IT-314
SOFTWARE ENGINEERING
LAB-6

-202201071
Nirva Patel

Use Case: Process Sale

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

Primary Actor: Cashier
Preconditions:
1.The cashier is logged into the POS system.
2.The POS system is connected to the backend catalog and inventory systems.

Trigger: The customer arrives at the checkout counter with goods to purchase.

Main Success Scenario:

1. The cashier initiates a new sale transaction in the POS system.
2. The cashier scans the barcode of each item, and the system retrieves the item name and price from the backend catalog system.
3. The system updates the inventory by reducing the stock amount of the scanned items.
4. The cashier can apply any active gift coupons if presented by the customer, and the system calculates the discount accordingly.
5. The system calculates the total amount due.
6. The customer selects a payment method (cash, credit card, check).
7. The system processes the payment and confirms its success.
8. The system generates and prints a receipt for the customer.
9. The sale transaction is completed, and the system updates the sales records.

Postconditions:

- The sale is recorded in the POS system.
- The inventory is updated.
- The payment is processed and reflected in the system.

Alternate Flows:

- **Invalid Item Barcode**: If an item's barcode is invalid or not found in the system, the cashier manually enters the item information or notifies the customer.

- **Insufficient Stock**: If an item is out of stock, the system notifies the cashier, who informs the customer and removes the item from the sale.
- **Payment Failure**: If the payment fails, the system prompts the cashier to retry or use a different payment method.

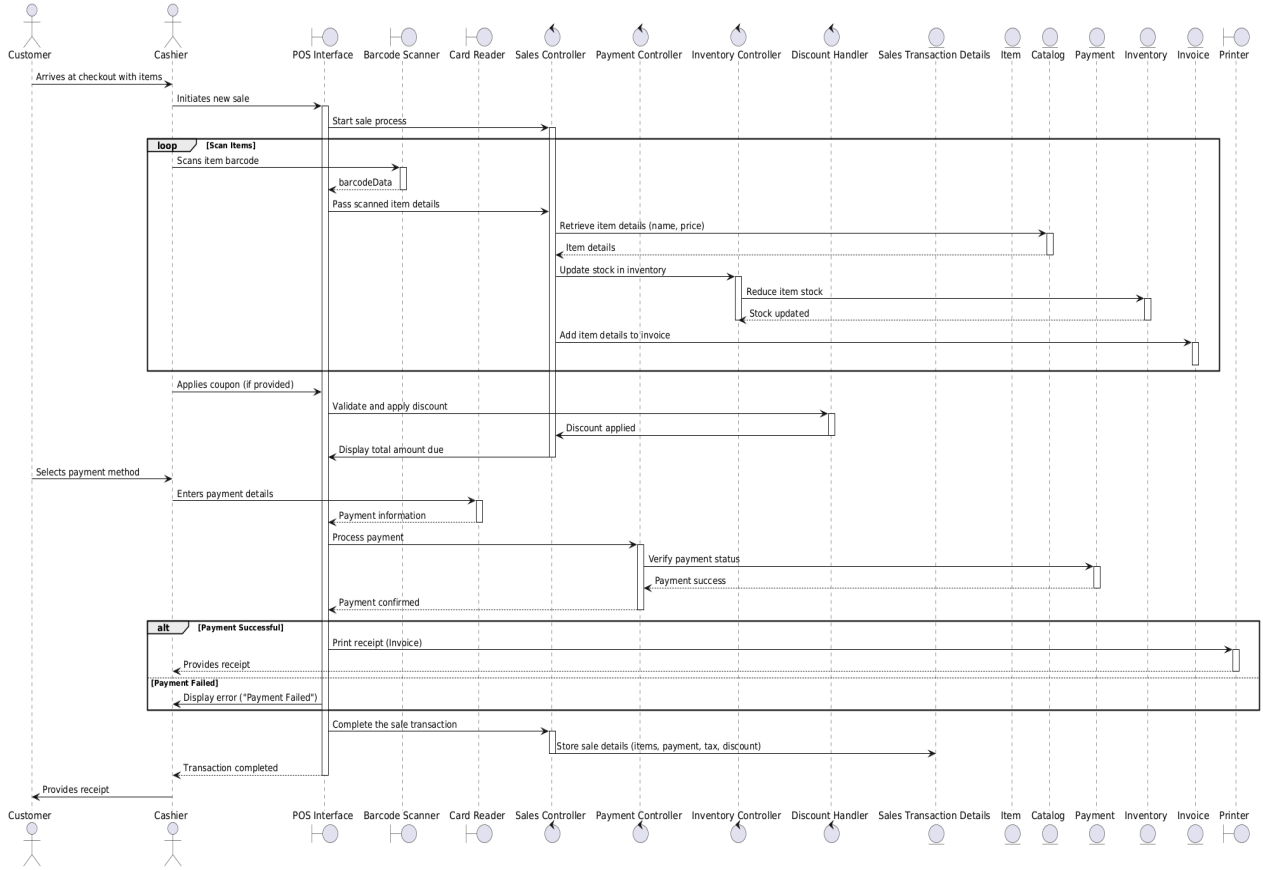
➢ **<u>Identify Entity/Boundary Control Objects:</u>**

Entity Objects:
1. Sales Transaction Details: Stores details of item sold, payment method, additional tax, discount, etc…
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 payment, status
6.Invoice: Item(s) name, price, total amount, tax, discount applied, customer-id,etc…

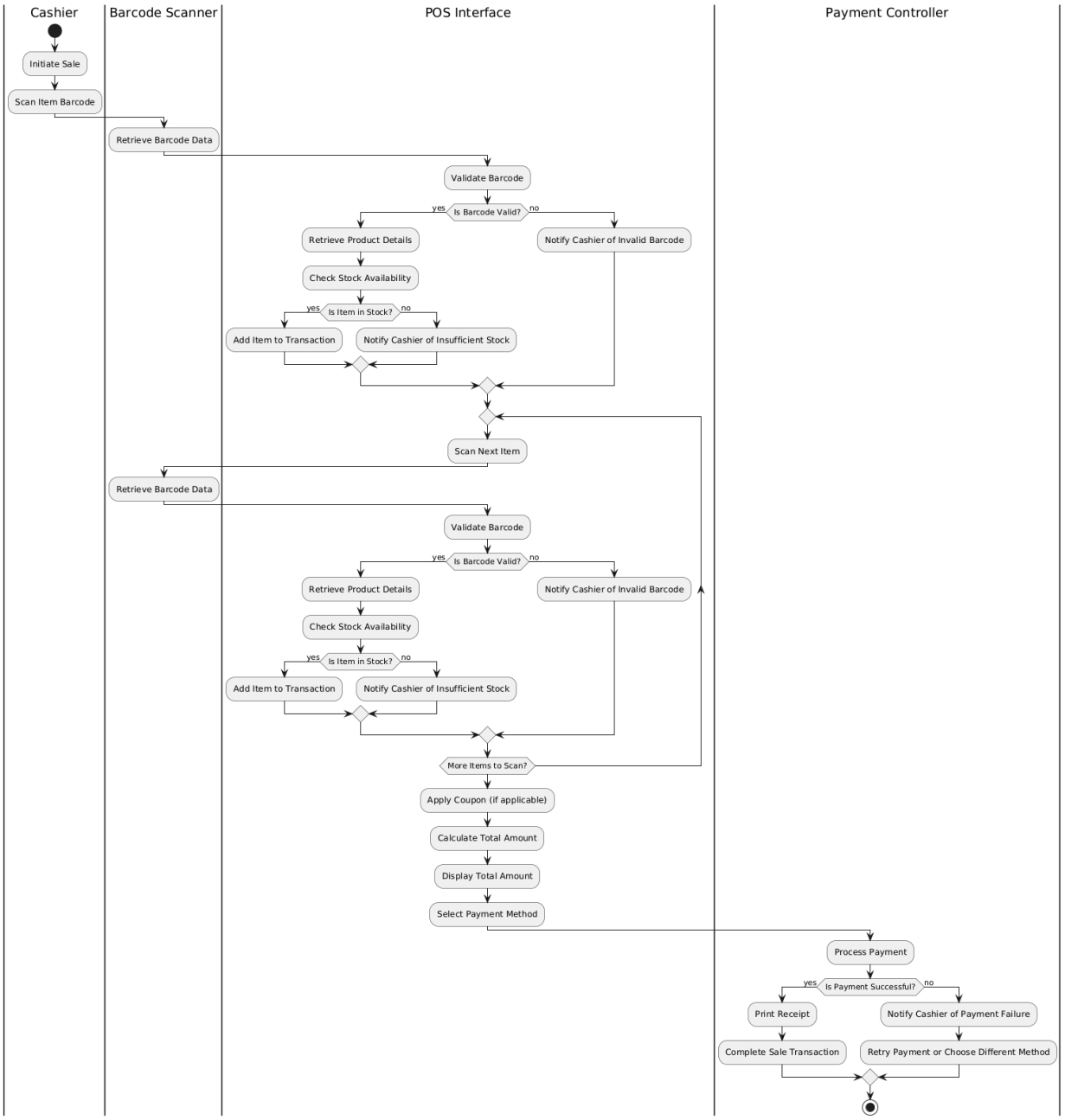
Boundary Objects:
1. POS interface: Screen on which cashier interacts
2. Barcode scanner: Takes the input of products
3.Card Reader: Hardware that reads the card(credit, debit) information
4.Printer: prints the receipt/invoice

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…

➢ **<u>Sequence Diagram:</u>**



➢ Activity Diagram:


Cashier
Barcode Scanner
POS Interface
Payment Controller
Initiate Sale
Scan Item Barcode
Retrieve Barcode Data
Validate Barcode
Is Barcode Valid?
yes
no
Retrieve Product Details
Notify Cashier of Invalid Barcode
Check Stock Availability
Is Item in Stock?
yes
no
Add Item to Transaction
Notify Cashier of Insufficient Stock
Scan Next Item
Retrieve Barcode Data
Validate Barcode
Is Barcode Valid?
yes
no
Retrieve Product Details
Notify Cashier of Invalid Barcode
Check Stock Availability
Is Item in Stock?
yes
no
Add Item to Transaction
Notify Cashier of Insufficient Stock
More Items to Scan?
Apply Coupon (if applicable)
Calculate Total Amount
Display Total Amount
Select Payment Method
Process Payment
Is Payment Successful?
yes
no
Print Receipt
Notify Cashier of Payment Failure
Complete Sale Transaction
Retry Payment or Choose Different Method

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.

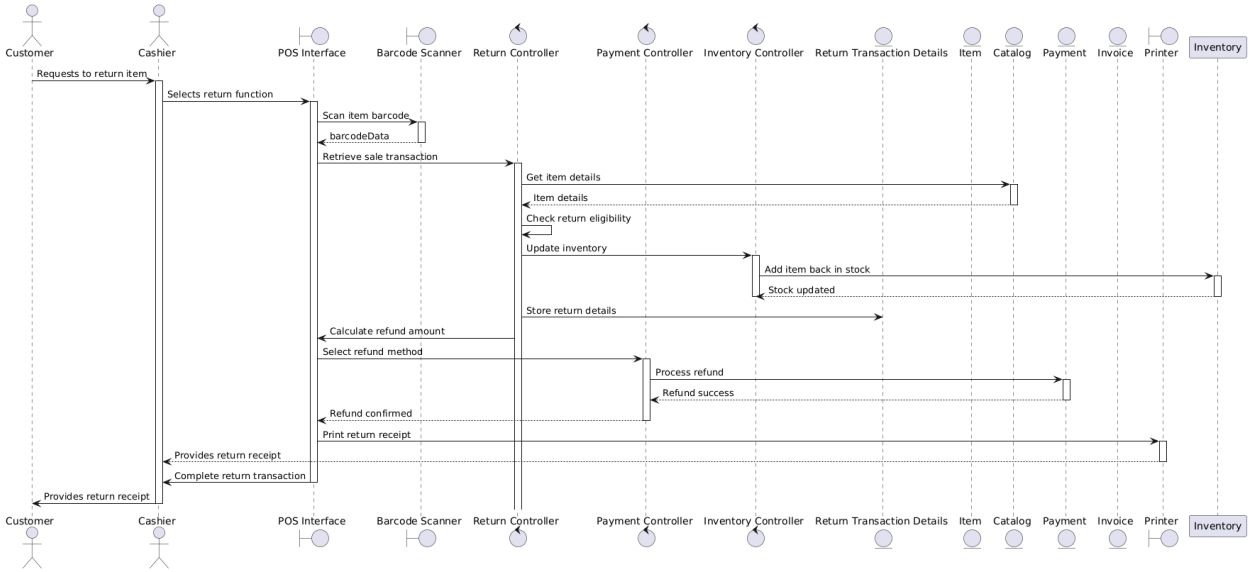
➢ **<u>Identify Entity/Boundary Control Objects:</u>**

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…

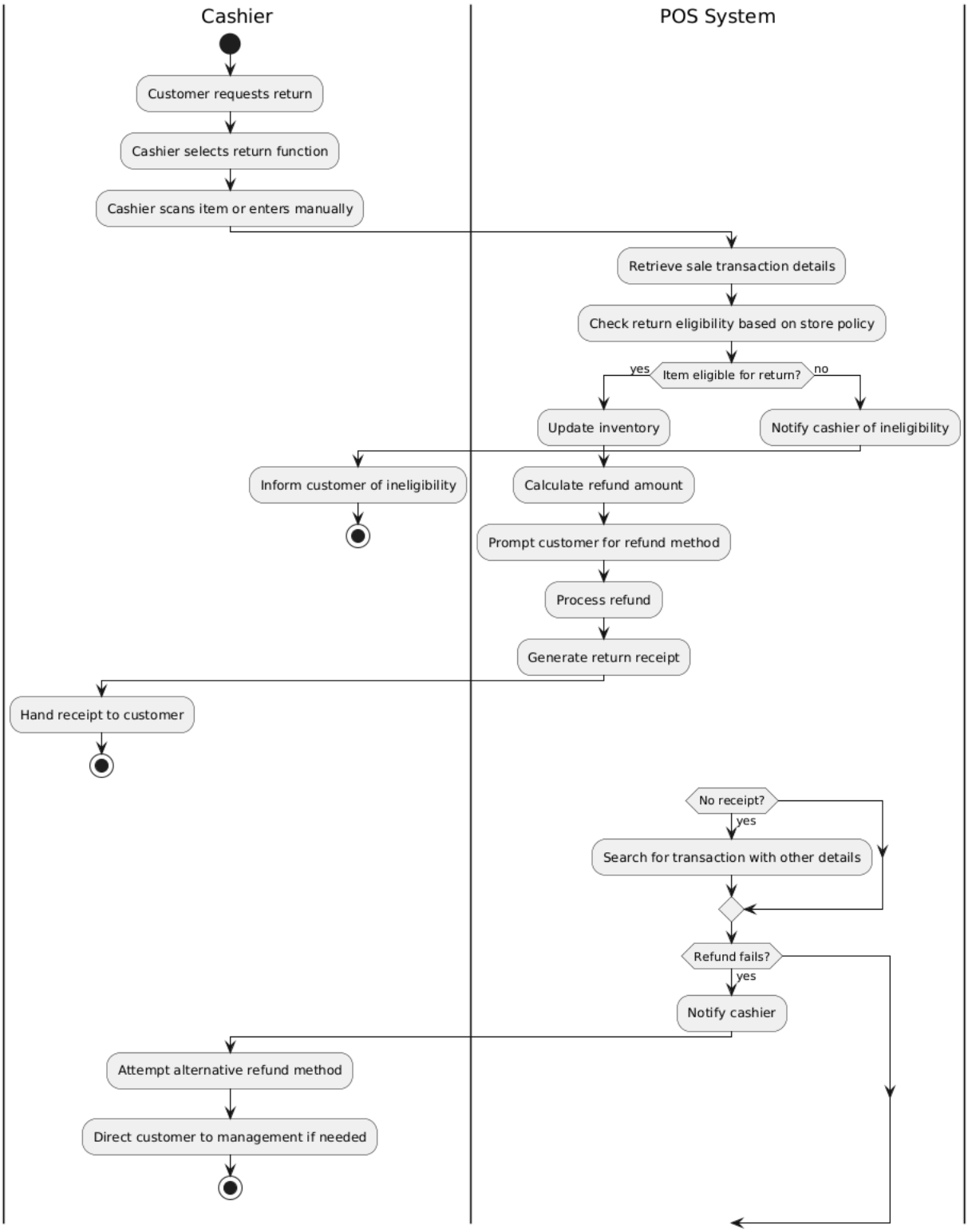
Boundary Objects:
1. POS interface: Screen on which cashier interacts
2. Barcode scanner: Takes the input of products
3.Printer: prints the return receipt

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:


Cashier
POS System
Customer requests return
Cashier selects return function
Cashier scans item or enters manually
Retrieve sale transaction details
Check return eligibility based on store policy
yes
Item eligible for return?
no
Update inventory
Notify cashier of ineligibility
Inform customer of ineligibility
Calculate refund amount
Prompt customer for refund method
Process refund
Generate return receipt
Hand receipt to customer
No receipt?
yes
Search for transaction with other details
Refund fails?
yes
Notify cashier
Attempt alternative refund method
Direct customer to management if needed

➢ Class Diagram: (Combined of both cases)