IT314 Software Engineering

Lab-6: Modeling Class Diagram and Activity Diagram

Dev Davda

202201242

Task 1:

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

Process Sale:

Actors:

- Cashier
- System

Pre-conditions:

- The POS system is online and operational.
- The cashier is logged into the system.

Basic Flow:

1. The customer arrives at the checkout with their items.
2. The cashier scans each item.
3. The system retrieves item information (e.g., description, price) from the database and displays it on the POS terminal.
4. The system calculates and displays the running subtotal, taxes, and total amount due.
5. The customer chooses a payment method (e.g., cash, credit card).
6. The cashier enters payment information and processes the transaction.
7. The system verifies the payment.
8. If payment is approved:
    - The system generates a receipt.
    - The system updates the inventory.
9. The cashier gives the receipt and any change to the customer.

Alternative Flows:

- Item not found: If an item's barcode is not recognized, the cashier can manually enter the item code. If the item is still not found, the cashier can consult with a manager.
- Payment declined: If the payment is declined, the cashier informs the customer and asks for an alternative payment method.
- Void/Cancel transaction: The cashier can void or cancel the transaction at any point before finalizing the payment.
- Customer loyalty program: The customer may provide their loyalty card or phone number to earn points or redeem rewards.

Post-conditions:

- The sale is complete.
- The customer has received their purchased items and a receipt.
- The system has updated the inventory and sales records.

Handle Return:

Actors:

- Cashier
- System

Pre-conditions:

- The POS system is online and operational.
- The cashier is logged into the system.

Basic Flow:

1. The customer approaches the cashier with the item(s) they want to return or exchange.
2. The cashier scans the item(s) and retrieves the original transaction details.
3. The system displays the original transaction information (e.g., date, items, price).
4. The cashier verifies the return policy and ensures the return is eligible.
5. If the return is approved:
    - The cashier selects "Return" or "Exchange" on the POS terminal.
    - For returns, the system calculates the refund amount.
    - For exchanges, the customer selects a replacement item.
6. The cashier processes the return/exchange in the system.
7. The system updates the inventory and sales records.
8. The customer receives a refund (if applicable) and/or the exchanged item.

Alternative Flows:

- Return without receipt: The customer may not have a receipt, in which case the cashier may use alternative methods to verify the purchase (e.g., loyalty program history).
- Ineligible return: The item may not be eligible for return (e.g., final sale, past the return window). The cashier informs the customer of the store's policy.
- Damaged/Defective item: The returned item may be damaged or defective. The cashier follows store procedures for handling such returns (e.g., separate processing, manager approval).

Post-conditions:

- The return or exchange is complete.
- The customer has received a refund (if applicable) or an exchange.
- The system has updated the inventory and sales records.

Task 2:

Identify Entity/Boundary Control Objects

Entity Objects:

- Sale
- Item
- Payment
- Customer
- Inventory

Boundary Objects:

- POS Terminal
- Receipt Printer
- Barcode Scanner

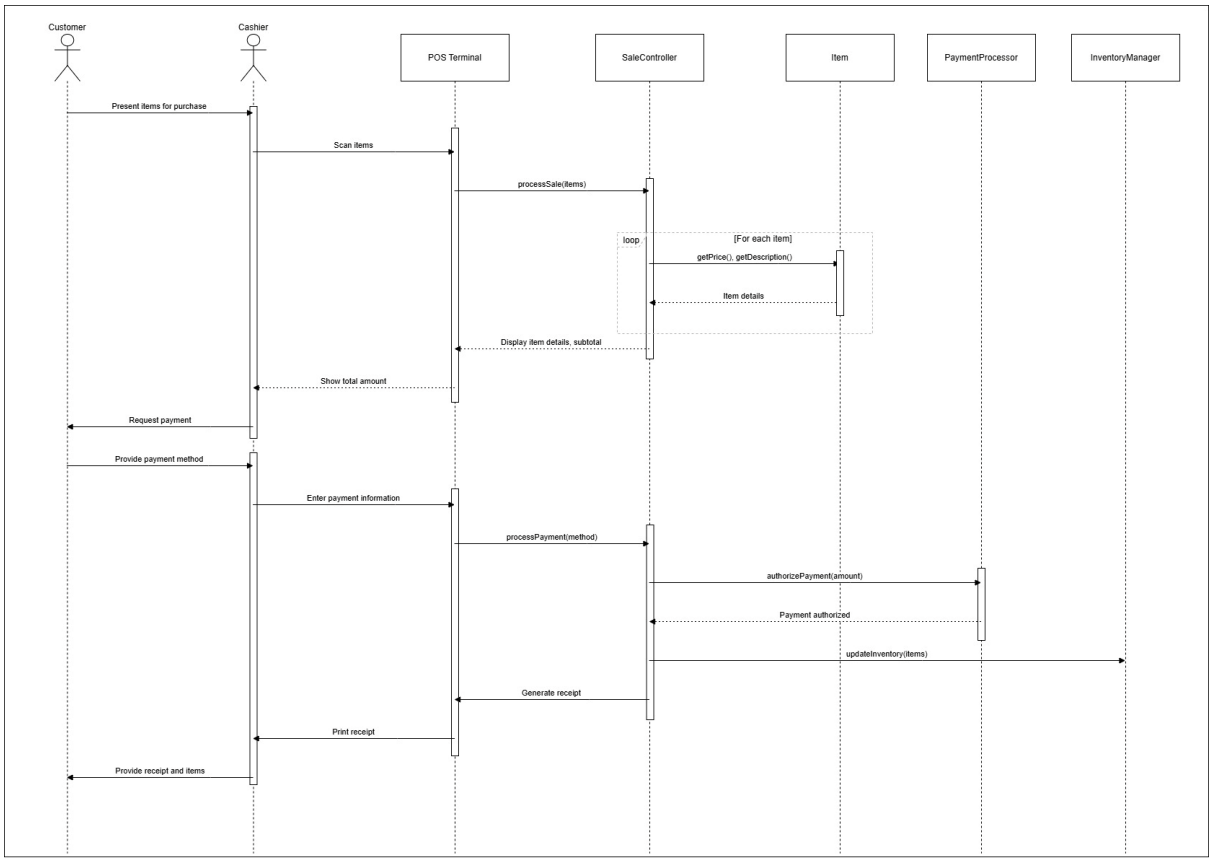
Control Objects:

- SaleController (manages the flow of the Process Sale use case)
- ReturnController (manages the flow of the Handle Return use case)
- PaymentProcessor (handles payment authorization and processing)
- InventoryManager (updates inventory levels)

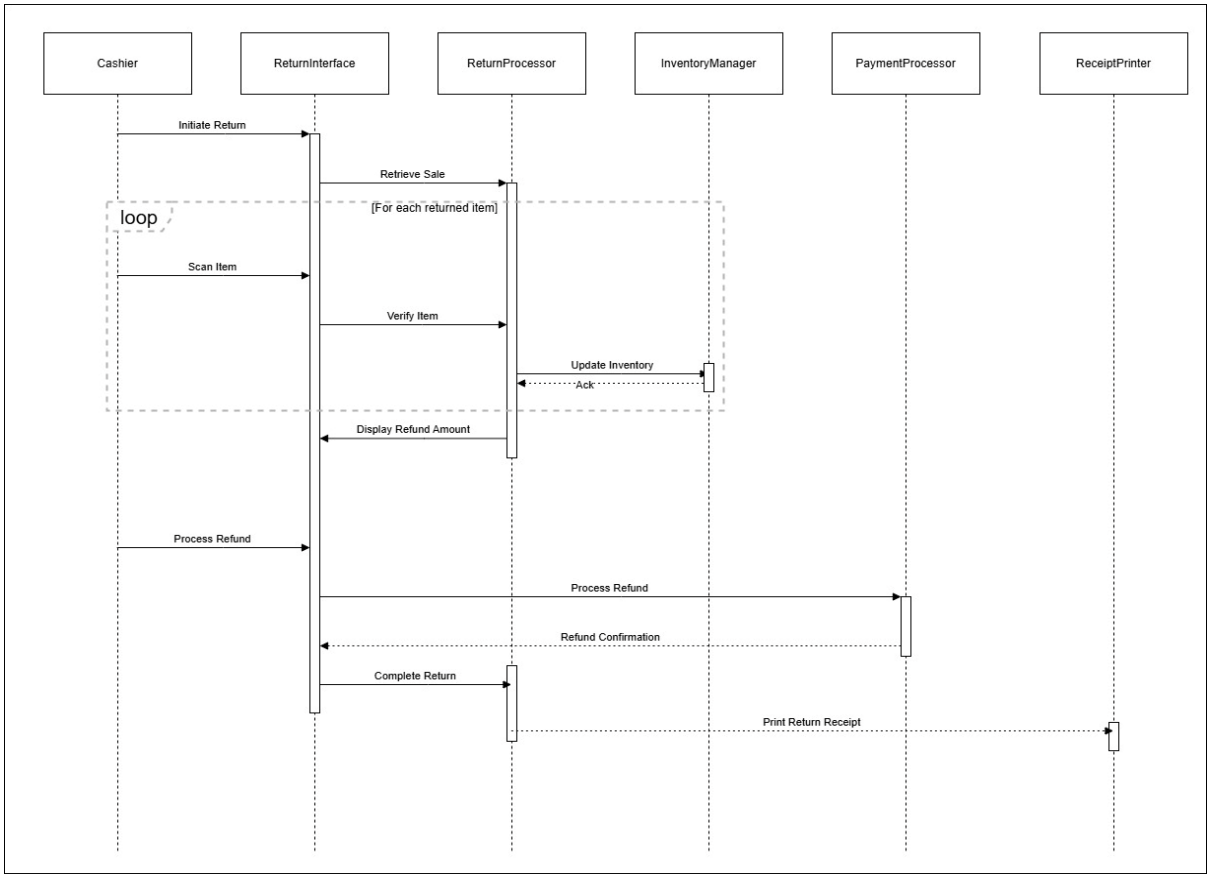
Task 3:

Develop Sequence Diagrams:

- **Process Sale**

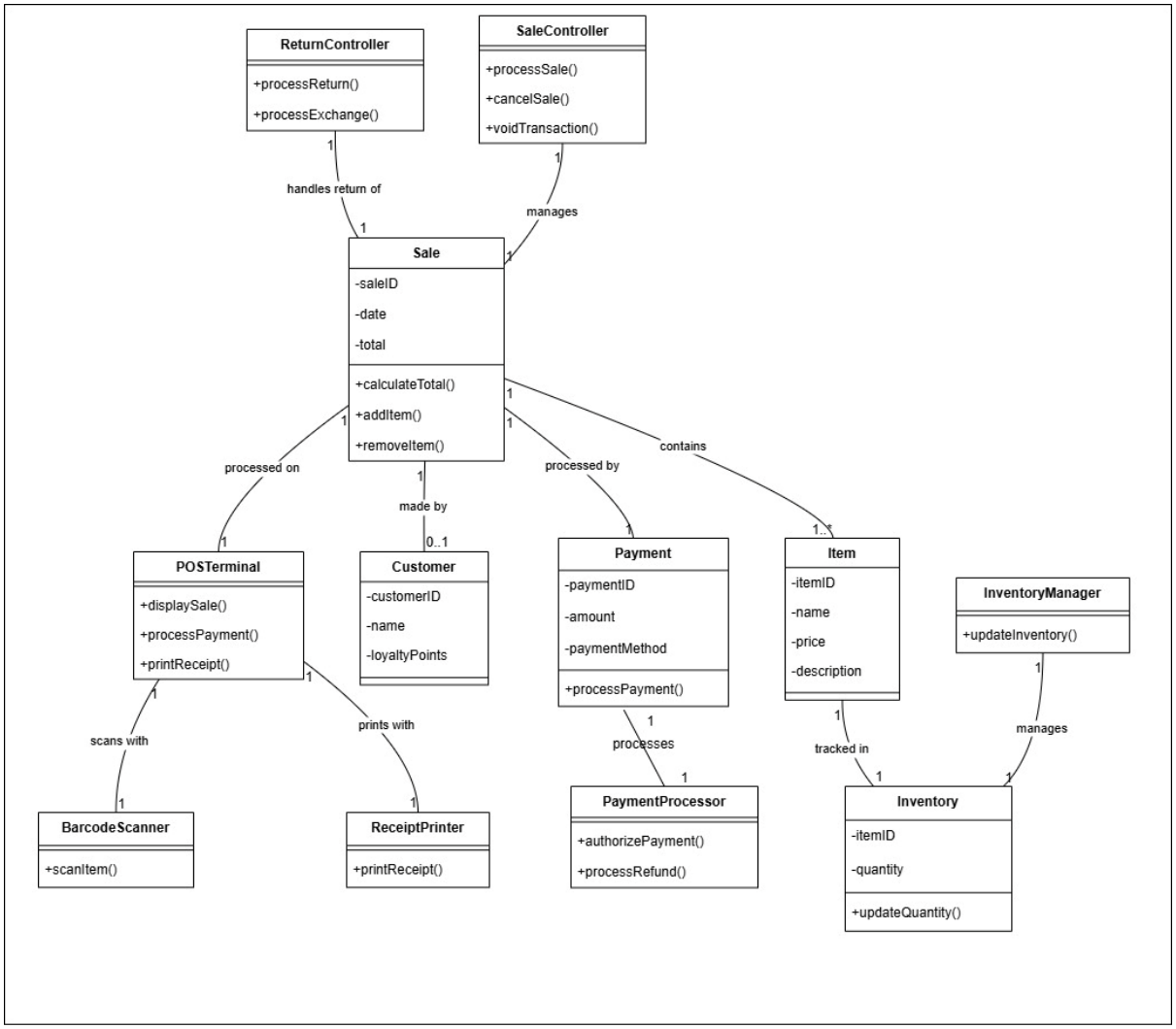


- **Handle Return**



Task 4:

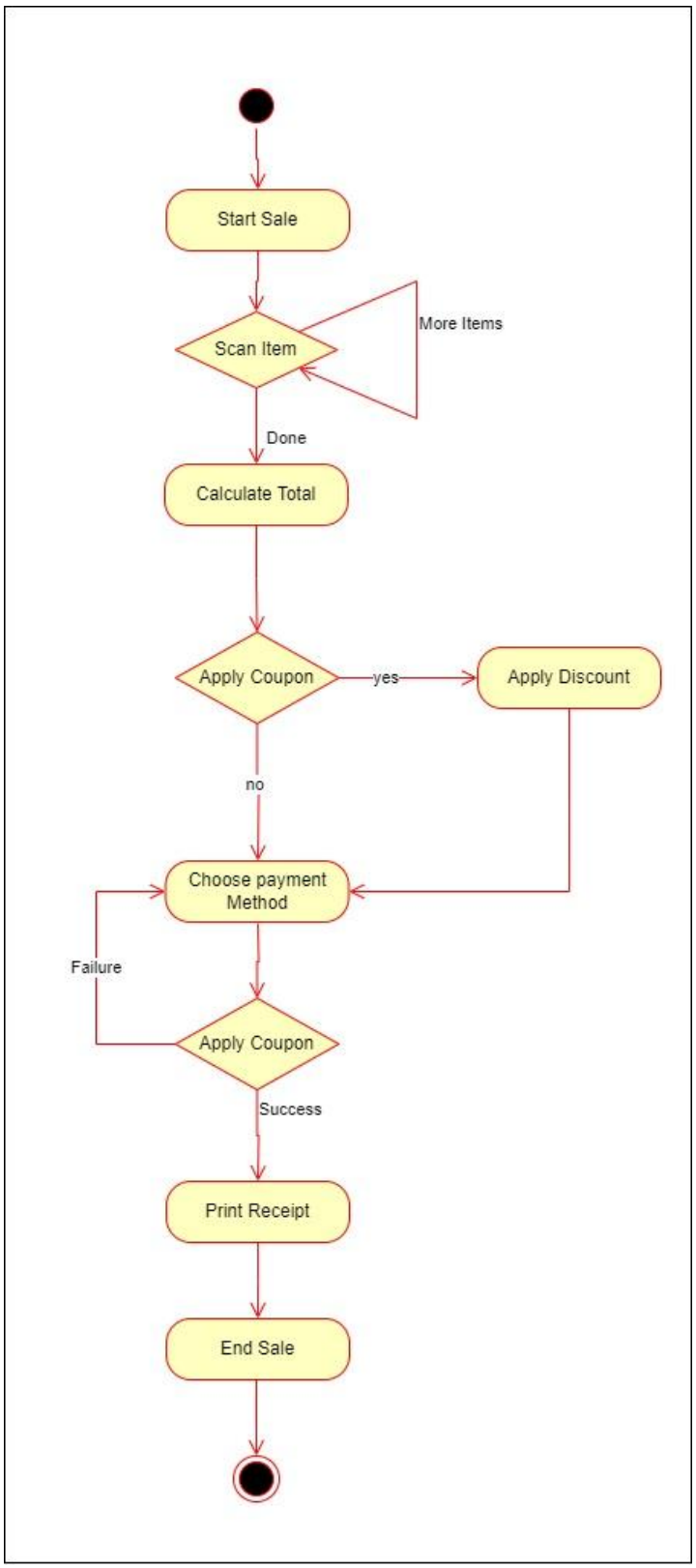
Develop Analysis Domain Models



Task 5:

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

- **Process Sale**



- **Handle Return:**