Software Engineering IT-314

Name : Kavit Patel

Student ID : 202201290

Use Case: Process Sale

Actor: Cashier

- ## Preconditions

1. The cashier has already logged into the point-of-sale (POS) system.
2. The POS system is successfully connected to both the catalog and inventory systems.

- ## Main Flow

1. The cashier begins a new sales transaction.
2. For every item in the transaction: a. The cashier scans the barcode of the item. b. The system retrieves the item's name and price from the product catalog. c. The inventory count is adjusted accordingly. d. The item is added to the ongoing transaction.
3. The system calculates the total transaction amount.
4. If the customer has a gift coupon: a. The cashier applies the coupon to the transaction. b. The system adjusts the total amount after applying the coupon.
5. The cashier informs the customer of the total due.
6. The customer selects a payment method (cash, credit card, or check).

7. The cashier processes the chosen payment method.

8. The system verifies the payment details.

9. A receipt is generated and printed by the system.

10. The transaction is closed and finalized by the system.

● <u>Alternative Flows</u>

4a. Invalid coupon scenario:

1. The system alerts the cashier that the coupon is not valid.
2. The cashier informs the customer of the invalid coupon and proceeds with the transaction from step 5.

8a. Payment validation failure:

1. The system notifies the cashier that the payment validation has failed.
2. The cashier requests that the customer provide a different payment method.
3. If the customer provides an alternative method, return to step 7; if not, cancel the transaction.

● <u>Postconditions</u>

1. The transaction is logged in the system.
2. Inventory is updated to reflect the sale.
3. The payment is successfully processed.
4. A receipt has been printed.

Use Case: Processing a Return

Actor: Cashier

● Preconditions

1. The cashier is already logged into the POS system.
2. The customer possesses a valid receipt for the items they wish to return.

● Main Flow

1. The cashier begins a new return transaction.
2. The cashier either scans the receipt or manually inputs the receipt details.
3. The system fetches the original transaction information.
4. For each item being returned: a. The cashier scans the item's barcode. b. The system checks that the item matches the original transaction. c. The inventory levels are updated to reflect the return. d. The system adds the item to the ongoing return transaction.
5. The system calculates the total refund that is due.
6. The cashier confirms the return details with the customer.
7. The system processes the refund, using the same payment method as the original purchase.
8. A return receipt is generated and printed by the system.

9. The system closes and completes the return transaction.

● Alternative Flows

2a. Receipt not found:

1. The system alerts the cashier that the receipt is invalid or not located.
2. The cashier informs the customer that the return cannot proceed and terminates the process.

4b. Item verification fails:

1. The system flags the item as not matching the original transaction.
2. The cashier informs the customer and either moves to the next item or stops the return process.

7a. Original payment method unavailable:

1. The cashier selects a different refund option (e.g., store credit or cash).
2. The system processes the refund using the newly chosen method.

● Postconditions

1. The return is recorded in the system.
2. Inventory is updated to reflect the returned item(s).
3. The refund has been successfully processed.
4. A return receipt is printed and given to the customer.

Identify Entity/Boundary/Control Objects

Entity Objects:

- Item
- Inventory
- Catalog
- Payment
- Receipt
- Coupon
- User (Cashier/Administrator)
- Return

Boundary Objects:

- POS Terminal Interface

- Scanner Interface

- Payment Processing Interface

- Receipt Printer Interface

Control Objects:

- Sale Manager

- Inventory Manager

- Catalog Manager

- Payment Manager

- User Authentication Manager

- Return Manager

Sequence Diagram:

Process Sale Sequence Diagram

Cashier — POS Terminal — Sale Manager — Inventory Controller — Catalog — Payment Controller — Receipt Printer

Initiate Sale → create new sale
Sale created
loop
 Get item details
 Item details
 Update Inventory
 Inventory updated
Scan item
 Add item to sale
 Item added
Calculate total
Total calculated
[Apply Coupon]
opt
 Recalculate total
 Apply Coupon   New total
Select Payment Method   Process Payment
 Payment Confirmation
 Print Receipt
 Receipt Printed
Transaction Complete

Sequence Diagram:

Handle Return Sequence Diagram

Analysis Domain Model

Customer

+customerID: String
+name: String
+email: String
+phone: String

+useCoupon(couponCode: String) : : void
+viewPurchaseHistory() : : List

1   1

has   interacts with
0..*   0..*

Purchase

+purchaseID: String
+purchaseDate: Date
+totalAmount: float
+items: List

Administrator

+adminID: String
+name: String

+manageUsers() : : void
+configureSecurity() : : void

Cashier

+employeeID: String
+name: String

+processSale() : : void
+handleReturn() : : void
+verifyCustomer(customerID: String) : : Customer

1   1
contains   1   manages   uses
0..*   1

Product

+productID: String
+name: String
+price: float
+stockQuantity: int
+category: String

POSSystem

+startTransaction() : : void
+scanProduct(barcode: String) : : Product
+applyCoupon(couponCode: String) : : void
+processPayment(amount: float) : : boolean
+printReceipt() : : void
+manageInventory() : : void

1

1   1

tracked by   manages   connects to   applies
0..*   1   0..*

InventorySystem

+retrieveProduct(barcode: String) : : Product
+updateInventory(productID: String, quantity: int) : : void
+checkStock(productID: String) : : int

PaymentGateway

+processPayment(amount: float, paymentMethod: String) : : boolean
+validatePaymentDetails(details: PaymentDetails) : : boolean

Coupon

+couponCode: String
+discountAmount: float
+expiryDate: Date

+isValid() : : boolean

1
validates
0..1

PaymentDetails

+paymentMethod: String
+cardNumber: String
+expiryDate: Date
+cvv: String

Process Sale Activity Diagram

start

initiate sale

scan item

YES

calculate total

More Items?

NO

retrieve item info

apply coupon?

YES

apply coupon

update display

NO

Recalculate total

update inventory

Generate receipt

process payment

YES

apply coupon?

NO

print receipt

Generate receipt

finalize Transaction

end

Handle Return Activity Diagram

start

initiate sale

scan Receipt

Retrieve Original sale

More items to return?

YES → scan Returned Item

NO → calculated total Refund

Item in original sale?

YES → calculated Refund amount

NO → cancel Item Return

Process refunds

generate return Receipt

print Return Recirpt

finalized return

update inventory

end