项目文档

# Functional Requirement

# 1. Functional Requirements  
  
## 1.1 Customer Registration Function   
\*\*Function ID:\*\* FR-01   
\*\*Description:\*\* Customers can register for an account by providing valid personal information, including email and password. The system must validate the email format, password strength, and ensure the email is not already registered.   
\*\*Input:\*\*   
- Name (string)   
- Email (string)   
- Password (string)   
- Optional: Address (string)   
\*\*Output:\*\*   
- Confirmation email sent to the customer's email address   
- Redirect to the login page   
- New customer account created in the system   
  
---  
  
## 1.2 Customer Login Function   
\*\*Function ID:\*\* FR-02   
\*\*Description:\*\* Customers can log into their account by entering their email and password. The system must validate the credentials, check if the account is active and verified, and log the login attempt.   
\*\*Input:\*\*   
- Email (string)   
- Password (string)   
\*\*Output:\*\*   
- Redirect to the homepage or account dashboard   
- Login record added to the system log   
- Session created for the logged-in customer   
  
---  
  
## 1.3 Customer Logout Function   
\*\*Function ID:\*\* FR-03   
\*\*Description:\*\* Customers can securely log out of their account. The system must terminate the session and redirect the customer to the homepage or login page.   
\*\*Input:\*\*   
- Session ID (string)   
- User ID (string)   
\*\*Output:\*\*   
- Session terminated   
- Redirect to the homepage or login page   
- Logout record added to the system log   
  
---  
  
## 1.4 Admin Login Function   
\*\*Function ID:\*\* FR-04   
\*\*Description:\*\* Administrators can log into the admin interface by entering their username and password. The system must validate the credentials, check if the account is active, and log the login attempt.   
\*\*Input:\*\*   
- Username (string)   
- Password (string)   
\*\*Output:\*\*   
- Redirect to the admin dashboard   
- Login record added to the system log   
- Session created for the logged-in admin   
  
---  
  
## 1.5 Admin Logout Function   
\*\*Function ID:\*\* FR-05   
\*\*Description:\*\* Administrators can securely log out of their account. The system must terminate the admin session and redirect them to the admin login page or homepage.   
\*\*Input:\*\*   
- Session ID (string)   
- Admin ID (string)   
\*\*Output:\*\*   
- Session terminated   
- Redirect to the admin login page or homepage   
- Logout record added to the system log   
  
---  
  
## 1.6 View Product Details Function   
\*\*Function ID:\*\* FR-06   
\*\*Description:\*\* Customers can view detailed information about a product, including name, description, price, specifications, and images.   
\*\*Input:\*\*   
- Product ID (string)   
- User ID (string)   
\*\*Output:\*\*   
- Displayed product details (name, description, price, specifications, images)   
- Option to add the product to the cart   
  
---  
  
## 1.7 Add to Cart Function   
\*\*Function ID:\*\* FR-07   
\*\*Description:\*\* Customers can add products to their shopping cart. The system must check product availability and inventory, and update the cart with the selected product and quantity.   
\*\*Input:\*\*   
- Product ID (string)   
- Quantity (integer)   
- Customer ID (string)   
\*\*Output:\*\*   
- Updated shopping cart with new item and quantity   
- Updated cart summary (total cost, item count)   
- Confirmation message that the product was added   
  
---  
  
## 1.8 Modify Cart Item Function   
\*\*Function ID:\*\* FR-08   
\*\*Description:\*\* Customers can modify the quantity of an existing item in their cart or remove it. The system must validate the updated quantity against inventory and update the cart summary accordingly.   
\*\*Input:\*\*   
- Cart Item ID (string)   
- New Quantity (integer) or removal request   
- Customer ID (string)   
\*\*Output:\*\*   
- Updated cart item with new quantity or removed item   
- Updated cart summary (total cost, item count)   
- Confirmation message of the modification   
  
---  
  
## 1.9 Remove Cart Item Function   
\*\*Function ID:\*\* FR-09   
\*\*Description:\*\* Customers can remove a specific item from their shopping cart. The system must verify the existence of the cart item and update the cart summary after removal.   
\*\*Input:\*\*   
- Cart Item ID (string)   
- Customer ID (string)   
\*\*Output:\*\*   
- Cart item removed from the database   
- Updated cart summary (total cost, item count)   
- Confirmation message that the item was removed   
  
---  
  
## 1.10 Checkout Order Function   
\*\*Function ID:\*\* FR-10   
\*\*Description:\*\* Customers can proceed to checkout with items in their cart. The system must validate the cart, process the selected payment method, update inventory, and send an order confirmation email.   
\*\*Input:\*\*   
- Customer ID (string)   
- Cart ID (string)   
- Payment method (string)   
- Shipping address (string)   
\*\*Output:\*\*   
- Created order record in the database   
- Updated inventory levels   
- Order confirmation email sent to the customer   
- Cart cleared after successful checkout   
  
---  
  
## 1.11 Confirm Order Function   
\*\*Function ID:\*\* FR-11   
\*\*Description:\*\* After the payment is processed, the system must confirm the order, update inventory, and send an email confirmation to the customer.   
\*\*Input:\*\*   
- Order ID (string)   
- Payment confirmation status (string)   
- Inventory system access   
\*\*Output:\*\*   
- Finalized order in the database   
- Inventory updated with deducted stock   
- Order confirmation email sent   
- Cart cleared   
  
---  
  
## 1.12 Send Order Confirmation Email Function   
\*\*Function ID:\*\* FR-12   
\*\*Description:\*\* The system must send an order confirmation email to the customer after the order is successfully placed. The email must contain order details such as items, total cost, and shipping information.   
\*\*Input:\*\*   
- Order ID (string)   
- Customer email (string)   
- Order details (items, total cost, shipping address)   
\*\*Output:\*\*   
- Order confirmation email sent to the customer   
- Email delivery status logged in the system   
- Updated order status indicating confirmation email was sent   
  
---  
  
## 1.13 Create Product Function   
\*\*Function ID:\*\* FR-13   
\*\*Description:\*\* Administrators can create a new product by entering its details and uploading documentation or images. The system must validate the input data and save the product in the database.   
\*\*Input:\*\*   
- Product name (string)   
- Description (string)   
- Price (float)   
- Category ID (string)   
- Specifications (string)   
- Documentation/image files   
\*\*Output:\*\*   
- New product record in the database   
- Product displayed in the catalog   
- Confirmation message shown to the administrator   
  
---  
  
## 1.14 Update Product Information Function   
\*\*Function ID:\*\* FR-14   
\*\*Description:\*\* Administrators can update the information of an existing product, including name, description, price, and category. The system must validate the new data and update the product in the database.   
\*\*Input:\*\*   
- Product ID (string)   
- Updated product name (string)   
- Updated description (string)   
- Updated price (float)   
- Updated category ID (string)   
- Updated documentation/image files (optional)   
\*\*Output:\*\*   
- Updated product record in the database   
- Updated product displayed in the catalog   
- Confirmation message shown to the administrator   
  
---  
  
## 1.15 Manage Inventory Function   
\*\*Function ID:\*\* FR-15   
\*\*Description:\*\* Administrators can update the inventory levels for products. The system must validate the new inventory quantity and update the database accordingly.   
\*\*Input:\*\*   
- Product ID (string)   
- New inventory quantity (integer)   
\*\*Output:\*\*   
- Updated inventory record in the database   
- Refreshed product catalog reflecting new inventory status   
- Confirmation message shown to the administrator   
  
---  
  
## 1.16 Categorize Products Function   
\*\*Function ID:\*\* FR-16   
\*\*Description:\*\* Administrators can assign or reassign a product to a specific category. The system must validate the selected category and update the product's category association in the database.   
\*\*Input:\*\*   
- Product ID (string)   
- Category ID (string)   
\*\*Output:\*\*   
- Updated product category in the database   
- Updated product catalog reflecting new categorization   
- Confirmation message shown to the administrator   
  
---  
  
## 1.17 Manage Plugin Function   
\*\*Function ID:\*\* FR-17   
\*\*Description:\*\* Administrators can install, update, or remove plugins from the system. The system must validate the plugin file and manage its status in the database.   
\*\*Input:\*\*   
- Plugin ID (string)   
- Plugin file or repository link (string)   
- Action (install, update, remove)   
\*\*Output:\*\*   
- Plugin status updated in the database   
- Plugin displayed as active/inactive in the plugin management interface   
- Confirmation message shown to the administrator   
  
---  
  
## 1.18 Access Plugin Documentation Function   
\*\*Function ID:\*\* FR-18   
\*\*Description:\*\* Administrators can view the documentation for a plugin. The system must retrieve and display the documentation in a structured and user-friendly format.   
\*\*Input:\*\*   
- Plugin ID (string)   
- Admin ID (string)   
\*\*Output:\*\*   
- Plugin documentation displayed (installation instructions, usage examples, etc.)   
- Option to search or navigate within the documentation   
- Error message if documentation is missing or corrupted   
  
---  
  
## 1.19 View Purchase History Function   
\*\*Function ID:\*\* FR-19   
\*\*Description:\*\* Customers can view their purchase history, which includes all completed orders. The system must retrieve and display order details such as date, items, and total cost.   
\*\*Input:\*\*   
- Customer ID (string)   
\*\*Output:\*\*   
- List of past orders with key details (order date, total cost, status)   
- Detailed view of selected order (items, quantities, shipping address, payment method)   
- Error message if no orders are found   
  
---  
  
## 1.20 Calculate Total Cost Function   
\*\*Function ID:\*\* FR-20   
\*\*Description:\*\* The system must calculate the total cost of the cart based on the quantity and price of each item.   
\*\*Input:\*\*   
- Cart Item list (CartItem objects with Product ID and Quantity)   
- Product Price (from Product entity)   
\*\*Output:\*\*   
- Updated TotalCost value for the cart   
- Cart summary updated with the new total cost   
  
---  
  
## 1.21 View Cart Summary Function   
\*\*Function ID:\*\* FR-21   
\*\*Description:\*\* The system must display a summary of the customer's shopping cart, including the number of items and the total cost.   
\*\*Input:\*\*   
- Cart ID (string)   
- Cart Item list (CartItem objects)   
- TotalCost (float)   
\*\*Output:\*\*   
- Cart summary displayed with item count and total cost   
- Error message if cart is empty   
  
---  
  
## 1.22 Manage Quantity Function   
\*\*Function ID:\*\* FR-22   
\*\*Description:\*\* The system must manage the quantity of each item in the cart, ensuring it is valid and does not exceed available inventory.   
\*\*Input:\*\*   
- Cart Item ID (string)   
- New quantity (integer)   
\*\*Output:\*\*   
- Updated quantity in the cart item   
- Updated TotalCost in the cart   
- Error message if the quantity is invalid or exceeds available stock   
  
---  
  
## 1.23 Manage Purchase History Function   
\*\*Function ID:\*\* FR-23   
\*\*Description:\*\* The system must maintain and manage the purchase history of each customer, recording each completed order and its details.   
\*\*Input:\*\*   
- Customer ID (string)   
- Order ID (string)   
- Product ID (string)   
- Purchase date (datetime)   
- Quantity (integer)   
\*\*Output:\*\*   
- Updated PurchaseHistory record in the database   
- Purchase history displayed on the customer account page   
  
---  
  
## 1.24 Manage User Function   
\*\*Function ID:\*\* FR-24   
\*\*Description:\*\* Administrators can manage user accounts, including creating, updating, or deactivating customer and admin accounts.   
\*\*Input:\*\*   
- User ID (string)   
- Username (string)   
- Email (string)   
- Password (string)   
- Status (string)   
- Role (string)   
\*\*Output:\*\*   
- Updated or created user account in the database   
- Confirmation message displayed to the admin   
- User account status reflected in the interface   
  
---  
  
These functional requirements cover the core capabilities of the \*\*GAMMA-J Web Store\*\* system as defined by the use cases and data model. Each function is designed to be feasible, consistent, and aligned with the system’s goals of providing secure, intuitive, and extensible online store management.

# External Description

# Chapter 2: External Interfaces  
  
This chapter defines the external interfaces of the \*\*GAMMA-J Web Store\*\* system. These interfaces represent how the system interacts with users, other software systems, hardware devices, and communication networks to fulfill its functional requirements.  
  
---  
  
## 2.1 User Interface  
  
The user interface is the primary way users interact with the system. It includes web pages, forms, dashboards, and confirmation messages displayed to customers and administrators.  
  
### 2.1.1 Customer Registration Page  
- \*\*Description:\*\* A form where customers enter their personal information (name, email, password, optional address).  
- \*\*Inputs:\*\* Name, Email, Password, Address (optional)  
- \*\*Outputs:\*\* Confirmation message, redirect to login page, and a new account created in the database.  
- \*\*Interaction Method:\*\* Web-based HTML/JavaScript form submission via HTTP POST request.  
  
### 2.1.2 Customer Login Page  
- \*\*Description:\*\* A form where customers enter their email and password to log in.  
- \*\*Inputs:\*\* Email, Password  
- \*\*Outputs:\*\* Redirect to homepage or dashboard, session created, login record logged.  
- \*\*Interaction Method:\*\* Web-based HTML/JavaScript form submission via HTTP POST request.  
  
### 2.1.3 Admin Login Page  
- \*\*Description:\*\* A form where administrators enter their username and password to access the admin interface.  
- \*\*Inputs:\*\* Username, Password  
- \*\*Outputs:\*\* Redirect to admin dashboard, session created, login record logged.  
- \*\*Interaction Method:\*\* Web-based HTML/JavaScript form submission via HTTP POST request.  
  
### 2.1.4 Product Detail View  
- \*\*Description:\*\* Displays detailed product information including name, description, price, specifications, and images.  
- \*\*Inputs:\*\* Product ID, User ID  
- \*\*Outputs:\*\* Displayed product details, option to add to cart.  
- \*\*Interaction Method:\*\* Web-based dynamic content rendering using REST API calls.  
  
### 2.1.5 Cart Summary Page  
- \*\*Description:\*\* Displays a summary of items in the shopping cart, including total cost and item count.  
- \*\*Inputs:\*\* Cart ID, Cart Item list  
- \*\*Outputs:\*\* Cart summary with item count and total cost, error if cart is empty.  
- \*\*Interaction Method:\*\* Web-based dynamic content rendering using REST API calls.  
  
### 2.1.6 Order Confirmation Page  
- \*\*Description:\*\* Displays order confirmation details after checkout, including items, total cost, and shipping address.  
- \*\*Inputs:\*\* Order ID, Customer ID  
- \*\*Outputs:\*\* Order confirmation display, error if no orders found.  
- \*\*Interaction Method:\*\* Web-based dynamic content rendering using REST API calls.  
  
### 2.1.7 Purchase History Page  
- \*\*Description:\*\* Displays a list of completed orders for the customer.  
- \*\*Inputs:\*\* Customer ID  
- \*\*Outputs:\*\* List of past orders with key details (order date, total cost, status), detailed view of selected order.  
- \*\*Interaction Method:\*\* Web-based dynamic content rendering using REST API calls.  
  
### 2.1.8 Admin Dashboard  
- \*\*Description:\*\* A centralized interface for administrators to manage products, inventory, users, and plugins.  
- \*\*Inputs:\*\* Admin ID, Session ID  
- \*\*Outputs:\*\* Access to various administrative functions, confirmation messages, and logs.  
- \*\*Interaction Method:\*\* Web-based dynamic content rendering using REST API calls.  
  
### 2.1.9 Plugin Management Interface  
- \*\*Description:\*\* Allows administrators to install, update, or remove plugins.  
- \*\*Inputs:\*\* Plugin ID, Action (install/update/remove), Plugin file/repository link  
- \*\*Outputs:\*\* Updated plugin status, confirmation message.  
- \*\*Interaction Method:\*\* Web-based dynamic content rendering using REST API calls.  
  
### 2.1.10 Plugin Documentation Viewer  
- \*\*Description:\*\* Displays documentation for a specific plugin.  
- \*\*Inputs:\*\* Plugin ID, Admin ID  
- \*\*Outputs:\*\* Structured documentation content, search/navigation options, error if documentation is missing.  
- \*\*Interaction Method:\*\* Web-based dynamic content rendering using REST API calls.  
  
---  
  
## 2.2 Hardware Interface  
  
Currently, the GAMMA-J Web Store does not interact directly with external hardware devices. Therefore, this section is \*\*not applicable\*\*.  
  
---  
  
## 2.3 Software Interface  
  
The system interacts with various software components and external services through APIs, databases, and internal modules.  
  
### 2.3.1 Database Interface  
- \*\*Description:\*\* The system communicates with a relational database to store and retrieve user data, product information, inventory levels, orders, and logs.  
- \*\*Input:\*\* SQL queries, JSON payloads, etc.  
- \*\*Output:\*\* Data retrieved from the database, updates made to the database.  
- \*\*Interaction Method:\*\* RESTful API or direct database connection using ORM tools.  
- \*\*Data Sources:\*\*  
 - `users` table  
 - `products` table  
 - `orders` table  
 - `cart\_items` table  
 - `purchase\_history` table  
 - `plugins` table  
 - `logs` table  
  
### 2.3.2 Email Service Interface  
- \*\*Description:\*\* The system uses an email service to send confirmation emails to customers.  
- \*\*Input:\*\* Email content, recipient address, subject, order details  
- \*\*Output:\*\* Sent email, delivery status logged  
- \*\*Interaction Method:\*\* SMTP protocol or third-party email API (e.g., SendGrid, Amazon SES)  
- \*\*Data Sources:\*\*  
 - Customer email address  
 - Order details (items, total cost, shipping address)  
  
### 2.3.3 Payment Gateway Interface  
- \*\*Description:\*\* The system integrates with a payment gateway to process transactions.  
- \*\*Input:\*\* Payment method, amount, customer details, order information  
- \*\*Output:\*\* Payment confirmation status, transaction ID  
- \*\*Interaction Method:\*\* REST API or SDK provided by the payment provider (e.g., Stripe, PayPal)  
- \*\*Data Sources:\*\*  
 - Payment method (credit card, digital wallet, etc.)  
 - Customer ID  
 - Cart ID  
 - Order details  
  
### 2.3.4 Inventory Management System  
- \*\*Description:\*\* The system interacts with an internal inventory management module to update stock levels.  
- \*\*Input:\*\* Product ID, New inventory quantity  
- \*\*Output:\*\* Updated inventory records  
- \*\*Interaction Method:\*\* Internal API call within the system  
- \*\*Data Sources:\*\*  
 - `inventory` table  
 - `products` table  
  
### 2.3.5 Logging System  
- \*\*Description:\*\* The system logs all user actions, such as login attempts, order creation, and plugin management.  
- \*\*Input:\*\* Log entry (action type, timestamp, user ID, details)  
- \*\*Output:\*\* Stored log records in the system database  
- \*\*Interaction Method:\*\* Internal logging framework or external logging service  
- \*\*Data Sources:\*\*  
 - `logs` table  
 - User ID  
 - Session ID  
  
---  
  
## 2.4 Communication Interface  
  
The system communicates with external entities via network protocols and messaging systems.  
  
### 2.4.1 Email Communication Interface  
- \*\*Description:\*\* The system sends order confirmation emails to customers.  
- \*\*Input:\*\* Order details, customer email, subject line  
- \*\*Output:\*\* Email sent, delivery status recorded  
- \*\*Interaction Method:\*\* SMTP protocol or third-party email API  
- \*\*Data Sources:\*\*  
 - Customer email address  
 - Order details (items, total cost, shipping address)  
  
### 2.4.2 Web Browser Communication Interface  
- \*\*Description:\*\* The system communicates with web browsers via HTTP/HTTPS protocols to serve web pages and handle user interactions.  
- \*\*Input:\*\* HTTP requests (GET, POST, PUT, DELETE)  
- \*\*Output:\*\* HTTP responses containing HTML, JSON, or XML data  
- \*\*Interaction Method:\*\* RESTful API endpoints  
- \*\*Data Sources:\*\*  
 - User input data (form submissions)  
 - Session tokens  
 - Product catalog  
 - Order history  
  
### 2.4.3 API Communication Interface  
- \*\*Description:\*\* The system exposes and consumes RESTful APIs for internal and external integration.  
- \*\*Input:\*\* API requests with query parameters, headers, and body data  
- \*\*Output:\*\* API responses with status codes, data payloads, or error messages  
- \*\*Interaction Method:\*\* HTTP/HTTPS with JSON/XML payload  
- \*\*Data Sources:\*\*  
 - User accounts  
 - Product catalog  
 - Orders  
 - Logs  
 - Plugins  
  
---  
  
## Summary of External Interfaces  
  
| Interface Type | Description | Interaction Method | Data Sources Involved |  
|----------------------|-----------------------------------------------------------------------------|----------------------------|----------------------------------------------------------------------------------------|  
| User Interface | All user-facing screens and forms | Web-based HTML/JS | Users, Products, Orders, Cart, Logs, Purchase History, Plugins |  
| Software Interface | Integration with database, payment gateways, inventory, and logging systems | REST API, SQL, ORM | Users, Products, Orders, Cart, Logs, Purchase History, Inventory, Plugins |  
| Communication Interface | Email notifications, web browsing, and API calls | SMTP, HTTP/HTTPS, REST | Users, Products, Orders, Cart, Logs, Purchase History, Inventory, Plugins |  
  
This chapter ensures that all external interactions are clearly defined, consistent with the functional requirements, and ready for implementation.

# Use Case

Use Case Name: Customer Registration   
Use Case ID: UC-01   
Actors: Customer, System   
Preconditions:   
- The system is operational.   
- The customer has access to the system interface.   
- The customer is not already registered in the system.   
  
Postconditions:   
- A new customer account is created in the system.   
- The customer receives a confirmation email.   
- The customer is redirected to the login page.   
  
Main Flow:   
1. The customer navigates to the registration page.   
2. The system displays the registration form.   
3. The customer fills in the required information (e.g., name, email, password).   
4. The customer submits the registration form.   
5. The system validates the input data (e.g., email format, password strength).   
6. The system creates a new customer account in the database.   
7. The system sends a confirmation email to the customer's provided email address.   
8. The system redirects the customer to the login page.   
  
Alternative Flow:   
1. The system detects invalid email format in step 5.   
 a. The system displays an error message and prompts the customer to correct the email.   
 b. The customer revises the email and resubmits the form.   
 c. The system revalidates the email format.   
  
2. The system detects that the email is already registered in step 5.   
 a. The system displays an error message indicating that the email is already in use.   
 b. The customer provides a new email and resubmits the form.   
 c. The system revalidates the new email.   
  
3. The system detects a weak password in step 5.   
 a. The system displays an error message and prompts the customer to choose a stronger password.   
 b. The customer revises the password and resubmits the form.   
 c. The system revalidates the password strength.  
  
Use Case Name: Customer Login   
Use Case ID: UC-02   
Actors: Customer, System   
Preconditions:   
- The system is operational.   
- The customer has a valid account in the system.   
- The customer has access to the login interface.   
  
Postconditions:   
- The customer is authenticated and logged into their account.   
- The customer is redirected to their account dashboard or the homepage.   
- A login record is added to the system's log.   
  
Main Flow:   
1. The customer navigates to the login page.   
2. The system displays the login form.   
3. The customer enters their email and password.   
4. The customer clicks the "Login" button.   
5. The system validates the entered credentials against the database.   
6. The system confirms the customer's identity.   
7. The system logs the customer in and redirects them to the homepage or account dashboard.   
  
Alternative Flow:   
1. The system detects invalid email or password in step 5.   
 a. The system displays an error message indicating incorrect login details.   
 b. The customer revises their email or password and resubmits the form.   
 c. The system revalidates the credentials.   
  
2. The system detects that the customer account is locked or disabled in step 5.   
 a. The system displays an error message stating that the account is locked or disabled.   
 b. The customer may request account recovery or contact support.   
  
3. The system detects that the customer has not completed email verification in step 5.   
 a. The system displays a prompt to verify the email address.   
 b. The customer clicks on a "Resend Verification Email" link.   
 c. The system sends a new verification email to the customer.  
  
Use Case Name: Customer Logout   
Use Case ID: UC-03   
Actors: Customer, System   
Preconditions:   
- The system is operational.   
- The customer is logged into their account.   
- The customer has access to the logout functionality.   
  
Postconditions:   
- The customer is logged out of their account.   
- The session is terminated.   
- The customer is redirected to the homepage or login page.   
  
Main Flow:   
1. The customer navigates to the account settings or user menu.   
2. The system displays the logout option.   
3. The customer selects the "Logout" button.   
4. The system verifies the customer's session.   
5. The system terminates the active session and clears session data.   
6. The system redirects the customer to the homepage or login page.   
  
Alternative Flow:   
1. The system detects an invalid or expired session in step 4.   
 a. The system displays an error message indicating the session is invalid.   
 b. The customer is redirected to the login page automatically.  
  
Use Case Name: Create Product   
Use Case ID: UC-04   
Actors: Administrator, System   
Preconditions:   
- The system is operational.   
- The administrator is logged into their account.   
- The administrator has access to the product management interface.   
  
Postconditions:   
- A new product is added to the system's database.   
- The product is displayed in the product catalog.   
- A confirmation message is shown to the administrator.   
  
Main Flow:   
1. The administrator navigates to the product management section.   
2. The system displays the "Create Product" form.   
3. The administrator fills in the product details (e.g., name, description, price, category, quantity, and specifications).   
4. The administrator uploads relevant product documentation or images.   
5. The administrator submits the form.   
6. The system validates the input data (e.g., correct price format, available category, valid specifications).   
7. The system creates a new product record in the database.   
8. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. The system detects invalid or missing product information in step 6.   
 a. The system displays an error message and highlights the missing or incorrect fields.   
 b. The administrator revises the product details and resubmits the form.   
 c. The system revalidates the data.   
  
2. The system detects that the product name already exists in the catalog in step 6.   
 a. The system displays an error message indicating the product name is not unique.   
 b. The administrator modifies the product name and resubmits the form.   
 c. The system revalidates the product name.   
  
3. The system detects an issue with the uploaded documentation in step 4.   
 a. The system displays an error message about the file type or size.   
 b. The administrator uploads valid documentation and resubmits the form.   
 c. The system accepts the new file and proceeds.  
  
Use Case Name: Manage Product   
Use Case ID: UC-05   
Actors: Administrator, System   
Preconditions:   
- The system is operational.   
- The administrator is logged into their account.   
- The administrator has access to the product management interface.   
- The product to be managed already exists in the system's database (for updates or deletions).   
  
Postconditions:   
- The product information is updated or removed from the system's database.   
- The product catalog is refreshed to reflect the changes.   
- A confirmation message is displayed to the administrator.   
  
Main Flow:   
1. The administrator navigates to the product management section.   
2. The system displays a list of existing products.   
3. The administrator selects a product to edit or delete.   
4. The system opens the product details form.   
5. The administrator modifies the product information (e.g., name, description, price, quantity, category, or specifications).   
6. The administrator clicks the "Save Changes" or "Delete Product" button.   
7. The system validates the updated information (e.g., correct price format, available category, valid specifications).   
8. The system updates or deletes the product record in the database.   
9. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. The system detects invalid or missing product information in step 7.   
 a. The system displays an error message and highlights the incorrect or missing fields.   
 b. The administrator corrects the information and resubmits the changes.   
 c. The system revalidates the updated data.   
  
2. The system detects that the product is associated with active orders in step 6.   
 a. The system displays a warning message that the product cannot be deleted due to active orders.   
 b. The administrator chooses to update the product instead of deleting it.   
 c. The system proceeds with the update process.   
  
3. The system detects that the product name is not unique after modification in step 7.   
 a. The system displays an error message indicating the product name already exists.   
 b. The administrator modifies the product name to make it unique.   
 c. The system revalidates the product name and saves the changes if valid.  
  
Use Case Name: View Product Details   
Use Case ID: UC-06   
Actors: Customer, System   
Preconditions:   
- The system is operational.   
- The customer is logged into their account or is a guest user.   
- The product exists in the system's database.   
  
Postconditions:   
- The customer views the detailed information of the selected product.   
- The product details are displayed in a user-friendly format.   
- The customer may add the product to their cart or continue browsing.   
  
Main Flow:   
1. The customer browses the product catalog or searches for a product.   
2. The customer clicks on a product to view its details.   
3. The system retrieves the product information from the database.   
4. The system displays the product details, including description, price, specifications, and images.   
5. The customer may review the product or add it to their cart.   
  
Alternative Flow:   
1. The system detects that the product no longer exists in the database in step 3.   
 a. The system displays an error message indicating the product is not available.   
 b. The customer is redirected to the product catalog or a relevant category.   
  
2. The system detects an issue loading product images or documentation in step 4.   
 a. The system displays a message indicating that some media could not be loaded.   
 b. The customer continues reviewing the available product details.   
  
3. The customer adds the product to their cart in step 5.   
 a. The system updates the cart with the new item.   
 b. The system displays a confirmation message and updates the cart summary.  
  
Use Case Name: Add to Cart   
Use Case ID: UC-07   
Actors: Customer, System   
Preconditions:   
- The system is operational.   
- The customer is logged into their account or is a guest user.   
- The product to be added exists in the product catalog.   
- The product has available inventory.   
  
Postconditions:   
- The selected product is added to the customer's shopping cart.   
- The cart summary is updated to reflect the new item and total cost.   
- The customer receives a confirmation message of the addition.   
  
Main Flow:   
1. The customer browses the product catalog or searches for a product.   
2. The customer clicks on the "Add to Cart" button for the desired product.   
3. The system checks the product's availability and inventory.   
4. The system adds the product as a cart item with the specified quantity.   
5. The system updates the cart summary, including the total cost and item count.   
6. The system displays a confirmation message that the product was successfully added to the cart.   
  
Alternative Flow:   
1. The system detects insufficient inventory in step 3.   
 a. The system displays an error message indicating that the product is out of stock or the requested quantity is not available.   
 b. The customer adjusts the quantity or selects another product.   
  
2. The system detects that the product does not exist in the catalog in step 3.   
 a. The system displays an error message indicating the product is not available.   
 b. The customer is redirected to the product catalog or a relevant category.   
  
3. The customer adds a product that is already in the cart in step 2.   
 a. The system increases the quantity of the existing cart item.   
 b. The system updates the cart summary accordingly.  
  
Use Case Name: Modify Cart Item   
Use Case ID: UC-08   
Actors: Customer, System   
Preconditions:   
- The system is operational.   
- The customer is logged into their account or is a guest user.   
- The customer has at least one item in their shopping cart.   
- The cart item to be modified exists in the system.   
  
Postconditions:   
- The selected cart item is updated with the new quantity or removed.   
- The cart summary is recalculated to reflect the changes in total cost and item count.   
- The customer receives a confirmation message of the modification.   
  
Main Flow:   
1. The customer navigates to the shopping cart page.   
2. The system displays the cart items with their details (e.g., product name, quantity, price).   
3. The customer selects a cart item to modify.   
4. The customer updates the quantity or removes the item.   
5. The system validates the new quantity against the available inventory.   
6. The system updates the cart item in the database.   
7. The system recalculates the total cost and updates the cart summary.   
8. The system displays a confirmation message of the modification.   
  
Alternative Flow:   
1. The system detects that the requested quantity exceeds available inventory in step 5.   
 a. The system displays an error message indicating the maximum available quantity.   
 b. The customer adjusts the quantity to an available amount.   
 c. The system validates the new quantity and updates the cart item.   
  
2. The system detects an invalid quantity (e.g., negative or non-numeric) in step 4.   
 a. The system displays an error message and prompts the customer to enter a valid quantity.   
 b. The customer revises the quantity and resubmits the change.   
 c. The system revalidates and updates the cart item if valid.   
  
3. The customer removes the cart item in step 4.   
 a. The system deletes the cart item from the database.   
 b. The system recalculates the total cost and updates the cart summary.   
 c. The system displays a confirmation message of the item removal.  
  
Use Case Name: Remove Cart Item   
Use Case ID: UC-09   
Actors: Customer, System   
Preconditions:   
- The system is operational.   
- The customer is logged into their account or is a guest user.   
- The customer has at least one item in their shopping cart.   
- The cart item to be removed exists in the system.   
  
Postconditions:   
- The selected cart item is removed from the customer's cart.   
- The cart summary is updated to reflect the changes in total cost and item count.   
- The customer receives a confirmation message of the item removal.   
  
Main Flow:   
1. The customer navigates to the shopping cart page.   
2. The system displays the cart items with their details (e.g., product name, quantity, price).   
3. The customer selects a cart item to remove.   
4. The system verifies the cart item's existence in the database.   
5. The system deletes the selected cart item from the customer's cart.   
6. The system recalculates the total cost and updates the cart summary.   
7. The system displays a confirmation message that the item was successfully removed.   
  
Alternative Flow:   
1. The system detects that the selected cart item does not exist in the cart in step 3.   
 a. The system displays an error message indicating the item is not in the cart.   
 b. The customer selects a valid item to remove.   
 c. The system proceeds with the removal process.   
  
2. The system detects an invalid session in step 4.   
 a. The system displays an error message and prompts the customer to log in again.   
 b. The customer logs in.   
 c. The system resumes the removal process.   
  
3. The system detects that the cart is empty after removal in step 5.   
 a. The system displays a message indicating the cart is now empty.   
 b. The customer is redirected to the homepage or product catalog.  
  
Use Case Name: Checkout Order   
Use Case ID: UC-10   
Actors: Customer, System, Payment Gateway, Inventory System   
Preconditions:   
- The system is operational.   
- The customer is logged into their account or is a guest user with a valid cart.   
- The customer has at least one item in their shopping cart.   
- The customer has selected a valid payment method and shipping address.   
- The inventory system is accessible and has sufficient stock for the items in the cart.   
  
Postconditions:   
- The order is created and stored in the system.   
- The payment is processed through the selected payment gateway.   
- The inventory is updated to reflect the deducted stock.   
- The customer receives an order confirmation email.   
- The cart is cleared after successful checkout.   
  
Main Flow:   
1. The customer navigates to the checkout page.   
2. The system displays the order summary, including items, quantities, and total cost.   
3. The customer reviews and confirms the shipping address.   
4. The customer selects a payment method (e.g., credit card, PayPal).   
5. The customer clicks the "Place Order" button.   
6. The system validates the cart items and their availability in the inventory.   
7. The system processes the payment through the selected payment gateway.   
8. The payment gateway confirms successful payment.   
9. The system creates a new order record in the database and associates it with the customer account.   
10. The system updates the inventory to reduce the stock of the ordered products.   
11. The system sends an order confirmation email to the customer.   
12. The system clears the customer’s cart.   
13. The system displays a confirmation message that the order was successfully placed.   
  
Alternative Flow:   
1. The system detects that an item in the cart is out of stock in step 6.   
 a. The system displays an error message and highlights the out-of-stock item.   
 b. The customer removes or replaces the out-of-stock item.   
 c. The system revalidates the cart and proceeds with checkout if valid.   
  
2. The system detects a failed payment in step 7.   
 a. The system displays an error message indicating the payment failed.   
 b. The customer selects an alternative payment method or retries the same method.   
 c. The system reprocesses the payment and proceeds if successful.   
  
3. The system detects an invalid or incomplete shipping address in step 3.   
 a. The system prompts the customer to correct or complete the shipping address.   
 b. The customer revises the address and resubmits it.   
 c. The system revalidates the shipping address and proceeds with checkout.   
  
4. The system detects an invalid session in step 5.   
 a. The system displays an error message and prompts the customer to log in again.   
 b. The customer logs in.   
 c. The system resumes the checkout process.  
  
Use Case Name: Confirm Order   
Use Case ID: UC-11   
Actors: Customer, System, Payment Gateway, Inventory System   
Preconditions:   
- The system is operational.   
- The customer has completed the checkout process (UC-10).   
- The customer has selected a valid payment method and shipping address.   
- The cart is not empty.   
- The inventory system is accessible and has sufficient stock for the items in the cart.   
  
Postconditions:   
- The order is confirmed and finalized in the system.   
- The payment is successfully processed.   
- The inventory is updated to reflect the deducted stock.   
- The order confirmation email is sent to the customer.   
- The cart is cleared.   
  
Main Flow:   
1. The customer reviews the order summary on the checkout page.   
2. The customer clicks the "Confirm Order" button.   
3. The system processes the payment via the selected payment gateway.   
4. The payment gateway returns a success status.   
5. The system updates the inventory to deduct the ordered product quantities.   
6. The system creates a new order record in the database and associates it with the customer.   
7. The system sends an order confirmation email to the customer.   
8. The system clears the customer’s cart.   
9. The system displays a confirmation message that the order was successfully placed.   
  
Alternative Flow:   
1. The system detects a failed payment in step 3.   
 a. The system displays an error message indicating the payment failed.   
 b. The customer selects an alternative payment method or retries the same method.   
 c. The system reprocesses the payment and proceeds if successful.   
  
2. The system detects that the inventory is insufficient for the order in step 5.   
 a. The system displays an error message and highlights the product with insufficient stock.   
 b. The customer adjusts the order (e.g., reduces quantity or removes item).   
 c. The system revalidates inventory and confirms the updated order if valid.   
  
3. The system detects an issue sending the confirmation email in step 7.   
 a. The system logs the error and displays a message that the email could not be sent.   
 b. The customer is still redirected to the order confirmation page.   
 c. The system retries sending the email in the background.  
  
Use Case Name: Send Order Confirmation Email   
Use Case ID: UC-12   
Actors: System, Customer   
Preconditions:   
- The system is operational.   
- An order has been successfully placed and processed (UC-10 or UC-11).   
- The customer's email address is valid and verified.   
- The system has access to the email service.   
  
Postconditions:   
- The order confirmation email is sent to the customer.   
- The customer receives a summary of the order, including items, total cost, and shipping details.   
- The system logs the email delivery status.   
  
Main Flow:   
1. The system generates an order confirmation email after the order is successfully placed.   
2. The system populates the email with the customer's details, order information, and transaction summary.   
3. The system sends the confirmation email to the customer's registered email address.   
4. The system logs the email as sent and updates the order status accordingly.   
5. The customer receives the email and reviews the order confirmation.   
  
Alternative Flow:   
1. The system detects a failure to send the email in step 3.   
 a. The system logs the error and displays a message that the confirmation email could not be sent.   
 b. The system retries sending the email in the background.   
  
2. The system detects an invalid or unverified email address for the customer in step 1.   
 a. The system displays an error message and prompts the customer to update their email address.   
 b. The customer provides a valid email address.   
 c. The system resends the confirmation email to the updated address.   
  
3. The customer does not receive the email within a reasonable time.   
 a. The customer contacts support or checks the spam/junk folder.   
 b. The system allows the customer to manually request a re-sent confirmation email.   
 c. The system sends the confirmation email again.  
  
Use Case Name: Manage Inventory   
Use Case ID: UC-13   
Actors: Administrator, System, Inventory System   
Preconditions:   
- The system is operational.   
- The administrator is logged into their account.   
- The administrator has access to the inventory management interface.   
- The inventory system is accessible and functioning.   
  
Postconditions:   
- The inventory levels for products are updated in the system's database.   
- The inventory status is reflected accurately in the product catalog.   
- A confirmation message is displayed to the administrator.   
  
Main Flow:   
1. The administrator navigates to the inventory management section.   
2. The system displays a list of products with their current inventory levels.   
3. The administrator selects a product to update its inventory.   
4. The administrator enters the new inventory quantity.   
5. The administrator clicks the "Update Inventory" button.   
6. The system validates the entered quantity (e.g., numeric, non-negative).   
7. The system updates the inventory record in the database.   
8. The system refreshes the product catalog to reflect the new inventory status.   
9. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. The system detects an invalid or negative inventory quantity in step 6.   
 a. The system displays an error message and prompts the administrator to enter a valid quantity.   
 b. The administrator revises the quantity and resubmits the update.   
 c. The system revalidates and proceeds if valid.   
  
2. The system detects a connection issue with the inventory system in step 7.   
 a. The system displays an error message and logs the issue.   
 b. The administrator retries the update after resolving the issue.   
 c. The system reattempts the inventory update.   
  
3. The system detects that the selected product does not exist in the database in step 3.   
 a. The system displays an error message indicating the product is not found.   
 b. The administrator selects a valid product.   
 c. The system proceeds with the inventory update.  
  
Use Case Name: Categorize Products   
Use Case ID: UC-14   
Actors: Administrator, System   
Preconditions:   
- The system is operational.   
- The administrator is logged into their account.   
- The administrator has access to the product categorization interface.   
- The product to be categorized exists in the system's database.   
- The category to be assigned exists in the system's database.   
  
Postconditions:   
- The product is assigned to the correct category in the system.   
- The product catalog is updated to reflect the new categorization.   
- A confirmation message is shown to the administrator.   
  
Main Flow:   
1. The administrator navigates to the product categorization section.   
2. The system displays a list of products and their current categories.   
3. The administrator selects a product to categorize or re-categorize.   
4. The system opens the categorization form for the selected product.   
5. The administrator selects or enters the appropriate category.   
6. The administrator clicks the "Save Category" button.   
7. The system validates the selected category (e.g., existence, correctness).   
8. The system updates the product’s category in the database.   
9. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. The system detects that the selected category does not exist in step 7.   
 a. The system displays an error message and prompts the administrator to select a valid category.   
 b. The administrator selects a valid category and resubmits.   
 c. The system validates and updates the product's category.   
  
2. The system detects an invalid session in step 3.   
 a. The system displays an error message and prompts the administrator to log in again.   
 b. The administrator logs in.   
 c. The system resumes the categorization process.   
  
3. The system detects that the product is already in the selected category in step 7.   
 a. The system displays a message indicating that the product is already categorized correctly.   
 b. The administrator chooses to close the form or update other details.  
  
Use Case Name: Update Product Information   
Use Case ID: UC-04   
Actors: Administrator, System   
Preconditions:   
- The system is operational.   
- The administrator is logged into their account.   
- The administrator has access to the product management interface.   
- The product to be updated already exists in the system's database.   
  
Postconditions:   
- The product information is updated in the system's database.   
- The updated product details are displayed in the product catalog.   
- A confirmation message is shown to the administrator.   
  
Main Flow:   
1. The administrator navigates to the product management section.   
2. The system displays a list of products with their current details.   
3. The administrator selects a product to update.   
4. The system opens the product details form with the current information pre-filled.   
5. The administrator modifies the product details (e.g., name, description, price, specifications, or category).   
6. The administrator uploads updated product documentation or images if needed.   
7. The administrator clicks the "Update Product" button.   
8. The system validates the updated information (e.g., correct price format, valid specifications).   
9. The system updates the product record in the database.   
10. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. The system detects invalid or missing product information in step 8.   
 a. The system displays an error message and highlights the incorrect or missing fields.   
 b. The administrator revises the product details and resubmits the form.   
 c. The system revalidates the data and proceeds if valid.   
  
2. The system detects that the product name is not unique after modification in step 8.   
 a. The system displays an error message indicating the product name already exists.   
 b. The administrator modifies the product name to make it unique.   
 c. The system revalidates the product name and saves the changes if valid.   
  
3. The system detects an issue with the uploaded documentation in step 6.   
 a. The system displays an error message about the file type or size.   
 b. The administrator uploads valid documentation and resubmits the form.   
 c. The system accepts the new file and proceeds with the update.  
  
Use Case Name: Admin Login   
Use Case ID: UC-01   
Actors: Administrator, System   
Preconditions:   
- The system is operational.   
- The administrator has access to the login interface.   
- The administrator has a valid account in the system.   
  
Postconditions:   
- The administrator is authenticated and logged into their account.   
- The administrator is redirected to the admin dashboard.   
- A login record is added to the system's log.   
  
Main Flow:   
1. The administrator navigates to the admin login page.   
2. The system displays the login form.   
3. The administrator enters their username and password.   
4. The administrator clicks the "Login" button.   
5. The system validates the entered credentials against the database.   
6. The system confirms the administrator's identity.   
7. The system logs the administrator in and redirects them to the admin dashboard.   
  
Alternative Flow:   
1. The system detects invalid username or password in step 5.   
 a. The system displays an error message indicating incorrect login details.   
 b. The administrator revises their username or password and resubmits the form.   
 c. The system revalidates the credentials.   
  
2. The system detects that the administrator account is locked or disabled in step 5.   
 a. The system displays an error message stating that the account is locked or disabled.   
 b. The administrator may request account recovery or contact support.   
  
3. The system detects an invalid session in step 4.   
 a. The system displays an error message and prompts the administrator to log in again.   
 b. The administrator logs in.   
 c. The system resumes the login process.  
  
Use Case Name: Admin Logout   
Use Case ID: UC-04   
Actors: Administrator, System   
Preconditions:   
- The system is operational.   
- The administrator is logged into their account.   
- The administrator has access to the logout functionality.   
  
Postconditions:   
- The administrator is logged out of their account.   
- The admin session is terminated.   
- The administrator is redirected to the admin login page or homepage.   
  
Main Flow:   
1. The administrator navigates to the admin dashboard or user menu.   
2. The system displays the logout option.   
3. The administrator selects the "Logout" button.   
4. The system verifies the admin's session.   
5. The system terminates the active session and clears session data.   
6. The system redirects the administrator to the admin login page or homepage.   
  
Alternative Flow:   
1. The system detects an invalid or expired session in step 4.   
 a. The system displays an error message indicating the session is invalid.   
 b. The administrator is redirected to the login page automatically.  
  
Use Case Name: View Purchase History   
Use Case ID: UC-14   
Actors: Customer, System   
Preconditions:   
- The system is operational.   
- The customer is logged into their account.   
- The customer has at least one completed order in the purchase history.   
- The system has access to the order database.   
  
Postconditions:   
- The customer views their purchase history.   
- The system displays a list of past orders with details (e.g., order date, items, total cost).   
- The customer may select an order for more detailed information.   
  
Main Flow:   
1. The customer navigates to the "Purchase History" section of their account dashboard.   
2. The system retrieves the customer's order history from the database.   
3. The system displays a list of past orders with key details (e.g., order number, date, total cost, status).   
4. The customer selects an order to view more details.   
5. The system retrieves and displays the full details of the selected order (e.g., items, quantities, shipping address, payment method).   
6. The customer reviews the order information.   
  
Alternative Flow:   
1. The system detects that the customer has no purchase history in step 2.   
 a. The system displays a message indicating no past orders found.   
 b. The customer is redirected to the homepage or another relevant section.   
  
2. The system detects an issue retrieving the order details in step 5.   
 a. The system displays an error message and logs the issue.   
 b. The customer is prompted to try again later or contact support.   
  
3. The system detects an invalid session in step 4.   
 a. The system displays an error message and prompts the customer to log in again.   
 b. The customer logs in.   
 c. The system resumes the process of viewing order details.  
  
Use Case Name: Manage Plugin   
Use Case ID: UC-15   
Actors: Administrator, System   
Preconditions:   
- The system is operational.   
- The administrator is logged into their account.   
- The administrator has access to the plugin management interface.   
- The plugin to be managed exists in the system (for updates or deletions).   
  
Postconditions:   
- The plugin is added, updated, or removed from the system.   
- The plugin status is reflected in the plugin management interface.   
- A confirmation message is displayed to the administrator.   
  
Main Flow:   
1. The administrator navigates to the plugin management section.   
2. The system displays a list of available plugins with their current status.   
3. The administrator selects an action (e.g., "Install Plugin," "Update Plugin," "Remove Plugin").   
4. If installing or updating, the administrator uploads the plugin file or selects it from a repository.   
5. The system validates the plugin (e.g., file format, compatibility).   
6. The system processes the plugin action (installation, update, or removal).   
7. The system updates the plugin status in the database.   
8. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. The system detects an invalid or incompatible plugin file in step 5.   
 a. The system displays an error message and highlights the issue.   
 b. The administrator uploads a valid and compatible plugin file.   
 c. The system revalidates and proceeds with the installation or update.   
  
2. The system detects that the plugin is already installed in step 3.   
 a. The system displays a message indicating the plugin is already installed.   
 b. The administrator chooses to update the plugin or cancel the action.   
 c. The system proceeds with the update or exits the process.   
  
3. The system detects an invalid session in step 3.   
 a. The system displays an error message and prompts the administrator to log in again.   
 b. The administrator logs in.   
 c. The system resumes the plugin management process.  
  
Use Case Name: Access Plugin Documentation   
Use Case ID: UC-16   
Actors: Administrator, System   
Preconditions:   
- The system is operational.   
- The administrator is logged into their account.   
- The administrator has access to the plugin documentation interface.   
- The plugin documentation exists in the system.   
  
Postconditions:   
- The administrator views the plugin documentation.   
- The documentation is displayed in a structured and user-friendly format.   
- The administrator may navigate to specific sections of the documentation.   
  
Main Flow:   
1. The administrator navigates to the plugin management section.   
2. The system displays a list of plugins with options to access documentation.   
3. The administrator selects the "View Documentation" option for a plugin.   
4. The system retrieves the plugin documentation from the database or file storage.   
5. The system displays the plugin documentation, including installation instructions, usage examples, and configuration details.   
6. The administrator reviews the documentation and may search or navigate to specific sections.   
  
Alternative Flow:   
1. The system detects that the plugin documentation is missing or cannot be retrieved in step 4.   
 a. The system displays an error message indicating that documentation is unavailable.   
 b. The administrator may request to upload or update the documentation.   
 c. The system proceeds with the upload or update process if initiated.   
  
2. The system detects an invalid session in step 3.   
 a. The system displays an error message and prompts the administrator to log in again.   
 b. The administrator logs in.   
 c. The system resumes the process of accessing the plugin documentation.   
  
3. The system detects an issue with the documentation file (e.g., corrupted or unsupported format) in step 4.   
 a. The system logs the error and displays a message to the administrator.   
 b. The administrator uploads a corrected documentation file.   
 c. The system retrieves and displays the updated documentation.