Software Requirements and Design Document

for

“Purrfect Nest”

Prepared by Ahmed Ansari (21i-2453) & Zainab Aslam (21i-2575)

2-12-2023

Table of Contents

1.Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document delineates the specifications for "Purrfect Nest." This document encompasses the entirety of the system, detailing its alignment with the overarching goals of the project. The system is envisioned as a comprehensive solution to streamline inventory management in pet stores, enhancing efficiency, accuracy, and customer satisfaction.

1.2 Product Scope

The scope of "Purrfect Nest" extends to creating an all-encompassing solution for pet inventory management. This system is designed to:

I. *Automated inventory tracking processes.*
II. *Provide real-time updates on product availability.*
III. *Generate detailed reports on sales trends and inventory needs.*
IV. *The software aligns with the strategic goals of modernizing pet care businesses, improving operational efficiency, and enhancing customer engagement.*

1.3 Title

"Purrfect Nest: Revolutionizing Pet Inventory Management."

1.4 Objectives

The primary objectives of the "Purrfect Nest" project are:

I. To automate and streamline pet inventory management, thereby reducing manual labor and minimizing inventory-related errors.
II. To provide customers with real-time updates on product availability, enhancing customer service and order fulfillment.
III. To generate comprehensive reports on sales data and inventory trends, aiding in informed business decision-making.
IV. To assess the technical, financial, and strategic feasibility of implementing the Pet Inventory Management System in pet stores.

1.5 Problem Statement

The project was conceived in response to the challenges faced by pet stores in efficiently managing their inventory using conventional methods. These challenges often lead to issues like overstocking, understocking, product expiration, and difficulty in identifying popular products, impacting customer experience and business profitability. The "Purrfect Nest" aims to address these issues, elevating the overall efficacy of pet store operations through its specialized inventory management system.

2.Overall Description

2.1 Product Perspective

"Purrfect Nest" is a pioneering software product, conceived as a standalone, self-contained system tailored for the pet care industry. It is designed to seamlessly integrate with existing business processes of pet stores, acting as a comprehensive solution for inventory management challenges.

2.2 Product Functions

Key functions of "Purrfect Nest" include:

I. *Automated tracking and management of pet products inventory.*
II. *Real-time updates on stock levels and product availability.*
III. *Advanced analytics for understanding sales trends and inventory replenishment needs.*
IV. *Customizable alerts and notifications for stock management.*
V. *Integration capabilities with existing sales and management systems in pet stores.*
VI. *User-friendly interface for easy navigation and use by store staff.*

These functions are organized to cater to the specific needs of the pet care industry, ensuring that "Purrfect Nest" is not just a tool, but a strategic asset for pet store businesses.

2.3 List of Use Cases

Here is the list of all the use case needed to implement the pet inventory management system:

I. Inventory history tracking
II. Add a new pet product.
III. Update pet product information
IV. Remove pet products.
V. Search for a pet product.
VI. View product details.
VII. Generate inventory reports.
VIII. Set low stock alerts.
IX. Process customer order
X. Handle return and refunds
XI. User authentication
XII. Backup and restore data.
XIII. Customer notification.

2.4 Extended Use Cases

There was a total of 13 different use cases we have made, below we have given the extended use case of each of them

Use Case Name	Inventory History Tracking	
Scope	Pet Supply Inventory Management System	
Level	System goal	
Primary actor	Inventory manager	
Stakeholders and interests	1. **Inventory Manager**: Wants to maintain a history of inventory changes for auditing purposes. 2. **Customers:** Expect accurate information about available products. 3. **Administrator:** A well-maintained inventory ensures optimized stock levels and financial performance.	
Preconditions	• The Inventory Manager is logged into the system.	
Postconditions	• The system successfully records and stores inventory changes.	
Main success scenario	Actor (Inventory manager)	System
	1. The Inventory Manager requests to track inventory changes.	2. The system verifies that the Inventory Manager is authorized to perform this action.
	3. The system starts monitoring inventory changes.	4.The system logs each change, including details such as the product affected, quantity, timestamp, and the responsible
	5.The Inventory Manager can access the inventory history log for auditing purposes.	party. 6. The system displays the inventory history, allowing the Inventory Manager to review changes and transactions.
Extensions	Actor (Inventory manager)	System

		4a. **Unauthorized Access:** If the system determines that the Inventory Manager is not authorized, it denies access and logs the unauthorized attempt.
		5a. **No Inventory History:** If there are no inventory changes to track, the system informs the Inventory Manager that there is no history available.

Use Case Name	Customer Notifications	
Scope	Pet Supply Inventory Management System	
Level	User goal	
Primary actor	Customer	
Stakeholders and interests	**1** **Inventory Manager**: The Inventory Manager expects to stay informed about customer notifications and leverage notification data for inventory planning and order fulfillment. **2** Customers Expects to receive notifications regarding order confirmations, shipping updates, and order delivery.. **3** **Administrator**: The Administrator expects to monitor, configure, and access notification logs for quality control and policy compliance.	
Preconditions	• The customer has placed an order and provided valid contact information	
Postconditions	• The system successfully sends the requested notifications.	
Main success scenario	Actor (Inventory manager)	System
	The customer places an order and provides	2. The system verifies the order and contact

	contact information.	details.
	3. The system generates notifications for order confirmation, shipping updates, and order delivery.	4. The system sends notifications to the customer via the specified communication channel (e.g., email or SMS).
	5. The customer receives and views the notifications.	6. The system logs the sent notifications for auditing purposes.
Extensions	Actor (Inventory manager)	System
		2a. Invalid Order or Contact Information: If the system identifies invalid order details or contact information, it informs the customer about the issue.
		4a. Notification Sending Failure: If the system encounters issues while sending notifications (e.g., network problems), it informs the customer about the delay and retries.
		4b. Notification Delivery Failure: If notifications cannot be delivered successfully, the system notifies the customer about the delivery failure and provides alternative contact options.

Use Case Name	Backup and Restore Data	
Scope	Pet Inventory Management System	
Level	User Goal	
Primary Actor	System Administrator	
Stakeholders and Interests	• System Administrator: A reliable backup and restoration process for data integrity and system availability, with audit logs for compliance. • Inventory Manager: Seamless data backup and restoration for reliable inventory data, critical for management and order fulfillment. • Customer: Data reliability and availability, indirectly dependent on successful backup and restoration, ensuring a smooth ordering experience.	
Preconditions	The system is operational, and there is data to backup and restore.	
Postcondition	Data is successfully backed up or restored as needed.	
Main success scenario	Actor (Inventory manager)	System
	1. The System Administrator initiates data backup or restoration.	2. The system verifies the administrator's authorization to perform this action.
	3. The system creates a backup of the data or restores data from a previous backup.	4. The system logs the backup or restoration process for auditing purposes.
	5. The System Administrator verifies the backup or restoration's success	
Extensions	Actor (Inventory manager)	System
		2a. Unauthorized Access: If the system determines that the System Administrator is not authorized, it denies access and logs the unauthorized attempt.

		4a. Backup or Restoration Failure: If issues occur during the backup or restoration process (e.g., insufficient storage or data corruption), the system notifies the Administrator of the failure.
		4b. Audit Log Error: In case the system encounters issues while logging the process, it notifies the Administrator of the log error.

Use Case Name	User Authentication and Authorization
Scope	Pet Inventory Management System
Level	User Goal
Primary Actor	System Administrator
Stakeholders and Interests	• System Administrator: Expects to control access to the system, manage user accounts, and ensure system security. • Inventory Manager: Expects secure access to the system for inventory management tasks • Customer: Expects secure access to place orders and manage their account.
Preconditions	The system is operational, and there are user accounts to manage.

Postcondition	The system successfully authenticates and authorizes users, granting appropriate access.	
Main success scenario	Actor (Inventory manager)	System
	1. The System Administrator initiates user authentication or authorization.	2. The system verifies the administrator's authorization to perform this action.
	3. The system authenticates user credentials (e.g., username and password) and authorizes access based on user roles and permissions.	4. The system logs the authentication and authorization process for auditing purposes.
	5. The System Administrator successfully manages user access.	
Extensions	Actor (Inventory manager)	System
		2a. Unauthorized Access: If the system determines that the System Administrator is not authorized, it denies access and logs the unauthorized attempt.
		3a. Invalid Credentials: If user credentials are invalid, the system informs the Administrator, preventing access.
		3b. Role-Based Authorization Failure: If authorization based on user roles and permissions fails, the system denies access and notifies the Administrator.
		4a. Audit Log Error: In case the system encounters issues while logging the process, it notifies the Administrator of the log error.

Use Case Name	Handle Returns and Refunds	
Scope	Pet Inventory Management System	
Level	User Goal	
Primary Actor	User	
Stakeholders and Interests	• Customer: Expects a straightforward process for handling returns and receiving refunds for unsatisfactory pet products. • Inventory Manager: Expects a mechanism to manage returns and refunds efficiently, maintaining customer satisfaction. • System Administrator: Expects the process to adhere to organizational policies and compliance requirements.	
Preconditions	The user is logged into the system, and a product return request has been submitted.	
Postcondition	The return request is processed, and if applicable, a refund is issued to the user's account.	
Main success scenario	Actor (Inventory manager)	System
	1. The user initiates a return request for a pet product.	2. The system verifies the user's authorization to request a return.
	3. The system processes the return request, checks product eligibility, and generates a refund if applicable.	4. The system notifies the user of the return and refund status.
	5. The user receives the refund, if applicable.	
Extensions	Actor (Inventory manager)	System
		2a. Unauthorized Access: If the system determines that the user is not authorized for a return request, it denies access and logs the unauthorized attempt.

		3a. Product Ineligibility: If the product does not meet return criteria, the system informs the user and proceeds accordingly.
		3b. Refund Processing Failure: If the system encounters issues while processing the refund (e.g., payment gateway errors), it notifies the user of the delay.
		4a. Notification Error: In case the system encounters issues while notifying the user, it informs the user about the notification error.

Use Case Name	Process Customer Orders	
Scope	Pet Inventory Management System	
Level	User Goal	
Primary Actor	User	
Stakeholders and Interests	• Customer: Expects a user-friendly process to create and manage orders for pet products, ensuring a seamless shopping experience. • Inventory Manager: Expects an efficient system to receive and fulfill customer orders accurately. • System Administrator: Expects order processing to align with organizational policies and compliance requirements.	
Preconditions	The user is logged into the system, and pet products are available in the inventory for ordering.	
Postcondition	The order is successfully created and recorded in the system, and the inventory is updated accordingly.	
Main success scenario	Actor (Inventory manager)	System

	. The user initiates an order for pet products, selecting items and specifying quantity.	2. The system verifies the user's authorization to create an order.
	3. The system records the order details, checks product availability, and updates the inventory.	4. The system generates an order confirmation for the user and notifies the user of the successful order.
	5 The user reviews the order confirmation and receives a delivery estimate.	
Extensions	Actor (Inventory manager)	System
		2a. Unauthorized Access: If the system determines that the user is not authorized to create orders, it denies access and logs the unauthorized attempt.
		3a. Product Unavailability: If a selected product is out of stock or unavailable, the system informs the user and proceeds accordingly.
		3b. Inventory Update Failure: If the system encounters issues while updating inventory (e.g., database errors), it notifies the user of the delay.
		4a. Order Confirmation Error: In case the system encounters issues while generating the order confirmation, it informs the user about the error.

Use Case Name	Set Low Stock Alerts	
Scope	Pet Inventory Management System	
Level	User Goal	
Primary Actor	Inventory Manager	
Stakeholders and Interests	• Inventory Manager: Expects to set low stock alerts for pet products to prevent stockouts and ensure timely replenishment. • System Administrator: Expects alert settings to adhere to organizational policies and compliance requirements.	
Preconditions	The Inventory Manager is logged into the system, and there are pet products with available stock.	
Postcondition	The low stock alerts are successfully configured in the system.	
Main success scenario	Actor (Inventory manager)	System
	1. The Inventory Manager accesses the system to set low stock alerts for specific pet products.	2. The system verifies the Inventory Manager's authorization to configure alerts.
	3. The Inventory Manager specifies the threshold quantity for each product, below which alerts should be triggered.	4. The system saves the alert settings for the selected products.
	5. The system monitors inventory levels and triggers alerts when stock falls below the specified threshold.	6. The Inventory Manager receives notifications about low stock for the configured products.
Extensions	Actor (Inventory manager)	System
	-	2a. Unauthorized Access: If the system determines that the Inventory Manager is not authorized to configure alerts, it denies access and logs the unauthorized attempt.
		4a. Alert Configuration Error: In case the

	-	system encounters issues while saving alert settings, it informs the Inventory Manager about the configuration error.
	-	6a. Notification Error: If the system encounters issues while sending low stock alerts, it informs the Inventory Manager about the notification error.

Use Case Name	Update Pet Product Information	
Scope	Pet Supply Inventory Management System	
Level	System goal	
Primary actor	Inventory manager	
Stakeholders and interests	1. **Inventory Manager**: Wants to update pet products efficiently to maintain an accurate inventory. 2. **Customer**: can get full information of the product 3. **System administrator**: Relies on accurate inventory data for decision-making	
Preconditions	1. The Inventory Manager has logged into the system. 2. The system is up and running and accessible. 3. The Inventory Manager is authorized to update available pet products.	
Postconditions	1. The pet product has successfully been updated to the inventory. 2. The modifications have been recorded in the database.	
Main success scenario	Actor (inventory manager)	System
	1.Select the "update pet product information" option in the system 3.choose the pet product you need to update 4.review and update the product in detail e.g. name, price etc. 5.submit updated information 6.recieve confirmation 7.log out of system	2.presents a list of pet products available that could be modified.
Extensions	Actor (Inventory manager)	System

	3a. If the Inventory Manager chooses to cancel the update at any point.	3b. system discards the changes made, and the product information remains unchanged.
	4a. the validation of the modified information fails. the system displays error messages. 4b. the Inventory manager is prompted to correct the information.	
		5a. if issue with updating the database, system notifies inventory manager of the problem
	5b. manager alerts IT department about the problem	

Use Case Name	Add a new pet product.	
Scope	Pet Supply Inventory Management System	
Level	User goal	
Primary actor	Inventory manager	
Stakeholders and interests	1. **Inventory Manager**: Wants to add pet products efficiently to maintain an accurate inventory. 2. **customer:** product is now available to buy 3. **System administrator**: Relies on accurate inventory data for decision-making	
Preconditions	4. The Inventory Manager has logged into the system. 5. The system is up and running and accessible. 6. The Inventory Manager is authorized to add pet products that are available.	
Postconditions	3. The inventory system's pet product information has been successfully updated. 4. The modifications have been recorded in the database.	
Main success scenario	Actor (inventory manager)	System
	1.inventory manager request to add product	2.system authorizes inventory manager to perform the action.
	3.system tells manager to provide additional detail of the new product.	4.system validates the input data for accuracy and checks for any existing product with same name.
	5.The manager confirms the addition of the new pet product	6.system adds the product to the database.
Extensions	Actor (Inventory manager)	System

		2a. Access: the manager is not authorized to enter and denies access. 4a. if same named product exists, the system will inform the inventory manger and does not duplicate product.

Use Case Name	Search for a pet product.	
Scope	Pet Supply Inventory Management System	
Level	User goal	
Primary actor	Inventory manager and customer	
Stakeholders and interests	1. **Inventory Manager**: Wants to search pet products efficiently to maintain an accurate inventory. 2. **Customer:** Wants to search for pet products to place orders or view details 3. **System Administrator:** Relies on accurate inventory data for decision-making	
Preconditions	1. The Inventory Manager has logged into the system. 2. The system is up and running and accessible.	
Postconditions	1. The customer has searched and found the pet product. 2. The system remains the same after search as it was before search.	
Main success scenario	Actor (inventory manager)	System
	1.User initiates search for a pet product. 3.user provides keywords or price range for a system to a search 5.user reviews the search result	2.System displays the search screen 4.system processes the search criteria and displays a list of pet products that meet the requirement of the search criteria.
Extensions	Actor (Inventory manager)	System

		3a. There is no result for the products the user has entered in the inventory. 4a. if the user provides invalid search criteria the system asks the user to correct the input.

Use Case Name	Generate inventory reports.	
Scope	Pet Supply Inventory Management System	
Level	System.	
Primary actor	Inventory manager	
Stakeholders and interests	1. **Inventory Manager**: Wants to generate reports to manage inventory efficiently. 2. **retailer**: benefit as they need to restock products or remove products according to report 3. **Administrator:** Relies on report for decision making regarding inventory	
Preconditions	1. The Inventory Manager has logged into the system. 2. The system is up and running and accessible.	
Postconditions	1. The system generates and provides the reports of the requested inventory 2. The system remains in the same state as before the report generation.	
Main success scenario	Actor (inventory manager)	System
	1.Inventory manger request to generate report 3. Inventory Manager selects the type of report to generate 5.The Inventory Manager receives and reviews the generated report.	2.System displays the report generation screen 4.system generates the selected report based on the current data
Extensions	Actor (Inventory manager)	System

	1a.The Inventory Manager can cancel the report generation at any point, returning to the previous state without receiving or using the report.	3a. If the Inventory Manager selects an invalid or unavailable report type, the system prompts the user to choose a valid report type. 4a. If there is no relevant data to include in the report, the system informs the Inventory Manager that no data was found for the selected report type.

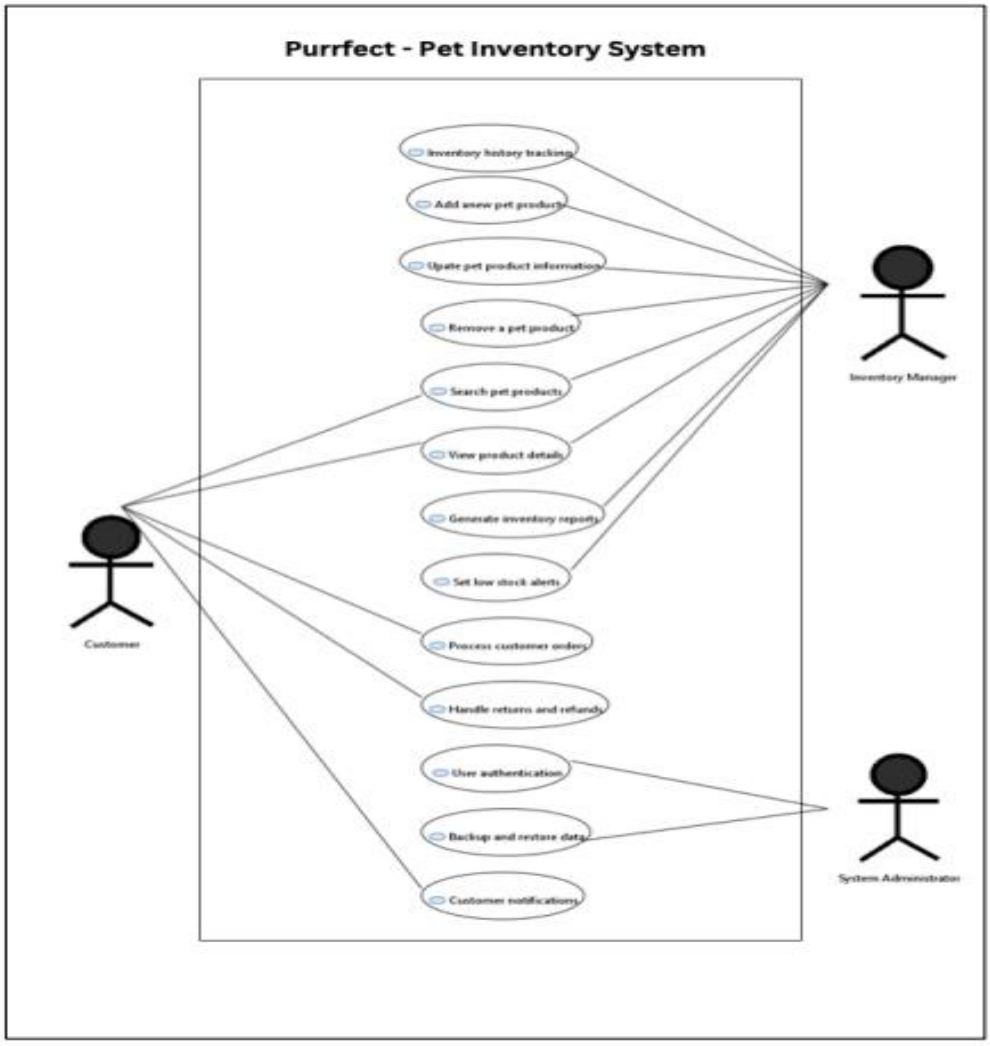
Use Case Name	Remove a pet product.	
Scope	Pet Supply Inventory Management System	
Level	User goal	
Primary actor	Inventory manager	
Stakeholders and interests	1. **Inventory Manager**: Wants to add pet products efficiently to maintain an accurate inventory. 2. **customer:** product is now available to buy 3. **System administrator**: Relies on accurate inventory data for decision-making	
Preconditions	7. The Inventory Manager has logged into the system. 8. The system is up and running and accessible. 9. The Inventory Manager is authorized to remove pet products that are available.	
Postconditions	5. The inventory system's pet product information has been successfully updated. 6. The modifications have been recorded in the database.	
Main success scenario	Actor (inventory manager)	System
	1.inventory manager request to remove product	2.system authorizes inventory manager to perform the action and starts monitoring inventory changes. 3.The system confirms with the inventory manager that they want to proceed with the removal.
	4. Inventory manager confirms the removal.	5. System update the inventory database and removes the selected item 6.system informs the manager that product has been successfully removed.
Extensions	Actor (Inventory manager)	System

	1a. If the system encounters an error during the removal process, manager performs task manually.	3a. If the submitted product information is invalid or do not match any current product, the system notifies the Inventory Manager and requests that proper details be provided.

Use Case Name	View product detail	
Scope	Pet Supply Inventory Management System	
Level	User goal	
Primary actor	Inventory manager and customer	
Stakeholders and interests	1. **Inventory Manager**: Wants to view product details for inventory management. 2. **customer:** Wants to view product details before making a purchase decision 3. **System administrator**: Relies on accurate inventory data for decision-making	
Preconditions	1. The Inventory Manager has logged into the system. 2. The system is up and running and accessible.	
Postconditions	1. The actor has successfully viewed the product information 2. The system remains in the same state as it was before	
Main success scenario	Actor (inventory manager)	System
	1. User selects a pet product to view details. 3. The user reviews the product details, including name, description, price, availability, and any associated images. 4. The user can choose to take further actions, such as adding the product to the shopping cart or returning to the product list	2. The system retrieves and displays detailed product information. 5. the system responds to the user's act allowing them to proceed with additio actions or return to browsing.

Extensions	Actor (Inventory manager)	System
		1a.the user attempts to view product d without selecting a specific product, t system prompts the user to choose a product. 2a. If the selected product is not found the system
	4a. The actor can choose to cancel the product details view at any point, returning to the previous state without viewing all the details	

2.5 Use Case Diagram



3 Other Nonfunctional Requirements

3.1 Performance Requirements

I. Inventory Tracking Performance: The system must efficiently track inventory changes, updating the inventory history log in real-time as changes occur.
II. Notification System Performance: The system must send notifications (order confirmations, shipping updates, etc.) to customers promptly, with minimal delay post-order placement.
III. Data Backup and Restoration: The system should perform data backups and restorations swiftly, ensuring data integrity with minimal downtime.

3.2 Safety Requirements

I. Data Handling Safety: The system must handle customer and inventory data securely, preventing any data loss or corruption.
II. Operational Safety: The system should have safeguards against operational errors that could lead to inventory mismanagement or customer dissatisfaction.
III. Regulatory Compliance: The system must comply with relevant data protection and privacy regulations, such as GDPR for European customers.

3.3 Security Requirements

I. User Authentication and Authorization: The system should have robust user authentication and role-based authorization mechanisms to secure access to sensitive information.

II. Data Encryption: Sensitive data transmitted by the system, such as customer details and transaction records, should be encrypted.
III. Audit Trails: The system must maintain audit trails for critical actions (inventory changes, user logins, etc.) for security and compliance purposes.

3.4 Software Quality Attributes

I. Usability: The system must be user-friendly, with an intuitive interface for both customers and inventory managers.
II. Reliability: The system should have a high uptime, with minimal downtime for maintenance or updates.
III. Adaptability: The system should be adaptable to varying business needs and scalable as the pet store grows.
IV. Interoperability: The system should be compatible with other software tools used in pet stores, such as POS systems and e-commerce platforms.

3.5 Business Rules

I. Inventory Management: Only authorized inventory managers should be able to add, remove, or update inventory items.
II. Customer Interaction: The system should facilitate customer interactions, such as order processing, returns, and refunds, in an efficient and user-friendly manner.

3.6 Operating Environment

I. Hardware Compatibility: The system should be compatible with standard computing hardware used in retail environments.
II. Software Compatibility: It should run on commonly used operating systems like Windows and macOS and be accessible through standard web browsers.

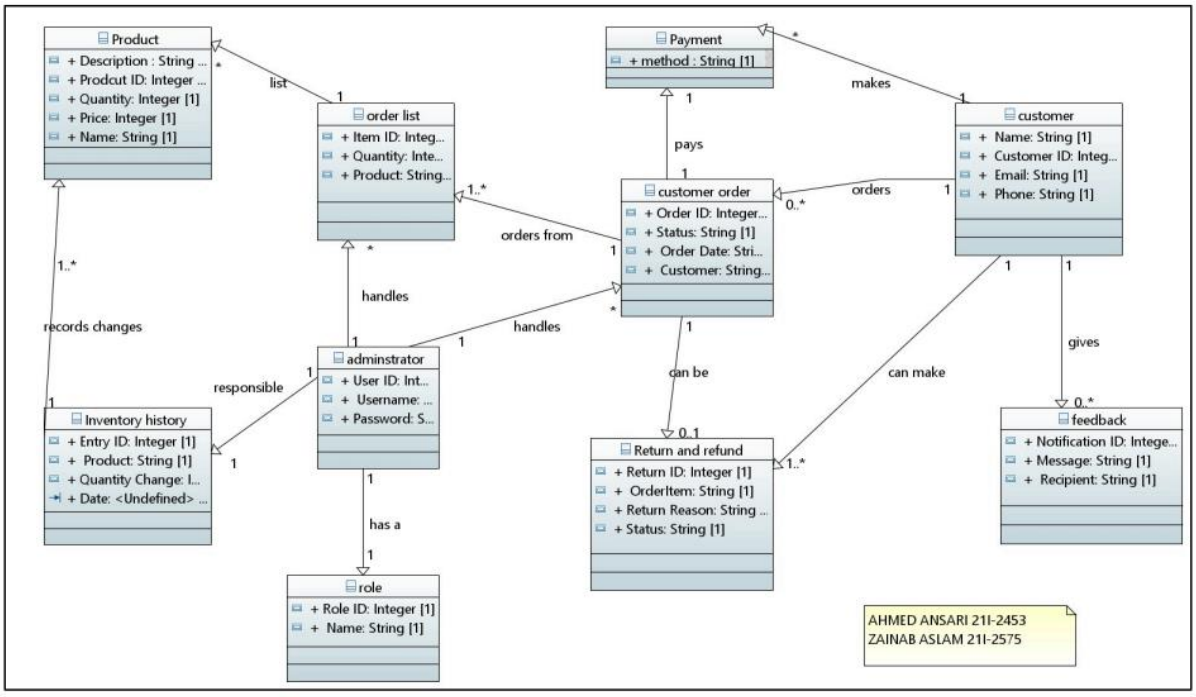
3.7User Interfaces

I. Interface Design: The system should have a clear and concise interface, with easy navigation for various user roles (inventory managers, system administrators, customers).

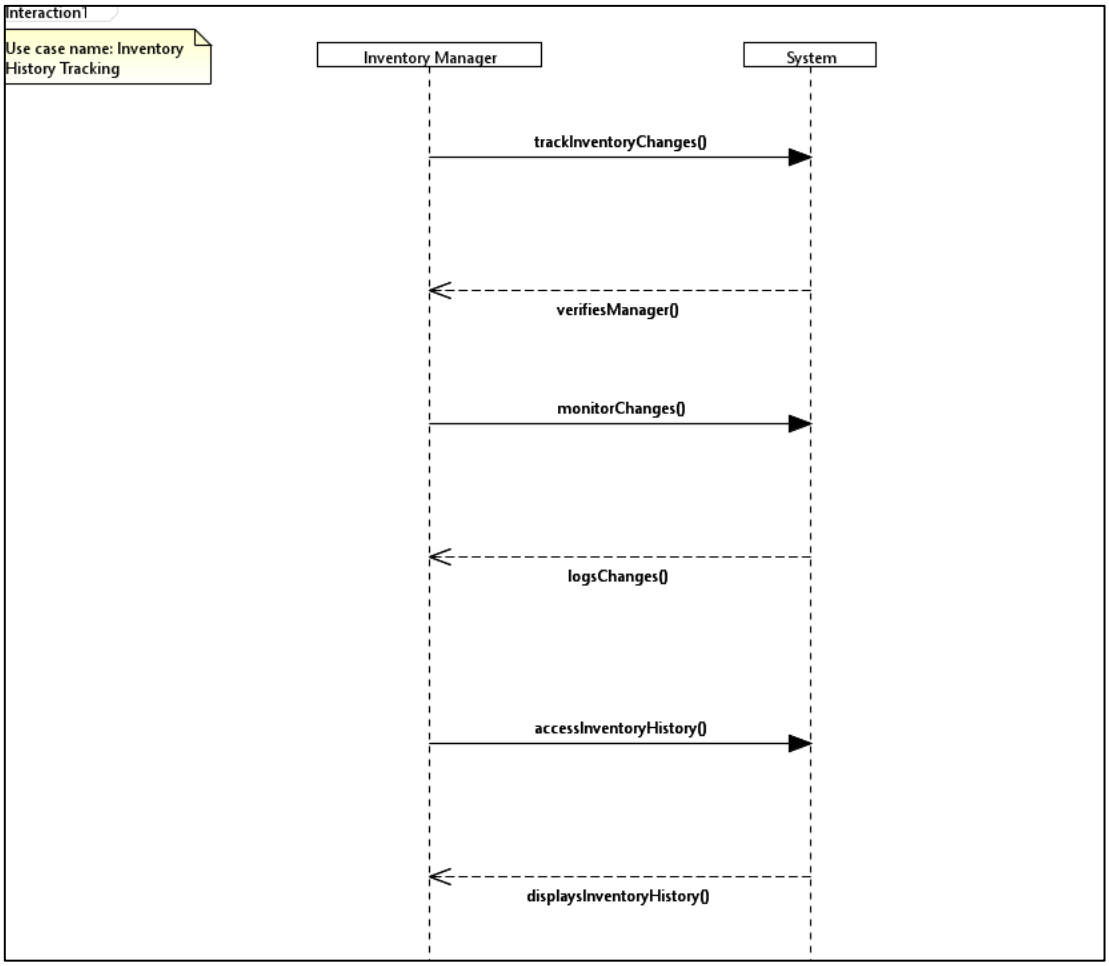
II. Accessibility Features: The interface should include accessibility features for users with disabilities.

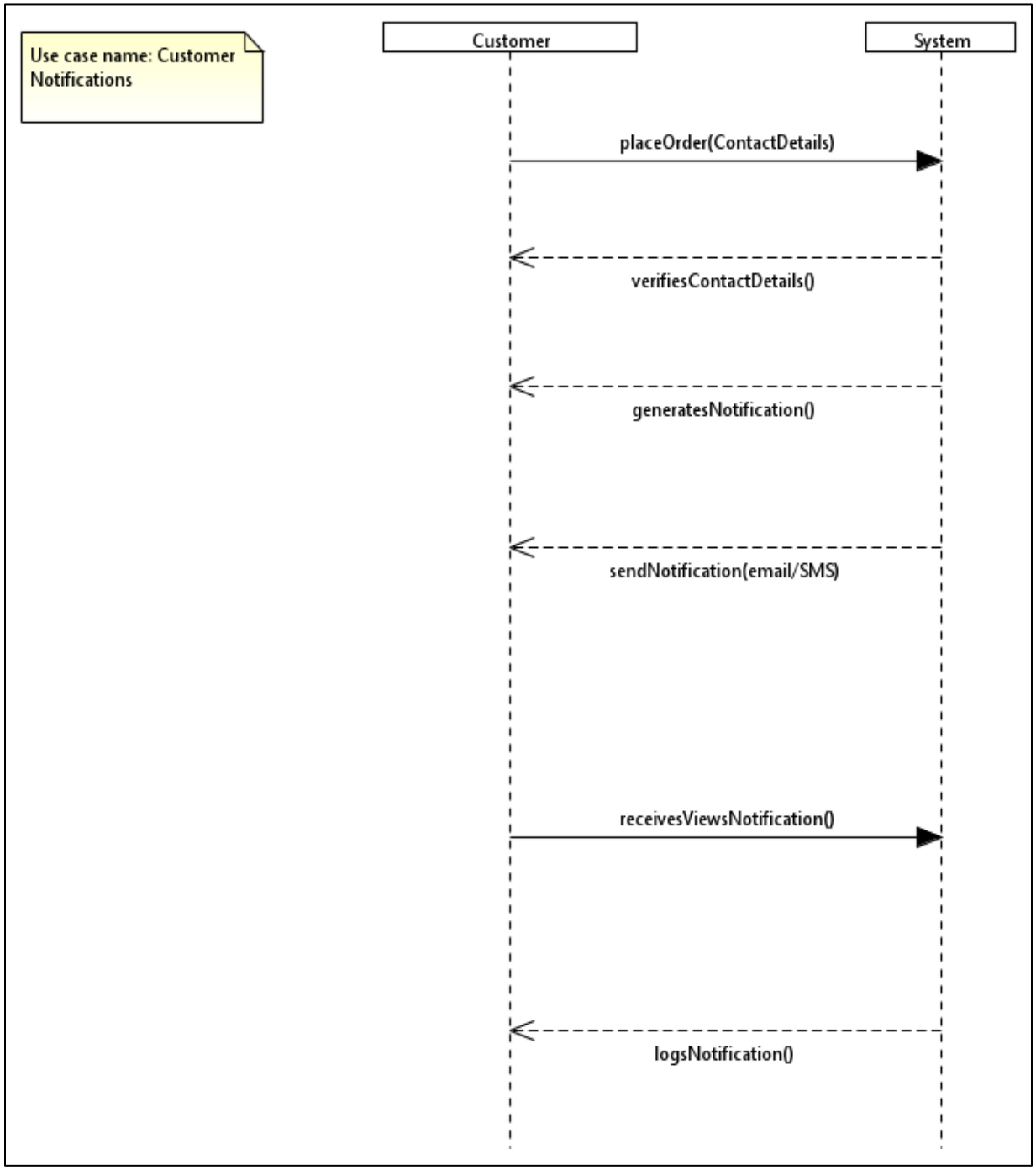
III. Error Messaging: Clear and informative error messages should be displayed for any user input errors or system issues.

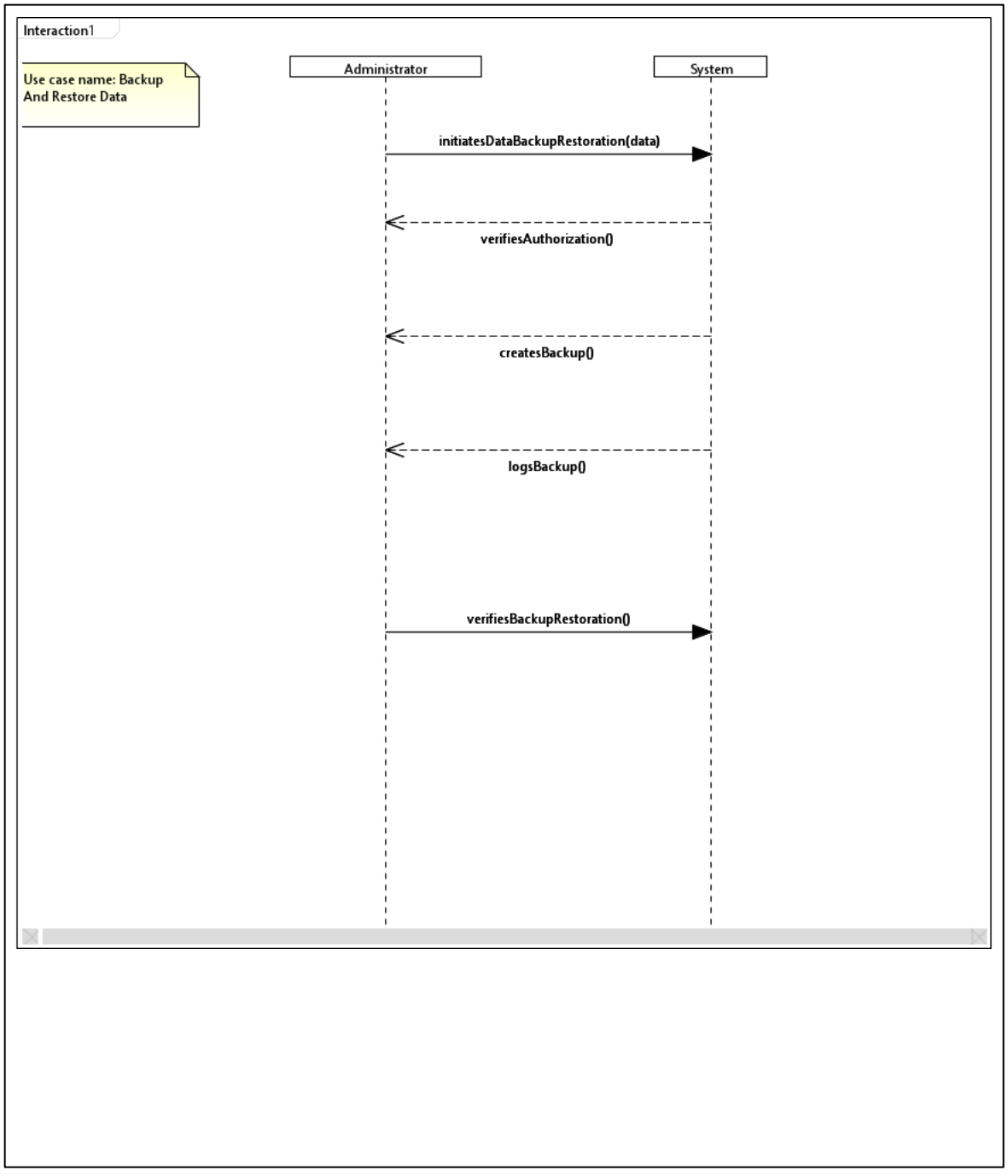
4.Domain Model

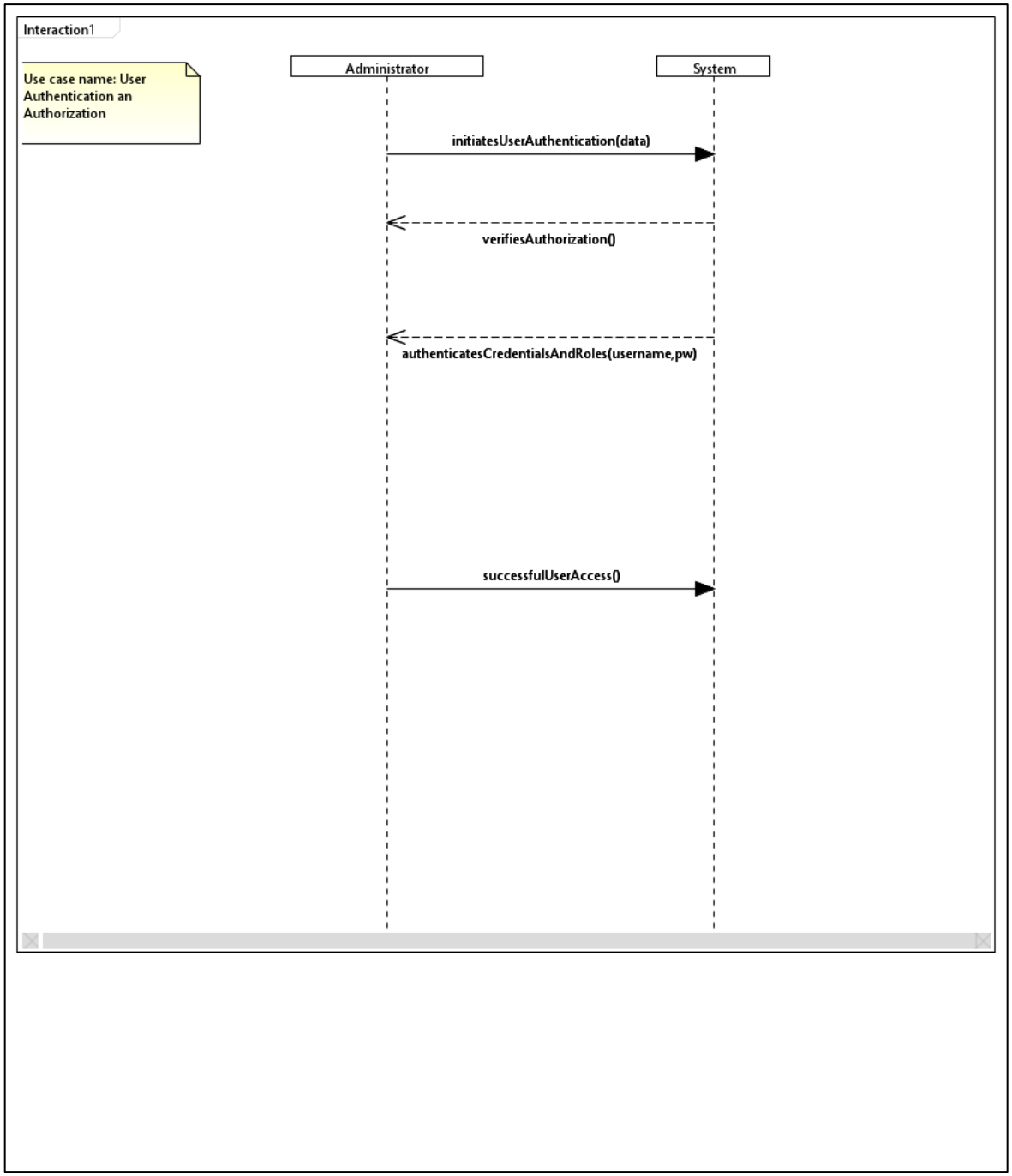


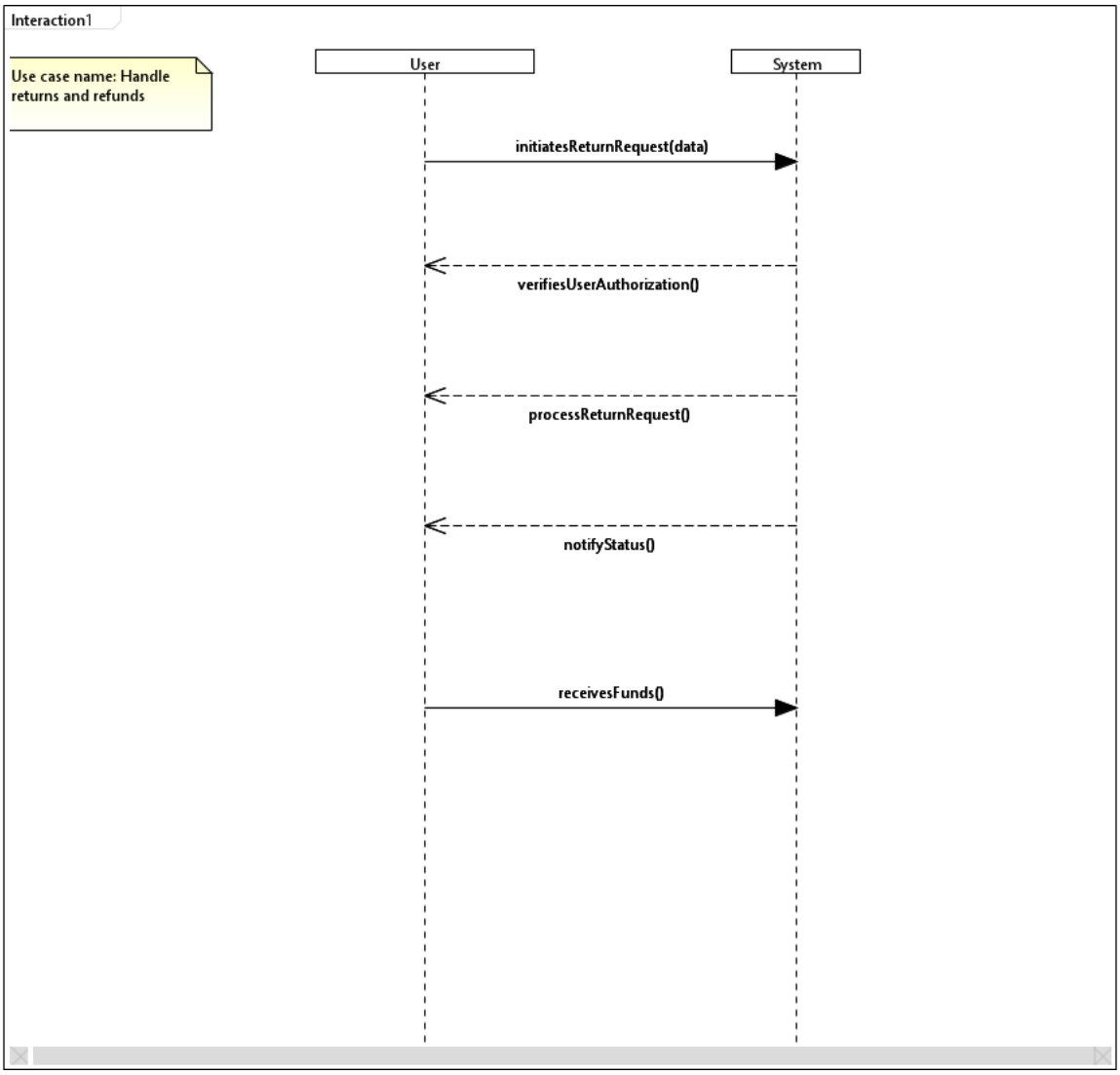
5.System Sequence Diagram

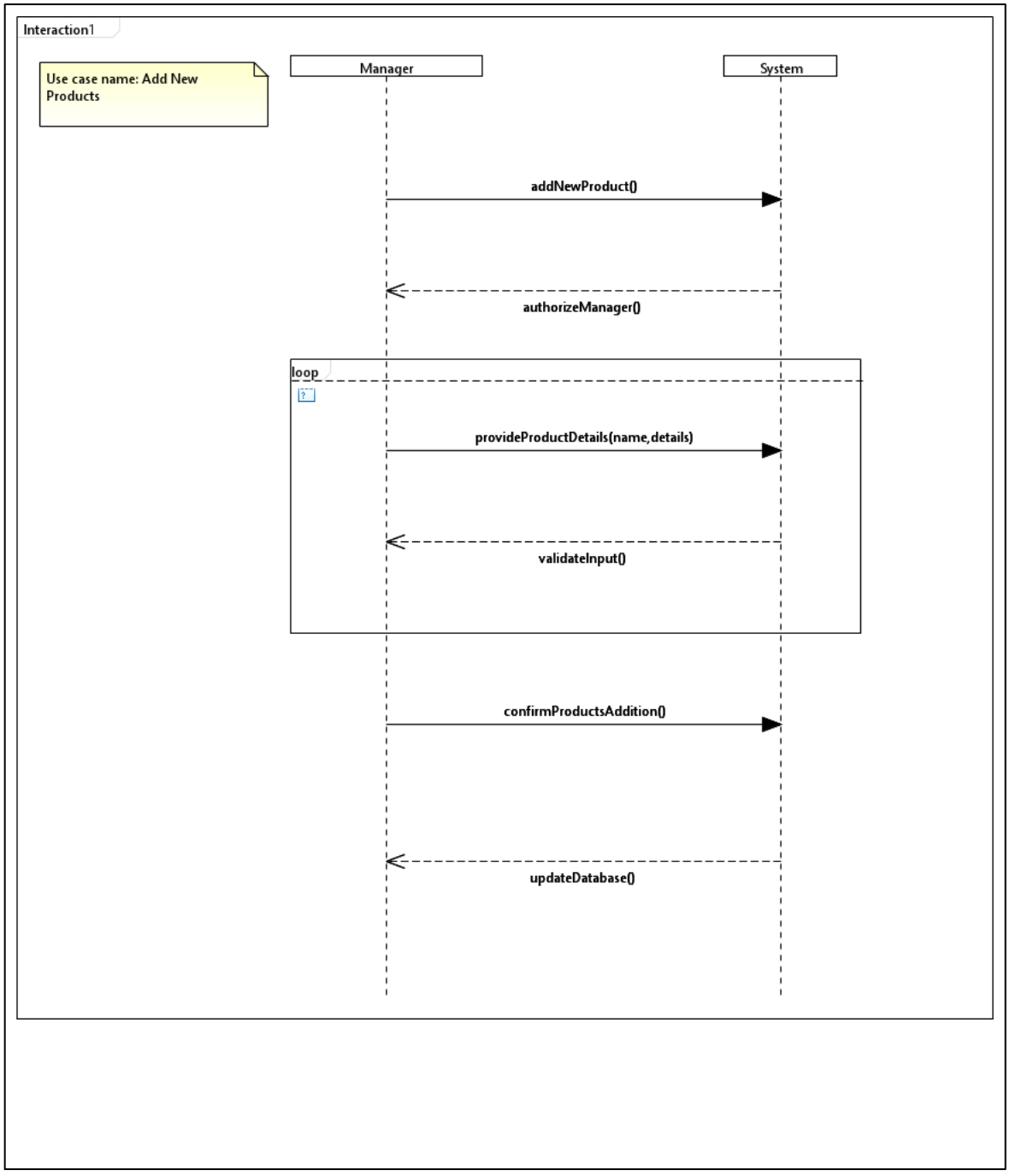



Use case name: Customer Notifications
Customer
System
placeOrder(ContactDetails)
verifiesContactDetails()
generatesNotification()
sendNotification(email/SMS)
receivesViewsNotification()
logsNotification()

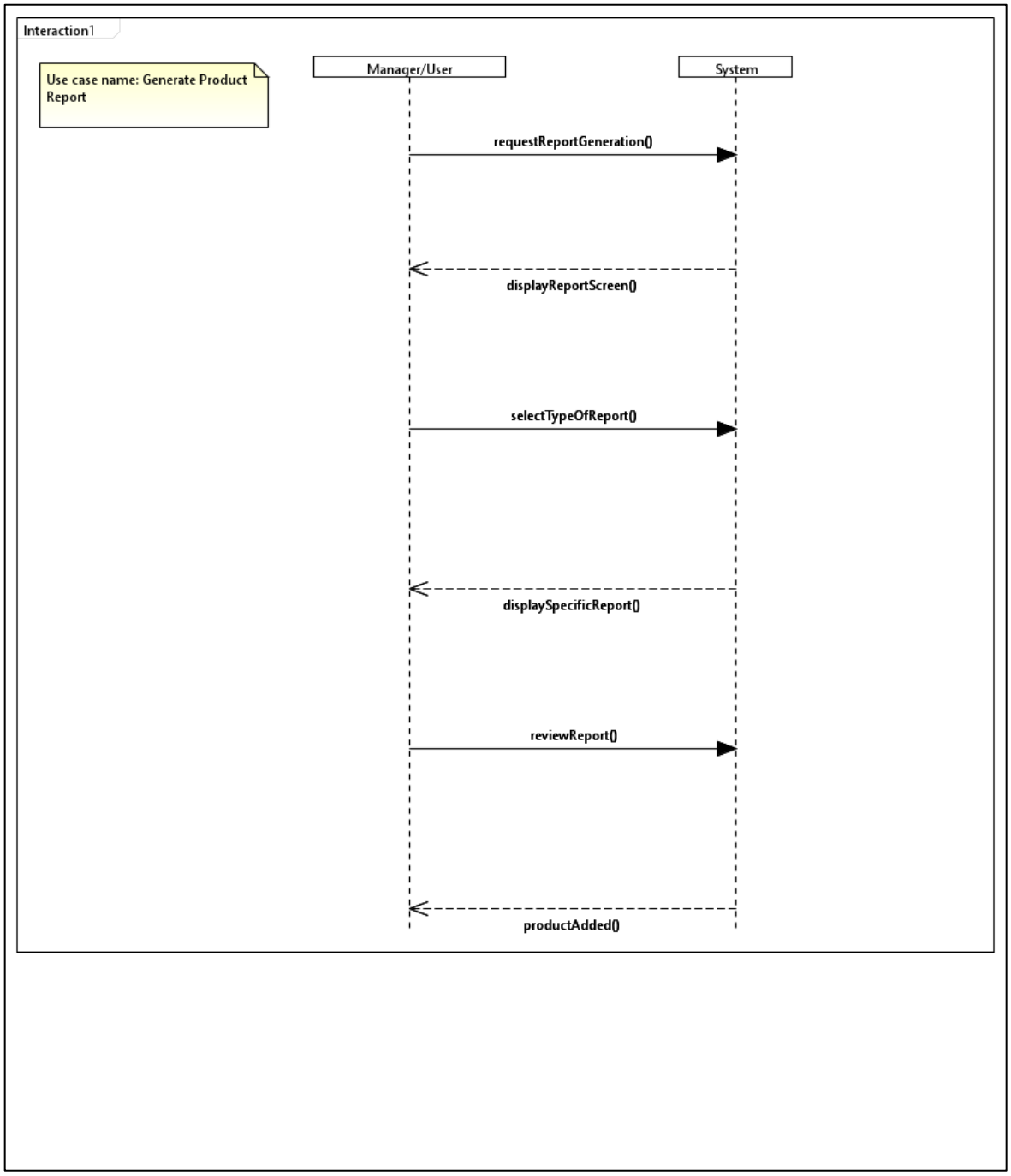

Interaction1
Use case name: Backup And Restore Data
Administrator
System
initiatesDataBackupRestoration(data)
verifiesAuthorization()
createsBackup()
logsBackup()
verifiesBackupRestoration()

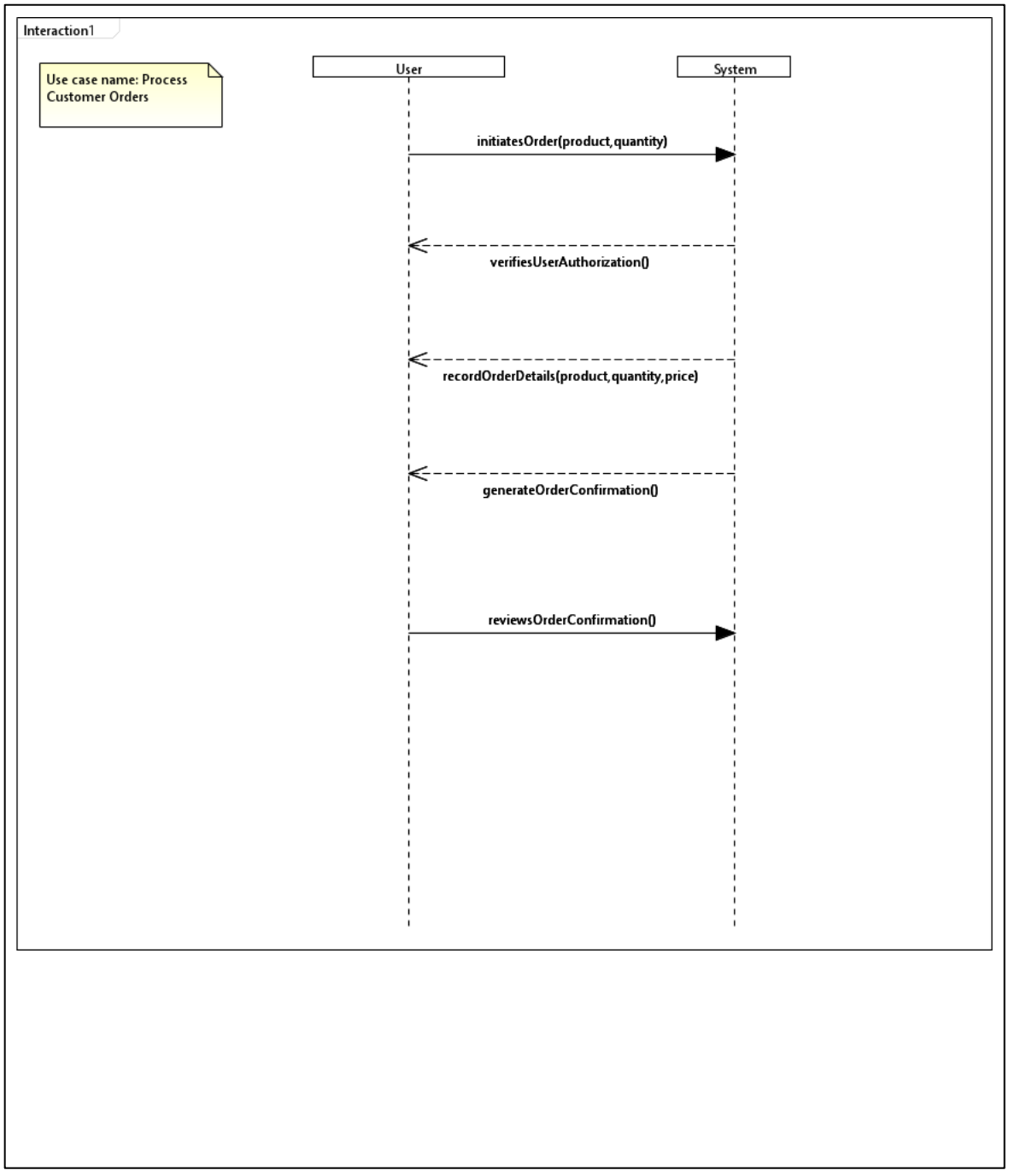

Interaction1
Use case name: User Authentication an Authorization
Administrator
System
initiatesUserAuthentication(data)
verifiesAuthorization()
authenticatesCredentialsAndRoles(username,pw)
successfulUserAccess()

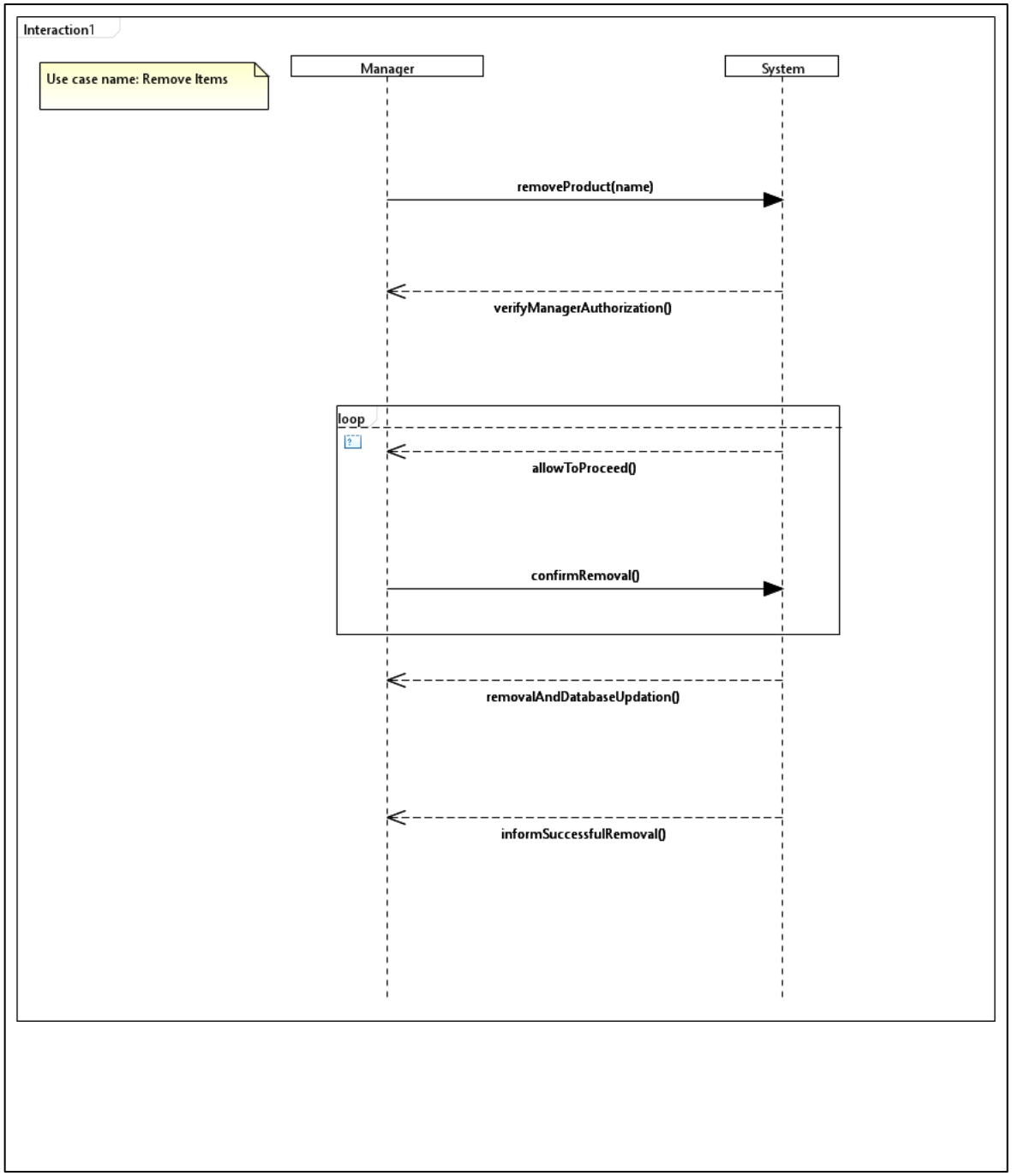

Interaction1
Use case name: Handle returns and refunds
User
System
initiatesReturnRequest(data)
verifiesUserAuthorization()
processReturnRequest()
notifyStatus()
receivesFunds()

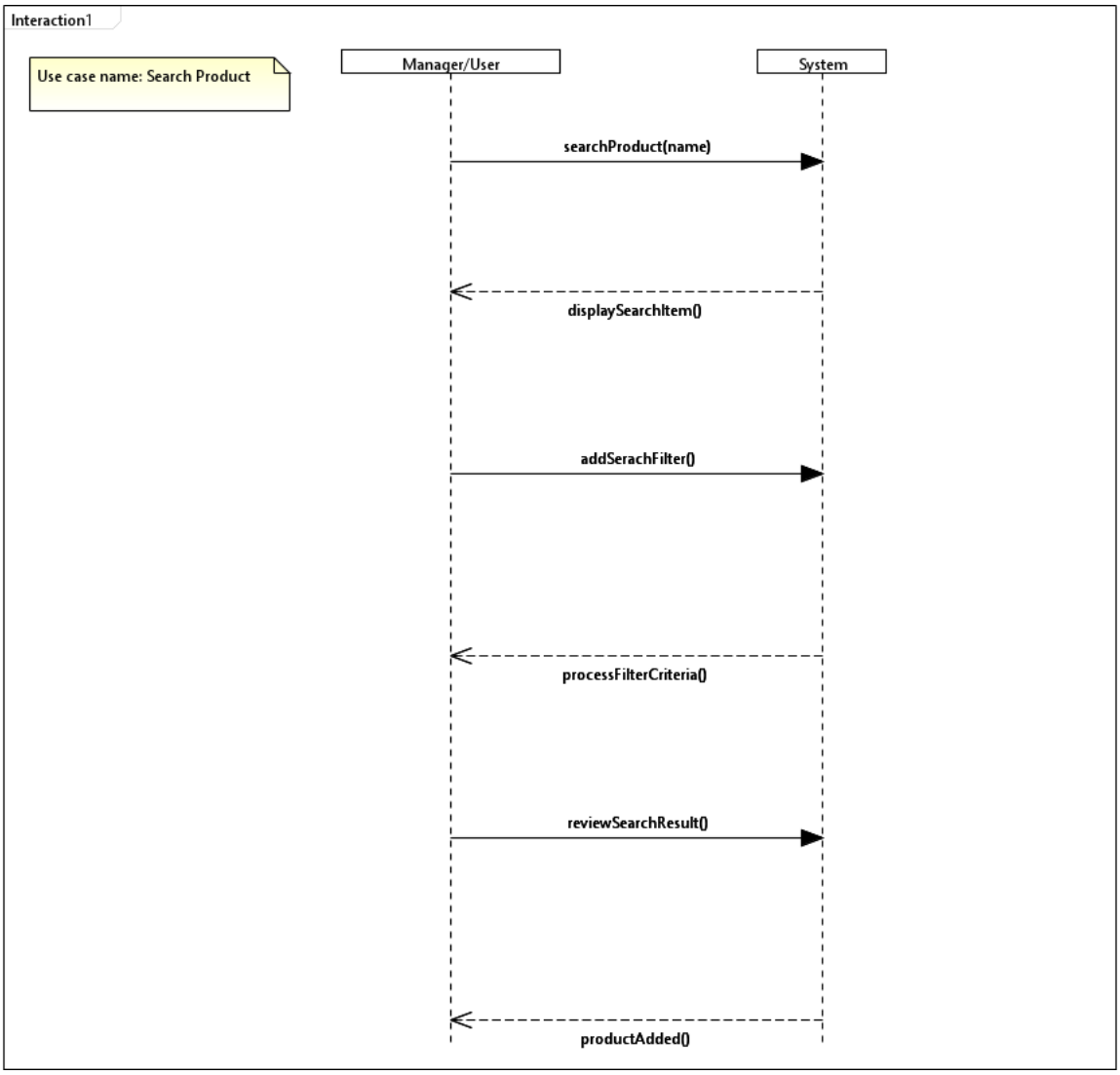

Interaction1
Use case name: Add New Products
Manager
System
addNewProduct()
authorizeManager()
loop
provideProductDetails(name,details)
validateInput()
confirmProductsAddition()
updateDatabase()

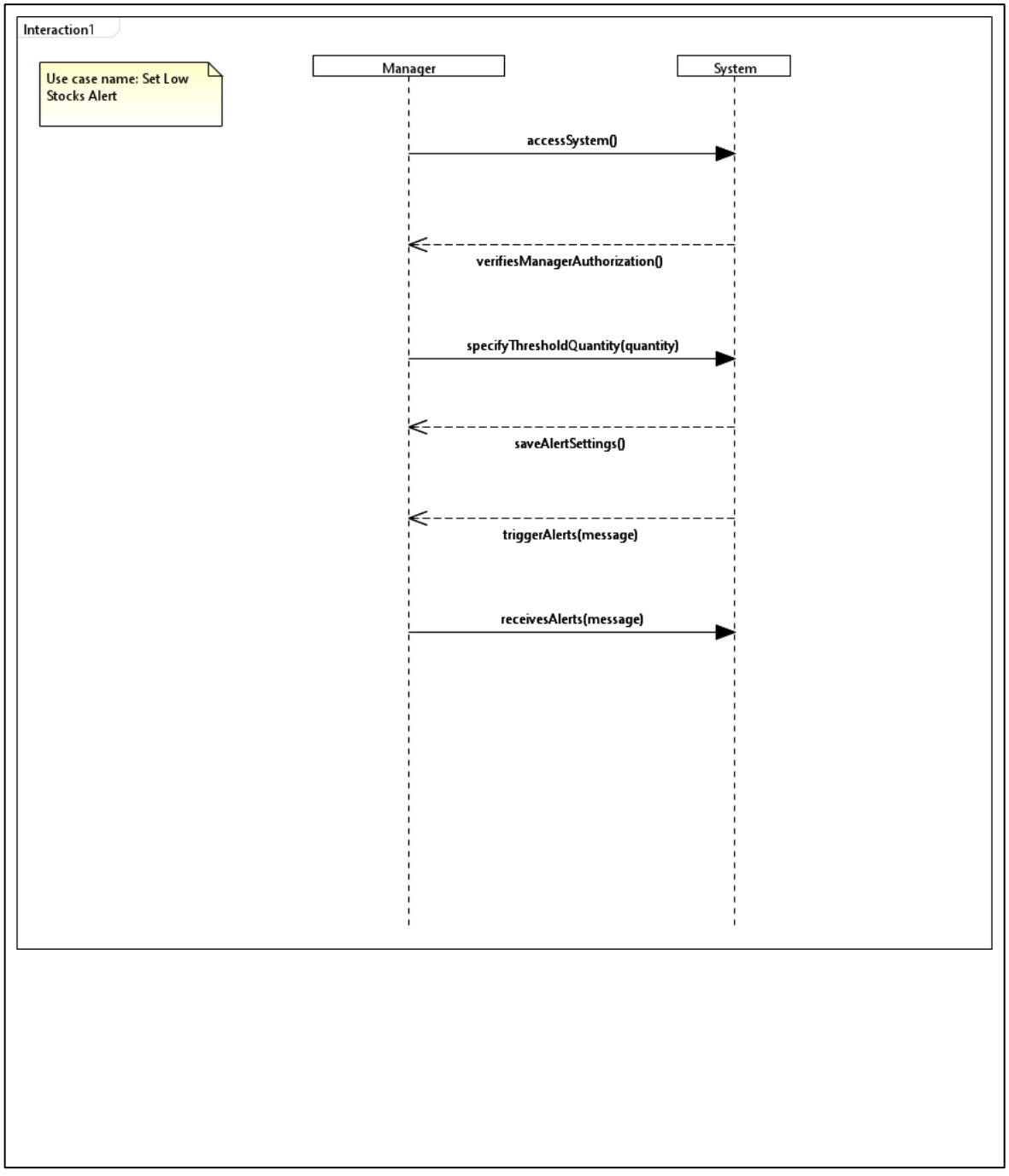
=

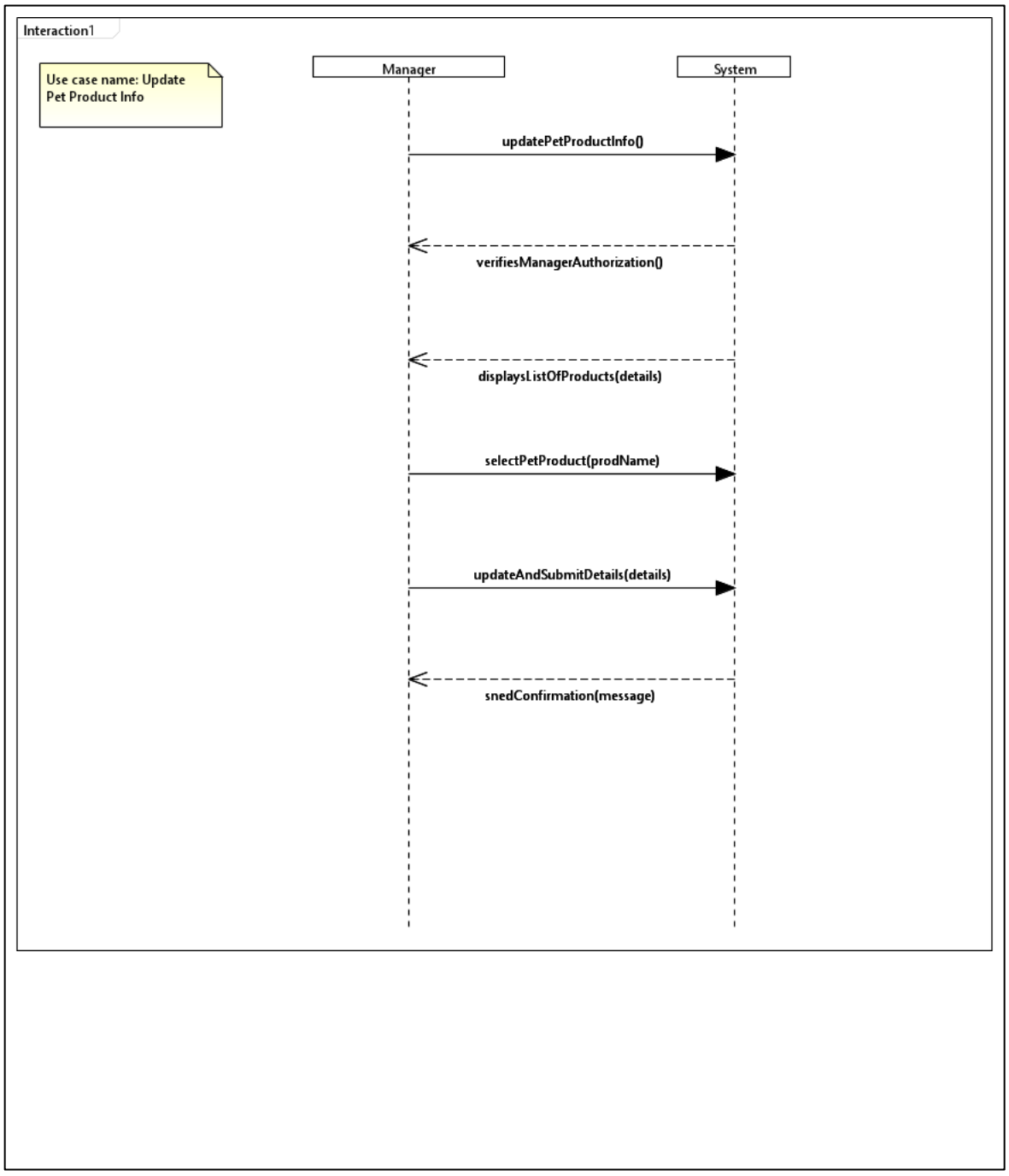

Interaction1
Use case name: Generate Product Report
Manager/User
System
requestReportGeneration()
displayReportScreen()
selectTypeOfReport()
displaySpecificReport()
reviewReport()
productAdded()


Interaction1
Use case name: Process Customer Orders
User
System
initiatesOrder(product, quantity)
verifiesUserAuthorization()
recordOrderDetails(product, quantity, price)
generateOrderConfirmation()
reviewsOrderConfirmation()

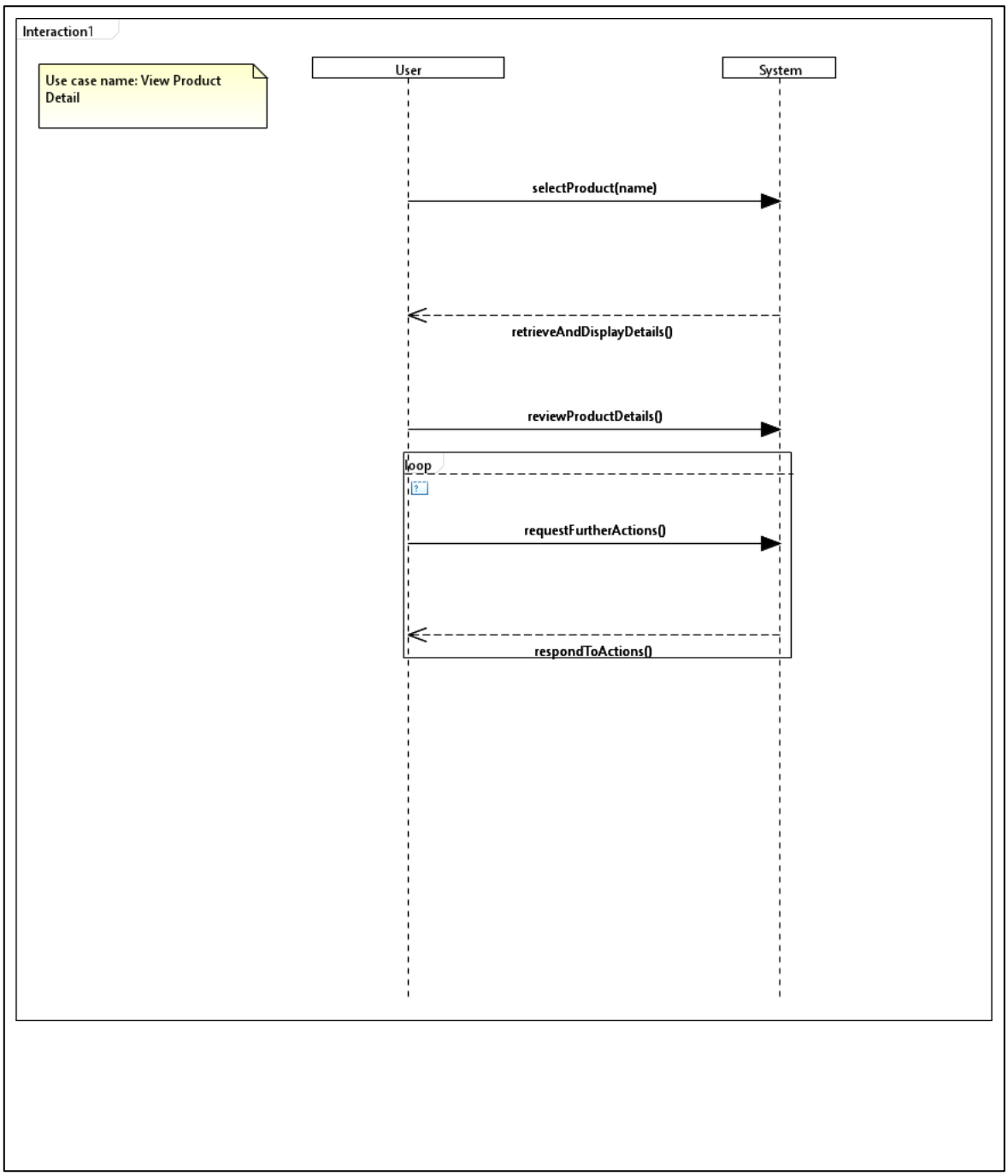

Interaction1
Use case name: Remove Items
Manager
System
removeProduct(name)
verifyManagerAuthorization()
loop
allowToProceed()
confirmRemoval()
removalAndDatabaseUpdation()
informSuccessfulRemoval()


Interaction1
Use case name: Search Product
Manager/User
System
searchProduct(name)
displaySearchItem()
addSerachFilter()
processFilterCriteria()
reviewSearchResult()
productAdded()


Interaction1
Use case name: Set Low Stocks Alert
Manager
System
accessSystem()
verifiesManagerAuthorization()
specifyThresholdQuantity(quantity)
saveAlertSettings()
triggerAlerts(message)
receivesAlerts(message)

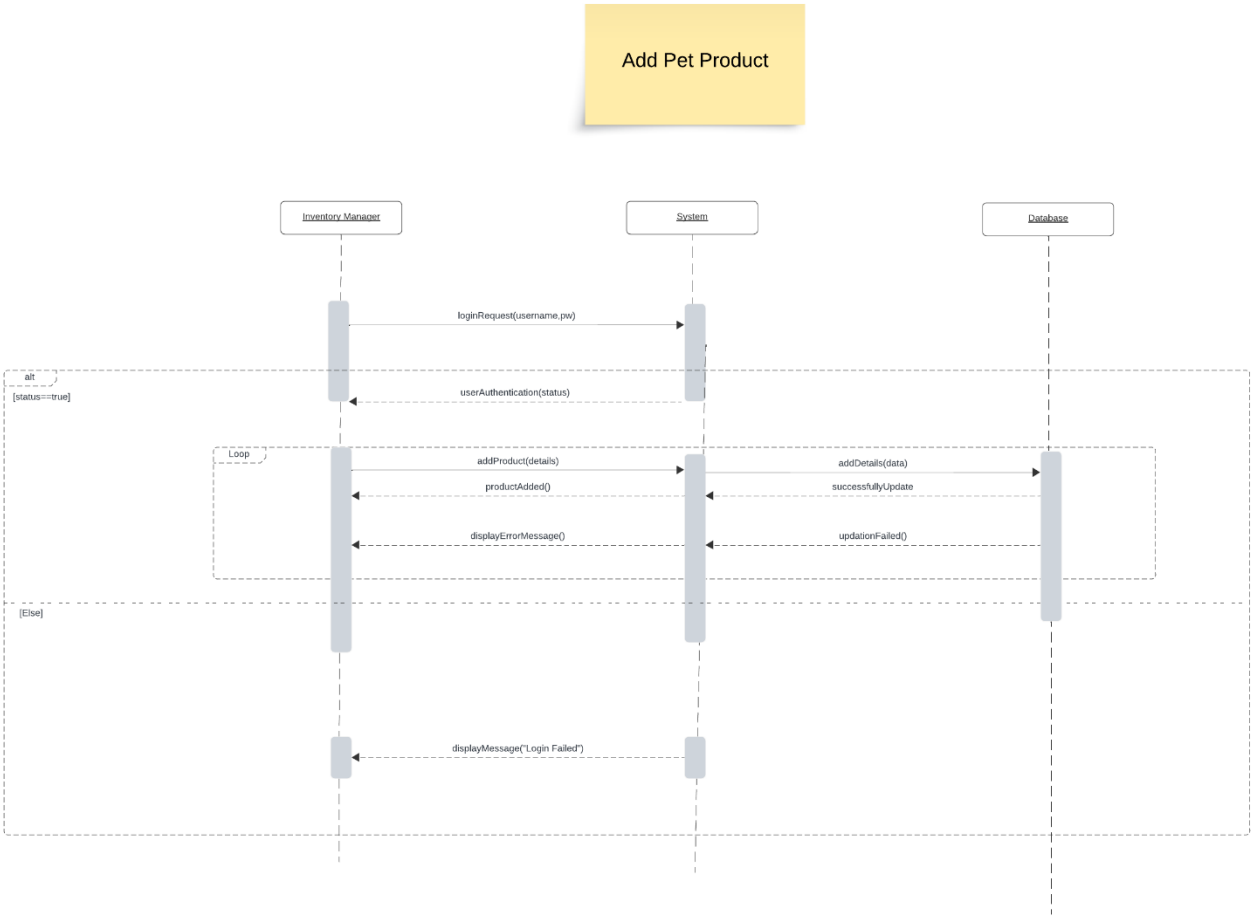

Interaction1
Use case name: Update Pet Product Info
Manager
System
updatePetProductInfo()
verifiesManagerAuthorization()
displaysListOfProducts(details)
selectPetProduct(prodName)
updateAndSubmitDetails(details)
snedConfirmation(message)

/

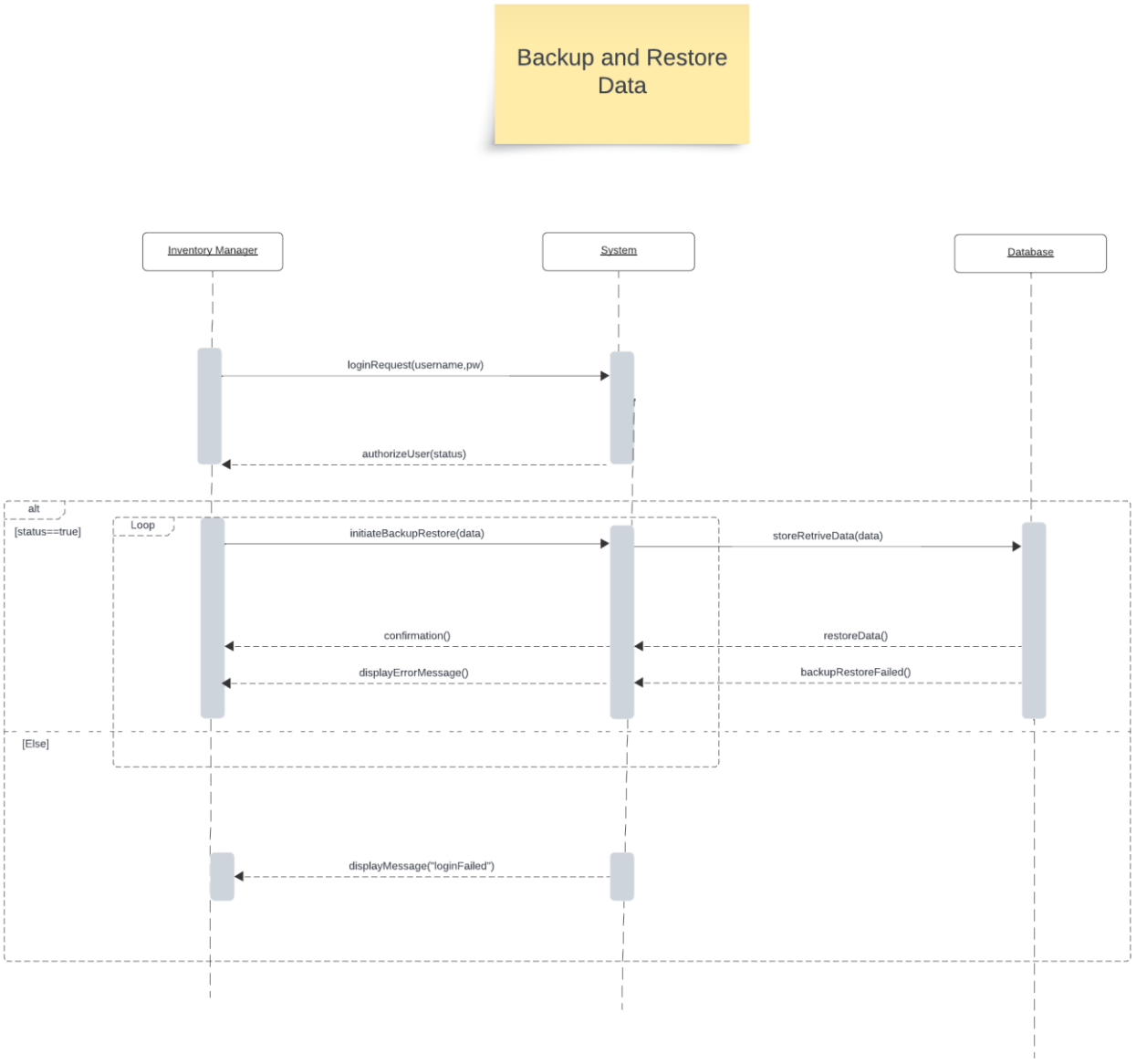

Interaction1
Use case name: View Product Detail
User
System
selectProduct(name)
retrieveAndDisplayDetails()
reviewProductDetails()
loop
requestFurtherActions()
respondToActions()

/

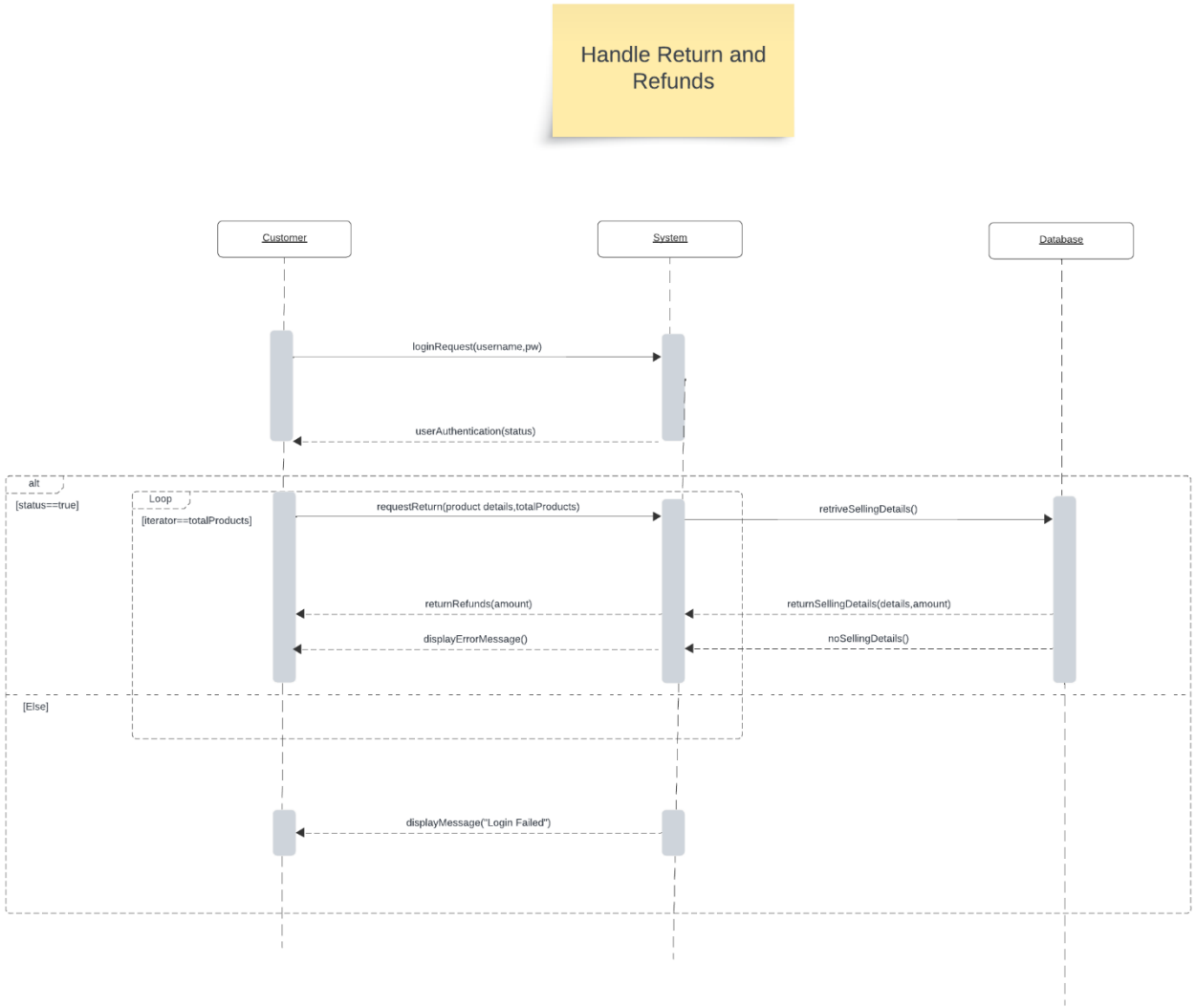
6.Sequence Diagram



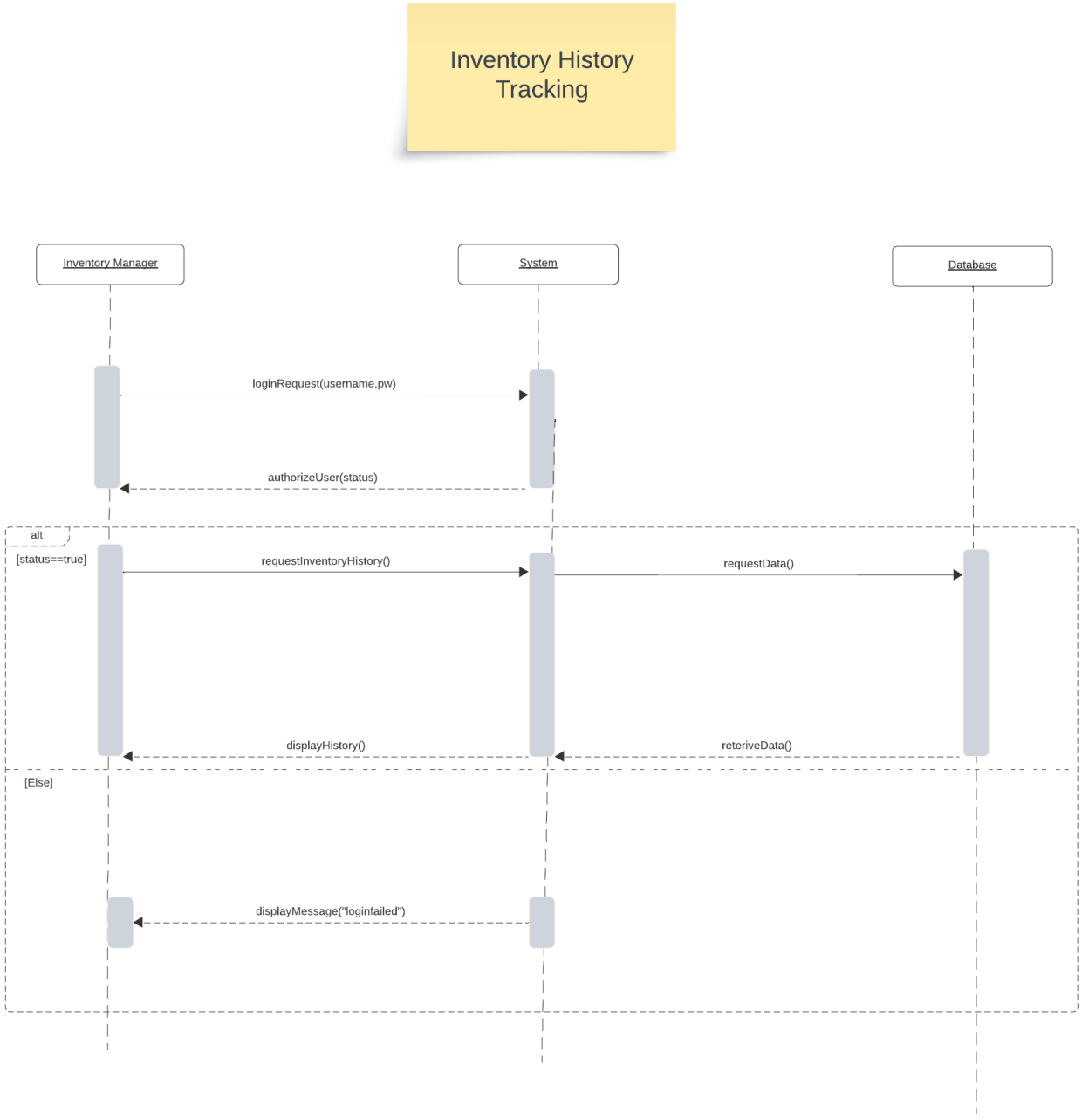
/


Backup and Restore Data
Inventory Manager
System
Database
loginRequest(username,pw)
authorizeUser(status)
alt
[status==true]
Loop
initiateBackupRestore(data)
storeRetriveData(data)
confirmation()
restoreData()
displayErrorMessage()
backupRestoreFailed()
[Else]
displayMessage("loginFailed")

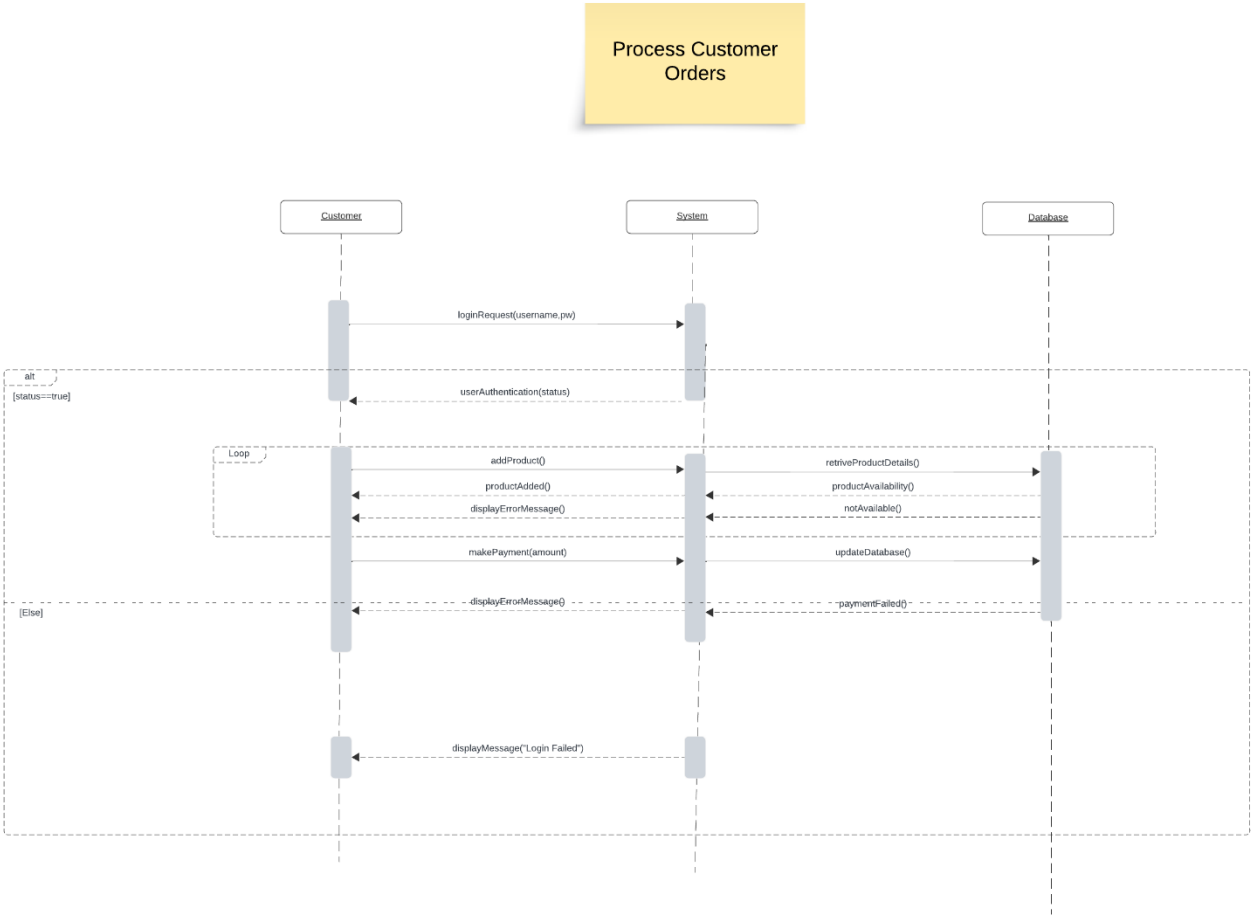
/


Handle Return and Refunds
Customer
System
Database
loginRequest(username,pw)
userAuthentication(status)
alt
[status==true]
Loop
[iterator==totalProducts]
requestReturn(product details,totalProducts)
retriveSellingDetails()
returnRefunds(amount)
returnSellingDetails(details,amount)
displayErrorMessage()
noSellingDetails()
[Else]
displayMessage("Login Failed")

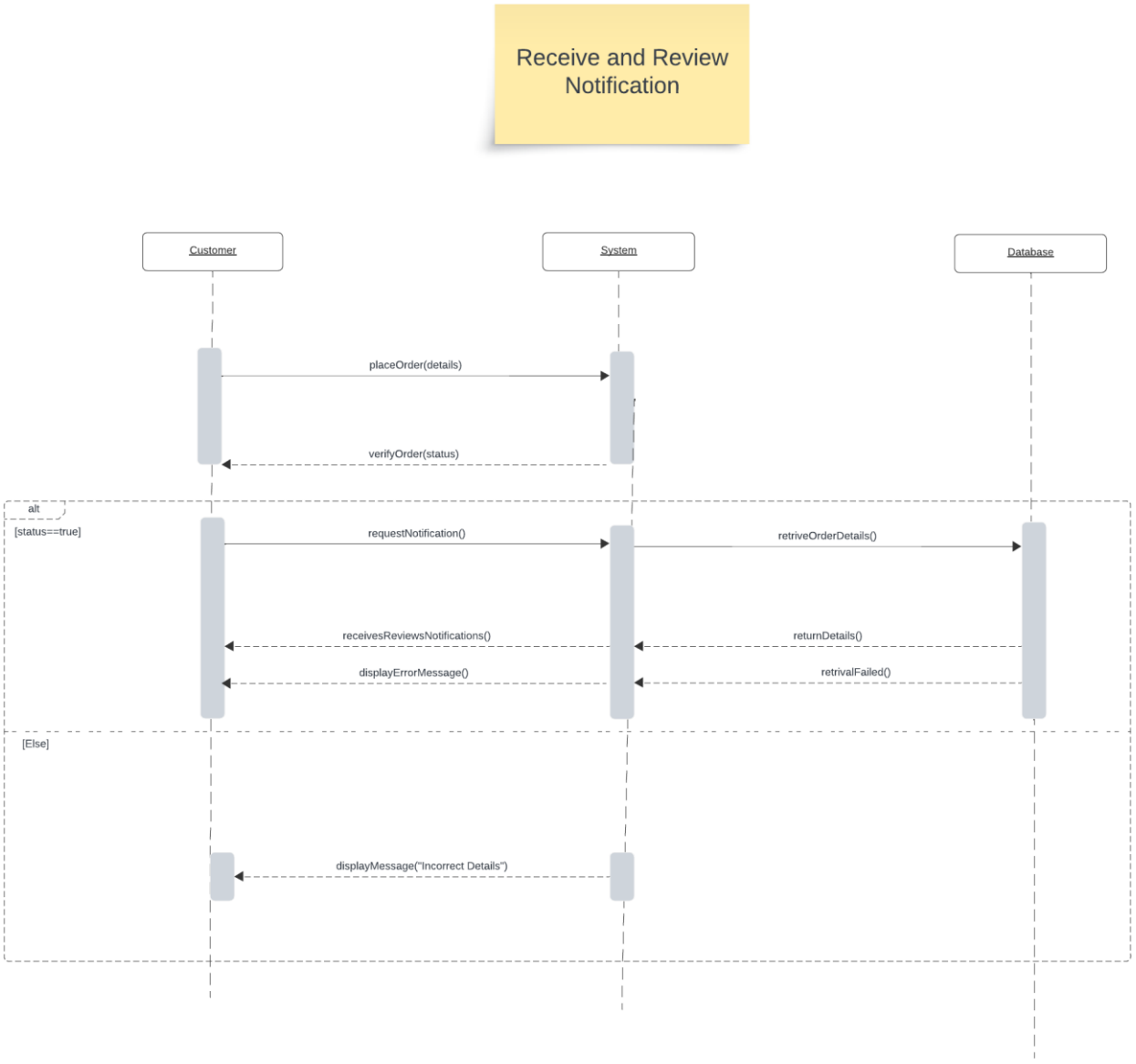
/


Inventory History Tracking
Inventory Manager
System
Database
loginRequest(username,pw)
authorizeUser(status)
alt
[status==true]
requestInventoryHistory()
requestData()
displayHistory()
reteriveData()
[Else]
displayMessage("loginfailed")

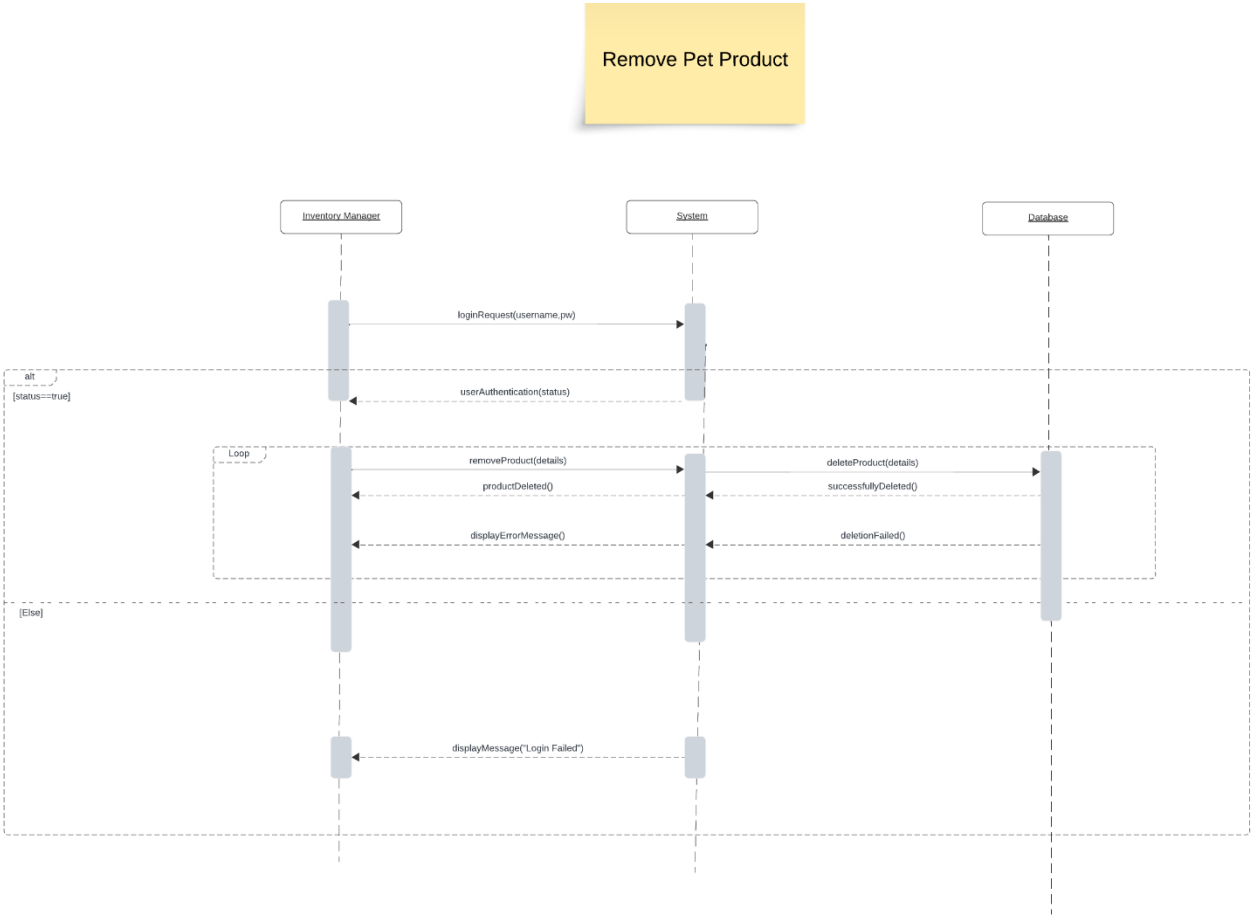
/


Process Customer Orders
Customer
System
Database
loginRequest(username,pw)
alt
[status==true]
userAuthentication(status)
Loop
addProduct()
productAdded()
displayErrorMessage()
retriveProductDetails()
productAvailability()
notAvailable()
makePayment(amount)
updateDatabase()
displayErrorMessage()
paymentFailed()
[Else]
displayMessage("Login Failed")

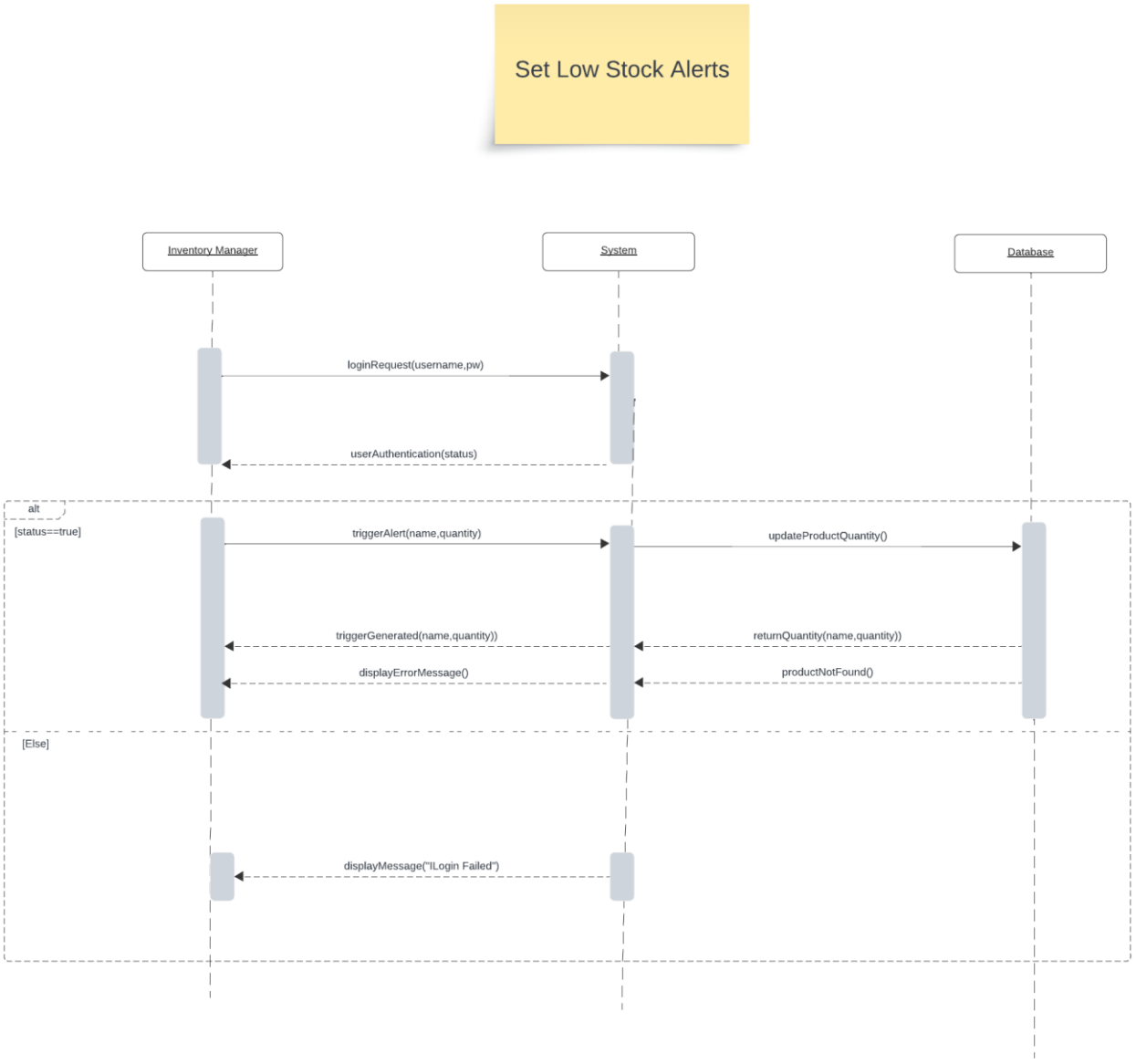
/


Receive and Review Notification
Customer
System
Database
placeOrder(details)
verifyOrder(status)
alt
[status==true]
requestNotification()
retriveOrderDetails()
receivesReviewsNotifications()
returnDetails()
displayErrorMessage()
retrivalFailed()
[Else]
displayMessage("Incorrect Details")

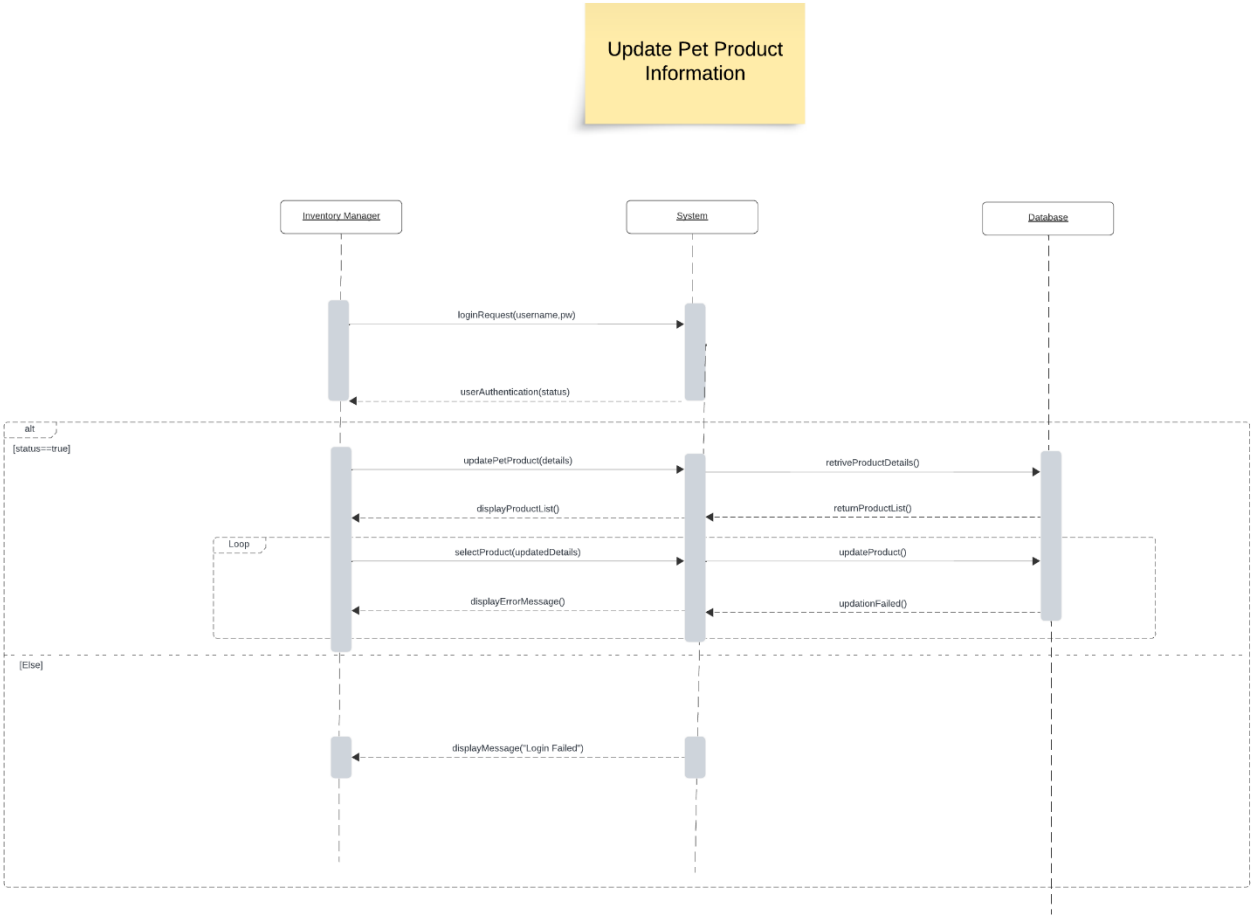
/


Remove Pet Product
Inventory Manager
System
Database
loginRequest(username,pw)
alt
[status==true]
userAuthentication(status)
Loop
removeProduct(details)
deleteProduct(details)
productDeleted()
successfullyDeleted()
displayErrorMessage()
deletionFailed()
[Else]
displayMessage("Login Failed")

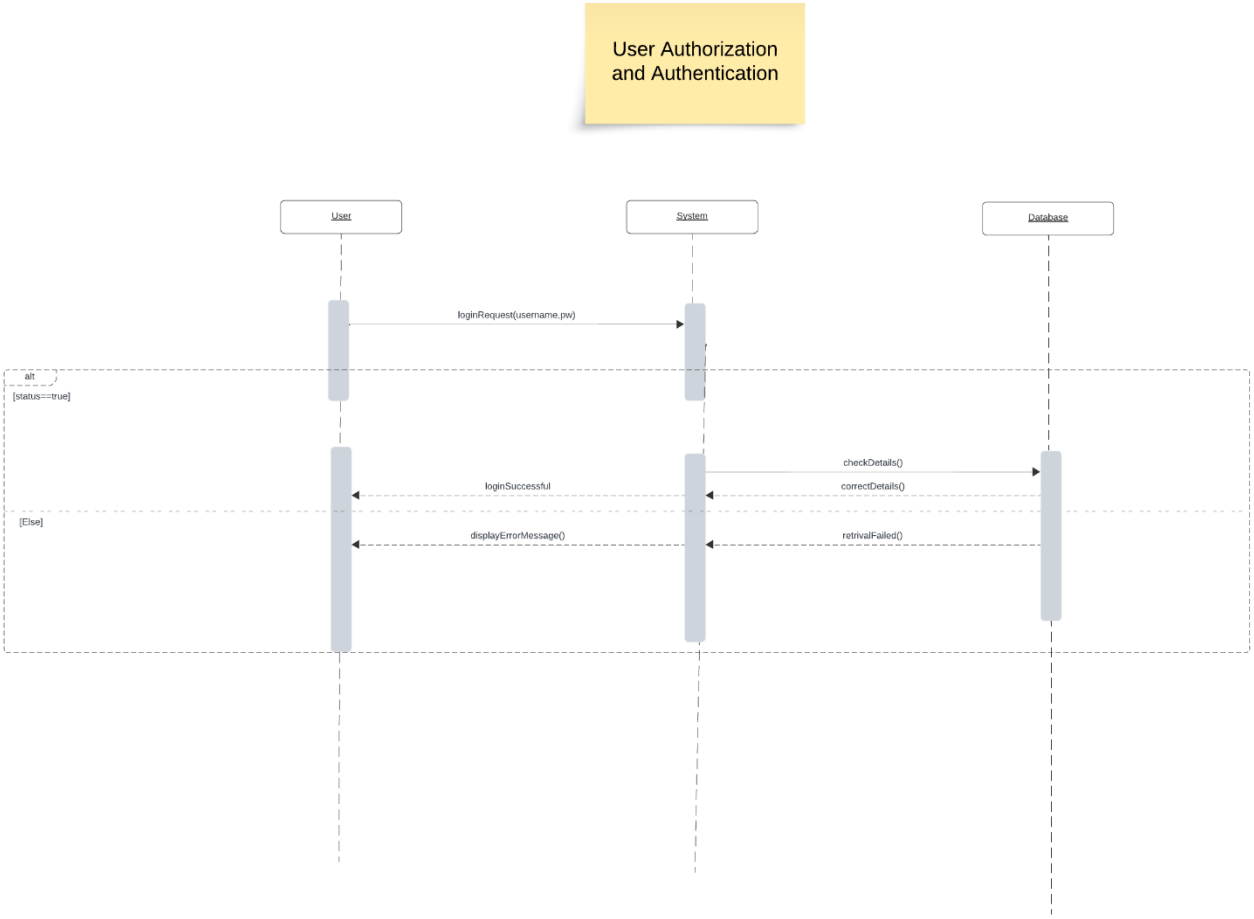
/


Set Low Stock Alerts
Inventory Manager
System
Database
loginRequest(username,pw)
userAuthentication(status)
alt
[status==true]
triggerAlert(name,quantity)
updateProductQuantity()
triggerGenerated(name,quantity))
returnQuantity(name,quantity))
displayErrorMessage()
productNotFound()
[Else]
displayMessage("ILogin Failed")

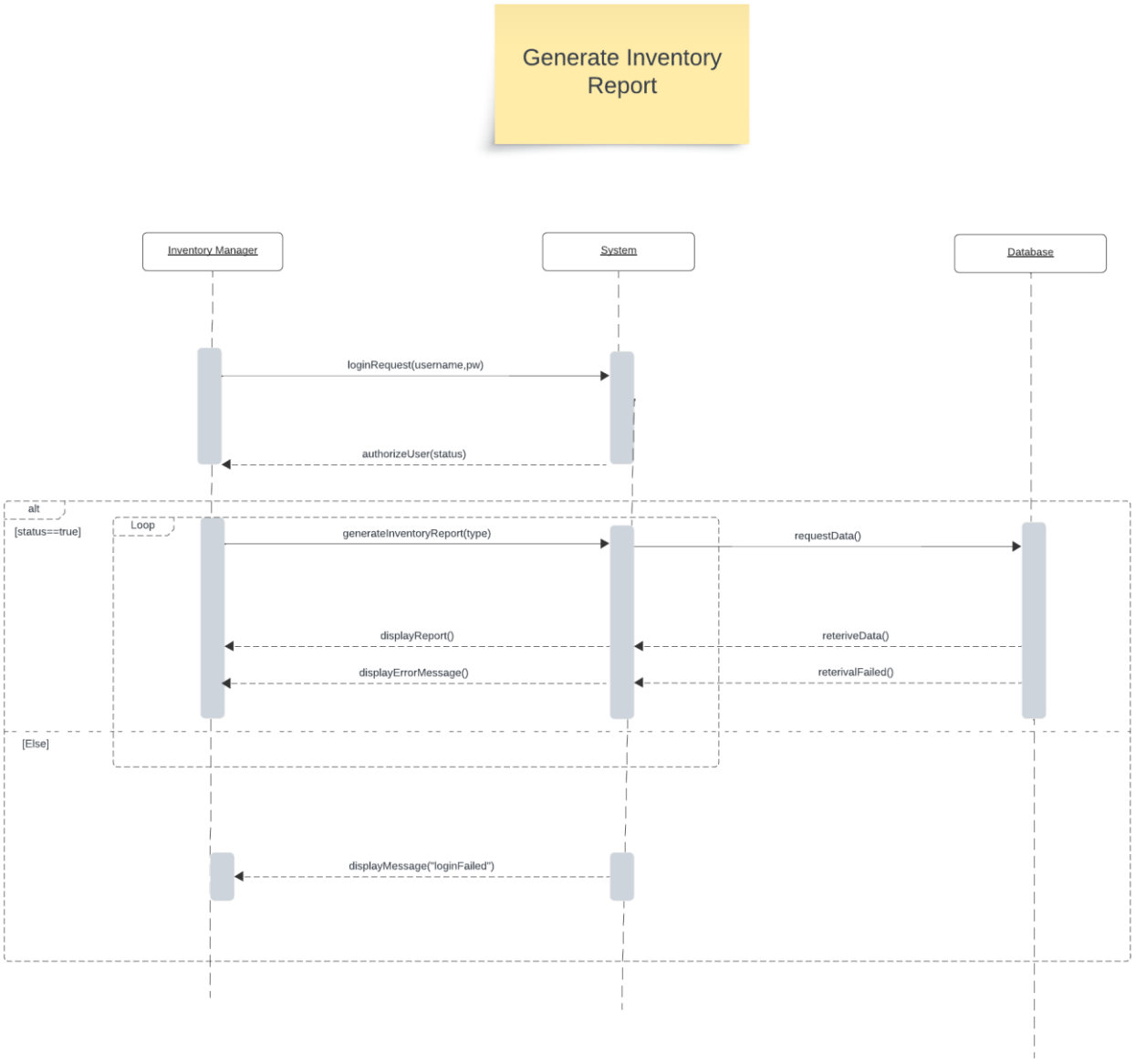
/


Update Pet Product Information
Inventory Manager
System
Database
loginRequest(username,pw)
userAuthentication(status)
alt
[status==true]
updatePetProduct(details)
retriveProductDetails()
displayProductList()
returnProductList()
Loop
selectProduct(updatedDetails)
updateProduct()
displayErrorMessage()
updationFailed()
[Else]
displayMessage("Login Failed")

/

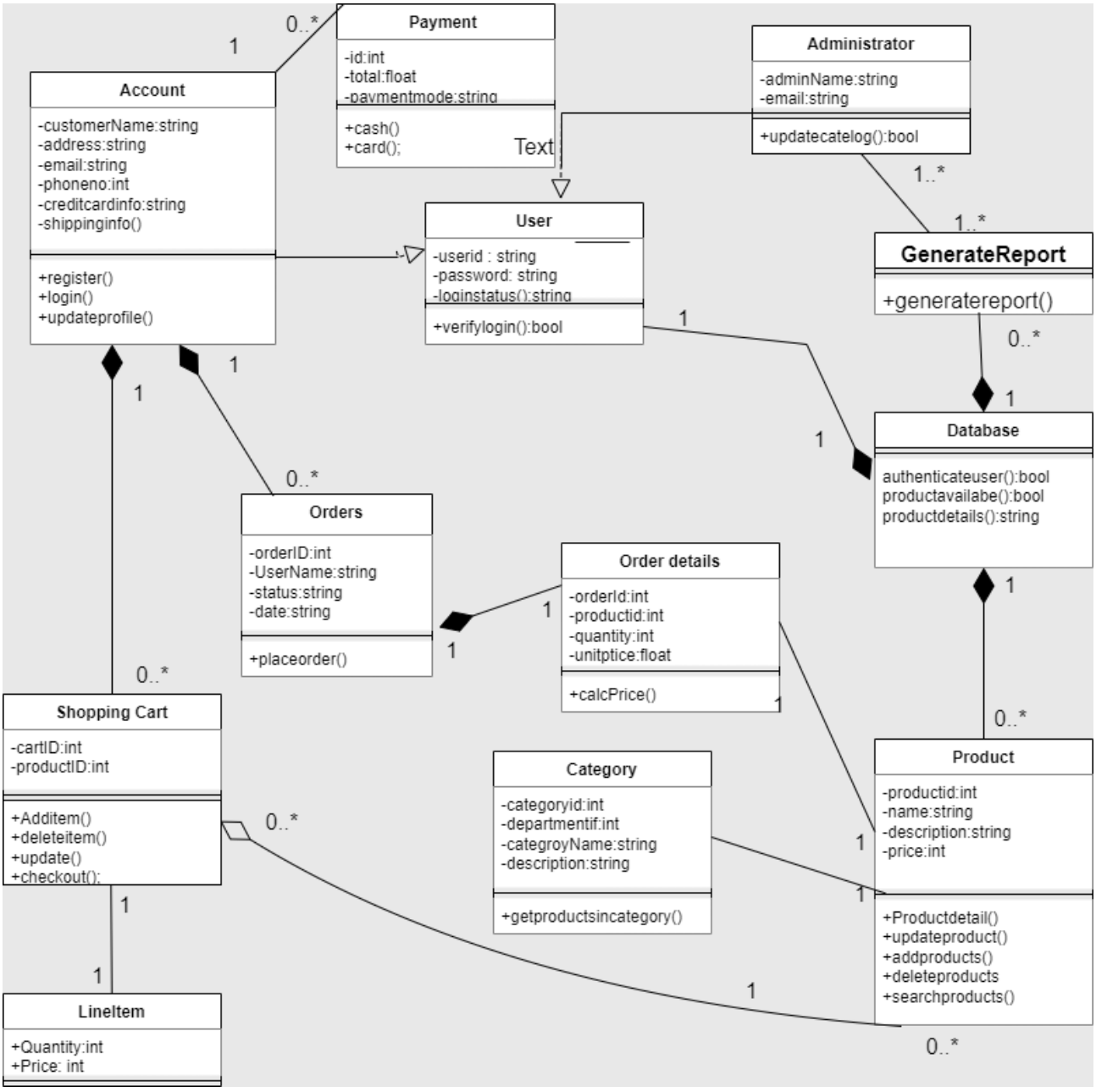

User Authorization and Authentication
User
System
Database
loginRequest(username,pw)
alt
[status==true]
checkDetails()
correctDetails()
loginSuccessful
[Else]
displayErrorMessage()
retrivalFailed()

/


Generate Inventory Report
Inventory Manager
System
Database
loginRequest(username,pw)
authorizeUser(status)
alt
[status==true]
Loop
generateInventoryReport(type)
requestData()
displayReport()
reteriveData()
displayErrorMessage()
retervalFailed()
[Else]
displayMessage("loginFailed")

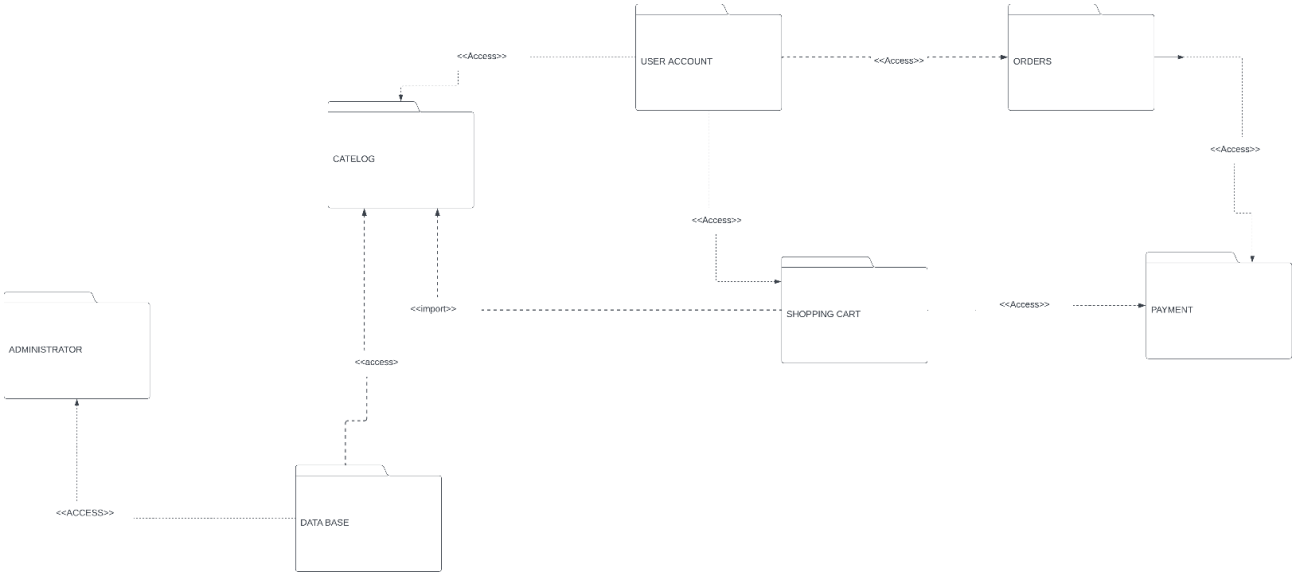
/

7.Class Diagram



/

8.Package Diagram



9.Deployment Diagram