# IT - 314
# SOFTWARE ENGINEERING

## Lab Session:
## Modeling Class Diagram and Activity Diagram (Point of Sale System):

NAME: PATEL MIHIRKUMAR SURESHBHAI
STUDENT ID: 202201506

# QUE : Develop Use Case Textual Description for "Process Sale" and "Handle Return" use cases.

# ANS:

## Use Case: Process Sale

**Use Case ID:** UC-001

**Use Case Name:** Process Sale

**Actor:** Cashier

**Trigger :** Cashier Initiates Purchase

**Preconditions:**

- The cashier is logged into the POS system.
- The customer is ready to make a purchase.

**Postconditions:**

- A sale transaction is completed.
- Inventory is updated.
- A receipt is printed and provided to the customer.

**Main Flow:**

1. The cashier starts a new sale transaction.
2. The cashier scans the barcode of the first item.
3. The POS system retrieves the name and price of the item from the catalog.
4. The system updates the inventory to reflect the deduction of the item's stock.
5. The cashier continues to scan items until the customer indicates they are finished.
6. The system displays a summary of the items purchased along with the total amount due.
7. The customer chooses a payment method (cash, credit card, check).

8. The cashier processes the payment through the appropriate method.
9. The system confirms the payment is successful.
10. The system generates a receipt.
11. The cashier provides the receipt to the customer and completes the transaction.

**Alternative Flows:**

- **A1:** Item Not Found
  If the barcode is not found in the catalog, the system alerts the cashier and prompts for manual entry or verification.
- **A2:** Payment Declined
  If the payment is declined (for credit card or check), the system notifies the cashier and prompts for an alternative payment method.

## Use Case: Handle Return

**Use Case ID:** UC-002

**Use Case Name:** Handle Return

**Actor:** Cashier

**Trigger :** Cashier Requests Return:

**Preconditions:**

- The cashier is logged into the POS system.
- The customer has items to return along with a receipt.

**Postconditions:**

- The return transaction is completed.
- Inventory is updated to reflect the returned items.
- A receipt for the return is printed.

**Main Flow:**

1. The cashier starts a new return transaction.

2. The cashier requests the receipt from the customer.
3. The cashier scans the barcode of the returned item(s) from the receipt.
4. The system verifies the item(s) against the original sale.
5. The system calculates the amount to be refunded.
6. The cashier processes the return and initiates the refund (cash, credit card reversal, etc.).
7. The system updates the inventory to reflect the return.
8. The system generates a return receipt.
9. The cashier provides the return receipt to the customer and completes the transaction.

**Alternative Flows:**

- B1: Item Not Eligible for Return
  If the item is not eligible for return (e.g., outside return window, not in original condition), the system alerts the cashier, and the return is denied.
- B2: Refund Processing Error
  If there's an error during the refund process, the system notifies the cashier, and the transaction may be escalated to the administrator for resolution.

# QUE : Identify Entity/Boundary Control Objects

ANS:

## Entity Objects

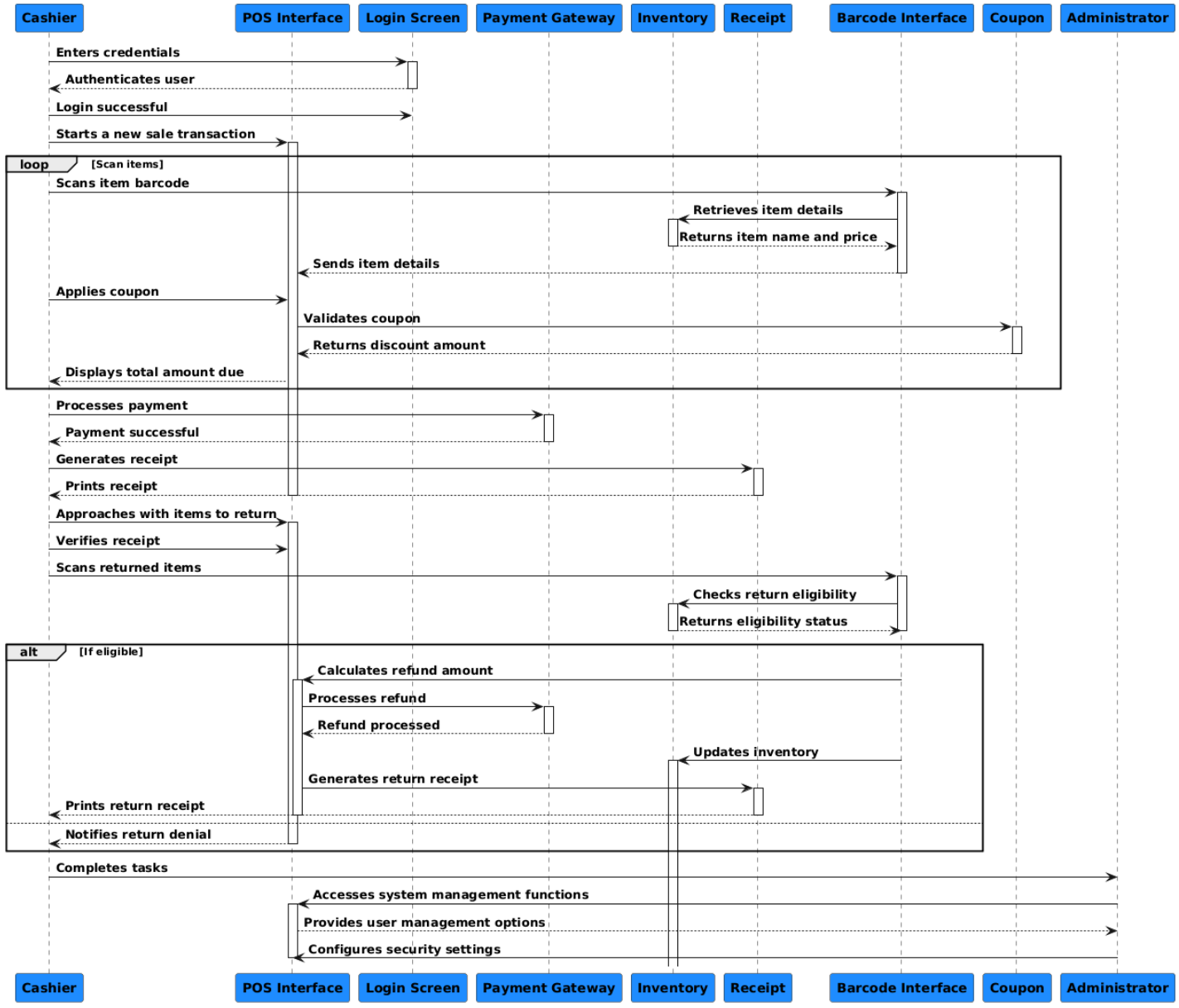
1. Product
2. Transaction
3. Cashier
4. Receipt
5. Coupon
6. Inventory

## Boundary Objects

1. POS Interface
2. Payment Gateway
3. Login Screen
4. Barcode interface
5. Printer Interface
6. Admin management

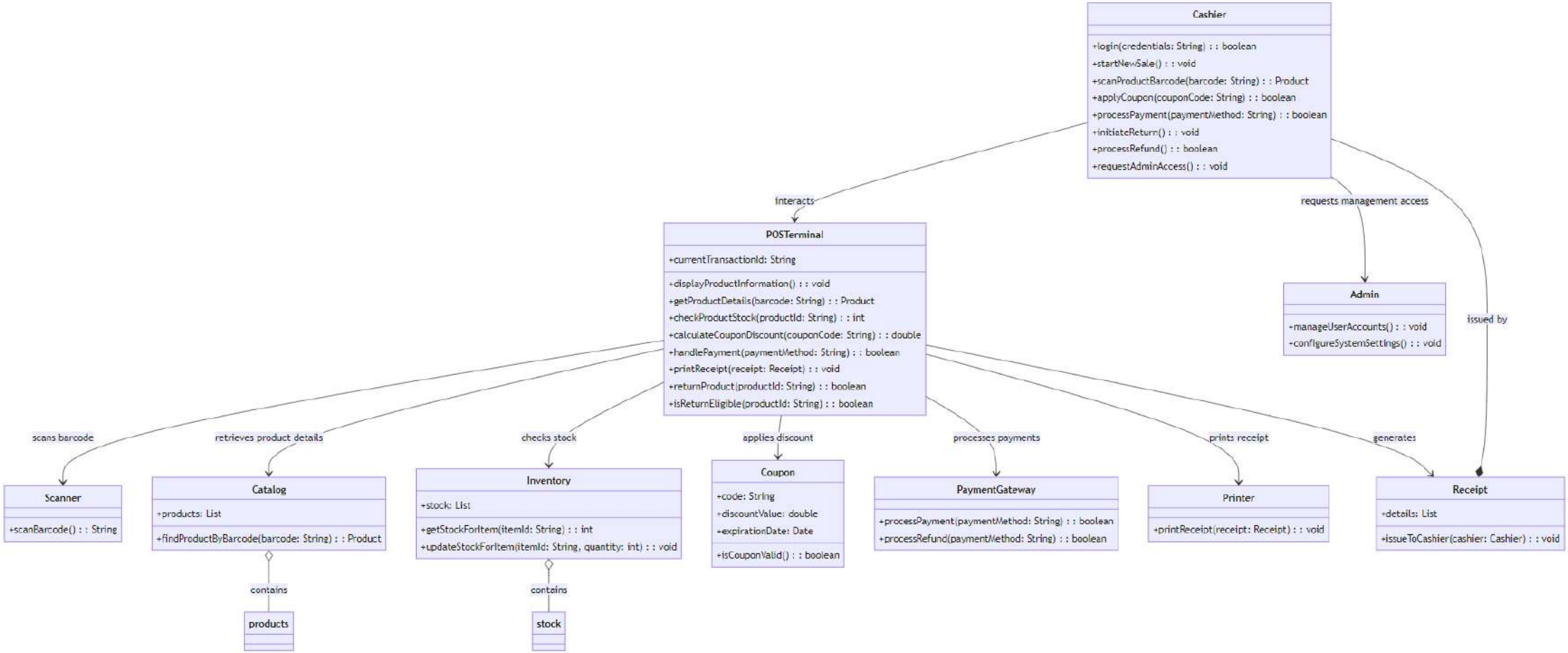
# QUE - Develop Sequence Diagrams.

## ANS:



## QUE - Develop Analysis Domain Models

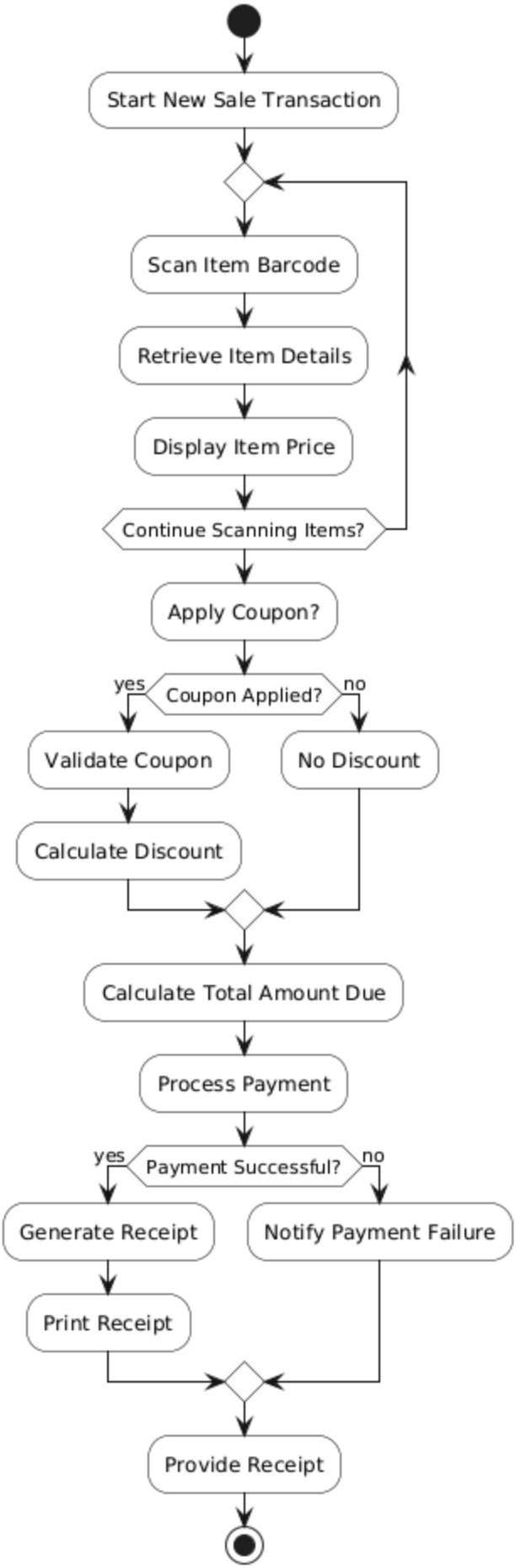
## ANS:


Cashier
+login(credentials: String) : : boolean
+startNewSale() : : void
+scanProductBarcode(barcode: String) : : Product
+applyCoupon(couponCode: String) : : boolean
+processPayment(paymentMethod: String) : : boolean
+initiateReturn() : : void
+processRefund() : : boolean
+requestAdminAccess() : : void
interacts
requests management access
issued by
POSTerminal
+currentTransactionId: String
+displayProductInformation() : : void
+getProductDetails(barcode: String) : : Product
+checkProductStock(productId: String) : : int
+calculateCouponDiscount(couponCode: String) : : double
+handlePayment(paymentMethod: String) : : boolean
+printReceipt(receipt: Receipt) : : void
+returnProduct(productId: String) : : boolean
+isReturnEligible(productId: String) : : boolean
Admin
+manageUserAccounts() : : void
+configureSystemSettings() : : void
scans barcode
retrieves product details
checks stock
applies discount
processes payments
prints receipt
generates
Scanner
+scanBarcode() : : String
Catalog
+products: List
+findProductByBarcode(barcode: String) : : Product
Inventory
+stock: List
+getStockForItem(itemId: String) : : int
+updateStockForItem(itemId: String, quantity: int) : : void
Coupon
+code: String
+discountValue: double
+expirationDate: Date
+isCouponValid() : : boolean
PaymentGateway
+processPayment(paymentMethod: String) : : boolean
+processRefund(paymentMethod: String) : : boolean
Printer
+printReceipt(receipt: Receipt) : : void
Receipt
+details: List
+issueToCashier(cashier: Cashier) : : void
contains
products
contains
stock

## QUE - Develop activity diagram for "Process Sale" and "Handle Return" use cases.

ANS:

PROCESS SALE:


Start New Sale Transaction
Scan Item Barcode
Retrieve Item Details
Display Item Price
Continue Scanning Items?
Apply Coupon?
yes
Coupon Applied?
no
Validate Coupon
No Discount
Calculate Discount
Calculate Total Amount Due
Process Payment
yes
Payment Successful?
no
Generate Receipt
Notify Payment Failure
Print Receipt
Provide Receipt

## HANDLE RETURN: