

## Domain Classes; Noun Technique

### Step 1 — Identify candidate nouns (source: use cases, system vision, technical reports, user stories)

Customer, Order, Order ID, Product, Product Description, Price, Inventory, InventoryItem, Warehouse, Stock Level, Replenishment Request, Supplier, Payment, Payment Authorization, Payment Gateway, Invoice, Shipment, Tracking Number, Courier, Order Status, Return Request, Refund, Customer Support Agent, Staff, SalesStaff, FinanceStaff, InventoryStaff, LogisticsStaff, SupportAgent, Report, KPI, Address, Delivery Details, OrderItem, Quantity, Transaction, Notification (system output).

---

### Step 2 — Refine and classify nouns (Include / Exclude / Research)

Use the textbook decision questions: is it a unique thing the system needs to remember; is it in scope; does the system need to store multiple instances; is it merely an output or attribute?

Noun	Decision	Short rationale (trace to project file)
Customer	Include	Appears as primary actor for orders/returns.
Order	Include	Core transaction; generated Order ID on placement.
OrderItem	Include	Represents items within Order (quantity, price).
Product	Include	Managed in catalog; used in verification and inventory.
InventoryItem	Include	Tracks stock levels and reservations.
Warehouse	Include	Location for stored inventory.
binLocation	Exclude (attribute)	No bin management use case; keep as attribute if needed.
Supplier	Include	Receives replenishment requests.
ReplenishmentRequest	Include	Triggered when stock low.
Payment	Include	Payment processing and authorization flows present.
Payment Gateway	Exclude (external actor)	External system, not a stored domain class.
Invoice	Include	Generated after payment; appears in scope.
Shipment	Include	Logistics plans shipments and assigns tracking.
Courier	Include	External delivery partner referenced in shipment flow.
ReturnRequest	Include	Customer returns/refunds workflow (US9).
Refund	Exclude (process/attribute)	Handled within ReturnRequest or Payment status.
Staff (abstract)	Include	Multiple staff roles appear as actors.
SalesStaff / FinanceStaff / InventoryStaff / LogisticsStaff / SupportAgent	Include	Domain role specializations (actors).
Noun	Decision	Short rationale (trace to project file)

Report	<b>Include</b>	Management reporting use case (US13).
Notification (Email/SMS)	<b>Exclude (system output)</b>	Output/event, not persistent domain class.
ProductCatalog	<b>Exclude</b>	Catalog referenced conceptually; project stores Product instances only.

---

### Step 3 — Produce final domain class list

Customer; Order; OrderItem; Product; InventoryItem; Warehouse; Supplier; ReplenishmentRequest; Payment; Invoice; Shipment; Courier; ReturnRequest; Staff (abstract) and subclasses (SalesStaff, FinanceStaff, InventoryStaff, LogisticsStaff, SupportAgent); Report.

---

### Step 4 — For each included class: candidate key attributes

Domain class	Key attributes (candidates)
Customer	customerId {key}, fullName, email, phone, defaultAddress
Order	orderId {key}, orderDate, status, totalAmount, customerId (FK)
OrderItem	orderItemId {key}, orderId (FK), productId (FK), quantity, unitPrice, lineTotal
Product	productId {key}, sku, name, description, price, category
InventoryItem	inventoryId {key}, productId (FK), warehouseId (FK), quantityOnHand, reorderPoint, status
Warehouse	warehouseId {key}, name, location
Supplier	supplierId {key}, name, contactInfo
ReplenishmentRequest	requestId {key}, productId (FK), supplierId (FK), quantity, requestDate, status
Payment	paymentId {key}, orderId (FK), amount, method, authorizationCode, status, paymentDate
Invoice	invoiceId {key}, orderId (FK), invoiceDate, total, tax, shippingCharges
Shipment	shipmentId {key}, orderId (FK), courierId (FK), trackingNumber, shipDate, estimatedDelivery
Courier	courierId {key}, name, contactInfo, apiEndpoint
ReturnRequest	returnId {key}, orderId (FK), reason, requestDate, status
Staff (abstract)	staffId {key}, fullName, role
Report	reportId {key}, type, generatedDate, parameters

---

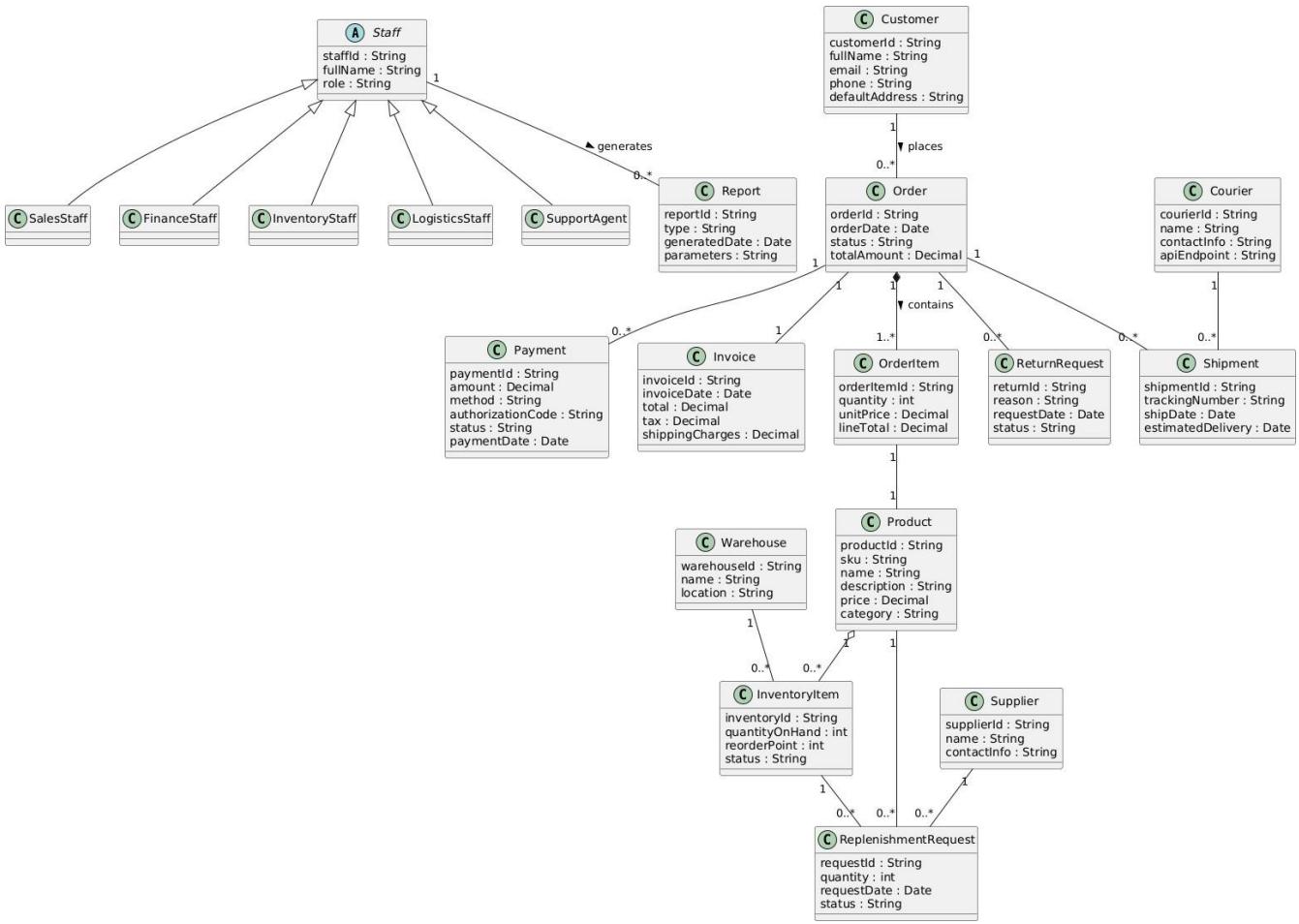
### Step 5 — Preliminary relationships, type, and multiplicity

From	To	Relationship type	Multiplicity (From → To)	Rationale / trace
Customer	Order	association	Customer 1 → Order 0..*	Customer places many orders.
Order	OrderItem	composition	Order 1 — OrderItem 1..*	Order owns its items; items don't exist without order.
From	To	Relationship type	Multiplicity (From → To)	Rationale / trace

OrderItem	Product	association	OrderItem * → Product 1	Each order line refers to a product.
Product	InventoryItem	aggregation	Product 1 o-- 0..* InventoryItem	Inventory entries reference products.
Warehouse	InventoryItem	association	Warehouse 1 → InventoryItem 0..*	Warehouse stores inventory records.
InventoryItem	ReplenishmentRequest	association	InventoryItem 1 → ReplenishmentRequest 0..*	Low stock triggers replenishment.
ReplenishmentRequest	Supplier	association	ReplenishmentRequest * → Supplier 1	Requests are sent to supplier.
Order	Payment	association	Order 1 → Payment 0..*	Multiple payment attempts/records possible.
Order	Invoice	association	Order 1 → Invoice 1..1	Invoice generated per order (adjust if split invoicing used).
Order	Shipment	association	Order 1 → Shipment 0..*	Allow split shipments.
Shipment	Courier	association	Shipment * → Courier 1	Courier assigned to each shipment.
Order	ReturnRequest	association	Order 1 → ReturnRequest 0..*	Returns reference orders.
Staff	Report	association	Staff 0..* → Report 0..*	Staff generate or request reports.
Staff (abstract)	SalesStaff, FinanceStaff, InventoryStaff, LogisticsStaff, SupportAgent	generalization	—	Role specializations for actor mapping.

---

## Step 6 — Class Diagram



## CRUD Technique

### CRUD Technique Applied to Final Use Cases

#### 1. Final Validated Use Case List

After applying the CRUD Technique and removing non-actor-goal use cases, the final set of valid use cases is:

1. Place Online Order
2. Process Customer Payment
3. Manage Product Catalog
4. Plan Shipment
5. Track Order Status
6. Initiate Return Request
7. Look Up Order History
8. Generate Management Reports
9. Manage User Permissions

#### 2. Use Cases Removed or Merged

##### Merged Use Cases

The following use cases were **merged** because they do not represent independent goals of any actor. Instead, they are **internal system responsibilities** that logically fall under larger, actor-driven use cases.

Removed / Merged Use Case	Merged Into	Reason for Merge
Handle Payment Authorization	Process Customer Payment	Authorization is part of the internal payment workflow (communicating with the payment gateway, confirming funds). No actor starts this step manually.
Verify Order Details	Place Online Order	The system automatically checks product availability, addresses, totals, and customer data during order placement. Customers do not trigger a separate "verify" action.
Reserve Inventory	Manage Product Catalog <i>(inventory logic)</i> + Place Online Order / Process Payment <i>(trigger)</i>	Inventory reservation happens automatically when an order is placed/paid. It is not a user request and does not appear as a standalone goal.
Send Replenishment Request	Manage Product Catalog	Replenishment is automatic when stock is low. No staff explicitly triggers "send supplier request" as a separate use case.
Transmit Shipment Manifest	Plan Shipment	Sending the shipment manifest to the courier system is part of the shipment planning workflow. No actor initiates this step independently.

## Removed Use Cases (Invalid Actor-Goal Use Cases)

The following were removed because they do **not** represent goals initiated by a human actor. They are **system-maintenance or internal process steps**, not true use cases:

- ◆ Verify Order Details
- ◆ Reserve Inventory
- ◆ Send Replenishment Request
- ◆ Transmit Shipment Manifest
- ◆ Handle Payment Authorization

All remaining use cases are valid actor-goal interactions.

## 3. CRUD Analysis by Data Entity

The CRUD validation for each major domain entity.

### 3.1 Entity: Order

CRUD Operation	Related Use Cases
Create	Place Online Order
Read	Track Order Status, Look Up Order History, Generate Management Reports
Update	Place Online Order, Process Customer Payment, Plan Shipment
Delete/Archive	Initiate Return Request

### 3.2 Entity: Payment

CRUD Operation	Related Use Cases
Create	Process Customer Payment
Read	Look Up Order History, Generate Management Reports
Update	Process Customer Payment (status update)
Delete/Archive	Initiate Return Request (refund resolution)

---

### 3.3 Entity: Inventory Item

CRUD Operation	Related Use Cases
Create	Manage Product Catalog
Read	Place Online Order, Generate Management Reports
Update	Manage Product Catalog (edit), Place Online Order / Process Payment (stock reduction)
Delete/Archive	Manage Product Catalog

---

### 3.4 Entity: Shipment

CRUD Operation	Related Use Cases
Create	Plan Shipment
Read	Track Order Status
Update	Plan Shipment (manifest, courier, routing updates)
Delete/Archive	Not Applicable

---

### 3.5 Entity: User / Staff Permissions

CRUD Operation	Related Use Cases
Create	Manage User Permissions
Read	Manage User Permissions
Update	Manage User Permissions
Delete/Archive	Manage User Permissions

---

### 3.6 Entity: Product Catalog

CRUD Operation	Related Use Cases
Create	Manage Product Catalog
Read	Place Online Order, Generate Management Reports
Update	Manage Product Catalog

CRUD Operation	Related Use Cases
Delete/Archive	Manage Product Catalog

---

## Use Cases Event Decomposition Technique

**Table 1:**

Event (Trigger)	Type	Use Case	Actor
Customer submits order	External Event	Place Online Order (US 7)	Customer
Customer submits payment details	External Event	Process Customer Payment (US 4)	Customer, Finance staff
Inventory Staff initiates catalog update	External Event	Manage Product Catalog (US 3)	Inventory Staff
Inventory confirms order is packaged	State Event	Plan Shipment (US 5)	Logistics Staff
Customer requests order status	External Event	Track Order Status (US 8)	Customer
Customer submits return request	External Event	Initiate Return Request (US 9)	Customer
Customer Support Agent needs history	External Event	Look Up Order History (US 6)	Customer Support Agent
Time to generate management reports	Temporal Event	Generate Management Reports (US 13)	Manager / Executive
System Administrator updates permissions	External Event	Manage User Permissions (US 14)	System Administrator

---

**Table 2:**

Use Case	Description
Place Online Order	The customer selects products from the catalog, adds them to the shopping cart, provides delivery details, and the system automatically verifies product availability, pricing, and customer information for accuracy before generating a unique Order ID and preparing the order for payment processing.
Process Customer Payment	The system securely processes the customer's chosen payment method by validating details, confirming authorization, verifying funds, recording the transaction, and updating the order status upon success.
Manage Product Catalog	Authorized Inventory Staff manage the catalog by adding, updating, or deleting products, while the system tracks inventory, reserves stock for orders, and automatically sends replenishment requests to the Supplier System when stock falls below the reorder point.
Plan Shipment	The Logistics Staff finalizes shipment details—including courier selection, delivery routes, and pickup scheduling—and the system automatically sends the shipment manifest with the customer's address and tracking number to the external Courier System for timely, accurate delivery.
Track Order Status	Allows the Customer to access the order tracking interface using their tracking ID to view the current operational status (e.g., Shipped, Delivered) and the estimated delivery date.
Initiate Return Request	Allows the Customer to submit a request for a return or refund via the designated system interface, ensuring the initiation of the returns and refunds workflow.

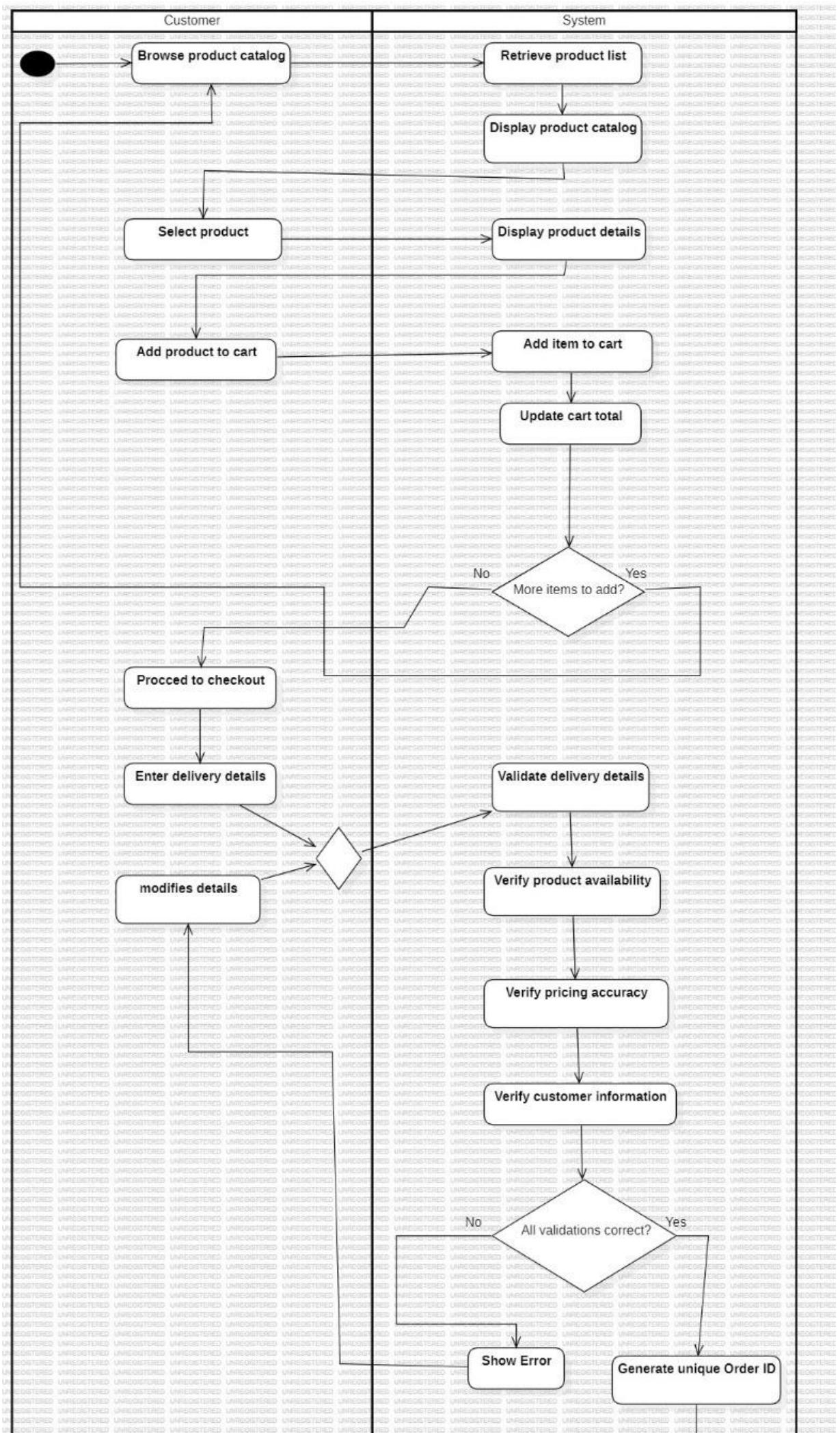
Use Case	Description
<b>Look Up Order History</b>	Allows the Customer Support Agent to search for and access a complete history of a customer's orders, status, and associated return/refund requests to reduce resolution time.
<b>Generate Management Reports</b>	Provides Managers and Executives with integrated analytical and summary reports displaying current data on Key Performance Indicators (KPIs) to aid in business decisions.
<b>Manage User Permissions</b>	Allows the System Administrator to manage user roles and specific access permissions for staff to enforce system security and maintain data integrity.

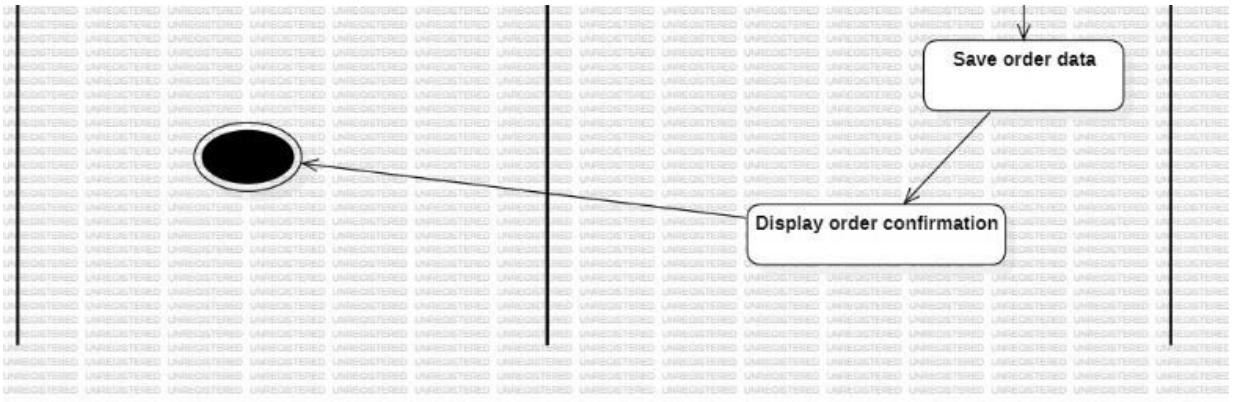
---

## Activity Diagram

### Activity Diagram 1:

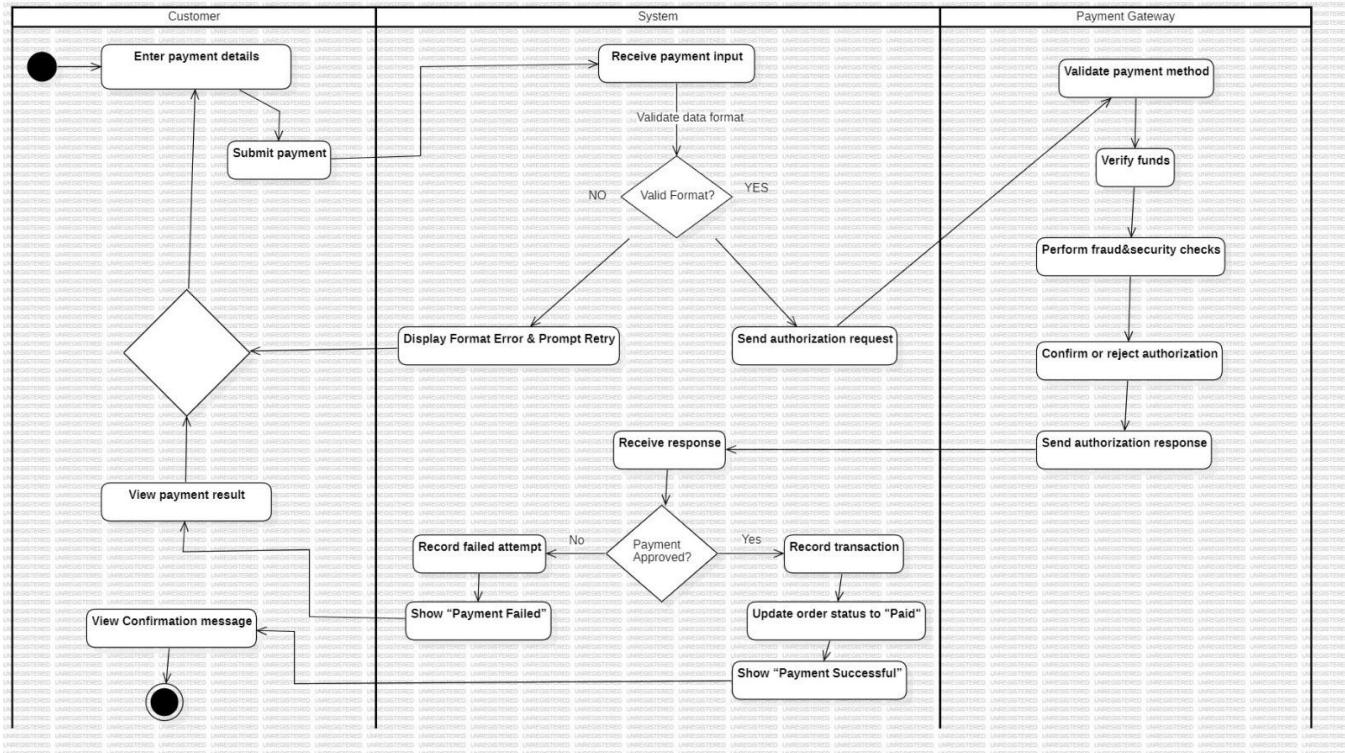
*Place online order*





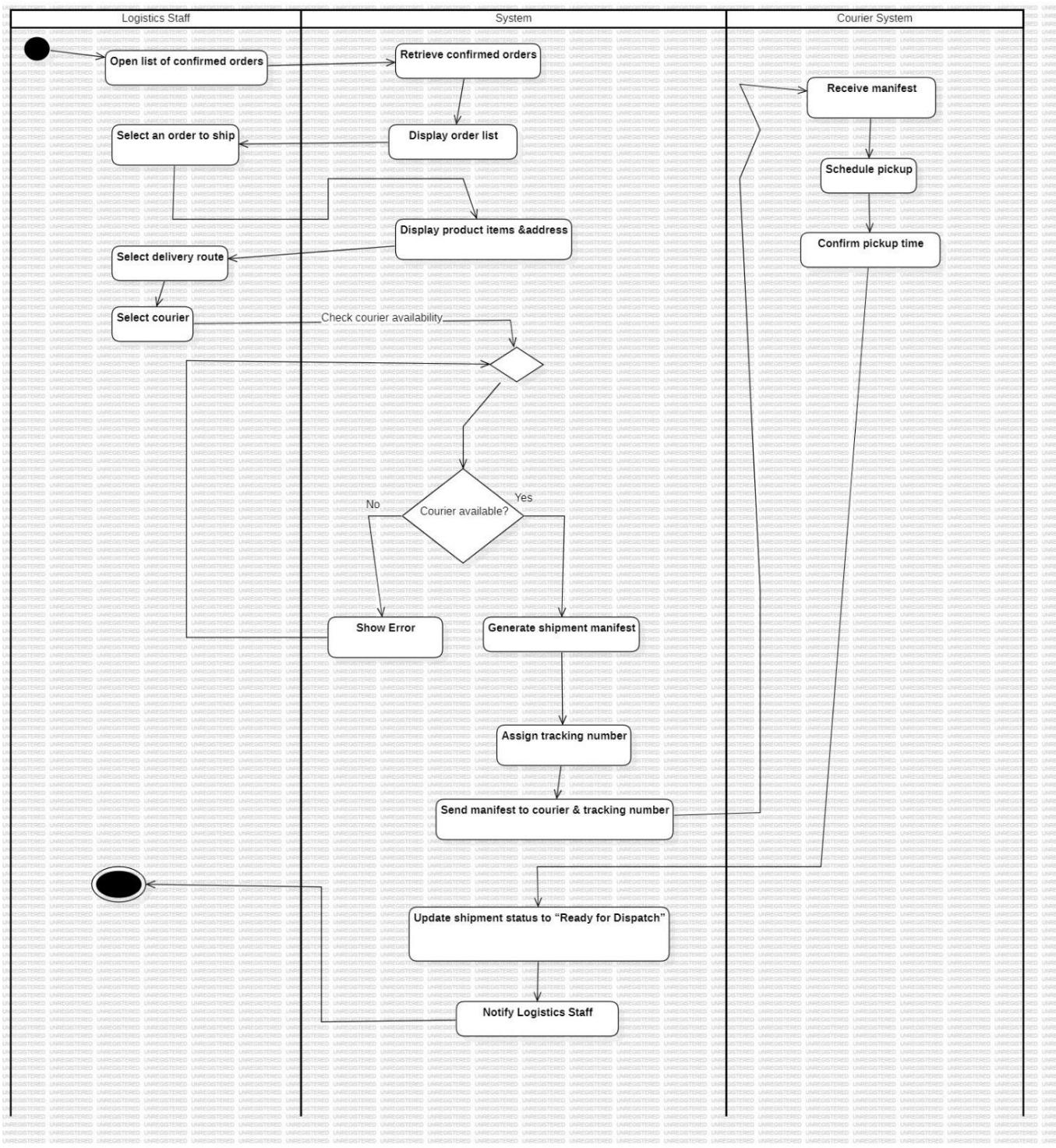
## Activity Diagram 2:

### **Process Customer Payment**



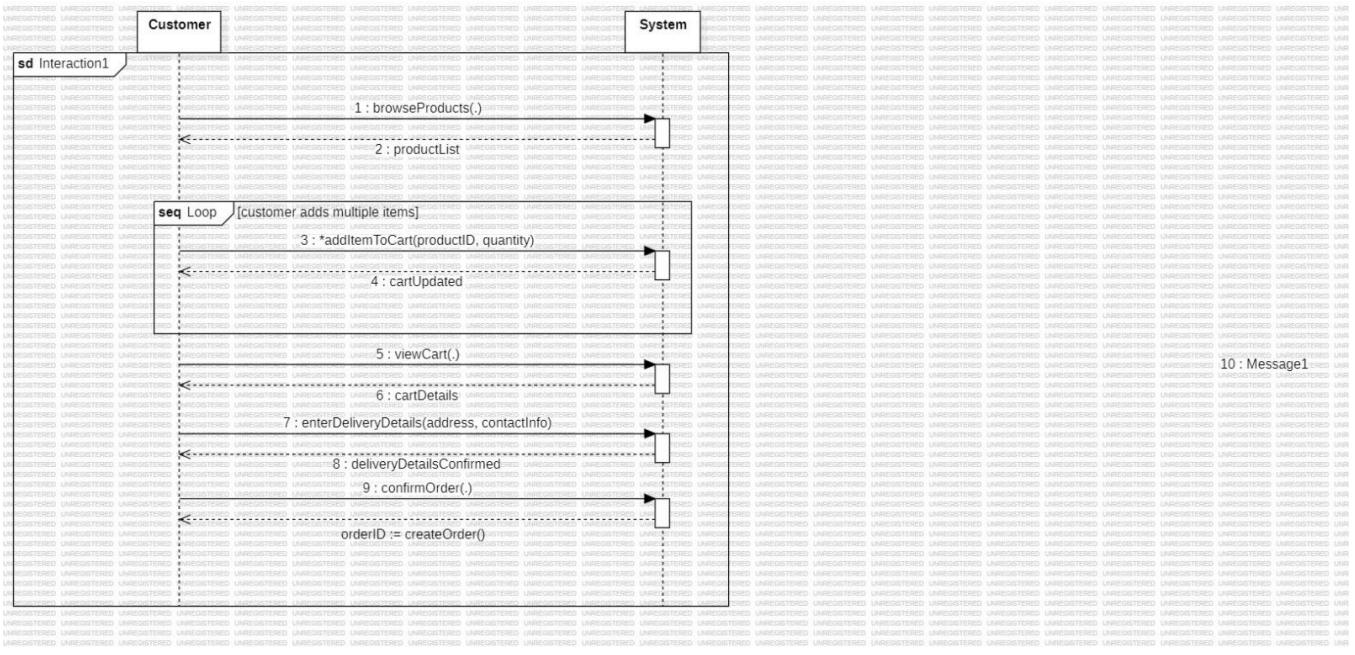
## Activity Diagram 3:

### **Plan Shipment**

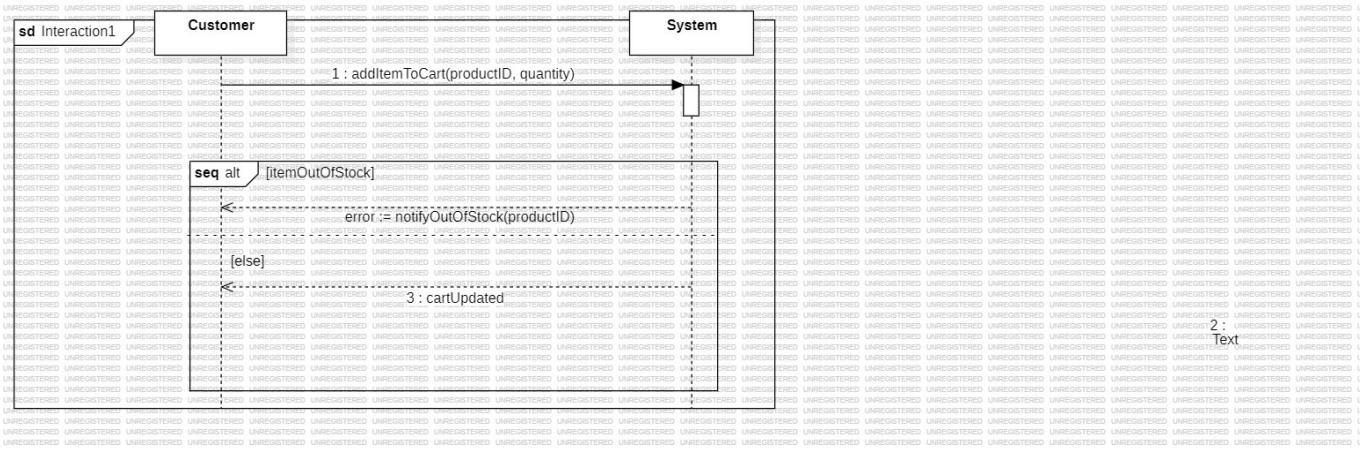


## System Sequence Diagram

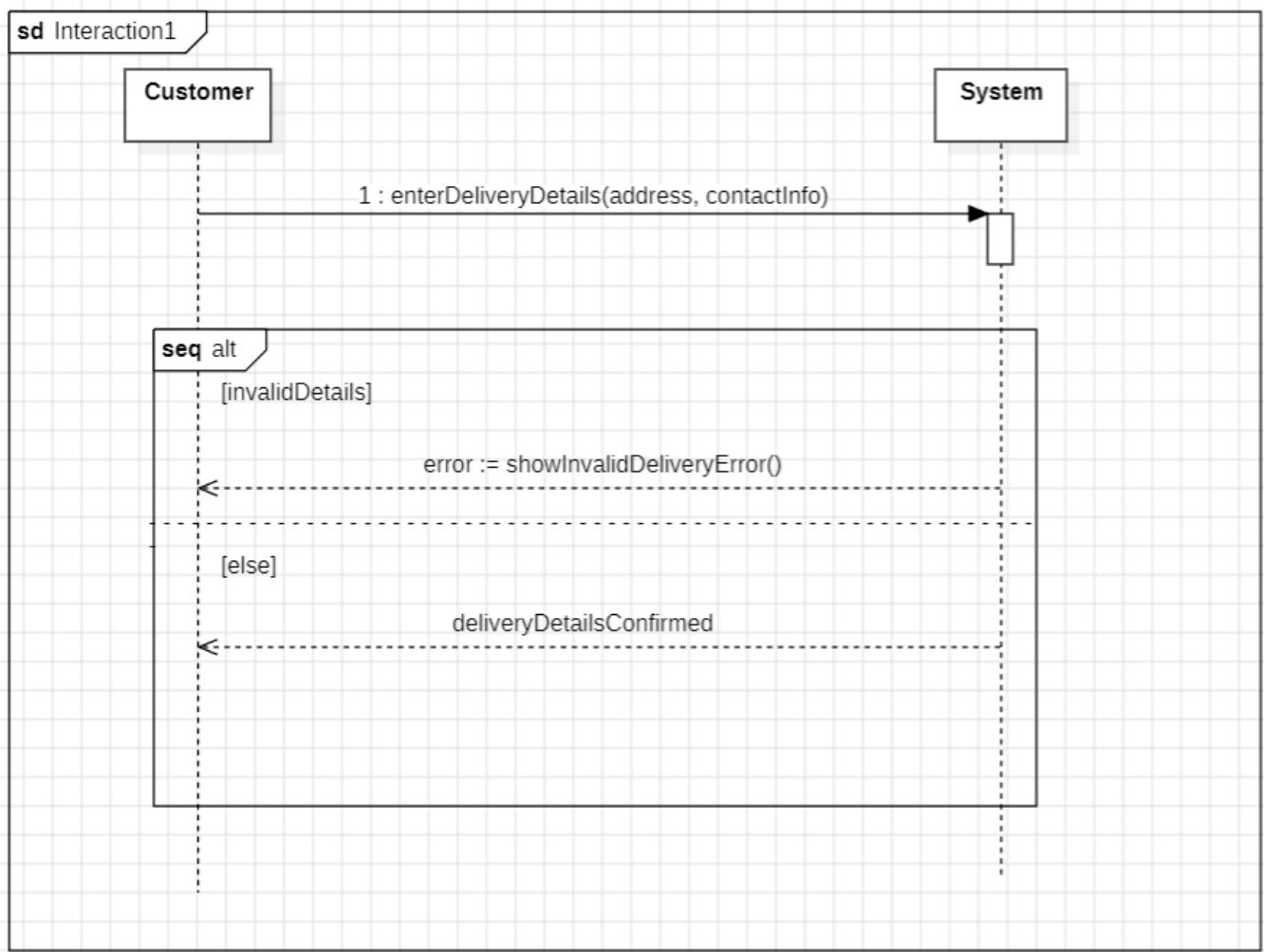
**SSD 1.1:**



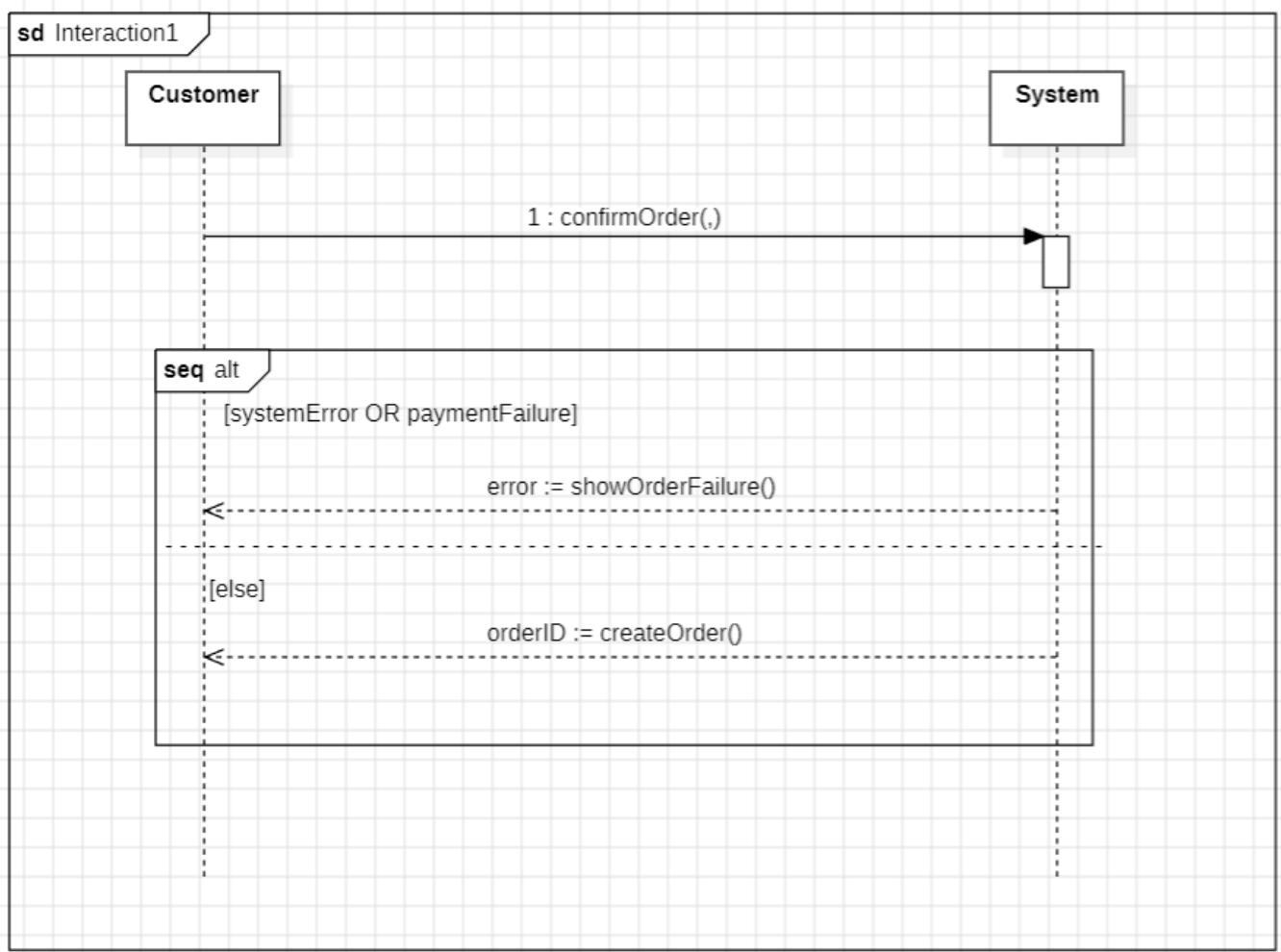
## SSD 1.2:



## SSD 1.3:

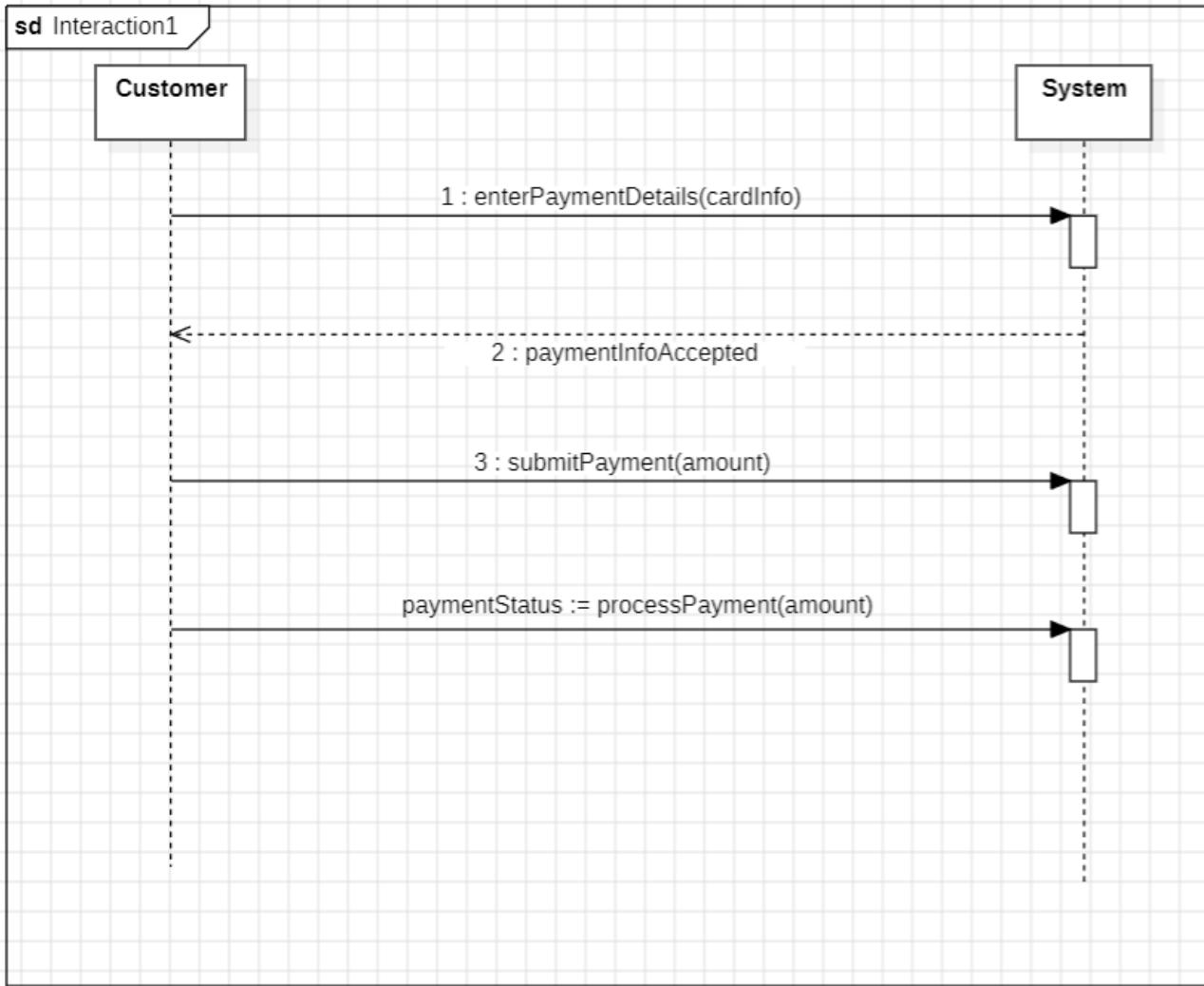


**SSD 1.4:**

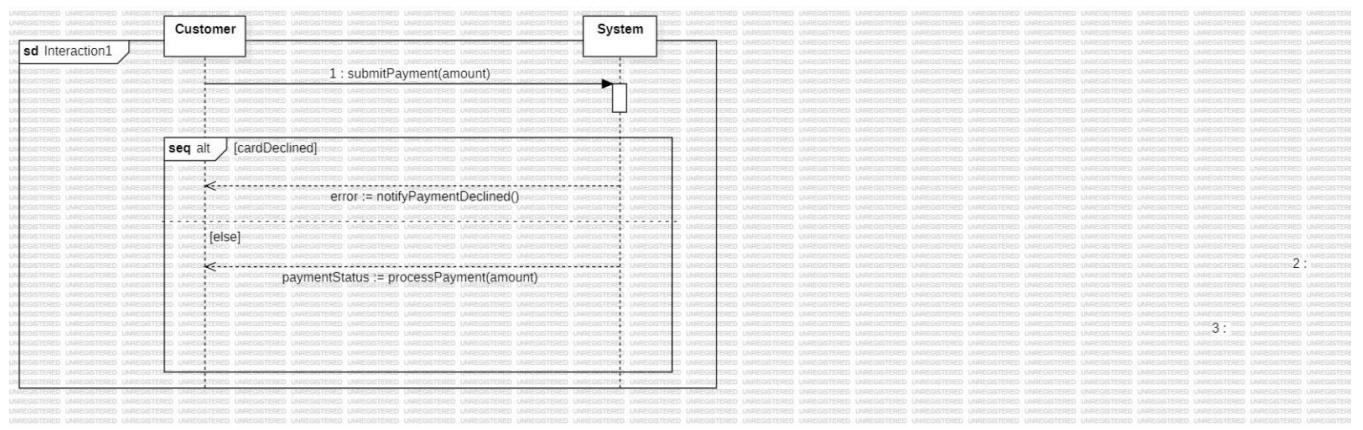


---

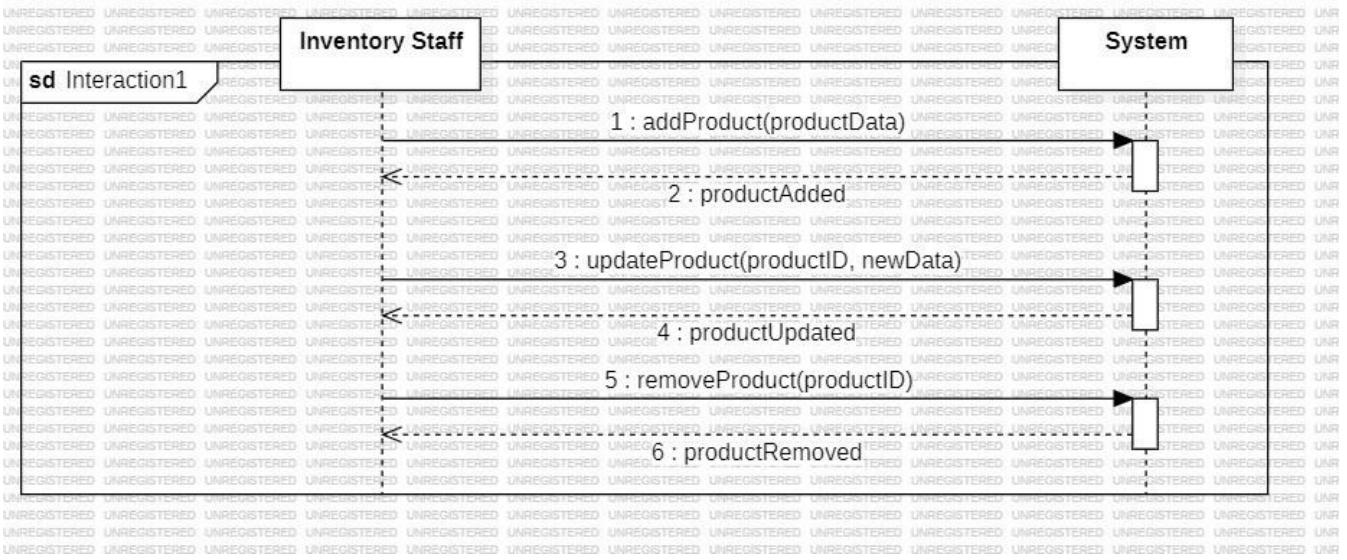
## SSD 2.1:



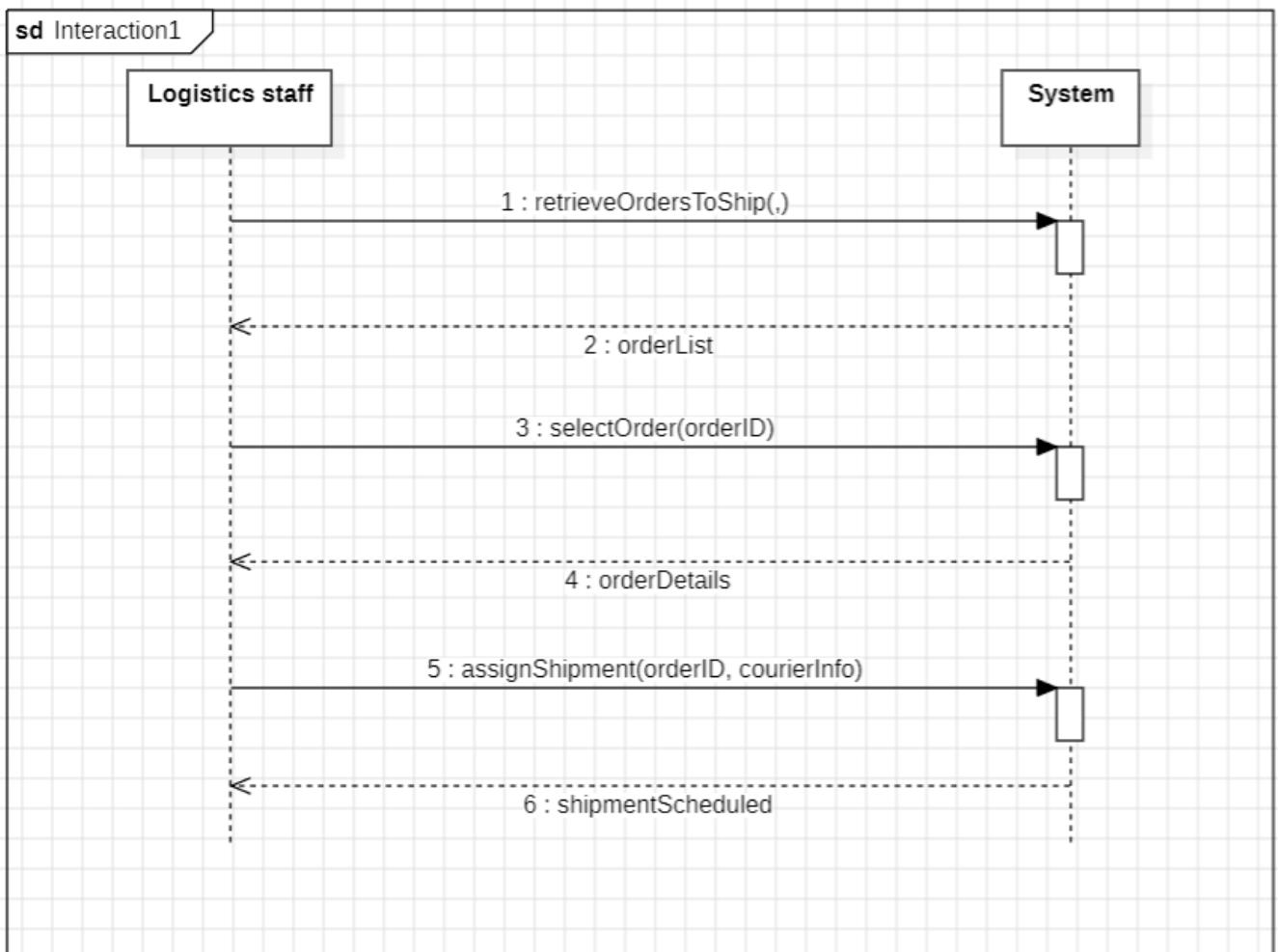
## SSD 2.2:



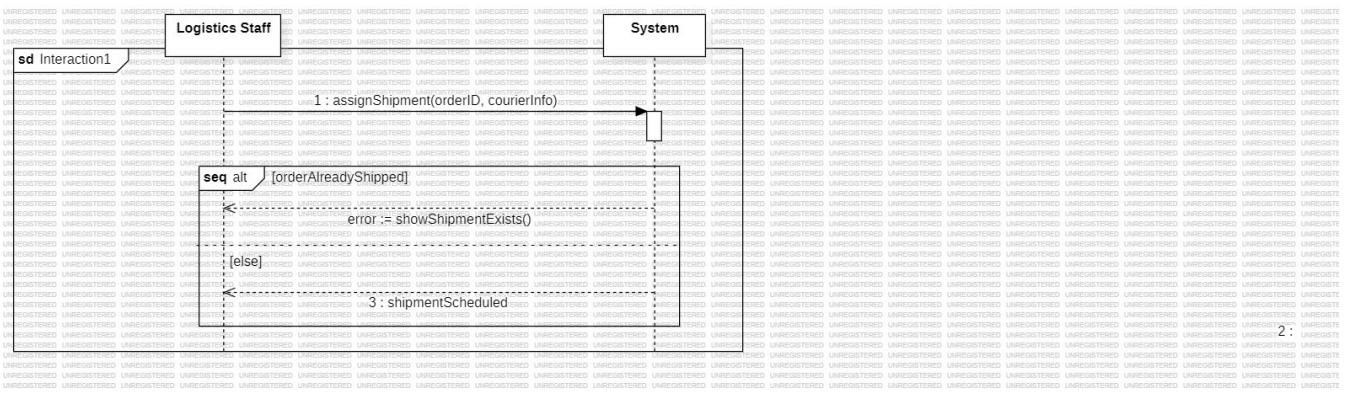
## SSD 3.1:



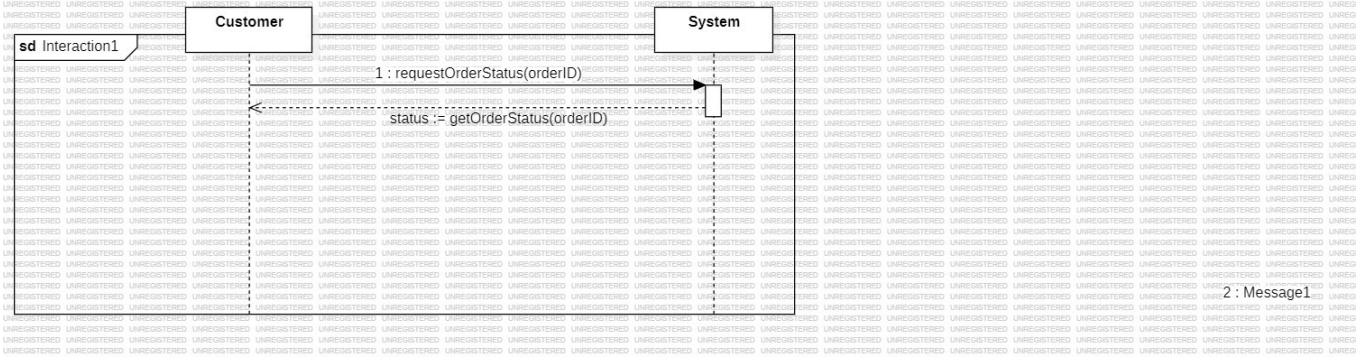
#### SSD 4.1:



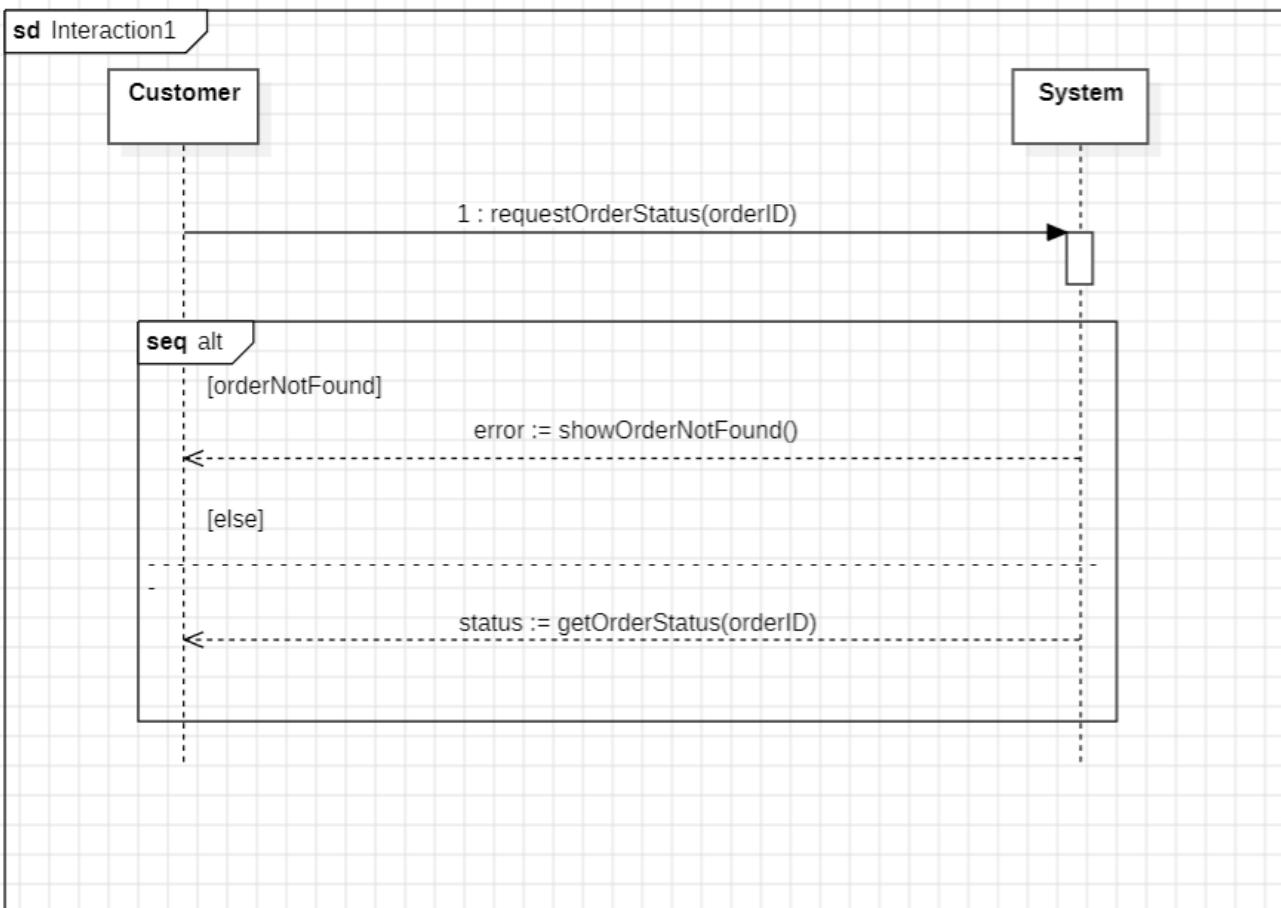
#### SSD 4.2:



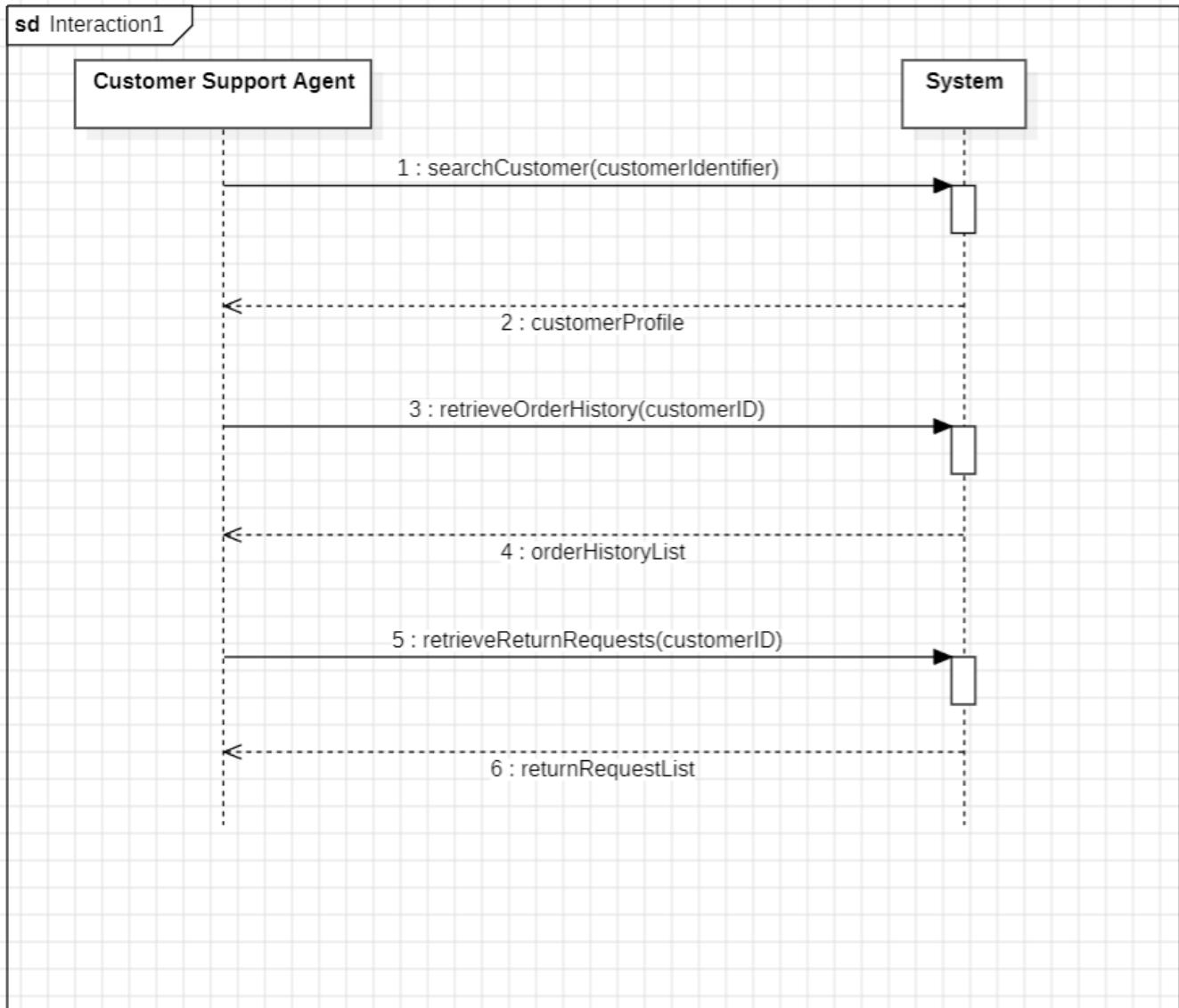
## SSD 5.1:



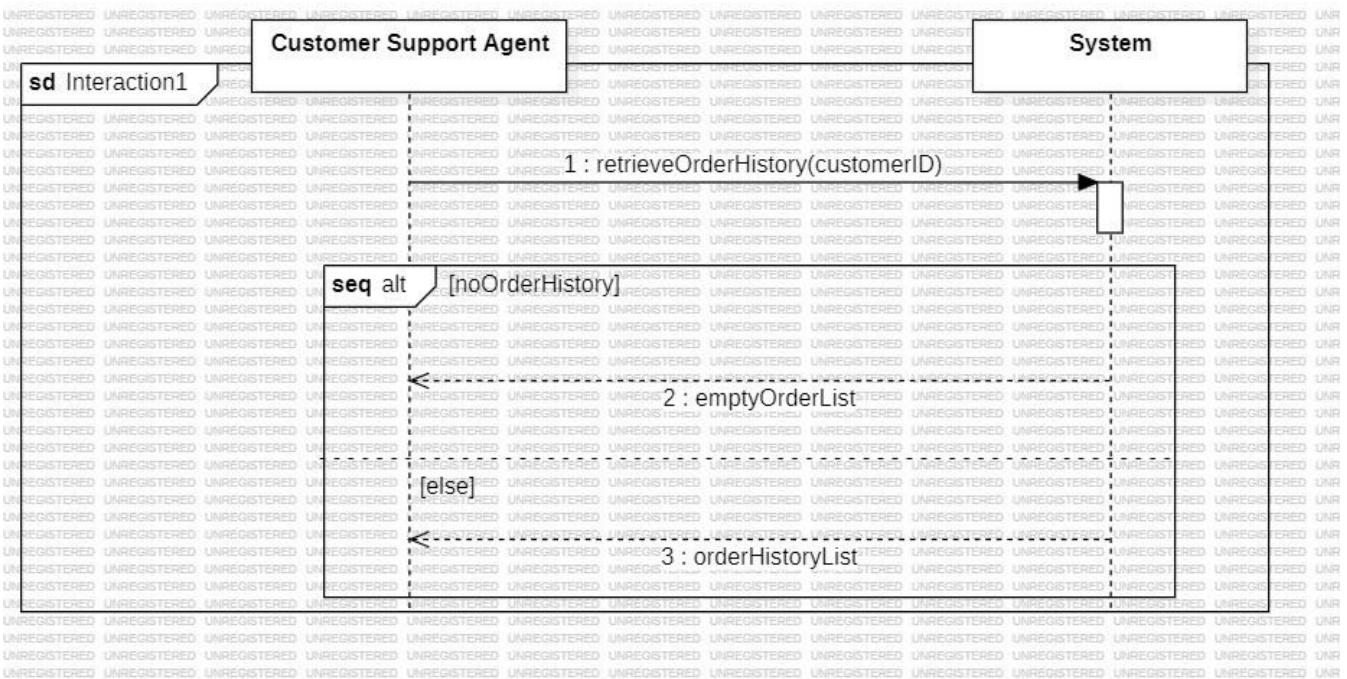
## SSD 5.2:



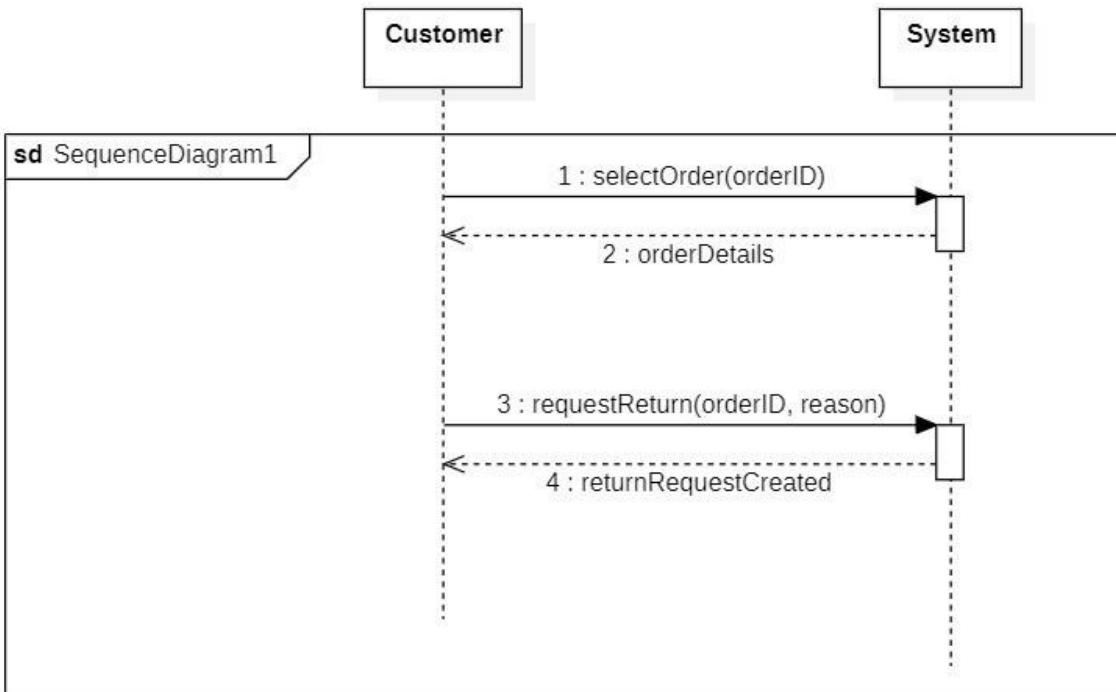
## SSD 6.1:



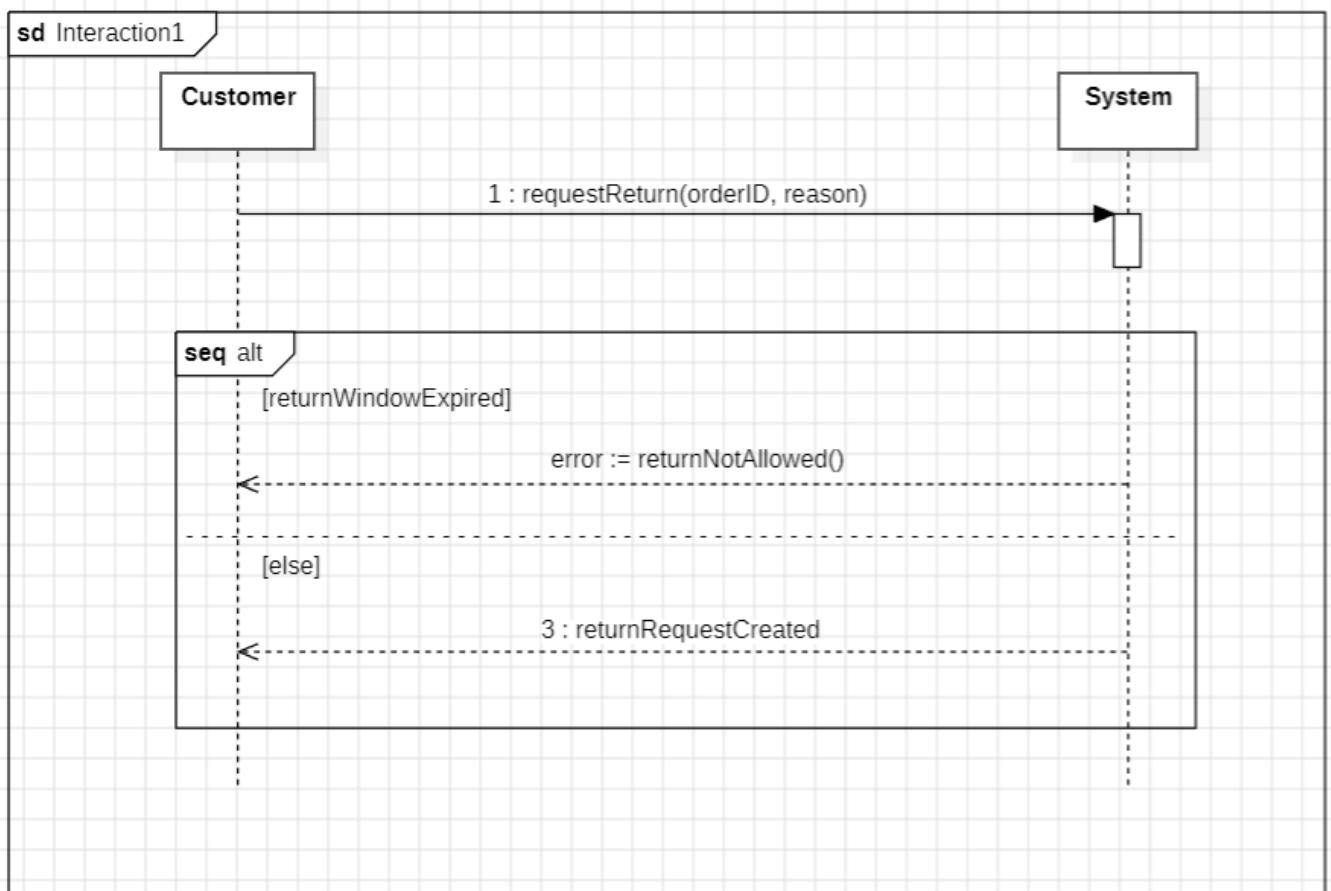
## SSD 6.2:



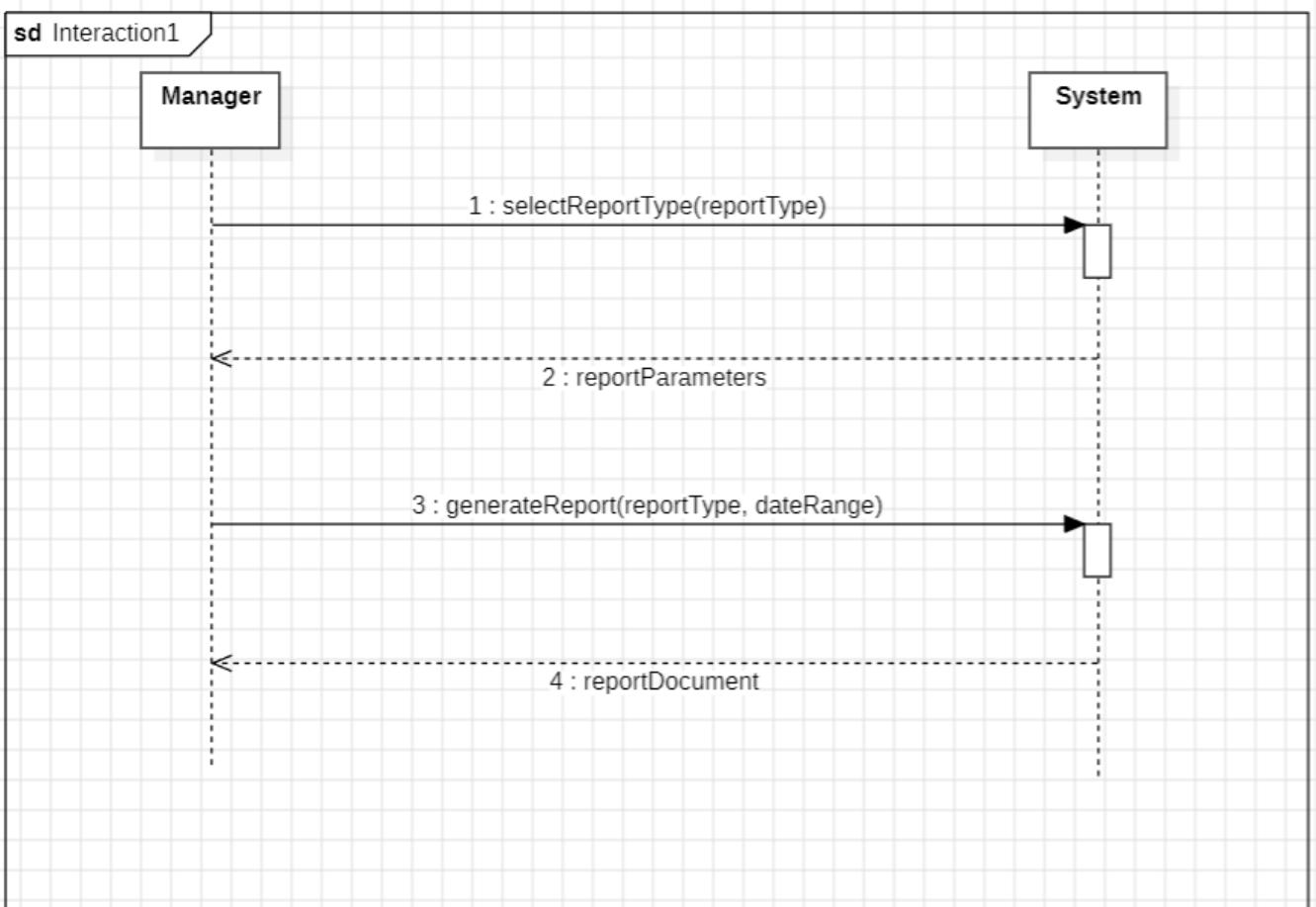
**SSD 7.1:**



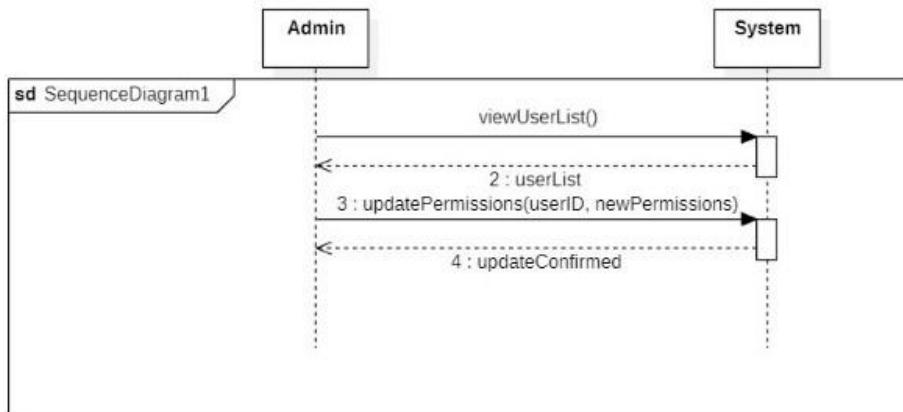
**SSD 7.2:**



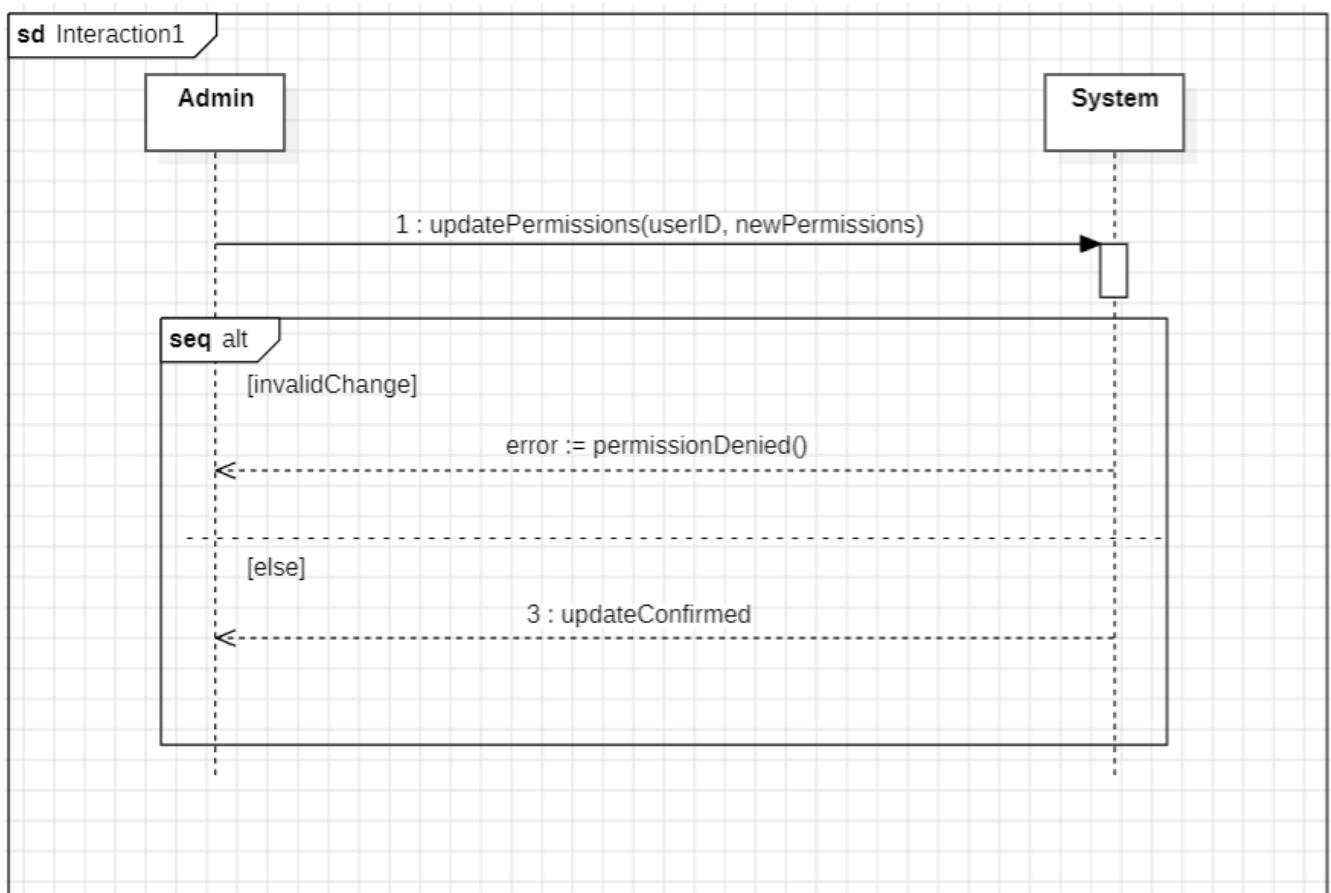
### SSD 8.1:



## SSD 9.1:

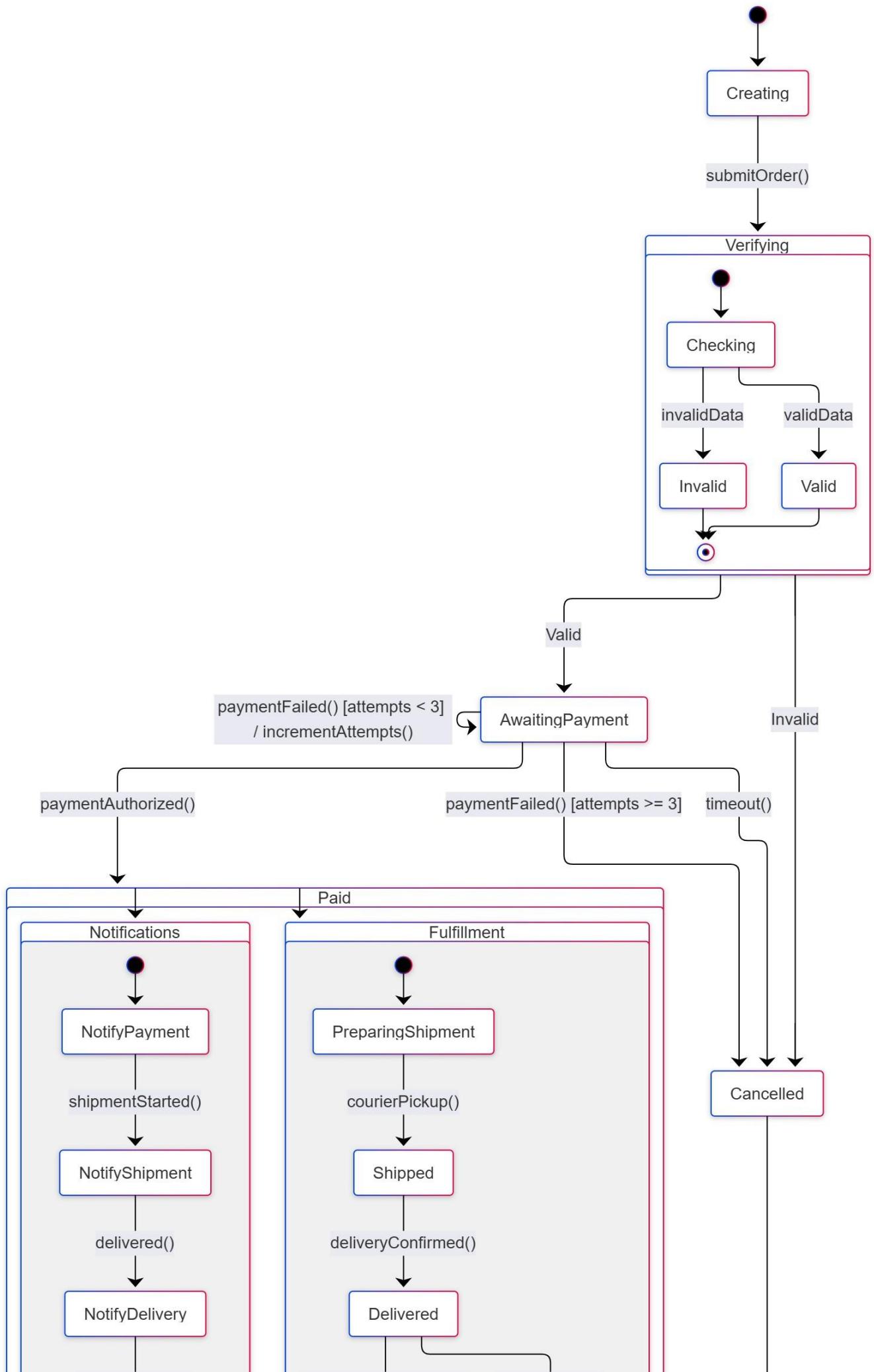


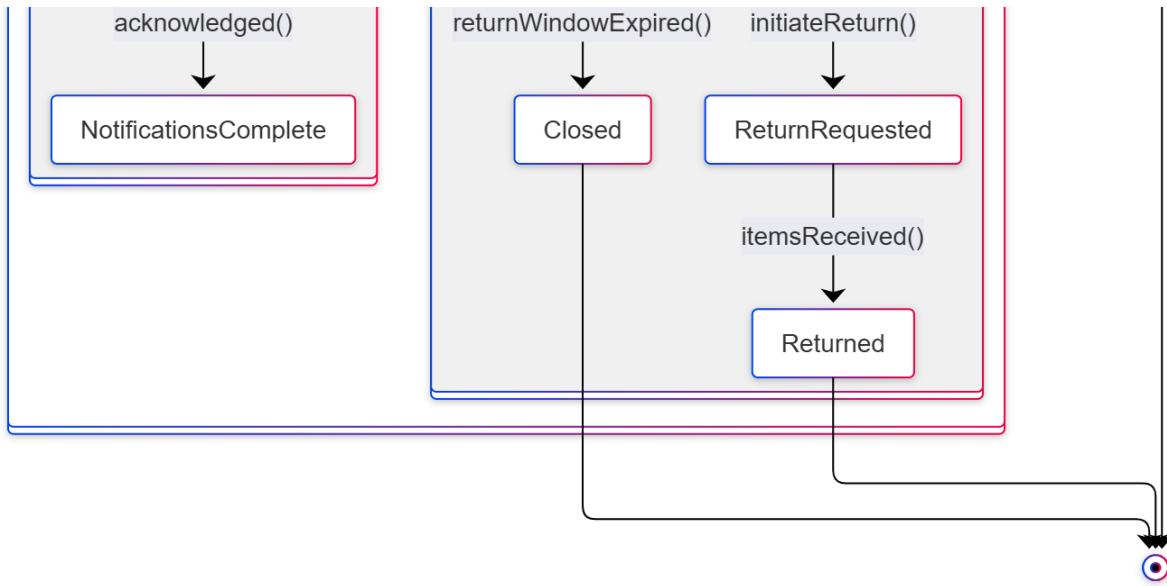
## SSD 9.2:



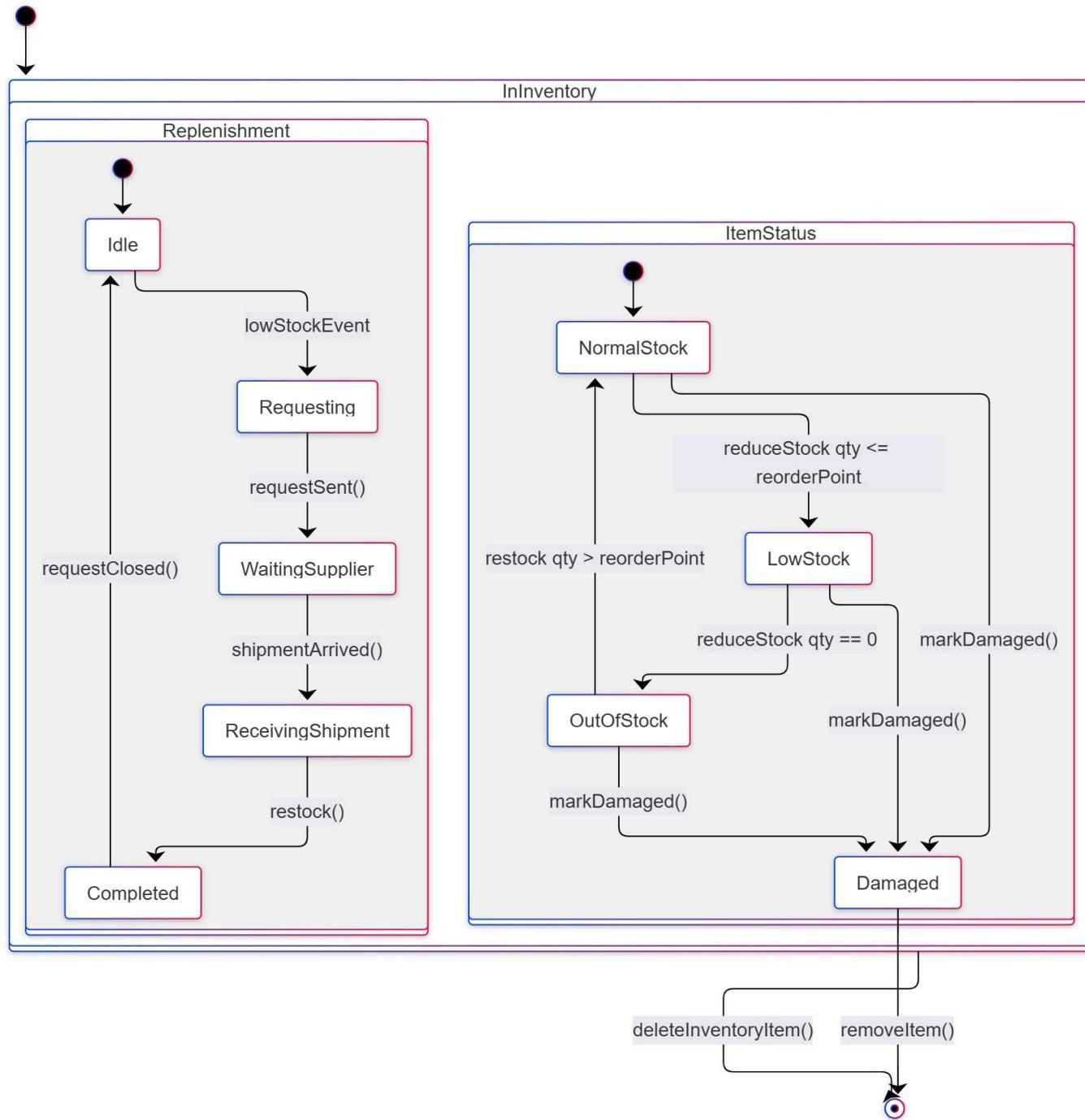
## State Machine Diagram

Order State Machine Diagram:





**InventoryItem State Machine Diagram:**



**Report 2**

# Volere

## Contents

### Project Drivers

1. The Purpose of the Project
2. The Stakeholders

### Project Constraints

3. Mandated Constraints
4. Naming Conventions and Terminology
5. Relevant Facts and Assumptions

### Functional Requirements

6. The Scope of the Work
7. Business Data Model & Data Dictionary
8. The Scope of the Product
9. Functional Requirements

### Non-functional Requirements

10. Look and Feel Requirements
11. Usability and Humanity Requirements
12. Performance Requirements
13. Operational and Environmental Requirements
14. Maintainability and Support Requirements
15. Security Requirements
16. Cultural Requirements
17. Legal Requirements

### Project Issues

18. Open Issues
19. Off-the-Shelf Solutions
20. New Problems
21. Task

## Atomic Requirements Shell

Requirement #: FR1-AR01	Requirement Type: Functional	Event/use case #: FR1 (Registration)
Description: The system shall allow a customer to create an account using a unique email address and password.		
Rationale: A unique account is required to identify customers and enable secure access.		
Source: Customer (End User)		
Fit Criteria: Registration succeeds only when the submitted email does not already exist in the system.		
Customer Satisfaction: 5	Customer Disatisfaction: 5	Priority: High
Dependencies: Customer entity in domain model		
Supporting Materials: Domain Model → Customer class		
History: Initial creation		

**Volere**

Copyright © Atlantic Systems Guild

Requirement #: FR1-AR02	Requirement Type: Functional	Event/use case #: FR1 (Login)
Description: The system shall authenticate users through verified email/username and password.		
Rationale: Authentication ensures that only valid users access system features.		
Source: Customer; All Staff Roles		
Fit Criteria: Login succeeds only when both identifier and password match stored credentials.		
Customer Satisfaction: 5	Customer Disatisfaction: 5	Priority: High
Dependencies: Credential storage; access control		
Supporting Materials: Domain Model → Staff subclasses		
History: Initial creation		

**Volere**

Copyright © Atlantic Systems Guild

Requirement #: FR1-AR03 Requirement Type: Event/use case #: US14  
Functional (Manage User Permissions)

Description: The system shall allow only the actions permitted for the user's assigned role.

Rationale: Protects system integrity by preventing unauthorized access.

Source: Administrator; Management

Fit Criteria: Attempting an action outside role permissions results in "access denied."

Customer Satisfaction: 4

Customer Disatisfaction: 5 Priority: High

Dependencies: User role definitions; permission mapping Conflicts: None

Supporting Materials: Use Case US14, Domain Model → Staff hierarchy

History: Initial creation

**Volere**

Copyright © Atlantic Systems Guild

Requirement #: FR2-AR01 Requirement Type: Event/use case #: US7 (Place Online Order)  
Functional

Description: The system shall allow customers to browse available products and view each product's name, description, price, category, and stock availability.

Rationale: Customers must be able to understand product details before placing an order.

Source: Customer (End User)

Fit Criteria: A customer selecting any product in the catalog shall see complete product details as defined in the domain model.

Customer Satisfaction: 5

Customer Disatisfaction: 5 Priority: High

Dependencies: Product class attributes;

Conflicts: None

Supporting Materials: Domain Model → Product, InventoryItem

History: Initial version

**Volere**

Copyright © Atlantic Systems Guild

Requirement #: <b>FR2-AR02</b>	Requirement Type: <b>Functional</b>	Event/use case #: <b>US7 (Place Online Order)</b>
Description: The system shall allow customers to add products to an order cart with specified quantities.		
Rationale: Customers need to assemble selected items before order placement.		
Source: Customer (End User)		
Fit Criteria: Adding an item updates the cart with product ID, quantity, and calculated line total.		
Customer Satisfaction: 4	Customer Disatisfaction: 5	Priority: High
Dependencies: Product; OrderItem	Conflicts: None	
Supporting Materials: Domain Model → Order, OrderItem		
History: Initial version		

**Volere**  
Copyright © Atlantic Systems Guild

Requirement #: <b>FR2-AR03</b>	Requirement Type: <b>Functional</b>	Event/use case #: <b>US7 (Place Online Order)</b>
Description: The system shall create a new order and automatically assign a unique Order ID when a customer submits the cart.		
Rationale: Every submitted order must be uniquely tracked throughout its lifecycle.		
Source: Customer (End User)		
Fit Criteria: Upon order submission, the system generates a unique Order ID and stores the order with its items.		
Customer Satisfaction: 5	Customer Disatisfaction: 5	Priority: High
Dependencies: Order entity; ID generation rules	Conflicts: None	
Supporting Materials: Domain Model → Order		
History: Initial version		

**Volere**  
Copyright © Atlantic Systems Guild

Requirement #: **FR3-AR01** Requirement Type: Event/use case #:  
Functional **US7 (Place Online Order)**

Description: The system shall verify that all products in the submitted order are available in sufficient quantity before the order is confirmed.

Rationale: Orders cannot proceed unless all requested items are in stock.

Source: **Sales Staff**

Fit Criteria: For every order item, the system confirms  $\text{quantityOnHand} \geq \text{requestedQuantity}$ . If not, order verification fails.

Customer Satisfaction: **5**

Customer Disatisfaction: **5**

Priority: **High**

Dependencies: **InventoryItem** stock levels

Conflicts: **None**

Supporting Materials: Domain Model → **InventoryItem**, **OrderItem**

History: **Initial version**

**Volere**

Copyright © Atlantic Systems Guild

Requirement #: **FR3-AR02**

Requirement Type: Functional

Event/use case #: **US7 (Place Online Order)**

Description: The system shall validate the customer's contact information (name, email, phone, and default address) before the order is confirmed.

Rationale: Incorrect or incomplete customer details prevent proper delivery and communication.

Source: **Sales Staff**

Fit Criteria: Order verification succeeds only when all required customer fields are populated and meet validation rules.

Customer Satisfaction: **4**

Customer Disatisfaction: **5**

Priority: **High**

Conflicts: **None**

Dependencies: **Customer** entity in domain model

Supporting Materials: Domain Model → **Customer**

History: **Initial version**

**Volere**

Copyright © Atlantic Systems Guild

Requirement #: **FR3-AR03** Requirement Type: Event/use case #: Functional **US7 (Place Online Order)**  
 Description: The system shall automatically cancel an order if required product or customer information fails verification and notify the customer.  
 Rationale: Prevents invalid orders from entering payment and shipment workflows.  
 Source: Sales Staff  
 Fit Criteria: An order failing verification is marked “Cancelled,” and a notification is sent to the customer.

Customer Satisfaction: **3** Customer Dissatisfaction: **5** Priority: **High**  
 Dependencies: **FR3-AR01, FR3-AR02** Conflicts: **None**  
 Supporting Materials: Order status rules in domain model  
 History: Initial version

**Volere**  
Copyright © Atlantic Systems Guild

Requirement #: **FR4-AR01** Requirement Type: Event/use case #: **US4** Functional (**Process Customer Payment**)  
 Description: The system shall validate the payment details (method, amount, and required authorization fields) before submitting the transaction to the payment gateway.  
 Rationale: Prevents invalid or incomplete payment requests from being sent to the gateway.  
 Source: Finance Staff  
 Fit Criteria: Payment is submitted only when all required payment fields are present and valid; otherwise the system rejects the transaction.

Customer Satisfaction: **4** Customer Dissatisfaction: **5**  
 Priority: **High** Conflicts: **None**  
 Dependencies: Payment entity; Order total; Gateway integration  
 Supporting Materials: Domain Model → Payment  
 History: Initial version

**Volere**  
Copyright © Atlantic Systems Guild

Requirement #: **FR4-AR03** Requirement Type: Event/use case #: **US4**  
Functional (Process Customer Payment)

Description: The system shall mark the order as “Paid” for successful transactions and “Cancelled” for failed transactions.

Rationale: Order flow depends on correct payment decisions; failed payments must not proceed to shipment.

Source: Finance Staff

Fit Criteria: Successful payments change the order status to “Paid”; failed payments change it to “Cancelled” and trigger customer notification.

Customer Satisfaction: **4**

Customer Dissatisfaction: **5**

Priority: **High**

Dependencies: **FR4-AR01, FR4-AR02**

Conflicts: **None**

Supporting Materials: Order status definitions in domain model

History: Initial version

**Volere**

Copyright © Atlantic Systems Guild

Requirement #: **FR5-AR01** Requirement Type: Event/use case #:  
Functional US3 (Manage Product Catalog),  
US7 (Order Placement)

Description: The system shall update and display current stock levels for each product based on inventory records and order activity.

Rationale: Accurate stock levels are required for catalog display, order verification, and replenishment triggers.

Source: Inventory Staff

Fit Criteria: For any product, the system displays quantityOnHand matching the latest committed inventory and order reservations.

Customer Satisfaction: **5**

Customer Disatisfaction: **5**

Priority: **High**

Conflicts: **None**

Dependencies: **InventoryItem entity**

Supporting Materials: Domain Model → InventoryItem(quantityOnHand)

History: Initial version

**Volere**

Copyright © Atlantic Systems Guild

Requirement #:	<b>FR5-AR02</b>	Requirement Type:	Event/use case #:
		Functional	<b>US7 (Place Online Order)</b>
Description: When an order is confirmed, the system shall reserve the ordered quantity by deducting it from available inventory.			
Rationale: Prevents overselling and ensures accurate availability for subsequent customers.			
Source: Sales Staff; Inventory Staff			
Fit Criteria: After an order is confirmed, quantityOnHand for each product is reduced by the ordered amount.			
Customer Satisfaction:	<b>4</b>	Customer Disatisfaction:	<b>5</b>
Priority:	<b>High</b>	Conflicts:	<b>None</b>
Dependencies: FR3-AR01 (product availability verification)			
Supporting Materials: Domain Model → InventoryItem			
History: Initial version			

**Volere**  
Copyright © Atlantic Systems Guild

Requirement #:	<b>FR5-AR03</b>	Requirement Type:	Event/use case #:
		Functional	System Event (Low Stock Trigger)
Description: When quantityOnHand falls below the product's reorder point, the system shall create a replenishment request for the supplier.			
Rationale: Maintains adequate stock and supports supplier communication.			
Source: Inventory Staff			
Fit Criteria: A replenishment request is automatically created when quantityOnHand < reorderPoint.			
Customer Satisfaction:	<b>4</b>	Customer Dissatisfaction:	<b>4</b>
Priority:	<b>Medium</b>	Conflicts:	<b>None</b>
Dependencies: ReplenishmentRequest entity; Supplier			
Supporting Materials: Domain Model → ReplenishmentRequest			
History: Initial version			

**Volere**  
Copyright © Atlantic Systems Guild

Requirement #:	<b>FR5-AR04</b>	Requirement Type:	Event/use case #:
		Functional	<b>US3</b> (Manage Product Catalog)
Description:	The system shall allow authorized inventory staff to add new products, edit product details, or remove products from the catalog.		
Rationale:	Ensures the product catalog remains current and accurate for all customer-facing operations.		
Source:	<b>Inventory Staff</b>		
Fit Criteria:	An authorized user can successfully create, modify, or delete any product; unauthorized users receive an access-denied response.		
Customer Satisfaction:	<b>5</b>	Customer Disatisfaction:	<b>4</b>
Priority:	<b>High</b>	Conflicts:	<b>None</b>
Dependencies:	<b>FR1-AR03</b> (role-based access control)		
Supporting Materials:	Domain Model → Product		
History:	<b>Initial version</b>		

**Volere**  
Copyright © Atlantic Systems Guild

Requirement #:	<b>FR6-AR01</b>	Requirement Type:	Event/use case #:
		Functional	<b>US5</b> (Plan Shipment)
Description:	The system shall assign a courier service and create a shipment record for an order once the package is ready for dispatch.		
Rationale:	Shipment planning is required to hand off the order to the logistics provider and initiate delivery.		
Source:	<b>Logistics Staff</b>		
Fit Criteria:	A shipment record is created containing: shipmentId, courier, order reference, ship date.		
Customer Satisfaction:	<b>4</b>	Customer Disatisfaction:	<b>5</b>
Priority:	<b>High</b>	Conflicts:	<b>None</b>
Dependencies:	Courier entity; Order status ("Ready for Shipment")		
Supporting Materials:	Domain Model → Shipment, Courier		
History:	<b>Initial version</b>		

**Volere**  
Copyright © Atlantic Systems Guild

Requirement #: **FR6-AR02** Requirement Type: Event/use case #: Functional US5 (Plan Shipment)  
 Description: The system shall generate a tracking number and calculate an estimated delivery date for each shipment.  
 Rationale: Customers and staff require traceable shipment data to monitor delivery progress.  
 Source: Logistics Staff  
 Fit Criteria: Each shipment record contains a unique tracking number and an estimated delivery timestamp.  
 Customer Satisfaction: 5 Customer Dissatisfaction: 5  
 Priority: High Customer Disatisfaction:  
 Dependencies: FR6-AR01; Courier details Conflicts: None  
 Supporting Materials: Domain Model → Shipment(trackingNumber, estimatedDelivery)  
 History: Initial version



Requirement #: **FR6-AR03** Requirement Type: Event/use case #: Functional US5 (Plan Shipment)  
 Description: The system shall update both shipment status and the corresponding order status (e.g., Processing → Shipped → Delivered).  
 Rationale: Accurate shipment and order statuses are required for customer tracking and internal workflows.  
 Source: Logistics Staff  
 Fit Criteria: Status transitions follow defined workflow rules and update both Shipment and Order entities consistently.  
 Customer Satisfaction: 5 Customer Disatisfaction: 4 Priority: High  
 Dependencies: Shipment entity; Order entity Conflicts: None  
 Supporting Materials: Domain Model → Shipment, Order  
 History: Initial version



Requirement #: **FR7-AR01** Requirement Type: Event/use case #:  
Functional US7 (Place Online Order)

Description: The system shall send an automated notification to the customer when an order is successfully submitted.

Rationale: Customers must receive immediate confirmation that the order was recorded.

Source: Customer (End User)

Fit Criteria: A confirmation notification containing the Order ID is delivered upon successful order creation.

Customer Satisfaction: 5

Customer Disatisfaction: 5

Priority: High

Conflicts: None

Dependencies: FR2-AR03 (Order creation)

Supporting Materials: Order entity with Order ID

History: Initial version

**Volere**  
Copyright © Atlantic Systems Guild

Requirement #: **FR7-AR02** Requirement Type: Event/use case #:  
Functional US5 (Plan Shipment)

Description: The system shall send automated notifications to the customer when an order is shipped and again when it is delivered.

Rationale: Customers expect real-time updates on shipment progress.

Source: Logistics Staff; Customer

Fit Criteria: Notifications are generated when the order status transitions to "Shipped" and "Delivered."

Customer Satisfaction: 5 Customer Dissatisfaction: 4 Priority:

Customer Satisfaction: Customer Dissatisfaction: High

Dependencies: FR6-AR03 (Shipment status updates) Conflicts: None

Supporting Materials: Shipment(trackingNumber), Order(status)

History: Initial version

**Volere**  
Copyright © Atlantic Systems Guild

Requirement #: **FR7-AR03** Requirement Type: **Functional** Event/use case #: **U98 (Track Order Status)**

Description: **The system shall allow customers to retrieve and view the current shipment status using a tracking number.**

Rationale: **Customers must be able to monitor their order's delivery progress.**

Source: **Customer (End User)**

Fit Criteria: **Entering a valid tracking number returns the shipment status and estimated delivery date.**

Customer Satisfaction: **5**

Customer Disatisfaction: **5**

Dependencies: **High**

Priority: **None**

Supporting Materials: **Shipment entity; Order status updates**

History: **Domain Model → Shipment, Order**

**Initial version**

**Volere**

Copyright © Atlantic Systems Guild

Requirement #: **FR8-AR01** Requirement Type: **Functional (Place Online Order) / Payment Completion** Event/use case #: **U97**

Description: **The system shall generate an invoice for each completed order, containing product details, quantities, totals, taxes, and shipping charges.**

Rationale: **Customers and Finance staff require a formal record of each completed transaction.**

Source: **Finance Staff; Customer**

Fit Criteria: **An invoice document is automatically created when order status changes to "Paid," containing all required fields defined in the domain model.**

Customer Satisfaction: **5**

Customer Disatisfaction: **4** Priority: **High**

Dependencies: **Payment confirmation; Order details** Conflicts: **None**

Supporting Materials: **Domain Model → Invoice(total, tax, shipping charges)**

History: **Initial version**

**Volere**

Copyright © Atlantic Systems Guild

Requirement #: <b>FR8-AR02</b>	Requirement Type: <b>Functional</b>	Event/use case #: <b>US6 (Look Up Order History)</b>
Description: The system shall store generated invoices and allow authorized users to retrieve them for customer support or financial review.		
Rationale: Invoices must remain accessible for customer support, audits, and record-keeping.		
Source: Customer Support; Finance Staff		
Fit Criteria: Given an order ID, the system returns the associated invoice if one exists.		
Customer Satisfaction: 4	Customer Dissatisfaction: 4	
Priority: Medium	Priority: Medium	
Dependencies: FR8-AR01; Invoice entity	Conflicts: None	
Supporting Materials: Domain Model → Invoice		
History: Initial version		

**Volere**  
Copyright © Atlantic Systems Guild

Requirement #: <b>FR8-AR03</b>	Requirement Type: <b>Functional</b>	Event/use case #: <b>US13</b> <b>(Generate Management Reports)</b>
Description: The system shall generate analytical and summary reports containing business metrics such as order volume, revenue totals, and category-level sales.		
Rationale: Management requires periodic insights to evaluate business performance.		
Source: Manager / Executive		
Fit Criteria: A report can be generated that includes at least the parameters defined in the Report entity: type, generatedDate, and report parameters.		
Customer Satisfaction: 5	Customer Dissatisfaction: 3	
Priority: Medium	Conflicts: None	
Dependencies: Order, Payment, Product data		
Supporting Materials: Domain Model → Report		
History: Initial version		

**Volere**  
Copyright © Atlantic Systems Guild

Requirement #: FR9-AR01 Requirement Type: Event/use case #: US9  
Functional (Initiate Return Request)

Description: The system shall allow customers to submit a return request or report an issue by specifying the order, reason, and supporting details.

Rationale: Customers must be able to request returns or report problems through the system instead of manual channels.

Source: Customer (End User)

Fit Criteria: A valid return request creates a new ReturnRequest record linked to the order.

Customer Satisfaction: 5

Customer Disatisfaction: 5 Priority: High

Dependencies: Order history; ReturnRequest entity Conflicts: None

Supporting Materials: Domain Model → ReturnRequest(returnId, reason, requestDate, status)

History: Initial version

**Volere**

Copyright © Atlantic Systems Guild

Requirement #: FR9-AR02 Requirement Type: Event/use case #: US9  
Functional (Initiate Return Request)

Description: The system shall allow Customer Support staff to update the status of a return request (e.g., Submitted → Under Review → Approved/Rejected).

Rationale: Return handling requires multiple status changes and follow-up actions by staff.

Source: Customer Support Staff

Fit Criteria: A support agent can change the status of any return they are authorized to access, and the system records the update.

Customer Satisfaction: 4

Customer Disatisfaction: 4 Priority: Medium

Dependencies: FR9-AR01; Staff role permissions Conflicts: None

Supporting Materials: Domain Model → ReturnRequest(status)

History: Initial version

**Volere**

Copyright © Atlantic Systems Guild

Requirement #: **FR9-AR03** Requirement Type: Functional Event/use case #: **US6**  
(Look Up Order History)

Description: The system shall allow Customer Support staff to retrieve the complete order history of any customer to assist with resolving issues or validating returns.

Rationale: Support agents require full order information to verify claims and provide resolutions.

Source: Customer Support Staff

Fit Criteria: Given a customer ID, the system displays all past orders with their associated details and statuses.

Customer Satisfaction: 5

Customer Disatisfaction: 4

Priority: High

Conflicts: None

Dependencies: Order entity; Invoice entity

Supporting Materials: Domain Model → Order, Invoice, ReturnRequest

History: Initial version

**Volere**  
Copyright © Atlantic Systems Guild

Requirement #: **FR10-AR01** Requirement Type: Functional Event/use case #: **US4**  
US4 (Process Customer Payment)

Description: The system shall communicate with the payment gateway to submit payment requests and receive authorization responses.

Rationale: Order completion requires external validation from the payment provider.

Source: Finance Staff

Fit Criteria: The system sends a valid payment request and receives an authorization code or failure response from the gateway.

Customer Satisfaction: 5

Customer Disatisfaction: 5 Priority: High

Dependencies: Payment API endpoint; Payment entity Conflicts: None

Supporting Materials: Domain Model → Payment(authorizationCode, status)

History: Initial version

**Volere**  
Copyright © Atlantic Systems Guild

Requirement #:  
**FR10-AR02**

Requirement Type:  
**Functional**

Event/use case #:  
**US5 (Plan Shipment),**

Description: The system shall retrieve shipment updateUS8 (Track Order Status) courier's tracking API using the assigned tracking number.

Rationale: Customers and staff require accurate real-time shipment status.

Source: Logistics Staff; Customer

Fit Criteria: Given a tracking number, the system displays status information retrieved from the courier API.

Customer Satisfaction: 5

Customer Disatisfaction: 4

Priority: High

Dependencies:

Conflicts: None

Dependencies: Courier(apiEndpoint), Shipment(trackingNumber)

Supporting Materials: Domain Model → Courier(apiEndpoint), Shipment

History: Initial version

**Volere**  
Copyright © Atlantic Systems Guild

Requirement #: FR10-AR03 Requirement Type: Functional Event/use case #: System Event (Low Stock Trigger)

Description: The system shall send a replenishment request to the appropriate supplier when product stock falls below the reorder point.

Rationale: Automated restocking ensures product availability and reduces manual workload.

Source: Inventory Staff

Fit Criteria: A replenishment request is transmitted to the supplier system when quantityOnHand < reorderPoint.

Customer Satisfaction: 4

Customer Disatisfaction: 4

Priority: Medium

Dependencies: Supplier(contactInfo)

Conflicts: None

Supporting Materials: Domain Model → Supplier, ReplenishmentRequest

History: Initial version

**Volere**  
Copyright © Atlantic Systems Guild

## 1. The Purpose of the Project

## 1a. The User Business or Background of the Project Effort

### Content

E-Market Inc. relies on disconnected e-commerce tools that operate independently across Sales, Finance, Inventory, Logistics, and Customer Support. This fragmentation leads to data duplication, communication gaps, inconsistent stock levels, and delayed order processing.

The organization requires a unified E-Commerce Order Management System (E-OMS) to support customers placing online orders and internal teams verifying orders, processing payments, managing inventory, planning shipments, and handling returns.

This system will deliver fully integrated order-to-delivery operations.

### Motivation

Without this project, operational inefficiencies will persist and worsen as order volume increases. Manual data handling, inconsistent stock visibility, and inter-departmental communication failures continue to cause cancellations, delays, and customer dissatisfaction.

A single integrated system provides justification for the project by ensuring accuracy, speed, and reliable coordination..

### Form

- **Current Situation Model:**

The System Vision Document and Deliverable 2 describe the fragmented workflow across Sales, Finance, Inventory, Logistics, and Customer Support, showing how disconnected tools lead to duplication and stock errors. These documents collectively serve as the current situation model.

(From *System Vision Document*, Deliverable 2 )

- **Business Process Models:**

The Event Decomposition results, Use Case models, and the order-to-delivery workflow captured in your earlier UML diagrams represent the business processes currently in use and the ones intended for the E-OMS.

(From Use Cases, Event Decomposition, and UML diagrams in your project files)

- **Samples of Current Documents:**

The Technical Reports and requirement summaries document the existing order handling process, including verification steps, stock checks, payment workflows, and shipping coordination.

(From Technical Report files already uploaded)

- **Website / Online Context:**

E-Market operates as an online retail platform, and the system is designed to support the end-to-end ordering process typical of e-commerce websites. Even though no external website captures are included, the online store context is fully defined within the project scope.

(From System Vision Document scope description)

- **Organization Structure:**

The stakeholder table in Deliverable 2 effectively reflects the organizational structure relevant to the project—Sales, Finance, Inventory, Logistics, Customer Support, Management, System Administrators, and external partners (Suppliers, Couriers, Payment Gateways).

(Deliverable 2 Stakeholders section

## 1b. Goals of the Project

### Content

The E-OMS aims to centralize and automate the entire order lifecycle so that E-Market Inc. can eliminate data duplication, reduce manual processing, improve stock accuracy, and provide customers with reliable order tracking. This contributes directly to the organization's overall goal of improving operational efficiency, strengthening partner coordination, and delivering a smoother, faster customer experience..

### Motivation

As development progresses, there is a risk that the team may drift away from the original business intent or expand the scope in ways not aligned with the problem being solved. Making the goals explicit prevents the system from wandering away from its intended purpose unless the client intentionally approves a change. The goals must remain visible throughout reviews so stakeholders can ensure the evolving solution continues to address the documented inefficiencies in order verification, payments, inventory coordination, logistics, and customer support.

### Examples

- We want to provide accurate, real-time order processing and tracking for customers who purchase products online.
- We want to ensure inventory levels remain synchronized across departments to reduce stock errors and delays.

### Measurement

A goal is only reasonable if it can be measured. The advantages gained by E-OMS can be quantified through:

- Reduction in duplicated data entries across Sales, Finance, and Inventory (service goal)
- Decrease in order processing time from placement to shipment (service goal)
- Increase in on-time deliveries and reduction in shipment delays (service goal)
- Increase in repeat customer orders or customer satisfaction indicators (revenue goal)

- Compliance with online payment security standards required by payment gateways (legal goal)

These measurements determine whether the system delivers the intended business value.

#### Form

##### **Purpose:**

To build an integrated system that automates and synchronizes the complete order lifecycle and removes operational inefficiencies caused by disconnected tools.

##### **Advantage:**

The business gains faster order handling, real-time stock accuracy, reduced errors, improved partner coordination, and a more reliable customer experience.

##### **Measurement:**

Success will be evaluated by tracking reductions in manual work, fewer stock inconsistencies, improved order processing time, better delivery accuracy, and measurable improvements in customer satisfaction.

## 2. The Stakeholders

### 2a. The Client

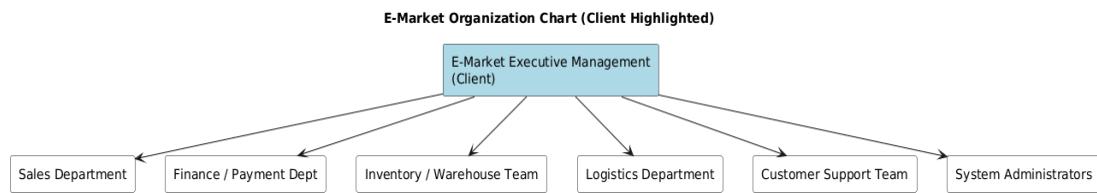
#### Content

The client for the E-OMS project is E-Market Executive Management. They authorize the project, define the business objectives, provide funding, and will approve the final system. They require consolidated operational visibility and accurate reporting across Sales, Finance, Inventory, Logistics, and Customer Support.

#### Motivation

Executive Management is responsible for ensuring the new system supports the company's strategic goals: reducing operational inefficiencies, improving coordination between departments, and strengthening customer experience. Their approval determines whether the delivered system satisfies the business needs.

## Form



### Decisions for Which the Client Is Responsible

1. Approval of the project scope, including all included and excluded system capabilities.
2. Approval of the system goals and success measurements, including operational efficiency targets and reporting requirements.
3. Final approval of all functional and non-functional requirements, including security, reliability, and performance expectations.
4. Authorization of the required budget and resources for system development and deployment.
5. Decisions regarding priority conflicts between departments (Sales, Finance, Inventory, Logistics, Customer Support).
6. Approval of integrations with external partners, including payment gateways, suppliers, and courier services.
7. Approval of system-level policies, such as data handling, access control, and compliance expectations.
8. Final acceptance of the completed E-OMS system, including acceptance testing results and deployment readiness.
9. Approval of any changes to scope, requirements, or timelines during the project lifecycle.
10. Approval of analytical reporting needs, KPI definitions, and management dashboards delivered by the system.

## 2b. The Customer

### Content

The primary customer of the E-OMS is **the end user who purchases products from E-Market's online store**. This includes individuals who browse the catalog, place orders, make payments, track shipments, and request returns. For the internal aspect of the system, **department managers** also act as customers because they decide whether their teams (Sales, Finance, Inventory, Logistics, Customer Support) will adopt and rely on the new integrated system.

### Motivation

The customer's needs define what the system must support: fast and reliable order placement, secure payment handling, accurate tracking, and an accessible returns process. Their use of the system determines whether it achieves its intended value.

### Form

#### Decisions for Which the Customer Is Responsible

1. Choosing to use the online ordering process.
2. Confirming the accuracy of order details during checkout.
3. Approving payment transactions through the provided gateway interface.
4. Deciding when to initiate a return or refund request.
5. Providing feedback during prototype or interface reviews.

#### Customer Review Checkpoints

1. Review of early catalog and checkout interface drafts.
2. Review of order tracking interface.
3. Review of returns submission flow.

## 2c. Other Stakeholders

### Content

Other stakeholders include all individuals and external parties who interact with the E-OMS or whose work depends on it. These stakeholders provide knowledge, constraints, or requirements that influence how the system must perform.

They are not the primary client or customer, but they directly affect the system's functionality and accuracy.

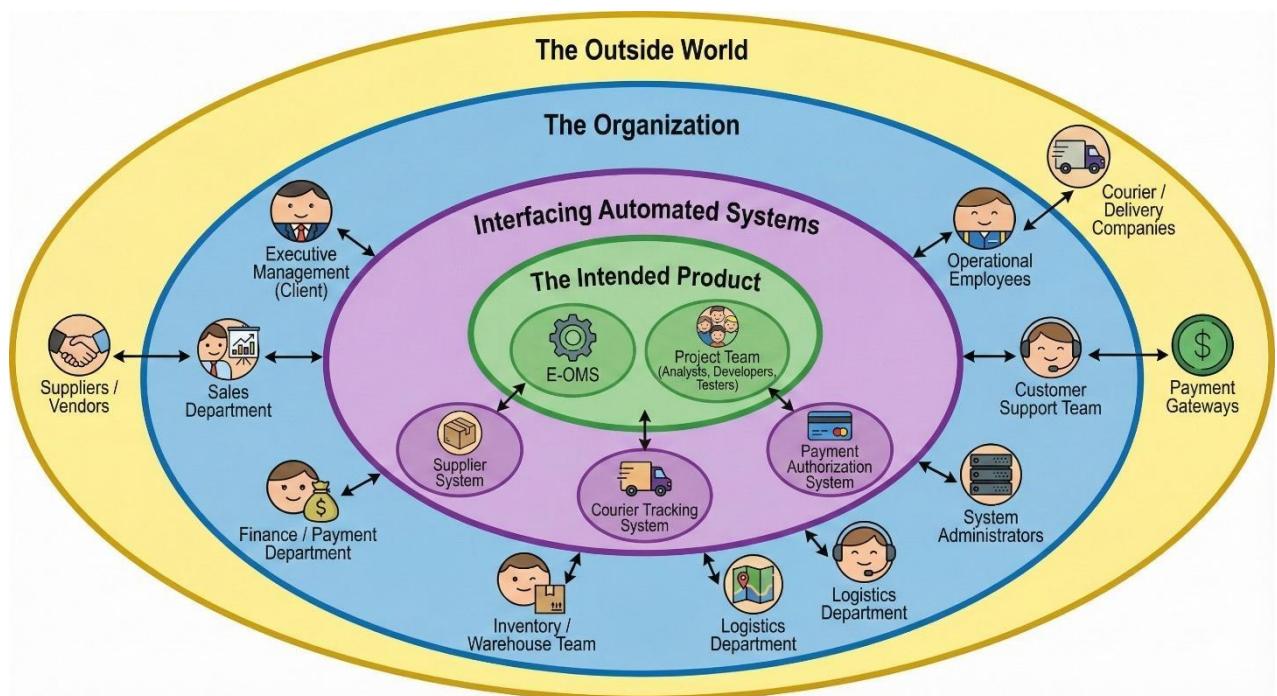
The stakeholders for this project are:

- Sales Department
- Finance / Payment Department
- Inventory / Warehouse Team
- Logistics Department
- Customer Support Team
- Suppliers / Vendors
- Payment Gateways
- Courier / Delivery Companies
- Operational Employees (Sales, Finance, Inventory, Logistics, Support)
- System Administrators

### Motivation

These stakeholders provide essential knowledge about order verification, payment authorization, stock management, shipment planning, and customer support processes. Their input ensures that the system accurately supports the real workflows of the organization and integrates correctly with external systems.

## Form



## 2d. The Hands-On Users of the Product

### Content

#### 1. Customers (End Users)

**User role:** Browse catalog, place orders, make payments, track shipments, submit returns.

**Subject matter experience:** Novice — general users purchasing products online.

**Technological experience:** Mixed; most are journeyman with common web/mobile interfaces.

**Other characteristics:**

- Physically diverse, broad age range
- Prefer simple interfaces and clear instructions
- Often mobile-first users
- Located anywhere geographically
- Varying English/Arabic proficiency

#### 2. Sales Staff

**User role:** Validate orders, check customer data, confirm product details.

**Subject matter experience:** Journeyman — familiar with order processes and customer data.

**Technological experience:** Journeyman — experience with sales/order entry tools.

**Other characteristics:**

- Office-based
- Task-driven, accuracy-focused
- Need fast screens and reliable data

#### 3. Finance / Payment Staff

**User role:** Process payments, manage invoices, verify payment gateway responses.

**Subject matter experience:** Master — strong understanding of financial workflows.

**Technological experience:** Journeyman — familiar with payment systems and authorization tools.

**Other characteristics:**

- Security-focused

- Must comply with financial standards (S1)
- Comfortable with structured data

#### **4. Inventory / Warehouse Staff**

**User role:** Manage stock levels, reserve items, update product catalog, trigger replenishment.

**Subject matter experience:** Master — deep knowledge of stock handling and product catalog rules.

**Technological experience:** Journeyman — familiar with basic inventory systems.

**Other characteristics:**

- Often working between office and warehouse
- Need immediate system responsiveness (real-time inventory)
- Accuracy is critical

#### **5. Logistics Staff**

**User role:** Plan shipments, assign couriers, generate tracking numbers, update delivery status.

**Subject matter experience:** Journeyman — strong logistics and routing knowledge.

**Technological experience:** Journeyman — used to handling courier platforms and manifests.

**Other characteristics:**

- Time-sensitive environment
- Need clear shipment workflows
- Coordinate with external systems (couriers)

#### **6. Customer Support Agents**

**User role:** Handle complaints, order issues, and returns/refunds workflow.

**Subject matter experience:** Journeyman — familiar with issue tracking and service processes.

**Technological experience:** Journeyman — used to CRM/issue-tracking tools.

**Other characteristics:**

- High communication load
- Need complete order history and fast lookup

- Must handle customers calmly and efficiently

## 7. System Administrators

**User role:** Manage user accounts, access permissions, and system configuration.

**Subject matter experience:** Master — deep understanding of system structure and policies.

**Technological experience:** Master — skilled in administrative and security tools.

**Other characteristics:**

- Security-critical responsibilities
- Must monitor system integrity and logs
- Usually central office based

### Motivation

The hands-on users determine how the E-OMS must function in daily operations. Their responsibilities—placing orders, verifying details, processing payments, managing stock, planning shipments, handling returns, and administering the system—directly shape the workflows, screens, and interactions the system must provide. The system has to accommodate the working patterns and capabilities of these users so they can complete their tasks accurately and without delay.

### Examples

- The hands-on users of the E-OMS come from a wide range of operational and public environments. They include:
  - Customers using the system over the Internet to browse products, place orders, make payments, track shipments, and initiate returns.
  - Clerical staff in Sales who verify order details and interact with customer/product information during order processing.
  - Warehouse and stock personnel who update inventory levels, manage catalog entries, and handle replenishment tasks.
  - Finance staff working with payment authorization systems, invoices, and sensitive transaction data.

- Logistics workers who plan shipments, schedule couriers, and work with external delivery systems.
- Customer Support agents who manage refunds, complaints, and telephone/email inquiries using complete order histories.
- System Administrators, trained technical users responsible for access control and system configuration.
- These examples reflect the range of real operational and public users who depend on the E-OMS to perform their daily activities.

**Form**

User Role	Representative Name	Subject Matter Experience	Technological Experience	Other Characteristics
<b>Customer (End User)</b>	General Public	Novice	Mixed (mostly journeyman with web/mobile use)	Wide age range, mobile-first, varying language proficiency, remote/online users
<b>Sales Staff</b>	Ali Tamer	Journeyman	Journeyman	Office-based, accuracy-focused, requires fast validation screens
<b>Finance / Payment Staff</b>	Abdelrahman Ahmed	Master	Journeyman	Security-focused, works with sensitive data, structured workflows
<b>Inventory / Warehouse Staff</b>	Omar Mohammed	Master	Journeyman	Works between warehouse/office, needs real-time updates, high precision
<b>Logistics Staff</b>	Kirillos Raafat	Journeyman	Journeyman	Time-sensitive tasks, interacts with courier systems, route planning needs
<b>Customer Support Agents</b>	Abdelrahman Amr	Journeyman	Journeyman	High communication load, needs complete order histories, service-oriented
<b>System Administrators</b>	(Admin Team)	Master	Master	Security-critical tasks, system oversight, access management

## 2e. Personas

### Content

Persona 1 — Youssef Hamouda

Age: 29

Job: Junior Web Developer

Family: Single, lives with parents in Alexandria

Hobbies: Gaming, football, building PC rigs

Where he lives: Tanta

Favourite food: Fried chicken

Favourite music: Arabic rap

Likes: Fast apps, clean UI, quick delivery times

Dislikes: Slow websites, unclear tracking information

Attitude to technology: Very comfortable; expects smooth, responsive interactions

Attitude to money: Careful with spending; tracks prices and deals

Holidays: Goes to Dahab every summer with friends

Characteristics influencing use:

Mobile-first online shopper

Impatient with delays during checkout

Reads quickly but skims instructions

Expects order tracking to instantly reflect status

Likely to abandon the cart if a page hangs for more than a few seconds

Persona 2 — Salma Hassan

Age: 34

Job: Inventory Coordinator at E-Market

Family: Married, one daughter

Hobbies: Cooking, organizing home spaces, listening to podcasts

Where she lives: Nasr City, Cairo

Favourite food: Molokhia with rice

Favourite music: Soft pop

Likes: Clear dashboards, accurate stock numbers

Dislikes: Data inconsistencies, manual double-entry

Attitude to technology: Comfortable but not technical; prefers

simple tools

Attitude to money: Budget-focused, avoids unnecessary spending

Holidays: Visits family in Mansoura during Eid

Characteristics influencing use:

Needs real-time stock updates

Accuracy-oriented and detail-focused

Works between warehouse and office, often using a laptop or tablet

Wants minimal steps for adding or editing catalog items

Errors or slow updates directly affect her workload

### Persona 3 — Kareem Abdelrahman

Age: 40

Job: Logistics Planner

Family: Married, two sons in school

Hobbies: Fishing, reading travel blogs

Where he lives: October City, Giza

Favourite food: Grilled fish

Favourite music: Old Egyptian classics

Likes: Predictable schedules, organized workflows

Dislikes: Missing courier updates, unclear delivery data

Attitude to technology: Journeyman; uses systems well but dislikes complex interfaces

Attitude to money: Practical; spends mainly on family activities

Holidays: Takes the family to Ain Sokhna twice a year

Characteristics influencing use:

Needs shipment planning tools to be extremely clear

Works under time pressure when assigning couriers

Relies on accurate customer address formats

Prefers step-by-step screens rather than complex dashboards

### Motivation

Among the three personas, Youssef (Customer) and Salma (Inventory Coordinator) carry the highest design priority.

Customers drive revenue and expect a smooth online experience; failing to satisfy them leads directly to abandoned orders and loss of business. Inventory coordination is critical because inaccurate

or slow stock updates disrupt order fulfillment and affect every other department.

Kareem (Logistics Planner) is also important, but his impact is operational rather than customer-facing. Delays in his workflow cause internal inefficiencies, not immediate loss of customers.

No persona in this project is considered “no-impact,” but customer-facing behavior has the strongest effect on business performance. Therefore, usability and responsiveness for the customer persona take precedence, followed by accuracy and workflow clarity for the internal operational personas.

Form

User Role	Representative Name	Subject Matter Experience	Technological Experience	Other Characteristics	Importance Rating
Customer (End User)	Youssef Saleh	Novice	Mixed	Wide age range, mobile-first, varying language proficiency	High
Sales Staff	Ali Tamer	J Journeyman	J Journeyman	Accuracy-focused, office-based, requires fast validation	Medium
Finance / Payment Staff	Abdelrahman Ahmed	Master	J Journeyman	Security-focused, handles sensitive data	Medium
Inventory / Warehouse Staff	Salma Hassan	Master	J Journeyman	Needs real-time stock updates, works between warehouse/office	High
Logistics Staff	Kareem Abdelrahman	J Journeyman	J Journeyman	Time-sensitive planning, relies on external systems	Medium
Customer Support Agents	Abdelrahman Amr	J Journeyman	J Journeyman	High communication load, needs full order histories	Low
System Administrators	(Admin Team)	Master	Master	Security and access control responsibility	Medium

## 2f. Priorities Assigned to Users

### Content

#### Key Users

Users whose correct experience and performance are critical to the success of the E-OMS:

Customers (End Users) — Their satisfaction directly affects revenue, order completion, and repeat business.

**Inventory / Warehouse Staff** — Their accuracy determines stock reliability, order fulfillment, and overall system integrity.  
(High impact supported by Deliverable 2: inventory accuracy and customer experience are central goals.)

Estimated proportion of overall consideration: ~50%  
(High design weight due to business impact.)

### Secondary Users

Users who rely on the system daily but whose preferences carry less business risk if unmet:

Sales Staff

Finance / Payment Staff

Logistics Staff

System Administrators

Their roles shape workflow correctness and efficiency, but they do not directly influence customer retention.

Estimated proportion of overall consideration: ~40%

### Unimportant Users

Users who interact with the system occasionally or have no influence on long-term success:

**Customer Support Agents** — They use the system to assist customers but do not drive revenue or operational accuracy.

Infrequent or incidental users within the organization.

Estimated proportion of overall consideration: ~10%

## Motivation

The users with the highest impact on E-OMS are customers and inventory staff, so their requirements take precedence. Sales, finance, logistics, and system administrators have medium impact. Customer support agents have low impact on long-term product success, so their requirements receive lower priority.

Form

User Role	Representative Name	Subject Matter Experience	Technological Experience	Other Characteristics	Importance Rating
Customer (End User)	Youssef Saleh	Novice	Mixed	Wide age range, mobile-first, varying language proficiency	Key User
Inventory / Warehouse Staff	Salma Hassan	Master	J Journeyman	Needs real-time stock updates, accuracy-focused, moves between warehouse and office	Key User
Sales Staff	Ali Tamer	J Journeyman	J Journeyman	Accuracy-driven, office-based, interacts heavily with customer/product data	Secondary User
Finance / Payment Staff	Abdelrahman Ahmed	Master	J Journeyman	Works with secure payment flows, handles financial data	Secondary User
Logistics Staff	Kareem Abdelrahman	J Journeyman	J Journeyman	Time-sensitive planning, depends on courier integrations	Secondary User
System Administrators	(Admin Team)	Master	Master	Responsible for access control,	Secondary User

				system stability, audit logs	
Customer Support Agents	Abdelrahman Amr	Journeyman	J Journeyman	Support-focused, uses system for inquiries, returns, and issue tracking	Unimportant User

## 2g. User Participation

### Content

Participation from all hands-on users is required to define accurate workflows, validate system behavior, and confirm usability expectations. Each user group contributes knowledge specific to its operational responsibilities, and their input will be used to refine both functional and non-functional requirements. The table below lists the users who will participate, the contributions expected from them, and the time each group will commit during requirements elicitation.

User Role	Expected Contribution	Minimum Participation Time
Customers (End Users)	<ul style="list-style-type: none"> <li>Validate usability of product catalog, checkout flow, order tracking, and returns interface</li> <li>Confirm expectations for delivery timelines and return policies</li> <li>Provide feedback on mobile responsiveness and notification clarity</li> </ul>	2–3 hours total (e.g., 1 usability test session + short survey)
Inventory / Warehouse Staff	<ul style="list-style-type: none"> <li>Define real-time stock reservation logic</li> <li>Clarify product catalog management workflow (add/update/delete)</li> <li>Specify triggers and data fields for supplier replenishment requests</li> </ul>	4–6 hours total (e.g., 1 workshop + 1 validation session)
Sales Staff	<ul style="list-style-type: none"> <li>Detail automated order verification rules</li> <li>Confirm data validation criteria (customer info, product availability)</li> <li>Define handoff conditions to Finance module</li> </ul>	2–3 hours
Finance / Payment Staff	<ul style="list-style-type: none"> <li>Specify secure payment processing requirements</li> <li>Define integration</li> </ul>	3–4 hours

	<p>expectations with payment gateways</p> <ul style="list-style-type: none"> <li>Clarify handling of success/failure scenarios and compliance with security standard S1</li> </ul>	
Logistics Staff	<ul style="list-style-type: none"> <li>Describe shipment planning steps</li> <li>Define courier selection criteria and tracking ID generation logic</li> <li>Validate estimated delivery calculation rules</li> </ul>	3–4 hours
Customer Support Agents	<ul style="list-style-type: none"> <li>Identify required data fields for resolving returns and complaints</li> <li>Confirm need for unified view of order history and return status</li> <li>Validate issue-tracking workflow</li> </ul>	2 hours
System Administrators	<ul style="list-style-type: none"> <li>Define role-based permission structure</li> <li>Specify audit logging and user management requirements</li> <li>Clarify security enforcement mechanisms</li> </ul>	2–3 hours
Managers / Executives	<ul style="list-style-type: none"> <li>Prioritize KPIs and report content</li> <li>Approve system goals,</li> </ul>	2 hours (primarily for review and sign-

	<p>scope boundaries, and success metrics</p> <ul style="list-style-type: none"> <li>• Validate data integration needs across departments</li> </ul>	off)
--	---	------

## Motivation

All contributions will be gathered through structured workshops, prototype walkthroughs, targeted interviews, and requirement review sessions. Sessions will be scheduled in advance to minimize disruption to daily operations.

## Form

Without committed user involvement—especially from Key Users (Customers and Inventory Staff)—the risk of building a system that fails to reflect real workflows or user expectations increases significantly. Clear upfront commitment of user time ensures that functional and non-functional requirements are grounded in actual business context, not assumptions.

## 2h. Maintenance Users and Service Technicians

### Content

The E-OMS will be maintained by technical personnel responsible for system stability, configuration, deployment, database integrity, monitoring, and long-term support. These users require elevated access privileges and tools that allow them to perform system-level tasks, apply changes, and ensure continuous operation. The table below lists the maintenance roles, their expertise, and key characteristics.

User Role	Representative Name	Subject Matter Experience	Technological Experience	Other Characteristics
System Administrator	Admin Team	Master	Master	<ul style="list-style-type: none"> <li>Manages system configuration and user accounts</li> <li>Enforces security policies (S1)</li> <li>Monitors logs, permissions, and performance</li> <li>Performs backups, patching, and routine maintenance</li> <li>Works in central IT with on-call responsibilities</li> </ul>
IT Infrastructure Engineer	Internal IT Team	Master	Master	<ul style="list-style-type: none"> <li>Manages hosting environment and deployments</li> <li>Maintains servers, network connectivity, and integration endpoints</li> <li>Ensures uptime, scalability, and recovery readiness</li> <li>Supports system updates without downtime</li> </ul>
Database Administrator (DBA)	Internal IT Team	Master	Master	<ul style="list-style-type: none"> <li>Oversees database performance and backups</li> <li>Manages schema changes and query optimization</li> <li>Ensures data integrity and compliance with retention</li> </ul>

				<p>policies</p> <ul style="list-style-type: none"> <li>• May support multiple enterprise systems</li> </ul>
--	--	--	--	---

## Motivation

The system requires dedicated maintenance users because reliability, security, scalability, and integration stability depend on their work. Their responsibilities directly influence non-functional requirements such as uptime, recovery time, data protection, and infrastructure performance.

## Form

User Role	Representative Name	Subject Matter Experience	Technological Experience	Other Characteristics	Importance Rating
<b>Customer (End User)</b>	Youssef Saleh	Novice	Mixed	Wide age range, mobile-first, varying language proficiency	<b>Key User</b>
<b>Inventory / Warehouse Staff</b>	Salma Hassan	Master	Journeyman	Accuracy-focused; requires real-time stock visibility; works between warehouse & office	<b>Key User</b>
<b>Sales Staff</b>	Ali Tamer	Journeyman	Journeyman	Office-based; accuracy-driven; heavy interaction with customer/product data	<b>Secondary User</b>
<b>Finance / Payment Staff</b>	Abdelrahman Ahmed	Master	Journeyman	Handles secure transactions; works with financial data; compliance-focused	<b>Secondary User</b>
<b>Logistics Staff</b>	Kareem	Journeyman	Journeyman	Time-sensitive	<b>Secondary</b>

User Role	Representative Name	Subject Matter Experience	Technological Experience	Other Characteristics	Importance Rating
	Abdelrahman	n		tasks; depends on courier integrations and accurate address data	User
System Administrators	Admin Team	Master	Master	Manages access control, system settings, logging, and security enforcement	Secondary User
Customer Support Agents	Abdelrahman Amr	Journeymann	Journeyman	Service-oriented; manages returns/complaints; needs full order history	Unimportant User
IT Infrastructure Engineer	Internal IT Team	Master	Master	Maintains hosting environment, deployments, integrations, and uptime	Maintenance User
Database Administrator (DBA)	Internal IT Team	Master	Master	Ensures data integrity, backups, schema changes, and database performance	Maintenance User

### 3. Naming Conventions and Terminology

3a. Definitions of All Terms, Including Acronyms, Used by Stakeholders Involved in the Project

#### Content

Term / Acronym	Definition
Customer	A person who browses the E-Market online store, places orders, makes payments, tracks shipments, and submits return requests.
Order	A purchase request created by a customer and processed through the E-OMS.
Order Verification	Sales staff review customer details, product availability, and pricing before approval.
Payment Authorization	The process where the payment gateway approves or rejects a transaction.
Payment Gateway	An external service (e.g., bank processor) integrated for secure online payments.
Inventory	All products available for sale, managed by the Inventory/Warehouse staff.
Stock Reservation	Temporary deduction of product quantities once a customer initiates an order.
Replenishment	The process of requesting or receiving new stock from suppliers when levels fall below thresholds.
Catalog	The list of products shown to customers with descriptions, categories, prices, and availability.
Logistics	The workflow involving shipment planning, courier assignment, and tracking updates.
Courier	An external delivery company responsible for transporting orders to customers.
Tracking Number	A unique identifier used to track the progress of an order shipment.
Return Request	A customer-submitted request to return an item and initiate a refund.
Refund	Payment returned to the customer after approval of a return request.
E-OMS (E-Commerce Order Management System)	The system being developed to manage the full order lifecycle for E-Market.
KPI (Key Performance Indicator)	A metric used by management to monitor operational and service performance.
S1 (Security Requirement S1)	A requirement defining standards for protecting customer data and ensuring secure payment operations.
System Administrator	IT personnel who manage user permissions, system configuration, logs, and security.

Term / Acronym	Definition
IT Infrastructure Engineer	Technical staff responsible for deployment, hosting environment, uptime, integrations, and scaling.
DBA (Database Administrator)	Technical staff responsible for database integrity, performance, backups, and schema changes.
API (Application Programming Interface)	A technical interface allowing communication between E-OMS and external systems such as couriers, suppliers, or payment gateways.

### Motivation

A shared glossary ensures that all stakeholders use consistent terminology when describing requirements, workflows, and system functions. Having precise definitions prevents miscommunication across departments and supports accurate modeling and documentation throughout the project.

### Form

The glossary is presented as a structured list of terms and acronyms, each with a clear definition, and is used as the reference vocabulary for all project stakeholders.

## 4. Relevant Facts and Assumptions

### 4a. Relevant Facts

#### Content

1. **E-Market currently uses disconnected tools**, leading to data duplication, communication gaps, and stock errors.
2. **The E-OMS depends on external partners**—payment gateways, suppliers, and courier companies—for core operations such as payment authorization, replenishment, and shipment tracking. These external systems are not under E-OMS control.
3. **If products are out of stock, replenishment requests must be sent to suppliers**, and supplier response times affect availability.
4. **Shipment planning relies on courier systems**, and delays or failures in those systems cause shipment planning issues.
5. **Order tracking updates originate from courier companies**, and their response intervals vary.

6. **The accuracy of inventory data depends on warehouse staff operations**, including stock updates and catalog maintenance.
7. **Real-time notifications (email/SMS)** are required for every order stage, and these depend on external messaging services.
8. **Returns and refunds workflows are managed internally by Customer Support**, and their effectiveness depends on complete and accurate order history data.
9. **The system must generate a unique Order ID automatically** for every placed order.
10. **Order verification, payment processing, inventory reservation, and logistics planning** each involve different departments, requiring accurate data synchronisation between them.

### Motivation

These facts describe real conditions that influence how the E-OMS must operate. They provide background understanding of the environment in which requirements are interpreted and help clarify dependencies that affect system behavior.

## 4b. Business Rules

### Content

The following business rules govern how E-Market conducts order processing, payment handling, inventory control, replenishment, shipment planning, and returns. These rules shape the requirements for the E-OMS and define behaviors that must always be followed in the business workflow.

#### **1. Every placed order must generate a unique Order ID.**

This rule is stated in the Order Validation and Processing description where the system “automatically records and generates a unique Order ID upon placement.”

#### **2. If payment authorization fails, the order is automatically cancelled.**

The Financial Processing section states that after sending payment details to the gateway, the system must “confirm payment (or trigger automatic cancellation upon failure).”

**3. Only authorized inventory staff may add, update, or delete products in the catalog.**

Catalog Management is restricted to “authorized inventory staff” who can add/update/delete product entries.

**4. If a requested item is out of stock, a replenishment request must be sent to suppliers.**

The Inventory and Logistics scope explicitly states: “If products are out of stock, a replenishment request is sent to suppliers.”

**5. Inventory must always be updated in real time when orders are placed or modified.**

The system requires “real-time inventory tracking and reservation,” meaning stock quantities must reflect live customer and internal activity.

**6. Shipment planning must assign a courier and generate a tracking number.**

The Logistics description states the system “plans shipment ... selects a courier, and assigns a tracking number and estimated delivery date.”

**7. Courier systems determine the timing of tracking updates.**

Interview notes show that shipment delays often come from “external system integration failures” with courier systems.

**8. Returns and refunds must be handled by Customer Support using complete order history data.**

The Customer Support role manages the “returns and refunds workflow,” requiring full and accurate order information.

**9. Notification messages must be sent to customers for each stage of the order lifecycle.**

Customer Interaction requires “Email/SMS notifications for every stage of the order process.”

**10. Replenishment requests are triggered only after Inventory Verification determines stock is low.**

Technical Report 2 shows that *Send Replenishment Request* is included when the *Reserve Inventory* process detects low stock.

## Motivation

These business rules define how E-Market’s operations must function and determine the system behaviors that E-OMS must support. They serve as triggers for requirements and ensure

the system aligns with established business policies across Sales, Finance, Inventory, Logistics, and Customer Support requirements.

#### Form

Each business rule is presented as a written statement describing the rule, the business reason behind it, and its authority derived directly from documented processes and stakeholder-defined workflows.

## 4c. Assumptions

### Content

- External systems (suppliers, couriers, payment gateways) are available and functional when E-OMS sends requests.**

This assumption appears in multiple interaction workflows where E-OMS sends replenishment requests, shipment manifests, and payment authorization requests. The documents state integration steps but do not include fallback or offline behavior.

- Warehouse staff will maintain accurate physical stock counts that align with digital inventory updates.**

Inventory accuracy issues are identified as a core problem, and the system depends on real-time and correct updates from the warehouse team.

- Customer delivery addresses are assumed to be valid and correctly formatted when provided.**  
Shipment planning delays were linked to address-related issues, implying the system assumes input is correct unless validation fails.

- Messaging channels (Email/SMS) used for customer notifications are available when the system triggers them.**

Customer Interaction includes mandatory notifications for each order stage, but the system does not control the messaging infrastructure.

- Each department (Sales, Finance, Inventory, Logistics, Customer Support) continues to follow the documented workflow.**

The use cases and user stories assume that operational teams continue to perform their roles as defined.

- The E-OMS is not responsible for marketing functions, recommendations, HR tasks, or accounting beyond payment recording.**

These exclusions are explicitly stated as project

limitations.

7. **Executives will use system-generated reports as the authoritative source for KPIs and decision-making.**  
The interviews assume management relies on the system for integrated analytics.
8. **System administrators will be available to manage access control, logs, and configuration changes.**  
Admin responsibilities are listed as ongoing operational duties of the system environment.
9. **Courier systems will acknowledge received manifests when shipments are assigned.**  
The Courier user story describes the expected “acknowledgment” but does not define handling of missing acknowledgments.
10. **Payment authorization responses will follow the gateway's documented format and status codes.**  
The payment user story assumes successful interpretation of gateway responses.

## Motivation

Stating these assumptions ensures that all stakeholders understand the operational expectations built into the analysis. These assumptions inform system behavior and help identify areas where requirements or constraints may need to be added later.

## Form

Each assumption is written as a clear statement followed by the impact on system interpretation if the assumption is false.

## 5. The Scope of the Work

### 5a. The Current Situation

#### Content

E-Market currently manages the order-to-delivery process using fragmented and partially manual tools. The lack of an integrated system results in data duplication, communication delays, and inconsistent stock information. The existing workflow depends on separate departmental processes that do not automatically synchronize with one another.

#### Current Business Processes

##### 1. Order Placement (Customer)

Customers browse products and place orders online, but downstream steps depend on manual or poorly connected tools.

##### 2. Order Verification (Sales Department)

Sales staff manually confirm customer details, pricing, and product availability before forwarding the order to Finance. This process is slowed by communication gaps with other departments.

##### 3. Payment Processing (Finance Department)

Finance uses external payment gateways to authorize payments. Transaction outcomes must be manually checked before confirming the order status.

##### 4. Inventory Management (Inventory/Warehouse Team)

Inventory staff update product quantities and catalog entries manually.

Real-time stock availability is often inaccurate, leading to stock errors and fulfillment issues.

##### 5. Supplier Replenishment (Suppliers/Vendors)

When items are out of stock, replenishment requests are sent to suppliers, but this process is not automated and depends on manual follow-up.

##### 6. Shipment Planning (Logistics Department)

Logistics staff plan shipments, assign couriers, generate tracking IDs, and prepare manifests.

Integration with courier systems is inconsistent, and missing data often delays planning.

## **7. Delivery and Tracking (Couriers)**

Tracking updates depend on courier systems that are external and not synchronized in real time, causing inconsistent customer information.

## **8. Returns and Refunds (Customer Support)**

Customer Support manages complaints, returns, and refunds manually.

They frequently lack complete or accurate order histories, slowing resolution time.

## **9. Reporting and Analytics (Executives)**

Management receives reports compiled from multiple departments, with no unified dashboard.

They lack real-time integrated KPIs for decision-making.

### **Overall Current State (From Problem Statement)**

- Data duplication is common across Sales, Inventory, and Finance.
- Communication gaps cause delays and cross-department confusion.
- Stock errors occur due to manual updates and disconnected systems.
- Manual tasks significantly slow down the order lifecycle.

This is the complete and accurate representation of the existing business workflow.

### **Motivation**

Understanding the current system highlights operational inefficiencies—manual coordination, inconsistent data, and delays—that the E-OMS is intended to eliminate. This analysis serves as the baseline for identifying where automation, integration, and workflow restructuring will provide the greatest benefit.

### **Form**

The current situation is described as a structured workflow summary based on stakeholder interviews, documented problems, and departmental processes. Further modeling can be represented with activity diagrams, swimlane diagrams, or process maps if required in the design phase.

## 5b. The Context of the Work

### Content

The work supported by the E-OMS involves the full lifecycle of an online order and the coordination between internal departments and external partners. The context includes all people, systems, and organizations that exchange information with the work. The boundaries of the work are defined by the business activities carried out by E-Market, while the adjacent systems represent external actors and systems that provide or receive information.

### Work Context

#### Internal Work (Inside E-Market)

The internal work includes all processes carried out by E-Market departments:

- **Sales Department** – verifies customer data and order details.
- **Finance Department** – processes payments through the payment gateway.
- **Inventory/Warehouse** – manages stock, catalog updates, and replenishment.
- **Logistics Department** – plans shipments, selects couriers, generates tracking IDs.
- **Customer Support** – manages returns, refunds, and issue tracking.
- **Executives/Managers** – consume summary reports and KPIs.
- **System Administrators** – manage permissions and system access.

These departments collectively perform the work that the E-OMS supports.

#### Adjacent Systems (Outside E-Market)

The following external systems interact with the work and exchange data with E-Market:

- **Supplier System** – receives replenishment requests when stock falls below threshold.

- **Payment Gateway** – receives payment authorization requests and returns approval/decline codes.
- **Courier/Delivery Company System** – receives shipment manifests and returns acknowledgment and tracking updates.
- **Customers (End Users)** – browse products, place orders, track shipments, and submit returns.

These adjacent systems provide the inputs and outputs that cross the work boundary.

#### Motivation

Defining the work context establishes the boundary of investigation for requirements discovery. Understanding which external actors and systems exchange information with E-Market ensures that the E-OMS fits correctly within its operational environment and supports all necessary interactions.

#### Form

##### Work Context Inputs and Outputs

Adjacent System	Input to E-Market	Output from E-Market
Customer	Order details, delivery information, return requests	Order confirmations, status updates, tracking information
Supplier System	—	Replenishment requests
Payment Gateway	Payment authorization response	Secure payment authorization request

Adjacent System	Input to E-Market	Output from E-Market
Courier System	Shipment acknowledgement, tracking updates	Shipment manifest with customer address and tracking ID

## 5c. Work Partitioning

### Content

#### Business Event List

**Table 1 — Events With Inputs and Outputs**

Event #	Event Name	Input (Trigger)	Output
1	Customer submits an order	Order details from Customer	Order confirmation to Customer
2	Sales verifies order	Order awaiting verification	Approved order to Finance or cancellation to Customer
3	Customer submits payment	Payment info from Customer	Authorization request to Payment Gateway
4	Payment Gateway returns result	Authorization code	Payment confirmation or cancellation to Customer
5	Inventory reserves	Paid order	Updated stock levels

	items		
<b>6</b>	Inventory detects low stock	Low-stock condition	Replenishment request to Supplier System
<b>7</b>	Catalog update	Catalog change from Inventory Staff	Updated product info to Customer
<b>8</b>	Order ready for shipment	Packaged order	Shipment planning request to Logistics
<b>9</b>	Logistics plans shipment	Packaging confirmation	Shipment manifest to Courier System
<b>10</b>	Courier acknowledges shipment	Shipment acknowledgment	Tracking updates to Customer
<b>11</b>	Customer requests tracking	Tracking ID	Shipment status
<b>12</b>	Customer submits return request	Return request	Return acknowledgement
<b>13</b>	Support retrieves order history	Customer ID or Order ID	Order history to Support
<b>14</b>	Time to generate reports	Scheduled interval	KPI reports to Executives

**Table 2 — Event Summaries (Mini-Scenarios)**

Event #	Summary of BUC
1	Create a new order and begin verification.
2	Verify customer details, pricing, and product availability.
3	Forward payment details for authorization.
4	Approve purchase or cancel order based on gateway response.
5	Deduct and reserve stock for the order.
6	Trigger supplier replenishment when stock falls below threshold.
7	Add, edit, or delete catalog items.
8	Notify Logistics that a packaged order is ready.
9	Assign courier, generate tracking ID, and prepare manifest.
10	Activate tracking after courier acknowledgment.
11	Retrieve shipment status for the customer.
12	Register a return and notify Customer Support.
13	Display full order history for issue resolution.
14	Compile KPIs and generate management reports.

## Motivation

Partitioning the work into business events identifies all external triggers and internal state changes that the E-OMS must support. Each business event forms the basis of a Business Use Case (BUC),

enabling focused analysis and ensuring that all interactions across the work boundary are understood.

## 5d. Specifying a Business Use Case (BUC)

### Content

#### **Business Use Case for Event #1 — Customer Submits an Order**

##### **Content (BUC Specification)**

###### **Event Name**

**Customer submits an order**

###### **Input (Trigger)**

Order details from **Customer**

###### **Output**

Order confirmation to **Customer**

#### **BUC Scenario — How the Business Responds**

1. The customer selects products from the online catalog and proceeds to checkout.
2. The customer provides personal details including name, address, and contact information.
3. The customer reviews the order summary containing items, quantities, prices, and delivery information.
4. The customer confirms the order by submitting it.
5. The business records the submitted order and assigns it a unique Order ID.
6. The business stores all order line items, total price, and delivery details.
7. The system sends an order confirmation message to the customer containing the Order ID and the summary of items ordered.
8. The order is then passed to the Sales Department for verification in the next business event.

## Motivation

This BUC provides a clear and complete description of how the business responds when a customer submits an order. Understanding this response allows the project team to determine which parts of the workflow must be automated by the E-OMS and which parts remain business tasks.

## Form

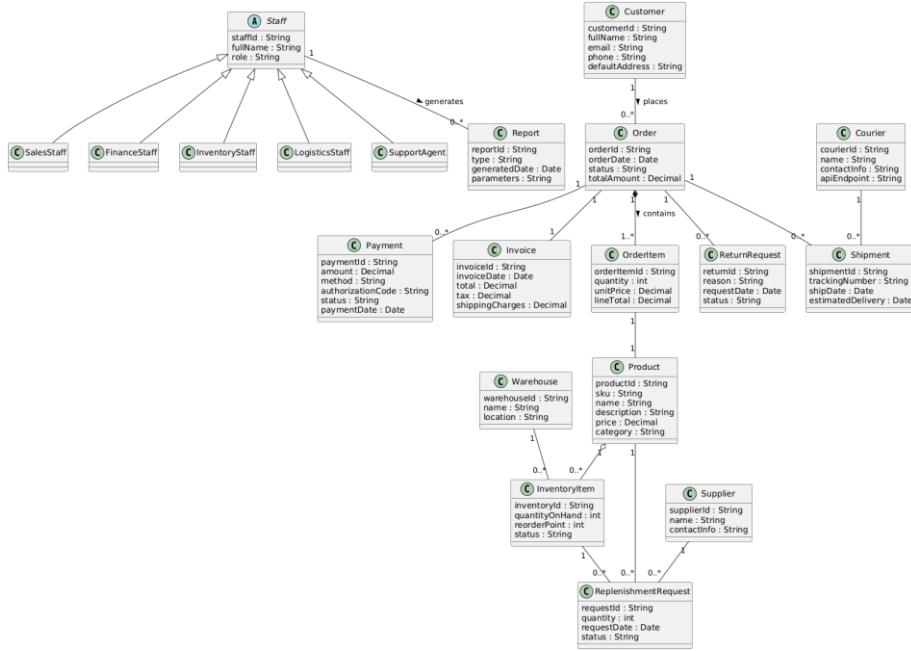
This BUC is presented as a structured scenario, consistent with the inputs and outputs defined in the Business Event List.

## 6. Business Data Model and Data Dictionary

### 6a. Business Data Model

#### Content

The business data model identifies the essential business objects used across E-Market's order lifecycle. These objects represent the information created, referenced, updated, or deleted by the business events in Section 6c. The model reflects the subject matter used by Sales, Finance, Inventory, Logistics, Customer Support, Executives, and external partners (Suppliers, Payment Gateway, Courier).



## Motivation

This model clarifies the essential business subject matter handled by E-Market's departments and external partners. It ensures consistency across events, workflows, and requirements, and supports later CRUD analysis, detailed requirements, and the data dictionary.

## Form

The business data model is shown as a UML class model capturing:

- Business classes
- Their attributes (business meaning only)
- Their relationships

## 6b. Data Dictionary

### Content

Name	Content (Business Meaning)	Type
<b>amount</b>	Monetary value recorded for a payment.	Attribute
<b>apiEndpoint</b>	Courier system endpoint used for sending shipment manifest data.	Attribute
<b>authorizationCode</b>	Code returned after payment authorization.	Attribute
<b>category</b>	Product classification used in browsing and filtering.	Attribute
<b>contactInfo (Courier)</b>	Courier contact details used for communication.	Attribute
<b>contactInfo (Supplier)</b>	Information needed to contact the supplier for replenishments.	Attribute
<b>Courier</b>	Delivery provider responsible for transporting shipments to customers.	Class
<b>courierId</b>	Unique identifier for a courier.	Attribute
<b>Customer</b>	Person placing orders and receiving shipments.	Class
<b>customerId</b>	Identifier for retrieving customer information and order history.	Attribute
<b>defaultAddress</b>	Customer's primary delivery location.	Attribute
<b>description</b>	Text describing the product's features.	Attribute
<b>email</b>	Customer email used for communication and notifications.	Attribute

Name	Content (Business Meaning)	Type
<b>estimatedDelivery</b>	Expected delivery date calculated during shipment planning.	Attribute
<b>FinanceStaff</b>	Staff responsible for processing payments and financial tasks.	Class
<b>fullName (Customer)</b>	Customer's full legal name.	Attribute
<b>fullName (Staff)</b>	Staff member's full name.	Attribute
<b>generatedDate</b>	Date a report was generated.	Attribute
<b>InventoryItem</b>	Represents stock levels and inventory status for a product.	Class
<b>inventoryId</b>	Identifier for an inventory item record.	Attribute
<b>Invoice</b>	Document summarizing order charges, taxes, and shipping.	Class
<b>invoiceDate</b>	Date the invoice was issued.	Attribute
<b>invoiceId</b>	Unique identifier for an invoice.	Attribute
<b>lineTotal</b>	Total monetary value for an OrderItem (quantity × unit price).	Attribute
<b>LogisticsStaff</b>	Staff responsible for shipment planning and courier assignment.	Class
<b>method</b>	Payment method selected by the customer.	Attribute
<b>name (Courier)</b>	Courier's name.	Attribute
<b>name (Product)</b>	Product name as shown in the catalog.	Attribute
<b>name (Supplier)</b>	Supplier company name.	Attribute

Name	Content (Business Meaning)	Type
<b>Order</b>	Customer purchase containing order items, payments, and shipments.	Class
<b>orderDate</b>	Date the customer submits the order.	Attribute
<b>orderId</b>	Unique identifier for an order.	Attribute
<b>OrderItem</b>	Line item inside an order linking a product and quantity.	Class
<b>orderItemId</b>	Identifier for an order item.	Attribute
<b>parameters</b>	Criteria used to configure a report.	Attribute
<b>Payment</b>	Record of a payment attempt or completion.	Class
<b>paymentDate</b>	Date the payment was processed.	Attribute
<b>paymentId</b>	Unique identifier for a payment record.	Attribute
<b>phone</b>	Customer phone number used for communication.	Attribute
<b>price</b>	Standard price of a product.	Attribute
<b>Product</b>	Item sold by E-Market and displayed in the catalog.	Class
<b>productId</b>	Unique identifier for a product.	Attribute
<b>quantity (OrderItem)</b>	Number of units for a product in a specific order.	Attribute
<b>quantity (ReplenishmentRequest)</b>	Amount of inventory requested from supplier.	Attribute
<b>quantityOnHand</b>	Current available units of	Attribute

Name	Content (Business Meaning)	Type
	a product in inventory.	
<b>reason</b>	Explanation provided by the customer for a return request.	Attribute
<b>ReplenishmentRequest</b>	Request sent to supplier when inventory is low.	Class
<b>Report</b>	Document generated for operational or management analysis.	Class
<b>reportId</b>	Identifier for a report.	Attribute
<b>requestDate (ReplenishmentRequest)</b>	Date the replenishment request is created.	Attribute
<b>requestDate (ReturnRequest)</b>	Date the customer submitted the return.	Attribute
<b>requestId</b>	Identifier for the replenishment request.	Attribute
<b>returnId</b>	Unique identifier for a return request.	Attribute
<b>ReturnRequest</b>	Customer-initiated request to return an item.	Class
<b>role</b>	Job role of a staff member.	Attribute
<b>SalesStaff</b>	Staff who verify orders and customer information.	Class
<b>shipDate</b>	Date when shipment was dispatched.	Attribute
<b>Shipment</b>	Shipment record containing tracking and delivery details.	Class
<b>shipmentId</b>	Unique identifier for a shipment.	Attribute
<b>shippingCharges</b>	Cost associated with delivery on the invoice.	Attribute
<b>sku</b>	Stock Keeping Unit used	Attribute

Name	Content (Business Meaning)	Type
	to uniquely identify product variations.	
<b>Staff</b>	Abstract class describing shared properties of all staff roles.	Class
<b>status (InventoryItem)</b>	Inventory status (e.g., Available, Out of Stock).	Attribute
<b>status (Order)</b>	Order lifecycle status (e.g., Pending, Verified, Paid).	Attribute
<b>status (Payment)</b>	Payment outcome (e.g., Authorized, Failed).	Attribute
<b>status (ReplenishmentRequest)</b>	Current progress of the replenishment request.	Attribute
<b>status (ReturnRequest)</b>	Processing state of the return request.	Attribute
<b>Supplier</b>	External vendor providing products to E-Market.	Class
<b>supplierId</b>	Supplier identifier.	Attribute
<b>SupportAgent</b>	Staff managing customer issues, returns, and complaints.	Class
<b>tax</b>	Tax amount applied to an invoice.	Attribute
<b>total (Invoice)</b>	Total monetary amount on the invoice.	Attribute
<b>totalAmount (Order)</b>	Order total across all OrderItems.	Attribute
<b>trackingNumber</b>	Identifier used to track a shipment through the courier.	Attribute
<b>unitPrice</b>	Price per unit recorded at the time of order.	Attribute
<b>Warehouse</b>	Physical storage location for inventory.	Class

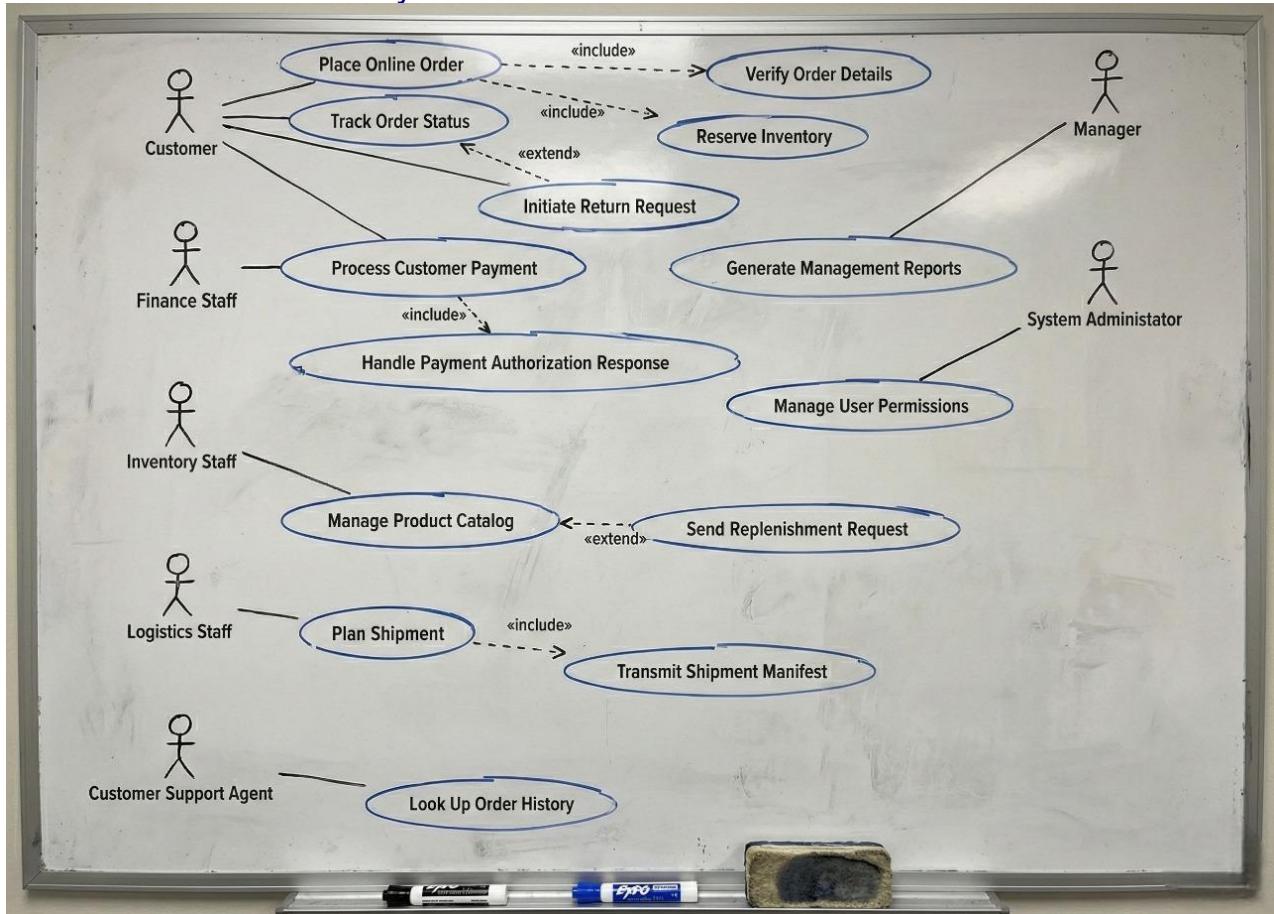
Name	Content (Business Meaning)	Type
<b>warehouseId</b>	Identifier of the warehouse.	Attribute
<b>Admin Relationship: Staff → Report</b>	Staff generate reports.	Relationship
<b>Customer → Order</b>	A customer places one or more orders.	Relationship
<b>Order → OrderItem</b>	An order contains multiple order items.	Relationship
<b>OrderItem → Product</b>	Each order item references a single product.	Relationship
<b>Product → InventoryItem</b>	A product has one or more inventory records.	Relationship
<b>Warehouse → InventoryItem</b>	Inventory is stored in a warehouse.	Relationship
<b>InventoryItem → ReplenishmentRequest</b>	Low stock triggers replenishment requests.	Relationship
<b>Supplier → ReplenishmentRequest</b>	Suppliers receive replenishment requests.	Relationship
<b>Order → Payment</b>	Payments are linked to a specific order.	Relationship
<b>Order → Invoice</b>	Each order generates one invoice.	Relationship
<b>Order → Shipment</b>	Orders may have multiple shipments.	Relationship
<b>Courier → Shipment</b>	Shipments are assigned to a courier.	Relationship
<b>Order → ReturnRequest</b>	Returns are linked to an order.	Relationship

## Form

The data dictionary is maintained as a structured table that lists each business class, its attributes, and the relationships defined in the business data model.

## 7. The Scope of the Product

### 7a. Product Boundary



### 7b. Product Use Case Table

PUC No	PUC Name	Actor(s)	Input & Output
1	Place Online Order	Customer	<p><b>Input:</b> Product selection, delivery &amp; contact details, cart contents</p> <p><b>Output:</b> Unique Order ID, “Processing” status, confirmation notification (email/SMS)</p>
2	Process Customer Payment	Customer, Finance Staff	<p><b>Input:</b> Payment details (tokenized), Order ID</p> <p><b>Output:</b> Payment authorization status (Success/Failure), updated order status (“Awaiting Payment” → “Confirmed” or “Cancelled”)</p>
3	Manage Product Catalog	Inventory Staff	<p><b>Input:</b> Product name, description, price, SKU, stock level</p> <p><b>Output:</b> Updated product catalog (visible to customers), real-time inventory sync</p>
6	Plan Shipment	Logistics Staff	<p><b>Input:</b> Order ID, courier selection, route preference</p> <p><b>Output:</b> Tracking ID, estimated delivery date, shipment manifest sent to Courier System</p>
7	Track Order Status	Customer	<p><b>Input:</b> Tracking ID or Order ID</p> <p><b>Output:</b> Real-time status (e.g., “Shipped”, “In Transit”, “Delivered”), estimated delivery date</p>

8	<b>Initiate Return Request</b>	Customer	<p><b>Input:</b> Order ID, return reason, return window validation</p> <p><b>Output:</b> Return request logged, confirmation notification (email/SMS), status updated in Support module</p>
9	<b>Look Up Order History</b>	Customer Support Agent	<p><b>Input:</b> Customer ID or Order ID</p> <p><b>Output:</b> Full order history, current status of each order, list of active/inactive return requests</p>
10	<b>Generate Management Reports</b>	Manager / Executive	<p><b>Input:</b> Report type, date range, KPI filters</p> <p><b>Output:</b> Integrated analytical report (sales, inventory, finance KPIs)</p>
11	<b>Manage User Permissions</b>	System Administrator or	<p><b>Input:</b> User ID, role, permission set</p> <p><b>Output:</b> Updated access control rules, system log of permission change</p>

---

### 7c. Individual Product Use Cases

#### 1. Place Online Order

Actor: Customer

Goal: Submit a new order with valid product and delivery information.

Preconditions:

- Customer is browsing the product catalog.
- Selected items are in stock.

Main Success Scenario:

1. Customer adds one or more products to the cart.
2. Proceeds to checkout and enters delivery and contact details.
3. System validates product availability, pricing, and customer information.
4. System generates a unique Order ID.
5. Order status is set to “Processing/Verification”.
6. Customer receives an immediate confirmation via Email or SMS.

Alternative/Exception Flows:

- If product data or customer info is invalid or missing, the system displays an error and blocks submission.
- If the system detects a duplicate order attempt, it blocks submission to prevent duplication.

Postconditions:

- Order is recorded in the system but inventory is not yet reserved. Payment has not been processed.

---

## **2. Process Customer Payment**

Actor: Customer, Finance Staff (system role)

Goal: Securely authorize payment and update order status.

Preconditions:

- Order status is “Awaiting Payment”.
- Customer has submitted valid payment details.

Main Success Scenario:

1. System sends a secure request to the integrated Payment Gateway.
2. Gateway returns a “Success” authorization code.
3. System updates order status to “Confirmed”.
4. System triggers automatic inventory reservation.

Exception Flows:

- If payment fails (e.g., declined card), the system cancels the order and notifies the customer.
- If the gateway times out or is unreachable, the system retries once; if it fails again, the order is cancelled.

Postconditions:

- Order is either confirmed (proceeds to fulfillment) or cancelled. All transaction data complies with Security Requirement S1.

---

### **3. Manage Product Catalog**

Actor: Inventory Staff

Goal: Maintain an accurate and up-to-date product catalog.

Preconditions:

- User is logged in with Inventory Staff permissions.

Main Success Scenario:

1. User selects to add, update, or delete a product.
2. Enters or modifies product name, description, price, and SKU.
3. System validates required fields and saves changes.
4. Updated catalog is immediately visible to customers.

Exception Flows:

- If required fields are missing, the system highlights the error and prevents saving.
- If the user attempts to delete a product with pending orders, the system warns and requires confirmation.

Postconditions:

- Product catalog reflects real-time changes. Data is used in order validation and inventory tracking.

---

## **4. Plan Shipment**

Actor: Logistics Staff

Goal: Assign courier, generate tracking ID, and schedule delivery.

Preconditions:

- Order has been packed and marked “Ready for Shipment” by Inventory.

Main Success Scenario:

1. Logistics Staff selects courier and confirms delivery details.
2. System sends shipment manifest to the external Courier System.
3. Courier System returns a unique tracking ID and estimated delivery date.
4. System updates order status to “Shipped” and notifies the customer.

Exception Flows:

- If the courier system is unavailable, the system logs the error and allows manual retry or selection of an alternate courier.

Postconditions:

- Order has a valid tracking ID and delivery estimate. Customer can track the shipment.
- 

## **5. Track Order Status**

Actor: Customer

Goal: View real-time status and delivery estimate of an order.

Preconditions:

- Customer has a valid Order ID or Tracking ID.

Main Success Scenario:

1. Customer enters the ID in the tracking interface.
2. System displays current status (e.g., “Shipped”, “In Transit”, “Delivered”) and estimated delivery date.

Exception Flows:

- If the ID is invalid or not found, the system displays “Order not found”.

Postconditions:

- Customer is informed of the current delivery state.

---

## **6. Initiate Return Request**

Actor: Customer

Goal: Start a return or refund workflow for a delivered item.

Preconditions:

- Order status is “Delivered”.
- Current date is within the allowed return window (e.g., 14 days).

Main Success Scenario:

1. Customer selects the order and submits a return reason.
2. System validates return eligibility.
3. System logs the request in the Customer Support module.
4. Customer receives an automated return confirmation via Email or SMS.

Exception Flows:

- If the return window has expired, the system displays “Return not allowed” and blocks submission.

Postconditions:

- Return request is active in the support queue and visible to agents.

---

## **7. Look Up Order History**

Actor: Customer Support Agent

Goal: Retrieve a customer's complete order and return history.

Preconditions:

- Agent is logged in with support permissions.

Main Success Scenario:

1. Agent enters a Customer ID or Order ID.
2. System displays all historical orders, their current status, and any associated return requests.
3. Agent can update issue status until resolution.

Exception Flows:

- If no matching customer is found, the system returns an empty result (no error).

Postconditions:

- Agent has full context to resolve customer issues efficiently.

---

## **8. Generate Management Reports**

Actor: Manager / Executive

Goal: Access integrated analytical and summary reports.

Preconditions:

- User has executive role permissions.

Main Success Scenario:

1. User selects report type (e.g., Sales, Inventory, Finance).
2. Sets date range and filters.
3. System queries integrated data and generates the report within 3 seconds.

Exception Flows:

- None expected; empty data is handled gracefully with a “No data” message.

Postconditions:

- Report displays synchronized, accurate KPIs from all relevant modules.

---

## **9. Manage User Permissions**

Actor: System Administrator

Goal: Assign or modify access rights for staff roles.

Preconditions:

- Admin is logged in with full privileges.

Main Success Scenario:

1. Admin selects a user or role.
2. Updates permission set (e.g., grants Logistics access to courier APIs).
3. System applies changes immediately and logs the action securely.

Exception Flows:

- If the change violates security policies (e.g., unauthorized permission), the system denies the request and shows “Permission denied”.

Postconditions:

- User can access only functions defined by their updated role. System enforces Security Requirement S1 and data integrity.

---

## 8. Functional Requirements

### 8a. Functional Requirements

#### Content

This section specifies the **atomic functional requirements** of the E-Commerce Order Management System (E-OMS). Each requirement

describes a single system behavior or capability that supports the order lifecycle—from user authentication to shipping, returns, and reporting.

The functional requirements include:

**1. User Registration & Authentication**

The system shall allow customers and staff to create accounts, log in, and access features based on their assigned roles.

**2. Product Catalog Browsing**

The system shall allow customers to view available products, including descriptions, images, prices, and stock availability.

**3. Order Placement**

The system shall allow customers to add items to a cart, enter delivery details, and place an order. A unique Order ID is generated automatically.

**4. Automated Order Verification**

The system shall verify order details, validate customer information, and check product availability. Invalid or incomplete orders shall be cancelled automatically.

**5. Payment Processing**

The system shall securely transmit payment data to the payment gateway and update order status based on the gateway's response.

**6. Real-Time Inventory Management**

The system shall update stock levels in real time, reserve items for paid orders, and prevent overselling.

**7. Low-Stock Replenishment Requests**

The system shall automatically send replenishment requests to suppliers when inventory falls below predefined thresholds.

**8. Shipment Planning & Courier Assignment**

The system shall assign a courier, generate a tracking number, and calculate an estimated delivery date for each confirmed order.

**9. Invoice Generation**

The system shall automatically generate an invoice after payment is confirmed and attach it to the order record.

## **10. Customer Order Tracking**

The system shall allow customers to track order status from placement to delivery using their tracking number.

## **11. Returns & Refunds Workflow**

The system shall support the submission, processing, and tracking of return or refund requests.

## **12. Notifications (Email/SMS)**

The system shall send customers notifications at each major order stage.

## **13. Reporting & Analytics**

The system shall generate performance, sales, operational, and customer satisfaction reports for managers.

## **14. User Role & Access Control**

Administrators shall be able to assign roles, manage permissions, and control staff access to system modules.

### **Motivation**

The purpose of outlining the functional requirements is to clearly define **what the E-OMS must do** in order to support business operations and meet stakeholder expectations. By specifying these functions:

- **Development teams gain clarity** on the system's required behaviors.
- **Stakeholders obtain a shared understanding** of the system's capabilities.
- **Ambiguities and misinterpretations are reduced**, leading to fewer redesigns later.
- **Testers can develop measurable acceptance criteria**, ensuring that each requirement is verifiable.
- **The system can be properly modeled**, enabling the creation of accurate use cases, domain models, sequence diagrams, and process flows.

Above all, the functional requirements ensure that the E-OMS:

- Streamlines and automates the order management lifecycle
- Reduces operational errors caused by the current fragmented system
- Enhances data accuracy and inter-departmental coordination
- Improves customer satisfaction through transparency and timely notifications

## Examples

The following examples illustrate why Functional Requirements are important:

### 1. Ensuring Accurate Order Processing

Without a clearly defined requirement for *Order Verification*, the system could accept incomplete or invalid orders, leading to customer dissatisfaction and operational delays.

### 2. Preventing Inventory Errors

A requirement like *Real-Time Inventory Reservation* prevents overselling. For example, if two customers purchase the last item simultaneously, the system must correctly reserve stock for the first successful payment.

### 3. Reducing Payment Failures and Disputes

By defining *Payment Processing* as a separate functional requirement, the system can ensure that failed payments result in an automatic order cancellation and a notification to the customer.

### 4. Streamlining Logistics

A clearly defined requirement for *Courier Assignment and Tracking Number Generation* ensures that the logistics department receives complete shipping information to avoid delivery delays.

### 5. Improving Staff Efficiency

Requirements like *Role-Based Access Control* make sure each staff member sees only the functions relevant to their department, reducing

confusion and ensuring secure data access.

## 6. Supporting Decision-Making

A requirement for *Reporting & Analytics* ensures that managers receive accurate KPIs, sales summaries, and operational performance data needed for informed decisions

### Fit Criterion

The **Fit Criterion** defines how the successful implementation of a requirement will be measured.

It converts each functional requirement into a **clear, testable condition**, ensuring that system behavior can be objectively verified during testing.

Fit Criteria serve the following purposes:

- Provide **measurable evidence** that a requirement has been fully satisfied.
- Remove ambiguity by specifying **exact conditions** under which the requirement is considered complete.
- Support **system acceptance testing** by giving testers a precise “pass or fail” benchmark.
- Ensure consistency between **requirements, user stories, and acceptance tests**.

A Fit Criterion typically uses a structured format such as:

#### Given – When – Then Format

This format clarifies:

- the *initial state (Given)*,
- the *triggering action (When)*,
- and the *expected measurable outcome (Then)*.

#### Example Fit Criterion

**Given** a new customer order has been submitted,  
**When** the system validates order details,

**Then** the system shall confirm validity and mark the order as “Awaiting Payment”,  
**Or** automatically cancel the order and notify the customer if validation fails.

Fit Criteria are essential to ensuring that every functional requirement is:

- **Testable**
- **Objective**
- **Unambiguous**
- **Consistent across stakeholders**

They provide the basis for designing test cases and determine whether the E-Commerce Order Management System (E-OMS) meets its required functionality.

### Considerations

The **Considerations** section outlines the important factors, constraints, and implications that must be taken into account when interpreting or implementing the functional requirements. These considerations ensure that the requirements are realistic, aligned with project boundaries, and compatible with both technical and business constraints.

Key considerations include:

### **1. System Scope and Boundaries**

Functional requirements must operate strictly within the defined project scope.

The E-OMS excludes marketing systems, HR modules, and accounting features, so requirements involving these areas must not be included or referenced beyond what is necessary for integration with external systems.

### **2. External System Dependencies**

Certain functions depend on external systems such as:

- Payment gateways
- Courier APIs
- Supplier systems

These external systems may introduce delays, data format requirements, or availability issues. Functional requirements must therefore account for dependency risks, interface specifications, and error-handling scenarios.

### **3. Performance and Real-Time Constraints**

Several requirements (e.g., inventory updates, order tracking, payment responses) rely on **real-time or near real-time processing**. Implementation must ensure:

- Minimal system latency
- Consistent data synchronization across departments
- Scalable handling of peak order volumes

### **4. Security and Access Control**

Requirements involving payment processing, staff authentication, and customer data must comply with:

- Role-based access control
- Data privacy policies
- Secure transaction handling
- Industry standards (e.g., PCI-DSS for payment data)

## **5. Data Integrity and Consistency**

Since E-OMS integrates Sales, Finance, Inventory, Logistics, and Support, each functional requirement must consider:

- Prevention of data duplication
- Avoidance of conflicting updates
- Coordination between modules
- Valid state transitions for each order lifecycle step

## **6. Usability and User Experience**

Requirements that involve user interaction (e.g., catalog browsing, order tracking) must consider:

- Clear navigation
- Responsive design
- Accessibility
- Easy-to-understand status updates for customers

## **7. Error Handling and Recovery**

Because of system integration, functional requirements must include coherent responses to:

- Failed payments

- Invalid orders
- Courier API errors
- Missing stock
- Supplier delays
- Customer mistakes (e.g., wrong address)

These scenarios must lead to predictable, controlled outcomes.

## **8. Compliance and Organizational Policies**

Functional requirements must align with internal business rules such as:

- Inventory restocking thresholds
  - Return/refund policies
  - Financial reporting rules
  - Customer notification guidelines
- 

### **Purpose of the Considerations Section**

This section ensures that each functional requirement is interpreted correctly and implemented consistently with project constraints, organizational standards, and external system interactions.

It provides developers, analysts, and testers with the contextual understanding needed to apply the requirements effectively.

## Form

The **Form** specifies the standardized structure used to document each atomic functional requirement. It ensures consistency, clarity, and completeness across all requirements in the E-OMS project. Each requirement is presented using a uniform template called the **requirements shell**, which defines the mandatory fields that must be included.

A complete Functional Requirement shall contain the following elements:

### 1. Requirement ID

A unique identifier (e.g., FR-01, FR-02) used to trace the requirement throughout the lifecycle.

### 2. Requirement Name

A short, descriptive label summarizing the required function.

### 3. Type

Indicates that the requirement is a **Functional Requirement**.

### 4. Description

A clear, concise statement specifying what the system must do. This statement must reflect *one and only one* behavior (atomicity).

### 5. Rationale

A brief justification explaining why this requirement is necessary, linking it to stakeholder needs or business processes.

### 6. Fit Criterion

A measurable condition that determines whether the implementation of the requirement is acceptable.

Often expressed using **Given – When – Then** format.

### 7. Source

The stakeholder, document, or user story from which the requirement was derived.

### 8. Priority

Indicates importance (e.g., Must, Should, Could), based on business value and system dependency.

## 9. Dependencies / Constraints

Lists any related requirements, preconditions, or system limitations that influence implementation.

---

### Purpose of the Form Section

The **Form** ensures that each Functional Requirement is:

- well-structured
- uniform across the entire document
- testable and traceable
- easy for designers, developers, and testers to interpret

By enforcing this standard format, inconsistencies are reduced and the requirements become easier to verify, validate, and reference during system design and implementation.

## 9. Look and Feel Requirements

### 9a. Appearance Requirements

#### Content

The Appearance Requirements define the visual style, layout, and aesthetic qualities of the E-Commerce Order Management System (E-OMS). These requirements specify how the system's user interface should *look* to users, ensuring consistency, professional presentation, and ease of visual comprehension across all screens and modules.

The system's appearance shall include:

#### 1. Consistent Color Palette

The system shall use a uniform color scheme (e.g., primary, secondary, neutral tones) across all pages and modules.

#### 2. Consistent Typography

Standardized font families, sizes, and weights shall be used for headings, labels, buttons, and body text.

#### 3. Clean and Minimal Layout

Screens shall be uncluttered, with clear spacing, readable sections, and intuitive alignment.

#### 4. Brand Identity Alignment

The system shall incorporate E-Market's branding elements (logo, colors, theme) consistently.

#### 5. Visual Consistency Across Devices

The interface should present a uniform look across desktop, tablet, and mobile screens.

#### 6. Clear Visual Hierarchy

Important items (e.g., order status, action buttons) shall be visually emphasized using size, color, or placement.

## **7. Quality Product Imagery**

Product listings shall display clear, high-resolution images without distortion.

### **Motivation**

The purpose of the Appearance Requirements is to ensure that the E-OMS delivers a **professional, intuitive, and trustworthy user experience**. A consistent and well-structured appearance:

- **Improves usability** by making navigation familiar and predictable.
- **Enhances credibility**, making customers more confident in browsing and purchasing.
- **Supports operational efficiency**, helping staff locate actions and information quickly.
- **Maintains brand identity** across all departments and customer-facing modules.
- **Reduces training time** for staff by providing a clean and uniform interface.

Clear Appearance Requirements ensure that developers, UI designers, and testers share the same expectations about the visual aspects of the system.

### **Examples**

The following examples illustrate why the Appearance Requirements are necessary:

#### **1. Improved Usability**

A clean, uncluttered layout helps staff quickly locate critical actions—such as verifying orders or checking inventory levels—resulting in faster task completion and fewer mistakes.

#### **2. Increased Customer Trust**

Customers are more likely to complete purchases when the interface uses uniform colors, clean typography, and high-quality product images, giving the system a reliable and professional appearance.

#### **3. Stronger Brand Identity**

Consistent use of the company logo and color palette across all pages reinforces E-Market's branding, ensuring customers immediately recognize the platform.

#### 4. **Reduced Training Time**

When all modules follow the same visual patterns, new staff learn faster. For example, if the Sales, Finance, and Logistics dashboards share similar button placement and typography, users adapt quickly regardless of their department.

#### 5. **Consistent Cross-Device Experience**

A responsive design ensures that the user interface remains visually clear and easy to navigate on mobile devices, tablets, and desktops, avoiding misalignment or unreadable text.

##### Fit Criterion

The appearance of the system will be considered acceptable when:

- **Given** a user opens any system screen,
- **When** navigating across modules (Sales, Inventory, Finance, Logistics, Support),
- **Then** all screens shall:
  - Use the same color palette and typography;
  - Display consistent layout spacing and alignment;
  - Maintain visible brand identity elements;
  - Render correctly on desktop, tablet, and mobile;
  - Display images clearly without distortion.

A UI review checklist shall be used during testing to confirm that each screen meets these criteria.

##### Considerations

###### 1. **Device and Resolution Variability**

The interface must adapt visually to different screen sizes while retaining visual consistency (responsive design).

## **2. Branding Guidelines**

Visual appearance decisions must align with E-Market's branding standards. If branding changes, appearance requirements may need updates.

## **3. Accessibility**

Colors, contrast ratios, and font sizes must remain usable for users with visual impairments.

## **4. Performance Trade-Offs**

High-quality images and animations must not significantly affect page load times.

## **5. Departmental Usage Needs**

Internal screens for staff (e.g., inventory dashboards) may require more data density but must still remain visually clear.

## **6. Internationalization (if applicable)**

Text expansion in different languages may affect spacing and layout.

## **7. Consistency vs. Flexibility**

While appearance should be uniform, certain modules (e.g., customer-facing vs. admin-facing) may require different visual emphasis.

## 9b. Style Requirements

### Content

The Style Requirements specify the rules governing **visual consistency and structural cohesion** across all interfaces of the E-OMS. Style standards define how text, colors, icons, alignment, spacing, and interactive elements must be presented to ensure uniformity throughout the system. These requirements apply to customer-facing pages (product catalog, checkout, tracking) and internal modules (Sales, Finance, Inventory, Logistics, Customer Support dashboards).

The goal is to ensure that every screen in the system shares a consistent visual identity and predictable layout patterns that improve usability and reduce cognitive load.

### Motivation

Style Requirements are necessary to maintain a consistent and professional user interface across all modules. A standardized style supports usability, improves navigation efficiency, ensures visual clarity, and strengthens the system's brand presentation. Consistent styling also reduces training time for staff members who interact with multiple modules daily. By applying unified layout grids, typography rules, button styles, and iconography, the E-OMS becomes easier to use, easier to maintain, and more trustworthy to customers and stakeholders.

# 10. Usability and Humanity Requirements

## 10a. Ease of Use Requirements

### Content

The Ease of Use Requirements define how the E-OMS should support quick learning, efficient navigation, and minimal cognitive load for both customers and internal staff. These requirements ensure that users can successfully perform tasks—such as placing orders, viewing inventory levels, processing payments, shipping items, or responding to customer inquiries—without confusion or the need for extensive training.

The system should:

- Present clear and intuitive navigation controls.
- Reduce the number of steps required to complete common tasks.
- Provide clear labels, tooltips, and feedback messages.
- Maintain predictable screen structures and workflows.
- Support both beginner and experienced users with accessible shortcuts or streamlined flows.
- Allow users to easily undo, redo, or correct actions.

## 10b. Ease of Use Requirements

### Content

The Ease of Use Requirements define how the E-OMS should support quick learning, efficient navigation, and minimal cognitive load for both customers and internal staff. These requirements ensure that users can successfully perform tasks—such as placing orders, viewing inventory levels, processing payments, shipping items, or responding to customer inquiries—without confusion or the need for extensive training.

The system should:

- Present clear and intuitive navigation controls.
- Reduce the number of steps required to complete common tasks.
- Provide clear labels, tooltips, and feedback messages.
- Maintain predictable screen structures and workflows.
- Support both beginner and experienced users with accessible shortcuts or streamlined flows.
- Allow users to easily undo, redo, or correct actions.

## Motivation

Ease of use is critical to ensuring that users—especially staff members working under time pressure—can perform tasks efficiently and accurately. A system that is simple to learn and easy to navigate increases productivity, reduces training time, and minimizes costly errors such as incorrect order verification, wrong shipments, or duplicate inventory updates. For customers, ease of use improves satisfaction and reduces cart abandonment, support requests, and overall friction in the shopping experience.

## 10c. Personalization and Internationalization Requirements

### Content

The Personalization and Internationalization Requirements specify how the E-OMS should adapt to individual user preferences and support users from different linguistic, cultural, and regional backgrounds. These requirements ensure that the system provides a personalized, comfortable experience for customers and staff, while also being accessible to a global user base if expanded beyond the initial region.

The system should:

- Allow users to customize certain interface elements (e.g., preferred language, notification settings).
- Support multiple languages and make all textual content translatable.
- Display dates, currencies, and numerical formats based on the user's locale.
- Provide consistent functionality regardless of language selection.
- Maintain cultural sensitivity in symbols, colors, and messaging.
- Ensure all translations are clear, accurate, and aligned with industry terminology.

## Motivation

Personalization enhances user satisfaction by allowing individuals to tailor

the system to their preferences, while internationalization expands the potential user base and ensures accessibility in diverse markets. For internal staff, personalization reduces cognitive load by letting them configure dashboards or notification preferences in ways that support their daily workflows. For customers, the ability to interact with the system in their preferred language increases trust, reduces errors, and makes the ordering process smoother.

Internationalization adds strategic flexibility, enabling the business to scale into new regions and improve inclusivity without rewriting major system components.

#### 10d. Learning Requirements

##### Content

Learning Requirements describe how easily new users—both customers and staff—should be able to become proficient in using the E-OMS. These requirements ensure that the system supports quick onboarding, provides helpful guidance, and minimizes the amount of formal training needed. The system should include clear instructions, intuitive workflows, informative help features, and consistent interaction patterns that enable users to operate effectively with minimal previous experience.

The system should:

- Be intuitive enough for new users to accomplish basic tasks without external assistance.
- Provide built-in help features such as tooltips, onboarding messages, or contextual guidance.
- Maintain consistent interactions across modules, reducing the need to relearn different parts of the system.
- Present clear instructions during complex workflows such as returns, inventory updates, or shipment planning.
- Support progressive learning, allowing users to explore advanced features over time.

## Motivation

A system that is easy to learn reduces training time, lowers operational costs, and minimizes errors caused by misunderstanding or incorrect usage. For internal staff, especially those who rotate between departments or handle multiple roles, learnability directly affects efficiency and accuracy. For customers, an intuitive design makes ordering, tracking, and requesting returns faster and more satisfying, reducing frustration and support inquiries.

Well-defined learning requirements ensure that the system remains accessible to new employees, seasonal workers, and customers with varying levels of technical ability.

## 10e. Understandability and Politeness Requirements

This section is concerned with discovering requirements related to concepts and metaphors that are familiar to the intended end users.

### Content

The Understandability and Politeness Requirements specify how the E-OMS should communicate with users through clear, respectful, and easy-to-understand language. These requirements ensure that all system messages—whether informational, instructional, or error-related—are concise, friendly, and free from technical jargon. The system must provide straightforward explanations, use polite phrasing, and maintain a professional tone that supports user confidence and reduces frustration.

This includes:

- Clear and simple language in all instructions, alerts, and notifications.
- Polite and respectful messaging, especially when preventing user errors.
- Explanations that help users understand the next step or how to correct mistakes.
- Avoidance of technical codes or backend terminology in user-facing

text.

- Consistent tone across all modules and communication channels.

## Motivation

A polite and understandable system enhances user satisfaction, reduces confusion, and enables smoother task completion. When messages are easy to interpret, users are less likely to make errors, abandon tasks, or require assistance. Politeness also affects the perception of professionalism; a respectful tone helps build trust between users and the system, especially during sensitive operations such as payment processing or returns.

For internal staff working under pressure, clear and polite instructions significantly reduce cognitive load. For customers, positive and supportive language improves the overall shopping experience and decreases support inquiry.

### 10f. Accessibility Requirements

#### Content

The Accessibility Requirements specify how the E-OMS must be designed to ensure that all users—including those with disabilities or limitations—can effectively access and interact with the system. This includes visual, auditory, motor, and cognitive accessibility.

The system must support a wide range of user needs by providing accessible controls, readable text, alternative formats, and compatibility with assistive technologies.

The system should:

- Comply with recognized accessibility standards such as WCAG 2.1 Level AA.
- Provide sufficient color contrast between text and background.
- Support screen readers by using semantic HTML structures and

descriptive labels.

- Allow keyboard-only navigation for users unable to use a mouse.
- Offer scalable text without breaking layout or readability.
- Ensure form fields, buttons, and links are large enough for easy selection.
- Avoid relying solely on color to convey critical information (e.g., status, errors).
- Present information in simple, understandable language for users with cognitive challenges.

#### Motivation

Accessibility ensures the E-OMS can be used by the widest possible audience, including customers with disabilities and staff members who may face temporary or permanent barriers.

By designing for inclusivity, the system becomes more usable, more compliant with international standards, and more aligned with legal and ethical expectations.

A highly accessible system:

- Reduces potential discrimination issues.
- Improves customer satisfaction and retention.
- Enhances employee productivity and reduces task errors.
- Future-proofs the platform for expansion into regions with accessibility laws.
- Supports individuals using assistive tools such as screen readers or speech input devices.

11.

**Performance Requirements**

Perf

## 11a. Speed and Latency Requirements

### Content

This section defines the maximum allowable time limits for system responses, internal processing, and communication with external partners. These requirements ensure the E-OMS operates efficiently under typical and peak workloads, while maintaining usability and reliability across departments such as Sales, Inventory, Finance, Logistics, and Customer Support.

---

### Motivation

The E-OMS handles real-time operations across multiple interconnected modules. Timely responses are essential to:

- Prevent order delays and abandoned carts during checkout
- Enable real-time stock visibility for Sales and Inventory
- Ensure smooth payment authorization with external gateways
- Maintain fast courier assignment and tracking updates
- Support Customer Support agents in quickly resolving issues

A failure to meet these performance targets could cause inaccurate stock levels, slow payment confirmations, delayed shipments, and reduced customer satisfaction

---

## 11b. Safety-Critical Requirements

### Content

This section defines the safety-critical requirements and risks related to data protection, financial operations, external system communication, and operational integrity.

Since E-OMS handles sensitive information (customer data, payment details) and coordinates logistics actions that affect property (customer shipments), safety standards are defined in terms of **data safety**, **transaction safety**, **system reliability**, and **legal compliance**.

---

### Motivation

While E-OMS is not a physical or life-critical system, failures can still cause:

- Unauthorized financial transactions
- Leakage of private customer information
- Shipment of incorrect or unsafe goods
- Loss of order, stock, or payment records
- Legal and compliance penalties
- Damage to customers' property (e.g., misrouted or lost shipments)
- Reputational harm to the company

Thus, safety-critical requirements focus on **protecting people, their property, and the organization from digital, financial, and operational harm**.

---

## 11c. Precision or Accuracy Requirements

### Content

This section defines the required accuracy for all calculations, measurements, timestamps, and data values generated or used by the E-OMS. These requirements ensure reliable financial transactions, inventory consistency, logistics coordination, and reporting accuracy across the system.

---

## **Motivation**

Because E-OMS manages pricing, payments, inventory quantities, shipment schedules, and performance reports, accuracy errors could lead to:

- incorrect charges to customers
- stock miscounts that cause overselling
- delayed or misrouted deliveries
- inconsistent financial or operational reports
- legal and compliance violations

Precision standards ensure internal departments and customers receive results that are mathematically, operationally, and financially reliable.

---

## **11d. Reliability and Availability Requirements**

### **Content**

This section defines how reliable and available the E-OMS must be during normal and peak operations. It specifies acceptable downtime, allowable failure rates, recovery expectations, and the minimum system uptime required to maintain continuous support for sales, customers, and logistics.

---

## **Motivation**

Because E-OMS manages time-sensitive operations—payments, order routing, inventory reservations, and courier updates—any downtime or

failure could:

- Prevent customers from placing orders
- Cause duplicated or lost transactions
- Create inventory mismatches
- Delay shipments
- Break communication with suppliers or couriers
- Undermine management reporting and decision-making

High reliability and availability ensure E-OMS remains dependable even under heavy load and during multi-department activity.

---

## **Reliability Requirements (Customized for E-OMS)**

### **R1. Failure Rate**

- The system shall maintain a failure rate of **less than 1 failure per 10,000 transactions** across all modules (Sales, Finance, Inventory, Logistics).
- No critical transaction (payment authorization, inventory reservation, shipment assignment) may fail silently; **100%** must produce a success or error response.

### **R2. Transaction Integrity**

- All financial transactions must be **atomic**, meaning that any incomplete transaction must be rolled back with **zero data corruption**.
- Inventory updates must never result in a negative or inconsistent stock count.

### **R3. Error Recovery**

- For recoverable errors (network delays, API timeouts, gateway

congestion), the system must automatically retry the operation **up to 3 times within 10 seconds**.

- If external systems (payment gateway, courier API) fail, E-OMS shall enter **safe mode**, logging the failure while preventing customer-facing errors or data losses.

#### R4. Logging & Diagnostics

- 100% of failures must be logged with timestamps, error codes, and affected modules.
  - Log integrity must be maintained even during partial system outages.
- 

### Availability Requirements (Customized for E-OMS)

#### A1. System Uptime

- E-OMS shall be available **24 hours per day, 365 days per year**, excluding scheduled maintenance.
- Minimum uptime requirement: **99.5% annually** (maximum downtime ≈ 1.83 days per year).
- For peak seasons (e.g., Black Friday), uptime must be **99.9%**.

#### A2. Scheduled Maintenance

- Planned maintenance shall occur **outside peak business hours** and must not exceed:
  - **2 hours per month** total
  - Notice period: **72 hours** before maintenance
- During maintenance, the system shall enter **read-only mode** for catalog browsing.

#### A3. Failover and Redundancy

- The system must fail over to a secondary server within **60 seconds** of detecting primary server failure.
- Database replication shall occur in real time with a maximum replication lag of **< 2 seconds**.

#### A4. External Systems Availability

- E-OMS shall not become unavailable due to failure in external systems.
  - If payment gateway is unreachable → system queues the request and notifies customer
  - If courier API is offline → order stays in “Awaiting Shipment” until reconnection
  - If supplier system is offline → replenishment requests remain queued

Availability must remain **> 99%** even when external systems are degraded.

---

#### 11e. Robustness or Fault-Tolerance Requirements

##### Content

This section defines how the E-OMS must continue operating when unexpected or abnormal conditions occur. It specifies system behavior under partial failures, degraded external services, network interruptions, and data integrity risks, ensuring that the system continues to provide essential functionality even during disruptions.

---

##### Motivation

The E-OMS integrates multiple internal departments (Sales, Finance,

Inventory, Logistics) and external partners (suppliers, couriers, payment gateways). Any single failure can propagate and break the entire order pipeline.

Robustness requirements ensure that:

- Essential services remain available
  - No corrupted or inconsistent data is created
  - Customer experience is minimally impacted
  - The system fails gracefully, not catastrophically
  - Recovery from failures is automatic and reliable
- 

## 11f. Capacity Requirements

### Content

This section defines the maximum number of users, transactions, orders, and data volume that the E-OMS must support. It also sets storage requirements, performance expectations under peak load, and capacity limits for external integrations such as payment gateways and courier APIs.

---

### Motivation

To ensure the system remains stable, responsive, and scalable under high demand, it must be able to process:

- thousands of orders
- simultaneous user sessions
- large product catalogs
- constant inventory updates

- real-time courier and payment interactions

If capacity is underestimated, the system may become slow, unresponsive, or even crash during peak periods—resulting in lost sales and operational failures.

## 11g. Scalability or Extensibility Requirements

### Content

This section defines how the E-OMS must scale over time to accommodate expected business growth. It covers increases in user traffic, order volume, product catalog size, storage capacity, and external integrations. It also addresses architectural extensibility to support future features and partner integrations.

### Motivation

E-Market Inc. is expected to expand its customer base, product range, and geographic coverage.

Without scalability:

- customer traffic may overwhelm the system
- order processing delays could occur
- inventory synchronization could fail
- payment and courier APIs could bottleneck
- reporting tools could slow down
- new business units or product types may be impossible to support

Scalability ensures the system continues functioning efficiently as demand grows and new capabilities are added.

## 11h. Longevity Requirements

### Content

This section specifies how long the E-OMS is expected to remain operational, maintainable, and cost-effective before requiring major replacement, redesign, or technology migration.

### Motivation

E-Market Inc. is investing in a robust, multi-department order management platform. To ensure that investment pays off, stakeholders must know:

- how long the system will remain viable
- how long it will receive updates and support
- when major upgrades may be needed
- how long performance and security standards can be maintained

## 12. Operational and Environmental Requirements

### 12a. Requirements for Interfacing with Adjacent Systems

#### Content

These requirements describe how the E-OMS must interact with external or “adjacent” systems that play a role in the order lifecycle. Adjacent systems include payment gateways, courier/shipping APIs, supplier platforms, email/SMS notification services, and any third-party integrations required for the workflow.

The E-OMS must:

- Support secure, real-time data exchange with payment gateways for authorization and verification.
- Integrate with courier APIs to generate tracking numbers and retrieve delivery status updates.

- Interface with supplier systems to send and receive replenishment requests automatically.
- Connect with communication services (SMS/email APIs) to send order notifications.
- Use standardized communication protocols (REST APIs, JSON, HTTPS).
- Validate, log, and handle errors gracefully when external systems fail or respond slowly.
- Maintain consistent data formats and ensure compatibility across updates.

These interfaces ensure seamless end-to-end order processing from purchase to delivery.

## 12b. Productization Requirements

### Content

The Productization Requirements describe how the E-OMS must be prepared, packaged, and standardized for deployment, distribution, maintenance, and long-term reuse. These requirements ensure that the system can be treated as a formal product—capable of being installed, configured, updated, and supported consistently across different environments or organizational contexts.

The system must:

- Provide a standardized installation package with clear instructions.
- Offer configuration tools for environment-specific settings (e.g., API keys, database connections).
- Include versioning, release notes, and changelog documentation for each update.
- Support modular deployment so organizations can enable or disable

features based on need.

- Offer automated update mechanisms or scripts.
- Include comprehensive user manuals, administrator guides, and API documentation.
- Provide diagnostic tools or logs to help with troubleshooting.
- Maintain backward compatibility across minor updates when possible.

## Motivation

Productization is essential for ensuring that the E-OMS can be easily deployed, maintained, and scaled in a consistent and reliable manner. Without clear productization standards, each installation may differ, leading to errors, increased costs, inconsistent performance, and difficulty in supporting the system across multiple environments.

A fully productized system:

- Reduces installation and onboarding time.
- Makes updates predictable and safe.
- Ensures consistent performance across deployments.
- Improves maintainability and troubleshooting.
- Supports business scalability and potential commercialization.
- Reduces dependency on specialized developers for routine system operations.

## 13. Maintainability and Support Requirements

### 13a. Maintenance Requirements

#### Content

The Maintenance Requirements define how the E-OMS must be designed, implemented, and documented to support efficient ongoing maintenance, updates, and problem resolution. These requirements ensure that administrators, support teams, and developers can diagnose issues, apply patches, perform updates, and enhance the system without causing disruptions or introducing inconsistencies.

The system must:

- Provide clean, modular, and well-documented code to simplify updates and debugging.
- Support routine maintenance tasks such as log rotation, cache clearing, and system health checks.
- Include monitoring tools to track system performance, errors, and usage patterns.
- Offer diagnostic tools and detailed logs to help identify issues quickly.
- Allow maintenance activities with minimal downtime by supporting rolling updates where possible.
- Maintain compatibility with previous data formats or provide automated migration scripts when changes occur.
- Provide clear procedures for applying bug fixes, feature enhancements, and security patches.

These requirements ensure that the system remains efficient, stable, and adaptable throughout its lifecycle.

#### Motivation

A maintainable system reduces the cost, time, and risk associated with ongoing updates and operational support. Since E-OMS is a mission-critical system that supports sales, inventory, finance, logistics, and customer service, maintenance must be both efficient and predictable.

Maintenance requirements:

- Improve system stability by ensuring issues can be detected and resolved quickly.
- Reduce downtime by making updates smoother and less disruptive.
- Allow technical teams to extend the system with new features safely.
- Ensure long-term sustainability by enabling consistent maintenance practices.
- Reduce operational risk caused by undocumented changes or inconsistent updates.

## 13b. Supportability Requirements

### Content

The Supportability Requirements define how the E-OMS should be designed and documented to ensure that it can be effectively supported by technical teams, customer service teams, and system administrators throughout its operational lifecycle. Supportability focuses on the tools, processes, and system characteristics that enable quick diagnosis, efficient troubleshooting, and effective user assistance.

The system must:

- Provide comprehensive technical documentation, including API references, configuration guides, and deployment instructions.
- Include built-in monitoring and alerting tools to track performance, errors, and unusual system behavior.
- Log critical events with sufficient detail to support issue diagnosis.
- Support remote diagnostics and remote access for authorized technical staff.
- Offer a structured support workflow for handling customer issues, including ticket tracking and categorization.
- Provide clear status indicators for modules (e.g., inventory synchronization, payment services, courier connectivity).
- Include tools to export diagnostic data such as logs, server metrics, and

configuration snapshots for troubleshooting.

- Support compatibility testing tools for third-party system integrations.

These requirements enable ongoing system reliability and help support personnel resolve issues efficiently.

## Motivation

A supportable system reduces downtime, minimizes user frustration, and ensures that technical and customer support teams can respond effectively when issues arise. Because the E-OMS interacts with multiple departments (Sales, Inventory, Logistics, Finance, Customer Service) and external systems (payment gateways, courier APIs, suppliers), problems can originate from many sources. Supportability requirements ensure that every stakeholder has the tools and information needed to identify and resolve problems quickly.

Benefits include:

- Faster response times to technical issues
- Improved customer satisfaction due to effective support
- Lower operational risk and fewer critical failures
- Reduced effort and cost for support teams
- Improved long-term sustainability and stability of the system

Without strong supportability requirements, diagnosing issues becomes difficult, leading to longer downtime and lower confidence in the system.

## 13c. Adaptability Requirements

### Content

The Adaptability Requirements describe how the E-OMS must support future changes, expansions, and environmental variations without requiring major redesigns. The system should be flexible enough to accommodate new business processes, additional modules, external integrations, changes in regulations, and evolving user needs.

The system must:

- Allow new features and modules to be added without disrupting existing ones (modular architecture).

- Support configuration-based updates where possible, minimizing code changes.
- Allow integration with new third-party APIs (payment gateways, courier systems, suppliers).
- Adapt to new business rules, workflows, or pricing structures through configuration settings.
- Remain functional under different deployment options (on-premises, cloud-based, hybrid).
- Support data model extensions without breaking existing functionality.
- Handle variations in user volume, product catalog size, and operational workflows.

These requirements ensure long-term usability and reduce the cost and complexity of future enhancements.

### Motivation

As the business grows, the E-OMS must evolve to support new operational needs, expanding product ranges, and additional market regions. Adaptability ensures that the system remains relevant and can scale with changing requirements without costly reengineering.

Well-defined adaptability requirements:

- Reduce long-term maintenance and upgrade costs
- Enable rapid integration with new services or technologies
- Allow business units to adjust workflows without software modifications
- Future-proof the system against evolving customer expectations
- Improve organizational agility when entering new markets or adopting new processes

Without adaptability, the system becomes rigid, expensive to update, and increasingly misaligned with the organization's needs.

## 14. Security Requirements

## 14a. Access Requirements

### Content

This section specifies which user roles are authorized to access the system, what data they may view or modify, and the conditions under which access is granted. It defines role-based access boundaries for internal departments, customers, and external partners.

### Motivation

The E-OMS handles sensitive data related to customers, payments, financial transactions, suppliers, and logistics partners.

Access requirements ensure:

- protection of confidential data
- prevention of unauthorized modifications
- compliance with privacy and financial regulations
- clear separation of duties across departments
- avoidance of fraud, errors, or misuse

This section defines what each role **may access, may modify, and must not see**.

---

## Access Requirements (Customized for E-OMS)

---

### A1. Customer Access

Function/Data	Access Level
Their own account information	View + Edit (contact details, password, addresses)

Their own orders	View
Their own returns/refunds	View
Payment methods (tokenized)	View + Add/Delete
Product catalog	View
Checkout process	Execute

Customers **shall not** view or modify any internal system data, staff actions, or other customers' information.

---

## A2. Sales Staff Access

Sales staff may access:

Function/Data	Access Level
Customer orders (all)	View
Order verification tasks	View + Edit
Order cancellation	Execute
Customer contact details	View
Product catalog	View + Edit

Sales staff **may not** access payment authorization data, financial records, or supplier information.

---

## A3. Finance Staff Access

Finance staff may access:

Function/Data	Access Level

Payment records	View
Refund processing	Execute
Invoices	View + Edit
Financial reports	View
Transaction logs	View

Finance staff **shall not** access inventory quantities, supplier data, or logistics planning information.

---

#### A4. Inventory Staff Access

Inventory staff may access:

Function/Data	Access Level
Inventory levels	View + Edit
Warehouse stock movements	Execute
Replenishment requests	Create + View
Supplier information	View
Stock audit reports	View

Inventory staff **shall not** access customer payment details, sales transaction data, or return/refund financial adjustments.

---

#### A5. Logistics Staff Access

Logistics staff may access:

Function/Data	Access Level
Shipment creation and routing	View + Edit
Courier selection and tracking data	View + Edit
Delivery status updates	Execute
Customer address for shipment	View
Shipping reports	View

Logistics staff **shall not** access pricing adjustments, inventory replenishment decisions, or sensitive financial information.

---

## A6. Customer Support Agents

Support agents may access:

Function/Data	Access Level
Customer order history	View
Return and complaint processing	Execute
Customer contact information	View
Shipment status	View
Refund status (but not modify)	View

Customer Support **may not** view internal financial details, inventory management records, or supplier pricing.

---

## A7. Administrators / Management

Admins (or authorized executives) may access:

Function/Data	Access Level
All system configuration settings	Edit
All user roles and permissions	Edit
System-wide reports (sales, finance, inventory, logistics)	View
Logs and audit trails	View
All operational data	View

Administrative access is restricted to a small set of trusted individuals and must be granted only for operational necessity.

---

## A8. External Partners

### Payment Gateway

- Access limited to receiving tokenized payment requests and sending back authorization results.
- No external partner may access internal customer profile data.

### Couriers

- May receive shipment details needed for delivery (name, address, phone, order ID).
- They may not access payment or financial data.

### Suppliers

- May receive replenishment requests and inventory quantities relevant to restocking.
  - They may not access customer data or sales analytics.
-

## Fit Criterion

The system must demonstrate that:

- Each system function and data object has a clearly defined list of roles permitted to **view, add, edit, or delete** it.
  - Access logs must confirm that only authorized roles can access or modify sensitive information.
  - Attempted unauthorized access must be recorded and rejected 100% of the time.
  - Every data field (e.g., payment status, inventory quantity, address information) must have an explicit access rule tied to specific user roles.
- 

## Considerations

- Some data—payment details, customer information, transaction history—is highly sensitive and must be strictly access-controlled.
  - Misuse of functions (e.g., issuing unauthorized refunds, modifying stock levels) can cause financial or operational damage.
  - Certain staff may not want management to access personal notes or internal communication; define boundaries where needed.
  - Avoid specifying implementation details (e.g., passwords, MFA, tokens). The requirement focuses only on **who** has access—**not how** access is enforced.
  - Security consultants should be involved if the product must meet higher-than-normal confidentiality standards, such as PCI-DSS.
- 

## 14b. Integrity Requirements

### Content

The E-OMS shall ensure the integrity of all business-critical data—including customer orders, product inventory, payment records, shipment manifests, and user permissions—by preventing unauthorized, invalid, or inconsistent modifications. The system must detect, reject, and log any attempt to introduce corrupted, incomplete, or logically invalid data. Additionally, the product shall maintain internal consistency across all modules (Sales, Inventory, Finance, Logistics, Support) even in the event of partial system failures, external integration

errors, or misuse by authorized users.

## Motivation

E-Market Inc. relies entirely on accurate, synchronized data to fulfill orders, manage stock, process payments, and serve customers. As stated in the System Vision Document, current inefficiencies stem from *data duplication*, *communication gaps*, and *stock errors* caused by disconnected systems. The E-OMS must eliminate these risks by enforcing strict data integrity at every layer. Failure to maintain integrity could result in overselling, payment discrepancies, shipment failures, or loss of customer trust—directly threatening business continuity.

## Examples

- The product shall reject order submissions with missing customer contact details or invalid product SKUs.
- The product shall prevent inventory stock levels from falling below zero due to race conditions or double-reservation.
- The product shall ensure that a confirmed order cannot be deleted or altered without an auditable trail.
- The product shall validate all incoming data from external partners (e.g., courier manifest, supplier replenishment response) against expected formats and business rules before processing.
- The product shall protect itself from intentional abuse, such as a support agent attempting to modify another user's order without proper authorization.

## Considerations

- All database transactions involving order state changes (e.g., payment → inventory reservation → shipment) must be **atomic** and **consistent** (ACID-compliant).
- Input validation must occur at both the **UI layer** (e.g., required fields, format checks) and the **business logic layer** (e.g., stock availability, payment gateway response verification).
- Data integrity must be preserved during **system failures** (e.g., power loss, crash) through transaction rollback and recovery mechanisms.
- Integrity rules must align with **Security Requirement S1** and **User Story US 14**, ensuring that only users with appropriate permissions can modify specific data types.

- External integrations (Payment Gateways, Couriers, Suppliers) are potential sources of invalid data; the E-OMS must **sanitize and validate** all inbound messages before updating internal state.
- Backups (as defined in Section 14) must be **consistent snapshots** that can restore the system to a valid, non-corrupted state.

## 14c. Privacy Requirements

### Content

The E-OMS shall protect the personal and financial data of customers in accordance with applicable privacy laws and E-Market Inc.'s internal data handling policies. The system shall ensure that customer data—including names, contact details, delivery addresses, order history, and payment information—is collected, stored, processed, and disclosed only with explicit user consent and strictly for legitimate business purposes (e.g., order fulfillment, support, or legal compliance). The system shall **never** share customer data with third parties (including marketing partners) without the customer's informed, opt-in consent.

### Motivation

Customer trust is foundational to E-Market's e-commerce business model. Mishandling personal data—especially sensitive payment and contact information—can result in legal penalties, reputational damage, and loss of customers. Compliance with privacy expectations is not only a legal necessity but a core component of customer satisfaction and brand integrity, as emphasized in user feedback and acceptance criteria (e.g., **US 7, US 9, S1**).

## 14d. Audit Requirements

### Content

The E-OMS shall automatically generate and retain tamper-evident audit logs for all critical system activities, including user logins, permission changes, order modifications, payment processing events, inventory adjustments, and return/refund approvals. Each log entry shall include:

- Date and time (UTC with local time zone)
- User ID or system actor
- Action performed (e.g., “Order Cancelled”, “Product Price Updated”, “Permission Granted to Role”)
- Affected data (e.g., Order ID, Product SKU, User ID)
- Source IP address (for web-based access)
- Outcome (success/failure)

Logs shall be stored securely for a minimum of **3 years** and shall not be deletable or modifiable by any user, including System Administrators. The system shall

provide read-only access to audit logs for authorized auditors and compliance officers.

## Motivation

To build a system that complies with internal financial controls, external regulatory expectations, and industry best practices for accountability. Audit trails are essential to:

- Investigate discrepancies in orders, payments, or inventory
- Verify compliance with Security Requirement **S1**
- Support forensic analysis in case of suspected misuse or data breach
- Fulfill potential legal or fiscal audit obligations

Without comprehensive, immutable logs, E-Market Inc. cannot demonstrate data integrity or user accountability—undermining trust in the system's reliability.

## 14e. Immunity Requirements

### Content

The Immunity Requirements specify how the E-OMS must protect itself from invalid data, malicious inputs, system misuse, and operational anomalies. These requirements ensure that the system can withstand incorrect or unexpected actions—whether intentional or accidental—without compromising performance, stability, or data integrity.

The system must:

- Validate all user inputs, API requests, and data entries to prevent injection, corruption, or malformed data.
- Reject unauthorized or suspicious operations while logging them for audit purposes.
- Enforce strict authentication and role-based authorization to prevent misuse by staff or external actors.
- Prevent harmful operations such as duplicate order submissions, invalid payment attempts, or conflicting inventory updates.
- Ensure that system processes degrade gracefully rather than failing abruptly when encountering abnormal data or conditions.
- Provide protection against automated attacks, including rate limiting and bot detection measures.

- Maintain high-integrity audit trails to track all sensitive operations.

These requirements ensure that the E-OMS maintains operational stability and data accuracy even when interacting with unreliable or hostile inputs.

### Motivation

The E-OMS handles sensitive data (customer information, payments, inventory, and logistics) and interacts with external systems. It must therefore remain resilient against errors, misuse, and malicious attempts to compromise functionality.

Immunity requirements:

- Protect business operations from disruption caused by corrupted or invalid data.
- Prevent security breaches caused by unvalidated inputs or unauthorized actions.
- Reduce the risk of financial loss due to fraudulent orders or payment attacks.
- Ensure that errors in one module do not propagate and cause system-wide failures.
- Enhance user trust by ensuring the system behaves reliably even in adverse conditions.

Without strong immunity, system reliability decreases, vulnerabilities increase, and the organization may face operational, financial, and reputational risks.

## 15. Cultural Requirements

### 15a. Cultural Requirements

#### Content

The Cultural Requirements specify how the E-OMS must accommodate the cultural expectations, norms, and preferences of users from different backgrounds. These requirements ensure that the system communicates appropriately, respectfully, and effectively across diverse cultural contexts,

especially if the organization expands to international markets or serves multicultural user groups.

The system must:

- Support culturally appropriate language, symbols, icons, and color usage.
- Provide localized content that respects cultural norms, holidays, time formats, and communication tone.
- Avoid culturally sensitive or offensive imagery, terminology, or design elements.
- Offer flexible address formats, name formats, and ordering conventions to support different countries.
- Accommodate cultural variations in purchasing behavior, customer support expectations, and user interaction patterns.
- Ensure neutrality in visual and textual representation to remain inclusive for diverse user groups.

These requirements ensure a respectful, familiar, and comfortable user experience across cultural contexts.

### Motivation

Cultural requirements are important because users from different regions may interpret colors, symbols, text, or workflows differently. A culturally conscious system improves user comfort, trust, and adoption—especially for customer-facing parts of the E-OMS.

Strong cultural adaptability:

- Enhances usability and acceptance in international markets
- Reduces misunderstandings caused by culturally inappropriate content
- Improves customer satisfaction by aligning with local expectations
- Strengthens brand reputation and global competitiveness
- Prevents potential cultural missteps that could cause frustration or offense

Without attention to cultural factors, the system risks alienating users, reducing engagement, or creating barriers that limit expansion into new markets.

## 16.

## Legal

### Requirements

#### 16a. Compliance Requirements

##### Content

The Compliance Requirements specify the legal, regulatory, and industry standards the E-OMS must adhere to during development, deployment, and operation. These requirements ensure that the system complies with applicable laws regarding data protection, consumer rights, digital transactions, financial reporting, accessibility, and industry-specific regulations.

The system must:

- Comply with data protection and privacy laws (e.g., GDPR, local data protection acts).
- Ensure secure handling and storage of payment data following PCI-DSS standards.
- Adhere to e-commerce regulations regarding electronic receipts, refund policies, and consumer protection.
- Maintain auditability to support financial reporting requirements.
- Ensure accessibility to meet legally mandated digital accessibility standards (e.g., WCAG 2.1 AA).
- Retain data for legally required durations and allow secure deletion when retention periods expire.
- Provide mechanisms for users to exercise their legal rights (e.g., access, correction, deletion of personal data).
- Support local taxation rules, including currency formatting, invoicing rules, and sales tax calculations.

These requirements ensure that the E-OMS operates within legal boundaries across all regions it serves.

##### Motivation

Legal compliance protects the organization from financial penalties, reputational

damage, and operational risks. As the E-OMS manages sensitive customer data, payment information, and transactional records, adherence to legal requirements is essential for trust, transparency, and long-term sustainability.

Compliance requirements:

- Reduce exposure to regulatory violations and lawsuits
- Ensure secure handling of personal and payment data
- Build customer trust through transparent and lawful operations
- Align the system with industry and governmental expectations
- Protect the organization when expanding to new regions with different legal frameworks

A lack of compliance can lead to severe legal, financial, and operational consequences.

## 16b. Standards Requirements

### Content

The Standards Requirements define the technical, operational, and documentation standards that the E-OMS must comply with to ensure interoperability, quality, and consistency. These standards cover coding practices, data formats, system integration protocols, documentation conventions, and industry-required specifications.

The system must:

- Follow standardized coding conventions and architectural guidelines to ensure maintainable and readable code.
- Use widely accepted communication standards (RESTful APIs, JSON/HTTPS, OAuth 2.0, etc.) for external system integration.
- Apply standard data formats for dates, currency, measurements, and timestamps (e.g., ISO 8601).
- Use standardized accessibility guidelines such as WCAG 2.1 for user interface design.
- Follow established logging, versioning, and documentation standards (e.g., semantic versioning, ISO documentation practices).
- Maintain consistent naming conventions across modules, APIs, and

database structures.

- Adhere to industry standards for software testing, validation, and deployment (e.g., ISTQB-aligned testing, CI/CD best practices).

These requirements ensure high-quality software that integrates smoothly with other systems and is easy to maintain and evolve.

### Motivation

Standards ensure consistency, reliability, and long-term sustainability. By adhering to recognized industry and technical standards, the E-OMS becomes easier to maintain, integrate, test, extend, and audit. Standards also reduce ambiguity, lower training requirements, and ensure that new developers can quickly understand the system.

Strong alignment with standards:

- Improves interoperability with external systems and third-party vendors
- Reduces development and onboarding time
- Ensures future-proof system evolution
- Enhances code quality, security, and maintainability
- Promotes organizational consistency across departments and teams
- Supports compliance and audit requirements

Without standardized approaches, the system risks inconsistencies, integration failures, and long-term technical debt.

## 17. Tasks

### 17a. Project Planning

#### Content

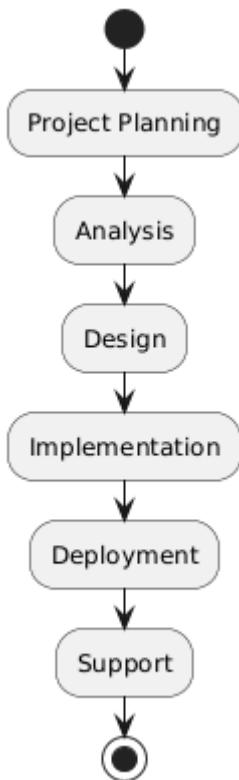
The project will follow a **Waterfall (predictive) development life cycle**. Phases

are executed in order: Project Planning, Analysis, Design, Implementation, Deployment, and Support. Each phase produces deliverables that serve as inputs to the next phase, and a phase is considered complete before the next begins.

### Motivation

Using a Waterfall life cycle provides a clear sequence of work, unambiguous phase boundaries, and well-defined handoffs, which simplifies coordination and verifies that each stage's outputs are available before subsequent work starts.

### Form



### 17b. Planning of the Development Phases

#### Content

The project is divided into a series of Waterfall phases. Each phase has a defined purpose and produces outputs that support the next stage of work. The phases provide a clear structure for organizing activities from initial study through completion.

#### Phases:

1. **Project Planning** – establish objectives, scope, and overall approach.
2. **Analysis** – examine the business problem and define the required functionality.
3. **Design** – structure the solution and prepare specifications to guide construction.

4. **Implementation** – build the components defined during design.
5. **Deployment** – prepare the system for use and complete the final transition.
6. **Support** – address issues and maintain the system after completion.

Each phase depends on the completion of the previous phase and uses its outputs as inputs to continue progress.

### Motivation

Breaking the work into phases makes the project manageable and predictable. Clear boundaries help coordinate tasks, maintain consistency, and ensure that essential decisions are made before subsequent work begins.

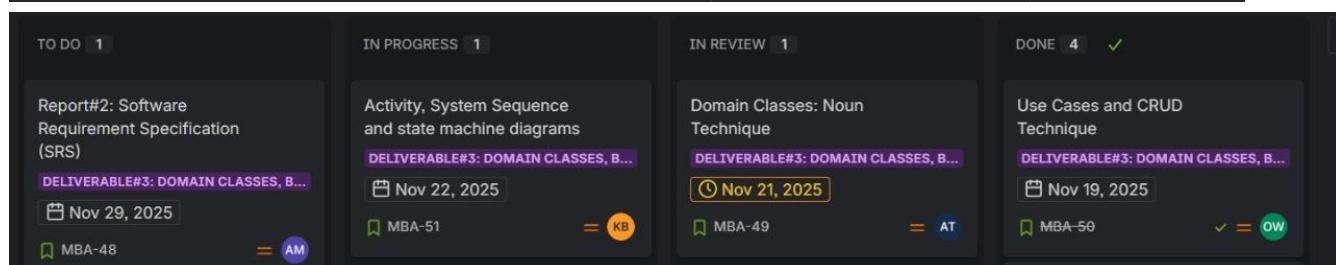
### Form

Phase	Value / Benefit	Required Operational Date	Operating Environment Components	Functional Requirements Included	Non-Functional Requirements Included
<b>Project Planning</b>	Defines direction and prepares scope	N/A	N/A	N/A	<b>R1 – 24/7 availability</b>
<b>Analysis</b>	Establishes complete understanding of required system behavior	N/A	N/A	N/A	<b>P1 – Payment + inventory update within 5s</b>
<b>Design</b>	Structures the solution and prepares for implementation	N/A	N/A	N/A	<b>U1 – ≤1 hour training per module</b>

<b>Implementation</b>	Constructs the defined solution components	N/A	N/A	N/A	<b>S1 – Secure access control, RBAC, encrypted communication</b>
<b>Deployment</b>	Makes the completed system ready for use	N/A	N/A	N/A	<b>A11 – Accessibility compliance (WCAG 2.1 AA)</b>
<b>Support</b>	Maintains system stability after deployment	N/A	N/A	N/A	<b>Reliability &amp; recovery expectations tied to R1</b>

# Jira Project Management Evidence

Work	Assignee	Status
❖ MBA-43 Deliverable#3: Domain Classes, Beh...	AT Ali Tamer	IN PROGRESS
❖ MBA-51 Activity, System Sequence and state machi...	KB Kirollos Bolos	DONE
❖ MBA-55 Abdelrahman Ahmed	AT Ali Tamer	DONE
❖ MBA-54 Kirollors	AT Ali Tamer	DONE
❖ MBA-50 Use Cases and CRUD Technique	OW Omar Abdel Wahab	DONE
❖ MBA-49 Domain Classes: Noun Technique	AT Ali Tamer	DONE
❖ MBA-53 Abdelrahman Amr	AA Abdelrahman AboAbdo	DONE
❖ MBA-52 Ali	AT Ali Tamer	DONE
❖ MBA-48 Report#2: Software Requirement Specifica...	AM Abdelrahman Ahmed...	IN REVIEW
❖ MBA-56 All team will contribute in report 2	AT Ali Tamer	IN REVIEW



# GitHub Evidence

Repository Link: <https://github.com/Kiro-create/E-Commerce-Order-Management-System-E-OMS->  
(All work, commits and pull requests are visible on GitHub.)

## Conclusion

Reports 1 and 2 together define a complete requirements baseline for the E-OMS. The first report establishes the domain model using the Noun Technique, CRUD Analysis, and Event Decomposition, identifying all core classes, relationships, and system behaviors, supported by UML diagrams that validate the workflow and state changes. The second report extends this into a full Volere requirements specification, detailing stakeholders, business goals, business events, constraints, and both functional and non-functional requirements. Combined, they provide a clear and traceable foundation for system design and implementation, ensuring the E-OMS aligns with real business processes and stakeholder needs.