Software Engineering IT314 - Lab 6

202201106 Vraj Dobariya

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

Process Sale

Primary Actor: Cashier

Goal: To complete a customer's purchase.

Precondition:

- The customer has chosen items to buy.
- The cashier is logged into the POS system.

Main Flow:

1. The cashier begins the sale by scanning the barcode of the first item.
2. The POS system connects to the catalog to get details about the item (such as the price and description).
3. The system adds the item to the ongoing sale.
4. The cashier repeats this process for all items the customer is purchasing.

5. Once all items are scanned, the cashier selects the customer’s preferred payment method (cash, credit card, etc.).
6. The POS system calculates the total amount and adjusts the stock levels in the inventory system.
7. The payment is processed.
8. A receipt is printed and given to the customer.

Postcondition:

- The sale is complete.
- The inventory is updated to reflect the items sold.

Alternative Flow:

1.1) Scan again if the system fails to scan the item for the first time.

2.1) Scan again if the system fails to fetch item details from the catalog system.

7.1) If the payment fails, the system asks the cashier to select another payment method or cancels the sale.

8.1) If the printer fails, then the system alerts the cashier that the printer is not working.

8.2) If the printer fails, then the system asks if the cashier can either retry printing or offer the customer an emailed or digital receipt.

Handle Return

Primary Actor: Cashier

Goal: To process a customer's return of purchased items.

Precondition:

- The customer has a valid receipt for the items.
- The cashier is logged into the POS system.

Main Flow:

1. The cashier selects the "Return" option in the POS system.
2. The system asks for the receipt information or the original transaction number.
3. The cashier scans the items the customer wants to return.
4. The POS system checks with the inventory system and prepares to restock the returned items.
5. The system calculates how much money needs to be refunded to the customer.
6. The cashier completes the return by either giving cash or processing a refund to the customer's card.
7. The system updates the inventory and prints a return receipt for the customer.

Postcondition:

- The return is successfully completed.
- The inventory is updated with the returned items.

Alternative Flow:

1.1) If the "Return" option is unavailable due to system maintenance or updates, the cashier can manually record the return for later processing.

3.1) Scan again if the system fails to scan the item for the first time.

7.1) If the printer fails, then the system alerts the cashier that the printer is not working.

7.2) If the printer fails, then the system asks if the cashier can either retry printing or offer the customer an emailed or digital receipt.

- **Identify Entity/Boundary Control Objects**

Entity Objects

customer	item	receipt
payment	sale	Cashier

Boundary objects

POS UI	UI for scanning and entering details
Inventory system	Updates item stock
Catalog system	Provides item details
Payment system	Handles cash/card payments

Control objects

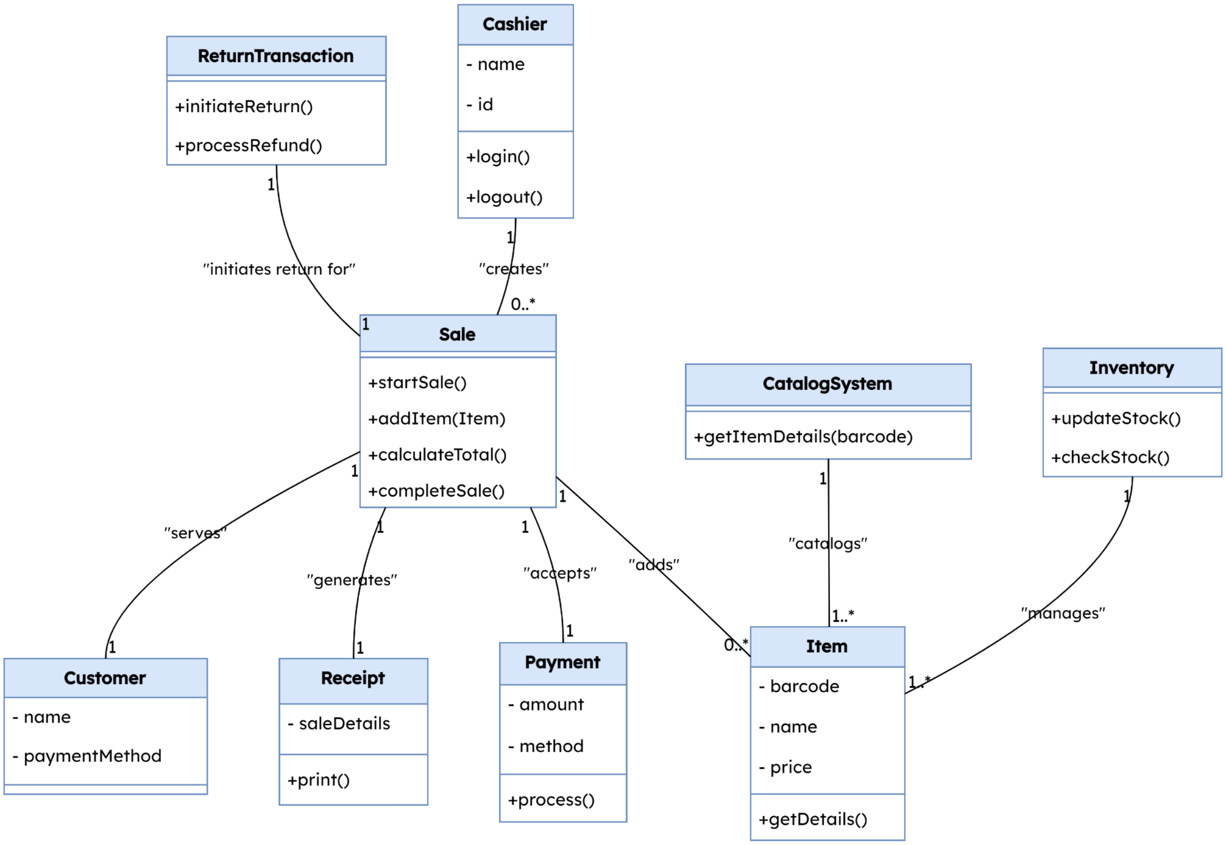
SaleController,
InventoryController
PaymentController
ReturnController
ReceiptController

- **Develop Sequence Diagrams**

Process sale: Attached at last

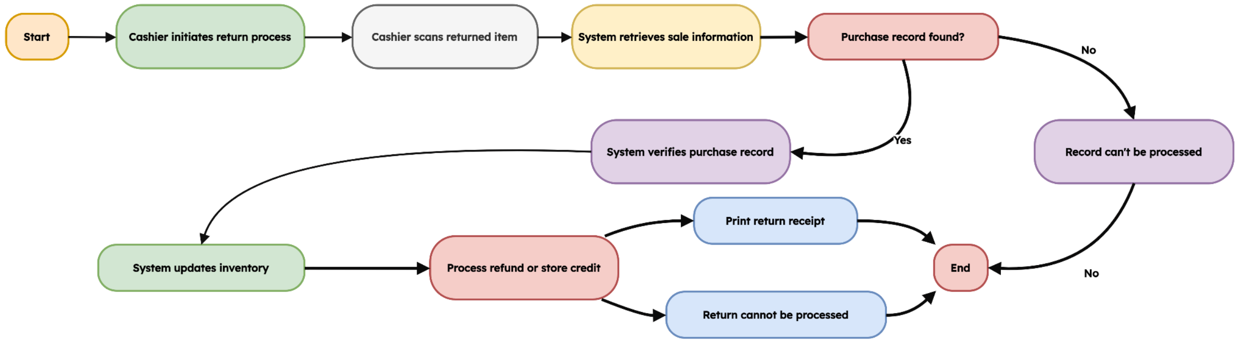
Handle return: Attached at last

- **Develop Analysis Domain Models**

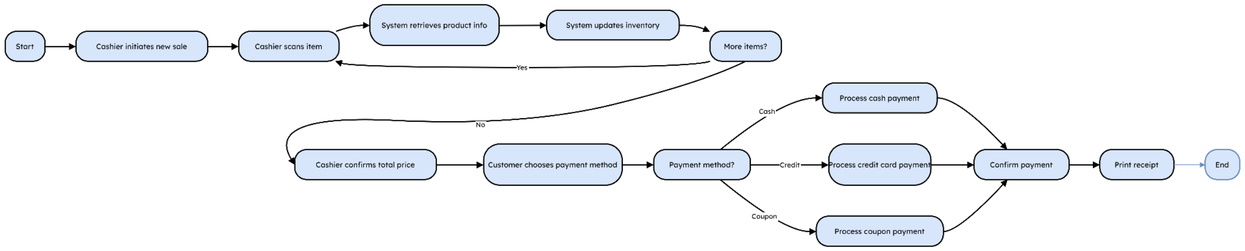


- **Develop activity diagrams for "Process Sale" and "Handle Return" use cases.**

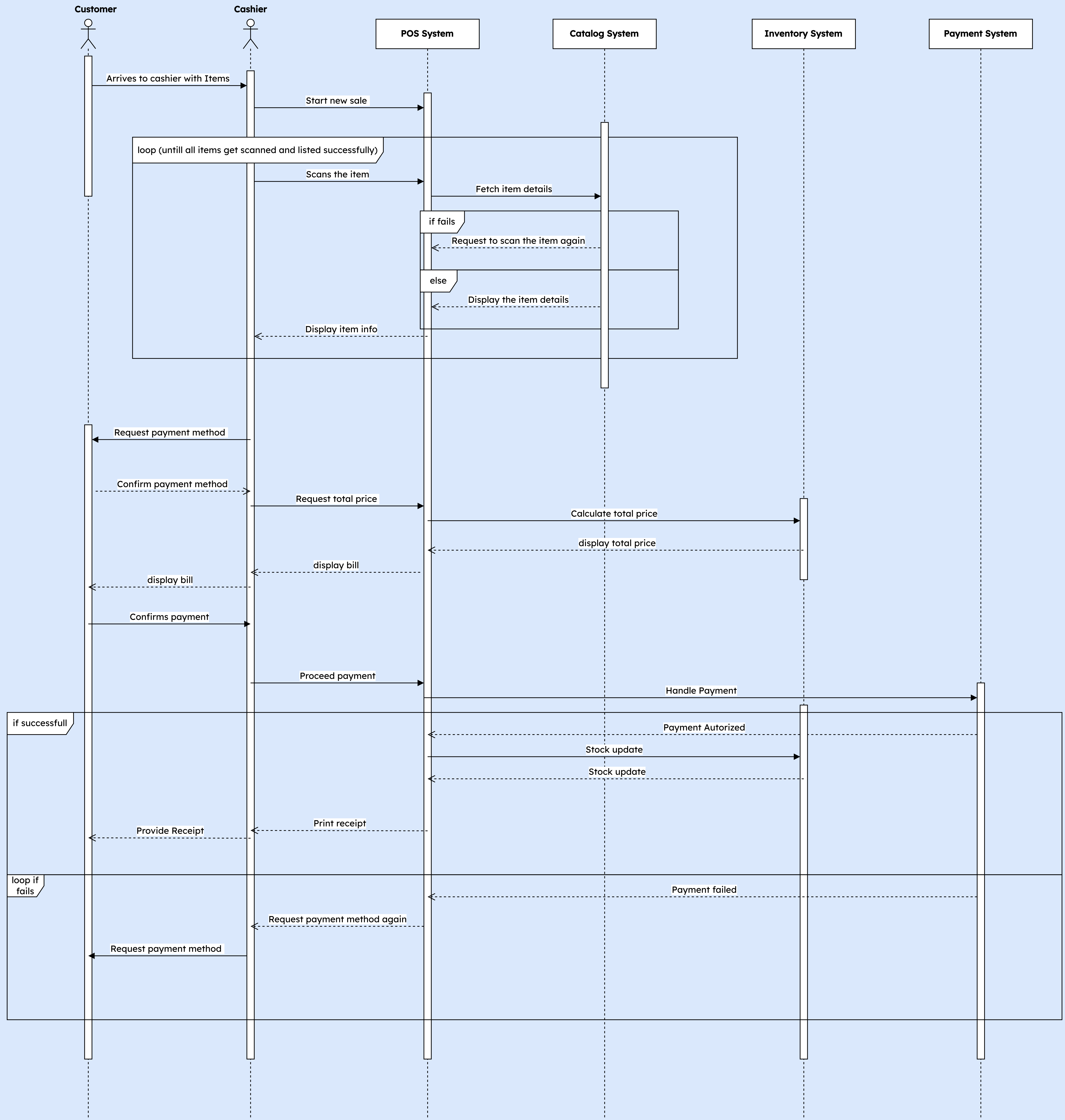
Use case: Handle return


Start
Cashier initiates return process
Cashier scans returned item
System retrieves sale information
Purchase record found?
No
Yes
System verifies purchase record
Record can't be processed
Print return receipt
System updates inventory
Process refund or store credit
End
No
Return cannot be processed

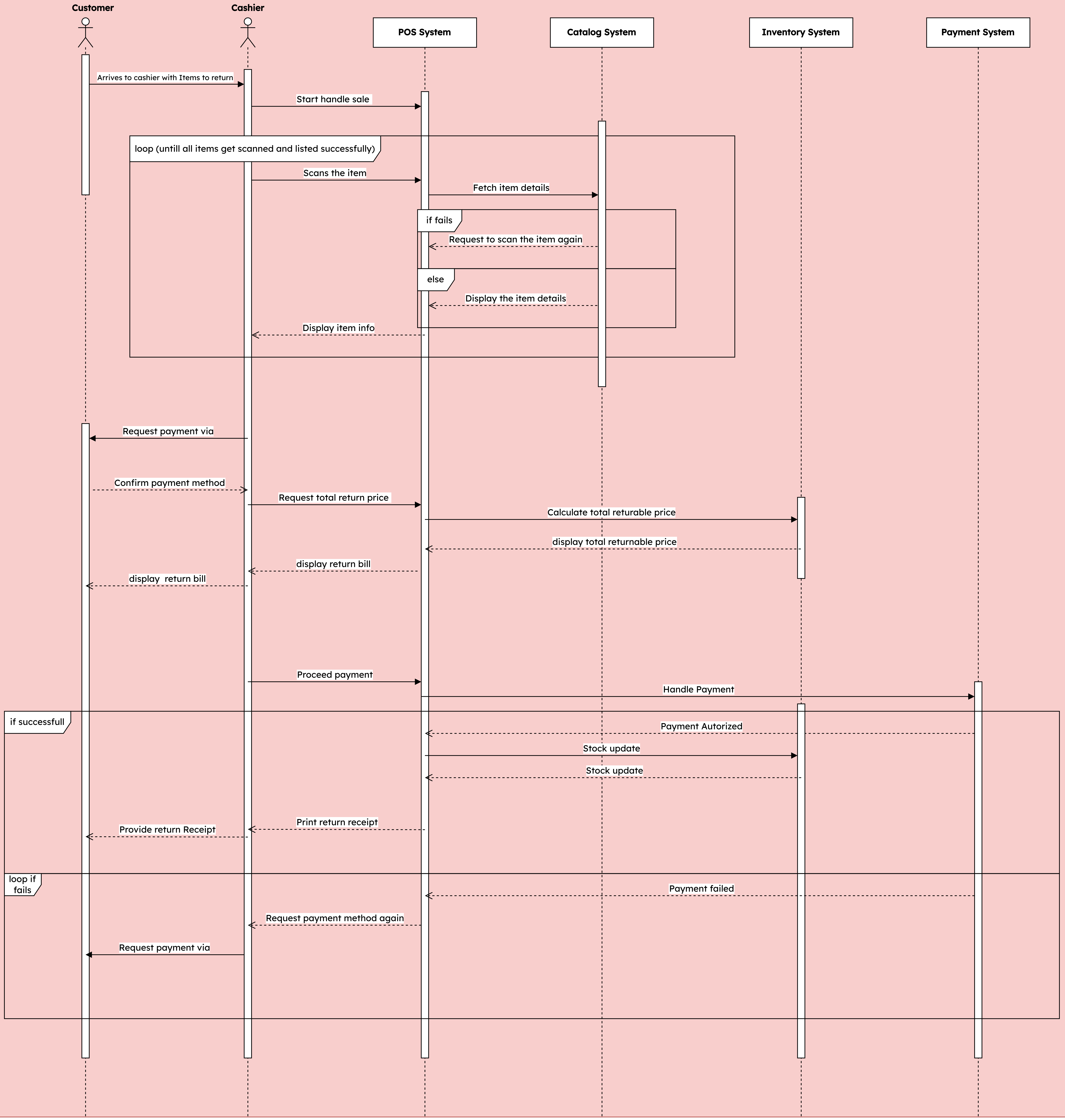
Use case: Process Sale



Process Sale



Handle Return


Customer
Cashier
POS System
Catalog System
Inventory System
Payment System
Arrives to cashier with Items to return
Start handle sale
loop (untill all items get scanned and listed successfully)
Scans the item
Fetch item details
if fails
Request to scan the item again
else
Display the item details
Display item info
Request payment via
Confirm payment method
Request total return price
Calculate total returable price
display total returnable price
display return bill
display return bill
Proceed payment
Handle Payment
if successfull
Payment Autorized
Stock update
Stock update
Print return receipt
Provide return Receipt
loop if fails
Payment failed
Request payment method again
Request payment via