# Functional Requirement

1. Functional Requirements   
1.1 User Registration Function   
Function ID: FR-01   
Description: The system allows a new user to register an account by providing personal information such as name, email, and password. The system validates the input and sends a confirmation email to the user.   
Input: Name, Email, Password, Preferred Category (optional)   
Output: A new UserAccount entity with a unique AccountID, and a new Customer entity with a unique CustomerID. A confirmation email is sent to the customer’s email address.   
  
1.2 User Login Function   
Function ID: FR-02   
Description: The system allows a registered user to log in by providing a valid email and password. The system authenticates the user and initiates a session.   
Input: Email, Password   
Output: Active session for the user, and a logged login event in the system log.   
  
1.3 User Logout Function   
Function ID: FR-03   
Description: The system allows a logged-in user to terminate their session by clicking the "Logout" button.   
Input: User session information   
Output: Termination of the session, redirection to the login or home page, and a logged logout event in the system log.   
  
1.4 Customer Account Creation Function   
Function ID: FR-04   
Description: The system allows a customer to create their account by entering personal information and agreeing to the terms and conditions.   
Input: Name, Email, Password, Terms and Conditions acceptance   
Output: A new Customer entity with a unique CustomerID and a new UserAccount entity linked to the customer. A confirmation email is sent to the customer’s email address.   
  
1.5 Customer Information Update Function   
Function ID: FR-05   
Description: The system allows a logged-in customer to update their personal and contact information in their account settings.   
Input: Updated Name, Address, Phone, Email (if changed), and valid session   
Output: Updated Customer entity in the database and a confirmation message to the customer. A logged update event is recorded in the system log.   
  
1.6 View Purchase History Function   
Function ID: FR-06   
Description: The system allows a logged-in customer or administrator to view the customer’s purchase history, including order details.   
Input: CustomerID and valid session   
Output: A list of PurchaseHistory entities for the customer, including HistoryID, ProductID, PurchaseDate, Quantity, TotalCost. A logged view event is recorded in the system log.   
  
1.7 Manage Purchase History Function   
Function ID: FR-07   
Description: The system allows an administrator to view, edit, or delete a customer’s purchase history.   
Input: CustomerID, selected PurchaseHistory record, and valid administrator session   
Output: Updated or deleted PurchaseHistory records in the database and a confirmation message to the administrator. A logged management event is recorded in the system log.   
  
1.8 Add Product to Cart Function   
Function ID: FR-08   
Description: The system allows a logged-in customer to add a product to their shopping cart, including the product quantity.   
Input: ProductID, Quantity, CustomerID, and valid session   
Output: Updated ShoppingCart entity with the added Product and updated TotalAmount. A logged add event is recorded in the system log.   
  
1.9 Modify Cart Contents Function   
Function ID: FR-09   
Description: The system allows a logged-in customer to modify the quantity of a product or remove it from their shopping cart.   
Input: CartID, ProductID, Quantity (new), and valid session   
Output: Updated ShoppingCart entity in the database and a confirmation message to the customer. A logged modification event is recorded in the system log.   
  
1.10 View Cart Summary Function   
Function ID: FR-10   
Description: The system allows a logged-in customer to view the summary of their shopping cart, including item details, quantities, prices, and total cost.   
Input: CustomerID and valid session   
Output: A list of products in the ShoppingCart, including product names, quantities, prices, and total cost. Estimated shipping cost and final total amount are displayed. A logged view event is recorded in the system log.   
  
1.11 Checkout Order Function   
Function ID: FR-11   
Description: The system allows a logged-in customer to complete the checkout process by entering shipping information and payment details. The system creates an Order, processes the PaymentInformation, and updates the Inventory.   
Input: ShoppingCartID, ShippingAddress, PaymentInformation (CardNumber, ExpiryDate, CVV, Amount), and valid session   
Output: A new Order entity, updated PaymentInformation, updated Inventory, and a confirmation message to the customer. A logged checkout event is recorded in the system log.   
  
1.12 Confirm Order Function   
Function ID: FR-12   
Description: The system confirms the order after the checkout process is completed, updating the order status and inventory, and generating an OrderConfirmationEmail.   
Input: OrderID, ShoppingCartID, Inventory records, and valid session   
Output: Updated Order status to "confirmed," updated Inventory, and a new OrderConfirmationEmail sent to the customer. A logged confirmation event is recorded in the system log.   
  
1.13 Send Order Confirmation Email Function   
Function ID: FR-13   
Description: The system sends an email confirmation to the customer after the order has been successfully created and payment processed.   
Input: OrderID, Customer Email, Order details (items, total amount, estimated delivery date), and valid email server connection   
Output: A new OrderConfirmationEmail entity, including EmailID, OrderID, SentDate, and Content. A logged email sending event is recorded in the system log.   
  
1.14 Product Inventory Management Function   
Function ID: FR-14   
Description: The system allows an administrator to manage inventory, including updating product stock quantities and other related details.   
Input: ProductID, New Inventory Quantity, and valid administrator session   
Output: Updated Inventory entity, including InventoryID, ProductID, Quantity, Location, and LastUpdated. A logged inventory management event is recorded in the system log.   
  
1.15 Add Product Function   
Function ID: FR-15   
Description: The system allows an administrator to add a new product, including product name, description, price, category, and initial inventory quantity.   
Input: Product name, description, price, category ID, initial inventory quantity, and valid administrator session   
Output: A new Product entity, updated Inventory, and a logged product addition event in the system log.   
  
1.16 Update Product Details Function   
Function ID: FR-16   
Description: The system allows an administrator to update a product’s details, including name, description, price, category, and inventory quantity.   
Input: ProductID, Updated name, description, price, category ID, and inventory quantity, and valid administrator session   
Output: Updated Product and Inventory entities in the database. A logged product update event is recorded in the system log.   
  
1.17 Delete Product Function   
Function ID: FR-17   
Description: The system allows an administrator to delete a product from the database.   
Input: ProductID and valid administrator session   
Output: Deleted Product and updated Inventory records. A logged product deletion event is recorded in the system log.   
  
1.18 View Product Category Function   
Function ID: FR-18   
Description: The system allows a customer or administrator to view products within a selected category.   
Input: CategoryID and valid session   
Output: A list of Product entities associated with the selected Category. A logged category view event is recorded in the system log.   
  
1.19 Manage Product Category Function   
Function ID: FR-19   
Description: The system allows an administrator to add, update, or delete product categories.   
Input: Category name, description, and valid administrator session   
Output: Updated Category entity in the database. A logged category management event is recorded in the system log.   
  
1.20 Install Plugin Function   
Function ID: FR-20   
Description: The system allows an administrator to install a new plugin, including validating compatibility and updating the plugin status.   
Input: Plugin file or Plugin details (Name, Version, Description, APIEndpoint), and valid administrator session   
Output: A new Plugin entity with a unique PluginID and updated status to "installed." A logged plugin installation event is recorded in the system log.   
  
1.21 Configure Plugin Function   
Function ID: FR-21   
Description: The system allows an administrator to configure the settings of an installed plugin.   
Input: PluginID, Configuration key-value pairs, and valid administrator session   
Output: A new or updated PluginConfiguration entity in the database. A logged plugin configuration event is recorded in the system log.

# External Description

\*\*Chapter 2: External Interfaces\*\*  
  
This chapter outlines the external interfaces that the system must interact with in order to fulfill its functional requirements. These interfaces include user interfaces, hardware interfaces, software interfaces (including databases and third-party systems), and communication interfaces (such as email). Each interface is described in terms of its role, interaction method, and data flow.  
  
---  
  
### \*\*2.1 User Interface Output\*\*  
  
The system provides a set of user-facing interfaces for customers and administrators to perform actions such as registration, login, account management, product browsing, cart manipulation, and order processing. These interfaces are primarily web-based and may also include mobile or desktop applications depending on the deployment environment.  
  
- \*\*User Login/Registration Screen\*\*:   
 A web form where users enter their email and password to log in or provide personal details to register.   
 - \*Input\*: Email, password, name, preferred category (optional)   
 - \*Output\*: Confirmation message, session initiation, or redirect to home page.  
  
- \*\*Customer Account Settings Page\*\*:   
 A web interface allowing logged-in users to update their personal information such as name, address, phone, and email.   
 - \*Input\*: Updated fields and valid session   
 - \*Output\*: Updated profile and confirmation message.  
  
- \*\*Shopping Cart Summary Page\*\*:   
 Displays the current contents of the shopping cart including product names, quantities, prices, and total cost.   
 - \*Input\*: CustomerID and valid session   
 - \*Output\*: List of products in the cart, estimated shipping cost, and final total amount.  
  
- \*\*Checkout Page\*\*:   
 Allows users to enter shipping and payment information to complete an order.   
 - \*Input\*: Shipping address, payment details (card number, expiry, CVV, amount)   
 - \*Output\*: Order confirmation and redirection to order summary page.  
  
- \*\*Order History View Page\*\*:   
 Displays a list of past orders for a customer or administrator.   
 - \*Input\*: CustomerID and valid session   
 - \*Output\*: List of purchase history records with order details.  
  
- \*\*Product Management Dashboard (Admin)\*\*:   
 Provides tools for adding, updating, and deleting products, managing inventory, and handling categories.   
 - \*Input\*: Product details, inventory quantity, category details   
 - \*Output\*: Updated product records and confirmation messages.  
  
- \*\*Plugin Management Interface (Admin)\*\*:   
 Allows administrators to install and configure plugins.   
 - \*Input\*: Plugin file or details, configuration settings   
 - \*Output\*: Installed plugin status and configuration updates.  
  
---  
  
### \*\*2.2 Hardware Interface Output\*\*  
  
The system currently does not require direct interaction with any external hardware devices. However, if future enhancements include integration with IoT devices or point-of-sale (POS) terminals, the following hardware interfaces may be considered:  
  
- \*\*Point-of-Sale (POS) Terminal Integration\*\*:   
 If implemented, the system will communicate via standard POS protocols (e.g., PCI DSS-compliant APIs) to process payments directly from physical terminals.   
 - \*Input\*: Payment transaction data (amount, card type, authorization code)   
 - \*Output\*: Transaction confirmation and order creation in the system.  
  
- \*\*Barcode Scanner Integration\*\*:   
 For inventory management, the system may support barcode scanning through USB or Bluetooth-connected scanners.   
 - \*Input\*: Scanned product ID   
 - \*Output\*: Automatic product lookup and inventory adjustment.  
  
\*Note: These interfaces are optional and depend on future feature development.\*  
  
---  
  
### \*\*2.3 Software Interface Output\*\*  
  
The system interacts with several internal and external software components, including databases, authentication services, and third-party APIs.  
  
#### \*\*2.3.1 Database Interfaces\*\*  
  
- \*\*UserAccount Table\*\*:   
 Stores user credentials and session-related data.   
 - \*Inputs\*: Name, Email, Password, SessionToken   
 - \*Outputs\*: UserAccount entity with unique AccountID  
  
- \*\*Customer Table\*\*:   
 Contains personal and contact information for registered customers.   
 - \*Inputs\*: Name, Email, Address, Phone   
 - \*Outputs\*: Customer entity with unique CustomerID  
  
- \*\*Product Table\*\*:   
 Stores product details such as name, description, price, and category.   
 - \*Inputs\*: Name, Description, Price, CategoryID   
 - \*Outputs\*: Product entity with unique ProductID  
  
- \*\*Inventory Table\*\*:   
 Tracks stock levels and location for each product.   
 - \*Inputs\*: ProductID, Quantity, Location   
 - \*Outputs\*: Inventory entity with unique InventoryID  
  
- \*\*Order Table\*\*:   
 Records all customer orders, including status, total cost, and shipping details.   
 - \*Inputs\*: CustomerID, ShoppingCartID, ShippingAddress, TotalAmount   
 - \*Outputs\*: Order entity with unique OrderID  
  
- \*\*PurchaseHistory Table\*\*:   
 Logs historical purchases for audit and reporting purposes.   
 - \*Inputs\*: CustomerID, ProductID, PurchaseDate, Quantity, TotalCost   
 - \*Outputs\*: PurchaseHistory entity with unique HistoryID  
  
- \*\*ShoppingCart Table\*\*:   
 Manages items added by customers before checkout.   
 - \*Inputs\*: ProductID, Quantity, CustomerID   
 - \*Outputs\*: ShoppingCart entity with unique CartID  
  
- \*\*Category Table\*\*:   
 Defines product categories and allows filtering of products.   
 - \*Inputs\*: CategoryName, Description   
 - \*Outputs\*: Category entity with unique CategoryID  
  
- \*\*Plugin Table\*\*:   
 Tracks installed plugins and their configurations.   
 - \*Inputs\*: PluginName, Version, Description, APIEndpoint   
 - \*Outputs\*: Plugin entity with unique PluginID  
  
- \*\*PluginConfiguration Table\*\*:   
 Stores configuration settings for each plugin.   
 - \*Inputs\*: PluginID, ConfigurationKey, ConfigurationValue   
 - \*Outputs\*: PluginConfiguration entity with unique ConfigurationID  
  
- \*\*EmailLog Table\*\*:   
 Logs all outgoing emails sent by the system (e.g., confirmation, order status).   
 - \*Inputs\*: EmailContent, SentDate, RecipientEmail, OrderID   
 - \*Outputs\*: EmailLog entity with unique EmailID  
  
#### \*\*2.3.2 Third-Party Software Interfaces\*\*  
  
- \*\*Email Service Provider (ESP) API\*\*:   
 Used to send confirmation and order-related emails to customers.   
 - \*Input\*: RecipientEmail, Subject, Content, OrderID   
 - \*Output\*: Email sent successfully or error response  
  
- \*\*Payment Gateway API (e.g., Stripe, PayPal)\*\*:   
 Processes credit card payments during checkout.   
 - \*Input\*: CardNumber, ExpiryDate, CVV, Amount   
 - \*Output\*: Payment success/failure status, transaction ID  
  
- \*\*Authentication Service (e.g., OAuth2, JWT)\*\*:   
 Validates user sessions and authenticates login requests.   
 - \*Input\*: Email, Password, Token   
 - \*Output\*: Authentication result (success/failure), session token  
  
---  
  
### \*\*2.4 Communication Interface Output\*\*  
  
The system communicates with external entities using various network-based methods, including email, API calls, and logging events.  
  
#### \*\*2.4.1 Email Communication Interface\*\*  
  
- \*\*Order Confirmation Email\*\*:   
 Sent to the customer after successful checkout and payment processing.   
 - \*Input\*: OrderID, CustomerEmail, OrderDetails (items, total, delivery date)   
 - \*Output\*: Email message with content, sent date, and associated EmailID in the EmailLog table.  
  
- \*\*Account Registration Confirmation Email\*\*:   
 Sent upon successful user registration.   
 - \*Input\*: CustomerEmail, VerificationLink   
 - \*Output\*: Email message with verification link and timestamp.  
  
#### \*\*2.4.2 Logging Interface\*\*  
  
- \*\*System Log Interface\*\*:   
 All user actions (login, logout, cart modification, order creation, etc.) are logged into a centralized system log for auditing and monitoring.   
 - \*Input\*: EventType (e.g., "Login", "Logout", "Order Created"), UserID, Timestamp, Details   
 - \*Output\*: Log entry stored in a database with unique LogID  
  
#### \*\*2.4.3 API Communication Interfaces\*\*  
  
- \*\*External Plugin API Endpoint\*\*:   
 When a plugin is installed, it may expose an API endpoint for integration with the system.   
 - \*Input\*: PluginID, RequestType (GET, POST, PUT, DELETE), Payload   
 - \*Output\*: Plugin response data or error message  
  
- \*\*Third-Party Product API (Optional)\*\*:   
 If integrated, used to fetch additional product details from external marketplaces or suppliers.   
 - \*Input\*: ProductSearchQuery, CategoryFilter   
 - \*Output\*: List of products from external source  
  
---  
  
### \*\*Summary of External Interfaces Covered\*\*  
  
| Interface Type | Description |  
|----------------|-------------|  
| \*\*User Interface\*\* | Web-based forms and dashboards for user interaction, including login, registration, account settings, product browsing, and order management. |  
| \*\*Database Interfaces\*\* | Internal tables for storing user accounts, customer data, products, orders, inventory, logs, and plugin configurations. |  
| \*\*Software Interfaces\*\* | Integration with third-party authentication, payment gateways, and email service providers. |  
| \*\*Communication Interfaces\*\* | Network-based interactions such as sending emails, logging system events, and calling external plugin APIs. |  
  
All external data sources referenced in the functional requirements have been accounted for and clearly defined in this section.