项目文档

# Functional Requirement

Based on the provided use cases and data model, here is the structured Software Requirements Specification for Chapter 1: Functional Requirements:  
  
\*\*1.1 Customer Account Registration\*\*   
Function ID: FR-01   
Description: Registers new customers by validating credentials, creating customer records, initializing shopping carts, and sending confirmation.   
Input: Name (string), Email (string), Password (string), Address (string), Phone (string)   
Output: Customer record (CustomerID, Name, Email, etc.), ShoppingCart record (CartID), confirmation message/email   
  
\*\*1.2 Customer Authentication\*\*   
Function ID: FR-02   
Description: Verifies customer credentials to establish authenticated sessions.   
Input: Email (string), Password (string)   
Output: Authenticated session token, LastLogin timestamp update, homepage redirect   
  
\*\*1.3 Session Termination\*\*   
Function ID: FR-03   
Description: Terminates active customer sessions and redirects to homepage.   
Input: Session token   
Output: Session invalidation, success message, homepage redirect   
  
\*\*1.4 Profile Retrieval\*\*   
Function ID: FR-04   
Description: Displays customer profile details including personal information and order history.   
Input: CustomerID   
Output: Customer record (Name, Email, Address, Phone), associated Order records   
  
\*\*1.5 Profile Modification\*\*   
Function ID: FR-05   
Description: Updates customer profile information after validation.   
Input: CustomerID, updated fields (Name/Email/Password/Address/Phone)   
Output: Updated Customer record, confirmation message   
  
\*\*1.6 Account Deletion\*\*   
Function ID: FR-06   
Description: Removes customer accounts and associated data per data policies.   
Input: CustomerID, confirmation flag   
Output: Deleted Customer record, anonymized associated data, confirmation message   
  
\*\*1.7 Administrator Authentication\*\*   
Function ID: FR-07   
Description: Verifies administrator credentials to grant system access.   
Input: Admin email (string), Password (string)   
Output: Authenticated admin session token, dashboard redirect   
  
\*\*1.8 Admin Session Termination\*\*   
Function ID: FR-08   
Description: Terminates active administrator sessions.   
Input: Admin session token   
Output: Session invalidation, success message, login page redirect   
  
\*\*1.9 Product Creation\*\*   
Function ID: FR-09   
Description: Creates new product records with validated details and category association.   
Input: Product details (Name, Description, Price, StockQuantity, ImageURL), CategoryID   
Output: New Product record, ProductCategory association, updated catalog   
  
\*\*1.10 Product Update\*\*   
Function ID: FR-10   
Description: Modifies existing product attributes and category associations.   
Input: ProductID, updated fields (Name/Description/Price/StockQuantity/ImageURL/CategoryID)   
Output: Updated Product record, modified ProductCategory association   
  
\*\*1.11 Product Removal\*\*   
Function ID: FR-11   
Description: Deletes product records after verifying no active order references.   
Input: ProductID, confirmation flag   
Output: Deleted Product record, updated catalog   
  
\*\*1.12 Product Display\*\*   
Function ID: FR-12   
Description: Retrieves and displays product details for viewing.   
Input: ProductID   
Output: Product record (Name, Description, Price, etc.)   
  
\*\*1.13 Category Creation\*\*   
Function ID: FR-13   
Description: Adds new product classification categories.   
Input: Category details (Name, Description)   
Output: New Category record, updated category list   
  
\*\*1.14 Category Modification\*\*   
Function ID: FR-14   
Description: Updates existing category attributes.   
Input: CategoryID, updated fields (Name/Description)   
Output: Updated Category record   
  
\*\*1.15 Category Removal\*\*   
Function ID: FR-15   
Description: Deletes unused categories after verifying no product associations.   
Input: CategoryID, confirmation flag   
Output: Deleted Category record   
  
\*\*1.16 Category Display\*\*   
Function ID: FR-16   
Description: Retrieves and displays category details with associated products.   
Input: CategoryID   
Output: Category record, associated Product records   
  
\*\*1.17 Cart Item Addition\*\*   
Function ID: FR-17   
Description: Adds products to shopping carts with stock validation.   
Input: CustomerID, ProductID, Quantity   
Output: New CartItem record, updated ShoppingCart total   
  
\*\*1.18 Cart Contents Retrieval\*\*   
Function ID: FR-18   
Description: Displays cart items with quantities, prices, and totals.   
Input: CustomerID   
Output: CartItem records, aggregated price/quantity totals   
  
\*\*1.19 Cart Modification\*\*   
Function ID: FR-19   
Description: Updates cart item quantities or removes items.   
Input: CartItemID, new Quantity (or removal flag)   
Output: Updated/deleted CartItem record, recalculated cart totals   
  
\*\*1.20 Order Placement\*\*   
Function ID: FR-20   
Description: Converts cart items to orders with payment processing and stock deduction.   
Input: CustomerID, PaymentMethodID, shipping/billing details   
Output: New Order record, OrderItem records, Payment record, stock update, cleared cart   
  
\*\*1.21 Order Retrieval\*\*   
Function ID: FR-21   
Description: Displays order details including products and status.   
Input: OrderID   
Output: Order record, associated OrderItem records   
  
\*\*1.22 Order Cancellation\*\*   
Function ID: FR-22   
Description: Cancels eligible orders and restocks products.   
Input: OrderID, confirmation flag   
Output: Updated Order status ("Canceled"), stock quantity adjustments   
  
\*\*1.23 Order Confirmation Delivery\*\*   
Function ID: FR-23   
Description: Sends order confirmation emails upon successful placement.   
Input: OrderID, Customer email   
Output: Email delivery status log, confirmation message   
  
\*\*1.24 Order Status Update\*\*   
Function ID: FR-24   
Description: Modifies order fulfillment status and notifies customers.   
Input: OrderID, new Status (e.g., "Shipped")   
Output: Updated Order status, notification email   
  
\*\*1.25 Payment Method Addition\*\*   
Function ID: FR-25   
Description: Stores validated customer payment instruments.   
Input: CustomerID, PaymentMethod details (Type, CardNumber, ExpiryDate, etc.)   
Output: New PaymentMethod record   
  
\*\*1.26 Payment Method Update\*\*   
Function ID: FR-26   
Description: Modifies existing payment method details.   
Input: PaymentMethodID, updated fields   
Output: Updated PaymentMethod record   
  
\*\*1.27 Payment Method Removal\*\*   
Function ID: FR-27   
Description: Deletes stored payment instruments.   
Input: PaymentMethodID, confirmation flag   
Output: Deleted PaymentMethod record   
  
\*\*1.28 Payment Method Display\*\*   
Function ID: FR-28   
Description: Shows stored payment instrument details.   
Input: PaymentMethodID   
Output: PaymentMethod record   
  
\*\*1.29 Payment Details Retrieval\*\*   
Function ID: FR-29   
Description: Displays transaction details for completed orders.   
Input: PaymentID   
Output: Payment record (Amount, TransactionDate, Status)   
  
\*\*1.30 Payment Status Update\*\*   
Function ID: FR-30   
Description: Modifies payment transaction status and synchronizes order state.   
Input: PaymentID, new Status (e.g., "Completed")   
Output: Updated Payment status, associated Order status update   
  
\*\*1.31 Category Assignment\*\*   
Function ID: FR-31   
Description: Links products to categories for classification.   
Input: ProductID, CategoryID   
Output: New ProductCategory association record   
  
\*\*1.32 Category Removal from Product\*\*   
Function ID: FR-32   
Description: Removes product-category classifications.   
Input: ProductID, CategoryID, confirmation flag   
Output: Deleted ProductCategory association record   
  
---  
This specification strictly adheres to the provided use cases and data model entities/attributes. All inputs/outputs reference defined data elements, and each function maintains technical feasibility through direct transformation of inputs to specified outputs. The structure follows the required sequential numbering and component definitions without introducing external functions.

# External Description

### 2. External Interfaces   
This section defines all external interfaces required by the system, classified into four categories based on functional requirements (Chapter 1). Each interface ensures seamless interaction between the system and external entities.   
  
---  
  
#### \*\*2.1 User Interfaces\*\*   
\*\*Web-Based Client Interface\*\*   
- \*\*Description\*\*: Primary interface for customers and administrators via responsive web browsers.   
- \*\*Interaction\*\*:   
 - \*Customer actions\*: Account registration (FR-01), authentication (FR-02), profile management (FR-04/FR-05), cart operations (FR-17/FR-19), and order placement (FR-20).   
 - \*Administrator actions\*: Product/category management (FR-09–FR-16), order status updates (FR-24).   
- \*\*Output\*\*: HTML/CSS/JS-rendered pages with forms, data displays (e.g., profile details in FR-04), and redirects (e.g., post-login redirects in FR-02/FR-07).   
  
\*\*Mobile-Optimized Web Interface\*\*   
- \*\*Description\*\*: Adaptive rendering of web content for mobile devices.   
- \*\*Interaction\*\*: Supports all customer functions identical to the web interface.   
- \*\*Output\*\*: Responsive UI components (e.g., collapsible menus for cart contents in FR-18).   
  
---  
  
#### \*\*2.2 Hardware Interfaces\*\*   
\*\*Server Infrastructure Interface\*\*   
- \*\*Description\*\*: Interaction with hosting hardware (cloud/server).   
- \*\*Interaction\*\*: Supports high-concurrency operations (e.g., order processing in FR-20, authentication in FR-02/FR-07).   
- \*\*Output\*\*: Hardware resource allocation logs (CPU/memory usage during peak loads).   
  
---  
  
#### \*\*2.3 Software Interfaces\*\*   
\*\*Relational Database (RDBMS) Interface\*\*   
- \*\*Description\*\*: PostgreSQL/MySQL interface for persistent data storage.   
- \*\*Interaction\*\*:   
 - \*CRUD operations\*: Customer records (FR-01/FR-06), product/category updates (FR-09/FR-15), order transactions (FR-20/FR-22).   
 - \*Data validation\*: Input sanitization (e.g., email format in FR-01).   
- \*\*Output\*\*:   
 - Structured data outputs (e.g., `Customer` records in FR-04, `OrderItem` lists in FR-21).   
 - ACID-compliant transaction logs.   
  
\*\*Third-Party Payment Gateway API\*\*   
- \*\*Description\*\*: Integration with external payment processors (e.g., Stripe/PayPal).   
- \*\*Interaction\*\*:   
 - Payment authorization/capture during order placement (FR-20).   
 - Status synchronization (FR-30).   
- \*\*Output\*\*:   
 - API call responses (success/failure codes).   
 - Payment status updates to `Payment` records (FR-29/FR-30).   
  
\*\*Email Service Provider API\*\*   
- \*\*Description\*\*: Connection to transactional email services (e.g., SendGrid/Mailgun).   
- \*\*Interaction\*\*: Sending confirmation emails (FR-01, FR-23, FR-24).   
- \*\*Output\*\*: Email delivery status logs (success/failure receipts).   
  
---  
  
#### \*\*2.4 Communication Interfaces\*\*   
\*\*HTTPS/REST API Interface\*\*   
- \*\*Description\*\*: Secure client-server communication over HTTPS.   
- \*\*Interaction\*\*:   
 - JSON payloads for data exchange (e.g., profile updates in FR-05, cart modifications in FR-19).   
 - Stateless authentication tokens (FR-02/FR-07).   
- \*\*Output\*\*: JSON responses (e.g., order details in FR-21) and HTTP status codes.   
  
\*\*SMTP/Email Interface\*\*   
- \*\*Description\*\*: Outbound email delivery subsystem.   
- \*\*Interaction\*\*:   
 - Triggers for order confirmations (FR-23) and status notifications (FR-24).   
- \*\*Output\*\*:   
 - Email content (templated HTML/text).   
 - Delivery receipts (logged in database).   
  
---  
  
### Cross-Check Validation   
- \*\*Database\*\*: Explicitly referenced in all CRUD-based FRs (e.g., FR-01, FR-09).   
- \*\*Payment Gateway\*\*: Required for FR-20 (payment processing) and FR-30 (status sync).   
- \*\*Email Service\*\*: Mandatory for FR-01, FR-23, and FR-24.   
- \*\*Web UI\*\*: All user-triggered functions (FR-01–FR-28).   
- \*No unaccounted external interfaces identified\*.   
  
Terminology aligns with Chapter 1 (e.g., `CustomerID`, `Order` records). Interfaces are implementation-agnostic to ensure adaptability.

# Use Case

Use Case Name: Register Customer   
Use Case ID: UC-01   
Actors: Customer, Administrator   
Preconditions:   
1. The system is operational.   
2. The customer has access to the system’s registration interface.   
3. The customer has not yet been registered in the system.   
  
Postconditions:   
1. A new customer record is created in the system.   
2. The customer receives a confirmation message.   
3. The shopping cart is initialized for the new customer.   
  
Main Flow:   
1. The customer accesses the registration page.   
2. The customer fills in the required information, including name, email, password, and address.   
3. The system validates the email format and checks for uniqueness.   
4. The system verifies the password meets the security requirements.   
5. The system saves the customer's information in the database.   
6. The system sends a confirmation message to the customer’s email.   
7. The customer is redirected to the homepage with a message indicating successful registration.   
  
Alternative Flow:   
1. If the email is already registered, the system displays an error message and prompts the customer to enter a different email.   
2. If the password does not meet the security requirements, the system provides feedback and asks the customer to re-enter a valid password.   
3. If the system fails to save the customer's information, the administrator is notified, and the customer is informed of the registration failure.  
  
Use Case Name: Login Customer   
Use Case ID: UC-02   
Actors: Customer   
Preconditions:   
1. The system is operational.   
2. The customer has already registered in the system.   
3. The customer has access to the login interface.   
  
Postconditions:   
1. The customer is successfully authenticated and logged into the system.   
2. The customer’s session is initiated.   
3. The system displays the homepage with personalized content for the customer.   
  
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 stored customer records.   
4. The system authenticates the customer and initiates a session.   
5. The customer is redirected to the homepage with a success message.   
  
Alternative Flow:   
1. If the email is not registered, the system displays an error message and prompts the customer to check the email or register.   
2. If the password is incorrect, the system displays an error message and allows the customer to re-enter the password.   
3. If the system fails to authenticate the customer, the customer is redirected to an error page and advised to contact support.  
  
Use Case Name: Logout Customer   
Use Case ID: UC-03   
Actors: Customer   
Preconditions:   
1. The system is operational.   
2. The customer is currently logged into the system.   
3. The logout feature is accessible on the current page.   
  
Postconditions:   
1. The customer is successfully logged out of the system.   
2. The session is terminated.   
3. The customer is redirected to the homepage or login page.   
  
Main Flow:   
1. The customer navigates to the account or profile section.   
2. The customer selects the logout option.   
3. The system confirms the logout request and terminates the session.   
4. The customer is redirected to the homepage or login page.   
5. The system displays a message confirming the successful logout.   
  
Alternative Flow:   
1. If the logout option is unavailable or not selected, the customer remains logged in.   
2. If the system fails to terminate the session, the customer is shown an error message and advised to try again or contact support.  
  
Use Case Name: View Customer Profile   
Use Case ID: UC-04   
Actors: Customer, Administrator   
Preconditions:   
1. The system is operational.   
2. The customer is logged into the system.   
3. The customer profile feature is accessible in the account section.   
  
Postconditions:   
1. The customer profile information is displayed to the customer.   
2. The administrator can access the customer profile for review or modification if authorized.   
3. The session remains active.   
  
Main Flow:   
1. The customer navigates to the account or profile section.   
2. The customer selects the "View Profile" option.   
3. The system retrieves the customer's information from the database.   
4. The system displays the customer profile, including name, email, address, and order history.   
5. The customer reviews the information.   
  
Alternative Flow:   
1. If the system fails to retrieve the customer's data, an error message is displayed, and the customer is advised to try again or contact support.   
2. If the administrator accesses the profile without proper authorization, access is denied and a security message is shown.   
3. If the customer profile is empty or incomplete, the system prompts the customer to update their information.  
  
Use Case Name: Update Customer Profile   
Use Case ID: UC-05   
Actors: Customer, Administrator   
Preconditions:   
1. The system is operational.   
2. The customer is logged into the system.   
3. The customer profile feature is accessible in the account section.   
  
Postconditions:   
1. The customer’s updated information is saved in the system.   
2. The system displays a confirmation message that the profile was successfully updated.   
3. The session remains active.   
  
Main Flow:   
1. The customer navigates to the account or profile section.   
2. The customer selects the "Edit Profile" option.   
3. The system displays the current profile information in editable fields.   
4. The customer modifies the necessary details, such as name, email, password, or address.   
5. The customer clicks the "Save Changes" button.   
6. The system validates the updated information (e.g., email format, password strength).   
7. The system updates the customer’s record in the database.   
8. The system displays a success message and refreshes the profile view.   
  
Alternative Flow:   
1. If the email is already registered by another user, the system displays an error message and prompts the customer to enter a different email.   
2. If the password does not meet the security requirements, the system provides feedback and asks the customer to re-enter a valid password.   
3. If the system fails to save the updated information, the customer is shown an error message and advised to try again or contact support.   
4. If the administrator modifies the profile, the system logs the change and updates the record accordingly.  
  
Use Case Name: Delete Customer Account   
Use Case ID: UC-06   
Actors: Customer, Administrator   
Preconditions:   
1. The system is operational.   
2. The customer is logged into the system or the administrator has access to the customer account management interface.   
3. The customer account to be deleted exists in the system.   
  
Postconditions:   
1. The customer account is removed from the system.   
2. All associated data (e.g., order history, cart items) is deleted or anonymized, as per data policies.   
3. The system sends a confirmation message to the customer or administrator.   
  
Main Flow:   
1. The customer or administrator navigates to the account settings or customer management section.   
2. The user selects the "Delete Account" option.   
3. The system prompts for confirmation to ensure the action is intentional.   
4. The user confirms the deletion request.   
5. The system verifies the user's identity or authorization.   
6. The system proceeds to delete the customer account and associated data.   
7. The system displays a confirmation message that the account has been successfully deleted.   
8. The user is redirected to the homepage or a post-deletion page.   
  
Alternative Flow:   
1. If the user cancels the deletion request, the system returns to the previous page without making changes.   
2. If the system fails to delete the account due to data constraints, an error message is displayed, and the user is advised to contact support.   
3. If the administrator attempts to delete an account without proper authorization, access is denied, and a security message is shown.   
4. If the system fails to send the confirmation message, the user is notified of the issue, and the administrator is alerted to investigate.  
  
Use Case Name: Login Administrator   
Use Case ID: UC-07   
Actors: Administrator   
Preconditions:   
1. The system is operational.   
2. The administrator has access to the system’s login interface.   
3. The administrator has a valid account and credentials in the system.   
  
Postconditions:   
1. The administrator is successfully authenticated and logged into the system.   
2. The administrator’s session is initiated.   
3. The system displays the administrator dashboard with access to management tools.   
  
Main Flow:   
1. The administrator navigates to the login page.   
2. The administrator enters their registered email and password.   
3. The system verifies the email and password against the stored administrator records.   
4. The system authenticates the administrator and initiates a session.   
5. The administrator is redirected to the dashboard with a success message.   
  
Alternative Flow:   
1. If the email is not registered as an administrator, the system displays an error message and prompts the administrator to check the email or contact support.   
2. If the password is incorrect, the system displays an error message and allows the administrator to re-enter the password.   
3. If the system fails to authenticate the administrator, the administrator is redirected to an error page and advised to contact support.  
  
Use Case Name: Logout Administrator   
Use Case ID: UC-08   
Actors: Administrator   
Preconditions:   
1. The system is operational.   
2. The administrator is currently logged into the system.   
3. The logout feature is accessible on the current page.   
  
Postconditions:   
1. The administrator is successfully logged out of the system.   
2. The administrator’s session is terminated.   
3. The administrator is redirected to the homepage or administrator login page.   
  
Main Flow:   
1. The administrator navigates to the account or profile section in the dashboard.   
2. The administrator selects the "Logout" option.   
3. The system confirms the logout request and terminates the session.   
4. The administrator is redirected to the homepage or login page.   
5. The system displays a message confirming the successful logout.   
  
Alternative Flow:   
1. If the logout option is unavailable or not selected, the administrator remains logged in.   
2. If the system fails to terminate the session, the administrator is shown an error message and advised to try again or contact support.  
  
Use Case Name: Add Product   
Use Case ID: UC-09   
Actors: Administrator   
Preconditions:   
1. The system is operational.   
2. The administrator is logged into the system.   
3. The product management interface is accessible.   
  
Postconditions:   
1. A new product record is created in the system.   
2. The product is associated with the correct category.   
3. The product details are displayed in the product catalog.   
  
Main Flow:   
1. The administrator navigates to the product management section.   
2. The administrator selects the "Add Product" option.   
3. The system displays a form for entering product information.   
4. The administrator fills in the product details, including name, description, price, stock quantity, and selects a category.   
5. The administrator uploads product images if available.   
6. The administrator submits the form.   
7. The system validates the input data (e.g., price is numeric, name is not empty).   
8. The system saves the product information in the database and links it to the selected category.   
9. The system displays a success message and updates the product catalog.   
  
Alternative Flow:   
1. If the product name is missing or invalid, the system displays an error message and prompts the administrator to enter a valid name.   
2. If the price is not a valid numeric value, the system provides feedback and asks the administrator to correct the input.   
3. If the system fails to save the product information, the administrator is shown an error message and advised to try again or contact support.   
4. If the selected category does not exist, the system displays an error and prompts the administrator to choose a valid category.  
  
Use Case Name: Update Product Details   
Use Case ID: UC-10   
Actors: Administrator   
Preconditions:   
1. The system is operational.   
2. The administrator is logged into the system.   
3. The product to be updated exists in the system and is accessible in the product management interface.   
  
Postconditions:   
1. The product details are updated in the database.   
2. The product catalog reflects the updated information.   
3. A confirmation message is displayed to the administrator.   
  
Main Flow:   
1. The administrator navigates to the product management section.   
2. The administrator selects the product to be updated from the catalog.   
3. The system displays the product details in editable fields.   
4. The administrator modifies the necessary product information, such as name, description, price, or stock quantity.   
5. The administrator uploads new product images if needed.   
6. The administrator clicks the "Update Product" button.   
7. The system validates the updated data (e.g., price is numeric, name is not empty).   
8. The system updates the product record in the database.   
9. The system displays a success message and refreshes the product catalog.   
  
Alternative Flow:   
1. If the product name is missing or invalid, the system displays an error message and prompts the administrator to enter a valid name.   
2. If the price is not a valid numeric value, the system provides feedback and asks the administrator to correct the input.   
3. If the system fails to update the product information, the administrator is shown an error message and advised to try again or contact support.   
4. If the selected category is changed, the system updates the product's category association accordingly.  
  
Use Case Name: Remove Product   
Use Case ID: UC-11   
Actors: Administrator   
Preconditions:   
1. The system is operational.   
2. The administrator is logged into the system.   
3. The product to be removed exists in the system and is accessible in the product management interface.   
  
Postconditions:   
1. The product record is removed from the system.   
2. The product is no longer displayed in the product catalog.   
3. A confirmation message is displayed to the administrator.   
  
Main Flow:   
1. The administrator navigates to the product management section.   
2. The administrator selects the product to be removed from the catalog.   
3. The system displays the product details and a "Remove Product" option.   
4. The administrator clicks the "Remove Product" button.   
5. The system prompts for confirmation to ensure the action is intentional.   
6. The administrator confirms the removal request.   
7. The system verifies the administrator's authorization.   
8. The system deletes the product record from the database.   
9. The system updates the product catalog and displays a success message.   
  
Alternative Flow:   
1. If the product is referenced in an active order, the system displays an error message and prevents deletion.   
2. If the administrator cancels the removal request, the system returns to the product details page without making changes.   
3. If the system fails to delete the product record, an error message is displayed, and the administrator is advised to try again or contact support.  
  
Use Case Name: View Product   
Use Case ID: UC-12   
Actors: Customer, Administrator   
Preconditions:   
1. The system is operational.   
2. The product exists in the system and is accessible via the product catalog or search feature.   
3. The customer or administrator has access to the product detail page.   
  
Postconditions:   
1. The product details are displayed to the customer or administrator.   
2. The session remains active.   
3. The product information is available for further actions, such as adding to the cart or purchasing.   
  
Main Flow:   
1. The customer or administrator navigates to the product catalog or uses the search feature to locate a product.   
2. The user selects a product from the list.   
3. The system retrieves the product information from the database.   
4. The system displays the product details, including name, description, price, stock quantity, and images.   
5. The user reviews the product information.   
  
Alternative Flow:   
1. If the product does not exist or is not available, the system displays an error message and redirects the user to the catalog or home page.   
2. If the system fails to retrieve the product information, an error message is displayed, and the user is advised to try again or contact support.   
3. If the administrator accesses the product without proper authorization, access is denied, and a security message is shown.  
  
Use Case Name: Add Category   
Use Case ID: UC-13   
Actors: Administrator   
Preconditions:   
1. The system is operational.   
2. The administrator is logged into the system.   
3. The category management interface is accessible.   
  
Postconditions:   
1. A new category record is created in the system.   
2. The category is added to the product catalog and available for product association.   
3. A confirmation message is displayed to the administrator.   
  
Main Flow:   
1. The administrator navigates to the category management section.   
2. The administrator selects the "Add Category" option.   
3. The system displays a form for entering category information.   
4. The administrator fills in the category name and description.   
5. The administrator submits the form.   
6. The system validates the input data (e.g., name is not empty, no duplicate categories).   
7. The system saves the category information in the database.   
8. The system displays a success message and updates the category list.   
  
Alternative Flow:   
1. If the category name is missing or invalid, the system displays an error message and prompts the administrator to enter a valid name.   
2. If the category name already exists, the system displays an error message and prompts the administrator to choose a different name.   
3. If the system fails to save the category information, the administrator is shown an error message and advised to try again or contact support.  
  
Use Case Name: Update Category   
Use Case ID: UC-14   
Actors: Administrator   
Preconditions:   
1. The system is operational.   
2. The administrator is logged into the system.   
3. The category to be updated exists in the system and is accessible in the category management interface.   
  
Postconditions:   
1. The category information is updated in the database.   
2. The category list reflects the updated information.   
3. A confirmation message is displayed to the administrator.   
  
Main Flow:   
1. The administrator navigates to the category management section.   
2. The administrator selects the category to be updated from the category list.   
3. The system displays the category details in editable fields.   
4. The administrator modifies the necessary category information, such as name or description.   
5. The administrator clicks the "Update Category" button.   
6. The system validates the updated data (e.g., name is not empty, no duplicate names).   
7. The system updates the category record in the database.   
8. The system displays a success message and refreshes the category list.   
  
Alternative Flow:   
1. If the category name is missing or invalid, the system displays an error message and prompts the administrator to enter a valid name.   
2. If the category name already exists, the system displays an error message and prompts the administrator to choose a different name.   
3. If the system fails to update the category information, the administrator is shown an error message and advised to try again or contact support.  
  
Use Case Name: Remove Category   
Use Case ID: UC-15   
Actors: Administrator   
Preconditions:   
1. The system is operational.   
2. The administrator is logged into the system.   
3. The category to be removed exists in the system and is accessible in the category management interface.   
  
Postconditions:   
1. The category record is removed from the system.   
2. The category is no longer displayed in the category list or product catalog.   
3. A confirmation message is displayed to the administrator.   
  
Main Flow:   
1. The administrator navigates to the category management section.   
2. The administrator selects the category to be removed from the category list.   
3. The system displays the category details and a "Remove Category" option.   
4. The administrator clicks the "Remove Category" button.   
5. The system prompts for confirmation to ensure the action is intentional.   
6. The administrator confirms the removal request.   
7. The system verifies the administrator's authorization.   
8. The system deletes the category record from the database.   
9. The system updates the category list and displays a success message.   
  
Alternative Flow:   
1. If the category is associated with active products, the system displays an error message and prevents deletion.   
2. If the administrator cancels the removal request, the system returns to the category details page without making changes.   
3. If the system fails to delete the category record, an error message is displayed, and the administrator is advised to try again or contact support.  
  
Use Case Name: View Category   
Use Case ID: UC-16   
Actors: Customer, Administrator   
Preconditions:   
1. The system is operational.   
2. The category exists in the system and is accessible via the category list or product catalog.   
3. The customer or administrator has navigated to the category section.   
  
Postconditions:   
1. The category details are displayed to the customer or administrator.   
2. The session remains active.   
3. The category information is available for further actions, such as browsing products within the category.   
  
Main Flow:   
1. The customer or administrator navigates to the category section or selects a category from the product catalog.   
2. The system retrieves the category information from the database.   
3. The system displays the category details, including name, description, and associated products.   
4. The user reviews the category and its contents.   
  
Alternative Flow:   
1. If the category does not exist or is not available, the system displays an error message and redirects the user to the catalog or home page.   
2. If the system fails to retrieve the category information, an error message is displayed, and the user is advised to try again or contact support.   
3. If the administrator accesses the category without proper authorization, access is denied, and a security message is shown.  
  
Use Case Name: Add to Shopping Cart   
Use Case ID: UC-17   
Actors: Customer   
Preconditions:   
1. The system is operational.   
2. The customer is logged into the system.   
3. The customer has viewed a product and is on the product detail page.   
4. The product is available in stock.   
  
Postconditions:   
1. The selected product is added to the customer's shopping cart.   
2. The shopping cart is updated in the system.   
3. The customer receives a confirmation message of the addition.   
  
Main Flow:   
1. The customer views a product in the product catalog or via search.   
2. The customer selects the "Add to Cart" option for the product.   
3. The system checks the product's availability and validity.   
4. The system adds the product to the customer's shopping cart.   
5. The system updates the cart's total items and price.   
6. The system displays a success message confirming the product was added.   
7. The customer is redirected to the shopping cart page or remains on the product detail page.   
  
Alternative Flow:   
1. If the product is out of stock, the system displays an error message and suggests alternatives or a return to the catalog.   
2. If the product does not exist or is invalid, the system displays an error message and redirects the customer to the catalog.   
3. If the system fails to update the shopping cart, an error message is shown, and the customer is advised to try again or contact support.   
4. If the customer adds multiple units of the product, the system checks against stock limits and updates the cart accordingly.  
  
Use Case Name: View Shopping Cart   
Use Case ID: UC-18   
Actors: Customer   
Preconditions:   
1. The system is operational.   
2. The customer is logged into the system.   
3. The shopping cart feature is accessible from the homepage, product detail page, or account section.   
  
Postconditions:   
1. The contents of the customer's shopping cart are displayed.   
2. The total quantity and price of items in the cart are shown.   
3. The session remains active.   
  
Main Flow:   
1. The customer navigates to the shopping cart page via a cart icon or account section.   
2. The system retrieves the customer's shopping cart data from the database.   
3. The system displays the list of items in the cart, including product name, quantity, price, and total amount.   
4. The customer reviews the items in the cart.   
5. The system provides options to modify quantities or remove items from the cart.   
  
Alternative Flow:   
1. If the shopping cart is empty, the system displays a message indicating no items are in the cart and offers to browse products.   
2. If the system fails to retrieve the shopping cart data, an error message is shown, and the customer is advised to try again or contact support.   
3. If the cart contains products that are now out of stock, the system displays a warning message and suggests removing or replacing the items.  
  
Use Case Name: Modify Shopping Cart   
Use Case ID: UC-19   
Actors: Customer   
Preconditions:   
1. The system is operational.   
2. The customer is logged into the system.   
3. The shopping cart contains at least one item.   
4. The shopping cart feature is accessible from the homepage, product detail page, or account section.   
  
Postconditions:   
1. The customer’s shopping cart is updated with the modified items.   
2. The total quantity and price of items in the cart are recalculated and displayed.   
3. The session remains active.   
  
Main Flow:   
1. The customer navigates to the shopping cart page via a cart icon or account section.   
2. The system retrieves the customer's shopping cart data from the database.   
3. The system displays the list of items in the cart, including product name, quantity, price, and total amount.   
4. The customer selects an item to modify (e.g., change quantity or remove item).   
5. The system updates the cart accordingly.   
6. The system recalculates and displays the updated total quantity and price.   
7. The system saves the modified shopping cart in the database.   
8. The system displays a confirmation message of the modification.   
  
Alternative Flow:   
1. If the customer attempts to modify the quantity to exceed available stock, the system displays an error message and limits the quantity to available stock.   
2. If the customer removes the last item from the cart, the system displays a message indicating the cart is now empty and offers to browse products.   
3. If the system fails to save the modified cart, an error message is shown, and the customer is advised to try again or contact support.   
4. If the product in the cart is no longer available, the system removes it and displays a message informing the customer.  
  
Use Case Name: Place Order   
Use Case ID: UC-20   
Actors: Customer, Administrator   
Preconditions:   
1. The system is operational.   
2. The customer is logged into the system.   
3. The customer's shopping cart contains at least one product.   
4. The customer has provided valid shipping and billing information.   
  
Postconditions:   
1. A new order record is created in the system.   
2. The products in the shopping cart are reserved or deducted from stock.   
3. The customer receives a confirmation message for the order.   
4. The shopping cart is cleared after order placement.   
  
Main Flow:   
1. The customer navigates to the shopping cart page.   
2. The customer selects the "Proceed to Checkout" option.   
3. The system displays the order summary and prompts the customer to enter or confirm shipping and billing details.   
4. The customer confirms the details and selects a payment method.   
5. The system validates the order information, including product availability and payment method.   
6. The system processes the order and reserves or deducts the stock for the ordered products.   
7. The system creates a new order record in the database.   
8. The system sends a confirmation message to the customer's email.   
9. The shopping cart is cleared.   
10. The customer is redirected to the order confirmation page.   
  
Alternative Flow:   
1. If the product is out of stock at the time of checkout, the system displays an error message and suggests removing or replacing the item.   
2. If the customer enters invalid shipping or billing information, the system provides feedback and prompts the customer to correct the details.   
3. If the selected payment method is invalid or unavailable, the system displays an error and offers to choose another method.   
4. If the system fails to process the order, an error message is shown, and the customer is advised to try again or contact support.   
5. If the system fails to update the stock, the administrator is notified, and the customer is informed of the issue.  
  
Use Case Name: View Order   
Use Case ID: UC-21   
Actors: Customer, Administrator   
Preconditions:   
1. The system is operational.   
2. The customer is logged into the system or the administrator has access to the order management interface.   
3. The order to be viewed exists in the system.   
  
Postconditions:   
1. The order details are displayed to the customer or administrator.   
2. The session remains active.   
3. The order information is available for further actions, such as tracking or modification.   
  
Main Flow:   
1. The customer or administrator navigates to the order section or order management interface.   
2. The user selects a specific order from the list of orders.   
3. The system retrieves the order information from the database.   
4. The system displays the order details, including order number, date, total amount, product list, and status.   
5. The user reviews the order information.   
  
Alternative Flow:   
1. If the order does not exist or is not accessible, the system displays an error message and redirects the user to the order list or homepage.   
2. If the system fails to retrieve the order data, an error message is shown, and the user is advised to try again or contact support.   
3. If the administrator accesses the order without proper authorization, access is denied, and a security message is shown.  
  
Use Case Name: Cancel Order   
Use Case ID: UC-22   
Actors: Customer, Administrator   
Preconditions:   
1. The system is operational.   
2. The customer is logged into the system or the administrator has access to the order management interface.   
3. The order to be canceled exists in the system and is in a cancellable state (e.g., pending or processing).   
  
Postconditions:   
1. The order is canceled and updated in the system.   
2. The products in the canceled order are returned to stock.   
3. A confirmation message is displayed to the customer or administrator.   
4. The order status is marked as "Canceled" in the system.   
  
Main Flow:   
1. The customer or administrator navigates to the order section or order management interface.   
2. The user selects an order that is eligible for cancellation.   
3. The system displays the order details and a "Cancel Order" option.   
4. The user clicks the "Cancel Order" button.   
5. The system prompts for confirmation to ensure the action is intentional.   
6. The user confirms the cancellation request.   
7. The system verifies the user's authorization or authentication.   
8. The system updates the order status to "Canceled" and returns the product quantities to the stock.   
9. The system sends a confirmation message to the customer's email and displays a success message.   
10. The order is no longer available for processing or shipment.   
  
Alternative Flow:   
1. If the order is already shipped or completed, the system displays an error message and prevents cancellation.   
2. If the user cancels the cancellation request, the system returns to the order details page without making changes.   
3. If the system fails to update the order status or return stock, an error message is displayed, and the user is advised to try again or contact support.   
4. If the administrator attempts to cancel an order without proper authorization, access is denied, and a security message is shown.  
  
Use Case Name: Send Order Confirmation   
Use Case ID: UC-23   
Actors: Customer, Administrator   
Preconditions:   
1. The system is operational.   
2. The customer has placed an order successfully.   
3. The order record exists in the system and is marked as "Placed".   
4. The customer has provided a valid email address.   
  
Postconditions:   
1. The order confirmation message is sent to the customer's email.   
2. The customer receives a copy of the order summary in the confirmation.   
3. The system logs the confirmation delivery.   
4. The order status remains as "Placed" until further processing.   
  
Main Flow:   
1. The system detects that an order has been successfully placed.   
2. The system generates an order confirmation message containing the order details.   
3. The system prepares to send the message to the customer's registered email.   
4. The system sends the confirmation message via email.   
5. The system logs the successful delivery of the confirmation.   
6. The customer receives the confirmation and reviews the order summary.   
  
Alternative Flow:   
1. If the customer's email is invalid or not properly formatted, the system displays an error message and prompts the customer to update their email.   
2. If the email sending fails, the system logs the failure and notifies the administrator to investigate.   
3. If the customer has not provided an email address, the system prompts the customer to enter one before sending the confirmation.   
4. If the system is unable to access the order data, an error is displayed, and the customer is advised to contact support.  
  
Use Case Name: Update Order Status   
Use Case ID: UC-24   
Actors: Administrator   
Preconditions:   
1. The system is operational.   
2. The administrator is logged into the system.   
3. The order to be updated exists in the system and is accessible in the order management interface.   
  
Postconditions:   
1. The order status is updated in the system.   
2. The customer is notified of the status change via email.   
3. The updated order status is visible to the customer and administrator.   
4. The system logs the status update.   
  
Main Flow:   
1. The administrator navigates to the order management section.   
2. The administrator selects an order from the list of orders to update its status.   
3. The system displays the order details, including the current status.   
4. The administrator selects the "Update Status" option.   
5. The system presents a dropdown or list of valid status options (e.g., Processing, Shipped, Delivered, Canceled).   
6. The administrator chooses a new status from the available options.   
7. The administrator confirms the status change.   
8. The system validates the selected status and verifies the administrator's authorization.   
9. The system updates the order status in the database.   
10. The system sends an email notification to the customer with the updated status.   
11. The system displays a success message to the administrator.   
  
Alternative Flow:   
1. If the administrator selects an invalid or unsupported status, the system displays an error message and prompts for a valid status.   
2. If the system fails to update the order status, an error message is shown, and the administrator is advised to try again or contact support.   
3. If the email notification fails to send, the system logs the failure and notifies the administrator to manually inform the customer.   
4. If the administrator attempts to update the status without proper authorization, access is denied, and a security message is shown.  
  
Use Case Name: Add Payment Method   
Use Case ID: UC-25   
Actors: Customer, Administrator   
Preconditions:   
1. The system is operational.   
2. The customer is logged into the system.   
3. The payment method management feature is accessible in the account section.   
4. The customer has not already added the payment method to be added.   
  
Postconditions:   
1. A new payment method is added to the customer's account.   
2. The payment method is stored securely in the system.   
3. The customer receives a confirmation message that the payment method was successfully added.   
4. The payment method becomes available for selection during checkout.   
  
Main Flow:   
1. The customer navigates to the account or payment method section.   
2. The customer selects the "Add Payment Method" option.   
3. The system displays a form for entering payment method details.   
4. The customer fills in the required payment information (e.g., credit card number, expiration date, CVV, cardholder name).   
5. The customer submits the form.   
6. The system validates the payment details (e.g., card number format, expiration date in the future, valid CVV).   
7. The system securely stores the payment method in the customer's profile.   
8. The system displays a success message confirming the payment method was added.   
9. The customer is redirected to the payment method management page or homepage.   
  
Alternative Flow:   
1. If the payment method already exists in the customer's profile, the system displays an error message and prompts the customer to choose a different method or confirms the existing one.   
2. If the payment information is invalid (e.g., expired card, incorrect CVV), the system provides feedback and asks the customer to re-enter the correct details.   
3. If the system fails to save the payment method, the customer is shown an error message and advised to try again or contact support.   
4. If the customer attempts to add a payment method without being logged in, the system redirects them to the login page.   
5. If the administrator adds a new system-wide supported payment method, the system updates the available payment options for all users accordingly.  
  
Use Case Name: Update Payment Method   
Use Case ID: UC-26   
Actors: Customer, Administrator   
Preconditions:   
1. The system is operational.   
2. The customer is logged into the system or the administrator has access to the customer account management interface.   
3. The payment method to be updated exists in the customer's profile or the system's supported payment methods.   
4. The payment method management feature is accessible in the account section or administrator dashboard.   
  
Postconditions:   
1. The selected payment method is updated in the customer's profile or the system's supported payment methods.   
2. The updated payment method is securely stored in the system.   
3. The customer receives a confirmation message that the payment method was successfully updated.   
4. The updated payment method becomes available for selection during checkout.   
  
Main Flow:   
1. The customer navigates to the account or payment method section.   
2. The customer selects the "Update Payment Method" option for an existing method.   
3. The system displays the current payment method details in editable fields.   
4. The customer modifies the necessary payment information (e.g., update card expiration date, CVV, or cardholder name).   
5. The customer clicks the "Save Changes" button.   
6. The system validates the updated payment details (e.g., card number format, expiration date in the future, valid CVV).   
7. The system securely updates the payment method in the customer's profile.   
8. The system displays a success message confirming the payment method was updated.   
9. The customer is redirected to the payment method management page or homepage.   
  
Alternative Flow:   
1. If the updated payment method already exists in the customer's profile, the system displays an error message and prompts the customer to choose a different method or confirms the existing one.   
2. If the payment information is invalid (e.g., expired card, incorrect CVV), the system provides feedback and asks the customer to re-enter the correct details.   
3. If the system fails to save the updated payment method, the customer is shown an error message and advised to try again or contact support.   
4. If the customer is not logged in, the system redirects them to the login page before allowing access to the payment method update feature.   
5. If the administrator updates a system-wide supported payment method, the system updates the available payment options for all users accordingly and logs the change.  
  
Use Case Name: Remove Payment Method   
Use Case ID: UC-27   
Actors: Customer, Administrator   
Preconditions:   
1. The system is operational.   
2. The customer is logged into the system or the administrator has access to the customer account management interface.   
3. The payment method to be removed exists in the customer's profile or the system's supported payment methods.   
4. The payment method management feature is accessible in the account section or administrator dashboard.   
  
Postconditions:   
1. The selected payment method is removed from the customer's profile or the system's supported payment methods.   
2. The system displays a confirmation message that the payment method was successfully removed.   
3. The removed payment method is no longer available for selection during checkout.   
4. If the administrator removes a system-wide method, the change is logged and applied to all users.   
  
Main Flow:   
1. The customer or administrator navigates to the payment method management section.   
2. The user selects the "Remove Payment Method" option for a specific payment method.   
3. The system displays a confirmation prompt to ensure the action is intentional.   
4. The user confirms the removal request.   
5. The system verifies the user's authorization or authentication.   
6. The system removes the selected payment method from the customer's profile or the system's available methods.   
7. The system updates the payment method list and displays a success message.   
8. The user is redirected to the payment method management page or homepage.   
  
Alternative Flow:   
1. If the user cancels the removal request, the system returns to the payment method management page without making changes.   
2. If the selected payment method is the only one available and the system requires at least one payment method, the system displays a warning and prevents removal.   
3. If the system fails to remove the payment method, an error message is shown, and the user is advised to try again or contact support.   
4. If the customer is not logged in, the system redirects them to the login page before allowing access to the remove payment method feature.   
5. If the administrator attempts to remove a system-wide payment method that is currently in use for active orders, the system displays a warning and may prevent deletion to avoid inconsistencies.  
  
Use Case Name: View Payment Method   
Use Case ID: UC-28   
Actors: Customer, Administrator   
Preconditions:   
1. The system is operational.   
2. The customer is logged into the system or the administrator has access to the customer account management interface.   
3. The payment method to be viewed exists in the customer's profile or the system's supported payment methods.   
4. The payment method management feature is accessible in the account section or administrator dashboard.   
  
Postconditions:   
1. The details of the selected payment method are displayed to the customer or administrator.   
2. The session remains active.   
3. The user can review the payment method information for reference or modification.   
  
Main Flow:   
1. The customer navigates to the account or payment method section.   
2. The customer selects the "View Payment Method" option for a specific payment method.   
3. The system retrieves the payment method details from the customer's profile or the system database.   
4. The system displays the payment method information, including the type of payment method, relevant details (e.g., last four digits of the card, expiration date), and status.   
5. The customer reviews the displayed payment method details.   
  
Alternative Flow:   
1. If the selected payment method does not exist or is not accessible, the system displays an error message and redirects the customer to the payment method management page.   
2. If the system fails to retrieve the payment method details, an error message is displayed, and the customer is advised to try again or contact support.   
3. If the administrator views a payment method without proper authorization, access is denied, and a security message is shown.   
4. If the customer is not logged in, the system redirects them to the login page before allowing access to the view payment method feature.  
  
Use Case Name: View Payment   
Use Case ID: UC-29   
Actors: Customer, Administrator   
Preconditions:   
1. The system is operational.   
2. The customer is logged into the system or the administrator has access to the customer account management interface.   
3. The payment to be viewed is associated with an existing order and is stored in the system.   
4. The payment management feature is accessible in the account section or administrator dashboard.   
  
Postconditions:   
1. The payment details are displayed to the customer or administrator.   
2. The session remains active.   
3. The user can review the payment information for reference or reconciliation.   
  
Main Flow:   
1. The customer or administrator navigates to the payment section or order detail page.   
2. The user selects the "View Payment" option for a specific order.   
3. The system retrieves the payment information from the database, including amount, payment method, date, and transaction status.   
4. The system displays the payment details in a clear and structured format.   
5. The user reviews the payment information.   
  
Alternative Flow:   
1. If the selected payment is not associated with a valid order, the system displays an error message and redirects the user to the order list or homepage.   
2. If the system fails to retrieve the payment information, an error message is displayed, and the user is advised to try again or contact support.   
3. If the administrator accesses the payment information without proper authorization, access is denied, and a security message is shown.   
4. If the customer is not logged in, the system redirects them to the login page before allowing access to the view payment feature.  
  
Use Case Name: Update Payment Status   
Use Case ID: UC-30   
Actors: Administrator   
Preconditions:   
1. The system is operational.   
2. The administrator is logged into the system.   
3. The payment to be updated is associated with an existing order and is accessible in the payment management interface.   
4. The payment is in a status that allows modification (e.g., "Pending", "Failed").   
  
Postconditions:   
1. The payment status is updated in the system.   
2. The associated order status is updated based on the new payment status.   
3. The customer is notified of the payment status change via email.   
4. The system logs the update and the reason for the change, if applicable.   
  
Main Flow:   
1. The administrator navigates to the payment management section or the order detail page.   
2. The administrator selects the "Update Payment Status" option for a specific payment.   
3. The system displays the current payment details and a list of valid status options (e.g., "Completed", "Failed", "Refunded", "Pending").   
4. The administrator chooses a new status for the payment.   
5. The administrator provides an optional reason for the status change (e.g., "Payment received", "Transaction declined").   
6. The administrator confirms the update request.   
7. The system validates the selected status and ensures it is applicable to the current payment state.   
8. The system updates the payment status in the database.   
9. The system updates the associated order status based on the new payment status (e.g., "Processing" if payment is "Completed").   
10. The system sends a notification to the customer's email about the updated payment status.   
11. The system displays a success message to the administrator and logs the change.   
  
Alternative Flow:   
1. If the administrator selects an invalid or unsupported status for the payment, the system displays an error message and prompts for a valid status.   
2. If the system fails to update the payment status, an error message is displayed, and the administrator is advised to try again or contact support.   
3. If the email notification fails to send, the system logs the failure, and the administrator is alerted to manually notify the customer.   
4. If the administrator attempts to update a payment status without proper authorization, access is denied, and a security message is shown.   
5. If the payment is already in a final state (e.g., "Completed", "Refunded"), the system prevents the update and displays a message explaining that the status cannot be changed.  
  
Use Case Name: Assign Product to Category   
Use Case ID: UC-31   
Actors: Administrator   
Preconditions:   
1. The system is operational.   
2. The administrator is logged into the system.   
3. The product to be assigned exists in the system.   
4. The category to which the product will be assigned exists in the system.   
5. The product is not already assigned to the selected category.   
  
Postconditions:   
1. The product is successfully associated with the selected category.   
2. The updated category includes the newly assigned product in its product list.   
3. The product catalog reflects the updated category associations.   
4. A confirmation message is displayed to the administrator.   
  
Main Flow:   
1. The administrator navigates to the product management section.   
2. The administrator selects a specific product from the product catalog.   
3. The system displays the product details and an option to assign it to a category.   
4. The administrator selects the "Assign to Category" option.   
5. The system displays a list of available categories for assignment.   
6. The administrator chooses a category from the list.   
7. The administrator confirms the assignment request.   
8. The system validates the selected category and ensures the product is not already assigned to it.   
9. The system updates the product's category association in the database.   
10. The system displays a success message and updates the product and category views accordingly.   
  
Alternative Flow:   
1. If the product is already assigned to the selected category, the system displays a message indicating the existing association and offers to assign to a different category.   
2. If the selected category does not exist, the system displays an error message and prompts the administrator to choose a valid category.   
3. If the system fails to update the category association, an error message is shown, and the administrator is advised to try again or contact support.   
4. If the administrator attempts to assign a product without proper authorization, access is denied, and a security message is shown.  
  
Use Case Name: Remove Product from Category   
Use Case ID: UC-32   
Actors: Administrator   
Preconditions:   
1. The system is operational.   
2. The administrator is logged into the system.   
3. The product to be removed is associated with the selected category in the system.   
4. The product exists in the product catalog.   
5. The category exists in the system and is accessible in the category management interface.   
  
Postconditions:   
1. The product is no longer associated with the selected category.   
2. The updated category list no longer includes the removed product.   
3. The product catalog reflects the updated category associations.   
4. A confirmation message is displayed to the administrator.   
  
Main Flow:   
1. The administrator navigates to the category management section.   
2. The administrator selects a specific category from the category list.   
3. The system displays the category details, including the list of associated products.   
4. The administrator selects the "Remove Product from Category" option for a specific product.   
5. The system prompts for confirmation to ensure the action is intentional.   
6. The administrator confirms the removal request.   
7. The system verifies the administrator's authorization.   
8. The system updates the category-product association in the database, removing the product from the category.   
9. The system displays a success message and updates the category and product views accordingly.   
  
Alternative Flow:   
1. If the product is not associated with the selected category, the system displays an error message and prevents the removal.   
2. If the administrator cancels the removal request, the system returns to the category details page without making changes.   
3. If the system fails to update the category-product association, an error message is displayed, and the administrator is advised to try again or contact support.   
4. If the administrator attempts to remove a product from a category without proper authorization, access is denied, and a security message is shown.