项目文档

# Functional Requirement

1.1 Customer Account Registration   
Function ID: FR-01   
Description: A customer can register for a new account by providing valid personal information, including a valid email address and a strong password. The system will validate the input, send an email confirmation, and activate the account after confirmation.   
Input: Name, email, password, and optional contact details.   
Output: A new customer account in the system, with email confirmed and account active.  
  
1.2 Customer Login   
Function ID: FR-02   
Description: A registered and confirmed customer can log in using their email and password. The system will verify the credentials and establish a session for the customer.   
Input: Customer email and password.   
Output: Active customer session and redirection to the home page or shopping cart.  
  
1.3 Customer Logout   
Function ID: FR-03   
Description: A logged-in customer can end their session by selecting the logout option. The system will terminate the session and redirect the user to the home page or login page.   
Input: Request to logout from the account menu.   
Output: Session termination and redirection to the home page or login page.  
  
1.4 Product Browsing   
Function ID: FR-04   
Description: Customers can view a list of available products, filter or sort by attributes such as category and price, and view product details, including inventory status.   
Input: Product listing page access or search criteria.   
Output: Display of product information, including name, price, description, inventory status, and an updated shopping cart.  
  
1.5 Product Search   
Function ID: FR-05   
Description: Customers can search for products by name, keyword, or apply filters. The system will return a list of matching products and their details.   
Input: Search query or filter criteria.   
Output: List of matching products with details such as name, price, and inventory status.  
  
1.6 Product Addition to Cart   
Function ID: FR-06   
Description: A customer can add a selected product to their shopping cart after verifying its availability. The system will update the cart with the product and quantity.   
Input: Product ID and quantity.   
Output: Updated shopping cart with the added product and total price.  
  
1.7 Cart Viewing   
Function ID: FR-07   
Description: A customer can view the contents of their shopping cart, including product names, quantities, and total cost.   
Input: Request to view cart.   
Output: Cart contents with product details and total price.  
  
1.8 Cart Modification   
Function ID: FR-08   
Description: A customer can modify the quantity of a product in the cart or remove items. The system will update the cart and recalculate the total cost.   
Input: Product ID and new quantity or removal request.   
Output: Updated cart with new quantities or removed items and recalculated total price.  
  
1.9 Cart Deletion   
Function ID: FR-09   
Description: A customer can delete all items in the shopping cart at once. The system will clear the cart and update the cart summary.   
Input: Request to delete all items in the cart.   
Output: Empty shopping cart and updated cart summary.  
  
1.10 Order Checkout   
Function ID: FR-10   
Description: A logged-in customer with a valid cart can proceed to checkout, provide shipping and payment details, and confirm the order. The system will process the payment, update inventory, and generate an order confirmation.   
Input: Cart contents, shipping address, payment method, and customer confirmation.   
Output: Confirmed order, updated inventory, and an order confirmation email sent to the customer.  
  
1.11 Order Confirmation   
Function ID: FR-11   
Description: After successful checkout, the system confirms the order by updating the order status and sending an email confirmation to the customer.   
Input: Confirmed order and payment details.   
Output: Order marked as confirmed in the system and an email confirmation sent to the customer.  
  
1.12 Order History Viewing   
Function ID: FR-12   
Description: A logged-in customer can view their order history, including order numbers, dates, total costs, and status. The system can also re-send order confirmation emails.   
Input: Request to view order history and optional request to re-send email.   
Output: Display of order history and re-sent email confirmation if requested.  
  
1.13 Product Inventory Management   
Function ID: FR-13   
Description: An administrator can add, update, or delete products in the inventory. The system will reflect these changes in the product list and inventory status.   
Input: Product details (name, description, price, category, quantity) and modification action (add, update, delete).   
Output: Updated inventory database and product list with a success message.  
  
1.14 Product Categorization   
Function ID: FR-14   
Description: An administrator can assign a product to a specific category or change its current category. The system will update the category assignment and display the result.   
Input: Product ID and new category ID or name.   
Output: Updated product category in the database and refreshed product list.  
  
1.15 Product Information Update   
Function ID: FR-15   
Description: An administrator can update the details of an existing product, such as its name, price, description, and category. The system will validate and store the updated information.   
Input: Product ID and updated fields (name, price, description, category).   
Output: Updated product information in the database and success message for the administrator.  
  
1.16 Administrator Login   
Function ID: FR-16   
Description: An administrator can log in using their registered email and password. The system will authenticate the credentials and redirect the administrator to the dashboard.   
Input: Administrator email and password.   
Output: Active administrator session and redirection to the dashboard.  
  
1.17 Administrator Logout   
Function ID: FR-17   
Description: A logged-in administrator can end their session by selecting the logout option. The system will terminate the session and redirect the user to the login page or home page.   
Input: Request to logout from the account menu.   
Output: Session termination and redirection to the login page or home page.  
  
1.18 Payment Processing   
Function ID: FR-18   
Description: The system processes a customer's payment during checkout, verifies the transaction, and updates the order and inventory accordingly.   
Input: Payment details (method, card information), order details, and customer session.   
Output: Processed and confirmed payment, updated order status, and updated inventory levels.  
  
1.19 Payment Method Configuration   
Function ID: FR-19   
Description: A customer can add or edit their preferred payment methods. The system will validate and securely store the updated payment information.   
Input: Payment method details (e.g., card number, expiration date, billing address).   
Output: Updated list of payment methods in the customer's account and confirmation message.  
  
1.20 Email Notification Setup   
Function ID: FR-20   
Description: A customer can configure their email notification preferences, such as for order updates, promotions, or inventory alerts. The system will validate the email and send a confirmation.   
Input: Notification preferences and valid email address.   
Output: Updated notification settings and confirmation email sent.  
  
1.21 Email Confirmation Sending   
Function ID: FR-21   
Description: The system sends an email confirmation to a customer for specific actions such as account registration, order placement, or notification setup.   
Input: Customer email address and confirmation action (e.g., registration, checkout).   
Output: Confirmation email with a verification link sent to the customer.  
  
1.22 Manage Product Info   
Function ID: FR-22   
Description: An administrator can modify the information of a product, such as its name, price, description, or category. The system will validate and store the updated information.   
Input: Product ID and updated information fields.   
Output: Updated product information in the database and success message for the administrator.  
  
1.23 Manage Order Info   
Function ID: FR-23   
Description: An administrator can modify the status, shipping address, or payment method of an order. The system will update the order and related records, and notify the customer of the changes.   
Input: Order ID and updated fields (e.g., status, address).   
Output: Updated order, inventory, and payment records; email notification sent to the customer.  
  
1.24 Manage Administrator Info   
Function ID: FR-24   
Description: An administrator can edit their own account information, such as email, password, or profile details. The system will validate and store the updated information.   
Input: Administrator ID and updated account details.   
Output: Updated administrator account in the system and a confirmation message.

# 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 external entities such as users, hardware, software components, and communication protocols.  
  
2.1 User Interface Output   
The user interface (UI) of the system is designed to support both customers and administrators in performing actions and receiving relevant feedback. The UI includes the following components:   
  
- \*\*Account Registration Page\*\*: Provides a form for customers to input their name, email, password, and optional contact details. After successful registration, the system displays a confirmation message and redirects the user to the login page.   
- \*\*Login Page\*\*: Allows registered users (both customers and administrators) to input their email and password. Upon successful authentication, the system redirects the user to their respective home page or dashboard.   
- \*\*Home Page\*\*: Displays the main navigation options, including product browsing, cart access, and order history.   
- \*\*Product Listing Page\*\*: Enables customers to view a list of products, apply filters (e.g., category, price), and sort results. The system displays product information such as name, price, description, and inventory status.   
- \*\*Product Detail Page\*\*: Provides detailed information about a selected product, including its availability and options to add it to the cart.   
- \*\*Shopping Cart Page\*\*: Shows the contents of the customer’s shopping cart, including product names, quantities, and total cost. The cart can be modified by changing quantities or removing items. The system also supports deletion of all items in the cart at once.   
- \*\*Checkout Page\*\*: Allows customers to provide shipping and payment details and confirm their order. The system displays a summary of the cart, shipping address, and payment method.   
- \*\*Order Confirmation Page\*\*: Displays a confirmation message after a successful checkout and provides a summary of the order details.   
- \*\*Order History Page\*\*: Shows a list of the customer’s past orders, including order numbers, dates, total costs, and statuses. It also allows the customer to request a re-sent confirmation email.   
- \*\*Administrator Dashboard\*\*: Provides administrators with tools to manage products, orders, and their own account information. It includes forms for adding, updating, and deleting products or orders.   
- \*\*Payment Method Configuration Page\*\*: Allows customers to add or edit their preferred payment methods. The system validates and securely stores the information.   
- \*\*Email Notification Preferences Page\*\*: Enables customers to configure their notification preferences, such as for order updates, promotions, or inventory alerts. The system validates the email and sends a confirmation.   
  
Each of these user interfaces is designed to be intuitive and user-friendly, with clear input fields, validation feedback, and output displays. The system ensures that all user interactions are secure, especially when handling sensitive data such as payment information.  
  
2.2 Hardware Interface Output   
The system does not directly interact with any external hardware devices. However, it is designed to be compatible with standard hardware used by web-based applications, including:   
  
- \*\*Web Browsers\*\*: The system is accessible through modern web browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge. It supports responsive design to ensure compatibility with various screen sizes, including desktops, laptops, tablets, and smartphones.   
- \*\*Mobile Devices\*\*: The system is optimized for use on mobile devices, with touch-friendly navigation and adaptive layouts.   
  
No specific hardware communication protocols are required for the system, as all interactions are conducted through standard web technologies.  
  
2.3 Software Interface Output   
The system interacts with several external software components, primarily through APIs and database connections. These include:   
  
- \*\*Database Systems\*\*: The system stores and retrieves data from a relational database. The database contains tables for customers, products, orders, payment methods, and notification preferences. The system uses SQL queries to interact with the database, ensuring data integrity and efficient data retrieval.   
 - \*\*Customer Database\*\*: Stores customer account information, including name, email, password, and contact details.   
 - \*\*Product Database\*\*: Stores product details such as name, description, price, category, and inventory status.   
 - \*\*Order Database\*\*: Stores order details, including order number, customer ID, product IDs, quantities, shipping address, payment method, and order status.   
 - \*\*Payment Database\*\*: Stores customer payment methods, including card information and billing addresses. The data is encrypted for security.   
 - \*\*Notification Database\*\*: Stores customer notification preferences, including email subscriptions for order updates, promotions, and inventory alerts.   
  
- \*\*Payment Gateway API\*\*: The system integrates with a third-party payment gateway API to process transactions securely. The API receives payment details (e.g., card number, expiration date, billing address) and returns a transaction confirmation or error message. The system uses HTTPS to communicate with the API, ensuring secure data transmission.   
  
- \*\*Email Service API\*\*: The system interacts with an email service API to send confirmation emails to customers. The API receives the customer’s email address and a confirmation action (e.g., registration, checkout, notification setup), and sends a confirmation email with a verification link. The system uses HTTPS to communicate with the API.   
  
- \*\*Inventory Management System\*\*: The system updates the inventory database during checkout and product management actions. This ensures that product availability is accurately reflected in the product listing and detail pages.   
  
2.4 Communication Interface Output   
The system communicates with external entities through the following methods:   
  
- \*\*Email Notifications\*\*: The system sends email notifications to customers for actions such as account registration, order placement, and notification preference updates. These emails include a verification link that customers can click to confirm their actions. The system uses an email service API to handle the sending of these notifications securely.   
- \*\*Web Browsing\*\*: The system is accessed via standard web protocols (HTTP/HTTPS). Users interact with the system through web browsers, and the system communicates with the client-side (e.g., JavaScript, HTML) to provide dynamic content and real-time updates.   
- \*\*API Communication\*\*: The system communicates with external APIs for payment processing and email notifications. The APIs are accessed using HTTPS, and the system sends and receives structured data in JSON format.   
  
Each communication interface is designed to be secure, reliable, and efficient, ensuring that the system can interact seamlessly with external services and users.

# Use Case

Use Case Name: Customer Registration   
Use Case ID: UC-01   
Actors: Customer, Administrator   
Preconditions:   
1. The system is operational and accessible to the customer.   
2. The customer has not previously registered an account.   
  
Postconditions:   
1. A new customer account is successfully created in the system.   
2. The customer receives an email confirmation for registration.   
  
Main Flow:   
1. The customer navigates to the registration page on the system.   
2. The system displays the customer registration form.   
3. The customer fills in the required information (e.g., name, email, password).   
4. The customer submits the registration form.   
5. The system validates the input data (e.g., checks for valid email format, password strength).   
6. The system creates a new customer account with the provided information.   
7. The system sends an email confirmation to the customer’s email address.   
8. The customer clicks the confirmation link in the email.   
9. The system confirms the customer’s email and completes the registration process.   
10. The customer is redirected to the login page or home page with a success message.   
  
Alternative Flow:   
1. If the system detects invalid or missing input data in step 5, it displays an error message to the customer and prompts them to correct the information.   
2. If the email is already registered in the system, the system displays an error message indicating that the email is already in use.   
3. If the customer does not click the confirmation link within a specified time (e.g., 24 hours), the system deactivates the account and requires the customer to re-initiate registration.   
4. If the email confirmation fails, the system logs the error and allows the customer to request a new confirmation email.  
  
Use Case Name: Customer Login   
Use Case ID: UC-02   
Actors: Customer   
Preconditions:   
1. The system is operational and accessible to the customer.   
2. The customer has already completed the registration process and received email confirmation.   
3. The customer has an active account in the system.   
  
Postconditions:   
1. The customer is successfully authenticated and logged into the system.   
2. The system displays the customer’s shopping cart or home page.   
  
Main Flow:   
1. The customer navigates to the login page on the system.   
2. The system displays the customer login form.   
3. The customer enters their registered email and password.   
4. The customer submits the login form.   
5. The system verifies the email and password against the stored customer account data.   
6. If the authentication is successful, the system logs the customer in and redirects them to their shopping cart or home page.   
7. The system displays a success message confirming the login.   
  
Alternative Flow:   
1. If the system detects invalid or incorrect email or password in step 5, it displays an error message and prompts the customer to re-enter the correct credentials.   
2. If the customer account is locked due to multiple failed login attempts, the system displays a message indicating the account is temporarily locked and suggests waiting or contacting support.   
3. If the customer forgets their password, they can click the "Forgot Password" link to initiate a password reset process.   
4. If the password reset is requested, the system generates a reset token, sends it to the customer's email, and allows the customer to reset their password by following the link in the email.  
  
Use Case Name: Customer Logout   
Use Case ID: UC-03   
Actors: Customer   
Preconditions:   
1. The system is operational and accessible to the customer.   
2. The customer is currently logged into the system.   
  
Postconditions:   
1. The customer's session is terminated.   
2. The customer is redirected to the home page or login page.   
  
Main Flow:   
1. The customer navigates to the account or user menu while logged in.   
2. The system displays the "Logout" option.   
3. The customer selects the "Logout" option.   
4. The system terminates the active session and clears the session data.   
5. The customer is redirected to the home page or login page.   
6. The system displays a message confirming the customer has been successfully logged out.   
  
Alternative Flow:   
1. If the customer closes the browser or navigates away without selecting "Logout," the system automatically logs them out after a session timeout.   
2. If the system fails to terminate the session, it logs the error and displays a message to the customer to try again or contact support.  
  
Use Case Name: Product Browsing   
Use Case ID: UC-04   
Actors: Customer   
Preconditions:   
1. The system is operational and accessible to the customer.   
2. The customer is either logged in or not logged in.   
  
Postconditions:   
1. The customer can view product details, categories, and availability.   
2. The customer can add products to the shopping cart.   
  
Main Flow:   
1. The customer navigates to the product listing page on the system.   
2. The system displays a list of available products categorized under different sections.   
3. The customer selects a product from the list or uses search functionality to locate a specific product.   
4. The system displays detailed information about the selected product, including price, description, and inventory status.   
5. The customer can filter or sort products by category, price, or other attributes.   
6. If the customer wishes to purchase the product, they can add it to their shopping cart.   
7. The system updates the shopping cart and confirms the addition of the product.   
  
Alternative Flow:   
1. If no products are found in the search results, the system displays a message indicating that no products match the criteria.   
2. If the product is out of stock, the system displays a message indicating the inventory status and may suggest similar products.   
3. If the customer is not logged in and attempts to add a product to the cart, the system allows the action and prompts the customer to create an account or log in to proceed with checkout.   
4. If the system fails to load product details, it displays an error message and logs the issue for administrator review.  
  
Use Case Name: Product Search   
Use Case ID: UC-05   
Actors: Customer   
Preconditions:   
1. The system is operational and accessible to the customer.   
2. The customer is either logged in or not logged in.   
3. The product database contains at least one product.   
  
Postconditions:   
1. The customer can view a list of products matching their search criteria.   
2. The system displays product details such as name, price, and inventory status.   
3. The customer can add products to the shopping cart if available.   
  
Main Flow:   
1. The customer navigates to the product search page or uses the search bar on the home page.   
2. The system displays a search input field and optional filters (e.g., category, price range).   
3. The customer enters a product name, keyword, or applies filters to narrow down the search.   
4. The customer submits the search request.   
5. The system queries the product database based on the search criteria.   
6. The system displays a list of relevant products along with their details.   
7. The customer selects a product from the search results.   
8. The system shows the product page with detailed information and allows the customer to add it to the shopping cart.   
  
Alternative Flow:   
1. If no products match the search criteria, the system displays a message stating "No products found."   
2. If the search results include products that are out of stock, the system indicates the inventory status and may suggest alternatives.   
3. If the system fails to retrieve the search results, it displays an error message and logs the issue for administrator review.   
4. If the customer is not logged in and adds a product to the cart, the system prompts the customer to create an account or log in to complete the purchase.  
  
Use Case Name: Product Addition to Cart   
Use Case ID: UC-06   
Actors: Customer   
Preconditions:   
1. The system is operational and accessible to the customer.   
2. The customer is browsing a product or has searched for a product.   
3. The product is available in the inventory.   
  
Postconditions:   
1. The selected product is added to the customer's shopping cart.   
2. The shopping cart is updated and displayed with the new product.   
  
Main Flow:   
1. The customer views a product on the product page or in search results.   
2. The customer selects the quantity of the product they wish to purchase.   
3. The customer clicks the "Add to Cart" button.   
4. The system checks the product's inventory status and confirms availability.   
5. The system adds the product and quantity to the shopping cart.   
6. The system displays a confirmation message that the product has been added to the cart.   
7. The system updates the shopping cart summary (e.g., total items, total price).   
  
Alternative Flow:   
1. If the product is out of stock, the system displays an error message and suggests similar products.   
2. If the selected quantity exceeds the available inventory, the system displays an error message and highlights the maximum available quantity.   
3. If the customer is not logged in, the system allows the addition to the cart and prompts them to create an account or log in to proceed with checkout.   
4. If the system fails to add the product to the cart, it displays an error message and logs the issue for administrator review.  
  
Use Case Name: Cart Viewing   
Use Case ID: UC-07   
Actors: Customer   
Preconditions:   
1. The system is operational and accessible to the customer.   
2. The customer has at least one product in their shopping cart.   
3. The customer is either logged in or not logged in.   
  
Postconditions:   
1. The customer can view the contents of their shopping cart, including product details and total price.   
2. The customer can modify or remove items from the cart.   
  
Main Flow:   
1. The customer navigates to the shopping cart page.   
2. The system displays the contents of the shopping cart.   
3. The system lists each product in the cart with its name, quantity, price, and total item cost.   
4. The customer can adjust the quantity of each product or remove items from the cart.   
5. The system updates the cart in real-time with the new quantity or item removal.   
6. The system recalculates and displays the updated total price of the cart.   
7. The customer can choose to proceed to checkout or continue shopping.   
  
Alternative Flow:   
1. If the customer’s cart is empty, the system displays a message indicating that no items are currently in the cart.   
2. If the system fails to load the cart contents, it displays an error message and logs the issue for administrator review.   
3. If the customer attempts to increase the quantity beyond the available inventory, the system displays an error message and highlights the maximum available quantity.   
4. If the customer is not logged in and modifies the cart, the system retains the changes temporarily and prompts them to create an account or log in to proceed with checkout.  
  
Use Case Name: Cart Modification   
Use Case ID: UC-08   
Actors: Customer   
Preconditions:   
1. The system is operational and accessible to the customer.   
2. The customer has at least one product in their shopping cart.   
3. The customer is either logged in or not logged in.   
  
Postconditions:   
1. The shopping cart is updated with the modified quantities or removed items.   
2. The total price of the shopping cart is recalculated and displayed.   
3. The system retains the modified cart until the customer proceeds to checkout or logs out.   
  
Main Flow:   
1. The customer navigates to the shopping cart page.   
2. The system displays the cart contents, including product name, quantity, and price.   
3. The customer selects an item in the cart and modifies the quantity or removes the item.   
4. The system updates the cart accordingly.   
5. The system recalculates the total cart price based on the updated contents.   
6. The system displays the updated cart and total price to the customer.   
7. The customer can choose to continue shopping or proceed to checkout.   
  
Alternative Flow:   
1. If the customer attempts to set a quantity higher than the available inventory, the system displays an error message and highlights the maximum available quantity.   
2. If the system fails to update the cart, it displays an error message and logs the issue for administrator review.   
3. If the customer removes all items from the cart, the system displays a message indicating that the cart is now empty.   
4. If the customer is not logged in and modifies the cart, the system retains the changes temporarily and prompts them to create an account or log in to proceed with checkout.  
  
Use Case Name: Cart Deletion   
Use Case ID: UC-09   
Actors: Customer   
Preconditions:   
1. The system is operational and accessible to the customer.   
2. The customer has a shopping cart with at least one item.   
3. The customer is either logged in or not logged in.   
  
Postconditions:   
1. The shopping cart is emptied of all items.   
2. The system updates the cart summary to reflect the deletion.   
3. The system retains the deletion action until the customer adds new items or logs out.   
  
Main Flow:   
1. The customer navigates to the shopping cart page.   
2. The system displays the cart contents, including product name, quantity, and price.   
3. The customer selects the "Delete Cart" option or clicks a "Clear Cart" button.   
4. The system prompts the customer to confirm the deletion of all items in the cart.   
5. The customer confirms the deletion.   
6. The system removes all items from the shopping cart.   
7. The system updates the cart summary and displays a confirmation message that the cart has been cleared.   
8. The customer can choose to continue shopping or return to the home page.   
  
Alternative Flow:   
1. If the customer cancels the deletion in step 4, the system retains the cart contents and does not perform any deletion.   
2. If the system fails to delete the cart contents, it displays an error message and logs the issue for administrator review.   
3. If the customer is not logged in and deletes the cart, the system clears the temporary cart and prompts the customer to create an account or log in to save future selections.  
  
Use Case Name: Order Checkout   
Use Case ID: UC-10   
Actors: Customer, Shopping Cart, Payment, Email Confirmation, Inventory, Order   
Preconditions:   
1. The system is operational and accessible to the customer.   
2. The customer has added at least one product to their shopping cart.   
3. The customer is logged into their account.   
4. The shopping cart contains products with confirmed availability in the inventory.   
  
Postconditions:   
1. The customer’s order is successfully placed and stored in the system.   
2. The payment is processed, and the transaction is recorded.   
3. The inventory is updated to reflect the deduction of ordered products.   
4. The customer receives an email confirmation for the order.   
  
Main Flow:   
1. The customer navigates to the shopping cart page and selects "Proceed to Checkout."   
2. The system displays the checkout form, including shipping address, payment method, and order summary.   
3. The customer fills in or selects their preferred shipping address and payment method.   
4. The customer reviews the cart contents and confirms the total price.   
5. The customer submits the checkout form.   
6. The system processes the payment and verifies the transaction.   
7. The system creates a new order and updates the inventory by deducting the ordered product quantities.   
8. The system sends an email confirmation to the customer’s registered email address.   
9. The system displays a success message and provides an order confirmation number.   
  
Alternative Flow:   
1. If the customer’s selected quantity exceeds the available inventory, the system displays an error message and highlights the maximum available quantity.   
2. If the payment fails, the system displays an error message and prompts the customer to select an alternative payment method.   
3. If the customer cancels the checkout process, the system returns them to the shopping cart page without making any changes.   
4. If the email confirmation fails to send, the system logs the error and allows the customer to request a new confirmation email.  
  
Use Case Name: Order Confirmation   
Use Case ID: UC-11   
Actors: Customer, Email Confirmation, Order, Inventory, Payment   
Preconditions:   
1. The system is operational and accessible to the customer.   
2. The customer has completed the checkout process and submitted the order.   
3. The payment has been successfully processed.   
4. The inventory has been updated to reflect the deduction of ordered products.   
  
Postconditions:   
1. The customer receives an email confirmation for their order.   
2. The order is marked as confirmed in the system.   
3. The customer is redirected to an order confirmation page.   
  
Main Flow:   
1. The system completes the payment and inventory update during the checkout process.   
2. The system generates an order confirmation message containing order details (e.g., order number, items, total cost, shipping address, and estimated delivery time).   
3. The system sends the confirmation email to the customer’s registered email address.   
4. The customer receives the email confirmation and can review the order summary.   
5. The system displays an order confirmation page to the customer, summarizing the order and providing a confirmation number.   
6. The system marks the order as confirmed in the database.   
  
Alternative Flow:   
1. If the email confirmation fails to send, the system logs the error and allows the customer to request a new confirmation email.   
2. If the system fails to display the confirmation page, it shows an error message and logs the issue for administrator review.   
3. If the customer’s email is invalid or unreachable, the system displays a message indicating that the confirmation email could not be sent and suggests contacting support.   
4. If the inventory update during checkout fails, the system rolls back the transaction and displays an error message to the customer, prompting them to retry the checkout process.  
  
Use Case Name: Order History Viewing   
Use Case ID: UC-12   
Actors: Customer, Order, Email Confirmation   
Preconditions:   
1. The system is operational and accessible to the customer.   
2. The customer is logged into their account.   
3. The customer has previously placed at least one order.   
  
Postconditions:   
1. The customer can view a list of their previous orders with details (e.g., order number, date, total cost, and status).   
2. The system displays order details, including product information and shipping address.   
3. The customer can access an email confirmation for a specific order if needed.   
  
Main Flow:   
1. The customer navigates to the "Order History" section in their account dashboard.   
2. The system retrieves the customer's order history from the database.   
3. The system displays a list of previous orders with summary information (e.g., order date, total price, and status).   
4. The customer selects a specific order to view detailed information.   
5. The system shows the full details of the selected order, including items, quantities, shipping address, and payment method.   
6. The customer can request a re-sent email confirmation for the selected order.   
7. The system sends the email confirmation containing the order details to the customer's registered email address.   
8. The system displays a success message confirming that the email has been sent.   
  
Alternative Flow:   
1. If the customer has no previous orders, the system displays a message stating "No order history found."   
2. If the system fails to retrieve the order history, it displays an error message and logs the issue for administrator review.   
3. If the customer's request to re-send the email confirmation fails, the system logs the error and displays a message allowing the customer to try again or contact support.   
4. If the selected order is not found in the system, the system displays an error message and redirects the customer back to the order history list.  
  
Use Case Name: Product Inventory Management   
Use Case ID: UC-13   
Actors: Administrator, Product, Inventory, Category   
Preconditions:   
1. The system is operational and accessible to the administrator.   
2. The administrator is logged into their account.   
3. The inventory database is initialized and contains product records.   
  
Postconditions:   
1. The administrator can add, update, or remove products in the inventory.   
2. Product information is updated in the system.   
3. Inventory levels are adjusted accordingly.   
4. The system displays updated inventory status to the administrator.   
  
Main Flow:   
1. The administrator navigates to the inventory management section of the system.   
2. The system displays a list of current products in the inventory with details (e.g., product name, quantity, category, and price).   
3. The administrator selects an action (e.g., add new product, update product, or delete product).   
4. If adding a new product, the administrator fills in product details such as name, description, price, category, and inventory quantity.   
5. The system validates the input data and saves the new product to the inventory database.   
6. If updating an existing product, the administrator modifies the product details and selects the "Save Changes" option.   
7. The system updates the product record in the inventory database.   
8. If deleting a product, the administrator selects the product and confirms the deletion.   
9. The system removes the product from the inventory database.   
10. The system updates the inventory list and displays a success message to the administrator.   
  
Alternative Flow:   
1. If the administrator enters invalid or incomplete product information in step 4 or 6, the system displays an error message and prompts the administrator to correct the data.   
2. If the selected product for deletion is associated with an active order, the system displays a warning and prevents the deletion until the order is canceled or completed.   
3. If the system fails to update or delete a product, it displays an error message and logs the issue for administrator review.   
4. If the category for a product does not exist, the administrator is prompted to create the category or select an existing one.  
  
Use Case Name: Product Categorization   
Use Case ID: UC-14   
Actors: Administrator, Product, Category   
Preconditions:   
1. The system is operational and accessible to the administrator.   
2. The administrator is logged into their account.   
3. There is at least one product and one category in the system.   
  
Postconditions:   
1. The product is assigned to the appropriate category.   
2. The category list is updated to reflect the new or modified categorization.   
3. The system displays a success message to the administrator.   
  
Main Flow:   
1. The administrator navigates to the product categorization section of the system.   
2. The system displays a list of products along with their current category assignments.   
3. The administrator selects a product to modify its category.   
4. The system shows the available categories and the current category of the selected product.   
5. The administrator chooses a new category for the product or confirms the current one.   
6. The system updates the product’s category in the database.   
7. The system refreshes the product list to show the updated categorization.   
8. The administrator receives a success message confirming the change.   
  
Alternative Flow:   
1. If the selected category does not exist, the system prompts the administrator to create a new category or select an existing one.   
2. If the product is already in the selected category, the system displays a message indicating that no changes were made.   
3. If the system fails to update the product’s category, it displays an error message and logs the issue for administrator review.   
4. If the administrator attempts to categorize a product that is not in the system, the system displays an error message and suggests checking the product database.  
  
Use Case Name: Product Information Update   
Use Case ID: UC-15   
Actors: Administrator, Product, Inventory, Category   
Preconditions:   
1. The system is operational and accessible to the administrator.   
2. The administrator is logged into their account.   
3. The product exists in the inventory database.   
  
Postconditions:   
1. The product's information (e.g., name, price, description, category) is updated in the system.   
2. The inventory status is adjusted if necessary.   
3. The system displays a success message to the administrator.   
  
Main Flow:   
1. The administrator navigates to the product management section of the system.   
2. The system displays a list of products with their current information.   
3. The administrator selects a specific product to update.   
4. The system shows the product details, including name, description, price, category, and inventory quantity.   
5. The administrator modifies the required product information (e.g., updates the price, changes the category, or edits the description).   
6. The administrator clicks the "Save Changes" button.   
7. The system validates the updated information (e.g., checks for valid price format, ensures the category exists).   
8. The system updates the product record in the database.   
9. The system refreshes the product list and displays a success message.   
  
Alternative Flow:   
1. If the administrator enters invalid or incomplete information in step 5, the system displays an error message and prompts the administrator to correct the data.   
2. If the selected category for the product does not exist, the system prompts the administrator to create a new category or select an existing one.   
3. If the system fails to update the product information, it displays an error message and logs the issue for administrator review.   
4. If the product is associated with active orders and the inventory quantity is reduced, the system displays a warning to ensure the change does not affect order fulfillment.  
  
Use Case Name: Administrator Login   
Use Case ID: UC-04   
Actors: Administrator   
Preconditions:   
1. The system is operational and accessible to the administrator.   
2. The administrator has a valid account with registered credentials.   
3. The administrator has not previously logged in.   
  
Postconditions:   
1. The administrator is successfully authenticated and logged into the system.   
2. The system displays the administrator dashboard or main interface.   
  
Main Flow:   
1. The administrator navigates to the administrator login page.   
2. The system displays the login form, including fields for email and password.   
3. The administrator enters their registered email and password.   
4. The administrator submits the login form.   
5. The system verifies the email and password against the stored administrator account data.   
6. If the authentication is successful, the system logs the administrator in and redirects them to the dashboard.   
7. The system displays a success message confirming the login.   
  
Alternative Flow:   
1. If the system detects invalid or incorrect email or password in step 5, it displays an error message and prompts the administrator to re-enter the correct credentials.   
2. If the administrator account is locked due to multiple failed login attempts, the system displays a message indicating the account is temporarily locked and suggests waiting or contacting support.   
3. If the administrator forgets their password, they can click the "Forgot Password" link to initiate a password reset process.   
4. If the password reset is requested, the system generates a reset token, sends it to the administrator's email, and allows them to reset their password by following the link in the email.  
  
Use Case Name: Administrator Logout   
Use Case ID: UC-05   
Actors: Administrator   
Preconditions:   
1. The system is operational and accessible to the administrator.   
2. The administrator is currently logged into the system.   
  
Postconditions:   
1. The administrator's session is terminated.   
2. The administrator is redirected to the home page or administrator login page.   
  
Main Flow:   
1. The administrator navigates to the account or user menu while logged in.   
2. The system displays the "Logout" option.   
3. The administrator selects the "Logout" option.   
4. The system terminates the active session and clears the session data.   
5. The administrator is redirected to the home page or login page.   
6. The system displays a message confirming the administrator has been successfully logged out.   
  
Alternative Flow:   
1. If the administrator closes the browser or navigates away without selecting "Logout," the system automatically logs them out after a session timeout.   
2. If the system fails to terminate the session, it logs the error and displays a message to the administrator to try again or contact support.  
  
Use Case Name: Payment Processing   
Use Case ID: UC-10   
Actors: Customer, Shopping Cart, Payment, Email Confirmation, Inventory, Order   
Preconditions:   
1. The system is operational and accessible to the customer.   
2. The customer has added at least one product to their shopping cart.   
3. The customer is logged into their account.   
4. The shopping cart contains products with confirmed availability in the inventory.   
  
Postconditions:   
1. The customer’s order is successfully placed and stored in the system.   
2. The payment is processed, and the transaction is recorded.   
3. The inventory is updated to reflect the deduction of ordered products.   
4. The customer receives an email confirmation for the order.   
  
Main Flow:   
1. The customer navigates to the shopping cart page and selects "Proceed to Checkout."   
2. The system displays the checkout form, including shipping address, payment method, and order summary.   
3. The customer fills in or selects their preferred shipping address and payment method.   
4. The customer reviews the cart contents and confirms the total price.   
5. The customer submits the checkout form.   
6. The system processes the payment and verifies the transaction.   
7. The system creates a new order and updates the inventory by deducting the ordered product quantities.   
8. The system sends an email confirmation to the customer’s registered email address.   
9. The system displays a success message and provides an order confirmation number.   
  
Alternative Flow:   
1. If the customer’s selected quantity exceeds the available inventory, the system displays an error message and highlights the maximum available quantity.   
2. If the payment fails, the system displays an error message and prompts the customer to select an alternative payment method.   
3. If the customer cancels the checkout process, the system returns them to the shopping cart page without making any changes.   
4. If the email confirmation fails to send, the system logs the error and allows the customer to request a new confirmation email.  
  
Use Case Name: Payment Method Configuration   
Use Case ID: UC-16   
Actors: Customer, Administrator, Payment, Order   
  
Preconditions:   
1. The system is operational and accessible to the customer.   
2. The customer is logged into their account.   
3. The customer has at least one order in progress or has previously configured payment methods.   
  
Postconditions:   
1. The customer's preferred payment method is updated or added in the system.   
2. The system displays a confirmation message indicating the successful configuration.   
3. The updated payment method is available for future orders.   
  
Main Flow:   
1. The customer navigates to the "Payment Methods" section in their account settings.   
2. The system displays a list of currently configured payment methods, if any.   
3. The customer selects the "Add New Payment Method" or "Edit Existing Payment Method" option.   
4. The customer enters the required payment details (e.g., credit card number, expiration date, CVV, billing address).   
5. The customer submits the payment method configuration.   
6. The system validates the input data (e.g., checks card number format, expiration date validity).   
7. The system securely stores the payment method in the customer's account.   
8. The system displays a confirmation message that the payment method has been successfully configured.   
  
Alternative Flow:   
1. If the system detects invalid or incomplete payment information in step 4, it displays an error message and prompts the customer to correct the data.   
2. If the customer attempts to add a duplicate payment method (e.g., same card number), the system displays a warning and allows the customer to proceed or cancel.   
3. If the system fails to save the payment method, it displays an error message and logs the issue for administrator review.   
4. If the customer cancels the configuration process, the system discards any unsaved changes and returns to the payment method list.  
  
Use Case Name: Plugin Installation   
Use Case ID: UC-17   
Actors: Administrator, Plugin, System   
Preconditions:   
1. The system is operational and accessible to the administrator.   
2. The administrator is logged into their account.   
3. A plugin is available for installation (e.g., from a repository or file upload).   
4. The system supports plugin installation functionality.   
  
Postconditions:   
1. The selected plugin is installed and activated in the system.   
2. The system displays a confirmation message indicating the plugin installation was successful.   
3. The administrator can manage or configure the plugin from the system interface.   
4. The plugin is available for use by relevant system components or users.   
  
Main Flow:   
1. The administrator navigates to the "Plugin Management" section in the system.   
2. The system displays a list of available plugins or provides an option to upload a new plugin.   
3. The administrator selects a plugin to install or uploads a new plugin file.   
4. The system validates the plugin file (e.g., checks for correct format, compatibility).   
5. The system proceeds to install the plugin.   
6. The system activates the plugin and integrates it into the system functionality.   
7. The system displays a confirmation message that the plugin has been successfully installed and activated.   
  
Alternative Flow:   
1. If the plugin file is invalid or incompatible, the system displays an error message and prevents installation.   
2. If the system detects that the plugin is already installed, it displays a message indicating the plugin is already active.   
3. If the installation process fails, the system logs the error and displays a message to the administrator, suggesting troubleshooting or reinstallation.   
4. If the administrator cancels the installation before completion, the system terminates the process and returns to the plugin management page without changes.  
  
Use Case Name: Plugin Configuration   
Use Case ID: UC-18   
Actors: Administrator, Plugin, System   
  
Preconditions:   
1. The system is operational and accessible to the administrator.   
2. The administrator is logged into their account.   
3. The plugin has been successfully installed and is available for configuration.   
4. The system provides a plugin configuration interface.   
  
Postconditions:   
1. The plugin is configured with the specified settings.   
2. The system applies the configuration changes to the plugin.   
3. The plugin is functional according to the updated settings.   
4. The administrator receives a confirmation message that the configuration was successful.   
  
Main Flow:   
1. The administrator navigates to the "Plugin Management" section in the system.   
2. The system displays a list of installed plugins, including the one to be configured.   
3. The administrator selects the plugin and clicks the "Configure" option.   
4. The system opens the plugin configuration interface with default or current settings.   
5. The administrator modifies the plugin settings as needed (e.g., enables/disables features, sets parameters).   
6. The administrator clicks the "Save Configuration" button.   
7. The system validates the configuration input and applies the changes to the plugin.   
8. The system displays a confirmation message indicating the plugin has been successfully configured.   
9. The administrator can test the plugin or return to the plugin management interface.   
  
Alternative Flow:   
1. If the system detects invalid or incomplete configuration input in step 5, it displays an error message and prompts the administrator to correct the data.   
2. If the plugin configuration is unchanged, the system displays a message indicating no changes were made.   
3. If the system fails to apply the configuration, it logs the error and displays a message to the administrator, suggesting troubleshooting or reconfiguration.   
4. If the administrator cancels the configuration process, the system discards any unsaved changes and returns to the plugin management page.  
  
Use Case Name: Email Notification Setup   
Use Case ID: UC-19   
Actors: Customer, Administrator, Email Confirmation   
Preconditions:   
1. The system is operational and accessible to the customer.   
2. The customer is logged into their account.   
3. The system supports email notification configuration.   
4. The administrator has configured the system to allow customers to set up email notifications.   
  
Postconditions:   
1. The customer's email notification preferences are updated in the system.   
2. The system sends a confirmation email to the customer to verify the setup.   
3. The customer receives notifications based on their selected preferences.   
4. The administrator can review and manage email notification configurations if required.   
  
Main Flow:   
1. The customer navigates to the "Email Notifications" section in their account settings.   
2. The system displays the current email notification preferences (e.g., order updates, promotions, inventory alerts).   
3. The customer selects or modifies their preferred notification options.   
4. The customer enters an email address to receive notifications (if not already set).   
5. The customer submits the notification setup form.   
6. The system validates the email address format and updates the notification preferences in the customer's account.   
7. The system sends an email confirmation to the provided email address.   
8. The system displays a success message to the customer indicating that the email notification setup is complete.   
  
Alternative Flow:   
1. If the customer provides an invalid email address, the system displays an error message and prompts the customer to correct it.   
2. If the email confirmation fails to send, the system logs the error and allows the customer to request a new confirmation email.   
3. If the customer does not confirm the email within a specified time (e.g., 24 hours), the system disables the notification settings until re-confirmation.   
4. If the system detects no changes in the notification preferences, it displays a message indicating that no updates were made.  
  
Use Case Name: Email Confirmation Sending   
Use Case ID: UC-20   
Actors: Customer, Administrator, Email Confirmation, System   
Preconditions:   
1. The system is operational and accessible to the customer.   
2. The customer has initiated an action that requires an email confirmation (e.g., registration, order placement, email notification setup).   
3. The system has access to an email service to send confirmation emails.   
4. The customer has provided a valid email address.   
  
Postconditions:   
1. The customer receives a confirmation email containing a verification link.   
2. The system logs the email confirmation request and delivery status.   
3. The email confirmation is successfully processed when the customer clicks the verification link.   
4. The system updates the relevant user or order status to reflect confirmation.   
  
Main Flow:   
1. The customer initiates an action requiring email confirmation (e.g., registration, order checkout).   
2. The system prepares an email confirmation message with a unique verification link.   
3. The system sends the confirmation email to the customer’s provided email address.   
4. The customer receives the email and clicks the verification link.   
5. The system validates the verification link and confirms the email address.   
6. The system updates the relevant account or order status to "confirmed."   
7. The system displays a success message to the customer upon confirmation.   
  
Alternative Flow:   
1. If the customer’s email address is invalid, the system logs the error and displays a message prompting the customer to check and update their email.   
2. If the email confirmation fails to send, the system logs the error and allows the customer to request a new confirmation email.   
3. If the verification link is clicked after it has expired, the system displays an error message and offers to send a new confirmation.   
4. If the system fails to update the status after confirmation, it logs the error and displays a message to the customer indicating the issue.  
  
Use Case Name: Manage Product Info   
Use Case ID: UC-21   
Actors: Administrator, Product, Inventory, Category   
  
Preconditions:   
1. The system is operational and accessible to the administrator.   
2. The administrator is logged into their account.   
3. The product exists in the system and is associated with the inventory and category.   
  
Postconditions:   
1. The product's information is updated or corrected in the system.   
2. The system reflects the latest product details in the product database and inventory.   
3. The system displays a success message to the administrator.   
4. The updated product information is available to customers for browsing and purchase.   
  
Main Flow:   
1. The administrator navigates to the "Product Information Management" section in the system.   
2. The system displays a list of products along with their current details (e.g., name, description, price, category, and inventory status).   
3. The administrator selects a product for which they want to manage or update the information.   
4. The system opens a detailed view of the product, allowing the administrator to edit fields such as name, description, price, and category.   
5. The administrator modifies the required product information.   
6. The administrator clicks the "Update Product Info" button.   
7. The system validates the new information (e.g., checks for valid price format, ensures the category exists).   
8. The system updates the product record in the database with the new information.   
9. The system refreshes the product list to reflect the changes and displays a success message to the administrator.   
  
Alternative Flow:   
1. If the administrator enters invalid or incomplete product information, the system displays an error message and prompts the administrator to correct the data.   
2. If the system detects that the selected category for the product does not exist, it prompts the administrator to either create the category or choose an existing one.   
3. If the system fails to update the product information, it displays an error message and logs the issue for administrator review.   
4. If the administrator cancels the update process before submission, the system returns to the product list without making any changes.   
5. If the product is associated with active orders and the changes could impact order fulfillment, the system displays a warning message to the administrator.  
  
Use Case Name: Manage Order Info   
Use Case ID: UC-22   
Actors: Administrator, Order, Customer, Inventory, Payment   
  
Preconditions:   
1. The system is operational and accessible to the administrator.   
2. The administrator is logged into their account.   
3. There is at least one order in the system that requires modification.   
4. The system maintains an order database with associated inventory and payment records.   
  
Postconditions:   
1. The administrator can update or modify the order details (e.g., status, delivery address, or payment method).   
2. The system updates the order status and related records (e.g., inventory, payment).   
3. The customer receives an email notification of the changes made to their order.   
4. The system logs the modification for audit or tracking purposes.   
  
Main Flow:   
1. The administrator navigates to the "Order Management" section in the system.   
2. The system displays a list of orders with key details (e.g., order number, customer name, total price, current status).   
3. The administrator selects an order to modify its information.   
4. The system shows the detailed view of the selected order, including items, shipping address, payment status, and order status.   
5. The administrator modifies the required order information (e.g., updates the status to "shipped," corrects the delivery address, or adjusts the payment method).   
6. The administrator clicks the "Save Changes" or "Update Order" button.   
7. The system validates the changes (e.g., checks for valid status options, ensures the delivery address is properly formatted).   
8. The system updates the order record in the database with the modified information.   
9. The system updates the inventory and payment records as needed (e.g., adjusts inventory levels if order status changes from "pending" to "cancelled").   
10. The system sends an email notification to the customer with the updated order details.   
11. The system displays a success message to the administrator confirming the order information has been updated.   
  
Alternative Flow:   
1. If the administrator attempts to modify an order that is already completed or cancelled, the system displays an error message and prevents the change.   
2. If the system detects invalid or incomplete order information, it displays an error message and prompts the administrator to correct the data.   
3. If the email notification to the customer fails to send, the system logs the error and allows the administrator to manually re-send the notification.   
4. If the system fails to update the order, inventory, or payment records, it displays an error message and logs the issue for administrator review.   
5. If the administrator cancels the modification process before saving, the system discards the changes and returns to the order list without updating any records.  
  
Use Case Name: Manage Administrator Info   
Use Case ID: UC-23   
Actors: Administrator, System   
  
Preconditions:   
1. The system is operational and accessible to the administrator.   
2. The administrator is logged into their account.   
3. The system maintains a database of administrator accounts with associated information (e.g., name, email, role, permissions).   
  
Postconditions:   
1. The administrator's account information is updated or modified in the system.   
2. The system displays a confirmation message to the administrator.   
3. The updated information is available for future system interactions and administrative functions.   
4. The system logs the modification for audit or tracking purposes.   
  
Main Flow:   
1. The administrator navigates to the "Account Management" section in the system.   
2. The system displays the current administrator account details, including name, email, role, and other relevant information.   
3. The administrator selects the "Edit Profile" option.   
4. The system opens a form with editable fields for the administrator to update their information (e.g., change email, update password, or modify profile details).   
5. The administrator modifies the required fields (e.g., updates their email or password).   
6. The administrator clicks the "Save Changes" button.   
7. The system validates the updated information (e.g., checks for valid email format, password strength if applicable).   
8. The system updates the administrator's account in the database with the new information.   
9. The system logs the change, including the date, time, and details of the modification.   
10. The system displays a success message to the administrator confirming the update.   
  
Alternative Flow:   
1. If the administrator enters invalid or incomplete information in step 5, the system displays an error message and prompts the administrator to correct the data.   
2. If the administrator attempts to change their email to one already associated with another account, the system displays an error message and prevents the change.   
3. If the system fails to update the administrator's information, it displays an error message and logs the issue for administrator review.   
4. If the administrator cancels the modification process before saving, the system discards any unsaved changes and returns to the account details page without updating any records.   
5. If the administrator modifies their password, the system may prompt for a re-login or session refresh to ensure updated credentials are applied.