Dhirubhai Ambani Institute of Information and Communication Technology

Lab :- 6
IT 314 :- Software Engineering
Name :- Aryan Solanki
ID :- 202201239

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

Use Case :- Process Sale

Primary Actor

- Cashier

Stakeholders and Interests

- Customer: Wants to purchase goods quickly and accurately
- Store: Wants to sell goods, update inventory, and receive payment
- Manager: Wants accurate sales records

Preconditions

- POS system is operational
- Cashier is logged in

Main Flow

1. Customer arrives at POS with goods to purchase
2. Cashier starts a new sale transaction
3. Cashier scans the barcode of each item
4. System retrieves item information and price from catalog
5. System updates inventory count
6. System calculates total price
7. Cashier informs customer of total price
8. Customer chooses payment method(Cash,Credit Card or Check)
9. If the customer has the coupon:
    a. Cashier apply the coupon to transaction
    b. POS system recalculate the total price
10. Cashier processes payment
11. System records sale
12. System prints receipt
13. Cashier gives receipt and goods to customer

Alternate Flow

3a. Barcode is unreadable

- Cashier manually enters item code 4a. Item not found in catalog
- Cashier notifies customer and removes item from sale 8a. Customer wants to use a gift coupon
- Cashier applies coupon to sale
- System recalculates total price 10a. Payment is declined

- Cashier asks for alternative payment method
- Use case resumes at step 8

Postconditions

- Sale is recorded
- Inventory is updated
- Payment is processed
- Receipt is printed

Use Case :- Handle Return

Primary Actor

- Cashier

Stakeholders and Interests

- Customer: Wants to return goods and receive refund quickly
- Store: Wants to process returns accurately and update inventory
- Manager: Wants accurate return records

Preconditions

- POS system is operational
- Cashier is logged in
- Customer has goods to return and original receipt

Main Flow

1. Customer arrives at POS with goods to return and original receipt

2. Cashier starts a new return transaction
3. Cashier scans the barcode of each returned item
4. System verifies item against original receipt
5. System calculates refund amount
6. System updates inventory count
7. Cashier confirms return details with customer
8. Cashier processes refund
9. System records return
10. System prints return receipt
11. Cashier gives return receipt and refund to customer

Alternate Flow

3a. Barcode is unreadable

- Cashier manually enters item code 4a. Item not found on original receipt
- Cashier notifies customer that item cannot be returned
- Cashier removes item from return transaction 7a. Customer decides not to return an item
- Cashier removes item from return transaction
- Use case resumes at step 5 8a. Refund method not available
- Cashier offers alternative refund method
- Use case resumes at step 8

Postconditions

- Return is recorded
- Inventory is updated
- Refund is processed

- Return receipt is printed

2. Identify Entity/Boundary Control Objects

Entity Objects:

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

Boundary Objects:

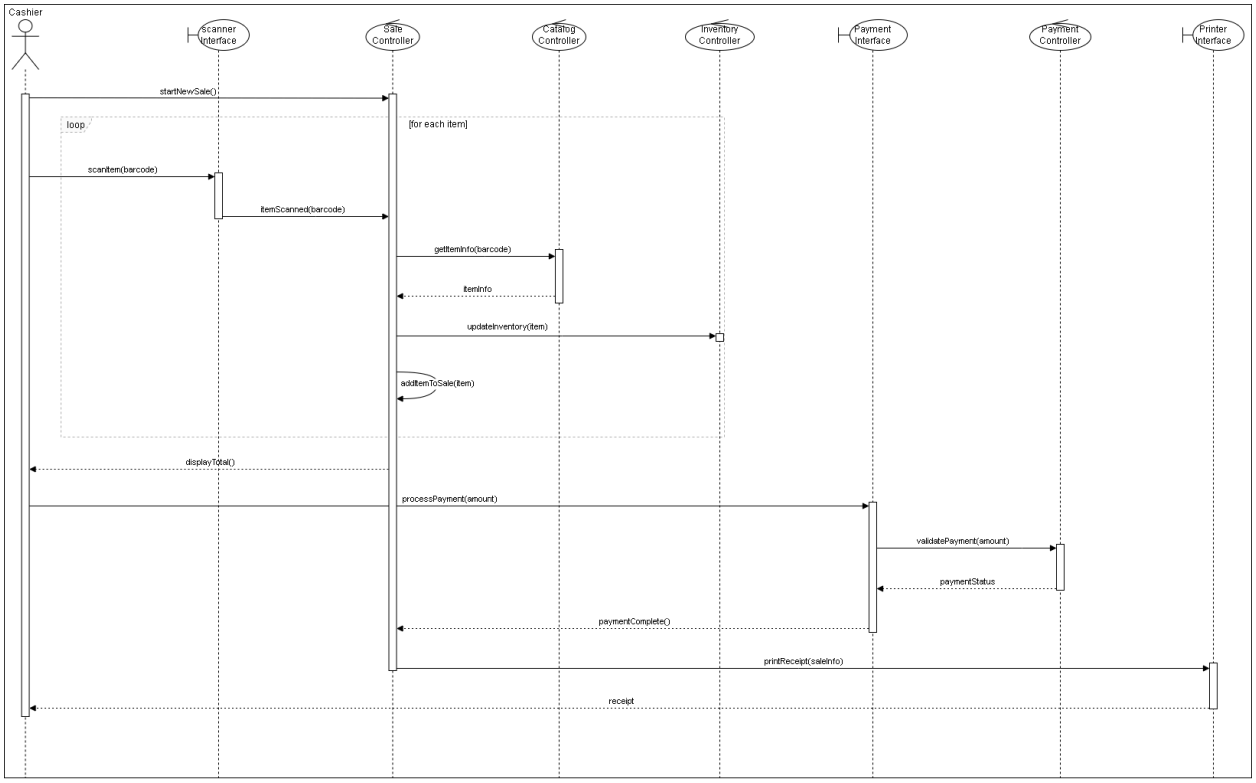
- ScannerInterface
- DisplayInterface
- PrinterInterface
- PaymentInterface
- LoginInterface

Control Objects:

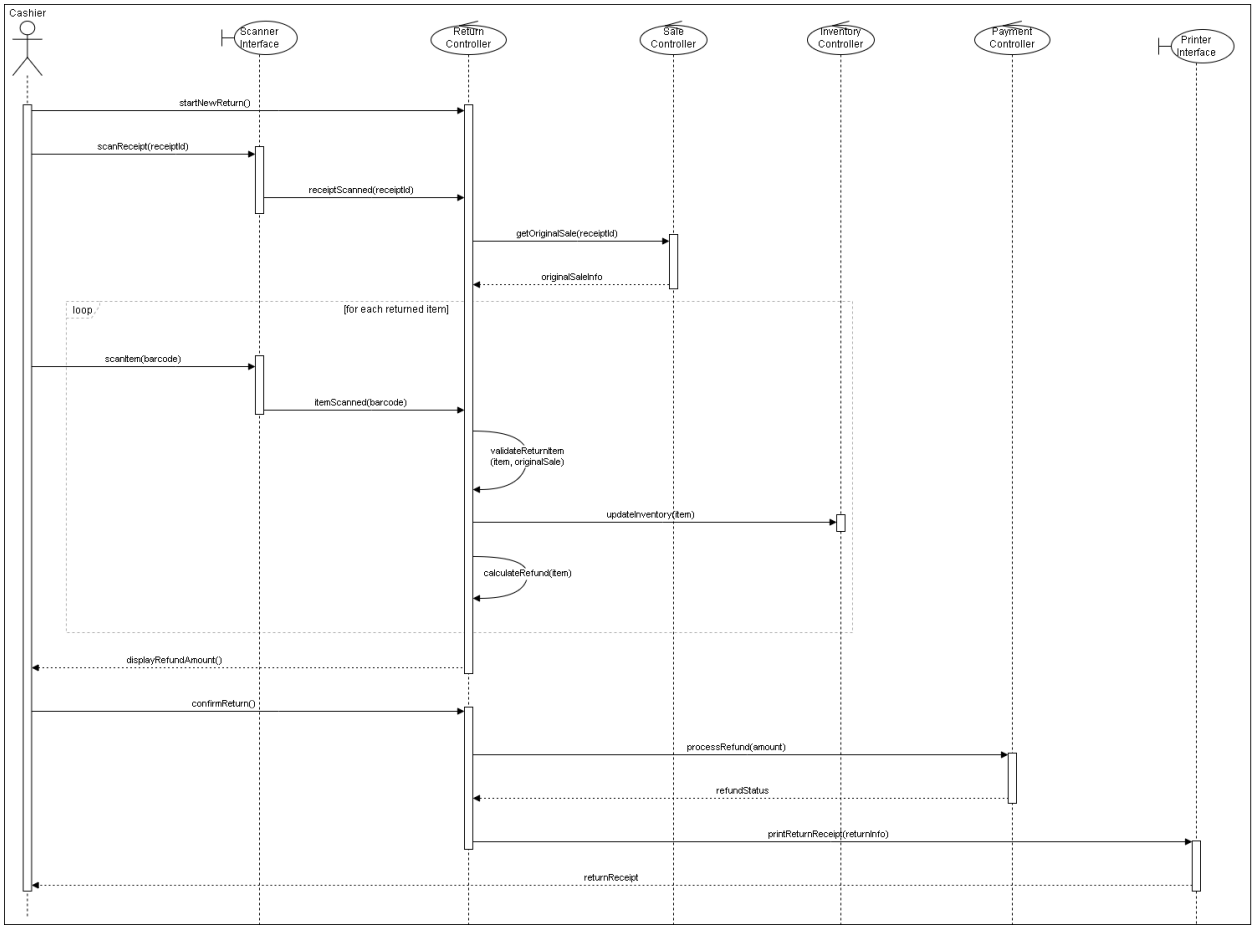
- SaleController
- InventoryController
- CatalogController
- PaymentController
- UserController
- ReturnController

3. Develop Sequence Diagrams

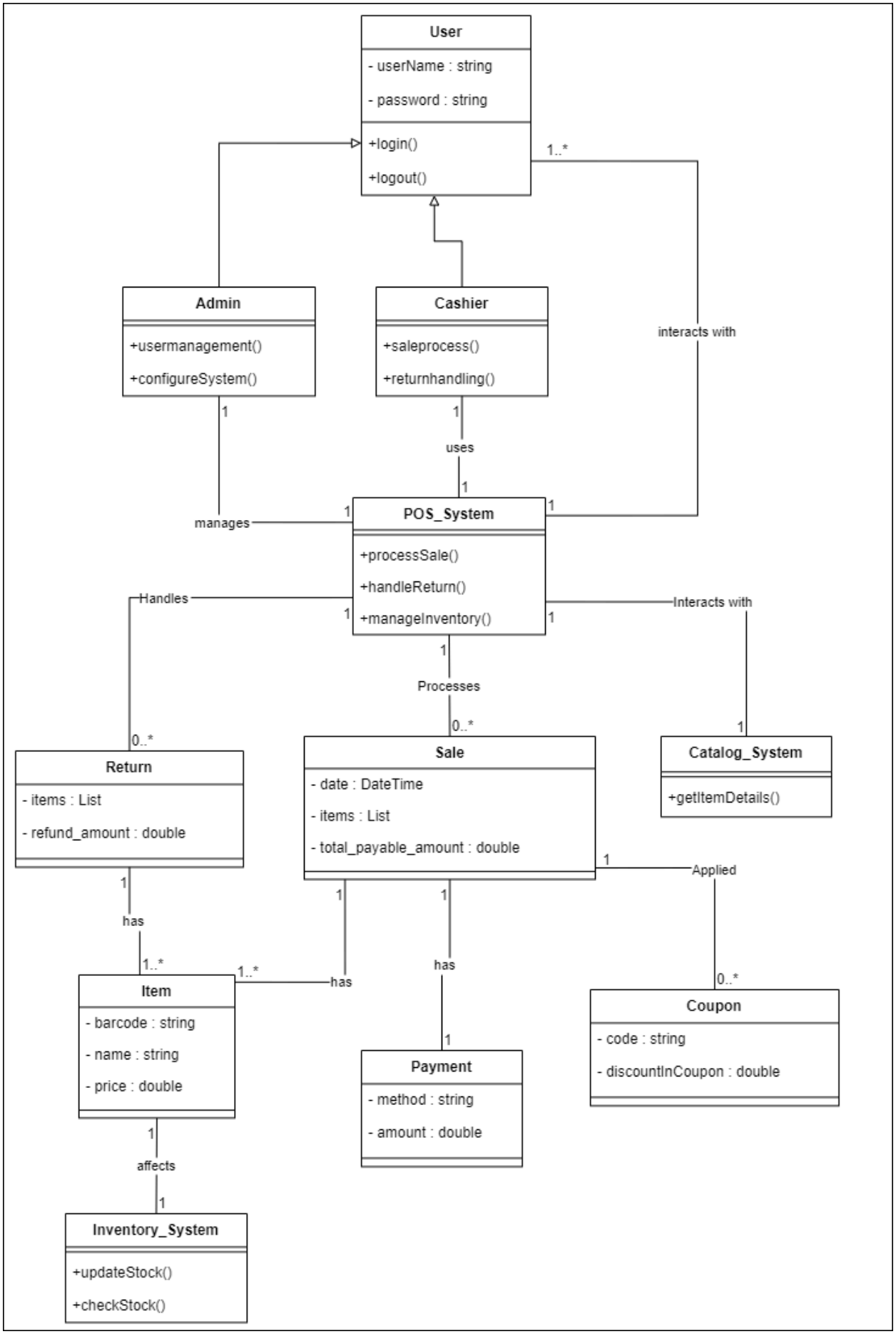
Process Sell:-



Handle Return

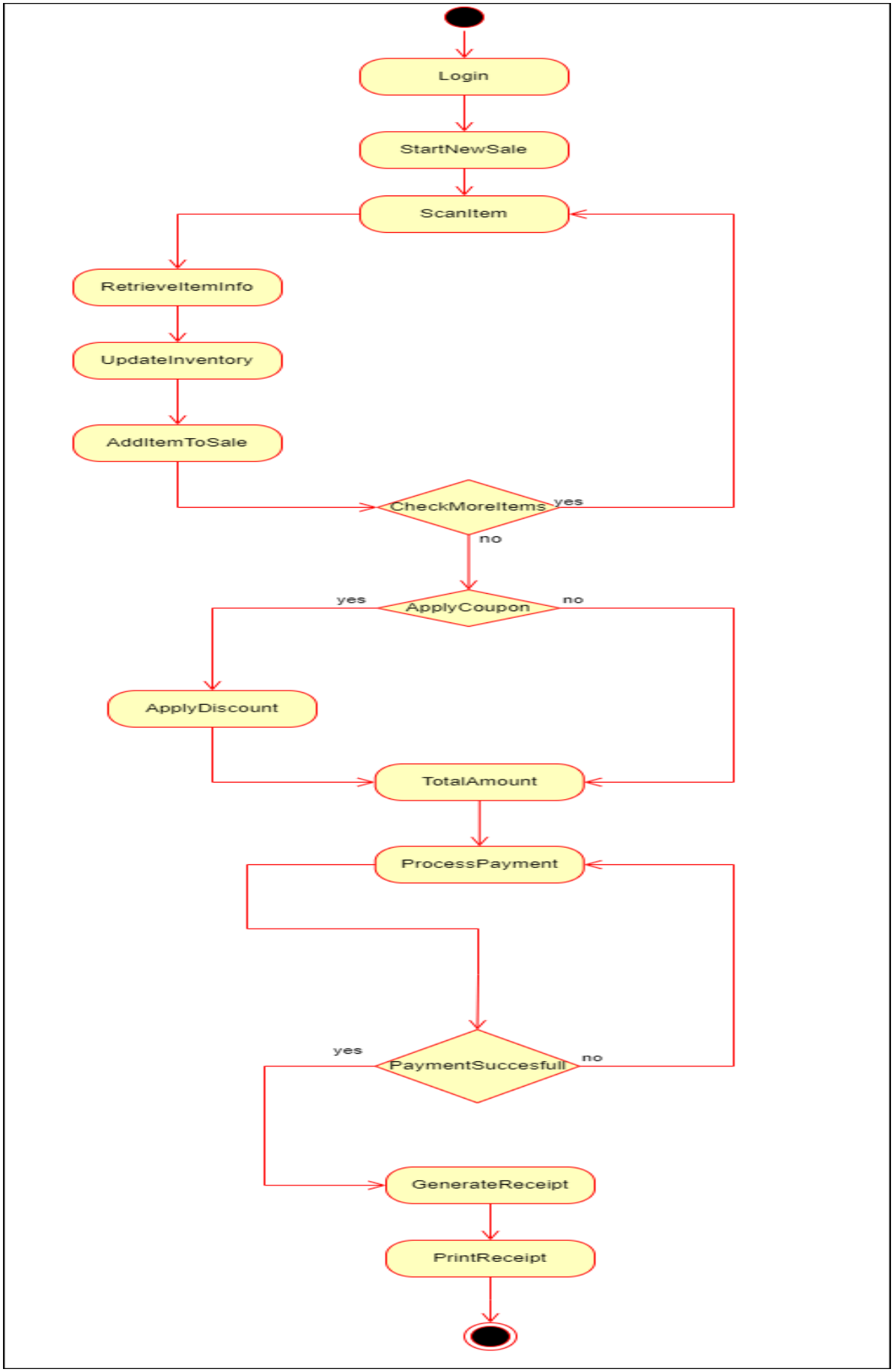


4. Develop Analysis Domain Models


User
- userName : string
- password : string
+login()
+logout()
1..*
Admin
+usermanagement()
+configureSystem()
Cashier
+saleprocess()
+returnhandling()
interacts with
1
1
uses
1
1
manages
1
POS_System
1
+processSale()
+handleReturn()
+manageInventory()
Handles
1
1
Interacts with
1
Processes
0..*
1
0..*
Return
- items : List
- refund_amount : double
Sale
- date : DateTime
- items : List
- total_payable_amount : double
Catalog_System
+getItemDetails()
1
has
1..*
1..*
has
1
1
has
1
Applied
0..*
Item
- barcode : string
- name : string
- price : double
Payment
- method : string
- amount : double
Coupon
- code : string
- discountInCoupon : double
1
affects
1
Inventory_System
+updateStock()
+checkStock()

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

Process Sell:-


Login
StartNewSale
ScanItem
RetrieveItemInfo
UpdateInventory
AddItemToSale
CheckMoreItems
yes
no
yes
ApplyCoupon
no
ApplyDiscount
TotalAmount
ProcessPayment
yes
PaymentSuccesfull
no
GenerateReceipt
PrintReceipt

Handle Return:-