IT-314

Lab - 05

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

Yash Mehta | 202201309

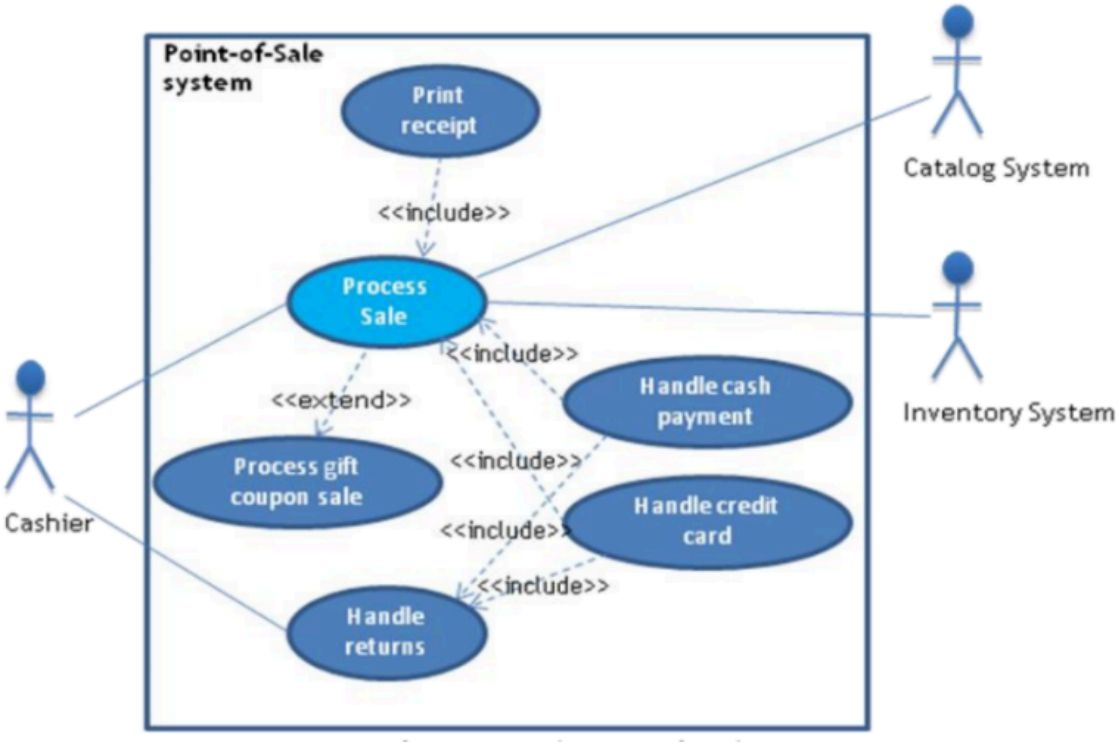
Table of Contents

1. *Problem statement……..……1*
2. *Solution……2*
    a. *Question 1……...2*
    b. *Question 2……...4*
    c. *Question 3……..…6*
    d. *Question 4……...7*
    e. *Question 5……...8*

Questions

Problem Statement:

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 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 issue 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.



Solutions

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

Use Case: Process Sale

Actors:

- **Primary Actor**: Cashier
- **Secondary Actors**: Customer, Inventory System, Payment System, Backend Catalog System

Preconditions:

- Cashier must be logged in to the POS system.
- The POS system is connected to the inventory and catalog system.
- The customer has goods to purchase.

Postconditions:

- The sale is completed.
- The stock is updated in the inventory system.
- The receipt is printed.
- The payment is recorded, and if applicable, the gift coupon is applied.

Main Flow:

- The cashier logs into the POS system.
- The cashier starts a new sale transaction.
- The cashier scans the barcode of each item.
- The POS system retrieves the item's name and price from the catalog system.
- The system interacts with the inventory system to deduct the stock amount.
- The cashier confirms the sale, and the system calculates the total price.
- The cashier applies any gift coupons (if available) to the transaction.
- The customer chooses a payment method (cash, credit, or check).
- The POS system processes the payment.
- Once the payment is successful, the system prints a receipt for the customer.
- The sale transaction is completed.

Alternative Flows:

- **Item Not Found**: If the scanned item is not found in the catalog, the system notifies the cashier to manually enter the item's details or remove it from the sale.
- **Payment Failure**: If the payment fails, the system prompts the cashier to retry or use a different payment method.

Use Case: Handle Return

Actors:

- **Primary Actor**: Cashier
- **Secondary Actors**: Customer, Inventory System

Preconditions:

- The customer must have a valid receipt for the return.
- The item being returned must be in acceptable condition as per the store's return policy.

Postconditions:

- The return is processed, and the item is either restocked or discarded.
- The customer receives a refund or store credit.

Main Flow:

- The cashier logs into the POS system.
- The customer presents a valid receipt for the item they wish to return.
- The cashier scans the receipt or enters the transaction ID into the system.
- The system retrieves the original sale details.
- The system verifies if the item is eligible for return based on the return policy.
- If eligible, the cashier confirms the return.
- The system updates the inventory, either restocking the item or marking it as discarded.
- The customer is refunded the amount paid, either in cash or store credit.
- The system generates a return receipt for the customer.
- The return transaction is completed.

Alternative Flows:

- **Receipt Not Found**: If the receipt is invalid or cannot be found, the cashier informs the customer and the return is denied.
- **Non-Returnable Item**: If the item does not meet the return policy (e.g., it's past the return period), the system alerts the cashier, and the return is not processed.

2. Identify Entity/Boundary Control Objects

a. Entity Objects

These represent the core data of the system:

- Sale: Represents a completed or ongoing sale transaction.
  - Attributes: saleID, totalAmount, date, paymentType
- Item: Represents each product being purchased or returned.
  - Attributes: itemID, name, price, quantity, stockAmount
- Customer: Represents the customer making the purchase or return.
  - Attributes: customerID, name, contactDetails
- Receipt: Represents the proof of purchase or return.
  - Attributes: receiptID, saleID, date, totalAmount, items[]
- Payment: Represents the payment details for the transaction.
  - Attributes: paymentID, amount, paymentType (cash, credit, check)
- Coupon: Represents any discount applied to the sale.
  - Attributes: couponCode, discountAmount
- Return: Represents an item that is being returned.
  - Attributes: returnID, saleID, itemID, refundAmount

b. Boundary Objects

These are interface objects that interact with users or external systems:

- CashierUI: The user interface through which the cashier interacts with the POS system.
  - Interfaces: logIn(), scanItem(), applyCoupon(), processPayment(), printReceipt(), handleReturn()
- Payment Terminal: Interface for interacting with the payment system for processing payments.
  - Interfaces: processCreditCard(), processCash(), processCheck()
- Catalog System Interface: Used to fetch product details (name, price) from the backend.
  - Interfaces: getProductDetails(barcode)
- Inventory System Interface: Interacts with the inventory system to update stock levels.
  - Interfaces: updateStock(itemID, quantity)
- ReceiptPrinter: Interface for printing receipts after a sale or return.
  - Interfaces: printReceipt()

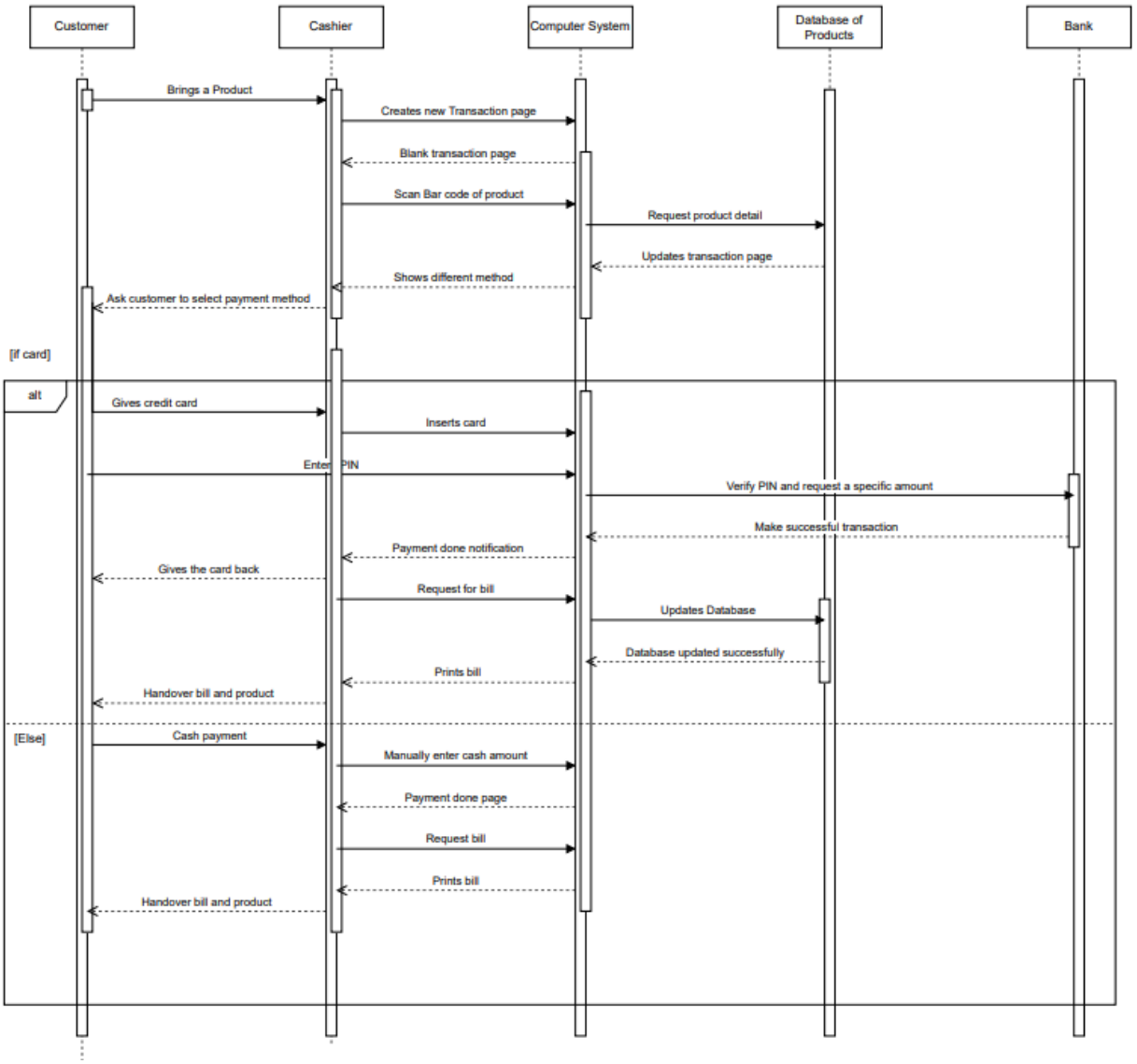
c. Control Objects

These manage the flow of operations for the use cases:

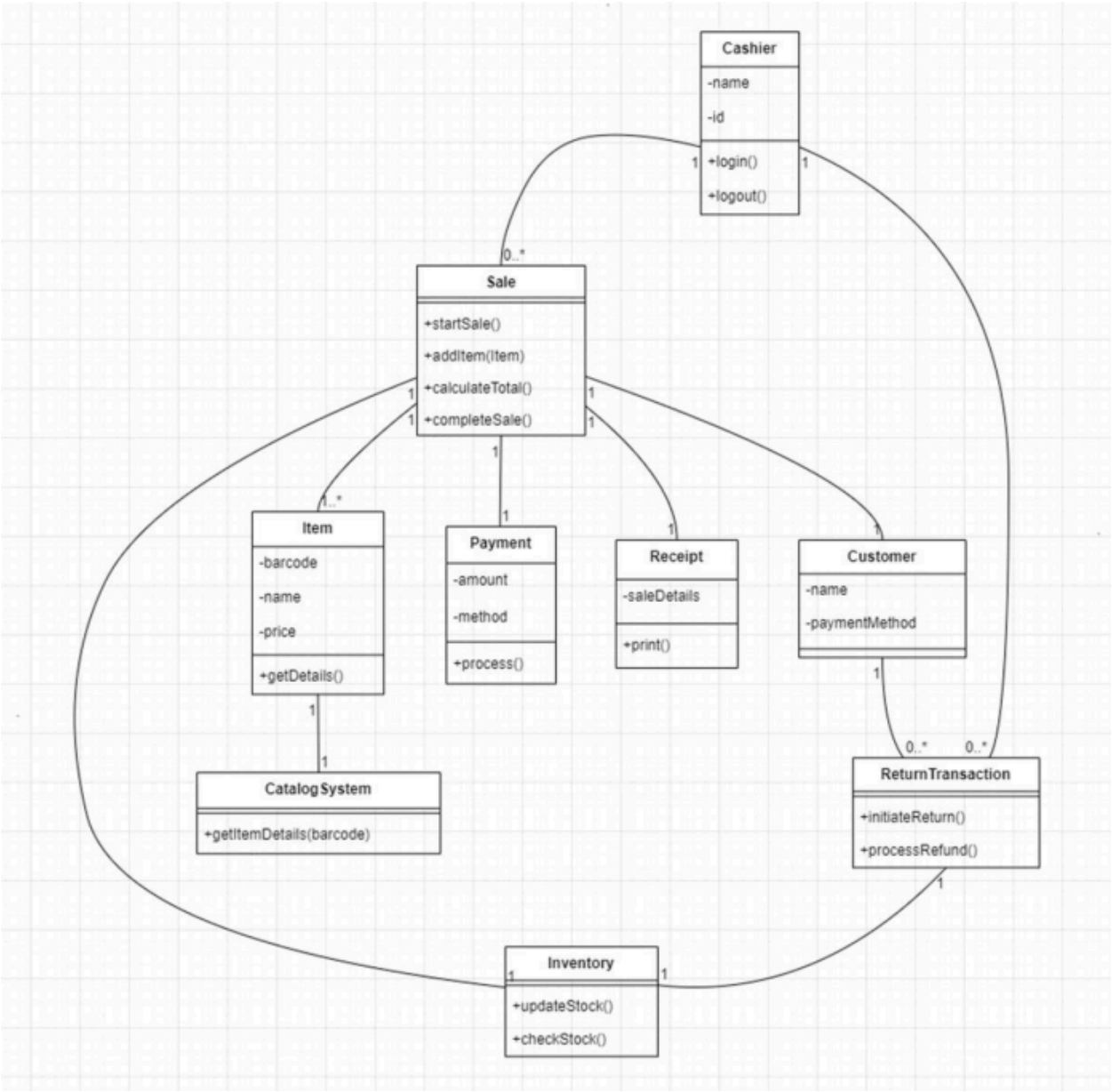
- SaleController: Manages the entire sale transaction process.

    - Functions: startSale(), scanItem(), calculateTotal(), applyCoupon(), completeSale(), printReceipt()
- PaymentController: Handles payment processing for the sale.
    - Functions: processPayment(paymentType, amount)
- ReturnController: Manages the return of items and refund processes.
    - Functions: scanReceipt(), verifyReturnEligibility(), processReturn(), printReturnReceipt()

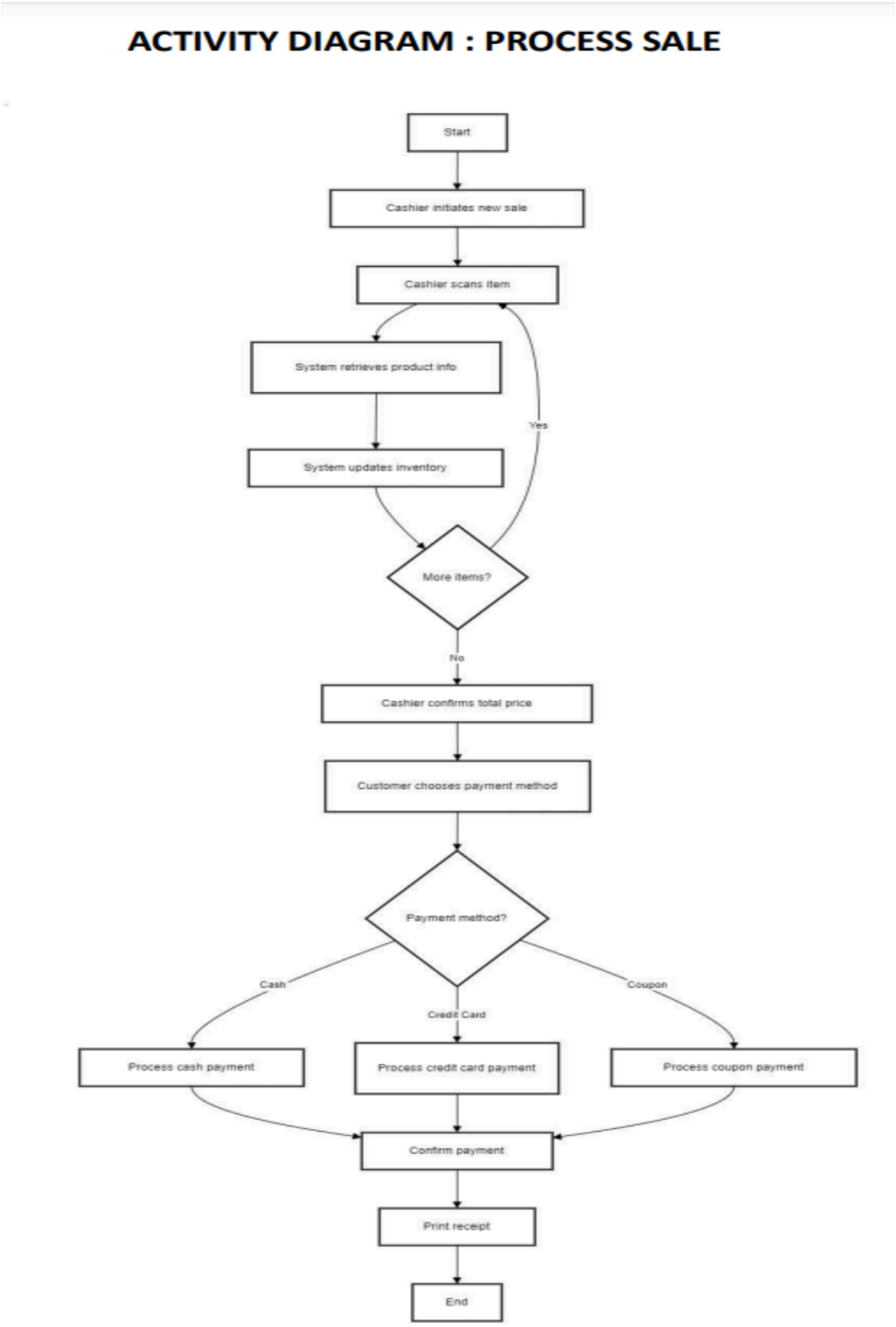
3. Develop Sequence Diagrams:



4. Develop Analysis Domain Models:



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



ACTIVITY DIAGRAM : PROCESS RETURN