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

Name: Praneel Vania
ID: 202201131

Task:

- **Develop Use Case Textual Description for "Process Sale" and "Handle Return" use cases.**
- **Identify Entity/Boundary Control Objects**
- **Develop Sequence Diagrams**
- **Develop Analysis Domain Models**
- **Develop activity diagram for "Process Sale" and "Handle Return" use cases.**

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 also receive a refund for that item.
- **Cashier:** Wants to process the return properly and update inventory respectively.
- **Store Manager:** Wants accurate records of returns and inventory.
- **Inventory System:** Needs to update stock levels based on returns.
- **Payment Processor:** Needs to reverse the payments securely.

Preconditions:

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

Postconditions:

1. The return should be recorded in the system.
2. The inventory has to be updated such that it reflects 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:

The cashier initiates a new return transaction in the POS system.

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

Extensions (Alternate Scenarios):

2a. Absence of Receipt:

When a customer lacks a receipt, the cashier can attempt to locate the transaction in the system using the customer's information (if available) or may need to decline the return based on store policy.

4a. Return Beyond Allowed Timeframe:

If a return request is made after the permitted timeframe, the POS system will show an error notification. The cashier will then inform the customer that the return cannot be completed.

5a. Partial Returns:

Should the customer wish to return only specific items from their purchase, the cashier

will select those items, allowing the POS system to calculate the corresponding refund amount.

These use cases offer a thorough understanding of the "Process Sale" and "Handle Return" functionalities, addressing all main and alternative scenarios.

2. Identify Entity/Boundary control objects

Identification of Objects for the POS System Use Cases

For the POS system scenarios involving "Process Sale" and "Handle Return," we can categorize various Entity, Boundary, and Control objects as follows:

Entity Objects

Entity objects encapsulate the essential business data and are generally persistent.

- Product: Represents items available for sale, including attributes such as barcode, name, price, and quantity in stock.
- SaleTransaction: Represents a sale event, with attributes like transaction ID, date, items sold, total amount, and payment method.
- SaleItem: Represents a specific item within a sale, detailing the product, quantity, and subtotal.
- ReturnTransaction: Represents a return event, including transaction ID, date, items returned, and refund amount.
- User: Represents system users, like cashiers or administrators, including user ID, username, password, and role.
- Customer: Represents the customer involved in a sale or return, when their information is recorded.
- GiftCoupon: Represents a discount coupon applicable to a sale, including coupon code, discount amount, and expiry date.
- Payment: Represents a payment made for a sale, with attributes such as payment ID, amount, payment type, and status.
- Inventory: Represents the stock status for each product, including product ID and quantity in stock.

Boundary Objects

Boundary objects act as the interface between the system and external entities, such as user interfaces and external systems.

- POSScreen: The interface for cashiers to interact with the POS system, facilitating sale processing, coupon application, and return management.
- BarcodeScanner: The hardware component utilized to scan product barcodes.
- ReceiptPrinter: The hardware used to print receipts for sales and returns.
- PaymentTerminal: The interface for handling credit card payments and other electronic transactions.
- CatalogSystem Interface: The interface connecting to the backend catalog system for retrieving product details.
- InventorySystem Interface: The interface for updating stock levels in the backend inventory system.
- PaymentProcessor Interface: The interface to the external payment gateway for processing payments.
- LoginScreen: The user interface for employee login into the system.

Control Objects

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

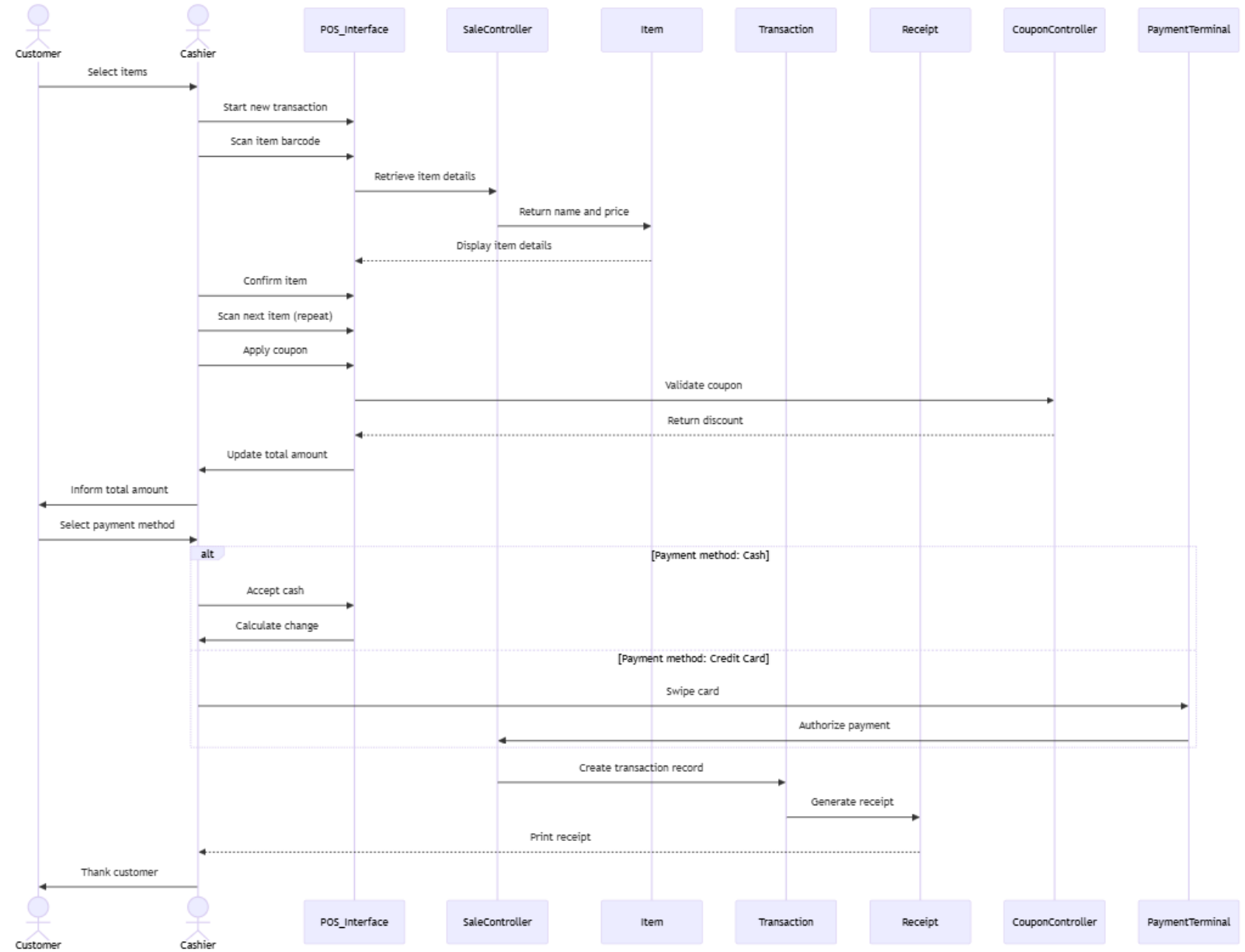
- SaleController: Oversees the complete process of a sale transaction, including item scanning, coupon application, total calculation, and payment management.
- ReturnController: Manages the workflow of a return transaction, including item validation, refund processing, and inventory updates.
- LoginController: Oversees the user login procedure, verifying credentials and determining user roles.
- PaymentController: Manages payment processing, including interactions with the payment terminal and processor, handling payment issues, and processing refunds.
- InventoryController: Coordinates the interaction between sales/returns and the inventory system, ensuring accurate stock updates.

- CouponController: Manages the application of gift coupons during a sale, validating coupons and calculating discounts.
- ReceiptController: Manages receipt printing for sales and returns, formatting the receipt data appropriately for the printer.

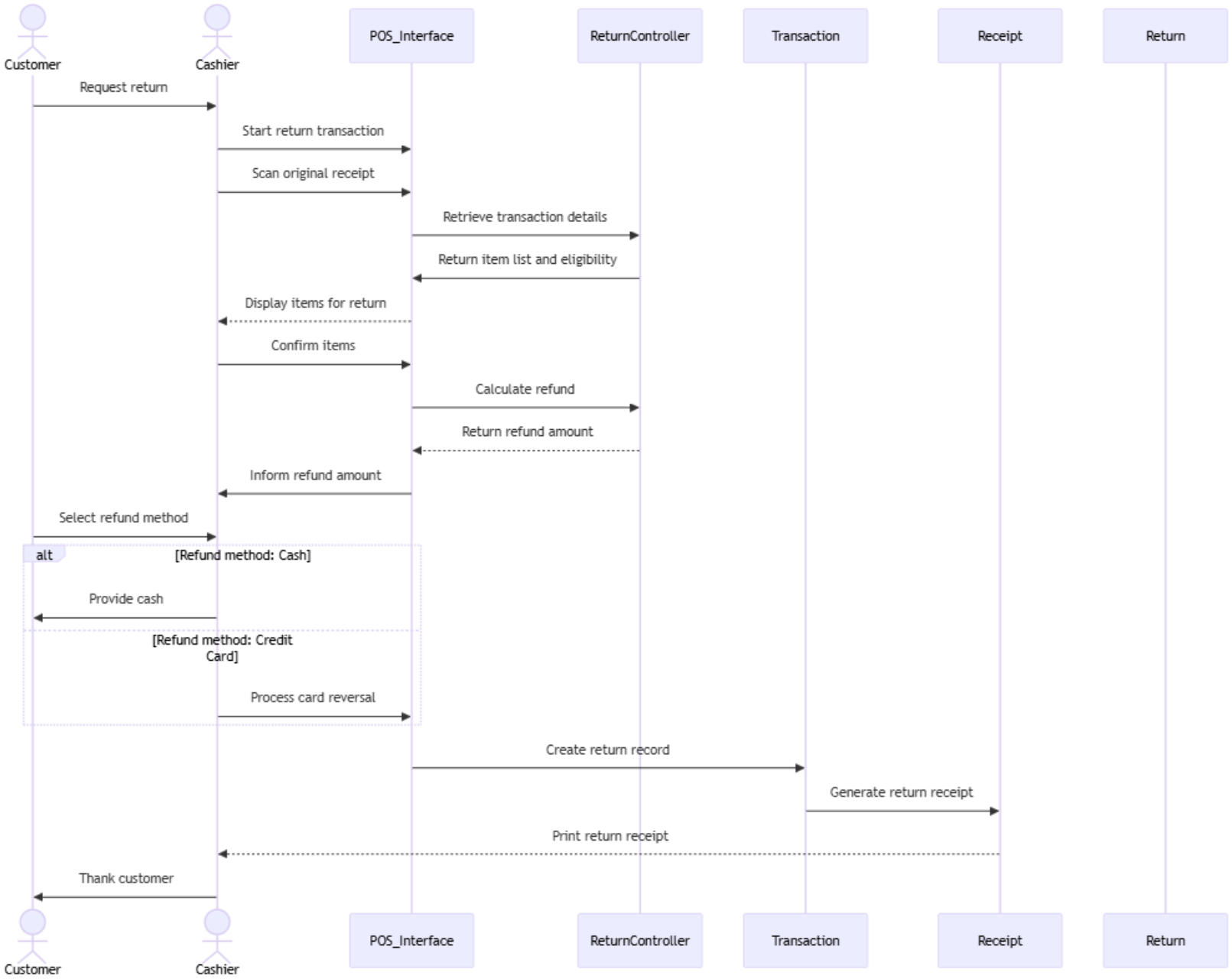
This categorization provides a structured view of the various components involved in the "Process Sale" and "Handle Return" functionalities of the POS system.

3. Develop Sequence Diagrams

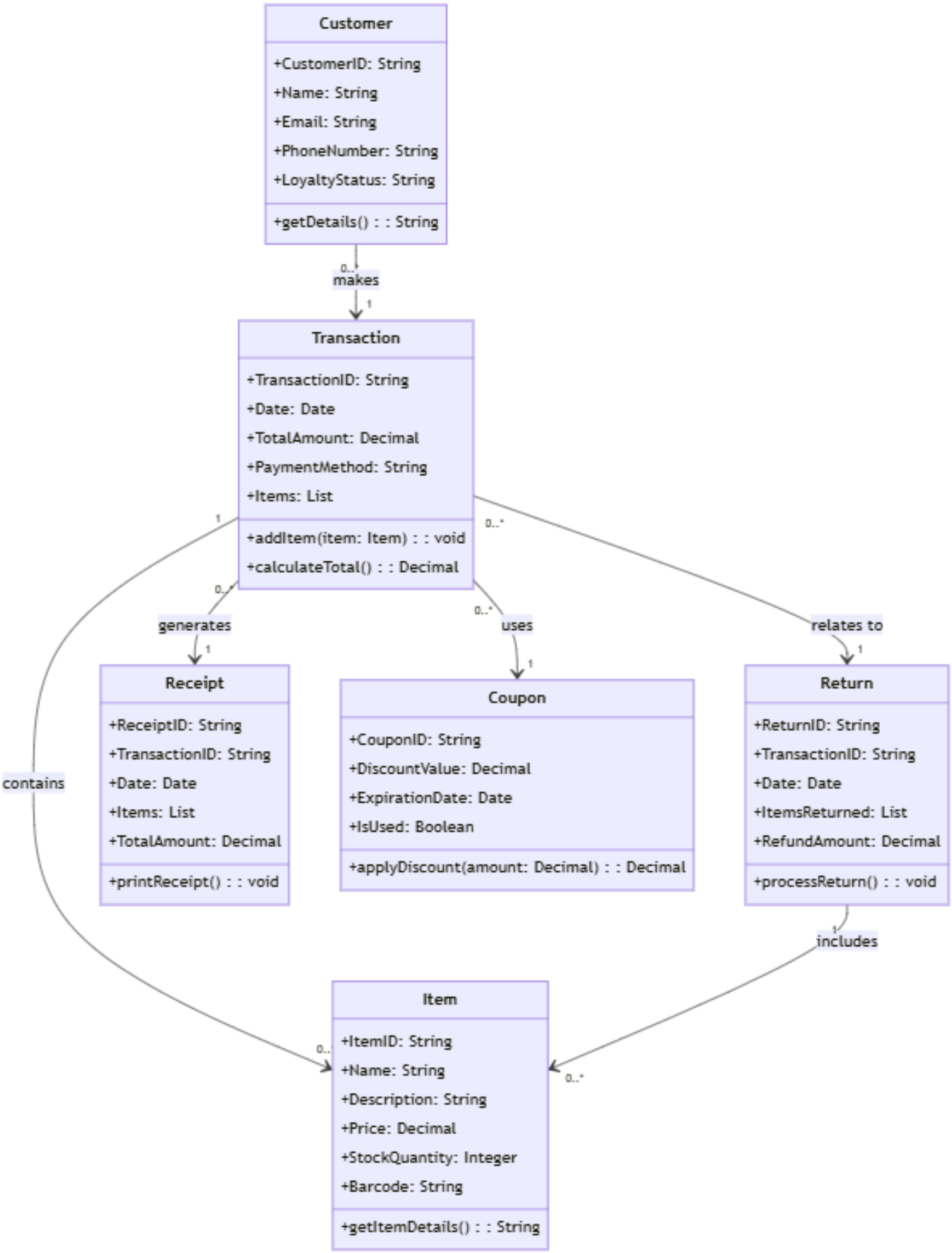
Process sales :



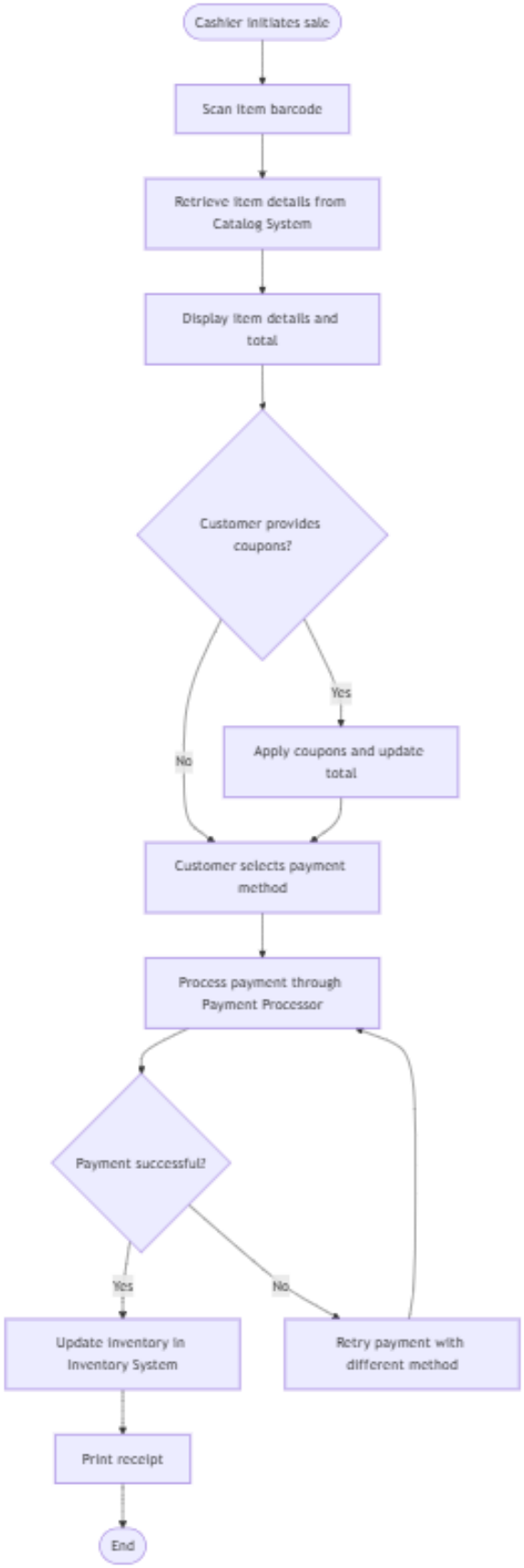
Handle return :

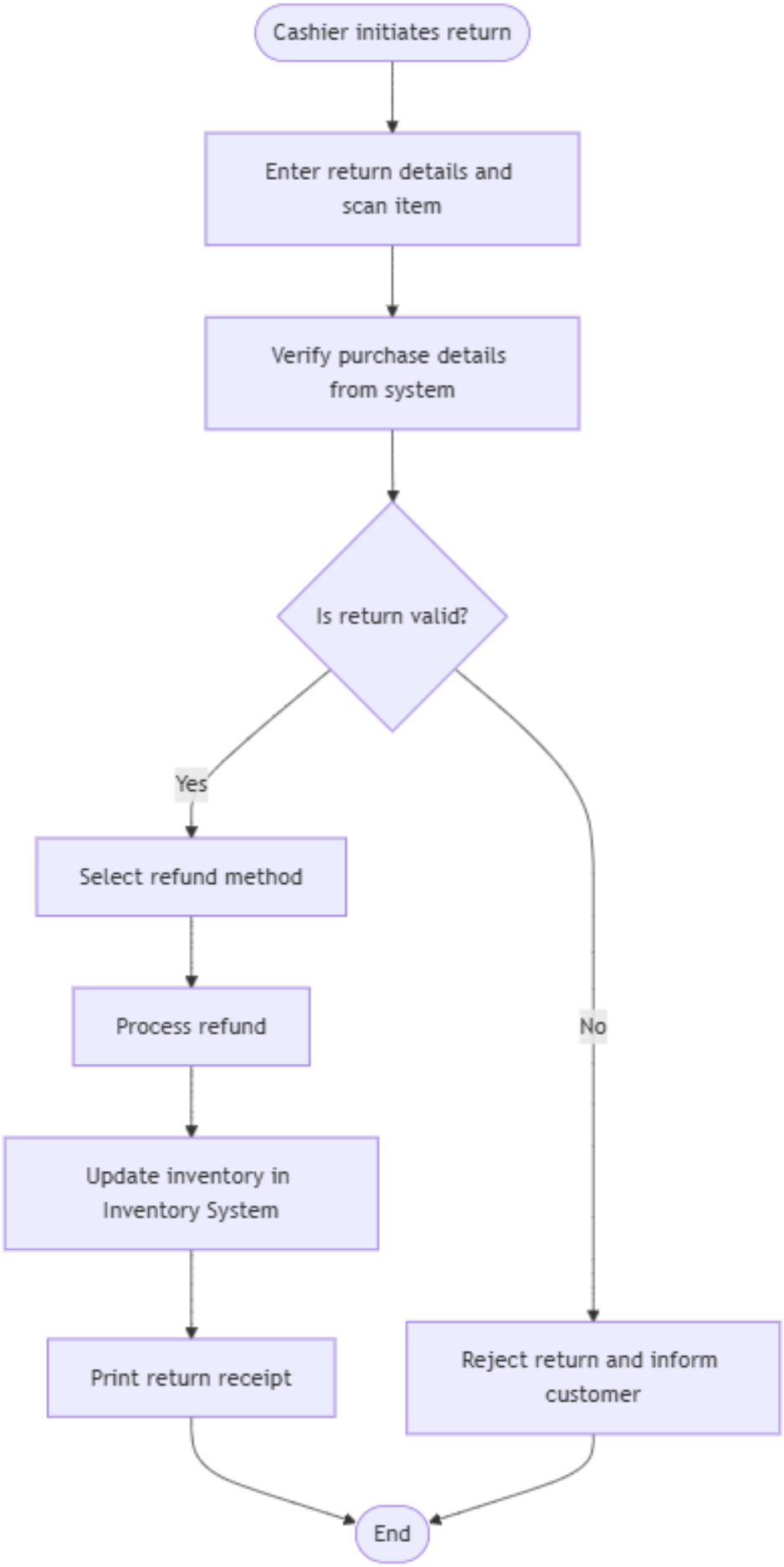


4. Develop Analysis Domain Models



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




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
End