项目文档

# Functional Requirement

Here is the Software Requirement Specification (SRS) for Chapter 1: Functional Requirements, following the provided guidelines and example format:  
  
---  
  
### 1.1 Account Registration Function   
\*\*Function ID:\*\* FR-01   
\*\*Description:\*\* Allows customers to register a new account by providing required personal information. The system validates the input and creates a new account upon successful verification.   
\*\*Input:\*\* Customer details (Name, Email, Password, Address).   
\*\*Output:\*\* New account record stored in the database with status "Unverified"; confirmation email sent to the customer.   
  
### 1.2 Account Login Function   
\*\*Function ID:\*\* FR-02   
\*\*Description:\*\* Authenticates customers or administrators using registered credentials and grants access to the system.   
\*\*Input:\*\* Email and Password.   
\*\*Output:\*\* Successful login redirects to the user dashboard; session data is initialized.   
  
### 1.3 Account Logout Function   
\*\*Function ID:\*\* FR-03   
\*\*Description:\*\* Terminates the current user session and clears session data.   
\*\*Input:\*\* Active user session.   
\*\*Output:\*\* Session terminated; user redirected to the login page.   
  
### 1.4 Profile Update Function   
\*\*Function ID:\*\* FR-04   
\*\*Description:\*\* Allows users to modify their profile information (e.g., name, address).   
\*\*Input:\*\* Updated profile data (e.g., Name, Address).   
\*\*Output:\*\* Updated profile record in the database; confirmation message displayed.   
  
### 1.5 Purchase History View Function   
\*\*Function ID:\*\* FR-05   
\*\*Description:\*\* Retrieves and displays a customer's past orders with details such as product name, quantity, and date.   
\*\*Input:\*\* Customer ID.   
\*\*Output:\*\* List of purchase history records displayed in the user interface.   
  
### 1.6 Order Details View Function   
\*\*Function ID:\*\* FR-06   
\*\*Description:\*\* Displays detailed information for a specific order, including products, quantities, and status.   
\*\*Input:\*\* Order ID.   
\*\*Output:\*\* Structured order details displayed in the user interface.   
  
### 1.7 Payment Information Update Function   
\*\*Function ID:\*\* FR-07   
\*\*Description:\*\* Allows customers to add, edit, or delete payment methods (e.g., credit card details).   
\*\*Input:\*\* Payment details (e.g., Card Number, Expiry Date, CVV).   
\*\*Output:\*\* Updated payment information stored in the database; confirmation message displayed.   
  
### 1.8 Inventory Management Function   
\*\*Function ID:\*\* FR-08   
\*\*Description:\*\* Enables administrators to update product stock levels and reorder thresholds.   
\*\*Input:\*\* Product ID and new inventory quantity.   
\*\*Output:\*\* Updated inventory record in the database; confirmation message displayed.   
  
### 1.9 Product Category Management Function   
\*\*Function ID:\*\* FR-09   
\*\*Description:\*\* Allows administrators to create, edit, or delete product categories.   
\*\*Input:\*\* Category details (Name, Description).   
\*\*Output:\*\* Updated category records in the database; confirmation message displayed.   
  
### 1.10 Product Details Update Function   
\*\*Function ID:\*\* FR-10   
\*\*Description:\*\* Enables administrators to modify product attributes (e.g., price, description).   
\*\*Input:\*\* Product ID and updated details (e.g., Price, Description).   
\*\*Output:\*\* Updated product record in the database; confirmation message displayed.   
  
### 1.11 Sales Report Generation Function   
\*\*Function ID:\*\* FR-11   
\*\*Description:\*\* Generates sales reports based on filters (e.g., date range, category).   
\*\*Input:\*\* Filter criteria (e.g., Date Range, Category).   
\*\*Output:\*\* Sales report data displayed or exported (e.g., PDF, Excel).   
  
### 1.12 Product Addition Function   
\*\*Function ID:\*\* FR-12   
\*\*Description:\*\* Allows administrators to add new products to the system.   
\*\*Input:\*\* Product details (Name, Description, Price, Category, Inventory).   
\*\*Output:\*\* New product record stored in the database; confirmation message displayed.   
  
### 1.13 Product Removal Function   
\*\*Function ID:\*\* FR-13   
\*\*Description:\*\* Enables administrators to remove products from the system.   
\*\*Input:\*\* Product ID.   
\*\*Output:\*\* Product record deleted from the database; confirmation message displayed.   
  
### 1.14 Product Search Function   
\*\*Function ID:\*\* FR-14   
\*\*Description:\*\* Retrieves products based on user search queries or filters.   
\*\*Input:\*\* Search term or filter criteria (e.g., Category, Price Range).   
\*\*Output:\*\* List of matching products displayed in the user interface.   
  
### 1.15 Shopping Cart Addition Function   
\*\*Function ID:\*\* FR-15   
\*\*Description:\*\* Adds a selected product to the customer's shopping cart.   
\*\*Input:\*\* Product ID and Quantity.   
\*\*Output:\*\* Updated shopping cart with the new item; confirmation message displayed.   
  
### 1.16 Shopping Cart Removal Function   
\*\*Function ID:\*\* FR-16   
\*\*Description:\*\* Removes a product from the customer's shopping cart.   
\*\*Input:\*\* Cart Item ID.   
\*\*Output:\*\* Updated shopping cart without the removed item; confirmation message displayed.   
  
### 1.17 Shopping Cart Quantity Update Function   
\*\*Function ID:\*\* FR-17   
\*\*Description:\*\* Modifies the quantity of a product in the shopping cart.   
\*\*Input:\*\* Cart Item ID and new Quantity.   
\*\*Output:\*\* Updated shopping cart with adjusted quantity; confirmation message displayed.   
  
### 1.18 Cart Summary View Function   
\*\*Function ID:\*\* FR-18   
\*\*Description:\*\* Displays a summary of the customer's shopping cart, including products and total cost.   
\*\*Input:\*\* Customer ID.   
\*\*Output:\*\* Shopping cart summary displayed in the user interface.   
  
### 1.19 Order Placement Function   
\*\*Function ID:\*\* FR-19   
\*\*Description:\*\* Processes the customer's shopping cart into a confirmed order, updates inventory, and notifies the administrator.   
\*\*Input:\*\* Customer ID and Payment Information.   
\*\*Output:\*\* New order record in the database; inventory updated; confirmation email sent.   
  
### 1.20 Order Cancellation Function   
\*\*Function ID:\*\* FR-20   
\*\*Description:\*\* Cancels a customer's order and restores inventory if applicable.   
\*\*Input:\*\* Order ID.   
\*\*Output:\*\* Order status updated to "Cancelled"; inventory adjusted (if unshipped); confirmation message displayed.   
  
### 1.21 Order Status View Function   
\*\*Function ID:\*\* FR-21   
\*\*Description:\*\* Retrieves and displays the current status of a customer's order.   
\*\*Input:\*\* Order ID.   
\*\*Output:\*\* Order status details displayed in the user interface.   
  
### 1.22 Order Confirmation Function   
\*\*Function ID:\*\* FR-22   
\*\*Description:\*\* Finalizes the order after customer confirmation, updates inventory, and notifies the administrator.   
\*\*Input:\*\* Order ID and Customer Confirmation.   
\*\*Output:\*\* Order status updated to "Confirmed"; inventory updated; confirmation email sent.   
  
### 1.23 Account Management Function   
\*\*Function ID:\*\* FR-23   
\*\*Description:\*\* Allows users to update account settings (e.g., password, email).   
\*\*Input:\*\* Updated account details (e.g., Password, Email).   
\*\*Output:\*\* Updated account record in the database; confirmation message displayed.   
  
### 1.24 Payment Information Management Function   
\*\*Function ID:\*\* FR-24   
\*\*Description:\*\* Enables users to manage their saved payment methods.   
\*\*Input:\*\* Payment details (e.g., Card Number, Expiry Date).   
\*\*Output:\*\* Updated payment records in the database; confirmation message displayed.   
  
### 1.25 Purchase History Management Function   
\*\*Function ID:\*\* FR-25   
\*\*Description:\*\* Allows administrators to view and filter customer purchase history.   
\*\*Input:\*\* Filter criteria (e.g., Customer ID, Date Range).   
\*\*Output:\*\* Filtered purchase history records displayed in the user interface.   
  
---   
  
This SRS chapter adheres to the provided rules, ensuring feasibility, completeness, and alignment with the use cases and data model. All functions are clearly defined with inputs, outputs, and descriptions.

# External Description

### Chapter 2: External Interfaces  
  
This chapter describes the external interfaces of the system, including user interfaces, hardware interfaces, software interfaces, and communication interfaces. The interfaces are classified based on their interaction with the system.  
  
---  
  
#### 2.1 User Interfaces  
  
The system interacts with users through various user interfaces, which include web pages, dashboards, and notifications. The following are the key user interfaces:  
  
1. \*\*Account Registration Page\*\*   
 - \*\*Function ID:\*\* FR-01   
 - \*\*Description:\*\* A web page where customers can enter their personal details (Name, Email, Password, Address) to register a new account.   
 - \*\*Interaction:\*\* The system validates the input and creates a new account record in the database. A confirmation email is sent to the customer.   
  
2. \*\*Login Page\*\*   
 - \*\*Function ID:\*\* FR-02   
 - \*\*Description:\*\* A web page where users (customers or administrators) can enter their credentials (Email and Password) to access the system.   
 - \*\*Interaction:\*\* Upon successful authentication, the user is redirected to their respective dashboard, and a session is initialized.   
  
3. \*\*Logout Functionality\*\*   
 - \*\*Function ID:\*\* FR-03   
 - \*\*Description:\*\* A button or link on the user dashboard that terminates the current session.   
 - \*\*Interaction:\*\* The session data is cleared, and the user is redirected to the login page.   
  
4. \*\*Profile Update Page\*\*   
 - \*\*Function ID:\*\* FR-04   
 - \*\*Description:\*\* A web page where users can update their profile information (e.g., Name, Address).   
 - \*\*Interaction:\*\* The updated data is saved to the database, and a confirmation message is displayed to the user.   
  
5. \*\*Purchase History Page\*\*   
 - \*\*Function ID:\*\* FR-05   
 - \*\*Description:\*\* A web page that displays a customer's past orders, including product details, quantity, and order date.   
 - \*\*Interaction:\*\* The system retrieves the purchase history from the database and displays it in a structured format.   
  
6. \*\*Order Details Page\*\*   
 - \*\*Function ID:\*\* FR-06   
 - \*\*Description:\*\* A web page that shows detailed information about a specific order, including product details, quantities, and order status.   
 - \*\*Interaction:\*\* The system retrieves the order details from the database and displays them in a structured format.   
  
7. \*\*Payment Information Management Page\*\*   
 - \*\*Function ID:\*\* FR-07   
 - \*\*Description:\*\* A web page where customers can add, edit, or delete their payment methods (e.g., credit card details).   
 - \*\*Interaction:\*\* The payment details are saved to the database, and a confirmation message is displayed.   
  
8. \*\*Shopping Cart Page\*\*   
 - \*\*Function ID:\*\* FR-15, FR-16, FR-17   
 - \*\*Description:\*\* A web page where customers can add, remove, or modify the quantity of products in their shopping cart.   
 - \*\*Interaction:\*\* The system updates the shopping cart in real-time and displays a confirmation message upon successful updates.   
  
9. \*\*Order Placement Confirmation Page\*\*   
 - \*\*Function ID:\*\* FR-19   
 - \*\*Description:\*\* A web page that confirms the placement of an order and displays a summary of the order.   
 - \*\*Interaction:\*\* The system updates the inventory in the database, sends a confirmation email to the customer, and notifies the administrator.   
  
10. \*\*Order Cancellation Confirmation Page\*\*   
 - \*\*Function ID:\*\* FR-20   
 - \*\*Description:\*\* A web page that confirms the cancellation of an order and displays the updated status.   
 - \*\*Interaction:\*\* The system updates the order status in the database and restores inventory if applicable.   
  
11. \*\*Order Status Page\*\*   
 - \*\*Function ID:\*\* FR-21   
 - \*\*Description:\*\* A web page that displays the current status of an order.   
 - \*\*Interaction:\*\* The system retrieves the order status from the database and displays it to the user.   
  
12. \*\*Administrator Dashboard\*\*   
 - \*\*Function IDs:\*\* FR-08, FR-09, FR-10, FR-11, FR-12, FR-13, FR-25   
 - \*\*Description:\*\* A web page where administrators can manage inventory, product categories, product details, sales reports, and purchase history.   
 - \*\*Interaction:\*\* The system retrieves and displays relevant data from the database based on the administrator's actions.   
  
---  
  
#### 2.2 Hardware Interfaces  
  
The system does not interact directly with any hardware devices. All interactions are mediated through software interfaces or user interfaces.  
  
---  
  
#### 2.3 Software Interfaces  
  
The system interacts with external software systems and databases to perform its functions. The following are the key software interfaces:  
  
1. \*\*Database Interface\*\*   
 - \*\*Description:\*\* The system interacts with a relational database to store and retrieve data.   
 - \*\*Interaction:\*\* The system performs CRUD (Create, Read, Update, Delete) operations on the database to manage accounts, orders, products, payment methods, and inventory.   
  
2. \*\*Email Service Interface\*\*   
 - \*\*Function IDs:\*\* FR-01, FR-19, FR-22   
 - \*\*Description:\*\* The system uses an email service to send confirmation emails to customers during account registration, order placement, and order confirmation.   
 - \*\*Interaction:\*\* The system sends email notifications to customers using an email service API.   
  
3. \*\*Payment Gateway Interface\*\*   
 - \*\*Function IDs:\*\* FR-07, FR-19   
 - \*\*Description:\*\* The system integrates with a payment gateway to process payments and manage payment methods.   
 - \*\*Interaction:\*\* The system communicates with the payment gateway API to validate payment details, process transactions, and store payment information securely.   
  
4. \*\*Third-Party APIs\*\*   
 - \*\*Description:\*\* The system may interact with third-party APIs for additional functionalities (e.g., product recommendations, shipping integration).   
 - \*\*Interaction:\*\* The system sends requests to third-party APIs and processes the responses to enhance functionality.   
  
---  
  
#### 2.4 Communication Interfaces  
  
The system communicates with external entities via network-based interfaces. The following are the key communication interfaces:  
  
1. \*\*Email Notifications\*\*   
 - \*\*Function IDs:\*\* FR-01, FR-19, FR-22   
 - \*\*Description:\*\* The system sends email notifications to customers during account registration, order placement, and order confirmation.   
 - \*\*Interaction:\*\* The system uses an email service API to send emails to customer accounts.   
  
2. \*\*Administrator Notifications\*\*   
 - \*\*Function IDs:\*\* FR-19, FR-22   
 - \*\*Description:\*\* The system sends notifications to administrators when an order is placed or confirmed.   
 - \*\*Interaction:\*\* The system sends email or in-app notifications to administrators to alert them of new orders.   
  
3. \*\*Web Browser Interface\*\*   
 - \*\*Description:\*\* The system is accessed via a web browser, which serves as the primary interface for users.   
 - \*\*Interaction:\*\* Users interact with the system through a web browser, which communicates with the server via HTTP/HTTPS requests.   
  
---  
  
This chapter ensures that all external interfaces are clearly defined and aligned with the functional requirements. The interfaces are designed to facilitate seamless interaction between the system and its users, as well as with external software and communication systems.

# Use Case

Use Case Name: Register Account   
Use Case ID: UC-01   
Actors: Customer, Administrator   
Preconditions: The system must be operational, and the Customer must not have an existing account.   
Postconditions: A new account for the Customer is successfully created and verified by the Administrator.   
  
Main Flow:   
1. The Customer navigates to the account registration page.   
2. The system displays the registration form with required fields (e.g., email, password, name).   
3. The Customer fills in the registration form with valid information.   
4. The system checks if the email is already registered.   
5. If the email is not registered, the system saves the Customer's information in the database.   
6. The system sends a confirmation email to the Customer's provided email address.   
7. The Administrator verifies the new account.   
8. The system updates the account status to "verified" and notifies the Customer.   
  
Alternative Flow:   
1. In step 4, if the email is already registered, the system displays an error message and prompts the Customer to provide a different email.   
2. In step 3, if the Customer provides invalid or incomplete information, the system displays an error message and prompts the Customer to correct the information.   
3. In step 7, if the Administrator rejects the account, the system updates the account status to "rejected" and notifies the Customer.  
  
Use Case Name: Login   
Use Case ID: UC-02   
Actors: Customer, Administrator   
Preconditions: The system must be operational, and the Customer or Administrator must have a verified account.   
Postconditions: The Customer or Administrator is successfully logged into the system and redirected to their respective dashboard.   
  
Main Flow:   
1. The Customer or Administrator navigates to the login page.   
2. The system displays the login form with required fields (e.g., email, password).   
3. The Customer or Administrator enters their valid email and password.   
4. The system validates the credentials against the database.   
5. If the credentials are valid, the system logs the user in and redirects them to the appropriate dashboard.   
6. The system displays a welcome message to the logged-in user.   
  
Alternative Flow:   
1. In step 3, if the email is invalid or not registered, the system displays an error message and prompts the user to provide a valid email.   
2. In step 3, if the password is incorrect, the system displays an error message and prompts the user to re-enter the correct password.   
3. In step 4, if the account is not verified, the system displays an error message and prompts the user to check their email for a verification link.   
4. In step 5, if the system encounters an error during login, it displays an error message and logs the issue for further investigation.  
  
Use Case Name: Logout   
Use Case ID: UC-03   
Actors: Customer, Administrator   
Preconditions: The system must be operational, and the Customer or Administrator must be logged in.   
Postconditions: The Customer or Administrator is successfully logged out, and their session is terminated.   
  
Main Flow:   
1. The Customer or Administrator navigates to the account settings or user menu.   
2. The system displays a "Logout" option.   
3. The Customer or Administrator selects the "Logout" option.   
4. The system terminates the current user session and clears session data.   
5. The system redirects the user to the login page.   
6. The system displays a confirmation message indicating that the user has been logged out.   
  
Alternative Flow:   
1. In step 3, if the user session is already expired or invalid, the system redirects the user to the login page without displaying a confirmation message.   
2. In step 4, if there is an error in terminating the session, the system displays an error message and logs the issue for further investigation.  
  
Use Case Name: Update Profile   
Use Case ID: UC-04   
Actors: Customer, Administrator   
Preconditions: The system must be operational, and the Customer or Administrator must be logged in with a verified account.   
Postconditions: The Customer's or Administrator's profile information is successfully updated in the system.   
  
Main Flow:   
1. The Customer or Administrator navigates to the profile settings page.   
2. The system displays the current profile information and an editable form.   
3. The Customer or Administrator modifies the desired information (e.g., name, address, contact details).   
4. The Customer or Administrator submits the updated form.   
5. The system validates the input data.   
6. If the data is valid, the system updates the profile information in the database.   
7. The system displays a confirmation message that the profile was successfully updated.   
  
Alternative Flow:   
1. In step 3, if the Customer or Administrator attempts to change the email address, the system checks if the new email is already registered.   
2. In step 5, if the input data is invalid or incomplete, the system displays an error message and prompts the user to correct the information.   
3. In step 6, if the system fails to update the profile, it displays an error message and logs the issue for further investigation.  
  
Use Case Name: View Purchase History   
Use Case ID: UC-05   
Actors: Customer, Administrator   
Preconditions: The system must be operational, and the Customer or Administrator must be logged in with a verified account.   
Postconditions: The purchase history of the Customer is successfully retrieved and displayed.   
  
Main Flow:   
1. The Customer or Administrator navigates to the "Purchase History" section in their account dashboard.   
2. The system retrieves the Customer's order records from the database.   
3. The system displays the list of previous purchases, including details such as product name, quantity, price, and order date.   
4. The user can filter or sort the purchase history by date, product category, or order status.   
5. The system updates the view to reflect the selected filters or sorting criteria.   
  
Alternative Flow:   
1. In step 2, if there are no orders found for the Customer, the system displays a message indicating that no purchase history is available.   
2. In step 4, if the user enters invalid filter criteria, the system displays an error message and prompts them to use valid options.   
3. In step 3, if the system encounters an error retrieving the data, it displays an error message and logs the issue for further investigation.  
  
Use Case Name: View Order Details   
Use Case ID: UC-06   
Actors: Customer, Administrator   
Preconditions: The system must be operational, and the Customer or Administrator must be logged in with a verified account. An Order must already exist in the system.   
Postconditions: The details of the selected Order are successfully retrieved and displayed to the user.   
  
Main Flow:   
1. The Customer or Administrator navigates to the "Order Details" section from the account dashboard or order list.   
2. The system displays a list of available orders with order IDs and basic information.   
3. The user selects a specific order to view.   
4. The system retrieves the detailed information of the selected Order, including Product, Quantity, Price, Order Date, and Order Status.   
5. The system displays the Order details in a structured format for the user to review.   
  
Alternative Flow:   
1. In step 3, if the selected Order ID is invalid or does not exist, the system displays an error message and prompts the user to select a valid order.   
2. In step 4, if there is an error retrieving the Order details, the system displays an error message and logs the issue for further investigation.  
  
Use Case Name: Update Payment Information   
Use Case ID: UC-07   
Actors: Customer, Administrator   
Preconditions: The system must be operational, and the Customer must be logged in with a verified account. The system must have an existing Order or Shopping Cart associated with the Customer.   
Postconditions: The Customer's payment information is successfully updated in the system, and the Administrator is notified of the change if applicable.   
  
Main Flow:   
1. The Customer navigates to the "Payment Information" section in their account settings.   
2. The system displays the current payment details and an editable form for updating the information.   
3. The Customer modifies the payment information (e.g., credit card number, expiration date, billing address).   
4. The Customer submits the updated payment form.   
5. The system validates the input data and checks for any inconsistencies.   
6. If the data is valid, the system updates the Customer's payment information in the database.   
7. The system displays a confirmation message that the payment information was successfully updated.   
  
Alternative Flow:   
1. In step 3, if the Customer attempts to update a payment method that is already in use, the system prompts them to confirm the change.   
2. In step 5, if the input data is invalid or incomplete, the system displays an error message and prompts the Customer to correct the information.   
3. In step 6, if the system fails to update the payment information, it displays an error message and logs the issue for further investigation.   
4. In step 4, if the Administrator has restricted payment updates for certain accounts, the system displays a message indicating that the action cannot be performed at this time.  
  
Use Case Name: Manage Inventory   
Use Case ID: UC-08   
Actors: Administrator   
Preconditions: The system must be operational, and the Administrator must be logged in with a verified account. The Product and Product Category data entities must exist in the database.   
Postconditions: The inventory levels of Products are successfully updated, and the system reflects the changes in the database and user interface.   
  
Main Flow:   
1. The Administrator navigates to the "Inventory Management" section from the dashboard.   
2. The system displays a list of Products along with their current inventory levels, categories, and prices.   
3. The Administrator selects a specific Product to modify its inventory.   
4. The system opens an editable form for the selected Product, allowing the Administrator to update inventory quantity.   
5. The Administrator enters the new inventory quantity and submits the form.   
6. The system validates the input quantity (e.g., must be a positive number).   
7. If valid, the system updates the Product's inventory level in the database.   
8. The system refreshes the inventory list and displays a confirmation message that the inventory was updated.   
  
Alternative Flow:   
1. In step 3, if the selected Product ID is invalid or does not exist, the system displays an error message and prompts the Administrator to select a valid Product.   
2. In step 5, if the input quantity is invalid (e.g., negative number or non-numeric value), the system displays an error message and prompts the Administrator to correct the input.   
3. In step 7, if the system fails to update the inventory, it displays an error message and logs the issue for further investigation.   
4. In step 4, if the Administrator attempts to modify a Product that is not in the system, the system displays a message indicating that the Product is unavailable.  
  
Use Case Name: Manage Product Categories   
Use Case ID: UC-09   
Actors: Administrator   
Preconditions: The system must be operational, and the Administrator must be logged in with a verified account. The Product and Product Category data entities must exist in the database.   
Postconditions: The Product Category is successfully added, edited, or removed from the system, and the changes are reflected in the database and the product listings.   
  
Main Flow:   
1. The Administrator navigates to the "Product Categories" section from the dashboard.   
2. The system displays a list of existing Product Categories, including their names and associated Products.   
3. The Administrator selects an option to add, edit, or delete a Product Category.   
4. If adding, the system displays a form for entering a new category name and description.   
5. If editing, the system displays the selected category's details in an editable form.   
6. If deleting, the system prompts the Administrator to confirm the deletion.   
7. The Administrator provides the required information or confirms the action.   
8. The system validates the input (if applicable) and updates the database accordingly.   
9. The system displays a confirmation message indicating the success of the action.   
  
Alternative Flow:   
1. In step 6, if the Administrator cancels the deletion, the system does not perform any changes and returns to the category list.   
2. In step 4 or 5, if the Administrator provides invalid or duplicate category information, the system displays an error message and prompts for valid input.   
3. In step 8, if the system encounters an error during the database update, it displays an error message and logs the issue for further investigation.   
4. In step 3, if the selected category does not exist, the system displays an error message and prompts the Administrator to select a valid category.  
  
Use Case Name: Update Product Details   
Use Case ID: UC-10   
Actors: Administrator   
Preconditions: The system must be operational, and the Administrator must be logged in with a verified account. A Product must already exist in the database.   
Postconditions: The selected Product's details (e.g., name, price, description, category) are successfully updated in the system and reflected in the database and user interface.   
  
Main Flow:   
1. The Administrator navigates to the "Product Management" section from the dashboard.   
2. The system displays a list of Products, including their names, prices, categories, and descriptions.   
3. The Administrator selects a specific Product to update its details.   
4. The system opens an editable form for the selected Product, displaying the current information.   
5. The Administrator modifies the desired fields (e.g., price, description, category).   
6. The Administrator submits the updated form.   
7. The system validates the input data (e.g., price must be a positive number).   
8. If valid, the system updates the Product's details in the database.   
9. The system displays a confirmation message that the Product was successfully updated.   
  
Alternative Flow:   
1. In step 5, if the Administrator attempts to change the Product Category to a non-existent one, the system displays an error message and prompts to select a valid category.   
2. In step 6, if the Administrator provides invalid or incomplete information, the system displays an error message and prompts for valid input.   
3. In step 8, if the system fails to update the Product, it displays an error message and logs the issue for further investigation.   
4. In step 3, if the selected Product ID is invalid or does not exist, the system displays an error message and prompts the Administrator to select a valid Product.  
  
Use Case Name: View Sales Reports   
Use Case ID: UC-11   
Actors: Administrator   
Preconditions: The system must be operational, and the Administrator must be logged in with a verified account. Sales data must exist in the database.   
Postconditions: The Administrator successfully views and filters the sales reports based on specified criteria.   
  
Main Flow:   
1. The Administrator navigates to the "Sales Reports" section from the dashboard.   
2. The system displays a sales report interface with options to filter by date range, product category, or salesperson.   
3. The Administrator selects the desired filter criteria.   
4. The system retrieves the relevant sales data from the database.   
5. The system displays the sales report, including total sales, revenue, and product performance.   
6. The Administrator can download the report in a specified format (e.g., PDF, Excel).   
  
Alternative Flow:   
1. In step 3, if the Administrator enters invalid or conflicting filter criteria, the system displays an error message and prompts for valid input.   
2. In step 4, if no sales data matches the selected criteria, the system displays a message indicating that no data is available for the selected filters.   
3. In step 6, if the system encounters an error during the download, it displays an error message and logs the issue for further investigation.  
  
Use Case Name: Add Product   
Use Case ID: UC-12   
Actors: Administrator   
Preconditions: The system must be operational, and the Administrator must be logged in with a verified account. The Product Category data entity must exist in the database.   
Postconditions: A new Product is successfully added to the system and displayed in the database and product listings.   
  
Main Flow:   
1. The Administrator navigates to the "Product Management" section from the dashboard.   
2. The system displays an "Add Product" button.   
3. The Administrator selects the "Add Product" button.   
4. The system opens a form for entering Product details (e.g., name, price, description, category, image, inventory quantity).   
5. The Administrator fills in the form with valid information.   
6. The Administrator submits the form.   
7. The system validates the input data (e.g., price must be a positive number, category must exist).   
8. If valid, the system saves the new Product in the database.   
9. The system displays a confirmation message that the Product was successfully added.   
  
Alternative Flow:   
1. In step 5, if the Administrator provides invalid or incomplete information, the system displays an error message and prompts for valid input.   
2. In step 7, if the selected Product Category does not exist, the system displays an error message and prompts the Administrator to select a valid category.   
3. In step 8, if the system fails to add the Product, it displays an error message and logs the issue for further investigation.  
  
Use Case Name: Remove Product   
Use Case ID: UC-13   
Actors: Administrator   
Preconditions: The system must be operational, and the Administrator must be logged in with a verified account. The Product to be removed must exist in the database.   
Postconditions: The specified Product is successfully removed from the system and the changes are reflected in the database and product listings.   
  
Main Flow:   
1. The Administrator navigates to the "Product Management" section from the dashboard.   
2. The system displays a list of Products with their details (e.g., name, category, price, inventory quantity).   
3. The Administrator selects a specific Product to remove.   
4. The system prompts the Administrator to confirm the removal of the selected Product.   
5. The Administrator confirms the removal.   
6. The system removes the Product from the database.   
7. The system displays a confirmation message that the Product was successfully removed.   
  
Alternative Flow:   
1. In step 3, if the selected Product ID is invalid or does not exist, the system displays an error message and prompts the Administrator to select a valid Product.   
2. In step 5, if the Administrator cancels the removal, the system does not perform any changes and returns to the product list.   
3. In step 6, if the system encounters an error during the removal process, it displays an error message and logs the issue for further investigation.  
  
Use Case Name: Search Product   
Use Case ID: UC-14   
Actors: Customer, Administrator   
Preconditions: The system must be operational, and the Customer or Administrator must be logged in with a verified account. At least one Product must exist in the database.   
Postconditions: The search results for Products are successfully retrieved and displayed to the user based on the search criteria.   
  
Main Flow:   
1. The Customer or Administrator navigates to the "Search Product" section on the homepage or dashboard.   
2. The system displays a search bar with options to filter by category, price range, or keyword.   
3. The user enters a keyword or selects a category to search for Products.   
4. The system processes the search query and retrieves matching Products from the database.   
5. The system displays the search results in a structured format, including product name, price, category, and image.   
6. The user can click on a specific Product to view more details.   
  
Alternative Flow:   
1. In step 3, if the search query is empty or invalid, the system displays a message prompting the user to enter a valid search term or select a category.   
2. In step 4, if no Products match the search criteria, the system displays a message indicating that no Products were found.   
3. In step 6, if the selected Product ID is invalid or does not exist, the system displays an error message and prompts the user to select a valid Product.   
4. In step 4, if the system encounters an error retrieving the search results, it displays an error message and logs the issue for further investigation.  
  
Use Case Name: Create Category   
Use Case ID: UC-15   
Actors: Administrator   
Preconditions: The system must be operational, and the Administrator must be logged in with a verified account. The Product Category data entity must not already exist with the same name.   
Postconditions: A new Product Category is successfully created and added to the system database and product listings.   
  
Main Flow:   
1. The Administrator navigates to the "Product Categories" section from the dashboard.   
2. The system displays a list of existing Product Categories and an "Add Category" button.   
3. The Administrator selects the "Add Category" button.   
4. The system opens a form for entering a new category name and description.   
5. The Administrator fills in the form with valid information.   
6. The Administrator submits the form.   
7. The system validates the input (e.g., name is not empty and not duplicated).   
8. If valid, the system creates and saves the new Product Category in the database.   
9. The system displays a confirmation message that the category was successfully created.   
  
Alternative Flow:   
1. In step 5, if the Administrator provides an empty or invalid category name, the system displays an error message and prompts for valid input.   
2. In step 7, if the category name already exists, the system displays an error message and prompts the Administrator to choose a different name.   
3. In step 8, if the system fails to create the category, it displays an error message and logs the issue for further investigation.  
  
Use Case Name: Delete Category   
Use Case ID: UC-16   
Actors: Administrator   
Preconditions: The system must be operational, and the Administrator must be logged in with a verified account. A Product Category must exist in the database and must not have any associated Products.   
Postconditions: The selected Product Category is successfully deleted from the system, and the changes are reflected in the database and product listings.   
  
Main Flow:   
1. The Administrator navigates to the "Product Categories" section from the dashboard.   
2. The system displays a list of existing Product Categories with options to delete.   
3. The Administrator selects a specific Product Category to delete.   
4. The system checks if the selected category has associated Products.   
5. If no Products are associated, the system prompts the Administrator to confirm the deletion.   
6. The Administrator confirms the deletion.   
7. The system removes the selected Product Category from the database.   
8. The system updates the category list and displays a confirmation message that the category was successfully deleted.   
  
Alternative Flow:   
1. In step 4, if the selected Product Category has associated Products, the system displays an error message and prompts the Administrator to either remove the Products first or cancel the deletion.   
2. In step 3, if the selected Product Category ID is invalid or does not exist, the system displays an error message and prompts the Administrator to select a valid category.   
3. In step 7, if the system encounters an error during the deletion process, it displays an error message and logs the issue for further investigation.   
4. In step 6, if the Administrator cancels the deletion, the system does not perform any changes and returns to the category list.  
  
Use Case Name: Update Category   
Use Case ID: UC-17   
Actors: Administrator   
Preconditions: The system must be operational, and the Administrator must be logged in with a verified account. The Product Category to be updated must already exist in the database.   
Postconditions: The selected Product Category's details (e.g., name, description) are successfully updated in the system and reflected in the database and product listings.   
  
Main Flow:   
1. The Administrator navigates to the "Product Categories" section from the dashboard.   
2. The system displays a list of existing Product Categories with options to edit.   
3. The Administrator selects a specific Product Category to update.   
4. The system opens an editable form for the selected category, displaying its current name and description.   
5. The Administrator modifies the category name or description as needed.   
6. The Administrator submits the updated form.   
7. The system validates the input (e.g., name is not empty and not duplicated).   
8. If valid, the system updates the Product Category in the database.   
9. The system displays a confirmation message that the category was successfully updated.   
  
Alternative Flow:   
1. In step 5, if the Administrator provides an empty or invalid category name, the system displays an error message and prompts for valid input.   
2. In step 7, if the new category name already exists, the system displays an error message and prompts the Administrator to choose a different name.   
3. In step 8, if the system fails to update the category, it displays an error message and logs the issue for further investigation.   
4. In step 3, if the selected category ID is invalid or does not exist, the system displays an error message and prompts the Administrator to select a valid category.  
  
Use Case Name: Add to Cart   
Use Case ID: UC-18   
Actors: Customer   
Preconditions: The system must be operational, and the Customer must be logged in with a verified account. A Product must be available for purchase, and the Shopping Cart data entity must exist in the system.   
Postconditions: The selected Product is successfully added to the Customer's Shopping Cart, and the Shopping Cart is updated in the system.   
  
Main Flow:   
1. The Customer browses Products on the website or application.   
2. The Customer selects a Product to add to their Shopping Cart.   
3. The system checks the Product's availability and inventory level.   
4. If the Product is available, the system adds it to the Customer's Shopping Cart with the specified quantity.   
5. The system updates the Shopping Cart display to show the added Product and total items.   
6. The system displays a confirmation message that the Product was successfully added to the cart.   
  
Alternative Flow:   
1. In step 3, if the Product is out of stock, the system displays an error message and suggests alternative Products or a restock date.   
2. In step 2, if the Customer attempts to add an invalid Product (e.g., non-existent or not available for purchase), the system displays an error message and does not update the cart.   
3. In step 4, if the system encounters an error while updating the Shopping Cart, it displays an error message and logs the issue for further investigation.   
4. In step 2, if the Customer tries to add a quantity exceeding the available inventory, the system displays an error message and prompts the Customer to adjust the quantity.  
  
Use Case Name: Remove from Cart   
Use Case ID: UC-19   
Actors: Customer   
Preconditions: The system must be operational, and the Customer must be logged in with a verified account. The Shopping Cart data entity must exist in the system and contain at least one Product.   
Postconditions: The selected Product is successfully removed from the Customer's Shopping Cart, and the Shopping Cart is updated in the system.   
  
Main Flow:   
1. The Customer navigates to the "Shopping Cart" page.   
2. The system displays the list of Products currently in the Customer's Shopping Cart.   
3. The Customer selects a specific Product to remove from the cart.   
4. The system prompts the Customer to confirm the removal of the selected Product.   
5. The Customer confirms the removal.   
6. The system removes the selected Product from the Shopping Cart.   
7. The system updates the Shopping Cart display to reflect the change.   
8. The system displays a confirmation message that the Product was successfully removed.   
  
Alternative Flow:   
1. In step 3, if the selected Product is not in the Shopping Cart, the system displays an error message and prompts the Customer to select a valid Product.   
2. In step 5, if the Customer cancels the removal, the system does not perform any changes and returns to the Shopping Cart view.   
3. In step 6, if the system encounters an error while removing the Product, it displays an error message and logs the issue for further investigation.  
  
Use Case Name: Modify Cart Quantity   
Use Case ID: UC-20   
Actors: Customer   
Preconditions: The system must be operational, and the Customer must be logged in with a verified account. The Shopping Cart data entity must exist and contain the Product for which the quantity will be modified.   
Postconditions: The quantity of the selected Product in the Customer's Shopping Cart is successfully updated, and the Shopping Cart is reflected in the system.   
  
Main Flow:   
1. The Customer navigates to the "Shopping Cart" page.   
2. The system displays the list of Products in the Shopping Cart along with their current quantities.   
3. The Customer selects a specific Product and adjusts its quantity using the provided input field or buttons.   
4. The system checks if the requested quantity is within the available inventory limits.   
5. If valid, the system updates the quantity of the selected Product in the Shopping Cart.   
6. The system recalculates the total price and updates the Shopping Cart display.   
7. The system displays a confirmation message that the quantity was successfully modified.   
  
Alternative Flow:   
1. In step 4, if the requested quantity exceeds the available inventory, the system displays an error message and allows the Customer to adjust the quantity to a valid value.   
2. In step 4, if the requested quantity is negative or invalid, the system displays an error message and prompts the Customer to enter a valid quantity.   
3. In step 3, if the Customer attempts to modify the quantity of a Product that is no longer available, the system displays an error message and suggests removing the Product from the cart.   
4. In step 5, if the system encounters an error while updating the quantity, it displays an error message and logs the issue for further investigation.  
  
Use Case Name: View Cart Summary   
Use Case ID: UC-21   
Actors: Customer   
Preconditions: The system must be operational, and the Customer must be logged in with a verified account. The Shopping Cart data entity must exist and contain at least one Product.   
Postconditions: The Customer successfully views a summary of their Shopping Cart, including the list of Products, quantities, prices, and the total amount.   
  
Main Flow:   
1. The Customer navigates to the "Shopping Cart" page.   
2. The system retrieves the Customer's Shopping Cart data from the database.   
3. The system displays the Cart Summary, including each Product's name, quantity, price, and total price.   
4. The system shows the overall total amount for the entire cart.   
5. The system provides options to proceed to checkout, modify quantities, or remove items.   
  
Alternative Flow:   
1. In step 2, if the Shopping Cart is empty, the system displays a message indicating that no items are currently in the cart.   
2. In step 3, if the system encounters an error retrieving or displaying Product details, it displays an error message and logs the issue for further investigation.  
  
Use Case Name: Place Order   
Use Case ID: UC-22   
Actors: Customer, Administrator   
Preconditions: The system must be operational, and the Customer must be logged in with a verified account. The Shopping Cart must contain at least one Product, and the inventory for each Product must be sufficient to fulfill the order. The payment information must be valid and up to date.   
Postconditions: The Customer successfully places an Order for the Products in their Shopping Cart. The Order is recorded in the system, the inventory levels are updated, and the Administrator is notified of the new Order.   
  
Main Flow:   
1. The Customer navigates to the "Shopping Cart" page and reviews the items.   
2. The Customer selects the "Proceed to Checkout" option.   
3. The system displays a confirmation page with the list of Products, quantities, prices, and total amount.   
4. The system verifies the Customer's payment information.   
5. The system processes the payment.   
6. If the payment is successful, the system creates a new Order record in the database.   
7. The system deducts the ordered quantities from the inventory of each Product.   
8. The system sends a confirmation email to the Customer with the Order details.   
9. The system notifies the Administrator of the new Order.   
10. The system displays a confirmation message that the Order was successfully placed.   
  
Alternative Flow:   
1. In step 4, if the Customer's payment information is invalid or incomplete, the system displays an error message and prompts the Customer to update their payment details.   
2. In step 5, if the payment fails, the system displays an error message and cancels the Order creation process.   
3. In step 7, if the system encounters an error updating the inventory, it displays an error message and logs the issue for further investigation.   
4. In step 3, if the Customer decides to modify the cart before proceeding, the system redirects them back to the Shopping Cart page.   
5. In step 2, if the Customer cancels the checkout, the system returns them to the Shopping Cart page without any changes.  
  
Use Case Name: Cancel Order   
Use Case ID: UC-23   
Actors: Customer, Administrator   
Preconditions: The system must be operational, and the Customer must be logged in with a verified account. An Order must already exist in the system and be in a cancellable status (e.g., "Pending" or "Processing").   
Postconditions: The selected Order is successfully cancelled, and the change is reflected in the system. Inventory levels for the Products in the Order are updated if applicable, and the Administrator is notified of the cancellation.   
  
Main Flow:   
1. The Customer navigates to the "Order History" section in their account dashboard.   
2. The system displays a list of Orders with their statuses.   
3. The Customer selects an Order that is in a cancellable status.   
4. The system prompts the Customer to confirm the cancellation of the selected Order.   
5. The Customer confirms the cancellation.   
6. The system updates the Order status to "Cancelled" in the database.   
7. If the Order has not been shipped, the system returns the Product quantities to the inventory.   
8. The system sends a cancellation confirmation email to the Customer.   
9. The system notifies the Administrator of the cancellation.   
10. The system displays a confirmation message that the Order was successfully cancelled.   
  
Alternative Flow:   
1. In step 3, if the selected Order is not in a cancellable status, the system displays an error message and indicates that cancellation is not allowed at this stage.   
2. In step 5, if the Customer cancels the operation, the system does not perform any changes and returns to the Order History view.   
3. In step 6, if the system fails to update the Order status, it displays an error message and logs the issue for further investigation.   
4. In step 7, if there is an error updating the inventory, the system displays an error message and logs the issue for further investigation.   
5. In step 3, if the selected Order ID is invalid or does not exist, the system displays an error message and prompts the Customer to select a valid Order.  
  
Use Case Name: View Order Status   
Use Case ID: UC-24   
Actors: Customer, Administrator   
Preconditions: The system must be operational, and the Customer or Administrator must be logged in with a verified account. An Order must already exist in the system.   
Postconditions: The current status of the selected Order is successfully retrieved and displayed to the user.   
  
Main Flow:   
1. The Customer or Administrator navigates to the "Order Status" section from the account dashboard or order history.   
2. The system displays a list of Orders with their order IDs and basic information.   
3. The user selects a specific Order to view its status.   
4. The system retrieves the current status of the selected Order from the database.   
5. The system displays the Order status along with relevant details (e.g., expected delivery date, shipping status).   
  
Alternative Flow:   
1. In step 3, if the selected Order ID is invalid or does not exist, the system displays an error message and prompts the user to select a valid Order.   
2. In step 4, if the system encounters an error retrieving the Order status, it displays an error message and logs the issue for further investigation.  
  
Use Case Name: Confirm Order   
Use Case ID: UC-25   
Actors: Customer, Administrator   
Preconditions: The system must be operational, and the Customer must be logged in with a verified account. The Shopping Cart must contain at least one Product, and the inventory for each Product must be sufficient to fulfill the order. The payment information must be valid and up to date. The Customer must have completed the checkout process and selected the "Confirm Order" option.   
Postconditions: The Order is successfully confirmed and finalized. The inventory levels for the Products in the Order are updated, and the Order is recorded in the system. The Customer receives a confirmation email, and the Administrator is notified of the new Order.   
  
Main Flow:   
1. The Customer reviews the final confirmation page of the Order, which includes product details, quantity, price, and total amount.   
2. The Customer selects the "Confirm Order" button.   
3. The system verifies the availability of Products and the validity of the payment information.   
4. The system processes the Order and updates the inventory levels for the Products involved.   
5. The system records the Order in the database with the status set to "Confirmed".   
6. The system sends a confirmation email to the Customer with the Order details.   
7. The system notifies the Administrator of the new confirmed Order.   
8. The system displays a confirmation message to the Customer indicating that the Order was successfully confirmed.   
  
Alternative Flow:   
1. In step 2, if the Customer decides to cancel the confirmation, the system returns them to the cart or order summary page without processing the Order.   
2. In step 3, if the inventory for any Product is insufficient or the Product is out of stock, the system displays an error message and prompts the Customer to adjust the cart or try again later.   
3. In step 4, if the payment processing fails, the system displays an error message and cancels the Order.   
4. In step 5, if the system encounters an error while updating the inventory or recording the Order, it displays an error message and logs the issue for further investigation.  
  
Use Case Name: Manage Account   
Use Case ID: UC-26   
Actors: Customer, Administrator   
Preconditions: The system must be operational, and the Customer or Administrator must be logged in with a verified account. The Account data entity must exist in the database for the logged-in user.   
Postconditions: The Customer's or Administrator's account settings are successfully updated in the system, and any changes are reflected in the database and user interface.   
  
Main Flow:   
1. The Customer or Administrator navigates to the "Account Management" section from the account dashboard or user menu.   
2. The system displays a form with editable fields for account settings (e.g., password, email, account preferences).   
3. The user modifies the desired fields (e.g., updates their password, changes their email address, or adjusts preferences).   
4. The user submits the updated account settings form.   
5. The system validates the input data (e.g., password strength, email format).   
6. If valid, the system updates the Account information in the database.   
7. The system displays a confirmation message that the account settings were successfully updated.   
  
Alternative Flow:   
1. In step 3, if the user attempts to change their email address to one that is already registered, the system displays an error message and prompts the user to choose a different email.   
2. In step 5, if the input data is invalid or incomplete, the system displays an error message and prompts the user to correct the information.   
3. In step 6, if the system fails to update the account information, it displays an error message and logs the issue for further investigation.   
4. In step 2, if the user attempts to access the account management section without proper permissions, the system displays an error message and redirects them to a restricted access page.  
  
Use Case Name: Manage Payment Information   
Use Case ID: UC-27   
Actors: Customer, Administrator   
Preconditions: The system must be operational, and the Customer or Administrator must be logged in with a verified account. The Payment Information data entity must exist in the system for the logged-in user.   
Postconditions: The Customer's or Administrator's Payment Information is successfully added, edited, or removed from the system, and the changes are reflected in the database and user interface.   
  
Main Flow:   
1. The Customer or Administrator navigates to the "Payment Information Management" section from the account settings or dashboard.   
2. The system displays the current Payment Information entries associated with the user's account, along with options to add, edit, or delete entries.   
3. The user selects an option to add a new payment method, edit an existing one, or delete a payment method.   
4. If adding a new payment method, the system opens a form to collect payment details (e.g., credit card number, expiration date, CVV, billing address).   
5. If editing an existing payment method, the system opens the selected method in an editable form.   
6. If deleting a payment method, the system prompts the user to confirm the deletion.   
7. The user provides the required information or confirms the action.   
8. The system validates the input data (e.g., card format, expiration date, billing address).   
9. If valid, the system updates the Payment Information database accordingly (add, edit, or delete).   
10. The system refreshes the Payment Information display to reflect the changes.   
11. The system displays a confirmation message indicating the success of the action.   
  
Alternative Flow:   
1. In step 6, if the user cancels the deletion, the system does not perform any changes and returns to the Payment Information view.   
2. In step 4 or 5, if the user provides invalid or incomplete payment information, the system displays an error message and prompts for valid input.   
3. In step 9, if the system fails to update the Payment Information, it displays an error message and logs the issue for further investigation.   
4. In step 3, if the selected payment method does not exist, the system displays an error message and prompts the user to select a valid method.   
5. In step 3, if the Administrator attempts to manage another user's payment information without proper authorization, the system displays an error message and denies the action.  
  
Use Case Name: Manage Purchase History   
Use Case ID: UC-28   
Actors: Administrator   
Preconditions: The system must be operational, and the Administrator must be logged in with a verified account. Purchase History data must exist in the database for at least one Customer.   
Postconditions: The Administrator successfully views, filters, or modifies Purchase History records of Customers, and the changes are reflected in the system.   
  
Main Flow:   
1. The Administrator navigates to the "Manage Purchase History" section from the dashboard.   
2. The system displays a list of Customer purchase records with options to filter by Customer, date range, product category, or order status.   
3. The Administrator selects a specific Customer or enters filter criteria.   
4. The system retrieves the filtered purchase history from the database.   
5. The system displays the relevant purchase records in a structured format, including details such as product name, quantity, price, order date, and status.   
6. The Administrator can select an individual Order to view more details or modify its status.   
7. If the Administrator chooses to modify an Order status, the system provides a dropdown menu or input field for selecting the new status.   
8. The Administrator submits the updated status.   
9. The system validates the new status and updates the Order in the database.   
10. The system refreshes the purchase history view and displays a confirmation message that the status was successfully modified.   
  
Alternative Flow:   
1. In step 3, if the Administrator selects a Customer who has no purchase history, the system displays a message indicating that no records are available for the selected Customer.   
2. In step 6, if the selected Order ID is invalid or does not exist, the system displays an error message and prompts the Administrator to select a valid Order.   
3. In step 7, if the Administrator attempts to change an Order status to an invalid one, the system displays an error message and prompts for a valid status.   
4. In step 9, if the system encounters an error updating the Order status, it displays an error message and logs the issue for further investigation.   
5. In step 4, if no purchase records match the filter criteria, the system displays a message indicating that no data is available for the selected filters.