# Functional Requirement

# 1. Functional Requirements  
  
## 1.1 Customer Registration Function   
\*\*Function ID\*\*: FR-01   
\*\*Description\*\*: Customers can create new accounts by providing personal information and receiving a confirmation email. The system must validate the email format, check for uniqueness, and ensure password strength.   
\*\*Input\*\*:   
- Name (First and Last)   
- Email   
- Password   
- Shipping Address   
- Preferred Category ID (optional)   
\*\*Output\*\*:   
- Created Customer entity in the database   
- Email confirmation sent to the customer   
- Updated inventory/category associations if applicable   
  
---  
  
## 1.2 Customer Login Function   
\*\*Function ID\*\*: FR-02   
\*\*Description\*\*: Registered and verified customers can log in using their email and password. The system must validate the credentials and initiate a session.   
\*\*Input\*\*:   
- Email   
- Password   
\*\*Output\*\*:   
- Active session for the customer   
- Display of the customer's homepage or dashboard   
- Error messages if login fails   
  
---  
  
## 1.3 View Product Details Function   
\*\*Function ID\*\*: FR-03   
\*\*Description\*\*: Customers can view detailed information of a selected product, such as name, description, price, and availability. The system must update the browsing history and display the data correctly.   
\*\*Input\*\*:   
- Product ID   
- Customer ID (optional for browsing history)   
\*\*Output\*\*:   
- Displayed product details (name, description, price, availability, image)   
- Updated customer activity log   
  
---  
  
## 1.4 Add Product to Cart Function   
\*\*Function ID\*\*: FR-04   
\*\*Description\*\*: A logged-in customer can add a product to their shopping cart. The system must check the product’s availability and update the cart accordingly.   
\*\*Input\*\*:   
- Product ID   
- Quantity   
- Customer ID   
\*\*Output\*\*:   
- Updated CartItem in the database   
- Updated cart display with new item and total   
- Error message if product is out of stock or unavailable   
  
---  
  
## 1.5 Update Shopping Cart Function   
\*\*Function ID\*\*: FR-05   
\*\*Description\*\*: A customer can modify the contents of their shopping cart, such as changing the quantity or removing items. The system must validate stock availability and update the cart and total accordingly.   
\*\*Input\*\*:   
- CartItem ID(s)   
- New quantity or removal request   
- Customer ID   
\*\*Output\*\*:   
- Updated CartItem or removed CartItem from the database   
- Updated cart display with new contents and total   
- Error message if stock is insufficient or system fails to update   
  
---  
  
## 1.6 Remove Product from Cart Function   
\*\*Function ID\*\*: FR-06   
\*\*Description\*\*: A customer can remove a selected product from their shopping cart. The system must update the cart and recalculate the total amount.   
\*\*Input\*\*:   
- CartItem ID   
- Customer ID   
\*\*Output\*\*:   
- Removed CartItem from the database   
- Updated cart display and total   
- Notification if cart is empty   
  
---  
  
## 1.7 Checkout Function   
\*\*Function ID\*\*: FR-07   
\*\*Description\*\*: A customer can proceed to checkout and complete the purchase process. The system must calculate the total amount, process payment, create an order, and send a confirmation email.   
\*\*Input\*\*:   
- CartItem(s)   
- Shipping address   
- Shipping method   
- Payment method and details   
- Customer ID   
\*\*Output\*\*:   
- Created Order and OrderDetail entities in the database   
- Updated Inventory with reduced stock   
- Sent email confirmation to the customer   
- Cleared shopping cart   
  
---  
  
## 1.8 Confirm Order Function   
\*\*Function ID\*\*: FR-08   
\*\*Description\*\*: A customer can confirm an order after providing all necessary checkout information. The system must process the payment, finalize the order, and update inventory accordingly.   
\*\*Input\*\*:   
- Order summary (items, shipping address, payment method)   
- Customer ID   
\*\*Output\*\*:   
- Confirmed Order entity   
- Updated Payment status and details   
- Updated Inventory   
- Sent Email Confirmation   
  
---  
  
## 1.9 Receive Order Confirmation Email Function   
\*\*Function ID\*\*: FR-09   
\*\*Description\*\*: The system must send an email confirmation to the customer after an order is successfully processed. The email must include the order summary, payment details, and shipping information.   
\*\*Input\*\*:   
- Order ID   
- Customer Email   
\*\*Output\*\*:   
- Sent confirmation email to the customer   
- Log entry indicating successful email delivery   
- Order status updated to "confirmed"   
  
---  
  
## 1.10 Administrator Login Function   
\*\*Function ID\*\*: FR-10   
\*\*Description\*\*: Administrators can log in using their username and password. The system must validate credentials and initiate a session.   
\*\*Input\*\*:   
- Username   
- Password   
\*\*Output\*\*:   
- Active session for the administrator   
- Display of the admin dashboard or control panel   
- Error messages if login fails   
  
---  
  
## 1.11 Manage Inventory Function   
\*\*Function ID\*\*: FR-11   
\*\*Description\*\*: Administrators can add, update, or delete inventory records. The system must validate the changes and update associated product data and logs.   
\*\*Input\*\*:   
- Product ID   
- Inventory quantity   
- Inventory status (e.g., reorder level)   
- Administrator ID   
\*\*Output\*\*:   
- Updated Inventory entity in the database   
- Updated product availability status   
- Log entry for the inventory change   
- Confirmation message for the administrator   
  
---  
  
## 1.12 Update Product Information Function   
\*\*Function ID\*\*: FR-12   
\*\*Description\*\*: Administrators can update product details such as name, price, description, and category. The system must validate the changes and update related entities.   
\*\*Input\*\*:   
- Product ID   
- Updated fields (e.g., Name, Price, Description, CategoryID)   
- Administrator ID   
\*\*Output\*\*:   
- Updated Product entity in the database   
- Updated Inventory and Category associations   
- Log entry for the product update   
- Confirmation message for the administrator   
  
---  
  
## 1.13 Manage Categories Function   
\*\*Function ID\*\*: FR-13   
\*\*Description\*\*: Administrators can add, update, or delete product categories. The system must ensure consistency with associated products and handle plugin restrictions if applicable.   
\*\*Input\*\*:   
- Category ID (for update or delete)   
- New category name and description (for add)   
- Administrator ID   
\*\*Output\*\*:   
- Updated Category entity in the database   
- Updated product listings and search filters   
- Log entry for the category change   
- Confirmation message for the administrator   
  
---  
  
## 1.14 Create Category Function   
\*\*Function ID\*\*: FR-14   
\*\*Description\*\*: Administrators can create a new category for organizing products. The system must validate the category name and description, and apply any plugin rules.   
\*\*Input\*\*:   
- Category name   
- Category description   
- Administrator ID   
\*\*Output\*\*:   
- Created Category entity in the database   
- Updated category list in the system   
- Log entry for the category creation   
- Confirmation message for the administrator   
  
---  
  
## 1.15 Delete Category Function   
\*\*Function ID\*\*: FR-15   
\*\*Description\*\*: Administrators can delete an existing category. If the category contains products, the system must prompt for reassignment or uncategorization.   
\*\*Input\*\*:   
- Category ID   
- Administrator ID   
- Optional: New Category ID for reassignment   
\*\*Output\*\*:   
- Deleted Category entity   
- Updated product records to reflect new category or uncategorized status   
- Log entry for the category deletion   
- Confirmation message for the administrator   
  
---  
  
## 1.16 Plugin Installation Function   
\*\*Function ID\*\*: FR-16   
\*\*Description\*\*: Administrators can install new plugins to extend the system's functionality. The system must verify plugin compatibility and dependencies before installation.   
\*\*Input\*\*:   
- Plugin name and version   
- Plugin file or source (e.g., ZIP or URL)   
- Administrator ID   
\*\*Output\*\*:   
- Installed Plugin entity in the database   
- Updated plugin list in the system   
- Log entry for the plugin installation   
- Confirmation message for the administrator   
  
---  
  
## 1.17 Plugin Configuration Function   
\*\*Function ID\*\*: FR-17   
\*\*Description\*\*: Administrators can configure installed plugins to suit their needs. The system must validate configuration inputs and apply them to the plugin.   
\*\*Input\*\*:   
- Plugin ID   
- Configuration parameters (e.g., API keys, thresholds, settings)   
- Administrator ID   
\*\*Output\*\*:   
- Updated Plugin Configuration entity in the database   
- Applied configuration settings to the plugin   
- Log entry for the plugin configuration   
- Confirmation message for the administrator   
  
---  
  
## 1.18 View Payment Information Function   
\*\*Function ID\*\*: FR-18   
\*\*Description\*\*: Customers can view their payment information for past orders. The system must retrieve and display the payment method, status, amount, and date.   
\*\*Input\*\*:   
- Order ID   
- Customer ID   
\*\*Output\*\*:   
- Displayed Payment details (method, amount, status, date)   
- Option to request a payment confirmation email   
- Error message if payment data is missing or invalid   
  
---  
  
## 1.19 Update Order Detail Function   
\*\*Function ID\*\*: FR-19   
\*\*Description\*\*: Administrators can update the details of an order, such as product quantity or price. The system must validate the changes, adjust inventory if necessary, and log the update.   
\*\*Input\*\*:   
- OrderDetail ID   
- Updated product quantity or price   
- Administrator ID   
\*\*Output\*\*:   
- Updated OrderDetail entity in the database   
- Adjusted Inventory if product quantity is changed   
- Log entry for the order detail update   
- Confirmation message for the administrator   
  
---  
  
Each function is derived directly from the provided use cases and data model, ensuring feasibility, consistency, and completeness. All inputs and outputs are defined within the system context, and no additional or invented functions are included.

# External Description

# 2. External Interfaces  
  
This chapter describes the external interfaces of the system, including user interfaces, hardware interfaces, software interfaces, and communication interfaces. These interfaces define how the system interacts with users, hardware, other software systems, and external communication channels. The descriptions are based on the functional requirements and ensure that all external data sources and interactions are clearly defined.  
  
## 2.1 User Interface  
  
The system interacts with its users through a graphical user interface (GUI) that is accessible via a web browser or mobile application. The following are the key user interface components and their roles:  
  
- \*\*Customer Registration Screen\*\*:   
 - \*\*Role\*\*: Allows new customers to provide personal information and create an account.   
 - \*\*Description\*\*: Includes input fields for name, email, password, shipping address, and optional preferred category. The system validates the email format, password strength, and checks for email uniqueness.   
 - \*\*Interaction Method\*\*: Form submission with validation feedback and confirmation message displayed after successful registration.  
  
- \*\*Customer Login Screen\*\*:   
 - \*\*Role\*\*: Enables registered and verified customers to log in using their email and password.   
 - \*\*Description\*\*: Contains input fields for email and password, and a login button. The system validates the credentials and initiates a session upon successful login.   
 - \*\*Interaction Method\*\*: Form submission with error messages displayed for invalid or incorrect credentials.  
  
- \*\*Product Details Page\*\*:   
 - \*\*Role\*\*: Displays detailed information about a product, such as name, description, price, availability, and image.   
 - \*\*Description\*\*: The page dynamically loads product data based on the provided Product ID. It also optionally updates the customer’s browsing history if a Customer ID is provided.   
 - \*\*Interaction Method\*\*: Product ID is passed through the URL or a query parameter. The system updates the customer’s activity log upon viewing.  
  
- \*\*Shopping Cart Page\*\*:   
 - \*\*Role\*\*: Displays the current items in a customer’s shopping cart and allows modification.   
 - \*\*Description\*\*: The page includes functionality to add, update, and remove items. The system validates stock availability and updates the cart total accordingly.   
 - \*\*Interaction Method\*\*: Customer ID is used to fetch cart data. CartItem IDs, new quantities, and removal requests are submitted via form or API calls.  
  
- \*\*Checkout Page\*\*:   
 - \*\*Role\*\*: Allows a customer to complete the purchase process by entering shipping and payment information.   
 - \*\*Description\*\*: The page includes fields for shipping address, shipping method, and payment method. The system calculates the total amount, processes payment, and creates an order upon submission.   
 - \*\*Interaction Method\*\*: CartItem(s), shipping and payment details, and Customer ID are submitted to finalize the order.  
  
- \*\*Order Confirmation Page\*\*:   
 - \*\*Role\*\*: Displays the final order summary and confirms the order after payment is processed.   
 - \*\*Description\*\*: Shows the items, total amount, shipping address, and payment method. The system updates the order status and clears the shopping cart after confirmation.   
 - \*\*Interaction Method\*\*: Order summary and Customer ID are used to confirm the order. A confirmation message is displayed, and a confirmation email is sent.  
  
- \*\*Admin Login Screen\*\*:   
 - \*\*Role\*\*: Enables administrators to log in using a username and password.   
 - \*\*Description\*\*: Contains fields for username and password, and a login button. The system validates the credentials and displays the admin dashboard or control panel.   
 - \*\*Interaction Method\*\*: Form submission with error messages for invalid or incorrect login attempts.  
  
- \*\*Admin Dashboard / Control Panel\*\*:   
 - \*\*Role\*\*: Provides administrators with tools to manage inventory, product information, categories, and plugins.   
 - \*\*Description\*\*: Includes sections for adding/updating/deleting inventory, products, and categories. Also includes a plugin management section for installation and configuration.   
 - \*\*Interaction Method\*\*: Various forms and controls are used to submit changes, which are validated and applied to the database.  
  
- \*\*Payment Information View\*\*:   
 - \*\*Role\*\*: Displays the payment details for a specific order.   
 - \*\*Description\*\*: Shows the payment method, amount, status, and date. Customers can also request a payment confirmation email.   
 - \*\*Interaction Method\*\*: Order ID and Customer ID are used to fetch and display the relevant payment data.  
  
- \*\*Order Detail Update Page (Admin)\*\*:   
 - \*\*Role\*\*: Enables administrators to update the details of an order, such as product quantity or price.   
 - \*\*Description\*\*: Displays the current order details and allows administrators to modify them. The system validates the changes and adjusts inventory if necessary.   
 - \*\*Interaction Method\*\*: OrderDetail ID and Administrator ID are used to fetch and update the data. A confirmation message is displayed after the update.  
  
## 2.2 Hardware Interface  
  
There are no direct hardware interfaces required for the system as described in the functional requirements. The system is designed to be accessed through standard web browsers and mobile applications, and does not require interaction with specific hardware devices. Therefore, no hardware interface specifications are necessary in this section.  
  
## 2.3 Software Interface  
  
The system interacts with external software components and databases to support its functionality. The following are the key software interfaces:  
  
- \*\*Database System (e.g., MySQL, PostgreSQL)\*\*   
 - \*\*Role\*\*: Stores and retrieves data related to customers, products, inventory, categories, orders, and plugins.   
 - \*\*Description\*\*: The system uses a relational database to manage persistent data. The database schema includes entities such as Customer, Product, Inventory, Category, Order, OrderDetail, and Plugin.   
 - \*\*Interaction Method\*\*: The system uses SQL queries or an ORM (Object-Relational Mapping) to interact with the database. Inputs such as Customer ID, Product ID, OrderDetail ID, etc., are used to fetch and update records. Outputs include created or updated entities, such as Customer, Product, Order, etc.  
  
- \*\*Email Service (e.g., SMTP Server, SendGrid, Amazon SES)\*\*   
 - \*\*Role\*\*: Sends confirmation emails to customers after registration or order confirmation.   
 - \*\*Description\*\*: The system requires an external email service to send emails. The email includes the order summary, payment details, and shipping information.   
 - \*\*Interaction Method\*\*: The system uses an API or SMTP protocol to send emails. Inputs include the recipient's email address, the email content, and the order details. Outputs include a log entry for successful email delivery and an updated order status.  
  
- \*\*Payment Gateway (e.g., Stripe, PayPal, Square)\*\*   
 - \*\*Role\*\*: Processes customer payments during the checkout and order confirmation processes.   
 - \*\*Description\*\*: The system integrates with a payment gateway to handle secure payment transactions. The payment gateway verifies the payment method, processes the payment, and returns a status.   
 - \*\*Interaction Method\*\*: The system sends payment details (e.g., amount, payment method, customer ID) to the payment gateway via API. The payment gateway responds with a payment status and transaction details. Outputs include updated payment status and confirmation messages.  
  
- \*\*Plugin Management Module\*\*   
 - \*\*Role\*\*: Allows administrators to install, configure, and manage plugins to extend system functionality.   
 - \*\*Description\*\*: The system provides a plugin interface that includes validation of plugin compatibility and dependencies. Configuration parameters are applied to the plugin and stored in the database.   
 - \*\*Interaction Method\*\*: The system uses a plugin API to install and configure plugins. Inputs include the plugin file, configuration parameters, and Administrator ID. Outputs include the installed plugin entity, updated plugin list, and confirmation messages.  
  
- \*\*Inventory Management Module\*\*   
 - \*\*Role\*\*: Maintains and updates product inventory levels and statuses.   
 - \*\*Description\*\*: The system tracks the availability and quantity of each product. Inventory is updated automatically when a product is added to a cart, updated, or deleted.   
 - \*\*Interaction Method\*\*: The system uses internal functions or triggers to update inventory records. Inputs include Product ID, Inventory quantity, and Inventory status. Outputs include updated Inventory entity and product availability status.  
  
## 2.4 Communication Interface  
  
The system communicates with external entities using standard communication protocols and services. The following are the key communication interfaces:  
  
- \*\*Email Communication\*\*   
 - \*\*Role\*\*: Sends confirmation emails to customers for registration and order processing.   
 - \*\*Description\*\*: The system uses an email service to send messages to customer email addresses. The email contains the order summary, payment details, and shipping information.   
 - \*\*Interaction Method\*\*: The system sends an email via an SMTP server or third-party email API. Inputs include the customer's email address and order details. Outputs include a log entry indicating successful delivery and an updated order status.  
  
- \*\*Web Browsing / API Communication\*\*   
 - \*\*Role\*\*: Supports customer and administrator access to the system through web browsers or API integrations.   
 - \*\*Description\*\*: The system is accessible via HTTP/HTTPS protocols and supports RESTful API for integration with other systems or services.   
 - \*\*Interaction Method\*\*: The system uses HTTP methods (GET, POST, PUT, DELETE) to interact with clients. Inputs are received via API endpoints or web forms. Outputs are returned in JSON or HTML format.  
  
- \*\*Payment Gateway Communication\*\*   
 - \*\*Role\*\*: Processes customer payments securely during checkout.   
 - \*\*Description\*\*: The system communicates with a payment gateway to verify and process transactions.   
 - \*\*Interaction Method\*\*: The system sends payment requests via HTTPS to the payment gateway API. Inputs include the customer's payment method and amount. Outputs include a payment status and transaction confirmation.  
  
- \*\*Plugin Communication Interface\*\*   
 - \*\*Role\*\*: Supports plugin installation and configuration by validating plugin compatibility and applying configuration parameters.   
 - \*\*Description\*\*: The system communicates with plugins to load and execute their functionality.   
 - \*\*Interaction Method\*\*: The system uses a plugin API to install and configure plugins. Inputs include the plugin file and configuration parameters. Outputs include updated plugin status and confirmation messages.  
  
---  
  
This chapter provides a comprehensive overview of the external interfaces required for the system to function correctly. Each interface is described in terms of its role, description, and interaction method to ensure clarity for developers and stakeholders.