Functional Requirement

# 1. Functional Requirements  
  
## 1.1 Customer Registration Function   
\*\*Function ID:\*\* FR-01   
\*\*Description:\*\* Customers can register for a new account by providing valid personal information and a valid email address. The system validates the email, checks for duplicate entries, and sends a confirmation email.   
\*\*Input:\*\* Name, email, password, and optional contact information.   
\*\*Output:\*\* A new customer account is created and stored in the Personal Information and Contact Information data entities. A confirmation email is sent to the customer's email address.  
  
## 1.2 Customer Login Function   
\*\*Function ID:\*\* FR-02   
\*\*Description:\*\* Customers can log in to the system by entering their registered email and password. The system validates the credentials and logs the login event.   
\*\*Input:\*\* Email and password.   
\*\*Output:\*\* The customer is authenticated and redirected to their personalized dashboard. The login event is recorded in the Purchase History data entity.  
  
## 1.3 Customer Logout Function   
\*\*Function ID:\*\* FR-03   
\*\*Description:\*\* Customers can log out of the system. The session is terminated and the logout event is recorded.   
\*\*Input:\*\* The customer clicks the "Logout" button.   
\*\*Output:\*\* The session is terminated, and the customer is redirected to the homepage or login page. The logout event is recorded in the Purchase History data entity.  
  
## 1.4 Personal Information Update Function   
\*\*Function ID:\*\* FR-04   
\*\*Description:\*\* Customers can update their personal information, such as name or email. The system validates the updated information and logs the event.   
\*\*Input:\*\* Updated personal information fields.   
\*\*Output:\*\* Updated personal information is stored in the Personal Information data entity. A confirmation email is sent, and the update event is recorded in the Purchase History data entity.  
  
## 1.5 Contact Information Management Function   
\*\*Function ID:\*\* FR-05   
\*\*Description:\*\* Customers can manage their contact information, such as phone number or address. The system validates the updated information and logs the event.   
\*\*Input:\*\* Updated contact information fields.   
\*\*Output:\*\* Updated contact information is stored in the Contact Information data entity. The update event is recorded in the Purchase History data entity.  
  
## 1.6 Purchase History Viewing Function   
\*\*Function ID:\*\* FR-06   
\*\*Description:\*\* Customers can view their purchase history, including order details and status. The system retrieves the history and logs the event.   
\*\*Input:\*\* The customer navigates to the "Purchase History" section.   
\*\*Output:\*\* The purchase history is displayed on the interface, including order number, date, product names, quantities, total amount, and status. The viewing event is recorded in the Purchase History data entity.  
  
## 1.7 Product Addition Function   
\*\*Function ID:\*\* FR-07   
\*\*Description:\*\* Administrators can add new products to the system, specifying details such as name, description, price, category, and inventory.   
\*\*Input:\*\* Product name, description, price, category, and inventory quantity.   
\*\*Output:\*\* A new product is stored in the Product data entity, associated with the selected category. The inventory is updated accordingly. The addition event is recorded in the Purchase History data entity.  
  
## 1.8 Product Removal Function   
\*\*Function ID:\*\* FR-08   
\*\*Description:\*\* Administrators can remove existing products from the system. The system checks for active orders or cart references before proceeding.   
\*\*Input:\*\* The product to be removed is selected and the "Remove Product" button is clicked.   
\*\*Output:\*\* The product is removed from the Product data entity, and the inventory is updated. The removal event is recorded in the Purchase History data entity.  
  
## 1.9 Product Information Update Function   
\*\*Function ID:\*\* FR-09   
\*\*Description:\*\* Administrators can update the information of existing products, such as name, description, price, category, or inventory.   
\*\*Input:\*\* Updated product information fields.   
\*\*Output:\*\* Updated product information is stored in the Product data entity. The inventory and category are updated accordingly. The update event is recorded in the Purchase History data entity.  
  
## 1.10 Product Categorization Function   
\*\*Function ID:\*\* FR-10   
\*\*Description:\*\* Administrators can assign or modify the category of a product. The system validates the category and updates the association.   
\*\*Input:\*\* Product and new category selection.   
\*\*Output:\*\* The product is assigned to the selected category in the Category data entity. The update is logged in the Purchase History data entity.  
  
## 1.11 Inventory Management Function   
\*\*Function ID:\*\* FR-11   
\*\*Description:\*\* Administrators can view and modify inventory levels for products. The system updates the inventory quantity and logs the event.   
\*\*Input:\*\* Updated inventory quantity for a product.   
\*\*Output:\*\* Updated inventory levels are stored in the Inventory data entity. The management event is recorded in the Purchase History data entity.  
  
## 1.12 Shopping Cart Creation Function   
\*\*Function ID:\*\* FR-12   
\*\*Description:\*\* Customers can add products to their shopping cart. The system checks product availability and updates inventory.   
\*\*Input:\*\* Product selection and "Add to Cart" action.   
\*\*Output:\*\* The product is added to the Shopping Cart data entity. Inventory quantity is updated. The customer is shown a confirmation message.  
  
## 1.13 Shopping Cart Modification Function   
\*\*Function ID:\*\* FR-13   
\*\*Description:\*\* Customers can modify the quantity or remove items from their shopping cart. The system updates the cart and inventory.   
\*\*Input:\*\* Cart item selection, quantity change, or "Remove" action.   
\*\*Output:\*\* The cart is updated in the Shopping Cart and CartItem data entities. Inventory levels are adjusted accordingly. The modification event is recorded in the Purchase History data entity.  
  
## 1.14 Shopping Cart Viewing Function   
\*\*Function ID:\*\* FR-14   
\*\*Description:\*\* Customers can view the contents of their shopping cart. The system displays a summary of the cart items and their total cost.   
\*\*Input:\*\* The customer navigates to the "Shopping Cart" section.   
\*\*Output:\*\* The system displays the list of cart items, including product name, quantity, price, and total amount. The viewing event is recorded in the Purchase History data entity.  
  
## 1.15 Checkout Function   
\*\*Function ID:\*\* FR-15   
\*\*Description:\*\* Customers can proceed to checkout, select a shipping address, and make a payment. The system processes the payment and creates an order.   
\*\*Input:\*\* Cart items, shipping address, and payment method.   
\*\*Output:\*\* An order is created in the Order data entity. Inventory levels are updated. A payment is processed and stored in the Payment data entity. An order confirmation email is sent. The checkout event is recorded in the Purchase History data entity.  
  
## 1.16 Order Confirmation Email Sending Function   
\*\*Function ID:\*\* FR-16   
\*\*Description:\*\* The system automatically sends an order confirmation email to the customer after successful checkout. The email includes order details and a confirmation message.   
\*\*Input:\*\* Order details from the Order data entity.   
\*\*Output:\*\* An Order Confirmation Email is stored in the EmailRecord data entity and sent to the customer's email address. The confirmation event is recorded in the Purchase History data entity.  
  
## 1.17 Payment Processing Function   
\*\*Function ID:\*\* FR-17   
\*\*Description:\*\* The system processes the customer's selected payment method and validates the payment details. A successful payment is stored and associated with the order.   
\*\*Input:\*\* Payment method and payment details.   
\*\*Output:\*\* The payment is stored in the Payment data entity. The order is created and stored in the Order data entity. Inventory levels are updated. The payment event is recorded in the Purchase History data entity.  
  
## 1.18 Plugin Development Function   
\*\*Function ID:\*\* FR-18   
\*\*Description:\*\* Administrators can develop or install new plugins to expand the system's functionality. The system validates the plugin and activates it.   
\*\*Input:\*\* Plugin details or file.   
\*\*Output:\*\* The plugin is stored in the Plugin data entity and activated. The system interface is updated to reflect the new functionality. The development or installation event is recorded in the Purchase History data entity.  
  
## 1.19 Plugin Configuration Function   
\*\*Function ID:\*\* FR-19   
\*\*Description:\*\* Administrators can configure the settings of an installed plugin. The system validates and applies the configuration.   
\*\*Input:\*\* Plugin configuration settings.   
\*\*Output:\*\* The Plugin data entity is updated with the new configuration. The system interface reflects the updated settings. The configuration event is recorded in the Purchase History data entity.  
  
## 1.20 User Interface Specification Compliance Checking Function   
\*\*Function ID:\*\* FR-20   
\*\*Description:\*\* Administrators can check the system's interface for compliance with the defined User Interface Specification. The system identifies and reports non-compliant elements.   
\*\*Input:\*\* The administrator initiates a compliance check.   
\*\*Output:\*\* A compliance report is generated and displayed to the administrator. Non-compliant elements are highlighted. The compliance check event is recorded in the Purchase History data entity.  
  
## 1.21 Manage CartItem Function   
\*\*Function ID:\*\* FR-21   
\*\*Description:\*\* Customers can modify the quantity or remove items in their shopping cart. The system updates the cart and inventory accordingly.   
\*\*Input:\*\* Cart item selection, quantity change, or "Remove" action.   
\*\*Output:\*\* The cart is updated in the CartItem data entity. Inventory levels are adjusted accordingly. The modification event is recorded in the Purchase History data entity.  
  
## 1.22 Manage OrderItem Function   
\*\*Function ID:\*\* FR-22   
\*\*Description:\*\* Administrators can edit, remove, or update the status of items in an order. The system updates the order and inventory accordingly.   
\*\*Input:\*\* Order item selection, quantity change, or "Remove" action.   
\*\*Output:\*\* The order item is updated in the OrderItem data entity. Inventory levels are adjusted accordingly. The modification event is recorded in the Purchase History data entity.  
  
## 1.23 Manage EmailRecord Function   
\*\*Function ID:\*\* FR-23   
\*\*Description:\*\* Administrators can view, edit, or delete email records. The system updates the EmailRecord data entity and logs the event.   
\*\*Input:\*\* Email record selection and action (view, edit, or delete).   
\*\*Output:\*\* The EmailRecord data entity is updated. The system displays a confirmation message. The management event is recorded in the Purchase History data entity.  
  
## 1.24 View EmailRecord Function   
\*\*Function ID:\*\* FR-24   
\*\*Description:\*\* Customers and administrators can view the details of an email record. The system displays the email content and logs the viewing event.   
\*\*Input:\*\* Email record selection.   
\*\*Output:\*\* The system displays the email details from the EmailRecord data entity. The viewing event is recorded in the Purchase History data entity.  
  
## 1.25 View CartItem Function   
\*\*Function ID:\*\* FR-25   
\*\*Description:\*\* Customers can view the details of a specific cart item. The system displays the product description, price, and quantity.   
\*\*Input:\*\* Cart item selection.   
\*\*Output:\*\* The system displays the cart item details from the CartItem data entity. The viewing event is recorded in the Purchase History data entity.  
  
## 1.26 View Payment Function   
\*\*Function ID:\*\* FR-26   
\*\*Description:\*\* Customers and administrators can view the details of a payment record. The system displays the payment method, amount, date, and status.   
\*\*Input:\*\* Payment record selection.   
\*\*Output:\*\* The system displays the payment details from the Payment data entity. The viewing event is recorded in the Purchase History data entity.  
  
## 1.27 Manage Administrator Function   
\*\*Function ID:\*\* FR-27   
\*\*Description:\*\* Administrators can manage other administrator accounts by updating or deactivating them. The system updates the administrator's information and logs the event.   
\*\*Input:\*\* Administrator account selection and action (update or deactivate).   
\*\*Output:\*\* The administrator account is updated in the Personal Information and Contact Information data entities. The management event is recorded in the Purchase History data entity.  
  
## 1.28 Manage Customer Function   
\*\*Function ID:\*\* FR-28   
\*\*Description:\*\* Administrators can manage customer accounts, including updating personal information or deactivating accounts.   
\*\*Input:\*\* Customer account selection and action (update or deactivate).   
\*\*Output:\*\* The customer's information is updated in the Personal Information and Contact Information data entities. The management event is recorded in the Purchase History data entity.  
  
## 1.29 Manage Category Function   
\*\*Function ID:\*\* FR-29   
\*\*Description:\*\* Administrators can add, update, or remove product categories. The system validates the category and updates the associations with products.   
\*\*Input:\*\* Category selection and action (add, update, or delete).   
\*\*Output:\*\* The Category data entity is updated. Product associations are modified accordingly. The management event is recorded in the Purchase History data entity.  
  
## 1.30 Manage User Interface Specification Function   
\*\*Function ID:\*\* FR-30   
\*\*Description:\*\* Administrators can modify the User Interface Specification to align with new design or functionality requirements. The system applies the changes and logs the event.   
\*\*Input:\*\* Updated UI specification details.   
\*\*Output:\*\* The User Interface Specification data entity is updated. The system interface is modified to reflect the changes. The management event is recorded in the Purchase History data entity.  
  
## 1.31 Manage Personal Information Function   
\*\*Function ID:\*\* FR-31   
\*\*Description:\*\* Customers or administrators can view or update a customer's personal information. The system validates the changes and logs the event.   
\*\*Input:\*\* Personal information selection and action (view or update).   
\*\*Output:\*\* The Personal Information data entity is updated if changes are made. The viewing or update event is recorded in the Purchase History data entity.  
  
## 1.32 Manage Contact Information Function   
\*\*Function ID:\*\* FR-32   
\*\*Description:\*\* Customers or administrators can view or update a customer's contact information. The system validates the changes and logs the event.   
\*\*Input:\*\* Contact information selection and action (view or update).   
\*\*Output:\*\* The Contact Information data entity is updated if changes are made. The viewing or update event is recorded in the Purchase History data entity.  
  
## 1.33 Order Cancellation Function   
\*\*Function ID:\*\* FR-33   
\*\*Description:\*\* Customers or administrators can cancel an order if it is in an eligible status. The system updates the order and inventory.   
\*\*Input:\*\* Order selection and "Cancel Order" action.   
\*\*Output:\*\* The order is marked as "Cancelled" in the Order data entity. Inventory levels are updated. The cancellation event is recorded in the Purchase History data entity.

External Description

# 2. External Interfaces  
  
## 2.1 User Interface  
  
The system interacts with users through a web-based graphical user interface (GUI) and mobile application interface (if applicable). These interfaces are designed to be intuitive and user-friendly, allowing customers and administrators to perform actions such as registration, login, managing personal and contact information, viewing and modifying shopping carts, and managing product and order data.  
  
### 2.1.1 Personal Information and Contact Information Interface  
- \*\*Description:\*\* This interface allows users to input, view, and update their personal and contact information.  
- \*\*Inputs:\*\* Name, email, password, phone number, address.  
- \*\*Outputs:\*\* Display of current information, confirmation messages, and error messages.  
- \*\*Interaction Method:\*\* Form-based input and output with validation feedback provided in real-time.  
  
### 2.1.2 Shopping Cart Interface  
- \*\*Description:\*\* This interface allows customers to view and modify the contents of their shopping cart.  
- \*\*Inputs:\*\* Product selection, quantity adjustments, remove actions.  
- \*\*Outputs:\*\* Display of cart items, total cost, and confirmation messages.  
- \*\*Interaction Method:\*\* Interactive form with real-time updates of inventory and cart status.  
  
### 2.1.3 Purchase History Interface  
- \*\*Description:\*\* This interface provides a detailed view of a customer's purchase history and order status.  
- \*\*Inputs:\*\* Navigation to the "Purchase History" section.  
- \*\*Outputs:\*\* List of orders with order number, date, product names, quantities, total amount, and status.  
- \*\*Interaction Method:\*\* Tabular or list-based display with filtering and sorting capabilities.  
  
### 2.1.4 Product Management Interface  
- \*\*Description:\*\* This interface is used by administrators to manage product data, including adding, updating, or removing products.  
- \*\*Inputs:\*\* Product name, description, price, category, inventory quantity.  
- \*\*Outputs:\*\* Display of product lists, confirmation messages, and error messages.  
- \*\*Interaction Method:\*\* Form-based input and output with interactive product lists and filters.  
  
### 2.1.5 Order Management Interface  
- \*\*Description:\*\* This interface allows administrators to manage order items and statuses.  
- \*\*Inputs:\*\* Order item selection, quantity change, or "Remove" action.  
- \*\*Outputs:\*\* Updated order status and confirmation messages.  
- \*\*Interaction Method:\*\* Interactive form with real-time updates to order and inventory data.  
  
### 2.1.6 Email Record Interface  
- \*\*Description:\*\* This interface allows both customers and administrators to view email records.  
- \*\*Inputs:\*\* Email record selection.  
- \*\*Outputs:\*\* Display of email content, including subject, body, date, and recipient.  
- \*\*Interaction Method:\*\* List-based display with clickable entries to view detailed email records.  
  
### 2.1.7 Payment Interface  
- \*\*Description:\*\* This interface allows users to view payment records.  
- \*\*Inputs:\*\* Payment record selection.  
- \*\*Outputs:\*\* Display of payment method, amount, date, and status.  
- \*\*Interaction Method:\*\* List-based or tabular display with filtering capabilities for easy navigation.  
  
### 2.1.8 Plugin Management Interface  
- \*\*Description:\*\* This interface allows administrators to install, configure, and manage plugins.  
- \*\*Inputs:\*\* Plugin details or file, configuration settings.  
- \*\*Outputs:\*\* Display of plugin status, configuration options, and system interface updates.  
- \*\*Interaction Method:\*\* Form-based input for plugin installation and configuration, with real-time feedback on success or failure.  
  
### 2.1.9 Administrator Management Interface  
- \*\*Description:\*\* This interface allows administrators to manage other administrator accounts.  
- \*\*Inputs:\*\* Administrator account selection and action (update or deactivate).  
- \*\*Outputs:\*\* Updated administrator information and confirmation messages.  
- \*\*Interaction Method:\*\* Interactive form with real-time validation and logging of management events.  
  
### 2.1.10 User Interface Specification Compliance Interface  
- \*\*Description:\*\* This interface allows administrators to check the system's compliance with defined UI specifications.  
- \*\*Inputs:\*\* Compliance check initiation.  
- \*\*Outputs:\*\* Compliance report with highlighted non-compliant elements.  
- \*\*Interaction Method:\*\* Button or menu-based initiation, followed by a detailed report display.  
  
## 2.2 Hardware Interface  
  
The system does not require direct interaction with external hardware devices for its core functionalities. However, the following hardware interfaces may be relevant for extended operations:  
  
### 2.2.1 Point of Sale (POS) Devices (if applicable)  
- \*\*Description:\*\* If the system is extended to support in-store purchases, it may interface with POS devices.  
- \*\*Inputs:\*\* Product scanning, payment processing, and inventory updates from the POS.  
- \*\*Outputs:\*\* Transaction confirmation, inventory updates, and order creation.  
- \*\*Interaction Method:\*\* Communication via standard POS APIs or serial interfaces.  
  
## 2.3 Software Interface  
  
The system interacts with various software components and data entities to manage its operations. These include internal data storage and external tools.  
  
### 2.3.1 Personal Information Data Entity  
- \*\*Description:\*\* Stores customer personal information such as name, email, and password.  
- \*\*Inputs:\*\* New or updated personal information from the user interface.  
- \*\*Outputs:\*\* Updated personal information for display and use in other functions.  
- \*\*Interaction Method:\*\* Direct database reads and writes with validation and logging.  
  
### 2.3.2 Contact Information Data Entity  
- \*\*Description:\*\* Stores customer contact information such as phone number and address.  
- \*\*Inputs:\*\* New or updated contact information from the user interface.  
- \*\*Outputs:\*\* Updated contact information for display and use in other functions.  
- \*\*Interaction Method:\*\* Direct database reads and writes with validation and logging.  
  
### 2.3.3 Product Data Entity  
- \*\*Description:\*\* Stores product information such as name, description, price, category, and inventory quantity.  
- \*\*Inputs:\*\* Product details from the user interface or system processes (e.g., checkout).  
- \*\*Outputs:\*\* Updated product information for display and use in other functions.  
- \*\*Interaction Method:\*\* Direct database reads and writes with validation and logging.  
  
### 2.3.4 Category Data Entity  
- \*\*Description:\*\* Stores product categories and their associations with products.  
- \*\*Inputs:\*\* Category details from the user interface or system processes.  
- \*\*Outputs:\*\* Updated category information for display and use in other functions.  
- \*\*Interaction Method:\*\* Direct database reads and writes with validation and logging.  
  
### 2.3.5 Inventory Data Entity  
- \*\*Description:\*\* Stores and manages inventory levels for all products.  
- \*\*Inputs:\*\* Inventory updates from the user interface or system processes (e.g., product addition, shopping cart modification, checkout).  
- \*\*Outputs:\*\* Updated inventory levels for display and use in other functions.  
- \*\*Interaction Method:\*\* Direct database reads and writes with validation and logging.  
  
### 2.3.6 Shopping Cart Data Entity  
- \*\*Description:\*\* Stores the contents of a customer's shopping cart, including product IDs, quantities, and pricing.  
- \*\*Inputs:\*\* Product selections and quantity adjustments from the user interface.  
- \*\*Outputs:\*\* Updated shopping cart for display and use in checkout or modification functions.  
- \*\*Interaction Method:\*\* Direct database reads and writes with validation and logging.  
  
### 2.3.7 CartItem Data Entity  
- \*\*Description:\*\* Stores individual items in the shopping cart, including product details and quantities.  
- \*\*Inputs:\*\* Cart item modifications from the user interface.  
- \*\*Outputs:\*\* Updated cart items for display and use in checkout or modification functions.  
- \*\*Interaction Method:\*\* Direct database reads and writes with validation and logging.  
  
### 2.3.8 Order Data Entity  
- \*\*Description:\*\* Stores customer orders, including order number, date, product details, customer information, and status.  
- \*\*Inputs:\*\* Order creation or modification from the user interface or system processes.  
- \*\*Outputs:\*\* Updated order information for display and use in other functions.  
- \*\*Interaction Method:\*\* Direct database reads and writes with validation and logging.  
  
### 2.3.9 OrderItem Data Entity  
- \*\*Description:\*\* Stores individual items within an order, including product ID, quantity, and price.  
- \*\*Inputs:\*\* Order item modifications from the user interface or system processes.  
- \*\*Outputs:\*\* Updated order items for display and use in order management functions.  
- \*\*Interaction Method:\*\* Direct database reads and writes with validation and logging.  
  
### 2.3.10 Payment Data Entity  
- \*\*Description:\*\* Stores payment information, including payment method, amount, date, and status.  
- \*\*Inputs:\*\* Payment details from the user interface or external payment gateways.  
- \*\*Outputs:\*\* Updated payment information for display and use in other functions.  
- \*\*Interaction Method:\*\* Direct database writes for internal payments and API calls for external payment gateways.  
  
### 2.3.11 EmailRecord Data Entity  
- \*\*Description:\*\* Stores all email records, including order confirmations, account updates, and other notifications.  
- \*\*Inputs:\*\* Email content and metadata from the user interface or system-generated events.  
- \*\*Outputs:\*\* Updated email records for display and use in other functions.  
- \*\*Interaction Method:\*\* Direct database reads and writes with validation and logging.  
  
### 2.3.12 Plugin Data Entity  
- \*\*Description:\*\* Stores plugin details, configuration settings, and activation status.  
- \*\*Inputs:\*\* Plugin files or configuration settings from the user interface.  
- \*\*Outputs:\*\* Updated plugin data for system use and display.  
- \*\*Interaction Method:\*\* Direct database writes for plugin installation and configuration.  
  
### 2.3.13 Purchase History Data Entity  
- \*\*Description:\*\* Logs all user interactions related to account activity, product and order management, and system events.  
- \*\*Inputs:\*\* Event details from functional processes (e.g., login, logout, checkout).  
- \*\*Outputs:\*\* Updated event logs for display and use in audit or reporting functions.  
- \*\*Interaction Method:\*\* Direct database writes with event tracking and logging.  
  
### 2.3.14 User Interface Specification Data Entity  
- \*\*Description:\*\* Stores the current UI specification for the system, including design rules and functionality requirements.  
- \*\*Inputs:\*\* Updated UI specification details from the user interface.  
- \*\*Outputs:\*\* Updated UI specification for system use and display.  
- \*\*Interaction Method:\*\* Direct database writes with validation and logging.  
  
## 2.4 Communication Interface  
  
The system communicates with external services and users through various communication channels, including email notifications and web-based interactions.  
  
### 2.4.1 Email Notification Interface  
- \*\*Description:\*\* The system sends email notifications to users for account confirmation, order confirmation, and other system events.  
- \*\*Inputs:\*\* Email content and recipient information from the system.  
- \*\*Outputs:\*\* Email sent to the specified address.  
- \*\*Interaction Method:\*\* Integration with an email service provider (e.g., SMTP, third-party email APIs) to send and log emails.  
  
### 2.4.2 Payment Gateway Interface  
- \*\*Description:\*\* The system communicates with external payment gateways to process customer payments.  
- \*\*Inputs:\*\* Payment method and payment details from the user interface.  
- \*\*Outputs:\*\* Payment confirmation or error message from the payment gateway.  
- \*\*Interaction Method:\*\* API calls to the payment gateway with secure data exchange and response handling.  
  
### 2.4.3 Web Browser Communication Interface  
- \*\*Description:\*\* The system is accessed through web browsers, and interactions are handled via HTTP/HTTPS protocols.  
- \*\*Inputs:\*\* User actions via browser (e.g., form submissions, navigation).  
- \*\*Outputs:\*\* Dynamic web page updates and data retrieval.  
- \*\*Interaction Method:\*\* RESTful API or web service communication with client-side rendering or server-side rendering depending on the architecture.  
  
### 2.4.4 Plugin Communication Interface  
- \*\*Description:\*\* The system may communicate with external plugins to extend its functionality.  
- \*\*Inputs:\*\* Plugin configuration or activation signals from the user interface.  
- \*\*Outputs:\*\* Updated system functionality or interface changes based on the plugin.  
- \*\*Interaction Method:\*\* Plugin APIs or system hooks to integrate and activate new functionalities.  
  
---  
  
This section defines all external interfaces that the system interacts with, ensuring clarity and consistency for developers and stakeholders. Each interface is categorized based on its role and method of interaction, with clear definitions and input/output examples to reduce implementation ambiguity.