项目文档

# Functional Requirement

1. Functional Requirements   
 1.1 Customer Account Creation   
 Function ID: FR-01   
 Description: The system allows a customer to create a new account by submitting their personal information, including username, password, email, and phone number. The system validates the input and generates a unique customer ID.   
 Input: PersonalInformation (FirstName, LastName, Email, Username, Password, PhoneNumber, ContactDetails)   
 Output: Customer (CustomerID, Username, Email, PhoneNumber, Address)  
  
 1.2 Customer Login   
 Function ID: FR-02   
 Description: The system authenticates a customer by verifying their email and password against the stored user account. A successful login establishes a session and displays the customer dashboard.   
 Input: Email, Password   
 Output: Customer session (authenticated status), Dashboard view  
  
 1.3 Customer Logout   
 Function ID: FR-03   
 Description: The system terminates the customer's active session and redirects them to the homepage or login page. The logout event is logged for auditing.   
 Input: Active session   
 Output: Terminated session, Redirect to homepage or login page  
  
 1.4 Update Customer Personal Information   
 Function ID: FR-04   
 Description: The system allows a logged-in customer to modify their personal details, such as name, phone number, and address. Changes are validated and saved in the database.   
 Input: Updated PersonalInformation (FirstName, LastName, PhoneNumber, Address)   
 Output: Updated Customer record in the database  
  
 1.5 View Purchase History   
 Function ID: FR-05   
 Description: The system retrieves and displays a customer’s historical purchases, including product details, quantity, total price, and purchase date.   
 Input: CustomerID   
 Output: List of PurchaseHistory records  
  
 1.6 Manage Cart   
 Function ID: FR-06   
 Description: The system allows a customer to add, remove, and update product quantities in their cart. The cart total is recalculated in real-time.   
 Input: CartID, ProductID, Quantity, Action (Add, Remove, Update)   
 Output: Updated Cart (ProductID, Quantity, SubTotal), Cart display  
  
 1.7 Checkout Order   
 Function ID: FR-07   
 Description: The system processes the customer's cart at checkout, confirms payment, creates an order, and clears the cart.   
 Input: CartID, ShippingAddress, PaymentMethodID   
 Output: New Order record, Cleared Cart, Order confirmation  
  
 1.8 Confirm Order   
 Function ID: FR-08   
 Description: The system confirms the order after successful payment processing, updates the order status, and logs the confirmation event.   
 Input: OrderID, PaymentInformation   
 Output: Confirmed Order status, Order confirmation message  
  
 1.9 Send Order Confirmation Email   
 Function ID: FR-09   
 Description: The system sends a confirmation email to the customer after an order is successfully confirmed.   
 Input: OrderID, CustomerEmail   
 Output: Email (Subject, Content, Timestamp), Email log entry  
  
 1.10 Admin Login   
 Function ID: FR-10   
 Description: The system authenticates an administrator by verifying their email and password. A successful login establishes an admin session and displays the admin dashboard.   
 Input: Email, Password   
 Output: Admin session (authenticated status), Admin dashboard  
  
 1.11 Admin Manage Inventory   
 Function ID: FR-11   
 Description: The system allows an administrator to add, update, or delete inventory records for products. Inventory alerts are triggered when stock falls below a threshold.   
 Input: ProductID, StockQuantity, ReorderLevel, Action (Add, Update, Delete)   
 Output: Updated Inventory record, Inventory alert (if applicable)  
  
 1.12 Admin Add Product   
 Function ID: FR-12   
 Description: The system allows an administrator to create a new product entry, including its name, description, price, category, and inventory quantity.   
 Input: Product (Name, Description, Price, CategoryID, InventoryID, Image)   
 Output: New Product record, Updated Inventory and Category associations  
  
 1.13 Admin Update Product Details   
 Function ID: FR-13   
 Description: The system allows an administrator to modify existing product details such as name, price, description, and category.   
 Input: ProductID, UpdatedProductDetails (Name, Description, Price, CategoryID)   
 Output: Updated Product record, Updated Inventory and Category associations  
  
 1.14 Admin Delete Product   
 Function ID: FR-14   
 Description: The system allows an administrator to delete a product, which removes the product record and updates the inventory accordingly.   
 Input: ProductID, Confirmation to delete   
 Output: Deleted Product record, Updated Inventory  
  
 1.15 Admin Categorize Products   
 Function ID: FR-15   
 Description: The system allows an administrator to assign or change the category of a product.   
 Input: ProductID, CategoryID   
 Output: Updated Product record, Updated Category association  
  
 1.16 Admin Manage User Accounts   
 Function ID: FR-16   
 Description: The system allows an administrator to create, update, or delete user accounts.   
 Input: UserAccount (Username, Email, Password, Role), Action (Create, Update, Delete)   
 Output: Updated or deleted UserAccount record  
  
 1.17 Admin Manage Payment Methods   
 Function ID: FR-17   
 Description: The system allows an administrator to add, update, or delete available payment methods for the store.   
 Input: PaymentMethod (MethodType, AccountDetails), Action (Add, Update, Delete)   
 Output: Updated or deleted PaymentMethod record  
  
 1.18 Admin Install Plugin   
 Function ID: FR-18   
 Description: The system allows an administrator to install a plugin, which extends the system’s functionality.   
 Input: Plugin file (Name, Version, Description)   
 Output: Installed Plugin record, Updated Plugin status  
  
 1.19 Admin Configure Plugin   
 Function ID: FR-19   
 Description: The system allows an administrator to configure plugin settings after installation.   
 Input: PluginID, Configuration parameters (e.g., API keys, enable/disable features)   
 Output: Updated Plugin configuration in the database  
  
 1.20 Admin Manage System Settings   
 Function ID: FR-20   
 Description: The system allows an administrator to configure and update system-wide settings, such as site configuration, payment gateway, and email settings.   
 Input: SystemSettings (Key, Value), Action (Update or Add)   
 Output: Updated or added SystemSettings record  
  
 1.21 Admin View System Logs   
 Function ID: FR-21   
 Description: The system allows an administrator to view all logged system activities for auditing and monitoring purposes.   
 Input: Filter criteria (e.g., date range, user, event type)   
 Output: List of SystemLogs, System log access record  
  
 1.22 Admin Manage System Logs   
 Function ID: FR-22   
 Description: The system allows an administrator to perform actions such as filtering, exporting, or clearing system logs.   
 Input: SystemLogs, Action (Filter, Export, Clear)   
 Output: Filtered or exported logs, Cleared logs (if applicable)  
  
 1.23 View Product Catalog   
 Function ID: FR-23   
 Description: The system displays a list of available products grouped by category, allowing the customer to search, filter, and view product details.   
 Input: Search term, Category filter, Price range   
 Output: Displayed product list, Product detail view  
  
 1.24 View Order Details   
 Function ID: FR-24   
 Description: The system retrieves and displays detailed information about a specific order, such as product list, total cost, payment status, and shipping information.   
 Input: OrderID   
 Output: Displayed Order details, Order status  
  
 1.25 Cancel Order   
 Function ID: FR-25   
 Description: The system allows a customer to cancel an order if it is in a cancellable state (e.g., pending or processing). The system updates the inventory accordingly.   
 Input: OrderID, Confirmation to cancel   
 Output: Updated Order status (Canceled), Updated Inventory  
  
 1.26 Manage Order Status   
 Function ID: FR-26   
 Description: The system tracks and updates the status of an order (e.g., pending, confirmed, shipped) based on system events or admin actions.   
 Input: OrderID, StatusName, StatusDescription   
 Output: Updated OrderStatus record, Order status display  
  
 1.27 Generate Sales Report   
 Function ID: FR-27   
 Description: The system allows an administrator to generate a sales report based on selected filters such as date range and category.   
 Input: Date range, Category filter, Export format (e.g., PDF, Excel)   
 Output: Generated sales report, Log of report generation  
  
 1.28 Modify User Permissions   
 Function ID: FR-28   
 Description: The system allows an administrator to modify user roles and permissions to control access to system features.   
 Input: UserID, UpdatedPermissions (e.g., role, feature access)   
 Output: Updated User record with new permissions, Log of permission change

# External Description

# 2. External Interfaces  
  
This section describes the external interfaces that the system interacts with during its operation. These interfaces include user interfaces, hardware interfaces, software interfaces (such as databases and APIs), and communication interfaces (such as email and messaging). Each interface is defined clearly, including the purpose, interaction method, and relevant inputs/outputs.  
  
---  
  
## 2.1 User Interface  
  
The system provides a user interface for both customers and administrators to interact with the application. The user interface includes web-based dashboards, forms, and navigation menus that allow users to perform various operations such as account creation, login, product browsing, order management, etc.  
  
### 2.1.1 Customer Dashboard  
- \*\*Description\*\*: A web-based interface that displays customer-specific information such as purchase history, cart contents, and order status.  
- \*\*Interaction Method\*\*: Web browser access via HTTP(S) protocol.  
- \*\*Inputs\*\*:  
 - User credentials (email, password)  
- \*\*Outputs\*\*:  
 - Display of dashboard content (e.g., cart, recent orders, profile settings)  
 - Interactive elements (buttons, links, form fields)  
  
### 2.1.2 Admin Dashboard  
- \*\*Description\*\*: A web-based interface that allows administrators to manage inventory, users, products, and system settings.  
- \*\*Interaction Method\*\*: Web browser access via HTTP(S) protocol.  
- \*\*Inputs\*\*:  
 - Admin credentials (email, password)  
- \*\*Outputs\*\*:  
 - Display of admin controls (inventory list, user list, payment methods, logs, etc.)  
 - Interactive elements (forms, tables, buttons)  
  
### 2.1.3 Product Catalog View  
- \*\*Description\*\*: A web page or section that displays available products grouped by category, allowing customers to search, filter, and view product details.  
- \*\*Interaction Method\*\*: Web browser access via HTTP(S) protocol.  
- \*\*Inputs\*\*:  
 - Search term  
 - Category filter  
 - Price range  
- \*\*Outputs\*\*:  
 - Displayed product list with images, descriptions, and pricing  
 - Product detail view on click  
  
### 2.1.4 Order Details View  
- \*\*Description\*\*: A web-based interface that shows detailed information about a specific order, including items, total cost, payment status, and shipping information.  
- \*\*Interaction Method\*\*: Web browser access via HTTP(S) protocol.  
- \*\*Inputs\*\*:  
 - Order ID  
- \*\*Outputs\*\*:  
 - Display of order summary and itemized breakdown  
 - Status indicators (e.g., confirmed, shipped, canceled)  
  
---  
  
## 2.2 Hardware Interface  
  
Currently, the system does not interact directly with any external hardware devices. Therefore, this section contains no defined hardware interfaces.  
  
---  
  
## 2.3 Software Interface  
  
The system interacts with several internal and external software components, including databases, third-party APIs, and other systems.  
  
### 2.3.1 Database System  
- \*\*Description\*\*: The system stores all data in one or more relational databases, including customer accounts, products, orders, inventory, logs, and system settings.  
- \*\*Interaction Method\*\*: Structured Query Language (SQL) over a database connection (e.g., MySQL, PostgreSQL).  
- \*\*Inputs\*\*:  
 - SQL queries for data retrieval, updates, and deletion  
- \*\*Outputs\*\*:  
 - Data records (customer info, product details, order statuses, etc.)  
 - Database transaction logs  
  
### 2.3.2 Payment Gateway API  
- \*\*Description\*\*: The system communicates with an external payment gateway (e.g., PayPal, Stripe) to process payments during checkout.  
- \*\*Interaction Method\*\*: RESTful API calls using HTTPS.  
- \*\*Inputs\*\*:  
 - Payment method ID  
 - Payment amount  
 - Card details or payment token  
- \*\*Outputs\*\*:  
 - Payment confirmation or error response  
 - Transaction ID for record-keeping  
  
### 2.3.3 Email Service API  
- \*\*Description\*\*: The system uses an email service API (e.g., SendGrid, SMTP) to send order confirmation emails and other notifications.  
- \*\*Interaction Method\*\*: RESTful API calls using HTTPS.  
- \*\*Inputs\*\*:  
 - Order ID  
 - Customer email address  
 - Email subject and body  
- \*\*Outputs\*\*:  
 - Confirmation of email delivery  
 - Log entry of sent email  
  
### 2.3.4 Plugin Management System  
- \*\*Description\*\*: The system allows plugins to be installed, configured, and managed. Plugins may extend functionality such as reporting, security, or user interface enhancements.  
- \*\*Interaction Method\*\*: File upload and configuration through the admin interface.  
- \*\*Inputs\*\*:  
 - Plugin file (name, version, description)  
 - Configuration parameters (API keys, enable/disable features)  
- \*\*Outputs\*\*:  
 - Installed plugin record  
 - Updated plugin configuration  
  
### 2.3.5 Logging System  
- \*\*Description\*\*: The system logs events such as user logins, order confirmations, and system errors into a centralized logging system for auditing and monitoring.  
- \*\*Interaction Method\*\*: Structured logging format (JSON, XML) via HTTP or direct database write.  
- \*\*Inputs\*\*:  
 - Event type (login, order confirmation, error)  
 - Timestamp, user ID, event message  
- \*\*Outputs\*\*:  
 - Log entries stored in a database or log file  
 - Access logs for audit purposes  
  
---  
  
## 2.4 Communication Interface  
  
The system communicates with external entities through various network-based communication channels, primarily via web and email.  
  
### 2.4.1 Web Browser Communication  
- \*\*Description\*\*: The system communicates with web browsers to render user interfaces and receive user input.  
- \*\*Interaction Method\*\*: HTTP/HTTPS protocol.  
- \*\*Inputs\*\*:  
 - User actions (clicks, form submissions, selections)  
- \*\*Outputs\*\*:  
 - Dynamic content rendering (HTML, JavaScript)  
 - Real-time updates (e.g., cart changes, order status)  
  
### 2.4.2 Email Notification  
- \*\*Description\*\*: The system sends email notifications to users for events such as order confirmation, account verification, and system alerts.  
- \*\*Interaction Method\*\*: SMTP or third-party email API (e.g., SendGrid, Mailgun).  
- \*\*Inputs\*\*:  
 - Order ID  
 - Customer email address  
 - Email template (subject, body)  
- \*\*Outputs\*\*:  
 - Sent email message  
 - Log entry of sent email  
  
### 2.4.3 System Logs Communication  
- \*\*Description\*\*: The system communicates with a logging server or central log storage system to store system events and audit trails.  
- \*\*Interaction Method\*\*: HTTP or direct database insertion.  
- \*\*Inputs\*\*:  
 - Log message, timestamp, user ID, event type  
- \*\*Outputs\*\*:  
 - Stored log entry  
 - Audit trail for system activity  
  
---  
  
## Summary of External Interfaces  
  
| Interface Type | Description | Interaction Method |  
|----------------------|-----------------------------------------------------------------------------|----------------------------|  
| User Interface | Web-based dashboards, forms, and views for customer and admin interactions | HTTP(S), JavaScript |  
| Software Interface | Databases, payment gateways, email services, and plugin systems | SQL, REST, SMTP, File |  
| Communication Interface | Email notifications, system logs, and real-time updates | HTTP(S), SMTP, JSON |  
  
All external data sources mentioned in the functional requirements are accounted for and clearly defined within this chapter. This ensures consistency and clarity for developers, testers, and stakeholders involved in the project.

# Use Case

Use Case Name: Customer Registration   
Use Case ID: UC-01   
Actors: Customer, System   
Preconditions:   
1. The system is operational.   
2. The customer is not yet registered in the system.   
3. The customer has access to the registration interface (e.g., a web form or mobile app).   
  
Postconditions:   
1. A new customer account is created in the system.   
2. The customer receives a confirmation email.   
3. The customer is redirected to the login page or dashboard.   
  
Main Flow:   
1. The customer navigates to the registration page.   
2. The system displays the registration form.   
3. The customer fills in the required information (e.g., name, email, password, phone number).   
4. The customer submits the registration form.   
5. The system validates the input data (e.g., unique email, valid password format).   
6. The system creates a new customer record in the database.   
7. The system sends a confirmation email to the customer.   
8. The system redirects the customer to the login page.   
  
Alternative Flow:   
1. If the email already exists in the system, the system displays an error message and prompts the customer to enter a different email.   
2. If the password does not meet the format requirements (e.g., too short or missing special characters), the system displays an error message and prompts the customer to re-enter a valid password.   
3. If the system fails to send the confirmation email, it logs the error and displays a message to the customer indicating the issue. The customer can then contact support or try again later.  
  
Use Case Name: Customer Login   
Use Case ID: UC-02   
Actors: Customer, System   
  
Preconditions:   
1. The system is operational.   
2. The customer has already registered an account in the system.   
3. The customer has access to the login interface (e.g., a web form or mobile app).   
  
Postconditions:   
1. The customer is authenticated and logged into their account.   
2. The system displays the customer's dashboard or homepage.   
3. The system records the login event in the system log.   
  
Main Flow:   
1. The customer navigates to the login page.   
2. The system displays the login form (email and password fields).   
3. The customer enters their registered email and password.   
4. The customer submits the login form.   
5. The system verifies the email and password against the database.   
6. The system authenticates the customer and establishes a session.   
7. The system displays the customer's homepage or dashboard.   
  
Alternative Flow:   
1. If the email is not registered, the system displays an error message and prompts the customer to check the email or register first.   
2. If the password is incorrect, the system displays an error message and prompts the customer to re-enter the correct password.   
3. If the system fails to authenticate the customer due to a technical issue, it logs the error and displays a message to the customer. The customer can then try again or contact support.  
  
Use Case Name: Customer Logout   
Use Case ID: UC-03   
Actors: Customer, System   
  
Preconditions:   
1. The system is operational.   
2. The customer is currently logged into their account.   
3. The customer has access to the logout option (e.g., in the account menu or settings).   
  
Postconditions:   
1. The customer's session is terminated.   
2. The customer is redirected to the login or homepage.   
3. The system records the logout event in the system log.   
  
Main Flow:   
1. The customer accesses the account menu or settings.   
2. The system displays the logout option.   
3. The customer clicks on the logout button.   
4. The system confirms the logout action.   
5. The system terminates the customer's session.   
6. The system redirects the customer to the homepage or login page.   
  
Alternative Flow:   
1. If the customer does not have an active session, the system displays a message indicating that the user is not logged in.   
2. If the system fails to terminate the session due to a technical issue, it logs the error and displays a message to the customer. The customer can then refresh the page or contact support.  
  
Use Case Name: Create Customer Account   
Use Case ID: UC-04   
Actors: Customer, System, Administrator   
  
Preconditions:   
1. The system is operational.   
2. The customer is not yet registered in the system.   
3. The customer has access to the account creation interface (e.g., web form or mobile app).   
4. The administrator has configured the account creation functionality.   
  
Postconditions:   
1. A new customer account is created and stored in the system.   
2. The customer receives a confirmation email with instructions to verify their account.   
3. The system logs the account creation event.   
4. The customer is redirected to the login page or dashboard.   
  
Main Flow:   
1. The customer navigates to the account creation page.   
2. The system displays the account creation form.   
3. The customer fills in personal information (e.g., name, email, password, phone number).   
4. The customer submits the form.   
5. The system validates the input data (e.g., unique email, valid password format).   
6. The system generates a unique customer ID and stores the record in the database.   
7. The system sends a confirmation email to the customer's provided email address.   
8. The system logs the account creation event.   
9. The system redirects the customer to the login page.   
  
Alternative Flow:   
1. If the email is already registered, the system displays an error message and prompts the customer to enter a different email.   
2. If the password does not meet the format requirements, the system displays an error message and prompts the customer to re-enter a valid password.   
3. If the system fails to send the confirmation email, it logs the error and displays a message to the customer. The customer may choose to retry or contact support.   
4. If the customer account creation fails due to a database error, the system displays an error message and logs the issue for administrator review.  
  
Use Case Name: View Customer Profile   
Use Case ID: UC-05   
Actors: Customer, System, Administrator   
  
Preconditions:   
1. The system is operational.   
2. The customer is logged into their account.   
3. The customer has an existing profile in the system.   
4. The administrator has access to the customer's data (if viewing on behalf of the customer).   
  
Postconditions:   
1. The customer profile information is displayed to the user.   
2. The system logs the profile viewing event.   
3. The customer can update or view their personal information.   
  
Main Flow:   
1. The customer navigates to their profile section.   
2. The system retrieves the customer's personal information from the database.   
3. The system displays the customer profile (e.g., name, email, phone number, address, order history).   
4. The customer can choose to view or edit specific fields.   
5. The system updates the display based on the customer's selection.   
  
Alternative Flow:   
1. If the system cannot retrieve the profile data due to a database error, it displays an error message and logs the issue for administrator review.   
2. If the customer is not authenticated, the system redirects them to the login page.   
3. If the administrator accesses the profile and lacks permission, the system displays an access denied message and logs the unauthorized attempt.  
  
Use Case Name: Update Personal Information   
Use Case ID: UC-06   
Actors: Customer, System, Administrator   
  
Preconditions:   
1. The system is operational.   
2. The customer is logged into their account.   
3. The customer has an existing account in the system.   
4. The system provides an interface for updating personal information.   
  
Postconditions:   
1. The customer's updated personal information is saved in the system.   
2. The system displays a confirmation message indicating the update was successful.   
3. The system logs the update event.   
  
Main Flow:   
1. The customer navigates to the account settings or profile page.   
2. The system displays the current personal information.   
3. The customer selects the "Edit Information" option.   
4. The system displays the editable form with current data.   
5. The customer modifies the desired fields (e.g., name, phone number, address).   
6. The customer submits the updated information.   
7. The system validates the input (e.g., correct email format, valid phone number).   
8. The system updates the customer's record in the database.   
9. The system displays a confirmation message.   
  
Alternative Flow:   
1. If the customer inputs invalid data (e.g., incorrect email format), the system displays an error message and prompts the customer to correct the input.   
2. If the customer is not authenticated, the system redirects them to the login page.   
3. If the system fails to update the database due to an error, it displays an error message and logs the issue for administrator review.   
4. If the administrator attempts to modify the information without proper permissions, the system displays an access denied message and logs the unauthorized attempt.  
  
Use Case Name: Manage Purchase History   
Use Case ID: UC-07   
Actors: Customer, System, Administrator   
  
Preconditions:   
1. The system is operational.   
2. The customer is logged into their account.   
3. The customer has at least one purchase record in the system.   
4. The administrator has access to customer purchase data for review or management.   
  
Postconditions:   
1. The customer can view their purchase history.   
2. The system allows the customer to filter, sort, or search their purchase records.   
3. The system logs the access or modification of purchase history data.   
4. The administrator can view or manage purchase history records if required.   
  
Main Flow:   
1. The customer navigates to the "Purchase History" section of their account.   
2. The system retrieves the customer's purchase records from the database.   
3. The system displays the purchase history, including details such as order date, product name, quantity, and total cost.   
4. The customer can apply filters (e.g., date range, product category) or sort the list (e.g., by date or order status).   
5. The system updates the display to reflect the applied filters or sorting.   
6. The customer selects an order to view more details.   
7. The system displays the selected order's information, including product details, shipping information, and payment status.   
  
Alternative Flow:   
1. If the system cannot retrieve purchase history data due to a database error, it displays an error message and logs the issue for administrator review.   
2. If the customer is not authenticated, the system redirects them to the login page.   
3. If no purchase history exists for the customer, the system displays a message indicating that there are no past purchases.   
4. If the administrator attempts to access the purchase history without proper permissions, the system displays an access denied message and logs the unauthorized attempt.  
  
Use Case Name: Add Product to Cart   
Use Case ID: UC-08   
Actors: Customer, System, Product, Cart   
  
Preconditions:   
1. The system is operational.   
2. The customer is logged into their account.   
3. The customer has selected a product to add to the cart.   
4. The product is available in the inventory.   
5. The cart functionality is enabled in the system.   
  
Postconditions:   
1. The selected product is added to the customer's cart.   
2. The system updates the cart display to reflect the new item.   
3. The inventory system is updated to reflect the addition to the cart (if inventory tracking is enabled).   
4. The system logs the action for audit purposes.   
  
Main Flow:   
1. The customer browses the product catalog.   
2. The customer selects a product and views its details.   
3. The customer clicks the "Add to Cart" button.   
4. The system checks the product's availability in the inventory.   
5. The system adds the product to the customer's cart.   
6. The system updates the cart display (e.g., item count, total price).   
7. The system logs the addition of the product to the cart.   
8. The customer receives a confirmation message that the product has been added.   
  
Alternative Flow:   
1. If the product is out of stock, the system displays a message indicating unavailability and does not add the product to the cart.   
2. If the customer is not logged in, the system adds the product to a temporary session cart and prompts the customer to log in or continue as a guest.   
3. If the system fails to update the cart due to a technical error, it displays an error message and logs the issue for administrator review.   
4. If the product does not exist in the inventory, the system displays an error message and does not add it to the cart.  
  
Use Case Name: Remove Product from Cart   
Use Case ID: UC-09   
Actors: Customer, System, Product, Cart   
  
Preconditions:   
1. The system is operational.   
2. The customer is logged into their account.   
3. The customer has at least one product in their cart.   
4. The cart functionality is enabled in the system.   
  
Postconditions:   
1. The selected product is removed from the customer's cart.   
2. The cart display is updated to reflect the removal.   
3. The inventory is updated to reflect the return of the product (if inventory tracking is enabled).   
4. The system logs the removal action for audit purposes.   
  
Main Flow:   
1. The customer navigates to the cart page.   
2. The system displays the current cart contents.   
3. The customer selects a product to remove.   
4. The customer clicks the "Remove" button next to the selected product.   
5. The system confirms the removal action with the customer.   
6. The system removes the product from the cart.   
7. The system updates the cart display (e.g., item count, total price).   
8. The system logs the removal of the product.   
9. The customer receives a confirmation message that the product has been removed.   
  
Alternative Flow:   
1. If the product is no longer available in the inventory, the system displays a message indicating the change but still removes the product from the cart.   
2. If the customer clicks "Cancel" during the removal confirmation, the system retains the product in the cart and displays no changes.   
3. If the system fails to update the cart due to a technical error, it displays an error message and logs the issue for administrator review.   
4. If the customer is not authenticated, the system redirects them to the login page before allowing removal of items from the cart.  
  
Use Case Name: View Cart Contents   
Use Case ID: UC-10   
Actors: Customer, System, Cart   
  
Preconditions:   
1. The system is operational.   
2. The customer is logged into their account.   
3. The customer has at least one item in their cart.   
4. The cart functionality is enabled in the system.   
  
Postconditions:   
1. The current cart contents are displayed to the customer.   
2. The system logs the cart viewing action for audit purposes.   
3. The customer can proceed with checkout or modify the cart.   
  
Main Flow:   
1. The customer navigates to the cart page.   
2. The system retrieves the customer's cart items from the database.   
3. The system displays the cart contents, including product name, quantity, price, and total.   
4. The system allows the customer to adjust quantities or remove items.   
5. The system updates the cart display in real-time as changes are made.   
6. The system logs the viewing action.   
  
Alternative Flow:   
1. If the cart is empty, the system displays a message indicating no items are in the cart and suggests browsing products.   
2. If the system fails to retrieve cart data due to a database or technical error, it displays an error message and logs the issue for administrator review.   
3. If the customer is not logged in, the system displays a temporary session cart or prompts the customer to log in or create an account.  
  
Use Case Name: Update Cart Quantity   
Use Case ID: UC-11   
Actors: Customer, System, Cart, Product, Inventory   
  
Preconditions:   
1. The system is operational.   
2. The customer is logged into their account.   
3. The customer has at least one product in their cart.   
4. The cart functionality is enabled in the system.   
5. The product exists in the inventory with sufficient stock.   
  
Postconditions:   
1. The quantity of the selected product in the cart is updated.   
2. The cart display is refreshed to reflect the new quantity and total price.   
3. The inventory is updated to reflect the change in cart quantity (if inventory tracking is enabled).   
4. The system logs the update action for audit purposes.   
  
Main Flow:   
1. The customer navigates to the cart page.   
2. The system displays the current cart items, including their quantities and prices.   
3. The customer selects a product in the cart and modifies its quantity using the quantity input field.   
4. The customer clicks the "Update Quantity" button.   
5. The system checks if the requested quantity is available in the inventory.   
6. The system updates the cart with the new quantity.   
7. The system recalculates the total price and updates the cart display.   
8. The system logs the quantity update action.   
9. The customer receives a confirmation message that the quantity has been updated.   
  
Alternative Flow:   
1. If the requested quantity exceeds the available stock, the system displays an error message and does not update the cart.   
2. If the customer inputs an invalid quantity (e.g., negative number or non-numeric input), the system displays an error message and prompts the customer to enter a valid quantity.   
3. If the system fails to update the cart due to a technical error, it displays an error message and logs the issue for administrator review.   
4. If the product is no longer available in the inventory, the system removes the product from the cart and displays a message indicating the unavailability.  
  
Use Case Name: Checkout Order   
Use Case ID: UC-12   
Actors: Customer, System, Cart, Payment, Order, Inventory   
  
Preconditions:   
1. The system is operational.   
2. The customer is logged into their account.   
3. The customer has at least one product in their cart.   
4. The cart functionality is enabled in the system.   
5. The payment gateway is accessible and configured.   
6. The inventory has sufficient stock for all items in the cart.   
  
Postconditions:   
1. A new order is created in the system.   
2. The payment is processed successfully.   
3. The cart is cleared of all items.   
4. The inventory is updated to reflect the reduction in stock.   
5. The system logs the checkout action for audit purposes.   
6. The customer receives a confirmation message and an order summary via email.   
  
Main Flow:   
1. The customer navigates to the cart page and selects the "Checkout" option.   
2. The system displays the checkout form with shipping and billing information.   
3. The customer fills in or confirms their shipping address and selects a delivery method.   
4. The customer selects a payment method (e.g., credit card, PayPal, etc.).   
5. The system processes the payment through the selected payment gateway.   
6. The system confirms the payment was successful.   
7. The system creates a new order record, including the products, quantities, and total amount.   
8. The system reduces the inventory quantities for the purchased items.   
9. The system clears the cart and sends a confirmation email to the customer.   
10. The system logs the checkout event and displays a success message to the customer.   
  
Alternative Flow:   
1. If the customer's input for shipping information is invalid, the system displays an error message and prompts the customer to correct the data.   
2. If the payment fails due to insufficient funds or incorrect card details, the system displays an error message and allows the customer to retry the payment or choose an alternative method.   
3. If the inventory does not have sufficient stock for an item in the cart, the system displays an error message, updates the cart to reflect the available quantity, or removes the item from the cart.   
4. If the system fails to create the order due to a database error, it displays an error message and logs the issue for administrator review.   
5. If the customer cancels the checkout process, the system returns them to the cart page without making any changes.   
6. If the payment gateway is unavailable, the system displays an error message and logs the issue. The customer is advised to try again later or contact support.  
  
Use Case Name: Confirm Order   
Use Case ID: UC-13   
Actors: Customer, System, Cart, Order, Payment   
  
Preconditions:   
1. The system is operational.   
2. The customer has completed the checkout process.   
3. The system has processed the payment successfully.   
4. The cart contains items ready for confirmation.   
5. The order has been created in the system.   
  
Postconditions:   
1. The customer receives a confirmation of their order.   
2. The order status is updated to "Confirmed" in the system.   
3. The cart is cleared of all items.   
4. The system logs the order confirmation event.   
5. The customer receives an order confirmation email.   
  
Main Flow:   
1. The system processes the payment and generates a new order.   
2. The system displays the order confirmation page to the customer.   
3. The system updates the order status in the database to "Confirmed".   
4. The system clears the customer's cart.   
5. The system sends an order confirmation email to the customer.   
6. The system logs the confirmation action.   
  
Alternative Flow:   
1. If the payment fails after order creation, the system cancels the order, retains the cart items, and displays an error message.   
2. If the system cannot update the order status due to a database error, it displays an error message and logs the issue for administrator review.   
3. If the order confirmation email fails to send, the system logs the error and displays a message to the customer, allowing them to request the email again or contact support.   
4. If the cart cannot be cleared due to a technical issue, the system displays an error message and logs the issue. The customer can proceed to view the cart again later.  
  
Use Case Name: Receive Order Confirmation Email   
Use Case ID: UC-14   
Actors: Customer, System, Email, Order   
  
Preconditions:   
1. The system is operational.   
2. The customer has completed the order confirmation process (UC-13).   
3. The system has created an order and processed the payment successfully.   
4. The email service is accessible and properly configured.   
5. The customer has provided a valid email address during registration or checkout.   
  
Postconditions:   
1. The customer receives an order confirmation email.   
2. The email contains order details such as order ID, product list, total amount, and expected delivery date.   
3. The system logs the email sending event.   
4. The customer can view the confirmation email and access any provided links (e.g., order tracking).   
  
Main Flow:   
1. The system completes the order confirmation process and prepares the email content.   
2. The system generates an email with the order details.   
3. The system sends the order confirmation email to the customer's registered email address.   
4. The system logs the email sending action in the system log.   
5. The customer receives and opens the confirmation email.   
  
Alternative Flow:   
1. If the email service is down or unreachable, the system logs the error and displays a message to the customer, allowing them to request the email again later.   
2. If the customer's email address is invalid or not found, the system logs the issue and displays a message to the customer to update their email in the profile settings.   
3. If the email is sent but not delivered to the customer's inbox (e.g., due to spam filters), the system logs the delivery attempt, and the customer may need to check their spam folder or contact support.   
4. If the customer does not receive the email within a reasonable time, they can manually request a resend via the order details page or contact customer support.  
  
Use Case Name: Admin Login   
Use Case ID: UC-15   
Actors: Administrator, System   
  
Preconditions:   
1. The system is operational.   
2. The administrator has a valid account in the system.   
3. The administrator has access to the admin login interface (e.g., a web form or mobile app).   
  
Postconditions:   
1. The administrator is authenticated and logged into their account.   
2. The system displays the admin dashboard.   
3. The system records the login event in the system log.   
  
Main Flow:   
1. The administrator navigates to the admin login page.   
2. The system displays the login form (email and password fields).   
3. The administrator enters their registered email and password.   
4. The administrator submits the login form.   
5. The system verifies the email and password against the administrator database.   
6. The system authenticates the administrator and establishes a session.   
7. The system displays the admin dashboard.   
  
Alternative Flow:   
1. If the email is not registered as an administrator, the system displays an error message and prompts the administrator to check the email or contact support.   
2. If the password is incorrect, the system displays an error message and prompts the administrator to re-enter the correct password.   
3. If the system fails to authenticate the administrator due to a technical issue, it logs the error and displays a message. The administrator can then try again or contact support.   
4. If the administrator account is locked or disabled, the system denies access and logs the attempt.  
  
Use Case Name: Manage Inventory   
Use Case ID: UC-16   
Actors: Administrator, System, Inventory, Product   
  
Preconditions:   
1. The system is operational.   
2. The administrator is logged into their account with appropriate permissions.   
3. The inventory system is accessible and properly configured.   
4. The product entity exists in the system.   
  
Postconditions:   
1. The inventory levels for one or more products are updated.   
2. The system logs the inventory management action.   
3. The updated inventory data is visible in the admin interface and on the customer-facing product catalog.   
4. Inventory alerts are triggered if stock levels fall below a predefined threshold.   
  
Main Flow:   
1. The administrator navigates to the inventory management section in the admin dashboard.   
2. The system retrieves and displays the current inventory data, including product names, quantities, and stock status.   
3. The administrator selects a product to update its inventory level.   
4. The administrator modifies the quantity or updates the stock status (e.g., in stock, out of stock).   
5. The administrator clicks the "Save Changes" button.   
6. The system validates the input and updates the inventory record in the database.   
7. The system logs the inventory update action.   
8. The system displays a confirmation message indicating the inventory has been successfully updated.   
  
Alternative Flow:   
1. If the administrator inputs an invalid quantity (e.g., negative number or non-numeric input), the system displays an error message and prompts for a valid quantity.   
2. If the system fails to update the inventory due to a database error, it displays an error message and logs the issue for administrator review.   
3. If the selected product does not exist in the inventory, the system displays an error message and prompts the administrator to select a valid product.   
4. If the administrator lacks the necessary permissions to manage inventory, the system displays an access denied message and logs the unauthorized attempt.  
  
Use Case Name: Add New Product   
Use Case ID: UC-17   
Actors: Administrator, System, Product, Inventory, Category   
  
Preconditions:   
1. The system is operational.   
2. The administrator is logged into their account with appropriate permissions.   
3. The product does not yet exist in the system.   
4. The inventory system is accessible and properly configured.   
5. The category entity exists in the system.   
  
Postconditions:   
1. A new product is added to the system with valid attributes.   
2. The inventory is updated to include the new product and its stock quantity.   
3. The product is associated with the correct category.   
4. The system logs the product addition for audit purposes.   
5. The product becomes available for customers to view and add to their cart.   
  
Main Flow:   
1. The administrator navigates to the product management section in the admin dashboard.   
2. The system displays the "Add New Product" interface.   
3. The administrator enters the product details (e.g., name, description, price, category, stock quantity, and image).   
4. The administrator clicks the "Save Product" button.   
5. The system validates the input (e.g., correct price format, valid category, and positive stock quantity).   
6. The system creates a new product record in the database.   
7. The system updates the inventory with the new product and its stock quantity.   
8. The system logs the product addition and displays a confirmation message.   
9. The administrator can view the updated product list or inventory.   
  
Alternative Flow:   
1. If the product name is not unique, the system displays an error message and prompts the administrator to provide a different name.   
2. If the product price is invalid (e.g., negative or non-numeric input), the system displays an error message and prompts for a valid price.   
3. If the selected category does not exist in the system, the system displays an error message and prompts the administrator to select a valid category.   
4. If the system fails to create the product due to a database error, it displays an error message and logs the issue for administrator review.   
5. If the administrator lacks the necessary permissions to add products, the system displays an access denied message and logs the unauthorized attempt.  
  
Use Case Name: Update Product Details   
Use Case ID: UC-18   
Actors: Administrator, System, Product, Inventory, Category   
  
Preconditions:   
1. The system is operational.   
2. The administrator is logged into their account with appropriate permissions.   
3. The product to be updated exists in the system.   
4. The inventory and category systems are accessible and properly configured.   
  
Postconditions:   
1. The product details (e.g., name, description, price, category, stock quantity) are updated in the system.   
2. The inventory is updated to reflect any changes in stock quantity.   
3. The product is associated with the correct category if modified.   
4. The system logs the update action for audit purposes.   
5. The updated product information is visible in the admin interface and the customer-facing product catalog.   
  
Main Flow:   
1. The administrator navigates to the product management section in the admin dashboard.   
2. The system displays a list of products and allows the administrator to search or filter.   
3. The administrator selects a specific product to update its details.   
4. The system retrieves and displays the current product information (e.g., name, price, description, category, stock quantity).   
5. The administrator modifies the desired product details (e.g., price, description, category, stock quantity).   
6. The administrator clicks the "Update Product" button.   
7. The system validates the input (e.g., correct price format, valid category, and positive stock quantity if applicable).   
8. The system updates the product record in the database.   
9. If the stock quantity is updated, the system updates the inventory accordingly.   
10. The system logs the update action and displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the product name is not unique and the administrator attempts to update it, the system displays an error message and prompts for a unique name.   
2. If the product price is invalid (e.g., negative or non-numeric input), the system displays an error message and prompts for a valid price.   
3. If the selected category does not exist in the system, the system displays an error message and prompts the administrator to select a valid category.   
4. If the system fails to update the product due to a database error, it displays an error message and logs the issue for administrator review.   
5. If the administrator lacks the necessary permissions to update products, the system displays an access denied message and logs the unauthorized attempt.  
  
Use Case Name: Delete Product   
Use Case ID: UC-19   
Actors: Administrator, System, Product, Inventory   
  
Preconditions:   
1. The system is operational.   
2. The administrator is logged into their account with appropriate permissions.   
3. The product to be deleted exists in the system.   
4. The inventory system is accessible and properly configured.   
  
Postconditions:   
1. The product is removed from the system.   
2. The inventory record for the product is deleted or marked as obsolete.   
3. The product is no longer visible in the product catalog or admin interface.   
4. The system logs the deletion action for audit purposes.   
  
Main Flow:   
1. The administrator navigates to the product management section in the admin dashboard.   
2. The system displays a list of products and allows the administrator to search or filter.   
3. The administrator selects a specific product to delete.   
4. The system prompts the administrator for confirmation to delete the product.   
5. The administrator confirms the deletion.   
6. The system removes the product record from the database.   
7. The system updates the inventory to remove or mark the product as deleted.   
8. The system logs the deletion action and displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the selected product does not exist in the system, the system displays an error message and prompts the administrator to select a valid product.   
2. If the administrator clicks "Cancel" during the deletion confirmation, the system retains the product and displays no changes.   
3. If the system fails to delete the product due to a database error, it displays an error message and logs the issue for administrator review.   
4. If the administrator lacks the necessary permissions to delete products, the system displays an access denied message and logs the unauthorized attempt.  
  
Use Case Name: Categorize Products   
Use Case ID: UC-20   
Actors: Administrator, System, Product, Category   
  
Preconditions:   
1. The system is operational.   
2. The administrator is logged into their account with appropriate permissions.   
3. The product exists in the system.   
4. The category entity is already defined in the system.   
  
Postconditions:   
1. The product is successfully categorized.   
2. The product is associated with the correct category in the database.   
3. The system logs the categorization action for audit purposes.   
4. The updated product information is visible in the admin interface and the customer-facing product catalog.   
  
Main Flow:   
1. The administrator navigates to the product management section in the admin dashboard.   
2. The system displays a list of products and allows the administrator to search or filter.   
3. The administrator selects a specific product to assign a category.   
4. The system retrieves and displays the current product details, including its existing category (if any).   
5. The administrator chooses a category from the available list.   
6. The administrator clicks the "Save Category" button.   
7. The system validates the category selection.   
8. The system updates the product record in the database with the new category association.   
9. The system logs the categorization action and displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the selected category does not exist in the system, the system displays an error message and prompts the administrator to select a valid category.   
2. If the system fails to update the category due to a database error, it displays an error message and logs the issue for administrator review.   
3. If the administrator lacks the necessary permissions to categorize products, the system displays an access denied message and logs the unauthorized attempt.  
  
Use Case Name: View Product Catalog   
Use Case ID: UC-21   
Actors: Customer, System, Product, Category   
  
Preconditions:   
1. The system is operational.   
2. The customer has access to the product catalog interface (e.g., homepage or dedicated catalog page).   
3. There are at least one or more products and categories in the system.   
  
Postconditions:   
1. The product catalog is displayed to the customer, including product names, descriptions, prices, and images.   
2. Products are organized by categories for easy navigation.   
3. The system logs the catalog viewing action for audit purposes.   
4. The customer can add products to their cart or view product details.   
  
Main Flow:   
1. The customer accesses the product catalog through the homepage or a dedicated catalog page.   
2. The system retrieves the product data and category information from the database.   
3. The system displays the catalog with a list of products grouped by category.   
4. The system allows the customer to search for products or filter by category, price, or other attributes.   
5. The customer selects a product to view its details (e.g., description, price, and availability).   
6. The system displays the product details and provides an "Add to Cart" option.   
7. The system logs the catalog viewing action.   
  
Alternative Flow:   
1. If no products are available in the catalog, the system displays a message indicating the catalog is empty.   
2. If the selected category does not exist or contains no products, the system displays a message suggesting the customer browse other categories.   
3. If the system fails to retrieve product data due to a database or technical error, it displays an error message and logs the issue for administrator review.   
4. If the customer inputs an invalid search term, the system displays an error message and prompts for a valid input.  
  
Use Case Name: Manage User Accounts   
Use Case ID: UC-22   
Actors: Administrator, System, User   
  
Preconditions:   
1. The system is operational.   
2. The administrator is logged into their account with appropriate permissions.   
3. The user account management interface is accessible.   
4. The user entity exists in the system.   
  
Postconditions:   
1. The administrator can create, update, or delete user accounts.   
2. The system logs all user account management actions for audit purposes.   
3. The user account is updated or removed in the database.   
4. The system displays confirmation messages for each action.   
  
Main Flow:   
1. The administrator navigates to the user account management section in the admin dashboard.   
2. The system displays a list of existing user accounts with details such as name, email, account status, and role.   
3. The administrator selects an action (Create, Update, or Delete) for a specific user or initiates a new user account creation.   
4. For a new user, the administrator fills in the required information (e.g., name, email, password, role).   
5. The system validates the input data (e.g., unique email, valid password format).   
6. The system saves the new user account or updates the existing one in the database.   
7. The system logs the action and displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the email is already registered, the system displays an error message and prompts the administrator to provide a different email.   
2. If the password does not meet the format requirements, the system displays an error message and prompts the administrator to re-enter a valid password.   
3. If the system fails to save or update the user account due to a database error, it displays an error message and logs the issue for administrator review.   
4. If the administrator attempts to delete a user account, the system prompts for confirmation before proceeding.   
5. If the administrator clicks "Cancel" during the deletion confirmation, the system retains the user account and displays no changes.   
6. If the administrator lacks the necessary permissions to manage user accounts, the system displays an access denied message and logs the unauthorized attempt.  
  
Use Case Name: Send Email Notification   
Use Case ID: UC-23   
Actors: System, Email, Customer, Administrator   
  
Preconditions:   
1. The system is operational.   
2. The email service is properly configured and accessible.   
3. The recipient (customer or administrator) has a valid email address stored in the system.   
4. A trigger event (e.g., order confirmation, account creation, inventory alert) has occurred.   
  
Postconditions:   
1. The email notification is sent to the recipient's registered email address.   
2. The system logs the email sending event for audit and troubleshooting.   
3. The recipient receives the notification with relevant details (e.g., order status, system alerts).   
4. The system records the success or failure of the notification process.   
  
Main Flow:   
1. A system event occurs (e.g., new order, account confirmation, inventory update).   
2. The system identifies the relevant recipient (e.g., customer, administrator) based on predefined rules or user settings.   
3. The system generates the email content, including subject, message body, and any dynamic data (e.g., order ID, user name).   
4. The system sends the email notification via the configured email service.   
5. The system logs the action as a successful email notification.   
6. The system provides a confirmation message to the user or admin interface if applicable.   
  
Alternative Flow:   
1. If the email service is unavailable, the system logs the error and displays a message indicating the failure. The notification can be resent later.   
2. If the recipient's email address is invalid or missing, the system logs the issue and displays a message prompting the administrator or customer to update their email.   
3. If the system fails to generate the email content due to a technical error, it logs the issue and displays an error message to the administrator for review.   
4. If the email is sent but not delivered to the recipient's inbox (e.g., due to spam filters or network issues), the system logs the delivery attempt, and the recipient may need to check their spam folder or request a resend.  
  
Use Case Name: Install Plugin   
Use Case ID: UC-24   
Actors: Administrator, System, Plugin   
  
Preconditions:   
1. The system is operational.   
2. The administrator is logged into their account with appropriate permissions.   
3. The plugin to be installed is available and compatible with the system.   
4. The system has the capability to accept and process plugin installations.   
  
Postconditions:   
1. The plugin is installed and configured within the system.   
2. The plugin is activated and functional.   
3. The system logs the plugin installation action for audit purposes.   
4. The administrator receives a confirmation message of the successful installation.   
  
Main Flow:   
1. The administrator navigates to the plugin management section in the admin dashboard.   
2. The system displays a list of available plugins or allows the administrator to upload a new plugin.   
3. The administrator selects a plugin for installation or uploads a new one.   
4. The system validates the plugin (e.g., checks for compatibility and required dependencies).   
5. The system installs the plugin and configures it based on default or user-defined settings.   
6. The system activates the plugin and updates the plugin status.   
7. The system logs the plugin installation and displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the plugin is not compatible with the system, the system displays an error message and does not proceed with installation.   
2. If the plugin file is missing or corrupted, the system displays an error message and prompts the administrator to provide a valid file.   
3. If the system fails to install the plugin due to a technical error, it displays an error message and logs the issue for administrator review.   
4. If the administrator lacks the necessary permissions to install plugins, the system displays an access denied message and logs the unauthorized attempt.  
  
Use Case Name: Configure Plugin Settings   
Use Case ID: UC-25   
Actors: Administrator, System, Plugin   
  
Preconditions:   
1. The system is operational.   
2. The administrator is logged into their account with appropriate permissions.   
3. The plugin is already installed and active in the system.   
4. The plugin settings interface is accessible.   
  
Postconditions:   
1. The plugin settings are updated to the desired configuration.   
2. The system logs the configuration action for audit purposes.   
3. The updated settings take effect in the system.   
4. The administrator receives a confirmation of the changes.   
  
Main Flow:   
1. The administrator navigates to the plugin management section in the admin dashboard.   
2. The system displays a list of installed plugins and their current status.   
3. The administrator selects the plugin for which they want to configure settings.   
4. The system displays the plugin configuration interface with available options.   
5. The administrator modifies the plugin settings (e.g., enable/disable features, update API keys, or adjust parameters).   
6. The administrator clicks the "Save Settings" button.   
7. The system validates the input and updates the plugin configuration in the database.   
8. The system logs the configuration change and displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the input values are invalid (e.g., incorrect format, missing required fields), the system displays an error message and prompts the administrator to correct the input.   
2. If the plugin configuration file is missing or corrupted, the system displays an error message and logs the issue for administrator review.   
3. If the system fails to update the plugin settings due to a technical error, it displays an error message and logs the issue for troubleshooting.   
4. If the administrator lacks the necessary permissions to configure plugins, the system displays an access denied message and logs the unauthorized attempt.  
  
Use Case Name: View Purchase History   
Use Case ID: UC-14   
Actors: Customer, System, Order, Purchase History   
  
Preconditions:   
1. The system is operational.   
2. The customer is logged into their account.   
3. The customer has at least one completed purchase in the system.   
4. The purchase history functionality is enabled in the system.   
  
Postconditions:   
1. The customer's purchase history is displayed on the screen.   
2. The system logs the access to the purchase history for audit purposes.   
3. The customer can view order details, such as order date, total amount, and product information.   
4. The system ensures the data is accurate and up to date.   
  
Main Flow:   
1. The customer navigates to the "Purchase History" section in their account.   
2. The system retrieves the customer's purchase history from the database.   
3. The system displays the list of past orders, including key details like order ID, date, total amount, and order status.   
4. The customer selects an order to view its details.   
5. The system shows the selected order's information, such as product names, quantities, and payment method.   
6. The system logs the viewing of the purchase history.   
  
Alternative Flow:   
1. If the customer has no purchase history, the system displays a message indicating no past purchases.   
2. If the system cannot retrieve the purchase history due to a database error, it displays an error message and logs the issue for administrator review.   
3. If the customer is not authenticated, the system redirects them to the login page.   
4. If the customer attempts to access another user's purchase history, the system displays an access denied message and logs the unauthorized attempt.  
  
Use Case Name: Manage Payment Information   
Use Case ID: UC-15   
Actors: Customer, System, Payment   
  
Preconditions:   
1. The system is operational.   
2. The customer is logged into their account.   
3. The customer has at least one payment method stored in the system or can add a new one.   
4. The payment functionality is enabled in the system.   
  
Postconditions:   
1. The customer can view, add, update, or remove payment methods.   
2. The system logs the action for audit purposes.   
3. The updated or new payment information is securely stored in the system.   
4. The customer receives a confirmation message for each action.   
  
Main Flow:   
1. The customer navigates to the "Payment Information" section in their account settings.   
2. The system retrieves and displays the customer's current payment methods (if any).   
3. The customer selects an action (Add, Update, or Remove) for a payment method.   
4. If adding a new payment method, the customer enters the required details (e.g., card number, expiration date, CVV).   
5. The system validates the input (e.g., correct card format, valid expiration date).   
6. The system securely stores the payment method in the database.   
7. The system updates the displayed payment methods.   
8. The system logs the action and displays a confirmation message to the customer.   
  
Alternative Flow:   
1. If the customer enters invalid payment information (e.g., incorrect card number or expired date), the system displays an error message and prompts for valid input.   
2. If the customer attempts to remove the only active payment method, the system displays a warning and prompts the customer to add a new method before proceeding.   
3. If the system fails to save or update the payment method due to a database error, it displays an error message and logs the issue for administrator review.   
4. If the customer is not authenticated, the system redirects them to the login page before allowing any action on payment information.   
5. If the customer attempts to manage another user’s payment information, the system displays an access denied message and logs the unauthorized attempt.  
  
Use Case Name: Update Payment Method   
Use Case ID: UC-16   
Actors: Customer, System, Payment   
  
Preconditions:   
1. The system is operational.   
2. The customer is logged into their account.   
3. The customer has at least one existing payment method in the system.   
4. The payment functionality is enabled in the system.   
  
Postconditions:   
1. The customer's selected payment method is updated with new or modified information.   
2. The system logs the update action for audit purposes.   
3. The updated payment method is securely stored in the system.   
4. The customer receives a confirmation message that the update was successful.   
  
Main Flow:   
1. The customer navigates to the "Payment Information" section in their account settings.   
2. The system displays the current list of stored payment methods.   
3. The customer selects the payment method they wish to update.   
4. The system displays an editable form with the current payment method details.   
5. The customer modifies the necessary fields (e.g., card number, expiration date, CVV).   
6. The customer clicks the "Update Payment Method" button.   
7. The system validates the input data (e.g., correct card format, valid expiration date).   
8. The system updates the payment method record in the database.   
9. The system logs the update action and displays a confirmation message to the customer.   
  
Alternative Flow:   
1. If the customer enters invalid payment information (e.g., incorrect card number or expired date), the system displays an error message and prompts for valid input.   
2. If the system fails to update the payment method due to a database error, it displays an error message and logs the issue for administrator review.   
3. If the customer is not authenticated, the system redirects them to the login page before allowing any changes to their payment method.   
4. If the customer attempts to update another user’s payment method, the system displays an access denied message and logs the unauthorized attempt.  
  
Use Case Name: View Order Details   
Use Case ID: UC-17   
Actors: Customer, System, Order   
  
Preconditions:   
1. The system is operational.   
2. The customer is logged into their account.   
3. The customer has at least one order in the system.   
4. The order details functionality is enabled in the system.   
  
Postconditions:   
1. The order details are displayed to the customer, including product list, total cost, payment status, and shipping information.   
2. The system logs the viewing of the order details for audit purposes.   
3. The customer can take further actions, such as updating the order status or initiating a return.   
  
Main Flow:   
1. The customer navigates to the "Order Details" section from the purchase history or order confirmation page.   
2. The system retrieves the selected order from the database.   
3. The system displays the order details, including order ID, date, product information, quantity, price, total amount, and payment status.   
4. The customer can view shipping and billing information associated with the order.   
5. The system logs the viewing action.   
  
Alternative Flow:   
1. If the selected order does not exist in the system, the system displays an error message and prompts the customer to select a valid order.   
2. If the system cannot retrieve the order details due to a database error, it displays an error message and logs the issue for administrator review.   
3. If the customer is not authenticated, the system redirects them to the login page before allowing access to order details.   
4. If the customer attempts to access another user’s order details, the system displays an access denied message and logs the unauthorized attempt.  
  
Use Case Name: Cancel Order   
Use Case ID: UC-18   
Actors: Customer, System, Order   
  
Preconditions:   
1. The system is operational.   
2. The customer is logged into their account.   
3. The customer has at least one active order in the system.   
4. The order is in a cancellable state (e.g., pending or processing).   
  
Postconditions:   
1. The selected order is marked as canceled in the system.   
2. The system logs the cancellation action for audit purposes.   
3. The inventory is updated to reflect the return of canceled items (if inventory tracking is enabled).   
4. The customer receives a confirmation message of the cancellation.   
  
Main Flow:   
1. The customer navigates to the "Purchase History" or "Order Details" section.   
2. The system displays a list of active orders.   
3. The customer selects an order to cancel.   
4. The system confirms the cancellation action with the customer.   
5. The customer confirms the cancellation.   
6. The system updates the order status to "Canceled".   
7. The system logs the cancellation action.   
8. The system updates the inventory to reflect the return of the items (if applicable).   
9. The customer receives a confirmation message that the order has been canceled.   
  
Alternative Flow:   
1. If the selected order is not in a cancellable state (e.g., already shipped or completed), the system displays an error message and does not allow cancellation.   
2. If the system fails to update the order status due to a database error, it displays an error message and logs the issue for administrator review.   
3. If the customer clicks "Cancel" during the cancellation confirmation, the system retains the order and displays no changes.   
4. If the customer is not authenticated, the system redirects them to the login page before allowing any action on the order.   
5. If the system fails to update the inventory, it logs the error and displays a message to the customer or administrator for review.  
  
Use Case Name: Manage Category Structure   
Use Case ID: UC-20   
Actors: Administrator, System, Category, Product   
  
Preconditions:   
1. The system is operational.   
2. The administrator is logged into their account with appropriate permissions.   
3. The category management interface is accessible.   
4. The system contains existing categories and products.   
  
Postconditions:   
1. The category structure is updated (e.g., categories are added, removed, or modified).   
2. The system logs the category management action for audit purposes.   
3. Products are properly associated with their updated categories.   
4. The changes to the category structure are reflected in the product catalog and admin interface.   
  
Main Flow:   
1. The administrator navigates to the "Category Management" section in the admin dashboard.   
2. The system displays the current category structure, including existing categories and their associated products.   
3. The administrator selects an action (Add, Edit, or Delete) for a category.   
4. If adding a new category, the administrator fills in the required details (e.g., category name and description).   
5. The system validates the input and updates the category structure in the database.   
6. If editing or deleting a category, the system updates the category or its associations accordingly.   
7. The system logs the action and displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the category name is not unique, the system displays an error message and prompts the administrator to provide a different name.   
2. If the system fails to update the category structure due to a database error, it displays an error message and logs the issue for administrator review.   
3. If the administrator attempts to delete a category that is currently associated with products, the system displays a warning and suggests updating product associations before deletion.   
4. If the administrator lacks the necessary permissions to manage categories, the system displays an access denied message and logs the unauthorized attempt.  
  
Use Case Name: Track Inventory Levels   
Use Case ID: UC-20   
Actors: Administrator, System, Inventory, Product   
  
Preconditions:   
1. The system is operational.   
2. The administrator is logged into their account with appropriate permissions.   
3. The inventory system is accessible and properly configured.   
4. There are products with defined inventory levels in the system.   
  
Postconditions:   
1. The administrator can view real-time inventory levels for all products.   
2. The system logs the inventory tracking action for audit purposes.   
3. Inventory alerts are triggered when stock levels fall below a predefined threshold.   
4. The updated inventory data is visible in the admin interface and product catalog.   
  
Main Flow:   
1. The administrator navigates to the "Inventory Tracking" section in the admin dashboard.   
2. The system retrieves and displays inventory data, including product names, stock levels, and status (e.g., in stock, low stock, out of stock).   
3. The administrator can filter or search for specific products.   
4. The system updates the displayed inventory data based on the administrator’s input.   
5. The system highlights products with stock levels below the threshold.   
6. The system logs the tracking action and displays a summary of the inventory status.   
  
Alternative Flow:   
1. If the inventory data cannot be retrieved due to a database error, the system displays an error message and logs the issue for administrator review.   
2. If no products are currently in the inventory, the system displays a message indicating the inventory is empty.   
3. If the administrator lacks the necessary permissions to track inventory, the system displays an access denied message and logs the unauthorized attempt.  
  
Use Case Name: Generate Sales Report   
Use Case ID: UC-20   
Actors: Administrator, System, Order, Product, Purchase History   
  
Preconditions:   
1. The system is operational.   
2. The administrator is logged into their account with appropriate permissions.   
3. The sales report generation functionality is enabled in the system.   
4. There are existing orders and product data in the system.   
  
Postconditions:   
1. A sales report is generated and displayed to the administrator.   
2. The report includes relevant data such as total sales, number of orders, top-selling products, and revenue by time period.   
3. The system logs the report generation action for audit purposes.   
4. The administrator can download or export the sales report in a chosen format (e.g., PDF, Excel).   
  
Main Flow:   
1. The administrator navigates to the "Sales Reports" section in the admin dashboard.   
2. The system displays options to filter the report by date range, product category, or other criteria.   
3. The administrator selects the desired parameters for the report.   
4. The administrator clicks the "Generate Report" button.   
5. The system retrieves and aggregates the necessary sales data from the database.   
6. The system generates a sales report with the selected filters.   
7. The system displays the report in the admin interface and provides a download option.   
8. The system logs the generation of the report.   
  
Alternative Flow:   
1. If the selected date range is invalid or no orders match the criteria, the system displays a message indicating no data is available for the report.   
2. If the system fails to retrieve sales data due to a database error, it displays an error message and logs the issue for administrator review.   
3. If the administrator lacks the necessary permissions to generate sales reports, the system displays an access denied message and logs the unauthorized attempt.  
  
Use Case Name: Modify User Permissions   
Use Case ID: UC-20   
Actors: Administrator, System, User   
  
Preconditions:   
1. The system is operational.   
2. The administrator is logged into their account with the appropriate permissions to modify user roles.   
3. The target user account exists in the system.   
4. The user permission management interface is accessible.   
  
Postconditions:   
1. The user's permissions are updated in the system.   
2. The system logs the permission modification action for audit purposes.   
3. The updated permissions are reflected in the user's access to system features.   
4. The administrator receives a confirmation message that the changes were successfully applied.   
  
Main Flow:   
1. The administrator navigates to the "User Management" section in the admin dashboard.   
2. The system displays a list of user accounts along with their current roles and permissions.   
3. The administrator selects the user account to modify permissions.   
4. The system displays the user's current permissions and available permission options.   
5. The administrator adjusts the permissions (e.g., grants or revokes access to specific features).   
6. The administrator clicks the "Save Permissions" button.   
7. The system validates the permission settings and updates the user's record in the database.   
8. The system logs the modification and displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the selected user does not exist in the system, the system displays an error message and prompts the administrator to select a valid user.   
2. If the system fails to update the user's permissions due to a database error, it displays an error message and logs the issue for administrator review.   
3. If the administrator lacks the necessary permissions to modify user roles, the system displays an access denied message and logs the unauthorized attempt.   
4. If the input for permissions is invalid (e.g., conflicting roles or missing required permissions), the system displays an error message and prompts the administrator to correct the input.  
  
Use Case Name: Access System Settings   
Use Case ID: UC-20   
Actors: Administrator, System   
  
Preconditions:   
1. The system is operational.   
2. The administrator is logged into their account with appropriate permissions.   
3. The system settings interface is accessible in the admin dashboard.   
  
Postconditions:   
1. The administrator can view and modify system settings.   
2. The system logs the access and modification of settings for audit purposes.   
3. The updated settings take effect in the system.   
4. The administrator receives a confirmation message of the changes.   
  
Main Flow:   
1. The administrator navigates to the "System Settings" section in the admin dashboard.   
2. The system displays the current system settings, such as site configuration, payment gateway settings, email service settings, and security options.   
3. The administrator selects a setting to modify.   
4. The system displays an editable form with the current values.   
5. The administrator updates the selected settings.   
6. The administrator clicks the "Save Settings" button.   
7. The system validates the input and updates the settings in the database.   
8. The system logs the action and displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the input values are invalid (e.g., incorrect format or missing required fields), the system displays an error message and prompts the administrator to correct the input.   
2. If the system fails to save the settings due to a database error, it displays an error message and logs the issue for administrator review.   
3. If the administrator lacks the necessary permissions to access system settings, the system displays an access denied message and logs the unauthorized attempt.  
  
Use Case Name: View System Logs   
Use Case ID: UC-20   
Actors: Administrator, System   
  
Preconditions:   
1. The system is operational.   
2. The administrator is logged into their account with appropriate permissions.   
3. The system log functionality is enabled and accessible.   
4. There are existing log entries in the system.   
  
Postconditions:   
1. The system logs are displayed to the administrator, including timestamps, event types, and relevant details.   
2. The system logs the access to the system logs for audit purposes.   
3. The administrator can filter or search the logs based on specific criteria (e.g., date, user, or event type).   
4. The system ensures the logs are accurate and up to date.   
  
Main Flow:   
1. The administrator navigates to the "System Logs" section in the admin dashboard.   
2. The system retrieves the system log entries from the database.   
3. The system displays the log entries in a structured format (e.g., table or list).   
4. The administrator can apply filters (e.g., date range, user, or event type) to narrow down the displayed logs.   
5. The system updates the log display based on the selected filters.   
6. The system logs the access to the system logs.   
  
Alternative Flow:   
1. If no log entries match the selected filters, the system displays a message indicating no logs were found.   
2. If the system cannot retrieve the logs due to a database or technical error, it displays an error message and logs the issue for administrator review.   
3. If the administrator lacks the necessary permissions to view system logs, the system displays an access denied message and logs the unauthorized attempt.  
  
Use Case Name: Manage System Logs   
Use Case ID: UC-26   
Actors: Administrator, System   
  
Preconditions:   
1. The system is operational.   
2. The administrator is logged into their account with appropriate permissions to manage system logs.   
3. The system logs are stored in the database.   
4. The system provides a log management interface in the admin dashboard.   
  
Postconditions:   
1. The administrator can perform actions such as viewing, filtering, exporting, or clearing system logs.   
2. The system logs the management action for audit purposes.   
3. The system retains or removes log entries based on the administrator’s actions.   
4. The log data is available in the desired format for analysis or reporting.   
  
Main Flow:   
1. The administrator navigates to the "System Logs Management" section in the admin dashboard.   
2. The system displays a list of system log entries, including event type, timestamp, user involved, and a summary of the action.   
3. The administrator selects an action to perform (e.g., filter logs by date, export logs, or clear logs).   
4. If filtering, the administrator specifies criteria (e.g., date range, user, or event type).   
5. The system applies the filter and displays the resulting log entries.   
6. If exporting, the system generates the log data in the selected format (e.g., CSV, PDF).   
7. If clearing logs, the system prompts the administrator for confirmation.   
8. The administrator confirms the clearing of logs.   
9. The system removes the selected log entries from the database.   
10. The system logs the management action and displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If no logs are found based on the administrator’s filter criteria, the system displays a message indicating no logs were found.   
2. If the system fails to export the logs due to a technical error, it displays an error message and logs the issue for administrator review.   
3. If the administrator cancels the log clearing confirmation, the system retains all logs and displays no changes.   
4. If the system fails to clear the logs due to a database error, it displays an error message and logs the issue for troubleshooting.   
5. If the administrator lacks the necessary permissions to manage system logs, the system displays an access denied message and logs the unauthorized attempt.  
  
Use Case Name: Manage Payment Method   
Use Case ID: UC-27   
Actors: Administrator, System, Payment Method, Customer   
  
Preconditions:   
1. The system is operational.   
2. The administrator is logged into their account with appropriate permissions to manage payment methods.   
3. The payment method management interface is accessible in the admin dashboard.   
4. The system contains a list of existing payment methods or allows the addition of new ones.   
  
Postconditions:   
1. The administrator can create, update, or delete payment methods in the system.   
2. The system logs all payment method management actions for audit purposes.   
3. The payment method database is updated accordingly.   
4. The system ensures the changes are reflected in the customer-facing checkout interface.   
  
Main Flow:   
1. The administrator navigates to the "Payment Method Management" section in the admin dashboard.   
2. The system displays a list of existing payment methods along with their status and configuration.   
3. The administrator selects an action (Add, Edit, or Delete) for a payment method.   
4. If adding a new payment method, the administrator fills in the required details (e.g., payment method name, description, and integration settings).   
5. The system validates the input and creates or updates the payment method in the database.   
6. If editing or deleting a payment method, the system updates or removes the method accordingly.   
7. The system logs the action and displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the payment method name is not unique, the system displays an error message and prompts the administrator to provide a different name.   
2. If the system fails to update the payment method due to a database error, it displays an error message and logs the issue for administrator review.   
3. If the administrator attempts to delete a payment method that is currently in use, the system displays a warning and suggests updating associated orders or settings before deletion.   
4. If the administrator lacks the necessary permissions to manage payment methods, the system displays an access denied message and logs the unauthorized attempt.