# Functional Requirement

\*\*Chapter 1: Functional Requirements\*\*   
\*System Name: GAMMA-J Web Store\*  
  
---  
  
### 1.1 Product Management Function   
\*\*Function ID:\*\* FR-01   
\*\*Description:\*\* Administrators and plugins can create, update, and delete products. Each product is associated with a category and can be part of an order.   
\*\*Input:\*\*   
- Product details (Name, Description, Price, Stock, CategoryID)   
- Administrator credentials and permissions   
- Plugin configuration (if applicable)   
\*\*Output:\*\*   
- A new or updated product record in the database   
- A notification to the administrator confirming the action   
- Error message (if validation or technical failure occurs)  
  
---  
  
### 1.2 Product View Function   
\*\*Function ID:\*\* FR-02   
\*\*Description:\*\* Administrators and customers can view the details of a specific product. This function retrieves and displays product information such as name, description, price, and stock status.   
\*\*Input:\*\*   
- ProductID or product name for search   
- User credentials (Administrator or Customer)   
\*\*Output:\*\*   
- Displayed product details   
- Error message if product not found or system failure occurs  
  
---  
  
### 1.3 Order Management Function   
\*\*Function ID:\*\* FR-03   
\*\*Description:\*\* Administrators can create, update, and delete orders. Orders are associated with a customer, and each order contains one or more order items.   
\*\*Input:\*\*   
- Order details (CustomerID, OrderDate, TotalAmount, Status)   
- OrderItems (ProductID, Quantity, Subtotal)   
- Administrator credentials and permissions   
\*\*Output:\*\*   
- A new or updated order record in the database   
- A notification to the administrator confirming the action   
- Error message if order or order item is invalid, or if system or plugin fails  
  
---  
  
### 1.4 Order View Function   
\*\*Function ID:\*\* FR-04   
\*\*Description:\*\* Administrators and customers can view the details of a specific order, including the associated products, payment status, and order status.   
\*\*Input:\*\*   
- OrderID   
- User credentials (Administrator or Customer)   
\*\*Output:\*\*   
- Displayed order details   
- Error message if order not found or system failure occurs  
  
---  
  
### 1.5 Order Status Update Function   
\*\*Function ID:\*\* FR-05   
\*\*Description:\*\* Administrators can update the status of an order (e.g., Processing, Shipped, Cancelled). The system validates the status and updates related records, including inventory and notifications.   
\*\*Input:\*\*   
- OrderID   
- New status (e.g., "Shipped")   
- Administrator credentials and permissions   
\*\*Output:\*\*   
- Updated order status in the database   
- Notifications to customer and administrator   
- Error message if invalid status or system/plugin failure occurs  
  
---  
  
### 1.6 Order Item Management Function   
\*\*Function ID:\*\* FR-06   
\*\*Description:\*\* Administrators can manage order items by editing or removing them from an order. This function updates the order total and product inventory accordingly.   
\*\*Input:\*\*   
- OrderItemID   
- Updated quantity or price (if editing)   
- Administrator credentials and permissions   
\*\*Output:\*\*   
- Updated or removed order item in the database   
- Updated order total and inventory   
- Notification to the administrator   
- Error message if order is finalized or system/plugin failure occurs  
  
---  
  
### 1.7 Customer Management Function   
\*\*Function ID:\*\* FR-07   
\*\*Description:\*\* Administrators can create, update, and delete customer records. A customer can also view their own profile.   
\*\*Input:\*\*   
- Customer details (Name, Email, PhoneNumber, Address)   
- Administrator credentials and permissions   
\*\*Output:\*\*   
- A new or updated customer record in the database   
- A notification to the administrator confirming the action   
- Error message if invalid data or dependencies exist  
  
---  
  
### 1.8 Customer View Function   
\*\*Function ID:\*\* FR-08   
\*\*Description:\*\* Customers and administrators can view a customer's profile and order history.   
\*\*Input:\*\*   
- CustomerID   
- User credentials (Administrator or Customer)   
\*\*Output:\*\*   
- Displayed customer profile and order history   
- Error message if customer not found or system failure occurs  
  
---  
  
### 1.9 Payment Processing Function   
\*\*Function ID:\*\* FR-09   
\*\*Description:\*\* The system processes payments for orders using a selected payment method. This function updates the payment and order status and sends a confirmation notification.   
\*\*Input:\*\*   
- OrderID   
- PaymentMethodID   
- Payment amount and date   
- User credentials (Customer or Administrator)   
\*\*Output:\*\*   
- A new or updated payment record in the database   
- Updated order status to "Paid"   
- Notification to customer and administrator   
- Error message if payment fails, order is invalid, or system/plugin failure occurs  
  
---  
  
### 1.10 Payment View Function   
\*\*Function ID:\*\* FR-10   
\*\*Description:\*\* Administrators and customers can view the details of a specific payment, including transaction ID, amount, method, and status.   
\*\*Input:\*\*   
- PaymentID   
- User credentials (Administrator or Customer)   
\*\*Output:\*\*   
- Displayed payment details   
- Error message if payment not found or system failure occurs  
  
---  
  
### 1.11 Notification Management Function   
\*\*Function ID:\*\* FR-11   
\*\*Description:\*\* The system can send and view notifications to users (customers and administrators). Notifications may be sent via plugins (e.g., email, SMS).   
\*\*Input:\*\*   
- Notification content (Message)   
- Recipient details (CustomerID or AdminID)   
- Plugin configuration (if applicable)   
\*\*Output:\*\*   
- Notification record in the database   
- Delivered notification to the user   
- Error message if recipient is invalid or plugin fails  
  
---  
  
### 1.12 Notification View Function   
\*\*Function ID:\*\* FR-12   
\*\*Description:\*\* Users (administrators and customers) can view their notifications. The system marks notifications as viewed upon display.   
\*\*Input:\*\*   
- UserID (Administrator or Customer)   
- User credentials   
\*\*Output:\*\*   
- List of notifications with content and timestamp   
- Marked notifications as viewed   
- Error message if no notifications or system failure occurs  
  
---  
  
### 1.13 Plugin Management Function   
\*\*Function ID:\*\* FR-13   
\*\*Description:\*\* Administrators can manage plugins, including adding, removing, and updating plugin configurations. Plugins can extend system functionality for notifications, payments, and order processing.   
\*\*Input:\*\*   
- Plugin details (Name, Description, Version, Configuration)   
- Plugin file (if adding a new plugin)   
- Administrator credentials and permissions   
\*\*Output:\*\*   
- Plugin record in the database   
- Plugin status updated (activated, deactivated, or removed)   
- Notification to the administrator   
- Error message if plugin is incompatible, fails to initialize, or in use  
  
---  
  
### 1.14 Product Category Management Function   
\*\*Function ID:\*\* FR-14   
\*\*Description:\*\* Administrators can manage product categories by creating, updating, or deleting them. This function ensures that products are correctly categorized and that dependencies are handled.   
\*\*Input:\*\*   
- Category details (Name, Description, ParentCategory)   
- Administrator credentials and permissions   
\*\*Output:\*\*   
- Updated or new product category record in the database   
- Notification to the administrator   
- Error message if category is in use or system/plugin failure occurs  
  
---  
  
### 1.15 Payment Method Management Function   
\*\*Function ID:\*\* FR-15   
\*\*Description:\*\* Administrators can manage available payment methods by creating, updating, or deleting them. This function ensures that payment methods are compatible with the system and not currently in use.   
\*\*Input:\*\*   
- Payment method details (Name, Description, Configuration)   
- Administrator credentials and permissions   
\*\*Output:\*\*   
- Updated or new payment method record in the database   
- Notification to the administrator   
- Error message if method is in use or system/plugin failure occurs  
  
---  
  
This concludes \*\*Chapter 1: Functional Requirements\*\* of the \*\*GAMMA-J Web Store Software Requirement Specification (SRS)\*\*. Each function is aligned with the provided use cases and data model, ensuring feasibility, completeness, and consistency.

# External Description

# \*\*Chapter 2: External Interfaces\*\*   
\*\*System Name: GAMMA-J Web Store\*\*  
  
This chapter defines the external interfaces that the GAMMA-J Web Store system must interact with to fulfill its functional requirements. These interfaces include user interfaces, hardware interfaces, software interfaces, and communication interfaces. Each interface is described in detail, including its purpose, interaction method, and relevant input/output data. The goal is to ensure that developers and stakeholders understand the external dependencies and communication channels required for the system to operate effectively.  
  
---  
  
## \*\*2.1 User Interface\*\*  
  
The GAMMA-J Web Store interacts with end users through a web-based user interface (UI) that is accessible via modern web browsers. The UI is designed to be responsive and user-friendly, supporting both administrators and customers in performing their respective tasks.  
  
### \*\*2.1.1 Administrator Interface\*\*  
- \*\*Purpose:\*\* Provides administrators with tools to manage products, orders, customers, payment methods, and plugins.  
- \*\*Interaction Method:\*\* Web-based dashboard with forms, buttons, and interactive elements.  
- \*\*Inputs:\*\*  
 - Product details (Name, Description, Price, Stock, CategoryID)  
 - Order details (CustomerID, OrderDate, TotalAmount, Status)  
 - OrderItems (ProductID, Quantity, Subtotal)  
 - Customer details (Name, Email, PhoneNumber, Address)  
 - Payment method details (Name, Description, Configuration)  
 - Plugin details (Name, Description, Version, Configuration)  
 - User credentials for authentication and authorization  
- \*\*Outputs:\*\*  
 - Confirmation messages for successful actions (e.g., "Product created successfully")  
 - Error messages for invalid input or system failures  
 - Updated displays of product, order, and customer data  
 - Notifications (e.g., success, warning, or error messages)  
 - Plugin status updates (activated, deactivated, removed)  
  
### \*\*2.1.2 Customer Interface\*\*  
- \*\*Purpose:\*\* Allows customers to view product details, place and view orders, check order status, and review their payment and notification history.  
- \*\*Interaction Method:\*\* Web-based front-end with search functionality and order management panels.  
- \*\*Inputs:\*\*  
 - ProductID or product name for search  
 - OrderID for viewing or managing an order  
 - CustomerID for viewing their profile and order history  
 - PaymentMethodID for selecting a payment method  
 - User credentials for authentication and authorization  
- \*\*Outputs:\*\*  
 - Displayed product details (name, description, price, stock status)  
 - Displayed order details (products, status, payment status)  
 - Displayed customer profile and order history  
 - Displayed payment details (transaction ID, amount, method, status)  
 - Notifications (e.g., order status updates, payment confirmations)  
 - Error messages for invalid inputs or system failures  
  
---  
  
## \*\*2.2 Hardware Interface\*\*  
  
The GAMMA-J Web Store does not have direct dependencies on specific hardware components beyond the standard infrastructure required for web application deployment and user access.  
  
### \*\*2.2.1 Web Browsers\*\*  
- \*\*Purpose:\*\* Enable user interaction with the system via graphical interfaces.  
- \*\*Description:\*\* The system must be compatible with major web browsers, including Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.  
- \*\*Supported Devices:\*\* Desktop computers, laptops, tablets, and mobile devices.  
- \*\*Requirements:\*\*  
 - Support for HTML5, CSS3, and modern JavaScript frameworks  
 - Responsive design for cross-device compatibility  
 - No special hardware requirements beyond a standard internet-connected device  
  
---  
  
## \*\*2.3 Software Interface\*\*  
  
The GAMMA-J Web Store interacts with several software components, including databases, plugins, and external tools for payment processing, notification delivery, and system management.  
  
### \*\*2.3.1 Database Interface\*\*  
- \*\*Purpose:\*\* Store and retrieve product, customer, order, payment, and notification data.  
- \*\*Description:\*\* The system interacts with a relational database (e.g., MySQL, PostgreSQL) to manage structured data.  
- \*\*Interaction Method:\*\* SQL queries or an ORM (Object-Relational Mapping) layer for database operations.  
- \*\*Inputs:\*\*  
 - Product, order, customer, payment, and notification data  
 - Queries for data retrieval (e.g., by ProductID, OrderID, CustomerID)  
- \*\*Outputs:\*\*  
 - Updated or newly inserted records in the database  
 - Retrieved records for display or further processing  
 - Error messages if database operations fail (e.g., invalid query, connection failure)  
  
### \*\*2.3.2 Plugin Interface\*\*  
- \*\*Purpose:\*\* Allow administrators to extend system functionality via third-party or custom plugins.  
- \*\*Description:\*\* Plugins are integrated into the system to support notifications, payment processing, and order management.  
- \*\*Interaction Method:\*\* Plugin files are uploaded and configured through the system’s Plugin Management UI. Each plugin must conform to a defined API or SDK to interact with the system.  
- \*\*Inputs:\*\*  
 - Plugin configuration data (Name, Description, Version, Configuration)  
 - Plugin files (e.g., .zip or .jar)  
 - Trigger events from the system (e.g., "Order Created", "Payment Success")  
- \*\*Outputs:\*\*  
 - Plugin status updates (activated, deactivated, removed)  
 - Notifications sent to users (via email, SMS, etc.)  
 - Payment processing results (success, failure, pending)  
 - Error messages if the plugin is incompatible or fails to initialize  
  
### \*\*2.3.3 Payment Gateway Interface\*\*  
- \*\*Purpose:\*\* Interface with external payment gateways to process customer payments.  
- \*\*Description:\*\* The system must support integration with various payment gateways (e.g., PayPal, Stripe, Alipay) via their APIs.  
- \*\*Interaction Method:\*\* RESTful API calls to the external payment gateway with appropriate authentication (e.g., API keys).  
- \*\*Inputs:\*\*  
 - OrderID and payment amount  
 - PaymentMethodID to determine the correct gateway  
 - Transaction data (e.g., customer information, payment method details)  
- \*\*Outputs:\*\*  
 - Payment confirmation or failure status  
 - Transaction ID and timestamp  
 - Updated payment and order status in the system  
 - Error messages for failed transactions or invalid inputs  
  
---  
  
## \*\*2.4 Communication Interface\*\*  
  
The system communicates with external entities using standard communication protocols and methods, including email, web services, and message queues.  
  
### \*\*2.4.1 Email Communication Interface\*\*  
- \*\*Purpose:\*\* Send notifications to users via email.  
- \*\*Description:\*\* The system sends emails to administrators and customers to confirm actions such as order updates, payment success, and plugin changes.  
- \*\*Interaction Method:\*\* SMTP (Simple Mail Transfer Protocol) or REST API integration with email service providers (e.g., SendGrid, Mailgun).  
- \*\*Inputs:\*\*  
 - Recipient email address (derived from user account)  
 - Notification message content (text or HTML)  
 - Timestamp of the event  
- \*\*Outputs:\*\*  
 - Email sent to the user  
 - Notification record stored in the database  
 - Error messages if the email fails to send  
  
### \*\*2.4.2 SMS Communication Interface (Optional)\*\*  
- \*\*Purpose:\*\* Send text notifications to users via SMS.  
- \*\*Description:\*\* The system may support SMS delivery for real-time notifications (e.g., order status changes, payment confirmations).  
- \*\*Interaction Method:\*\* REST API integration with SMS service providers (e.g., Twilio, Nexmo).  
- \*\*Inputs:\*\*  
 - Recipient phone number (from user account)  
 - Notification message content  
 - Timestamp of the event  
- \*\*Outputs:\*\*  
 - SMS message sent to the user  
 - Notification record stored in the database  
 - Error messages if the SMS fails to send  
  
### \*\*2.4.3 Web Communication Interface\*\*  
- \*\*Purpose:\*\* Enable interaction between the system and external web services or APIs.  
- \*\*Description:\*\* The system uses HTTP/HTTPS for communication with external systems, such as payment gateways and notification plugins.  
- \*\*Interaction Method:\*\* RESTful API calls with JSON or XML payloads.  
- \*\*Inputs:\*\*  
 - API request parameters (e.g., OrderID, PaymentMethodID, Plugin configuration)  
 - Authentication tokens or API keys  
- \*\*Outputs:\*\*  
 - API response data (e.g., payment confirmation, plugin status)  
 - Error responses if the external service is unavailable or returns an error  
  
### \*\*2.4.4 Notification Push Interface (Optional)\*\*  
- \*\*Purpose:\*\* Deliver real-time notifications to users via push messages (e.g., web push, mobile push).  
- \*\*Description:\*\* This interface allows the system to send instant alerts to users who have opted-in.  
- \*\*Interaction Method:\*\* Integration with push notification services (e.g., Firebase Cloud Messaging, Apple Push Notification Service).  
- \*\*Inputs:\*\*  
 - User device token  
 - Notification message content  
 - Timestamp and priority level  
- \*\*Outputs:\*\*  
 - Push notification delivered to the user's device  
 - Notification record stored in the database  
 - Error messages if the push fails (e.g., invalid token)  
  
---  
  
## \*\*2.5 External Data Sources\*\*  
  
All external data sources referenced in the functional requirements are covered under the \*\*Software Interface\*\* (Section 2.3), particularly the \*\*Database Interface\*\* and \*\*Plugin Interface\*\*. The system does not rely on external hardware data sources directly but may access external APIs for data processing and delivery.  
  
- \*\*Database:\*\* Used to store and retrieve all core system data including products, orders, customers, and notifications.  
- \*\*Plugins:\*\* Extend the system's functionality and may interface with external services such as payment gateways or notification channels.  
- \*\*Payment Gateways:\*\* External services used to process transactions and validate payment methods.  
  
---  
  
## \*\*2.6 Interface Summary\*\*  
  
| Interface Type | Description | Interaction Method |  
|-----------------------|-----------------------------------------------------------------------------|-----------------------------------------------------|  
| User Interface | Web-based UI for administrators and customers | Web forms, buttons, and interactive dashboards |  
| Web Browsers | Standard web browser interaction | HTML5, CSS3, JavaScript |  
| Database Interface | Relational database for storing and retrieving system data | SQL or ORM (e.g., Hibernate, Sequelize) |  
| Plugin Interface | Integration with third-party or custom plugins | REST API or system-defined plugin SDK |  
| Payment Gateway | External services for processing customer payments | RESTful API calls (e.g., Stripe, PayPal) |  
| Email Communication | Sending notifications via email | SMTP or REST API (e.g., SendGrid) |  
| SMS Communication | Optional interface for sending SMS notifications | REST API (e.g., Twilio) |  
| Notification Push | Optional interface for real-time push notifications | Push notification services (e.g., Firebase) |  
| Web Communication | General HTTP/HTTPS communication for API integrations | RESTful API calls with JSON/XML payloads |  
  
---  
  
This chapter concludes the definition of \*\*Chapter 2: External Interfaces\*\* in the \*\*GAMMA-J Web Store Software Requirements Specification (SRS)\*\*. It provides a comprehensive overview of how the system interacts with users, hardware, software, and external communication channels. Developers can use this section to understand the required integrations and ensure that the system is built with compatibility and scalability in mind.