项目文档

# Functional Requirement

1. Functional Requirements   
 1.1 Customer Account Creation Function   
 Function ID: FR-01   
 Description: Customers can register and create an account with the system by providing personal information such as name, email, and password.   
 Input: Customer's name, email, password, and address.   
 Output: A new customer account with associated information stored in the database.   
  
 1.2 Customer Login Function   
 Function ID: FR-02   
 Description: Registered customers can log in to their account using their email and password.   
 Input: Email and password provided by the customer.   
 Output: A successful login session with the system, and the login event recorded in the activity log.   
  
 1.3 Customer Logout Function   
 Function ID: FR-03   
 Description: Customers can log out of their account to end their session.   
 Input: Customer selects the "Logout" option from the user interface.   
 Output: The session is terminated, and the logout event is recorded in the activity log.   
  
 1.4 Product Catalog Display Function   
 Function ID: FR-04   
 Description: The system displays a categorized and searchable product catalog for customers to browse available products.   
 Input: Customer navigates to the product catalog section.   
 Output: A dynamically updated list of products with their names, descriptions, prices, and availability.   
  
 1.5 Product Addition to Cart Function   
 Function ID: FR-05   
 Description: Customers can add selected products to their shopping cart with a specified quantity.   
 Input: Product ID, quantity, and the customer's cart.   
 Output: Updated shopping cart with the newly added product and quantity.   
  
 1.6 Cart Quantity Update Function   
 Function ID: FR-06   
 Description: Customers can modify the quantity of products in their shopping cart.   
 Input: Cart item ID and new quantity.   
 Output: Updated cart with adjusted quantity and recalculated total cost.   
  
 1.7 Cart Total Cost Calculation Function   
 Function ID: FR-07   
 Description: The system calculates and displays the total cost of all items in the shopping cart.   
 Input: Product prices and quantities in the customer's cart.   
 Output: Total cost displayed on the cart page for the customer to review.   
  
 1.8 Order Confirmation Function   
 Function ID: FR-08   
 Description: Customers can confirm their orders after reviewing the cart contents and selected payment method.   
 Input: Cart contents, payment method, and customer confirmation.   
 Output: A confirmed order stored in the system, and inventory updated to reflect the reduction in stock.   
  
 1.9 Order Confirmation Email Function   
 Function ID: FR-09   
 Description: After an order is confirmed, the system sends an email confirmation to the customer.   
 Input: Customer's email address, order details, and confirmation number.   
 Output: An email notification with order details sent to the customer's email address.   
  
 1.10 Inventory Management Function   
 Function ID: FR-10   
 Description: Administrators can manage the inventory by adding, updating, or removing products.   
 Input: Product details such as name, description, price, and quantity.   
 Output: Updated inventory with the latest product information stored in the database.   
  
 1.11 Product Categorization Function   
 Function ID: FR-11   
 Description: Administrators can assign products to specific categories to improve navigation and organization.   
 Input: Product ID and category ID or name.   
 Output: Updated product database with the correct category assignment.   
  
 1.12 Product Details Update Function   
 Function ID: FR-12   
 Description: Administrators can update product information such as price, description, or stock quantity.   
 Input: Product ID and updated information (e.g., new price, updated description).   
 Output: Updated product details stored in the database.   
  
 1.13 Administrator Login Function   
 Function ID: FR-13   
 Description: Administrators can log in to the system using their email and password to access the control panel.   
 Input: Email and password provided by the administrator.   
 Output: A successful admin login session, and the login event recorded in the activity log.   
  
 1.14 Administrator Logout Function   
 Function ID: FR-14   
 Description: Administrators can log out of their account to end their session securely.   
 Input: Administrator selects the "Logout" option from the admin interface.   
 Output: The admin session is terminated, and the logout event is recorded in the activity log.   
  
 1.15 Purchase History Display Function   
 Function ID: FR-15   
 Description: Customers can view their purchase history, including previous orders and details.   
 Input: Customer ID and filter criteria (e.g., date range, product).   
 Output: A list of past orders with details such as product names, quantities, and total cost.   
  
 1.16 Payment Information Management Function   
 Function ID: FR-16   
 Description: Customers can manage their payment information by adding, editing, or deleting payment methods.   
 Input: Payment details (e.g., card number, expiration date, billing address) and action (add/edit/delete).   
 Output: Updated customer account with stored payment information, and validation results displayed.   
  
 1.17 Plugin Installation Function   
 Function ID: FR-17   
 Description: Administrators can install plugins from the system’s plugin repository to extend system functionality.   
 Input: Plugin ID and confirmation to install.   
 Output: Plugin installed and activated, and the system updated to reflect the new functionality.   
  
 1.18 Plugin API Documentation Access Function   
 Function ID: FR-18   
 Description: Administrators can access the API documentation for installed plugins to understand their integration and usage.   
 Input: Plugin ID and search/filter criteria.   
 Output: Displayed API documentation with endpoints, methods, and usage examples.   
  
 1.19 Transaction Processing Function   
 Function ID: FR-19   
 Description: The system processes transactions during checkout, including payment execution and inventory updates.   
 Input: Cart contents, selected payment method, and customer confirmation.   
 Output: A processed transaction with quantity and cost recorded, and inventory updated accordingly.   
  
 1.20 Email Notification Sending Function   
 Function ID: FR-20   
 Description: The system can send email notifications to customers for events such as order confirmations or password resets.   
 Input: Recipient email address, message content, and subject.   
 Output: Email notification sent to the customer, and the event recorded in the system logs.   
  
 1.21 User Activity Log Display Function   
 Function ID: FR-21   
 Description: Administrators can view and filter user activity logs to track user actions such as login, logout, and cart updates.   
 Input: Filter criteria such as user ID, action type, or date range.   
 Output: A structured log of user activities displayed for review by the administrator.

# External Description

# 2. External Interfaces  
  
This section describes the external interfaces that the system interacts with to fulfill its functional requirements. These interfaces include user interfaces, hardware interfaces, software interfaces (including databases and third-party systems), and communication interfaces such as email or web services.  
  
---  
  
## 2.1 User Interface  
  
### 2.1.1 Customer Login/Logout UI  
- \*\*Description\*\*: The system provides a user interface for customers to log in and log out.  
- \*\*Interaction Method\*\*: Web-based interface where users enter their email and password to authenticate. A "Logout" button is available on the user dashboard.  
- \*\*Input\*\*: Email address and password for login; user action (clicking "Logout") for logout.  
- \*\*Output\*\*: Display of login form, success message upon login, and confirmation message upon logout.  
- \*\*Purpose\*\*: Enables secure user authentication and session management.  
  
---  
  
### 2.1.2 Product Catalog UI  
- \*\*Description\*\*: The system presents a user interface for browsing and searching products.  
- \*\*Interaction Method\*\*: Web-based interface with filters, search bar, and product listing.  
- \*\*Input\*\*: User navigation to the catalog page, search queries, and filter selections.  
- \*\*Output\*\*: Dynamically updated list of products with name, description, price, and availability.  
- \*\*Purpose\*\*: Facilitates customer product discovery and selection.  
  
---  
  
### 2.1.3 Shopping Cart UI  
- \*\*Description\*\*: The system provides a user interface for managing items in the shopping cart.  
- \*\*Interaction Method\*\*: Web-based interface with buttons to add/remove items, adjust quantities, and view total cost.  
- \*\*Input\*\*: User interaction to add, remove, or update item quantities.  
- \*\*Output\*\*: Updated cart display showing current items, quantities, and total cost.  
- \*\*Purpose\*\*: Supports seamless online purchasing experience.  
  
---  
  
### 2.1.4 Order Confirmation UI  
- \*\*Description\*\*: The system provides a user interface for confirming orders.  
- \*\*Interaction Method\*\*: Web-based interface where users review cart contents and select payment method.  
- \*\*Input\*\*: User confirmation of order and selected payment method.  
- \*\*Output\*\*: Display of confirmation screen and success message.  
- \*\*Purpose\*\*: Ensures accurate order processing and customer confirmation.  
  
---  
  
### 2.1.5 Purchase History UI  
- \*\*Description\*\*: The system provides a user interface for viewing purchase history.  
- \*\*Interaction Method\*\*: Web-based interface with filters by date range, product, or other criteria.  
- \*\*Input\*\*: User filter criteria (e.g., date range, product).  
- \*\*Output\*\*: List of past orders with product names, quantities, and total cost.  
- \*\*Purpose\*\*: Allows customers to track their previous purchases.  
  
---  
  
### 2.1.6 Payment Information Management UI  
- \*\*Description\*\*: The system provides a user interface for managing payment methods.  
- \*\*Interaction Method\*\*: Web-based interface where users can add, edit, or delete payment details.  
- \*\*Input\*\*: User input of payment information and selection of action (add/edit/delete).  
- \*\*Output\*\*: Updated payment information stored in the system and validation results displayed.  
- \*\*Purpose\*\*: Enables secure and flexible payment management.  
  
---  
  
### 2.1.7 Admin Dashboard UI  
- \*\*Description\*\*: The system provides a user interface for administrators to manage inventory, plugins, and user activity logs.  
- \*\*Interaction Method\*\*: Web-based interface with sections for product management, plugin installation, and activity log viewing.  
- \*\*Input\*\*: User actions such as adding a product, installing a plugin, or filtering logs.  
- \*\*Output\*\*: Dynamic display of product data, plugin status, and user activity logs.  
- \*\*Purpose\*\*: Enables efficient administration and monitoring of the system.  
  
---  
  
### 2.1.8 Plugin API Documentation UI  
- \*\*Description\*\*: The system provides a user interface for accessing plugin API documentation.  
- \*\*Interaction Method\*\*: Web-based interface with search and filter functionality.  
- \*\*Input\*\*: User input of plugin ID or search terms.  
- \*\*Output\*\*: Displayed API documentation including endpoints, methods, and usage examples.  
- \*\*Purpose\*\*: Supports developers and administrators in integrating and utilizing plugins.  
  
---  
  
### 2.1.9 Email Notification UI  
- \*\*Description\*\*: The system provides a user interface for sending email notifications.  
- \*\*Interaction Method\*\*: Web-based interface where administrators can compose and send emails.  
- \*\*Input\*\*: Recipient email address, message content, and subject.  
- \*\*Output\*\*: Confirmation of email sent and event logged.  
- \*\*Purpose\*\*: Enables automated communication with users.  
  
---  
  
### 2.1.10 User Activity Log UI  
- \*\*Description\*\*: The system provides a user interface for viewing and filtering user activity logs.  
- \*\*Interaction Method\*\*: Web-based interface with filters by user ID, action type, or date range.  
- \*\*Input\*\*: Filter criteria (e.g., user ID, action type, date range).  
- \*\*Output\*\*: Structured log of user activities for review.  
- \*\*Purpose\*\*: Provides visibility into user behavior and system usage patterns.  
  
---  
  
## 2.2 Hardware Interface  
  
\*\*No hardware interfaces are used in this system.\*\*  
  
---  
  
## 2.3 Software Interface  
  
### 2.3.1 Database Interface  
- \*\*Description\*\*: The system interacts with a relational database to store and retrieve customer, product, order, and activity data.  
- \*\*Interaction Method\*\*: SQL-based database access through application programming interfaces (APIs) or direct database calls.  
- \*\*Input\*\*: Data from functional requirements such as customer information, product details, order records, etc.  
- \*\*Output\*\*: Data retrieved for use in various functions (e.g., login, cart updates, order processing).  
- \*\*Purpose\*\*: Ensures persistent storage and retrieval of critical system data.  
  
### 2.3.2 Email Service Interface  
- \*\*Description\*\*: The system communicates with an external email service to send transactional emails such as order confirmations and password reset links.  
- \*\*Interaction Method\*\*: RESTful API calls to an email service provider (e.g., SendGrid, Amazon SES).  
- \*\*Input\*\*: Recipient email address, message content, and subject line.  
- \*\*Output\*\*: Email notification sent to the recipient, with confirmation of delivery.  
- \*\*Purpose\*\*: Enables automated email communication with users.  
  
### 2.3.3 Payment Gateway Interface  
- \*\*Description\*\*: The system interacts with a payment gateway to process payments during checkout.  
- \*\*Interaction Method\*\*: RESTful API integration with a payment service provider (e.g., Stripe, PayPal).  
- \*\*Input\*\*: Payment method details (e.g., card number, expiration date), cart contents, and customer confirmation.  
- \*\*Output\*\*: Payment confirmation, transaction ID, and any error messages.  
- \*\*Purpose\*\*: Enables secure and reliable payment processing.  
  
### 2.3.4 Plugin Repository Interface  
- \*\*Description\*\*: The system interacts with a plugin repository to install and manage plugins.  
- \*\*Interaction Method\*\*: RESTful API calls to a plugin management system.  
- \*\*Input\*\*: Plugin ID and installation confirmation.  
- \*\*Output\*\*: Plugin installed and activated, with system updates reflecting new functionality.  
- \*\*Purpose\*\*: Extends system capabilities through modular additions.  
  
---  
  
## 2.4 Communication Interface  
  
### 2.4.1 Email Communication Interface  
- \*\*Description\*\*: The system sends email notifications to customers for events such as order confirmations, password resets, and account updates.  
- \*\*Interaction Method\*\*: SMTP protocol or RESTful API for sending emails.  
- \*\*Input\*\*: Recipient email address, message content, and subject.  
- \*\*Output\*\*: Email notification sent to the recipient, and event recorded in the system logs.  
- \*\*Purpose\*\*: Provides timely and relevant communication to users.  
  
### 2.4.2 Web Browsing Interface  
- \*\*Description\*\*: The system allows users to interact with it via web browsers.  
- \*\*Interaction Method\*\*: HTTP/HTTPS protocol for client-server communication.  
- \*\*Input\*\*: User requests via web browser (e.g., login, product search, order placement).  
- \*\*Output\*\*: HTML pages rendered in the browser with dynamic content based on user actions.  
- \*\*Purpose\*\*: Enables broad accessibility and usability across devices.  
  
### 2.4.3 API Communication Interface  
- \*\*Description\*\*: The system provides APIs for internal and external integrations.  
- \*\*Interaction Method\*\*: RESTful API using JSON format for data exchange.  
- \*\*Input\*\*: API requests containing parameters such as product IDs, user credentials, and cart details.  
- \*\*Output\*\*: API responses with structured data (e.g., product listings, order statuses, user activity logs).  
- \*\*Purpose\*\*: Supports integration with other systems and enables programmatic access to system functionalities.  
  
---  
  
## Summary of External Interfaces  
  
| Interface Type | Description | Input | Output |  
|----------------------|-----------------------------------------------------------------------------|----------------------------------------------------------------------|------------------------------------------------------------------------|  
| User Interface | All user-facing interactions (login, product browsing, cart, etc.) | User inputs (email, password, search terms, etc.) | Screen displays, confirmation messages, updated views |  
| Software Interface | Interaction with databases, email services, payment gateways, and plugins | Data from functional requirements (e.g., customer info, product data)| Stored data, API responses, payment confirmations, plugin activation |  
| Communication Interface | Email, web browsing, and API communications | Email addresses, web requests, API calls | Emails, web pages, API responses |  
  
All external interfaces have been clearly defined and mapped to the functional requirements to ensure consistency and clarity for developers and stakeholders.

# Use Case

Use Case Name: Customer Register   
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. A new customer account is successfully created.   
2. The customer receives a confirmation message or email.   
3. The system updates the user database with the new customer information.   
  
Main Flow:   
1. The customer navigates to the registration page.   
2. The system displays the registration form with required fields (e.g., name, email, password, phone number).   
3. The customer fills in the required information and submits the form.   
4. The system validates the input data (e.g., checks if the email is unique and in correct format).   
5. The system creates a new customer account in the database.   
6. The system sends a confirmation email to the customer.   
7. The system displays a success message to the customer.   
  
Alternative Flow:   
1. If the email is already registered, the system displays an error message.   
2. If the input data is invalid (e.g., missing fields, incorrect format), the system displays an error message and prompts the customer to correct the data.   
3. If the system fails to send the confirmation email, the system logs the error and displays a message to the customer, allowing them to retry or contact support.  
  
Use Case Name: Customer Login   
Use Case ID: UC-02   
Actors: Customer, System   
Preconditions:   
1. The customer has a valid account registered in the system.   
2. The system is operational and accessible.   
3. The customer is not currently logged in.   
  
Postconditions:   
1. The customer is successfully logged into their account.   
2. The system displays the customer's dashboard or homepage.   
3. The system records the login event in the user activity log.   
  
Main Flow:   
1. The customer navigates to the login page.   
2. The system displays the login form with fields for email and password.   
3. The customer enters their registered email and password.   
4. The customer submits the login form.   
5. The system verifies the email and password against the database.   
6. If valid, the system logs in the customer and redirects them to the homepage or dashboard.   
7. The system displays a welcome message and updates the session status.   
  
Alternative Flow:   
1. If the email is not registered, the system displays an error message.   
2. If the password is incorrect, the system displays an error message and allows the customer to retry.   
3. If the system fails to authenticate the customer, it displays an error message and logs the failed attempt for security monitoring.   
4. If the customer account is locked or disabled, the system displays a message and prevents login.  
  
Use Case Name: Customer Logout   
Use Case ID: UC-03   
Actors: Customer, System   
Preconditions:   
1. The customer is currently logged in.   
2. The system is operational and accessible.   
  
Postconditions:   
1. The customer's session is terminated.   
2. The customer is redirected to the login or homepage.   
3. The system records the logout event in the user activity log.   
  
Main Flow:   
1. The customer navigates to the logout option in the user interface.   
2. The system prompts the customer to confirm the logout action.   
3. The customer confirms the logout.   
4. The system terminates the active session and clears session data.   
5. The system redirects the customer to the login page or homepage.   
6. The system logs the logout event for audit and tracking purposes.   
  
Alternative Flow:   
1. If the customer cancels the logout confirmation, the system returns to the previous page without any changes.   
2. If the system fails to terminate the session, it displays an error message and logs the issue for troubleshooting.  
  
Use Case Name: Create Account   
Use Case ID: UC-04   
Actors: User, System   
Preconditions:   
1. The user is not yet registered in the system.   
2. The system is operational and accessible.   
3. The user has navigated to the account creation page.   
  
Postconditions:   
1. A new user account is successfully created.   
2. The user receives a confirmation email or message.   
3. The system updates the account database with the new user information.   
  
Main Flow:   
1. The user navigates to the account creation page.   
2. The system displays the account creation form with required fields (e.g., full name, email, password, and contact information).   
3. The user fills in the required information and submits the form.   
4. The system validates the input data (e.g., unique email, correct password format, complete fields).   
5. The system creates a new account for the user in the database.   
6. The system sends a confirmation email to the user's provided email address.   
7. The system displays a success message indicating the account creation is complete.   
  
Alternative Flow:   
1. If the email is already registered, the system displays an error message.   
2. If the input data is incomplete or invalid, the system displays an error message and prompts the user to correct the information.   
3. If the confirmation email cannot be sent, the system logs the error and displays a message to the user, offering options to retry or contact support.  
  
Use Case Name: View Product Catalog   
Use Case ID: UC-05   
Actors: Customer, System   
Preconditions:   
1. The customer is logged into the system.   
2. The system is operational and has an updated inventory.   
3. The product catalog is accessible and contains valid product data.   
  
Postconditions:   
1. The customer views the product catalog with available product details.   
2. The system records the catalog access in the user activity log.   
3. If a product is clicked, the system displays its detailed information.   
  
Main Flow:   
1. The customer navigates to the "Product Catalog" section from the homepage or dashboard.   
2. The system retrieves the latest product information from the database.   
3. The system displays the product catalog in a categorized and searchable format.   
4. The customer selects a product to view more details.   
5. The system shows the product's specifications, price, and availability.   
6. The system updates the activity log to reflect the catalog browsing.   
  
Alternative Flow:   
1. If the product catalog is empty or not updated, the system displays a message indicating no products are available.   
2. If the system fails to retrieve product data, it logs the error and displays a message to the customer, suggesting they refresh the page or contact support.   
3. If the selected product has no details available, the system shows an error message and returns to the catalog view.  
  
Use Case Name: Add Product to Cart   
Use Case ID: UC-06   
Actors: Customer, System   
Preconditions:   
1. The customer is logged into the system.   
2. The product is available in the inventory.   
3. The system is operational and accessible.   
4. The customer has navigated to the product details page.   
  
Postconditions:   
1. The selected product is added to the customer's shopping cart.   
2. The cart is updated in the system with the new product information.   
3. The system displays a confirmation message that the product was successfully added to the cart.   
  
Main Flow:   
1. The customer browses the product catalog and selects a product.   
2. The system displays the product details, including price, quantity, and specifications.   
3. The customer clicks the "Add to Cart" button.   
4. The system checks the product availability in the inventory.   
5. The system adds the product to the customer's shopping cart with the selected quantity.   
6. The system updates the cart in the database.   
7. The system displays a confirmation message that the product was successfully added to the cart.   
  
Alternative Flow:   
1. If the product is out of stock, the system displays an error message and does not add the product to the cart.   
2. If the selected quantity exceeds the available stock, the system displays a warning and adjusts the quantity to the maximum available.   
3. If the system fails to update the cart, it displays an error message and logs the issue for troubleshooting.  
  
Use Case Name: Update Cart Quantity   
Use Case ID: UC-07   
Actors: Customer, System   
Preconditions:   
1. The customer is logged into the system.   
2. The customer has at least one product in their shopping cart.   
3. The system is operational and accessible.   
  
Postconditions:   
1. The quantity of the selected product in the shopping cart is updated.   
2. The cart is saved in the system with the new quantity.   
3. The system recalculates and displays the updated total cost.   
  
Main Flow:   
1. The customer navigates to the shopping cart page.   
2. The system displays the current cart contents, including product names, quantities, and prices.   
3. The customer selects a product and adjusts the quantity using the provided controls.   
4. The customer clicks the "Update Quantity" button.   
5. The system verifies that the new quantity is within the available stock limits.   
6. The system updates the cart with the new quantity and recalculates the total cost.   
7. The system displays a confirmation message indicating the cart has been updated.   
  
Alternative Flow:   
1. If the new quantity exceeds available stock, the system displays a warning and adjusts the quantity to the maximum available.   
2. If the new quantity is less than one, the system removes the product from the cart and displays a message.   
3. If the system fails to update the cart, it displays an error message and logs the issue for troubleshooting.  
  
Use Case Name: View Cart Total Cost   
Use Case ID: UC-08   
Actors: Customer, System   
Preconditions:   
1. The customer is logged into the system.   
2. The customer has at least one product in their shopping cart.   
3. The system is operational and accessible.   
  
Postconditions:   
1. The total cost of the shopping cart is displayed to the customer.   
2. The system records the cart view activity in the user activity log.   
  
Main Flow:   
1. The customer navigates to the shopping cart page.   
2. The system retrieves the cart details, including product prices and quantities.   
3. The system calculates the total cost based on the current items in the cart.   
4. The system displays the total cost prominently on the cart page.   
5. The system logs the cart view event for audit and tracking purposes.   
  
Alternative Flow:   
1. If the cart is empty, the system displays a message indicating no items are in the cart.   
2. If the system fails to calculate the total cost due to an error, it displays an error message and logs the issue for troubleshooting.  
  
Use Case Name: Confirm Order   
Use Case ID: UC-09   
Actors: Customer, System, Payment, Inventory, Administrator   
  
Preconditions:   
1. The customer is logged into the system.   
2. The customer has at least one product in their shopping cart.   
3. The system has validated the cart contents against the inventory.   
4. The customer has selected a valid payment method.   
  
Postconditions:   
1. The order is successfully confirmed and stored in the system.   
2. The inventory is updated to reflect the reduction in product stock.   
3. A confirmation message or email is sent to the customer.   
4. The payment is processed, and the transaction is recorded.   
5. The system generates an order confirmation document.   
  
Main Flow:   
1. The customer navigates to the cart confirmation page.   
2. The system displays the cart contents, total cost, and selected payment method.   
3. The customer reviews the order and clicks the "Confirm Order" button.   
4. The system checks the inventory for sufficient stock of all items in the cart.   
5. The system initiates the payment processing using the selected method.   
6. If the payment is successful, the system confirms the order and updates the inventory.   
7. The system generates a confirmation document and sends it to the customer's email.   
8. The system displays a success message to the customer, including the order details and confirmation number.   
  
Alternative Flow:   
1. If the inventory does not have sufficient stock for any item, the system displays an error message and prevents order confirmation.   
2. If the payment fails, the system displays an error message and logs the failed transaction.   
3. If the system fails to send the confirmation email, it logs the error and displays a message to the customer, offering to retry or contact support.   
4. If the customer cancels the confirmation, the system returns to the cart page without processing the order.   
5. If the system encounters an error during order confirmation, it displays an error message and logs the issue for troubleshooting.  
  
Use Case Name: Receive Order Confirmation Email   
Use Case ID: UC-10   
Actors: Customer, System, Email, Order, Payment, Transaction   
  
Preconditions:   
1. The customer has successfully confirmed their order.   
2. The system has completed the payment processing and updated the inventory.   
3. The customer has provided a valid email address during registration or checkout.   
4. The email service is operational and integrated with the system.   
  
Postconditions:   
1. The customer receives an order confirmation email with the order details.   
2. The system logs the successful email delivery.   
3. The email contains a confirmation number and links to view the order status.   
  
Main Flow:   
1. The system triggers the email notification process after the order is confirmed.   
2. The system retrieves the customer's email address and order details from the database.   
3. The system generates an order confirmation email with information such as order ID, product list, total cost, and delivery status.   
4. The system sends the email to the customer's registered email address.   
5. The system logs the email sending event for tracking and audit purposes.   
6. The customer receives the email and can access the order details.   
  
Alternative Flow:   
1. If the customer's email address is invalid or missing, the system logs the error and displays a message to the customer to update their email information.   
2. If the email service is unavailable, the system logs the failure and displays a message to the customer, offering to retry or contact support.   
3. If the customer does not receive the email, the system allows them to request a re-sending of the confirmation email.   
4. If the system fails to generate the email content, it displays an error message and logs the issue for troubleshooting.  
  
Use Case Name: Manage Inventory   
Use Case ID: UC-11   
Actors: Administrator, System, Inventory, Product   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The system is operational and accessible.   
3. The inventory is accessible and contains valid product data.   
  
Postconditions:   
1. The inventory is updated with the latest product information (e.g., added, removed, or modified products).   
2. The system records the inventory management actions for audit and tracking.   
3. Product availability is accurately reflected in the system.   
  
Main Flow:   
1. The administrator navigates to the "Inventory Management" section of the system.   
2. The system displays the current inventory with product details such as name, quantity, price, and category.   
3. The administrator selects an action (e.g., add product, remove product, update quantity or price).   
4. The system prompts the administrator to enter or confirm the necessary details for the selected action.   
5. The administrator submits the changes.   
6. The system validates the input data (e.g., correct format, valid quantity, existing product for update).   
7. The system updates the inventory with the new or modified product information.   
8. The system displays a confirmation message that the inventory has been successfully updated.   
  
Alternative Flow:   
1. If the input data is invalid or incomplete, the system displays an error message and prompts the administrator to correct the data.   
2. If the selected product does not exist in the inventory, the system displays an error message and prevents the update.   
3. If the system fails to update the inventory, it displays an error message and logs the issue for troubleshooting.   
4. If the administrator cancels the action, the system returns to the inventory management page without making any changes.  
  
Use Case Name: Categorize Products   
Use Case ID: UC-12   
Actors: Administrator, System, Product, Category   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The system is operational and accessible.   
3. The product data exists in the database.   
4. The category list is available and up to date.   
  
Postconditions:   
1. The product is successfully assigned to one or more categories.   
2. The system updates the product database with the new category information.   
3. The product appears in the correct category view for customers.   
4. The system logs the categorization action for audit and tracking purposes.   
  
Main Flow:   
1. The administrator navigates to the "Product Management" section of the system.   
2. The system displays a list of products along with their current category assignments.   
3. The administrator selects a product to edit its categories.   
4. The system shows a list of available categories and the current category of the product.   
5. The administrator chooses the appropriate category or categories for the product.   
6. The administrator submits the changes.   
7. The system validates the selected category and ensures it exists in the database.   
8. The system updates the product's category information in the database.   
9. The system displays a confirmation message that the product has been successfully categorized.   
  
Alternative Flow:   
1. If the selected category does not exist, the system displays an error message and prompts the administrator to select a valid category.   
2. If the system fails to update the product's category, it displays an error message and logs the issue for troubleshooting.   
3. If the administrator cancels the categorization process, the system returns to the product management page without applying any changes.   
4. If the product has already been assigned to the selected category, the system displays a message indicating no change is needed.  
  
Use Case Name: Update Product Details   
Use Case ID: UC-13   
Actors: Administrator, System, Product, Inventory   
  
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 product database.   
4. The administrator has the necessary permissions to modify product details.   
  
Postconditions:   
1. The product details (e.g., name, price, description, or specifications) are successfully updated in the database.   
2. The system logs the update action for audit and tracking.   
3. The updated product information is reflected in the product catalog and inventory.   
4. If applicable, the product's availability is updated in the inventory.   
  
Main Flow:   
1. The administrator navigates to the "Product Management" section of the system.   
2. The system displays a list of products along with their current details.   
3. The administrator selects a product to edit.   
4. The system opens the product details form, pre-filled with the current information.   
5. The administrator modifies the required fields (e.g., price, description, stock quantity, or specifications).   
6. The administrator submits the updated product details.   
7. The system validates the input data (e.g., correct price format, valid stock quantity).   
8. The system updates the product database with the new information.   
9. The system displays a confirmation message that the product details have been successfully updated.   
  
Alternative Flow:   
1. If the product does not exist in the database, the system displays an error message and prevents the update.   
2. If the input data is invalid (e.g., negative price, non-numeric quantity), the system displays an error message and prompts the administrator to correct the data.   
3. If the system fails to update the product database, it displays an error message and logs the issue for troubleshooting.   
4. If the administrator cancels the update process, the system returns to the product management page without applying any changes.  
  
Use Case Name: Admin Login   
Use Case ID: UC-14   
Actors: Administrator, System   
Preconditions:   
1. The administrator has a valid account registered in the system.   
2. The system is operational and accessible.   
3. The administrator is not currently logged in.   
  
Postconditions:   
1. The administrator is successfully logged into their account.   
2. The system displays the administrator's dashboard or control panel.   
3. The system records the login event in the administrator activity log.   
  
Main Flow:   
1. The administrator navigates to the admin login page.   
2. The system displays the login form with fields for email and password.   
3. The administrator enters their registered email and password.   
4. The administrator submits the login form.   
5. The system verifies the email and password against the admin database.   
6. If valid, the system logs in the administrator and redirects them to the admin dashboard.   
7. The system displays a welcome message and updates the session status for the administrator.   
  
Alternative Flow:   
1. If the email is not registered as an administrator, the system displays an error message.   
2. If the password is incorrect, the system displays an error message and allows the administrator to retry.   
3. If the system fails to authenticate the administrator, it displays an error message and logs the failed attempt for security monitoring.   
4. If the administrator account is locked or disabled, the system displays a message and prevents login.  
  
Use Case Name: Admin Logout   
Use Case ID: UC-15   
Actors: Administrator, System   
Preconditions:   
1. The administrator is currently logged in.   
2. The system is operational and accessible.   
  
Postconditions:   
1. The administrator's session is terminated.   
2. The administrator is redirected to the admin login page or homepage.   
3. The system records the logout event in the administrator activity log.   
  
Main Flow:   
1. The administrator navigates to the logout option in the admin interface.   
2. The system prompts the administrator to confirm the logout action.   
3. The administrator confirms the logout.   
4. The system terminates the active admin session and clears session data.   
5. The system redirects the administrator to the admin login page or homepage.   
6. The system logs the logout event for audit and tracking purposes.   
  
Alternative Flow:   
1. If the administrator cancels the logout confirmation, the system returns to the previous page without any changes.   
2. If the system fails to terminate the session, it displays an error message and logs the issue for troubleshooting.  
  
Use Case Name: View Purchase History   
Use Case ID: UC-16   
Actors: Customer, System, Purchase History, Account   
  
Preconditions:   
1. The customer is logged into the system.   
2. The customer has at least one completed purchase recorded in the system.   
3. The system is operational and accessible.   
4. The purchase history is stored in the database.   
  
Postconditions:   
1. The customer views their purchase history with details of previous orders.   
2. The system logs the access to the purchase history for audit purposes.   
3. The customer can filter or search their purchase history based on criteria such as date or product.   
  
Main Flow:   
1. The customer navigates to the "Purchase History" section from the account dashboard.   
2. The system retrieves the customer's purchase history from the database.   
3. The system displays the list of previous orders, including order ID, date, total cost, and status.   
4. The customer selects an order to view more details.   
5. The system shows the order details, including product list, quantities, and payment method.   
6. The system updates the activity log to reflect the purchase history view.   
  
Alternative Flow:   
1. If the customer has no previous orders, the system displays a message indicating there is no purchase history available.   
2. If the system fails to retrieve the purchase history, it displays an error message and logs the issue for troubleshooting.   
3. If the selected order does not exist or is invalid, the system displays an error message and returns to the purchase history list.   
4. If the customer tries to access another user's purchase history, the system denies access and logs the unauthorized attempt for security monitoring.  
  
Use Case Name: Manage Payment Information   
Use Case ID: UC-17   
Actors: Customer, System, Payment, Account   
  
Preconditions:   
1. The customer is logged into the system.   
2. The system is operational and accessible.   
3. The customer has an existing account with the system.   
4. The customer has navigated to the payment management section of their account.   
  
Postconditions:   
1. The customer's payment information is successfully added, updated, or removed.   
2. The system updates the customer's account with the new payment information.   
3. The system verifies the validity of the payment method (if applicable).   
4. The system logs the payment information management action for audit and tracking.   
  
Main Flow:   
1. The customer navigates to the "Payment Information" section of their account.   
2. The system displays the current payment methods (if any) and provides options to add, edit, or delete payment information.   
3. The customer selects an action (e.g., "Add New Payment Method").   
4. The system displays a form for entering payment details (e.g., card number, expiration date, CVV, billing address).   
5. The customer fills in the required information and submits the form.   
6. The system validates the input data (e.g., correct card number format, valid expiration date).   
7. The system securely stores the payment information in the customer's account.   
8. The system displays a confirmation message that the payment method has been successfully managed.   
  
Alternative Flow:   
1. If the input data is invalid (e.g., incorrect card number, expired date), the system displays an error message and prompts the customer to correct the information.   
2. If the customer attempts to delete a payment method, the system prompts for confirmation and performs the action upon approval.   
3. If the system fails to store the payment information, it displays an error message and logs the issue for troubleshooting.   
4. If the customer cancels the action, the system returns to the payment information page without making any changes.   
5. If the customer selects an existing payment method to update, the system pre-fills the form with current data for easy modification.  
  
Use Case Name: Install Plugin   
Use Case ID: UC-18   
Actors: Administrator, System, Plugin, API   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The system is operational and accessible.   
3. The plugin is available for installation via the system's API or plugin repository.   
4. The administrator has the necessary permissions to install plugins.   
  
Postconditions:   
1. The plugin is successfully installed and activated in the system.   
2. The system updates the plugin list to reflect the new installation.   
3. The system logs the plugin installation event for audit and tracking.   
4. The plugin's functionality is available to the system and users as configured.   
  
Main Flow:   
1. The administrator navigates to the "Plugin Management" section of the system.   
2. The system displays a list of available plugins and their descriptions.   
3. The administrator selects a plugin to install.   
4. The system prompts the administrator to confirm the installation.   
5. The administrator confirms the installation.   
6. The system retrieves the plugin from the API or repository.   
7. The system installs and activates the plugin.   
8. The system updates the plugin list and displays a success message.   
  
Alternative Flow:   
1. If the selected plugin is already installed, the system displays a message indicating no action is needed.   
2. If the system fails to retrieve the plugin, it displays an error message and logs the issue for troubleshooting.   
3. If the plugin installation fails due to compatibility or configuration issues, the system displays an error message and allows the administrator to review or cancel the process.   
4. If the administrator cancels the installation confirmation, the system returns to the plugin management page without changes.   
5. If the system fails to activate the plugin after installation, it displays an error message and logs the failure for further investigation.  
  
Use Case Name: Access Plugin API Documentation   
Use Case ID: UC-19   
Actors: Administrator, System, Plugin, API   
Preconditions:   
1. The administrator is logged into the system.   
2. The system is operational and accessible.   
3. The plugin exists in the system and has associated API documentation.   
4. The administrator has navigated to the plugin management or API documentation section.   
  
Postconditions:   
1. The administrator views the plugin's API documentation.   
2. The system records the access to the API documentation in the activity log.   
3. The administrator can search, filter, or navigate through the documentation as needed.   
  
Main Flow:   
1. The administrator navigates to the "Plugin API Documentation" section in the system.   
2. The system displays a list of plugins with available API documentation.   
3. The administrator selects a plugin to view its documentation.   
4. The system retrieves and displays the API documentation, including endpoints, parameters, and usage examples.   
5. The administrator can search for specific API functions or sections within the documentation.   
6. The system updates the activity log to reflect the documentation access.   
  
Alternative Flow:   
1. If the selected plugin has no API documentation available, the system displays a message indicating the documentation is missing.   
2. If the system fails to retrieve the documentation, it displays an error message and logs the issue for troubleshooting.   
3. If the administrator cancels or navigates away from the documentation page, the system returns to the plugin management page without further action.  
  
Use Case Name: Process Transaction   
Use Case ID: UC-20   
Actors: Customer, System, Payment, Inventory   
  
Preconditions:   
1. The customer is logged into the system.   
2. The customer has at least one product in their shopping cart.   
3. The system has validated the cart contents against the inventory.   
4. The customer has selected a valid payment method.   
  
Postconditions:   
1. The transaction is successfully processed and recorded in the system.   
2. The inventory is updated to reflect the reduction in product stock.   
3. The customer receives a confirmation message or email.   
4. The payment is processed, and the transaction is completed.   
5. The system generates an order confirmation document.   
  
Main Flow:   
1. The customer navigates to the checkout page.   
2. The system displays the cart contents, total cost, and selected payment method for review.   
3. The customer confirms the order and clicks the "Process Transaction" button.   
4. The system verifies the availability of all items in the cart.   
5. The system initiates the payment process using the selected payment method.   
6. The system waits for confirmation of the payment success.   
7. If the payment is successful, the system reduces the product quantities in the inventory.   
8. The system generates an order confirmation document and sends it to the customer via email.   
9. The system displays a success message to the customer, including the order confirmation number.   
  
Alternative Flow:   
1. If any product is out of stock, the system displays an error message and prevents the transaction.   
2. If the payment fails, the system displays an error message and logs the failed transaction.   
3. If the system fails to send the confirmation email, it displays a message to the customer, offering to retry or contact support.   
4. If the customer cancels the transaction, the system returns to the cart page without processing the order.   
5. If the system encounters an error during transaction processing, it displays an error message and logs the issue for troubleshooting.  
  
Use Case Name: Send Email Notification   
Use Case ID: UC-21   
Actors: System, Email, Customer   
  
Preconditions:   
1. The system has an event or action that requires sending an email notification (e.g., order confirmation, account activation).   
2. The customer has provided a valid email address in their account or during the transaction.   
3. The email service is configured and operational.   
4. The system has the necessary information to compose the email (e.g., message content, subject, recipient).   
  
Postconditions:   
1. The email notification is successfully sent to the customer.   
2. The system logs the email sending event.   
3. The customer receives the notification and can take further action if required.   
  
Main Flow:   
1. The system detects an event that requires an email notification (e.g., order confirmation, password reset).   
2. The system retrieves the customer's email address from the account database.   
3. The system generates the email content based on the event and includes relevant details (e.g., order ID, transaction status, instructions).   
4. The system sends the email to the customer's registered email address via the configured email service.   
5. The system logs the email notification event in the system log.   
6. The system displays a success message to the user (if applicable) indicating the email was sent.   
  
Alternative Flow:   
1. If the customer's email address is invalid or missing, the system logs the error and displays a message to the user requesting them to update their email information.   
2. If the email service is unavailable, the system logs the failure and displays an error message to the user, offering to retry or contact support.   
3. If the system fails to generate the email content, it logs the error and displays a message to the user indicating the issue.   
4. If the email is sent but not delivered (e.g., due to spam filters), the system logs the event and provides an option for the user to manually re-send the email.  
  
Use Case Name: View User Activity Log   
Use Case ID: UC-22   
Actors: Administrator, System, User Activity Log   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The system is operational and accessible.   
3. The user activity log exists in the database and contains valid entries.   
4. The administrator has the necessary permissions to access the activity log.   
  
Postconditions:   
1. The administrator views the user activity log with details of user actions.   
2. The system records the access to the activity log for audit purposes.   
3. The administrator can filter or search the log based on criteria such as date, user, or action type.   
  
Main Flow:   
1. The administrator navigates to the "User Activity Log" section in the admin dashboard.   
2. The system retrieves the user activity log entries from the database.   
3. The system displays the log entries in a structured format, including timestamp, user ID, action performed, and additional details.   
4. The administrator can apply filters (e.g., by date range or specific user).   
5. The system updates the displayed log based on the filter criteria.   
6. The system logs the access to the activity log for audit and tracking.   
  
Alternative Flow:   
1. If the activity log is empty or no entries match the filter criteria, the system displays a message indicating no data found.   
2. If the system fails to retrieve the activity log, it displays an error message and logs the issue for troubleshooting.   
3. If the administrator tries to access the activity log without proper permissions, the system denies access and logs the unauthorized attempt.   
4. If the selected user does not exist, the system displays an error message and prevents further filtering.