IT313 Software Engineering Lab-6

Roll No: 202201320 Name: Sumit Vishwakarma

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

Use Case: Process Sale

1. **Primary Actor:**
   - Cashier
2. **Stakeholders:**
   - **Customer:** Wants to purchase goods and pay for it..
   - **Store:** Seeks to complete the sale and update inventory.
   - **Inventory System:** Keeps stock levels correctly.
   - **Catalog System:** Provides product details such as name and price.
   - **Payment Gateway :** Handles payment processing for card or check.
3. **Preconditions:**
   - Cashier must be logged into the POS system.
   - The catalog system is up and running to retrieve product information.
   - The inventory system is operational to manage and update stock information correctly.
   - The payment gateway is available for non-cash/online payments (credit card, UPI).
4. **Postconditions:**
   - The sale is recorded in the system.
   - Inventory is updated to reflect items sold.
   - A receipt is printed for the customer.
   - The customer has successfully paid using their selected method.
5. **Success Scenario:**
   - Cashier logs in to the POS system.
   - Customers arrive with goods to purchase.
   - Cashier scans the barcodes of the goods.
   - The POS retrieves the name and price from the catalog system.
   - The POS updates the inventory system to deduct stock.
   - Customers provide payment via cash, credit card or UPI.
   - POS processes the payment through the appropriate method.
   - A receipt is printed and handed to the customer.
6. **Extensions:**

  - **3a. Barcode cannot be scanned:** Cashier manually enters the product code or product price.
  - **6a. Insufficient stock:** Cashier informs the customer and suggests alternatives or removes the unavailable item from the transaction.
  - **6b. Payment fails:** POS notifies the cashier, and the cashier asks the customer for an alternative payment method, if possible or retry.
  - **8a. Printer fails:** Cashier offers to email or provide a handwritten receipt.

Use Case: Handle Return

1. **Primary Actor:**
   - Cashier
2. **Stakeholders:**
   - **Customer:** Wants to return goods and receive a refund.
   - **Store:** Needs to accept or reject returns and update inventory.
   - **Inventory System:** Manages stock updates when goods are returned.
   - **Payment Gateway:** Handles refund processing for credit card or UPI transactions.
   - **Administrator:** Manages system users and security settings.
3. **Preconditions:**
   - Cashier must be logged into the POS system.
   - Customer provides the original receipt for the goods to be returned.
   - The catalog system is operational to validate the returned items.
   - The inventory system is operational to update stock.
4. **Postconditions:**
   - The returned goods are logged in the system.
   - Inventory is updated to reflect the returned items.
   - A refund is issued to the customer based on the original payment method.
   - If a coupon was used, the system adjusts the refund accordingly.
5. **Success Scenario:**
   - Customer arrives at the POS counter with goods to return and provides the receipt.
   - Cashier scans the receipt to retrieve the original transaction.
   - The POS system verifies that the goods are eligible for return (e.g., within the return period).
   - Cashier scans the barcodes of the returned goods.
   - The POS retrieves product information from the catalog system and updates inventory.
   - POS calculates the refund amount based on the original transaction.
   - The customer receives the refund via the original payment method (cash, card, check).

    - A return receipt is printed and provided to the customer.

6. **Extensions:**
    - **2a. Receipt is missing:** Cashier verifies the original transaction using customer information or another method.
    - **3a. Return period exceeded:** Cashier informs the customer that the item is not eligible for return.
    - **5a. Item damaged or incomplete:** Cashier may offer a partial refund or reject the return based on store policy.
    - **7a. Payment method issue (e.g., card not available):** Cashier may offer store credit as an alternative refund method.

Task 2: Identify Entity/Boundary Control Objects

Entities:

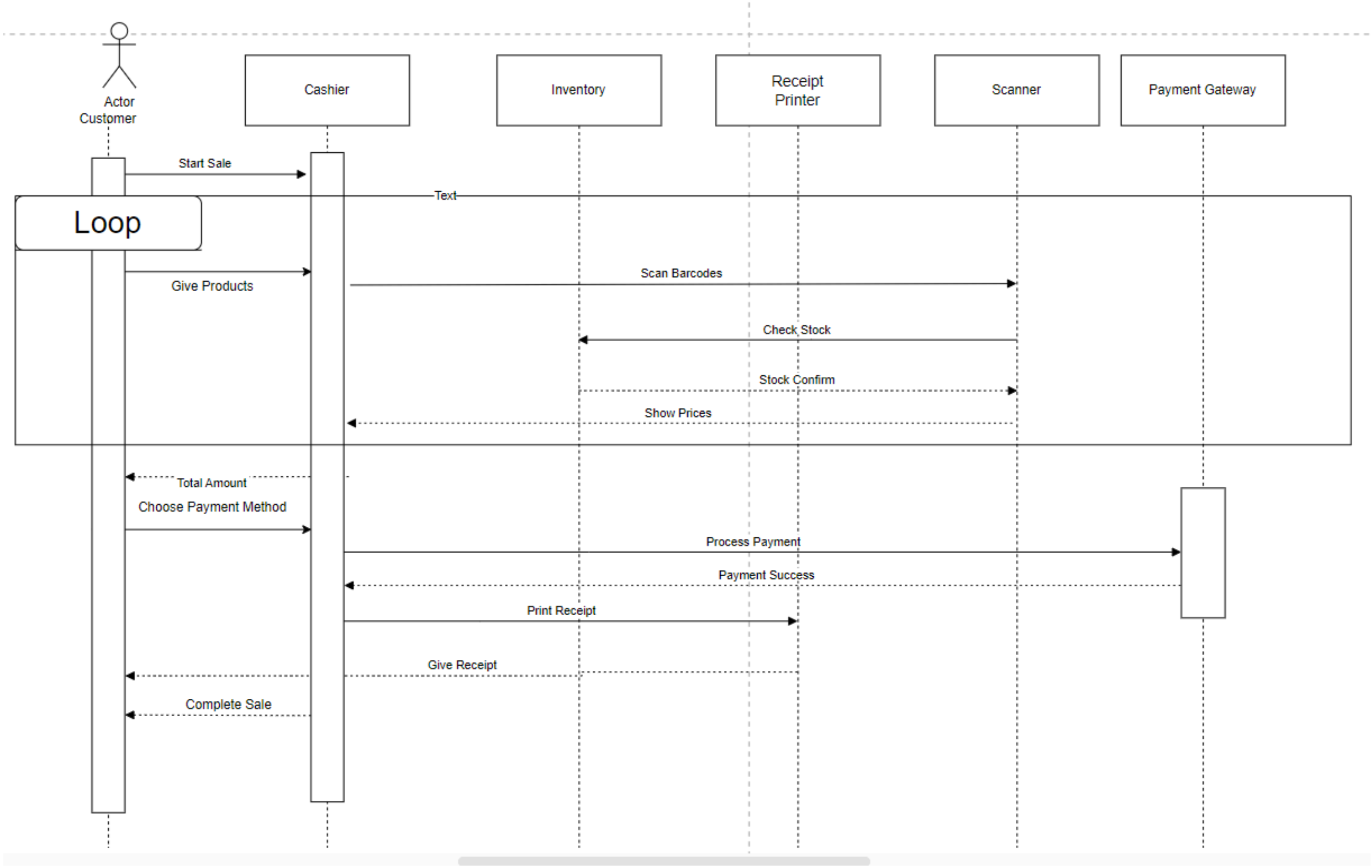
- **Product**: Has information about the goods, including name, price, and stock status.
- **Promotion**: Available gift coupons and discounts.
- **Payment**: Contains details about the transactions, whether it's cash, credit card, or check.
- **Inventory**: Keeps data of the stock levels of each product and updates when items are sold.
- **Customer:** Contains customer details if needed for rewards, coupons, or receipts.
- **Receipt**: Stores details of the products purchased, total amount, applied discounts and gift coupons.
- **Return**: Keeps track of items returned, along with their reasons..
- **Sale:** Stores the details of products sold.
- **Refund:** Stores the details of the refund for the returned items, whether by cash, card.

Boundary Objects:

- **User Interface (UI):** Used by the cashier to interact with the system, scan items, process payments, and print receipts.
- **Payment Gateway**: Interface for processing card or check payments.

- **Receipt Printer**: Outputs the receipt for the customer.
- **Barcode Scanner**: Device that captures the product's barcode for the system to process.

Task 3: Develop Sequence Diagrams:



Task 4: Develop Analysis Domain Models

Analysis domain models generally have information about the classes and their attributes along with their description.

For Process of sale (POS):

Product:

- **Attributes**: `productID`, `name`, `price`, `stockLevel`, `category`

- **Description**: Represents the goods sold in the store.

Promotion:

- **Attributes**: `couponID`, `discountValue`, `validityPeriod`
- **Description**: Represents the gift coupons or promotional offers available to the customer.

Payment:

- **Attributes**: `paymentID`, `amount`, `method`, `status`
- **Description**: Represents the customer's payment for the sale, whether by cash, card, or check.

Inventory:

- **Attributes**: `productID`, `currentStock`
- **Description**: Represents the stock levels of products in the store and handles updates after a sale.

Customer:

- **Attributes**: `customerID`, name, `contactInfo`
- **Description**: Represents the customer if the system needs to associate purchases with a customer account.

For Handle Return:

Product:

- **Attributes:** `productID`, name, `price`, `returnable`
- **Description:** Represents the goods returned by the customer.

Inventory:

- **Attributes**: `productID`, `currentStock`
- **Description**: Updates the stock levels when items are returned.

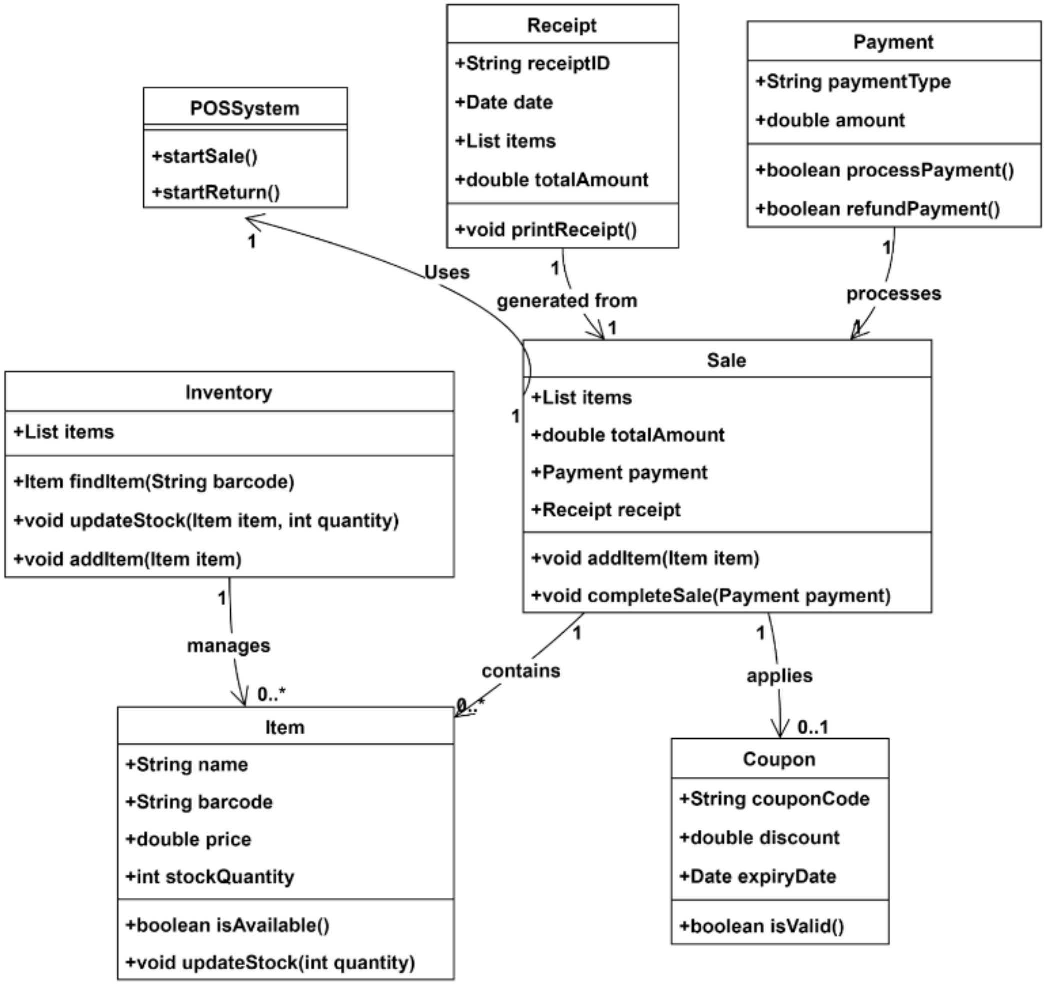
Refund:

- **Attributes**: `refundID`, `amount`, `method`, `status`
- **Description**: Represents the refund for the returned items, whether by cash, card, or store credit.

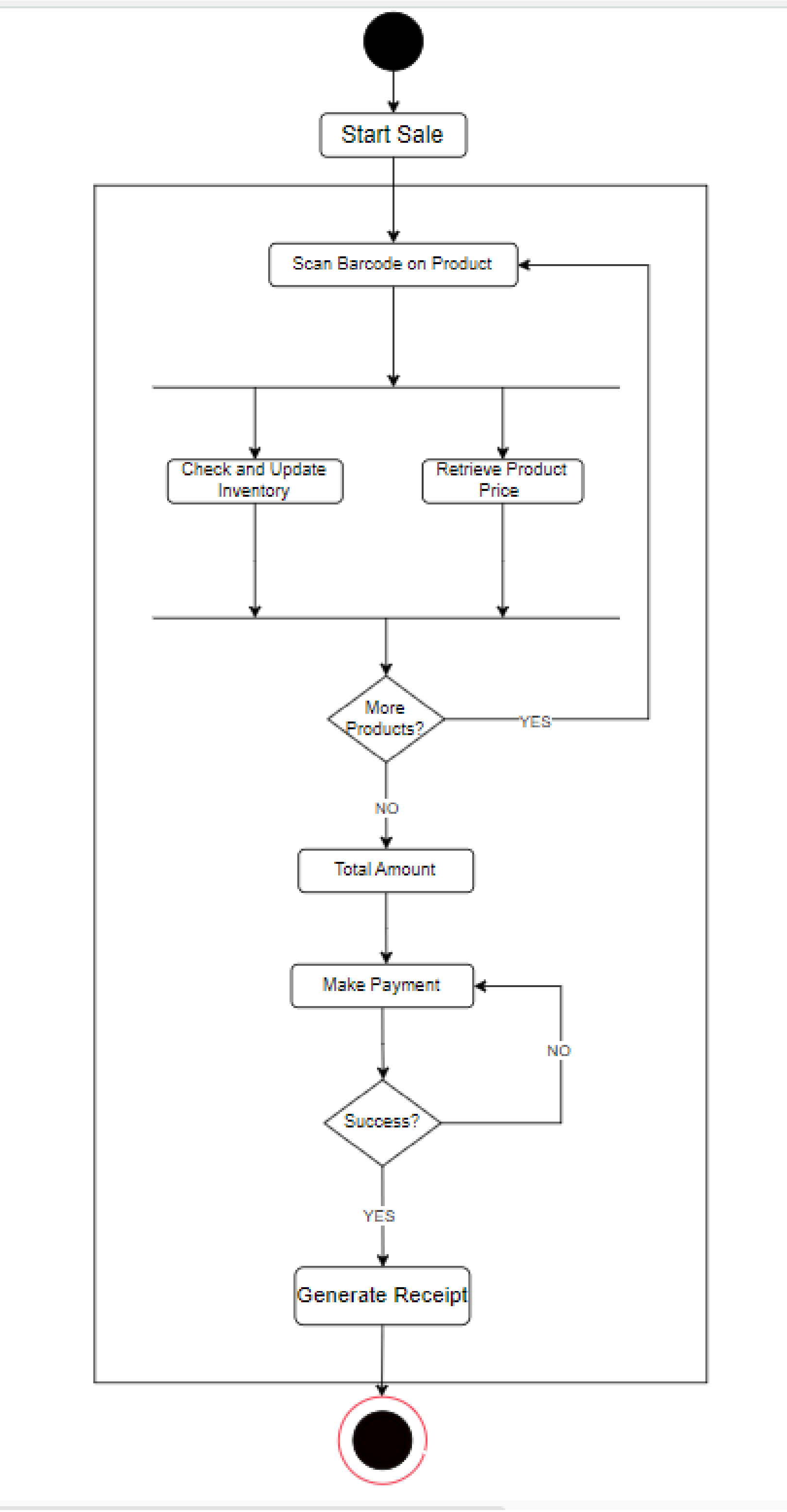
Customer:

- **Attributes**: `customerID`, `name`, `contactInfo`
- **Description**: Represents the customer making the return.

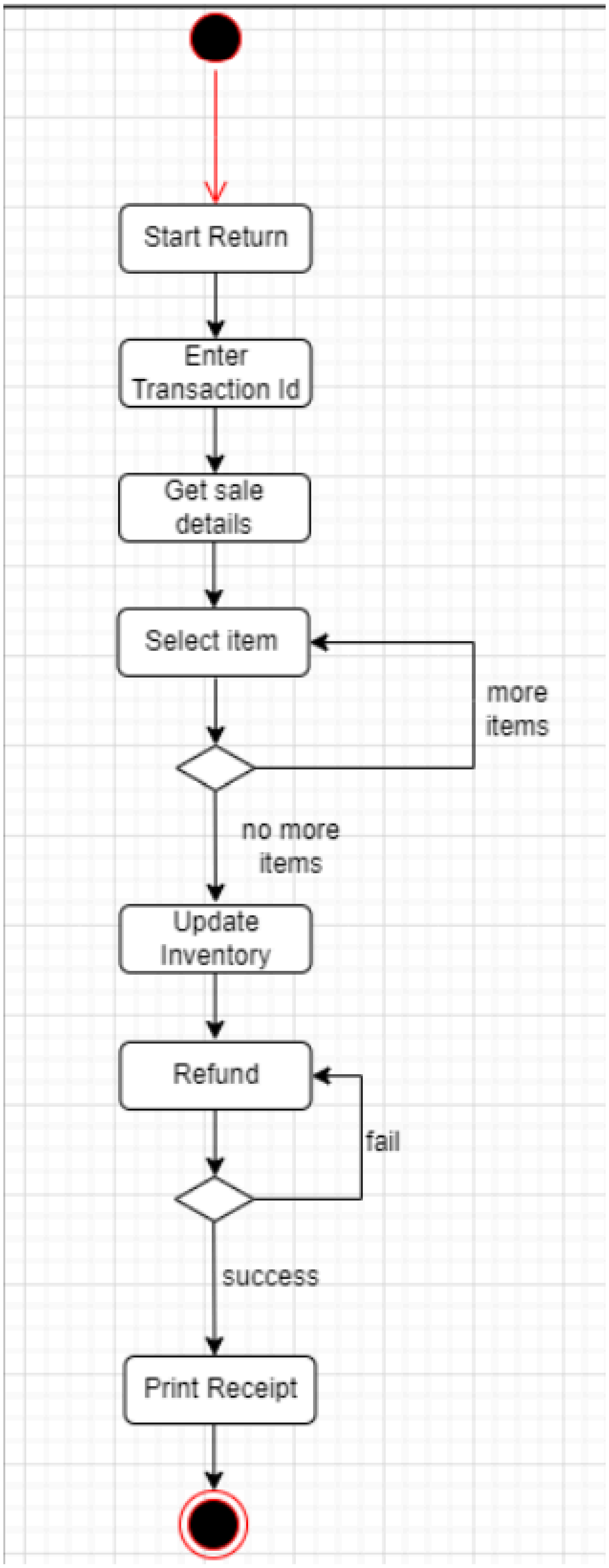
Class Diagram:



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



Handle Returns:


Start Return
Enter Transaction Id
Get sale details
Select item
more items
no more items
Update Inventory
Refund
fail
success
Print Receipt