Dhirubhai Ambani
Institute of Information and Communication Technology

Lab_06 – Point Of Sale System

IT-314 Software Engineering

Jal Dani

202201315

Task-1

Use Case: Process Sale

Primary Actor:

Cashier

Stakeholders and Interests:

- **Cashier:** Completes the sale transaction efficiently.
- **Customer:** Wants to purchase goods quickly and correctly.
- **Catalog System:** Provides product information (name, price).
- **Inventory System:** Updates stock after the sale.

Preconditions:

- Cashier must be logged into the POS system.
- Goods to be purchased must have barcodes.

Postconditions:

- Sale transaction is recorded.
- Stock levels in the inventory system are updated.
- Payment is processed, and a receipt is printed.

Main Success Scenario:

1. The cashier scans the barcode of each item.
2. The POS system retrieves the product information from the **Catalog System**.
3. The POS system updates the stock in the **Inventory System**.
4. The cashier selects the payment method (cash, credit card).
5. The POS system processes the payment.
6. Upon successful payment, the POS system prints the receipt.

Extensions:

- 1a. If product's Expiry date passed out.
    - The product then should replace.
- 4a. The customer presents a gift coupon:
    - The cashier enters the coupon details.
    - The system applies the discount to the sale.
- 4b. Payment fails:
    - The cashier reattempts payment or asks the customer for another payment method.

Special Requirements:

- The system must interface with the **Catalog** and **Inventory Systems** for real-time updates.
- The POS system should ensure security for payment transactions.

Use Case: Handle Return

Primary Actor:

Cashier

Stakeholders and Interests:

- **Cashier:** Completes the return transaction efficiently.
- **Customer:** Wants a refund or exchange for returned goods.
- **Inventory System:** Updates stock levels after the return.

Preconditions:

- The customer must present the receipt or proof of purchase.
- Cashier must be logged into the POS system.

Postconditions:

- The return is recorded.
- Stock levels in the inventory system are updated (if the item is returned in physical condition).

Main Success Scenario:

1. The customer presents the goods to be returned.
2. The cashier verifies the receipt or proof of purchase.
3. The POS system retrieves the sale transaction details.
4. The cashier processes the return in the POS system.

5. The system updates the stock levels in the **Inventory System**.
6. The customer is refunded or given store credit.

Extensions:

- 2a. The receipt is not available:
    - The cashier asks the customer for additional information or denies the return based on store policy.
- 2b. If the return date is passed out:
    - The cashier then refused to take the things to return.
- 6a. The goods are damaged:
    - The cashier follows the store's return policy for damaged goods, possibly issuing partial credit or denying the return.

Special Requirements:

- The system must allow verification of the original sale transaction.
- Secure handling of refunds, especially for credit card transactions.

Task-2

1. **Entity Objects**

Sale	• Represents a sale transaction, including items purchased, their quantities, and total amounts.
Product	• Represents a single product, including its barcode, name, price, and stock level.
Customer	• Represents the customer who is buying or returning goods .
Inventory	• Represents the stock levels for products in the inventory.
Gift Coupon	• Represents a discount or promotion that can be applied to a sale.
Payment	• Represents a payment transaction, including payment method (cash, credit, etc.).
Receipt	• Represents the printed receipt after the sale is complete.
User (Cashier/Admin)	• Represents the user logged into the system with different privileges.
Return	• Represents the return transaction for products.

2. Boundary Objects:

POS Interface:	• The user interface through which the cashier interacts with the system.
Payment Terminal:	• Interface for handling payment methods such as cash, credit card, or gift coupons.
Catalog System Interface:	• Interface used by the POS system to retrieve product details (name, price) from the catalog.
Inventory System Interface:	• Interface used by the POS system to update stock levels and verify product availability.
Receipt Printer /Bar code scanner:	• Interface that prints the receipt for the customer after a successful transaction and scan items using scanner.

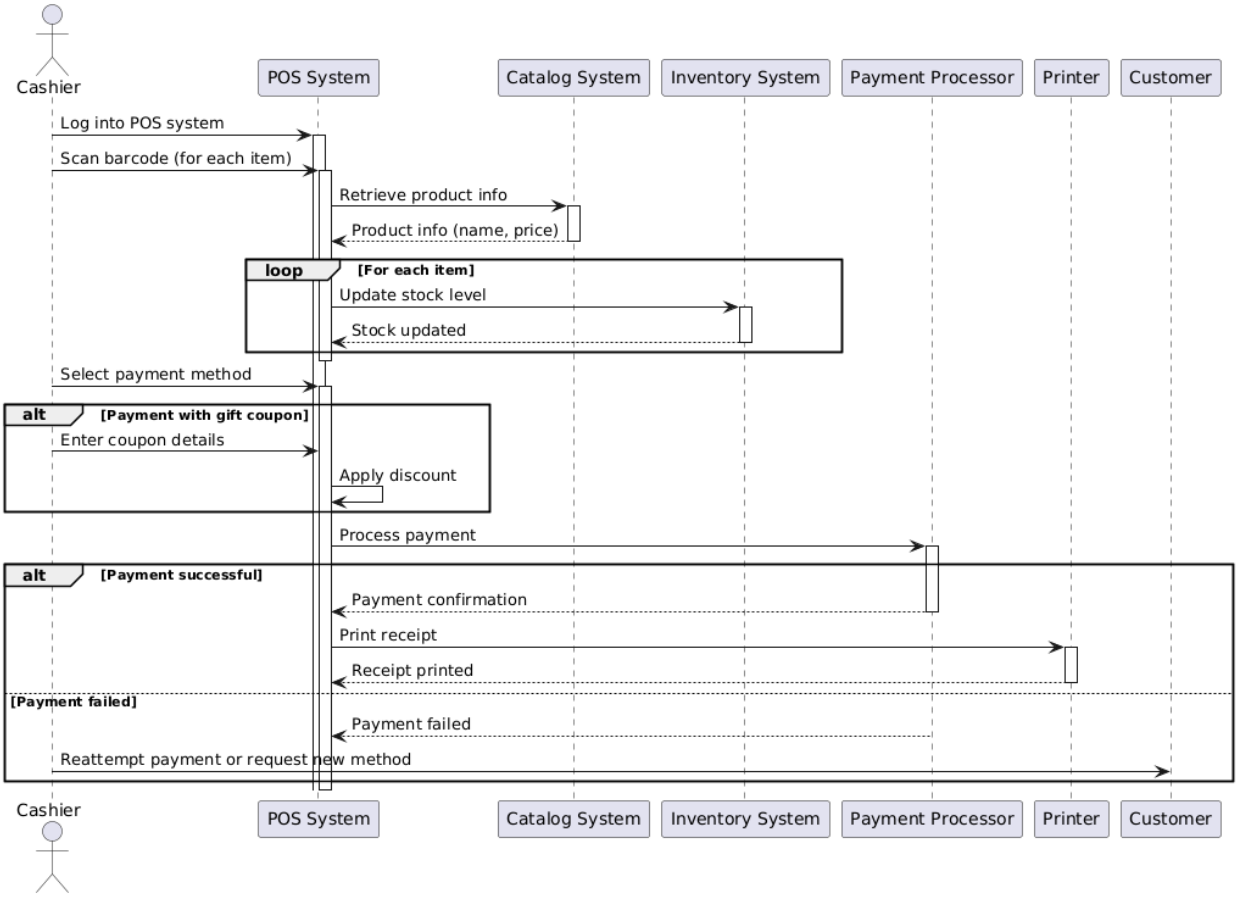
3. Control Objects:

• **Sale Controller**:	• Manages the sale process. Handles interactions between the cashier, product catalog, inventory system, and payment processing. Ensures stock levels are updated, and receipts are printed.

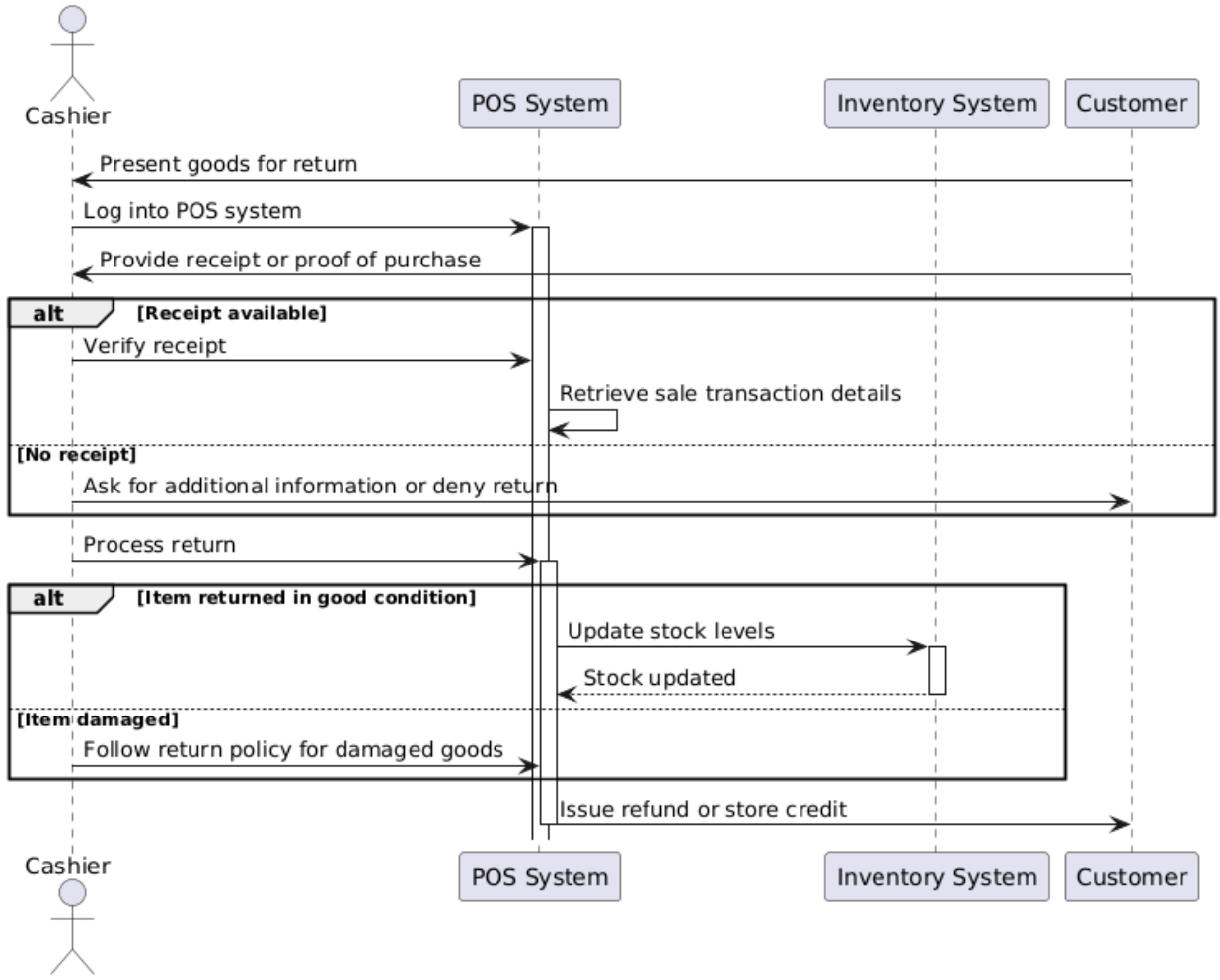
Payment Controller:	• Handles payment processing. It interacts with the payment terminal and ensures the correct amount is processed based on the selected payment method (cash, credit card, or gift coupon).
Return Controller:	• Manages the return process. Validates the receipt, updates stock levels, and processes refunds or exchanges.
User Authentication Controller:	• Ensures that users (cashiers/admins) are logged into the system and have appropriate access rights.
Inventory Controller:	• Handles interactions with the Inventory System, ensuring stock levels are updated in real time after sales or returns.

Sequence Diagram:

Process sale:



Handle Return:



Analysis Domain Model:

Classes and Attributes:

For Process of sale (POS)

1. **Sale**
    a. `saleID`: Unique identifier for the sale transaction.
    b. `date`: Date of the sale.
    c. `totalAmount`: Total amount for the sale.
    d. `status`: Completed, Pending, etc.

2. **Product**
    a. `productID`: Unique identifier for the product.
    b. `name`: Name of the product.
    c. `price`: Price of the product.
    d. `quantity`: Quantity being purchased.
    e. `stockLevel`: Current stock level in the inventory.

3. **Customer**
    a. `customerID`: Unique identifier for the customer.
    b. `name`: Name of the customer.

4. **Cashier**
    a. `employeeID`: Unique identifier for the cashier.
    b. `name`: Cashier's name.

5. **Payment**
    a. `paymentID`: Unique identifier for the payment transaction.
    b. `method`: Cash, Credit Card, Gift Coupon.
    c. `amount`: Amount paid.
6. **Receipt**
    a. `receiptID`: Unique identifier for the receipt.
    b. `saleID`: Linked to the corresponding sale.
    c. `printDate`: Date the receipt was printed.

7. **Inventory**

a. `inventoryID`: Unique identifier for the inventory record.
b. `stockLevel`: Current stock of a product.

Classes and Attributes:

For Handle Return:

1. **Return**
    a. `returnID`: Unique identifier for the return transaction.
    b. `returnDate`: Date the return was processed.
    c. `reason`: Reason for return (e.g., damaged item, wrong product).
    d. `status`: Status of the return (Approved, Denied, Pending).

2. **Sale**
    a. `saleID`: Unique identifier for the original sale transaction.
    b. `date`: Date of the original sale.
    c. `totalAmount`: The total sale amount.
    d. `receiptAvailable`: Boolean indicating whether the receipt is available.

3. **Product**
    a. `productID`: Unique identifier for the product being returned.
    b. `name`: Name of the product.
    c. `price`: Price of the product.
    d. `quantity`: Quantity being returned.
    e. `condition`: Condition of the product being returned (Good, Damaged, etc.).

4. **Customer**
    a. `customerID`: Unique identifier for the customer returning the item.
    b. `name`: Name of the customer.

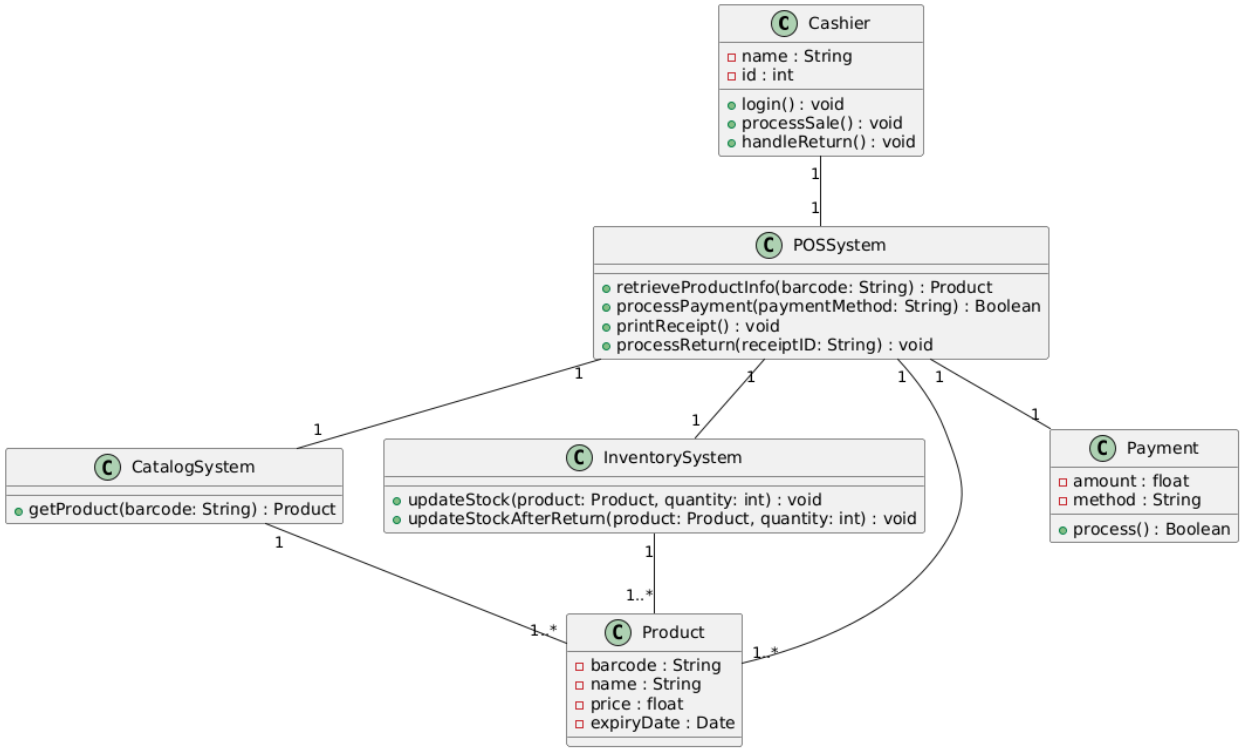
5. **Cashier**

a. `employeeID`: Unique identifier for the cashier handling the return.
b. `name`: Cashier's name.

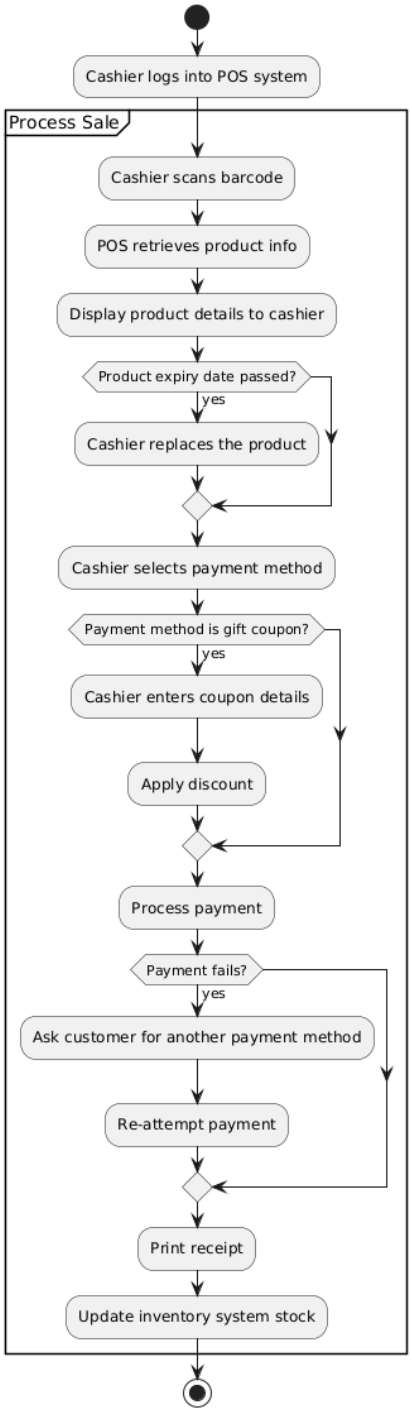
6. **Refund**
    a. `refundID`: Unique identifier for the refund transaction.
    b. `amount`: The refund amount.
    c. `refundMethod`: The method of refund (Cash, Store Credit, etc.).

7. **Inventory**
    a. `inventoryID`: Unique identifier for the inventory record.
    b. `stockLevel`: Current stock level after the return.
    c. `updatedStockLevel`: Stock level after updating due to returned items.

Class diagram for this Use Cases:



Activity Diagram for POS:



Activity diagram for Handle Return: