项目文档

# Functional Requirement

# 1. Functional Requirements  
  
## 1.1 Customer Registration Function   
\*\*Function ID\*\*: FR-01   
\*\*Description\*\*: Customers can create new accounts by providing their personal information, including name, email, and password. The system validates the email and password, stores the account information, and sends a confirmation email.   
\*\*Input\*\*: Full name, email address, password   
\*\*Output\*\*: Registered customer account, confirmation email sent to customer, success message displayed to customer   
  
## 1.2 Customer Login Function   
\*\*Function ID\*\*: FR-02   
\*\*Description\*\*: Registered customers can log in using their email and password. The system validates the credentials, checks account status, and initiates a session.   
\*\*Input\*\*: Email address, password   
\*\*Output\*\*: Active customer session, redirect to dashboard or homepage, success message displayed to customer   
  
## 1.3 Customer Logout Function   
\*\*Function ID\*\*: FR-03   
\*\*Description\*\*: Customers can log out of their account. The system terminates the session and redirects the customer to the homepage or login page.   
\*\*Input\*\*: Logout request from customer   
\*\*Output\*\*: Terminated customer session, redirect to homepage or login page, success message displayed to customer   
  
## 1.4 Create Product Function   
\*\*Function ID\*\*: FR-04   
\*\*Description\*\*: Administrators can create new products by entering product details such as name, description, price, category, and inventory quantity. The system validates the input and stores the product information.   
\*\*Input\*\*: Product name, description, price, category, inventory quantity, product images   
\*\*Output\*\*: New product record in the database, unique product ID assigned, product displayed in inventory   
  
## 1.5 Manage Inventory Function   
\*\*Function ID\*\*: FR-05   
\*\*Description\*\*: Administrators can update the inventory quantity of existing products. The system ensures the quantity is non-negative and updates the inventory record accordingly.   
\*\*Input\*\*: Product ID, updated inventory quantity   
\*\*Output\*\*: Updated inventory record in the database, inventory status reflected in product listing   
  
## 1.6 View Product Details Function   
\*\*Function ID\*\*: FR-06   
\*\*Description\*\*: Customers can view detailed information about a product, including name, description, price, category, and inventory status.   
\*\*Input\*\*: Product ID or name   
\*\*Output\*\*: Product details displayed to customer, including price, description, and availability   
  
## 1.7 Add to Cart Function   
\*\*Function ID\*\*: FR-07   
\*\*Description\*\*: Customers can add products to their shopping cart by selecting a product and specifying the quantity. The system checks inventory and updates the cart.   
\*\*Input\*\*: Product ID, quantity   
\*\*Output\*\*: Updated shopping cart with added product, updated total cost, confirmation message displayed to customer   
  
## 1.8 Modify Cart Item Function   
\*\*Function ID\*\*: FR-08   
\*\*Description\*\*: Customers can modify the quantity of an item in their cart or remove it. The system recalculates the total cost after modification.   
\*\*Input\*\*: Cart item ID, new quantity or remove request   
\*\*Output\*\*: Updated cart contents, recalculated total cost, confirmation message displayed to customer   
  
## 1.9 Checkout Order Function   
\*\*Function ID\*\*: FR-09   
\*\*Description\*\*: Customers can proceed to checkout, enter shipping details, select a payment method, and complete the transaction. The system processes the payment, updates the inventory, and clears the cart.   
\*\*Input\*\*: Cart contents, shipping address, shipping method, payment method, payment details   
\*\*Output\*\*: Confirmed order in database, updated inventory, cleared cart, order confirmation email sent, success message displayed to customer   
  
## 1.10 Confirm Order Function   
\*\*Function ID\*\*: FR-10   
\*\*Description\*\*: The system confirms the order after successful payment and updates the order status. It also reduces inventory and sends a confirmation email to the customer.   
\*\*Input\*\*: Payment confirmation, order details   
\*\*Output\*\*: Confirmed order status in database, updated inventory, order confirmation email sent to customer, administrator notified of new order   
  
## 1.11 Send Order Confirmation Email Function   
\*\*Function ID\*\*: FR-11   
\*\*Description\*\*: The system generates and sends an email to the customer confirming the order. The email includes order details and delivery information.   
\*\*Input\*\*: Customer email address, order details   
\*\*Output\*\*: Order confirmation email sent to customer, updated order status to "Email Sent", administrator notified of email sent   
  
## 1.12 View Purchase History Function   
\*\*Function ID\*\*: FR-12   
\*\*Description\*\*: Customers can view their past orders in the purchase history section. The system displays order details, including date, total cost, and items.   
\*\*Input\*\*: Customer login status, request to view purchase history   
\*\*Output\*\*: List of past orders displayed to customer, including order details and status   
  
## 1.13 Update Personal Information Function   
\*\*Function ID\*\*: FR-13   
\*\*Description\*\*: Customers can update their personal information, such as address, phone number, and date of birth. The system validates the input and updates the stored information.   
\*\*Input\*\*: Updated personal information (e.g., address, phone number)   
\*\*Output\*\*: Updated customer record in the database, success message displayed to customer   
  
## 1.14 Admin Login Function   
\*\*Function ID\*\*: FR-14   
\*\*Description\*\*: Administrators can log in to the system using their email and password. The system validates the credentials, checks account status, and initiates a session.   
\*\*Input\*\*: Email address, password   
\*\*Output\*\*: Active admin session, redirect to admin dashboard, success message displayed to administrator   
  
## 1.15 Manage Categories Function   
\*\*Function ID\*\*: FR-15   
\*\*Description\*\*: Administrators can create, update, or delete product categories. The system validates the input and updates the category list accordingly.   
\*\*Input\*\*: Category name, description, action (create, update, delete)   
\*\*Output\*\*: Updated category list in the database, changes reflected in product listings and navigation   
  
## 1.16 Update Product Details Function   
\*\*Function ID\*\*: FR-16   
\*\*Description\*\*: Administrators can update product details such as name, description, price, and category. The system validates the input and reflects the changes in the database.   
\*\*Input\*\*: Product ID, updated product details (e.g., name, description, price, category, images)   
\*\*Output\*\*: Updated product record in the database, changes reflected in product listing   
  
## 1.17 Install Plugin Function   
\*\*Function ID\*\*: FR-17   
\*\*Description\*\*: Administrators can install new plugins to extend the system's functionality. The system checks plugin compatibility and installs it if valid.   
\*\*Input\*\*: Plugin name or identifier, installation request   
\*\*Output\*\*: Installed plugin in the system, updated plugin list, success message displayed to administrator   
  
## 1.18 Access Plugin API Documentation Function   
\*\*Function ID\*\*: FR-18   
\*\*Description\*\*: Administrators can access the documentation for the Plugin API. The system retrieves and displays the documentation in a user-friendly format.   
\*\*Input\*\*: Plugin name or identifier, request to view documentation   
\*\*Output\*\*: Plugin API documentation displayed to administrator, including endpoints and usage examples   
  
## 1.19 Manage User Accounts Function   
\*\*Function ID\*\*: FR-19   
\*\*Description\*\*: Administrators can view, update, or delete user accounts. The system validates the input and updates the account status or removes the account if valid.   
\*\*Input\*\*: User account ID or email, action (view, update, delete), updated user details   
\*\*Output\*\*: Updated or deleted user account in the database, confirmation message displayed to administrator, affected user notified via email if needed   
  
## 1.20 Manage Payment Method Function   
\*\*Function ID\*\*: FR-20   
\*\*Description\*\*: Customers can view, add, update, or delete their saved payment methods. The system validates the input and updates the payment method records accordingly.   
\*\*Input\*\*: Payment method details (e.g., card number, expiration date), action (add, update, delete)   
\*\*Output\*\*: Updated payment method records in the database, confirmation message displayed to customer, updated payment method list shown   
  
## 1.21 Calculate Total Cost Function   
\*\*Function ID\*\*: FR-21   
\*\*Description\*\*: The system calculates the total cost of items in the shopping cart based on item prices and quantities. It also applies any configured discounts or taxes.   
\*\*Input\*\*: Cart item IDs and quantities   
\*\*Output\*\*: Total cost displayed to customer, including discounts and taxes if applicable   
  
These functional requirements form the core of the GAMMA-J Web Store system, covering customer and administrator interactions, shopping cart operations, order processing, and plugin management capabilities.

# External Description

# Chapter 2: External Interfaces  
  
This chapter defines the external interfaces that the GAMMA-J Web Store system interacts with. These interfaces are categorized into \*\*User Interface\*\*, \*\*Hardware Interface\*\*, \*\*Software Interface\*\*, and \*\*Communication Interface\*\*. Each interface is described in detail, including its role, interaction method, and input/output data.  
  
---  
  
## 2.1 User Interface  
  
### 2.1.1 Customer Registration UI  
- \*\*Description\*\*: The user interface allows customers to enter their personal information (name, email, password) to register for an account.  
- \*\*Role\*\*: Facilitates customer registration by presenting input fields and displaying success or error messages.  
- \*\*Input\*\*: Full name, email address, password.  
- \*\*Output\*\*: Display of success message, confirmation email, and redirect to login page if successful.  
- \*\*Interaction Method\*\*: Web form submission via HTTP POST request.  
  
### 2.1.2 Customer Login UI  
- \*\*Description\*\*: The user interface allows registered customers to log in using their email and password.  
- \*\*Role\*\*: Provides a secure login mechanism with validation feedback.  
- \*\*Input\*\*: Email address, password.  
- \*\*Output\*\*: Redirect to dashboard or homepage upon successful login, display error message on failure.  
- \*\*Interaction Method\*\*: Web form submission via HTTP POST request.  
  
### 2.1.3 Customer Logout UI  
- \*\*Description\*\*: The user interface allows customers to log out of their session.  
- \*\*Role\*\*: Terminates the current session and redirects the user to the homepage or login page.  
- \*\*Input\*\*: Logout button click.  
- \*\*Output\*\*: Redirect to homepage or login page, display success message.  
- \*\*Interaction Method\*\*: Web form submission or JavaScript event trigger.  
  
### 2.1.4 Product Detail View UI  
- \*\*Description\*\*: The user interface displays detailed product information such as name, description, price, category, and inventory status.  
- \*\*Role\*\*: Enables customers to view product details before purchasing.  
- \*\*Input\*\*: Product ID or name.  
- \*\*Output\*\*: Display of product details, availability, and price.  
- \*\*Interaction Method\*\*: AJAX call or direct URL access based on product ID.  
  
### 2.1.5 Shopping Cart UI  
- \*\*Description\*\*: The user interface allows customers to add, modify, or remove items from their shopping cart.  
- \*\*Role\*\*: Manages the visual representation of the cart and provides real-time updates.  
- \*\*Input\*\*: Product ID, quantity, or remove action.  
- \*\*Output\*\*: Updated cart contents, total cost, and confirmation message.  
- \*\*Interaction Method\*\*: AJAX call or direct form submission.  
  
### 2.1.6 Checkout UI  
- \*\*Description\*\*: The user interface allows customers to proceed through the checkout process, entering shipping and payment details.  
- \*\*Role\*\*: Guides the customer through the final steps of purchasing.  
- \*\*Input\*\*: Shipping address, shipping method, payment method, payment details.  
- \*\*Output\*\*: Confirmation of order, updated cart, and redirect to order confirmation page.  
- \*\*Interaction Method\*\*: Multi-step form submission via HTTP POST requests.  
  
### 2.1.7 Purchase History UI  
- \*\*Description\*\*: The user interface displays a list of past orders for logged-in customers.  
- \*\*Role\*\*: Provides visibility into previous purchases and order status.  
- \*\*Input\*\*: Customer login status.  
- \*\*Output\*\*: Display of order history, including date, total cost, and items.  
- \*\*Interaction Method\*\*: AJAX call or direct query based on user session.  
  
### 2.1.8 Admin Dashboard UI  
- \*\*Description\*\*: The user interface provides administrators with tools to manage users, products, categories, and plugins.  
- \*\*Role\*\*: Central hub for administrative functions.  
- \*\*Input\*\*: Admin login credentials, actions (create, update, delete).  
- \*\*Output\*\*: Display of relevant admin panels, success/error messages.  
- \*\*Interaction Method\*\*: Web form submission or API calls.  
  
---  
  
## 2.2 Hardware Interface  
  
### 2.2.1 No Hardware Interfaces  
- \*\*Description\*\*: The GAMMA-J Web Store system does not interact directly with external hardware devices.  
- \*\*Role\*\*: No hardware communication is required.  
- \*\*Input/Output\*\*: None.  
- \*\*Interaction Method\*\*: N/A.  
  
---  
  
## 2.3 Software Interface  
  
### 2.3.1 Database Interface  
- \*\*Description\*\*: The system communicates with a database to store and retrieve user accounts, products, orders, and other data.  
- \*\*Role\*\*: Central repository for persistent storage of application data.  
- \*\*Input\*\*: Structured data (e.g., JSON, SQL queries).  
- \*\*Output\*\*: Data retrieved from or stored in the database.  
- \*\*Interaction Method\*\*: RESTful API calls or direct SQL queries depending on the backend architecture.  
  
### 2.3.2 Email Service Interface  
- \*\*Description\*\*: The system uses an email service to send confirmation emails, order confirmations, and notifications.  
- \*\*Role\*\*: Enables automated communication with users.  
- \*\*Input\*\*: Customer email address, order details, confirmation message.  
- \*\*Output\*\*: Sent email notification.  
- \*\*Interaction Method\*\*: SMTP API or third-party email service integration (e.g., SendGrid, Mailgun).  
  
### 2.3.3 Payment Gateway Interface  
- \*\*Description\*\*: The system integrates with a payment gateway to process transactions and handle payments.  
- \*\*Role\*\*: Securely processes payment information and confirms transaction status.  
- \*\*Input\*\*: Payment method details (e.g., card number, expiration date), amount, currency.  
- \*\*Output\*\*: Payment confirmation, transaction ID, success/failure status.  
- \*\*Interaction Method\*\*: RESTful API calls to a payment service provider (e.g., Stripe, PayPal).  
  
### 2.3.4 Plugin Management Interface  
- \*\*Description\*\*: The system interacts with plugin APIs to extend functionality through installed plugins.  
- \*\*Role\*\*: Allows dynamic addition and management of features.  
- \*\*Input\*\*: Plugin name, installation request, API endpoint.  
- \*\*Output\*\*: Installed plugin, updated plugin list, API documentation.  
- \*\*Interaction Method\*\*: RESTful API calls to plugin endpoints or local file system access.  
  
### 2.3.5 External API Integration Interface  
- \*\*Description\*\*: The system may integrate with external services (e.g., shipping providers, tax calculators) via APIs.  
- \*\*Role\*\*: Enhances functionality by leveraging external data and services.  
- \*\*Input\*\*: API keys, request parameters, data format (e.g., JSON).  
- \*\*Output\*\*: API response data, processed results.  
- \*\*Interaction Method\*\*: RESTful API calls to external services.  
  
---  
  
## 2.4 Communication Interface  
  
### 2.4.1 Email Communication Interface  
- \*\*Description\*\*: The system sends emails to customers for order confirmation, account verification, and other notifications.  
- \*\*Role\*\*: Ensures timely communication with users.  
- \*\*Input\*\*: Recipient email address, message content, subject line.  
- \*\*Output\*\*: Sent email message.  
- \*\*Interaction Method\*\*: SMTP protocol or email service API (e.g., SendGrid, Mailgun).  
  
### 2.4.2 Web Browsing Interface  
- \*\*Description\*\*: The system is accessed via web browsers, enabling user interaction through HTTP/HTTPS protocols.  
- \*\*Role\*\*: Enables user access to all system functionalities via standard web clients.  
- \*\*Input\*\*: HTTP GET/POST requests, user input via forms or links.  
- \*\*Output\*\*: HTML pages, JSON responses, redirects.  
- \*\*Interaction Method\*\*: HTTP/HTTPS communication between client and server.  
  
### 2.4.3 Session Management Interface  
- \*\*Description\*\*: The system maintains user sessions across interactions to preserve login state and cart contents.  
- \*\*Role\*\*: Ensures continuity of user experience during multiple visits.  
- \*\*Input\*\*: Session token, user authentication credentials.  
- \*\*Output\*\*: Active session, session expiration, logout signal.  
- \*\*Interaction Method\*\*: Cookie-based or token-based session management via HTTP headers.  
  
---  
  
## Summary of External Data Sources  
  
| External Data Source | Type | Description |  
|----------------------|--------------|-------------|  
| Database | Software | Stores user accounts, products, orders, etc. |  
| Email Service | Communication| Sends confirmation and order emails |  
| Payment Gateway | Software | Processes transactions and payment methods |  
| Plugin API | Software | Extends system functionality via plugins |  
| External APIs | Software | Integrates with third-party services |  
| User Interface | User Interface | Facilitates user interaction |  
| Web Browser | Communication| Enables access to the system via HTTP |  
| Session Management | Communication| Maintains user session state |  
  
---  
  
This section ensures that all external data sources and interfaces referenced in the functional requirements are clearly defined, categorized, and described, providing a comprehensive foundation for system design and implementation.

# Use Case

Use Case Name: Customer Registration   
Use Case ID: UC-01   
Actors: Customer, System   
Preconditions:   
- The customer is not yet registered in the system.   
- The system is accessible and operational.   
  
Postconditions:   
- The customer is successfully registered in the system.   
- A confirmation email is sent to the customer.   
- The customer can now log in using their registered email and password.   
  
Main Flow:   
1. The customer navigates to the registration page.   
2. The customer enters their personal information, including full name, email address, and password.   
3. The system validates the email format and checks if the email is already registered.   
4. The system confirms that the password meets the required criteria (e.g., length, complexity).   
5. The system saves the customer's information to the database.   
6. The system generates and sends a confirmation email to the customer's email address.   
7. The customer receives the confirmation email and clicks on the link provided.   
8. The system confirms the customer's email and completes the registration process.   
9. The customer is redirected to the login page and notified of successful registration.   
  
Alternative Flow:   
1. If the email format is invalid, the system displays an error message and prompts the customer to re-enter the email.   
2. If the email is already registered, the system displays an error message indicating that the email is taken and suggests using a different email.   
3. If the password does not meet the required criteria, the system displays an error message and prompts the customer to re-enter a valid password.   
4. If the system fails to send the confirmation email (e.g., due to server issues), it displays an error message and allows the customer to request the email to be resent.   
5. If the customer does not click the confirmation link within a specified time period, the registration remains unconfirmed, and the system prompts the customer to request a new confirmation email.  
  
Use Case Name: Customer Login   
Use Case ID: UC-02   
Actors: Customer, System   
  
Preconditions:   
- The customer is already registered in the system.   
- The customer has an active account with a confirmed email address.   
- The system is accessible and operational.   
  
Postconditions:   
- The customer is successfully authenticated and logged in.   
- The customer's session is initiated.   
- The customer is redirected to their account dashboard or the homepage.   
  
Main Flow:   
1. The customer navigates to the login page.   
2. The customer enters their registered email address and password.   
3. The system validates the email and password against the database.   
4. The system confirms that the account is active and the email has been verified.   
5. The system creates a session for the customer and stores relevant session data.   
6. The customer is redirected to their account dashboard or the homepage.   
7. A success message is displayed to the customer indicating they are logged in.   
  
Alternative Flow:   
1. If the email is not registered, the system displays an error message and prompts the customer to check their email or register.   
2. If the password is incorrect, the system displays an error message and allows the customer to retry entering the password.   
3. If the account is inactive or the email is unconfirmed, the system displays a message and provides an option to resend the confirmation email.   
4. If the system detects multiple failed login attempts, it temporarily locks the account and notifies the customer via email.   
5. If the customer forgets their password, they can click the "Forgot Password" link to initiate a password reset process.  
  
Use Case Name: Customer Logout   
Use Case ID: UC-03   
Actors: Customer, System   
  
Preconditions:   
- The customer is currently logged in to their account.   
- The system is accessible and operational.   
  
Postconditions:   
- The customer's session is terminated.   
- The customer is redirected to the homepage or login page.   
- The system clears session data associated with the customer.   
  
Main Flow:   
1. The customer navigates to the account settings or user menu.   
2. The customer selects the "Logout" option.   
3. The system confirms the logout request.   
4. The system terminates the active session and clears session cookies or tokens.   
5. The customer is redirected to the homepage or login page.   
6. A confirmation message is displayed to the customer indicating they have been successfully logged out.   
  
Alternative Flow:   
1. If the system fails to terminate the session due to technical issues, it displays an error message and prompts the customer to try again or contact support.   
2. If the customer closes the browser without logging out, the system automatically logs them out after a period of inactivity as defined by the session timeout policy.   
3. If the customer tries to access restricted pages after logging out, the system redirects them to the login page and displays an access denied message.  
  
Use Case Name: Create Product   
Use Case ID: UC-04   
Actors: Administrator, System   
  
Preconditions:   
- The administrator is logged in to their account.   
- The administrator has the necessary permissions to create products.   
- The system is accessible and operational.   
  
Postconditions:   
- A new product is successfully added to the system.   
- The product information is saved in the database.   
- The product is displayed in the product listing or inventory.   
  
Main Flow:   
1. The administrator navigates to the product management section of the system.   
2. The administrator selects the "Create Product" option.   
3. The system displays a product creation form with fields for product name, description, price, category, and inventory quantity.   
4. The administrator fills out the product form with the required details.   
5. The administrator uploads product images or other relevant files if applicable.   
6. The administrator submits the product creation request.   
7. The system validates the input data, including checking for duplicate product names and valid price formats.   
8. The system saves the product details to the database and assigns a unique product ID.   
9. The system displays a success message confirming the product was created.   
10. The new product is added to the inventory and appears in the product listing.   
  
Alternative Flow:   
1. If the product name is already taken, the system displays an error message and prompts the administrator to enter a different name.   
2. If the price format is invalid, the system displays an error message and prompts the administrator to re-enter a valid price.   
3. If required fields are left empty, the system displays an error message and prompts the administrator to complete all mandatory fields.   
4. If the file upload fails (e.g., unsupported format, too large), the system displays an error message and allows the administrator to re-upload the file.   
5. If the system fails to save the product due to technical issues, it displays an error message and prompts the administrator to try again or contact support.  
  
Use Case Name: Manage Inventory   
Use Case ID: UC-05   
Actors: Administrator, System   
  
Preconditions:   
- The administrator is logged in to their account.   
- The administrator has the necessary permissions to manage inventory.   
- The system is accessible and operational.   
  
Postconditions:   
- The inventory is updated with the latest product quantities.   
- The system reflects changes in product availability.   
- A confirmation message is displayed to the administrator.   
  
Main Flow:   
1. The administrator navigates to the inventory management section.   
2. The administrator selects the product they wish to update.   
3. The system displays the current inventory quantity and product details.   
4. The administrator modifies the inventory quantity (e.g., increase or decrease stock).   
5. The administrator submits the updated inventory information.   
6. The system validates the input data and confirms the new quantity is non-negative.   
7. The system updates the inventory record in the database.   
8. The system displays a success message indicating the inventory was updated.   
9. The updated inventory is reflected in the product listing and inventory dashboard.   
  
Alternative Flow:   
1. If the product is not found, the system displays an error message and prompts the administrator to enter a valid product ID.   
2. If the new inventory quantity is negative, the system displays an error message and prompts the administrator to enter a valid quantity.   
3. If the system fails to update the inventory due to database errors, it displays an error message and allows the administrator to retry the operation.   
4. If the administrator tries to update a product without sufficient permissions, the system displays an access denied message.   
5. If no changes are made to the inventory, the system displays a message indicating that no updates were performed.  
  
Use Case Name: View Product Details   
Use Case ID: UC-06   
Actors: Customer, System   
  
Preconditions:   
- The customer is logged in or can access the product listing as a guest.   
- The system contains at least one product with details available.   
- The product ID or name is provided to the system.   
  
Postconditions:   
- The product details are displayed to the customer.   
- The customer can view the product's name, description, price, images, and inventory status.   
- The customer can decide to add the product to their cart or continue browsing.   
  
Main Flow:   
1. The customer navigates to the product listing or search results.   
2. The customer clicks on a specific product to view its details.   
3. The system retrieves the product information from the database based on the product ID.   
4. The system displays the product details, including name, description, price, category, and available inventory.   
5. The customer reviews the product details and decides whether to add it to their cart or return to the product listing.   
  
Alternative Flow:   
1. If the product is not found, the system displays an error message and suggests checking the product ID or browsing other products.   
2. If the system cannot retrieve the product details due to a database error, it displays an error message and prompts the customer to try again later.   
3. If the product is out of stock, the system displays a message indicating the product is unavailable and may suggest similar products.   
4. If the customer is not logged in, the system allows viewing the product details but restricts actions like adding to a saved cart or wishlist unless the customer logs in or creates an account.   
5. If the product details are incomplete or missing, the system displays a message and allows the administrator to update the product information.  
  
Use Case Name: Add to Cart   
Use Case ID: UC-07   
Actors: Customer, System   
  
Preconditions:   
- The customer is logged in or is browsing as a guest.   
- The product is available in the inventory.   
- The product details are correctly displayed on the product page.   
- The system is accessible and operational.   
  
Postconditions:   
- The selected product is added to the customer's shopping cart.   
- The cart is updated with the product's quantity and total cost.   
- A confirmation message is displayed to the customer.   
  
Main Flow:   
1. The customer browses the product listing or searches for a product.   
2. The customer clicks on the product to view its details.   
3. The customer selects the desired quantity of the product.   
4. The customer clicks the "Add to Cart" button.   
5. The system checks the product's availability and validates the quantity.   
6. The system adds the product and quantity to the customer's cart.   
7. The system updates the cart display with the new item and total cost.   
8. A confirmation message is displayed to the customer that the product was added successfully.   
  
Alternative Flow:   
1. If the product is out of stock, the system displays an error message and does not add it to the cart.   
2. If the selected quantity exceeds the available inventory, the system displays an error message and prompts the customer to adjust the quantity.   
3. If the system fails to add the product due to a technical error, it displays an error message and allows the customer to retry the action.   
4. If the customer is a guest, the system stores the cart in a temporary session or cookie and allows them to continue shopping.   
5. If the customer already has the same product in their cart, the system increases the quantity instead of adding a new item and updates the total cost accordingly.  
  
Use Case Name: Modify Cart Item   
Use Case ID: UC-08   
Actors: Customer, System   
  
Preconditions:   
- The customer is logged in or is browsing as a guest with an active cart session.   
- The customer has at least one item in their shopping cart.   
- The system is accessible and operational.   
  
Postconditions:   
- The selected cart item is updated with the new quantity or removed from the cart.   
- The total cost in the cart is recalculated based on the updated items.   
- A confirmation message is displayed to the customer.   
  
Main Flow:   
1. The customer navigates to their shopping cart.   
2. The customer selects the item they wish to modify.   
3. The customer updates the quantity of the item or removes it from the cart.   
4. The system validates the new quantity (e.g., ensuring it is within available inventory).   
5. The system updates the cart with the modified item or removes it.   
6. The system recalculates the total cost based on the updated cart contents.   
7. A confirmation message is displayed to the customer indicating the cart has been modified.   
  
Alternative Flow:   
1. If the new quantity exceeds the available inventory, the system displays an error message and prompts the customer to adjust the quantity.   
2. If the new quantity is less than or equal to zero, the system automatically removes the item from the cart and updates the total cost.   
3. If the system fails to update the cart due to a technical error, it displays an error message and allows the customer to retry the action.   
4. If the customer tries to modify an item that no longer exists in the system, the system removes the item from the cart and displays a message.   
5. If the customer is a guest and the session expires, the system may lose the cart data unless it is stored in a persistent cookie or the customer logs in to save the cart.  
  
Use Case Name: Checkout Order   
Use Case ID: UC-09   
Actors: Customer, System, Payment Gateway   
  
Preconditions:   
- The customer is logged in to their account.   
- The customer has at least one item in their shopping cart.   
- The system is accessible and operational.   
- The payment gateway is connected and functional.   
  
Postconditions:   
- The order is successfully placed in the system.   
- The payment is processed through the payment gateway.   
- The cart is cleared after successful checkout.   
- The customer receives a confirmation message and an order confirmation email.   
- The inventory is updated to reflect the items purchased.   
  
Main Flow:   
1. The customer navigates to their shopping cart.   
2. The customer reviews the items in the cart and proceeds to checkout.   
3. The system prompts the customer to enter or confirm their shipping address.   
4. The customer selects a shipping method.   
5. The system displays the total cost, including item prices, shipping fees, and taxes.   
6. The customer selects a payment method and enters the necessary payment details.   
7. The system processes the payment through the payment gateway.   
8. The system confirms the payment was successful.   
9. The system creates a new order record in the database with the customer’s information, selected items, and payment details.   
10. The system sends an order confirmation email to the customer.   
11. The system clears the customer’s shopping cart.   
12. The customer is redirected to an order confirmation page with a success message.   
  
Alternative Flow:   
1. If the customer does not have a valid shipping address, the system prompts them to enter or update their address.   
2. If no shipping method is selected, the system displays an error message and requires the customer to choose a method.   
3. If the payment gateway returns an error (e.g., declined payment), the system displays an error message and allows the customer to try a different payment method.   
4. If the system fails to process the payment, it displays an error message and prompts the customer to check their information or try again later.   
5. If the inventory is insufficient to fulfill the order after payment, the system cancels the order, refunds the payment, and displays a message to the customer.   
6. If the customer chooses to cancel the checkout process, the system returns them to the cart page without placing the order.  
  
Use Case Name: Confirm Order   
Use Case ID: UC-10   
Actors: Customer, System, Payment Gateway, Administrator   
  
Preconditions:   
- The customer has completed the checkout process and submitted the order.   
- The payment has been processed and confirmed by the payment gateway.   
- The system is accessible and operational.   
- The order details are temporarily stored in the system.   
  
Postconditions:   
- The order is officially confirmed and marked as "Confirmed" in the system.   
- The customer receives a confirmation message and email.   
- The inventory is updated to reflect the reduction in stock for the ordered items.   
- The order details are saved in the system's database.   
- The administrator is notified of a new confirmed order.   
  
Main Flow:   
1. The system receives a successful payment confirmation from the payment gateway.   
2. The system updates the order status from "Pending" to "Confirmed".   
3. The system reduces the inventory quantities for the items included in the order.   
4. The system records the confirmed order in the database and associates it with the customer’s account.   
5. The system sends an order confirmation email to the customer with order details and expected delivery information.   
6. The system displays a confirmation message on the customer's screen, indicating the order has been successfully confirmed.   
7. The administrator receives a notification of the new confirmed order in the system dashboard.   
  
Alternative Flow:   
1. If the payment gateway fails to confirm the payment, the system displays an error and keeps the order in a "Pending" status until the issue is resolved.   
2. If the inventory update fails due to insufficient stock or database error, the system rolls back the order confirmation and displays an error message to the customer.   
3. If the system fails to send the confirmation email, it logs the error and provides the customer with an option to manually request the email.   
4. If the customer closes the browser before confirmation is completed, the system retains the order status as "Pending" and allows the customer to resume the process if they return.   
5. If the administrator manually cancels the order before confirmation, the system updates the status to "Cancelled" and refunds the payment if applicable.  
  
Use Case Name: Send Order Confirmation Email   
Use Case ID: UC-11   
Actors: System, Customer, Administrator   
  
Preconditions:   
- The customer has completed the checkout process and the order has been confirmed.   
- The customer’s email address is valid and verified.   
- The system is accessible and operational.   
- The payment gateway has confirmed the payment.   
- The order details are stored in the system's database.   
  
Postconditions:   
- The system successfully sends an order confirmation email to the customer.   
- The customer receives the email with order details and delivery information.   
- The order status is updated to "Email Sent" in the system.   
- The administrator can view the order in the system with confirmation details.   
  
Main Flow:   
1. The system triggers the email sending process after the order is confirmed.   
2. The system retrieves the customer’s email address and order details from the database.   
3. The system generates the order confirmation email with relevant content, including order summary, total cost, shipping address, and estimated delivery time.   
4. The system sends the email through the configured email service.   
5. The system updates the order status to "Email Sent" and logs the action.   
6. The customer receives the confirmation email and can review the order details.   
7. The administrator is notified of the sent confirmation email in the system dashboard.   
  
Alternative Flow:   
1. If the customer’s email address is invalid, the system logs the error and displays a message to the customer indicating the email could not be sent.   
2. If the email service is unavailable or returns an error, the system retries the email sending after a delay or allows the administrator to manually resend the email.   
3. If the system fails to update the order status, it logs the error and displays a message to the administrator for review.   
4. If the customer does not receive the email, they can contact support or use the "Resend Confirmation Email" feature in their account.   
5. If the order confirmation email is sent successfully but the customer does not open it, the system may send a reminder email after a specified period.  
  
Use Case Name: View Purchase History   
Use Case ID: UC-12   
Actors: Customer, System   
  
Preconditions:   
- The customer is logged in to their account.   
- The system contains at least one order associated with the customer’s account.   
- The system is accessible and operational.   
  
Postconditions:   
- The customer can view their purchase history with details of previous orders.   
- The system displays each order with its status, total cost, and items purchased.   
- The customer can select an order to view more detailed information.   
  
Main Flow:   
1. The customer navigates to their account dashboard.   
2. The customer selects the "Purchase History" option.   
3. The system retrieves the customer’s order history from the database.   
4. The system displays a list of past orders, including order ID, date, total cost, and order status.   
5. The customer selects a specific order to view its details.   
6. The system shows the order details, such as product names, quantities, prices, and shipping information.   
  
Alternative Flow:   
1. If the customer has no previous orders, the system displays a message indicating that there is no purchase history available.   
2. If the system fails to retrieve order history due to a database error, it displays an error message and prompts the customer to try again later.   
3. If the customer selects an order that has been deleted or is invalid, the system displays a message that the order could not be found.   
4. If the system cannot load product details for an order, it displays a placeholder or message and logs the issue for administrator review.   
5. If the customer tries to access the purchase history without being logged in, the system redirects them to the login page and displays an access denied message.  
  
Use Case Name: Update Personal Information   
Use Case ID: UC-13   
Actors: Customer, System   
  
Preconditions:   
- The customer is logged in to their account.   
- The system is accessible and operational.   
- The customer has existing personal information stored in the system.   
  
Postconditions:   
- The customer’s personal information is successfully updated in the system.   
- The updated information is saved in the database.   
- A confirmation message is displayed to the customer.   
  
Main Flow:   
1. The customer navigates to the "Account Settings" or "Personal Information" section.   
2. The system displays the current personal information (e.g., name, phone number, address).   
3. The customer modifies the desired fields with new information.   
4. The customer clicks the "Save Changes" button.   
5. The system validates the input data (e.g., correct phone number format, valid address).   
6. The system updates the customer’s personal information in the database.   
7. The system displays a success message confirming the information was updated.   
  
Alternative Flow:   
1. If the system detects invalid input (e.g., incorrect phone number format), it displays an error message and prompts the customer to correct the information.   
2. If the system fails to update the information due to a database error, it displays an error message and allows the customer to retry the update.   
3. If the customer does not make any changes, the system displays a message indicating that no updates were performed.   
4. If the customer cancels the update process, the system reverts to the previous information and displays a message confirming the cancellation.   
5. If the customer tries to update another user’s information, the system denies the request and displays an access denied message.  
  
Use Case Name: Admin Login   
Use Case ID: UC-14   
Actors: Administrator, System   
  
Preconditions:   
- The administrator is not currently logged in to their account.   
- The system is accessible and operational.   
- The administrator has a valid account with a registered email and password.   
  
Postconditions:   
- The administrator is successfully authenticated and logged in.   
- The administrator's session is initiated.   
- The administrator is redirected to the admin dashboard.   
  
Main Flow:   
1. The administrator navigates to the admin login page.   
2. The administrator enters their registered email address and password.   
3. The system validates the email and password against the database.   
4. The system confirms that the account is active and the email has been verified.   
5. The system creates a session for the administrator and stores relevant session data.   
6. The administrator is redirected to the admin dashboard.   
7. A success message is displayed indicating they are logged in.   
  
Alternative Flow:   
1. If the email is not registered, the system displays an error message and prompts the administrator to check their email or contact support.   
2. If the password is incorrect, the system displays an error message and allows the administrator to retry entering the password.   
3. If the account is inactive or the email is unconfirmed, the system displays a message and provides an option to resend the confirmation email.   
4. If the system detects multiple failed login attempts, it temporarily locks the account and notifies the administrator via email.   
5. If the administrator forgets their password, they can click the "Forgot Password" link to initiate a password reset process.  
  
Use Case Name: Manage Categories   
Use Case ID: UC-15   
Actors: Administrator, System   
  
Preconditions:   
- The administrator is logged in to their account.   
- The administrator has the necessary permissions to manage categories.   
- The system contains at least one category or is ready for category creation.   
- The system is accessible and operational.   
  
Postconditions:   
- The category list is updated based on the administrator's actions (create, update, or delete).   
- The system reflects the current state of categories in the product listing and category navigation.   
- A confirmation message is displayed to the administrator indicating the success or failure of the operation.   
  
Main Flow:   
1. The administrator navigates to the "Category Management" section in the system.   
2. The system displays a list of existing categories and provides options to create, edit, or delete categories.   
3. The administrator selects one of the following actions:   
 - Create a new category.   
 - Edit an existing category.   
 - Delete a category.   
4. If creating a new category, the system displays a form for the category name and description.   
5. If editing an existing category, the system pre-fills the form with the current details.   
6. The administrator fills in the category information and submits the form.   
7. The system validates the input data (e.g., non-empty name, no duplicates for creation).   
8. The system updates the category information in the database.   
9. The system displays a success message to the administrator.   
10. The updated category list is reflected in the system, affecting product listings and navigation.   
  
Alternative Flow:   
1. If the category name is empty or invalid, the system displays an error message and prompts the administrator to enter a valid name.   
2. If the administrator attempts to create a category with a name that already exists, the system displays an error message and suggests using a different name.   
3. If the administrator tries to delete a category that is currently associated with products in the inventory, the system displays a warning and prevents deletion until the association is resolved.   
4. If the system fails to update the category due to a database error, it displays an error message and allows the administrator to retry the operation.   
5. If the administrator cancels the action before submission, the system reverts to the previous category list and displays a cancellation message.  
  
Use Case Name: Update Product Details   
Use Case ID: UC-06   
Actors: Administrator, System   
  
Preconditions:   
- The administrator is logged in to their account.   
- The administrator has the necessary permissions to update product details.   
- The system contains at least one product with existing details.   
- The product ID is provided to the system.   
- The system is accessible and operational.   
  
Postconditions:   
- The product's details are successfully updated in the system.   
- The updated product information is saved in the database.   
- The product listing and inventory reflect the updated details.   
- A confirmation message is displayed to the administrator.   
  
Main Flow:   
1. The administrator navigates to the product management section.   
2. The administrator selects the product they wish to update.   
3. The system displays the current product details in an editable form.   
4. The administrator modifies the product information (e.g., name, description, price, category).   
5. The administrator uploads new product images or removes existing ones if applicable.   
6. The administrator clicks the "Update Product" button.   
7. The system validates the input data, including checking for valid price format and duplicate product names.   
8. The system updates the product details in the database.   
9. The system displays a success message confirming the product was updated.   
10. The updated product is reflected in the product listing and inventory.   
  
Alternative Flow:   
1. If the product name is already taken, the system displays an error message and prompts the administrator to enter a different name.   
2. If the price format is invalid, the system displays an error message and prompts the administrator to re-enter a valid price.   
3. If required fields are left empty, the system displays an error message and prompts the administrator to complete all mandatory fields.   
4. If the file upload fails (e.g., unsupported format, too large), the system displays an error message and allows the administrator to re-upload the file.   
5. If the system fails to update the product due to technical issues, it displays an error message and prompts the administrator to try again or contact support.  
  
Use Case Name: Install Plugin   
Use Case ID: UC-16   
Actors: Administrator, System, Plugin API   
  
Preconditions:   
- The administrator is logged in to their account.   
- The administrator has the necessary permissions to install plugins.   
- The system is accessible and operational.   
- The Plugin API is available and functional.   
  
Postconditions:   
- The plugin is successfully installed and activated in the system.   
- The system reflects the new functionality provided by the plugin.   
- A confirmation message is displayed to the administrator.   
- The plugin is registered in the system’s plugin list.   
  
Main Flow:   
1. The administrator navigates to the "Plugin Management" section.   
2. The administrator selects the "Install Plugin" option.   
3. The system displays a list of available plugins from the Plugin API.   
4. The administrator selects a plugin to install.   
5. The system fetches the plugin metadata and details from the Plugin API.   
6. The system validates the plugin compatibility with the current system version.   
7. The system downloads and installs the plugin.   
8. The system activates the plugin and updates the plugin list.   
9. A success message is displayed to the administrator confirming the plugin installation.   
  
Alternative Flow:   
1. If the Plugin API is unreachable, the system displays an error message and suggests checking the connection or trying again later.   
2. If the plugin is not compatible with the current system version, the system displays a warning and prevents installation.   
3. If the plugin has already been installed, the system displays a message and provides an option to update it instead.   
4. If the system fails to download or install the plugin due to technical issues, it displays an error message and allows the administrator to retry the installation.   
5. If the administrator cancels the installation before completion, the system reverts to the plugin list and displays a cancellation message.  
  
Use Case Name: Access Plugin API Documentation   
Use Case ID: UC-17   
Actors: Administrator, System, Plugin API   
  
Preconditions:   
- The administrator is logged in to their account.   
- The administrator has the necessary permissions to access the Plugin API Documentation.   
- The Plugin API is accessible and operational.   
- The system is connected to the Plugin API.   
  
Postconditions:   
- The administrator can view the Plugin API documentation.   
- The documentation includes endpoints, request/response formats, and usage examples.   
- The administrator can understand how to integrate or configure the plugin.   
  
Main Flow:   
1. The administrator navigates to the "Plugin Management" section.   
2. The administrator selects the "Access API Documentation" option for a specific plugin.   
3. The system retrieves the documentation from the Plugin API.   
4. The system displays the documentation in a user-friendly format (e.g., web page or modal window).   
5. The administrator reviews the documentation and understands the plugin's functionality and integration process.   
  
Alternative Flow:   
1. If the Plugin API is unreachable, the system displays an error message and suggests checking the connection or trying again later.   
2. If the selected plugin does not have documentation available, the system displays a message indicating that no documentation is provided.   
3. If the system fails to load the documentation, it displays an error message and allows the administrator to refresh or contact support.   
4. If the administrator does not have permission to access the documentation, the system displays an access denied message.   
5. If the documentation is outdated or incompatible with the current system version, the system displays a warning to the administrator.  
  
Use Case Name: View Order Status   
Use Case ID: UC-10   
Actors: Customer, System   
Preconditions:   
- The customer is logged in to their account.   
- The system contains at least one order associated with the customer’s account.   
- The system is accessible and operational.   
  
Postconditions:   
- The customer is shown the current status of their order (e.g., confirmed, shipped, delivered).   
- The system displays relevant order information such as order ID, date, total cost, and items.   
- The customer can take further actions based on the order status.   
  
Main Flow:   
1. The customer navigates to the "Order Status" section in their account dashboard.   
2. The system displays a list of the customer’s orders with order IDs and statuses.   
3. The customer selects a specific order to view its status in detail.   
4. The system retrieves the order information from the database, including status, items, and shipping details.   
5. The system displays the order status and relevant details to the customer.   
6. The system provides an option to contact support or track the order further if applicable.   
  
Alternative Flow:   
1. If the customer has no orders, the system displays a message indicating no order status is available.   
2. If the system fails to retrieve the order status due to a database error, it displays an error message and prompts the customer to try again later.   
3. If the selected order is invalid or has been deleted, the system displays a message that the order could not be found.   
4. If the order status changes during the viewing process, the system updates the display to reflect the latest status.   
5. If the customer is not logged in, the system redirects them to the login page and displays an access denied message.  
  
Use Case Name: Calculate Total Cost   
Use Case ID: UC-08   
Actors: Customer, System   
  
Preconditions:   
- The customer has added at least one item to their shopping cart.   
- The system is accessible and operational.   
- Product prices and inventory quantities are correctly stored in the database.   
- The cart contains valid items with quantities not exceeding available inventory.   
  
Postconditions:   
- The total cost of the cart is calculated and displayed to the customer.   
- The total cost includes the price of all items multiplied by their quantities.   
- The system reflects any applicable discounts or taxes.   
- A confirmation message is displayed if the calculation is successful.   
  
Main Flow:   
1. The customer navigates to their shopping cart.   
2. The system retrieves the items in the cart along with their respective quantities and prices.   
3. The system calculates the total cost by summing the product of each item's price and quantity.   
4. The system applies any applicable discounts or taxes to the total cost.   
5. The system displays the updated total cost to the customer.   
6. The system provides an option to proceed to checkout with the calculated total.   
  
Alternative Flow:   
1. If an item's price is not available, the system displays an error message and prevents the total cost from being calculated until the issue is resolved.   
2. If the system fails to retrieve cart items due to a database error, it displays an error message and prompts the customer to try again later.   
3. If no applicable discounts or taxes are configured, the system displays a message indicating that the total cost is the sum of the items’ prices.   
4. If the cart is empty, the system displays a message indicating that no items are selected for purchase.   
5. If the customer modifies the cart (e.g., adds or removes items), the system automatically recalculates and updates the total cost.  
  
Use Case Name: Manage User Accounts   
Use Case ID: UC-18   
Actors: Administrator, System, Customer   
  
Preconditions:   
- The administrator is logged in to their account.   
- The administrator has the necessary permissions to manage user accounts.   
- The system is accessible and operational.   
- The system contains at least one registered user account.   
  
Postconditions:   
- The administrator can view, update, or delete user accounts.   
- The system reflects the updated or deleted user account status.   
- A confirmation message is displayed for each action performed.   
- The affected user is notified (if applicable) of any changes to their account.   
  
Main Flow:   
1. The administrator navigates to the "User Management" section in the system.   
2. The system displays a list of registered user accounts with options to view, edit, or delete each account.   
3. The administrator selects a specific user account to manage.   
4. The administrator chooses to either view details, update the account, or delete the account.   
5. If updating, the system displays an editable form with the user's current information.   
6. The administrator modifies the necessary fields (e.g., status, role, email).   
7. The administrator submits the update or deletion request.   
8. The system validates the input data and confirms the action (e.g., ensures no invalid email format).   
9. The system updates or deletes the user account in the database.   
10. The system displays a confirmation message to the administrator indicating the action was successful.   
11. If the account was deleted, the system removes it from the user list and logs the action.   
12. If the account was updated, the system reflects the changes in the user list and the affected user is notified via email if needed.   
  
Alternative Flow:   
1. If the selected user account does not exist, the system displays an error message and prompts the administrator to enter a valid user ID or email.   
2. If the system fails to retrieve the user account data due to a database error, it displays an error message and allows the administrator to retry the operation.   
3. If the administrator attempts to delete their own account, the system displays a warning and prevents the action.   
4. If the system fails to update the user account due to a technical error, it displays an error message and allows the administrator to retry the action.   
5. If the administrator cancels the operation before submission, the system reverts to the previous state and displays a cancellation message.  
  
Use Case Name: Manage Payment Method   
Use Case ID: UC-19   
Actors: Customer, System   
  
Preconditions:   
- The customer is logged in to their account.   
- The system is accessible and operational.   
- The customer has at least one payment method stored in the system (if applicable).   
  
Postconditions:   
- The customer can view, add, update, or delete their saved payment methods.   
- The system reflects the current list of payment methods associated with the customer's account.   
- A confirmation message is displayed to the customer for each action performed.   
  
Main Flow:   
1. The customer navigates to the "Payment Methods" section in their account settings.   
2. The system displays the list of saved payment methods (if any) with options to edit or delete each one.   
3. The customer selects one of the following actions:   
 - Add a new payment method.   
 - Edit an existing payment method.   
 - Delete a payment method.   
4. If adding a new payment method, the system displays a form for entering payment details (e.g., card number, expiration date, CVV, billing address).   
5. If editing an existing payment method, the system pre-fills the form with the selected payment method's information.   
6. The customer enters or modifies the payment method details.   
7. The customer submits the form.   
8. The system validates the payment method information (e.g., checks for valid card details, format of expiration date, and ensures billing address is complete if required).   
9. The system saves or updates the payment method in the database.   
10. The system displays a success message confirming the action was completed.   
11. The updated list of payment methods is shown to the customer.   
  
Alternative Flow:   
1. If the customer enters invalid payment details (e.g., incorrect card number or expired date), the system displays an error message and prompts the customer to correct the information.   
2. If the system fails to save or update the payment method due to a database error, it displays an error message and allows the customer to retry the operation.   
3. If the customer attempts to delete their last saved payment method, the system displays a warning and allows them to confirm or cancel the action.   
4. If the customer cancels the operation before submission, the system reverts to the previous state and displays a cancellation message.   
5. If the customer tries to access the payment method section without being logged in, the system redirects them to the login page and displays an access denied message.