LAB-6
Name: Mausam Kamdar
ID: 202201372

1. Use Case Textual Descriptions:

Use Case: Process Sale

Goal: To handle a customer sale and process payment.

Primary Actor: Cashier

Preconditions:

- Cashier is logged into the POS system.
- The system is connected to the inventory and catalog systems.
- The system should be fully operational

Trigger: A customer arrives at the counter for purchase of goods

Main Scenario Flow:

1. The cashier selects "New Sale" in the POS system.
2. The cashier scans the barcode of each product.
3. The system retrieves and displays the product details (name, price) from the catalog system.
4. The system checks the inventory system and updates stock quantities for each scanned product.
5. The cashier applies any gift coupons or promotions.
6. The system calculates the total price.
7. The customer chooses the payment method (cash, credit card, or check).
8. The system processes the payment.
9. The system prints the receipt.
10. The sale is completed.

Extensions/Alternate scenario:

- 3a. If a barcode is invalid, the system prompts the cashier to enter the product manually.
- 7a. If payment is declined, the system requests a different payment method.
-

Postconditions:

- The inventory is updated.

- The sale transaction is saved in the system.

Use Case: Handle Return

Goal: To process a customer return and adjust inventory.

Primary Actor: Cashier

Trigger: A customer requests to return items purchased from the store.

Preconditions:

- Cashier is logged into the POS system.
- The customer has a valid receipt or proof of purchase.

Main Scenario Flow:

1. The cashier selects "Return Item" in the POS system.
2. The cashier scans the receipt
3. The system retrieves the transaction and the details of the items purchased.
4. The cashier selects the items to be returned.
5. The system verifies the return policy (e.g., return period, condition of items).
6. The system processes the return, adjusts the inventory, and updates stock levels.
7. The system initiates the refund process (cash, card, or store credit).
8. The system prints a return receipt.
9. The return is completed.

Extensions/ Alternate Scenario::

- 5a. If the return policy is not met (e.g., past return period), the system rejects the return.
- 7a. If the current payment method does not work, the customer chooses another one
- 2a. If the scanner is unable to identify the receipt, the cashier enters details manually

Postconditions:

- The inventory is updated.
- The return transaction is saved in the system.

2. Identify Entity/Boundary Control Objects

Use Case: Process Sale
Entity object:
1. Cashier,
2. Catalog system
3. Inventory system
4. Product

Boundary objects:
1. Barcode scanner
2. User Interface
3. Payment interface
4. Receipt printer

Control objects:
1. Payment controller
2. Sale processor
3. Inventory manager
4. Catalog manager

Use Case:Handle return
Entity object:
1. Cashier
2. Inventory
3. Product

Boundary object:
1. Cashier interface
2. Refund interface

Control object:
1. Refund controller
2. Return processor
3. Inventory manager

Cashier | Sale interface | Process Sale Controller | Catalog lookup controller | Catalog system | Inventory update controller | Inventory system | Payment Controller | Receipt Printing interface

Start sale transaction

Process Sale

Request item details

Catalog lookup

Return item detail

Send item detail

Check and Update Stock

Update stock details

Stock updated

Confirm stock update

Select payment method

Payment method

Payment

Payment success

Payemnt confirmation

Generate bill

Sale
+saleID : string
+date : DateTime
+total : float
+calculateTotal()

Item
+itemID : string
+name : string
+price : float
+description : string

PaymentMethod
+processPayment(amount : float)

Receipt
+receiptID : string
+saleID : string
+items : Item[]
+total : float
+paymentMethod : string
+printReceipt()

User
+userID : string
+username : string
+role : string

InventoryRecord
+itemID : string
+quantity : int
+updateQuantity(change : int)

CardPayment
+cardNumber : string
+expiryDate : string
+processPayment(amount : float)

CashPayment
+amountTendered : float
+processPayment(amount : float)
+calculateChange() : : float

GiftCoupon
+couponCode : string
+value : float
+processPayment(amount : float)

```
                    ●

              ┌──────────────┐
              │ Login to POS │
              │    system    │
              └──────────────┘
                     │
              ┌──────────────┐
              │  Start sale  │
              │ transaction  │
              └──────────────┘
                     │
              ┌──────────────┐
              │  Scan item   │
              └──────────────┘
                     │
              ┌────────────────────┐
              │ Retrieve item details │
              └────────────────────┘
                     │
              ┌──────────────┐
              │ Update stock │
              └──────────────┘
                     │
                    ◇ Payment type
     Card                              Gift card
      │         Cash      Cheque           │
┌──────────┐ ┌──────────┐ ┌──────────┐ ┌──────────┐
│ Process  │ │ Process  │ │ Process  │ │ Process  │
│   Card   │ │   Cash   │ │  Cheque  │ │ Gift card│
│ Payment  │ │ payment  │ │ payment  │ │ payment  │
└──────────┘ └──────────┘ └──────────┘ └──────────┘
                  │
           ┌──────────────┐
           │ Generate bill │
           └──────────────┘
                  │
                  ●
```