Software engineering
Lab 6

202201151
Pandavadara Sahil Rasikbhai

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

Use Case 2: Handle Return

Use Case Name: Handle Return
Actors: Cashier, Customer, Inventory System
Preconditions:

- Cashier must be logged into the POS system.
- The customer must provide a valid receipt for the returned items.

Postconditions:

- The return transaction is successfully completed.
- Inventory is updated to reflect the returned goods.
- A refund is issued or store credit is provided.

Main Success Scenario:

1. The customer approaches the cashier with the items to return and provides a receipt.
2. The cashier selects the "Return" function in the POS system.
3. The cashier scans the receipt or manually enters the transaction details.
4. The system retrieves the sale information and verifies that the items are eligible for return.
5. The cashier scans the items being returned.
6. The system updates the return transaction with the item details and calculates the refund amount.
7. The inventory system is updated to reflect the returned items.

8. The cashier confirms the refund method with the customer (cash, credit card, or store credit).
9. The system processes the refund or issues store credit.
10. A return receipt is printed, and the transaction is completed.

Extensions (Alternative Scenarios):

- **Step 4:** If the receipt is not valid or the items are not eligible for return, the system alerts the cashier, and the return cannot proceed.
- **Step 8:** If the customer paid with a credit card, the refund is processed to the same card. If the payment method was cash, the cashier refunds the cash amount.

Identify Entity/Boundary Control Objects

1. Entity Objects (Domain Objects)

- Sale Transaction: Represents a single sale with details of items, total amount, discounts, and payment status.
- Item: Represents an individual product sold in the store with attributes such as name, price, and barcode.
- Customer: Represents a customer making a purchase or return (optional if not explicitly tracked).
- Inventory: Represents the stock levels of products in the store, interacting with the inventory system to manage item counts.
- Receipt: Represents the document that details a completed sale, return, or transaction.
- Payment: Represents a payment record including amount, and payment method (cash, credit card, check).
- Return Transaction: Represents a return event where the customer returns goods.
- Coupon: Represents a gift coupon that can be used for discounts or promotions.
- User: Represents the store's employee (cashier or admin) with details such as name, role, and credentials.

2. Boundary Objects (Interface Objects)

- POS Terminal Interface: User interface where the cashier interacts with the system to process sales, returns, and other functions.
- Barcode Scanner: Interface used by the cashier to scan items and retrieve information from the catalogue.
- Receipt Printer: Interface to the hardware that prints out sales and return receipts.
- Payment Terminal Interface: Interface to interact with the customer for card payments or cash entry.
- Login Screen: Interface where users (cashiers or administrators) log in to use the POS system.
- Catalogue System Interface: Interface that interacts with the backend catalogue system to fetch product details (e.g., name, price) based on the scanned barcode.
- Inventory System Interface: Interface to update the inventory in real-time when items are sold or returned.
- Coupon Validator Interface: Interface where coupons are validated for promotions during the transaction.

3. Control Objects (Controller Objects)

Control objects are responsible for coordinating the logic of use cases. They handle the workflow between entity objects and boundary objects.

- Sale Controller: Manages creating a sale transaction, including scanning items, applying discounts, interacting with the inventory and catalogue systems, and finalizing payment.
- Return Controller: Manages the workflow for returning items, verifying receipts, updating inventory, and processing refunds.
- Payment Controller: Coordinates interactions with the payment terminal (e.g., processing credit card, cash, or check payments) and ensures the transaction is finalized only after successful payment.
- Login Controller: Handles the authentication of users (cashiers or administrators) and ensures appropriate access rights (e.g., admin features for user management).
- Inventory Controller: Updates stock levels in the inventory system when sales or returns occur.
- Coupon Controller: Validates and applies coupons during a sale, ensuring that discounts are applied correctly.

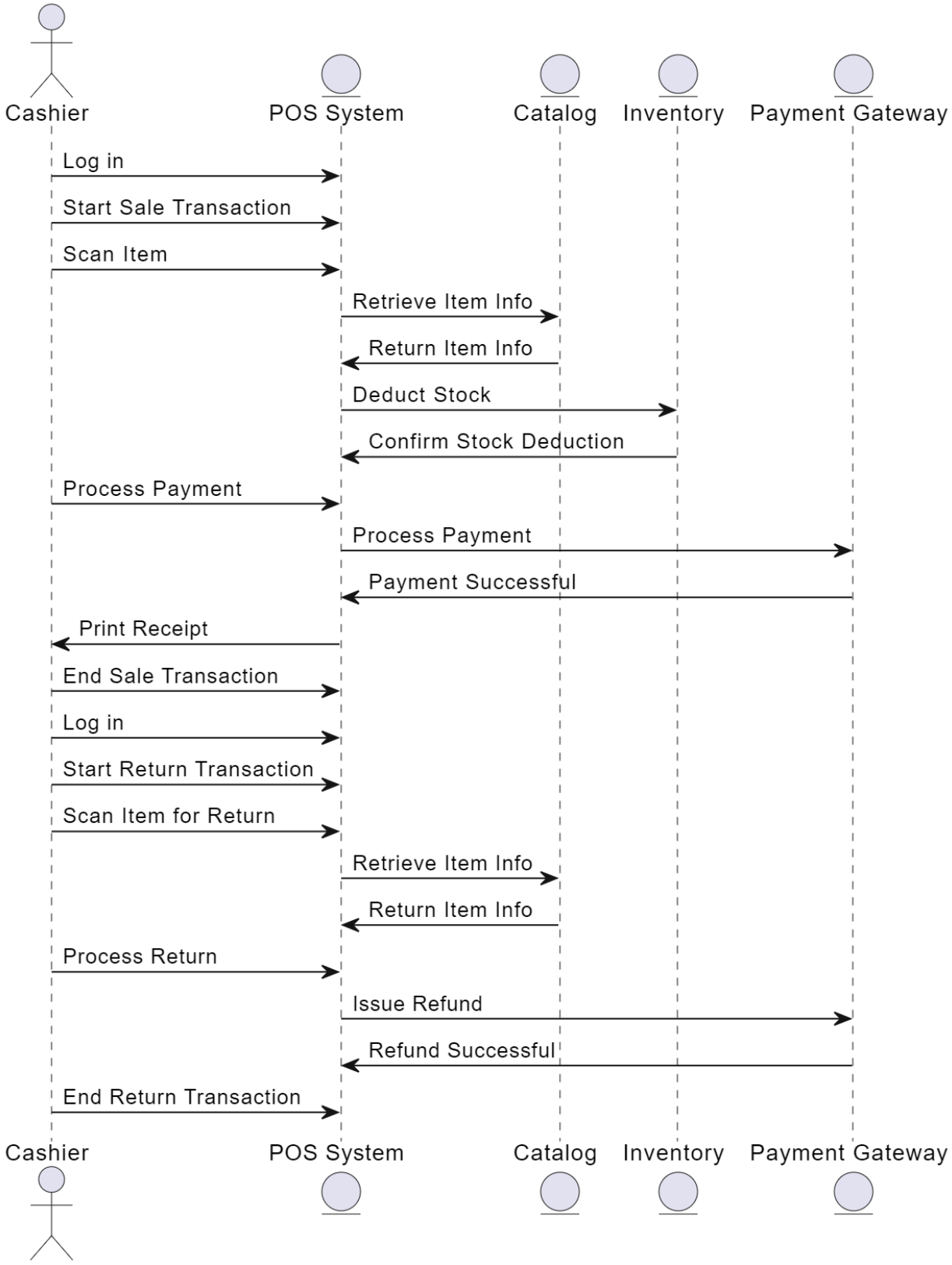
Example Mapping for Use Case: Process Sale

- Entity Objects:
    - Sale Transaction, Item, Inventory, Payment, Receipt, Coupon, User
- Boundary Objects:
    - POS Terminal Interface, Barcode Scanner, Payment Terminal Interface, Receipt Printer, Catalog System Interface, Inventory System Interface, Coupon Validator Interface
- Control Objects:
    - Sale Controller, Payment Controller, Inventory Controller, Coupon Controller

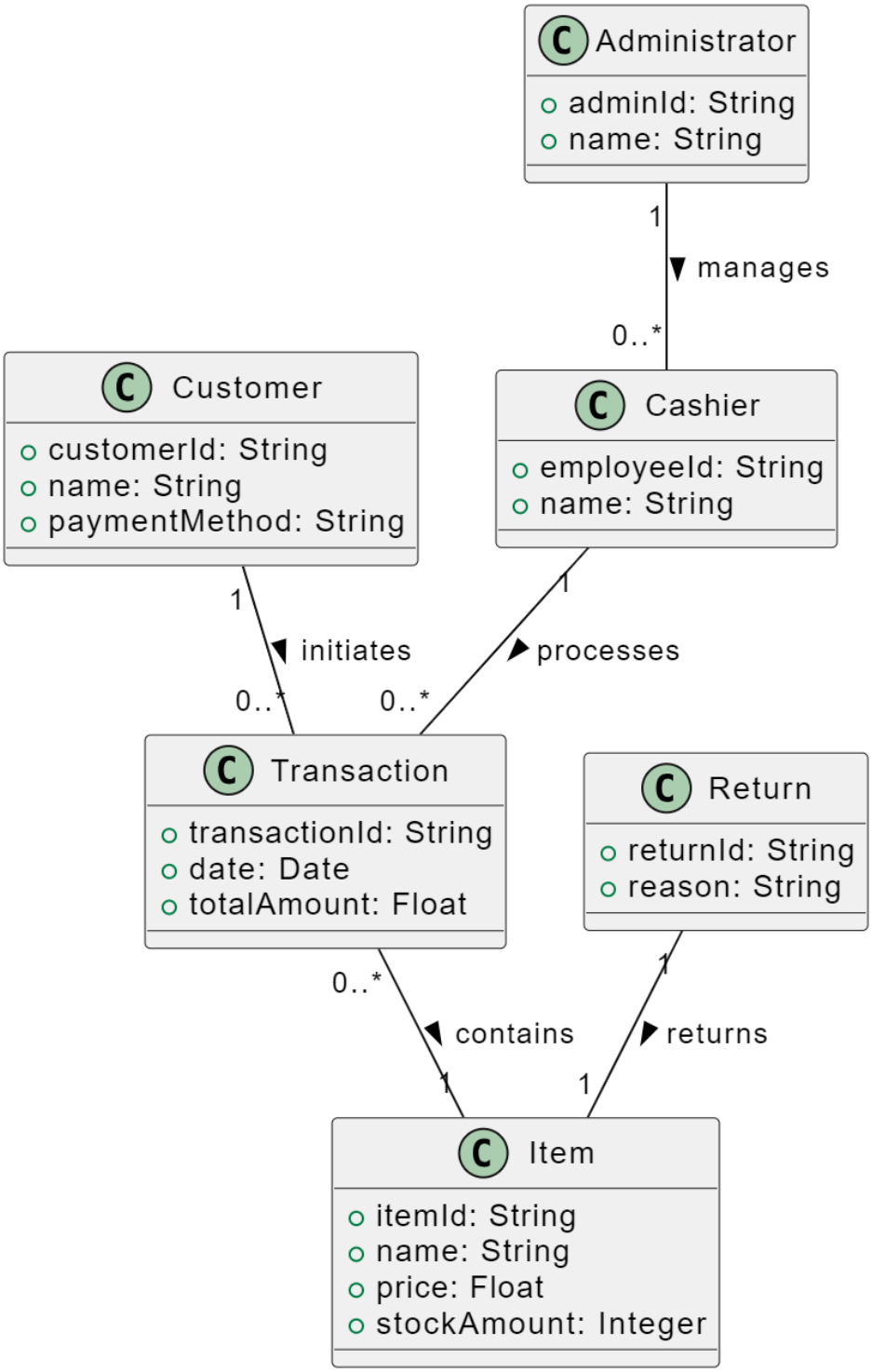
Example Mapping for Use Case: Handle Return

- Entity Objects:
    - Return Transaction, Item, Inventory, Payment, Receipt, User
- Boundary Objects:
    - POS Terminal Interface, Barcode Scanner, Payment Terminal Interface, Receipt Printer, Catalog System Interface, Inventory System Interface
- Control Objects:
    - Return Controller, Payment Controller, Inventory Controller

Develop Sequence Diagrams



Develop Analysis Domain Models



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