项目文档

# Functional Requirement

# 1. Functional Requirements  
  
## 1.1 Customer Registration Function   
Function ID: FR-01   
Description: Customers can register a new account by providing their name, email, and password. The system validates the email and password and sends a confirmation email.   
Input: Name, email, password, and optional contact details.   
Output: A successfully registered customer account and a confirmation email sent to the customer.  
  
## 1.2 Customer Login Function   
Function ID: FR-02   
Description: Customers can log in using their registered email and password. The system validates the credentials and logs the login activity.   
Input: Email and password.   
Output: A successful login session and the customer's dashboard or homepage.  
  
## 1.3 Customer Logout Function   
Function ID: FR-03   
Description: Customers can end their current session by logging out. The system clears session data and redirects the customer to the login or homepage.   
Input: Logout request from the customer.   
Output: Session cleared and the customer redirected to the login or homepage.  
  
## 1.4 View Product Information Function   
Function ID: FR-04   
Description: Customers can view detailed information about a product, including name, description, price, and availability. The system logs the request.   
Input: Product ID or search criteria.   
Output: Displayed product details and a logged request.  
  
## 1.5 Search Products Function   
Function ID: FR-05   
Description: Customers can search for products using keywords, categories, or other relevant criteria. The system retrieves matching products and logs the search activity.   
Input: Search query (e.g., name, category, keyword).   
Output: A list of matching products and a logged search activity.  
  
## 1.6 Add Product to Cart Function   
Function ID: FR-06   
Description: Customers can add a product to their cart, specifying the quantity. The system checks stock availability and logs the addition.   
Input: Product ID, quantity, and customer ID.   
Output: Updated cart with the added product and a confirmation message.  
  
## 1.7 Create Order Function   
Function ID: FR-07   
Description: Customers can create an order from their cart. The system validates the order, processes the payment, and logs the order creation.   
Input: Cart contents, shipping address, and selected payment method.   
Output: A new order record, updated product inventory, and an order confirmation email sent to the customer.  
  
## 1.8 Checkout Function   
Function ID: FR-08   
Description: Customers can complete the checkout process by confirming the order and payment. The system processes the payment and creates the order.   
Input: Cart contents, shipping address, and payment information.   
Output: A completed order, updated inventory, and an order confirmation email.  
  
## 1.9 Receive Order Confirmation Email Function   
Function ID: FR-09   
Description: The system sends an order confirmation email to the customer after a successful checkout.   
Input: Order details, including customer name, order ID, and total amount.   
Output: A confirmation email sent to the customer and a logged email activity.  
  
## 1.10 Update Order Status Function   
Function ID: FR-10   
Description: Administrators can update the status of an existing order. The system logs the update and may send a notification to the customer.   
Input: Order ID and new status (e.g., Processing, Shipped, Delivered, Cancelled).   
Output: Updated order status and a logged activity with optional email notification.  
  
## 1.11 View Order History Function   
Function ID: FR-11   
Description: Customers can view their past orders, including order details and status. The system retrieves and displays the order history and logs the access.   
Input: Customer ID.   
Output: A list of past orders with details and a logged access activity.  
  
## 1.12 Delete Order Function   
Function ID: FR-12   
Description: Customers can delete an order if it is in a cancellable status. The system verifies the status, deletes the order, and updates inventory if necessary.   
Input: Order ID and confirmation of deletion.   
Output: Deleted order record, updated inventory, and an optional confirmation email sent to the customer.  
  
## 1.13 Manage Product Inventory Function   
Function ID: FR-13   
Description: Administrators can manage product inventory, including adding, updating, or deleting products. The system logs the changes and updates the product database.   
Input: Product details (e.g., name, price, stock, description) and action (add, update, delete).   
Output: Updated product inventory and a logged activity.  
  
## 1.14 Add New Product Function   
Function ID: FR-14   
Description: Administrators can add new products to the system by providing product details. The system validates the data and updates the product database.   
Input: Product name, description, price, stock, and category.   
Output: A new product record and a confirmation message to the administrator.  
  
## 1.15 Update Product Details Function   
Function ID: FR-15   
Description: Administrators can update product details such as name, price, stock, or category. The system validates the data and updates the product record.   
Input: Product ID and updated product details.   
Output: Updated product record and a confirmation message to the administrator.  
  
## 1.16 Delete Product Function   
Function ID: FR-16   
Description: Administrators can delete a product if it is not part of any active or pending orders. The system verifies the deletion eligibility and logs the action.   
Input: Product ID and confirmation of deletion.   
Output: Deleted product record and a confirmation message to the administrator.  
  
## 1.17 Administrator Registration Function   
Function ID: FR-17   
Description: Administrators can register a new account by providing name, email, password, and role. The system validates the data and sends a confirmation email.   
Input: Name, email, password, and role.   
Output: A new administrator record and a confirmation email sent to the administrator.  
  
## 1.18 Administrator Login Function   
Function ID: FR-18   
Description: Administrators can log in using their registered email and password. The system validates the credentials and logs the login activity.   
Input: Email and password.   
Output: A successful login session and the administrator's dashboard or homepage.  
  
## 1.19 Manage Customer Accounts Function   
Function ID: FR-19   
Description: Administrators can view, update, or delete customer accounts. The system logs all actions and may send notifications to customers.   
Input: Customer ID and action (view, update, delete).   
Output: Updated or deleted customer record and a logged activity with optional email notification.  
  
## 1.20 Manage Orders Function   
Function ID: FR-20   
Description: Administrators can view, update, or delete orders. The system logs the changes and may send notifications to customers.   
Input: Order ID and action (view, update, delete).   
Output: Updated or deleted order record and a logged activity with optional email notification.  
  
## 1.21 View Payment Information Function   
Function ID: FR-21   
Description: Customers can view their payment history, including method, amount, and transaction date. The system retrieves and displays the information and logs the access.   
Input: Customer ID.   
Output: A list of payment records and a logged access activity.  
  
## 1.22 Manage Cart Function   
Function ID: FR-22   
Description: Customers can manage their shopping cart by adding, updating, or removing items. The system logs the cart modifications and updates the total.   
Input: Product ID, quantity, and action (add, update, remove).   
Output: Updated cart contents and a confirmation message.  
  
## 1.23 Manage Order Status Function   
Function ID: FR-23   
Description: Administrators can update the status of an order. The system logs the status change and may send a notification to the customer.   
Input: Order ID and new status.   
Output: Updated order status and a logged activity with optional email notification.  
  
## 1.24 Manage Payment Method Function   
Function ID: FR-24   
Description: Administrators can manage available payment methods, including adding, updating, or deleting a method. The system validates and logs the changes.   
Input: Payment method details (e.g., method name, description, configuration) and action (add, update, delete).   
Output: Updated payment method configuration and a confirmation message.  
  
## 1.25 Manage Email Settings Function   
Function ID: FR-25   
Description: Administrators can manage the email configuration, including server settings and templates. The system validates the input and logs the changes.   
Input: Email server settings (e.g., host, port, username, password) and action (update, edit, enable/disable).   
Output: Updated email configuration and a confirmation message.  
  
## 1.26 Add Plugin Function   
Function ID: FR-26   
Description: Administrators can add new plugins to the system. The system validates the plugin and logs the addition.   
Input: Plugin details (e.g., name, version, description) and plugin file or API configuration.   
Output: A new plugin record and a confirmation message to the administrator.  
  
## 1.27 Update Plugin Function   
Function ID: FR-27   
Description: Administrators can update plugin configurations. The system validates the changes and logs the update.   
Input: Plugin ID and updated configuration details.   
Output: Updated plugin configuration and a confirmation message.  
  
## 1.28 Delete Plugin Function   
Function ID: FR-28   
Description: Administrators can delete a plugin if it is not in use. The system verifies the deletion eligibility and logs the action.   
Input: Plugin ID and confirmation of deletion.   
Output: Deleted plugin record and a confirmation message.  
  
## 1.29 View Plugin Documentation Function   
Function ID: FR-29   
Description: Administrators can view documentation for a specific plugin. The system retrieves the documentation and logs the access.   
Input: Plugin ID.   
Output: Displayed plugin documentation and a logged access activity.  
  
## 1.30 Access API Specifications Function   
Function ID: FR-30   
Description: Administrators can view API specifications for plugins or system components. The system retrieves the documentation and logs the access.   
Input: API ID.   
Output: Displayed API documentation and a logged access activity.  
  
## 1.31 View Order Status Function   
Function ID: FR-31   
Description: Customers can view the current status of their orders. The system retrieves the status and logs the request.   
Input: Order ID.   
Output: Displayed order status and a logged access activity.

# External Description

# 2. External Interfaces   
  
This section describes the external interfaces of the system. These interfaces include user interfaces, hardware interfaces, software interfaces, and communication interfaces. The system interacts with these external interfaces to provide the required functionalities and ensure smooth operation.   
  
## 2.1 User Interface   
  
The system provides several user interfaces for both customers and administrators to interact with its functionalities. These interfaces are designed to be intuitive and user-friendly.   
  
- \*\*Customer Registration Interface\*\*:   
 - Description: A web-based interface where customers can input their name, email, and password to register a new account. Optional contact details can also be added.   
 - Interaction: The system validates the email and password format and sends a confirmation email.   
  
- \*\*Customer Login Interface\*\*:   
 - Description: A web-based interface that allows customers to log in using their registered email and password.   
 - Interaction: The system validates the credentials and logs the login activity.   
  
- \*\*Customer Logout Interface\*\*:   
 - Description: A web-based interface that allows customers to log out, ending their session.   
 - Interaction: The system clears session data and redirects the user to the login or homepage.   
  
- \*\*Product Information Interface\*\*:   
 - Description: A web-based interface displaying product details, such as name, description, price, and availability.   
 - Interaction: The system retrieves and displays product information based on the product ID or search criteria.   
  
- \*\*Product Search Interface\*\*:   
 - Description: A web-based interface where customers can search for products using keywords, categories, or other criteria.   
 - Interaction: The system retrieves matching products and logs the search activity.   
  
- \*\*Shopping Cart Interface\*\*:   
 - Description: A web-based interface where customers can manage their cart, including adding, updating, or removing items.   
 - Interaction: The system updates the cart contents and logs the modification.   
  
- \*\*Checkout Interface\*\*:   
 - Description: A web-based interface for customers to confirm their order and payment information.   
 - Interaction: The system processes the payment and creates the order.   
  
- \*\*Order History Interface\*\*:   
 - Description: A web-based interface where customers can view their past orders, including order status and details.   
 - Interaction: The system retrieves order history based on the customer ID and logs the access.   
  
- \*\*Payment Information Interface\*\*:   
 - Description: A web-based interface where customers can view their payment history, including method, amount, and transaction date.   
 - Interaction: The system retrieves and displays payment records and logs the access.   
  
- \*\*Order Status Interface\*\*:   
 - Description: A web-based interface for customers to view the current status of a specific order.   
 - Interaction: The system retrieves the order status based on the order ID and logs the request.   
  
- \*\*Administrator Dashboard Interface\*\*:   
 - Description: A web-based interface for administrators to manage products, orders, customers, plugins, and system settings.   
 - Interaction: The system retrieves and updates data based on administrative actions and logs these activities.   
  
- \*\*Plugin Management Interface\*\*:   
 - Description: A web-based interface where administrators can add, update, or delete plugins.   
 - Interaction: The system validates the plugin and logs the addition, update, or deletion.   
  
- \*\*API Specifications Interface\*\*:   
 - Description: A web-based interface where administrators can view API specifications for system components or plugins.   
 - Interaction: The system retrieves and displays the relevant API documentation and logs the access.   
  
## 2.2 Hardware Interface   
  
The system does not require direct interaction with any hardware devices. All functionalities are implemented in a software environment.   
  
## 2.3 Software Interface   
  
The system interacts with various software components and external systems to perform its functions.   
  
- \*\*Product Database\*\*:   
 - Description: A database storing product information, including product ID, name, description, price, stock, and category.   
 - Interaction: The system retrieves, adds, updates, and deletes product records as required by the functional requirements.   
 - Input/Output:   
 - Input: Product details (e.g., name, description, price, stock, category) and action (add, update, delete).   
 - Output: Updated or deleted product records.   
  
- \*\*Order Database\*\*:   
 - Description: A database storing order information, including order ID, customer ID, product details, shipping address, order status, and payment information.   
 - Interaction: The system retrieves, updates, and deletes order records as required by the functional requirements.   
 - Input/Output:   
 - Input: Order details (e.g., customer ID, product ID, quantity, shipping address, status).   
 - Output: Updated or deleted order records.   
  
- \*\*Customer Database\*\*:   
 - Description: A database storing customer information, including customer ID, name, email, password, and optional contact details.   
 - Interaction: The system retrieves, updates, and deletes customer records as required by the functional requirements.   
 - Input/Output:   
 - Input: Customer details (e.g., name, email, password, contact details) and action (add, update, delete).   
 - Output: Updated or deleted customer records.   
  
- \*\*Payment System\*\*:   
 - Description: A third-party software system for processing payments. The system interacts with this component to validate and process customer payments.   
 - Interaction: The system sends payment information (e.g., cart contents, payment method, amount) to the payment system and receives confirmation of payment success or failure.   
 - Input/Output:   
 - Input: Payment information (e.g., customer ID, payment method, total amount).   
 - Output: Payment confirmation or error message.   
  
- \*\*Email Server\*\*:   
 - Description: A software system responsible for sending and receiving email notifications. The system interacts with this component to send confirmation emails to customers and administrators.   
 - Interaction: The system sends order confirmation emails, account confirmation emails, and other notifications to the email server, which then delivers the emails to the appropriate recipients.   
 - Input/Output:   
 - Input: Email content (e.g., recipient email address, message body, subject).   
 - Output: Confirmation that the email was sent successfully.   
  
- \*\*Plugin Management System\*\*:   
 - Description: A software component that allows administrators to manage plugins by adding, updating, or deleting them.   
 - Interaction: The system validates the plugin details (e.g., name, version, description) and configuration, and logs the changes.   
 - Input/Output:   
 - Input: Plugin details (e.g., name, version, description) and action (add, update, delete).   
 - Output: Updated or deleted plugin records.   
  
- \*\*Authentication System\*\*:   
 - Description: A software system that handles user authentication for both customers and administrators.   
 - Interaction: The system validates user credentials (e.g., email and password) against the authentication system and logs login activity.   
 - Input/Output:   
 - Input: Email and password.   
 - Output: Authentication result (success or failure) and session creation.   
  
## 2.4 Communication Interface   
  
The system communicates with external entities through network-based protocols and messaging systems.   
  
- \*\*Email Communication Interface\*\*:   
 - Description: The system uses an email communication interface to send confirmation emails to customers and administrators.   
 - Interaction: The system sends email content (e.g., order confirmation, account confirmation) to the email server via SMTP or another email protocol.   
 - Input/Output:   
 - Input: Order details (e.g., customer name, order ID, total amount) or account details (e.g., email, password).   
 - Output: Confirmation that the email has been sent to the intended recipient.   
  
- \*\*Web Communication Interface\*\*:   
 - Description: The system uses HTTP/HTTPS protocols to communicate with web clients (e.g., browsers) and web services.   
 - Interaction: The system receives user inputs (e.g., search queries, login requests) via HTTP/HTTPS and sends responses (e.g., product lists, dashboard content) in JSON or HTML format.   
 - Input/Output:   
 - Input: Web requests containing user inputs (e.g., search query, login credentials).   
 - Output: Web responses containing system outputs (e.g., product details, order confirmation).   
  
- \*\*API Communication Interface\*\*:   
 - Description: The system provides and consumes APIs to interact with external components (e.g., payment gateways, email servers).   
 - Interaction: The system sends and receives data in JSON or XML format through RESTful APIs.   
 - Input/Output:   
 - Input: API requests containing data (e.g., product details, order status updates).   
 - Output: API responses containing results (e.g., updated inventory, confirmation messages).   
  
- \*\*Logging and Monitoring System\*\*:   
 - Description: The system communicates with a logging and monitoring service to record activity logs, errors, and performance metrics.   
 - Interaction: The system sends logs in a structured format (e.g., JSON) to the logging system for storage and analysis.   
 - Input/Output:   
 - Input: Log data (e.g., action performed, user ID, timestamp).   
 - Output: Log confirmation or error status from the logging system.   
  
## 2.5 Summary of External Interfaces   
  
| Interface Type | Interface Name | Description | Communication Method |   
|----------------|----------------|-------------|----------------------|   
| User Interface | Customer Registration Interface | Allows customers to register a new account | Web-based form submission |   
| User Interface | Customer Login Interface | Allows customers to log in with their email and password | Web-based form submission |   
| User Interface | Shopping Cart Interface | Allows customers to manage their cart | Web-based form submission and updates |   
| Software Interface | Product Database | Stores and retrieves product information | SQL queries or ORM methods |   
| Software Interface | Order Database | Stores and retrieves order information | SQL queries or ORM methods |   
| Software Interface | Payment System | Processes customer payments | REST API or direct integration |   
| Communication Interface | Email Server | Sends and receives confirmation emails | SMTP or other email protocols |   
| Communication Interface | Web Communication | Handles user requests via web browsers | HTTP/HTTPS protocols |   
| Communication Interface | Logging and Monitoring System | Logs system activities for debugging and auditing | REST API or message queues |   
  
This section ensures that all external data sources and interfaces referenced in the functional requirements are clearly defined and categorized, providing developers and stakeholders with a comprehensive understanding of the system's external interactions.

# Use Case

Use Case Name: Register as Customer   
Use Case ID: UC-01   
Actors: Customer, System   
Preconditions:   
1. The customer is not yet registered in the system.   
2. The system is operational and accessible.   
  
Postconditions:   
1. The customer is successfully registered in the system.   
2. A confirmation email is sent to the customer's email address.   
  
Main Flow:   
1. The customer accesses the registration page of the system.   
2. The customer fills in the required information (e.g., name, email, password, contact details).   
3. The customer submits the registration form.   
4. The system validates the input data (e.g., checks for valid email format, password strength, uniqueness of email).   
5. The system creates a new customer record in the database.   
6. The system triggers an email plugin to send a confirmation email to the customer.   
7. The system displays a success message to the customer.   
  
Alternative Flow:   
1. If the system detects invalid or incomplete data during validation (e.g., missing fields, invalid email), it displays an error message and prompts the customer to correct the information.   
2. If the system fails to send the confirmation email, it logs the error and displays a message to the customer, suggesting to check their email or contact the administrator.   
3. If the email is already registered, the system displays an error message indicating that the email is already in use and prompts the customer to use a different one.  
  
Use Case Name: Login as Customer   
Use Case ID: UC-02   
Actors: Customer, System, Email Plugin, Administrator   
  
Preconditions:   
1. The customer is already registered in the system.   
2. The system is operational and accessible.   
3. The customer has a valid email and password.   
  
Postconditions:   
1. The customer is successfully logged into the system.   
2. The system displays the customer’s dashboard or homepage.   
3. The system logs the login activity for security and audit purposes.   
  
Main Flow:   
1. The customer accesses the login page of the system.   
2. The customer enters their email and password.   
3. The customer clicks the "Login" button.   
4. The system validates the email and password against the database.   
5. The system confirms the login credentials are correct.   
6. The system logs the successful login attempt.   
7. The system redirects the customer to their dashboard or homepage.   
  
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 system detects multiple failed login attempts, it locks the account temporarily and notifies the administrator.   
3. If the system fails to log the login activity, it logs the error internally and continues with the user login process.   
4. If the customer clicks "Forgot Password," the system triggers the email plugin to send a password reset link to the customer's registered email.  
  
Use Case Name: Logout as Customer   
Use Case ID: UC-03   
Actors: Customer, System   
  
Preconditions:   
1. The customer is already logged into the system.   
2. The system is operational and accessible.   
  
Postconditions:   
1. The customer is successfully logged out of the system.   
2. The session data associated with the customer is cleared.   
3. The system redirects the customer to the login or homepage.   
  
Main Flow:   
1. The customer navigates to the account settings or profile section.   
2. The customer clicks the "Logout" button.   
3. The system verifies the customer's session and initiates the logout process.   
4. The system clears the session data and ends the current session.   
5. The system redirects the customer to the login or homepage.   
6. The system displays a confirmation message that the customer has been logged out.   
  
Alternative Flow:   
1. If the system fails to clear the session data, it logs the error and displays a message to the customer, suggesting to try again or contact the administrator.   
2. If the customer closes the browser or navigates away before the logout process is completed, the system automatically clears the session on the next access.   
3. If the customer tries to access restricted pages after logging out, the system redirects them to the login page and displays an access denied message.  
  
Use Case Name: View Product Information   
Use Case ID: UC-04   
Actors: Customer, System, Product   
  
Preconditions:   
1. The customer is logged into the system.   
2. The system is operational and accessible.   
3. The product information is available in the database.   
  
Postconditions:   
1. The customer successfully views the product details.   
2. The system displays the relevant product information.   
3. The system logs the customer's request to view the product information.   
  
Main Flow:   
1. The customer navigates to the product listing or search page.   
2. The customer selects a specific product to view.   
3. The system retrieves the product information from the database.   
4. The system displays the product details, including name, description, price, and availability.   
5. The system logs the customer's request for viewing the product.   
  
Alternative Flow:   
1. If the system cannot find the requested product, it displays an error message and suggests searching for a similar product.   
2. If the system fails to retrieve the product information, it logs the error and displays a message to the customer, suggesting to try again later or contact the administrator.   
3. If the product is out of stock, the system displays a message indicating the product is unavailable and may suggest alternatives.  
  
Use Case Name: Search Products   
Use Case ID: UC-05   
Actors: Customer, System, Product, API, Documentation   
  
Preconditions:   
1. The customer is logged into the system.   
2. The system is operational and accessible.   
3. The product database or external API is available.   
4. The customer has navigated to the product search page.   
  
Postconditions:   
1. The customer receives a list of products matching their search criteria.   
2. The system displays the relevant search results.   
3. The system logs the search activity for analytics purposes.   
  
Main Flow:   
1. The customer enters a search query (e.g., product name, category, or keyword) in the search box.   
2. The customer clicks the "Search" button or presses Enter.   
3. The system processes the query and searches the product database or interacts with an external API.   
4. The system retrieves the relevant product data.   
5. The system displays the search results in a user-friendly format.   
6. The system logs the search request for future reference or analysis.   
  
Alternative Flow:   
1. If no products match the search query, the system displays a message indicating that no results were found and suggests refining the search.   
2. If the system cannot access the product database or external API, it displays an error message and suggests trying again later or checking the Documentation for troubleshooting steps.   
3. If the search query is invalid or too broad, the system prompts the customer to provide more specific search terms.  
  
Use Case Name: Add Product to Cart   
Use Case ID: UC-06   
Actors: Customer, System, Product, Order   
  
Preconditions:   
1. The customer is logged into the system.   
2. The system is operational and accessible.   
3. The product the customer wants to add is available in the database.   
4. The customer has navigated to the product details page.   
  
Postconditions:   
1. The product is successfully added to the customer's cart.   
2. The system updates the cart information and displays a confirmation message.   
3. The system logs the addition of the product to the cart.   
  
Main Flow:   
1. The customer views the product details.   
2. The customer clicks the "Add to Cart" button.   
3. The system verifies the product availability and validity.   
4. The system updates the customer's cart with the selected product and quantity.   
5. The system displays a confirmation message that the product has been added.   
6. The system logs the cart update for tracking purposes.   
  
Alternative Flow:   
1. If the product is out of stock, the system displays an error message and suggests alternative products or checking back later.   
2. If the system fails to update the cart, it logs the error and displays a message to the customer, suggesting to try again or contact the administrator.   
3. If the customer adds a product with an invalid quantity (e.g., negative or exceeding stock), the system displays an error and prompts for a valid quantity.  
  
Use Case Name: Create Order   
Use Case ID: UC-07   
Actors: Customer, System, Product, Order, Payment, Email Plugin   
  
Preconditions:   
1. The customer is logged into the system.   
2. The customer has at least one product in their cart.   
3. The system is operational and accessible.   
4. The product(s) in the cart are available in the database.   
5. The payment gateway is accessible and functional.   
  
Postconditions:   
1. A new order is successfully created in the system.   
2. The payment is processed for the order.   
3. The customer receives a confirmation email for the order.   
4. The system updates the product stock accordingly.   
5. The system logs the order creation activity.   
  
Main Flow:   
1. The customer navigates to the shopping cart.   
2. The customer reviews the items in the cart and proceeds to checkout.   
3. The system displays the order summary, including total price and product details.   
4. The customer selects a payment method and enters payment information.   
5. The system validates the payment information and processes the payment.   
6. The system creates a new order record in the database with the customer and product details.   
7. The system updates the stock of the purchased products.   
8. The system triggers the email plugin to send a confirmation email to the customer.   
9. The system displays a success message to the customer.   
  
Alternative Flow:   
1. If the system detects invalid payment information, it displays an error message and prompts the customer to correct the details.   
2. If the payment fails, the system cancels the order creation and displays an error message, suggesting to try again or contact the administrator.   
3. If the product stock is insufficient at the time of order creation, the system displays an error message and suggests removing or adjusting the quantity of the affected product.   
4. If the system fails to send the confirmation email, it logs the error and displays a message to the customer, suggesting to check their email or contact the administrator.   
5. If the customer cancels the checkout process, the system returns them to the cart without creating an order.  
  
Use Case Name: Checkout   
Use Case ID: UC-08   
Actors: Customer, System, Product, Order, Payment, Email Plugin, Administrator   
  
Preconditions:   
1. The customer is logged into the system.   
2. The customer has at least one product in their cart.   
3. The system is operational and accessible.   
4. The product(s) in the cart are available in the database.   
5. The payment gateway is accessible and functional.   
  
Postconditions:   
1. The customer successfully completes the checkout process.   
2. An order is created and stored in the system.   
3. Payment is processed and confirmed.   
4. The customer receives an order confirmation email.   
5. The product stock is updated accordingly.   
6. The system logs the checkout and order creation activity.   
  
Main Flow:   
1. The customer navigates to the shopping cart.   
2. The customer clicks the "Proceed to Checkout" button.   
3. The system displays the checkout page with order summary, shipping details, and payment options.   
4. The customer fills in or confirms their shipping address.   
5. The customer selects a payment method and enters the required payment information.   
6. The system validates the shipping and payment information.   
7. The system processes the payment through the payment gateway.   
8. The system confirms the payment success and creates the order.   
9. The system updates the stock levels of the ordered products.   
10. The system triggers the email plugin to send an order confirmation email.   
11. The system displays a success message to the customer and provides an order reference number.   
  
Alternative Flow:   
1. If the customer's shipping address is invalid or incomplete, the system displays an error and prompts the customer to correct it.   
2. If the selected payment method is unavailable or the payment gateway is down, the system displays an error and suggests trying another method or contacting the administrator.   
3. If the payment is declined, the system cancels the order and displays an error message, prompting the customer to try again or check their payment details.   
4. If the product stock is insufficient during the final checkout step, the system displays an error and suggests adjusting the order.   
5. If the system fails to send the confirmation email, it logs the error and displays a message to the customer, suggesting they check their inbox or contact the administrator.   
6. If the customer decides to cancel the checkout, the system returns them to the cart without processing the order.  
  
Use Case Name: Receive Order Confirmation Email   
Use Case ID: UC-09   
Actors: Customer, System, Email Plugin, Order   
  
Preconditions:   
1. The customer has completed the checkout process successfully.   
2. The system is operational and accessible.   
3. The email plugin is available and functional.   
4. The customer has provided a valid email address during registration or checkout.   
  
Postconditions:   
1. The customer receives an order confirmation email.   
2. The system logs the email sending activity.   
3. The email contains the order details and a reference number.   
  
Main Flow:   
1. The system completes the order creation process.   
2. The system prepares the order confirmation message with the customer's name, order details, and total amount.   
3. The system triggers the email plugin to send the confirmation email to the customer's registered email address.   
4. The email plugin sends the confirmation email to the customer.   
5. The system logs the successful email delivery.   
  
Alternative Flow:   
1. If the email plugin is not available or fails to send the email, the system logs the error and displays a message to the customer, suggesting they check their email or contact the administrator.   
2. If the customer's email address is invalid or missing, the system displays an error message and prompts the customer to provide a valid email address.   
3. If the system fails to retrieve the order details for the email, it logs the error and notifies the administrator for further investigation.  
  
Use Case Name: Update Order Status   
Use Case ID: UC-10   
Actors: Administrator, System, Order   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The system is operational and accessible.   
3. There is an existing order in the system that needs status updates.   
  
Postconditions:   
1. The order status is successfully updated in the system.   
2. The system logs the status change and the administrator who performed it.   
3. If configured, the customer receives a notification about the updated status.   
  
Main Flow:   
1. The administrator navigates to the order management section of the system.   
2. The administrator selects a specific order to update.   
3. The system displays the current order details, including status.   
4. The administrator chooses a new status for the order (e.g., Processing, Shipped, Delivered, Cancelled).   
5. The administrator confirms the status change.   
6. The system updates the order status in the database.   
7. The system logs the change and the administrator responsible.   
8. If applicable, the system sends a notification to the customer via the email plugin.   
  
Alternative Flow:   
1. If the selected status is invalid or not applicable for the current order, the system displays an error message and prompts the administrator to select a valid status.   
2. If the system fails to update the order status, it logs the error and displays a message to the administrator, suggesting to try again or contact the administrator for support.   
3. If the system fails to send the notification to the customer, it logs the error and displays a message to the administrator, suggesting to verify the customer's email or retry the notification.  
  
Use Case Name: View Order History   
Use Case ID: UC-11   
Actors: Customer, System, Order, Email Plugin, Administrator   
  
Preconditions:   
1. The customer is logged into the system.   
2. The system is operational and accessible.   
3. The customer has at least one order in the system.   
  
Postconditions:   
1. The customer successfully views their order history.   
2. The system displays a list of previous orders with details.   
3. The system logs the customer's request to view order history.   
  
Main Flow:   
1. The customer navigates to the "Order History" section in their account.   
2. The system retrieves all past orders associated with the customer from the database.   
3. The system displays the order history in a list, including order date, total amount, status, and order reference number.   
4. The customer can click on a specific order to view detailed information.   
5. The system logs the customer's access to order history for audit purposes.   
  
Alternative Flow:   
1. If the customer has no past orders, the system displays a message indicating that there is no order history available.   
2. If the system fails to retrieve the order data, it logs the error and displays a message to the customer, suggesting they try again later or contact the administrator.   
3. If the customer's session expires while viewing order history, the system redirects them to the login page and displays an access denied message.  
  
Use Case Name: Delete Order   
Use Case ID: UC-12   
Actors: Customer, System, Order, Email Plugin, Administrator   
  
Preconditions:   
1. The customer is logged into the system.   
2. The system is operational and accessible.   
3. The customer has at least one existing order.   
4. The order to be deleted is in a cancellable status (e.g., Pending, Created).   
  
Postconditions:   
1. The specified order is successfully deleted from the system.   
2. The system logs the deletion action and the customer involved.   
3. If configured, the customer receives a confirmation email about the deletion.   
4. The product stock levels are adjusted accordingly if the deletion affects inventory.   
  
Main Flow:   
1. The customer navigates to the "Order History" section in their account.   
2. The customer selects the order they wish to delete.   
3. The system verifies the order status and confirms it is eligible for deletion.   
4. The system displays a confirmation dialog to the customer.   
5. The customer confirms the deletion.   
6. The system deletes the order from the database.   
7. The system updates the product stock if necessary.   
8. The system triggers the email plugin to send a deletion confirmation email to the customer.   
9. The system displays a success message to the customer.   
  
Alternative Flow:   
1. If the order is not in a cancellable status (e.g., Shipped, Delivered), the system displays an error message and prevents deletion.   
2. If the system fails to delete the order, it logs the error and displays a message to the customer, suggesting to try again or contact the administrator.   
3. If the email plugin fails to send the confirmation email, the system logs the error and displays a message to the customer, suggesting they check their email or contact the administrator.   
4. If the customer cancels the deletion during the confirmation step, the system retains the order and returns to the order history view.  
  
Use Case Name: Manage Product Inventory   
Use Case ID: UC-13   
Actors: Administrator, System, Product, Order   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The system is operational and accessible.   
3. The product inventory data is available in the database.   
  
Postconditions:   
1. The product inventory is updated based on the administrator's actions (e.g., adding, removing, or modifying product stock).   
2. The system logs the inventory changes and the administrator who performed them.   
3. The updated inventory is reflected in the product listings and order processing functionalities.   
  
Main Flow:   
1. The administrator navigates to the "Inventory Management" section in the system.   
2. The administrator selects an action: add a new product, update an existing product, or delete a product.   
3. If adding a new product, the administrator enters product details (e.g., name, price, stock, description).   
4. The system validates the input data (e.g., correct price format, valid stock quantity).   
5. The system updates the product inventory in the database.   
6. The system logs the inventory update.   
7. The system displays a confirmation message that the inventory has been updated.   
  
Alternative Flow:   
1. If the system detects invalid or incomplete product data during validation, it displays an error message and prompts the administrator to correct the information.   
2. If the system fails to update the inventory in the database, it logs the error and displays a message to the administrator, suggesting to try again or contact the administrator for support.   
3. If the administrator tries to delete a product that is part of an existing order, the system displays a warning message and prevents deletion to avoid affecting order fulfillment.  
  
Use Case Name: Add New Product   
Use Case ID: UC-14   
Actors: Administrator, System, Product, Plugin, Email Plugin   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The system is operational and accessible.   
3. The product database is available and functional.   
4. The administrator has the necessary permissions to add products.   
  
Postconditions:   
1. A new product is successfully added to the system.   
2. The system updates the product inventory and displays a confirmation message.   
3. The system logs the addition of the product.   
4. If configured, the administrator receives a notification via the email plugin.   
  
Main Flow:   
1. The administrator navigates to the "Product Management" section.   
2. The administrator clicks the "Add New Product" button.   
3. The system displays a product creation form.   
4. The administrator fills in the product details (e.g., name, description, price, stock, category).   
5. The administrator submits the form.   
6. The system validates the input data (e.g., correct price format, valid stock quantity).   
7. The system creates a new product record in the database.   
8. The system updates the product inventory.   
9. The system logs the new product addition.   
10. The system displays a success message to the administrator.   
  
Alternative Flow:   
1. If the system detects invalid or incomplete product data during validation, it displays an error message and prompts the administrator to correct the information.   
2. If the system fails to create the product record in the database, it logs the error and displays a message to the administrator, suggesting to try again or contact the administrator for support.   
3. If the email plugin fails to send the notification, the system logs the error and displays a message to the administrator, suggesting they verify the email settings or retry the action.  
  
Use Case Name: Update Product Details   
Use Case ID: UC-15   
Actors: Administrator, System, Product, Email Plugin   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The system is operational and accessible.   
3. The product to be updated exists in the database.   
4. The administrator has the necessary permissions to modify product details.   
  
Postconditions:   
1. The product details are successfully updated in the system.   
2. The system logs the modification and the administrator who performed it.   
3. If configured, the administrator receives a confirmation email about the update.   
4. The updated product details are reflected in the product listing and customer views.   
  
Main Flow:   
1. The administrator navigates to the "Product Management" section.   
2. The administrator selects an existing product to update.   
3. The system displays the current product details in an editable form.   
4. The administrator modifies the product information (e.g., price, description, stock, or category).   
5. The administrator submits the updated form.   
6. The system validates the new data (e.g., correct price format, valid stock quantity).   
7. The system updates the product record in the database.   
8. The system logs the update action.   
9. The system triggers the email plugin to send a confirmation email to the administrator.   
10. The system displays a success message that the product has been updated.   
  
Alternative Flow:   
1. If the system detects invalid or incomplete product data during validation, it displays an error message and prompts the administrator to correct the information.   
2. If the system fails to update the product record in the database, it logs the error and displays a message to the administrator, suggesting to try again or contact the administrator for support.   
3. If the email plugin fails to send the confirmation email, the system logs the error and displays a message to the administrator, suggesting they verify the email settings or retry the action.   
4. If the administrator cancels the update during the submission step, the system retains the original product details and returns to the product management view.  
  
Use Case Name: Delete Product   
Use Case ID: UC-16   
Actors: Administrator, System, Product, Email Plugin   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The system is operational and accessible.   
3. The product to be deleted exists in the database.   
4. The administrator has the necessary permissions to delete products.   
  
Postconditions:   
1. The specified product is successfully deleted from the system.   
2. The system logs the deletion action and the administrator who performed it.   
3. If configured, the administrator receives a confirmation email about the deletion.   
4. The product is removed from product listings and order processing functionalities.   
  
Main Flow:   
1. The administrator navigates to the "Product Management" section.   
2. The administrator selects the product they wish to delete.   
3. The system displays a confirmation dialog to the administrator.   
4. The administrator confirms the deletion.   
5. The system verifies that the product is not part of any active or pending orders.   
6. The system deletes the product record from the database.   
7. The system logs the deletion action.   
8. The system triggers the email plugin to send a deletion confirmation email to the administrator.   
9. The system displays a success message to the administrator.   
  
Alternative Flow:   
1. If the system detects that the product is part of an active or pending order, it displays an error message and prevents deletion to avoid affecting order fulfillment.   
2. If the system fails to delete the product record from the database, it logs the error and displays a message to the administrator, suggesting to try again or contact the administrator for support.   
3. If the email plugin fails to send the confirmation email, the system logs the error and displays a message to the administrator, suggesting they verify the email settings or retry the action.   
4. If the administrator cancels the deletion during the confirmation step, the system retains the product and returns to the product management view.  
  
Use Case Name: Register as Administrator   
Use Case ID: UC-17   
Actors: Administrator, System, Email Plugin   
  
Preconditions:   
1. The administrator is not yet registered in the system.   
2. The system is operational and accessible.   
3. The email plugin is available and functional.   
  
Postconditions:   
1. The administrator is successfully registered in the system.   
2. A confirmation email is sent to the administrator's email address.   
3. The system logs the registration activity.   
  
Main Flow:   
1. The administrator accesses the administrator registration page of the system.   
2. The administrator fills in the required information (e.g., name, email, password, role).   
3. The administrator submits the registration form.   
4. The system validates the input data (e.g., valid email format, password strength, uniqueness of email).   
5. The system creates a new administrator record in the database.   
6. The system triggers the email plugin to send a confirmation email to the administrator.   
7. The system displays a success message to the administrator.   
  
Alternative Flow:   
1. If the system detects invalid or incomplete data during validation (e.g., missing fields, invalid email), it displays an error message and prompts the administrator to correct the information.   
2. If the system fails to send the confirmation email, it logs the error and displays a message to the administrator, suggesting to check their email or contact the system support.   
3. If the email is already registered, the system displays an error message indicating that the email is already in use and prompts the administrator to use a different one.  
  
Use Case Name: Login as Administrator   
Use Case ID: UC-04   
Actors: Administrator, System, Email Plugin   
  
Preconditions:   
1. The administrator is not currently logged in to the system.   
2. The system is operational and accessible.   
3. The administrator has a valid email and password.   
  
Postconditions:   
1. The administrator is successfully logged into the system.   
2. The system displays the administrator dashboard or homepage.   
3. The system logs the login activity for security and audit purposes.   
  
Main Flow:   
1. The administrator accesses the administrator login page of the system.   
2. The administrator enters their email and password.   
3. The administrator clicks the "Login" button.   
4. The system validates the email and password against the database.   
5. The system confirms the login credentials are correct.   
6. The system logs the successful login attempt.   
7. The system redirects the administrator to their dashboard or homepage.   
  
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 system detects multiple failed login attempts, it locks the account temporarily and notifies the administrator.   
3. If the system fails to log the login activity, it logs the error internally and continues with the user login process.   
4. If the administrator clicks "Forgot Password," the system triggers the email plugin to send a password reset link to the administrator's registered email.  
  
Use Case Name: Manage Customer Accounts   
Use Case ID: UC-18   
Actors: Administrator, System, Customer, Email Plugin   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The system is operational and accessible.   
3. There is at least one customer account in the system.   
  
Postconditions:   
1. The administrator can view, update, or delete customer accounts as needed.   
2. The system logs the administrator's actions for audit purposes.   
3. If applicable, the customer receives a notification via the email plugin.   
  
Main Flow:   
1. The administrator navigates to the "Customer Management" section of the system.   
2. The system displays a list of existing customer accounts with basic details (e.g., name, email, account status).   
3. The administrator selects a specific customer account to manage.   
4. The system retrieves and displays the full details of the selected customer (e.g., contact information, registration date, order history).   
5. The administrator chooses an action: view details, update information, or delete the account.   
6. If updating, the administrator modifies the customer's information (e.g., email, name, account status).   
7. The system validates the updated data and applies the changes to the customer record.   
8. If deleting, the system verifies that the account is not linked to any active orders and proceeds with deletion.   
9. The system logs the action and the administrator who performed it.   
10. If configured, the system triggers the email plugin to notify the customer of any changes.   
11. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the system fails to retrieve customer details, it logs the error and displays a message to the administrator, suggesting they try again or contact the administrator for support.   
2. If the administrator attempts to delete an account linked to active orders, the system displays a warning and prevents the deletion.   
3. If the system detects invalid or incomplete data during an update, it displays an error message and prompts the administrator to correct the information.   
4. If the email plugin fails to send a notification, the system logs the error and displays a message to the administrator, suggesting they verify the email settings or retry the action.   
5. If the administrator cancels the operation during the confirmation step, the system retains the original customer data and returns to the customer management view.  
  
Use Case Name: Manage Product Listings   
Use Case ID: UC-19   
Actors: Administrator, System, Product, Email Plugin   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The system is operational and accessible.   
3. The product database is available and functional.   
4. The administrator has the necessary permissions to manage product listings.   
  
Postconditions:   
1. The administrator successfully manages product listings (e.g., adds, updates, or deletes products).   
2. The system logs all changes made to product listings.   
3. The updated product information is reflected in the product catalog and customer views.   
4. If configured, the administrator receives a confirmation email for each action taken.   
  
Main Flow:   
1. The administrator navigates to the "Product Listings Management" section in the system.   
2. The administrator selects an action: Add Product, Update Product, or Delete Product.   
3. If adding a product, the administrator fills in the product details and submits the form.   
4. If updating a product, the administrator selects an existing product, edits the details, and submits the form.   
5. If deleting a product, the administrator selects the product and confirms the deletion.   
6. The system validates the input data and performs the selected action.   
7. The system updates the product database and logs the action.   
8. The system triggers the email plugin to send a confirmation email to the administrator.   
9. The system displays a success message to the administrator.   
  
Alternative Flow:   
1. If the system detects invalid or incomplete product data during validation, it displays an error message and prompts the administrator to correct the information.   
2. If the system fails to update or delete a product, it logs the error and displays a message to the administrator, suggesting to try again or contact the administrator for support.   
3. If the email plugin fails to send the confirmation email, the system logs the error and displays a message to the administrator, suggesting they verify the email settings or retry the action.   
4. If the administrator cancels the action during the confirmation step, the system retains the current product listing and returns to the management view.  
  
Use Case Name: Manage Orders   
Use Case ID: UC-20   
Actors: Administrator, System, Order, Customer, Email Plugin, Payment   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The system is operational and accessible.   
3. There are existing orders in the system.   
4. The administrator has the necessary permissions to manage orders.   
5. The email plugin and payment gateway are functional if needed.   
  
Postconditions:   
1. The administrator successfully manages orders (e.g., views, updates, or deletes orders).   
2. The system logs all actions taken by the administrator.   
3. The updated order status or details are reflected in the system.   
4. If configured, the customer receives an email notification about the changes.   
5. The payment status is updated if required.   
  
Main Flow:   
1. The administrator navigates to the "Order Management" section of the system.   
2. The system displays a list of all orders, including order ID, customer name, total amount, and current status.   
3. The administrator selects a specific order to manage.   
4. The system retrieves and displays the full details of the selected order (e.g., customer information, product list, payment status, and shipping details).   
5. The administrator chooses an action: view order details, update order status, or delete the order.   
6. If updating the status, the administrator selects the new status (e.g., Processing, Shipped, Delivered, Cancelled) and confirms the change.   
7. The system updates the order status in the database.   
8. The system logs the action and the administrator who performed it.   
9. If configured, the system sends an email notification to the customer via the email plugin.   
10. If deleting an order, the system verifies the order is eligible for deletion and removes it from the database.   
11. The system displays a success message to the administrator.   
  
Alternative Flow:   
1. If the system fails to retrieve the order details, it logs the error and displays a message to the administrator, suggesting they try again or contact the administrator for support.   
2. If the administrator attempts to delete an order that is already shipped or delivered, the system displays a warning and prevents the deletion.   
3. If the system detects invalid or incorrect status updates, it displays an error message and prompts the administrator to select a valid status.   
4. If the email plugin fails to send a notification to the customer, the system logs the error and displays a message to the administrator, suggesting they verify the email settings or retry the action.   
5. If the administrator cancels the action during the confirmation step, the system retains the original order status or details and returns to the order management view.  
  
Use Case Name: View Payment Information   
Use Case ID: UC-21   
Actors: Customer, System, Payment, Email Plugin, Administrator   
  
Preconditions:   
1. The customer is logged into the system.   
2. The system is operational and accessible.   
3. There is at least one payment record associated with the customer.   
4. The payment information is stored in the system's database.   
  
Postconditions:   
1. The customer successfully views their payment details.   
2. The system displays the relevant payment information, such as payment method, amount, and date.   
3. The system logs the customer's request to view payment information.   
  
Main Flow:   
1. The customer navigates to the "Payment History" section in their account.   
2. The system retrieves all payment records associated with the customer from the database.   
3. The system displays the payment information in a list, including payment method, amount, date, and order reference.   
4. The customer can click on a specific payment to view more detailed information.   
5. The system logs the customer's access to payment information for audit purposes.   
  
Alternative Flow:   
1. If the customer has no payment history, the system displays a message indicating that there are no payment records available.   
2. If the system fails to retrieve the payment data, it logs the error and displays a message to the customer, suggesting they try again later or contact the administrator.   
3. If the customer's session expires while viewing payment information, the system redirects them to the login page and displays an access denied message.  
  
Use Case Name: Process Payment   
Use Case ID: UC-07   
Actors: Customer, System, Payment, Email Plugin, Administrator   
  
Preconditions:   
1. The customer is logged into the system.   
2. The customer has at least one product in their cart.   
3. The system is operational and accessible.   
4. The product(s) in the cart are available in the database.   
5. The payment gateway is accessible and functional.   
  
Postconditions:   
1. The payment is successfully processed.   
2. A new order is created in the system.   
3. The customer receives a confirmation email for the order.   
4. The product stock is updated accordingly.   
5. The system logs the payment and order creation activity.   
  
Main Flow:   
1. The customer navigates to the shopping cart.   
2. The customer clicks the "Proceed to Checkout" button.   
3. The system displays the payment form with total amount and payment method options.   
4. The customer selects a payment method and enters the required information.   
5. The system validates the payment information.   
6. The system processes the payment through the payment gateway.   
7. The system confirms the payment success and creates the order.   
8. The system updates the product stock levels.   
9. The system sends an order confirmation email to the customer.   
10. The system displays a success message and provides an order reference number.   
  
Alternative Flow:   
1. If the system detects invalid or incomplete payment information, it displays an error message and prompts the customer to correct the details.   
2. If the payment gateway is unavailable or the payment fails, the system displays an error and suggests trying again or contacting the administrator.   
3. If the product stock is insufficient at the time of payment, the system displays an error message and suggests adjusting the order.   
4. If the system fails to send the confirmation email, it logs the error and displays a message to the customer, suggesting to check their email or contact the administrator.   
5. If the customer cancels the payment process, the system returns them to the cart without creating an order.  
  
Use Case Name: Add Plugin   
Use Case ID: UC-17   
Actors: Administrator, System, Plugin, Email Plugin   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The system is operational and accessible.   
3. The plugin database is available and functional.   
4. The administrator has the necessary permissions to add plugins.   
  
Postconditions:   
1. The specified plugin is successfully added to the system.   
2. The system logs the addition of the plugin and the administrator who performed it.   
3. The plugin is activated and ready for use.   
4. If configured, the administrator receives a confirmation email about the plugin addition.   
  
Main Flow:   
1. The administrator navigates to the "Plugin Management" section in the system.   
2. The administrator clicks the "Add Plugin" button.   
3. The system displays a plugin upload form or integration options.   
4. The administrator selects the plugin (e.g., uploads a file or enters an API integration).   
5. The administrator submits the plugin details.   
6. The system validates the plugin (e.g., checks compatibility, correct format, or API availability).   
7. The system adds the plugin to the database and activates it.   
8. The system logs the plugin addition.   
9. The system triggers the email plugin to send a confirmation email to the administrator.   
10. The system displays a success message to the administrator.   
  
Alternative Flow:   
1. If the system detects invalid or incompatible plugin details during validation, it displays an error message and prompts the administrator to correct the information.   
2. If the system fails to add the plugin to the database, it logs the error and displays a message to the administrator, suggesting to try again or contact the administrator for support.   
3. If the email plugin fails to send the confirmation email, the system logs the error and displays a message to the administrator, suggesting they verify the email settings or retry the action.   
4. If the administrator cancels the addition during the confirmation step, the system retains the current plugin configuration and returns to the plugin management view.  
  
Use Case Name: Update Plugin   
Use Case ID: UC-18   
Actors: Administrator, System, Plugin, Email Plugin   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The system is operational and accessible.   
3. The plugin to be updated exists in the system.   
4. The administrator has the necessary permissions to manage plugins.   
5. The plugin database is available and functional.   
  
Postconditions:   
1. The plugin is successfully updated with new settings or configurations.   
2. The system logs the update action and the administrator who performed it.   
3. The updated plugin is activated and ready for use.   
4. If configured, the administrator receives a confirmation email about the plugin update.   
  
Main Flow:   
1. The administrator navigates to the "Plugin Management" section of the system.   
2. The administrator selects the plugin to be updated.   
3. The system displays the current plugin configuration in an editable form.   
4. The administrator modifies the plugin settings (e.g., API keys, configuration parameters, or activation status).   
5. The administrator submits the updated form.   
6. The system validates the new plugin data (e.g., correct format, valid API access).   
7. The system updates the plugin configuration in the database.   
8. The system logs the update action.   
9. The system triggers the email plugin to send a confirmation email to the administrator.   
10. The system displays a success message to the administrator.   
  
Alternative Flow:   
1. If the system detects invalid or incomplete plugin data during validation, it displays an error message and prompts the administrator to correct the information.   
2. If the system fails to update the plugin configuration in the database, it logs the error and displays a message to the administrator, suggesting to try again or contact the administrator for support.   
3. If the email plugin fails to send the confirmation email, the system logs the error and displays a message to the administrator, suggesting they verify the email settings or retry the action.   
4. If the administrator cancels the update during the confirmation step, the system retains the original plugin settings and returns to the plugin management view.  
  
Use Case Name: Delete Plugin   
Use Case ID: UC-19   
Actors: Administrator, System, Plugin, Email Plugin   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The system is operational and accessible.   
3. The plugin to be deleted exists in the system and is properly registered.   
4. The administrator has the necessary permissions to delete plugins.   
5. The plugin database is available and functional.   
  
Postconditions:   
1. The specified plugin is successfully deleted from the system.   
2. The system logs the deletion action and the administrator who performed it.   
3. If configured, the administrator receives a confirmation email about the plugin deletion.   
4. The plugin is no longer active or available for use.   
  
Main Flow:   
1. The administrator navigates to the "Plugin Management" section in the system.   
2. The administrator selects the plugin they wish to delete.   
3. The system displays a confirmation dialog to the administrator.   
4. The administrator confirms the deletion.   
5. The system verifies that the plugin is not in use by any critical system functions.   
6. The system deletes the plugin from the database.   
7. The system logs the deletion action.   
8. The system triggers the email plugin to send a confirmation email to the administrator.   
9. The system displays a success message to the administrator.   
  
Alternative Flow:   
1. If the system detects that the plugin is currently in use, it displays a warning message and prevents deletion to avoid disrupting system functionality.   
2. If the system fails to delete the plugin from the database, it logs the error and displays a message to the administrator, suggesting to try again or contact the administrator for support.   
3. If the email plugin fails to send the confirmation email, the system logs the error and displays a message to the administrator, suggesting they verify the email settings or retry the action.   
4. If the administrator cancels the deletion during the confirmation step, the system retains the plugin and returns to the plugin management view.  
  
Use Case Name: View Plugin Documentation   
Use Case ID: UC-20   
Actors: Administrator, System, Plugin, Documentation   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The system is operational and accessible.   
3. The plugin documentation is available in the system's documentation repository.   
4. The administrator has navigated to the plugin management or documentation section.   
  
Postconditions:   
1. The administrator successfully views the plugin documentation.   
2. The system displays the relevant documentation for the selected plugin.   
3. The system logs the access to the plugin documentation.   
  
Main Flow:   
1. The administrator navigates to the "Plugin Management" section of the system.   
2. The administrator selects a specific plugin from the list.   
3. The system displays a menu or options related to the plugin, including a "View Documentation" button.   
4. The administrator clicks the "View Documentation" button.   
5. The system retrieves the plugin documentation from the documentation repository.   
6. The system displays the plugin documentation in a user-friendly format.   
7. The system logs the access for audit and reference purposes.   
  
Alternative Flow:   
1. If the system cannot find the requested plugin documentation, it displays an error message and suggests checking the Documentation section for updates or contacting the administrator.   
2. If the system fails to retrieve or display the documentation, it logs the error and displays a message to the administrator, suggesting they try again later or contact the administrator for support.   
3. If the administrator closes the documentation view or navigates away, the system retains the plugin selection and returns to the plugin management view.  
  
Use Case Name: Access API Specifications   
Use Case ID: UC-21   
Actors: Administrator, System, API, Documentation   
Preconditions:   
1. The administrator is logged into the system.   
2. The system is operational and accessible.   
3. The API documentation is available in the system's documentation repository.   
4. The administrator has navigated to the API management or documentation section.   
  
Postconditions:   
1. The administrator successfully views the API specifications.   
2. The system displays the relevant API documentation in a structured format.   
3. The system logs the access to the API documentation for audit purposes.   
  
Main Flow:   
1. The administrator navigates to the "API Management" section of the system.   
2. The administrator selects a specific API from the list of available APIs.   
3. The system displays a menu or options related to the API, including a "View Specifications" button.   
4. The administrator clicks the "View Specifications" button.   
5. The system retrieves the API documentation from the documentation repository.   
6. The system displays the API specifications in a user-friendly format.   
7. The system logs the access for audit and reference purposes.   
  
Alternative Flow:   
1. If the system cannot find the requested API documentation, it displays an error message and suggests checking the Documentation section for updates or contacting the administrator.   
2. If the system fails to retrieve or display the API documentation, it logs the error and displays a message to the administrator, suggesting they try again later or contact the administrator for support.   
3. If the administrator closes the documentation view or navigates away, the system retains the API selection and returns to the API management view.  
  
Use Case Name: Manage Cart   
Use Case ID: UC-22   
Actors: Customer, System, Product, Order, Email Plugin   
  
Preconditions:   
1. The customer is logged into the system.   
2. The system is operational and accessible.   
3. The customer has at least one product in their cart or is viewing an empty cart.   
4. The product database is available and functional.   
  
Postconditions:   
1. The customer successfully manages their cart (e.g., adds, updates, or removes products).   
2. The system updates the cart information and displays confirmation messages for each action.   
3. The system logs the cart modification activity.   
4. If applicable, the customer receives a notification via the email plugin when the cart is updated.   
  
Main Flow:   
1. The customer navigates to the shopping cart page.   
2. The system displays the current contents of the cart, including product names, quantities, and prices.   
3. The customer selects an action: add a product, update the quantity of a product, or remove a product.   
4. If adding a product, the customer selects a product from the product listing and clicks "Add to Cart."   
5. The system verifies the product availability and adds it to the cart.   
6. If updating a product quantity, the customer adjusts the quantity and clicks "Update Cart."   
7. The system validates the new quantity (e.g., not exceeding available stock, not less than zero).   
8. The system updates the cart with the new quantity.   
9. If removing a product, the customer selects the product and clicks "Remove."   
10. The system removes the product from the cart and updates the total.   
11. The system displays a confirmation message for each action taken.   
12. The system logs the cart modification for tracking and audit purposes.   
  
Alternative Flow:   
1. If the system detects insufficient stock when adding or updating a product, it displays an error message and suggests adjusting the quantity or removing the item.   
2. If the system fails to update the cart (e.g., database error), it logs the error and displays a message to the customer, suggesting to try again or contact the administrator.   
3. If the customer tries to add a product with invalid or non-numeric quantity, the system displays an error and prompts for a valid input.   
4. If the system fails to display the cart, it logs the error and displays a message to the customer, suggesting they refresh the page or contact the administrator.   
5. If the email plugin fails to send a notification about the cart update, the system logs the error and displays a message to the customer, suggesting they verify their email settings or try again later.  
  
Use Case Name: Manage Order Status   
Use Case ID: UC-23   
Actors: Administrator, System, Order, Email Plugin, Customer   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The system is operational and accessible.   
3. There is an existing order in the system that the administrator wishes to manage.   
4. The email plugin is functional if email notifications are configured.   
  
Postconditions:   
1. The order status is successfully updated in the system.   
2. The system logs the status change and the administrator who performed it.   
3. If configured, the customer receives an email notification about the updated order status.   
4. The updated status is reflected in the order history and dashboard for the customer.   
  
Main Flow:   
1. The administrator navigates to the "Order Management" section of the system.   
2. The administrator selects a specific order from the list of orders.   
3. The system displays the current status and details of the selected order.   
4. The administrator chooses a new status for the order (e.g., Processing, Shipped, Delivered, Cancelled).   
5. The administrator confirms the status change.   
6. The system updates the order status in the database.   
7. The system logs the change along with the administrator’s identity.   
8. If email notifications are enabled, the system triggers the email plugin to send a status update notification to the customer.   
9. The system displays a success message to the administrator confirming the status update.   
  
Alternative Flow:   
1. If the system detects that the new status is not valid for the current order (e.g., attempting to mark an order as "Delivered" when it is still in "Processing"), it displays an error message and prompts the administrator to select a valid status.   
2. If the system fails to update the order status in the database, it logs the error and displays a message to the administrator, suggesting to retry the action or contact the administrator for support.   
3. If the email plugin fails to send the notification to the customer, the system logs the error and displays a message to the administrator, suggesting to verify the email settings or retry the notification.   
4. If the administrator cancels the action during the confirmation step, the system retains the original order status and returns to the order management view.  
  
Use Case Name: Manage Payment Method   
Use Case ID: UC-24   
Actors: Administrator, System, Payment Method, Email Plugin   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The system is operational and accessible.   
3. The payment method database is available and functional.   
4. The administrator has the necessary permissions to manage payment methods.   
  
Postconditions:   
1. The administrator successfully manages payment methods (e.g., adds, updates, or deletes a payment method).   
2. The system logs the action taken and the administrator who performed it.   
3. The updated payment method information is reflected in the system and available for customer use.   
4. If configured, the administrator receives a confirmation email about the action.   
  
Main Flow:   
1. The administrator navigates to the "Payment Method Management" section in the system.   
2. The system displays a list of existing payment methods with their details (e.g., method name, status, and configuration).   
3. The administrator selects an action: Add Payment Method, Update Payment Method, or Delete Payment Method.   
4. If adding a new payment method, the administrator fills in the required details (e.g., name, description, configuration parameters).   
5. The system validates the input data (e.g., correct format, uniqueness of method name).   
6. The system performs the selected action (add, update, or delete) on the payment method.   
7. The system updates the payment method database and logs the action.   
8. The system triggers the email plugin to send a confirmation email to the administrator.   
9. The system displays a success message to the administrator.   
  
Alternative Flow:   
1. If the system detects invalid or incomplete data during validation, it displays an error message and prompts the administrator to correct the information.   
2. If the system fails to update or delete the payment method in the database, it logs the error and displays a message to the administrator, suggesting to try again or contact the administrator for support.   
3. If the email plugin fails to send the confirmation email, the system logs the error and displays a message to the administrator, suggesting they verify the email settings or retry the action.   
4. If the administrator cancels the action during the confirmation step, the system retains the current payment method configuration and returns to the payment method management view.   
5. If the administrator attempts to delete a payment method that is currently being used for active orders, the system displays a warning message and prevents the deletion to avoid disrupting order processing.  
  
Use Case Name: Manage Email   
Use Case ID: UC-25   
Actors: Administrator, System, Email Plugin   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The system is operational and accessible.   
3. The email plugin is installed and available for configuration.   
4. The administrator has the necessary permissions to manage email settings.   
  
Postconditions:   
1. The administrator successfully manages email-related configurations (e.g., updates email server settings, configures templates, or enables/disables the email plugin).   
2. The system logs the changes made to the email configuration.   
3. The updated email settings are reflected in the system's email-sending functionality.   
4. If configured, the administrator receives a confirmation email about the changes.   
  
Main Flow:   
1. The administrator navigates to the "Email Management" section of the system.   
2. The system displays the current email configuration (e.g., SMTP settings, template details, and plugin status).   
3. The administrator selects an action: update email server settings, edit email templates, or toggle the email plugin's activation status.   
4. If updating server settings, the administrator enters new email server details (e.g., host, port, username, password).   
5. If editing templates, the administrator modifies the content of predefined email messages (e.g., registration confirmation, order status update).   
6. The system validates the input data (e.g., correct format for server details, valid template syntax).   
7. The system applies the selected changes to the email plugin configuration.   
8. The system logs the modification and the administrator who performed it.   
9. If configured, the system triggers the email plugin to send a confirmation email to the administrator.   
10. The system displays a success message to the administrator confirming the email settings have been updated.   
  
Alternative Flow:   
1. If the system detects invalid or incomplete email server settings (e.g., incorrect port number, missing host), it displays an error message and prompts the administrator to correct the information.   
2. If the system fails to apply the changes to the email plugin (e.g., database update fails), it logs the error and displays a message to the administrator, suggesting to retry the action or contact the administrator for support.   
3. If the email plugin fails to send the confirmation email after configuration changes, the system logs the error and displays a message to the administrator, suggesting they verify the email settings or retry the action.   
4. If the administrator cancels the action during the confirmation step, the system retains the original email configuration and returns to the email management view.   
5. If the system detects that the email plugin is in use for critical functions and the administrator attempts to disable it, the system displays a warning message and prevents the action to avoid disrupting operations.  
  
Use Case Name: Manage API Documentation   
Use Case ID: UC-22   
Actors: Administrator, System, API, Documentation, Email Plugin   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The system is operational and accessible.   
3. The API documentation database or repository is available and functional.   
4. The administrator has the necessary permissions to manage API documentation.   
  
Postconditions:   
1. The administrator successfully manages API documentation (e.g., adds, updates, or deletes documentation for specific APIs).   
2. The system logs the documentation management actions and the administrator who performed them.   
3. The updated documentation is accessible in the system for reference.   
4. If configured, the administrator receives a confirmation email about the changes.   
  
Main Flow:   
1. The administrator navigates to the "API Documentation Management" section in the system.   
2. The system displays a list of APIs with associated documentation status (e.g., available, outdated, or missing).   
3. The administrator selects an API for which they want to manage documentation.   
4. The administrator chooses an action: Add Documentation, Update Documentation, or Delete Documentation.   
5. If adding documentation, the administrator uploads or inputs the documentation content (e.g., in Markdown, PDF, or HTML format).   
6. If updating documentation, the administrator edits the existing content and saves the changes.   
7. The system validates the documentation format and content.   
8. The system updates the API documentation in the repository or database.   
9. The system logs the documentation action.   
10. The system triggers the email plugin to send a confirmation email to the administrator.   
11. The system displays a success message to the administrator.   
  
Alternative Flow:   
1. If the system detects invalid or unsupported documentation format, it displays an error message and prompts the administrator to provide a valid format.   
2. If the system fails to save the documentation in the repository or database, it logs the error and displays a message to the administrator, suggesting to retry the action or contact the administrator for support.   
3. If the administrator attempts to delete documentation for an API that is in active use, the system displays a warning message and prevents the deletion to avoid disrupting API access.   
4. If the email plugin fails to send the confirmation email, the system logs the error and displays a message to the administrator, suggesting they verify the email settings or retry the action.   
5. If the administrator cancels the action during the confirmation step, the system retains the current documentation and returns to the API documentation management view.  
  
Use Case Name: View Order Status   
Use Case ID: UC-23   
Actors: Customer, System, Order, Email Plugin   
  
Preconditions:   
1. The customer is logged into the system.   
2. The system is operational and accessible.   
3. The customer has at least one order in the system.   
4. The order status information is available in the database.   
  
Postconditions:   
1. The customer successfully views the status of their order.   
2. The system displays the current order status and relevant details (e.g., order date, total amount, and expected delivery time).   
3. The system logs the customer's request to view the order status.   
4. If the customer requests additional information about the order status and it is not available, the system may suggest contacting the administrator.   
  
Main Flow:   
1. The customer navigates to the "Order History" section in their account.   
2. The system displays a list of past orders, including order ID, date, total amount, and current status.   
3. The customer selects a specific order to view its status.   
4. The system retrieves the order details from the database.   
5. The system displays the current status of the selected order and any additional relevant information (e.g., tracking details if available).   
6. The system logs the customer's access to the order status.   
  
Alternative Flow:   
1. If the system fails to retrieve the order status information, it logs the error and displays a message to the customer, suggesting they try again later or contact the administrator.   
2. If the customer's session expires while viewing the order status, the system redirects them to the login page and displays an access denied message.   
3. If the order does not exist or has been deleted, the system displays an error message and suggests checking the order history again or contacting the administrator for assistance.