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

A.)

Use Case: Process Sale

Use Case ID: UC-01
Use Case Name: Process Sale
Actor: Cashier
Preconditions:

- The cashier is logged into the POS system and is ready with the new sale transaction.
- The items to be purchased are ready for scanning.

Postconditions:

- A sales transaction is completed, and a receipt is generated.
- Payment confirmation is sent both to the cashier and customer.
- Inventory levels are updated to reflect the sale.

Main Flow:

1. The cashier initiates a new sale transaction in the POS system.
2. The cashier scans the barcode of the first item.
3. The system retrieves the item name and price from the backend catalog.
4. The system displays the item details on the screen.
5. The cashier confirms the item and proceeds to the next item, repeating steps 2-4 for all items.
6. The system calculates the total amount for all items, including any applicable taxes and discounts.
7. The cashier informs the customer of the total amount due.
8. The customer selects a payment method (cash, credit card, or check).
    - If the customer has a gift coupon/voucher, the cashier applies it to the transaction.
9. The system processes the payment:
    - If payment is by credit card, the system connects to the payment gateway for authorization.

  - If cash, the cashier accepts the cash and gives change as necessary.

10. The system confirms the payment status.
11. The system prints a receipt for the customer, summarizing the transaction.
12. The sale transaction is recorded in the system, and inventory is updated accordingly.

Alternate Flow:

- **Invalid Payment**: If the payment fails, the system displays an error message, and the cashier can either retry the payment or ask for an alternative payment method.
- **Item Not Found**: If a scanned item is not found in the catalog, the system prompts the cashier to check the barcode or item availability.

Use Case: Handle Return

Use Case ID: UC-02
Use Case Name: Handle Return
Actor: Cashier
Preconditions:

- The cashier is logged into the POS system.
- The customer has items to return and provides a receipt.

Postconditions:

- The return transaction is completed, and inventory is updated.
- A return receipt is generated for the customer.

Main Flow:

1. The cashier initiates a return transaction in the POS system.
2. The cashier requests the customer to provide the original receipt.
3. The cashier scans the receipt barcode to retrieve transaction details.
4. The system displays the items eligible for return.
5. The cashier verifies the items with the customer.
6. The cashier scans each item being returned.
7. The system checks the eligibility of each item for return (e.g., within return period, condition).

8. The system calculates the total amount to be refunded.
9. The cashier informs the customer of the refund amount.
10. The customer selects a refund method (cash, credit to card, or store credit).
11. The system processes the refund:
    - If cash, the cashier provides the cash refund.
    - If credit to card, the system processes the refund through the payment gateway.
12. The system confirms the refund status.
13. The system prints a return receipt for the customer, summarizing the return transaction.
14. The return transaction is recorded in the system, and inventory is updated to reflect the returned items.

Alternate Flow:

- **Item Not Eligible for Return**: If an item is not eligible for return, the system displays a message, and the cashier explains to the customer the reason.
- **Receipt Not Found**: If the receipt cannot be scanned or is invalid, the system prompts the cashier to manually check the transaction details.

Q.)Identify Entity/Boundary Control Objects.

A.)

Entity Objects

These represent the core data and business logic of the system.

1. **Product**: Represents an item for sale, including attributes like name, price, SKU, and stock quantity.
2. **Customer**: Contains information about customers, including name, contact details, and purchase history.
3. **Transaction**: Represents a sale or return transaction, containing details like transaction ID, date, total amount, items purchased, and payment method.
4. **Receipt**: Represents the printed receipt for a transaction, summarizing the items purchased, total cost, and any discounts applied.
5. **Coupon**: Represents promotional gift coupons, including coupon code, discount value, and expiration date.

6. **Inventory**: Represents the current stock levels and related management functions.

Boundary Objects

These define the interaction/interface points between the system and the users.

1. **POS Interface**: The main user interface for cashiers, allowing them to initiate sales, process payments, and handle returns.
2. **Login Screen**: A boundary object for user authentication, enabling cashiers and administrators to log into the system.
3. **Receipt Printer**: A physical or virtual interface for printing transaction receipts.
4. **Payment Gateway Interface**: A boundary for processing electronic payments (credit/debit cards).
5. **Return Interface**: The section of the POS interface that allows cashiers to initiate and process return transactions.

Control Objects

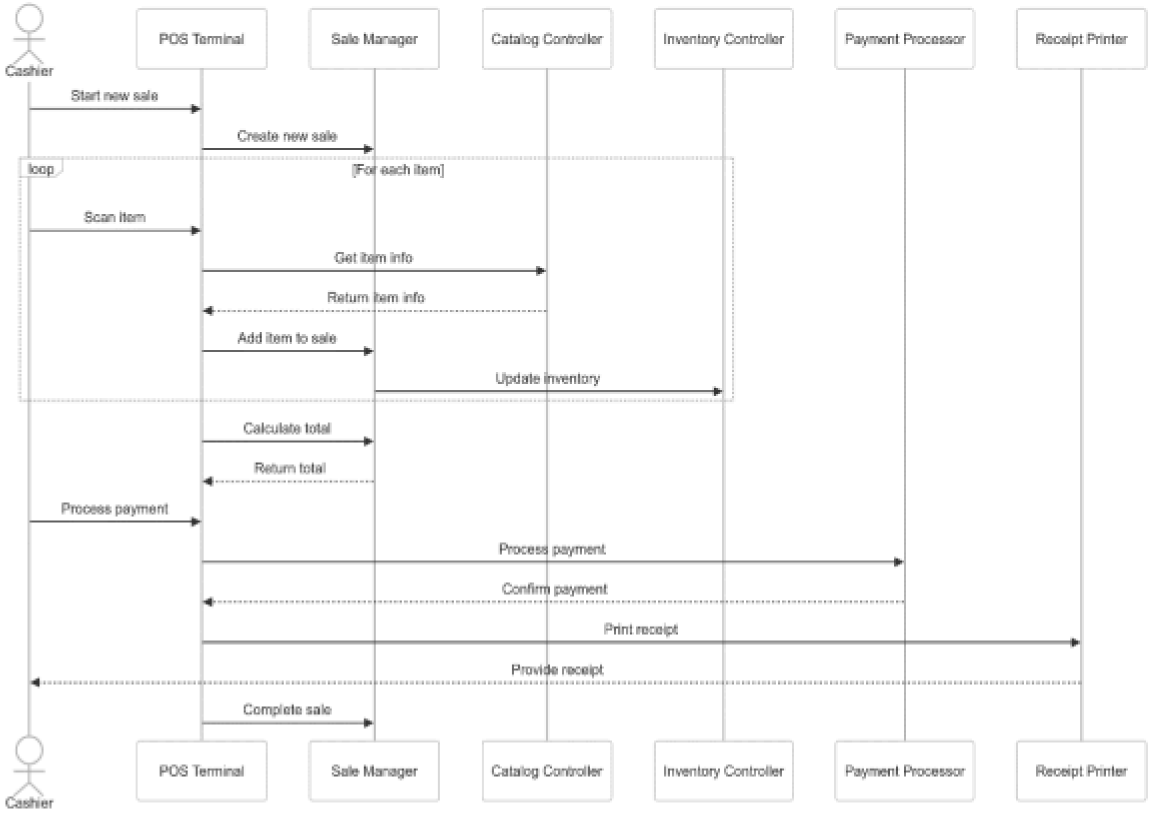
These manage the flow of the application and coordinate between entity and boundary objects.

1. **TransactionController**: Manages the process of initiating, executing, and finalizing sale transactions, including calculating totals and handling payments.
2. **ReturnController**: Handles the return process, including verifying eligibility and managing refunds.
3. **InventoryController**: Responsible for updating and managing stock levels based on sales and returns.
4. **UserController**: Manages user authentication and authorization, ensuring that cashiers and administrators have appropriate access levels.
5. **PaymentController**: Coordinates payment processing, interacting with the payment gateway and handling different payment methods.

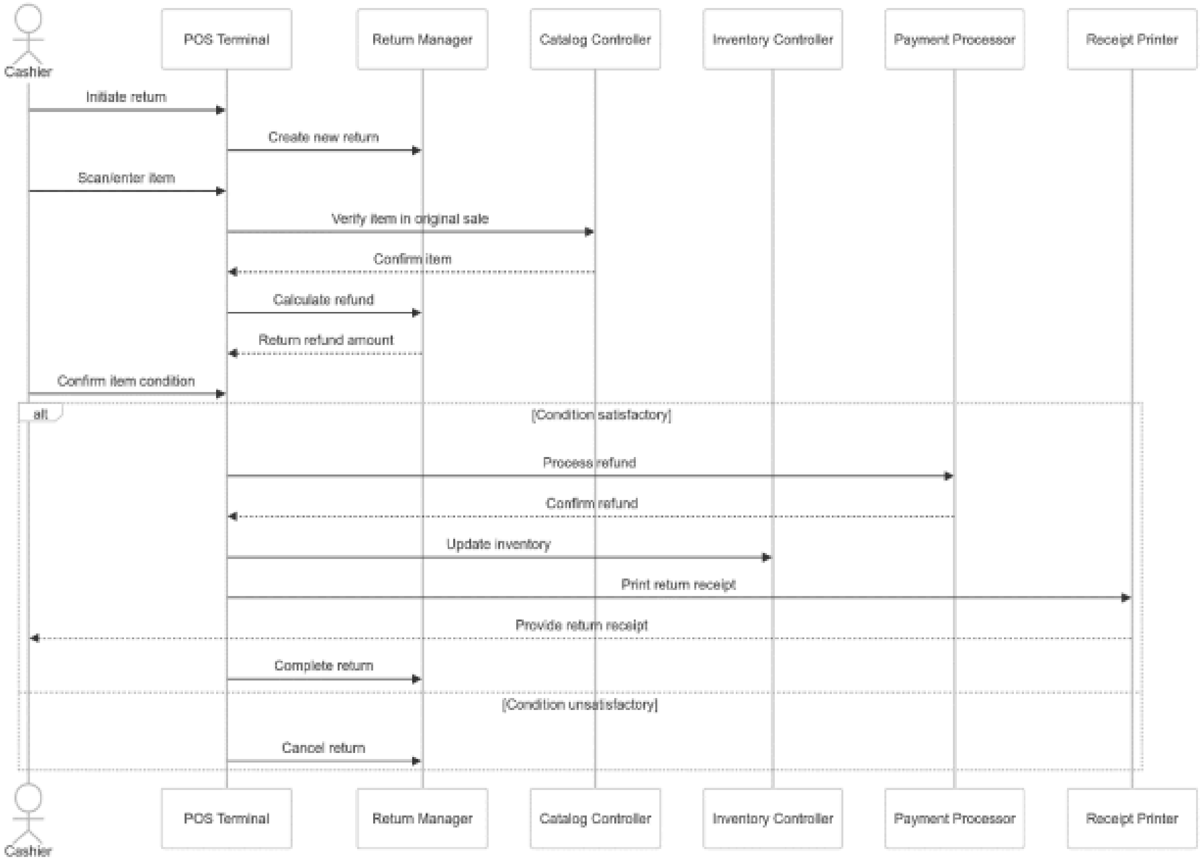
Q.) Develop Sequence Diagrams

A.)

Use Case: Process Sales

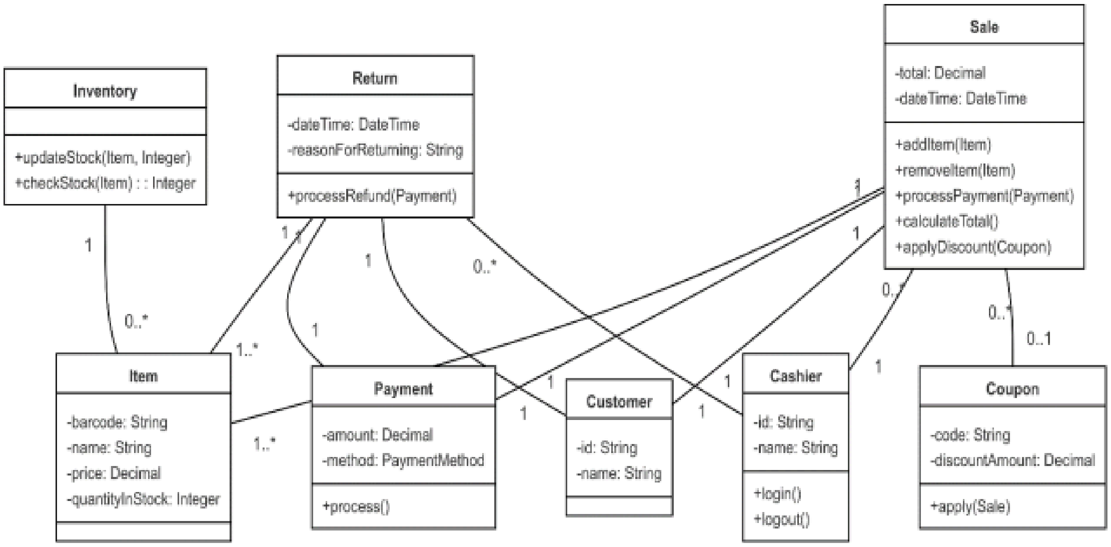


Use Case: Handle Return



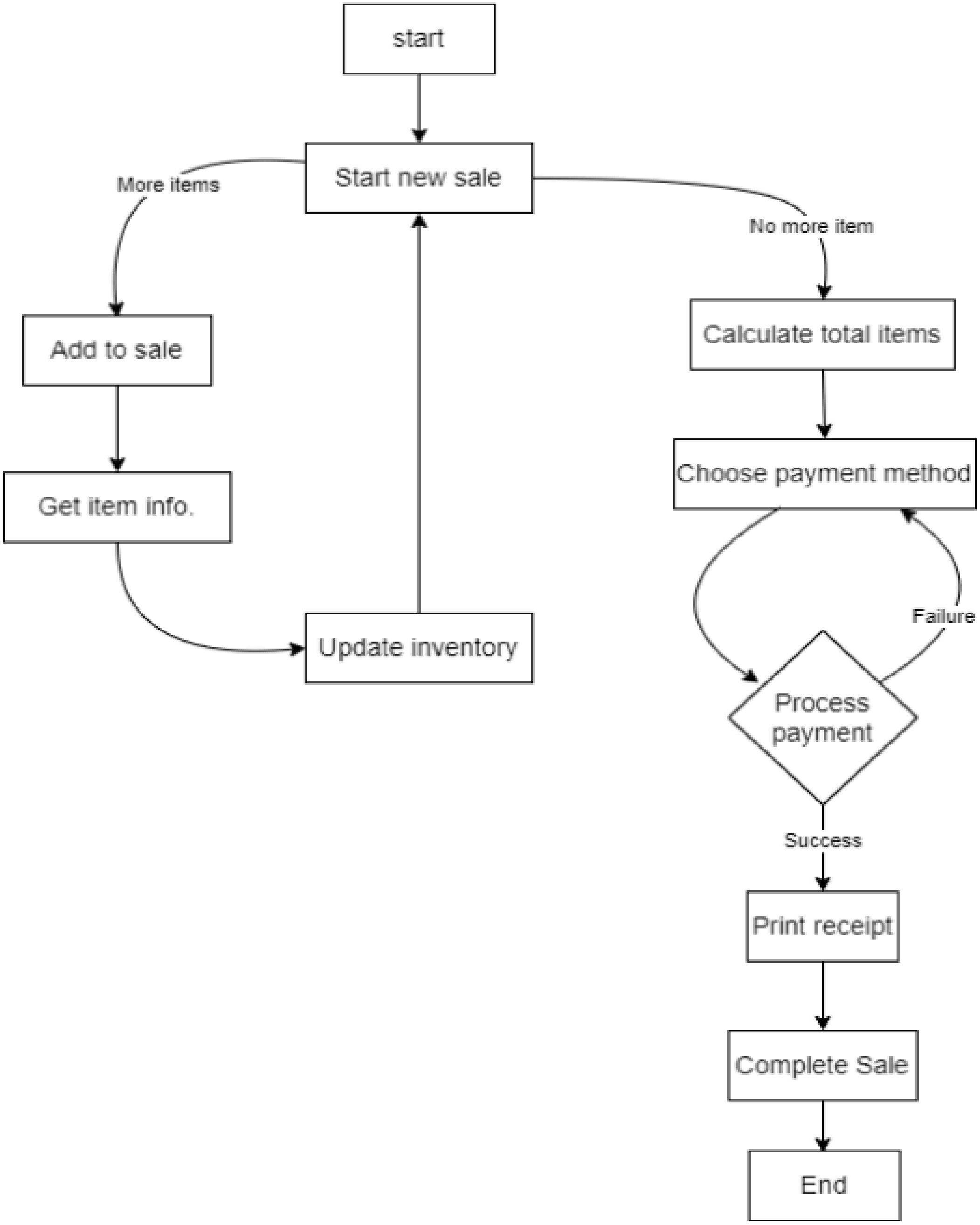
Q.) Develop Analysis Domain Models.

A.)

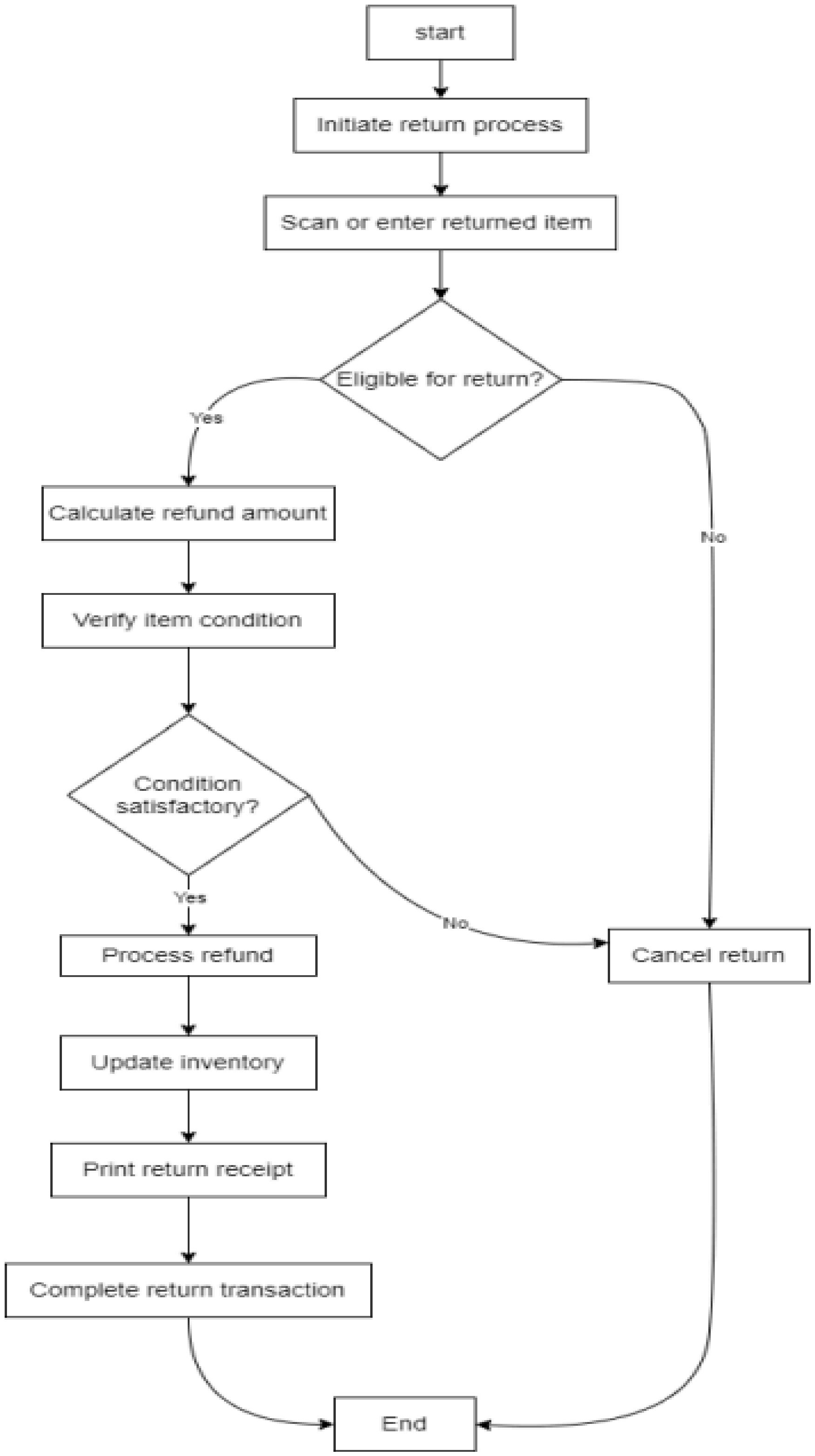


Q.) Develop activity diagrams for "Process Sale" and "Handle Return" use cases

A.) **Process Sales**



Handle returns


start
Initiate return process
Scan or enter returned item
Eligible for return?
Yes
No
Calculate refund amount
Verify item condition
Condition satisfactory?
Yes
No
Process refund
Cancel return
Update inventory
Print return receipt
Complete return transaction
End