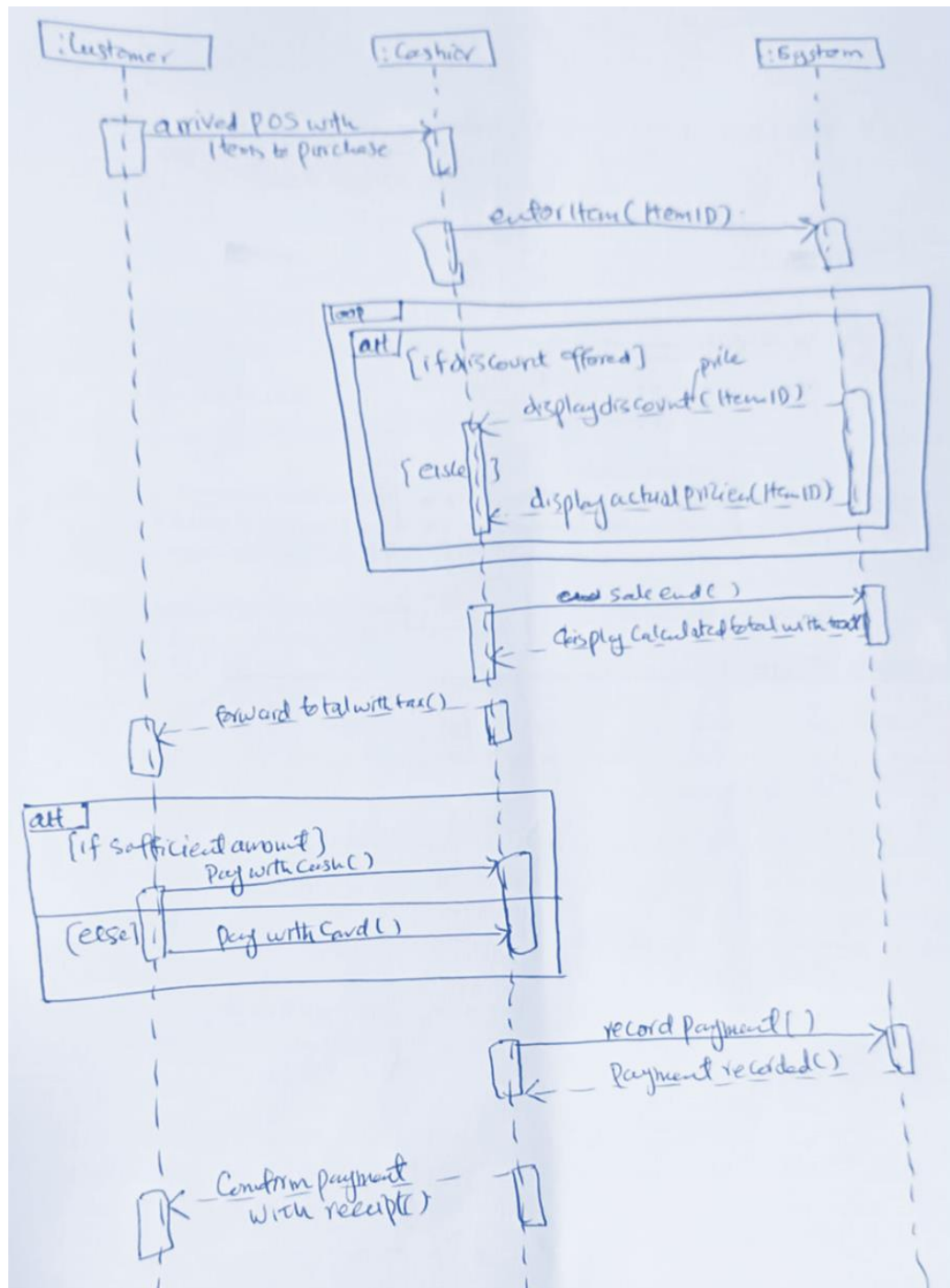| Course Code: CS3004	Course Name: Software Design & Analysis
Instructor Name: Dr. Syed Muazzam Ali Shah	
Student Roll No:	Section No:

Time: 25 minutes.   **Max Marks**: 10 Points

Consider the following scenario and create a comprehensive sequence diagram. You need to create three objects, including Customer, Cashier, and System, and draw proper lifelines with activation boxes. You need to clearly show the message passing and the acknowledgment from one object to another.

1) The Customer arrives at the point of sale (POS) checkout with goods to purchase (to initiate a new sale).

2) The cashier enters the item's identifier or scans the item's bar code.

3) If a discount is offered for the items, then the System records the sale line and displays the discount price for each price and item description.

4) If there is no discount offered for the items, then the System records the sale line and displays the actual price for each price and the item description.

5) Cashier repeats steps 2-4 until indicates done.

6) System presents the total with taxes calculated.

7) The cashier tells the Customer the total and asks for payment.

8) If the Customer has sufficient amount, then the Customer pays the amount (in cash) and System handles the payment.

9) If the customer does not have enough cash on hand, he or she can make the payment with a debit or credit card.

10) Finally, the system confirms the payment.

Solution:

A UML sequence diagram with three lifelines: `:Customer`, `:Cashier`, and `:System`.

- Customer → Cashier: arrived POS with items to purchase
- Cashier → System: enter Item (ItemID)
- **loop**
  - **alt** [if discount offered] price
    - System → Cashier: display discount (Item ID)
  - [else]
    - System → Cashier: display actual price (Item ID)
- Cashier → System: sale end ()
- System → Cashier: display calculated total with tax
- Cashier → Customer: forward total with tax()
- **alt** [if sufficient amount]
  - Customer → Cashier: Pay with Cash ()
  - [else]
  - Customer → Cashier: Pay with Card ()
- Cashier → System: record Payment ()
- System → Cashier: Payment recorded()
- Cashier → Customer: Confirm payment with receipt()

Question 2: Draw an Activity diagram for the following Online Railway Reservation System using Swim-lane showing activities of Customer and System

Railway Reservation System is a system used for booking tickets over internet. Any Customer can book tickets for different trains by logging into system through Gmail or Facebook. System authenticates the customer within 3 seconds if authentication fails go back to login. Customer can book single/multiple ticket/s only if the tickets are available.

Customer searches for the availability of tickets and System checks the availability of the tickets, if tickets are unavailable go back to search and if the tickets are available he books the tickets by initially filling reservation details in a form. System verifies the reservation details filled by the customer if details are incorrect go back to form filling activity. System provides two options to book the tickets by i-ticket or by e-ticket booking.

In both cases, customer can book the tickets online. System will calculate bill after booking. Customer can pay through credit/debit card, system will check the card details to book the ticket, if card details are not correct transaction is failed and tickets are not booked and activity ends else tickets will be booked successfully. After successful booking customer can just logout or chose to cancel the booking. Booking cancellation of both types of tickets can be done online.

Solution:

The diagram is a swimlane activity diagram with two swimlanes: **User** and **Database**.

- Start node → **Login**
- Login → **Authentication** (Database)
- Authentication → decision diamond
  - **Invalid User** → back to Login
  - **Valid User** → **Show Menu** (Database)
- Show Menu → decision diamond
  - **Cancel** → Cancel Ticket
  - **Book** → **Search Train**
- Search Train → **Availability** (Database)
- Availability → decision diamond
  - **Available** → **Book**
  - **Not Available** → back to Show Menu
- Book → **Pay**
- Pay → **Create Ticket** (Database)
- **Cancel Ticket** → **Refund** (Database)
- Cancel Ticket → **Logout**
- Refund → Logout
- Logout → End node

Handwritten annotations: "3 Sessions" (top right), "interruption edge" (left side).