项目文档

# Functional Requirement

1.1 Customer Registration Function   
 Function ID: FR-01   
 Description: A new customer can register by providing personal information such as name, email, password, address, and phone number. The system validates the input and creates a customer account, followed by sending an email confirmation.   
 Input: Customer information (Name, Email, Password, Address, PhoneNumber)   
 Output: A new Customer entity in the database and an EmailConfirmation entity generated and sent to the customer.  
  
1.2 Customer Login Function   
 Function ID: FR-02   
 Description: A registered and verified customer can log in to the system using their email and password. The system authenticates the credentials and starts a session.   
 Input: Email and Password   
 Output: A valid session initiated for the customer and a confirmation of successful login.  
  
1.3 View Product Details Function   
 Function ID: FR-03   
 Description: A customer (logged in or guest) can view detailed information about a specific product, such as name, description, price, and category. The system also logs the view activity.   
 Input: ProductID   
 Output: Product details and a logged ProductView activity in the system.  
  
1.4 Add Product to Cart Function   
 Function ID: FR-04   
 Description: A customer can add a product to their shopping cart. The system checks the product's availability in the inventory and updates the cart and inventory accordingly.   
 Input: ProductID, Quantity   
 Output: A new CartItem entity added to the ShoppingCart and an updated Inventory record reflecting reduced stock.  
  
1.5 Update Shopping Cart Function   
 Function ID: FR-05   
 Description: A customer can update the quantity of a product in their shopping cart or remove it. The system validates the updated quantity against the available stock and adjusts the cart and inventory.   
 Input: CartItemID, NewQuantity or Remove flag   
 Output: Updated CartItem and ShoppingCart entities, and an updated Inventory record.  
  
1.6 Remove Product from Cart Function   
 Function ID: FR-06   
 Description: A customer can remove a specific product from their shopping cart. The system updates the cart and inventory accordingly.   
 Input: CartItemID   
 Output: Updated CartItem and ShoppingCart entities, and an updated Inventory record.  
  
1.7 Checkout Function   
 Function ID: FR-07   
 Description: A customer can proceed with the checkout process by providing shipping address, selecting a payment method, and confirming the order. The system processes the payment, creates an order, and sends a confirmation email.   
 Input: ShoppingCart, ShippingAddress, PaymentMethodID   
 Output: A new Order entity, updated Inventory records, a new Payment entity, and an EmailConfirmation generated and sent.  
  
1.8 Confirm Order Function   
 Function ID: FR-08   
 Description: A customer can confirm their order after the checkout process. The system verifies the availability of all items in the cart, confirms the payment, and finalizes the order.   
 Input: OrderID   
 Output: Updated Order entity marked as confirmed, updated Inventory records, and a confirmation message displayed to the customer.  
  
1.9 Receive Order Confirmation Email Function   
 Function ID: FR-09   
 Description: After an order is confirmed, the system sends an email to the customer with order details. The system logs the email delivery and provides the necessary information for the customer to review.   
 Input: OrderID, CustomerEmail   
 Output: An EmailConfirmation entity in the database and the email sent to the customer.  
  
1.10 Administrator Login Function   
 Function ID: FR-10   
 Description: An administrator can log in to the system using their username and password. The system authenticates the credentials and starts a session.   
 Input: AdminEmail, AdminPassword   
 Output: A valid session initiated for the administrator and a confirmation of successful login.  
  
1.11 Manage Inventory Function   
 Function ID: FR-11   
 Description: An administrator can manage the inventory by adding, removing, or modifying products. The system updates the Inventory and Product entities and logs the changes.   
 Input: Product details (Name, Description, Price, Quantity, CategoryID) or Inventory action (Add, Remove, Modify)   
 Output: Updated Inventory and Product entities, and a confirmation message displayed to the administrator.  
  
1.12 Manage Product Categories Function   
 Function ID: FR-12   
 Description: An administrator can manage product categories by creating, editing, or deleting categories. The system updates the Category and Product entities accordingly and logs the activity.   
 Input: Category details (Name, Description, ParentCategoryID) or Category action (Add, Edit, Delete)   
 Output: Updated Category and Product entities, and a confirmation message displayed to the administrator.  
  
1.13 Update Product Information Function   
 Function ID: FR-13   
 Description: An administrator can update product details such as name, description, price, stock level, and category. The system validates and updates the relevant Product and Inventory entities and logs the action.   
 Input: ProductID, NewProductDetails (Name, Description, Price, Quantity, CategoryID)   
 Output: Updated Product and Inventory entities, and a confirmation message displayed to the administrator.  
  
1.14 Delete Product Function   
 Function ID: FR-14   
 Description: An administrator can delete a product from the system if it is not part of any active order. The system removes the product from the Inventory and updates the Category associations and logs the action.   
 Input: ProductID   
 Output: Deleted Product entity, updated Inventory and Category entities, and a confirmation message displayed to the administrator.  
  
1.15 Manage Payment Methods Function   
 Function ID: FR-15   
 Description: An administrator can manage available payment methods by adding, updating, or removing them. The system validates and updates the PaymentMethod entity and logs the activity.   
 Input: PaymentMethod details (Name, Description, Status) or PaymentMethod action (Add, Update, Remove)   
 Output: Updated PaymentMethod entity and a confirmation message displayed to the administrator.  
  
1.16 View Purchase History Function   
 Function ID: FR-16   
 Description: A customer can view their previous orders and associated product details. The system retrieves and displays the order history and logs the access.   
 Input: CustomerID   
 Output: List of Order entities with associated OrderItem and Product details, and a confirmation of access logged.  
  
1.17 Manage Plugins Function   
 Function ID: FR-17   
 Description: An administrator can manage plugins by installing, updating, or removing them. The system validates the plugin file, updates the Plugin entity, and logs the activity.   
 Input: Plugin details (Name, Version, Description, Status) or Plugin action (Install, Update, Remove)   
 Output: Updated Plugin entity and a confirmation message displayed to the administrator.  
  
1.18 View Email Confirmation Function   
 Function ID: FR-18   
 Description: A customer can view the content of an email confirmation by clicking on the provided link. The system retrieves the confirmation content from the database and logs the access.   
 Input: ConfirmationCode, CustomerID   
 Output: Displayed EmailConfirmation content, and a logged access activity.  
  
1.19 Manage Cart Item Function   
 Function ID: FR-19   
 Description: A customer can modify or remove a specific item in their shopping cart. The system validates the action and updates the CartItem, ShoppingCart, and Inventory entities accordingly.   
 Input: CartItemID, NewQuantity or Remove flag   
 Output: Updated CartItem and ShoppingCart entities, and an updated Inventory record.  
  
1.20 Manage Order Item Function   
 Function ID: FR-20   
 Description: An administrator can modify or remove an item from an existing order. The system validates the action, updates the OrderItem, Order, and Inventory entities, and logs the activity.   
 Input: OrderItemID, NewQuantity or Remove flag   
 Output: Updated OrderItem and Order entities, and an updated Inventory record.

# External Description

2. External Interfaces  
  
2.1 User Interface Output   
The system interacts with users (customers and administrators) through a web-based graphical user interface (GUI) and a mobile-responsive interface. These interfaces support the following interactions:  
  
- \*\*FR-01 (Customer Registration):\*\* The system provides a registration form for customers to input personal details. After successful registration, a confirmation message is displayed, and an email confirmation is sent.  
- \*\*FR-02 (Customer Login):\*\* A login form is presented to customers, allowing them to enter their email and password. Upon successful authentication, a confirmation message is displayed, and a session is initiated.  
- \*\*FR-03 (View Product Details):\*\* The system displays detailed information about a product, including name, description, price, and category. A visual indicator or log message may be shown to the user to confirm that their activity has been recorded.  
- \*\*FR-04 (Add Product to Cart):\*\* A confirmation message is shown to the user indicating the product was successfully added to the cart. The system also provides real-time updates to the shopping cart interface.  
- \*\*FR-05 (Update Shopping Cart):\*\* The system updates the cart interface to reflect the new quantity or removal of a product. A confirmation message is displayed for the action.  
- \*\*FR-06 (Remove Product from Cart):\*\* The system updates the cart interface and displays a confirmation message that the product was removed.  
- \*\*FR-07 (Checkout):\*\* The system presents a checkout page where the customer can enter their shipping address and select a payment method. After the payment is processed and the order is created, a confirmation message is displayed.  
- \*\*FR-08 (Confirm Order):\*\* A confirmation message is shown to the customer after the system verifies the order and finalizes it.  
- \*\*FR-10 (Administrator Login):\*\* A login form is presented to administrators, and a session is initiated with a confirmation message after successful authentication.  
- \*\*FR-16 (View Purchase History):\*\* The system displays a list of previous orders, including associated product details, to the customer. A confirmation message is shown after access is logged.  
- \*\*FR-18 (View Email Confirmation):\*\* The system retrieves and displays the content of an email confirmation when a customer clicks on the provided link.  
- \*\*FR-20 (Manage Order Item):\*\* A confirmation message is shown to the administrator after an order item is updated or removed.  
  
2.2 Hardware Interface Output   
The system does not directly interact with any external hardware devices. It is expected to be compatible with standard web and mobile devices, including desktops, laptops, tablets, and smartphones. No special hardware interfaces or communication protocols are required for system functionality.  
  
2.3 Software Interface Output   
The system interacts with several external software components and databases to fulfill its functional requirements. These interfaces are described below:  
  
- \*\*Database (DB):\*\*   
 The system maintains a relational database to store entities such as Customer, Product, ShoppingCart, CartItem, Order, OrderItem, Inventory, PaymentMethod, Plugin, and EmailConfirmation. The system performs CRUD (Create, Read, Update, Delete) operations on these entities to support functional requirements.  
  
 - \*\*FR-01, FR-07, FR-09:\*\* The system creates and updates Customer and EmailConfirmation entities in the database.   
 - \*\*FR-02, FR-10:\*\* The system reads and updates session data in the database to manage user authentication.   
 - \*\*FR-03, FR-16:\*\* The system retrieves Product and Order entities from the database and logs activities.   
 - \*\*FR-04, FR-05, FR-06, FR-19:\*\* The system updates the ShoppingCart and CartItem entities and adjusts Inventory records.   
 - \*\*FR-07, FR-08:\*\* The system creates and updates Order and Payment entities and modifies Inventory records.   
 - \*\*FR-11, FR-13, FR-14:\*\* The system updates Inventory and Product entities.   
 - \*\*FR-12, FR-17:\*\* The system updates Category and Plugin entities.   
 - \*\*FR-15:\*\* The system manages PaymentMethod entities.   
 - \*\*FR-20:\*\* The system updates OrderItem, Order, and Inventory entities.  
  
- \*\*Email Service (Email API):\*\*   
 The system integrates with an external email service to send and log confirmation emails to customers. The email service is responsible for delivering the message to the customer's inbox.  
  
 - \*\*FR-01, FR-09:\*\* The system sends an email to the customer’s email address with a confirmation link or order details.   
 - \*\*FR-07, FR-08:\*\* An email is sent to the customer after the order is confirmed.   
 - \*\*FR-18:\*\* The system retrieves the EmailConfirmation content from the database and displays it to the customer when they click on the confirmation link.  
  
- \*\*Payment Gateway (Payment API):\*\*   
 The system integrates with a third-party payment gateway to process customer payments during the checkout process. The payment gateway provides a secure interface for transaction processing and returns a payment status to the system.  
  
 - \*\*FR-07, FR-08, FR-15:\*\* The system communicates with the Payment Gateway to process payments, verify payment methods, and update the Payment entity in the database.  
  
- \*\*Logging System (Log API):\*\*   
 The system uses a logging service to record user activities, such as viewing product details, updating the cart, or managing orders. The logging system stores these activities for audit and analytics purposes.  
  
 - \*\*FR-03, FR-05, FR-06, FR-11, FR-12, FR-13, FR-14, FR-15, FR-16, FR-17, FR-19, FR-20:\*\* The system logs activities related to user interactions and administrative changes.  
  
2.4 Communication Interface Output   
The system uses network-based communication to interact with external services and users:  
  
- \*\*Email Communication (Email API):\*\*   
 The system sends confirmation emails to customers using an external email service. The email includes a unique confirmation code or order details. The system also logs the delivery of these emails for tracking purposes.  
  
 - \*\*FR-01, FR-07, FR-08, FR-09, FR-18:\*\* The system communicates with the email service to send and verify email confirmations.  
  
- \*\*Web Browsing (HTTP/HTTPS):\*\*   
 The system is accessed through a web browser using HTTP/HTTPS protocols. This allows users to interact with the system via the GUI and perform actions such as registration, login, and product viewing.  
  
 - \*\*All user-facing functions (FR-01 to FR-08, FR-16 to FR-18, FR-19 to FR-20):\*\* The system communicates with users through a web browser to display interfaces and receive input.  
  
- \*\*Payment Gateway Communication (HTTP/HTTPS):\*\*   
 The system communicates with the Payment Gateway via HTTP/HTTPS to securely process transactions. This includes sending payment requests and receiving transaction confirmations.  
  
 - \*\*FR-07, FR-08, FR-15:\*\* The system uses secure communication to interact with the Payment Gateway during checkout and payment management.  
  
- \*\*API Communication for Inventory and Product Management:\*\*   
 The system communicates with internal APIs to manage Inventory and Product entities. These APIs ensure consistency between inventory levels and product information.  
  
 - \*\*FR-04, FR-05, FR-06, FR-11, FR-13, FR-14, FR-19, FR-20:\*\* The system interacts with inventory-related APIs to update stock levels and product data.  
  
- \*\*Plugin Communication (Plugin API):\*\*   
 The system interacts with a plugin API to install, update, or remove plugins. This API validates the plugin file and updates the Plugin entity accordingly.  
  
 - \*\*FR-17:\*\* The system uses the Plugin API to manage plugins.  
  
By defining these external interfaces, the system ensures clear communication with users, external services, and internal software components. Developers can use this information to implement the necessary integrations and interactions for the system.

# Use Case

Use Case Name: Customer Registration   
Use Case ID: UC-01   
Actors: Customer, Administrator, System   
Preconditions:   
1. The customer is not registered in the system.   
2. The system is accessible and operational.   
  
Postconditions:   
1. The customer is successfully registered in the system.   
2. An email confirmation is sent to the customer.   
  
Main Flow:   
1. The customer accesses the registration page of the system.   
2. The system displays the customer registration form.   
3. The customer fills in the required information (e.g., name, email, password, address, phone number).   
4. The customer submits the registration form.   
5. The system validates the input data (e.g., checks for valid email format, unique email, and password strength).   
6. The system creates a new customer profile in the customer database.   
7. The system generates an email confirmation and sends it to the provided email address.   
8. The customer receives the confirmation email and clicks on the verification link.   
9. The system confirms the customer's email and updates the profile status to "verified."   
10. The system displays a registration confirmation message to the customer.   
  
Alternative Flow:   
1. If the customer provides an invalid email format, the system displays an error message and prompts the customer to correct the email.   
2. If the email is already registered, the system displays an error message and prompts the customer to enter a different email.   
3. If the password does not meet the strength requirements, the system displays an error message and prompts the customer to choose a stronger password.   
4. If the system fails to send the email confirmation, the system displays an error message and suggests the customer try again or contact support.   
5. If the customer does not click the verification link within a specified time, the link expires, and the system prompts the customer to request a new confirmation email.  
  
Use Case Name: Customer Login   
Use Case ID: UC-02   
Actors: Customer, System   
Preconditions:   
1. The customer is already registered in the system and has a verified email.   
2. The system is accessible and operational.   
  
Postconditions:   
1. The customer is successfully logged in to the system.   
2. The customer’s session is initiated.   
  
Main Flow:   
1. The customer accesses the login page of the system.   
2. The system displays the login form with fields for email and password.   
3. The customer enters their registered email and password.   
4. The customer submits the login form.   
5. The system verifies the email and password against the customer database.   
6. If the credentials are valid, the system authenticates the customer and starts a session.   
7. The system redirects the customer to their account dashboard or the home page.   
  
Alternative Flow:   
1. If the customer enters an incorrect email or password, the system displays an error message and prompts the customer to try again.   
2. If the customer account is locked due to multiple failed login attempts, the system displays a message and suggests contacting support or waiting for a cooldown period.   
3. If the customer forgets their password, they can click on the "Forgot Password" link, and the system will initiate the password reset process via email.   
4. If the system fails to authenticate the customer, it displays an error message and logs the event for security monitoring.  
  
Use Case Name: View Product Details   
Use Case ID: UC-03   
Actors: Customer, System, Product, Category   
Preconditions:   
1. The customer is logged in to the system or is a guest user.   
2. The product exists in the inventory.   
3. The system is accessible and operational.   
  
Postconditions:   
1. The customer has viewed the product details.   
2. The system has recorded the product view activity.   
  
Main Flow:   
1. The customer browses the product catalog or searches for a product.   
2. The customer selects a specific product from the list.   
3. The system retrieves the product details from the product database.   
4. The system displays the product information, including name, description, price, image, and category.   
5. The customer reviews the product details.   
6. The system logs the product view in the activity history.   
  
Alternative Flow:   
1. If the product is not available in the inventory, the system displays a message indicating the product is out of stock.   
2. If the system fails to retrieve product details, it displays an error message and suggests trying again later.   
3. If the customer is not logged in, the system may prompt them to register or log in to access additional features.   
4. If the product image is missing, the system displays a placeholder image and a message indicating the issue.  
  
Use Case Name: Add Product to Cart   
Use Case ID: UC-04   
Actors: Customer, System, Product, Shopping Cart, Inventory   
  
Preconditions:   
1. The customer is logged in to the system or is a guest user.   
2. The product exists in the inventory and is available for purchase.   
3. The shopping cart is accessible and functional.   
4. The system is accessible and operational.   
  
Postconditions:   
1. The selected product is successfully added to the shopping cart.   
2. The shopping cart is updated to reflect the new product addition.   
3. The inventory records the product's current stock level after the addition.   
  
Main Flow:   
1. The customer views a product or browses through the product catalog.   
2. The customer clicks the "Add to Cart" button for the selected product.   
3. The system checks the product's availability in the inventory.   
4. The system adds the product to the customer's shopping cart.   
5. The system updates the cart display to show the newly added item.   
6. The system updates the inventory to reflect the reduced stock for the product.   
7. The system provides a confirmation message that the product was added to the cart.   
  
Alternative Flow:   
1. If the product is out of stock, the system displays a message indicating the unavailability and suggests alternatives.   
2. If the system cannot update the inventory, it displays an error message and prevents the product from being added to the cart.   
3. If the customer attempts to add more items than available in the inventory, the system displays an error message and limits the quantity to the available stock.   
4. If the shopping cart is not accessible, the system displays an error message and suggests refreshing the page or contacting support.   
5. If the customer is not logged in and tries to add an item to the cart, the system may prompt them to log in or continue as a guest.  
  
Use Case Name: Update Shopping Cart   
Use Case ID: UC-05   
Actors: Customer, System, Shopping Cart, Product, Inventory   
  
Preconditions:   
1. The customer is logged in to the system or is a guest user.   
2. The shopping cart contains at least one item.   
3. The system is accessible and operational.   
4. The product to be updated exists in the inventory.   
  
Postconditions:   
1. The shopping cart is updated with the new quantity or removal of items.   
2. The inventory records the updated stock level based on the cart changes.   
3. The system provides confirmation of the cart update to the customer.   
  
Main Flow:   
1. The customer accesses the shopping cart page.   
2. The system displays the current items in the customer’s shopping cart.   
3. The customer selects an item and modifies its quantity or removes it.   
4. The system validates the updated quantity against the available stock.   
5. The system updates the shopping cart with the changes.   
6. The system updates the inventory to reflect any changes in stock.   
7. The system provides a confirmation message that the cart has been updated.   
  
Alternative Flow:   
1. If the updated quantity exceeds the available stock, the system displays an error message and limits the quantity to the available stock.   
2. If the system cannot update the cart due to an error, it displays an error message and suggests refreshing the page or contacting support.   
3. If the customer removes all items from the cart, the system displays a message indicating the cart is now empty.   
4. If the inventory is not accessible, the system prevents the cart update and displays an error message.   
5. If the customer is not logged in and tries to update the cart, the system may prompt them to log in or continue as a guest.  
  
Use Case Name: Remove Product from Cart   
Use Case ID: UC-06   
Actors: Customer, System, Shopping Cart, Product, Inventory   
  
Preconditions:   
1. The customer is logged in to the system or is a guest user.   
2. The shopping cart contains at least one item.   
3. The selected product exists in the shopping cart.   
4. The system is accessible and operational.   
  
Postconditions:   
1. The selected product is removed from the shopping cart.   
2. The shopping cart is updated to reflect the removal.   
3. The inventory records the updated stock level after the removal.   
4. The system provides confirmation of the product removal to the customer.   
  
Main Flow:   
1. The customer accesses the shopping cart page.   
2. The system displays the current items in the customer’s shopping cart.   
3. The customer selects a product and clicks the "Remove" button.   
4. The system removes the selected product from the shopping cart.   
5. The system updates the cart display to reflect the removal.   
6. The system updates the inventory to increase the stock of the removed product.   
7. The system provides a confirmation message that the product was removed from the cart.   
  
Alternative Flow:   
1. If the selected product does not exist in the cart, the system displays an error message indicating the product could not be removed.   
2. If the system cannot update the inventory, it displays an error message and prevents the removal of the product from the cart.   
3. If the shopping cart is not accessible, the system displays an error message and suggests refreshing the page or contacting support.   
4. If the customer is not logged in and tries to remove an item from the cart, the system may prompt them to log in or continue as a guest.  
  
Use Case Name: Checkout   
Use Case ID: UC-07   
Actors: Customer, System, Shopping Cart, Product, Inventory, Order, Payment, Email Confirmation   
  
Preconditions:   
1. The customer is logged in to the system or is a guest user.   
2. The shopping cart contains at least one item.   
3. The selected products exist in the inventory and are available for purchase.   
4. The system is accessible and operational.   
5. The payment gateway is functional.   
  
Postconditions:   
1. The customer has successfully completed the checkout process.   
2. An order has been created and stored in the order database.   
3. Payment has been processed and recorded.   
4. Inventory has been updated to reflect the reduction in stock for the purchased items.   
5. An email confirmation has been sent to the customer with order details.   
  
Main Flow:   
1. The customer accesses the shopping cart page and clicks the "Checkout" button.   
2. The system displays the checkout form, which includes shipping address, payment method, and order summary.   
3. The customer fills in the required shipping information.   
4. The customer selects a payment method (e.g., credit card, PayPal, or bank transfer).   
5. The customer reviews the order summary, including product details, quantities, and total price.   
6. The customer submits the checkout form.   
7. The system verifies the shipping address and payment method.   
8. The system processes the payment through the selected payment gateway.   
9. If the payment is successful, the system creates a new order in the order database.   
10. The system updates the inventory to reduce the stock for each purchased product.   
11. The system generates an email confirmation and sends it to the customer's registered email address.   
12. The system displays a confirmation message to the customer, including the order number and estimated delivery time.   
  
Alternative Flow:   
1. If the customer is not logged in and tries to checkout, the system may prompt them to register or log in to proceed.   
2. If the shipping address is invalid or incomplete, the system displays an error message and prompts the customer to correct the address.   
3. If the selected payment method is not valid or the payment fails, the system displays an error message and allows the customer to choose another payment option.   
4. If the inventory is not accessible during checkout, the system prevents the order from being completed and displays an error message.   
5. If the system fails to send the email confirmation, it displays an error message and suggests the customer check their inbox or contact support.   
6. If the customer cancels the checkout process, the system returns them to the shopping cart page without any changes to the inventory or order database.  
  
Use Case Name: Confirm Order   
Use Case ID: UC-08   
Actors: Customer, System, Shopping Cart, Product, Inventory, Order, Payment, Email Confirmation   
  
Preconditions:   
1. The customer has completed the checkout process and provided valid shipping and payment information.   
2. The shopping cart contains at least one item.   
3. The system has processed the payment successfully.   
4. The system is accessible and operational.   
  
Postconditions:   
1. The customer’s order is confirmed and finalized.   
2. The order is stored in the order database.   
3. The inventory is updated to reflect the sale.   
4. The payment is recorded as successful.   
5. An email confirmation is sent to the customer.   
  
Main Flow:   
1. The customer completes the checkout process and clicks the "Confirm Order" button.   
2. The system verifies all items in the shopping cart are still available in the inventory.   
3. The system confirms the payment has been processed.   
4. The system creates a new order in the order database with the customer's information and product details.   
5. The system updates the inventory to reduce the stock of each purchased item.   
6. The system generates an email confirmation containing the order details and sends it to the customer's registered email address.   
7. The system displays an order confirmation message to the customer, including the order number and estimated delivery time.   
  
Alternative Flow:   
1. If any item in the shopping cart is no longer available, the system displays an error message and prevents the order confirmation.   
2. If the payment fails during the confirmation step, the system displays an error message and allows the customer to reattempt the payment.   
3. If the system fails to update the inventory, it displays an error message and cancels the order confirmation to prevent data inconsistency.   
4. If the system fails to send the email confirmation, it displays an error message and suggests the customer check their inbox or contact support.   
5. If the customer cancels the order confirmation, the system returns them to the checkout page without proceeding with the order or inventory update.  
  
Use Case Name: Receive Order Confirmation Email   
Use Case ID: UC-09   
Actors: Customer, System, Email Confirmation, Order, Payment   
  
Preconditions:   
1. The customer has successfully completed the checkout and confirmed the order.   
2. The system has processed the payment successfully.   
3. The order has been created and stored in the order database.   
4. The system has access to the email server.   
  
Postconditions:   
1. The customer receives an order confirmation email.   
2. The email contains the order details, including order number, product list, total amount, and estimated delivery time.   
3. The system logs the successful email delivery.   
  
Main Flow:   
1. The system triggers the email confirmation process after the order is confirmed.   
2. The system retrieves the customer’s email address and order details from the database.   
3. The system generates the order confirmation email with relevant information.   
4. The system sends the email to the customer's registered email address.   
5. The customer receives the email and can view the order confirmation.   
  
Alternative Flow:   
1. If the system fails to send the email confirmation, it displays an error message and suggests the customer check their inbox or contact support.   
2. If the customer's email address is invalid or missing, the system logs the issue and notifies the administrator for review.   
3. If the email server is unavailable, the system displays an error message and reschedules the email delivery.   
4. If the customer does not receive the confirmation email, they can request a resend through the account dashboard.  
  
Use Case Name: Administrator Login   
Use Case ID: UC-10   
Actors: Administrator, System   
  
Preconditions:   
1. The administrator has valid login credentials.   
2. The system is accessible and operational.   
3. The administrator has not previously logged in or their session has expired.   
  
Postconditions:   
1. The administrator is successfully logged in to the system.   
2. The administrator’s session is initiated.   
  
Main Flow:   
1. The administrator accesses the administrator login page of the system.   
2. The system displays the login form with fields for username and password.   
3. The administrator enters their username and password.   
4. The administrator submits the login form.   
5. The system validates the entered credentials against the administrator database.   
6. If the credentials are valid, the system authenticates the administrator and starts a session.   
7. The system redirects the administrator to the admin dashboard.   
  
Alternative Flow:   
1. If the administrator enters an incorrect username or password, the system displays an error message and prompts them to try again.   
2. If the administrator account is locked due to multiple failed login attempts, the system displays a message and suggests contacting support or waiting for a cooldown period.   
3. If the system fails to authenticate the administrator, it displays an error message and logs the event for security monitoring.   
4. If the administrator login page is inaccessible, the system displays an error message and suggests refreshing the page or contacting support.  
  
Use Case Name: Manage Inventory   
Use Case ID: UC-11   
Actors: Administrator, System, Inventory, Product, Category, Plugin   
Preconditions:   
1. The administrator is logged in to the system.   
2. The system is accessible and operational.   
3. The inventory contains at least one product.   
  
Postconditions:   
1. The inventory has been successfully updated (products added, removed, or modified).   
2. The system logs the inventory management activity.   
3. Product and category information is updated in the database.   
  
Main Flow:   
1. The administrator navigates to the inventory management section of the admin dashboard.   
2. The system displays the current inventory, including product details, stock levels, and categories.   
3. The administrator selects an action (e.g., add new product, update stock, remove product, or modify product details).   
4. The system opens the appropriate form or interface for the selected action.   
5. The administrator enters or modifies the required information (e.g., product name, description, price, stock quantity, or category).   
6. The administrator submits the changes.   
7. The system validates the input data (e.g., checks for valid product IDs, stock levels, and categories).   
8. The system updates the inventory and product databases accordingly.   
9. The system logs the inventory management action.   
10. The system displays a confirmation message to the administrator indicating the inventory has been updated.   
  
Alternative Flow:   
1. If the administrator enters invalid product information (e.g., duplicate product ID or invalid stock quantity), the system displays an error message and prompts for corrections.   
2. If the selected category does not exist, the system displays an error message and suggests creating a new category or selecting an existing one.   
3. If the system fails to update the inventory, it displays an error message and suggests trying again or contacting support.   
4. If the administrator attempts to remove a product that is currently in an active order, the system displays a warning message and prevents the removal.   
5. If the inventory management section is inaccessible, the system displays an error message and suggests refreshing the page or contacting support.  
  
Use Case Name: Manage Product Categories   
Use Case ID: UC-12   
Actors: Administrator, System, Category, Product, Plugin   
  
Preconditions:   
1. The administrator is logged in to the system.   
2. The system is accessible and operational.   
3. The category database is available.   
  
Postconditions:   
1. The product categories have been successfully added, removed, or modified.   
2. The system logs the category management activity.   
3. The product database is updated to reflect the new category associations.   
4. The system displays a confirmation message to the administrator.   
  
Main Flow:   
1. The administrator navigates to the "Manage Categories" section of the admin dashboard.   
2. The system displays a list of existing categories along with options to add, edit, or delete categories.   
3. The administrator selects an action (e.g., create a new category, update an existing category, or delete a category).   
4. The system opens the appropriate form for the selected action.   
5. The administrator enters or modifies the category information (e.g., category name, description, or associated products).   
6. The administrator submits the changes.   
7. The system validates the input data (e.g., checks for duplicate category names and valid product associations).   
8. The system updates the category database and modifies the product records accordingly.   
9. The system logs the category management action.   
10. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the administrator enters a duplicate category name, the system displays an error message and prompts for a unique name.   
2. If the system cannot validate the category information, it displays an error message and prevents the update.   
3. If the administrator attempts to delete a category that is associated with active products, the system displays a warning and suggests removing the association first.   
4. If the category database is not accessible, the system displays an error message and suggests trying again or contacting support.   
5. If the system fails to update the product records with the new category association, it displays an error message and logs the issue for review.  
  
Use Case Name: Update Product Information   
Use Case ID: UC-13   
Actors: Administrator, System, Product, Inventory, Category   
  
Preconditions:   
1. The administrator is logged in to the system.   
2. The product to be updated exists in the inventory.   
3. The system is accessible and operational.   
4. The category database is available.   
  
Postconditions:   
1. The product information is successfully updated in the database.   
2. The inventory reflects any changes in stock levels.   
3. The product category is updated if modified.   
4. The system logs the update action.   
5. The system displays a confirmation message to the administrator.   
  
Main Flow:   
1. The administrator navigates to the product management section of the admin dashboard.   
2. The system displays a list of products with their current details.   
3. The administrator selects a specific product and clicks the "Edit" button.   
4. The system opens the product update form with the current product information.   
5. The administrator modifies the product details (e.g., name, description, price, stock, or category).   
6. The administrator submits the updated form.   
7. The system validates the input data (e.g., checks for valid price, stock quantity, and category).   
8. The system updates the product information in the product database.   
9. If the category is modified, the system updates the category association for the product.   
10. The system updates the inventory to reflect any changes in stock levels.   
11. The system logs the product update activity.   
12. The system displays a confirmation message to the administrator.   
  
Alternative Flow:   
1. If the administrator enters an invalid price format, the system displays an error message and prompts for a valid price.   
2. If the administrator enters an invalid stock quantity (e.g., negative or non-numeric), the system displays an error message and prompts for a valid quantity.   
3. If the selected category is not valid or does not exist, the system displays an error message and suggests selecting a valid category.   
4. If the product database is not accessible, the system displays an error message and prevents the update.   
5. If the system fails to update the product information, it displays an error message and suggests trying again or contacting support.  
  
Use Case Name: Delete Product   
Use Case ID: UC-14   
Actors: Administrator, System, Product, Inventory, Category   
  
Preconditions:   
1. The administrator is logged in to the system.   
2. The product to be deleted exists in the inventory.   
3. The system is accessible and operational.   
4. The product is not currently in any active order.   
  
Postconditions:   
1. The product is successfully removed from the inventory.   
2. The product is no longer visible to customers in the catalog.   
3. The category associated with the product is updated to remove the reference.   
4. The system logs the deletion action.   
5. The system displays a confirmation message to the administrator.   
  
Main Flow:   
1. The administrator navigates to the product management section of the admin dashboard.   
2. The system displays a list of products with their details.   
3. The administrator selects a specific product and clicks the "Delete" button.   
4. The system confirms the deletion action by displaying a prompt.   
5. The administrator confirms the deletion.   
6. The system checks if the product is associated with any active orders.   
7. If no active orders are found, the system proceeds to remove the product from the inventory.   
8. The system updates the category database to remove the product reference.   
9. The system logs the deletion activity.   
10. The system displays a confirmation message to the administrator indicating the product has been deleted.   
  
Alternative Flow:   
1. If the product is associated with an active order, the system displays a warning message and prevents deletion.   
2. If the system fails to remove the product from the inventory, it displays an error message and suggests trying again or contacting support.   
3. If the system cannot update the category association, it displays an error message and logs the issue for review.   
4. If the administrator cancels the deletion prompt, the system returns to the product management page without making changes.   
5. If the product database is not accessible, the system displays an error message and prevents the deletion.  
  
Use Case Name: Manage Payment Methods   
Use Case ID: UC-15   
Actors: Customer, System, Payment   
Preconditions:   
1. The customer is logged in to the system.   
2. The system is accessible and operational.   
3. The customer has an existing account with at least one registered payment method or no payment methods.   
4. The payment gateway is functional.   
  
Postconditions:   
1. The customer’s payment methods are successfully added, updated, or removed.   
2. The system updates the customer’s account to reflect the changes in payment methods.   
3. The system logs the payment method management activity.   
4. The system displays a confirmation message to the customer.   
  
Main Flow:   
1. The customer accesses the payment method management section from their account dashboard.   
2. The system displays a list of the customer’s currently registered payment methods (if any) and provides options to add, edit, or remove a method.   
3. The customer selects an action (e.g., add a new payment method, edit an existing one, or delete a method).   
4. The system opens the appropriate form for the selected action.   
5. The customer enters or modifies payment method details (e.g., card number, expiration date, billing address).   
6. The customer submits the form.   
7. The system validates the payment method information (e.g., checks for valid card format and expiration date).   
8. If the action is to add a new method, the system registers the new payment method in the customer’s account.   
9. If the action is to edit, the system updates the existing payment method with the provided information.   
10. If the action is to delete, the system removes the selected payment method from the customer’s account.   
11. The system logs the action and displays a confirmation message to the customer.   
  
Alternative Flow:   
1. If the customer enters an invalid card number or expiration date, the system displays an error message and prompts for correction.   
2. If the customer attempts to delete the only active payment method, the system displays a warning message and suggests keeping at least one valid method.   
3. If the system cannot connect to the payment gateway, it displays an error message and prevents the addition or update of the payment method.   
4. If the system fails to update the customer’s account, it displays an error message and suggests trying again or contacting support.   
5. If the customer cancels the action during the process, the system returns to the payment method management page without making any changes.  
  
Use Case Name: View Purchase History   
Use Case ID: UC-16   
Actors: Customer, System, Order, Product, Email Confirmation   
Preconditions:   
1. The customer is logged in to the system.   
2. The customer has at least one completed order in the order database.   
3. The system is accessible and operational.   
4. The purchase history feature is enabled in the system.   
  
Postconditions:   
1. The customer has viewed their purchase history.   
2. The system has displayed the list of previous orders with relevant details.   
3. The system logs the customer’s access to the purchase history.   
4. The customer can view order details, including products, quantities, and total price.   
  
Main Flow:   
1. The customer navigates to the "Purchase History" section from their account dashboard.   
2. The system retrieves the customer's order history from the order database.   
3. The system displays a list of previous orders with details such as order number, date, total price, and order status.   
4. The customer selects a specific order to view more details.   
5. The system displays the order details, including product list, quantities, and price breakdown.   
6. The customer reviews the information.   
7. The system logs the access to the purchase history.   
  
Alternative Flow:   
1. If the customer has no completed orders, the system displays a message indicating that no purchase history is available.   
2. If the system fails to retrieve the order history, it displays an error message and suggests trying again later.   
3. If the selected order does not exist or is invalid, the system displays an error message and returns to the purchase history list.   
4. If the purchase history section is inaccessible, the system displays an error message and suggests refreshing the page or contacting support.   
5. If the system is unable to display product details for an order, it shows a message indicating the issue and allows the customer to contact support for assistance.  
  
Use Case Name: Manage Plugins   
Use Case ID: UC-17   
Actors: Administrator, System, Plugin   
Preconditions:   
1. The administrator is logged in to the system.   
2. The system is accessible and operational.   
3. The plugin management feature is enabled in the system.   
4. At least one plugin is available in the plugin repository or the system supports plugin installation.   
  
Postconditions:   
1. Plugins have been successfully added, removed, or updated in the system.   
2. The system logs the plugin management activity.   
3. The system reflects the updated plugin status in the admin dashboard.   
4. The administrator receives a confirmation message for the plugin action.   
  
Main Flow:   
1. The administrator navigates to the "Manage Plugins" section of the admin dashboard.   
2. The system displays a list of currently installed plugins with options to activate, deactivate, update, or delete them.   
3. The administrator selects an action (e.g., install a new plugin, update an existing one, or remove a plugin).   
4. The system opens the appropriate form or interface for the selected action.   
5. If installing or updating, the system connects to the plugin repository to fetch the latest version or new plugin.   
6. The administrator confirms the action (e.g., selects the plugin file or confirms the update).   
7. The system performs the action (installation, update, or deletion) and applies the necessary changes.   
8. The system updates the plugin status in the database and logs the activity.   
9. The system displays a confirmation message to the administrator indicating the result of the action.   
  
Alternative Flow:   
1. If the selected plugin is already installed and up-to-date, the system displays a message and prevents redundant updates.   
2. If the system cannot connect to the plugin repository, it displays an error message and suggests checking the network connection or trying again later.   
3. If the plugin file is invalid or corrupted, the system displays an error message and prevents the installation or update.   
4. If the administrator attempts to remove a plugin that is in use by the system, the system displays a warning message and suggests disabling it first or resolving dependencies.   
5. If the system fails to update the plugin status in the database, it displays an error message and suggests trying again or contacting support.  
  
Use Case Name: View Email Confirmation   
Use Case ID: UC-18   
Actors: Customer, System, Email Confirmation   
  
Preconditions:   
1. The customer has received an email confirmation.   
2. The system has stored the email confirmation content in the database.   
3. The system is accessible and operational.   
4. The customer has access to their email inbox.   
  
Postconditions:   
1. The customer has viewed the email confirmation content.   
2. The system has recorded the customer’s access to the email confirmation.   
3. The customer is informed about the relevant details in the email.   
  
Main Flow:   
1. The customer opens the email confirmation received in their inbox.   
2. The customer clicks on the link or opens the content of the email.   
3. The system retrieves the email confirmation content from the database.   
4. The system displays the email confirmation details to the customer, including the purpose of the email, relevant actions, and additional information.   
5. The customer reviews the content and takes the required action (e.g., verify account, view order details).   
6. The system logs the customer’s access to the email confirmation.   
  
Alternative Flow:   
1. If the email confirmation link is expired, the system displays a message and prompts the customer to request a new confirmation.   
2. If the email content cannot be retrieved, the system displays an error message and suggests checking the email or contacting support.   
3. If the customer is not logged in and attempts to access the email confirmation link, the system may prompt them to log in or provide additional verification.   
4. If the system fails to display the email content, it logs the issue and shows an error message to the customer.   
5. If the email confirmation is not found in the database, the system informs the customer and suggests verifying the link or contacting support.  
  
Use Case Name: Manage Cart Item   
Use Case ID: UC-19   
Actors: Customer, System, Shopping Cart, Product, Inventory   
  
Preconditions:   
1. The customer is logged in to the system or is a guest user.   
2. The shopping cart contains at least one item.   
3. The system is accessible and operational.   
4. The product associated with the cart item exists in the inventory.   
  
Postconditions:   
1. The cart item is successfully modified or removed from the shopping cart.   
2. The shopping cart is updated to reflect the changes.   
3. The inventory records the updated stock level based on the cart item changes.   
4. The system provides confirmation of the cart item management action to the customer.   
  
Main Flow:   
1. The customer accesses the shopping cart page.   
2. The system displays the current items in the customer’s shopping cart, including product details and quantities.   
3. The customer selects a specific cart item and chooses an action (e.g., modify the quantity or remove the item).   
4. The system validates the requested action and checks the product’s availability in the inventory.   
5. If the action is to modify the quantity, the customer enters the new quantity.   
6. The system updates the cart item with the new quantity or removes it from the cart.   
7. The system updates the shopping cart to reflect the changes.   
8. The system updates the inventory to adjust the stock level accordingly.   
9. The system provides a confirmation message that the cart item has been successfully managed.   
  
Alternative Flow:   
1. If the customer attempts to modify the quantity to a value higher than the available stock, the system displays an error message and limits the quantity to the available stock.   
2. If the system fails to validate the requested action, it displays an error message and prevents the cart item modification or removal.   
3. If the inventory is not accessible, the system prevents the update and displays an error message.   
4. If the customer is not logged in and tries to manage a cart item, the system may prompt them to log in or continue as a guest.   
5. If the shopping cart is not accessible, the system displays an error message and suggests refreshing the page or contacting support.  
  
Use Case Name: Manage Order Item   
Use Case ID: UC-20   
Actors: Administrator, System, Order, Product, Order Item, Inventory   
  
Preconditions:   
1. The administrator is logged in to the system.   
2. The system is accessible and operational.   
3. An order exists in the order database.   
4. The order contains at least one item.   
5. The product associated with the order item exists in the inventory.   
  
Postconditions:   
1. The order item is successfully modified or removed from the order.   
2. The inventory is updated to reflect the changes in stock levels.   
3. The order database is updated to reflect the modified or removed order item.   
4. The system logs the order item management action.   
5. The system displays a confirmation message to the administrator.   
  
Main Flow:   
1. The administrator navigates to the order management section of the admin dashboard.   
2. The system displays a list of orders with their details, including the associated items.   
3. The administrator selects a specific order and views its items.   
4. The administrator selects an order item and chooses an action (e.g., modify the quantity or remove the item from the order).   
5. The system validates the requested action and checks the product’s availability in the inventory.   
6. If the action is to modify the quantity, the administrator enters the new quantity.   
7. The system updates the order item with the new quantity or removes it from the order.   
8. The system updates the order database to reflect the changes.   
9. The system updates the inventory to adjust the stock level accordingly.   
10. The system logs the order item management action.   
11. The system displays a confirmation message to the administrator indicating the result of the action.   
  
Alternative Flow:   
1. If the administrator attempts to modify the quantity to a value higher than the available stock, the system displays an error message and limits the quantity to the available stock.   
2. If the system fails to validate the requested action, it displays an error message and prevents the modification or removal of the order item.   
3. If the inventory is not accessible, the system prevents the update and displays an error message.   
4. If the administrator attempts to modify an order item that is already shipped, the system displays a warning and suggests that changes can only be made to pending orders.   
5. If the system fails to update the order database, it displays an error message and suggests trying again or contacting support.  
  
Use Case Name: Manage Payment Method   
Use Case ID: UC-21   
Actors: Administrator, System, Payment Method   
  
Preconditions:   
1. The administrator is logged in to the system.   
2. The system is accessible and operational.   
3. The payment method database is available.   
4. The system supports the addition, modification, or removal of payment methods.   
  
Postconditions:   
1. The payment method configuration is successfully updated in the system.   
2. The system reflects the changes in available payment options.   
3. The system logs the payment method management activity.   
4. The administrator receives a confirmation message for the action performed.   
  
Main Flow:   
1. The administrator navigates to the "Manage Payment Methods" section of the admin dashboard.   
2. The system displays a list of currently supported payment methods with options to add, edit, or remove them.   
3. The administrator selects an action (e.g., add a new payment method, modify an existing one, or remove a method).   
4. The system opens the appropriate form or interface for the selected action.   
5. If adding a new payment method, the administrator enters the method details (e.g., name, description, and configuration parameters).   
6. If editing, the administrator modifies the existing payment method details.   
7. The administrator submits the changes.   
8. The system validates the input data (e.g., ensures the method name is unique and the configuration is correct).   
9. The system updates the payment method database accordingly.   
10. The system logs the payment method management action.   
11. The system displays a confirmation message to the administrator indicating the result of the action.   
  
Alternative Flow:   
1. If the administrator enters a duplicate payment method name, the system displays an error message and prompts for a unique name.   
2. If the system cannot validate the configuration parameters for a new or modified payment method, it displays an error message and prevents the update.   
3. If the administrator attempts to remove a payment method that is currently in use by active orders, the system displays a warning and suggests disabling it first or resolving dependencies.   
4. If the payment method database is not accessible, the system displays an error message and prevents the action.   
5. If the system fails to update the payment method configuration, it displays an error message and suggests trying again or contacting support.