IT314 SOFTWARE ENGINEERING
202201525-Heer Shah
Lab6-POS system

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

&

2) Identify Entity/Boundary Control Objects

Use Case: Process Sale

Actor: Cashier
Preconditions:

- **Cashier is logged into the POS system.**
- **Goods are available in stock.**

Postconditions:

- **Sale transaction is completed.**
- **Inventory is updated.**
- **Receipt is printed.**

Basic Flow:

1. **The cashier initiates a new sale transaction.**
2. **The cashier scans the barcode of the goods.**
3. **The system retrieves the product details (name and price) from the catalog.**
4. **System adds the product to the transaction.**

5. **The cashier can apply any gift coupons for discounts.**
6. **The system calculates the total amount due.**
7. **The customer makes the payment (cash, credit card, or check).**
8. **The system processes the payment and confirms its success.**
9. **A receipt is printed for the customer.**

Alternative Flow:

- **If the payment fails, the cashier gives the option to the customer to try a new payment method.**

Entity Objects:

- **Product (contains details such as name, price, barcode, etc.)**
- **Cashier**
- **Customer**
- **Inventory (manages stock levels)**
- **Transaction (Represents sale transactions)**
- **Receipt (Contains details of sales or returns)**
- **Coupon (To avail discount)**

Boundary Objects:

- **POS Interface**

Control Objects:

- **CatalogManager**
- **Payment Processor**
- **Inventory Controller**
- **SaleManager**
- **CouponManager**

Use Case: Handle Return

Actor: Cashier
Preconditions:

- **Cashier is logged into the POS system.**
- **Customer has a valid purchase receipt.**

Postconditions:

- **Returned goods are processed.**
- **Inventory is updated.**
- **Receipt for the return is printed.**

Basic Flow:

1. **The cashier initiates the return process.**
2. **The cashier scans the barcode of the returned goods.**
3. **The system verifies the item against the purchase receipt.**
4. **The system checks the return policy (validity period, condition of goods).**
5. **If the return is valid, the system updates the inventory to reflect the returned item.**
6. **The system processes the refund (cash, credit, or store credit).**
7. **A return receipt is printed for the customer.**

Alternative Flow:

- **If the item is not eligible for return (out of return period, damaged, etc.), the system informs the cashier and cancels the return process.**

Entity Objects:

- **Product (contains details such as name, price, barcode, etc.)**
- **Cashier**
- **Inventory (manages stock levels)**
- **Receipt (tracks purchase and return information)**
- **Coupon (To avail discount)**
- **ReceiptPrinter**

Boundary Objects:

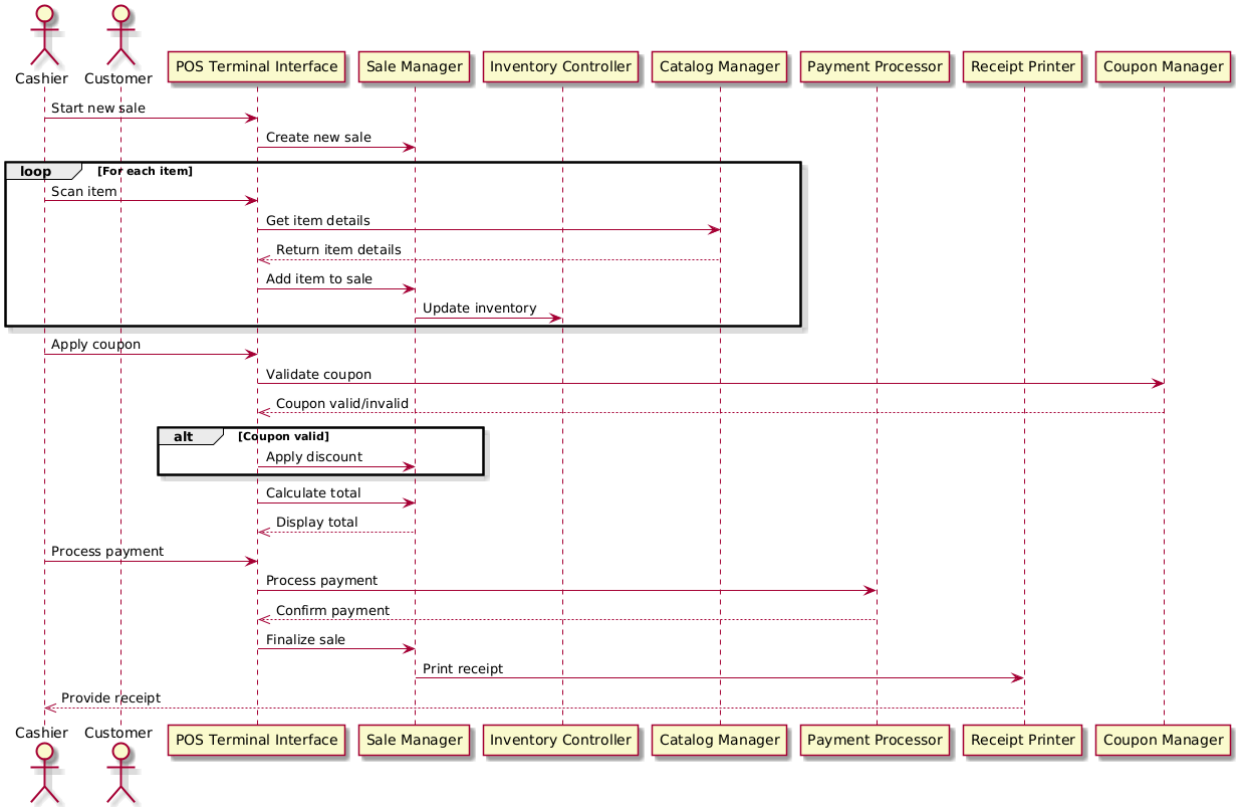
- **POS Interface**

Control Objects:

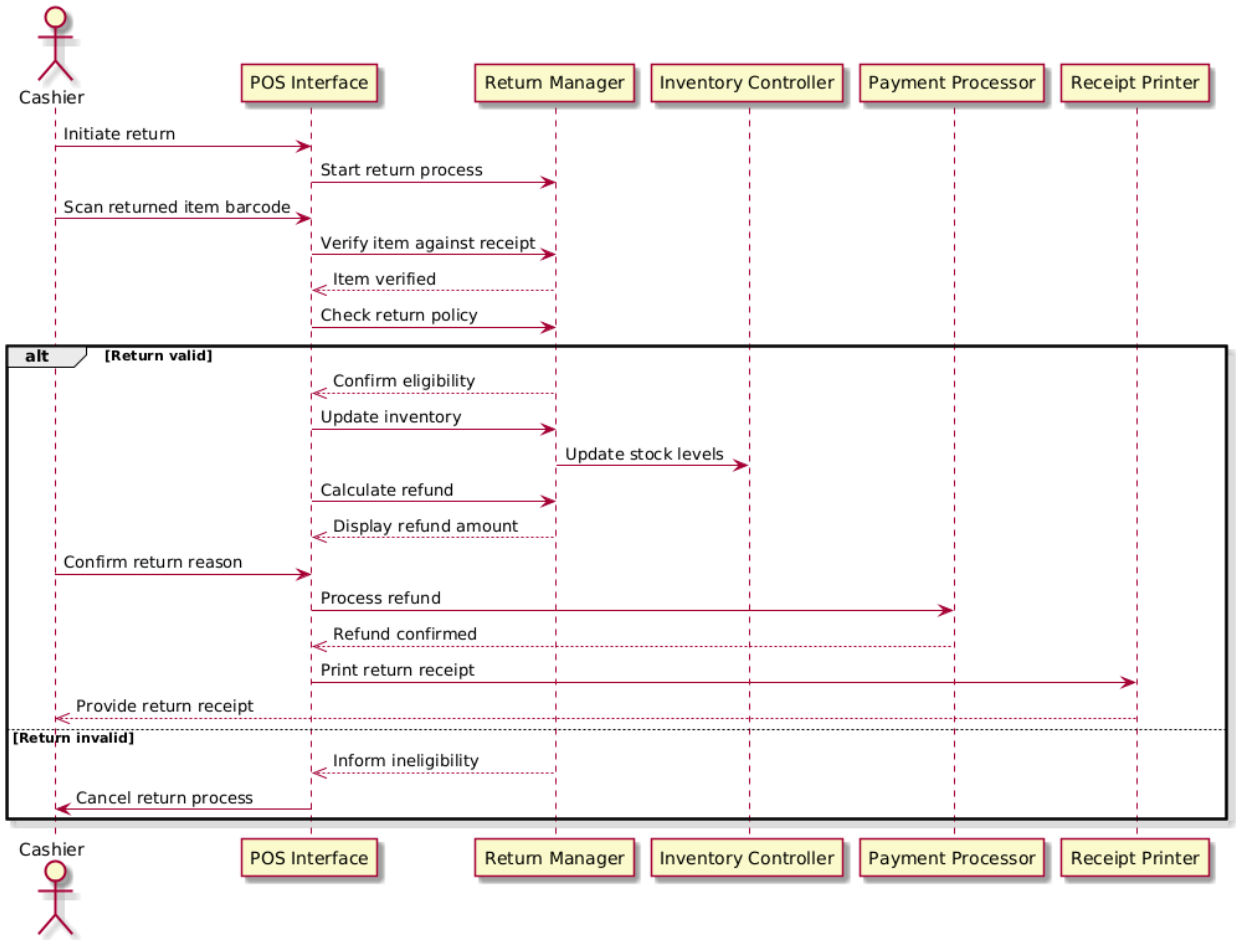
- **ReturnManager (Manages the return process, including verifying eligibility and calculating refunds.)**
- **InventoryController**
- **PaymentProcessor**

3) Develop Sequence Diagrams

i) Process Sale

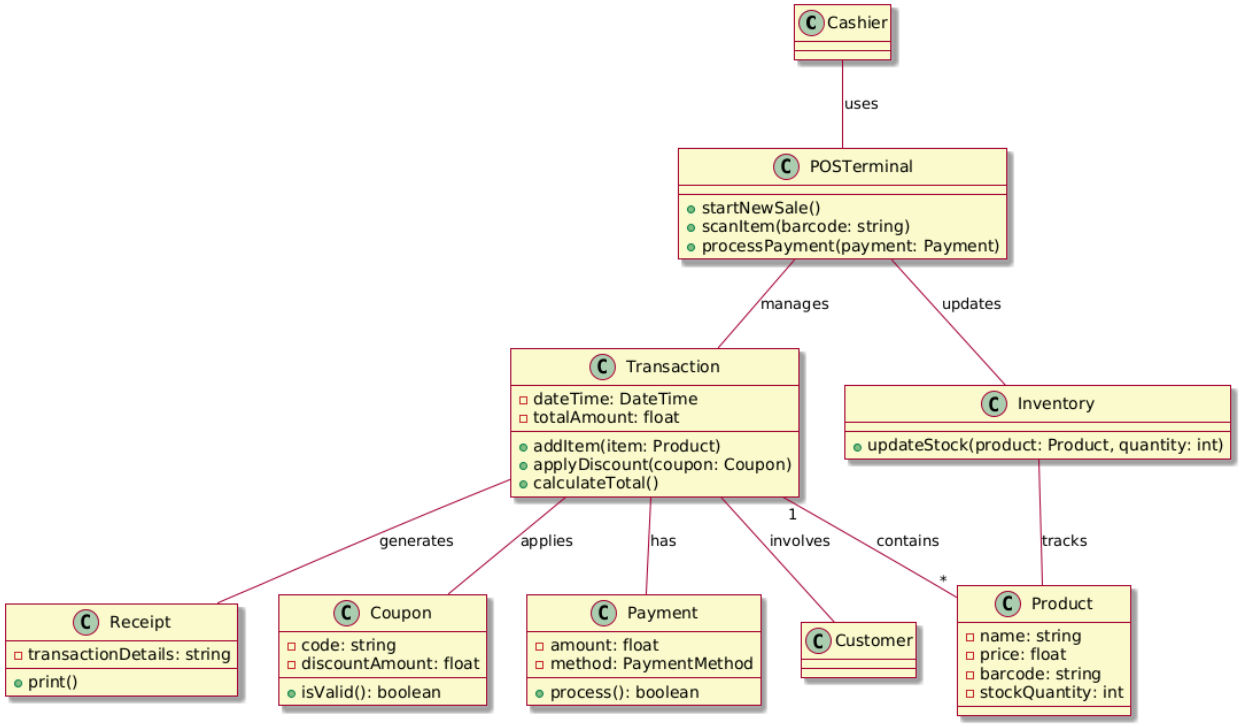


ii) Handle Return

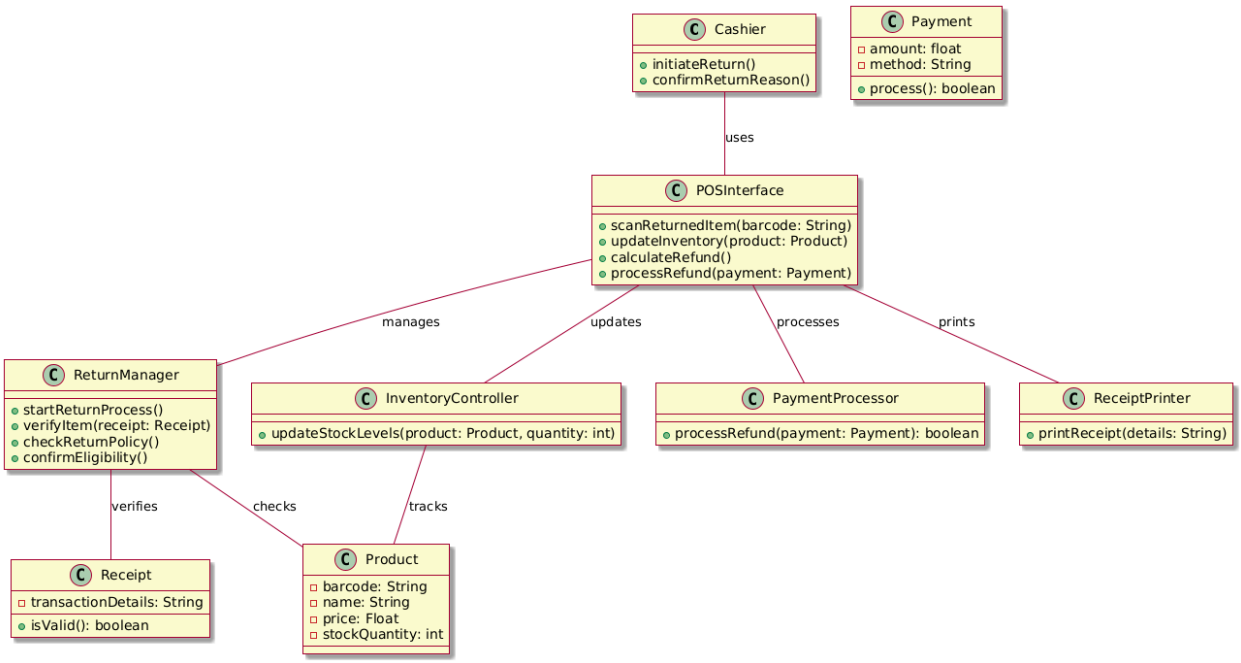


4) Develop Analysis Domain Models

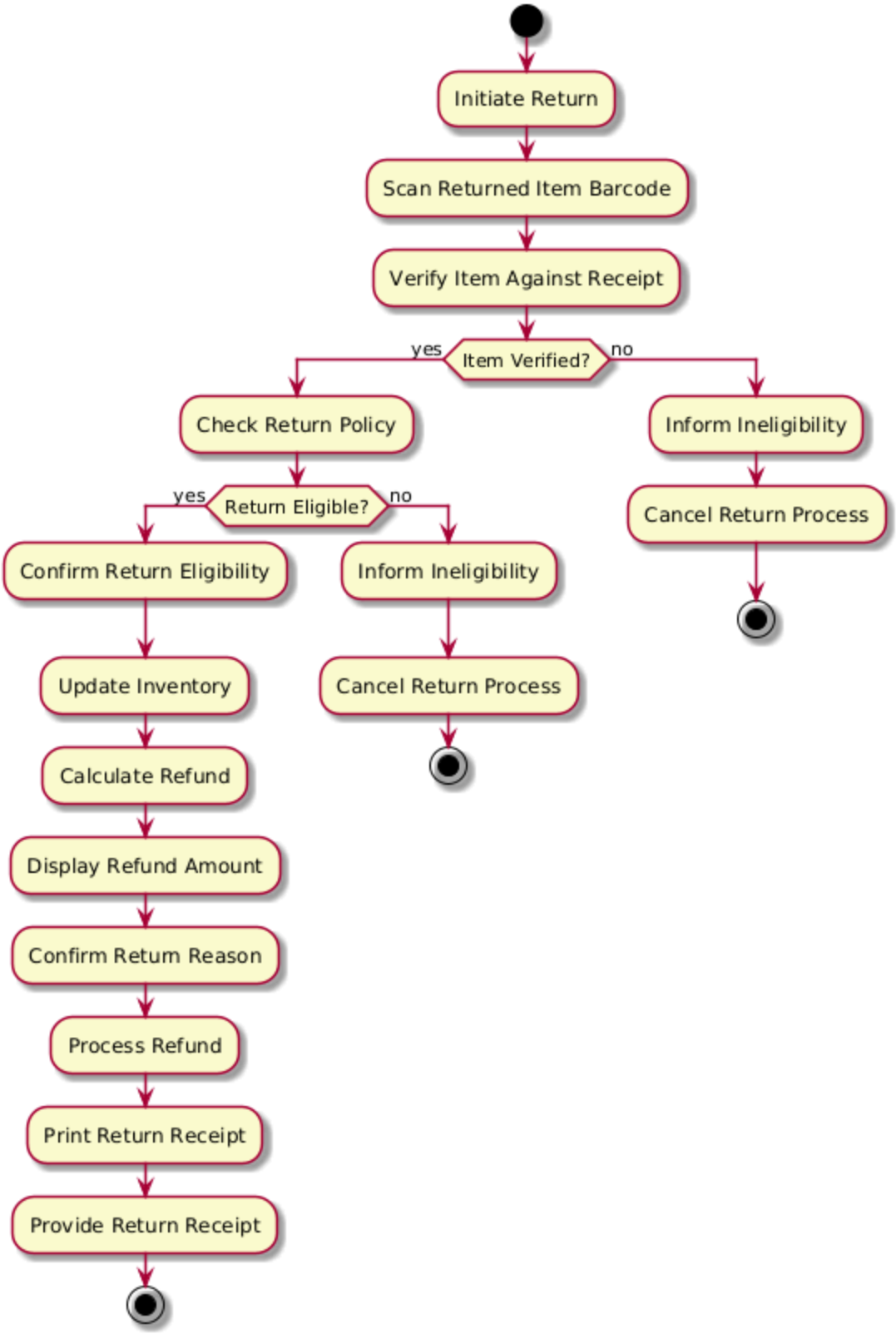
i) Process Sale:



ii) Handle Return



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


Initiate Return
Scan Returned Item Barcode
Verify Item Against Receipt
Item Verified?
yes
no
Check Return Policy
Inform Ineligibility
Cancel Return Process
Return Eligible?
yes
no
Confirm Return Eligibility
Inform Ineligibility
Update Inventory
Cancel Return Process
Calculate Refund
Display Refund Amount
Confirm Return Reason
Process Refund
Print Return Receipt
Provide Return Receipt

ii) Handle Return