# 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.