Software Engineering(IT314)

LAB-6

NAME:Moradiya Rajan Rakeshbhai
Student ID:202201063

Use Case: Process Sale

Goal:
Complete the sale of goods by the customer and handle payment.

Actors:

- Cashier (Primary Actor)
- Customer (Supporting Actor)

Preconditions:

- The cashier is logged into the POS system.
- Goods are available for purchase.

Main Flow:

1. The cashier starts a new sale transaction.
2. The cashier scans the barcodes of the goods.
3. The POS system retrieves the name and price of each item from the catalog system.
4. The POS system communicates with the inventory system to update stock.
5. The cashier enters or selects a payment method (cash, credit card, check).
6. The customer makes the payment.
7. The POS system processes the payment.
8. A receipt is printed, and the sale is finalized.
9. If applicable, the POS system applies gift coupons for a discount.

Postconditions:

- The sale transaction is recorded.
- The inventory system is updated to reflect the sale.
- A receipt is issued to the customer.

Use Case: Handle Return

Goal:
Process a customer's request to return previously purchased goods.

Actors:

- Cashier (Primary Actor)
- Customer (Supporting Actor)

Preconditions:

- The cashier is logged into the POS system.
- The customer has the original receipt or proof of purchase.

Main Flow:

1. The cashier starts a return transaction.
2. The customer provides the original receipt.
3. The cashier scans the items being returned.
4. The POS system retrieves the details of the original sale.
5. The POS system interacts with the inventory system to update the stock.
6. The cashier processes the return by refunding the payment (in cash, credit card, or check).
7. The POS system finalizes the return transaction and prints a receipt for the return.

Postconditions:

- The return transaction is recorded.
- The inventory is updated to reflect the returned goods.
- The customer receives a refund or store credit.

Entity/Boundary/Control Objects

Entity Objects (data-centric objects):

- **Product**: Represents the goods being sold or returned (name, price, barcode, etc.).
- **Inventory**: Tracks the quantity of products in stock.
- **Sale Transaction**: Represents a sale and stores details like items, total amount, payment method.
- **Return Transaction**: Represents a return and stores details like items returned, refund, etc.
- **User**: Represents a logged-in employee (cashier or administrator).

Boundary Objects (interface between the system and users):

- **POS Interface**: The user interface where the cashier interacts with the system to process sales and returns.
- **Barcode Scanner**: Device for scanning product barcodes.
- **Receipt Printer**: Device for printing sale or return receipts.

Control Objects (handle the flow of operations):

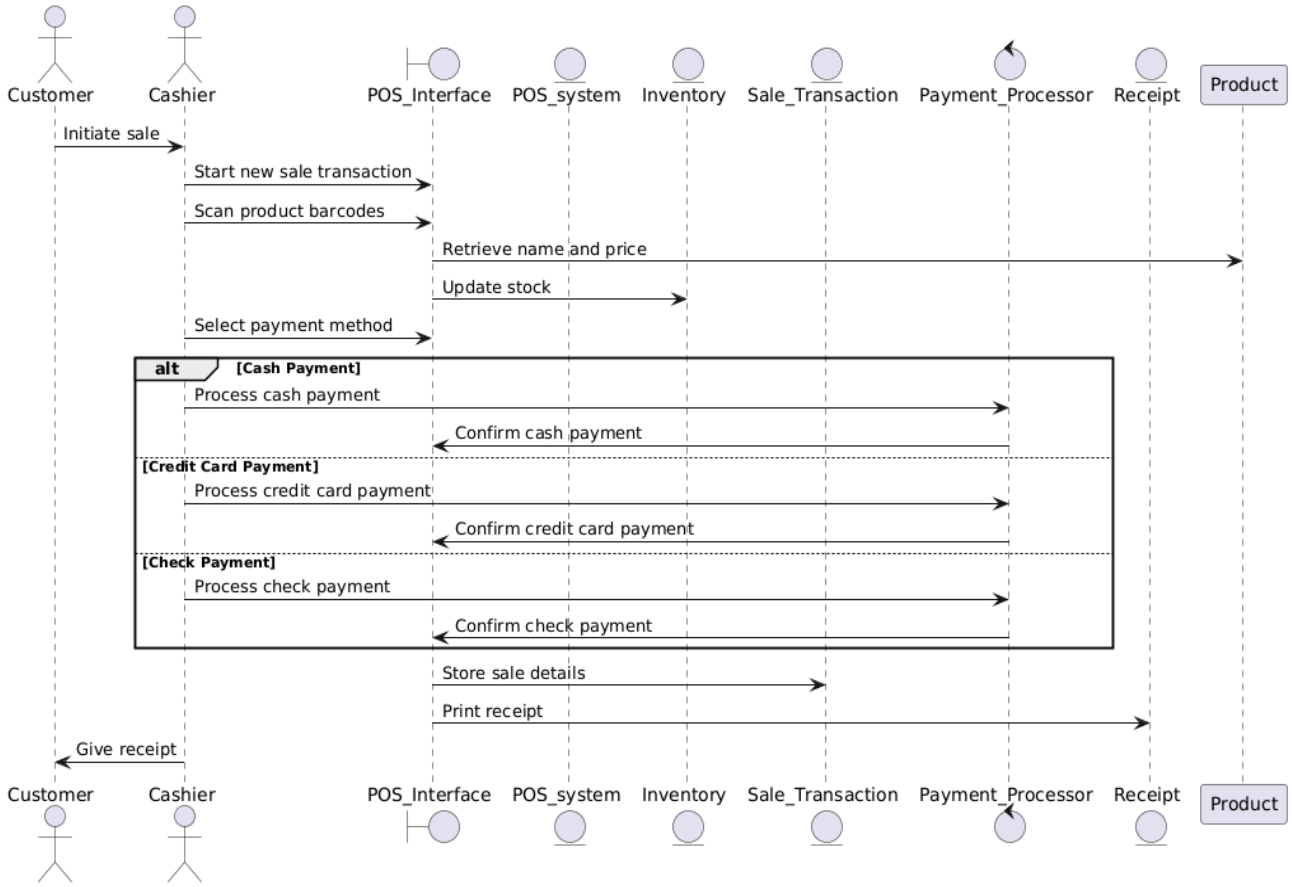
- **Sale Processor**: Manages the steps involved in processing a sale (retrieving item details, calculating total, applying coupons).
- **Return Processor**: Manages the steps involved in processing a return (validating receipt, refunding payment).

- **Inventory Manager**: Interacts with the inventory to update stock based on sales or returns.
- **Payment Processor**: Handles different payment methods (cash, credit, check).

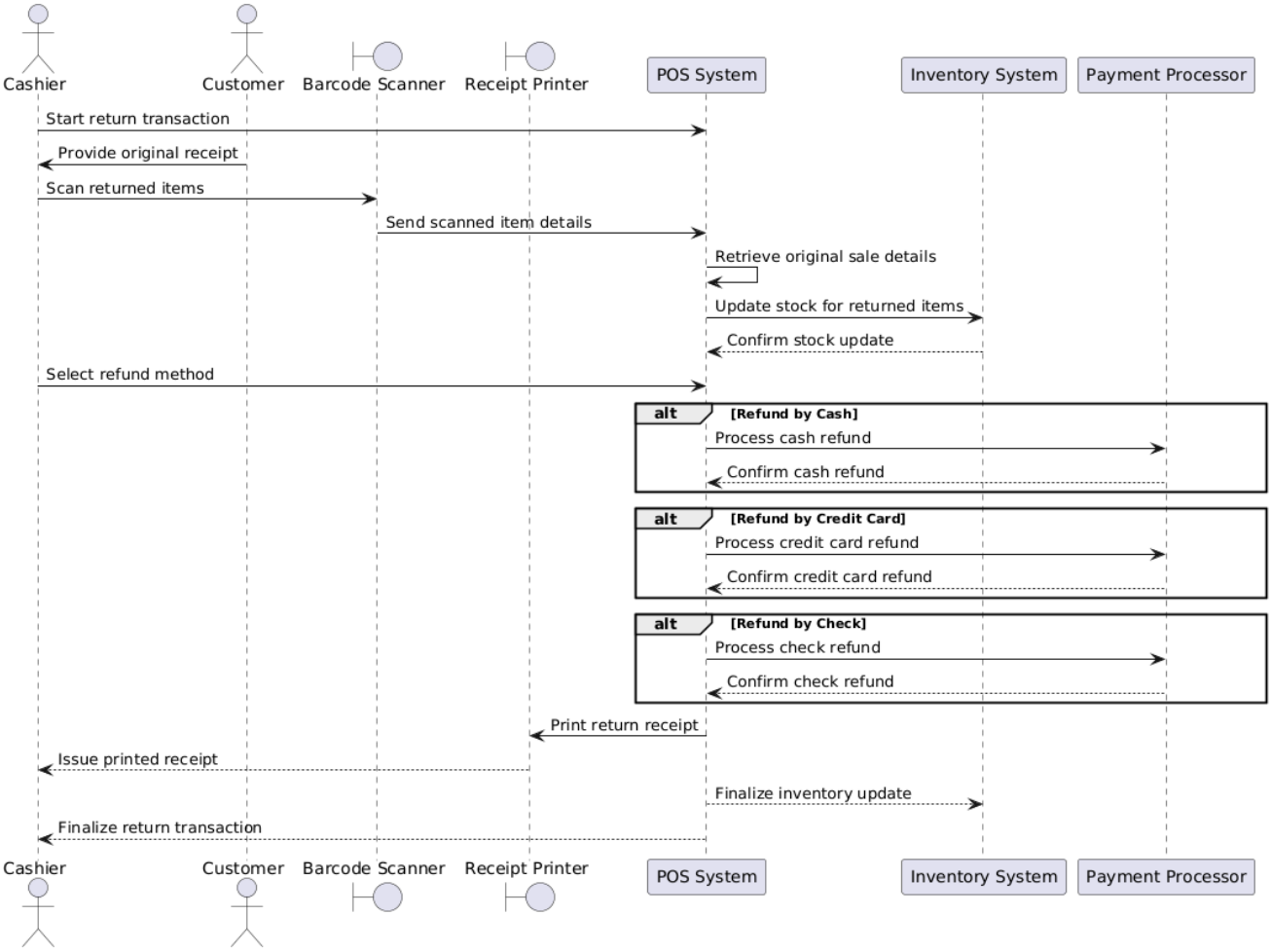
These objects work together to manage and process the tasks of sales and returns in a retail environment.

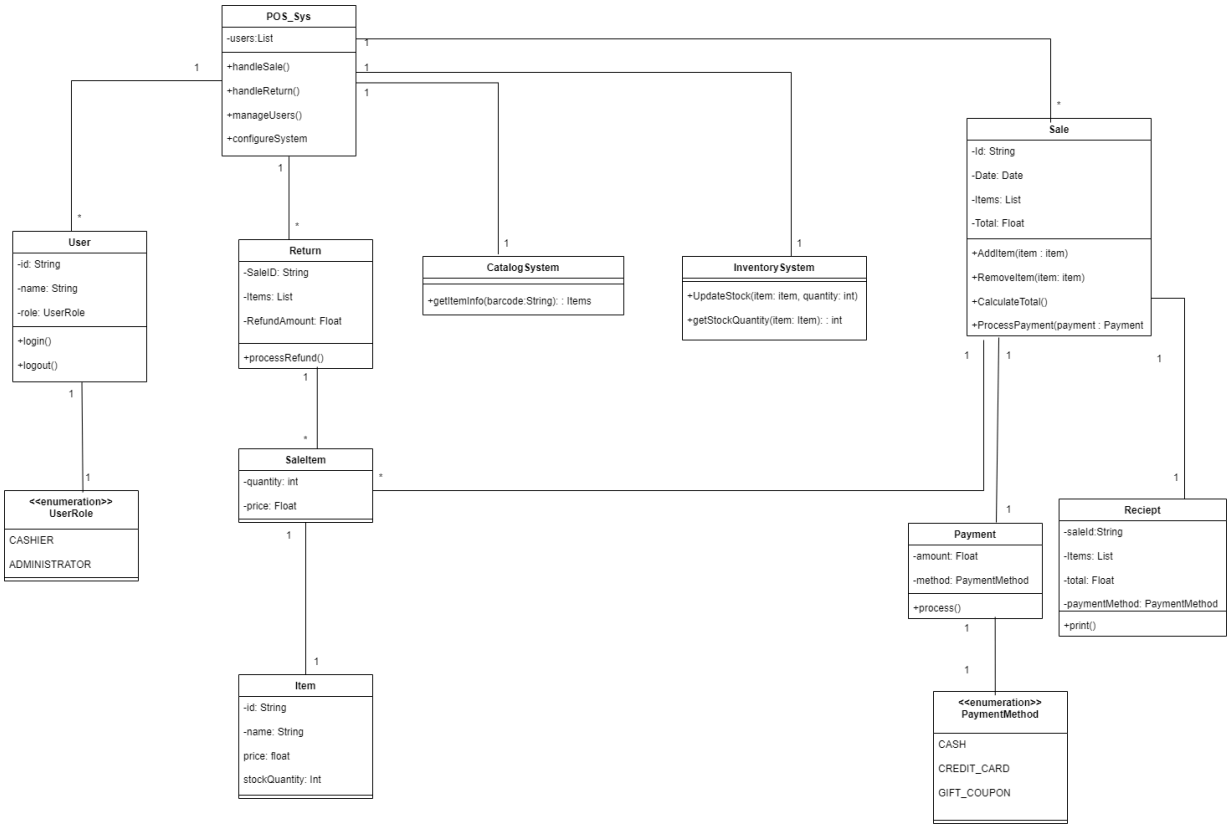
Sequence Diagrams

Process Sale



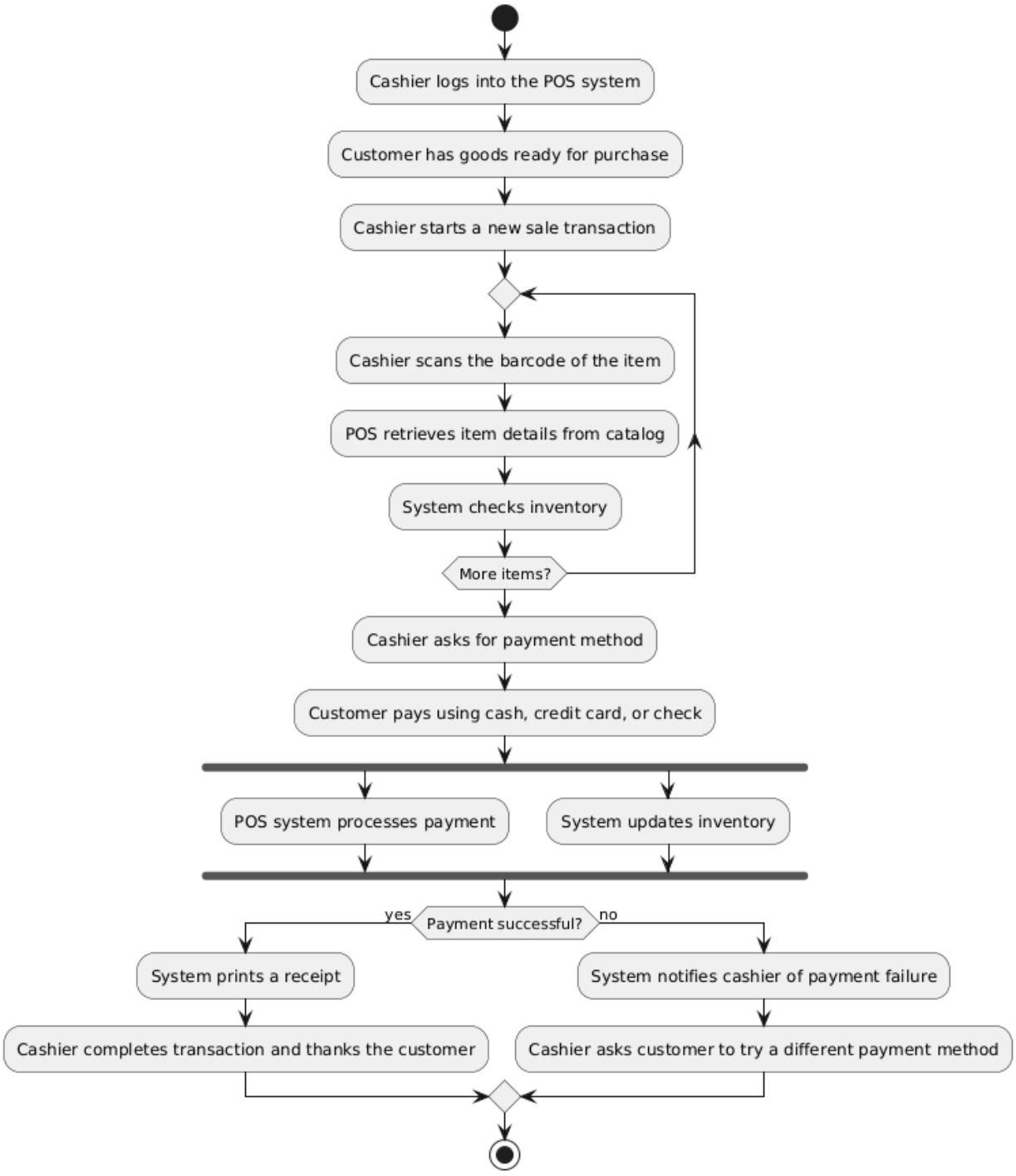
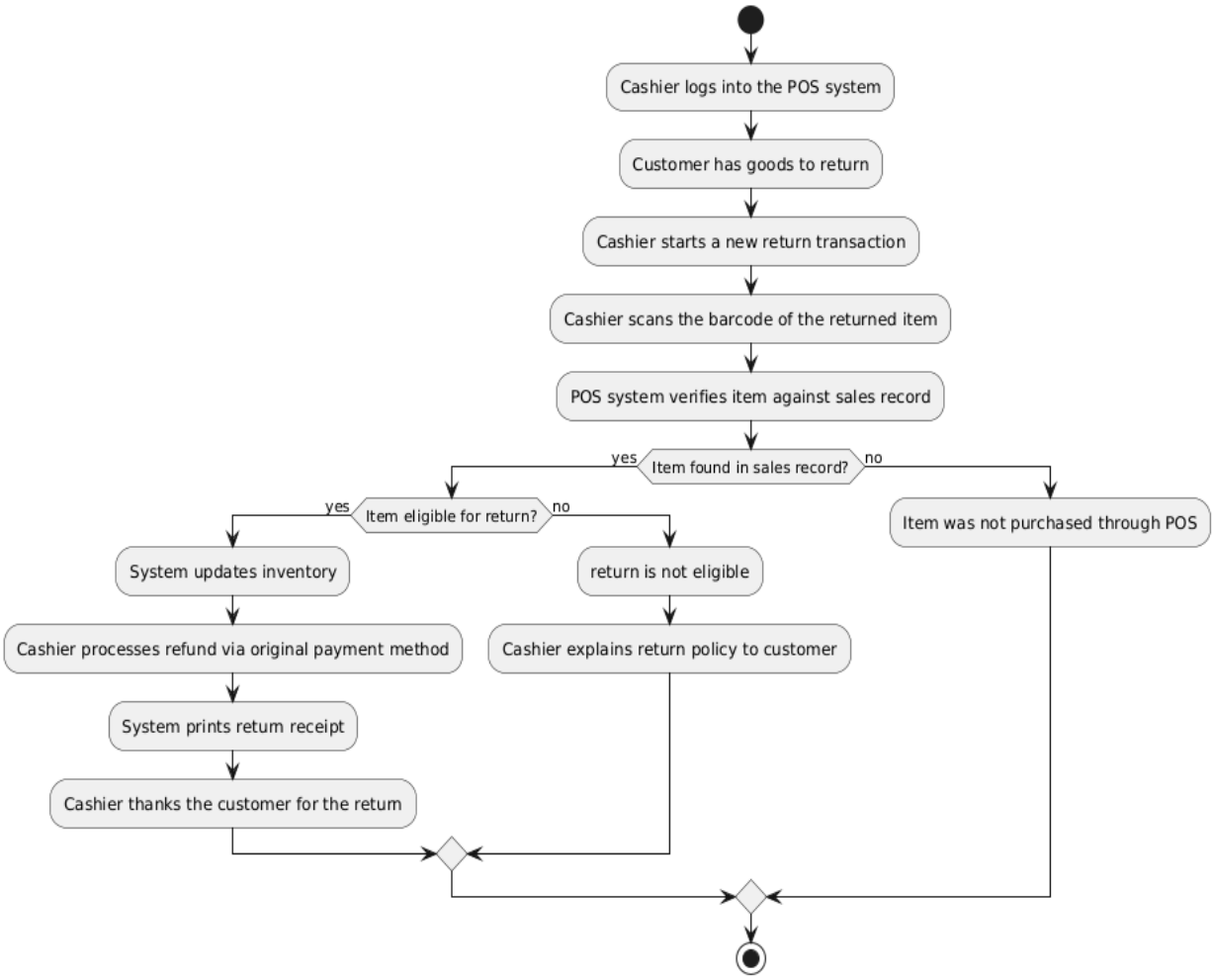
Handle Return




POS_Sys
-users:List
+handleSale()
+handleReturn()
+manageUsers()
+configureSystem
User
-id: String
-name: String
-role: UserRole
+login()
+logout()
<<enumeration>>
UserRole
CASHIER
ADMINISTRATOR
Return
-SaleID: String
-Items: List
-RefundAmount: Float
+processRefund()
Catalog System
+getItemInfo(barcode:String): : Items
Inventory System
+UpdateStock(item: item, quantity: int)
+getStockQuantity(item: Item): : int
Sale
-Id: String
-Date: Date
-Items: List
-Total: Float
+AddItem(item : item)
+RemoveItem(item: item)
+CalculateTotal()
+ProcessPayment(payment : Payment
SaleItem
-quantity: int
-price: Float
Item
-id: String
-name: String
price: float
stockQuantity: Int
Payment
-amount: Float
-method: PaymentMethod
+process()
<<enumeration>>
PaymentMethod
CASH
CREDIT_CARD
GIFT_COUPON
Reciept
-saleId:String
-Items: List
-total: Float
-paymentMethod: PaymentMethod
+print()

Activity diagram

Process Sale



Handle Return