项目文档

# Functional Requirement

### 1. Functional Requirements   
#### 1.1 Customer Registration Function   
\*\*Function ID:\*\* FR-01   
\*\*Description:\*\* Allows new customers to register by submitting personal information. Validates input, creates a customer record, and sends a confirmation email.   
\*\*Input:\*\* Registration form data (Name, Email, Password, Address, Phone).   
\*\*Output:\*\* New record in Customer entity; confirmation email sent.   
  
#### 1.2 Customer Login Function   
\*\*Function ID:\*\* FR-02   
\*\*Description:\*\* Authenticates registered customers using credentials. Grants access upon successful verification.   
\*\*Input:\*\* Email and Password from login form.   
\*\*Output:\*\* Active customer session; session status update in Customer entity.   
  
#### 1.3 Customer Logout Function   
\*\*Function ID:\*\* FR-03   
\*\*Description:\*\* Terminates an active customer session and clears session data.   
\*\*Input:\*\* Logout request from authenticated session.   
\*\*Output:\*\* Session termination; logout activity logged.   
  
#### 1.4 Update Customer Information Function   
\*\*Function ID:\*\* FR-04   
\*\*Description:\*\* Allows customers to modify personal details (e.g., address, phone). Validates inputs before updating records.   
\*\*Input:\*\* Updated fields (Name, Address, Phone, etc.) via editable form.   
\*\*Output:\*\* Modified Customer entity record; confirmation message.   
  
#### 1.5 View Customer Profile Function   
\*\*Function ID:\*\* FR-05   
\*\*Description:\*\* Displays a customer’s profile information and order history.   
\*\*Input:\*\* Customer ID from active session.   
\*\*Output:\*\* Customer profile data (Name, Email, Address, Order History) displayed.   
  
#### 1.6 Delete Customer Function   
\*\*Function ID:\*\* FR-06   
\*\*Description:\*\* Allows administrators to permanently remove customer accounts and associated data.   
\*\*Input:\*\* Administrator selection of Customer ID to delete.   
\*\*Output:\*\* Removal of Customer entity record and linked Shopping Cart/Order data; activity logged.   
  
#### 1.7 Create Product Function   
\*\*Function ID:\*\* FR-07   
\*\*Description:\*\* Enables administrators to add new products with details (name, price, stock).   
\*\*Input:\*\* Product details (Name, Description, Price, StockQuantity, Category) via form.   
\*\*Output:\*\* New Product and Inventory entity records.   
  
#### 1.8 Update Product Details Function   
\*\*Function ID:\*\* FR-08   
\*\*Description:\*\* Allows administrators to edit product attributes and inventory levels.   
\*\*Input:\*\* Modified product fields (e.g., Price, StockQuantity).   
\*\*Output:\*\* Updated Product and Inventory entity records.   
  
#### 1.9 View Product Details Function   
\*\*Function ID:\*\* FR-09   
\*\*Description:\*\* Displays product information to customers or administrators.   
\*\*Input:\*\* Product ID from selection.   
\*\*Output:\*\* Product details (Name, Description, Price, StockStatus) displayed.   
  
#### 1.10 Delete Product Function   
\*\*Function ID:\*\* FR-10   
\*\*Description:\*\* Allows administrators to remove products and adjust inventory.   
\*\*Input:\*\* Administrator selection of Product ID to delete.   
\*\*Output:\*\* Removal of Product and linked Inventory records.   
  
#### 1.11 Create Administrator Function   
\*\*Function ID:\*\* FR-11   
\*\*Description:\*\* Allows privileged administrators to create new admin accounts.   
\*\*Input:\*\* Admin details (Name, Email, Password, Role) via form.   
\*\*Output:\*\* New Administrator entity record; confirmation email sent.   
  
#### 1.12 Update Administrator Information Function   
\*\*Function ID:\*\* FR-12   
\*\*Description:\*\* Enables administrators to modify their account details.   
\*\*Input:\*\* Updated fields (e.g., Password, Contact) via form.   
\*\*Output:\*\* Modified Administrator entity record; confirmation message.   
  
#### 1.13 Delete Administrator Function   
\*\*Function ID:\*\* FR-13   
\*\*Description:\*\* Allows authorized administrators to remove other admin accounts.   
\*\*Input:\*\* Administrator selection of Admin ID to delete.   
\*\*Output:\*\* Removal of Administrator entity record.   
  
#### 1.14 Administrator Login Function   
\*\*Function ID:\*\* FR-14   
\*\*Description:\*\* Authenticates administrators using credentials. Grants dashboard access upon success.   
\*\*Input:\*\* Email and Password from login form.   
\*\*Output:\*\* Active administrator session; session status update.   
  
#### 1.15 Administrator Logout Function   
\*\*Function ID:\*\* FR-15   
\*\*Description:\*\* Terminates an active administrator session.   
\*\*Input:\*\* Logout request from authenticated session.   
\*\*Output:\*\* Session termination; activity logged.   
  
#### 1.16 Create Inventory Function   
\*\*Function ID:\*\* FR-16   
\*\*Description:\*\* Allows administrators to create inventory records linked to products.   
\*\*Input:\*\* Product ID and inventory details (StockLevel, ReorderThreshold).   
\*\*Output:\*\* New Inventory entity record linked to Product.   
  
#### 1.17 Update Inventory Levels Function   
\*\*Function ID:\*\* FR-17   
\*\*Description:\*\* Enables administrators to adjust stock quantities or thresholds.   
\*\*Input:\*\* Inventory ID and updated fields (e.g., StockLevel).   
\*\*Output:\*\* Modified Inventory entity record.   
  
#### 1.18 View Inventory Levels Function   
\*\*Function ID:\*\* FR-18   
\*\*Description:\*\* Displays current inventory data for administrators.   
\*\*Input:\*\* Inventory ID or Product ID.   
\*\*Output:\*\* Inventory details (StockLevel, Location) displayed.   
  
#### 1.19 Delete Inventory Function   
\*\*Function ID:\*\* FR-19   
\*\*Description:\*\* Allows administrators to remove inventory records.   
\*\*Input:\*\* Administrator selection of Inventory ID to delete.   
\*\*Output:\*\* Removal of Inventory entity record.   
  
#### 1.20 Add Product to Shopping Cart Function   
\*\*Function ID:\*\* FR-20   
\*\*Description:\*\* Adds a product to the customer’s shopping cart with specified quantity.   
\*\*Input:\*\* Product ID and Quantity from product page.   
\*\*Output:\*\* Product added/updated in Shopping Cart entity; cart total recalculated.   
  
#### 1.21 View Shopping Cart Function   
\*\*Function ID:\*\* FR-21   
\*\*Description:\*\* Displays contents of the customer’s shopping cart.   
\*\*Input:\*\* Shopping Cart ID from session.   
\*\*Output:\*\* Cart items (Product ID, Quantity, Price) and total cost displayed.   
  
#### 1.22 Update Shopping Cart Function   
\*\*Function ID:\*\* FR-22   
\*\*Description:\*\* Modifies product quantities or removes items from the cart.   
\*\*Input:\*\* Cart Item ID and new Quantity (or removal request).   
\*\*Output:\*\* Updated/removed item in Shopping Cart entity; cart total recalculated.   
  
#### 1.23 Place Order Function   
\*\*Function ID:\*\* FR-23   
\*\*Description:\*\* Converts shopping cart items into an order, updates inventory, and sends confirmation.   
\*\*Input:\*\* Shipping address, Payment method, and Cart ID.   
\*\*Output:\*\* New Order and Order Item records; Inventory updated; confirmation email sent.   
  
#### 1.24 View Order History Function   
\*\*Function ID:\*\* FR-24   
\*\*Description:\*\* Displays a customer’s past orders.   
\*\*Input:\*\* Customer ID from session.   
\*\*Output:\*\* List of orders (Order ID, Date, Total, Status) displayed.   
  
#### 1.25 Confirm Order Function   
\*\*Function ID:\*\* FR-25   
\*\*Description:\*\* Finalizes a pending order after customer review, reducing inventory stock.   
\*\*Input:\*\* Order ID and customer confirmation.   
\*\*Output:\*\* Order status updated to "confirmed"; Inventory levels adjusted.   
  
#### 1.26 Delete Order Function   
\*\*Function ID:\*\* FR-26   
\*\*Description:\*\* Allows administrators to remove orders and optionally restore inventory.   
\*\*Input:\*\* Administrator selection of Order ID to delete.   
\*\*Output:\*\* Removal of Order and Order Item records; Inventory optionally restored.   
  
#### 1.27 Install Plugin Function   
\*\*Function ID:\*\* FR-27   
\*\*Description:\*\* Integrates new plugins into the system after validation.   
\*\*Input:\*\* Plugin file or configuration.   
\*\*Output:\*\* New Plugin entity record; plugin activated.   
  
#### 1.28 Update Plugin Function   
\*\*Function ID:\*\* FR-28   
\*\*Description:\*\* Upgrades existing plugins to new versions.   
\*\*Input:\*\* Updated plugin file or configuration.   
\*\*Output:\*\* Modified Plugin entity record; plugin reactivated.   
  
#### 1.29 View Available Plugins Function   
\*\*Function ID:\*\* FR-29   
\*\*Description:\*\* Lists installed and available plugins for administrators.   
\*\*Input:\*\* None (requires admin session).   
\*\*Output:\*\* Plugin details (Name, Version, Status) displayed.   
  
#### 1.30 Uninstall Plugin Function   
\*\*Function ID:\*\* FR-30   
\*\*Description:\*\* Removes plugins and deactivates their functionality.   
\*\*Input:\*\* Administrator selection of Plugin ID to uninstall.   
\*\*Output:\*\* Removal of Plugin entity record; plugin deactivated.   
  
#### 1.31 Manage Order Item Function   
\*\*Function ID:\*\* FR-31   
\*\*Description:\*\* Allows administrators to modify order items (e.g., quantity, status).   
\*\*Input:\*\* Order Item ID and updated fields (e.g., Status, Quantity).   
\*\*Output:\*\* Modified Order Item record; Inventory adjusted if needed.   
  
#### 1.32 View Administrator Profile Function   
\*\*Function ID:\*\* FR-32   
\*\*Description:\*\* Displays an administrator’s profile information.   
\*\*Input:\*\* Admin ID from session.   
\*\*Output:\*\* Administrator profile data (Name, Email, Role) displayed.   
  
#### 1.33 Create Shopping Cart Function   
\*\*Function ID:\*\* FR-33   
\*\*Description:\*\* Initializes a new shopping cart for a customer or guest.   
\*\*Input:\*\* Customer ID (or guest session token).   
\*\*Output:\*\* New Shopping Cart entity record.   
  
#### 1.34 Delete Shopping Cart Function   
\*\*Function ID:\*\* FR-34   
\*\*Description:\*\* Allows administrators to remove inactive shopping carts.   
\*\*Input:\*\* Administrator selection of Cart ID to delete.   
\*\*Output:\*\* Removal of Shopping Cart entity record.   
  
#### 1.35 Login Administrator Function   
\*\*Function ID:\*\* FR-35   
\*\*Description:\*\* Authenticates administrators to access the admin dashboard.   
\*\*Input:\*\* Email and Password via login form.   
\*\*Output:\*\* Active administrator session; session status updated.

# External Description

### Chapter 2: External Interfaces  
  
#### 2.1 User Interface  
  
The system interacts with users through a graphical user interface (GUI) that includes web pages and forms for both customers and administrators. The interface will be implemented using standard web technologies such as HTML, CSS, and JavaScript. The GUI will provide an intuitive and user-friendly experience, including:  
  
- \*\*Customer Registration Form\*\*: A form for new customers to input personal information such as name, email, password, address, and phone number.  
- \*\*Customer Login Form\*\*: A form for customers to input their email and password to authenticate.  
- \*\*Customer Profile Page\*\*: A page displaying the customer's profile information and order history.  
- \*\*Product Details Page\*\*: A page displaying product information such as name, description, price, and stock status.  
- \*\*Shopping Cart Page\*\*: A page displaying the contents of the customer's shopping cart, including product IDs, quantities, prices, and total cost.  
- \*\*Order History Page\*\*: A page listing the customer's past orders, including order ID, date, total, and status.  
- \*\*Administrator Dashboard\*\*: A dashboard for administrators to manage products, inventory, orders, customer accounts, and plugins.  
  
#### 2.2 Hardware Interface  
  
The system will interact with external hardware devices through standard communication protocols. The primary hardware interfaces include:  
  
- \*\*Printers\*\*: For printing order confirmations, invoices, and shipping labels. The system will support standard printing protocols such as USB and network printing.  
- \*\*Barcode Scanners\*\*: For scanning product barcodes during inventory management and order fulfillment. The system will support standard barcode scanning protocols such as USB and Bluetooth.  
  
#### 2.3 Software Interface  
  
The system will interact with several software components, including databases, third-party services, and plugins. These interfaces are crucial for the system's functionality and include:  
  
- \*\*Database Interface\*\*: The system will use a relational database management system (RDBMS) such as MySQL or PostgreSQL to store data entities such as customers, products, orders, and inventory. The database will be accessed using SQL queries through an ORM (Object-Relational Mapping) tool.  
- \*\*Email Server Interface\*\*: The system will integrate with an email server using the SMTP protocol to send confirmation emails, order notifications, and other system-generated emails.  
- \*\*Payment Gateway Interface\*\*: The system will integrate with a third-party payment gateway such as PayPal or Stripe to process payments. The integration will use RESTful APIs to handle payment processing securely.  
- \*\*Plugin Interface\*\*: The system will support the installation, updating, and uninstallation of plugins. Plugins will be integrated using a standardized plugin framework that allows for easy extension of system functionality.  
  
#### 2.4 Communication Interface  
  
The system will use various communication interfaces to interact with external systems and users. These interfaces include:  
  
- \*\*Web Browser\*\*: The system will be accessible through standard web browsers such as Chrome, Firefox, and Safari. The system will use HTTP/HTTPS protocols to communicate with the web server.  
- \*\*Email Notifications\*\*: The system will send email notifications for events such as registration confirmation, order confirmation, and password reset. These notifications will be sent using the SMTP protocol.  
- \*\*API Communication\*\*: The system will use RESTful APIs to communicate with third-party services such as payment gateways and email servers. The APIs will be secured using HTTPS and authentication mechanisms such as OAuth or API keys.  
  
By carefully defining these external interfaces, the system will be able to interact seamlessly with users, hardware, and other software components, ensuring a robust and scalable solution.

# Use Case

Use Case Name: Register Customer   
Use Case ID: UC-01   
Actors: Customer, Administrator, System   
Preconditions:   
1. The system is operational and accessible.   
2. The customer is not yet registered in the system.   
3. The administrator has provided access to the registration interface.   
  
Postconditions:   
1. A new customer account is created in the system.   
2. The customer receives a confirmation message.   
3. The customer is added to the customer data entity with valid information.   
  
Main Flow:   
1. The customer accesses the registration page of the system.   
2. The system displays the customer registration form.   
3. The customer fills in the required information (e.g., name, email, password, address).   
4. The customer submits the registration form.   
5. The system validates the input data (e.g., checks for valid email format, unique email, password strength).   
6. The system creates a new customer record in the Customer data entity.   
7. The system sends a confirmation message to the customer's email.   
8. The system logs the registration activity for administrative review.   
9. The system redirects the customer to a welcome page or login page.   
  
Alternative Flow:   
1. If the system detects invalid or incomplete data (e.g., missing fields, invalid email format), it displays an error message and prompts the customer to correct the input.   
2. If the email address is already registered, the system informs the customer that the email is taken and suggests using a different one.   
3. If the system fails to send the confirmation email, the system logs an error and notifies the administrator for resolution.  
  
Use Case Name: Login Customer   
Use Case ID: UC-02   
Actors: Customer, System, Administrator   
  
Preconditions:   
1. The system is operational and accessible.   
2. The customer has already registered an account in the system.   
3. The customer is not currently logged in.   
  
Postconditions:   
1. The customer is successfully logged into the system.   
2. The system updates the customer's session status.   
3. The system logs the login activity for administrative review.   
  
Main Flow:   
1. The customer navigates to the login page of the system.   
2. The system displays the login form with fields for email and password.   
3. The customer enters their registered email and password.   
4. The customer submits the login form.   
5. The system verifies the email and password against the Customer data entity.   
6. If the credentials are valid, the system grants access to the customer's account.   
7. The system creates a session for the customer and redirects them to their dashboard or home page.   
  
Alternative Flow:   
1. If the system detects invalid credentials (e.g., incorrect email or password), it displays an error message and prompts the customer to try again.   
2. If the customer enters the wrong password multiple times, the system locks the account temporarily and notifies the administrator.   
3. If the customer account is inactive or suspended, the system displays an appropriate message and prevents login.  
  
Use Case Name: Logout Customer   
Use Case ID: UC-03   
Actors: Customer, System   
  
Preconditions:   
1. The system is operational and accessible.   
2. The customer is currently logged in.   
  
Postconditions:   
1. The customer is no longer logged in.   
2. The system terminates the customer's session.   
3. The system logs the logout activity for administrative review.   
  
Main Flow:   
1. The customer accesses the account settings or navigation menu.   
2. The system displays the logout option.   
3. The customer selects the logout option.   
4. The system confirms the logout request.   
5. The system terminates the active session and clears session data.   
6. The system redirects the customer to the login or home page.   
  
Alternative Flow:   
1. If the customer cancels the logout request, the system retains the session and returns to the previous page.   
2. If the system fails to terminate the session, it displays an error message and logs the issue for administrator review.  
  
Use Case Name: Update Customer Information   
Use Case ID: UC-04   
Actors: Customer, System, Administrator   
  
Preconditions:   
1. The system is operational and accessible.   
2. The customer is already registered and logged in.   
3. The customer has access to the account settings or personal information page.   
  
Postconditions:   
1. The customer's information is updated in the Customer data entity.   
2. The system logs the update activity for administrative review.   
3. The customer receives a confirmation message of the successful update.   
  
Main Flow:   
1. The customer navigates to the account settings or personal information page.   
2. The system displays the current customer information in an editable form.   
3. The customer modifies the desired fields (e.g., name, address, contact information).   
4. The customer submits the updated information.   
5. The system validates the new input data (e.g., checks for valid email format, password strength if changed).   
6. The system updates the Customer data entity with the new information.   
7. The system sends a confirmation message to the customer.   
8. The system logs the update activity and redirects the customer to the account dashboard.   
  
Alternative Flow:   
1. If the system detects invalid or incomplete data (e.g., missing required fields, invalid email format), it displays an error message and prompts the customer to correct the input.   
2. If the customer attempts to change the email to one already registered, the system displays an error and suggests using a different email.   
3. If the system fails to update the information, it displays an error message and logs the issue for administrator review.  
  
Use Case Name: View Customer Profile   
Use Case ID: UC-05   
Actors: Customer, System, Administrator   
  
Preconditions:   
1. The system is operational and accessible.   
2. The customer is logged in.   
3. The customer has an existing profile in the Customer data entity.   
  
Postconditions:   
1. The customer's profile information is displayed to them.   
2. The system logs the profile view activity for administrative review.   
3. The customer can choose to update or view additional details.   
  
Main Flow:   
1. The customer navigates to the "View Profile" section from their account dashboard.   
2. The system retrieves the customer's information from the Customer data entity.   
3. The system displays the customer's profile information (e.g., name, email, address, order history).   
4. The system logs the profile view activity.   
5. The customer can choose to update information or return to the dashboard.   
  
Alternative Flow:   
1. If the system is unable to retrieve the customer's profile data, it displays an error message and logs the issue for administrator review.   
2. If the customer profile has been deleted or is not found, the system displays a message indicating the profile is unavailable and suggests contacting support.   
3. If the customer tries to access another user's profile, the system denies access and displays an error message.  
  
Use Case Name: Delete Customer   
Use Case ID: UC-06   
Actors: Administrator, System, Customer   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged in and has the necessary permissions to delete customer accounts.   
3. The customer to be deleted has an existing account in the Customer data entity.   
  
Postconditions:   
1. The customer's account is removed from the Customer data entity.   
2. The system logs the deletion activity for administrative review.   
3. The shopping cart and order history associated with the customer are either deleted or archived as per system policies.   
  
Main Flow:   
1. The administrator navigates to the customer management interface.   
2. The system displays a list of existing customer accounts.   
3. The administrator selects the customer to be deleted.   
4. The system confirms the deletion request.   
5. The administrator confirms the deletion.   
6. The system deletes the customer's account from the Customer data entity.   
7. The system removes the customer's associated data from the Shopping Cart and Order data entities.   
8. The system logs the deletion activity.   
9. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the system detects that the customer has active orders or unpaid invoices, it displays a warning and prevents deletion until resolved.   
2. If the administrator cancels the deletion request, the system retains the customer account and returns to the customer management interface.   
3. If the system fails to delete the account due to technical issues, it displays an error message and logs the issue for administrator review.  
  
Use Case Name: Create Product   
Use Case ID: UC-07   
Actors: Administrator, System, Product, Inventory   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged in and has the necessary permissions to create products.   
3. The Inventory data entity is accessible and ready to update.   
  
Postconditions:   
1. A new product is added to the Product data entity.   
2. The product information is updated in the Inventory data entity.   
3. The system logs the creation activity for administrative review.   
  
Main Flow:   
1. The administrator navigates to the product management interface.   
2. The system displays the "Create Product" form.   
3. The administrator fills in the required product details (e.g., product name, description, price, stock quantity, category, image).   
4. The administrator submits the form.   
5. The system validates the input data (e.g., checks for missing fields, valid price format, valid stock quantity).   
6. The system creates a new product record in the Product data entity.   
7. The system updates the Inventory data entity with the initial stock quantity.   
8. The system logs the creation activity and displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the system detects invalid or incomplete data (e.g., missing product name, invalid price format), it displays an error message and prompts the administrator to correct the input.   
2. If the product name already exists, the system warns the administrator and suggests a unique name or edits the existing product instead.   
3. If the system fails to update the inventory, it displays an error message and logs the issue for administrator review.  
  
Use Case Name: Update Product Details   
Use Case ID: UC-08   
Actors: Administrator, System, Product, Inventory   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged in and has the necessary permissions to update product details.   
3. The product to be updated exists in the Product data entity.   
  
Postconditions:   
1. The product details are updated in the Product data entity.   
2. The Inventory data entity is updated if stock quantity or other inventory-related fields are modified.   
3. The system logs the update activity for administrative review.   
  
Main Flow:   
1. The administrator navigates to the product management interface.   
2. The system displays a list of existing products.   
3. The administrator selects the product to update and clicks "Edit."   
4. The system displays the product details in an editable form.   
5. The administrator modifies the necessary fields (e.g., product name, description, price, stock quantity, category, image).   
6. The administrator submits the updated form.   
7. The system validates the new input data (e.g., checks for missing fields, valid price format, valid stock quantity).   
8. The system updates the Product data entity with the new information.   
9. If stock quantity or inventory-related fields are modified, the system updates the Inventory data entity accordingly.   
10. The system logs the update activity and displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the system detects invalid or incomplete data (e.g., missing product name, invalid price format), it displays an error message and prompts the administrator to correct the input.   
2. If the product name is changed to one that already exists, the system warns the administrator and suggests a unique name or edits the existing product instead.   
3. If the system fails to update the product or inventory data due to technical issues, it displays an error message and logs the issue for administrator review.  
  
Use Case Name: View Product Details   
Use Case ID: UC-09   
Actors: Customer, System, Product, Administrator   
  
Preconditions:   
1. The system is operational and accessible.   
2. The customer is logged in or can browse the system as a guest.   
3. The product to be viewed exists in the Product data entity.   
  
Postconditions:   
1. The customer is presented with detailed information about the selected product.   
2. The system logs the product view activity for administrative review.   
3. The customer may choose to add the product to the shopping cart or continue browsing.   
  
Main Flow:   
1. The customer navigates to the product listing page or searches for a product.   
2. The customer selects a specific product from the list.   
3. The system retrieves the product details from the Product data entity.   
4. The system displays the product information (e.g., name, description, price, stock status, images).   
5. The system logs the product view activity.   
6. The customer may choose to add the product to their Shopping Cart or return to the product listing.   
  
Alternative Flow:   
1. If the system is unable to retrieve the product details, it displays an error message and logs the issue for administrator review.   
2. If the product is out of stock, the system displays a message indicating the unavailability and suggests alternatives if available.   
3. If the customer is not logged in, the system may prompt them to create an account or proceed as a guest to view the product details.  
  
Use Case Name: Delete Product   
Use Case ID: UC-10   
Actors: Administrator, System, Product, Inventory   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged in and has the necessary permissions to delete products.   
3. The product to be deleted exists in the Product data entity.   
4. The Inventory data entity is accessible and contains the relevant stock information.   
  
Postconditions:   
1. The product is removed from the Product data entity.   
2. The Inventory data entity is updated to remove or adjust stock records associated with the product.   
3. The system logs the deletion activity for administrative review.   
4. The product is no longer visible to customers in the product listing or search results.   
  
Main Flow:   
1. The administrator navigates to the product management interface.   
2. The system displays a list of existing products.   
3. The administrator selects the product to delete and clicks "Delete."   
4. The system confirms the deletion request.   
5. The administrator confirms the deletion.   
6. The system removes the product record from the Product data entity.   
7. The system updates the Inventory data entity by removing or adjusting the stock quantity related to the deleted product.   
8. The system logs the deletion activity.   
9. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the system detects that the product is associated with active orders or shopping carts, it displays a warning and prevents deletion until the references are resolved.   
2. If the administrator cancels the deletion request, the system retains the product and returns to the product management interface.   
3. If the system fails to delete the product or update the inventory due to technical issues, it displays an error message and logs the issue for administrator review.  
  
Use Case Name: Create Administrator   
Use Case ID: UC-11   
Actors: System, Administrator   
  
Preconditions:   
1. The system is operational and accessible.   
2. The current user has the necessary permissions to create a new administrator account.   
3. The Administrator data entity is accessible and ready to be updated.   
  
Postconditions:   
1. A new administrator account is created in the Administrator data entity.   
2. The system logs the creation activity for administrative review.   
3. The administrator receives a confirmation message.   
  
Main Flow:   
1. The current administrator navigates to the administrator management interface.   
2. The system displays the "Create Administrator" form.   
3. The administrator fills in the required information (e.g., full name, email, password, role).   
4. The administrator submits the form.   
5. The system validates the input data (e.g., checks for valid email format, unique email, password strength).   
6. The system creates a new administrator record in the Administrator data entity.   
7. The system logs the creation activity.   
8. The system sends a confirmation message to the new administrator's email.   
9. The system displays a success message to the current administrator.   
  
Alternative Flow:   
1. If the system detects invalid or incomplete data (e.g., missing fields, invalid email format), it displays an error message and prompts the administrator to correct the input.   
2. If the email address is already registered for another administrator, the system informs the administrator that the email is taken and suggests using a different one.   
3. If the system fails to send the confirmation email, it logs an error and notifies the current administrator for resolution.  
  
Use Case Name: Update Administrator Information   
Use Case ID: UC-12   
Actors: Administrator, System   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged in and has the necessary permissions to update their information.   
3. The administrator has an existing account in the Administrator data entity.   
  
Postconditions:   
1. The administrator's information is updated in the Administrator data entity.   
2. The system logs the update activity for administrative review.   
3. The administrator receives a confirmation message of the successful update.   
  
Main Flow:   
1. The administrator navigates to the account settings or personal information page.   
2. The system displays the current administrator information in an editable form.   
3. The administrator modifies the desired fields (e.g., name, contact information, password).   
4. The administrator submits the updated information.   
5. The system validates the new input data (e.g., checks for valid email format, password strength if changed).   
6. The system updates the Administrator data entity with the new information.   
7. The system sends a confirmation message to the administrator.   
8. The system logs the update activity and redirects the administrator to the account dashboard.   
  
Alternative Flow:   
1. If the system detects invalid or incomplete data (e.g., missing required fields, invalid email format), it displays an error message and prompts the administrator to correct the input.   
2. If the administrator attempts to change the email to one already registered, the system displays an error and suggests using a different email.   
3. If the system fails to update the information, it displays an error message and logs the issue for administrator review.  
  
Use Case Name: Delete Administrator   
Use Case ID: UC-13   
Actors: Administrator, System   
  
Preconditions:   
1. The system is operational and accessible.   
2. The current administrator is logged in and has the necessary permissions to delete other administrator accounts.   
3. The administrator to be deleted has an existing account in the Administrator data entity.   
  
Postconditions:   
1. The administrator's account is removed from the Administrator data entity.   
2. The system logs the deletion activity for administrative review.   
3. The administrator receives a confirmation message.   
  
Main Flow:   
1. The current administrator navigates to the administrator management interface.   
2. The system displays a list of existing administrator accounts.   
3. The current administrator selects the administrator to be deleted.   
4. The system confirms the deletion request.   
5. The current administrator confirms the deletion.   
6. The system deletes the selected administrator's account from the Administrator data entity.   
7. The system logs the deletion activity.   
8. The system displays a confirmation message to the current administrator.   
  
Alternative Flow:   
1. If the system detects that the selected administrator has active responsibilities or pending actions, it displays a warning and may prevent deletion until the situation is resolved.   
2. If the current administrator cancels the deletion request, the system retains the administrator account and returns to the administrator management interface.   
3. If the system fails to delete the administrator's account due to technical issues, it displays an error message and logs the issue for administrator review.  
  
Use Case Name: Login Administrator   
Use Case ID: UC-14   
Actors: Administrator, System   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator has a valid account in the Administrator data entity.   
3. The administrator is not currently logged in.   
  
Postconditions:   
1. The administrator is successfully logged into the system.   
2. The system updates the administrator's session status.   
3. The system logs the login activity for administrative review.   
  
Main Flow:   
1. The administrator navigates to the login page of the system.   
2. The system displays the login form with fields for email and password.   
3. The administrator enters their registered email and password.   
4. The administrator submits the login form.   
5. The system verifies the email and password against the Administrator data entity.   
6. If the credentials are valid, the system grants access to the administrator's account.   
7. The system creates a session for the administrator and redirects them to the admin dashboard.   
  
Alternative Flow:   
1. If the system detects invalid credentials (e.g., incorrect email or password), it displays an error message and prompts the administrator to try again.   
2. If the administrator enters the wrong password multiple times, the system locks the account temporarily and logs the activity.   
3. If the administrator account is inactive or suspended, the system displays an appropriate message and prevents login.  
  
Use Case Name: Logout Administrator   
Use Case ID: UC-15   
Actors: Administrator, System   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is currently logged in.   
  
Postconditions:   
1. The administrator is no longer logged in.   
2. The system terminates the administrator's session.   
3. The system logs the logout activity for administrative review.   
  
Main Flow:   
1. The administrator accesses the account settings or navigation menu.   
2. The system displays the logout option.   
3. The administrator selects the logout option.   
4. The system confirms the logout request.   
5. The system terminates the active session and clears session data.   
6. The system redirects the administrator to the login or home page.   
  
Alternative Flow:   
1. If the administrator cancels the logout request, the system retains the session and returns to the previous page.   
2. If the system fails to terminate the session, it displays an error message and logs the issue for administrator review.  
  
Use Case Name: Create Inventory   
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 the necessary permissions to create inventory records.   
3. The Product data entity contains valid products that can be associated with inventory.   
4. The Inventory data entity is accessible and ready to be updated.   
  
Postconditions:   
1. A new inventory record is added to the Inventory data entity.   
2. The inventory data is linked to an existing product in the Product data entity.   
3. The system logs the creation activity for administrative review.   
4. The inventory status is updated and available for viewing and managing by the administrator.   
  
Main Flow:   
1. The administrator navigates to the inventory management interface.   
2. The system displays the "Create Inventory" form.   
3. The administrator selects a product from the Product data entity to associate with the inventory.   
4. The administrator enters inventory details (e.g., stock quantity, warehouse location, restock threshold, supplier information).   
5. The administrator submits the form.   
6. The system validates the input data (e.g., checks for valid stock quantity, valid warehouse location).   
7. The system creates a new inventory record in the Inventory data entity and links it to the selected product.   
8. The system logs the creation activity.   
9. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the system detects invalid or incomplete data (e.g., negative stock quantity, missing warehouse location), it displays an error message and prompts the administrator to correct the input.   
2. If the selected product does not exist in the Product data entity, the system displays an error message and prevents inventory creation until a valid product is selected.   
3. If the system fails to create the inventory record due to technical issues, it displays an error message and logs the issue for administrator review.  
  
Use Case Name: Update Inventory Levels   
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 the necessary permissions to update inventory levels.   
3. The product exists in the Product data entity and has an associated inventory record in the Inventory data entity.   
4. The inventory data to be updated is available and accessible in the Inventory data entity.   
  
Postconditions:   
1. The inventory levels are updated in the Inventory data entity.   
2. The product's stock status is adjusted accordingly in the system.   
3. The system logs the inventory update activity for administrative review.   
4. The updated inventory information is available for viewing and for use in customer interactions.   
  
Main Flow:   
1. The administrator navigates to the inventory management interface.   
2. The system displays a list of inventory records for existing products.   
3. The administrator selects the inventory record to be updated and clicks "Edit."   
4. The system displays the current inventory details (e.g., product name, stock quantity, warehouse location, restock threshold).   
5. The administrator modifies the necessary fields (e.g., updates stock quantity, changes warehouse location).   
6. The administrator submits the updated form.   
7. The system validates the input data (e.g., checks for valid stock quantity, valid warehouse location).   
8. The system updates the Inventory data entity with the new information.   
9. The system logs the update activity.   
10. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the system detects invalid or incomplete data (e.g., negative stock quantity, missing warehouse location), it displays an error message and prompts the administrator to correct the input.   
2. If the product is not available in the system, the system displays an error message and prevents the update.   
3. If the system fails to update the inventory due to technical issues, it displays an error message and logs the issue for administrator review.  
  
Use Case Name: View Inventory Levels   
Use Case ID: UC-18   
Actors: Administrator, System, Inventory, Product   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged in and has the necessary permissions to view inventory levels.   
3. The Inventory data entity contains valid inventory records.   
4. The selected product has an associated inventory record.   
  
Postconditions:   
1. The administrator is presented with the current inventory levels of the selected product.   
2. The system logs the inventory view activity for administrative review.   
3. The administrator may decide to update inventory levels or take other actions based on the displayed information.   
  
Main Flow:   
1. The administrator navigates to the inventory management interface.   
2. The system displays a list of products with their inventory records.   
3. The administrator selects a product to view its inventory levels.   
4. The system retrieves the inventory details for the selected product from the Inventory data entity.   
5. The system displays the product's inventory levels (e.g., current stock quantity, warehouse location, restock threshold).   
6. The system logs the inventory view activity.   
7. The administrator can choose to return to the inventory list or proceed to update the inventory levels.   
  
Alternative Flow:   
1. If the system is unable to retrieve the inventory data for the selected product, it displays an error message and logs the issue for administrator review.   
2. If the product does not have an associated inventory record, the system displays a message indicating that no inventory data is available.   
3. If the administrator is not logged in or lacks the required permissions, the system denies access and displays an appropriate error message.  
  
Use Case Name: Delete Inventory   
Use Case ID: UC-19   
Actors: Administrator, System, Product, Inventory   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged in and has the necessary permissions to delete inventory records.   
3. The inventory record to be deleted exists in the Inventory data entity.   
4. The associated product exists in the Product data entity.   
  
Postconditions:   
1. The inventory record is removed from the Inventory data entity.   
2. The system logs the deletion activity for administrative review.   
3. The inventory information is no longer available for viewing or managing.   
  
Main Flow:   
1. The administrator navigates to the inventory management interface.   
2. The system displays a list of inventory records.   
3. The administrator selects the inventory record to be deleted and clicks "Delete."   
4. The system confirms the deletion request.   
5. The administrator confirms the deletion.   
6. The system removes the inventory record from the Inventory data entity.   
7. The system logs the deletion activity.   
8. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the system detects that the inventory record is linked to active orders or shopping carts, it displays a warning and prevents deletion until the references are resolved.   
2. If the administrator cancels the deletion request, the system retains the inventory record and returns to the inventory management interface.   
3. If the system fails to delete the inventory record due to technical issues, it displays an error message and logs the issue for administrator review.  
  
Use Case Name: Add Product to Shopping Cart   
Use Case ID: UC-20   
Actors: Customer, System, Product, Shopping Cart   
  
Preconditions:   
1. The system is operational and accessible.   
2. The customer is logged in or is browsing as a guest.   
3. The product to be added exists in the Product data entity.   
4. The product has available stock in the Inventory data entity.   
5. The Shopping Cart is accessible for the current customer session.   
  
Postconditions:   
1. The selected product is added to the Shopping Cart.   
2. The quantity of the product in the Shopping Cart is updated if it already exists.   
3. The system logs the addition of the product to the Shopping Cart for administrative review.   
4. The Shopping Cart reflects the updated contents for the customer.   
  
Main Flow:   
1. The customer views a product detail page.   
2. The customer clicks the "Add to Cart" button for the product.   
3. The system checks the product's availability in the Inventory data entity.   
4. The system adds the product to the Shopping Cart or updates the quantity if it already exists.   
5. The system displays a confirmation message (e.g., "Product added to cart").   
6. The system logs the action in the system activity log.   
7. The customer is redirected back to the product page or shopping cart overview.   
  
Alternative Flow:   
1. If the product is out of stock, the system displays an error message and prevents the addition to the cart.   
2. If the customer attempts to add more quantity than available in inventory, the system displays a warning and limits the addition to available stock.   
3. If the Shopping Cart is not accessible due to technical issues, the system displays an error message and logs the issue for administrator review.  
  
Use Case Name: View Shopping Cart   
Use Case ID: UC-21   
Actors: Customer, System, Shopping Cart, Product, Inventory   
  
Preconditions:   
1. The system is operational and accessible.   
2. The customer is logged in or is browsing as a guest with an active Shopping Cart session.   
3. The Shopping Cart data entity exists and is accessible.   
4. The customer has at least one product in their Shopping Cart.   
  
Postconditions:   
1. The customer is presented with the contents of their Shopping Cart.   
2. The system logs the shopping cart view activity for administrative review.   
3. The customer may choose to modify the cart, proceed to checkout, or continue browsing.   
  
Main Flow:   
1. The customer navigates to the "Shopping Cart" section from the navigation menu or a confirmation page after adding a product.   
2. The system retrieves the customer's Shopping Cart data from the Shopping Cart data entity.   
3. The system displays the list of products in the cart, including product name, quantity, price, and total amount.   
4. The system updates the cart display if there are changes in inventory (e.g., product out of stock, price adjustments).   
5. The system logs the shopping cart view activity.   
6. The customer may choose to continue shopping, update cart items, or proceed to checkout.   
  
Alternative Flow:   
1. If the system detects that the Shopping Cart is empty, it displays a message indicating no items are in the cart and suggests browsing products.   
2. If the system is unable to retrieve the Shopping Cart data, it displays an error message and logs the issue for administrator review.   
3. If a product in the cart is no longer available (e.g., out of stock), the system displays a warning and removes or disables the item from the cart.  
  
Use Case Name: Update Shopping Cart   
Use Case ID: UC-22   
Actors: Customer, System, Shopping Cart, Product, Inventory   
  
Preconditions:   
1. The system is operational and accessible.   
2. The customer is logged in or is browsing as a guest with an active Shopping Cart session.   
3. The Shopping Cart data entity exists and contains at least one product.   
4. The product to be modified exists in the Product data entity and has an associated inventory record.   
  
Postconditions:   
1. The customer's Shopping Cart is updated with the new product quantity or removed items.   
2. The system logs the update activity for administrative review.   
3. The Shopping Cart reflects the updated contents for the customer.   
  
Main Flow:   
1. The customer navigates to the "Shopping Cart" section from the navigation menu.   
2. The system displays the contents of the Shopping Cart, including product names, quantities, prices, and total amount.   
3. The customer selects a product to modify (e.g., change quantity or remove it).   
4. The customer updates the quantity or clicks "Remove" for the selected product.   
5. The system validates the requested quantity (e.g., ensures it is not higher than available stock).   
6. The system updates the Shopping Cart data entity with the modified quantity or removes the product if requested.   
7. The system recalculates the cart total and updates the display.   
8. The system logs the update activity.   
9. The system displays a confirmation message to the customer (e.g., "Cart updated successfully").   
  
Alternative Flow:   
1. If the requested quantity exceeds the available stock, the system displays an error message and limits the quantity to the available stock.   
2. If the system is unable to update the Shopping Cart due to technical issues, it displays an error message and logs the issue for administrator review.   
3. If the product is no longer available (e.g., removed from inventory), the system displays a warning and removes the item from the cart automatically.  
  
Use Case Name: Delete Shopping Cart Item   
Use Case ID: UC-23   
Actors: Customer, System, Shopping Cart, Product, Inventory   
  
Preconditions:   
1. The system is operational and accessible.   
2. The customer is logged in or is browsing as a guest with an active Shopping Cart session.   
3. The Shopping Cart data entity exists and contains at least one product.   
4. The product to be removed exists in the Shopping Cart and is associated with a valid product in the Product data entity.   
  
Postconditions:   
1. The selected item is removed from the Shopping Cart data entity.   
2. The system logs the deletion activity for administrative review.   
3. The Shopping Cart display is updated to reflect the removal of the item.   
  
Main Flow:   
1. The customer navigates to the "Shopping Cart" section from the navigation menu.   
2. The system displays the contents of the Shopping Cart, including product names, quantities, prices, and total amount.   
3. The customer selects a product and clicks the "Remove" button.   
4. The system confirms the removal request.   
5. The customer confirms the removal.   
6. The system deletes the selected item from the Shopping Cart data entity.   
7. The system recalculates the cart total and updates the display.   
8. The system logs the deletion activity.   
9. The system displays a confirmation message to the customer (e.g., "Item removed from cart successfully").   
  
Alternative Flow:   
1. If the system is unable to locate the item in the Shopping Cart, it displays an error message and logs the issue for administrator review.   
2. If the customer cancels the removal request, the system retains the item in the cart and returns to the cart view.   
3. If the system fails to update the cart due to technical issues, it displays an error message and logs the issue for administrator review.  
  
Use Case Name: Place Order   
Use Case ID: UC-24   
Actors: Customer, Shopping Cart, Product, Inventory, Order, System, Administrator   
  
Preconditions:   
1. The system is operational and accessible.   
2. The customer is logged in or is browsing as a guest with an active Shopping Cart session.   
3. The Shopping Cart contains at least one product.   
4. The product(s) in the Shopping Cart are in stock and available for purchase.   
5. The customer has provided valid shipping and payment information.   
  
Postconditions:   
1. A new Order is created in the Order data entity.   
2. The product quantities in the Inventory data entity are updated to reflect the order.   
3. The Shopping Cart is cleared or updated if partial order is placed.   
4. The system logs the order placement activity for administrative review.   
5. The customer receives a confirmation message of the order.   
  
Main Flow:   
1. The customer navigates to the "Checkout" page from the Shopping Cart.   
2. The system displays the order summary, including product details, quantity, price, and total amount.   
3. The customer provides or confirms their shipping address and payment method.   
4. The system validates the shipping and payment information.   
5. The system checks the Inventory data entity to ensure product availability.   
6. The system creates a new Order record in the Order data entity with the selected items.   
7. The system updates the Inventory data entity by reducing the stock quantities for the ordered products.   
8. The system clears the Shopping Cart or updates it if partial order is allowed.   
9. The system sends a confirmation message to the customer's email.   
10. The system logs the order placement activity.   
11. The customer is redirected to an order confirmation page.   
  
Alternative Flow:   
1. If the system detects that a product in the cart is no longer in stock, it displays a warning and prevents the order from being placed until the issue is resolved.   
2. If the customer provides invalid or incomplete shipping/payment information, the system displays an error message and prompts for correction.   
3. If the system fails to update the Inventory data entity due to technical issues, it displays an error message and logs the issue for administrator review.   
4. If the system fails to send the confirmation email, it logs an error and notifies the administrator for resolution.  
  
Use Case Name: View Order History   
Use Case ID: UC-25   
Actors: Customer, System, Order   
  
Preconditions:   
1. The system is operational and accessible.   
2. The customer is logged in.   
3. The customer has at least one order in the Order data entity.   
4. The Order data entity is accessible and contains valid order records.   
  
Postconditions:   
1. The customer is presented with a list of their previous orders.   
2. The system logs the order history view activity for administrative review.   
3. The customer may choose to view specific order details or return to their dashboard.   
  
Main Flow:   
1. The customer navigates to the "Order History" section from their account dashboard.   
2. The system retrieves the customer's order records from the Order data entity.   
3. The system displays a list of orders, including order ID, date, total amount, and order status.   
4. The system logs the order history view activity.   
5. The customer may click on a specific order to view detailed information (e.g., items ordered, shipping details).   
6. The system displays the selected order details.   
7. The customer can return to the order history list or proceed to other account functions.   
  
Alternative Flow:   
1. If the system is unable to retrieve the order history, it displays an error message and logs the issue for administrator review.   
2. If the customer has no order history, the system displays a message indicating no previous orders.   
3. If the customer tries to access another user's order history, the system denies access and displays an error message.  
  
Use Case Name: Confirm Order   
Use Case ID: UC-26   
Actors: Customer, System, Shopping Cart, Product, Inventory, Order, Administrator   
  
Preconditions:   
1. The system is operational and accessible.   
2. The customer has placed an order in the system (UC-24).   
3. The order is in a pending or review state.   
4. The customer is logged in or has an active session.   
  
Postconditions:   
1. The order is confirmed and moved to a processed or awaiting fulfillment state.   
2. The Inventory data entity is updated to reflect the reduction in stock for the ordered products.   
3. The system logs the order confirmation activity for administrative review.   
4. The customer receives a confirmation message and order details.   
  
Main Flow:   
1. The customer navigates to the "Confirm Order" page from the checkout or order summary.   
2. The system displays a summary of the order, including product details, quantity, price, and total amount.   
3. The customer reviews the order and clicks the "Confirm Order" button.   
4. The system verifies the order details and checks the inventory for product availability.   
5. The system updates the Inventory data entity by reducing the stock quantity for the ordered products.   
6. The system updates the Order data entity to mark the order as confirmed.   
7. The system sends a confirmation message to the customer's email.   
8. The system logs the confirmation activity.   
9. The customer is redirected to an order confirmation page with a summary of the completed transaction.   
  
Alternative Flow:   
1. If the system detects that a product in the order is no longer in stock or has been modified since the order was placed, it displays an error message and prevents confirmation until the issue is resolved.   
2. If the customer cancels the order confirmation, the system retains the order in a pending state and returns to the order summary page.   
3. If the system fails to update the inventory or order status due to technical issues, it displays an error message and logs the issue for administrator review.  
  
Use Case Name: Delete Order   
Use Case ID: UC-27   
Actors: Administrator, System, Order, Customer   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged in and has the necessary permissions to delete orders.   
3. The order to be deleted exists in the Order data entity.   
4. The customer associated with the order has been identified.   
  
Postconditions:   
1. The order is removed from the Order data entity.   
2. The system logs the deletion activity for administrative review.   
3. The inventory quantities for the ordered products are optionally restored, depending on system policies.   
4. The customer may receive a notification of the deletion, if required.   
  
Main Flow:   
1. The administrator navigates to the order management interface.   
2. The system displays a list of orders, including order details such as order ID, customer, date, and status.   
3. The administrator selects the order to be deleted and clicks the "Delete" option.   
4. The system confirms the deletion request.   
5. The administrator confirms the deletion.   
6. The system removes the order record from the Order data entity.   
7. The system logs the deletion activity.   
8. The system optionally updates the Inventory data entity to restore stock levels.   
9. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the system detects that the order has already been fulfilled, it displays a warning and prevents deletion until the status is changed to pending.   
2. If the administrator cancels the deletion request, the system retains the order and returns to the order management interface.   
3. If the system fails to delete the order or update the inventory due to technical issues, it displays an error message and logs the issue for administrator review.  
  
Use Case Name: Install Plugin   
Use Case ID: UC-28   
Actors: Administrator, System, Plugin   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged in and has the necessary permissions to install plugins.   
3. The Plugin data entity is accessible and ready to be updated.   
4. A valid plugin file or configuration is available for installation.   
  
Postconditions:   
1. The plugin is added to the Plugin data entity.   
2. The system logs the plugin installation activity for administrative review.   
3. The plugin is activated and integrated into the system functionality.   
4. The administrator receives a confirmation message of the successful installation.   
  
Main Flow:   
1. The administrator navigates to the plugin management interface.   
2. The system displays a list of available plugins or provides an option to upload a new plugin.   
3. The administrator selects a plugin to install or uploads a new one.   
4. The system validates the plugin (e.g., checks for compatibility, correct format, and required metadata).   
5. The system installs the plugin and updates the Plugin data entity.   
6. The system activates the plugin and integrates it into the system.   
7. The system logs the installation activity.   
8. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the system detects an incompatible or invalid plugin, it displays an error message and prevents the installation.   
2. If the plugin already exists in the Plugin data entity, the system warns the administrator and suggests updating the existing plugin instead.   
3. If the system fails to install or activate the plugin due to technical issues, it displays an error message and logs the issue for administrator review.  
  
Use Case Name: Update Plugin   
Use Case ID: UC-29   
Actors: Administrator, System, Plugin   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged in and has the necessary permissions to update plugins.   
3. The Plugin data entity contains a valid and installed plugin that needs to be updated.   
4. A new or updated version of the plugin file or configuration is available for upload.   
  
Postconditions:   
1. The plugin in the Plugin data entity is updated to the new version.   
2. The system logs the plugin update activity for administrative review.   
3. The updated plugin is activated and integrated into the system functionality.   
4. The administrator receives a confirmation message of the successful update.   
  
Main Flow:   
1. The administrator navigates to the plugin management interface.   
2. The system displays a list of installed plugins with their current versions.   
3. The administrator selects a plugin and clicks the "Update" option.   
4. The system validates the current plugin (e.g., confirms it is installed and active).   
5. The administrator uploads the new plugin file or selects an updated version from the system.   
6. The system validates the new plugin file (e.g., checks for compatibility, correct format, and required metadata).   
7. The system updates the Plugin data entity with the new version details.   
8. The system deactivates the old plugin, applies the update, and reactivates the new version.   
9. The system logs the update activity.   
10. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the system detects an incompatible or invalid plugin file, it displays an error message and prevents the update.   
2. If the plugin is already the latest version, the system informs the administrator and suggests no action.   
3. If the system fails to deactivate or update the plugin due to technical issues, it displays an error message and logs the issue for administrator review.  
  
Use Case Name: View Available Plugins   
Use Case ID: UC-30   
Actors: Administrator, System, Plugin   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged in and has the necessary permissions to view plugins.   
3. The Plugin data entity contains a list of available or installed plugins.   
  
Postconditions:   
1. The administrator is presented with a list of available plugins.   
2. The system logs the plugin view activity for administrative review.   
3. The administrator may choose to install, update, or configure the plugins.   
  
Main Flow:   
1. The administrator navigates to the plugin management interface.   
2. The system retrieves the list of available plugins from the Plugin data entity.   
3. The system displays the plugin names, descriptions, versions, and status (e.g., installed, available, or outdated).   
4. The system logs the plugin view activity.   
5. The administrator reviews the plugin information and can choose to take further actions (e.g., install or update).   
  
Alternative Flow:   
1. If the system is unable to retrieve the plugin list, it displays an error message and logs the issue for administrator review.   
2. If no plugins are available, the system displays a message indicating that no plugins are currently available.   
3. If the administrator tries to access the plugin list without proper permissions, the system denies access and displays an appropriate error message.  
  
Use Case Name: Uninstall Plugin   
Use Case ID: UC-31   
Actors: Administrator, System, Plugin   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged in and has the necessary permissions to uninstall plugins.   
3. The Plugin data entity contains a valid and installed plugin.   
  
Postconditions:   
1. The plugin is removed from the Plugin data entity.   
2. The system logs the uninstallation activity for administrative review.   
3. The plugin is deactivated and its functionality is no longer available in the system.   
4. The administrator receives a confirmation message of the successful uninstallation.   
  
Main Flow:   
1. The administrator navigates to the plugin management interface.   
2. The system displays a list of installed plugins with their details.   
3. The administrator selects a plugin and clicks the "Uninstall" option.   
4. The system confirms the uninstallation request.   
5. The administrator confirms the uninstallation.   
6. The system deactivates the selected plugin and removes it from the Plugin data entity.   
7. The system logs the uninstallation activity.   
8. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the system detects that the plugin is in use or has dependencies, it displays a warning and prevents uninstallation until the dependencies are resolved.   
2. If the administrator cancels the uninstallation request, the system retains the plugin and returns to the plugin management interface.   
3. If the system fails to uninstall the plugin due to technical issues, it displays an error message and logs the issue for administrator review.  
  
Use Case Name: Manage Order Item   
Use Case ID: UC-32   
Actors: Administrator, System, Order, Order Item   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged in and has the necessary permissions to manage order items.   
3. The Order data entity contains an existing order.   
4. The selected order has at least one Order Item associated with it.   
  
Postconditions:   
1. The administrator can modify, remove, or update the status of an Order Item.   
2. The Order data entity is updated to reflect the changes in the associated Order Items.   
3. The system logs the order item management activity for administrative review.   
4. The inventory levels are optionally adjusted based on the changes made to the Order Item.   
  
Main Flow:   
1. The administrator navigates to the order management interface.   
2. The system displays a list of orders with their order IDs and statuses.   
3. The administrator selects a specific order to manage its items.   
4. The system retrieves the list of Order Items associated with the selected order from the Order Item data entity.   
5. The system displays the Order Items with details such as product name, quantity, price, and status.   
6. The administrator selects an Order Item to modify, remove, or update its status (e.g., mark as shipped, canceled, or pending).   
7. The system validates the requested changes (e.g., ensures valid quantity, status, and product availability).   
8. The system updates the Order Item data entity with the new information.   
9. If the change affects inventory (e.g., item is canceled and stock is to be restored), the system updates the Inventory data entity accordingly.   
10. The system logs the activity and displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the system detects that the requested quantity is invalid or inconsistent with the product's inventory levels, it displays an error message and prevents the update.   
2. If the Order Item status cannot be changed due to business rules (e.g., a shipped item cannot be canceled), the system displays a warning and prevents the modification.   
3. If the system fails to update the Order Item or Inventory data entities due to technical issues, it displays an error message and logs the issue for administrator review.  
  
Use Case Name: View Administrator Profile   
Use Case ID: UC-33   
Actors: Administrator, System   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged in.   
3. The administrator has an existing profile in the Administrator data entity.   
  
Postconditions:   
1. The administrator's profile information is displayed to them.   
2. The system logs the profile view activity for administrative review.   
3. The administrator can choose to update information or view additional details.   
  
Main Flow:   
1. The administrator navigates to the "View Profile" section from their admin dashboard.   
2. The system retrieves the administrator's information from the Administrator data entity.   
3. The system displays the administrator's profile information (e.g., full name, email, role, account status).   
4. The system logs the profile view activity.   
5. The administrator can choose to update their information or return to the dashboard.   
  
Alternative Flow:   
1. If the system is unable to retrieve the administrator's profile data, it displays an error message and logs the issue for administrator review.   
2. If the administrator profile has been deleted or is not found, the system displays a message indicating the profile is unavailable and suggests contacting support.   
3. If the administrator tries to access another administrator's profile, the system denies access and displays an error message.  
  
Use Case Name: Create Shopping Cart   
Use Case ID: UC-34   
Actors: Customer, System, Shopping Cart   
  
Preconditions:   
1. The system is operational and accessible.   
2. The customer is logged in or is browsing as a guest.   
3. The Shopping Cart data entity is accessible and ready to be initialized.   
4. The customer has not yet created a shopping cart session or has a cleared cart.   
  
Postconditions:   
1. A new Shopping Cart session is created for the customer.   
2. The Shopping Cart data entity is initialized with no items.   
3. The system logs the creation activity for administrative review.   
4. The customer can now add products to the cart and proceed with checkout.   
  
Main Flow:   
1. The customer accesses the product listing page or views a product detail page.   
2. The system detects that the customer does not have an active Shopping Cart session or the cart is empty.   
3. The system initializes a new Shopping Cart session for the customer.   
4. The system creates a new record in the Shopping Cart data entity.   
5. The system logs the creation of the Shopping Cart session.   
6. The system displays the Shopping Cart as available for product addition (e.g., through a "Add to Cart" button).   
7. The customer can now proceed to add products to the cart.   
  
Alternative Flow:   
1. If the system is unable to create a new Shopping Cart session due to technical issues, it displays an error message and logs the issue for administrator review.   
2. If the customer already has an active cart and attempts to create a new one, the system retains the existing cart and displays a message indicating that a cart is already active.   
3. If the customer is logged in and the system fails to associate the cart with the customer account, it displays an error message and logs the issue for administrator review.  
  
Use Case Name: Delete Shopping Cart   
Use Case ID: UC-35   
Actors: Administrator, System, Customer, Shopping Cart   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged in and has the necessary permissions to delete shopping cart records.   
3. The Shopping Cart data entity exists and contains the cart to be deleted.   
4. The customer associated with the cart is identified.   
  
Postconditions:   
1. The selected Shopping Cart is removed from the Shopping Cart data entity.   
2. The system logs the deletion activity for administrative review.   
3. The customer is no longer able to access the deleted cart.   
4. The inventory levels are not affected by the deletion.   
  
Main Flow:   
1. The administrator navigates to the shopping cart management interface.   
2. The system displays a list of active shopping carts with details such as customer, cart contents, and timestamp.   
3. The administrator selects the shopping cart to be deleted and clicks the "Delete" option.   
4. The system confirms the deletion request.   
5. The administrator confirms the deletion.   
6. The system removes the selected Shopping Cart record from the Shopping Cart data entity.   
7. The system logs the deletion activity.   
8. The system displays a confirmation message to the administrator indicating the cart has been deleted.   
  
Alternative Flow:   
1. If the system detects that the shopping cart contains items that are part of an active order, it displays a warning and prevents deletion until the order is finalized or removed.   
2. If the administrator cancels the deletion request, the system retains the shopping cart and returns to the shopping cart management interface.   
3. If the system fails to delete the shopping cart due to technical issues, it displays an error message and logs the issue for administrator review.