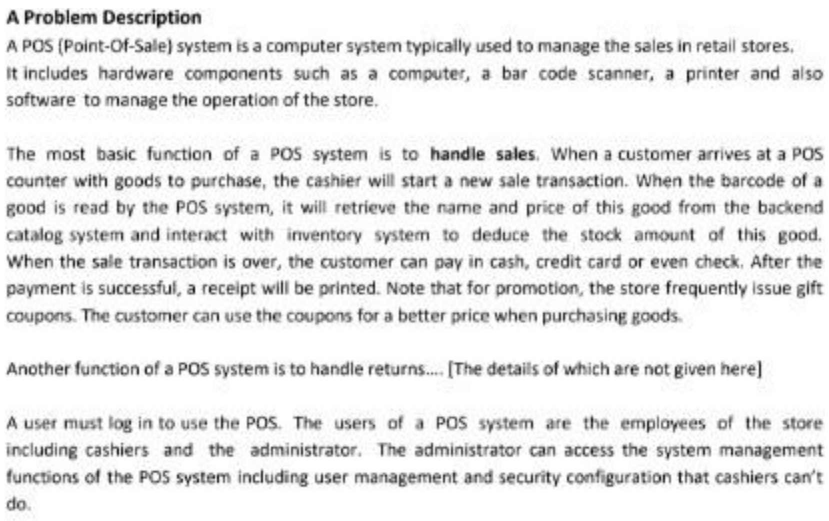
Lab-6

Meet Mahaliya

Roll No:202201204



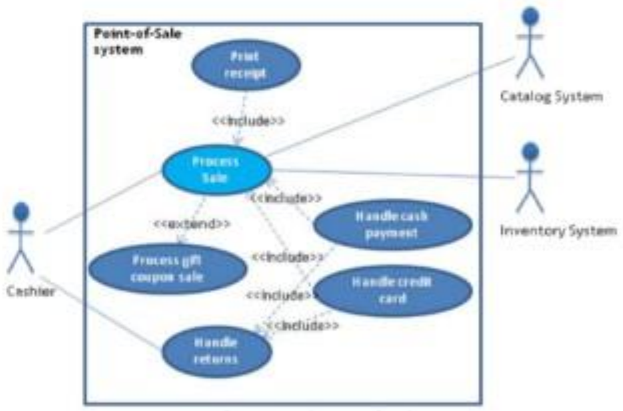
A Problem Description

A POS (Point-Of-Sale) system is a computer system typically used to manage the sales in retail stores. It includes hardware components such as a computer, a bar code scanner, a printer and also software to manage the operation of the store.

The most basic function of a POS system is to **handle sales**. When a customer arrives at a POS counter with goods to purchase, the cashier will start a new sale transaction. When the barcode of a good is read by the POS system, it will retrieve the name and price of this good from the backend catalog system and interact with inventory system to deduce the stock amount of this good. When the sale transaction is over, the customer can pay in cash, credit card or even check. After the payment is successful, a receipt will be printed. Note that for promotion, the store frequently issue gift coupons. The customer can use the coupons for a better price when purchasing goods.

Another function of a POS system is to handle returns.... [The details of which are not given here]

A user must log in to use the POS. The users of a POS system are the employees of the store including cashiers and the administrator. The administrator can access the system management functions of the POS system including user management and security configuration that cashiers can't do.



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

Use Case: Process Sale

Use Case Name: Process Sale
Primary Actor: Cashier
Stakeholders and Interests:

- **Customer:** Wants to purchase items and receive a receipt.
- **Cashier:** Wants to efficiently process the sale, ensure correct pricing, and update inventory.
- **Store Manager:** Wants accurate sales and inventory records.
- **Inventory System:** Needs to update stock levels.
- **Payment Processor:** Needs to process payments securely.

Preconditions:

1. The cashier must be logged into the POS system.
2. The POS system is connected to the backend catalog and inventory systems.
3. The POS system is connected to the payment processor.

Postconditions:

1. The sale is recorded in the system.
2. The inventory is updated to reflect the items sold.
3. A receipt is printed for the customer.
4. The customer's payment is processed successfully.

Main Success Scenario:

1. The cashier initiates a new sale transaction in the POS system.
2. The cashier scans the barcode of each item being purchased.
3. The POS system retrieves the name, price, and availability of each item from the backend catalog system.
4. The POS system displays the item details and the total amount on the screen.
5. The customer provides any gift coupons to the cashier.
6. The cashier applies the gift coupons, and the POS system recalculates the total amount.
7. The customer selects a payment method (cash, credit card, or check).
8. The cashier processes the payment using the selected method.
9. The payment is approved by the payment processor.
10. The POS system deducts the purchased quantities from the inventory.
11. The POS system prints a receipt for the customer.
12. The cashier hands the receipt to the customer, completing the sale.

Extensions (Alternate Scenarios): 1a. **Item Not Found:**

- If the scanned barcode is not found in the catalog, the POS system displays an error message.

- The cashier can manually enter the item information or ask the customer if they want to proceed without the item.

7a. **Invalid Coupon:**

- If the coupon is invalid, the POS system displays a message, and the cashier informs the customer.

8a. **Payment Fails:**

- If the payment fails, the POS system displays an error message.
- The customer selects an alternative payment method, and the cashier retries the payment process.

Use Case: Handle Return

Use Case Name: Handle Return
Primary Actor: Cashier
Stakeholders and Interests:

- **Customer:** Wants to return items and receive a refund.
- **Cashier:** Wants to process the return efficiently and update inventory.
- **Store Manager:** Wants accurate records of returns and inventory.
- **Inventory System:** Needs to update stock levels based on returns.
- **Payment Processor:** Needs to reverse payments securely.

Preconditions:

1. The cashier must be logged into the POS system.
2. The item being returned must have been previously purchased from the store.
3. The return is within the allowed return period as per store policy.

Postconditions:

1. The return is recorded in the system.
2. The inventory is updated to reflect the returned items.
3. A receipt or proof of return is printed for the customer.
4. The customer's refund is processed successfully.

Main Success Scenario:

1. The cashier initiates a new return transaction in the POS system.
2. The customer provides the receipt or proof of purchase for the item(s) being returned.
3. The cashier scans the barcode of the item(s) being returned.
4. The POS system verifies the purchase details, including date and price.
5. The cashier confirms the return and selects the refund method (cash, credit card, store credit).
6. The POS system updates the inventory to reflect the returned items.
7. The POS system processes the refund using the selected method.
8. The POS system prints a return receipt for the customer.
9. The cashier hands the return receipt to the customer, completing the return.

Extensions (Alternate Scenarios): 2a. **No Receipt:**

- If the customer does not have a receipt, the cashier can search for the purchase in the system using the customer's details (if available) or refuse the return as per store policy.

4a. **Return Outside Allowed Period:**

- If the return is outside the allowed period, the POS system displays an error message.
- The cashier informs the customer that the return cannot be processed.

5a. **Partial Return:**

- If the customer wants to return only some of the purchased items, the cashier selects the items to be returned, and the POS system calculates the appropriate refund amount.

These use cases provide a comprehensive view of the "Process Sale" and "Handle Return" functionalities, covering all primary and alternate scenarios.

2) Identify Entity/Boundary Control Objects

For the given POS system use cases ("Process Sale" and "Handle Return"), we can identify various **Entity**, **Boundary**, and **Control** objects as follows:

1. Entity Objects

Entity objects represent the core business data and are typically persistent.

- **Product:** Represents each item available for sale, including attributes like `barcode`, `name`, `price`, and `quantityInStock`.
- **SaleTransaction:** Represents a sale transaction, with attributes like `transactionID`, `date`, `itemsSold`, `totalAmount`, and `paymentMethod`.
- **SaleItem:** Represents an individual item within a sale transaction, including `product`, `quantity`, and `subtotal`.
- **ReturnTransaction:** Represents a return transaction, including `transactionID`, `date`, `itemsReturned`, and `refundAmount`.
- **User:** Represents a system user, such as a cashier or administrator, including `userID`, `username`, `password`, and `role`.
- **Customer:** Represents the customer involved in the sale or return, if customer information is recorded.
- **GiftCoupon:** Represents a discount coupon that can be applied to a sale, including `couponCode`, `discountAmount`, and `expiryDate`.
- **Payment:** Represents a payment made for a sale, including attributes like `paymentID`, `amount`, `paymentType`, and `status`.
- **Inventory:** Represents the inventory status for each product, including `productID` and `quantityInStock`.

2. Boundary Objects

Boundary objects represent the interface between the system and the external world (e.g., user interfaces, external systems).

- **POSScreen:** Represents the interface for the cashier to interact with the POS system, including sale processing, applying coupons, and managing returns.
- **BarcodeScanner:** Represents the hardware component used to scan product barcodes.
- **ReceiptPrinter:** Represents the hardware component used to print sale and return receipts.
- **PaymentTerminal:** Represents the interface for processing credit card payments and other electronic transactions.
- **CatalogSystem Interface:** Represents the interface to the backend catalog system for retrieving product information.
- **InventorySystem Interface:** Represents the interface to the backend inventory system for updating stock levels.
- **PaymentProcessor Interface:** Represents the interface to the external payment gateway for processing payments.

- **LoginScreen:** Represents the user interface for employees to log into the system.

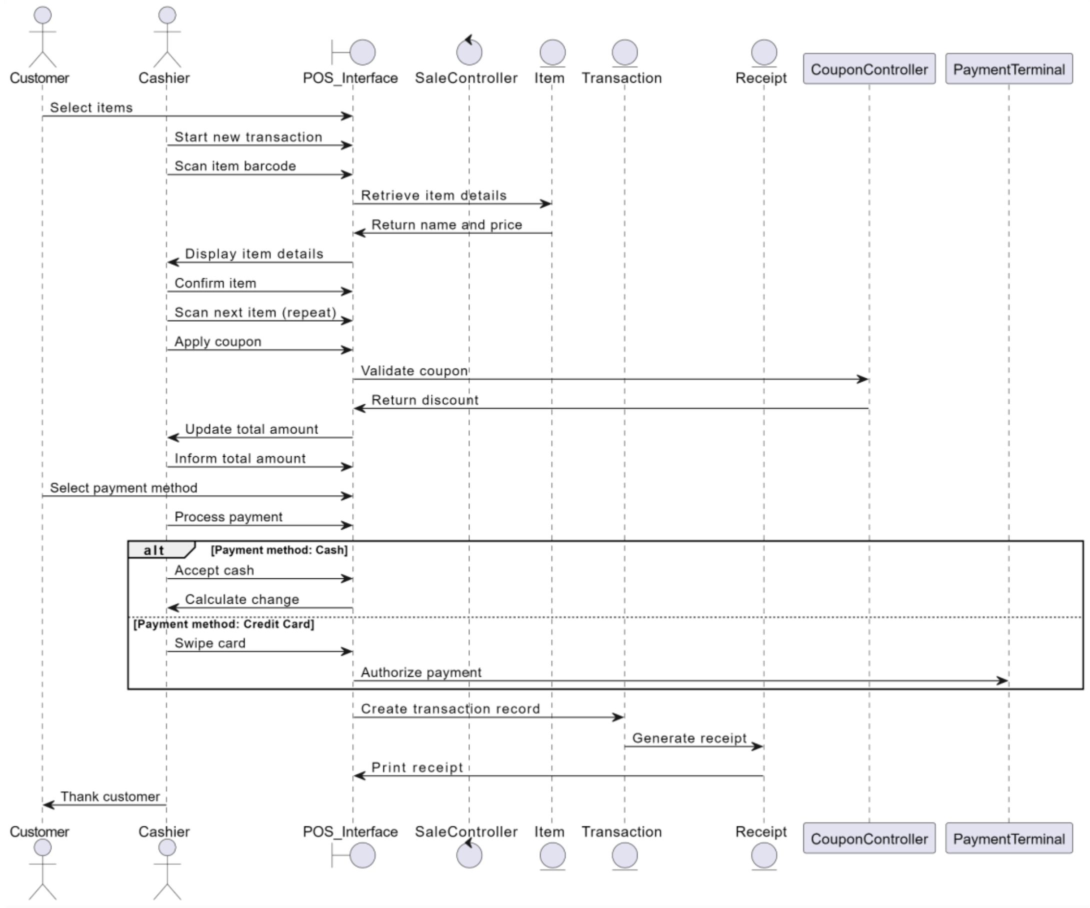
3. Control Objects

Control objects encapsulate the logic for coordinating the actions between entity and boundary objects, managing the workflow of the use case.

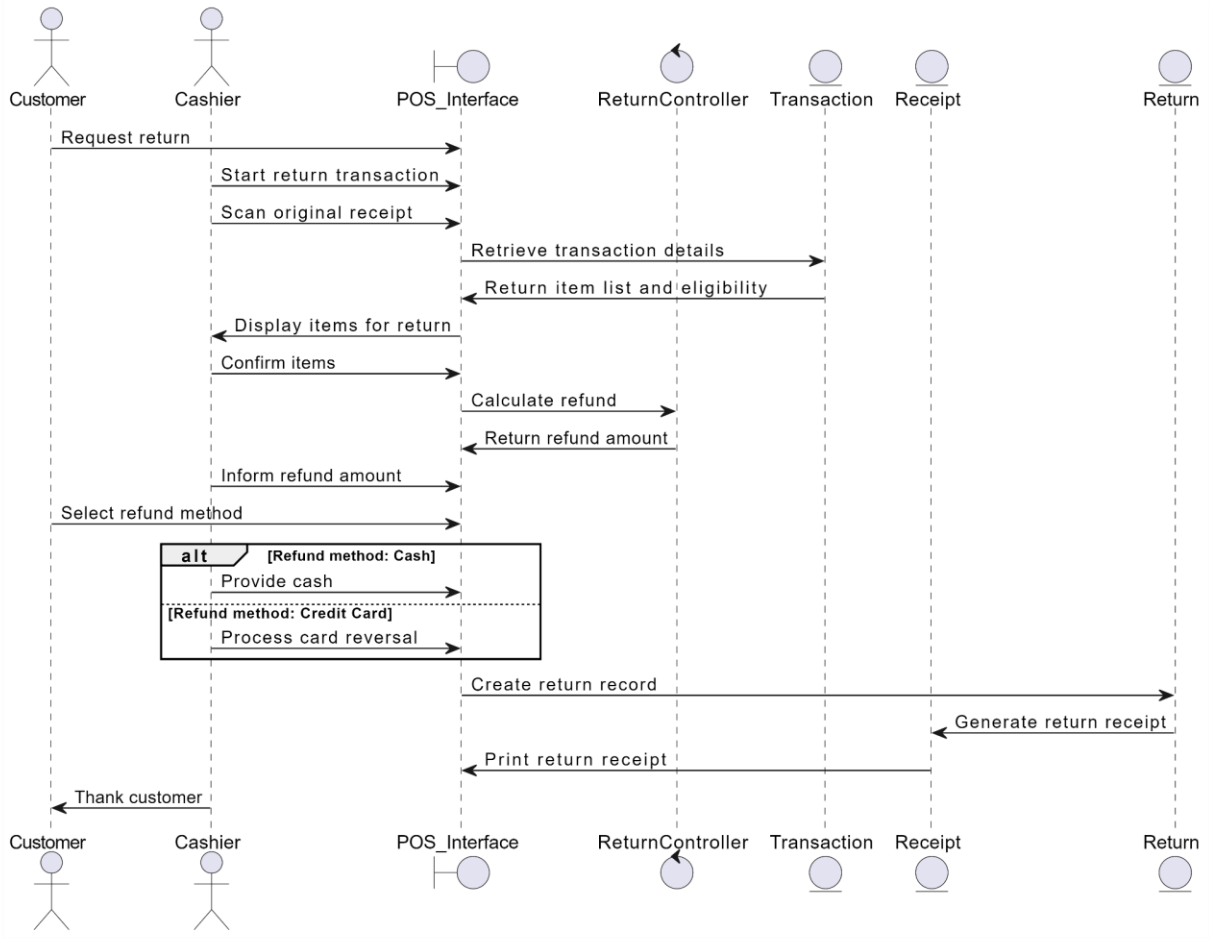
- **SaleController:** Handles the entire process of a sale transaction, including scanning items, applying coupons, calculating totals, and managing payments.
- **ReturnController:** Manages the workflow of a return transaction, including validating return items, processing refunds, and updating inventory.
- **LoginController:** Manages the user login process, verifying credentials, and determining user roles.
- **PaymentController:** Manages the payment process, including interaction with the payment terminal and payment processor, handling payment failures, and processing refunds.
- **InventoryController:** Coordinates the interaction between the sale/return process and the inventory system, ensuring inventory updates are accurately reflected.
- **CouponController:** Manages the application of gift coupons to a sale, validating coupons, and calculating discounts.
- **ReceiptController:** Manages the printing of receipts for sales and returns, formatting the receipt data for the printer.

3) Develop Sequence Diagrams

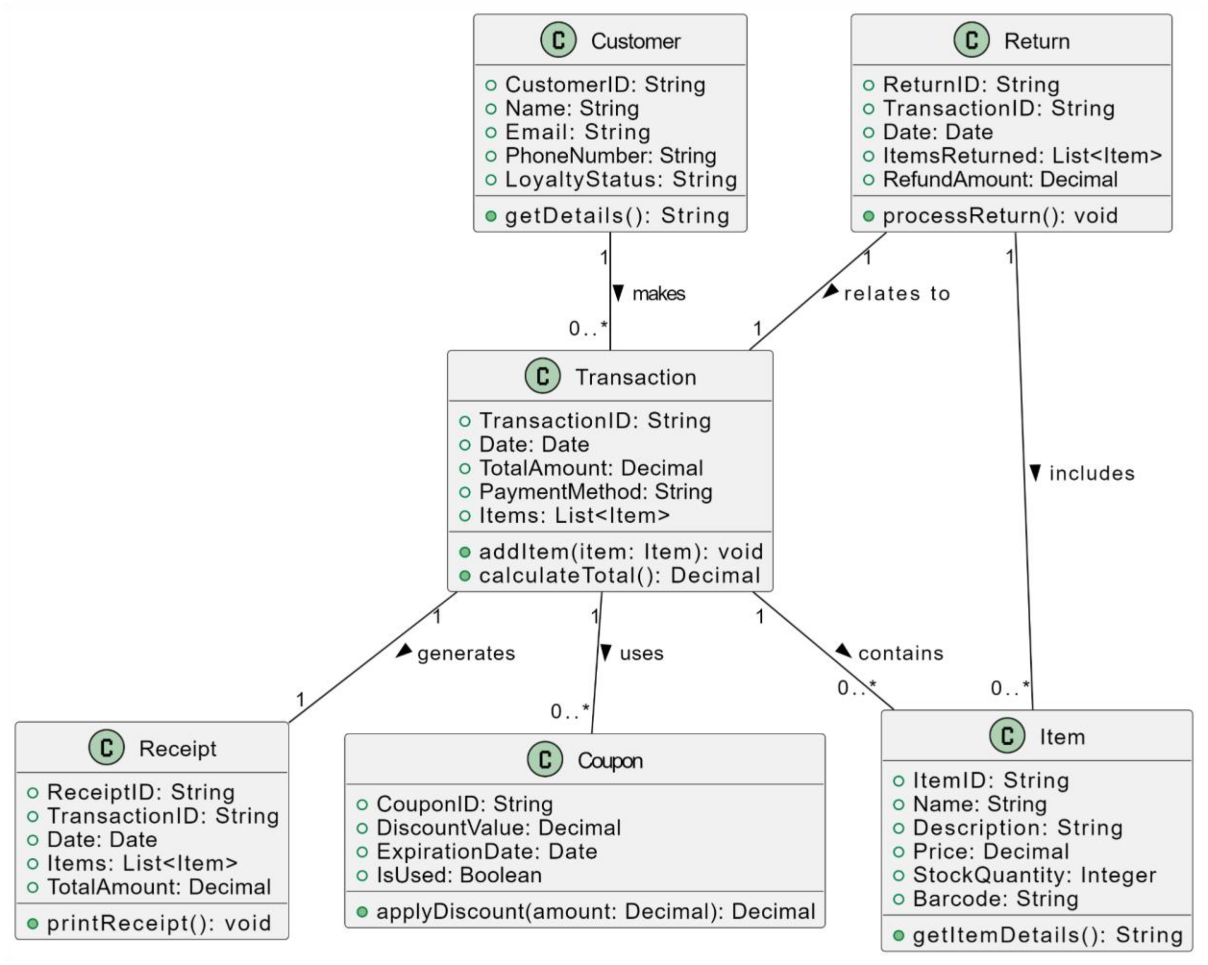
Process Sale-


Customer
Cashier
POS_Interface
SaleController
Item
Transaction
Receipt
CouponController
PaymentTerminal
Select items
Start new transaction
Scan item barcode
Retrieve item details
Return name and price
Display item details
Confirm item
Scan next item (repeat)
Apply coupon
Validate coupon
Return discount
Update total amount
Inform total amount
Select payment method
Process payment
alt
[Payment method: Cash]
Accept cash
Calculate change
[Payment method: Credit Card]
Swipe card
Authorize payment
Create transaction record
Generate receipt
Print receipt
Thank customer

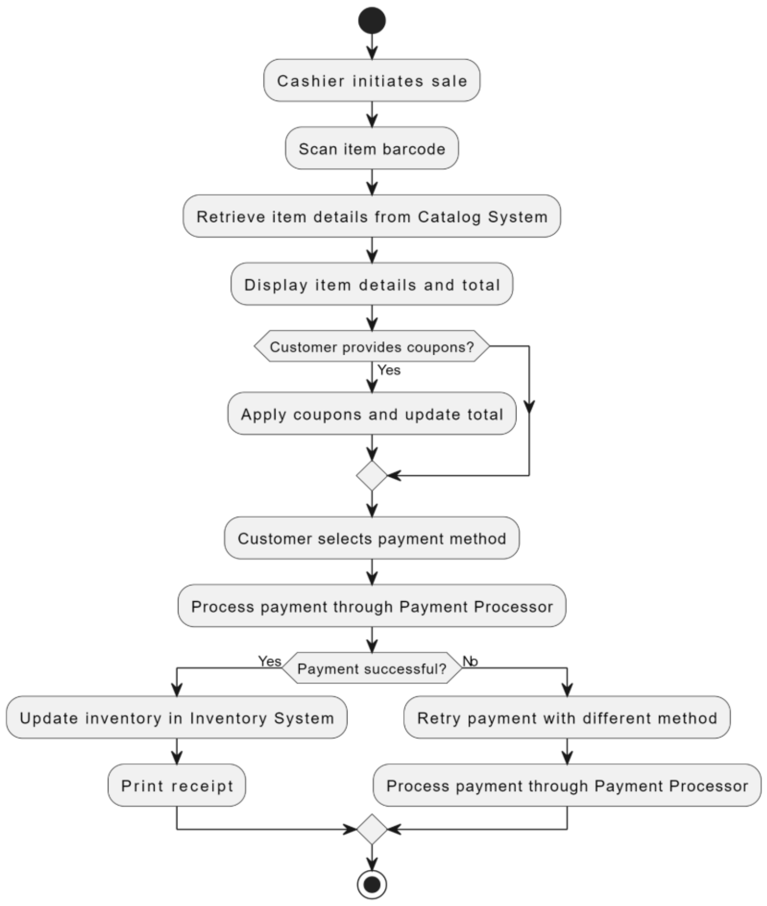
Handle Return

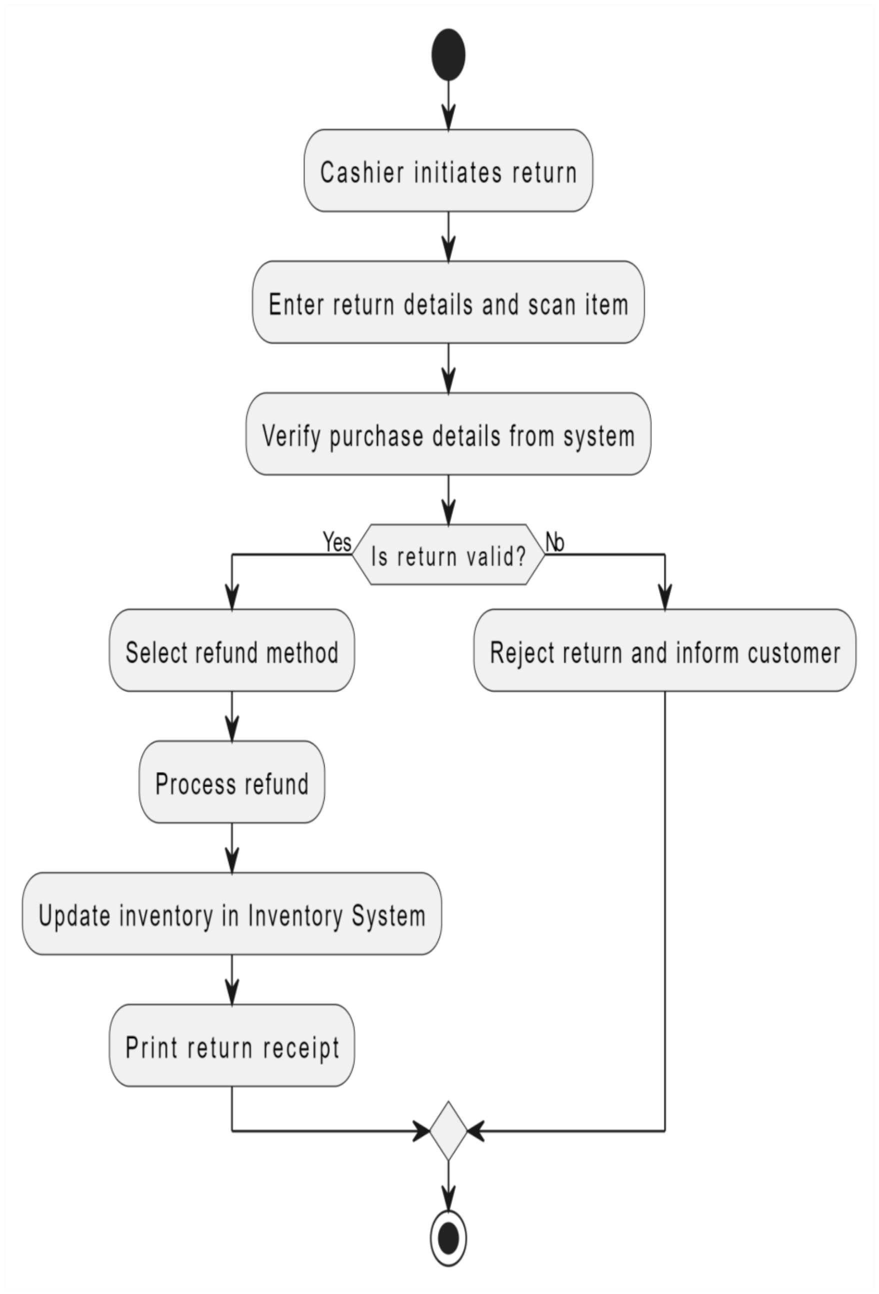


4) Develop Analysis Domain Models



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


Cashier initiates sale
Scan item barcode
Retrieve item details from Catalog System
Display item details and total
Customer provides coupons?
Yes
Apply coupons and update total
Customer selects payment method
Process payment through Payment Processor
Payment successful?
Yes
No
Update inventory in Inventory System
Print receipt
Retry payment with different method
Process payment through Payment Processor


Cashier initiates return
Enter return details and scan item
Verify purchase details from system
Is return valid?
Yes
No
Select refund method
Process refund
Update inventory in Inventory System
Print return receipt
Reject return and inform customer