

**Software Requirement Engineering Lab Assignment 1**

Total Marks: 10

**Instructions:**

Diagrams must be created using StarUML only, Coding is not required in this assignment.

***Case Study: Online Food Delivery & Restaurant Management System*****Description:**

A system where customers can order food online from multiple restaurants, track delivery in real-time, make payments, and rate food. Restaurants can manage menus, process orders, and assign delivery personnel. Delivery personnel can receive delivery requests and update status. The system should handle multiple users simultaneously, maintain order history, and ensure secure payment processing.

**Question # 01: Requirements & Challenges****(2 Marks)**

- Identify **functional requirements** and **non-functional requirements** for the Online Food Delivery System.
- Explain **2 challenges** in requirement elicitation for such a system.

**Question # 02: Use Case Modeling****(4 Marks)**

- Draw a **Use Case Diagram** including at least **6 use cases** and all main actors.
- Briefly describe **interactions** between Customer, Restaurant, and Delivery Personnel.

**Question # 03: Use Case Narration & Activity Diagram****(3 Marks)**

- Write a **use case narration** for “Place Order” including: Actor, Pre-condition, Main Flow, Alternate Flow, Post-condition.
- Draw the **Activity Diagram** for “Track Order” process.

**Question # 04: Collaboration Diagram****(1 Mark)**

Draw a **Collaboration Diagram** for the “Assign Delivery Personnel” process.

**Submission Requirements:**

- StarUML **must be used** for all diagrams (Use Case, Activity, Sequence, Class, and Collaboration).
- Coding **must be in Python or C# only**. No other programming languages allowed.
- Security analysis is **theoretical only**; students should not attempt hacking.
- **GitHub submission is mandatory**, include all diagrams, SRS, and code.
- Assignment 1 must be submitted on the LMS in PDF format by the specified deadline.

---

## Question #01: Requirements & Challenges

### *a) Functional Requirements*

The system should be able to:

- Allow customers to register and log in to their accounts.
- Provide browsing access to restaurants along with their menus.
- Enable customers to place food orders and track delivery status.
- Support secure payment transactions.
- Let users rate food and submit feedback after order completion.
- Allow restaurants to manage menus, orders, and delivery personnel.
- Enable delivery personnel management including task assignment and status updates.

### *Non-Functional Requirements*

The system must also ensure:

- Highly secure and reliable payment handling.
- Real-time order and delivery tracking.
- Good performance and ability to scale with increasing user load.
- A smooth, intuitive, and user-friendly interface.

---

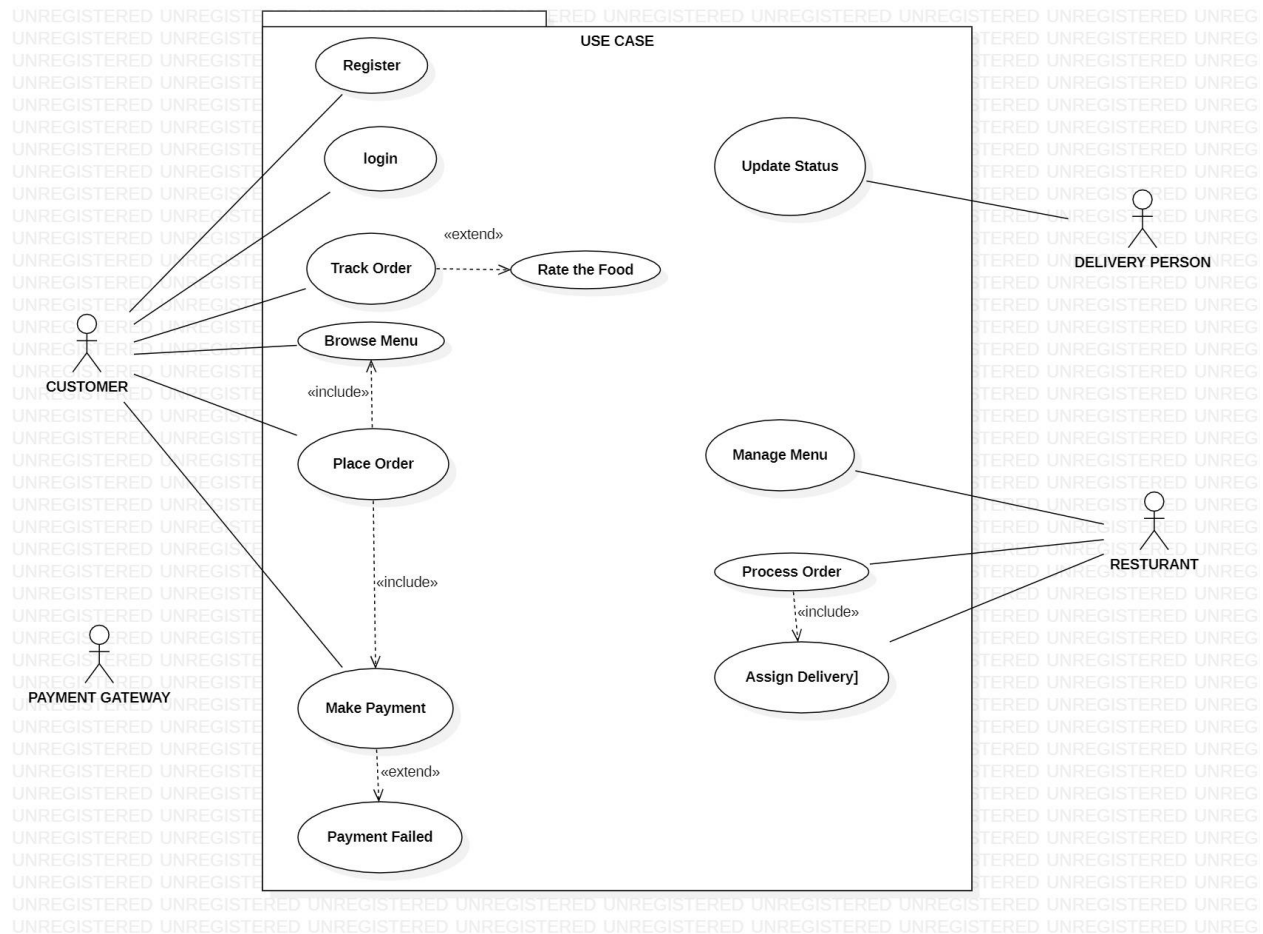
## b) Challenges in Requirement Elicitation

- Collecting requirements from different stakeholders such as customers, restaurant owners, and delivery staff, each having their own expectations and concerns.
- Adapting requirements to fast-changing food delivery trends and improving them based on user behavior and market needs.

---

## Question #02: Use Case Modeling

### a) Use Case Diagram



#### Actors:

- Customer
- Restaurant
- Delivery Personnel
- Payment Gateway

#### Use Cases:

Register | Login | Browse Menu | Place Order | Track Order  
Manage Menu | Process Order | Assign Delivery | Update Status | Make Payment

### b) Interaction Flow

- A customer selects items and places an order.
- The restaurant receives and processes the order.
- Restaurant assigns a delivery rider for dispatch.
- Delivery personnel updates order status throughout the route.
- Customer can view the tracking progress until order is delivered.

---

## Question #03: Use Case Narration & Activity Diagram

### *a) Use Case Narration — Place Order*

**Actor:** Customer

**Pre-Condition:** User must be logged in and have selected food items in the cart.

**Main Flow:**

1. Customer reviews selected items and confirms order.
2. System processes the payment through the payment gateway.
3. Restaurant is notified to prepare the order.

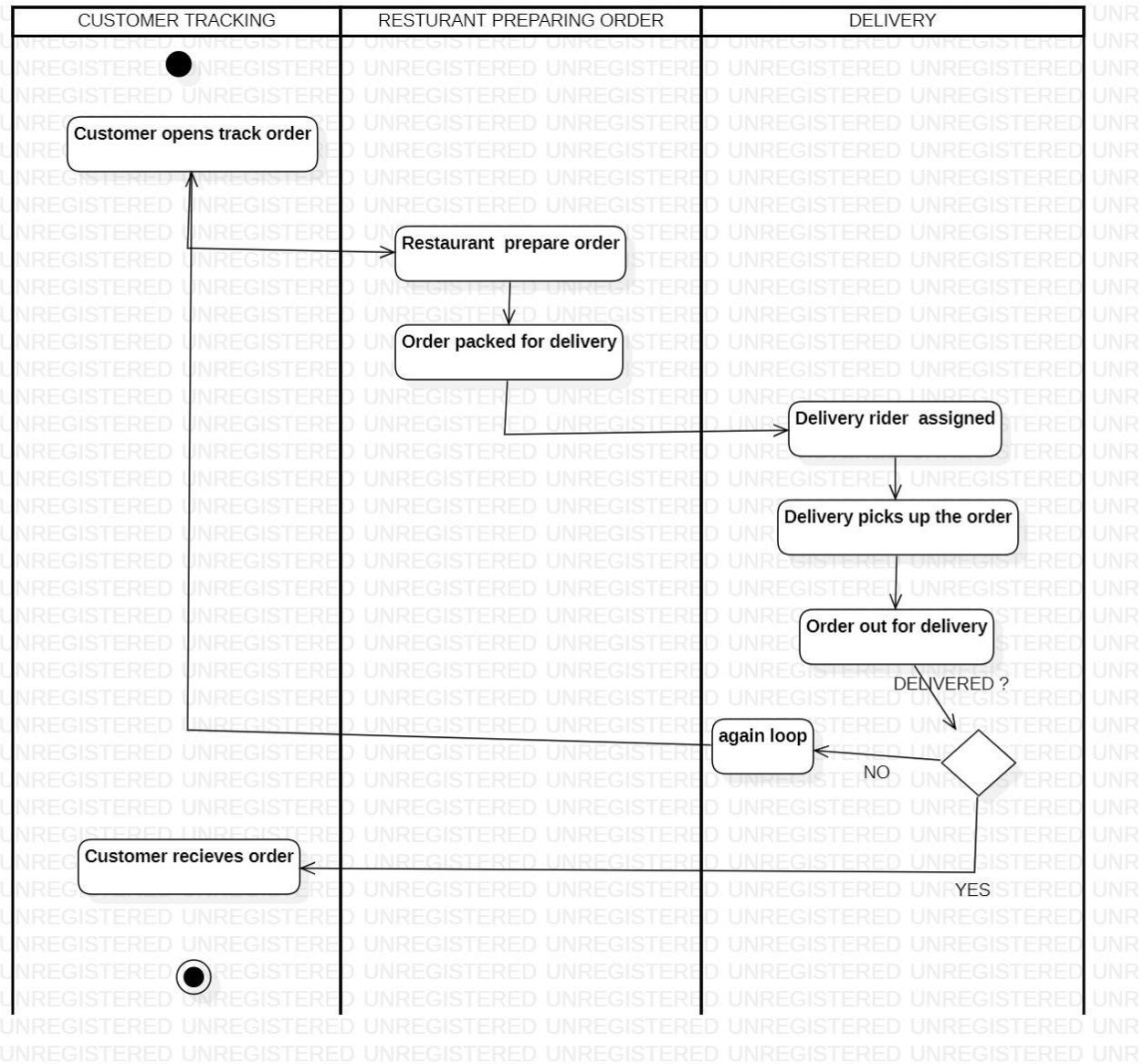
**Alternate Flow:**

- If the payment process fails, the order is cancelled automatically.

**Post-Condition:**

- Order is successfully placed and payment confirmation is recorded.
-

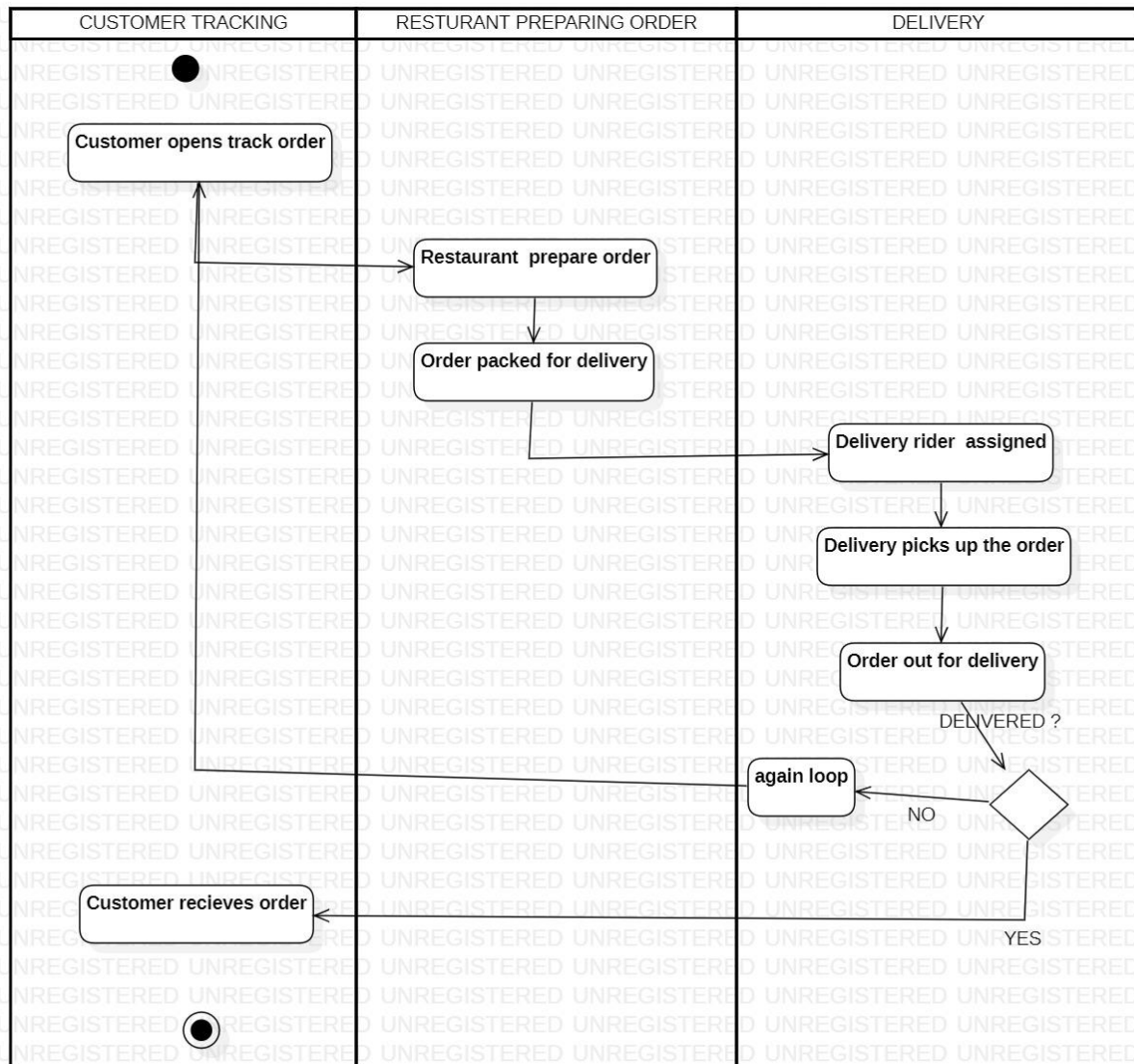
b) Activity Diagram — Track Order



Flow should reflect:

- Customer checks order status.
- Restaurant prepares the meal.
- Delivery personnel receives the order and delivers it to the customer.
- Customer can track location and status in real-time.

#### Question #04: Collaboration Diagram



Suggested nodes:

1. Restaurant system selects an available delivery rider.
2. Rider receives order details.
3. Delivery status updates are pushed to customer until completion.