Dhirubhai Ambani Institute of Information and Communication Technology

IT314 – Software Engineering

Lab 6 Report

Modeling Class Diagram and Activity Diagram (Point of Sale System):

Ayush Chaudhari -202201517

Task-1 Develop Use Case Textual Description for "Process Sale" and "Handle Return" use cases

Use Case 1: Process Sale

- **Use Case Name**: Process Sale
- **Actors**:
    - **Primary**: Cashier, Customer
    - **Secondary**: Inventory System, Catalog System, Payment System, Receipt Printer
- **Preconditions**:
    1. The cashier must be logged into the POS system.
    2. The item(s) being purchased must be available in stock.
    3. The customer must provide a valid payment method.
- **Postconditions**:
    1. The sale is recorded in the system.
    2. The stock is updated.
    3. A receipt is printed and provided to the customer.

Basic Flow:

1. **Start Transaction**: The cashier initiates a new sale transaction.
2. **Scan Items**:
    a. The cashier scans each item's barcode.
    b. The system retrieves the item's name and price from the **Catalog System**.
    c. The system checks the item's stock in the **Inventory System** and deducts the available stock for each item.
3. **Apply Discounts**:
    a. The cashier applies any available coupons or discounts.
    b. The total amount is updated in the system.
4. **Payment**:
    a. The customer chooses a payment method (cash, card, or check).
    b. The cashier processes the payment through the **Payment System**.
    c. If the payment is successful, the transaction is finalized.
5. **Print Receipt**:
    a. The system prints the receipt using the **Receipt Printer**.
    b. The cashier provides the receipt to the customer.

6. **End Transaction**: The cashier concludes the sale transaction, and the system records it.

Alternate Flows:

- **Flow 2a: Item Not in Stock** (Occurs at Step 2):
  - **Trigger**: The cashier scans an item, but it is out of stock.
  - **Steps**:
    1. The POS system notifies the cashier that the item is not available.
    2. The cashier informs the customer, who may choose to either remove the item from the purchase or replace it with an alternative.
    3. The cashier continues scanning any remaining items.
- **Flow 4a: Payment Declined** (Occurs at Step 4):
  - **Trigger**: The customer's payment method is declined by the payment system.
  - **Steps**:
    1. The **Payment System** notifies the cashier that the payment was unsuccessful.
    2. The cashier informs the customer and asks them to provide an alternative payment method (e.g., a different card or cash).
    3. The transaction continues once a successful payment is made.
    4. If no valid payment method is available, the transaction is canceled.
- **Flow 3a: Coupon/Discount Invalid** (Occurs at Step 3):
  - **Trigger**: The customer presents an invalid or expired coupon/discount.
  - **Steps**:
    1. The POS system alerts the cashier that the coupon or discount is invalid.
    2. The cashier informs the customer, and the coupon is not applied to the total.
    3. The customer either agrees to proceed without the discount or cancels the transaction.

Use Case 2: Handle Return

- **Use Case Name**: Handle Return

- **Actors**:
  - **Primary**: Cashier, Customer
  - **Secondary**: Inventory System, Catalog System, Payment System
- **Preconditions**:
  1. The cashier must be logged into the POS system.
  2. The item must have been purchased previously, and the return must be within the allowable return window.
- **Postconditions**:
  1. The return is recorded in the system.
  2. The stock is updated to reflect the returned item.
  3. A return receipt is printed and provided to the customer.

Basic Flow:

1. **Start Return Transaction**: The cashier starts a new return transaction in the POS system.
2. **Scan Item**:
   a. The cashier scans the barcode of the item being returned.
   b. The system retrieves the item's details and original purchase information from the **Catalog System**.
3. **Verify Purchase**:
   a. The system verifies if the item was purchased within the return period and is eligible for return.
4. **Update Stock**:
   a. If the return is eligible, the system updates the stock in the **Inventory System** to reflect the returned item.
5. **Process Refund**:
   a. The cashier processes the refund through the **Payment System**.
   b. The customer is refunded based on the original payment method (cash, card, or check).
6. **Print Return Receipt**:
   a. The system prints a return receipt.
   b. The cashier provides the return receipt to the customer.
7. **End Transaction**: The cashier ends the return transaction, and the system records the return.

Alternate Flows:

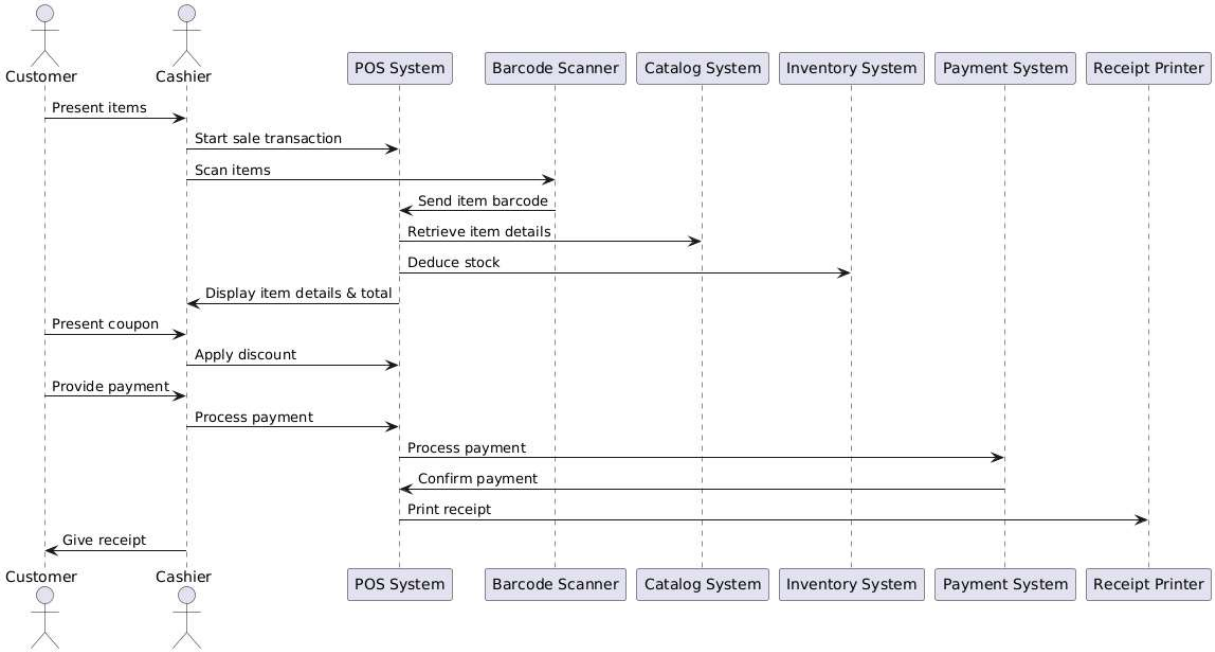
- **Flow 3a: Item Not Eligible for Return** (Occurs at Step 3):
    - **Trigger**: The cashier scans an item that is not eligible for return (e.g., past the return window, damaged, or not purchased from the store).
    - **Steps**:
        1. The system alerts the cashier that the item is ineligible for return.
        2. The cashier informs the customer and explains the reason for ineligibility.
        3. The customer may either accept the decision or escalate to the store manager for further assistance.
- **Flow 5a: Refund Failed** (Occurs at Step 5):
    - **Trigger**: The refund process fails (e.g., technical issue with the payment system or invalid card).
    - **Steps**:
        1. The system alerts the cashier that the refund was unsuccessful.
        2. The cashier informs the customer and requests an alternative refund method (e.g., cash or different card).
        3. If no alternative refund method is available, the issue is escalated to a manager for further resolution.
- **Flow 2a: Partial Return** (Occurs at Step 2):
    - **Trigger**: The customer wishes to return only some items from a previous purchase.
    - **Steps**:
        1. The cashier scans the items the customer wants to return.
        2. The system processes the return for only the selected items.
        3. The refund is calculated and processed accordingly.
        4. The system prints a return receipt only for the returned items.

Task -2 Identify Entity/Boundary Control Objects

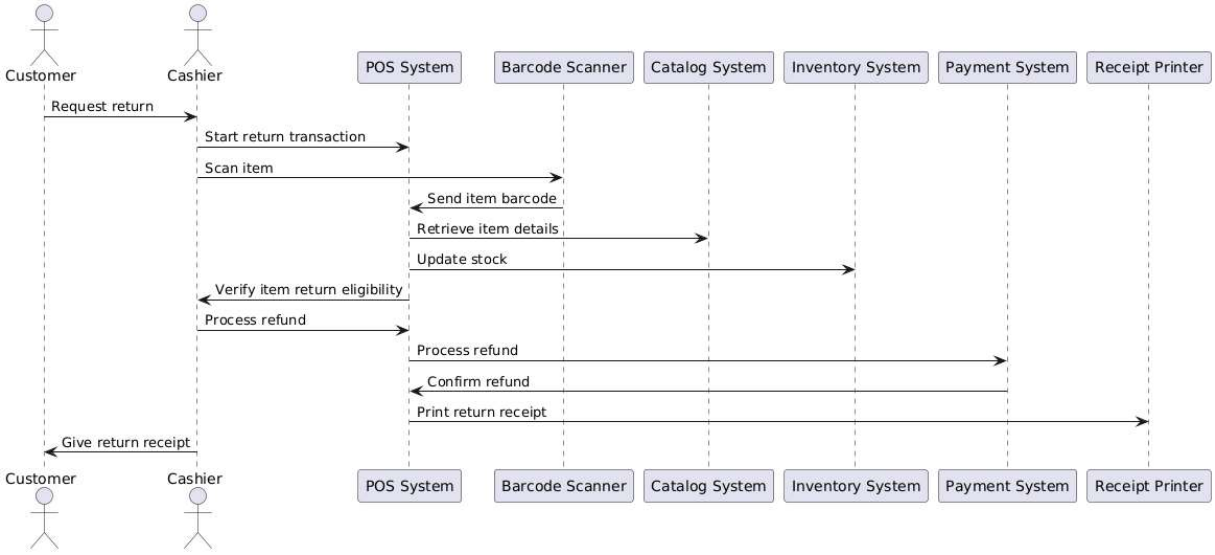
- **Entity Objects**:
    1. Product
    2. SaleTransaction
    3. Payment
    4. Coupon
    5. Inventory
    6. Receipt
- **Boundary Objects**:
    1. Sales Screen
    2. Barcode Scanner Interface
    3. Payment Screen
    4. Receipt Printer Interface
    5. Catalog System API
    6. Inventory System API
- **Control Objects**:
    1. SaleController
    2. PaymentController
    3. ReceiptController
    4. InventoryController
    5. CouponController

Task-3 Develop Sequence Diagrams

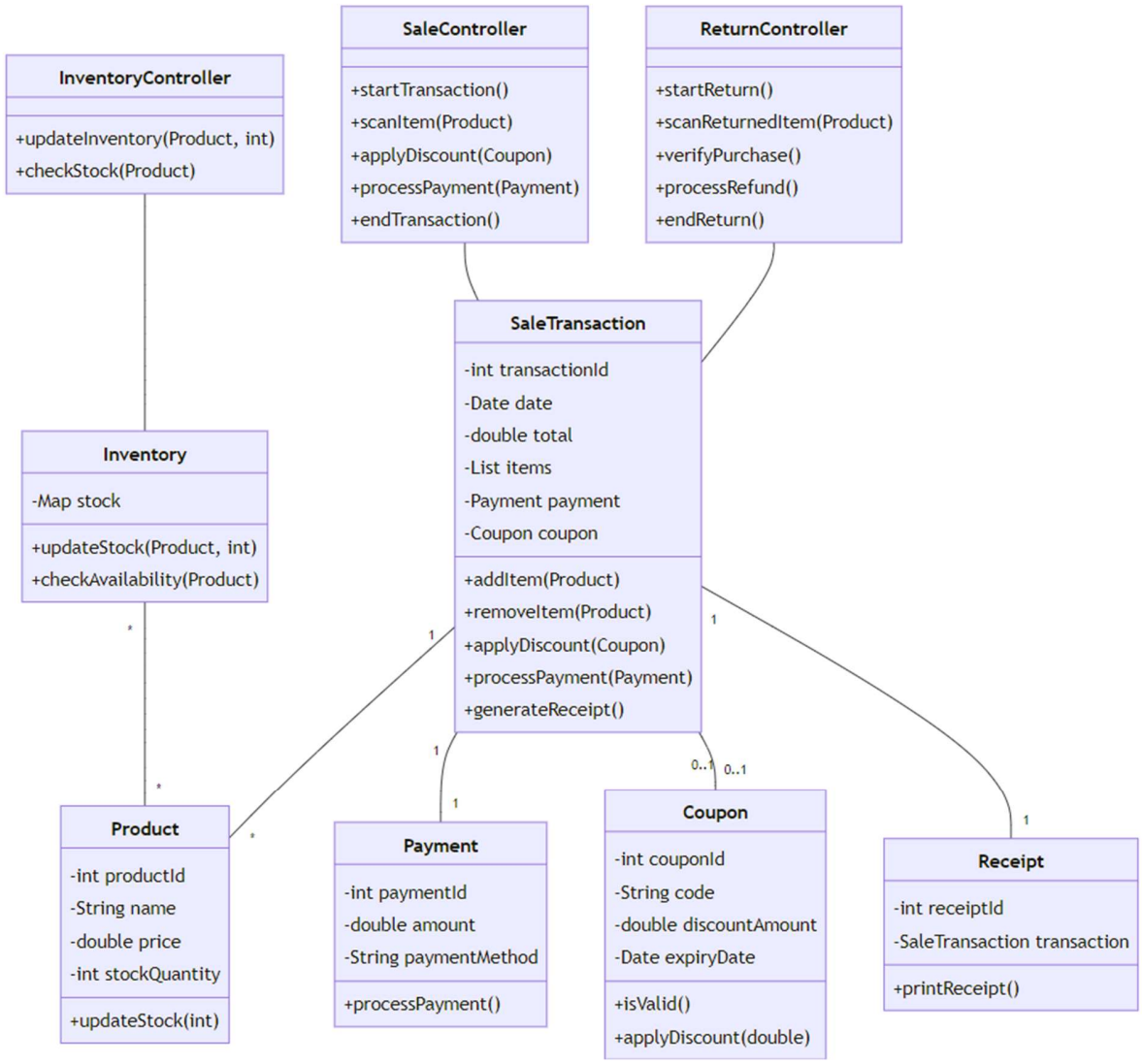
1. Sequence Diagram for Process Sale



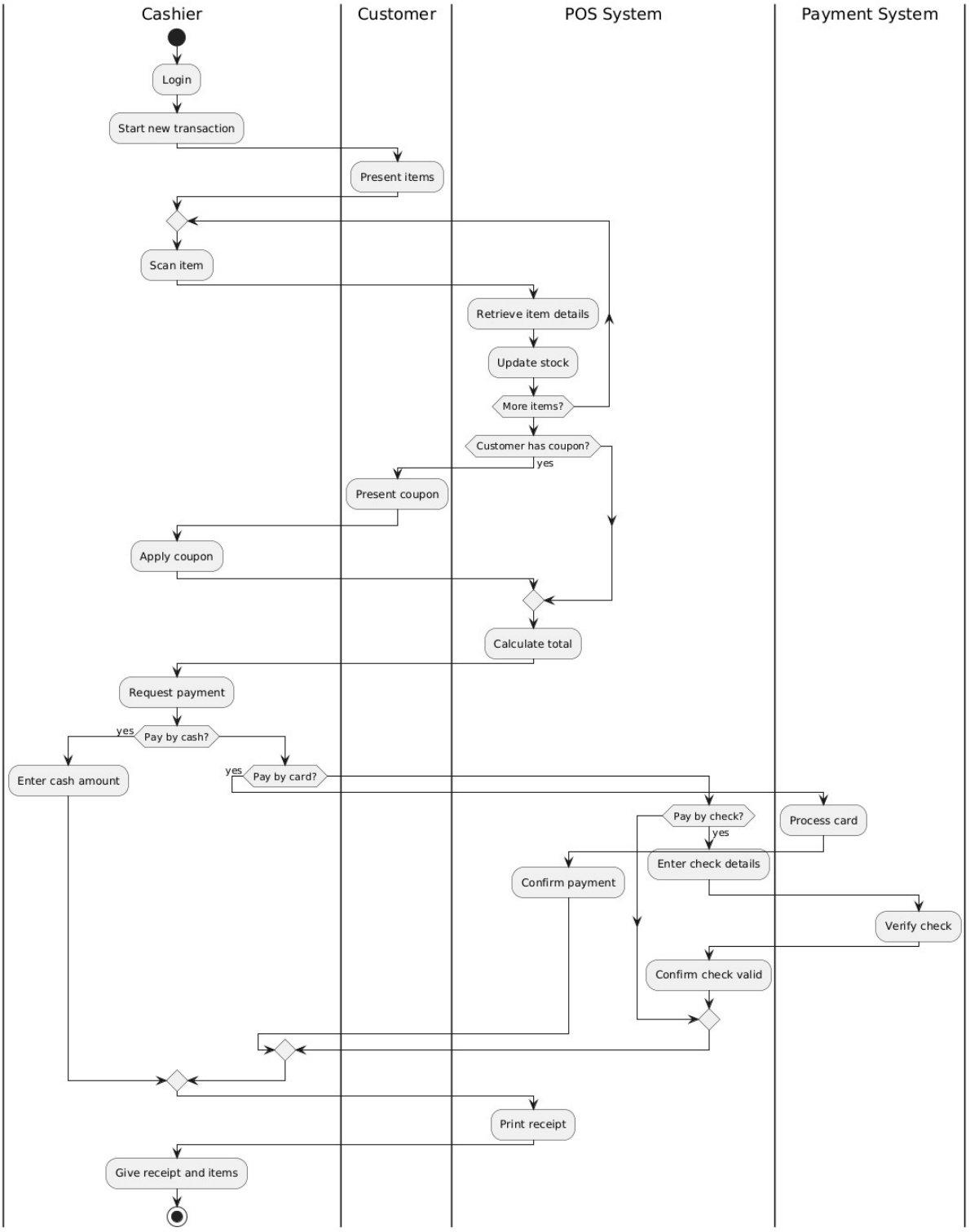
2. Sequence Diagram for Handle Return



Task-4 Develop Analysis Domain Models



Task-5 Develop activity diagram for "Process Sale" and "Handle Return" use cases.



Activity Diagram for Handle Return Case