**Problem Statement**

The Food Delivery System aims to provide a seamless experience for users to order food from various restaurants. Users can select a restaurant, view the menu, add items to their cart, and proceed to checkout with different payment methods. The system handles cart management, order processing, and order tracking efficiently.

**Functional Requirements**

**1. Restaurant Management**

* Add restaurants with unique IDs and names.
* Each restaurant should have a menu consisting of multiple items.

**2. Menu Management**

* Each menu item should have a name and price.
* Users can view the menu items of a selected restaurant.

**3. Cart Management**

* Users can:
  + Add items to their cart with specified quantities.
  + Remove items from their cart.
  + View the contents of their cart, including the total price.
  + Clear their cart.

**4. Order Management**

* Users can:
  + Place an order based on items in their cart.
  + Track the status of their order.
* Each order has:
  + A unique ID.
  + Status (e.g., PLACED, OUT\_FOR\_DELIVERY, DELIVERED).
  + A unique tracking ID.

**5. Payment Processing**

* Payment methods include UPI, Card, and Cash on Delivery.
* The system processes payments and confirms orders.

**Class Descriptions**

**1. Restaurant**

* **Attributes:**
  + id: Unique identifier.
  + name: Restaurant name.
  + menuItems: Map of menu items available.
* **Methods:**
  + addMenuItem(MenuItem item): Adds an item to the menu.
  + getMenuItem(String name): Retrieves an item by name.
  + getMenuItems(): Returns all menu items.

**2. Menu Item**

* **Attributes:**
  + name: Name of the menu item.
  + price: Price of the menu item.
* **Methods:**
  + getName(): Returns the name of the item.
  + getPrice(): Returns the price.

**3. Cart**

* **Attributes:**
  + items: Map of menu items and their quantities.
* **Methods:**
  + addItem(MenuItem item, int quantity): Adds an item to the cart.
  + removeItem(MenuItem item, int quantity): Removes an item from the cart.
  + clearCart(): Clears the cart.
  + getItems(): Returns the items in the cart.
  + getTotalPrice(): Calculates the total price.

**4. Order**

* **Attributes:**
  + id: Unique identifier.
  + status: Current status.
  + cart: Associated cart.
  + trackingId: Unique tracking ID.
* **Methods:**
  + getId(): Returns the order ID.
  + getStatus(): Returns the order status.
  + updateStatus(String newStatus): Updates the order status.

**5. Payment (Interface)**

* **Methods:**
  + processPayment(double amount): Processes the payment.

**Sample Execution Flow**

**Restaurant Selection**

Available Restaurants:

1. Burger Joint
2. Pizza Place

Select a restaurant by number: **1**

**Menu Display**

Menu for Burger Joint:

* Burger - $5.0
* Fries - $2.0
* Soda - $1.5

**Cart Options**

Cart Options:

1. View Cart
2. Add Item
3. Remove Item
4. Clear Cart
5. Proceed with Order

Choose an option: **2**

Enter item name: **Burger**

Enter quantity: **3**

Burger added to cart.

**Viewing the Cart**

Items in your cart:

Burger x 3 - $15.0

Total Price: **$15.0**

**Removing Items**

Enter item name to remove: **Burger**

Enter quantity to remove (up to 3): **1**

1 Burger(s) removed from cart.

**Clearing the Cart**

Cart cleared. Do you want to continue with the same restaurant? (yes/no)

no

**Selecting a New Restaurant**

Available Restaurants:

1. Burger Joint
2. Pizza Place

Select a restaurant by number: **2**

**Adding Items and Proceeding with Order**

Menu for Pizza Place:

Pizza - $10.0

Salad - $3.0

Choose an option: **5**

Choose payment method:

1. UPI
2. Card
3. Cash on Delivery

Processing UPI payment of $20.0

Your order has been placed. Tracking ID: 3bf3958f-abb6-4a73-beec-4efa9611327b

**Tracking the Order**

Track your order (Enter anything to check status): **ok**

Order status: **OUT\_FOR\_DELIVERY**

Track your order (Enter anything to check status): **ok**

Order status: **DELIVERED - Delivered Successfully!**