

## EXPERIMENT-5

5) Draw a UML diagram for a food ordering system Systems. The activities of the food ordering system are listed below. Receive the Customer food orders, Produce the customer ordered food, Serve the customer with their ordered food, collect payment from Customers, Store customer payment details, Order Raw Materials for food products, Pay for Raw Materials and Pay for Labour.

### Aim:

To design a UML Diagram for a Food Ordering System, illustrating the interactions between customers, kitchen staff, payment processing, and supply management.

### Procedure:

#### 1. Identify the Main Entities (Classes)

- **Customer:** Places food orders and makes payments.
- **Order:** Contains details of the food ordered by the customer.
- **Kitchen:** Prepares the food ordered.
- **Payment:** Handles customer payments and payment details storage.
- **Supplier:** Provides raw materials for food production.
- **Restaurant Staff:** Serve food and manage payment processing.

#### 2. Define Attributes and Methods

##### Customer

- **Attributes:** customerID, name, contact, paymentDetails
- **Methods:** placeOrder(), makePayment()

##### Order

- **Attributes:** orderID, foodItems, orderStatus, totalCost
- **Methods:** updateOrderStatus(), calculateTotal()

##### Kitchen

- **Attributes:** kitchenID, availableStaff, foodInventory
- **Methods:** prepareFood(), checkIngredients()

##### Payment

- **Attributes:** paymentID, amount, paymentMethod, transactionStatus

- **Methods:** processPayment(), storePaymentDetails()

#### Supplier

- **Attributes:** supplierID, rawMaterials, cost
- **Methods:** deliverMaterials(), receivePayment()

#### Restaurant Staff

- **Attributes:** staffID, role
- **Methods:** serveFood(), collectPayment()

### 3. Establish Relationships

- Customer ↔ Order: One customer can place multiple orders.
- Order ↔ Kitchen: The kitchen prepares the ordered food.
- Customer ↔ Payment: Each customer payment is linked to an order.
- Supplier ↔ Kitchen: Kitchen orders raw materials from suppliers.
- Restaurant Staff ↔ Customer: Staff serve food and collect payments.

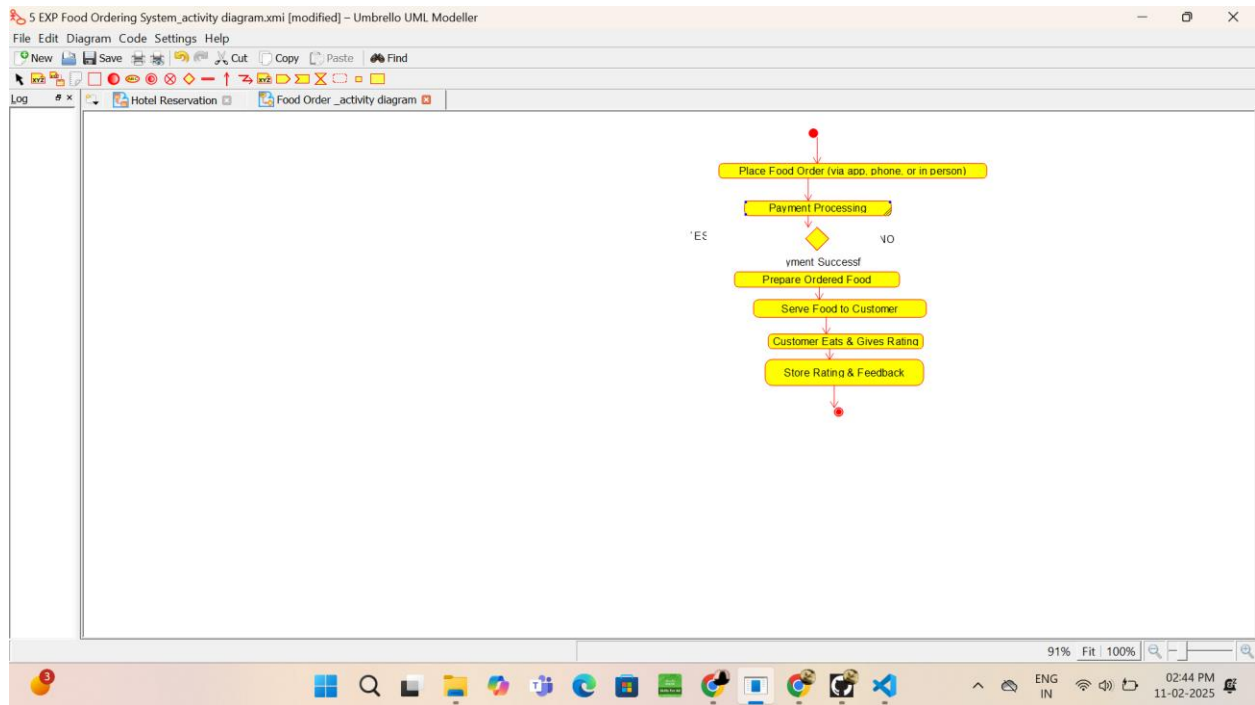
### 4. Draw the UML Class Diagram

- Represent **classes** as rectangles with attributes and methods.
- Use **associations** to connect related classes.
- Apply **aggregation and composition** for relationships like order and payment.

### 5. Verify the Diagram

- Ensure that all food ordering system activities are represented correctly.

#### Diagram:



## Result:

A **UML Diagram** for the **Food Ordering System** was successfully designed, showing interactions between customers, kitchen staff, suppliers, and payment processing.