项目文档

# Functional Requirement

1. Functional Requirements   
  
1.1 Customer Account Registration Function   
 Function ID: FR-01   
 Description: Customers can create a new account by providing personal information such as name, email, password, and contact details. The system must validate the input data, ensure the email is unique, and send a confirmation message or email.   
 Input: Name, email address, password, and contact details.   
 Output: A new customer account stored in the system, and a confirmation message or email sent to the customer.   
  
1.2 Customer Login Function   
 Function ID: FR-02   
 Description: Registered customers can log in using their email and password. The system must validate the credentials, initiate a session, and redirect the customer to the homepage or shopping cart.   
 Input: Email address and password.   
 Output: An authenticated session initiated for the customer, and redirection to the homepage or shopping cart.   
  
1.3 Customer Logout Function   
 Function ID: FR-03   
 Description: Customers can terminate their session by logging out. The system must clear session data and redirect the customer to the homepage or login page.   
 Input: Customer session or logout request.   
 Output: Session terminated, and customer redirected to the homepage or login page.   
  
1.4 Product Detail View Function   
 Function ID: FR-04   
 Description: Customers can view detailed information of a selected product, including name, description, price, images, and availability.   
 Input: Product ID or search term.   
 Output: Displayed product details in a user-friendly format.   
  
1.5 Product Search Function   
 Function ID: FR-05   
 Description: Customers can search for products using terms such as product name, category, or keywords. The system must display matching products and allow selection.   
 Input: Search term (e.g., product name, keyword).   
 Output: List of products matching the search criteria displayed to the customer.   
  
1.6 Add Product to Shopping Cart Function   
 Function ID: FR-06   
 Description: Customers can add a product to their shopping cart by specifying the desired quantity. The system must verify product availability in the inventory, update the cart, and reflect the updated stock.   
 Input: Product ID and quantity.   
 Output: Updated Shopping Cart and Inventory stock levels, and a confirmation message.   
  
1.7 Update Shopping Cart Function   
 Function ID: FR-07   
 Description: Customers can modify the quantity of items in their shopping cart or remove items. The system must update the cart and inventory accordingly, and recalculate the total cost.   
 Input: Cart item ID, new quantity, or removal request.   
 Output: Updated Shopping Cart and Inventory stock levels, and a confirmation message.   
  
1.8 Remove Product from Shopping Cart Function   
 Function ID: FR-08   
 Description: Customers can remove a product from their shopping cart. The system must update the cart and inventory to restore the product's stock level.   
 Input: Cart item ID and removal request.   
 Output: Updated Shopping Cart and Inventory stock levels, and a confirmation message.   
  
1.9 View Shopping Cart Summary Function   
 Function ID: FR-09   
 Description: Customers can view a summary of their shopping cart, including product names, quantities, prices, and total cost. The system must also display stock availability and warnings for low or out-of-stock items.   
 Input: Shopping Cart ID or session identifier.   
 Output: Cart summary displayed, including product details, total cost, and availability status.   
  
1.10 Checkout Function   
 Function ID: FR-10   
 Description: Customers can proceed to checkout by entering or confirming their shipping and billing information and selecting a payment method. The system must validate the data, process the payment, create an order, and clear the cart.   
 Input: Shopping Cart ID, shipping and billing information, selected payment method.   
 Output: Confirmed order stored in the system, updated Inventory, cleared Shopping Cart, and a confirmation message or email sent to the customer.   
  
1.11 Order Confirmation Function   
 Function ID: FR-11   
 Description: After completing the checkout, the system must confirm the order and update the payment and inventory accordingly.   
 Input: Checkout data and validated payment information.   
 Output: Confirmed order status, updated Payment and Inventory entities, and a confirmation message or email sent to the customer.   
  
1.12 Order Confirmation Email Function   
 Function ID: FR-12   
 Description: The system must generate and send an email to the customer confirming the order details, such as order number, items purchased, total cost, and estimated delivery time.   
 Input: Order details and customer email address.   
 Output: Order confirmation email sent to the customer and a log entry indicating the email was sent.   
  
1.13 Administrator Login Function   
 Function ID: FR-13   
 Description: Administrators can log in using their username and password to access the admin dashboard. The system must validate the credentials and initiate a session.   
 Input: Administrator username and password.   
 Output: An authenticated session initiated for the administrator, and redirection to the admin dashboard.   
  
1.14 Administrator Logout Function   
 Function ID: FR-14   
 Description: Administrators can terminate their session by logging out. The system must clear session data and redirect the administrator to the homepage or login page.   
 Input: Administrator session or logout request.   
 Output: Session terminated, and administrator redirected to the homepage or login page.   
  
1.15 Inventory Management Function   
 Function ID: FR-15   
 Description: Administrators can view, update, or modify the stock levels and status of products. The system must validate input and reflect changes in the inventory and product information.   
 Input: Product ID, new stock level, or inventory status changes.   
 Output: Updated Inventory and Product entities, and a confirmation message.   
  
1.16 Product Information Update Function   
 Function ID: FR-16   
 Description: Administrators can update a product’s details such as name, price, description, and availability. The system must validate the input and reflect the changes in the product entity and inventory.   
 Input: Product ID and updated details (e.g., price, description, stock level).   
 Output: Updated Product and Inventory entities, and a confirmation message.   
  
1.17 Product Categorization Function   
 Function ID: FR-17   
 Description: Administrators can assign products to one or more categories. The system must validate the selected categories and update the product and inventory data accordingly.   
 Input: Product ID and selected category(ies).   
 Output: Updated Product entity with category information, and confirmation message.   
  
1.18 Plugin Development Function   
 Function ID: FR-18   
 Description: Administrators can develop new or update existing plugins. The system must validate the plugin code, register it in the database, and link it to relevant modules.   
 Input: Plugin code and configuration files.   
 Output: Registered Plugin entity and confirmation message of successful deployment.   
  
1.19 Plugin Configuration Function   
 Function ID: FR-19   
 Description: Administrators can configure plugin settings such as API keys and behavior rules. The system must validate the configuration data and update the plugin entity accordingly.   
 Input: Plugin ID and configuration parameters.   
 Output: Updated Plugin entity, and confirmation message of successful configuration.   
  
1.20 Customer Management Function   
 Function ID: FR-20   
 Description: Administrators can view, edit, or delete customer account information. The system must validate input data and update the customer entity in the database.   
 Input: Customer ID and updated or deleted information.   
 Output: Updated or deleted Customer entity, and confirmation message.   
  
1.21 Product Management Function   
 Function ID: FR-21   
 Description: Administrators can add, edit, or delete product records. The system must validate product data and update the product and inventory entities accordingly.   
 Input: Product ID and new or modified product data.   
 Output: Updated or deleted Product entity, and updated Inventory if applicable.   
  
1.22 Order Management Function   
 Function ID: FR-22   
 Description: Administrators can view, update, or cancel orders. The system must validate input, update the order, inventory, and payment entities accordingly, and send notifications if necessary.   
 Input: Order ID and requested action (e.g., update, cancel).   
 Output: Updated or canceled Order entity, updated Inventory and Payment entities, and a confirmation message.   
  
1.23 Payment Management Function   
 Function ID: FR-23   
 Description: Administrators can view, update, or process refunds for payments. The system must validate input and update the payment and order entities accordingly.   
 Input: Payment ID and requested action (e.g., update, refund).   
 Output: Updated Payment and Order entities, and a confirmation message.   
  
1.24 Administrator Management Function   
 Function ID: FR-24   
 Description: Administrators can add, edit, or remove other administrator accounts. The system must validate input and update the administrator entity in the database.   
 Input: Administrator ID and requested action (e.g., add, edit, delete).   
 Output: Updated or deleted Administrator entity, and a confirmation message.

# External Description

# 2. External Interfaces   
  
This section describes the external interfaces that the system must support to ensure proper functionality, usability, and integration with other systems or components. These interfaces include user interfaces, hardware interfaces, software interfaces, and communication interfaces. Each interface is described in detail, including its role, expected inputs, and outputs.   
  
## 2.1 User Interfaces   
  
The user interfaces of the system are primarily web-based and mobile-responsive, allowing customers and administrators to interact with the system through a browser or mobile application. The interfaces are designed to be intuitive and user-friendly, enabling seamless navigation and interaction.   
  
- \*\*Customer Login Interface\*\*:   
 - \*Description\*: A web-based interface where customers enter their email and password to authenticate and start a session.   
 - \*Inputs\*: Email address and password.   
 - \*Outputs\*: Authentication result, redirection to the homepage or shopping cart.   
  
- \*\*Customer Registration Interface\*\*:   
 - \*Description\*: A web-based interface for customers to provide personal information and create a new account.   
 - \*Inputs\*: Name, email address, password, and contact details.   
 - \*Outputs\*: Account creation confirmation, email verification prompt.   
  
- \*\*Product Detail Interface\*\*:   
 - \*Description\*: A web-based interface displaying detailed product information, including images, descriptions, price, and availability status.   
 - \*Inputs\*: Product ID or search term.   
 - \*Outputs\*: Product details, availability status, and stock warnings.   
  
- \*\*Product Search Interface\*\*:   
 - \*Description\*: A web-based interface that allows customers to search for products using keywords, product names, or categories.   
 - \*Inputs\*: Search term (e.g., product name, keyword).   
 - \*Outputs\*: A list of products matching the search criteria.   
  
- \*\*Shopping Cart Interface\*\*:   
 - \*Description\*: A web-based interface for customers to view, add, update, or remove items from their shopping cart.   
 - \*Inputs\*: Product ID and quantity for adding/updating, cart item ID for removal.   
 - \*Outputs\*: Updated cart summary, total cost, and stock availability status.   
  
- \*\*Checkout Interface\*\*:   
 - \*Description\*: A web-based interface where customers confirm their shipping and billing information, select a payment method, and proceed with the order.   
 - \*Inputs\*: Shopping Cart ID, shipping and billing information, selected payment method.   
 - \*Outputs\*: Order confirmation, updated inventory, cleared shopping cart.   
  
- \*\*Order Confirmation Interface\*\*:   
 - \*Description\*: A web-based interface that displays the order confirmation after checkout.   
 - \*Inputs\*: Order details and session identifier.   
 - \*Outputs\*: Order confirmation message, order number, and estimated delivery time.   
  
- \*\*Admin Dashboard Interface\*\*:   
 - \*Description\*: A web-based interface for administrators to manage products, inventory, customers, orders, and plugins.   
 - \*Inputs\*: Administrator username and password.   
 - \*Outputs\*: Access to the admin dashboard with various management modules.   
  
- \*\*Inventory Management Interface\*\*:   
 - \*Description\*: A part of the admin dashboard that allows administrators to view and modify product stock levels and statuses.   
 - \*Inputs\*: Product ID, new stock level, or status changes.   
 - \*Outputs\*: Updated inventory display, confirmation message.   
  
- \*\*Product Management Interface\*\*:   
 - \*Description\*: A part of the admin dashboard for administrators to add, update, or delete product records.   
 - \*Inputs\*: Product ID and new or modified product data.   
 - \*Outputs\*: Updated product information in the interface, confirmation message.   
  
- \*\*Order Management Interface\*\*:   
 - \*Description\*: A part of the admin dashboard where administrators can view, update, or cancel orders.   
 - \*Inputs\*: Order ID and requested action (e.g., update, cancel).   
 - \*Outputs\*: Updated order status in the interface, confirmation message.   
  
- \*\*Payment Management Interface\*\*:   
 - \*Description\*: A part of the admin dashboard for administrators to view, update, or process refunds for payments.   
 - \*Inputs\*: Payment ID and requested action (e.g., update, refund).   
 - \*Outputs\*: Updated payment status in the interface, confirmation message.   
  
- \*\*Administrator Management Interface\*\*:   
 - \*Description\*: A part of the admin dashboard for administrators to manage other admin accounts.   
 - \*Inputs\*: Administrator ID and requested action (e.g., add, edit, delete).   
 - \*Outputs\*: Updated or deleted administrator records, confirmation message.   
  
- \*\*Plugin Management Interface\*\*:   
 - \*Description\*: A part of the admin dashboard for administrators to develop, configure, or manage plugins.   
 - \*Inputs\*: Plugin code, configuration parameters, or plugin ID.   
 - \*Outputs\*: Updated plugin configuration or status, confirmation message.   
  
## 2.2 Hardware Interfaces   
  
The system does not directly interact with specific hardware devices. However, it is expected to support common hardware configurations used by web and mobile clients, such as:   
  
- \*\*Web Browsers\*\*:   
 - \*Description\*: The system must be compatible with major web browsers, including Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.   
 - \*Supported Protocols\*: HTTP/HTTPS, WebSocket for real-time interactions.   
  
- \*\*Mobile Devices\*\*:   
 - \*Description\*: The system must be responsive and functional on various mobile devices, including smartphones and tablets.   
 - \*Supported Protocols\*: HTTP/HTTPS, WebSocket for real-time interactions.   
  
- \*\*Input Devices\*\*:   
 - \*Description\*: The system should support standard input devices such as keyboards, touchscreens, and mice.   
 - \*Supported Protocols\*: Standard input/output (I/O) protocols.   
  
- \*\*Output Devices\*\*:   
 - \*Description\*: The system should be compatible with standard output devices such as monitors and mobile screens.   
 - \*Supported Protocols\*: Standard rendering protocols for web and mobile interfaces.   
  
## 2.3 Software Interfaces   
  
The system interacts with various software components, including databases, external APIs, and third-party tools, to perform its functions.   
  
- \*\*Database Interface (Customer, Product, Order, Inventory, Plugin, Administrator entities)\*\*:   
 - \*Description\*: The system must interact with a relational database to store and retrieve data related to customers, products, orders, inventory, plugins, and administrators.   
 - \*Data Access Methods\*: SQL queries, ORM (Object-Relational Mapping), and database transactions.   
 - \*Inputs\*: Data to be stored, such as customer information, product details, order records, inventory levels, plugin configurations, and admin credentials.   
 - \*Outputs\*: Retrieved data for display or processing, such as customer account details, product information, order status, inventory stock levels, plugin status, and admin account details.   
  
- \*\*Payment Gateway API (Third-Party)\*\*:   
 - \*Description\*: The system must integrate with a third-party payment gateway API to process customer payments during checkout.   
 - \*Supported Protocols\*: RESTful API, HTTPS.   
 - \*Inputs\*: Payment method, amount, customer billing information.   
 - \*Outputs\*: Payment confirmation, transaction status, and response data from the gateway.   
  
- \*\*Email Service API (Third-Party)\*\*:   
 - \*Description\*: The system must use an email service API to send confirmation messages and order confirmation emails to customers.   
 - \*Supported Protocols\*: RESTful API, SMTP.   
 - \*Inputs\*: Recipient email address, message content, subject line.   
 - \*Outputs\*: Email delivery confirmation, log entry indicating successful email sending.   
  
- \*\*Search Engine API (Optional Third-Party)\*\*:   
 - \*Description\*: If implemented, the system may integrate with a search engine API to enhance product search functionality.   
 - \*Supported Protocols\*: RESTful API, HTTPS.   
 - \*Inputs\*: Search term, category, or keyword.   
 - \*Outputs\*: List of products matching the search criteria.   
  
- \*\*Plugin Interface\*\*:   
 - \*Description\*: The system must support a plugin architecture where administrators can develop or update plugins.   
 - \*Supported Protocols\*: RESTful API or internal module integration.   
 - \*Inputs\*: Plugin code, configuration files.   
 - \*Outputs\*: Registered or updated plugin, confirmation message.   
  
- \*\*Session Management System\*\*:   
 - \*Description\*: The system must manage user sessions for both customers and administrators, ensuring secure and persistent session handling.   
 - \*Data Access Methods\*: Session tokens, cookie-based authentication.   
 - \*Inputs\*: Session identifier or login credentials.   
 - \*Outputs\*: Session initiation or termination status, user redirection.   
  
## 2.4 Communication Interfaces   
  
The system must communicate with external services and users via network-based interfaces. These include:   
  
- \*\*Email Communication Interface\*\*:   
 - \*Description\*: The system must send confirmation and order confirmation emails to customers using an email service.   
 - \*Supported Protocols\*: SMTP, RESTful API.   
 - \*Inputs\*: Customer email address, order details, message template.   
 - \*Outputs\*: Email sent to the customer, log entry indicating email delivery.   
  
- \*\*Web Communication Interface (HTTP/HTTPS)\*\*:   
 - \*Description\*: The system must support HTTP/HTTPS protocols to allow customers and administrators to access the system through web browsers.   
 - \*Supported Protocols\*: HTTP/1.1, HTTP/2, HTTPS.   
 - \*Inputs\*: HTTP requests from clients (e.g., login, search, checkout).   
 - \*Outputs\*: HTTP responses, including HTML pages, JSON data, and redirections.   
  
- \*\*WebSocket Communication Interface (Real-Time Updates)\*\*:   
 - \*Description\*: The system may use WebSocket for real-time communication, such as inventory updates or order status changes.   
 - \*Supported Protocols\*: WebSocket over HTTPS.   
 - \*Inputs\*: Real-time event triggers (e.g., stock change, order update).   
 - \*Outputs\*: Real-time notifications or updates to the client.   
  
- \*\*API Communication Interface (Plugin and Admin Modules)\*\*:   
 - \*Description\*: The system must provide internal or external APIs for plugin and admin module interactions.   
 - \*Supported Protocols\*: RESTful API, JSON format.   
 - \*Inputs\*: Plugin code, configuration parameters, admin commands.   
 - \*Outputs\*: Plugin registration status, admin operation confirmation.   
  
By defining these external interfaces, the system ensures that all necessary interactions with users, hardware, software, and communication channels are clearly understood and can be implemented effectively.

# Use Case

Use Case Name: Customer Register   
Use Case ID: UC-01   
Actors: Customer, Administrator   
Preconditions:   
1. The system is operational and accessible.   
2. The customer does not have an existing account in the system.   
3. The customer has access to the registration page.   
  
Postconditions:   
1. A new customer account is created and stored in the system.   
2. The customer receives a confirmation message or email.   
3. The customer is redirected to the login page or shopping cart.   
  
Main Flow:   
1. The customer navigates to the registration page.   
2. The customer enters personal information, including name, email address, password, and contact details.   
3. The customer confirms the entered information and submits the registration form.   
4. The system validates the input data (e.g., checks for valid email format, password strength, and uniqueness of the email).   
5. The system creates a new Customer entity and stores it in the database.   
6. The system sends a confirmation message or email to the customer's provided email address.   
7. The customer is redirected to the login page or shopping cart.   
  
Alternative Flow:   
1. The system detects invalid or incomplete input data.   
 a. The system displays an error message indicating the issue.   
 b. The customer corrects the input and resubmits the form.   
2. The customer enters an email address that is already registered.   
 a. The system displays an error message indicating that the email is already in use.   
 b. The customer chooses to log in or provides a new email address and resubmits the form.   
3. The system fails to send a confirmation email.   
 a. The system logs the error and displays a message to the customer.   
 b. The customer can manually check their inbox or contact support for assistance.   
4. The customer cancels the registration process.   
 a. The system does not create a new account.   
 b. The customer is redirected to the homepage or login page.  
  
Use Case Name: Customer Login   
Use Case ID: UC-02   
Actors: Customer   
Preconditions:   
1. The system is operational and accessible.   
2. The customer has already registered an account in the system.   
3. The customer is on the login page or has access to it.   
  
Postconditions:   
1. The customer is authenticated and logged into the system.   
2. The customer is redirected to the homepage or shopping cart.   
3. The session is initiated and the customer's information is available for the system to use.   
  
Main Flow:   
1. The customer navigates to the login page.   
2. The customer enters their registered email address and password.   
3. The customer clicks the "Login" button.   
4. The system validates the email and password against the stored Customer entity in the database.   
5. The system confirms the customer's identity.   
6. The system initiates a session for the customer and stores relevant session data.   
7. The customer is redirected to the homepage or shopping cart.   
  
Alternative Flow:   
1. The customer enters an incorrect email or password.   
 a. The system displays an error message indicating the login failed.   
 b. The customer may attempt to log in again with correct information.   
2. The customer account is locked due to multiple failed attempts.   
 a. The system displays a message indicating the account is temporarily locked.   
 b. The customer must wait or contact support to unlock the account.   
3. The customer forgets their password.   
 a. The customer clicks the "Forgot Password" link.   
 b. The system prompts the customer to enter their registered email address.   
 c. The system sends a password reset link to the customer's email.   
 d. The customer follows the link to reset their password.   
4. The customer cancels the login process.   
 a. The system terminates the login attempt.   
 b. The customer is redirected to the homepage or shopping cart.  
  
Use Case Name: Customer Logout   
Use Case ID: UC-03   
Actors: Customer   
Preconditions:   
1. The system is operational and accessible.   
2. The customer is currently logged in.   
3. The customer has access to the logout functionality (e.g., a "Logout" button or link).   
  
Postconditions:   
1. The customer's session is terminated.   
2. The customer is redirected to the homepage or login page.   
3. The system no longer displays personalized content for the customer.   
  
Main Flow:   
1. The customer accesses the logout functionality.   
2. The customer clicks on the "Logout" button or selects the "Logout" option.   
3. The system verifies the customer's session and initiates the logout process.   
4. The system terminates the session and clears session-related data.   
5. The system redirects the customer to the homepage or login page.   
6. The system confirms the logout was successful (e.g., via a message or status indicator).   
  
Alternative Flow:   
1. The system fails to terminate the session.   
 a. The system logs the error and displays a message to the customer.   
 b. The customer may manually close the browser or try again later.   
2. The customer navigates away before the logout is complete.   
 a. The system does not complete the session termination.   
 b. The customer may be prompted to confirm logout or encounter an error.   
3. The customer cancels the logout process.   
 a. The system does not terminate the session.   
 b. The customer remains logged in and is redirected back to the previous page.  
  
Use Case Name: View Product Details   
Use Case ID: UC-04   
Actors: Customer, Product, Shopping Cart   
Preconditions:   
1. The system is operational and accessible.   
2. The customer is on the product listing page or has navigated to a specific product page.   
3. The requested product exists in the Product entity.   
  
Postconditions:   
1. The customer has viewed the detailed information of the selected product.   
2. The product details are displayed in a clear and user-friendly format.   
3. The customer may choose to add the product to their Shopping Cart or continue browsing.   
  
Main Flow:   
1. The customer selects a product from the product listing or searches for a specific product.   
2. The system retrieves the corresponding Product entity from the database.   
3. The system displays the product details, including name, description, price, images, and availability.   
4. The customer reviews the information and may decide to add the product to their Shopping Cart.   
5. If the customer adds the product, the system updates the Shopping Cart accordingly.   
6. The customer may return to the product listing or proceed to checkout.   
  
Alternative Flow:   
1. The requested product does not exist.   
 a. The system displays an error message indicating the product is not available.   
 b. The customer may return to the product listing or search for another product.   
2. The product information is incomplete or outdated.   
 a. The system displays the available information with a note indicating some details may be missing.   
 b. The customer may contact support or continue browsing.   
3. The customer chooses not to add the product to the Shopping Cart.   
 a. The system remains on the product details page or redirects the customer to another section.   
4. The customer navigates away without completing the action.   
 a. The system retains the Shopping Cart state if items were added.   
 b. The system discards any temporary selections if no action was taken.  
  
Use Case Name: Search Product   
Use Case ID: UC-05   
Actors: Customer, Product, Shopping Cart   
Preconditions:   
1. The system is operational and accessible.   
2. The customer is logged in or browsing as a guest.   
3. The customer has access to the search functionality (e.g., a search bar or search page).   
  
Postconditions:   
1. The system displays a list of products matching the search criteria.   
2. The customer has the option to view product details or add products to the Shopping Cart.   
3. The Shopping Cart is updated if the customer adds any products.   
  
Main Flow:   
1. The customer accesses the search functionality on the homepage or product listing page.   
2. The customer enters a search term, such as product name, category, or keyword.   
3. The customer submits the search request.   
4. The system processes the search and queries the Product entity for matching items.   
5. The system displays the results in a list, including product names, images, prices, and availability.   
6. The customer selects a product from the search results to view more details or add it to the Shopping Cart.   
7. The system updates the Shopping Cart or navigates to the product details page.   
  
Alternative Flow:   
1. No products match the search criteria.   
 a. The system displays a message indicating no results were found.   
 b. The customer may refine the search term or return to the product listing.   
2. The search term is ambiguous or incomplete.   
 a. The system provides suggestions for related or similar products.   
 b. The customer selects a suggested product or modifies the search term.   
3. The system encounters an error while retrieving search results.   
 a. The system logs the error and displays a message to the customer.   
 b. The customer may retry the search or contact support for assistance.   
4. The customer cancels the search process.   
 a. The system does not display any search results.   
 b. The customer is redirected back to the homepage or product listing page.  
  
Use Case Name: Add Product to Cart   
Use Case ID: UC-06   
Actors: Customer, Product, Shopping Cart, Inventory   
Preconditions:   
1. The system is operational and accessible.   
2. The customer is logged in or browsing as a guest.   
3. The customer has viewed the product details.   
4. The product is available in the Inventory.   
5. The customer has access to the "Add to Cart" functionality.   
  
Postconditions:   
1. The selected product is added to the customer's Shopping Cart.   
2. The quantity of the product in the Shopping Cart is updated if it is already present.   
3. The Inventory reflects the updated product quantity.   
4. The customer receives a confirmation message that the product was added.   
5. The customer may continue shopping or proceed to checkout.   
  
Main Flow:   
1. The customer views the product details.   
2. The customer selects the desired quantity of the product.   
3. The customer clicks the "Add to Cart" button.   
4. The system checks the product's availability in the Inventory.   
5. The system adds the product (with quantity) to the Shopping Cart.   
6. The system updates the Inventory to reflect the reduction in stock.   
7. The system displays a confirmation message that the product was successfully added to the cart.   
8. The customer is redirected to the Shopping Cart page or remains on the product details page.   
  
Alternative Flow:   
1. The selected quantity exceeds the available stock.   
 a. The system displays an error message indicating insufficient stock.   
 b. The customer adjusts the quantity and retries the action.   
2. The product is out of stock.   
 a. The system displays a message indicating the product is unavailable.   
 b. The customer may remove the product from the cart or choose a similar item.   
3. The system encounters an error while updating the Shopping Cart or Inventory.   
 a. The system logs the error and displays a message to the customer.   
 b. The customer may retry the action or contact support for assistance.   
4. The customer cancels the action before the product is added to the cart.   
 a. The system does not modify the Shopping Cart or Inventory.   
 b. The customer remains on the product details page.  
  
Use Case Name: Update Cart   
Use Case ID: UC-07   
Actors: Customer, Shopping Cart, Product, Inventory, Payment   
Preconditions:   
1. The system is operational and accessible.   
2. The customer is logged in or browsing as a guest.   
3. The customer has at least one item in their Shopping Cart.   
4. The product quantities in the Shopping Cart are valid and match available inventory.   
  
Postconditions:   
1. The Shopping Cart is updated with the new quantities or removed items.   
2. The Inventory is updated to reflect the changes in the Shopping Cart.   
3. The customer receives a confirmation message of the update.   
4. The cart total and related information (e.g., tax, shipping) are recalculated and displayed.   
  
Main Flow:   
1. The customer accesses the Shopping Cart page.   
2. The customer selects an item in the cart and updates the quantity or removes the item.   
3. The system verifies the updated quantity against the available Inventory.   
4. The system updates the Shopping Cart with the new quantity or removes the item.   
5. The system adjusts the Inventory accordingly.   
6. The system recalculates the cart total and displays the updated information.   
7. The system provides a confirmation message of the update.   
8. The customer may continue shopping or proceed to checkout.   
  
Alternative Flow:   
1. The updated quantity exceeds the available stock.   
 a. The system displays an error message indicating insufficient stock.   
 b. The customer adjusts the quantity or removes the item.   
2. The customer removes an item from the cart.   
 a. The system updates the Shopping Cart by removing the item.   
 b. The system updates the Inventory to restore the removed item's quantity.   
3. The system encounters an error while updating the Shopping Cart or Inventory.   
 a. The system logs the error and displays a message to the customer.   
 b. The customer may retry the action or contact support for assistance.   
4. The customer cancels the update process.   
 a. The system does not modify the Shopping Cart or Inventory.   
 b. The customer remains on the Shopping Cart page.  
  
Use Case Name: Remove Product from Cart   
Use Case ID: UC-08   
Actors: Customer, Shopping Cart, Product, Inventory   
Preconditions:   
1. The system is operational and accessible.   
2. The customer is logged in or browsing as a guest.   
3. The customer has at least one item in their Shopping Cart.   
4. The customer has access to the Shopping Cart page and the "Remove" functionality.   
  
Postconditions:   
1. The selected product is removed from the Shopping Cart.   
2. The Inventory is updated to reflect the restored quantity of the removed product.   
3. The cart total and related information (e.g., tax, shipping) are recalculated and displayed.   
4. The customer receives a confirmation message that the product was removed.   
  
Main Flow:   
1. The customer navigates to the Shopping Cart page.   
2. The customer selects a product in the cart and clicks the "Remove" button.   
3. The system verifies the product in the cart and its quantity.   
4. The system removes the product from the Shopping Cart.   
5. The system updates the Inventory by increasing the quantity of the removed product.   
6. The system recalculates the cart total and displays the updated information.   
7. The system provides a confirmation message indicating the product was successfully removed.   
8. The customer may continue shopping or proceed to checkout.   
  
Alternative Flow:   
1. The customer clicks the "Remove" button but decides not to proceed.   
 a. The system prompts for confirmation.   
 b. The customer cancels the removal.   
 c. The Shopping Cart remains unchanged.   
2. The system encounters an error while updating the Inventory.   
 a. The system logs the error and displays a message to the customer.   
 b. The customer may retry the action or contact support for assistance.   
3. The customer removes multiple items in a single session.   
 a. The system processes each removal individually.   
 b. The Inventory and cart total are updated accordingly after each removal.   
4. The customer is not logged in and the Shopping Cart is session-based.   
 a. The system removes the product from the temporary cart.   
 b. The Inventory is not updated until the customer proceeds to checkout or logs in.  
  
Use Case Name: View Cart Summary   
Use Case ID: UC-09   
Actors: Customer, Shopping Cart, Product, Inventory   
Preconditions:   
1. The system is operational and accessible.   
2. The customer is logged in or browsing as a guest.   
3. The customer has at least one item in their Shopping Cart.   
4. The customer has access to the cart summary view (e.g., through a navigation menu or a dedicated cart page).   
  
Postconditions:   
1. The customer sees a summary of all items in their Shopping Cart.   
2. The summary includes product names, quantities, prices, and total cost.   
3. The system displays any relevant messages, such as out-of-stock items or low stock alerts.   
4. The customer may choose to proceed with checkout, update the cart, or continue shopping.   
  
Main Flow:   
1. The customer accesses the cart summary view (e.g., by clicking on a cart icon).   
2. The system retrieves the customer's Shopping Cart and associated Product entities.   
3. The system displays the cart summary, including product details and total cost.   
4. The system checks the Inventory for the availability of each product in the cart.   
5. If any product is out of stock or has low stock, the system displays a warning message.   
6. The customer reviews the cart summary and may choose to continue shopping or proceed to checkout.   
  
Alternative Flow:   
1. The Shopping Cart is empty.   
 a. The system displays a message indicating the cart is empty.   
 b. The customer may return to the product listing or homepage to add items.   
2. The system encounters an error retrieving cart data.   
 a. The system logs the error and displays a message to the customer.   
 b. The customer may retry the action or contact support for assistance.   
3. The customer is not logged in and the cart is session-based.   
 a. The system displays the temporary cart summary.   
 b. The customer may log in to merge the temporary cart with their account cart.   
4. The customer cancels the cart summary view.   
 a. The system does not modify the cart or inventory.   
 b. The customer is redirected back to the previous page.  
  
Use Case Name: Checkout   
Use Case ID: UC-10   
Actors: Customer, Shopping Cart, Product, Inventory, Order, Payment, Administrator   
Preconditions:   
1. The system is operational and accessible.   
2. The customer is logged in or has provided guest checkout information.   
3. The customer has at least one item in their Shopping Cart.   
4. The product quantities in the cart are valid and available in the Inventory.   
5. The customer has access to the checkout page.   
  
Postconditions:   
1. A new Order entity is created and stored in the system.   
2. The Payment is processed and confirmed.   
3. The Inventory is updated to reflect the reduced stock for each product in the order.   
4. The Shopping Cart is cleared of all items.   
5. The customer receives a confirmation message or email about the order.   
6. The Administrator is notified of the new order (if applicable).   
  
Main Flow:   
1. The customer navigates to the checkout page.   
2. The system displays the cart summary, including product details, quantities, and total cost.   
3. The customer enters or confirms their shipping and billing information.   
4. The customer selects a preferred payment method (e.g., credit card, PayPal).   
5. The system validates the shipping and billing information.   
6. The system processes the Payment transaction.   
7. The system creates a new Order entity and stores it in the database.   
8. The system updates the Inventory by reducing the stock of each product in the order.   
9. The system clears the Shopping Cart of all items.   
10. The system sends a confirmation message or email to the customer.   
11. The customer is redirected to an order confirmation page.   
  
Alternative Flow:   
1. The customer’s shipping or billing information is invalid.   
 a. The system displays an error message.   
 b. The customer corrects the information and resubmits the checkout.   
2. The selected payment method fails during transaction processing.   
 a. The system displays an error message and prompts the customer to try another method.   
 b. The customer selects a different payment method and retries the checkout.   
3. The system detects that a product in the cart is no longer available in the requested quantity.   
 a. The system displays an error message and highlights the unavailable item.   
 b. The customer removes or updates the quantity of the item and retries the checkout.   
4. The customer cancels the checkout process.   
 a. The system does not create an Order or modify the Inventory.   
 b. The customer remains on the checkout page or is redirected back to the cart.   
5. The system encounters an error while creating the Order.   
 a. The system logs the error and displays a message to the customer.   
 b. The customer may retry the checkout or contact support for assistance.  
  
Use Case Name: Confirm Order   
Use Case ID: UC-11   
Actors: Customer, Order, Payment, Administrator   
Preconditions:   
1. The system is operational and accessible.   
2. The customer has completed the checkout process and submitted the order.   
3. The Payment information is valid and ready for processing.   
4. The customer has access to the order confirmation page.   
  
Postconditions:   
1. The order is confirmed and stored as an Order entity in the system.   
2. The Payment is processed and confirmed.   
3. The customer receives an order confirmation message or email.   
4. The Administrator is notified of the new order (if applicable).   
5. The Shopping Cart is cleared of all items.   
  
Main Flow:   
1. The customer submits the order after completing the checkout.   
2. The system validates the order details, including shipping, billing, and product quantities.   
3. The system processes the Payment transaction using the selected method.   
4. The system confirms the Payment and updates the order status to "Confirmed."   
5. The system creates and stores the Order entity with all relevant details in the database.   
6. The system clears the Shopping Cart of all items.   
7. The system sends a confirmation message or email to the customer.   
8. The customer is redirected to an order confirmation page.   
  
Alternative Flow:   
1. The Payment processing fails.   
 a. The system displays an error message and prompts the customer to try another payment method.   
 b. The customer selects a different method and retries the confirmation.   
2. The system detects a discrepancy in the order (e.g., product no longer in stock).   
 a. The system displays an error message and highlights the issue.   
 b. The customer adjusts the order and resubmits.   
3. The customer cancels the order confirmation.   
 a. The system does not process the Payment or update the order status.   
 b. The customer is redirected back to the checkout page.   
4. The system encounters an error while confirming the order.   
 a. The system logs the error and displays a message to the customer.   
 b. The customer may retry the confirmation or contact support for assistance.  
  
Use Case Name: Receive Order Confirmation Email   
Use Case ID: UC-12   
Actors: Customer, Order, Payment, Administrator   
Preconditions:   
1. The system is operational and accessible.   
2. The customer has successfully confirmed their order.   
3. The Payment has been processed and confirmed.   
4. The system is configured to send confirmation emails.   
  
Postconditions:   
1. The customer receives an email confirming the order details.   
2. The email contains relevant information such as order number, items purchased, total cost, and estimated delivery time.   
3. The Administrator may be notified of the new order via email or system alert.   
  
Main Flow:   
1. The system successfully processes the customer’s order and Payment.   
2. The system generates an order confirmation email with the relevant details.   
3. The system sends the confirmation email to the customer’s registered email address.   
4. The customer receives the email and reviews the order summary.   
5. The system logs the confirmation email as sent in the Order entity.   
  
Alternative Flow:   
1. The system fails to send the confirmation email.   
 a. The system logs the error and displays a message to the customer.   
 b. The customer can manually check their inbox or contact support for assistance.   
2. The customer’s email address is invalid or unreachable.   
 a. The system logs the error and displays a message to the customer.   
 b. The customer is prompted to update their email address or contact support.   
3. The customer does not receive the confirmation email within a reasonable time.   
 a. The system provides a "Resend Confirmation" option.   
 b. The customer clicks the option, and the system attempts to resend the email.   
4. The system sends the confirmation email but the customer does not open it.   
 a. The system assumes the email was delivered and does not take further action.   
 b. The customer may later access the order confirmation page directly through the system.  
  
Use Case Name: Administrator Login   
Use Case ID: UC-13   
Actors: Administrator   
Preconditions:   
1. The system is operational and accessible.   
2. The administrator has a valid account in the system.   
3. The administrator has access to the administrator login page.   
  
Postconditions:   
1. The administrator is authenticated and logged into the system.   
2. The administrator is redirected to the admin dashboard.   
3. The session is initiated and the administrator's information is available for the system to use.   
  
Main Flow:   
1. The administrator navigates to the admin login page.   
2. The administrator enters their username and password.   
3. The administrator clicks the "Login" button.   
4. The system validates the credentials against the stored Administrator entity in the database.   
5. The system confirms the administrator's identity.   
6. The system initiates a session for the administrator and stores relevant session data.   
7. The administrator is redirected to the admin dashboard.   
  
Alternative Flow:   
1. The administrator enters an incorrect username or password.   
 a. The system displays an error message indicating the login failed.   
 b. The administrator may attempt to log in again with correct information.   
2. The administrator account is locked due to multiple failed attempts.   
 a. The system displays a message indicating the account is temporarily locked.   
 b. The administrator must wait or contact support to unlock the account.   
3. The administrator forgets their password.   
 a. The administrator clicks the "Forgot Password" link.   
 b. The system prompts the administrator to enter their registered email address.   
 c. The system sends a password reset link to the administrator's email.   
 d. The administrator follows the link to reset their password.   
4. The administrator cancels the login process.   
 a. The system terminates the login attempt.   
 b. The administrator is redirected to the homepage or login page.  
  
Use Case Name: Administrator Logout   
Use Case ID: UC-14   
Actors: Administrator   
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is currently logged in.   
3. The administrator has access to the logout functionality (e.g., a "Logout" button or link).   
  
Postconditions:   
1. The administrator's session is terminated.   
2. The administrator is redirected to the homepage or login page.   
3. The system no longer displays administrative content or functions.   
  
Main Flow:   
1. The administrator accesses the logout functionality.   
2. The administrator clicks on the "Logout" button or selects the "Logout" option.   
3. The system verifies the administrator's session and initiates the logout process.   
4. The system terminates the session and clears session-related data.   
5. The system redirects the administrator to the homepage or login page.   
6. The system confirms the logout was successful (e.g., via a message or status indicator).   
  
Alternative Flow:   
1. The system fails to terminate the session.   
 a. The system logs the error and displays a message to the administrator.   
 b. The administrator may manually close the browser or try again later.   
2. The administrator navigates away before the logout is complete.   
 a. The system does not complete the session termination.   
 b. The administrator may be prompted to confirm logout or encounter an error.   
3. The administrator cancels the logout process.   
 a. The system does not terminate the session.   
 b. The administrator remains logged in and is redirected back to the previous page.  
  
Use Case Name: Manage Inventory   
Use Case ID: UC-15   
Actors: Administrator, Inventory, Product   
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged into the system.   
3. The administrator has access to the inventory management interface.   
4. The Inventory and Product entities are populated with relevant data.   
  
Postconditions:   
1. The Inventory is updated with new stock levels or product information.   
2. The Product entity is updated if necessary (e.g., price, description, availability).   
3. The administrator receives a confirmation message of the inventory update.   
4. The system reflects the changes in inventory levels for customers and other stakeholders.   
  
Main Flow:   
1. The administrator navigates to the inventory management interface.   
2. The system displays a list of products with their current stock levels and status.   
3. The administrator selects a product to update.   
4. The administrator modifies the product's stock level or other inventory-related details (e.g., reorder threshold, product status).   
5. The administrator submits the changes.   
6. The system validates the input and updates the Inventory and Product entities in the database.   
7. The system provides a confirmation message that the inventory was successfully updated.   
8. The administrator may continue managing inventory or return to the admin dashboard.   
  
Alternative Flow:   
1. The administrator enters an invalid or negative stock level.   
 a. The system displays an error message indicating the invalid input.   
 b. The administrator corrects the input and resubmits the changes.   
2. The system fails to update the Inventory.   
 a. The system logs the error and displays a message to the administrator.   
 b. The administrator may retry the action or contact support for assistance.   
3. The administrator cancels the inventory update before submission.   
 a. The system does not modify the Inventory or Product entities.   
 b. The administrator is redirected back to the inventory management interface.   
4. The selected product does not exist in the Inventory.   
 a. The system displays an error message indicating the product is not found.   
 b. The administrator selects a valid product or adds a new one.  
  
Use Case Name: Update Product Information   
Use Case ID: UC-16   
Actors: Administrator, Product, Inventory   
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged into the system.   
3. The administrator has access to the product management interface.   
4. The product to be updated exists in the Product entity.   
  
Postconditions:   
1. The product's information is updated in the Product entity.   
2. The Inventory is updated if the stock level is modified.   
3. The changes are reflected on the product listing and product detail pages.   
4. The administrator receives a confirmation message that the product information was successfully updated.   
  
Main Flow:   
1. The administrator navigates to the product management interface.   
2. The system displays a list of products with their current details.   
3. The administrator selects a specific product to update.   
4. The administrator modifies relevant product information, such as price, description, stock level, or availability.   
5. The administrator submits the updated information.   
6. The system validates the input (e.g., checks for valid price, non-negative stock level).   
7. The system updates the Product entity and adjusts the Inventory if the stock level is changed.   
8. The system provides a confirmation message indicating the product was successfully updated.   
9. The administrator may view the updated product details or continue managing other products.   
  
Alternative Flow:   
1. The administrator enters invalid data (e.g., negative price or non-numeric stock level).   
 a. The system displays an error message indicating the invalid input.   
 b. The administrator corrects the data and resubmits the changes.   
2. The system fails to update the Product entity.   
 a. The system logs the error and displays a message to the administrator.   
 b. The administrator may retry the update or contact support for assistance.   
3. The administrator cancels the update process before submission.   
 a. The system does not modify the Product or Inventory entities.   
 b. The administrator is redirected back to the product management interface.   
4. The selected product does not exist in the system.   
 a. The system displays an error message indicating the product is not found.   
 b. The administrator selects a valid product or adds a new product.  
  
Use Case Name: Categorize Product   
Use Case ID: UC-17   
Actors: Administrator, Product, Inventory, Plugin   
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged into the system.   
3. The administrator has access to the product categorization interface.   
4. The product to be categorized exists in the Product entity.   
  
Postconditions:   
1. The product is assigned to one or more categories.   
2. The categorization is stored in the Product entity.   
3. The Inventory reflects the updated categorization for accurate stock management.   
4. The administrator receives a confirmation message that the product has been successfully categorized.   
5. The Plugin may use the categorization for additional functionalities (e.g., search filters or recommendations).   
  
Main Flow:   
1. The administrator navigates to the product categorization interface.   
2. The system displays a list of products with their current categorization status.   
3. The administrator selects a product to categorize.   
4. The administrator assigns the product to one or more categories (e.g., electronics, clothing, home goods).   
5. The administrator submits the categorization.   
6. The system validates the selected categories and updates the Product entity.   
7. The system updates the Inventory to reflect the categorization for accurate stock tracking.   
8. The system provides a confirmation message that the categorization was successful.   
9. The administrator may continue categorizing other products or return to the admin dashboard.   
  
Alternative Flow:   
1. The administrator selects an invalid or non-existent category.   
 a. The system displays an error message indicating the category is invalid.   
 b. The administrator selects a valid category and resubmits the categorization.   
2. The system fails to update the Product entity with the new categorization.   
 a. The system logs the error and displays a message to the administrator.   
 b. The administrator may retry the action or contact support for assistance.   
3. The administrator cancels the categorization process before submission.   
 a. The system does not modify the Product or Inventory entities.   
 b. The administrator is redirected back to the product categorization interface.   
4. The Plugin encounters an issue processing the new categorization.   
 a. The system logs the error and notifies the administrator.   
 b. The administrator may review the categorization or contact the plugin developer for support.  
  
Use Case Name: Develop Plugin   
Use Case ID: UC-18   
Actors: Administrator, Plugin   
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged into the system.   
3. The administrator has access to the plugin development or configuration interface.   
4. The Plugin entity is initialized or ready for deployment.   
  
Postconditions:   
1. A new or updated Plugin is developed and registered in the system.   
2. The Plugin is functional and integrated with the system's features (e.g., inventory, product, customer, or payment modules).   
3. The administrator receives a confirmation message that the plugin was successfully developed and deployed.   
4. The Plugin is available for use by the system or can be activated for testing or production.   
  
Main Flow:   
1. The administrator navigates to the plugin development or configuration interface.   
2. The administrator selects the "Develop New Plugin" or "Update Existing Plugin" option.   
3. The administrator defines the plugin's functionality, such as custom search filters, payment gateway integration, or inventory tracking.   
4. The administrator writes or uploads the plugin code and configuration files.   
5. The system validates the plugin's syntax and compatibility with the existing system architecture.   
6. The system registers the Plugin entity in the database and links it to relevant modules (e.g., Inventory, Product, or Customer).   
7. The system provides a confirmation message indicating the plugin was successfully developed and deployed.   
8. The administrator may activate the plugin or return to the admin dashboard.   
  
Alternative Flow:   
1. The plugin code contains errors or is incompatible with the system.   
 a. The system displays an error message indicating the issue.   
 b. The administrator corrects the code and resubmits the plugin.   
2. The system fails to register the Plugin entity.   
 a. The system logs the error and displays a message to the administrator.   
 b. The administrator may retry the action or contact support for assistance.   
3. The administrator cancels the plugin development process.   
 a. The system does not register or deploy the plugin.   
 b. The administrator is redirected back to the plugin development or admin dashboard.   
4. The plugin conflicts with an existing system function or another plugin.   
 a. The system displays a warning message about the conflict.   
 b. The administrator reviews the conflict and modifies the plugin accordingly.  
  
Use Case Name: Configure Plugin   
Use Case ID: UC-19   
Actors: Administrator, Plugin   
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged into the system.   
3. The administrator has access to the plugin configuration interface.   
4. The Plugin entity is registered in the system and ready for configuration.   
  
Postconditions:   
1. The Plugin is configured with the desired settings or parameters.   
2. The configuration is stored in the Plugin entity.   
3. The Plugin is either activated or ready for activation based on the configuration.   
4. The administrator receives a confirmation message that the configuration was successful.   
  
Main Flow:   
1. The administrator navigates to the plugin configuration interface.   
2. The system displays a list of available plugins along with their current status and configuration options.   
3. The administrator selects a specific Plugin to configure.   
4. The administrator modifies the plugin's configuration parameters (e.g., API keys, integration settings, or behavior rules).   
5. The administrator submits the configuration.   
6. The system validates the configuration inputs and updates the Plugin entity in the database.   
7. The system provides a confirmation message indicating the configuration was successfully applied.   
8. The administrator may activate the plugin or return to the admin dashboard.   
  
Alternative Flow:   
1. The administrator enters invalid or incomplete configuration data.   
 a. The system displays an error message indicating the issue.   
 b. The administrator corrects the data and resubmits the configuration.   
2. The system fails to update the Plugin entity with the new configuration.   
 a. The system logs the error and displays a message to the administrator.   
 b. The administrator may retry the configuration or contact support for assistance.   
3. The administrator cancels the configuration process before submission.   
 a. The system does not modify the Plugin entity.   
 b. The administrator is redirected back to the plugin configuration interface.   
4. The Plugin is not compatible with the current system version or other plugins.   
 a. The system displays a warning message about the incompatibility.   
 b. The administrator reviews the warning and modifies the configuration or disables the plugin.  
  
Use Case Name: Manage Customer   
Use Case ID: UC-20   
Actors: Administrator, Customer   
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged into the system.   
3. The administrator has access to the customer management interface.   
4. The Customer entity is populated with relevant data.   
  
Postconditions:   
1. The administrator can view, update, or delete customer account information.   
2. The Customer entity is updated in the database if changes are made.   
3. The system reflects the updated customer information as needed.   
4. The administrator receives a confirmation message of the action performed.   
  
Main Flow:   
1. The administrator navigates to the customer management interface.   
2. The system displays a list of registered customers with basic information (e.g., name, email, account status).   
3. The administrator selects a specific customer to manage.   
4. The system retrieves the customer's details from the Customer entity.   
5. The administrator chooses to view, edit, or delete the customer's account.   
6. If editing, the administrator updates relevant customer information (e.g., contact details, account status).   
7. The administrator submits the changes or confirms the deletion.   
8. The system validates the input (if editing) or confirms the deletion action.   
9. The system updates the Customer entity in the database with the new information or removes the record if deleted.   
10. The system provides a confirmation message that the action was successfully completed.   
11. The administrator may continue managing other customers or return to the admin dashboard.   
  
Alternative Flow:   
1. The administrator attempts to edit a customer with invalid data (e.g., invalid email format).   
 a. The system displays an error message indicating the invalid input.   
 b. The administrator corrects the data and resubmits the changes.   
2. The system fails to update or delete the customer record.   
 a. The system logs the error and displays a message to the administrator.   
 b. The administrator may retry the action or contact support for assistance.   
3. The administrator cancels the action before submission.   
 a. The system does not modify the Customer entity.   
 b. The administrator is redirected back to the customer management interface.   
4. The selected customer does not exist in the system.   
 a. The system displays an error message indicating the customer is not found.   
 b. The administrator selects a valid customer or adds a new customer.  
  
Use Case Name: Manage Product   
Use Case ID: UC-21   
Actors: Administrator, Product, Inventory   
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged into the system.   
3. The administrator has access to the product management interface.   
4. The Product and Inventory entities are populated with relevant data.   
  
Postconditions:   
1. The administrator can add, edit, or delete product information.   
2. The Product entity is updated in the database if changes are made.   
3. The Inventory is updated if stock levels or availability are modified.   
4. The system reflects the updated product information for customers and other stakeholders.   
5. The administrator receives a confirmation message for the action performed.   
  
Main Flow:   
1. The administrator navigates to the product management interface.   
2. The system displays a list of existing products with their details (e.g., name, price, stock level, category).   
3. The administrator selects an option: "Add New Product," "Edit Product," or "Delete Product."   
4. If adding a new product, the administrator enters product details such as name, description, price, category, and stock level.   
5. If editing a product, the administrator selects a specific product and modifies the relevant information.   
6. If deleting a product, the administrator selects a specific product and confirms the deletion.   
7. The administrator submits the changes or deletion request.   
8. The system validates the input data (if adding or editing) and confirms the deletion (if deleting).   
9. The system updates the Product entity in the database with the new or modified information or removes the product record if deleted.   
10. If stock levels are modified, the system updates the Inventory entity accordingly.   
11. The system provides a confirmation message indicating the action was successfully completed.   
12. The administrator may continue managing other products or return to the admin dashboard.   
  
Alternative Flow:   
1. The administrator enters invalid or incomplete product data during the add or edit process.   
 a. The system displays an error message indicating the issue.   
 b. The administrator corrects the data and resubmits the changes.   
2. The system fails to update or delete the product record.   
 a. The system logs the error and displays a message to the administrator.   
 b. The administrator may retry the action or contact support for assistance.   
3. The administrator cancels the action before submission.   
 a. The system does not modify the Product or Inventory entities.   
 b. The administrator is redirected back to the product management interface.   
4. The selected product does not exist in the system.   
 a. The system displays an error message indicating the product is not found.   
 b. The administrator selects a valid product or adds a new product.  
  
Use Case Name: Manage Order   
Use Case ID: UC-22   
Actors: Administrator, Order, Customer, Product, Inventory, Payment   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged into the system.   
3. The administrator has access to the order management interface.   
4. The Order entity is populated with relevant data.   
5. The system has access to related entities such as Customer, Product, Inventory, and Payment.   
  
Postconditions:   
1. The administrator can view, update, or cancel order details.   
2. The Order entity is updated in the database if changes are made (e.g., order status, delivery address).   
3. The Inventory and Payment entities are updated accordingly if an order is canceled or modified.   
4. The system reflects the updated order status for customers and administrative functions.   
5. The administrator receives a confirmation message of the action performed.   
  
Main Flow:   
1. The administrator navigates to the order management interface.   
2. The system displays a list of orders with basic information (e.g., order ID, customer name, order status, total cost).   
3. The administrator selects a specific order to manage.   
4. The system retrieves the order details from the Order entity, including customer information, product list, payment status, and shipping details.   
5. The administrator chooses to view, update, or cancel the order.   
6. If updating, the administrator modifies relevant information (e.g., order status to "Shipped" or "Processing," or updates the delivery address).   
7. If canceling, the administrator confirms the cancellation of the order.   
8. The administrator submits the changes or cancellation request.   
9. The system validates the input (if updating) and confirms the cancellation (if canceling).   
10. If the order is canceled, the system updates the Inventory to restore the product quantities and adjusts the Payment entity if necessary.   
11. The system updates the Order entity in the database with the new information or marks it as canceled.   
12. The system sends a notification (if applicable) to the customer regarding the changes.   
13. The system provides a confirmation message that the action was successfully completed.   
14. The administrator may continue managing other orders or return to the admin dashboard.   
  
Alternative Flow:   
1. The administrator attempts to update an order with invalid data (e.g., incorrect delivery address format).   
 a. The system displays an error message indicating the invalid input.   
 b. The administrator corrects the data and resubmits the changes.   
2. The system fails to update or cancel the order.   
 a. The system logs the error and displays a message to the administrator.   
 b. The administrator may retry the action or contact support for assistance.   
3. The administrator cancels the action before submission.   
 a. The system does not modify the Order, Inventory, or Payment entities.   
 b. The administrator is redirected back to the order management interface.   
4. The selected order does not exist in the system.   
 a. The system displays an error message indicating the order is not found.   
 b. The administrator selects a valid order or creates a new one.   
5. The administrator attempts to cancel an order that has already been shipped.   
 a. The system displays a message indicating the order cannot be canceled.   
 b. The administrator may update the order status to "Completed" or take other appropriate actions.  
  
Use Case Name: Manage Inventory   
Use Case ID: UC-15   
Actors: Administrator, Inventory, Product   
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged into the system.   
3. The administrator has access to the inventory management interface.   
4. The Inventory and Product entities are populated with relevant data.   
  
Postconditions:   
1. The Inventory is updated with new stock levels or product information.   
2. The Product entity is updated if necessary (e.g., price, description, availability).   
3. The administrator receives a confirmation message of the inventory update.   
4. The system reflects the changes in inventory levels for customers and other stakeholders.   
  
Main Flow:   
1. The administrator navigates to the inventory management interface.   
2. The system displays a list of products with their current stock levels and status.   
3. The administrator selects a product to update.   
4. The administrator modifies the product's stock level or other inventory-related details (e.g., reorder threshold, product status).   
5. The administrator submits the changes.   
6. The system validates the input and updates the Inventory and Product entities in the database.   
7. The system provides a confirmation message that the inventory was successfully updated.   
8. The administrator may continue managing inventory or return to the admin dashboard.   
  
Alternative Flow:   
1. The administrator enters an invalid or negative stock level.   
 a. The system displays an error message indicating the invalid input.   
 b. The administrator corrects the input and resubmits the changes.   
2. The system fails to update the Inventory.   
 a. The system logs the error and displays a message to the administrator.   
 b. The administrator may retry the action or contact support for assistance.   
3. The administrator cancels the inventory update before submission.   
 a. The system does not modify the Inventory or Product entities.   
 b. The administrator is redirected back to the inventory management interface.   
4. The selected product does not exist in the Inventory.   
 a. The system displays an error message indicating the product is not found.   
 b. The administrator selects a valid product or adds a new one.  
  
Use Case Name: Manage Payment   
Use Case ID: UC-23   
Actors: Administrator, Payment, Order, Customer   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged into the system.   
3. The administrator has access to the payment management interface.   
4. The Payment and Order entities are populated with relevant data.   
5. The system has access to the Customer entity for reference.   
  
Postconditions:   
1. The administrator can view, update, or refund payment details.   
2. The Payment entity is updated in the database if changes are made (e.g., payment status, refund amount).   
3. The Order entity is updated to reflect the new payment status or refund information.   
4. The system reflects the updated payment status for administrative and customer views.   
5. The administrator receives a confirmation message of the action performed.   
  
Main Flow:   
1. The administrator navigates to the payment management interface.   
2. The system displays a list of payments with basic information (e.g., payment ID, customer name, amount, payment status, and associated order ID).   
3. The administrator selects a specific payment to manage.   
4. The system retrieves the payment details from the Payment entity, including the associated Order information, customer details, and payment method.   
5. The administrator chooses to view, update, or initiate a refund for the payment.   
6. If updating, the administrator modifies relevant information (e.g., payment status to "Completed" or "Failed").   
7. If initiating a refund, the administrator confirms the refund amount and selects the refund reason.   
8. The administrator submits the changes or refund request.   
9. The system validates the input (if updating) and confirms the refund (if initiating).   
10. If a refund is initiated, the system updates the Payment entity to reflect the refund and adjusts the Order status accordingly (e.g., "Refunded" or "Partially Refunded").   
11. The system updates the Payment entity in the database with the new information.   
12. The system provides a confirmation message that the action was successfully completed.   
13. The system may send a notification (if applicable) to the customer regarding the refund.   
14. The administrator may continue managing other payments or return to the admin dashboard.   
  
Alternative Flow:   
1. The administrator attempts to update a payment with invalid data (e.g., negative refund amount or incorrect payment status).   
 a. The system displays an error message indicating the invalid input.   
 b. The administrator corrects the data and resubmits the changes.   
2. The system fails to update or process the refund.   
 a. The system logs the error and displays a message to the administrator.   
 b. The administrator may retry the action or contact support for assistance.   
3. The administrator cancels the action before submission.   
 a. The system does not modify the Payment or Order entities.   
 b. The administrator is redirected back to the payment management interface.   
4. The selected payment does not exist in the system.   
 a. The system displays an error message indicating the payment is not found.   
 b. The administrator selects a valid payment or reviews the payment list.   
5. The administrator initiates a refund for a payment that is not eligible for refund (e.g., already processed, or order has been shipped).   
 a. The system displays a message indicating the refund cannot be processed.   
 b. The administrator may review the refund policy or take other appropriate actions.  
  
Use Case Name: Manage Administrator   
Use Case ID: UC-24   
Actors: Administrator   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged into the system with appropriate permissions.   
3. The administrator has access to the administrator management interface.   
4. The Administrator entity is populated with relevant data.   
  
Postconditions:   
1. The administrator can add, edit, or remove other administrator accounts.   
2. The Administrator entity is updated in the database if changes are made.   
3. The system reflects the updated administrator information for administrative functions.   
4. The administrator receives a confirmation message of the action performed.   
  
Main Flow:   
1. The administrator navigates to the administrator management interface.   
2. The system displays a list of existing administrators with their details (e.g., username, email, role, status).   
3. The administrator selects an option: "Add New Administrator," "Edit Administrator," or "Remove Administrator."   
4. If adding a new administrator, the administrator enters details such as full name, email, username, and password.   
5. If editing an existing administrator, the administrator selects a specific account and modifies relevant information (e.g., role, status, or contact details).   
6. If removing an administrator, the administrator selects a specific account and confirms the deletion.   
7. The administrator submits the changes or deletion request.   
8. The system validates the input data (if adding or editing) and confirms the deletion (if removing).   
9. The system updates the Administrator entity in the database with the new or modified information or removes the record if deleted.   
10. The system provides a confirmation message that the action was successfully completed.   
11. The administrator may continue managing other administrators or return to the admin dashboard.   
  
Alternative Flow:   
1. The administrator enters invalid or incomplete data during the add or edit process.   
 a. The system displays an error message indicating the issue.   
 b. The administrator corrects the data and resubmits the changes.   
2. The system fails to update or delete the administrator account.   
 a. The system logs the error and displays a message to the administrator.   
 b. The administrator may retry the action or contact support for assistance.   
3. The administrator cancels the action before submission.   
 a. The system does not modify the Administrator entity.   
 b. The administrator is redirected back to the administrator management interface.   
4. The selected administrator does not exist in the system.   
 a. The system displays an error message indicating the administrator is not found.   
 b. The administrator selects a valid administrator or adds a new one.   
5. The administrator attempts to delete their own account.   
 a. The system displays a warning message to prevent accidental self-deletion.   
 b. The administrator may cancel the action or select a different account for deletion.  
  
Use Case Name: Manage Plugin   
Use Case ID: UC-25   
Actors: Administrator, Plugin, Product, Inventory, Payment, Order   
  
Preconditions:   
1. The system is operational and accessible.   
2. The administrator is logged into the system with appropriate permissions.   
3. The administrator has access to the plugin management interface.   
4. The Plugin entity is registered in the system and associated with relevant modules.   
  
Postconditions:   
1. The administrator can activate, deactivate, or configure existing plugins.   
2. The Plugin entity is updated in the database to reflect the new status or configuration.   
3. The system adjusts the functionality of the plugin based on the administrator's actions.   
4. The administrator receives a confirmation message of the action performed.   
5. The Plugin’s impact on the system (e.g., inventory, product, payment, or order modules) is reflected accordingly.   
  
Main Flow:   
1. The administrator navigates to the plugin management interface.   
2. The system displays a list of available plugins, including their status (e.g., active, inactive), version, and associated modules.   
3. The administrator selects a specific plugin to manage.   
4. The administrator chooses an action: "Activate Plugin," "Deactivate Plugin," or "Configure Plugin."   
5. If configuring, the administrator modifies plugin settings (e.g., API keys, behavior parameters, or integration preferences).   
6. The administrator submits the changes or action.   
7. The system validates the action and updates the Plugin entity in the database.   
8. If activated or configured, the system applies the changes to the relevant system modules.   
9. The system provides a confirmation message that the action was successfully completed.   
10. The administrator may continue managing other plugins or return to the admin dashboard.   
  
Alternative Flow:   
1. The administrator attempts to activate a plugin that is incompatible with the current system version or other plugins.   
 a. The system displays a warning message about the incompatibility.   
 b. The administrator may review the warning, resolve the conflict, or choose not to activate the plugin.   
2. The system fails to update the Plugin entity with the new status or configuration.   
 a. The system logs the error and displays a message to the administrator.   
 b. The administrator may retry the action or contact support for assistance.   
3. The administrator cancels the action before submission.   
 a. The system does not modify the Plugin entity.   
 b. The administrator is redirected back to the plugin management interface.   
4. The selected plugin does not exist in the system.   
 a. The system displays an error message indicating the plugin is not found.   
 b. The administrator selects a valid plugin or reviews the plugin list.   
5. The administrator attempts to deactivate a plugin that is currently in use or critical to system functionality.   
 a. The system displays a warning message about the potential impact.   
 b. The administrator may choose to proceed after reviewing the implications or cancel the action.