项目文档

# Functional Requirement

1.1 Customer Registration Function   
 Function ID: FR-01   
 Description: Allows a new customer to register an account by providing valid personal information. The system validates the email format and password complexity, and sends a confirmation email.   
 Input: Name, Email, Password   
 Output: Active customer account with a confirmation email sent to the provided email address.  
  
1.2 Customer Login Function   
 Function ID: FR-02   
 Description: Authenticates a customer using their email and password. A successful login creates a session and redirects the customer to their account dashboard.   
 Input: Email, Password   
 Output: Active session for the customer, redirected to the account dashboard.  
  
1.3 Customer Logout Function   
 Function ID: FR-03   
 Description: Terminates an active customer session and clears session-related data.   
 Input: Active customer session   
 Output: Session terminated, customer redirected to login or home page.  
  
1.4 Product Information Viewing Function   
 Function ID: FR-04   
 Description: Allows a customer to view detailed product information, including name, description, price, and stock status.   
 Input: ProductID or search query   
 Output: Displayed product details including name, description, price, and availability.  
  
1.5 Add Product to Cart Function   
 Function ID: FR-05   
 Description: Enables a customer to add a selected product to their shopping cart, provided the product is in stock.   
 Input: ProductID, CustomerID (optional), Quantity   
 Output: Updated shopping cart with the added product and confirmation message.  
  
1.6 Create Order Function   
 Function ID: FR-06   
 Description: Allows a customer to proceed with checkout and create a new order after confirming their shipping details and selecting a payment method.   
 Input: CustomerID, ProductID list, Quantity list, Shipping Address, Payment Method   
 Output: Created order record, updated product inventory, and initiation of payment processing.  
  
1.7 View Order Details Function   
 Function ID: FR-07   
 Description: Enables a customer to view the details of a specific order, including product list, order status, and total amount.   
 Input: OrderID, CustomerID   
 Output: Displayed order details including product list, quantities, prices, date, and status.  
  
1.8 Update Order Status Function   
 Function ID: FR-08   
 Description: Allows an administrator to update the status of an order, such as from "Processing" to "Shipped" or "Delivered," and notify the customer via email.   
 Input: OrderID, New Status   
 Output: Updated order status and a notification email sent to the customer.  
  
1.9 Delete Order Function   
 Function ID: FR-09   
 Description: Enables an administrator to delete an order from the system if it is in a cancellable or deletable status and updates the inventory accordingly.   
 Input: OrderID, CustomerID, Current Order Status   
 Output: Deleted order record, updated product inventory, and a notification email sent to the customer.  
  
1.10 Create Payment Function   
 Function ID: FR-10   
 Description: Facilitates the creation of a payment record for a specific order after the customer confirms the payment method and provides the necessary payment details.   
 Input: OrderID, Payment Method, Payment Details (e.g., card information)   
 Output: Created payment record, updated order status to "Paid," and a confirmation email sent to the customer.  
  
1.11 Process Payment Function   
 Function ID: FR-11   
 Description: Processes the payment using the configured payment gateway API and updates the payment and order status accordingly.   
 Input: PaymentID, Transaction Details   
 Output: Processed payment record, updated order status, and logs for payment processing.  
  
1.12 View Payment History Function   
 Function ID: FR-12   
 Description: Allows a customer to view their historical payment records, including payment dates, amounts, and associated order information.   
 Input: CustomerID   
 Output: Displayed list of payment records with details such as date, amount, method, and order ID.  
  
1.13 Administrator Registration Function   
 Function ID: FR-13   
 Description: Enables a new administrator to register an account by providing valid personal information. The system validates the email format and password complexity and sends a confirmation email.   
 Input: Name, Email, Password   
 Output: Active administrator account with a confirmation email sent to the provided email address.  
  
1.14 Administrator Login Function   
 Function ID: FR-14   
 Description: Authenticates an administrator using their email and password. A successful login creates a session and redirects the administrator to the admin dashboard.   
 Input: Email, Password   
 Output: Active session for the administrator, redirected to the admin dashboard.  
  
1.15 Administrator Logout Function   
 Function ID: FR-15   
 Description: Terminates an active administrator session and clears session-related data.   
 Input: Active administrator session   
 Output: Session terminated, administrator redirected to login or home page.  
  
1.16 Add Product Function   
 Function ID: FR-16   
 Description: Enables an administrator to add a new product to the system database by providing product details and uploading an image.   
 Input: Product name, description, price, category, image   
 Output: New product record added to the database and a confirmation message displayed to the administrator.  
  
1.17 Update Product Information Function   
 Function ID: FR-17   
 Description: Allows an administrator to modify an existing product's details, such as name, description, price, or image.   
 Input: ProductID, updated product details, new image (optional)   
 Output: Updated product record in the database and a confirmation message displayed to the administrator.  
  
1.18 Delete Product Function   
 Function ID: FR-18   
 Description: Enables an administrator to delete a product from the system if it is not associated with any active orders.   
 Input: ProductID, current product status, list of associated orders   
 Output: Deleted product record from the database and a confirmation message displayed to the administrator.  
  
1.19 View Email Template Function   
 Function ID: FR-19   
 Description: Allows an administrator to view the content and structure of an email template used for notifications such as order confirmation or registration.   
 Input: TemplateID or template name   
 Output: Displayed email template content including subject, body, and placeholders for dynamic data.  
  
1.20 Send Order Confirmation Email Function   
 Function ID: FR-20   
 Description: Triggers the system to send a confirmation email to a customer after their order has been successfully processed.   
 Input: OrderID, CustomerID, Email Template   
 Output: Sent order confirmation email and a log entry for the email delivery status.  
  
1.21 Manage Plugin Function   
 Function ID: FR-21   
 Description: Enables an administrator to manage plugins by enabling, disabling, or configuring them based on system needs.   
 Input: PluginID, configuration parameters   
 Output: Updated plugin status or configuration in the database and a confirmation message to the administrator.  
  
1.22 View API Documentation Function   
 Function ID: FR-22   
 Description: Allows an administrator to view API documentation for the plugins integrated into the system.   
 Input: APID or API name   
 Output: Displayed API documentation in a structured format, including endpoints and usage examples.

# External Description

\*\*Chapter 2: External Interfaces\*\*  
  
---  
  
### 2.1 User Interface  
  
The system interacts with both \*\*customers\*\* and \*\*administrators\*\* through a series of user interfaces designed to support account management, product browsing, order creation, and administrative tasks.  
  
- \*\*Customer Registration Interface (FR-01)\*\*   
 - \*\*Description:\*\* A web-based form where new customers can input their \*\*Name\*\*, \*\*Email\*\*, and \*\*Password\*\*.   
 - \*\*Interaction Method:\*\* Form submission with real-time validation for email format and password complexity.   
 - \*\*Output:\*\* A confirmation message and a \*\*confirmation email\*\* sent to the customer’s email address.  
  
- \*\*Customer Login Interface (FR-02)\*\*   
 - \*\*Description:\*\* A login form where customers enter their \*\*Email\*\* and \*\*Password\*\* to authenticate.   
 - \*\*Interaction Method:\*\* Form submission with server-side authentication.   
 - \*\*Output:\*\* A successful login redirects the customer to the \*\*account dashboard\*\*; an error message is displayed for invalid credentials.  
  
- \*\*Customer Logout Interface (FR-03)\*\*   
 - \*\*Description:\*\* A logout button or menu option accessible after a customer has logged in.   
 - \*\*Interaction Method:\*\* Clicking or selecting the logout option terminates the session.   
 - \*\*Output:\*\* The customer is redirected to the \*\*login page\*\* or \*\*home page\*\*.  
  
- \*\*Product Information Interface (FR-04)\*\*   
 - \*\*Description:\*\* A product detail page or search result interface where customers can view information about a product.   
 - \*\*Interaction Method:\*\* Product details are displayed in a structured format after entering a \*\*ProductID\*\* or a \*\*search query\*\*.   
 - \*\*Output:\*\* Displays \*\*product name\*\*, \*\*description\*\*, \*\*price\*\*, \*\*stock status\*\*, and \*\*availability\*\*.  
  
- \*\*Add Product to Cart Interface (FR-05)\*\*   
 - \*\*Description:\*\* A button or interface element (e.g., “Add to Cart”) on the product detail page.   
 - \*\*Interaction Method:\*\* Customers click the button and optionally specify \*\*Quantity\*\*.   
 - \*\*Output:\*\* The shopping cart is updated with the product, and a \*\*confirmation message\*\* is displayed.  
  
- \*\*Create Order Interface (FR-06)\*\*   
 - \*\*Description:\*\* A checkout interface where customers confirm their \*\*shipping address\*\* and \*\*payment method\*\* before submitting the order.   
 - \*\*Interaction Method:\*\* Form submission with validation for shipping and payment details.   
 - \*\*Output:\*\* Displays a \*\*confirmation screen\*\* with order details and initiates \*\*payment processing\*\*.  
  
- \*\*View Order Details Interface (FR-07)\*\*   
 - \*\*Description:\*\* A page or modal where customers can view the details of a specific \*\*OrderID\*\*.   
 - \*\*Interaction Method:\*\* Customers select an order from a list or enter an \*\*OrderID\*\* directly.   
 - \*\*Output:\*\* Displays \*\*product list\*\*, \*\*quantities\*\*, \*\*prices\*\*, \*\*order date\*\*, and \*\*status\*\*.  
  
- \*\*Administrator Registration Interface (FR-13)\*\*   
 - \*\*Description:\*\* A form for new administrators to input their \*\*Name\*\*, \*\*Email\*\*, and \*\*Password\*\*.   
 - \*\*Interaction Method:\*\* Form submission with validation for email format and password complexity.   
 - \*\*Output:\*\* A confirmation message and a \*\*confirmation email\*\* sent to the administrator’s email.  
  
- \*\*Administrator Login Interface (FR-14)\*\*   
 - \*\*Description:\*\* A login form where administrators enter their \*\*Email\*\* and \*\*Password\*\* to access the admin dashboard.   
 - \*\*Interaction Method:\*\* Form submission with server-side authentication.   
 - \*\*Output:\*\* A successful login redirects the administrator to the \*\*admin dashboard\*\*; an error message is displayed for invalid credentials.  
  
- \*\*Administrator Logout Interface (FR-015)\*\*   
 - \*\*Description:\*\* A logout button or menu option accessible after an administrator has logged in.   
 - \*\*Interaction Method:\*\* Clicking or selecting the logout option terminates the session.   
 - \*\*Output:\*\* The administrator is redirected to the \*\*login page\*\* or \*\*home page\*\*.  
  
- \*\*Add Product Interface (FR-16)\*\*   
 - \*\*Description:\*\* A product creation form in the admin dashboard where administrators input \*\*product name\*\*, \*\*description\*\*, \*\*price\*\*, \*\*category\*\*, and upload a \*\*product image\*\*.   
 - \*\*Interaction Method:\*\* Form submission with server-side validation.   
 - \*\*Output:\*\* Displays a \*\*confirmation message\*\* upon successful product creation.  
  
- \*\*Update Product Information Interface (FR-17)\*\*   
 - \*\*Description:\*\* A form or modal where administrators can modify existing product details, including \*\*name\*\*, \*\*description\*\*, \*\*price\*\*, \*\*category\*\*, and optionally \*\*product image\*\*.   
 - \*\*Interaction Method:\*\* Select a \*\*ProductID\*\*, modify the fields, and submit the form.   
 - \*\*Output:\*\* Displays a \*\*confirmation message\*\* upon successful update.  
  
- \*\*Delete Product Interface (FR-18)\*\*   
 - \*\*Description:\*\* A confirmation dialog or button in the admin dashboard for deleting a product.   
 - \*\*Interaction Method:\*\* Select a \*\*ProductID\*\*, confirm the deletion based on product status and associated orders.   
 - \*\*Output:\*\* Displays a \*\*confirmation message\*\* upon successful deletion.  
  
- \*\*View Email Template Interface (FR-19)\*\*   
 - \*\*Description:\*\* A section in the admin dashboard where administrators can view and edit \*\*email templates\*\* used for notifications.   
 - \*\*Interaction Method:\*\* Search or select a \*\*TemplateID\*\* or \*\*template name\*\* to view the template content.   
 - \*\*Output:\*\* Displays the \*\*subject\*\*, \*\*body\*\*, and \*\*placeholders\*\* for dynamic data (e.g., customer name, order ID).  
  
- \*\*Manage Plugin Interface (FR-21)\*\*   
 - \*\*Description:\*\* A plugin management panel in the admin dashboard where administrators can \*\*enable\*\*, \*\*disable\*\*, or \*\*configure plugins\*\*.   
 - \*\*Interaction Method:\*\* Select a \*\*PluginID\*\* and modify its configuration parameters.   
 - \*\*Output:\*\* Displays a \*\*confirmation message\*\* upon successful plugin status or configuration update.  
  
- \*\*View API Documentation Interface (FR-22)\*\*   
 - \*\*Description:\*\* A documentation viewer in the admin dashboard where administrators can access \*\*API documentation\*\* for integrated plugins.   
 - \*\*Interaction Method:\*\* Select an \*\*APID\*\* or \*\*API name\*\* to view the structured documentation.   
 - \*\*Output:\*\* Displays \*\*endpoints\*\*, \*\*usage examples\*\*, \*\*request/response formats\*\*, and \*\*authentication methods\*\*.  
  
---  
  
### 2.2 Hardware Interface  
  
No direct interaction with hardware devices is required for the current system. The system is designed to operate in a \*\*software-only\*\* environment, with no dependencies on physical hardware such as printers, scanners, or IoT devices.  
  
- \*\*No hardware interfaces defined.\*\*  
  
---  
  
### 2.3 Software Interface  
  
The system interacts with various \*\*software components\*\*, including \*\*databases\*\*, \*\*third-party APIs\*\*, and \*\*external tools\*\*, to support its functionalities.  
  
- \*\*Database (System Database)\*\*   
 - \*\*Description:\*\* The system uses a relational database to store \*\*customer accounts\*\*, \*\*product records\*\*, \*\*order data\*\*, \*\*payment records\*\*, \*\*email templates\*\*, and \*\*plugin configurations\*\*.   
 - \*\*Interaction Method:\*\* CRUD (Create, Read, Update, Delete) operations via SQL or ORM (Object-Relational Mapping) queries.   
 - \*\*Inputs/Outputs:\*\*   
 - Input: Data from user interfaces or API calls.   
 - Output: Updated or retrieved records for display or processing.  
  
- \*\*Payment Gateway API (External Third-Party Service)\*\*   
 - \*\*Description:\*\* The system integrates with a \*\*payment gateway API\*\* (e.g., Stripe, PayPal) to process customer payments.   
 - \*\*Interaction Method:\*\* RESTful API calls to initiate and confirm payments.   
 - \*\*Inputs/Outputs:\*\*   
 - Input: \*\*Payment Method\*\*, \*\*Payment Details\*\*, and \*\*OrderID\*\*.   
 - Output: \*\*Processed Payment Record\*\*, \*\*updated Order Status\*\*, and \*\*logs\*\* for payment transactions.  
  
- \*\*Email Service API (External Third-Party Service)\*\*   
 - \*\*Description:\*\* The system uses an \*\*email service API\*\* (e.g., SendGrid, Amazon SES) to send \*\*confirmation emails\*\*, \*\*order status updates\*\*, and \*\*notifications\*\* to customers.   
 - \*\*Interaction Method:\*\* RESTful API calls with email templates and dynamic data.   
 - \*\*Inputs/Outputs:\*\*   
 - Input: \*\*Email Template\*\*, \*\*CustomerID\*\*, \*\*OrderID\*\*, and dynamic content (e.g., order details, customer name).   
 - Output: \*\*Email sent\*\* to the customer, and a \*\*log entry\*\* for delivery status.  
  
- \*\*Plugin APIs (Internal or External Plugins)\*\*   
 - \*\*Description:\*\* The system supports \*\*plugin-based extensions\*\* that can be managed and configured by administrators. Each plugin may expose its own API for integration.   
 - \*\*Interaction Method:\*\* RESTful or GraphQL API calls based on the plugin’s specifications.   
 - \*\*Inputs/Outputs:\*\*   
 - Input: \*\*PluginID\*\*, \*\*configuration parameters\*\*, and \*\*API requests\*\*.   
 - Output: \*\*Plugin status updates\*\*, \*\*configuration changes\*\*, and \*\*API responses\*\*.  
  
---  
  
### 2.4 Communication Interface  
  
The system communicates with \*\*external services\*\* and \*\*users\*\* through \*\*network-based protocols\*\* such as \*\*HTTP/HTTPS\*\*, \*\*email\*\*, and \*\*message queues\*\* (if applicable).  
  
- \*\*Email Communication (FR-01, FR-03, FR-08, FR-10, FR-20)\*\*   
 - \*\*Description:\*\* The system sends \*\*emails\*\* to customers for \*\*registration confirmation\*\*, \*\*logout notifications\*\*, \*\*order status updates\*\*, and \*\*payment confirmations\*\*.   
 - \*\*Interaction Method:\*\* Email is sent via a \*\*third-party email service API\*\* using a \*\*structured template\*\* with dynamic data insertion.   
 - \*\*Inputs/Outputs:\*\*   
 - Input: \*\*Email template\*\*, \*\*customer or order data\*\*, and \*\*email address\*\*.   
 - Output: \*\*Email delivery status\*\* and \*\*log entries\*\* for tracking.  
  
- \*\*Web Communication (HTTP/HTTPS)\*\*   
 - \*\*Description:\*\* The system uses \*\*HTTP/HTTPS\*\* protocols to serve web pages and APIs to both customers and administrators.   
 - \*\*Interaction Method:\*\* RESTful web services for API interactions and \*\*HTML/CSS/JavaScript\*\* for user-facing web pages.   
 - \*\*Inputs/Outputs:\*\*   
 - Input: HTTP requests with query parameters, headers, and payloads.   
 - Output: JSON or HTML responses, including \*\*session tokens\*\*, \*\*product data\*\*, \*\*order status\*\*, and \*\*payment records\*\*.  
  
- \*\*Payment Gateway Communication (FR-10, FR-11)\*\*   
 - \*\*Description:\*\* The system communicates with an external \*\*payment gateway API\*\* to handle \*\*payment creation\*\*, \*\*processing\*\*, and \*\*status updates\*\*.   
 - \*\*Interaction Method:\*\* RESTful API calls with secure authentication (e.g., API keys or OAuth tokens).   
 - \*\*Inputs/Outputs:\*\*   
 - Input: \*\*Payment Method\*\*, \*\*Payment Details\*\*, \*\*OrderID\*\*, and \*\*Transaction Details\*\*.   
 - Output: \*\*Payment confirmation\*\*, \*\*transaction logs\*\*, and \*\*updated order status\*\*.  
  
- \*\*Plugin Communication (FR-21, FR-22)\*\*   
 - \*\*Description:\*\* The system communicates with \*\*plugin APIs\*\* to manage plugin configurations and access their documentation.   
 - \*\*Interaction Method:\*\* RESTful or GraphQL API calls, depending on the plugin's interface.   
 - \*\*Inputs/Outputs:\*\*   
 - Input: \*\*PluginID\*\*, \*\*configuration parameters\*\*, and \*\*API requests\*\*.   
 - Output: \*\*Plugin status changes\*\*, \*\*configuration updates\*\*, and \*\*API documentation\*\* retrieval.  
  
---  
  
This chapter outlines the \*\*external interfaces\*\* the system must interact with, including \*\*user interfaces\*\*, \*\*software components\*\*, and \*\*communication protocols\*\*. Developers should ensure that these interfaces are clearly implemented and integrated to maintain system functionality and consistency with the defined \*\*functional requirements\*\*.

# Use Case

Use Case Name: Register as Customer   
Use Case ID: UC-01   
Actors: Customer, System   
Preconditions: The customer is not yet registered in the system. The system is operational and accessible.   
Postconditions: The customer is successfully registered in the system and receives a confirmation email.   
  
Main Flow:   
1. The customer accesses the registration page on the system.   
2. The customer enters their personal information, including name, email address, and password.   
3. The system validates the email format and checks if it is already registered.   
4. The system confirms that the password meets the required complexity criteria.   
5. The system creates a new customer account with the provided information.   
6. The system sends a confirmation email to the customer's email address.   
7. The customer clicks on the confirmation link in the email.   
8. The system verifies the confirmation link and updates the customer account status to "Active."   
9. The customer is redirected to a welcome page, and the registration process is complete.   
  
Alternative Flow:   
1. If the email format is invalid (e.g., missing domain), the system displays an error message and prompts the customer to correct the input.   
2. If the email is already registered, the system informs the customer and suggests resetting their password or logging in.   
3. If the password does not meet the complexity requirements, the system displays an error message and prompts the customer to choose a stronger password.   
4. If the confirmation email fails to be sent, the system logs the error and prompts the customer to check their email address or contact support.  
  
Use Case Name: Login as Customer   
Use Case ID: UC-02   
Actors: Customer, System   
Preconditions: The customer has already registered and has an active account. The system is operational and accessible.   
Postconditions: The customer is successfully authenticated and redirected to their account dashboard.   
  
Main Flow:   
1. The customer navigates to the login page on the system.   
2. The customer enters their registered email address and password.   
3. The system validates the email and password against the stored customer data.   
4. If the credentials are valid, the system authenticates the customer and creates a session.   
5. The system redirects the customer to their account dashboard.   
  
Alternative Flow:   
1. If the email is not registered, the system displays an error message and prompts the customer to register or check their email input.   
2. If the password is incorrect, the system displays an error message and allows the customer to retry, with a limit on the number of attempts.   
3. If the system detects multiple failed login attempts, it locks the account temporarily and notifies the customer via email.   
4. If the system is unable to validate credentials due to a technical error, it logs the issue and displays a generic error message, prompting the customer to try again later or contact support.  
  
Use Case Name: Logout as Customer   
Use Case ID: UC-03   
Actors: Customer, System   
Preconditions: The customer is logged into their account. The system is operational and accessible.   
Postconditions: The customer session is terminated, and they are redirected to the login or home page.   
  
Main Flow:   
1. The customer navigates to the account settings or profile menu.   
2. The customer selects the "Logout" option.   
3. The system terminates the active session and clears session-related data.   
4. The system redirects the customer to the login page or home page.   
5. The logout process is confirmed by displaying a message or removing the logged-in state indicators.   
  
Alternative Flow:   
1. If the customer is not logged in, the system displays an error message indicating that no active session exists.   
2. If the system fails to terminate the session due to a technical error, it logs the issue and displays a generic error message, prompting the customer to try again or contact support.  
  
Use Case Name: View Product Information   
Use Case ID: UC-04   
Actors: Customer, System   
Preconditions: The customer is logged in or not logged in. The system is operational and accessible. The product information is available in the database.   
Postconditions: The customer is able to view detailed product information, including product name, description, price, and availability.   
  
Main Flow:   
1. The customer navigates to the product catalog or searches for a specific product.   
2. The customer selects a product from the list or search results.   
3. The system retrieves the product information from the database.   
4. The system displays the product details on a dedicated product information page.   
5. The customer reviews the product information.   
  
Alternative Flow:   
1. If the product is not available in the database, the system displays an error message and suggests checking the product name or browsing other products.   
2. If the system fails to load the product information due to a technical error, it logs the issue and displays a generic error message, prompting the customer to try again later or contact support.   
3. If the product is out of stock, the system displays a notification indicating the current stock status and may suggest similar products or an option to be notified when available.  
  
Use Case Name: Add Product to Cart   
Use Case ID: UC-05   
Actors: Customer, System   
Preconditions: The customer is logged in or not logged in. The system is operational and accessible. The customer has selected a product to add to the cart. The product is available in the database and in stock.   
Postconditions: The selected product is successfully added to the customer's shopping cart. The cart is updated to reflect the addition.   
  
Main Flow:   
1. The customer views a product information page.   
2. The customer clicks the "Add to Cart" button.   
3. The system checks the product's availability and stock status.   
4. The system adds the product to the customer's shopping cart.   
5. The system updates the cart display to show the added product and the new total.   
6. The system provides a confirmation message that the product was added to the cart.   
  
Alternative Flow:   
1. If the product is out of stock, the system displays a message indicating that it cannot be added to the cart and may suggest similar products.   
2. If the system fails to add the product due to a technical error, it logs the issue and displays a generic error message, prompting the customer to try again or contact support.   
3. If the customer is not logged in, the system stores the cart information temporarily and prompts the customer to log in or register to save the cart permanently.  
  
Use Case Name: Create Order   
Use Case ID: UC-06   
Actors: Customer, System, Product, Order, Payment, Administrator   
Preconditions: The customer is logged in. The shopping cart contains at least one product. The product information is valid and in stock. The system is operational and accessible. The payment method is configured.   
Postconditions: The order is successfully created in the system. The product inventory is updated. The customer receives a confirmation email for the order. The payment is processed or marked for processing.   
  
Main Flow:   
1. The customer accesses the shopping cart page.   
2. The customer clicks the "Proceed to Checkout" button.   
3. The system prompts the customer to enter or confirm their shipping address and contact details.   
4. The customer selects a payment method and confirms the order.   
5. The system verifies the product availability and updates the inventory.   
6. The system creates a new order record with the selected products, quantity, and pricing.   
7. The system initiates the payment processing via the configured payment gateway.   
8. The system sends a confirmation email to the customer with the order details.   
9. The system updates the order status to "Processing."   
  
Alternative Flow:   
1. If the customer does not provide a valid shipping address, the system displays an error message and prompts the customer to correct it.   
2. If the selected payment method is not available or fails, the system displays an error message and allows the customer to choose an alternative method.   
3. If the product is no longer in stock during checkout, the system alerts the customer and offers options to modify the order or remove the product.   
4. If the system fails to create the order due to a technical error, it logs the issue and displays a generic error message, prompting the customer to try again or contact support.   
5. If the email fails to be sent, the system logs the error and displays a message for the customer to check their email or contact support.  
  
Use Case Name: View Order Details   
Use Case ID: UC-07   
Actors: Customer, System, Order, Email   
Preconditions: The customer is logged in. The system is operational and accessible. The customer has at least one order in the system.   
Postconditions: The customer is able to view detailed information about a specific order, including product list, order status, and total amount.   
  
Main Flow:   
1. The customer navigates to the "My Orders" section on their account dashboard.   
2. The customer selects a specific order from the list by clicking on it.   
3. The system retrieves the order details from the database.   
4. The system displays the order information, including product list, quantities, prices, order date, and status.   
5. The customer reviews the displayed order details.   
  
Alternative Flow:   
1. If the selected order is not found in the database, the system displays an error message and suggests checking the order ID or browsing other orders.   
2. If the system fails to load the order details due to a technical error, it logs the issue and displays a generic error message, prompting the customer to try again later or contact support.   
3. If the order details are incomplete or corrupted, the system displays a warning and provides the available data while suggesting contacting support for further assistance.  
  
Use Case Name: Update Order Status   
Use Case ID: UC-08   
Actors: Administrator, System, Order   
Preconditions: The administrator is logged in. The system is operational and accessible. The order exists in the system and has a current status.   
Postconditions: The order status is successfully updated in the system. The customer is notified via email of the status change.   
  
Main Flow:   
1. The administrator navigates to the "Order Management" section in the admin dashboard.   
2. The administrator selects a specific order from the list.   
3. The system displays the order details, including the current status.   
4. The administrator chooses a new status from the available options (e.g., Processing, Shipped, Delivered, Cancelled).   
5. The administrator confirms the status update.   
6. The system updates the order status in the database.   
7. The system generates and sends a notification email to the customer with the updated status.   
8. The system displays a confirmation message to the administrator that the status has been updated.   
  
Alternative Flow:   
1. If the selected order does not exist in the database, the system displays an error message and suggests verifying the order ID.   
2. If the new status is not valid for the selected order, the system displays an error and explains the available status transitions.   
3. If the system fails to update the order status due to a technical error, it logs the issue and displays a generic error message, prompting the administrator to try again or contact support.   
4. If the email notification fails to be sent, the system logs the error and displays a message to the administrator to check the email configuration or retry the update.  
  
Use Case Name: Delete Order   
Use Case ID: UC-09   
Actors: Customer, System, Order, Email, Administrator   
Preconditions: The customer is logged in. The system is operational and accessible. The customer has at least one order in the system. The order is in a cancellable status (e.g., Processing).   
Postconditions: The selected order is successfully deleted from the system. The product inventory is updated accordingly. The customer is notified via email of the deletion. The order status is changed to "Cancelled."   
  
Main Flow:   
1. The customer navigates to the "My Orders" section on their account dashboard.   
2. The customer selects a specific order and clicks the "Delete" or "Cancel" button.   
3. The system prompts the customer to confirm the deletion of the order.   
4. The customer confirms the deletion.   
5. The system verifies that the order is in a cancellable status.   
6. The system removes the order from the database and updates the product inventory.   
7. The system sends a confirmation email to the customer indicating the order has been deleted.   
8. The system updates the order status to "Cancelled."   
9. The system displays a confirmation message to the customer that the order was successfully deleted.   
  
Alternative Flow:   
1. If the customer does not confirm the deletion, the system cancels the action and returns to the order details page.   
2. If the order is not in a cancellable status (e.g., Shipped or Delivered), the system displays an error message and does not allow deletion.   
3. If the system fails to delete the order due to a technical error, it logs the issue and displays a generic error message, prompting the customer to try again or contact support.   
4. If the email notification fails to be sent, the system logs the error and displays a message to the customer to check their email or contact support.   
5. If the order contains products that cannot be restocked (e.g., digital products), the system displays a warning before deletion and updates the inventory accordingly.  
  
Use Case Name: Create Payment   
Use Case ID: UC-10   
Actors: Customer, System, Payment, Order, Administrator, Email   
Preconditions: The customer is logged in. An order has been created and is in the "Processing" status. The system is operational and accessible. The payment gateway is configured and available.   
Postconditions: The payment is successfully created and processed. The order status is updated to "Paid." The customer receives a payment confirmation email.   
  
Main Flow:   
1. The customer navigates to the payment confirmation page after completing an order.   
2. The system displays the total amount due and the selected payment method.   
3. The customer confirms the payment.   
4. The system initiates the payment transaction using the configured payment gateway API.   
5. The payment gateway processes the transaction and returns a success status.   
6. The system records the payment details in the database and links them to the corresponding order.   
7. The system updates the order status to "Paid."   
8. The system sends a confirmation email to the customer with the payment and order details.   
9. The customer is redirected to a payment success page with a summary of the transaction.   
  
Alternative Flow:   
1. If the payment gateway returns an error or fails to process the transaction, the system displays an error message and allows the customer to retry or choose an alternative payment method.   
2. If the customer cancels the payment before confirmation, the system returns to the order details page without processing the payment.   
3. If the system fails to update the order status due to a technical error, it logs the issue and displays a generic error message, prompting the customer to try again or contact support.   
4. If the confirmation email fails to be sent, the system logs the error and displays a message for the customer to check their email or contact support.   
5. If the payment is partially processed or requires manual verification, the system updates the payment status to "Pending" and notifies the administrator for further action.  
  
Use Case Name: Process Payment   
Use Case ID: UC-11   
Actors: Customer, System, Payment, Order, Administrator, Email, API, Plugin, Documentation   
Preconditions: The customer is logged in. An order is in the "Processing" status and ready for payment. The payment gateway API is configured and operational. The system is accessible and functioning.   
Postconditions: The payment is successfully processed and recorded in the system. The order status is updated to "Paid." The customer receives a payment confirmation email. If needed, the system logs payment details for review by the administrator.   
  
Main Flow:   
1. The customer navigates to the payment page after creating an order.   
2. The system displays the total amount due, the selected products, and the chosen payment method.   
3. The customer confirms the payment and provides necessary payment details (e.g., card information).   
4. The system validates the payment information and initiates a transaction using the configured payment gateway API.   
5. The payment gateway processes the transaction and returns a success or failure response.   
6. If successful, the system records the payment in the database and links it to the corresponding order.   
7. The system updates the order status to "Paid" and logs the payment transaction.   
8. The system sends a confirmation email to the customer with the payment and order details.   
9. The customer is redirected to a payment success page with a summary of the transaction.   
  
Alternative Flow:   
1. If the payment gateway returns an error or the transaction is declined, the system displays an error message and allows the customer to retry or choose an alternative payment method.   
2. If the customer cancels the payment before completion, the system returns to the order details page without processing the payment.   
3. If the system fails to record the payment due to a technical error, it logs the issue and displays a generic error message, prompting the customer to try again or contact support.   
4. If the confirmation email fails to be sent, the system logs the error and displays a message for the customer to check their email or contact support.   
5. If the payment requires manual verification (e.g., due to fraud detection), the system updates the payment status to "Pending" and notifies the administrator via the system for review.   
6. If the payment method is not supported by the configured plugin or API, the system displays a message and suggests using a supported method.   
7. If the customer inputs invalid payment information, the system displays an error message and prompts for correction.  
  
Use Case Name: View Payment History   
Use Case ID: UC-12   
Actors: Customer, System, Payment, Order, Email   
Preconditions: The customer is logged in. The system is operational and accessible. The customer has at least one completed payment in the system.   
Postconditions: The customer is able to view their payment history, including payment dates, amounts, and associated order information.   
  
Main Flow:   
1. The customer navigates to the "Payment History" section on their account dashboard.   
2. The system retrieves all completed payment records associated with the customer’s account from the database.   
3. The system displays the payment history in a table or list format, including payment date, amount, payment method, and the corresponding order ID.   
4. The customer reviews their payment history.   
5. The system allows the customer to click on a specific payment record to view more detailed information.   
6. The system displays the detailed payment information, including transaction ID, status, and payment confirmation.   
  
Alternative Flow:   
1. If no completed payment records are found for the customer, the system displays a message indicating that there is no payment history available.   
2. If the system fails to retrieve the payment history due to a technical error, it logs the issue and displays a generic error message, prompting the customer to try again later or contact support.   
3. If the customer selects a payment record that cannot be displayed due to missing or corrupted data, the system shows a warning and provides the available information, while suggesting contacting support.   
4. If the system is unable to load detailed payment information, it logs the issue and displays an error message, allowing the customer to return to the main payment history view.  
  
Use Case Name: Register as Administrator   
Use Case ID: UC-13   
Actors: Administrator, System   
Preconditions: The administrator is not yet registered in the system. The system is operational and accessible. The administrator has access to the registration interface for administrators.   
Postconditions: The administrator is successfully registered in the system and receives a confirmation email. The account is activated after email confirmation.   
  
Main Flow:   
1. The administrator accesses the administrator registration page on the system.   
2. The administrator enters their personal information, including name, email address, and password.   
3. The system validates the email format and checks if it is already registered.   
4. The system confirms that the password meets the required complexity criteria.   
5. The system creates a new administrator account with the provided information.   
6. The system sends a confirmation email to the administrator's email address.   
7. The administrator clicks on the confirmation link in the email.   
8. The system verifies the confirmation link and updates the administrator account status to "Active."   
9. The administrator is redirected to the admin dashboard, and the registration process is complete.   
  
Alternative Flow:   
1. If the email format is invalid (e.g., missing domain), the system displays an error message and prompts the administrator to correct the input.   
2. If the email is already registered, the system informs the administrator and suggests resetting their password or logging in.   
3. If the password does not meet the complexity requirements, the system displays an error message and prompts the administrator to choose a stronger password.   
4. If the confirmation email fails to be sent, the system logs the error and prompts the administrator to check their email address or contact support.  
  
Use Case Name: Login as Administrator   
Use Case ID: UC-14   
Actors: Administrator, System   
Preconditions: The administrator has already registered and has an active account. The system is operational and accessible.   
Postconditions: The administrator is successfully authenticated and redirected to the admin dashboard.   
  
Main Flow:   
1. The administrator navigates to the login page on the system.   
2. The administrator enters their registered email address and password.   
3. The system validates the email and password against the stored administrator data.   
4. If the credentials are valid, the system authenticates the administrator and creates a session.   
5. The system redirects the administrator to the admin dashboard.   
  
Alternative Flow:   
1. If the email is not registered, the system displays an error message and prompts the administrator to check their email input or contact support.   
2. If the password is incorrect, the system displays an error message and allows the administrator to retry, with a limit on the number of attempts.   
3. If the system detects multiple failed login attempts, it locks the account temporarily and notifies the administrator via email.   
4. If the system is unable to validate credentials due to a technical error, it logs the issue and displays a generic error message, prompting the administrator to try again later or contact support.  
  
Use Case Name: Logout as Administrator   
Use Case ID: UC-15   
Actors: Administrator, System   
Preconditions: The administrator is logged into their account. The system is operational and accessible.   
Postconditions: The administrator session is terminated, and they are redirected to the login or home page.   
  
Main Flow:   
1. The administrator navigates to the account settings or profile menu.   
2. The administrator selects the "Logout" option.   
3. The system terminates the active session and clears session-related data.   
4. The system redirects the administrator to the login page or home page.   
5. The logout process is confirmed by displaying a message or removing the logged-in state indicators.   
  
Alternative Flow:   
1. If the administrator is not logged in, the system displays an error message indicating that no active session exists.   
2. If the system fails to terminate the session due to a technical error, it logs the issue and displays a generic error message, prompting the administrator to try again or contact support.  
  
Use Case Name: Add Product   
Use Case ID: UC-04   
Actors: Administrator, System, Product, Email   
Preconditions: The administrator is logged in. The system is operational and accessible. The product information is valid and not already in the database.   
Postconditions: The product is successfully added to the system database. The administrator receives a confirmation message. The product is available for viewing and purchasing by customers.   
  
Main Flow:   
1. The administrator navigates to the "Product Management" section in the admin dashboard.   
2. The administrator clicks the "Add New Product" button.   
3. The system displays a product information form with fields for product name, description, price, category, and image upload.   
4. The administrator fills out the form with the required product details.   
5. The administrator uploads a product image or selects an existing one.   
6. The administrator submits the form.   
7. The system validates the input data for completeness and correctness.   
8. The system saves the new product information to the database.   
9. The system displays a confirmation message indicating the product was added successfully.   
  
Alternative Flow:   
1. If the product name is already in the database, the system displays an error message and suggests editing the existing product instead.   
2. If the product information is incomplete (e.g., missing price or description), the system displays an error and prompts the administrator to fill in the missing fields.   
3. If the uploaded image is invalid or exceeds the file size limit, the system displays an error and prompts the administrator to upload a valid image.   
4. If the system fails to save the product due to a technical error, it logs the issue and displays a generic error message, prompting the administrator to try again or contact support.  
  
Use Case Name: Update Product Information   
Use Case ID: UC-16   
Actors: Administrator, System, Product, Email   
Preconditions: The administrator is logged in. The system is operational and accessible. The product to be updated exists in the database. The administrator has access to the product management interface.   
Postconditions: The product information is successfully updated in the system. The administrator receives a confirmation message. The updated product is available for viewing and purchasing by customers.   
  
Main Flow:   
1. The administrator navigates to the "Product Management" section in the admin dashboard.   
2. The administrator selects a specific product from the list to update.   
3. The system displays the product's current information in an editable form.   
4. The administrator modifies the product details, such as name, description, price, or availability.   
5. The administrator uploads a new product image if needed or selects an existing one.   
6. The administrator submits the updated form.   
7. The system validates the updated input data for completeness and correctness.   
8. The system updates the product information in the database.   
9. The system displays a confirmation message indicating the product was updated successfully.   
  
Alternative Flow:   
1. If the product does not exist in the database, the system displays an error message and suggests verifying the product ID or adding a new product instead.   
2. If the updated product information is incomplete or invalid (e.g., missing price, invalid image format), the system displays an error message and prompts the administrator to correct the input.   
3. If the system fails to update the product due to a technical error, it logs the issue and displays a generic error message, prompting the administrator to try again or contact support.   
4. If the confirmation email for the update fails to be sent, the system logs the error and displays a message for the administrator to check the email configuration or retry the update.  
  
Use Case Name: Delete Product   
Use Case ID: UC-17   
Actors: Administrator, System, Product, Email   
Preconditions: The administrator is logged in. The system is operational and accessible. The product to be deleted exists in the database. The product is not associated with any active orders.   
Postconditions: The product is successfully deleted from the system. The administrator receives a confirmation message. The product is no longer available for viewing or purchasing by customers.   
  
Main Flow:   
1. The administrator navigates to the "Product Management" section in the admin dashboard.   
2. The administrator selects a specific product from the list and clicks the "Delete" button.   
3. The system prompts the administrator to confirm the deletion.   
4. The administrator confirms the deletion.   
5. The system checks whether the product is linked to any active orders.   
6. If no active orders are found, the system deletes the product from the database.   
7. The system displays a confirmation message indicating the product was successfully deleted.   
  
Alternative Flow:   
1. If the administrator does not confirm the deletion, the system cancels the action and returns to the product management page.   
2. If the product is associated with an active or paid order, the system displays an error message and prevents deletion, suggesting that the product must be removed from all orders before proceeding.   
3. If the product does not exist in the database, the system displays an error message and suggests verifying the product ID or checking the product list.   
4. If the system fails to delete the product due to a technical error, it logs the issue and displays a generic error message, prompting the administrator to try again or contact support.  
  
Use Case Name: View Email Template   
Use Case ID: UC-18   
Actors: Administrator, System, Email, Plugin, Documentation   
Preconditions: The administrator is logged in. The system is operational and accessible. The email template exists in the system and is associated with a specific use case (e.g., order confirmation, payment confirmation, registration confirmation). The plugin or API for email functionality is properly configured.   
Postconditions: The administrator is able to view the content and structure of the selected email template. The system displays the template without modifying it.   
  
Main Flow:   
1. The administrator navigates to the "Email Templates" section in the admin dashboard.   
2. The administrator selects a specific email template from the list (e.g., "Order Confirmation Template").   
3. The system retrieves the selected email template from the database.   
4. The system displays the template content, including subject, body, and placeholders for dynamic data.   
5. The administrator reviews the template details.   
  
Alternative Flow:   
1. If the selected email template does not exist in the database, the system displays an error message and suggests verifying the template name or selecting another template.   
2. If the system fails to load the email template due to a technical error, it logs the issue and displays a generic error message, prompting the administrator to try again later or contact support.   
3. If the email template is empty or corrupted, the system displays a warning and provides the available data while suggesting contacting support for further assistance.  
  
Use Case Name: Send Order Confirmation Email   
Use Case ID: UC-19   
Actors: System, Order, Email, Plugin, API   
Preconditions: The system has an order in the "Paid" status. The email plugin and API are configured and operational. The customer's email address is valid and stored in the system.   
Postconditions: The customer receives an order confirmation email with the order details. The system logs the email delivery status.   
  
Main Flow:   
1. The system identifies an order with the status "Paid."   
2. The system retrieves the order details and customer information from the database.   
3. The system loads the appropriate email template for order confirmation.   
4. The system populates the template with dynamic data, such as order ID, product list, and total amount.   
5. The system uses the configured email plugin or API to send the confirmation email to the customer.   
6. The system logs the successful delivery of the email.   
  
Alternative Flow:   
1. If the email address is invalid or missing, the system logs the error and displays a message for the administrator to correct the data.   
2. If the email template is missing or corrupted, the system logs the issue and displays a warning to the administrator.   
3. If the email plugin or API is not operational, the system logs the error and displays a message for the administrator to check the configuration or retry.   
4. If the system fails to send the email due to a technical error, it logs the issue and displays a generic error message to the administrator for further action.  
  
Use Case Name: Manage Plugin   
Use Case ID: UC-20   
Actors: Administrator, System, Plugin, API, Documentation   
Preconditions: The administrator is logged in. The system is operational and accessible. The plugin or API to be managed is registered in the system.   
Postconditions: The plugin or API is successfully enabled, disabled, or configured in the system. The administrator receives a confirmation message. The system updates its behavior based on the plugin status.   
  
Main Flow:   
1. The administrator navigates to the "Plugin Management" section in the admin dashboard.   
2. The administrator selects a specific plugin or API from the list of available plugins.   
3. The system displays the plugin details, including its status, configuration options, and associated documentation.   
4. The administrator chooses to enable, disable, or configure the plugin.   
5. The administrator provides any necessary configuration details or confirms the action.   
6. The system updates the plugin status or configuration in the database.   
7. The system displays a confirmation message indicating the plugin has been successfully managed.   
  
Alternative Flow:   
1. If the selected plugin does not exist in the system, the system displays an error message and suggests verifying the plugin name.   
2. If the configuration data provided is invalid or incomplete, the system displays an error and prompts the administrator to correct the input.   
3. If the system fails to update the plugin due to a technical error, it logs the issue and displays a generic error message, prompting the administrator to try again or contact support.   
4. If the plugin requires additional documentation or setup steps, the system displays a warning and provides guidance for the administrator to follow.  
  
Use Case Name: View API Documentation   
Use Case ID: UC-21   
Actors: Administrator, System, API, Plugin, Documentation   
Preconditions: The administrator is logged in. The system is operational and accessible. The API documentation exists and is properly formatted. The plugin responsible for accessing documentation is configured and active.   
Postconditions: The administrator is able to view the API documentation. The system displays the documentation in a readable format. The administrator can navigate through the documentation sections.   
  
Main Flow:   
1. The administrator navigates to the "API Management" section in the admin dashboard.   
2. The administrator selects the "View API Documentation" option.   
3. The system retrieves the API documentation from the designated storage or external source.   
4. The system displays the documentation in a structured format, including endpoints, parameters, and usage examples.   
5. The administrator reviews the documentation and can search or navigate to specific sections.   
6. The system provides links to related plugins or APIs if applicable.   
  
Alternative Flow:   
1. If the API documentation is not found, the system displays an error message and suggests verifying the API name or configuration.   
2. If the documentation is not accessible due to a plugin misconfiguration, the system logs the issue and displays a warning to the administrator.   
3. If the system fails to load the documentation due to a technical error, it logs the issue and displays a generic error message, prompting the administrator to try again later or contact support.   
4. If the documentation is outdated or needs to be refreshed, the system displays a notification and allows the administrator to trigger an update.  
  
Use Case Name: View Product Information   
Use Case ID: UC-04   
Actors: Customer, System   
Preconditions: The customer is logged in or not logged in. The system is operational and accessible. The product information is available in the database.   
Postconditions: The customer is able to view detailed product information, including product name, description, price, and availability.   
  
Main Flow:   
1. The customer navigates to the product catalog or searches for a specific product.   
2. The customer selects a product from the list or search results.   
3. The system retrieves the product information from the database.   
4. The system displays the product details on a dedicated product information page.   
5. The customer reviews the product information.   
  
Alternative Flow:   
1. If the product is not available in the database, the system displays an error message and suggests checking the product name or browsing other products.   
2. If the system fails to load the product information due to a technical error, it logs the issue and displays a generic error message, prompting the customer to try again later or contact support.   
3. If the product is out of stock, the system displays a notification indicating the current stock status and may suggest similar products or an option to be notified when available.  
  
Use Case Name: Add Product to Cart   
Use Case ID: UC-22   
Actors: Customer, System   
Preconditions: The customer is logged in or not logged in. The system is operational and accessible. The customer has selected a product to add to the cart. The product is available in the database and in stock.   
Postconditions: The selected product is successfully added to the customer's shopping cart. The cart is updated to reflect the addition.   
  
Main Flow:   
1. The customer views a product information page.   
2. The customer clicks the "Add to Cart" button.   
3. The system checks the product's availability and stock status.   
4. The system adds the product to the customer's shopping cart.   
5. The system updates the cart display to show the added product and the new total.   
6. The system provides a confirmation message that the product was added to the cart.   
  
Alternative Flow:   
1. If the product is out of stock, the system displays a message indicating that it cannot be added to the cart and may suggest similar products.   
2. If the system fails to add the product due to a technical error, it logs the issue and displays a generic error message, prompting the customer to try again or contact support.   
3. If the customer is not logged in, the system stores the cart information temporarily and prompts the customer to log in or register to save the cart permanently.  
  
It seems that the use case "Create Order" (UC-06) has already been defined in your provided list. Could you please clarify if you meant to request a different use case or if there is a specific aspect of UC-06 that needs to be modified or expanded? I’m here to help!  
  
Use Case Name: View Order Details   
Use Case ID: UC-23   
Actors: Customer, System, Order, Email   
Preconditions: The customer is logged in. The system is operational and accessible. The customer has at least one order in the system.   
Postconditions: The customer is able to view detailed information about a specific order, including product list, order status, and total amount.   
  
Main Flow:   
1. The customer accesses their account dashboard.   
2. The customer selects the "My Orders" option from the dashboard menu.   
3. The system displays a list of the customer's previous and current orders.   
4. The customer clicks on a specific order to view its details.   
5. The system retrieves the selected order's information from the database.   
6. The system displays the order details, including order ID, product list, quantities, prices, order date, and current status.   
7. The customer reviews the displayed information.   
  
Alternative Flow:   
1. If the selected order does not exist in the database, the system displays an error message and suggests checking the order ID or browsing other orders.   
2. If the system fails to retrieve the order details due to a technical error, it logs the issue and displays a generic error message, prompting the customer to try again later or contact support.   
3. If the order details are incomplete or corrupted, the system displays a warning and provides the available data while suggesting contacting support for further assistance.  
  
Use Case Name: Update Order Status   
Use Case ID: UC-24   
Actors: Administrator, System, Order, Email   
Preconditions: The administrator is logged in. The system is operational and accessible. The order to be updated exists in the database. The order is in a status that allows for updates (e.g., Processing, Shipped).   
Postconditions: The order status is successfully updated in the system. The customer is notified via email of the status change. The updated status is reflected in the order history.   
  
Main Flow:   
1. The administrator navigates to the "Order Management" section in the admin dashboard.   
2. The administrator selects a specific order from the list.   
3. The system displays the order details, including the current status.   
4. The administrator selects a new status from the available options (e.g., Shipped, Delivered, Cancelled).   
5. The system validates the transition from the current status to the new status.   
6. The administrator confirms the update.   
7. The system updates the order status in the database.   
8. The system generates and sends a notification email to the customer with the updated status.   
9. The system displays a confirmation message to the administrator that the status was successfully updated.   
  
Alternative Flow:   
1. If the selected order does not exist in the database, the system displays an error message and suggests verifying the order ID or checking the order list.   
2. If the new status is not valid for the current order status (e.g., attempting to change to "Shipped" when the order is already "Delivered"), the system displays an error and explains the allowed status transitions.   
3. If the system fails to update the order status due to a technical error, it logs the issue and displays a generic error message, prompting the administrator to try again or contact support.   
4. If the email notification fails to be sent, the system logs the error and displays a message to the administrator to check the email configuration or retry the update.   
5. If the administrator cancels the update before confirmation, the system reverts to the order details page without making any changes.  
  
Use Case Name: Delete Order   
Use Case ID: UC-25   
Actors: Administrator, System, Order, Email, Customer   
Preconditions: The administrator is logged in. The system is operational and accessible. The order to be deleted exists in the database. The order is in a deletable status (e.g., Processing, Cancelled). The customer's email address is valid and stored in the system.   
Postconditions: The selected order is successfully deleted from the system. The product inventory is updated accordingly. The customer is notified via email of the deletion. The order status is changed to "Deleted."   
  
Main Flow:   
1. The administrator navigates to the "Order Management" section in the admin dashboard.   
2. The administrator selects a specific order from the list and clicks the "Delete" button.   
3. The system prompts the administrator to confirm the deletion of the order.   
4. The administrator confirms the deletion.   
5. The system verifies that the order is in a deletable status.   
6. The system removes the order from the database and updates the product inventory to add back the removed products.   
7. The system sends a confirmation email to the customer indicating the order has been deleted.   
8. The system updates the order status to "Deleted."   
9. The system displays a confirmation message to the administrator that the order was successfully deleted.   
  
Alternative Flow:   
1. If the administrator does not confirm the deletion, the system cancels the action and returns to the order management page.   
2. If the order is not in a deletable status (e.g., Shipped or Delivered), the system displays an error message and does not allow deletion, suggesting that the order must be in a valid status for deletion.   
3. If the system fails to delete the order due to a technical error, it logs the issue and displays a generic error message, prompting the administrator to try again or contact support.   
4. If the email notification fails to be sent, the system logs the error and displays a message to the administrator to check the email configuration or retry the deletion.   
5. If the order contains products that cannot be restocked (e.g., digital products or items with a limited inventory), the system displays a warning before deletion and updates the inventory accordingly.