`

LAB-6

IT314

Kanishk Kunj
(202201080)

A Problem Description

A POS (Point-Of-Sale) system is a computer system typically used to manage the sales in retail stores. It includes hardware components such as a computer, a bar code scanner, a printer and also software to manage the operation of the store.

The most basic function of a POS system is to **handle sales**. When a customer arrives at a POS counter with goods to purchase, the cashier will start a new sale transaction. When the barcode of a good is read by the POS system, it will retrieve the name and price of this good from the backend catalog system and interact with the inventory system to deduce the stock amount of this good. When the sale transaction is over, the customer can pay in cash, credit card or even check. After the payment is successful, a receipt will be printed. Note that for promotion, the store frequently issues gift coupons. The customer can use the coupons for a better price when purchasing goods. Another function of a POS system is to handle returns.... [The details of which are not given here] A user must log in to use the POS. The users of a POS system are the employees of the store including cashiers and the administrator. The administrator can access the system management functions of the POS system including user management and security configuration that cashiers can’t do.

Use Case Textual description for “process sale” and “handle return” cases

Use Case 1: Process Sale

Use Case ID: UC-01
Use Case Name: Process Sale
Actors: Cashier, Customer
Preconditions:

- The POS system is logged into by the cashier.
- The client needs to buy products.

Postconditions:

- The sales transaction has been finished.
- The amount of inventory is updated.
- The client receives a printed receipt.

Main Flow:

- In the POS system, a new sale transaction is started by the cashier.
- Every item that a customer presents is scanned by the cashier.
    1. **System Action:** The item name and price are retrieved by the POS from the backend catalog.
- The amount of scanned goods is subtracted by the system when it refreshes the inventory.
- The cashier verifies the products and shows the customer the total amount owed.
- The payment method (cash, credit card, or cheque) is chosen by the consumer.
- Using the POS system, the cashier completes the payment processing.
    1. **System Action:** The payment method is confirmed by the system.
- Following a successful payment, a receipt is generated by the system.
- After completing the purchase, the cashier gives the customer their receipt.

Alternative Flows:

- **Payment Failure:** The cashier is notified by the system if the payment is failed, and they have the opportunity to provide an other method of payment or try again.
- **Application of Coupons:** The cashier inputs the coupon code before verifying the total if the customer has a gift coupon. The new amount owed is displayed once the system recalculates the total.

Use Case Textual Descriptions

Use Case 1: Process Sale

Use Case ID: UC-01
Use Case Name: Process Sale
Actors: Cashier, Customer
Preconditions:

- The cashier is logged into the POS system.
- The customer has goods to purchase.

Postconditions:

- A sales transaction is completed.
- Inventory levels are updated.
- A receipt is printed for the customer.

Main Flow:

1. The cashier initiates a new sale transaction in the POS system.
2. The cashier scans the barcode of each item presented by the customer.
    - **System Action:** The POS retrieves the item name and price from the backend catalog.
3. The system updates the inventory by deducting the quantity of scanned items.
4. The cashier confirms the items and displays the total amount due to the customer.
5. The customer selects a payment method (cash, credit card, or check).

6. The cashier processes the payment through the POS system.
    - **System Action:** The system verifies the payment method.
7. Upon successful payment, the system generates a receipt.
8. The cashier hands the receipt to the customer and completes the transaction.

Alternative Flows:

- **Payment Failure:** If the payment is unsuccessful, the system notifies the cashier, who can retry the payment or offer alternative options.
- **Coupon Application:** If the customer has a gift coupon, the cashier enters the coupon code before confirming the total. The system recalculates the total and displays the new amount due.

Use Case 2: Handle Return

Use Case ID: UC-02
Use Case Name: Handle Return
Actors: Cashier, Customer

Preconditions:

- The cashier is logged into the POS system.
- The customer presents items for return.

Postconditions:

- The return transaction is completed.
- Inventory levels are updated accordingly.
- A return receipt is printed.

Main Flow:

1. The cashier initiates a return transaction in the POS system.
2. The cashier scans the barcode of the item(s) being returned.
    - **System Action:** The POS verifies the item's eligibility for return based on store policy (e.g., return period, condition).
3. The system displays the original sale information (price, date of sale).
4. The cashier confirms the return with the customer.
5. The system processes the return:

    - If the item is eligible, it updates the inventory by adding the returned item back to stock.
    - The system calculates any refund amount.
6. The cashier processes the refund to the customer via the original payment method (cash, credit card, etc.).
7. A return receipt is generated and printed for the customer.
8. The cashier hands the receipt to the customer and completes the return transaction.

Alternative Flows:

- **Item Not Eligible for Return:** If the item is not eligible for return, the system informs the cashier, who communicates the reason to the customer.
- **Partial Refund:** If a coupon was used during the original purchase, the system calculates any applicable deductions from the refund amount.

Entity/Boundary Control Objects.

Entity Objects

Entity objects represent the core data and state of the system. In a POS system, they may include:

1. **Product**

    ○ Attributes: barcode, name, price, description, stock amount.

2. **Transaction**

○ Attributes: transaction ID, date/time, total amount, payment method, list of products, status (completed, refunded).

3. **User**

○ Attributes: user ID, username, password, role (cashier or administrator), permissions.

4. **Coupon**

○ Attributes: coupon code, discount amount, expiration date, applicable products.

5. **Receipt**

○ Attributes: receipt ID, transaction details, date/time, total amount.

6. **Inventory**

○ Attributes: product ID, quantity available, reorder level.

Boundary Objects

Boundary objects define the interaction between the user (cashiers, administrators) and the system. They handle user inputs and outputs. In a POS system, they may include:

1. **Login Screen**

○ Accepts user credentials and initiates session.

2. **Sales Interface**

○ Displays product information, allows barcode scanning, shows current transaction total, and processes payments.

3. **Return Interface**

○ Allows cashiers to process product returns, display return policies, and handle refunds.

4. **Admin Dashboard**

○ Interface for administrators to manage users, view reports, and configure system settings.

5. **Receipt Printer**

○ Interface for printing transaction receipts.

Control Objects

Control objects manage the flow of information between entity and boundary objects. They enforce business rules and coordinate system operations. In a POS system, they may include:

1. **Transaction Controller**

    ○ Managesthe lifecycle of a transaction, including starting, processing payments, and completing or cancelling the transaction.

2. **User Management Controller**

    ○ Handles user login/logout, user creation, and permission assignment.

3. **Inventory Controller**

    ○ Manage Stock levels, updates inventory upon sales or returns, and checks for restocking needs.

4. **Coupon Controller**

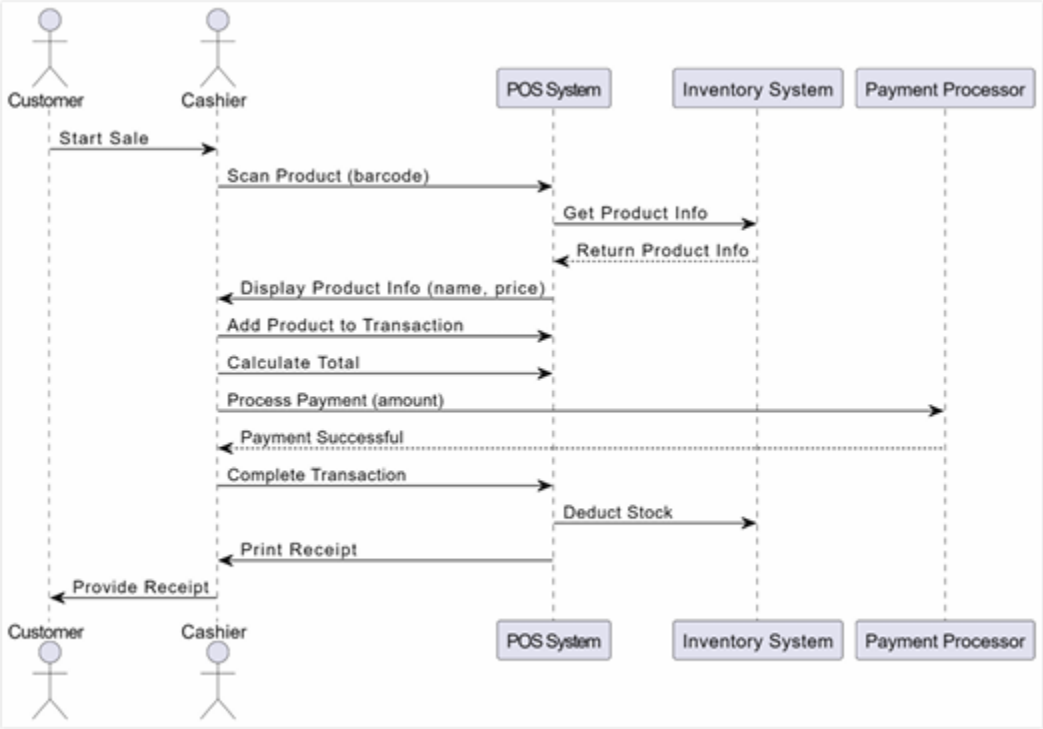
    ○ Validates coupon codes, applies discounts, and checks coupon eligibility based on conditions.

5. **Payment Processor**

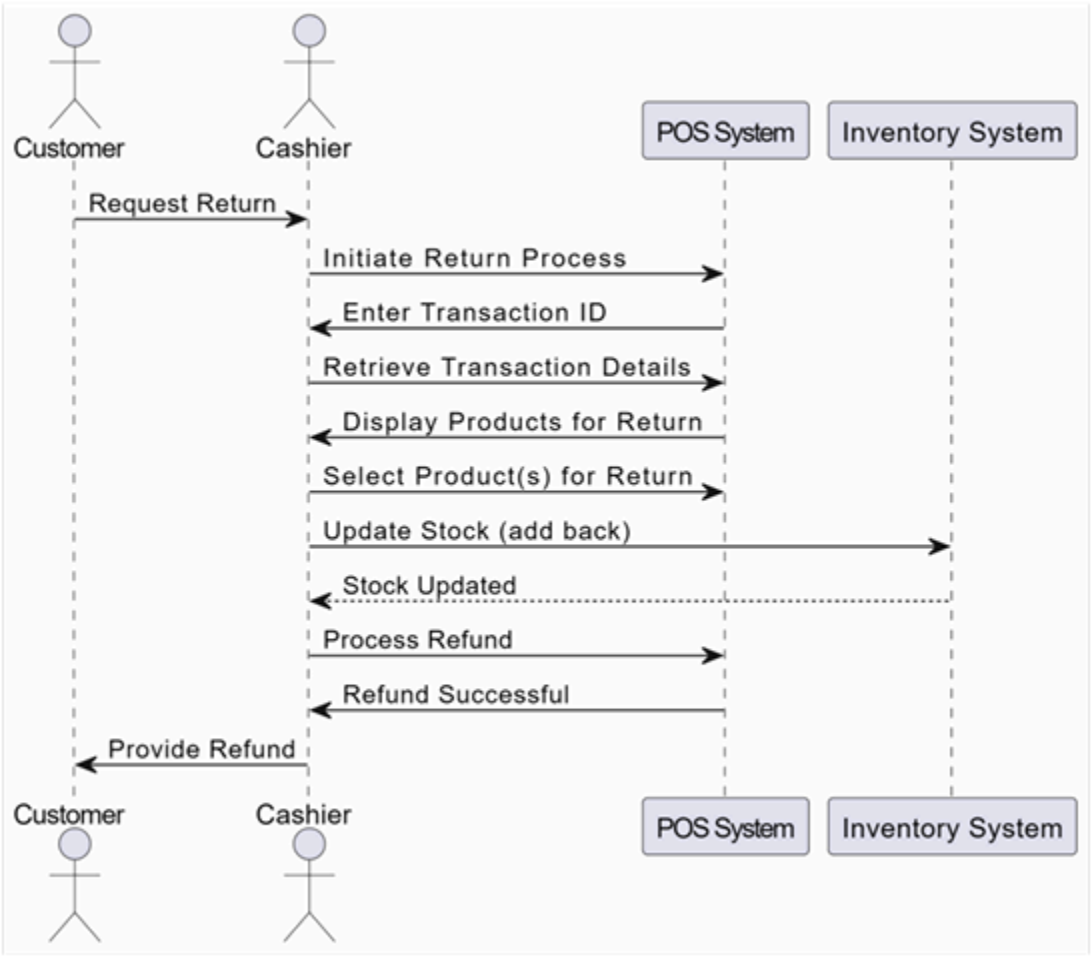
    ○ Handles different payment methods, verifies payment, and updates transaction

Sequence Diagrams

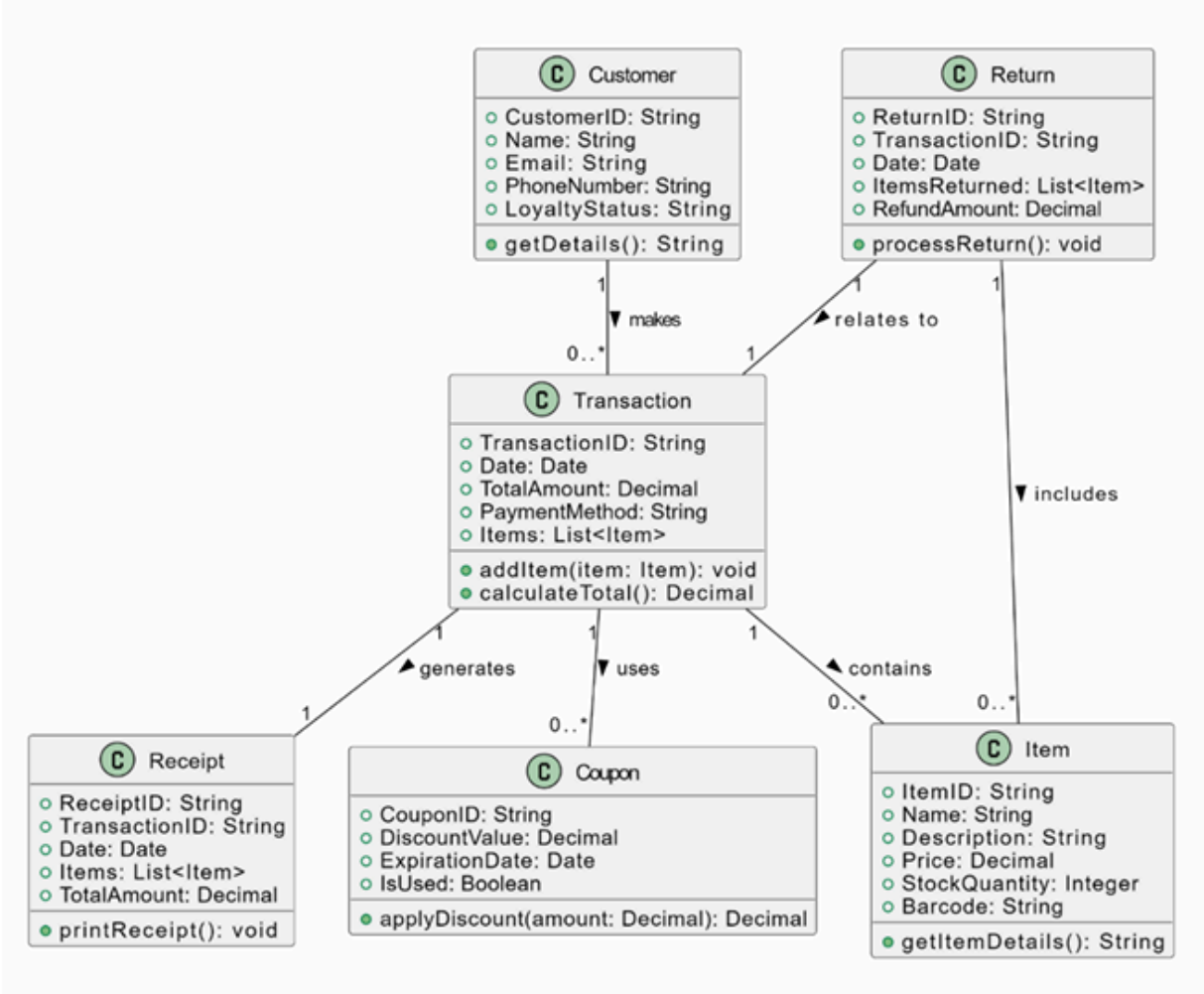
1. Sequence Diagram for Processing a Sale Transaction



2. Sequence Diagram for Handling a Return.

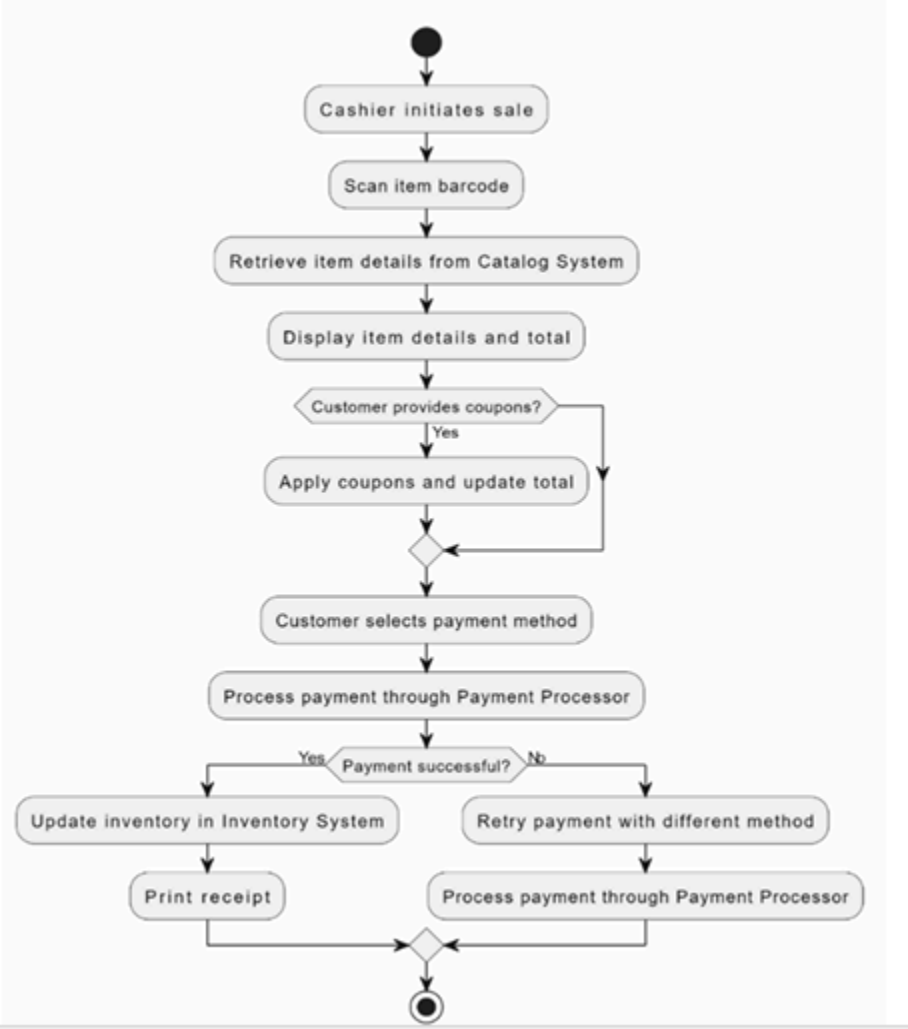


Analysis Domain Models



Activity diagram for "Process Sale" and "Handle Return" use cases.

1. Process Sale



2. Handle Return