FUNCTIONAL REQUIREMENTS

1. **Order Creation and Management**
   - The system should allow staff to create customer orders by selecting products, quantities, and applying discounts if applicable.
   - Orders should support both in-store purchases and delivery orders.

2. **Billing and Invoice Generation**
   - The system should automatically generate bills with item-wise prices, taxes, and final payable amount.
   - After successful payment, a finalized invoice and receipt should be generated.

3. **Payment Processing**
   - The system should support multiple payment options such as cash, UPI, cards and wallets.
   - It should record payment status (successful, failed, pending) for each order.

4. **Inventory Monitoring and Updates**
   - The system should maintain real-time stock levels for each inventory item.
   - Stock should automatically reduce after a sale and increase after goods receipt.

5. **Reorder and Supplier Management**
   - The system should notify when stock falls below the reorder level.
   - It should support creation and tracking of purchase orders to suppliers.

6. **Delivery Order Processing**
   - The system should allow marking orders for home delivery.
   - Delivery status should be updated from dispatch to completion.

NON-FUNCTIONAL REQUIREMENTS

1. **Usability**
   - The system should be simple and easy to use by shop staff with minimal technical training.
   - Common operations like billing and stock updates should require minimal steps.

2. **Performance**
   - Order creation, billing, and payment confirmation should be processed with minimal delay, especially during peak hours.
   - Inventory updates should reflect immediately after a transaction.

3. **Reliability**
   - The system should not lose order, payment, or inventory data in case of failures.
   - Transactions should be consistent and recoverable.

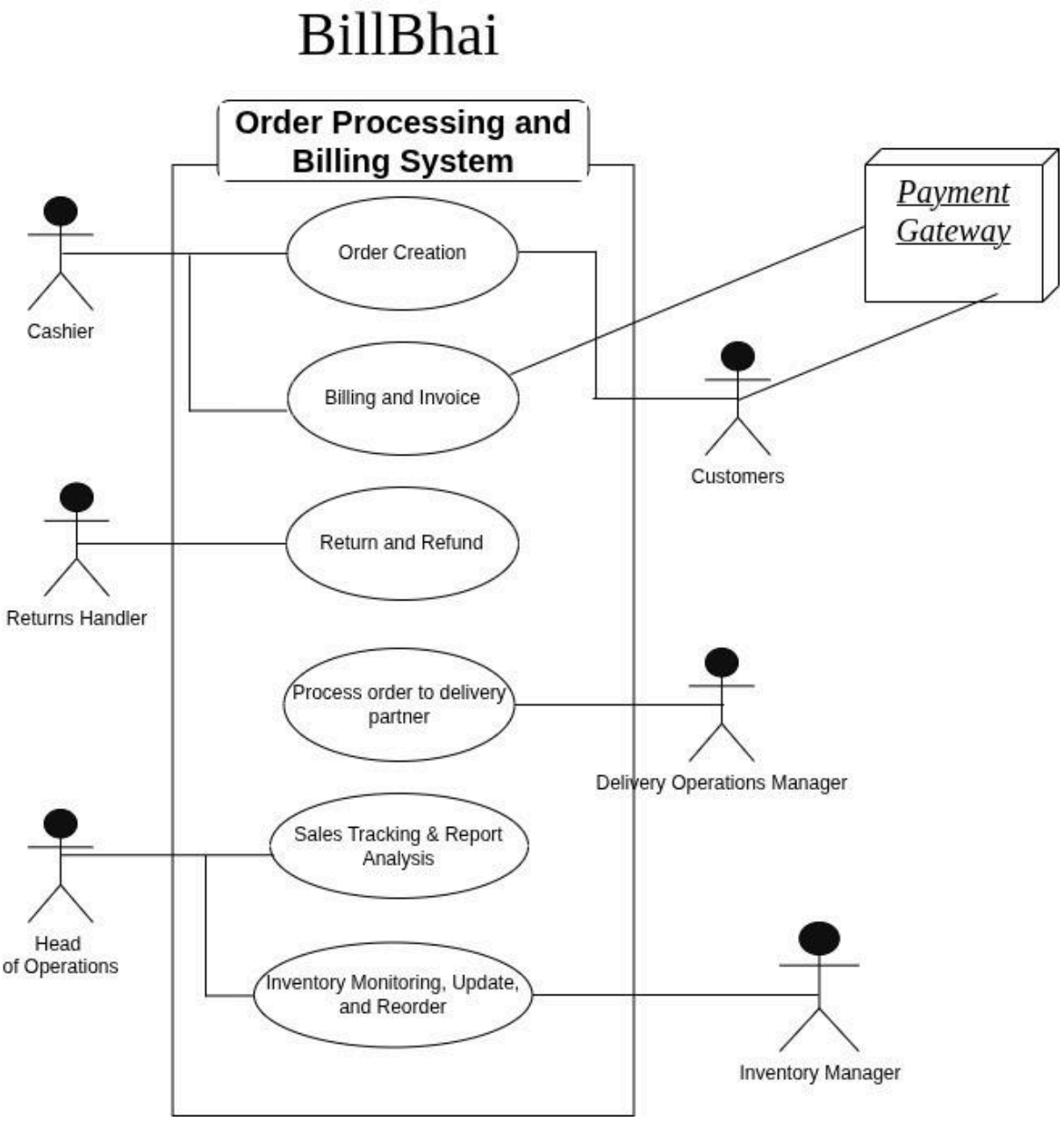
4. **Scalability**
   - The system should handle increasing product range, customers, and transactions as the business grows.
   - It should support future expansion to multiple branches or warehouses.

5. **Security**
   - Sensitive data such as payment details and invoices should be protected from unauthorized access.
   - Only authorized users should be allowed to perform critical operations like stock adjustment.

6. **Auditability and Traceability**
   - The system should maintain logs for key actions such as stock adjustments, order cancellations, and refunds.
   - This supports accountability and issue resolution.

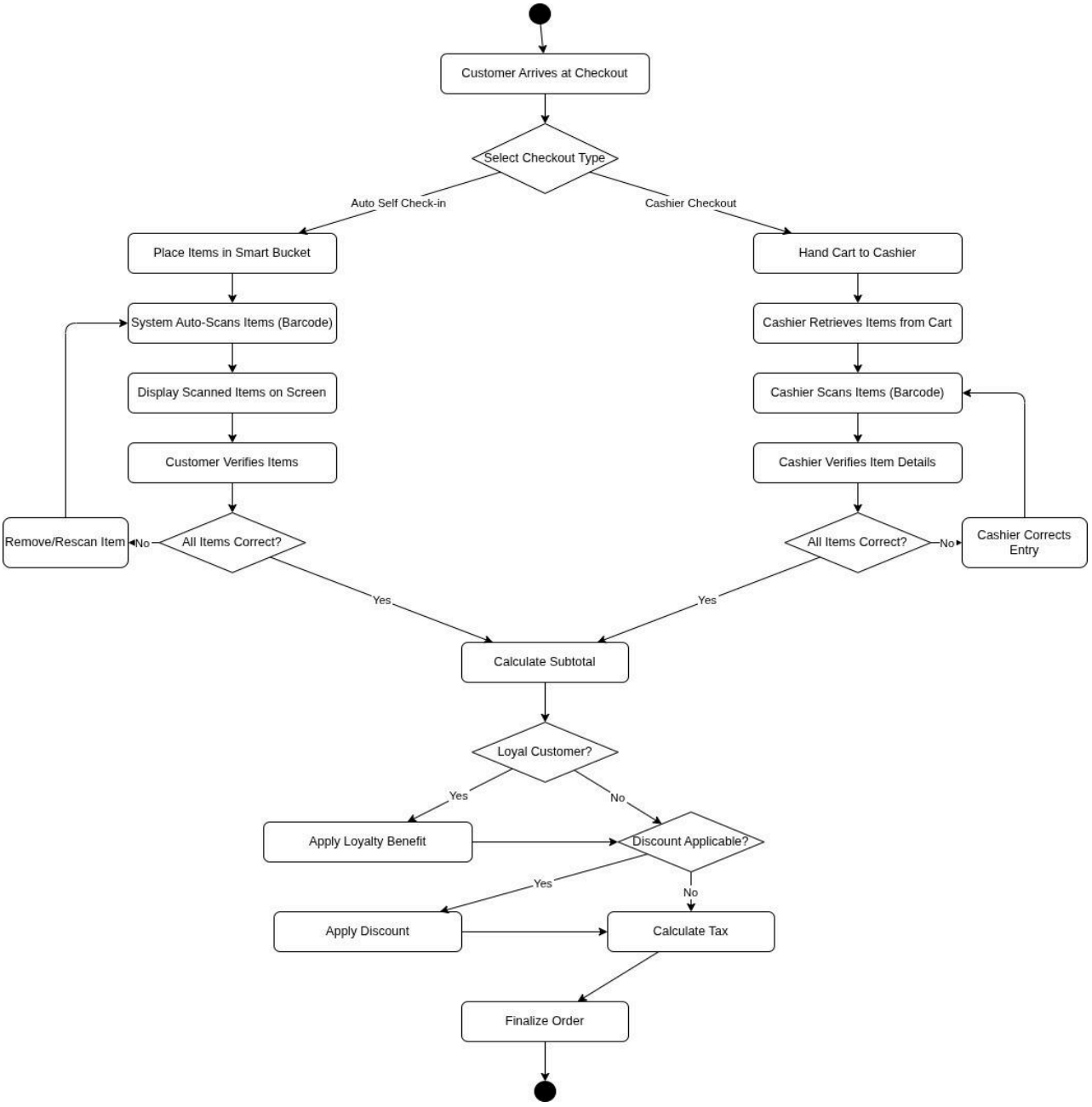
UML Diagrams

1. Use Case Diagram

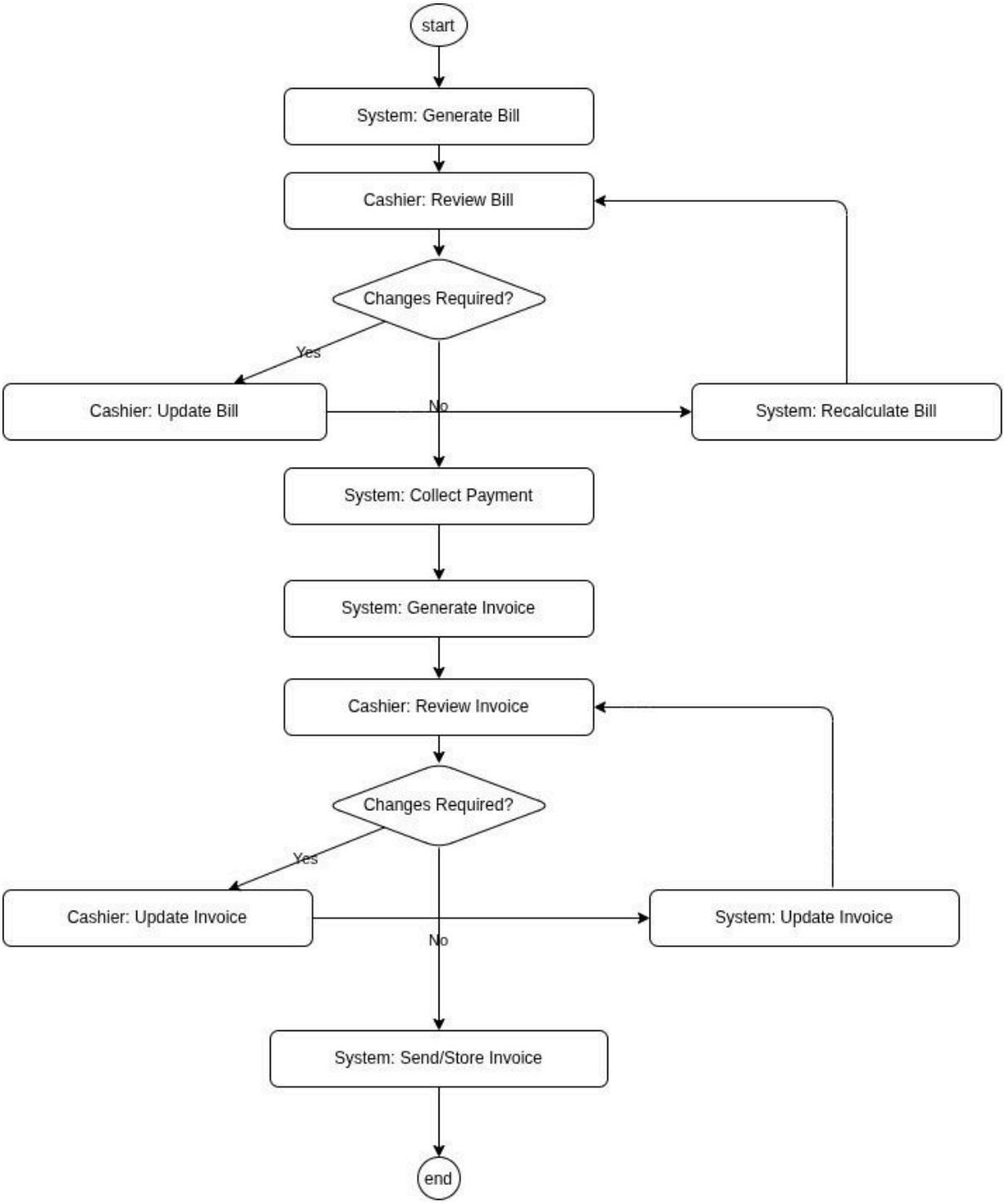


2. **Activity Diagrams**

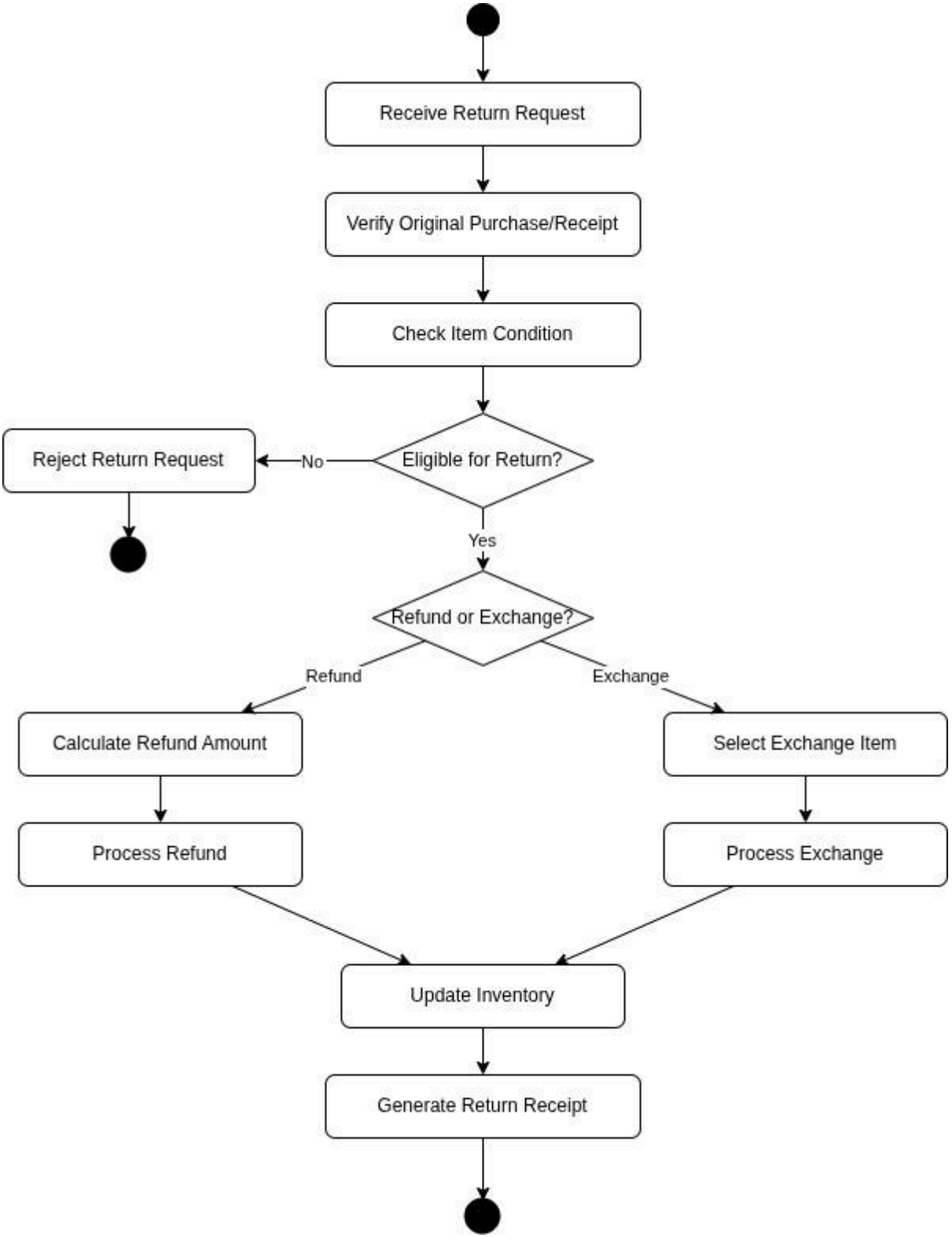
i. Order Creation



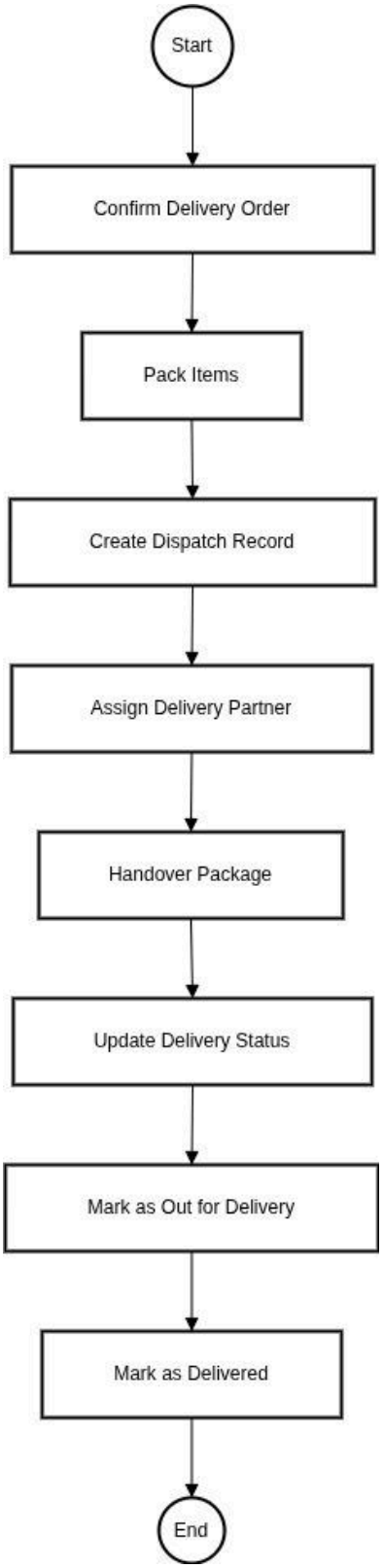
ii. Billing and Invoice



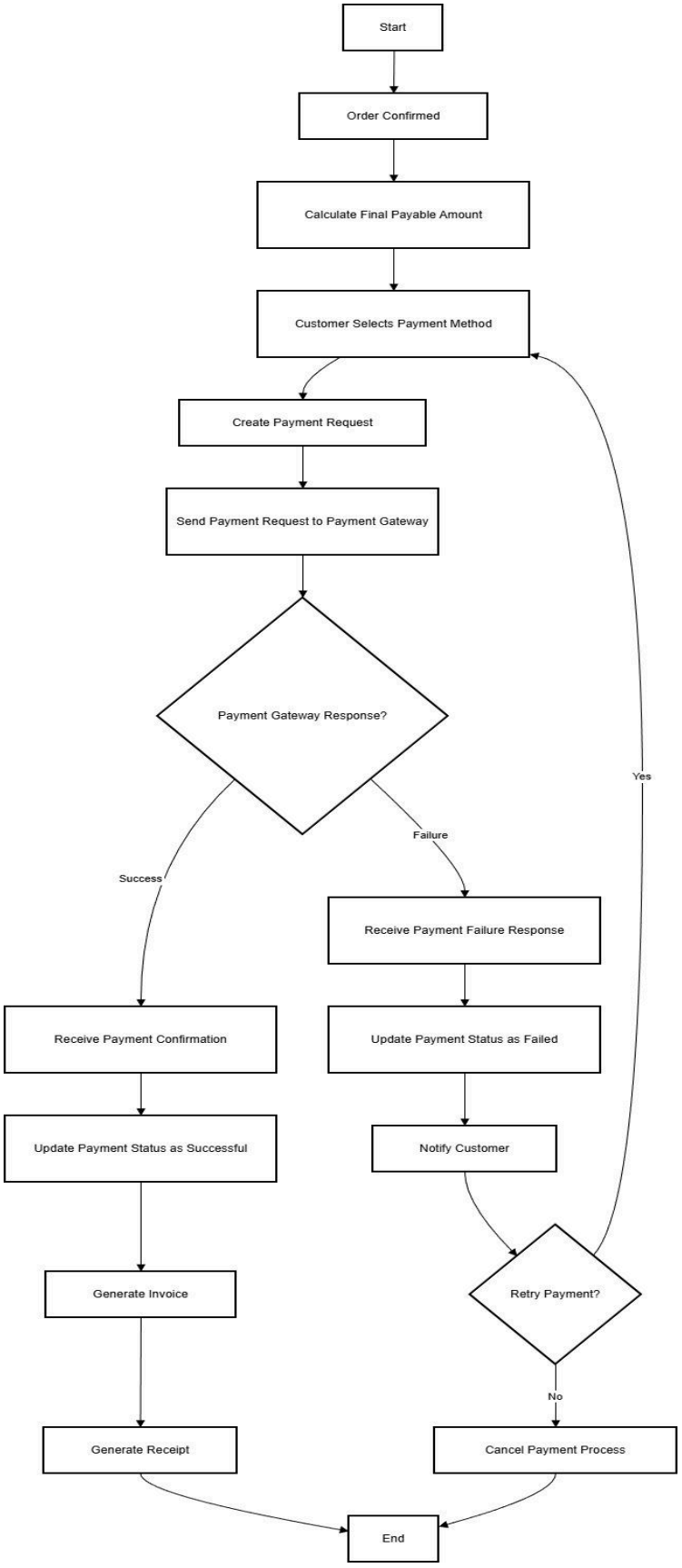
iii. Return Item Processing



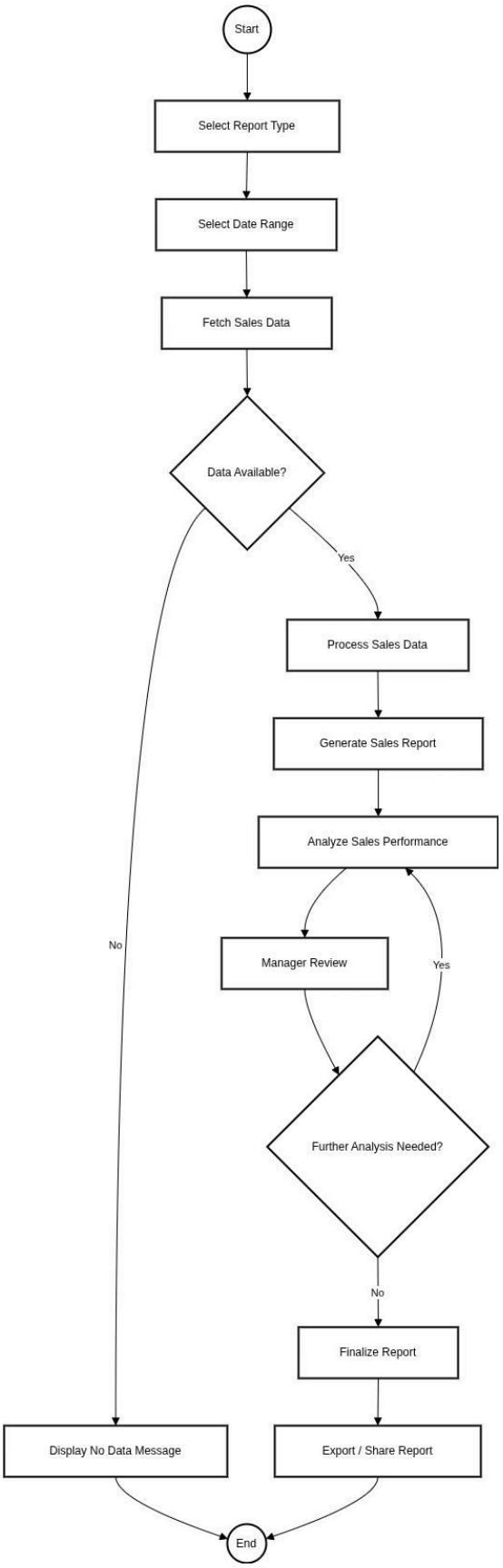
iv. Delivery Partner



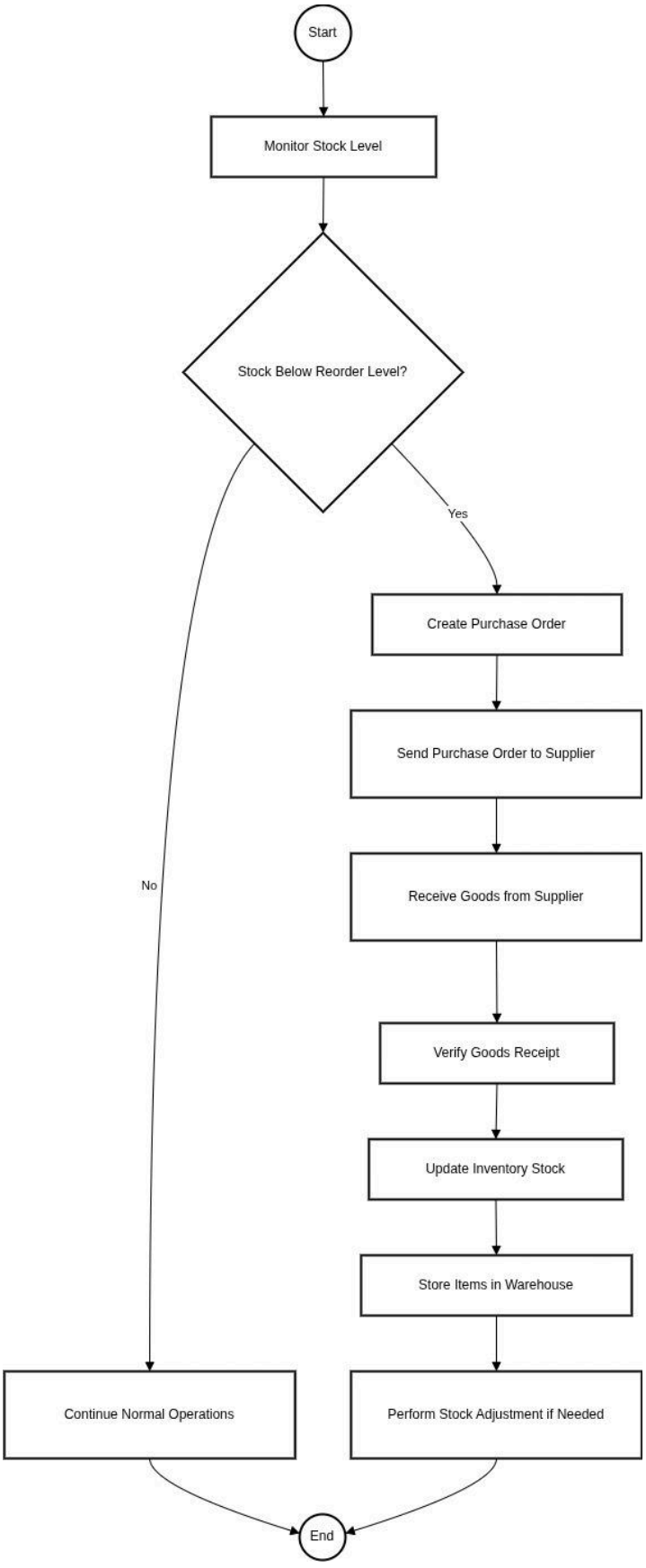
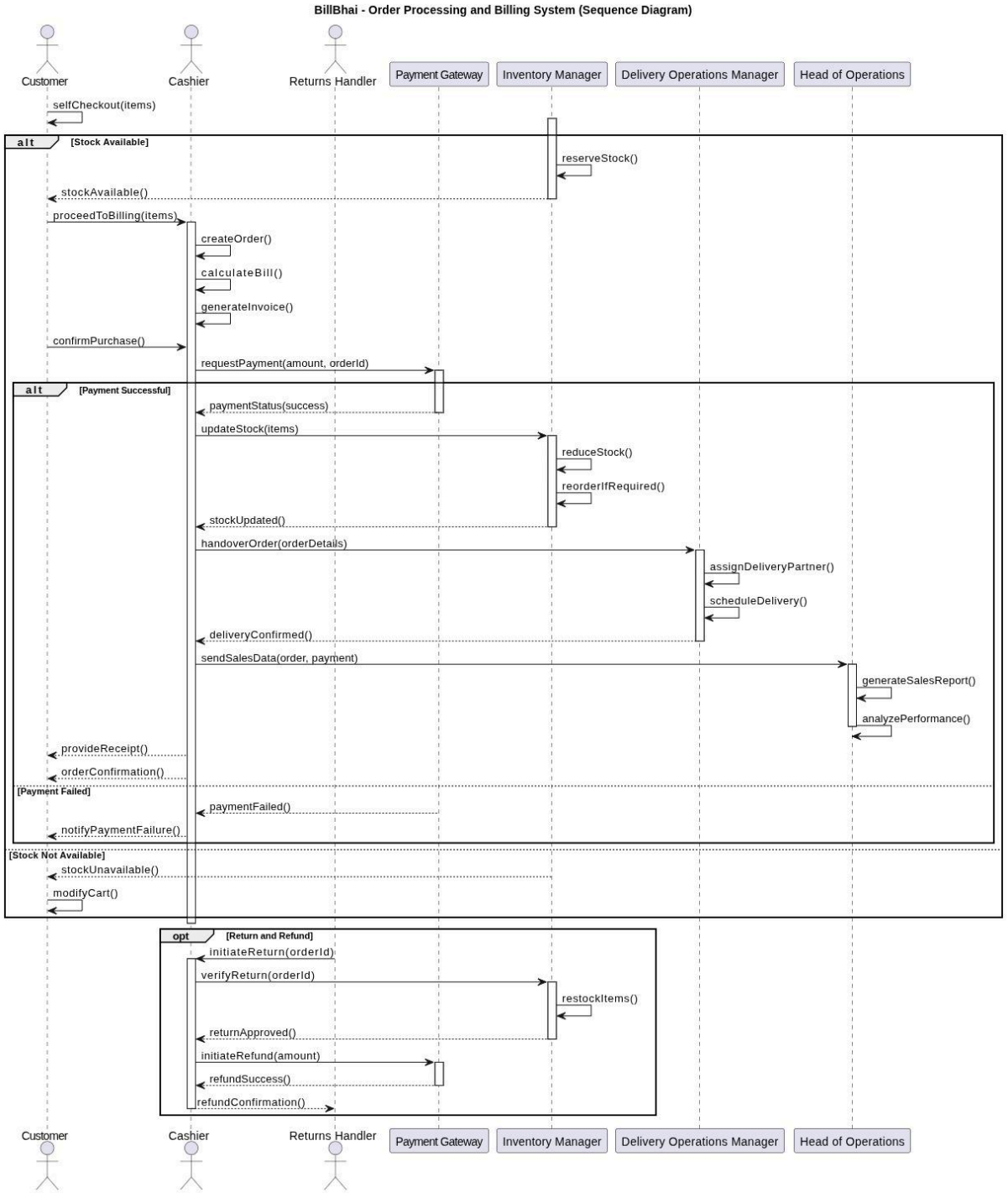
v. Payment Gateway



vi. Sales T & Report A



vii. Inventory Monitoring



3. **Sequence Diagram**