项目文档

# Functional Requirement

### Chapter 1: Functional Requirements   
#### 1.1 Customer Registration Function   
\*\*Function ID:\*\* FR-01   
\*\*Description:\*\* Enables new customers to register by providing personal information. Validates input, checks email uniqueness, creates customer profile, and sends confirmation email.   
\*\*Input:\*\* FirstName, LastName, Email, Password, Phone, Address.   
\*\*Output:\*\* New records in Customer, Account, and Contact Information entities; confirmation email sent.   
  
#### 1.2 Customer Login Function   
\*\*Function ID:\*\* FR-02   
\*\*Description:\*\* Authenticates registered customers using email and password. Initiates session upon successful validation.   
\*\*Input:\*\* Email, Password.   
\*\*Output:\*\* Active customer session; redirect to homepage/shopping cart.   
  
#### 1.3 Customer Logout Function   
\*\*Function ID:\*\* FR-03   
\*\*Description:\*\* Terminates active customer sessions and clears session data.   
\*\*Input:\*\* Active session token.   
\*\*Output:\*\* Session terminated; redirect to login page/homepage.   
  
#### 1.4 Customer Account Management Function   
\*\*Function ID:\*\* FR-04   
\*\*Description:\*\* Allows customers or administrators to update, suspend, or delete accounts. Handles associated data cleanup.   
\*\*Input:\*\* AccountID, updated fields (e.g., Password, Email), action type (update/suspend/delete).   
\*\*Output:\*\* Modified Customer/Account entities; notifications; cleanup of linked Shopping Cart/Order/Payment entities if deleted.   
  
#### 1.5 Add to Shopping Cart Function   
\*\*Function ID:\*\* FR-05   
\*\*Description:\*\* Adds selected products to the customer’s shopping cart after inventory availability checks.   
\*\*Input:\*\* ProductID, CustomerID, Quantity (default=1).   
\*\*Output:\*\* New/modified Shopping Cart entity record; cart display updated.   
  
#### 1.6 Modify Shopping Cart Function   
\*\*Function ID:\*\* FR-06   
\*\*Description:\*\* Adjusts quantities or removes items in the shopping cart. Validates against inventory limits.   
\*\*Input:\*\* CartID, updated Quantity or removal flag.   
\*\*Output:\*\* Updated Shopping Cart entity; recalculated total price.   
  
#### 1.7 View Shopping Cart Function   
\*\*Function ID:\*\* FR-07   
\*\*Description:\*\* Displays items, quantities, prices, and total cost in the customer’s cart.   
\*\*Input:\*\* CustomerID.   
\*\*Output:\*\* Shopping Cart entity data with product details and total price.   
  
#### 1.8 Place Order Function   
\*\*Function ID:\*\* FR-08   
\*\*Description:\*\* Processes checkout by creating orders, deducting inventory, processing payments, and clearing cart items.   
\*\*Input:\*\* CustomerID, Payment Information, Shipping/Billing details.   
\*\*Output:\*\* New Order and Order Item entities; updated Inventory; cleared Shopping Cart items; confirmation email.   
  
#### 1.9 View Order History Function   
\*\*Function ID:\*\* FR-09   
\*\*Description:\*\* Lists customer’s past orders with details (e.g., status, items, prices).   
\*\*Input:\*\* CustomerID.   
\*\*Output:\*\* Order and Order Item entity data filtered by CustomerID.   
  
#### 1.10 Order Confirmation Function   
\*\*Function ID:\*\* FR-10   
\*\*Description:\*\* Automatically generates and sends order confirmation emails post-payment.   
\*\*Input:\*\* OrderID.   
\*\*Output:\*\* Confirmation email; Order entity status updated to "Confirmed".   
  
#### 1.11 Administrator Login Function   
\*\*Function ID:\*\* FR-11   
\*\*Description:\*\* Authenticates administrators using credentials and initiates admin-specific sessions.   
\*\*Input:\*\* Username, Password.   
\*\*Output:\*\* Active administrator session; redirect to admin dashboard.   
  
#### 1.12 Administrator Logout Function   
\*\*Function ID:\*\* FR-12   
\*\*Description:\*\* Terminates active administrator sessions.   
\*\*Input:\*\* Active admin session token.   
\*\*Output:\*\* Session terminated; redirect to login page/homepage.   
  
#### 1.13 Manage Products Function   
\*\*Function ID:\*\* FR-13   
\*\*Description:\*\* Allows administrators to add, edit, or delete products. Validates inputs and handles inventory associations.   
\*\*Input:\*\* Product details (ProductName, Description, Price, CategoryID), action type (add/update/delete).   
\*\*Output:\*\* Added/updated/deleted Product entity; linked Inventory updates if deleted.   
  
#### 1.14 Manage Inventory Function   
\*\*Function ID:\*\* FR-14   
\*\*Description:\*\* Updates product stock levels or adds new inventory records.   
\*\*Input:\*\* ProductID, Quantity, Location.   
\*\*Output:\*\* Added/updated Inventory entity record.   
  
#### 1.15 Manage Orders Function   
\*\*Function ID:\*\* FR-15   
\*\*Description:\*\* Updates order statuses (e.g., to "shipped" or "canceled"), triggers refunds, and adjusts inventory.   
\*\*Input:\*\* OrderID, updated Status, refund flag (if canceled).   
\*\*Output:\*\* Modified Order entity; Inventory restocked if canceled; Payment Information refund processed.   
  
#### 1.16 Manage Payment Information Function   
\*\*Function ID:\*\* FR-16   
\*\*Description:\*\* Stores, updates, or deletes customer payment methods. Validates card details.   
\*\*Input:\*\* CardNumber, ExpiryDate, CVV, Billing Address, action type (add/update/delete).   
\*\*Output:\*\* Added/updated/deleted Payment Information entity record.   
  
#### 1.17 Manage Contact Information Function   
\*\*Function ID:\*\* FR-17   
\*\*Description:\*\* Handles customer contact details (e.g., phone, address). Validates formats and uniqueness.   
\*\*Input:\*\* Phone, Address, action type (add/update/delete).   
\*\*Output:\*\* Added/updated/deleted Contact Information entity record.   
  
#### 1.18 Manage Category Function   
\*\*Function ID:\*\* FR-18   
\*\*Description:\*\* Creates, edits, or deletes product categories. Reassigns products if categories are deleted.   
\*\*Input:\*\* CategoryName, Description, action type (add/update/delete).   
\*\*Output:\*\* Added/updated/deleted Category entity; Product entity reassignments if needed.   
  
#### 1.19 Manage Administrator Function   
\*\*Function ID:\*\* FR-19   
\*\*Description:\*\* Adds, edits, or deletes administrator accounts. Validates permissions and uniqueness.   
\*\*Input:\*\* Admin credentials (Username, Password, Role), action type (add/update/delete).   
\*\*Output:\*\* Added/updated/deleted Administrator entity record.   
  
#### 1.20 Manage Order Item Function   
\*\*Function ID:\*\* FR-20   
\*\*Description:\*\* Modifies or removes items within orders (admin-only). Adjusts inventory and order totals.   
\*\*Input:\*\* OrderItemID, updated Quantity or removal flag.   
\*\*Output:\*\* Updated Order Item entity; Order total recalculated; Inventory restocked if items removed.

# External Description

### Chapter 2: External Interfaces  
  
This chapter describes the external interfaces of the system, including user interfaces, hardware interfaces, software interfaces, and communication interfaces. These interfaces are derived from the functional requirements and ensure seamless interaction between the system and external entities.  
  
---  
  
### 2.1 User Interface  
  
#### 2.1.1 Customer Registration Interface   
\*\*Description:\*\*   
A user interface that allows new customers to input personal information (FirstName, LastName, Email, Password, Phone, Address) and submit the registration form.   
\*\*Interaction Method:\*\*   
- Input: Users enter data into form fields.   
- Output: Confirmation of successful registration and a redirect to the login page.   
  
#### 2.1.2 Customer Login Interface   
\*\*Description:\*\*   
A user interface that enables customers to enter their email and password to authenticate.   
\*\*Interaction Method:\*\*   
- Input: Users submit their email and password.   
- Output: Successful login results in an active session and redirection to the homepage or shopping cart.   
  
#### 2.1.3 Customer Logout Interface   
\*\*Description:\*\*   
A user interface that terminates the active customer session.   
\*\*Interaction Method:\*\*   
- Input: Session token.   
- Output: Termination of the session and redirection to the login page or homepage.   
  
#### 2.1.4 Customer Account Management Interface   
\*\*Description:\*\*   
A user interface that allows customers to update their account details, such as password, email, phone, or address.   
\*\*Interaction Method:\*\*   
- Input: Updated fields (e.g., Password, Email).   
- Output: Confirmation of the update and reflection of changes in the user profile.   
  
#### 2.1.5 Shopping Cart Interface   
\*\*Description:\*\*   
A user interface that displays the contents of the customer’s shopping cart, including item details, quantities, and total price.   
\*\*Interaction Method:\*\*   
- Input: Customer adds or modifies items in the cart.   
- Output: Updated cart display with recalculated totals.   
  
#### 2.1.6 Order Placement Interface   
\*\*Description:\*\*   
A user interface that allows customers to review their order, enter payment and shipping details, and confirm the order.   
\*\*Interaction Method:\*\*   
- Input: Customer provides payment information, shipping, and billing details.   
- Output: Order confirmation, cart clearing, and redirection to a confirmation page.   
  
#### 2.1.7 Administrator Login Interface   
\*\*Description:\*\*   
A user interface that authenticates administrators using their credentials (Username, Password).   
\*\*Interaction Method:\*\*   
- Input: Administrator enters their username and password.   
- Output: Successful login results in an active session and redirection to the admin dashboard.   
  
#### 2.1.8 Administrator Logout Interface   
\*\*Description:\*\*   
A user interface that terminates the active administrator session.   
\*\*Interaction Method:\*\*   
- Input: Session token.   
- Output: Termination of the session and redirection to the login page.   
  
---  
  
### 2.2 Hardware Interface  
  
\*\*Description:\*\*   
The system does not interact directly with external hardware devices beyond standard computer peripherals (e.g., keyboard, mouse, monitor). All hardware interactions are mediated through the operating system and standard input/output mechanisms.  
  
---  
  
### 2.3 Software Interface  
  
#### 2.3.1 Customer and Account Database   
\*\*Description:\*\*   
A database interface responsible for storing and managing customer and account-related data.   
\*\*Interaction Method:\*\*   
- Inputs: Customer registration details (FirstName, LastName, Email, Password, Phone, Address), account updates, and deletions.   
- Outputs: Creation, modification, or deletion of records in the Customer, Account, and Contact Information entities.   
  
#### 2.3.2 Product and Inventory Database   
\*\*Description:\*\*   
A database interface responsible for storing and managing product and inventory-related data.   
\*\*Interaction Method:\*\*   
- Inputs: Product details (ProductName, Description, Price, CategoryID), inventory updates (ProductID, Quantity, Location).   
- Outputs: Addition, modification, or deletion of records in the Product and Inventory entities.   
  
#### 2.3.3 Order and Payment Database   
\*\*Description:\*\*   
A database interface responsible for storing and managing order and payment-related data.   
\*\*Interaction Method:\*\*   
- Inputs: Order details (CustomerID, Payment Information, Shipping/Billing details), order status updates (OrderID, updated Status).   
- Outputs: Creation, modification, or deletion of records in the Order, Order Item, and Payment Information entities.   
  
#### 2.3.4 Email Service API   
\*\*Description:\*\*   
An external API interface used to send confirmation emails (e.g., registration, order confirmation).   
\*\*Interaction Method:\*\*   
- Inputs: Recipient email address, email content, and subject.   
- Outputs: Confirmation of successful email delivery.   
  
#### 2.3.5 Payment Processing API   
\*\*Description:\*\*   
An external API interface used to process customer payments and refunds.   
\*\*Interaction Method:\*\*   
- Inputs: Payment details (CardNumber, ExpiryDate, CVV, Billing Address), refund requests.   
- Outputs: Payment confirmation or refund processing status.   
  
---  
  
### 2.4 Communication Interface  
  
#### 2.4.1 Email Notifications   
\*\*Description:\*\*   
A communication interface used to send automated emails (e.g., registration confirmation, order confirmation, password reset requests).   
\*\*Interaction Method:\*\*   
- Inputs: Recipient email address, email content, and subject.   
- Outputs: Email sent to the recipient.   
  
#### 2.4.2 Order Confirmation Email   
\*\*Description:\*\*   
A communication interface that sends an email to the customer after a successful order placement.   
\*\*Interaction Method:\*\*   
- Inputs: OrderID, customer email address.   
- Outputs: Confirmation email sent to the customer.   
  
#### 2.4.3 Refund Notifications   
\*\*Description:\*\*   
A communication interface used to notify customers of refunds processed for canceled orders.   
\*\*Interaction Method:\*\*   
- Inputs: Customer email address, refund details.   
- Outputs: Refund notification email sent to the customer.   
  
---  
  
This chapter ensures that all external interfaces are clearly defined and aligned with the functional requirements. Developers can use this information to design and implement the system’s interaction with external entities effectively.

# Use Case

Use Case Name: Customer Registration   
Use Case ID: UC-01   
Actors: Customer, Administrator   
Preconditions:   
1. The system is operational and accessible.   
2. The customer is not yet registered in the system.   
3. The administrator has the necessary permissions to review or approve customer registration requests.   
  
Postconditions:   
1. A new customer account is successfully created in the system.   
2. The customer receives a confirmation message or email.   
3. The customer's information is stored securely in the database.   
  
Main Flow:   
1. The customer navigates to the registration page on the system.   
2. The customer fills out the registration form with personal information (e.g., name, email, password, address).   
3. The system validates the customer's input (e.g., checks for valid email format, password strength).   
4. The system checks if the email is already registered.   
5. If the email is not registered, the system creates a new customer profile and stores it in the Customer data entity.   
6. The system sends a confirmation message to the customer's email.   
7. The customer confirms their email by clicking the confirmation link.   
8. The system updates the customer's status to "Active" in the database.   
9. The customer is redirected to the homepage or shopping cart.   
  
Alternative Flow:   
1. If the system detects that the email is already registered, it displays an error message to the customer.   
2. If the customer fails to confirm their email within a specified time frame, the system marks the registration as "Expired" and requires the customer to re-initiate the process.   
3. If the customer provides invalid or incomplete information during registration, the system prompts them to correct the data before proceeding.   
4. If the system fails to send the confirmation email, the customer is informed of the failure, and the administrator is notified to investigate and resolve the issue.  
  
Use Case Name: Customer Login   
Use Case ID: UC-02   
Actors: Customer, System   
  
Preconditions:   
1. The system is operational and accessible.   
2. The customer has already registered and has a valid account.   
3. The customer has a valid email and password stored in the Customer data entity.   
  
Postconditions:   
1. The customer is successfully logged into the system.   
2. The customer's session is initiated.   
3. The customer is redirected to their homepage or shopping cart.   
  
Main Flow:   
1. The customer navigates to the login page on the system.   
2. The customer enters their registered email and password into the login form.   
3. The system validates the input format (e.g., email format, password length).   
4. The system checks the Customer data entity to verify the email and password match.   
5. If the credentials are valid, the system initiates a session for the customer.   
6. The system redirects the customer to their homepage or shopping cart.   
  
Alternative Flow:   
1. If the email or password is invalid, the system displays an error message prompting the customer to re-enter the correct information.   
2. If the customer enters incorrect credentials multiple times, the system may lock the account temporarily and notify the customer via email.   
3. If the customer's account is not active (e.g., unconfirmed email), the system prompts the customer to verify their email before logging in.   
4. If the system detects a technical issue during the login process, it displays an error message and logs the issue for further investigation.  
  
Use Case Name: Customer Logout   
Use Case ID: UC-03   
Actors: Customer, System   
  
Preconditions:   
1. The system is operational and accessible.   
2. The customer is currently logged in and has an active session.   
  
Postconditions:   
1. The customer's session is terminated.   
2. The customer is redirected to the login or homepage.   
3. The system no longer recognizes the customer as logged in.   
  
Main Flow:   
1. The customer navigates to the account or profile section on the homepage.   
2. The customer selects the "Logout" option from the available menu.   
3. The system verifies the customer's session and initiates the logout process.   
4. The system clears the session data and cookies associated with the customer.   
5. The system redirects the customer to the login page or homepage.   
6. The system displays a confirmation message that the customer has been successfully logged out.   
  
Alternative Flow:   
1. If the system fails to clear session data, the customer remains logged in and the system displays an error message.   
2. If the customer attempts to access restricted pages after logging out, the system prompts them to log in again.   
3. If the customer closes the browser without explicitly logging out, the system automatically logs them out after a period of inactivity.  
  
Use Case Name: Customer Account Management   
Use Case ID: UC-04   
Actors: Customer, Administrator   
Preconditions:   
1. The system is operational and accessible.   
2. The customer is already registered and has a valid account.   
3. The administrator has the necessary permissions to manage customer accounts.   
  
Postconditions:   
1. The customer account is updated, suspended, or deleted based on the action performed.   
2. The system logs the change made to the customer account.   
3. The customer is notified of any changes made to their account status.   
  
Main Flow:   
1. The customer or administrator accesses the account management section.   
2. The customer or administrator selects an account management action (e.g., update personal information, change password, delete account, suspend account).   
3. The system verifies the user's identity and permissions to perform the action.   
4. The system displays the current customer account information.   
5. The user modifies the relevant information or confirms the action.   
6. The system validates the input and updates the Customer data entity accordingly.   
7. The system logs the action and sends a confirmation or notification to the customer.   
  
Alternative Flow:   
1. If the user does not have sufficient permissions, the system displays an error and denies the action.   
2. If the user attempts to delete or suspend their own account, the system requires additional confirmation to proceed.   
3. If the user enters invalid or incomplete information during an update, the system prompts them to correct the data.   
4. If the system fails to update the account due to a technical error, it displays an error message and logs the issue for investigation.   
5. If the customer account is deleted, the system removes all associated data from the Shopping Cart, Order, and Payment Information data entities.  
  
Use Case Name: Add to Shopping Cart   
Use Case ID: UC-05   
Actors: Customer, System   
  
Preconditions:   
1. The system is operational and accessible.   
2. The customer is logged in and has an active session.   
3. The customer has browsed and selected a product from the Product data entity.   
4. The Shopping Cart data entity exists and is associated with the customer.   
  
Postconditions:   
1. The selected product is added to the customer's Shopping Cart.   
2. The quantity and product details are updated in the Shopping Cart.   
3. The customer receives a confirmation message that the product was successfully added.   
  
Main Flow:   
1. The customer browses the product catalog and selects a product.   
2. The customer clicks the "Add to Cart" button on the product page.   
3. The system checks the product's availability in the Inventory data entity.   
4. The system adds the product to the Shopping Cart data entity with the default quantity of 1.   
5. The system updates the Shopping Cart display to reflect the new item.   
6. The system displays a confirmation message to the customer.   
  
Alternative Flow:   
1. If the product is out of stock, the system displays a message indicating the product is unavailable.   
2. If the customer tries to add more items than available in the Inventory, the system shows an error and suggests the maximum quantity.   
3. If the customer is not logged in, the system prompts them to log in or register before adding to the cart.   
4. If the system fails to update the Shopping Cart due to a technical error, it displays an error message and logs the issue for investigation.  
  
Use Case Name: Modify Shopping Cart   
Use Case ID: UC-06   
Actors: Customer, System   
  
Preconditions:   
1. The system is operational and accessible.   
2. The customer is logged in and has an active session.   
3. The customer has at least one item in their Shopping Cart.   
4. The Shopping Cart data entity is associated with the customer.   
  
Postconditions:   
1. The Shopping Cart is updated with the modified items (e.g., quantity changes, item removals).   
2. The system displays the updated Shopping Cart.   
3. The customer receives a confirmation message of the changes.   
  
Main Flow:   
1. The customer navigates to the Shopping Cart page.   
2. The system displays the current items in the customer's Shopping Cart.   
3. The customer selects an item and modifies its quantity or removes it.   
4. The system validates the changes (e.g., quantity not exceeding available inventory).   
5. The system updates the Shopping Cart data entity with the new information.   
6. The system recalculates the total price and updates the cart display.   
7. The system displays a confirmation message to the customer.   
  
Alternative Flow:   
1. If the customer tries to set a quantity higher than the available stock, the system displays an error message and suggests the maximum available quantity.   
2. If the customer attempts to modify the cart without being logged in, the system prompts them to log in or register.   
3. If the system fails to update the Shopping Cart due to a technical error, it displays an error message and logs the issue for investigation.   
4. If the customer removes all items from the cart, the system displays a message indicating the cart is empty and may suggest browsing products again.  
  
Use Case Name: View Shopping Cart   
Use Case ID: UC-07   
Actors: Customer, System   
  
Preconditions:   
1. The system is operational and accessible.   
2. The customer is logged in and has an active session.   
3. The customer has previously added items to their Shopping Cart.   
4. The Shopping Cart data entity exists and is associated with the customer.   
  
Postconditions:   
1. The customer can view the items currently in their Shopping Cart.   
2. The system displays the total price and quantity of items in the cart.   
3. The customer is informed if there are any changes to product availability or pricing.   
  
Main Flow:   
1. The customer navigates to the Shopping Cart page from the homepage or product page.   
2. The system retrieves the customer's Shopping Cart data from the Shopping Cart data entity.   
3. The system displays the list of items in the cart, including product details, quantity, and price.   
4. The system calculates and displays the total price of all items in the cart.   
5. The customer can review the cart and choose to proceed to checkout or continue shopping.   
  
Alternative Flow:   
1. If the customer's Shopping Cart is empty, the system displays a message indicating the cart is empty and suggests browsing products.   
2. If the system detects that an item in the cart is no longer available in the Inventory, it updates the cart and displays a warning message.   
3. If the system fails to retrieve Shopping Cart data due to a technical issue, it displays an error message and logs the issue for investigation.   
4. If the customer is not logged in, the system prompts them to log in or register before viewing the cart.  
  
Use Case Name: Place Order   
Use Case ID: UC-08   
Actors: Customer, System, Administrator   
  
Preconditions:   
1. The system is operational and accessible.   
2. The customer is logged in and has an active session.   
3. The Shopping Cart contains at least one item.   
4. The customer has selected a valid payment method and provided necessary Payment Information.   
5. The Inventory has sufficient stock for the items in the Shopping Cart.   
  
Postconditions:   
1. A new Order is created and stored in the Order data entity.   
2. The Shopping Cart is cleared of the items included in the order.   
3. The Payment Information is processed and recorded in the Payment Information data entity.   
4. The Inventory quantities are updated to reflect the deduction of ordered items.   
5. The customer receives a confirmation message or email for the order.   
  
Main Flow:   
1. The customer navigates to the checkout page from the Shopping Cart.   
2. The system displays the items in the cart, total price, and requires shipping and billing information.   
3. The customer provides or confirms their shipping and billing details.   
4. The customer selects a preferred payment method and completes the Payment Information form.   
5. The system validates the payment details and processes the payment.   
6. The system creates a new Order in the Order data entity, including items, total price, and customer information.   
7. The system updates the Inventory data entity by reducing the stock of each ordered product.   
8. The system clears the Shopping Cart of the ordered items.   
9. The system sends a confirmation message to the customer and logs the order details.   
  
Alternative Flow:   
1. If the customer's provided payment information is invalid, the system displays an error and prompts for correction.   
2. If the system is unable to process the payment, it displays an error and logs the issue for the Administrator to review.   
3. If the Inventory does not have sufficient stock for an item, the system updates the cart and displays an error message.   
4. If the customer cancels the order during checkout, the system returns them to the Shopping Cart page.   
5. If the system fails to create the order due to a technical error, it displays an error message and logs the issue for investigation.  
  
Use Case Name: View Order History   
Use Case ID: UC-09   
Actors: Customer, System   
  
Preconditions:   
1. The system is operational and accessible.   
2. The customer is logged in and has an active session.   
3. The customer has previously placed at least one order.   
4. The Order data entity contains the customer's order records.   
  
Postconditions:   
1. The customer can view a list of their previous orders, including order details and status.   
2. The system displays the order history in a clear and organized manner.   
3. The system logs the access of order history for security and audit purposes.   
  
Main Flow:   
1. The customer navigates to the "Order History" section in their account.   
2. The system retrieves the customer's order records from the Order data entity.   
3. The system displays the order history, including order ID, date, total price, and status for each order.   
4. The customer can click on an order to view detailed information, such as products purchased, shipping address, and payment method.   
5. The system provides the ability to download or print the order details if needed.   
  
Alternative Flow:   
1. If the customer has no previous orders, the system displays a message indicating the order history is empty.   
2. If the system fails to retrieve order data, it displays an error message and logs the issue for investigation.   
3. If the customer is not logged in, the system prompts them to log in or register before viewing order history.   
4. If the customer requests additional order details not available, the system may prompt the customer to contact customer support for further assistance.  
  
Use Case Name: Order Confirmation   
Use Case ID: UC-10   
Actors: Customer, System   
  
Preconditions:   
1. The system is operational and accessible.   
2. The customer has successfully placed an order and completed the payment process.   
3. The Order data entity contains the order details.   
4. The Payment Information has been verified as valid.   
  
Postconditions:   
1. The customer receives a confirmation message or email for their order.   
2. The order status is updated to "Confirmed" in the Order data entity.   
3. The system logs the order confirmation event.   
  
Main Flow:   
1. After completing the payment, the system automatically triggers the order confirmation process.   
2. The system generates a confirmation message containing the order details, such as order ID, items purchased, total price, and estimated delivery time.   
3. The system sends the confirmation message via email to the customer.   
4. The system updates the order status in the Order data entity to "Confirmed."   
5. The customer receives the confirmation and can view it in their account or through the provided email.   
  
Alternative Flow:   
1. If the system fails to send the confirmation email, the customer is shown a message that the confirmation is being processed, and the system logs the failure for the Administrator to review.   
2. If the customer does not receive the confirmation email within a reasonable timeframe, they can manually request it through their account or contact customer support.   
3. If the system detects an issue with the order (e.g., inventory inconsistency), it may delay confirmation and notify the customer via email while the issue is resolved.   
4. If the customer is not logged in when the order is placed, the confirmation message will still be sent, but the order will be linked to the customer’s account once they log in.  
  
Use Case Name: Administrator Login   
Use Case ID: UC-11   
Actors: Administrator, System   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator has an existing account in the system.   
3. The administrator has not yet logged in.   
  
Postconditions:   
1. The administrator is successfully logged into the system.   
2. The administrator's session is initiated.   
3. The administrator is redirected to the administrator dashboard.   
  
Main Flow:   
1. The administrator navigates to the login page for the admin section of the system.   
2. The administrator enters their registered username and password into the login form.   
3. The system validates the input format (e.g., username format, password length).   
4. The system checks the Administrator data entity to verify the username and password match.   
5. If the credentials are valid, the system initiates a session for the administrator.   
6. The system redirects the administrator to the admin dashboard.   
7. The system logs the login event for security and audit purposes.   
  
Alternative Flow:   
1. If the username or password is invalid, the system displays an error message prompting the administrator to re-enter the correct information.   
2. If the administrator enters incorrect credentials multiple times, the system may lock the account temporarily and notify the administrator via email.   
3. If the system detects a technical issue during the login process, it displays an error message and logs the issue for further investigation.  
  
Use Case Name: Administrator Logout   
Use Case ID: UC-12   
Actors: Administrator, System   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is currently logged in and has an active session.   
  
Postconditions:   
1. The administrator's session is terminated.   
2. The administrator is redirected to the login page or homepage.   
3. The system no longer recognizes the administrator as logged in.   
  
Main Flow:   
1. The administrator navigates to the account or profile section on the admin dashboard.   
2. The administrator selects the "Logout" option from the available menu.   
3. The system verifies the administrator's session and initiates the logout process.   
4. The system clears the session data and cookies associated with the administrator.   
5. The system redirects the administrator to the login page or homepage.   
6. The system displays a confirmation message that the administrator has been successfully logged out.   
  
Alternative Flow:   
1. If the system fails to clear session data, the administrator remains logged in and the system displays an error message.   
2. If the administrator attempts to access restricted admin pages after logging out, the system prompts them to log in again.   
3. If the administrator closes the browser without explicitly logging out, the system automatically logs them out after a period of inactivity.  
  
Use Case Name: Manage Inventory   
Use Case ID: UC-13   
Actors: Administrator, System   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged in and has an active session.   
3. The administrator has the necessary permissions to manage inventory.   
4. The Inventory data entity contains product stock information.   
  
Postconditions:   
1. The Inventory data entity is updated with new or modified stock levels.   
2. The system logs the inventory changes for audit purposes.   
3. The administrator receives a confirmation message of the changes.   
  
Main Flow:   
1. The administrator navigates to the Inventory Management section from the admin dashboard.   
2. The system displays the current inventory levels for all products.   
3. The administrator selects a product to modify its stock level.   
4. The administrator updates the product's stock quantity or adds a new product to the inventory.   
5. The system validates the input data (e.g., numeric quantity, product details).   
6. The system updates the Inventory data entity with the new or modified information.   
7. The system logs the change and displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the administrator enters invalid or non-numeric data for stock quantity, the system displays an error and prompts for correction.   
2. If the system fails to update the inventory due to a technical error, it displays an error message and logs the issue for investigation.   
3. If the administrator attempts to modify a product that does not exist, the system displays an error message and suggests verifying the product ID.   
4. If the administrator tries to set a stock level below zero, the system displays a warning and prevents the update.  
  
Use Case Name: Manage Products   
Use Case ID: UC-14   
Actors: Administrator, System   
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged in and has an active session.   
3. The administrator has the necessary permissions to manage products.   
4. The Product data entity exists and contains the product catalog.   
  
Postconditions:   
1. The Product data entity is updated with new, modified, or removed products.   
2. The system logs the product management actions for audit purposes.   
3. The administrator receives a confirmation message of the changes.   
  
Main Flow:   
1. The administrator navigates to the Product Management section from the admin dashboard.   
2. The system displays a list of existing products, including details like product ID, name, price, description, and stock.   
3. The administrator selects an action (e.g., add new product, edit existing product, delete product).   
4. For adding or editing, the administrator fills in or updates product details (e.g., name, price, description, category, stock quantity).   
5. The system validates the input (e.g., numeric price, valid description length).   
6. The system updates the Product data entity with the new or modified product information.   
7. The system logs the action and displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the administrator enters invalid or incomplete product details, the system displays an error message and prompts for correction.   
2. If the system fails to update the Product data entity due to a technical error, it displays an error message and logs the issue for investigation.   
3. If the administrator attempts to delete a product that is currently in an active order, the system displays a warning and prevents the deletion.   
4. If the administrator tries to add a product with a duplicate product ID, the system displays an error and suggests using a unique ID.   
5. If the administrator cancels the action before submitting, the system returns to the previous screen without making any changes.  
  
Use Case Name: Product Addition   
Use Case ID: UC-15   
Actors: Administrator, System   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged in and has an active session.   
3. The administrator has the necessary permissions to add new products.   
4. The Product data entity is ready to receive new entries.   
  
Postconditions:   
1. A new product is successfully added to the Product data entity.   
2. The system logs the product addition for audit purposes.   
3. The product becomes available for customers to view and add to their shopping cart.   
  
Main Flow:   
1. The administrator navigates to the Product Addition section from the admin dashboard.   
2. The system displays a form for entering new product details.   
3. The administrator fills in the product information (e.g., product ID, name, price, description, category, stock quantity).   
4. The system validates the input (e.g., numeric price, valid product ID format).   
5. The system saves the new product information to the Product data entity.   
6. The system logs the addition and displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the administrator enters invalid or incomplete product details, the system displays an error message and prompts for correction.   
2. If the system fails to save the product due to a technical error, it displays an error message and logs the issue for investigation.   
3. If the administrator tries to add a product with a duplicate product ID, the system displays an error and suggests using a unique ID.   
4. If the administrator cancels the action before submitting, the system returns to the previous screen without making any changes.  
  
Use Case Name: Product Update   
Use Case ID: UC-16   
Actors: Administrator, System, Product, Inventory   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged in and has an active session.   
3. The administrator has the necessary permissions to update product information.   
4. The Product data entity contains the product to be updated.   
5. The Inventory data entity reflects the current stock levels of the product.   
  
Postconditions:   
1. The Product data entity is updated with the new or modified product information.   
2. The Inventory data entity is updated if the stock quantity is modified.   
3. The system logs the product update action for audit purposes.   
4. The administrator receives a confirmation message that the update was successful.   
  
Main Flow:   
1. The administrator navigates to the Product Management section from the admin dashboard.   
2. The system displays a list of existing products, including details like product ID, name, price, description, and stock quantity.   
3. The administrator selects a product to update.   
4. The system displays the current product information in an editable form.   
5. The administrator modifies the product details (e.g., price, description, stock quantity).   
6. The system validates the input (e.g., numeric price, valid description length, stock quantity not below zero).   
7. The system updates the Product data entity with the new information.   
8. If the stock quantity is changed, the system also updates the Inventory data entity.   
9. The system logs the update and displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the administrator enters invalid or incomplete product details, the system displays an error message and prompts for correction.   
2. If the system fails to update the Product or Inventory data entity due to a technical error, it displays an error message and logs the issue for investigation.   
3. If the administrator tries to set a stock level below zero, the system displays a warning and prevents the update.   
4. If the administrator cancels the update action before submitting, the system returns to the previous screen without making any changes.  
  
Use Case Name: Product Deletion   
Use Case ID: UC-17   
Actors: Administrator, System, Product, Inventory   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged in and has an active session.   
3. The administrator has the necessary permissions to delete products.   
4. The Product data entity contains the product to be deleted.   
5. The Inventory data entity contains the stock information for the product.   
  
Postconditions:   
1. The specified product is removed from the Product data entity.   
2. The Inventory data entity is updated to remove the product's stock information.   
3. The system logs the product deletion for audit purposes.   
4. The administrator receives a confirmation message that the deletion was successful.   
  
Main Flow:   
1. The administrator navigates to the Product Management section from the admin dashboard.   
2. The system displays a list of existing products, including product ID, name, price, and stock.   
3. The administrator selects a product to delete.   
4. The system prompts the administrator to confirm the deletion.   
5. The administrator confirms the deletion.   
6. The system checks if the product is associated with any active orders.   
7. If the product is not in any active orders, the system removes the product from the Product data entity.   
8. The system also deletes the corresponding inventory record from the Inventory data entity.   
9. The system logs the deletion and displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the product is part of an active order, the system displays a warning and prevents the deletion.   
2. If the administrator cancels the deletion after the confirmation prompt, the system returns to the product list without making any changes.   
3. If the system fails to delete the product due to a technical error, it displays an error message and logs the issue for investigation.   
4. If the administrator selects a product that does not exist, the system displays an error and suggests verifying the product ID.  
  
Use Case Name: Manage Orders   
Use Case ID: UC-18   
Actors: Customer, Administrator, System, Order, Inventory, Payment Information   
  
Preconditions:   
1. The system is operational and accessible.   
2. The customer or administrator is logged in and has an active session.   
3. The Order data entity contains existing order records.   
4. The Inventory and Payment Information data entities are accessible for order-related updates.   
  
Postconditions:   
1. The order status is updated in the Order data entity (e.g., confirmed, shipped, canceled).   
2. The Inventory is updated if the order is canceled or modified.   
3. The Payment Information is updated if a refund is processed.   
4. The system logs the order management actions for audit purposes.   
5. The customer and administrator are notified of the changes.   
  
Main Flow:   
1. The customer or administrator navigates to the "Manage Orders" section in their respective account.   
2. The system displays a list of orders, including order ID, date, total price, and current status.   
3. The user selects an order to view or modify.   
4. The system retrieves and displays the order details (e.g., items, payment method, shipping address).   
5. The user chooses an action (e.g., cancel order, update status to shipped, modify order items).   
6. The system validates the action based on the current status of the order and user permissions.   
7. If the action is valid, the system updates the Order data entity with the new status or modifications.   
8. If the order is canceled, the system refunds the Payment Information and restores the product stock in the Inventory.   
9. The system logs the action and sends a confirmation message to the customer.   
  
Alternative Flow:   
1. If the user attempts to modify an order that is no longer editable (e.g., already shipped), the system displays an error message.   
2. If the system fails to update the order status due to a technical error, it displays an error message and logs the issue for investigation.   
3. If the administrator tries to cancel an order with a payment that cannot be refunded, the system displays a warning and prevents the action.   
4. If the user is not logged in, the system prompts them to log in before managing orders.   
5. If the order is modified (e.g., items removed or added), the system recalculates the total price and updates the Payment Information accordingly.  
  
Use Case Name: Checkout   
Use Case ID: UC-04   
Actors: Customer, System, Administrator   
Preconditions:   
1. The system is operational and accessible.   
2. The customer is logged in and has an active session.   
3. The Shopping Cart contains at least one item.   
4. The Inventory has sufficient stock for the items in the Shopping Cart.   
5. The customer has provided valid Payment Information.   
  
Postconditions:   
1. The customer completes the checkout process and receives an order confirmation.   
2. A new Order is created and stored in the Order data entity.   
3. The Shopping Cart is cleared of the items included in the checkout.   
4. The Inventory quantities are updated to reflect the deduction of ordered items.   
5. The Payment Information is processed and recorded.   
  
Main Flow:   
1. The customer navigates to the checkout page from the Shopping Cart.   
2. The system displays the items in the cart, total price, and requires shipping and billing information.   
3. The customer provides or confirms their shipping and billing details.   
4. The customer selects a preferred payment method and completes the Payment Information form.   
5. The system validates the payment details and processes the payment.   
6. The system creates a new Order in the Order data entity, including items, total price, and customer information.   
7. The system updates the Inventory data entity by reducing the stock of each ordered product.   
8. The system clears the Shopping Cart of the ordered items.   
9. The system sends a confirmation message to the customer and logs the checkout event.   
  
Alternative Flow:   
1. If the customer's provided payment information is invalid, the system displays an error and prompts for correction.   
2. If the system is unable to process the payment, it displays an error and logs the issue for the Administrator to review.   
3. If the Inventory does not have sufficient stock for an item, the system updates the cart and displays an error message.   
4. If the customer cancels the checkout during the process, the system returns them to the Shopping Cart page.   
5. If the system fails to create the order due to a technical error, it displays an error message and logs the issue for investigation.  
  
Use Case Name: Manage Payment Information   
Use Case ID: UC-19   
Actors: Customer, Administrator, System   
Preconditions:   
1. The system is operational and accessible.   
2. The customer is logged in and has an active session.   
3. The customer has at least one item in their Shopping Cart and has initiated the checkout process.   
4. The Payment Information data entity exists and can store or update payment details.   
  
Postconditions:   
1. The customer's payment information is stored, updated, or removed from the Payment Information data entity.   
2. The system validates and processes the payment information for order completion.   
3. The system logs any changes made to the payment information for audit purposes.   
4. The customer receives a confirmation of their payment information being saved or modified.   
  
Main Flow:   
1. The customer navigates to the checkout or payment section of the system.   
2. The system displays the customer's existing payment information, if any, from the Payment Information data entity.   
3. The customer selects an existing payment method or chooses to add a new one.   
4. The customer enters new payment details (e.g., credit card number, expiration date, CVV, billing address).   
5. The system validates the payment information (e.g., correct format, valid card details).   
6. The system stores the new payment information in the Payment Information data entity.   
7. The system confirms the successful storage of the payment method to the customer.   
8. The customer can select this payment method for future orders.   
  
Alternative Flow:   
1. If the customer tries to edit or delete a payment method, the system verifies their identity and permissions.   
2. If the customer provides invalid or incomplete payment details, the system displays an error message and prompts for correction.   
3. If the system fails to store or update the payment information due to a technical error, it displays an error message and logs the issue for investigation.   
4. If the administrator accesses the payment information section, they can view, edit, or delete a customer’s payment records, with appropriate confirmation and logging.   
5. If the administrator attempts to delete a payment method that is associated with an existing order, the system displays a warning and prevents the deletion.  
  
Use Case Name: Manage Account   
Use Case ID: UC-20   
Actors: Customer, Administrator, System   
  
Preconditions:   
1. The system is operational and accessible.   
2. The customer or administrator is logged in and has an active session.   
3. The user has a valid account in the Customer or Administrator data entity.   
4. The user has the necessary permissions to perform account management actions.   
  
Postconditions:   
1. The user's account information is updated, suspended, or deleted based on the action performed.   
2. The system logs the change made to the account for audit purposes.   
3. The user is notified of any changes to their account status or information.   
4. Associated data (e.g., Shopping Cart, Order, Payment Information) is appropriately handled based on the action taken.   
  
Main Flow:   
1. The customer or administrator navigates to the "Account Management" section from their profile or dashboard.   
2. The system displays the user’s current account details, including personal information, contact details, and account status.   
3. The user selects an action to perform (e.g., update personal information, change password, delete account, suspend account).   
4. The system verifies the user’s identity and permissions to perform the selected action.   
5. The user modifies the relevant information or confirms the action (e.g., entering a new password or confirming account deletion).   
6. The system validates the input (e.g., new password strength, valid email format for contact information).   
7. The system updates the Customer or Administrator data entity with the new or modified information.   
8. If the action involves deletion or suspension, the system updates the account status and handles associated data (e.g., clearing the Shopping Cart, archiving orders).   
9. The system logs the action and displays a confirmation message to the user.   
  
Alternative Flow:   
1. If the user does not have sufficient permissions to perform a specific action, the system displays an error message and denies the request.   
2. If the user attempts to delete or suspend their own account, the system requires additional confirmation to proceed.   
3. If the user provides invalid or incomplete information during an update, the system displays an error and prompts for correction.   
4. If the system fails to update the account due to a technical error, it displays an error message and logs the issue for investigation.   
5. If the user cancels the action before submitting, the system returns to the previous screen without making any changes.  
  
Use Case Name: Manage Contact Information   
Use Case ID: UC-21   
Actors: Customer, Administrator, System   
  
Preconditions:   
1. The system is operational and accessible.   
2. The customer or administrator is logged in and has an active session.   
3. The user has a valid account in the Customer or Administrator data entity.   
4. The Contact Information data entity exists and contains the user's current contact details.   
  
Postconditions:   
1. The user's contact information is updated, added, or removed in the Contact Information data entity.   
2. The system logs the changes made to the contact information for audit purposes.   
3. The user receives a confirmation message that their contact information has been successfully modified.   
4. If the administrator modifies a customer’s contact information, the system notifies the customer via email or in-app message.   
  
Main Flow:   
1. The customer or administrator navigates to the "Contact Information" section from their account settings or profile.   
2. The system displays the user’s current contact details, including phone number, address, and email.   
3. The user selects an action (e.g., update contact information, add a new contact method, remove an existing contact method).   
4. The system verifies the user’s identity and permissions to perform the selected action.   
5. The user modifies or enters new contact information (e.g., updates phone number, adds a secondary email).   
6. The system validates the input (e.g., correct phone number format, valid email format).   
7. The system updates the Contact Information data entity with the new or modified details.   
8. The system logs the action and displays a confirmation message to the user.   
9. If the administrator is performing the action on a customer's account, the customer is notified of the update.   
  
Alternative Flow:   
1. If the user provides invalid or improperly formatted contact information, the system displays an error message and prompts for correction.   
2. If the user attempts to remove the primary email or phone number and no valid alternative contact method is provided, the system displays a warning and prevents the removal.   
3. If the system fails to update the Contact Information data entity due to a technical error, it displays an error message and logs the issue for investigation.   
4. If the user cancels the action before submitting, the system returns to the previous screen without making any changes.   
5. If the administrator modifies a customer's contact information and the new email is already registered, the system displays an error and requires a unique email address.  
  
Use Case Name: Manage Order Item   
Use Case ID: UC-22   
Actors: Administrator, System, Order, Product, Inventory   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged in and has an active session.   
3. The administrator has the necessary permissions to modify order items.   
4. The Order data entity contains an existing order with at least one item.   
5. The Product data entity contains the relevant product information.   
6. The Inventory data entity is accessible for stock updates if applicable.   
  
Postconditions:   
1. The Order Item is updated, removed, or modified in the Order data entity.   
2. The Inventory is updated if an item is removed or modified (e.g., restocking).   
3. The system logs the action for audit and security purposes.   
4. The customer is notified of any changes to their order items.   
5. The system recalculates the order total and updates the order accordingly.   
  
Main Flow:   
1. The administrator navigates to the "Manage Orders" section from the admin dashboard.   
2. The system displays a list of orders, and the administrator selects an order to modify.   
3. The system retrieves and displays the items included in the selected order from the Order data entity.   
4. The administrator selects an item within the order to modify or remove.   
5. The system provides options to update the item (e.g., change quantity, remove item) or view product details.   
6. The administrator modifies the item details (e.g., reduces the quantity, removes the item from the order).   
7. The system validates the changes (e.g., quantity not exceeding the original ordered amount, valid product status).   
8. If the item is removed or the quantity is reduced, the system updates the Inventory to restore stock.   
9. The system recalculates the order total based on the changes.   
10. The system updates the Order data entity with the new item details or removal.   
11. The system logs the action and sends a notification to the customer regarding the changes.   
12. The administrator receives a confirmation message that the order item has been successfully managed.   
  
Alternative Flow:   
1. If the administrator attempts to increase the quantity of an order item beyond the original order, the system displays a warning and prevents the change.   
2. If the system fails to update the Inventory due to a technical error, it displays an error message and logs the issue for investigation.   
3. If the administrator tries to modify an order item that is no longer available in the Product data entity, the system displays a warning and suggests alternatives if applicable.   
4. If the system fails to update the Order data entity due to a technical error, it displays an error message and logs the issue for investigation.   
5. If the administrator cancels the modification or removal of an order item, the system returns to the order details without making any changes.   
6. If the system cannot notify the customer due to an invalid email or other delivery issue, it logs the failure and allows the administrator to manually inform the customer.  
  
Use Case Name: Manage Category   
Use Case ID: UC-23   
Actors: Administrator, System   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged in and has an active session.   
3. The administrator has the necessary permissions to manage product categories.   
4. The Category data entity exists and contains the current category structure.   
  
Postconditions:   
1. The Category data entity is updated with new, modified, or removed categories.   
2. The system logs the category management actions for audit purposes.   
3. The administrator receives a confirmation message of the changes.   
4. If a category is modified or deleted, the system ensures that existing product associations are updated or preserved as needed.   
  
Main Flow:   
1. The administrator navigates to the "Category Management" section from the admin dashboard.   
2. The system displays a list of existing categories, including category ID, name, description, and associated products.   
3. The administrator selects an action (e.g., add new category, edit existing category, delete category).   
4. For adding or editing, the administrator fills in or updates category details (e.g., category name, description, parent category if applicable).   
5. The system validates the input (e.g., valid category name, description length, unique category ID for new entries).   
6. The system updates the Category data entity with the new or modified information.   
7. If the category is deleted, the system ensures that all products in the category are reassigned or moved to a default category before deletion.   
8. The system logs the action and displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the administrator enters invalid or incomplete category details, the system displays an error message and prompts for correction.   
2. If the system fails to update the Category data entity due to a technical error, it displays an error message and logs the issue for investigation.   
3. If the administrator attempts to delete a category that contains products, the system displays a warning and suggests reassigning the products first.   
4. If the administrator tries to add a category with a duplicate category ID, the system displays an error and suggests using a unique ID.   
5. If the administrator cancels the action before submitting, the system returns to the previous screen without making any changes.  
  
Use Case Name: Manage Inventory   
Use Case ID: UC-13   
Actors: Administrator, System   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged in and has an active session.   
3. The administrator has the necessary permissions to manage inventory.   
4. The Inventory data entity contains product stock information.   
  
Postconditions:   
1. The Inventory data entity is updated with new or modified stock levels.   
2. The system logs the inventory changes for audit purposes.   
3. The administrator receives a confirmation message of the changes.   
  
Main Flow:   
1. The administrator navigates to the Inventory Management section from the admin dashboard.   
2. The system displays the current inventory levels for all products.   
3. The administrator selects a product to modify its stock level.   
4. The administrator updates the product's stock quantity or adds a new product to the inventory.   
5. The system validates the input data (e.g., numeric quantity, product details).   
6. The system updates the Inventory data entity with the new or modified information.   
7. The system logs the change and displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the administrator enters invalid or non-numeric data for stock quantity, the system displays an error and prompts for correction.   
2. If the system fails to update the inventory due to a technical error, it displays an error message and logs the issue for investigation.   
3. If the administrator attempts to modify a product that does not exist, the system displays an error message and suggests verifying the product ID.   
4. If the administrator tries to set a stock level below zero, the system displays a warning and prevents the update.  
  
Use Case Name: Manage Administrator   
Use Case ID: UC-24   
Actors: System, Administrator   
  
Preconditions:   
1. The system is operational and accessible.   
2. An administrator is logged in and has an active session.   
3. The administrator has the necessary permissions to manage other administrator accounts.   
4. The Administrator data entity exists and contains administrator account information.   
  
Postconditions:   
1. The Administrator data entity is updated with new, modified, or removed administrator accounts.   
2. The system logs the administrator management actions for audit and security purposes.   
3. The administrator receives a confirmation message that the action was successfully performed.   
4. If an administrator is removed, their access to the system is revoked.   
5. If an administrator is added or modified, the changes take effect immediately or after approval if required.   
  
Main Flow:   
1. The administrator navigates to the "Administrator Management" section from the admin dashboard.   
2. The system displays a list of existing administrator accounts, including usernames, roles, and statuses.   
3. The administrator selects an action (e.g., add new administrator, edit administrator details, delete administrator).   
4. For adding or editing, the administrator fills in or updates the relevant details (e.g., username, password, role, permissions).   
5. The system validates the input (e.g., valid email or username format, unique username for new entries, password strength).   
6. The system updates the Administrator data entity with the new or modified information.   
7. The system logs the action and displays a confirmation message to the administrator.   
8. If a new administrator is added, the system may send an invitation or confirmation email to the new administrator.   
9. The system ensures that changes to roles or permissions are applied to the respective administrator account.   
  
Alternative Flow:   
1. If the administrator enters invalid or incomplete information during account creation or modification, the system displays an error message and prompts for correction.   
2. If the system fails to update the Administrator data entity due to a technical error, it displays an error message and logs the issue for investigation.   
3. If the administrator attempts to delete an account that is currently logged in or being used by another session, the system displays a warning and prevents the deletion.   
4. If the administrator tries to add a new administrator with a duplicate username or email, the system displays an error and requires a unique identifier.   
5. If the administrator cancels the action before submitting, the system returns to the previous screen without making any changes.   
6. If the system fails to send an email to a new administrator, the system logs the failure and allows the current administrator to manually inform the new user.