项目文档

# Functional Requirement

# 1. Functional Requirements  
  
## 1.1 Product Creation Function   
\*\*Function ID\*\*: FR-01   
\*\*Description\*\*: The Administrator can create a new Product in the system. The system validates the input data and uses a Plugin for additional configurations or validations.   
\*\*Input\*\*: Product information including name, description, price, category, and status.   
\*\*Output\*\*: A new Product record stored in the database with a generated ProductID.   
  
---  
  
## 1.2 Product Update Function   
\*\*Function ID\*\*: FR-02   
\*\*Description\*\*: The Administrator can update an existing Product's details. The system validates the input and uses a Plugin for additional logic.   
\*\*Input\*\*: Updated Product information including name, description, price, category, and status, along with the ProductID.   
\*\*Output\*\*: The updated Product record stored in the database.   
  
---  
  
## 1.3 Product Deletion Function   
\*\*Function ID\*\*: FR-03   
\*\*Description\*\*: The Administrator can delete an existing Product. The system checks for dependencies and uses a Plugin for additional validations.   
\*\*Input\*\*: ProductID of the Product to be deleted and a confirmation request.   
\*\*Output\*\*: The Product is removed from the database, and the deletion is recorded.   
  
---  
  
## 1.4 Product Viewing Function   
\*\*Function ID\*\*: FR-04   
\*\*Description\*\*: The Administrator or Customer can view the details of a specific Product.   
\*\*Input\*\*: ProductID of the Product to be viewed.   
\*\*Output\*\*: Displayed Product details including name, price, description, and availability.   
  
---  
  
## 1.5 Product Inventory Management Function   
\*\*Function ID\*\*: FR-05   
\*\*Description\*\*: The Administrator can adjust the inventory level of a Product. The system validates the new quantity and uses a Plugin to process additional inventory logic.   
\*\*Input\*\*: ProductID and the new inventory quantity.   
\*\*Output\*\*: Updated InventoryRecord stored in the database, and a Notification sent to the Administrator.   
  
---  
  
## 1.6 Customer Registration Function   
\*\*Function ID\*\*: FR-06   
\*\*Description\*\*: The Administrator can register a new Customer in the system.   
\*\*Input\*\*: Customer information including name, email, address, and status.   
\*\*Output\*\*: A new Customer record stored in the database with a generated CustomerID.   
  
---  
  
## 1.7 Customer Profile Viewing Function   
\*\*Function ID\*\*: FR-07   
\*\*Description\*\*: The Administrator can view the profile of an existing Customer, including order history.   
\*\*Input\*\*: CustomerID of the Customer to be viewed.   
\*\*Output\*\*: Displayed Customer details including name, email, address, and order history.   
  
---  
  
## 1.8 Customer Information Update Function   
\*\*Function ID\*\*: FR-08   
\*\*Description\*\*: The Administrator can update an existing Customer's information.   
\*\*Input\*\*: Updated Customer information including name, email, address, and status, along with the CustomerID.   
\*\*Output\*\*: The updated Customer record stored in the database.   
  
---  
  
## 1.9 Customer Account Deletion Function   
\*\*Function ID\*\*: FR-09   
\*\*Description\*\*: The Administrator can delete a Customer account. The system checks for active Orders and archives or deletes associated data.   
\*\*Input\*\*: CustomerID of the account to be deleted and a confirmation request.   
\*\*Output\*\*: The Customer account and associated data are removed or archived, and the deletion is recorded in the database.   
  
---  
  
## 1.10 Order Placement Function   
\*\*Function ID\*\*: FR-10   
\*\*Description\*\*: The Customer can place an Order by selecting Products and providing payment and shipping information. The system validates the order and uses a Plugin for payment processing.   
\*\*Input\*\*: CustomerID, list of ProductIDs with quantities, shipping information, and payment information.   
\*\*Output\*\*: A new Order record stored in the database with a generated OrderID, and updated InventoryRecords if applicable.   
  
---  
  
## 1.11 Order Viewing Function   
\*\*Function ID\*\*: FR-11   
\*\*Description\*\*: The Administrator or Customer can view the details of an existing Order, including associated Products, Payment status, and shipping information.   
\*\*Input\*\*: OrderID of the Order to be viewed.   
\*\*Output\*\*: Displayed Order details including order number, date, customer name, total amount, and status.   
  
---  
  
## 1.12 Order Status Update Function   
\*\*Function ID\*\*: FR-12   
\*\*Description\*\*: The Administrator can update the status of an Order. The system uses a Plugin to process status-specific logic and sends Notifications.   
\*\*Input\*\*: OrderID and the new status.   
\*\*Output\*\*: The updated Order status stored in the database and a Notification sent to the Administrator and Customer.   
  
---  
  
## 1.13 Order Cancellation Function   
\*\*Function ID\*\*: FR-13   
\*\*Description\*\*: The Administrator or Customer can cancel an Order if it is eligible. The system uses a Plugin to process cancellation logic and adjusts inventory if necessary.   
\*\*Input\*\*: OrderID and a confirmation request.   
\*\*Output\*\*: The Order is marked as cancelled in the database, inventory is updated if applicable, and a Notification is sent to the Administrator and Customer.   
  
---  
  
## 1.14 Payment Processing Function   
\*\*Function ID\*\*: FR-14   
\*\*Description\*\*: The system processes payment for an Order using a Plugin. It verifies the payment and updates the Payment and Order status accordingly.   
\*\*Input\*\*: OrderID and payment information (e.g., card details, payment method).   
\*\*Output\*\*: A new Payment record stored in the database and the Order status updated to reflect the completed payment.   
  
---  
  
## 1.15 Payment History Viewing Function   
\*\*Function ID\*\*: FR-15   
\*\*Description\*\*: The Administrator or Customer can view their payment history.   
\*\*Input\*\*: User's identity (AdministratorID or CustomerID).   
\*\*Output\*\*: Displayed list of Payment records including amount, date, status, and associated Order.   
  
---  
  
## 1.16 Administrator Account Management Function   
\*\*Function ID\*\*: FR-16   
\*\*Description\*\*: The Administrator can manage their own account by updating information such as email, password, and role.   
\*\*Input\*\*: Updated Administrator information including name, email, password, and role, along with the AdminID.   
\*\*Output\*\*: The updated Administrator account stored in the database and a Notification sent to the Administrator.   
  
---  
  
## 1.17 Notification Sending Function   
\*\*Function ID\*\*: FR-17   
\*\*Description\*\*: The system sends Notifications to Administrators or Customers based on system events, using a Plugin to determine the delivery channel and format.   
\*\*Input\*\*: Notification message, recipient (AdminID or CustomerID), and event context.   
\*\*Output\*\*: A new Notification record stored in the database and the message sent to the recipient.   
  
---  
  
## 1.18 Notification Viewing Function   
\*\*Function ID\*\*: FR-18   
\*\*Description\*\*: The Administrator or Customer can view Notifications associated with their account.   
\*\*Input\*\*: User's identity (AdminID or CustomerID).   
\*\*Output\*\*: Displayed list of Notifications including message, timestamp, and related entity (e.g., Product, Order).   
  
---  
  
## 1.19 Plugin Management Function   
\*\*Function ID\*\*: FR-19   
\*\*Description\*\*: The Administrator can activate, deactivate, or configure a Plugin.   
\*\*Input\*\*: PluginID and requested action (activate, deactivate, configure), along with new configuration values if applicable.   
\*\*Output\*\*: Updated Plugin status or configuration stored in the database and a Notification sent to the Administrator.   
  
---  
  
## 1.20 Plugin Installation Function   
\*\*Function ID\*\*: FR-20   
\*\*Description\*\*: The Administrator can install a new Plugin into the system. The system checks for compatibility and logs the installation.   
\*\*Input\*\*: Plugin file or identifier.   
\*\*Output\*\*: A new Plugin record stored in the database with a generated PluginID and a Notification sent to the Administrator.   
  
---  
  
## 1.21 Plugin Uninstallation Function   
\*\*Function ID\*\*: FR-21   
\*\*Description\*\*: The Administrator can uninstall an existing Plugin. The system checks for dependencies and removes the Plugin configuration.   
\*\*Input\*\*: PluginID and a confirmation request.   
\*\*Output\*\*: The Plugin is removed from the system and its configuration deleted from the database, with a Notification sent to the Administrator.   
  
---  
  
## 1.22 Plugin Configuration Function   
\*\*Function ID\*\*: FR-22   
\*\*Description\*\*: The Administrator can configure the settings of an installed Plugin.   
\*\*Input\*\*: PluginID and new configuration values (e.g., API keys, behavior parameters).   
\*\*Output\*\*: Updated PluginSetting records stored in the database and a Notification sent to the Administrator.   
  
---  
  
## 1.23 OrderItem Management Function   
\*\*Function ID\*\*: FR-23   
\*\*Description\*\*: The Administrator can modify an OrderItem (e.g., change quantity or remove the item) within an Order. The system validates the input and recalculates the Order total if necessary.   
\*\*Input\*\*: OrderID, OrderItemID, and updated quantity or removal request.   
\*\*Output\*\*: Updated OrderItem record stored in the database, Order total recalculated if necessary, and a Notification sent to the Administrator and Customer.   
  
---  
  
## 1.24 InventoryRecord Management Function   
\*\*Function ID\*\*: FR-24   
\*\*Description\*\*: The Administrator can modify the inventory record of a Product. The system validates the input and uses a Plugin for additional logic (e.g., audit trails, alerts).   
\*\*Input\*\*: InventoryID, updated quantity, and additional notes (e.g., reason for change).   
\*\*Output\*\*: Updated InventoryRecord stored in the database and a Notification sent to the Administrator.

# External Description

# 2. External Interfaces   
  
This section describes the external interfaces of the system, including the user interfaces, hardware interfaces, software interfaces, and communication interfaces. These interfaces define the system's interactions with external entities such as users, hardware, software components, and communication channels.   
  
## 2.1 User Interfaces   
  
The system interacts with two primary user roles: \*\*Administrator\*\* and \*\*Customer\*\*. Each role has specific access and interaction patterns with the system.   
  
- \*\*Administrator Interface\*\*:   
 - \*\*Description\*\*: This interface allows the Administrator to perform product and customer management, order processing, and plugin configuration.   
 - \*\*Interaction Method\*\*: Web-based graphical user interface (GUI) with login authentication.   
 - \*\*Inputs\*\*:   
 - Product creation/update/deletion: Name, Description, Price, Category, Status, and ProductID.   
 - Customer registration/update/deletion: Name, Email, Address, Status, and CustomerID.   
 - Order placement/status update/cancellation: CustomerID, ProductID with quantity, shipping information, payment information, OrderID, and new status.   
 - Plugin installation/configuration/management: Plugin file or identifier, PluginID, configuration values, and action (e.g., activate, deactivate).   
 - \*\*Outputs\*\*:   
 - Confirmation messages for product/customer/order/plugin operations.   
 - Displayed product/customer/order details.   
 - Notification history displayed for the Administrator.   
  
- \*\*Customer Interface\*\*:   
 - \*\*Description\*\*: This interface allows the Customer to view product details, place orders, view order status, and check their payment and notification history.   
 - \*\*Interaction Method\*\*: Web-based GUI with login authentication.   
 - \*\*Inputs\*\*:   
 - Product viewing: ProductID.   
 - Order placement: ProductID with quantity, shipping information, payment information, and CustomerID.   
 - Order viewing: OrderID and CustomerID.   
 - Payment history viewing: CustomerID.   
 - Notification viewing: CustomerID.   
 - Order cancellation: OrderID and confirmation request.   
 - \*\*Outputs\*\*:   
 - Product details displayed to the Customer.   
 - Order status and history displayed to the Customer.   
 - Payment history displayed to the Customer.   
 - Notification history displayed to the Customer.   
  
## 2.2 Hardware Interfaces   
  
No direct hardware interfaces are required for the system as it is a software-based application that does not interact with physical hardware components.   
  
## 2.3 Software Interfaces   
  
The system interacts with several software components and databases to store, retrieve, and manage data.   
  
- \*\*Database Interface\*\*:   
 - \*\*Description\*\*: The system maintains a central database to store all records of Products, Customers, Orders, Inventory, Plugins, and Notifications.   
 - \*\*Interaction Method\*\*: SQL or ORM-based queries for data insertion, retrieval, update, and deletion.   
 - \*\*Inputs/Outputs\*\*:   
 - \*\*Inputs\*\*: Data such as ProductID, CustomerID, OrderID, PluginID, InventoryID, and their associated attributes (e.g., name, price, status).   
 - \*\*Outputs\*\*: Database records including:   
 - Product records (with generated ProductID).   
 - Customer records (with generated CustomerID).   
 - Order records (with generated OrderID and updated status).   
 - Inventory records (with updated quantity and timestamp).   
 - Plugin records (with generated PluginID and configuration details).   
 - Notification records (with message content, timestamp, and recipient details).   
  
- \*\*Plugin Interface\*\*:   
 - \*\*Description\*\*: The system supports the integration and execution of external plugins for additional logic, validations, and processing.   
 - \*\*Interaction Method\*\*: Plugin is invoked via a defined API or event trigger. It receives input data and returns processed output to the system.   
 - \*\*Inputs/Outputs\*\*:   
 - \*\*Inputs\*\*: PluginID, configuration values, system event context (e.g., order status change, product deletion).   
 - \*\*Outputs\*\*: Processed results such as validation outcomes, payment confirmation, audit trail logs, or alerts.   
 - \*\*Usage in Functional Requirements\*\*:   
 - Used in FR-01 (Product creation) for additional configurations or validations.   
 - Used in FR-02 (Product update) for additional logic.   
 - Used in FR-03 (Product deletion) for additional validations.   
 - Used in FR-05 (Inventory management) for processing inventory logic.   
 - Used in FR-10 (Order placement) for payment processing.   
 - Used in FR-12 (Order status update) for status-specific logic and sending notifications.   
 - Used in FR-13 (Order cancellation) for processing cancellation logic and inventory adjustments.   
 - Used in FR-17 (Notification sending) for determining delivery channels and message format.   
 - Used in FR-24 (InventoryRecord management) for audit trails or alerts.   
  
## 2.4 Communication Interfaces   
  
The system communicates with external systems and users via several channels to provide notifications, process payments, and ensure system integration.   
  
- \*\*Notification Communication Interface\*\*:   
 - \*\*Description\*\*: The system sends notifications to Administrators and Customers through one or more communication channels, such as email or in-app messages.   
 - \*\*Interaction Method\*\*: The system logs the notification in the database and sends it via the selected communication channel.   
 - \*\*Inputs/Outputs\*\*:   
 - \*\*Inputs\*\*: Notification message, recipient (AdminID or CustomerID), and event context (e.g., product deletion, order status change).   
 - \*\*Outputs\*\*:   
 - A new Notification record stored in the database.   
 - The message delivered to the recipient via the configured channel (e.g., email or push notification).   
  
- \*\*Payment Processing Communication Interface\*\*:   
 - \*\*Description\*\*: The system communicates with a payment plugin to process and verify customer payments.   
 - \*\*Interaction Method\*\*: The system passes the payment information to the plugin, which processes the transaction and returns a confirmation or error message.   
 - \*\*Inputs/Outputs\*\*:   
 - \*\*Inputs\*\*: OrderID and payment information (e.g., card details, payment method).   
 - \*\*Outputs\*\*:   
 - A new Payment record stored in the database.   
 - Order status updated to reflect the payment result.   
  
- \*\*Email or Messaging Interface (for Notification Delivery)\*\*:   
 - \*\*Description\*\*: The system may send notifications via email or another messaging service to inform users of events such as order status changes, product updates, or payment confirmations.   
 - \*\*Interaction Method\*\*: The plugin uses an external API (e.g., SMTP, REST-based email service) to send messages.   
 - \*\*Inputs/Outputs\*\*:   
 - \*\*Inputs\*\*: Recipient email address, message content, and timestamp.   
 - \*\*Outputs\*\*: Email or message sent to the recipient.   
  
- \*\*Web Browsing Interface (for System Access)\*\*:   
 - \*\*Description\*\*: Both Administrators and Customers access the system through a web browser.   
 - \*\*Interaction Method\*\*: HTTP/HTTPS-based web access with secure login and session management.   
 - \*\*Inputs/Outputs\*\*:   
 - \*\*Inputs\*\*: User credentials (username and password), navigation actions, and input forms.   
 - \*\*Outputs\*\*: Web pages and dynamic content rendered in the browser, including product listings, order details, and notification displays.   
  
## Summary of External Interfaces   
  
| Interface Type | Interface Name | Description |  
|-----------------------------|----------------------------------------|-----------------------------------------------------------------------------|  
| \*\*User Interface\*\* | Administrator Interface | GUI for managing products, customers, orders, and plugins. |  
| \*\*User Interface\*\* | Customer Interface | GUI for viewing products, placing orders, and checking history. |  
| \*\*Software Interface\*\* | Database Interface | Centralized database for storing all system records. |  
| \*\*Software Interface\*\* | Plugin Interface | Integration with plugins for additional logic, validations, and processing. |  
| \*\*Communication Interface\*\* | Notification Communication Interface | Sends notifications to users via email or in-app messages. |  
| \*\*Communication Interface\*\* | Payment Processing Communication Interface | Processes payments via a payment plugin using external APIs. |  
| \*\*Communication Interface\*\* | Web Browsing Interface | Allows access to the system through HTTP/HTTPS protocols. |  
  
This section ensures that all external data sources and interfaces referenced in the functional requirements are clearly defined, and their interactions with the system are well documented for implementation purposes.

# Use Case

Use Case Name: Create Product   
Use Case ID: UC-01   
Actors: Administrator, Plugin   
Preconditions:   
- The Administrator is authenticated and logged into the system.   
- The Plugin is active and properly configured for product creation.   
  
Postconditions:   
- A new Product is successfully added to the system.   
- The Product details are stored in the database.   
- A Notification is sent to the Administrator confirming the product creation.   
  
Main Flow:   
1. The Administrator selects the "Create Product" option from the system menu.   
2. The system displays a form for entering Product details.   
3. The Administrator fills in the form with the necessary Product information (e.g., name, price, description).   
4. The Administrator submits the form.   
5. The system validates the input data.   
6. The system uses the Plugin to process any additional configurations or validations.   
7. The system creates the Product and saves it to the database.   
8. A Notification is generated and sent to the Administrator.   
  
Alternative Flow:   
1. If the input data is invalid, the system displays an error message and prompts the Administrator to correct the data.   
2. If the Plugin is not active or fails during processing, the system displays an error message and does not create the Product.  
  
Use Case Name: Update Product   
Use Case ID: UC-02   
Actors: Administrator, Plugin   
Preconditions:   
- The Administrator is authenticated and logged into the system.   
- The Plugin is active and properly configured for product updates.   
- A Product already exists in the system and is selected for modification.   
  
Postconditions:   
- The selected Product is successfully updated in the system.   
- The updated Product details are stored in the database.   
- A Notification is sent to the Administrator confirming the product update.   
  
Main Flow:   
1. The Administrator selects the "Update Product" option from the system menu.   
2. The system displays the details of the selected Product in an editable form.   
3. The Administrator modifies the necessary Product information (e.g., name, price, description).   
4. The Administrator submits the updated form.   
5. The system validates the modified input data.   
6. The system uses the Plugin to process any additional configurations or validations.   
7. The system updates the Product in the database.   
8. A Notification is generated and sent to the Administrator.   
  
Alternative Flow:   
1. If the input data is invalid, the system displays an error message and prompts the Administrator to correct the data.   
2. If the Plugin is not active or fails during processing, the system displays an error message and does not update the Product.  
  
Use Case Name: Delete Product   
Use Case ID: UC-03   
Actors: Administrator, Plugin   
Preconditions:   
- The Administrator is authenticated and logged into the system.   
- The Plugin is active and properly configured for product deletion.   
- A Product already exists in the system and is selected for deletion.   
  
Postconditions:   
- The selected Product is successfully removed from the system.   
- The deletion is recorded in the database.   
- A Notification is sent to the Administrator confirming the product deletion.   
  
Main Flow:   
1. The Administrator selects the "Delete Product" option from the system menu.   
2. The system displays the details of the selected Product and prompts for confirmation.   
3. The Administrator confirms the deletion.   
4. The system validates the request and checks for any dependencies (e.g., active Orders or Payments).   
5. The system uses the Plugin to process any additional checks or configurations.   
6. The system deletes the Product from the database.   
7. A Notification is generated and sent to the Administrator.   
  
Alternative Flow:   
1. If the Product is associated with an active Order or Payment, the system displays an error message and does not delete the Product.   
2. If the Plugin is not active or fails during processing, the system displays an error message and does not delete the Product.  
  
Use Case Name: View Product Details   
Use Case ID: UC-04   
Actors: Administrator, Customer   
Preconditions:   
- The Administrator or Customer is authenticated and logged into the system.   
- A Product already exists in the system and is selected for viewing.   
  
Postconditions:   
- The details of the selected Product are displayed to the user.   
- No changes are made to the Product or database.   
  
Main Flow:   
1. The Administrator or Customer selects a Product from the product list or search results.   
2. The system retrieves the Product details from the database.   
3. The system displays the Product information (e.g., name, price, description, availability).   
4. The user reviews the displayed Product details.   
  
Alternative Flow:   
1. If the selected Product does not exist or cannot be retrieved, the system displays an error message.   
2. If the Product is not available for viewing (e.g., due to access restrictions), the system displays a message indicating insufficient permissions.  
  
Use Case Name: Manage Product Inventory   
Use Case ID: UC-05   
Actors: Administrator, Plugin   
Preconditions:   
- The Administrator is authenticated and logged into the system.   
- The Plugin is active and properly configured for inventory management.   
- A Product already exists in the system.   
  
Postconditions:   
- The Product's inventory status is updated in the system.   
- The updated inventory data is stored in the database.   
- A Notification is sent to the Administrator confirming the inventory change.   
  
Main Flow:   
1. The Administrator selects the "Manage Product Inventory" option from the system menu.   
2. The system displays the inventory status of the selected Product.   
3. The Administrator adjusts the inventory quantity (e.g., adds or subtracts stock).   
4. The Administrator submits the changes.   
5. The system validates the new inventory quantity (e.g., no negative stock).   
6. The system uses the Plugin to handle any additional inventory logic (e.g., stock alerts).   
7. The system updates the Product's inventory in the database.   
8. A Notification is generated and sent to the Administrator.   
  
Alternative Flow:   
1. If the inventory quantity is invalid (e.g., negative or exceeds maximum limit), the system displays an error message and prompts the Administrator to correct the input.   
2. If the Plugin is not active or fails during processing, the system displays an error message and does not update the inventory.  
  
Use Case Name: Register Customer   
Use Case ID: UC-06   
Actors: Administrator   
Preconditions:   
- The Administrator is authenticated and logged into the system.   
- The system is accessible and functioning properly.   
  
Postconditions:   
- A new Customer is successfully added to the system.   
- The Customer details are stored in the database.   
- A Notification is sent to the Administrator confirming the customer registration.   
  
Main Flow:   
1. The Administrator selects the "Register Customer" option from the system menu.   
2. The system displays a form for entering Customer information (e.g., name, email, address).   
3. The Administrator fills in the form with the necessary details.   
4. The Administrator submits the form.   
5. The system validates the input data (e.g., checks for valid email format and unique email address).   
6. The system creates the Customer and saves the information to the database.   
7. A Notification is generated and sent to the Administrator.   
  
Alternative Flow:   
1. If the input data is invalid, the system displays an error message and prompts the Administrator to correct the data.   
2. If the email address is already registered, the system displays an error message and does not register the Customer.  
  
Use Case Name: View Customer Profile   
Use Case ID: UC-07   
Actors: Administrator   
Preconditions:   
- The Administrator is authenticated and logged into the system.   
- A Customer already exists in the system and is selected for viewing.   
  
Postconditions:   
- The details of the selected Customer are displayed to the Administrator.   
- No changes are made to the Customer or database.   
  
Main Flow:   
1. The Administrator selects the "View Customer Profile" option from the system menu.   
2. The system displays the Customer details (e.g., name, email, address, order history).   
3. The Administrator reviews the displayed Customer information.   
  
Alternative Flow:   
1. If the selected Customer does not exist or cannot be retrieved, the system displays an error message.   
2. If the Administrator does not have permission to view the Customer profile, the system displays a message indicating insufficient permissions.  
  
Use Case Name: Update Customer Information   
Use Case ID: UC-08   
Actors: Administrator   
Preconditions:   
- The Administrator is authenticated and logged into the system.   
- A Customer already exists in the system and is selected for modification.   
  
Postconditions:   
- The selected Customer's information is successfully updated in the system.   
- The updated Customer details are stored in the database.   
- A Notification is sent to the Administrator confirming the customer update.   
  
Main Flow:   
1. The Administrator selects the "Update Customer Information" option from the system menu.   
2. The system displays the details of the selected Customer in an editable form.   
3. The Administrator modifies the necessary Customer information (e.g., name, email, address).   
4. The Administrator submits the updated form.   
5. The system validates the modified input data (e.g., checks for valid email format and uniqueness).   
6. The system updates the Customer's information in the database.   
7. A Notification is generated and sent to the Administrator.   
  
Alternative Flow:   
1. If the input data is invalid (e.g., incorrect format, duplicate email), the system displays an error message and prompts the Administrator to correct the data.   
2. If the Administrator does not have permission to modify the Customer information, the system displays a message indicating insufficient permissions.  
  
Use Case Name: Delete Customer Account   
Use Case ID: UC-09   
Actors: Administrator   
Preconditions:   
- The Administrator is authenticated and logged into the system.   
- A Customer account already exists in the system and is selected for deletion.   
  
Postconditions:   
- The selected Customer account is successfully removed from the system.   
- All associated data (e.g., Orders, Payments) are archived or deleted as per system policy.   
- The deletion is recorded in the database.   
- A Notification is sent to the Administrator confirming the customer account deletion.   
  
Main Flow:   
1. The Administrator selects the "Delete Customer Account" option from the system menu.   
2. The system displays the details of the selected Customer and prompts for confirmation.   
3. The Administrator confirms the deletion.   
4. The system validates the request and checks for any system restrictions (e.g., active Orders).   
5. The system removes the Customer account and associated data from the database.   
6. A Notification is generated and sent to the Administrator.   
  
Alternative Flow:   
1. If the Customer has active Orders or Payments, the system displays an error message and does not delete the account.   
2. If the Administrator does not have permission to delete the Customer account, the system displays a message indicating insufficient permissions.  
  
Use Case Name: Place Order   
Use Case ID: UC-10   
Actors: Customer, Administrator, Plugin   
Preconditions:   
- The Customer is authenticated and logged into the system.   
- At least one Product is available in the system.   
- The Plugin is active and properly configured for order placement.   
  
Postconditions:   
- A new Order is successfully created in the system.   
- The Order details are stored in the database.   
- A Notification is sent to the Customer and Administrator confirming the order.   
- Inventory is adjusted if applicable.   
  
Main Flow:   
1. The Customer selects Products to add to their cart.   
2. The Customer reviews the cart contents and proceeds to checkout.   
3. The system prompts the Customer to provide shipping and payment information.   
4. The Customer enters the required information and confirms the order.   
5. The system validates the order details (e.g., product availability, payment information).   
6. The system uses the Plugin to process payment and any additional order validations.   
7. The system creates the Order and saves it to the database.   
8. Inventory is updated if necessary (e.g., stock reduced).   
9. A Notification is generated and sent to the Customer and Administrator.   
  
Alternative Flow:   
1. If the selected Products are out of stock, the system displays an error message and prevents order submission.   
2. If the payment information is invalid or the Plugin fails during payment processing, the system displays an error message and does not place the Order.   
3. If the Customer does not have permission to place an order (e.g., account is suspended), the system displays a message indicating insufficient permissions.  
  
Use Case Name: View Order Details   
Use Case ID: UC-11   
Actors: Administrator, Customer   
Preconditions:   
- The Administrator or Customer is authenticated and logged into the system.   
- An Order already exists in the system and is selected for viewing.   
  
Postconditions:   
- The details of the selected Order are displayed to the user.   
- No changes are made to the Order or database.   
  
Main Flow:   
1. The Administrator or Customer selects the "View Order Details" option from the system menu or by clicking on an Order in the list.   
2. The system retrieves the selected Order details from the database, including Products, Payment status, and shipping information.   
3. The system displays the Order information (e.g., order number, date, customer name, total amount, status).   
4. The user reviews the displayed Order details.   
  
Alternative Flow:   
1. If the selected Order does not exist or cannot be retrieved, the system displays an error message.   
2. If the user does not have permission to view the Order (e.g., Customer views another Customer's Order), the system displays a message indicating insufficient permissions.  
  
Use Case Name: Update Order Status   
Use Case ID: UC-12   
Actors: Administrator, Plugin   
Preconditions:   
- The Administrator is authenticated and logged into the system.   
- The Plugin is active and properly configured for order status updates.   
- An Order already exists in the system and is selected for modification.   
  
Postconditions:   
- The selected Order's status is successfully updated in the system.   
- The updated Order status is stored in the database.   
- A Notification is sent to the Administrator and Customer confirming the status change.   
  
Main Flow:   
1. The Administrator selects the "Update Order Status" option from the system menu or by clicking on an Order.   
2. The system displays the current Order status and a dropdown list of available statuses.   
3. The Administrator selects a new status from the dropdown list.   
4. The Administrator submits the update request.   
5. The system validates the selected status and checks for any system restrictions.   
6. The system uses the Plugin to process any additional status-specific logic (e.g., shipping integration).   
7. The system updates the Order status in the database.   
8. A Notification is generated and sent to the Administrator and Customer.   
  
Alternative Flow:   
1. If the selected status is invalid or not applicable to the current Order, the system displays an error message and prompts the Administrator to select a valid status.   
2. If the Plugin is not active or fails during processing, the system displays an error message and does not update the Order status.  
  
Use Case Name: Cancel Order   
Use Case ID: UC-13   
Actors: Administrator, Customer, Plugin   
Preconditions:   
- The Administrator or Customer is authenticated and logged into the system.   
- An Order already exists in the system and is selected for cancellation.   
- The Plugin is active and properly configured for order cancellation.   
  
Postconditions:   
- The selected Order is successfully cancelled in the system.   
- The cancellation is recorded in the database.   
- Inventory is adjusted if applicable (e.g., stock increased).   
- A Notification is sent to the Administrator and Customer confirming the cancellation.   
  
Main Flow:   
1. The Administrator or Customer selects the "Cancel Order" option from the system menu or by clicking on an Order in the list.   
2. The system displays the Order details and prompts for confirmation of cancellation.   
3. The user confirms the cancellation.   
4. The system validates the request and checks if the Order is eligible for cancellation (e.g., not already shipped).   
5. The system uses the Plugin to process any additional checks or configurations (e.g., refund policies).   
6. The system cancels the Order and updates the status in the database.   
7. Inventory is adjusted if necessary (e.g., restock Products).   
8. A Notification is generated and sent to the Administrator and Customer.   
  
Alternative Flow:   
1. If the Order is not eligible for cancellation (e.g., already shipped or processed), the system displays an error message and does not cancel the Order.   
2. If the Plugin is not active or fails during processing, the system displays an error message and does not proceed with the cancellation.   
3. If the user does not have permission to cancel the Order (e.g., Customer cancels another Customer's Order), the system displays a message indicating insufficient permissions.  
  
Use Case Name: Process Payment   
Use Case ID: UC-14   
Actors: Customer, Administrator, Plugin   
Preconditions:   
- The Customer or Administrator is authenticated and logged into the system.   
- An Order exists and is selected for payment processing.   
- The Plugin is active and properly configured for payment handling.   
  
Postconditions:   
- The Payment is successfully processed and recorded in the system.   
- The Order status is updated to reflect the completed Payment.   
- A Notification is sent to the Customer and Administrator confirming the payment.   
  
Main Flow:   
1. The Customer or Administrator selects the "Process Payment" option for a specific Order.   
2. The system displays the Order details and prompts the user to enter payment information (e.g., card details, payment method).   
3. The user enters the required payment information and confirms the transaction.   
4. The system validates the payment data (e.g., correct format, valid payment method).   
5. The system uses the Plugin to initiate and verify the payment transaction.   
6. The Plugin returns a success status after processing the payment.   
7. The system updates the Payment and Order records in the database.   
8. A Notification is generated and sent to the Customer and Administrator confirming the payment success.   
  
Alternative Flow:   
1. If the payment information is invalid, the system displays an error message and prompts the user to correct the data.   
2. If the Plugin is not active or fails during payment processing (e.g., network issues, payment gateway error), the system displays an error message and does not record the Payment.   
3. If the Order is already paid or invalid, the system displays an error message and does not proceed with the payment.  
  
Use Case Name: View Payment History   
Use Case ID: UC-15   
Actors: Administrator, Customer   
Preconditions:   
- The Administrator or Customer is authenticated and logged into the system.   
- A Payment record exists in the system and is associated with the user.   
- The system is accessible and functioning properly.   
  
Postconditions:   
- The Payment history of the user is displayed.   
- No changes are made to the Payment records or database.   
  
Main Flow:   
1. The Administrator or Customer selects the "View Payment History" option from the system menu.   
2. The system retrieves all Payment records associated with the user from the database.   
3. The system displays the Payment history in a list or table format, including details such as payment amount, date, status, and associated Order.   
4. The user reviews the displayed Payment history.   
  
Alternative Flow:   
1. If no Payment records are found for the user, the system displays a message indicating that there is no payment history available.   
2. If the user does not have permission to view their own or another user's Payment history, the system displays a message indicating insufficient permissions.   
3. If there is an error retrieving Payment records, the system displays an error message and suggests trying again later.  
  
Use Case Name: Manage Administrator Account   
Use Case ID: UC-16   
Actors: Administrator, Plugin   
Preconditions:   
- The Administrator is authenticated and logged into the system.   
- The Plugin is active and properly configured for administrator account management.   
- An Administrator account already exists in the system and is selected for modification.   
  
Postconditions:   
- The selected Administrator account is successfully updated in the system.   
- The updated account details are stored in the database.   
- A Notification is sent to the Administrator confirming the account update.   
  
Main Flow:   
1. The Administrator selects the "Manage Administrator Account" option from the system menu.   
2. The system displays the Administrator's account details in an editable form.   
3. The Administrator modifies the necessary information (e.g., password, email, role).   
4. The Administrator submits the updated form.   
5. The system validates the modified input data (e.g., password strength, valid email format).   
6. The system uses the Plugin to process any additional configurations or validations.   
7. The system updates the Administrator's account in the database.   
8. A Notification is generated and sent to the Administrator.   
  
Alternative Flow:   
1. If the input data is invalid (e.g., weak password, duplicate email), the system displays an error message and prompts the Administrator to correct the data.   
2. If the Plugin is not active or fails during processing, the system displays an error message and does not update the Administrator account.   
3. If the Administrator does not have permission to modify the account (e.g., attempting to change another Administrator's role), the system displays a message indicating insufficient permissions.  
  
Use Case Name: Send Notification   
Use Case ID: UC-17   
Actors: Administrator, Plugin   
Preconditions:   
- The Administrator is authenticated and logged into the system.   
- The Plugin is active and properly configured for notification sending.   
- A Notification message is generated based on a system event (e.g., product creation, order status change).   
  
Postconditions:   
- The Notification is successfully sent to the intended recipient (e.g., Administrator, Customer).   
- The Notification is recorded in the system for tracking purposes.   
- The recipient receives the notification via the configured channel (e.g., email, in-app message).   
  
Main Flow:   
1. A system event occurs that requires a notification (e.g., a Product is created, an Order is updated).   
2. The system generates a Notification message based on the event and user roles.   
3. The system uses the Plugin to determine the appropriate channel and format for sending the Notification.   
4. The system sends the Notification to the intended recipient.   
5. The system records the Notification in the database for reference.   
  
Alternative Flow:   
1. If the Plugin is not active or fails to send the Notification, the system displays an error message and logs the failure for review.   
2. If the recipient cannot be identified or is invalid, the Notification is not sent, and an error is recorded.  
  
Use Case Name: View Notification   
Use Case ID: UC-18   
Actors: Administrator, Customer   
Preconditions:   
- The Administrator or Customer is authenticated and logged into the system.   
- A Notification exists in the system and is associated with the user.   
- The system is accessible and functioning properly.   
  
Postconditions:   
- The details of the Notification are displayed to the user.   
- The Notification remains in the system for reference.   
- No changes are made to the Notification or database.   
  
Main Flow:   
1. The Administrator or Customer selects the "View Notification" option from the system menu or notification center.   
2. The system retrieves all Notification records associated with the user from the database.   
3. The system displays the Notification details (e.g., title, message, timestamp, related entity).   
4. The user reviews the displayed Notification.   
  
Alternative Flow:   
1. If no Notification records are found for the user, the system displays a message indicating that there are no notifications available.   
2. If the user does not have permission to view the Notification (e.g., viewing another user's private notification), the system displays a message indicating insufficient permissions.   
3. If there is an error retrieving Notification records, the system displays an error message and suggests trying again later.  
  
Use Case Name: Manage Plugin   
Use Case ID: UC-19   
Actors: Administrator   
Preconditions:   
- The Administrator is authenticated and logged into the system.   
- The Plugin exists in the system and is selected for management.   
  
Postconditions:   
- The Plugin is either activated, deactivated, or configured successfully.   
- The Plugin status or settings are updated in the database.   
- A Notification is sent to the Administrator confirming the plugin management action.   
  
Main Flow:   
1. The Administrator selects the "Manage Plugin" option from the system menu.   
2. The system displays a list of available Plugins, including their current status and configuration options.   
3. The Administrator selects a Plugin for management (e.g., activate, deactivate, configure).   
4. The system prompts the Administrator to input new configuration details if applicable.   
5. The Administrator submits the management request.   
6. The system validates the Plugin and the requested action.   
7. The system updates the Plugin's status or configuration in the database.   
8. A Notification is generated and sent to the Administrator confirming the action.   
  
Alternative Flow:   
1. If the Plugin is invalid or does not exist, the system displays an error message and does not proceed with the management action.   
2. If the configuration data is invalid, the system displays an error message and prompts the Administrator to correct the input.   
3. If the Administrator does not have permission to manage Plugins, the system displays a message indicating insufficient permissions.  
  
Use Case Name: Install Plugin   
Use Case ID: UC-20   
Actors: Administrator   
Preconditions:   
- The Administrator is authenticated and logged into the system.   
- The Plugin is available for installation and not already installed.   
- The system is accessible and functioning properly.   
  
Postconditions:   
- The Plugin is successfully installed and activated in the system.   
- The Plugin configuration is stored in the database.   
- A Notification is sent to the Administrator confirming the plugin installation.   
  
Main Flow:   
1. The Administrator selects the "Install Plugin" option from the system menu.   
2. The system displays a list of available Plugins for installation.   
3. The Administrator selects a Plugin from the list.   
4. The system validates the Plugin's compatibility and requirements.   
5. The system installs the Plugin and activates it.   
6. The system saves the Plugin configuration in the database.   
7. A Notification is generated and sent to the Administrator confirming the installation.   
  
Alternative Flow:   
1. If the Plugin is already installed, the system displays a message indicating the Plugin is already active.   
2. If the Plugin is incompatible with the system, the system displays an error message and does not proceed with the installation.   
3. If the installation fails due to technical issues, the system displays an error message and logs the failure for review.   
4. If the Administrator does not have permission to install Plugins, the system displays a message indicating insufficient permissions.  
  
Use Case Name: Uninstall Plugin   
Use Case ID: UC-21   
Actors: Administrator   
Preconditions:   
- The Administrator is authenticated and logged into the system.   
- The Plugin is already installed in the system.   
- The system is accessible and functioning properly.   
  
Postconditions:   
- The selected Plugin is successfully uninstalled from the system.   
- The Plugin's configuration is removed from the database.   
- A Notification is sent to the Administrator confirming the plugin uninstallation.   
  
Main Flow:   
1. The Administrator selects the "Uninstall Plugin" option from the system menu.   
2. The system displays a list of installed Plugins.   
3. The Administrator selects a Plugin for uninstallation.   
4. The system prompts the Administrator for confirmation.   
5. The Administrator confirms the uninstallation.   
6. The system validates the Plugin and checks for dependencies.   
7. The system uninstalls the Plugin and removes its configuration from the database.   
8. A Notification is generated and sent to the Administrator.   
  
Alternative Flow:   
1. If the Plugin is in use (e.g., required for active functionalities), the system displays an error message and does not proceed with the uninstallation.   
2. If the uninstallation fails due to technical issues, the system displays an error message and logs the failure for review.   
3. If the Administrator does not have permission to uninstall Plugins, the system displays a message indicating insufficient permissions.  
  
---  
  
Use Case Name: Configure Plugin Settings   
Use Case ID: UC-22   
Actors: Administrator   
Preconditions:   
- The Administrator is authenticated and logged into the system.   
- The Plugin is already installed and active.   
- The system is accessible and functioning properly.   
  
Postconditions:   
- The Plugin's settings are successfully updated in the system.   
- The updated configuration is stored in the database.   
- A Notification is sent to the Administrator confirming the configuration change.   
  
Main Flow:   
1. The Administrator selects the "Configure Plugin Settings" option from the system menu.   
2. The system displays the configuration options for the selected Plugin.   
3. The Administrator modifies the necessary settings (e.g., API keys, behavior parameters).   
4. The Administrator submits the configuration changes.   
5. The system validates the new configuration data.   
6. The system updates the Plugin's settings in the database.   
7. A Notification is generated and sent to the Administrator confirming the changes.   
  
Alternative Flow:   
1. If the configuration data is invalid, the system displays an error message and prompts the Administrator to correct the input.   
2. If the Plugin is not active or fails during configuration processing, the system displays an error message and does not apply the changes.   
3. If the Administrator does not have permission to configure Plugins, the system displays a message indicating insufficient permissions.  
  
Use Case Name: Manage OrderItem   
Use Case ID: UC-23   
Actors: Administrator   
Preconditions:   
- The Administrator is authenticated and logged into the system.   
- An Order exists in the system and is selected for modification.   
- An OrderItem is associated with the selected Order and is selected for management.   
  
Postconditions:   
- The selected OrderItem is successfully modified (e.g., quantity updated, removed from the Order).   
- The changes to the OrderItem are stored in the database.   
- The Order status is updated if necessary (e.g., total price recalculated).   
- A Notification is sent to the Administrator and Customer confirming the OrderItem change.   
  
Main Flow:   
1. The Administrator selects the "View Order Details" option to access the Order containing the OrderItem.   
2. The system displays the Order details, including a list of OrderItems.   
3. The Administrator selects a specific OrderItem for modification (e.g., change quantity, remove item).   
4. The system prompts the Administrator to input the new details for the OrderItem.   
5. The Administrator submits the changes.   
6. The system validates the input data (e.g., quantity must be positive and within available stock).   
7. The system updates the OrderItem in the database and recalculates the Order total if necessary.   
8. A Notification is generated and sent to the Administrator and Customer confirming the OrderItem modification.   
  
Alternative Flow:   
1. If the input data is invalid (e.g., negative quantity, exceeds available stock), the system displays an error message and prompts the Administrator to correct the data.   
2. If the Administrator does not have permission to modify the OrderItem (e.g., attempting to change an Order not assigned to them), the system displays a message indicating insufficient permissions.   
3. If the selected OrderItem does not exist or cannot be retrieved, the system displays an error message.  
  
Use Case Name: Manage InventoryRecord   
Use Case ID: UC-24   
Actors: Administrator, Plugin   
Preconditions:   
- The Administrator is authenticated and logged into the system.   
- The Plugin is active and properly configured for inventory record management.   
- An InventoryRecord exists in the system and is selected for modification.   
  
Postconditions:   
- The selected InventoryRecord is successfully updated or modified in the system.   
- The updated InventoryRecord details are stored in the database.   
- A Notification is sent to the Administrator confirming the inventory record change.   
  
Main Flow:   
1. The Administrator selects the "Manage InventoryRecord" option from the system menu.   
2. The system displays the InventoryRecord details for the selected Product or OrderItem.   
3. The Administrator modifies the necessary InventoryRecord information (e.g., quantity adjustment, reason for change, date of adjustment).   
4. The Administrator submits the updated InventoryRecord.   
5. The system validates the input data (e.g., correct format, valid quantity values, no negative inventory).   
6. The system uses the Plugin to process any additional logic (e.g., audit trails, inventory alerts).   
7. The system updates the InventoryRecord in the database.   
8. A Notification is generated and sent to the Administrator confirming the changes.   
  
Alternative Flow:   
1. If the input data is invalid (e.g., incorrect format, negative quantity, or exceeding limits), the system displays an error message and prompts the Administrator to correct the data.   
2. If the Plugin is not active or fails during processing, the system displays an error message and does not update the InventoryRecord.   
3. If the Administrator does not have permission to modify the InventoryRecord, the system displays a message indicating insufficient permissions.