项目文档

# Functional Requirement

1. Functional Requirements  
  
1.1 Customer Account Creation Function   
Function ID: FR-01   
Description: The system allows a customer to create a new user account by providing personal information such as name, email, password, and contact details. The system must validate the email format and password strength, send a confirmation email, and activate the account after confirmation.   
Input: Name, Email, Password, Address, PhoneNumber   
Output: New UserAccount with status "Pending" or "Active", Confirmation Email sent to the provided email address   
  
1.2 User Login Function   
Function ID: FR-02   
Description: The system authenticates a user by verifying their email and password against the UserAccount database. Upon successful login, a UserSession is created and logged.   
Input: Email, Password   
Output: Active UserSession, User redirected to appropriate dashboard   
  
1.3 User Logout Function   
Function ID: FR-03   
Description: The system terminates the current UserSession and logs the logout event.   
Input: Current UserSession   
Output: Inactive UserSession, User redirected to login or home page   
  
1.4 View Personal Information Function   
Function ID: FR-04   
Description: The system retrieves and displays the personal information of the logged-in user from the UserAccount entity.   
Input: UserID (from UserSession)   
Output: Displayed Name, Email, Address, PhoneNumber, Role, Account Status   
  
1.5 Update Payment Information Function   
Function ID: FR-05   
Description: The system allows the user to edit their PaymentInformation, such as card number, expiration date, and billing address. Validation of the card and expiration date is required.   
Input: PaymentID, CardNumber, ExpiryDate, CVV, BillingAddress   
Output: Updated PaymentInformation, System logs the update   
  
1.6 View Purchase History Function   
Function ID: FR-06   
Description: The system retrieves and displays the user's Order records from the Order database, including product details, total price, and order status.   
Input: UserID (from UserSession)   
Output: List of Orders with associated Product, TotalAmount, OrderDate, Status, and PaymentInformation   
  
1.7 Add Product to Cart Function   
Function ID: FR-07   
Description: The system allows a customer to add a selected product to their ShoppingCart. The system checks product availability from the Inventory and updates the CartItem and CartSummary.   
Input: ProductID, CartID, Quantity   
Output: Updated ShoppingCart, Updated CartItem, Updated CartSummary   
  
1.8 Modify Cart Contents Function   
Function ID: FR-08   
Description: The system allows a customer to update the quantity of a CartItem or remove it from the ShoppingCart. The system recalculates the CartSummary and updates the Inventory if necessary.   
Input: CartItemID, NewQuantity, Action (update/remove)   
Output: Updated CartItem, Updated CartSummary, Updated InventoryQuantityUpdateLog if applicable   
  
1.9 View Cart Summary Function   
Function ID: FR-09   
Description: The system retrieves and displays the total price, total quantity, and list of CartItems for the current ShoppingCart.   
Input: CartID   
Output: Displayed CartSummary including TotalItems and TotalPrice, List of CartItems   
  
1.10 Checkout Order Function   
Function ID: FR-10   
Description: The system processes the customer’s ShoppingCart by validating payment information, reducing inventory, creating an Order, and sending an OrderConfirmationEmail.   
Input: CartID, PaymentID, ShippingMethod   
Output: Created Order, Updated Inventory, Sent OrderConfirmationEmail, Cleared ShoppingCart   
  
1.11 Confirm Order Function   
Function ID: FR-11   
Description: The system confirms the customer’s Order after payment validation. Inventory is updated, and an OrderConfirmationEmail is sent.   
Input: OrderID, PaymentID   
Output: Confirmed Order, Updated Inventory, Sent OrderConfirmationEmail   
  
1.12 Receive Order Confirmation Email Function   
Function ID: FR-12   
Description: The system sends an OrderConfirmationEmail to the customer’s registered email address after an order is confirmed.   
Input: OrderID, CustomerEmail   
Output: Sent OrderConfirmationEmail with Order details, Log of email delivery   
  
1.13 Manage Inventory Function   
Function ID: FR-13   
Description: The system allows the administrator to view, update, and log changes to the Inventory. Product availability is updated accordingly.   
Input: ProductID, NewQuantity, InventoryID   
Output: Updated Inventory, Updated Product availability status, InventoryQuantityUpdateLog   
  
1.14 Add Product Function   
Function ID: FR-14   
Description: The system allows the administrator to add a new Product to the database, including name, price, description, category, and inventory quantity.   
Input: Name, Description, Price, CategoryID, InventoryQuantity, ImageURL   
Output: New Product record, Updated Inventory, Confirmation message   
  
1.15 Update Product Details Function   
Function ID: FR-15   
Description: The system allows the administrator to modify Product details such as name, price, description, or category. Inventory is updated if necessary.   
Input: ProductID, NewName, NewDescription, NewPrice, NewCategoryID, NewInventoryQuantity   
Output: Updated Product record, Updated Inventory, Log of the update   
  
1.16 Delete Product Function   
Function ID: FR-16   
Description: The system allows the administrator to delete a Product and its associated Inventory entry. Products in ShoppingCarts or Orders must be handled before deletion.   
Input: ProductID   
Output: Deleted Product, Updated or deleted Inventory entry, Log of the deletion   
  
1.17 Create Category Function   
Function ID: FR-17   
Description: The system allows the administrator to create a new Category with a name and description.   
Input: Name, Description   
Output: New Category record, Updated product catalog display   
  
1.18 Update Category Function   
Function ID: FR-18   
Description: The system allows the administrator to update the name or description of an existing Category.   
Input: CategoryID, NewName, NewDescription   
Output: Updated Category record, Updated product catalog display   
  
1.19 Delete Category Function   
Function ID: FR-19   
Description: The system allows the administrator to delete a Category, and optionally reassign or mark products as uncategorized.   
Input: CategoryID   
Output: Deleted Category, Updated Product records, Log of the deletion   
  
1.20 View Product List Function   
Function ID: FR-20   
Description: The system retrieves and displays a list of available Products with details such as name, price, description, category, and availability.   
Input: Optional filters (e.g., category, price range, availability)   
Output: Displayed list of Products, User can select to view or add to cart   
  
1.21 Manage Orders Function   
Function ID: FR-21   
Description: The system allows the user to view, update, or cancel their Orders. The system logs the activity and updates the Order status.   
Input: OrderID, Action (update/cancel), NewStatus   
Output: Updated Order record, Log of the action, Confirmation message   
  
1.22 View Order Details Function   
Function ID: FR-22   
Description: The system retrieves and displays detailed information about a specific Order, including product list, total price, and payment and shipping status.   
Input: OrderID   
Output: Displayed Order details, Log of access   
  
1.23 Process Payment Function   
Function ID: FR-23   
Description: The system processes the customer’s payment using the selected PaymentInformation and updates the Order and Inventory accordingly.   
Input: OrderID, PaymentID   
Output: Updated Order status (Paid), Updated Inventory, Sent OrderConfirmationEmail, Log of payment processing   
  
1.24 View Payment Information Function   
Function ID: FR-24   
Description: The system retrieves and displays the PaymentInformation for the logged-in user.   
Input: UserID (from UserSession)   
Output: Displayed Payment details including CardNumber, ExpiryDate, BillingAddress   
  
1.25 Install Plugin Function   
Function ID: FR-25   
Description: The system allows the administrator to install a new Plugin, validate it, and activate it.   
Input: Plugin file or PluginID   
Output: Installed Plugin, Updated system configuration, Log of installation   
  
1.26 View Plugin Documentation Function   
Function ID: FR-26   
Description: The system allows the administrator to view documentation related to a Plugin.   
Input: PluginID   
Output: Displayed Plugin documentation, Log of access   
  
1.27 Manage Cart Item Function   
Function ID: FR-28   
Description: The system allows the customer to manage individual CartItems, including updating quantity or removing them.   
Input: CartItemID, NewQuantity, Action (update/remove)   
Output: Updated CartItem, Updated CartSummary, Log of modification if applicable

# External Description

# 2. External Interfaces  
  
This section describes the external interfaces that the system interacts with in order to fulfill its functional requirements. These interfaces include user interfaces, hardware interfaces, software interfaces (including databases and third-party services), and communication interfaces such as email notifications.  
  
---  
  
## 2.1 User Interface  
  
### 2.1.1 User Account Creation UI   
\*\*Description:\*\* A form-based interface where users can input their personal information (name, email, password, address, phone number) to create a new account.   
\*\*Inputs:\*\* Name, Email, Password, Address, PhoneNumber   
\*\*Outputs:\*\* Display of confirmation message or error messages; redirection to login page or dashboard upon success.  
  
### 2.1.2 User Login UI   
\*\*Description:\*\* A login screen where users enter their email and password for authentication.   
\*\*Inputs:\*\* Email, Password   
\*\*Outputs:\*\* Redirect to appropriate dashboard if successful, or display an error message if authentication fails.  
  
### 2.1.3 User Logout UI   
\*\*Description:\*\* A button or link that allows the user to log out of their session.   
\*\*Inputs:\*\* Current UserSession   
\*\*Outputs:\*\* Inactive session status; redirect to login or home page.  
  
### 2.1.4 View Personal Information UI   
\*\*Description:\*\* A profile page displaying the user’s personal details (name, email, address, phone number, role, account status).   
\*\*Inputs:\*\* UserID from current session   
\*\*Outputs:\*\* Displayed user information in a structured format.  
  
### 2.1.5 Update Payment Information UI   
\*\*Description:\*\* A form allowing the user to update their payment details (card number, expiration date, CVV, billing address).   
\*\*Inputs:\*\* PaymentID, CardNumber, ExpiryDate, CVV, BillingAddress   
\*\*Outputs:\*\* Confirmation message on successful update; error message on validation failure.  
  
### 2.1.6 View Purchase History UI   
\*\*Description:\*\* A page listing all previous orders made by the user, including product details, total price, order status, and payment information.   
\*\*Inputs:\*\* UserID from current session   
\*\*Outputs:\*\* List of Orders with associated Product, TotalAmount, OrderDate, Status, and PaymentInformation.  
  
### 2.1.7 Add Product to Cart UI   
\*\*Description:\*\* A button or interface allowing the user to add a selected product to their shopping cart.   
\*\*Inputs:\*\* ProductID, CartID, Quantity   
\*\*Outputs:\*\* Updated ShoppingCart, CartItem, and CartSummary displayed on the screen.  
  
### 2.1.8 Modify Cart Contents UI   
\*\*Description:\*\* An interface allowing the user to adjust quantities or remove items from the cart.   
\*\*Inputs:\*\* CartItemID, NewQuantity, Action (update/remove)   
\*\*Outputs:\*\* Updated CartItem, CartSummary, and InventoryQuantityUpdateLog if applicable.  
  
### 2.1.9 View Cart Summary UI   
\*\*Description:\*\* A summary page showing the total price, quantity, and list of items in the user's current cart.   
\*\*Inputs:\*\* CartID   
\*\*Outputs:\*\* Display of CartSummary with TotalItems, TotalPrice, and list of CartItems.  
  
### 2.1.10 Checkout Order UI   
\*\*Description:\*\* A checkout page where the user selects a payment method and shipping option before finalizing the purchase.   
\*\*Inputs:\*\* CartID, PaymentID, ShippingMethod   
\*\*Outputs:\*\* Created Order, updated Inventory, sent OrderConfirmationEmail, and cleared ShoppingCart.  
  
### 2.1.11 Confirm Order UI   
\*\*Description:\*\* A confirmation screen or backend process confirming the order after payment validation.   
\*\*Inputs:\*\* OrderID, PaymentID   
\*\*Outputs:\*\* Confirmed Order, updated Inventory, and sent OrderConfirmationEmail.  
  
### 2.1.12 Receive Order Confirmation Email UI   
\*\*Description:\*\* A system-generated email sent to the user after the order is confirmed.   
\*\*Inputs:\*\* OrderID, CustomerEmail   
\*\*Outputs:\*\* Sent OrderConfirmationEmail with Order details and a log entry of delivery.  
  
### 2.1.13 Manage Inventory UI   
\*\*Description:\*\* An administrative interface for viewing, updating, and logging changes to inventory levels.   
\*\*Inputs:\*\* ProductID, NewQuantity, InventoryID   
\*\*Outputs:\*\* Updated Inventory, updated Product availability status, and InventoryQuantityUpdateLog.  
  
### 2.1.14 Add Product UI   
\*\*Description:\*\* An administrative interface for adding new products to the database.   
\*\*Inputs:\*\* Name, Description, Price, CategoryID, InventoryQuantity, ImageURL   
\*\*Outputs:\*\* New Product record, updated Inventory, and confirmation message.  
  
### 2.1.15 Update Product Details UI   
\*\*Description:\*\* An administrative interface for modifying product details.   
\*\*Inputs:\*\* ProductID, NewName, NewDescription, NewPrice, NewCategoryID, NewInventoryQuantity   
\*\*Outputs:\*\* Updated Product record, updated Inventory, and log of the update.  
  
### 2.1.16 Delete Product UI   
\*\*Description:\*\* An administrative interface for deleting a product and its inventory entry.   
\*\*Inputs:\*\* ProductID   
\*\*Outputs:\*\* Deleted Product, updated or deleted Inventory entry, and log of deletion.  
  
### 2.1.17 Create Category UI   
\*\*Description:\*\* An administrative interface for creating new categories.   
\*\*Inputs:\*\* Name, Description   
\*\*Outputs:\*\* New Category record and updated product catalog display.  
  
### 2.1.18 Update Category UI   
\*\*Description:\*\* An administrative interface for updating category names or descriptions.   
\*\*Inputs:\*\* CategoryID, NewName, NewDescription   
\*\*Outputs:\*\* Updated Category record and updated product catalog display.  
  
### 2.1.19 Delete Category UI   
\*\*Description:\*\* An administrative interface for deleting a category and optionally reassigning or marking products as uncategorized.   
\*\*Inputs:\*\* CategoryID   
\*\*Outputs:\*\* Deleted Category, updated Product records, and log of deletion.  
  
### 2.1.20 View Product List UI   
\*\*Description:\*\* A product listing page with optional filters (category, price range, availability).   
\*\*Inputs:\*\* Optional filters (e.g., category, price range, availability)   
\*\*Outputs:\*\* Displayed list of Products with name, price, description, category, and availability; ability to view or add to cart.  
  
### 2.1.21 Manage Orders UI   
\*\*Description:\*\* A user-facing interface for viewing, updating, or canceling orders.   
\*\*Inputs:\*\* OrderID, Action (update/cancel), NewStatus   
\*\*Outputs:\*\* Updated Order record, log of action, and confirmation message.  
  
### 2.1.22 View Order Details UI   
\*\*Description:\*\* A detailed view of a specific order, including product list, total price, and payment/shipping status.   
\*\*Inputs:\*\* OrderID   
\*\*Outputs:\*\* Displayed Order details and log of access.  
  
### 2.1.23 Process Payment UI   
\*\*Description:\*\* A payment processing interface that validates and processes customer payment.   
\*\*Inputs:\*\* OrderID, PaymentID   
\*\*Outputs:\*\* Updated Order status (Paid), updated Inventory, sent OrderConfirmationEmail, and log of payment processing.  
  
### 2.1.24 View Payment Information UI   
\*\*Description:\*\* A user-facing interface that displays the user’s payment details.   
\*\*Inputs:\*\* UserID from current session   
\*\*Outputs:\*\* Displayed Payment details including CardNumber, ExpiryDate, BillingAddress.  
  
### 2.1.25 Install Plugin UI   
\*\*Description:\*\* An administrative interface for installing plugins.   
\*\*Inputs:\*\* Plugin file or PluginID   
\*\*Outputs:\*\* Installed Plugin, updated system configuration, and log of installation.  
  
### 2.1.26 View Plugin Documentation UI   
\*\*Description:\*\* An administrative interface for viewing plugin documentation.   
\*\*Inputs:\*\* PluginID   
\*\*Outputs:\*\* Displayed Plugin documentation and log of access.  
  
### 2.1.27 Manage Cart Item UI   
\*\*Description:\*\* A user-facing interface for managing individual cart items (updating quantity or removing).   
\*\*Inputs:\*\* CartItemID, NewQuantity, Action (update/remove)   
\*\*Outputs:\*\* Updated CartItem, updated CartSummary, and log of modification if applicable.  
  
---  
  
## 2.2 Hardware Interface  
  
There are no direct hardware interfaces required for this system. All operations are conducted through software components and web-based interactions. However, the system may be deployed on servers or cloud infrastructure that require standard hardware support, such as:  
  
- \*\*Server Hardware:\*\* Standard server configurations (CPU, RAM, storage).  
- \*\*Network Equipment:\*\* Routers, switches, and firewalls for secure data transmission.  
- \*\*Client Devices:\*\* Web browsers and mobile devices used by customers and administrators.  
  
These hardware components are considered part of the deployment environment and not directly interfaced with by the application logic.  
  
---  
  
## 2.3 Software Interface  
  
### 2.3.1 Database Interface   
\*\*Description:\*\* The system interacts with one or more relational databases to store and retrieve data related to Users, Orders, Products, Inventory, PaymentInformation, and Logs.  
  
\*\*Data Sources:\*\*  
- \*\*UserAccount\*\* – Stores user registration details, roles, and account status.  
- \*\*PaymentInformation\*\* – Stores credit card details, billing addresses, etc.  
- \*\*Order\*\* – Tracks order details, status, and payment information.  
- \*\*Product\*\* – Contains product metadata like name, price, description, and category.  
- \*\*Inventory\*\* – Tracks available stock for each product.  
- \*\*Cart\*\* – Manages user shopping carts and cart items.  
- \*\*Category\*\* – Defines product categories.  
- \*\*Plugin\*\* – Stores plugin configurations and documentation.  
- \*\*Log\*\* – Tracks system events, updates, and user actions.  
  
\*\*Interaction Method:\*\* SQL queries via ORM or direct SQL execution for CRUD operations.  
  
### 2.3.2 Third-Party API Interfaces   
\*\*Description:\*\* The system may interact with external APIs for payment processing, email notification, or other functionalities not implemented internally.  
  
\*\*Examples:\*\*  
- \*\*Payment Gateway API\*\* – For validating and processing payments (e.g., Stripe, PayPal).  
- \*\*Email Service API\*\* – For sending confirmation emails (e.g., SendGrid, SMTP).  
  
\*\*Interaction Method:\*\* RESTful API calls using HTTP verbs (GET, POST, PUT, DELETE) with JSON payload formats.  
  
### 2.3.3 Plugin Management Interface   
\*\*Description:\*\* The system provides an interface for installing, configuring, and managing plugins. Plugins may extend functionality or modify existing behavior.  
  
\*\*Interaction Method:\*\* File upload or PluginID-based activation, followed by validation and integration into the system.  
  
---  
  
## 2.4 Communication Interface  
  
### 2.4.1 Email Notification Interface   
\*\*Description:\*\* The system sends email notifications to users regarding account creation, order confirmation, and other relevant events.  
  
\*\*Interaction Method:\*\*  
- \*\*Input:\*\* OrderID, CustomerEmail  
- \*\*Output:\*\* Sent OrderConfirmationEmail with order details.  
- \*\*Communication Protocol:\*\* SMTP or use of third-party email service API (e.g., SendGrid).  
  
### 2.4.2 Network Communication Interface   
\*\*Description:\*\* The system communicates over HTTP/HTTPS protocols for web traffic between the client browser and the server backend.  
  
\*\*Interaction Method:\*\*  
- \*\*Input:\*\* HTTP requests from clients (e.g., GET, POST)  
- \*\*Output:\*\* JSON/XML responses or HTML content depending on the request type.  
  
### 2.4.3 Logging Interface   
\*\*Description:\*\* The system logs various events (user actions, system errors, database updates, etc.) to ensure traceability and auditing.  
  
\*\*Interaction Method:\*\*  
- \*\*Input:\*\* LogEvent (type, timestamp, user, action, details)  
- \*\*Output:\*\* Stored in a logging database or file system.  
  
---  
  
## Summary  
  
The system relies on a combination of user interfaces, software interfaces (databases and third-party APIs), and communication interfaces (email and network protocols) to fulfill its functional requirements. Each external interface has been clearly defined with its inputs, outputs, and interaction methods to guide development and implementation.

# Use Case

Use Case Name: Create Customer Account   
Use Case ID: UC-01   
Actors: Customer, Administrator   
Preconditions:   
1. The system must be operational.   
2. The customer must have access to the system's registration interface.   
3. The administrator must have access to the account management interface.   
  
Postconditions:   
1. A new customer account is successfully created and stored in the system.   
2. The customer receives a confirmation email or message.   
3. The administrator can view and manage the newly created account.   
  
Main Flow:   
1. The customer navigates to the registration page.   
2. The customer enters their personal information (e.g., name, email, password, and contact details).   
3. The system validates the input data (e.g., checks for valid email format and password strength).   
4. If validation is successful, the system creates a new user account and assigns it a unique ID.   
5. The system sends a confirmation email to the customer's provided email address.   
6. The customer confirms their account via the link in the email.   
7. The system updates the account status to "active."   
8. The administrator can view the new account in the account management section.   
  
Alternative Flow:   
1. If the customer enters an invalid email format, the system displays an error message and prompts the customer to correct it.   
2. If the customer enters a password that does not meet the strength requirements, the system displays an error message and prompts the customer to choose a stronger password.   
3. If the customer does not confirm their account within the specified time frame, the system automatically deactivates the account.   
4. If the customer already has an account with the same email, the system displays an error message indicating the account already exists and provides options to recover the account or log in.  
  
Use Case Name: Login User   
Use Case ID: UC-02   
Actors: Customer, Administrator   
Preconditions:   
1. The system must be operational.   
2. The user must have a valid and active account (Customer or Administrator).   
3. The user must have access to the login interface.   
  
Postconditions:   
1. The user is authenticated and logged into the system.   
2. The user is redirected to their appropriate dashboard or homepage.   
3. The system logs the login activity for security and auditing purposes.   
  
Main Flow:   
1. The user navigates to the login page.   
2. The user enters their email and password associated with their account.   
3. The system validates the entered credentials against the stored User Account information.   
4. If the credentials are valid, the system grants access and displays the user's homepage.   
5. The system logs the successful login attempt.   
  
Alternative Flow:   
1. If the user enters an incorrect email or password, the system displays an error message and prompts the user to try again.   
2. If the user enters the wrong password multiple times, the system may lock the account temporarily and notify the user via email.   
3. If the user account is inactive or suspended, the system displays an error message and prevents login.   
4. If the user forgets their password, they can click on the "Forgot Password" link, triggering a password reset process via email.  
  
Use Case Name: Logout User   
Use Case ID: UC-03   
Actors: Customer, Administrator   
Preconditions:   
1. The system must be operational.   
2. The user (Customer or Administrator) must be logged in.   
3. The user must have access to the logout functionality.   
  
Postconditions:   
1. The user is successfully logged out of the system.   
2. The session associated with the user is terminated.   
3. The user is redirected to the login or home page.   
4. The system logs the logout activity for security and auditing purposes.   
  
Main Flow:   
1. The user navigates to the account or session management section.   
2. The user clicks on the "Logout" button or link.   
3. The system prompts the user to confirm the logout action.   
4. The user confirms the logout.   
5. The system terminates the current session and clears session-related data.   
6. The user is redirected to the login page or home page.   
7. The system logs the logout event.   
  
Alternative Flow:   
1. If the user cancels the logout confirmation, the system retains the current session and does not log the user out.   
2. If the system fails to terminate the session due to technical issues, the system displays an error message and allows the user to manually close the browser or session.   
3. If the user closes the browser or navigates away without using the logout button, the session may still be active until it times out based on the system's session expiration policy.  
  
Use Case Name: View Personal Information   
Use Case ID: UC-04   
Actors: Customer, Administrator   
Preconditions:   
1. The system must be operational.   
2. The user (Customer or Administrator) must be logged in.   
3. The user must have access to the personal information section.   
  
Postconditions:   
1. The user's personal information is displayed on the screen.   
2. The user can view details such as name, email, contact information, and account status.   
3. No changes are made to the user account unless an edit action is initiated.   
  
Main Flow:   
1. The user navigates to the "View Personal Information" section from their dashboard.   
2. The system retrieves the user's account information from the User Account database.   
3. The system displays the user's personal information in a readable format.   
4. The user reviews the displayed information.   
  
Alternative Flow:   
1. If the system fails to retrieve the user's account information, it displays an error message and allows the user to retry or contact support.   
2. If the user does not have permission to view certain information (e.g., administrator-only fields), the system hides or does not display those fields.   
3. If the user's account is deleted or suspended, the system displays an error message and prevents access to the personal information section.  
  
Use Case Name: Update Payment Information   
Use Case ID: UC-05   
Actors: Customer, Administrator   
Preconditions:   
1. The system must be operational.   
2. The user (Customer or Administrator) must be logged in.   
3. The user must have access to the payment information management section.   
4. The user must have an existing payment information record in the Payment Information database.   
  
Postconditions:   
1. The user's payment information is successfully updated and stored in the system.   
2. The system confirms the update to the user.   
3. If initiated by the administrator, the updated information is marked as reviewed or verified.   
4. The system logs the update activity for auditing purposes.   
  
Main Flow:   
1. The user navigates to the "Payment Information" section from their account settings.   
2. The system displays the current payment information (e.g., card number, expiration date, billing address).   
3. The user edits the payment information as needed.   
4. The user clicks on the "Save Changes" button.   
5. The system validates the updated information (e.g., checks for valid card number and expiration date).   
6. If validation is successful, the system updates the Payment Information database with the new data.   
7. The system sends a confirmation message to the user indicating the update was successful.   
  
Alternative Flow:   
1. If the user enters an invalid card number, the system displays an error message and prompts the user to correct it.   
2. If the user enters an invalid expiration date, the system displays an error message and prompts the user to enter a valid date.   
3. If the system fails to update the Payment Information due to technical issues, it displays an error message and allows the user to retry or contact support.   
4. If the user does not have permission to update payment information (e.g., a regular customer attempting to edit another user's data), the system displays an access denied message and terminates the action.   
5. If the user cancels the update process, the system retains the original payment information and does not apply any changes.  
  
Use Case Name: View Purchase History   
Use Case ID: UC-06   
Actors: Customer, Administrator   
Preconditions:   
1. The system must be operational.   
2. The user (Customer or Administrator) must be logged in.   
3. The user must have at least one order in the Order database.   
4. The user must have access to the purchase history section.   
  
Postconditions:   
1. The user's purchase history is displayed on the screen.   
2. The user can view details of each order, such as product names, quantities, order dates, and total prices.   
3. If initiated by the administrator, additional order details (e.g., payment status, shipping status) may be displayed.   
4. No changes are made to the order data unless an edit or cancellation action is initiated.   
  
Main Flow:   
1. The user navigates to the "Purchase History" section from their account dashboard.   
2. The system retrieves the user's order records from the Order database.   
3. The system displays the list of orders in a clear and organized format.   
4. The user selects an order to view more details.   
5. The system retrieves and displays the Order Confirmation Email, Product details, and Payment Information associated with the selected order.   
6. The user reviews the order information.   
  
Alternative Flow:   
1. If the user has no order history, the system displays a message indicating that there are no orders found.   
2. If the system fails to retrieve order data, it displays an error message and allows the user to retry or contact support.   
3. If the user does not have permission to view another user's purchase history (e.g., a regular customer attempting to view an administrator's orders), the system displays an access denied message and terminates the action.   
4. If the order is canceled or not yet confirmed, the system displays a status indicator to inform the user.  
  
Use Case Name: Add Product to Cart   
Use Case ID: UC-07   
Actors: Customer   
Preconditions:   
1. The system must be operational.   
2. The customer must be logged in and browsing the product catalog.   
3. The customer must have selected a product to add to the cart.   
4. The product must be available in the Inventory.   
  
Postconditions:   
1. The selected product is added to the Shopping Cart.   
2. The Shopping Cart is updated with the new item, including quantity and price.   
3. The system provides feedback to the customer confirming the addition of the product.   
  
Main Flow:   
1. The customer browses the product catalog and selects a product.   
2. The customer clicks on the "Add to Cart" button on the product detail page.   
3. The system checks the product's availability in the Inventory.   
4. If available, the system adds the product to the Shopping Cart with default quantity (e.g., 1).   
5. The system updates the cart summary and displays a confirmation message.   
6. The customer can choose to continue shopping or proceed to checkout.   
  
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 customer adds a product with a specific quantity, and that quantity exceeds the available inventory, the system adjusts the quantity to the maximum available and notifies the customer.   
3. If the customer tries to add a product that is already in the cart, the system increases the quantity of the existing item and updates the cart summary.   
4. If the system fails to update the Shopping Cart due to technical issues, it displays an error message and allows the customer to retry the action.   
5. If the customer is not logged in, the system may prompt them to log in or create an account before adding the product to the cart.  
  
Use Case Name: Modify Cart Contents   
Use Case ID: UC-08   
Actors: Customer   
Preconditions:   
1. The system must be operational.   
2. The customer must be logged in.   
3. The customer must have at least one item in their Shopping Cart.   
4. The customer must have access to the Shopping Cart interface.   
  
Postconditions:   
1. The Shopping Cart is updated with the modified item(s), including quantity or removal.   
2. The cart summary reflects the updated total price and item count.   
3. The system provides feedback to the customer confirming the modification.   
4. If the product is removed from the cart, it no longer appears in the cart summary.   
  
Main Flow:   
1. The customer navigates to the Shopping Cart page.   
2. The system displays the list of items currently in the cart.   
3. The customer selects an item to modify its quantity or remove it from the cart.   
4. The customer updates the quantity or clicks the "Remove" button for the selected item.   
5. The system updates the Shopping Cart database with the new quantity or removes the item.   
6. The system recalculates the total price and updates the cart summary.   
7. The system provides a confirmation message that the cart has been updated.   
  
Alternative Flow:   
1. If the updated quantity exceeds the available inventory, the system displays an error message and adjusts the quantity to the maximum available.   
2. If the customer enters a quantity less than one, the system displays an error message and prompts the customer to enter a valid quantity.   
3. If the system fails to update the cart due to technical issues, it displays an error message and allows the customer to retry the action.   
4. If the customer removes an item, the system logs the removal for auditing or tracking purposes.   
5. If the customer is not logged in, the system may prompt them to log in or create an account to modify their cart.  
  
Use Case Name: View Cart Summary   
Use Case ID: UC-09   
Actors: Customer   
Preconditions:   
1. The system must be operational.   
2. The customer must be logged in.   
3. The customer must have at least one item in their Shopping Cart.   
4. The customer must have access to the cart summary interface.   
  
Postconditions:   
1. The cart summary is displayed to the customer.   
2. The summary includes the total number of items and the total price.   
3. The system provides a clear overview of the items in the cart, including product name, quantity, and unit price.   
4. The customer is able to proceed to checkout or continue shopping based on the summary.   
  
Main Flow:   
1. The customer navigates to the Shopping Cart page.   
2. The system retrieves the items currently in the customer's Shopping Cart.   
3. The system displays a cart summary, including the total number of items, total price, and individual item details.   
4. The customer reviews the cart summary.   
5. The system provides options for the customer to proceed to checkout or continue shopping.   
  
Alternative Flow:   
1. If the customer's Shopping Cart is empty, the system displays a message indicating that the cart is empty and offers to browse products.   
2. If the system fails to retrieve cart data, it displays an error message and allows the customer to retry or contact support.   
3. If the product in the cart is no longer available in the Inventory, the system removes the item and notifies the customer.   
4. If the customer is not logged in, the system may prompt them to log in or create an account to view their cart summary.  
  
Use Case Name: Checkout Order   
Use Case ID: UC-10   
Actors: Customer   
Preconditions:   
1. The system must be operational.   
2. The customer must be logged in.   
3. The Shopping Cart must contain at least one item.   
4. The customer must have valid Payment Information stored in the system.   
5. The customer must have selected a shipping method and confirmed the order details.   
  
Postconditions:   
1. The order is successfully created and stored in the Order database.   
2. The Payment Information is processed, and the order is marked as paid.   
3. The Shopping Cart is cleared of the items included in the order.   
4. The customer receives an Order Confirmation Email.   
5. The Inventory is updated to reflect the reduction in stock for the purchased items.   
  
Main Flow:   
1. The customer navigates to the Shopping Cart page and clicks "Proceed to Checkout."   
2. The system displays the order summary, including item details, total price, and shipping method.   
3. The customer reviews the order and confirms the selected Payment Information.   
4. The customer clicks the "Place Order" button.   
5. The system validates the Payment Information and processes the payment.   
6. If the payment is successful, the system creates a new Order record and updates the Inventory.   
7. The system clears the Shopping Cart of the purchased items.   
8. The system sends an Order Confirmation Email to the customer.   
  
Alternative Flow:   
1. If the selected Payment Information is invalid, the system displays an error message and prompts the customer to update or correct the payment details.   
2. If the payment processing fails (e.g., insufficient funds), the system displays an error message and allows the customer to try a different payment method.   
3. If the Inventory is insufficient for the items in the cart, the system displays an error message and suggests reducing the quantity or removing the out-of-stock item.   
4. If the system fails to create the order due to technical issues, it displays an error message and allows the customer to retry the checkout process.   
5. If the customer cancels the checkout process, the system retains the items in the Shopping Cart and does not process the order.   
6. If the customer does not have any Payment Information on file, the system prompts them to add or update their payment details before proceeding.  
  
Use Case Name: Confirm Order   
Use Case ID: UC-11   
Actors: Customer, Administrator   
Preconditions:   
1. The system must be operational.   
2. The customer must have an active Shopping Cart with items selected for purchase.   
3. The customer must have valid Payment Information stored in the system.   
4. The customer must have completed the checkout process and reviewed the order summary.   
5. The customer must have access to the order confirmation interface.   
  
Postconditions:   
1. The order is successfully confirmed and stored in the Order database.   
2. The Payment Information is processed, and the order is marked as confirmed.   
3. The Shopping Cart is cleared of the items included in the order.   
4. The customer receives an Order Confirmation Email.   
5. The Inventory is updated to reflect the reduction in stock for the purchased items.   
  
Main Flow:   
1. The customer reviews the order summary on the checkout page.   
2. The customer confirms the Payment Information and shipping details.   
3. The customer clicks the "Confirm Order" button.   
4. The system validates the Payment Information and checks the Inventory for item availability.   
5. If validation and availability are successful, the system creates and confirms the Order.   
6. The system updates the Inventory with the new stock levels.   
7. The system sends an Order Confirmation Email to the customer.   
8. The system clears the Shopping Cart and displays a confirmation message to the customer.   
  
Alternative Flow:   
1. If the Payment Information is invalid, the system displays an error message and prompts the customer to correct it.   
2. If the Inventory is insufficient for the items in the order, the system displays an error message and suggests modifying the order.   
3. If the system fails to confirm the order due to technical issues, it displays an error message and allows the customer to retry the confirmation.   
4. If the customer cancels the confirmation process, the system retains the items in the Shopping Cart and does not process the order.   
5. If the Order Confirmation Email cannot be sent, the system logs the issue and displays a message to the customer to check their inbox or contact support.  
  
Use Case Name: Receive Order Confirmation Email   
Use Case ID: UC-12   
Actors: Customer, Administrator   
Preconditions:   
1. The system must be operational.   
2. An Order must have been successfully created and confirmed.   
3. The customer must have provided a valid email address in their User Account.   
4. The Order Confirmation Email feature must be enabled in the system settings or Plugin.   
  
Postconditions:   
1. The customer receives an Order Confirmation Email with order details.   
2. The email includes information such as order number, product list, total amount, and expected delivery time.   
3. The email is sent to the correct email address as per the User Account.   
4. The system logs the successful email delivery for auditing or tracking purposes.   
  
Main Flow:   
1. After the order is confirmed and stored in the Order database, the system triggers the Order Confirmation Email process.   
2. The system compiles the necessary order details, including product information, total price, and shipping address.   
3. The system generates and sends an Order Confirmation Email to the customer's registered email address.   
4. The system logs the email as having been sent to the customer.   
  
Alternative Flow:   
1. If the customer’s email address is invalid or missing, the system logs an error and may prompt the customer to update their email address.   
2. If the system fails to send the Order Confirmation Email due to server or network issues, it logs the failure and may retry sending the email at a later time or notify the customer via an in-app message.   
3. If the Order Confirmation Email feature is disabled in the system settings or Plugin, the system does not send the email and logs the status accordingly.   
4. If the customer does not receive the email, the system provides an option to request a resend of the confirmation email.  
  
Use Case Name: Manage Inventory   
Use Case ID: UC-13   
Actors: Administrator   
Preconditions:   
1. The system must be operational.   
2. The administrator must be logged in with appropriate permissions.   
3. The administrator must have access to the Inventory management interface.   
4. The Inventory must be connected to the Product and Category databases.   
  
Postconditions:   
1. The Inventory is updated with the latest product stock levels.   
2. The administrator is provided with a confirmation of the inventory changes.   
3. The system logs the inventory modification for auditing purposes.   
4. Product availability status is updated based on the new stock levels.   
  
Main Flow:   
1. The administrator navigates to the Inventory management section.   
2. The system displays the current list of products and their stock levels.   
3. The administrator selects a product to update its stock.   
4. The administrator enters the new stock quantity.   
5. The system validates the input and updates the Inventory database.   
6. The system updates the product's availability status in the Product database.   
7. The system provides a confirmation message that the inventory was successfully updated.   
  
Alternative Flow:   
1. If the administrator enters a stock quantity less than zero, the system displays an error message and prompts for a valid quantity.   
2. If the system fails to update the Inventory database due to technical issues, it displays an error message and allows the administrator to retry the action.   
3. If the product is not found in the Inventory, the system displays an error message and prompts the administrator to check the product ID or name.   
4. If the administrator does not have permission to modify the Inventory, the system displays an access denied message and terminates the action.   
5. If the system cannot update the product's availability status in the Product database, it logs the issue and displays an error message for the administrator to resolve.  
  
Use Case Name: Add Product   
Use Case ID: UC-04   
Actors: Administrator   
Preconditions:   
1. The system must be operational.   
2. The administrator must be logged in.   
3. The administrator must have access to the product management interface.   
4. The product must not already exist in the Product database (optional).   
  
Postconditions:   
1. A new product is successfully added to the Product database.   
2. The product details are stored, including name, price, description, category, and inventory quantity.   
3. The product is displayed in the product catalog for browsing.   
4. The system logs the addition of the product for auditing purposes.   
  
Main Flow:   
1. The administrator navigates to the "Add Product" section in the product management interface.   
2. The administrator enters the product details (e.g., name, price, description, category, and inventory quantity).   
3. The administrator uploads product images if required.   
4. The administrator clicks the "Save" or "Add Product" button.   
5. The system validates the input data (e.g., checks for valid price format, required fields, and category existence).   
6. If validation is successful, the system stores the product in the Product database and updates the Inventory.   
7. The system displays a confirmation message that the product was added successfully.   
  
Alternative Flow:   
1. If the administrator enters an invalid price format (e.g., negative value or non-numeric input), the system displays an error message and prompts for a valid price.   
2. If the product name is missing or too short, the system displays an error message and prompts the administrator to provide a valid product name.   
3. If the selected category does not exist in the Category database, the system displays an error message and prompts the administrator to select a valid category.   
4. If the system fails to save the product due to technical issues, it displays an error message and allows the administrator to retry the action.   
5. If the administrator cancels the product addition process, the system discards all entered data and returns to the product management interface.  
  
Use Case Name: Update Product Details   
Use Case ID: UC-14   
Actors: Administrator   
Preconditions:   
1. The system must be operational.   
2. The administrator must be logged in with appropriate permissions.   
3. The administrator must have access to the product management interface.   
4. The product to be updated must exist in the Product database.   
5. The product must be associated with an entry in the Inventory database.   
  
Postconditions:   
1. The product details are successfully updated in the Product database.   
2. The Inventory is updated to reflect any changes in stock quantity.   
3. The product catalog is updated to display the new information.   
4. The system logs the update activity for auditing purposes.   
  
Main Flow:   
1. The administrator navigates to the "Product Management" section.   
2. The system displays a list of products available for editing.   
3. The administrator selects the product they wish to update.   
4. The system retrieves the current product details from the Product and Inventory databases.   
5. The administrator modifies the product information (e.g., name, price, description, or category).   
6. The administrator updates the stock quantity if necessary.   
7. The administrator clicks the "Save Changes" button.   
8. The system validates the input data (e.g., checks for valid price format and required fields).   
9. If validation is successful, the system updates the Product database and Inventory database.   
10. The system displays a confirmation message that the product was updated successfully.   
  
Alternative Flow:   
1. If the administrator enters an invalid price format (e.g., negative value or non-numeric input), the system displays an error message and prompts for a valid price.   
2. If the product name is missing or too short, the system displays an error message and prompts the administrator to provide a valid product name.   
3. If the selected category does not exist in the Category database, the system displays an error message and prompts the administrator to select a valid category.   
4. If the system fails to update the Product or Inventory databases due to technical issues, it displays an error message and allows the administrator to retry the action.   
5. If the administrator cancels the update process, the system retains the original product details and does not apply any changes.  
  
Use Case Name: Delete Product   
Use Case ID: UC-15   
Actors: Administrator   
Preconditions:   
1. The system must be operational.   
2. The administrator must be logged in with appropriate permissions.   
3. The administrator must have access to the product management interface.   
4. The product to be deleted must exist in the Product database.   
5. The product must be associated with an entry in the Inventory database.   
  
Postconditions:   
1. The selected product is removed from the Product database.   
2. The associated inventory entry is removed or updated to reflect the deletion.   
3. The product catalog no longer displays the deleted product.   
4. The system logs the deletion activity for auditing purposes.   
  
Main Flow:   
1. The administrator navigates to the "Product Management" section.   
2. The system displays a list of products available for deletion.   
3. The administrator selects the product to be deleted.   
4. The system prompts the administrator to confirm the deletion.   
5. The administrator confirms the deletion.   
6. The system removes the product from the Product database and updates or removes the related Inventory entry.   
7. The system logs the deletion and displays a confirmation message that the product was successfully deleted.   
  
Alternative Flow:   
1. If the administrator cancels the deletion confirmation, the system retains the product and does not perform any deletion.   
2. If the product is currently in the Shopping Cart or Order database, the system displays a warning and prevents deletion until all references are cleared.   
3. If the system fails to delete the product due to technical issues, it displays an error message and allows the administrator to retry the action.   
4. If the administrator does not have permission to delete products, the system displays an access denied message and terminates the action.  
  
Use Case Name: Create Category   
Use Case ID: UC-16   
Actors: Administrator   
Preconditions:   
1. The system must be operational.   
2. The administrator must be logged in with appropriate permissions.   
3. The administrator must have access to the category management interface.   
4. The category must not already exist in the Category database (optional).   
  
Postconditions:   
1. A new category is successfully created and stored in the Category database.   
2. The category is displayed in the product catalog for browsing.   
3. The system provides feedback to the administrator confirming the category creation.   
4. The system logs the category creation for auditing purposes.   
  
Main Flow:   
1. The administrator navigates to the "Create Category" section in the category management interface.   
2. The administrator enters the category details (e.g., category name and description).   
3. The administrator clicks the "Save" or "Create Category" button.   
4. The system validates the input data (e.g., checks for required fields and duplicate category names).   
5. If validation is successful, the system stores the category in the Category database.   
6. The system displays a confirmation message that the category was created successfully.   
  
Alternative Flow:   
1. If the category name is missing or too short, the system displays an error message and prompts the administrator to provide a valid category name.   
2. If the category already exists in the database, the system displays a warning message and prevents the duplication.   
3. If the system fails to save the category due to technical issues, it displays an error message and allows the administrator to retry the action.   
4. If the administrator cancels the category creation process, the system discards all entered data and returns to the category management interface.  
  
Use Case Name: Update Category   
Use Case ID: UC-17   
Actors: Administrator   
Preconditions:   
1. The system must be operational.   
2. The administrator must be logged in with appropriate permissions.   
3. The administrator must have access to the category management interface.   
4. The category to be updated must exist in the Category database.   
  
Postconditions:   
1. The category details are successfully updated in the Category database.   
2. The product catalog is updated to reflect any changes in the category.   
3. The system logs the update activity for auditing purposes.   
4. The system provides feedback to the administrator confirming the update.   
  
Main Flow:   
1. The administrator navigates to the "Category Management" section.   
2. The system displays a list of categories available for editing.   
3. The administrator selects the category they wish to update.   
4. The system retrieves the current category details from the Category database.   
5. The administrator modifies the category information (e.g., name, description).   
6. The administrator clicks the "Save Changes" button.   
7. The system validates the input data (e.g., checks for required fields and duplicate category names).   
8. If validation is successful, the system updates the Category database with the new information.   
9. The system displays a confirmation message that the category was updated successfully.   
  
Alternative Flow:   
1. If the category name is missing or too short, the system displays an error message and prompts the administrator to provide a valid category name.   
2. If the updated category name already exists in the database, the system displays a warning message and prevents the duplication.   
3. If the system fails to update the category due to technical issues, it displays an error message and allows the administrator to retry the action.   
4. If the administrator cancels the update process, the system retains the original category details and does not apply any changes.  
  
Use Case Name: Delete Category   
Use Case ID: UC-18   
Actors: Administrator   
Preconditions:   
1. The system must be operational.   
2. The administrator must be logged in with appropriate permissions.   
3. The administrator must have access to the category management interface.   
4. The category to be deleted must exist in the Category database.   
5. The category must not be associated with any active products in the Product database (optional).   
  
Postconditions:   
1. The selected category is removed from the Category database.   
2. All products previously associated with the category are either reassigned or marked as uncategorized.   
3. The product catalog no longer displays the deleted category.   
4. The system logs the deletion activity for auditing purposes.   
5. The system provides feedback to the administrator confirming the deletion.   
  
Main Flow:   
1. The administrator navigates to the "Category Management" section.   
2. The system displays a list of categories available for deletion.   
3. The administrator selects the category they wish to delete.   
4. The system prompts the administrator to confirm the deletion.   
5. The administrator confirms the deletion.   
6. The system checks for any products associated with the selected category.   
7. If there are no associated products, or if the administrator chooses to proceed regardless, the system deletes the category from the Category database.   
8. The system logs the deletion and displays a confirmation message.   
  
Alternative Flow:   
1. If the administrator cancels the deletion confirmation, the system retains the category and does not perform any deletion.   
2. If the category is associated with one or more products, the system displays a warning message and asks the administrator whether to reassign the products to another category or proceed with deletion.   
3. If the system fails to delete the category due to technical issues, it displays an error message and allows the administrator to retry the action.   
4. If the administrator does not have permission to delete categories, the system displays an access denied message and terminates the action.  
  
Use Case Name: View Product List   
Use Case ID: UC-04   
Actors: Customer, Administrator   
Preconditions:   
1. The system must be operational.   
2. The user (Customer or Administrator) must be logged in.   
3. The user must have access to the product catalog interface.   
  
Postconditions:   
1. The product list is displayed to the user.   
2. The list includes product details such as name, price, description, category, and availability.   
3. The user is able to search, filter, or sort the product list as needed.   
  
Main Flow:   
1. The user navigates to the product catalog page.   
2. The system retrieves the list of available products from the Product and Inventory databases.   
3. The system displays the product list in a user-friendly format.   
4. The user can optionally apply filters (e.g., by category, price range, or availability).   
5. The system updates the displayed product list based on the applied filters.   
6. The user can click on a product to view its details or add it to the Shopping Cart.   
  
Alternative Flow:   
1. If the product catalog is empty, the system displays a message indicating that no products are available.   
2. If the system fails to retrieve the product list, it displays an error message and allows the user to retry or contact support.   
3. If the user applies filters that result in no matching products, the system displays a message indicating that no products match the criteria.   
4. If the user is not logged in, the system may prompt them to log in or create an account to access the product catalog.  
  
Use Case Name: Manage Orders   
Use Case ID: UC-14   
Actors: Customer, Administrator   
Preconditions:   
1. The system must be operational.   
2. The user (Customer or Administrator) must be logged in.   
3. The user must have access to the order management interface.   
4. There must be at least one order in the Order database for the user.   
  
Postconditions:   
1. The user can view, update, or cancel their orders.   
2. Order status and details are updated in the Order database.   
3. The system logs any changes made to the orders.   
4. The user receives feedback confirming the action taken.   
  
Main Flow:   
1. The user navigates to the "Order Management" section from their account dashboard.   
2. The system retrieves and displays a list of the user's orders.   
3. The user selects an order to view or modify its status.   
4. The system shows the order details, including product list, total price, payment status, and shipping status.   
5. The user chooses an action (e.g., cancel order, update status).   
6. The system validates the user's permission to perform the selected action.   
7. The system updates the Order database with the new status or action.   
8. The system provides a confirmation message that the order was successfully updated.   
  
Alternative Flow:   
1. If the user does not have permission to modify the order (e.g., a customer attempting to change another user's order), the system displays an access denied message and terminates the action.   
2. If the system fails to retrieve the order list due to technical issues, it displays an error message and allows the user to retry or contact support.   
3. If the order is already completed or canceled, the system displays a message indicating the action cannot be performed.   
4. If the user cancels the modification process, the system retains the original order status and does not apply any changes.  
  
Use Case Name: View Order Details   
Use Case ID: UC-12   
Actors: Customer, Administrator   
Preconditions:   
1. The system must be operational.   
2. The user (Customer or Administrator) must be logged in.   
3. The user must have at least one order in the Order database.   
4. The user must have access to the order details interface.   
  
Postconditions:   
1. The order details are displayed to the user.   
2. The user can view the product list, quantities, total price, payment status, and shipping status of the selected order.   
3. No changes are made to the order unless an edit or cancellation action is initiated.   
4. The system logs the access to the order details for auditing purposes.   
  
Main Flow:   
1. The user navigates to the "Purchase History" or "Order Management" section.   
2. The user selects a specific order from the list of orders.   
3. The system retrieves the selected order's details from the Order, Product, and Payment Information databases.   
4. The system displays the order details, including product names, quantities, prices, payment method, and order status.   
5. The user reviews the information.   
  
Alternative Flow:   
1. If the selected order is not found in the system, the system displays an error message and allows the user to return to the order list.   
2. If the system fails to retrieve order details due to technical issues, it displays an error message and allows the user to retry or contact support.   
3. If the user does not have permission to view the order (e.g., a customer attempting to view an administrator’s order), the system displays an access denied message and terminates the action.   
4. If the order is in a restricted state (e.g., payment pending or processing), the system displays a status message to inform the user.  
  
Use Case Name: Process Payment   
Use Case ID: UC-19   
Actors: Customer, Administrator   
Preconditions:   
1. The system must be operational.   
2. The customer must have completed the checkout process and confirmed the order.   
3. The customer must have valid Payment Information stored in the system.   
4. The Shopping Cart must contain items ready for purchase.   
5. The system must have access to a payment gateway or processing service.   
  
Postconditions:   
1. The payment is successfully processed and recorded in the Payment Information database.   
2. The order is marked as paid and updated in the Order database.   
3. The Shopping Cart is cleared of the purchased items.   
4. The Inventory is updated to reflect the reduction in stock for the purchased items.   
5. The customer receives an Order Confirmation Email.   
6. The system logs the payment processing activity for auditing purposes.   
  
Main Flow:   
1. The customer confirms the order and selects a payment method.   
2. The system retrieves the customer's Payment Information from the database.   
3. The system initiates the payment processing through the payment gateway.   
4. The payment gateway validates the transaction and returns a success or failure status.   
5. If the payment is successful, the system marks the order as paid and updates the Order database.   
6. The system clears the Shopping Cart and updates the Inventory with new stock levels.   
7. The system sends an Order Confirmation Email to the customer.   
8. The system logs the successful transaction and displays a confirmation message to the customer.   
  
Alternative Flow:   
1. If the Payment Information is invalid, the system displays an error message and prompts the customer to update or correct the payment details.   
2. If the payment processing fails (e.g., due to insufficient funds or declined card), the system displays an error message and allows the customer to try a different payment method.   
3. If the system fails to communicate with the payment gateway, it displays an error message and logs the issue for troubleshooting.   
4. If the customer cancels the payment process, the system retains the items in the Shopping Cart and does not process the order.   
5. If the Order Confirmation Email cannot be sent, the system logs the issue and displays a message to the customer to check their inbox or contact support.  
  
Use Case Name: View Payment Information   
Use Case ID: UC-05   
Actors: Customer, Administrator   
Preconditions:   
1. The system must be operational.   
2. The user (Customer or Administrator) must be logged in.   
3. The user must have access to the payment information section.   
4. The user must have at least one payment information record stored in the Payment Information database.   
  
Postconditions:   
1. The user's Payment Information is displayed on the screen.   
2. The information includes card details, expiration date, billing address, and other relevant payment data.   
3. No changes are made to the payment information unless an edit action is initiated.   
4. The system logs the access to the payment information for auditing purposes.   
  
Main Flow:   
1. The user navigates to the "Payment Information" section from their account settings.   
2. The system retrieves the user's Payment Information from the database.   
3. The system displays the payment details in a secure and user-friendly format.   
4. The user reviews the displayed payment information.   
  
Alternative Flow:   
1. If the system fails to retrieve the payment information, it displays an error message and allows the user to retry or contact support.   
2. If the user has no payment information on file, the system displays a message indicating that no payment data is available and provides an option to add new payment details.   
3. If the user does not have permission to view the payment information (e.g., a customer attempting to view another user's data), the system displays an access denied message and terminates the action.   
4. If the user's account is deleted or suspended, the system displays an error message and prevents access to the payment information section.  
  
Use Case Name: Install Plugin   
Use Case ID: UC-20   
Actors: Administrator   
Preconditions:   
1. The system must be operational.   
2. The administrator must be logged in with appropriate permissions.   
3. The Plugin must be available for installation (e.g., from a repository or uploaded file).   
4. The system must have the capability to install and activate plugins.   
  
Postconditions:   
1. The Plugin is successfully installed and activated in the system.   
2. The Plugin functionality becomes available to the relevant users or system components.   
3. The system logs the installation activity for auditing purposes.   
4. The administrator is provided with confirmation of the plugin installation.   
  
Main Flow:   
1. The administrator navigates to the Plugin management section.   
2. The system displays a list of available plugins or allows the administrator to upload a new plugin file.   
3. The administrator selects the plugin to be installed.   
4. The system validates the plugin file or source (e.g., checks compatibility, file integrity, or dependencies).   
5. If validation is successful, the system proceeds to install the plugin.   
6. The system activates the plugin and updates the system configuration accordingly.   
7. The system displays a confirmation message that the plugin was installed successfully.   
  
Alternative Flow:   
1. If the plugin file is invalid or corrupted, the system displays an error message and prevents installation.   
2. If the plugin is incompatible with the current system version, the system displays a warning message and does not proceed with installation.   
3. If the system fails to install the plugin due to technical issues, it displays an error message and allows the administrator to retry the action.   
4. If the administrator cancels the installation process, the system discards the plugin and returns to the plugin management interface.   
5. If the plugin requires additional dependencies that are not met, the system prompts the administrator to install or update the required components first.  
  
Use Case Name: View User Account   
Use Case ID: UC-04   
Actors: Customer, Administrator   
Preconditions:   
1. The system must be operational.   
2. The user (Customer or Administrator) must be logged in.   
3. The user must have access to the user account section.   
4. The user account must exist in the User Account database.   
  
Postconditions:   
1. The user’s account details are displayed on the screen.   
2. The user can view information such as name, email, account status, and role.   
3. No changes are made to the account unless an edit action is initiated.   
4. The system logs the access to the user account for auditing purposes.   
  
Main Flow:   
1. The user navigates to the "User Account" section from their dashboard.   
2. The system retrieves the user’s account information from the User Account database.   
3. The system displays the account details in a clear and secure format.   
4. The user reviews the displayed information.   
  
Alternative Flow:   
1. If the system fails to retrieve the user’s account information, it displays an error message and allows the user to retry or contact support.   
2. If the user does not have permission to view the account (e.g., a customer attempting to view another user’s details), the system displays an access denied message and terminates the action.   
3. If the user’s account is deleted or suspended, the system displays an error message and prevents access to the account section.  
  
Use Case Name: Delete User Account   
Use Case ID: UC-19   
Actors: Customer, Administrator   
Preconditions:   
1. The system must be operational.   
2. The user (Customer or Administrator) must be logged in.   
3. The user must have access to the account deletion interface.   
4. The user account to be deleted must exist in the User Account database.   
  
Postconditions:   
1. The selected user account is permanently deleted from the system.   
2. All associated data (e.g., Shopping Cart, Order History, Payment Information) is also removed or anonymized, depending on data retention policies.   
3. The system logs the deletion activity for auditing purposes.   
4. The user receives a confirmation message or email indicating the account has been deleted.   
  
Main Flow:   
1. The user navigates to the "Account Management" or "Delete Account" section from their dashboard.   
2. The system prompts the user to confirm the deletion of their account.   
3. The user provides a valid reason or confirmation (e.g., typing their password or selecting a confirmation checkbox).   
4. The user clicks the "Delete Account" button.   
5. The system verifies the user's identity and authorization to delete the account.   
6. The system initiates the deletion process, removing the account and associated data from the database.   
7. The system logs the deletion and sends a confirmation message or email to the user.   
8. The user is redirected to the home or login page with a message confirming the account has been deleted.   
  
Alternative Flow:   
1. If the user cancels the deletion confirmation, the system retains the account and does not perform any deletion.   
2. If the system fails to delete the account due to technical issues (e.g., database connection failure), it displays an error message and allows the user to retry the action.   
3. If the user does not provide valid confirmation (e.g., incorrect password), the system displays an error message and prevents the deletion.   
4. If the user is not authorized to delete the account (e.g., a customer attempting to delete an administrator account), the system displays an access denied message and terminates the action.   
5. If the deletion is initiated by the administrator, the system logs the action and may send a notification to the affected user.  
  
Use Case Name: Send Order Confirmation Email   
Use Case ID: UC-16   
Actors: Customer, Administrator   
Preconditions:   
1. The system must be operational.   
2. An order must have been successfully created and confirmed.   
3. The customer must have provided a valid email address in their User Account.   
4. The Order Confirmation Email feature must be enabled in the system settings or Plugin.   
  
Postconditions:   
1. The Order Confirmation Email is sent to the customer's registered email address.   
2. The email contains the order details, such as order number, product list, total amount, and shipping information.   
3. The system logs the email as successfully sent for auditing or tracking purposes.   
4. The customer is notified via the email that the order is confirmed and being processed.   
  
Main Flow:   
1. After the order is confirmed and saved in the Order database, the system automatically triggers the Order Confirmation Email process.   
2. The system compiles the relevant order details from the Order, Product, and Payment Information databases.   
3. The system generates the email content, including the order summary and expected delivery time.   
4. The system sends the Order Confirmation Email to the email address associated with the customer's User Account.   
5. The system logs the successful email delivery and updates the order status accordingly.   
  
Alternative Flow:   
1. If the customer’s email address is invalid or missing, the system logs an error and may prompt the customer to update their contact information.   
2. If the system fails to send the email due to server or network issues, it logs the failure and may retry sending the email or notify the customer via an in-app message.   
3. If the Order Confirmation Email feature is disabled in the Plugin or system settings, the system does not send the email and logs the status accordingly.   
4. If the customer does not receive the email, the system allows them to request a resend of the confirmation email.  
  
Use Case Name: Update Inventory Quantity   
Use Case ID: UC-14   
Actors: Administrator   
Preconditions:   
1. The system must be operational.   
2. The administrator must be logged in with appropriate permissions.   
3. The administrator must have access to the Inventory management interface.   
4. The selected product must exist in the Product database and have an associated Inventory entry.   
  
Postconditions:   
1. The Inventory quantity for the selected product is updated successfully.   
2. The product's availability status is updated based on the new quantity.   
3. The system logs the inventory update for auditing purposes.   
4. The updated inventory information is reflected in the product catalog and order processing.   
  
Main Flow:   
1. The administrator navigates to the Inventory management section.   
2. The system displays the list of products with current inventory quantities.   
3. The administrator selects a product to update its inventory quantity.   
4. The administrator enters the new quantity value.   
5. The system validates the input (e.g., ensures it is a non-negative number).   
6. If valid, the system updates the Inventory database with the new quantity.   
7. The system updates the product’s availability status in the Product database.   
8. The system provides a confirmation message that the inventory quantity was updated.   
  
Alternative Flow:   
1. If the administrator enters a negative quantity, the system displays an error message and prompts for a valid input.   
2. If the system fails to update the Inventory database due to technical issues, it displays an error message and allows the administrator to retry the action.   
3. If the product is not found in the Inventory, the system displays an error message and prompts the administrator to check the product ID or name.   
4. If the administrator does not have permission to update inventory, the system displays an access denied message and terminates the action.   
5. If the system cannot update the product’s availability status, it logs the issue and displays an error message for the administrator to resolve.  
  
Use Case Name: Manage Plugins   
Use Case ID: UC-20   
Actors: Administrator   
Preconditions:   
1. The system must be operational.   
2. The administrator must be logged in with appropriate permissions.   
3. The Plugin must be available for installation, update, or deactivation (e.g., from a repository or uploaded file).   
4. The system must have the capability to manage plugins.   
  
Postconditions:   
1. The Plugin is installed, updated, or deactivated as requested.   
2. The Plugin functionality becomes available, modified, or removed from the system.   
3. The system logs the plugin management activity for auditing purposes.   
4. The administrator receives confirmation of the action taken.   
  
Main Flow:   
1. The administrator navigates to the Plugin management section.   
2. The system displays a list of available plugins, including installed, pending, or inactive plugins.   
3. The administrator selects an action (e.g., Install, Update, Deactivate) for the chosen plugin.   
4. The system validates the plugin (e.g., checks compatibility, file integrity, or dependencies).   
5. If validation is successful, the system executes the selected action (e.g., installs, updates, or deactivates the plugin).   
6. The system updates the system configuration and logs the action.   
7. The system displays a confirmation message to the administrator indicating the action was completed.   
  
Alternative Flow:   
1. If the plugin file is invalid or corrupted, the system displays an error message and prevents installation or update.   
2. If the plugin is incompatible with the current system version, the system displays a warning message and does not proceed with the action.   
3. If the system fails to execute the action due to technical issues, it displays an error message and allows the administrator to retry the action.   
4. If the administrator cancels the action, the system discards the changes and returns to the plugin management interface.   
5. If the plugin requires additional dependencies that are not met, the system prompts the administrator to install or update the required components first.  
  
Use Case Name: View Plugin Documentation   
Use Case ID: UC-21   
Actors: Administrator   
Preconditions:   
1. The system must be operational.   
2. The administrator must be logged in with appropriate permissions.   
3. The Plugin must be installed or available for installation in the system.   
4. The Plugin must have documentation accessible within the system or from an external source.   
  
Postconditions:   
1. The administrator views the Plugin's documentation.   
2. The documentation includes information about the Plugin's features, configuration options, and usage instructions.   
3. The system logs the access to the Plugin documentation for auditing purposes.   
4. The administrator can make informed decisions regarding the Plugin's usage or management.   
  
Main Flow:   
1. The administrator navigates to the Plugin management section.   
2. The system displays a list of installed or available plugins.   
3. The administrator selects a plugin and clicks on the "View Documentation" option.   
4. The system retrieves and displays the Plugin's documentation, either internally or by redirecting to an external source.   
5. The administrator reviews the documentation to understand the Plugin's functionality and settings.   
  
Alternative Flow:   
1. If the Plugin does not have associated documentation, the system displays a message indicating that no documentation is available.   
2. If the system cannot retrieve the documentation due to technical issues, it displays an error message and allows the administrator to retry or contact support.   
3. If the administrator does not have permission to access Plugin documentation, the system displays an access denied message and terminates the action.   
4. If the Plugin is not installed, the system may provide a warning and offer to install the Plugin before viewing its documentation.  
  
Use Case Name: Manage Cart Item   
Use Case ID: UC-22   
Actors: Customer   
  
Preconditions:   
1. The system must be operational.   
2. The customer must be logged in and have access to their Shopping Cart.   
3. The Shopping Cart must contain at least one Cart Item.   
4. The system must have access to the Cart Item and Inventory databases.   
  
Postconditions:   
1. The customer can add, remove, or update the quantity of items in their Shopping Cart.   
2. The Shopping Cart is updated with the new Cart Item information.   
3. The system recalculates the total price and updates the cart summary.   
4. If the quantity is changed, the Inventory is updated accordingly (if necessary).   
5. The system logs the modification of the Cart Item for auditing purposes.   
  
Main Flow:   
1. The customer navigates to the Shopping Cart page.   
2. The system displays the list of Cart Items with their current quantity and price.   
3. The customer selects a specific Cart Item to modify.   
4. The customer updates the quantity of the selected item or removes it entirely.   
5. The system validates the requested action (e.g., checks for valid quantity or confirms the removal).   
6. The system updates the Cart Item in the Shopping Cart database.   
7. If the quantity is changed, the system checks the Inventory to ensure the updated quantity is available.   
8. The system recalculates the total price and updates the cart summary.   
9. The system displays a confirmation message indicating that the Cart Item has been successfully modified.   
  
Alternative Flow:   
1. If the updated quantity exceeds the available stock in the Inventory, the system displays a warning and adjusts the quantity to the maximum available.   
2. If the customer enters a quantity less than one, the system displays an error message and prompts the customer to enter a valid quantity.   
3. If the system fails to update the Cart Item due to technical issues, it displays an error message and allows the customer to retry the action.   
4. If the customer removes an item, the system logs the removal and updates the Inventory if necessary.   
5. If the customer is not logged in, the system may prompt them to log in or create an account before performing any modifications to the Cart Items.