**FPT University – SWD (COMET Method)**

**PT2: Dynamic Modeling – Online Food Ordering System**

**I. Project Context**

You are part of a software design team developing the **Online Food Ordering System (OFOS)** for a food delivery startup. The system allows customers to browse nearby restaurants, order food online, make payments, and track their delivery status in real-time.

**User Requirements**

The system should support the following user scenarios:

1. **Customer Registration & Login**
   * A customer can create an account using email or phone number.
   * Login using password or OTP.
2. **Restaurant Browsing & Menu Viewing**
   * Customers can browse restaurants by location, rating, or category.
   * Each restaurant has a digital menu with items, prices, and availability.
3. **Placing an Order**
   * Customers can select multiple items, customize (extra toppings, no spice, etc.), and place an order.
   * System should calculate total cost including tax and delivery fee.
4. **Payment**
   * Customers can pay using card, e-wallet, or cash-on-delivery.
   * System must confirm payment and issue an invoice.
5. **Order Tracking & Delivery**
   * Customers can track their order status (e.g., *Placed → Accepted → Cooking → Out for Delivery → Delivered*).
   * System should update customers in real time.
6. **Feedback & Rating**
   * After delivery, customers can rate the food and delivery experience.
   * Feedback should be stored and visible to the restaurant admin.

**II. Progress Test Tasks**

**Question 1 – Identify Subsystems (10%)**

Based on the user requirements above:

* Identify **logical subsystems** of the Online Food Ordering System.
* Explain **the criteria** used for decomposition (e.g., functional grouping, user roles, or transaction flow).
* Justify **why the chosen decomposition is appropriate** for this system.

|  |  |
| --- | --- |
| Subsystem Name | Responsibilities |
| User & Authentication Service | Manages customer registration, login (password/OTP), profiles, and authentication tokens. |
| Restaurant & Menu Service | Handles restaurant listings, browsing, searching, and viewing digital menus with item details and availability. |
| Order Management Service | Core service that processes order creation, customization, cost calculation, and manages the order lifecycle. |
| Payment Service | Manages all payment-related transactions, including card, e-wallet, and cash-on-delivery, and issues invoices. |
| Delivery & Tracking Service | Handles the logistics of assigning drivers, and provides real-time status updates to the customer from pickup to delivery. |
| Feedback & Rating Service | Collects and stores customer ratings and feedback for both food and delivery, making it available to restaurants. |

-Criteria for Decomposition: The primary criterion used for this decomposition is **functional grouping**, also known as decomposition by feature. This approach involves identifying the main, distinct capabilities of the system and organizing them into separate, self-contained subsystems. Each subsystem is responsible for a specific business capability from end to end.

- Justification:

1. **Easier Maintenance and Updates:** This method creates independent modules with clear responsibilities (**high cohesion, low coupling**), which makes it easier to fix or upgrade one feature without impacting others.
2. **Efficient Scalability:** It allows individual parts of the system to be **scaled** as needed. For example, you can add more resources to just the menu service during peak hours instead of scaling the entire application, which is more cost-effective.
3. **Faster Development:** It enables **parallel development**, as different teams can work on separate modules simultaneously, accelerating the project completion timeline.

*Deliverable:* Table listing subsystems + 1 paragraph explanation.

**Question 2 – High-Level Communication Diagram (15%)**

Draw a **High-Level Communication Diagram** showing how the subsystems interact.  
Each arrow should represent a communication or data exchange between subsystems.  
For example, “Order Service → Payment Service: Request Payment Authorization”.

*Deliverable:* UML communication diagram (architecture-level).

A diagram of a company

AI-generated content may be incorrect.

**Question 3 – Choose One Subsystem (15%)**

Select one **critical subsystem**.

* Identify its **actors**.
* List its **use cases**.
* Draw a **Use Case Diagram** for this subsystem.

*Deliverable:* Use Case Diagram.

A diagram of a food order

AI-generated content may be incorrect.

**Question 4 – State Machine Diagram (20%)**

Draw a **Finite State Machine Diagram** for the **main control class** in the selected subsystem.

Each transition should be triggered by a clear **event** (e.g., “restaurantAcceptsOrder”, “deliveryCompleted”).

*Deliverable:* UML State Machine Diagram.

A diagram of a delivery service

AI-generated content may be incorrect.

**Question 5 – Implementation-Level Class Diagram (20%)**

Create a **Class Diagram** for the chosen subsystem, using **COMET stereotypes**:

* boundary – interface classes interacting with actors
* control – logic coordination classes
* entity – data persistence classes

*Deliverable:* UML Class Diagram (implementation-level).

A blue diagram with black text

AI-generated content may be incorrect.

**Question 6 – Use Case Specification (10%)**

Pick one **key use case** from your subsystem (e.g., “Place Food Order”) and describe it using the template:

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC-01: Place Food Order | | |
| Created By: | Le Sy Hiep-HE180170 | Date Created: | 20/10/2025 |
| Primary Actor: | Customer | Secondary Actors: | Restaurant Service, Payment Service |
| Trigger: | The Customer clicks the "Confirm Order" button in their shopping cart. | | |
| Description: | This use case allows a logged-in customer to confirm the items in their shopping cart, proceed to payment, and create a new order in the system. | | |
| Preconditions: | 1. The Customer is logged into the system. 2. The Customer's shopping cart contains at least one item. | | |
| Post-conditions: | 1. A new Order entity is created in the system with a "Pending Confirmation" status. 2. An Invoice is generated for the order. 3. The system redirects the Customer to the payment gateway. | | |
| Normal Flow: | 1. The Customer reviews items in their cart on the CartUI. 2. The Customer clicks the "Confirm Order" button. 3. The OrderUI sends the cart details to the OrderController. 4. The OrderController validates item availability and calculates the total cost (including tax and delivery fee). 5. The OrderController creates a new Order object and saves it to the database. 6. The system displays the final order summary and prompts the customer for payment. | | |
| Alternative Flows: | **A1: Item becomes unavailable:** 1. In Normal Flow step 4, if an item is no longer available, the system displays an error message. 2. The system prompts the Customer to remove the unavailable item from the cart. 3. The use case continues from Normal Flow step 1. | | |
| Exceptions: | **E1: System connection failure:** 1. If the system cannot connect to the database at any point, it displays a "System Error" message to the Customer and logs the error. The use case terminates. | | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Business Rules: | BR-1: A delivery fee is applied based on the distance between the restaurant and the customer's address. BR-2: A 10% Value Added Tax (VAT) is applied to all orders. | | |
| Other Information: |  | | |
| Assumptions: | The customer's delivery address is pre-filled from their user profile. | | |

**Question 7 – System Sequence Diagram (SSD) (10%)**

Draw a **System Sequence Diagram** for the selected use case in Question 6.  
Include actor interactions and system messages in sequence order.

*Deliverable:* UML SSD diagram.

A diagram of a diagram

AI-generated content may be incorrect.

**III. Deliverables**

| **Deliverable** | **Description** |
| --- | --- |
| Subsystem Table | Identification & justification |
| Communication Diagram | Architecture-level communication |
| Use Case Diagram | For chosen subsystem |
| FSM Diagram | Main control class |
| Class Diagram | Implementation-level (with stereotypes) |
| Use Case Specification | Text document |
| SSD | System interaction diagram |

**File name:** OFOS\_COMET\_PT2\_<StudentID>\_<Name>.pdf  
**Submission:** eduNext

**\* Tools & Notes**

* Use **Visual Paradigm and create only one project** for diagrams.
* Apply **COMET stereotypes** correctly (boundary, control, entity).
* Maintain **naming consistency** across diagrams.