Dhirubhai Ambani
Institute of Information and Communication Technology

Subject: Software Engineering

Subject code: IT 314

Lab 6

Date of submission: 24-09-2024

Name: Dave Preet A.

ID : 202201072

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

Task-1:

Develop Use Case Textual Description for "Process Sale" and “Handle Files”

Use Case: Process Sale

- **Actor:** Cashier
- **Precondition:**
    - Cashier is logged into the POS system.
    - POS system is operational.

Main Flow:

1. **Start Sale:**
    - Cashier starts a new sale transaction in the POS system.
2. **Scan Items:**
    - **For each item:**
        - Cashier scans the item's barcode.
        - System retrieves item details from the Catalog System.
        - System adds item to the sale.
        - System updates the Inventory System to reflect the reduction in stock**.**
3. **Calculate Total:**
    - System calculates the total amount for the sale.
4. **Apply Coupon (If applicable):**
    - If the customer has a gift coupon:
        - Cashier applies the coupon.
        - System recalculates the total based on the coupon.
5. **Process Payment:**
    - Customer selects a payment method (cash, card, check).
    - Cashier processes the payment in the system.
6. **Successful Payment:**
    - If payment is successful:
        - System prints the receipt.
        - System completes the sale and updates records (inventory, sales).

Alternate Flows:

1. **Invalid Coupon:**
    - If the coupon is invalid:
        - System notifies cashier.
        - Cashier proceeds without the coupon or retries.
2. **Payment Failure:**
    - If payment is unsuccessful:
        - System notifies cashier.
        - Cashier retries payment or chooses an alternative payment method.
        - After multiple failures, system cancels the sale.

Exception Handling:

1. **Barcode Not Recognized**:
   - System prompts cashier to manually enter the barcode if it cannot be scanned.
2. **Out of Stock**:
   - System notifies the cashier if an item is out of stock.
   - Cashier removes the item or offers a substitute.

Postconditions:

- Sale is completed, and a receipt is printed.
- Inventory and sales records are updated in the system.

Use Case: Handle Return

- **Actor**: Cashier
- **Precondition**:
  - Cashier is logged into the POS system.
  - POS system is operational.

Main Flow:

1. **Start Return**:
   - Cashier starts a new return transaction in the POS system.
2. **Scan Item**:
   - Cashier scans the item's barcode.
   - System retrieves item details from the **Catalog System**.
3. **Verify Return Eligibility**:
   - System verifies if the item is eligible for return based on store policies (e.g., return window, condition).
4. **If Item is Eligible**:
   - **a.** System adds the item to the return transaction.
   - **b.** System calculates the refund amount based on the original purchase.
   - **c.** Cashier processes the refund based on the payment method used during the purchase.
   - **d.** If the refund is successful:

        - **i.** System updates the **Inventory System**, increasing the stock.
        - **ii.** System prints a return receipt for the customer.
        - **iii.** System completes and records the return in the sales database.

Alternate Flows:

1. **Item Not Eligible for Return**:
    - If the item is not eligible for return:
        - System notifies the cashier.
        - Cashier informs the customer and cancels the return transaction.
2. **Refund Failure**:
    - If the refund is unsuccessful:
        - System notifies the cashier.
        - Cashier retries the refund or cancels the return.
        - System restores the item to the original state (removes it from return).

Exception Handling:

1. **Barcode Not Recognized**:
    - If the barcode cannot be read:
        - System prompts the cashier to manually enter the barcode.
2. **Out of Stock in System**:
    - If the returned item cannot be updated in the **Inventory System**:
        - System logs an error for later resolution but allows the return to proceed.

Postconditions:

- Return is completed, recorded in the system, and the stock is updated.
- A receipt is provided to the customer.

Task 2 :

Identify Entity/Boundary Control Objects

Entities:

- User
- Sale
- Return
- Item
- Payment
- GiftCoupon

Boundary:

- POSInterface (for cashier interactions)
- AdminInterface (for administrator functions)
- ReceiptPrinter

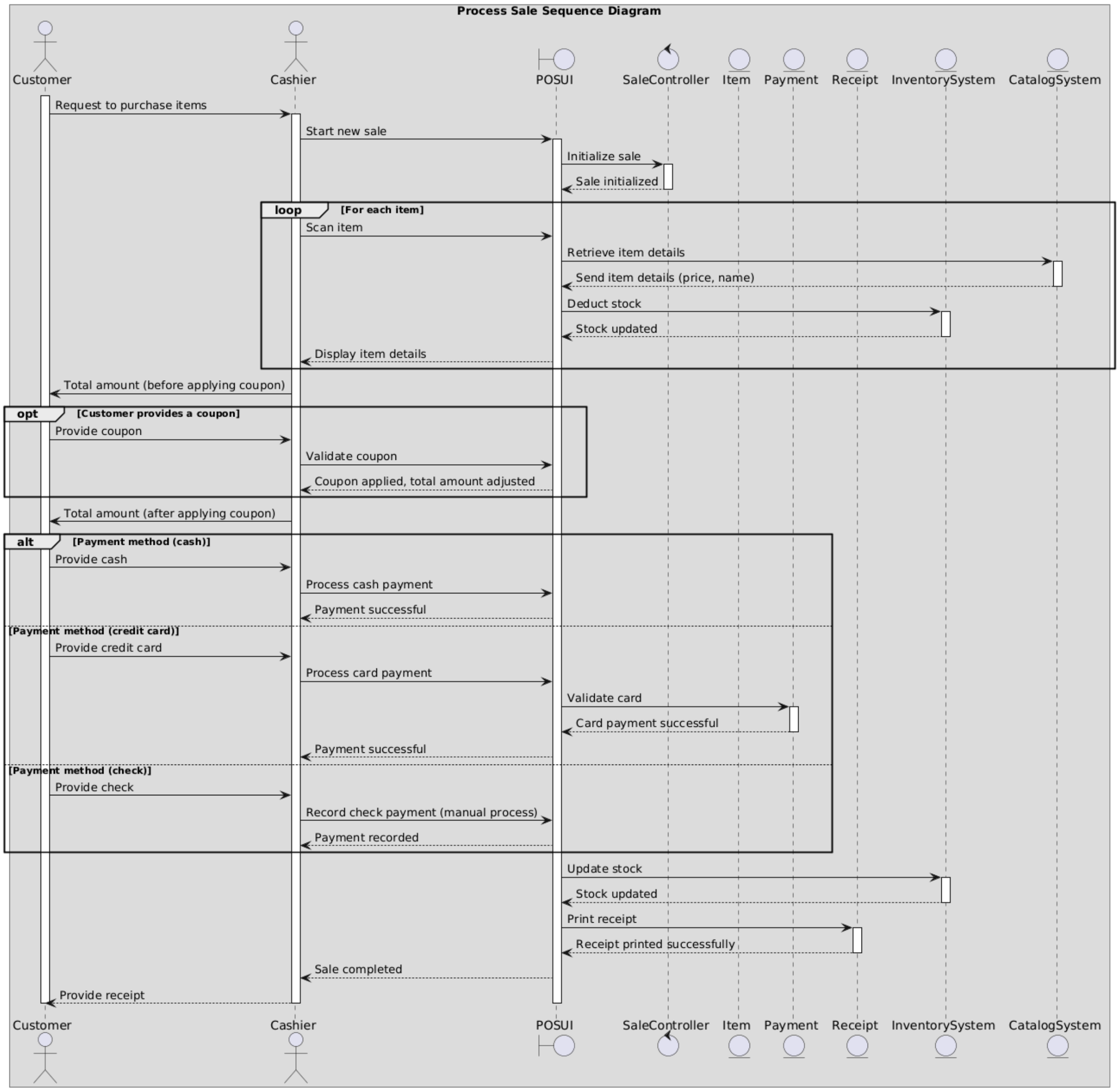
Control:

- SaleController
- ReturnController
- InventoryController
- CatalogController
- PaymentController

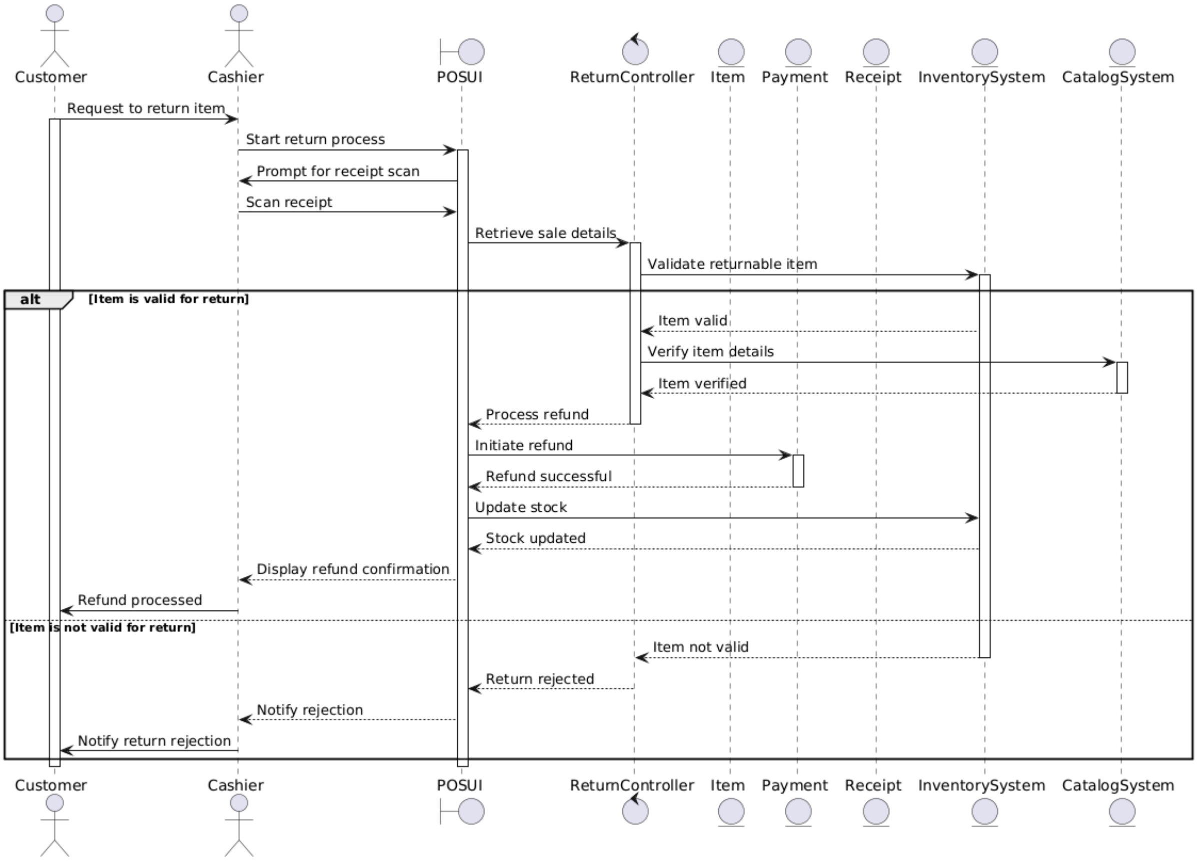
Task 3:

Develop Sequence Diagrams

1) For Process Sale

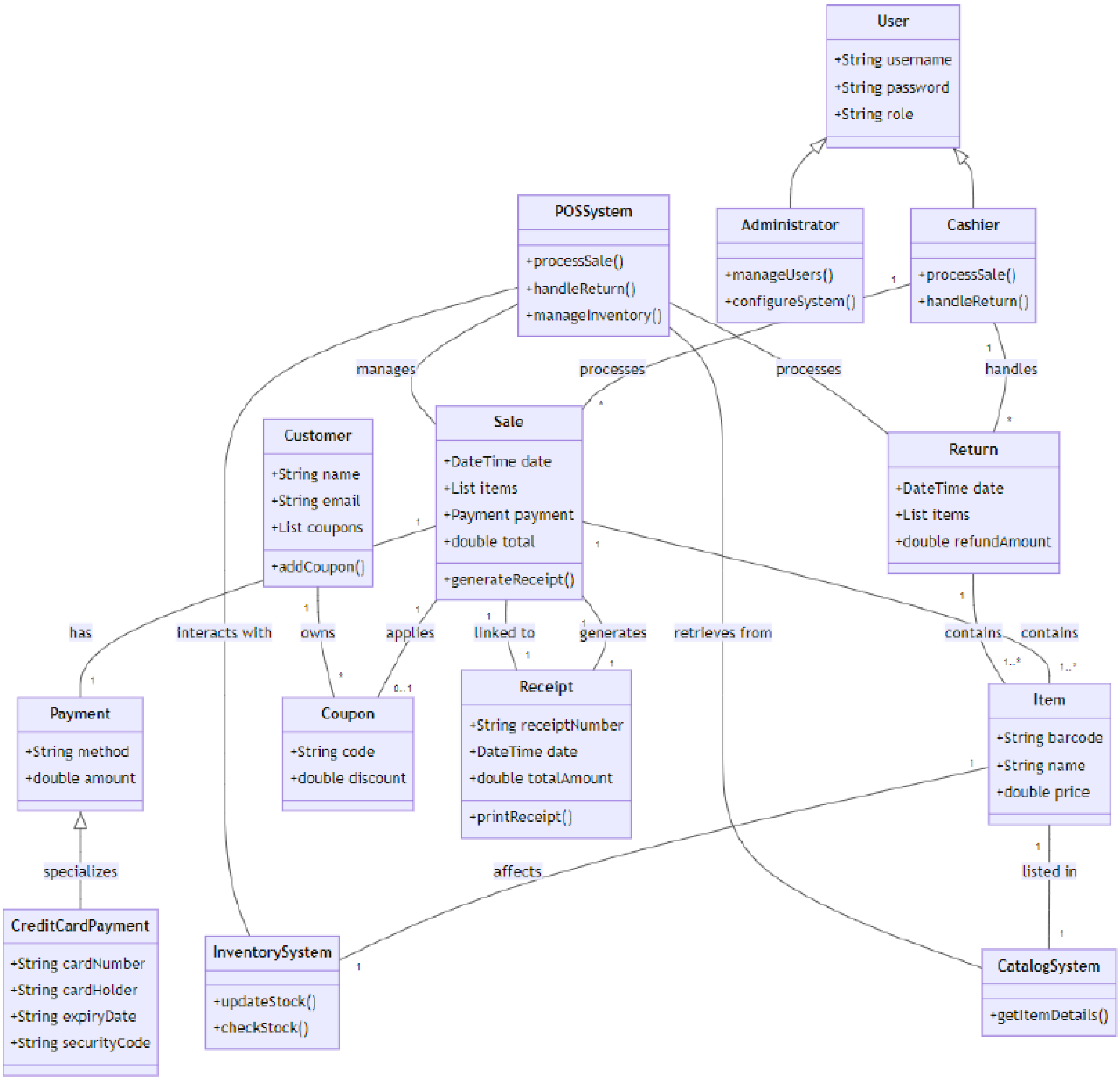


2) Handle Return Sale



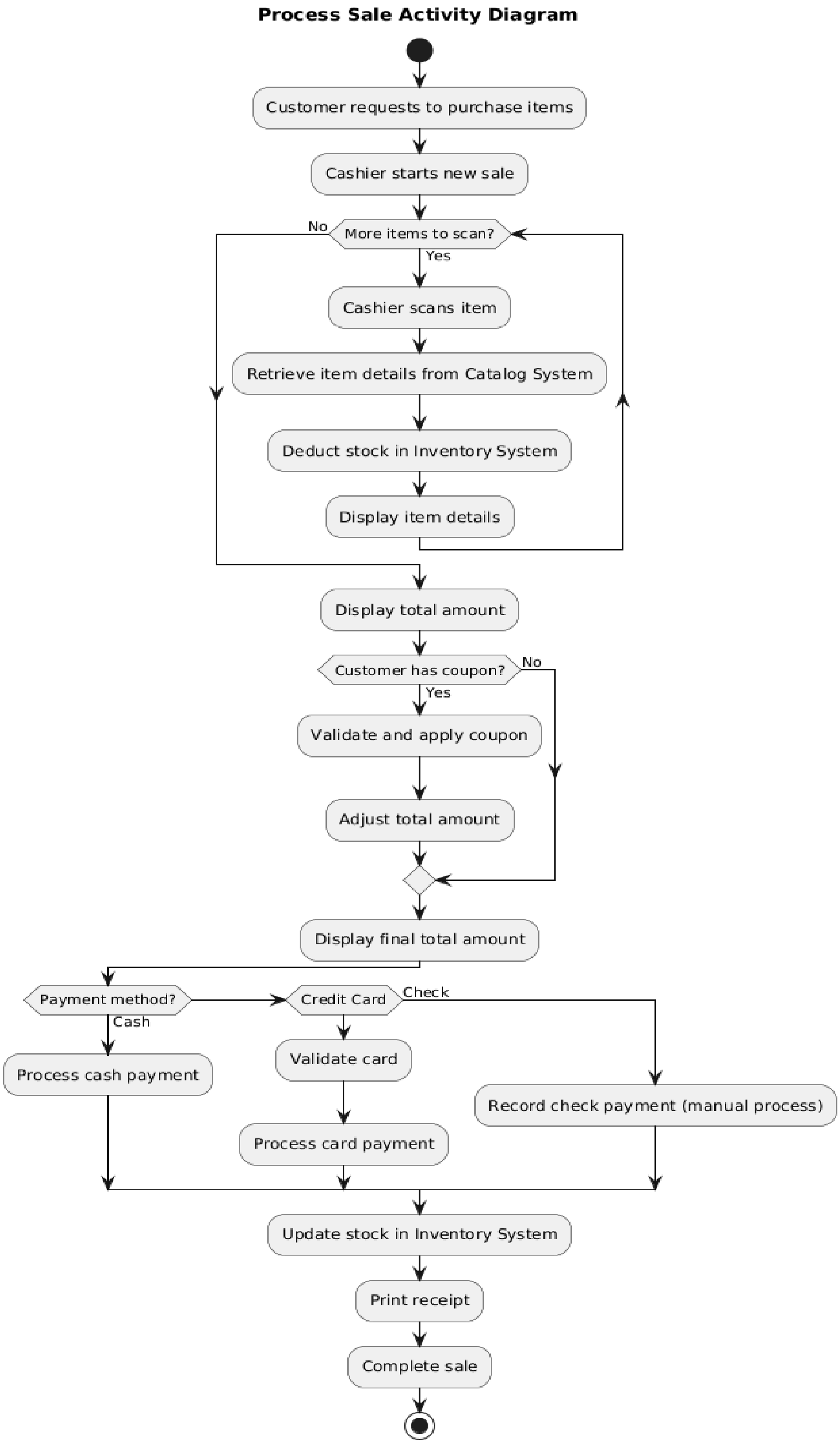
Task 3:

Develop Analysis Domain Models



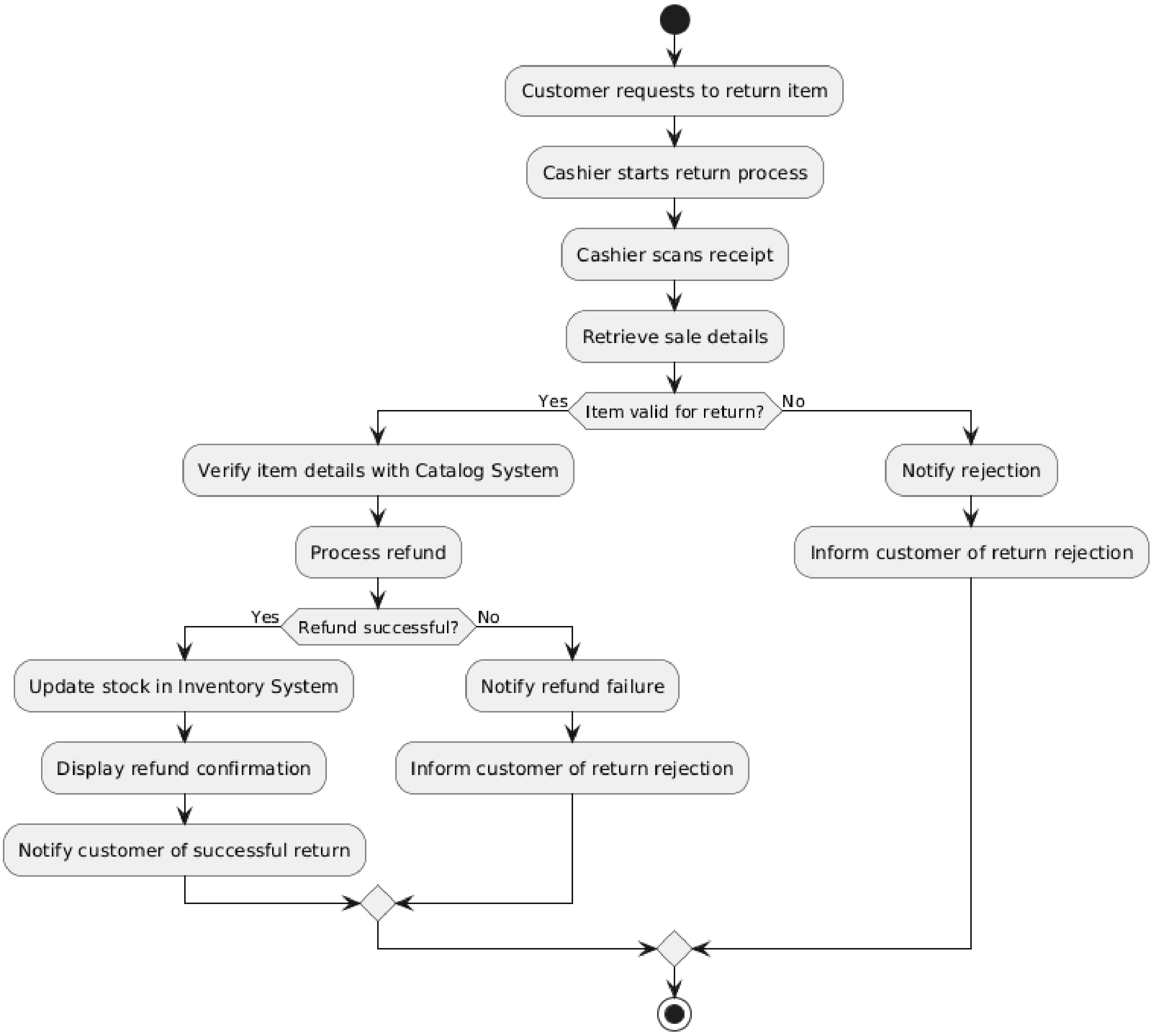
Task 4:

Develop activity diagram for "Process Sale".



Task 4:

Develop activity diagram for "Handle Return".



Customer requests to return item
Cashier starts return process
Cashier scans receipt
Retrieve sale details
Item valid for return?
Yes
No
Verify item details with Catalog System
Process refund
Refund successful?
Yes
No
Update stock in Inventory System
Display refund confirmation
Notify customer of successful return
Notify refund failure
Inform customer of return rejection
Notify rejection
Inform customer of return rejection