA

Log In

*eat whatever u like*



Back



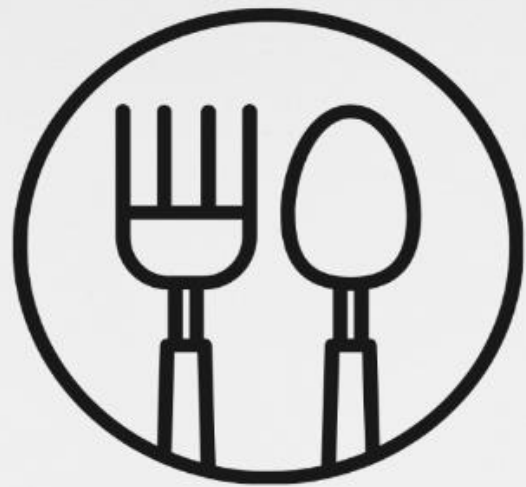
LOGIN  
AS

Admin

Restaurant

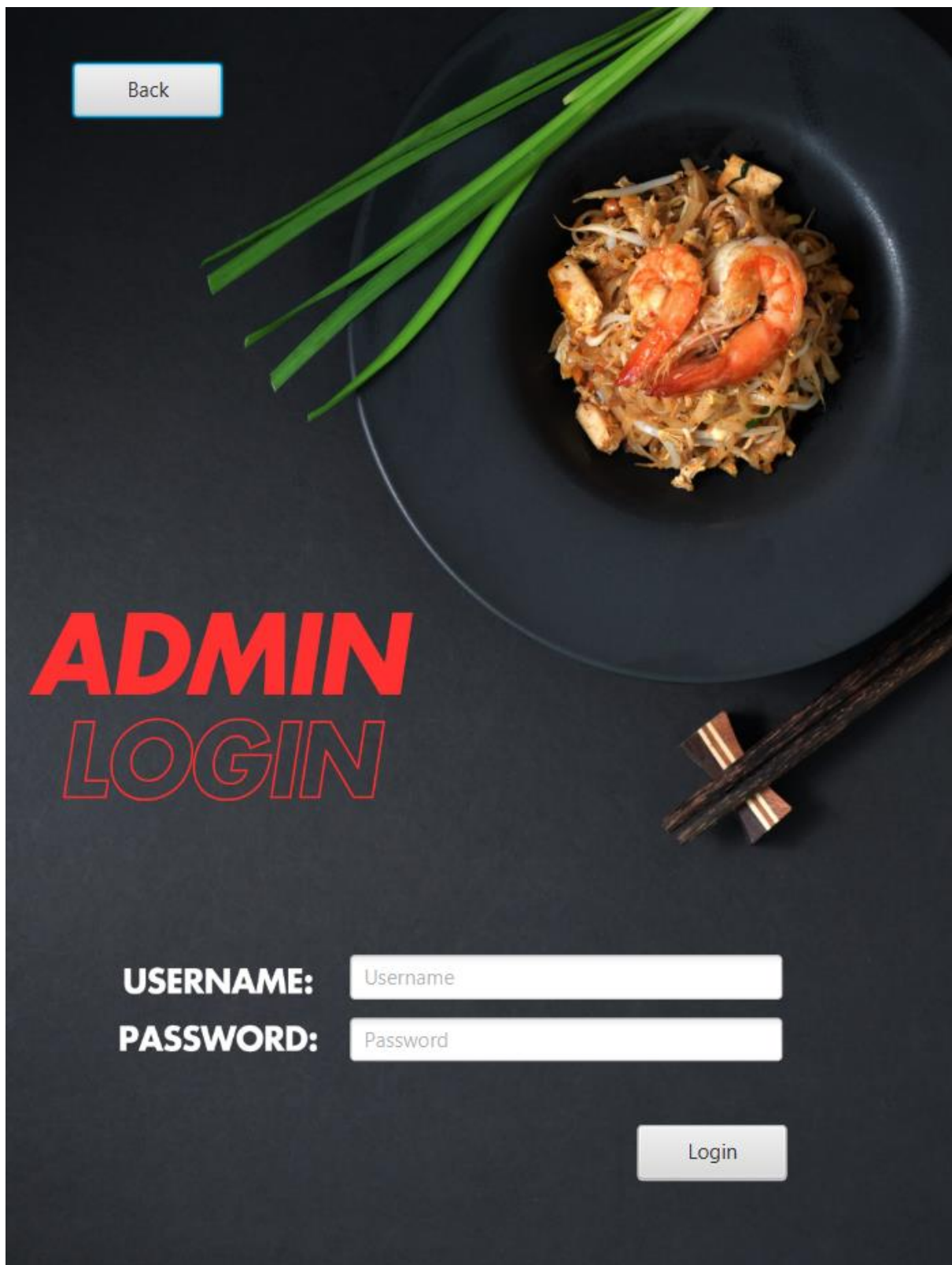
Customer

Rider



[www.FoodNinja.com](http://www.FoodNinja.com)

## Admin Login



Back

# ADMIN LOGIN

**USERNAME:**

**PASSWORD:**

Login

## Customer login



# Customer

SIGN IN

**Username:**

**Password:**

[Back](#) [Signup](#) [Sign In](#)

## Customer Signup



# CUSTOMER SIGN UP

NAME:

USERNAME:

MOBILE NO:

DOB:



PASSWORD:

CONFIRM PW:

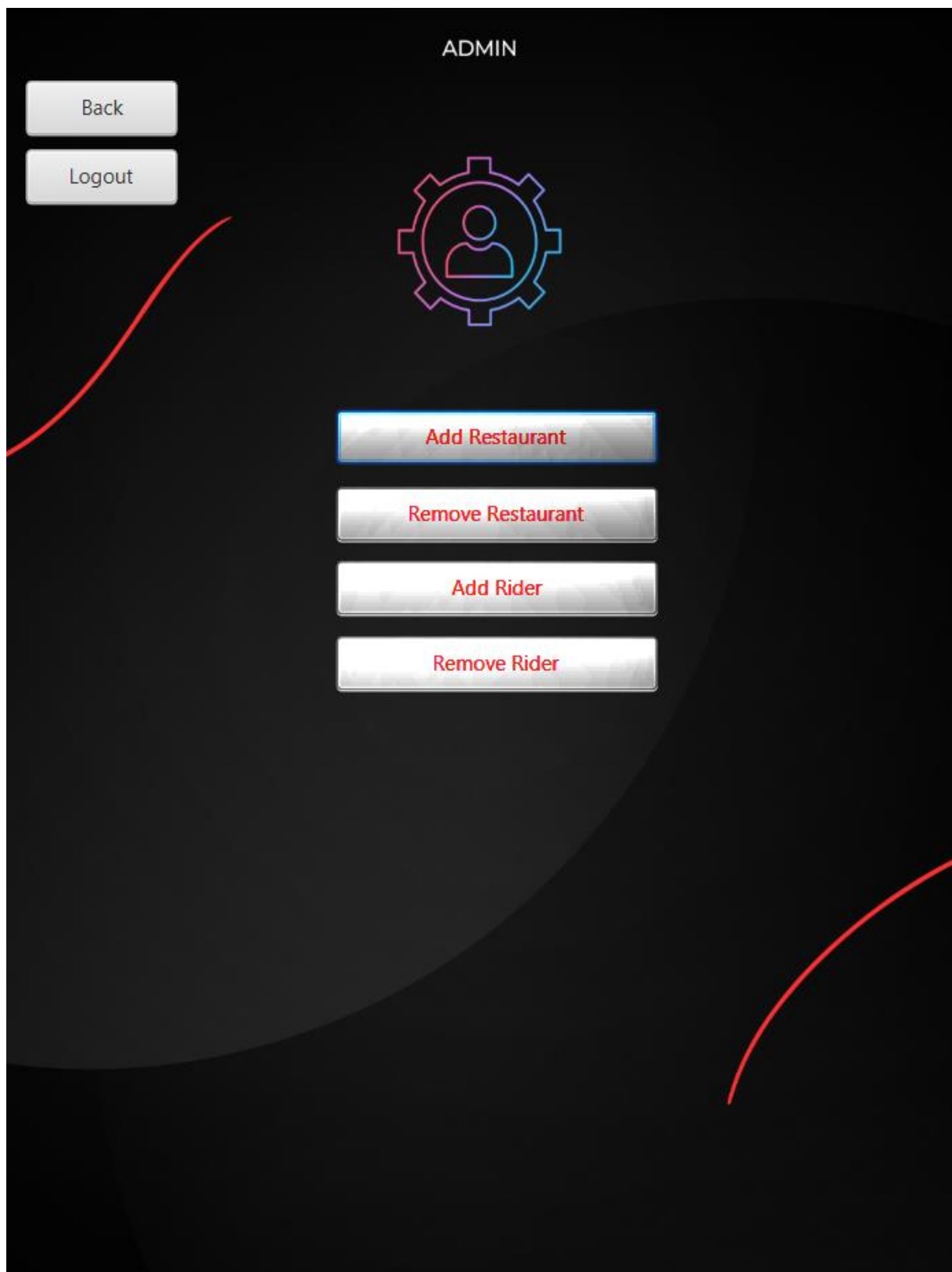
Back

Signup





## Admin Main Menu



## Add restaurant



**ADD RESTAURANT**


Restsaurant Name:

Restsaurant ID:

CITY:

Contact Number:

Location:

Password:

Confirm Password:

## Add rider



**ADD RIDER**

**Rider ID:**

**Rider Name:**

**Username:**

**Mobile No:**

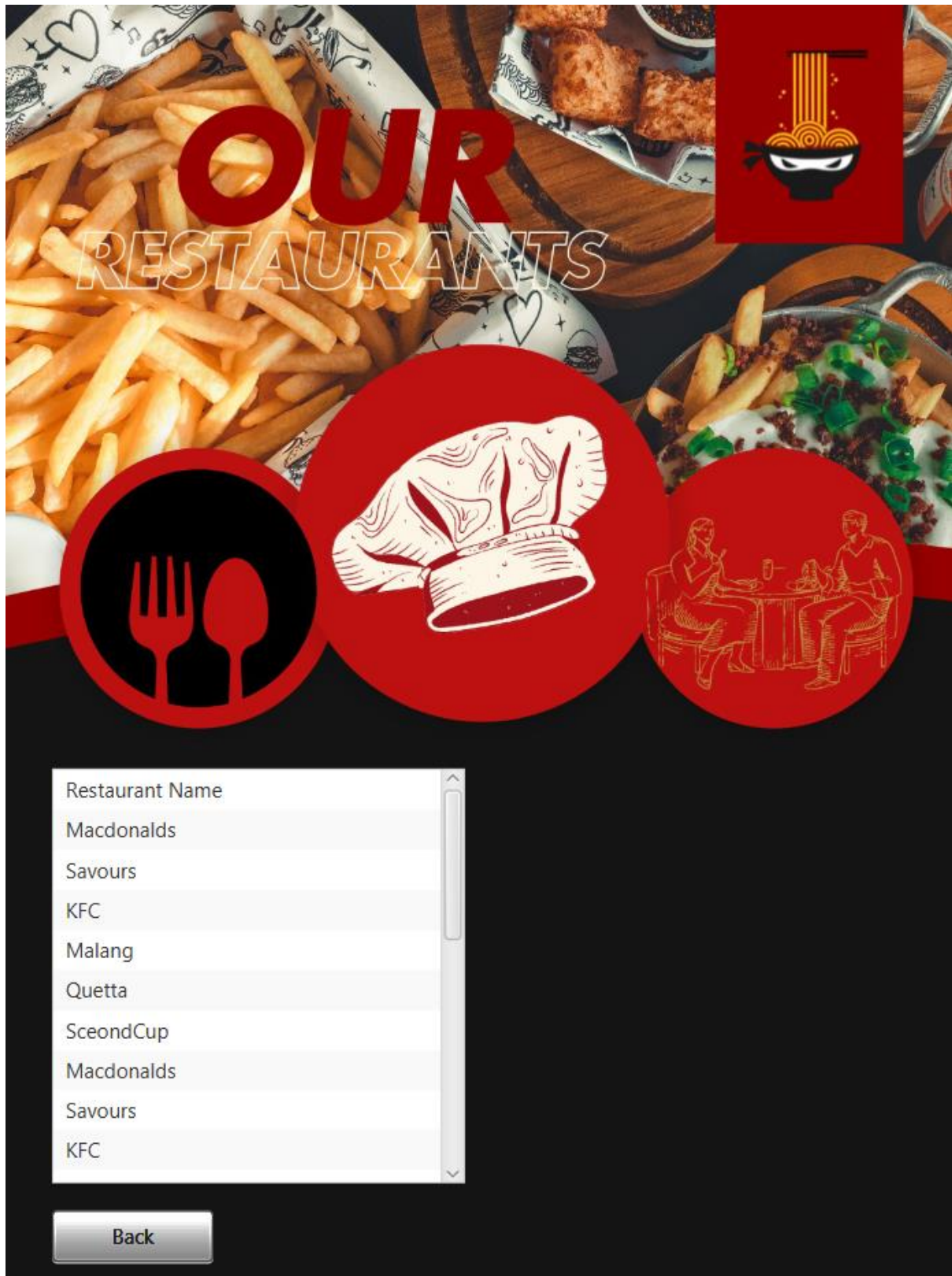
**DOB:**  

**Password:**

**Confirm PW:**



## View Restaurants





## View Riders



### OUR RIDERS

Rider Name

Sheraz Ahmed

Adil Ali

Hamza Akbar

ahmer

Sheraz Ahmed

Adil Ali

Hamza Akbar

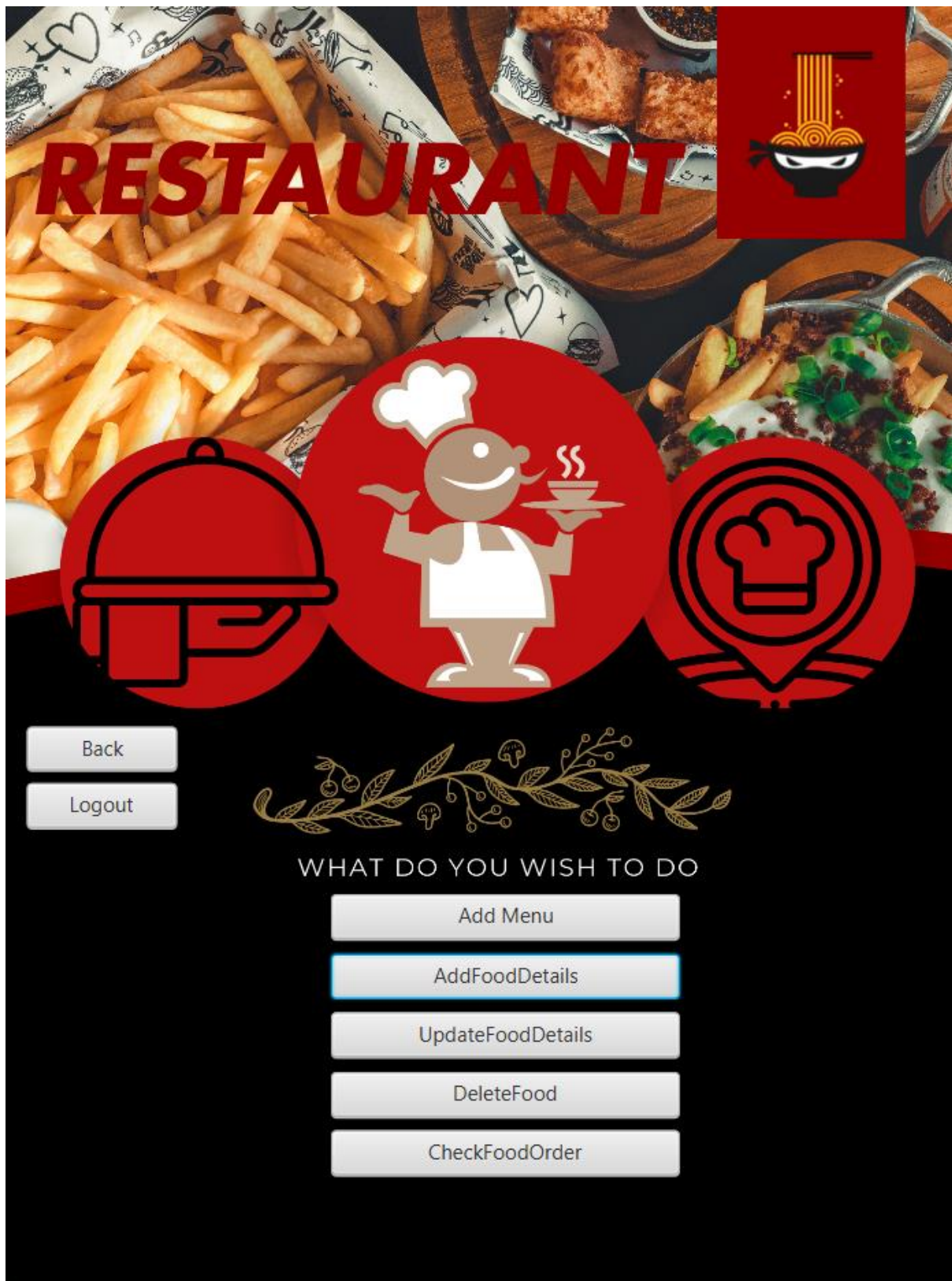
ahmer

Sheraz Ahmed

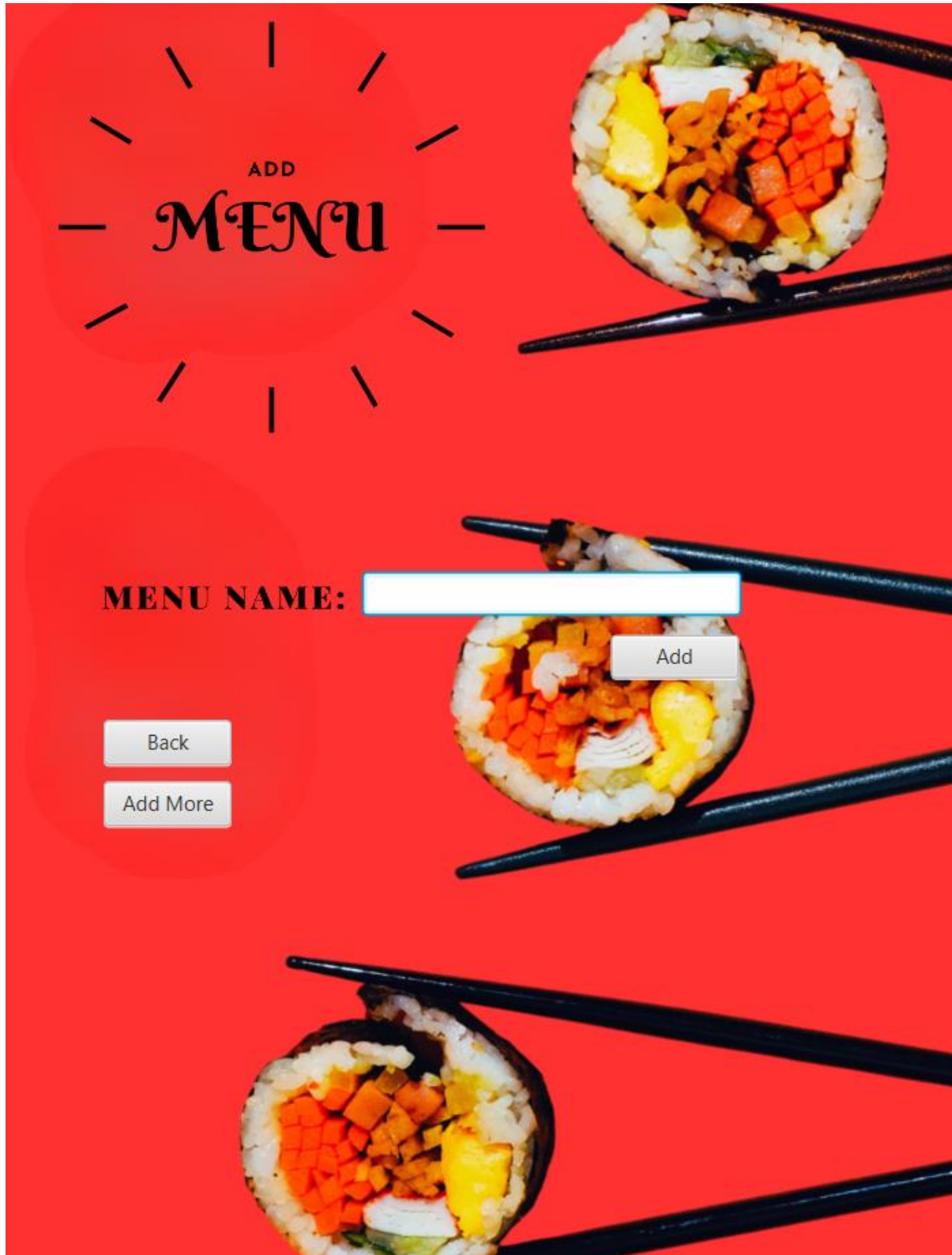
Back



## Restaurant Main Menu



## Add Menu Details



ADD  
**MENU**

**MENU NAME:**

Back

Add More

Add

## Food menu





[illegible]

**Cart**



**Payment**



BACK

# PAYMENT METHOD

**CASH** ☐

**CARD** ☐

Place Order



THANK YOU



111-696-717-272



[www.foodninja.com](http://www.foodninja.com)



[foodninja@nu.edu.pk](mailto:foodninja@nu.edu.pk)

**View Order**





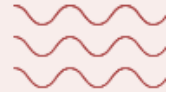
**Rider Main Menu**



Back

Logout

# Rider



Received Order

Deliver Received Order



## **GRASP Patterns:**

### **Creator:**

Customer->order

User->all other users (customer, admin, etc)

admin->restaurant/rider

restaurant->menu

### **Information Expert:**

Restaurant->Menu.

Restaurant knows its menu.

Cart line item

Rider->Order. Rider knows delivery address

### **Low Coupling:**

payment parent to different ways to pay

user class parent to different types of users

**High Cohesion:** Ensure classes like Customer, Rider, and Restaurant focus on their roles.

Order doesn't know how rider implements it.

Hidden

File handling so right info passed b/w classes

### **Session controller:**

User class manages user login, logout, and session tracking

An Order class coordinates the customer's cart, payment, and order creation processes.

## **GoF Patterns:**

**Factory:** To create instances of users (e.g., Customer, Rider) dynamically.

User class creates instances of rider, admin and customer.

Payment for different option. (Cash/Online)

**Strategy:** For payment options (e.g., implementing different payment methods).

Implement different payment strategies (Cash/Online)

Assigning order to drivers based on customer preference

**Observer:** For order status updates, where customers get notified when their order status changes.