项目文档

# Functional Requirement

# \*\*GAMMA-J Web Store\*\*   
## \*\*Chapter 1: Functional Requirements\*\*  
  
---  
  
### \*\*1.1 Create Customer Account Function\*\*   
\*\*Function ID\*\*: FR-01   
\*\*Description\*\*: The system allows the Administrator to create a new Customer account with valid information. The system also sends a confirmation email to the customer.   
\*\*Input\*\*:   
- Name (string)   
- Email (string, unique)   
- Password (string)   
- Address (string)   
- PhoneNumber (string)   
\*\*Output\*\*:   
- New Customer record in the Customer data entity   
- Email Notification to the customer confirming the account creation   
  
---  
  
### \*\*1.2 Login as Customer Function\*\*   
\*\*Function ID\*\*: FR-02   
\*\*Description\*\*: The system authenticates a Customer using their email and password. If successful, the system initiates a session and redirects the Customer to the homepage or dashboard.   
\*\*Input\*\*:   
- Email (string)   
- Password (string)   
\*\*Output\*\*:   
- Active LoginSession record   
- Redirect to homepage or dashboard   
- Welcome message displayed   
  
---  
  
### \*\*1.3 Logout as Customer Function\*\*   
\*\*Function ID\*\*: FR-03   
\*\*Description\*\*: The system ends the Customer's session and updates their session status to "logged out".   
\*\*Input\*\*:   
- Customer session ID   
\*\*Output\*\*:   
- Updated LoginSession status to "logged out"   
- Redirect to login page or homepage   
- Confirmation message displayed   
  
---  
  
### \*\*1.4 View Product Information Function\*\*   
\*\*Function ID\*\*: FR-04   
\*\*Description\*\*: The system retrieves and displays detailed information about a selected Product, including its name, description, price, and inventory status.   
\*\*Input\*\*:   
- Product ID (string) or search query (string)   
- Active Customer session ID   
\*\*Output\*\*:   
- Displayed product details (name, description, price, inventory status)   
- Log entry in the system for analytics   
  
---  
  
### \*\*1.5 Search Product Function\*\*   
\*\*Function ID\*\*: FR-05   
\*\*Description\*\*: The system allows the Customer to search for products using a query and displays a list of matching products with key details.   
\*\*Input\*\*:   
- Search query (string)   
- Active Customer session ID   
\*\*Output\*\*:   
- List of matching Product records with name, description, price, and inventory status   
- Log entry for the search activity   
  
---  
  
### \*\*1.6 Add Product to Inventory Function\*\*   
\*\*Function ID\*\*: FR-06   
\*\*Description\*\*: The system allows the Administrator to add a new product to the Inventory and updates the Product data entity accordingly.   
\*\*Input\*\*:   
- Product name (string)   
- Price (numeric)   
- Description (string)   
- Quantity (numeric)   
- Category ID (string)   
- Active Administrator session ID   
\*\*Output\*\*:   
- New Product record in the Product data entity   
- Updated Inventory record   
- Confirmation message displayed   
  
---  
  
### \*\*1.7 Update Product Details Function\*\*   
\*\*Function ID\*\*: FR-07   
\*\*Description\*\*: The system allows the Administrator to modify product details such as name, price, description, or category and updates both the Product and Inventory data entities.   
\*\*Input\*\*:   
- Product ID (string)   
- Updated product details (name, price, description, category ID, quantity)   
- Active Administrator session ID   
\*\*Output\*\*:   
- Updated Product record in the Product data entity   
- Updated Inventory record if stock changes   
- Confirmation message displayed   
  
---  
  
### \*\*1.8 Delete Product Function\*\*   
\*\*Function ID\*\*: FR-08   
\*\*Description\*\*: The system allows the Administrator to delete a Product record and removes the reference in the Inventory.   
\*\*Input\*\*:   
- Product ID (string)   
- Active Administrator session ID   
\*\*Output\*\*:   
- Deleted Product record   
- Updated Inventory record   
- Confirmation message displayed   
  
---  
  
### \*\*1.9 Create Order Function\*\*   
\*\*Function ID\*\*: FR-09   
\*\*Description\*\*: The system allows the Customer to create an Order after confirming product selection, quantity, and valid Payment Information.   
\*\*Input\*\*:   
- List of selected Product IDs and quantities   
- Payment Information (card number, expiry date, billing address)   
- Active Customer session ID   
\*\*Output\*\*:   
- New Order record in the Order data entity   
- Updated Inventory stock levels   
- Email Notification to the Customer   
- Confirmation message displayed   
  
---  
  
### \*\*1.10 View Order Details Function\*\*   
\*\*Function ID\*\*: FR-10   
\*\*Description\*\*: The system retrieves and displays detailed information about a specific Order, including products, quantities, total price, and status.   
\*\*Input\*\*:   
- Order ID (string)   
- Active Customer or Administrator session ID   
\*\*Output\*\*:   
- Displayed order details (order ID, product list, quantities, total price, status)   
- Log entry for the order view activity   
  
---  
  
### \*\*1.11 Update Order Status Function\*\*   
\*\*Function ID\*\*: FR-11   
\*\*Description\*\*: The system allows the Administrator to update the status of an Order (e.g., Shipped, Cancelled, Processing) and adjusts the Inventory accordingly.   
\*\*Input\*\*:   
- Order ID (string)   
- New status (string)   
- Active Administrator session ID   
\*\*Output\*\*:   
- Updated Order record with new status   
- Updated Inventory stock if status change affects availability   
- Email Notification to the Customer   
- Log entry for the status update   
  
---  
  
### \*\*1.12 Delete Order Function\*\*   
\*\*Function ID\*\*: FR-12   
\*\*Description\*\*: The system allows the Customer or Administrator to delete an Order and restores the product stock in the Inventory.   
\*\*Input\*\*:   
- Order ID (string)   
- Active Customer or Administrator session ID   
\*\*Output\*\*:   
- Deleted Order record   
- Updated Inventory stock levels   
- Email Notification to the Customer   
- Log entry for the deletion activity   
  
---  
  
### \*\*1.13 View Inventory Levels Function\*\*   
\*\*Function ID\*\*: FR-13   
\*\*Description\*\*: The system allows the Administrator to view current stock levels of all or selected products in the Inventory.   
\*\*Input\*\*:   
- Active Administrator session ID   
- Optional filter criteria (category, product name, etc.)   
\*\*Output\*\*:   
- Displayed list of products with current stock levels and status (in stock, low stock, out of stock)   
- Log entry for the inventory view activity   
  
---  
  
### \*\*1.14 Manage Inventory Stock Function\*\*   
\*\*Function ID\*\*: FR-14   
\*\*Description\*\*: The system allows the Administrator to update the stock level of a specific product in the Inventory.   
\*\*Input\*\*:   
- Product ID (string)   
- New stock quantity (numeric)   
- Active Administrator session ID   
\*\*Output\*\*:   
- Updated Inventory record   
- Confirmation message displayed   
- Log entry for the stock update activity   
  
---  
  
### \*\*1.15 Create Payment Information Function\*\*   
\*\*Function ID\*\*: FR-15   
\*\*Description\*\*: The system allows the Customer or Administrator to create new Payment Information for the user.   
\*\*Input\*\*:   
- Card number (string)   
- Expiry date (string)   
- Billing address (string)   
- Active Customer or Administrator session ID   
\*\*Output\*\*:   
- New Payment Information record   
- Email Notification to the Customer   
- Log entry for the creation activity   
  
---  
  
### \*\*1.16 View Payment Information Function\*\*   
\*\*Function ID\*\*: FR-16   
\*\*Description\*\*: The system retrieves and displays stored Payment Information for the Customer or Administrator.   
\*\*Input\*\*:   
- Active Customer or Administrator session ID   
\*\*Output\*\*:   
- Displayed Payment Information (card type, masked card number, expiry date, billing address)   
- Log entry for the view activity   
  
---  
  
### \*\*1.17 Update Payment Method Function\*\*   
\*\*Function ID\*\*: FR-17   
\*\*Description\*\*: The system allows the Customer or Administrator to update existing Payment Information with new or modified data.   
\*\*Input\*\*:   
- Payment ID (string)   
- Updated card number (string)   
- Updated expiry date (string)   
- Updated billing address (string)   
- Active Customer or Administrator session ID   
\*\*Output\*\*:   
- Updated Payment Information record   
- Email Notification to the Customer   
- Log entry for the update activity   
  
---  
  
### \*\*1.18 Delete Payment Method Function\*\*   
\*\*Function ID\*\*: FR-18   
\*\*Description\*\*: The system allows the Customer or Administrator to delete a Payment Information record from the system.   
\*\*Input\*\*:   
- Payment ID (string)   
- Active Customer or Administrator session ID   
\*\*Output\*\*:   
- Deleted Payment Information record   
- Email Notification to the Customer   
- Log entry for the deletion activity   
  
---  
  
### \*\*1.19 Register Administrator Function\*\*   
\*\*Function ID\*\*: FR-19   
\*\*Description\*\*: The system allows the creation of a new Administrator account with valid information and sends a confirmation email.   
\*\*Input\*\*:   
- Name (string)   
- Email (string, unique)   
- Password (string)   
- Active Administrator session ID   
\*\*Output\*\*:   
- New Administrator record in the Administrator data entity   
- Email Notification to the Administrator   
- Log entry for the registration activity   
  
---  
  
### \*\*1.20 Login as Administrator Function\*\*   
\*\*Function ID\*\*: FR-20   
\*\*Description\*\*: The system authenticates an Administrator using their email and password and initiates a session.   
\*\*Input\*\*:   
- Email (string)   
- Password (string)   
\*\*Output\*\*:   
- Active LoginSession record   
- Redirect to admin dashboard   
- Welcome message displayed   
  
---  
  
### \*\*1.21 Logout as Administrator Function\*\*   
\*\*Function ID\*\*: FR-21   
\*\*Description\*\*: The system ends the Administrator's session and updates the session status to "logged out".   
\*\*Input\*\*:   
- Administrator session ID   
\*\*Output\*\*:   
- Updated LoginSession status to "logged out"   
- Redirect to login page or homepage   
- Confirmation message displayed   
  
---  
  
### \*\*1.22 Send Email Notification Function\*\*   
\*\*Function ID\*\*: FR-22   
\*\*Description\*\*: The system sends an email notification to a Customer or Administrator based on predefined triggers such as order confirmation or account creation.   
\*\*Input\*\*:   
- Recipient ID (Customer or Administrator)   
- Message content or template   
- Notification subject (string)   
\*\*Output\*\*:   
- Email Notification record in the Email Notification data entity   
- Email sent to the recipient   
- Log entry for the notification activity   
  
---  
  
### \*\*1.23 View Notification Logs Function\*\*   
\*\*Function ID\*\*: FR-23   
\*\*Description\*\*: The system allows the Administrator to view logs of all sent Email Notifications, including delivery status and timestamps.   
\*\*Input\*\*:   
- Active Administrator session ID   
- Optional filter criteria (date, recipient, status)   
\*\*Output\*\*:   
- Displayed list of Email Notification records   
- Log entry for the access activity   
  
---  
  
### \*\*1.24 Configure Plugin Settings Function\*\*   
\*\*Function ID\*\*: FR-24   
\*\*Description\*\*: The system allows the Administrator to modify settings of an installed Plugin and applies the changes to the Plugin's functionality.   
\*\*Input\*\*:   
- Plugin ID (string)   
- Updated settings (key-value pairs or configuration parameters)   
- Active Administrator session ID   
\*\*Output\*\*:   
- Updated Plugin configuration   
- Log entry for the configuration activity   
- Plugin reloaded or restarted   
  
---  
  
### \*\*1.25 Manage Product Category Function\*\*   
\*\*Function ID\*\*: FR-25   
\*\*Description\*\*: The system allows the Administrator to add, update, or delete a Product Category and updates associated Product records.   
\*\*Input\*\*:   
- Category name (string)   
- Description (string)   
- Action (Add, Update, Delete)   
- Active Administrator session ID   
\*\*Output\*\*:   
- Updated Product Category data entity   
- Updated Product records if category is changed or deleted   
- Log entry for the activity   
- Plugin configuration updated if necessary   
  
---  
  
### \*\*1.26 Manage Product Details Function\*\*   
\*\*Function ID\*\*: FR-26   
\*\*Description\*\*: The system allows the Administrator to modify product details such as name, price, description, or category, and updates the associated Product, Inventory, and Plugin data if required.   
\*\*Input\*\*:   
- Product ID (string)   
- Updated product details (name, price, description, category ID, etc.)   
- Active Administrator session ID   
\*\*Output\*\*:   
- Updated Product record in the Product data entity   
- Updated Inventory record if stock or availability is affected   
- Log entry for the activity   
- Plugin configuration updated if necessary   
  
---  
  
以上为\*\*GAMMA-J Web Store\*\*系统功能需求章节（Chapter 1: Functional Requirements）的完整内容，严格按照用户提供的用例和数据模型编写，确保功能描述清晰、输入输出完整、可实现性高。

# External Description

# \*\*Chapter 2: External Interfaces\*\*   
## \*\*GAMMA-J Web Store\*\*   
  
---  
  
### \*\*2.1 User Interface Output\*\*   
This section defines the user interface (UI) components and interactions required for the GAMMA-J Web Store system. These interfaces are used by both \*\*Customers\*\* and \*\*Administrators\*\* to interact with the system.  
  
#### \*\*2.1.1 Customer Login Interface\*\*   
\*\*Description\*\*: A web-based login form where the Customer enters their email and password to authenticate.   
\*\*Interaction Method\*\*:   
- Input: Email and password via text fields.   
- Output: A welcome message and redirection to the homepage or dashboard upon successful login.  
  
#### \*\*2.1.2 Customer Registration Interface\*\*   
\*\*Description\*\*: A web-based form allowing the Administrator to create a new Customer account.   
\*\*Interaction Method\*\*:   
- Input: Name, Email, Password, Address, and PhoneNumber.   
- Output: Confirmation message and email sent to the customer.  
  
#### \*\*2.1.3 Customer Dashboard Interface\*\*   
\*\*Description\*\*: A web-based interface displaying order history, product search, and inventory status for the Customer.   
\*\*Interaction Method\*\*:   
- Input: Active Customer session ID.   
- Output: Display of product details, order information, and search results.  
  
#### \*\*2.1.4 Product Search Interface\*\*   
\*\*Description\*\*: A search bar and results page where the Customer can search for products.   
\*\*Interaction Method\*\*:   
- Input: Search query and active Customer session ID.   
- Output: List of matching products with name, description, price, and inventory status.  
  
#### \*\*2.1.5 Product Detail Interface\*\*   
\*\*Description\*\*: A web-based page that displays detailed information about a selected product.   
\*\*Interaction Method\*\*:   
- Input: Product ID or search query and active Customer session ID.   
- Output: Product name, description, price, and inventory status displayed.  
  
#### \*\*2.1.6 Order Creation Interface\*\*   
\*\*Description\*\*: A web-based interface where the Customer can finalize an order by entering payment details and confirming selections.   
\*\*Interaction Method\*\*:   
- Input: List of Product IDs and quantities, payment information (card number, expiry, billing address), and active Customer session ID.   
- Output: Confirmation message, email notification, and redirection to the order confirmation page.  
  
#### \*\*2.1.7 Order Detail Interface\*\*   
\*\*Description\*\*: A web-based interface that displays the details of a specific order, including product list, quantities, total price, and status.   
\*\*Interaction Method\*\*:   
- Input: Order ID and active Customer/Administrator session ID.   
- Output: Order details displayed with a log entry for the view activity.  
  
#### \*\*2.1.8 Inventory Management Interface (Admin)\*\*   
\*\*Description\*\*: A web-based interface for the Administrator to manage products in the Inventory, including adding, updating, and deleting products.   
\*\*Interaction Method\*\*:   
- Input: Product details (name, price, description, quantity, category ID), Order ID, Payment ID, etc., along with an active Administrator session ID.   
- Output: Updated Inventory status, product records, and confirmation messages.  
  
#### \*\*2.1.9 Payment Management Interface\*\*   
\*\*Description\*\*: A web-based interface where the Customer or Administrator can view, add, update, or delete payment methods.   
\*\*Interaction Method\*\*:   
- Input: Card number, expiry date, billing address, or Payment ID, with an active session ID.   
- Output: Display of masked card details, confirmation messages, and email notifications.  
  
#### \*\*2.1.10 Email Notification Interface\*\*   
\*\*Description\*\*: A system-generated email interface used to send notifications to Customers and Administrators.   
\*\*Interaction Method\*\*:   
- Input: Recipient ID, message content or template, and notification subject.   
- Output: Email sent to the recipient with a log entry for the notification.  
  
#### \*\*2.1.11 Notification Logs Interface (Admin)\*\*   
\*\*Description\*\*: A web-based interface for the Administrator to view logs of all sent email notifications.   
\*\*Interaction Method\*\*:   
- Input: Active Administrator session ID and optional filters (date, recipient, status).   
- Output: Display of Email Notification records with delivery status and timestamps.  
  
#### \*\*2.1.12 Plugin Configuration Interface (Admin)\*\*   
\*\*Description\*\*: A web-based interface for the Administrator to modify Plugin settings.   
\*\*Interaction Method\*\*:   
- Input: Plugin ID, updated settings (key-value pairs), and active Administrator session ID.   
- Output: Updated Plugin configuration with a log entry and Plugin reload or restart.  
  
---  
  
### \*\*2.2 Hardware Interface Output\*\*   
There are no direct hardware interfaces involved in the GAMMA-J Web Store system. The system is a web-based application and does not require physical hardware interaction beyond standard web server and client hardware (e.g., browsers, servers, and network devices).   
  
\*\*Description\*\*:   
- The system operates on standard web servers and client devices (e.g., PCs, smartphones, tablets) that support modern web technologies.   
- No special hardware is required for the system to function.   
  
\*\*Interaction Method\*\*:   
- Standard web protocols (HTTP/HTTPS) over TCP/IP network connections.   
- Client-side rendering via browsers (HTML, CSS, JavaScript).   
  
---  
  
### \*\*2.3 Software Interface Output\*\*   
This section defines the software components and data entities that interact with the GAMMA-J Web Store system. These include internal and external software interfaces such as databases, APIs, and external tools.  
  
#### \*\*2.3.1 Customer Data Entity\*\*   
\*\*Description\*\*: A software data entity that stores information about registered Customers.   
\*\*Interaction Method\*\*:   
- Input: Name, Email, Password, Address, and PhoneNumber.   
- Output: New or updated Customer record in the database.  
  
#### \*\*2.3.2 Product Data Entity\*\*   
\*\*Description\*\*: A software data entity that stores information about available Products.   
\*\*Interaction Method\*\*:   
- Input: Product ID, name, price, description, category ID, and quantity.   
- Output: New or updated Product record in the database.  
  
#### \*\*2.3.3 Inventory Data Entity\*\*   
\*\*Description\*\*: A software data entity that maintains stock levels and availability status for Products.   
\*\*Interaction Method\*\*:   
- Input: Product ID, quantity, and status updates.   
- Output: Updated Inventory record in the database.  
  
#### \*\*2.3.4 Order Data Entity\*\*   
\*\*Description\*\*: A software data entity that records all Orders placed by Customers.   
\*\*Interaction Method\*\*:   
- Input: List of Product IDs and quantities, payment information, and Order ID.   
- Output: New or updated Order record in the database.  
  
#### \*\*2.3.5 LoginSession Data Entity\*\*   
\*\*Description\*\*: A software data entity that tracks active sessions for both Customers and Administrators.   
\*\*Interaction Method\*\*:   
- Input: Email, password, or session ID.   
- Output: Active or updated LoginSession record in the database.  
  
#### \*\*2.3.6 Payment Information Data Entity\*\*   
\*\*Description\*\*: A software data entity that stores secure payment details for Customers and Administrators.   
\*\*Interaction Method\*\*:   
- Input: Card number, expiry date, billing address, and Payment ID.   
- Output: New or updated Payment Information record in the database.  
  
#### \*\*2.3.7 Email Notification Data Entity\*\*   
\*\*Description\*\*: A software data entity that logs all email notifications sent by the system.   
\*\*Interaction Method\*\*:   
- Input: Recipient ID, message content, and subject.   
- Output: New Email Notification record in the database.  
  
#### \*\*2.3.8 Plugin Data Entity\*\*   
\*\*Description\*\*: A software data entity that stores configuration and status information for installed plugins.   
\*\*Interaction Method\*\*:   
- Input: Plugin ID, configuration parameters, and action (Add, Update, Delete).   
- Output: Updated Plugin configuration in the database.  
  
#### \*\*2.3.9 Logging System\*\*   
\*\*Description\*\*: A software interface for logging user activities and system events for analytics and auditing purposes.   
\*\*Interaction Method\*\*:   
- Input: Session ID, activity type, and relevant data (e.g., product ID, order ID, search query).   
- Output: Log entry stored in the system database.  
  
---  
  
### \*\*2.4 Communication Interface Output\*\*   
This section defines the communication interfaces used by the GAMMA-J Web Store system to interact with external systems, such as email services and web-based user sessions.  
  
#### \*\*2.4.1 Email Notification Service\*\*   
\*\*Description\*\*: An external communication interface used to send email notifications to Customers and Administrators.   
\*\*Interaction Method\*\*:   
- Input: Recipient email address, message content, and subject.   
- Output: Email sent to the recipient, with a confirmation and a log entry.  
  
#### \*\*2.4.2 Web Browser Communication\*\*   
\*\*Description\*\*: The system communicates with users through standard web browsers.   
\*\*Interaction Method\*\*:   
- Input: User input via web forms and session IDs.   
- Output: Dynamic web pages and redirects based on user actions.  
  
#### \*\*2.4.3 Session Management Communication\*\*   
\*\*Description\*\*: The system manages user sessions via HTTP cookies or token-based authentication (e.g., JWT).   
\*\*Interaction Method\*\*:   
- Input: Session ID or authentication token.   
- Output: Session status updates and redirection to appropriate pages.  
  
#### \*\*2.4.4 Plugin Communication Interface\*\*   
\*\*Description\*\*: A communication interface that allows the system to interact with plugins for extended functionality (e.g., payment gateways, inventory integrations).   
\*\*Interaction Method\*\*:   
- Input: Plugin configuration parameters and commands (e.g., reload, restart).   
- Output: Plugin behavior changes and status updates.  
  
---  
  
### \*\*Summary of External Interfaces\*\*   
The GAMMA-J Web Store system primarily interacts through \*\*web-based user interfaces\*\*, \*\*software data entities\*\*, and \*\*email notification services\*\*. There are no direct hardware interfaces. Communication with users is facilitated via \*\*web browsers\*\*, \*\*session tokens\*\*, and \*\*email services\*\*, while data is managed through \*\*databases\*\* and \*\*logging systems\*\*. Plugins are managed through a \*\*plugin configuration interface\*\*, which can also affect system behavior and data entities.  
  
All external data sources referenced in the functional requirements (e.g., Customer, Product, Inventory, Order, Payment Information, LoginSession, Email Notification, Plugin) are clearly defined in this chapter and mapped to appropriate interfaces for clarity and implementation.

# Use Case

Use Case Name: Create Customer Account   
Use Case ID: UC-01   
Actors: Customer, Administrator   
Preconditions:   
1. The system is operational and accessible to the Administrator.   
2. The Administrator is authenticated and authorized to manage customer accounts.   
3. The customer's information is available for input.   
  
Postconditions:   
1. A new customer account is successfully created in the system.   
2. The customer receives an email notification confirming the account creation.   
3. The system updates the Customer data entity with the new account details.   
  
Main Flow:   
1. The Administrator navigates to the "Create Customer Account" section of the system.   
2. The Administrator inputs the customer's details, including name, email, and contact information.   
3. The system validates the input data to ensure completeness and correctness.   
4. The system generates a unique customer ID and creates a new Customer data entity.   
5. The system sends an email notification to the customer's provided email address.   
6. The system updates the Customer data entity in the database.   
7. The system displays a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the input data is incomplete or incorrect, the system displays an error message and prompts the Administrator to correct the information.   
2. If the email notification fails to send, the system logs the error and allows the Administrator to manually retry the notification.   
3. If the customer ID generation fails, the system alerts the Administrator and suggests an alternative method for assigning the ID.  
  
Use Case Name: Login as Customer   
Use Case ID: UC-02   
Actors: Customer   
Preconditions:   
1. The system is operational and accessible to the Customer.   
2. The Customer has already created an account in the system.   
3. The Customer has a valid username and password.   
  
Postconditions:   
1. The Customer is successfully authenticated and logged into the system.   
2. The system updates the session status of the Customer.   
3. The Customer is redirected to the main dashboard or homepage.   
  
Main Flow:   
1. The Customer navigates to the login page of the system.   
2. The Customer enters their username and password.   
3. The system validates the credentials against the stored Customer data entity.   
4. The system confirms the Customer's identity and initiates a session.   
5. The system redirects the Customer to the main dashboard or homepage.   
6. The system displays a welcome message with the Customer's name.   
  
Alternative Flow:   
1. If the entered username or password is invalid, the system displays an error message and prompts the Customer to re-enter the correct credentials.   
2. If multiple login attempts fail, the system locks the account temporarily and sends an Email Notification to the Customer.   
3. If the session cannot be initiated due to technical issues, the system displays an error and suggests retrying the login or contacting support.  
  
Use Case Name: Logout as Customer   
Use Case ID: UC-03   
Actors: Customer   
Preconditions:   
1. The system is operational and accessible to the Customer.   
2. The Customer is currently logged in and has an active session.   
  
Postconditions:   
1. The Customer is successfully logged out of the system.   
2. The session status of the Customer is updated to "logged out."   
3. The Customer is redirected to the login or homepage.   
  
Main Flow:   
1. The Customer navigates to the account or session management section of the system.   
2. The Customer selects the "Logout" option.   
3. The system verifies the active session and processes the logout request.   
4. The system updates the Customer's session status to "logged out."   
5. The system redirects the Customer to the login page or homepage.   
6. The system displays a confirmation message that the Customer has been logged out.   
  
Alternative Flow:   
1. If the logout request fails due to technical issues, the system displays an error message and suggests retrying the action.   
2. If the session is already expired or invalid, the system automatically redirects the Customer to the login page without further action.  
  
Use Case Name: View Product Information   
Use Case ID: UC-04   
Actors: Customer   
Preconditions:   
1. The system is operational and accessible to the Customer.   
2. The Customer is logged in and has an active session.   
3. The Product data entity contains at least one product record.   
  
Postconditions:   
1. The Customer successfully views the detailed information of the selected product.   
2. The system displays relevant product details, such as name, description, price, and inventory status.   
3. The system logs the Customer's access to the product information for analytics.   
  
Main Flow:   
1. The Customer navigates to the product catalog or search page.   
2. The Customer selects a product or enters a search query.   
3. The system retrieves the relevant Product data entity from the database.   
4. The system displays the product details, including inventory status and pricing.   
5. The system logs the Customer's request for the product information.   
  
Alternative Flow:   
1. If no product matches the search query, the system displays a message indicating no results found.   
2. If the selected product is not available in the Product data entity, the system displays an error and suggests similar products or returning to the catalog.   
3. If the inventory status cannot be retrieved due to a system error, the system displays a warning and allows the Customer to proceed with the product details regardless.  
  
Use Case Name: Search Product   
Use Case ID: UC-05   
Actors: Customer   
Preconditions:   
1. The system is operational and accessible to the Customer.   
2. The Customer is logged in and has an active session.   
3. The Product data entity contains at least one product record.   
  
Postconditions:   
1. The Customer receives a list of products matching the search query.   
2. The system displays relevant product details, such as name, description, price, and inventory status.   
3. The system logs the search activity for analytics purposes.   
  
Main Flow:   
1. The Customer navigates to the product search page.   
2. The Customer enters a search query (e.g., product name or keyword).   
3. The system processes the search query and retrieves matching Product records.   
4. The system displays the list of matching products with key details.   
5. The system logs the Customer's search activity.   
  
Alternative Flow:   
1. If no products match the search query, the system displays a message indicating no results were found.   
2. If the search query is invalid or contains inappropriate characters, the system displays an error and prompts the Customer to refine the search.   
3. If the system cannot retrieve the product data due to technical issues, it displays an error message and suggests retrying the search or contacting support.  
  
Use Case Name: Add Product to Inventory   
Use Case ID: UC-06   
Actors: Administrator, Product, Inventory   
Preconditions:   
1. The system is operational and accessible to the Administrator.   
2. The Administrator is authenticated and authorized to manage inventory.   
3. The product information is available for input.   
  
Postconditions:   
1. A new product is successfully added to the Inventory data entity.   
2. The Product data entity is updated with the new product details.   
3. The system displays a confirmation message to the Administrator.   
  
Main Flow:   
1. The Administrator navigates to the "Add Product to Inventory" section of the system.   
2. The Administrator inputs the product details, including name, price, description, and quantity.   
3. The system validates the input data for completeness and correctness.   
4. The system generates a unique product ID and creates a new Product data entity.   
5. The system updates the Inventory data entity with the new product and its quantity.   
6. The system displays a confirmation message that the product has been added.   
  
Alternative Flow:   
1. If the input data is incomplete or incorrect, the system displays an error message and prompts the Administrator to correct the information.   
2. If the product ID generation fails, the system alerts the Administrator and suggests an alternative method for assigning the ID.   
3. If the system is unable to update the Inventory due to technical issues, it displays an error message and logs the failure for review.  
  
Use Case Name: Update Product Details   
Use Case ID: UC-07   
Actors: Administrator, Product, Inventory   
Preconditions:   
1. The system is operational and accessible to the Administrator.   
2. The Administrator is authenticated and authorized to manage product details.   
3. The product to be updated exists in the Product data entity.   
  
Postconditions:   
1. The product details in the Product data entity are successfully updated.   
2. The Inventory data entity is updated to reflect any changes in product quantity or availability.   
3. The system displays a confirmation message to the Administrator.   
  
Main Flow:   
1. The Administrator navigates to the "Manage Products" section of the system.   
2. The Administrator selects the product they wish to update from the list.   
3. The system retrieves the current Product data entity and displays the details.   
4. The Administrator modifies the product information, such as name, price, description, or quantity.   
5. The system validates the updated data for correctness and consistency.   
6. The system updates the Product data entity with the modified information.   
7. The system updates the Inventory data entity if the quantity has changed.   
8. The system displays a confirmation message that the product details have been updated.   
  
Alternative Flow:   
1. If the updated data is incomplete or incorrect, the system displays an error message and prompts the Administrator to correct the information.   
2. If the selected product does not exist in the system, the system displays an error and suggests verifying the product ID or name.   
3. If the system fails to update the Product or Inventory data entities due to technical issues, it displays an error message and logs the failure for review.  
  
Use Case Name: Delete Product   
Use Case ID: UC-08   
Actors: Administrator, Product, Inventory   
Preconditions:   
1. The system is operational and accessible to the Administrator.   
2. The Administrator is authenticated and authorized to manage products.   
3. The product to be deleted exists in the Product data entity and is associated with an Inventory record.   
  
Postconditions:   
1. The selected product is successfully deleted from the Product data entity.   
2. The Inventory data entity is updated to remove the product reference.   
3. The system displays a confirmation message to the Administrator.   
  
Main Flow:   
1. The Administrator navigates to the "Manage Products" section of the system.   
2. The Administrator selects the product they wish to delete.   
3. The system retrieves the selected Product data entity and confirms the deletion.   
4. The Administrator confirms the deletion request.   
5. The system removes the product from the Product data entity.   
6. The system updates the Inventory data entity to remove the product reference.   
7. The system displays a confirmation message that the product has been deleted.   
  
Alternative Flow:   
1. If the selected product does not exist, the system displays an error message and suggests verifying the product ID or name.   
2. If the deletion is not confirmed by the Administrator, the system cancels the operation and returns to the product management page.   
3. If the system fails to delete the product or update the Inventory due to technical issues, it displays an error message and logs the failure for review.  
  
Use Case Name: Create Order   
Use Case ID: UC-09   
Actors: Customer, Product, Inventory, Order, Payment Information, Email Notification   
Preconditions:   
1. The system is operational and accessible to the Customer.   
2. The Customer is logged in and has an active session.   
3. The Customer has selected at least one product from the Product data entity.   
4. The Inventory data entity contains sufficient stock for the selected products.   
5. The Customer has provided valid Payment Information.   
  
Postconditions:   
1. A new Order data entity is created and saved in the system.   
2. The Inventory data entity is updated to reflect the reduction in product stock.   
3. The system sends an Email Notification to the Customer confirming the order.   
4. The system logs the order creation activity for tracking and analytics.   
  
Main Flow:   
1. The Customer navigates to the shopping cart or checkout page.   
2. The Customer reviews the selected products and confirms the quantities.   
3. The system checks the Inventory data entity to ensure the selected products are available.   
4. The Customer provides or confirms their Payment Information.   
5. The system validates the Payment Information for correctness and availability.   
6. The system generates a unique order ID and creates a new Order data entity.   
7. The system updates the Inventory data entity to reduce the stock of the ordered products.   
8. The system sends an Email Notification to the Customer with the order confirmation.   
9. The system displays a confirmation message to the Customer that the order has been successfully placed.   
  
Alternative Flow:   
1. If the selected products are not available in the Inventory, the system displays an error message and suggests adjusting the order or selecting alternative products.   
2. If the Payment Information is invalid or incomplete, the system displays an error message and prompts the Customer to correct the details.   
3. If the system fails to generate an order ID, it alerts the Customer and suggests retrying the order creation or contacting support.   
4. If the Email Notification fails to send, the system logs the error and displays a message to the Customer, offering to resend the notification or provide the confirmation details manually.   
5. If the system cannot update the Inventory due to technical issues, it displays an error message and logs the failure for review.  
  
Use Case Name: View Order Details   
Use Case ID: UC-10   
Actors: Customer, Order, Product, Inventory, Administrator   
Preconditions:   
1. The system is operational and accessible to the Customer or Administrator.   
2. The Customer or Administrator is logged in and has an active session.   
3. The Order data entity contains at least one order record.   
  
Postconditions:   
1. The Customer or Administrator successfully views the detailed information of the selected order.   
2. The system displays order details, such as order ID, product list, quantities, total price, and order status.   
3. The system logs the access to the order details for tracking purposes.   
  
Main Flow:   
1. The Customer or Administrator navigates to the "View Order Details" section of the system.   
2. The Customer or Administrator selects an order from the list of their orders.   
3. The system retrieves the relevant Order data entity from the database.   
4. The system displays the order details, including associated Product records, Inventory status, and Payment Information.   
5. The system logs the request for viewing order details.   
  
Alternative Flow:   
1. If the selected order does not exist, the system displays an error message and suggests verifying the order ID or returning to the order list.   
2. If the system cannot retrieve the order details due to technical issues, it displays an error message and suggests retrying the action or contacting support.   
3. If the Inventory status for a product in the order is unavailable, the system displays a warning but still shows the order details without the inventory information.  
  
Use Case Name: Update Order Status   
Use Case ID: UC-11   
Actors: Administrator, Order, Inventory   
Preconditions:   
1. The system is operational and accessible to the Administrator.   
2. The Administrator is authenticated and authorized to manage order statuses.   
3. The selected order exists in the Order data entity.   
4. The Inventory data entity contains the relevant product stock information.   
  
Postconditions:   
1. The Order data entity is updated with the new status (e.g., Shipped, Cancelled, Processing).   
2. The Inventory data entity is updated if the order status change impacts product stock (e.g., cancellation increases stock).   
3. The system sends an Email Notification to the Customer associated with the order.   
4. The system logs the status update for tracking and auditing purposes.   
  
Main Flow:   
1. The Administrator navigates to the "Manage Orders" section of the system.   
2. The Administrator selects the order they wish to update.   
3. The system retrieves the Order data entity and displays the current status.   
4. The Administrator chooses a new status from the available options (e.g., Shipped, Cancelled).   
5. The system validates the status change and checks for inventory adjustments if needed.   
6. The system updates the Order data entity with the new status.   
7. If the status change affects inventory (e.g., order cancellation), the system updates the Inventory data entity accordingly.   
8. The system sends an Email Notification to the Customer to inform them of the status change.   
9. The system logs the update activity and displays a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the selected order does not exist, the system displays an error message and suggests verifying the order ID or name.   
2. If the status change is invalid or not allowed (e.g., attempting to ship a cancelled order), the system displays an error and prompts the Administrator to select a valid status.   
3. If the system cannot update the Inventory due to technical issues, it displays an error message and logs the failure for review.   
4. If the Email Notification fails to send, the system logs the error and allows the Administrator to manually retry the notification.   
5. If the system fails to update the Order data entity, it displays an error message and logs the failure for review.  
  
Use Case Name: Delete Order   
Use Case ID: UC-12   
Actors: Customer, Administrator, Order, Inventory, Email Notification   
Preconditions:   
1. The system is operational and accessible to the Customer or Administrator.   
2. The Customer or Administrator is authenticated and has an active session.   
3. The selected order exists in the Order data entity.   
4. The Inventory data entity contains the relevant product stock information.   
  
Postconditions:   
1. The selected Order data entity is successfully deleted from the system.   
2. The Inventory data entity is updated to reflect the return of ordered product quantities.   
3. The system sends an Email Notification to the Customer confirming the deletion.   
4. The system logs the deletion activity for tracking and auditing purposes.   
  
Main Flow:   
1. The Customer or Administrator navigates to the "Manage Orders" section of the system.   
2. The Customer or Administrator selects the order they wish to delete.   
3. The system retrieves the Order data entity and displays the details for confirmation.   
4. The Customer or Administrator confirms the deletion request.   
5. The system validates the confirmation and ensures the order is in a deletable state (e.g., not yet shipped).   
6. The system deletes the Order data entity from the database.   
7. The system updates the Inventory data entity to restore the product stock.   
8. The system sends an Email Notification to the Customer informing them that the order has been deleted.   
9. The system logs the deletion activity and displays a confirmation message to the user.   
  
Alternative Flow:   
1. If the selected order does not exist, the system displays an error message and suggests verifying the order ID or name.   
2. If the order is not in a deletable state (e.g., already shipped), the system displays an error and prevents deletion.   
3. If the system fails to update the Inventory due to technical issues, it displays an error message and logs the failure for review.   
4. If the Email Notification fails to send, the system logs the error and allows the Administrator to manually retry the notification.   
5. If the deletion confirmation is canceled by the user, the system returns to the order management page without making any changes.  
  
Use Case Name: View Inventory Levels   
Use Case ID: UC-13   
Actors: Administrator, Inventory, Product   
Preconditions:   
1. The system is operational and accessible to the Administrator.   
2. The Administrator is authenticated and authorized to view inventory levels.   
3. The Inventory data entity contains at least one product record.   
  
Postconditions:   
1. The Administrator successfully views the inventory levels for all or selected products.   
2. The system displays product names, quantities, and status (e.g., in stock, low stock, out of stock).   
3. The system logs the inventory view activity for tracking and auditing purposes.   
  
Main Flow:   
1. The Administrator navigates to the "Inventory Management" section of the system.   
2. The Administrator selects the "View Inventory Levels" option.   
3. The system retrieves all records from the Inventory data entity.   
4. The system displays the list of products with their respective quantities and status.   
5. The system logs the Administrator's access to inventory data.   
  
Alternative Flow:   
1. If no products are available in the Inventory data entity, the system displays a message indicating that the inventory is empty.   
2. If the system fails to retrieve the inventory data due to technical issues, it displays an error message and suggests retrying the action or contacting support.   
3. If the Administrator filters the inventory by product category or name, the system applies the filter and displays only the relevant product records.  
  
Use Case Name: Manage Inventory Stock   
Use Case ID: UC-14   
Actors: Administrator, Inventory, Product   
Preconditions:   
1. The system is operational and accessible to the Administrator.   
2. The Administrator is authenticated and authorized to manage inventory stock.   
3. The selected product exists in the Product data entity and is associated with an Inventory record.   
  
Postconditions:   
1. The Inventory data entity is updated with the new stock level for the selected product.   
2. The system displays a confirmation message to the Administrator.   
3. The system logs the inventory stock update for tracking and auditing purposes.   
  
Main Flow:   
1. The Administrator navigates to the "Inventory Management" section of the system.   
2. The Administrator selects the product for which the stock needs to be adjusted.   
3. The system retrieves the current stock level from the Inventory data entity and displays it.   
4. The Administrator enters the new stock quantity for the selected product.   
5. The system validates the new stock quantity to ensure it is a valid number and does not cause negative stock.   
6. The system updates the Inventory data entity with the new stock level.   
7. The system logs the stock adjustment and displays a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the new stock quantity is invalid or causes negative stock, the system displays an error message and prompts the Administrator to enter a valid quantity.   
2. If the selected product is not found in the system, the system displays an error and suggests verifying the product ID or name.   
3. If the system fails to update the Inventory data entity due to technical issues, it displays an error message and logs the failure for review.  
  
Use Case Name: Create Payment Information   
Use Case ID: UC-15   
Actors: Customer, Administrator, Payment Information, Email Notification   
  
Preconditions:   
1. The system is operational and accessible to the Customer or Administrator.   
2. The Customer or Administrator is authenticated and has an active session.   
3. The Customer has provided valid account information.   
4. The Payment Information data entity is accessible for updates.   
  
Postconditions:   
1. A new Payment Information data entity is successfully created in the system.   
2. The system sends an Email Notification to the Customer confirming the creation of the payment information.   
3. The system logs the creation activity for tracking and auditing purposes.   
  
Main Flow:   
1. The Customer or Administrator navigates to the "Payment Information" section of the system.   
2. The Customer or Administrator selects the "Create New Payment Method" option.   
3. The system prompts the user to input payment details, such as card number, expiration date, and billing address.   
4. The user inputs the required payment information.   
5. The system validates the input data for completeness and correctness.   
6. The system generates a unique identifier and creates a new Payment Information data entity.   
7. The system sends an Email Notification to the Customer confirming the successful creation.   
8. The system displays a confirmation message to the user.   
  
Alternative Flow:   
1. If the input data is incomplete or incorrect, the system displays an error message and prompts the user to correct the information.   
2. If the system fails to generate a unique identifier, it alerts the user and suggests retrying the action or contacting support.   
3. If the Email Notification fails to send, the system logs the error and allows the user to manually retry the notification.   
4. If the system fails to create the Payment Information due to technical issues, it displays an error message and logs the failure for review.  
  
Use Case Name: View Payment Information   
Use Case ID: UC-16   
Actors: Customer, Administrator, Payment Information, Email Notification   
Preconditions:   
1. The system is operational and accessible to the Customer or Administrator.   
2. The Customer or Administrator is authenticated and has an active session.   
3. The Payment Information data entity contains at least one payment method associated with the user.   
  
Postconditions:   
1. The Customer or Administrator successfully views their stored Payment Information.   
2. The system displays details such as card type, last four digits, expiration date, and billing address.   
3. The system logs the access to Payment Information for tracking and auditing.   
  
Main Flow:   
1. The Customer or Administrator navigates to the "Payment Information" section of the system.   
2. The system retrieves the Payment Information data entities linked to the user's account.   
3. The system displays a summary of the stored payment methods, including masked card numbers and relevant details.   
4. The system logs the access to the Payment Information.   
5. The system displays a confirmation message that the Payment Information is ready for viewing.   
  
Alternative Flow:   
1. If no Payment Information is found for the user, the system displays a message indicating that no payment methods are available.   
2. If the system fails to retrieve the Payment Information due to technical issues, it displays an error message and suggests retrying the action or contacting support.   
3. If the user does not have permission to view the Payment Information, the system displays an error and redirects them to a restricted page.  
  
Use Case Name: Update Payment Method   
Use Case ID: UC-17   
Actors: Customer, Administrator, Payment Information, Email Notification   
Preconditions:   
1. The system is operational and accessible to the Customer or Administrator.   
2. The Customer or Administrator is authenticated and has an active session.   
3. The existing Payment Information data entity is available for modification.   
4. The user has provided valid and updated payment details.   
  
Postconditions:   
1. The Payment Information data entity is successfully updated with the new payment method details.   
2. The system sends an Email Notification to the Customer confirming the update.   
3. The system logs the update activity for tracking and auditing purposes.   
  
Main Flow:   
1. The Customer or Administrator navigates to the "Payment Information" section of the system.   
2. The system retrieves and displays the existing Payment Information data entities linked to the user's account.   
3. The user selects the payment method they wish to update.   
4. The system prompts the user to input updated payment details, such as card number, expiration date, and billing address.   
5. The user inputs the new payment information.   
6. The system validates the updated data for completeness and correctness.   
7. The system updates the Payment Information data entity with the new details.   
8. The system sends an Email Notification to the Customer confirming the successful update.   
9. The system displays a confirmation message to the user.   
  
Alternative Flow:   
1. If the input data is incomplete or incorrect, the system displays an error message and prompts the user to correct the information.   
2. If the system fails to validate the updated payment information, it displays an error and suggests re-entering the details.   
3. If the Email Notification fails to send, the system logs the error and allows the user to manually retry the notification.   
4. If the system fails to update the Payment Information due to technical issues, it displays an error message and logs the failure for review.  
  
Use Case Name: Delete Payment Method   
Use Case ID: UC-18   
Actors: Customer, Administrator, Payment Information, Email Notification   
  
Preconditions:   
1. The system is operational and accessible to the Customer or Administrator.   
2. The Customer or Administrator is authenticated and has an active session.   
3. The Payment Information data entity contains the payment method to be deleted.   
  
Postconditions:   
1. The selected Payment Information data entity is successfully deleted from the system.   
2. The system sends an Email Notification to the Customer confirming the deletion.   
3. The system logs the deletion activity for tracking and auditing purposes.   
  
Main Flow:   
1. The Customer or Administrator navigates to the "Payment Information" section of the system.   
2. The system retrieves and displays the Payment Information data entities linked to the user's account.   
3. The user selects the payment method they wish to delete.   
4. The system prompts the user to confirm the deletion of the selected payment method.   
5. The user confirms the deletion request.   
6. The system deletes the selected Payment Information data entity from the database.   
7. The system sends an Email Notification to the Customer confirming the deletion.   
8. The system logs the deletion and displays a confirmation message to the user.   
  
Alternative Flow:   
1. If the selected payment method does not exist, the system displays an error message and suggests verifying the payment method details.   
2. If the user cancels the deletion request, the system returns to the Payment Information list without making any changes.   
3. If the system fails to delete the Payment Information due to technical issues, it displays an error message and logs the failure for review.   
4. If the Email Notification fails to send, the system logs the error and allows the user to manually retry the notification.  
  
Use Case Name: Register Administrator   
Use Case ID: UC-19   
Actors: Administrator, Email Notification, Plugin   
  
Preconditions:   
1. The system is operational and accessible to the Administrator.   
2. The Administrator has valid registration information (e.g., name, email, password).   
3. The Plugin is configured to handle administrative account creation.   
  
Postconditions:   
1. A new Administrator account is successfully created in the system.   
2. The Administrator receives an Email Notification confirming the registration.   
3. The system updates the Administrator data entity with the new account details.   
  
Main Flow:   
1. The system administrator initiates the registration process via the admin setup interface.   
2. The system prompts for the administrator’s details, including name, email, and password.   
3. The administrator enters the required information.   
4. The system validates the input data for completeness and correctness.   
5. The system generates a unique administrator ID and creates a new Administrator data entity.   
6. The system sends an Email Notification to the administrator's email to confirm the registration.   
7. The system logs the registration activity and displays a confirmation message to the user.   
  
Alternative Flow:   
1. If the input data is incomplete or incorrect, the system displays an error message and prompts the administrator to correct the information.   
2. If the system fails to generate a unique administrator ID, it alerts the user and suggests retrying the registration or contacting support.   
3. If the Email Notification fails to send, the system logs the error and allows the administrator to manually retry the notification.   
4. If the Plugin is not properly configured, the system displays an error message and prevents the registration until the Plugin is fixed.  
  
Use Case Name: Login as Administrator   
Use Case ID: UC-20   
Actors: Administrator   
Preconditions:   
1. The system is operational and accessible to the Administrator.   
2. The Administrator has already created an account in the system.   
3. The Administrator has a valid username and password.   
  
Postconditions:   
1. The Administrator is successfully authenticated and logged into the system.   
2. The system updates the session status of the Administrator.   
3. The Administrator is redirected to the admin dashboard.   
  
Main Flow:   
1. The Administrator navigates to the login page of the system.   
2. The Administrator enters their username and password.   
3. The system validates the credentials against the stored Administrator data entity.   
4. The system confirms the Administrator's identity and initiates a session.   
5. The system redirects the Administrator to the admin dashboard.   
6. The system displays a welcome message with the Administrator's name.   
  
Alternative Flow:   
1. If the entered username or password is invalid, the system displays an error message and prompts the Administrator to re-enter the correct credentials.   
2. If multiple login attempts fail, the system locks the account temporarily and sends an Email Notification to the Administrator.   
3. If the session cannot be initiated due to technical issues, the system displays an error and suggests retrying the login or contacting support.  
  
Use Case Name: Logout as Administrator   
Use Case ID: UC-21   
Actors: Administrator   
Preconditions:   
1. The system is operational and accessible to the Administrator.   
2. The Administrator is currently logged in and has an active session.   
  
Postconditions:   
1. The Administrator is successfully logged out of the system.   
2. The session status of the Administrator is updated to "logged out."   
3. The Administrator is redirected to the login page or homepage.   
  
Main Flow:   
1. The Administrator navigates to the account or session management section of the system.   
2. The Administrator selects the "Logout" option.   
3. The system verifies the active session and processes the logout request.   
4. The system updates the Administrator's session status to "logged out."   
5. The system redirects the Administrator to the login page or homepage.   
6. The system displays a confirmation message that the Administrator has been logged out.   
  
Alternative Flow:   
1. If the logout request fails due to technical issues, the system displays an error message and suggests retrying the action.   
2. If the session is already expired or invalid, the system automatically redirects the Administrator to the login page without further action.  
  
Use Case Name: Send Email Notification   
Use Case ID: UC-22   
Actors: System, Customer, Administrator, Email Notification   
Preconditions:   
1. The system is operational and accessible to the Administrator or automated system processes.   
2. The recipient (Customer or Administrator) has a valid email address associated with their account.   
3. The system has the necessary content or message template for the notification.   
4. The Email Notification data entity is properly configured and accessible.   
  
Postconditions:   
1. The email notification is successfully sent to the recipient.   
2. The system logs the email notification activity for tracking and auditing.   
3. The recipient receives the notification and can take appropriate action based on the message content.   
  
Main Flow:   
1. The system triggers the "Send Email Notification" process based on a specific event (e.g., order confirmation, account creation).   
2. The system retrieves the recipient’s email address from the relevant data entity (e.g., Customer or Administrator).   
3. The system generates the email message using a predefined template and includes relevant details (e.g., order ID, account information).   
4. The system sends the email through the Email Notification data entity.   
5. The system logs the successful delivery of the notification.   
6. The system displays a confirmation message to the user initiating the process (e.g., Administrator or automated system).   
  
Alternative Flow:   
1. If the recipient’s email address is missing or invalid, the system displays an error and prevents the notification from being sent.   
2. If the email template is not available or contains errors, the system alerts the user and suggests correcting the template.   
3. If the Email Notification fails to send, the system logs the error and provides an option to manually retry the notification.   
4. If the system fails to log the notification activity, it displays an error message and logs the failure for review.  
  
Use Case Name: View Notification Logs   
Use Case ID: UC-23   
Actors: Administrator   
Preconditions:   
1. The system is operational and accessible to the Administrator.   
2. The Administrator is authenticated and has an active session.   
3. The Email Notification data entity contains at least one notification log.   
  
Postconditions:   
1. The Administrator successfully views the notification logs.   
2. The system displays details such as recipient, message content, timestamp, and delivery status.   
3. The system logs the access to the notification logs for auditing purposes.   
  
Main Flow:   
1. The Administrator navigates to the "Notification Logs" section of the system.   
2. The system retrieves all records from the Email Notification data entity.   
3. The system displays a list of notifications, including recipient, message content, timestamp, and delivery status.   
4. The Administrator can filter or search the logs based on specific criteria (e.g., date, recipient, status).   
5. The system logs the Administrator's access to the notification logs.   
6. The system displays a confirmation message that the logs are ready for viewing.   
  
Alternative Flow:   
1. If no notification logs are found, the system displays a message indicating that there are no records available.   
2. If the system fails to retrieve the notification logs due to technical issues, it displays an error message and suggests retrying the action or contacting support.   
3. If the Administrator enters invalid or inappropriate filter criteria, the system displays an error and prompts them to refine the search.  
  
Use Case Name: Configure Plugin Settings   
Use Case ID: UC-24   
Actors: Administrator, Plugin   
Preconditions:   
1. The system is operational and accessible to the Administrator.   
2. The Administrator is authenticated and authorized to configure plugin settings.   
3. A Plugin is already installed and registered in the system.   
  
Postconditions:   
1. The Plugin settings are successfully updated in the system.   
2. The system logs the configuration changes for tracking and auditing.   
3. The Plugin is either reloaded or restarted to apply the new settings.   
  
Main Flow:   
1. The Administrator navigates to the "Plugin Management" section of the system.   
2. The Administrator selects the Plugin they wish to configure.   
3. The system retrieves the current Plugin settings and displays them.   
4. The Administrator modifies the Plugin settings as needed (e.g., enabling/disabling features, setting thresholds).   
5. The system validates the updated settings for correctness and compatibility.   
6. The system saves the updated Plugin configuration.   
7. The system logs the configuration change.   
8. The system reloads or restarts the Plugin to apply the new settings.   
9. The system displays a confirmation message to the Administrator indicating that the settings have been updated.   
  
Alternative Flow:   
1. If the Plugin settings are invalid or incompatible, the system displays an error message and prompts the Administrator to correct the values.   
2. If the Plugin cannot be reloaded or restarted due to technical issues, the system displays an error message and logs the failure for review.   
3. If the selected Plugin is not found in the system, the system displays an error message and suggests verifying the plugin name or ID.   
4. If the Administrator cancels the configuration process, the system returns to the Plugin management page without making any changes.  
  
Use Case Name: Manage Product Category   
Use Case ID: UC-25   
Actors: Administrator, Product Category, Product, Plugin   
  
Preconditions:   
1. The system is operational and accessible to the Administrator.   
2. The Administrator is authenticated and authorized to manage product categories.   
3. The Product Category data entity exists and is accessible for modification.   
4. The Plugin is configured to support product category management.   
  
Postconditions:   
1. The Product Category data entity is successfully updated, added, or deleted.   
2. The associated Product data entities are updated to reflect any changes in category assignments.   
3. The system logs the activity for tracking and auditing purposes.   
4. The Plugin settings, if applicable, are updated to align with the new category structure.   
  
Main Flow:   
1. The Administrator navigates to the "Product Category Management" section of the system.   
2. The Administrator selects an action: "Add New Category," "Edit Existing Category," or "Delete Category."   
3. For "Add New Category," the Administrator inputs the category name and description.   
4. For "Edit Existing Category," the Administrator selects a category and modifies its name or description.   
5. For "Delete Category," the Administrator selects a category and confirms the deletion.   
6. The system validates the input data for correctness and consistency.   
7. The system updates the Product Category data entity based on the selected action.   
8. If the category is deleted, the system checks for associated Product records and updates their category references accordingly.   
9. The system logs the activity (addition, modification, or deletion) for auditing.   
10. If the Plugin is affected by the changes, the system updates or reloads the Plugin configuration.   
11. The system displays a confirmation message to the Administrator indicating the success of the action.   
  
Alternative Flow:   
1. If the input data is incomplete or incorrect, the system displays an error message and prompts the Administrator to correct the information.   
2. If the selected category for deletion contains products, the system displays a warning and asks whether to proceed with reassigning the products or cancel the deletion.   
3. If the system fails to update the Product Category data entity due to technical issues, it displays an error message and logs the failure for review.   
4. If the system cannot update the associated Product records (e.g., due to constraints or errors), it alerts the Administrator and prevents the category deletion or modification.   
5. If the Plugin is not properly configured to support category management, the system displays an error and prevents the update until the Plugin is fixed.   
6. If the Administrator cancels the action at any point, the system returns to the Product Category Management page without making any changes.  
  
Use Case Name: Manage Product Details   
Use Case ID: UC-26   
Actors: Administrator, Product, Inventory, Plugin   
  
Preconditions:   
1. The system is operational and accessible to the Administrator.   
2. The Administrator is authenticated and authorized to manage product details.   
3. The selected product exists in the Product data entity.   
4. The Plugin is configured to support advanced product detail management.   
  
Postconditions:   
1. The Product data entity is successfully updated with modified details (e.g., name, price, description, category).   
2. The Inventory data entity is updated if the changes affect product stock or availability.   
3. The system logs the activity for tracking and auditing purposes.   
4. The Plugin, if involved, reflects the updated product details in its functionality or interface.   
  
Main Flow:   
1. The Administrator navigates to the "Manage Product Details" section of the system.   
2. The Administrator selects the product they wish to modify from the product list.   
3. The system retrieves the Product data entity and displays the current details (e.g., name, price, description, category, and inventory status).   
4. The Administrator modifies the product details as needed (e.g., updates pricing, changes category, or revises product description).   
5. The system validates the updated information for correctness and consistency.   
6. If the category is changed, the system updates the Product Category data entity accordingly.   
7. If the modification affects inventory (e.g., availability or stock status), the system updates the Inventory data entity.   
8. The system logs the modification activity, including the user, timestamp, and changes made.   
9. If the Plugin is affected by the changes, the system updates or reloads the Plugin configuration.   
10. The system displays a confirmation message to the Administrator indicating the product details have been successfully updated.   
  
Alternative Flow:   
1. If the selected product does not exist in the system, the system displays an error message and suggests verifying the product ID or name.   
2. If the updated product details are incomplete or incorrect, the system displays an error message and prompts the Administrator to correct the information.   
3. If the system fails to update the Product data entity due to technical issues, it displays an error message and logs the failure for review.   
4. If the system cannot update the Inventory data entity (e.g., due to constraints or errors), it alerts the Administrator and prevents the modification until the issue is resolved.   
5. If the Plugin is not properly configured to support the changes, the system displays an error message and prevents the modification until the Plugin is fixed.   
6. If the Administrator cancels the modification process at any point, the system returns to the product management page without making any changes.