项目文档

# Functional Requirement

# 1. Functional Requirements   
  
## 1.1 Customer Registration Function   
\*\*Function ID:\*\* FR-01   
\*\*Description:\*\* Allows a customer to register an account by providing their personal information. The system validates the input data and sends an email confirmation to activate the account.   
\*\*Input:\*\*   
- Customer name (string)   
- Email address (string, must be valid and unique)   
- Password (string, must meet strength requirements)   
- Optional: Address (string), Phone number (string)   
\*\*Output:\*\*   
- A new customer account in the database with an assigned CustomerID   
- A confirmation email sent to the customer's email address   
- Customer redirected to login page or homepage   
  
---  
  
## 1.2 Customer Login Function   
\*\*Function ID:\*\* FR-02   
\*\*Description:\*\* Enables a customer to log in using their registered email and password. The system authenticates the user and generates a session token to maintain login state.   
\*\*Input:\*\*   
- Customer's email (string)   
- Customer's password (string)   
\*\*Output:\*\*   
- A session token assigned to the customer   
- Customer redirected to dashboard or homepage   
- Login status updated in the system   
  
---  
  
## 1.3 View Product Details Function   
\*\*Function ID:\*\* FR-03   
\*\*Description:\*\* Displays detailed information about a product, including name, description, price, and images. The system logs the customer's view of the product.   
\*\*Input:\*\*   
- ProductID (integer, unique identifier of the product)   
- CustomerID (integer, optional, for tracking views)   
\*\*Output:\*\*   
- Product details displayed on the product page   
- A log entry in the system indicating the customer viewed the product   
- Option to add product to cart   
  
---  
  
## 1.4 Add Product to Cart Function   
\*\*Function ID:\*\* FR-04   
\*\*Description:\*\* Adds a selected product and quantity to the customer's shopping cart. The system checks inventory availability and updates the cart's total items and cost.   
\*\*Input:\*\*   
- ProductID (integer)   
- Quantity (integer, must be greater than 0 and within inventory limits)   
- CustomerID (integer, unique identifier of the customer)   
\*\*Output:\*\*   
- Updated ShoppingCart record in the database with the added item   
- Updated cart display with new total items and cost   
- Confirmation message displayed to the customer   
  
---  
  
## 1.5 Modify Cart Contents Function   
\*\*Function ID:\*\* FR-05   
\*\*Description:\*\* Allows a customer to adjust the quantity of a product in the cart or remove it. The system verifies inventory availability and recalculates the cart total.   
\*\*Input:\*\*   
- CartItemID (integer)   
- New quantity (integer, optional)   
- CustomerID (integer)   
- Action type: "update quantity" or "remove item"   
\*\*Output:\*\*   
- Updated CartItem record in the database   
- Updated ShoppingCart with new total items and cost   
- Confirmation message displayed to the customer   
  
---  
  
## 1.6 Remove Product from Cart Function   
\*\*Function ID:\*\* FR-06   
\*\*Description:\*\* Removes a specific product from the customer's shopping cart. The system recalculates the cart total and confirms the removal.   
\*\*Input:\*\*   
- CartItemID (integer)   
- CustomerID (integer)   
\*\*Output:\*\*   
- Updated ShoppingCart with reduced total items and cost   
- CartItem removed from the database   
- Confirmation message displayed to the customer   
  
---  
  
## 1.7 Checkout Function   
\*\*Function ID:\*\* FR-07   
\*\*Description:\*\* Processes the customer's cart to create an order and handle payment. The system updates inventory, clears the cart, and sends an email confirmation.   
\*\*Input:\*\*   
- CustomerID (integer)   
- Shipping address (string)   
- Billing address (string)   
- Payment method (string, e.g., credit card, PayPal)   
- Cart content (list of CartItems)   
\*\*Output:\*\*   
- A new Order record in the database with OrderID and associated OrderItems   
- Inventory updated with reduced stock for purchased items   
- Payment record created with PaymentID and transaction details   
- Email confirmation sent to customer   
- Shopping cart cleared of the purchased items   
  
---  
  
## 1.8 Confirm Order Function   
\*\*Function ID:\*\* FR-08   
\*\*Description:\*\* Finalizes an order after checkout by confirming the payment and updating the order status. The system ensures inventory and payment are processed correctly.   
\*\*Input:\*\*   
- OrderID (integer)   
- CustomerID (integer)   
- Payment confirmation status (boolean)   
- Inventory status (boolean)   
\*\*Output:\*\*   
- Order status updated to "confirmed" or "completed"   
- Inventory updated with reduced stock   
- Payment status updated to "processed"   
- Email confirmation sent to the customer   
- Customer redirected to order confirmation page   
  
---  
  
## 1.9 View Purchase History Function   
\*\*Function ID:\*\* FR-09   
\*\*Description:\*\* Displays a list of all previous orders made by a customer, along with order details such as items, total cost, and status. The system allows the customer to request an email confirmation for a specific order.   
\*\*Input:\*\*   
- CustomerID (integer)   
- Optional: OrderID (integer, for viewing a specific order)   
\*\*Output:\*\*   
- List of orders displayed in the "Purchase History" section   
- Order details (products, quantities, prices, status) retrieved and displayed   
- Email confirmation sent to the customer (if requested)   
- Log entry indicating access to purchase history   
  
---  
  
## 1.10 Administrator Login Function   
\*\*Function ID:\*\* FR-10   
\*\*Description:\*\* Authenticates an administrator using their username and password, and grants access to the admin dashboard. The system generates a session token and logs the login attempt.   
\*\*Input:\*\*   
- Admin username (string)   
- Admin password (string)   
\*\*Output:\*\*   
- A session token assigned to the administrator   
- Administrator redirected to the admin dashboard   
- Login status updated in the system   
  
---  
  
## 1.11 Manage Inventory Function   
\*\*Function ID:\*\* FR-11   
\*\*Description:\*\* Enables the administrator to add, edit, or remove products from the inventory. The system ensures that all associated data (e.g., category, stock levels) are synchronized and logs the action.   
\*\*Input:\*\*   
- ProductID (integer, optional)   
- Product name (string)   
- Description (string)   
- Price (decimal)   
- QuantityInStock (integer)   
- CategoryID (integer)   
- Action type: "add", "edit", or "remove"   
\*\*Output:\*\*   
- Updated Inventory record in the database   
- Product details updated or removed   
- Plugin triggered (if applicable) to notify stakeholders   
- Confirmation message displayed to the administrator   
  
---  
  
## 1.12 Update Product Information Function   
\*\*Function ID:\*\* FR-12   
\*\*Description:\*\* Allows the administrator to modify product details such as name, description, price, and category. The system validates the input and updates the inventory accordingly.   
\*\*Input:\*\*   
- ProductID (integer)   
- New product name (string, optional)   
- New description (string, optional)   
- New price (decimal, optional)   
- New category ID (integer, optional)   
- New stock level (integer, optional)   
- Action type: "update"   
\*\*Output:\*\*   
- Updated Product record in the database   
- Updated Inventory record if stock level changed   
- Plugin triggered (if applicable) to notify stakeholders   
- Confirmation message displayed to the administrator   
  
---  
  
## 1.13 Manage Categories Function   
\*\*Function ID:\*\* FR-13   
\*\*Description:\*\* Allows the administrator to create, edit, or delete product categories. The system ensures that product records are updated if a category is changed and logs the action.   
\*\*Input:\*\*   
- CategoryID (integer, optional)   
- Category name (string)   
- Description (string, optional)   
- ParentCategoryID (integer, optional)   
- Action type: "add", "edit", or "delete"   
\*\*Output:\*\*   
- Updated Category record in the database   
- Updated Product records if their category is changed   
- Plugin triggered (if applicable) to notify stakeholders   
- Confirmation message displayed to the administrator   
  
---  
  
## 1.14 Plugin Installation Function   
\*\*Function ID:\*\* FR-14   
\*\*Description:\*\* Enables the administrator to install and activate plugins that extend system functionality. The system validates plugin compatibility and updates the configuration.   
\*\*Input:\*\*   
- Plugin name (string)   
- Plugin version (string)   
- Plugin description (string)   
- Plugin file (compatible plugin package)   
- Action type: "install"   
\*\*Output:\*\*   
- Plugin installed and activated   
- Plugin record added to the Plugin table   
- System configuration updated   
- Confirmation message displayed to the administrator   
  
---  
  
## 1.15 Email Confirmation Function   
\*\*Function ID:\*\* FR-15   
\*\*Description:\*\* Sends a confirmation email to the customer for specific actions such as registration, order confirmation, or account verification. The system logs the email sending action and confirms delivery.   
\*\*Input:\*\*   
- CustomerID (integer)   
- OrderID (integer, optional)   
- Action type: "registration", "order confirmation", or "account verification"   
- Email template (string, dynamically generated based on action)   
\*\*Output:\*\*   
- EmailConfirmation record created in the database   
- Confirmation email sent to the customer   
- System logs the email sending attempt   
- Customer redirected to confirmation page or shown success message   
  
---  
  
## 1.16 View Order Status Function   
\*\*Function ID:\*\* FR-16   
\*\*Description:\*\* Displays the current status of a customer’s order (e.g., pending, shipped, delivered). The system retrieves the latest order data and provides options for further actions if applicable.   
\*\*Input:\*\*   
- OrderID (integer)   
- CustomerID (integer)   
\*\*Output:\*\*   
- Order status displayed (e.g., "pending", "shipped", "delivered")   
- Order details (products, quantities, dates) displayed   
- Option to cancel or contact support (if applicable)   
- Log entry for order status access   
  
---  
  
## 1.17 Manage Cart Item Function   
\*\*Function ID:\*\* FR-17   
\*\*Description:\*\* Allows a customer to modify individual items in the shopping cart, including updating quantity, replacing product, or removing the item. The system recalculates the cart total and validates against inventory.   
\*\*Input:\*\*   
- CartItemID (integer)   
- New quantity (integer, optional)   
- New ProductID (integer, optional)   
- CustomerID (integer)   
- Action type: "update", "replace", or "remove"   
\*\*Output:\*\*   
- Updated CartItem record in the database   
- Updated ShoppingCart with new total items and cost   
- Confirmation message displayed to the customer   
- Inventory status updated if replacement product is added   
  
---  
  
## 1.18 Manage Order Item Function   
\*\*Function ID:\*\* FR-18   
\*\*Description:\*\* Enables the administrator to modify individual items in an order, such as changing quantity or replacing a product. The system updates the order and inventory accordingly and may notify the customer.   
\*\*Input:\*\*   
- OrderItemID (integer)   
- New quantity (integer, optional)   
- New ProductID (integer, optional)   
- OrderID (integer)   
- AdministratorID (integer)   
- Action type: "update", "replace", or "remove"   
\*\*Output:\*\*   
- Updated OrderItem record in the database   
- Updated Order with new total cost and items   
- Inventory updated if product stock is affected   
- Email notification sent to customer if order item is modified   
- Log entry of the administrator's action   
  
---  
  
## 1.19 Manage Product Update Record Function   
\*\*Function ID:\*\* FR-19   
\*\*Description:\*\* Allows the administrator to view, edit, or delete records of product updates. The system ensures data integrity and logs any modifications or deletions.   
\*\*Input:\*\*   
- UpdateRecordID (integer)   
- ProductID (integer)   
- AdminID (integer)   
- OldInfo (string or JSON, optional)   
- NewInfo (string or JSON, optional)   
- Action type: "view", "edit", or "delete"   
\*\*Output:\*\*   
- Updated or deleted ProductUpdateRecord in the database   
- Log entry of the administrator's action   
- Plugin triggered (if applicable) to notify stakeholders   
- Confirmation message displayed to the administrator   
  
---  
  
以上是 \*\*GAMMA-J Web Store\*\* 系统中 \*\*功能需求（Functional Requirements）\*\* 的完整描述。每个功能均基于提供的用例和数据模型提取并定义，确保输入输出清晰、可实现，并与系统上下文一致。

# External Description

# 2. External Interfaces  
  
This chapter describes the external interfaces of the \*\*GAMMA-J Web Store\*\* system. These interfaces define the system's interactions with external entities such as users, hardware, software, and communication channels. The interfaces are derived from the functional requirements and are organized into four categories: \*\*User Interfaces\*\*, \*\*Hardware Interfaces\*\*, \*\*Software Interfaces\*\*, and \*\*Communication Interfaces\*\*.  
  
---  
  
## 2.1 User Interface Output  
  
The system interacts with users through web-based interfaces. These interfaces include web pages, buttons, input fields, and modal windows that allow customers and administrators to perform actions and receive visual feedback.  
  
### 2.1.1 Customer Registration Page   
- \*\*Description:\*\* A web page where customers can provide their personal information to register a new account.  
- \*\*Inputs:\*\* Customer name, email address, password, optional address and phone number.  
- \*\*Outputs:\*\*   
 - Success or error messages based on input validation.  
 - Confirmation message and redirection to the login page or homepage.  
 - Email confirmation sent to the user (see 2.4.1).  
  
### 2.1.2 Customer Login Page   
- \*\*Description:\*\* A web page where registered customers can log in using their email and password.  
- \*\*Inputs:\*\* Email and password.  
- \*\*Outputs:\*\*   
 - Success or error messages.  
 - Redirection to the dashboard or homepage upon successful login.  
  
### 2.1.3 Product Detail Page   
- \*\*Description:\*\* A web page that displays detailed product information including name, description, price, and images.  
- \*\*Inputs:\*\* ProductID and optionally CustomerID for tracking.  
- \*\*Outputs:\*\*   
 - Product information displayed to the user.  
 - Log entry in the system (see 2.4.2).  
 - Option to add the product to the cart.  
  
### 2.1.4 Shopping Cart Interface   
- \*\*Description:\*\* A web interface where customers can view and modify their selected items for purchase.  
- \*\*Inputs:\*\* ProductID, Quantity, CustomerID for adding items; CartItemID, new quantity, or action type for modifying items.  
- \*\*Outputs:\*\*   
 - Updated cart display with total items and cost.  
 - Confirmation messages for add, update, or remove actions.  
 - Inventory status updates if applicable.  
  
### 2.1.5 Checkout Page   
- \*\*Description:\*\* A web page where customers can review their cart and enter shipping and billing details to complete the purchase.  
- \*\*Inputs:\*\* CustomerID, shipping address, billing address, payment method, and cart content.  
- \*\*Outputs:\*\*   
 - Order confirmation message.  
 - Redirection to the order confirmation page.  
 - Email confirmation sent to the customer (see 2.4.3).  
  
### 2.1.6 Order Confirmation Page   
- \*\*Description:\*\* A web page displayed to the customer after a successful checkout and payment confirmation.  
- \*\*Inputs:\*\* OrderID and CustomerID.  
- \*\*Outputs:\*\*   
 - Order status confirmation (e.g., confirmed or completed).  
 - Summary of purchased items and total cost.  
 - Redirection to the homepage or purchase history.  
  
### 2.1.7 Purchase History Page   
- \*\*Description:\*\* A web page where customers can view all their previous orders with details such as items, total cost, and status.  
- \*\*Inputs:\*\* CustomerID and optionally OrderID.  
- \*\*Outputs:\*\*   
 - List of orders with their details displayed.  
 - Option to request an email confirmation for a specific order.  
 - Log entry indicating access (see 2.4.4).  
  
### 2.1.8 Admin Login Page   
- \*\*Description:\*\* A web page where administrators can log in using their username and password.  
- \*\*Inputs:\*\* Admin username and password.  
- \*\*Outputs:\*\*   
 - Session token assigned to the admin.  
 - Redirection to the admin dashboard.  
 - Log entry of login attempt (see 2.4.5).  
  
### 2.1.9 Admin Dashboard Interface   
- \*\*Description:\*\* A web interface for administrators to manage inventory, product information, categories, and order items.  
- \*\*Inputs:\*\* ProductID, CategoryID, OrderItemID, etc., along with new information and action types.  
- \*\*Outputs:\*\*   
 - Confirmation messages for admin actions (e.g., product or category updates).  
 - Updated data displayed in real time or after refresh.  
 - Plugin notification if triggered (see 2.4.6).  
  
### 2.1.10 Order Status Page   
- \*\*Description:\*\* A web page where customers can view the current status of a specific order.  
- \*\*Inputs:\*\* OrderID and CustomerID.  
- \*\*Outputs:\*\*   
 - Order status (e.g., pending, shipped, delivered).  
 - Order details displayed.  
 - Option to cancel or contact support if applicable.  
 - Log entry of access (see 2.4.7).  
  
---  
  
## 2.2 Hardware Interface Output  
  
The \*\*GAMMA-J Web Store\*\* system does not require direct interaction with specific hardware devices for its core functionalities. However, the following hardware-related considerations may apply to the system's operation:  
  
### 2.2.1 Web Browsers   
- \*\*Description:\*\* The system must be compatible with standard web browsers such as Chrome, Firefox, Safari, and Edge.  
- \*\*Interaction Method:\*\* HTML5, CSS3, JavaScript, and responsive design principles.  
- \*\*Purpose:\*\* Ensures users can access and interact with the system regardless of the browser they are using.  
  
---  
  
## 2.3 Software Interface Output  
  
The system interacts with various software components, including databases, plugins, and internal modules, to provide functionality and maintain data integrity.  
  
### 2.3.1 Customer Database   
- \*\*Description:\*\* A database storing customer account information such as CustomerID, name, email, password (hashed), address, and phone number.  
- \*\*Interaction Method:\*\* SQL queries or ORM (Object-Relational Mapping) tools.  
- \*\*Purpose:\*\* Provides persistent storage for customer data and supports registration, login, and purchase history functions.  
  
### 2.3.2 Product and Inventory Database   
- \*\*Description:\*\* A database storing product details (ProductID, name, description, price, image URLs) and inventory information (QuantityInStock).  
- \*\*Interaction Method:\*\* SQL queries or ORM tools.  
- \*\*Purpose:\*\* Ensures accurate tracking of product availability and supports product management, cart updates, and order processing.  
  
### 2.3.3 Order and Payment Database   
- \*\*Description:\*\* A database storing order records (OrderID, CustomerID, status, date) and payment records (PaymentID, transaction details, amount).  
- \*\*Interaction Method:\*\* SQL queries or ORM tools.  
- \*\*Purpose:\*\* Manages order lifecycle and payment processing, supporting checkout, order confirmation, and purchase history.  
  
### 2.3.4 Session Management Module   
- \*\*Description:\*\* A module responsible for generating and managing session tokens for both customers and administrators.  
- \*\*Interaction Method:\*\* Internal API calls or direct database access.  
- \*\*Purpose:\*\* Maintains user authentication state and supports secure access to user-specific data.  
  
### 2.3.5 Plugin Framework   
- \*\*Description:\*\* A software framework allowing administrators to install and activate plugins to extend system functionality.  
- \*\*Interaction Method:\*\* Plugin installation via file upload and activation through admin interface.  
- \*\*Purpose:\*\* Enhances system flexibility and supports notifications or additional features.  
  
---  
  
## 2.4 Communication Interface Output  
  
The \*\*GAMMA-J Web Store\*\* system communicates with external systems and services through various communication channels such as email and logging mechanisms.  
  
### 2.4.1 Email Confirmation Service   
- \*\*Description:\*\* An external email service used to send confirmation emails to customers for account registration, order confirmation, and account verification.  
- \*\*Interaction Method:\*\* SMTP or REST API to send emails, with dynamic email templates.  
- \*\*Inputs:\*\*   
 - CustomerID  
 - OrderID (optional)  
 - Action type (e.g., "registration", "order confirmation", "account verification")  
- \*\*Outputs:\*\*   
 - Email confirmation sent to the customer.  
 - EmailConfirmation record added to the database.  
 - System logs the email sending attempt.  
  
### 2.4.2 Inventory Logging Service   
- \*\*Description:\*\* A logging system that records when a product is viewed or when inventory is modified.  
- \*\*Interaction Method:\*\* Internal logging framework or external logging service.  
- \*\*Inputs:\*\*   
 - ProductID  
 - CustomerID (optional)  
 - Action type (e.g., "view", "add", "remove")  
- \*\*Outputs:\*\*   
 - Log entry created with timestamp and action details.  
  
### 2.4.3 Order Status Notification System   
- \*\*Description:\*\* A communication module that provides updates on order status to customers and logs these interactions.  
- \*\*Interaction Method:\*\* Internal system calls and email notifications.  
- \*\*Inputs:\*\*   
 - OrderID  
 - CustomerID  
- \*\*Outputs:\*\*   
 - Order status displayed to the customer.  
 - Email notification sent if the order is modified.  
 - Log entry of access or modification.  
  
### 2.4.4 Admin Action Logging System   
- \*\*Description:\*\* A system that logs all administrator actions, such as inventory management, product updates, and plugin installation.  
- \*\*Interaction Method:\*\* Internal logging system or external audit logging service.  
- \*\*Inputs:\*\*   
 - AdminID  
 - Action type (e.g., "add product", "delete category", "install plugin")  
- \*\*Outputs:\*\*   
 - Log entry created with timestamp, admin ID, and action details.  
  
### 2.4.5 Plugin Notification System   
- \*\*Description:\*\* A communication system that notifies stakeholders or external services when a plugin is installed or activated.  
- \*\*Interaction Method:\*\* REST API or message queue (e.g., RabbitMQ).  
- \*\*Inputs:\*\*   
 - Plugin name  
 - Plugin version  
 - Plugin description  
- \*\*Outputs:\*\*   
 - Plugin status updated in the system.  
 - Notification sent to stakeholders.  
 - Confirmation message displayed to the admin.  
  
---  
  
## Summary of External Interfaces  
  
| Interface Name | Type | Description |  
|------------------------------------|---------------------|-----------------------------------------------------------------------------|  
| Customer Registration Page | User Interface | Web page for user registration with input validation and email confirmation |  
| Customer Login Page | User Interface | Web page for logging in with authentication and session token generation |  
| Product Detail Page | User Interface | Displays product details and logs user views |  
| Shopping Cart Interface | User Interface | Allows users to add, update, or remove items from the cart |  
| Checkout Page | User Interface | Process order and payment with confirmation and email notification |  
| Order Confirmation Page | User Interface | Displays final order confirmation details |  
| Purchase History Page | User Interface | Displays a list of past orders and allows email reconfirmation |  
| Admin Login Page | User Interface | Secure login for administrators |  
| Admin Dashboard Interface | User Interface | Allows admins to manage products, categories, and orders |  
| Order Status Page | User Interface | Displays current status and details of a specific order |  
| Web Browsers | Hardware Interface | Compatibility with standard web browsers for user access |  
| Customer Database | Software Interface | Stores and retrieves customer account data |  
| Product and Inventory Database | Software Interface | Stores and updates product and stock information |  
| Order and Payment Database | Software Interface | Stores and updates order and payment records |  
| Session Management Module | Software Interface | Manages session tokens for user authentication |  
| Plugin Framework | Software Interface | Supports plugin installation and activation |  
| Email Confirmation Service | Communication Interface | Sends confirmation emails to customers |  
| Inventory Logging Service | Communication Interface | Logs product view and inventory modification events |  
| Order Status Notification System | Communication Interface | Sends order status updates to customers |  
| Admin Action Logging System | Communication Interface | Logs admin actions for audit and traceability |  
| Plugin Notification System | Communication Interface | Notifies stakeholders of plugin-related changes |  
  
---  
  
This section ensures that all external interfaces required for the \*\*GAMMA-J Web Store\*\* system are clearly defined and consistent with the functional requirements. Developers can use this section as a reference for implementing and integrating the system with external components.

# Use Case

Use Case Name: Customer Registration   
Use Case ID: UC-01   
Actors: Customer, Administrator   
Preconditions:   
1. The customer has access to the system interface.   
2. The system is operational and allows new registrations.   
3. The administrator has configured the registration system and verified the required fields.   
  
Postconditions:   
1. A new customer account is created in the system.   
2. The customer receives an email confirmation.   
3. The customer is added to the customer database.   
  
Main Flow:   
1. The customer accesses the registration page.   
2. The system displays the registration form with required fields (e.g., name, email, password).   
3. The customer fills in the form with valid information.   
4. The customer submits the form.   
5. The system validates the customer's input (e.g., checks for valid email format, password strength).   
6. The system creates a new customer record and assigns a unique customer ID.   
7. The system sends an email confirmation to the customer's provided email address.   
8. The customer confirms the email by clicking the link provided in the confirmation email.   
9. The system marks the customer's account as active.   
10. The customer is redirected to the login page or homepage.   
  
Alternative Flow:   
1. If the customer enters an invalid email address or a duplicate email, the system displays an error message and prompts the customer to re-enter a valid email.   
2. If the customer enters a weak password, the system provides feedback on password requirements and asks the customer to choose a stronger password.   
3. If the system fails to send the email confirmation, the system logs the error and displays a message to the customer. The administrator can manually review and resend the confirmation.   
4. If the customer does not confirm the email within a specified time period, the system automatically deactivates the account and may prompt the customer to re-register.  
  
Use Case Name: Customer Login   
Use Case ID: UC-02   
Actors: Customer   
Preconditions:   
1. The customer has a valid and active account.   
2. The system is operational and accessible.   
3. The customer has navigated to the login page.   
  
Postconditions:   
1. The customer is successfully authenticated and logged into the system.   
2. The system updates the customer's login status.   
3. The customer is redirected to their dashboard or homepage.   
  
Main Flow:   
1. The customer enters their registered email and password into the login form.   
2. The customer submits the form.   
3. The system validates the email and password against the customer database.   
4. The system confirms the account is active.   
5. The system authenticates the customer and generates a session token.   
6. The system redirects the customer to their dashboard or homepage.   
  
Alternative Flow:   
1. If the email or password is invalid, the system displays an error message and prompts the customer to re-enter the correct details.   
2. If the customer account is inactive or not confirmed, the system displays a message and may redirect the customer to an account verification page.   
3. If the system detects multiple failed login attempts, it may temporarily lock the account and notify the administrator.   
4. If the system fails to generate a session token, it logs the error and displays a message to the customer, prompting them to try again later.  
  
Use Case Name: View Product Details   
Use Case ID: UC-03   
Actors: Customer   
Preconditions:   
1. The customer has a valid and active account.   
2. The customer is logged into the system.   
3. The product exists in the inventory database.   
4. The product details are accessible and displayed on the product page.   
  
Postconditions:   
1. The customer has viewed the product details.   
2. The system logs the customer's view of the product.   
3. The customer may add the product to the shopping cart.   
  
Main Flow:   
1. The customer navigates to a product page from the homepage or search results.   
2. The system displays the product details, including name, description, price, and images.   
3. The customer reviews the product information.   
4. The system records the view in the product's view history.   
5. The customer may choose to add the product to their shopping cart.   
6. The system updates the shopping cart with the selected product.   
  
Alternative Flow:   
1. If the product is out of stock, the system displays a message indicating the unavailability.   
2. If the product details cannot be retrieved due to a system error, the system displays an error message and logs the issue.   
3. If the customer does not have permission to view the product (e.g., restricted access), the system denies access and displays an appropriate message.   
4. If the customer closes the page without adding the product to the cart, the system does not persist any changes.  
  
Use Case Name: Add Product to Cart   
Use Case ID: UC-04   
Actors: Customer, Shopping Cart, Inventory, Product   
Preconditions:   
1. The customer has a valid and active account and is logged in.   
2. The product is available in the inventory database.   
3. The shopping cart is accessible and initialized for the customer.   
  
Postconditions:   
1. The product is added to the customer's shopping cart.   
2. The shopping cart is updated with the new item and quantity.   
3. The inventory system reflects the updated product availability.   
  
Main Flow:   
1. The customer views a product on the product details page.   
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 the selected quantity to the customer's shopping cart.   
6. The system updates the cart's total items and price.   
7. The system provides a confirmation message that the product has been successfully added to the cart.   
8. The customer is redirected back to the product page or to the cart page.   
  
Alternative Flow:   
1. If the requested quantity exceeds the available stock, the system displays an error message and suggests the maximum available quantity.   
2. If the system cannot update the shopping cart due to an error, it logs the issue and displays a message to the customer to try again.   
3. If the product is not available (e.g., out of stock or discontinued), the system displays a message and does not add the product to the cart.   
4. If the customer tries to add a product that does not exist in the inventory, the system displays an error and prompts the customer to select a valid product.  
  
Use Case Name: Modify Cart Contents   
Use Case ID: UC-05   
Actors: Customer, Shopping Cart, Product, Inventory   
Preconditions:   
1. The customer has a valid and active account and is logged in.   
2. The shopping cart is initialized for the customer and contains at least one product.   
3. The product to be modified is available in the inventory.   
  
Postconditions:   
1. The customer's shopping cart is updated with the modified product(s) or quantities.   
2. The cart's total items and price are recalculated.   
3. The inventory reflects any changes in product availability based on the cart update.   
  
Main Flow:   
1. The customer navigates to the shopping cart page.   
2. The system displays the current contents of the shopping cart, including product names, quantities, and prices.   
3. The customer selects a product to modify and adjusts the quantity or removes it.   
4. The system verifies the updated quantity against the available inventory.   
5. The system updates the shopping cart with the new quantity or removes the product as requested.   
6. The system recalculates the cart's total price and updates the display.   
7. The system provides a confirmation message that the cart has been updated.   
8. The customer is redirected back to the cart page or another relevant page (e.g., product listing).   
  
Alternative Flow:   
1. If the updated quantity exceeds the available stock, the system displays an error message and suggests the maximum available quantity.   
2. If the system cannot update the cart due to an error, it logs the issue and displays a message to the customer to try again.   
3. If the product is no longer available (e.g., out of stock or discontinued), the system removes it from the cart and displays a message.   
4. If the customer tries to modify a product that is not in the cart, the system displays an error and prompts the customer to select a valid product.  
  
Use Case Name: Remove Product from Cart   
Use Case ID: UC-06   
Actors: Customer, Shopping Cart, Product, Inventory   
  
Preconditions:   
1. The customer has a valid and active account and is logged in.   
2. The shopping cart is initialized for the customer and contains at least one product.   
3. The product to be removed is currently in the shopping cart.   
  
Postconditions:   
1. The selected product is removed from the customer's shopping cart.   
2. The cart's total items and price are recalculated.   
3. The inventory remains unchanged as no stock has been committed to the order.   
  
Main Flow:   
1. The customer navigates to the shopping cart page.   
2. The system displays the current contents of the shopping cart, including product names, quantities, and prices.   
3. The customer selects a product and clicks the "Remove" button.   
4. The system verifies that the product exists in the cart.   
5. The system removes the product from the shopping cart.   
6. The system recalculates the cart's total items and price.   
7. The system provides a confirmation message that the product has been successfully removed.   
8. The customer is redirected back to the cart page or another relevant page (e.g., product listing).   
  
Alternative Flow:   
1. If the customer tries to remove a product that is no longer in the cart, the system displays an error and prompts the customer to select a valid product.   
2. If the system fails to update the cart due to an error, it logs the issue and displays a message to the customer to try again.   
3. If the customer cancels the removal action, the cart remains unchanged and the system returns to the previous state.  
  
Use Case Name: Checkout   
Use Case ID: UC-07   
Actors: Customer, Shopping Cart, Product, Inventory, Order, Payment, Email Confirmation   
  
Preconditions:   
1. The customer has a valid and active account and is logged in.   
2. The shopping cart contains at least one product.   
3. The product(s) in the cart are available in the inventory.   
4. The system supports multiple payment methods.   
5. The system is configured to send email confirmations.   
  
Postconditions:   
1. A new order is created in the system with the selected products and payment details.   
2. The inventory is updated to reflect the reduction in stock for the purchased product(s).   
3. The payment is processed successfully.   
4. The customer receives an email confirmation of the order.   
5. The shopping cart is cleared of the purchased items.   
  
Main Flow:   
1. The customer navigates to the shopping cart page and reviews the contents.   
2. The customer clicks the "Proceed to Checkout" button.   
3. The system displays the checkout form with fields for shipping address, billing address, and payment method.   
4. The customer fills in the required information and selects a payment method.   
5. The customer confirms the order details and clicks "Place Order".   
6. The system verifies the inventory for the product(s) in the cart.   
7. The system creates a new order record and associates it with the customer's account.   
8. The system processes the selected payment method and confirms the transaction.   
9. The system updates the inventory to reduce the stock for the purchased product(s).   
10. The system clears the shopping cart of the purchased items.   
11. The system sends an email confirmation to the customer with the order details.   
12. The customer is redirected to an order confirmation page.   
  
Alternative Flow:   
1. If the customer does not provide a valid shipping or billing address, the system displays an error message and prompts the customer to correct the information.   
2. If the selected payment method fails (e.g., insufficient funds, invalid card details), the system displays an error message and allows the customer to try a different payment method.   
3. If the inventory check fails (e.g., product is no longer available in the requested quantity), the system displays a message and allows the customer to adjust the order.   
4. If the system fails to send the email confirmation, it logs the error and displays a message to the customer. The administrator can manually review and resend the confirmation.   
5. If the customer cancels the checkout process, the shopping cart remains unchanged and the system returns to the cart page.  
  
Use Case Name: Confirm Order   
Use Case ID: UC-08   
Actors: Customer, Shopping Cart, Product, Inventory, Order, Payment, Email Confirmation   
  
Preconditions:   
1. The customer has a valid and active account and is logged in.   
2. The shopping cart contains at least one product.   
3. The customer has completed the checkout process and provided valid shipping and billing information.   
4. The payment method has been selected and validated.   
5. The inventory has sufficient stock for the products in the cart.   
  
Postconditions:   
1. The order is confirmed and marked as pending or completed in the system.   
2. The inventory is updated to reflect the reduction in stock for the purchased product(s).   
3. The payment is processed and recorded.   
4. The customer receives an email confirmation with the order details.   
5. The shopping cart is cleared of the purchased items.   
  
Main Flow:   
1. The customer reviews the order summary on the order confirmation page.   
2. The customer clicks the "Confirm Order" button.   
3. The system verifies all order details, including shipping address, billing address, and payment method.   
4. The system processes the payment transaction.   
5. The system creates a finalized order record and associates it with the customer's account.   
6. The system updates the inventory to reduce the stock for the purchased products.   
7. The system clears the shopping cart of the purchased items.   
8. The system sends an email confirmation to the customer with the order details.   
9. The customer is redirected to an order confirmation page.   
  
Alternative Flow:   
1. If the payment fails, the system displays an error message and allows the customer to try a different payment method.   
2. If the inventory is insufficient at the time of confirmation, the system displays a message and allows the customer to adjust the order or cancel.   
3. If the system fails to send the email confirmation, it logs the error and displays a message to the customer. The administrator can manually review and resend the confirmation.   
4. If the customer cancels the order confirmation, the order is not finalized, and the system returns to the checkout page.  
  
Use Case Name: View Purchase History   
Use Case ID: UC-09   
Actors: Customer, Order, Email Confirmation   
  
Preconditions:   
1. The customer has a valid and active account.   
2. The customer is logged into the system.   
3. The customer has at least one order in the system.   
4. The system is configured to display order history.   
  
Postconditions:   
1. The customer's purchase history is displayed.   
2. The system logs the customer's access to the purchase history.   
3. The customer may view order details or initiate an email confirmation request for a specific order.   
  
Main Flow:   
1. The customer navigates to their account dashboard.   
2. The system displays a "Purchase History" section with a list of previous orders.   
3. The customer selects an order from the list to view details.   
4. The system retrieves and displays the order details, including products, quantities, prices, and order status.   
5. The customer may request an email confirmation for the selected order.   
6. The system sends an email confirmation with the order details to the customer's registered email address.   
7. The system logs the request and updates the order history view.   
8. The customer is redirected back to the purchase history page or another relevant page.   
  
Alternative Flow:   
1. If the customer has no previous orders, the system displays a message indicating that no purchase history is available.   
2. If the system fails to retrieve order details, it displays an error message and logs the issue.   
3. If the system cannot send an email confirmation, it logs the error and displays a message to the customer. The administrator can manually review and resend the confirmation.   
4. If the customer cancels the view or the email request, the system returns to the previous page without making any changes.  
  
Use Case Name: Administrator Login   
Use Case ID: UC-10   
Actors: Administrator   
Preconditions:   
1. The administrator has a valid account and credentials.   
2. The system is operational and accessible.   
3. The administrator has navigated to the login page for administrators.   
  
Postconditions:   
1. The administrator is successfully authenticated and logged into the system.   
2. The system updates the administrator's login status.   
3. The administrator is redirected to the admin dashboard.   
  
Main Flow:   
1. The administrator accesses the admin login page.   
2. The system displays the login form with fields for username and password.   
3. The administrator enters their username and password.   
4. The administrator submits the login form.   
5. The system validates the credentials against the administrator database.   
6. The system confirms the account is active and has valid permissions.   
7. The system authenticates the administrator and generates a session token.   
8. The system redirects the administrator to the admin dashboard.   
  
Alternative Flow:   
1. If the username or password is invalid, the system displays an error message and prompts the administrator to re-enter the correct details.   
2. If the administrator account is inactive or locked, the system displays a message and may redirect to an account recovery or unlock page.   
3. If the system fails to generate a session token, it logs the error and displays a message to the administrator to try again later.   
4. If the login attempt is flagged for security reasons (e.g., multiple failed attempts), the system may lock the account and notify the administrator via email.  
  
Use Case Name: Manage Inventory   
Use Case ID: UC-11   
Actors: Administrator, Inventory, Product, Category, Plugin   
  
Preconditions:   
1. The administrator has a valid and active account.   
2. The administrator is logged into the system.   
3. The system is operational and allows inventory management.   
4. The product and category data are accessible in the system.   
  
Postconditions:   
1. The inventory is updated with new or modified product information.   
2. The product is added, removed, or updated in the inventory database.   
3. The product's category and stock status are synchronized with the inventory.   
4. The system may trigger a plugin to notify relevant stakeholders of inventory changes.   
  
Main Flow:   
1. The administrator navigates to the "Inventory Management" section in the admin dashboard.   
2. The system displays a list of products, along with their stock levels, categories, and other inventory-related details.   
3. The administrator selects an action (e.g., add product, edit product, remove product).   
4. The system prompts the administrator to provide or confirm product details (e.g., name, category, price, stock quantity).   
5. The administrator submits the changes or additions.   
6. The system validates the input data (e.g., correct format, valid category, sufficient stock for removal).   
7. The system updates the inventory database with the new or modified product information.   
8. The system may trigger a plugin to notify relevant stakeholders (e.g., restock alerts, product availability updates).   
9. The system displays a confirmation message that the inventory has been successfully updated.   
10. The administrator is redirected back to the inventory management page.   
  
Alternative Flow:   
1. If the administrator enters invalid product details (e.g., negative stock quantity, missing category), the system displays an error message and prompts for correction.   
2. If the selected product does not exist in the inventory, the system displays an error and prompts the administrator to select a valid product.   
3. If the system fails to update the inventory due to a database error, it logs the issue and displays a message to the administrator to try again.   
4. If the plugin fails to execute (e.g., notification not sent), the system logs the error and allows the administrator to manually trigger the plugin or review the issue.   
5. If the administrator cancels the operation, the inventory remains unchanged and the system returns to the inventory management page.  
  
Use Case Name: Update Product Information   
Use Case ID: UC-12   
Actors: Administrator, Product, Inventory, Category   
  
Preconditions:   
1. The administrator has a valid and active account.   
2. The administrator is logged into the system.   
3. The product to be updated exists in the inventory database.   
4. The system is operational and allows product updates.   
  
Postconditions:   
1. The product information is updated in the inventory database.   
2. The product's category and stock status are synchronized with the updated information.   
3. The system may trigger a plugin to notify relevant stakeholders of the product update.   
4. The product details are reflected on the product page and in the shopping cart.   
  
Main Flow:   
1. The administrator navigates to the "Product Management" section in the admin dashboard.   
2. The system displays a list of products along with their details (e.g., name, category, price, stock quantity).   
3. The administrator selects a product to update and clicks the "Edit" button.   
4. The system opens an update form with the current product information pre-filled.   
5. The administrator modifies the product details (e.g., price, description, category, stock quantity).   
6. The administrator submits the updated information.   
7. The system validates the input data (e.g., correct format, valid category, non-negative stock quantity).   
8. The system updates the product record in the inventory database.   
9. The system may trigger a plugin to notify stakeholders (e.g., restock alerts, price changes).   
10. The system displays a confirmation message that the product information has been successfully updated.   
11. The administrator is redirected back to the product management page.   
  
Alternative Flow:   
1. If the administrator enters invalid or incomplete product information, the system displays an error message and prompts for correction.   
2. If the selected product does not exist in the inventory, the system displays an error and prompts the administrator to select a valid product.   
3. If the system fails to update the product in the database due to an error, it logs the issue and displays a message to the administrator to try again.   
4. If the plugin fails to execute (e.g., notification not sent), the system logs the error and allows the administrator to manually trigger the plugin or review the issue.   
5. If the administrator cancels the update operation, the product information remains unchanged and the system returns to the product management page.  
  
Use Case Name: Manage Categories   
Use Case ID: UC-13   
Actors: Administrator, Category, Product, Inventory, Plugin   
Preconditions:   
1. The administrator has a valid and active account and is logged in.   
2. The system is operational and allows category management.   
3. The category data is accessible in the system.   
4. The product and inventory systems are synchronized with the category information.   
  
Postconditions:   
1. The category is added, edited, or removed in the system.   
2. Products are updated with the new category information if needed.   
3. The inventory reflects the updated category status.   
4. The system may trigger a plugin to notify relevant stakeholders of category changes.   
  
Main Flow:   
1. The administrator navigates to the "Category Management" section in the admin dashboard.   
2. The system displays a list of existing categories along with their details (e.g., name, description, number of products).   
3. The administrator selects an action (e.g., add new category, edit existing category, remove category).   
4. The system prompts the administrator to provide or confirm category details (e.g., name, description, parent category if applicable).   
5. The administrator submits the changes or additions.   
6. The system validates the input data (e.g., unique category name, valid format).   
7. The system updates the category database with the new or modified information.   
8. The system updates product records if the category of a product is changed.   
9. The system may trigger a plugin to notify stakeholders (e.g., website updates, product reclassification).   
10. The system displays a confirmation message that the category has been successfully updated.   
11. The administrator is redirected back to the category management page.   
  
Alternative Flow:   
1. If the administrator enters an invalid or duplicate category name, the system displays an error message and prompts for a valid, unique name.   
2. If the selected category does not exist, the system displays an error and prompts the administrator to select a valid category.   
3. If the system fails to update the category due to a database error, it logs the issue and displays a message to the administrator to try again.   
4. If the plugin fails to execute (e.g., notification not sent), the system logs the error and allows the administrator to manually trigger the plugin or review the issue.   
5. If the administrator cancels the operation, the category remains unchanged and the system returns to the category management page.  
  
Use Case Name: Plugin Installation   
Use Case ID: UC-14   
Actors: Administrator, Plugin   
Preconditions:   
1. The administrator has a valid and active account and is logged into the system.   
2. The system is operational and allows plugin installation.   
3. The plugin to be installed is available and compatible with the system.   
4. The administrator has the necessary permissions to install plugins.   
  
Postconditions:   
1. The plugin is successfully installed and activated in the system.   
2. The system configuration is updated to reflect the new plugin.   
3. The plugin is ready for use and may affect system functionality (e.g., notification, analytics).   
4. The administrator is notified of the installation status.   
  
Main Flow:   
1. The administrator navigates to the "Plugin Management" section in the admin dashboard.   
2. The system displays a list of available plugins, including their names, descriptions, and installation status.   
3. The administrator selects a plugin to install and clicks the "Install" button.   
4. The system validates the plugin's compatibility and checks for any dependencies.   
5. The system downloads and installs the plugin.   
6. The system activates the plugin and updates the system configuration.   
7. The system displays a confirmation message that the plugin has been successfully installed.   
8. The administrator is redirected back to the plugin management page.   
  
Alternative Flow:   
1. If the plugin is already installed, the system displays a message indicating that the plugin is active and no further action is required.   
2. If the plugin is incompatible or missing dependencies, the system displays an error and prompts the administrator to resolve the issue before proceeding.   
3. If the system fails to install the plugin due to a technical error, it logs the issue and displays a message to the administrator to try again.   
4. If the administrator cancels the installation, the plugin remains uninstalled and the system returns to the plugin management page.  
  
Use Case Name: Email Confirmation   
Use Case ID: UC-15   
Actors: Customer, Email Confirmation   
Preconditions:   
1. The customer has a valid and active account.   
2. The customer is logged into the system.   
3. The system is configured to send email confirmations.   
4. The customer has requested an email confirmation for an order or registration.   
  
Postconditions:   
1. The customer receives a confirmation email with the requested information.   
2. The system logs the email confirmation request.   
3. The customer may take further action based on the confirmation email (e.g., verify account, view order details).   
  
Main Flow:   
1. The customer requests an email confirmation for an order or account-related action.   
2. The system generates the confirmation email with relevant details (e.g., order number, date, items, or account activation link).   
3. The system sends the confirmation email to the customer's registered email address.   
4. The system logs the request and the email sending action.   
5. The system provides a message to the customer indicating that the confirmation email has been sent.   
6. The customer receives the email and clicks the confirmation link or takes the required action.   
7. The system verifies the confirmation request and updates the relevant record (e.g., marks the account as verified or confirms the order status).   
8. The customer is redirected to a confirmation page or receives a success message.   
  
Alternative Flow:   
1. If the system fails to send the email due to technical issues, it logs the error and displays a message to the customer. The administrator can manually review and resend the confirmation.   
2. If the customer's email address is invalid or unreachable, the system displays an error message and prompts the customer to update their email address.   
3. If the confirmation link is clicked after it has expired, the system displays an error and prompts the customer to request a new confirmation.   
4. If the customer does not take any action on the confirmation email, the system may mark the confirmation as pending or require re-submission of the request.  
  
Use Case Name: View Order Status   
Use Case ID: UC-10   
Actors: Customer, Order   
Preconditions:   
1. The customer has a valid and active account.   
2. The customer is logged into the system.   
3. The customer has at least one order in the system.   
4. The system is operational and allows customers to view order status.   
  
Postconditions:   
1. The customer is shown the current status of their order (e.g., pending, shipped, delivered).   
2. The system logs the customer's access to the order status.   
3. The customer may take further actions based on the status (e.g., cancel order, contact support).   
  
Main Flow:   
1. The customer navigates to their account dashboard.   
2. The system displays a "My Orders" section with a list of orders.   
3. The customer selects an order to view its status.   
4. The system retrieves the selected order from the order database.   
5. The system displays the order details, including status, order date, expected delivery date, and items.   
6. The customer reviews the status and other relevant information.   
7. The system provides an option to take further actions, if applicable (e.g., cancel order).   
8. The customer is redirected back to the "My Orders" page.   
  
Alternative Flow:   
1. If the customer has no orders in the system, the system displays a message indicating that no orders are available.   
2. If the system fails to retrieve the order details, it logs the error and displays a message to the customer to try again later.   
3. If the order status has changed recently, the system ensures the displayed information is up to date.   
4. If the customer attempts to view an order that does not exist in the system, the system displays an error and prompts the customer to select a valid order.  
  
Use Case Name: Manage Cart Item   
Use Case ID: UC-16   
Actors: Customer, Shopping Cart, Product, Inventory   
  
Preconditions:   
1. The customer has a valid and active account and is logged in.   
2. The shopping cart is initialized for the customer and contains at least one item.   
3. The product associated with the cart item is available in the inventory database.   
4. The system is operational and allows customers to modify cart items.   
  
Postconditions:   
1. The cart item is updated, removed, or replaced as requested by the customer.   
2. The shopping cart's total items and price are recalculated.   
3. The inventory system reflects any changes in product availability based on the cart item modification.   
4. The system provides a confirmation message to the customer regarding the cart item change.   
  
Main Flow:   
1. The customer navigates to the shopping cart page.   
2. The system displays the current contents of the shopping cart, including product names, quantities, prices, and total cost.   
3. The customer selects a specific cart item for modification (e.g., change quantity, replace product, remove item).   
4. The system verifies that the selected cart item exists in the customer's cart.   
5. If the customer adjusts the quantity, the system checks the updated quantity against the available inventory.   
6. If the customer replaces the product, the system displays options for available products and validates the new product's availability.   
7. The system updates the cart item based on the customer's request (e.g., updates quantity, replaces product, removes item).   
8. The system recalculates the cart's total items and price.   
9. The system provides a confirmation message that the cart item has been successfully modified.   
10. The customer is redirected back to the cart page or another relevant page (e.g., product listing).   
  
Alternative Flow:   
1. If the updated quantity exceeds the available stock, the system displays an error message and suggests the maximum available quantity.   
2. If the customer attempts to replace a product with one that is not available, the system displays an error message and prompts for a valid replacement.   
3. If the system cannot update the cart item due to an error, it logs the issue and displays a message to the customer to try again.   
4. If the customer cancels the modification, the cart item remains unchanged and the system returns to the cart page.   
5. If the selected cart item no longer exists in the cart (e.g., already removed), the system displays an error and prompts the customer to select a valid cart item.  
  
Use Case Name: Manage Order Item   
Use Case ID: UC-17   
Actors: Administrator, Order, Order Item, Product, Inventory   
  
Preconditions:   
1. The administrator has a valid and active account and is logged into the system.   
2. The system is operational and allows administrators to manage order items.   
3. The order and its associated order items exist in the database.   
4. The product data is accessible and synchronized with the order items.   
5. The administrator has the necessary permissions to modify order items.   
  
Postconditions:   
1. The order item is updated, removed, or replaced as requested by the administrator.   
2. The order details are updated to reflect the changes in order items.   
3. The inventory is adjusted accordingly if the modification affects product stock.   
4. The system logs the administrator's action on the order item.   
5. The customer may receive an email notification of the order item change.   
  
Main Flow:   
1. The administrator navigates to the "Order Management" section in the admin dashboard.   
2. The system displays a list of orders with relevant details (e.g., order number, date, customer, and status).   
3. The administrator selects an order and clicks to view its items.   
4. The system retrieves and displays the list of order items, including product names, quantities, prices, and status.   
5. The administrator selects an order item for modification (e.g., update quantity, replace product, remove item).   
6. If the administrator updates the quantity, the system checks the updated quantity against the available inventory.   
7. If the administrator replaces the product, the system displays options for available products and validates the new product's availability.   
8. The system updates the order item based on the administrator's request.   
9. The system recalculates the order's total price and updates the order record.   
10. If the modification affects inventory (e.g., reducing or increasing stock), the system updates the inventory accordingly.   
11. The system may trigger an email confirmation to the customer to inform them of the order item change.   
12. The system logs the administrator's action and displays a confirmation message that the order item has been successfully modified.   
13. The administrator is redirected back to the order management page or to the selected order's details.   
  
Alternative Flow:   
1. If the updated quantity exceeds the available stock, the system displays an error message and suggests the maximum available quantity.   
2. If the administrator attempts to replace an order item with a product that is not available, the system displays an error message and prompts for a valid replacement.   
3. If the system cannot update the order item due to a database error, it logs the issue and displays a message to the administrator to try again.   
4. If the inventory update fails (e.g., due to insufficient permissions or technical issues), the system displays an error message and logs the issue for review.   
5. If the system fails to send the email confirmation to the customer, it logs the error and displays a message to the administrator. The administrator can manually review and resend the email if needed.   
6. If the administrator cancels the modification, the order item remains unchanged and the system returns to the order management page.  
  
Use Case Name: Manage Product Update Record   
Use Case ID: UC-18   
Actors: Administrator, Product, Inventory, ProductUpdateRecord, Plugin   
  
Preconditions:   
1. The administrator has a valid and active account and is logged into the system.   
2. The system is operational and allows administrators to view and manage product update records.   
3. The product update records exist in the database and are associated with specific products.   
4. The administrator has the necessary permissions to access and manage the product update history.   
  
Postconditions:   
1. The product update record is viewed, edited, or deleted as requested by the administrator.   
2. The system logs the administrator's action on the product update record.   
3. If a product update record is deleted or modified, the system ensures data integrity and consistency with the product and inventory records.   
4. The system may trigger a plugin to notify relevant stakeholders of changes to the product update record.   
  
Main Flow:   
1. The administrator navigates to the "Product Update History" section in the admin dashboard.   
2. The system displays a list of product update records, including the product name, date of update, updated fields, and the user who made the update.   
3. The administrator selects a specific product update record for viewing, editing, or deletion.   
4. The system retrieves and displays the detailed information of the selected product update record.   
5. If the administrator chooses to edit the record, the system allows them to modify the relevant details (e.g., comments, timestamp, or user reference).   
6. If the administrator chooses to delete the record, the system prompts for confirmation and verifies the deletion is allowed (e.g., not a critical system record).   
7. The system updates the product update record database with the new information or removes the record if deleted.   
8. The system may trigger a plugin to notify stakeholders about the modification or deletion of a product update record.   
9. The system logs the administrator's action and displays a confirmation message that the product update record has been successfully modified or deleted.   
10. The administrator is redirected back to the product update history page.   
  
Alternative Flow:   
1. If the selected product update record does not exist in the database, the system displays an error message and prompts the administrator to select a valid record.   
2. If the system fails to retrieve the product update record due to an error, it logs the issue and displays a message to the administrator to try again.   
3. If the administrator attempts to edit a record with invalid or inconsistent data (e.g., invalid timestamp format), the system displays an error message and prompts for correction.   
4. If the deletion of a product update record is not permitted (e.g., it is a historical audit record), the system displays a warning and prevents the deletion.   
5. If the plugin fails to execute (e.g., notification not sent), the system logs the error and allows the administrator to manually trigger the plugin or review the issue.   
6. If the administrator cancels the operation, the product update record remains unchanged and the system returns to the product update history page.