项目文档

# Functional Requirement

### 1.1 Customer Registration Function  
\*\*Function ID:\*\* FR-01   
\*\*Description:\*\* Allows new customers to create accounts by providing personal information. Validates input data and sends confirmation messages.   
\*\*Input:\*\* Customer name, email, password, address (registration form fields).   
\*\*Output:\*\* New customer record stored in the Customer entity (CustomerID, Name, Email, Password, RegistrationDate).  
  
### 1.2 Customer Login Function  
\*\*Function ID:\*\* FR-02   
\*\*Description:\*\* Authenticates registered customers using credentials to grant system access.   
\*\*Input:\*\* Registered email and password.   
\*\*Output:\*\* Authentication status (success/failure), and upon success, active user session with dashboard redirect.  
  
### 1.3 Customer Logout Function  
\*\*Function ID:\*\* FR-03   
\*\*Description:\*\* Terminates active customer sessions securely.   
\*\*Input:\*\* Active session token.   
\*\*Output:\*\* Session termination and redirection to login/homepage.  
  
### 1.4 Administrator Login Function  
\*\*Function ID:\*\* FR-04   
\*\*Description:\*\* Authenticates administrators using credentials to grant access to admin controls.   
\*\*Input:\*\* Administrator username and password.   
\*\*Output:\*\* Authentication status (success/failure), and upon success, active admin session with dashboard redirect.  
  
### 1.5 Administrator Logout Function  
\*\*Function ID:\*\* FR-05   
\*\*Description:\*\* Terminates active administrator sessions securely.   
\*\*Input:\*\* Active admin session token.   
\*\*Output:\*\* Session termination and redirection to login/homepage.  
  
### 1.6 Customer Profile Management Function  
\*\*Function ID:\*\* FR-06   
\*\*Description:\*\* Enables customers to view or update personal details (e.g., name, address).   
\*\*Input:\*\* Modified customer attributes (e.g., updated name/address).   
\*\*Output:\*\* Updated Customer entity record in the database.  
  
### 1.7 Customer Account Deletion Function  
\*\*Function ID:\*\* FR-07   
\*\*Description:\*\* Permanently removes customer accounts and associated data (e.g., orders, payment methods).   
\*\*Input:\*\* Account deletion confirmation.   
\*\*Output:\*\* Removal of Customer record and cascaded deletion of dependent entities (ShoppingCart, PaymentMethod, Order).  
  
### 1.8 Product Creation Function  
\*\*Function ID:\*\* FR-08   
\*\*Description:\*\* Allows administrators to add new products with details (name, price, description).   
\*\*Input:\*\* Product name, description, price, stock, image URL.   
\*\*Output:\*\* New Product record stored in the database.  
  
### 1.9 Product Update Function  
\*\*Function ID:\*\* FR-09   
\*\*Description:\*\* Enables administrators to modify existing product attributes.   
\*\*Input:\*\* Modified product fields (e.g., price, stock).   
\*\*Output:\*\* Updated Product record in the database.  
  
### 1.10 Product Deletion Function  
\*\*Function ID:\*\* FR-10   
\*\*Description:\*\* Removes products from the system if not associated with active orders.   
\*\*Input:\*\* Product deletion confirmation.   
\*\*Output:\*\* Removal of Product record from the database.  
  
### 1.11 Category Creation Function  
\*\*Function ID:\*\* FR-11   
\*\*Description:\*\* Allows administrators to define new product categories.   
\*\*Input:\*\* Category name and description.   
\*\*Output:\*\* New Category record stored in the database.  
  
### 1.12 Category Update Function  
\*\*Function ID:\*\* FR-12   
\*\*Description:\*\* Enables administrators to modify category details.   
\*\*Input:\*\* Modified category fields (e.g., name, description).   
\*\*Output:\*\* Updated Category record in the database.  
  
### 1.13 Category Deletion Function  
\*\*Function ID:\*\* FR-13   
\*\*Description:\*\* Removes unused categories not linked to products.   
\*\*Input:\*\* Category deletion confirmation.   
\*\*Output:\*\* Removal of Category record from the database.  
  
### 1.14 Product-Category Assignment Function  
\*\*Function ID:\*\* FR-14   
\*\*Description:\*\* Associates products with categories via the ProductCategory mapping entity.   
\*\*Input:\*\* ProductID and CategoryID.   
\*\*Output:\*\* New ProductCategory relationship record in the database.  
  
### 1.15 Cart Item Addition Function  
\*\*Function ID:\*\* FR-15   
\*\*Description:\*\* Adds products to a customer’s shopping cart with specified quantities.   
\*\*Input:\*\* ProductID and quantity.   
\*\*Output:\*\* New CartItem record linked to ShoppingCart in the database.  
  
### 1.16 Cart Item Modification Function  
\*\*Function ID:\*\* FR-16   
\*\*Description:\*\* Updates quantities or removes items from a shopping cart.   
\*\*Input:\*\* CartItemID and new quantity (or removal flag).   
\*\*Output:\*\* Updated CartItem record or removal from the database.  
  
### 1.17 Order Placement Function  
\*\*Function ID:\*\* FR-17   
\*\*Description:\*\* Converts shopping cart items into a confirmed order, deducts inventory, and clears cart.   
\*\*Input:\*\* Shipping address, selected PaymentMethodID, cart contents.   
\*\*Output:\*\* New Order and OrderItem records; updated Product stock; cleared ShoppingCart.  
  
### 1.18 Order Retrieval Function  
\*\*Function ID:\*\* FR-18   
\*\*Description:\*\* Returns order lists filtered by user role (customer sees own orders; admin sees all).   
\*\*Input:\*\* User role (customer/admin) and optional filters (status/date).   
\*\*Output:\*\* List of Order records with summaries (OrderID, date, total, status).  
  
### 1.19 Order Detail View Function  
\*\*Function ID:\*\* FR-19   
\*\*Description:\*\* Fetches detailed information for a specific order.   
\*\*Input:\*\* Valid OrderID.   
\*\*Output:\*\* Complete Order entity data with linked OrderItems, shipping, and payment details.  
  
### 1.20 Order Cancellation Function  
\*\*Function ID:\*\* FR-20   
\*\*Description:\*\* Marks cancellable orders as "Cancelled" and restocks inventory.   
\*\*Input:\*\* OrderID and cancellation confirmation.   
\*\*Output:\*\* Updated Order status ("Cancelled"); restored Product stock quantities.  
  
### 1.21 Order Status Update Function  
\*\*Function ID:\*\* FR-21   
\*\*Description:\*\* Allows administrators to modify order statuses (e.g., "Shipped" to "Delivered").   
\*\*Input:\*\* OrderID and new status.   
\*\*Output:\*\* Updated Order status in the database; optional customer notifications.  
  
### 1.22 Payment Method Addition Function  
\*\*Function ID:\*\* FR-22   
\*\*Description:\*\* Securely stores new payment credentials for a customer.   
\*\*Input:\*\* Payment details (card number, expiry, CVV, type).   
\*\*Output:\*\* New PaymentMethod record linked to Customer.  
  
### 1.23 Payment Method Update Function  
\*\*Function ID:\*\* FR-23   
\*\*Description:\*\* Modifies existing payment credentials (e.g., expiration date).   
\*\*Input:\*\* PaymentMethodID and updated fields.   
\*\*Output:\*\* Updated PaymentMethod record in the database.  
  
### 1.24 Payment Method Removal Function  
\*\*Function ID:\*\* FR-24   
\*\*Description:\*\* Deletes stored payment methods not tied to active orders.   
\*\*Input:\*\* PaymentMethodID and deletion confirmation.   
\*\*Output:\*\* Removal of PaymentMethod record from the database.  
  
### 1.25 Product Browsing Function  
\*\*Function ID:\*\* FR-25   
\*\*Description:\*\* Returns products filtered by categories or keywords.   
\*\*Input:\*\* CategoryID or search keyword (optional).   
\*\*Output:\*\* List of Product records with details (name, price, description, stock).  
  
### 1.26 Cart View Function  
\*\*Function ID:\*\* FR-26   
\*\*Description:\*\* Displays current shopping cart contents with item quantities and prices.   
\*\*Input:\*\* Active ShoppingCartID.   
\*\*Output:\*\* List of CartItem records with linked Product data and total cost.

# External Description

### Chapter 2: External Interfaces   
Based on functional requirements analysis, the system interacts with the following external interfaces:   
  
#### 2.1 User Interfaces   
- \*\*Web Application Interface\*\*   
 - \*Description\*: Browser-based UI for user interactions.   
 - \*Roles\*:   
 - \*Customer Portal\*: Handles registration/login (FR-01, FR-02), profile management (FR-06), product browsing (FR-25), cart operations (FR-15, FR-16, FR-26), order placement/viewing (FR-17, FR-18, FR-19), and payment method management (FR-22, FR-23, FR-24).   
 - \*Admin Portal\*: Manages product/category CRUD (FR-08–FR-13), order status updates (FR-21), and administrator login/logout (FR-04, FR-05).   
 - \*Interactions\*: HTML/CSS/JavaScript frontend. Inputs mirror functional requirements (e.g., registration forms, cart modification inputs). Outputs include dashboards, confirmation prompts, and data display.   
  
#### 2.2 Hardware Interfaces   
\*Not applicable.\*   
- No hardware-specific interactions identified in functional requirements.   
  
#### 2.3 Software Interfaces   
- \*\*Relational Database System\*\*   
 - \*Description\*: Central datastore for all entities (Customer, Product, Category, Order, etc.).   
 - \*Roles\*:   
 - \*Data Persistence\*: Stores/retrieves records for all CRUD operations (e.g., FR-01 output to `Customer` entity; FR-17 updates `Order`, `OrderItem`, and `Product` tables).   
 - \*Constraints Enforcement\*: Cascades deletions (FR-07, FR-10, FR-13) and validates relationships (e.g., FR-14’s `ProductCategory` mapping).   
 - \*Interactions\*: SQL queries via ORM/DAO layer. Inputs: Structured data per FR specifications. Outputs: Stored records or transaction status.   
  
- \*\*Third-Party Payment Gateway API\*\*   
 - \*Description\*: External service for payment processing (implied by FR-17, FR-22–FR-24).   
 - \*Roles\*: Securely processes payments during order placement (FR-17) and validates payment credentials (FR-22, FR-23).   
 - \*Interactions\*: REST API calls. Inputs: Encrypted payment details. Outputs: Authorization status (success/failure).   
  
#### 2.4 Communication Interfaces   
- \*\*Email/SMS Notification Service\*\*   
 - \*Description\*: External messaging platform for user notifications.   
 - \*Roles\*:   
 - Sends registration confirmations (FR-01 output).   
 - Delivers order status updates (FR-21 output).   
 - \*Interactions\*: API integration (e.g., SMTP/Twilio). Inputs: User contact info, message templates. Outputs: Delivery receipts.   
  
- \*\*Session Management Protocol\*\*   
 - \*Description\*: HTTP-based session handling via tokens.   
 - \*Roles\*: Manages user authentication states (FR-02/FR-04 login outputs; FR-03/FR-05 logout inputs).   
 - \*Interactions\*: Cookie/token exchanges. Inputs: Credentials; Outputs: Session IDs with redirects.   
  
---  
  
### Cross-Check Validation   
- \*\*Database References\*\*: All FR outputs (e.g., `Customer`, `Product`, `Order` entities) map to §2.3’s database interface.   
- \*\*External Services\*\*:   
 - Payment processing (FR-17/FR-22) → §2.3 Payment Gateway API.   
 - Notifications (FR-01/FR-21) → §2.4 Email/SMS Service.   
- \*\*UI Consistency\*\*: All input forms and output displays align with §2.1’s Web Application Interface.   
  
\*Terminology note\*: Entity names (e.g., `ShoppingCart`, `PaymentMethod`) and IDs (e.g., `OrderID`) maintain consistency with functional requirements.

# Use Case

Use Case Name: Register Customer   
Use Case ID: UC-01   
Actors: Customer, Administrator   
Preconditions:   
1. The system is operational and accessible.   
2. The Customer has not yet registered in the system.   
  
Postconditions:   
1. The Customer is successfully registered in the system.   
2. The system sends a confirmation message to the Customer.   
  
Main Flow:   
1. The Customer accesses the registration page of the system.   
2. The system displays the registration form.   
3. The Customer fills in the required information (e.g., name, email, password, address).   
4. The Customer submits the registration form.   
5. The system validates the input data (e.g., checks for valid email format and password strength).   
6. The system confirms the registration and creates a new Customer account.   
7. The system sends a confirmation message to the Customer's email.   
8. The use case ends.   
  
Alternative Flow:   
1. If the Customer provides invalid or incomplete data in step 3, the system displays an error message and prompts the Customer to correct the information.   
2. If the email address is already registered in the system, the system displays an error message and informs the Customer that the email is already in use.   
3. If the system fails to send the confirmation email in step 7, the system displays an error message and allows the Customer to request a resend.  
  
Use Case Name: Login Customer   
Use Case ID: UC-02   
Actors: Customer, System   
Preconditions:   
1. The system is operational and accessible.   
2. The Customer has already registered in the system.   
3. The Customer has a valid account with correct login credentials.   
  
Postconditions:   
1. The Customer is successfully logged into the system.   
2. The system displays the Customer's dashboard or homepage.   
  
Main Flow:   
1. The Customer navigates to the login page of the system.   
2. The system displays the 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 data.   
6. If the credentials are valid, the system logs the Customer in and redirects them to their dashboard or homepage.   
7. The use case ends.   
  
Alternative Flow:   
1. If the Customer enters an invalid email or password in step 3, the system displays an error message and prompts the Customer to try again.   
2. If the Customer enters the wrong password multiple times, the system locks the account temporarily and notifies the Customer.   
3. If the system is unable to authenticate the Customer in step 5, the system displays an error and logs the failed attempt for security review.  
  
Use Case Name: Logout Customer   
Use Case ID: UC-03   
Actors: Customer, System   
Preconditions:   
1. The system is operational and accessible.   
2. The Customer is currently logged into the system.   
  
Postconditions:   
1. The Customer's session is terminated.   
2. The system redirects the Customer to the login or homepage.   
  
Main Flow:   
1. The Customer navigates to the account or logout section of the system.   
2. The system displays the logout option.   
3. The Customer clicks on the logout button.   
4. The system verifies the Customer's session and logs them out.   
5. The system redirects the Customer to the login or homepage.   
6. The use case ends.   
  
Alternative Flow:   
1. If the logout action fails due to a system error in step 4, the system displays an error message and prompts the Customer to try again.   
2. If the Customer has active processes (e.g., a checkout session) in progress, the system may prompt for confirmation before logging out.  
  
Use Case Name: Manage Customer Profile   
Use Case ID: UC-04   
Actors: Customer, Administrator   
Preconditions:   
1. The system is operational and accessible.   
2. The Customer is logged into the system.   
3. The Customer has an existing profile in the system.   
  
Postconditions:   
1. The Customer's profile information is updated or viewed.   
2. The Administrator may access and modify Customer data if authorized.   
  
Main Flow:   
1. The Customer navigates to their profile section in the system.   
2. The system displays the Customer's current profile information.   
3. The Customer selects an option to edit or view their profile.   
4. The system allows the Customer to update personal details (e.g., name, address, contact information).   
5. The Customer submits the changes.   
6. The system validates the updated information.   
7. The system updates the Customer profile in the database.   
8. The system displays a confirmation message to the Customer.   
9. The use case ends.   
  
Alternative Flow:   
1. If the Customer enters invalid or incomplete data during the update, the system displays an error message and prompts them to correct the information.   
2. If the Administrator accesses the Customer profile, they may edit or deactivate the account, depending on their permissions.   
3. If the system fails to update the profile in step 7, it displays an error message and allows the Customer to try again.   
4. If the Customer attempts to modify restricted fields (e.g., email already in use by another account), the system prevents the change and displays an appropriate error message.  
  
Use Case Name: Delete Customer Account   
Use Case ID: UC-05   
Actors: Customer, Administrator   
Preconditions:   
1. The system is operational and accessible.   
2. The Customer has an existing account in the system.   
3. The Customer is logged in or the Administrator is authenticated with appropriate permissions.   
  
Postconditions:   
1. The Customer account is successfully deleted from the system.   
2. All associated data (e.g., shopping cart, order history) is removed or archived.   
3. The system confirms the deletion to the Customer or Administrator.   
  
Main Flow:   
1. The Customer or Administrator navigates to the account management or user settings section.   
2. The system displays an option to delete the Customer account.   
3. The Customer or Administrator selects the delete account option.   
4. The system prompts for confirmation to proceed with the deletion.   
5. The Customer or Administrator confirms the deletion request.   
6. The system verifies the account ownership or administrative authority.   
7. The system deletes the Customer account and associated data.   
8. The system displays a confirmation message indicating the successful deletion.   
9. The use case ends.   
  
Alternative Flow:   
1. If the Customer or Administrator cancels the deletion in step 4, the system returns to the previous page without making any changes.   
2. If the system fails to verify account ownership or administrative authority in step 6, it displays an error message and denies the deletion request.   
3. If the deletion of associated data (e.g., orders) fails in step 7, the system displays an error message and may roll back the deletion to preserve data integrity.   
4. If the system is unable to delete the account due to external constraints (e.g., account is linked to a paid subscription), it displays an error message and provides guidance on how to resolve the issue before proceeding.  
  
Use Case Name: Login Administrator   
Use Case ID: UC-06   
Actors: Administrator, System   
Preconditions:   
1. The system is operational and accessible.   
2. The Administrator has already registered in the system.   
3. The Administrator has valid login credentials.   
  
Postconditions:   
1. The Administrator is successfully logged into the system.   
2. The system displays the Administrator's dashboard or control panel.   
  
Main Flow:   
1. The Administrator navigates to the administrator login page of the system.   
2. The system displays the login form for administrators.   
3. The Administrator enters their registered username and password.   
4. The Administrator submits the login form.   
5. The system verifies the username and password against the stored Administrator data.   
6. If the credentials are valid, the system logs the Administrator in and redirects them to the admin dashboard.   
7. The use case ends.   
  
Alternative Flow:   
1. If the Administrator enters an invalid username or password in step 3, the system displays an error message and prompts them to try again.   
2. If the Administrator enters the wrong password multiple times, the system locks the account temporarily and notifies the Administrator.   
3. If the system is unable to authenticate the Administrator in step 5, it displays an error and logs the failed attempt for security monitoring.  
  
Use Case Name: Logout Administrator   
Use Case ID: UC-07   
Actors: Administrator, System   
Preconditions:   
1. The system is operational and accessible.   
2. The Administrator is currently logged into the system.   
  
Postconditions:   
1. The Administrator's session is terminated.   
2. The system redirects the Administrator to the login or homepage.   
  
Main Flow:   
1. The Administrator navigates to the account or logout section of the system.   
2. The system displays the logout option.   
3. The Administrator clicks on the logout button.   
4. The system verifies the Administrator's session and logs them out.   
5. The system redirects the Administrator to the login or homepage.   
6. The use case ends.   
  
Alternative Flow:   
1. If the logout action fails due to a system error in step 4, the system displays an error message and prompts the Administrator to try again.   
2. If the Administrator has active administrative tasks (e.g., editing a product or managing an order) in progress, the system may prompt for confirmation before logging out.  
  
Use Case Name: Browse Products   
Use Case ID: UC-08   
Actors: Customer, System   
Preconditions:   
1. The system is operational and accessible.   
2. The Customer is logged into the system or is browsing as a guest.   
3. The Product and Category data entities are populated in the system.   
  
Postconditions:   
1. The Customer views a list of available Products.   
2. The Customer may filter or search for Products based on Category or other criteria.   
3. The Customer may add Products to their Shopping Cart.   
  
Main Flow:   
1. The Customer navigates to the product browsing section of the system.   
2. The system displays a list of Products or Categories.   
3. The Customer selects a Category to filter the Products.   
4. The system shows the Products within the selected Category.   
5. The Customer views the details of a specific Product (e.g., description, price, availability).   
6. The Customer may add the Product to their Shopping Cart.   
7. The system confirms the addition of the Product to the Shopping Cart.   
8. The use case ends.   
  
Alternative Flow:   
1. If no Products are available in the selected Category, the system displays a message indicating that no Products match the criteria.   
2. If the Customer searches for a Product using a keyword, the system displays relevant Products or a "No Results Found" message if none match.   
3. If the system fails to load Product data in step 2 or step 4, it displays an error message and allows the Customer to retry or contact support.   
4. If the Customer attempts to add a Product that is out of stock to their Shopping Cart, the system displays an error message and prevents the addition.  
  
Use Case Name: Add to Cart   
Use Case ID: UC-09   
Actors: Customer, System   
Preconditions:   
1. The system is operational and accessible.   
2. The Customer is logged into the system or is browsing as a guest.   
3. The Product the Customer wants to add is available in the system.   
  
Postconditions:   
1. The selected Product is added to the Customer's Shopping Cart.   
2. The Shopping Cart is updated in the system.   
3. The system provides feedback to the Customer confirming the addition.   
  
Main Flow:   
1. The Customer browses the Products in the system.   
2. The Customer selects a Product to add to their Shopping Cart.   
3. The system prompts the Customer to specify the quantity.   
4. The Customer enters the desired quantity and confirms the addition.   
5. The system checks the Product's availability and inventory.   
6. The system adds the Product with the specified quantity to the Shopping Cart.   
7. The system updates the Shopping Cart display to reflect the new addition.   
8. The system provides a confirmation message to the Customer.   
9. The use case ends.   
  
Alternative Flow:   
1. If the Customer enters an invalid or non-numeric quantity in step 3, the system displays an error message and prompts the Customer to enter a valid quantity.   
2. If the requested quantity exceeds the available inventory in step 5, the system displays an error message and allows the Customer to adjust the quantity.   
3. If the system fails to update the Shopping Cart in step 6, it displays an error message and allows the Customer to retry the action.   
4. If the Customer is browsing as a guest, the system may prompt them to log in or create an account to save the Shopping Cart for future sessions.  
  
Use Case Name: Manage Cart   
Use Case ID: UC-10   
Actors: Customer, System   
Preconditions:   
1. The system is operational and accessible.   
2. The Customer is logged into the system or is browsing as a guest.   
3. The Shopping Cart exists in the system (either as a new or existing cart).   
  
Postconditions:   
1. The Customer can view, add, remove, or update items in their Shopping Cart.   
2. The Shopping Cart is updated in the system.   
3. The system provides confirmation or error messages as needed.   
  
Main Flow:   
1. The Customer navigates to the Shopping Cart section of the system.   
2. The system displays the current contents of the Shopping Cart, including product names, quantities, and prices.   
3. The Customer selects an item to remove or update the quantity.   
4. The system processes the change and updates the Shopping Cart.   
5. The system recalculates the total cost and displays the updated Shopping Cart.   
6. The Customer may choose to continue shopping or proceed to checkout.   
7. The use case ends.   
  
Alternative Flow:   
1. If the Customer tries to add the same Product again, the system increases the quantity of that item instead of creating a new entry.   
2. If the requested quantity exceeds the available inventory, the system displays an error message and allows the Customer to adjust the quantity.   
3. If the system fails to update the Shopping Cart in step 4, it displays an error message and allows the Customer to retry the action.   
4. If the Customer is browsing as a guest and logs out, the Shopping Cart may be saved temporarily or lost unless they log back in with the same session or account.  
  
Use Case Name: Place Order   
Use Case ID: UC-11   
Actors: Customer, System, Administrator   
Preconditions:   
1. The system is operational and accessible.   
2. The Customer is logged into the system.   
3. The Shopping Cart contains at least one valid Product.   
4. The Customer has valid payment and shipping information.   
  
Postconditions:   
1. The Customer successfully places an Order.   
2. The Shopping Cart is cleared of the ordered items.   
3. The system generates a confirmation message for the Customer.   
4. The Order is recorded in the system for tracking and management.   
  
Main Flow:   
1. The Customer navigates to the checkout page.   
2. The system displays the items in the Shopping Cart with their quantities and prices.   
3. The Customer reviews the items and confirms the purchase.   
4. The system prompts the Customer to enter or confirm their shipping address.   
5. The Customer selects or enters their preferred shipping address.   
6. The system prompts the Customer to select a payment method.   
7. The Customer selects a payment method and submits the payment details.   
8. The system validates the payment and shipping information.   
9. The system processes the Order and deducts the quantities from the Product inventory.   
10. The system confirms the Order and provides an Order ID and summary.   
11. The system clears the Shopping Cart.   
12. The use case ends.   
  
Alternative Flow:   
1. If the Customer cancels the checkout in step 3, the system returns to the Shopping Cart page without processing the Order.   
2. If the shipping address is invalid or incomplete in step 4, the system displays an error message and prompts the Customer to correct the information.   
3. If the selected payment method is invalid or the payment fails in step 7, the system displays an error message and allows the Customer to try again with a different method.   
4. If the system fails to validate the payment or shipping information in step 8, it displays an error message and prevents the Order from being processed.   
5. If the inventory is insufficient to fulfill the Order in step 9, the system displays an error message and allows the Customer to adjust the quantities or remove items.   
6. If the Order is successfully placed but the system fails to clear the Shopping Cart in step 11, the system displays an error message and allows the Customer to manually clear the cart or contact support.  
  
Use Case Name: View Order History   
Use Case ID: UC-12   
Actors: Customer, System   
Preconditions:   
1. The system is operational and accessible.   
2. The Customer is logged into the system.   
3. The Customer has at least one Order recorded in the system.   
  
Postconditions:   
1. The Customer can view a list of their previous Orders.   
2. The system displays Order details, including date, status, total amount, and items purchased.   
3. The Customer may filter or sort the Order history.   
  
Main Flow:   
1. The Customer navigates to the "Order History" section in their account.   
2. The system retrieves the Customer's Order records from the database.   
3. The system displays the list of Orders with relevant details (e.g., Order ID, date, total, status).   
4. The Customer selects a specific Order to view more details.   
5. The system shows the Order summary, including items, quantities, and prices.   
6. The Customer may download the Order receipt or view shipping details.   
7. The use case ends.   
  
Alternative Flow:   
1. If no Orders are found for the Customer in step 2, the system displays a message indicating that there is no order history.   
2. If the system fails to retrieve Order data in step 2, it displays an error message and allows the Customer to retry or contact support.   
3. If the Customer attempts to access another user's Order history, the system denies access and displays an error message.   
4. If the Customer filters the Order history by date or status and no matching Orders are found, the system displays a message stating that no results match the criteria.  
  
Use Case Name: View Order Details   
Use Case ID: UC-13   
Actors: Customer, Administrator, System   
Preconditions:   
1. The system is operational and accessible.   
2. The Customer is logged into the system or the Administrator is authenticated with appropriate permissions.   
3. An Order exists in the system with a unique Order ID.   
  
Postconditions:   
1. The Customer or Administrator can view detailed information about a specific Order.   
2. The system displays the Order items, total amount, shipping address, and current status.   
3. The Customer or Administrator may take further actions based on the Order status (e.g., cancel, track, or leave feedback).   
  
Main Flow:   
1. The Customer or Administrator navigates to the Order History or Order Management section.   
2. The system displays a list of Orders with Order IDs and summaries.   
3. The Customer or Administrator selects a specific Order to view its details.   
4. The system retrieves the selected Order and its associated items, shipping, and payment details from the database.   
5. The system displays the Order details, including items, quantities, prices, total amount, shipping address, and Order status.   
6. The use case ends.   
  
Alternative Flow:   
1. If the selected Order ID is invalid or does not exist in step 3, the system displays an error message and prompts the Customer or Administrator to select a valid Order.   
2. If the system fails to retrieve the Order details in step 4, it displays an error message and allows the Customer or Administrator to retry the action or contact support.   
3. If the Customer is viewing an Order that is in a restricted state (e.g., pending payment), the system may display a warning or prevent certain actions.   
4. If the Administrator views an Order, they may have additional options to update the Order status or view customer information for support purposes.  
  
Use Case Name: Cancel Order   
Use Case ID: UC-14   
Actors: Customer, Administrator, System   
Preconditions:   
1. The system is operational and accessible.   
2. The Customer is logged into the system.   
3. The Order exists in the system and is in a cancellable state (e.g., pending or processing).   
4. The Customer is the owner of the Order or the Administrator has appropriate permissions.   
  
Postconditions:   
1. The Order is marked as cancelled in the system.   
2. The associated Products are returned to inventory if applicable.   
3. The system provides confirmation of the cancellation to the Customer or Administrator.   
  
Main Flow:   
1. The Customer or Administrator navigates to the Order History or Order Management section.   
2. The system displays a list of Orders with their current statuses.   
3. The Customer or Administrator selects an Order to cancel.   
4. The system prompts for confirmation to proceed with the cancellation.   
5. The Customer or Administrator confirms the cancellation.   
6. The system checks the Order status to determine if cancellation is allowed.   
7. If cancellation is allowed, the system updates the Order status to "Cancelled."   
8. The system returns the quantities of the Products to the inventory if the Order has not been fulfilled.   
9. The system provides a confirmation message to the Customer or Administrator.   
10. The use case ends.   
  
Alternative Flow:   
1. If the selected Order is not in a cancellable state in step 6, the system displays an error message and prevents the cancellation.   
2. If the Customer or Administrator cancels the action in step 4, the system returns to the previous page without making any changes.   
3. If the system fails to update the Order status in step 7, it displays an error message and allows the Customer or Administrator to retry the action.   
4. If the system fails to return the Product quantities to inventory in step 8, it displays an error message and logs the issue for administrative review.   
5. If the Customer attempts to cancel another user's Order, the system denies the action and displays an error message.  
  
Use Case Name: Create Product   
Use Case ID: UC-15   
Actors: Administrator, System   
Preconditions:   
1. The system is operational and accessible.   
2. The Administrator is logged into the system with appropriate permissions.   
3. The Product and Category data entities are properly defined in the system.   
  
Postconditions:   
1. A new Product is successfully created in the system.   
2. The Product is associated with a valid Category.   
3. The system provides feedback to the Administrator confirming the creation.   
  
Main Flow:   
1. The Administrator navigates to the "Create Product" section in the admin panel.   
2. The system displays a Product creation form with required fields (e.g., name, price, description, Category selection).   
3. The Administrator fills in the required information for the new Product.   
4. The Administrator submits the Product creation form.   
5. The system validates the input data (e.g., checks for valid price format, required fields, and existing Category).   
6. The system creates the new Product and associates it with the selected Category.   
7. The system provides a confirmation message to the Administrator.   
8. The use case ends.   
  
Alternative Flow:   
1. If the Administrator provides invalid or incomplete data in step 3, the system displays an error message and prompts them to correct the information.   
2. If the selected Category does not exist or is invalid in step 3, the system displays an error message and allows the Administrator to select a valid Category.   
3. If the system fails to validate or save the new Product in step 5 or 6, it displays an error message and allows the Administrator to retry the action.   
4. If the Product name already exists in the system, the system displays a warning and allows the Administrator to modify the name or proceed with a duplicate entry.  
  
Use Case Name: Manage Product   
Use Case ID: UC-16   
Actors: Administrator, System   
Preconditions:   
1. The system is operational and accessible.   
2. The Administrator is logged into the system with appropriate permissions.   
3. The Product and Category data entities exist in the system.   
  
Postconditions:   
1. The Product is either created, updated, or deleted in the system.   
2. The system provides confirmation of the action to the Administrator.   
3. The Product's status is updated accordingly in the database.   
  
Main Flow:   
1. The Administrator navigates to the "Manage Product" section in the admin panel.   
2. The system displays a list of existing Products with their details (e.g., name, price, Category, status).   
3. The Administrator selects an option to create, update, or delete a Product.   
4. If creating, the system displays the Product creation form. If updating, it displays the selected Product's details in an editable form.   
5. The Administrator fills in or modifies the Product information.   
6. The Administrator submits the form.   
7. The system validates the input data (e.g., price format, required fields, and valid Category).   
8. The system performs the selected action (create, update, or delete) on the Product.   
9. The system provides a confirmation message to the Administrator.   
10. The use case ends.   
  
Alternative Flow:   
1. If the Administrator provides invalid or incomplete data in step 5, the system displays an error message and prompts them to correct the information.   
2. If the selected Category is invalid or does not exist in step 5, the system displays an error message and allows the Administrator to select a valid Category.   
3. If the system fails to validate or process the action in step 7 or 8, it displays an error message and allows the Administrator to retry the action.   
4. If the Product name already exists and the Administrator attempts to create a new one, the system displays a warning and allows the Administrator to modify the name or proceed with a duplicate entry.   
5. If the Administrator attempts to delete a Product that is part of an existing Order, the system displays an error message and prevents deletion to maintain data integrity.  
  
Use Case Name: Create Category   
Use Case ID: UC-17   
Actors: Administrator, System   
Preconditions:   
1. The system is operational and accessible.   
2. The Administrator is logged into the system with appropriate permissions.   
3. The Category data entity is defined in the system.   
  
Postconditions:   
1. A new Category is successfully created in the system.   
2. The Category is added to the database and can be used for organizing Products.   
3. The system provides feedback to the Administrator confirming the creation.   
  
Main Flow:   
1. The Administrator navigates to the "Manage Categories" section in the admin panel.   
2. The system displays an option to create a new Category.   
3. The Administrator selects the "Create Category" option.   
4. The system displays a Category creation form with required fields (e.g., category name, description).   
5. The Administrator fills in the required information for the new Category.   
6. The Administrator submits the Category creation form.   
7. The system validates the input data (e.g., checks for valid name format and required fields).   
8. The system creates the new Category and stores it in the database.   
9. The system provides a confirmation message to the Administrator.   
10. The use case ends.   
  
Alternative Flow:   
1. If the Administrator provides invalid or incomplete data in step 5, the system displays an error message and prompts them to correct the information.   
2. If the category name already exists in the system, the system displays a warning and allows the Administrator to modify the name or proceed with a duplicate entry.   
3. If the system fails to validate or save the new Category in step 7 or 8, it displays an error message and allows the Administrator to retry the action.   
4. If the Administrator cancels the creation in step 5, the system returns to the "Manage Categories" page without making any changes.  
  
Use Case Name: Manage Category   
Use Case ID: UC-18   
Actors: Administrator, System   
Preconditions:   
1. The system is operational and accessible.   
2. The Administrator is logged into the system with appropriate permissions.   
3. The Category data entity is defined in the system.   
  
Postconditions:   
1. The Category is either created, updated, or deleted in the system.   
2. The system provides confirmation of the action to the Administrator.   
3. The Category's status is updated accordingly in the database.   
  
Main Flow:   
1. The Administrator navigates to the "Manage Categories" section in the admin panel.   
2. The system displays a list of existing Categories with their details (e.g., name, description, status).   
3. The Administrator selects an option to create, update, or delete a Category.   
4. If creating, the system displays the Category creation form. If updating, it displays the selected Category's details in an editable form.   
5. The Administrator fills in or modifies the Category information.   
6. The Administrator submits the form.   
7. The system validates the input data (e.g., checks for valid name format and required fields).   
8. The system performs the selected action (create, update, or delete) on the Category.   
9. The system provides a confirmation message to the Administrator.   
10. The use case ends.   
  
Alternative Flow:   
1. If the Administrator provides invalid or incomplete data in step 5, the system displays an error message and prompts them to correct the information.   
2. If the category name already exists and the Administrator attempts to create a new one, the system displays a warning and allows the Administrator to modify the name or proceed with a duplicate entry.   
3. If the system fails to validate or process the action in step 7 or 8, it displays an error message and allows the Administrator to retry the action.   
4. If the Administrator cancels the action in step 5, the system returns to the "Manage Categories" page without making any changes.   
5. If the Administrator attempts to delete a Category that is associated with existing Products, the system displays an error message and prevents deletion to maintain data integrity.  
  
Use Case Name: View Orders   
Use Case ID: UC-19   
Actors: Customer, Administrator, System   
Preconditions:   
1. The system is operational and accessible.   
2. The Customer is logged into the system or the Administrator is authenticated with appropriate permissions.   
3. There are existing Orders in the system.   
  
Postconditions:   
1. The Customer or Administrator can view a list of Orders.   
2. The system displays Order details, such as Order ID, date, total amount, and status.   
3. The Customer or Administrator may filter or sort the Orders as needed.   
  
Main Flow:   
1. The Customer or Administrator navigates to the "Orders" section of the system.   
2. The system retrieves the list of Orders from the database based on the user's role.   
3. The system displays the Orders with relevant details (e.g., Order ID, date, total amount, status).   
4. The Customer or Administrator may select filters (e.g., by date, status) to refine the list.   
5. The system updates the displayed Orders based on the selected filters.   
6. The use case ends.   
  
Alternative Flow:   
1. If no Orders match the filtering criteria in step 4, the system displays a message indicating no results were found.   
2. If the system fails to retrieve the Order data in step 2, it displays an error message and allows the user to retry the action or contact support.   
3. If the user is not authorized to view certain Orders (e.g., a Customer tries to view another Customer's Orders), the system displays an error message and restricts access.   
4. If the user selects an Order to view more details, the system redirects to the "View Order Details" use case (UC-13).  
  
Use Case Name: Update Order Status   
Use Case ID: UC-20   
Actors: Administrator, System   
Preconditions:   
1. The system is operational and accessible.   
2. The Administrator is logged into the system with appropriate permissions.   
3. An Order exists in the system with a unique Order ID.   
4. The Order is in a status that can be updated (e.g., "Processing," "Shipped").   
  
Postconditions:   
1. The Order status is updated in the system.   
2. The system provides confirmation of the status update to the Administrator.   
3. The associated Customer may receive a notification or email regarding the status change.   
  
Main Flow:   
1. The Administrator navigates to the Order Management section in the admin panel.   
2. The system displays a list of Orders with their current status.   
3. The Administrator selects a specific Order to update its status.   
4. The system retrieves the Order and displays its details, including the current status.   
5. The Administrator selects a new status for the Order (e.g., "Shipped," "Delivered").   
6. The Administrator confirms the status update.   
7. The system updates the Order status in the database.   
8. The system provides a confirmation message to the Administrator.   
9. The use case ends.   
  
Alternative Flow:   
1. If the selected Order ID is invalid or does not exist in step 3, the system displays an error message and prompts the Administrator to select a valid Order.   
2. If the new status is not applicable for the selected Order (e.g., attempting to mark as "Delivered" when it is "Pending"), the system displays an error message and prevents the update.   
3. If the system fails to update the Order status in step 7, it displays an error message and allows the Administrator to retry the action.   
4. If the Administrator cancels the update in step 6, the system returns to the Order details page without making any changes.   
5. If the system fails to send the notification to the Customer, it logs the issue and allows the Administrator to manually send the notification or contact support.  
  
Use Case Name: Add Payment Method   
Use Case ID: UC-21   
Actors: Customer, System   
Preconditions:   
1. The system is operational and accessible.   
2. The Customer is logged into the system.   
3. The Payment Method data entity is defined in the system.   
  
Postconditions:   
1. The Customer successfully adds a new Payment Method to their account.   
2. The system stores the Payment Method securely.   
3. The system provides feedback to the Customer confirming the addition.   
  
Main Flow:   
1. The Customer navigates to the "Payment Methods" section in their account settings.   
2. The system displays an option to add a new Payment Method.   
3. The Customer selects the "Add Payment Method" option.   
4. The system displays a form for entering payment details (e.g., card number, expiration date, CVV, cardholder name).   
5. The Customer fills in the required information for the new Payment Method.   
6. The Customer submits the form.   
7. The system validates the input data (e.g., checks for valid card format, expiration date, and CVV).   
8. The system securely stores the Payment Method in the database.   
9. The system provides a confirmation message to the Customer.   
10. The use case ends.   
  
Alternative Flow:   
1. If the Customer provides invalid or incomplete data in step 5, the system displays an error message and prompts them to correct the information.   
2. If the system fails to validate the payment information in step 7, it displays an error message and allows the Customer to retry the action.   
3. If the system fails to store the Payment Method in step 8, it displays an error message and allows the Customer to retry the action.   
4. If the Customer attempts to add a Payment Method that is already associated with their account, the system displays a warning and allows the Customer to confirm or cancel the action.   
5. If the Customer is browsing as a guest and attempts to add a Payment Method, the system displays a prompt to log in or create an account before proceeding.  
  
Use Case Name: Update Payment Method   
Use Case ID: UC-22   
Actors: Customer, System   
Preconditions:   
1. The system is operational and accessible.   
2. The Customer is logged into the system.   
3. The Customer has at least one Payment Method already stored in their account.   
4. The Payment Method data entity is defined in the system.   
  
Postconditions:   
1. The Customer's selected Payment Method is successfully updated in the system.   
2. The system provides confirmation of the update to the Customer.   
3. The updated Payment Method is securely stored in the database.   
  
Main Flow:   
1. The Customer navigates to the "Payment Methods" section in their account settings.   
2. The system displays a list of existing Payment Methods with their details (e.g., card number ending, expiration date, and default status).   
3. The Customer selects a Payment Method to update.   
4. The system displays an editable form with the current details of the selected Payment Method.   
5. The Customer modifies the relevant fields (e.g., update card expiration date, change default payment method).   
6. The Customer submits the updated Payment Method.   
7. The system validates the updated data (e.g., checks for valid card format, expiration date, and ensures no duplicate default methods if applicable).   
8. The system updates the Payment Method in the database.   
9. The system provides a confirmation message to the Customer.   
10. The use case ends.   
  
Alternative Flow:   
1. If the Customer provides invalid or incomplete data in step 5, the system displays an error message and prompts them to correct the information.   
2. If the system fails to validate the updated payment information in step 7, it displays an error message and allows the Customer to retry the action.   
3. If the system fails to update the Payment Method in step 8, it displays an error message and allows the Customer to retry the action.   
4. If the Customer attempts to update a Payment Method that is currently in use for an active Order, the system displays a warning and may restrict certain modifications to ensure consistency.   
5. If the Customer is browsing as a guest and attempts to update a Payment Method, the system displays a prompt to log in or create an account before proceeding.  
  
Use Case Name: Remove Payment Method   
Use Case ID: UC-23   
Actors: Customer, System   
Preconditions:   
1. The system is operational and accessible.   
2. The Customer is logged into the system.   
3. The Customer has at least one Payment Method stored in their account.   
4. The selected Payment Method is not marked as the default method for an active Order.   
  
Postconditions:   
1. The selected Payment Method is successfully removed from the Customer's account.   
2. The system updates the list of Payment Methods and removes the deleted method.   
3. The system provides confirmation of the removal to the Customer.   
  
Main Flow:   
1. The Customer navigates to the "Payment Methods" section in their account settings.   
2. The system displays a list of existing Payment Methods with their details (e.g., card number ending, expiration date, and default status).   
3. The Customer selects a Payment Method to remove.   
4. The system prompts for confirmation to proceed with the removal.   
5. The Customer confirms the removal.   
6. The system checks if the selected Payment Method is the default one for any active Orders.   
7. If not, the system proceeds to delete the Payment Method from the database.   
8. The system updates the list of Payment Methods to reflect the deletion.   
9. The system provides a confirmation message to the Customer.   
10. The use case ends.   
  
Alternative Flow:   
1. If the Customer cancels the removal in step 4, the system returns to the Payment Methods page without making any changes.   
2. If the selected Payment Method is the default one for an active Order in step 6, the system displays a warning message and prevents the removal to ensure the Order can still be processed.   
3. If the system fails to delete the Payment Method in step 7, it displays an error message and allows the Customer to retry the action.   
4. If the system fails to update the list of Payment Methods in step 8, it displays an error message and allows the Customer to refresh the page or contact support.  
  
Use Case Name: Assign Product to Category   
Use Case ID: UC-24   
Actors: Administrator, System   
Preconditions:   
1. The system is operational and accessible.   
2. The Administrator is logged into the system with appropriate permissions.   
3. The Product and Category data entities exist in the system.   
4. The selected Product is already created in the system.   
5. The selected Category is already created in the system.   
  
Postconditions:   
1. The selected Product is successfully assigned to the selected Category.   
2. The system updates the relationship between the Product and Category in the database.   
3. The system provides confirmation to the Administrator that the assignment was completed.   
  
Main Flow:   
1. The Administrator navigates to the "Manage Products" section in the admin panel.   
2. The system displays a list of existing Products with their details (e.g., product name, price, status, and current Category).   
3. The Administrator selects a Product to modify its Category.   
4. The system displays an option to assign the Product to a new Category.   
5. The Administrator selects the desired Category from the list of available Categories.   
6. The Administrator confirms the assignment.   
7. The system updates the Product's Category association in the database.   
8. The system provides a confirmation message to the Administrator indicating the successful assignment.   
9. The use case ends.   
  
Alternative Flow:   
1. If the Administrator selects an invalid Category in step 5, the system displays an error message and allows them to select a valid Category.   
2. If the system fails to update the Category association in step 7, it displays an error message and allows the Administrator to retry the action.   
3. If the Administrator cancels the assignment in step 6, the system returns to the Product details page without making any changes.   
4. If the selected Product is already assigned to the same Category, the system displays a message indicating that the assignment is redundant and allows the Administrator to proceed or cancel.   
5. If the system is unable to retrieve the list of available Categories in step 4, it displays an error message and allows the Administrator to retry or contact support.