IT314 - SE
LAB - 06
202201265 - Prachi Chaudhari

1. Use Case: Process Sale

Actor:
- Cashier

Preconditions:
- The cashier is logged into the POS system
- The customer has items to purchase

Main Flow:
1. The cashier starts a new sale transaction
2. The cashier scans the barcode of each item
3. For each scanned item:
   a. The system retrieves the item name and price from the catalog system
   b. The system updates the inventory by deducting the stock amount
   c. The system displays the item details and updates the total
4. If the customer has gift coupons, the cashier applies them to the transaction
5. The cashier informs the customer of the final total amount
6. The customer chooses a payment method (cash, credit card, or check)
7. The cashier processes the payment
8. The system verifies the payment
9. The system generates and prints a receipt
10. The cashier gives the receipt and purchased items to the customer

Alternate Flows:
- 2a. If an item's barcode can't be scanned:
  1. The cashier manually enters the item code
  2. The flow continues from step 3
- 6a. If the customer wants to cancel the transaction:
  1. The cashier cancels the transaction in the system
  2. The system reverses any inventory changes
- 8a. If the payment is declined:
  1. The system notifies the cashier
  2. The cashier asks for an alternative payment method
  3. The flow returns to step 6

Postconditions:

- The sale is recorded in the system
- The inventory is updated
- The payment is processed
- A receipt is printed

Use Case: Handle Return

Actor:
- Cashier

Preconditions:
- The cashier is logged into the POS system
- The customer has item(s) to return
- The customer has the original receipt or the transaction can be found in the system

Main Flow:
1. The cashier initiates a return transaction
2. The cashier scans or enters the receipt number
3. The system displays the original transaction details
4. The cashier scans the barcode of the item(s) being returned
5. For each returned item:
   a. The system verifies the item against the original transaction
   b. The system calculates the refund amount
   c. The system updates the inventory by increasing the stock amount
6. The cashier confirms the return details with the customer
7. The system processes the refund
8. The system generates and prints a return receipt
9. The cashier gives the return receipt to the customer

Alternate Flows:
- 2a. If the customer doesn't have the receipt:
  1. The cashier searches for the transaction using other details (date, credit card, etc.)
  2. If found, the flow continues from step 3
  3. If not found, the return cannot be processed
- 5a. If an item is not found in the original transaction:
  1. The system notifies the cashier
  2. The cashier informs the customer that the item cannot be returned
- 6a. If the customer decides not to proceed with the return:
  1. The cashier cancels the return transaction
  2. The system reverses any inventory changes

Postconditions:
- The return is recorded in the system

- The inventory is updated
- The refund is processed
- A return receipt is printed

2.Entity/Boundary/Control Objects

Let's identify the Entity, Boundary, and Control objects based on the problem description:

Entity Objects:

- Sale
- Item
- Payment
- Receipt
- Return
- Customer
- Inventory
- User (Employee)
- GiftCoupon
- Catalog

Boundary Objects:

- POSInterface
- BarCodeScanner
- ReceiptPrinter
- PaymentTerminal

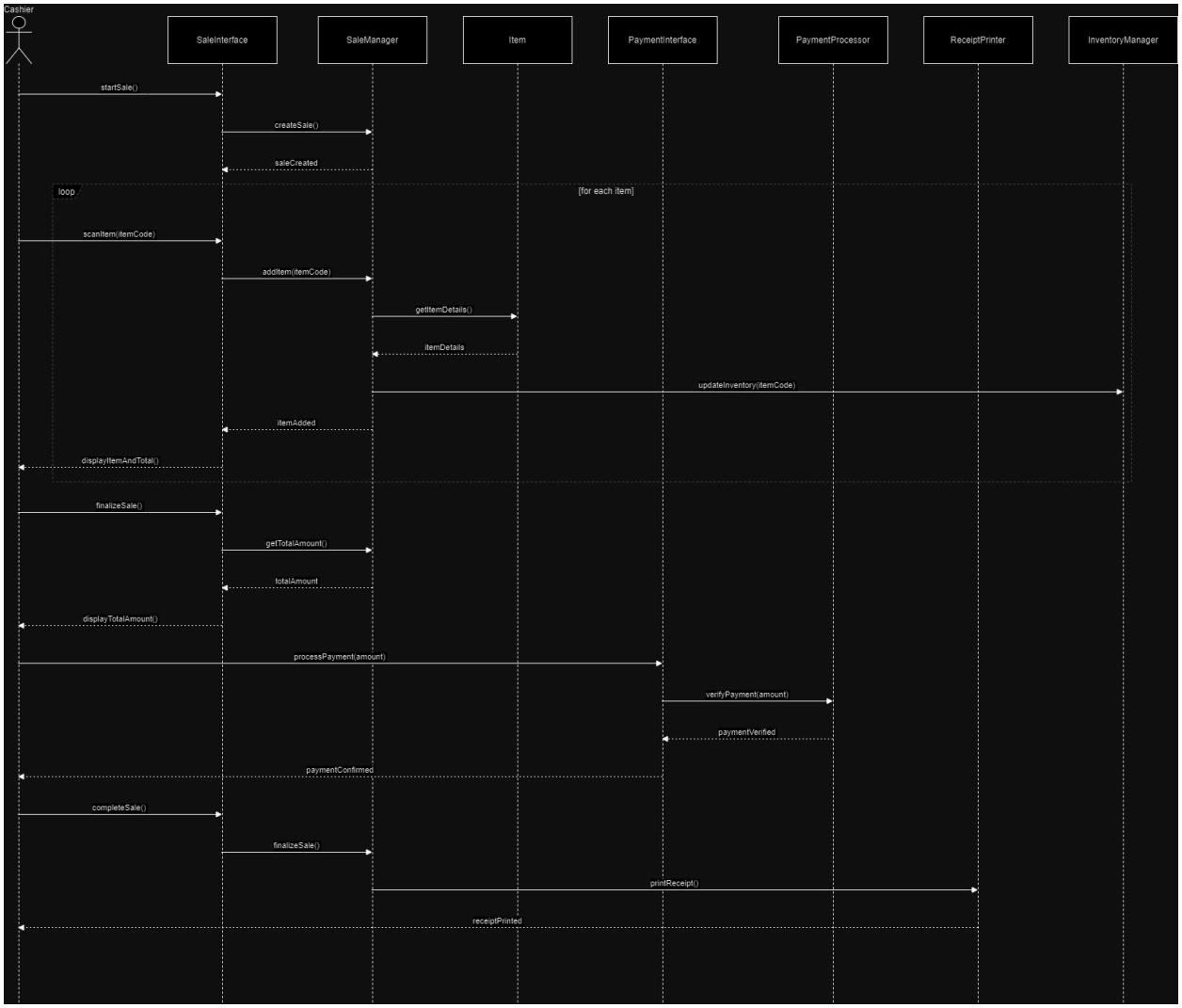
Control Objects:

- SaleManager
- ReturnManager
- PaymentProcessor
- InventoryManager

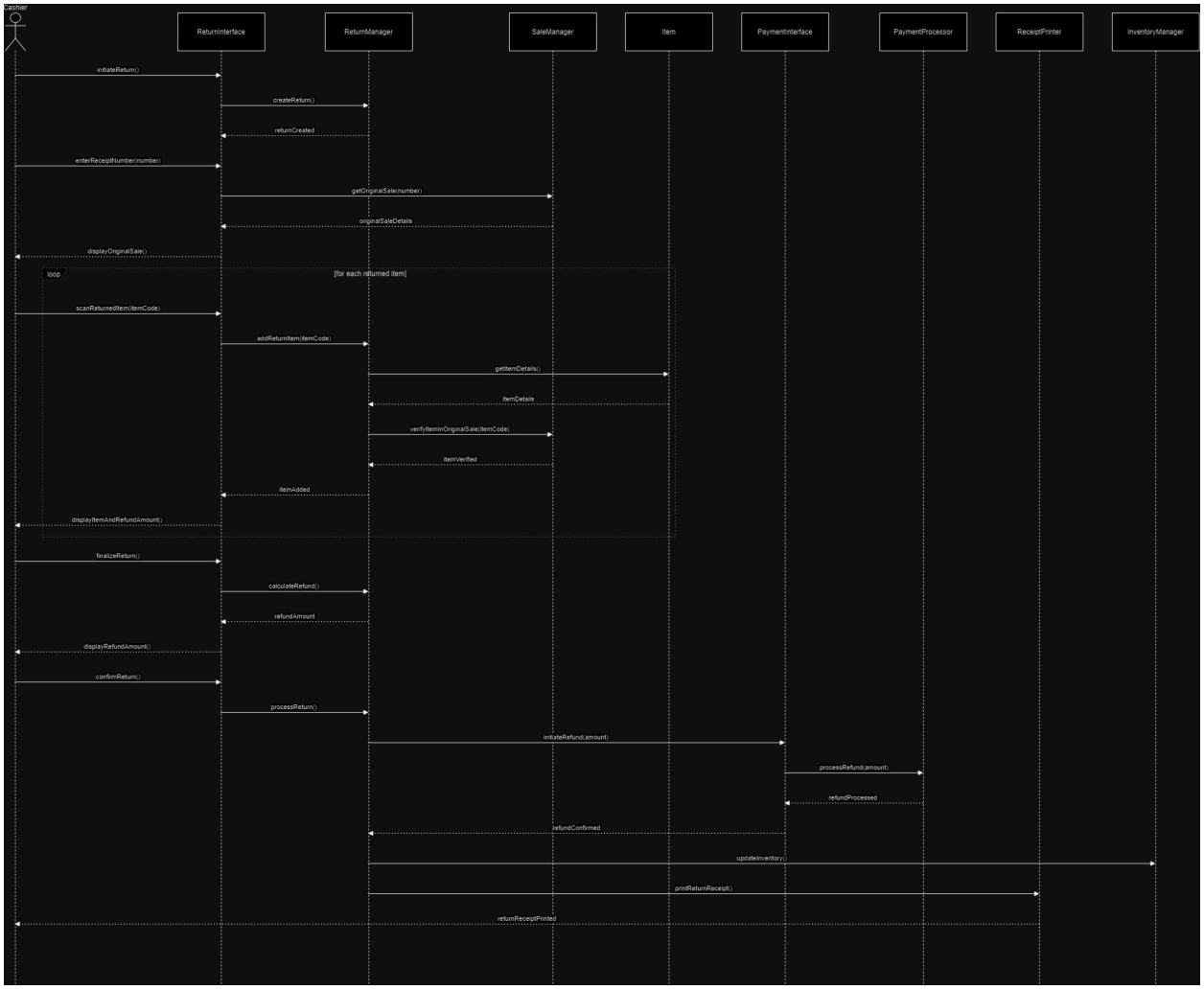
- CatalogManager
- UserManager
- SecurityManager

3. Sequence Diagram:

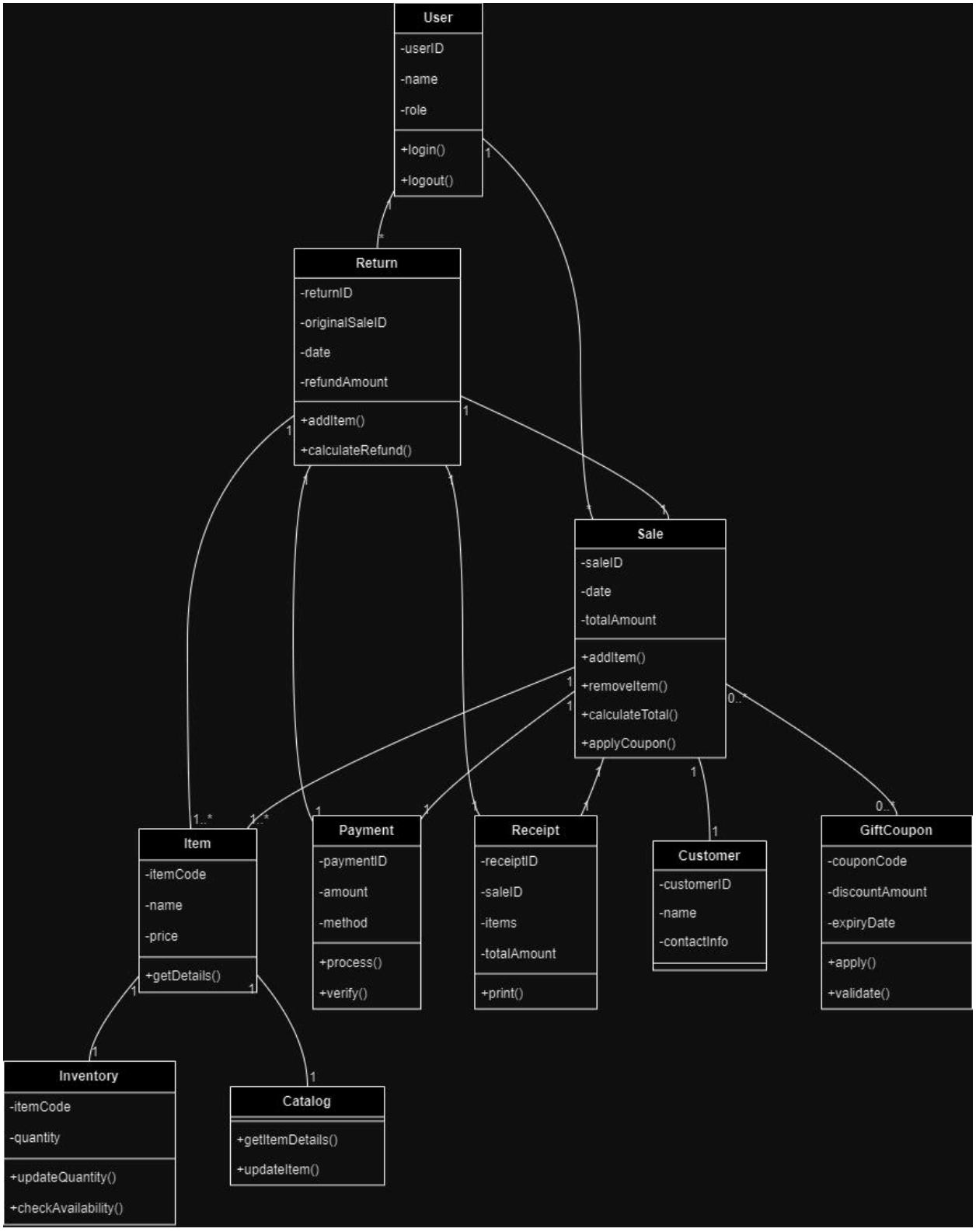
Process Sale:



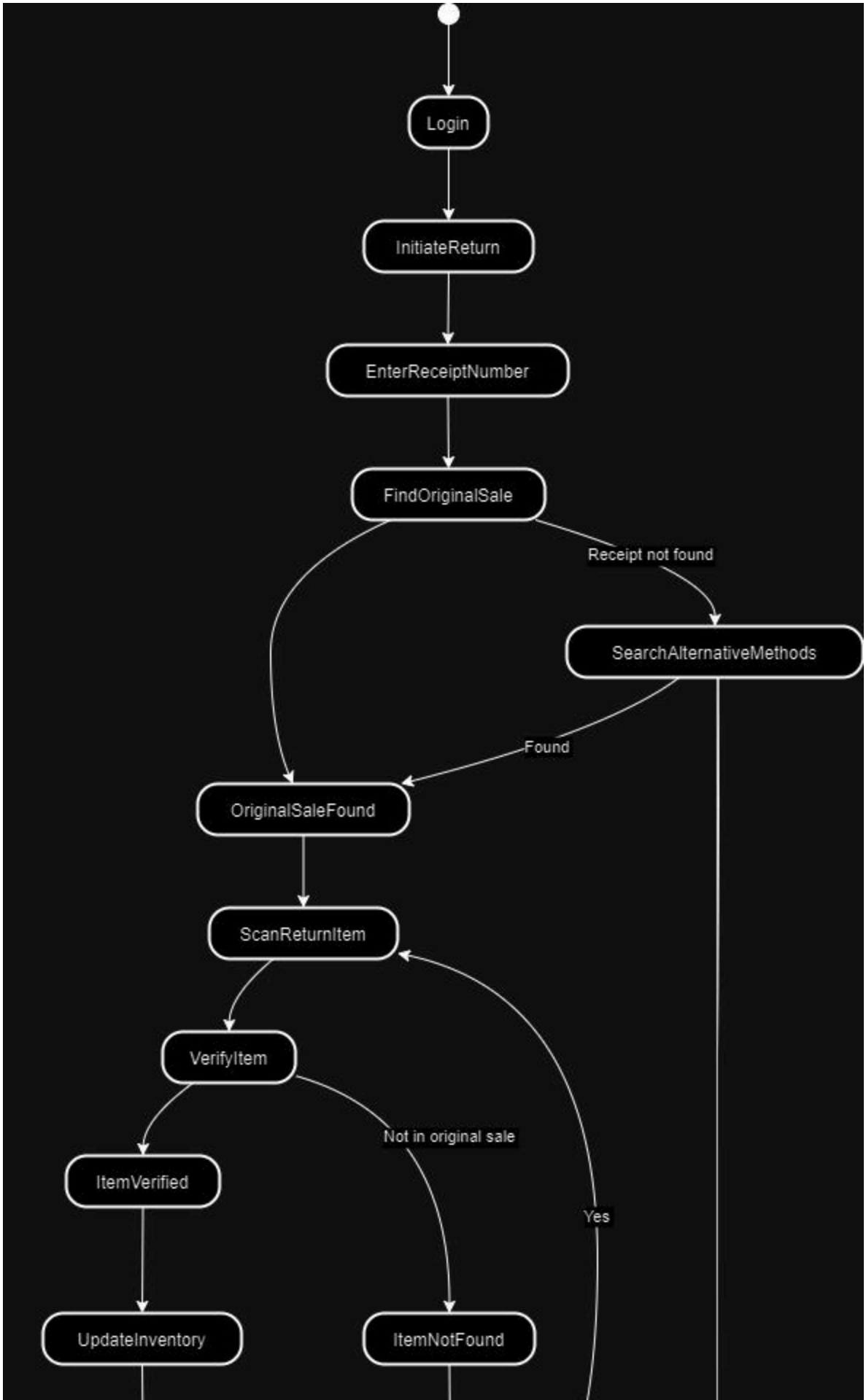
Handle Return:


Cashier
ReturnInterface
ReturnManager
SaleManager
Item
PaymentInterface
PaymentProcessor
ReceiptPrinter
InventoryManager
initiateReturn()
createReturn()
returnCreated
enterReceiptNumber(number)
getOriginalSale(number)
originalSaleDetails
displayOriginalSale()
loop
[for each returned item]
scanReturnedItem(itemCode)
addReturnItem(itemCode)
getItemDetails()
itemDetails
verifyItemInOriginalSale(itemCode)
itemVerified
itemAdded
displayItemAndRefundAmount()
finalizeReturn()
calculateRefund()
refundAmount
displayRefundAmount()
confirmReturn()
processReturn()
initiateRefund(amount)
processRefund(amount)
refundProcessed
refundConfirmed
updateInventory()
printReturnReceipt()
returnReceiptPrinted

Q4:


User
-userID
-name
-role
+login()
+logout()
Return
-returnID
-originalSaleID
-date
-refundAmount
+addItem()
+calculateRefund()
Sale
-saleID
-date
-totalAmount
+addItem()
+removeItem()
+calculateTotal()
+applyCoupon()
Item
-itemCode
-name
-price
+getDetails()
Payment
-paymentID
-amount
-method
+process()
+verify()
Receipt
-receiptID
-saleID
-items
-totalAmount
+print()
Customer
-customerID
-name
-contactInfo
GiftCoupon
-couponCode
-discountAmount
-expiryDate
+apply()
+validate()
Inventory
-itemCode
-quantity
+updateQuantity()
+checkAvailability()
Catalog
+getItemDetails()
+updateItem()

Q5:


Login
InitiateReturn
EnterReceiptNumber
FindOriginalSale
Receipt not found
SearchAlternativeMethods
Found
OriginalSaleFound
ScanReturnItem
VerifyItem
Not in original sale
ItemVerified
Yes
UpdateInventory
ItemNotFound