IT314 - Software Engineering
Lab 6

Name: Bhavya Shah

ID: 202201366

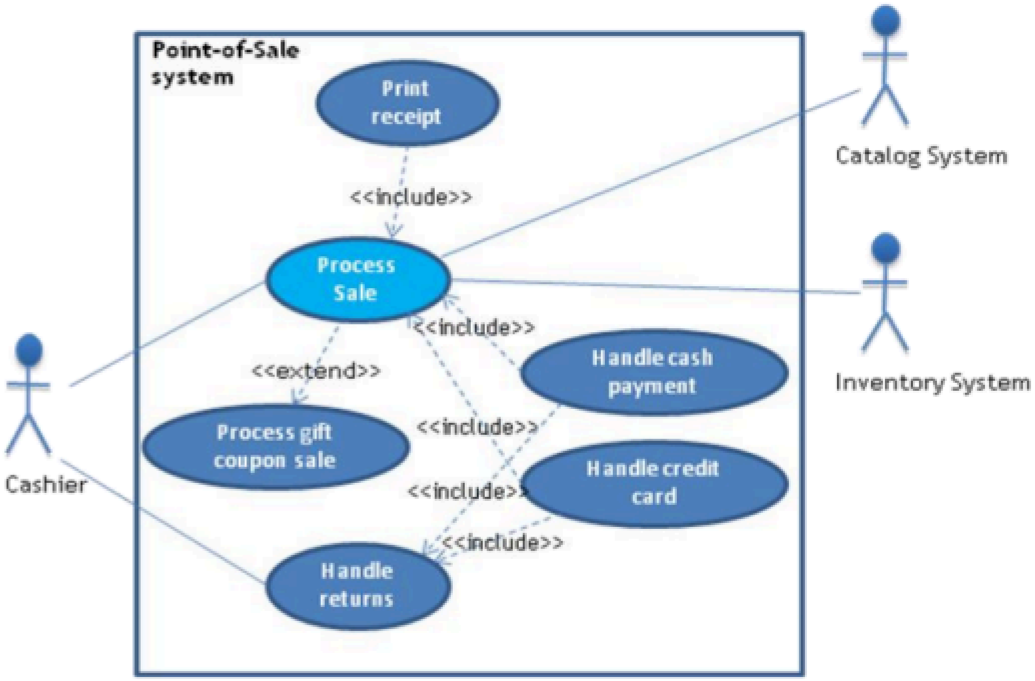
A Problem Description

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.



Process Sale:

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

   Use Case: Process Sale

   Actors: Cashier

   Preconditions:

   - The cashier is logged into the in-store Point of Sale (POS) system.
   - The system has the latest inventory and pricing information.
   - Payment systems (like card readers) are connected or offline-ready for cash transactions.

   Basic Flow:

   1. Customer arrives at the checkout point with products.
   2. The cashier scans or manually enters the items being purchased.
   3. The POS system retrieves the price, description, and stock status from the stored database.
   4. The cashier confirms the total price, including any applicable taxes or discounts.
   5. The system calculates the final amount based on locally stored tax and discount rules.
   6. The cashier tells the final amount to the customer
   7. Customer selects a payment method and the cashier processes the payment.

8. Once the payment is successful, the system deducts the sold items from the inventory.
9. The system prints a receipt for the customer.
10. The sale is recorded in the database.

Post Condition:

- The sale is logged in the POS system.
- Inventory is updated to reflect the sold items.

Alternate Flow:

2.1 Barcode Scan Error: System prompts an error and cashier manually enters the code.

2.2 Remove an item: Cashier removes the item from the list and bill is updated.

4.1 Amount mismatch: The customer updates

7.1 Promotional Coupons: Customer presents a coupon to the cashier and cashier enters/scans the code and amount is updated.

7.2 Payment Failure: The customer's card payment is declined, and the cashier requests an alternative payment method.

2. Identify Entity/Boundary Control Objects

Entity Objects:

- Product
- Inventory System
- Cashier

- Customer
- Receipt
- Payment

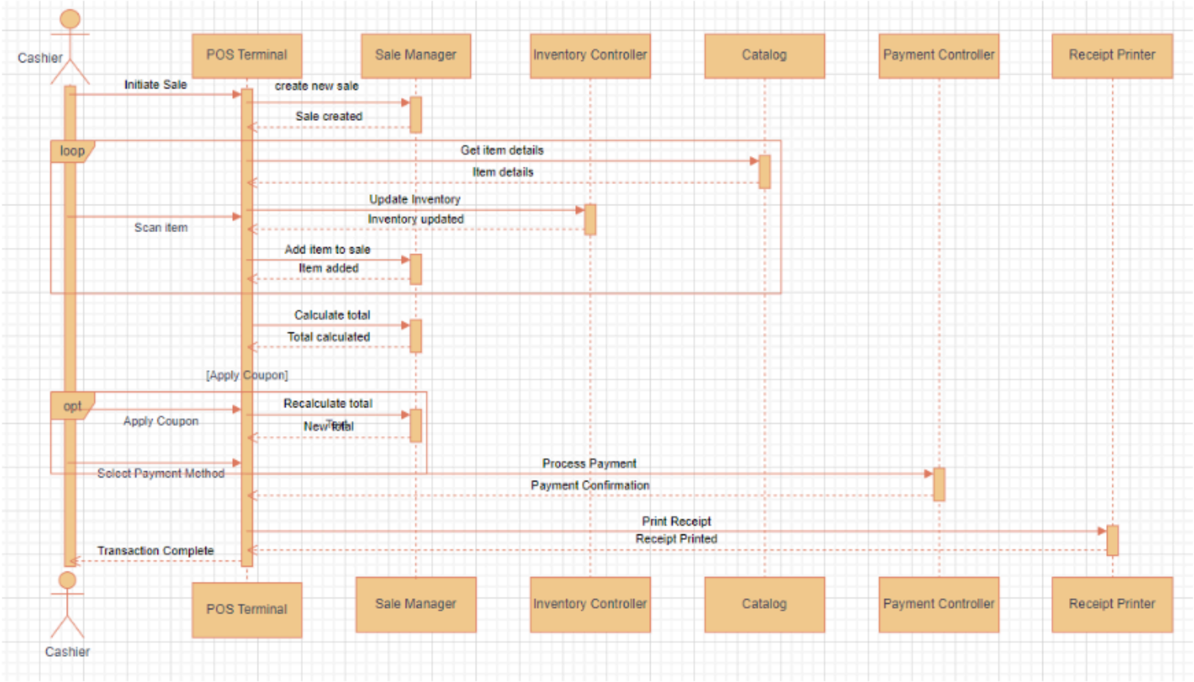
Boundary Objects

- POS Interface
- Barcode Scanner
- Card Reader
- Receipt Printer
- Display

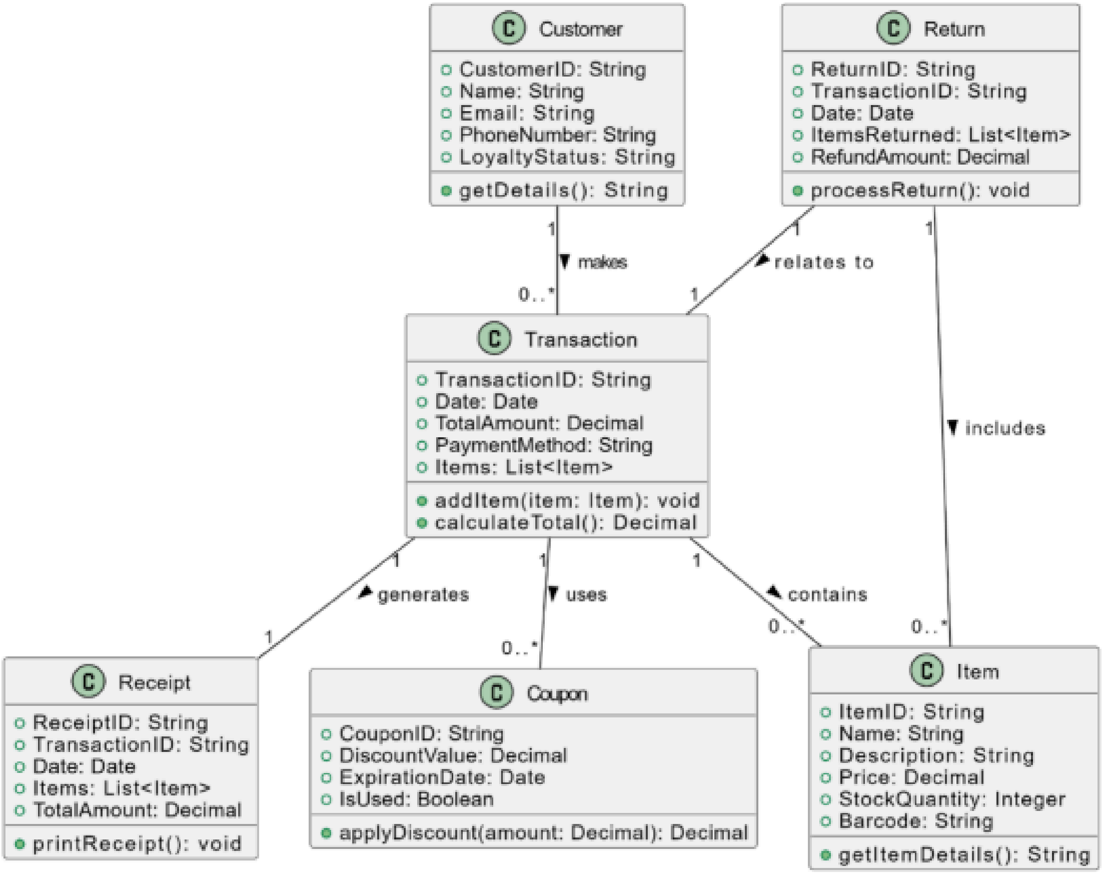
Control Objects

- Inventory Manager
- Payment Processor
- Sale manager
- Error Handler

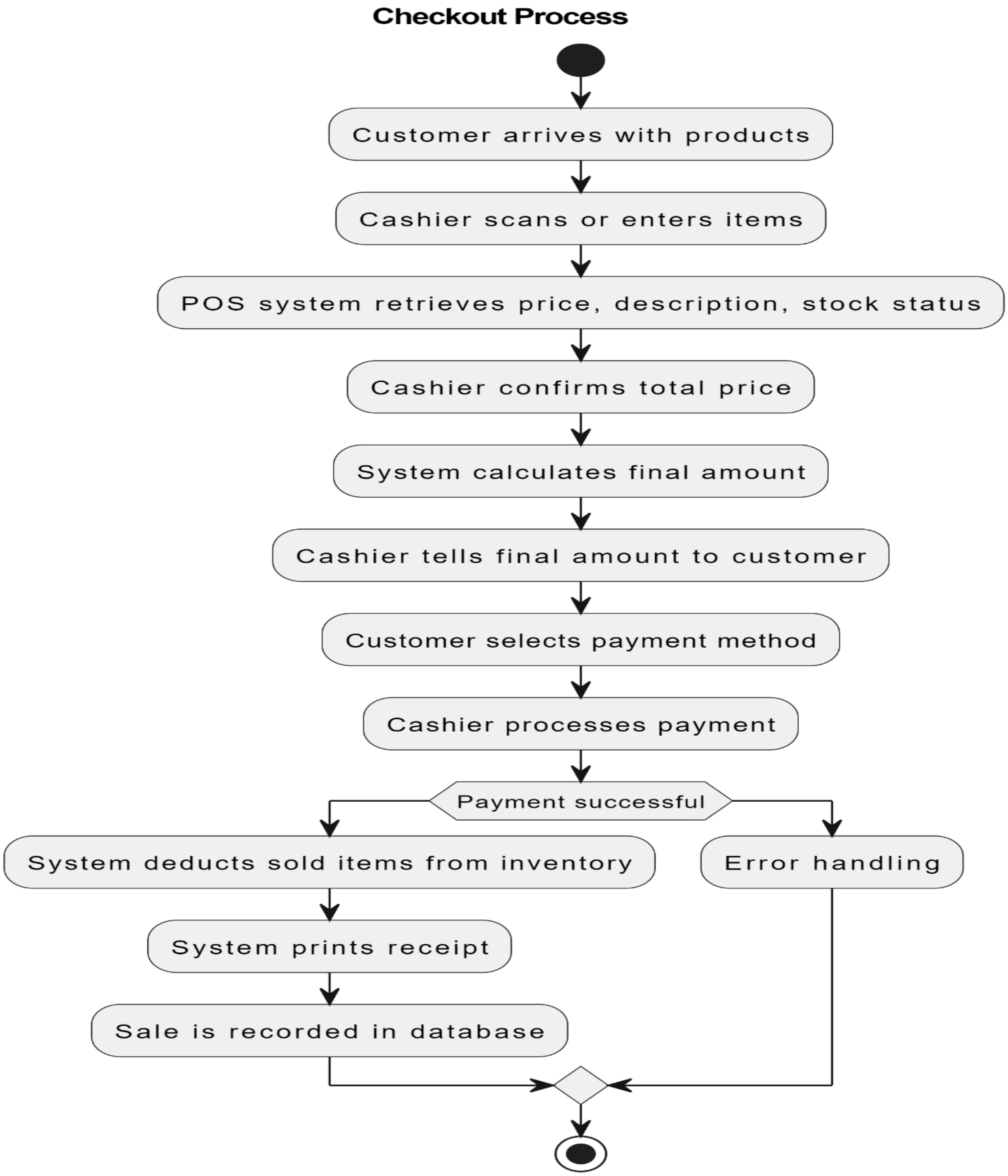
3. Develop Sequence diagram



4. Develop Analysis Domain Models



5. Develop activity diagram


Checkout Process
Customer arrives with products
Cashier scans or enters items
POS system retrieves price, description, stock status
Cashier confirms total price
System calculates final amount
Cashier tells final amount to customer
Customer selects payment method
Cashier processes payment
Payment successful
System deducts sold items from inventory
Error handling
System prints receipt
Sale is recorded in database

Handle Returns:

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

   Use Case: Handle

   Returns **Actor**: Cashier

   Precondition:

   - The POS system is operational.
   - The customer provides a valid receipt or proof of purchase.
   - Sales data is accessible.

   Flow:

   1. The customer requests a return.
   2. The cashier manually searches for the sale in the local database using the receipt or transaction ID.
   3. The system verifies if the items are eligible for return based on the store policy.
   4. The cashier confirms the return and processes it.
   5. The system calculates the refund based on the original transaction (refund will likely be cash-based since online payment methods may not be accessible).
   6. The system updates the local inventory and stores the return transaction locally.
   7. A return receipt is printed for the customer.

Postcondition:

- The return is recorded in the local database.
- Local inventory is updated to reflect the returned items.
- The system waits to sync with the central server once connectivity is restored.

Alternate Flow:

2.1 Product Not Found in the System:

The system prompts an error indicating the product cannot be found in the database, and the cashier manually verifies the purchase receipt.

3.1 No Receipt Available:

The customer does not have a receipt, so the cashier requests alternate proof of purchase (e.g., loyalty account, card transaction, etc.).

4.1 Item Condition Not Acceptable:

The item is damaged or not in acceptable return condition, and the cashier informs the customer of the return policy.

5.1 Partial Refund or Exchange:

Instead of a full refund, the customer opts for an exchange or partial refund based on the store's return policy.

6.1 Payment Method Mismatch:

The customer requests the refund via a different payment method (e.g., wants cash for a card transaction), but the system only allows the refund to the original payment method.

7.1 System Error During Refund:

A system error occurs during the refund process, and the cashier manually processes the refund or issues store credit to the customer.

2. Identify Entity/Boundary Control Objects

Entity Objects:

1. Product
2. Receipt
3. Return
4. Refund
5. Inventory System
6. Customer
7. Cashier

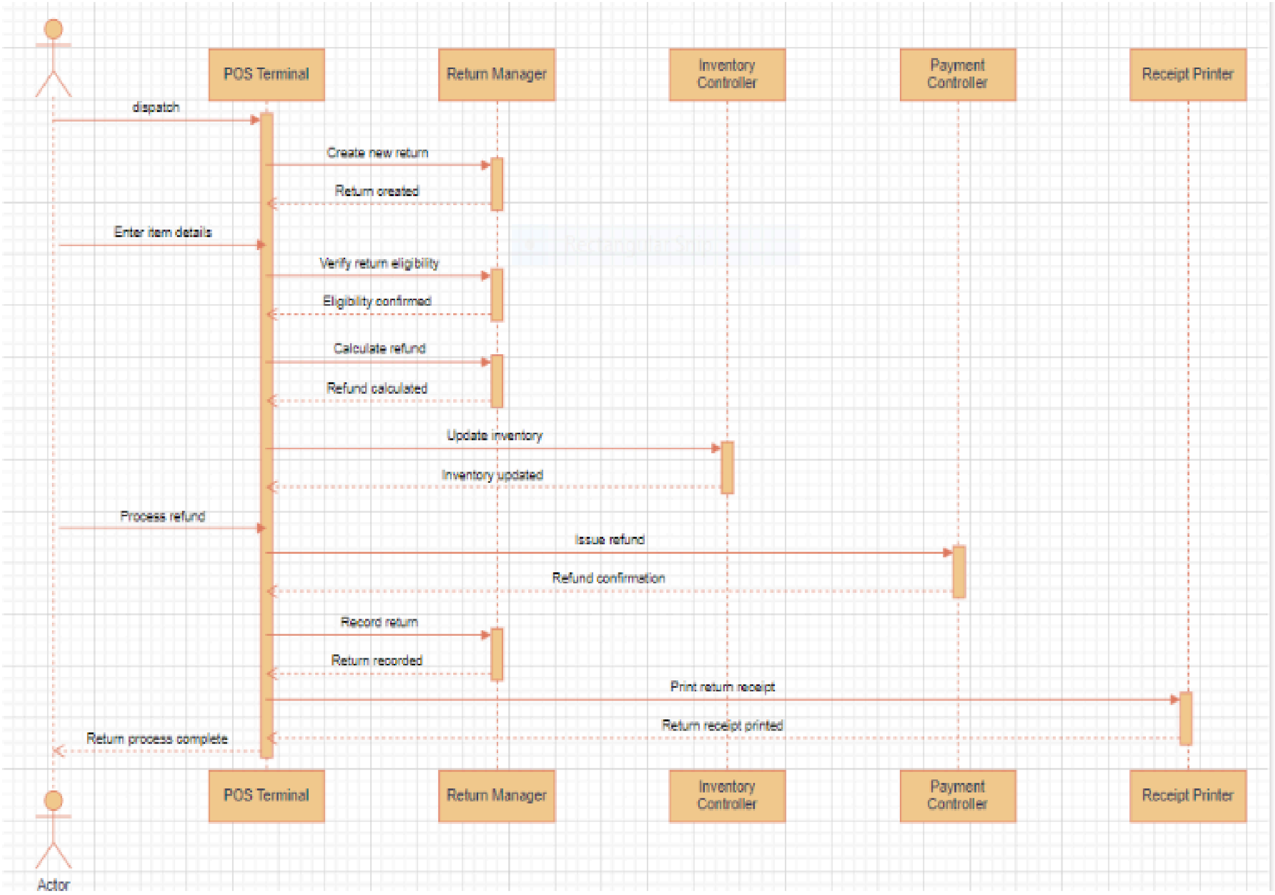
Boundary Objects:

1. POS Interface
2. Barcode Scanner
3. Display

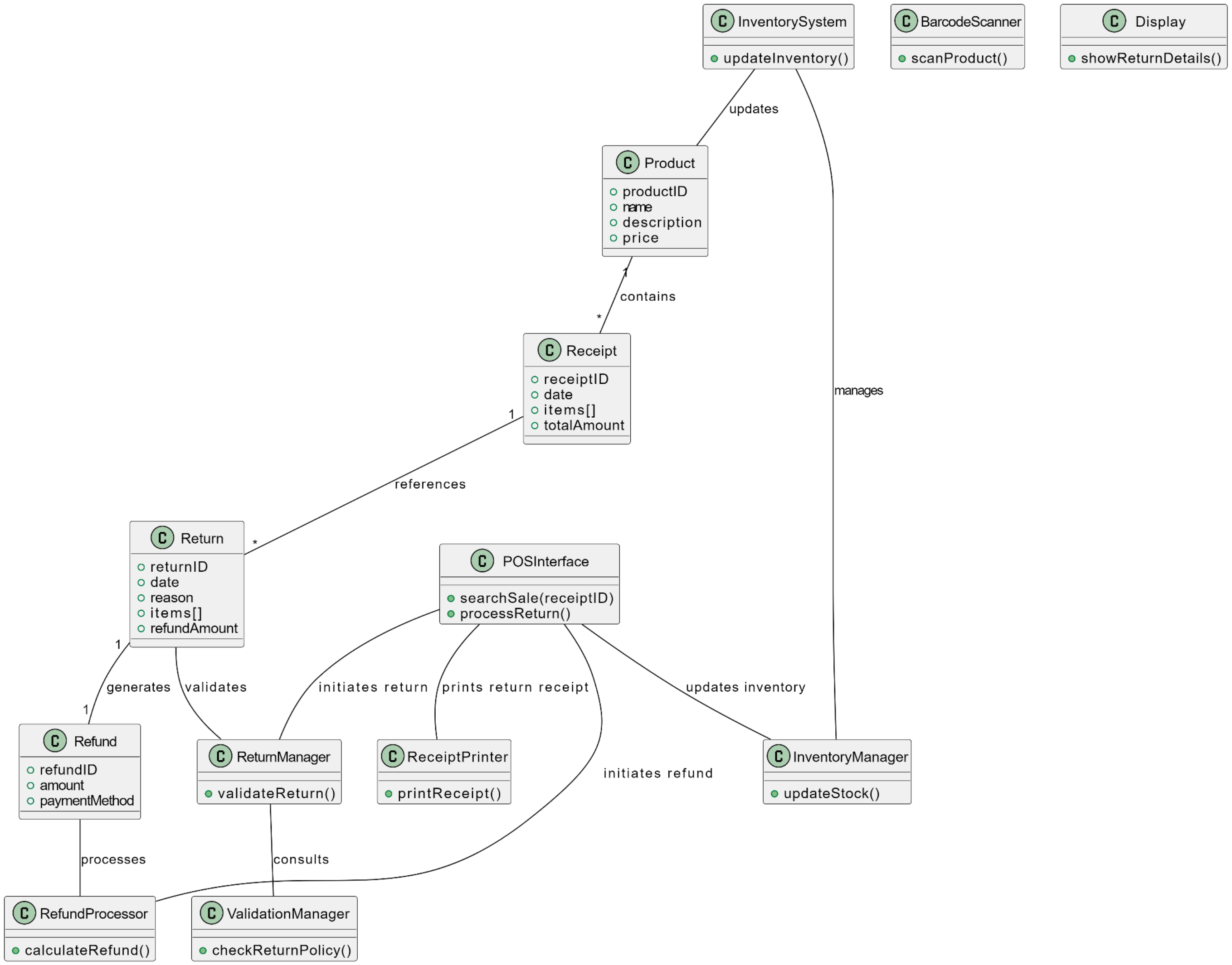
Control Objects:

1. Return Manager
2. Refund Processor
3. Inventory Manager
4. Error Handler
5. Validation Manager

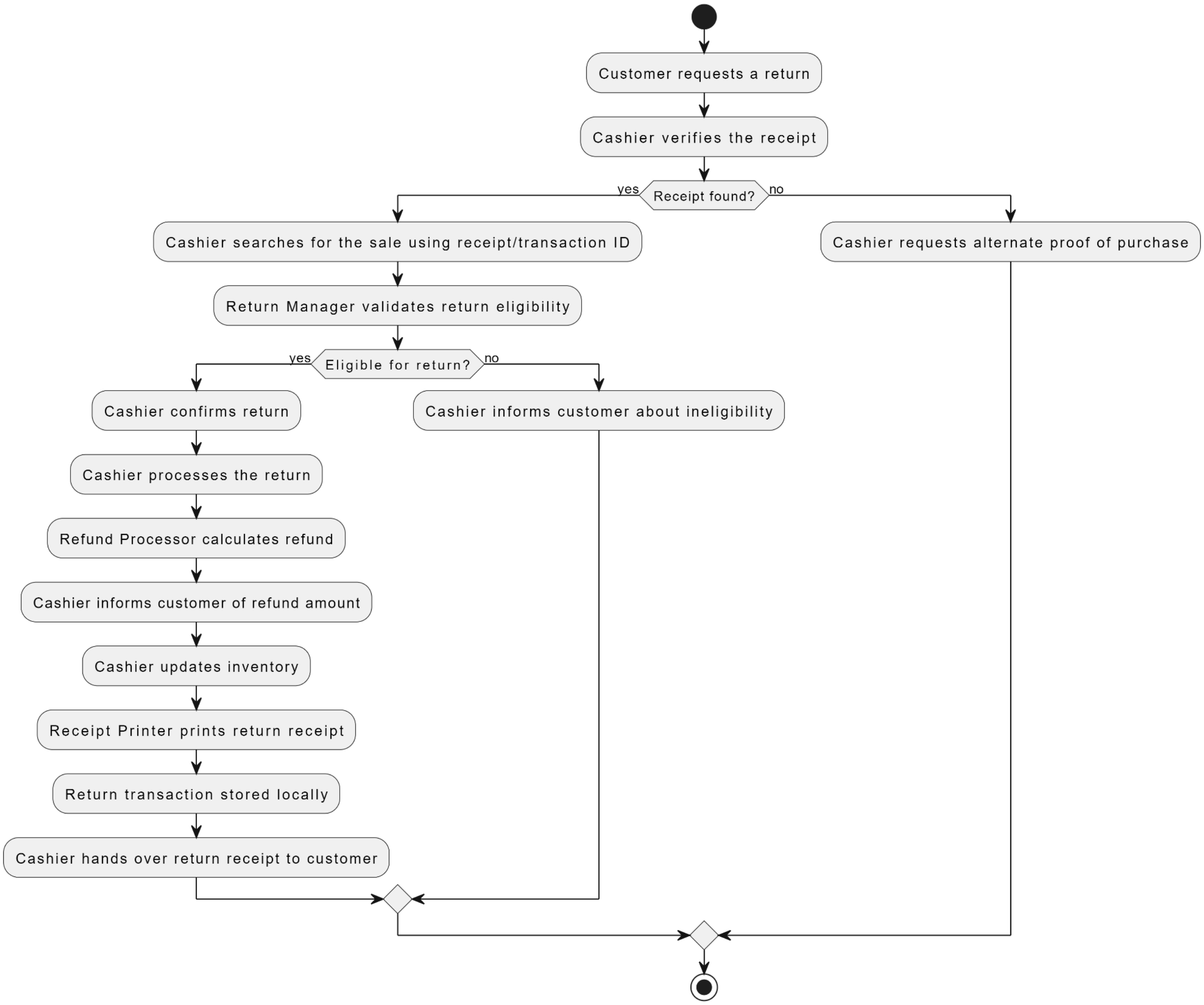
3. Develop Sequence diagram



4. Develop Analysis Domain Models



5. Develop activity diagram


Customer requests a return
Cashier verifies the receipt
Receipt found?
yes
no
Cashier searches for the sale using receipt/transaction ID
Cashier requests alternate proof of purchase
Return Manager validates return eligibility
Eligible for return?
yes
no
Cashier confirms return
Cashier informs customer about ineligibility
Cashier processes the return
Refund Processor calculates refund
Cashier informs customer of refund amount
Cashier updates inventory
Receipt Printer prints return receipt
Return transaction stored locally
Cashier hands over return receipt to customer