项目文档

# Functional Requirement

### 1.1 Customer Registration   
\*\*Function ID\*\*: FR-01   
\*\*Description\*\*: Allows new customers to create accounts by providing personal information. Validates email uniqueness and password complexity. Creates a customer record and sends confirmation email.   
\*\*Input\*\*: Name (string), email (string), password (string)   
\*\*Output\*\*: New customer record stored in Customer entity (CustomerID, Name, Email, Password)   
  
### 1.2 User Login   
\*\*Function ID\*\*: FR-02   
\*\*Description\*\*: Authenticates customers/administrators using email and password. Grants access to respective dashboards upon successful verification.   
\*\*Input\*\*: Email (string), password (string)   
\*\*Output\*\*: Active user session, redirect to customer/admin dashboard   
  
### 1.3 User Logout   
\*\*Function ID\*\*: FR-03   
\*\*Description\*\*: Terminates active user sessions, clears session data, and redirects to the login page.   
\*\*Input\*\*: Session token   
\*\*Output\*\*: Session termination, redirect to login page   
  
### 1.4 Update Customer Information   
\*\*Function ID\*\*: FR-04   
\*\*Description\*\*: Enables customers to modify profile details (e.g., address, phone). Validates field integrity and email uniqueness.   
\*\*Input\*\*: CustomerID (int), updated fields (e.g., Address, Phone)   
\*\*Output\*\*: Modified attributes in Customer entity   
  
### 1.5 View Customer Information   
\*\*Function ID\*\*: FR-05   
\*\*Description\*\*: Displays customer details (e.g., order history, contact info) for administrators or the customer themselves.   
\*\*Input\*\*: CustomerID (int)   
\*\*Output\*\*: Customer entity data (Name, Email, Address, Phone)   
  
### 1.6 Add Product   
\*\*Function ID\*\*: FR-06   
\*\*Description\*\*: Allows administrators to create new product entries with details (name, price, category) and associate them with categories.   
\*\*Input\*\*: Name (string), Description (text), Price (float), StockQuantity (int), CategoryID (int)   
\*\*Output\*\*: New record in Product entity; association in ProductCategory entity   
  
### 1.7 Update Product   
\*\*Function ID\*\*: FR-07   
\*\*Description\*\*: Enables administrators to edit product attributes (e.g., price, description) and update category associations.   
\*\*Input\*\*: ProductID (int), modified fields (e.g., Price, Description, CategoryID)   
\*\*Output\*\*: Updated attributes in Product entity; modified ProductCategory linkage   
  
### 1.8 Delete Product   
\*\*Function ID\*\*: FR-08   
\*\*Description\*\*: Removes a product from the system if no active orders reference it. Prevents deletion for data integrity.   
\*\*Input\*\*: ProductID (int)   
\*\*Output\*\*: Removal of Product entity record   
  
### 1.9 Browse Products   
\*\*Function ID\*\*: FR-09   
\*\*Description\*\*: Lists available products with core details (name, price) for customers/admins.   
\*\*Input\*\*: None (catalog request)   
\*\*Output\*\*: List of Product entities (ProductID, Name, Price)   
  
### 1.10 View Product Details   
\*\*Function ID\*\*: FR-10   
\*\*Description\*\*: Displays comprehensive product information (description, stock, category) when selected.   
\*\*Input\*\*: ProductID (int)   
\*\*Output\*\*: Full Product entity data + linked Category details   
  
### 1.11 Add Category   
\*\*Function ID\*\*: FR-11   
\*\*Description\*\*: Creates new product categories with unique names for product classification.   
\*\*Input\*\*: Name (string), Description (text)   
\*\*Output\*\*: New record in Category entity   
  
### 1.12 Update Category   
\*\*Function ID\*\*: FR-12   
\*\*Description\*\*: Modifies category attributes (name, description) while ensuring name uniqueness.   
\*\*Input\*\*: CategoryID (int), updated fields (Name/Description)   
\*\*Output\*\*: Modified attributes in Category entity   
  
### 1.13 Delete Category   
\*\*Function ID\*\*: FR-13   
\*\*Description\*\*: Removes a category if no products are associated with it.   
\*\*Input\*\*: CategoryID (int)   
\*\*Output\*\*: Removal of Category entity record   
  
### 1.14 View Category Details   
\*\*Function ID\*\*: FR-14   
\*\*Description\*\*: Shows category information and linked products.   
\*\*Input\*\*: CategoryID (int)   
\*\*Output\*\*: Category entity data + list of associated Products   
  
### 1.15 Add to Shopping Cart   
\*\*Function ID\*\*: FR-15   
\*\*Description\*\*: Adds a product to the customer’s cart after stock verification. Updates cart totals.   
\*\*Input\*\*: CustomerID (int), ProductID (int), Quantity (int)   
\*\*Output\*\*: New CartItem record; updated ShoppingCart total   
  
### 1.16 Update Shopping Cart   
\*\*Function ID\*\*: FR-16   
\*\*Description\*\*: Modifies product quantities or removes items from the cart. Recalculates cart totals.   
\*\*Input\*\*: CartItemID (int), new Quantity (int; 0 for removal)   
\*\*Output\*\*: Updated CartItem.Quantity or record deletion; revised ShoppingCart total   
  
### 1.17 View Shopping Cart   
\*\*Function ID\*\*: FR-17   
\*\*Description\*\*: Displays all items in the cart, quantities, prices, and total cost.   
\*\*Input\*\*: CustomerID (int)   
\*\*Output\*\*: List of CartItems + Product details; ShoppingCart total amount   
  
### 1.18 Place Order   
\*\*Function ID\*\*: FR-18   
\*\*Description\*\*: Converts cart items into an order, processes payment, reduces inventory, and sends confirmation.   
\*\*Input\*\*: CustomerID (int), PaymentMethodID (int), shipping/billing details   
\*\*Output\*\*: New Order and OrderItem records; reduced Product.StockQuantity; email confirmation   
  
### 1.19 Browse Orders   
\*\*Function ID\*\*: FR-19   
\*\*Description\*\*: Lists orders (for customers: their own; for admins: all orders).   
\*\*Input\*\*: CustomerID (int, for customers) or none (for admins)   
\*\*Output\*\*: Order summaries (OrderID, OrderDate, TotalAmount)   
  
### 1.20 View Order Details   
\*\*Function ID\*\*: FR-20   
\*\*Description\*\*: Shows comprehensive order information, including items and status.   
\*\*Input\*\*: OrderID (int)   
\*\*Output\*\*: Order entity data + OrderItem list + PaymentMethod details   
  
### 1.21 Cancel Order   
\*\*Function ID\*\*: FR-21   
\*\*Description\*\*: Cancels eligible orders (e.g., not shipped), restocks inventory, and notifies the customer.   
\*\*Input\*\*: OrderID (int)   
\*\*Output\*\*: Order.Status set to "Canceled"; increased Product.StockQuantity   
  
### 1.22 Update Order Status   
\*\*Function ID\*\*: FR-22   
\*\*Description\*\*: Allows admins to change order status (e.g., "Shipped") and notify customers.   
\*\*Input\*\*: OrderID (int), new Status (string)   
\*\*Output\*\*: Updated Order.Status; notification email   
  
### 1.23 Manage Inventory   
\*\*Function ID\*\*: FR-23   
\*\*Description\*\*: Adjusts product stock levels (e.g., after shipments/returns).   
\*\*Input\*\*: ProductID (int), new StockQuantity (int)   
\*\*Output\*\*: Updated Product.StockQuantity   
  
### 1.24 Manage Customer Payment Methods   
\*\*Function ID\*\*: FR-24   
\*\*Description\*\*: Adds/edits/removes customer payment methods (e.g., credit cards).   
\*\*Input\*\*: PaymentMethodID (int, for edit/remove), Type (string), TokenizedCard (string), ExpiryDate (date)   
\*\*Output\*\*: New/updated/deleted PaymentMethod record   
  
### 1.25 Assign Product to Category   
\*\*Function ID\*\*: FR-25   
\*\*Description\*\*: Links a product to a category via the ProductCategory entity.   
\*\*Input\*\*: ProductID (int), CategoryID (int)   
\*\*Output\*\*: New ProductCategory association record   
  
### 1.26 Manage Administrator Accounts   
\*\*Function ID\*\*: FR-26   
\*\*Description\*\*: Adds/edits/deletes administrator accounts with role permissions.   
\*\*Input\*\*: AdminID (int, for edit/delete), Name (string), Email (string), Role (string)   
\*\*Output\*\*: New/updated/deleted Administrator record   
  
### 1.27 Delete Customer   
\*\*Function ID\*\*: FR-27   
\*\*Description\*\*: Removes customer accounts if no active orders exist.   
\*\*Input\*\*: CustomerID (int)   
\*\*Output\*\*: Deleted Customer record + dependent PaymentMethods   
  
### 1.28 Delete Order   
\*\*Function ID\*\*: FR-28   
\*\*Description\*\*: Permanently deletes orders in cancellable states (e.g., "Canceled").   
\*\*Input\*\*: OrderID (int)   
\*\*Output\*\*: Removal of Order and OrderItem records   
  
### 1.29 Update Order Items   
\*\*Function ID\*\*: FR-29   
\*\*Description\*\*: Modifies items in pending orders (e.g., quantity changes), adjusts inventory and order totals.   
\*\*Input\*\*: OrderID (int), OrderItemID (int), new Quantity (int)   
\*\*Output\*\*: Updated OrderItem.Quantity; revised Order.TotalAmount; adjusted Product.StockQuantity

# External Description

### Chapter 2: External Interfaces   
This chapter defines all external interfaces required by the system, classified into four categories based on interaction type. All interfaces referenced in functional requirements (Section 1) are comprehensively defined below.   
  
---  
  
#### \*\*2.1 User Interface Output\*\*   
Defines human-system interaction mechanisms via graphical interfaces.   
  
| \*\*Interface Component\*\* | \*\*Description\*\* | \*\*Interaction Method\*\* |   
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|   
| \*\*Web Application Interface\*\* | Primary interface for customers and administrators. Renders dashboards, forms, and data displays. | - Input: User actions (clicks/form submissions) via browser<br>- Output: HTML/CSS/JS rendering (e.g., dashboards in FR-02, product catalog in FR-09) |   
| \*\*Registration Module\*\* | Captures new customer data with real-time validation (FR-01). | - Input: Name/email/password fields<br>- Output: Success confirmation or error prompts for duplicate email/complexity rules |   
| \*\*Shopping Cart UI\*\* | Displays cart items, prices, and totals (FR-17). Supports quantity updates/removals (FR-16). | - Input: Quantity adjustments via +/- buttons<br>- Output: Dynamically updated cart totals and item list |   
| \*\*Order Management Portal\*\* | Shows order summaries (FR-19) and detailed views (FR-20). Enables status updates (FR-22) and cancellations (FR-21) for admins. | - Input: OrderID selection<br>- Output: Order details (items, status, payment) or batch lists |   
  
---  
  
#### \*\*2.2 Hardware Interface Output\*\*   
Describes interactions with physical devices.   
  
| \*\*Interface Component\*\* | \*\*Description\*\* | \*\*Interaction Method\*\* |   
|-------------------------|---------------------------------------------------|------------------------|   
| \*None required\* | The system operates purely as a web application and does not interface with specialized hardware devices. | N/A |   
  
---  
  
#### \*\*2.3 Software Interface Output\*\*   
Defines integrations with external software systems and data stores.   
  
| \*\*Interface Component\*\* | \*\*Description\*\* | \*\*Interaction Method\*\* |   
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|   
| \*\*Relational Database (RDBMS)\*\* | Central data store for all entities (Customer, Product, Order, etc.). Referenced in 22/29 FRs (e.g., FR-01, FR-06, FR-18). | - \*\*Input\*\*: SQL queries (e.g., `INSERT INTO Customer` for FR-01)<br>- \*\*Output\*\*: Persistent records (e.g., new CustomerID) and query results (e.g., order history in FR-05) |   
| \*\*Payment Gateway API\*\* | Third-party service for processing payments during order placement (FR-18). | - \*\*Input\*\*: Tokenized payment data (via PaymentMethodID), amount<br>- \*\*Output\*\*: Transaction status (success/failure) and authorization codes |   
| \*\*Inventory Management Hook\*\* | External warehouse system integration for real-time stock synchronization (implied by FR-23). | - \*\*Input\*\*: ProductID and new StockQuantity<br>- \*\*Output\*\*: Stock adjustment confirmation |   
  
---  
  
#### \*\*2.4 Communication Interface Output\*\*   
Specifies network-based communication protocols.   
  
| \*\*Interface Component\*\* | \*\*Description\*\* | \*\*Interaction Method\*\* |   
|--------------------------|---------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|   
| \*\*Email Service (SMTP)\*\* | Sends transactional emails for key events (referenced in FR-01, FR-18, FR-21, FR-22). | - \*\*Input\*\*: Recipient email, template (e.g., order confirmation)<br>- \*\*Output\*\*: Email delivery status logs |   
| \*\*Session Token System\*\* | Manages user authentication states via HTTP cookies (used in FR-02, FR-03). | - \*\*Input\*\*: Session token<br>- \*\*Output\*\*: Session creation/deletion (e.g., redirects post-logout in FR-03) |   
  
---  
  
### Cross-Verification Summary   
- \*\*Database\*\*: Covered under \*2.3 Software Interface Output\* (supports 22 FRs including core entities).   
- \*\*Email Service\*\*: Explicitly defined in \*2.4 Communication Interface Output\* (links to FR-01/18/21/22).   
- \*\*Payment Gateway\*\*: Documented in \*2.3 Software Interface Output\* (critical for FR-18).   
- \*\*User Interfaces\*\*: Mapped to web components in \*2.1 User Interface Output\* (all GUI-dependent FRs).   
- \*No omissions identified\*; all external references in Section 1 are addressed.   
  
> Terminology alignment: Consistent use of \*CustomerID\*, \*ProductID\*, \*OrderID\*, and entity names (e.g., \*ShoppingCart\*) from functional requirements.

# Use Case

Use Case Name: Register Customer   
Use Case ID: UC-01   
Actors: Customer, Administrator   
Preconditions: The system is operational, and the customer is not already registered. The administrator has access to the system.   
Postconditions: A new customer account is created and stored in the system. The customer receives a confirmation message.   
  
Main Flow:   
1. The customer accesses the registration page of the system.   
2. The customer enters their personal information, including name, email address, and password.   
3. The system validates the entered email to ensure it is unique.   
4. The system confirms the password meets the required criteria (e.g., length, complexity).   
5. The customer submits the registration form.   
6. The system creates a new customer record in the database.   
7. The system sends a confirmation message to the customer's email.   
8. The registration is completed, and the customer is redirected to the login page.   
  
Alternative Flow:   
1. If the entered email is already registered, the system displays an error message and prevents registration.   
2. If the password does not meet the criteria, the system displays an error and prompts the customer to re-enter a valid password.   
3. If the system fails to send the confirmation email, the administrator is notified to manually verify the registration.  
  
Use Case Name: Login   
Use Case ID: UC-02   
Actors: Customer, Administrator   
Preconditions: The system is operational. The customer or administrator has a valid account. The user is on the login page.   
Postconditions: The user is successfully authenticated and redirected to their respective dashboard.   
  
Main Flow:   
1. The user accesses the login page of the system.   
2. The user enters their email address and password.   
3. The system verifies the entered credentials against the stored records.   
4. The system authenticates the user as either a customer or an administrator.   
5. The user is redirected to their corresponding dashboard.   
  
Alternative Flow:   
1. If the email is not registered, the system displays an error message and prompts the user to check the email.   
2. If the password is incorrect, the system displays an error message and allows the user to re-enter the password.   
3. If the user exceeds the allowed number of login attempts, the system temporarily locks the account and notifies the administrator.  
  
Use Case Name: Logout   
Use Case ID: UC-03   
Actors: Customer, Administrator   
Preconditions: The system is operational. The customer or administrator is currently logged in and has access to their dashboard or user interface.   
Postconditions: The user session is terminated, and the user is redirected to the login page.   
  
Main Flow:   
1. The customer or administrator clicks on the "Logout" button in the system's interface.   
2. The system verifies the user is currently logged in.   
3. The system terminates the active session and clears session data.   
4. The user is redirected to the login page.   
5. The system displays a confirmation message that the user has been successfully logged out.   
  
Alternative Flow:   
1. If the user is not logged in, the system displays an error message indicating that no active session exists.   
2. If the system fails to terminate the session due to a technical error, the user is shown an error message, and the administrator is notified to investigate the issue.  
  
Use Case Name: Update Customer Information   
Use Case ID: UC-04   
Actors: Customer, Administrator   
Preconditions: The system is operational. The customer is logged in. The customer has an existing account in the system. The administrator has access to the system.   
Postconditions: The customer's information is updated in the system. A confirmation message is displayed to the customer.   
  
Main Flow:   
1. The customer navigates to the "Profile" section of their dashboard.   
2. The customer selects the "Edit Information" option.   
3. The system displays the current customer information in a form.   
4. The customer modifies the required fields (e.g., address, phone number).   
5. The customer submits the updated information.   
6. The system validates the new information for accuracy and completeness.   
7. The system updates the customer record in the database.   
8. The system displays a confirmation message that the information has been successfully updated.   
  
Alternative Flow:   
1. If the customer does not have permission to edit certain fields, the system displays an error message and only allows the administrator to modify those fields.   
2. If the updated email address is already registered by another user, the system displays an error message and prevents the update.   
3. If the system fails to update the customer information due to a database error, an error message is shown to the customer, and the administrator is notified to resolve the issue.  
  
Use Case Name: View Customer Information   
Use Case ID: UC-05   
Actors: Administrator, Customer   
Preconditions: The system is operational. The administrator is logged in and has access to the customer management interface. The customer is logged in and has an existing account.   
Postconditions: The requested customer information is displayed. The administrator or customer has access to the relevant details.   
  
Main Flow:   
1. The administrator navigates to the "Customer Management" section or the customer views their own profile.   
2. The system displays a list of customers (for administrator) or the customer's own information (for customer).   
3. The administrator selects a specific customer from the list or the customer views their profile.   
4. The system retrieves the selected customer's information from the database.   
5. The system presents the customer details (e.g., name, email, address, order history) in a user-friendly format.   
  
Alternative Flow:   
1. If no customers are found in the database, the system displays a message indicating the absence of data.   
2. If the selected customer does not exist, the system shows an error message and prompts the administrator to try again.   
3. If the system fails to retrieve the information due to a database error, an error message is shown, and the administrator is notified to investigate.  
  
Use Case Name: Add Product   
Use Case ID: UC-06   
Actors: Administrator   
Preconditions: The system is operational. The administrator is logged in and has access to the product management interface. A valid product category exists in the system.   
Postconditions: A new product is added to the system and stored in the database. The product becomes available for viewing and purchasing.   
  
Main Flow:   
1. The administrator navigates to the "Product Management" section of the dashboard.   
2. The administrator selects the "Add Product" option.   
3. The system displays a form for entering product details.   
4. The administrator fills in the product information, including name, price, description, and selects a category.   
5. The administrator uploads an image of the product.   
6. The administrator submits the form.   
7. The system validates the input data for correctness and completeness.   
8. The system adds the product to the database and associates it with the selected category.   
9. The system displays a confirmation message that the product was successfully added.   
  
Alternative Flow:   
1. If the product name is already taken, the system displays an error message and prevents the addition.   
2. If the price is invalid (e.g., negative or non-numeric), the system shows an error and prompts the administrator to correct the value.   
3. If no category is selected or the category does not exist, the system displays an error message and prevents the addition.   
4. If the image upload fails, the system shows an error and allows the administrator to try again.   
5. If the system fails to store the product in the database, an error message is displayed, and the administrator is notified to resolve the issue.  
  
Use Case Name: Update Product   
Use Case ID: UC-07   
Actors: Administrator   
Preconditions: The system is operational. The administrator is logged in and has access to the product management interface. The product to be updated exists in the system.   
Postconditions: The product details are updated in the system. A confirmation message is displayed to the administrator.   
  
Main Flow:   
1. The administrator navigates to the "Product Management" section of the dashboard.   
2. The administrator selects the "Edit Product" option for the desired product.   
3. The system displays the current product information in an editable form.   
4. The administrator modifies the necessary product details (e.g., name, price, description, category).   
5. The administrator uploads a new product image if needed.   
6. The administrator submits the updated information.   
7. The system validates the input data for correctness and completeness.   
8. The system updates the product record in the database.   
9. The system displays a confirmation message that the product was successfully updated.   
  
Alternative Flow:   
1. If the new product name is already taken by another product, the system displays an error message and prevents the update.   
2. If the price is invalid (e.g., negative or non-numeric), the system shows an error and prompts the administrator to correct the value.   
3. If the selected category does not exist, the system displays an error message and prevents the update.   
4. If the image upload fails, the system shows an error and allows the administrator to try again.   
5. If the system fails to update the product in the database, an error message is displayed, and the administrator is notified to resolve the issue.  
  
Use Case Name: Delete Product   
Use Case ID: UC-08   
Actors: Administrator   
Preconditions: The system is operational. The administrator is logged in and has access to the product management interface. The product to be deleted exists in the database.   
Postconditions: The product is removed from the database. A confirmation message is displayed to the administrator.   
  
Main Flow:   
1. The administrator navigates to the "Product Management" section of the dashboard.   
2. The administrator selects the "Delete Product" option for the desired product.   
3. The system prompts the administrator to confirm the deletion.   
4. The administrator confirms the deletion request.   
5. The system removes the product record from the database.   
6. The system displays a confirmation message that the product was successfully deleted.   
  
Alternative Flow:   
1. If the product is associated with active orders, the system displays an error message and prevents deletion to avoid data inconsistency.   
2. If the selected product does not exist, the system shows an error message and prompts the administrator to try again.   
3. If the system fails to delete the product due to a database error, an error message is displayed, and the administrator is notified to investigate.  
  
Use Case Name: View Product   
Use Case ID: UC-09   
Actors: Customer, Administrator   
Preconditions: The system is operational. The product exists in the database. The customer or administrator is logged in and has access to the product catalog or management interface.   
Postconditions: The requested product details are displayed. The customer or administrator can view the relevant product information.   
  
Main Flow:   
1. The customer or administrator navigates to the product catalog or product management interface.   
2. The system displays a list of available products.   
3. The user selects a specific product from the list.   
4. The system retrieves the selected product's details from the database.   
5. The system presents the product information (e.g., name, price, description, image, category) in a user-friendly format.   
  
Alternative Flow:   
1. If the selected product does not exist, the system displays an error message and prompts the user to select a valid product.   
2. If the system fails to retrieve product information due to a database error, an error message is shown, and the administrator is notified to resolve the issue.  
  
Use Case Name: Add Category   
Use Case ID: UC-10   
Actors: Administrator   
Preconditions: The system is operational. The administrator is logged in and has access to the category management interface.   
Postconditions: A new category is added to the system and stored in the database. The category becomes available for product association.   
  
Main Flow:   
1. The administrator navigates to the "Category Management" section of the dashboard.   
2. The administrator selects the "Add Category" option.   
3. The system displays a form for entering category details.   
4. The administrator fills in the category information, including name and description.   
5. The administrator submits the form.   
6. The system validates the input data for correctness and completeness.   
7. The system adds the category to the database.   
8. The system displays a confirmation message that the category was successfully added.   
  
Alternative Flow:   
1. If the category name is already taken, the system displays an error message and prevents the addition.   
2. If the input data is incomplete or incorrect, the system shows an error and prompts the administrator to correct the information.   
3. If the system fails to store the category in the database, an error message is displayed, and the administrator is notified to resolve the issue.  
  
Use Case Name: Update Category   
Use Case ID: UC-11   
Actors: Administrator   
Preconditions: The system is operational. The administrator is logged in and has access to the category management interface. The category to be updated exists in the database.   
Postconditions: The category details are updated in the system. A confirmation message is displayed to the administrator.   
  
Main Flow:   
1. The administrator navigates to the "Category Management" section of the dashboard.   
2. The administrator selects the "Edit Category" option for the desired category.   
3. The system displays the current category information in an editable form.   
4. The administrator modifies the necessary category details (e.g., name, description).   
5. The administrator submits the updated information.   
6. The system validates the input data for correctness and completeness.   
7. The system updates the category record in the database.   
8. The system displays a confirmation message that the category was successfully updated.   
  
Alternative Flow:   
1. If the new category name is already taken, the system displays an error message and prevents the update.   
2. If the input data is incomplete or incorrect, the system shows an error and prompts the administrator to correct the information.   
3. If the system fails to update the category in the database, an error message is displayed, and the administrator is notified to resolve the issue.  
  
Use Case Name: Delete Category   
Use Case ID: UC-12   
Actors: Administrator   
Preconditions: The system is operational. The administrator is logged in and has access to the category management interface. The category to be deleted exists in the database and is not associated with any active products.   
Postconditions: The category is removed from the database. A confirmation message is displayed to the administrator.   
  
Main Flow:   
1. The administrator navigates to the "Category Management" section of the dashboard.   
2. The administrator selects the "Delete Category" option for the desired category.   
3. The system prompts the administrator to confirm the deletion.   
4. The administrator confirms the deletion request.   
5. The system checks if the category is associated with any products.   
6. The system removes the category record from the database.   
7. The system displays a confirmation message that the category was successfully deleted.   
  
Alternative Flow:   
1. If the category is associated with active products, the system displays an error message and prevents deletion to avoid data inconsistency.   
2. If the selected category does not exist, the system shows an error message and prompts the administrator to try again.   
3. If the system fails to delete the category due to a database error, an error message is displayed, and the administrator is notified to investigate.  
  
Use Case Name: View Category   
Use Case ID: UC-13   
Actors: Customer, Administrator   
Preconditions: The system is operational. The category exists in the database. The customer or administrator is logged in and has access to the product catalog or category management interface.   
Postconditions: The requested category details are displayed. The customer or administrator can view the relevant category information.   
  
Main Flow:   
1. The customer or administrator navigates to the category section of the product catalog or category management interface.   
2. The system displays a list of available categories.   
3. The user selects a specific category from the list.   
4. The system retrieves the selected category's details from the database.   
5. The system presents the category information (e.g., name, description, associated products) in a user-friendly format.   
  
Alternative Flow:   
1. If the selected category does not exist, the system displays an error message and prompts the user to select a valid category.   
2. If the system fails to retrieve category information due to a database error, an error message is shown, and the administrator is notified to resolve the issue.  
  
Use Case Name: Add Product to Shopping Cart   
Use Case ID: UC-14   
Actors: Customer   
Preconditions: The system is operational. The customer is logged in and browsing the product catalog. The product to be added exists in the system.   
Postconditions: The product is added to the customer's shopping cart. The shopping cart is updated in the system. A confirmation message is displayed to the customer.   
  
Main Flow:   
1. The customer browses the product catalog and selects a product.   
2. The system displays the product details.   
3. The customer clicks on the "Add to Cart" button.   
4. The system checks the availability of the product.   
5. The system adds the product to the customer's shopping cart.   
6. The system updates the cart total and item count.   
7. The system displays a confirmation message that the product was successfully added to the cart.   
  
Alternative Flow:   
1. If the product is out of stock, the system displays an error message and prevents the addition.   
2. If the customer is not logged in, the system prompts them to log in or register before adding the product to the cart.   
3. If the shopping cart cannot be updated due to a technical error, the system shows an error message and notifies the administrator to investigate.  
  
Use Case Name: Update Shopping Cart   
Use Case ID: UC-15   
Actors: Customer   
Preconditions: The system is operational. The customer is logged in and has an existing shopping cart. The product to be updated exists in the system.   
Postconditions: The shopping cart is updated with the new quantity or product removal. The cart total and item count are recalculated. A confirmation message is displayed to the customer.   
  
Main Flow:   
1. The customer navigates to the shopping cart section of the system.   
2. The system displays the current items in the cart, including quantities and prices.   
3. The customer selects an item to update the quantity or remove it from the cart.   
4. The system updates the cart accordingly, either changing the quantity or removing the product.   
5. The system recalculates the cart total and updates the item count.   
6. The system displays a confirmation message that the cart has been updated.   
  
Alternative Flow:   
1. If the customer tries to add a quantity exceeding the product's available stock, the system displays an error message and limits the quantity to the available stock.   
2. If the selected item does not exist in the cart, the system displays an error message and prompts the customer to select a valid item.   
3. If the system fails to update the cart due to a technical error, an error message is shown to the customer, and the administrator is notified to resolve the issue.  
  
Use Case Name: View Shopping Cart   
Use Case ID: UC-16   
Actors: Customer   
Preconditions: The system is operational. The customer is logged in and has an existing shopping cart. The customer has accessed the shopping cart section of the system.   
Postconditions: The customer's shopping cart is displayed. The customer can view the items, quantities, and total price in their cart.   
  
Main Flow:   
1. The customer navigates to the shopping cart section from the main interface.   
2. The system retrieves the customer's shopping cart data from the database.   
3. The system displays the items in the cart, including product names, quantities, individual prices, and the total amount.   
4. The system allows the customer to review the cart and proceed to checkout or update items.   
  
Alternative Flow:   
1. If the shopping cart is empty, the system displays a message indicating that no items are currently in the cart.   
2. If the system fails to retrieve the cart data due to a technical error, an error message is displayed, and the administrator is notified to resolve the issue.  
  
Use Case Name: Place Order   
Use Case ID: UC-17   
Actors: Customer, Administrator   
Preconditions: The system is operational. The customer is logged in and has at least one item in their shopping cart. The selected products are in stock. The payment gateway is functional.   
Postconditions: A new order is created and stored in the system. The shopping cart is cleared. The customer receives an order confirmation. Inventory levels are updated.   
  
Main Flow:   
1. The customer navigates to the shopping cart section.   
2. The customer clicks on the "Proceed to Checkout" button.   
3. The system displays the cart summary and prompts the customer to enter shipping and billing information.   
4. The customer enters the required details and selects a payment method.   
5. The system processes the payment through the payment gateway.   
6. The system verifies the payment was successful.   
7. The system creates a new order record in the database, including items, total price, and customer information.   
8. The system updates the inventory by reducing the stock of the ordered items.   
9. The system clears the shopping cart.   
10. The system sends an order confirmation message to the customer's email.   
11. The order is successfully placed, and the customer is redirected to an order confirmation page.   
  
Alternative Flow:   
1. If the payment fails, the system displays an error message and allows the customer to retry or choose another payment method.   
2. If the customer enters invalid shipping or billing information, the system shows an error and prompts for corrections.   
3. If the selected product is out of stock during the checkout process, the system displays an error and suggests removing or replacing the item.   
4. If the system fails to create the order record due to a database error, an error message is shown, and the administrator is notified to resolve the issue.   
5. If the system fails to update the inventory, the order is not finalized, and an error message is displayed with a request to contact support.  
  
Use Case Name: View Order   
Use Case ID: UC-18   
Actors: Customer, Administrator   
Preconditions: The system is operational. The customer is logged in and has an existing order. The administrator is logged in and has access to the order management interface.   
Postconditions: The requested order details are displayed. The customer or administrator can view the relevant order information.   
  
Main Flow:   
1. The customer or administrator navigates to the "Order History" or "Order Management" section.   
2. The system displays a list of orders (for customer) or all orders (for administrator).   
3. The user selects a specific order from the list.   
4. The system retrieves the selected order's details from the database.   
5. The system presents the order information (e.g., order ID, items, total price, order status, date placed) in a user-friendly format.   
  
Alternative Flow:   
1. If the selected order does not exist, the system displays an error message and prompts the user to select a valid order.   
2. If the system fails to retrieve order information due to a database error, an error message is shown, and the administrator is notified to resolve the issue.  
  
Use Case Name: Cancel Order   
Use Case ID: UC-19   
Actors: Customer, Administrator   
Preconditions: The system is operational. The customer is logged in and has an existing order. The administrator is logged in and has access to the order management interface. The order is in a cancellable state (e.g., not yet shipped).   
Postconditions: The order is marked as canceled in the system. The inventory is updated to reflect the cancellation. A confirmation message is displayed to the customer.   
  
Main Flow:   
1. The customer navigates to the "Order History" section of their dashboard.   
2. The customer selects the "Cancel" option for an eligible order.   
3. The system verifies that the order is in a cancellable state.   
4. The system marks the order as canceled in the database.   
5. The system updates the inventory by increasing the stock of the canceled items.   
6. The system sends a confirmation message to the customer.   
7. The system displays a confirmation that the order was successfully canceled.   
  
Alternative Flow:   
1. If the order is not in a cancellable state, the system displays an error message and prevents the cancellation.   
2. If the system fails to update inventory levels due to a technical error, an error message is shown, and the administrator is notified to resolve the issue.   
3. If the selected order does not exist, the system displays an error message and prompts the customer to select a valid order.  
  
Use Case Name: Update Order Status   
Use Case ID: UC-20   
Actors: Administrator   
Preconditions: The system is operational. The administrator is logged in and has access to the order management interface. The order exists in the system and is not yet completed or canceled.   
Postconditions: The order status is updated in the system. A confirmation message is displayed to the administrator. The customer is notified of the status change.   
  
Main Flow:   
1. The administrator navigates to the "Order Management" section of the dashboard.   
2. The administrator selects a specific order from the list.   
3. The system displays the order details, including its current status.   
4. The administrator chooses a new status (e.g., Processing, Shipped, Delivered).   
5. The administrator submits the updated status.   
6. The system validates the new status against allowed options and the order's current state.   
7. The system updates the order status in the database.   
8. The system sends a notification to the customer regarding the status change.   
9. The system displays a confirmation message to the administrator that the order status was successfully updated.   
  
Alternative Flow:   
1. If the selected order is in a non-updatable state (e.g., already delivered or canceled), the system displays an error message and prevents the status change.   
2. If the selected order does not exist, the system displays an error message and prompts the administrator to select a valid order.   
3. If the system fails to update the order status due to a database error, an error message is shown to the administrator, and they are notified to resolve the issue.   
4. If the system fails to send the notification to the customer, the administrator is alerted to manually send the update.  
  
Use Case Name: Manage Inventory   
Use Case ID: UC-21   
Actors: Administrator   
Preconditions: The system is operational. The administrator is logged in and has access to the inventory management interface. The product exists in the database.   
Postconditions: The inventory levels for the product are updated. A confirmation message is displayed to the administrator.   
  
Main Flow:   
1. The administrator navigates to the "Inventory Management" section of the dashboard.   
2. The administrator selects a product from the list to update its inventory.   
3. The system displays the current stock quantity of the selected product.   
4. The administrator enters the new stock quantity or adjusts it using increment/decrement controls.   
5. The administrator submits the updated inventory data.   
6. The system validates the input quantity (e.g., must be a non-negative number).   
7. The system updates the product's stock quantity in the database.   
8. The system displays a confirmation message that the inventory was successfully updated.   
  
Alternative Flow:   
1. If the new stock quantity is negative, the system displays an error message and prevents the update.   
2. If the selected product does not exist, the system shows an error message and prompts the administrator to select a valid product.   
3. If the system fails to update the inventory due to a database error, an error message is displayed, and the administrator is notified to investigate.  
  
Use Case Name: Manage Payment Method   
Use Case ID: UC-22   
Actors: Administrator, Customer   
Preconditions: The system is operational. The customer or administrator is logged in and has access to the payment settings section. The system supports multiple payment methods.   
Postconditions: The payment method configuration is updated in the system. The customer or administrator receives a confirmation message.   
  
Main Flow:   
1. The customer or administrator navigates to the "Payment Settings" section of their account or system dashboard.   
2. The system displays a list of currently supported payment methods.   
3. The user selects an option to add, edit, or remove a payment method.   
4. If adding a new payment method, the user enters the necessary details (e.g., method name, description, integration parameters).   
5. If editing an existing payment method, the user modifies the relevant fields and submits the changes.   
6. If removing a payment method, the system prompts the user to confirm the deletion.   
7. The system validates the input data for correctness and completeness.   
8. The system updates the payment method configuration in the database.   
9. The system displays a confirmation message that the payment method was successfully managed (added, updated, or removed).   
  
Alternative Flow:   
1. If the payment method name is not unique, the system displays an error message and prevents the addition.   
2. If the user tries to remove a payment method that is currently being used in active orders, the system displays an error message and prevents deletion to avoid data inconsistency.   
3. If the input data is incomplete or incorrect, the system shows an error and prompts the user to correct the information.   
4. If the system fails to update the payment method configuration due to a database error, an error message is displayed, and the administrator is notified to resolve the issue.  
  
Use Case Name: Manage Product Category   
Use Case ID: UC-23   
Actors: Administrator   
Preconditions: The system is operational. The administrator is logged in and has access to the product and category management interfaces. The product and category entities are properly linked in the database.   
Postconditions: The product is associated with the correct category. The system updates the product-category relationship in the database. A confirmation message is displayed to the administrator.   
  
Main Flow:   
1. The administrator navigates to the "Product Management" or "Category Management" section of the dashboard.   
2. The administrator selects an option to assign or change the category of a product.   
3. The system displays a list of products and available categories.   
4. The administrator selects a product and chooses a new category for it.   
5. The administrator submits the new category assignment.   
6. The system validates the selected product and category for existence and compatibility.   
7. The system updates the product's category association in the database.   
8. The system displays a confirmation message that the product category was successfully updated.   
  
Alternative Flow:   
1. If the selected category does not exist, the system displays an error message and prompts the administrator to choose a valid category.   
2. If the selected product does not exist, the system shows an error message and asks the administrator to select a valid product.   
3. If the system fails to update the product-category relationship due to a database error, an error message is displayed, and the administrator is notified to resolve the issue.  
  
Use Case Name: Manage Administrator   
Use Case ID: UC-24   
Actors: Administrator   
Preconditions: The system is operational. The administrator is logged in and has access to the administrator management interface. The system contains existing administrator accounts or allows the creation of new ones.   
Postconditions: The administrator account is either added, updated, or removed from the system. A confirmation message is displayed to the administrator.   
  
Main Flow:   
1. The administrator navigates to the "Administrator Management" section of the dashboard.   
2. The administrator selects an option to add, edit, or delete an administrator account.   
3. If adding a new administrator, the system displays a form for entering the administrator's details (e.g., name, email, password, role permissions).   
4. The administrator fills in the required information and submits the form.   
5. If editing an existing administrator, the system displays the current information in an editable form.   
6. The administrator modifies the necessary details and submits the changes.   
7. If deleting an administrator, the system prompts the administrator to confirm the deletion.   
8. The system validates the input data for correctness and completeness.   
9. The system updates the administrator record in the database based on the selected action.   
10. The system displays a confirmation message that the administrator was successfully managed (added, updated, or removed).   
  
Alternative Flow:   
1. If the administrator attempts to add a new administrator with an email already in use, the system displays an error message and prevents the addition.   
2. If the administrator tries to delete an account that is currently logged in or is the last active administrator, the system displays an error message and prevents deletion.   
3. If the input data is incomplete or incorrect, the system shows an error and prompts the administrator to correct the information.   
4. If the system fails to update or delete the administrator due to a database error, an error message is displayed, and the administrator is notified to resolve the issue.  
  
Use Case Name: Delete Customer   
Use Case ID: UC-25   
Actors: Administrator   
Preconditions: The system is operational. The administrator is logged in and has access to the customer management interface. The customer to be deleted exists in the database and is not associated with any active orders.   
Postconditions: The customer account is removed from the database. A confirmation message is displayed to the administrator.   
  
Main Flow:   
1. The administrator navigates to the "Customer Management" section of the dashboard.   
2. The administrator selects the "Delete Customer" option for the desired customer.   
3. The system prompts the administrator to confirm the deletion.   
4. The administrator confirms the deletion request.   
5. The system checks if the customer is associated with any active or pending orders.   
6. The system removes the customer record from the database.   
7. The system displays a confirmation message that the customer was successfully deleted.   
  
Alternative Flow:   
1. If the customer is associated with active or pending orders, the system displays an error message and prevents deletion to avoid data inconsistency.   
2. If the selected customer does not exist, the system shows an error message and prompts the administrator to try again.   
3. If the system fails to delete the customer due to a database error, an error message is displayed, and the administrator is notified to investigate.  
  
Use Case Name: Delete Order   
Use Case ID: UC-26   
Actors: Administrator   
Preconditions: The system is operational. The administrator is logged in and has access to the order management interface. The order to be deleted exists in the system and is in a deletable state (e.g., canceled or not yet processed).   
Postconditions: The order is removed from the database. A confirmation message is displayed to the administrator.   
  
Main Flow:   
1. The administrator navigates to the "Order Management" section of the dashboard.   
2. The administrator selects the "Delete Order" option for the desired order.   
3. The system prompts the administrator to confirm the deletion.   
4. The administrator confirms the deletion request.   
5. The system verifies that the order is in a deletable state.   
6. The system removes the order record from the database.   
7. The system displays a confirmation message that the order was successfully deleted.   
  
Alternative Flow:   
1. If the order is not in a deletable state (e.g., already shipped or delivered), the system displays an error message and prevents the deletion.   
2. If the selected order does not exist, the system shows an error message and prompts the administrator to select a valid order.   
3. If the system fails to delete the order due to a database error, an error message is displayed, and the administrator is notified to investigate.  
  
Use Case Name: Manage Order Items   
Use Case ID: UC-27   
Actors: Administrator   
Preconditions: The system is operational. The administrator is logged in and has access to the order management interface. The order exists in the system and is not yet completed or canceled. The order items are associated with the order.   
Postconditions: The order items are either updated, removed, or modified in the system. The inventory levels are adjusted accordingly. A confirmation message is displayed to the administrator.   
  
Main Flow:   
1. The administrator navigates to the "Order Management" section of the dashboard.   
2. The administrator selects a specific order from the list.   
3. The system displays the order details, including the list of items in the order.   
4. The administrator selects an option to edit or remove an item from the order.   
5. If editing an item, the administrator updates the quantity or product details.   
6. If removing an item, the system prompts the administrator to confirm the removal.   
7. The administrator confirms the action.   
8. The system validates the changes for correctness and compatibility with the order status.   
9. The system updates the order item in the database or removes it if confirmed.   
10. The system recalculates the total order price based on the changes.   
11. The system adjusts the inventory levels by returning or reducing stock accordingly.   
12. The system displays a confirmation message that the order items were successfully managed.   
  
Alternative Flow:   
1. If the order is in a non-editable state (e.g., already shipped or delivered), the system displays an error message and prevents the modification of order items.   
2. If the administrator tries to edit an item with a quantity exceeding the available stock, the system displays an error message and limits the quantity to the available stock.   
3. If the selected item does not exist in the order, the system shows an error message and prompts the administrator to select a valid item.   
4. If the system fails to update or remove the order item due to a database error, an error message is displayed, and the administrator is notified to resolve the issue.   
5. If the system fails to adjust the inventory levels, an error message is shown, and the administrator is alerted to investigate the issue.