项目文档

# Functional Requirement

1.1 Customer Registration Function   
Function ID: FR-01   
Description: Customers can register a new account by providing personal details such as name, email, and password. The system validates the email and password, and sends a confirmation email to activate the account.   
Input: Name, email address, password, confirmation of terms and conditions.   
Output: A new customer account is created and stored in the system. The customer receives an email confirmation and a success message after clicking the confirmation link.  
  
1.2 Customer Login Function   
Function ID: FR-02   
Description: Customers can log in to their account by entering their registered email and password. The system validates the credentials, logs in the customer, and redirects them to their account dashboard.   
Input: Registered email address, password.   
Output: The customer is successfully logged in, and the login activity is recorded. The customer is redirected to the account dashboard.  
  
1.3 View Product List Function   
Function ID: FR-03   
Description: Customers can view a list of available products, including name, price, description, and category. The system supports filtering and sorting options.   
Input: User request to view products, optional filter or sort criteria.   
Output: A displayed list of products with relevant details. The system logs the access to the product list.  
  
1.4 Add Product to Cart Function   
Function ID: FR-04   
Description: Customers can add a selected product to their shopping cart. The system checks the product’s availability and updates the cart accordingly.   
Input: Selected product, quantity (optional).   
Output: The product is added to the shopping cart. The cart display is updated, and a confirmation message is shown.  
  
1.5 Update Cart Contents Function   
Function ID: FR-05   
Description: Customers can modify the quantity of items in their shopping cart or remove items. The system recalculates the total cost and saves the updated cart.   
Input: Cart item selection, new quantity or removal request.   
Output: Updated cart with modified items. The system recalculates and displays the total cost, and saves the cart state.  
  
1.6 Remove Product from Cart Function   
Function ID: FR-06   
Description: Customers can remove a selected product from their shopping cart. The system updates the cart and displays the new total.   
Input: Selected product to remove.   
Output: The product is removed from the cart. The system updates the cart and displays the new total.  
  
1.7 View Cart Summary Function   
Function ID: FR-07   
Description: Customers can view a summary of their shopping cart, including all items, their quantities, prices, and the total amount.   
Input: User request to view cart summary.   
Output: Cart summary displayed with items and total amount. The system logs the access to the cart summary.  
  
1.8 Checkout Function   
Function ID: FR-08   
Description: Customers can proceed to checkout with their selected items. The system collects shipping details, processes the selected payment method, and creates a new order.   
Input: Shopping cart contents, shipping address, contact information, selected payment method.   
Output: A new order is created and processed. The shopping cart is cleared, and the customer receives an email confirmation.  
  
1.9 Confirm Order Function   
Function ID: FR-09   
Description: After a successful payment, the customer can confirm their order. The system updates the order status and sends a confirmation email.   
Input: User request to confirm order.   
Output: Order status is updated to "confirmed", and the customer receives an email with the order summary and expected delivery date.  
  
1.10 Receive Order Confirmation Email Function   
Function ID: FR-10   
Description: After the order is confirmed, the customer receives an order confirmation email containing the order details and delivery information.   
Input: Order confirmation event.   
Output: Order confirmation email sent to the customer. The system logs the email delivery.  
  
1.11 Administrator Login Function   
Function ID: FR-11   
Description: Administrators can log in to the system using their username and password. The system validates the credentials and redirects them to the admin dashboard.   
Input: Administrator username, password.   
Output: Administrator is successfully logged in and redirected to the admin dashboard. The login activity is recorded.  
  
1.12 Manage Inventory Function   
Function ID: FR-12   
Description: Administrators can manage the inventory by adding, updating, or removing products. The system validates input data and updates the inventory database.   
Input: Product information (e.g., name, price, stock quantity), action (add, update, remove).   
Output: Inventory database is updated with the new or modified product data. The system logs the action and displays a confirmation message.  
  
1.13 Update Product Details Function   
Function ID: FR-13   
Description: Administrators can update the details of an existing product, including name, price, description, category, and stock quantity.   
Input: Product ID, updated product information.   
Output: Product record in the database is updated. Inventory is adjusted if necessary. The system logs the action.  
  
1.14 Categorize Products Function   
Function ID: FR-14   
Description: Administrators can assign products to specific categories to improve product organization and searchability.   
Input: Product ID, selected category.   
Output: Product is updated with the new category. The system logs the categorization action.  
  
1.15 Create Category Function   
Function ID: FR-15   
Description: Administrators can create new categories to organize products. The system validates the category name and prevents duplicates.   
Input: Category name, description.   
Output: New category is created and stored in the system. The system logs the action and displays a confirmation message.  
  
1.16 Delete Category Function   
Function ID: FR-16   
Description: Administrators can delete an existing category if it is not assigned to any active products.   
Input: Category ID, confirmation to delete.   
Output: Category is removed from the system. The system logs the deletion and displays a confirmation message.  
  
1.17 Manage Plugin Function   
Function ID: FR-17   
Description: Administrators can manage plugins by installing, updating, deactivating, or deleting them. The system validates compatibility and logs the action.   
Input: Plugin name, version, configuration details, action (install, update, deactivate, delete).   
Output: Plugin status is updated in the system. The system logs the action and displays a confirmation message.  
  
1.18 View Payment Information Function   
Function ID: FR-18   
Description: Customers can view their saved payment methods, including card type, last four digits, expiration date, and default status.   
Input: User request to view payment information.   
Output: Saved payment methods are displayed. The system logs the access.  
  
1.19 Manage Customer Function   
Function ID: FR-19   
Description: Administrators can manage customer accounts by editing details, blocking accounts, or deleting accounts. The system validates the selected action and logs it.   
Input: Customer ID, action (edit, block, delete), updated information if applicable.   
Output: Customer account is modified or removed. The system logs the action and displays a confirmation message.  
  
1.20 Manage Product Function   
Function ID: FR-20   
Description: Administrators can add new products, update existing products, or delete products from the system. The system validates input data and updates the inventory if necessary.   
Input: Product information (e.g., name, price, description, category, stock quantity), action (add, update, delete).   
Output: Product database is updated. Inventory is adjusted if needed. The system logs the action and displays a confirmation message.  
  
1.21 Manage Order Function   
Function ID: FR-21   
Description: Administrators can manage orders by updating their status, modifying order details, or deleting orders. The system validates the action and sends notifications to customers if applicable.   
Input: Order ID, action (update status, modify details, delete), new status or updated information.   
Output: Order database is updated. Inventory is adjusted if items are modified or removed. The system logs the action and notifies the customer if needed.

# External Description

2. External Interfaces   
The system interacts with various external interfaces to fulfill its functional requirements. These interfaces are categorized into user interfaces, hardware interfaces, software interfaces, and communication interfaces.   
  
2.1 User Interface Output   
User interfaces are the primary means of interaction between the system and its users (customers and administrators). The system provides a responsive and intuitive web-based interface for all user-facing functions.   
  
- \*\*Customer Account Management Interface\*\*:   
 This interface allows customers to register, log in, and manage their account. It includes input fields for name, email, password, and terms of service acceptance. The output includes success/failure messages, confirmation links, and redirection to the account dashboard.   
  
- \*\*Product Browsing and Filtering Interface\*\*:   
 Customers can view a list of products with details such as name, price, description, and category. The interface supports sorting and filtering by category, price, and other attributes. It also displays a log of product list accesses.   
  
- \*\*Shopping Cart Interface\*\*:   
 The interface allows customers to add, update, or remove items from their shopping cart. It displays the items, quantities, prices, and total cost in real time. The cart is updated dynamically based on user actions, and a confirmation message is shown for each change.   
  
- \*\*Checkout Interface\*\*:   
 This interface collects the customer's shipping details, contact information, and selected payment method. It provides a summary of the order and processes the checkout. After a successful transaction, the cart is cleared and a confirmation message is displayed.   
  
- \*\*Order Confirmation Interface\*\*:   
 Customers can confirm their order after payment. The interface provides a summary of the order, delivery date, and confirmation status. The system updates the order status and sends a confirmation email.   
  
- \*\*Admin Dashboard Interface\*\*:   
 Administrators access this interface to manage inventory, products, categories, plugins, and customer accounts. It includes input forms for product and category details, action buttons for managing plugins and orders, and displays confirmation messages after each action.   
  
- \*\*Payment Method Management Interface\*\*:   
 Customers can view their saved payment methods, including card type, last four digits, expiration date, and default status. This interface is used to display and manage payment information securely.   
  
2.2 Hardware Interface Output   
The system does not directly interact with any hardware devices. Therefore, no hardware interfaces are required for this system.   
  
2.3 Software Interface Output   
The system interacts with several software components, including databases, third-party APIs, and internal modules.   
  
- \*\*Customer Database\*\*:   
 The system stores and retrieves customer data, including name, email, password, and account status. It validates email and password inputs and updates the database upon successful registration or login. The database also tracks login activity and customer account modifications.   
  
- \*\*Product Inventory Database\*\*:   
 This database holds product information such as name, price, description, category, and stock quantity. Administrators update or delete products, and the system checks product availability when adding to the cart. The database is also updated when products are added, modified, or removed.   
  
- \*\*Order Management Database\*\*:   
 The system records all order details, including customer information, product list, quantities, shipping address, payment method, and order status. It updates the order status when a customer confirms their order or when an administrator modifies it. The database also logs order-related activities and adjusts inventory accordingly.   
  
- \*\*Email Service API\*\*:   
 The system uses an external email service API to send confirmation emails to customers after registration, checkout, and order confirmation. The API receives the customer's email address and the content of the message, then sends it and logs the delivery status.   
  
- \*\*Payment Processing API\*\*:   
 During checkout, the system interacts with a payment processing API to validate and process the customer's selected payment method. The API receives payment information and returns a success or failure response. This interaction is essential for completing the checkout process.   
  
- \*\*Plugin Management Module\*\*:   
 The system includes a plugin management module that allows administrators to install, update, deactivate, or delete plugins. The module validates plugin compatibility and updates the system accordingly. It also logs each plugin-related action.   
  
2.4 Communication Interface Output   
The system communicates with external systems and users through various communication channels.   
  
- \*\*Email Communication\*\*:   
 The system sends emails to customers for account confirmation, order confirmation, and delivery updates. The email communication is triggered by specific events, such as customer registration or order completion. The system logs all email deliveries for tracking and auditing purposes.   
  
- \*\*Web Browsing (HTTP/HTTPS)\*\*:   
 Customers and administrators access the system through web browsers using HTTP/HTTPS protocols. This communication allows them to interact with the system's user interface and perform actions such as logging in, browsing products, and managing orders.   
  
- \*\*API Communication (RESTful)\*\*:   
 The system communicates with external services like the email service and payment processing service using RESTful APIs. These APIs are accessed via HTTP/HTTPS and follow standard JSON-based request and response formats.   
  
- \*\*Internal System Logs\*\*:   
 The system generates internal logs for all user activities and system events, such as login attempts, cart modifications, order status changes, and plugin actions. These logs are stored in a centralized logging system for auditing and troubleshooting.   
  
By clearly defining these external interfaces, the system ensures seamless interaction with users, databases, APIs, and communication channels, enabling efficient and reliable operation.

# Use Case

Use Case Name: Customer Registration   
Use Case ID: UC-01   
Actors: Customer, System   
Preconditions: The system is accessible to the customer. The customer does not have an existing account.   
Postconditions: A new customer account is successfully created and stored in the system. The customer receives a confirmation email.   
  
Main Flow:   
1. The customer navigates to the registration page on the system.   
2. The customer enters personal details (e.g., name, email, password).   
3. The system validates the email format and checks if the email is already registered.   
4. The system confirms the password strength.   
5. The customer agrees to the terms and conditions.   
6. The system creates a new customer account and stores it in the database.   
7. The system sends an email confirmation to the customer.   
8. The customer clicks the confirmation link in the email.   
9. The system verifies the confirmation link and updates the account status to "active".   
10. The system displays a registration success message to the customer.   
  
Alternative Flow:   
1. If the email is already registered, the system displays an error message and prompts the customer to use a different email.   
2. If the password is not strong enough, the system provides feedback and asks the customer to choose a stronger password.   
3. If the customer does not agree to the terms and conditions, the system prevents account creation and displays a message.   
4. If the email confirmation fails or is not clicked within the specified time, the system marks the account as inactive and prompts the customer to request a new confirmation email.  
  
Use Case Name: Customer Login   
Use Case ID: UC-02   
Actors: Customer, System   
Preconditions: The customer has an active account. The system is accessible. The customer is not currently logged in.   
Postconditions: The customer is successfully logged in and redirected to the account dashboard. The system records the login activity.   
  
Main Flow:   
1. The customer navigates to the login page of the system.   
2. The customer enters their registered email and password.   
3. The system validates the input format of the email and password.   
4. The system checks the database for a matching email and password.   
5. If a match is found, the system logs in the customer and redirects them to the account dashboard.   
6. The system displays a welcome message and updates the login timestamp in the customer record.   
  
Alternative Flow:   
1. If the email format is invalid, the system displays an error message and prompts the customer to correct it.   
2. If the password is incorrect, the system displays an error message and allows the customer to retry.   
3. If the account is inactive, the system displays a message and prompts the customer to check their email for a confirmation link.   
4. If the customer exceeds the allowed number of login attempts, the system locks the account temporarily and notifies the customer.   
5. If the customer forgets their password, they can click "Forgot Password" to initiate the password recovery process.  
  
Use Case Name: View Product List   
Use Case ID: UC-03   
Actors: Customer, System   
Preconditions: The system is accessible to the customer. The customer is logged in or not logged in. The product database is available.   
Postconditions: The customer is shown a list of available products. The system logs the customer's access to the product list.   
  
Main Flow:   
1. The customer navigates to the product listing section of the system.   
2. The system retrieves the list of products from the database.   
3. The system displays the products, including details such as name, price, description, and category.   
4. The customer can filter or sort the product list using available options.   
5. The system updates the displayed product list based on the customer's selection.   
  
Alternative Flow:   
1. If no products are available, the system displays a message indicating the product list is empty.   
2. If the database is unavailable, the system displays an error message and prompts the customer to try again later.   
3. If the customer selects an invalid filter or sort option, the system ignores the invalid input and displays the default product list.  
  
Use Case Name: Add Product to Cart   
Use Case ID: UC-04   
Actors: Customer, System, Shopping Cart   
Preconditions: The customer is logged in or browsing as a guest. The product is available in the inventory. The shopping cart is accessible.   
Postconditions: The selected product is added to the customer's shopping cart. The cart is updated in the system. The customer is notified of the addition.   
  
Main Flow:   
1. The customer views a product in the product list.   
2. The customer selects the product to add to the cart.   
3. The system checks the product's availability in the inventory.   
4. The system adds the product to the customer's shopping cart.   
5. The system updates the cart display to reflect the new item.   
6. The system logs the addition of the product to the cart.   
7. The system displays a confirmation message to the customer that the product was added.   
  
Alternative Flow:   
1. If the product is out of stock, the system displays a message and does not add the product to the cart.   
2. If the customer attempts to add more quantity than available, the system displays an error and limits the addition to the available stock.   
3. If the customer is not logged in, the system stores the cart in a session or prompts for login to persist the cart.   
4. If the cart is not accessible or encounters an error, the system displays an error message and suggests the customer try again.  
  
Use Case Name: Update Cart Contents   
Use Case ID: UC-05   
Actors: Customer, System, Shopping Cart   
Preconditions: The customer is logged in or browsing as a guest. The shopping cart is accessible. The product to be updated is already in the cart.   
Postconditions: The cart contents are modified (e.g., quantity changed, item removed). The updated cart is saved in the system. The customer is notified of the changes.   
  
Main Flow:   
1. The customer navigates to the shopping cart page.   
2. The system displays the current cart contents, including product name, quantity, and price.   
3. The customer selects an item and modifies the quantity or removes it from the cart.   
4. The system updates the cart in real-time based on the customer's changes.   
5. The system recalculates the total price of the cart and displays the updated total.   
6. The system saves the updated cart information in the database or session.   
7. The system provides a confirmation message to the customer about the cart update.   
  
Alternative Flow:   
1. If the customer tries to set a quantity higher than available stock, the system displays an error and adjusts the quantity to the maximum available.   
2. If the cart update fails due to a system error, the system displays an error message and prompts the customer to try again.   
3. If the customer removes the last item from the cart, the system displays a message indicating the cart is now empty.   
4. If the cart session expires, the system prompts the customer to log in or restart the shopping process.  
  
Use Case Name: Remove Product from Cart   
Use Case ID: UC-06   
Actors: Customer, System, Shopping Cart   
Preconditions: The customer is logged in or browsing as a guest. The shopping cart is accessible and contains at least one product.   
Postconditions: The selected product is removed from the shopping cart. The cart is updated in the system. The customer is notified of the removal.   
  
Main Flow:   
1. The customer navigates to the shopping cart page.   
2. The system displays the current cart contents, including product name, quantity, and price.   
3. The customer selects a product to remove from the cart.   
4. The system confirms the product removal and updates the cart in real-time.   
5. The system recalculates the total price of the cart and displays the updated total.   
6. The system saves the updated cart information in the database or session.   
7. The system provides a confirmation message to the customer that the product was removed.   
  
Alternative Flow:   
1. If the customer removes the last item from the cart, the system displays a message indicating the cart is now empty.   
2. If the cart is not accessible or encounters an error during removal, the system displays an error message and suggests the customer try again.   
3. If the customer cancels the removal, the system retains the original cart contents and does not make any changes.  
  
Use Case Name: View Cart Summary   
Use Case ID: UC-07   
Actors: Customer, System, Shopping Cart   
Preconditions: The customer is logged in or browsing as a guest. The shopping cart is accessible and contains at least one item.   
Postconditions: The customer is shown a summary of their shopping cart. The system logs the customer's access to the cart summary.   
  
Main Flow:   
1. The customer navigates to the shopping cart summary page.   
2. The system retrieves the current cart contents from the database or session.   
3. The system displays the cart summary, including product names, quantities, individual prices, and the total amount.   
4. The system allows the customer to review the items and proceed to checkout or continue shopping.   
  
Alternative Flow:   
1. If the cart is empty, the system displays a message indicating that no items are in the cart.   
2. If the cart data is not accessible or encounters an error, the system displays an error message and prompts the customer to try again.   
3. If the customer selects an option to update the cart, the system redirects to the "Update Cart Contents" use case.   
4. If the customer selects an option to remove an item, the system redirects to the "Remove Product from Cart" use case.  
  
Use Case Name: Checkout   
Use Case ID: UC-08   
Actors: Customer, System, Shopping Cart, Order, Payment   
Preconditions: The customer is logged in or browsing as a guest. The shopping cart is accessible and contains at least one item. The payment system is operational.   
Postconditions: The customer's order is created and processed. The payment is completed. The shopping cart is cleared. The customer receives an email confirmation.   
  
Main Flow:   
1. The customer navigates to the checkout page.   
2. The system displays the cart summary and prompts the customer to enter shipping details.   
3. The customer provides shipping address and contact information.   
4. The system validates the shipping details and prompts the customer to choose a payment method.   
5. The customer selects a payment method (e.g., credit card, PayPal).   
6. The system processes the payment transaction.   
7. If the payment is successful, the system creates a new order and records the transaction details.   
8. The system clears the shopping cart.   
9. The system sends an email confirmation to the customer with the order details.   
10. The system displays a confirmation message to the customer that the order was successfully placed.   
  
Alternative Flow:   
1. If the customer does not provide valid shipping details, the system displays an error and prompts the customer to correct the information.   
2. If the payment method is invalid or the transaction fails, the system displays an error and allows the customer to retry or choose an alternative payment method.   
3. If the inventory is updated during checkout (e.g., item becomes unavailable), the system displays a message and allows the customer to adjust the cart or proceed with available items.   
4. If the email confirmation fails to send, the system displays a message and allows the customer to manually request a confirmation email.   
5. If the customer cancels the checkout process, the system retains the cart contents and returns to the cart summary page.  
  
Use Case Name: Confirm Order   
Use Case ID: UC-09   
Actors: Customer, System, Shopping Cart, Order, Payment, Email Confirmation   
Preconditions: The customer has completed the checkout process. The order details are available. The payment has been successfully processed. The system is accessible.   
Postconditions: The order is confirmed and finalized. The customer receives an email confirmation. The order status is updated to "confirmed".   
  
Main Flow:   
1. The system displays the order confirmation page after the payment is successfully processed.   
2. The system lists the order details, including items, quantities, total price, and shipping address.   
3. The customer reviews the order details to ensure accuracy.   
4. The customer clicks the "Confirm Order" button.   
5. The system updates the order status to "confirmed" and records the confirmation timestamp.   
6. The system sends an email confirmation to the customer with the order summary and expected delivery date.   
7. The system displays a success message to the customer indicating that the order has been confirmed.   
  
Alternative Flow:   
1. If the customer clicks "Cancel" instead of "Confirm Order", the system reverts to the cart summary page without finalizing the order.   
2. If the email confirmation fails to send, the system displays an error message and allows the customer to manually request a confirmation email.   
3. If the system is unable to update the order status due to an error, the system displays an error message and prompts the customer to contact support.  
  
Use Case Name: Receive Order Confirmation Email   
Use Case ID: UC-10   
Actors: Customer, System, Email Confirmation   
Preconditions: The customer has completed the checkout process. The order has been confirmed. The email system is operational.   
Postconditions: The customer receives an order confirmation email. The email contains the order summary and delivery details.   
  
Main Flow:   
1. The system generates an order confirmation message after the order is finalized.   
2. The system retrieves the customer's email address from their account.   
3. The system sends the confirmation email to the customer.   
4. The email includes the order number, items purchased, total amount, and expected delivery date.   
5. The customer receives the email in their inbox.   
6. The system logs the successful email delivery.   
  
Alternative Flow:   
1. If the customer's email address is invalid or missing, the system displays an error and prompts the customer to update their contact information.   
2. If the email confirmation fails to send, the system displays an error and allows the customer to manually request the email to be resent.   
3. If the customer does not receive the email, they can check their spam folder or contact customer support to resolve the issue.  
  
Use Case Name: Administrator Login   
Use Case ID: UC-11   
Actors: Administrator, System   
Preconditions: The system is accessible to the administrator. The administrator has a valid account. The administrator is not currently logged in.   
Postconditions: The administrator is successfully logged in and redirected to the administrator dashboard. The system records the login activity.   
  
Main Flow:   
1. The administrator navigates to the admin login page of the system.   
2. The administrator enters their username and password.   
3. The system validates the input format of the username and password.   
4. The system checks the database for a matching username and password with administrative privileges.   
5. If a match is found, the system logs in the administrator and redirects them to the admin dashboard.   
6. The system updates the login timestamp in the administrator's record.   
7. The system displays a welcome message to the administrator.   
  
Alternative Flow:   
1. If the username or password format is invalid, the system displays an error message and prompts the administrator to correct it.   
2. If the password is incorrect, the system displays an error message and allows the administrator to retry.   
3. If the administrator account is locked or disabled, the system displays an error message and prevents login.   
4. If the administrator exceeds the allowed number of login attempts, the system locks the account temporarily and notifies the administrator.   
5. If the administrator forgets their password, they can click "Forgot Password" to initiate the password recovery process.  
  
Use Case Name: Manage Inventory   
Use Case ID: UC-12   
Actors: Administrator, System, Inventory   
Preconditions: The administrator is logged in. The system is accessible. The inventory database is available.   
Postconditions: The inventory is updated (e.g., products added, removed, or modified). The system logs the administrator's inventory management actions.   
  
Main Flow:   
1. The administrator navigates to the inventory management section of the system.   
2. The system displays the current inventory, including product names, quantities, and statuses.   
3. The administrator selects an action (e.g., add new product, update stock, remove product).   
4. The system prompts the administrator to provide the necessary details for the selected action.   
5. The administrator enters the product information or modifies the existing data.   
6. The system validates the input data (e.g., correct format, positive stock quantity).   
7. The system updates the inventory database with the new or modified information.   
8. The system logs the action taken by the administrator.   
9. The system displays a confirmation message that the inventory was successfully updated.   
  
Alternative Flow:   
1. If the input data is invalid, the system displays an error message and prompts the administrator to correct the input.   
2. If the product already exists and the administrator tries to add it again, the system displays a message and allows the administrator to update the existing product instead.   
3. If the inventory database is unavailable, the system displays an error message and prompts the administrator to try again later.   
4. If the administrator cancels the action, the system does not modify the inventory and returns to the inventory management page.  
  
Use Case Name: Update Product Details   
Use Case ID: UC-13   
Actors: Administrator, System, Product, Inventory   
Preconditions: The administrator is logged in. The system is accessible. The product to be updated exists in the inventory.   
Postconditions: The product details are updated in the system. The inventory is updated if necessary. The system logs the administrator's action.   
  
Main Flow:   
1. The administrator navigates to the product management section of the system.   
2. The system displays a list of available products with their details.   
3. The administrator selects a product to update.   
4. The system opens the product details form for editing.   
5. The administrator modifies the product information (e.g., name, price, description, category, stock quantity).   
6. The administrator submits the changes.   
7. The system validates the input data (e.g., correct price format, positive stock quantity).   
8. The system updates the product record in the database.   
9. The system updates the inventory if the stock quantity has changed.   
10. The system displays a confirmation message that the product details were successfully updated.   
  
Alternative Flow:   
1. If the input data is invalid (e.g., negative stock quantity, incorrect price format), the system displays an error message and prompts the administrator to correct the input.   
2. If the product does not exist, the system displays an error message and suggests the administrator to add a new product instead.   
3. If the inventory update fails, the system displays an error and allows the administrator to retry or manually adjust inventory levels.   
4. If the administrator cancels the update, the system does not modify the product details and returns to the product management page.   
5. If the system encounters an error during the update process, it displays an error message and prompts the administrator to try again or contact support.  
  
Use Case Name: Categorize Products   
Use Case ID: UC-14   
Actors: Administrator, System, Product, Category   
Preconditions: The administrator is logged in. The system is accessible. The product and category databases are available.   
Postconditions: The product is assigned to the correct category. The system updates the product and category records. The system logs the administrator's action.   
  
Main Flow:   
1. The administrator navigates to the product categorization section of the system.   
2. The system displays a list of products that need to be categorized.   
3. The administrator selects a product to assign a category.   
4. The system displays a list of available categories.   
5. The administrator chooses the appropriate category for the selected product.   
6. The system validates the category selection and confirms the assignment.   
7. The system updates the product record with the new category information.   
8. The system logs the categorization action and displays a confirmation message.   
  
Alternative Flow:   
1. If the category database is unavailable, the system displays an error and prompts the administrator to try again later.   
2. If the administrator selects an invalid category, the system displays an error and allows them to choose a valid one.   
3. If the product already has a category assigned, the system displays the current category and allows the administrator to change it if needed.   
4. If the system fails to update the product record, it displays an error and allows the administrator to retry the operation.   
5. If the administrator cancels the categorization, the system does not make any changes and returns to the product list.  
  
Use Case Name: Create Category   
Use Case ID: UC-15   
Actors: Administrator, System, Category   
Preconditions: The administrator is logged in. The system is accessible. The category database is available.   
Postconditions: A new category is successfully created and stored in the system. The system logs the administrator's action.   
  
Main Flow:   
1. The administrator navigates to the category management section of the system.   
2. The system displays a form for creating a new category.   
3. The administrator enters the category name and description.   
4. The system validates the input data (e.g., non-empty name, valid description length).   
5. The system checks if a category with the same name already exists.   
6. If no duplicate is found, the system creates the new category and stores it in the database.   
7. The system logs the category creation and displays a confirmation message.   
  
Alternative Flow:   
1. If the category name is empty or invalid, the system displays an error message and prompts the administrator to provide a valid name.   
2. If the category name already exists, the system displays a message and prevents the creation of a duplicate category.   
3. If the category database is unavailable, the system displays an error and prompts the administrator to try again later.   
4. If the administrator cancels the operation, the system does not create the category and returns to the category management page.   
5. If the system encounters an error during category creation, it displays an error message and allows the administrator to retry or contact support.  
  
Use Case Name: Delete Category   
Use Case ID: UC-16   
Actors: Administrator, System, Category   
Preconditions: The administrator is logged in. The system is accessible. The category exists in the database. The category is not currently assigned to any active products.   
Postconditions: The selected category is removed from the system. The system logs the deletion.   
  
Main Flow:   
1. The administrator navigates to the category management section of the system.   
2. The system displays a list of existing categories.   
3. The administrator selects a category to delete.   
4. The system prompts the administrator to confirm the deletion.   
5. The administrator confirms the deletion.   
6. The system removes the category from the database.   
7. The system logs the deletion action and displays a confirmation message.   
  
Alternative Flow:   
1. If the category is assigned to active products, the system displays an error and prevents deletion.   
2. If the administrator cancels the deletion, the system retains the category and returns to the category management page.   
3. If the category database is unavailable, the system displays an error and prompts the administrator to try again later.   
4. If the system encounters an error during deletion, it displays an error message and allows the administrator to retry or contact support.  
  
Use Case Name: Manage Plugin   
Use Case ID: UC-17   
Actors: Administrator, System, Plugin   
Preconditions: The administrator is logged in. The system is accessible. The plugin database is available.   
Postconditions: The plugin is added, updated, or removed from the system. The system logs the administrator's action.   
  
Main Flow:   
1. The administrator navigates to the plugin management section of the system.   
2. The system displays a list of installed plugins with their status and details.   
3. The administrator selects an action (e.g., install new plugin, update plugin, deactivate or delete plugin).   
4. The system prompts the administrator to provide the necessary information for the selected action.   
5. The administrator enters the plugin name, version, and any configuration details.   
6. The system validates the input data and checks the compatibility of the plugin with the current system.   
7. The system performs the selected action (e.g., installs, updates, or deactivates the plugin).   
8. The system logs the action and displays a confirmation message.   
  
Alternative Flow:   
1. If the plugin name is empty or invalid, the system displays an error message and prompts the administrator to provide a valid name.   
2. If the plugin is already installed and the administrator tries to install it again, the system displays a message and allows the administrator to update the plugin instead.   
3. If the plugin database is unavailable, the system displays an error and prompts the administrator to try again later.   
4. If the system encounters an error during the plugin installation or update, it displays an error message and allows the administrator to retry or contact support.   
5. If the administrator cancels the action, the system does not make any changes and returns to the plugin management page.  
  
Use Case Name: View Payment Information   
Use Case ID: UC-18   
Actors: Customer, System, Payment   
Preconditions: The customer is logged in. The system is accessible. The customer has at least one payment method on file.   
Postconditions: The customer is shown their saved payment methods. The system logs the customer's access to the payment information.   
  
Main Flow:   
1. The customer navigates to the "Payment Information" section of their account.   
2. The system retrieves the customer's saved payment methods from the database.   
3. The system displays the payment information, including card type, last four digits, expiration date, and default status.   
4. The customer can select a payment method to edit or set as default.   
5. The system updates the display based on the customer's selection.   
  
Alternative Flow:   
1. If the customer has no saved payment methods, the system displays a message and provides an option to add a new payment method.   
2. If the database is unavailable, the system displays an error message and prompts the customer to try again later.   
3. If the customer selects an option to edit or remove a payment method, the system redirects to the appropriate use case (e.g., "Edit Payment Method" or "Remove Payment Method").   
4. If the system encounters an error retrieving payment information, it displays an error message and suggests the customer contact support.  
  
Use Case Name: Manage Customer   
Use Case ID: UC-19   
Actors: Administrator, System, Customer   
Preconditions: The administrator is logged in. The system is accessible. The customer database is available. The administrator has the necessary permissions to manage customer accounts.   
Postconditions: The customer account is updated, blocked, or removed based on the administrator's action. The system logs the administrator's action.   
  
Main Flow:   
1. The administrator navigates to the customer management section of the system.   
2. The system displays a list of registered customers, including their email, name, account status, and recent activity.   
3. The administrator selects a specific customer to manage.   
4. The system displays the customer's detailed profile and account information.   
5. The administrator chooses an action (e.g., edit customer details, block account, delete account).   
6. The system validates the selected action and prompts the administrator for confirmation if necessary.   
7. The system performs the selected action (e.g., updates the customer's information, blocks access, or removes the account).   
8. The system logs the action taken by the administrator.   
9. The system displays a confirmation message indicating the result of the action.   
  
Alternative Flow:   
1. If the customer database is unavailable, the system displays an error message and prompts the administrator to try again later.   
2. If the administrator selects an invalid or non-existent customer, the system displays an error and suggests selecting a valid account.   
3. If the system encounters an error during the management process (e.g., database error, permission denied), it displays an error message and allows the administrator to retry or contact support.   
4. If the administrator cancels the action, the system does not modify the customer account and returns to the customer list.   
5. If the administrator attempts to delete a customer with active orders, the system displays a warning and prompts for confirmation before proceeding.  
  
Use Case Name: Manage Product   
Use Case ID: UC-20   
Actors: Administrator, System, Product, Inventory   
Preconditions: The administrator is logged in. The system is accessible. The product database is available. The administrator has the necessary permissions to manage product information.   
Postconditions: The product is added, updated, or removed from the system. The inventory is adjusted accordingly. The system logs the administrator's action.   
  
Main Flow:   
1. The administrator navigates to the product management section of the system.   
2. The system displays a list of existing products with their details (e.g., name, price, description, stock quantity).   
3. The administrator selects an action (e.g., add new product, update product, delete product).   
4. The system opens the appropriate form or interface for the selected action.   
5. If adding a new product, the administrator enters product details (e.g., name, price, description, category, stock quantity).   
6. If updating an existing product, the administrator selects a product and modifies its information.   
7. If deleting a product, the administrator selects a product and confirms the deletion.   
8. The system validates the input data (e.g., correct price format, positive stock quantity, valid category).   
9. The system updates the product database and adjusts the inventory if necessary.   
10. The system logs the action taken by the administrator and displays a confirmation message.   
  
Alternative Flow:   
1. If the input data is invalid (e.g., negative stock quantity, incorrect price format), the system displays an error message and prompts the administrator to correct the input.   
2. If the administrator tries to add a product with the same name and category as an existing product, the system displays a message and allows the administrator to update the existing product instead.   
3. If the product database is unavailable, the system displays an error and prompts the administrator to try again later.   
4. If the administrator cancels the action, the system does not modify the product database and returns to the product management page.   
5. If the system encounters an error during the product management process, it displays an error message and allows the administrator to retry the operation or contact support.  
  
Use Case Name: Manage Order   
Use Case ID: UC-21   
Actors: Administrator, System, Order, Customer   
Preconditions: The administrator is logged in. The system is accessible. The order database is available. The administrator has the necessary permissions to manage orders.   
Postconditions: The order is updated (e.g., status changed, details modified) or deleted based on the administrator's action. The system logs the administrator's order management actions. The customer is notified of any changes if applicable.   
  
Main Flow:   
1. The administrator navigates to the order management section of the system.   
2. The system displays a list of existing orders, including order number, customer name, total amount, and current status.   
3. The administrator selects a specific order to manage.   
4. The system displays the detailed order information, including items, quantities, shipping address, and payment status.   
5. The administrator chooses an action (e.g., update order status, edit order details, delete order).   
6. The system validates the selected action and prompts the administrator for confirmation if necessary.   
7. If updating the order status, the administrator selects a new status (e.g., "Processing", "Shipped", "Delivered", "Cancelled").   
8. If editing order details, the administrator modifies relevant information (e.g., shipping address, item quantities).   
9. If deleting an order, the system checks if the order is eligible for deletion (e.g., no shipped items, no active payment).   
10. The system performs the selected action, updates the order database, and adjusts inventory if items are removed or modified.   
11. The system logs the action taken by the administrator.   
12. If the order status or details are changed, the system sends a notification to the customer (e.g., via email or in-app message).   
13. The system displays a confirmation message indicating the result of the action.   
  
Alternative Flow:   
1. If the order database is unavailable, the system displays an error message and prompts the administrator to try again later.   
2. If the administrator selects an invalid or non-existent order, the system displays an error and suggests selecting a valid order.   
3. If the system encounters an error during the order management process (e.g., database error, permission denied), it displays an error message and allows the administrator to retry the operation or contact support.   
4. If the administrator cancels the action, the system does not modify the order and returns to the order list.   
5. If the administrator tries to delete an order with shipped or delivered items, the system displays a warning and prevents the deletion.   
6. If the customer notification fails to send, the system displays an error and allows the administrator to manually send the notification.