# Functional Requirement

1. \*\*Functional Requirements\*\*  
  
1.1 \*\*Customer Account Registration\*\*   
Function ID: FR-01   
Description: Customers can register a new account by providing valid personal information. The system validates the email format and password strength and calls the Plugin API to ensure the email is not already registered.   
Input: Customer personal information (FirstName, LastName, Email, Password).   
Output: A new Customer record stored in the database with a "Verified" status.  
  
1.2 \*\*Customer Login\*\*   
Function ID: FR-02   
Description: Customers can log in using their registered email and password. The system validates the credentials and calls the Plugin API for authentication.   
Input: Email and password.   
Output: A valid Session record created and associated with the Customer.  
  
1.3 \*\*Customer Logout\*\*   
Function ID: FR-03   
Description: Customers can log out of the system, which invalidates their current session.   
Input: User session token.   
Output: Session invalidated and system logs the event.  
  
1.4 \*\*Customer Profile Management\*\*   
Function ID: FR-04   
Description: Customers can view, update, or delete their own profile information, such as name, email, and billing address.   
Input: Customer profile data (FirstName, LastName, Email, BillingAddress, etc.).   
Output: Updated or deleted Customer record in the database.  
  
1.5 \*\*Customer Purchase History View\*\*   
Function ID: FR-05   
Description: Customers can view a list of their completed orders, including order date, items, and total cost.   
Input: CustomerID.   
Output: A list of Order records with associated OrderDetail and Product data.  
  
1.6 \*\*Shopping Cart Management\*\*   
Function ID: FR-06   
Description: Customers can manage the contents of their shopping cart, including adding, removing, or modifying items.   
Input: ProductID, Quantity, CartID, ActionType (Add, Remove, Modify).   
Output: Updated CartItem records in the database with current cart state and recalculated total.  
  
1.7 \*\*Shopping Cart Checkout\*\*   
Function ID: FR-07   
Description: Customers can proceed to checkout by selecting a valid payment method, confirming the order, and converting the shopping cart to an order.   
Input: CartID, PaymentID, ShippingAddress.   
Output: A new Order record created with associated OrderDetails, and the CartItem records deleted.  
  
1.8 \*\*Order Status View\*\*   
Function ID: FR-08   
Description: Customers can view the current status of their orders, such as Processing, Shipped, Delivered, or Cancelled.   
Input: OrderID.   
Output: Order status and related details retrieved and displayed.  
  
1.9 \*\*Order Cancellation\*\*   
Function ID: FR-09   
Description: Customers can cancel an order if it has not yet been shipped. The system updates the order status and logs the event.   
Input: OrderID.   
Output: Order status updated to "Canceled" and the event is logged.  
  
1.10 \*\*Payment Information Management\*\*   
Function ID: FR-10   
Description: Customers can add, update, or delete their payment information.   
Input: CardNumberHash, ExpiryDate, HolderName, BillingAddress, ActionType (Add, Update, Delete).   
Output: Updated or deleted PaymentInfo records in the database.  
  
1.11 \*\*Inventory Management\*\*   
Function ID: FR-12   
Description: Administrators can manage product inventory by adding, updating, or deleting product records.   
Input: ProductID, Name, Description, Price, StockQuantity, ImageURL, CategoryID, ActionType (Add, Update, Delete).   
Output: Updated or deleted Product records in the database.  
  
1.12 \*\*Product Category Management\*\*   
Function ID: FR-13   
Description: Administrators can manage product categories by creating, updating, or deleting Category records.   
Input: CategoryID, Name, Description, ActionType (Create, Update, Delete).   
Output: Updated or deleted Category records in the database.  
  
1.13 \*\*Plugin Installation and Management\*\*   
Function ID: FR-14   
Description: Administrators can install, configure, or uninstall plugins to extend system functionality.   
Input: PluginID, ConfigurationSettings, ActionType (Install, Configure, Uninstall).   
Output: Updated PluginInstance records in the database, and system configuration adjusted accordingly.  
  
1.14 \*\*Order Confirmation Email Notification\*\*   
Function ID: FR-15   
Description: The system sends an email confirmation to the customer after order creation.   
Input: OrderID, CustomerID, Email template data (e.g., items, total price, shipping details).   
Output: EmailNotification record created and Plugin API called to send the email.  
  
1.15 \*\*Discount Code Application\*\*   
Function ID: FR-16   
Description: Customers can apply a discount code during the checkout process to reduce the total cost.   
Input: DiscountCode, CartID.   
Output: Updated CartItem records with adjusted total price and a log of the applied discount.  
  
1.16 \*\*Plugin API Integration\*\*   
Function ID: FR-17   
Description: The system integrates with the Plugin API for payment processing, inventory checks, email notifications, and other external functions.   
Input: PluginID, API request parameters (varies per function).   
Output: Plugin API response confirmation and corresponding system action (e.g., payment processed, inventory updated).  
  
1.17 \*\*Session Management\*\*   
Function ID: FR-18   
Description: The system manages user sessions to ensure secure login and logout operations.   
Input: CustomerID, Session token, IP address, UserAgent.   
Output: Session record created, updated, or deleted based on user activity.  
  
1.18 \*\*Order Detail Management\*\*   
Function ID: FR-19   
Description: Administrators can view and manage the details of each order, such as quantity and unit price per item.   
Input: OrderID, ProductID, Quantity, UnitPrice.   
Output: Updated OrderDetail records in the database and recalculated order total.  
  
1.19 \*\*Shopping Cart to Order Conversion\*\*   
Function ID: FR-20   
Description: The system converts the contents of a Shopping Cart into an Order when the customer completes the checkout process.   
Input: CartID, CustomerID, PaymentID, ShippingAddress.   
Output: New Order and OrderDetail records created, CartItem records deleted, and inventory updated.  
  
1.20 \*\*Email Notification Sending\*\*   
Function ID: FR-21   
Description: The system sends email notifications to customers for events such as order confirmation or account registration.   
Input: CustomerID, Email content (Subject, Body), OrderID (optional).   
Output: EmailNotification record created and Plugin API called for delivery.  
  
1.21 \*\*Administrator Account Management\*\*   
Function ID: FR-22   
Description: Administrators can manage other administrator accounts by creating, updating, or deleting them.   
Input: AdminID, Username, PasswordHash, Role, Permissions, ActionType.   
Output: Updated or deleted Administrator records in the database.  
  
1.22 \*\*System Event Logging\*\*   
Function ID: FR-23   
Description: The system logs all user actions and events for auditing and troubleshooting purposes.   
Input: Event type (e.g., login, checkout, plugin installation), ActorID (CustomerID or AdminID), Timestamp.   
Output: Event log entry stored in the system.  
  
1.23 \*\*Shopping Cart Creation and Access\*\*   
Function ID: FR-24   
Description: A Shopping Cart is created when a customer logs in and can be accessed to view or modify items.   
Input: CustomerID.   
Output: New or existing Shopping Cart record retrieved and displayed.  
  
1.24 \*\*Product Availability Check\*\*   
Function ID: FR-25   
Description: The system checks the availability of products in the inventory via the Plugin API before adding them to the cart or proceeding to checkout.   
Input: ProductID.   
Output: Availability status returned from the Plugin API and reflected in the system.  
  
1.25 \*\*Shopping Cart Total Calculation\*\*   
Function ID: FR-26   
Description: The system calculates the total price of the Shopping Cart based on the quantity and price of the selected items.   
Input: CartItem records (ProductID, Quantity, UnitPrice).   
Output: Updated cart total price and display.  
  
1.26 \*\*Inventory Update on Purchase\*\*   
Function ID: FR-27   
Description: The system updates the stock quantity of products in the inventory after an order is successfully placed.   
Input: OrderDetail records (ProductID, Quantity).   
Output: Updated Product records with reduced StockQuantity.

# External Description

# Chapter 2: External Interfaces  
  
This chapter defines the external interfaces that the system interacts with. These interfaces are categorized into four types: \*\*User Interface\*\*, \*\*Hardware Interface\*\*, \*\*Software Interface\*\*, and \*\*Communication Interface\*\*. Each interface is described in terms of its purpose, interaction method, and relevant data flows to ensure clarity for developers and stakeholders.  
  
---  
  
## 2.1 User Interface  
  
### 2.1.1 Web Application UI   
\*\*Description:\*\* The primary user interface through which customers and administrators interact with the system. It includes forms for registration, login, profile management, product browsing, shopping cart viewing/editing, checkout process, order history viewing, and inventory/category/order management for administrators.   
\*\*Interaction Method:\*\* Users navigate web pages via browsers and interact with HTML forms, buttons, tables, and modal dialogs.   
\*\*Data Flows:\*\*  
- Input from users (e.g., personal information, payment details, product selections).  
- Output to users (e.g., shopping cart summary, order status, error messages).  
  
---  
  
## 2.2 Hardware Interface  
  
### 2.2.1 Customer Devices   
\*\*Description:\*\* The system supports access from standard customer devices such as desktops, laptops, tablets, and smartphones.   
\*\*Interaction Method:\*\* Standard HTTP(S) communication over wired or wireless internet connections.   
\*\*Supported Features:\*\*  
- Responsive design for different screen sizes.  
- Compatibility with major operating systems (Windows, macOS, iOS, Android).  
  
### 2.2.2 Administrator Workstations   
\*\*Description:\*\* Administrators use standard computing devices to manage products, categories, orders, plugins, and other administrative functions.   
\*\*Interaction Method:\*\* Same as above, but restricted to authenticated users with admin roles.   
\*\*Security Requirements:\*\* Secure login and session management required.  
  
---  
  
## 2.3 Software Interface  
  
### 2.3.1 Plugin API Integration   
\*\*Description:\*\* A third-party Plugin API used to perform critical functions including email verification during registration, payment processing, email notifications, inventory checks, and authentication.   
\*\*Interaction Method:\*\* RESTful API calls using HTTPS protocol.   
\*\*Key Functions Supported:\*\*  
- Email validation (used in FR-01, FR-15)  
- Payment processing (used in FR-07, FR-10)  
- Inventory availability check (used in FR-25)  
- Authentication (used in FR-02)  
- Email notification delivery (used in FR-15, FR-21)  
  
\*\*Sample Inputs/Outputs:\*\*  
- \*\*Input:\*\* PluginID, request parameters (e.g., email, payment token, template data)  
- \*\*Output:\*\* JSON response indicating success/failure, along with any necessary data (e.g., payment confirmation, email delivery status)  
  
### 2.3.2 Database Management System (DBMS)   
\*\*Description:\*\* Backend relational database system responsible for storing and managing all persistent data including customer accounts, orders, products, sessions, payments, logs, and plugin configurations.   
\*\*Interaction Method:\*\* SQL queries executed by the application server to read/write data.   
\*\*Key Tables Involved:\*\*  
- `Customer`, `Administrator`, `Product`, `Category`, `Order`, `OrderDetail`, `PaymentInfo`, `ShoppingCart`, `CartItem`, `PluginInstance`, `EmailNotification`, `EventLog`  
  
\*\*Data Integrity Requirements:\*\* ACID-compliant transactions for operations involving order creation, payment updates, and inventory changes (FR-07, FR-20, FR-27).  
  
### 2.3.3 Session Management Module   
\*\*Description:\*\* A software module that manages user sessions securely, handling login, logout, and session expiration.   
\*\*Interaction Method:\*\* Token-based authentication using session tokens stored in cookies or local storage.   
\*\*Security Mechanisms:\*\*  
- Encrypted session tokens  
- IP address and User-Agent validation  
- Expiration after inactivity or explicit logout  
  
\*\*Used By:\*\* FR-02, FR-03, FR-18  
  
---  
  
## 2.4 Communication Interface  
  
### 2.4.1 Email Notification Service   
\*\*Description:\*\* External email service used to send automated emails to customers for events like account registration, order confirmation, and other notifications.   
\*\*Interaction Method:\*\* Triggered via Plugin API (see Section 2.3.1), using SMTP or similar protocols.   
\*\*Sample Emails Sent:\*\*  
- Account registration confirmation (FR-01)  
- Order confirmation (FR-15, FR-21)  
  
\*\*Inputs Required:\*\*  
- Recipient email address  
- Email subject and body content  
- Optional attachments or template variables  
  
\*\*Output:\*\* Email sent confirmation or failure status recorded in `EmailNotification` table.  
  
### 2.4.2 Web Browsing (HTTP/HTTPS)   
\*\*Description:\*\* Primary communication protocol used between client browsers and the server.   
\*\*Interaction Method:\*\* RESTful endpoints exposed over HTTPS for secure data transmission.   
\*\*Endpoints Used For:\*\*  
- Registration, login/logout (FR-01–03)  
- Shopping cart operations (FR-06, FR-24, FR-26)  
- Order processing (FR-07–09, FR-20, FR-27)  
- Profile and payment management (FR-04, FR-10)  
- Admin operations (FR-11–14, FR-19, FR-22)  
  
\*\*Security Requirements:\*\*  
- TLS encryption  
- Input validation and sanitization  
- Protection against CSRF/XSS attacks  
  
---  
  
## Cross-Check Summary  
  
| Functional Requirement | Referenced External Data Source | Covered in This Chapter |  
|------------------------|----------------------------------|----------------------------|  
| FR-01 (Registration) | Plugin API (email check), DBMS (customer data) | Yes (2.3.1, 2.3.2) |  
| FR-02 (Login) | Plugin API (auth), DBMS, Session Module | Yes (2.3.1, 2.3.2, 2.3.3) |  
| FR-03 (Logout) | Session Module | Yes (2.3.3) |  
| FR-04 (Profile) | DBMS | Yes (2.3.2) |  
| FR-05 (Purchase History) | DBMS | Yes (2.3.2) |  
| FR-06–07, 20, 26–27 (Cart & Checkout) | DBMS, Plugin API (inventory), Session | Yes (2.3.1, 2.3.2, 2.3.3) |  
| FR-08–09 (Order Status/Cancellation) | DBMS | Yes (2.3.2) |  
| FR-10 (Payment Info) | DBMS | Yes (2.3.2) |  
| FR-11–12, 19 (Admin Inventory/Products) | DBMS | Yes (2.3.2) |  
| FR-13 (Categories) | DBMS | Yes (2.3.2) |  
| FR-14 (Plugins) | DBMS, Plugin API | Yes (2.3.1, 2.3.2) |  
| FR-15, 21 (Email Notif) | Plugin API, Email Service | Yes (2.3.1, 2.4.1) |  
| FR-16 (Discount Code) | DBMS (optional) | Implied under DBMS (2.3.2) |  
| FR-17 (Plugin API) | Plugin API | Yes (2.3.1) |  
| FR-18 (Session) | Session Module | Yes (2.3.3) |  
| FR-22 (Admin Accounts) | DBMS | Yes (2.3.2) |  
| FR-23 (Event Logging) | DBMS | Yes (2.3.2) |  
| FR-24–25 (Cart & Availability) | DBMS, Plugin API | Yes (2.3.1, 2.3.2) |  
  
All external data sources referenced in the functional requirements have been clearly defined and categorized in this chapter.