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 personal information and receiving a confirmation email. The system must validate the email format, check for uniqueness, and ensure password strength.   
\*\*Input\*\*:   
- Name (First and Last)   
- Email   
- Password   
- Shipping Address   
- Preferred Category ID (optional)   
\*\*Output\*\*:   
- Created Customer entity in the database   
- Email confirmation sent to the customer   
- Updated inventory/category associations if applicable   
  
---  
  
## 1.2 Customer Login Function   
\*\*Function ID\*\*: FR-02   
\*\*Description\*\*: Registered and verified customers can log in using their email and password. The system must validate the credentials and initiate a session.   
\*\*Input\*\*:   
- Email   
- Password   
\*\*Output\*\*:   
- Active session for the customer   
- Display of the customer's homepage or dashboard   
- Error messages if login fails   
  
---  
  
## 1.3 View Product Details Function   
\*\*Function ID\*\*: FR-03   
\*\*Description\*\*: Customers can view detailed information of a selected product, such as name, description, price, and availability. The system must update the browsing history and display the data correctly.   
\*\*Input\*\*:   
- Product ID   
- Customer ID (optional for browsing history)   
\*\*Output\*\*:   
- Displayed product details (name, description, price, availability, image)   
- Updated customer activity log   
  
---  
  
## 1.4 Add Product to Cart Function   
\*\*Function ID\*\*: FR-04   
\*\*Description\*\*: A logged-in customer can add a product to their shopping cart. The system must check the product’s availability and update the cart accordingly.   
\*\*Input\*\*:   
- Product ID   
- Quantity   
- Customer ID   
\*\*Output\*\*:   
- Updated CartItem in the database   
- Updated cart display with new item and total   
- Error message if product is out of stock or unavailable   
  
---  
  
## 1.5 Update Shopping Cart Function   
\*\*Function ID\*\*: FR-05   
\*\*Description\*\*: A customer can modify the contents of their shopping cart, such as changing the quantity or removing items. The system must validate stock availability and update the cart and total accordingly.   
\*\*Input\*\*:   
- CartItem ID(s)   
- New quantity or removal request   
- Customer ID   
\*\*Output\*\*:   
- Updated CartItem or removed CartItem from the database   
- Updated cart display with new contents and total   
- Error message if stock is insufficient or system fails to update   
  
---  
  
## 1.6 Remove Product from Cart Function   
\*\*Function ID\*\*: FR-06   
\*\*Description\*\*: A customer can remove a selected product from their shopping cart. The system must update the cart and recalculate the total amount.   
\*\*Input\*\*:   
- CartItem ID   
- Customer ID   
\*\*Output\*\*:   
- Removed CartItem from the database   
- Updated cart display and total   
- Notification if cart is empty   
  
---  
  
## 1.7 Checkout Function   
\*\*Function ID\*\*: FR-07   
\*\*Description\*\*: A customer can proceed to checkout and complete the purchase process. The system must calculate the total amount, process payment, create an order, and send a confirmation email.   
\*\*Input\*\*:   
- CartItem(s)   
- Shipping address   
- Shipping method   
- Payment method and details   
- Customer ID   
\*\*Output\*\*:   
- Created Order and OrderDetail entities in the database   
- Updated Inventory with reduced stock   
- Sent email confirmation to the customer   
- Cleared shopping cart   
  
---  
  
## 1.8 Confirm Order Function   
\*\*Function ID\*\*: FR-08   
\*\*Description\*\*: A customer can confirm an order after providing all necessary checkout information. The system must process the payment, finalize the order, and update inventory accordingly.   
\*\*Input\*\*:   
- Order summary (items, shipping address, payment method)   
- Customer ID   
\*\*Output\*\*:   
- Confirmed Order entity   
- Updated Payment status and details   
- Updated Inventory   
- Sent Email Confirmation   
  
---  
  
## 1.9 Receive Order Confirmation Email Function   
\*\*Function ID\*\*: FR-09   
\*\*Description\*\*: The system must send an email confirmation to the customer after an order is successfully processed. The email must include the order summary, payment details, and shipping information.   
\*\*Input\*\*:   
- Order ID   
- Customer Email   
\*\*Output\*\*:   
- Sent confirmation email to the customer   
- Log entry indicating successful email delivery   
- Order status updated to "confirmed"   
  
---  
  
## 1.10 Administrator Login Function   
\*\*Function ID\*\*: FR-10   
\*\*Description\*\*: Administrators can log in using their username and password. The system must validate credentials and initiate a session.   
\*\*Input\*\*:   
- Username   
- Password   
\*\*Output\*\*:   
- Active session for the administrator   
- Display of the admin dashboard or control panel   
- Error messages if login fails   
  
---  
  
## 1.11 Manage Inventory Function   
\*\*Function ID\*\*: FR-11   
\*\*Description\*\*: Administrators can add, update, or delete inventory records. The system must validate the changes and update associated product data and logs.   
\*\*Input\*\*:   
- Product ID   
- Inventory quantity   
- Inventory status (e.g., reorder level)   
- Administrator ID   
\*\*Output\*\*:   
- Updated Inventory entity in the database   
- Updated product availability status   
- Log entry for the inventory change   
- Confirmation message for the administrator   
  
---  
  
## 1.12 Update Product Information Function   
\*\*Function ID\*\*: FR-12   
\*\*Description\*\*: Administrators can update product details such as name, price, description, and category. The system must validate the changes and update related entities.   
\*\*Input\*\*:   
- Product ID   
- Updated fields (e.g., Name, Price, Description, CategoryID)   
- Administrator ID   
\*\*Output\*\*:   
- Updated Product entity in the database   
- Updated Inventory and Category associations   
- Log entry for the product update   
- Confirmation message for the administrator   
  
---  
  
## 1.13 Manage Categories Function   
\*\*Function ID\*\*: FR-13   
\*\*Description\*\*: Administrators can add, update, or delete product categories. The system must ensure consistency with associated products and handle plugin restrictions if applicable.   
\*\*Input\*\*:   
- Category ID (for update or delete)   
- New category name and description (for add)   
- Administrator ID   
\*\*Output\*\*:   
- Updated Category entity in the database   
- Updated product listings and search filters   
- Log entry for the category change   
- Confirmation message for the administrator   
  
---  
  
## 1.14 Create Category Function   
\*\*Function ID\*\*: FR-14   
\*\*Description\*\*: Administrators can create a new category for organizing products. The system must validate the category name and description, and apply any plugin rules.   
\*\*Input\*\*:   
- Category name   
- Category description   
- Administrator ID   
\*\*Output\*\*:   
- Created Category entity in the database   
- Updated category list in the system   
- Log entry for the category creation   
- Confirmation message for the administrator   
  
---  
  
## 1.15 Delete Category Function   
\*\*Function ID\*\*: FR-15   
\*\*Description\*\*: Administrators can delete an existing category. If the category contains products, the system must prompt for reassignment or uncategorization.   
\*\*Input\*\*:   
- Category ID   
- Administrator ID   
- Optional: New Category ID for reassignment   
\*\*Output\*\*:   
- Deleted Category entity   
- Updated product records to reflect new category or uncategorized status   
- Log entry for the category deletion   
- Confirmation message for the administrator   
  
---  
  
## 1.16 Plugin Installation Function   
\*\*Function ID\*\*: FR-16   
\*\*Description\*\*: Administrators can install new plugins to extend the system's functionality. The system must verify plugin compatibility and dependencies before installation.   
\*\*Input\*\*:   
- Plugin name and version   
- Plugin file or source (e.g., ZIP or URL)   
- Administrator ID   
\*\*Output\*\*:   
- Installed Plugin entity in the database   
- Updated plugin list in the system   
- Log entry for the plugin installation   
- Confirmation message for the administrator   
  
---  
  
## 1.17 Plugin Configuration Function   
\*\*Function ID\*\*: FR-17   
\*\*Description\*\*: Administrators can configure installed plugins to suit their needs. The system must validate configuration inputs and apply them to the plugin.   
\*\*Input\*\*:   
- Plugin ID   
- Configuration parameters (e.g., API keys, thresholds, settings)   
- Administrator ID   
\*\*Output\*\*:   
- Updated Plugin Configuration entity in the database   
- Applied configuration settings to the plugin   
- Log entry for the plugin configuration   
- Confirmation message for the administrator   
  
---  
  
## 1.18 View Payment Information Function   
\*\*Function ID\*\*: FR-18   
\*\*Description\*\*: Customers can view their payment information for past orders. The system must retrieve and display the payment method, status, amount, and date.   
\*\*Input\*\*:   
- Order ID   
- Customer ID   
\*\*Output\*\*:   
- Displayed Payment details (method, amount, status, date)   
- Option to request a payment confirmation email   
- Error message if payment data is missing or invalid   
  
---  
  
## 1.19 Update Order Detail Function   
\*\*Function ID\*\*: FR-19   
\*\*Description\*\*: Administrators can update the details of an order, such as product quantity or price. The system must validate the changes, adjust inventory if necessary, and log the update.   
\*\*Input\*\*:   
- OrderDetail ID   
- Updated product quantity or price   
- Administrator ID   
\*\*Output\*\*:   
- Updated OrderDetail entity in the database   
- Adjusted Inventory if product quantity is changed   
- Log entry for the order detail update   
- Confirmation message for the administrator   
  
---  
  
Each function is derived directly from the provided use cases and data model, ensuring feasibility, consistency, and completeness. All inputs and outputs are defined within the system context, and no additional or invented functions are included.

# External Description

# 2. External Interfaces  
  
This chapter describes the external interfaces of the system, including user interfaces, hardware interfaces, software interfaces, and communication interfaces. These interfaces define how the system interacts with users, hardware, other software systems, and external communication channels. The descriptions are based on the functional requirements and ensure that all external data sources and interactions are clearly defined.  
  
## 2.1 User Interface  
  
The system interacts with its users through a graphical user interface (GUI) that is accessible via a web browser or mobile application. The following are the key user interface components and their roles:  
  
- \*\*Customer Registration Screen\*\*:   
 - \*\*Role\*\*: Allows new customers to provide personal information and create an account.   
 - \*\*Description\*\*: Includes input fields for name, email, password, shipping address, and optional preferred category. The system validates the email format, password strength, and checks for email uniqueness.   
 - \*\*Interaction Method\*\*: Form submission with validation feedback and confirmation message displayed after successful registration.  
  
- \*\*Customer Login Screen\*\*:   
 - \*\*Role\*\*: Enables registered and verified customers to log in using their email and password.   
 - \*\*Description\*\*: Contains input fields for email and password, and a login button. The system validates the credentials and initiates a session upon successful login.   
 - \*\*Interaction Method\*\*: Form submission with error messages displayed for invalid or incorrect credentials.  
  
- \*\*Product Details Page\*\*:   
 - \*\*Role\*\*: Displays detailed information about a product, such as name, description, price, availability, and image.   
 - \*\*Description\*\*: The page dynamically loads product data based on the provided Product ID. It also optionally updates the customer’s browsing history if a Customer ID is provided.   
 - \*\*Interaction Method\*\*: Product ID is passed through the URL or a query parameter. The system updates the customer’s activity log upon viewing.  
  
- \*\*Shopping Cart Page\*\*:   
 - \*\*Role\*\*: Displays the current items in a customer’s shopping cart and allows modification.   
 - \*\*Description\*\*: The page includes functionality to add, update, and remove items. The system validates stock availability and updates the cart total accordingly.   
 - \*\*Interaction Method\*\*: Customer ID is used to fetch cart data. CartItem IDs, new quantities, and removal requests are submitted via form or API calls.  
  
- \*\*Checkout Page\*\*:   
 - \*\*Role\*\*: Allows a customer to complete the purchase process by entering shipping and payment information.   
 - \*\*Description\*\*: The page includes fields for shipping address, shipping method, and payment method. The system calculates the total amount, processes payment, and creates an order upon submission.   
 - \*\*Interaction Method\*\*: CartItem(s), shipping and payment details, and Customer ID are submitted to finalize the order.  
  
- \*\*Order Confirmation Page\*\*:   
 - \*\*Role\*\*: Displays the final order summary and confirms the order after payment is processed.   
 - \*\*Description\*\*: Shows the items, total amount, shipping address, and payment method. The system updates the order status and clears the shopping cart after confirmation.   
 - \*\*Interaction Method\*\*: Order summary and Customer ID are used to confirm the order. A confirmation message is displayed, and a confirmation email is sent.  
  
- \*\*Admin Login Screen\*\*:   
 - \*\*Role\*\*: Enables administrators to log in using a username and password.   
 - \*\*Description\*\*: Contains fields for username and password, and a login button. The system validates the credentials and displays the admin dashboard or control panel.   
 - \*\*Interaction Method\*\*: Form submission with error messages for invalid or incorrect login attempts.  
  
- \*\*Admin Dashboard / Control Panel\*\*:   
 - \*\*Role\*\*: Provides administrators with tools to manage inventory, product information, categories, and plugins.   
 - \*\*Description\*\*: Includes sections for adding/updating/deleting inventory, products, and categories. Also includes a plugin management section for installation and configuration.   
 - \*\*Interaction Method\*\*: Various forms and controls are used to submit changes, which are validated and applied to the database.  
  
- \*\*Payment Information View\*\*:   
 - \*\*Role\*\*: Displays the payment details for a specific order.   
 - \*\*Description\*\*: Shows the payment method, amount, status, and date. Customers can also request a payment confirmation email.   
 - \*\*Interaction Method\*\*: Order ID and Customer ID are used to fetch and display the relevant payment data.  
  
- \*\*Order Detail Update Page (Admin)\*\*:   
 - \*\*Role\*\*: Enables administrators to update the details of an order, such as product quantity or price.   
 - \*\*Description\*\*: Displays the current order details and allows administrators to modify them. The system validates the changes and adjusts inventory if necessary.   
 - \*\*Interaction Method\*\*: OrderDetail ID and Administrator ID are used to fetch and update the data. A confirmation message is displayed after the update.  
  
## 2.2 Hardware Interface  
  
There are no direct hardware interfaces required for the system as described in the functional requirements. The system is designed to be accessed through standard web browsers and mobile applications, and does not require interaction with specific hardware devices. Therefore, no hardware interface specifications are necessary in this section.  
  
## 2.3 Software Interface  
  
The system interacts with external software components and databases to support its functionality. The following are the key software interfaces:  
  
- \*\*Database System (e.g., MySQL, PostgreSQL)\*\*   
 - \*\*Role\*\*: Stores and retrieves data related to customers, products, inventory, categories, orders, and plugins.   
 - \*\*Description\*\*: The system uses a relational database to manage persistent data. The database schema includes entities such as Customer, Product, Inventory, Category, Order, OrderDetail, and Plugin.   
 - \*\*Interaction Method\*\*: The system uses SQL queries or an ORM (Object-Relational Mapping) to interact with the database. Inputs such as Customer ID, Product ID, OrderDetail ID, etc., are used to fetch and update records. Outputs include created or updated entities, such as Customer, Product, Order, etc.  
  
- \*\*Email Service (e.g., SMTP Server, SendGrid, Amazon SES)\*\*   
 - \*\*Role\*\*: Sends confirmation emails to customers after registration or order confirmation.   
 - \*\*Description\*\*: The system requires an external email service to send emails. The email includes the order summary, payment details, and shipping information.   
 - \*\*Interaction Method\*\*: The system uses an API or SMTP protocol to send emails. Inputs include the recipient's email address, the email content, and the order details. Outputs include a log entry for successful email delivery and an updated order status.  
  
- \*\*Payment Gateway (e.g., Stripe, PayPal, Square)\*\*   
 - \*\*Role\*\*: Processes customer payments during the checkout and order confirmation processes.   
 - \*\*Description\*\*: The system integrates with a payment gateway to handle secure payment transactions. The payment gateway verifies the payment method, processes the payment, and returns a status.   
 - \*\*Interaction Method\*\*: The system sends payment details (e.g., amount, payment method, customer ID) to the payment gateway via API. The payment gateway responds with a payment status and transaction details. Outputs include updated payment status and confirmation messages.  
  
- \*\*Plugin Management Module\*\*   
 - \*\*Role\*\*: Allows administrators to install, configure, and manage plugins to extend system functionality.   
 - \*\*Description\*\*: The system provides a plugin interface that includes validation of plugin compatibility and dependencies. Configuration parameters are applied to the plugin and stored in the database.   
 - \*\*Interaction Method\*\*: The system uses a plugin API to install and configure plugins. Inputs include the plugin file, configuration parameters, and Administrator ID. Outputs include the installed plugin entity, updated plugin list, and confirmation messages.  
  
- \*\*Inventory Management Module\*\*   
 - \*\*Role\*\*: Maintains and updates product inventory levels and statuses.   
 - \*\*Description\*\*: The system tracks the availability and quantity of each product. Inventory is updated automatically when a product is added to a cart, updated, or deleted.   
 - \*\*Interaction Method\*\*: The system uses internal functions or triggers to update inventory records. Inputs include Product ID, Inventory quantity, and Inventory status. Outputs include updated Inventory entity and product availability status.  
  
## 2.4 Communication Interface  
  
The system communicates with external entities using standard communication protocols and services. The following are the key communication interfaces:  
  
- \*\*Email Communication\*\*   
 - \*\*Role\*\*: Sends confirmation emails to customers for registration and order processing.   
 - \*\*Description\*\*: The system uses an email service to send messages to customer email addresses. The email contains the order summary, payment details, and shipping information.   
 - \*\*Interaction Method\*\*: The system sends an email via an SMTP server or third-party email API. Inputs include the customer's email address and order details. Outputs include a log entry indicating successful delivery and an updated order status.  
  
- \*\*Web Browsing / API Communication\*\*   
 - \*\*Role\*\*: Supports customer and administrator access to the system through web browsers or API integrations.   
 - \*\*Description\*\*: The system is accessible via HTTP/HTTPS protocols and supports RESTful API for integration with other systems or services.   
 - \*\*Interaction Method\*\*: The system uses HTTP methods (GET, POST, PUT, DELETE) to interact with clients. Inputs are received via API endpoints or web forms. Outputs are returned in JSON or HTML format.  
  
- \*\*Payment Gateway Communication\*\*   
 - \*\*Role\*\*: Processes customer payments securely during checkout.   
 - \*\*Description\*\*: The system communicates with a payment gateway to verify and process transactions.   
 - \*\*Interaction Method\*\*: The system sends payment requests via HTTPS to the payment gateway API. Inputs include the customer's payment method and amount. Outputs include a payment status and transaction confirmation.  
  
- \*\*Plugin Communication Interface\*\*   
 - \*\*Role\*\*: Supports plugin installation and configuration by validating plugin compatibility and applying configuration parameters.   
 - \*\*Description\*\*: The system communicates with plugins to load and execute their functionality.   
 - \*\*Interaction Method\*\*: The system uses a plugin API to install and configure plugins. Inputs include the plugin file and configuration parameters. Outputs include updated plugin status and confirmation messages.  
  
---  
  
This chapter provides a comprehensive overview of the external interfaces required for the system to function correctly. Each interface is described in terms of its role, description, and interaction method to ensure clarity for developers and stakeholders.

# 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 customer is not already registered in the system.   
  
Postconditions:   
1. The customer is successfully registered in the system.   
2. The customer receives an email confirmation of the registration.   
3. The system updates the customer information in the database.   
  
Main Flow:   
1. The customer navigates to the registration page.   
2. The customer enters personal information, including name, email, password, and shipping address.   
3. The customer selects a preferred category for browsing.   
4. The system validates the email format and checks for uniqueness.   
5. The system validates the password strength.   
6. The system creates a new customer profile and stores it in the database.   
7. The system adds the customer to the relevant product category.   
8. The system generates and sends an email confirmation to the customer's inbox.   
9. The customer receives the confirmation email and completes the registration process by clicking the verification link.   
10. The system marks the customer's account as verified.   
  
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 and suggests the customer log in instead.   
3. If the password does not meet the strength requirements, the system displays a warning and allows the customer to re-enter a stronger password.   
4. If the email confirmation is not received or the verification link is not clicked within the allowed time, the system may send a reminder email or require the customer to re-initiate the registration process.  
  
Use Case Name: Customer Login   
Use Case ID: UC-02   
Actors: Customer, System   
Preconditions:   
1. The customer has already registered in the system and received email confirmation.   
2. The customer is not currently logged in.   
3. The system is operational and accessible.   
  
Postconditions:   
1. The customer is successfully logged into the system.   
2. The system displays the customer's dashboard or homepage.   
3. The customer's session is initiated.   
  
Main Flow:   
1. The customer navigates to the login page.   
2. The customer enters their registered email and password.   
3. The system verifies the email and password against the database.   
4. If the credentials are valid, the system logs the customer in.   
5. The system displays the customer's homepage or dashboard.   
  
Alternative Flow:   
1. If the email is not registered, the system displays an error message and prompts the customer to register or check their email.   
2. If the password is incorrect, the system displays an error message and allows the customer to re-enter the password.   
3. If the customer exceeds the maximum number of failed login attempts, the system temporarily locks the account and sends an email notification.   
4. If the customer forgets their password, they can click on the "Forgot Password" link to initiate a password reset process.  
  
Use Case Name: View Product Details   
Use Case ID: UC-03   
Actors: Customer, System   
Preconditions:   
1. The customer is logged into the system.   
2. The system contains at least one product in the inventory.   
3. The customer navigates to the product listing or search page.   
  
Postconditions:   
1. The customer is shown the detailed information of the selected product.   
2. The system updates the customer's browsing history or activity log.   
3. The shopping cart remains unchanged unless the customer adds the product to it.   
  
Main Flow:   
1. The customer browses the product list or searches for a product.   
2. The customer selects a specific product to view details.   
3. The system retrieves the product details, including description, price, availability, and images.   
4. The system displays the product details to the customer.   
5. The customer may choose to add the product to the shopping cart or continue browsing.   
  
Alternative Flow:   
1. If the requested product is not found, the system displays an error message and suggests similar products or a search.   
2. If the product is out of stock, the system notifies the customer and may suggest alternatives or a back-in-stock notification option.   
3. If the system fails to load product images or data, the system displays a loading error and allows the customer to refresh or try another product.   
4. If the customer adds the product to the shopping cart, the system updates the cart and displays a confirmation message.  
  
Use Case Name: Add Product to Cart   
Use Case ID: UC-04   
Actors: Customer, System, Shopping Cart, Product, Inventory   
Preconditions:   
1. The customer is logged into the system.   
2. The customer has viewed the product details.   
3. The product is available in the inventory.   
4. The shopping cart is accessible and initialized for the customer.   
  
Postconditions:   
1. The product is added to the customer's shopping cart.   
2. The system updates the cart display to reflect the new addition.   
3. The inventory remains unchanged unless the cart functionality includes a "reserve" feature.   
  
Main Flow:   
1. The customer views a product's details.   
2. The customer selects the "Add to Cart" option.   
3. The system checks the product's availability in the inventory.   
4. The system adds the product to the customer's shopping cart.   
5. The system confirms the addition and updates the cart display.   
6. The customer is redirected to the cart or receives a confirmation message.   
  
Alternative Flow:   
1. If the product is out of stock, the system displays a message and does not add it to the cart.   
2. If the system cannot add the product due to an error, it displays a message and allows the customer to retry.   
3. If the customer adds multiple quantities of the product, the system checks if the requested amount is available and adjusts accordingly.   
4. If the customer is not logged in, the system may prompt to log in or create an account before adding the product to the cart.  
  
Use Case Name: Update Shopping Cart   
Use Case ID: UC-05   
Actors: Customer, Shopping Cart, Product, Inventory, System   
  
Preconditions:   
1. The customer is logged into the system.   
2. The customer has at least one item in their shopping cart.   
3. The shopping cart is accessible from the homepage or product detail page.   
4. The inventory contains the relevant products.   
  
Postconditions:   
1. The shopping cart is updated with the customer's changes.   
2. The system reflects the updated cart in the user interface.   
3. The inventory remains unchanged unless a "reserve" feature is implemented.   
  
Main Flow:   
1. The customer navigates to the shopping cart.   
2. The customer reviews the items currently in the cart.   
3. The customer selects an item to modify (e.g., change quantity, remove item, or replace with another product).   
4. The system updates the cart accordingly.   
5. The system recalculates the total price and updates the display.   
6. The system confirms the cart update and displays the new cart contents.   
  
Alternative Flow:   
1. If the selected item is out of stock, the system removes it from the cart and displays a warning message.   
2. If the customer attempts to add more quantity than available, the system displays an error and adjusts the quantity to the maximum available.   
3. If the system encounters an error while updating the cart, it displays an error message and allows the customer to retry the operation.   
4. If the customer removes all items from the cart, the system displays a message indicating that the cart is empty and may suggest browsing products again.  
  
Use Case Name: Remove Product from Cart   
Use Case ID: UC-06   
Actors: Customer, Shopping Cart, System, Product, Inventory   
  
Preconditions:   
1. The customer is logged into the system.   
2. The shopping cart contains at least one product.   
3. The system is accessible and functioning properly.   
4. The customer has navigated to the shopping cart page.   
  
Postconditions:   
1. The specified product is removed from the customer's shopping cart.   
2. The system updates the cart display to reflect the removal.   
3. The inventory remains unchanged.   
  
Main Flow:   
1. The customer navigates to the shopping cart page.   
2. The system displays the current items in the cart.   
3. The customer selects the product to be removed.   
4. The customer clicks on the "Remove" button for the selected product.   
5. The system confirms the removal and updates the cart contents.   
6. The system recalculates the total price and displays the updated cart summary.   
  
Alternative Flow:   
1. If the customer cancels the removal action, the system retains the product in the cart and displays the original cart.   
2. If the system fails to remove the product due to an error, it displays an error message and allows the customer to retry the action.   
3. If the cart becomes empty after the removal, the system displays a message indicating the cart is empty and may suggest browsing products again.  
  
Use Case Name: Checkout   
Use Case ID: UC-07   
Actors: Customer, Shopping Cart, Order, Payment, System, Inventory   
  
Preconditions:   
1. The customer is logged into the system.   
2. The customer has at least one item in their shopping cart.   
3. The shopping cart is accessible and displayed correctly.   
4. The inventory contains the items in the cart.   
5. The system has access to a payment processing gateway.   
  
Postconditions:   
1. The customer's order is created and stored in the system.   
2. The payment is processed successfully.   
3. The shopping cart is emptied.   
4. The inventory is updated to reflect the reduction in product quantity.   
5. The customer receives an email confirmation of the order and payment.   
  
Main Flow:   
1. The customer navigates to the shopping cart.   
2. The customer reviews the items and confirms the order.   
3. The system displays the checkout page with shipping and billing information.   
4. The customer enters or confirms their shipping address.   
5. The customer selects a preferred shipping method.   
6. The system calculates the total price, including taxes and shipping.   
7. The customer selects a payment method and enters the necessary payment details.   
8. The system processes the payment and confirms its success.   
9. The system creates a new order and updates the inventory by reducing the product quantities.   
10. The system sends an email confirmation to the customer.   
11. The shopping cart is cleared of all items.   
  
Alternative Flow:   
1. If the customer's shipping address is invalid or incomplete, the system displays an error and prompts for correction.   
2. If the selected shipping method is unavailable, the system suggests alternative options.   
3. If the payment fails, the system displays an error and allows the customer to retry with a different payment method.   
4. If the inventory does not have enough stock for the requested items, the system displays a warning and suggests adjusting the order.   
5. If the email confirmation cannot be sent, the system displays a message and allows the customer to request a resend.   
6. If the customer cancels the checkout process, the system returns to the shopping cart without making any changes.  
  
Use Case Name: Confirm Order   
Use Case ID: UC-08   
Actors: Customer, Order, Payment, System, Email Confirmation   
  
Preconditions:   
1. The customer is logged into the system.   
2. The shopping cart contains at least one item.   
3. The customer has completed the checkout process and provided necessary information.   
4. The system is connected to a payment gateway.   
  
Postconditions:   
1. The order is confirmed and finalized.   
2. The payment is processed and recorded in the system.   
3. The customer receives an email confirmation of the order.   
4. The inventory is updated to reflect the reduction in product quantity.   
5. The shopping cart is emptied.   
  
Main Flow:   
1. The customer completes the checkout process and reaches the final confirmation page.   
2. The system displays a summary of the order, including items, shipping address, payment method, and total amount.   
3. The customer reviews the order summary and clicks the "Confirm Order" button.   
4. The system initiates the payment transaction through the payment gateway.   
5. The payment gateway processes the payment and returns a success status.   
6. The system creates a new order and records the transaction details.   
7. The system updates the inventory by reducing the stock of ordered items.   
8. The system sends an email confirmation to the customer's registered email address.   
9. The system clears the shopping cart of all items.   
10. The customer receives a confirmation message on the screen and in their email.   
  
Alternative Flow:   
1. If the customer decides not to confirm the order, the system allows them to return to the checkout page for modifications.   
2. If the payment gateway returns a failure, the system displays an error message and allows the customer to retry or choose another payment method.   
3. If the inventory is insufficient for the order at the time of confirmation, the system cancels the order and notifies the customer.   
4. If the email confirmation cannot be sent, the system logs an error and allows the customer to request a manual resend.   
5. If the system encounters an error during order creation, it displays an error message and provides an option to retry the confirmation process.  
  
Use Case Name: Receive Order Confirmation Email   
Use Case ID: UC-09   
Actors: Customer, System, Email Confirmation, Order   
  
Preconditions:   
1. The customer has completed the checkout process and confirmed the order.   
2. The system has successfully processed the payment.   
3. The customer has a valid and registered email address.   
4. The system is connected to an email server.   
  
Postconditions:   
1. The customer receives an email confirmation with the order details.   
2. The system logs the email as successfully sent.   
3. The customer is aware of the order confirmation and can reference the details in the email.   
  
Main Flow:   
1. The system triggers the email confirmation process after the order is successfully confirmed.   
2. The system generates an email containing the order summary, including product list, total amount, shipping address, and payment method.   
3. The system sends the email to the customer's registered email address.   
4. The customer receives the email confirmation in their inbox.   
5. The system marks the email as successfully delivered in the logs.   
  
Alternative Flow:   
1. If the email cannot be sent due to a server issue, the system logs an error and allows the customer to request a manual resend.   
2. If the customer's email address is invalid, the system displays an error message and prompts the customer to update their contact information.   
3. If the customer does not receive the email, they can navigate to the order history in the system to view the order details and request the email to be resent.   
4. If the customer clicks on a tracking link included in the email, the system redirects them to the order status page.  
  
Use Case Name: Administrator Login   
Use Case ID: UC-10   
Actors: Administrator, System   
Preconditions:   
1. The administrator has a valid account in the system.   
2. The system is operational and accessible.   
3. The administrator is not currently logged in.   
  
Postconditions:   
1. The administrator is successfully logged into the system.   
2. The system displays the administrator dashboard or control panel.   
3. The administrator's session is initiated.   
  
Main Flow:   
1. The administrator navigates to the administrator login page.   
2. The administrator enters their username and password.   
3. The system verifies the credentials against the database.   
4. If the credentials are valid, the system grants access to the administrator.   
5. The system displays the administrator dashboard with available management options.   
  
Alternative Flow:   
1. If the username is not found, the system displays an error message and prompts the administrator to re-enter the username.   
2. If the password is incorrect, the system displays an error message and allows the administrator to retry.   
3. If the administrator exceeds the maximum number of failed login attempts, the system temporarily locks the account and sends an email notification.   
4. If the administrator forgets their password, they can click on the "Forgot Password" link to initiate a password reset process.  
  
Use Case Name: Manage Inventory   
Use Case ID: UC-11   
Actors: Administrator, Inventory, Product, Category, Plugin   
Preconditions:   
1. The administrator is logged into the system.   
2. The system has access to the inventory database.   
3. The administrator has the necessary permissions to manage inventory.   
4. The system is connected to product and category data entities.   
  
Postconditions:   
1. The inventory is updated with the administrator's changes.   
2. Product availability is adjusted accordingly.   
3. The system logs the inventory update for tracking purposes.   
4. The administrator receives a confirmation message of the inventory changes.   
  
Main Flow:   
1. The administrator navigates to the inventory management section.   
2. The system displays a list of products along with their current stock levels and categories.   
3. The administrator selects a product to update its inventory.   
4. The administrator modifies the product's stock quantity, updates product details, or changes its category.   
5. The system validates the changes and checks for consistency (e.g., stock quantity cannot be negative).   
6. The system updates the inventory database with the new information.   
7. The system displays a confirmation message indicating the inventory has been successfully updated.   
8. The administrator can view the updated inventory or perform additional actions.   
  
Alternative Flow:   
1. If the administrator enters a negative stock quantity, the system displays an error and prompts for a valid quantity.   
2. If the product is not found in the inventory, the system displays an error and allows the administrator to search for the product.   
3. If the system fails to update the inventory due to a database error, it displays an error message and allows the administrator to retry the operation.   
4. If the administrator attempts to modify a product managed by a plugin, the system checks plugin rules and may restrict or redirect the action accordingly.   
5. If the category of a product is changed, the system updates all related product listings and search results to reflect the new category.  
  
Use Case Name: Update Product Information   
Use Case ID: UC-12   
Actors: Administrator, Product, Inventory, Category, Plugin   
Preconditions:   
1. The administrator is logged into the system.   
2. The system has access to the product and inventory databases.   
3. The administrator has the necessary permissions to update product information.   
4. The product to be updated exists in the inventory.   
  
Postconditions:   
1. The product's information is updated in the system.   
2. The inventory reflects the updated product details.   
3. The product is re-categorized if necessary.   
4. The system logs the update for tracking purposes.   
5. The administrator receives a confirmation message of the changes.   
  
Main Flow:   
1. The administrator navigates to the product management section.   
2. The system displays a list of products with their current details.   
3. The administrator selects a product to update.   
4. The administrator modifies the product's name, price, description, category, or other relevant details.   
5. The system validates the changes (e.g., price must be a positive number, description must not be empty).   
6. The system updates the product information in the database.   
7. The system updates the inventory and category associations if applicable.   
8. The system displays a confirmation message indicating the product has been successfully updated.   
  
Alternative Flow:   
1. If the administrator enters an invalid price (e.g., negative or non-numeric), the system displays an error and prompts for a valid price.   
2. If the product is managed by a plugin and the update violates plugin rules, the system displays a warning and may restrict the update.   
3. If the system fails to update the product information due to a database error, it displays an error message and allows the administrator to retry the operation.   
4. If the category selected is invalid or does not exist, the system displays an error and prompts the administrator to select a valid category.   
5. If the product does not exist in the inventory, the system displays an error and allows the administrator to search for the product or create a new one.  
  
Use Case Name: Manage Categories   
Use Case ID: UC-13   
Actors: Administrator, Category, Product, Plugin   
Preconditions:   
1. The administrator is logged into the system.   
2. The system has access to the category and product databases.   
3. The administrator has the necessary permissions to manage categories.   
4. The system is connected to plugins if any category-related rules are enforced.   
  
Postconditions:   
1. The category is added, edited, or deleted in the system.   
2. All related product listings are updated to reflect category changes.   
3. The system logs the category management action for tracking purposes.   
4. The administrator receives a confirmation message of the changes.   
  
Main Flow:   
1. The administrator navigates to the category management section.   
2. The system displays a list of existing categories.   
3. The administrator selects an action (e.g., add new category, edit existing category, delete category).   
4. If adding a new category, the administrator enters the category name and description.   
5. If editing or deleting, the administrator selects the specific category and makes the necessary changes.   
6. The system validates the category name and checks for conflicts with existing categories.   
7. The system updates the category database with the new or modified information.   
8. The system updates all product listings and search filters to reflect the category changes.   
9. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the category name is already in use, the system displays an error and prompts the administrator to choose a different name.   
2. If the category is managed by a plugin and the action violates plugin rules, the system displays a warning and may restrict the action.   
3. If the system fails to update the category due to a database error, it displays an error message and allows the administrator to retry.   
4. If the administrator attempts to delete a category with associated products, the system prompts for confirmation and provides an option to reassign those products to another category.  
  
Use Case Name: Create Category   
Use Case ID: UC-14   
Actors: Administrator, Category, Plugin   
Preconditions:   
1. The administrator is logged into the system.   
2. The system is operational and accessible.   
3. The administrator has the necessary permissions to create a new category.   
4. The system is connected to the category database and any relevant plugins.   
  
Postconditions:   
1. A new category is successfully created and stored in the system.   
2. The system updates the category list to include the new category.   
3. The system logs the category creation for tracking purposes.   
4. The administrator receives a confirmation message that the category was created.   
  
Main Flow:   
1. The administrator navigates to the category management section.   
2. The administrator selects the "Create New Category" option.   
3. The administrator enters the category name and description.   
4. The system validates the category name for uniqueness and correctness.   
5. The system checks if any plugin rules apply to the category creation.   
6. The system creates the new category and stores it in the category database.   
7. The system updates the category list and displays the confirmation message.   
8. The administrator can now use the new category for product organization.   
  
Alternative Flow:   
1. If the category name is already in use, the system displays an error and prompts the administrator to choose a different name.   
2. If the category name is invalid (e.g., empty or contains special characters), the system displays an error and prompts the administrator to re-enter the name.   
3. If the system encounters a database error during category creation, it displays an error message and allows the administrator to retry the operation.   
4. If a plugin restricts the category creation, the system displays a warning and may prevent the action or suggest modifications to comply with the rules.  
  
Use Case Name: Delete Category   
Use Case ID: UC-15   
Actors: Administrator, Category, Product, Plugin   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The system has access to the category and product databases.   
3. The administrator has the necessary permissions to delete a category.   
4. The category to be deleted exists in the system.   
5. The system is connected to plugins if any category-related rules are enforced.   
  
Postconditions:   
1. The specified category is deleted from the system.   
2. Products previously associated with the category are either reassigned or marked as uncategorized.   
3. The system updates the category list and related product listings.   
4. The system logs the deletion for tracking purposes.   
5. The administrator receives a confirmation message that the category was deleted.   
  
Main Flow:   
1. The administrator navigates to the category management section.   
2. The system displays a list of existing categories.   
3. The administrator selects the category to be deleted.   
4. The system checks if the category contains any associated products.   
5. If the category has associated products, the system prompts the administrator to reassign them to another category or mark them as uncategorized.   
6. The administrator confirms the deletion and selects an option for handling the associated products.   
7. The system deletes the category from the database.   
8. The system updates the product records and category list accordingly.   
9. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the category is managed by a plugin and the deletion violates plugin rules, the system displays a warning and may restrict the action.   
2. If the administrator cancels the deletion after selecting the category, the system retains the category and displays the category list.   
3. If the system fails to delete the category due to a database error, it displays an error message and allows the administrator to retry the operation.   
4. If the administrator chooses not to reassign products and the system defaults to marking them as uncategorized, the system updates the product status and logs the action.   
5. If the category does not contain any products, the system proceeds with the deletion without prompting for product reassignment.  
  
Use Case Name: Plugin Installation   
Use Case ID: UC-16   
Actors: Administrator, Plugin, System   
Preconditions:   
1. The administrator is logged into the system.   
2. The system is operational and accessible.   
3. The plugin to be installed is available and compatible with the system.   
4. The administrator has the necessary permissions to install plugins.   
5. The system has a plugin management interface.   
  
Postconditions:   
1. The plugin is successfully installed and activated in the system.   
2. The system updates the list of installed plugins.   
3. The administrator receives a confirmation message of the plugin installation.   
4. The plugin is integrated with relevant system components (e.g., inventory, categories, or email confirmation).   
  
Main Flow:   
1. The administrator navigates to the plugin management section.   
2. The system displays a list of available plugins.   
3. The administrator selects the plugin to be installed.   
4. The system checks for compatibility and dependencies of the selected plugin.   
5. The administrator confirms the installation.   
6. The system downloads and installs the plugin.   
7. The system activates the plugin and integrates it with the relevant system modules.   
8. The system updates the list of installed plugins.   
9. The administrator receives a confirmation message that the plugin has been successfully installed.   
  
Alternative Flow:   
1. If the selected plugin is not compatible with the system, the system displays an error and prevents the installation.   
2. If the plugin has missing dependencies, the system prompts the administrator to install them first.   
3. If the system fails to download or install the plugin due to an error, it displays an error message and allows the administrator to retry the operation.   
4. If the administrator cancels the installation, the system returns to the plugin list without making any changes.   
5. If the plugin requires configuration after installation, the system redirects the administrator to a setup page for plugin settings.  
  
Use Case Name: Plugin Configuration   
Use Case ID: UC-17   
Actors: Administrator, Plugin, System   
Preconditions:   
1. The administrator is logged into the system.   
2. The plugin has been successfully installed and is listed in the plugin management interface.   
3. The system is operational and accessible.   
4. The administrator has the necessary permissions to configure plugins.   
  
Postconditions:   
1. The plugin is configured with the specified settings.   
2. The system updates the plugin configuration in the database.   
3. The plugin is active and functioning according to the new settings.   
4. The administrator receives a confirmation message of the configuration.   
  
Main Flow:   
1. The administrator navigates to the plugin management section.   
2. The system displays a list of installed plugins.   
3. The administrator selects a plugin to configure.   
4. The system opens the plugin configuration page with current settings.   
5. The administrator modifies the plugin settings as needed.   
6. The system validates the new settings for correctness and completeness.   
7. The system saves the updated configuration to the database.   
8. The system reactivates the plugin with the new settings.   
9. The administrator receives a confirmation message that the plugin has been successfully configured.   
  
Alternative Flow:   
1. If the plugin configuration contains invalid or missing values, the system displays an error and prompts the administrator to correct the input.   
2. If the plugin is not compatible with the current system settings, the system displays a warning and may restrict the configuration.   
3. If the system fails to save the configuration due to a database error, it displays an error message and allows the administrator to retry the operation.   
4. If the administrator cancels the configuration process, the system retains the previous settings and displays the plugin list.   
5. If the plugin requires a system restart to apply the changes, the system prompts the administrator to restart the system and confirms the action after it is completed.  
  
Use Case Name: View Payment Information   
Use Case ID: UC-04   
Actors: Customer, Payment, System   
Preconditions:   
1. The customer is logged into the system.   
2. The customer has at least one order in the system.   
3. The system contains payment records for the customer's orders.   
4. The customer has navigated to the order history or payment information section.   
  
Postconditions:   
1. The customer views the payment details for a specific order.   
2. The system displays the payment method, transaction status, and amount.   
3. The customer is aware of the payment information and can take further actions if needed.   
  
Main Flow:   
1. The customer navigates to their order history or payment information section.   
2. The system displays a list of past orders with relevant details.   
3. The customer selects a specific order to view its payment information.   
4. The system retrieves the payment details associated with the selected order.   
5. The system displays the payment method used, transaction status, payment amount, and date.   
6. The customer reviews the displayed payment information.   
  
Alternative Flow:   
1. If the selected order does not have a payment record, the system displays an error and suggests checking other orders.   
2. If the system fails to retrieve payment information due to an error, it displays an error message and allows the customer to retry.   
3. If the customer clicks on a "Resend Payment Confirmation" option, the system sends an email confirmation of the payment details.   
4. If the customer wants to update or change their payment method for future orders, the system redirects them to the payment settings page.  
  
Use Case Name: Manage Cart Item   
Use Case ID: UC-18   
Actors: Customer, Shopping Cart, System, Product, Inventory   
  
Preconditions:   
1. The customer is logged into the system.   
2. The shopping cart contains at least one item.   
3. The system is accessible and functioning properly.   
4. The customer has navigated to the shopping cart page or the relevant product page.   
  
Postconditions:   
1. The cart item is modified or removed as per the customer's request.   
2. The shopping cart is updated to reflect the changes.   
3. The system recalculates and displays the updated total price.   
4. The inventory remains unchanged unless the cart has a "reserve" functionality.   
  
Main Flow:   
1. The customer navigates to the shopping cart page or views a product.   
2. The customer selects an item in the cart to manage (e.g., change quantity, update product, or remove item).   
3. The system displays options to modify the selected cart item.   
4. The customer modifies the cart item by changing the quantity, updating the product (if applicable), or selecting to remove it.   
5. The system checks the availability of the updated quantity in the inventory.   
6. The system updates the cart item with the new information.   
7. The system recalculates the total price of the cart.   
8. The system confirms the cart item update and displays the revised cart contents.   
  
Alternative Flow:   
1. If the requested quantity exceeds the available stock, the system displays a warning and adjusts the quantity to the maximum available.   
2. If the customer selects to update a cart item to a product that is out of stock, the system displays an error and does not apply the change.   
3. If the system encounters an error while updating the cart item, it displays an error message and allows the customer to retry the operation.   
4. If the customer attempts to remove the last item in the cart, the system displays a message that the cart is now empty and may suggest browsing products again.   
5. If the customer cancels the cart item modification, the system retains the original cart item and displays the unchanged cart.  
  
Use Case Name: Update Order Detail   
Use Case ID: UC-19   
Actors: Administrator, Order, Order Detail, Product, Inventory, System   
  
Preconditions:   
1. The administrator is logged into the system.   
2. The system contains at least one order with associated order details.   
3. The administrator has the necessary permissions to update order details.   
4. The system is connected to the order and product databases.   
5. The order is in a modifiable state (e.g., not yet shipped or canceled).   
  
Postconditions:   
1. The order detail is updated in the system with the new information.   
2. The system reflects the updated order detail in the user interface.   
3. The inventory is adjusted if the product quantity in the order detail is changed.   
4. The system logs the update for tracking purposes.   
5. The administrator receives a confirmation message of the changes.   
  
Main Flow:   
1. The administrator navigates to the order management section.   
2. The system displays a list of orders with their associated details.   
3. The administrator selects a specific order to view and update its details.   
4. The system displays the order details, including the list of products, quantities, prices, and total amount.   
5. The administrator selects an order detail item to modify.   
6. The administrator updates the product quantity, replaces the product with another available item, or modifies the price.   
7. The system validates the changes (e.g., quantity must not exceed available stock, price must be a positive number).   
8. If the product quantity is updated, the system recalculates the total order amount.   
9. If a product is replaced, the system updates the inventory to reflect the change (e.g., reduce stock of the new product and increase stock of the removed one).   
10. The system updates the order detail in the database and displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the updated product quantity exceeds the available stock, the system displays an error and prompts the administrator to adjust the quantity.   
2. If the administrator attempts to update an order that has already been shipped or canceled, the system displays a warning and prevents the update.   
3. If the system encounters an error during the update, it displays an error message and allows the administrator to retry the operation.   
4. If the administrator replaces a product in the order with one that is not in stock, the system displays an error and does not apply the change.   
5. If the administrator cancels the update action after making changes, the system reverts to the original order detail and displays the order list.