项目文档

# Functional Requirement

### Chapter 1: Functional Requirements   
#### 1.1 Customer Registration   
- \*\*Function ID\*\*: FR-01   
- \*\*Description\*\*: Enables unregistered customers to create accounts by submitting required personal information. Validates inputs, checks email uniqueness, and stores profiles.   
- \*\*Input\*\*: Customer details (name, email, password, address).   
- \*\*Output\*\*: New customer profile stored in Customer data entity; confirmation message/email.   
  
#### 1.2 User Registration   
- \*\*Function ID\*\*: FR-02   
- \*\*Description\*\*: Allows unregistered users to create accounts. Validates inputs, checks email uniqueness, and stores user profiles.   
- \*\*Input\*\*: User details (name, email, password, contact information).   
- \*\*Output\*\*: New user profile stored in User data entity; confirmation message/email.   
  
#### 1.3 View Customer Information   
- \*\*Function ID\*\*: FR-03   
- \*\*Description\*\*: Permits administrators to retrieve and display customer profiles, including personal details and order history.   
- \*\*Input\*\*: Customer ID or selection from list.   
- \*\*Output\*\*: Customer details (name, email, address, order history); audit log entry.   
  
#### 1.4 Update Customer Information   
- \*\*Function ID\*\*: FR-04   
- \*\*Description\*\*: Enables customers or administrators to modify existing customer profiles (e.g., address, phone). Validates updates before saving.   
- \*\*Input\*\*: Modified customer fields (address, phone, email).   
- \*\*Output\*\*: Updated customer profile in Customer data entity; confirmation message; audit log entry.   
  
#### 1.5 Deactivate Customer Account   
- \*\*Function ID\*\*: FR-05   
- \*\*Description\*\*: Allows administrators to mark customer accounts as inactive. Verifies account status before deactivation.   
- \*\*Input\*\*: Customer ID selection.   
- \*\*Output\*\*: "Inactive" status in Customer data entity; confirmation message; audit log entry.   
  
#### 1.6 User Login   
- \*\*Function ID\*\*: FR-06   
- \*\*Description\*\*: Authenticates users/administrators via credentials. Creates sessions upon successful validation.   
- \*\*Input\*\*: Credentials (email, password).   
- \*\*Output\*\*: Active session; authentication status update in User Authentication data entity; audit log entry.   
  
#### 1.7 User Logout   
- \*\*Function ID\*\*: FR-07   
- \*\*Description\*\*: Terminates active user/administrator sessions and clears session data.   
- \*\*Input\*\*: "Logout" command.   
- \*\*Output\*\*: Session termination; "logged out" status in User Authentication data entity; audit log entry.   
  
#### 1.8 Add Product   
- \*\*Function ID\*\*: FR-08   
- \*\*Description\*\*: Allows administrators to add new products to inventory. Validates inputs (e.g., price format).   
- \*\*Input\*\*: Product details (name, description, price, quantity, category).   
- \*\*Output\*\*: New product stored in Product data entity; confirmation message; audit log entry.   
  
#### 1.9 View Product Details   
- \*\*Function ID\*\*: FR-09   
- \*\*Description\*\*: Displays product attributes (e.g., name, price, availability) to customers or administrators.   
- \*\*Input\*\*: Product selection or search query.   
- \*\*Output\*\*: Product details; audit log entry.   
  
#### 1.10 Update Product Information   
- \*\*Function ID\*\*: FR-10   
- \*\*Description\*\*: Enables administrators to modify product details (e.g., price, description). Validates inputs before saving.   
- \*\*Input\*\*: Modified product fields (price, description, quantity).   
- \*\*Output\*\*: Updated product in Product data entity; confirmation message; audit log entry.   
  
#### 1.11 Remove Product   
- \*\*Function ID\*\*: FR-11   
- \*\*Description\*\*: Allows administrators to delete products from inventory. Removes product from all shopping carts.   
- \*\*Input\*\*: Product ID selection.   
- \*\*Output\*\*: Product removed from Product data entity and Shopping Cart data entities; confirmation message; audit log entry.   
  
#### 1.12 Add Product to Shopping Cart   
- \*\*Function ID\*\*: FR-12   
- \*\*Description\*\*: Enables customers to add in-stock products to their cart. Updates quantities if product exists.   
- \*\*Input\*\*: Product selection.   
- \*\*Output\*\*: Product added/updated in Shopping Cart data entity; confirmation message; audit log entry.   
  
#### 1.13 View Shopping Cart   
- \*\*Function ID\*\*: FR-13   
- \*\*Description\*\*: Displays cart contents (items, quantities, total price) to authenticated customers.   
- \*\*Input\*\*: Access request to cart section.   
- \*\*Output\*\*: Cart items and total price; audit log entry.   
  
#### 1.14 Modify Shopping Cart   
- \*\*Function ID\*\*: FR-14   
- \*\*Description\*\*: Allows customers to adjust cart contents (e.g., update quantities, remove items). Recalculates totals.   
- \*\*Input\*\*: Modification action (e.g., "remove item," "update quantity").   
- \*\*Output\*\*: Updated Shopping Cart data entity; recalculated total price; confirmation message; audit log entry.   
  
#### 1.15 Checkout and Process Order   
- \*\*Function ID\*\*: FR-15   
- \*\*Description\*\*: Processes cart checkout: verifies item availability, collects shipping/payment details, processes payment, creates order, and clears cart.   
- \*\*Input\*\*: Shipping address, payment method, cart contents.   
- \*\*Output\*\*: New order in Order data entity; payment record in Payment Information data entity; cleared cart; confirmation message/email; audit log entry.   
  
#### 1.16 View Order History   
- \*\*Function ID\*\*: FR-16   
- \*\*Description\*\*: Displays transaction history (e.g., order IDs, dates, amounts) to customers or administrators. Supports filtering/sorting.   
- \*\*Input\*\*: Access request to transaction history.   
- \*\*Output\*\*: List of orders with details (date, status, amount); audit log entry.   
  
#### 1.17 View Order Details   
- \*\*Function ID\*\*: FR-17   
- \*\*Description\*\*: Shows specific order information (items, payment status, shipping address) upon selection.   
- \*\*Input\*\*: Order ID or selection.   
- \*\*Output\*\*: Order details; audit log entry.   
  
#### 1.18 Update Order Status   
- \*\*Function ID\*\*: FR-18   
- \*\*Description\*\*: Allows administrators to change order statuses (e.g., "Shipped," "Cancelled"). Notifies customers if applicable.   
- \*\*Input\*\*: New status selection.   
- \*\*Output\*\*: Updated status in Order data entity; customer notification; confirmation message; audit log entry.   
  
#### 1.19 Cancel Order   
- \*\*Function ID\*\*: FR-19   
- \*\*Description\*\*: Permits customers/administrators to cancel eligible orders (e.g., "Pending" status). Restocks inventory.   
- \*\*Input\*\*: Cancellation confirmation.   
- \*\*Output\*\*: "Cancelled" status in Order data entity; inventory restocked in Product data entity; customer notification; audit log entry.   
  
#### 1.20 View Payment Information   
- \*\*Function ID\*\*: FR-20   
- \*\*Description\*\*: Displays payment details (e.g., method, masked card numbers) to customers or administrators.   
- \*\*Input\*\*: Access request to payment section.   
- \*\*Output\*\*: Payment details; audit log entry.   
  
#### 1.21 Add Plugin   
- \*\*Function ID\*\*: FR-21   
- \*\*Description\*\*: Enables administrators to install new plugins by providing configuration details.   
- \*\*Input\*\*: Plugin details (name, description, version, configuration).   
- \*\*Output\*\*: New plugin stored in Plugin data entity; confirmation message; audit log entry.   
  
#### 1.22 Update Plugin   
- \*\*Function ID\*\*: FR-22   
- \*\*Description\*\*: Allows administrators to modify plugin configurations or metadata.   
- \*\*Input\*\*: Updated plugin fields (e.g., configuration settings).   
- \*\*Output\*\*: Updated plugin in Plugin data entity; confirmation message; audit log entry.   
  
#### 1.23 Remove Plugin   
- \*\*Function ID\*\*: FR-23   
- \*\*Description\*\*: Permits administrators to uninstall plugins.   
- \*\*Input\*\*: Plugin selection.   
- \*\*Output\*\*: Plugin removed from Plugin data entity; confirmation message; audit log entry.   
  
#### 1.24 Create Administrator Account   
- \*\*Function ID\*\*: FR-24   
- \*\*Description\*\*: Allows authenticated administrators to create new admin accounts with assigned roles.   
- \*\*Input\*\*: Administrator details (name, email, password, role).   
- \*\*Output\*\*: New administrator account in User Authentication data entity; confirmation message; audit log entry.   
  
#### 1.25 Update Administrator Account   
- \*\*Function ID\*\*: FR-25   
- \*\*Description\*\*: Enables administrators to modify existing admin accounts (e.g., roles, passwords).   
- \*\*Input\*\*: Updated administrator fields (role, password).   
- \*\*Output\*\*: Updated administrator account in User Authentication data entity; confirmation message; audit log entry.   
  
#### 1.26 Delete Administrator Account   
- \*\*Function ID\*\*: FR-26   
- \*\*Description\*\*: Permits administrators to remove admin accounts.   
- \*\*Input\*\*: Administrator ID selection.   
- \*\*Output\*\*: Administrator account removed from User Authentication data entity; confirmation message; audit log entry.   
  
#### 1.27 Add Authentication Method   
- \*\*Function ID\*\*: FR-27   
- \*\*Description\*\*: Allows administrators to configure new authentication mechanisms (e.g., OAuth).   
- \*\*Input\*\*: Authentication details (type, parameters).   
- \*\*Output\*\*: New authentication method in User Authentication data entity; confirmation message; audit log entry.   
  
#### 1.28 Update Authentication Method   
- \*\*Function ID\*\*: FR-28   
- \*\*Description\*\*: Enables administrators to modify existing authentication configurations.   
- \*\*Input\*\*: Updated authentication fields (parameters).   
- \*\*Output\*\*: Updated authentication method in User Authentication data entity; confirmation message; audit log entry.   
  
#### 1.29 Remove Authentication Method   
- \*\*Function ID\*\*: FR-29   
- \*\*Description\*\*: Permits administrators to delete authentication methods.   
- \*\*Input\*\*: Authentication method selection.   
- \*\*Output\*\*: Authentication method removed from User Authentication data entity; confirmation message; audit log entry.   
  
---   
\*Note: All functions adhere to referenced use cases, ensure input-output feasibility, and maintain terminological consistency.\*

# External Description

### Chapter 2: External Interfaces  
  
#### 2.1 User Interface   
The system will interact with users through a graphical user interface (GUI) that provides access to all functional requirements. The interface will include:   
- \*\*Input\*\*: User interactions such as form submissions, button clicks, and search queries.   
- \*\*Output\*\*: Screen displays showing customer information, product details, shopping cart contents, order history, and confirmation messages.   
- \*\*Description\*\*: The GUI will be designed to be user-friendly, with clear navigation and intuitive controls. It will support both customer and administrator roles, with role-based access to features.   
  
---  
  
#### 2.2 Hardware Interface   
The system will not directly interact with any external hardware devices beyond standard computing peripherals (e.g., keyboards, mice, displays).   
  
---  
  
#### 2.3 Software Interface   
The system will interact with the following external software components:   
  
1. \*\*Customer Data Entity\*\*   
 - \*\*Description\*\*: A database storing customer profiles, including personal details and order history.   
 - \*\*Interaction\*\*: The system will read and write customer data for registration, updates, and deactivation.   
 - \*\*Input/Output\*\*: Customer details (name, email, address) and order history.   
  
2. \*\*User Data Entity\*\*   
 - \*\*Description\*\*: A database storing user profiles, including administrators.   
 - \*\*Interaction\*\*: The system will read and write user data for registration, login, and account management.   
 - \*\*Input/Output\*\*: User details (name, email, password) and authentication status.   
  
3. \*\*Product Data Entity\*\*   
 - \*\*Description\*\*: A database storing product information, including inventory levels.   
 - \*\*Interaction\*\*: The system will read and write product details for adding, updating, and removing products.   
 - \*\*Input/Output\*\*: Product details (name, description, price, quantity).   
  
4. \*\*Shopping Cart Data Entity\*\*   
 - \*\*Description\*\*: A database storing temporary shopping cart contents for customers.   
 - \*\*Interaction\*\*: The system will read and write cart data for adding, modifying, and clearing items.   
 - \*\*Input/Output\*\*: Cart contents (products, quantities, totals).   
  
5. \*\*Order Data Entity\*\*   
 - \*\*Description\*\*: A database storing completed orders and their status.   
 - \*\*Interaction\*\*: The system will read and write order data for processing, updating, and viewing orders.   
 - \*\*Input/Output\*\*: Order details (items, status, shipping address).   
  
6. \*\*Payment Information Data Entity\*\*   
 - \*\*Description\*\*: A database storing payment details, including masked card numbers.   
 - \*\*Interaction\*\*: The system will read payment details for order processing and display.   
 - \*\*Input/Output\*\*: Payment details (method, masked card numbers).   
  
7. \*\*Plugin Data Entity\*\*   
 - \*\*Description\*\*: A database storing installed plugins and their configurations.   
 - \*\*Interaction\*\*: The system will read and write plugin data for adding, updating, and removing plugins.   
 - \*\*Input/Output\*\*: Plugin details (name, description, version, configuration).   
  
8. \*\*User Authentication Data Entity\*\*   
 - \*\*Description\*\*: A database storing authentication methods and administrator accounts.   
 - \*\*Interaction\*\*: The system will read and write authentication data for login, account management, and authentication method configuration.   
 - \*\*Input/Output\*\*: Authentication details (email, password, roles) and method configurations.   
  
9. \*\*Third-Party Authentication Services\*\*   
 - \*\*Description\*\*: External authentication services (e.g., OAuth) used for user login.   
 - \*\*Interaction\*\*: The system will integrate with these services to validate credentials and create sessions.   
 - \*\*Input/Output\*\*: Authentication tokens and user credentials.   
  
10. \*\*Email Service\*\*   
 - \*\*Description\*\*: An external email service used to send confirmation emails and notifications.   
 - \*\*Interaction\*\*: The system will send emails for account registration, order confirmations, and status updates.   
 - \*\*Input/Output\*\*: Email content and recipient addresses.   
  
---  
  
#### 2.4 Communication Interface   
The system will use the following communication interfaces to interact with external systems:   
  
1. \*\*Email Notifications\*\*   
 - \*\*Description\*\*: The system will send emails to users and administrators for confirmations, updates, and notifications.   
 - \*\*Interaction\*\*: Emails will be sent via an SMTP server.   
 - \*\*Input/Output\*\*: Email content (e.g., confirmation messages, order updates).   
  
2. \*\*API Calls\*\*   
 - \*\*Description\*\*: The system will interact with external APIs for authentication (e.g., OAuth) and payment processing.   
 - \*\*Interaction\*\*: RESTful API calls will be used to validate credentials and process payments.   
 - \*\*Input/Output\*\*: API requests (e.g., authentication tokens, payment details) and responses (e.g., success/failure status).   
  
3. \*\*Web Browser Communication\*\*   
 - \*\*Description\*\*: The system will interact with web browsers to display the user interface and handle user input.   
 - \*\*Interaction\*\*: The system will use HTTP/HTTPS protocols to communicate with the client-side web application.   
 - \*\*Input/Output\*\*: HTTP requests (e.g., form submissions) and responses (e.g., HTML pages, JSON data).   
  
---  
  
### Summary   
The external interfaces described in this chapter ensure seamless interaction between the system and its users, databases, third-party services, and communication channels. Each interface is clearly defined to support the functional requirements outlined in Chapter 1.

# Use Case

Use Case Name: Register Customer   
Use Case ID: UC-01   
Actors: Customer, System   
Preconditions:   
1. The customer is not yet registered in the system.   
2. The system is accessible and operational.   
  
Postconditions:   
1. A new customer account is successfully created in the system.   
2. The customer receives a confirmation message.   
  
Main Flow:   
1. The customer accesses the registration page of the system.   
2. The customer fills in the required information (e.g., name, email, password, address).   
3. The customer submits the registration form.   
4. The system validates the customer's input (e.g., checks for valid email format and password strength).   
5. The system checks if the email is already registered.   
6. If the email is not registered, the system creates a new customer profile and saves it in the Customer data entity.   
7. The system sends a confirmation message to the customer's email.   
8. The customer is redirected to the login page.   
  
Alternative Flow:   
1. If the email is already registered (step 5), the system displays an error message and prompts the customer to use a different email address.   
2. If the input validation fails (step 4), the system displays an error message for each invalid field and prompts the customer to correct the information.  
  
Use Case Name: View Customer Information   
Use Case ID: UC-02   
Actors: Administrator, System   
Preconditions:   
1. The administrator is authenticated and logged into the system.   
2. The system contains existing customer information in the Customer data entity.   
  
Postconditions:   
1. The administrator successfully views the requested customer information.   
2. The system logs the access of customer information for audit purposes.   
  
Main Flow:   
1. The administrator navigates to the customer information section in the system.   
2. The administrator selects a specific customer from the list or enters a customer ID.   
3. The system retrieves the customer's information from the Customer data entity.   
4. The system displays the customer's details (e.g., name, email, address, order history).   
5. The system logs the access event in the system audit log.   
  
Alternative Flow:   
1. If the selected customer does not exist (step 2 or 3), the system displays an error message and prompts the administrator to enter a valid customer ID.   
2. If the administrator does not have sufficient permissions to view the information (step 1), the system denies access and displays an appropriate error message.  
  
Use Case Name: Update Customer Information   
Use Case ID: UC-03   
Actors: Customer, Administrator, System   
Preconditions:   
1. The customer is authenticated and logged into the system, or the administrator is authenticated and has access to the customer management section.   
2. The system contains existing customer information in the Customer data entity.   
3. The system is accessible and operational.   
  
Postconditions:   
1. The customer's information is successfully updated in the Customer data entity.   
2. The system provides a confirmation message to the user.   
3. The system logs the update event for audit purposes.   
  
Main Flow:   
1. The customer or administrator navigates to the customer profile or management section.   
2. The user selects the specific customer whose information needs to be updated.   
3. The system displays the current customer information for editing.   
4. The user modifies the required fields (e.g., address, phone number, email).   
5. The user submits the updated information.   
6. The system validates the updated input (e.g., checks for valid email format and phone number).   
7. The system updates the customer's information in the Customer data entity.   
8. The system logs the update event in the system audit log.   
9. The system displays a confirmation message indicating that the information has been successfully updated.   
  
Alternative Flow:   
1. If the input validation fails (step 6), the system displays an error message for each invalid field and prompts the user to correct the information.   
2. If the customer or administrator does not have sufficient permissions to update the information (step 1), the system denies access and displays an appropriate error message.   
3. If the selected customer does not exist (step 2 or 4), the system displays an error message and prompts the user to enter a valid customer ID.  
  
Use Case Name: Deactivate Customer   
Use Case ID: UC-04   
Actors: Administrator, System   
Preconditions:   
1. The administrator is authenticated and logged into the system.   
2. The system contains the customer information in the Customer data entity.   
3. The administrator has the necessary permissions to deactivate customer accounts.   
  
Postconditions:   
1. The customer account is marked as inactive in the Customer data entity.   
2. The system provides a confirmation message to the administrator.   
3. The system logs the deactivation event for audit purposes.   
  
Main Flow:   
1. The administrator navigates to the customer management section in the system.   
2. The administrator selects a specific customer to deactivate.   
3. The system confirms the customer's current active status.   
4. The administrator initiates the deactivation process.   
5. The system updates the customer's status to "inactive" in the Customer data entity.   
6. The system logs the deactivation event in the audit log.   
7. The system provides a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the selected customer does not exist (step 2), the system displays an error message and prompts the administrator to enter a valid customer ID.   
2. If the customer is already inactive (step 3), the system displays a message indicating that the customer is already deactivated.   
3. If the administrator does not have sufficient permissions (step 1), the system denies access and displays an appropriate error message.  
  
Use Case Name: Register User   
Use Case ID: UC-05   
Actors: User, System   
Preconditions:   
1. The user is not yet registered in the system.   
2. The system is accessible and operational.   
  
Postconditions:   
1. A new user account is successfully created in the system.   
2. The user receives a confirmation message.   
  
Main Flow:   
1. The user accesses the registration page of the system.   
2. The user fills in the required information (e.g., name, email, password, contact details).   
3. The user submits the registration form.   
4. The system validates the user's input (e.g., checks for valid email format and password strength).   
5. The system checks if the email is already registered.   
6. If the email is not registered, the system creates a new user profile and stores it in the User data entity.   
7. The system sends a confirmation message to the user's email.   
8. The user is redirected to the login page.   
  
Alternative Flow:   
1. If the email is already registered (step 5), the system displays an error message and prompts the user to use a different email address.   
2. If the input validation fails (step 4), the system displays an error message for each invalid field and prompts the user to correct the information.  
  
Use Case Name: Manage User Sessions   
Use Case ID: UC-06   
Actors: System, User, Administrator   
  
Preconditions:   
1. The system has access to the User Authentication data entity.   
2. The user or administrator is attempting to log in or log out.   
3. The system is accessible and operational.   
  
Postconditions:   
1. The user or administrator session is either created or terminated.   
2. The session status is updated in the User Authentication data entity.   
3. The system logs the session activity for audit purposes.   
  
Main Flow:   
1. The user or administrator accesses the login page of the system.   
2. The user or administrator enters their credentials (e.g., email and password).   
3. The system validates the credentials against the User Authentication data entity.   
4. If the credentials are valid, the system creates a session and marks the user or administrator as logged in.   
5. The system logs the login event in the audit log.   
6. The user or administrator is redirected to the appropriate dashboard or homepage.   
  
Alternative Flow:   
1. If the credentials are invalid (step 3), the system displays an error message and prompts the user or administrator to re-enter their information.   
2. If the session already exists and the user or administrator attempts to log in again, the system updates the session expiration time.   
3. If the user or administrator attempts to log out, the system terminates the session and updates the status in the User Authentication data entity.   
4. If the session termination fails, the system displays an error message and prompts the user or administrator to try again.  
  
Use Case Name: Authenticate User   
Use Case ID: UC-07   
Actors: User, Administrator, System   
Preconditions:   
1. The system has access to the User Authentication data entity.   
2. The user or administrator has provided valid login credentials (e.g., email and password).   
3. The system is accessible and operational.   
  
Postconditions:   
1. The user or administrator is successfully authenticated and logged in.   
2. A session is created for the user or administrator.   
3. The system logs the authentication event for audit purposes.   
  
Main Flow:   
1. The user or administrator accesses the login page of the system.   
2. The user or administrator enters their login credentials (e.g., email and password).   
3. The system validates the credentials against the User Authentication data entity.   
4. If the credentials are valid, the system creates a session and updates the authentication status.   
5. The system logs the successful authentication event.   
6. The user or administrator is redirected to their respective dashboard or homepage.   
  
Alternative Flow:   
1. If the credentials are invalid (step 3), the system displays an error message and prompts the user or administrator to re-enter their information.   
2. If the user or administrator does not have an account (step 2), the system displays an error message and suggests registration.   
3. If the system fails to create a session (step 4), it displays an error message and prompts the user or administrator to try again.  
  
Use Case Name: Logout User   
Use Case ID: UC-08   
Actors: User, Administrator, System   
  
Preconditions:   
1. The user or administrator is authenticated and has an active session in the system.   
2. The system has access to the User Authentication data entity.   
3. The system is accessible and operational.   
  
Postconditions:   
1. The session for the user or administrator is terminated.   
2. The authentication status is updated to "logged out" in the User Authentication data entity.   
3. The system logs the logout event for audit purposes.   
  
Main Flow:   
1. The user or administrator clicks on the "Logout" button on the interface.   
2. The system verifies the active session.   
3. The system terminates the session and updates the authentication status in the User Authentication data entity.   
4. The system logs the logout event in the audit log.   
5. The user or administrator is redirected to the login page or homepage with no session data retained.   
  
Alternative Flow:   
1. If the session verification fails (step 2), the system displays an error message and prompts the user or administrator to try again or contact support.   
2. If the system fails to update the authentication status (step 3), it displays an error message and prompts the user or administrator to try again.   
3. If the user or administrator attempts to perform actions after logout, the system denies access and prompts them to log in.  
  
Use Case Name: Process Transaction   
Use Case ID: UC-09   
Actors: Customer, System, Payment Information, Shopping Cart   
  
Preconditions:   
1. The customer is authenticated and has an active session.   
2. The customer has items in their Shopping Cart.   
3. The system has access to the Payment Information and Shopping Cart data entities.   
4. The system is accessible and operational.   
  
Postconditions:   
1. The transaction is successfully processed and recorded in the system.   
2. The Shopping Cart is cleared of the purchased items.   
3. The Payment Information is securely stored or updated.   
4. The customer receives a confirmation of the transaction.   
5. The system logs the transaction event for audit purposes.   
  
Main Flow:   
1. The customer reviews the items in their Shopping Cart.   
2. The customer selects the "Checkout" option.   
3. The system prompts the customer to enter or confirm their payment information.   
4. The customer provides or selects their preferred payment method and details.   
5. The system verifies the availability of the items in the Shopping Cart.   
6. The system processes the payment through the Payment Information data entity.   
7. If the payment is successful, the system records the transaction in the Order data entity.   
8. The system clears the Shopping Cart of the purchased items.   
9. The system sends a confirmation message to the customer.   
10. The system logs the transaction details in the audit log.   
  
Alternative Flow:   
1. If the payment verification fails (step 6), the system displays an error message and prompts the customer to provide a different or valid payment method.   
2. If any item in the Shopping Cart is unavailable (step 5), the system displays a message and offers alternatives or prompts the customer to adjust their cart.   
3. If the customer cancels the transaction during checkout, the system returns to the Shopping Cart without making any changes.   
4. If the system fails to process the transaction, it displays an error message and suggests trying again later or contacting support.  
  
Use Case Name: View Transaction History   
Use Case ID: UC-10   
Actors: Customer, Administrator, System, Order, Payment Information   
  
Preconditions:   
1. The customer or administrator is authenticated and has an active session in the system.   
2. The system has access to the Order and Payment Information data entities.   
3. The system is accessible and operational.   
4. There is existing transaction data in the Order and Payment Information data entities.   
  
Postconditions:   
1. The customer or administrator successfully views the transaction history.   
2. The system logs the access of transaction history for audit purposes.   
3. The displayed information includes order details and payment status.   
  
Main Flow:   
1. The customer or administrator navigates to the transaction history section in the system.   
2. The system retrieves the relevant transaction records from the Order and Payment Information data entities.   
3. The system displays the transaction history, including order ID, date, total amount, and payment status.   
4. The customer or administrator can optionally filter or sort the transactions by date, status, or amount.   
5. The system logs the access event in the audit log.   
  
Alternative Flow:   
1. If no transaction records exist for the user (step 2), the system displays a message indicating that there is no transaction history available.   
2. If the system fails to retrieve the transaction data (step 2), it displays an error message and prompts the user to try again later.   
3. If the user does not have sufficient permissions to view the transaction history (step 1), the system denies access and displays an appropriate error message.   
4. If an error occurs during filtering or sorting (step 4), the system displays an error message and retains the original transaction list.  
  
Use Case Name: Create Order   
Use Case ID: UC-11   
Actors: Customer, System, Shopping Cart, Product, Payment Information   
  
Preconditions:   
1. The customer is authenticated and has an active session in the system.   
2. The customer has added at least one product to the Shopping Cart.   
3. The system has access to the Shopping Cart, Product, and Payment Information data entities.   
4. The system is accessible and operational.   
  
Postconditions:   
1. A new order is created in the Order data entity.   
2. The Shopping Cart is cleared of the purchased items.   
3. The payment information is securely processed and stored.   
4. The customer receives a confirmation message for the order.   
5. The system logs the order creation event for audit purposes.   
  
Main Flow:   
1. The customer reviews the items in their Shopping Cart.   
2. The customer selects the "Proceed to Checkout" option.   
3. The system verifies the availability and stock of the items in the Shopping Cart.   
4. The system prompts the customer to enter or confirm their payment and shipping information.   
5. The customer selects their preferred payment method and submits the order.   
6. The system processes the payment through the Payment Information data entity.   
7. If the payment is successful, the system creates a new order in the Order data entity.   
8. The system updates the inventory status of the products in the Product data entity.   
9. The system clears the Shopping Cart of the purchased items.   
10. The system sends a confirmation message to the customer.   
11. The system logs the order creation event in the audit log.   
  
Alternative Flow:   
1. If the items in the Shopping Cart are out of stock or unavailable (step 3), the system displays a message and prompts the customer to adjust their cart.   
2. If the payment processing fails (step 6), the system displays an error message and prompts the customer to provide a different or valid payment method.   
3. If the customer cancels the order during checkout, the system returns to the Shopping Cart without making any changes.   
4. If the customer does not provide valid shipping information (step 4), the system displays an error message and prompts the customer to correct the information.   
5. If the system fails to create the order (step 7), it displays an error message and suggests trying again later or contacting support.  
  
Use Case Name: View Order Details   
Use Case ID: UC-12   
Actors: Customer, Administrator, System, Order   
  
Preconditions:   
1. The customer or administrator is authenticated and has an active session in the system.   
2. The system contains existing order records in the Order data entity.   
3. The system is accessible and operational.   
4. The user has selected a specific order to view.   
  
Postconditions:   
1. The order details are successfully displayed to the user.   
2. The system logs the access of the order details for audit purposes.   
3. The user can view the order status, items, payment method, and shipping information.   
  
Main Flow:   
1. The customer or administrator navigates to the order details section of the system.   
2. The user selects a specific order from the list (e.g., by order ID).   
3. The system retrieves the order details from the Order data entity.   
4. The system displays the order information, including items, total amount, payment status, and shipping address.   
5. The system logs the access event in the audit log.   
  
Alternative Flow:   
1. If the selected order does not exist (step 2 or 3), the system displays an error message and prompts the user to enter a valid order ID.   
2. If the user does not have permission to view the selected order (step 1), the system denies access and displays an appropriate error message.   
3. If the system fails to retrieve the order details (step 3), it displays an error message and prompts the user to try again later.  
  
Use Case Name: Update Order Status   
Use Case ID: UC-13   
Actors: Administrator, System, Order   
  
Preconditions:   
1. The administrator is authenticated and has an active session in the system.   
2. The system contains an existing order in the Order data entity.   
3. The administrator has the necessary permissions to update order statuses.   
4. The system is accessible and operational.   
  
Postconditions:   
1. The order status is updated in the Order data entity.   
2. The system logs the status update for audit purposes.   
3. The customer (if applicable) receives a notification of the updated status.   
  
Main Flow:   
1. The administrator navigates to the order management section in the system.   
2. The administrator selects a specific order to update its status.   
3. The system retrieves the current order details from the Order data entity.   
4. The administrator chooses a new status (e.g., "Shipped," "Processing," "Cancelled").   
5. The administrator confirms the update.   
6. The system updates the order status in the Order data entity.   
7. The system logs the status change in the audit log.   
8. The system sends a notification to the customer (if applicable) regarding the updated status.   
9. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the selected order does not exist (step 2 or 3), the system displays an error message and prompts the administrator to enter a valid order ID.   
2. If the administrator does not have sufficient permissions (step 1), the system denies access and displays an appropriate error message.   
3. If the system fails to update the order status (step 6), it displays an error message and suggests trying again or contacting support.   
4. If the administrator cancels the update before confirmation (step 5), the system retains the original order status and returns to the order list.  
  
Use Case Name: Cancel Order   
Use Case ID: UC-14   
Actors: Customer, Administrator, System, Order   
  
Preconditions:   
1. The customer or administrator is authenticated and has an active session in the system.   
2. The system contains an existing order in the Order data entity.   
3. The order is in a cancellable status (e.g., "Pending," "Processing").   
4. The system is accessible and operational.   
  
Postconditions:   
1. The order is marked as "Cancelled" in the Order data entity.   
2. The system logs the cancellation event for audit purposes.   
3. The items in the cancelled order are returned to inventory in the Product data entity (if applicable).   
4. The customer receives a confirmation message about the cancellation.   
  
Main Flow:   
1. The customer or administrator navigates to the order management or order details section.   
2. The user selects a specific order to cancel.   
3. The system checks if the order is eligible for cancellation.   
4. The system prompts the user to confirm the cancellation.   
5. The user confirms the cancellation request.   
6. The system updates the order status to "Cancelled" in the Order data entity.   
7. The system logs the cancellation event in the audit log.   
8. The system sends a confirmation message to the customer.   
9. If applicable, the system updates the inventory status of the products in the Product data entity.   
  
Alternative Flow:   
1. If the selected order does not exist (step 2 or 3), the system displays an error message and prompts the user to enter a valid order ID.   
2. If the order is not in a cancellable status (step 3), the system displays a message and does not allow the cancellation.   
3. If the user does not have sufficient permissions to cancel the order (step 1), the system denies access and displays an appropriate error message.   
4. If the system fails to update the order status (step 6), it displays an error message and suggests trying again or contacting support.   
5. If the user cancels the cancellation before confirmation (step 5), the system retains the original order status and returns to the order list.  
  
Use Case Name: Add Product   
Use Case ID: UC-15   
Actors: Administrator, System, Product, Shopping Cart   
  
Preconditions:   
1. The administrator is authenticated and has an active session in the system.   
2. The system is accessible and operational.   
3. The administrator has the necessary permissions to add products.   
4. The system has access to the Product and Shopping Cart data entities.   
  
Postconditions:   
1. A new product is successfully added to the Product data entity.   
2. The product is optionally added to the Shopping Cart for demonstration or testing purposes.   
3. The system provides a confirmation message to the administrator.   
4. The system logs the addition of the product for audit purposes.   
  
Main Flow:   
1. The administrator navigates to the product management section of the system.   
2. The administrator selects the "Add Product" option.   
3. The system displays a form for entering product details.   
4. The administrator fills in the required product information (e.g., product name, description, price, quantity, category).   
5. The administrator submits the form.   
6. The system validates the input (e.g., ensures required fields are filled, checks for valid price format).   
7. The system adds the new product to the Product data entity.   
8. The system logs the product addition in the audit log.   
9. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the input validation fails (step 6), the system displays an error message for each invalid field and prompts the administrator to correct the information.   
2. If the administrator does not have sufficient permissions (step 1), the system denies access and displays an appropriate error message.   
3. If the system fails to add the product (step 7), it displays an error message and suggests trying again or contacting support.  
  
Use Case Name: View Product Details   
Use Case ID: UC-16   
Actors: Customer, Administrator, System, Product   
  
Preconditions:   
1. The customer or administrator is authenticated and has an active session in the system.   
2. The system is accessible and operational.   
3. The system contains existing product records in the Product data entity.   
4. The user has selected or searched for a specific product.   
  
Postconditions:   
1. The product details are successfully displayed to the user.   
2. The system logs the access of product details for audit purposes.   
3. The user can view product attributes such as name, price, description, and availability.   
  
Main Flow:   
1. The customer or administrator navigates to the product listing or search page.   
2. The user selects a specific product to view its details.   
3. The system retrieves the product information from the Product data entity.   
4. The system displays the product details, including name, description, price, and inventory status.   
5. The system logs the access event in the audit log.   
  
Alternative Flow:   
1. If the selected product does not exist (step 2 or 3), the system displays an error message and prompts the user to select a valid product.   
2. If the system fails to retrieve the product details (step 3), it displays an error message and prompts the user to try again later.   
3. If the user does not have sufficient permissions to view the product details (step 1), the system denies access and displays an appropriate error message.  
  
Use Case Name: Update Product Information   
Use Case ID: UC-17   
Actors: Administrator, System, Product   
Preconditions:   
1. The administrator is authenticated and has an active session in the system.   
2. The system contains an existing product in the Product data entity.   
3. The administrator has the necessary permissions to update product information.   
4. The system is accessible and operational.   
  
Postconditions:   
1. The product information is successfully updated in the Product data entity.   
2. The system provides a confirmation message to the administrator.   
3. The system logs the update event for audit purposes.   
  
Main Flow:   
1. The administrator navigates to the product management section in the system.   
2. The administrator selects a specific product to update.   
3. The system retrieves the current product details from the Product data entity.   
4. The administrator modifies the required fields (e.g., product name, price, description, quantity, or category).   
5. The administrator submits the updated information.   
6. The system validates the input (e.g., checks for valid price format and ensures required fields are filled).   
7. The system updates the product information in the Product data entity.   
8. The system logs the update event in the audit log.   
9. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the selected product does not exist (step 2 or 3), the system displays an error message and prompts the administrator to enter a valid product ID.   
2. If the administrator does not have sufficient permissions (step 1), the system denies access and displays an appropriate error message.   
3. If the input validation fails (step 6), the system displays an error message for each invalid field and prompts the administrator to correct the information.   
4. If the system fails to update the product information (step 7), it displays an error message and suggests trying again or contacting support.  
  
Use Case Name: Remove Product   
Use Case ID: UC-18   
Actors: Administrator, System, Product, Shopping Cart   
  
Preconditions:   
1. The administrator is authenticated and has an active session in the system.   
2. The system contains an existing product in the Product data entity.   
3. The administrator has the necessary permissions to remove products.   
4. The system is accessible and operational.   
  
Postconditions:   
1. The product is successfully removed from the Product data entity.   
2. The product is removed from any Shopping Cart it may be in.   
3. The system provides a confirmation message to the administrator.   
4. The system logs the removal event for audit purposes.   
  
Main Flow:   
1. The administrator navigates to the product management section in the system.   
2. The administrator selects a specific product to remove.   
3. The system retrieves the product details from the Product data entity.   
4. The system checks if the product is in any Shopping Cart.   
5. The system prompts the administrator to confirm the removal.   
6. The administrator confirms the removal request.   
7. The system removes the product from the Product data entity.   
8. The system removes the product from all associated Shopping Carts.   
9. The system logs the removal event in the audit log.   
10. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the selected product does not exist (step 2 or 3), the system displays an error message and prompts the administrator to enter a valid product ID.   
2. If the administrator does not have sufficient permissions (step 1), the system denies access and displays an appropriate error message.   
3. If the system fails to remove the product (step 7), it displays an error message and suggests trying again or contacting support.   
4. If the administrator cancels the removal before confirmation (step 6), the system retains the product and returns to the product list.  
  
Use Case Name: Add Product to Shopping Cart   
Use Case ID: UC-19   
Actors: Customer, System, Product, Shopping Cart   
  
Preconditions:   
1. The customer is authenticated and has an active session in the system.   
2. The system contains an existing product in the Product data entity.   
3. The product is available for purchase (i.e., in stock).   
4. The system is accessible and operational.   
  
Postconditions:   
1. The selected product is added to the customer's Shopping Cart.   
2. The quantity of the product in the Shopping Cart is updated if the product is already present.   
3. The system provides a confirmation message to the customer.   
4. The system logs the addition of the product to the Shopping Cart for audit purposes.   
  
Main Flow:   
1. The customer browses the product listing or search results.   
2. The customer selects a specific product to add to their Shopping Cart.   
3. The system retrieves the product details from the Product data entity.   
4. The system checks if the product is in stock and available.   
5. The system adds the product to the customer's Shopping Cart or updates the quantity if the product is already in the cart.   
6. The system provides a confirmation message indicating the product was successfully added.   
7. The system logs the action in the audit log.   
  
Alternative Flow:   
1. If the selected product is out of stock or not available (step 4), the system displays an error message and does not add it to the Shopping Cart.   
2. If the system fails to add the product to the Shopping Cart (step 5), it displays an error message and suggests trying again.   
3. If the customer is not authenticated (step 1), the system prompts them to log in or register before proceeding.   
4. If the product does not exist in the system (step 3), the system displays an error message and prompts the customer to select a valid product.  
  
Use Case Name: View Shopping Cart   
Use Case ID: UC-20   
Actors: Customer, System, Shopping Cart, Product   
  
Preconditions:   
1. The customer is authenticated and has an active session in the system.   
2. The system contains a Shopping Cart data entity associated with the customer.   
3. The system is accessible and operational.   
4. The Shopping Cart may contain one or more products.   
  
Postconditions:   
1. The customer successfully views the contents of their Shopping Cart.   
2. The system displays the product details, quantity, and total price.   
3. The system logs the access of the Shopping Cart for audit purposes.   
  
Main Flow:   
1. The customer navigates to the "Shopping Cart" section of the system.   
2. The system retrieves the customer's Shopping Cart from the Shopping Cart data entity.   
3. The system displays all the items in the cart, including product name, price, quantity, and total amount.   
4. The system logs the access event in the audit log.   
  
Alternative Flow:   
1. If the Shopping Cart is empty (step 2), the system displays a message indicating that no items are currently in the cart.   
2. If the system fails to retrieve the Shopping Cart (step 2), it displays an error message and prompts the customer to try again later.   
3. If the customer is not authenticated (step 1), the system prompts them to log in or register before proceeding.  
  
Use Case Name: Modify Shopping Cart   
Use Case ID: UC-21   
Actors: Customer, System, Shopping Cart, Product   
  
Preconditions:   
1. The customer is authenticated and has an active session in the system.   
2. The system is accessible and operational.   
3. The customer has at least one item in their Shopping Cart.   
4. The system contains the relevant product information in the Product data entity.   
  
Postconditions:   
1. The Shopping Cart is successfully modified (items added, removed, or quantities updated).   
2. The system provides a confirmation message to the customer.   
3. The system logs the modification event for audit purposes.   
  
Main Flow:   
1. The customer navigates to the "Shopping Cart" section of the system.   
2. The system displays the items currently in the Shopping Cart.   
3. The customer selects an action to modify the cart (e.g., increase quantity, decrease quantity, remove item).   
4. The system updates the Shopping Cart accordingly.   
5. The system recalculates and displays the total price.   
6. The system logs the modification event in the audit log.   
7. The system provides a confirmation message to the customer.   
  
Alternative Flow:   
1. If the system fails to update the Shopping Cart (step 4), it displays an error message and suggests trying again.   
2. If the customer attempts to remove an item that does not exist in the cart (step 3), the system displays an error message and retains the cart.   
3. If the system fails to recalculate the total price (step 5), it displays an error message and retains the previous total.   
4. If the customer is not authenticated (step 1), the system prompts them to log in or register before proceeding.   
5. If the product no longer exists in the system (step 4), the system displays an error message and removes the invalid item from the cart.  
  
Use Case Name: Checkout Shopping Cart   
Use Case ID: UC-22   
Actors: Customer, System, Shopping Cart, Payment Information, Order   
  
Preconditions:   
1. The customer is authenticated and has an active session in the system.   
2. The customer has at least one item in their Shopping Cart.   
3. The system has access to the Shopping Cart, Payment Information, and Order data entities.   
4. The system is accessible and operational.   
  
Postconditions:   
1. The customer's Shopping Cart is cleared of the purchased items.   
2. A new order is created in the Order data entity.   
3. Payment information is securely processed and stored.   
4. The customer receives a confirmation message for the completed checkout.   
5. The system logs the checkout event for audit purposes.   
  
Main Flow:   
1. The customer navigates to the "Checkout" page from the Shopping Cart.   
2. The system verifies the items in the Shopping Cart and their availability.   
3. The system prompts the customer to enter or confirm their shipping address and contact information.   
4. The customer selects or enters their preferred payment method.   
5. The system validates the payment information.   
6. The system processes the payment through the Payment Information data entity.   
7. If the payment is successful, the system creates a new order in the Order data entity.   
8. The system clears the Shopping Cart of the purchased items.   
9. The system sends a confirmation message to the customer.   
10. The system logs the checkout event in the audit log.   
  
Alternative Flow:   
1. If the items in the Shopping Cart are unavailable (step 2), the system displays an error message and prompts the customer to adjust their cart.   
2. If the customer does not provide valid shipping information (step 3), the system displays an error message and prompts the customer to correct the information.   
3. If the payment validation fails (step 5), the system displays an error message and prompts the customer to provide a different or valid payment method.   
4. If the payment processing fails (step 6), the system displays an error message and prompts the customer to try again or contact support.   
5. If the system fails to create the order (step 7), it displays an error message and suggests trying again later or contacting support.   
6. If the customer cancels the checkout process, the system returns to the Shopping Cart without making any changes.  
  
Use Case Name: Process Payment   
Use Case ID: UC-23   
Actors: Customer, System, Payment Information, Shopping Cart, Order   
  
Preconditions:   
1. The customer is authenticated and has an active session in the system.   
2. The customer has items in their Shopping Cart.   
3. The system has access to the Payment Information, Shopping Cart, and Order data entities.   
4. The system is accessible and operational.   
  
Postconditions:   
1. The payment is processed and recorded in the Payment Information data entity.   
2. The Shopping Cart is cleared of the purchased items.   
3. A new order is created in the Order data entity.   
4. The customer receives a confirmation message of the completed payment.   
5. The system logs the payment event for audit purposes.   
  
Main Flow:   
1. The customer selects the "Checkout" option from the Shopping Cart.   
2. The system displays the payment details and prompts the customer to confirm.   
3. The customer selects or enters their payment method and submits the payment.   
4. The system validates the payment information.   
5. The system processes the payment via the Payment Information data entity.   
6. The system creates a new order in the Order data entity.   
7. The system clears the Shopping Cart of the purchased items.   
8. The system sends a confirmation message to the customer.   
9. The system logs the payment and order creation in the audit log.   
  
Alternative Flow:   
1. If the payment information is invalid (step 4), the system displays an error message and prompts the customer to correct the details.   
2. If the payment processing fails (step 5), the system displays an error message and suggests trying a different payment method.   
3. If the customer cancels the payment (step 3), the system returns to the Shopping Cart without making any changes.   
4. If the system fails to create the order (step 6), it displays an error message and suggests trying again or contacting support.  
  
Use Case Name: View Payment Information   
Use Case ID: UC-24   
Actors: Customer, Administrator, System, Payment Information   
  
Preconditions:   
1. The customer or administrator is authenticated and has an active session in the system.   
2. The system contains existing payment information in the Payment Information data entity.   
3. The user has selected to view their payment details.   
4. The system is accessible and operational.   
  
Postconditions:   
1. The payment information is successfully displayed to the user.   
2. The system logs the access of payment information for audit purposes.   
3. The user can view payment method, transaction history, and stored card details (if applicable).   
  
Main Flow:   
1. The customer or administrator navigates to the payment information section in the system.   
2. The user selects to view their payment details.   
3. The system retrieves the relevant payment information from the Payment Information data entity.   
4. The system displays the payment information, including method, card details (masked), and transaction history.   
5. The system logs the access event in the audit log.   
  
Alternative Flow:   
1. If the selected payment information does not exist (step 3), the system displays an error message and prompts the user to check their inputs.   
2. If the user does not have permission to view the payment information (step 1), the system denies access and displays an appropriate error message.   
3. If the system fails to retrieve the payment information (step 3), it displays an error message and prompts the user to try again later.  
  
Use Case Name: View Order Information   
Use Case ID: UC-12   
Actors: Customer, Administrator, System, Order   
  
Preconditions:   
1. The customer or administrator is authenticated and has an active session in the system.   
2. The system contains existing order records in the Order data entity.   
3. The user has selected a specific order to view.   
4. The system is accessible and operational.   
  
Postconditions:   
1. The order information is successfully displayed to the user.   
2. The system logs the access of the order information for audit purposes.   
3. The user can view details such as order status, items, total amount, and payment method.   
  
Main Flow:   
1. The customer or administrator navigates to the order details section of the system.   
2. The user selects a specific order by entering an order ID or choosing from a list.   
3. The system retrieves the selected order information from the Order data entity.   
4. The system displays the order details, including items, total amount, payment status, and order date.   
5. The system logs the access event in the audit log.   
  
Alternative Flow:   
1. If the selected order does not exist (step 2 or 3), the system displays an error message and prompts the user to enter a valid order ID.   
2. If the user does not have permission to view the selected order (step 1), the system denies access and displays an appropriate error message.   
3. If the system fails to retrieve the order information (step 3), it displays an error message and prompts the user to try again later.  
  
Use Case Name: Manage Administrator Account   
Use Case ID: UC-25   
Actors: Administrator, System, User Authentication   
  
Preconditions:   
1. The administrator is authenticated and has an active session in the system.   
2. The system has access to the User Authentication data entity.   
3. The administrator has the necessary permissions to manage administrator accounts.   
4. The system is accessible and operational.   
  
Postconditions:   
1. The administrator account is successfully created, updated, or deleted in the system.   
2. The system logs the account management event for audit purposes.   
3. The system provides a confirmation message to the administrator.   
  
Main Flow:   
1. The administrator navigates to the administrator account management section in the system.   
2. The administrator selects an action to create, update, or delete an administrator account.   
3. If creating a new account, the administrator enters the new administrator's details (e.g., name, email, password, role).   
4. If updating an account, the administrator selects an existing administrator and modifies the required fields.   
5. If deleting an account, the administrator selects an existing administrator and confirms the deletion.   
6. The system validates the input (e.g., checks for valid email format, password strength, and role assignment).   
7. The system updates the User Authentication data entity accordingly.   
8. The system logs the account management action in the audit log.   
9. The system provides a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the email is already registered (step 3), the system displays an error message and prompts the administrator to use a different email address.   
2. If the input validation fails (step 6), the system displays an error message for each invalid field and prompts the administrator to correct the information.   
3. If the administrator does not have sufficient permissions (step 2), the system denies access and displays an appropriate error message.   
4. If the selected administrator account does not exist (step 5), the system displays an error message and prompts the administrator to enter a valid administrator ID.   
5. If the system fails to update the User Authentication data entity (step 7), it displays an error message and suggests trying again or contacting support.  
  
Use Case Name: Add Plugin   
Use Case ID: UC-26   
Actors: Administrator, System, Plugin   
  
Preconditions:   
1. The administrator is authenticated and has an active session in the system.   
2. The system is accessible and operational.   
3. The administrator has the necessary permissions to add plugins.   
4. The system has access to the Plugin data entity.   
  
Postconditions:   
1. A new plugin is successfully added to the Plugin data entity.   
2. The system provides a confirmation message to the administrator.   
3. The system logs the addition of the plugin for audit purposes.   
  
Main Flow:   
1. The administrator navigates to the plugin management section in the system.   
2. The administrator selects the "Add Plugin" option.   
3. The system displays a form for entering plugin details.   
4. The administrator fills in the required plugin information (e.g., name, description, version, configuration settings).   
5. The administrator submits the form.   
6. The system validates the input (e.g., ensures required fields are filled, checks for valid format).   
7. The system adds the new plugin to the Plugin data entity.   
8. The system logs the plugin addition in the audit log.   
9. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the input validation fails (step 6), the system displays an error message for each invalid field and prompts the administrator to correct the information.   
2. If the administrator does not have sufficient permissions (step 1), the system denies access and displays an appropriate error message.   
3. If the system fails to add the plugin (step 7), it displays an error message and suggests trying again or contacting support.  
  
Use Case Name: Manage Plugin   
Use Case ID: UC-27   
Actors: Administrator, System, Plugin   
  
Preconditions:   
1. The administrator is authenticated and has an active session in the system.   
2. The system is accessible and operational.   
3. The administrator has the necessary permissions to manage plugins.   
4. The system has access to the Plugin data entity.   
  
Postconditions:   
1. The plugin is successfully added, updated, or removed from the Plugin data entity.   
2. The system provides a confirmation message to the administrator.   
3. The system logs the plugin management event for audit purposes.   
  
Main Flow:   
1. The administrator navigates to the plugin management section in the system.   
2. The administrator selects an action to add, update, or remove a plugin.   
3. For adding a plugin, the administrator enters the plugin's details (e.g., name, description, version, configuration).   
4. For updating a plugin, the administrator selects an existing plugin and modifies the required fields.   
5. For removing a plugin, the administrator selects an existing plugin and confirms the deletion.   
6. The system validates the input (e.g., required fields, valid format).   
7. The system updates the Plugin data entity accordingly.   
8. The system logs the action in the audit log.   
9. The system provides a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the input validation fails (step 6), the system displays an error message for each invalid field and prompts the administrator to correct the information.   
2. If the administrator does not have sufficient permissions (step 2), the system denies access and displays an appropriate error message.   
3. If the system fails to update the Plugin data entity (step 7), it displays an error message and suggests trying again or contacting support.   
4. If the selected plugin does not exist (step 5), the system displays an error message and prompts the administrator to enter a valid plugin ID.  
  
Use Case Name: Remove Plugin   
Use Case ID: UC-28   
Actors: Administrator, System, Plugin   
  
Preconditions:   
1. The administrator is authenticated and has an active session in the system.   
2. The system contains an existing plugin in the Plugin data entity.   
3. The administrator has the necessary permissions to remove plugins.   
4. The system is accessible and operational.   
  
Postconditions:   
1. The selected plugin is successfully removed from the Plugin data entity.   
2. The system provides a confirmation message to the administrator.   
3. The system logs the removal of the plugin for audit purposes.   
  
Main Flow:   
1. The administrator navigates to the plugin management section in the system.   
2. The administrator selects a specific plugin to remove.   
3. The system retrieves the plugin details from the Plugin data entity.   
4. The system prompts the administrator to confirm the removal.   
5. The administrator confirms the removal request.   
6. The system removes the plugin from the Plugin data entity.   
7. The system logs the removal event in the audit log.   
8. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the selected plugin does not exist (step 2 or 3), the system displays an error message and prompts the administrator to enter a valid plugin ID.   
2. If the administrator does not have sufficient permissions (step 1), the system denies access and displays an appropriate error message.   
3. If the system fails to remove the plugin (step 6), it displays an error message and suggests trying again or contacting support.   
4. If the administrator cancels the removal before confirmation (step 5), the system retains the plugin and returns to the plugin list.  
  
Use Case Name: Manage Shopping Cart   
Use Case ID: UC-29   
Actors: Customer, System, Shopping Cart, Product   
  
Preconditions:   
1. The customer is authenticated and has an active session in the system.   
2. The system contains a Shopping Cart data entity associated with the customer.   
3. The system is accessible and operational.   
4. The Shopping Cart may contain one or more products.   
  
Postconditions:   
1. The Shopping Cart is successfully modified (items added, removed, or quantities updated).   
2. The system provides a confirmation message to the customer.   
3. The system logs the modification event for audit purposes.   
4. The total price in the Shopping Cart is recalculated and displayed.   
  
Main Flow:   
1. The customer navigates to the "Shopping Cart" section of the system.   
2. The system displays the items currently in the Shopping Cart, including product name, quantity, price, and total.   
3. The customer selects an action to modify the cart (e.g., add a new item, remove an existing item, or update the quantity of an item).   
4. The system performs the requested modification to the Shopping Cart.   
5. The system recalculates the total price of the Shopping Cart.   
6. The system updates the Shopping Cart data entity with the new information.   
7. The system provides a confirmation message to the customer indicating the modification was successful.   
8. The system logs the modification event in the audit log.   
  
Alternative Flow:   
1. If the system fails to update the Shopping Cart (step 4), it displays an error message and suggests trying again.   
2. If the customer attempts to modify an item that does not exist in the cart (step 3), the system displays an error message and retains the current cart.   
3. If the system fails to recalculate the total price (step 5), it displays an error message and retains the previous total.   
4. If the customer is not authenticated (step 1), the system prompts them to log in or register before proceeding.   
5. If the product being added or modified no longer exists in the system (step 3), the system displays an error message and removes the invalid item from the cart.  
  
Use Case Name: Manage User Authentication   
Use Case ID: UC-30   
Actors: Administrator, System, User, User Authentication   
  
Preconditions:   
1. The administrator is authenticated and has an active session in the system.   
2. The system has access to the User Authentication data entity.   
3. The administrator has the necessary permissions to manage user authentication.   
4. The system is accessible and operational.   
  
Postconditions:   
1. The user authentication settings are successfully created, updated, or deleted in the User Authentication data entity.   
2. The system provides a confirmation message to the administrator.   
3. The system logs the authentication management event for audit purposes.   
4. User login functionality is affected accordingly based on the changes made.   
  
Main Flow:   
1. The administrator navigates to the user authentication management section in the system.   
2. The administrator selects an action to add, modify, or delete user authentication settings.   
3. For adding a new authentication method, the administrator enters the required details (e.g., authentication type, configuration parameters).   
4. For modifying an existing authentication method, the administrator selects the method and updates the relevant fields.   
5. For deleting an authentication method, the administrator selects the method and confirms the deletion.   
6. The system validates the input (e.g., ensures correct configuration format and valid authentication types).   
7. The system updates the User Authentication data entity accordingly.   
8. The system logs the action in the audit log.   
9. The system provides a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the input validation fails (step 6), the system displays an error message for each invalid field and prompts the administrator to correct the information.   
2. If the administrator does not have sufficient permissions (step 2), the system denies access and displays an appropriate error message.   
3. If the system fails to update the User Authentication data entity (step 7), it displays an error message and suggests trying again or contacting support.   
4. If the selected authentication method does not exist (step 5), the system displays an error message and prompts the administrator to enter a valid authentication method ID.