项目文档

# Functional Requirement

# 1. Functional Requirements   
  
## 1.1 Customer Account Creation   
\*\*Function ID\*\*: FR-01   
\*\*Description\*\*: The system allows a customer to register an account by providing valid email, password, and other required information.   
\*\*Input\*\*: Customer's email, password, full name, and other optional registration fields.   
\*\*Output\*\*: A new customer account stored in the system with a confirmation email sent.   
  
## 1.2 Customer Login   
\*\*Function ID\*\*: FR-02   
\*\*Description\*\*: The system authenticates a customer’s login request using their email and password.   
\*\*Input\*\*: Email and password.   
\*\*Output\*\*: A successful login with a session established.   
  
## 1.3 Customer Logout   
\*\*Function ID\*\*: FR-03   
\*\*Description\*\*: The system allows a logged-in customer to end their session and log out securely.   
\*\*Input\*\*: Customer session ID or authentication token.   
\*\*Output\*\*: Session data cleared and customer redirected to login/homepage.   
  
## 1.4 Manage Customer Profile   
\*\*Function ID\*\*: FR-04   
\*\*Description\*\*: The system allows a customer to update or delete their profile information, such as email, password, and addresses.   
\*\*Input\*\*: Customer session, new profile data (e.g., updated email, password, address).   
\*\*Output\*\*: Updated or deleted customer profile in the system.   
  
## 1.5 View Purchase History   
\*\*Function ID\*\*: FR-05   
\*\*Description\*\*: The system allows a customer to view a list of their previous orders with detailed information.   
\*\*Input\*\*: Customer session and query parameters (e.g., date range, order status).   
\*\*Output\*\*: A list of order details (e.g., order number, date, total amount, items).   
  
## 1.6 Add Payment Information   
\*\*Function ID\*\*: FR-06   
\*\*Description\*\*: The system allows a customer to add a new payment method, such as credit card details and billing address.   
\*\*Input\*\*: Customer session, payment type, card number (last four digits), expiration date, CVV, billing address ID.   
\*\*Output\*\*: New payment information stored securely in the system.   
  
## 1.7 Update Payment Information   
\*\*Function ID\*\*: FR-07   
\*\*Description\*\*: The system allows a customer to modify an existing payment method.   
\*\*Input\*\*: Customer session, payment ID, updated payment details.   
\*\*Output\*\*: Updated payment information stored in the system.   
  
## 1.8 Delete Payment Information   
\*\*Function ID\*\*: FR-08   
\*\*Description\*\*: The system allows a customer to remove a payment method from their profile.   
\*\*Input\*\*: Customer session, payment ID, and identity verification (e.g., password).   
\*\*Output\*\*: Payment information removed from the system.   
  
## 1.9 Add Product to Shopping Cart   
\*\*Function ID\*\*: FR-09   
\*\*Description\*\*: The system allows a customer to add a product to their shopping cart with a specified quantity.   
\*\*Input\*\*: Customer session (or guest cart ID), product ID, and quantity.   
\*\*Output\*\*: Updated shopping cart with the new item and quantity.   
  
## 1.10 View Shopping Cart   
\*\*Function ID\*\*: FR-10   
\*\*Description\*\*: The system allows a customer to view the current contents of their shopping cart, including product details, quantity, and total cost.   
\*\*Input\*\*: Customer session (or guest cart ID).   
\*\*Output\*\*: Cart contents with total items and total price displayed.   
  
## 1.11 Modify Cart Item Quantity   
\*\*Function ID\*\*: FR-11   
\*\*Description\*\*: The system allows a customer to change the quantity of an item in the shopping cart.   
\*\*Input\*\*: Customer session (or guest cart ID), cart item ID, and new quantity.   
\*\*Output\*\*: Updated cart item with the new quantity and updated total price.   
  
## 1.12 Remove Cart Item   
\*\*Function ID\*\*: FR-12   
\*\*Description\*\*: The system allows a customer to remove a specific item from their shopping cart.   
\*\*Input\*\*: Customer session (or guest cart ID), cart item ID.   
\*\*Output\*\*: Cart item removed and updated cart summary displayed.   
  
## 1.13 Proceed to Checkout   
\*\*Function ID\*\*: FR-13   
\*\*Description\*\*: The system allows a customer to proceed from the shopping cart to the checkout process, where payment and shipping information is collected.   
\*\*Input\*\*: Customer session (or guest cart ID), selected payment method, shipping address.   
\*\*Output\*\*: A checkout page with all cart items, selected payment, and shipping address.   
  
## 1.14 Confirm Order Placement   
\*\*Function ID\*\*: FR-14   
\*\*Description\*\*: The system allows a customer to confirm and finalize their order after completing the checkout process.   
\*\*Input\*\*: Customer session, confirmed cart items, payment method, and shipping address.   
\*\*Output\*\*: A new order record created in the system with updated inventory and a confirmation email sent.   
  
## 1.15 Send Order Confirmation Email   
\*\*Function ID\*\*: FR-15   
\*\*Description\*\*: The system sends a confirmation email to the customer after an order is successfully placed.   
\*\*Input\*\*: Order ID, customer email, and order details.   
\*\*Output\*\*: Email notification sent to the customer with the order summary.   
  
## 1.16 Manage Inventory   
\*\*Function ID\*\*: FR-16   
\*\*Description\*\*: The system allows an administrator to add, update, or delete product inventory records.   
\*\*Input\*\*: Admin session, product ID, stock quantity, restock threshold, and restock date.   
\*\*Output\*\*: Updated inventory records in the system.   
  
## 1.17 Add New Product   
\*\*Function ID\*\*: FR-17   
\*\*Description\*\*: The system allows an administrator to add a new product, including name, price, description, category, and image.   
\*\*Input\*\*: Admin session, product name, price, description, category ID, stock quantity, and image file (optional).   
\*\*Output\*\*: A new product record stored in the database and associated with a category.   
  
## 1.18 Update Product Details   
\*\*Function ID\*\*: FR-18   
\*\*Description\*\*: The system allows an administrator to update an existing product's details, such as name, price, description, category, and image.   
\*\*Input\*\*: Admin session, product ID, updated product details.   
\*\*Output\*\*: Updated product record in the database.   
  
## 1.19 Delete Product   
\*\*Function ID\*\*: FR-19   
\*\*Description\*\*: The system allows an administrator to delete a product from the system, ensuring it is not part of any active orders.   
\*\*Input\*\*: Admin session, product ID.   
\*\*Output\*\*: Product record removed from the system and associated inventory updated.   
  
## 1.20 Assign Product to Category   
\*\*Function ID\*\*: FR-20   
\*\*Description\*\*: The system allows an administrator to assign a product to a specific category.   
\*\*Input\*\*: Admin session, product ID, category ID.   
\*\*Output\*\*: Updated product-category association stored in the database.   
  
## 1.21 Create Product Category   
\*\*Function ID\*\*: FR-21   
\*\*Description\*\*: The system allows an administrator to create a new product category with a name and description.   
\*\*Input\*\*: Admin session, category name, description, and parent category ID (optional).   
\*\*Output\*\*: A new category record stored in the database.   
  
## 1.22 Update Product Category   
\*\*Function ID\*\*: FR-22   
\*\*Description\*\*: The system allows an administrator to modify an existing product category’s name or description.   
\*\*Input\*\*: Admin session, category ID, updated category name and description.   
\*\*Output\*\*: Updated category record in the database.   
  
## 1.23 Delete Product Category   
\*\*Function ID\*\*: FR-23   
\*\*Description\*\*: The system allows an administrator to delete a product category after ensuring no products are associated with it.   
\*\*Input\*\*: Admin session, category ID.   
\*\*Output\*\*: Category record removed from the database.   
  
## 1.24 Process Payment   
\*\*Function ID\*\*: FR-24   
\*\*Description\*\*: The system processes the customer's selected payment method during checkout and confirms the transaction.   
\*\*Input\*\*: Order ID, payment method details (e.g., card number, expiration, billing address).   
\*\*Output\*\*: Payment processed, order status updated to "Paid", and transaction recorded.   
  
## 1.25 Manage Order Detail   
\*\*Function ID\*\*: FR-25   
\*\*Description\*\*: The system allows an administrator to modify order details such as product quantity or item status.   
\*\*Input\*\*: Admin session, order detail ID, updated product quantity or status.   
\*\*Output\*\*: Updated order detail stored in the database.   
  
## 1.26 View Order Details   
\*\*Function ID\*\*: FR-26   
\*\*Description\*\*: The system allows a customer or administrator to view the detailed information of a specific order.   
\*\*Input\*\*: Order ID and user session.   
\*\*Output\*\*: Display of order items, total amount, payment method, and status.   
  
## 1.27 Cancel Order   
\*\*Function ID\*\*: FR-28   
\*\*Description\*\*: The system allows a customer to cancel an order before it is shipped and updates inventory accordingly.   
\*\*Input\*\*: Customer session, order ID.   
\*\*Output\*\*: Order status updated to "Canceled", inventory updated, and cancellation email sent.   
  
## 1.28 Manage Address   
\*\*Function ID\*\*: FR-29   
\*\*Description\*\*: The system allows a customer to add, update, or delete shipping and billing addresses.   
\*\*Input\*\*: Customer session, address ID (for update/delete), address details (for add/update).   
\*\*Output\*\*: Updated or deleted address record stored in the system.   
  
## 1.29 Manage Session   
\*\*Function ID\*\*: FR-30   
\*\*Description\*\*: The system manages customer login sessions, including session creation, expiration, and invalidation.   
\*\*Input\*\*: Customer login credentials, session ID, or logout request.   
\*\*Output\*\*: Session created, updated, or destroyed.   
  
## 1.30 Manage Notification Preference   
\*\*Function ID\*\*: FR-31   
\*\*Description\*\*: The system allows a customer to set their notification preferences (e.g., enable/disable email, SMS, push notifications).   
\*\*Input\*\*: Customer session, notification preference ID, and updated preferences.   
\*\*Output\*\*: Updated notification preference stored in the system.   
  
## 1.31 Manage Plugin   
\*\*Function ID\*\*: FR-32   
\*\*Description\*\*: The system allows an administrator to install, uninstall, or configure plugins to extend system functionality.   
\*\*Input\*\*: Admin session, plugin ID, plugin file (for install), configuration parameters (for update).   
\*\*Output\*\*: Plugin installed, uninstalled, or configured, and system functionality updated accordingly.   
  
## 1.32 View Order History   
\*\*Function ID\*\*: FR-33   
\*\*Description\*\*: The system allows a customer to view a summary of all their previous orders.   
\*\*Input\*\*: Customer session.   
\*\*Output\*\*: A list of orders with order numbers, dates, total prices, and statuses.   
  
## 1.33 Send Email Notification   
\*\*Function ID\*\*: FR-34   
\*\*Description\*\*: The system sends email notifications to customers for various events, such as account confirmation, order updates, or reminders.   
\*\*Input\*\*: Notification type, recipient email, and content.   
\*\*Output\*\*: Email sent to the customer and notification status updated in the system.

# External Description

# Chapter 2: External Interfaces  
  
This chapter defines the external interfaces that the system interacts with, categorized into \*\*User Interface\*\*, \*\*Hardware Interface\*\*, \*\*Software Interface\*\*, and \*\*Communication Interface\*\*. These interfaces facilitate interaction between the system and its users, other systems, hardware devices, or external services.  
  
---  
  
## 2.1 User Interface  
  
### 2.1.1 Web-Based User Interface (UI)  
  
\*\*Description\*\*:   
The system provides a web-based user interface through which customers and administrators interact with the system to perform various functions such as account creation, product browsing, order placement, and inventory management.  
  
\*\*Functions Supported\*\*:  
- Customer Account Creation (FR-01)  
- Customer Login/Logout (FR-02, FR-03)  
- Profile Management (FR-04)  
- Purchase History & Order Details (FR-05, FR-26, FR-33)  
- Shopping Cart Management (FR-09–FR-13)  
- Checkout and Order Placement (FR-13, FR-14, FR-24)  
- Payment Information Management (FR-06–FR-08)  
- Address Management (FR-29)  
- Notification Preferences (FR-31)  
- Administrator Product and Category Management (FR-16–FR-23)  
- Order Management (FR-25, FR-28)  
- Plugin Management (FR-32)  
  
\*\*Interaction Method\*\*:  
- HTML/CSS/JavaScript-based frontend.  
- Responsive design for desktop and mobile devices.  
- Input via form fields, buttons, dropdowns, etc.  
- Output includes dynamic content rendering (e.g., cart summary, order history).  
  
---  
  
## 2.2 Hardware Interface  
  
There are no direct interactions with specific hardware devices in the current functional scope of the system. The system is designed to operate on standard computing hardware and does not require any specialized peripheral devices for normal operation.  
  
\*\*Note\*\*: In future extensions, hardware interfaces may be introduced (e.g., barcode scanners, payment terminals), but they are not part of the current requirements.  
  
---  
  
## 2.3 Software Interface  
  
### 2.3.1 Database System  
  
\*\*Description\*\*:   
The system interacts with a relational database system to store and retrieve persistent data such as customer accounts, product information, orders, inventory, payment details, and notification preferences.  
  
\*\*Data Entities\*\*:  
- Customers (FR-01–FR-04)  
- Products (FR-17–FR-23)  
- Inventory (FR-16)  
- Orders (FR-14, FR-25–FR-28)  
- Payments (FR-06–FR-08, FR-24)  
- Addresses (FR-29)  
- Categories (FR-20–FR-23)  
- Notifications (FR-31, FR-34)  
- Plugins (FR-32)  
  
\*\*Database Interaction Method\*\*:  
- CRUD operations via SQL queries or ORM framework.  
- Secure access using authentication credentials.  
- Transactions used where applicable (e.g., during order placement and inventory update).  
- Encrypted storage of sensitive data (e.g., passwords, payment details).  
  
### 2.3.2 Third-Party Payment Gateway API  
  
\*\*Description\*\*:   
The system integrates with a third-party payment gateway to securely process payments during checkout.  
  
\*\*Supported Functions\*\*:  
- Process Payment (FR-24)  
  
\*\*Interaction Method\*\*:  
- RESTful API calls to the payment service provider.  
- Input: Order ID, payment method details (card number, expiration date, billing address).  
- Output: Transaction status (success/failure), transaction ID.  
- Security: TLS encryption, tokenization of card data, PCI-DSS compliance.  
  
### 2.3.3 Email Service API  
  
\*\*Description\*\*:   
The system uses an email service API to send automated emails to customers for notifications such as account confirmation, order confirmation, cancellations, and reminders.  
  
\*\*Supported Functions\*\*:  
- Send Order Confirmation Email (FR-15)  
- Send Email Notification (FR-34)  
  
\*\*Interaction Method\*\*:  
- Integration with SMTP server or cloud-based email service (e.g., SendGrid, Amazon SES).  
- Input: Recipient email, subject, message body, attachments (if any).  
- Output: Delivery status (sent/delivered/failed) logged in the system.  
  
---  
  
## 2.4 Communication Interface  
  
### 2.4.1 HTTP/HTTPS Protocol  
  
\*\*Description\*\*:   
The system communicates over the internet using HTTP/HTTPS protocols for all client-server interactions, ensuring secure and encrypted communication.  
  
\*\*Supported Functions\*\*:  
- All user-facing operations (e.g., login, registration, shopping cart, checkout).  
- Administrator functions (product and order management).  
  
\*\*Interaction Method\*\*:  
- RESTful API endpoints for backend-to-frontend communication.  
- JSON format for request/response payloads.  
- HTTPS for encryption in transit.  
- Session tokens or cookies used for state management.  
  
### 2.4.2 Email Communication  
  
\*\*Description\*\*:   
Email is used as a communication channel to notify users about important events related to their account, orders, and system updates.  
  
\*\*Supported Functions\*\*:  
- Send Order Confirmation Email (FR-15)  
- Send Email Notification (FR-34)  
  
\*\*Interaction Method\*\*:  
- Asynchronous communication triggered by system events.  
- Emails sent via integrated email service.  
- Content dynamically generated based on event type and user context.  
  
---  
  
## Summary Table of External Interfaces  
  
| Interface Type | Interface Name | Associated Functional Requirements |  
|----------------------|------------------------------|--------------------------------------------------------|  
| User Interface | Web-Based UI | FR-01–FR-04, FR-05–FR-14, FR-16–FR-34 |  
| Software Interface | Database System | FR-01–FR-08, FR-14–FR-29, FR-31–FR-34 |  
| Software Interface | Payment Gateway API | FR-24 |  
| Software Interface | Email Service API | FR-15, FR-34 |  
| Communication Interface | HTTP/HTTPS Protocol | All |  
| Communication Interface | Email Communication | FR-15, FR-34 |  
  
All external interfaces have been identified and mapped directly from the functional requirements to ensure traceability and completeness.

# Use Case

Use Case Name: Register Customer Account   
Use Case ID: UC-01   
Actors: Customer, Administrator   
Preconditions:   
1. The system is operational and accessible to the customer.   
2. The customer has not yet registered an account.   
3. The administrator has configured the account registration functionality.   
  
Postconditions:   
1. A new customer account is successfully created in the system.   
2. The customer receives a confirmation email.   
3. The customer is logged in and redirected to the dashboard or homepage.   
  
Main Flow:   
1. The customer navigates to the registration page.   
2. The customer fills in the required information (e.g., full name, email address, password, and other details).   
3. The customer submits the registration form.   
4. The system validates the input data (e.g., email format, password strength).   
5. The system checks if the email is already registered.   
6. The system creates the customer account and stores the information in the database.   
7. The system sends a confirmation email to the customer.   
8. The customer clicks the confirmation link in the email.   
9. The system verifies the email and activates the account.   
10. The customer is redirected to the homepage or dashboard and is now logged in.   
  
Alternative Flow:   
1. If the email is already registered (Step 5), the system displays an error message and prompts the customer to use a different email.   
2. If the input data is invalid (Step 4), the system displays an error message and highlights the fields that need correction.   
3. If the customer does not click the confirmation link within a specified time (e.g., 24 hours), the system may send a reminder email or deactivate the unconfirmed account.  
  
Use Case Name: Manage Customer Profile   
Use Case ID: UC-02   
Actors: Customer, Administrator   
Preconditions:   
1. The customer has a registered and active account.   
2. The system is accessible to the customer.   
3. The administrator has enabled the profile management feature.   
  
Postconditions:   
1. The customer profile is updated or deleted as requested.   
2. The system logs the changes made to the customer profile.   
3. The customer receives a confirmation email if the profile is updated or deleted.   
  
Main Flow:   
1. The customer logs into the system.   
2. The customer navigates to the profile management section.   
3. The customer selects an option to either update or delete their profile.   
4. If updating, the customer modifies the necessary information (e.g., address, phone number, email).   
5. The customer submits the changes.   
6. The system validates the input data (e.g., new email format, password if required).   
7. The system updates the customer's information in the database.   
8. The system sends a confirmation email to the customer.   
9. The system displays a success message to the customer.   
  
If deleting, the customer confirms the deletion request.   
The system verifies the customer's identity (e.g., via password).   
The system deletes the customer's profile and associated data (e.g., orders, cart items).   
The system sends a confirmation email to the customer.   
The system logs out the customer and redirects them to the homepage.   
  
Alternative Flow:   
1. If the customer enters invalid data during an update (Step 6), the system displays an error message and highlights the fields that need correction.   
2. If the customer attempts to delete their profile and has active orders (Step 10), the system displays a warning and prevents deletion until all orders are resolved.   
3. If the customer fails to confirm their identity before deletion (Step 11), the system denies the deletion request and prompts for re-verification.   
4. If the system fails to send the confirmation email (Step 8 or 13), the customer is notified, and the administrator is alerted to investigate.  
  
Use Case Name: View Purchase History   
Use Case ID: UC-03   
Actors: Customer, Administrator   
Preconditions:   
1. The customer has a registered and active account.   
2. The system is accessible and operational.   
3. The administrator has enabled the purchase history feature.   
  
Postconditions:   
1. The customer's purchase history is displayed on the screen.   
2. The system logs the access to the purchase history.   
3. If there is no purchase history, the system informs the customer.   
  
Main Flow:   
1. The customer logs into the system.   
2. The customer navigates to the "Purchase History" section.   
3. The system retrieves the customer's order records from the database.   
4. The system displays the list of previous orders, including order details (e.g., product, quantity, date, total price).   
5. The customer can select an individual order to view more detailed information.   
6. The system presents the selected order’s information, including product names, prices, and order status.   
  
Alternative Flow:   
1. If the customer has no previous orders (Step 3), the system displays a message indicating that there is no purchase history available.   
2. If the system fails to retrieve the order records (Step 3), it displays an error message and logs the issue for the administrator to review.   
3. If the customer's session expires during the process, the system prompts them to log in again and resumes from the previous page if possible.  
  
Use Case Name: Add Payment Information   
Use Case ID: UC-04   
Actors: Customer, Administrator   
Preconditions:   
1. The customer has a registered and active account.   
2. The system is accessible and operational.   
3. The administrator has enabled the payment information management feature.   
  
Postconditions:   
1. The customer's payment information is successfully added and stored in the system.   
2. The system logs the addition of the payment information.   
3. The customer receives a confirmation email.   
  
Main Flow:   
1. The customer logs into the system.   
2. The customer navigates to the "Payment Information" section.   
3. The customer selects the option to add a new payment method.   
4. The customer enters the necessary payment details (e.g., card number, expiration date, CVV, billing address).   
5. The customer submits the payment information.   
6. The system validates the input data (e.g., card format, expiration date).   
7. The system securely stores the payment information in the database.   
8. The system sends a confirmation email to the customer.   
9. The system displays a success message to the customer.   
  
Alternative Flow:   
1. If the input data is invalid (Step 6), the system displays an error message and highlights the fields that need correction.   
2. If the customer does not provide all required fields, the system prompts them to complete the missing information.   
3. If the system fails to send the confirmation email (Step 8), the customer is notified, and the administrator is alerted to investigate.   
4. If the customer cancels the operation before submission, the system discards the entered data without saving.  
  
Use Case Name: Update Payment Information   
Use Case ID: UC-05   
Actors: Customer, Administrator   
Preconditions:   
1. The customer has a registered and active account.   
2. The system is accessible and operational.   
3. The customer has existing payment information in the system.   
4. The administrator has enabled the payment information management feature.   
  
Postconditions:   
1. The customer's payment information is successfully updated in the system.   
2. The system logs the update action.   
3. The customer receives a confirmation email.   
  
Main Flow:   
1. The customer logs into the system.   
2. The customer navigates to the "Payment Information" section.   
3. The customer selects an existing payment method to update.   
4. The customer modifies the necessary payment details (e.g., card number, expiration date, billing address).   
5. The customer submits the updated payment information.   
6. The system validates the input data (e.g., card format, expiration date).   
7. The system securely updates the payment information in the database.   
8. The system sends a confirmation email to the customer.   
9. The system displays a success message to the customer.   
  
Alternative Flow:   
1. If the input data is invalid (Step 6), the system displays an error message and highlights the fields that need correction.   
2. If the customer cancels the operation before submission, the system reverts to the original payment information without saving changes.   
3. If the system fails to send the confirmation email (Step 8), the customer is notified, and the administrator is alerted to investigate.   
4. If the customer attempts to update a payment method and the system does not support the selected card type, the system displays an error message and prompts the customer to choose a supported method.  
  
Use Case Name: Delete Payment Information   
Use Case ID: UC-06   
Actors: Customer, Administrator   
Preconditions:   
1. The customer has a registered and active account.   
2. The system is accessible and operational.   
3. The customer has existing payment information in the system.   
4. The administrator has enabled the payment information management feature.   
  
Postconditions:   
1. The selected payment information is successfully deleted from the system.   
2. The system logs the deletion action.   
3. The customer receives a confirmation email.   
  
Main Flow:   
1. The customer logs into the system.   
2. The customer navigates to the "Payment Information" section.   
3. The customer selects an existing payment method to delete.   
4. The customer confirms the deletion request.   
5. The system verifies the customer's identity (e.g., via password).   
6. The system deletes the selected payment information from the database.   
7. The system sends a confirmation email to the customer.   
8. The system displays a success message to the customer.   
  
Alternative Flow:   
1. If the input data is invalid (Step 4), the system displays an error message and highlights the fields that need correction.   
2. If the customer cancels the operation before deletion, the system retains the original payment information without making any changes.   
3. If the system fails to send the confirmation email (Step 7), the customer is notified, and the administrator is alerted to investigate.   
4. If the customer fails to confirm their identity before deletion (Step 5), the system denies the deletion request and prompts for re-verification.  
  
Use Case Name: Login to Account   
Use Case ID: UC-07   
Actors: Customer, Administrator   
Preconditions:   
1. The customer has a registered and active account.   
2. The system is accessible and operational.   
3. The administrator has enabled the login functionality.   
  
Postconditions:   
1. The customer is successfully logged into their account.   
2. The system logs the login event.   
3. The customer is redirected to the dashboard or homepage.   
  
Main Flow:   
1. The customer navigates to the login page.   
2. The customer enters their registered email and password.   
3. The customer clicks the "Login" button.   
4. The system validates the email and password.   
5. The system verifies if the account is active.   
6. If valid and active, the system logs the customer in and redirects them to the homepage or dashboard.   
  
Alternative Flow:   
1. If the email or password is incorrect (Step 4), the system displays an error message and prompts the customer to re-enter the correct information.   
2. If the account is not active (e.g., not confirmed via email) (Step 5), the system displays a message indicating the account is inactive and provides a link to check the confirmation email.   
3. If the customer exceeds the maximum number of login attempts, the system locks the account temporarily and notifies the administrator.   
4. If the system fails to log the login event, the administrator is alerted for review.  
  
Use Case Name: Logout from Account   
Use Case ID: UC-08   
Actors: Customer, Administrator   
Preconditions:   
1. The customer has a registered and active account.   
2. The customer is currently logged into the system.   
3. The system is accessible and operational.   
4. The administrator has enabled the logout functionality.   
  
Postconditions:   
1. The customer is successfully logged out of their account.   
2. The system logs the logout event.   
3. The customer is redirected to the homepage or login page.   
  
Main Flow:   
1. The customer navigates to the account settings or profile section.   
2. The customer selects the "Logout" option.   
3. The system verifies the customer's session and initiates the logout process.   
4. The system clears the customer's session data and cookies.   
5. The system logs the logout event.   
6. The customer is redirected to the homepage or login page.   
  
Alternative Flow:   
1. If the system fails to clear the session data (Step 4), the system displays an error message and prompts the customer to try again or contact support.   
2. If the customer's session has already expired, the system automatically redirects them to the login page without requiring a logout.   
3. If the system fails to log the logout event, the administrator is alerted for review.  
  
Use Case Name: Manage Inventory   
Use Case ID: UC-09   
Actors: Administrator, Product, Category, Order   
Preconditions:   
1. The administrator has a registered and active account.   
2. The system is accessible and operational.   
3. The administrator has the necessary permissions to manage inventory.   
4. The system contains existing product and category data.   
  
Postconditions:   
1. The inventory is updated with added, modified, or removed products.   
2. The system logs the inventory management actions.   
3. The changes in inventory are reflected in the product listings and order processing.   
  
Main Flow:   
1. The administrator logs into the system.   
2. The administrator navigates to the "Inventory Management" section.   
3. The administrator selects an option to add, update, or delete a product.   
4. If adding a product, the administrator enters product details (e.g., name, price, description, category, stock quantity).   
5. The system validates the product information (e.g., price format, stock quantity).   
6. The system adds the new product to the database and updates the inventory.   
7. The system displays a success message to the administrator.   
  
If updating a product, the administrator selects an existing product and modifies the necessary details.   
The system validates the updated information.   
The system updates the product in the database.   
The system displays a success message to the administrator.   
  
If deleting a product, the administrator selects an existing product and confirms the deletion.   
The system verifies if the product is associated with active orders.   
The system deletes the product from the database if no active orders are linked.   
The system displays a success message to the administrator.   
  
Alternative Flow:   
1. If the input data is invalid (Step 5 or 9), the system displays an error message and highlights the fields that need correction.   
2. If the administrator attempts to delete a product that is part of an active order (Step 12), the system displays a warning and prevents deletion until the order is resolved.   
3. If the administrator cancels the operation before submission, the system reverts to the previous state without saving changes.   
4. If the system fails to log the inventory management action, the administrator is alerted for review.  
  
Use Case Name: Add New Product   
Use Case ID: UC-10   
Actors: Administrator, Product, Category   
Preconditions:   
1. The administrator has a registered and active account.   
2. The system is accessible and operational.   
3. The administrator has the necessary permissions to add new products.   
4. At least one category exists in the system.   
  
Postconditions:   
1. A new product is successfully added to the system and associated with a category.   
2. The product details are stored in the database.   
3. The system logs the addition of the new product.   
  
Main Flow:   
1. The administrator logs into the system.   
2. The administrator navigates to the "Product Management" section.   
3. The administrator selects the "Add New Product" option.   
4. The administrator enters the product details (e.g., name, price, description, category, stock quantity, and image if applicable).   
5. The system validates the product information (e.g., price format, stock quantity, and category selection).   
6. The system adds the new product to the database and associates it with the selected category.   
7. The system displays a success message to the administrator.   
  
Alternative Flow:   
1. If the input data is invalid (Step 5), the system displays an error message and highlights the fields that need correction.   
2. If no category is selected (Step 4), the system prompts the administrator to select a valid category before proceeding.   
3. If the system fails to store the product information (Step 6), it displays an error message and logs the issue for the administrator to review.   
4. If the administrator cancels the operation before submission, the system discards the entered data without saving.  
  
Use Case Name: Update Product Details   
Use Case ID: UC-11   
Actors: Administrator, Product, Category   
Preconditions:   
1. The administrator has a registered and active account.   
2. The system is accessible and operational.   
3. The administrator has the necessary permissions to update product details.   
4. At least one product exists in the system.   
5. The relevant category for the product is already defined.   
  
Postconditions:   
1. The selected product's details are successfully updated in the system.   
2. The product is associated with the updated category (if applicable).   
3. The system logs the update action.   
4. The changes are reflected in product listings and related functionalities (e.g., shopping cart, order processing).   
  
Main Flow:   
1. The administrator logs into the system.   
2. The administrator navigates to the "Product Management" section.   
3. The administrator selects an existing product to update.   
4. The administrator modifies the product details (e.g., name, price, description, category, stock quantity, or image).   
5. The system validates the updated product information (e.g., price format, stock quantity, category selection).   
6. The system updates the product details in the database.   
7. The system displays a success message to the administrator.   
  
Alternative Flow:   
1. If the input data is invalid (Step 5), the system displays an error message and highlights the fields that need correction.   
2. If the administrator attempts to assign the product to a non-existent category (Step 4), the system prompts for a valid category selection.   
3. If the system fails to update the product information (Step 6), it displays an error message and logs the issue for the administrator to review.   
4. If the administrator cancels the operation before submission, the system reverts to the original product details without saving changes.  
  
Use Case Name: Delete Product   
Use Case ID: UC-12   
Actors: Administrator, Product, Category, Order   
Preconditions:   
1. The administrator has a registered and active account.   
2. The system is accessible and operational.   
3. The administrator has the necessary permissions to delete products.   
4. The product to be deleted exists in the system.   
5. The system contains existing order data.   
  
Postconditions:   
1. The selected product is successfully deleted from the system.   
2. The system logs the deletion action.   
3. The product is removed from the associated category.   
4. The product is no longer available for purchase in the shopping cart or on the homepage.   
  
Main Flow:   
1. The administrator logs into the system.   
2. The administrator navigates to the "Product Management" section.   
3. The administrator selects an existing product to delete.   
4. The administrator confirms the deletion request.   
5. The system verifies if the product is associated with any active orders.   
6. If no active orders are linked, the system proceeds to delete the product from the database.   
7. The system removes the product from its associated category.   
8. The system logs the deletion action.   
9. The system displays a success message to the administrator.   
  
Alternative Flow:   
1. If the product is part of an active order (Step 5), the system displays a warning and prevents deletion until the order is resolved.   
2. If the administrator cancels the deletion request (Step 4), the system retains the product and displays a cancellation message.   
3. If the system fails to delete the product from the database (Step 6), it displays an error message and logs the issue for the administrator to review.   
4. If the system fails to log the deletion action, the administrator is alerted for review.  
  
Use Case Name: Assign Product to Category   
Use Case ID: UC-13   
Actors: Administrator, Product, Category   
Preconditions:   
1. The administrator has a registered and active account.   
2. The system is accessible and operational.   
3. The administrator has the necessary permissions to assign products to categories.   
4. The product and category to be assigned exist in the system.   
  
Postconditions:   
1. The product is successfully assigned to the selected category.   
2. The system logs the assignment action.   
3. The product is now listed under the assigned category in product listings.   
  
Main Flow:   
1. The administrator logs into the system.   
2. The administrator navigates to the "Product Management" section.   
3. The administrator selects an existing product to assign to a category.   
4. The administrator chooses a category from the list of available categories.   
5. The system validates the category selection.   
6. The system updates the product's category association in the database.   
7. The system displays a success message to the administrator.   
  
Alternative Flow:   
1. If the selected category does not exist (Step 4), the system displays an error message and prompts the administrator to select a valid category.   
2. If the system fails to update the category association (Step 6), it displays an error message and logs the issue for the administrator to review.   
3. If the administrator cancels the operation before submission, the system retains the product's original category without making changes.   
4. If the system fails to log the assignment action, the administrator is alerted for review.  
  
Use Case Name: Create Product Category   
Use Case ID: UC-14   
Actors: Administrator, Category   
Preconditions:   
1. The administrator has a registered and active account.   
2. The system is accessible and operational.   
3. The administrator has the necessary permissions to create product categories.   
4. The system allows the creation of new categories.   
  
Postconditions:   
1. A new product category is successfully created and stored in the database.   
2. The system logs the creation of the new category.   
3. The new category is available for use in product management.   
  
Main Flow:   
1. The administrator logs into the system.   
2. The administrator navigates to the "Category Management" section.   
3. The administrator selects the option to create a new category.   
4. The administrator enters the category details (e.g., category name, description).   
5. The system validates the input data (e.g., name uniqueness, description length).   
6. The system creates the new category and stores it in the database.   
7. The system displays a success message to the administrator.   
  
Alternative Flow:   
1. If the input data is invalid (Step 5), the system displays an error message and highlights the fields that need correction.   
2. If the category name is already taken (Step 5), the system displays a message and prompts the administrator to choose a different name.   
3. If the system fails to store the category in the database (Step 6), it displays an error message and logs the issue for the administrator to review.   
4. If the administrator cancels the operation before submission, the system discards the entered data without saving.  
  
Use Case Name: Update Product Category   
Use Case ID: UC-15   
Actors: Administrator, Product, Category   
Preconditions:   
1. The administrator has a registered and active account.   
2. The system is accessible and operational.   
3. The administrator has the necessary permissions to update product categories.   
4. The product to be updated exists in the system.   
5. The new category to be assigned exists in the system.   
  
Postconditions:   
1. The product is successfully updated to the new category.   
2. The system logs the category update action.   
3. The product is now listed under the updated category in product listings.   
  
Main Flow:   
1. The administrator logs into the system.   
2. The administrator navigates to the "Product Management" section.   
3. The administrator selects an existing product whose category needs to be updated.   
4. The administrator chooses a new category from the list of available categories.   
5. The system validates the category selection.   
6. The system updates the product's category association in the database.   
7. The system displays a success message to the administrator.   
  
Alternative Flow:   
1. If the selected category does not exist (Step 4), the system displays an error message and prompts the administrator to select a valid category.   
2. If the system fails to update the category association (Step 6), it displays an error message and logs the issue for the administrator to review.   
3. If the administrator cancels the operation before submission, the system retains the product's original category without making changes.   
4. If the system fails to log the update action, the administrator is alerted for review.  
  
Use Case Name: Delete Product Category   
Use Case ID: UC-16   
Actors: Administrator, Product, Category, Order   
Preconditions:   
1. The administrator has a registered and active account.   
2. The system is accessible and operational.   
3. The administrator has the necessary permissions to delete product categories.   
4. The category to be deleted exists in the system.   
5. No products are currently assigned to the category, or all associated products have been reassigned or deleted.   
  
Postconditions:   
1. The selected product category is successfully deleted from the system.   
2. The system logs the deletion action.   
3. The category is removed from the product listings and category dropdowns.   
4. Any products previously associated with the category are updated accordingly (e.g., moved to a default or other category).   
  
Main Flow:   
1. The administrator logs into the system.   
2. The administrator navigates to the "Category Management" section.   
3. The administrator selects an existing category to delete.   
4. The administrator confirms the deletion request.   
5. The system verifies if any products are still associated with the category.   
6. If no products are linked, the system proceeds to delete the category from the database.   
7. The system updates any related references (e.g., product listings, dropdowns).   
8. The system logs the deletion action.   
9. The system displays a success message to the administrator.   
  
Alternative Flow:   
1. If products are still associated with the category (Step 5), the system displays a warning and prevents deletion until all products are reassigned or deleted.   
2. If the administrator cancels the deletion request (Step 4), the system retains the category and displays a cancellation message.   
3. If the system fails to delete the category from the database (Step 6), it displays an error message and logs the issue for the administrator to review.   
4. If the system fails to log the deletion action, the administrator is alerted for review.  
  
Use Case Name: Add Product to Shopping Cart   
Use Case ID: UC-17   
Actors: Customer, Product, Shopping Cart   
Preconditions:   
1. The customer has a registered and active account or is browsing as a guest.   
2. The system is accessible and operational.   
3. The product to be added exists in the system.   
4. The shopping cart functionality is enabled.   
  
Postconditions:   
1. The selected product is added to the customer's shopping cart.   
2. The shopping cart is updated with the new product and quantity.   
3. The system logs the addition of the product to the cart.   
4. The customer is informed of the successful addition (e.g., via a message or cart summary).   
  
Main Flow:   
1. The customer browses the product catalog or searches for a product.   
2. The customer selects a product to view its details.   
3. The customer clicks the "Add to Cart" button.   
4. The system validates the product availability and stock quantity.   
5. The system adds the product to the shopping cart with the default quantity (usually 1).   
6. The system updates the cart summary (e.g., total items, total price).   
7. The system displays a confirmation message to the customer (e.g., "Product added to cart").   
  
Alternative Flow:   
1. If the product is out of stock (Step 4), the system displays an error message and suggests alternatives or a restock date.   
2. If the customer specifies a quantity exceeding available stock (Step 4), the system adjusts the quantity to the maximum available and notifies the customer.   
3. If the customer is logged in, the cart is associated with their account; if not, it is stored as a guest cart until checkout.   
4. If the system fails to update the cart summary (Step 6), it displays an error message and logs the issue for the administrator to review.   
5. If the system fails to log the cart update, the administrator is alerted for review.  
  
Use Case Name: View Shopping Cart   
Use Case ID: UC-18   
Actors: Customer, Shopping Cart, Product   
Preconditions:   
1. The customer has a registered and active account or is browsing as a guest.   
2. The system is accessible and operational.   
3. The shopping cart functionality is enabled.   
4. At least one product has been added to the shopping cart.   
  
Postconditions:   
1. The customer can view the current contents of their shopping cart.   
2. The system displays the product details, quantities, and total price.   
3. The system logs the view action.   
4. The customer can proceed with checkout or modify the cart contents.   
  
Main Flow:   
1. The customer navigates to the shopping cart section from the homepage, product page, or account dashboard.   
2. The system retrieves the cart items associated with the customer (logged in or guest).   
3. The system displays the list of products in the cart, including product name, price, quantity, and subtotal.   
4. The system shows the total cart price and any applicable discounts.   
5. The customer can choose to proceed to checkout or continue shopping.   
  
Alternative Flow:   
1. If no products are in the cart (Step 2), the system displays a message indicating that the cart is empty.   
2. If the system fails to retrieve the cart items (Step 2), it displays an error message and logs the issue for the administrator to review.   
3. If the customer is logged in and views a guest cart, the system merges the guest cart with the customer's account cart.   
4. If the system fails to log the view action, the administrator is alerted for review.  
  
Use Case Name: Modify Cart Contents   
Use Case ID: UC-19   
Actors: Customer, Shopping Cart, Product   
  
Preconditions:   
1. The customer has a registered and active account or is browsing as a guest.   
2. The system is accessible and operational.   
3. The shopping cart functionality is enabled.   
4. At least one product is already in the shopping cart.   
  
Postconditions:   
1. The cart contents are modified (e.g., quantity updated, product removed, or replaced).   
2. The system logs the modification action.   
3. The cart summary (e.g., total price, total items) is updated accordingly.   
4. The customer is informed of the successful modification.   
  
Main Flow:   
1. The customer navigates to the shopping cart section.   
2. The system displays the current cart contents.   
3. The customer selects an option to modify the cart (e.g., change quantity, remove product).   
4. If changing quantity, the customer enters the new quantity for a specific product.   
5. The system validates the new quantity against the product’s stock availability.   
6. The system updates the cart with the new quantity or removes the product as requested.   
7. The system recalculates the total price and updates the cart summary.   
8. The system displays a confirmation message to the customer.   
  
Alternative Flow:   
1. If the new quantity exceeds available stock (Step 5), the system displays an error message and adjusts the quantity to the maximum available.   
2. If the system fails to update the cart (Step 6), it displays an error message and logs the issue for the administrator to review.   
3. If the system fails to recalculate the cart summary (Step 7), it displays an error message and logs the issue for the administrator to review.   
4. If the customer cancels the modification, the system retains the original cart contents.   
5. If the system fails to log the modification action, the administrator is alerted for review.  
  
Use Case Name: Remove Product from Cart   
Use Case ID: UC-20   
Actors: Customer, Shopping Cart, Product   
Preconditions:   
1. The customer has a registered and active account or is browsing as a guest.   
2. The system is accessible and operational.   
3. The shopping cart functionality is enabled.   
4. At least one product is already in the shopping cart.   
  
Postconditions:   
1. The selected product is successfully removed from the shopping cart.   
2. The cart summary (e.g., total price, total items) is updated accordingly.   
3. The system logs the removal action.   
4. The customer is informed of the successful removal.   
  
Main Flow:   
1. The customer navigates to the shopping cart section.   
2. The system displays the current cart contents, including product details.   
3. The customer selects a product to remove from the cart.   
4. The customer confirms the removal request.   
5. The system removes the selected product from the cart.   
6. The system recalculates the total price and updates the cart summary.   
7. The system displays a confirmation message to the customer (e.g., "Product removed from cart").   
  
Alternative Flow:   
1. If the customer cancels the removal request (Step 4), the system retains the product in the cart and displays a cancellation message.   
2. If the system fails to remove the product from the cart (Step 5), it displays an error message and logs the issue for the administrator to review.   
3. If the system fails to recalculate the cart summary (Step 6), it displays an error message and logs the issue for the administrator to review.   
4. If the system fails to log the removal action, the administrator is alerted for review.  
  
Use Case Name: Proceed to Checkout   
Use Case ID: UC-21   
Actors: Customer, Shopping Cart, Product, Order, Payment Information, Email Notification   
  
Preconditions:   
1. The customer has a registered and active account or is browsing as a guest.   
2. The system is accessible and operational.   
3. At least one product is in the customer's shopping cart.   
4. The customer has provided or selected valid payment information.   
5. The shopping cart and checkout functionalities are enabled.   
  
Postconditions:   
1. The customer is directed to the checkout page with all selected cart items.   
2. The system generates a new order with the selected items and payment method.   
3. The system logs the checkout process and order creation.   
4. The customer receives an order confirmation email.   
5. The shopping cart is cleared or updated accordingly.   
  
Main Flow:   
1. The customer views the shopping cart and selects "Proceed to Checkout."   
2. The system validates the cart contents (e.g., product availability, stock quantity).   
3. The system displays the checkout page, including customer details, shipping address, and selected payment method.   
4. The customer reviews and confirms the order details (e.g., items, price, shipping information).   
5. The system processes the payment using the selected payment method.   
6. The system confirms the successful payment and creates a new order record in the database.   
7. The system sends an order confirmation email to the customer.   
8. The system clears the shopping cart or updates it for future use.   
9. The system displays a success message to the customer.   
  
Alternative Flow:   
1. If a product in the cart is out of stock (Step 2), the system displays a warning and removes the product from the cart.   
2. If the customer’s payment method is invalid or not supported (Step 5), the system displays an error message and prompts the customer to select a valid payment option.   
3. If the system fails to process the payment (Step 5), it displays an error message and logs the issue for the administrator to review.   
4. If the system fails to send the confirmation email (Step 7), the customer is notified, and the administrator is alerted for review.   
5. If the customer cancels the checkout process (Step 4), the system returns to the cart view without making any changes to the order or payment.  
  
Use Case Name: Confirm Order Placement   
Use Case ID: UC-22   
Actors: Customer, Shopping Cart, Product, Order, Payment Information, Email Notification   
  
Preconditions:   
1. The customer has a registered and active account or is browsing as a guest.   
2. The system is accessible and operational.   
3. At least one product is in the customer’s shopping cart.   
4. The customer has selected a valid payment method.   
5. The system has completed the checkout process and generated a draft order.   
6. The order placement functionality is enabled.   
  
Postconditions:   
1. The order is confirmed and finalized in the system.   
2. The system logs the order confirmation.   
3. The customer receives an email notification confirming the order.   
4. The shopping cart is cleared or updated as appropriate.   
5. Inventory quantities are updated based on the order.   
  
Main Flow:   
1. The customer proceeds to checkout and confirms the order details.   
2. The system verifies the selected payment method and cart contents.   
3. The customer clicks the "Confirm Order" button.   
4. The system processes the payment transaction.   
5. The system confirms the payment and creates the order record in the database.   
6. The system updates the product inventory by reducing stock quantities.   
7. The system sends a confirmation email to the customer.   
8. The system clears the shopping cart.   
9. The system displays a confirmation message to the customer (e.g., "Your order has been placed").   
  
Alternative Flow:   
1. If the payment fails (Step 4), the system displays an error message and prompts the customer to try a different payment method.   
2. If the product inventory is insufficient after confirmation (Step 6), the system displays a warning and adjusts the order or cancels it.   
3. If the system fails to send the confirmation email (Step 7), the customer is notified, and the administrator is alerted for review.   
4. If the customer cancels the order confirmation (Step 3), the system returns to the checkout page without finalizing the order.  
  
Use Case Name: Receive Order Confirmation Email   
Use Case ID: UC-23   
Actors: Customer, Email Notification, Order, Payment Information   
Preconditions:   
1. The customer has placed an order and completed the checkout process.   
2. The system has successfully processed the payment and created the order record.   
3. The email notification functionality is enabled.   
4. The customer has provided a valid email address during registration or checkout.   
  
Postconditions:   
1. The customer receives a confirmation email with the order details.   
2. The system logs the email notification event.   
3. The customer is informed that the order has been successfully placed.   
  
Main Flow:   
1. The system completes the order confirmation process.   
2. The system prepares an email notification containing the order summary (e.g., order number, items, total amount, and estimated delivery time).   
3. The system sends the email to the customer’s registered email address.   
4. The system logs the email notification as successfully sent.   
5. The customer receives the confirmation email and reviews the order details.   
  
Alternative Flow:   
1. If the customer's email address is invalid or not properly formatted, the system logs the error and displays a message prompting the customer to update their email information.   
2. If the system fails to send the email (e.g., due to server issues), it logs the error and notifies the administrator for resolution.   
3. If the customer does not receive the email, the system provides an option for the customer to request a resend of the confirmation email.  
  
Use Case Name: Process Payment   
Use Case ID: UC-24   
Actors: Customer, Payment Information, Order, Email Notification   
Preconditions:   
1. The customer has a registered and active account or is browsing as a guest.   
2. The system is accessible and operational.   
3. The customer has at least one product in the shopping cart and has selected a valid payment method.   
4. The checkout process has been completed, and the order is ready for payment processing.   
5. The payment processing functionality is enabled in the system.   
  
Postconditions:   
1. The payment is successfully processed and recorded in the system.   
2. The order status is updated to reflect the completed payment.   
3. The system logs the payment process.   
4. The customer receives an email notification confirming the payment and order details.   
5. Inventory quantities are updated based on the order items.   
  
Main Flow:   
1. The customer completes the checkout process and selects a valid payment method.   
2. The customer clicks the "Process Payment" button.   
3. The system validates the payment information (e.g., card number, expiration date, CVV).   
4. The system initiates the payment transaction with the selected payment gateway.   
5. The payment gateway processes the transaction and returns a confirmation.   
6. The system updates the order status to "Paid" and records the transaction details.   
7. The system reduces the stock quantity for each product in the order.   
8. The system sends a payment confirmation email to the customer.   
9. The system displays a success message to the customer (e.g., "Payment processed successfully").   
  
Alternative Flow:   
1. If the payment information is invalid (Step 3), the system displays an error message and prompts the customer to correct the data.   
2. If the payment gateway declines the transaction (Step 5), the system displays an error message and suggests trying a different payment method.   
3. If the system fails to update the order status (Step 6), it logs the error and displays a message to the customer, prompting them to contact support or retry.   
4. If the system fails to send the confirmation email (Step 8), the customer is notified, and the administrator is alerted for review.   
5. If the customer cancels the payment process before completion, the system returns to the checkout page without finalizing the transaction.  
  
Use Case Name: Manage Plugins   
Use Case ID: UC-25   
Actors: Administrator, Plugin   
Preconditions:   
1. The administrator has a registered and active account.   
2. The system is accessible and operational.   
3. The administrator has the necessary permissions to manage plugins.   
4. The plugin management feature is enabled in the system.   
5. The system contains existing plugins or allows the installation of new plugins.   
  
Postconditions:   
1. The selected plugin is either added, updated, or deleted from the system.   
2. The system logs the plugin management actions.   
3. The changes to plugins are reflected in the system's functionality.   
4. The administrator is informed of the success or failure of the action.   
  
Main Flow:   
1. The administrator logs into the system.   
2. The administrator navigates to the "Plugin Management" section.   
3. The administrator selects an option to add, update, or delete a plugin.   
4. If adding, the administrator uploads or selects a new plugin and confirms the installation.   
5. The system validates the plugin and installs it if valid.   
6. If updating, the administrator selects an existing plugin and uploads a new version.   
7. The system validates the update and applies it to the plugin.   
8. If deleting, the administrator selects an existing plugin and confirms the deletion.   
9. The system verifies if the plugin is in use or has dependencies.   
10. The system removes the plugin and logs the action.   
11. The system displays a success message to the administrator.   
  
Alternative Flow:   
1. If the plugin file is invalid or corrupted (Step 4 or 6), the system displays an error message and prompts the administrator to upload a valid file.   
2. If the plugin is in use or has dependencies (Step 9), the system displays a warning and prevents deletion until the dependencies are resolved.   
3. If the system fails to validate or install the plugin (Step 5), it displays an error message and logs the issue for the administrator to review.   
4. If the administrator cancels the operation at any step, the system reverts to the previous state without making any changes.   
5. If the system fails to log the plugin management action, the administrator is alerted for review.  
  
Use Case Name: Install Plugin   
Use Case ID: UC-26   
Actors: Administrator, Plugin   
Preconditions:   
1. The administrator has a registered and active account.   
2. The system is accessible and operational.   
3. The administrator has the necessary permissions to install plugins.   
4. The plugin management feature is enabled in the system.   
5. The system allows the installation of new plugins.   
  
Postconditions:   
1. The selected plugin is successfully installed and activated in the system.   
2. The system logs the installation event.   
3. The plugin's functionality is integrated and available for use.   
4. The administrator is informed of the successful installation.   
  
Main Flow:   
1. The administrator logs into the system.   
2. The administrator navigates to the "Plugin Management" section.   
3. The administrator selects the "Install Plugin" option.   
4. The administrator uploads or selects a plugin file.   
5. The system validates the plugin file (e.g., format, compatibility).   
6. The system installs the plugin and activates its functionality.   
7. The system logs the installation action.   
8. The system displays a success message to the administrator (e.g., "Plugin installed successfully").   
  
Alternative Flow:   
1. If the plugin file is invalid or incompatible (Step 5), the system displays an error message and prompts the administrator to upload a valid plugin.   
2. If the system fails to install the plugin (Step 6), it displays an error message and logs the issue for the administrator to review.   
3. If the administrator cancels the installation before completion, the system discards the plugin file and reverts to the previous state.   
4. If the system fails to log the installation event, the administrator is alerted for review.  
  
Use Case Name: Uninstall Plugin   
Use Case ID: UC-27   
Actors: Administrator, Plugin   
Preconditions:   
1. The administrator has a registered and active account.   
2. The system is accessible and operational.   
3. The plugin management feature is enabled.   
4. The plugin to be uninstalled is currently installed and active in the system.   
  
Postconditions:   
1. The selected plugin is successfully uninstalled and deactivated.   
2. The system logs the uninstallation event.   
3. The plugin’s functionality is removed from the system.   
4. The administrator is informed of the successful uninstallation.   
  
Main Flow:   
1. The administrator logs into the system.   
2. The administrator navigates to the "Plugin Management" section.   
3. The administrator selects an installed plugin from the list.   
4. The administrator clicks the "Uninstall" option.   
5. The system checks if the plugin is in use or has dependencies.   
6. The administrator confirms the uninstallation request.   
7. The system uninstalls and deactivates the plugin.   
8. The system logs the uninstallation action.   
9. The system displays a success message to the administrator.   
  
Alternative Flow:   
1. If the plugin is in use or has dependencies (Step 5), the system displays a warning and prevents uninstallation until the dependencies are resolved.   
2. If the administrator cancels the uninstallation request (Step 6), the system retains the plugin and displays a cancellation message.   
3. If the system fails to uninstall the plugin (Step 7), it displays an error message and logs the issue for the administrator to review.   
4. If the system fails to log the uninstallation event, the administrator is alerted for review.  
  
Use Case Name: Configure Plugin Settings   
Use Case ID: UC-28   
Actors: Administrator, Plugin   
Preconditions:   
1. The administrator has a registered and active account.   
2. The system is accessible and operational.   
3. The plugin management feature is enabled.   
4. At least one plugin is installed and available for configuration.   
  
Postconditions:   
1. The plugin settings are successfully updated or configured.   
2. The system logs the configuration action.   
3. The plugin’s behavior is adjusted according to the new settings.   
4. The administrator is informed of the success or failure of the configuration.   
  
Main Flow:   
1. The administrator logs into the system.   
2. The administrator navigates to the "Plugin Management" section.   
3. The administrator selects an installed plugin for configuration.   
4. The system displays the plugin's current settings and configuration options.   
5. The administrator modifies the settings as needed (e.g., enable/disable features, set thresholds, or update API keys).   
6. The administrator submits the configuration changes.   
7. The system validates the configuration input (e.g., format, value ranges).   
8. The system updates the plugin's configuration in the database.   
9. The system logs the configuration update.   
10. The system displays a success message to the administrator (e.g., "Plugin settings updated successfully").   
  
Alternative Flow:   
1. If the configuration input is invalid (Step 7), the system displays an error message and highlights the fields that need correction.   
2. If the administrator cancels the configuration before submission, the system reverts to the original settings.   
3. If the system fails to update the plugin configuration (Step 8), it displays an error message and logs the issue for the administrator to review.   
4. If the system fails to log the configuration action, the administrator is alerted for review.  
  
Use Case Name: Send Email Notification   
Use Case ID: UC-29   
Actors: Administrator, Email Notification, Customer   
Preconditions:   
1. The administrator has a registered and active account.   
2. The system is accessible and operational.   
3. The email notification feature is enabled.   
4. A valid email address for the customer is available in the system.   
5. The notification content or reason is specified (e.g., order confirmation, account activation).   
  
Postconditions:   
1. The email notification is successfully sent to the customer.   
2. The system logs the email notification event.   
3. The customer is informed of the specified event or message.   
4. If the email is not delivered, the system logs the failure and alerts the administrator.   
  
Main Flow:   
1. The administrator selects the option to send an email notification.   
2. The administrator specifies the recipient (customer) and the type of notification.   
3. The system prepares the email content based on the notification type and customer information.   
4. The system sends the email to the customer’s registered email address.   
5. The system logs the email notification as successfully sent.   
6. The administrator receives a confirmation that the email has been sent.   
  
Alternative Flow:   
1. If the customer's email address is invalid or not properly formatted (Step 2), the system displays an error message and prompts the administrator to update the email information.   
2. If the system fails to send the email (e.g., due to server issues) (Step 4), it logs the error and notifies the administrator for resolution.   
3. If the customer does not receive the email, the system provides an option for the administrator to request a resend of the notification.   
4. If the administrator cancels the email notification request before submission, the system does not send the email and displays a cancellation message.  
  
Use Case Name: View Order Details   
Use Case ID: UC-30   
Actors: Customer, Administrator, Order, Product, Payment Information   
Preconditions:   
1. The customer has a registered and active account or is viewing as an administrator.   
2. The system is accessible and operational.   
3. At least one order exists in the system for the customer.   
4. The order details feature is enabled.   
  
Postconditions:   
1. The order details are displayed to the customer or administrator.   
2. The system logs the access to the order details.   
3. The customer or administrator can view the status, items, payment method, and other relevant information.   
  
Main Flow:   
1. The customer logs into the system.   
2. The customer navigates to the "Order History" or "Orders" section.   
3. The customer selects a specific order to view its details.   
4. The system retrieves the order information from the database, including product details, quantity, total price, and payment status.   
5. The system displays the order details on the screen, including order number, date, status, and items purchased.   
6. The customer or administrator can review the information as needed.   
  
Alternative Flow:   
1. If the selected order does not exist (Step 3), the system displays an error message and prompts the customer to select a valid order.   
2. If the system fails to retrieve the order information (Step 4), it displays an error message and logs the issue for the administrator to review.   
3. If the customer’s session expires during the process, the system prompts them to log in again and resumes from the previous page if possible.  
  
Use Case Name: Cancel Order   
Use Case ID: UC-31   
Actors: Customer, Administrator, Order, Payment Information, Email Notification   
Preconditions:   
1. The customer has a registered and active account.   
2. The system is accessible and operational.   
3. The customer has at least one active order in the system.   
4. The cancel order functionality is enabled.   
5. The customer has selected the order to be canceled.   
  
Postconditions:   
1. The selected order is successfully canceled.   
2. The system logs the cancellation action.   
3. The customer receives a confirmation email about the canceled order.   
4. The payment information associated with the order is updated (e.g., refunded if applicable).   
5. Inventory quantities are adjusted based on the cancellation.   
  
Main Flow:   
1. The customer logs into the system.   
2. The customer navigates to the "Order History" or "Orders" section.   
3. The customer selects an active order to cancel.   
4. The customer confirms the cancellation request.   
5. The system verifies the order status and eligibility for cancellation (e.g., not already shipped).   
6. The system cancels the order and updates the status in the database.   
7. The system processes a refund if applicable and updates the payment information.   
8. The system adjusts the product inventory by increasing the stock quantity.   
9. The system sends a cancellation confirmation email to the customer.   
10. The system displays a success message to the customer (e.g., "Order has been canceled successfully").   
  
Alternative Flow:   
1. If the order is not eligible for cancellation (e.g., already shipped) (Step 5), the system displays an error message and prevents the cancellation.   
2. If the customer cancels the operation before submission (Step 4), the system retains the original order status without changes.   
3. If the system fails to update the order status (Step 6), it displays an error message and logs the issue for the administrator to review.   
4. If the system fails to process the refund (Step 7), it displays an error message and logs the issue for the administrator to review.   
5. If the system fails to send the confirmation email (Step 9), the customer is notified, and the administrator is alerted for review.  
  
Use Case Name: Manage Cart Item   
Use Case ID: UC-32   
Actors: Customer, Shopping Cart, Product   
  
Preconditions:   
1. The customer has a registered and active account or is browsing as a guest.   
2. The system is accessible and operational.   
3. The shopping cart functionality is enabled.   
4. At least one cart item exists in the customer's cart.   
  
Postconditions:   
1. The cart item is successfully modified or removed.   
2. The system logs the cart item management action.   
3. The cart summary (e.g., total items, total price) is updated accordingly.   
4. The customer is informed of the successful modification or removal.   
  
Main Flow:   
1. The customer logs into the system or browses as a guest.   
2. The customer navigates to the shopping cart section.   
3. The system displays the list of cart items, including product name, price, quantity, and subtotal.   
4. The customer selects an option to modify or remove a specific cart item.   
5. If modifying, the customer updates the quantity for the selected item.   
6. The system validates the new quantity against product stock availability.   
7. The system updates the cart item in the database with the modified quantity or removes it.   
8. The system recalculates the total price and updates the cart summary.   
9. The system displays a success message to the customer (e.g., "Cart item updated" or "Cart item removed").   
  
Alternative Flow:   
1. If the new quantity exceeds available stock (Step 6), the system displays an error message and adjusts the quantity to the maximum available.   
2. If the system fails to update or remove the cart item (Step 7), it displays an error message and logs the issue for the administrator to review.   
3. If the system fails to recalculate the cart summary (Step 8), it displays an error message and logs the issue for the administrator to review.   
4. If the customer cancels the modification or removal request, the system retains the original cart item details.   
5. If the system fails to log the cart item management action, the administrator is alerted for review.  
  
Use Case Name: Manage Order Detail   
Use Case ID: UC-33   
Actors: Administrator, Customer, Order, Product, OrderDetail   
  
Preconditions:   
1. The customer has a registered and active account.   
2. The system is accessible and operational.   
3. The customer has at least one order in the system.   
4. The order detail management feature is enabled.   
5. The administrator has the necessary permissions to manage order details (if applicable).   
  
Postconditions:   
1. The order detail is successfully modified or updated in the system.   
2. The system logs the action taken on the order detail.   
3. The updated order detail is reflected in the order summary and related reports.   
4. The customer or administrator is informed of the successful update or modification.   
  
Main Flow:   
1. The customer or administrator navigates to the "Order History" or "Orders" section.   
2. The customer selects a specific order to view or edit its details.   
3. The system retrieves the order details, including associated products, quantities, and prices.   
4. The customer or administrator selects a specific item (OrderDetail) to modify (e.g., change quantity, update product details).   
5. The system validates the requested changes (e.g., new quantity, product status).   
6. The system updates the OrderDetail in the database.   
7. The system recalculates the order total and updates the order summary.   
8. The system logs the modification of the order detail.   
9. The system displays a success message (e.g., "Order detail updated successfully").   
  
Alternative Flow:   
1. If the requested modification is not allowed due to order status (e.g., order is already shipped) (Step 4), the system displays an error message and prevents the change.   
2. If the system fails to validate the requested changes (Step 5), it displays an error message and highlights the invalid fields.   
3. If the system fails to update the OrderDetail (Step 6), it displays an error message and logs the issue for the administrator to review.   
4. If the system fails to recalculate the order total (Step 7), it displays an error message and logs the issue for the administrator to review.   
5. If the customer cancels the modification request, the system retains the original order detail without making any changes.   
6. If the system fails to log the order detail modification, the administrator is alerted for review.