



National University
Of Computer and Emerging Sciences

Spring-2025

Assignment# 03

Max. Marks: 10

Course Code: SE-4001

Course Title: Software Re-Engineering

Question: Scenario for Sequence Diagram- Online Food Delivery System:

In this scenario, the customer begins by launching the mobile app for an online food delivery service. The app prompts the user to log in or create an account. After entering the correct credentials, the customer successfully logs in, and the app loads the home screen, displaying a list of restaurants based on the user's location. The customer browses through available options, using filters to narrow down their choices, and adds desired items to the cart. Once ready, the customer proceeds to checkout, confirming their delivery address, selected items, and total cost. The mobile app then sends this order request to the backend server, which verifies the details and forwards the request to the restaurant for confirmation. The restaurant accepts or rejects the order based on item availability, and if accepted, a confirmation is sent back to the app, allowing the customer to proceed with payment. If the restaurant cannot fulfill the order, the app notifies the customer and prompts them to revise their selections.

Next, the customer initiates payment by submitting their payment details through the mobile app. These details are sent securely to the backend server, which in turn communicates with the payment gateway for processing. Once the payment is authorized, the payment gateway sends a confirmation to the backend server, which updates the order status to "Paid." The app notifies the customer that the payment was successful, and the restaurant begins preparing the food.

At this point, the backend server locates an available delivery partner near the restaurant. The request is sent to eligible delivery partners, and one accepts the job. The backend server confirms the assignment to both the restaurant and the delivery partner, and the mobile app provides the customer with an estimated delivery time along with a live tracking feature.

Once the food is prepared, the restaurant marks the order as "Ready for Pickup," and the delivery partner arrives at the restaurant. The delivery partner confirms the pickup, and the backend server updates the order status to "Out for Delivery." The mobile app continuously provides the customer with real-time updates, including the delivery partner's location and any changes in estimated arrival time. After reaching the customer's address, the delivery partner hands over the food, and the customer confirms receipt through the app. The order status is then updated to "Completed."

After the delivery, the mobile app prompts the customer to rate the experience and leave a review. These ratings and reviews are sent to the backend server and stored for future reference, where they may also be displayed to other users. The delivery partner is notified of the completed delivery and can view their earnings for the trip.

Throughout this process, if any issues arise, such as payment failure, the app will prompt the customer to retry the payment or choose a different method. Similarly, if there is a delay in delivery, the customer will receive notifications updating them on the new expected arrival time. If the restaurant cancels the order after accepting it, the app will issue a refund and offer the customer alternative options.

This scenario illustrates the interaction between multiple components— the customer, mobile app, backend server, restaurant, payment gateway, and delivery partner— each contributing to the seamless process of ordering and delivering food via mobile app.

Guidelines:

This scenario can be broken into distinct lifelines in diagram:

- Each actor (Customer, Mobile App, Server, Restaurant, Delivery Partner, and Payment Gateway) gets a lifeline.
- Messages represent actions like "login request", "fetch restaurant data", "submit order", "authorize payment", "assign delivery", "pickup confirmation", and "delivery confirmation".
- You can add alt, opt, and loop fragments to represent conditional flows (e.g., payment success/failure, reassign delivery).

Good Luck!