IT313: Software Engineering

NAME:- Gunjan Saroliya

ID :- 202201225

Lab-6

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

1. Use Case: Process Sale

- **Primary Actor:** Cashier
- **Stakeholders:** Cashier, Customer
- **Trigger:** The customer arrives at the checkout counter with goods to purchase.
- **Preconditions:**
    - The cashier is logged into the POS system.
    - The catalog and inventory systems are online and accessible.
- **Postconditions:**
    - The sale transaction is completed.
    - The inventory is updated to reflect the goods sold.
    - A receipt is printed for the customer.
- **Basic Flow:**
    - The cashier initiates a new sale transaction.
    - The cashier scans the barcode of each item.
    - The system retrieves the name and price of the item from the catalog system.
    - The system deducts the stock amount of each item from the inventory system.
    - The cashier confirms the total sale amount.
    - The customer chooses a payment method.

- The system processes the payment.
- The system prints a receipt for the customer.

Alternate Flows:

- 3a. Item Not Found in Catalog:
  - The system notifies the cashier that the item is not found, and the cashier manually enters the price or notifies the customer.
- 6a. Insufficient Stock:
  - The system notifies the cashier that an item is out of stock, and the cashier either cancels the transaction or removes the item from the sale.
- 7a. Payment Fails:
  - If the payment fails, the cashier asks the customer for an alternative payment method.
- 7b. Gift Coupon Used:
  - The cashier applies the gift coupon to the sale, reducing the total price.

➢ **Identifying Entity/boundary control objects:**

⇒ **Entity Objects:**

1. Sales Transaction Details: Stores details of item sold, payment method, additional tax, discount, etc…
2. Product: Name, date, specifications, etc…
3. Inventory: Warehouse that contains all the items
4. Catalog: Details of product item, barcodes, price
5. Payment: Method of payment, status
6. Invoice: Item(s) name, price, total amount, tax, discount applied, customer-id, etc…

⇒ **Boundary Objects:**

1. POS interface: Screen on which cashier interacts
2. Barcode scanner: Takes the input of products
3. Card Reader: Hardware that reads the card(credit, debit) information
4. Printer: prints the receipt/invoice

⇒ **Controller Objects:**

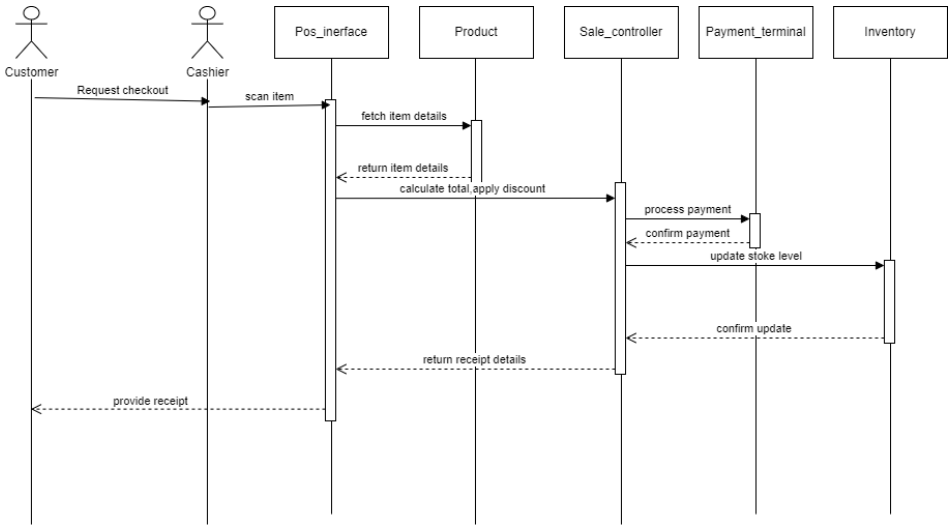
1.Sales Controller: Controls the sales process of scanning, payment, inventory updates

2. Payment Controller: Handles the payment process, checks pin, payment status

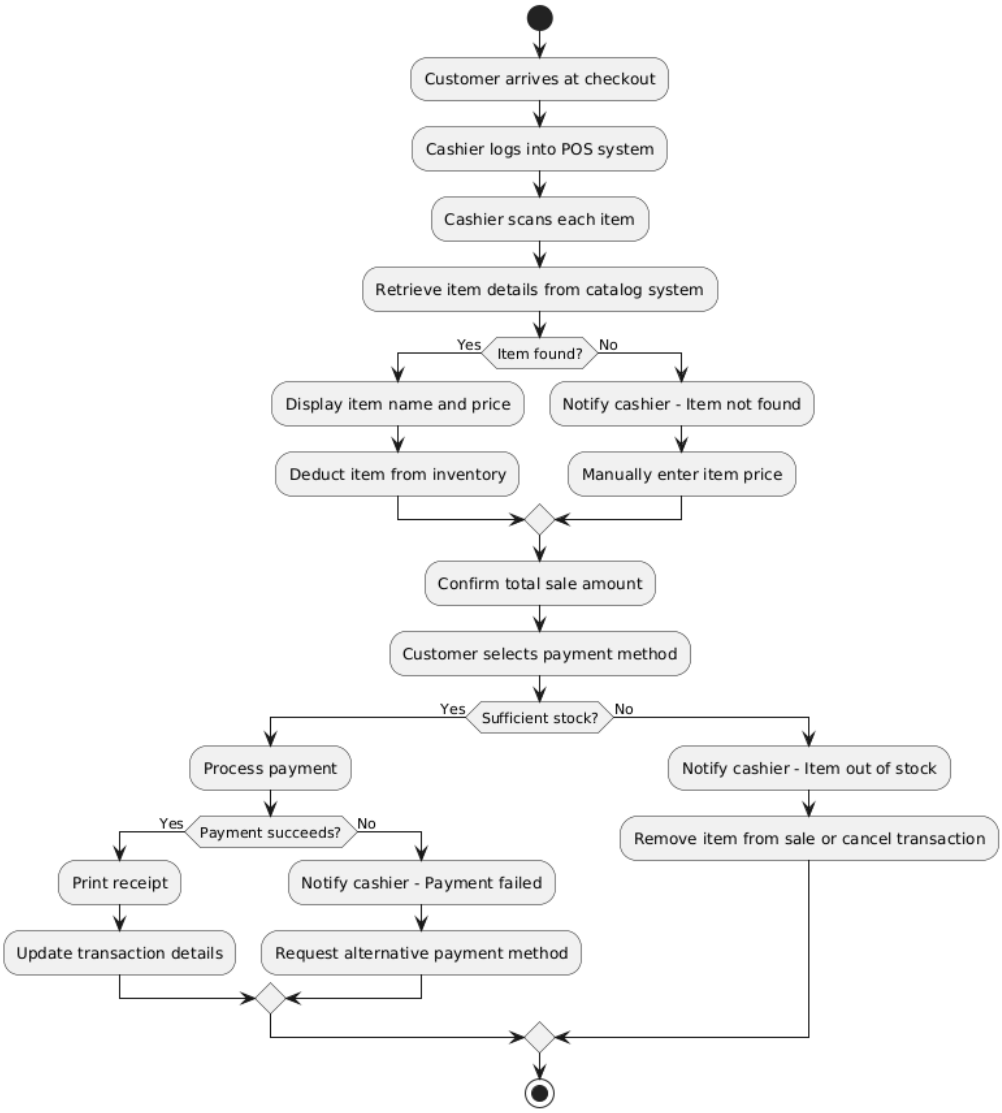
3. Inventory Controller: Does necessary updates in inventory, manage stacks of products according to purchase, creates reports of most demanding products, etc.

4.Discount Handler: Handles necessary reductions in amount when coupon applied, customer is allowed to apply coupons etc…

⇒ **Sequence diagram:**



Activity Diagram:



Use Case: Handle Return

- **Develop Use Case Textual Description for "Handle Return" use cases.**

Primary Actor: Cashier

Preconditions:

1.The cashier is logged into the POS system.

2.The customer provides the original receipt or relevant details for the return.

Trigger: The customer requests to return an item.

Main Success Scenario:

1. The cashier selects the return function in the POS system.

2. The cashier scans the returned item's barcode or enters it manually.

3. The system retrieves the sale transaction associated with the returned item based on the receipt or transaction details.

4. The system checks if the item is eligible for return based on the store's return policy (e.g., return window, item condition).

5. The cashier confirms the return, and the system updates the inventory by adding the returned item back in stock.

6. The system calculates the refund amount, including any discounts or coupons that were applied to the original purchase.

7. The customer selects a refund method (cash, credit card refund, store credit).

8. The system processes the refund and generates a receipt for the return.

9. The return transaction is completed, and the system updates the sales records.

Postconditions:

- The return is recorded in the POS system.
- The inventory is updated.
- The refund is processed and reflected in the system.

Alternate Flows:

- No Receipt: If the customer does not have a receipt, the cashier attempts to locate the original transaction using other details (e.g., date, customer info).
- Item Not Eligible for Return: If the item is ineligible for return, the system notifies the cashier, who informs the customer and cancels the return process.
- Refund Failure: If the refund fails (e.g., due to system issues), the cashier attempts an alternative refund method or directs the customer to store management.

- **Identify Entity/Boundary Control Objects:**

Entity Objects:

1.Return Transaction Details: Details of returned item, refunded amount
2. Item: Name, date, specifications, etc…
3. Inventory: Warehouse that contains all the items
4.Catalog: Details of product item, barcodes, price
5.Payment: Method of refunded payment, status
6.Invoice: Item(s) name, price, total amount, tax, discount applied, customer-id, etc…

Boundary Objects:

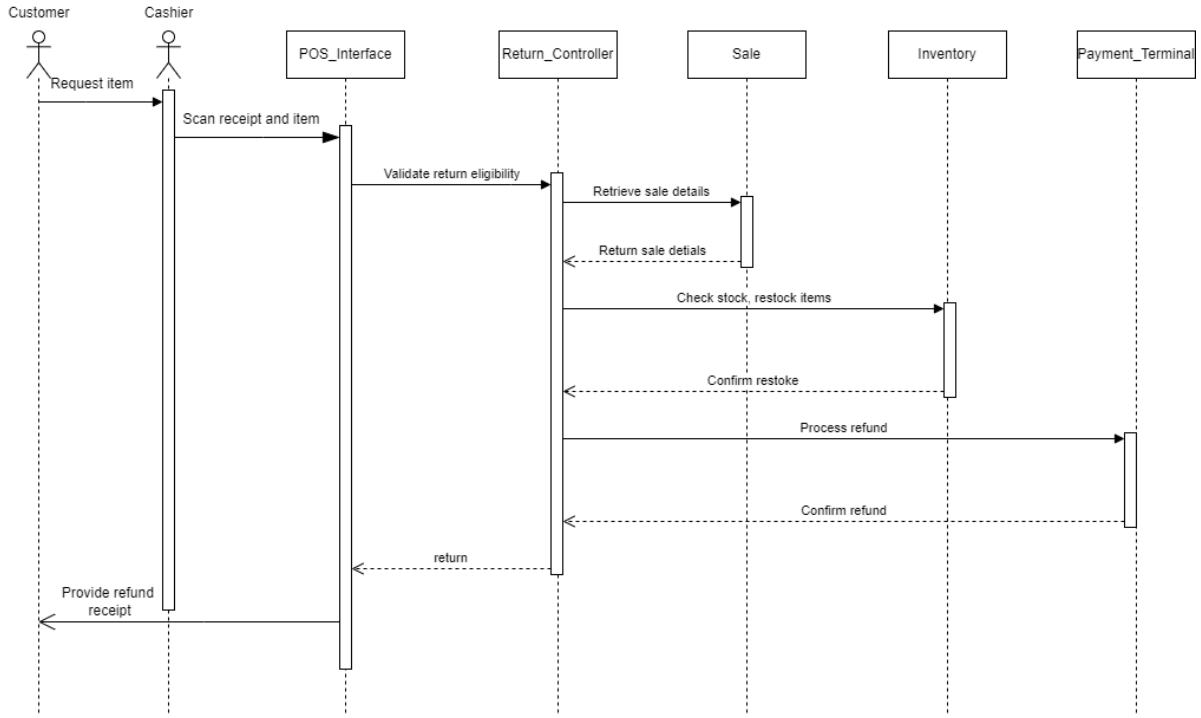
1. POS interface: Screen on which cashier interacts
2. Barcode scanner: Takes the input of products
3.Printer: prints the return receipt

Controller Objects:

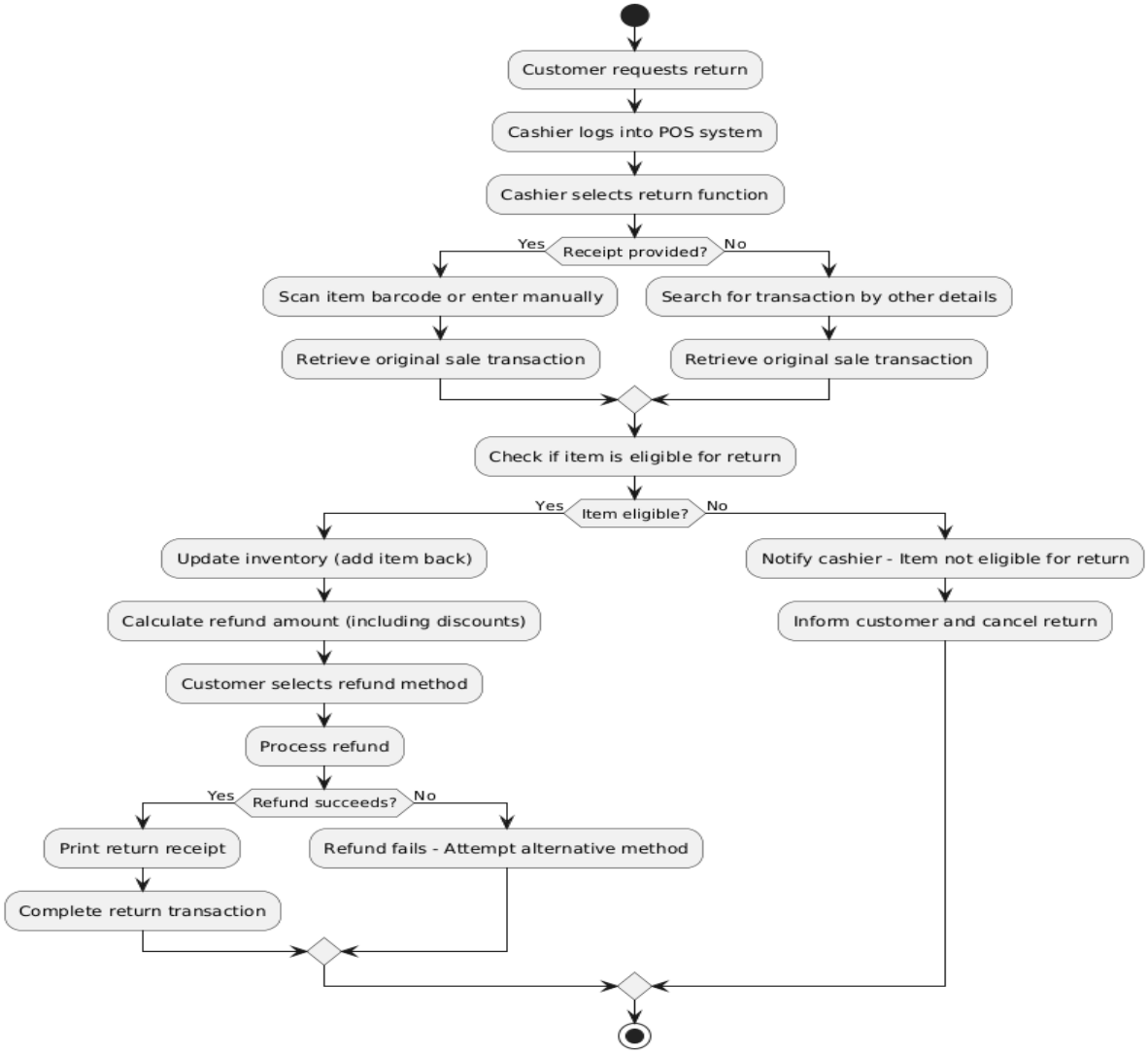
1. Return Controller: Manages return process, eligibility criteria,etc…
2. Payment Controller:Handles the processing of refunds, including verifying refund methods and payment status.
3. Inventory Controller: Does necessary updates in inventory, manage stacks of products according to purchase, creates reports of most demanding products, etc..

4.Discount Handler: Handles necessary reductions in amount when coupon applied, customer is allowed to apply coupons etc…

- **Sequence Diagram:**



➢ Activity diagram:



➢ Analysis Domain Models(for both):