IT-314

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

Anuj Valambhiya
202201481

Process Sale

1.Use Case Description:

- **Description**:

This use case begins when a cashier scans a customer's goods at the POS system. The system retrieves the product information from the catalog system and checks inventory. The cashier can apply discounts or gift coupons. After the transaction is finalized, the customer can choose a payment method (cash, credit card, or check). Upon successful payment, the system prints a receipt, and the stock is updated

- **Actors**: Cashier
- **Trigger:** When the cashier scans the customer's goods.
- **Preconditions**:

The cashier must be logged into the system.

The system must be connected to the catalog and inventory systems.

- **Postconditions**:

The sale is recorded.

Stock is updated in the inventory system.

The receipt is printed.

Payment is completed.

- **Basic Flow**:

  Customer brings goods to the cashier.

  Cashier scans items.

  System retrieves product information and checks stock.

  Cashier applies gift coupons (optional).

  Customer selects a payment method.

  System processes the payment.

  Receipt is printed, and sale is finalized.

2. Identify Entity/Boundary Control Objects

Entity Objects:

- **Product**:

  Represents the physical goods available for sale in the store. It is critical in the POS system because it is the item the customer wishes to purchase. The product holds information like its identification, description, and availability, and interacts with the catalog and inventory systems.

- **SaleTransaction**:

  A record of the entire sale process. It logs the products purchased, the total amount, the method of payment, and other details related to the transaction. It interacts with various components of the system, including inventory, payment gateways, and receipt generation.

- **Receipt**:

  Represents a printed or digital confirmation of the sale. It is typically issued to the customer as proof of purchase, detailing the items bought, amounts paid, and payment method. The receipt also links to promotional discounts or gift coupons used during the sale.

- **Inventory**:

  A system that tracks the stock levels of each product. During a sale, it is responsible for adjusting the stock of the products sold and ensuring the cashier and customer are informed if an item is out of stock

Boundary Objects:

POS System Interface:

- The graphical interface used by the cashier to interact with the POS system. It allows scanning of products, managing payments, applying discounts, and printing receipts. The interface acts as a central hub for the cashier's interactions with the underlying systems.

Catalog System Interface:

- Retrieves and provides information about the products, such as names, prices, and descriptions, when a product is scanned. It is closely tied to the inventory system to ensure accurate product details are available at the POS.

Inventory System Interface:

- Interfaces with the back-end to monitor and manage product stock. It ensures that stock levels are updated in real-time, preventing sales of out-of-stock items.

Payment Gateway Interface:

- Facilitates the payment process, whether through cash, card, or check. It ensures the transaction is securely processed and completed before issuing a receipt. It also handles scenarios where a payment fails and prompts for retrying or canceling the sale.

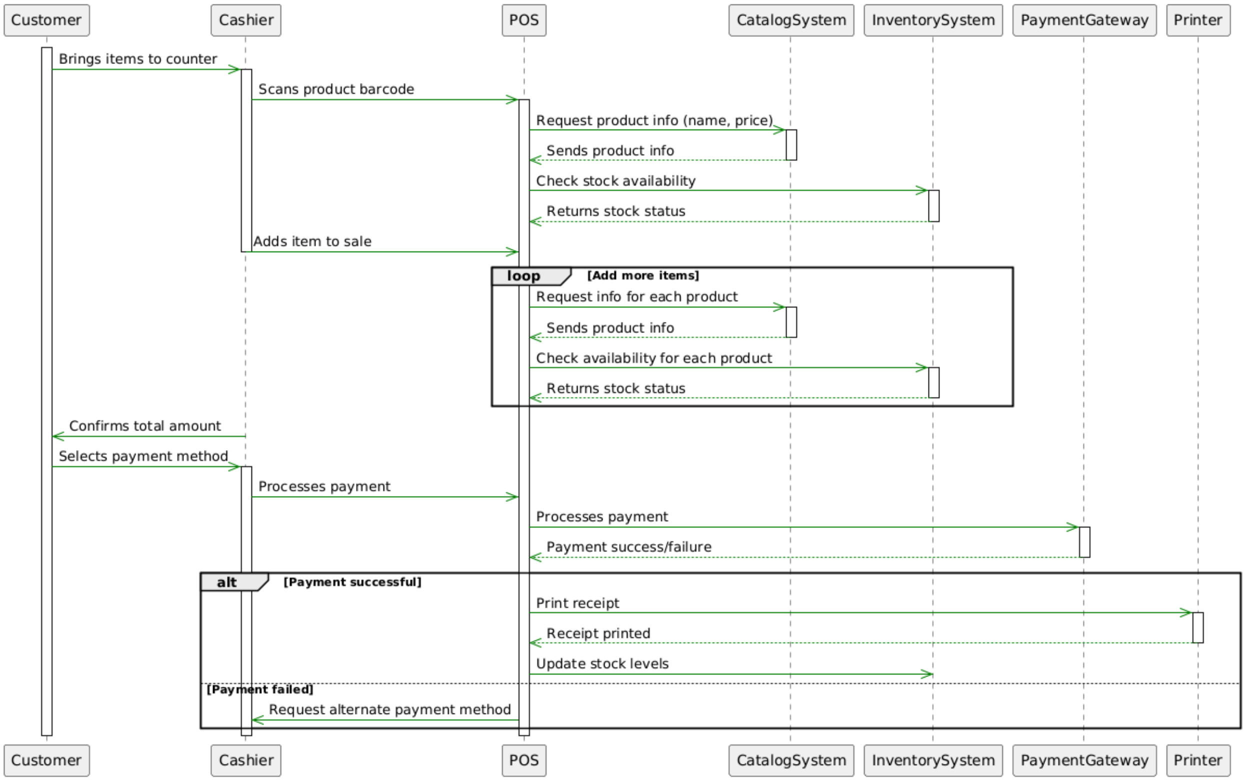
Printer Interface:

- Prints out the receipt once the transaction is complete. The printer acts as the final boundary object that delivers proof of the completed sale to the customer.

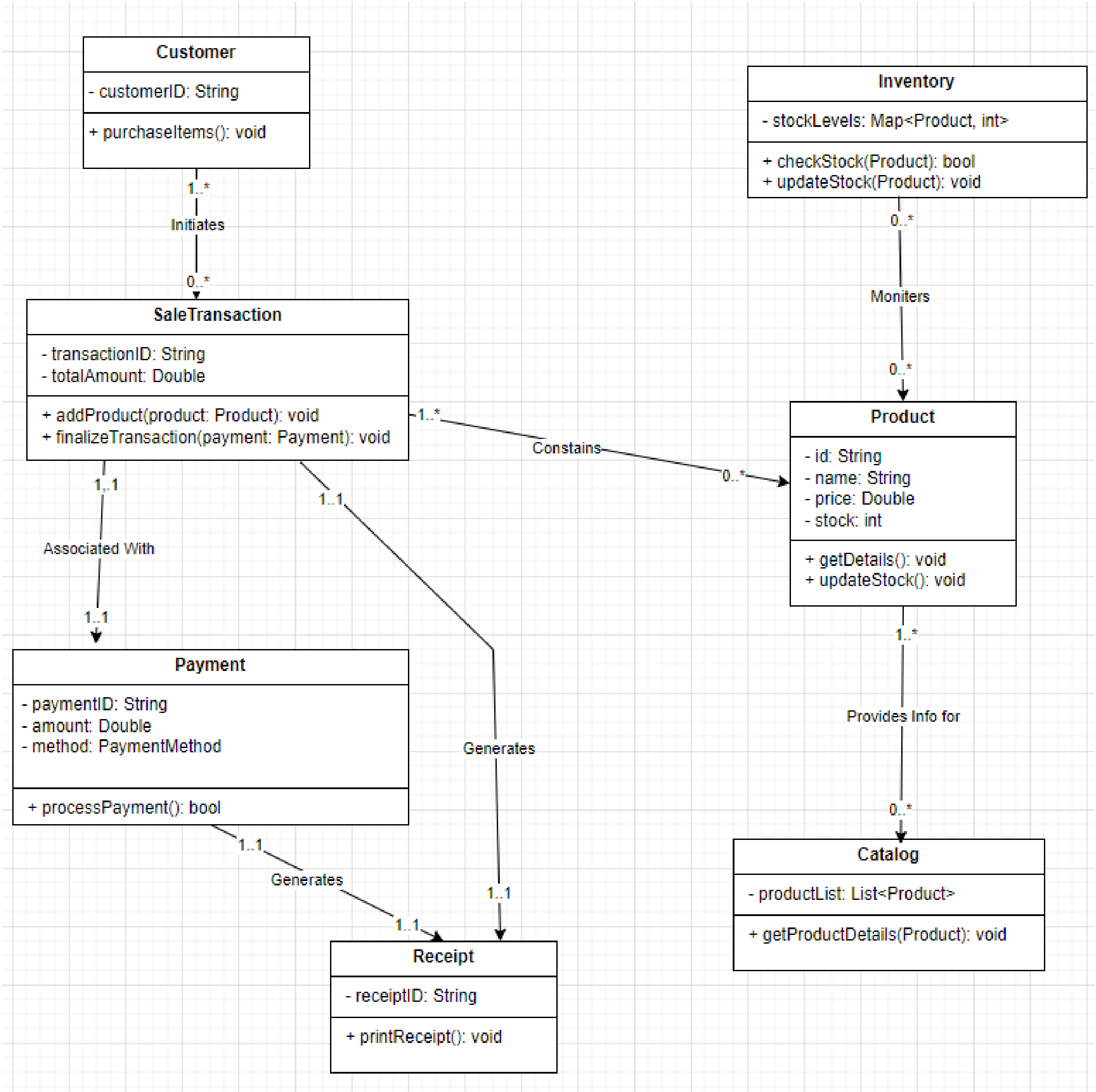
Control Objects:

1. **Sale Controller**:
   - Coordinates the entire sales process, starting from scanning items to finalizing the transaction. It is responsible for invoking methods from the catalog and inventory systems to retrieve product data and manage stock. It also interfaces with the payment gateway to handle payments and the printer to issue receipts.
2. **Payment Controller**:
   - Manages the payment process within the sale. It interacts with the payment gateway interface, checks for successful transactions, and handles exceptions like payment failures. If the payment is successful, it signals the completion of the sale process.

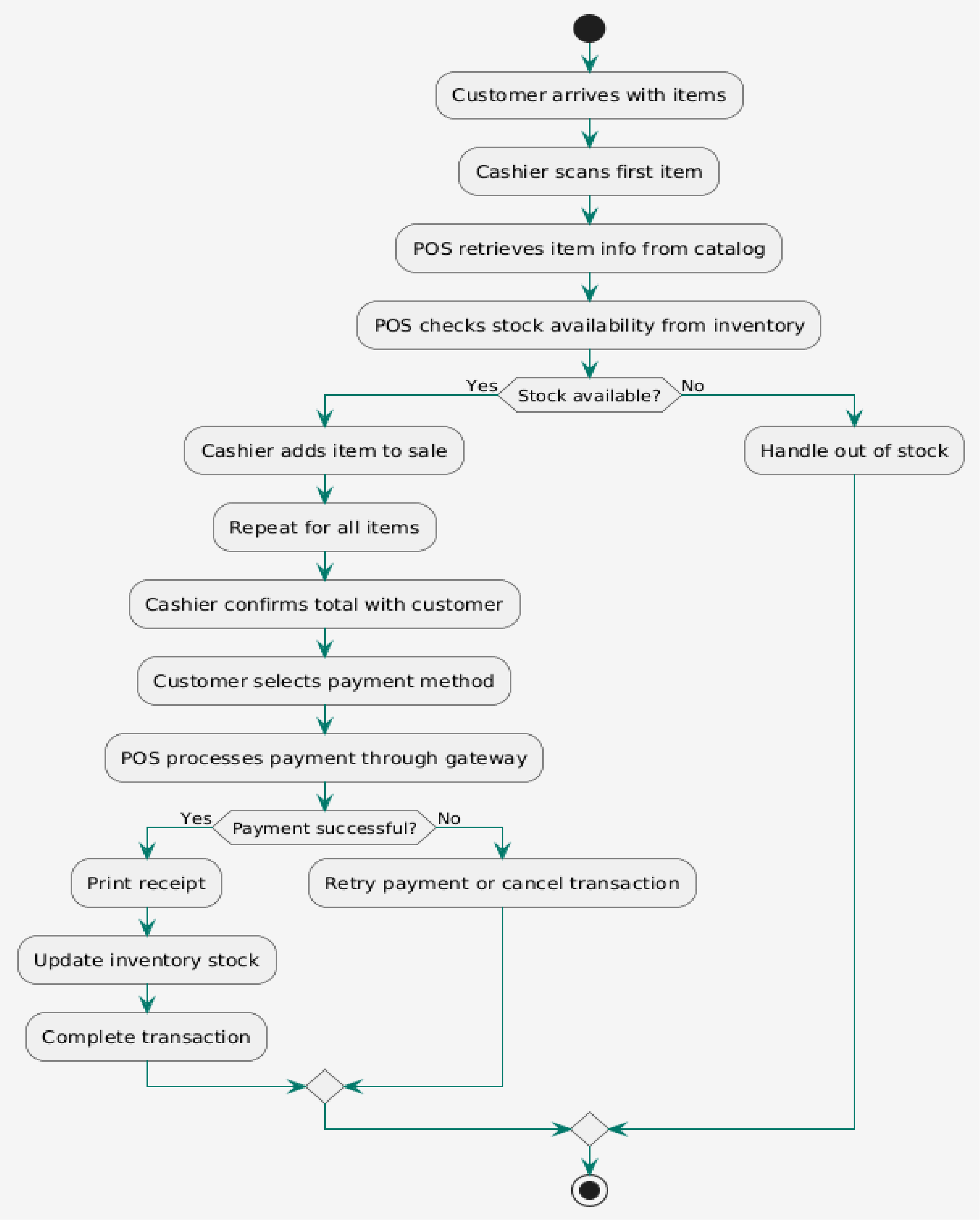
3. Sequence Diagram:



4. Analysis Domain Model:



5. Activity Diagram:


Customer arrives with items
Cashier scans first item
POS retrieves item info from catalog
POS checks stock availability from inventory
Yes
Stock available?
No
Cashier adds item to sale
Handle out of stock
Repeat for all items
Cashier confirms total with customer
Customer selects payment method
POS processes payment through gateway
Yes
Payment successful?
No
Print receipt
Retry payment or cancel transaction
Update inventory stock
Complete transaction

Handle Return:

1.Use Case Description:

- **Description**:

  This use case begins when a customer requests to return a product. The cashier initiates the return in the POS system by scanning the receipt or manually selecting the product from a list. The system checks the original transaction and validates the return policy. Upon successful validation, the return is processed, and the stock is updated accordingly.

- **Actors**: Cashier, Customer
- **Trigger:** When The Customer Handles the Item that he/she wants to return
- **Preconditions**:

  The customer has a valid receipt or proof of purchase.

  The product is eligible for return according to store policies.

- **Postconditions**:

  The return is processed.

  Stock is updated in the inventory system.

- **Basic Flow**:

  Customer requests a return with a receipt.

  Cashier scans the receipt or selects the product.

  System checks the original transaction.

  System validates the return policy.

The return is processed, and stock is updated.

2. Entity, Boundary, and Control Objects :

Entity Objects:

1. **Product**:
    - Represents the item being returned by the customer. It carries details such as its identification and price. In a return scenario, it interacts with the inventory system to update stock levels and the original sale record to confirm the item's eligibility for return.
2. **ReturnTransaction**:
    - A record of the return process. It tracks the item being returned, the reason for the return, and whether the return is accepted or rejected based on the store's policy. It is linked to the original sale for verification purposes.

Boundary Objects:

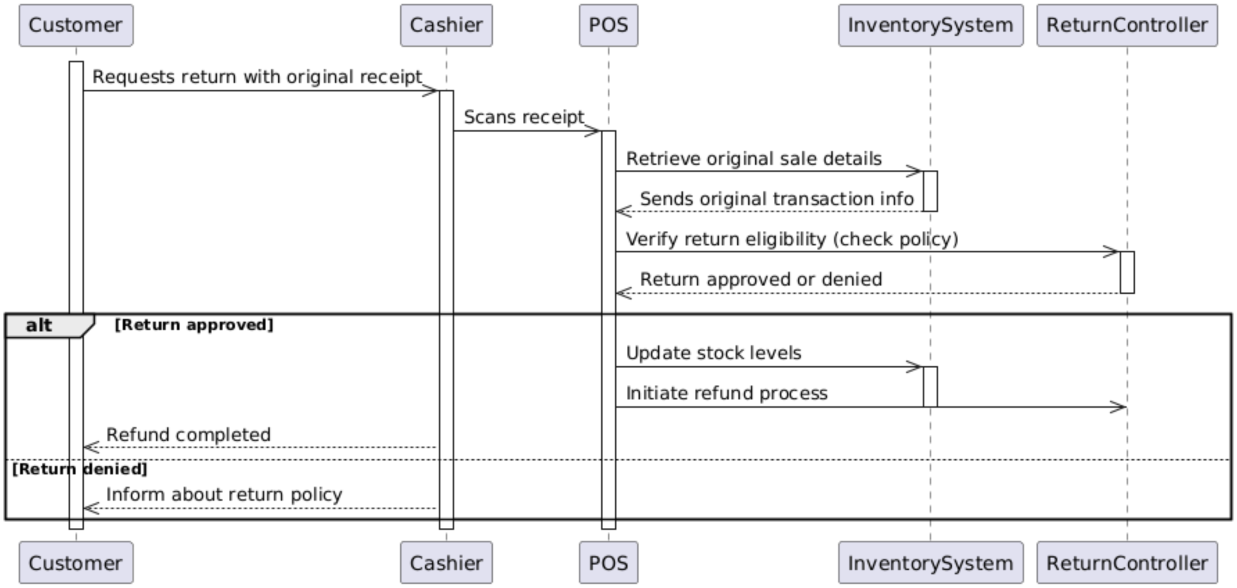
1. **POS System Interface**:
    - The interface the cashier uses to process the return. It allows the cashier to scan the original receipt, verify return eligibility, and process the refund or exchange.
2. **Inventory System Interface**:
    - Interacts with the inventory to check stock availability during the return and to update stock levels if the return is successful. It also ensures the returned item is re-entered into the stock if eligible.

Control Objects:

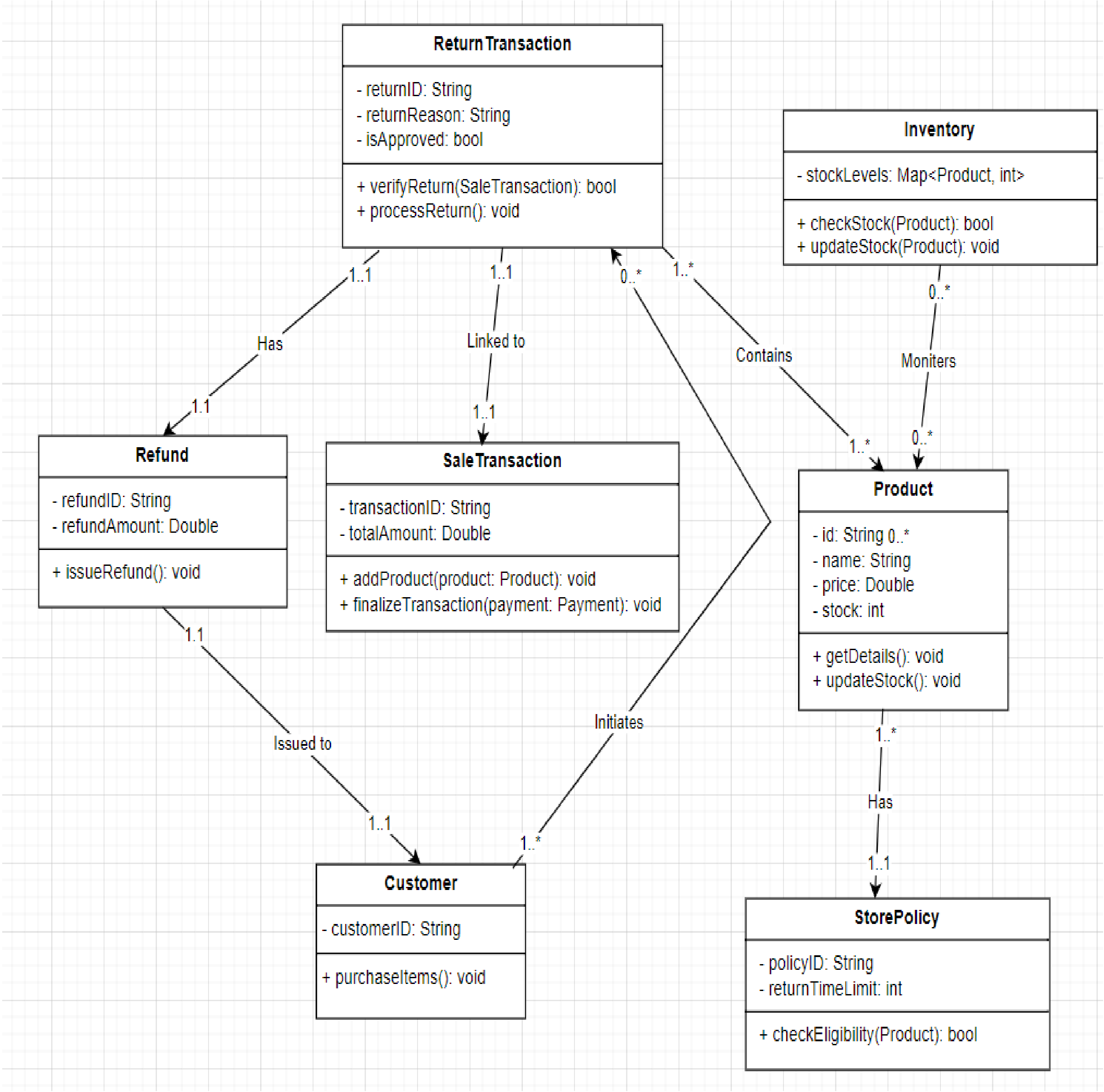
1. **Return Controller**:
    - Manages the return process by verifying the product’s eligibility for return (based on the store's policy), updating inventory after the return is

processed, and finalizing the refund or exchange. It handles any exceptional conditions, such as rejecting returns based on policy violations.

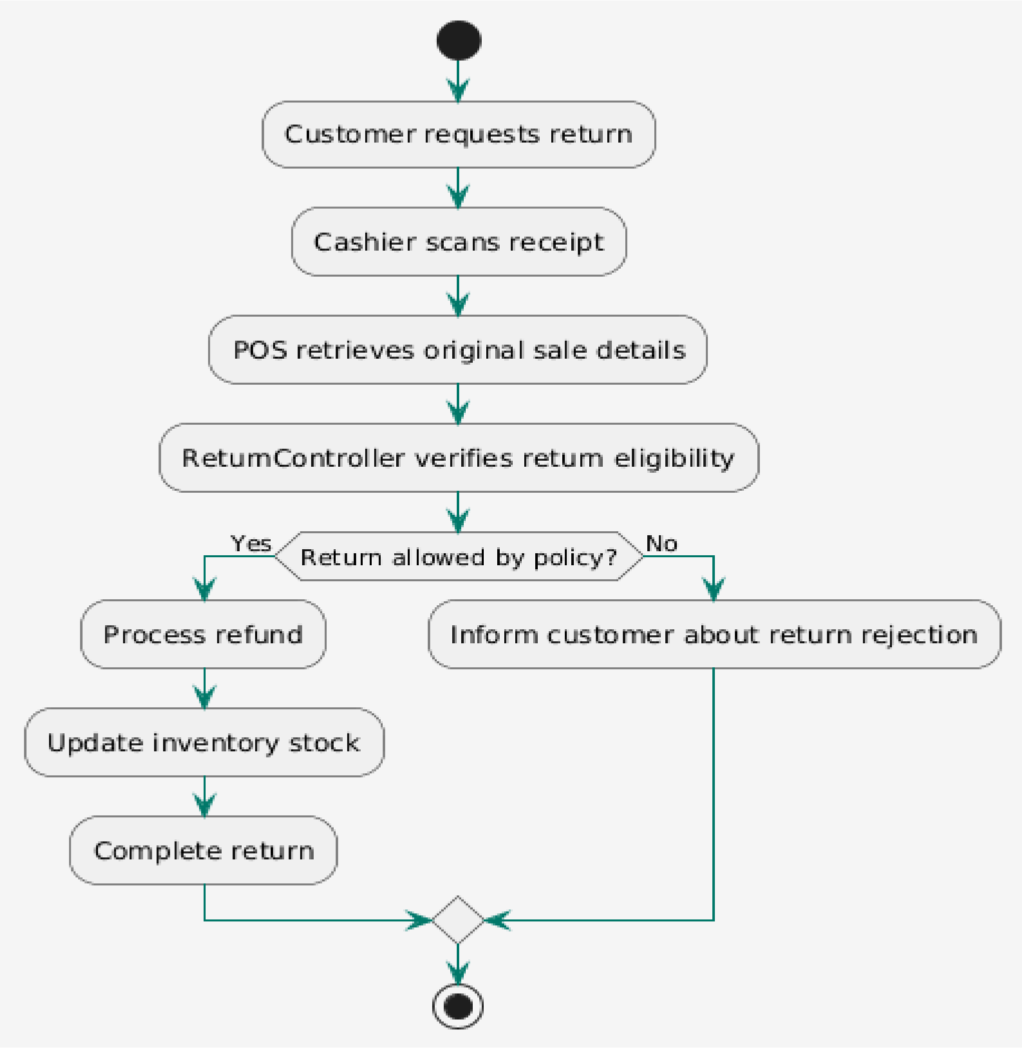
3. Sequence Diagram:



4. Analysis Domain Model:



5. Activity Diagram:


Customer requests return
Cashier scans receipt
POS retrieves original sale details
RetumController verifies retum eligibility
Return allowed by policy?
Yes
No
Process refund
Update inventory stock
Complete return
Inform customer about return rejection